

UNIVERSIDAD DEL BÍO – BÍO

FACULTAD DE CIENCIAS EMPRESARIALES

DEPARTAMENTO DE AUDITORIA E INFORMATICA

**DESARROLLO DE AGENTE INTELIGENTE MOVIL,
GUIADO A TRAVES DE COORDENADAS
GEOGRAFICAS Y ANALISIS DE SU AMBIENTE**

MEMORIA PARA OPTAR AL TITULO DE INGENIERO DE EJECUCION
EN COMPUTACION E INFORMATICA

Juan Carlos Figueroa Durán
Sebastián Alexis Moraga Garrido

LUIS GAJARDO DIAZ
MARIA ANTONIETA SOTO CHICO

**CHILLAN
JUNIO, 2006**

Agradecimientos

En primer lugar agradezco a Dios por su infinita paciencia y amor en ayudarme a cumplir esta meta. Debo agradecer también a mi inagotable compañero de proyecto de título, Sebastián sin tu ayuda y comprensión hubiera sido imposible, gracias amigo. A mi madre por ser como sólo una madre puede ser, gracias por el alimento de cuerpo y alma que nos diste. A mi Abuelo, hombre de pocas palabras, pero muy sabias, gracias por formar al hombre que soy hoy, no hubiera podido pedir mejor padre. Paola, gracias por la comprensión, por la espera, tu fe ciega, en definitiva por el amor.

Juan Carlos Figueroa Durán.

Agradezco en primer lugar e infinitamente a mi Dios, por habernos dado toda la claridad mental y las capacidades para haber concretado este proyecto. Agradezco a mis padres, Malvina y Sebastián, y a mi hermana, Claudia, a mis abuelitos y a la familia Fuentealba, por haberme entregado su apoyo incondicional en palabras, oraciones y recursos básicos. Agradezco a mi hermano y amigo en la fe, Pedro (infame25), quien nos ayudó en la parte mecánica en un momento decisivo. Agradezco a Carlitos, Denisse, Donato, Douglas y Evelyn, quienes fielmente nos apoyaban con sus oraciones, y me dieron fuerzas con sus palabras de apoyo.

Sebastián Alexis Moraga Garrido.

Agradecemos a nuestro profesor Luis Gajardo, por apoyar el sueño que dio comienzo a esta aventura. Al profesor Elvis Aravena, quien puso la piedra angular en el análisis de la IA del robot. A Don Hugo, quien nos ayudó desinteresadamente a construir la estructura metálica del robot. Al grupo de robótica, al cual vimos nacer y seguiremos sus pasos con interés y orgullo, esperando que enaltezca aún más el nombre de nuestra Universidad.

En definitiva, muchas gracias.

Los Autores.

Dedicatoria

Este trabajo se lo dedico a mi abuela y madre, Irma Del Carmen Vergara Melo, la mujer que con su nombre enalteció a la muerte.

Juan Carlos Figueroa Durán.

A mi Dios, a mis padres, a mi hermana, a mis abuelos, a la familia Fuentealba, a mis amigos en la fe, a mi profesor Luis Gajardo, a mi profesor Elvis Aravena, y a mis compañeros de robótica.

Sebastián Alexis Moraga Garrido.

Resumen

En este trabajo, perteneciente al área de la robótica, se presenta el desarrollo de un Agente Autónomo Móvil, capaz de avanzar por un entorno estocástico, evitando obstáculos, mientras busca lograr cada uno de los puntos de paso para finalizar su recorrido, al alcanzar el objetivo de destino.

El principal problema abordado consiste en el diseño de un mecanismo que dote de “Inteligencia Artificial” al agente móvil. Este mecanismo estará a cargo de la herramienta llamada “MABE”, diseñada por los autores del documento, y presentada como posible solución, la cual se basa en una de las herramientas de Redes Neuronales llamada BAM.

Para obtener un esquema de su ambiente, se ha escogido como herramienta el ultrasonido. Además, deberá obtener su orientación mediante una brújula electrónica, y así lograr una referencia exacta de los puntos que deberá recorrer. Todos los sensores mencionados estarán subordinados a un microcontrolador. Específicamente, un microcontrolador llamado “Basic Stamp 2”, muy utilizado en robótica educativa. También, se utilizara tecnología de posicionamiento global para conseguir un concepto de ubicación espacial de sí mismo y de los objetivos que se deberán conseguir.

Referente a la movilidad del agente, ésta estará a cargo de dos tipos de efectores: motores DC y servomotores, para dirección, tracción y sensamientos.

Respecto del software, se empleó el lenguaje de modelamiento UML y se implementó a través del lenguaje de programación orientado a objetos Java, utilizando un ciclo de vida iterativo incremental.

La herramienta MABE resultó ser bastante adecuada para el problema planteado. Sin embargo, el tiempo dedicado al entrenamiento que se debe dar a ésta es bastante extenso y tedioso.

Finalmente, este tipo de proyectos se ven potenciados en un ambiente multidisciplinario.

Índice General

CAPITULO 1:	ANTECEDENTES GENERALES.....	16
1.1	INTRODUCCIÓN	16
1.2	DESCRIPCIÓN DEL PROBLEMA.....	19
1.3	OBJETIVOS DEL PROYECTO	20
1.3.1	<i>Objetivo General</i>	20
1.3.2	<i>Objetivos Específicos</i>	20
1.4	METODOLOGÍA UTILIZADA EN EL TRABAJO.....	20
1.5	TRABAJOS RELACIONADOS	20
CAPITULO 2:	MARCO TEÓRICO	24
2.1	CONCEPTOS Y TECNOLOGÍA.....	25
2.1.1	<i>Redes Neuronales</i>	25
2.1.1.1	Modelo de McCulloch-Pits	26
2.1.1.2	Teoría de Aprendizaje de Hebb.....	28
2.1.1.3	Estructura Básica de un Elemento de Procesamiento.....	29
2.1.1.4	Memoria Bidireccional Asociativa.....	31
2.1.2	<i>Comunicaciones Inalámbricas</i>	34
2.1.2.1	Wi-Fi.....	34
2.1.2.2	Bluetooth.....	35
2.1.2.3	IR	36
2.1.3	<i>Sistemas de Posicionamiento Global</i>	38
2.1.3.1	GPS	38
2.1.3.2	DGPS	39
2.1.3.3	Geodésico.....	40
2.1.3.4	Estaciones Radiales	41
2.1.4	<i>Microcontroladores</i>	42
2.1.5	<i>Sensores Ultrasónicos</i>	42
2.1.6	<i>Compás Magnético</i>	44
2.1.6.1	Magnetismo Terrestre	45
2.2	EVOLUCIÓN DE LA ROBÓTICA	45
2.2.1	<i>La Robótica en el Cine</i>	47
2.2.2	<i>Leyes de la Robótica de Isaac Asimov</i>	48
2.2.3	<i>Robótica al Servicio de la Ciencia de la Exploración</i>	49
2.2.4	<i>Auge de la Robótica</i>	50
2.3	APLICACIONES DE LAS TECNOLOGÍAS	51

CAPITULO 3:	DESARROLLO DEL PROYECTO	54
3.1	Lenguaje de Modelamiento a Utilizar	55
3.2	Fase de Planificación y Elaboración	55
3.2.1	<i>Conocimiento de Requerimientos del Sistema</i>	55
3.2.1.1	Presentación General	55
3.2.1.2	Clientes	55
3.2.1.3	Metas	56
3.2.1.4	Funciones del Sistema	56
3.2.1.5	Atributos del Sistema	57
3.2.1.6	Tabla Combinada	58
3.2.1.7	Observación	58
3.2.2	<i>Diagrama de Componentes del Sistema</i>	59
3.2.3	<i>Descripción de Procesos</i>	60
3.2.3.1	Diagramas de Casos de Uso	60
3.2.3.2	Casos de Uso de Alto Nivel	61
3.2.3.3	Casos de Uso Expandidos	64
3.2.4	<i>Clasificación y Programación de los Casos de Uso</i>	73
3.2.4.1	Asignación de los Casos de Uso a Incrementos	76
3.3	ANÁLISIS ORIENTADO A OBJETO: PRIMER INCREMENTO	76
3.3.1	<i>Modelo Conceptual</i>	77
3.3.2	<i>Diagramas de Secuencia de SiM</i>	77
3.3.3	<i>Diagramas de Secuencia del SAI</i>	78
3.4	DISEÑO ORIENTADO A OBJETO: PRIMER INCREMENTO	80
3.4.1	<i>Lenguaje de Programación</i>	80
3.4.2	<i>Definición de Casos de Uso Reales para el SiM</i>	81
3.4.3	<i>Diagrama de Clase del SiM</i>	86
3.4.3.1	Diagrama General de Paquetes	86
3.4.3.2	Diagrama de Clase (Paquete grafica)	87
3.4.3.3	Diagrama de Clase (Paquete logica)	88
3.4.3.4	Bean	89
3.4.3.5	Excepción	89
3.4.4	<i>Diagramas de Colaboración del SiM</i>	90
3.4.5	<i>Diagrama de Clase del SAI</i>	92
3.4.5.1	Diagrama General de Paquetes	92
3.4.5.2	Diagrama de Clase (Paquete comunicacion)	93
3.4.5.3	Diagrama de Clase (Paquete logica)	94
3.4.5.4	Diagrama de Clase (Paquete dispositivo)	95
3.4.5.5	Bean	96
3.4.5.6	Excepción	96

3.4.6	<i>Diagramas de Colaboración del SAI</i>	97
3.5	ANÁLISIS ORIENTADO A OBJETO: SEGUNDO INCREMENTO	99
3.5.1	<i>Modelo Conceptual</i>	99
3.5.2	<i>Diagramas de Secuencia de SiM</i>	100
3.5.3	<i>Diagramas de Secuencia de SAI</i>	103
3.6	DISEÑO ORIENTADO A OBJETO: SEGUNDO INCREMENTO.....	107
3.6.1	<i>Definición de Casos de Uso Reales para SiM</i>	107
3.6.2	<i>Diagrama de Clase SiM</i>	115
3.6.2.1	Diagrama General de Paquetes.....	115
3.6.2.2	Diagrama de Clase (Paquete logica).....	116
3.6.2.3	Diagrama de Clase (Paquete grafica)	117
3.6.2.4	Bean	118
3.6.2.5	Excepción.....	118
3.6.3	<i>Diagramas de Colaboración del SiM</i>	119
3.6.4	<i>Diagrama de Clase SAI</i>	124
3.6.4.1	Diagrama General de Paquetes.....	124
3.6.4.2	Diagrama de Clase (Paquete Comunicacion)	125
3.6.4.3	Diagrama de Clase (Paquete Dispositivo)	126
3.6.4.4	Diagrama de Clase (Paquete Logica)	127
3.6.4.5	Diagrama de Clase (Paquete BaseDeConocimiento).....	128
3.6.4.6	Beans.....	129
3.6.4.7	Excepciones	130
3.6.5	<i>Modelo Entidad Relación</i>	131
3.6.6	<i>Diagramas de Colaboración del SAI</i>	131
CAPITULO 4: IMPLEMENTACIÓN		164
4.1	IDE	165
4.2	MEMORIA DEL AGENTE INTELIGENTE	165
4.3	SELECCIÓN DE LA TÉCNICA ADECUADA PARA LA INTELIGENCIA DEL AGENTE INTELIGENTE	166
4.3.1	<i>Ejemplo del Funcionamiento de la MABE</i>	169
4.4	IMPLEMENTACIÓN DE LA BS2	171
4.5	SENSORES ULTRASÓNICOS	174
4.6	BRÚJULA.....	175
4.7	DGPS.....	175
4.8	ESTRUCTURA FÍSICA DEL AGENTE INTELIGENTE	177
CAPITULO 5: PRUEBAS.....		180
CAPITULO 6: CONCLUSIONES.....		182
6.1	CONCLUSIONES GENERALES	183

6.2	TRABAJOS FUTUROS	184
	BIBLIOGRAFÍA	185
	ABREVIATURAS	188
	GLOSARIO	189
	ANEXO A: PRUEBAS.....	A-1
	ESPECIFICACIÓN DE CASO DE PRUEBA COMUNICACIÓN CON BASICSTAMP 2.	A-2
	ESPECIFICACIÓN DE CASO DE PRUEBA: INTELIGENCIA DEL AUTÓMATA.	A-11
	ESPECIFICACIÓN DE CASO DE PRUEBA: SERVICIOS A CLIENTES DE MONITOREO.....	A-19
	ANEXO B: MABE	B-1
	DEFINICIÓN FORMAL DE MEMORIA DE ASOCIACIÓN CON BÚSQUEDA EUCLIDIANA EN HIPERESPACIOS	B-2

Índice Figuras

Figura 1-1. Shake junto al equipo desarrollador.....	21
Figura 1-2. Equipo de la Carnegie Mellon University.	21
Figura 1-3. Stanford Cart.....	22
Figura 1-4. Evolución de los robots de Honda.	23
Figura 1-5. Evolución de los robots de Sony.	23
Figura 2-1. Divergencia y Convergencia.....	26
Figura 2-2. Realimentación.	27
Figura 2-3. Ejemplo de la teoría de aprendizaje de Hebb.	28
Figura 2-4. Estructura básica de un PE.....	29
Figura 2-5. Estructura de una BAM.	31
Figura 2-6. Ejemplo de satélites GPS.....	39
Figura 2-7. Ejemplo DGPS.....	40
Figura 2-8. Esquema datum del geoide.	41
Figura 2-9. Triangulación a través cuatro estaciones radiales.....	41
Figura 2-10. Estructura de un microcontrolador.	42
Figura 2-11. Brújula o Compás magnético.....	44
Figura 2-12. Campo magnético terrestre.	45
Figura 2-13. Juego de Football Socker de ROBOCUP modalidad humanoides.....	50
Figura 2-14. Robot COG, del MIT.....	51
Figura 3-1. Diagrama de Componentes de Sistema.	59
Figura 3-2. Diagrama de Casos de Uso de SAI.....	60
Figura 3-3. Diagrama de Casos de Uso de SiM.	61
Figura 3-4. Diagrama de asignación de casos de uso a los incrementos.....	76
Figura 3-5. Diagrama de Modelo Conceptual SiM.	77
Figura 3-6. Diagrama de Modelo Conceptual SAI.....	77
Figura 3-7. Diagrama Monitorear SAI.	77
Figura 3-8. Diagrama Iniciar SiM.	78
Figura 3-9. Diagrama Consultar estado interno.....	78

Figura 3-10. Diagrama Iniciar SAI.....	79
Figura 3-11. Diagrama Conectar con SAI.....	79
Figura 3-12. Sistema de Monitoreo sin Conexión.....	81
Figura 3-13. Sistema de Monitoreo con Conexión.....	82
Figura 3-14. Sistema de Monitoreo con Error en Sistemas Complementarios.	83
Figura 3-15. Diálogo de Ingreso del Password de Sistema de Monitoreo.	84
Figura 3-16. Diagrama general de paquetes.	86
Figura 3-17. Diagrama de Clases, paquete “grafica”.	87
Figura 3-18. Diagrama de Clases, paquete “logica”.....	88
Figura 3-19. Clases Bean del problema.....	89
Figura 3-20. Clase de excepción del problema.....	89
Figura 3-21. Diagrama de Colaboración monitorizar.....	90
Figura 3-22. Diagrama de Colaboración autenticar.	91
Figura 3-23. Diagrama general de paquetes.	92
Figura 3-24. Diagrama de clases, paquete “comunicacion”.....	93
Figura 3-25. Diagrama de clases, paquete “logica”.....	94
Figura 3-26. Diagrama de clases, paquete “dispositivo”.....	95
Figura 3-27. Clases Bean del problema.....	96
Figura 3-28. Clase de excepción del problema.....	96
Figura 3-29. Diagrama de Colaboración obtener estado interno.....	97
Figura 3-30. Diagrama de Colaboración establecer conexión.....	98
Figura 3-31. Diagrama de Modelo Conceptual SiM.	99
Figura 3-32. Diagrama de Modelo Conceptual SAI.....	99
Figura 3-33. Diagrama Monitorear SAI.	100
Figura 3-34. Diagrama Iniciar SiM.	100
Figura 3-35. Diagrama probar conexión con sistemas complementarios.....	101
Figura 3-36. Diagrama Ejecutar protocolo de inicio.	101
Figura 3-37. Diagrama Detener agente inteligente.....	102
Figura 3-38. Diagrama Probar efectores del SAI.	102
Figura 3-39. Diagrama Probar sensores del SAI.	102
Figura 3-40. Diagrama Consultar estado interno.....	103

Figura 3-41. Diagrama Iniciar SAI.....	103
Figura 3-42. Diagrama probar conexión con sistemas complementarios.....	104
Figura 3-43. Diagrama Conectar con SAI.....	104
Figura 3-44. Diagrama detener agente inteligente manual.....	104
Figura 3-45. Diagrama detener agente inteligente.....	105
Figura 3-46. Diagrama probar efectores del SAI.	105
Figura 3-47. Diagrama probar sensores del SAI.	105
Figura 3-48. Diagrama probar conexión con SAI.	106
Figura 3-49. Sistema de Monitoreo sin Conexión.....	107
Figura 3-50. Sistema de Monitoreo con Conexión.....	108
Figura 3-51. Sistema de Monitoreo con Error en Sistemas Complementarios.	109
Figura 3-52. Dialogo de Ingreso del Password de Sistema de Monitoreo.	110
Figura 3-53. Diagrama general de paquetes.	115
Figura 3-54. Diagrama de Clases, paquete “logica”.....	116
Figura 3-55. Diagrama de Clases, paquete “grafica”.	117
Figura 3-56. Clases Bean del problema.....	118
Figura 3-57. Clase de excepción del problema.....	118
Figura 3-58. Diagrama de Colaboración Monitorizar.	119
Figura 3-59. Diagrama de Colaboración Autenticar.....	120
Figura 3-60. Diagrama de Colaboración probar conexión con el SAI.....	121
Figura 3-61. Diagrama de Colaboración probar conexión con sistemas complementarios.	121
Figura 3-62. Diagrama de Colaboración probar efectores.	122
Figura 3-63. Diagrama de Colaboración probar sensores.	123
Figura 3-64. Diagrama general de paquetes.	124
Figura 3-65. Diagrama de Clases, paquete “Comunicacion”.....	125
Figura 3-66. Diagrama de Clases, paquete “Dispositivo”.....	126
Figura 3-67. Diagrama de Clases, paquete “Logica”.	127
Figura 3-68. Diagrama de Clases, paquete “BaseDeConocimiento”.	128
Figura 3-69. Diagrama de Clases“Beans”.....	129
Figura 3-70. Diagrama de Clases“Beans”.....	130

Figura 3-71. Diagrama de Clases “Excepciones”.....	130
Figura 3-72. Diagrama Modelo Entidad Relación.....	131
Figura 3-73. Diagrama de Colaboración obtener estado interno.....	132
Figura 3-74. Diagrama de Colaboración establecer conexión.....	133
Figura 3-75. Diagrama de Colaboración probar conexión con memoria.	134
Figura 3-76. Diagrama de Colaboración probar conexión con microcontrolador, primera parte.	135
Figura 3-77. Diagrama de Colaboración probar conexión con microcontrolador, segunda parte.	136
Figura 3-78. Diagrama de Colaboración probar conexión con microcontrolador, tercera parte y final.	137
Figura 3-79. Diagrama de Colaboración detener agente inteligente, primera parte.....	138
Figura 3-80. Diagrama de Colaboración detener agente inteligente, segunda parte.	139
Figura 3-81. Diagrama de Colaboración detener agente inteligente, tercera parte.	140
Figura 3-82. Diagrama de Colaboración detener agente inteligente, cuarta parte.	141
Figura 3-83. Diagrama de Colaboración detener agente inteligente, quinta parte.	142
Figura 3-84. Diagrama de Colaboración detener agente inteligente, sexta parte y final....	143
Figura 3-85. Diagrama de Colaboración probar dirección, primera parte.....	144
Figura 3-86. Diagrama de Colaboración probar dirección, segunda parte.....	145
Figura 3-87. Diagrama de Colaboración probar dirección, tercera parte.	146
Figura 3-88. Diagrama de Colaboración probar dirección, cuarta parte.	147
Figura 3-89. Diagrama de Colaboración probar dirección, quinta parte y final.....	148
Figura 3-90. Diagrama de Colaboración probar tracción, primera parte.....	148
Figura 3-91. Diagrama de Colaboración probar tracción, segunda parte.....	149
Figura 3-92. Diagrama de Colaboración probar tracción, tercera parte.	150
Figura 3-93. Diagrama de Colaboración probar tracción, cuarta parte.	151
Figura 3-94. Diagrama de Colaboración probar tracción, quinta parte y final.....	152
Figura 3-95. Diagrama de Colaboración probar sensores ultrasónicos de ambiente, primera parte.	153
Figura 3-96. Diagrama de Colaboración probar sensores ultrasónicos de ambiente, segunda parte.	154

Figura 3-97. Diagrama de Colaboración probar sensores ultrasónicos de ambiente, tercera parte.	155
Figura 3-98. Diagrama de Colaboración probar sensores ultrasónicos de ambiente, cuarta parte y final.	156
Figura 3-99. Diagrama de Colaboración probar sensor ultrasónico de terreno, primera parte.	157
Figura 3-100. Diagrama de Colaboración probar sensor ultrasónico de terreno, segunda parte.	158
Figura 3-101. Diagrama de Colaboración probar sensor ultrasónico de terreno, tercera parte.	159
Figura 3-102. Diagrama de Colaboración probar sensor ultrasónico de terreno, cuarta parte y final.	160
Figura 3-103. Diagrama de Colaboración probar brújula.	161
Figura 3-104. Diagrama de Colaboración probar DGPS.	162
Figura 3-105. Diagrama de Colaboración probar conexión con SAI.	163
Figura 4-1. Imagen representativa del conjunto de ejemplares almacenados.	168
Figura 4-2. Imagen representativa de un punto problema graficado en el espacio de la MABE.	168
Figura 4-3. Imagen representativa de los distintos conjuntos de puntos pertenecientes a diversos patrones almacenados en la MABE.	169
Figura 4-4. Diagrama de Estados del Servidor en el Microcontrolador BS2.	172
Figura 4-5. Protocolo de comunicación entre el servidor en el microcontrolador BS2 y el cliente en el computador del software del Agente Inteligente.	173
Figura 4-6. Imagen de ejemplo de la reflexión del sonido sobre una superficie irregular.	174
Figura 4-7. Modelo CAD de la estructura física del Agente Inteligente y su implementación.	177
Figura 4-8. Pieza para la adaptación del engrane de tracción del motor.	178
Figura 4-9. Implementación de la dirección del Agente Inteligente.	179

Índice Tablas

Tabla 2-1. Notación mediante lógica proposicional.....	27
Tabla 3-1. Funcionalidades del sistema.....	56
Tabla 3-2. Atributos del sistema.....	57
Tabla 3-3. Tabla combinada entre requerimientos funcionales y no funcionales del sistema.	58
Tabla 3-4. Actores de los casos de uso.....	61
Tabla 3-5. Casos de uso “Iniciar el SAI”.....	62
Tabla 3-6. Casos de uso “Probar Conexión con Sistemas Complementarios”.....	62
Tabla 3-7. Casos de uso “Probar Efectores del SAI”.	62
Tabla 3-8. Caso de uso “Probar Sensores del SAI”.....	62
Tabla 3-9. Caso de uso “Conectar con el SAI”.	62
Tabla 3-10. Caso de uso “Probar conexión con el SAI”.	62
Tabla 3-11. Caso de uso “Consultar Estado Interno”.....	63
Tabla 3-12. Caso de uso “Detener al SAI”.....	63
Tabla 3-13. Caso de uso “Detener al SAI Manualmente”.....	63
Tabla 3-14. Caso de uso “Iniciar el SiM”.....	63
Tabla 3-15. Caso de uso “Probar Conexión con Sistemas Complementarios”.	63
Tabla 3-16. Caso de uso “Probar Efectores del SAI”.....	64
Tabla 3-17. Caso de uso “Probar Sensores del SAI”.....	64
Tabla 3-18. Caso de uso “Monitorear el SAI”.....	64
Tabla 3-19. Caso de uso “Detener al Agente Inteligente”.....	64
Tabla 3-20. Caso de uso “Ejecutar Protocolo de Inicio”.....	64
Tabla 3-21. Caso de uso “Consultar Estado Interno”.....	65
Tabla 3-22. Caso de uso “Iniciar SAI”.....	65
Tabla 3-23. Caso de uso “Probar Conexión con Sistemas Complementarios”.	66
Tabla 3-24. Caso de uso “Conectar Con SAI”.	66
Tabla 3-25. Caso de uso “Detener a Agente Inteligente Manualmente”.....	67
Tabla 3-26. Caso de uso “Detener a Agente Inteligente”.....	67

Tabla 3-27. Caso de uso “Probar Efectores del SAI”	68
Tabla 3-28. Caso de uso “Probar Sensores del SAI”	68
Tabla 3-29. Caso de uso “Probar Conexión con SAI”.	69
Tabla 3-30. Caso de uso “Monitorear SAI”.	69
Tabla 3-31. Caso de uso “Iniciar SiM”.	70
Tabla 3-32. Caso de uso “Probar Conexión con Sistemas Complementarios”.	70
Tabla 3-33. Caso de uso “Ejecutar Protocolo de Inicio”	71
Tabla 3-34. Caso de uso “Detener a Agente inteligente”	71
Tabla 3-35. Caso de uso “Probar Efectores del SAI”	72
Tabla 3-36. Caso de uso “Probar Sensores del SAI”	72
Tabla 3-37. Clasificación de los casos de uso del SAI	74
Tabla 3-38. Clasificación de los casos de uso del SiM.	75
Tabla 3-39. Caso de uso “Monitorear SAI”.	84
Tabla 3-40. Caso de uso “Iniciar SiM”.	85
Tabla 3-41. Caso de uso “Probar Conexión con Sistemas Complementarios”.	111
Tabla 3-42. Caso de uso “Detener Agente inteligente”	112
Tabla 3-43. Caso de uso “Probar Efectores del SAI”	113
Tabla 3-44. Caso de uso “Probar Sensores del SAI”	114
Tabla 4-1 Definición BNF del protocolo de comunicación entre el servidor en el microcontrolador BS2 y el cliente en el computador del software del agente inteligente.	173
Tabla 4-2. Composición del código hacia el cliente desde el servidor en el microcontrolador BS2.	174
Tabla 4-3. Resultados de mediciones realizados con dispositivo DGPS en distintos ambientes.	176
Tabla 4-4. Materiales para la implementación de la estructura física.	178

CAPITULO 1: ANTECEDENTES GENERALES

CAPITULO 1: ANTECEDENTES GENERALES

1.1 Introducción

En las últimas dos décadas, el área de la inteligencia artificial ha tenido un fuerte desarrollo. Una de las áreas donde podemos ver reflejado este crecimiento es la robótica, la cual debido a su carácter multidisciplinario ha agrupado un gran número de adeptos, gracias a su aplicación, por ejemplo, en la automatización de procesos industriales.

El desarrollo de la robótica y la automatización dentro de nuestra Universidad está teniendo un fuerte auge, por parte de los campos de la Mecánica y la Electrónica. Sin embargo, áreas tan importantes de la Informática como la Inteligencia Artificial (Visión Artificial, Razonamiento y Aprendizaje Automáticos), Computación Gráfica y Redes de Comunicación, no han sido aplicados a los proyectos realizados.

Por lo anterior, nuestro proyecto está orientado a la integración de las distintas disciplinas expuestas, dándole especial atención al desarrollo de la informática en sus áreas mencionadas, obteniendo, así, un agente robótico capaz de tomar decisiones con mayor flexibilidad, autonomía e inteligencia. Para más detalles técnicos sobre las tecnologías aplicadas aquí, véase el CAPITULO 2.

Dado que el desarrollo de la robótica dentro de las universidades más importantes del país ha estado orientado mayoritariamente al desempeño en ambientes controlados, no existe gran información para realizar comparaciones con el presente proyecto. El hecho de que no exista un desarrollo mayor de algún proyecto semejante se debe a que gran parte de estos desarrolladores son de disciplinas distintas a la informática, lo que implica que los sistemas robóticos involucrados resultan ser más sencillos. Sin embargo, cabe destacar lo realizado en el proyecto IRMA (Investigación Robótica Móvil Autónoma), de la Universidad Técnica Federico Santa María. Allí se estudian y desarrollan algoritmos evolutivos multi-objetivos para robots móviles autónomos. Es decir, el estudio y desarrollo de algoritmos basados en la evolución natural para mejorar las capacidades de desempeño de los robots móviles actuales.

Para la industria y la ingeniería este es un problema de especial interés, puesto que se desea utilizar robótica móvil para aplicaciones que muchas veces son repetitivas, riesgosas, insalubres, imposibles o simplemente comunes y sencillas. Cada una de estas aplicaciones están

CAPITULO 1: ANTECEDENTES GENERALES

compuestas de muchos objetivos (multi-objetivos), los cuales tienen que ser cumplidos para poder efectuar la aplicación de interés. Algunos de estos objetivos son la navegación a un objetivo, modelamiento ambiental, detección de objetos y el mover objetos a puntos específicos. (Tomás Arredondo V. *et al*, 2006)

En Chile, existen iniciativas que impulsan el desarrollo de la robótica. Algunas de estas iniciativas toman formas de competencias (Grupo Robótica, 2005) (UTFSM, 2005)(PUC, 2006), que motivan a las mentes de grupos interesados a innovar y crear nuevas soluciones. Uno de estos eventos, y el más cercano que se puede destacar, fue el realizado en Talca, llamado “Concurso de Robótica Agrícola” (UT, 2005), organizado por el Centro de alumnos de la carrera de Ingeniería Civil en Computación de la Universidad de Talca. Su primera versión se desarrolló los días 17 y 18 de Diciembre de 2004. El objetivo de este concurso es promover el desarrollo de la tecnología agrícola de la región, a través del desarrollo de robots agrícolas, un ejemplo de esto fue el desafío del año 2005, el cual exigía la capacidad de recorrer hileras de viñas.

Este informe está organizado como sigue:

El “CAPITULO 2: MARCO TEÓRICO” hace referencia a las explicaciones de las tecnologías aplicadas al desarrollo del proyecto; los avances evidenciados en la robótica; y las aplicaciones de los puntos anteriores al proyecto.

El “CAPITULO 3: DESARROLLO DEL PROYECTO” se refiere al desarrollo de la solución propuesta, donde se expondrán los puntos de análisis y diseño, los cuales se abordarán por incremento.

En el “CAPITULO 4: IMPLEMENTACIÓN” se explica con detalle cada una de las tecnologías utilizadas (sensores ultrasónicos y compás, GPS, comunicación wireless, software utilizados y técnicas de inteligencia artificial) en la implementación de la solución y justificando la utilización de cada una de ellas.

En el “CAPITULO 5: PRUEBAS” se exponen las pruebas JUnit realizadas a las funcionalidades más relevantes de la solución.

CAPITULO 1: ANTECEDENTES GENERALES

Al final del documento, se encuentra el “CAPITULO 6: CONCLUSIONES”, el cual da a conocer las inferencias obtenidas del desarrollo de este proyecto.

1.2 Descripción del Problema

El problema planteado en este proyecto está dado por la guía de un robot móvil en un ambiente estocástico, esto es, en un ambiente no controlado. El ambiente estocástico estará dado por objetos móviles y objetos estáticos que obstaculicen el desplazamiento del robot hacia su destino, dado por un punto geográfico predefinido. La ruta seguida por el robot, para conseguir llegar a su destino, estará dada por puntos geográficos intermedios predeterminados con los cuales podrá verificar el correcto enrutamiento, tal ruta sólo será una guía general (esbozo) a seguir por parte del robot. El robot, para estos efectos, deberá poseer sensores que le permitan identificar los obstáculos presentes en su ambiente y así tomar decisiones con respecto a ellos, a fin de llegar, por la ruta general previamente definida, a su destino final.

Los aportes que se pretenden dejar con la solución a este problema son los siguientes:

- Sentar el precedente de que se pueden realizar proyectos de esta índole en nuestro Departamento.
- Demostrar que los conocimientos adquiridos en la formación profesional son los básicos para desarrollar este tipo de proyectos.
- Innovar en los temas de aplicación de la informática en los proyectos de título hacia el área de la robótica.
- Dejar infraestructura para desarrollar con más detalle el tema que trata el proyecto.
- Dejar una guía donde se explique cómo desarrollar proyectos de este tipo.
- Dejar este proyecto como un medio de difusión de los desarrollos tecnológicos realizados en nuestra Universidad.

1.3 Objetivos del Proyecto

1.3.1 Objetivo General

Desarrollar un Agente Inteligente capaz de desplazarse en un ambiente estocástico.

1.3.2 Objetivos Específicos

1. Investigar las tecnologías necesarias más adecuadas para comunicaciones, para Inteligencia Artificial y para SIGs.
2. Construir un manual de referencias que contenga el procedimiento de uso de las tecnologías involucradas en este trabajo.
3. Desarrollar un software con la inteligencia artificial necesaria para guiar un robot móvil a través de un ambiente estocástico.
4. Construir un software de monitoreo que permita, a través de comunicación inalámbrica, saber el estado del robot en tiempo real (ubicación geográfica, información de sensores, decisiones tomadas).
5. Utilizar y modificar un robot móvil construido por el Grupo de Robótica UBB - Chillán¹.

1.4 Metodología Utilizada en el Trabajo

La metodología a utilizar será el desarrollo incremental, y el enfoque empleado para el desarrollo del software será la Orientación a Objetos (OO). El modelado de análisis y diseño estarán bajo los estándares de UML. Para lograr la implementación de OO se empleará el lenguaje Java de SunMicrosystem (Java, 2006), codificando bajo sus estándares.

1.5 Trabajos Relacionados

En esta sección se encuentran los trabajos más destacables realizados por instituciones relacionadas con el tema desarrollado en este documento. Éstos se expondrán en orden cronológico dando a conocer el trabajo histórico sobre ésta área hasta la actualidad.

- **1966:** El Stanford Research Institute (luego conocido como SRI Technology) crea a Shakey (véase Figura 1-1), el primer robot móvil conocido que tomaba sus propias decisiones. Éste se desempeñaba en ambientes controlados interiores. (SRI Internacional, 2006)

¹ **Grupo de Robótica UBB – Chillán:** Grupo de alumnos de la Universidad del Bío-Bío, sede Chillán, orientado al desarrollo de proyectos relacionados con robótica.

CAPITULO 1: ANTECEDENTES GENERALES



Figura 1-1. Shake junto al equipo desarrollador.

- **Reto DARPA:** DARPA (Defense Advanced Research Projects Agency) es una Agencia dependiente del Ministerio de Defensa de los EEUU fundada después de que la ex-URSS pusiera a Sputnik en órbita. Esta Agencia incentiva el desarrollo de proyectos tecnológicos. El reto DARPA (DARPA Grand Challenge) (DARPA, 2005) es una competición que apunta a crear vehículos completamente autónomos capaces de ir desde Los Ángeles a Las Vegas, a través del desierto de Mojave (véase Figura 1-2). Este desafío anual se celebró por primera vez el 13 de Marzo de 2004.



Figura 1-2. Equipo de la Carnegie Mellon University.

El equipo de Carnegie Mellon University (véase Figura 1-2), llegó hasta el kilómetro 12 durante el reto de Marzo de 2004. El mejor clasificado, empleó un vehículo HUMVEE que, finalmente, se estrelló con una roca y sufrió un incendio que fue sofocado por los miembros del equipo.

CAPITULO 1: ANTECEDENTES GENERALES

Otro evento fue realizado el 9 de Octubre de 2005. El próximo desafío se lanzará en 2007, y tendrá un formato distinto al ya conocido, dado que será en un sector urbano.

- **1970:** Stanford University produce el Stanford Cart (véase Figura 1-3). Este fue diseñado con el propósito de ser capaz de seguir una línea, y a su vez era controlado desde una computadora por medio de radio frecuencias (History of Computing Foundation, 2005). Para mayor detalle sobre radio frecuencias véase la sección 2.1.2.1.



Figura 1-3. Stanford Cart.

- **1979:** El Stanford Cart fue reconstruido por Hans Moravec y añade un sistema de visión más robusto, el cual permite una mayor autonomía. Este sistema, fue una de las primeras experiencias de mapeo 3D de ambientes. (History of Computing Foundation, 2005)
- **1986:** Honda comienza con un programa de investigación y posterior construcción de un robot que se sustenta en la premisa de que el robot "debería coexistir y cooperar con los seres humanos, haciendo lo que una persona no puede y cultivando una nueva dimensión en movilidad para el beneficio final de la sociedad". (Departamento de Ciencia de la Computación e Inteligencia Artificial. 2005)
- **1987:** Harris, C. B., Goss, R., Krakiwsky, E. J., y Karimi, H. A., participan del desarrollo del Sistema de Información Automatic Vehicle Location (AVL), el cual estaba concebido para determinar la ruta a seguir por un vehículo.
- **1989:** Un robot caminante llamado Genghis es develado por el Mobile Robots Group del MIT. Éste llegó a ser conocido como "the way it walk", y popularmente como "Genghis gait". (History of Computing Foundation, 2005)

CAPITULO 1: ANTECEDENTES GENERALES

- **1993:** P2, el primer humanoide que causó gran impacto en el público que lo veía con sus movimientos humanos realistas (Honda Robots, 2006). (véase Figura 1-4)
- **1993:** P3, fue la evolución en tamaño y peso de su antecesor P2 (Honda Robots, 2006). (véase Figura 1-4)
- **2000:** ASIMO evoluciona en tamaño, peso y la tecnología de caminar de P3 (Honda Robots, 2006). (véase Figura 1-4)
- **2000:** Sony devela SDR-3X humanoide, mostrado por Sony Dream Robots (SDR), en la exposición de Robodex (Sony, 2006a). (véase Figura 1-5)
- **2001:** Sony lanza la segunda generación de sus perros robot Aibo (Sony, 2006b). (véase Figura 1-5)
- **2003:** El 10 de Junio la NASA lanza el MER-A "Spirit", carro autónomo destinado a llegar a Marte. El 7 de Julio la NASA lanza el MER-B "Opportunity" (Nasa, 2006).



Figura 1-4. Evolución de los robots de Honda.



Figura 1-5. Evolución de los robots de Sony.

CAPITULO 2: MARCO TEÓRICO

2.1 Conceptos y Tecnología

En esta sección del informe se darán a conocer las tecnologías utilizadas en este proyecto, con el objetivo de normalizar conocimientos para facilitar al lector la comprensión del documento.

2.1.1 Redes Neuronales

Las figuras y la mayoría de los textos presentados bajo este título fueron extraídos de (Freeman y Skapura, 1993).

“La aparición de los computadores digitales y el desarrollo de las modernas teorías acerca del aprendizaje y del procesamiento neuronal se produjeron aproximadamente al mismo tiempo, a fines de los años cuarenta. A partir de este momento, los computadores digitales han sido utilizados como herramientas para modelar neuronas individuales, así como agrupaciones de neuronas, que se denominan redes neuronales. Desde entonces se han acumulado una gran cantidad de investigaciones neurofisiológicas.”

“Las técnicas de Redes Neuronales toman conceptos de las investigaciones neurológicas y las aplica a la resolución de problemas pertenecientes a algunas ramas de las ciencias y la ingeniería.”

“Los modelos de neuronas artificiales (ANS, por sus siglas en inglés) actuales han sido inspiradas totalmente en la comprensión que hasta el momento se tiene del funcionamiento neuronal biológico. Sin embargo, estos modelos de ANS no tienen, necesariamente, que apearse en su totalidad a esta comprensión.”

“Los primeros ejemplos de estos nuevos sistemas aparecieron al final de la década de los cincuenta. La referencia histórica más corriente es la que alude al trabajo realizado por Frank Rosenblatt en un dispositivo denominado *perceptrón*. Hay otros ejemplos, sin embargo, tales como el desarrollo del Adaline por el profesor Bernard Widrow.”

Desde el año 1969 hasta principios de los ochenta la investigación sobre esta técnica se desvaneció. La razón, suele pensarse que fue la aparición del libro *Perceptrons*, escrito por Marvin Minsky y Seymour Papert (Minsky y Papert, 1969). Gracias a la aparición de este texto y

CAPITULO 2: MARCO TEÓRICO

sus críticas sobre las limitadas capacidades del perceptrón, se piensa, fueron el freno que restringió todas aquellas investigaciones que pudieron haber seguido los pasos de las investigaciones neurológicas. Sin embargo, durante este periodo glacial para esta técnica hubo brotes aislados de investigaciones que siguieron adelante. En la actualidad se ha dado un nuevo realce a las investigaciones sobre las redes neuronales, esto debido a la asignación de fondos significativos, motivación por parte de las grandes universidades a sus alumnos, y un nuevo grupo de profesionales interesados en las investigaciones. Otro incentivo para que se retomaran las investigaciones en el área fue la aparición en 1986 del libro *Parallel Distributed Processing (PDP)*, vols. I y II, escrito por David Rumelhart y James McClelland (Rumelhart y McClelland, 1986).

“Desde hace mucho tiempo las personas que diseñan sistemas computacionales se han preguntado cómo se podrían hacer computadoras más inteligentes, que sean capaces de reconocer formas de imágenes, incluso distinguir entre clases de objetos similares, aprender desde experiencias en lugar de repetir indefinidamente un conjunto explícito de instrucciones generado por algún programador humano.”

2.1.1.1 Modelo de McCulloch-Pits

“Dentro de la evolución de este tipo de tecnologías, encontramos un modelo de redes neuronales que corresponde a las bases de una gran evolución: El modelo de McCulloch – Pitts.”

“Este modelo trata el tema de las redes neuronales realizando una combinación con el álgebra de Boole, como se puede apreciar en la Figura 2-1 y Figura 2-2 y en la Tabla 2-1.”

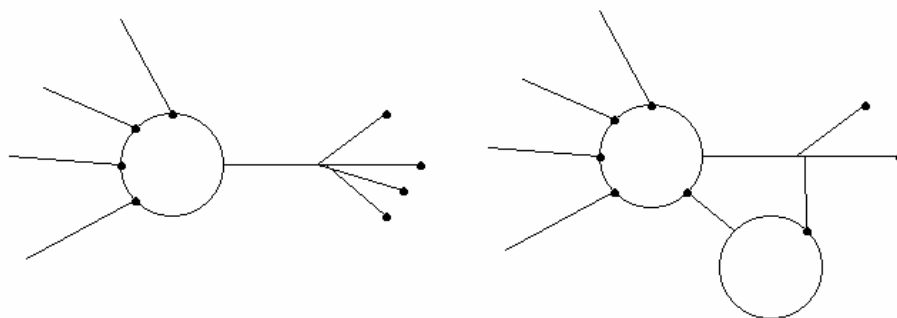


Figura 2-1. Divergencia y Convergencia.

CAPITULO 2: MARCO TEÓRICO

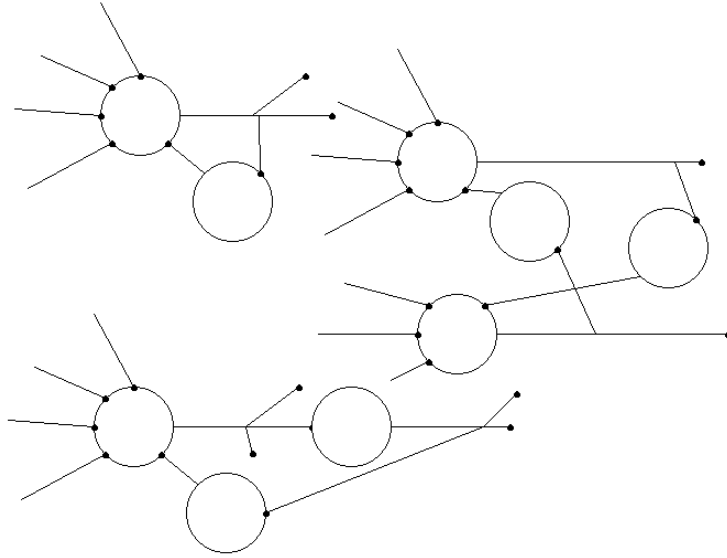


Figura 2-2. Realimentación.

	$N_2(t) = N_1(t-1)$
	$N_3(t) = N_1(t-1) \vee N_2(t-1)$
	$N_3(t) = N_1(t-1) \wedge N_2(t-1)$
	$N_3(t) = N_1(t-1) \wedge \neg N_2(t-1)$

Tabla 2-1. Notación mediante lógica proposicional.

“La tecnología de las redes neuronales, al ser inspirada directamente de los estudios de la neurofisiología, considera métodos para el aprendizaje y entrenamiento de estas estructuras, con lo cual se consigue mayor adaptabilidad y flexibilidad por parte de los sistemas que las utilicen.”

2.1.1.2 Teoría de Aprendizaje de Hebb

“La base de aprendizaje de las Redes Neuronales se encuentra en la teoría de Hebb, y corresponde a lo siguiente: Dada una red de neuronas, Hebb postula que si una neurona no participa en la estimulación de otra, pero ésta está próxima a ella y cuando ella está estimulada esta también se estimula, la unión entre esta y ella comienza a crecer.”

Ej.: Sea A , B y C neuronas. Sean S_{BA} y S_{BC} unidades sinápticas entre A y B , y B y C respectivamente. Sean una entrada sonora, otra visual y una salida para la salivación de un individuo determinado, la red neuronal resultante se vería como se muestra en la Figura 2-3.

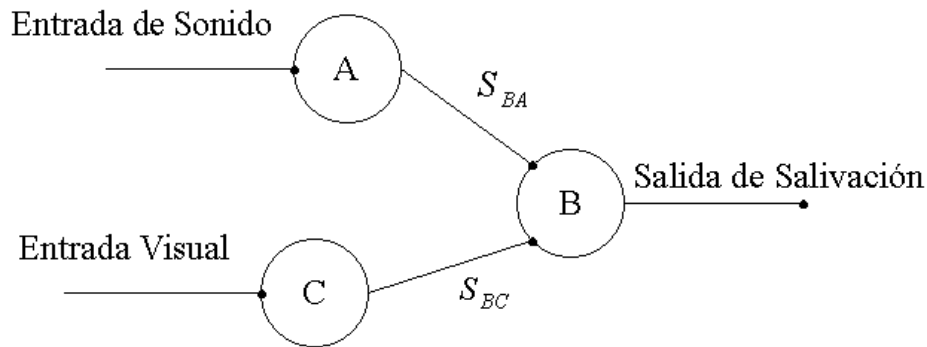


Figura 2-3. Ejemplo de la teoría de aprendizaje de Hebb.

Si al individuo se le estimula visualmente con comida, ocurre un proceso de salivación; dado que ocurre, con, por ejemplo, un plato de comida una estimulación de la neurona C , lo que implica una señal en la unión S_{BC} , que finalmente permite que se estimule la neurona B . Dado que este camino ya está creado.

Si, adicionalmente, a la estimulación se le añade una variable de sonido, se produciría una estimulación de la neurona A que provocaría una señal en la unión S_{BA} . Sin embargo, la unión

S_{BA} aún no existe, por lo que, como ahora, A comienza a tener participación cuando ocurre una señal S_{BC} , la unión S_{BA} se fortalece a través del condicionamiento.

2.1.1.3 Estructura Básica de un Elemento de Procesamiento

“Los elementos primordiales de una Red Neuronal son las estructuras comúnmente llamados nodos, unidades o elementos de procesamiento (PEs.), y es un error pensar que los PEs. mantienen una relación biunívoca con las neuronas biológicas. En ocasiones es mejor tomar los PEs. como representaciones de la actividad colectiva de un grupo de neuronas.”

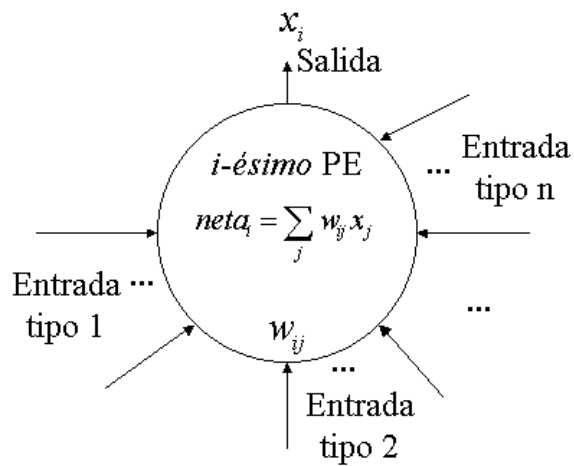


Figura 2-4. Estructura básica de un PE.

“La estructura en la Figura 2-4, representa un único PE de una red. Las conexiones de entrada se representan en forma de flechas procedentes de otros elementos de procesamiento. Cada conexión de entrada tiene asociado una cantidad de w_{ij} que se denomina peso, que se puede aplicar a otras unidades.”

Sopena, analogías con una neurona verdadera:

- El PE de la red tiene muchas entradas, pero una sola salida, la cual se puede aplicar a muchos otros PEs como una entrada para ellos.
- La entrada que recibe el i -ésimo PE tienen asociado una magnitud llamada PESO o INTENSIDAD DE CONEXIÓN.

CAPITULO 2: MARCO TEÓRICO

- El peso de la conexión que procede del j-ésimo nodo y que llega al i-ésimo nodo se denota mediante w_{ij} .
- La salida del PE se corresponde con la frecuencia de disparo de la neurona.
- Los pesos corresponden a la intensidad de las conexiones sinápticas entre neuronas. En nuestro modelo estas cantidades serán representadas mediante números reales.
- Las entradas de las neuronas se clasifican en diversos tipos, dado que cada uno puede tener una de varios efectos en el PE.
 - “Una conexión puede ser excitatoria o inhibidora. Otros términos como *ganancia*, *amortiguación*, y *disparo fortuito* aluden a otro tipo de conexiones de propósito especial.”

“En cuanto al funcionamiento de un elemento de procesamiento, cada uno determina un valor de entrada neto basándose en todas las conexiones de entrada. Típicamente, este valor es el resultado de una suma ponderada: $net a_i = \sum_j x_j w_{ij}$...donde el índice j recorre todas las conexiones que tenga el PE.”

“La excitación y la inhibición se tienen en cuenta automáticamente a través de su peso.”

“Una vez calculado el valor neto, se deberá calcular el valor de activación o simplemente activación: $a_i(t) = F_i(a_i(t-1), net a_i(t))$...lo que indica que la activación es una función explícita, dado un PE determinado, de la entrada neta actual y la activación anterior, denotada con t como una medida discreta de pasos.”

“El primer argumento de la función F_i está incluido sólo por generalidad.”

“En la mayoría de los casos la activación es idéntica al valor neto, y se utilizan de forma intercambiable. Sin embargo, en otras ocasiones no es así y se debe prestar atención a las diferencias.”

“Luego de haber calculado el valor de activación, se deberá determinar el valor de salida aplicando la función de salida.”

$$x_i = f_i(a_i)$$

“...dado que generalmente $a_i = neta_i$, entonces... $x_i = f_i(neta_i)$.”

“Finalmente, el PE dispara x_i hacia el resto de los PEs con los cuales se conecta, provocando un efecto sistemático con lo que se consigue, con los pesos adecuados, el resultado que se busca, previo entrenamiento de la red.”

2.1.1.4 Memoria Bidireccional Asociativa

Dentro de la discusión de las Redes Neuronales conviene realizar mención sobre las Memorias Asociativas. El concepto de memoria asociativa es bastante intuitivo: La memoria asociativa parece ser una de las funciones primordiales del cerebro. Nosotros asociamos fácilmente la cara de un amigo con su nombre, o un nombre con un número de teléfono. Algunos ejemplos de memorias asociativas existentes y conocidas son: Los bancos de memoria de los computadores, los cuales asocian datos con direcciones; y la programación orientada a objeto, la cual asocia a través de redes de herencia atributos, servicios, y otras características.

“La Memoria de Bidireccional Asociativa (BAM por sus siglas en inglés) consta de dos capas de elementos de proceso que están completamente interconectados entre capas. Las unidades pueden o no tener conexiones de realimentación consigo mismas. El caso general se ha ilustrado en la siguiente imagen:”

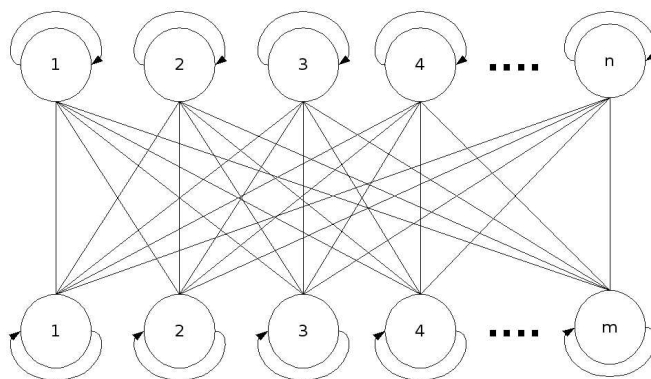


Figura 2-5. Estructura de una BAM.

CAPITULO 2: MARCO TEÓRICO

“La BAM, que se muestra en la Figura 2-5, posee n unidades en la capa x , y m unidades en la capa y . Por comodidad, denominaremos vector de entrada al vector x , y vector de salida al vector y . En esta red, ambos vectores pertenecen al espacio de Hamming (Freeman y Skapura, 1993 p. 134). Todas las conexiones entre unidades son bidireccionales, con pesos en ambos extremos. La información va y viene de una capa a otra, a través de estas conexiones. Las conexiones de realimentación de cada unidad pueden no estar presentes en todas las arquitecturas BAM.”

“Al igual que en otras redes neuronales, en la arquitectura de la BAM hay unos pesos asociados a las conexiones entre elementos de proceso.”

“A diferencia de otras redes neuronales, los pesos de una BAM pueden ser calculados de forma anticipada mediante la siguiente ecuación basada en el procedimiento de un Asociador Lineal (Freeman y Skapura, 1993 p. 136):”

$$\hat{w} = \hat{y}_1 \hat{x}_1^t + \hat{y}_2 \hat{x}_2^t + \dots + \hat{y}_L \hat{x}_L^t$$

“...donde L es el número de ejemplares que se desea la BAM reconozca posteriormente.”

“Esta ecuación da los pesos de las conexiones procedentes de la capa x y con destino a la capa y . Por ejemplo, el valor \hat{w}_{23} es el peso de la conexión procedente de la tercera unidad de la capa x y con destino a la segunda unidad de la capa y . Para construir los pesos de las unidades de la capa x , sólo hay que tomar la transpuesta de la matriz de pesos, \hat{w}^t .”

“Posterior a la construcción de la matriz de pesos con la fórmula anterior, la BAM se puede utilizar para recordar información. Este procedimiento se lleva a cabo mediante los siguientes pasos:”

1. “Se aplica un par de vectores iniciales (\hat{x}_0, \hat{y}_0) a los elementos de proceso de la BAM.”
2. “Se propaga la información de la capa x a la capa y , y se actualizan los valores de las unidades de la capa y .”
3. “Se vuelve a propagar la información actualizada desde y hacia la capa x , y se actualizan las unidades que se encuentran allí.”
4. “Se repiten los pasos 2 y 3 hasta que ya no hayan cambios en las unidades de ambas capas.”

CAPITULO 2: MARCO TEÓRICO

“Las unidades calculan sumas de productos de las entradas por los pesos, para determinar un valor de entrada neto. En la capa y :”

$$neto^y = \hat{w} \hat{x}$$

“...donde $neto^y$ es el vector de valores netos de la capa y .”

“En términos de las unidades individuales y_i , tenemos:”

$$neto_i^y = \sum_{j=1}^n w_{ij} x_j$$

“...y en la capa x :”

$$neto^x = \hat{w}^t \hat{x}$$

$$neto_i^x = \sum_{j=1}^m y_j w_{ji}$$

“El valor de salida para cada elemento de proceso depende del valor neto de la entrada, y del valor actual de salida de esa capa. El nuevo valor de y en el instante $t+1$, $\hat{y}(t+1)$, está relacionado con el valor de y en el instante t , $\hat{y}(t)$, mediante:”

$$\hat{y}(t+1) = \begin{cases} +1 & neto_i^y > 0 \\ y_i(t) & neto_i^y = 0 \\ -1 & neto_i^y < 0 \end{cases}$$

“De manera similar, $\hat{x}(t+1)$ está relacionado con $\hat{x}(t)$ mediante:”

$$\hat{x}(t+1) = \begin{cases} +1 & neto_i^x > 0 \\ x_i(t) & neto_i^x = 0 \\ -1 & neto_i^x < 0 \end{cases}$$

CAPITULO 2: MARCO TEÓRICO

“Uno de los problemas que se podrían ocasionar en este tipo de redes neuronales es el llamado Interferencia Cruzada, fenómeno causado por el exceso en el ingreso de ejemplares en una BAM (Freeman y Skapura, 1993).”

2.1.2 Comunicaciones Inalámbricas

2.1.2.1 Wi-Fi

Las comunicaciones Wireless son un sistema flexible de transmisión de datos implementados como una extensión, o como alternativa, de una red cableada. Utiliza tecnología de radio frecuencia (Modulación de Ondas Electromagnéticas), que transmite y recibe datos utilizando como medio el aire, minimizando la necesidad de una conexión de cable, permitiendo la combinación conectividad - movilidad.

Las implementaciones de redes inalámbricas están basadas en el estándar IEEE 802.11, el cual es para las redes inalámbricas, lo que el Ethernet es para las LAN. Es por esto que este estándar es una extensión inalámbrica de las LAN, ofreciendo todos los servicios de Ethernet cableado. Sin embargo, en particular esta tecnología se basa en el estándar 802.11g. Además esta tecnología ofrece estándares de seguridad nuevos, como el 802.11i, ofreciendo también los servicios de encriptación WEP, el soporte WPA y encriptación AES de alta seguridad, todo esto sobre una banda de transmisión de 2.4 [GHz].

Las redes inalámbricas actuales ofrecen velocidades de 54 [Mbps] y 108 [Mbps], en comparación a las redes cableadas que ofrecen velocidades desde 10 [Mbps] hasta 1 [Gbps] o más.

De igual modo, es importante considerar el lugar físico, puesto que el alcance de la señal y la velocidad de transmisión dependen también del área que se está cubriendo, se presentan problemas con ciertos materiales como el concreto, que no permite el paso de la señal, por lo que muros y columnas disminuyen el rendimiento de la red inalámbrica.

2.1.2.2 Bluetooth

El texto que se presenta a continuación es un extracto de (Wikipedia, 2006a).

Bluetooth es la norma que define un estándar global de comunicación inalámbrica que posibilita la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia.

Bluetooth proporciona una vía de interconexión inalámbrica entre diversos aparatos que tengan dentro de sí esta tecnología, como celulares (Nokia 6600), consolas (Nokia N-Gage), computadoras de mano (Palm, Pocket PC), cámaras, computadores portátiles, impresoras o simplemente cualquier dispositivo que un fabricante necesite interconectar, usando por supuesto una conexión segura de radio de muy corto alcance. El alcance que logran tener estos dispositivos es de 10 [m]. Para mejorar la comunicación es recomendable que nada físico (como una pared) se interponga.

El primer objetivo para los productos Bluetooth de primera generación eran los entornos de la gente de negocios que viaja frecuentemente. Esto originaba una serie de cuestiones previas que deberían solucionarse tales como:

- El sistema debería operar en todo el mundo.
- El emisor de radio deberá consumir poca energía, ya que debe integrarse en equipos alimentados por baterías.
- La conexión deberá soportar voz y datos, y por lo tanto aplicaciones multimedia.

2.1.2.2.1 Tecnología

La especificación de Bluetooth define un canal de comunicación de máximo 720 [Kbps] con rango óptimo de 10 [m] (opcionalmente 100 [m]).

La frecuencia de radio con la que trabaja está en el rango de 2.4 a 2.48 [Ghz] con amplio espectro y saltos de frecuencia con posibilidad de transmitir en Full Duplex con un máximo de 1600 [saltos/seg]. Los saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1 [Mhz]; esto permite dar seguridad y robustez.

CAPITULO 2: MARCO TEÓRICO

La potencia de salida para transmitir a una distancia máxima de 10 [m] es de 0 [dBm] (1 [mW]), mientras que la versión de largo alcance transmite entre 20 y 30 [dBm] (entre 100 [mW] y 1 [W]).

El protocolo de banda base (canales simples por línea) combina conmutación de circuitos y paquetes. Para asegurar que los paquetes no lleguen fuera de orden, los slots pueden ser reservados por paquetes síncronos, un salto diferente de señal es usado para cada paquete. Por otro lado, la conmutación de circuitos puede ser asíncrona o síncrona. Tres canales de datos síncronos (voz), o un canal de datos síncrono y uno asíncrono, pueden ser soportados en un solo canal. Cada canal de voz puede soportar una tasa de transferencia de 64 [Kbps] en cada sentido, la cual es suficientemente adecuada para la transmisión de voz. Un canal asíncrono puede transmitir como mucho 721 [Kbps] en una dirección y 56 [Kbps] en la dirección opuesta. Sin embargo, para una conexión asíncrona es posible soportar 432,6 [Kbps] en ambas direcciones si el enlace es simétrico.

2.1.2.3 IR

Los texto presentados a continuación son un extracto de (Wikipedia, 2006b).

Los enlaces infrarrojos se encuentran limitados por el espacio y los obstáculos. El hecho de que la longitud de onda de los rayos infrarrojos sea tan pequeña (850-900 [nm]), hace que no pueda propagarse de la misma forma en que lo hacen las señales de radio.

Es por este motivo que las redes infrarrojas suelen estar dirigidas a oficinas o plantas de oficinas de reducido tamaño.

Por otro lado, las transmisiones infrarrojas presentan la ventaja, frente a las de radio, de no transmitir a frecuencias bajas, donde el espectro está más limitado, no teniendo que restringir, por tanto, su ancho de banda a las frecuencias libres.

2.1.2.3.1 Modos de transmisión

Al momento de transmitir, las estaciones infrarrojas pueden usar tres tipos de métodos para ello:

- Punto a punto
- Cuasi-difuso
- Difuso.

En el modo *punto a punto*, el tipo de emisión por parte del transmisor se hace de forma direccional. Por ello, las estaciones deben verse directamente, para poder dirigir el haz de luz directamente de una hacia la otra. Por este motivo, este es el tipo de red inalámbrica más limitado, pues a todos los inconvenientes de las comunicaciones infrarrojas hay que unir el hecho de tener que colocar las estaciones enfrentadas. Este método se suele usar en redes inalámbricas Token Ring, donde el anillo está formado por una unión de enlaces punto a punto entre las distintas estaciones, conformando cada uno de los segmentos.

En el modo *cuasi-difuso*, el tipo de emisión es radial; esto es, la emisión se produce en todas direcciones, al contrario que en el modo punto a punto. Para conseguir esto, lo que se hace es transmitir hacia distintas superficies reflectantes, las cuales redirigirán el haz de luz hacia la/s estación/es receptora/s. De esta forma, se rompe la limitación impuesta en el modo punto a punto de la direccionalidad del enlace. En función de cómo sea esta superficie reflectante, podemos distinguir dos tipos de reflexión: pasiva y activa. En la reflexión pasiva, la superficie reflectante simplemente refleja la señal, debido a las cualidades reflexivas del material. En la reflexión activa, por el contrario, el medio reflectante no sólo refleja la señal, sino que además la amplifica. En este caso, el medio reflectante se conoce como satélite. Destacar que, mientras la reflexión pasiva es más flexible y barata, requiere de una mayor potencia de emisión por parte de las estaciones, debido al hecho de no contar con etapa repetidora.

El modo de emisión *difuso*, por otro lado, se diferencia del cuasi-difuso en que debe ser capaz de abarcar, mediante múltiples reflexiones, todo el recinto en el cual se encuentran las estaciones. Obviamente, esto requiere una potencia de emisión mayor que los dos modos

CAPITULO 2: MARCO TEÓRICO

anteriores, puesto que el número de rebotes incide directamente en el camino recorrido por la señal y las pérdidas aumentan.

Según el caso comentado de las empresas que utilizaban enlaces de un edificio a otro mediante antenas en las ventanas, se observa que, obviamente, este enlace será punto a punto, mientras que en las redes interiores es más lógico realizar enlaces difusos.

2.1.3 Sistemas de Posicionamiento Global

2.1.3.1 GPS

El Global Positioning System (GPS) o Sistema de Posicionamiento Global originalmente llamado NAVSTAR, es un **Sistema Global de Navegación por Satélite** (GNSS) el cual permite determinar en todo el mundo la posición de un aparato receptor con una desviación de cuatro metros. El sistema fue desarrollado e instalado, y actualmente es operado, por el Departamento de Defensa de los Estados Unidos.

El GPS funciona mediante una red de satélites (véase Figura 2-6) que se encuentran orbitando alrededor de la tierra. Cuando se desea determinar la posición, el aparato que se utiliza para ello localiza automáticamente como mínimo cuatro satélites de la red, y como máximo, actualmente, doce satélites (aparatos civiles), de los que recibe unas señales indicando la posición y el reloj (tiempo cronológico) de cada uno de ellos. En base a estas señales, el aparato sincroniza el reloj del GPS y calcula el retraso de las señales, es decir, la distancia al satélite. Por "triangulación" calcula la posición en que éste se encuentra. La triangulación consiste en averiguar el ángulo de cada una de las tres señales respecto al punto de medición. Conocidos los tres ángulos se determina fácilmente la propia posición relativa respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o coordenadas reales del punto de medición. También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que desde tierra sincronizan a los satélites.

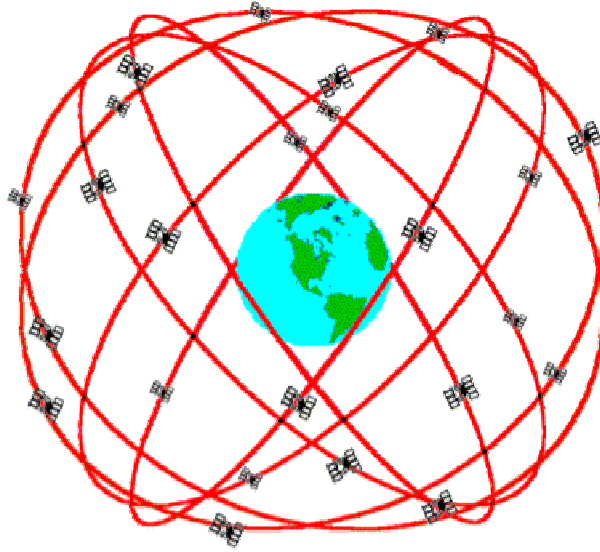


Figura 2-6. Ejemplo de satélites GPS.

2.1.3.2 DGPS

El DGPS (Differential GPS) o GPS diferencial es un sistema que proporciona a los receptores de GPS correcciones a los datos recibidos de los satélites GPS (véase Figura 2-7). Estas correcciones, una vez aplicadas, proporcionan una mayor precisión en la posición calculada.

El sistema de correcciones funciona de la siguiente manera:

1. Una estación base en tierra, con coordenadas muy bien definidas, escucha los satélites GPS.
2. Calcula su posición por los datos recibidos de los satélites.
3. Dado que su posición está bien definida, calcula el error entre su posición verdadera y la calculada, estimando el error en cada satélite.

Se envía estas correcciones al receptor a través de algún medio.

Existen varias formas de obtener las correcciones DGPS. Las más usadas son:

1. Recibidas por radio a través de algún canal preparado para ello, como el RDS en una emisora de FM.
2. Descargadas de Internet con una conexión inalámbrica.
3. Proporcionadas por algún sistema de satélites diseñado para tal efecto. En Estados Unidos existe el WAAS, en Europa el EGNOS y en Japón el MSAS, todos compatibles entre sí.

Para que las correcciones DGPS sean válidas, el receptor tiene que estar relativamente cerca de alguna estación DGPS, generalmente, a menos de mil kilómetros.

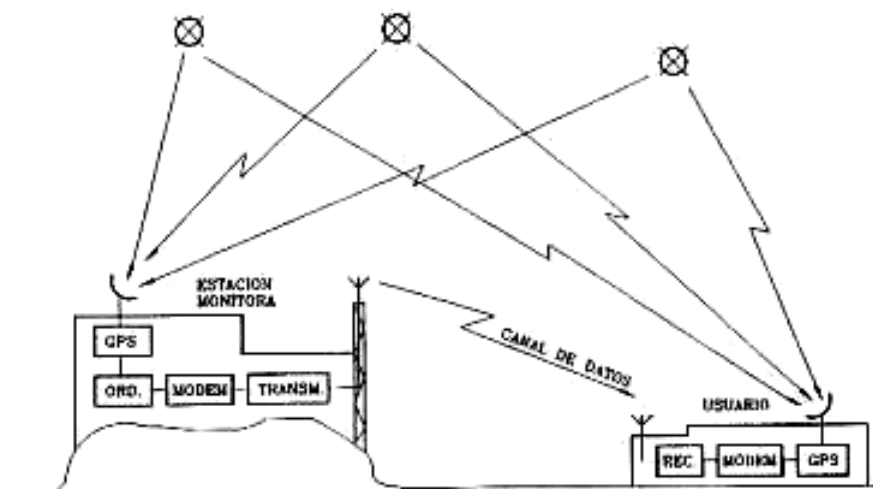


Figura 2-7. Ejemplo DGPS.

2.1.3.3 Geodésico

El presente texto es un extracto de (Instituto de Información Territorial del Estado de Jalisco, 2006).

GPS que utiliza modelos matemáticos diseñados para que la medición obtenida se ajuste lo mejor posible a parte del geoide o todo él (véase Figura 2-8). Se define por un elipsoide y se relaciona con un punto de la superficie topográfica para establecer el origen del datum. Esta relación se puede definir por seis cantidades: la latitud y longitud geodésica y la altura del origen, los dos componentes de la deflexión de la vertical en el origen y el azimut geodésico de una línea de dicho origen a cualquier punto.

Este aparato puede conseguir un error como mínimo de menos de un centímetro y como máximo de dos metros.

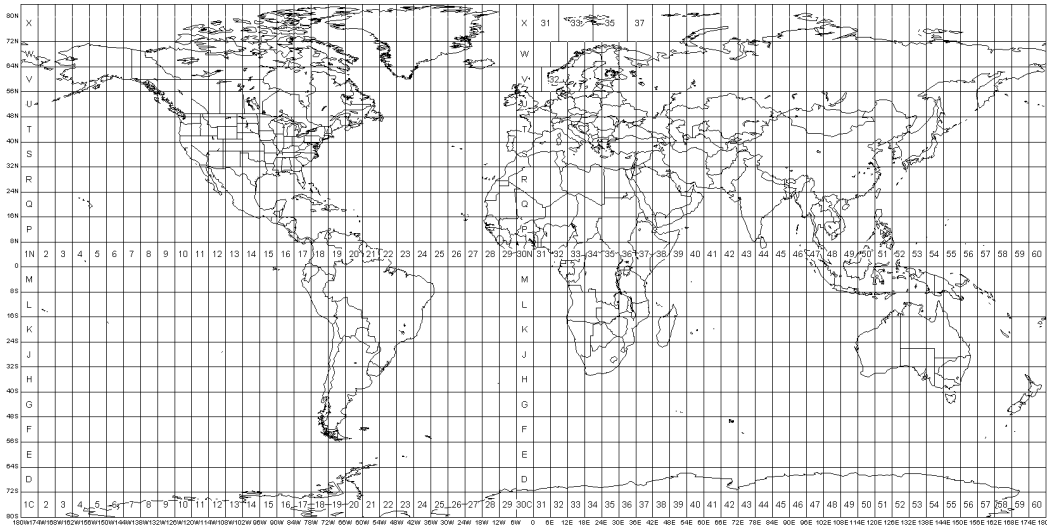


Figura 2-8. Esquema datum del geode.

2.1.3.4 Estaciones Radiales

El trabajo con estaciones radiales consiste en la triangulación de la posición averiguando el ángulo de cada una de las señales radiales (como mínimo dos) respecto al punto de medición (véase Figura 2-9). Conocidos los ángulos se determina fácilmente la propia posición relativa respecto a las antenas emisoras. Conociendo además las coordenadas o posición de cada una de ellas por la señal que emiten, se obtiene la posición absoluta o coordenadas reales del punto de medición.

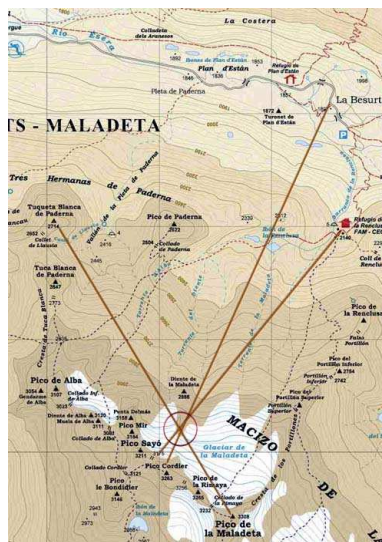


Figura 2-9. Triangulación a través cuatro estaciones radiales.

2.1.4 Microcontroladores

Los microcontroladores son circuitos integrados que incorporan todos los bloques funcionales de un sistema microprocesador en un único encapsulado (véase Figura 2-10), el cual para funcionar necesitan una tensión continua estable entre 5 [V] y 1.5 [V], además de un oscilador.

Los microcontroladores son capaces de interpretar o decodificar combinaciones de bits (instrucciones) y generan señales digitales internas y/o externas, para dar cumplimiento de manera continua a una secuencia de estas instrucciones (programa) que permita controlar un sistema o subsistema electrónico.

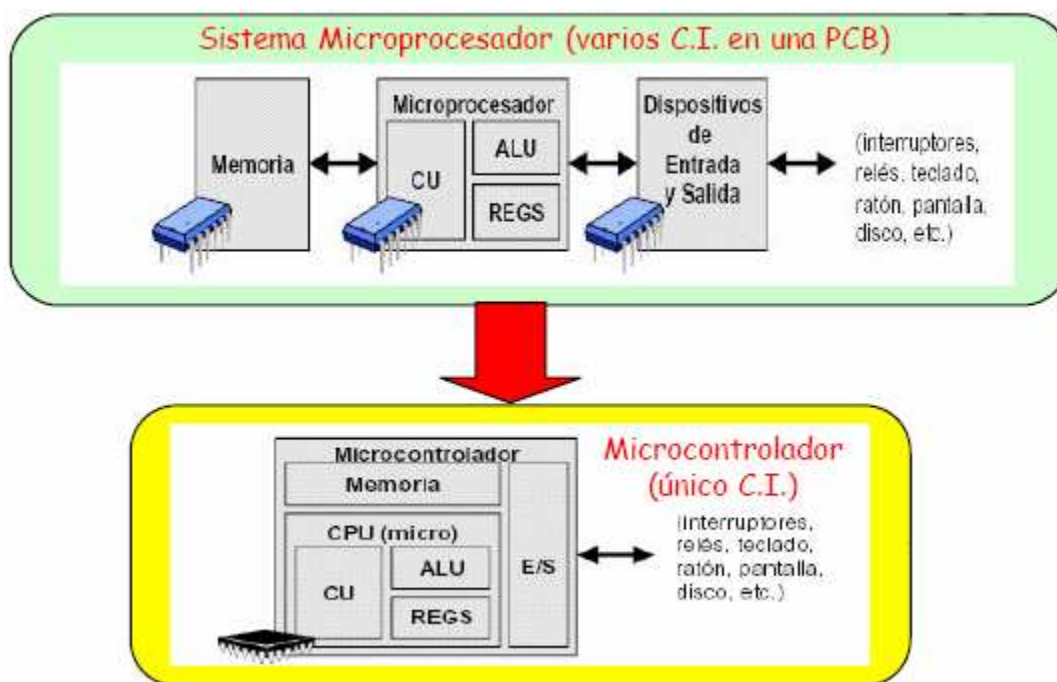


Figura 2-10. Estructura de un microcontrolador.

2.1.5 Sensores Ultrasónicos.

El ultrasonido es una vibración mecánica con un rango mayor al audible por el oído humano (el sonido audible está en el rango de los 16 [Khz] a 20 [Khz], en cambio el ultrasonido está desde los 20 [Khz] en adelante) que se transmite a través de un medio físico, el cual es registrado y medido en Hertz. Es por estas características que los sensores ultrasónicos son utilizados con distintas áreas con distintos fines, por ejemplo: medición de distancias,

CAPITULO 2: MARCO TEÓRICO

caracterización interna de materiales, ensayos no destructivos, sonografías y otros. Además se aprovecha que las vibraciones pueden desprender incrustaciones de suciedad en la limpieza de diversos objetos, especialmente en joyería. Para poder realizar todas estas aplicaciones, la tecnología del ultrasonido se basa en el choque de la onda sonora o en su posterior eco, el rebote de esta onda sonora posee mucha información útil, gracias a la impedancia acústica (Z), que es la resistencia de los materiales al paso del sonido, pudiendo identificar materiales gracias a este producto ($Z = D * V$)², otra información que podemos encontrar en el rebote de la onda sonora, es la distancia a la que se encuentra el objeto en el cual se produjo el eco, esto se puede calcular gracias a una de las fórmulas más básicas de la física ($Z = \frac{T}{V}$)³.

Los anales de esta tecnología se encuentra en 1881, cuando Jacques y Pierre Curie publicaron los resultados obtenidos al experimentar la aplicación de un campo eléctrico alternante sobre cristales de cuarzo y turmalina (lo que más tarde se conocería como generador piezoeléctrico), los cuales produjeron ondas sonoras de muy altas frecuencias.

Posteriormente esta tecnología quedó olvidada hasta 1912 cuando la tragedia del Titanic, hizo proponer a L. F. Richardson, la utilización de ecos ultrasónicos para detectar objetos sumergidos.

Entre 1914 y 1918 durante la Primera Guerra Mundial, se trabajó intensamente en esta propuesta, intentando detectar submarinos enemigos.

En 1929, Sergei Sokolov, científico ruso, propuso el uso del ultrasonido para detectar grietas en metal, y también para microscopía.

Entre 1939 y 1945, durante la Segunda Guerra Mundial, el sistema inicial desarrollado por Langevin⁴, se convirtió en el equipo de norma para detectar submarinos, conocido como ASDIC (Allied Detection Investigation Committes). Además se colocaron sondas ultrasónicas en los torpedos, las cuales los guiaban hacia sus blancos. Más adelante, el sistema se convertiría en

² Z (Impedancia Acústica) = D (Densidad del medio) * V (Velocidad del sonido en dicho medio).

³ D (Distancia) = T (Tiempo transcurrido entre el envío del pulso y la recepción del eco) / V (Velocidad del sonido en el medio).

⁴ Primer generador piezoeléctrico de ultrasonido, cuyo cristal servía también como receptor, y generaba cambios eléctricos al recibir vibraciones mecánicas.

CAPITULO 2: MARCO TEÓRICO

el SONAR (Sound Navigation and Ranging), cuya técnica muy mejorada es norma en la navegación.

Desde 1942 con Karl Dussik (Médico Psiquiatra, intentó detectar tumores cerebrales), hasta 1994 (Dr. Gonzalo E. Díaz, introdujo el postproceso en Color para imágenes sonográficas) los avances relacionados con la tecnología del ultrasonido se han resumido en el área de la salud. Pero en 2002 se empezó a hablar de la sonofusión. La sonofusión es capaz de generar reacciones nucleares a partir de la implosión de diminutas burbujas, lo que supone disponer de la habilidad de usar una simple fuerza mecánica, por primera vez en la historia, para iniciar condiciones comparables con las que hay en el interior de las estrellas.

2.1.6 Compás Magnético

La brújula, también llamada compás magnético, es un instrumento que consta de un imán suspendido por su centro de gravedad, que gira horizontalmente con la mayor libertad posible, de modo que adopta una determinada posición en función del campo magnético al que esté sometido. Así el imán, al seguir las líneas de fuerza del campo magnético de la tierra, proporciona una indicación permanente en grados (véase Figura 2-11) del rumbo respecto al Norte magnético terrestre, correspondiente a los 0° . Este instrumento es la referencia básica para mantener la dirección de cualquier vehículo, pero inútil en las zonas polares, norte y sur, debido a la convergencia de las líneas de fuerza del campo magnético terrestre.



Figura 2-11. Brújula o Compás magnético.

2.1.6.1 Magnetismo Terrestre

El fenómeno del magnetismo terrestre se debe a que toda la Tierra se comporta como un gigantesco imán (véase Figura 2-12). Aunque no fue hasta 1600 DC que se señaló esta similitud. Los efectos del magnetismo terrestre se habían utilizado mucho antes en las brújulas primitivas. El nombre dado a los polos de un imán (Norte y Sur) se debe a esta similitud.

Un hecho a destacar es que los polos magnéticos de la Tierra no coinciden con los polos geográficos de su eje. Las posiciones de los polos magnéticos no son constantes y muestran ligeros cambios de un año para otro, e incluso existe una pequeñísima variación diurna solo detectable con instrumentos especiales.

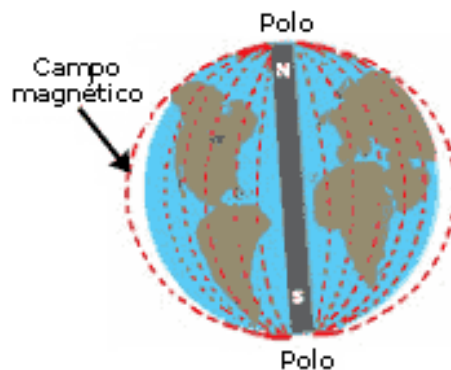


Figura 2-12. Campo magnético terrestre.

2.2 Evolución de la Robótica

Algunos de los textos presentados bajo este título son extracto literal de (Prieto-Moreno, Andrés, 2006).

“Desde que en 1917 el escritor Karel Capek usara el término robot en su obra “Rossum’s Universal Robots” como “robota” (que en checo significa trabajador forzado) para referirse a unas máquinas en forma de humanoide.”

“En el libro *Robótica Industrial* (P. GROOVER, MIKE *et al*, 1990) se define el término “robótica” como una ciencia aplicada que surge de la combinación de la tecnología de las

CAPITULO 2: MARCO TEÓRICO

máquinas-herramienta y de la informática. Una máquina-herramienta se define como una máquina que efectúa cualquier trabajo manual, y la informática como la ciencia del tratamiento automático y racional de la información. Uniendo ambos conceptos, la robótica surge al automatizar de manera racional las máquinas-herramienta, es decir al permitir que un programa informático controle las operaciones que antes realizaba un operario.”

“Ligado a la robótica aparece el robot. Si lo primero es la ciencia lo segundo es el objeto. Se considera la aparición del primer robot, en su concepción moderna, como la unión del control numérico y de la telequérica. El control numérico fue una de las primeras formas de la informática y la telequérica es la ciencia que estudia los manipuladores controlados a distancia por un ser humano o teleoperadores, que se pueden considerar como una máquina-herramienta avanzada. A partir de aquí la evolución de la robótica ha estado íntimamente relacionada con el desarrollo de la tecnología eléctrica e informática, todo con la aparición de nuevos y mejores sistemas de control.”

“En 1959 se introdujo el primer robot comercial por Planet Corporation, estaba controlado por interruptores de fin de carrera y levas.”

“En 1971 la Universidad de Stanford desarrolló el *Stanford Arm*, un pequeño brazo robot de accionamiento eléctrico.”

“En 1981 la Universidad Carnegie-Mellon diseñó un robot que utilizaba motores eléctricos situados en las articulaciones del manipulador sin las transmisiones mecánicas habituales.”

“Las primeras aplicaciones prácticas de los robots se encuentran en la industria y sobre todo en las industrias del motor. En 1961 la Ford Motor Company utilizó uno para controlar una máquina de fundición en troquel y en 1974 Kawasaki instaló otro para soldar las estructuras de las motocicletas.”

“En paralelo, los lenguajes de programación de robots fueron mejorando, y en la exposición “Robots 8” (1984) se mostraron los primeros lenguajes de programación, que permitían el desarrollo de programas de control utilizando gráficos interactivos en un computador personal que luego se podían cargar en el robot.”

CAPITULO 2: MARCO TEÓRICO

“Al ser la industria la que adquirió la mayoría de ellos, una de las primeras definiciones del término robot la proporcionó el Robotics Industries Association (RIA) y vino a decir:”

“Un robot industrial es un manipulador multifuncional reprogramable diseñado para desplazar materiales, piezas, herramientas o dispositivos especiales mediante movimientos programados variables para la ejecución de una diversidad de tareas.”

“Bajo el punto de vista anterior es fácil comprender la definición dada por la RIA, aunque actualmente abarca una muy reducida parte de lo que hoy es un robot, ya que existen muchas más aplicaciones que no responden a las características anteriores. Además, se está produciendo un cambio en la actitud de la sociedad respecto a la robótica, todo por la sucesiva aparición de aplicaciones que, de no haber sido por ella, no podrían haberse llevado a cabo. Al principio existió un sentimiento generalizado de oposición al robot, se consideraban más enemigos que amigos, siendo la razón principal el temor a ser sustituidos. Evidentemente esta opinión tiene su origen en un momento puntual de la historia y, más específicamente, en todas aquellas personas que perdieron sus puestos de trabajo al introducir los robots en las fábricas. Pero ahora se puede decir que este hecho permitió la evolución de los trabajos, ya que se eliminaron las tareas repetitivas que no permitían un enriquecimiento mental, y se crearon nuevos puestos dentro de las empresas, que permitían una participación mayor del empleado, siendo su trabajo más humano que repetitivo.”

2.2.1 La Robótica en el Cine

Otro concepto era la forma que tenía que tener un robot. Al principio la mayoría opinaba que un robot tendría un aspecto humanoide, con pies, manos, cabeza, más o menos inteligente y que estaba al servicio de su amo. Si hubiese que buscar un motivo de esto se podría decir que fueron las películas, libros y series de TV los medios de difusión de esta idea. Se podrían citar las numerosas obras del escritor Isaac Asimov que tenían como protagonistas a robots humanoides. Fue este escritor el que enunció las tres leyes de la robótica (“Runaround”, publicada por primera vez en la revista “Astounding Science Fiction” en 1942, línea 7, página 100. Editor John Campbell.), en ellas se puede ver la preocupación inicial del hombre en mantener el control de estos seres mecánicos, debido al temor existente a una revolución de esta raza. El temor pudo ser gatillado por ser un nuevo y radical concepto que afectaba a la sociedad. Este argumento se ha usado en infinidad de películas y cuentos, teniendo un ejemplo clarísimo en el robot T-800 de

Terminator (“The Terminador”, película dirigida por James Cameron en 1984, de la productora cinematográfica Orion Pictures International).

2.2.2 Leyes de la Robótica de Isaac Asimov

“Las tres leyes de la robótica, enumeradas más abajo, sirvieron en un principio para discutir el futuro amenazador de los robots, pero dado que todavía no se ha alcanzado dicho momento se han utilizado más para escribir novelas que para construir robots, al menos hasta que no se alcance un desarrollo de inteligencia mayor al existente. Las Leyes de la Robótica de Isaac Asimov son :”

1. “Un robot no puede herir a un ser humano ni dejar de intervenir si éste está en peligro de sufrir algún daño.”
2. “Un robot debe obedecer las órdenes de los seres humanos, salvo cuando éstas contradigan la primera ley.”
3. “Un robot debe proteger su propia existencia, siempre y cuando dicha existencia no contradiga las leyes primera y segunda.”

“Actualmente, la robótica sigue avanzando con paso firme, pero con un ritmo menor que el previsto por los expertos en los años 80. Adicionalmente, la sociedad ya ha asumido el papel de estos, considerándolos como herramientas, sin una forma típica y con una inteligencia mínima en comparación con la del ser humano. La evolución de la tecnología ha permitido desarrollar nuevos materiales, sensores y chips que han facilitado y abaratado considerablemente la construcción de robots. Sin embargo, todavía no se ha producido la explosión que ocurrió con los computadores personales, que pasaron de ser un privilegio de empresas multinacionales y universidades, a ser un electrodoméstico más dentro de una casa. Es más, es normal que al adquirir un computador se le permita al comprador seleccionar los diferentes componentes que lo van a integrar. Y todo se debe al acercamiento que sufrió la informática al público no especializado, al momento en que su uso se volvió más sencillo, económico y con aplicaciones más potentes y flexibles que las tradicionales. En este punto, se puede destacar el paulatino desuso de las máquinas de escribir en favor de las aplicaciones de procesamiento de textos.”

“La reflexión anterior induce a pensar en una posible revolución de la robótica, pasando de las industrias hasta llegar a ser un elemento más dentro del hogar y de la sociedad en general.

CAPITULO 2: MARCO TEÓRICO

Ésta opinión es una apuesta de futuro, avalada por la aparición, a lo largo de estos últimos años, de robots y proyectos muy llamativos que han facilitado los cambios de opinión de las personas hacia los robots. Ya no se ven como los grandes enemigos, sino todo lo contrario, son tratados como herramientas amigas, que tratan de ayudar y de facilitar el trabajo.”

2.2.3 Robótica al Servicio de la Ciencia de la Exploración

“En 1997 un pequeño robot móvil aterrizó en Marte, permitiendo que millones de espectadores pudiesen apreciar fotografías captadas desde 191 millones de kilómetros de la Tierra. Este pequeño robot llamado Sojourner, más conocido como *Pathfinder* (nombre de la misión). Este robot poseía seis ruedas y 10,5 [Kg]. Comenzó a recorrer Marte el 5 de julio de 1997, un día después que su nave nodriza, Mars Pathfinder, aterrizara en una llanura rocosa del hemisferio norte del planeta. Su tamaño era de 62 [cm] de longitud y 32 [cm] de altura, fue diseñado para obtener datos de la superficie marciana que luego serían interpretados por científicos en la Tierra. Se le ha calificado de geólogo interplanetario con visión estereoscópica, espectrómetro de partículas alfa, protones y rayos X, con inteligencia suficiente para llevar a cabo las órdenes dadas desde la Tierra y de esperar nuevas órdenes cuando algún inconveniente imposibilitaba o ponía en peligro la ejecución de la misión. (Información obtenida del reportaje de William R. Newcott en la revista National Geographic)”

“Este robot recordó los primeros pasos de la llegada del hombre a la Luna y aumentó las esperanzas de que algún día, más próximo, el hombre llegará a Marte. Otro caso parecido fue el robot sumergible que bajó a 3500 metros de profundidad para introducirse en los restos del *Titanic* con objeto de fotografiarlo. En esta ocasión el robot se llamaba *Jason Junior* y se operaba por control remoto.”

“Los ejemplos anteriores muestran aplicaciones fuera de la industria como tal. No son las únicas. Otros campos, como la medicina, aeronáutica y agricultura se beneficiaron con la llegada de robots telecontrolados para realizar operaciones, aviones espías autónomos y los sistemas de recolección y envasado directo. Aún así, estos robots son excesivamente caros y su función está muy especializada, para lograr su integración total se necesitará hacerlos cada vez menos costosos y generales, de manera que sean económicamente accesibles a las personas ajenas a toda esta evolución.”

2.2.4 Auge de la Robótica

Las dos características anteriores, económico y general, se encuentran en un tipo de robot que poco a poco se está abriendo camino. Los lugares dónde se puede ver este tipo de desarrollos es en entornos universitarios y, más recientemente, en algunas compañías jugueteras. La razón fundamental, es que la informática, electrónica y mecánica han pasado de ser un privilegio de algunos a ser materia de estudio en muchos centros. La tecnología ha evolucionado tanto que los precios se han reducido mucho. Además construir un pequeño robot se ha convertido en un hobby para muchas personas, llenos de imaginación y en muchas ocasiones motivados por los concursos que se realizan por el mundo. Uno de los más famosos es ROBOCUP (ROBOCUP , 2006), que consiste en un partido de fútbol entre robots (véase Figura 2-13).

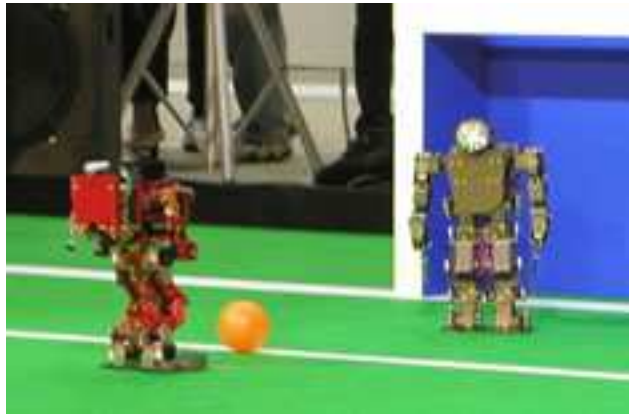


Figura 2-13. Juego de Football Socker de ROBOCUP modalidad humanoides.

“La aparición de concursos ha hecho que la aplicación práctica de investigaciones informáticas y electrónicas se lleven a cabo mediante pequeños robots que realizan actividades sorprendentes. Uno de esos robots es COG (véase Figura 2-14), desarrollado en el MIT, que es capaz de jugar tenis de mesa.”

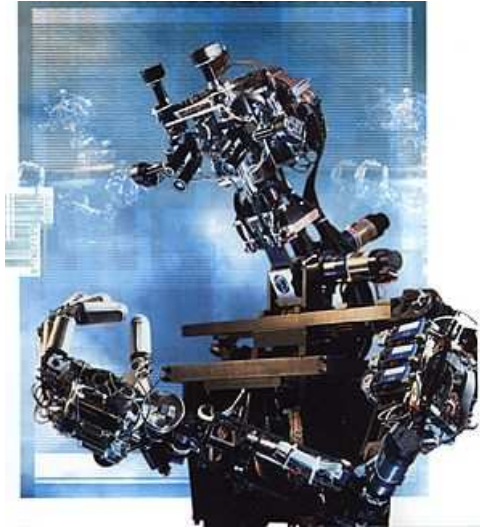


Figura 2-14. Robot COG, del MIT.

“En él se ensayan estudios de visión artificial, mecánica, elasticidad en robótica y también actitudes de comportamiento, esto último más ligado a la inteligencia artificial que a la robótica, pero que sirve para darle un carácter más humano.”

“Por último, han sido varias las compañías jugueteras que han sacado kits de aprendizaje o de iniciación a la robótica, la más avanzada ha sido LEGO con su kit LEGO MindStorms, pero también hay que destacar a Parallax, Robotix, Mattel y Bizak, éste último más enfocado a un producto de entretenimiento que a uno didáctico.”

2.3 Aplicaciones de las Tecnologías

Debido al objetivo del presente proyecto, mencionado en la sección 1.2, se hizo imprescindible contar con tecnologías que permitieran la conformación de un bosquejo acerca del entorno que rodearía al autómatas. Una solución relativamente confiable, sencilla y económica al problema de bosquejar el entorno que lo rodea, es mediante la utilización de la tecnología de ultrasonido, el cual utiliza un eco sónico para medir distancias; otra solución para la medición de distancia a objetos es la tecnología IR (infrarrojo), la cual utiliza un haz de luz infrarroja que al rebotar contra el objeto sufre una alteración, y dependiendo de tal alteración se puede calcular la distancia. Sin embargo, en contraste con la tecnología IR, el ultrasonido resulta ser de mayor precisión y, por lo tanto, más confiable al momento de realizar un bosquejo del entorno, esto debido a que, la alteración del haz de luz no es provocada solamente por la distancia hacia el

CAPITULO 2: MARCO TEÓRICO

objeto, sino que también la determina la tonalidad de color de la superficie del objeto; por otra parte, la tecnología láser es mucho más precisa que la ultrasónica. Sin embargo, el costo de esta es demasiado elevado, y por lo tanto inaccesible para este proyecto. Finalmente, existe la tecnología sensible al tacto, que resulta ser un método invasivo al entorno y lento en el rescate de datos, por lo tanto resulta poco práctico en comparación con las otras tecnologías mencionadas.

La necesidad de poder monitorizar la percepción del entorno que obtiene el autómata, al igual que las decisiones tomadas sobre aquel entorno, hizo necesaria la existencia de un medio por el cual se pudiera llevar a cabo esta tarea. Dicha tarea, por la naturaleza del proyecto, se debería realizar independiente de la estructura física del robot, es decir, a través de un medio físico que no coarte la movilidad de robot. Una solución al problema es la tecnología de redes de comunicación cableada, la cual, si bien es cierto, reduce la dependencia estructural del monitoreo, limita el ámbito de movimiento del mismo, por lo cual resulta poco práctica e incómoda. Por otra parte, está la tecnología de redes de comunicación inalámbrica, la cual es más onerosa que las cableadas, pero tiene la gran ventaja de la independencia estructural total del emisor/receptor con el cual se pretende intercambiar datos. En este sentido, esta tecnología permitiría realizar monitoreos a distancia de las percepciones del autómata, de manera práctica y cómoda.

La ruta seguida por el Autómata para conseguir llegar a su destino, según la definición mencionada en la sección 1.2, estará dada por puntos geográficos intermedios los cuales tendrá que lograr alcanzar antes de llegar a su destino. Para tener el concepto de ubicación espacial determinada por aquellos puntos geográficos intermedios, existen diversas tecnologías utilizadas en ambientes generales y controlados. En ambientes generales, las más conocidas son las descritas en la sección 2.1.3. Y para los segundos, las más conocidas son las técnica de guía por hitos, triangularización de la posición a través de diversos medios y visión artificial. En este proyecto, dado que está enfocado a ambientes estocásticos, para lograr la noción de la ubicación espacial no se podrá utilizar las técnicas de ambiente controlados.

Para realizar el control directo de componentes, como la tecnología de ultrasonido y compás, se utilizará el microcontrolador. De esta forma se podrá realizar operaciones de alto nivel a través de este microcontrolador sobre los dispositivos conectados a él.

CAPITULO 2: MARCO TEÓRICO

Para el manejo de alguno de los efectores del cuerpo del autómatas se considerarán otros microcontroladores adicionales (para mayor información sobre microcontroladores, véase sección 2.1.4), los cuales permitirán ordenar a voluntad servomotores, con la ventaja de realizar movimientos en paralelo y con un alto nivel de abstracción.

La problemática planteada al momento de que un autómatas se desempeñe en un entorno no controlado es la enorme cantidad de variables y situaciones que se deben considerar que se definen en aquel ambiente estocástico. Una forma de resumir toda la complejidad que esto involucra, es a través de la técnica de inteligencia artificial de redes neuronales. Con esta técnica el autómatas será capaz de reconocer, en el ambiente, situaciones a través de patrones de semejanza con otras situaciones conocidas dado su previo entrenamiento, y de las cuales pueda extrapolar acciones a seguir. El entrenamiento previo entregado a las redes neuronales del autómatas podrá ser perfeccionado a través del tiempo y la inteligencia de las acciones que realice éste dependerá de dos factores: El concepto de inteligencia que tenga el observador que lo juzgue y la cuantía del entrenamiento al cual sea sometido el autómatas. Finalmente, la técnica de las redes neuronales no busca la perfección en el reconocimiento de situaciones, sino la eficacia en aquello, con lo que las acciones realizadas por el autómatas quedan expuestas a posibles incoherencias debido a alguna falencia o situación no considerada en su previo entrenamiento. Sin embargo, a través de esta técnica se asegura que siempre se encuentre una respuesta, aunque esta no sea la mejor, que es lo que ya se había mencionado sobre los conceptos de la eficacia versus lo perfecto.

CAPITULO 3: DESARROLLO DEL PROYECTO

CAPITULO 3: DESARROLLO DEL PROYECTO

3.1 Lenguaje de Modelamiento a Utilizar

UML son las siglas de Unified Modeling Language (Lenguaje Unificado De Construcción de Modelos), notación, esquemática en su mayor parte, con que se construyen sistemas por medio de conceptos orientados a objetos.

En las etapas de desarrollo de alguna solución de software, muchas veces, quienes están involucrados se realizan las siguientes preguntas: ¿Cómo deberían asignarse las responsabilidades a las clases de objetos?, ¿Cómo deberían interactuar estos?, ¿Qué papel debe destinársele a cada clase? Estas son preguntas muy importantes cuando se diseña un sistema.

Este lenguaje de modelamiento es un estándar de comunicación trasversal a todo el proceso de desarrollo de software, es decir, abarca todas las etapas comúnmente utilizadas para el desarrollo del software. Es por esto, que para este proyecto se hará uso de esta herramienta, dado que resulta ser la más adecuada para realizar un modelado de un sistema que considera diversos conceptos lógicos, además de ser la herramienta con la que más familiarizados están los desarrolladores.

3.2 Fase de Planificación y Elaboración

3.2.1 Conocimiento de Requerimientos del Sistema

3.2.1.1 Presentación General

Este proyecto tiene por objetivo el desarrollo de un agente inteligente capaz de desplazarse en un ambiente estocástico guiado por sensores ultrasónicos, un compás electrónico y un DGPS, y monitoreado mediante tecnología wireless desde un computador remoto.

La documentación presentada aquí será bajo la notación utilizada en (Larman Craig, 1999).

3.2.1.2 Clientes

- CIDCIE.

Este proyecto cuenta con el auspicio del CIDCIE, a través del proyecto FIDIE N° 132005.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.2.1.3 Metas

En general lo que se busca con este proyecto es:

- Investigar las tecnologías necesarias, más adecuadas para comunicaciones, inteligencia artificial y SIGs.
- Construir un manual de referencias que contenga el procedimiento de uso de las tecnologías utilizadas.
- Desarrollar un software con la inteligencia artificial necesaria para guiar un robot móvil a través de un ambiente estocástico.
- Construir un software de monitoreo que permita, a través de comunicación inalámbrica, saber el estado del robot en tiempo real (ubicación geográfica, información de sensores, decisiones tomadas).
- Utilizar y modificar un robot móvil construido por el grupo de robótica.

3.2.1.4 Funciones del Sistema

A continuación, se presentan las funcionalidades requeridas para la implementación de la solución en la Tabla 3-1.

Ref #	Función	Categoría
R 1	Capturar datos del ambiente, para hacerse una imagen del entorno, y de sí mismo en él.	Evidente
R 2	Determinar una dirección a través de datos del ambiente para llegar a un objetivo final.	Oculto
R 3	Utilizar efectores para guiar al cuerpo del agente inteligente.	Evidente
R 4	Mantener una comunicación constante con el agente inteligente.	Oculto
R 5	Saber las decisiones que ha tomado y su motivo.	Evidente
R 6	Influir en la trayectoria o directamente en el movimiento del cuerpo del agente inteligente.	Evidente
R 7	Saber si los distintos dispositivos están activos y en correcto funcionamiento.	Evidente
R 8	Proveer de las acciones correspondientes a cada usuario del sistema.	Oculto
R 9	Proporcionar un mecanismo para detener totalmente el funcionamiento del cuerpo del agente inteligente a distancia.	Evidente

Tabla 3-1. Funcionalidades del sistema.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.2.1.5 Atributos del Sistema

A continuación, se muestra una lista de características necesarias que deberá lograr la solución, las cuales se exponen en la Tabla 3-2.

<i>Atributo</i>	<i>Detalles y restricciones de frontera</i>
<i>Facilidad de Uso.</i>	<ol style="list-style-type: none"> 1. Sistema de monitoreo que permite la retroalimentación ante los usuarios de las actividades que está llevando a cabo el agente inteligente. 2. Interfaz de usuario muy visual (diseño agradable a la vista) y con descripción significativa (ayuda en tiempo real). 3. Opción de entrenamiento del agente inteligente llevado a cabo en el sistema lógico a bordo.
<i>Metáfora de Interfaz.</i>	<ol style="list-style-type: none"> 1. Ambiente Gráfico: <ol style="list-style-type: none"> i. Símbolos de diversos colores y con formas contrastantes entre sí, es decir, notoriamente diferentes, para la descripción del terreno que ha reconocido el agente inteligente. ii. Gráficos explicativos de la información que ha obtenido el agente inteligente que identifiquen de qué sensores proviene. iii. Área para poder observar los estados internos de las decisiones sobre la nueva ruta a seguir del agente inteligente. 2. La información representada gráficamente obtenida de los distintos sensores del agente inteligente deberá tener una mínima diferencia con respecto a la realidad.
<i>Tolerancia a Fallas.</i>	<ol style="list-style-type: none"> 1. El cuerpo del agente inteligente proporcionará un mecanismo manual para poder detener el funcionamiento de su cuerpo.
<i>Tiempo de Respuesta.</i>	<ol style="list-style-type: none"> 1. Obtención de datos desde el ambiente como máximo de 1 segundos. <ol style="list-style-type: none"> i. Mediciones obtenidas desde el entorno mediante los distintos sensores. ii. Mediciones de dirección y posicionamiento geográfico. 2. Consulta de datos desde el agente inteligente como máximo de 1 segundos. 3. Toma de decisión de próxima trayectoria a seguir como máximo de 4 segundos. 4. Transmisión de orden de detención desde el sistema remoto de monitoreo hacia el agente inteligente en un máximo de 1 segundo.
<i>Seguridad</i>	<ol style="list-style-type: none"> 1. Debe dar las opciones de operación en el sistema correspondiente al nivel de cada usuario.

Tabla 3-2. Atributos del sistema.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.2.1.6 Tabla Combinada

En la Tabla 3-3 se detallan las relaciones entre las funcionalidades de la solución descritas en la Tabla 3-1 y los atributos (véase Tabla 3-2) necesarios para cada una de ellas.

Ref #	Función	Categoría	Atributo
R 1	<i>Capturar datos del ambiente, para hacerse una imagen del entorno, y de si mismo en él.</i>	<i>Evidente</i>	<i>Metáfora de Interfaz (1.i, 1.ii, 2); Tiempo de Respuesta (1); Facilidad de Utilizar (1, 2)</i>
R 2	<i>Determinar una dirección a través de datos del ambiente para llegar a un objetivo final.</i>	<i>Ocultas</i>	<i>Tiempo de Respuesta (3)</i>
R 3	<i>Utilizar efectores para guiar al cuerpo del agente inteligente.</i>	<i>Evidente</i>	<i>Tiempo de Respuesta (4)</i>
R 4	<i>Mantener una comunicación constante con el agente inteligente.</i>	<i>Ocultas</i>	<i>Tiempo de Respuesta (2)</i>
R 5	<i>Saber las decisiones que ha tomado y su motivo.</i>	<i>Evidente</i>	<i>Tiempo de Respuesta (2)</i>
R 6	<i>Influir en la trayectoria o directamente en el movimiento del cuerpo del agente inteligente.</i>	<i>Evidente</i>	<i>Tolerancia a Fallas; Tiempo de Respuesta (2)</i>
R 7	<i>Saber si los distintos dispositivos están activos y en correcto funcionamiento.</i>	<i>Evidente</i>	<i>Tiempo de Respuesta (2); Facilidad de Utilizar (1, 2); Metáfora de Interfaz (1.i; 1.ii; 2)</i>
R 8	<i>Proveer de las acciones correspondientes a cada usuario del sistema.</i>	<i>Ocultas</i>	<i>Seguridad</i>
R 9	<i>Proporcionar un mecanismo para detener totalmente el funcionamiento del cuerpo del agente inteligente a distancia.</i>	<i>Evidente</i>	<i>Facilidad de Utilizar (1, 2); Metáfora de Interfaz (1.i; 1.ii); Tiempo de Respuesta (4)</i>

Tabla 3-3. Tabla combinada entre requerimientos funcionales y no funcionales del sistema.

3.2.1.7 Observación

Se puede desprender, de las funcionalidades necesarias para el agente inteligente y su monitorización, que para darle independencia de movilidad, se deberán utilizar dos sistemas comunicados entre sí inalámbricamente, y lograr, así, cumplir con la característica de monitoreo remoto. Estos sistemas identificados los denominaremos como:

- Sistema de Agente Inteligente. (SAI)
- Sistema de Monitoreo. (SiM)

3.2.2 Diagrama de Componentes del Sistema

En la Figura 3-1 se muestra un esquema de componentes de la solución que se deberá lograr.

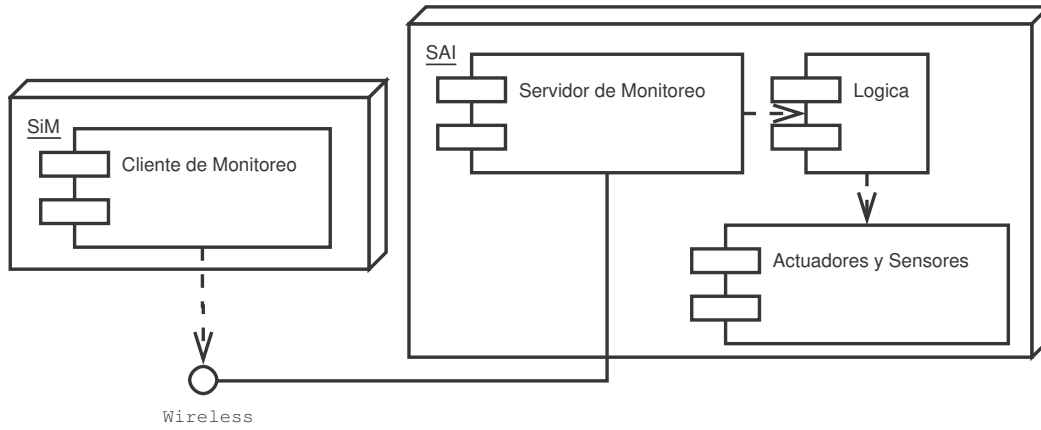


Figura 3-1. Diagrama de Componentes de Sistema.

3.2.3 Descripción de Procesos

En la Figura 3-2 y Figura 3-3 se darán a conocer los diagramas de casos de uso correspondientes a los dos sistemas: SAI y SiM.

3.2.3.1 Diagramas de Casos de Uso

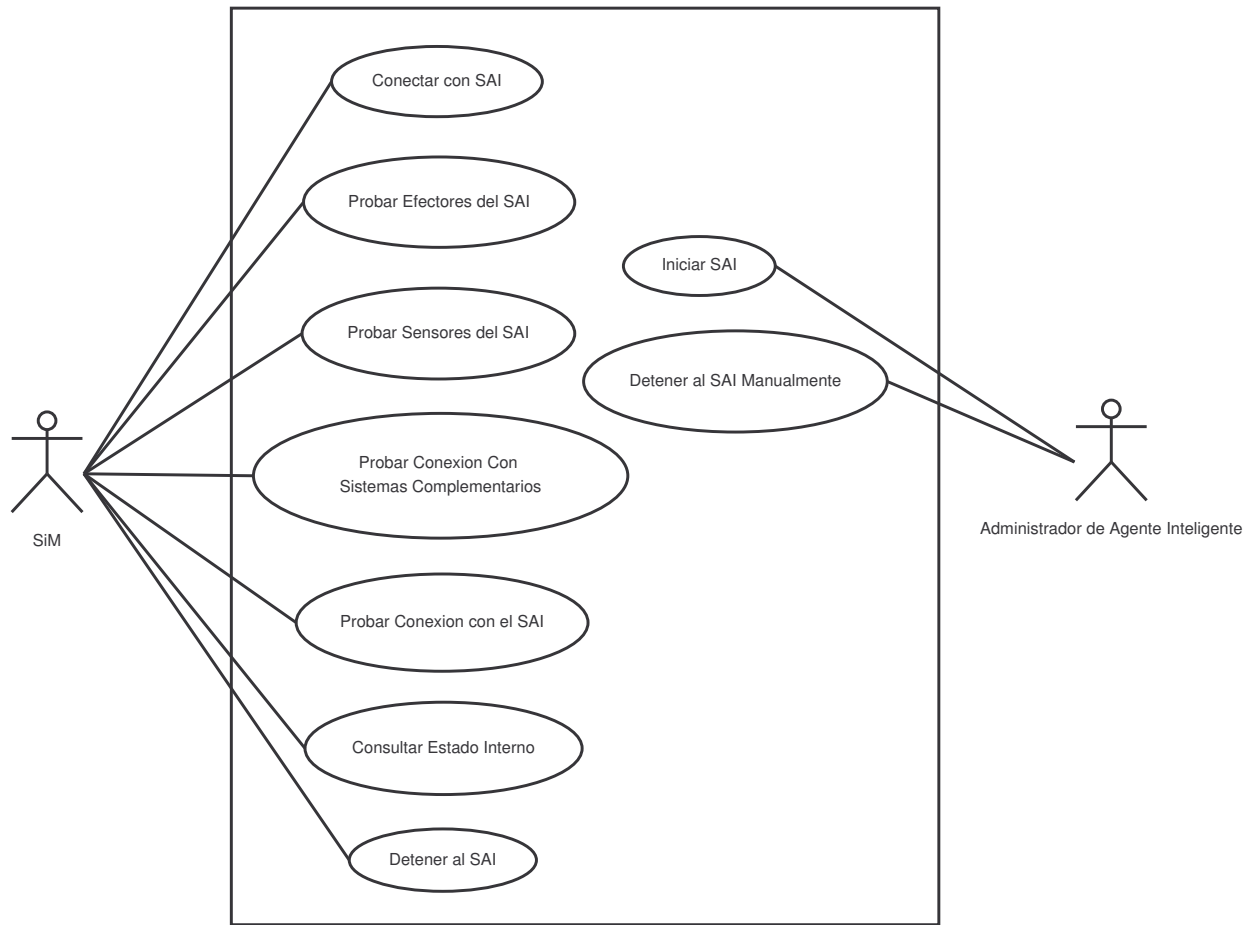


Figura 3-2. Diagrama de Casos de Uso de SAI.

CAPITULO 3: DESARROLLO DEL PROYECTO

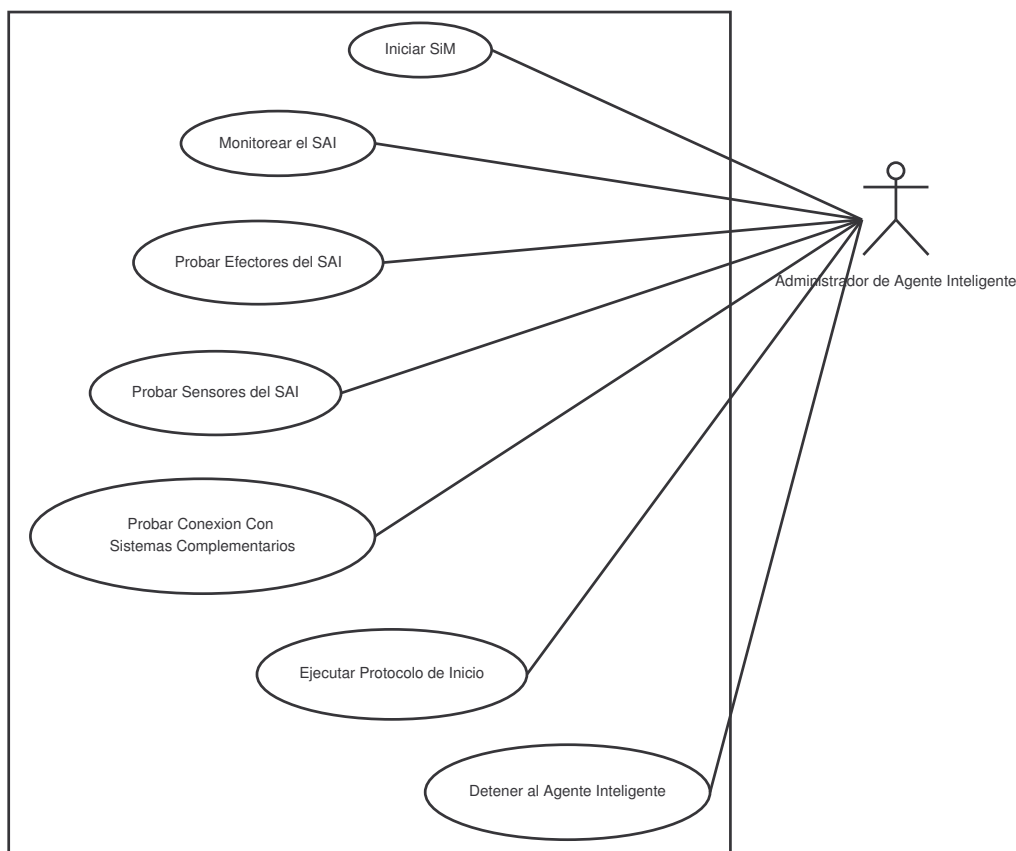


Figura 3-3. Diagrama de Casos de Uso de SiM.

3.2.3.2 Casos de Uso de Alto Nivel

3.2.3.2.1 Justificación de Casos de Uso

Craig Larman (Larman Craig, 1999) menciona que cada caso de uso corresponde a una funcionalidad del sistema. Por lo cual, se considerarán cada uno de los requerimientos funcionales como un caso de uso en particular, salvo aquellos que ameriten un poco más de especificación.

Del análisis de los requerimientos funcionales, podemos desprender, en la Tabla 3-4, los siguientes actores involucrados en cada uno de los escenarios de uso de cada sistema.

<i>Sistema de Agente Inteligente:</i>	<i>Sistema de Monitoreo:</i>
1. <i>Sistema de Monitoreo.</i>	1. <i>Usuario Monitor.</i>
<i>Comunes:</i>	
1. <i>Administrador de Agente Inteligente.</i>	

Tabla 3-4. Actores de los casos de uso.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.2.3.2.2 Casos de Uso para Sistema de Agente Inteligente

Desde la Tabla 3-5 hasta la Tabla 3-13 se darán a conocer los casos de uso de alto nivel para el SAI.

<i>Caso de Uso:</i>	<i>Iniciar el SAI.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Administrador del Agente Inteligente enciende el computador a bordo del Agente Inteligente y, posteriormente, luego de haber entrado en el sistema operativo, procede a iniciar el Sistema de Agente Inteligente.</i>

Tabla 3-5. Casos de uso “Iniciar el SAI”.

<i>Caso de Uso:</i>	<i>Probar Conexión con Sistemas Complementarios.</i>
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Sistema de Monitoreo solicita la prueba de la conexión con los Sistemas Complementarios del SAI.</i>

Tabla 3-6. Casos de uso “Probar Conexión con Sistemas Complementarios”.

<i>Caso de Uso:</i>	<i>Probar Efectores del SAI.</i>
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Sistema de Monitoreo solicita la prueba de reacción de los motores de dirección, tracción.</i>

Tabla 3-7. Casos de uso “Probar Efectores del SAI”.

<i>Caso de Uso:</i>	<i>Probar Sensores del SAI.</i>
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Sistema de Monitoreo solicita la prueba de los sensores de ultrasonido de ambiente, de terreno y de la brújula.</i>

Tabla 3-8. Caso de uso “Probar Sensores del SAI”.

<i>Caso de Uso:</i>	<i>Conectar con el SAI.</i>
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Sistema de Monitoreo solicita una conexión al SAI, para obtener información del estado interno y/o enviarle órdenes a este, para esto el SIM deberá autenticarse, dependiendo del tipo de autenticación se le asignará privilegios de monitorización y/o mando.</i>

Tabla 3-9. Caso de uso “Conectar con el SAI”.

<i>Caso de Uso:</i>	<i>Probar conexión con el SAI.</i>
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Sistema de Monitoreo solicita una prueba de conexión con el SAI, para obtener información del estado de la conexión, este esperará una confirmación de reconocimiento de la ejecución de la prueba por el SAI.</i>

Tabla 3-10. Caso de uso “Probar conexión con el SAI”.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Consultar Estado Interno.</i>
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>Sistema de Monitoreo se inicia y solicita información del estado interno del Sistema de Agente Inteligente, esto es, las mediciones realizadas por los sensores (ultrasónicos, DGPS, compás) y las decisiones tomadas por el Agente Inteligente.</i>

Tabla 3-11. Caso de uso “Consultar Estado Interno”.

<i>Caso de Uso:</i>	<i>Detener al SAI.</i>
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Sistema de Monitoreo (SiM) pide la detención de actividades del Agente Inteligente. Para lo cual solicitará al Sistema de Agente Inteligente, el cese de sus actividades.</i>

Tabla 3-12. Caso de uso “Detener al SAI”.

<i>Caso de Uso:</i>	<i>Detener al SAI Manualmente.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Administrador de Agente Inteligente, una vez iniciado el Sistema de Monitoreo, necesitará la detención de actividades del Agente Inteligente manualmente, cuando la orden de detención falle o para la finalización normal de actividades de este. Para lo cual existe un switch físico en el cuerpo del Agente Inteligente, el cual es activado por el Administrador de Agente Inteligente.</i>

Tabla 3-13. Caso de uso “Detener al SAI Manualmente”.

3.2.3.2.3 Casos de Uso para Sistema de Monitoreo

Desde la Tabla 3-14 hasta Tabla 3-20 se darán a conocer los casos de uso de alto nivel para el SiM.

<i>Caso de Uso:</i>	<i>Iniciar el SiM.</i>
<i>Actores:</i>	<i>Usuario Monitor y Administrador de Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Usuario Monitor o el Administrador de Agente Inteligente inicia el software de Sistema de Monitoreo, y procede a autenticarse para dar inicio, y proveerle de las opciones correspondientes.</i>

Tabla 3-14. Caso de uso “Iniciar el SiM”.

<i>Caso de Uso:</i>	<i>Probar Conexión con Sistemas Complementarios.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Administrador de Agente Inteligente solicita la prueba de la conexión con los Sistemas Complementarios del SAI.</i>

Tabla 3-15. Caso de uso “Probar Conexión con Sistemas Complementarios”.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Probar Efectores del SAI.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Administrador de Agente inteligente solicita la prueba de los efectores del SAI. Este comprobará visualmente si el resultado de la prueba es satisfactoria.</i>

Tabla 3-16. Caso de uso “Probar Efectores del SAI”.

<i>Caso de Uso:</i>	<i>Probar Sensores del SAI.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Administrador de Agente Inteligente solicita la prueba de los sensores del SAI, el cual entregará un informe de la operatividad de los sensores.</i>

Tabla 3-17. Caso de uso “Probar Sensores del SAI”.

<i>Caso de Uso:</i>	<i>Monitorear el SAI.</i>
<i>Actores:</i>	<i>Usuario Monitor, Administrador de Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Usuario Monitor o el Administrador de Agente Inteligente procede a establecer una conexión con el SAI, y así solicitarle datos sobre su entorno, su dirección, ubicación geográfica, decisión de ruta a seguir, además de otras variables. Todo lo anterior, con el fin de realizar el respectivo despliegue de información gráfica para el Usuario Monitor o el Administrador de Agente Inteligente.</i>

Tabla 3-18. Caso de uso “Monitorear el SAI”.

<i>Caso de Uso:</i>	<i>Detener al Agente Inteligente.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Administrador de Agente Inteligente desea detener el Agente Inteligente. Para lo cual solicita al SiM para que éste se comuniquen con el SAI indicándole que detenga el movimiento del cuerpo físico.</i>

Tabla 3-19. Caso de uso “Detener al Agente Inteligente”.

<i>Caso de Uso:</i>	<i>Ejecutar Protocolo de Inicio.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Administrador de Agente Inteligente verifica si todos los sensores, efectores y sistemas complementarios, están operativos y listos, mediante solicitudes al SiM, para así hacer trabajar al Agente Inteligente.</i>

Tabla 3-20. Caso de uso “Ejecutar Protocolo de Inicio”.

3.2.3.3 Casos de Uso Expandidos

A continuación se realizará una definición de los casos de uso expandidos. Dentro de las referencias cruzadas se hace alusión a los requisitos funcionales, los cuales se encuentran codificados como se acordó en la sección 3.2.1.4 anteriormente tratada.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.2.3.3.1 Sistema de Agente Inteligente

Desde la Tabla 3-21 hasta la Tabla 3-28 se darán a conocer los casos de uso expandidos para el SAI.

<i>Caso de Uso:</i>	<i>Consultar Estado Interno.</i>	
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>	
<i>Propósito:</i>	<i>Proveer al Sistema de Monitoreo una conexión mediante la cual pueda consultar el estado interno del agente.</i>	
<i>Resumen:</i>	<i>Sistema de Monitoreo se inicia y solicita información del estado interno del Sistema de Agente Inteligente, esto es, las mediciones realizadas por los sensores (ultrasonicos, DGPS, compás) y las decisiones tomadas por el Agente Inteligente.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	<i>R 1, R 2, R 5.1, R 5.3</i>	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- El sistema remoto cliente (SiM) , pide información del estado interno del SAI.</i>	<i>2.- El SAI envía un compendio de información al SiM.</i>	

Tabla 3-21. Caso de uso “Consultar Estado Interno”.

<i>Caso de Uso:</i>	<i>Iniciar SAI.</i>	
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>	
<i>Propósito:</i>	<i>Encender el computador, y ejecutar el software del Agente Inteligente.</i>	
<i>Resumen:</i>	<i>El Administrador del Agente Inteligente enciende el computador a bordo del Agente Inteligente y, posteriormente, luego de haber entrado en el sistema operativo, procede a iniciar el Sistema de Agente Inteligente.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	<i>R 7</i>	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- Este caso de uso comienza con el encendido del computador a bordo del robot por el Administrador de Agente Inteligente.</i>		
<i>2.- El Administrador de Agente Inteligente, una vez ingresado al sistema operativo, inicia un software de verificación de las distintas conexiones.</i>	<i>3a.- El SAI retorna una respuesta satisfactoria a todos los intentos de conexiones.</i>	
<i>4a.- El Administrador de Agente Inteligente inicia el SAI del robot.</i>		
<i>Curso Alternativo de Eventos:</i>		
	<i>3b.- El SAI retorna como respuesta la imposibilidad de conectar con uno o más sistemas.</i>	
<i>4b.- El Administrador de Agente Inteligente no intenta iniciar el SAI y busca el error.</i>		

Tabla 3-22. Caso de uso “Iniciar SAI”.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Probar Conexión con Sistemas Complementarios.</i>	
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>	
<i>Tipo:</i>	<i>Primario.</i>	
<i>Descripción:</i>	<i>El Sistema de Monitoreo solicita la prueba de la conexión con los Sistemas Complementarios del SAI.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	<i>R 4</i>	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- Este caso de uso comienza cuando el Sistema de Monitoreo solicita la prueba de la conexión con los Sistemas Complementarios del SAI.</i>	<i>2.- El SAI ejecuta las pruebas de conexiones con los Sistemas Complementarios.</i>	
	<i>3a.- El SAI informa al SiM que todas las conexiones con los Sistemas Complementarios fueron logradas.</i>	
<i>Curso Alternativo de Eventos:</i>		
	<i>3b.- El SAI informa al SiM que una o más conexiones con los Sistemas Complementarios no fueron logradas.</i>	

Tabla 3-23. Caso de uso “Probar Conexión con Sistemas Complementarios”.

<i>Caso de Uso:</i>	<i>Conectar Con SAI.</i>	
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>	
<i>Propósito:</i>	<i>Establecer una conexión para el intercambio de información.</i>	
<i>Resumen:</i>	<i>El Sistema de Monitoreo solicita una conexión al SAI, para obtener información del estado interno y/o enviar ordenes a este, para esto el SiM deberá autenticarse, dependiendo del tipo de autenticación se le asignara privilegios de monitorización y/o mando.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	<i>R 4, R 7</i>	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- Este caso de uso comienza con el recibimiento de una orden proveniente de un SiM, solicitando una conexión con SAI.</i>		
	<i>2a.- El sistema autentifica el password proporcionado y acepta la conexión.</i>	
<i>Curso Alternativo de Eventos:</i>		
	<i>2b.- El sistema rechaza la conexión, ya que la autenticación del password resultó negativa.</i>	

Tabla 3-24. Caso de uso “Conectar Con SAI”.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Detener al SAI Manualmente.</i>	
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>	
<i>Propósito:</i>	<i>Detener el todos los dispositivos involucrados en el agente.</i>	
<i>Resumen:</i>	<i>El Administrador de Agente Inteligente, una vez iniciado el Sistema de Monitoreo, necesitará la detención de actividades del Agente Inteligente manualmente, cuando la orden de detención falle o para la finalización normal de actividades de este. Para lo cual existe un switch físico en el cuerpo del Agente Inteligente, el cual es activado por el Administrador de Agente Inteligente.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	<i>R 6</i>	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- Este caso de uso comienza cuando el Administrador del Agente Inteligente quiere suspender la actividad de los efectores del Agente Inteligente, desactivando el switch de energía de estos.</i>	<i>2.- El cuerpo del Agente Inteligente responde deteniéndose.</i>	

Tabla 3-25. Caso de uso “Detener a Agente Inteligente Manualmente”.

<i>Caso de Uso:</i>	<i>Detener al SAI.</i>	
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>	
<i>Propósito:</i>	<i>Detener el desplazamiento del robot</i>	
<i>Resumen:</i>	<i>El Sistema de Monitoreo (SiM) pide la detención de actividades del Agente Inteligente. Para lo cual solicitara al Sistema de Agente Inteligente, el cese de sus actividades.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	<i>R 6</i>	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- Este caso de uso comienza con el recibimiento de una orden crítica de parada de parte del Sistema de Monitoreo.</i>		
	<i>2.- El SAI verifica si la conexión establecida por ese Sistema de Monitoreo se registró como administrador.</i>	
	<i>3a.- El SAI ejecuta la orden de cese total de actividades.</i>	
<i>Curso Alternativo de Eventos:</i>		
	<i>3b.- El SAI rechaza la orden por no tener el Sistema de Monitoreo permisos de administrador.</i>	

Tabla 3-26. Caso de uso “Detener a Agente Inteligente”.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Probar Efectores del SAI.</i>	
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>	
<i>Propósito:</i>	<i>Determinar si la dirección del Agente Inteligente funciona correctamente.</i>	
<i>Resumen:</i>	<i>El Sistema de Monitoreo solicita la prueba de reacción de los motores de dirección, tracción.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	<i>R 7</i>	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- Este caso de uso comienza con el recibimiento de una orden proveniente de un SiM, solicitando una prueba de efectores.</i>		
	<i>2.- El SAI verifica si la conexión establecida por ese Sistema de Monitoreo se registro como administrador.</i>	
	<i>3a.- El SAI ejecuta la orden de prueba de efectores.</i>	
<i>Curso Alternativo de Eventos:</i>		
	<i>3b.- El SAI rechaza la orden por no tener, el Sistema de Monitoreo, permisos de administrador.</i>	

Tabla 3-27. Caso de uso “Probar Efectores del SAI”.

<i>Caso de Uso:</i>	<i>Probar Sensores del SAI.</i>	
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>	
<i>Propósito:</i>	<i>Determinar si la tracción del Agente Inteligente funciona correctamente.</i>	
<i>Resumen:</i>	<i>El Sistema de Monitoreo solicita la prueba de los sensores de ultrasonido de ambiente, de terreno y del la brújula.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	<i>R 7</i>	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- Este caso de uso comienza con el recibimiento de una orden proveniente de un SiM, solicitando una prueba de sensores.</i>		
	<i>2.- El SAI verifica si la conexión establecida por ese Sistema de Monitoreo se registró como administrador.</i>	
	<i>3a.- El SAI ejecuta la orden de prueba de sensores.</i>	
<i>Curso Alternativo de Eventos:</i>		
	<i>3b.- El SAI rechaza la orden por no tener, el Sistema de Monitoreo, permisos de administrador.</i>	

Tabla 3-28. Caso de uso “Probar Sensores del SAI”.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Probar Conexión con SAI.</i>
<i>Actores:</i>	<i>Sistema de Monitoreo.</i>
<i>Propósito:</i>	<i>Determinar si el SiM tiene una conexión activa.</i>
<i>Resumen:</i>	<i>El Sistema de Monitoreo solicita una prueba de conexión con el SAI, para obtener información del estado de la conexión, este esperara un confirmación de reconocimiento de la ejecución de la prueba por el SAI.</i>
<i>Tipo:</i>	<i>Primario y Esencial.</i>
<i>Referencia Cruzada:</i>	<i>R 7</i>
<i>Curso Normal de Eventos:</i>	
<i>Actor</i>	<i>Sistema</i>
<i>1.- Este caso de uso comienza con el recibimiento de una orden proveniente de un SiM, solicitando una prueba de conexión.</i>	
	<i>2a.- El SAI retorna al SiM una confirmación de la conexión.</i>
<i>Curso Alternativo de Eventos:</i>	
	<i>2b.- El SAI no retorna ningún asentimiento.</i>

Tabla 3-29. Caso de uso “Probar Conexión con SAI”.

3.2.3.3.2 Sistema de Monitoreo

Desde la Tabla 3-30 hasta la Tabla 3-36 se muestran casos de uso expandidos para el SiM.

<i>Caso de Uso:</i>	<i>Monitorear SAI.</i>
<i>Actores:</i>	<i>Usuario Monitor ó Administrador de Agente Inteligente.</i>
<i>Propósito:</i>	<i>El seguimiento y análisis de las distintas mediciones sensoriales y variables internas del Agente inteligente.</i>
<i>Resumen:</i>	<i>El Usuario Monitor o el Administrador de Agente Inteligente piden al SiM monitorear al SAI, solicitando éste datos sobre su entorno, su dirección, ubicación geográfica, decisión de ruta a seguir. Todo lo anterior, con el fin de realizar el respectivo despliegue de información gráfica para el Usuario Monitor o el Administrador de Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario y Esencial.</i>
<i>Referencia Cruzada:</i>	<i>R 5.1, R 5.2, 5.3, R 7</i>
<i>Curso Normal de Eventos:</i>	
<i>Actor</i>	<i>Sistema</i>
<i>1.- Este caso de uso comienza cuando el Usuario Monitor o el Administrador de Agente Inteligente da la orden al SiM de monitorear al SAI.</i>	
	<i>2a.- El SiM muestra en pantalla la información de monitoreo.</i>
<i>Curso Alternativo de Eventos:</i>	
	<i>2b.- El SiM rechaza la orden porque el Administrador de Agente Inteligente no ha ejecutado el protocolo de inicio.</i>

Tabla 3-30. Caso de uso “Monitorear SAI”.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Iniciar SiM.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente y/o Usuario Monitor.</i>
<i>Propósito:</i>	<i>Identificar al Usuario del Sistema Monitor, y así dar las opciones correspondientes.</i>
<i>Resumen:</i>	<i>El Usuario Monitor o el Administrador de Agente Inteligente inicia el software de Sistema de Monitoreo, y procede a autenticarse para dar inicio, y proveerle de las opciones correspondientes.</i>
<i>Tipo:</i>	<i>Primario y Esencial.</i>
<i>Referencia Cruzada:</i>	<i>R 8</i>
<i>Curso Normal de Eventos:</i>	
<i>Actor</i>	<i>Sistema</i>
<i>1.- Este caso de uso comienza con la autenticación del usuario, proveyendo éste de un password.</i>	
	<i>2.- El SiM pide una conexión al SAI.</i>
	<i>3a.- El SiM informa al Administrador de Agente Inteligente o Usuario Monitor del logro de la conexión, y se habilitaran todas las opciones ó sólo la de monitoreo, según corresponda.</i>
<i>Curso Alternativo de Eventos:</i>	
	<i>3b.- El SiM informa al Administrador de Agente Inteligente o Usuario Monitor que no se ha podido lograr la conexión.</i>

Tabla 3-31. Caso de uso “Iniciar SiM”.

<i>Caso de Uso:</i>	<i>Probar Conexión con Sistemas Complementarios.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario.</i>
<i>Descripción:</i>	<i>El Administrador de Agente Inteligente solicita la prueba de la conexión con los Sistemas Complementarios del SAI.</i>
<i>Tipo:</i>	<i>Primario y Esencial.</i>
<i>Referencia Cruzada:</i>	<i>R 4</i>
<i>Curso Normal de Eventos:</i>	
<i>Actor</i>	<i>Sistema</i>
<i>1.- Este caso de uso comienza cuando el Administrador de Agente Inteligente, da la orden al SiM de probar la conexión con SAI.</i>	<i>2.-El SiM ejecuta la orden de petición de prueba de conexión con SAI.</i>
	<i>3a.- El SiM informa al Administrador de Agente Inteligente que la conexión con el SAI está operativa.</i>
<i>4.- El Administrador de Agente Inteligente solicita la prueba de la conexión con los Sistemas Complementarios del SAI.</i>	<i>5.- El SiM ejecuta la orden de petición de pruebas de conexiones con los Sistemas Complementarios.</i>
	<i>6a.- El SiM informa al Administrador de Agente Inteligente que todas las conexiones con los Sistemas Complementarios fueron logradas.</i>
<i>Curso Alternativo de Eventos:</i>	
	<i>3a.- El SiM informa al Administrador de Agente Inteligente que la conexión con SAI no esta operativa.</i>
	<i>6b.- El SiM informa al Administrador de Agente Inteligente que una o más conexiones con los Sistemas Complementarios no fueron logradas.</i>

Tabla 3-32. Caso de uso “Probar Conexión con Sistemas Complementarios”.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Ejecutar Protocolo de Inicio.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>
<i>Propósito:</i>	<i>Determinar si los distintos sistemas complementarios y efectores están funcionando correctamente.</i>
<i>Resumen:</i>	<i>El Administrador de Agente Inteligente verifica si todos los sensores, efectores y sistemas complementarios, están operativos y listos, mediante solicitudes al SiM, para así hacer trabajar al Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario y Esencial.</i>
<i>Referencia Cruzada:</i>	<i>R 7, Probar Efectores del SAI, Probar Sensores del SAI.</i>
<i>Curso Normal de Eventos:</i>	
<i>Actor</i>	<i>Sistema</i>
<i>1.- El Administrador del Agente Inteligente ejecuta el caso de uso Probar Conexión con Sistemas Complementarios.</i>	
<i>2.- Ahora, el Administrador del Agente Inteligente ejecuta el caso de uso Probar Efectores del SAI.</i>	
<i>3.- Ahora, el Administrador del Agente Inteligente ejecuta el caso de uso Probar Sensores del SAI.</i>	

Tabla 3-33. Caso de uso “Ejecutar Protocolo de Inicio”.

<i>Caso de Uso:</i>	<i>Detener al Agente inteligente.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>
<i>Propósito:</i>	<i>Detener el desplazamiento del robot</i>
<i>Resumen:</i>	<i>El Administrador de Agente Inteligente desea detener el Agente Inteligente. Para lo cual solicita al SiM para que éste se comuniquen con el SAI indicándole que detenga el movimiento del cuerpo físico.</i>
<i>Tipo:</i>	<i>Primario y Esencial.</i>
<i>Referencia Cruzada:</i>	<i>R 6</i>
<i>Curso Normal de Eventos:</i>	
<i>Actor</i>	<i>Sistema</i>
<i>1.- Este caso de uso comienza con orden de parada dada por el Administrador del Agente Inteligente.</i>	
	<i>2.- El SiM envía una orden crítica de parada a SAI.</i>
	<i>3a.- El SiM confirma al Administrador de Agente Inteligente que el SAI ha detenido sus acciones.</i>
<i>Curso Alternativo de Eventos:</i>	
	<i>3b.- El SiM informa al Administrador de Agente Inteligente que el SAI no ha podido detener sus acciones.</i>

Tabla 3-34. Caso de uso “Detener a Agente inteligente”.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Probar Efectores del SAI.</i>	
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>	
<i>Propósito:</i>	<i>Determinar si la dirección del Agente Inteligente funciona correctamente.</i>	
<i>Resumen:</i>	<i>El Administrador de Agente inteligente solicita la prueba de los efectores del SAI. Este comprobará visualmente si el resultado de la prueba es satisfactoria.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	R 7	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- Este caso de uso comienza con la orden, proveniente del Administrador de Agente Inteligente, solicitando una prueba de efectores.</i>		
	<i>2.- El SiM envía una orden al SAI pidiendo ejecutar la prueba de efectores.</i>	
<i>3a.- El Administrador de Agente Inteligente confirma visualmente que la prueba de los efectores se ha realizado correctamente.</i>		
<i>Curso Alternativo de Eventos:</i>		
<i>3b.- El Administrador de Agente Inteligente determina visualmente que la prueba de los efectores no se ha realizado correctamente.</i>		

Tabla 3-35. Caso de uso “Probar Efectores del SAI”.

<i>Caso de Uso:</i>	<i>Probar Sensores del SAI.</i>	
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>	
<i>Propósito:</i>	<i>Determinar si la tracción del Agente Inteligente funciona correctamente.</i>	
<i>Resumen:</i>	<i>El Administrador de Agente Inteligente solicita la prueba de los sensores del SAI, el cual entregará un informe de la operatividad de los sensores.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	R 7	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- Este caso de uso comienza con el recibimiento de una orden proveniente del Administrador de Agente Inteligente, solicitando una prueba de sensores.</i>		
	<i>2.- El SiM envía una orden al SAI pidiendo ejecutar la prueba de sensores.</i>	
	<i>3a.- El SiM informa que las mediciones realizadas por los sensores son válidas.</i>	
<i>Curso Alternativo de Eventos:</i>		
	<i>3b.- El SiM informa que las mediciones realizadas por los sensores no son válidas.</i>	

Tabla 3-36. Caso de uso “Probar Sensores del SAI”.

3.2.4 Clasificación y Programación de los Casos de Uso

Para realizar una clasificación de los casos de uso de a nivel de contexto se ha considerado la metodología expresada en (Larman Craig, 1999), donde la estrategia consiste en escoger primero los casos que influyen profundamente en la arquitectura básica, determinados a través de puntuaciones (posiblemente incrementadas con ponderación), basándose en cualidades que inciden en la clasificación. (véanse Tabla 3-37 y Tabla 3-38)

Casos de Uso para Sistema de Agente Inteligente:

1. Iniciar SAI.
2. Probar Conexión con Sistemas Complementarios
3. Probar Efectores del SAI.
4. Probar Sensores del SAI.
5. Conectar con SAI.
6. Probar Conexión con SAI.
7. Consultar Estado Interno.
8. Detener al SAI.
9. Detener al SAI Manualmente.

Casos de Uso para Sistema de Monitoreo:

1. Iniciar SiM.
2. Probar Conexión con Sistemas Complementarios.
3. Probar Efectores del SAI.
4. Probar Sensores del SAI.
5. Monitorear al SAI.
6. Detener al Agente Inteligente.
7. Ejecutar Protocolo de Inicio.

CAPITULO 3: DESARROLLO DEL PROYECTO

Puntuación: 0 – 10.

- a) Tener una fuerte repercusión en el diseño arquitectónico; por ejemplo, incorporar muchas clases a la capa del dominio o requerir servicios de persistencia.
- b) Con relativamente poco esfuerzo obtener información e ideas importantes sobre el diseño.
- c) Incluir funciones riesgosas, urgentes o complejas.
- d) Requerir una inversión a fondo o tecnología nueva y riesgosa.
- e) Representar procesos primarios de la línea de negocio.
- f) Apoyar directamente el aumento de ingresos a la reducción de costos.

<i>Sistema de Agente Inteligente (SAI)</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	Σ	<i>Prioridad</i>	<i>Orden más racional</i>
<i>Iniciar SAI.</i>	0	6	5	0	5	0	16	9	2
<i>Probar Conexión con Sistemas Complementarios.</i>	3	4	5	5	1	0	18	5	3
<i>Conectar con SAI.</i>	8	8	10	3	10	0	39	2	4
<i>Probar Efectores del SAI.</i>	3	4	5	5	1	0	18	7	8
<i>Probar Sensores del SAI.</i>	3	4	5	5	1	0	18	8	9
<i>Consultar Estado Interno.</i>	10	10	10	10	10	0	50	1	1
<i>Probar conexión con el SAI.</i>	3	4	5	5	1	0	18	6	7
<i>Detener a Agente Inteligente.</i>	3	4	10	8	10	0	35	4	6
<i>Detener a Agente Inteligente manualmente.</i>	3	4	10	9	10	0	36	3	5

Tabla 3-37. Clasificación de los casos de uso del SAI.

Finalmente, Consultar Estado Interno es el caso de uso con mayor prioridad para el SAI , dado que es la base para el resto.

Por lo tanto, la clasificación de los casos de uso según la prioridad es la siguiente:

1. Consultar Estado Interno.
2. Iniciar SAI.
3. Probar Conexión con Sistemas Complementarios.
4. Conectar con SAI.
5. Detener a Agente Inteligente manualmente.
6. Detener a Agente Inteligente.
7. Probar conexión con SAI.
8. Probar Efectores de SAI.
9. Probar Sensores de SAI.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Sistema de Monitoreo (SiM)</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	Σ	<i>Prioridad</i>	<i>Orden más racional</i>
<i>Iniciar SiM.</i>	5	5	5	0	5	0	15	7	2
<i>Probar Conexión con Sistemas Complementarios.</i>	4	3	5	5	3	0	20	4	3
<i>Probar Efectores del SAI.</i>	4	3	5	5	3	0	20	5	6
<i>Probar Sensores del SAI.</i>	4	3	5	5	3	0	20	6	7
<i>Monitoreo de SAI.</i>	10	10	10	10	10	0	50	1	1
<i>Detener a Agente Inteligente.</i>	4	4	10	5	10	0	33	3	5
<i>Ejecutar Protocolo de Inicio.</i>	7	10	10	5	10	0	42	2	4

Tabla 3-38. Clasificación de los casos de uso del SiM.

Finalmente, Monitoreo de SAI es el caso de uso con mayor prioridad para SiM.

Por lo tanto, la clasificación de los casos de uso según la prioridad es la siguiente:

1. Monitorear SAI.
2. Iniciar SiM.
3. Probar Conexión con Sistemas Complementarios.
4. Ejecutar Protocolo de Inicio.
5. Detener a Agente Inteligente.
6. Probar de Efectores de SAI.
7. Probar de Sensores de SAI.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.2.4.1 Asignación de los Casos de Uso a Incrementos

En la Figura 3-4 se da a conocer los incrementos y sus respectivos casos de uso para cada uno de los sistemas.

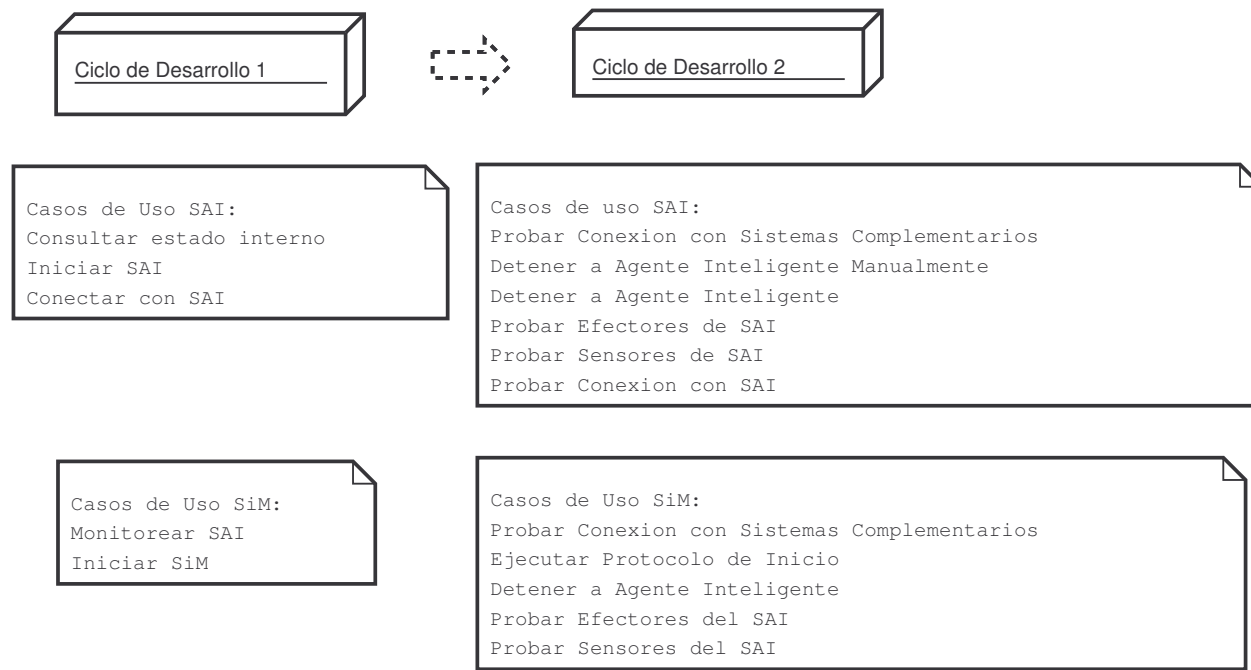


Figura 3-4. Diagrama de asignación de casos de uso a los incrementos.

La elección de los casos de uso para los primeros incrementos se justifica por lo crítico de la comunicación inalámbrica, es por eso que se ha escogido la captura remota de los datos como de primera prioridad en el desarrollo, ya que esto determina la factibilidad del proyecto. Además, los casos de uso referentes a iniciar sistema se hacen imprescindibles al momento de iniciar cualquier ciclo de desarrollo, dado que son la base para trabajar con el sistema.

3.3 Análisis Orientado a Objeto: Primer Incremento

En la presente sección se darán a conocer aspectos del análisis correspondientes al primer incremento de los sistemas de Monitoreo y Agente Inteligente, donde se encontrarán los respectivos modelos conceptuales y diagramas de secuencia definidos para los casos de uso mostrados en la Figura 3-4.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.3.1 Modelo Conceptual

A continuación se presentarán los modelos conceptuales de los sistemas, en la Figura 3-5 y la Figura 3-6, tomado conceptos del dominio del problema descrito en la sección 1.2.

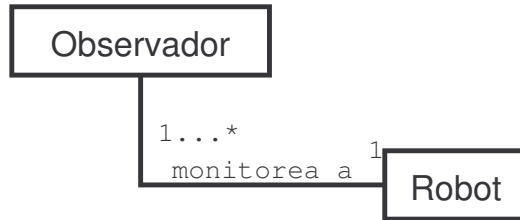


Figura 3-5. Diagrama de Modelo Conceptual SiM.

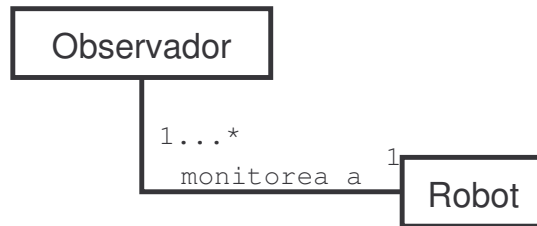


Figura 3-6. Diagrama de Modelo Conceptual SAI.

3.3.2 Diagramas de Secuencia de SiM

La Figura 3-7 y la Figura 3-8 dan a conocer los diagramas de secuencia correspondientes a los casos de uso asignados al primer incremento para el SiM. Estos diagramas están relacionados con los casos de uso mostrados en la Tabla 3-30 y la Tabla 3-31.

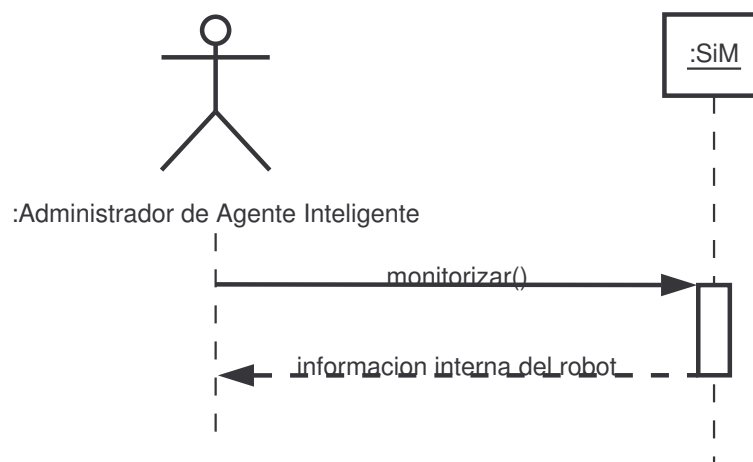


Figura 3-7. Diagrama Monitorear SAI.

CAPITULO 3: DESARROLLO DEL PROYECTO

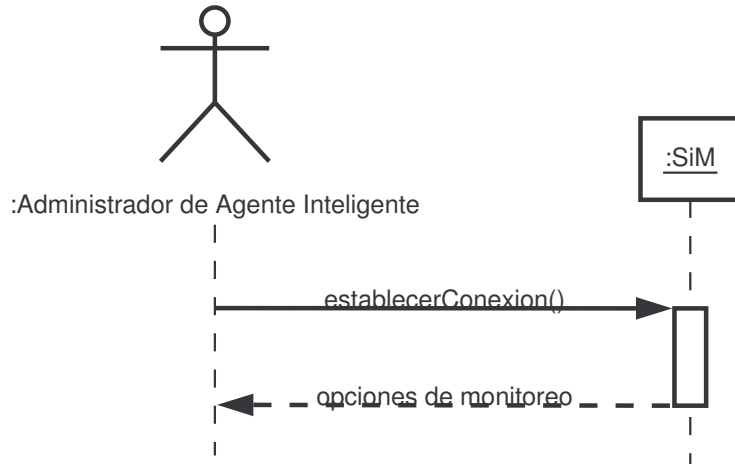


Figura 3-8. Diagrama Iniciar SiM.

3.3.3 Diagramas de Secuencia del SAI

Desde la Figura 3-9 hasta la Figura 3-11 se darán a conocer los diagramas de secuencia correspondientes a los casos de uso asignados al primer incremento para el SAI. Estos diagramas están relacionados con los casos de uso mostrados en la Tabla 3-21, Tabla 3-22 y la Tabla 3-24.

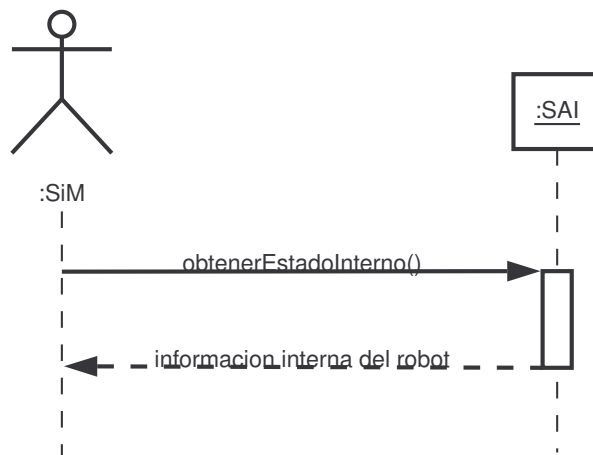


Figura 3-9. Diagrama Consultar estado interno.

CAPITULO 3: DESARROLLO DEL PROYECTO

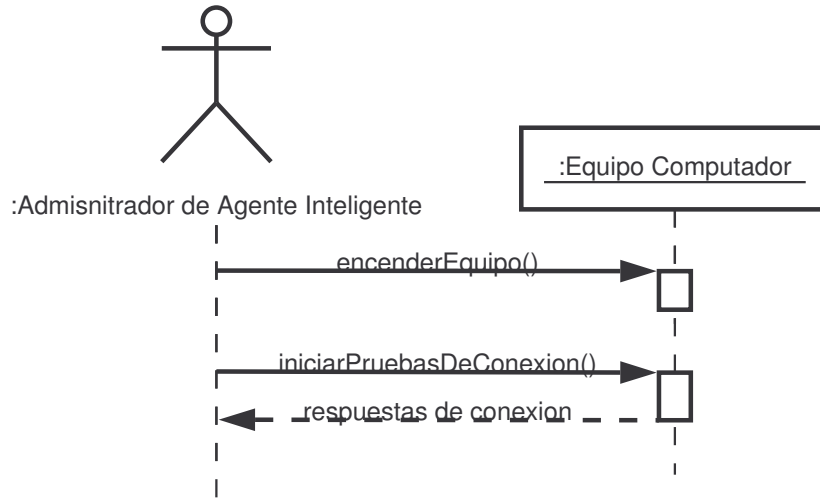


Figura 3-10. Diagrama Iniciar SAI.

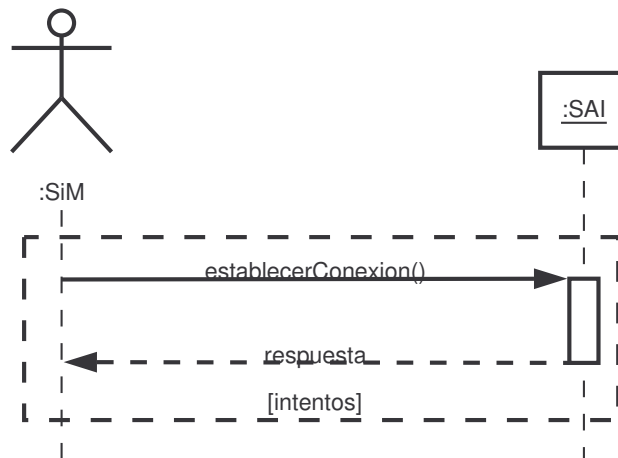


Figura 3-11. Diagrama Conectar con SAI.

3.4 Diseño Orientado a Objeto: Primer Incremento

En la presente sección, se darán a conocer aspectos concernientes al diseño del primer incremento de los sistemas de Monitoreo y Agente Inteligente, donde se mostrarán los casos de uso reales, diagramas de colaboración y los respectivos diagramas de clases. Adicionalmente, se definirá el lenguaje de programación con el cual finalmente se implementarán los sistemas.

3.4.1 Lenguaje de Programación

El lenguaje de programación a utilizar será Java, el cual nos permitirá modelar cada elemento lógico del Agente Inteligente de una forma más natural. Además, para implementar la comunicación entre el Agente Inteligente y el microcontrolador, será necesario una comunicación serial con protocolo RS-232, para lo cual, este lenguaje provee de una API llamada JavaComm.

3.4.2 Definición de Casos de Uso Reales para el SiM

La Tabla 3-39 y la Tabla 3-40 se darán a conocer los casos de uso reales, los cuales harán referencia, a través de una codificación, a los diagramas de interfaces gráficas, mostrados desde la Figura 3-12 hasta la Figura 3-15.

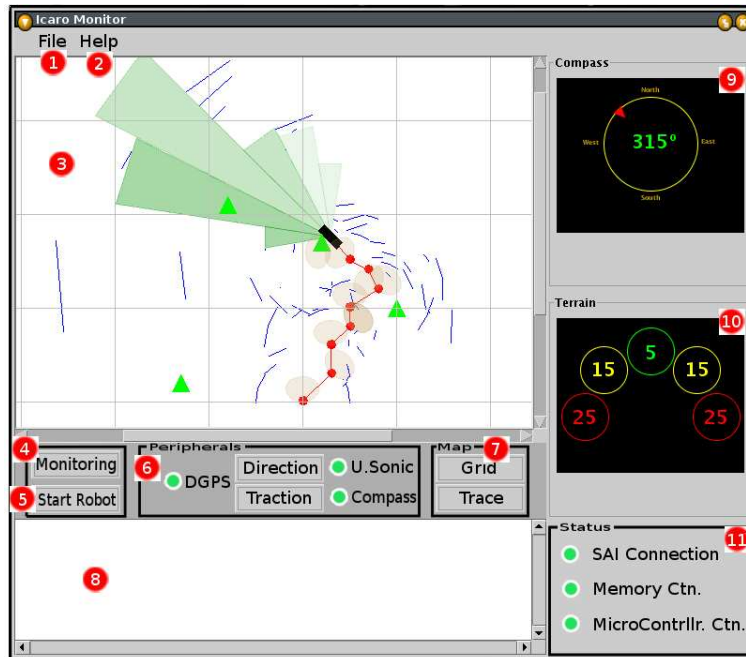


Figura 3-12. Sistema de Monitoreo sin Conexión.

Aquí se usará la abreviación SMSC (Sistema de Monitoreo sin Conexión), acompañada por una de las siguientes referencias para aludir a la Figura 3-12:

1. Opciones básicas del programa.
2. Ayuda para el usuario, sobre las operaciones posibles y soluciones a problemas comunes.
3. Área de representación de entorno del agente inteligente.
4. Botón para el inicio del monitoreo del agente inteligente.
5. Botón para dar comienzo a las acciones de la inteligencia del agente.
6. Área de botones de prueba de periféricos.
7. Botones para configuraciones visuales.
8. Área de texto para visualizar mensajes del agente inteligente.
9. Área de graficación del compás del agente inteligente.
10. Área de graficación de sensor de terreno del Agente Inteligente.
11. Área de prueba de las conexiones y visualización del estado actual de los sistemas complementarios del Agente Inteligente.

CAPITULO 3: DESARROLLO DEL PROYECTO

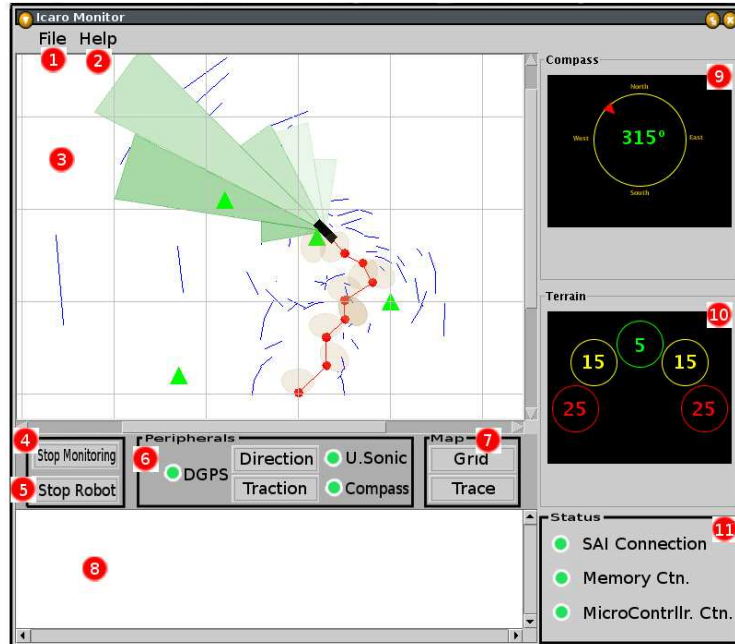


Figura 3-13. Sistema de Monitoreo con Conexión.

Aquí se usará la abreviación SMCC (Sistema de Monitoreo con Conexión), acompañada por uno de las siguientes referencias para aludir a la Figura 3-13:

1. Opciones básicas del programa.
2. Ayuda para el usuario, sobre las operaciones posibles y soluciones a problemas comunes.
3. Área de representación de entorno del agente inteligente.
4. Botón para la detención del monitoreo del agente inteligente.
5. Botón para detener las acciones de la inteligencia del agente.
6. Área de botones de prueba de periféricos.
7. Botones para configuraciones visuales.
8. Área de texto para visualizar mensajes del agente inteligente.
9. Área de graficación del compás del agente inteligente.
10. Área de graficación de sensor de terreno del agente inteligente.
11. Área de prueba de las conexiones y visualización del estado actual de los sistemas complementarios del agente inteligente.

CAPITULO 3: DESARROLLO DEL PROYECTO

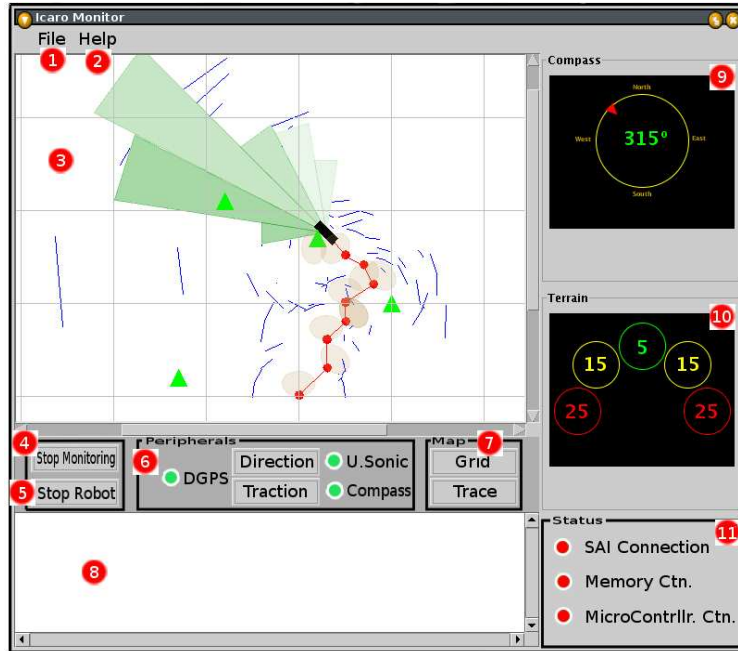


Figura 3-14. Sistema de Monitoreo con Error en Sistemas Complementarios.

Aquí se usará la abreviación *SMCE* (*Sistema de Monitoreo con Error en Sistemas Complementarios*), acompañada por uno de las siguientes referencias para aludir a la Figura 3-14:

1. Opciones básicas del programa.
2. Ayuda para el usuario, sobre las operaciones posibles y soluciones a problemas comunes.
3. Área de representación de entorno del agente inteligente.
4. Botón para la detención del monitoreo del agente inteligente.
5. Botón para detener las acciones de la Inteligencia del agente inteligente.
6. Área de botones de prueba de periféricos.
7. Botones para configuraciones visuales.
8. Área de texto para visualizar mensajes de la Inteligencia del agente inteligente.
9. Área de graficación del compás del agente inteligente.
10. Área de graficación de sensor de terreno del agente inteligente.
11. Área de prueba de las conexiones y visualización del estado actual de los *Sistemas Complementarios* del agente inteligente.

CAPITULO 3: DESARROLLO DEL PROYECTO

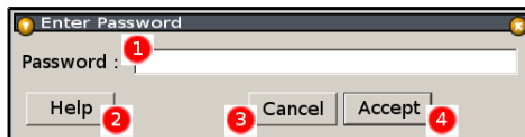


Figura 3-15. Diálogo de Ingreso del Password de Sistema de Monitoreo.

Aquí se usará la abreviación *DIPSM* (*Diálogo de Ingreso del Password de Sistema de Monitoreo*), acompañado por una de las siguientes referencias para aludir a la Figura 3-15:

1. Campo de texto para el ingreso del password.
2. Botón con ayuda del contexto.
3. Botón para cancelar la acción de ingreso del password.
4. Botón para permitir que el sistema de monitoreo capture el password dado en el campo de texto.

<i>Caso de Uso:</i>	<i>Monitorear SAI.</i>
<i>Actores:</i>	<i>Usuario Monitor ó Administrador de Agente Inteligente.</i>
<i>Propósito:</i>	<i>El seguimiento y análisis de las distintas mediciones sensoriales y variables internas del Agente inteligente.</i>
<i>Resumen:</i>	<i>El Usuario Monitor o el Administrador de Agente Inteligente piden al SiM monitorear al SAI, solicitando éste datos sobre su entorno, su dirección, ubicación geográfica, decisión de ruta a seguir. Todo lo anterior, con el fin de realizar el respectivo despliegue de información gráfica para el Usuario Monitor o el Administrador de Agente Inteligente.</i>
<i>Tipo:</i>	<i>Primario y Esencial.</i>
<i>Referencia Cruzada:</i>	<i>R 5.1, R 5.2, 5.3, R 7</i>
<i>Curso Normal de Eventos:</i>	
<i>Actor</i>	<i>Sistema</i>
<i>1.- Este caso de uso comienza cuando el Usuario Monitor o el Administrador de Agente Inteligente da la orden al SiM de monitorear al SAI, presionando el botón « Monitoring » (SMSC 4).</i>	
	<i>2a.- El SiM muestra en las áreas de representación de datos de monitoreo (SMCC 3,8,9,10) la información de monitoreo.</i>
<i>Curso Alternativo de Eventos:</i>	
	<i>2b.- El SiM rechaza la orden, por que el Administrador de Agente Inteligente no ha ejecutado el protocolo de inicio.</i>

Tabla 3-39. Caso de uso “Monitorear SAI”.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Iniciar SiM.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>
<i>Propósito:</i>	<i>Identificar al Usuario del Sistema Monitor, y así dar las opciones correspondientes.</i>
<i>Resumen:</i>	<i>El Usuario Monitor o el Administrador de Agente Inteligente inicia el software del Sistema de Monitoreo, y procede a autenticarse para dar inicio, y proveerle de las opciones correspondientes.</i>
<i>Tipo:</i>	<i>Primario y Esencial.</i>
<i>Referencia Cruzada:</i>	<i>R 8</i>
<i>Curso Normal de Eventos:</i>	
<i>Actor</i>	<i>Sistema</i>
<i>1.- Este caso de uso comienza con la autenticación del usuario, proveyendo éste de un password en el Campo de texto para el ingreso de password (DIPSM 1) en el Diálogo de Ingreso de Password del Sistema de Monitoreo.</i>	
	<i>2.- El SiM verifica si la clave esta registrada.</i>
	<i>3a.- Una vez reconocido el password de administrador, el SiM habilita todas las opciones para las pruebas y monitoreo del Agente Inteligente. (SMCC 3, 4, 5, 6, 7, 8, 9, 10, 11)</i>
<i>Curso Alternativo de Eventos:</i>	
	<i>3b.- Una vez reconocido el password de Usuario Monitor, el SiM habilita sólo la opción de monitoreo. (SMCC 3, 4, 8, 9, 10, 11)</i>
	<i>3c.- Una vez verificado que el password ingresado no corresponde a un Usuario Monitor ni al Administrador de Agente Inteligente, el SiM no habilita opción alguna de monitoreo.</i>

Tabla 3-40. Caso de uso “Iniciar SiM”.

3.4.3 Diagrama de Clase del SiM

Los paquetes, desde la Figura 3-17 hasta la Figura 3-20, que a continuación se presentan, corresponden a la división de la arquitectura multicapa orientada a objeto, en la cual cada paquete representado tomará el nombre de las capas de “gráfica” y “lógica” de este modelo. En la Figura 3-16, se grafica el diagrama general de paquetes.

Del modelo conceptual presentado en la Figura 3-5, se desprenden las clases de los paquetes “grafica” y “logica”, dada la necesidad inherente del “Observador” de monitorear al “Robot”. Para tal efecto, en el paquete “grafica” existen clases que permitirán mostrar en tiempo real las distintas percepciones, acciones y decisiones del robot. Además, en el paquete “logica” existe una clase controladora que realizará el nexo entre los dos paquetes, otra que permitirá comunicación a través de sockets bajo el modelo Cliente-Servidor y una interfaz para esta última, las que permitirán obtener la información del robot de forma remota.

3.4.3.1 Diagrama General de Paquetes

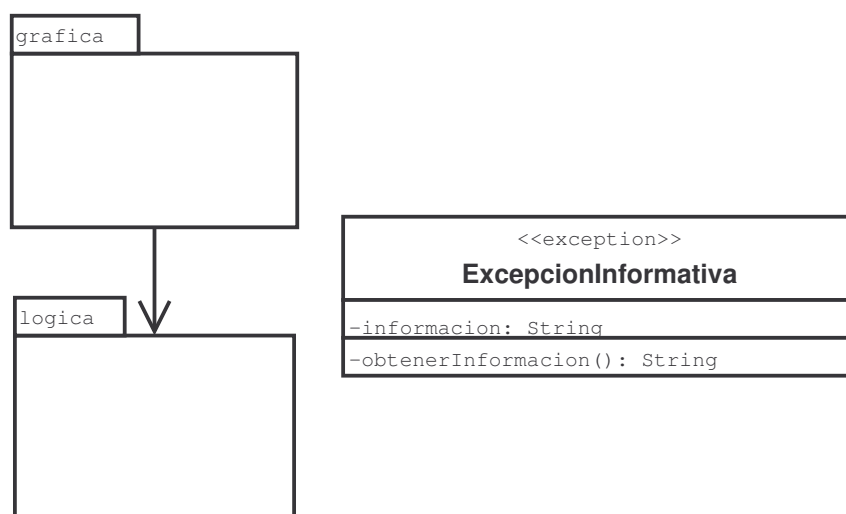


Figura 3-16. Diagrama general de paquetes.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.4.3.2 Diagrama de Clase (Paquete grafica)

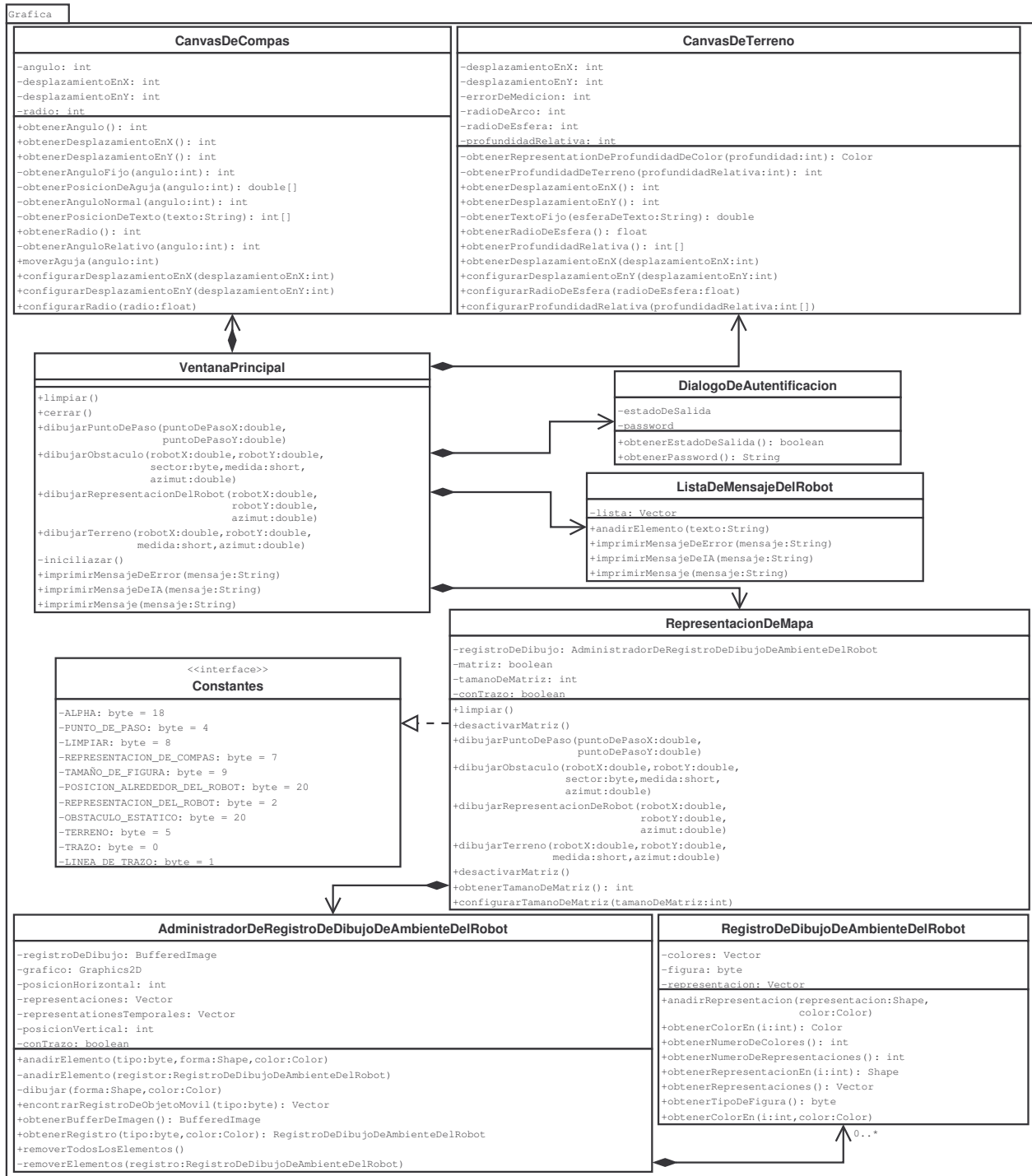


Figura 3-17. Diagrama de Clases, paquete “grafica”.

CAPITULO 3: DESARROLLO DEL PROYECTO

En este punto es necesario realizar algunos comentarios relacionados con la Figura 3-17:

1. Para este diagrama se omitieron las relaciones y métodos concernientes a los componentes gráficos provistos por el lenguaje Orientado a Objetos ocupado, por ser poco relevantes de explicar en relación a otros puntos que el proyecto pretende lograr.
2. La clase *VentanaPrincipal* estará compuesta de 3 canvases y un área de texto, los cuales representan la graficación de los distintos sensores además de la narración en texto del robot a través del tiempo, respectivamente. Adicionalmente, esta clase proporciona tres áreas de interacción, en las cuales se encuentran botones para realizar operaciones relacionadas con la conectividad y pruebas de los distintos subsistemas, sensores y efectores del sistema remoto SAI.

3.4.3.3 Diagrama de Clase (Paquete logica)

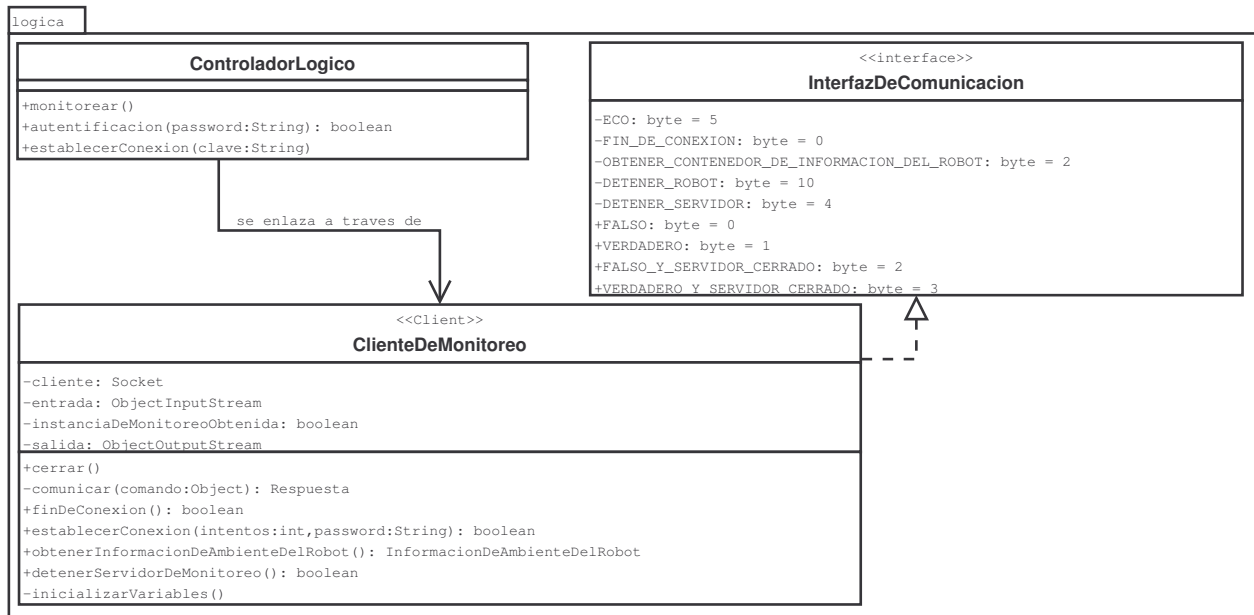


Figura 3-18. Diagrama de Clases, paquete “logica”.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.4.3.4 Bean

Referente a estas clases, podemos decir que su objetivo es operar a través de las diferentes capas de la arquitectura del software, transportando información de forma encapsulada.

En la Figura 3-19 se muestra el Bean definido para el problema.

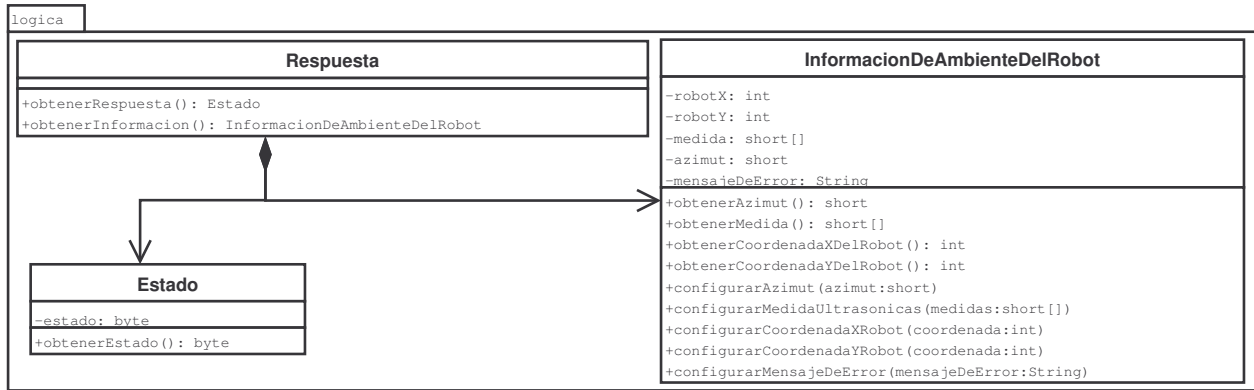


Figura 3-19. Clases Bean del problema.

3.4.3.5 Excepción

Referente a estas clases, podemos decir que su objetivo es operar a través de las diferentes capas de la arquitectura del software, transportando información de forma encapsulada sobre algún evento crítico ocurrido.

En la Figura 3-20 se muestra la clase excepción definida para el dominio del problema.

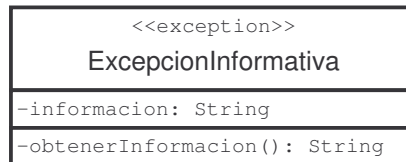


Figura 3-20. Clase de excepción del problema.

3.4.4 Diagramas de Colaboración del SiM

En la Figura 3-21 y Figura 3-22 se dan a conocer los diagramas de colaboración para el primer incremento del Sistema de Monitoreo. Estos diagramas tienen relación con los diagramas de secuencia de sistema mostrados en la sección 3.3.2.

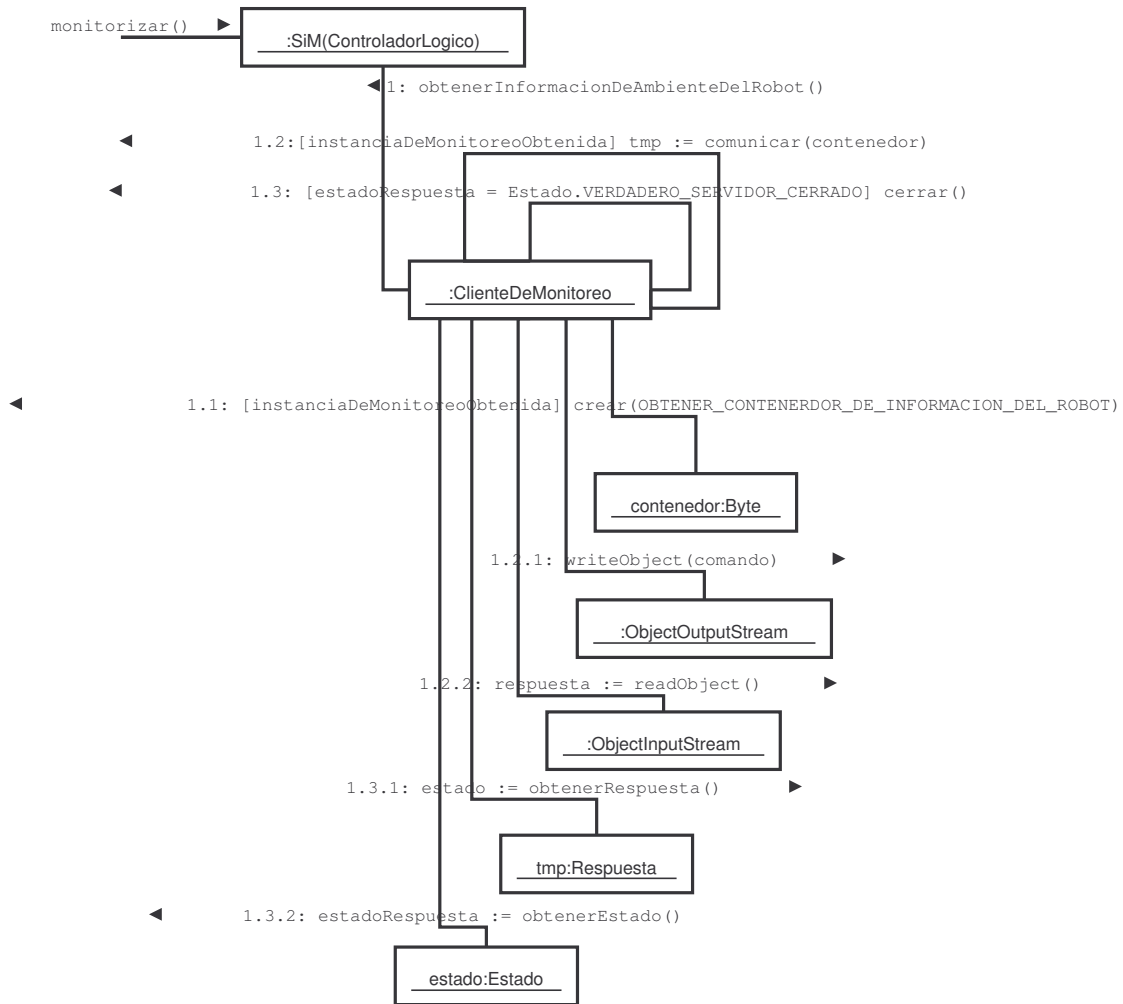


Figura 3-21. Diagrama de Colaboración monitorizar.

CAPITULO 3: DESARROLLO DEL PROYECTO

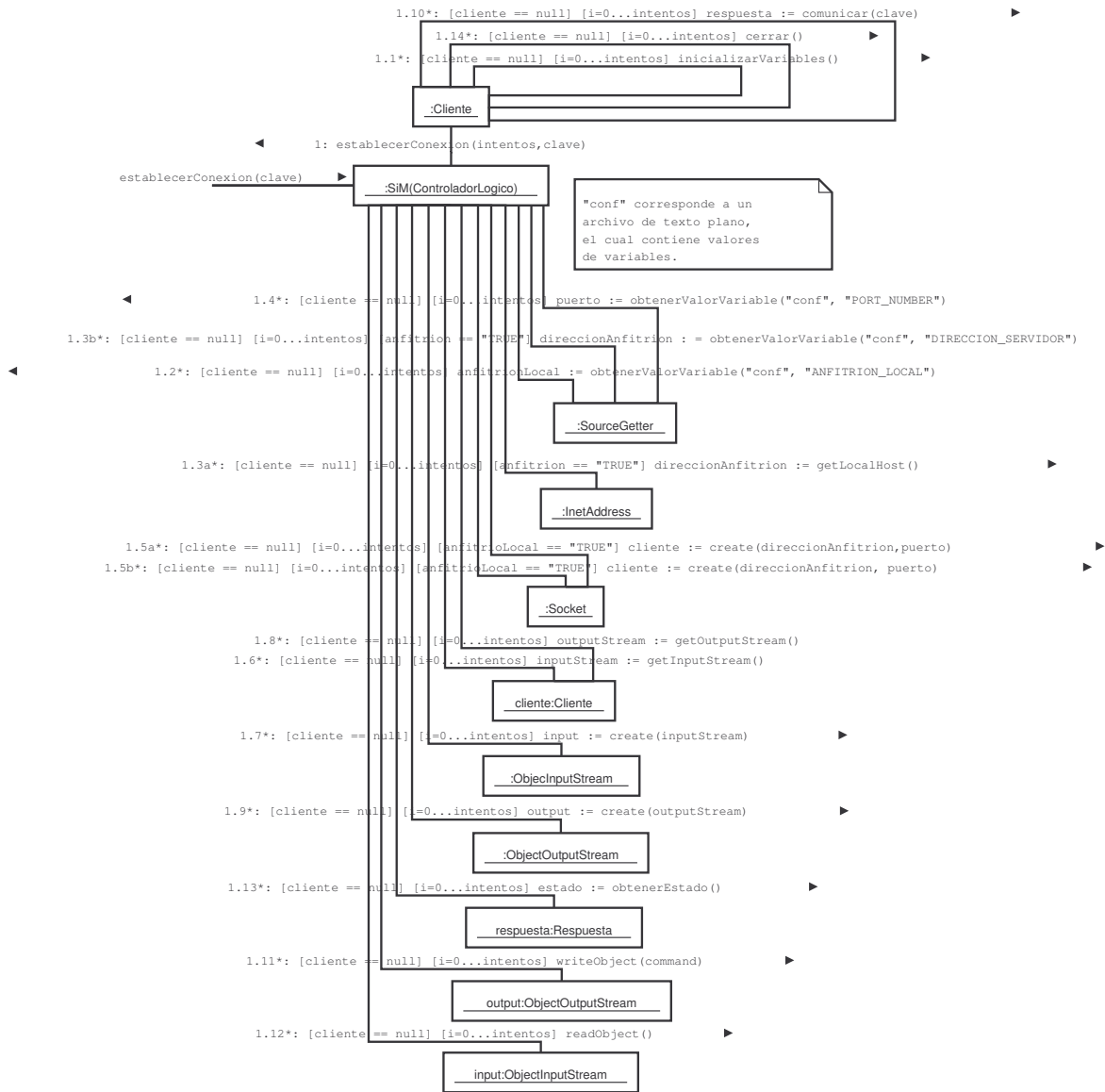


Figura 3-22. Diagrama de Colaboración autenticar.

3.4.5 Diagrama de Clase del SAI

Los paquetes, desde la Figura 3-24 hasta la Figura 3-28, que a continuación se presentan, corresponden a la división de la arquitectura multicapa orientada a objeto, en la cual cada paquete representado pertenecerá solamente a la capa Lógica de este modelo. En la Figura 3-23, se grafica el diagrama general de paquetes.

Del modelo conceptual presentado en la Figura 3-6, se desprenden las clases de los paquetes “comunicacion”, “logica” y “dispositivos”. Dada la necesidad del “Observador” (SiM) de monitorear remotamente, es que se deberá implementar un servidor, el cual residirá dentro del paquete “comunicacion” que proveerá la información del “Robot”. Por lo anterior, ya que el “Robot” deberá proveer la información capturada de su ambiente es que deberá implementar el paquete “logica”, en el cual residirá la representación lógica de los dispositivos físicos, lo que permitirá capturar datos del ambiente y manipular los efectores. Para lograr la comunicación entre las representaciones lógicas y los dispositivos físicos, se deberá implementar una clase que pertenecerá al paquete “dispositivos”, ésta se comunicará bajo un protocolo definido más adelante en la sección 4.4.

3.4.5.1 Diagrama General de Paquetes

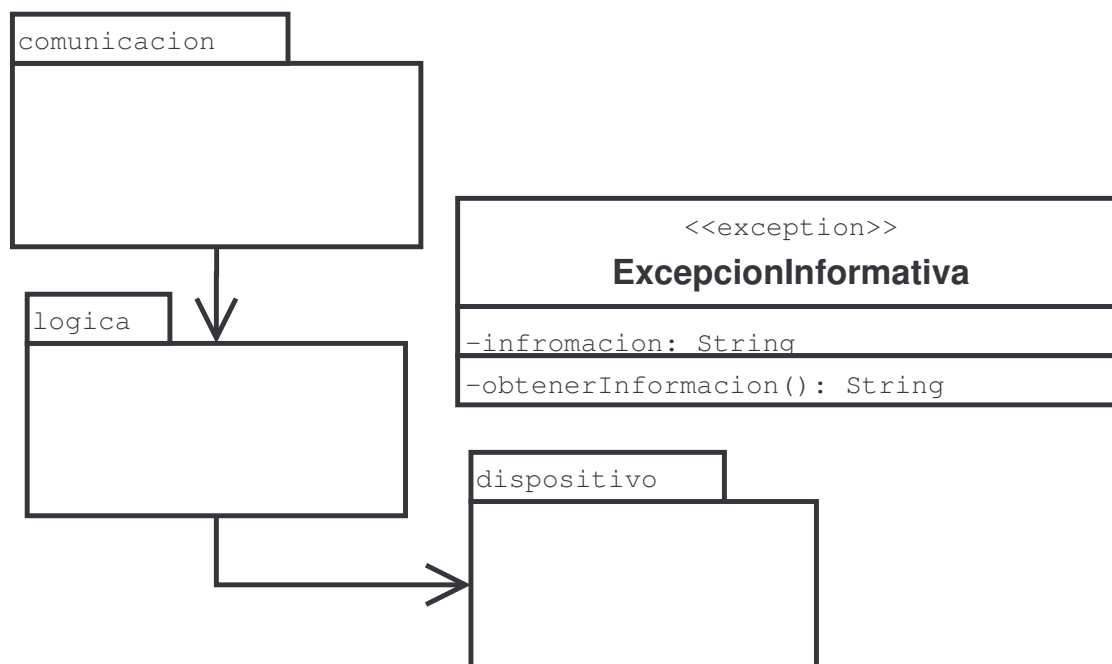


Figura 3-23. Diagrama general de paquetes.

3.4.5.2 Diagrama de Clase (Paquete comunicacion)

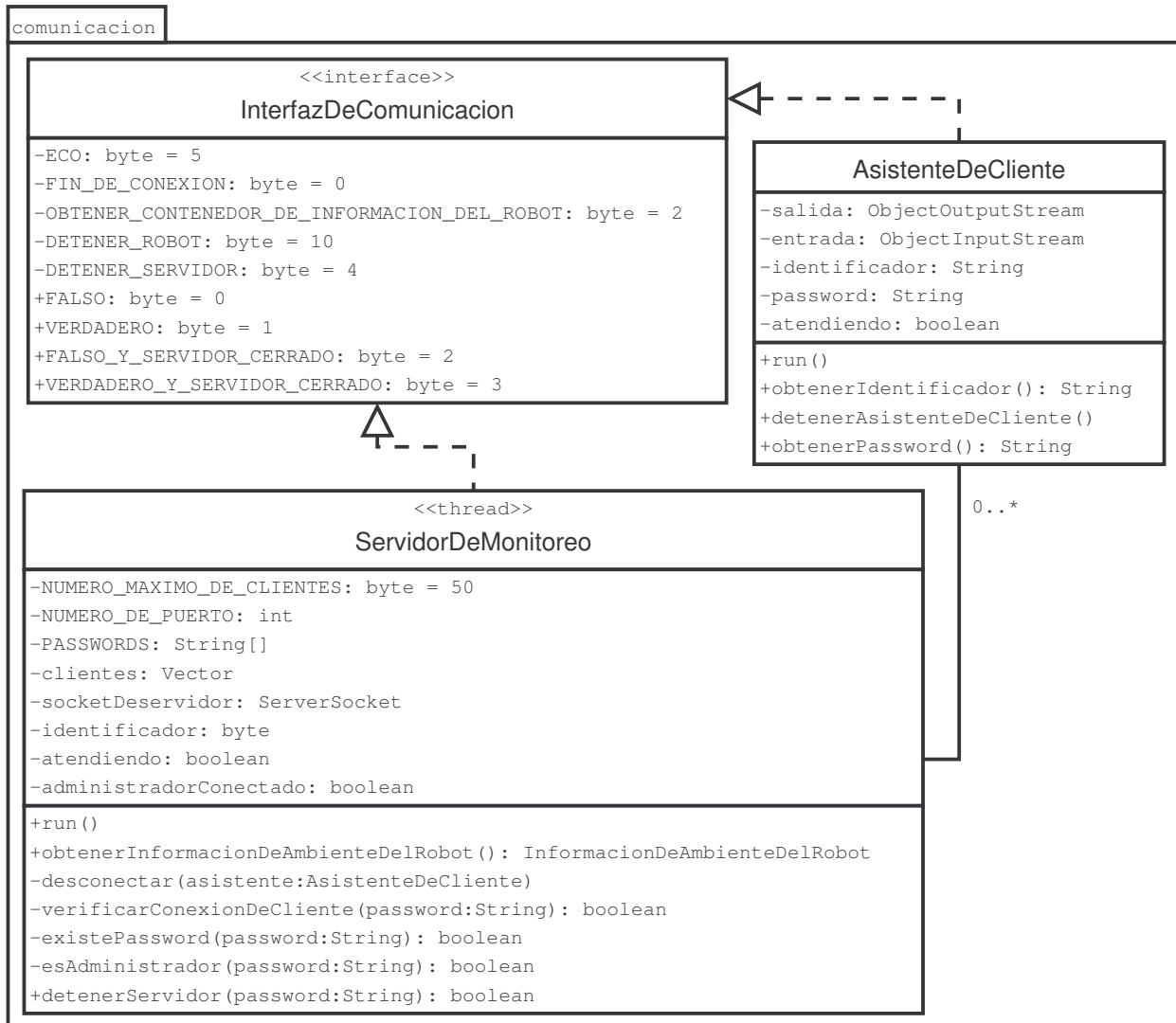


Figura 3-24. Diagrama de clases, paquete “comunicacion”.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.4.5.3 Diagrama de Clase (Paquete logica)

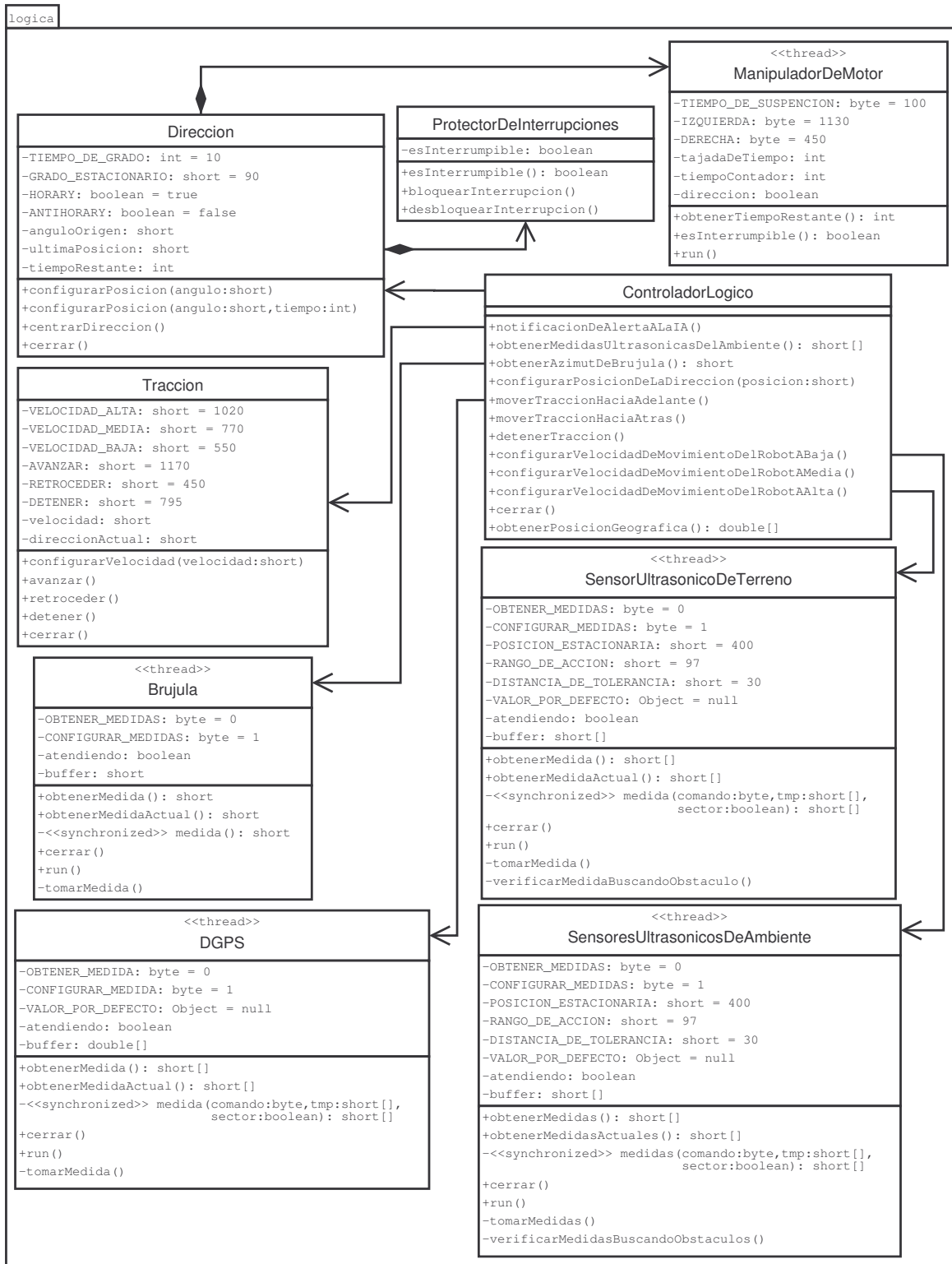


Figura 3-25. Diagrama de clases, paquete “logica”.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.4.5.4 Diagrama de Clase (Paquete dispositivo)



Figura 3-26. Diagrama de clases, paquete “dispositivo”.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.4.5.5 Bean

Referente a estas clases, se puede decir que su objetivo es operar a través de las diferentes capas de la arquitectura del software, transportando información de forma encapsulada.

En la Figura 3-27 se muestra el Bean definido para el problema.

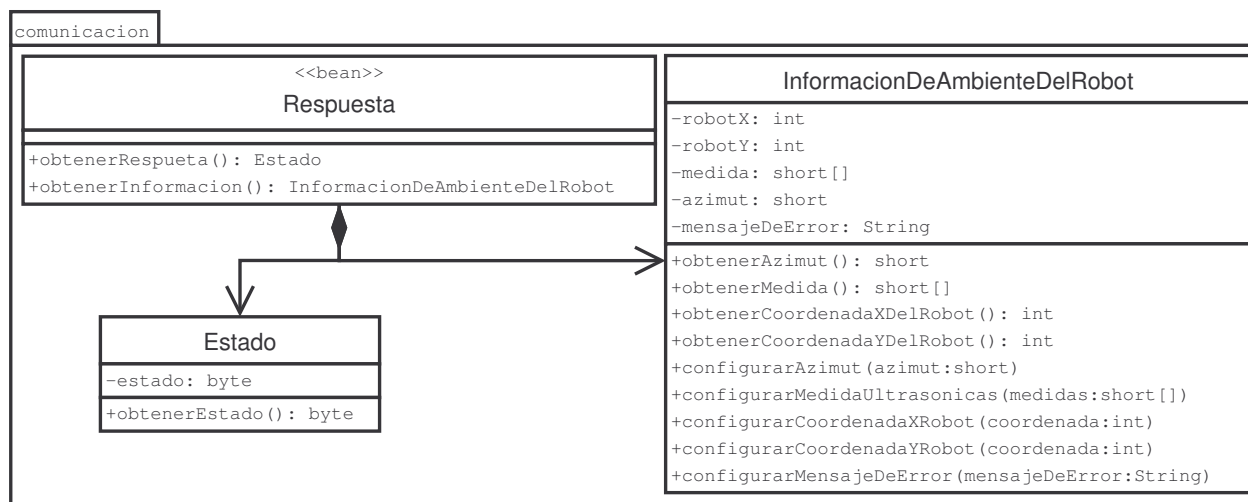


Figura 3-27. Clases Bean del problema.

3.4.5.6 Excepción

Referente a estas clases, podemos decir que su objetivo es operar a través de las diferentes capas de la arquitectura del software, transportando información de forma encapsulada sobre algún evento crítico ocurrido.

En la Figura 3-28 se muestra la clase excepción definidos para el dominio del problema.

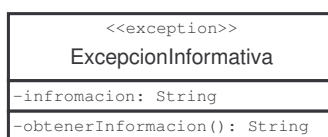


Figura 3-28. Clase de excepción del problema.

3.4.6 Diagramas de Colaboración del SAI

En la Figura 3-29 y Figura 3-30 se dan a conocer los diagramas de colaboración para el primer incremento del Sistema de Agente Inteligente.

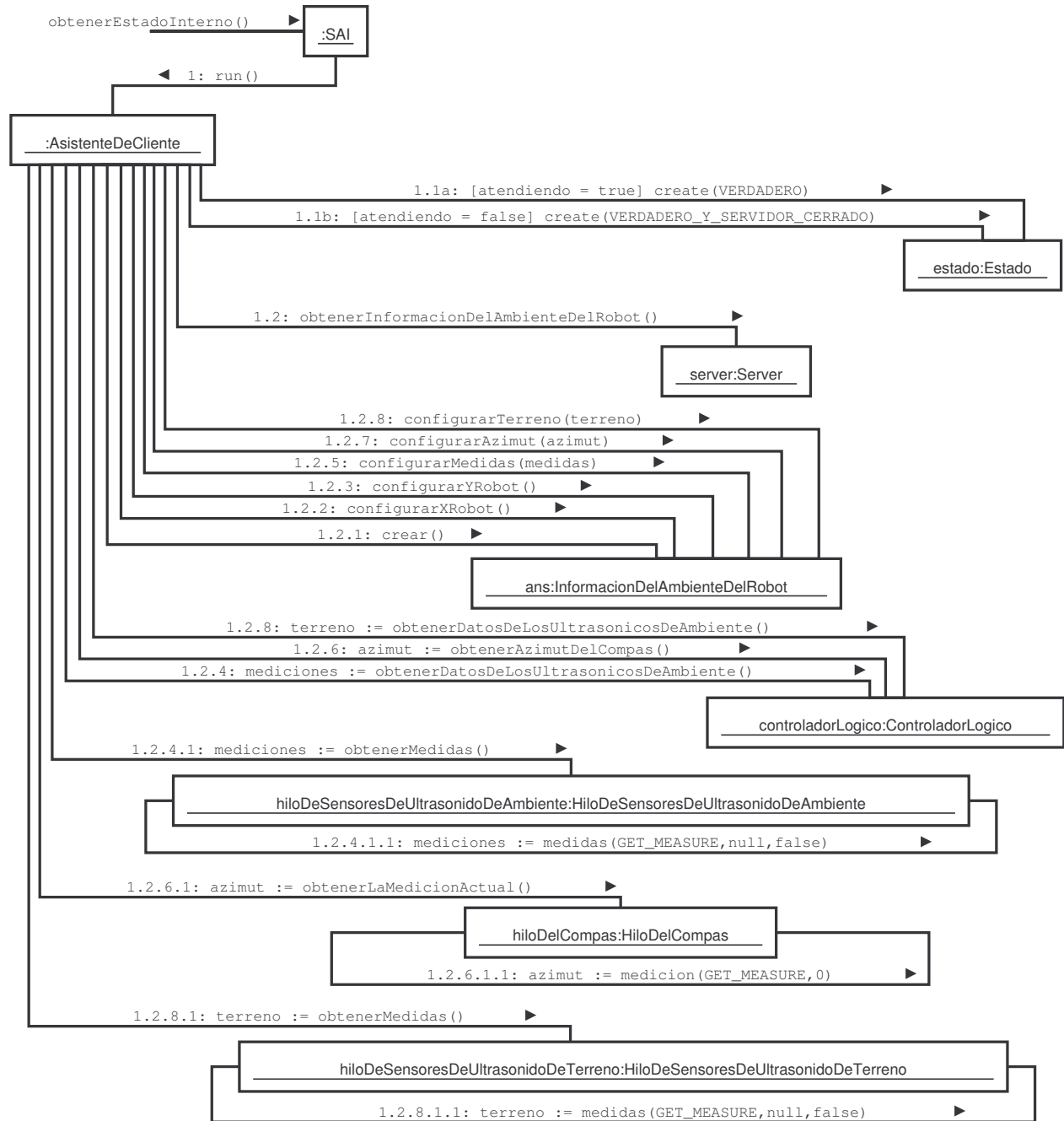


Figura 3-29. Diagrama de Colaboración obtener estado interno.

CAPITULO 3: DESARROLLO DEL PROYECTO

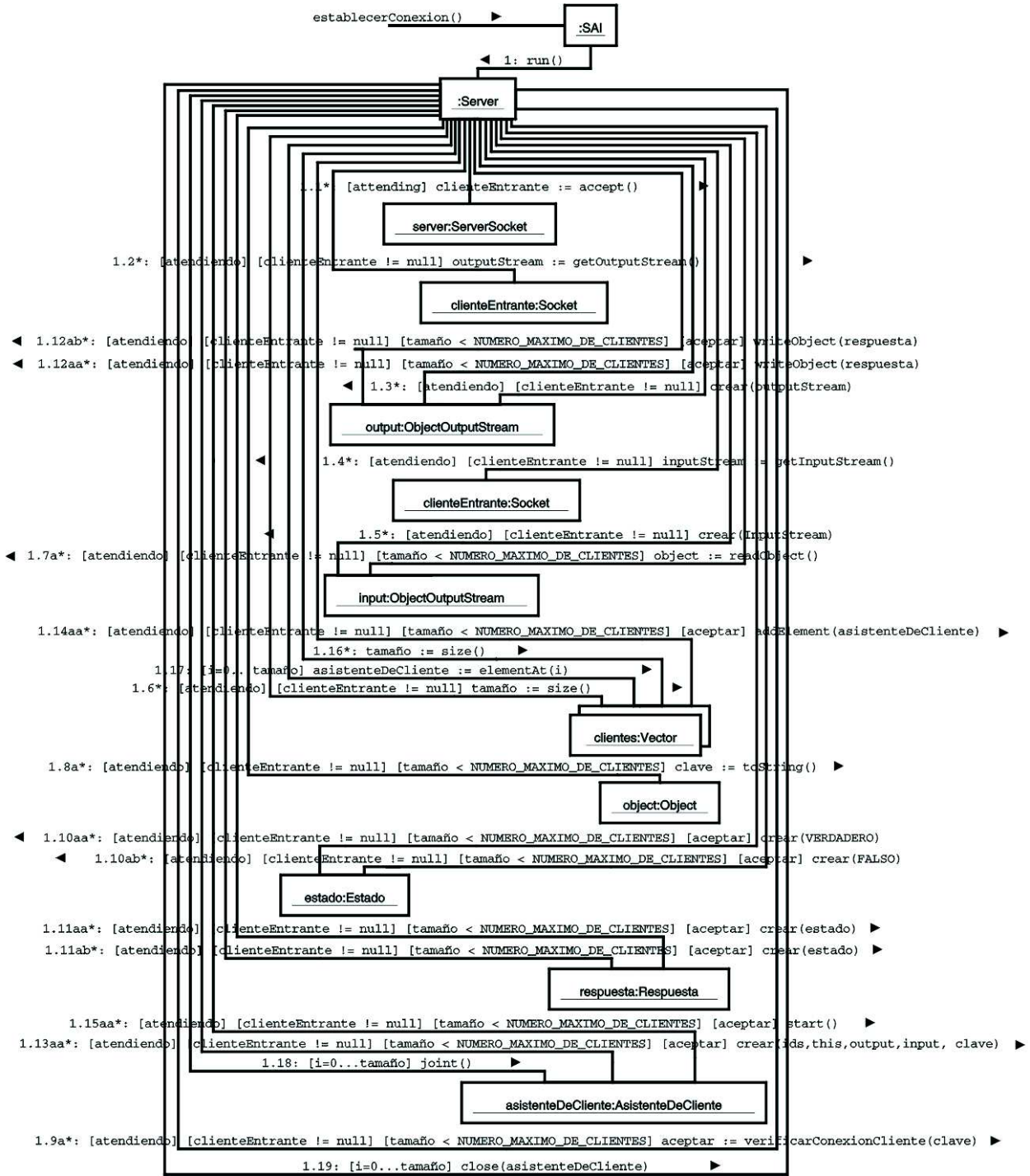


Figura 3-30. Diagrama de Colaboración establecer conexión.

3.5 Análisis Orientado a Objeto: Segundo Incremento

En la presente sección, se darán a conocer aspectos del análisis correspondientes al segundo incremento de los sistemas de Monitoreo y Agente Inteligente, donde se encontrarán los respectivos modelos conceptuales y diagramas de secuencia.

3.5.1 Modelo Conceptual

A continuación se presentarán los modelos conceptuales de los sistemas, en la Figura 3-31 y Figura 3-32, tomado conceptos del dominio del problema descrito en la sección 1.2.

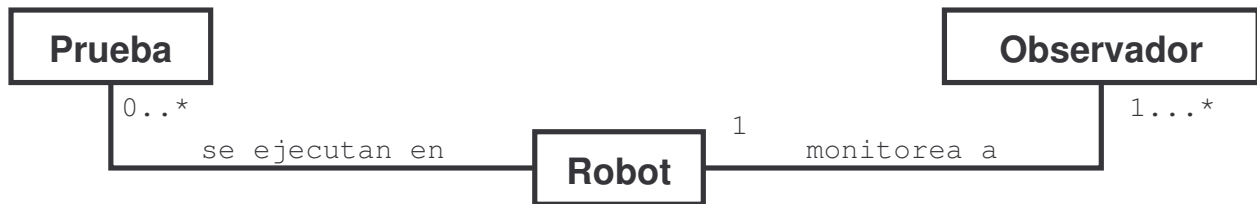


Figura 3-31. Diagrama de Modelo Conceptual SiM.

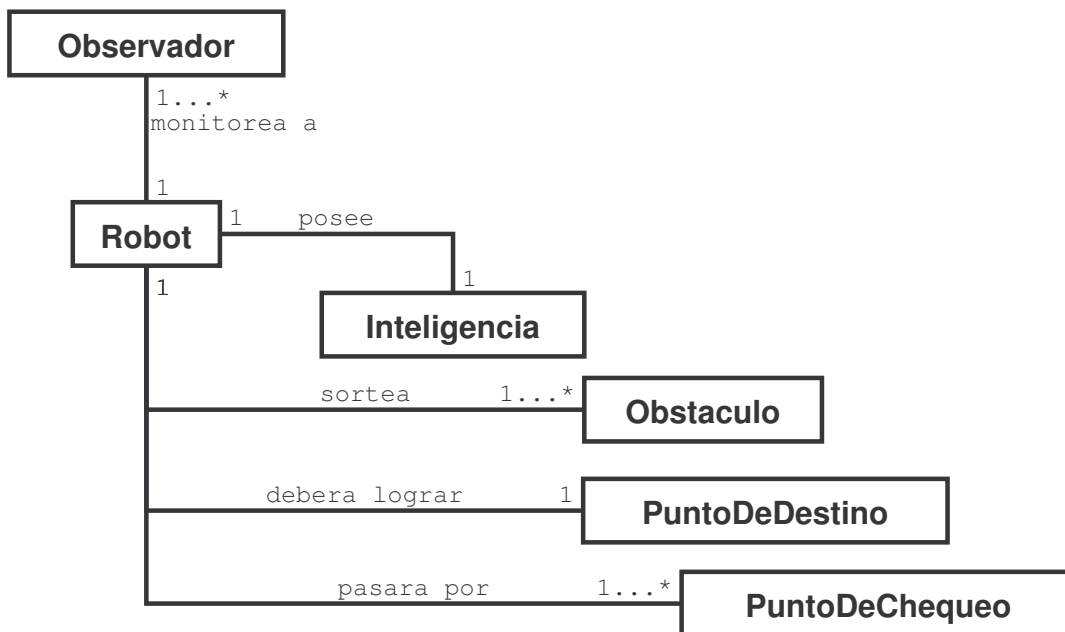


Figura 3-32. Diagrama de Modelo Conceptual SAI.

3.5.2 Diagramas de Secuencia de SiM

Desde la Figura 3-33 hasta la Figura 3-39 se darán a conocer los diagramas de secuencia correspondientes al segundo incremento para el sistema SiM. Estos diagramas están relacionados con los casos de uso mostrados desde la Tabla 3-30 hasta la Tabla 3-36.

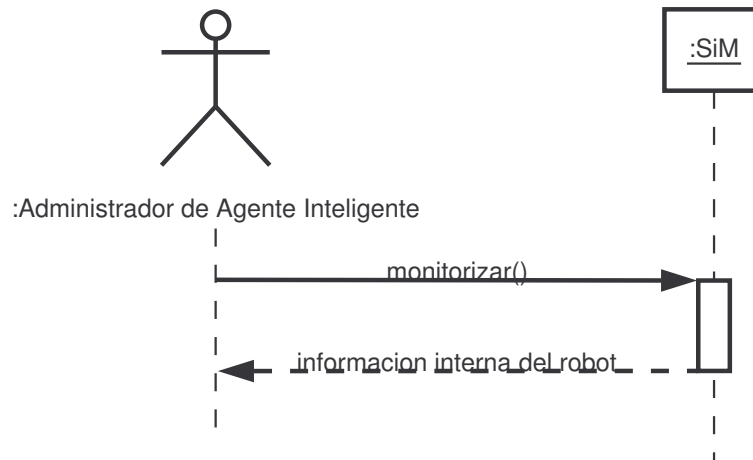


Figura 3-33. Diagrama Monitorear SAI.

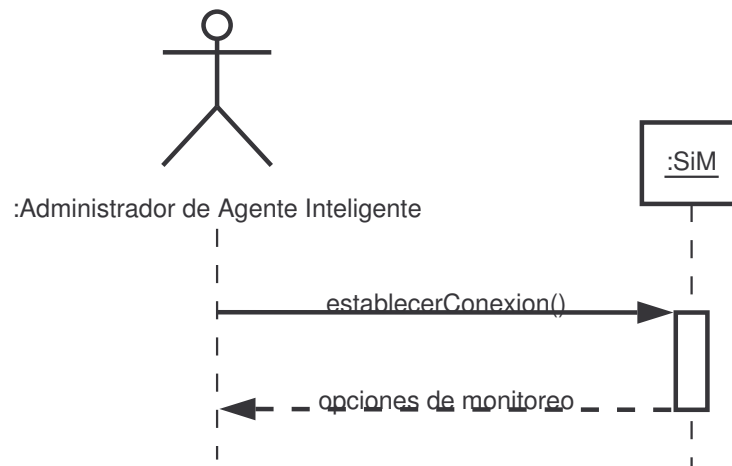


Figura 3-34. Diagrama Iniciar SiM.

CAPITULO 3: DESARROLLO DEL PROYECTO

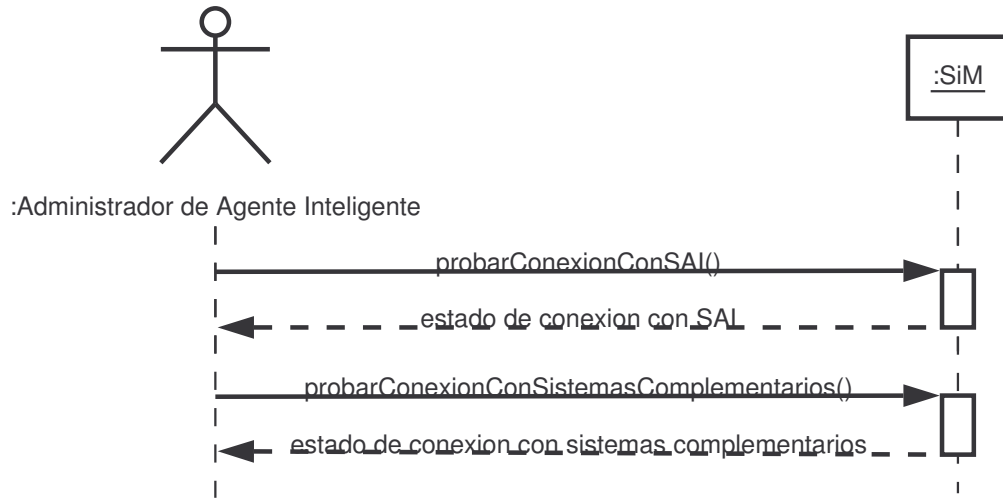


Figura 3-35. Diagrama probar conexión con sistemas complementarios.

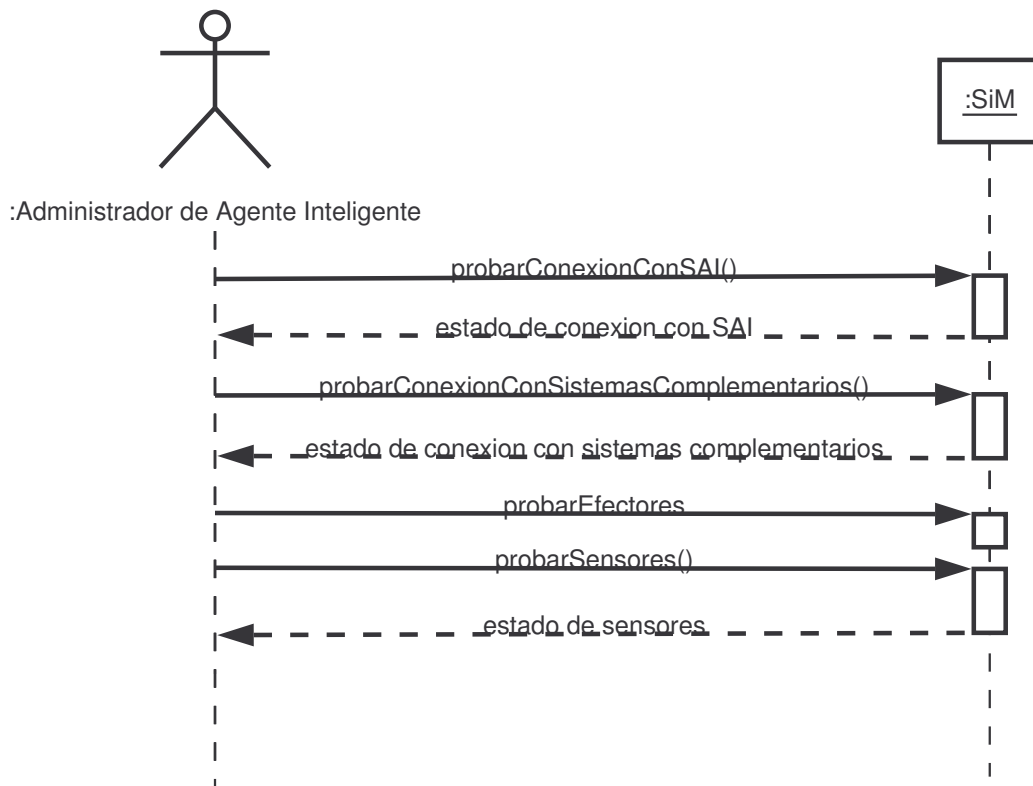


Figura 3-36. Diagrama Ejecutar protocolo de inicio.

CAPITULO 3: DESARROLLO DEL PROYECTO

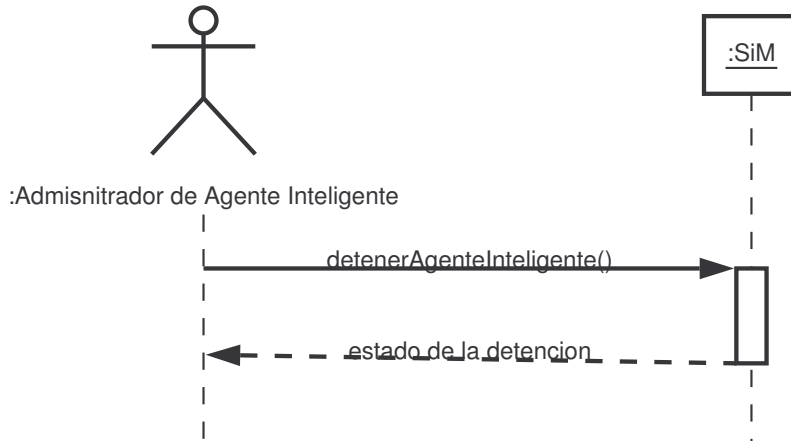


Figura 3-37. Diagrama Detener agente inteligente.

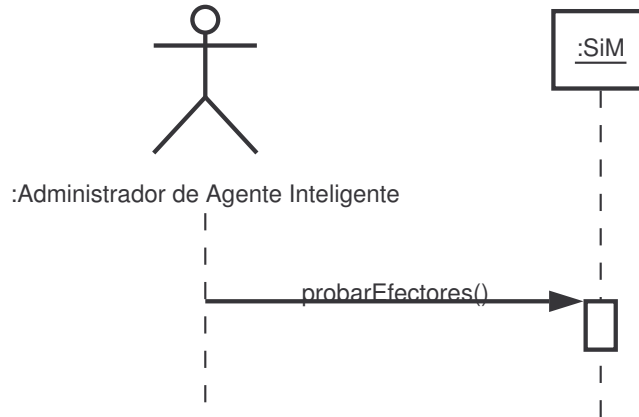


Figura 3-38. Diagrama Probar efectores del SAI.

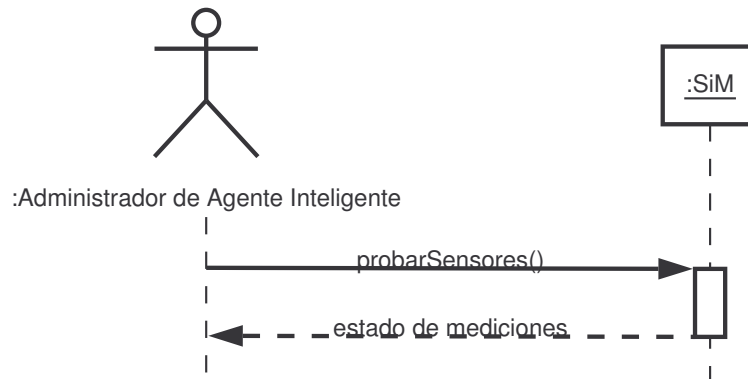


Figura 3-39. Diagrama Probar sensores del SAI.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.5.3 Diagramas de Secuencia de SAI

Desde la Figura 3-40 hasta la Figura 3-48 se darán a conocer los diagramas de secuencia correspondientes al segundo incremento para el sistema SAI. Estos diagramas están relacionados con los casos de uso mostrados desde la Tabla 3-21 hasta la Tabla 3-28.

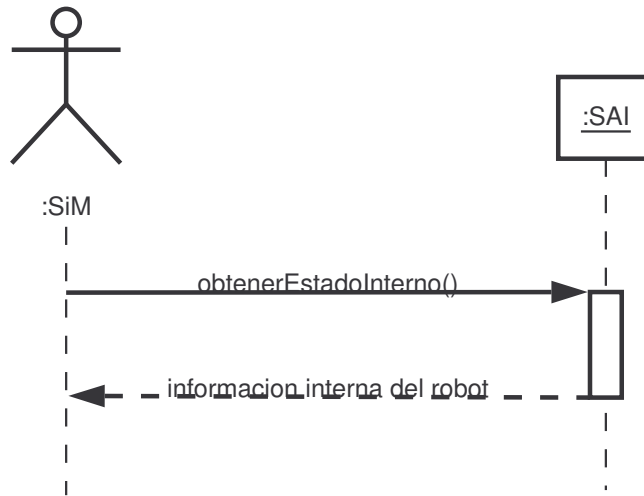


Figura 3-40. Diagrama Consultar estado interno.

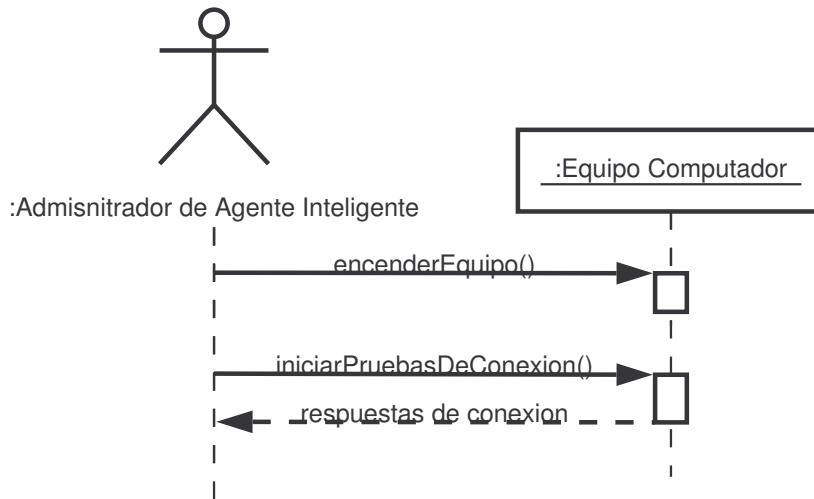


Figura 3-41. Diagrama Iniciar SAI.

CAPITULO 3: DESARROLLO DEL PROYECTO

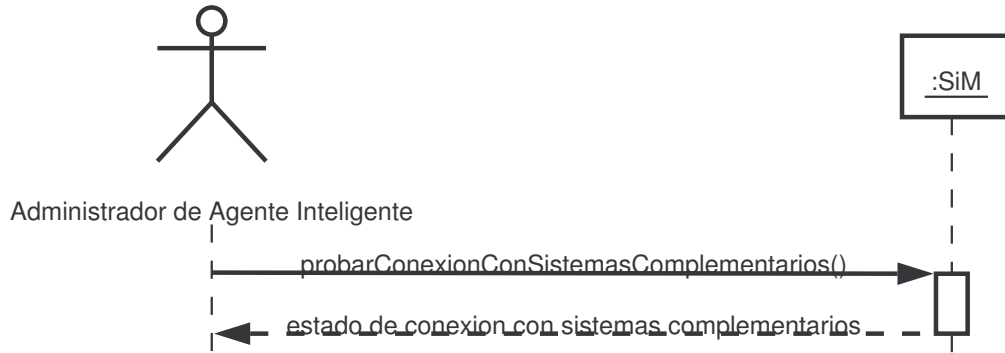


Figura 3-42. Diagrama probar conexión con sistemas complementarios.

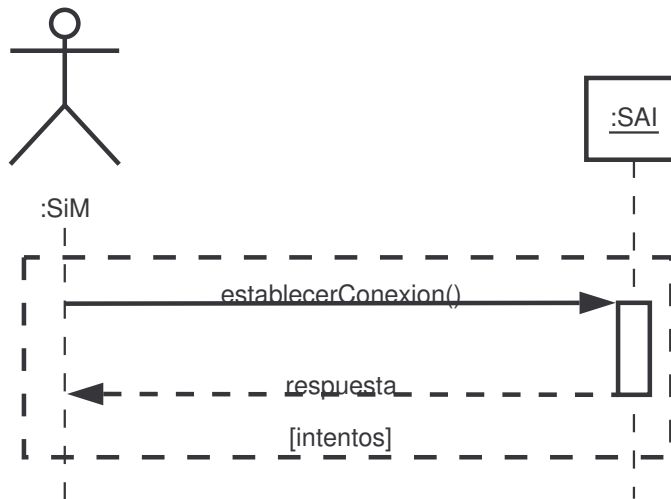


Figura 3-43. Diagrama Conectar con SAI.

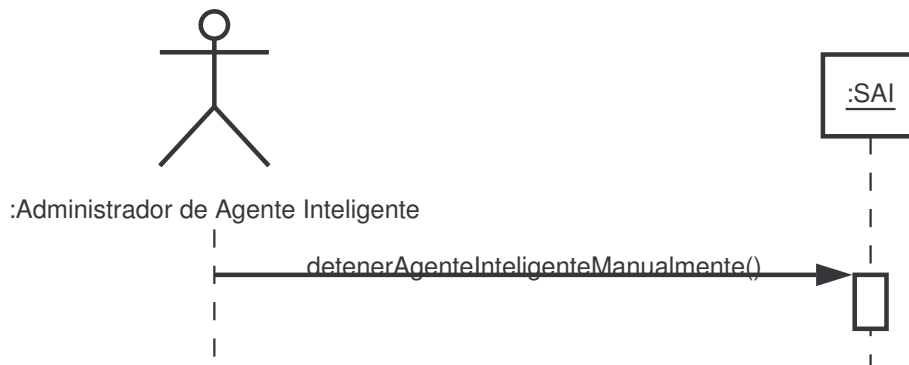


Figura 3-44. Diagrama detener agente inteligente manual.

CAPITULO 3: DESARROLLO DEL PROYECTO

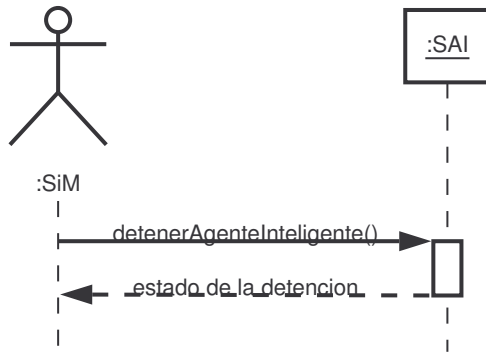


Figura 3-45. Diagrama detener agente inteligente.

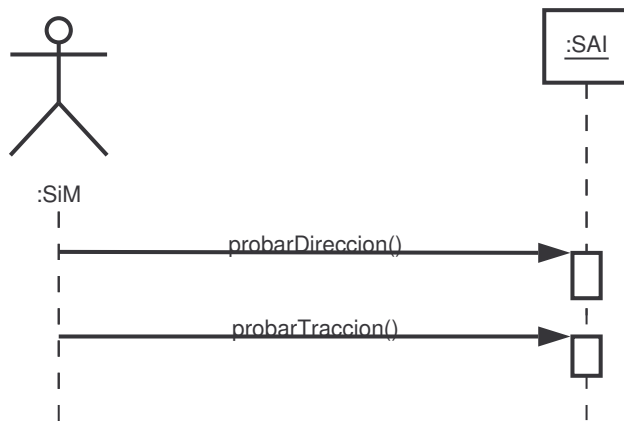


Figura 3-46. Diagrama probar efectores del SAI.

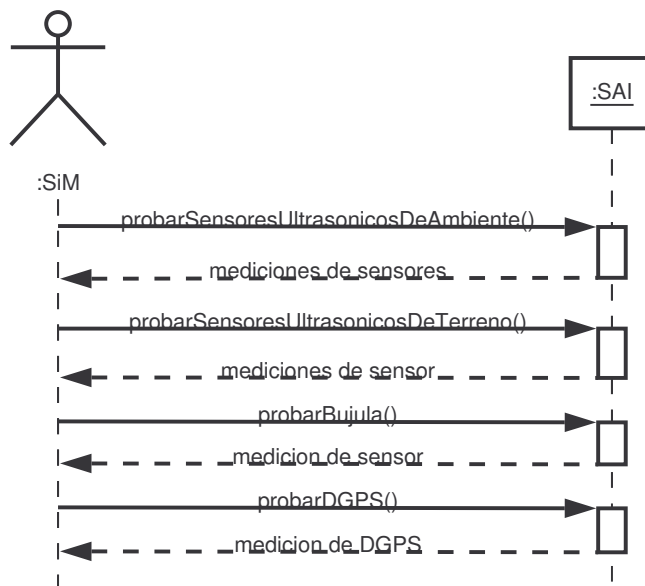


Figura 3-47. Diagrama probar sensores del SAI.

CAPITULO 3: DESARROLLO DEL PROYECTO

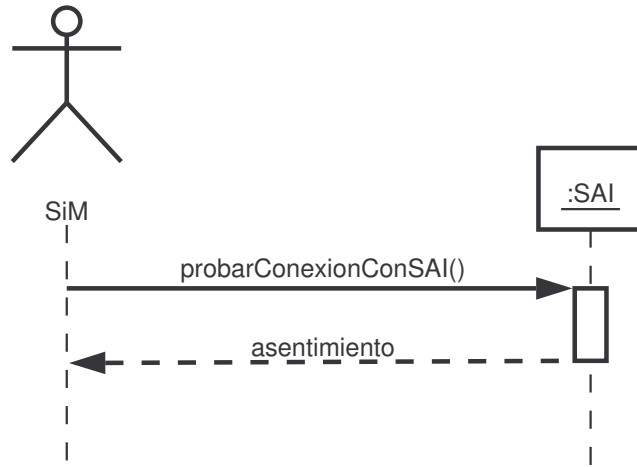


Figura 3-48. Diagrama probar conexión con SAI.

3.6 Diseño Orientado a Objeto: Segundo Incremento

En la presente sección, se darán a conocer aspectos concernientes al diseño del segundo incremento de los sistemas de Monitoreo y Agente Inteligente, donde se mostrarán los casos reales de uso, diagramas de colaboración y los respectivos diagramas de clases para cada uno.

3.6.1 Definición de Casos de Uso Reales para SiM

Desde la Tabla 3-41 hasta la Tabla 3-44 se darán a conocer los casos de uso reales, los cuales harán referencia, a través de una codificación, a los diagramas de interfaces gráficas, mostrados desde la Figura 3-49 hasta la Figura 3-52.

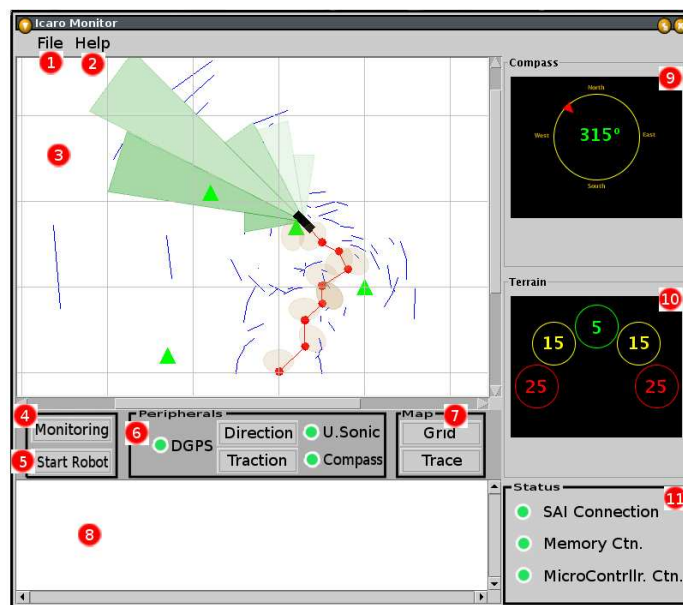


Figura 3-49. Sistema de Monitoreo sin Conexión.

Aquí se usará la abreviación SMSC (Sistema de Monitoreo sin Conexión), acompañada por una de las siguientes referencias para aludir a la Figura 3-49:

1. Opciones básicas del programa.
2. Ayuda para el usuario, sobre las operaciones posibles y soluciones a problemas comunes.
3. Área de representación de entorno del agente inteligente.
4. Botón para el inicio del monitoreo del agente inteligente.
5. Botón para dar comienzo a las acciones de la inteligencia del agente inteligente.
6. Área de botones de prueba de periféricos.
7. Botones para configuraciones visuales.
8. Área de texto para visualizar mensajes del agente inteligente.
9. Área de graficación del compás del agente inteligente.
10. Área de graficación de sensor de terreno del agente inteligente.
11. Área de prueba de las conexiones y visualización del estado actual de los sistemas complementarios del agente inteligente.

CAPITULO 3: DESARROLLO DEL PROYECTO

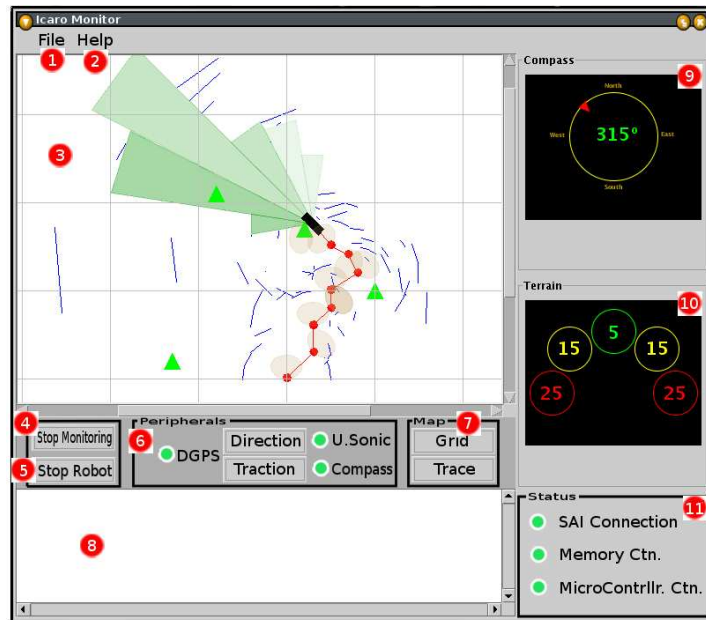


Figura 3-50. Sistema de Monitoreo con Conexión.

Aquí se usará la abreviación SMCC (Sistema de Monitoreo con Conexión), acompañada por uno de las siguientes referencias para aludir a la Figura 3-50:

1. Opciones básicas del programa.
2. Ayuda para el usuario, sobre las operaciones posibles y soluciones a problemas comunes.
3. Área de representación de entorno del agente inteligente.
4. Botón para la detención del monitoreo del agente inteligente.
5. Botón para detener las acciones de la Inteligencia del agente inteligente.
6. Área de botones de prueba de periféricos
7. Botones para configuraciones visuales.
8. Área de texto para visualizar mensajes del agente inteligente.
9. Área de graficación del compás del agente inteligente.
10. Área de graficación de sensor de terreno del agente inteligente.
11. Área de prueba de las conexiones y visualización del estado actual de los sistemas complementarios del agente inteligente.

CAPITULO 3: DESARROLLO DEL PROYECTO

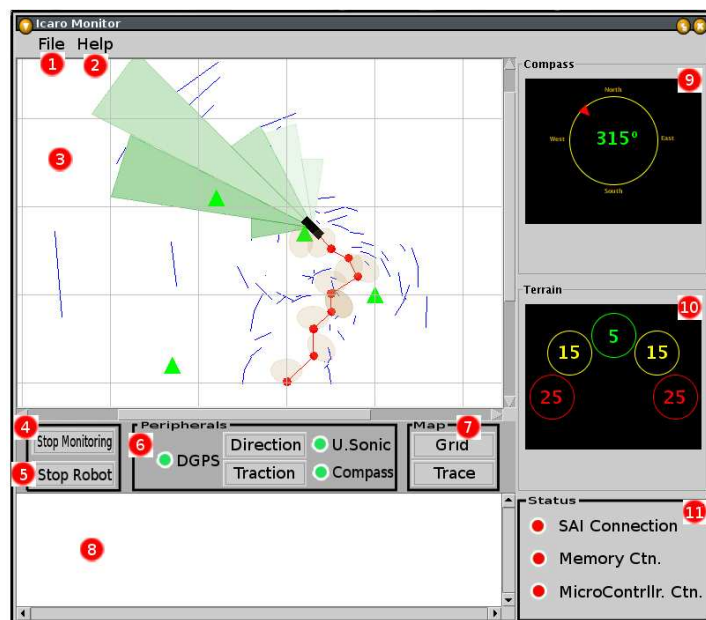


Figura 3-51. Sistema de Monitoreo con Error en Sistemas Complementarios.

Aquí se usará la abreviación SMCE (Sistema de Monitoreo con Error en Sistemas Complementarios), acompañada por uno de las siguientes referencias para aludir a la Figura 3-51:

1. Opciones básicas del programa.
2. Ayuda para el usuario, sobre las operaciones posibles y soluciones a problemas comunes.
3. Área de representación de entorno del agente inteligente.
4. Botón para la detención del monitoreo del agente inteligente.
5. Botón para detener las acciones de la Inteligencia del agente inteligente.
6. Área de botones de prueba de periféricos
7. Botones para configuraciones visuales.
8. Área de texto para visualizar mensajes del agente inteligente.
9. Área de graficación del compás del agente inteligente.
10. Área de graficación de sensor de terreno del agente inteligente.
11. Área de prueba de las conexiones y visualización del estado actual de los sistemas complementarios del agente inteligente.

CAPITULO 3: DESARROLLO DEL PROYECTO

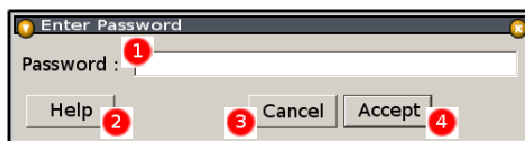


Figura 3-52. Dialogo de Ingreso del Password de Sistema de Monitoreo.

Aquí se usará la abreviación DIPSM (Dialogo de Ingreso del Password de Sistema de Monitoreo), acompañado por una de las siguientes referencias para aludir a la Figura 3-52:

1. *Campo de texto para el ingreso del password.*
2. *Botón con ayuda del contexto.*
3. *Botón para cancelar la acción de ingreso del password.*
4. *Botón para permitir que el sistema de monitoreo capture el password dado en el campo de texto.*

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Probar Conexión con Sistemas Complementarios.</i>	
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>	
<i>Tipo:</i>	<i>Primario.</i>	
<i>Descripción:</i>	<i>Este caso de uso comienza cuando el Administrador de Agente Inteligente solicita la prueba de la conexión con los Sistemas Complementarios del SAI.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	<i>R 4</i>	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- Este caso de uso comienza cuando el Administrador de Agente Inteligente, da la orden al SiM de probar la conexión con el SAI, a través de la realización de un clic sobre el título « SAI Connection » en el área de prueba de las conexiones y visualización del estado actual de los Sistemas Complementarios del Agente Inteligente. (SMCC 11)</i>	<i>2.-El SiM ejecuta la orden de petición de prueba de conexión con SAI.</i>	
	<i>3a.- El SiM informa al Administrador de Agente Inteligente, a través de un círculo verde al lado derecho del título « SAI Connection » (SMCC 11), que la conexión con el SAI está operativa.</i>	
<i>4.- El Administrador de Agente Inteligente solicita la prueba de la conexión con los Sistemas Complementarios del SAI a través de la realización de clic en el área de prueba de las conexiones y visualización del estado actual de los Sistemas Complementarios del Agente Inteligente. (SMCC 11).</i>	<i>5.- El SiM ejecuta la orden de petición de pruebas de conexiones con los Sistemas Complementarios.</i>	
	<i>6a.- El SiM informa al Administrador de Agente Inteligente, a través de círculos verdes al lado derecho de cada nombre de Sistema Complementario (SMCC 11), que todas las conexiones con los Sistemas Complementarios fueron logradas.</i>	
<i>Curso Alternativo de Eventos:</i>		
	<i>3a.- El SiM informa al Administrador de Agente Inteligente, a través de un círculo rojo al lado derecho del título « SAI Connection » (SMCC 11), que la conexión con SAI no esta operativa.</i>	
	<i>6b.- El SiM informa al Administrador de Agente Inteligente, a través de círculos rojos al lado derecho de cada nombre de Sistema Complementario (SMCC 11), que una o más conexiones con los Sistemas Complementarios no fueron logradas.</i>	

Tabla 3-41. Caso de uso “Probar Conexión con Sistemas Complementarios”.

CAPITULO 3: DESARROLLO DEL PROYECTO

En el caso de uso Ejecutar Protocolo de Inicio, dado que corresponden a llamadas que derivan a otros casos del SiM, no se realizará mayor detalle en esta etapa del proyecto, quedando su definición idéntica a la de la etapa de Análisis.

<i>Caso de Uso:</i>	<i>Detener Agente inteligente.</i>	
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>	
<i>Propósito:</i>	<i>Detener el desplazamiento del robot</i>	
<i>Resumen:</i>	<i>El Administrador de Agente Inteligente necesita la detención de actividades del Agente Inteligente. Para lo cual el SiM solicitará al Sistema de Agente Inteligente, el cese de sus actividades.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	<i>R 6</i>	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- Este caso de uso comienza con una orden de parada dada por el Administrador del Agente Inteligente, a través de apretar el botón « Stop Robot » (SMCC 5).</i>		
	<i>2.- El SiM envía un comando de parada al SAI.</i>	
	<i>3a.- El SiM confirma al Administrador de Agente Inteligente que el SAI ha detenido sus acciones a través de un mensaje dado por el área de texto para visualizar mensajes del Agente Inteligente. (SMCC 8)</i>	
<i>Curso Alternativo de Eventos:</i>		
	<i>3b.- El SiM informa al Administrador de Agente Inteligente que el SAI no ha podido detener sus acciones a través de un mensaje dado por el área de texto para visualizar mensajes del Agente Inteligente. (SMCC 8)</i>	

Tabla 3-42. Caso de uso “Detener Agente inteligente”.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Probar Efectores del SAI.</i>	
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>	
<i>Propósito:</i>	<i>Determinar si la dirección del Agente Inteligente funciona correctamente.</i>	
<i>Resumen:</i>	<i>Este caso de uso comienza cuando el Sistema de Monitoreo solicita la prueba de los efectores del SAI.</i>	
<i>Tipo:</i>	<i>Primario y Esencial.</i>	
<i>Referencia Cruzada:</i>	<i>R 7</i>	
<i>Curso Normal de Eventos:</i>		
<i>Actor</i>	<i>Sistema</i>	
<i>1.- Este caso de uso comienza con la orden, proveniente del Administrador de Agente Inteligente, solicitando una prueba de efectores a través de la realización de un clic sobre los botones « Direction » y « Traction » de forma secuencial, los cuales se encuentran en el área de botones de prueba de periféricos. (SMCC 6)</i>		
	<i>2.- El SiM envía una orden al SAI pidiendo ejecutar la prueba de efectores.</i>	
<i>3a.- El Administrador de Agente Inteligente confirma visualmente que la prueba de los efectores se ha realizado correctamente.</i>		
<i>Curso Alternativo de Eventos:</i>		
<i>3b.- El Administrador de Agente Inteligente determina visualmente que la prueba de los efectores no se ha realizado correctamente.</i>		

Tabla 3-43. Caso de uso “Probar Efectores del SAI”.

CAPITULO 3: DESARROLLO DEL PROYECTO

<i>Caso de Uso:</i>	<i>Probar Sensores del SAI.</i>
<i>Actores:</i>	<i>Administrador de Agente Inteligente.</i>
<i>Propósito:</i>	<i>Determinar si la tracción del Agente Inteligente funciona correctamente.</i>
<i>Resumen:</i>	<i>Este caso de uso comienza cuando el Sistema de Monitoreo solicita la prueba de los sensores del SAI.</i>
<i>Tipo:</i>	<i>Primario y Esencial.</i>
<i>Referencia Cruzada:</i>	<i>R 7</i>
<i>Curso Normal de Eventos:</i>	
<i>Actor</i>	<i>Sistema</i>
<i>1.- Este caso de uso comienza con el recibimiento de una orden, proveniente del Administrador de Agente Inteligente, solicitando una prueba de sensores a través de la realización de clic sobre los nombres de « U.Sonic », « Compass » y « DGPS » en el área de botones de prueba de periféricos. (SMCC 6)</i>	
	<i>2.- El SiM envía una orden al SAI pidiendo ejecutar la prueba de sensores.</i>
	<i>3a.- El SAI determina que las mediciones realizadas por los sensores son válidas, por lo cual el SiM lo informa al Administrador de Agente Inteligente a través de círculos verdes al lado derecho de los nombres de « U.Sonic », « Compass » y « DGPS » en el área de botones de prueba de periféricos. (SMCC 6)</i>
<i>Curso Alternativo de Eventos:</i>	
	<i>3b.- El SiM determina que las mediciones realizadas por los sensores no son válidas, y lo informa al Administrador de Agente Inteligente a través de círculos rojos al lado derecho de los nombres de « U.Sonic », « Compass » y « DGPS » en el área de botones de prueba de periféricos. (SMCC 6).</i>

Tabla 3-44. Caso de uso “Probar Sensores del SAI”.

3.6.2 Diagrama de Clase SiM

Los paquetes que a continuación se presentan desde la Figura 3-54 hasta la Figura 3-57, corresponden a la división de la arquitectura multicapa orientada a objeto, en la cual cada paquete representado tomará el nombre de las capas de Gráfica y Lógica de este modelo. En la Figura 3-53, se grafica el diagrama general de paquetes.

Del modelo conceptual presentado en la Figura 3-31, y a partir de lo definido en la sección 3.4.3, no se agregarán nuevas clases al diagrama. Sin embargo, dado el concepto “Prueba”, es necesario agregar nuevas funcionalidades a las clases “ControladorLogico” y “ClienteDeMonitoreo” y nuevos valores a la interfaz “InterfazDeComunicacion” del paquete “logica”.

3.6.2.1 Diagrama General de Paquetes

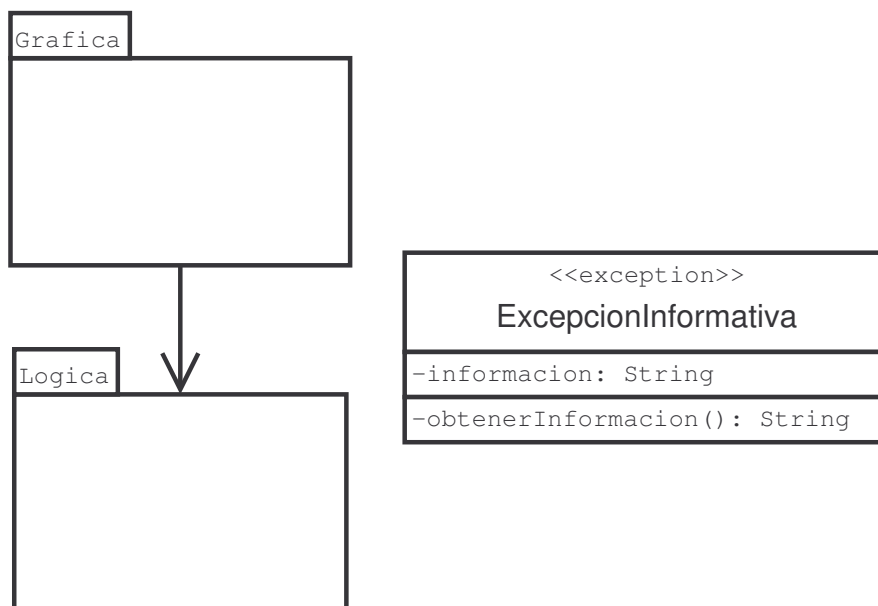


Figura 3-53. Diagrama general de paquetes.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.6.2.2 Diagrama de Clase (Paquete logica)

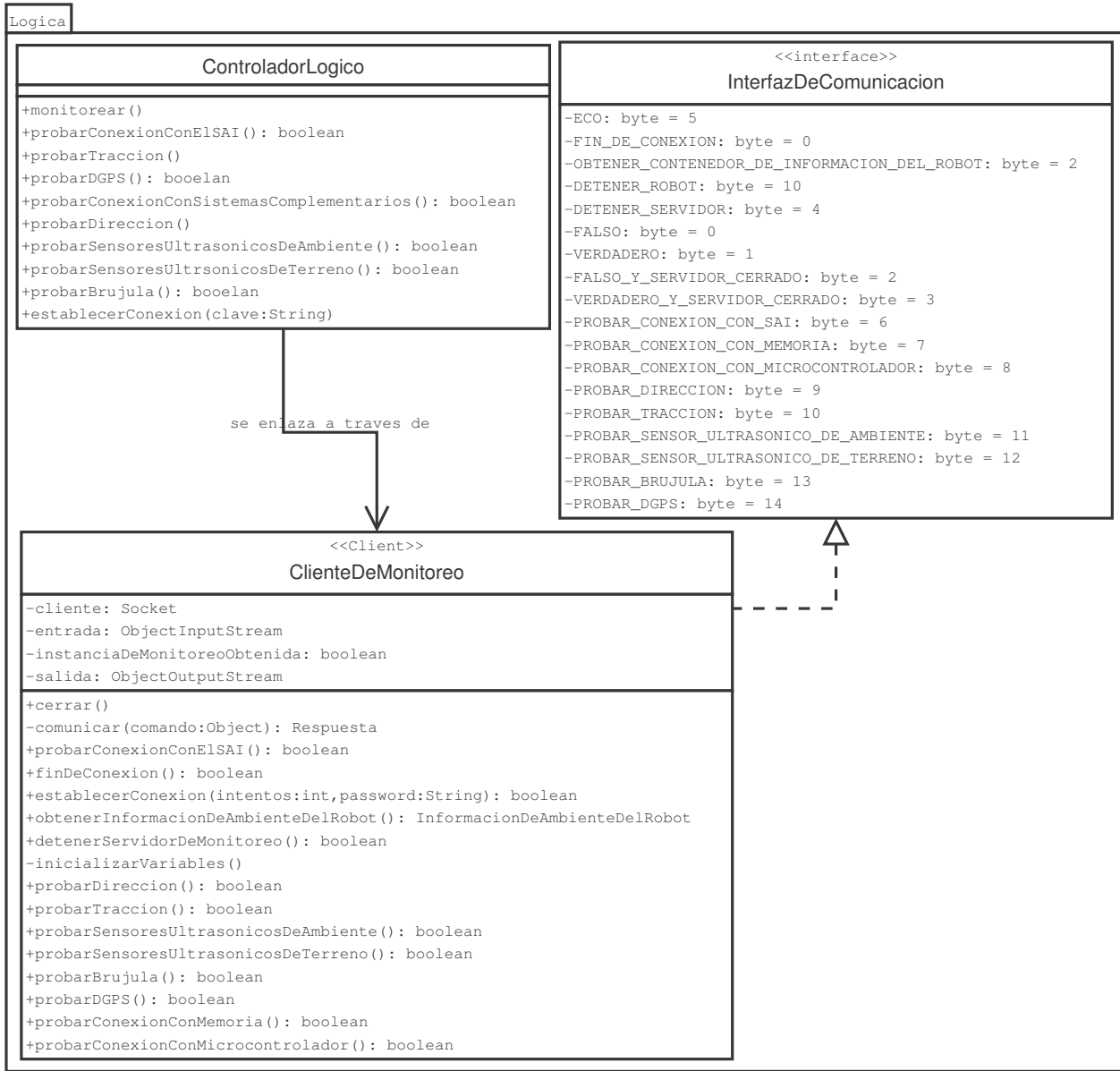


Figura 3-54. Diagrama de Clases, paquete “logica”.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.6.2.3 Diagrama de Clase (Paquete grafica)

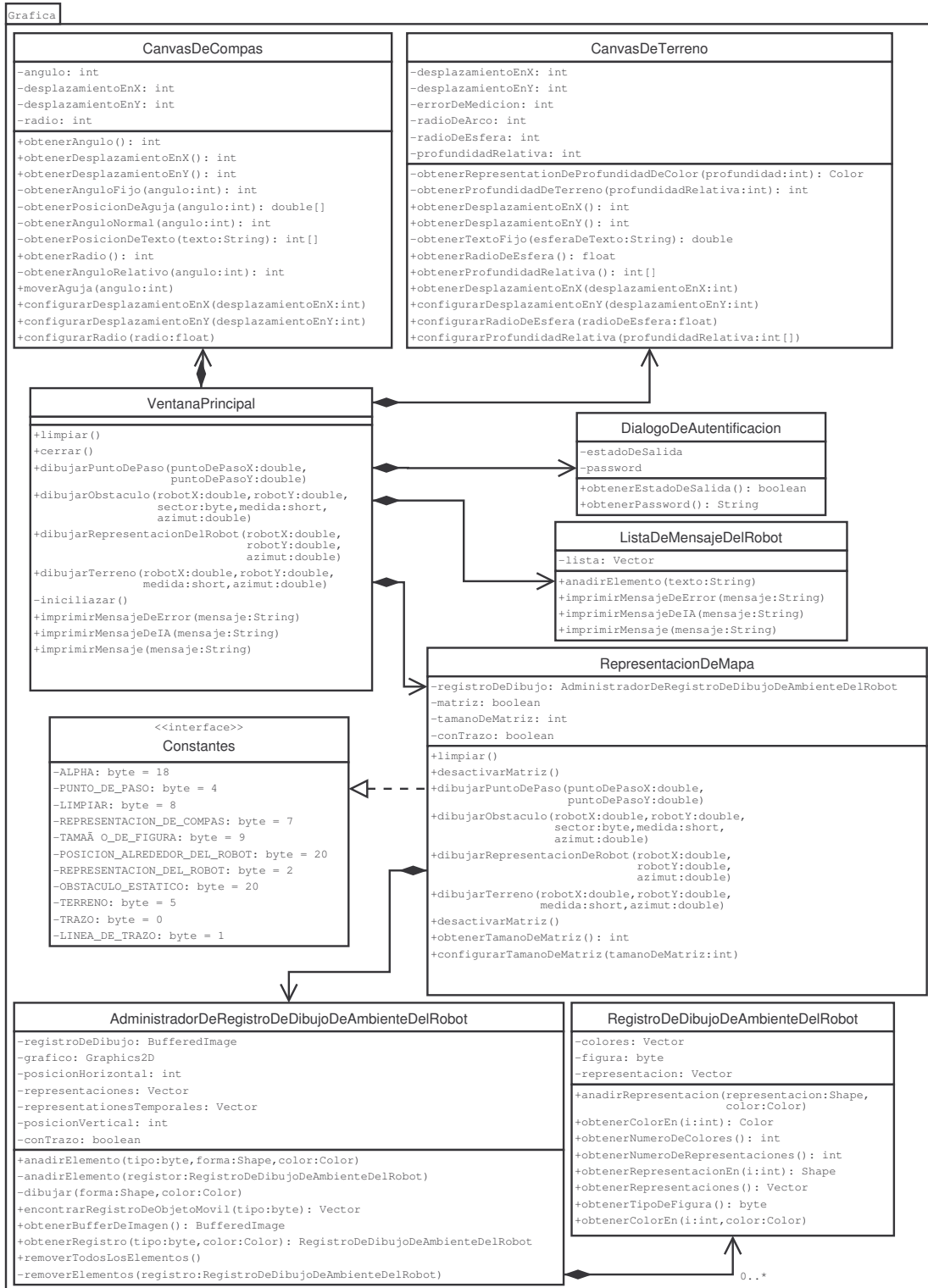


Figura 3-55. Diagrama de Clases, paquete “grafica”.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.6.2.4 Bean

Referente a estas clases, podemos decir que su objetivo es operar a través de las diferentes capas de la arquitectura del software, transportando información de forma encapsulada.

En la Figura 3-56 se muestra el Bean definido para el problema.

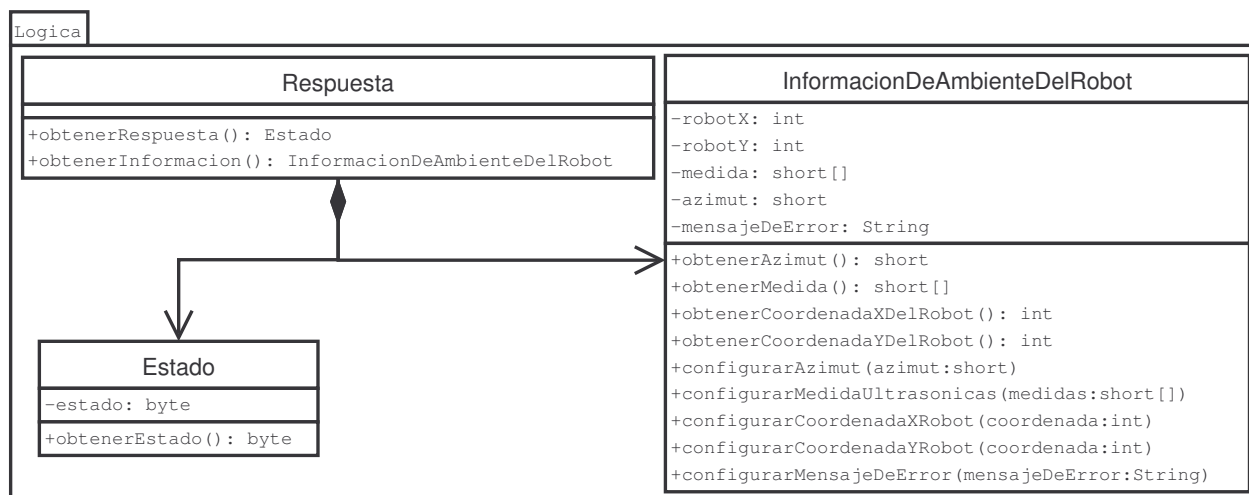


Figura 3-56. Clases Bean del problema.

3.6.2.5 Excepción

Referente a estas clases, podemos decir que su objetivo es operar a través de las diferentes capas de la arquitectura del software, transportando información de forma encapsulada sobre algún evento crítico ocurrido.

En la Figura 3-57 se muestra la clase excepción definidos para el dominio del problema.

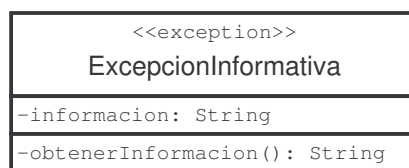


Figura 3-57. Clase de excepción del problema.

3.6.3 Diagramas de Colaboración del SiM

Desde la Figura 3-58 hasta la Figura 3-63 se dan a conocer los diagramas de colaboración para el segundo incremento del Sistema de Monitoreo.

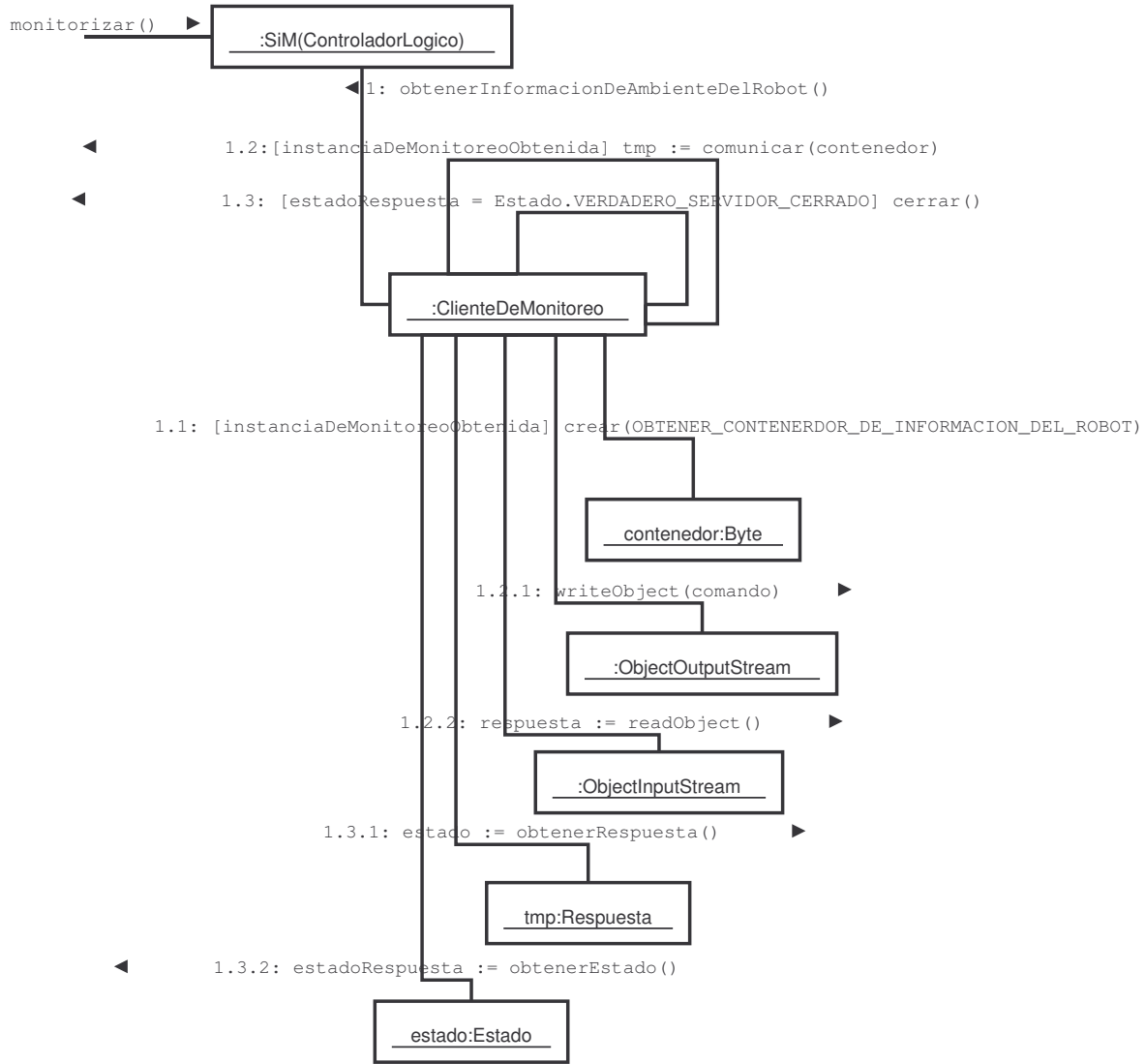


Figura 3-58. Diagrama de Colaboración Monitorizar.

CAPITULO 3: DESARROLLO DEL PROYECTO

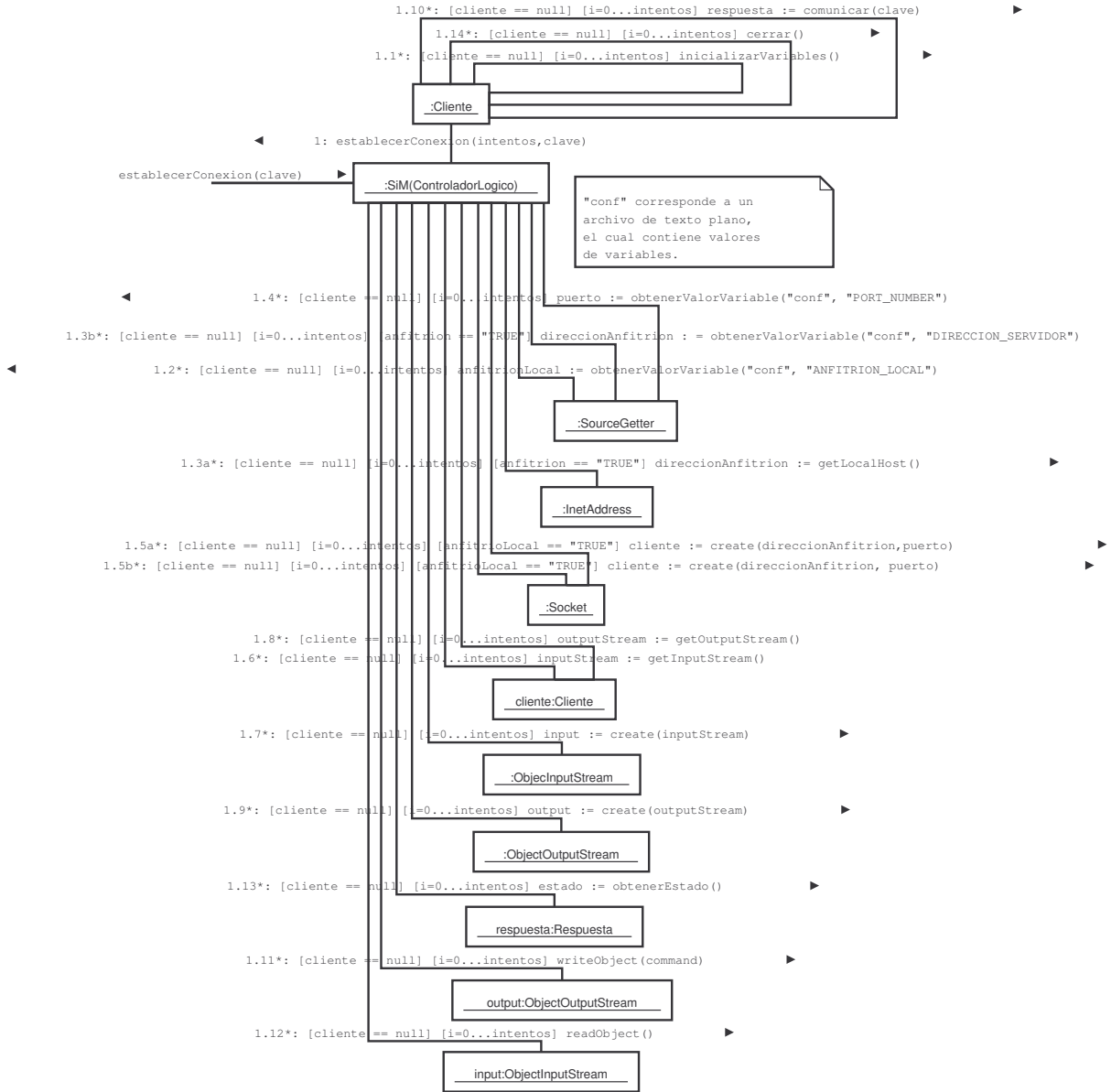


Figura 3-59. Diagrama de Colaboración Autenticar.

CAPITULO 3: DESARROLLO DEL PROYECTO

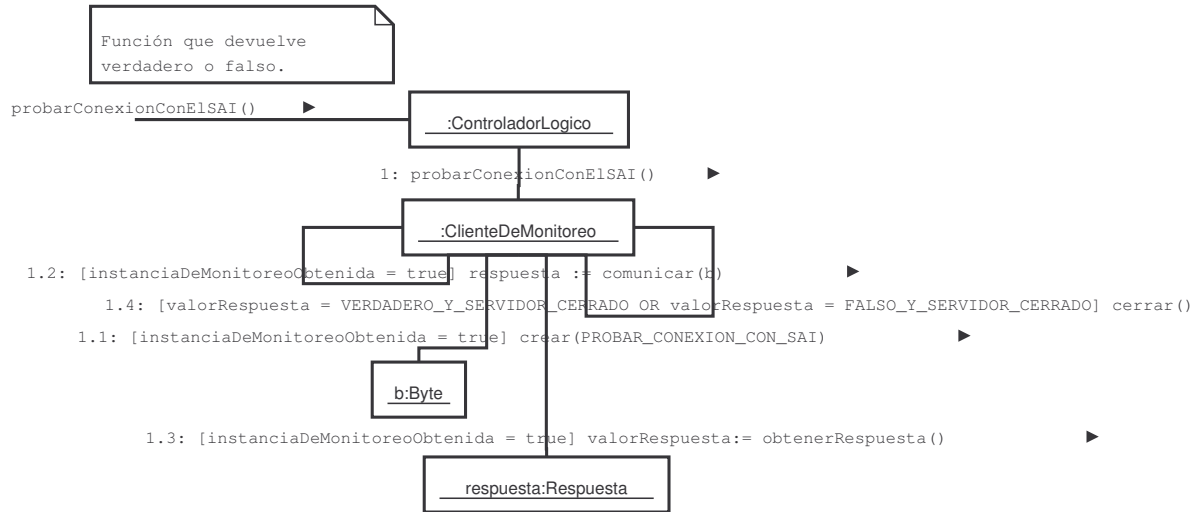


Figura 3-60. Diagrama de Colaboración probar conexión con el SAI.

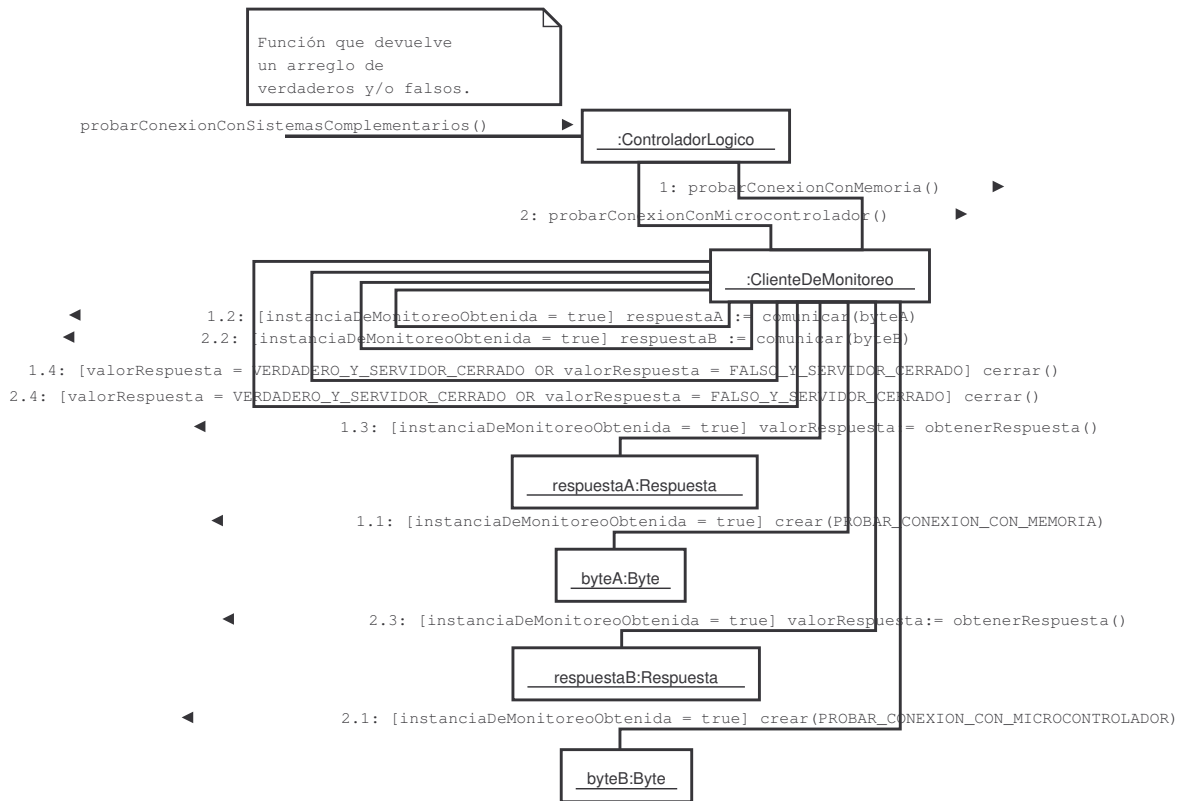


Figura 3-61. Diagrama de Colaboración probar conexión con sistemas complementarios.

CAPITULO 3: DESARROLLO DEL PROYECTO

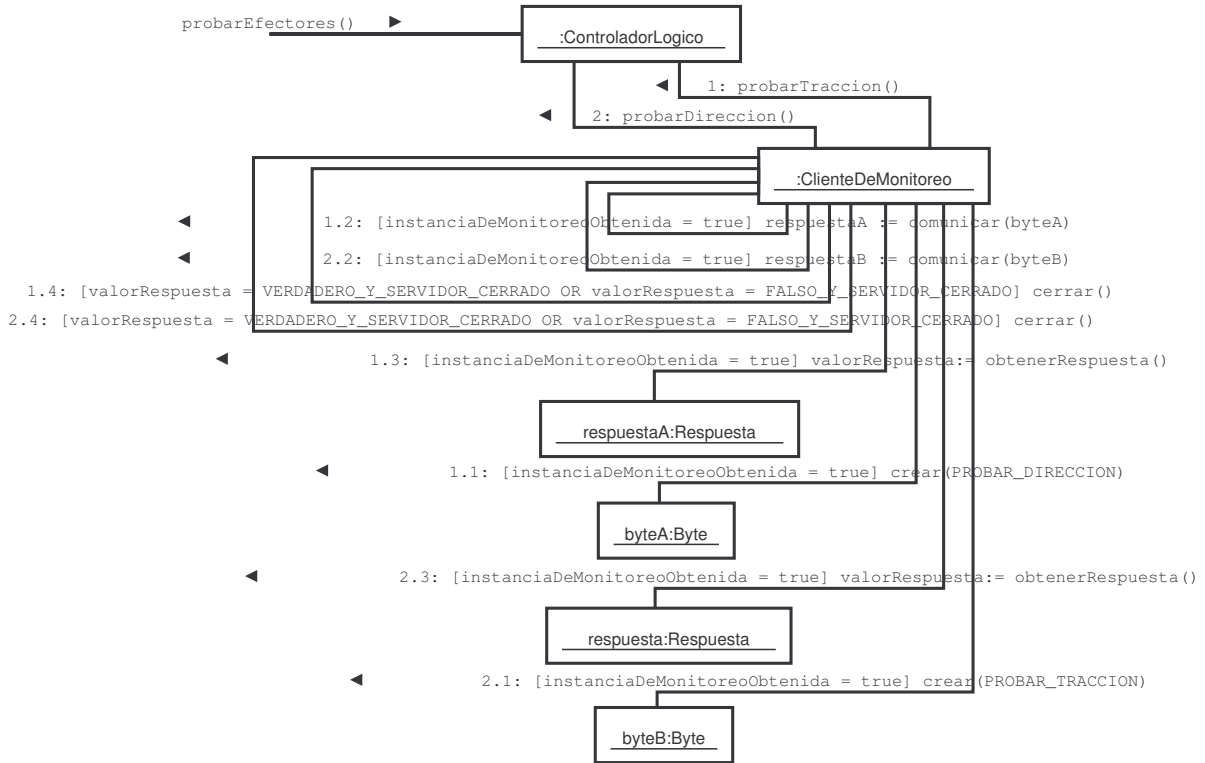


Figura 3-62. Diagrama de Colaboración probar efectores.

CAPITULO 3: DESARROLLO DEL PROYECTO

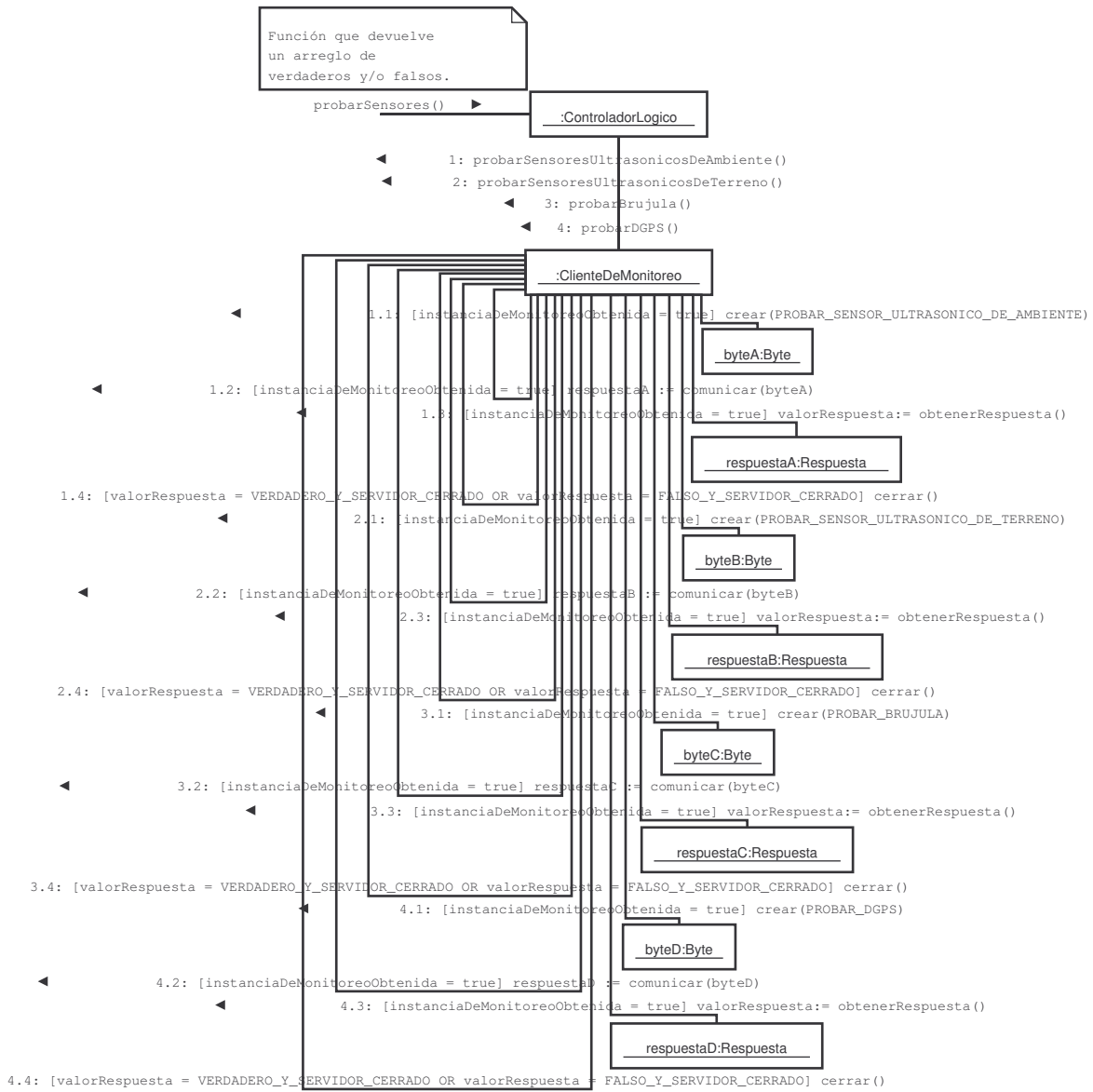


Figura 3-63. Diagrama de Colaboración probar sensores.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.6.4 Diagrama de Clase SAI

Los paquetes que a continuación se presentarán desde la Figura 3-65 hasta la Figura 3-71, corresponden a la división de la arquitectura multicapa orientada a objeto, en la cual cada paquete representado se clasificará dentro de las capas de Lógica y Datos, según corresponda. . En la Figura 3-64, se grafica el diagrama general de paquetes.

Del modelo conceptual presentado en la Figura 3-32, y a partir de lo definido en la sección 3.4.5, se agregará el paquete “baseDeConocimiento” y las clases “MemoriaDeAsociacionConBusquedaEuclidiana”, “ContenedorDePosicion”, “AVLPatron”, “ModuloDeInteligenciaArtificial” al paquete “logica”, las que incluirán el concepto de “Inteligencia” al este diagrama. Los conceptos de “PuntoDeDestino” y “PuntoDeChequeo” serán incluidos dentro de la “Inteligencia” como variables que determinen la toma de decisiones de esta última.

3.6.4.1 Diagrama General de Paquetes

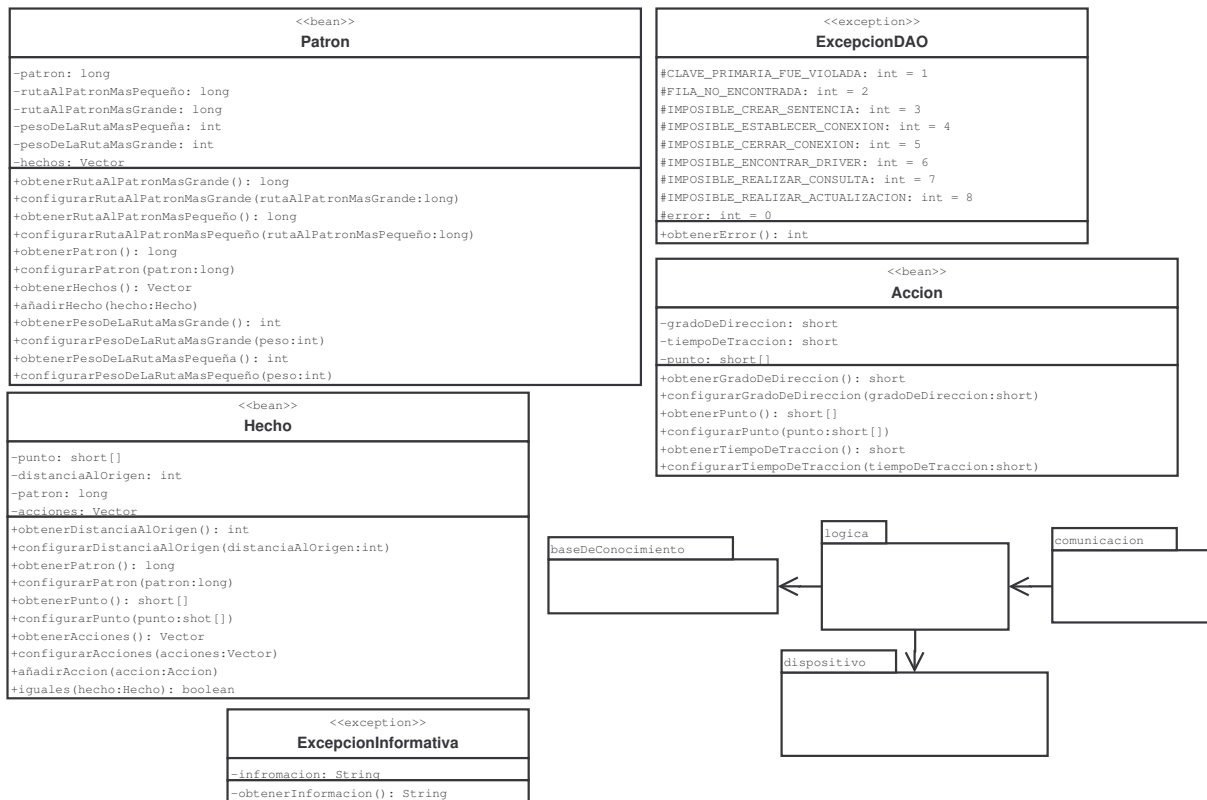


Figura 3-64. Diagrama general de paquetes.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.6.4.2 Diagrama de Clase (Paquete Comunicacion)



Figura 3-65. Diagrama de Clases, paquete “Comunicacion”.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.6.4.3 Diagrama de Clase (Paquete Dispositivo)



Figura 3-66. Diagrama de Clases, paquete “Dispositivo”.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.6.4.4 Diagrama de Clase (Paquete Logica)

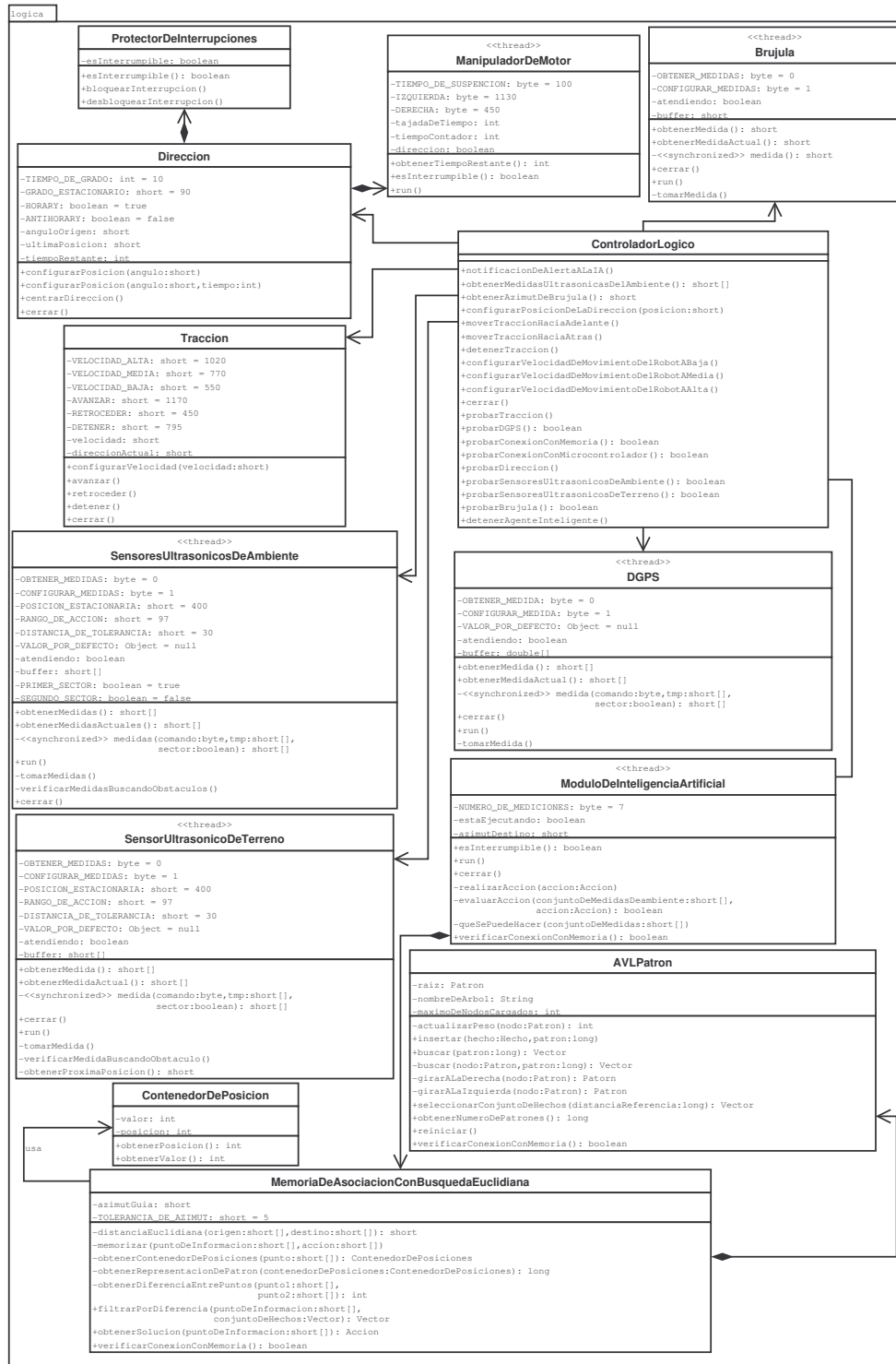


Figura 3-67. Diagrama de Clases, paquete “Logica”.

3.6.4.5 Diagrama de Clase (Paquete BaseDeConocimiento)

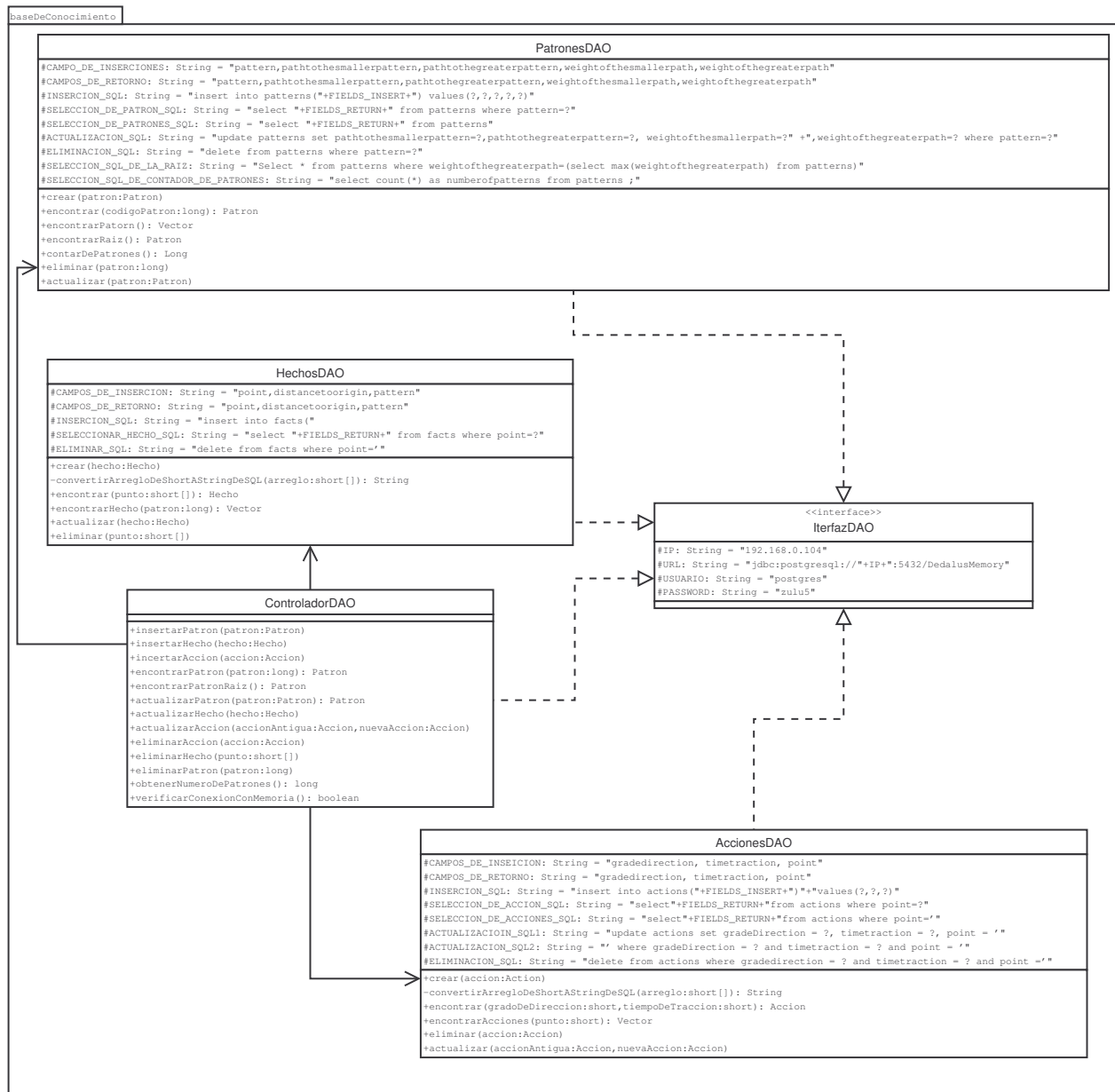


Figura 3-68. Diagrama de Clases, paquete “BaseDeConocimiento”.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.6.4.6 Beans

Referente a estas clases, podemos decir que su objetivo es operar a través de las diferentes capas de la arquitectura del software, transportando información de forma encapsulada.

3.6.4.6.1 Beans de la Base de Datos

En la Figura 3-69 se muestra los Beans definido para la base de datos.

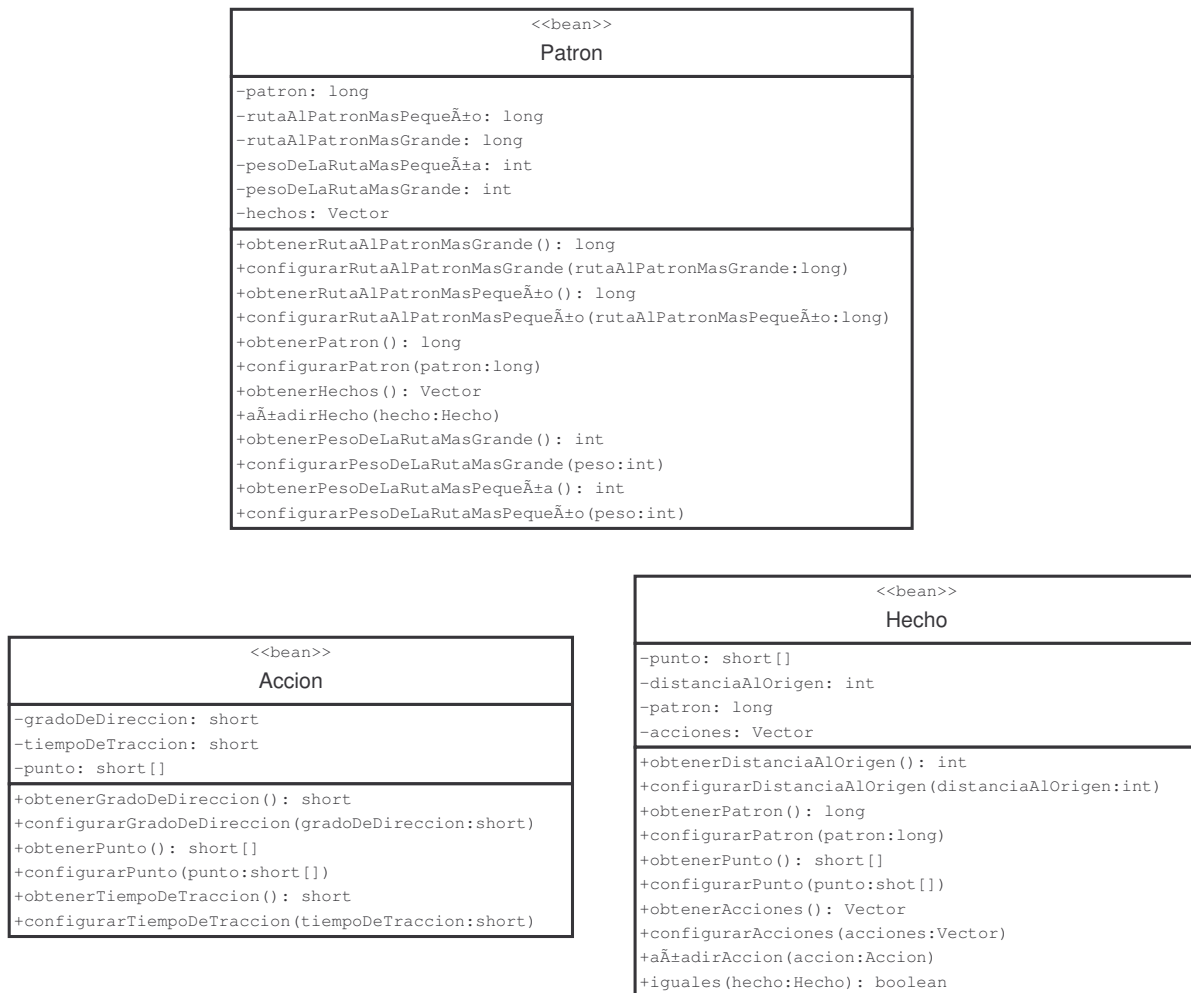


Figura 3-69. Diagrama de Clases“Beans”.

CAPITULO 3: DESARROLLO DEL PROYECTO

3.6.4.6.2 Beans del Paquete de Lógica

En la Figura 3-70 se muestran los Beans definidos para el paquete comunicación en el paquete lógica.

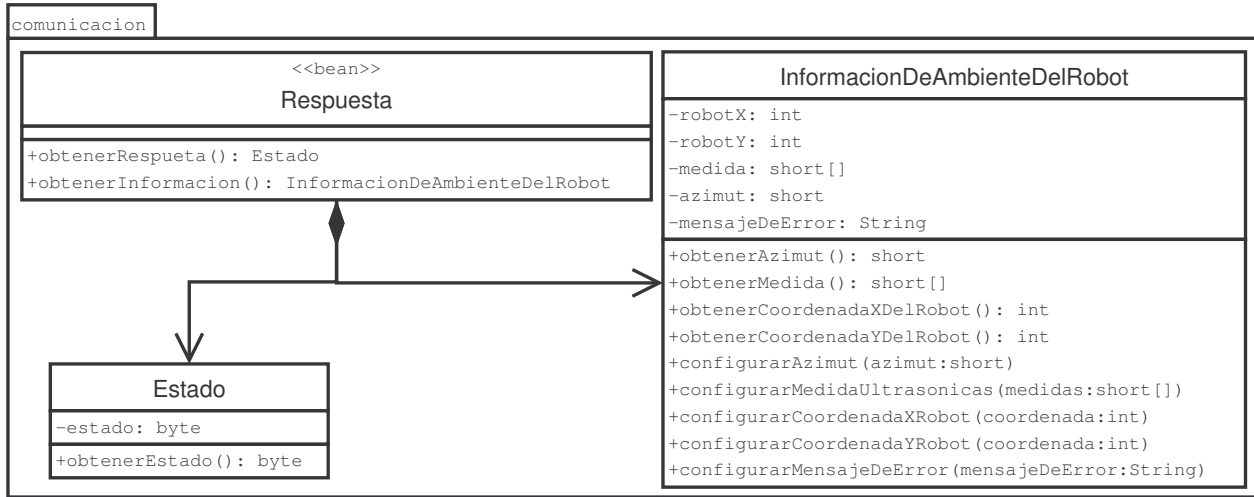


Figura 3-70. Diagrama de Clases“Beans”.

3.6.4.7 Excepciones

Referente a estas clases, podemos decir que su objetivo es operar a través de las diferentes capas de la arquitectura del software, transportando información de forma encapsulada sobre algún evento crítico ocurrido.

En la Figura 3-71 se muestran las excepción definidas para el dominio del problema.

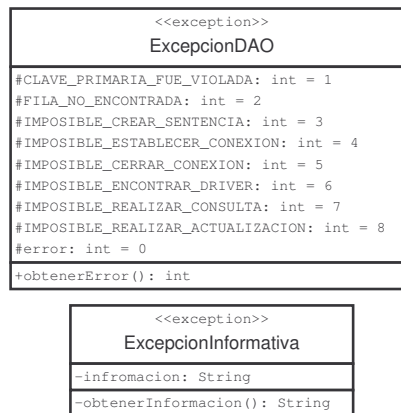


Figura 3-71. Diagrama de Clases “Excepciones”.

3.6.5 Modelo Entidad Relación

En la Figura 3-72, se presenta el MER referente a la base de conocimiento del agente inteligente.

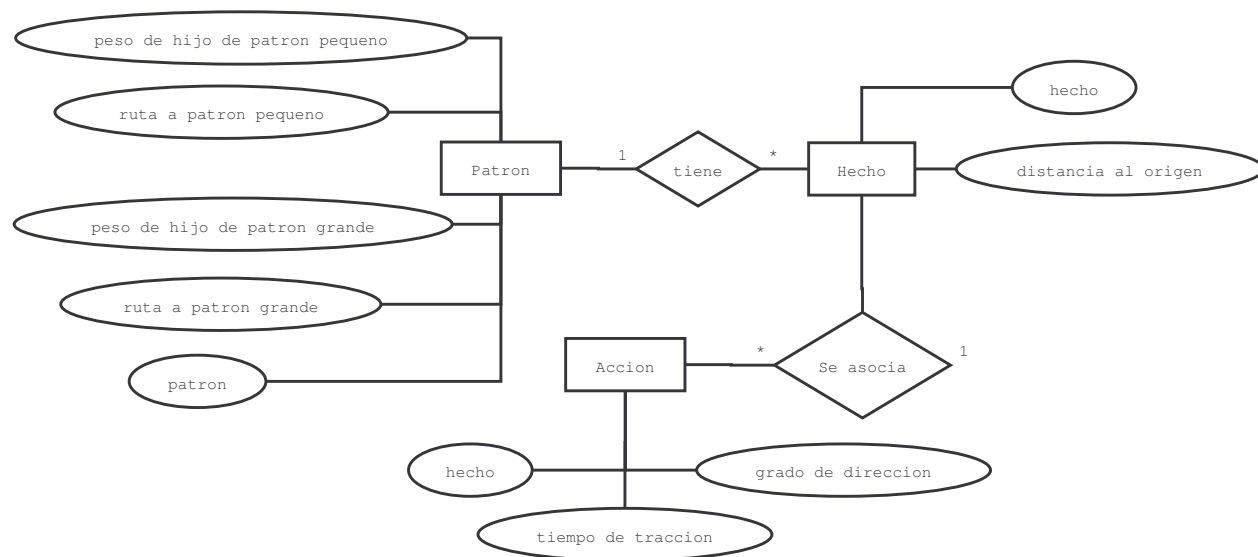


Figura 3-72. Diagrama Modelo Entidad Relación.

3.6.6 Diagramas de Colaboración del SAI

Desde la Figura 3-73 hasta la Figura 3-105 se dan a conocer los diagramas de colaboración para el segundo incremento del Sistema de Agente Inteligente.

Cabe señalar que en esta sección los diagramas “Diagrama de Colaboración probar conexión con microcontrolador” (Figura 3-76, Figura 3-77 y Figura 3-78), “Diagrama de Colaboración detener agente inteligente” (Figura 3-79 hasta la Figura 3-84), “Diagrama de Colaboración probar dirección” (Figura 3-85 hasta la Figura 3-89) y “Diagrama de Colaboración probar tracción” (Figura 3-90 hasta la Figura 3-94), poseen dos variables llamadas “*entradaLista*” y “*salidaLista*”, las cuales son manipuladas externamente por los hilos del buffer de entrada y salida de la comunicación serial provista por la API JavaComm mencionada en la sección 3.4.1.

CAPITULO 3: DESARROLLO DEL PROYECTO

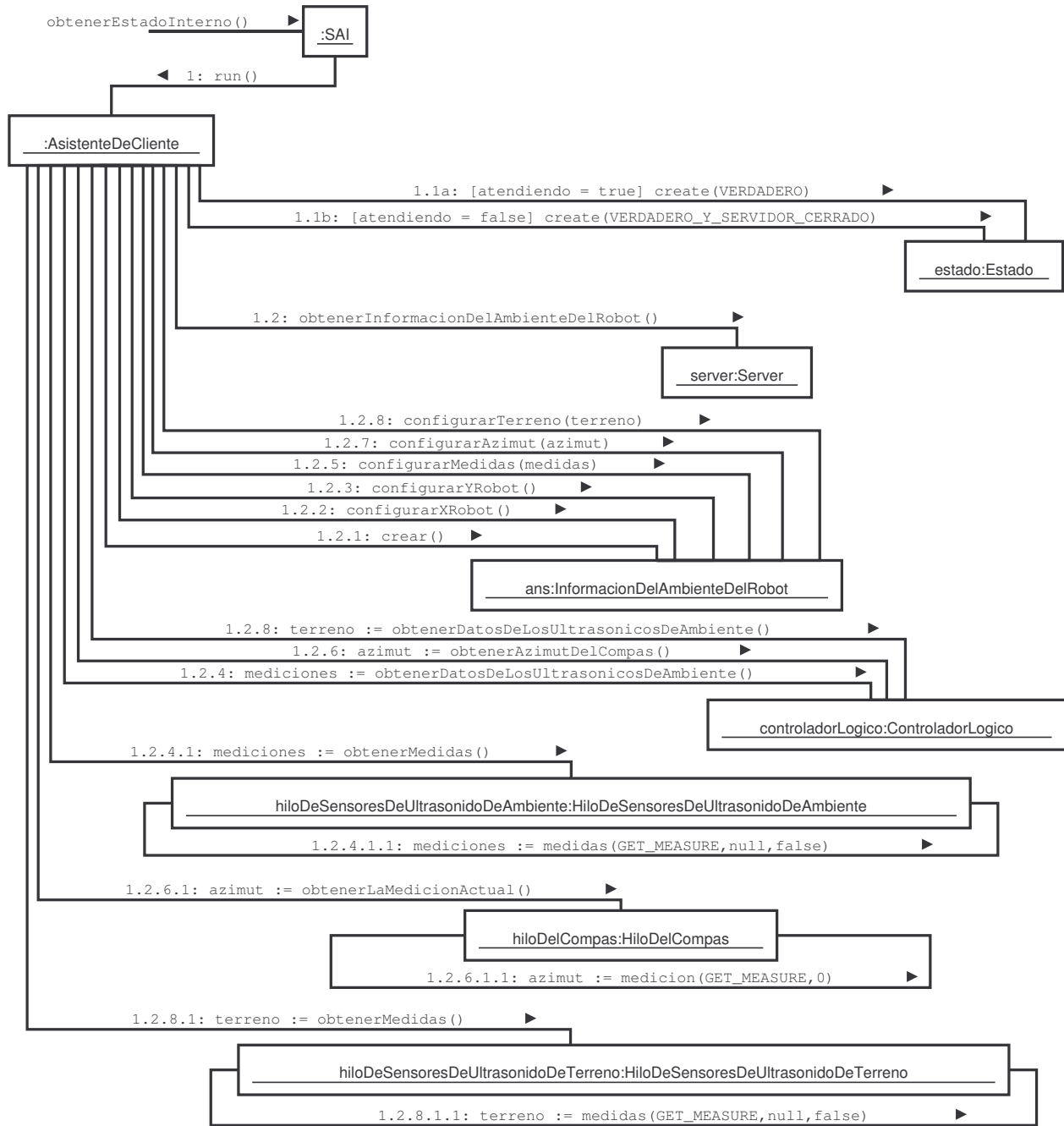


Figura 3-73. Diagrama de Colaboración obtener estado interno.

CAPITULO 3: DESARROLLO DEL PROYECTO

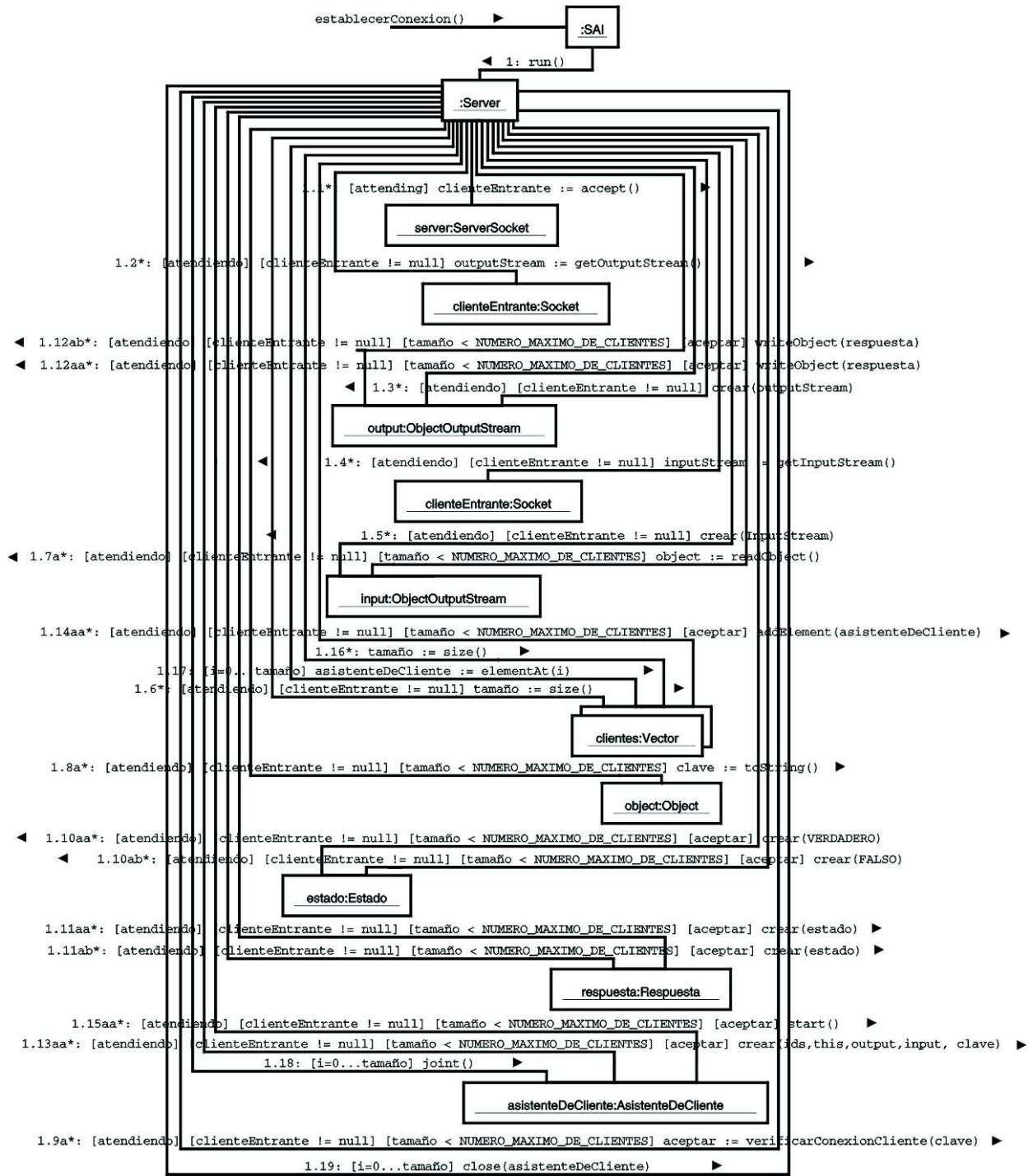


Figura 3-74. Diagrama de Colaboración establecer conexión.

CAPITULO 3: DESARROLLO DEL PROYECTO

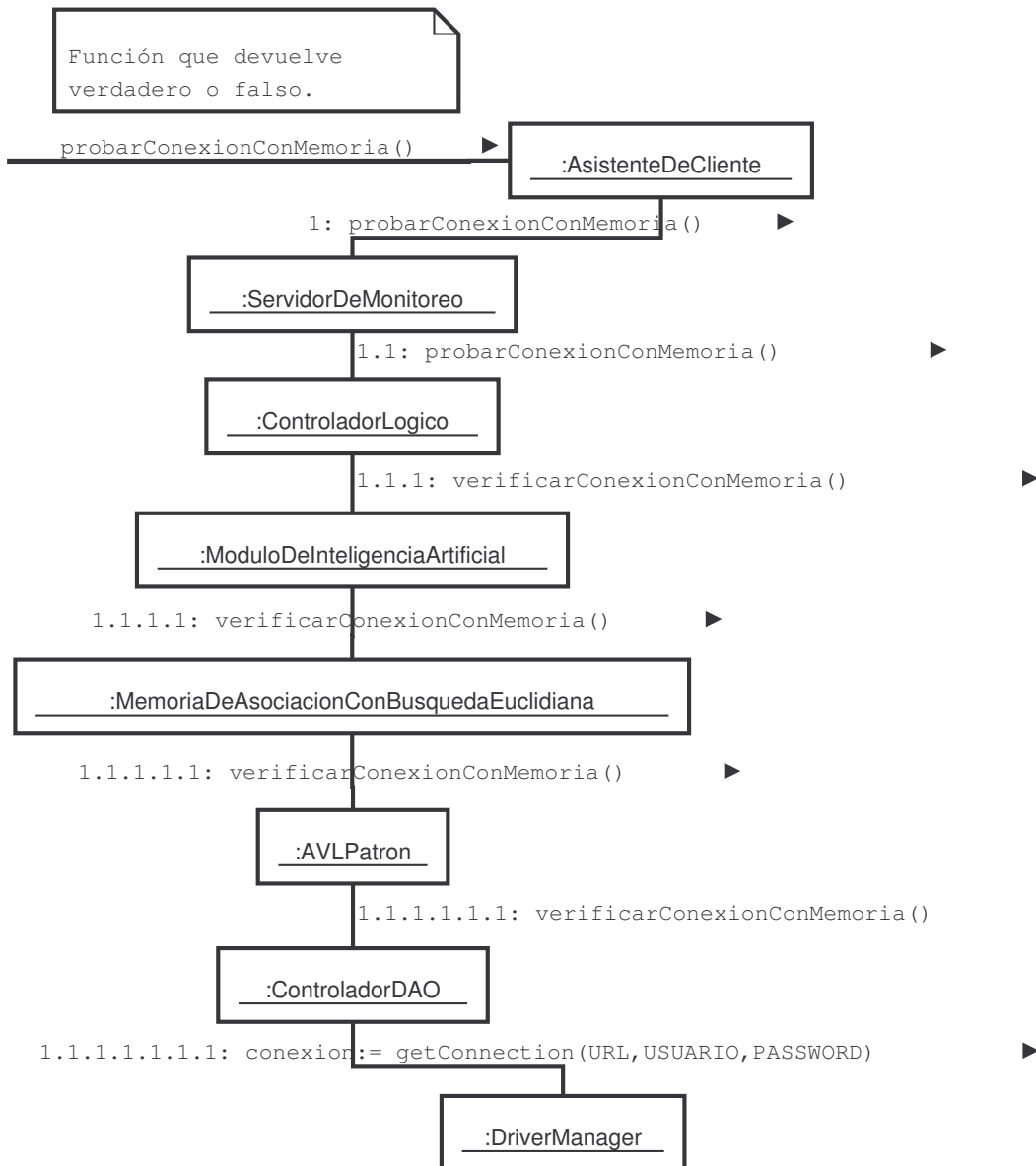


Figura 3-75. Diagrama de Colaboración probar conexión con memoria.

CAPITULO 3: DESARROLLO DEL PROYECTO

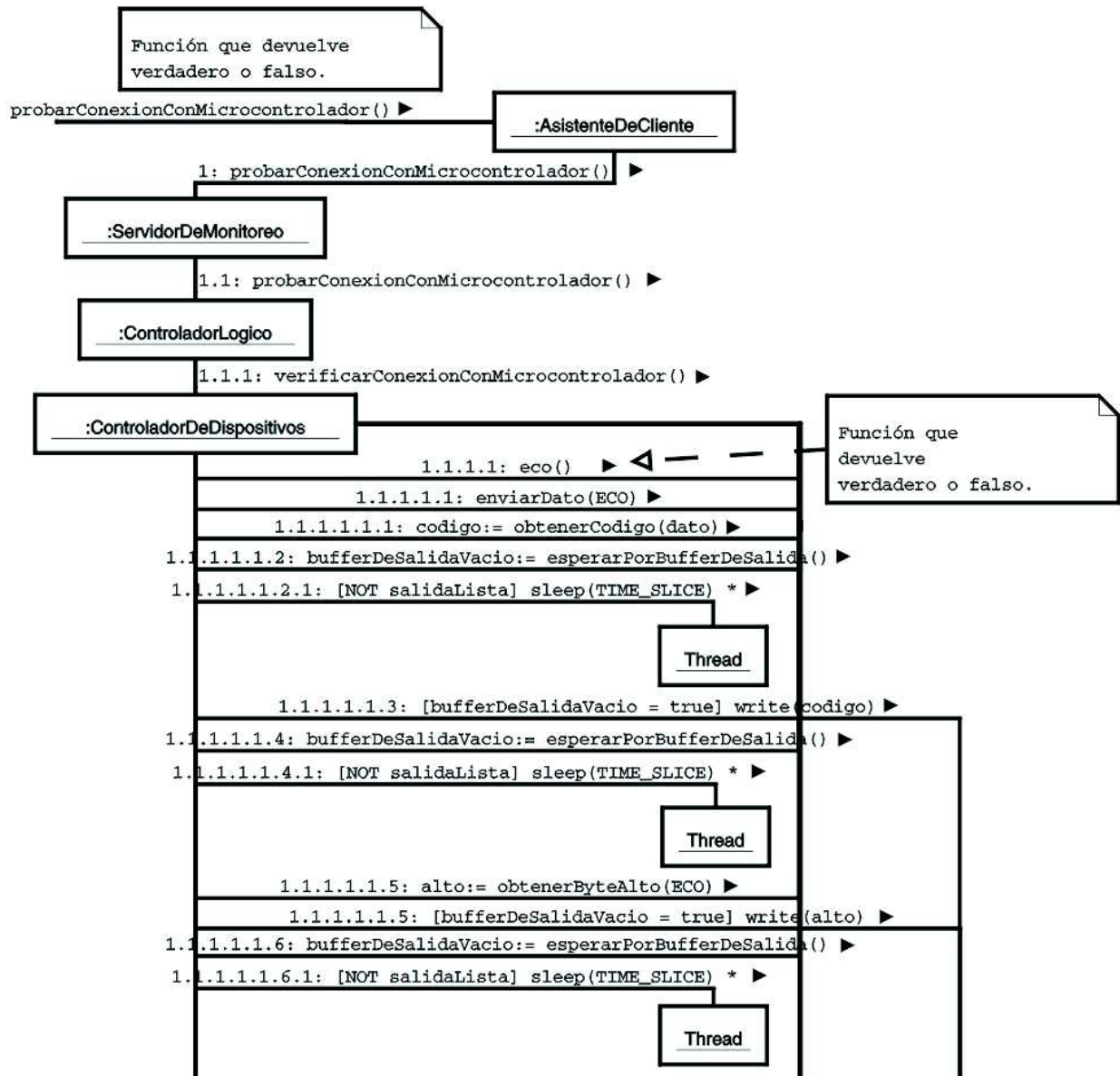


Figura 3-76. Diagrama de Colaboración probar conexión con microcontrolador, primera parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

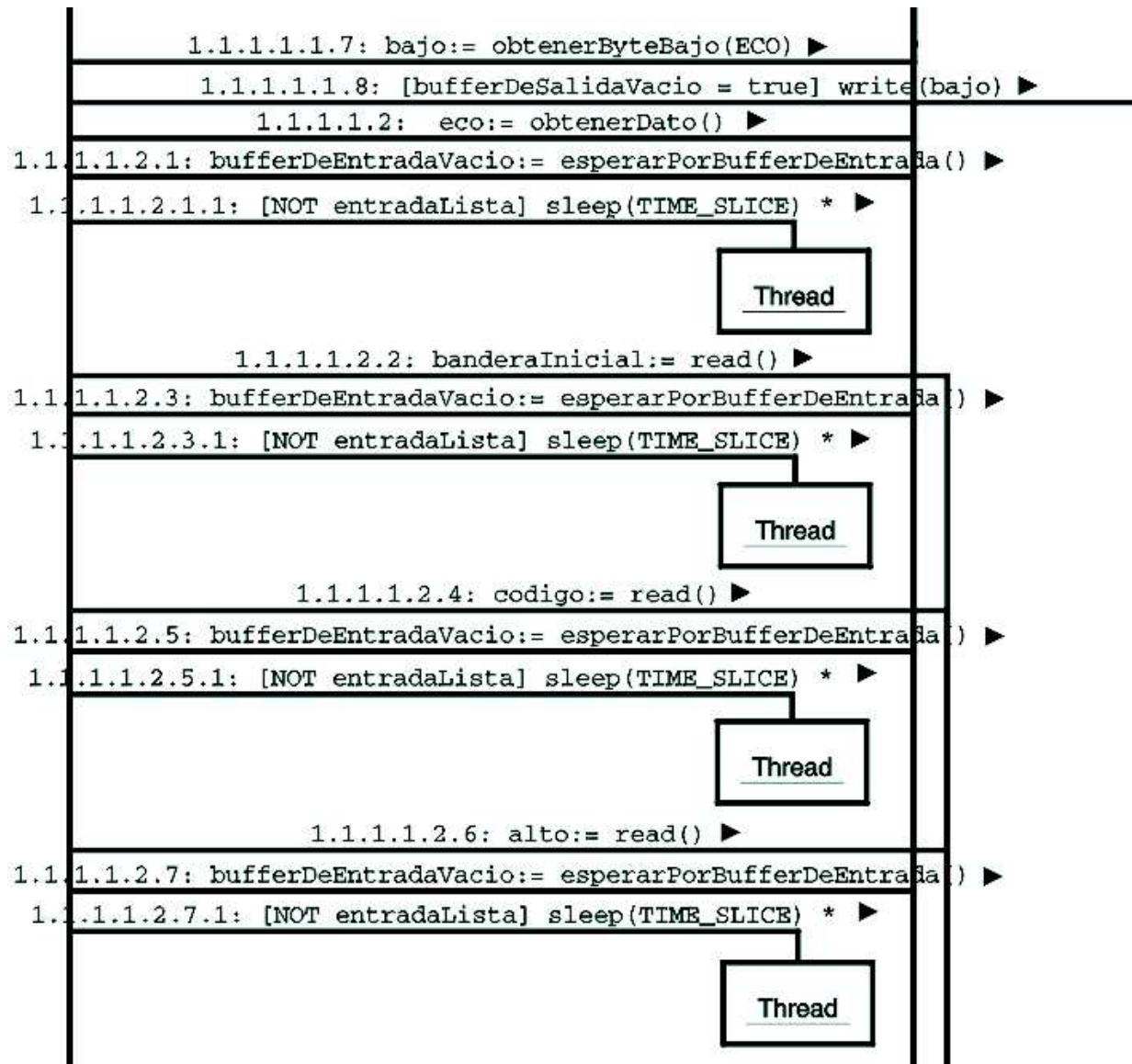


Figura 3-77. Diagrama de Colaboración probar conexión con microcontrolador, segunda parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

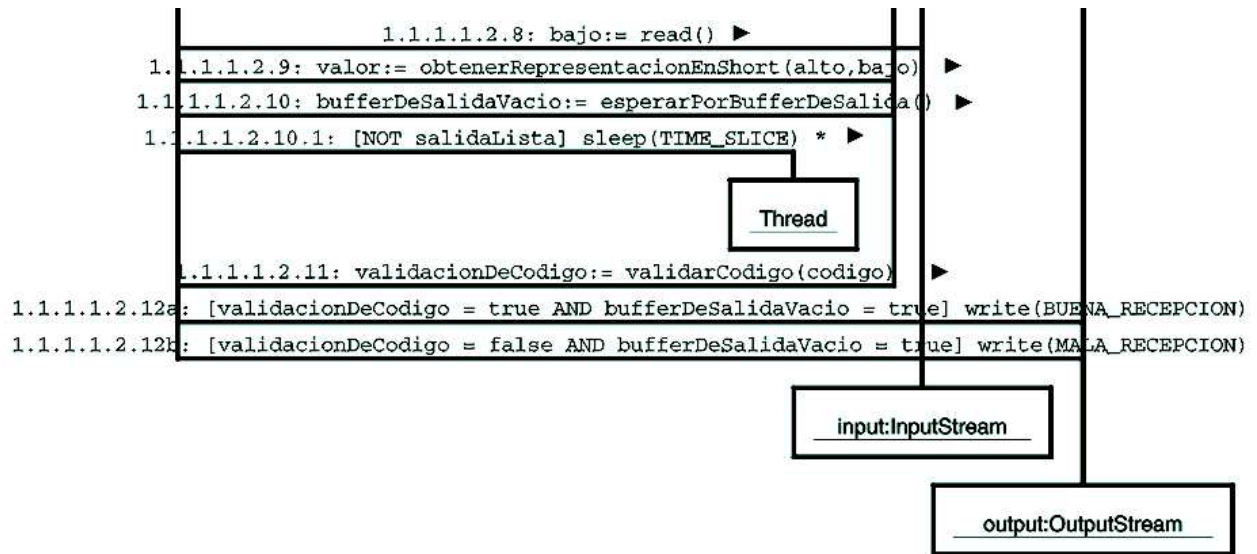


Figura 3-78. Diagrama de Colaboración probar conexión con microcontrolador, tercera parte y final.

CAPITULO 3: DESARROLLO DEL PROYECTO

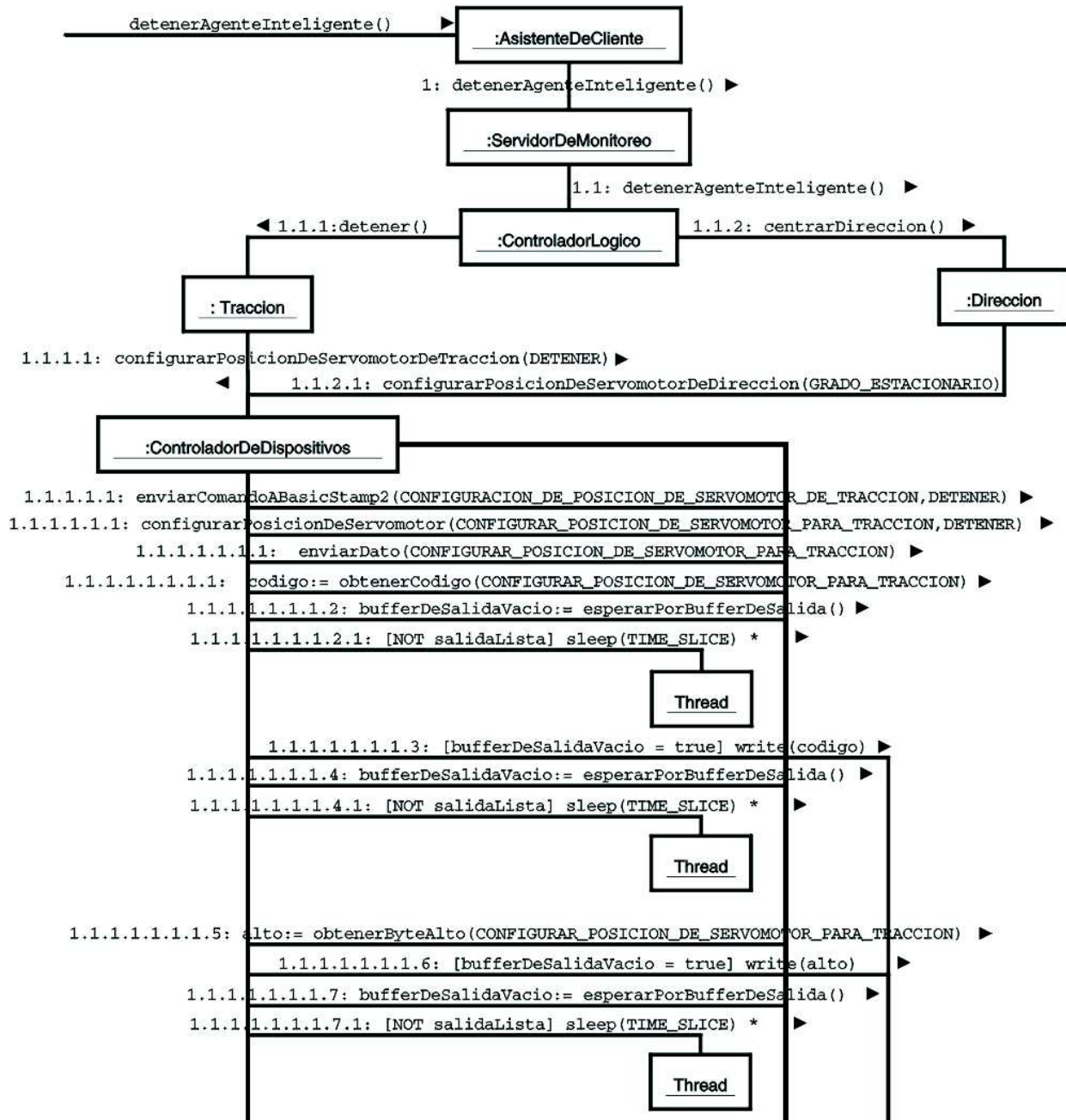


Figura 3-79. Diagrama de Colaboración detener agente inteligente, primera parte.

CAPITULO 3: DESARROLLO DEL PROYECTO



Figura 3-80. Diagrama de Colaboración detener agente inteligente, segunda parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

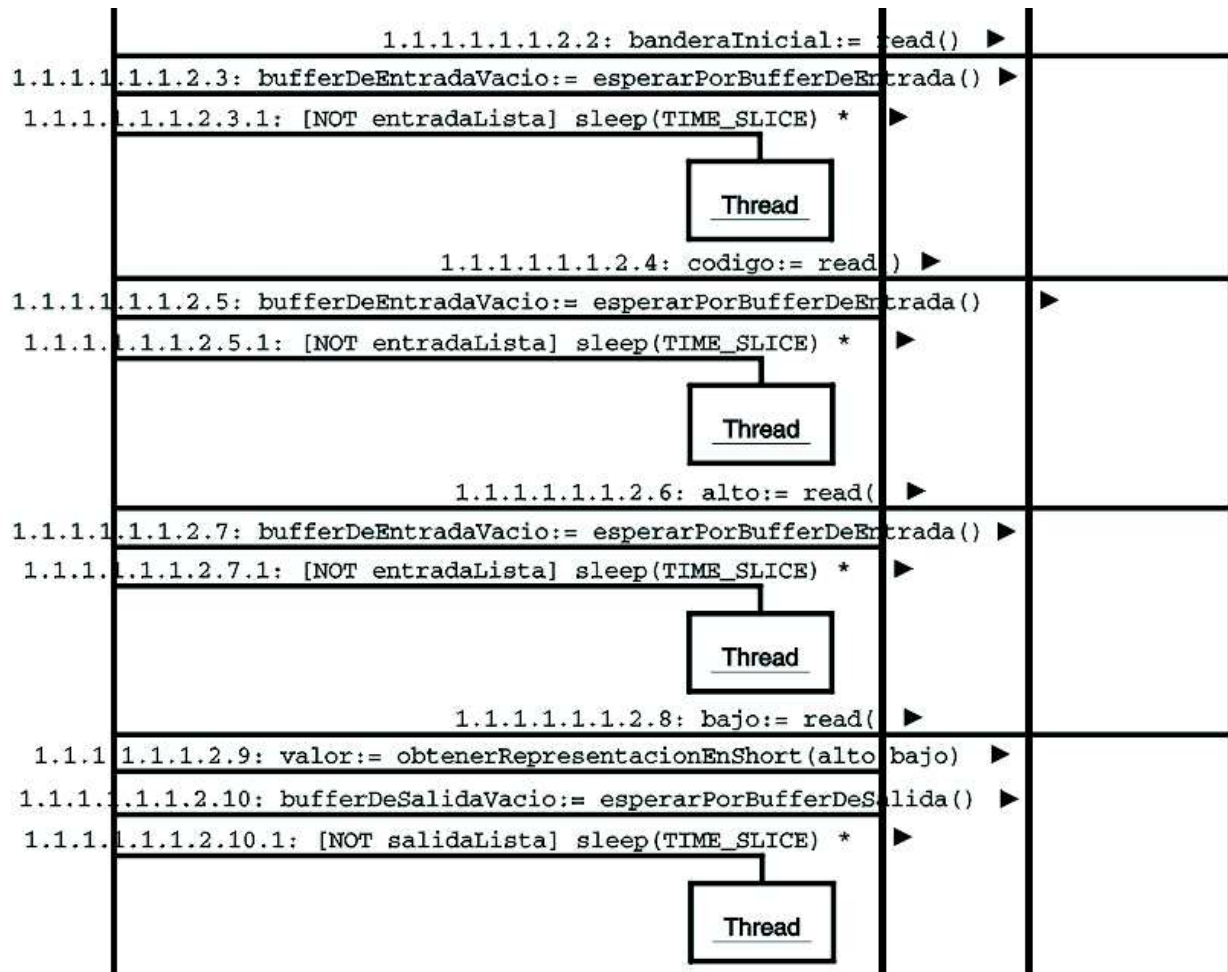


Figura 3-81. Diagrama de Colaboración detener agente inteligente, tercera parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

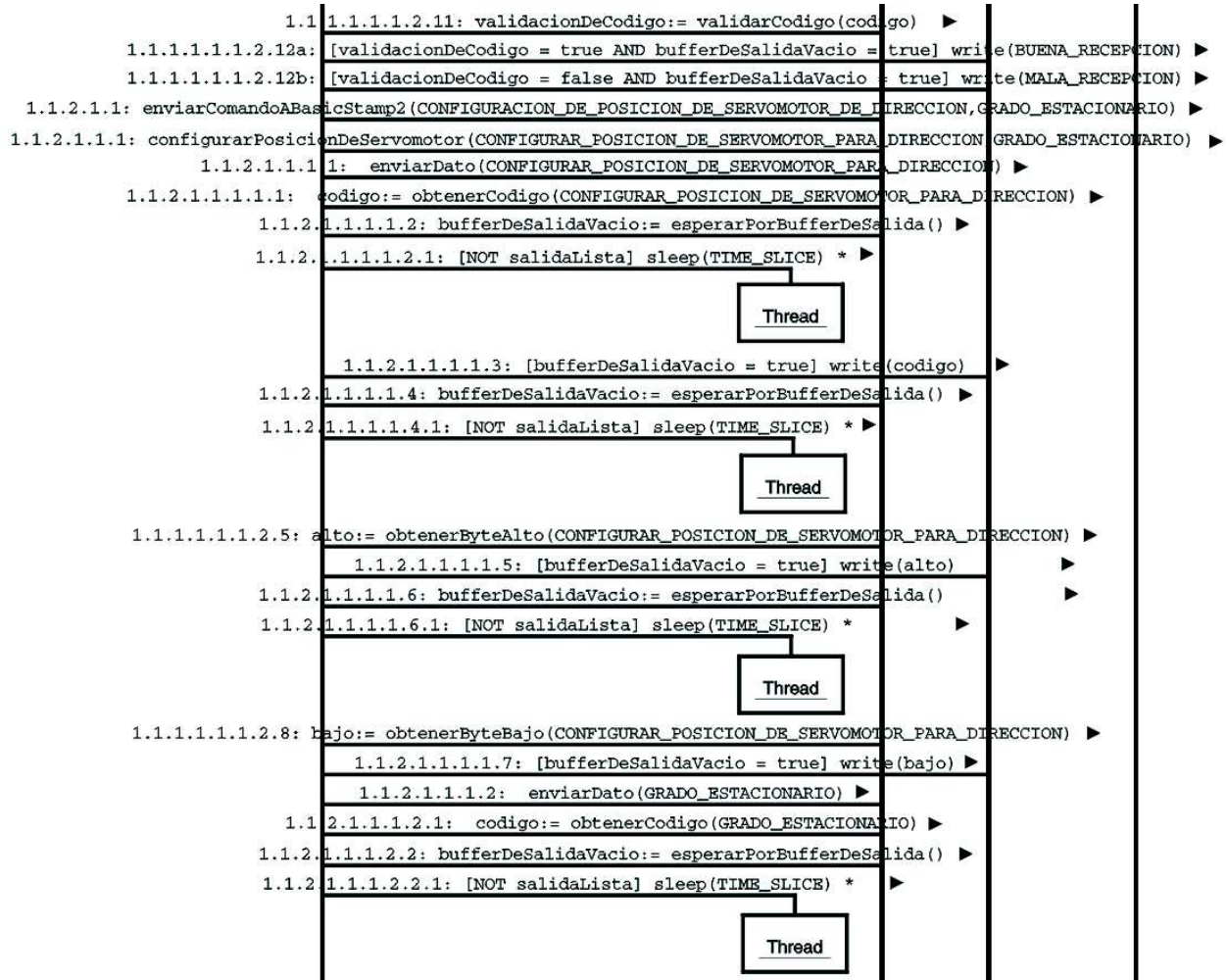


Figura 3-82. Diagrama de Colaboración detener agente inteligente, cuarta parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

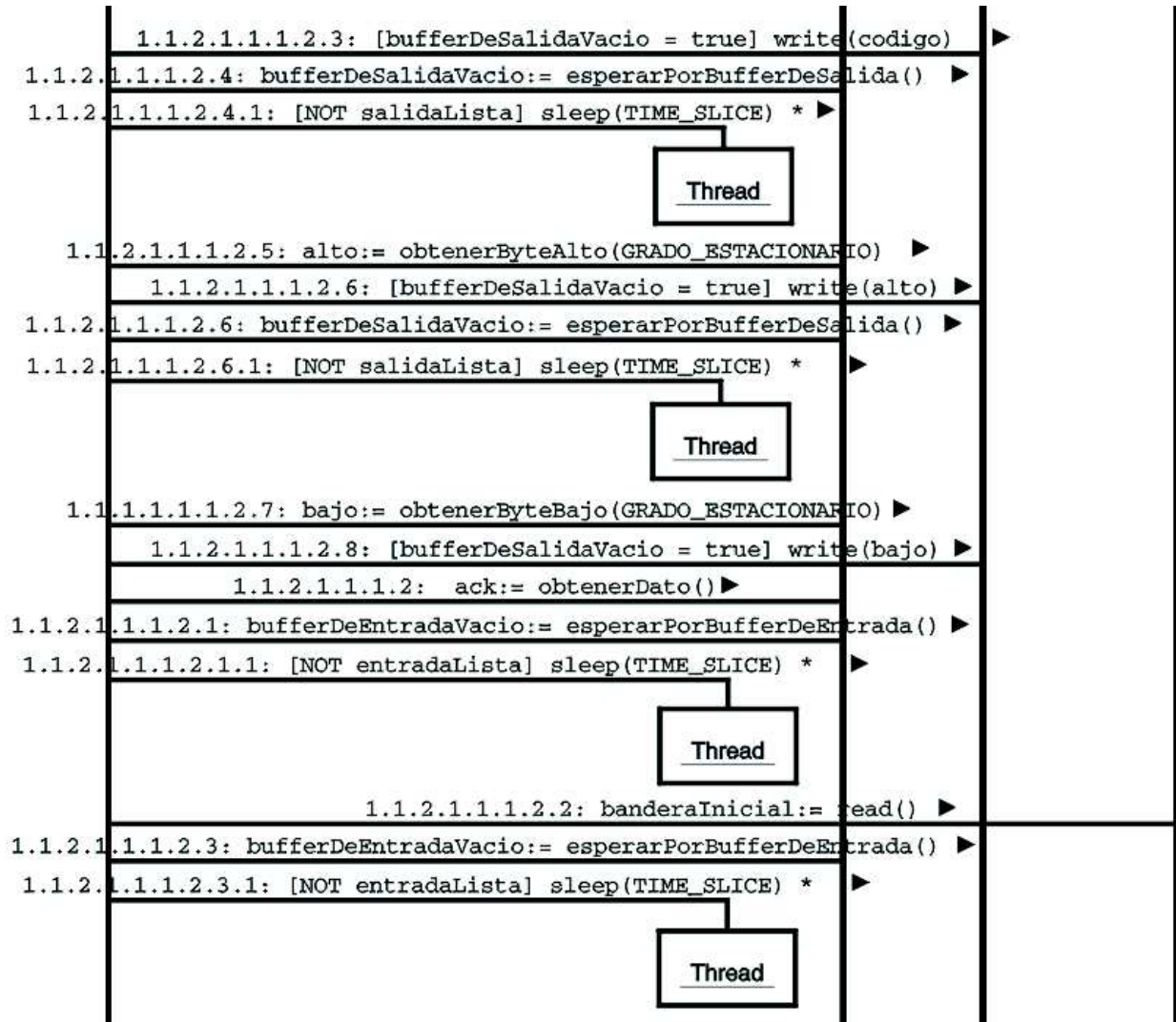


Figura 3-83. Diagrama de Colaboración detener agente inteligente, quinta parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

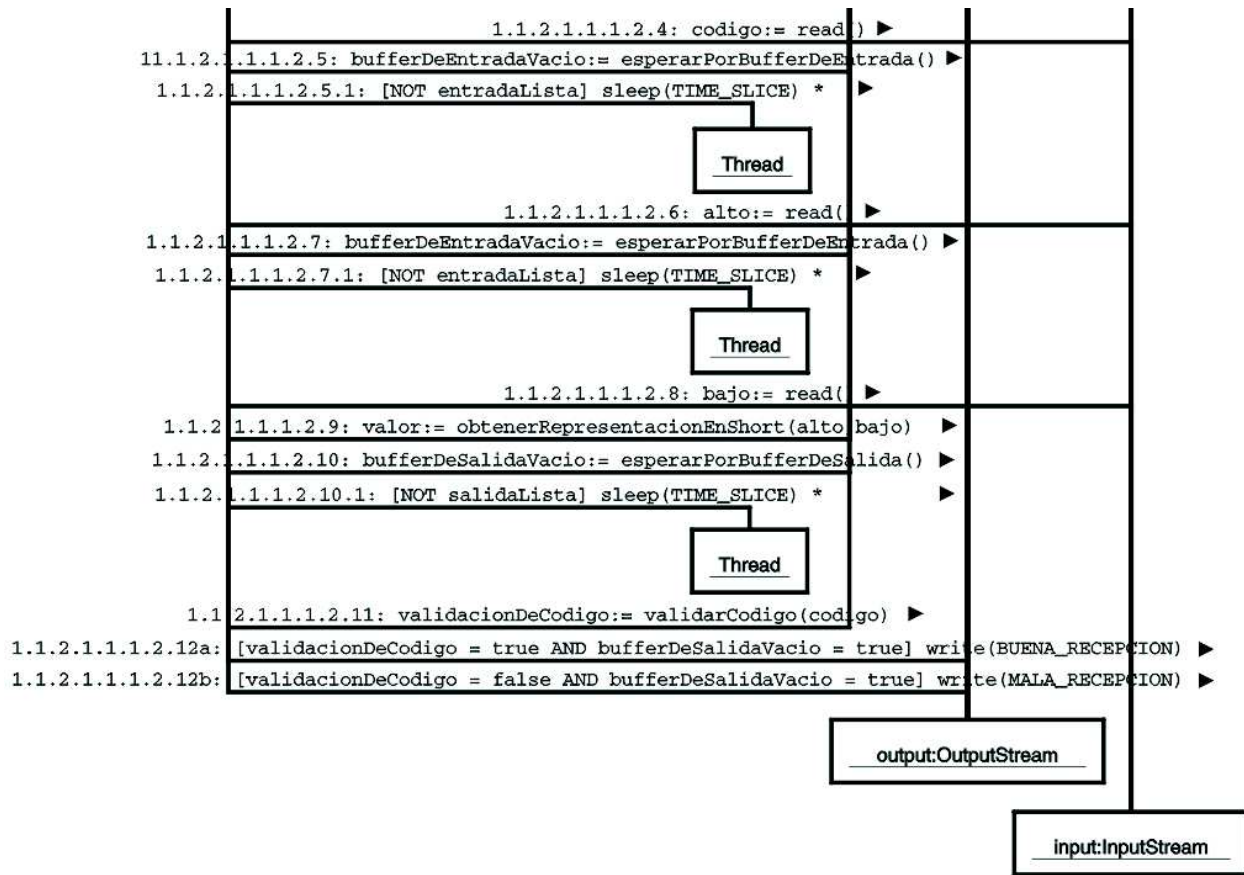


Figura 3-84. Diagrama de Colaboración detener agente inteligente, sexta parte y final.

CAPITULO 3: DESARROLLO DEL PROYECTO

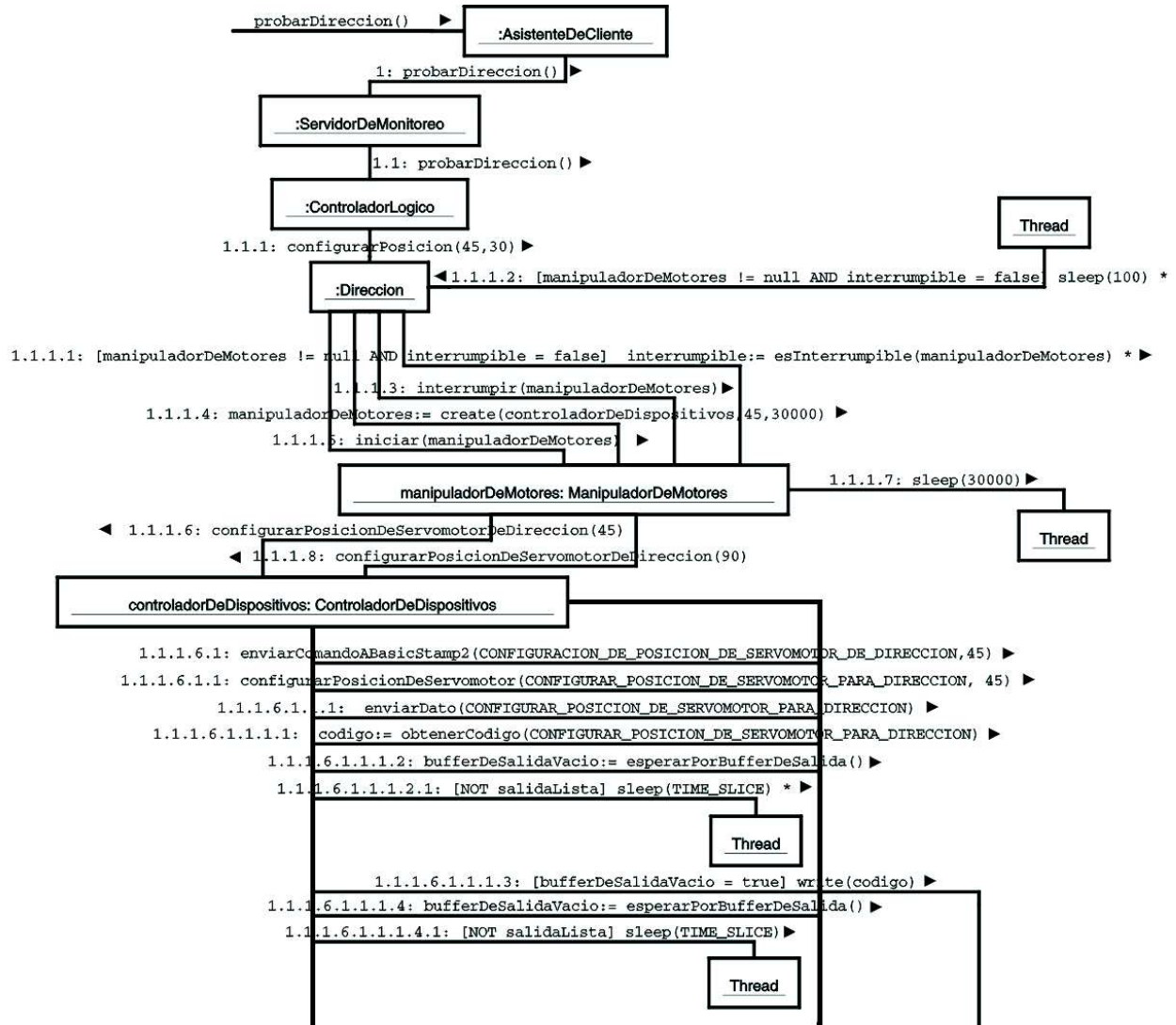


Figura 3-85. Diagrama de Colaboración probar dirección, primera parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

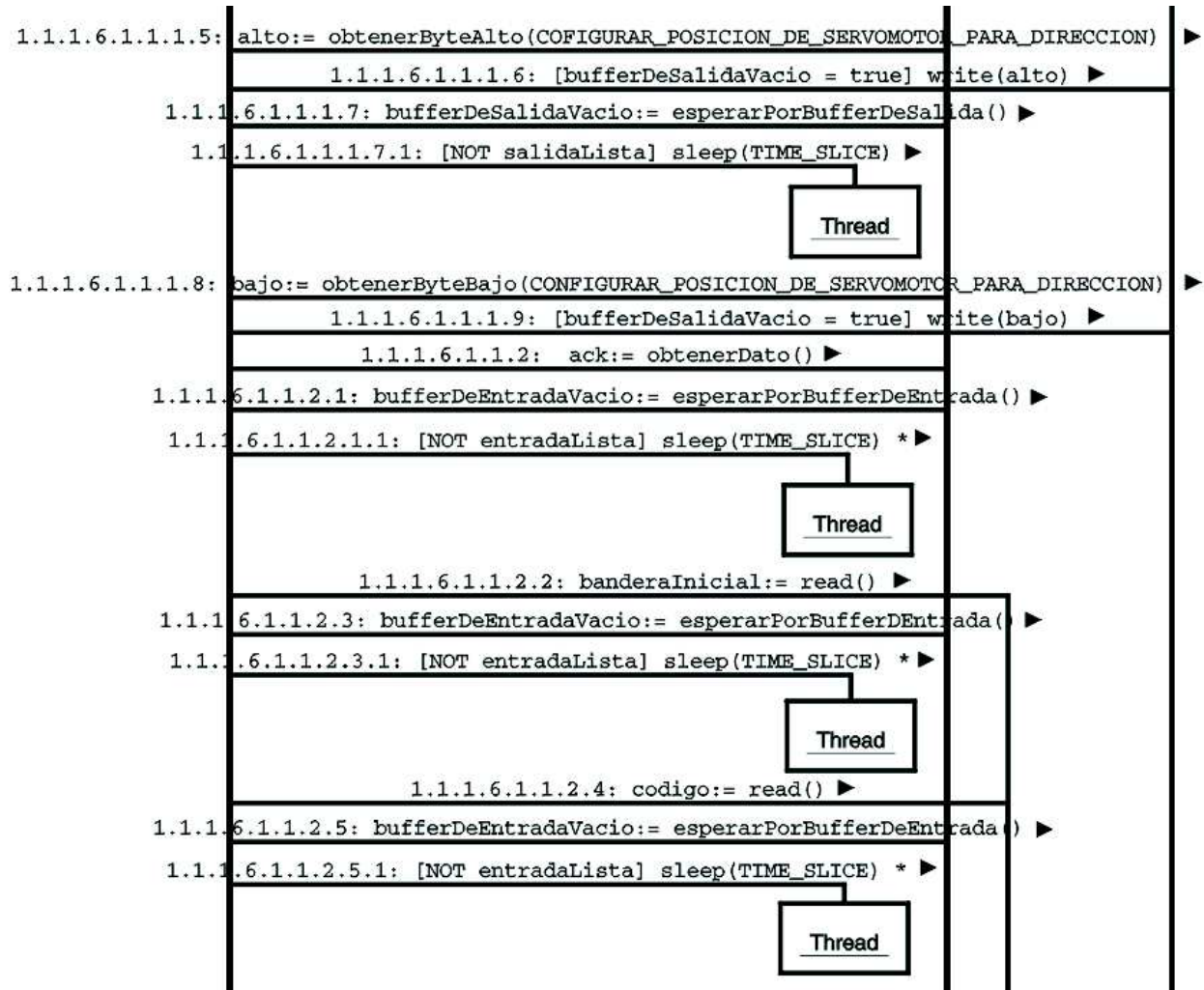


Figura 3-86. Diagrama de Colaboración probar dirección, segunda parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

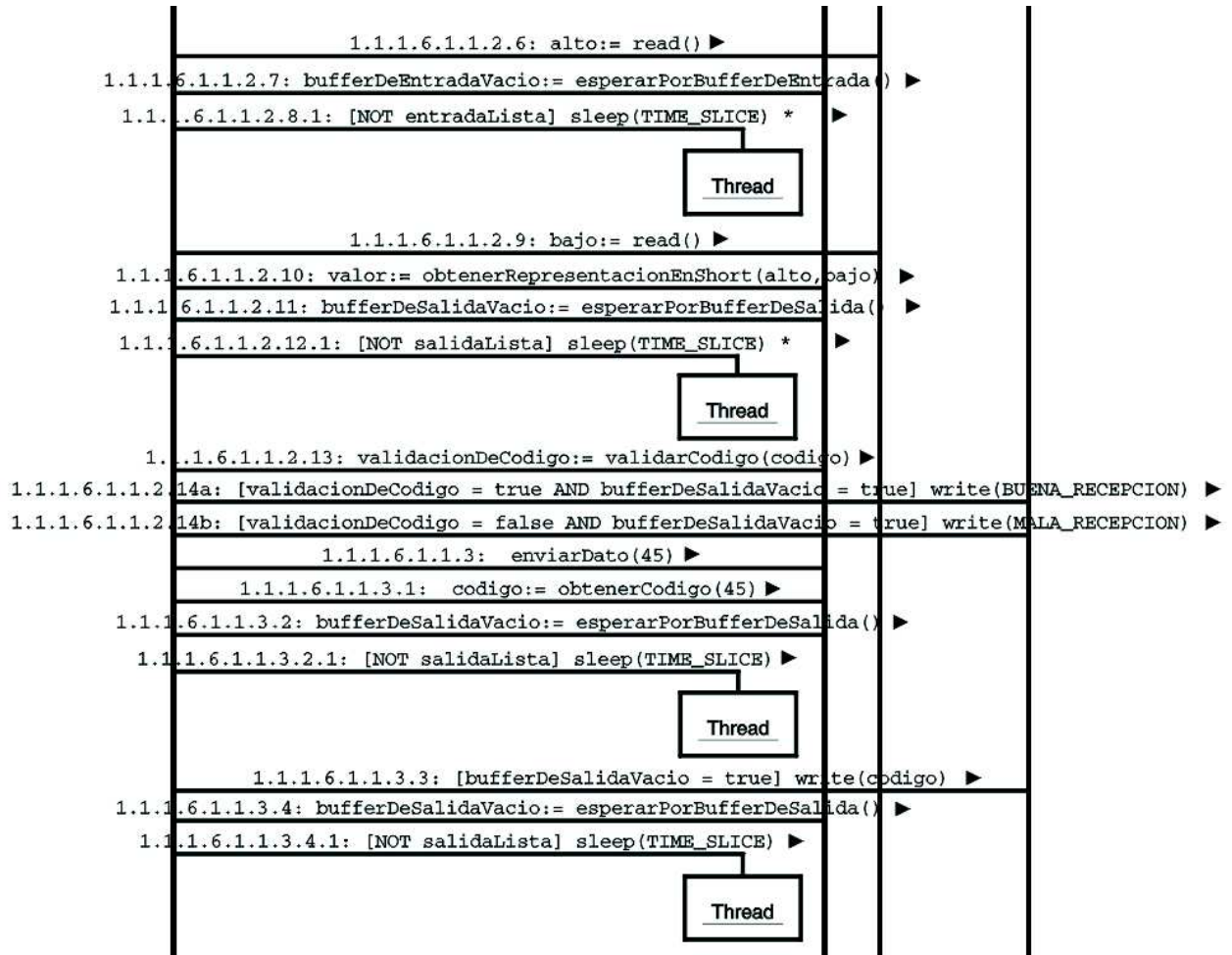


Figura 3-87. Diagrama de Colaboración probar dirección, tercera parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

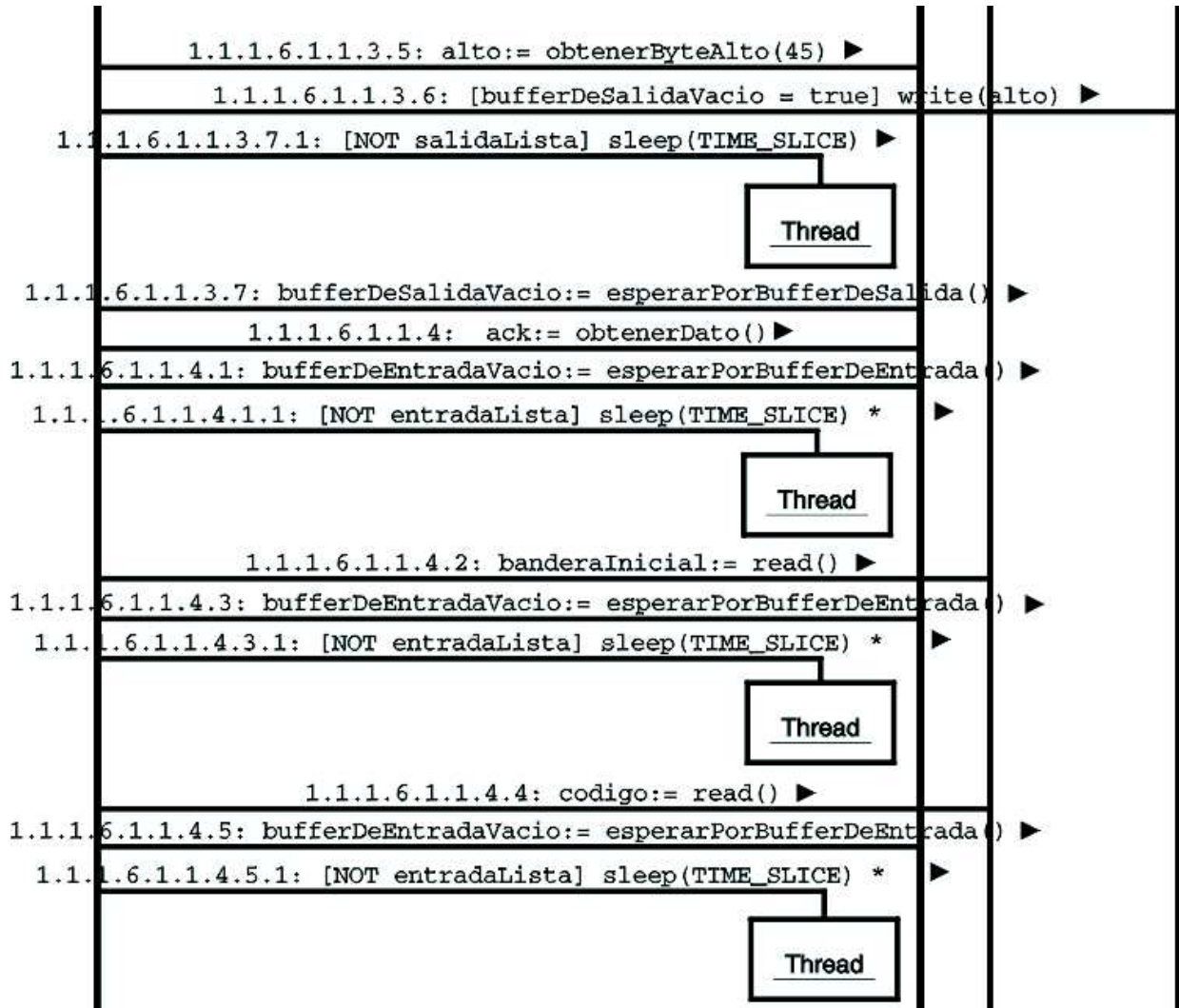


Figura 3-88. Diagrama de Colaboración probar dirección, cuarta parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

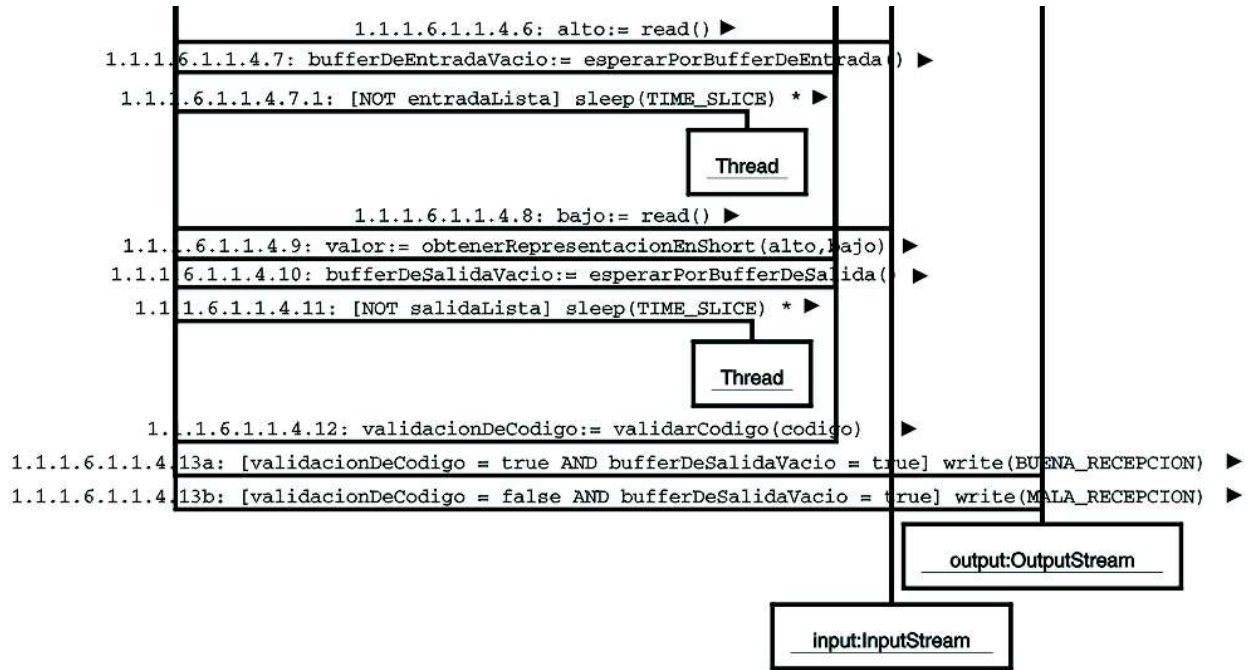


Figura 3-89. Diagrama de Colaboración probar dirección, quinta parte y final.

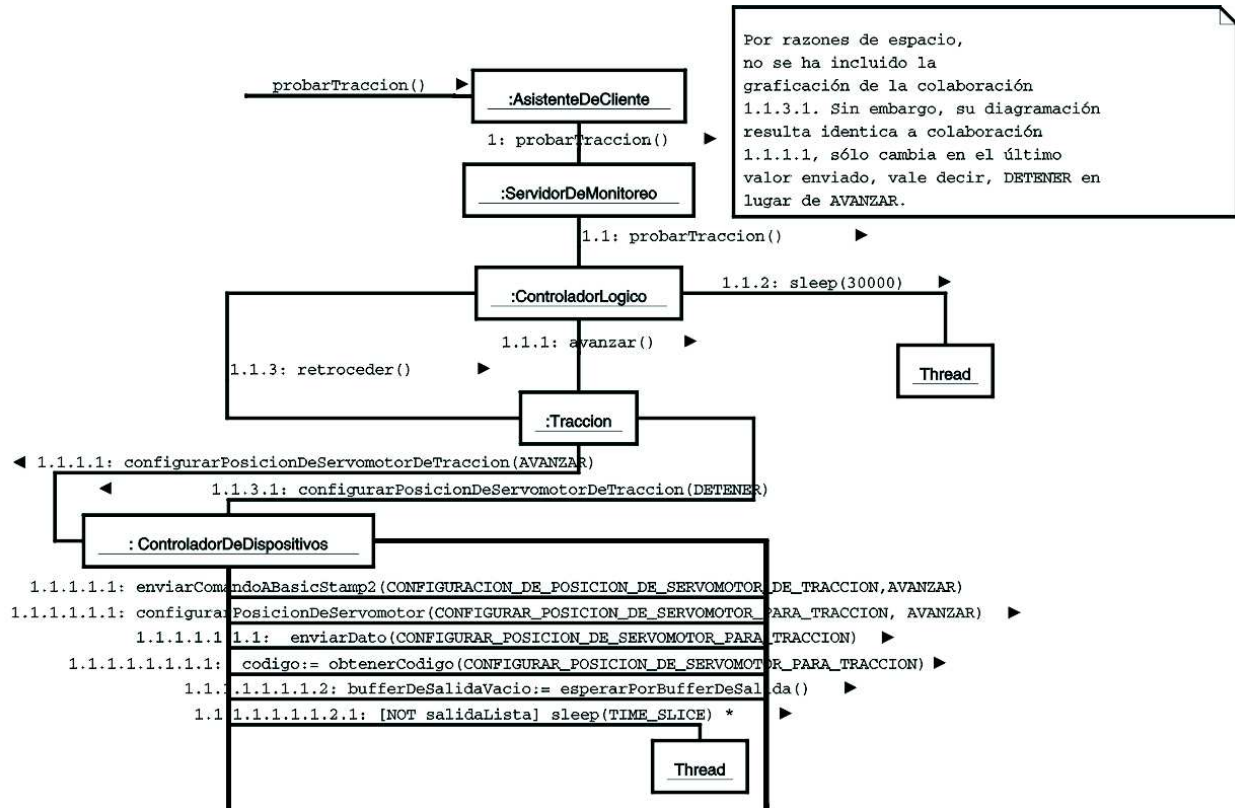


Figura 3-90. Diagrama de Colaboración probar tracción, primera parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

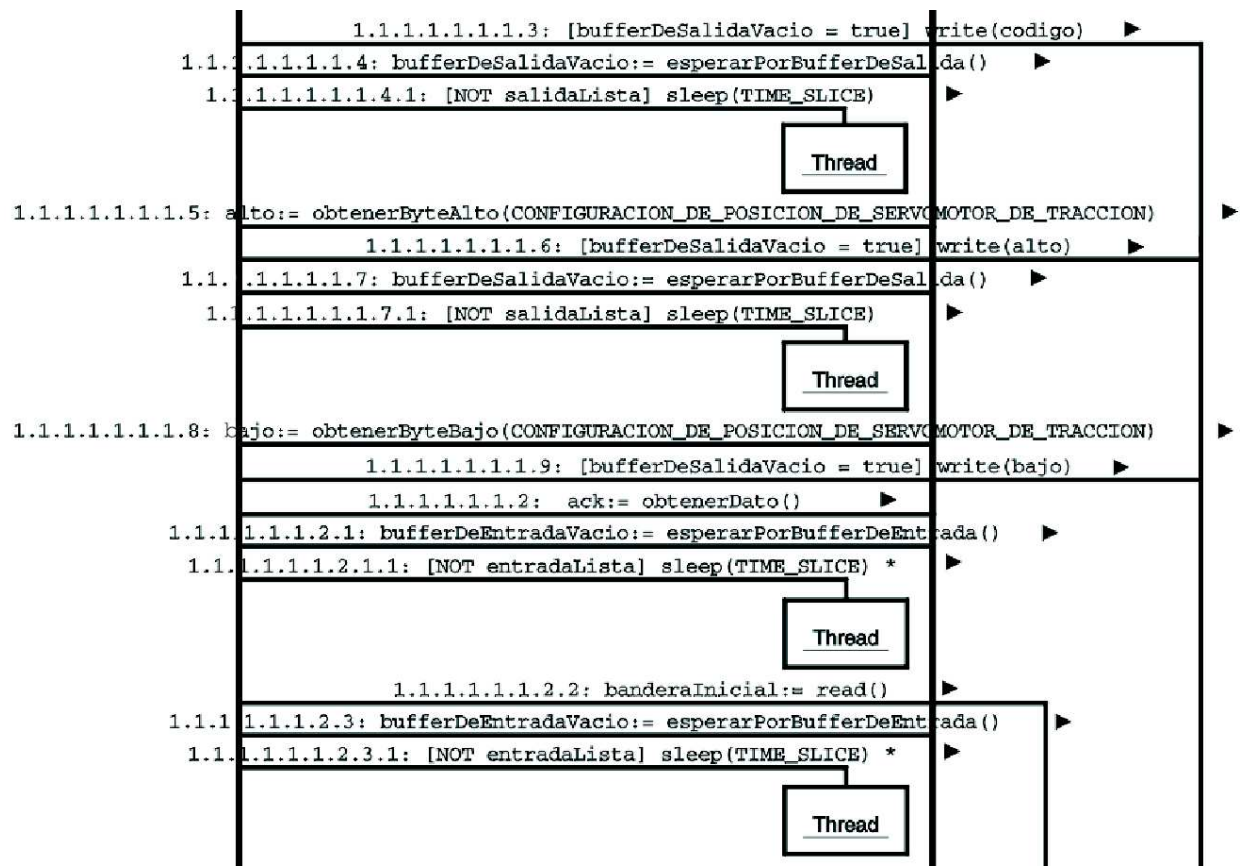


Figura 3-91. Diagrama de Colaboración probar tracción, segunda parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

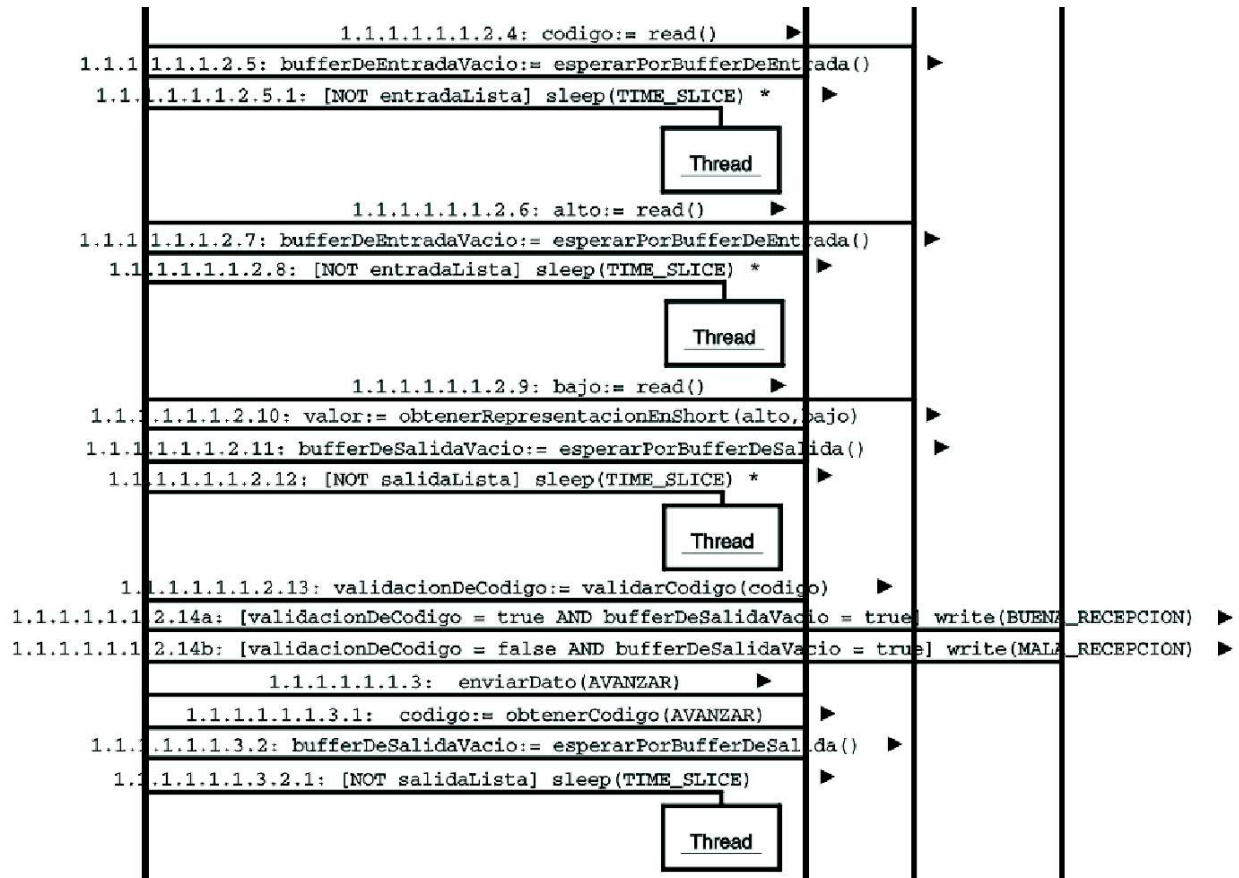


Figura 3-92. Diagrama de Colaboración probar tracción, tercera parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

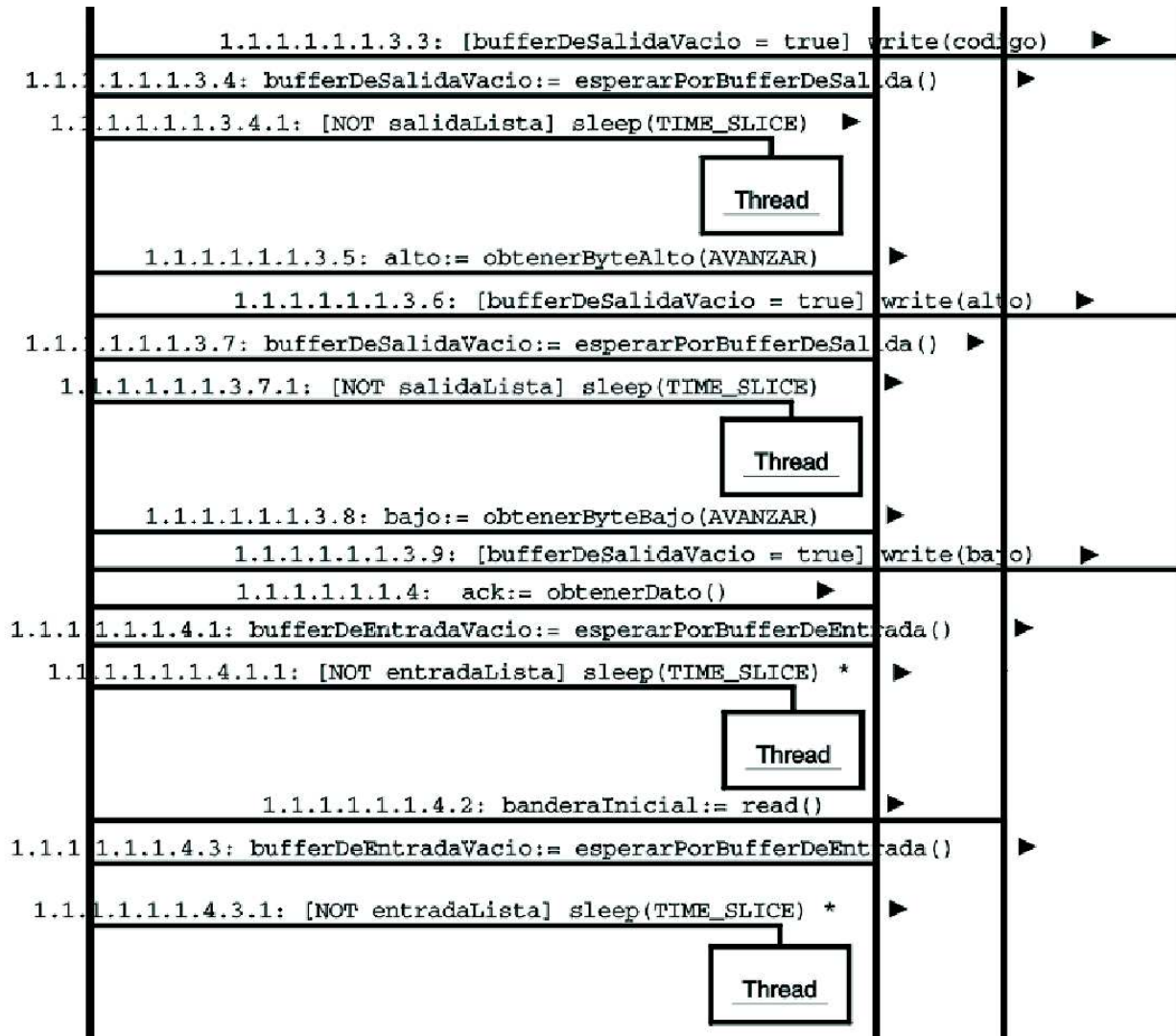


Figura 3-93. Diagrama de Colaboración probar tracción, cuarta parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

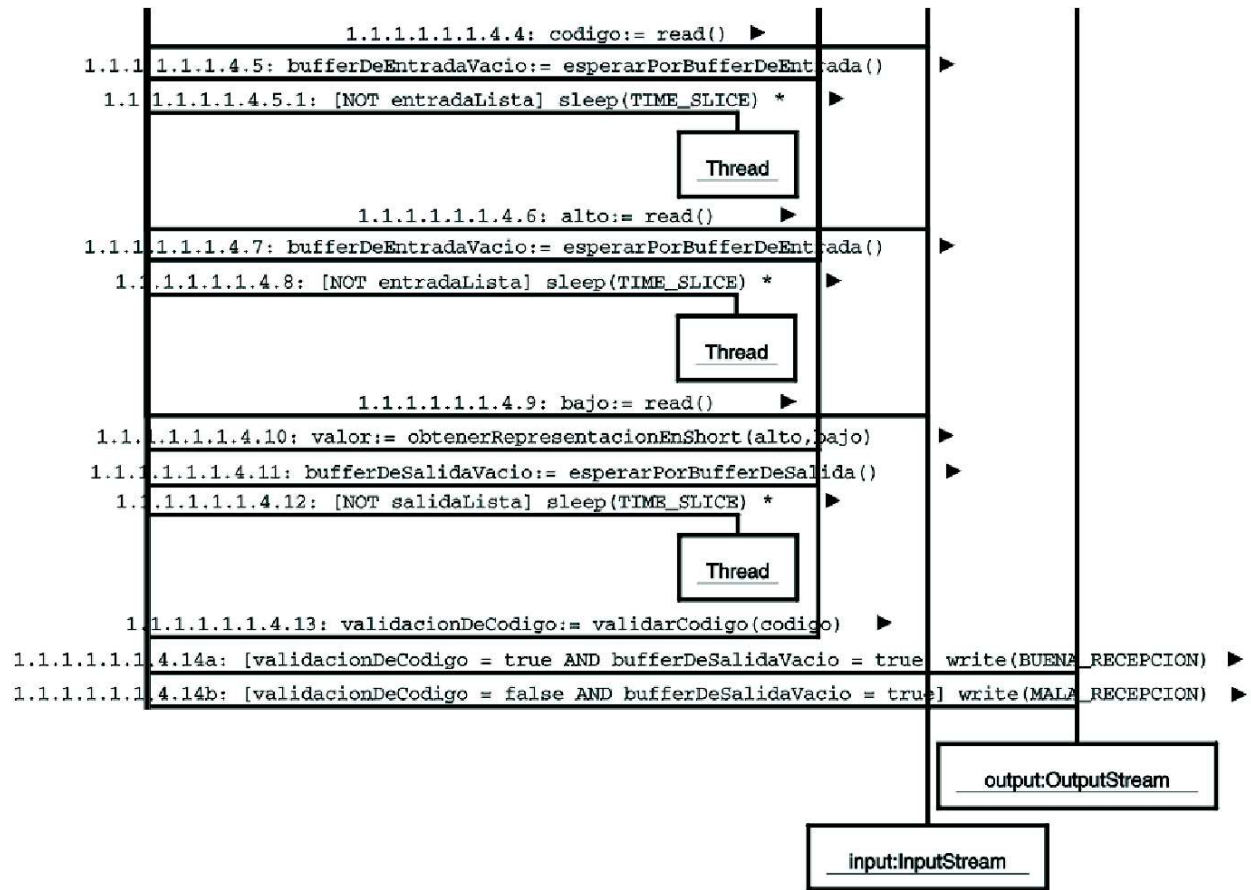


Figura 3-94. Diagrama de Colaboración probar tracción, quinta parte y final.

CAPITULO 3: DESARROLLO DEL PROYECTO

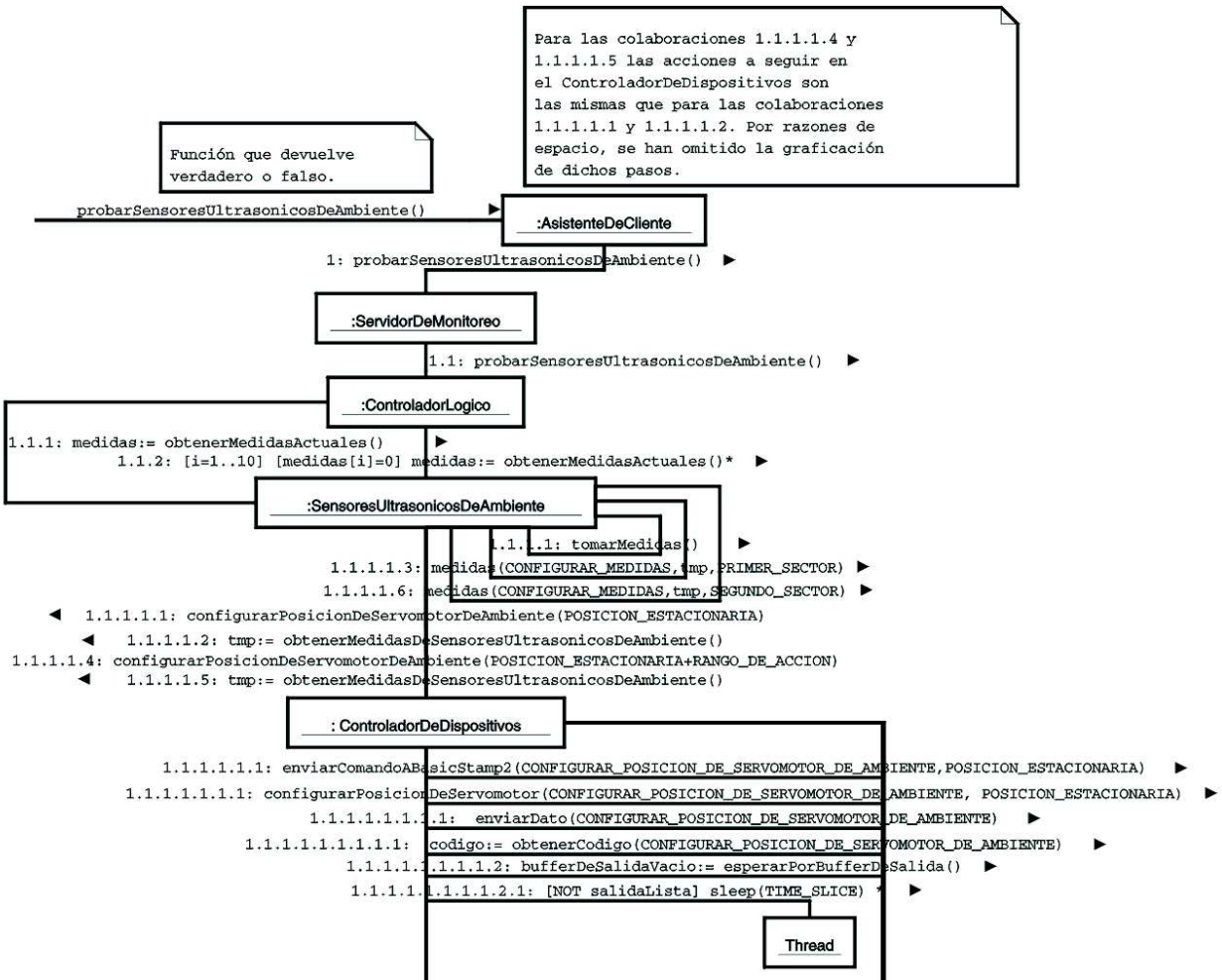


Figura 3-95. Diagrama de Colaboración probar sensores ultrasónicos de ambiente, primera parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

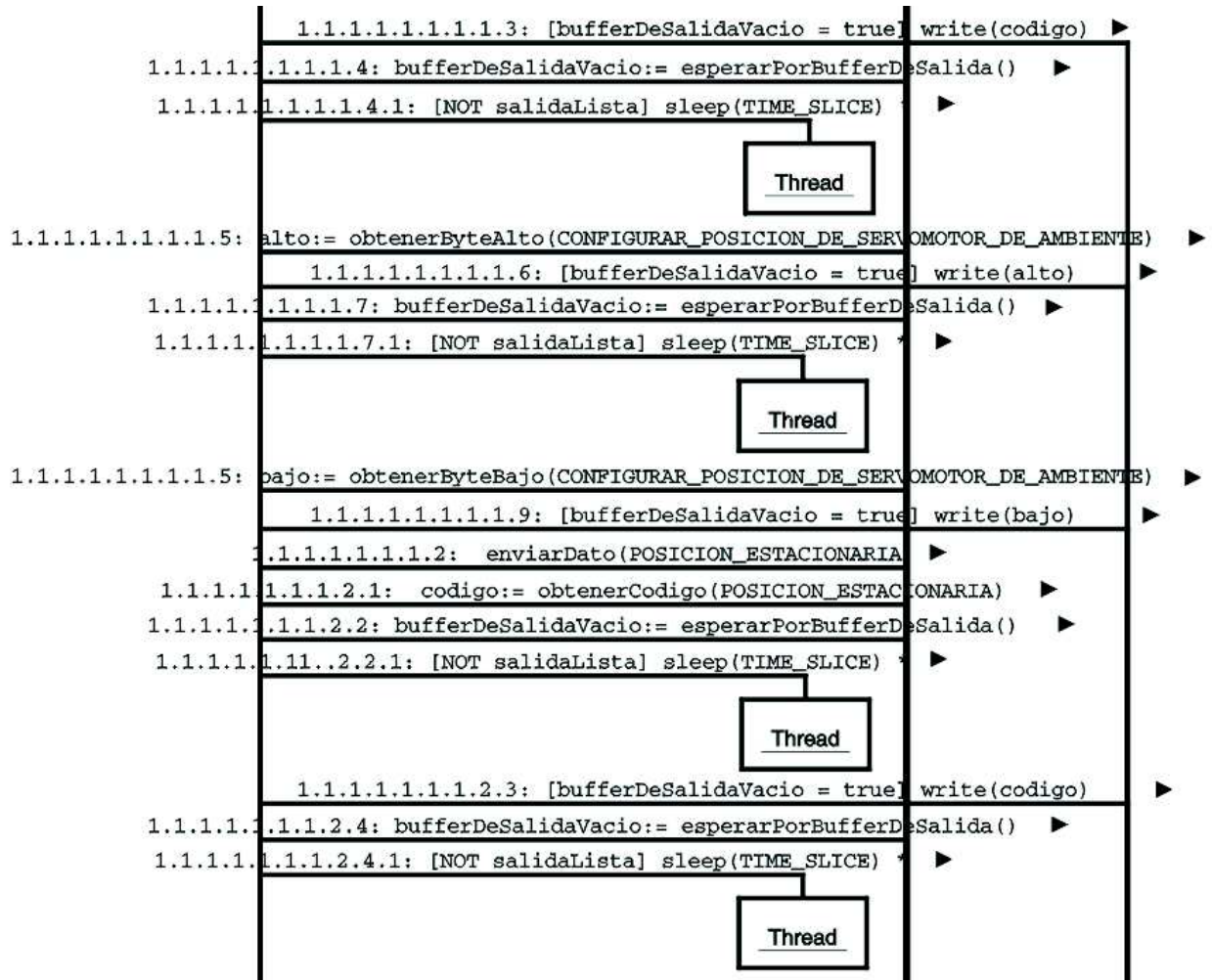


Figura 3-96. Diagrama de Colaboración probar sensores ultrasónicos de ambiente, segunda parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

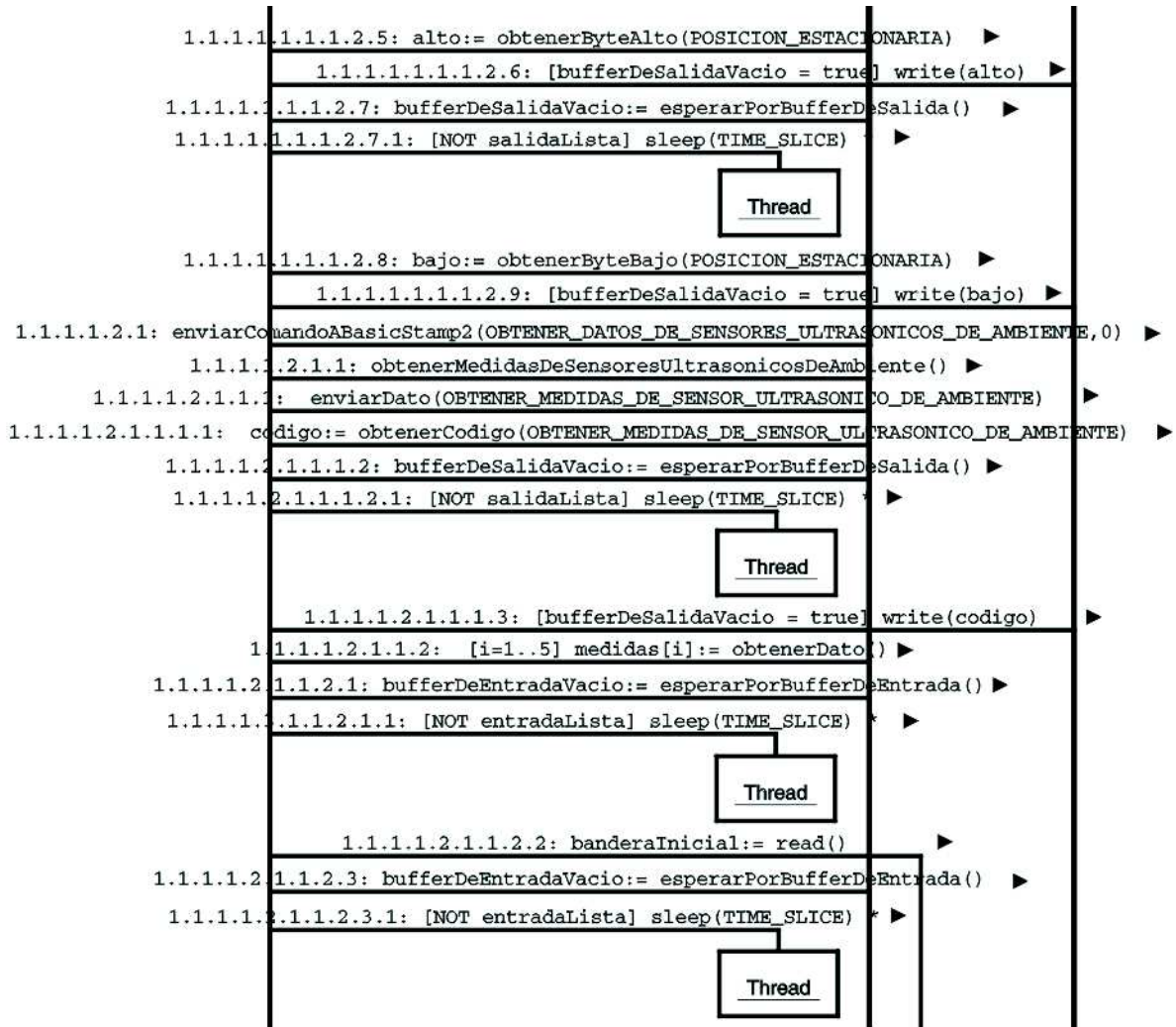


Figura 3-97. Diagrama de Colaboración probar sensores ultrasónicos de ambiente, tercera parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

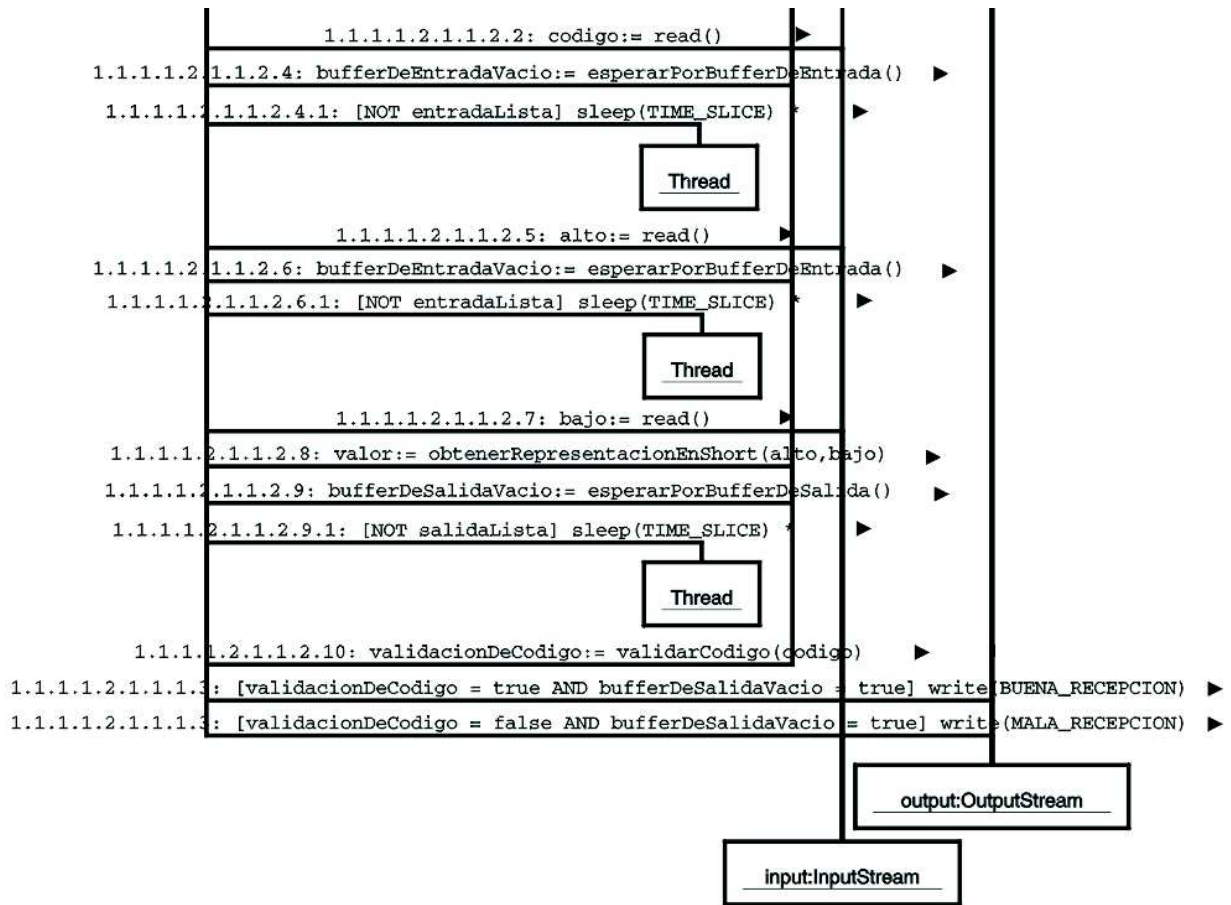


Figura 3-98. Diagrama de Colaboración probar sensores ultrasónicos de ambiente, cuarta parte y final.

CAPITULO 3: DESARROLLO DEL PROYECTO

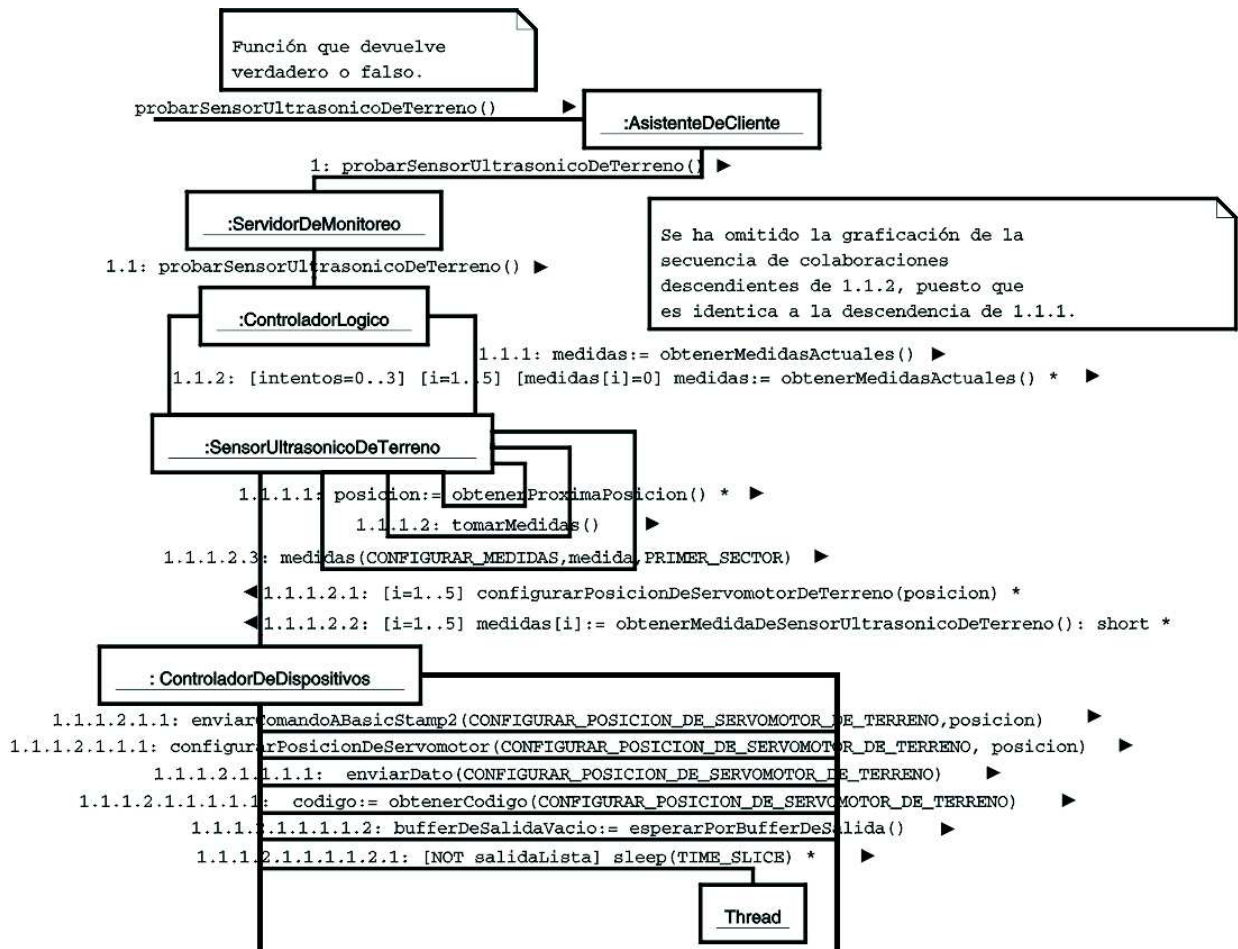


Figura 3-99. Diagrama de Colaboración probar sensor ultrasónico de terreno, primera parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

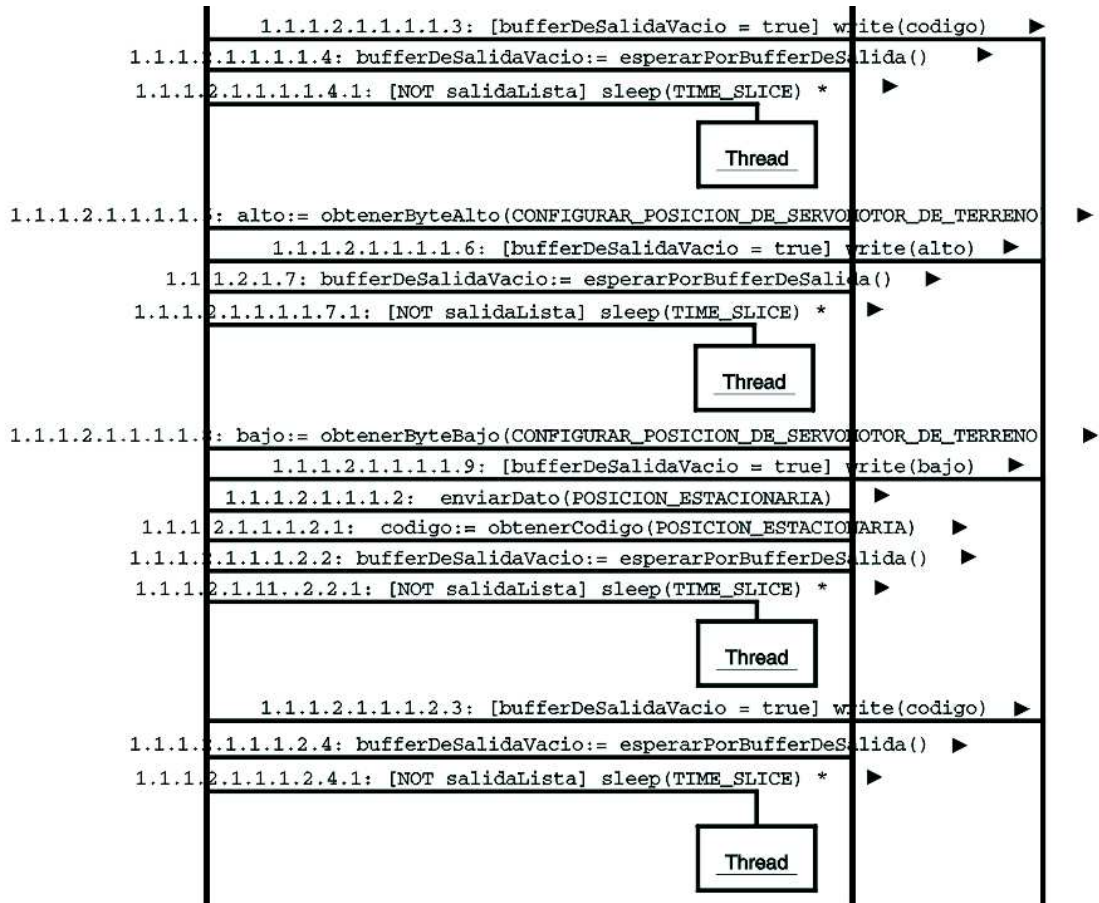


Figura 3-100. Diagrama de Colaboración probar sensor ultrasónico de terreno, segunda parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

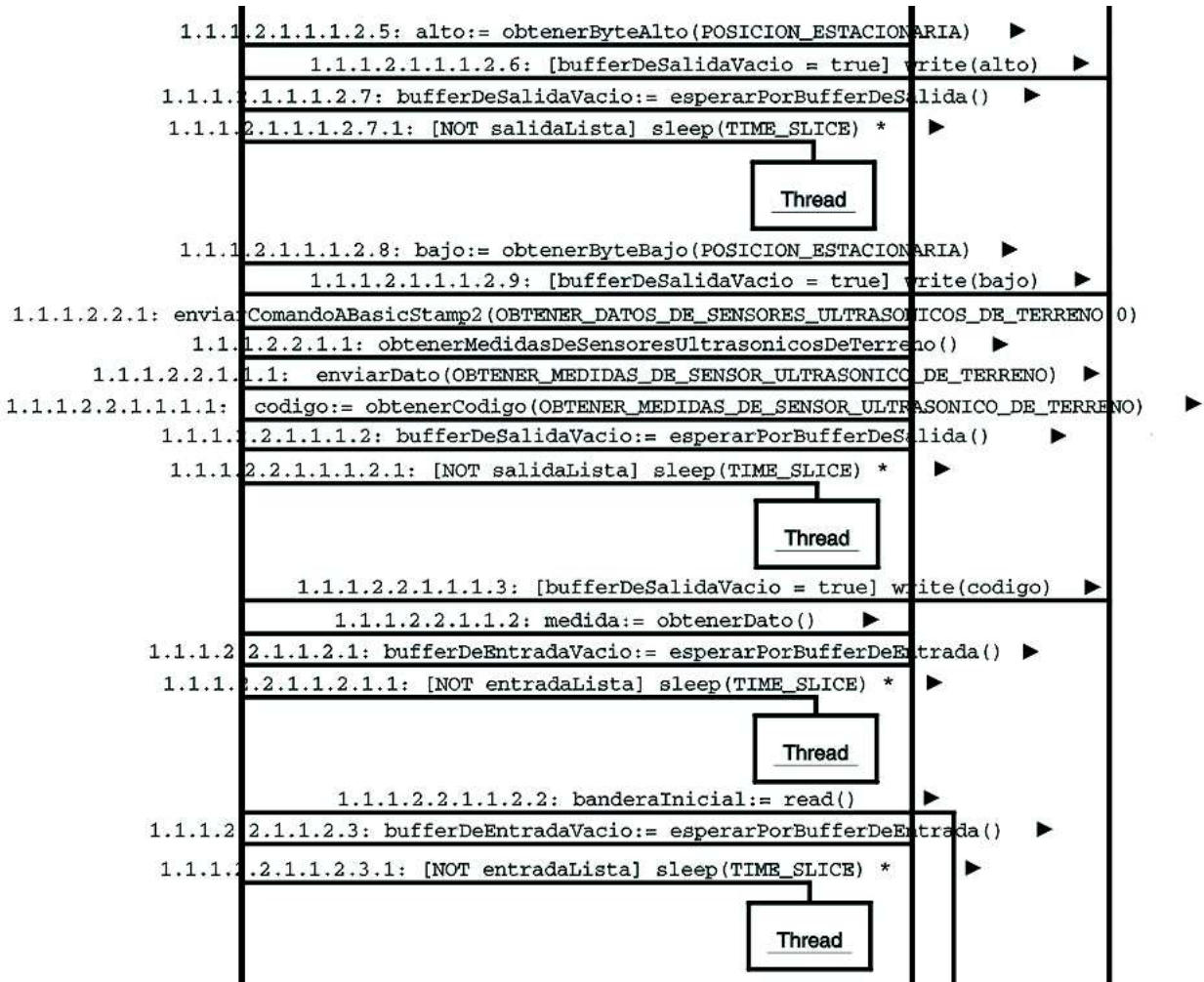


Figura 3-101. Diagrama de Colaboración probar sensor ultrasónico de terreno, tercera parte.

CAPITULO 3: DESARROLLO DEL PROYECTO

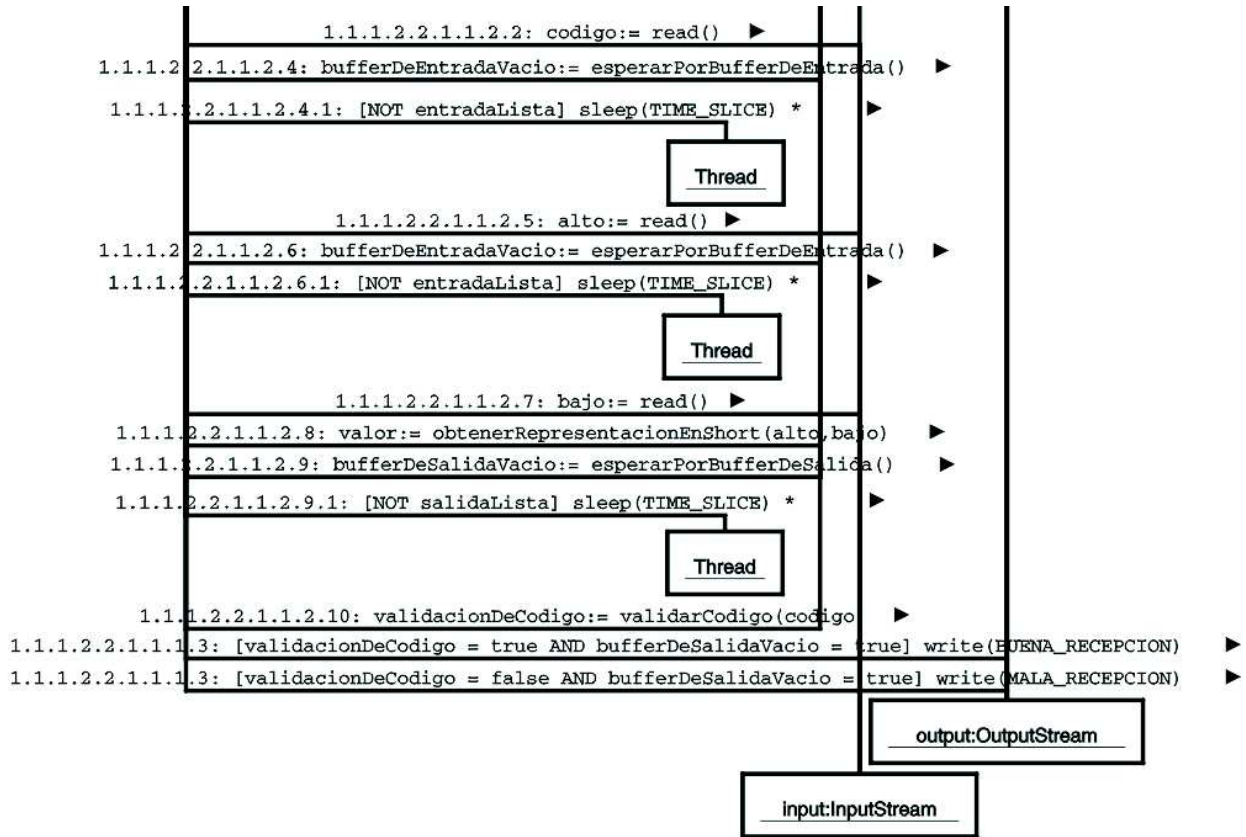


Figura 3-102. Diagrama de Colaboración probar sensor ultrasónico de terreno, cuarta parte y final.

CAPITULO 3: DESARROLLO DEL PROYECTO

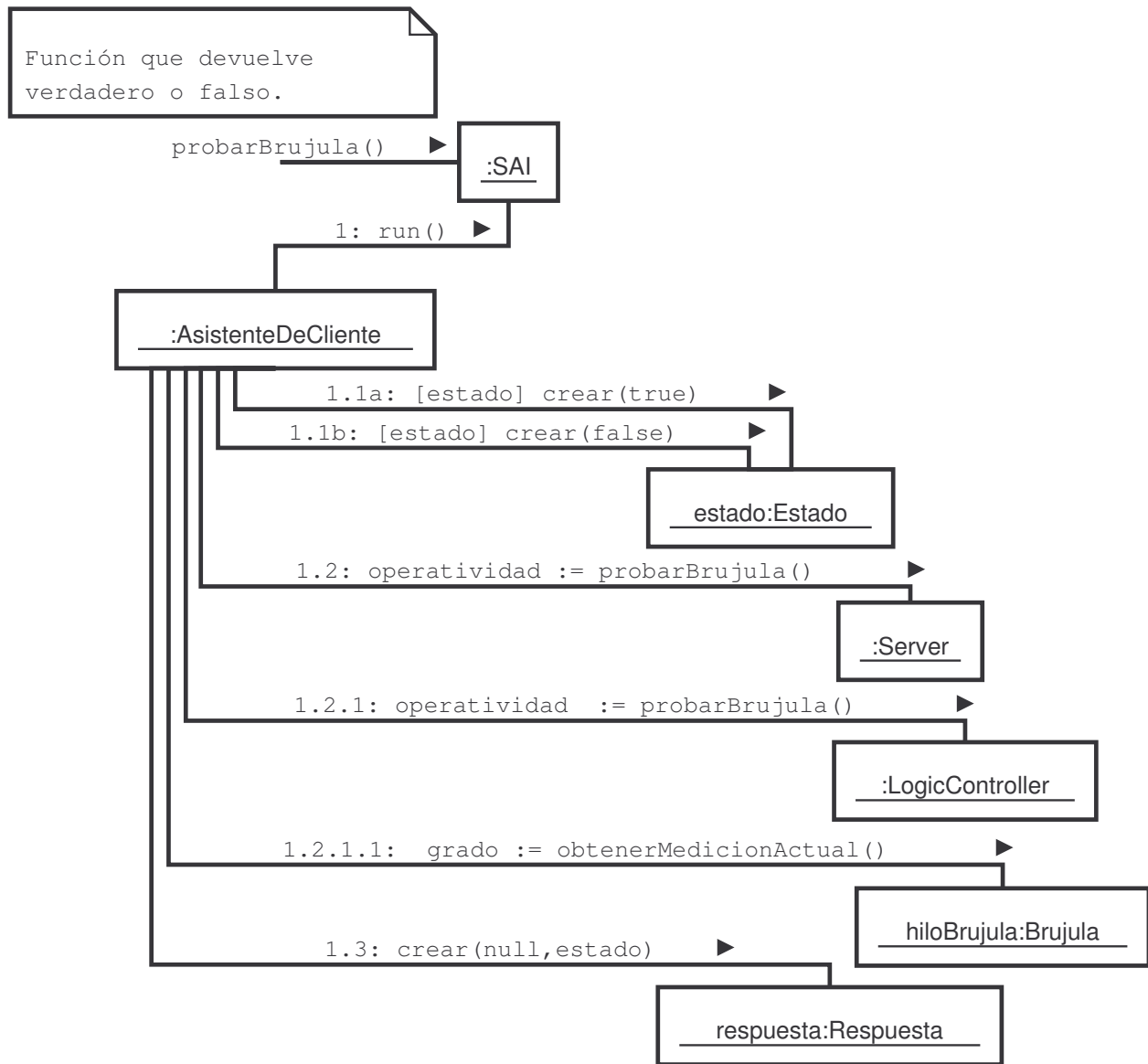


Figura 3-103. Diagrama de Colaboración probar brújula.

CAPITULO 3: DESARROLLO DEL PROYECTO

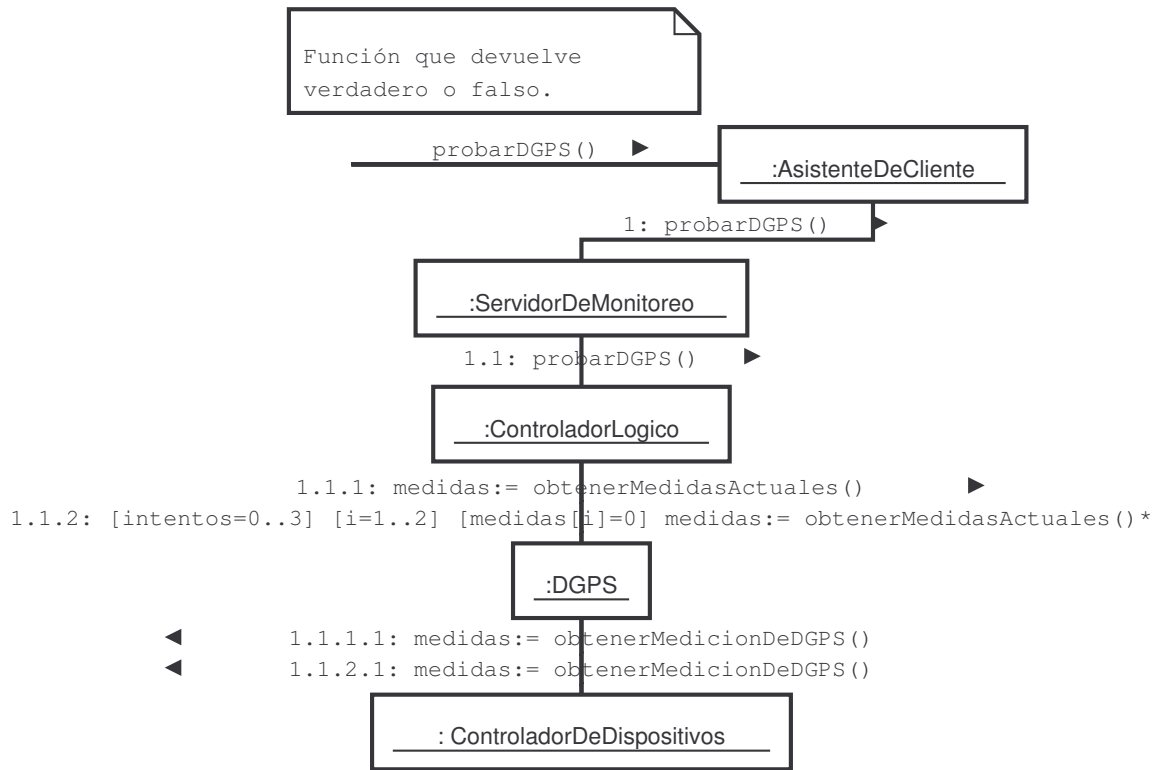


Figura 3-104. Diagrama de Colaboración probar DGPS.

CAPITULO 3: DESARROLLO DEL PROYECTO

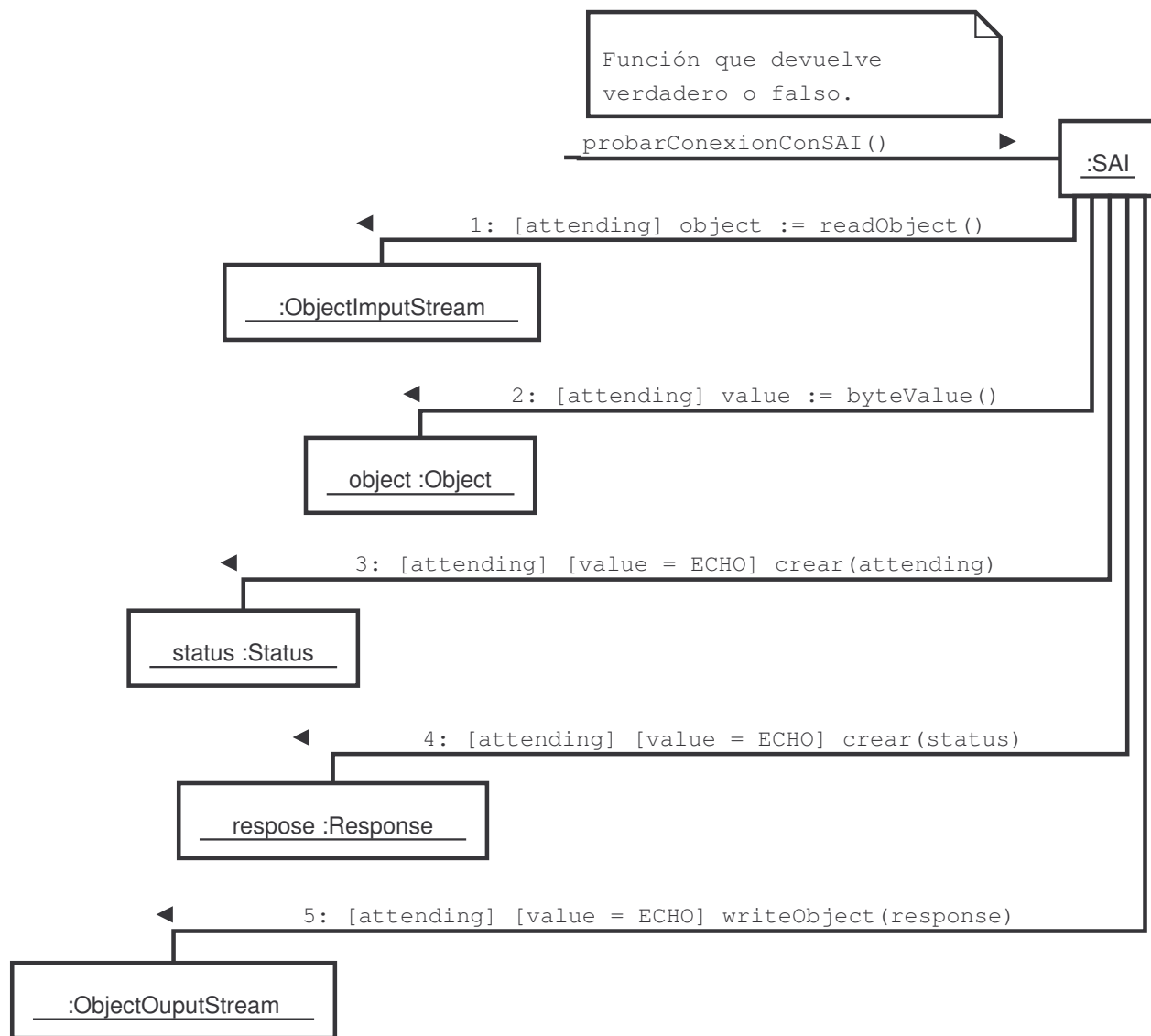


Figura 3-105. Diagrama de Colaboración probar conexión con SAI.

CAPITULO 4: IMPLEMENTACIÓN

4.1 IDE

Se utilizará la IDE libre Eclipse 3.1, de la cual existe conocimiento sobre su funcionamiento aplicado al desarrollo de proyecto de software y pruebas softwares sobre los mismos, en el lenguaje mencionado en la sección 3.4.1. (Eclipse.org, 2006)

4.2 Memoria del Agente Inteligente

Debido a la necesidad de implementar una memoria para el Agente Inteligente, donde registrar los puntos de información, se analizaron varias alternativas.

Como se debía almacenar una estructura arbórea en memoria secundaria, dado el tamaño de ésta, la primera solución fue la serialización de dicha estructura. El problema que a continuación se suscitó, fue la necesidad de paginamiento en el rescate de la estructura arbórea, lo cual resultó ser una complejidad adicional no manejable en el tiempo asignado al proyecto, por lo cual se tuvo que migrar a soluciones que contaran con esta característica. La solución con menor complejidad y que cumplía con las características anteriormente dichas fue la utilización de un mapeador Objetos-Relacional. Esta última fue desechada, por el excesivo tiempo ocupado por esta herramienta en responder una consulta. Finalmente, se escogió mapear directamente la estructura arbórea en una base de datos relacional.

La base de datos escogida debía ser ejecutada en un equipo con sistema operativo Microsoft Windows 98 SE y que poseyera las siguientes cualidades de hardware:

- Procesador Pentium II 400 MHz.
- Memoria RAM de 114 [MB].
- Disco Duro de 2,7 [GB]

CAPITULO 4: IMPLEMENTACIÓN

Además, esta base de datos debía soportar el tipo de dato “*array*”, para lo cual existía un conjunto de tres bases de datos donde se debía hacer la elección. Estas tres bases de datos provistas fueron:

- MySQL.
- Oracle.
- PostgreSQL.

MySQL no da soporte al tipo de dato “*array*”, por lo cual fue descartado. PostgreSQL no soporta sistema de archivos FAT32, por lo cual no era posible ocuparla. Por otra parte, Oracle tiene muy altas exigencias de hardware, por lo que no era posible su ejecución en la máquina antes mencionada.

Finalmente, se decidió montar un servidor de base de datos de forma remota, para lo cual el criterio de selección entre los dos sistemas de base de datos que daban soporte al tipo de dato “*array*” fue la licencia. Por lo tanto, se escogió la base de datos PostgreSQL versión 8.1, de licencia libre, con conexión nativa JDBC3. (PostgreSQL.org, 2006)

4.3 Selección de la Técnica Adecuada para la Inteligencia del Agente Inteligente

Buscando soluciones mediante distintas técnicas, como Bases de Conocimiento, búsqueda en profundidad, redes neuronales, redes bayesianas, se decidió adoptar la técnica de redes neuronales. Dentro de esta técnica, se analizaron diversas herramientas posiblemente aplicables al problema, las cuales fueron: Adaline (Adaptive Linear Elemento), BPN (Back Propagation Network) y BAM (Bidirectional Association Memory). (Freeman y Skapura, 1993)

Dentro de la herramienta Adaline, se llegó a la conclusión de que para la integración de una red de este tipo se requiere de mucho conocimiento empírico, siendo casi un arte, el hecho de determinar la arquitectura más idónea del número de capas de la red y neuronas por capa, la cual no se podría lograr en el tiempo considerado para el proyecto y, por ende, la búsqueda de otra herramienta, que eliminara este factor subjetivo, continuó.

CAPITULO 4: IMPLEMENTACIÓN

La herramienta BPN se ajustaba a los requerimientos del problema, ya que es una herramienta capaz de determinar modelos matemáticos muy complejos mediante la asociación de una salida dado un conjunto de determinadas entradas, lo cual resultaba ideal para el problema que plantea este proyecto. Sin embargo, una vez más existe el factor de la subjetividad al momento de determinar la arquitectura más idónea para el problema planteado.

La herramienta BAM, mencionada en la sección 2.1.1.4, se ajustaba a los requerimientos del problema, además de eliminar la subjetividad de la herramienta de la técnica de Redes Neuronales analizadas anteriormente. Sin embargo, al momento de realizar las pruebas sobre esta herramienta los resultados obtenidos al someterla a varias situaciones del comportamiento que tendrá que resolver el agente inteligente, no fueron del todo satisfactorias, dada la estrategia que ocupa la herramienta para la resolución de problemas (hipercubos de Hamming).

Finalmente, tomando una drástica decisión, se optó por implementar una herramienta que diera solución a los requerimientos del proyecto, a través del rescate de determinadas características de una de las herramienta analizadas anteriormente, la cual fue la que mejor se comportó en los resultados arrojados y que más se ajustaba a los requerimientos del problema (BAM). Dicha herramienta, MABE (Memoria de Asociación de Búsqueda Euclidiana), posee la capacidad de asociar puntos en hiperespacios distintos, los cuales representan situaciones y acciones respectivamente, a través de hechos representados por patrones deducidos de aquellas situaciones almacenadas o, también llamados, conjunto de ejemplares (para más detalles sobre MABE, véase Anexo B).

CAPITULO 4: IMPLEMENTACIÓN

A continuación, desde la Figura 4-1 hasta la Figura 4-3, se mostrará un ejemplo del comportamiento de MABE en un espacio tridimensional:

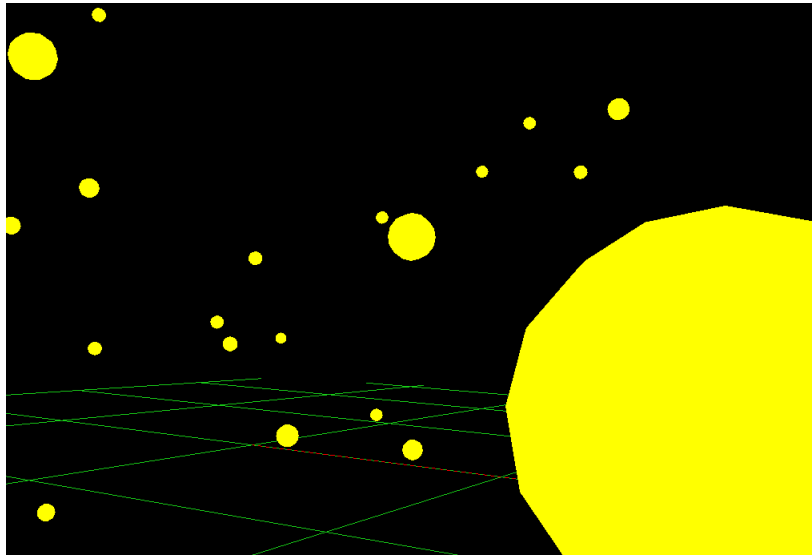


Figura 4-1. Imagen representativa del conjunto de ejemplares almacenados.

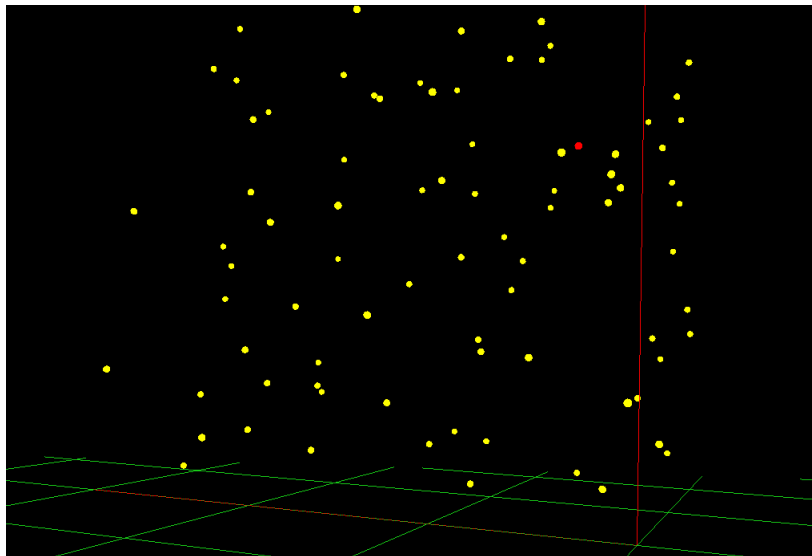


Figura 4-2. Imagen representativa de un punto problema graficado en el espacio de la MABE.

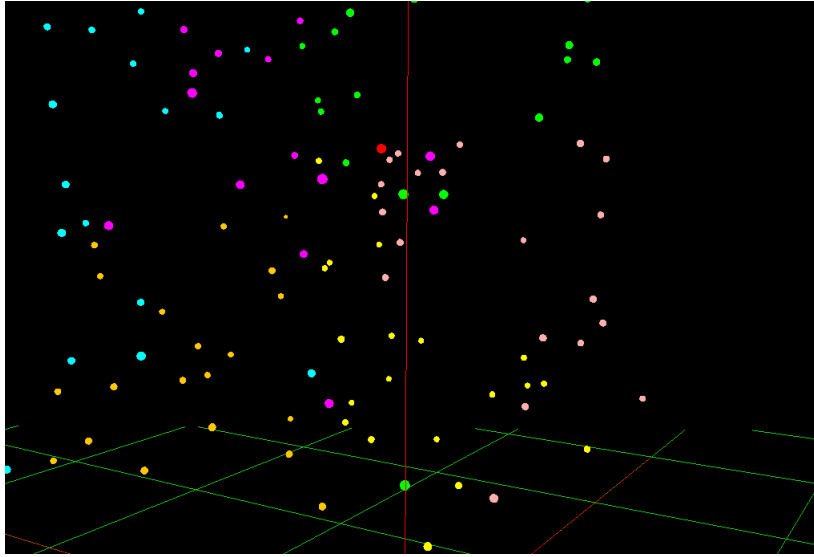


Figura 4-3. Imagen representativa de los distintos conjuntos de puntos pertenecientes a diversos patrones almacenados en la MABE.

Los patrones de la Figura 4-3 son deducidos del ordenamiento menor a mayor de las mediciones obtenidas desde el ambiente del Agente Inteligente.

4.3.1 Ejemplo del Funcionamiento de la MABE

Sea A un Agente Inteligente cualquiera, el cual posee 3 sensores $S (s_1, s_2, s_3)$.

Sean $p_1 = \{50,30,20\}$ y $p_2 = \{20,50,30\}$ puntos de conocimiento pertenecientes al conjunto de ejemplares de la MABE de A .

Sean a_1 y a_2 acciones determinadas.

Sean h_1 y h_2 hechos pertenecientes al conjunto de ejemplares de la MABE de A , donde: $h_1 = p_1 \rightarrow a_1$ y $h_2 = p_2 \rightarrow a_2$.

Sea $m_1 = \{15,45,28\}$ una medición obtenida desde el ambiente de A mediante los sensores S .

CAPITULO 4: IMPLEMENTACIÓN

Para la solución del caso representada por la medición m_1 , el primer paso será la obtención del patrón que representará a dicha medición. Para tal efecto, se procederá a ordenar la medición de los tres sensores S de menor a mayor, de la siguiente forma:

Sea k_1 el patrón resultante del ordenamiento de m_1 , donde $k_1 = \{i_1, i_2, i_3\}$, tal que i_j será el índice correspondiente al sensor que captó el dato en m_1 .

Ordenado de menor a mayor, el patrón de m_1 quedaría: $k_1 = \{1,3,2\}$.

Ahora, la búsqueda de la acción a realizar por parte de A , va a estar dada por el subconjunto seleccionado de los ejemplares contenidos en la MABE que cumplan con el patrón k_1 . Una forma gráfica de mostrar esta situación, se puede ver en la Figura 4-3, la cual representa los subconjuntos de ejemplares agrupados por patrón.

Sea K_1 el subconjunto de ejemplares que cumplen con el patrón k_1 , tal que:

$K_1 = \{h_2\}$, porque $h_2 = p_2 \rightarrow a_2$, donde $p_2 = \{20,50,30\}$, el cual es el único punto de conocimiento de los dos existentes (p_1 y p_2) que cumple con el patrón k_1 .

El segundo paso, una vez obtenido el subconjunto K_1 , es el de determinar la distancia euclidiana más pequeña entre los elementos de este subconjunto y el punto m_1 . Puesto que sólo existe un elemento dentro del subconjunto de ejemplares K_1 , este elemento será la solución más idónea.

Finalmente, dada la situación m_1 , la acción deducida a realizar será a_2 .

Concluyendo, las pruebas realizadas sobre esta herramienta han resultado satisfactorias, dado que permiten el manejo de las decisiones bajo patrones, lo cual da al agente inteligente una mayor flexibilidad. Esto último, está dado por la simplificación del modelo del problema en las distintas variables que lo definen, asociando a un conjunto de instancias de éstas una acción ad-hoc.

CAPITULO 4: IMPLEMENTACIÓN

El problema de esta herramienta es que si se quisiera una respuesta perfecta para cada situación presentada al agente inteligente, se debería realizar un entrenamiento equivalente a la función de permutación del número de variables que definan el problema, vale decir, un ejemplo. En un ambiente donde 10 variables definan el problema, se tendrá que realizar un entrenamiento de ${}_{10}P_{10}^5$, lo cual arroja 3.628.800 patrones de entrenamiento.

Se realizaron pruebas con 7 variables, lo cual corresponde a un entrenamiento acabado de 5.040 patrones. Sin embargo, con un entrenamiento 0.21% del total de patrones, el agente inteligente probado tuvo un comportamiento suficientemente aceptable.

El problema planteado en este proyecto está definido por 17, lo cual implica un entrenamiento acabado de 355.687.428.096.000 patrones.

Lo que se pretende con esta herramienta, no es lograr una solución perfecta, sino una buena solución, lo cual se obtiene con sólo un porcentaje del entrenamiento de patrones (0.2% en promedio, según las pruebas realizadas) y no con la totalidad de ellos.

4.4 Implementación de la BS2

La implementación de la comunicación con los dispositivos periféricos del agente inteligente, por ejemplo sensores y efectores, está a cargo del microcontrolador BS2 (refiérase a la sección 2.1.4).

- Servidor: Para satisfacer los múltiples requerimientos de utilización de sensores y efectores, se programó en este dispositivo un pequeño software de servidor, el cual mediante un código numérico respondería a la realización de una acción por parte de un efector o la toma de mediciones de alguno de los sensores, según corresponda.

A continuación, se muestra el diagrama de estados del servidor del microcontrolador BS2:

⁵ **Permutación:** Es todo arreglo de elementos en donde interesa el lugar o posición que ocupa cada uno de los elementos que constituyen dicho arreglo. La fórmula de la permutación es: $nPr = \frac{n!}{(n-r)!}$.

CAPITULO 4: IMPLEMENTACIÓN

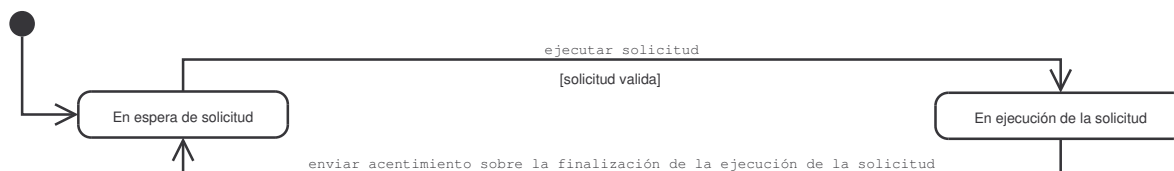


Figura 4-4. Diagrama de Estados del Servidor en el Microcontrolador BS2.

Para mayores referencias a la implementación del servidor en el microcontrolador BS2, refiérase a su código fuente.

- Comunicación vía RS-232: Este protocolo se utiliza para la comunicación entre el computador (cliente) y el microcontrolador BS2 (servidor), permitiendo al cliente enviar peticiones y al servidor responder a éstas. Adicionalmente, en el microcontrolador BS2 y en el cliente fueron implementados un protocolo de más alto nivel, el cual permitió detectar y manipular los errores presentes en las recepciones de los datos.

A continuación, se hablará brevemente de la implementación del protocolo de alto nivel mencionado anteriormente.

La comunicación básica entre el servidor, en el microcontrolador BS2, y el cliente, en el computador donde se encuentra el software del Agente Inteligente, posee el esquema mostrado en la Figura 4-5.

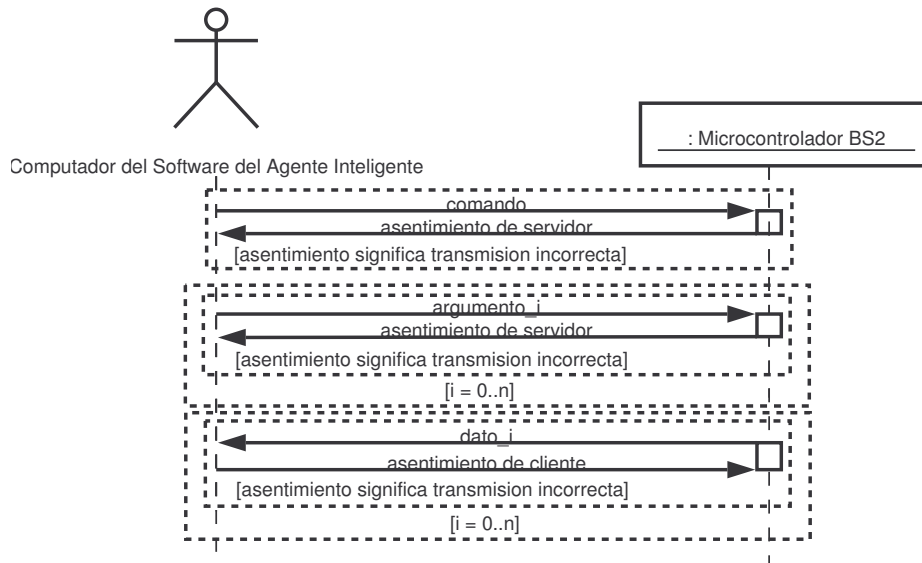


Figura 4-5. Protocolo de comunicación entre el servidor en el microcontrolador BS2 y el cliente en el computador del software del Agente Inteligente.

En la Figura 4-5, cada elemento de ésta se definirá en BNF como sigue:

```

comando ::= <código_hacia_servidor> <comando_para_servidor>
<código_hacia_servidor> ::= 85 | 170
<comando_para_servidor> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
asentimiento_de_servidor ::= 1 | 9
argumento ::= <código_hacia_servidor> <byte>
dato ::= <código_hacia_cliente> <byte>
<código_hacia_cliente> ::= <byte>
<byte> ::= 0 | 1 | 2 | ... | 255
asentimiento_de_cliente ::= 1 | 9
    
```

Tabla 4-1 Definición BNF del protocolo de comunicación entre el servidor en el microcontrolador BS2 y el cliente en el computador del software del agente inteligente.

Dada la definición en la Tabla 4-1, es necesario aclarar ciertos puntos.

El valor de <código_hacia_servidor> está determinado mediante la siguiente función:

$$cod(x) = \begin{cases} 85; x \text{ MOD } 2 = 0 \\ 170; x \text{ MOD } 2 = 1 \end{cases} / x = \text{comando} \vee \text{argumento.}$$

CAPITULO 4: IMPLEMENTACIÓN

...la cual permite determinar el valor del código hacia el servidor, que le permitirá saber a éste si el valor del comando o el argumento que se le transmite es un valor par ($85_{10} = 01010101_2$, con los bits pares en 1) o un valor impar ($170_{10} = 10101010_2$, con los bits impares en 1).

El valor de <código_hacia_cliente> tiene el tamaño de un byte y el valor asociado a cada par de bits de éste se muestra en la Tabla 4-2.

Bits							
7	6	5	4	3	2	1	0
Representa el número de la secuencia de forma correlativa, que va desde 0 hasta 3. ($0_{10} = 00_2$, $1_{10} = 01_2$, $2_{10} = 10_2$, $3_{10} = 11_2$)		Representa la paridad del número a codificar: Par ($2_{10} = 10_2$) o impar ($1_{10} = 01_2$).		Representa si el valor numérico a codificar corresponde a un valor unitario ($1_{10} = 01_2$), de decena ($2_{10} = 10_2$) o centena ($3_{10} = 11_2$).		Representa el número de la secuencia. (0..3)	

Tabla 4-2. Composición del código hacia el cliente desde el servidor en el microcontrolador BS2.

4.5 Sensores Ultrasónicos

De la implementación de los sensores de ultrasonido para el Agente Inteligente, se pudo desprender que existen ciertos casos donde se producen errores en el sensamiento del ambiente. Esto debido a la reflexión que sufre el sonido sobre superficies irregulares. Un ejemplo de esto se puede ver en la imagen de abajo:

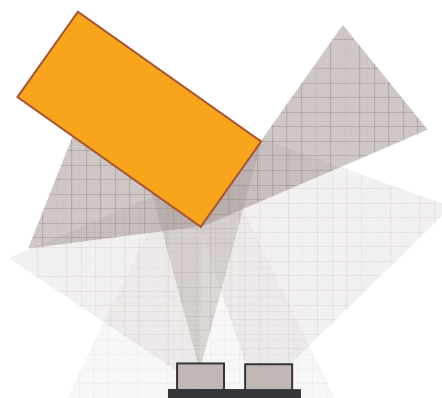


Figura 4-6. Imagen de ejemplo de la reflexión del sonido sobre una superficie irregular.

CAPITULO 4: IMPLEMENTACIÓN

Como se ve en la imagen, la reflexión del sonido en una superficie irregular se dispersa en el ambiente, pudiendo tener, así, diversas instancias del mismo impulso sónico de medición en el lapso de tiempo asignado por el hardware del sensor para la recepción de éstos, lo cual resultará en una medición con un factor de error que dependerá del ángulo de incidencia en la superficie irregular.

Ya que los sensores ultrasónicos generan un cono sónico donde se producirán las reflexiones para calcular la distancia, se realizaron pruebas con el objetivo de determinar el ángulo efectivo de este cono. Finalmente, el resultado obtenido de estas pruebas fue que el cono posee una abertura de 18° en su volumen, teniendo una altura máxima de efectividad de 2,5 m. aproximadamente.

Otras pruebas de mediciones realizadas sobre este tipo de sensores, arrojaron que sobre superficies esponjosas o de muchas capas de materiales flexibles, el sonido es absorbido casi en su totalidad, lo que distorsiona la medición de forma significativa.

Los sensores de ultrasonido de ambiente en el cuerpo del Agente Inteligente estarán dispuestos de forma estratégica, de tal manera que ninguno se sobreponga a otro, pero consecutivos en su disposición, de manera que 5 sensores completarán 90° de medición del entorno.

El sensor de ultrasonido de terreno en el cuerpo del Agente Inteligente estará dispuesto de tal forma que realice 5 mediciones en frente del cuerpo del Agente Inteligente completando, así, la longitud del arco que describe este frente.

4.6 Brújula

Para la implementación de este dispositivo, se realizaron pruebas, las cuales arrojaron que el comportamiento de este dispositivo se ve fuertemente afectado por campos electromagnéticos cercanos, que no necesariamente son de gran intensidad.

4.7 DGPS

Debido a que no se cuenta con conocimientos para lograr implementar técnicas como estaciones de radio, se decidió la utilización de la tecnología de DGPS. Ésta,

CAPITULO 4: IMPLEMENTACIÓN

proporciona una mayor precisión (hasta 1 [m] de error en sectores despejados) en comparación a la tecnología GPS (hasta 15 [m] de error en sectores despejados), aunque menor que un GPS Geodésico (hasta 1 [cm] de error en sectores despejados). Todas estas tecnologías resultan prohibitivas para el proyecto, dado el alto costo. Si bien es cierto, la tecnología GPS es prohibitiva, se podrá utilizar, ya que es proporcionada de forma voluntaria por una institución agrícola nacional de la zona. Por tanto, la tecnología de DGPS resulta absolutamente adecuada para el proyecto en cuanto a costo y precisión ofrecida.

La tecnología de DGPS proporciona, además de la capacidad de generar un concepto de ubicación espacial, el concepto de orientación en el espacio, es decir, la capacidad de saber si se está orientado hacia el Norte, Sur, Este u Oeste, o algún otro grado intermedio. Sin embargo, dado que en el ambiente no controlado existen demasiadas variables no manejables, como se verá en la Tabla 4-3, la tecnología DGPS tiene comportamientos impredecibles en cuanto a la entrega de información, de manera que la información de la ubicación espacial y orientación perdería continuidad, lo cual podría afectar el comportamiento del autómata. Para minimizar dicho efecto, se decidió incorporar la tecnología de compás electrónico, la cual permitiría conocer en tiempo real y sin perder continuidad la orientación física del autómata, dejando este ruido ambiental sólo a la ubicación espacial y no a la orientación en el espacio.

Dentro del trabajo con este dispositivo, se realizaron pruebas de mediciones con éste en estado estático, las cuales arrojaron los siguientes resultados:

<i>Mediciones estáticas con dispositivo DGPS.</i>		
<i>En lugares cerrados.</i>	<i>En lugares abiertos con edificaciones.</i>	<i>En lugares abiertos .</i>
<i>En lugares interiores (habitaciones en general) no se recibe señal GPS, por lo que no se pueden obtener mediciones del dispositivo DGPS.</i>	<i>En lugares abiertos, pero donde existen construcciones alrededor o árboles que encierren dicho lugar, existe pérdida parcial de la señal GPS, lo cual afecta en gran medida la exactitud de la medición arrojada, existiendo casos en que se pierde la totalidad de la señal GPS.</i>	<i>En lugares abiertos, la señal GPS es recepcionada sin problemas, aún cuando exista nubosidad, lo cual permite una fuente de datos continua para el Agente Inteligente.</i>

Tabla 4-3. Resultados de mediciones realizados con dispositivo DGPS en distintos ambientes.

Finalmente, el dispositivo DGPS se probó en movimiento, resultando su comportamiento muy similar al estado estático antes mencionado.

4.8 Estructura Física del Agente Inteligente

La concepción de la estructura física del agente inteligente fue modelada en un software libre de diseño CAD, llamado QCAD (RibbonSoft, 2006). En la Figura 4-7 se aprecia la concepción y posterior implementación de éste.

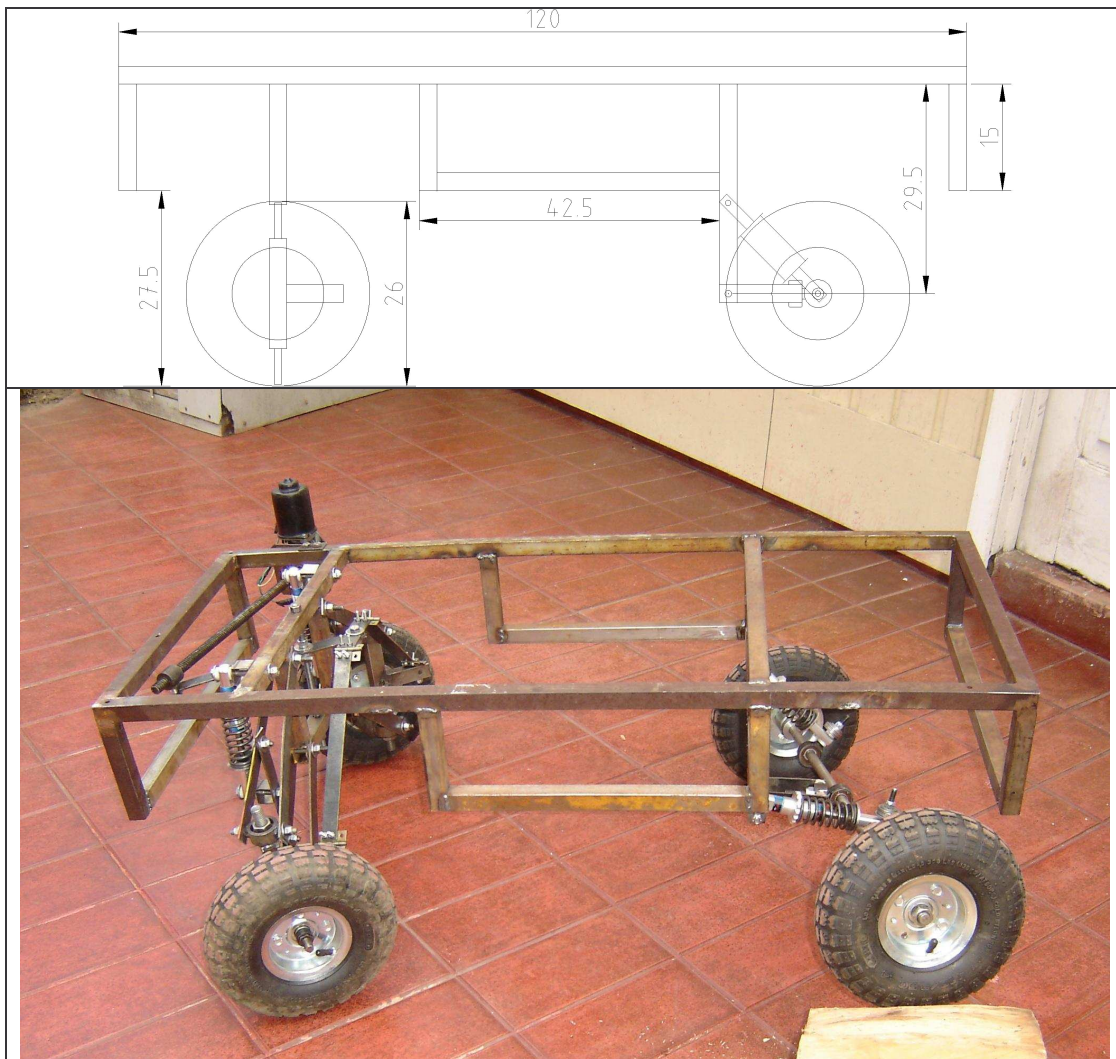


Figura 4-7. Modelo CAD de la estructura física del Agente Inteligente y su implementación.

CAPITULO 4: IMPLEMENTACIÓN

Los materiales ocupados fueron los que se muestran en la siguiente tabla:

Perfil cuadrado tubular, de 1 pulgada.
Barra plana de 1 pulgada.
Tuercas de seguridad y pernos de $\frac{1}{4}$, $\frac{5}{16}$, $\frac{3}{8}$.
4 Bisagras de 2,5 cm.
Hilo sin fin de $\frac{3}{8}$.
Rodamientos de 16 mm.
Barra de acero de 65 cm de longitud y 16 mm de diámetro.

Tabla 4-4. Materiales para la implementación de la estructura física.

Todos estos elementos fueron reflejados fielmente en el software de diseño QCAD.

La estructura física del agente inteligente fue unida a través de soldadura de tipo MIG, la cual permitió un acabado de mayor fortaleza y calidad.

La tracción del vehículo fue implementada mediante un motor de 24 V obtenido desde un scooter, el cual tiene una capacidad de tracción de 80 Kg. Para la adaptación del motor de scooter, fue necesaria la creación de piezas que fijaran el engrane de tracción del motor al eje trasero del vehículo, como se ve en la Figura 4-8.

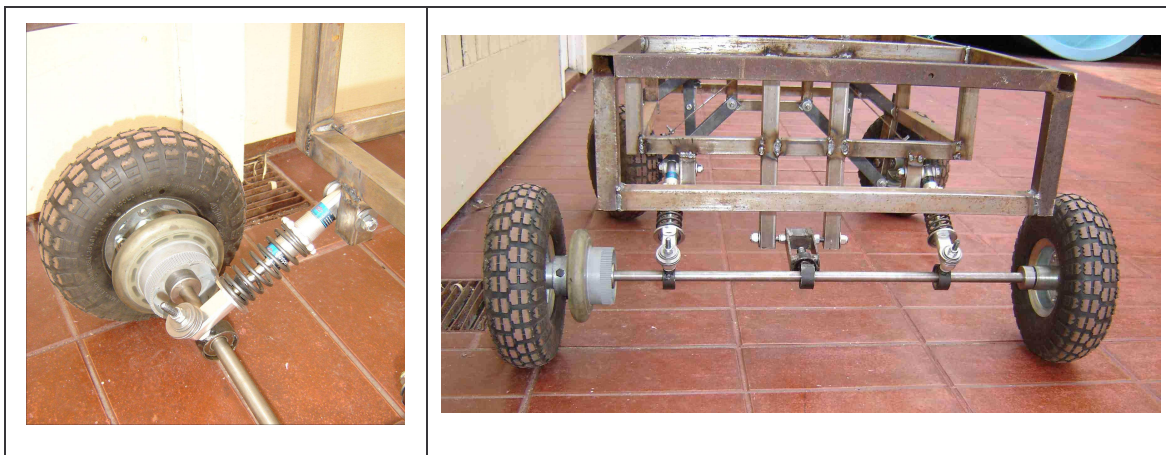


Figura 4-8. Pieza para la adaptación del engrane de tracción del motor.

CAPITULO 4: IMPLEMENTACIÓN

La dirección del vehículo fue implementada mediante un motor de 12 V obtenido de un mecanismo de limpiaparabrisas de un vehículo Renault, el cual tiene una capacidad de torque desconocida. Para suplir esta interrogante, se decidió potenciar el mecanismo de dirección a través del principio de palanca. (ver Figura 4-9)

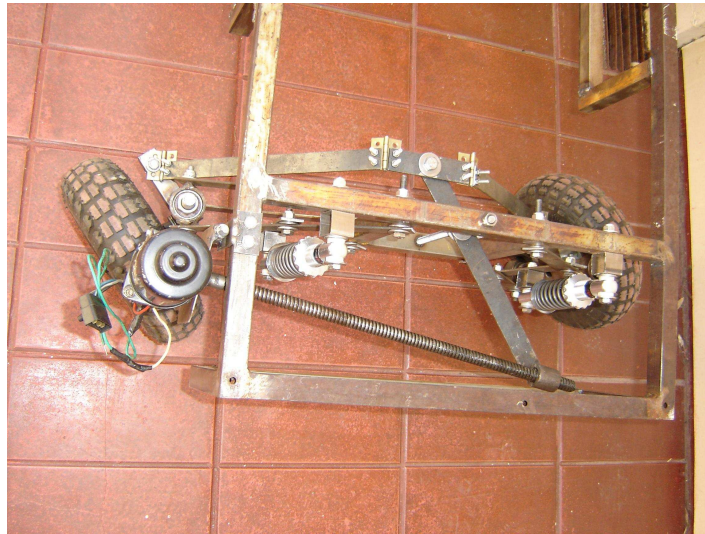


Figura 4-9. Implementación de la dirección del Agente Inteligente.

CAPITULO 5: PRUEBAS

CAPITULO 5: PRUEBAS

Las pruebas realizadas sobre el sistema de software del Autómata fueron hechas con la herramienta JUnit sobre las tres clases más críticas de éste, las cuales fueron seleccionadas sobre la base de que tienen la mayor cohesión dentro de la lógica del Autómata. Estas son :

- *device.DeviceController.*
- *logic.AssociationMemoryWithEuclideanSearching.*
- *wireless.MonitoringServer.*

Se descartó el incluir el resto de las pruebas realizadas sobre el sistema debido a lo extensa que iba a resultar su documentación y al estrecho tiempo que se disponía para esto.

Dado que las pruebas se realizaron sobre clases críticas, como se había explicado en el párrafo anterior, los métodos de éstas, en su mayoría, permiten probar la integridad sobre el sistema del Autómata dada su relación con otras clases de éste. A su vez, el resto de los métodos fuera del conjunto ya mencionado, permitieron la realización de pruebas de unidad.

Para poder analizar las pruebas anteriormente mencionadas, por favor, refiérase al Anexo A.

CAPITULO 6: CONCLUSIONES

6.1 Conclusiones Generales

Los puntos en que se divide la conclusión serán los siguientes:

- Técnica de inteligencia artificial de redes neuronales.
- Comunicaciones vía protocolo RS-232.
- La necesidad de un equipo multidisciplinario.

En relación al primer punto, buscando soluciones en las distintas herramientas que provee esta técnica (Adaline (Adaptive Linear Elemento), BPN (Back Propagation Network) y BAM (Bidirectional Association Memory)), se puede decir que en la construcción de una red de este tipo se requiere de mucho conocimiento empírico para determinar la arquitectura más idónea, esto es el número de capas de la red y neuronas por capa, el cual no se logró en el tiempo considerado para el proyecto y, por ende, la búsqueda de otra herramienta, que eliminara este factor subjetivo, continuó. Se optó por implementar una herramienta que diera solución a los requerimientos del proyecto. Dicha herramienta fue llamada MABE (Memoria de Asociación de Búsqueda Euclidiana). El problema de esta herramienta es que si se quisiera una respuesta perfecta para cada situación presentada al Agente Inteligente, se debería realizar un entrenamiento equivalente a la función de permutación del número de variables que definan el problema. Con problemas definidos con un número de variables sobre 7, el número de patrones, en un entrenamiento acabado, sería de sobre 5040.

Con respecto al segundo punto, se puede destacar la dificultad existente al momento de comunicar dos dispositivos que trabajan a una velocidad de procesamiento notablemente desigual, lo que provoca colisiones en los datos transmitidos, este error es conocido como Frame Errors, y se produce cuando el bit de parada de la trama se pierde.

Finalmente, reflexionando sobre los proyectos referentes a temas de robótica, se pudo desprender que la exigencia de conocimiento de distintas áreas de la ingeniería, y nociones sobre ciencias exactas, se hacen imprescindibles. Como epílogo se recomienda que los proyectos de esta índole sean realizados por equipos multidisciplinarios, ya que si este proyecto se hubiese hecho en un ambiente multidisciplinario, habría sido finalizado en

un tiempo mucho menor, o en su defecto, en el mismo tiempo, pero con resultados más elaborados.

6.2 Trabajos Futuros

Al producto de este proyecto se le podría dar un desarrollo más elaborado, sustituyendo la implementación actual del reconocimiento de patrones por visión artificial para recorrer espacios físicos, o darle un propósito más específico que sólo el desplazamiento desde un punto geográfico a otro. Un ejemplo de lo anterior, es la adaptación del producto para la implementación de un lazarillo artificial.

El producto de este proyecto está en condiciones de participar en la competencia de robótica agrícola realizada por la Universidad de Talca, mencionada en la sección 0, realizando algunos ajustes en sus partes mecánica y software dependiendo de las normas actuales del concurso.

Bibliografía

- DARPA. 2005. Overview. [en línea]
<<http://www.darpa.mil/grandchallenge/overview.asp>> [consulta: 27 de mayo 2006].
- Departamento de Ciencia de la Computación e Inteligencia Artificial. 2005. Conceptos. [en línea]
<http://www.dccia.ua.es/dccia/inf/asignaturas/ROB/optativos/historia_honda/Concepto.htm> [consultado: 29 de mayo 2006].
- Dimitri Loukakos. 2003. Automatic Vehicle Location. [en línea]
<http://www.calccit.org/itsdecision/serv_and_tech/Automatic_vehicle_location/automatic_vehicle_location_summary.html> [consultado: 29 de mayo 2006].
- Eclipse.org. 2006. About Us. [en línea] <<http://www.eclipse.org/org/>> [consulta: 6 febrero 2006].
- Freeman, J. Y Skapura, D. 1993. Redes neuronales. Algoritmos, aplicaciones y técnicas de programación. Edición única en español.
- Grupo Robótica. 2005. Expo Robótica. [en línea]
<http://www.face.ubiobio.cl/webubb.php?id_pagina=675> [consulta: 28 de septiembre 2005].
- History of Computing Foundation. 2005. Timeline of robotics. [en línea]
<<http://www.thocp.net/reference/robotics/robotics2.htm>> [consulta: 27 de mayo 2006].
- Honda Robots. 2006. TimeLine. [en línea] <<http://www.honda-robots.com/english/html/asimo/frameset2.html>> [consulta: 30 enero 2006].
- Instituto de Información Territorial del Estado de Jalisco. 2006. Glosario.[en línea]
<<http://iit.jalisco.gob.mx/html/glosario/>> [consulta: 14 abril 2006].
- Java. 2006. About Java Technology. [en línea] <<http://java.com/en/about/>> [consulta: 18 enero 2006].
- Larman, Craig. 1999. UML Y PATRONES: introducción al análisis y diseño orientado a objetos. México, Prentice Hall. 536p.

- Mcclelland, J. Y Rumelhart, D. 1986. Explorations in Parallel Distributed Processing. MIT Press. Cambridge, MA. Vols. I y II.
- Minsky, M. Y Papert, S. 1969. Perceptrons. MIT Press. Cambridge, MA.
- Nasa. 2002. Launch Vehicle. [en línea]
<http://marsrovers.nasa.gov/mission/launch_vehicle.html> [consulta: 31 de enero 2006].
- PostgreSQL.org. 2006. About. [en línea] <<http://www.postgresql.org/about/>> [consultado: 20 febrero 2006].
- Prieto-Moreno, Andrés. Página personal de Andrés Prieto-Moreno Torres . 2006. [en línea]
<<http://www.iearobotics.com/personal/andres/proyectos/pucho/documentacion/capitulo1.pdf>>. [consulta: 16 abril 2006].
- PUC. 2006. LARC. [en línea] <<http://www.robopuc.cl/larc/index.php?idiom=sp>> [consulta: 26 de mayo 2006].
- RibbonSoft. 2006. QCad. [en línea] <<http://www.ribbonsoft.com/qcad.html>> [consulta: 30 de mayo 2006].
- ROBOCUP. 2006. RoboCup. [en línea] <www.robocup.org> [consulta: 14 de abril 2006].
- Robótica Industrial. 1990. Por P. Groover, Mike "et al". Editorial Mc Graw Hill.
- Sony. 2006a. Press Release. [en línea]
<http://www.sony.net/SonyInfo/News/Press_Archive/200011/00-057E2/> [consulta: 30 de enero 2006].
- Sony. 2006b. Press Release. [en línea]
<http://www.sony.net/SonyInfo/News/Press_Archive/200105/01-027E/> [consulta: 30 de enero 2006].
- SRI Internacional. 2006. Shakey the Robot. [en línea]
<<http://www.sri.com/about/timeline/shakey.html>> [consulta: 27 de mayo 2006].
- Tomás Arredondo, Wolfgang Freund, Cesar Muñoz, Nicolas Navarro, Fernando Quirós. 2006. IRMA, Investigación Robótica Móvil Autónoma. [en línea]
<<http://profesores.elo.utfsm.cl/~tarredondo/irma.html>> [consulta: 17 abril 2006].
- UTFSM. 2005. Muestra ELO. [en línea] <www.elo.utfsm.cl> [consulta: 28 de septiembre 2005].

- UT. 2005. Segundo Encuentro de Robótica Agrícola. [en línea] <<http://icc.otalca.cl/robotica/>> [consulta: 28 de septiembre 2005].
- Wikipedia. 2006a. Bluetooth. [en línea] <<http://es.wikipedia.org/wiki/Bluetooth>> [consulta: 10 abril 2006].
- Wikipedia. 2006b. Enlace Infrarrojo. [en línea] <http://es.wikipedia.org/wiki/Enlaces_Infrarrojos> [consulta: 10 abril 2006].

Abreviaturas

- **ANS:** Neurona Artificial.
- **SAI:** Sistema de Agente Inteligente.
- **SiM:** Sistema de Monitoreo.

Glosario

- **Agente Inteligente:** Es cualquier cosa capaz de percibir su medio ambiente con la ayuda de sensores y actuar en ese medio utilizando actuadores, de tal forma que consiga una maximización del rendimiento en las acciones realizadas.
- **Algoritmos Evolutivos Múlti-objetivos:** Estrategia basada en un modelo artificial de la evolución para poder hacer una búsqueda global de posibles soluciones a problemas complejos como la navegación y modelamiento ambiental robótica. Los principales métodos para el desarrollo de algoritmos evolutivos conocidos actualmente incluyen los algoritmos genéticos y la programación genética.
- **Autómata:** Ente representado por la unión de elementos físicos y lógicos.
- **Datum:** Modelo matemático de la tierra.
- **DGPS:** (Differential Global Positioning System) GPS Diferencial, el cual es una variante del GPS, que considera la señal de una antena terrestre especial para realizar una corrección sobre la actual ubicación. Dicha antena es aquella que se encuentre más cercana al dispositivo de DGPS.
- **Efectores:** Servomotores y motores DC que permiten el desplazamiento físico de un Autómata.
- **Estaciones Radiales:** Lugar donde se transmiten señales mediante la modulación de ondas electromagnéticas.
- **Estocástico:** Condición incierta de alguna situación o ambiente. Azaroso.
- **Geoide:** Superficie equipotencial de la Tierra, que coincide con el nivel medio del mar (NMM) sin perturbaciones y que se extiende de manera continúa por debajo de los continentes. Es una aproximación a la forma real de la Tierra, la cual es difícil de describir matemáticamente debido a las irregularidades de las superficies locales y las variaciones en el lecho marino.

- **GPS:** Sistema de posicionamiento global, que utiliza satélites geoestacionarios especializados para determinar su ubicación geográfica sobre la superficie de la tierra.
- **GPS Geodésico:** Sistema de posicionamiento global de precisión, el cual logra hasta un 99% más de efectividad que el DGPS.
- **Robot Agrícola:** Robots que en forma autónoma realizan trabajos en ambientes naturales (no controlados), prescindiendo de un operador humano. Se utilizan en diversas tareas, las cuales van desde la poda hasta la cosecha.
- **Robots Móviles Autónomos:** Robot capaz de desplazarse sin la intervención de un humano.
- **Subsistemas Complementarios:** Subsistemas de Software que dan apoyo entregando información al SAI para que éste último pueda actuar.
- **Sonofusión:** Pulso sónico capaz de generar reacciones nucleares a partir de la implosión de diminutas burbujas.

Anexos

Anexo A: Pruebas

Especificación de Caso de Prueba Comunicación con BasicStamp 2.

Historial de Revisiones

<i>Fecha</i>	<i>Versión</i>	<i>Descripción</i>	<i>Autor</i>
20/04/06	0.0	Pruebas preliminares para comprobar la conectividad con el dispositivo BasicStamp 2.	Sebastián Alexis Moraga Garrido.

Descripción

Este artefacto cubre el conjunto de pruebas realizadas sobre la Clase *device.DeviceController*.

Las pruebas realizadas aquí se realizarán sobre los siguientes métodos:

- *echo(): boolean.*
- *getCode(short): byte.*
- *getCompassAzimut(): short.*
- *getCompassData(): short.*
- *getEnviromentUltrasonicData(): short[].*
- *getEnviromentUltrasonicMeasure(): short[].*
- *getShortRepresentation(byte,byte): short.*

El entorno del cual se partió para realizar las pruebas es el de una nueva conexión con la placa BasicStamp 2 por cada función probada.

El código fuente de éste caso de prueba es *DeviceControllerTest.java*.

En la Ilustración 1 se muestra la imagen representativa de la prueba JUnit:

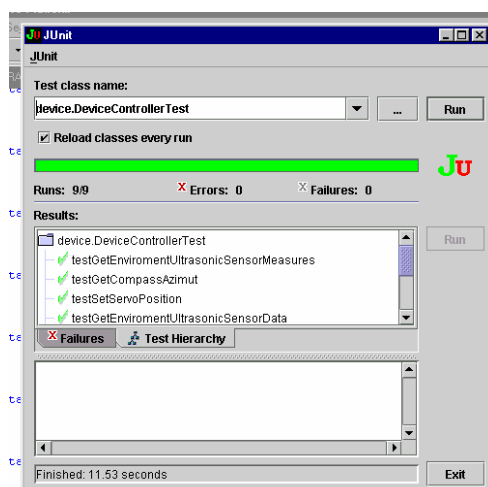


Ilustración 1. Resultados de la prueba JUnit.

echo(): boolean:

Descripción

El objetivo de esta prueba será el comprobar la conexión con la BasicStamp 2 a través de una función que enviará una señal a ella y retornará un valor que indicará que la conexión es correcta.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, la BasicStamp 2 deberá estar activada.

Entrada

Ninguna.

Resultado esperado

Si las condiciones de ejecución se cumplen, la función *echo():boolean* debería retornar verdadero.

Evaluación de la Prueba

Satisfactoria.

getCode(short): byte:

Descripción

El objetivo de esta prueba será la obtención de un byte, el cual será un código representativo de un valor (en un tipo de dato primitivo de 2 bytes) dado como parámetro.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, la BasicStamp 2 deberá estar activada.

Entrada de Clases de Caso de Prueba:

La Tabla Anexo 1 muestra las entradas para este caso de prueba.

1	→ 2.
2	→ 3.
3	→ 590.
4	→ 171.

Tabla Anexo 1. Datos de entrada.

Resultado esperado

Si las condiciones de ejecución se cumplen, la *función getCode(short):byte* debería retornar el byte representativo, mostrado en la Tabla Anexo 2, de aquel argumento.

Para cada Clase de Caso de Prueba, los resultados esperados son:

1	→ 85.
2	→ -86.
3	→ 85.
4	→ -86.

Tabla Anexo 2. Resultados esperados.

Evaluación de la Prueba

Satisfactoria.

getCompassAzimut(): short:

Descripción

El objetivo de esta prueba es el envío del comando a la BasicStamp 2 para la obtención de la medida de azimut actual desde la ella, codificada en dos bytes.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, la BasicStamp 2 deberá estar activada.

Entrada

Ninguna.

Resultado esperado

Si las condiciones de ejecución se cumplen, la función *getCompassAzimut():short* debería retornar el azimut actual representado en dos bytes, cuyo valor numérico sea mayor que -1.

Evaluación de la Prueba

Satisfactorio

getCompassData(): short:

Descripción

El objetivo de esta prueba es el envío del comando para la obtención de la medida de azimut a la BasicStamp 2 y el retorno la medida codificada en dos bytes.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, la BasicStamp 2 deberá estar activada.

Entrada

Ninguna.

Resultado esperado

Si las condiciones de ejecución se cumplen, la función *getCompassData():short* debería retornar el azimut actual representado en dos bytes, cuyo valor numérico sea mayor que -1.

Evaluación de la Prueba

Satisfactorio.

`getEnvironmentUltrasonicData(): short[]:`

Descripción

El objetivo de esta prueba es el envío del comando a la BasicStamp 2 para la obtención de las medidas ultrasónicas de distancias del ambiente.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, la BasicStamp 2 deberá estar activada al igual que el dispositivo ServoController.

Entrada

Ninguna.

Resultado esperado

Si las condiciones de ejecución se cumplen, la función `getEnvironmentUltrasonicData():short[]` debería retornar las medidas de distancias del entorno que lo rodea, las cuales deberían ser mayores a cero.

Evaluación de la Prueba

Satisfactorio.

`getEnvironmentUltrasonicMeasure(): short[]:`

Descripción

El objetivo de esta prueba es el envío del comando a la BasicStamp 2 para la obtención de las medidas ultrasónicas de distancias del ambiente.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, la BasicStamp 2 deberá estar activada al igual que el dispositivo ServoController.

Entrada

Ninguna.

Resultado esperado

Si las condiciones de ejecución se cumplen, la función `getEnvironmentUltrasonicMeasure():short[]` debería retornar las medidas de distancias del entorno que lo rodea, las cuales deberían ser mayores a cero.

Evaluación de la Prueba

Satisfactorio.

getShortRepresentation(byte,byte): short:

Descripción

El objetivo de esta prueba es la obtención de un valor único a partir de dos bytes dados como parámetro.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, la BasicStamp 2 deberá estar activada.

Entrada de Clases de Caso de Prueba:

La Tabla Anexo 3 muestra las entradas para este caso de prueba.

1	→ (057)
2	→ (0133)
3	→ (144)
4	→ (13186)

Tabla Anexo 3. Datos de entrada.

Resultado esperado

Si las condiciones de ejecución se cumplen, la función getShortRepresentation(byte,byte):short debería retornar los valores presentados en la Tabla Anexo 4, los cuales serán resultado de la integración de los dos bytes separados.

Para cada Clase de Caso de Prueba, los resultados esperados son:

1	→ 57.
2	→ 133.
3	→ 300.
4	→ 3514.

Tabla Anexo 4. Resultados esperados.

Evaluación de la Prueba

Satisfactorio.

Especificación de Caso de Prueba: Inteligencia del Autómata.

Historial de Revisiones

<i>Fecha</i>	<i>Versión</i>	<i>Descripción</i>	<i>Autor</i>
21/04/06	1.0	Pruebas preliminares para comprobar el correcto funcionamiento de la inteligencia del Autómata.	Sebastián Alexis Moraga Garrido.

Descripción

Este artefacto cubre el conjunto de pruebas realizadas sobre la Clase *logic.AssociationMemoryWithEuclideanSearching*.

Las pruebas realizadas aquí se realizarán sobre los siguientes métodos:

- *euclideanDistance(short[],short[]): double.*
- *getPositionContainer(short[]): PositionContainer[].*
- *getPatternRepresentation(PositionContainer[]): long.*
- *getDifferenceOfVectors(short[],short[]): int.*
- *filterByDifference(short[],Vector): Vector.*
- *getSolution(short[]informationPoint): Action.*

El entorno del cual se partió para realizar las pruebas es el de una nueva conexión con el servidor de Base de Conocimiento del Autómata.

El código fuente de este caso de prueba está en el archivo:

AssociationMemoryWithEuclideanSearchingTest.java.

En la Ilustración 2, se muestra una imagen representativa de la prueba JUnit aplicada:

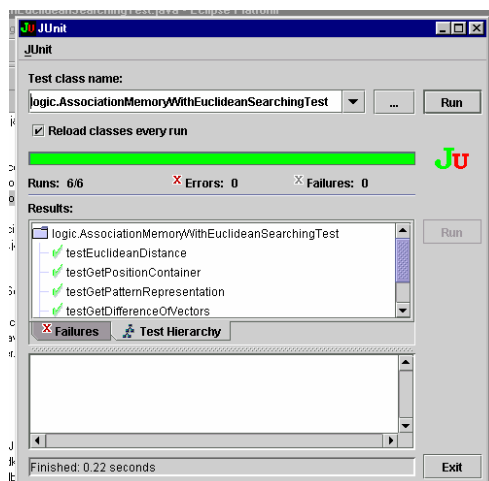


Ilustración 2. Resultados de la prueba JUnit.

euclideanDistance(short[],short[]): double:

Descripción

El objetivo de esta prueba será el comprobar el cálculo de distancia Euclidiana entre los parámetros dados a la función.

Condiciones de ejecución

Ninguna.

Entrada

En la Tabla Anexo 5 se muestran los arreglos de valores que representan puntos.

#	Vector Origen	Vector Destino
1	1,1,1	1,1,1
2	1,1,1	2,2,2
3	4,5,6,7,8,9	1,2,4,5,6
4	9,7,8	3,3,2
5	1	2,7

Tabla Anexo 5. Datos de entrada.

Resultado Esperado

La función *euclideanDistance(short[],short[]):double* debería retornar los valores de las distancias Euclideanas entre los dos vectores dados como parámetro, reflejados en la Tabla Anexo 6.

#	Resultados
1	0
2	1,732
3	-1
4	9,381
5	-1

Tabla Anexo 6. Resultados esperados.

Evaluación de la Prueba

Satisfactoria.

getPositionContainer(short[]): PositionContainer[]:

Descripción

El objetivo de esta prueba será la obtención de un arreglo de valores e índices ordenado de menor a mayor. Dichos índices serán los índices anteriores al ordenamiento de cada uno de los valores, con lo cual se podrá obtener un patrón.

Condiciones de ejecución

Ninguna.

Entrada

En la Tabla Anexo 7 muestran los arreglos de valores que representan mediciones del entorno.

#	Arreglo de Entrada
1	1,3,6
2	98,87,3
3	1,2,3,5,79,8

Tabla Anexo 7. Datos de entrada.

Resultado Esperado

La función `getPositionContainer(short[]):PositionContainer[]` debería retornar los resultados mostrados en la Tabla Anexo 8.

#	Arreglo de Entrada
1	0,1,2
2	2,1,0
3	0,1,2,3,5,4

Tabla Anexo 8. Resultados esperados.

Evaluación de la Prueba

Satisfactoria.

getPatternRepresentation(PositionContainer[]): long:

Descripción

El objetivo de esta prueba es la obtención de una representación numérica de un patrón dado un vector representativo de él.

Condiciones de ejecución

Ninguna.

Entrada

En la Tabla Anexo 9 se muestra un arreglo de valores que representan patrones de mediciones realizadas.

#	Arreglo de Entrada
1	1,3,6
2	98,87,3
3	1,2,3,5,79,8

Tabla Anexo 9. Datos de entrada.

Resultado esperado

La función *getPatternRepresentation(PositionContainer[]):long* debería retornar los valor reflejados en la Tabla Anexo 10.

#	Arreglo de Entrada
1	12
2	210
3	12354

Tabla Anexo 10. Resultados esperados.

Evaluación de la Prueba

Satisfactorio.

getDifferenceOfVectors(short[],short[]): int:

Descripción

El objetivo de esta prueba es la obtención de la diferencia existente entre dos arreglos dados como parámetro.

Condiciones de ejecución

Ninguna.

Entrada

En la Tabla Anexo 11 existen dos arreglos representativos de puntos en el espacio.

#	Vector Origen	Vector Destino
1	1,1,1	1,1,1
2	1,1,1	2,2,2
3	4,5,6,7,8,9	1,2,4,5,6
4	9,7,8	3,3,2
5	1	2,7

Tabla Anexo 11. Datos de entrada.

Resultado esperado

La función *getDifferenceOfVectors(short[],short[]):int* debería retornar los valores de la Tabla Anexo 12.

#	Resultados
1	0
2	3
3	-1
4	16
5	-1

Tabla Anexo 12. Resultados esperados.

Evaluación de la Prueba

Satisfactorio.

filterByDifference(short[],Vector): Vector:

Descripción

El objetivo de esta prueba es la obtención de un conjunto de elementos representativos de una determinada solución. Dicho conjunto es obtenido bajo el criterio de la menor diferencia existente de cada elemento dado en el segundo parámetro de la función con respecto al arreglo representativo de un punto en el espacio dado como primer parámetro en la misma.

Condiciones de ejecución

Ninguna.

Entrada

En la Tabla Anexo 13 se muestran las entradas para esta prueba, donde el primer argumento corresponde a un punto y el segundo a un conjunto de uno o más hechos.

<i>Punto de Origen</i>	<i>Conjunto de Puntos</i>	<i>Punto de Origen</i>	<i>Conjunto de Puntos</i>
50,4,34,4	2,9,3,2	1,7,6	5,6,2
	2,2,3,9		0,3,1
	1,77,9,5		1,3,9

Tabla Anexo 13. Datos de entrada.

Resultado esperado

La función *filterByDifference(short[],Vector):Vector* debería retornar los valores de las distancias de la Tabla Anexo 14.

Se destaca el o los puntos soluciones y sus respectivas distancias (tendientes a cero) del conjunto retornado de hechos.

<i>Pto. de Origen</i>	<i>Conjunto de Ptos.</i>	<i>Dist. Result.</i>	<i>Pto. de Origen</i>	<i>Conjunto de Ptos.</i>	<i>Dist. Result.</i>
50,4,34,4	2,9,3,2	57,393	1,7,6	5,6,2	5,745
	2,2,3,9	57,393		0,3,1	6,481
	1,77,9,5	91,411		1,3,9	5

Tabla Anexo 14. Resultados esperados.

Evaluación de la Prueba

Satisfactorio.

getSolution(short[]informationPoint): Action:

Descripción

El objetivo de esta prueba es la obtención de una acción dada una medición del entorno.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, la Base de Conocimiento deberá contener sólo los tres patrones reconocidos mostrados en la Tabla Anexo 15, en un espacio de 7 dimensiones:

#	Arreglo de Entrada
1	1,3,6
2	98,87,3
3	1,2,3,5,

Tabla Anexo 15. Datos iniciales.

Entrada

En la Tabla Anexo 16, se muestran arreglos representativos de mediciones del entorno.

#	Arreglo de Entrada	Patrones
1	1,3,6	12
2	98,87,3	210
3	1,2,3,5,	123

Tabla Anexo 16. Datos de entrada.

Resultado esperado

Si las condiciones de ejecución se cumplen, la función *getEnviromentUltrasonicMeasure():short[]* debería retornar las medidas de distancias del entorno que lo rodea.

Evaluación de la Prueba

Satisfactorio.

Especificación de Caso de Prueba: Servicios a Clientes de Monitoreo.

Historial de Revisiones

<i>Fecha</i>	<i>Versión</i>	<i>Descripción</i>	<i>Autor</i>
21/04/06	1.0	Pruebas preliminares para comprobar los servicios de obtención de información para realizar monitoreo al Automata en sus funciones.	Sebastián Alexis Moraga Garrido.

Descripción

Este artefacto cubre el conjunto de pruebas realizadas sobre la Clase *wireless.MonitoringServer*.

Las pruebas realizadas aquí se realizarán sobre los siguientes métodos:

- *getRobotInformationEnviroment(): RobotInformationEnviroment*.
- *verifyClientLogin(String): boolean*.
- *existPassword(String): boolean*.
- *isAdministrator(String): boolean*.
- *stopServer(String): boolean*.

El entorno del cual se partió para realizar las pruebas es el inicio del servidor de monitoreo por cada prueba.

El código fuente de este caso de prueba es *MonitoringServerTest.java*.

En la Ilustración 3, se presenta una imagen representativa de la prueba JUnit:

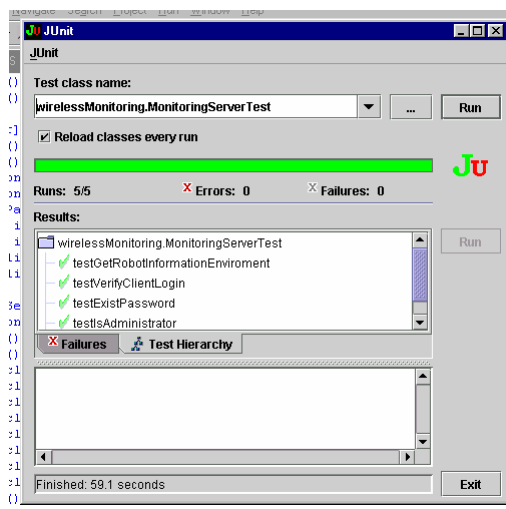


Ilustración 3. Resultados de la prueba JUnit.

getRobotInformationEnvironment():

RobotInformationEnvironment:

Descripción

El objetivo de esta prueba será la obtención de un bean representativo de la información captada por el autómata desde su entorno y las inferencias realizadas por él.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, la BasicStamp 2 y la conexión con la base de conocimiento deberán estar activadas.

Entrada

Ninguna.

Resultado esperado

Si las condiciones de ejecución se cumplen, la función *getRobotInformationEnvironment():RobotInformationEnvironment* debería retornar el bean representativo de las medidas de ambiente e inferencias realizadas por el autómata.

Evaluación de la Prueba

Satisfactoria.

verifyClientLogin(String): boolean:

Descripción

El objetivo de esta prueba será la verificación de aceptación o rechazo de un determinado cliente dado su password.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, deberá existir el registro de tres passwords en el archivo de passwords del servidor de monitoreo.

Entrada

En la Tabla Anexo 17 se muestran las passwords de cliente entrante.

Valores
7
7
1
123

Tabla Anexo 17. Datos de entrada.

Resultado esperado

Si las condiciones de ejecución se cumplen, la función *verifyClientLogin(String):boolean* debería retornar los resultados de la Tabla Anexo 18, que corresponden a la aceptación o rechazo (True o False) de un cliente entrante.

Valores	Resultados
7	False
7	False
1	True
123	False

Tabla Anexo 18. Resultados esperados.

Evaluación de la Prueba

Satisfactoria.

existPassword(String): boolean:

Descripción

El objetivo de esta prueba es la verificación de la existencia de una password en el registro de passwords del servidor de monitoreo.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, deberá existir el registro de tres passwords en el archivo de passwords del servidor de monitoreo.

Entrada

En la Tabla Anexo 19 se muestran las password a verificar.

Valores
7
1
123

Tabla Anexo 19. Datos de entrada.

Resultado esperado

Si las condiciones de ejecución se cumplen, la función *existPassword(String):boolean* debería retornar los resultados mostrados en la Tabla Anexo 20.

Valores	Resultados
7	False
1	True
123	False

Tabla Anexo 20. Resultados.

Evaluación de la Prueba

Satisfactorio.

isAdministrator(String): boolean:

Descripción

El objetivo de esta prueba es verificar si la password dada corresponde o no a la del administrador del servidor de monitoreo.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, deberá existir el registro de al menos dos passwords en el archivo de passwords del servidor de monitoreo, donde una de ellas sea la del administrador.

Entrada

En la Tabla Anexo 21 se dan a conocer las passwords a verificar.

Valores
7
1

Tabla Anexo 21. Valores de entrada.

Resultado esperado

Si las condiciones de ejecución se cumplen, la función *isAdministrator(String):boolean* debería retornar los resultados de la Tabla Anexo 22.

Valores	Resultados
7	Trae
1	False

Tabla Anexo 22. Resultados esperados.

Evaluación de la Prueba

Satisfactorio.

stopServer(String): boolean:

Descripción

El objetivo de esta prueba es verificar que la detención del servidor de monitoreo sea realizado sólo por el administrador.

Condiciones de ejecución

Antes de realizarse esta instancia del caso de prueba, deberá existir el registro de al menos dos passwords en el archivo de passwords del servidor de monitoreo, donde una de ellas sea la del administrador.

Entrada

En la Tabla Anexo 23 se muestran los passwords entrante para realizar la detención del servidor de monitoreo.

Valores
7
1

Tabla Anexo 23. Valores de entrada.

Resultado esperado

Si las condiciones de ejecución se cumplen, la función *stopServer(String):boolean* deberá retornar “True” si quien solicita la detención es administrador, o “False” en caso contrario, como se muestra en la Tabla Anexo 24.

Valores	Resultados
7	True
1	False

Tabla Anexo 24. Resultados esperados.

Evaluación de la Prueba

Satisfactorio.

Anexo B: MABE

Definición Formal de Memoria de Asociación con Búsqueda Euclidiana en Hiperespacios

Esta técnica se basa en la ubicación, en un hiperespacio, de puntos representativos de información sobre hechos o situaciones puntuales.

El procesamiento llevado a cabo por esta técnica corresponde a la medición de distancias euclidianas de un punto determinado, correspondiente a una información o situación sometida a evaluación, hacia todos los puntos del universo del hiperespacio.

El proceso de aprendizaje corresponde a la ubicación de puntos en el hiperespacio fuera del volumen que conforma el radio descrito desde un punto ya existente, y definido por un umbral predeterminado: Este umbral se llamará **Radio de Tolerancia** y el volumen descrito por él, **Volumen de Tolerancia**; los puntos en el hiperespacio se denominarán **Puntos de Información** o **Situaciones**; al conjunto de Situaciones ingresados inicialmente como conocimiento base serán el **Conjunto de Ejemplares** (Freeman y Skapura, 1993), esto es sólo un proceso opcional. Toda Situación ingresada posteriormente pertenecerá al Conjunto de Ejemplares.

Una Memoria de Asociación con Búsqueda Euclidiana en Hiperespacios se define como un universo de relaciones existentes entre tres subconjuntos que posteriormente se darán a conocer. Este universo será llamado U .

Un Conjunto de Ejemplares se define como un hiperespacio de reales. Sea E el conjunto de Ejemplares de norma L_E , tal que: $E \subset \mathbb{R}^n \wedge \|E\| = L_E$, donde n corresponde a las dimensiones del hiperespacio.

Una Situación se define como un punto perteneciente al Conjunto de Ejemplares. Sea \hat{v}_k una Situación, entonces:

$$\forall \hat{v}_i \in E; i = 1..L_E, \hat{v}_i = \{x_1, \dots, x_n\} / x_j \in \mathbb{R}; j = 1..n$$

$$\therefore (\hat{v}_k \in E \wedge E \subset \mathbb{R}^n) \Rightarrow \hat{v}_k \in \mathbb{R}^n$$

Un conjunto de Acciones se define como un hiperespacio de reales. Sea A el Conjunto de Acciones conocidas, tal que $A \in \mathbb{R}^m$, donde m corresponde a las dimensiones del hiperespacio.

Se define Acción como un punto perteneciente al conjunto de Acciones. Sea \hat{a}_k una Acción determinada del Universo de Acciones A de norma L_A , tal que: $\hat{a}_k \in A$, entonces:

$$\forall \hat{a}_i \in A; i = 1..L_A, \hat{a}_i = \{x_1, \dots, x_m\} / x_j \in \mathbb{R}; j = 1..m$$

$$\therefore (\hat{a}_k \in A \wedge A \subset \mathbb{R}^m) \Rightarrow \hat{a}_k \in \mathbb{R}^m$$

Un conjunto de Hechos se define como un conjunto de relaciones establecidas entre un punto perteneciente al Conjunto de Ejemplares y otro perteneciente al Conjunto de Acciones. Sea H el Conjunto de Hechos de norma L_H y h_k un hecho cualquiera, tal que:

$$H = \{(h_1); \dots; (h_{L_H})\} \wedge \|H\| = L_H$$

Se define un Hecho como una relación existente entre una Situación y una Acción determinada. Sea \hat{v}_k una Situación y \hat{a}_k una Acción determinada, entonces se define un Hecho h_k como:

$$h_k = \hat{v}_k \rightarrow \hat{a}_k / h_k \in H \wedge \hat{v}_k \in E \wedge \hat{a}_k \in A$$

Cabe realizar un parangón con respecto a las BAM referente a la relación que existe entre los vectores \hat{x}_k e \hat{y}_k en los Hipercubos de Hamming (Freeman y Skapura, 1993) y los puntos \hat{v}_k y \hat{a}_k en los Hiperespacios de reales, en cuanto a sus definiciones:

$$(\hat{x}_k \in H^n \wedge \hat{y}_k \in H^m) \cong (\hat{v}_k \in \mathbb{R}^n \wedge \hat{a}_k \in \mathbb{R}^m)$$

Esta relación es debido a que se pretende una generalización de la técnica de la Memoria Bidireccional Asociativa (BAM).

Es preciso ahora realizar una definición para la Memoria de Asociación con Búsqueda Euclidiana en Hiperespacios, la cual anteriormente se denominó U :

$$U = E \cup A \cup H$$

...estos son los tres subconjuntos de los que se había mencionado que estaba compuesta la memoria.

$$\therefore E \subset U \wedge A \subset U \wedge H \subset U$$

Finalmente, el Conjunto de Hechos H establece una relación entre los Conjuntos de Acciones A y de Ejemplares E .

Sea $f(\hat{v})$ una función de búsqueda definida sobre el Conjunto de Hechos H mediante una Situación \hat{v}_k . Sea h_k un determinado hecho y \hat{a}_k una determinada acción, entonces:

$$f : \hat{v}_k \in E \rightarrow h_k \in H; h_k = \hat{v}_k \rightarrow \hat{a}_k$$

$$\therefore f(\hat{v}_k) = h_k \text{ si y sólo si } h_k = \hat{v}_k \rightarrow \hat{a}_k$$

La función de búsqueda $f(\hat{v})$ define una búsqueda sobre el Conjunto de Hechos H , permitiendo obtener un subconjunto de quienes cumplan con la condición definida en la función ($h_k = \hat{v}_k \rightarrow \hat{a}_k$).

Sea \hat{v}_l una Situación determinada y \hat{v}_i una Situación cualquiera. Sea e la diferencia existente entre \hat{v}_l y \hat{v}_i . La función $f(\hat{v})$ considera el hecho de que la diferencia e siempre debe ser mínima sobre el Conjunto de Ejemplares, esto es:

$$f(\hat{v}_l) = \{h_1, \dots, h_o\} / h_i = \hat{v}_i \rightarrow \hat{a}_i \wedge (\hat{v}_l \Delta \hat{v}_i) = e; e \rightarrow 0 \wedge i = 0..L_H$$

...donde el subíndice o corresponde a la norma del conjunto solución de Hechos ($\{h_1, \dots, h_o\}$).

La diferencia e puede ser calculada mediante la diferencia entre distancias Euclidianas desde el origen hacia los puntos que representan las situaciones. A la función de distancia Euclidianas desde el origen hacia una Situación \hat{v} determinada, se llamará $\varepsilon(\hat{v})$, de tal forma que:

$$\varepsilon(\hat{v}) = \sqrt{x_1^2 + \dots + x_n^2} / x_k \in \hat{v} \wedge x_k \in \mathbb{R} \wedge \hat{v} \in E$$

Ahora, la función $f(\hat{v}_i)$ considerará las soluciones a aquellos Hechos h_i cuya Situaciones \hat{v}_i posean una diferencia mínima con respecto a \hat{v}_i . Tal diferencia mínima estará expresada por: $(\hat{v}_i \Delta \hat{v}_i) = e; e \rightarrow 0$, como ya se había mencionado. Por lo tanto:

$$(\hat{v}_i \Delta \hat{v}_i) = |\varepsilon(\hat{v}_i) - \varepsilon(\hat{v}_i)| = e; i = 0..L_H$$

$$\therefore |\varepsilon(\hat{v}_i) - \varepsilon(\hat{v}_i)| = e \wedge e \rightarrow 0; i = 0..L_H$$

Finalmente, la definición de $f(\hat{v})$ corresponde a:

$$f(\hat{v}_i) = \{h_1, \dots, h_o\} / h_i = \hat{v}_i \rightarrow \hat{a}_i \wedge |\varepsilon(\hat{v}_i) - \varepsilon(\hat{v}_i)| \rightarrow 0; i = 0..L_H$$

Por lo tanto, los resultados de la función $f(\hat{v})$ podrán arrojar más de una Acción a considerar.

Finalmente, es necesario dejar en claro que estas definiciones son sólo de propósito general. Para la resolución de problemas puntuales que requieran de la adaptación sobre las definiciones expresadas anteriormente, deberán ser realizadas con el fin de conseguir la o las respuestas más adecuadas a las necesidades del problema.