



Universidad  
Carlos III de Madrid

**ESCUELA POLITÉCNICA SUPERIOR**

**INGENIERÍA EN INFORMÁTICA**

**Área de Ciencia de la Computación e Inteligencia Artificial**

**Proyecto Fin de Carrera**

**Implementación de Arquitectura de Control 3T en la  
RoboCup Small Size League: Formación de Defensa**

**AUTOR: Alejandro Caparrós Hernando**

**TUTOR: Agapito Ledezma Espino**

**DIRECTOR: José Antonio Iglesias Martínez**

**Leganés, Septiembre 2012**





## Agradecimientos

*Este proyecto no habría sido posible sin la ayuda y apoyo que he recibido por parte de toda la gente que me ha estado a mi lado a lo largo de todo el desarrollo del mismo.*

*En primer lugar, quiero agradecer todo el apoyo que me ha dado mi familia, en especial a mis padres, a mi hermano Rubén y a su pareja, Carla, los cuales me han animado en todo momento y nunca han dudado en ayudarme cuando lo he necesitado.*

*A mis amigos, por animarme en los momentos de agobio, permitiéndome ver las cosas desde un punto de vista más optimista.*

*A mi tutores de proyecto, José Antonio y Agapito, por darme la oportunidad de participar en este proyecto y guiarme en el desarrollo del mismo.*

*A mis compañeros de Universidad y ya mis amigos, Fran, Javi, Luis, Rober y Samu, porque nunca dudaron en ayudarme y hacer que los momentos más difíciles pudiera superarlos con sentido del humor y una sonrisa.*

*Al grupo Parsian Robotic, en especial a Ali y Mani, por permitirme colaborar en su grupo de desarrollo y así enriquecer el proyecto.*

*A Isaac, por colaborar en el desarrollo de pruebas con el robot físico.*

*A Noemí, Hilario, Alicia, Julián y al pequeño Izan por ayudarme y hacerme sentir uno más de la familia.*

*Por último, a Vero, la persona más importante de mi vida y la que me ha hecho tan feliz. Sin tu ayuda y tu apoyo no habría podido llegar a finalizar este proyecto. Gracias por estar siempre a mi lado, animarme y ayudarme siempre que lo que he necesitado.*



## Resumen

La RoboCup [4] es un proyecto internacional que aparece en 1997 para promover, a través de competencias integradas por robots autónomos, la investigación y educación sobre inteligencia artificial. [2] Esta se compone de distintos tipos de competiciones que tratan de conseguir una investigación sobre el comportamiento, coordinación y trabajo en equipo de los robots.

El objetivo principal de este proyecto es la implementación de una arquitectura 3T [36] para los robots de la RoboCup SSL [5] y desarrollar algunas habilidades básicas para el control de los robots y de esta forma asentar una base sobre la que trabajar en el futuro.

Además, se realizan pruebas del trabajo realizado tanto en un simulador como en un robot real para comprobar el correcto funcionamiento de la arquitectura y comparar la ejecución en un entorno simulado y en un entorno real.



## Abstract

The RoboCup [4] is an international Project that appears in 1997 to promote, through integrated skills for autonomous robots, research and education in artificial intelligence. [2] This is composed of different types of competitions that seek to achieve behavioral research, coordination and teamwork of the robots.

The main objective of this project is the implementation of a 3T architecture [36] for the robots of the RoboCup SSL [5] and develop some basic skills to control robots and thus laid a foundation on which to work in the future.

In addition, tests are done the work done both in a simulator and in a real robot to verify the proper functioning of the architecture and compare the performance in a simulated environment and real environment.



## Índice de Contenido

<b>Agradecimientos</b> .....	<b>3</b>
<b>Resumen</b> .....	<b>4</b>
<b>Abstract</b> .....	<b>5</b>
<b>Índice de Contenido</b> .....	<b>6</b>
<b>Índice de Ilustraciones</b> .....	<b>9</b>
<b>Índice de Tablas</b> .....	<b>12</b>
<b>1. Introducción</b> .....	<b>16</b>
<b>1.1. Objetivos</b> .....	<b>16</b>
<b>1.2. Fases del PFC</b> .....	<b>18</b>
<b>1.3. Medios Utilizados</b> .....	<b>21</b>
1.3.1. Software .....	21
1.3.2. Hardware .....	25
<b>1.4. Estructura del documento</b> .....	<b>27</b>
<b>2. Estado del Arte</b> .....	<b>30</b>
<b>2.1. Robots y Robótica en la Actualidad</b> .....	<b>30</b>
<b>2.2. Paradigmas de la Robótica</b> .....	<b>30</b>
2.2.1. Paradigma Reactivo .....	30
2.2.2. Paradigma Jerárquico .....	31
2.2.3. Paradigma Híbrido .....	32
<b>2.3. Arquitectura 3T</b> .....	<b>33</b>
<b>2.4. RoboCup</b> .....	<b>34</b>
2.4.1. Competiciones RoboCup .....	35
2.4.2. Small Size League.....	44
2.4.3. Reglas para la SSL .....	50
2.4.4. Simuladores de la SSL .....	51
<b>3. Trabajo realizado</b> .....	<b>54</b>
<b>3.1. Requisitos de Usuario</b> .....	<b>54</b>
<b>3.2. Simulador grSim</b> .....	<b>55</b>
3.2.1. Funcionamiento del Simulador .....	55
3.2.2. Interfaz del Simulador .....	57
3.2.3. Representación Gráfica .....	62
3.2.4. Adaptación de los Robots a Tres Ruedas.....	63
3.2.5. Sistema de Visión en grSim .....	65
<b>3.3. Clientes Java de Envío y Recepción de Paquetes del Simulador grSim</b> .....	<b>66</b>
3.3.1. Cliente Java de Envío de Comandos .....	67
3.3.2. Cliente Java de Recepción de Paquetes del Simulador .....	68
<b>3.4. Adaptación de Arquitectura 3T para Desarrollar las Habilidades</b> .....	<b>69</b>
3.4.1. Diseño Arquitectónico .....	69
3.4.2. Diseño Detallado del Sistema .....	73



3.4.3. Diagrama de Ejecución .....	78
<b>3.5. Habilidades .....</b>	<b>79</b>
3.5.1. Habilidad: Giro Óptimo.....	82
3.5.2. Habilidad: Defender Portería (Portero).....	84
3.5.3. Habilidad: Defender Portería (Jugador) .....	85
3.5.4. Habilidad: Pase .....	86
3.5.5. Habilidad: Esquivar .....	88
<b>3.6. Adaptación a Robots Físicos .....</b>	<b>89</b>
3.6.1. Características Técnicas.....	89
3.6.2. Conexiones .....	94
3.6.3. Sistema de Visión.....	94
<b>4. Experimentos .....</b>	<b>99</b>
<b>4.1. Experimentos en Simulador grSim .....</b>	<b>99</b>
4.1.1. Experimento con Cliente de Recepción Java .....	99
4.1.2. Experimento con Cliente de Envío Java.....	101
4.1.3. Experimento con la Habilidad Giro Óptimo I.....	103
4.1.4. Experimento con la Habilidad Giro Óptimo II.....	104
4.1.5. Experimento con la Habilidad Defender Portería en Portero .....	105
4.1.6. Experimento con la Habilidad Defender Portería en Jugador .....	105
4.1.7. Experimento con la Habilidad Pase .....	106
4.1.8. Experimento con la Habilidad Esquivar .....	107
<b>4.2. Experimentos en Robot Físico.....</b>	<b>108</b>
4.2.1. Experimento Buscar Pelota .....	109
4.2.2. Problemas Surgidos con el Robot Físico .....	110
4.3. Resultados Finales .....	110
<b>5. Presupuesto .....</b>	<b>113</b>
<b>5.1. Descripción del Proyecto .....</b>	<b>113</b>
<b>5.2. Desglose del Presupuesto.....</b>	<b>113</b>
5.2.1. Personal.....	113
5.2.2. Material .....	114
<b>5.3. Resumen del Proyecto.....</b>	<b>116</b>
<b>6. Conclusiones .....</b>	<b>118</b>
<b>7. Futuras líneas de trabajo .....</b>	<b>121</b>
<b>Bibliografía.....</b>	<b>125</b>
<b>ANEXO I: Definiciones y acrónimos .....</b>	<b>131</b>
Definiciones .....	131
Acrónimos.....	133
<b>ANEXO II: Requisitos de Usuario .....</b>	<b>135</b>
<b>1 Requisitos de simulador grSim .....</b>	<b>136</b>
<b>2 Requisitos de cliente Java de envío .....</b>	<b>137</b>
<b>3 Requisitos de cliente Java de recepción .....</b>	<b>139</b>
<b>4 Requisitos de arquitectura 3T .....</b>	<b>140</b>
<b>5 Requisitos de habilidades de SSL.....</b>	<b>142</b>



<b>ANEXO III: Manual de Usuario .....</b>	<b>146</b>
<b>Simulador grSim .....</b>	<b>146</b>
Requisitos Mínicos del Simulador grSim .....	146
Instalación del Simulador grSim .....	146
Uso del Simulador grSim .....	147
<b>Arquitectura 3T .....</b>	<b>147</b>
Requisitos mínimos de la arquitectura 3T .....	147
Ejecución de la arquitectura 3T .....	148
<b>ANEXO IV: Manual de Referencia.....</b>	<b>151</b>
<b>Características del sistema .....</b>	<b>151</b>
<b>Usuarios del sistema .....</b>	<b>152</b>
<b>Funcionalidad del sistema.....</b>	<b>152</b>
<b>ANEXO V: Gestión de Proyecto .....</b>	<b>155</b>
<b>Fases .....</b>	<b>155</b>
<b>Ciclo de Vida.....</b>	<b>157</b>
<b>Diagrama de Gantt .....</b>	<b>158</b>
<b>ANEXO VI: Reglamento SSL de 2011.....</b>	<b>162</b>
<b>ANEXO VII: Reglamento 2012 .....</b>	<b>200</b>



## Índice de Ilustraciones

Ilustración 1: esquema de fases del PFC .....	19
Ilustración 2: Parsian Robotic.....	21
Ilustración 3: Java [16].....	21
Ilustración 4: Microsoft Office 2010 [22] .....	21
Ilustración 5: Microsoft Word [22].....	22
Ilustración 6: Microsoft Excel [22].....	22
Ilustración 7: Microsoft PowerPoint [22] .....	22
Ilustración 8: ssl-vision [12] .....	23
Ilustración 9: Netbeans [23] .....	23
Ilustración 10: Gimp [24] .....	23
Ilustración 11: Mac OS X Lion [25].....	24
Ilustración 12: Ubuntu [26] .....	24
Ilustración 13: Microsoft Windows [27].....	25
Ilustración 14: logotipo Wireshark [28].....	25
Ilustración 15: MacBook Pro [29] .....	26
Ilustración 16: robot físico [30] [31] [32].....	26
Ilustración 17: AVT Stingray [33] .....	27
Ilustración 18: paradigma reactivo.....	31
Ilustración 19: paradigma jerárquico .....	32
Ilustración 20: paradigma híbrido .....	33
Ilustración 21: arquitectura 3T [36].....	34
Ilustración 22: logotipo RoboCup [4].....	34
Ilustración 23: logotipo de la RoboCup Rescue [4] .....	35
Ilustración 24: liga de simulación de RoboCup Rescue [4].....	36
Ilustración 25: liga de robots reales de RoboCup Rescue [4].....	36
Ilustración 26: logotipo RoboCup Junior [4].....	37
Ilustración 27: liga soccer de RoboCup Junior [4] .....	37
Ilustración 28: liga rescue de RoboCup Junior [4] .....	37
Ilustración 29: liga dance de RoboCup Junior [4] .....	38
Ilustración 30: logotipo RoboCup @Home [4] .....	38
Ilustración 31: RoboCup @Home [4].....	39
Ilustración 32: liga simulación en 2D de RoboCup Soccer [4] .....	40
Ilustración 33: liga simulación en 3D de RoboCup Soccer [4] .....	40
Ilustración 34: AIBO de Sony usados en la liga de plataforma estándar [4] [35].....	41
Ilustración 35: liga de plataforma estándar de RoboCup Soccer [4] [34] .....	42
Ilustración 36: liga de robots humanoides entre 30 y 60 cm de RoboCup Soccer [4] ..	42
Ilustración 37: liga de robots humanoides entre 100 y 120 cm de RoboCup Soccer [4] .....	43
Ilustración 38: liga de robots humanoides más de 130 cm de RoboCup Soccer [4] .....	43
Ilustración 39: liga de robots medianos de RoboCup Soccer [4].....	44



Ilustración 40: liga de robots pequeños de RoboCup Soccer [4] .....	45
Ilustración 41: ejemplo de liga de robots pequeños de RoboCup Soccer [4] .....	45
Ilustración 42: relación entre equipos en activo y formados [5] .....	47
Ilustración 43: participación por países en la SSL [5] .....	48
Ilustración 44: logotipo de Twenta [6] .....	51
Ilustración 45: logotipo grSim [7] [15].....	52
Ilustración 46: logotipo del equipo CMDragons [8] .....	52
Ilustración 47: funcionamiento grSim [7] [15] .....	56
Ilustración 48: interfaz gráfica del simulador grSim [7] [15].....	59
Ilustración 49: modificación robot .....	60
Ilustración 50: cambio de formación de los robots.....	61
Ilustración 51: cambio de opciones.....	61
Ilustración 52: cambio de posición y orientación.....	62
Ilustración 53: signo de las posiciones en el terreno de juego .....	62
Ilustración 54: orientación en el terreno de juego.....	63
Ilustración 55: robots en el simulador grSim [7] [15] con 4 ruedas .....	64
Ilustración 56: robots en el simulador grSim [7] [15] con 3 ruedas .....	64
Ilustración 57: modificaciones en el simulador grSim [7] [15].....	65
Ilustración 58: distribución de las ruedas.....	65
Ilustración 59: Google Protocol Buffers [12] .....	66
Ilustración 60: interfaz cliente de envío Java [16].....	67
Ilustración 61: arquitectura 3T [36].....	70
Ilustración 62: controlador .....	70
Ilustración 63: secuenciador.....	72
Ilustración 64: planificador .....	73
Ilustración 65: diagrama de clases de Infraestructura .....	74
Ilustración 66: diagrama de clases de Controlador .....	75
Ilustración 67: diagrama de clases de Secuenciador.....	76
Ilustración 68: diagrama de clases de Planificador .....	77
Ilustración 69: diagrama de ejecución .....	78
Ilustración 70: gráfica de habilidades elaboradas para la liga SSL [5].....	81
Ilustración 71: cálculo dirección entre dos puntos .....	82
Ilustración 72: giro óptimo del robot sobre sí mismo .....	83
Ilustración 73: ejemplo de habilidad giro óptimo .....	84
Ilustración 74: habilidad defender portería con el portero .....	85
Ilustración 75: habilidad defender portería con jugador .....	86
Ilustración 76: habilidad pase.....	87
Ilustración 77: habilidad esquivar .....	89
Ilustración 78: sistema de procesamiento [31] [32].....	90
Ilustración 79: controlador Wi-Fi RCM4400-Rabbit [31] [32] .....	90
Ilustración 80: situación de la batería en el robot [31] [32].....	91
Ilustración 81: rueda del robot [31] [32].....	92
Ilustración 82: sistema de <i>dribbler</i> [31] [32] .....	92
Ilustración 83: sistema dribbler en el robot [31] [32] .....	93
Ilustración 84: robot físico [31] [32].....	93



Ilustración 85: interfaz SSL-Vision [12] .....	95
Ilustración 86: vista de la cámara con SSL-Vision [13] .....	96
Ilustración 87: representación del campo en SSL-Vision [13] .....	97
Ilustración 88: experimento con cliente de recepción Java .....	101
Ilustración 89: cliente de envío Java.....	102
Ilustración 90: entorno de realización de las pruebas con el robot físico .....	108
Ilustración 91: robot físico .....	109
Ilustración 92: ciclo de vida cascada .....	157
Ilustración 93: diagrama de Gantt.....	159
Ilustración 94: partido perteneciente a la SSL.....	162
Ilustración 95: sistema de visión .....	163
Ilustración 96: dimensiones de la superficie de juego .....	164
Ilustración 97: dimensiones de las portería .....	165
Ilustración 98: sistema de visión .....	166
Ilustración 99: cámara Stingray .....	167
Ilustración 100: identificación de los robots .....	169
Ilustración 101: dimensiones de los robots.....	170
Ilustración 102: sistema dribbler .....	171
Ilustración 103: posición de la pelota respecto al robot.....	187



## Índice de Tablas

Tabla 1: ranking de victorias en SSL [5] .....	49
Tabla 2: paquete recibido del simulador grSim [7] [15] .....	57
Tabla 3: paquete enviado al simulador grSim [7] [15] .....	57
Tabla 4: habilidades elaboradas para la liga SSL [5] .....	80
Tabla 5: descripción del proyecto .....	113
Tabla 6: coste de personal .....	114
Tabla 7: coste de material .....	115
Tabla 8: resumen del proyecto .....	116
Tabla 9: acrónimos .....	133
Tabla 10: requisito .....	135
Tabla 11: REQ-SIM-01 .....	136
Tabla 12: REQ-SIM-02 .....	137
Tabla 13: REQ-SIM-03 .....	137
Tabla 14: REQ-ENV-01 .....	137
Tabla 15: REQ-ENV-02 .....	137
Tabla 16: REQ-ENV-03 .....	138
Tabla 17: REQ-ENV-04 .....	138
Tabla 18: REQ-ENV-05 .....	138
Tabla 19: REQ-ENV-06 .....	139
Tabla 20: REQ-REC-01 .....	139
Tabla 21: REQ-REC-02 .....	139
Tabla 22: REQ-ENV-03 .....	139
Tabla 23: REQ-ARQ-01 .....	140
Tabla 24: REQ-ARQ-02 .....	140
Tabla 25: REQ-ARQ-03 .....	140
Tabla 26: REQ-ARQ-04 .....	141
Tabla 27: REQ-ARQ-05 .....	141
Tabla 28: REQ-ARQ-06 .....	141
Tabla 29: REQ-ARQ-07 .....	141
Tabla 30: REQ-ARQ-08 .....	142
Tabla 31: REQ-ARQ-09 .....	142
Tabla 32: REQ-HAB-02 .....	142
Tabla 33: REQ-HAB-02 .....	142
Tabla 34: REQ-HAB-03 .....	143
Tabla 35: REQ-HAB-04 .....	143
Tabla 36: REQ-HAB-05 .....	143
Tabla 37: REQ-HAB-06 .....	143
Tabla 38: REQ-HAB-07 .....	144
Tabla 39: REQ-HAB-08 .....	144
Tabla 40: REQ-HAB-09 .....	144



Tabla 41: identificación de equipos.....	149
Tabla 42: cámaras del SSL-Vision [12] .....	167
Tabla 43: modificaciones reglas RoboCup [17] 2012 respecto a las reglas RoboCup [17] 2011 .....	201





# CAPÍTULO 1: INTRODUCCIÓN

---

## 1. Introducción

La robótica es una ciencia que combina distintas disciplinas como son la inteligencia artificial, la mecánica, la electrónica, la informática y la ingeniería de control. [2]

El término robot dio a conocer gracias al éxito de la obra "*Robots Universales Rossum*" escrita por Karel Capek en el año 1920. En la traducción al inglés de dicha obra, la palabra checa *robota*, que significa *trabajos forzados*, fue traducida al inglés como *robot*. [2]

Este proyecto trata sobre el desarrollo de una arquitectura híbrida 3T [36] para robots de la RoboCup [4] Small Size League (desde ahora se denominara SSL como es comúnmente conocida) [5]. La finalidad de esta competición es desarrollar equipos de robots que sean capaces de jugar un partido de fútbol, lo que lleva a realizar una gran labor de investigación y desarrollo en el mundo de la robótica. A pesar de esto, el objetivo oficial de la RoboCup [4] es conseguir que a mediados de siglo XXI un equipo de robots sea capaz de vencer al equipo ganador de la Copa Mundial de la FIFA siguiendo sus propias reglas.

A continuación, se detallarán los objetivos que se buscan con el desarrollo de este proyecto basado en la competición SSL [5], los medios utilizados en el desarrollo del mismo y la estructura que seguirá el documento.

### 1.1. Objetivos

El objetivo principal de este proyecto es la creación de una arquitectura 3T [36] que controle robots pertenecientes a la liga SSL [5] de la RoboCup [4]. En esta arquitectura se implementarán algunas habilidades básicas para el control de estos robots.

La decisión de crear esta arquitectura para la 3T [36] sirve, principalmente, para establecer las bases para poder definir un equipo para la liga SSL [5] de la competición RoboCup. Este proyecto, por lo tanto, tiene como objetivo principal definir esta



arquitectura para dejar establecido la futura continuación y ampliación para la creación de un equipo para la SSL [5].

Esta arquitectura dará la posibilidad de ofrecer movimientos básicos relacionados con el fútbol, que engloban, a su vez, las habilidades definidas. Esto permite dotar a los robots, tanto físicos como simulados, de comportamiento *inteligente* que va estrechamente relacionado con los movimientos que surgen en el fútbol.

Para poder establecer esta arquitectura y, debido a las características de la SSL [5], es necesario utilizar previamente un simulador. Como parte de este objetivo, se incluye la investigación realizada para obtener este simulador, así como el aprendizaje del mismo y las modificaciones necesarias para adaptarlo a los robots físicos utilizados en este proyecto (que posee la Universidad Carlos III de Madrid). Para ello, se ha utilizado el simulador grSim [7] [15], perteneciente a un equipo en activo en la liga SSL: Parsian Robotic [7] [15].

Además, es necesario realizar todas las comunicaciones entre todos los módulos, para lograr una plena compenetración que permita que todo el sistema en conjunto logre funcionar correctamente. Por este motivo, se han desempeñado funciones relacionadas con la comunicación y el entendimiento de los módulos, por ejemplo, la comunicación entre la arquitectura y el simulador grSim [7] [15].

Finalmente, como objetivo se verificará el pleno funcionamiento de todo el sistema creado, desde las habilidades a las pruebas con robots físicos, incluyendo las pruebas con el cliente Java [16], etc.

Dentro de este objetivo principal que engloba todo este trabajo, existen subobjetivos que definen y concretan toda la actividad realizada. Esos subobjetivos son detallados a continuación:

- El objetivo principal es **definir una arquitectura de control 3T [36] para la SSL** [5] que defina habilidades básicas orientadas a la formación de defensa. Para

lograr este objetivo general es necesario realizar los siguientes subobjetivos que se enumeran a continuación:

- Se adaptará la arquitectura de control 3T a partir de una implementación propia de un PFC de la Universidad Carlos III de Madrid [20].
- Se buscará, analizará y utilizará un simulador existente, perteneciente a un equipo en activo. En este caso se considera el simulador grSim [7] [15], que pertenece al equipo Parsian Robotic de Irán.
- Se cambiará y adaptará el simulador grSim [7] [15] utilizado, para poder acoplarlo al robot físico que pertenece a la Universidad Carlos III de Madrid [30] [31] [32].
- Se implementarán habilidades básicas orientadas a la formación de defensa dentro de la liga SSL [18].
- Se realizarán pruebas de todo el sistema, de forma independiente en cada módulo, así como todo el sistema en conjunto.

## 1.2. Fases del PFC

En este punto se detallarán las fases que engloban este PFC y que se han seguido a lo largo del desarrollo del mismo. En el siguiente esquema queda representadas las fases de forma esquemática y justo a continuación se explicarán en detalle.

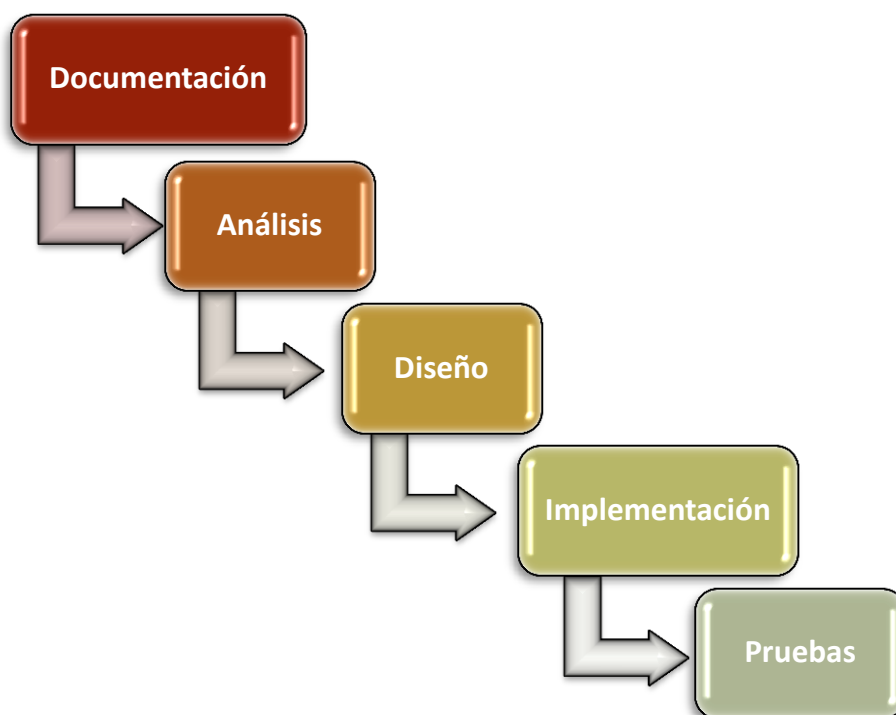


Ilustración 1: esquema de fases del PFC

La primera fase corresponde a la **documentación e investigación** sobre el tema a tratar. Este bloque ha consistido en el asentamiento de las bases del conocimiento para, posteriormente, poder realizar las siguientes fases. La documentación se ha basado en adquirir información global sobre la competición RoboCup [4], así como ahondar en la liga en la que se enmarca este PFC: SSL [5]. Por otro lado, se han buscado los posibles simuladores para realizar la simulación del comportamiento de los robots, que plasmará exactamente las exigencias de la SSL [5]. Además, se ha indagado sobre las distintas arquitecturas y paradigmas de la robótica para conocer las posibles alternativas que existen.

La segunda fase corresponde a **la fase de análisis**, en donde se ha realizado un estudio del trabajo a realizar, definiendo qué había que hacer. Debido a que este PFC abarca muchas materias y van apareciendo nuevas tareas a realizar una vez que se está haciendo, es necesario analizar con minuciosidad todo el proceso.

La tercera fase equivale al **diseño de la arquitectura**, detallando el nivel arquitectónico y el detallado. Este trabajo tiene distintas vertientes que se unifican posteriormente,

por lo que el diseño se ha hecho de todas ellas. Esto se debe a que, por un lado, existe el trabajo desarrollado de cara a adaptar la arquitectura 3T [36] ya existente y, por otro lado, todo el trabajo de adaptación del simulador grSim [7] [15] a los robots físicos y a la arquitectura. La unificación de estas dos tareas da como resultado las bases para poder elaborar las habilidades básicas propias de un robot de la SSL [5].

La cuarta fase equivale a la **implementación del sistema**. Como se ha descrito en los párrafos anteriores, tras realizar un análisis y diseño del trabajo a realizar, se procedió a llevarlo a cabo. Esta implementación lleva asociadas dos áreas diferenciadas, por un lado la integración y modificación de la arquitectura 3T [36] para la adaptación a los robots de la SSL [5] y, por otro lado, la adaptación del simulador grSim [7] [15]. Una vez que estas dos áreas se unifican, siendo la arquitectura capaz de comunicarse con el simulador grSim [7] [15], se procede a realizar las habilidades básicas del robot.

La quinta fase corresponde a la **fase de pruebas**, en donde se ha corroborado que los robots realizan las habilidades programadas. Para ello, se ha probado con el simulador grSim [7] [15] y robots simulados. Además, y a modo de ampliación de este PFC, se ha probado con robots físicos.

Por último, a pesar de que se ha ido realizando en cada una de las fases de forma incremental, la sexta fase consiste en la **creación de la presente memoria**, que incluye la recopilación de toda la documentación obtenida.

### 1.3. Medios Utilizados

En este apartado se van a describir todos los medios software y hardware empleados en el desarrollo del proyecto.

#### 1.3.1. Software

El software utilizado para el desarrollo del proyecto es el siguiente:

- **Simulador grSim [7] [15]:** simulador de la RoboCup [4] SSL [5] desarrollado en el lenguaje de programación C++ [17] por el equipo Parsian Robotic:



Ilustración 2: Parsian Robotic

- **Java Virtual Machine (JVM):** máquina virtual de Java [16] que permite ejecutar todo el código compilado de Java (bytecode).



Ilustración 3: Java [16]

- **Microsoft Office 2010:** suite ofimática que abarca e interrelaciona aplicaciones de escritorio, servidores y servicios para sistemas operativos Windows y Mac OS X. [22]



Ilustración 4: Microsoft Office 2010 [22]

Las aplicaciones utilizadas de Office son:

- **Microsoft Word:** software utilizado para el procesamiento de textos. Utilizado principalmente para realizar la documentación del proyecto. [22]



Ilustración 5: Microsoft Word [22]

- **Microsoft Excel:** aplicación para manejar hojas de cálculo. Ha sido utilizado para la realización de gráficas contenidas en esta memoria. [22]



Ilustración 6: Microsoft Excel [22]

- **Microsoft Powerpoint:** programa para realizar presentaciones. Es utilizado para la creación de la presentación del proyecto. [22]



Ilustración 7: Microsoft PowerPoint [22]

- **SSL-Vision:** sistema de visión utilizado para la SSL [5]. Recopila la información de los robots y la pelota y la envía a los ordenadores de los participantes. [12]



Ilustración 8: ssl-vision [12]

- **Netbeans:** plataforma de desarrollo de aplicaciones, es utilizado principalmente para el desarrollo de aplicaciones Java [16]. Además existe una gran cantidad de módulos para ampliarlo. La decisión del uso de este entorno de desarrollo ha sido porque es gratuito, de software libre y está hecho para la programación en lenguaje de programación Java [16]. [23]



Ilustración 9: Netbeans [23]

- **Gimp:** programa de edición de imágenes digitales. Se ha usado este programa de edición de imágenes porque es gratuito y se puede utilizar en distintos sistemas operativos, como Windows y Ubuntu. [24]

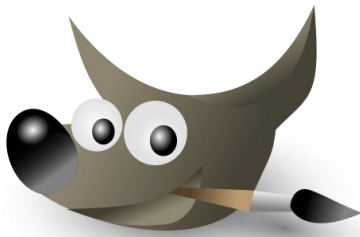


Ilustración 10: Gimp [24]

- **Mac OS X Lion:** sistema operativo de la empresa Apple con el cual se ha desarrollado todo el trabajo de documentación. [25]



Ilustración 11: Mac OS X Lion [25]

- **Ubuntu:** sistema operativo libre basado en Debian, cuyo núcleo es Linux. Ha sido utilizado para el desarrollo del código y la ejecución del simulador grSim [7] [15]. El simulador grSim [7] [15] requiere una distribución Linux, se ha utilizado Ubuntu por la recomendación directa de los creadores de este simulador, así como su simplicidad y facilidad de uso. Se podría haber usado cualquier otra distribución de Linux, ya que grSim [7] [15] es compatible.[26]



Ilustración 12: Ubuntu [26]

- **Windows 7:** sistema operativo desarrollado por Microsoft. La utilización de este sistema operativo para desarrollar la documentación es debido a que el software de ofimática que ofrece es completo e incluye herramientas útiles que facilitan la elaboración de esta. Por ejemplo, la creación de diagramas de Gantt o las herramientas de diseño. [27]





Ilustración 13: Microsoft Windows [27]

- **Wireshark:** analizador de protocolos utilizado para captar los paquetes enviados al simulador grSim [7] [15] y recibidos por este. Se ha utilizado este software por ser un programa gratuito y fácil de utilizar. [28]



Ilustración 14: logotipo Wireshark [28]

### 1.3.2. Hardware

En este apartado se van a describir los medios Hardware utilizados para el desarrollo del proyecto:

- **Ordenador portátil:** para el desarrollo del proyecto se ha hecho uso de un ordenador portátil. En concreto un Apple MacBook Pro de 13,3 pulgadas. Este equipo ha sido utilizado para todo el desarrollo y la implementación de la arquitectura y para la ejecución del simulador grSim [7] [15].



Ilustración 15: MacBook Pro [29]

- **Robot físico:** robot desarrollado por el departamento de Automática de la Universidad Carlos III de Madrid. El robot ha sido diseñado para cumplir con las normas de la SSL [5]. [30] [31] [32]

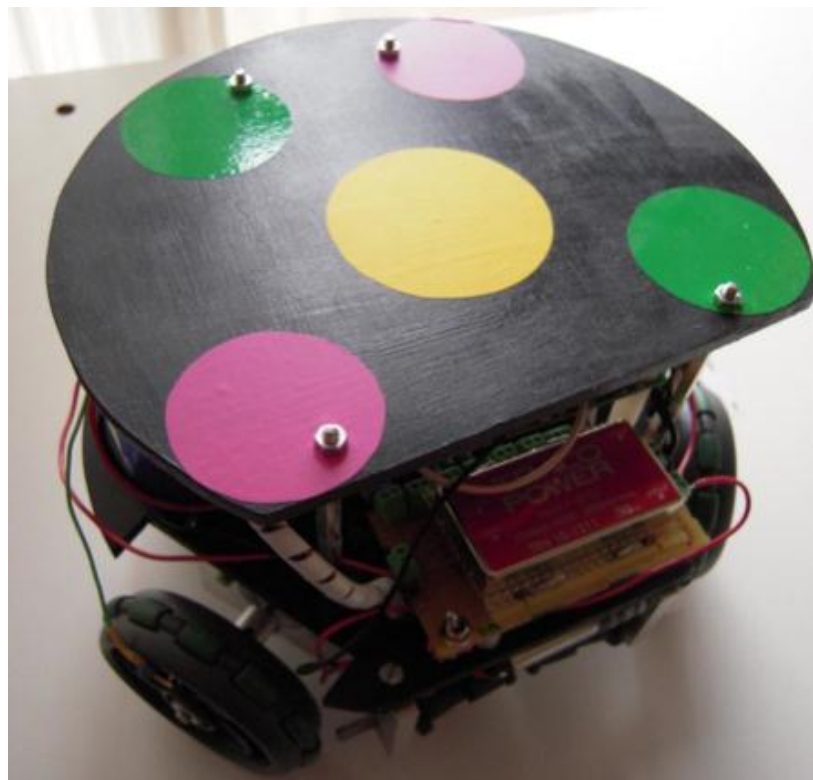


Ilustración 16: robot físico [30] [31] [32]

- **Cámara AVT Stingray:** cámara utilizada para capturar las imágenes del terreno de juego y enviárselas al sistema de visión SSL-Vision [12] de la SSL [5]. [33]



Ilustración 17: AVT Stingray [33]

#### 1.4. Estructura del documento

A lo largo de esta sección se documentará la estructura del documento, aportando una pequeña descripción en cada punto desarrollado.

El primer capítulo corresponde a la presente **introducción** que se ha desarrollado en este punto. Se ha procedido a describir una pequeña introducción del presente trabajo, lo que se pretende, los medios utilizados tanto de software como de hardware y los objetivos de este.

A continuación se ha desarrollado el **estado del arte**, realizando una visión amplia de todo lo que engloba la temática elegida de este PFC, así como todo el trabajo que ya existe en la actualidad. Para ello se han detallado los robots de la actualidad, los paradigmas de la robótica y, finalmente, todo lo relacionado con la RoboCup [4], ahondando en la competición de la que trata este PFC, SSL.

Posteriormente se detalla el **trabajo desarrollado**. De tal manera, se representará todo lo que se ha realizado a lo largo de la creación del presente trabajo, se especifican los requisitos que deberá cumplir el sistema, se explica el funcionamiento del simulador grSim [7] [15], el cliente Java [16] creado, como se ha realizado la inclusión de la arquitectura 3T [36] en el simulador y las habilidades desarrolladas.



En el siguiente capítulo, se indicarán los **experimentos** realizados, así como los resultados aportados por el desarrollo técnico del proyecto, citando las pruebas elaboradas, tanto en el simulador grSim [7] [15] como en los robots físicos.

Seguidamente se desarrollará el **presupuesto** que describe todo el trabajo realizado en este PFC.

Finalmente, se escribirán las **conclusiones** que se han aportado al trabajo realizado y se citará el **trabajo futuro** que se obtiene a partir de este PFC.

Tras ello, se concluirá con **siete anexos**. El primero serán las definiciones y acrónimos necesarios para el pleno entendimiento de esta memoria. El siguiente anexo especifica el conjunto de requisitos de usuario del sistema. Tras ello, se aportarán los manuales de usuario y de referencia. Se citará después la gestión realizada de este trabajo, el reglamento de la SSL [5] de 2011 y, por último, los cambios que se han producido en las reglas de la SSL [5] en 2012 [18] respecto al 2011 [5], en la que se basa todo el trabajo realizado.



## CAPÍTULO 2: ESTADO DEL ARTE

---

## 2. Estado del Arte

A continuación se proporciona una visión global sobre los aspectos que tienen relación con el proyecto. Se hará una breve introducción sobre las distintas arquitecturas de los robots, profundizando en la utilizada para el desarrollo de este proyecto. Además, se proporcionará información sobre la RoboCup [4], profundizando en la SSL [5] ya que es la principal que se trata en este proyecto.

### 2.1. Robots y Robótica en la Actualidad

El diccionario de la Real Academia de la Lengua Española, define un robot como una máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a personas [1]. Los robots pueden ser entidades virtuales o mecanismos físicos.

La robótica es la ciencia que se encarga del estudio y desarrollo de robots. Esta ciencia combina la utilización de distintas disciplinas como la inteligencia artificial, la electrónica, la mecánica y la informática [2].

Actualmente, la robótica es aplicada en varios ámbitos como la industria o el comercio. Aunque se está investigando sobre la utilización de robots en ámbitos como la medicina, la aviación, la educación, etc.

### 2.2. Paradigmas de la Robótica

Un paradigma es un conjunto de técnicas que caracterizan el enfoque que se le da a un problema. En la actualidad existen distintos paradigmas aplicados a la robótica como el reactivo, el jerárquico o el híbrido.

#### 2.2.1. Paradigma Reactivo

El paradigma reactivo trata de emular ciertos aspectos de la biología. Este paradigma se basa en las conductas que consisten en respuestas a estímulos externos.

Para poder aplicar este paradigma, es necesario que el robot esté dotado de sensores para captar información del exterior y, de esta manera, formular la información en términos matemáticos. [3]



Ilustración 18: paradigma reactivo

Toda la información recogida por los sensores se utiliza para llevar a cabo una acción concreta enviando información a los actuadores del robot.

El tiempo de respuesta debe ser lo más breve posible para que el robot reaccione de forma rápida y correcta a los estímulos externos.

### 2.2.2. Paradigma Jerárquico

El paradigma jerárquico es el más antiguo. Este se basa en la forma que tienen los seres humanos para solventar los problemas. Este paradigma es conocido también como paradigma percepción, planificación y acción, debido a las operaciones que lo componen. Las tres operaciones se realizan siempre de forma secuencial, siguiendo el mismo nombre en el que han sido nombradas las operaciones.

La percepción es una operación fundamental en el control de robots autónomos, ya que es la forma de obtener información del entorno físico del robot.

La fase de planificación es la encargada de elegir las acciones que se llevarán a cabo en función de la información obtenida de la etapa de percepción. Para permitir efectuar la

planificación, las percepciones deberán haber sido procesadas con anterioridad y transformadas en una abstracción del entorno.

En la fase de acción se realizan acciones de alto nivel que se seleccionaron en la fase de planificación, traduciéndose en los movimientos de los actuadores del robot. [3]

La ilustración 14 muestra una representación de este paradigma:



Ilustración 19: paradigma jerárquico

### 2.2.3. Paradigma Híbrido

El paradigma híbrido trata de llegar a un compromiso entre las tareas deliberativas y las reactivas. Este paradigma se compone de un modelo deliberativo y de un modelo reactivo.

El modelo deliberativo lleva a cabo la planificación de alto nivel. El modelo reactivo entra en funcionamiento cuando es necesario realizar una acción inmediata, como por ejemplo, para evitar colisiones.

Otro tipo de modelo híbrido es el de planificación, percepción – acción. En este modo el robot planifica como descomponer una tarea en subtareas y elige cuales son los comportamientos más adecuados para cumplir con estas tareas.

Se muestra a continuación una representación de este paradigma:





Ilustración 20: paradigma híbrido

### 2.3. Arquitectura 3T

En el año 1997 aparece por primera vez la arquitectura 3T [36] de la mano de Bonasso, con el objetivo de hacerla funcionar en distintos robots independientemente del entorno y sin tener en cuenta el hardware de los mismos. Esta arquitectura se engloba dentro del paradigma híbrido explicado anteriormente.

En el esquema de la ilustración 21 se muestra la estructura que sigue esta arquitectura. En dicho esquema se observa que la parte superior de la arquitectura hace referencia al nivel deliberativo y en este se sitúa el planificador, el cual crea un plan que se ejecutará en función de las condiciones del entorno.

La siguiente capa se corresponde con el secuenciador, el cual se basa en dividir el plan en distintas actividades llamadas acciones.

La última capa hace referencia al controlador, el cual se sitúa en el nivel reactivo de la arquitectura y se encarga de la ejecución de las habilidades.

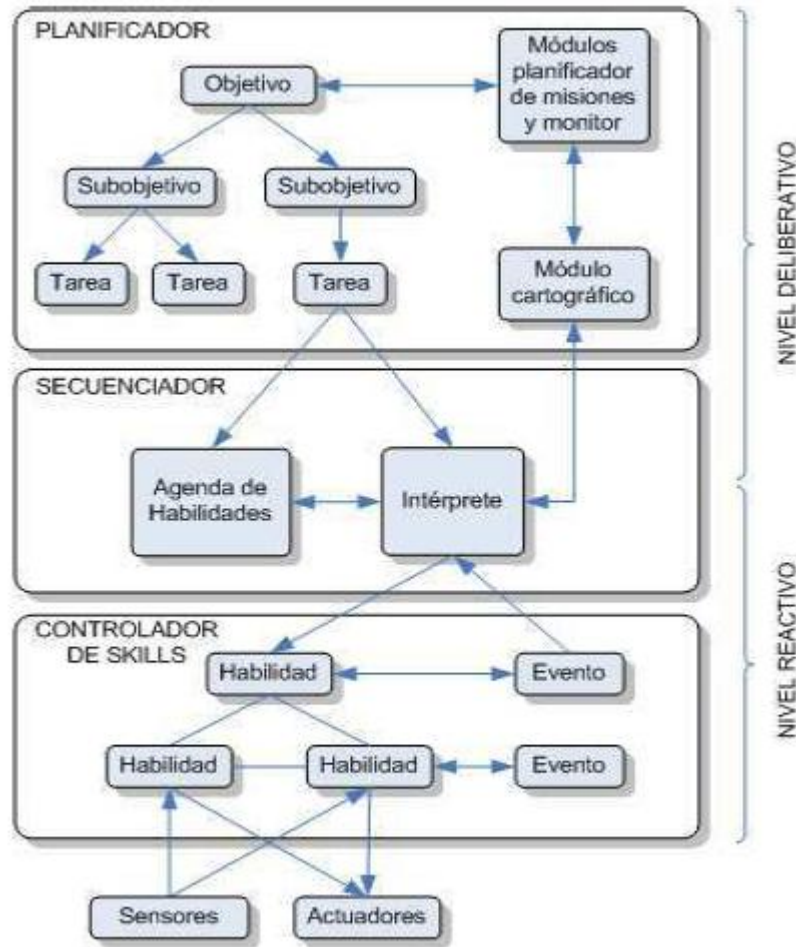


Ilustración 21: arquitectura 3T [36]

## 2.4. RoboCup

La RoboCup [4] es una competición internacional cuyo objetivo es avanzar en campos como la robótica y la inteligencia artificial.



Ilustración 22: logotipo RoboCup [4]

La idea de crear una competición con robots que juegan al fútbol se planteó por vez primera en el documento científico “On Seeing Robots” realizado por el profesor Alan Mackworth en 1992 [10].

Esta competición se inició en 1997, con el objetivo de crear equipos de robots que sean capaces de ganar al equipo campeón de la copa mundial de la FIFA para el año 2050.

Con el transcurso del tiempo y dado que este objetivo está relacionado con muy diferentes campos, han ido apareciendo nuevas categorías, las cuales son: RoboCup Rescue, RoboCup Junior y RoboCup Soccer [4].

#### **2.4.1. Competiciones RoboCup**

En este punto se detallarán las competiciones que incluye la RoboCup [4].

##### **2.4.1.1. RoboCup Rescue**

La RoboCup Rescue es una de las categorías que ofrece la competición.



**Ilustración 23: logotipo de la RoboCup Rescue [4]**

El objetivo de esta categoría es poner a prueba a los robots en labores de búsqueda y salvamento de víctimas sobre un terreno desfavorable. En esta categoría los robots pueden ser autónomos o guiados remotamente.

En las pruebas, un equipo de robots debe ser capaz de encontrar el camino adecuado en la zona indicada y crear un mapa con las posiciones de los distintos obstáculos existentes y de las víctimas. [4]

Existen dos ligas dentro de la competición:

- **Simulación:**



Ilustración 24: liga de simulación de RoboCup Rescue [4]

- **Robots reales:**

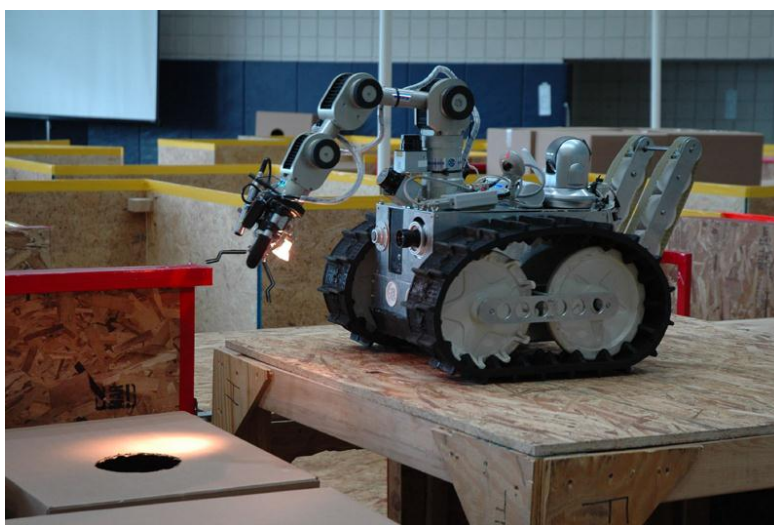


Ilustración 25: liga de robots reales de RoboCup Rescue [4]

#### **2.4.1.2. RoboCup Junior**

Es una iniciativa orientada a la educación y a acercar los objetivos de la RoboCup [4] a estudiantes de educación primaria y secundaria.



Ilustración 26: logotipo RoboCup Junior [4]

Esta categoría cuenta con varias ligas:

- **Soccer:** se juegan partidos de 2 contra 2.

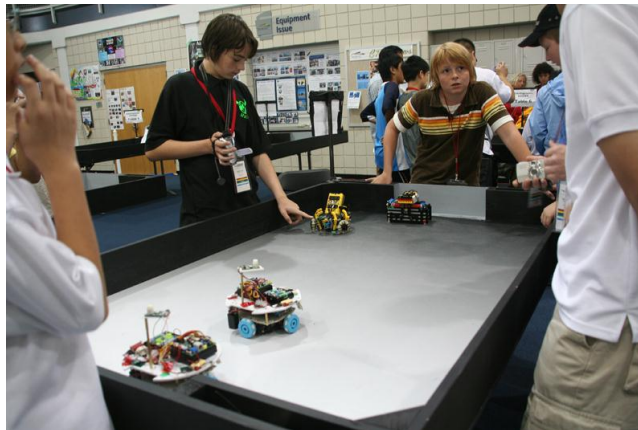


Ilustración 27: liga soccer de RoboCup Junior [4]

- **Rescue:** se utilizan reglas similares a la competición global, aunque el escenario es más sencillo.

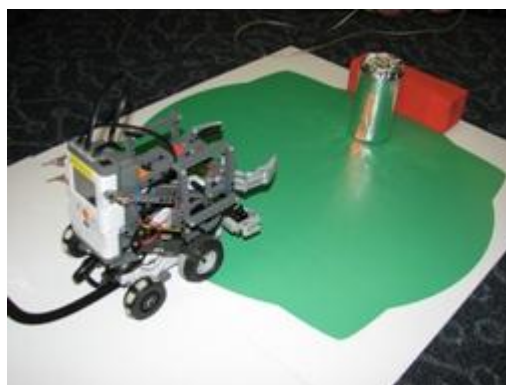


Ilustración 28: liga rescue de RoboCup Junior [4]

- **Dance:** los participantes escogen una canción y robots autónomos bailan al son de la música. [4]



Ilustración 29: liga dance de RoboCup Junior [4]

#### 2.4.1.3. *RoboCup @Home*

Esta categoría apareció en 2006 en la competición celebrada en Bremen (Alemania). Esta competición consiste en la aplicación de robots autónomos en la vida diaria y en las relaciones de los hombres y los robots.

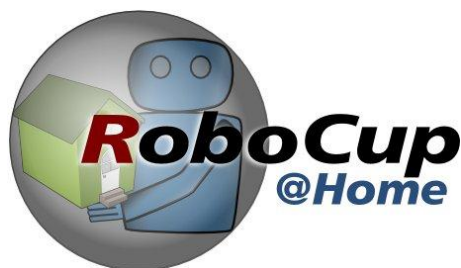


Ilustración 30: logotipo RoboCup @Home [4]

El escenario utilizado es la vida real y es donde deben demostrarse las capacidades del robot. [4]





Ilustración 31: RoboCup @Home [4]

#### **2.4.1.4. RoboCup Soccer**

Esta es la categoría principal de la competición la cual se basa en partidos de fútbol entre dos equipos de robots, donde el objetivo principal es el desarrollo de robots y sistemas multiagente en entornos dinámicos.

Dentro de esta categoría existen distintas ligas en función de la morfología de los robots y cada una de estas ligas posee sus propias reglas. [4]

##### **Simulation League**

Esta es una de las ligas más antiguas creadas. Se basa en la simulación de partidos.

En esta liga no existen robots reales, pero los espectadores pueden seguir el partido en una pantalla. Cada robot simulado puede tener su propia estrategia de juego, es decir, cada equipo está compuesto por un conjunto de programas. [4]

Existen dos subligas según las dimensiones:

- **2D:** simulación de robots en dos dimensiones.



Ilustración 32: liga simulación en 2D de RoboCup Soccer [4]

- **3D:** aportando una dimensión más, complica considerablemente el problema, los robots que se simulan son los NAO de Aldebarán Robotics [34].



Ilustración 33: liga simulación en 3D de RoboCup Soccer [4]

### Standard Platform League



La liga de plataforma estándar parte de un robot físico previamente elegido, por lo que los equipos se tienen que centrar exclusivamente en el software, ya que los robots solo tienen que comprarlos y ponerlos a funcionar.

Cuando esta liga comenzó, en 1998, un año después de la aparición de la RoboCup [4], se utilizaban los robots con forma de perro de Sony, llamados AIBO [35], sin embargo, en 2006 Sony cancela la fabricación de estos robots. La liga duró dos años más, hasta que finalmente se empezó a utilizar el NAO de Aldebaran Robotics [34].

En la ilustración se puede ver el robot AIBO de Sony [35]:

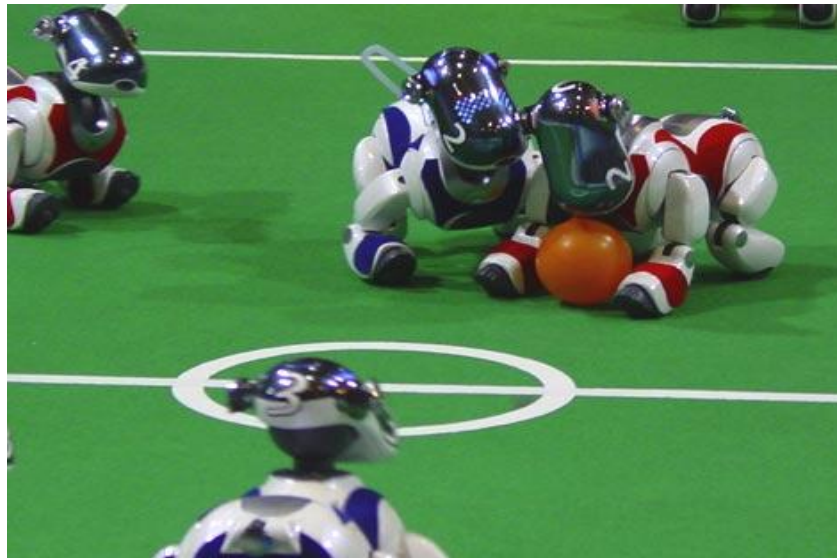


Ilustración 34: AIBO de Sony usados en la liga de plataforma estándar [4] [35]

Y en este por el que finalmente fue sustituido, el NAO de Aldebaran Robotics [34]:

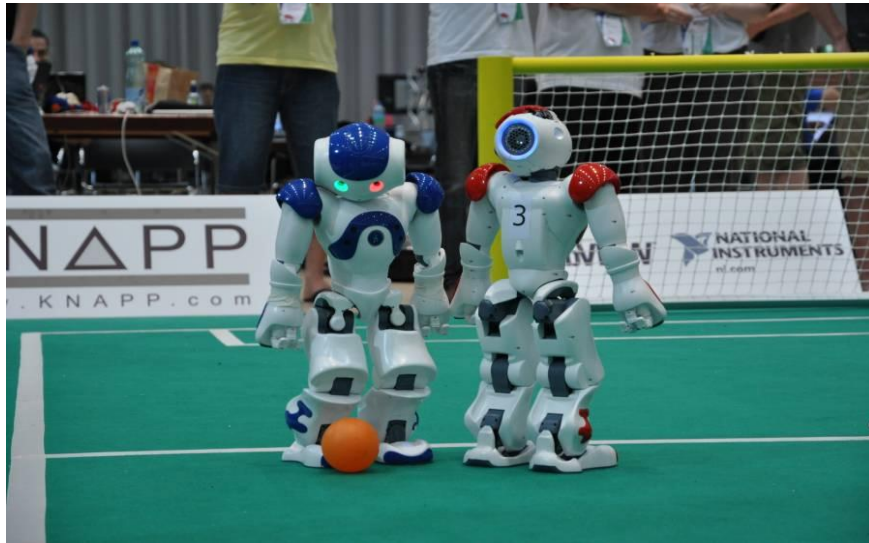


Ilustración 35: liga de plataforma estándar de RoboCup Soccer [4] [34]

### Humanoid League

Los robots de esta liga tienen una morfología humanoide, estos robots tratan de emular a los seres humanos en la forma de actuar, así como en la estructura.

Esta liga está dividida a su vez en 3 subligas en función del tamaño de los robots [4]:

- Entre 30 y 60 centímetros:



Ilustración 36: liga de robots humanoides entre 30 y 60 cm de RoboCup Soccer [4]

- **Entre 100 y 120 centímetros:**



Ilustración 37: liga de robots humanoides entre 100 y 120 cm de RoboCup Soccer [4]

- **Más de 130 centímetros:**

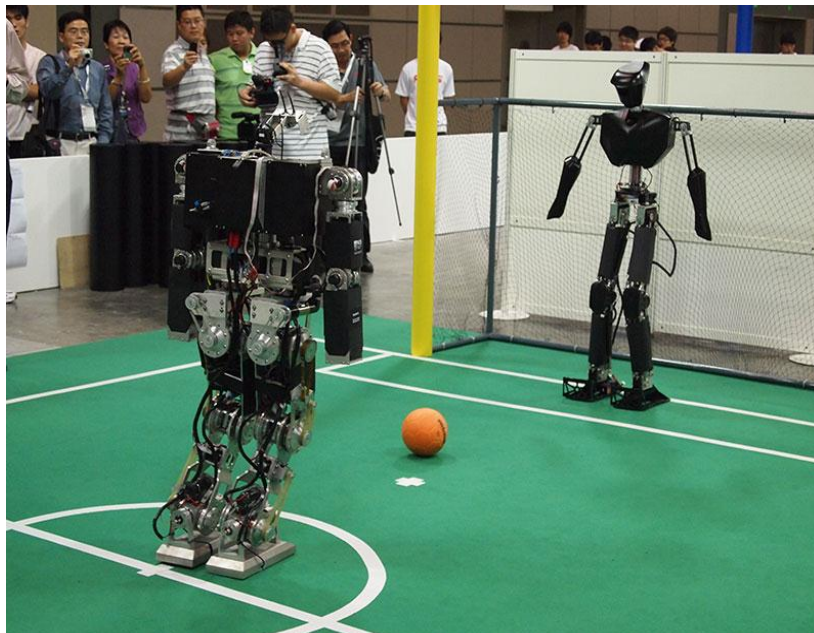


Ilustración 38: liga de robots humanoides más de 130 cm de RoboCup Soccer [4]



### **Middle Size League**

Los robots de esta liga poseen unas dimensiones de 50 cm de diámetro como máximo en equipo de 5 robots. Los partidos se juegan con balones del tamaño oficial utilizado por la FIFA. Los robots están equipados con sensores y puede utilizarse una conexión wifi para comunicarse con los robots.

El enfoque de la investigación se basa en la autonomía de los robots y la cooperación entre los mismos. [4]



Ilustración 39: liga de robots medianos de RoboCup Soccer [4]

### **Small Size League**

Esta liga es la base del presente PFC por lo que se explicará de forma extensa en el siguiente punto.

#### **2.4.2. Small Size League**

Conocida comúnmente por SSL [5], esta es una de las ligas más antiguas creadas. Se utilizan equipos de 5 robots con unas dimensiones de 18 cm de diámetro y 15 cm de altura, y una pelota de golf de color naranja.

Los robots son identificados por 2 cámaras conectadas a un ordenador que enviará la información de la pelota y los robots a los ordenadores de los dos equipos. [4]



Ilustración 40: liga de robots pequeños de RoboCup Soccer [4]

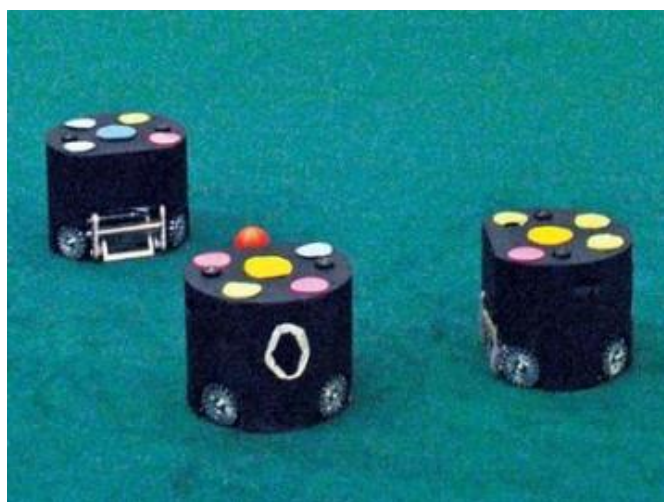


Ilustración 41: ejemplo de liga de robots pequeños de RoboCup Soccer [4]

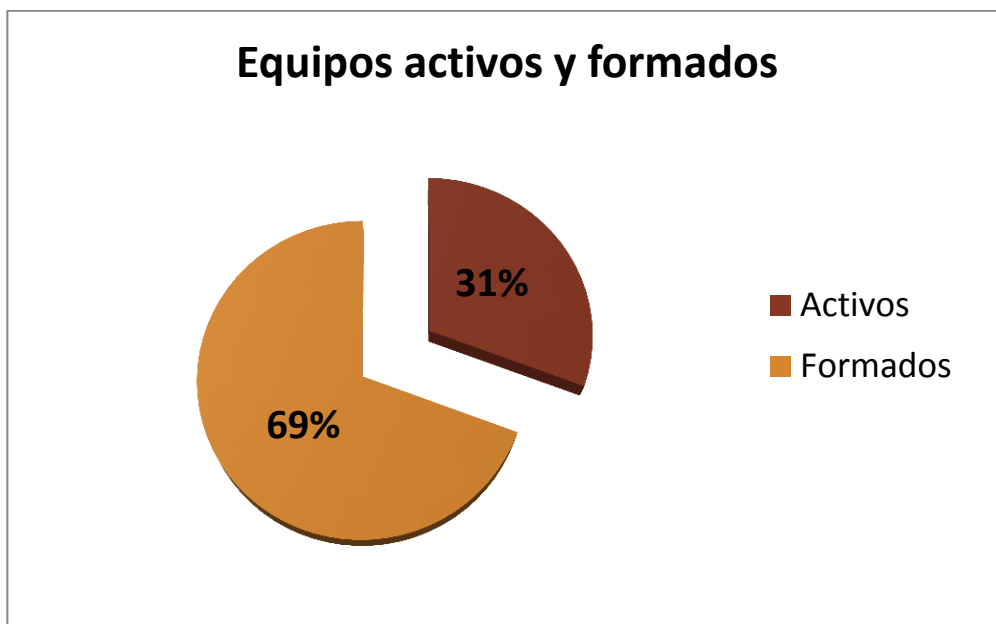
Este PFC va a tratar exclusivamente sobre el desarrollo de una arquitectura para esta liga, por lo que se hace un análisis exhaustivo. Existen actualmente los siguientes equipos en activo que participan, exactamente 23:

- BRocks de Turquía.
- CMDragons de Estados Unidos.
- Eagle Knights – Robobulls de Méjico y Estados Unidos.
- ER-Force de Alemania.



- Immortals de Irán.
- KIKS de Japón.
- MRL de Irán.
- Odens de Japón.
- Omid de Irán.
- Owaribito-CU de Japón.
- Parsian de Irán.
- RFC Cambridge de Estados Unidos.
- RoboDragons de Japón.
- RoboJackets de Estados Unidos.
- RoboFEI de Brasil.
- Robopet de Brasil.
- RoboTurk de Turquía.
- Skuba de Tailandia.
- Strive de China.
- Thunderbots de Canadá.
- TPOTS de Singapur.
- Wright Eagle de China.
- ZjuNlict de China. [5]

Por otro lado, existen otros equipos, 52, que no han llegado a participar, debido a que no han logrado pasar las pruebas previas que realiza la organización. Considerando este, en la ilustración 42 se muestra la relación entre equipos que participan y que no llegan a poder participar.



**Ilustración 42: relación entre equipos en activo y formados [5]**

Además, el gráfico de la ilustración 43 representa la relación de los equipos en activo y los equipos formados que no han llegado a participar asociados a su país de procedencia. Dicho gráfico permite conocer qué países están más en activo en esta competición.

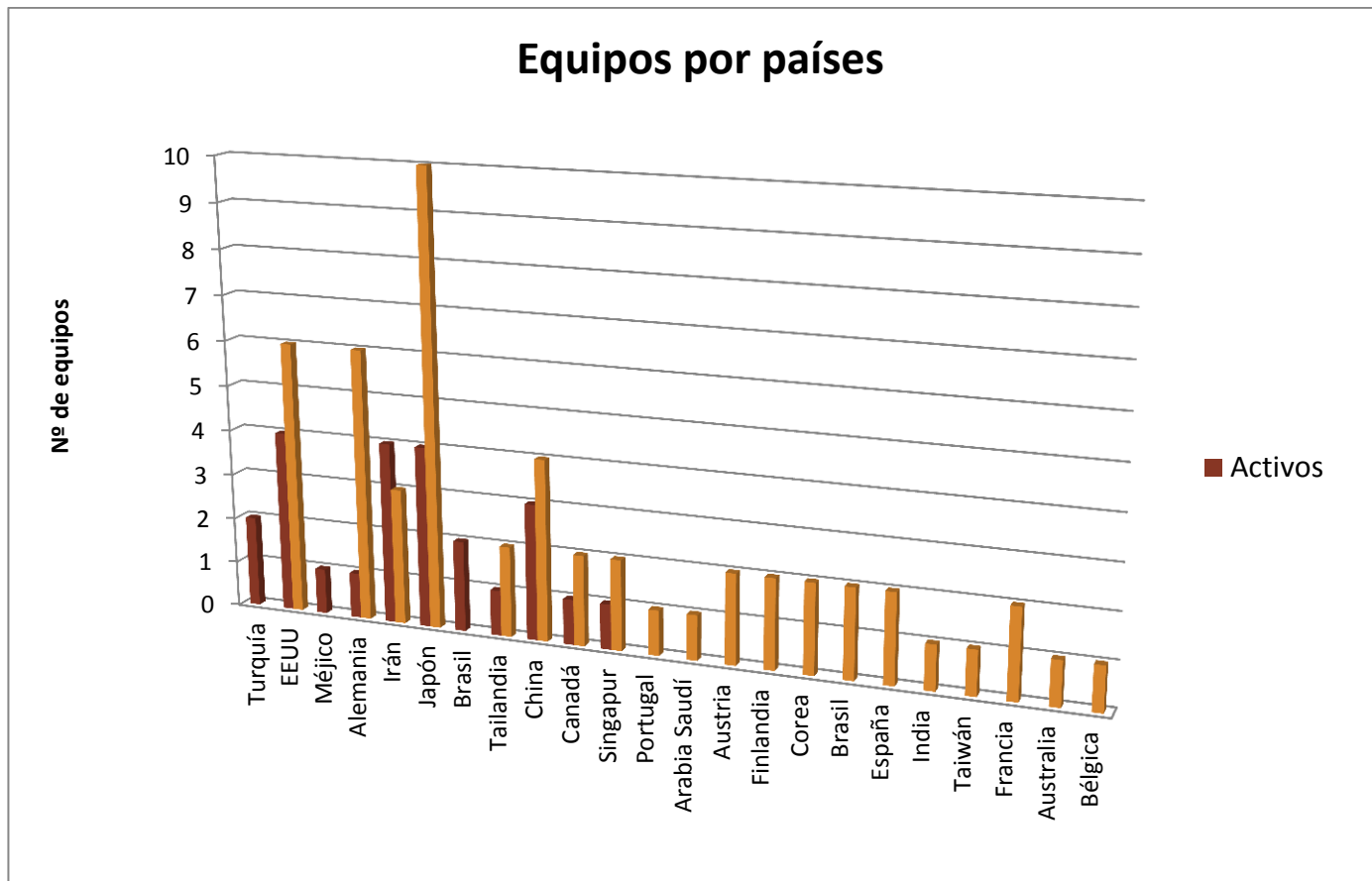


Ilustración 43: participación por países en la SSL [5]



Dentro de los equipos participantes en esta liga, existe un ranking que establece las victorias que han ido consiguiendo los equipos con el paso de las ediciones y que se detallan en la tabla 1.

Posición	Equipo	Primeras posiciones	Segundas posiciones	Terceras posiciones	Cuartas posiciones
1ª	CMDragons	1997 1998 2006 2007	2008 2010		
2ª	Cornell Big Red	1999 2000 2002 2003	2005	2003	
3ª	Skuba	2009 2010 2011		2008	
4ª	FU Fighters	2004 2005	1999 2000 2002	2003	2001
5ª	PasmaZ	2008	2007	2006 2009	
6ª	Lucky Star	2001		1999 2000 2002 2004	
7ª	RoboRoos		1998 2003 2004		
8ª	Field Rangers		2001	2005	2006
9ª	5dpo		2006	1998	
10ª	RoboDragons		2009	2007	
11ª	Naist		1997		
12ª	Immortals		2011		
13ª	MRL			2010 2011	
14ª	CMRoboDragons				2004 2005
15ª	ZjuNlict				2007 2008
16ª	KIKS				2010 2011
17ª	Cambridge				1998
18ª	Odeons				2009

Tabla 1: ranking de victorias en SSL [5]

### 2.4.3. Reglas para la SSL

La liga SSL [5], así como el resto de ligas de la RoboCup [4] consta de unas reglas que permiten explicar y acotar el pleno funcionamiento de esta liga. Para ello, todos los años se publican unas reglas que es obligatorio seguir para poder participar.

Estas reglas constan de diecisiete leyes, además de dos anexos explicativos, se muestra a continuación un resumen, que puede ser ampliable en el “Anexo VI: Reglamento SSL de 2011”:

- **LEY 1:** se establecen las dimensiones concretas del terreno de juego.
- **LEY 2:** descripción del balón.
- **LEY 3:** definición del número de robots por cada equipo.
- **LEY 4:** establecimiento de las características del equipo de robots.
- **LEY 5:** competencias del árbitro.
- **LEY 6:** competencias del árbitro asistente.
- **LEY 7:** duración del partido completo.
- **LEY 8:** cuándo se inicia y se reanuda el juego.
- **LEY 9:** cuándo el balón se para y está en juego.
- **LEY 10:** cuándo se puntúa.
- **LEY 11:** cuándo se produce fuera de juego.
- **LEY 12:** establece cuándo se considera falta y cuándo una conducta antideportiva.
- **LEY 13:** cuándo se asigna un tiro libre.

- **LEY 14:** define cuándo se realiza un tiro de penalti.
- **LEY 15:** define cuándo se hace un saque de banda.
- **LEY 16:** describe cuándo se hace un saque de puerta.
- **LEY 17:** cuándo se produce un saque de esquina.
- **Apéndice A:** describe las reglas de competencia de los participantes.
- **Apéndice B:** describe el sistema de visión y el rol de experto en visión.

#### 2.4.4. Simuladores de la SSL

Para desarrollar el sistema de control de los robots, es necesario la utilización de un simulador que aplique las reglas de la SSL [5], para ello, se han encontrado los siguientes simuladores:

- **Tewnta:** el primero simulador es Tewnta, es un simulador sencillo y fácil de usar. El objetivo es que en un futuro pueda ser elegido como simulador estándar para el desarrollo de equipos de la SSL [5]. Aunque en la actualidad no se está trabajando en su desarrollo y no está excesivamente elaborado, por lo que es complicado utilizarlo de forma fidedigna, ya que no tiene soporte y no hay un equipo detrás que trabaje en él [6].



Ilustración 44: logotipo de Twenta [6]

- **grSim [7] [15]:** simulador 3D desarrollado por el equipo Parsian Robotic. Es compatible con los paquetes enviados por el sistema SSL-Vision [12] y es uno de los más avanzados en la actualidad. [7] Posee una interfaz gráfica muy intuitiva para seguir el desarrollo del juego.



Ilustración 45: logotipo grSim [7] [15]

- **Carnegie Mellon University:** simulador implementado en C++ [17], desarrollado por el equipo CMDragons. [8] Este simulador no se encuentra disponible para el público.



Ilustración 46: logotipo del equipo CMDragons [8]

- **RoboCup-Simulator:** simulador poco desarrollado actualmente. No está terminado. [9] El objetivo era crear un simulador libre para equipos que no posean uno. La versión actual del proyecto no cumple con toda la funcionalidad básica que debería reunir un simulador.



## CAPÍTULO 3: TRABAJO REALIZADO

---

### 3. Trabajo realizado

El principal objetivo del proyecto es el desarrollo de una arquitectura 3T que implemente ciertas habilidades básicas para los robots de la SSL [5] de la RoboCup [4]. Para ello es necesaria la utilización de un simulador de esta liga para poder producir las habilidades y trabajar en su elaboración de forma cómoda y sencilla. En este caso se ha hecho uso del simulador grSim [7] [15] y a partir de este se han desarrollado las distintas habilidades que posteriormente se utilizarán en los robots reales.

En este punto se explican los distintos puntos de desarrollo del proyecto. Cabe destacar los siguientes puntos de desarrollo técnico que se han llevado a cabo y que se comentarán en los sucesivos puntos:

- Simulador grSim [7] [15] de la SSL [5].
- Cliente sencillo Java [16] de envío y recepción de paquetes UDP.
- Integración de habilidades a arquitectura híbrida en Java [16].
- Integración de las habilidades en robots físicos.

#### 3.1. Requisitos de Usuario

El primer paso para llevar a cabo el desarrollo del proyecto es definir los requisitos de usuario que deberán ser cumplidos.

Los requisitos de usuario se especifican en las tablas contenidas en el “Anexo II: Requisitos de Usuario”. Estos requisitos son clasificados por los siguientes tipos:

- Simulador grSim.
- Cliente Java de envío.
- Cliente Java de recepción.
- Arquitectura 3T.
- Habilidades de SSL.

En los requisitos se especifican todas las funcionalidades que deben cumplirse en el desarrollo del proyecto. En estas tablas se muestra el identificador de los requisitos, el nombre, la descripción, la necesidad, la prioridad, la estabilidad y la claridad.

### 3.2. Simulador grSim

El simulador utilizado para desarrollar las habilidades básicas de los robots ha sido el simulador grSim [7] [15] desarrollado por el equipo Parsian Robotic. Este simulador 3D cuenta con muchas ventajas por las que se ha decidido utilizar:

- Es un simulador de código abierto.
- Los paquetes que envía el simulador tienen el mismo formato que los enviados por el programa SSL-Vision. [12]
- La recepción de comandos se basa en la utilización de Google Protocol Buffers. [13]
- Simulación real y eficiente utilizando Open Dynamics Engine. [14]
- Soporte para múltiples robots.
- Interfaz sencilla e intuitiva.
- Modificación de la configuración en tiempo de ejecución.

#### 3.2.1. Funcionamiento del Simulador

El simulador grSim [7] [15] tiene un funcionamiento idéntico al del programa SSL-Vision [12], el cual se utiliza para la comunicación con los robots reales.

El simulador envía paquetes UDP por multicast, que contienen la información de la pelota y de todos los robots que intervienen en el juego. Para decodificar los paquetes se utiliza Google Protocol Buffers [13], el cual sirve para codificar información de forma estructurada, eficiente y de forma muy sencilla. Esto facilita en gran medida el intercambio de información entre el sistema y el simulador.

La siguiente imagen muestra un pequeño resumen con las características del simulador grSim [7] [15]:



Ilustración 47: funcionamiento grSim [7] [15]

La estructura de un paquete enviado por el simulador se muestra en la tabla 2.

Parámetros	Paquete
<b>Información general del paquete:</b> <ul style="list-style-type: none"> <li>Número de frame.</li> <li>Tiempo de captura.</li> <li>Tiempo de envío.</li> <li>Identificador de la cámara.</li> </ul>	<pre>detection {   frame_number: 5657   t_capture: 90.944   t_sent: 90.944   camera_id: 0 }</pre>
<b>Información de la pelota:</b> <ul style="list-style-type: none"> <li>Confianza.</li> <li>Posición X.</li> <li>Posición Y.</li> <li>Altura</li> <li>Píxel en la posición X.</li> <li>Píxel en la posición Y.</li> </ul>	<pre>balls {   confidence: 0.9850978   x: 0.0   y: 0.0   z: 21.499933   pixel_x: 0.0   pixel_y: 0.0 }</pre>
<b>Información de los robots amarillos:</b> <ul style="list-style-type: none"> <li>Confianza.</li> <li>Identificador del robot.</li> <li>Posición X.</li> <li>Posición Y.</li> <li>Orientación.</li> <li>Píxel en la posición X.</li> <li>Píxel en la posición Y.</li> </ul>	<pre>robots_yellow {   confidence: 1.0   robot_id: 0   x: 1000.0206   y: 750.0   orientation: 3.1415925   pixel_x: 1000.0206   pixel_y: 750.0 }</pre>
<b>Información de los robots azules:</b> <ul style="list-style-type: none"> <li>Confianza.</li> <li>Identificador del robot.</li> </ul>	<pre>robots_blue {   confidence: 1.0   robot_id: 0 }</pre>



<ul style="list-style-type: none"> <li>• <b>Posición X.</b></li> <li>• <b>Posición Y.</b></li> <li>• <b>Orientación.</b></li> <li>• <b>Píxel en la posición X.</b></li> <li>• <b>Píxel en la posición Y.</b></li> </ul>	<pre>x: -999.9981 y: 750.0 orientation: 1.4901161E-8 pixel_x: -999.9981 pixel_y: 750.0 }</pre>
---	--

Tabla 2: paquete recibido del simulador grSim [7] [15]

Los comandos que se envían en paquetes UDP utilizan la estructura destacada en la tabla 3.

Parámetros	Paquete
<b>Información general del comando enviado:</b> <ul style="list-style-type: none"> <li>• <b>Tiempo de retardo.</b></li> <li>• <b>Pertenece el robot al equipo azul o amarillo.</b></li> </ul>	<pre>commands {   timestamp: 0.0   isteamyellow: true }</pre>
<b>Información del robot del comando enviado:</b> <ul style="list-style-type: none"> <li>• <b>Identificador del robot.</b></li> <li>• <b>Velocidad de disparo X.</b></li> <li>• <b>Velocidad de disparo Z.</b></li> <li>• <b>Velocidad tangencial.</b></li> <li>• <b>Velocidad normal.</b></li> <li>• <b>Velocidad angular.</b></li> <li>• <b>Activación del spinner.</b></li> <li>• <b>Envío de la velocidad a las ruedas.</b></li> <li>• <b>Velocidad de la rueda 1.</b></li> <li>• <b>Velocidad de la rueda 2.</b></li> <li>• <b>Velocidad de la rueda 3.</b></li> <li>• <b>Velocidad de la rueda 4.</b></li> </ul>	<pre>robot_commands {   id: 0   kickspeedx: 0.0   kickspeedz: 0.0   veltangent: 0.0   velnormal: 0.0   velangular: 0.0   spinner: false   wheelsspeed: false   wheel1: 0.0   wheel2: 0.0   wheel3: 0.0   wheel4: 0.0 }</pre>

Tabla 3: paquete enviado al simulador grSim [7] [15]

### 3.2.2. Interfaz del Simulador

La interfaz gráfica del simulador grSim [7] [15] se compone de los siguientes elementos:

- Una representación gráfica del terreno de juego en la que se representa el estado de la simulación en tiempo real.



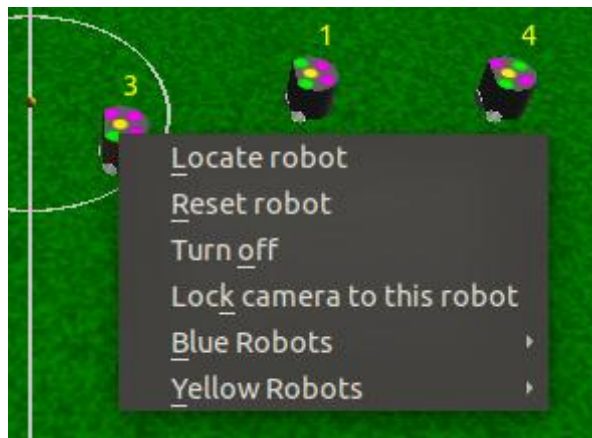
- La columna izquierda muestra información sobre los robots, pelota, la física, la comunicación. Además de una zona en la que se muestra cual es el robot actual y se pueden cambiar parámetros como la posición, la activación del robot, resetear, etc.
- En la parte inferior se muestra una línea de mensajes en la que se muestra información como los puertos de escucha o la dirección multicast.

La ilustración 48 muestra la interfaz del simulador.



Ilustración 48: interfaz gráfica del simulador grSim [7] [15]

Además, es posible cambiar parámetros de un robot pulsando con el botón derecho sobre el robot deseado como se muestra en la ilustración 49.



**Ilustración 49: modificación robot**

Como se observa en la imagen se despliega un menú en el que se puede:

- Modificar la posición del robot.
- *Resetear* el robot seleccionado.
- Desactivar el robot seleccionado.
- Fijar la cámara en el robot seleccionado.
- Cambiar la formación de los robots del equipo azul y amarillo.

Para cambiar la formación de los equipos o sacar los robots del terreno de juego se utiliza el menú desplegable de la ilustración 50.

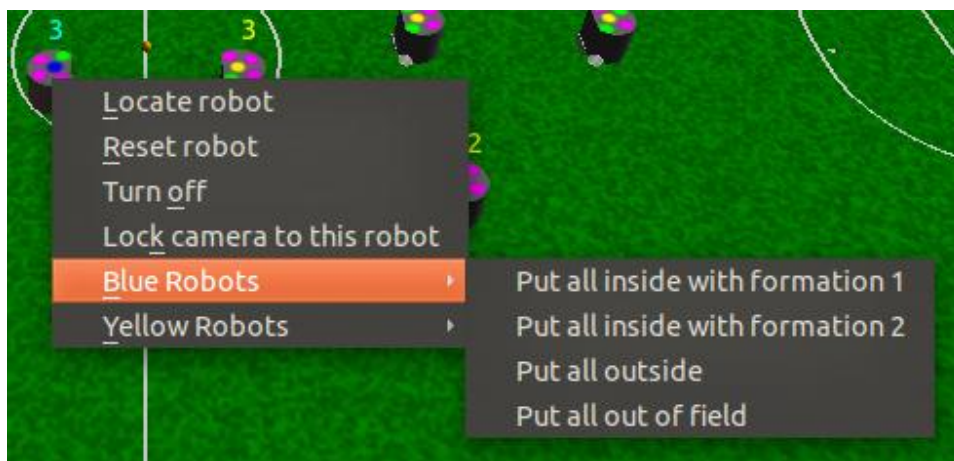


Ilustración 50: cambio de formación de los robots

También se pueden cambiar otras opciones como:

- La localización de la pelota.
- La localización del robot seleccionado.
- La posición de la cámara.
- La formación de los robots de los dos equipos.

Para ello debe pulsarse con el botón derecho sobre la superficie de juego como se muestra a continuación:

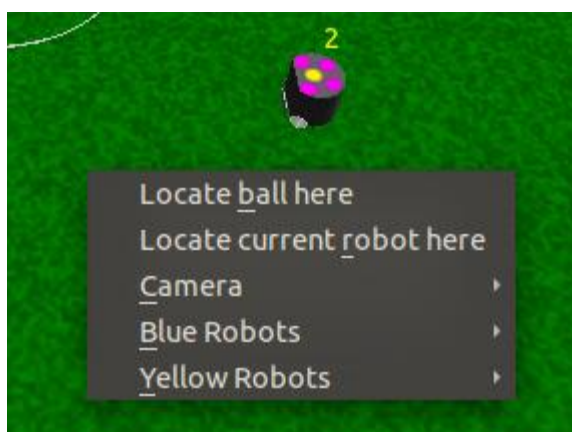


Ilustración 51: cambio de opciones

Si se pulsa sobre la opción de cambiar la posición (Set Position) se muestra el siguiente menú en el que se puede modificar la posición X, la posición Y y la orientación:



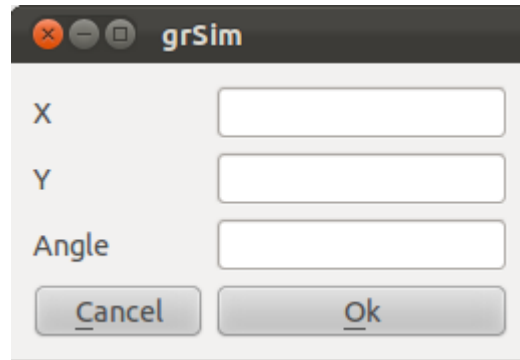


Ilustración 52: cambio de posición y orientación

### 3.2.3. Representación Gráfica

La representación gráfica se realiza en el terreno de juego tomando como punto origen de coordenadas el punto del centro del campo, el cual se corresponde con la posición (0, 0). En la siguiente imagen se muestra el signo de las posiciones en el terreno de juego:

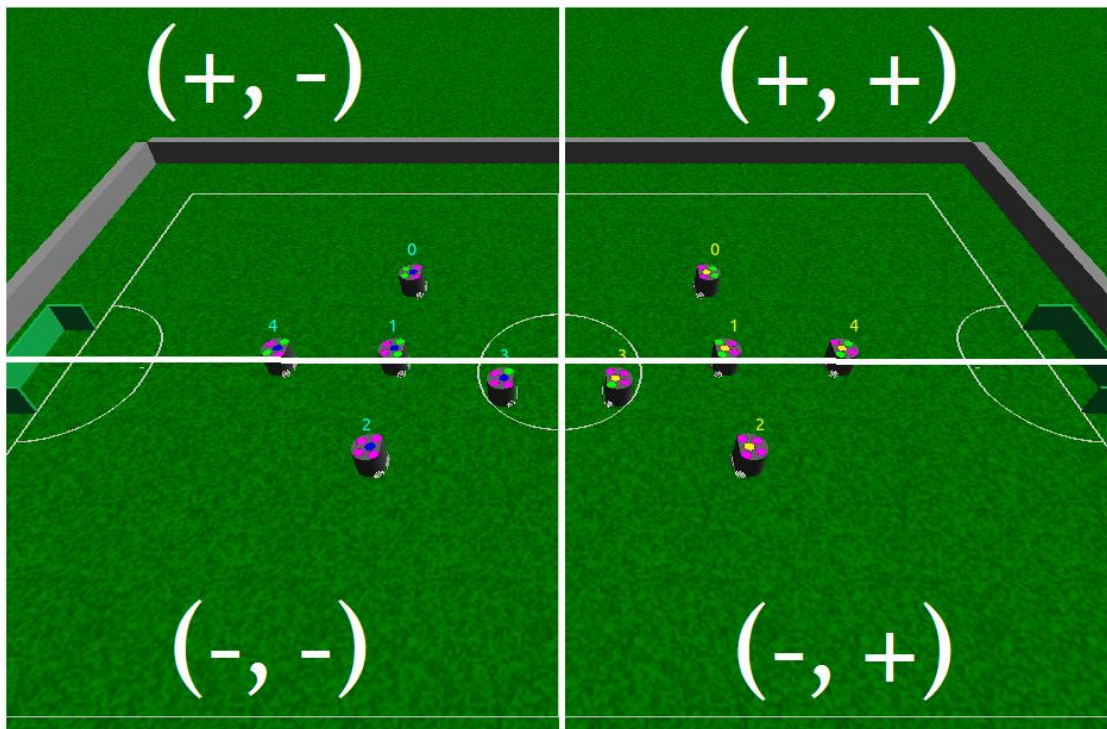


Ilustración 53: signo de las posiciones en el terreno de juego

Respecto a la orientación que toman los robots, se calcula como se muestra en la siguiente imagen:

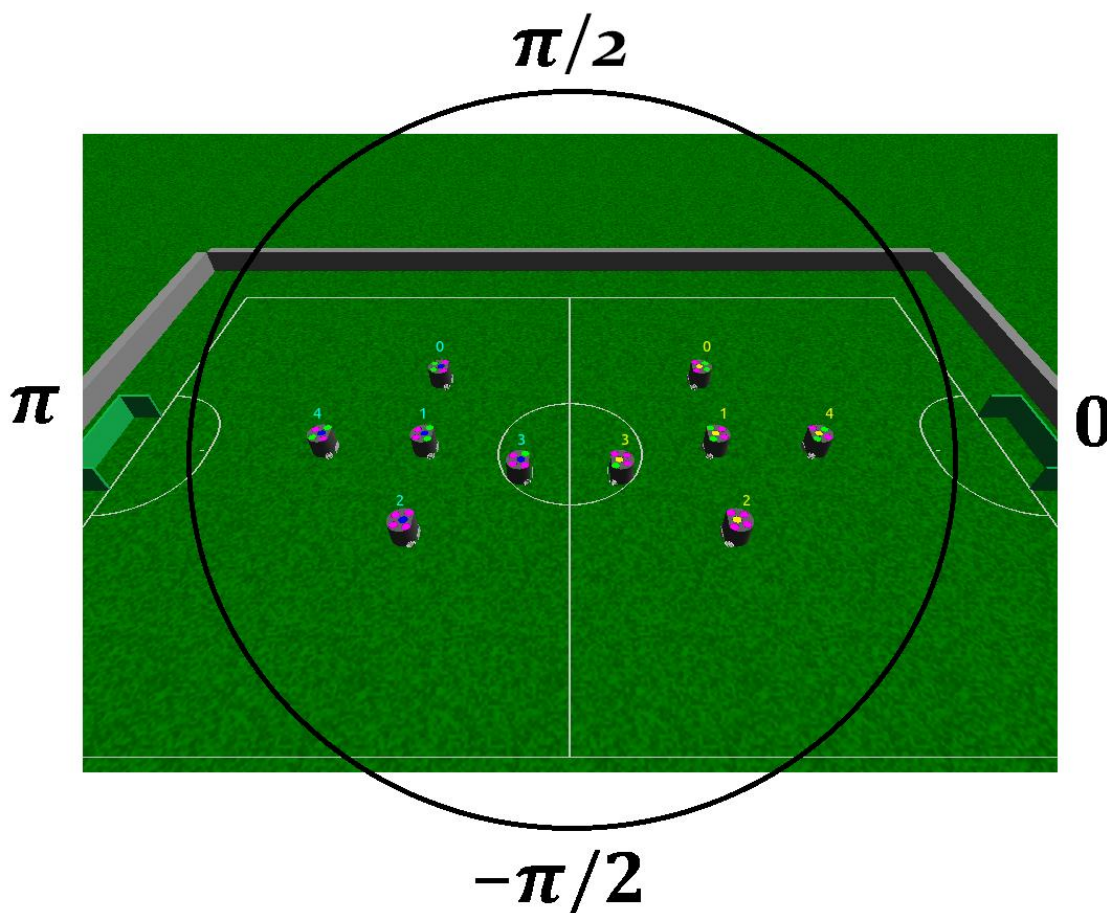


Ilustración 54: orientación en el terreno de juego

Si se desea modificar la orientación de los robots utilizando la interfaz que se mostró en la ilustración 52 debe introducirse la orientación en grados, por lo tanto, es necesario transformar el valor de radianes a grados.

#### 3.2.4. Adaptación de los Robots a Tres Ruedas

Dado que los robots físicos utilizados tienen 3 ruedas en lugar de 4, ha sido necesario modificar el simulador para adaptarlo y quitar una rueda de los robots. Esto se debe a que por defecto, los robots representados en el simulador grSim [7] [15] disponen de cuatro ruedas, y por lo tanto, es necesario suprimir una rueda de estos robots para simular el robot físico del que se dispone para el desarrollo del proyecto.

En la siguiente imagen se muestra como es la representación de los robots en el simulador con 4 ruedas:



Ilustración 55: robots en el simulador grSim [7] [15] con 4 ruedas

Una vez adaptados los robots del simulador a tres ruedas, estos quedan representados en el simulador como se muestran en la ilustración 56.



Ilustración 56: robots en el simulador grSim [7] [15] con 3 ruedas



Asimismo, se ofrece una representación gráfica de lo comentado anteriormente:

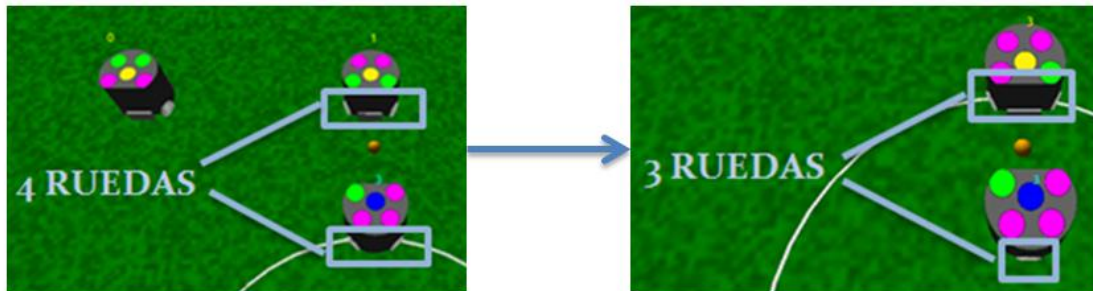


Ilustración 57: modificaciones en el simulador grSim [7] [15]

En la siguiente ilustración se muestra la estructura de las ruedas del robot físico de la ilustración 58.

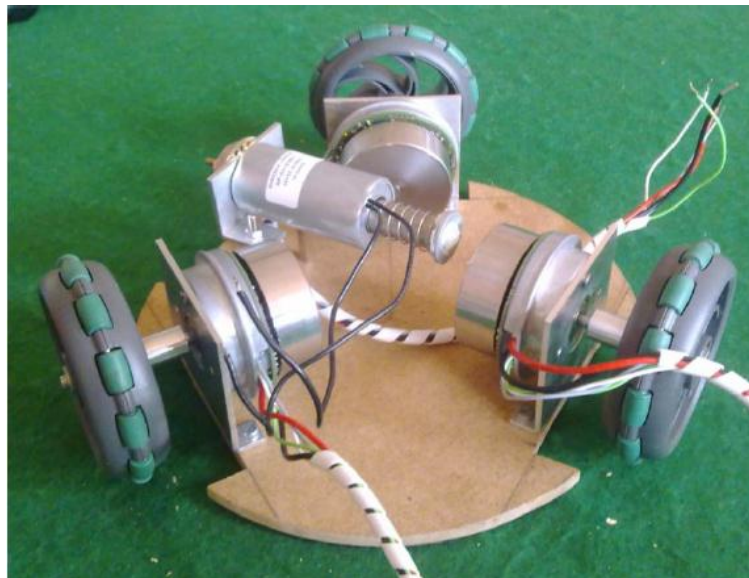


Ilustración 58: distribución de las ruedas

### 3.2.5. Sistema de Visión en grSim

El simulador grSim [7] [15] no solo emula el comportamiento de los robots y la situación en el terreno de juego de estos, sino que, además, plasma el funcionamiento real del sistema de visión de esta liga.

En este punto se va a detallar como envía el simulador la información para que los clientes reciban la información del estado del partido.

Este sistema codifica los paquetes con información utilizando Google Protocol Buffers [13], este sistema permite codificar estructuras de datos de una forma eficiente. En la ilustración 59 se muestra el funcionamiento de Google Protocol Buffers [13].

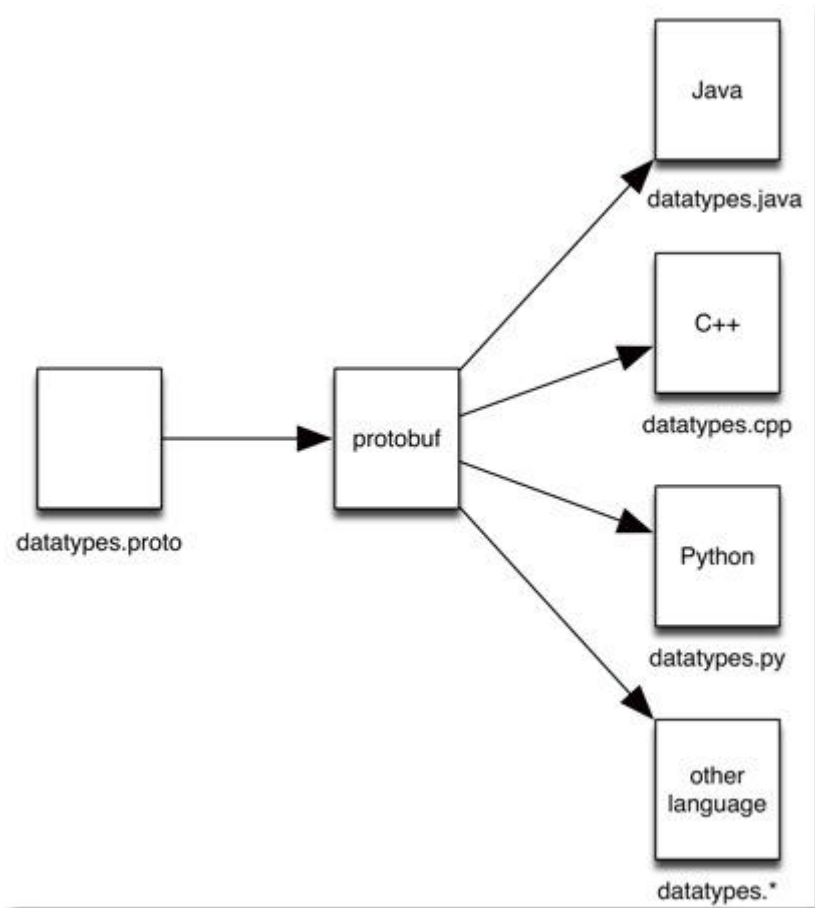


Ilustración 59: Google Protocol Buffers [12]

El simulador envía una estructura de datos codificada con Google Protocol Buffers [13] y una vez recibidos los datos se transforma en código para permitir tratar la información obtenida y poder almacenarla.

### 3.3. Clientes Java de Envío y Recepción de Paquetes del Simulador grSim

Para realizar las conexiones necesarios con el simulador se han creado dos pequeños clientes en Java [16]. Uno para enviar comandos al simulador y este sea capaz de interpretarlos, y otro que reciba los paquetes que envía el simulador con la información del estado del juego.

Estos clientes han sido utilizados para crear las comunicaciones en la arquitectura.

### 3.3.1. Cliente Java de Envío de Comandos

Para enviar comandos al simulador y este sea capaz de interpretarlos, se ha creado una pequeña aplicación Java [16] con una interfaz simple e intuitiva que permite el envío de paquetes UDP con los datos de los comandos que se envían al simulador.

A continuación, se muestra una captura de la interfaz creada para el cliente Java [16]:

The screenshot shows a web-based form for sending commands to a simulator. The form is organized into two columns. The left column contains: Simulator Address (127.0.0.1), Id (0), Velocity X (m/s) (0), Velocity Y (m/s) (0), Velocity Z (m/s) (0), a checkbox for 'Send Velocity? (or wheels)', Chip (m/s) (0), and a checkbox for 'Spin'. The right column contains: Simulator Port (20011), a dropdown menu set to 'Yellow', Wheel1 (m/s) (0), Wheel2 (m/s) (0), Wheel3 (m/s) (0), Wheel4 (m/s) (0), and Kick (m/s) (0). At the bottom of the form are two buttons: 'Send' and 'Reset'.

Ilustración 60: interfaz cliente de envío Java [16]

Como puede observarse en la imagen, se trata de una interfaz muy sencilla e intuitiva en la que se rellenan los parámetros con la información que se desea enviar a los robots.

Los parámetros que se introducen son:

- Dirección IP a la que se van a enviar los paquetes de datos.
- Puerto al que se envían los paquetes de datos.
- El identificador del robot al que se envía la información.

- El equipo al que pertenece el robot al que se le envía la información.
- Velocidad en el eje X medida en metros por segundo.
- Velocidad en el eje Y medida en metros por segundo.
- Velocidad en el eje Z medida en metros por segundo.
- Velocidad de las ruedas 1, 2, 3 y 4 medidas en metros por segundo.
- Posibilidad de enviar la velocidad directamente o a las ruedas de forma independiente.
- Disparo de la pelota medido en metros por segundo.
- Disparo elevado de la pelota en metros por segundo.
- Activación del *spinner* para mantener el control de la pelota.
- Botón de envío del paquete.
- Botón de *reset* que pone todos los parámetros a cero.

Los datos numéricos de la velocidad deben introducirse como valores decimales en coma flotante y el identificador del robot como un entero sin decimales.

Si se introduce la velocidad con valor negativo, la rueda girará en sentido contrario. Al enviar varios valores de velocidades en distintos ejes, estas se combinan para obtener el movimiento final.

Para enviar los comandos, la información se codifica haciendo uso de Google Protocol Buffers [13].

Dado que se están utilizando robots con tres ruedas en lugar de cuatro como estaba implementado el simulador, la cuarta rueda queda sin efecto.

### **3.3.2. Cliente Java de Recepción de Paquetes del Simulador**

Para recibir paquetes del simulador grSim [7] [15] y del sistema de visión SSL-Vision [12] se ha creado un pequeño cliente que recibe paquetes codificados con Google Protocol Buffers [13].

Este cliente recibe un paquete enviado por el simulador o por el sistema de visión y lo muestra por pantalla. Para ello se conecta a la dirección multicast del simulador.

El paquete recibido contiene información de todos los robots de los dos equipos y de la pelota.

Este cliente se ha utilizado posteriormente para realizar la recepción de datos por parte de la arquitectura 3T [36] y de esta forma que la arquitectura procese los datos y actualice las posiciones tanto de los robots como de la pelota.

### **3.4. Adaptación de Arquitectura 3T para Desarrollar las Habilidades**

Para llevar a cabo la implementación de las habilidades básicas, se ha hecho uso de una arquitectura 3T [36] implementada en Java [16]. Esta arquitectura fue desarrollada por José Luis Cebrián, en su PFC “Arquitectura de control híbrida para robots autónomos” [20]. El objetivo de este PFC fue el de crear una arquitectura haciendo uso del paradigma híbrido, que une los paradigmas deliberativo y reactivo, para asentar las bases de un marco de trabajo que permita dotar de comportamiento inteligente a los robots.

A esta arquitectura desarrollada se le han incluido las clases necesarias para el desarrollo del proyecto. Para ello, ha sido necesario conocer el funcionamiento de la arquitectura para poder hacer uso de esta de forma correcta.

#### **3.4.1. Diseño Arquitectónico**

Esta arquitectura se encuentra modularizada para facilitar su desarrollo y la inclusión de nuevas funcionalidades. En la siguiente ilustración 61 se muestra la arquitectura.

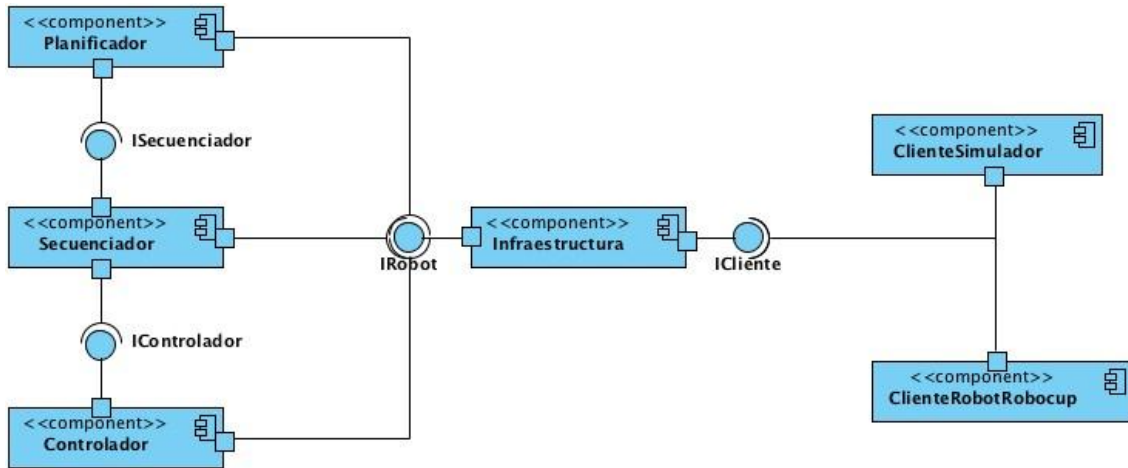


Ilustración 61: arquitectura 3T [36]

El subsistema Infraestructura es el que hace uso de la arquitectura para cualquier robot con características similares al Pioneer 3-DX [37]. En este caso ha sido necesario adaptarlo para los robots de la SSL [5], por lo tanto, se ha incluido una interfaz *Robot* y una clase *RobotRoboCup* para adaptar la arquitectura a las características de estos robots.

El siguiente componente es el controlador, el cual se muestra en la siguiente ilustración 62.

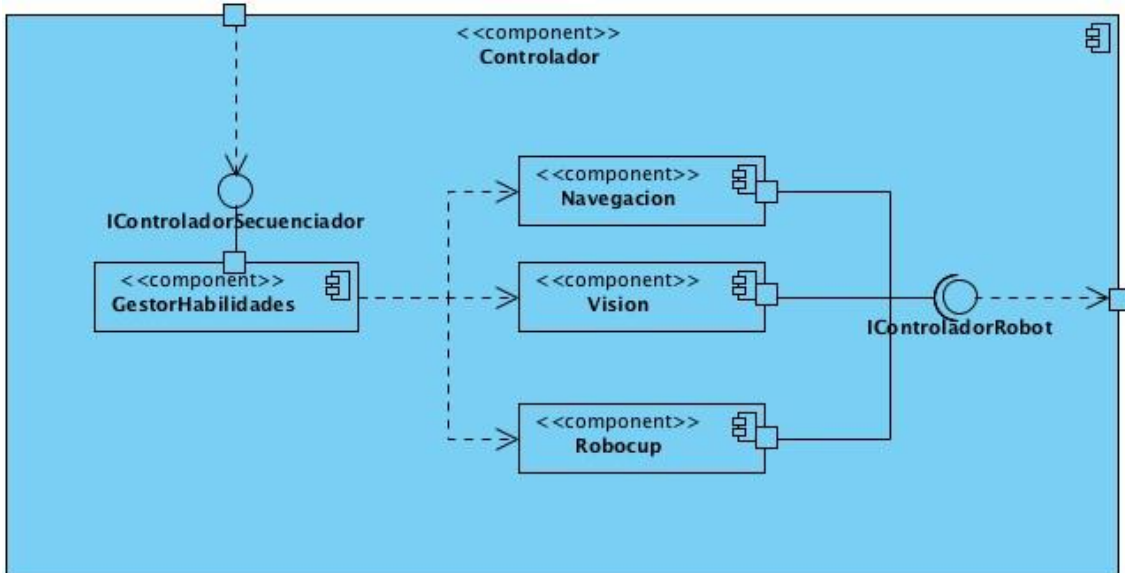


Ilustración 62: controlador

Este componente está compuesto por 4 subcomponentes. El primero de ellos es el gestor de habilidades, el cual se encarga de enviar al secuenciador acciones necesarias para utilizar las habilidades del robot, activándolas o desactivándolas según vaya siendo necesaria su ejecución. Además este subcomponente se encarga de controlar el hilo de ejecución del controlador, el cual gestiona sus comunicaciones y monitoriza la ejecución.

Los otros tres subcomponentes son los que contienen las distintas habilidades implementadas y modularizadas en subcomponentes. Por un lado, se encuentran los subcomponentes de Navegación y Visión que ya estaban desarrollados en el PFC de José Luis Cebrián [20] y el subcomponente de *RoboCup* [5] el cual ha sido desarrollado tanto en este PFC como el realizado por Verónica Raspeño Gutiérrez [21]. En este subcomponente *RoboCup* [5] se han incluido todas las habilidades desarrolladas para los robots de la SSL [5] tanto físicos como para el simulador grSim [7] [15]. Sería posible añadir nuevos módulos de habilidades en el futuro que agrupen habilidades con características en común.

El siguiente componente que se va a detallar es el secuenciador, el cual se muestra en el siguiente diagrama:

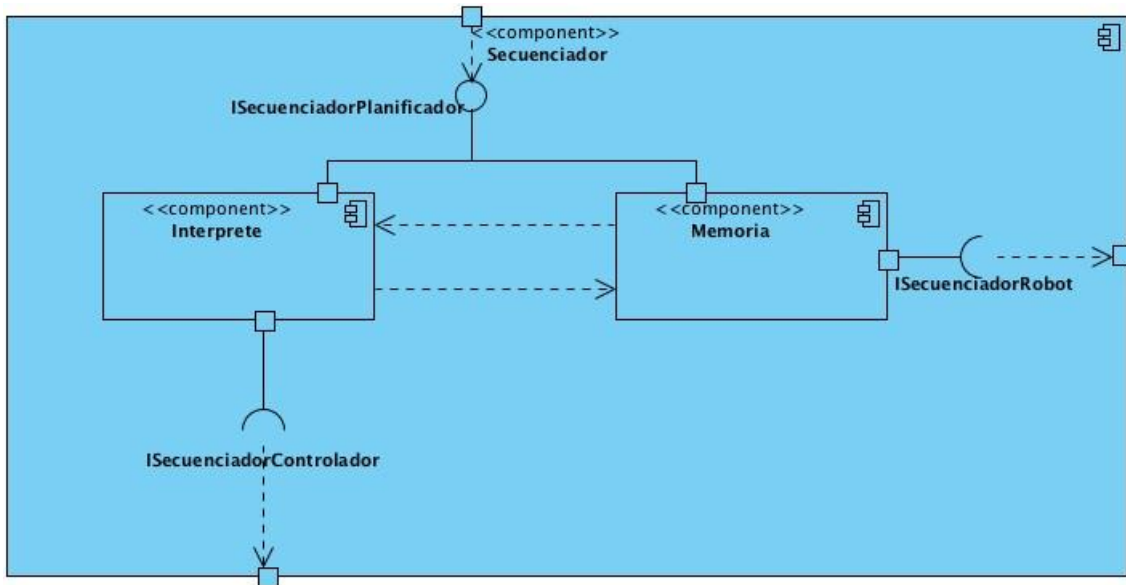


Ilustración 63: secuenciador

Este componente está compuesto por el subcomponente intérprete y el subcomponente *memoria*. El intérprete recibe los planes del secuenciador y los interpreta convirtiendo las tareas y acciones en habilidades que ejecutará el controlador. Además, activa y desactiva conjuntos de habilidades que siguiendo la secuenciación permite al robot cumplir con la meta definida.

Por otro lado, la memoria almacena planes utilizados anteriormente y el resultado. Monitorizando esta información, es posible realizar ciertas modificaciones en el plan y así ayudar al secuenciador en la secuenciación de tareas.

El último componente que forma la arquitectura del sistema es el planificador, el cual se muestra en la ilustración 64.



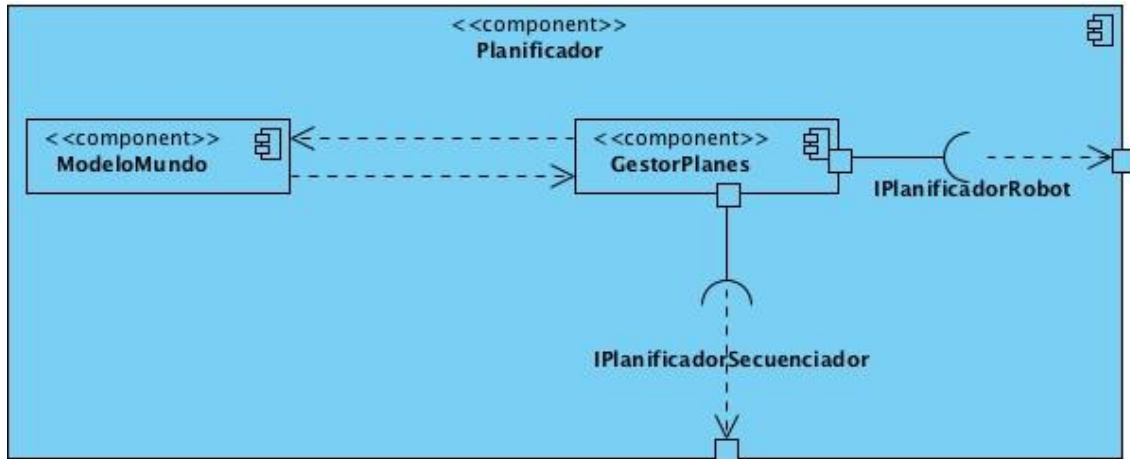


Ilustración 64: planificador

Este componente no se ha tenido en cuenta para el desarrollo del sistema, dado que no ha sido necesario realizar planes, y por lo tanto, se explicará muy brevemente su funcionamiento.

Está formado por dos subcomponentes que funcionan de forma conjunta para crear planes que utilizará el robot.

Por un lado, se encuentra el gestor de planes el cual incluye un planificador genérico que utiliza un algoritmo de búsqueda para la creación de planes a partir de las metas definidas.

El otro subcomponente es el modelo del mundo, el cual se utiliza para representar el entorno en el que se sitúa el robot.

### 3.4.2. Diseño Detallado del Sistema

A partir del diseño arquitectónico se detalla el diseño del sistema. Para ello se muestran diagramas de clases, en los que se muestran las clases más relevantes y donde se puede ver como se comunican las distintas clases del sistema.

En primer lugar se detalla el componente *Infraestructura*, el cual se muestra en la ilustración 65.

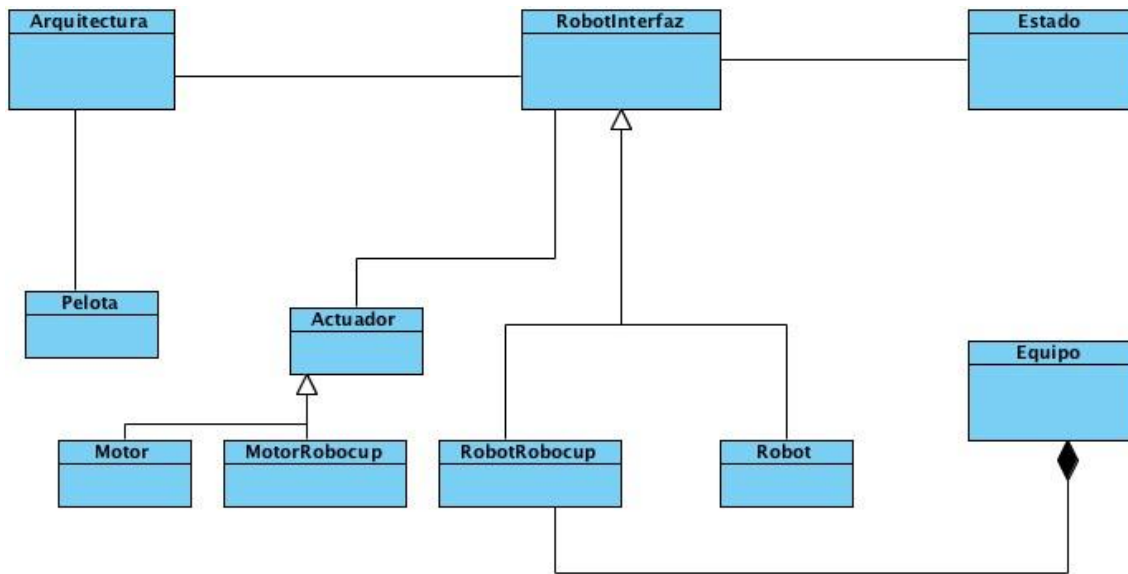


Ilustración 65: diagrama de clases de Infraestructura

Como puede observarse en el diagrama, se ha creado una interfaz *RobotInterfaz* de la que heredan la clase *Robot* y la clase *RobotRoboCup*. La clase *Robot* era la que se creó en el proyecto anterior [20] y se ha creado la clase *RobotRoboCup* en la cual se incluyen características propias de los robots que forman los equipos de la RoboCup como si este robot es portero, el equipo al que pertenecen, la posesión de la pelota o el identificador dentro del propio equipo.

Las clases que implementan *RobotInterfaz* permiten acceder al resto de la arquitectura, a su estado y al resto de componentes. La clase *Arquitectura* es la que se encarga de ejecutar todos los hilos correspondientes a las capas del sistema de controlador, secuenciador y planificador.

El siguiente componente es el *Controlador*, el cual se muestra en la ilustración 66.

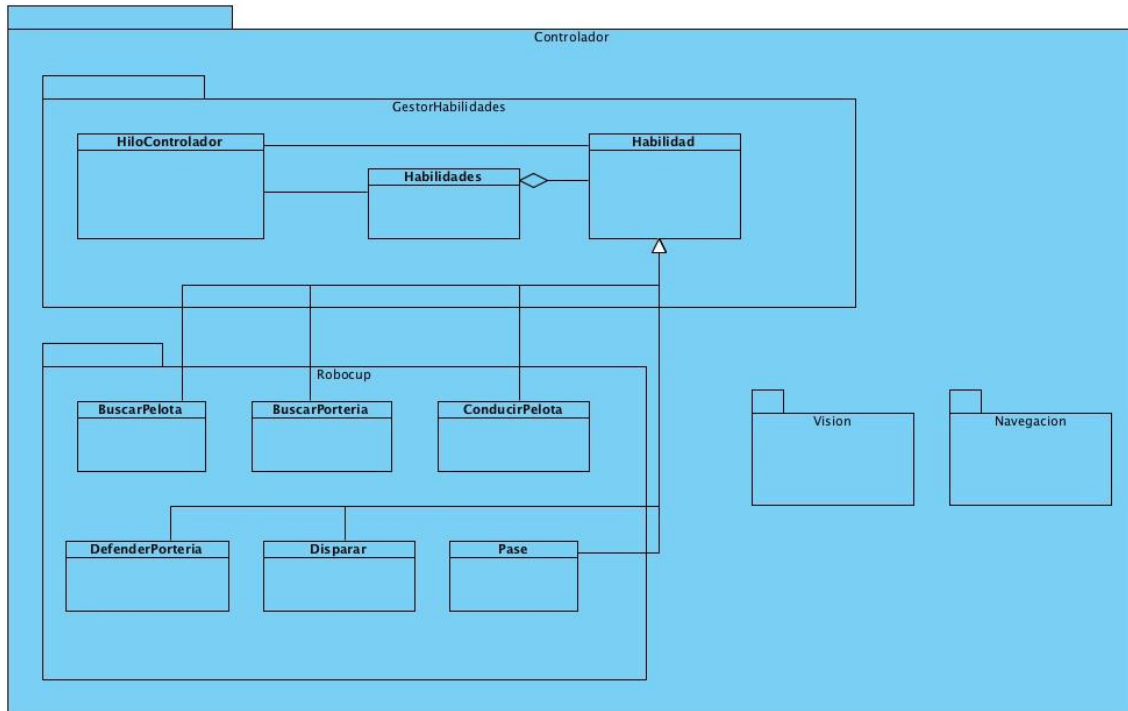


Ilustración 66: diagrama de clases de Controlador

Como puede apreciarse en el diagrama del componente *Controlador*, este se divide a su vez en distintos subcomponentes, los cuales son *GestorHabilidades*, *RoboCup*, *Vision* y *Navegacion*. Se corresponde con la capa reactiva de la arquitectura.

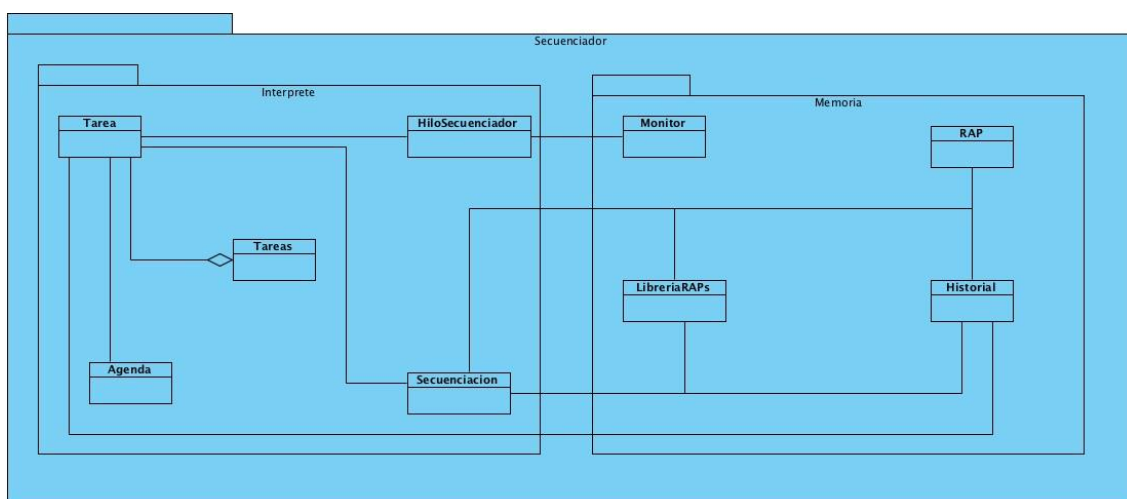
Los subcomponentes *Vision* y *Navegacion* contienen habilidades implementadas anteriormente en el PFC de José Luis [20] y por lo tanto, no se prestará atención a ellos.

El subcomponente *GestorHabilidades* contiene la clase *HiloControlador* que es el hilo de ejecución de esta capa, además, esta clase proporciona todos los métodos necesarios para que el *Secuenciador* haga uso de ellos.

De la clase *Habilidad* heredan todas las clases del resto de componentes, aunque en este caso solo se presta atención a las habilidades del subcomponente *RoboCup*. A su vez, la clase *Habilidades* es un agregado de objetos de tipo *Habilidad*. Estas clases son utilizadas por *HiloControlador* para poder hacer uso de las habilidades de las que el robot dispone.

Dado que este proyecto se ha basado en el desarrollo de habilidades para los robots utilizados en la SSL [5], el componente *RoboCup* es el que tiene mayor peso en este proyecto. En este componente se han implementado todas las habilidades que realizará el robot. Dado que estas habilidades heredan de la clase *Habilidad*, todas implementan los métodos *ejecutar()* y *comprobarExito()* en los cuales se define el comportamiento que deben cumplir los robots y las condiciones de éxito para que estas habilidades se den por realizadas respectivamente.

El siguiente componente es el *Secuenciador*, la capa intermedia de la arquitectura. Este componente se detalla en el siguiente diagrama:



**Ilustración 67: diagrama de clases de Secuenciador**

Este componente trabaja a nivel reactivo y deliberativo y se divide en los subcomponentes *Intérprete* y *Memoria*.

El subcomponente *Intérprete* se encuentra la clase *HiloSecuenciador*, cuyo funcionamiento es muy similar al de la clase *HiloControlador*, es decir, contiene el hilo de ejecución de esta capa. Esta clase hace uso de la clase *Agenda* que es una agenda de tareas para almacenar los planes de la capa superior como elementos más utilizables que son objetos de la clase *Tarea*.

La clase *Secuenciación* se encarga de realizar las operaciones necesarias para secuenciar las tareas y asignarles conjuntos de habilidades.

Por otro lado, la clase *Monitor* monitoriza la ejecución del sistema y el estado del robot para informar de los cambios a la clase *HiloSecuenciador*, y con ello volver a planificar la agenda si esto fuera necesario.

El subcomponente *Memoria* también contiene una clase *Historial* que almacena los planes pasados y la clase *LibreriaRAPs* que contiene los RAP definidos para la arquitectura y las tareas que pueden llevar a cabo los robots. En esta clase se incluyeron los RAP utilizados para ejecutar las habilidades de los robots de la SSL [5].

La última capa de la arquitectura es el componente *Planificador*, el cual queda reflejado en el siguiente diagrama:

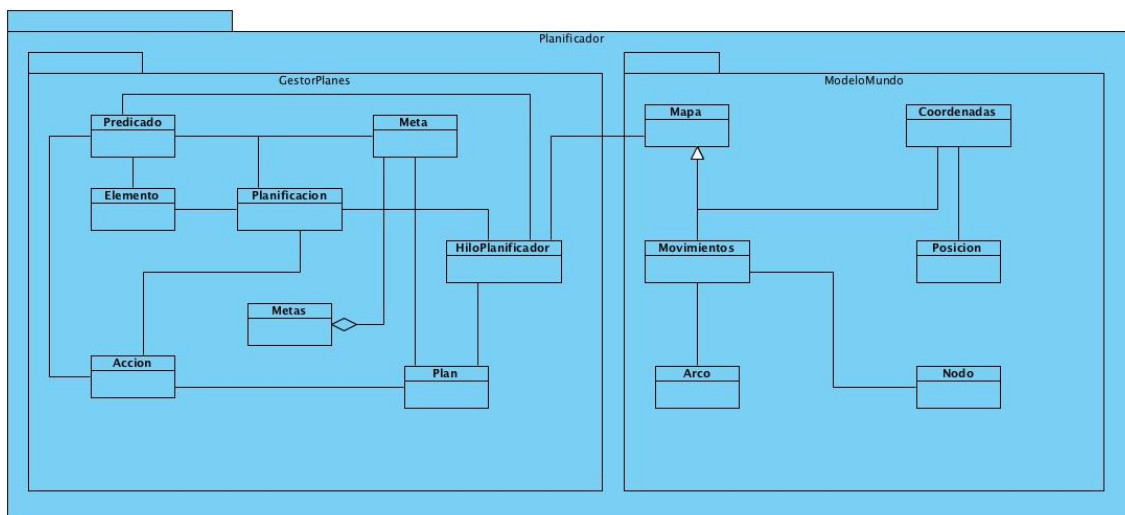


Ilustración 68: diagrama de clases de Planificador

Este componente se divide en dos subsistemas el *GestorPlanes* y el *ModeloMundo*. El modelo del mundo contiene los elementos que representan el entorno del robot. Este subcomponente no es utilizado en este proyecto, dado que el entorno es el terreno de juego y no es necesario incluir ningún tipo de información adicional aparte de la que recibe el sistema.

Respecto al subcomponente *GestorPlanes* contiene una clase llamada *HiloPlanificador*, que representa el hilo de ejecución de esta capa de la arquitectura. El planificador está compuesto por las clases *Planificación*, *Predicado*, *Acción* y *Elemento*. La clase *Planificación* tiene una función similar a la clase *Secuenciación* del componente *Secuenciador*, se encarga de elaborar un plan a partir de unas metas definidas y unas acciones definidas en la clase *Plan*. Para ello hace uso de predicados, acciones y los elementos que los componen. Las acciones pueden asociarse a las tareas por medio de las submetas, lo que permite escoger RAP y habilidades para ejecutarlas.

El planificador no es utilizado en el proyecto como tal, simplemente se definen unas metas y unas acciones para que se ejecuten secuencias de habilidades y de esta forma los robots puedan ejecutar estas acciones.

### 3.4.3. Diagrama de Ejecución

En el siguiente diagrama se muestra la ejecución de un plan creado:

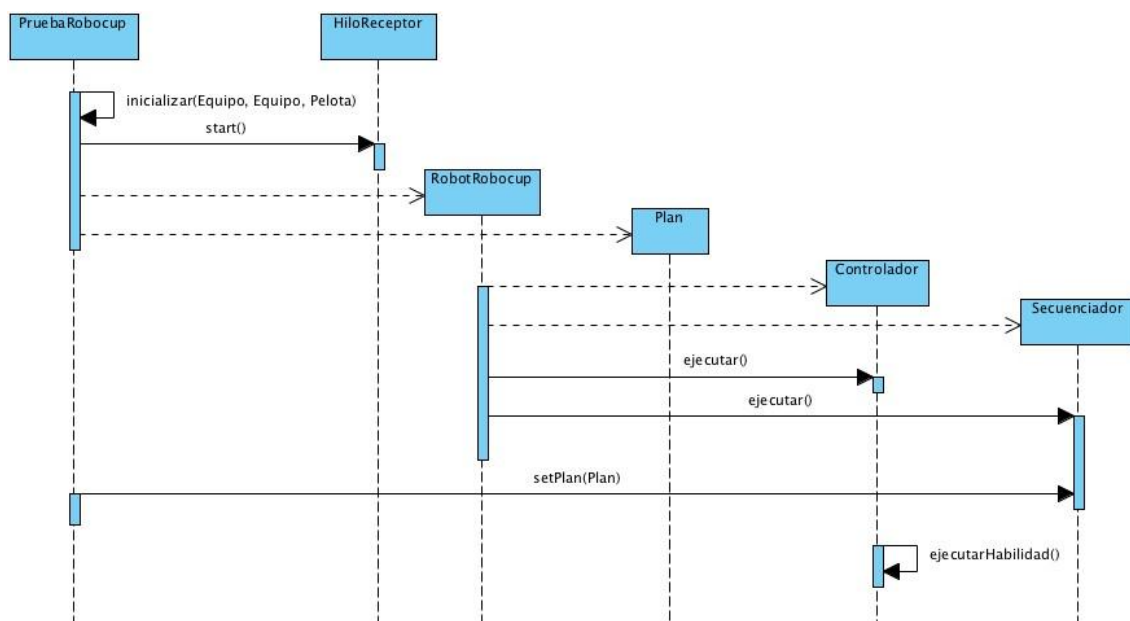


Ilustración 69: diagrama de ejecución

Como puede observarse en la imagen superior la clase *PruebaRoboCup* funciona como clase principal y se encarga de realizar las llamadas necesarias para ejecutar un plan

bien en el simulador o en el robot físico [30] [31] [32]. Esta clase en primer lugar inicializa las posiciones de los robots de los dos equipos que participan en el juego y la posición de la pelota.

Posteriormente esta clase inicializa el hilo receptor que se encarga de ejecutar el hilo de recepción de información de la clase *HiloReceptor*, el cual actualizará continuamente las posiciones de los robots y de la pelota y de esta forma se consigue precisión en la ejecución de las habilidades.

Una vez se crea un objeto de tipo *RobotRoboCup*, se inicializan los hilos *Controlador* y *Secuenciador* y se asigna al secuenciador el plan creado para llevar a cabo su ejecución.

El controlador recibirá las habilidades que se ejecutarán según el plan establecido y las habilidades se encargarán de enviar los comandos pertinentes a los robots físicos o a los robots del simulador.

### 3.5. Habilidades

El eje principal de este PFC representa el conjunto de habilidades que se han definido, porque logran realizar una implementación para la competición RoboCup [4], en concreto para la liga SSL [5].

Se han desarrollado distintas habilidades básicas que permiten representar movimientos propios del fútbol, como puede ser el pase o la defensa de una portería, todas estas habilidades se desarrollarán a lo largo de este punto y los siguientes.

No solo eso, para completar todas las habilidades y tener un mayor número, este proyecto se complementa con otro realizado al mismo tiempo, realizado por Verónica Raspeño Gutiérrez [21]. Por eso mismo, las habilidades han sido desarrolladas por dos proyectos y cada una de ellas está dotada de un porcentaje de realización por los autores, como se detalla en la tabla 4.




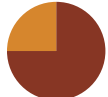





Rol	Habilidad	% PFC Formación de defensa	% PFC Formación de ataque	Gráfico
<b>Todos</b>	Giro óptimo	50%	50%	
<b>Portero</b>	Buscar portería	25%	75%	
	Defender portería	75%	25%	
<b>Jugador defensor</b>	Defender portería	75%	25%	
<b>Jugador atacante</b>	Buscar pelota	40%	60%	
	Conducir pelota	40%	60%	
	Pase	60%	40%	
	Esquivar	60%	40%	
	Disparar	25%	75%	

Tabla 4: habilidades elaboradas para la liga SSL [5]



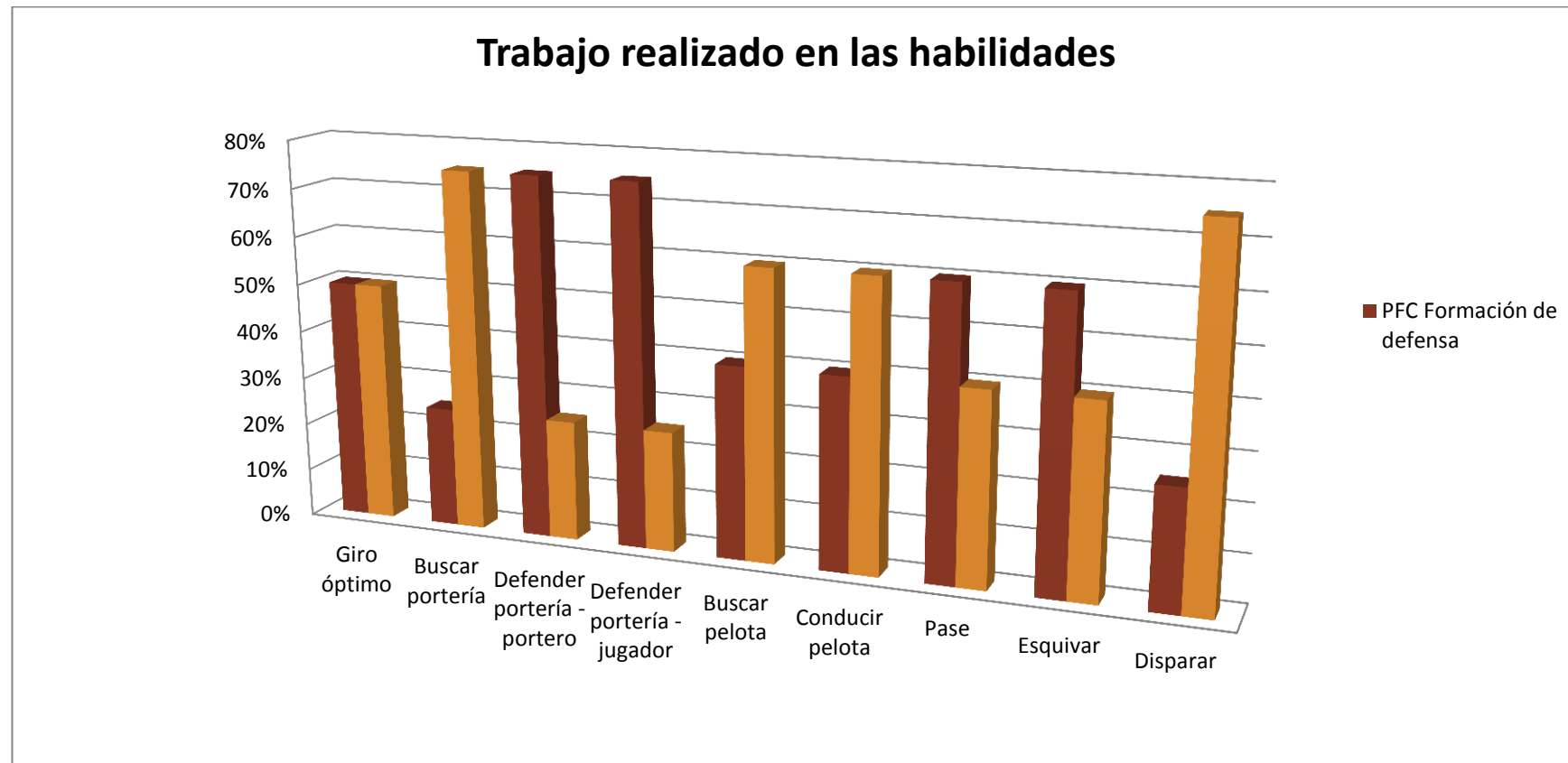


Ilustración 70: gráfica de habilidades elaboradas para la liga SSL [5]

### 3.5.1. Habilidad: Giro Óptimo

Esta característica se ha otorgado a todas las habilidades creadas que incluyan giros, tanto giro sobre el robot mismo como giros alrededor de un objeto.

El giro óptimo significa girar hacia la derecha o hacia la izquierda según el camino más corto. La forma de calcularlo es la siguiente: el robot previamente tiene información sobre a qué punto se debe dirigir, por ejemplo, si va hacia la portería o hacia las coordenadas de la pelota que le ha enviado el sistema de visión, calcula internamente su orientación y la orientación del lugar hacia donde se le ha indicado ir respecto a él.

Para calcular la orientación del robot simplemente hay que recurrir al paquete que envía el simulador grSim [7] [15] o, en su defecto, el SSL-Vision [12] en los robots físicos.

Para calcular la orientación del objetivo del robot (orientado hacia la portería, coordenadas concretas o pelota) hay que realizar las operaciones trigonométricas propias de calcular la orientación dados dos puntos, tomando como referencia el punto del objeto. Los dos puntos que se cogen son únicamente el de las x y el de las y, correspondientes a la segunda dimensión de tal manera:

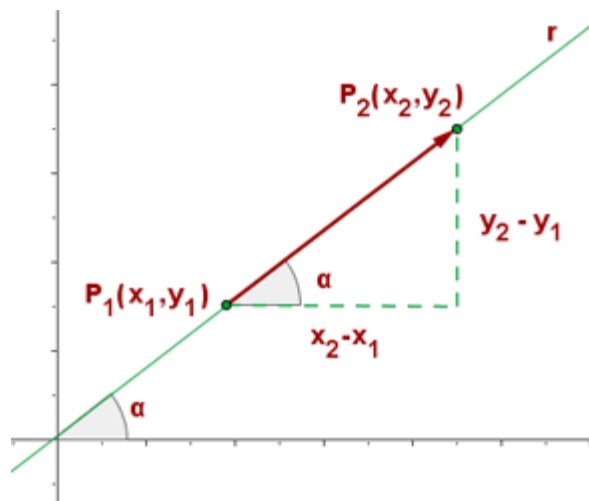


Ilustración 71: cálculo dirección entre dos puntos

Una vez calculada la orientación, se obtiene la forma óptima para llegar a ese punto, girando para la derecha o para la izquierda. Lo difícil en este aspecto es controlar las coordenadas y orientación, ya que pueden ser positivas o negativas y eso interfiere en el resultado final.

Dentro del giro óptimo, existen dos variantes: giro del robot sobre sí mismo, giro del robot en torno a otro objeto.

El giro del robot sobre sí mismo se realiza en la mayoría de las ocasiones, siempre que el robot quiere ir a una posición y su orientación no es la correcta. Por ejemplo, en la imagen 72, si el robot quisiera ir hacia la pelota, realizaría un giro como representa la flecha azul:



**Ilustración 72: giro óptimo del robot sobre sí mismo**

Por otro lado, existe el giro sobre un objeto, que siempre es la pelota. Esto se da cuando el robot va hacia la pelota y luego quiere situarse orientado a, por ejemplo, la portería y mantener la pelota. La forma más eficiente de hacerlo sin riesgo a perder la pelota es que el robot gire siendo el centro del giro la pelota, orientándose a las coordenadas objetivo, hasta situarse. Un ejemplo de giro es el representado en la ilustración 73.



Ilustración 73: ejemplo de habilidad giro óptimo

### 3.5.2. Habilidad: Defender Portería (Portero)

Esta habilidad es ejecutada por el robot del equipo que cumple con el rol de portero dentro del equipo y tiene como principal objetivo la de evitar que el equipo contrario meta un gol en la portería del equipo defensor.

Esta habilidad se complementa con la habilidad buscar portería, la cual está explicada con gran detalle en el PFC de Verónica Raspeño Gutiérrez [21]. Esto se debe a que gracias a la habilidad buscar portería, el robot que cumple el rol de portero localiza la portería a defender y se sitúa en la línea de gol para evitar que el balón se introduzca en la portería.

Para defender la portería, el robot situado en la línea de gol cambia de posición en función del movimiento de la pelota. Para ello se mueve sobre la línea de gol consiguiendo así cubrir cualquier posición de disparo por parte del equipo contrario.

Para que el robot no pierda la posición, es necesario que este realice pequeñas correcciones para mantener la posición y de esta forma no desviarse de la línea de gol. Esto se debe a la descompensación de velocidades provocada por el cambio de los robots del simulador [7] de 4 a 3 ruedas.

En la siguiente imagen se muestra los movimientos a realizar por el portero al realizar la habilidad:

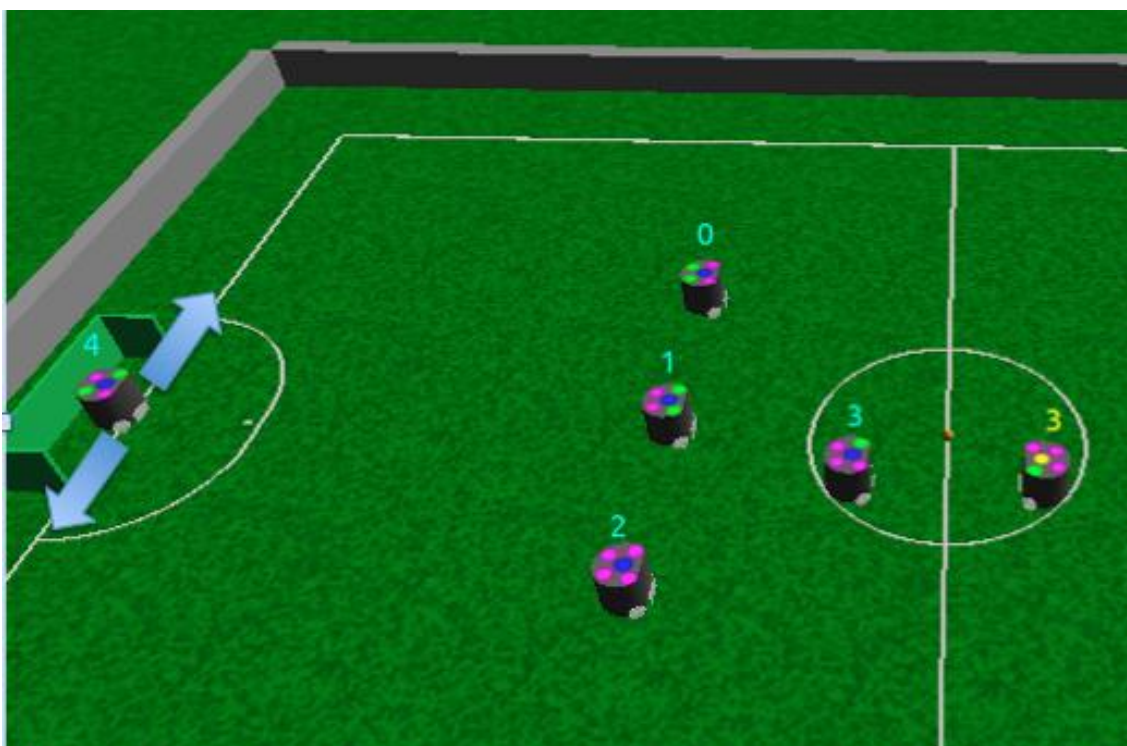


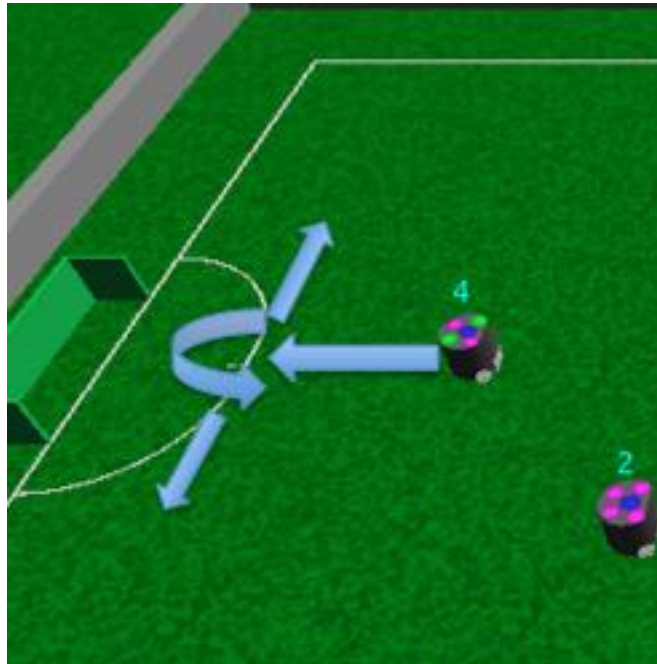
Ilustración 74: habilidad defender portería con el portero

### 3.5.3. Habilidad: Defender Portería (Jugador)

Esta habilidad como su propio nombre indica realiza la misma función que la habilidad de defender portería en portero, pero en este caso es realizada por un robot con el rol de jugador de campo y en lugar de mantener la posición en la línea de gol, mantiene la posición en el borde del área.

Para ello el robot busca la línea del área de la portería defendida y se sitúa de cara al balón para poder moverse en la frontal del área y de esta forma evitar que los robots del equipo contrario se internen en el área.

Los movimientos realizados por el robot quedan representados en la siguiente imagen:



**Ilustración 75: habilidad defender portería con jugador**

#### **3.5.4. Habilidad: Pase**

La habilidad de pase es una de las más complejas realizadas, dado que encadena una serie de movimientos y habilidades que deben realizarse de forma conjunta para conseguir realizar un pase preciso al jugador deseado.

Para llevar a cabo esta habilidad es necesario tener en cuenta distintos factores que pueden influir en la consecución de un pase correcto. Es importante destacar que a diferencia de las habilidades antes expuestas, en este caso entran en juego dos robots distintos, el que realiza el pase y el receptor.

El objetivo de esta habilidad es que el robot que va a realizar el pase le entregue la pelota de la forma más precisa posible al robot que va a recibir la pelota. Para ello, es



necesario que el robot que realiza el pase se sitúe frente a la pelota y se oriente hacia el robot que va a recibir el pase para posteriormente realizar un golpeo con la velocidad necesaria para que esta llegue al robot de destino.

Por lo tanto, en esta habilidad es necesario indicar el identificador del robot al que va dirigido el pase. El robot que va a realizar el pase, una vez que se ha situado frente a la pelota, deberá girar sobre la misma hasta alcanzar la orientación correcta hacia el robot de destino y una vez orientado, este deberá avanzar y golpear la pelota para que esta llegue al robot indicado.

En la siguiente imagen puede observarse una representación gráfica de los movimientos que realiza el robot para realizar el pase:

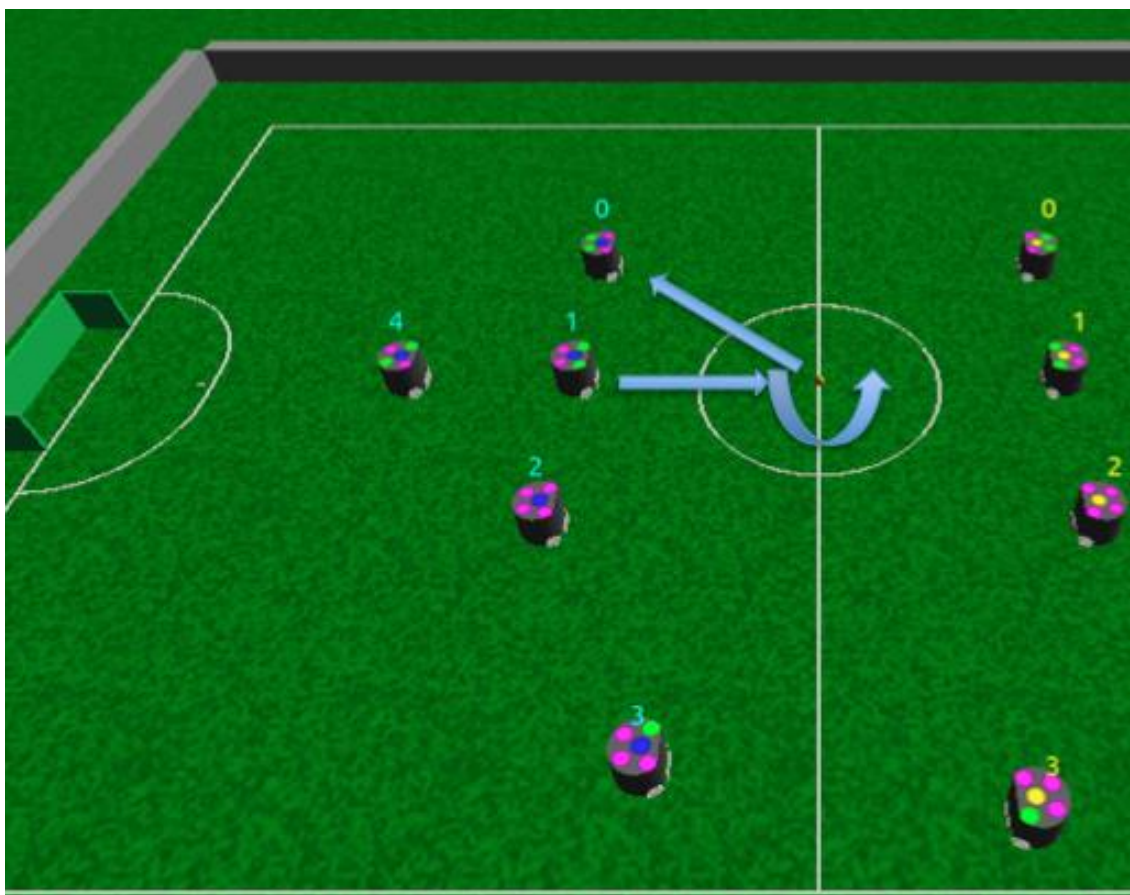


Ilustración 76: habilidad pase

### 3.5.5. Habilidad: Esquivar

La última habilidad que se presenta trata de esquivar otros robots que puedan obstaculizar el movimiento del robot y de esta forma conseguir que se realicen las habilidades independientemente de los distintos obstáculos que exista en el terreno de juego.

Esta habilidad no se ha desarrollado de forma similar al resto de habilidades desarrolladas, sino que se ha desarrollado en un hilo de ejecución que en caso de ejecutarlo se ejecutaría en paralelo del resto de habilidades. De esta forma, el robot podría esquivar obstáculos en caso de ejecutar este hilo de ejecución mientras el robot se encuentra ejecutando alguna habilidad, como por ejemplo, buscar pelota.

Por lo tanto, es posible activar o desactivar según se desee la opción de esquivar en algunas de las habilidades explicadas como puede ser buscar portería, buscar pelota o conducir pelota.

Debido a la complejidad que se supone la coordinación de los movimientos de esquivar, el control de la pelota y la búsqueda de objetivos, en algunos casos no se ha conseguido una gran precisión, siendo una posible línea de mejora para el futuro utilizar distintos algoritmos para mejorar estas situaciones y así conseguir una mayor competitividad.

La siguiente imagen muestra un pequeño ejemplo gráfico de los movimientos que realizaría el robot si se decide ejecutar este hilo de ejecución:



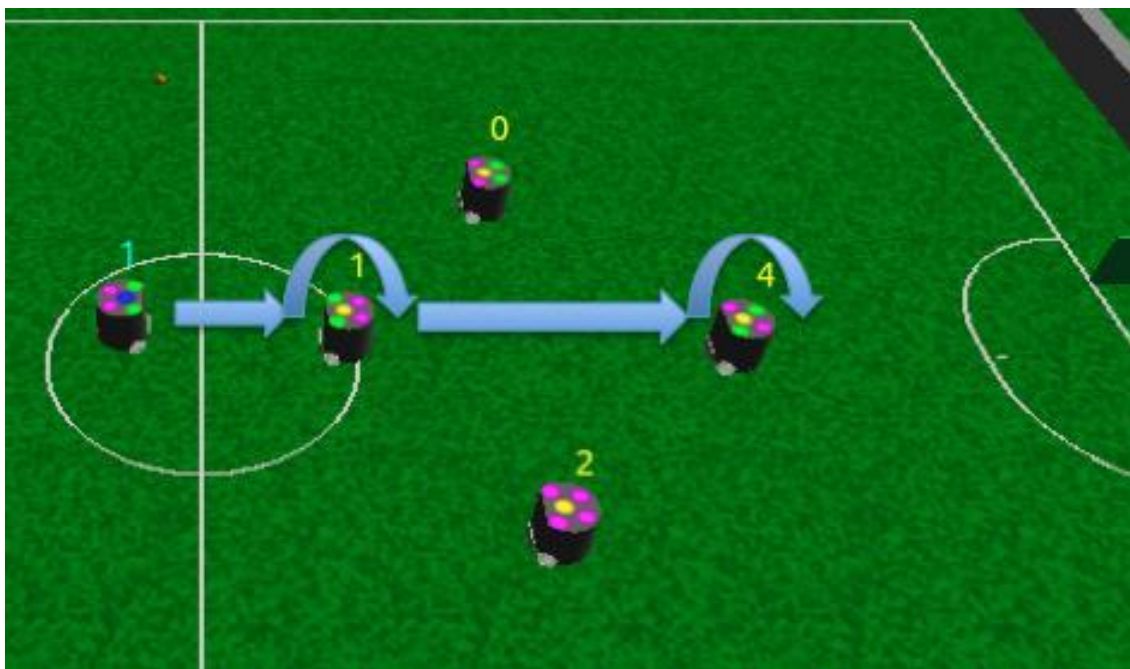


Ilustración 77: habilidad esquivar

### 3.6. Adaptación a Robots Físicos

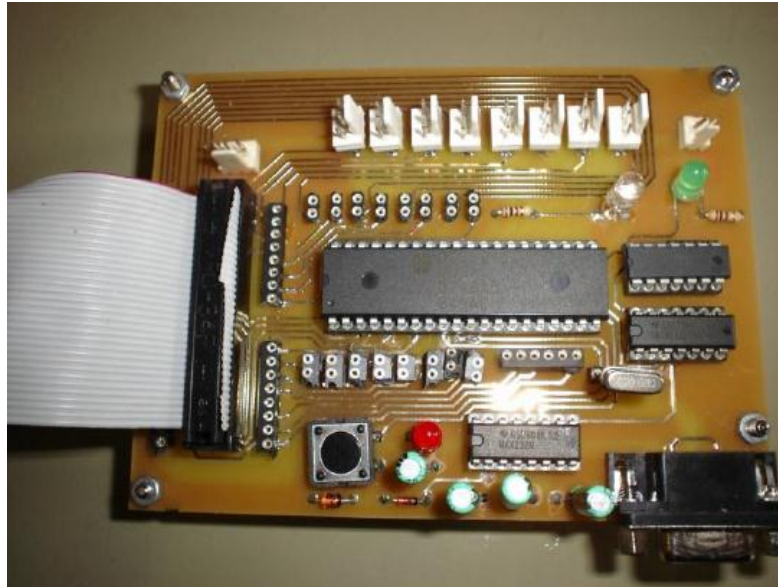
Dado que el verdadero fin del proyecto es la utilización de la arquitectura desarrollada en robots físicos y de esta forma en un futuro conseguir la formación de un equipo que cumpla con las normas de la Robocup SSL [5], se han realizado pequeñas pruebas utilizando el robot desarrollado por el departamento de automática de la Universidad Carlos III de Madrid.

A continuación se detallarán las características técnicas que posee el robot físico, así como los elementos de interés para la integración de la arquitectura a los robots reales.

#### 3.6.1. Características Técnicas

El robot físico ha sido desarrollado en dos proyectos [31] [32] en los cuales se detallan todas las características de las que dispone el robot.

En primer lugar, cabe destacar el sistema de procesamiento, el cual se encarga de procesar y enviar las órdenes para que el robot realice las acciones deseadas. Cabe destacar que este sistema funciona de forma inalámbrica.



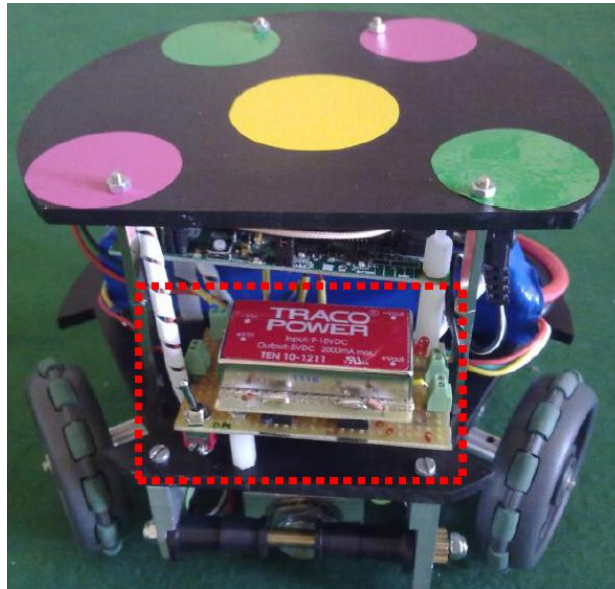
**Ilustración 78: sistema de procesamiento [31] [32]**

Para realizar las conexiones inalámbricas con el robot, el robot dispone de un módulo de comunicación Wi-Fi que es capaz de enviar y recibir información. El robot hace uso del controlador RCM4400-Rabbit.



**Ilustración 79: controlador Wi-Fi RCM4400-Rabbit [31] [32]**

Respecto a la estructura externa del robot, cumple con las reglas de la competición, las cuales quedan explicadas en el apartado “ANEXO VI: Reglamento SSL de 2011”. Como se muestra en la siguiente imagen, el robot respeta las dimensiones, los colores y la situación de la batería:



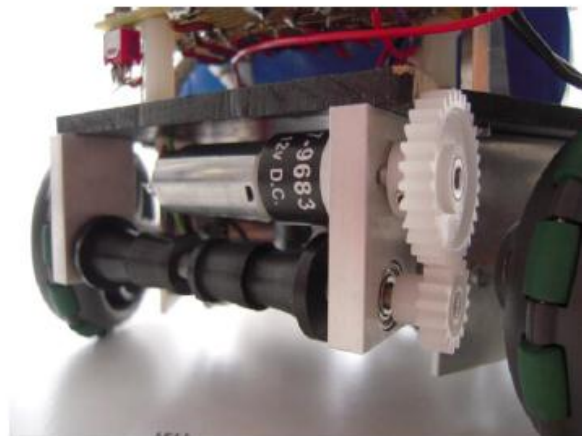
**Ilustración 80: situación de la batería en el robot [31] [32]**

Por decisión de diseño de los creadores, el robot dispone de tres ruedas en lugar de cuatro, de esta forma se reducen los costes y se consigue un buen resultado. En la ilustración 81 se muestra el tipo de rueda utilizado.



**Ilustración 81: rueda del robot [31] [32]**

Otro elemento importante a destacar es el dribbler, el cual consiste en una barra cilíndrica que permite mantener el control de la pelota y poder conducirla sin perderla.



**Ilustración 82: sistema de *dribbler* [31] [32]**

En la siguiente imagen se muestra el resultado final del *dribbler*:

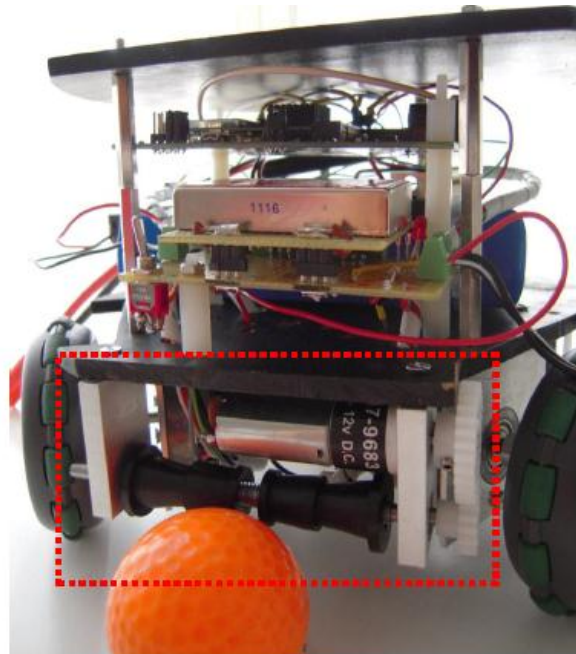


Ilustración 83: sistema dribbler en el robot [31] [32]

El resultado final se puede apreciar en la siguiente imagen:

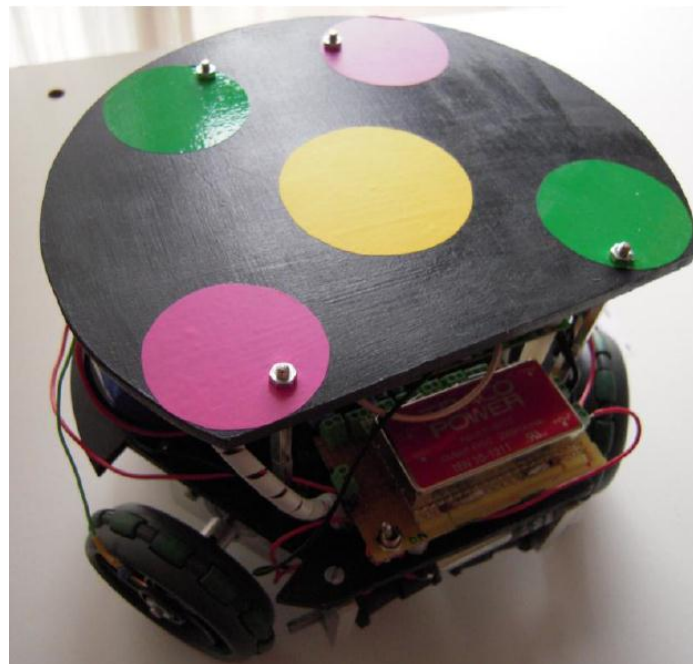


Ilustración 84: robot físico [31] [32]



### 3.6.2. Conexiones

Para llevar a cabo la comunicación de la arquitectura con el robot físico, ha sido desarrollado un firmware para el robot por Isaac Ruíz Agrazal, perteneciente al Área de Ciencias de la Computación e Inteligencia Artificial de la Universidad Carlos III de Madrid.

Este firmware es capaz de recibir los comandos enviados por la arquitectura y de interpretarlos y de esta forma que el robot realice las acciones correspondientes.

### 3.6.3. Sistema de Visión

Debido a que los robots de la SSL [5] no poseen sensores externos para obtener información del exterior y realizar un control reactivo, estos robots necesitan el sistema de visión, el cual recoge la información del terreno de juego y se la envía a todos los robots que están participando en el partido.

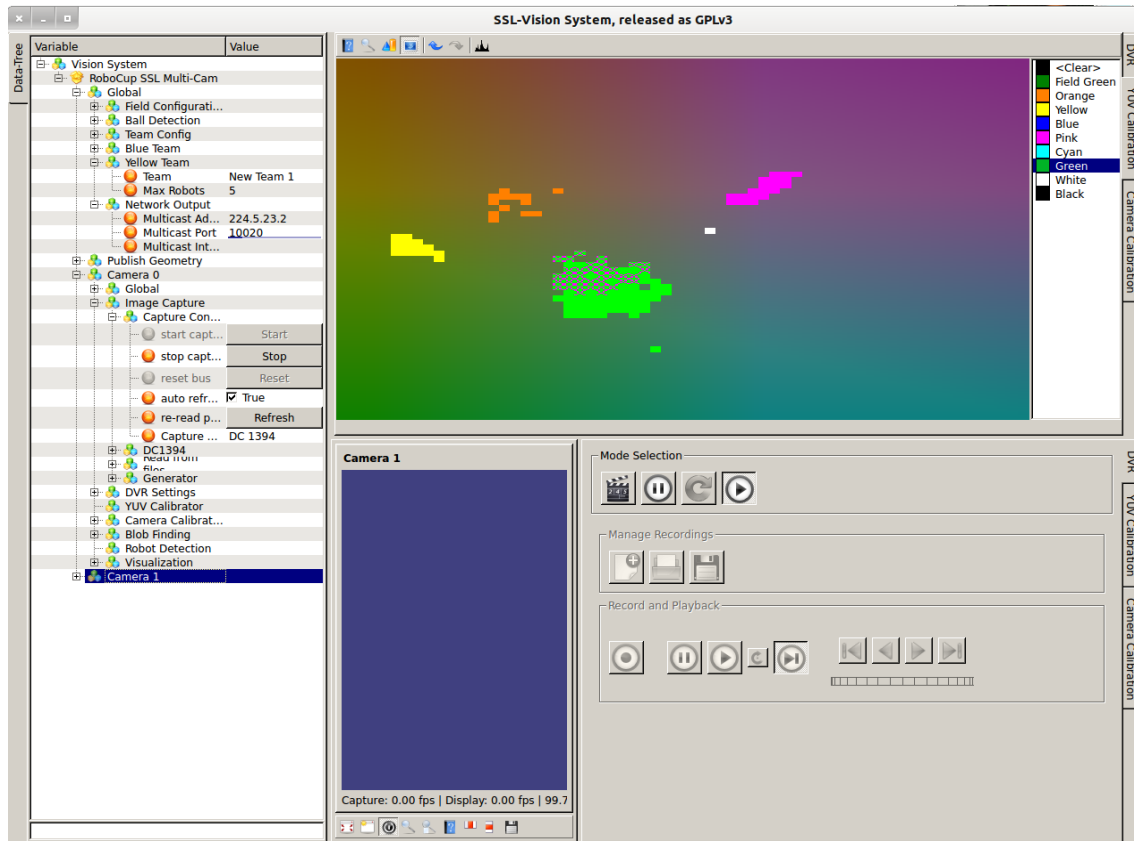
Al principio de esta competición las reglas permitían a cada equipo participante, tener su propio sistema de visión. Se pensó que los equipos gastaban muchos recursos en el desarrollo de estos sistemas y esto suponía un freno para la evolución de la competición. Esto unido a que los equipos creaban sistemas de visión muy similares, llevó a la creación de un sistema de visión común para todos los equipos llamado SSL-Vision [12].

En 2009, se instauró el sistema SSL-Vision [12] como sistema estándar que deberían utilizar todos los equipos participantes. Este sistema ha sido desarrollado por varios voluntarios de los equipos participantes y en la actualidad se encuentra en pleno desarrollo y mejora.

Como se explicó anteriormente en el apartado [3.2.5 Sistema de Visión en grSim](#), el simulador grSim [7] [10] envía los paquetes de información igual que lo hace el sistema SSL-Vision [12], aunque en este caso envía la información que recibe por la cámara y

procesa el sistema. Por lo tanto, estos paquetes enviados son codificados con Google Protocol Buffers [13].

La interfaz de la aplicación SSL-Vision [12] se muestra en la siguiente imagen:



**Ilustración 85: interfaz SSL-Vision [12]**

Además de la interfaz mostrada, a continuación se muestra una representación de cómo interpreta el sistema la información recibida por la cámara en la ilustración 86.

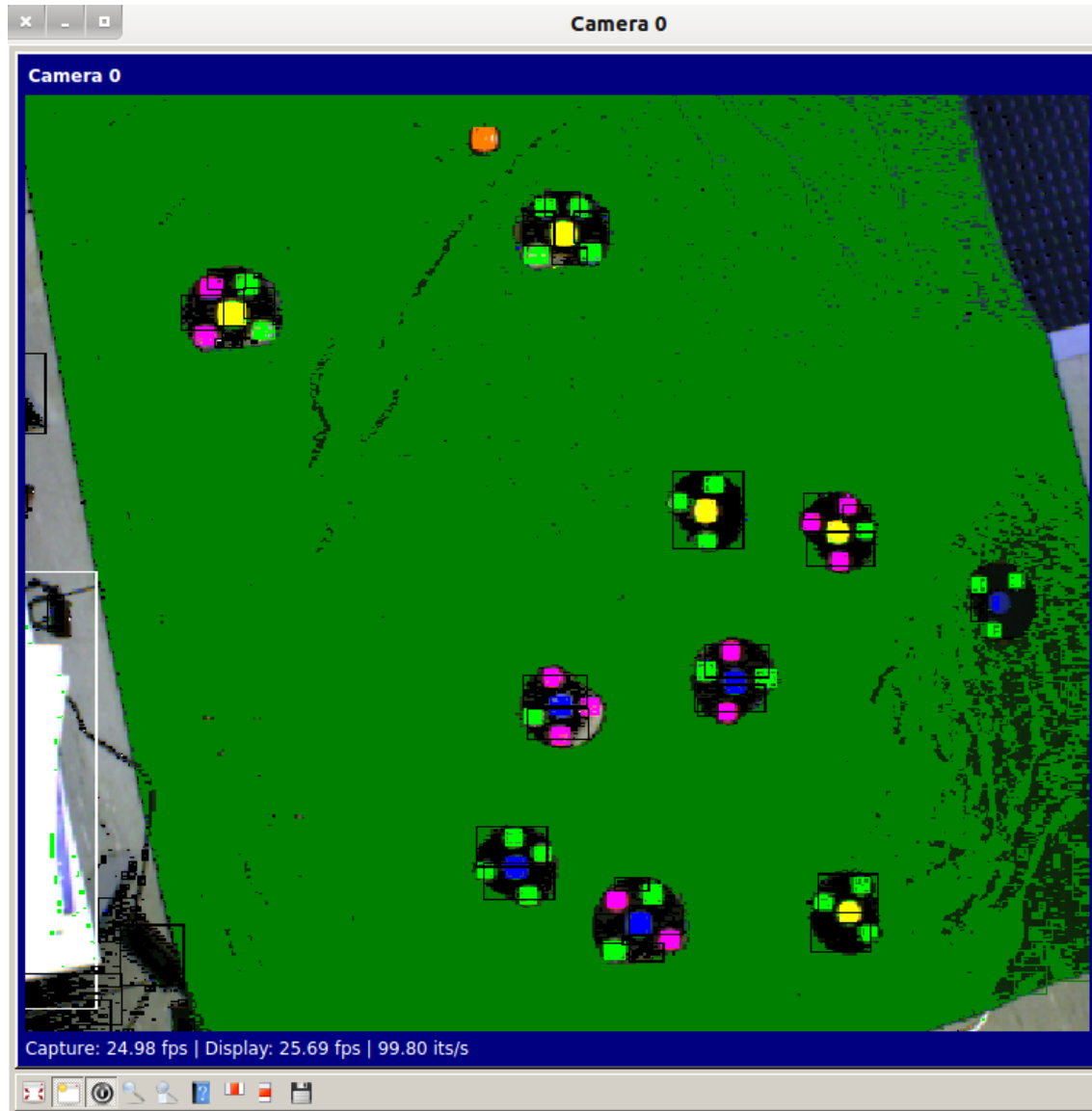


Ilustración 86: vista de la cámara con SSL-Vision [13]

Toda la información recogida por la cámara, es traducida por el sistema a unas posiciones que toman los robots, las cuales quedan representadas en la pantalla de la ilustración 87.



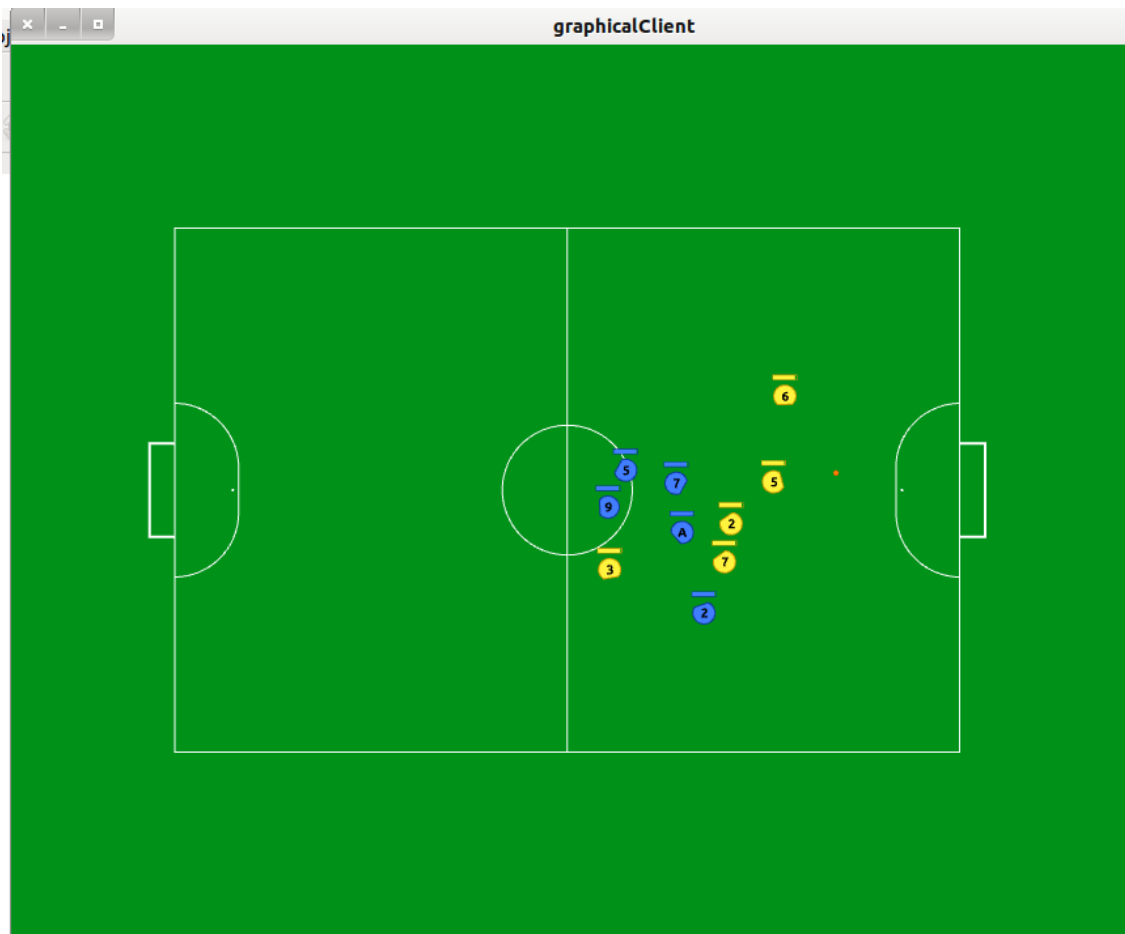


Ilustración 87: representación del campo en SSL-Vision [13]



## CAPÍTULO 4: EXPERIMENTOS

---

## 4. Experimentos

En este apartado se detallan todos los experimentos realizados para comprobar el éxito alcanzado con todo el trabajo realizado en el proyecto. En un primer lugar se explicarán todos los experimentos realizados con los clientes Java [16] de envío y recepción de datos, a continuación se detallarán los experimentos realizados con la arquitectura sobre el simulador grSim [7] [10], y por último, se detallarán las pruebas que se han realizado con el robot físico. Finalmente, se realiza una comparativa que muestra las diferencias de realizar los experimentos sobre el simulador y sobre los robots físicos.

En estos experimentos se mostrarán vídeos con el resultado de las pruebas. Estos vídeos se encuentran alojados en la página Youtube [38].

Es importante destacar que las pruebas realizadas en robots físicos se han realizado con el objetivo de comprobar la integración del trabajo realizado con robots reales que puedan participar en la competición SSL [5], aunque no se haya conseguido un resultado óptimo, estas pruebas sirven para comprobar la complejidad de trabajar con robots reales y las diferencias existentes con la realización de pruebas en un simulador.

### 4.1. Experimentos en Simulador grSim

En este punto se detallan todos los experimentos desarrollados en el simulador grSim [7] [10], tanto con los clientes Java [16] como con la arquitectura 3T desarrollada.

#### 4.1.1. Experimento con Cliente de Recepción Java

Este experimento verifica el correcto funcionamiento del cliente de recepción de datos implementado en Java [16]. Como ya se explicó en el apartado “3.3.2 Cliente Java de Recepción de Paquetes del Simulador” este cliente se ha creado para probar la conexión que permite recibir los paquetes con información que se obtienen del

simulador grSim [7] [10] o del sistema de visión SSL-Vision [12] y decodificar la información para poder ser utilizada en el desarrollo de la arquitectura.

- **NOMBRE:** cliente de recepción Java [16].
- **OBJETIVO:** comprobar que el sistema es capaz de recibir correctamente los paquetes con información que envía el simulador grSim [7] [10] y el sistema SSL-Vision [12].
- **DESCRIPCIÓN:** el cliente Java [16] de recepción de datos, se ha creado con el objetivo de crear la conexión que recibe los datos con la información del estado del partido, por lo tanto, con esta prueba se busca verificar que funciona correctamente.
- **EJECUCIÓN DE LA PRUEBA:** para realizar esta prueba ha sido necesario desarrollar el cliente Java [16]. Para ello, se ha creado un socket multicast para que reciba los paquetes con información, y una vez recibido el buffer con toda la información se realiza la decodificación utilizando Google Protocol Buffers [13].
- **RESULTADOS:** como puede observarse en la siguiente imagen, los datos se obtienen de forma correcta, y por lo tanto, se cumple con el objetivo buscado.

```
Output - ClienteReceptorInformacion (run)
run:
10
Puerto 10020
detection {
  frame_number: 401
  t_capture: 7.325
  t_sent: 7.325
  camera_id: 0
  balls {
    confidence: 0.9983925
    x: 0.0
    y: 0.0
    z: 21.499933
    pixel_x: 0.0
    pixel_y: 0.0
  }
  robots_yellow {
    confidence: 1.0
    robot_id: 0
    x: 1000.0206
    y: 750.0
    orientation: 3.1415925
    pixel_x: 1000.0206
    pixel_y: 750.0
  }
  robots_yellow {
    confidence: 1.0
    robot_id: 1
    x: 1000.0206
    y: -7.293178E-5
    orientation: 3.1415927
  }
}
```

Ilustración 88: experimento con cliente de recepción Java

- **PROBLEMAS:** el principal problema surgido para realizar este experimento es el cálculo del buffer de recepción, dado que no existe ningún tipo de método que permita conocer previamente el contenido, y por lo tanto, es necesario hacerlo todo de forma manual.
- **RESULTADO:** esta prueba verifica el correcto funcionamiento de la recepción de paquetes enviados por el simulador grSim [7] [10] o el sistema de visión SSL-Vision [12].

#### 4.1.2. Experimento con Cliente de Envío Java

Este experimento es realizado con el cliente de envío de comandos implementado en Java [16].

- **NOMBRE:** cliente de envío Java [16].
- **OBJETIVO:** verificar que los comandos que se envían son bien interpretados por el simulador grSim [7] [10] y por el robot físico.
- **DESCRIPCIÓN:** el cliente de envío de comandos en Java [16] se ha implementado con el objetivo de crear las conexiones de envío de comandos al simulador y el robot, por lo tanto, esta prueba tiene como fin comprobar que se envían correctamente los datos para que el robot realice los movimientos deseados.
- **EJECUCIÓN DE LA PRUEBA:** para realizar la ejecución de la prueba, el cliente Java [16] envía un paquete codificado con Google Protocol Buffers [13] a través de una conexión UDP.

Para facilitar el envío de paquetes, se ha creado una sencilla interfaz que permite el envío de comandos de forma muy intuitiva. Esta interfaz se muestra en la siguiente imagen:

The image shows a graphical user interface for a Java client. It features two columns of input fields. The left column contains: Simulator Address (127.0.0.1), Id (0), Velocity X (m/s) (2), Velocity Y (m/s) (0), Velocity Z (m/s) (0), a checkbox for 'Send Velocity? (or wheels)', Chip (m/s) (0), and a checkbox for 'Spin'. The right column contains: Simulator Port (20011), a dropdown menu set to 'Yellow', Wheel1 (m/s) (0), Wheel2 (m/s) (0), Wheel3 (m/s) (0), Wheel4 (m/s) (0), and Kick (m/s) (0). At the bottom, there are two buttons: 'Send' and 'Reset'.

Ilustración 89: cliente de envío Java

En la interfaz se pueden introducir los datos que se desean enviar al robot y para enviar el comando debe pulsarse en el botón “Send”. Si se desea resetear todos los parámetros de la interfaz, hay que pulsar sobre el botón “Reset”.

- **PROBLEMAS:** el principal problema surgido al realizar este experimento es la utilización de Google Protocol Buffers [13] para enviar el paquete correcto, así como la configuración del socket UDP para enviar el comando bien al robot o al simulador grSim [7] [10].
- **RESULTADO:** una vez enviado el paquete, se consigue que el robot indicado en la interfaz se mueva en función de los datos que se han enviado. Se puede observar en los vídeos [39] [40].

#### 4.1.3. Experimento con la Habilidad Giro Óptimo I

El experimento con la habilidad de giro óptimo tiene como finalidad comprobar que la habilidad del giro óptimo implementada, cumple con el objetivo para el que fue creada.

- **NOMBRE:** habilidad giro óptimo I.
- **OBJETIVO:** demostrar que la habilidad giro óptimo cumple con su cometido y el robot gira en el sentido de las agujas del reloj.
- **DESCRIPCIÓN:** esta prueba se realiza para comprobar que el robot es capaz de orientarse hacia la pelota realizando el giro óptimo en función de la posición en la que se encuentre la pelota. En este caso el robot se orienta hacia la pelota girando en el sentido de las agujas del reloj.
- **EJECUCIÓN DE LA PRUEBA:** para realizar la ejecución de la prueba es necesario ejecutar cualquier habilidad que contenga la habilidad de giro óptimo implícita desde la arquitectura 3T [36].
- **PROBLEMAS:** el principal problema surgido para la realización de la habilidad y el experimento fue el cálculo de las velocidades que debían tomar las ruedas para realizar el giro deseado. Para ello fue necesario estudiar el sistema de

referencias que utiliza el campo para poder realizar los giros deseados en función de esta información.

- **RESULTADO:** el robot gira correctamente hacia el lado deseado, cumpliendo con éxito con el objetivo de este experimento. Se puede observar en el vídeo [41].

#### 4.1.4. Experimento con la Habilidad Giro Óptimo II

El experimento con la habilidad de giro óptimo tiene como finalidad comprobar que la habilidad del giro óptimo implementada, cumple con el objetivo para el que fue creada.

- **NOMBRE:** habilidad giro óptimo II.
- **OBJETIVO:** demostrar que la habilidad giro óptimo cumple con su cometido y el robot gira en el sentido contrario de las agujas del reloj.
- **DESCRIPCIÓN:** esta prueba se realiza para comprobar que el robot es capaz de orientarse hacia la pelota realizando el giro óptimo en función de la posición en la que se encuentre la pelota. En este caso el robot se orienta hacia la pelota girando en el sentido contrario al de las agujas del reloj.
- **EJECUCIÓN DE LA PRUEBA:** para realizar la ejecución de la prueba es necesario ejecutar cualquier habilidad que contenga la habilidad de giro óptimo implícita desde la arquitectura 3T [36].
- **PROBLEMAS:** el principal problema surgido para la realización de la habilidad y el experimento fue el cálculo de las velocidades que debían tomar las ruedas para realizar el giro deseado. Para ello fue necesario estudiar el sistema de referencias que utiliza el campo para poder realizar los giros deseados en función de esta información.
- **RESULTADO:** el robot gira correctamente hacia el lado deseado, cumpliendo con éxito con el objetivo de este experimento. Se puede observar en el vídeo [42].



#### 4.1.5. Experimento con la Habilidad Defender Portería en Portero

El experimento con la habilidad de defender portería con el portero tiene como finalidad comprobar que la habilidad de defender la portería con el portero, cumple con el objetivo para el que fue creada.

- **NOMBRE:** habilidad defender portería en portero.
- **OBJETIVO:** demostrar que la habilidad defender portería en portero cumple con su cometido y el robot con el rol de portero es capaz de defender la portería para evitar que la pelota se introduzca en la portería propia.
- **DESCRIPCIÓN:** este experimento trata de verificar el correcto funcionamiento de la habilidad defender portería por el portero, la cual consiste en que el robot con el rol de portero se sitúa en la línea de gol de la portería que defiende y se mueve siguiendo la pelota con la intención de detener los disparos que se realicen sobre la portería.
- **EJECUCIÓN DE LA PRUEBA:** para ejecutar el experimento es necesario ejecutar en la arquitectura 3T [36], la habilidad que encadena las habilidades buscar portería y defender portería.
- **PROBLEMAS:** el principal problema surgido para la realización de la habilidad y el experimento fue el control de las posiciones para que el robot se mueva en el rango de posiciones de la portería y en especial, las correcciones necesarias para que el robot mantenga la posición en la línea de gol.
- **RESULTADO:** el robot mantiene la posición correctamente y sigue la posición de la pelota. Se puede observar en los vídeos [43] [44].

#### 4.1.6. Experimento con la Habilidad Defender Portería en Jugador

El experimento con la habilidad de defender portería con un robot con el rol de jugador de campo, tiene como finalidad comprobar que la habilidad de defender la portería, cumple con el objetivo para el que fue creada.

- **NOMBRE:** habilidad defender portería en jugador.

- **OBJETIVO:** demostrar que la habilidad defender portería en jugador cumple con su cometido y el robot con el rol de jugador es capaz de defender la portería para evitar que la pelota se introduzca en la portería propia.
- **DESCRIPCIÓN:** este experimento trata de verificar el correcto funcionamiento de la habilidad defender portería por un jugador, la cual consiste en que un robot con el rol de jugador se sitúa en la línea del área del campo propio y se mueve siguiendo la pelota con la intención de detener los disparos que se realicen sobre la portería.
- **EJECUCIÓN DE LA PRUEBA:** para ejecutar el experimento es necesario ejecutar en la arquitectura 3T [36], la habilidad que encadena las habilidades buscar área y defender portería.
- **PROBLEMAS:** el principal problema surgido para la realización de la habilidad y el experimento fue el control de las posiciones para que el robot se mueva en el rango de posiciones de la portería y en especial, las correcciones necesarias para que el robot mantenga la posición en la línea del área.
- **RESULTADO:** el robot mantiene la posición correctamente y sigue la posición de la pelota. Se puede observar en el vídeo [45].

#### 4.1.7. Experimento con la Habilidad Pase

El experimento con la habilidad de pase consiste en que se pruebe la habilidad de pase, en la que un robot realiza un pase a otro robot del equipo.

- **NOMBRE:** habilidad pase.
- **OBJETIVO:** demostrar que la habilidad pase cumple con su cometido y un robot realiza un pase a otro robot del mismo equipo.
- **DESCRIPCIÓN:** este experimento trata de comprobar el correcto funcionamiento de la habilidad pase, la cual consiste en que un robot se sitúa frente a la pelota, se orienta hacia el robot al que desea realizar el pase y realiza el pase.

- **EJECUCIÓN DE LA PRUEBA:** para ejecutar el experimento es necesario ejecutar en la arquitectura 3T [36], la habilidad que encadena las habilidades buscar portería y pase.
- **PROBLEMAS:** el principal problema a la hora de realizar el experimento fue conseguir que el robot se orientara en torno a la pelota hacia el robot al que se desea realizar el pase. Además, posteriormente se debería realizar el pase con la fuerza necesaria para que la pelota llegue al jugador receptor.
- **RESULTADO:** el robot realiza el pase hacia el jugador deseado de forma correcta. Se puede observar en los vídeos [46] [47] [48].

#### 4.1.8. Experimento con la Habilidad Esquivar

El experimento con la habilidad de esquivar trata de que el robot sea capaz de moverse por el campo de juego esquivando los obstáculos que se encuentre por el camino sin impactar con ninguno de ellos.

- **NOMBRE:** habilidad esquivar.
- **OBJETIVO:** demostrar que la habilidad de esquivar funciona correctamente y el robot esquiva todos los obstáculos que se encuentren en el camino.
- **DESCRIPCIÓN:** este experimento trata de verificar el correcto funcionamiento de la habilidad de esquivar, la cual consiste en que un robot se mueve por el terreno de juego sin impactar con ningún robot, evitando así colisiones.
- **EJECUCIÓN DE LA PRUEBA:** para ejecutar el experimento es necesario ejecutar en la arquitectura 3T [36] cualquier habilidad habiendo activado la opción de esquivar.
- **PROBLEMAS:** el principal problema surgido para la realización de la habilidad y el experimento fue el control de las posiciones de todos los robots y los movimientos a realizar para que el robot esquivara los obstáculos, además de permitir la ejecución en paralelo con cualquiera de las habilidades en las que se ha integrado la habilidad de esquivar.

- **RESULTADO:** el robot esquiva los robots que se encuentra por el camino mientras se mueve por el terreno de juego. Se puede observar en el vídeo [49].

#### 4.2. Experimentos en Robot Físico

Además de todos los experimentos explicados anteriormente que han sido realizados en el simulador, se han realizado las siguiente pruebas utilizando el robot físico del que dispone la universidad.

Estos experimentos son más sencillos que los realizados en el simulador, debido a las limitaciones existentes como que solo se dispone de un robot físico, el espacio del que se dispone es limitado y la dificultad de adaptar todo lo desarrollado a los robots físicos.

Todas estas pruebas fueron realizadas en el laboratorio 2.2.B06 en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid. En la siguiente imagen se muestra el entorno en el que se ha trabajado:

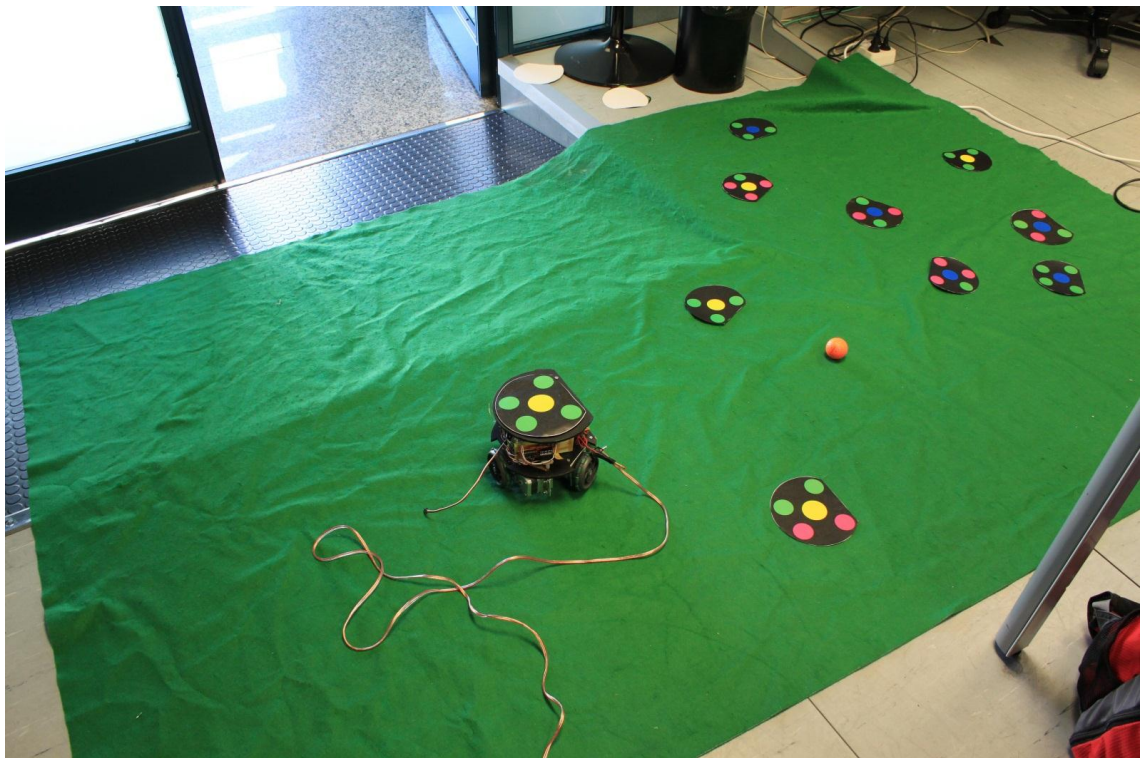


Ilustración 90: entorno de realización de las pruebas con el robot físico

Todos los experimentos realizados tienen como objetivo adaptar la arquitectura 3T [36] desarrollada, en el robot físico desarrollado por la Universidad Carlos III de Madrid en los proyectos de fin de carrera de Lidia Escudero Jiménez [31] y Jose Luis Martín Gómez [32].

El robot recibe todos los comandos vía Wi-Fi y es alimentado por cable. En la siguiente imagen se muestra el robot utilizado:

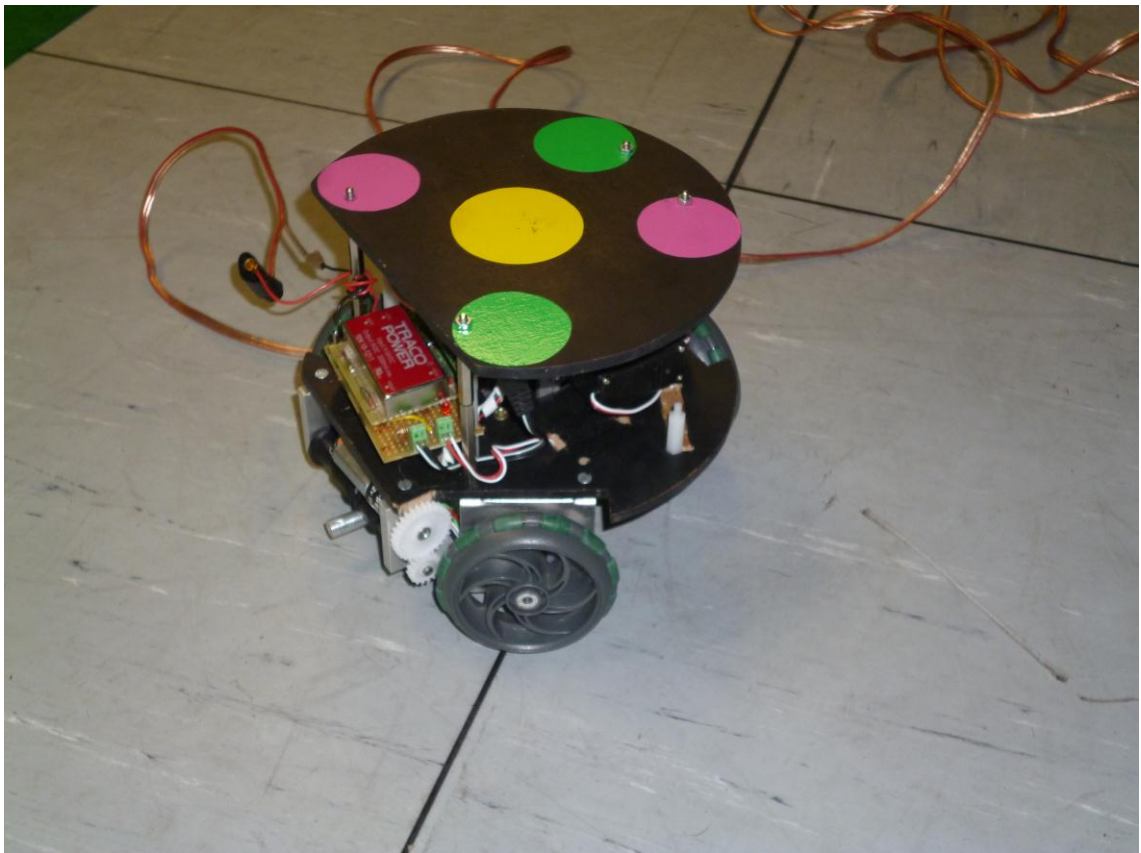


Ilustración 91: robot físico

A continuación se explican todas las pruebas realizadas:

#### 4.2.1. Experimento Buscar Pelota

Se ha realizado la prueba de buscar pelota con el robot físico.

- **NOMBRE:** buscar pelota.

- **OBJETIVO:** demostrar que la habilidad pase cumple con su cometido y el robot busca la pelota.
- **DESCRIPCIÓN:** con este experimento se intenta conseguir que el robot físico encuentre la pelota situada en el terreno de juego.
- **EJECUCIÓN DE LA PRUEBA:** para ejecutar hay que indicar la velocidad que se le da a las ruedas.
- **PROBLEMAS:** el principal problema a la hora de realizar el experimento fue corregir las velocidades de las ruedas para que el robot llegue hacia la pelota.
- **RESULTADO:** el robot encuentra la pelota correctamente como se muestra en los vídeos [50] [51] [56].

#### 4.2.2. Problemas Surgidos con el Robot Físico

En la realización de las pruebas con el robot físico, han surgido diversos problemas debido a la dificultad que supone la utilización de un robot físico en lugar del simulador.

La mayoría de estos problemas surgen como consecuencia de la asignación de velocidades a las ruedas como se muestra en los siguientes vídeos [52] [53].

También surgen problemas con el estado del terreno y las limitaciones de espacio. En el siguiente vídeo se ve como el robot se choca con la rampa existente en el laboratorio [55].

Otro tipo de problemas es el cálculo erróneo de posiciones como ocurre en el siguiente vídeo [54].

#### 4.3. Resultados Finales

A continuación, se detallan los resultados obtenidos tras la realización de todos los experimentos tanto en el simulador como en los robots físicos.

Se puede deducir que las pruebas realizadas en el robot físico tienen mayor dificultad de ejecución que las pruebas desarrolladas en el simulador. Esto se debe



a diversos factores tales como las condiciones del entorno tales como la rugosidad del terreno, luminosidad del ambiente, etc. También afecta el rozamiento de las ruedas y la precisión de las velocidades de las ruedas, siendo necesario modificar las velocidades para eliminar el desfase existente.

Respecto a las conexiones se ha comprobado que funcionan correctamente, dado que el sistema es capaz de enviar comandos al simulador y al robot físico, y además el sistema recibe los paquetes con información y los decodifica para así poder utilizar esa información.

También se puede deducir que las habilidades tienen un comportamiento correcto y preciso adaptado a las condiciones en las que se trabaja, pero para el futuro es posible que haya que realizar pequeñas modificaciones para adaptar todo a las distintas condiciones que se presentan.

Tras analizar todos los datos obtenidos en la realización de los distintos experimentos, se puede afirmar que se ha conseguido desarrollar con éxito los objetivos planteados inicialmente.



## CAPÍTULO 5: PRESUPUESTO

---



## 5. Presupuesto

En este apartado se detallará el presupuesto del proyecto. Por un lado se mostrará la descripción general del proyecto, donde se indican los datos básicos del proyecto. También se detallan todos los gastos que implica el desarrollo del proyecto. Por último, se muestra un resumen total del coste final del proyecto.

### 5.1. Descripción del Proyecto

En la siguiente tabla se muestra un resumen con los datos más relevantes del proyecto:

<b>Nombre del proyecto</b>	<b>Implementación de Arquitectura de Control 3T en la RoboCup Small Size League: Formación de Defensa</b>
<b>Autor</b>	Alejandro Caparrós Hernando
<b>Duración</b>	9 meses
<b>Tasa de costes indirectos</b>	20%

Tabla 5: descripción del proyecto

### 5.2. Desglose del Presupuesto

A continuación se muestra desglosado el presupuesto en función de los distintos tipos de coste.

#### 5.2.1. Personal

A continuación se detalla todo el coste del proyecto en personal. Se detalla en la siguiente tabla:



Nombre y apellidos	Categoría	Duración (meses)	Coste por mes (€)	Coste total del proyecto (€)
Alejandro Caparrós Hernando	Ingeniero	9	2.200 €	19.800 €

Tabla 6: coste de personal

### 5.2.2. Material

A continuación se detalla todo el coste del proyecto en material. Se detalla en la siguiente tabla:

Producto	Coste (€)	% Uso dedicado al proyecto	Dedicación (meses)	Periodo de depreciación (meses)	Coste imputable
Apple MacBook Pro 13,3''	1.145 €	10%	9	60	17,18 €
Microsoft Office 2010	139 €	30%	9	48	7,92 €
Ubuntu 11	0 €	80%	9	60	0 €
Gimp	0 €	80%	9	60	0 €
Cámara AVT Stingray	858 €	50%	9	60	64,35 €
Robot Físico	1.143,18 €	30%	9	60	51,44 €
<b>Total</b>	<b>140,89 €</b>				

Tabla 7: coste de material

### 5.3. Resumen del Proyecto

En la siguiente tabla se muestra un resumen con los datos más relevantes del proyecto:

<b>Concepto</b>	<b>Coste</b>
<b>Personal</b>	19.800,00 €
<b>Material</b>	140,89 €
<b>Total</b>	<b>19.940,89 €</b>

Tabla 8: resumen del proyecto



## CAPÍTULO 6: CONCLUSIONES

---

## 6. Conclusiones

En este punto se detallan las conclusiones obtenidas tras el desarrollo de este proyecto.

Cabe recordar que el objetivo principal de este proyecto consistía en implementar una arquitectura 3T [36] en robots que cumplieran los requisitos necesarios para participar en la Competición Internacional RoboCup SSL [5]. Además, era necesario desarrollar ciertas habilidades básicas defensivas que se complementarían con las desarrolladas en el PFC de Verónica Raspeño Gutiérrez [21], las cuales fueron desarrolladas de forma conjunta.

Tras realizar todas las pruebas y analizar los resultados obtenidos se puede afirmar que el resultado final ha sido satisfactorio, habiendo cumplido con los objetivos marcados inicialmente y que detallaremos en los siguientes párrafos. Además, también se ha conseguido asentar las bases para la formación de un equipo que sea capaz de participar en la RoboCup SSL [5] en el futuro.

El primero de los objetivos alcanzado estaba relacionado con la creación de un cliente en Java [16] capaz de enviar comandos simples al simulador grSim [7] [10] y a los robots físicos. Para ello, se ha creado una interfaz muy sencilla en la que se pueden indicar los distintos parámetros configurables y enviarlos a la dirección IP y puerto indicados. Análogamente, se ha creado un pequeño cliente en Java [16] que sea capaz de recibir e interpretar los datos con información que envía el simulador grSim [7] [10] en paquetes UDP *multicast*.

El siguiente objetivo alcanzado ha sido la adaptación de la arquitectura 3T [36] desarrollado en el PFC de José Luis Díaz Cebrián [20] a los robots de la RoboCup SSL [5]. Para llevar a cabo este objetivo, se han utilizado las conexiones creadas en los clientes Java [16] para integrarlas en la arquitectura y de esta manera que sea capaz de enviar comandos y actualizar la información de los equipos.

También es destacable que ha sido necesario modificar el simulador grSim [7] [10] para adaptar los robots simulados a los robots físicos de los que se dispone. En este caso el cambio consistía en eliminar una rueda de los robots, puesto que los robots del simulador por defecto tienen 4 ruedas y el robot físico dispone de 3 ruedas. Además se ha colaborado de forma muy activa con el equipo iraní Parsian Robotic [11] ayudando en la creación de documentación y aportando información de utilidad sobre el simulador grSim [7] [10].

Como pilar principal de este proyecto se han implementado ciertas habilidades básicas para los robots de la RoboCup SSL [5] orientadas a acciones de defensa, que se complementan con las habilidades de ataque desarrolladas en el proyecto de Verónica Raspeño Gutiérrez [21]. Estas habilidades dotan a los robots de la capacidad de realizar algunas habilidades básicas para poder participar en la competición SSL.

Además, las habilidades implementadas se han probado en el simulador y en el robot físico, lo que permite comparar la diferencia entre trabajar sobre un simulador o sobre un robot real, el cual presenta mayores problemas como la velocidad de los motores, la inercia, etc.

Como conclusión general, cabe destacar que se han alcanzado todos los objetivos propuestos inicialmente y se ha conseguido asentar las bases para continuar con el desarrollo de un equipo que sea capaz de participar en la RoboCup SSL [5] cumpliendo con los exigentes requisitos de esta competición.



## CAPÍTULO 7: LÍNEAS DE TRABAJO

---



## 7. Futuras líneas de trabajo

El trabajo realizado en el desarrollo de este proyecto asienta las bases para el desarrollo de un equipo, que en un futuro cumpla las normas de la RoboCup SSL [5] y tenga opciones de participar en ella (actualmente no hay ningún equipo español en esta competición). Puesto que con el desarrollo de este proyecto se consigue que los robots sean capaces de realizar ciertas habilidades básicas, este proyecto es fácilmente ampliable para conseguir una funcionalidad correcta, precisa y eficaz, mientras se cumplen los requisitos básicos para participar en esta competición.

La primera mejora que se puede realizar sobre la arquitectura ampliada sería la de ampliar el número de habilidades existentes y conseguir mayor precisión en las habilidades desarrolladas para conseguir un mejor funcionamiento de los robots en el transcurso de un partido y que de esta manera, los robots sean capaces de realizar todo tipo de habilidades necesarias para intentar ganar el partido. Algunas posibles habilidades que se podrían implementar para complementar las desarrolladas en este proyecto y en el PFC de Verónica Raspeño Gutiérrez [21] son crear formaciones, realizar saques y lanzamientos de penalti, defender en bloque, etc.

Otra posible línea de mejora para el futuro sería la creación de roles para los distintos robots que componen el equipo. Los posibles roles podrían ser delantero, defensa, portero, etc. En la actualidad únicamente existen dos roles en los robots: portero, y no portero (que se comportan como jugadores *normales* de campo). Estos roles permitirían tener comportamientos específicos en función del rol asignado a cada robot.

El planificador de la arquitectura 3T [36] no ha sido implementado, ya que en este proyecto se han desarrollado habilidades que se ejecutan en la capa reactiva (Controlador) de la arquitectura, por lo tanto, una línea de mejora sería la implementación del planificador que permita la creación de planes y de esta forma, permitir que se ejecuten distintas estrategias en función de las condiciones del partido. Por ejemplo, si un equipo va perdiendo y queda poco para la finalización del partido, el

equipo podría ejecutar un plan de formación ofensiva y hacer así que todos los robots del equipo ataquen para intentar remontar el partido.

También sería interesante la creación de distintas formaciones del equipo como son la ofensiva, la defensiva, la de presión, etc. Estas formaciones permiten cambiar el ritmo del partido en función de los intereses del equipo en ese momento del partido.

La adaptación de las habilidades a los robots físicos es una mejora importante a desarrollar, dado que debido a las características del robot no siempre se desarrolla todo al igual que en el simulador, donde en todo momento se toman unas condiciones ideales del entorno y se ejecutan de forma correcta. Por lo tanto, es importante perfeccionar estas habilidades para que el robot físico sea capaz de realizar las habilidades de la forma más precisa posible.

Además de esto sería interesante crear mayor número de robots físicos, para permitir la cooperación entre los robots y de esta forma poder probar mayor número de habilidades con los robots físicos: pases entre robots, utilizar un robot que cumpla el rol de portero y otro que haga el rol de delantero, etc. Esto ayudaría en gran medida en la integración de las habilidades desarrolladas en los robots reales.

El trabajo realizado también podría ser utilizado para su inclusión en otras ligas con características similares. Así podrían aprovecharse las bases de la arquitectura para integrarlas a robots de otras categorías como puede ser la humanoide, tamaño mediano, simulación, etc. En estos casos sería importante estudiar todo el funcionamiento de las ligas y ver cuáles son las reglas para adaptar todo el trabajo a estos nuevos robots.

Además, también puede ser interesante la obtención de algoritmos que permitan a los robots aprender de sus errores y que permitan que estos mejoren su comportamiento. La utilización de técnicas de la I.A. como redes de neuronas o diferentes clasificadores podría permitir a los robots realizar determinadas acciones en el campo de juego en función de los resultados de dichas técnicas y mejorar así su juego.



Debido a las características que tiene el simulador grSim [7] [15], se podrían incluir ciertas mejoras que facilite la integración de estas habilidades y se podrían dotar de ciertas características al simulador que mejoren la integración de los equipos. Así, por ejemplo, que se podría permitir la inclusión de mayor número de robots o el cambio de las características de los robots simulados. Esto sería una mejora importante, ya que permitiría a cada equipo personalizar los robots en función de las características que éstos posean.

Como última conclusión cabe destacar que la RoboCup SSL [5] es una de las ligas más avanzadas y maduras en la actualidad y en la que se encuentran trabajando una gran cantidad de equipos de todo el mundo. Esto hace que podamos obtener herramientas y documentos científicos al respecto; sin embargo y a pesar de los muchos esfuerzos que se han hecho a nivel internacional en esta competición y lo que ella conlleva, todavía queda mucho por mejorar y perfeccionar. Así, este PFC a contribuido a desarrollar este objetivo.



## BIBLIOGRAFÍA

---

## Bibliografía

1. Real Academia Española:  
<http://www.rae.es/>
2. Wikipedia:  
<http://www.wikipedia.org/>
3. Robots autónomos:  
<http://www.fleifel.net/ia/robotsyaprendizaje.php>  
FLEIFEL, F., *Robots Autónomos y Aprendizaje por Refuerzo*, 2002.
4. RoboCup:  
<http://www.robocup.org/>
5. Small Size League:  
<http://small-size.informatik.uni-bremen.de/>
6. Tewnta:  
<http://code.google.com/p/tewnta/>
7. grSim:  
<http://www.parsianrobotic.ir/grsim/>
8. Carnegie Mello University:  
<http://www.cs.cmu.edu/~robosoccer/small/>
9. RoboCup University:  
<http://code.google.com/p/robocup-simulator/feeds/>
10. MACKWORTH, A.: "On Seeing Robots", 1992
11. Parsian Robotic:  
<http://www.parsianrobotic.ir/>
12. SSL-Vision:  
<http://code.google.com/p/ssl-vision/>
13. Google Protocol Buffers:  
[developers.google.com/protocol-buffers/](http://developers.google.com/protocol-buffers/)
14. Open Dynamics Engine:  
<http://www.ode.org/>

15. VALIALLAH, M., KOOCHAKZADEH, A., SHIRY GHIDARY, S., *RoboCup 2011: Robot Soccer World Cup XV*, 2011.
16. Java:  
<http://www.java.com/es/>
17. C++:  
<http://www.cplusplus.com/>
18. Reglas de RoboCup SSL 2012:  
[http://small-size.informatik.uni-bremen.de/ media/rules:ssl-rules-2012-diff.pdf](http://small-size.informatik.uni-bremen.de/media/rules:ssl-rules-2012-diff.pdf)
19. Linux:  
[http://es.wikipedia.org/wiki/N%C3%BAcleo\\_Linux](http://es.wikipedia.org/wiki/N%C3%BAcleo_Linux)
20. DÍAZ CEBRIÁN, José Luis, *Implementación de una Arquitectura de Control Híbrida para Robots Autónomos*, Universidad Carlos III de Madrid, Leganés (Madrid), 2011.
21. RASPEÑO GUTIÉRREZ, Verónica, *Implementación de Arquitectura de Control 3T en la RoboCup Small Size League: Formación de Ataque*, Universidad Carlos III de Madrid, Leganés (Madrid), 2012.
22. Microsoft Office:  
<http://office.microsoft.com/es-es/>
23. Netbeans:  
<http://netbeans.org/>
24. Gimp:  
<http://www.gimp.org.es/>
25. Apple Mac OS X:  
<http://www.apple.com/es/macosex/>
26. Ubuntu:  
<http://www.ubuntu.com/>
27. Windows:  
<http://windows.microsoft.com/es-es/windows/home/>  
Wireshark:  
<http://www.wireshark.org/>
29. Apple:

- <http://store.apple.com>
30. GARCÍA LÓPEZ, Álvaro, *Desarrollo del Sistema de Locomoción de una Plataforma Hardware para RoboCup Small Soccer League (SSL)*, Universidad Carlos III de Madrid, Leganés (Madrid), 2010.
  31. ESCUDERO JIMÉNEZ, Lidia, *Sistema de Procesamiento, Alimentación y Estructura de un Microrobot*, Universidad Carlos III de Madrid, Leganés (Madrid), 2012.
  32. MARTÍN GÓMEZ, José Luis, *Diseño del Sistema de Locomoción y Dribbler de un Microrobot*, Universidad Carlos III de Madrid, Leganés (Madrid), 2012.
  33. Cámara AVT Stingray:  
<http://www.1stvision.com/cameras/AVT/Stingray-F-046-B-C.html>
  34. Aldebaran Robotics:  
<http://www.aldebaran-robotics.com/en/>
  35. Sony:  
<http://www.sony.es>
  36. BONASSO, R.P., FIRBY, J., GAT, E., KORTENKAMP, D., MILLER, D., SLACK, M., Experiences with an Architecture for Intelligent Reactive Agents. Journal of Experimental and Theoretical Artificial Intelligence, vol. 9, no. 2. 2007.
  37. Pioneer:  
<http://www.pioneer.eu/es/body.html>
  38. Youtube:  
<http://www.youtube.com/>
  39. Prueba cliente de envío Java 1:  
<http://www.youtube.com/watch?v=X0fqGrTm52c&feature=youtu.be>, consultado el día 10/9/2012.
  40. Prueba cliente de envío Java 2:  
<http://www.youtube.com/watch?v=EsnGRW2pfJY&feature=youtu.be>, consultado el día 10/9/2012.
  41. Prueba giro óptimo a la izquierda:  
<http://www.youtube.com/watch?v=JfjsG5IHF8w>, consultado el día 10/9/2012.
  42. Prueba giro óptimo a la derecha:

<http://www.youtube.com/watch?v=MyT1AU1ZFVvk&feature=youtu.be>,

consultado el día 10/9/2012.

43. Prueba portero 1:

<http://www.youtube.com/watch?v=I9DiCL5t4HI>, consultado el día 12/9/2012.

44. Prueba portero 2:

<http://www.youtube.com/watch?v=1EMJS2N6LW8>, consultado el día 12/9/2012.

45. Prueba defender:

<http://www.youtube.com/watch?v=liCuucSxESA>, consultado el día 12/9/2012.

46. Prueba pase 1:

<http://www.youtube.com/watch?v=sj3Skve-rww>, consultado el día 14/9/2012.

47. Prueba pase 2:

[http://www.youtube.com/watch?v=X\\_uXxrpVVys](http://www.youtube.com/watch?v=X_uXxrpVVys), consultado el día 14/9/2012.

48. Prueba pase 3:

[http://www.youtube.com/watch?v=oaWlQJFkz\\_w](http://www.youtube.com/watch?v=oaWlQJFkz_w), consultado el día 14/9/2012.

49. Prueba esquivar:

<http://www.youtube.com/watch?v=ZVKIYiVxra0>, consultado el día 14/9/2012.

50. Prueba buscar pelota en robot físico 1:

<http://www.youtube.com/watch?v=odSd6EvFvwY>, consultado el día 15/9/2012.

51. Prueba buscar pelota en robot físico 2:

[http://www.youtube.com/watch?v=r6\\_vFLh0vNM&feature=youtu.be](http://www.youtube.com/watch?v=r6_vFLh0vNM&feature=youtu.be), consultado el día 15/9/2012.

52. Prueba búsqueda de pelota errónea 1:

[http://www.youtube.com/watch?v=XFm89a5H\\_R4&feature=youtu.be](http://www.youtube.com/watch?v=XFm89a5H_R4&feature=youtu.be), consultado el día 15/9/2012.

53. Prueba búsqueda de pelota errónea 2:

<http://www.youtube.com/watch?v=4QsU9cnSUmQ>, consultado el día 15/9/2012.

54. Prueba búsqueda de pelota errónea 3:





<http://www.youtube.com/watch?v=6mb9XIWx2OA&feature=youtu.be>, consultado el día 15/9/2012.

55. Prueba búsqueda de pelota errónea 4:

<http://www.youtube.com/watch?v=HZMxQ6pZ2I0&feature=youtu.be>, consultado el día 15/9/2012.

56. Prueba búsqueda pelota:

<http://www.youtube.com/watch?v=vopgQXR1Ob8&feature=youtu.be>, consultado el día 15/9/2012.



# ANEXO I: DEFINICIONES Y ACRÓNIMOS

---

## ANEXO I: Definiciones y acrónimos

A continuación se reúnen todas las definiciones y acrónimos de interés aparecidos a lo largo del proyecto.

### Definiciones

- **Aldebaran Robotics:** empresa francesa con sede en París que desarrolla el robot “Nao”. [2]
- **Apple:** empresa multinacional estadounidense que diseña y produce equipos electrónicos y software. [2]
- **AVT Stingray:** cámara utilizada para el sistema de visión SSL Vision [12] de la Robocup SSL [5].
- **C++:** lenguaje de programación cuyo objetivo era el de extender el lenguaje de programación C con mecanismos que permiten la manipulación de objetos. [2]
- **Carnegie Mellon:** universidad que se ubica en Pittsburgh (Pensilvania) y es uno de los centros de investigación superior más destacados de Estados Unidos en el área de informática y robótica. [2] [8]
- **FIFA:** Federación Internacional de Fútbol Asociación, universalmente conocida por las siglas FIFA, es la institución que gobierna las federaciones de fútbol en todo el planeta. [2]
- **Google:** empresa propietaria de la marca Google, cuyo principal producto es el motor de búsqueda de contenido en Internet del mismo nombre. [2]
- **Google Protocol Buffers:** sistema que se utiliza para codificar datos estructurados en un formato eficiente pero ampliable. [2]
- **Java:** lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems en 1995. [2]
- **Linux:** núcleo libre de sistema operativo basado en Unix. [2]
- **Microsoft:** empresa multinacional dedicada a la informática de origen estadounidense, fundada en 1975 por Bill Gates y Paul Allen. [2]
- **Microsoft Excel 2010:** aplicación para manejar hojas de cálculo. Es desarrollado y distribuido por Microsoft. [2]

- **Microsoft Office:** suite ofimática que abarca e interrelaciona aplicaciones de escritorio, servidores y servicios. [2]
- **Microsoft PowerPoint 2010:** programa para crear presentaciones, desarrollado por la empresa Microsoft. [2]
- **Microsoft Project 2010:** software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft. [2]
- **Microsoft Word 2010:** procesador de textos desarrollado por Microsoft, integrado en la suite ofimática Microsoft Office 2010. [2]
- **Multicast:** envío de información en una red a múltiples destinos de forma simultánea. [2]
- **Netbeans:** entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un gran número de módulos que añaden funcionalidades. [2]
- **NAO:** organismo autónomo, programable y de mediana estatura, desarrollado por la empresa francesa Aldebaran Robotics. [2]
- **OS XLion:** octava versión del sistema operativo de Apple para sus ordenadores de escritorio, portátiles y servidores. [2]
- **Parsian Robotic:** grupo de estudiantes que trabajan en el departamento de ingeniería eléctrica de la universidad Amir Kabir en Irán. Componen un equipo participante de la RoboCup SSL [5]. [11]
- **Sony:** es una de las empresas más grandes del mundo de origen japonés y uno de los fabricantes líder en la electrónica de consumo, el audio y el vídeo, los videojuegos y las tecnologías de la información y la comunicación. [2]
- **Tewnta:** simulador utilizado para la RoboCup SSL [5]. [6]
- **Ubuntu:** sistema operativo mantenido por Canonical y la comunidad de desarrolladores. Utiliza núcleo Linux y su origen está basado en Debian. [2]
- **Wifi:** mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. [2]
- **Windows 7:** versión más reciente del sistema operativo Microsoft Windows. [2]
- **Wireshark:** analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica para educación. [2]
- **Youtube:** sitio web en el cual los usuarios pueden subir y compartir vídeos. [2]



## Acrónimos

<b>Acrónimo</b>	<b>Definición</b>
<b>cm</b>	Centímetros
<b>FIFA</b>	Fédération Internationale de Football Association
<b>Id</b>	Identificador
<b>PFC</b>	Proyecto Final de Carrera
<b>SSL</b>	Small Size League
<b>UDP</b>	User Datagram Protocol

Tabla 9: acrónimos



## ANEXO II: REQUISITOS DE USUARIO

---

## ANEXO II: Requisitos de Usuario

A continuación se detallarán los requisitos de usuario, siguiendo la siguiente estructura:

<b>ID</b>	Identificador		
<b>Nombre</b>	Nombre de requisito		
<b>Descripción</b>	Descripción de requisito		
<b>Necesidad</b>	Necesidad de requisito	<b>Prioridad</b>	Prioridad de requisito
<b>Estabilidad</b>	Estabilidad de requisito	<b>Claridad</b>	Claridad de requisito

Tabla 10: requisito

Cuyos campos significan:

- **ID:** identificador del requisito, con la estructura REQ-TIPO-XX, siendo:
  - **TIPO:**
    - **SIM:** requisitos del simulador grSim [7] [15].
    - **ENV:** requisitos de cliente Java [16] de envío.
    - **REC:** requisitos de cliente Java [16] de recepción.
    - **ARQ:** requisitos de arquitectura 3T [36].
    - **HAB:** requisitos de habilidades.
    - **FIS:** requisitos de aplicación en robots físicos.
  - **XX:** el número de requisito.
- **Nombre:** identificativo del requisito, es un pequeño resumen de lo que representa el requisito.
- **Descripción:** breve descripción del requisito.
- **Necesidad:** del requisito, cómo de necesitado es para el sistema, los posibles valores son:
  - Esencial.
  - Deseable.

- Opcional.
- **Prioridad:** importancia del requisito. Siendo posibles los siguientes valores:
  - Alta.
  - Media.
  - Baja.
- **Estabilidad:** cómo de estable es el requisito, pudiendo ser:
  - Estable.
  - No estable.
- **Claridad:** si el requisito es claro o no, pudiendo ser:
  - Alta.
  - Media.
  - Baja.

## 1 Requisitos de simulador grSim

<b>ID</b>	<b>REQ-SIM-01</b>		
<b>Nombre</b>	Sistema operativo		
<b>Descripción</b>	El simulador utilizado será ejecutable en cualquier sistema operativo Unix.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 11: REQ-SIM-01

<b>ID</b>	<b>REQ-SIM-02</b>		
<b>Nombre</b>	Simulación		
<b>Descripción</b>	El simulador simulará el terreno de juego, robots y balón propios de la SSL [5].		



<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 12: REQ-SIM-02

<b>ID</b>	<b>REQ-SIM-03</b>		
<b>Nombre</b>	SSL-Vision		
<b>Descripción</b>	El simulador simulará el procesamiento de imágenes, así como el SSL-Vision [12] propio de la liga SSL [5].		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 13: REQ-SIM-03

## 2 Requisitos de cliente Java de envío

<b>ID</b>	<b>REQ-ENV-01</b>		
<b>Nombre</b>	Lenguaje programación		
<b>Descripción</b>	Se hará un cliente en lenguaje Java [16] que permita enviar paquetes al simulador grSim [7] [15] y a los robots físicos.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 14: REQ-ENV-01

<b>ID</b>	<b>REQ-ENV-02</b>		
<b>Nombre</b>	Interfaz		
<b>Descripción</b>	Se creará una interfaz en Java [16] en donde se puedan configurar todos los parámetros relacionados con los robots.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 15: REQ-ENV-02

<b>ID</b>	REQ-ENV-03		
<b>Nombre</b>	Movimiento robots simulados		
<b>Descripción</b>	El cliente permitirá mover los robots en el simulador grSim [7] [15] con el paquete que envíe.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 16: REQ-ENV-03

<b>ID</b>	REQ-ENV-04		
<b>Nombre</b>	Movimiento robots físicos		
<b>Descripción</b>	El cliente permitirá mover los robots físicos con el paquete que envíe.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 17: REQ-ENV-04

<b>ID</b>	REQ-ENV-05		
<b>Nombre</b>	Estructura paquete		
<b>Descripción</b>	El paquete enviado seguirá un estándar dado por la organización de la competición RoboCup [4].		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 18: REQ-ENV-05

<b>ID</b>	REQ-ENV-06		
<b>Nombre</b>	Google Protocol Buffers [13]		
<b>Descripción</b>	El paquete enviado estará codificado por Google Protocol Buffers [13].		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 19: REQ-ENV-06

### 3 Requisitos de cliente Java de recepción

<b>ID</b>	<b>REQ-REC-01</b>		
<b>Nombre</b>	Lenguaje programación		
<b>Descripción</b>	Se hará un cliente en lenguaje Java [16] que permita recibir paquetes al simulador grSim [7] [15] y a los robots físicos.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 20: REQ-REC-01

<b>ID</b>	<b>REQ-REC-02</b>		
<b>Nombre</b>	Procesamiento de paquetes		
<b>Descripción</b>	El cliente será capaz de procesar el paquete recibido, siempre que respete la estructura dada por la organización de la competición RoboCup [4].		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 21: REQ-REC-02

<b>ID</b>	<b>REQ-REC-03</b>		
<b>Nombre</b>	Google Protocol Buffers [13]		
<b>Descripción</b>	El paquete recibido estará codificado por Google Protocol Buffers [13], el cliente será capaz de interpretarlo.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 22: REQ-ENV-03

## 4 Requisitos de arquitectura 3T

<b>ID</b>	REQ-ARQ-01		
<b>Nombre</b>	Arquitectura SSL [5]		
<b>Descripción</b>	Creación de una arquitectura para la SSL [9] de la RoboCup [4].		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 23: REQ-ARQ-01

<b>ID</b>	REQ-ARQ-02		
<b>Nombre</b>	Lenguaje programación		
<b>Descripción</b>	La arquitectura se programará en el lenguaje de programación Java [16].		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 24: REQ-ARQ-02

<b>ID</b>	REQ-ARQ-03		
<b>Nombre</b>	Tipo arquitectura		
<b>Descripción</b>	Se utilizará la arquitectura 3T [36], basada en la creada por José Luis Cebrián Díaz. [20]		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 25: REQ-ARQ-03

<b>ID</b>	REQ-ARQ-04		
<b>Nombre</b>	Comunicación con simulador		
<b>Descripción</b>	La arquitectura se comunicará con el simulador grSim [7] [15].		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 26: REQ-ARQ-04

<b>ID</b>	<b>REQ-ARQ-05</b>		
<b>Nombre</b>	Recepción datos		
<b>Descripción</b>	La arquitectura será capaz de procesar y utilizar los datos enviados por el simulador grSim [7] [15].		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 27: REQ-ARQ-05

<b>ID</b>	<b>REQ-ARQ-06</b>		
<b>Nombre</b>	Envío datos		
<b>Descripción</b>	La arquitectura será capaz de enviar datos al simulador grSim [7] [15] y que este los procese y utilice.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 28: REQ-ARQ-06

<b>ID</b>	<b>REQ-ARQ-07</b>		
<b>Nombre</b>	Datos de entrada robots simulados		
<b>Descripción</b>	Los datos de entrada serán proporcionados exclusivamente por el paquete que se recibe del grSim [7] [15] en la simulación.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 29: REQ-ARQ-07

<b>ID</b>	<b>REQ-ARQ-08</b>		
<b>Nombre</b>	Datos de entrada robots físicos		
<b>Descripción</b>	Los datos de entrada serán proporcionados exclusivamente por el paquete que se recibe del SSL-Vision [12] en los robots físicos.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta

<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta
--------------------	---------	-----------------	------

Tabla 30: REQ-ARQ-08

<b>ID</b>	<b>REQ-ARQ-09</b>		
<b>Nombre</b>	Hilo recepción paquetes		
<b>Descripción</b>	Se tendrá un hilo de ejecución para recibir los paquetes que envía el simulador [7] [15] en caso de la simulación y el SSL-Vision [12] en caso de los robots físicos.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 31: REQ-ARQ-09

## 5 Requisitos de habilidades de SSL

<b>ID</b>	<b>REQ-HAB-01</b>		
<b>Nombre</b>	Habilidades		
<b>Descripción</b>	Se crearán distintas habilidades básicas que imiten el comportamiento de robots que juegan al fútbol.		
<b>Necesidad</b>	Esencial	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 32: REQ-HAB-02

<b>ID</b>	<b>REQ- HAB-02</b>		
<b>Nombre</b>	Hilo habilidad esquivar		
<b>Descripción</b>	Se creará un hilo de ejecución para realizar la habilidad de esquivar por parte de los robots.		
<b>Necesidad</b>	Deseable	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 33: REQ-HAB-02

<b>ID</b>	<b>REQ- HAB-03</b>		
<b>Nombre</b>	Habilidades complementarias		
<b>Descripción</b>	Las habilidades se complementarán con otras habilidades pertenecientes al PFC de Verónica Raspeño Gutiérrez [21].		
<b>Necesidad</b>	Deseable	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 34: REQ-HAB-03

<b>ID</b>	<b>REQ- HAB-04</b>		
<b>Nombre</b>	Habilidad robot portero		
<b>Descripción</b>	Un grupo de habilidades se centrará en emular el comportamiento del robot portero.		
<b>Necesidad</b>	Deseable	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 35: REQ-HAB-04

<b>ID</b>	<b>REQ- HAB-05</b>		
<b>Nombre</b>	Habilidad robot defensa		
<b>Descripción</b>	Un grupo de habilidades se centrará en emular el comportamiento de los robots defensa.		
<b>Necesidad</b>	Deseable	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 36: REQ-HAB-05

<b>ID</b>	<b>REQ- HAB-06</b>		
<b>Nombre</b>	Habilidad robot atacante		
<b>Descripción</b>	Un grupo de habilidades se centrará en emular el comportamiento de los robots atacantes.		
<b>Necesidad</b>	Deseable	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 37: REQ-HAB-06

<b>ID</b>	REQ- HAB-07		
<b>Nombre</b>	Habilidad defender portería robot portero		
<b>Descripción</b>	El robot con rol de portero defenderá la portería, manteniendo la posición en la línea de gol.		
<b>Necesidad</b>	Deseable	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 38: REQ-HAB-07

<b>ID</b>	REQ- HAB-08		
<b>Nombre</b>	Habilidad defender portería robot defensa		
<b>Descripción</b>	El robot con rol de jugador defenderá la portería, manteniendo la posición en la línea del área.		
<b>Necesidad</b>	Deseable	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 39: REQ-HAB-08

<b>ID</b>	REQ- HAB-09		
<b>Nombre</b>	Habilidad pase		
<b>Descripción</b>	Un robot realiza un pase a otro jugador del mismo equipo.		
<b>Necesidad</b>	Deseable	<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable	<b>Claridad</b>	Alta

Tabla 40: REQ-HAB-09





## ANEXO III: MANUAL DE USUARIO

---

## ANEXO III: Manual de Usuario

### Simulador grSim

El simulador grSim [7] [15] es utilizado para probar toda la implementación de la arquitectura y las habilidades. A continuación se detallarán los requisitos mínimos del sistema y el manual de instalación y de uso.

### Requisitos Mínimos del Simulador grSim

Para poder instalar el simulador grSim [7] [15] se requieren unos requisitos mínimos. GrSim [7] [15] funcionará correctamente en un ordenador moderno Dual Core que utilice un sistema operativo Linux y una buena tarjeta gráfica. Una configuración típica es:

- Dual Core CPU (2.0 Ghz+)
- 1GB of RAM
- 256MB nVidia or ATI graphics card
- Ubuntu Linux 9.10+

### Instalación del Simulador grSim

Para instalar el simulador en un equipo deben seguirse los siguientes pasos:

- Descargar e instalar los siguientes paquetes en el equipo:
  - libqt4-dev
  - libgl1-mesa-dev
  - libglu1-mesa-dev
  - VarTypes
  - ode
  - protobuf
- A continuación deben introducirse los siguientes comandos por consola:
  - cd proto/pb

- protoc \*.proto --cpp\_out=../
- qmake Simulator.pro
- make
- El archivo binario se almacena en grSim/bin.

### Uso del Simulador grSim

Recibir información del simulador grSim [7] [15] es exactamente igual que recibirla del sistema de visión SSL-Vision [12] utilizando Google Protobuf [13].

Para enviar comandos al simulador se hace uso de Google Protobuf [13].

### Arquitectura 3T

En este manual de usuario se detallará la ejecución de la arquitectura 3T [36] para robots de la Robocup [17], tanto para robots físicos reales como para el simulador grSim [7] [8]. Esta arquitectura contiene la implementación de todas las habilidades de la Robocup [17] detalladas anteriormente.

El objetivo de este manual es ayudar al usuario y guiarle en la ejecución de la arquitectura, así como facilitar la labor de inclusión de nuevas funcionalidades y módulos. El manual detalla toda la información importante para el usuario, de la forma más precisa posible, a fin de facilitar el aprendizaje de uso del sistema y permitir una rápida toma de contacto y uso.

### Requisitos mínimos de la arquitectura 3T

Los requisitos mínimos que deben cumplirse para ejecutar la arquitectura 3T son los que se indican a continuación:

- Sistema operativo Windows, Mac OS X o Linux.
- Java Runtime Environment (JRE).
- Dual Core CPU (2.0 Ghz+).
- 1 GB of RAM.

### Ejecución de la arquitectura 3T

Para llevar a cabo la ejecución de las distintas habilidades se han definido varias clases ejecutables, correspondientes a cada una de las habilidades.

Las clases que se han definido para ejecutar las distintas habilidades son:

- **PruebaPortero**: ejecuta la habilidad del portero, en la cual, este busca la portería y una vez situado en la línea de gol, se mantiene controlando la posición de la pelota.
- **PruebaBuscarPelota**: la ejecución de esta clase permite que el robot busque la pelota y se sitúe frente a ella. Para ello realiza el giro óptimo para encontrar la pelota lo más pronto posible.
- **PruebaConducirPelota**: esta clase ejecuta en primer lugar la habilidad de buscar pelota y una vez situado el robot frente a la pelota, este la conduce hasta la portería rival.
- **PruebaDisparo**: esta prueba ejecuta la habilidad de conducir pelota, pero cuando llega a la portería contraria realiza un disparo.
- **PruebaDefensa**: ejecuta la habilidad del defensa, en la cual, este busca la línea del área y una vez situado en esta, se mantiene controlando la posición de la pelota.
- **PruebaPase**: ejecuta la habilidad de buscar pelota y de pase, en la cual el robot se sitúa frente a la pelota, se orienta hacia el robot al que va a realizar el pase y realiza el pase.
- **PruebaEsquivar**: ejecuta la habilidad de esquivar, para ello el robot se mueve esquivando a los robots que se encuentra en su camino.

Para elegir el robot que va a ejecutar la habilidad, debe introducirse en el parámetro args [0] el identificador del robot y en el parámetro args[1] y el número cero para indicar si el equipo es el azul y uno si el equipo es amarillo, de la manera que se indica en la tabla 41.



Equipos	Color	Número asociado
Equipo azul		0
Equipo amarillo		1

Tabla 41: identificación de equipos



## **ANEXO IV: MANUAL DE REFERENCIA**

---

## ANEXO IV: Manual de Referencia

En el manual de referencia se explica toda la funcionalidad del sistema y el conjunto de elementos que componen el mismo.

El sistema es una arquitectura 3T [36] que permite la ejecución de habilidades en robots de la Robocup [4], cumpliendo las reglas de la competición.

### Características del sistema

Las características principales de la arquitectura 3T [36] para el control de robots de la RoboCup [4] son:

- **Modularidad:** la arquitectura tiene una estructura modular que facilita su futura ampliación con nuevas funcionalidades y la inclusión de nuevas habilidades para los robots.
- **Estándares de la RoboCup [5]:** dado que la arquitectura es capaz de recibir e interpretar la información enviada por el sistema SSL-Vision [12], esta sigue el estándar de la competición y puede utilizarse para el control de robots físico a través del sistema SSL-Vision [12].
- **Posibilidad de incluir planes:** puesto que se utiliza una arquitectura híbrida, es posible crear planes para que los robots ejecuten las habilidades determinadas en función de unas condiciones y así conseguir los objetivos fijados.
- **Fiabilidad:** dado que el sistema actualiza las posiciones de los robots continuamente, la información que maneja la arquitectura sobre los robots es fiable y gracias a esto se obtienen movimientos precisos.

## Usuarios del sistema

El sistema solo dispone de un tipo de usuario, el cual es capaz de ejecutar las habilidades en el simulador grSim [7] [15] o en un robot físico, utilizando el sistema SSL-Vision [12].

Este usuario debe ser una persona que tenga conocimientos sobre el funcionamiento básico de la RoboCup SSL [5] para conocer cuál es el funcionamiento de estos robots para llevar a cabo las habilidades.

## Funcionalidad del sistema

La funcionalidad del sistema se basa en el control de robots a través de la arquitectura 3T [36]. En este proyecto se han implementado nuevas habilidades para que los robots pertenecientes a la categoría SSL [5] de la RoboCup [4] sean capaces de ejecutarlas. Hay que destacar que el sistema está diseñado para controlar un robot, por lo tanto, para controlar más de un robot de forma simultánea deberá realizarse en otro ordenador o creando otro proceso de ejecución en el mismo.

Como funcionalidades cabe destacar:

- Ejecución de habilidades simples para los robots de la RoboCup [4]. Las habilidades que se han desarrollado en este proyecto y en el proyecto de Alejandro Caparrós Hernando [32] son:
  - Giro óptimo del robot.
  - Buscar portería.
  - Defender portería.
  - Buscar pelota.
  - Conducir pelota.
  - Pase.





- Esquivar obstáculos.
- Disparar a portería.

Ejecución de planes que contienen distintas habilidades y mediante los cuales se obtiene una meta definida previamente.



## **ANEXO V: GESTIÓN DE PROYECTO**

---

## ANEXO V: Gestión de Proyecto

En este anexo se detalla la descripción de la gestión del proyecto, detallando las fases de las que este se compone, las tareas en las que se divide y un diagrama de Gantt que aporta una representación gráfica.

### Fases

A continuación, se van a detallar todas las fases de las que se compone el proyecto, así como las tareas que componen estas fases.

- **Documentación:** fase en la cual se busca información relevante para el desarrollo del proyecto, se extrae la más importante y se sintetiza con el objetivo de facilitar el desarrollo del proyecto. Esta fase se compone de las distintas tareas:
  - **Investigación simuladores:** esta tarea consiste en la búsqueda de un simulador válido para simular robots de la liga SSL [5]. Para ello se encontraron varias opciones y se estudiaron las características de cada uno para escoger el que más se adapta a las necesidades del proyecto.
  - **Lectura PFCs:** se leen distintos proyectos relacionados con la temática para obtener mayor información sobre los recursos de los que se dispone para el desarrollo del proyecto.
- **Análisis:** fase en la que se extraen los requisitos que deberá tener el proyecto, realizando un estudio y estimando recursos para el desarrollo del proyecto.
  - **Estudio del simulador:** tarea en la cual se analizan todas las características de las que dispone el simulador y se estudia la forma de realizar las conexiones.
  - **Estudio de la arquitectura 3T [36]:** se analiza la arquitectura 3T [36] creada en el PFC de Jose Luis Díaz Cebrián [20].

- **Identificación de requisitos:** esta tarea consiste en extraer todos los requisitos del proyecto y clasificarlos de forma que sean fácilmente identificables.
- **Diseño:** fase en la cual se realiza todo el diseño de la arquitectura y la integración con el código ya existente previamente.
  - **Diseño arquitectónico:** se realiza el diseño arquitectónico utilizando un diagrama de componentes que representa la arquitectura del sistema.
  - **Diseño detallado:** se realiza el diseño detallado utilizando un diagrama de clases y de ejecución en los que se refleja la estructura interna del sistema.
- **Desarrollo:** fase en la cual se implementa todo el código necesario para el cumplir con los objetivos del proyecto.
  - **Cliente Java [16] envío:** se desarrolla el cliente Java [16] para enviar pequeños comandos al simulador o al robot físico.
  - **Cliente Java [16] recepción:** se implementa el cliente que recibe los paquetes UDP con la información del terreno de juego.
  - **Integración de arquitectura 3T [36]:** se realiza el proceso de integración de la arquitectura 3T [36] existente y se adapta a las necesidades del proyecto.
  - **Habilidades:** implementación de todas las habilidades básicas desarrolladas.
  - **Modificación simulador grSim [7] [10]:** adaptación de los robots simulados para que cumplan con las características del robot físico disponible.
  - **Aplicación robots físicos:** modificaciones realizadas para adaptar el código a los robots físicos.
- **Pruebas:** fase en la que se prueba el correcto funcionamiento de la arquitectura, así como de todas las habilidades desarrolladas.

- **Pruebas en simulador:** tarea en la cual se prueban todas las habilidades en el simulador.
- **Pruebas en robots físicos:** se realizan las pruebas pertinentes para comprobar que el robot físico funciona correctamente.
- **Memoria:** fase en la cual se desarrolla toda la documentación del proyecto y se explica toda la información referente con el proyecto.
  - **Recopilación de documentación:** se recoge toda la documentación importante para el desarrollo del proyecto.
  - **Realización memoria:** se escribe el documento que explica todo el desarrollo del proyecto.

### Ciclo de Vida

Para la elaboración de este proyecto se ha seguido un ciclo de vida en cascada, el cual se muestra en el siguiente esquema:

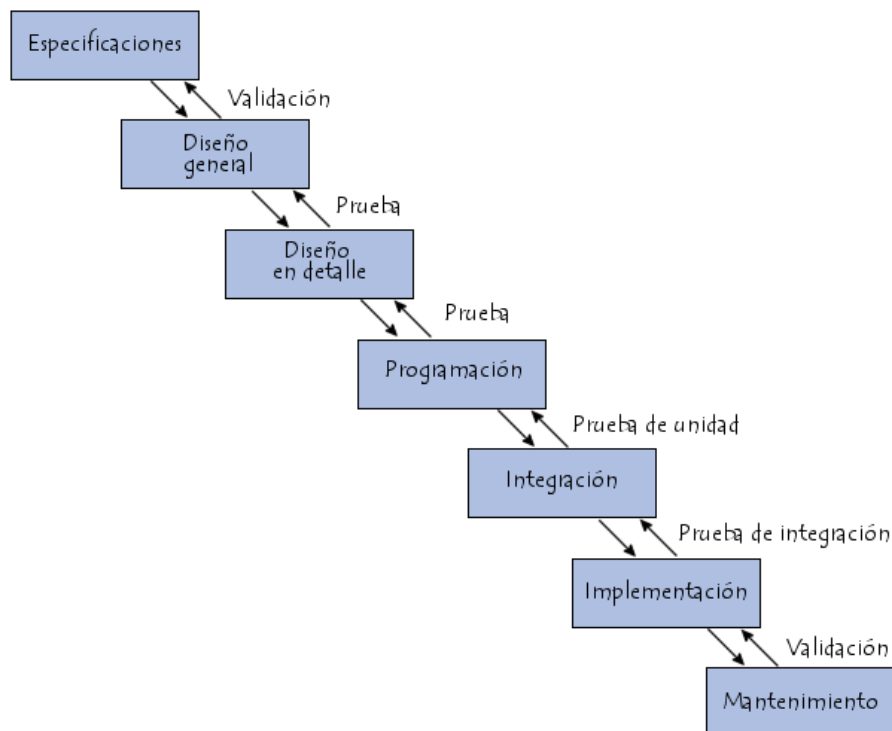


Ilustración 92: ciclo de vida cascada



Este ciclo de vida ordena las etapas de desarrollo del proyecto, lo que produce que ante cualquier error es necesario rediseñar.

### **Diagrama de Gantt**

En el diagrama de Gantt se muestra el progreso seguido durante el desarrollo del proyecto. En este diagrama de Gantt se muestran todas las fases seguidas para el desarrollo del proyecto, así como las fechas en las que se han realizado las siguientes etapas. Este diagrama queda representado en la ilustración 93.

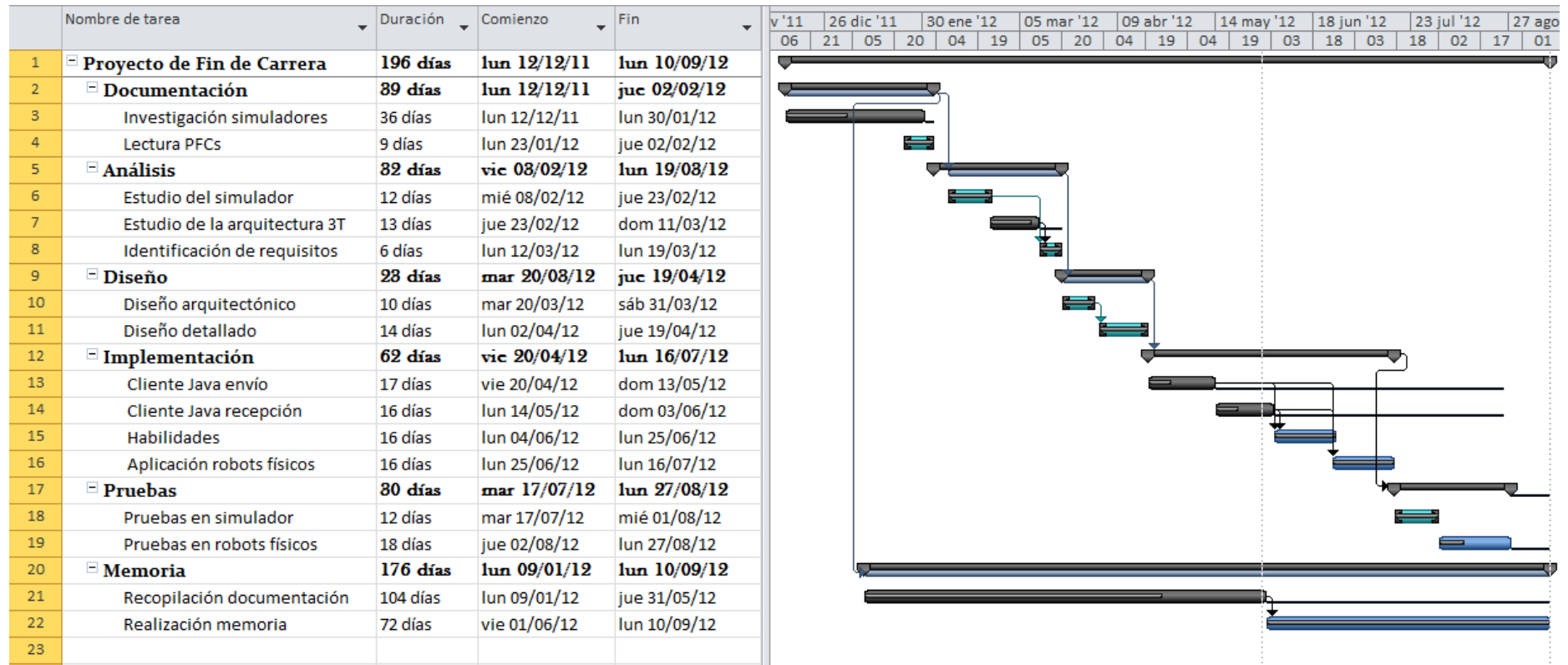


Ilustración 93: diagrama de Gantt



Como puede apreciarse en el diagrama de Gantt, la duración total del proyecto ha sido de 196 días, hay que destacar que este PFC ha sido desarrollado conjuntamente con el PFC de Verónica Raspeño Gutiérrez [21]. Además, hay que destacar que este proyecto ha sido combinado con otras tareas relacionadas con la Universidad.





## **ANEXO VI: REGALEMENTO SSL DE 2011**

---

## ANEXO VI: Reglamento SSL de 2011

A continuación, se explican las reglas de la categoría SSL que es en la que se basa este proyecto.



Ilustración 94: partido perteneciente a la SSL

La liga SSL o F180 como también se la conoce, se basa en la investigación de la cooperación entre sistemas multi agente y el control de robots en entornos dinámicos, gracias a la utilización de un sistema que recoge y distribuye la información del entorno. [5]

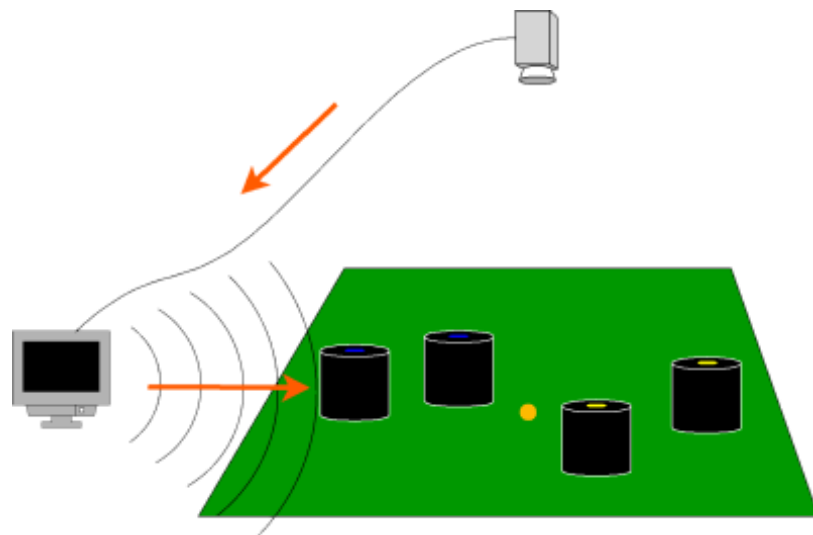


Ilustración 95: sistema de visión

### *Ley 1: El terreno de juego*

#### **Dimensiones**

La superficie de juego debe tener forma rectangular y las dimensiones deben ser 6050 mm de longitud y 4050 mm de ancho. Las dimensiones quedan representadas en la siguiente imagen:

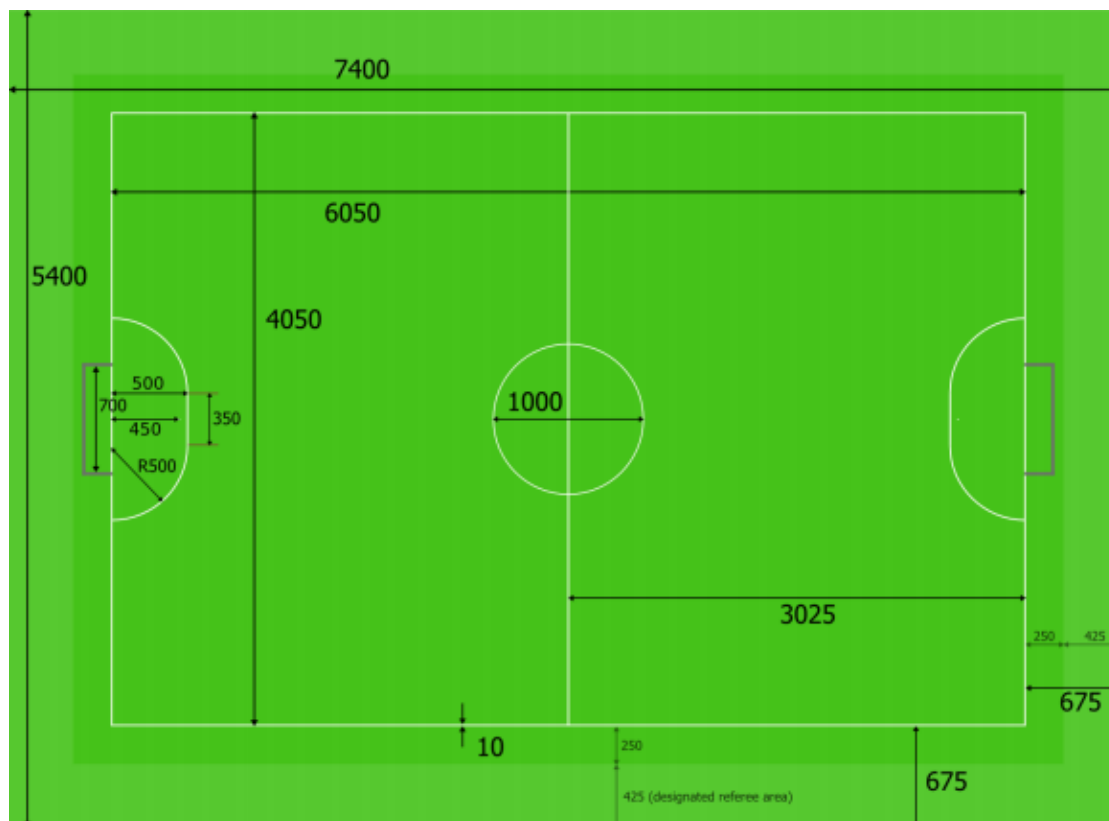


Ilustración 96: dimensiones de la superficie de juego

El terreno de juego se aumentará en 675 mm más allá de las líneas fronterizas por todo el contorno. Los 425 mm del exterior de esta zona de escape se utilizan para el paso del árbitro designado a esta zona. En el borde de la superficie del campo, una pared de 100 mm de altura impedirá que la pelota y los robots salgan fuera.

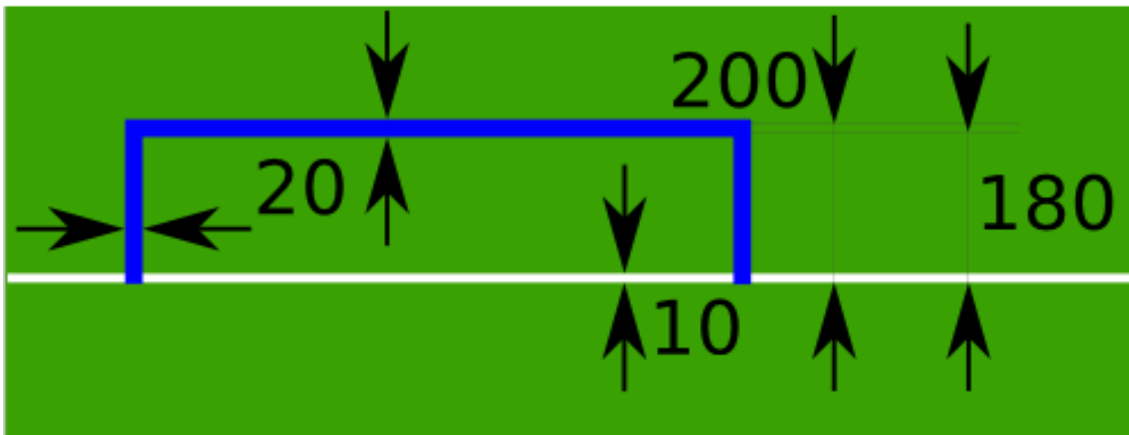
La superficie de juego es una alfombra de fieltro de color verde y está marcada con líneas que indican las áreas, el centro del campo y las bandas del terreno de juego.

### Líneas del Terreno de Juego

El campo tendrá dibujadas las líneas de las áreas de ambos equipos, las bandas y el punto que indica el centro del campo. Todas las líneas tienen un grosor de 10 milímetros de ancho y son blancas. El centro del campo se caracteriza por estar rodeado de un círculo de 1000mm con un diámetro.

### Porterías

Las porterías se sitúan en el centro de cada línea de fondo del campo y las dimensiones se especifican en la siguiente imagen: [5]



**Ilustración 97: dimensiones de la portería**

Constan de dos paredes laterales verticales de 160 mm unidas por una pared trasera de 160 mm. Además tienen una altura de 180 mm.

El material de las porterías debe ser de color blanco y la cara interna tiene que estar recubierta con un material absorbente de energía, para ayudar a controlar el impacto de la pelota y disminuir la velocidad de las desviaciones.

### Áreas

Las áreas del campo se sitúan en cada extremo del terreno, con unas dimensiones de dos cuartos de círculo con un radio de 500 mm. que se encuentran a 450 mm desde el punto medio entre los postes y equidistante a ellos. La marca es un círculo de 10 mm de diámetro de pintura blanca.

### Sistema de Visión

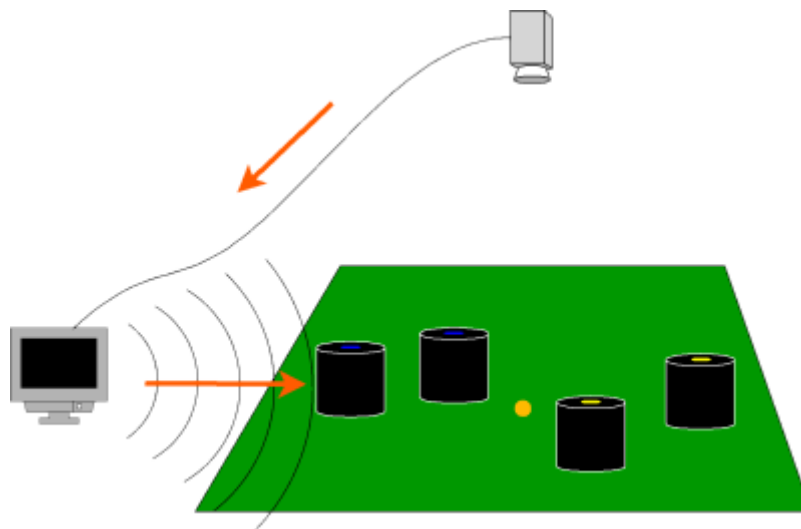


Ilustración 98: sistema de visión

El sistema de visión proporciona la información necesaria sobre el estado del partido a cada uno de los equipos. El sistema de visión se compone de cámaras que envían la información a un servidor al que se conectan los equipos para obtener la información del estado del partido. El software del sistema de visión es común para todos los equipos y se llama SSL-Vision [12] y los equipos deben comprobar que sus equipos son compatibles con el sistema de visión.

La información que envía el software SSL-Vision [12] consiste en paquetes UDP con las posiciones de los robots de ambos equipos y la posición de la pelota. La comunicación con los robots debe realizarse de forma inalámbrica mediante comunicación wifi. [5]

Las cámaras recomendadas para utilizar el sistema de visión son la AVT Stingray [33] y la AVT Marlin. En la siguiente tabla se muestran las características de las cámaras:

Cámara	Resolución	Frecuencia	Codificación de la imagen
AVT Stingray [33]	780x580	60 Hz	YUV422
AVT Marlin	780x580	35 Hz	YUV422

Tabla 42: cámaras del SSL-Vision [12]



Ilustración 99: cámara Stingray

### Decisiones del Comité Técnico de la Competición:

- **Decisión 1:** el comité organizador del evento debe proporcionar luz difusa de condiciones uniformes, de aproximadamente 500 LUX como mínimo. No se utilizará un equipo de iluminación especial para proporcionar estas condiciones. El brillo no está garantizado ni se espera que esté completamente uniforme a través de la superficie del campo. Se espera que los equipos sean autosuficientes para hacer frente a las variaciones que se produzcan cuando se utiliza la iluminación ambiente. El comité organizador dará a conocer detalles de la iluminación de acuerdo a la competición tan pronto como sea posible.
- **Decisión 2:** ningún tipo de publicidad comercial, ya sea real o virtual, está permitido en el terreno de juego y el equipo de campo (incluidas las redes y las áreas que delimitan) desde el momento en que los equipos entran en el terreno de juego hasta el descanso y desde este hasta el momento en que vuelven a entrar en el terreno de juego hasta el final del partido. En particular, ningún material de publicidad de cualquier tipo puede aparecer dentro de los objetos o las paredes. Los equipos ajenos (cámaras, micrófonos, etc,) también se ajustarán a estas normas.

- **Decisión 3:** el color específico y la textura de la superficie no se especifican y puede variar de una competición a otra (como los campos de fútbol reales pueden variar). La superficie por debajo de la alfombra estará nivelada y dura. Entre las superficies autorizadas se incluyen: cemento, linóleo, pisos de madera, madera contrachapada, mesas de ping-pong y tableros de partículas. Moqueta o superficies acolchadas no están permitidas. Se pondrá todo el empeño en asegurar que la superficie sea plana, sin embargo, corresponde a los equipos individuales el diseño de sus robots para hacer frente a la ligera curvatura de la superficie.

### ***Ley 2: Pelota***

La pelota debe ser una pelota de golf estándar, de color naranja de aproximadamente 46 gramos de masa y 43 mm de diámetro. Si la pelota tiene algún defecto el partido se detiene, y se reemplaza la pelota por una en buen estado. Solo podrá sustituirse la pelota con la autorización del árbitro.[5]

### ***Ley 3: Número de Robots***

Los partidos son disputados por dos equipos de 5 robots cada uno, uno de los cuales puede ser el portero.

Cada uno de los robots es identificado de forma sencilla para que tanto el árbitro como los integrantes del equipo sean capaces de identificar los distintos robots.

Los partidos no pueden comenzar hasta que los equipos tengan al menos un robot disponible.

Para identificar los robots se utilizan unos patrones de colores que indican que robot es cada uno. A continuación se muestra una imagen con el significado de los patrones que se sitúa en la parte superior de los robots:



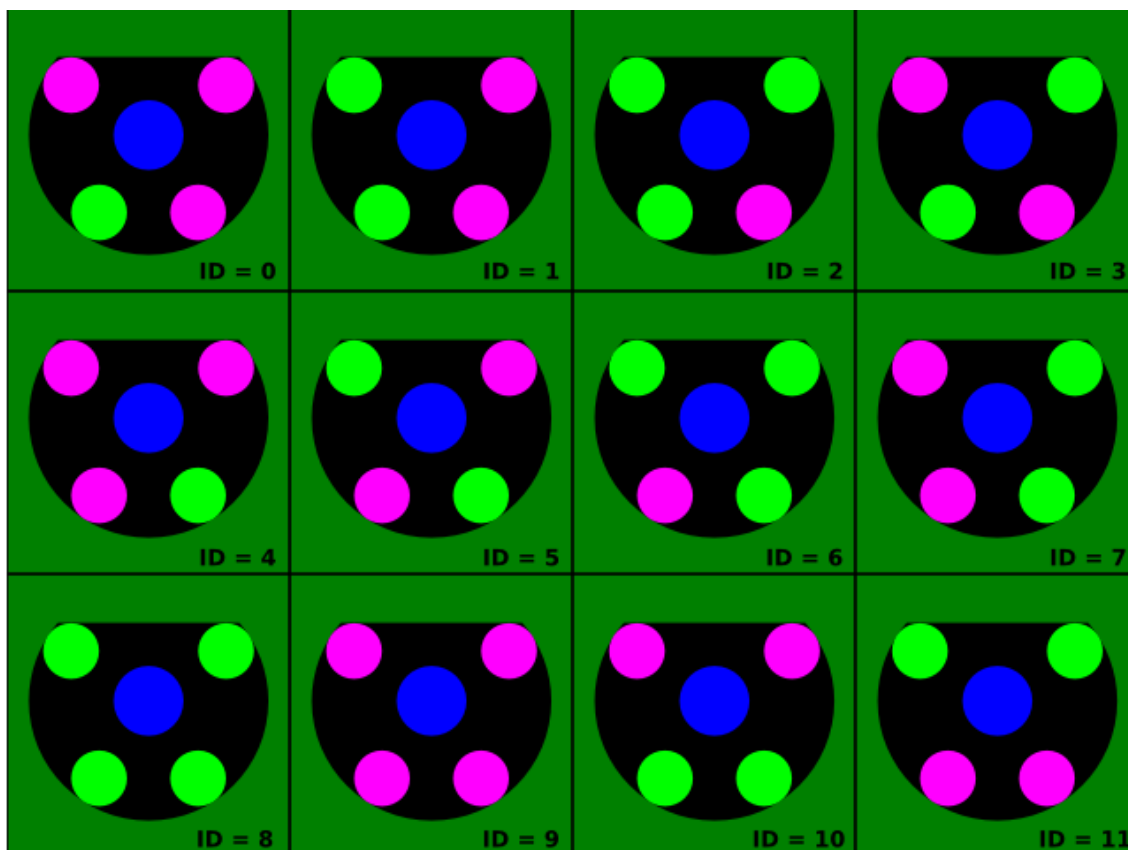


Ilustración 100: identificación de los robots

El color del círculo central indica el color del equipo al que pertenecen los robots que puede ser azul o amarillo. [5]

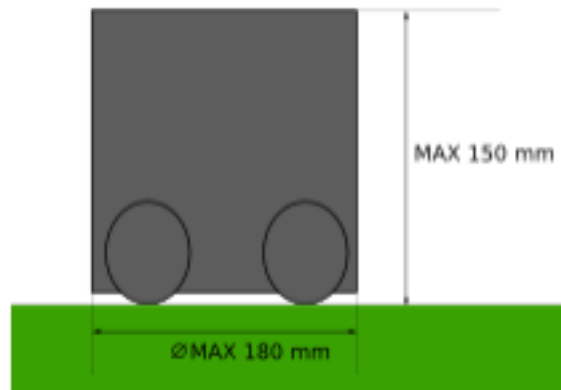
#### Decisiones del Comité Técnico de la Competición:

- **Decisión 1:** cada equipo debe tener un único controlador de robot encargado de realizar el intercambio cuando sea necesario. No hay otros miembros del equipo que puedan invadir el área que rodean el campo. El movimiento de los robots por el controlador no está permitido.

#### *Ley 4: El Equipo de Robótica*

#### Dimensiones de los Robots

Los robots deben tener como máximo 180 mm de diámetro y 150 mm de altura o menos. En la siguiente imagen se muestran las dimensiones de forma gráfica: [5]



**Ilustración 101: dimensiones de los robots**

La parte superior debe ajustarse al tamaño y forma del patrón estándar.

### **Autonomía de los Robots**

Los robots deben ser totalmente autónomos y no se permite a los operarios del equipo controlar remotamente a los robots durante el transcurso del juego. [5]

### **Regates**

El sistema de dribbling son dispositivos que permiten mantener el control de la pelota a los robots. Estos dispositivos ejercen fuerza de forma perpendicular a la superficie del terreno de juego y de esta forma se mantiene el control de la pelota.

En la siguiente imagen se muestra la forma que tiene el dribbler: [5]

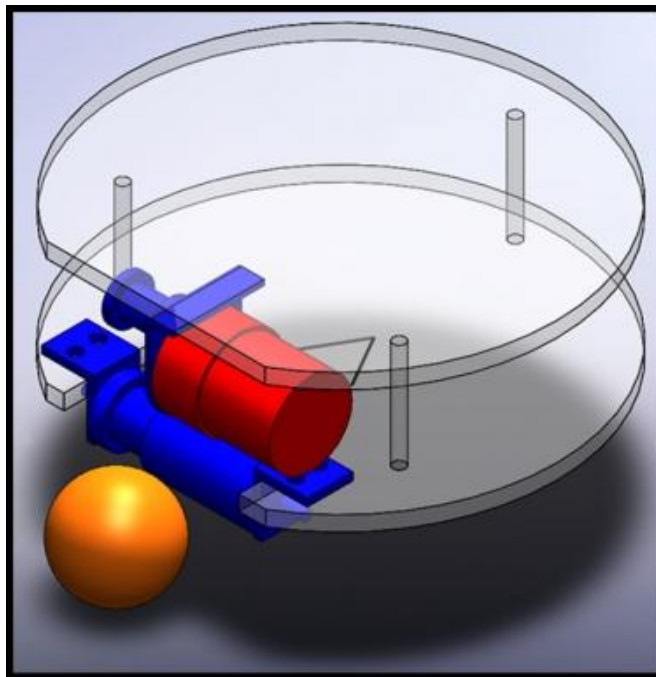


Ilustración 102: sistema dribbler

### **Infracciones y Sanciones**

Si se produce cualquier infracción durante el partido:

- Debe detenerse el juego.
- El árbitro ordena que el robot que no cumpla con la normativa abandone el campo de juego para corregir el defecto.
- El robot abandona el campo cuando se para el juego.
- Los robots que han abandonado el terreno de juego no pueden reincorporarse hasta que el árbitro de la orden.
- El árbitro comprueba que el robot cumple con las reglas antes de permitir la vuelta al campo de juego.
- Solo se permite la entrada de un robot al campo cuando el juego se ha detenido.
- Si el robot entra al terreno de juego sin el permiso del árbitro, este recibirá una tarjeta amarilla. [5]

### **Decisiones del Comité Técnico de la Competición:**

- **Decisión 1:** los participantes que utilizan las comunicaciones inalámbricas notificarán al comité organizador local el método de comunicación inalámbrica, potencia y frecuencia. El comité organizador local será notificado de cualquier cambio después de la inscripción tan pronto como sea posible. Con el fin de evitar interferencias, un equipo debe ser capaz de seleccionar entre dos frecuencias portadoras antes del partido. El tipo de comunicación inalámbrica se ajustará a las normas legales del país donde se celebre la competición. El cumplimiento de las leyes locales es responsabilidad de los equipos que compiten, no de la Federación RoboCup [4]. El tipo de comunicación inalámbrica puede también ser restringido por el comité organizador local. El comité de organización local dará a conocer cualquier restricción a la comunidad lo antes posible.
- **Decisión 2:** están permitidos los dispositivos de golpeo y disparo.
- **Decisión 3:** están prohibidas las puntas de metal y el material velcro, con el propósito de la locomoción.
- **Decisión 4:** está prohibida la comunicación inalámbrica por Bluetooth.
- **Decisión 5:** los colores oficiales serán proporcionados por el comité organizador. Los equipos deben usar los colores oficiales a menos que ambos equipos no estén de acuerdo.
- **Decisión 6:** adhesivos, como pegamento o cinta, no pueden ser utilizados con fines de control del balón o para construir dribladores (sistemas de regateo). El uso de dispositivos que utilizan, por ejemplo, un adhesivo para adherir la pelota a un robot, se consideran una violación de la Regla 12, Decisión 4, por "la eliminación de todos los grados de libertad de la pelota". Además, el uso de adhesivos para cualquier propósito en el robot que provoque residuos sobre el

balón o el campo se considera como daño y son sancionados siguiendo la Ley 12.

- **Decisión 7:** antes del primer partido de la competición se realiza un chequeo de las normas. Si se considera que algún componente de cualquier equipo infringe una norma, el robot debe modificarse para ser compatible, y permitir su participación en los partidos.

### ***Ley 5: Árbitro***

Los partidos son controlados por un árbitro que trata de hacer que se cumplan todas las leyes del juego.

El árbitro tiene poder para:

- Hace cumplir las Leyes del Juego.
- Controla el partido en colaboración con los árbitros asistentes.
- Se asegura de que cualquier pelota utilizada cumpla los requisitos de la Ley 2.
- Asegura que el equipo de robótica cumple con los requisitos de la Ley 4.
- Informa a los árbitros asistentes de cuándo comienzan y terminan los períodos de tiempo, de conformidad con la Ley 7.
- Detener, suspender o terminar el partido ante cualquier infracción producida en el juego.
- Parar el partido si es probable que algún robot cause daños graves a otros robots, a seres humanos o a sí mismo.
- Situar la pelota en una posición neutral si se queda bloqueada en el transcurso del juego.
- Aplicar la ley de la ventaja, si el equipo contra el que se ha cometido una falta se beneficia.
- Castigar a un robot con la pena máxima, cuando este ha cometido más de una falta en el mismo tiempo.

- Tomar medidas ante malos comportamientos por parte de los responsables del equipo.
- Asegurar que ninguna persona no autorizada entra en el campo.
- Tener en cuenta las sugerencias de los árbitros asistentes.
- Reanudar el partido tras haber sido detenido.
- Proporcionar al comité técnico un resumen del partido, recogiendo información sobre cualquier acción disciplinaria tomada contra los equipos.

Además, el árbitro envía señales durante el juego para indicar los eventos que se producen como el comienzo del partido o el final. [5]

### **Decisiones del árbitro**

El árbitro solo puede cambiar una decisión al darse cuenta de que esta es incorrecta o debida al consejo del árbitro asistente, siempre que siga el juego parado.

### **Equipo de señalización del árbitro**

El dispositivo necesario se suministra para convertir las señales del árbitro en serie y Ethernet. Las señales de comunicación se transmiten a ambos equipos. Los equipos serán operados por el árbitro asistente. Los detalles del equipamiento serán suministrados por la organización local antes de la competición.

### **Señales enviadas por el árbitro**

Durante el juego, el árbitro dará las señales de inicio y fin del partido. El árbitro asistente enviará señales que reflejarán las decisiones del árbitro a cada uno de los equipos. No está permitida ninguna interpretación de las señales del árbitro por los operadores humanos.

La señal del silbato indica que el árbitro ha parado el juego y que todos los robots deben separarse 500 milímetros de la pelota para que el árbitro pueda colocar la pelota para reiniciar el sistema. Todos los robots tienen la obligación de situarse a 500 milímetros de la bola mientras esta se mueve a la posición de reiniciar.

Cuando uno de los equipos marca un gol (véase Ley 10), o surge una precaución o se produce una salida de la pelota del campo de juego (véase Ley 12), una señal de información es enviada a los equipos para indicar la decisión del árbitro.

La señal de reinicio indicará el tipo de reinicio.

Los robots deben moverse a posiciones legales a la recepción de esta señal. Para reiniciar otras acciones que no sean un saque inicial (véase Ley 8) o un penalti (véase Ley 14), el robot que saque puede patear el balón cuando esté listo, sin esperar más señales del árbitro.

Para un saque inicial (véase Ley 8), o un penalti (véase Ley 14), una señal de arranque será enviada para indicar que el robot que lance puede proceder. Esta señal será distinta a otros tipos de señales de reinicio del juego.

Se enviarán señales que indiquen los períodos de tiempo de espera y el tiempo perdido también se enviará cuando sea necesario.

Se considerará que el árbitro ha dado una señal cuando el árbitro asistente envíe esta señal a los equipos mediante las comunicaciones.

#### **Decisiones del Comité Técnico de la Competición:**

- **Decisión 1:** el árbitro (o el árbitro asistente, según el caso) no es el responsable de:
  - Cualquier tipo de perjuicio sufrido por un componente del equipo o un espectador
  - Cualquier daño a la propiedad de cualquier tipo.
  - Cualquier otra pérdida sufrida por cualquier persona, club, empresa, asociación u otro organismo, que es debido o que puede ser debido a cualquier decisión que se tome en virtud de los términos de las leyes del juego o en el caso de los procedimientos normales requeridos para conservar, reproducir y controlar un partido. Esto puede incluir:

- La decisión de que la condición del terreno de juego o sus alrededores son tales como para permitir o no que un partido que tenga lugar.
  - La decisión de abandonar un partido por cualquier razón.
  - Una decisión en cuanto a la condición de los accesorios o equipos utilizados durante un partido como el campo y la pelota.
  - La decisión de detener o no detener un partido debido a la interferencia del espectador o cualquier problema en el área de los espectadores.
  - La decisión de detener o no detener el juego para permitir que un robot dañado pueda ser eliminado del campo de juego para su reparación.
  - La decisión de solicitar o insistir en que un robot dañado se retire del terreno de juego para su reparación.
  - La decisión de permitir o no permitir ciertos colores.
  - La decisión (en la medida en que esta puede ser su responsabilidad) para permitir o no permitir a las personas (incluyendo el equipo o funcionarios del estadio, oficiales de seguridad, fotógrafos u otros medios, representantes, etc.) estar presentes en las inmediaciones del campo de juego
  - Cualquier otra decisión que pueda tomar de acuerdo con las Reglas de Juego o de conformidad con sus obligaciones bajo los términos de la Federación RoboCup [5] o las normas o regulaciones bajo las cuales se juega el partido.
- **Decisión 2:** los hechos relacionados con el partido serán incluidos tanto si se marca un gol o no, así como el resultado del encuentro.
  - **Decisión 3:** el árbitro usará un bastón negro o algún otro dispositivo o herramienta para el reposicionamiento de la pelota, reduciendo el riesgo de interferencias con los sistemas de visión o SSL-Vision [12].
  - **Decisión 4:** el árbitro podrá ser asistido por aplicaciones autónomas de arbitraje proporcionadas por uno o ambos de los equipos que compiten, si ambos equipos están de acuerdo. Asimismo, el árbitro podrá ser asistido por una aplicación autónoma o semi-autónoma proporcionada por un equipo que no participe en el



partido, según el criterio del árbitro; teniendo en cuenta que la aplicación deberá ser operada y monitorizada de manera neutral.

- **Decisión 5:** la región externa de la superficie del campo que es más allá de 250 milímetros de distancia de la línea divisoria y es utilizada como zona de paseo designado por el árbitro y/o el árbitro asistente durante el juego. Los equipos deben controlar a sus robots para permanecer fuera de esta zona para no interferir con los árbitros. Los árbitros no son responsables de cualquier obstrucción a los robots o sistemas de visión dentro del área. Sin embargo, los árbitros deberán llevar ropa y zapatos que no contengan ningún color reservado para la bola o los marcadores de los robots.

### ***Ley 6: El Árbitro Asistente***

#### **Deberes del árbitro asistente**

Las funciones del árbitro asistente, respetando las decisiones del árbitro, son las siguientes:

- Actuar como medidor de tiempo y llevar un registro del juego.
- Operar el equipo de comunicaciones para transmitir las señales del árbitro sobre los enlaces de comunicaciones.
- Supervisar a los operadores de robots para evitar que señales ilegales sean enviadas a los robots.
- Indicar cuándo se solicita un intercambio.
- Indicar cuándo una mala conducta o cualquier otro incidente se ha producido fuera de la vista del árbitro.
- Indicar cuándo se comete una infracción si los asistentes se acercan más a la acción que el árbitro (esto incluye, en determinadas circunstancias, las faltas cometidas en la defensa del área).
- Indicar si en los penaltis, el guardameta se ha movido hacia delante antes de que el balón ha sido golpeado y si el balón ha cruzado la línea de meta.

### **Asistencia del árbitro asistente**

Los árbitros asistentes ayudan al árbitro a controlar el partido, de acuerdo a las Leyes del Juego. En el caso de una interferencia, el árbitro liberará de sus obligaciones al árbitro asistente.

### **Decisiones del Comité Técnica de la Competición:**

- **Decisión 1:** se utilizará un segundo árbitro asistente si es posible. El segundo árbitro asistente se encarga de ayudar al árbitro durante el juego y a vigilar el cumplimiento de las Leyes del Juego.

### ***Ley 7: La Duración del Partido***

#### **Períodos del juego del partido**

El partido tiene dos periodos iguales de 10 minutos, salvo mutuo acuerdo del árbitro y los dos equipos. Cualquier acuerdo para alterar los períodos de juego (por ejemplo, para reducir cada mitad a 7 minutos a causa de un horario limitado) debe hacerse antes el inicio del juego y deben cumplir con las normas de competencia.

#### **Intermedio del partido**

Los equipos tienen derecho a un intermedio a mitad del tiempo medio de un intervalo que no deberá exceder de 5 minutos. Las normas de competencia deben indicar la duración del intermedio o descanso. La duración del descanso puede ser modificado únicamente con el consentimiento de ambos equipos y el árbitro.

#### **Tiempos de espera durante el partido**

A cada equipo se le otorgan cuatro tiempos de espera al comienzo del partido. Se permite un total de 5 minutos para todos los tiempos de espera. Por ejemplo, un equipo puede pedir tres tiempos de espera de un minuto de duración y, posteriormente, sólo tienen un tiempo de espera de hasta dos minutos de duración.

Los tiempos de espera sólo pueden ser consumidos durante una interrupción del juego. El tiempo es controlado y registrado por el árbitro asistente.

### **Compensación por el tiempo perdido en el transcurso del partido**

Se tiene en cuenta cualquier período de tiempo perdido debido a la evaluación de los daños en los robots, la eliminación de los robots dañados en el terreno de juego y cualquier otra causa que suponga la pérdida de tiempo. La compensación por el tiempo perdido es a discreción del árbitro.

### **Tiempo adicional**

Serán aplicadas las normas de competencia, podrán prever dos tiempos suplementarios iguales, según las condiciones de la Ley 8.

### **Abandono del partido**

Un partido abandonado se repite mientras las normas no dispongan de lo contrario.

### **Decisiones del Comité Técnica de la Competición**

- **Decisión 1:** el comité hará todo lo posible para proporcionar acceso a los equipos de la competición al menos dos horas antes del inicio de la competición. También se esforzará por permitir al menos una hora de tiempo de configuración antes de cada partido. Los participantes deben ser conscientes, sin embargo, que puede ocurrir que este tiempo no se pueda proporcionar.

### ***Ley 8: El Inicio y la Reanudación de Juego***

#### **Preliminares**

Si ambos equipos tienen una frecuencia preferida común para las comunicaciones inalámbricas, el comité organizador local asignará la frecuencia para la primera mitad

del partido. Si ambos equipos tienen un color preferido común, el comité organizador local asignará el color de la primera la mitad del partido.

La forma de asignar esta información es a través del lanzamiento de una moneda y el equipo que gane el sorteo decidirá qué portería atacará en la primera la mitad del partido, cambiando la portería en el segundo tiempo. El otro equipo realiza el saque para comenzar el partido. En la segunda parte, se procederá al contrario, el equipo que gane el sorteo tiene el saque inicial para comenzar la segunda mitad del partido.

Sin embargo, si los equipos no están de acuerdo para cambiar campos, pueden permanecer en los mismos que el primer tiempo con el consentimiento del árbitro.

### **Saque desde el centro del campo**

Un saque desde el centro del campo es una forma de iniciar o reiniciar el juego en las siguientes condiciones:

- En el inicio del partido.
- Después de la anotación de un gol.
- Al comienzo de la segunda parte del partido.
- Al comienzo de cada período de tiempo adicional.

El procedimiento a seguir es el siguiente:

- Todos los robots se encuentran en su propia mitad del campo.
- Los oponentes del equipo que realiza el saque del partido están por lo menos a 500mm de la bola hasta que el balón está en el juego.
- El balón está parado en el centro del campo hasta que el árbitro da la señal de saque.
- El árbitro da la señal de saque.
- La pelota está en juego cuando es pateada y se mueve hacia delante.

- El lanzador no podrá tocar el balón por segunda vez hasta que haya tocado a otro robot.

### **Situación de la pelota:**

La colocación de la pelota es una forma de reanudar el partido tras una parada temporal que haya sido necesaria, mientras la pelota estaba en juego, por alguna razón no mencionada en las normas.

### **Circunstancias especiales:**

Un tiro libre concedido al equipo defensor dentro de su propia área de defensa se realiza desde la posición de tiro cercana a donde se produjo la infracción, elegida por el propio equipo.

Un tiro libre concedido al equipo atacante en el área de defensa de sus oponentes es lazado desde la posición legal predefinida de tiro libre más cercana al lugar donde se produjo la infracción.

Una pelota que esté en condiciones de reiniciar el partido después de que la jugada ha sido detenida temporalmente en el interior de la zona defensiva se coloca sobre la posición legal de tiro libre más cercana a donde se encontraba el balón cuando la jugada se detuvo.

## ***Ley 9: El Balón en Juego y Parado***

### **Balón parado**

La pelota está parada cuando se han cruzado los límites del campo por suelo o aire o el juego ha sido detenido por el árbitro.

Si la pelota sale fuera del terreno de juego, los robots deben seguir estando a 500 milímetros de la bola mientras esta se coloca, hasta que la señal de reinicio sea dada por el árbitro.

### **Balón en juego**

La pelota está en juego en cualquier otro momento.

### **Infracciones/Sanciones:**

Si en el momento en que la pelota entra en el terreno de juego un miembro del equipo que saca está a una distancia inferior de 200 milímetros de la zona de defensa del oponente, se concede un tiro libre indirecto al equipo contrario. Este tiro se lanzará desde la ubicación en la que se encontraba la pelota cuando se produjo la infracción (véase Ley 13).

Si después de que la pelota entra en el terreno de juego el robot que tira toca el balón por segunda vez antes de que lo haya tocado otro robot, se concede un tiro libre indirecto al equipo contrario. Este tiro se lanzará desde la ubicación en la que se encontraba la pelota cuando se produjo la infracción (véase Ley 13).

Si después de que la pelota entra en el terreno de juego el robot que tira sostiene el balón antes de que lo haya tocado otro robot, se concede un tiro libre indirecto al equipo contrario. Este tiro se lanzará desde la ubicación en la que se encontraba la pelota cuando se produjo la infracción (véase Ley 13).

Si después de darse una señal para reiniciar el juego el balón no entra en el juego en menos de 10 segundos, se procederá a detener el juego por una señal del árbitro, los robots se moverán a 500 milímetros de la pelota y se indicará un saque neutral.

### **Decisiones del Comité Técnico de la Competición:**

- **Decisión 1:** para todos los reinicios en que las leyes establecen que la pelota está en juego bien sea golpeándola o regateando, los robots deben claramente hacer lo posible para que esta se mueva. Se entiende que la pelota puede permanecer en contacto con el robot o ser golpeada por el robot varias veces a corta distancia, pero bajo ninguna circunstancia el robot mantendrá el contacto o se mantendrá tocando la pelota después de haber recorrido una distancia de

50mm, a menos que el balón haya tocado antes a otro robot. Los robots pueden utilizar los dispositivos de regateo y patada en los lanzamientos de las faltas.

- **Decisión 2:** la zona de exclusión de 200 milímetros de la zona de la defensa del oponente se designa para permitir a la defensa de los equipos tomar una posición defensiva contra un lanzamiento sin la interferencia de los oponentes.

### ***Ley 10: El Método de Puntuación***

#### **Puntuación de gol**

Se marca un gol cuando el conjunto de la pelota pasa por encima de la línea de meta, entre las paredes de meta o por debajo del travesaño, sin que se haya cometido una infracción de las reglas de juego con anterioridad por parte del equipo que anota el gol.

#### **Equipo ganador**

El equipo que anota el mayor número de goles durante un partido es el ganador. Si los dos equipos marcan un número igual de goles, o si no marcó ningún gol, el partido se da como empatado.

#### **Normas de competencia**

Para los partidos que terminan en un empate, las normas de competencia podrán estipular un tiempo suplementario u otro método, siempre aprobado por la Federación RoboCup [4], para determinar el ganador del partido.

### ***Ley 11: Fuera de Juego***

La regla del fuera de juego no se usa en esta competición.

### ***Ley 12: Faltas y Conducta Antideportiva***

#### **Tiro libre directo**

Un tiro libre directo es concedido al equipo adversario si un robot comete cualquiera de las siguientes infracciones:

- Hacer contacto sustancial con un oponente.
- Retener un oponente.
- Sostener el balón deliberadamente (excepto para el guardameta dentro de su ámbito de la defensa propia).
- El segundo robot de la defensa y ocupa el área de la defensa del equipo de tal forma que afecte sustancialmente el juego.

### **Tiro libre indirecto**

Un tiro libre indirecto es concedido al equipo adversario si el guardameta, dentro de su propia área defensiva, comete cualquiera de las siguientes infracciones:

- Transcurren más de quince segundos mientras sostiene la pelota antes de liberarla de su posesión.
- Tiene el balón de nuevo después de haber sido liberado de su posesión y no lo ha tocado otro robot.

Un tiro libre indirecto, además, es concedido si:

- Entra en contacto con el portero y el punto de contacto está en el área de defensa.
- Conduce el balón a una distancia superior a 500mm
- Toca la pelota de tal manera que la parte superior de la bola alcanza una altura superior a 150mm respecto del suelo y el balón entra en la meta de su oponente, salvo que haya sido tocado previamente por un compañero de equipo, o que manteniéndose en contacto con el suelo alcance dicha altura y entre en la meta de su oponente debido a un rebote.



- Patea la pelota de tal manera que supera los 10 m/s de velocidad.
- Comete cualquier otra infracción no citada, por la que se interrumpirá el juego por precaución o para expulsar al robot.

### **Tiro de penalti**

Un tiro de penalti se otorga si alguna de las anteriores cuatro infracciones es cometida por un robot dentro del área de defensa propia, independientemente de la posición de la pelota, siempre y cuando esta esté en juego.

### **Sanciones disciplinarias**

Un equipo será amonestado y recibirá una tarjeta amarilla si uno de os robots:

- Es culpable de conducta antideportiva.
- Es culpable de graves y violentos contactos.
- Infringe persistentemente las Reglas de Juego.
- Retrasa la reanudación del juego.
- No respeta la distancia reglamentaria cuando el juego se reanude con un saque de meta, saque de esquina o tiro libre.
- Modifica o provoca daños en el campo o pelota.
- Deliberadamente entra o se desplaza dentro de la zona de tránsito del árbitro.

Al recibir una tarjeta amarilla, un robot del equipo penalizado debe moverse inmediatamente fuera y ser sacado del campo. Después de dos minutos de juego, tiempo que medirá el árbitro o el asistente, el robot puede entrar de nuevo en el campo en la próxima parada del juego.

### **Expulsión de sancionados**

Un equipo recibe la tarjeta roja si uno de los robots o el equipo es culpable de un comportamiento antideportivo grave. El número de robots en el equipo se reduce en uno después de cada tarjeta roja.

### **Decisiones de la Competición**

- **Decisión 1:** un contacto calificado como importante es aquel suficiente para desalojar al robot de su orientación actual, posición o movimiento, en el caso de que se esté moviendo. Cuando los dos robots se mueven a velocidades similares y la causa de contacto no es evidente, el árbitro permitirá que el juego continúe. Esta Ley está diseñada para proteger a los robots que son lentos o permanecen quietos en el momento del contacto y, por tanto, deben ser detectados por los sistemas de evasión de obstáculos.
- **Decisión 2:** existen precauciones para evitar contactos graves y violentos. Ejemplos claros de infracciones son: el movimiento incontrolado, las malas evasiones de obstáculos, empujar o girar rápidamente mientras se está junto a un oponente. En un escenario típico, el árbitro podrá advertir al equipo, y se espera que se modifique su sistema a fin de reducir la violencia de su juego. Si el árbitro aún no está satisfecho, dictará una amonestación.
- **Decisión 3:** un robot que se coloca en el campo, pero claramente no es capaz de moverse, será sancionado por conducta antideportiva.
- **Decisión 4:** se considera que un robot está reteniendo el balón si tiene el control de la pelota mediante la eliminación de todos sus grados de libertad, normalmente, sujetando la pelota con el cuerpo del robot o rodeándolo para evitar el acceso de otros. El 80% de la superficie de la bola debe ser visible desde arriba, de forma que estará fuera de la parte convexa del robot. Otro robot debe ser capaz de quitar el balón al que lo posee. Esta limitación se aplica también a todos los dispositivos de regateo y golpeo, incluso si tal infracción es momentánea. Se muestra en la siguiente ilustración, siendo la vista aérea del robot:

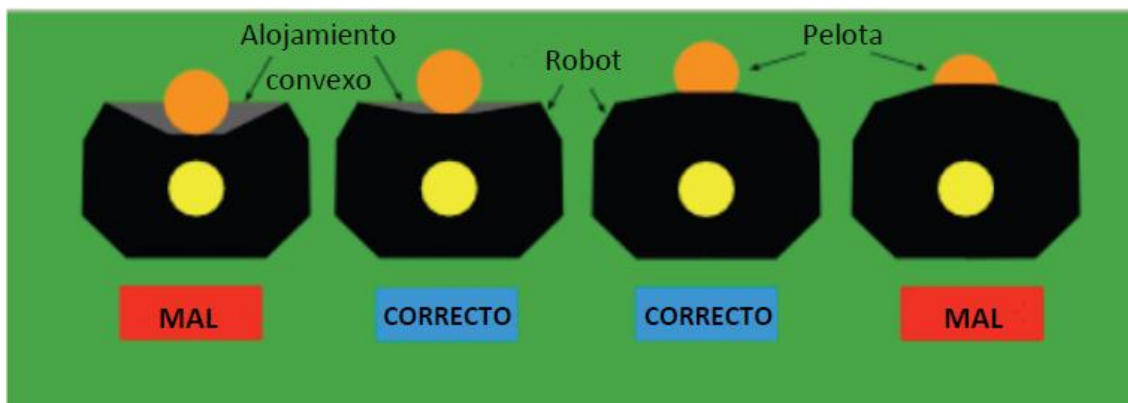


Ilustración 103: posición de la pelota respecto al robot

- **Decisión 5:** un robot comienza el regateo cuando tiene contacto con el balón y se detiene el regateo cuando hay una separación observable entre la pelota y el robot.

La restricción de la distancia en el regateo sirve para evitar que un robot con una mecánica superior pueda tener un indiscutible control de la pelota en el ataque. La restricción de la distancia permite a los atacantes dar y recibir pases, girar con el balón y detenerse con la pelota. Los sistemas de regateo se pueden utilizar para regatear a grandes distancias con el balón, siempre y cuando el robot pierda habitualmente la posesión. El comité técnico espera que la regla de distancia sea auto-forzada, es decir, que los equipos dispongan de un software que la cumpla con antelación y se les pueda pedir una demostración previa a una competición.

Los árbitros podrán seguir señalando faltas y pueden señalar amonestaciones por situaciones de violación sistemática de dicha regla.

- **Decisión 6:** la limitación de velocidad de disparo de la pelota previene que un robot con un disparo mecánicamente superior tenga demasiada ventaja sobre sus oponentes o que se pateo la pelota a una velocidad no apta para los espectadores.

- **Decisión 7:** cita las siguientes condiciones:
  - Si un robot lanza la pelota por encima de un adversario y el balón, posteriormente, entra en la propia portería, después de permanecer por debajo de 15 milímetros de altura todo el tiempo tras haber tocado al robot oponente, el equipo oponente también obtiene un tanto.
  - Si un robot lanza la pelota por encima de un adversario y el balón, posteriormente, entra en propia portería, después de haber estado por encima de 150 milímetros durante algún tiempo (y no habiendo estado en contacto permanente con el suelo), después de tocar al robot oponente, el equipo oponente no puntúa.
- **Decisión 8:** la infracción cometida al entrar deliberadamente en la zona de tránsito del árbitro fue añadido para desalentar a los equipos de la conducción de vehículos por esta zona para obtener ventajas tácticas. En particular, debe prevenir que los equipos exploten el hecho de que otros equipos no podrían tener cobertura de visión del árbitro caminando por dicha área. Se entiende que, en ocasiones, un robot puede entrar en la zona si está fuera de control, o si ha sido empujado a esta área. Estos casos no deben ser considerados infracciones. Sin embargo, la decisión final en cuanto a lo que constituye una violación deliberada del reglamento la decide el árbitro.

### ***Ley 13: Tiros Libres***

Serán directos o indirectos. Tanto en los directos como en los indirectos, la pelota debe ser parada cuando se comete la falta y el robot lanzador no puede tocar el balón por segunda vez hasta que lo haya tocado otro robot.

#### **Tiro libre directo**

Si un tiro libre entra directamente en la meta del oponente, se concede un gol.

Si un tiro libre entra directamente a gol en propia meta, se concede un gol al equipo oponente.

### **Tiro libre indirecto**

Si un tiro libre indirecto entra directamente en la meta del oponente, se concede un saque de puerta.

Si un tiro libre indirecto entra directamente en la propia meta del equipo, se concede un saque de esquina al equipo contrario.

### **Procedimiento par los tiros libres**

Si el tiro libre se concede dentro del área de defensa, el tiro libre se lanza desde un punto a 600 milímetros de la línea de gol y a 100 milímetros desde la línea de contacto más cercana a donde se produjo la infracción.

Si el tiro libre es concedido al equipo atacante a 700 milímetros de la zona de defensa, la pelota se traslada al punto más cercano a 700 milímetros desde el área de defensa.

Si el tiro libre se lanza desde el lugar donde se produjo la infracción.

Todos los robots oponentes se colocarán a una distancia mínima de 500 milímetros de la pelota y esta está en juego cuando es dada y se mueve.

### **Infracciones/Sanciones**

Si cuando se lanza un tiro libre, el oponente más cercano a la bola no se encuentra a la distancia requerida, el tiro se repetirá.

Para otro tipo de infracción relacionada con esta Ley, el tiro se repetirá.

### ***Ley 14: El tiro de Penalti***

Un tiro de penalti se otorga contra un equipo que comete una de las cinco infracciones por las que se concede un tiro libre directo, dentro de su área de defensa y mientras la

bola está en juego. El tiempo que se emplee para un tiro de penalti se añadirá al final de cada mitad o al final de los períodos de tiempo extra.

### **Posición de la bola y los robots**

Para el lanzamiento del penalti, el balón se coloca en el punto de penalti. El robot que lanza el penalti tiene que estar debidamente identificado. Asimismo, el guardameta defensor se mantiene entre los postes de la portería, toca la línea de meta y se orienta hacia la cara externa de la meta, hasta que el balón es lanzado. Se le permite el movimiento antes de que el balón haya sido golpeado, siempre y cuando no se infrinja alguna de las anteriores restricciones. El resto de los robots se encuentran dentro del terreno de juego y detrás de una línea paralela a la línea de gol y a 400 milímetros detrás del punto penalti.

### **El árbitro**

El árbitro dará la señal de tiro de penalti cuando los robots estén colocados correctamente como marca el anterior punto. Es el árbitro quien decide si el tiro penal se ha completado.

### **Procedimiento**

El robot que lanza el penalti, golpea la pelota hacia delante, no pudiendo tocar el balón por segunda vez hasta que haya sido tocado por otro robot. La pelota está en juego cuando es golpeada y se mueve hacia delante.

Cuando un tiro de penalti se lanza durante el curso normal del juego, el tiempo se ha ampliado en la primera mitad o al final del partido para permitir que un lanzamiento de penalti sea lanzado, se concede un gol si entra directamente o si antes de que el balón pase entre los postes y por debajo del travesaño, la pelota toca uno o ambos postes de la portería, el travesaño y/o el portero.

### Infracciones/Sanciones

Si el árbitro da la señal de un tiro de penalti y, antes de que el balón esté en juego, se produce una de las siguientes situaciones:

- El robot que lanza el penalti infringe las Reglas del Juego:
  - El árbitro permitirá que continúe la jugada.
  - Si el balón entra en la meta, se repetirá el tiro.
  - Si el balón no entra en la meta, el lanzamiento no se repetirá.
- El guardameta infringe las Reglas de Juego:
  - El árbitro permitirá que continúe la jugada.
  - Si el balón entra en la meta, se concede un gol.
  - Si el balón no entra en la meta, se repetirá el tiro.
- Un compañero del robot que lanza entra en el área de los 400 milímetros detrás del punto de penalti:
  - El árbitro permitirá que continúe la jugada.
  - Si el balón entra en la meta, se repetirá el tiro.
  - Si el balón no entra en la meta, el lanzamiento no se repetirá.
  - Si el balón rebota en el guardameta, el travesaño o el poste de la meta y es tocado por el presente robot, el árbitro interrumpirá el juego y reanudará el partido con un tiro libre indirecto a favor del equipo que defiende.
- Un compañero del guardameta penetra en la zona de los 400mm detrás del punto de penalti:

- El árbitro permitirá que continúe la jugada.
- Si el balón entra en la meta, se concede un gol.
- Si el balón no entra en la meta, se repetirá el lanzamiento.
- Un robot de ambos equipos, de la defensa y el equipo atacante, infringen las Reglas de Juego:
  - El tiro se repetirá.
  - Tras el cumplimiento de la pena toda infracción que se enumeran en la Ley 9 se tratará en consonancia.
  - El balón es tocado por un agente externo, y se mueve hacia delante:
    - El tiro se repetirá.
    - El balón rebota en el terreno de juego tras tocar al guardameta, el travesaño o los postes, y es entonces tocado por un agente externo:
      - El árbitro detiene el juego.

El juego se reanudará con un toque neutral en el lugar donde la pelota tocó al agente externo (véase Ley 13).

### ***Ley 15: El Saque de Banda***

Un saque de banda es un método de reinicio el juego. Sin embargo, un gol no puede ser marcado directamente desde un saque de banda.

Un saque de banda se concede:

- Cuando la totalidad de la pelota pasa por encima del límite de contacto (línea de banda), ya sea por tierra o por aire.



- Desde el punto, a 100 milímetros, perpendicular a la línea de banda donde la pelota cruzó el límite.
- Al equipo contrario al último robot que toca el balón.

### **Procedimiento**

- El árbitro pone el balón en la posición designada.
- Todos los robots oponentes se distancian por lo menos 500 milímetros de la pelota.
- La pelota está en juego cuando es pateada y se mueve.

### **Infracciones/Sanciones**

Cuando un saque de banda se realiza y un oponente está más cercano a la pelota de la distancia requerida el saque de banda, se repetirá. Para cualquier otra infracción el tiro se repetirá.

### ***Ley 16: El Saque de Puerta***

Un saque de puerta es un método de reinicio el juego. Un gol puede ser anotado directamente por un saque de puerta, pero sólo si entra en la portería contraria.

Un saque de puerta es otorgado cuando la totalidad de la pelota, después de haber sido tocada por un robot del equipo atacante, pasa por encima de la línea de límite de gol, ya sea por tierra o aire y no se concede un tanto de conformidad con la Ley 10.

### **Procedimiento**

- La pelota es golpeada desde el punto a 500 milímetros de la línea de gol y a 100 milímetros de la línea de banda más cercano a donde la pelota pasó por la línea de gol.
- Los opositores estarán a 500 milímetros de la bola hasta que el balón está en juego.

- El lanzador no puede jugar el balón por segunda vez hasta que haya tocado a otro robot.
- La pelota está en juego cuando es golpeada y se mueve.

### **Infracciones/Sanciones**

Toda infracción que se enumeran en la Ley 9 se tratará en consonancia. Para cualquier otra infracción de la presente Ley el tiro se repetirá.

### ***Ley 17: El Saque de Esquina***

Un saque de esquina es un método de reinicio el juego. Un gol puede ser anotado directamente de un saque de esquina, pero solamente contra el equipo contrario.

Un saque de esquina se concede cuando la totalidad de la pelota, después de haber tocado un robot del equipo defensor, pasa por encima de la línea de gol, ya sea por tierra o aire, y no se concede un gol de conformidad con la Ley 10.

### **Procedimiento**

- La pelota es golpeada desde la esquina más cercana, a 100mm en la línea de gol y de la línea de banda.
- Los contrarios siguen estando a 500 milímetros de la bola hasta que el balón está en juego.
- El lanzador no puede jugar el balón por segunda vez hasta que haya tocado a otro robot.
- La pelota está en juego cuando es golpeada y se mueve.

### **Infracciones/Sanciones**

Toda infracción que se enumeran en la Ley 9 se tratará en consonancia. Para cualquier otra infracción de la presente Ley el tiro se repetirá.

## ***Apéndice A: Reglas de Competencia***

Este apéndice describe los procedimientos adicionales necesarios para completar las reglas de la SSL [5].

### **Tiempo extra**

Si el resultado del partido es de empate después del final del segundo período y el partido necesita terminar con un ganador, se jugará un tiempo extra (véase Leyes 7 y 10). Antes de la primera mitad del tiempo extra habrá un intervalo que no deberá exceder de 5 minutos.

### **Periodos de juego durante el tiempo extra**

El tiempo extra dura dos períodos iguales de 5 minutos, salvo mutuo acuerdo entre el árbitro y los dos equipos participantes. Cualquier acuerdo para alterar los períodos de tiempo extra debe hacerse antes del inicio del juego y deben cumplir con las normas de competencia.

### **Descanso**

Los equipos tienen derecho a un descanso en el intervalo entre las dos mitades del tiempo extra. El plazo de tiempo no debe exceder de 2 minutos.

La duración del descanso en dicho intervalo de tiempo puede ser modificado únicamente con el consentimiento de ambos equipos y el árbitro.

### **Tiempos de espera**

Cada equipo tiene asignado dos tiempos de espera en el comienzo del tiempo extra. Se permite un total de 5 minutos para todos los tiempos de espera. El número de tiempos de espera y el tiempo, no utilizados en el juego regular, no se agregan. Los tiempos de espera en el tiempo extra siguen las mismas reglas que en el juego regular (véase la ley 7).

### **Tanda de penaltis**

Si el partido termina en empate después del final de la segunda parte de la prórroga, el resultado final se decidirá en los penaltis.

### **Preparación**

Antes del inicio de los penaltis, habrá un intervalo que no deberá exceder de 2 minutos. Este tiempo se designa para ser utilizado por los equipos en el diálogo con el árbitro y sus asistentes para comprobar que la posición del portero es correcta (en la línea) y que todas las demás normas se cumplen (véase Ley 14). El árbitro determina qué equipo defiende la portería, así como qué equipo tiene que lanzar el primer penalti.

### **Procedimiento**

Durante los tiros desde el punto de penalti, un máximo de 2 robots por equipo estarán en el campo con el fin de evitar interferencias. Los tiros desde el punto penalti se harán alternativamente por parte de ambos equipos hasta que cada equipo haya lanzado cinco disparos. Si se toma una decisión para un equipo, los lanzamientos se interrumpirán por decisión del árbitro. Para todos los lanzamientos, se aplican las normas de la ley 14. Un segundo tiro (por ejemplo, si la pelota rebota en la portería o un poste de la portería o el robot que lanza recupera la pelota) no puntuará; ya que el penalti no será válido si el lazador vuelve a tocar la pelota después del primer disparo. Durante los lanzamientos, desde el punto penalti no habrá tiempos muertos. Los robots pueden ser intercambiados entre los lanzamientos siguiendo las reglas de intercambio de la ley 3. Como el intercambio de los campos entre ambos equipos costaría demasiado tiempo y se forzaría a los equipos a variar sus sistemas, se usarán ambas porterías.

Si después de 10 tiros no hay un vencedor, cada equipo tiene un lanzamiento de penalti en el mismo orden en que lo hicieran anteriormente. Este procedimiento (un penalti por equipo) se continúa hasta que haya un vencedor.

### ***Apéndice B: Expertos en Visión***

Durante las competiciones, los expertos en visión están a cargo del sistema compartido de visión de cada campo. La asignación y el tiempo del período de servicio son designados por los organizadores de la competición. Esto se debe realizar de tal forma que cada sistema de visión compartido tenga asignado, al menos, un experto en visión.

#### **Deberes**

El experto en visión tiene los siguientes deberes:

- Comprobar el hardware del sistema compartido de visión e informar de cualquier problema relacionado al árbitro y a los organizadores locales.
- Hacer el proceso de calibración del SSL-Vision [12] cuando sea necesario o los equipos lo requieran durante los tiempos de configuración.
- Calibrar o realizar el mantenimiento durante el partido del SSL-Vision [12] cuando el árbitro lo requiera.
- Antes de cada partido, comprobar que ambos equipos reciben los paquetes del SSL-Vision [12] correctamente.
- Antes de cada partido, comprobar que ambos equipos utilizan los correctos patrones estandarizados, que la altura de sus robots está calibrada con exactitud y que los datos de localización recibidos son correctos.
- Vigilar el estado del sistema compartido de visión durante el partido y reportar inmediatamente cualquier tipo de problema al árbitro.
- Recibir las quejas de los equipos sobre el sistema de visión compartido durante el partido y, si fuera necesario, preguntar al árbitro para parar



el juego de tal forma que se pueda diagnosticar y solucionar el problema.

Avisar al árbitro si hay alguna queja no solucionable de algún equipo acerca del sistema de visión. En este caso, el árbitro, tiene la autoridad definitiva para fallar en cualquier modo con respecto a sus poderes y deberes (ver Ley 5), incluyendo la habilidad para avisar y/o sancionar a los equipos de mal comportamiento si las exigencias de los equipos son infundadas y continúan obstruyendo el juego (véase Sanciones Disciplinarias en Ley 12).



## **ANEXO VII: REGLAMENTO 2012**

---

## ANEXO VII: Reglamento 2012

En 2012 se producen ciertas modificaciones en el reglamento de la RoboCup Small-Size League. [5]

Los cambios introducidos este año en el reglamento de la RoboCup [4] SSL [5] son:

- El radio del área ha pasado a ser 800 mm en lugar de los 500 mm que eran anteriormente.
- La distancia entre la portería y el borde del área es ahora 750 mm en lugar de los 450 mm que era anteriormente.
- El número de robots por equipo ha pasado a ser 6 en lugar de 5 como era anteriormente.
- Se produce un lanzamiento directo si se produce un vuelco, una rotura o la pérdida de una pieza del robot de manera que de ventaja de forma injusta al equipo.
- Se produce un lanzamiento indirecto, si un robot dispara la pelota con una velocidad superior a 8 m/s, en lugar de 10 m/s.
- En la ley 12 se ha añadido una novena decisión que dice: “Si un robot vuelca o se rompe por algún motivo y este no constituye ningún peligro para otro robot o ser humano, el árbitro permitirá que el juego continúe hasta que se produzca otra introducción en el juego. La decisión final sobre lo que constituye peligro o ventaja injusta recae sobre el árbitro”.
- En el apéndice A se ha incluido el apartado de criterio de clasificación Round-Robin, según este criterio la clasificación de cada equipo en cada grupo será determinada por los siguientes criterios:
  - Mayor número de puntos obtenidos en todos los partidos de grupo.
  - Diferencia de goles en todos los partidos de grupo.
  - Mayor número de goles marcados en los partidos de grupo. [18]

Asimismo, se muestra una tabla resumen que representa lo comentado anteriormente:





	<b>2011</b>	<b>2012</b>
<b>Número de robots por equipo</b>	5	6
<b>Radio del área</b>	500 mm	800 mm
<b>Distancia entre portería y borde</b>	750 mm	450 mm
<b>Lanzamiento indirecto</b>	Si un robot dispara la pelota con una velocidad superior a 10 m/s	Si un robot dispara la pelota con una velocidad superior a 8 m/s

Tabla 43: modificaciones reglas RoboCup [17] 2012 respecto a las reglas RoboCup [17] 2011

