



**UNIVERSIDAD DEL BÍO BÍO**

FACULTAD DE CIENCIAS EMPRESARIALES

DEPARTAMENTO DE AUDITORIA E INFORMATICA

**AGENTES ROBOTICOS MOVILES GUIADOS POR  
VISION ARTIFICIAL EN TIEMPO REAL**

PROYECTO PARA OBTENER EL TITULO DE INGENIERO DE  
EJECUCION EN COMPUTACION E INFORMATICA

Roberto Andres Medina Fuentes

Ricardo Antonio Quezada Muñoz

Profesor Guía: Luis Gajardo

Profesor Informante: Miguel Romero

Chillán 2006



## *Agradecimientos*

Deseo agradecer en primer lugar a Dios, quien fue mi fortaleza en momentos difíciles y la base espiritual, que en gran medida me ayudó a cumplir esta meta. Por otro lado, el agradecimiento infinito a mi madre, que en todos los aspectos de mi vida, me ha dado su confianza, su amor y su fe. A mi hermana y mis abuelos quienes forman mi familia, a los cuales amo. A Mónica y Fernando por creer en mí y apoyarme siempre. Finalmente agradezco a mis amigos y compañeros que siempre me dieron su apoyo.

### ***Roberto Medina***

Agradezco a mi madre por apoyarme en mis ideales y a mi familia en general por confiar en mí.

Agradezco a mi compañero Roberto por su compromiso en el trabajo y por haber aceptado ser parte de este proyecto. A Loretto por su amor, por esperarme y aceptar mi poca disponibilidad. Debo agradecer, además, a todos mis compañeros que confiaron en nosotros. Por último, agradezco a toda persona que haya contribuido al logro del de este proyecto.

### ***Ricardo Quezada***

Agradecemos a nuestro profesor guía Luis Gajardo, quien nos entregó su apoyo y conocimiento, dando vida al proyecto dentro y fuera de la Universidad. A nuestro profesor informante, Miguel Romero, por el esfuerzo que significó su ayuda y su gran aporte al mejoramiento de este trabajo. Al grupo de robótica por proporcionarnos la ayuda logística. A Miguel Pincheira por su constante disposición a ayudarnos. Y finalmente al Sr. Merino por su atención para con nosotros y a nuestros compañeros que en todo momento nos dieron su apoyo.

### ***Los Autores***

*Dedicatoria*

A mi padre, quien me cuida y observa desde algún paradero celestial y en el silencio rebosa de orgullo por esto que he logrado.

***Roberto Medina***

Dedicado en primer lugar a Lucía del Carmen Muñoz Valenzuela, mi madre. A Loretto Macarena Fierro Castillo, mi amor. Y por último, a la familia Muñoz Valenzuela, mi familia.

***Ricardo Quezada***



## Resumen

El presente proyecto, relacionado con el área de la visión artificial y su aplicación en la robótica, contempla el desarrollo de dos agentes robóticos móviles, capaces de interactuar entre ellos y con su ambiente, utilizando como base de sus decisiones un modulo de visión artificial independiente y en tiempo real. Este proyecto se enfoca a la competencia internacional Robocup, y pretende lograr que los agentes se desenvuelvan bajo condiciones de estrategia programadas.

El problema abordado consiste en la construcción de dos agentes robóticos y el desarrollo de un modulo de visión artificial, que permita la aplicación de estrategias a dichos agentes. Para lo anterior, se dispone de un campo de fútbol a escala y una cámara de video ubicada en la parte superior del campo de juego. Esta cámara provee de imágenes en tiempo real, a un computador destinado al procesamiento de ellas. De esta forma, dependiendo de la estrategia aplicada, se pueden inferir movimientos mediante comunicación inalámbrica a los agentes robóticos antes mencionados.

Este trabajo se dividió en tres módulos principales: módulo de visión, el cual es el encargado de la captura de imágenes, su segmentación, el reconocimiento de patrones y colores; el módulo de estrategia, donde se infieren movimientos en base a cálculos matemáticos y de cinemática; y por ultimo y el módulo de comunicación, el cual hace entrega de la decisión tomada por la estrategia, a los agentes robóticos vía inalámbrica.

Para la programación del framework y sus módulos, se utilizó modelamiento UML, lenguaje orientado a objetos C++ y un ciclo de desarrollo incremental, el cual consta de 3 incrementos funcionales que abarcan el desarrollo de los 3 módulos mencionados mas la construcción de los agentes robóticos y su ambiente.

En definitiva, se obtuvo un módulo de visión estable y 3 estrategias para la aplicación y estudio de los agentes: seguir la pelota, tiro penal y monito mayor.

El factor más importante a considerar en futuras investigaciones, es sin duda el dispositivo de video, el cual determina la calidad de las capturas y estabilidad del sistema final.

Finalmente, este tipo de trabajos se ven potenciados en ambientes multidisciplinarios.

# Índice General

Capítulo I – Introducción .....	18
1.1 Descripción del Problema .....	19
1.1.1 RoboCup.....	19
1.1.1.1 Nacimiento .....	19
1.1.1.2 Objetivos y Metas.....	19
1.1.1.3 Características de la Competencia.....	20
1.1.2 Ambito del problema a resolver .....	23
1.1.2.1 Alcances .....	23
1.1.2.2 Limitaciones .....	24
1.2 Objetivos del proyecto .....	25
1.2.1 Objetivo general .....	25
1.2.2 Objetivos específicos.....	25
1.3 Metodología a utilizar .....	25
1.4 Trabajos relacionados.....	27
1.5 Organización del documento.....	29
Capítulo II – Bases Teóricas .....	31
2.1 Robótica y comunicación .....	31
2.1.1 Microcontroladores .....	31
2.1.1.1 Nacimiento .....	31
2.1.1.2 ¿Qué es un microcontrolador?.....	32
2.1.1.3 Diferencias entre un Microprocesador y un Microcontrolador .....	32
2.1.1.4 Aplicación de los dispositivos más comunes .....	33
2.1.2 Comunicación Inalámbrica .....	34
2.1.2.1 ¿Qué es la comunicación inalámbrica? .....	34
2.1.2.2 WIFI .....	35
2.1.2.3 Bluetooth.....	36
2.1.2.4 Infrarrojo .....	37
2.1.2.4.1 Clasificación.....	38
2.1.2.5 Antena .....	39
2.1.2.5.1 Parámetros de una antena.....	39

2.1.2.5.2	Tipos de antena.....	40
2.1.2.5.3	Radio Frecuencia.....	41
2.1.2.5.4	Propagación de ondas.....	42
2.2	Visión artificial.....	43
2.2.1	¿Que es la Visión Artificial? .....	43
2.2.2	Historia .....	44
2.2.3	Problemas .....	44
2.2.4	Relación con otros campos.....	45
2.2.5	Aplicaciones de la Visión artificial .....	46
2.2.6	Niveles de procesamiento.....	46
2.2.7	Funciones de la visión artificial .....	47
2.2.7.1	Adquisición de la imagen.....	47
2.2.7.2	Pre-procesamiento.....	47
2.2.7.3	Extracción de características .....	47
2.2.7.4	Detección / segmentación.....	48
2.2.7.5	Procesamiento de alto nivel.....	48
2.2.8	Color.....	48
2.2.8.1	Modelo de color .....	48
2.2.8.1.1	Modelo RYB .....	48
2.2.8.1.2	Modelo RGB .....	49
2.2.8.1.3	Modelo CMY .....	49
2.2.8.1.4	Modelo YUV.....	50
2.2.8.1.5	Otros modelos .....	51
2.2.8.2	Espacios de color.....	51
2.2.8.2.1	Espacio RGB .....	51
2.2.8.2.2	Espacio CMY .....	52
2.2.8.2.3	Espacio YIQ .....	53
2.2.8.2.4	Espacio HSB .....	53
2.2.8.2.5	Otros espacios de color .....	54
2.2.9	Segmentación de la imagen.....	54
2.2.9.1	Técnicas de segmentación de imágenes .....	55
2.2.9.1.1	Segmentación basada en bordes .....	55
2.2.9.1.2	Segmentación basada en regiones .....	56
2.2.9.1.3	Segmentación de texturas.....	57

2.2.9.1.4	Segmentación por agrupación de píxeles .....	58
2.2.10	Reconocimiento de patrones .....	59
2.3	Inteligencia Artificial .....	59
2.3.1	Agentes inteligentes .....	59
2.3.1.1	¿Qué es un Agente?.....	60
2.3.1.2	Características .....	61
2.3.1.3	Clasificación.....	63
Capítulo III	– Análisis y Diseño.....	65
3.1	Introducción .....	65
3.2	Planificación y especificación de requerimientos .....	65
3.2.1	Requerimientos.....	65
3.2.1.1	Requerimientos Funcionales .....	66
3.2.1.2	Requerimientos no funcionales .....	66
3.2.1.3	Planilla combinada .....	67
3.2.2	Actores del sistema.....	68
3.2.2.1	Descripción de los casos de uso de alto nivel .....	69
3.2.2.2	Descripción de los casos de uso expandidos .....	70
3.2.2.3	Diagrama de caso de uso .....	72
3.2.2.4	Planificación de Casos de uso según Ciclos de Desarrollo .....	73
3.3	Análisis del Sistema .....	73
3.3.1	Modelo conceptual .....	74
3.3.2	Diagramas de secuencia del sistema .....	75
3.4	Diseño del Sistema .....	76
3.4.1	Casos de uso reales.....	76
3.4.2	Diagrama Arquitectónico .....	78
3.4.3	Diagrama de paquetes .....	79
3.4.4	Diagramas de clases por módulos .....	79
3.4.4.1	Diagrama de clases simplificado módulo visión.....	79
3.4.4.2	Diagrama de clases completo módulo visión.....	80
3.4.4.3	Diagrama de clases simplificado módulo estrategia .....	81
3.4.4.4	Diagrama de clases completo módulo estrategia .....	81
3.4.4.5	Diagrama de clases simplificado módulo comunicación .....	82
3.4.4.6	Diagrama de clases completo módulo comunicación .....	82
Capítulo IV	– Implementación y Pruebas .....	84

4.1	Introducción .....	84
4.2	Construcción del robot jugador .....	84
4.2.1	Características del ambiente y los robots según Robocup .....	84
4.2.1.1	El Campo de Juego.....	84
4.2.1.2	La pelota.....	85
4.2.1.3	El robot.....	85
4.2.1.4	El dribbling del robot .....	85
4.2.1.5	Colores y Marcadores.....	86
4.2.1.6	Comunicación inalámbrica.....	87
4.2.2	Construcción del prototipo .....	87
4.2.2.1	Ensamblaje de motores y estructura base.....	87
4.2.2.2	Ensamblaje del segundo nivel y circuitos .....	89
4.2.2.3	Ensamblaje del tercer nivel y cabeza .....	89
4.3	Construcción de la Estructura Metálica .....	90
4.3.1	Diseño Inicial .....	90
4.3.1.1	Aspecto del diseño inicial .....	91
4.3.1.2	Problemas con el diseño inicial de la estructura metálica.....	91
4.3.2	Diseño final estructura metálica.....	91
4.3.2.1	Aspecto final de la estructura metálica .....	92
4.4	Control de Movimiento .....	93
4.4.1	Tecnología utilizada .....	94
4.4.2	Características del transmisor y receptor de radiofrecuencia.....	94
4.4.3	Circuito de cada transmisor y receptor.....	96
4.4.4	Características del controlador de motores .....	97
4.4.5	Circuito del Micro Dual Serial Motor Controller.....	98
4.4.6	Protocolo de comunicación .....	98
4.4.7	Envío de instrucciones .....	99
4.4.8	Módulo interno del robot.....	100
4.4.9	Algoritmos para el control de motores .....	100
4.4.9.1	Algoritmo del transmisor .....	100
4.4.9.1.1	Código BS2 para el transmisor.....	100
4.4.9.2	Algoritmo que controla a los robots .....	101
4.4.9.2.1	Código BS2 para el robot .....	101
4.5	Implementación de la visión artificial.....	102

4.5.1	Librería de visión CMVision.....	102
4.5.2	Reconocimiento de patrones .....	104
	Elementos .....	104
4.5.2.1	Descripción del problema.....	105
4.5.2.2	Convenciones .....	105
4.5.2.3	Algoritmo aplicado.....	106
4.5.3	Posicionamiento del robot.....	107
4.5.3.1	Descripción del problema.....	107
4.5.3.2	Consideraciones importantes.....	107
4.5.3.3	Algoritmo aplicado.....	108
4.6	Implementación de estrategias .....	110
4.6.1	Seguir la pelota.....	110
4.6.1.1	Movimiento rectilíneo .....	110
4.6.1.2	Movimiento curvilíneo.....	111
4.6.1.3	Algoritmo aplicado.....	111
4.6.1.4	Consideraciones de movimiento y estimación .....	112
4.6.2	Tiro penal .....	113
4.6.2.1	Algoritmo aplicado.....	113
4.6.3	Perseguir compañero .....	114
4.6.3.1	Algoritmo aplicado.....	114
4.7	Pruebas .....	115
4.7.1	Pruebas de caja negra módulo de visión .....	115
4.7.1.1	Prueba a la conectividad de la cámara con el sistema.....	115
4.7.1.2	Prueba a la imagen captada, sin sensibilidad de luz.....	115
4.7.1.3	Prueba a la imagen captada, con sensibilidad de la luz.....	116
4.7.2	Pruebas de caja negra módulo de estrategia.....	116
4.7.2.1	Prueba a la entrega de datos con un robot y la pelota .....	116
4.7.3	Pruebas de caja negra módulo de comunicación.....	118
4.7.3.1	Prueba a la recepción de los datos con interferencia.....	118
4.7.3.2	Prueba a la recepción de los datos sin interferencia.....	119
4.7.4	Pruebas de caja blanca.....	120
4.7.4.1	Clase Analizador – Función calcular .....	120
4.7.4.1.1	Camino básico .....	120
4.7.4.1.2	Caminos a probar .....	121

4.7.4.1.3	Tabla de casos de prueba.....	121
4.7.4.2	Clase Inferencia: Función de Velocidad Dirección .....	122
	Camino básico .....	123
4.7.4.2.1	Caminos a probar .....	123
4.7.4.2.2	Tabla de casos de prueba.....	124
Capítulo V	– Conclusiones.....	126
5.1	Conclusiones Generales .....	126
5.1.1	Respecto de la visión artificial .....	126
5.1.2	Respecto de la estrategia .....	127
5.1.3	Respecto de la comunicación .....	127
5.2	Trabajos posteriores .....	128
Bibliografía.....		129
Abreviaturas .....		131
Anexo A: Filtro de Kalman .....		133
Filtro de Kalman.....		133
El proceso para estimar .....		133
Los orígenes del filtro .....		134
Funcionamiento del filtro de Kalman.....		136

# Índice de Figuras

Figura 1.1: (A) Simulation League y (B) Small Size League. ....	20
Figura 1.2: (A) Middle Size League y (B) categoría Four-Legged League. ....	21
Figura 1.3: (A) RoboCup Rescue y (B) Humanoid League. ....	22
Figura 1.4: Proyectos RoboCupJunior. ....	22
Figura 1.5: Esquema del problema a resolver. ....	23
Figura 1.6: Equipo UChile-2 categoría F180, participando en el American Open 2003. ....	27
Figura 1.7: CMDragons campeones categoría F180, RoboCup 1997. ....	28
Figura 2.1: Microcontrolador Intel 4004. ....	32
Figura 2.2: Esquema básico de microprocesador y microcontrolador. ....	33
Figura 2.3: Ejemplo de una red WIFI. ....	35
Figura 2.4: Ejemplo de Piconet. ....	37
Figura 2.5: Ejemplo de comunicación con dispositivos mediante infrarrojo. ....	39
Figura 2.6: Modos de propagación de ondas de radio. ....	43
Figura 2.7: Ejemplo de rectificación de perspectiva. ....	45
Figura 2.8: Ejemplo de rectificación de distorsión de lente. ....	45
Figura 2.9: Círculo cromático RYB. ....	49
Figura 2.10: Modelo RGB. ....	49
Figura 2.11: Círculo cromático CMY. ....	50
Figura 2.12: Cubo de color RGB. ....	52
Figura 2.13: Representación de los colores CMYK. ....	53
Figura 2.14: Cubo de color YIQ. ....	53
Figura 2.15: Ejes HSB o HSV. ....	54
Figura 2.16: Diseño del clasificador. ....	59
Figura 2.17: Comportamiento básico de una agente inteligente. ....	63
Figura 3.1: Actores del sistema: Cámara de video, operador del sistema y robot. ....	68
Figura 3.2: Diagrama de casos de uso del sistema. ....	72
Figura 3.3: Detalle de la división del trabajo en la creación de casos de uso, para los 3 incrementos propuestos. ....	73
Figura 3.4: Modelo conceptual de sistema. ....	74

Figura 3.5: Diagrama de secuencia – Iniciar sistema de visión .....	75
Figura 3.6: Diagrama de secuencia – Inicio de estrategia de movimiento.....	75
Figura 3.7: Diagrama de secuencia – Captura de imágenes.....	75
Figura 3.8: Diagrama de secuencia – Envío de acción .....	76
Figura 3.9: Pantalla principal. ....	77
Figura 3.10: Diagrama arquitectónico del sistema. ....	78
Figura 3.11: Diagrama de paquetes del sistema. ....	79
Figura 3.12: Diagrama de clases simplificado - Módulo de visión.....	79
Figura 3.13: Diagrama de clases completo - Módulo de visión.....	80
Figura 3.14: Diagrama de clase simplificado – Módulo de estrategia. ....	81
Figura 3.15: Diagrama de clases completo – Módulo de estrategia.....	81
Figura 3.16: Diagrama de clases simplificado - Módulo de comunicación. ....	82
Figura 3.17: Diagrama de clases completo - Módulo de comunicación. ....	82
Figura 4.1: Medidas de la cancha expresadas en milímetros. ....	85
Figura 4.2: Posición del dispositivo de dribbling del robot. ....	86
Figura 4.3: Ejemplo de distribución de marcadores de colores para equipo amarillo. ....	86
Figura 4.4: Disposición de las perforaciones en la base (modelo 2D). ....	87
Figura 4.5: Rueda Omnidireccional (1) y rueda convencional (2). ....	88
Figura 4.6: Aspecto de la estructura base.....	88
Figura 4.7: Disposición perforaciones en la mica del segundo nivel (modelo 2D). ....	89
Figura 4.8: Aspecto del robot luego de instalado el tercer nivel.....	90
Figura 4.9: Renderizado de la estructura metálica según diseño inicial. ....	91
Figura 4.10: Vista lateral y frontal de la estructura (medidas en centímetros). ....	92
Figura 4.11: Renderizado estructura metálica según diseño final.....	92
Figura 4.12: Aspecto de la estructura según el modelo final. ....	93
Figura 4.13: Transmisor (#27980), 4 pines. ....	94
Figura 4.14: Receptor (#27981), 5 pines.....	95
Figura 4.15: El circuito del transmisor a la izquierda y a la derecha el circuito del receptor. .	96
Figura 4.16: Micro Dual Serial Motor Controller (SMC02B). ....	97
Figura 4.16: Esquema de conexión de los motores, la BS2, la batería y el controlador. ....	98
Figura 4.17: Esquema del flujo de datos. ....	99
Figura 4.18: A la izquierda la imagen de video original, y a la derecha la imagen con colores clasificados por los umbrales YUV predefinidos.....	102

Figura 4.19: Un histograma YUV de un campo de fútbol de RoboCup, con un umbral definido para un color amarillo. ....	103
Figura 4.20: Elementos del Problema. ....	104
Figura 4.21: Ejemplo de colores superiores para cada Robot. ....	105
Figura 4.22: Ejemplo de matriz que maneja 2 jugadores y 3 colores para estos. ....	106
Figura 4.23: Diferentes ángulos que puede tomar el robot con respecto de si mismo. ....	108
Figura 4.24: Diferentes ángulos del robot (en 0°) respecto de la pelota. ....	108
Figura 4.25: Visual superior del robot. ....	108
Figura 4.26: Ejemplo de dos puntos en el plano. ....	109
Figura 4.27: Grafico de resultados prueba de recepción de datos con interferencia. ....	118
Figura 4.28: Grafico de resultados prueba de recepción de datos sin interferencia. ....	119
Figura 4.29: Grafo asociado a los caminos básicos (función calcular). ....	120
Figura 4.30: Grafo asociado a los caminos básicos (función detVelocidadDireccion). ....	123

## Índice de Tablas.

Tabla 2.1: Empresas fabricantes de microcontroladores y sus productos más populares.....	34
Tabla 2.2: Bandas de radio con sus frecuencias.....	42
Tabla 2.3: Clasificación de los agentes.....	63
Tabla 3.1: funcionalidades del sistema .....	66
Tabla 3.2: Atributos del sistema de visión.....	67
Tabla 3.3: Tabla combinada entre requisitos funcionales y no funcionales del sistema.....	67
Tabla 3.4: Caso de uso alto nivel – Iniciar sistema de visión.....	69
Tabla 3.5: Caso de uso alto nivel – inicio de una estrategia de movimiento.....	69
Tabla 3.6: Caso de uso de alto nivel – captura de imágenes.....	69
Tabla 3.7: Caso de uso de alto nivel – Envío de acción.....	69
Tabla 3.8: Caso de uso expandido – Iniciar sistema de visión.....	70
Tabla 3.9: Caso de uso expandido – Inicio de estrategia de movimiento.....	70
Tabla 3.10: Caso de uso expandido – Captura de imágenes.....	71
Tabla 3.11: Caso de uso expandido – Envío de acción.....	71
Tabla 3.12: Casos de uso reales del sistema.....	78
Tabla 4.1: Pines utilizados del transmisor (#27980).....	95
Tabla 4.2: Pines utilizados del receptor (#27981).....	95
Tabla 4.3: Pines del SMC02B y su función .....	97
Tabla 4.4: Códigos para controlar los motores.....	100
Tabla 4.5: Algoritmo del Transmisor.....	100
Tabla 4.6: Algoritmo control de motores.....	101
Tabla 4.7: Ángulos de entrada y resultados obtenidos, prueba a la estrategia “seguir la pelota”. .....	117
Tabla 4.8: Código función “calcular” .....	120
Tabla 4.9: Casos de prueba función calcular.....	121
Tabla 4.10: Código función “detVelocidadDiureccion” .....	122
Tabla 4.11: Casos de prueba función detVelocidadDireccion.....	124

## CAPITULO I – INTRODUCCION

## Capítulo I – Introducción

Quizás uno de los campos de la inteligencia artificial con más auge y que mejor ha representado a esta ciencia, resolviendo problemas complicados de manera óptima, es la visión artificial. Ésta, a grandes rasgos tiene por objetivo modelar la imagen digital suministrada por un sensor, de forma que un computador sea capaz de interpretarla y utilizarla como base para la toma de decisiones de un agente<sup>1</sup>.

Las investigaciones realizadas en base a la visión artificial van desde el reconocimiento de patrones, objetos y personas hasta el control de calidad, triangulación y visión humana. Además, abarca áreas tan dispares como la construcción, la medicina, aplicaciones comerciales, entretenimiento y educación.

Inmersa en estas áreas, las empresas y universidades mundiales invierten en mejorar las técnicas y procedimientos para producir valor a través de este campo. Es así como nace en 1997 en Japón, el proyecto ahora internacionalmente conocido como RoboCup, cuyo principal objetivo es promover la investigación y educación en inteligencia artificial, usando robots autónomos en diferentes categorías, donde dos equipos deben disputar un partido de fútbol.

La categoría F180, de robots pequeños (small size), apunta directamente al modelo que se desea construir en el presente trabajo. Este contempla la creación de un framework o marco de trabajo, el cual deberá contener módulos de visión artificial, construcción de un robot o agente móvil, comunicación inalámbrica entre computador-robot, inteligencia del agente móvil y control de movimientos del agente, entre otros. Todos estos deberán estar integrados para obtener las imágenes del robot en el campo de juego, luego procesarlas en tiempo real, lograr una decisión de movimiento y finalmente transmitir la decisión para que el agente actúe.

---

<sup>1</sup> **Agente:** un tipo de programa informático que, por encargo de un usuario u otro programa, realiza de forma autónoma tareas que requieren cierto grado de inteligencia y aprendizaje.

## **1.1 Descripción del Problema**

A continuación se presenta un marco introductorio de lo que es la competencia RoboCup y sus diferentes categorías de competencia, con el objetivo de lograr un completo entendimiento del problema que se quiere abordar.

### **1.1.1 RoboCup**

#### **1.1.1.1 Nacimiento**

La idea fue iniciada por el profesor Alan Mackworth, de la universidad de British Columbia de Canadá, en un artículo titulado “On seen robots” presentado en 1992 y publicado un año después, en donde por primera vez se aludía a la creación de jugadores robóticos de fútbol. En forma independiente, un grupo de investigadores japoneses organizaron un seminario de “Grandes retos en inteligencia artificial”, llevado a cabo en Tokio de 1992. En este seminario se discutió seriamente el juego de fútbol soccer para promover la ciencia y la tecnología.

En 1993, investigadores japoneses lanzaron la primera competencia robótica llamada “Robot J-League”. Poco tiempo después se decidió invitar al resto de la comunidad científica a esta competencia por lo que su nuevo nombre fue “Robot World Cup Initiative”, el que luego cambiarían por “RoboCup”.

A partir de esta fecha, se realizaron varios encuentros para el desarrollo de esta gran empresa, siendo hasta 1997 cuando la primera competencia oficial de RoboCup se llevó a cabo en Japón, país que impulsó la idea. Desde entonces, cada año se reúne parte de la comunidad científica y académica mundial, presentando sus respectivos avances y compitiendo en las diferentes categorías que el certamen ofrece.

#### **1.1.1.2 Objetivos y Metas**

La RoboCup tiene como finalidad, promover la investigación en robótica y la inteligencia artificial (IA) a través de un problema estándar (el juego del fútbol) en donde una amplia gama de tecnologías pueden integrarse y examinarse, así como utilizarse en proyectos educativos y de investigación.

La RoboCup y sus impulsores, se han puesto la siguiente meta: “Para el año 2050, conseguir desarrollar un equipo de robots humanoides totalmente autónomos que pueda ganar un partido de fútbol al equipo humano campeón del mundo”, la cual sin duda ha causado un gran impacto social, convirtiéndose en el mayor reto de la robótica moderna.

### 1.1.1.3 Características de la Competencia

Cada año la competencia reúne a más de 250 equipos (280 el 2005) y visitantes (20.000 el 2005) de 36 países del mundo, a través de 6 ligas, las cuales incluyen diferentes tipos de robots electro-mecánicos y agentes simulados computacionalmente.

La competencia no solo consta de fútbol, también existen otras disciplinas enmarcadas en categorías especiales donde se mide el nivel de especialización que alcanzan los robots cada año. Además durante la competencia se realizan diferentes exposiciones, foros, conferencias técnicas, programas de desafíos y programas educativos.

Las categorías generales de la competencia se detallan a continuación:

- **Simulation League:** Consiste en la simulación de un partido de fútbol, muy similar a un juego de computadora, pero sin robots de por medio ni humanos que los manejen. Cada jugador tiene su estrategia de juego y características, y los equipos consisten en una colección de programas estratégicamente acoplados. Los partidos duran 10 minutos y se ven por una pantalla gigante, como se indica en la figura 1.1 A.
- **Small Size League:** Esta categoría contempla a 2 equipos de 5 pequeños robots, los cuales juegan un partido de fútbol. Al igual que la anterior, no hay participación de humanos en el desarrollo del juego. Los robots llevan un patrón de color sobre ellos (ver figura 1.1 B) para diferenciarlos de los jugadores del equipo rival y sus compañeros. Una cámara dispuesta sobre el lugar capta los movimientos propios y del rival retroalimentando el movimiento de los robots, mediante comunicación inalámbrica. Esta es la categoría donde se enmarca este proyecto.

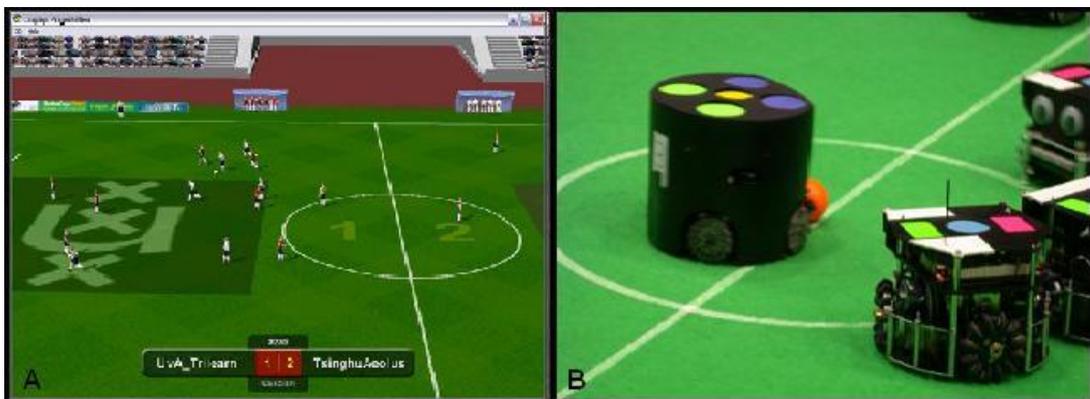


Figura 1.1: (A) Simulation League y (B) Small Size League.

- **Middle Size League:** Esta categoría es básicamente un juego de fútbol entre 2 equipos de 4 robots, los cuales poseen los sensores sobre ellos; como se ve en la figura 1.2 A, sensor que habitualmente es una cámara. No hay participación externa de humanos y se permite la comunicación inalámbrica entre los robots.
- **Four-Legged League:** Acá son 2 equipos de 4 robots AIBO<sup>2</sup> (ver Figura 1.2 B), los cuales son 100% programables y con todos los sensores necesarios integrados. Son identificados por un color y la comunicación entre ellos es inalámbrica. Al igual que en las anteriores no hay intervención de humanos.

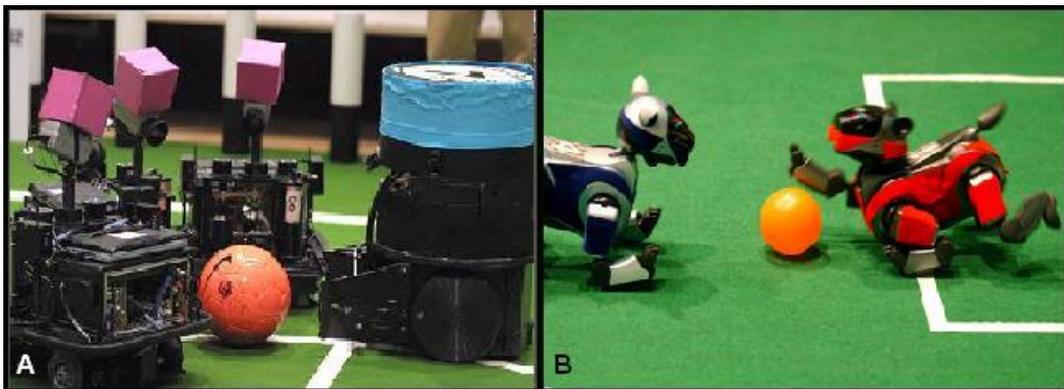


Figura 1.2: (A) Middle Size League y (B) categoría Four-Legged League.

- **Humanoid League:** Esta liga es nueva y contempla la competencia en un partido de fútbol a 2 equipos de 2 robots humanoides, es decir, robots con aspecto y sentidos humanos (ver figura 1.3 B). Esta categoría se divide en 2: *KidSize*, para robots de menos de 60 cms. de altura y *TeenSize* para robots de más de 60 cms. de altura. Acá entran en juego muchas otras variantes como el balance del robot al correr o patear la pelota.
- **RoboCupRescue:** Esta categoría consta de la resolución de un problema en ambientes hostiles, como catástrofes naturales o situaciones de peligro extremo. Esta dividida en 2 ligas: *Simulation League*, donde no hay presencia de robots, como se ve en la figura 1.3 A y *Real Robot League* donde hay presencia de equipos de robots. Con esta liga se pretende crear en el futuro una brigada internacional de rescate, robotizada.

---

<sup>2</sup> AIBO: ("amigo" en japonés) es un perro robot fabricado por Sony, el cual posee múltiples sensores integrados, conexión inalámbrica a Internet y lo mas importante: módulos programables con lo cual se le puede dar inteligencia.

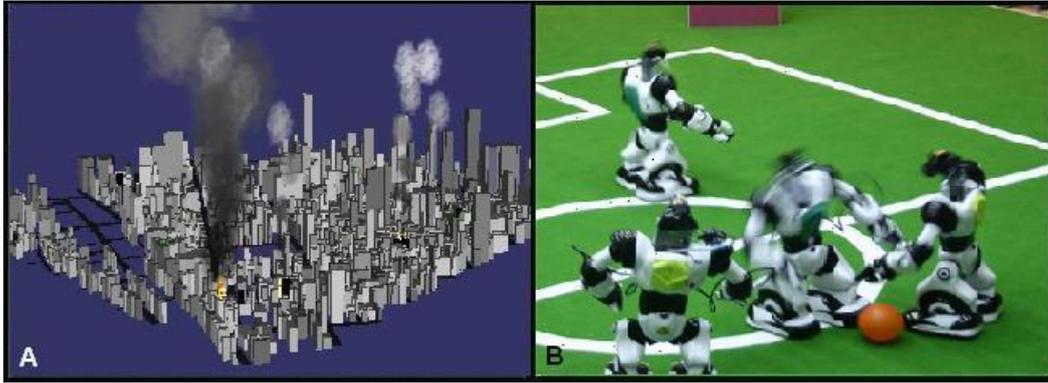


Figura 1.3: (A) RoboCup Rescue y (B) Humanoid League.

- **RoboCupJunior:** Categoría enfocada a proyectos educacionales locales, nacionales e internacionales, donde el objetivo es introducir a niños de primaria y secundaria, la robótica a través de la RoboCup (ver figura 1.4). Dentro de esta categoría existen desafíos de fútbol, baile y rescate, donde se ponen a prueba las habilidades de los niños sobre el control de sus robots.



Figura 1.4: Proyectos RoboCupJunior.

**RoboCup@Home:** Es una nueva liga dentro de la competencia RoboCup orientada a la creación de aplicaciones del mundo real e interacción con máquinas-humanoides y robots autónomos. Acá se escogen los ganadores mediante la puntuación de los robots en distintas áreas como la interacción con humanos, aspectos sociales relevantes (como saludar, caminar, etc.), aplicaciones dirigidas u orientadas, entregar un aporte científico relevante y que tome poco tiempo procesar las acciones.

## 1.1.2 Ambito del problema a resolver

### 1.1.2.1 Alcances

Este proyecto contempla la construcción de dos robots o agentes móviles con las características de la categoría Robocup Small Size F180, dispuestos en una cancha a escala y observados por una cámara ubicada en la parte superior de ella. Este dispositivo estará conectado a un computador, el cual procesará la imagen e inferirá las acciones a seguir por los robots, dependiendo de la elección de una estrategia implementada. La figura 1.5 muestra el esquema del problema.

Para esto, se desarrollará una librería de visión, que procese las imágenes en tiempo real y dictamine qué acciones corresponde hacer según las capturas realizadas al ambiente de los robots. Esta librería incluirá algoritmos de identificación de patrones, posicionamiento, direccionamiento, control de movimiento y toda la inteligencia necesaria para entender y decidir correctamente. Además debe tener una interfaz de comunicación inalámbrica entre el computador y los robots.

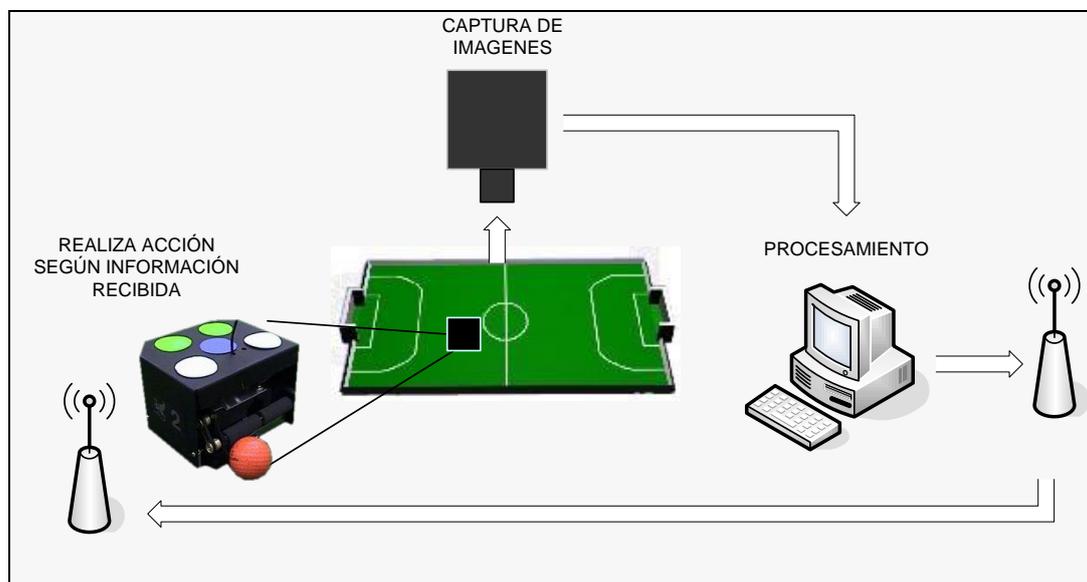


Figura 1.5: Esquema del problema a resolver.

### **1.1.2.2 Limitaciones**

El sistema a desarrollar, no contempla las siguientes funciones y características propias de la competencia:

- El sistema no permitirá la total interacción entre los robots, pues solo se le programarán algunas estrategias a modo de ejemplo, para demostrar el trabajo realizado en la visión artificial.
- Los robots no poseerán un sistema de dribbling (ver Capítulo 4, sección 4.2.1.4 Dribbling del robot) que les permita patear la pelota.
- La arquitectura de las ruedas no permitirá a los robots, moverse libremente. Pues no se poseen ruedas omnidireccionales para este propósito.
- El sistema de comunicación no permitirá que los robots realicen tareas paralelas. Además no existe un mecanismo de interrupción de acciones, por lo que la ejecución de una siguiente instrucción no podrá hacerse, mientras no termine la actual.

## **1.2 Objetivos del proyecto**

### **1.2.1 Objetivo general**

Implementar un módulo de visión para sistemas robóticos en tiempo real sobre ambientes dinámicos, capaz de entregar órdenes dependiendo de la entrada recibida. La idea es que el módulo sea independiente al factor: agente robótico o no robótico. Para la demostración de su aplicación, se contempla la construcción 2 robots que se desenvuelvan en un campo de fútbol.

### **1.2.2 Objetivos específicos**

- Desarrollar un módulo de visión que comprenda reconocimiento de patrones.
- Implementar un módulo de visión que comprenda direccionamiento y posicionamiento del robot en el ambiente, junto a un control de movimiento en aplicaciones robóticas y no robóticas. Esto en tiempo real.
- Obtener una librería para proyectos posteriores, la cual pueda ser utilizada para diferentes aplicaciones, ya sean éstas robóticas o no.
- Crear un módulo reutilizable e independientemente funcional, para que el grupo de robótica de nuestra Universidad continúe la construcción de un equipo de fútbol de robots para la RoboCup©.

## **1.3 Metodología a utilizar**

Se utilizará el modelo de desarrollo incremental bajo la metodología de desarrollo orientado a objetos, con el propósito de realizar pruebas y documentación efectiva por cada incremento, además de aclarar aspectos técnicos de la funcionalidad. Esto con estándares de UML y herramientas de apoyo como CMVision y lenguaje C++, todo bajo ambiente Linux.

Los incrementos propuestos son desarrollados bajo una estructura lógica de construcción, logrando tener un producto funcional al término de cada uno de ellos. Es así como el primer incremento constituirá la construcción del robot y algunos módulos de comunicación inalámbrica que permitan moverlo manualmente. Posteriormente en el segundo incremento se realiza la construcción de la estructura que sostiene la iluminación y la cámara. Además también se comienza el desarrollo del módulo de estrategia. Por ultimo en el tercer y último

incremento se termina de desarrollar los módulos y se integran para dar funcionalidad completa al sistema.

## 1.4 Trabajos relacionados

A continuación se detallan los proyectos desarrollados que poseen características similares a éste, correspondientes a destacadas universidades nacionales y extranjeras:

- **Universidad de Chile**

La Universidad de Chile tiene participación activa en la RoboCup y eventos latinoamericanos similares desde el año 2003. Sin embargo, es en el año 2004 donde obtiene un notable desempeño en la RoboCup realizada en Lisboa Portugal. Aquí el equipo UChile-1 participó en la categoría Four Legged (cuadrúpedos) con perros AIBO de Sony, en la cual obtuvieron el 19º lugar de 24 participantes en fútbol y el 8º puesto en el challenger de condiciones adversas (luego el 2005 obtendrían el 7º lugar en este mismo challenger).

Actualmente la Universidad de Chile tiene 2 equipos de robots: el UChile-1 que es de cuadrúpedos y el UChile-2 que es de robots pequeños para la categoría small size F180 (categoría en la que se enmarca el presente proyecto y que se muestra en la figura 1.6).



Figura 1.6: Equipo UChile-2 categoría F180, participando en el American Open 2003.

- **Otras Universidades Chilenas**

Otras universidades Chilenas que también poseen grupos dedicados especialmente a esta competencia son la Universidad Federico Santa María (Valparaíso) y la Universidad de Concepción, esta última en la categoría Four Legged con perros AIBO. Por otro lado, la Pontificia Universidad Católica de Chile está comenzando a incursionar en el tema participando en retos latinoamericanos de robótica y en categorías de RoboCup small size.

- **Equipo CMDragons de la Universidad Carnegie Mellon de USA**

Esta prestigiosa universidad estadounidense ha tenido una destacada participación en la competencia con la categoría F180 desde 1997, obteniendo 3 veces el primer lugar (97', 98', 06'), llegando 2 veces a las semifinales y 2 veces a los cuartos de final. Además, esta universidad tiene equipos en las categorías de simulación, cuadrúpedos y F2000 donde han obtenido numerosos triunfos. En la figura 1.7 se muestra el equipo de 1997.

Por otro lado, esta universidad ha aportado múltiples herramientas software y múltiples publicaciones para el desarrollo de aplicaciones que apuntan a mejorar y simplificar el uso de agentes robóticos móviles. Entre éstas podemos encontrar UberSim, simulador realista 3D que incorpora ingeniería física en el movimiento con parámetros de robots configurables. CMVision, la cual es una rápida y poderosa herramienta de procesamiento del color en tiempo real. El sistema realiza la visión del color global a bajo nivel con la tarjeta de video sin el uso del hardware especial para este propósito. Esta herramienta es la base del sistema desarrollado en este trabajo.

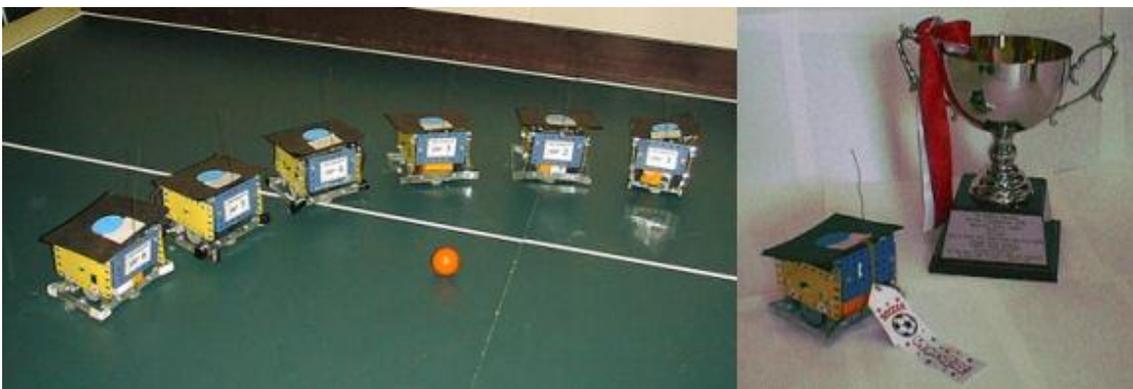


Figura 1.7: CMDragons campeones categoría F180, RoboCup 1997.

- **Equipo FU-Fighters de la Universidad Freie de Berlin**

Otro destacado equipo es *FU-Fighters* de la Universidad de Berlín, Alemania. Ellos participaron por primera vez en 1999 en la Robocup de ese año en Suecia, donde obtuvieron el segundo lugar en la categoría F180. Se han coronado 4 veces campeón de Europa y 2 veces campeón de la RoboCup (2004, 2005) en la categoría F180.

Este equipo se destaca además por poseer una excelente participación en la categoría F2000 y numerosos aportes al campo de la visión artificial e inteligencia artificial.

## **1.5 Organización del documento**

Finalmente, este informe está organizado como se detalla a continuación:

- El “CAPITULO 1: Introducción” da a conocer al lector el problema a resolver, mostrando detalladamente los objetivos y la forma en que se abordará el problema.
- El “CAPITULO 2: Bases Teóricas” hace referencia a las tecnologías aplicadas al proyecto; desde la comunicación inalámbrica a los detalles del funcionamiento de la visión artificial.
- En el “CAPITULO 3: Análisis y Diseño” se detallan las 2 etapas más importantes en el desarrollo de sistemas, modularizando el contenido en 3 partes principales: visión, estrategia y comunicación.
- En el “CAPITULO 4: Implementación y Pruebas” se explica la forma en que usaron las tecnologías (comunicación inalámbrica, software, microcontroladores, estrategia) en la implementación de la solución del sistema.

En el “CAPITULO 5 – Conclusiones” se presentan las conclusiones generales y los trabajos futuros que pueden ser desarrollados a partir de este proyecto.

## **CAPITULO II – BASES TEORICAS**

## Capítulo II – Bases Teóricas

Las bases teóricas que fundamentan el trabajo realizado en este proyecto se dividen en 3 grandes temas, para facilitar la lectura y comprensión de las materias que involucran el posterior desarrollo del mismo. Los temas son: robótica y comunicación, visión artificial y por ultimo inteligencia artificial.

### **2.1 Robótica y comunicación**

La robótica es una ciencia o rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano o que requieren del uso de inteligencia. Las ciencias y tecnologías de las que deriva son: el álgebra, los autómatas programables, las máquinas de estados, la mecánica y la informática [WIK 06].

#### **2.1.1 Microcontroladores**

##### **2.1.1.1 Nacimiento**

Inicialmente, entre los años 1960 y 1970, los ingenieros diseñaban circuitos que implicaban muchos componentes electrónicos y cálculos matemáticos. Los sistemas de control se implementaban usando exclusivamente lógica de componentes, lo que hacía que fuesen dispositivos de gran tamaño y muy pesados. Además los circuitos electrónicos se diseñaban para una función específica, la cual no podía modificarse sin cambiar físicamente las conexiones (lógica cableada).

Para lograr una mayor velocidad y mejorar la eficiencia de estos dispositivos de control, se intentó reducir su tamaño, logrando en 1969 la construcción del primer microprocesador, el cual originó un cambio decisivo en las técnicas de diseño de la mayoría de los equipos, pues ahora la lógica no se modificaba físicamente, sino que se programaba (lógica programada).

La historia de los microprocesadores surge desde dos vías de desarrollo paralelas; una desde Intel y otra desde Texas Instruments. El primer microprocesador comercial fue el 4004 de Intel (ver figura 2.1), el cual manipulaba datos de 4 bits y era usado sólo por algunas

calculadoras. Aún así, el primer microprocesador completo de propósito general, fue el TMS1000 de Texas Instruments, el cual integraba un reloj, procesador, ROM de 1 Kb, RAM de 32 Bits, y soportes de E/S.

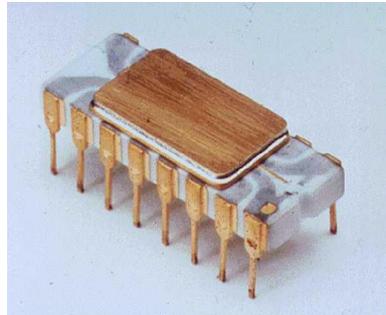


Figura 2.1: Microcontrolador Intel 4004.

En el año 1976, gracias al aumento de la densidad de la integración, salió a la luz la primera computadora en un chip (computer on a chip), es decir, se integraron junto con el microprocesador los subsistemas que anteriormente formaban unidades especializadas e independientes. A ese nuevo circuito integrado se le denominó "Microcontrolador".

#### **2.1.1.2 ¿Qué es un microcontrolador?**

Su nombre nos indica sus principales características: "Micro" por su tamaño, ya que es un dispositivo muy pequeño, y "Controlador" por que se utiliza principalmente para controlar otros circuitos o dispositivos electro-mecánicos.

En términos más técnicos, es un chip que incluye en su interior las tres unidades funcionales: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un sólo circuito integrado. Aunque sus prestaciones son limitadas, además de dicha integración, su característica principal es su alto nivel de especialización. Aunque los hay del tamaño de una estampilla, lo normal es que sean incluso más pequeños, ya que, lógicamente, forman parte del dispositivo que controlan [WIK 06].

#### **2.1.1.3 Diferencias entre un Microprocesador y un Microcontrolador**

Las principales características que diferencian a un microcontrolador de un microprocesador pueden verse en la figura 2.2 y en los detalles expuestos a continuación:

1. El microcontrolador emula más fácilmente un computador en funcionamiento, pues solo necesita un mínimo de chips externos de apoyo. La idea es que el chip se coloque en el dispositivo, enganchado a la fuente de energía y de información. Un microprocesador tradicional no le permitirá hacer esto, ya que espera que todas estas tareas sean manejadas por otros chips.

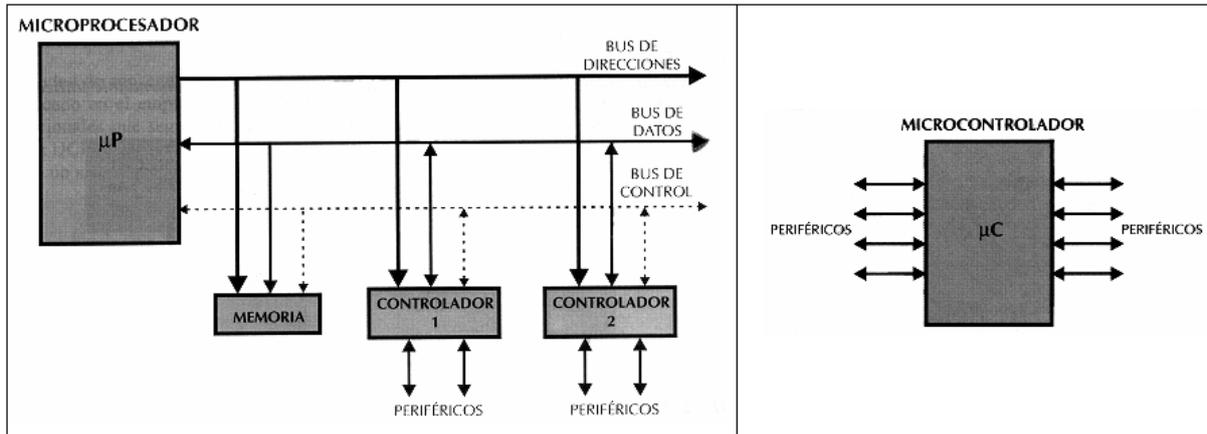


Figura 2.2: Esquema básico de microprocesador y microcontrolador.

2. El microcontrolador es de propósito específico, es decir, son programados o reprogramados para realizar una única tarea, mientras que los microprocesadores son de propósito general.
3. El costo para un sistema basado en microcontrolador es mucho menor y, lo mejor de todo, el tiempo de desarrollo de su proyecto electrónico se disminuye considerablemente al trabajar con una caja negra y no con cada componente por separado.

#### 2.1.1.4 Aplicación de los dispositivos más comunes

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16, 32 y hasta 64 bits. Aunque las prestaciones de los microcontroladores de 16, 32 y 64 bits son superiores a los de 4 y 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado pues son apropiados para la gran mayoría de las aplicaciones. En la tabla 2.1 podemos ver los principales proveedores de microcontroladores y su producto mas vendido.

Los microcontroladores a menudo se encuentran en aplicaciones domésticas (microondas, televisión, equipos de música), computadores con sus componentes (impresoras, módems,

lectores de discos), automóviles (control de climatización, seguridad eléctrica), instrumentación, aeronáutica, milicia y robótica, donde la comunicación entre controladores es una gran ventaja. Esto hace posible muchas tareas específicas al distribuir un gran número de microcontroladores por todo el sistema. La comunicación entre cada microcontrolador y uno central, permite procesar la información por un computador, o transmitirlo a otros microcontroladores del sistema.

<b>Fabricante</b>	<b>Microcontrolador mas Popular</b>
Comfile Technology Inc.	CUBLOC
Paralax	Basic Stamp 2
Microchip Technology	PIC16F8XX, PIC18F4550
Intel	80186, 80188 de 16 bits
Texas Instruments	TMS370, MSP430
Atmel	AT91SAM (ARM architecture)
Silicon Motion	SM350 – MP3/JPEG
Toshiba	TLCS-900 (16 and 32-bit CISC), TX19A (32-bit RISC)
Ibicom	IP2022 de 8 bits, IP3022 de 32 bits

Tabla 2.1: Empresas fabricantes de microcontroladores y sus productos más populares.

## 2.1.2 Comunicación Inalámbrica

Las comunicaciones inalámbricas están presentes en muchas de nuestras actividades diarias y su uso ha llegado a ser tan común, que perdemos la percepción de lo útil y a veces lo indispensable que puede llegar a ser. Las redes celulares para transmitir voz y datos han surgido para proveer la movilidad y disponibilidad de la comunicación que el ritmo acelerado de vida de las grandes urbes exige. La utilización de sensores infrarrojos y de radiofrecuencia proveen la comodidad de controlar y operar a distancia aparatos electrónicos volviendo más sencillo nuestro quehacer diario. Asimismo, la creación de estándares de comunicaciones inalámbricas en las redes de transmisión de datos ha abierto oportunidades de desarrollo de estas tecnologías, aprovechando la utilización de interfaces aéreas operadas bajo frecuencias no licenciadas [ENT 04].

### 2.1.2.1 ¿Qué es la comunicación inalámbrica?

La comunicación inalámbrica es transmisión de datos sin necesidad de cables, usa frecuencias de radio u ondas infrarrojas y como medio de comunicación ocupa ondas electromagnéticas.

Sus principales ventajas son que permiten una amplia libertad de movimientos y facilita la reubicación de las estaciones de trabajo evitando la necesidad de establecer cableado.

Algunas de las técnicas utilizadas en las redes inalámbricas son: infrarrojos, microondas, láser y radio.

### 2.1.2.2 WIFI

Cuando hablamos de WIFI nos referimos a una de las tecnologías de comunicación inalámbrica más utilizada hoy en día. WIFI es una abreviatura de Wireless Fidelity, también llamada WLAN (wireless lan, red inalámbrica) o estándar IEEE 802.11, que es el homólogo de ethernet para LAN.

En la actualidad podemos encontrarnos con dos tipos de comunicación WIFI:

- 802.11b, que emite a 11 Mb/seg,
- 802.11g, más rápida, a 54 MB/seg.

En comparación con las redes cableadas aun se deben mejorar muchos puntos como los de ancho de banda (las redes cableadas emiten desde los 10MB hasta 1GB o mas) y el alcance de la señal (las redes cableadas lo solucionan con repetidores).

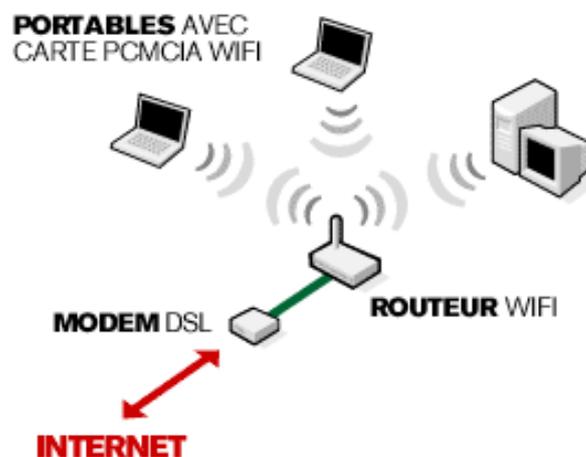


Figura 2.3: Ejemplo de una red WIFI.

La seguridad en el intercambio de datos esta dada gracias a WPA (Wi-Fi Protected Access), el cual es mucho más seguro que su predecesor WEP, basado en el nuevo estándar IEEE 802.11i adoptando nuevas características de seguridad, como la generación dinámica de la clave de acceso. También existe la posibilidad de configurar el punto de acceso para que emita sólo a ciertos dispositivos, usando la MAC de los mismos.

### 2.1.2.3 Bluetooth

Bluetooth forma parte de las tecnologías creadas para proveer comunicación inalámbrica en áreas de uso personal. Es una tecnología desarrollada por Ericsson en 1994, que hace factible la conectividad inalámbrica entre dispositivos a corta distancia, éstos pueden llegar a formar redes con diversos equipos de comunicación: computadoras móviles, radiolocalizadores, teléfonos celulares, PDAs, e, inclusive, electrodomésticos.

Algunas características de esta tecnología:

- **Tecnología inalámbrica:** Su espectro de uso abarca distancias que no exceden los 10 metros, alcanzando velocidades del rango de 1Mbps.
- **Bajo consumo de potencia:** Lo pequeño de los dispositivos y su portabilidad requieren de un uso adecuado de la energía, el cual provee esta tecnología.
- **Bajo costo:** Los dispositivos de comunicación que soporta vienen en una tendencia a bajar de costo. Asimismo, su operación se efectúa bajo una banda de frecuencias no licenciada (2.4GHZ), lo que ayuda a su bajo costo.
- **Integración de servicios:** Puede soportar transmisiones de voz y datos de manera simultánea.
- **Transmisión omnidireccional:** Debido a que su comunicación se basa en radiofrecuencia, no requiere línea de vista y permite configuraciones punto-multi-punto.
- **Seguridad:** Utiliza Spread Spectrum Frequency Hopping como técnica de multiplexaje, lo que disminuye el riesgo de que las comunicaciones sean interceptadas o presenten interferencia con otras aplicaciones. Provee también especificaciones para autenticar dispositivos que intenten conectarse a la red Bluetooth, así como cifrado en el manejo de llaves para proteger la información.
- **Establecimiento de redes:** Tiene la característica de formar redes en una topología donde un dispositivo hace las veces de maestro y hasta siete más operando como esclavos. Esta configuración se conoce como "Piconet". Un grupo de piconets, no más de diez, es referido como "Scatternet".



Figura 2.4: Ejemplo de Piconet.

#### 2.1.2.4 Infrarrojo

La siguiente información esta basada en su mayoría de [www.wikipedia.es](http://www.wikipedia.es), 2006

Las transmisiones de láser de infrarrojo directo envuelven las mismas técnicas empleadas en la transmisión por fibra óptica, excepto que el medio en este caso es el aire libre. El láser tiene un alcance de hasta 16,09 kilómetros, pero las aplicaciones de intercambio de información no pueden ser lo suficientemente robustas a más de 10 metros.

Las redes por infrarrojos permiten la comunicación entre dos nodos, usando una serie de leds infrarrojos para ello. La redes IR (Infrarrojo) permiten conectar dos dispositivos con puertos IR incorporados y ubicados en el mismo espacio o ambiente. Simplemente se alinean los puertos IR de cada dispositivo y se transmiten o envían los datos.

Las ideas a tener presente a la hora de pensar en infrarrojos deben ser: visión directa, pues la luz infrarroja se comporta similar a la luz visible: se refleja en superficies brillantes, pasa a través del vidrio y no atraviesa objetos sólidos opacos. También las distancias deben ser relativamente cortas, esperando velocidades de entre 9600 bps, y 5 Mbps, y comunicación inalámbrica entre dispositivos que pueden requerir de un software (protocolo de comunicación) que garantice su sincronización.

Estos rayos que se usan domésticamente en los controles remotos de los televisores, también se utilizan para redes de computadores con una pequeña luz infrarroja que es muy útil en las transmisiones en distancias cortas, la desventaja es que no debe haber ningún obstáculo entre el emisor y el receptor. Mientras las frecuencias de radio se acercan a las frecuencias de la luz visible se comportan menos como radio y más como luz. La luz infrarroja no se puede usar en exteriores porque el sol las anula.

Para resolver el problema de que la brillantez del sol anula la luz infrarroja, se usan rayos láser en pequeñas distancias. El rayo láser es una luz unidireccional muy potente y coherente (que no se dispersa fácilmente con la distancia).

#### 2.1.2.4.1 Clasificación

De acuerdo al ángulo de apertura con que se emite la información en el transmisor, los sistemas infrarrojo pueden clasificarse en sistemas de corta apertura, también llamados de rayo dirigido o de línea de vista (line of sight, LOS) gran apertura, reflejados o difusos (diffused), recogidos por la norma 802.11.

**Infrarrojo de corta apertura:** Los sistemas infrarrojo de corta apertura, están constituidos por un cono de haz infrarrojo altamente direccional y funcionan de manera similar a los controles remotos de los televisores y otros equipos de consumo: el emisor debe orientarse hacia el receptor antes de transferir información, lo que limita un tanto su funcionalidad. Este mecanismo solo es operativo en enlaces punto a punto exclusivamente. Por ello se considera que es un sistema inalámbrico pero no móvil, o sea que esta más orientado a la portabilidad que a la movilidad.

**Infrarrojo de gran apertura:** Los sistemas de gran apertura permiten la información en ángulo mucho más amplio por lo que el transmisor no tiene que estar alineado con el receptor. Una topología muy común para redes locales inalámbricas basadas en esta tecnología, consiste en colocar en el techo de la oficina un nodo central llamado punto de acceso, hacia el cual dirigen los dispositivos inalámbricos su información, y desde el cual ésta es difundida hacia esos mismos dispositivos. Desgraciadamente la dispersión utilizada en este tipo de red hace que la señal transmitida rebote en techo y paredes, introduciendo un efecto de interferencia en el receptor, que limita la velocidad de transmisión (la trayectoria reflejada llega con un retraso al receptor) Esta es una de las dificultades que han retrasado el desarrollo del sistema infrarrojo en la norma 802.11.



Figura 2.5: Ejemplo de comunicación con dispositivos mediante infrarrojo.

### 2.1.2.5 Antena

Una antena es un dispositivo capaz de emitir o recibir ondas de radio. Está constituida por un conjunto de conductores diseñados para radiar (transmitir) un campo electromagnético cuando se le aplica una fuerza electromotriz alterna [WIK 06].

El tamaño de las antenas está relacionado con la longitud de onda de la señal de radiofrecuencia transmitida o recibida, en general es un múltiplo o submúltiplo exacto de esta longitud de onda. Por eso, a medida que se van utilizando frecuencias mayores las antenas disminuyen su tamaño.

#### 2.1.2.5.1 Parámetros de una antena

Las antenas se caracterizan eléctricamente por una serie de parámetros, los más comunes se mencionan a continuación.

- Ancho de banda: Es el margen de frecuencia de funcionamiento de la antena. Normalmente se da a  $3 \text{ dB}^3$ , es decir, el intervalo entre las frecuencias que las que el nivel

---

<sup>3</sup>**dB**: Unidad relativa empleada en para expresar la relación entre dos magnitudes, acústicas o eléctricas, o entre la magnitud que se estudia y una magnitud de referencia.

de energía radiado cae a la mitad. Varía mucho entre los tipos de antenas, siendo uno de los parámetros que más condicionan su elección.

- **Directividad:** Es la relación entre la potencia radiada en la dirección de máxima radiación y la radiación total de la antena promediada a lo largo del área de la esfera. Este parámetro sólo depende del diagrama de radiación y no de la eficiencia ni potencia radiada.
- **Ganancia:** Es la directividad (en dB) menos las pérdidas en la antena (en dB). Refleja el comportamiento real de la antena al tener en cuenta su geometría a través de la directividad y los materiales que la componen, tanto conductores como dieléctricos, incluidos en las pérdidas.
- **Impedancia de entrada:** Es el parámetro circuital de la antena, o sea, la relación entre el voltaje de entrada de la antena y la corriente de entrada.
- **Anchura de haz:** Es un parámetro de radiación, ligado a la ganancia. Se suele indicar a 3 dB y es el intervalo angular dentro del cual la potencia relativa radiada por la antena es superior a la mitad de la ganancia.
- **Polarización:** La onda electromagnética posee el campo eléctrico vibrando en un plano transversal a la dirección de propagación, pudiendo tener diversas orientaciones sobre el mismo. La polarización de la antena hace referencia a la orientación del campo eléctrico radiado. Así, se puede observar como:
  - Una elipse (onda polarizada elípticamente).
  - Una circunferencia (polarización circular).
  - Una línea recta (polarización horizontal o vertical)

Es importante señalar que, para que una antena "responda" a una onda incidente, tiene que tener la misma polarización que la onda.

#### 2.1.2.5.2 Tipos de antena

- **Antenas Lineales:** Son antenas cuyos elementos radiantes son lineales, es decir, los conductores que las forman tienen una sección de grosor despreciable respecto a la longitud de onda de trabajo y respecto a su longitud física total. Se utilizan extensamente en bajas frecuencias, se pueden utilizar en altas frecuencias y se pueden agrupar en arreglos que trabajan en bandas hasta la UHF. Ejemplos de antenas lineales son:

- El monopolo vertical
- El dipolo y su evolución, la antena Yagi
- La hélice
- Antenas de apertura. Son aquellas que utilizan superficies o aperturas para direccionar el haz electromagnético de forma que concentran la emisión y recepción de su sistema radiante en una dirección, formando ángulos sólidos, la más conocida y utilizada en la actualidad es la antena parabólica, tanto en enlaces de radio terrestres como satelitales, la ganancia de dichas antenas estará relacionada con la superficie de la parábola, a mayor tamaño mayor colimación del haz tendremos y por lo tanto mayor ganancia en una menor apertura angular. El elemento radiante es el Iluminador, el cual puede iluminar en forma directa a la parábola o en forma indirecta mediante un sub reflector, dependiendo del diseño de la misma. El iluminador está generalmente ubicado en el foco de la parábola. Hay 2 tipos de antenas de apertura: De ancho estándar y de apertura sintética (por software).

Ejemplos de antenas de apertura son:

- La antena de bocina
- La antena parabólica
- Superficies reflectoras en general

#### 2.1.2.5.3 Radio Frecuencia

El término radiofrecuencia, también denominado espectro de radiofrecuencia o RF, se aplica a la porción del espectro electromagnético en el que se pueden generar ondas electromagnéticas aplicando corriente alterna a una antena. La tabla 2.2 muestra las bandas del espectro que cubren las frecuencias.

Nombre	Sigla	Frecuencias	Longitud de onda
		Inferior a 3 Hz	> 100.000 km
Extra baja frecuencia (Extremely low frequency)	ELF	3-30 Hz	100.000 km – 10.000 km
Super baja frecuencia (Super low frequency)	SLF	30-300 Hz	10.000 km – 1000 km
Ultra baja frecuencia (Ultra low frequency)	ULF	300–3000 Hz	1000 km – 100 km

Muy baja frecuencia (Very low frequency)	VLF	3–30 kHz	100 km – 10 km
Baja frecuencia (Low frequency)	LF	30–300 kHz	10 km – 1 km
Media frecuencia (Medium frequency)	MF	300–3000 kHz	1 km – 100 m
Alta frecuencia (High frequency)	HF	3–30 MHz	100 m – 10 m
Muy alta frecuencia (Very high frequency)	VHF	30–300 MHz	10 m – 1 m
Ultra alta frecuencia (Ultra high frequency)	UHF	300–3000 MHz	1 m – 100 mm
Super alta frecuencia (Super high frequency)	SHF	3-30 GHz	100 mm – 10 mm
Extra alta frecuencia (Extremely high frequency)	EHF	30-300 GHz	10 mm – 1 mm
		Mayor a 300 GHz	< 1 mm

Tabla 2.2: Bandas de radio con sus frecuencias.

Es importante señalar que las frecuencias entre 430 y 440 MHz corresponden a radiolocalización (radioaficionados). Esto porque la frecuencia a utilizar es 433 MHz para la comunicación.

### Frecuencias de transmisión inalámbrica

- Desde 30MHz hasta 1GHz: Aplicaciones omnidireccionales, intervalo de ondas de radio.
- Desde 2GHz hasta 40GHz: Frecuencias microondas, altamente direccionales, enlaces punto a punto, comunicaciones satelitales.
- Rango de frecuencias comprendido entre  $3 \times 10^{11}$  y  $2 \times 10^{14}$ : Infrarrojos, conexiones locales.

#### 2.1.2.5.4 Propagación de ondas

Existen 4 modos de propagación de ondas, como se muestra en la figura 2.6, y se describen a continuación:

- Onda celeste: También denominada onda de cielo, es lanzada al espacio. Dependiendo de la frecuencia de la onda y del ángulo de lanzamiento, esta puede atravesar la atmósfera y salir al espacio libre, o en caso contrario, puede ser refractada hacia la tierra para ser posteriormente captada por la antena receptora.

- Onda LOS (Line Of Sight) directa: Es el modo de propagación más sencillo ya que sigue una trayectoria recta entre la antena de transmisión y la de recepción.
- Onda reflejada de tierra: Dependiendo del patrón de radiación de la antena involucrada, es posible que parte de la energía de la onda se dirija hacia tierra, a partir de lo cual, por reflexión, cambia su curso para dirigirse finalmente a la antena de recepción.
- Onda de superficie: Su modo de propagación es directamente sobre la tierra, desde el mismo momento de abandonar la antena de transmisión. Esta onda continúa su curso sobre la tierra hasta llegar a su destino final en el sitio de la antena receptora.

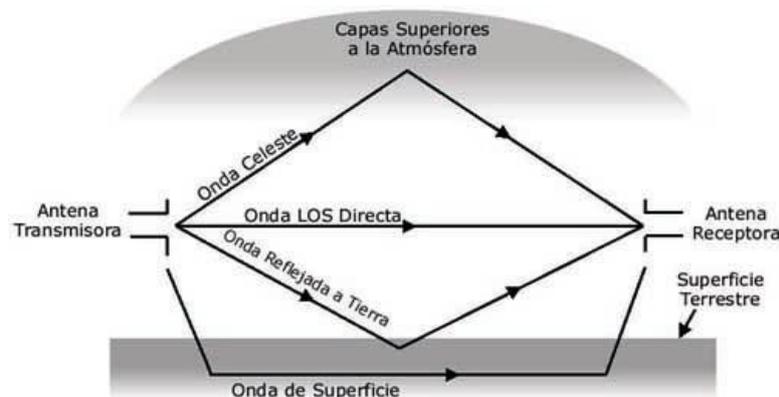


Figura 2.6: Modos de propagación de ondas de radio.

## 2.2 Visión artificial

### 2.2.1 ¿Que es la Visión Artificial?

La Visión Artificial es una herramienta para establecer la relación entre el mundo tridimensional y sus vistas bidimensionales tomadas de él. Por medio de esta teoría se puede hacer, por una parte, una reconstrucción del espacio tridimensional a partir de sus vistas y, por otra parte, llevar a cabo una simulación de una proyección de una escena tridimensional en la posición deseada a un plano bidimensional.

Se puede definir como la ciencia que desarrolla la base teórica y algorítmica mediante la que se extrae y analiza información útil sobre el mundo/entorno, a partir de: una imagen, un conjunto de imágenes o una secuencia de imágenes. Los sensores no ofrecen esta información directamente, hace falta procesado.

### 2.2.2 Historia

Una cámara produce imágenes planas de un mundo físico percibido como tridimensional. Antes de la invención de la fotografía existía un gran interés en representar este mundo 3D en imágenes planas 2D, como es el caso de la pintura.

Los griegos conocieron muchas de las propiedades geométricas que se usan hoy en día. Es el caso de Thales de Mileto y Euclides. Posteriormente los pintores del renacimiento fueron los primeros en entender la formación de imágenes y estudiar la geometría para reproducir correctamente los efectos de la perspectiva en las imágenes que observaban.

A partir de la teoría del plano cartesiano introducida por descartes se empieza a concebir la geometría desde el punto de vista algebraico. Así, las entidades geométricas son descritas como coordenadas y como entidades algebraicas.

En 1826 el francés Niepce lleva a cabo la primera fotografía. Posteriormente el francés Daguerre hizo el primer proceso fotográfico práctico.

Actualmente se utilizan cámaras que emplean lentes para incrementar la potencia de la luz y mejorar el enfoque de la imagen.

### 2.2.3 Problemas

En la visión artificial hay 2 unidades de problemas a solucionar:

- **Unidades de observación.** Se debe interpretar la secuencia de enteros que representa la imagen.
- **Unidades de análisis.** Se debe hacer una descripción de alto nivel: identificar a los objetos, determinar su ubicación y propiedades, identificar las relaciones.

Las unidades de observación no son iguales a las unidades de análisis. Las unidades de observación se componen de la posición y el valor del píxel, mientras que las unidades de análisis presentan la identidad, forma, posición, orientación.

Existen otros problemas adicionales como sombras, brillos, reflejos, ruido, fondo complejo, distorsión de proyección, etc. Estos son problemas difíciles para todos los sistemas de visión.

## 2.2.4 Relación con otros campos

- **Procesamiento digital de imágenes.** Técnicas de transformación de imágenes en las que no hay interpretación, por ejemplo: mejoramiento, compresión y eliminación del ruido. En las figuras 2.7 y 2.8 se muestran ejemplos de rectificación métrica.

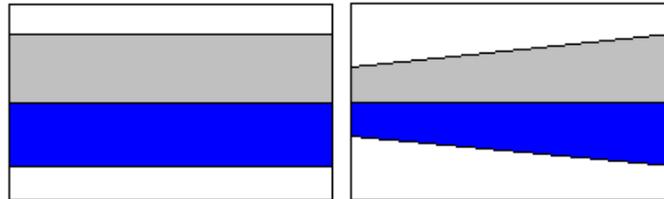


Figura 2.7: Ejemplo de rectificación de perspectiva.

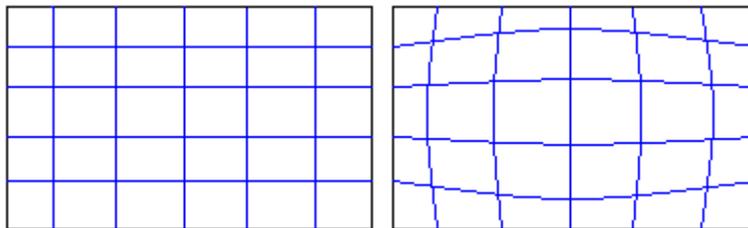


Figura 2.8: Ejemplo de rectificación de distorsión de lente.

- **Computación gráfica.** Generación de imágenes sintéticas a partir de descripciones de la escena. Si se tiene un modelo de la formación de la imagen  $f: 3D \rightarrow 2D$ , es posible entonces simular gráficamente las vistas bidimensionales que se obtendrían de un objeto tridimensional. Las aplicaciones de realidad virtual emplean esta teoría.
- **Reconocimiento de Patrones.** Clasificación de información numérica (estadística) y simbólica (sintáctica).
- **Inteligencia Artificial.** Estudia aspectos computacionales de la inteligencia y e intenta diseñar sistemas que se comporten inteligentemente. En otras palabras es la percepción de la inteligencia artificial.
- **Psicofísica.** Estudia el efecto de los fenómenos o estímulos físicos en el organismo. Se ha estudiado mucho la visión humana.
- **Estimación de movimiento.** Mediante una cámara que toma imágenes de un objeto en movimiento es posible estimar el movimiento del objeto a partir de los puntos de correspondencia en la secuencia de imágenes.

Nuestro proyecto está estrechamente relacionado con el reconocimiento de patrones y la estimación de movimiento.

### 2.2.5 Aplicaciones de la Visión artificial

Las aplicaciones de la visión artificial son tan diversas como:

- Diagnóstico médico: Análisis de imágenes tomadas por rayos x, análisis de imágenes tomadas por ultrasonidos y análisis de sangre.
- Meteorología: Análisis de imágenes satelitales.
- Robótica: Guiado de robots industriales, navegación de robots móviles.
- Agricultura: Análisis de imágenes tomadas por satélites, análisis de plantaciones: crecimiento, enfermedades.
- Seguridad: Vigilancia de edificios, detección de explosivos por rayos x.
- Control de tráfico: Identificación de matrículas de vehículos, control de tráfico diario.
- Identificación: Reconocimiento de caras, identificación automática de huellas dactilares.
- Control de Calidad: Verificación de etiquetas, inspección de contenedores, etc.

### 2.2.6 Niveles de procesamiento

Existen 4 niveles de procesamientos sobre los cuales se puede trabajar una imagen y se mencionan a continuación:

- **Cálculos a nivel píxel:** el resultado se calcula independientemente para cada píxel. Un ejemplo de este nivel es el aumento de contraste.
- **Cálculos a nivel local:** el resultado se calcula a partir de un conjunto de vecinos del píxel en cuestión. Un ejemplo de este nivel es el suavizado.
- **Cálculos a nivel global:** el resultado se calcula a partir de todos los píxeles de la imagen. Estas operaciones hacen que un sistema de visión sea lento. Un ejemplo de esto sería un histograma.
- **Cálculos a nivel objeto:** esto implica identificar los píxeles que pertenecen a un mismo objeto, y luego calcular descriptores utilizando esos píxeles. Esto es el objetivo de la

visión por computador. Un ejemplo de esto es la identificación de objetos por perímetro y por número de píxeles.

## **2.2.7 Funciones de la visión artificial**

La organización de un sistema de la visión de computadora es altamente dependiente del uso. A continuación se presentan las funciones típicas de muchos sistemas de visión por computador.

### **2.2.7.1 Adquisición de la imagen**

Una imagen digital es producida por uno o varios sensores de imagen que, además de las variadas tipos de cámaras fotográficas sensibles a la luz, incluye los sensores de rango, los dispositivos tomográficos, el radar, las cámaras ultrasónicas, etc. Dependiendo del tipo de sensor, la imagen resultante es una imagen 2D ordinaria, un volumen 3D, o una secuencia de imagen. Los valores del píxel corresponden típicamente a la intensidad de luz en uno o varios espectros.

### **2.2.7.2 Pre-procesamiento**

Antes de aplicar cualquier método de visión sobre los datos de la imagen para obtener información específica, es necesario generalmente procesar los datos para asegurar que cumple con los requisitos mínimos para tal efecto. Ejemplos de esto son la eliminación del ruido para no entregar información falsa, el re-muestreo para asegurar que la imagen está correcta y realce de contraste para asegurar que la información relevante de la imagen pueda ser detectada.

### **2.2.7.3 Extracción de características**

Las características de la imagen, en sus diversos niveles de complejidad, se extraen de los datos de la imagen. Los ejemplos típicos de tales características son: líneas y bordes, puntos de interés como esquinas o puntos. Otras características más complejas se pueden relacionar con la textura, la forma o el movimiento.

#### **2.2.7.4 Detección / segmentación**

En este punto se determinan los puntos o las regiones de la imagen que son relevantes para la transformación posterior. Ejemplos de esta etapa son: la selección de un grupo específico de puntos de interés y la segmentación de una o varias regiones de la imagen que contienen un objeto específico del interés.

#### **2.2.7.5 Procesamiento de alto nivel**

En este paso la entrada es típicamente un conjunto de datos pequeño, por ejemplo un conjunto de puntos o una región de la imagen que contiene un objeto específico. Ejemplos de este proceso son: clasificación de un objeto en diversas categorías, verificar si los datos satisfacen un modelo y especificaciones de una aplicación, y valoración de los parámetros de uso, como el tamaño del objeto.

### **2.2.8 Color**

Para poder representar una imagen se utilizan modelos y espacios de color. El uso de cada modelo depende de la aplicación.

#### **2.2.8.1 Modelo de color**

Es un modelo matemático abstracto mediante el que se identifica el color con tuplas de números, por lo general 3 ó 4. A continuación se definen algunos modelos:

##### **2.2.8.1.1 Modelo RYB**

En este modelo el rojo, el amarillo y el azul son los colores primarios, y en teoría, el resto de colores puros o color materia, pueden ser creados mezclando pintura roja, amarilla y azul. La figura 2.9 muestra la representación de este color.

El modelo RYB es utilizado en general en conceptos de arte y pintura tradicionales, y en raras ocasiones usado en exteriores en la mezcla de pigmentos de pintura. Aún siendo usado como guía para la mezcla de pigmentos, el modelo RYB no representa con precisión los colores que deberían resultar de mezclar los 3 colores RYB primarios. En el 2004, la ciencia reconoció que este modelo es incorrecto, pero continúa siendo utilizado habitualmente en arte.



Figura 2.9: Círculo cromático RYB.

#### 2.2.8.1.2 Modelo RGB

La mezcla de colores luz, normalmente rojo, verde y azul (RGB), se realiza utilizando el sistema de color aditivo, también referido como el modelo RGB o el espacio de color RGB (figura 2.10). Todos los colores posibles que pueden ser creados por la mezcla de estas 3 luces de color es aludido como el espectro de color de estas luces en concreto. Al ser mezclados en partes iguales crean el blanco; cuando ningún color luz está presente, uno percibe el negro. Los colores luz tienen aplicación en los monitores de un ordenador, televisiones, proyectores de vídeo y todos aquellos que utilizan combinaciones de fósforos rojos, verdes y azules.

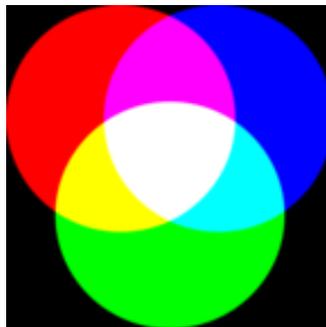


Figura 2.10: Modelo RGB.

#### 2.2.8.1.3 Modelo CMY

Para impresión, los colores usados son cian, magenta y amarillo; este sistema es denominado modelo CMY y se muestra en la figura 2.11. En el modelo CMY, el negro es creado por mezcla de todos los colores, y el blanco es la ausencia de cualquier color, asumiendo que el papel sea blanco. Como la mezcla de los colores es sustractiva, también es llamado modelo de color sustractivo. Una mezcla de cian, magenta y amarillo en realidad resulta en un color negro turbio por lo que normalmente se utiliza tinta negra de verdad. Cuando el negro es añadido, este modelo de color es denominado modelo CMYK. Recientemente, se ha

demostrado que el modelo de color CMY es también más preciso para las mezclas de pigmento.

Se debe tener en cuenta que sólo con unos colores "primarios" ficticios se puede llegar a conseguir todos los colores posibles. Estos primarios son conceptos arbitrarios utilizados en modelos de color matemáticos que no representan las sensaciones de color reales o incluso los impulsos nerviosos reales o procesos cerebrales. En otras palabras, todos los colores "primarios" perfectos son completamente imaginarios, lo que implica que todos los colores primarios que se utilizan en las mezclas son incompletos o imperfectos.

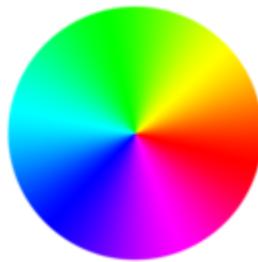


Figura 2.11: Círculo cromático CMY

#### 2.2.8.1.4 Modelo YUV

El modelo **YUV** define un espacio de color en términos de una componente de luminancia y dos componentes de crominancia. El modelo YUV es usado en los sistemas PAL y NTSC de difusión de televisión, el cual es el estándar en la mayoría del mundo.

El modelo YUV está más próximo al modelo humano de percepción que el estándar RGB usado en el hardware de gráficos por computador, pero no tan cerca como los espacios de color HSL y HSV.

Las siguientes ecuaciones se usan para calcular  $Y$ ,  $U$  y  $V$  a partir de  $R$ ,  $G$  y  $B$ :

#### **Conversión RGB a YUV**

$$Y = (0.257 * R) + (0.504 * G) + (0.098 * B) + 16$$

$$Cr = V = (0.439 * R) - (0.368 * G) - (0.071 * B) + 128$$

$$Cb = U = -(0.148 * R) - (0.291 * G) + (0.439 * B) + 128$$

**Conversión YUV a RGB**

$$B = 1.164 (Y - 16) + 2.018 (U - 128)$$

$$G = 1.164 (Y - 16) - 0.813 (V - 128) - 0.391 (U - 128)$$

$$R = 1.164 (Y - 16) + 1.596 (V - 128)$$

Cabe destacar que éste es el modelo de color utilizado, en el desarrollo del presente proyecto.

**2.2.8.1.5 Otros modelos**

Otros modelos son: XYZ (estándar CIE), HSI (Tono, saturación, intensidad), CMY (Cian, magenta, amarillo), HSV (Tono, saturación, valor).

**2.2.8.2 Espacios de color**

Un espacio de color define un modelo de composición del color. Por lo general, un espacio de color lo define una base de N vectores; por ejemplo, el espacio RGB lo forman 3 vectores: Rojo, Verde y Azul; cuya combinación lineal genera todo el espacio de color. Los espacios de color más generales intentan englobar la mayor cantidad posible de los colores visibles por el ojo humano, aunque existen espacios de color que intentan aislar tan solo un subconjunto de ellos.

Existen espacios de color de:

- 1 dimensión: Escala de grises, escala Jet, etc.
- 2 dimensiones: sub-espacio rg, sub-espacio xy, etc.
- 3 dimensiones: espacio RGB, HSV, YCbCr, YUV, Y'I'Q', etc.
- 4 dimensiones: espacio CMYK.

**2.2.8.2.1 Espacio RGB**

RGB es conocido como un espacio de color aditivo porque cuando la luz de dos diferentes frecuencias viajan juntas, desde el punto de vista del observador, estos colores son sumados para crear nuevos tipos de colores. Los colores: rojo, verde y azul fueron escogidos porque cada uno corresponde aproximadamente con uno de los tres tipos de conos sensitivos al color en el ojo humano: 65% sensibles al rojo, 33% sensibles al verde y 2% sensibles al azul. Con

la combinación apropiada de rojo, verde y azul se pueden reproducir muchos de los colores que pueden percibir los humanos, como se muestra en la figura 2.12. Por ejemplo, rojo puro y verde claro producen amarillo, rojo y azul producen magenta, verde y azul combinados crean cian y los tres juntos mezclados a máxima intensidad, crean el blanco. Esta información es de [http://es.wikipedia.org/wiki/Teoría\\_del\\_color](http://es.wikipedia.org/wiki/Teoría_del_color).

Existe también el espacio derivado **RGBA** el cual añade el canal *alpha* (de transparencia) al espacio RGB original.

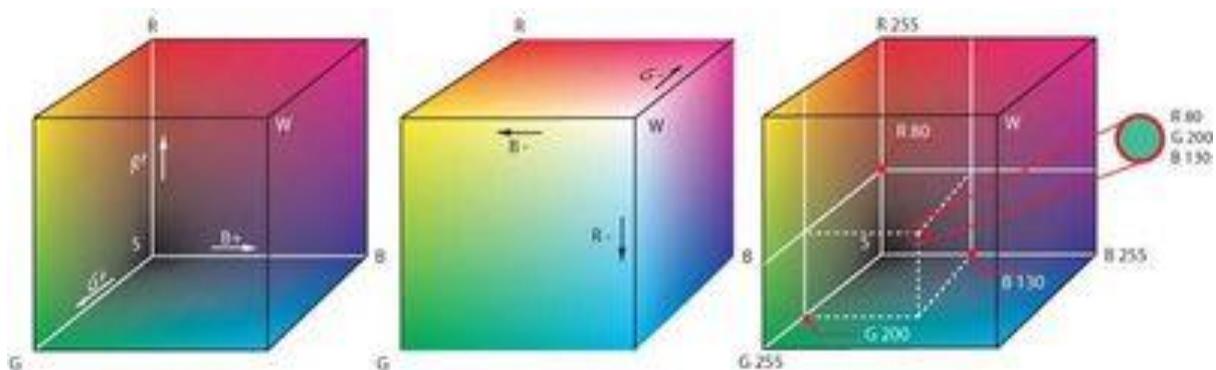


Figura 2.12: Cubo de color RGB.

#### 2.2.8.2.2 Espacio CMY

CMY trabaja mediante la absorción de la luz (colores secundarios).

Los colores que se ven son de parte de la luz que no es absorbida. En CMY magenta más amarillo producen rojo, magenta más cian producen azul, cian más amarillo generan verde y la combinación de cian, magenta y amarillo forman negro (ver figura 2.13). Debido a que el negro generado por la mezcla de colores primarios sustractivos, no es tan denso como el color negro puro. Es por esto que al CMY original se ha añadido un canal clave *key* que normalmente es el canal negro *black* para formar el espacio **CMYK** o **CMYB**. Actualmente las impresoras de cuatro colores, utilizan un cartucho negro además de los colores primarios de este espacio, lo cual genera un mejor contraste. Sin embargo el color que una persona ve en una pantalla de computador difiere del mismo color en una impresora, debido a que los modelos RGB y CMY son distintos. El color en RGB es hecho por la reflexión o emisión de luz, mientras que el CMY mediante la absorción de ésta.

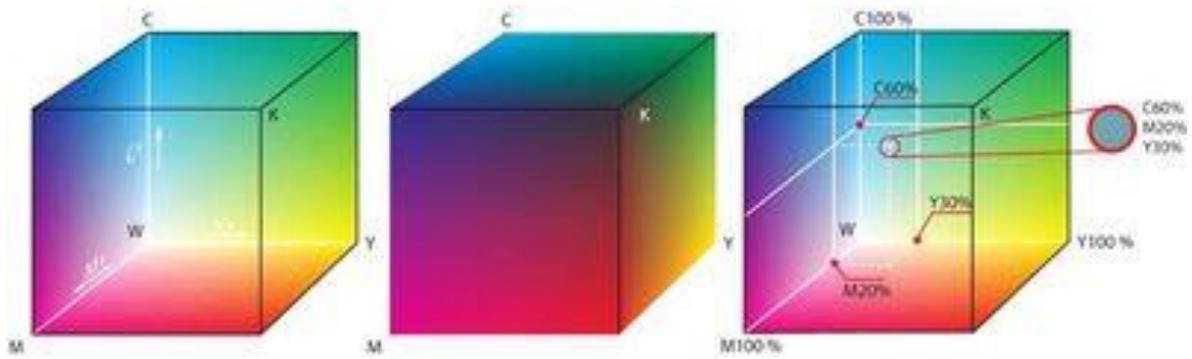


Figura 2.13: Representación de los colores CMYK.

### 2.2.8.2.3 Espacio YIQ

Fue una recodificación realizada para la televisión americana (NTSC), la cual tenía que ser compatible con la televisión en blanco y negro que solamente requiere del componente de iluminación. Los nombres de los componentes de este modelo son Y por iluminación (*luminance*), I fase (*in-phase*) y Q cuadratura (*quadrature*). Estas últimas generan la cromaticidad del color. Los parámetros I y Q son nombrados en relación al método de modulación utilizada para codificar la señal portadora. Los valores de RGB, son sumados para producir una única señal Y' que representa la iluminación o brillo general de un punto en particular. La señal I luego es creada al restar el Y' de la señal azul de los valores RGB originales y luego el Q se realiza restando la señal Y' del rojo. La figura 2.13 muestra como queda el cubo luego de la transformación.

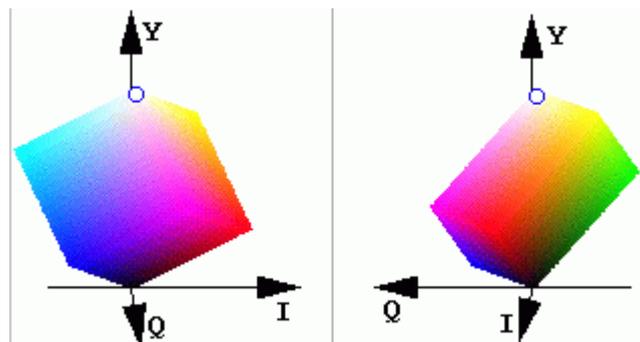


Figura 2.14: Cubo de color YIQ.

### 2.2.8.2.4 Espacio HSB

Es un espacio cilíndrico, pero normalmente asociado a un cono o cono hexagonal, debido a que es un subconjunto visible del espacio original con valores válidos de RGB (figura 2.15).

- **Tonalidad (Hue):** Se refiere a la frecuencia dominante del color dentro del espectro visible. Es la percepción de un tipo de color, normalmente la que uno distingue en un arcoiris, es decir, es la sensación humana de acuerdo a la cual un área parece similar a otra o cuando existe un tipo de longitud de onda dominante. Incrementa su valor mientras nos movemos de forma antihoraria en el cono, con el rojo en el ángulo 0.
- **Saturación (Saturation):** Se refiere a la cantidad del color o a la "pureza" de éste. Va de un color "claro" a un color más vivo (azul cielo – azul oscuro). También se puede considerar como la mezcla de un color con blanco o gris.
- **Brillo (Brightness):** Es la intensidad de luz de un color.

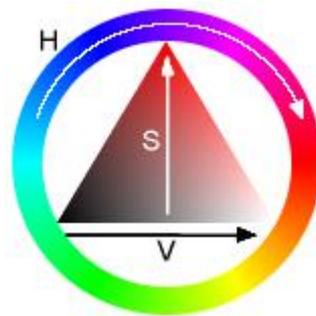


Figura 2.15: Ejes HSB o HSV.

#### 2.2.8.2.5 Otros espacios de color

Otros espacios de color son: HSI (Tono, saturación, intensidad) y YUV.

### 2.2.9 Segmentación de la imagen

La segmentación de la imagen es la separación de las imágenes en regiones u objetos. El nivel de esta división depende de la aplicación, es decir, la segmentación termina cuando se hayan detectado todos los objetos de interés para la aplicación.

La detección de regiones puede parecer algo sencillo, pero existe una serie de dificultades al momento de segmentar, como por ejemplo: ruido, forma, volumen parcial, no homogeneidad, etc.

### 2.2.9.1 Técnicas de segmentación de imágenes

Entre las más utilizadas están: técnicas basadas en bordes, técnicas basadas en regiones y la umbralización. Existen otras técnicas como segmentación de textura, clustering, morfología matemática, redes neuronales, etc.

#### 2.2.9.1.1 Segmentación basada en bordes

Esta técnica se basa en que las regiones están separadas por bordes detectables. Por tanto, se busca detectar esas “fronteras” y a partir de ellas definir las regiones. Los bordes de los objetos se ven en la imagen como discontinuidades de ciertas propiedades: Intensidad, color, textura.

Para segmentar la imagen en función de sus bordes se realizan los siguientes pasos:

- Calcular una imagen de gradiente: primera derivada discreta de la imagen, donde los bordes aparecerán como máximos locales, idealmente. Esto se puede hacer mediante cualquier filtro de gradiente.
- Umbralizar la imagen de gradiente para quedarnos sólo con los puntos que superan un cierto valor de gradiente.
- Los puntos con gradiente alto pueden reforzarse en función del valor de sus puntos vecinos, para obtener zonas de contorno continuo.
- Puede haber zonas donde los bordes desaparecen o presentan discontinuidades. Mediante el seguimiento de los mismos se intenta unir las distintas zonas hasta obtener contornos completos.

Hay 3 formas de detectar los bordes: a partir de la primera derivada (máximos), a partir de la segunda derivada (cruces por cero) y ajustando la imagen a un modelo paramétrico de los bordes.

Para calcular el gradiente o derivada discreta de una imagen existen diversas máscaras espaciales como el filtro de Roberts (2 x 2), de Prewitt y el de Sobel (3 x 3).

$$\text{Roberts: } h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\text{Prewitt: } h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

$$\text{Sobel: } h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 & -2 \\ -1 & 0 & -1 \\ -2 & -1 & 0 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

El filtrado de Laplace emplea la segunda derivada para detectar los bordes. El ruido afecta al doble a este filtro, pero se puede reducir el ruido antes de aplicar el detector de bordes. El filtro ideal para esto es el Gaussiano:

$$\text{Laplace: } \nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\text{Gaussiano: } G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\nabla^2 (G(x, y, \sigma) * f(x, y))$$

$$h(x, y) = c \left( \frac{x^2 + y^2}{\sigma^2} - 1 \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Para aplicar el filtrado laplaciano se debe procesar la imagen con un filtro gaussiano y calcular la transformada de Laplace de la imagen resultante. Luego se detectan los puntos donde la imagen resultante tiene un cruce por cero.

Otros métodos para detectar bordes más complejos son el detector de bordes de Canny y la transformada de Hough.

#### 2.2.9.1.2 Segmentación basada en regiones

Los objetos a segmentar son regiones homogéneas desde algún punto de vista. Se utiliza un criterio de homogeneidad H que cumple cada región individualmente, pero no se cumple en el caso de unión de regiones.

Mencionaremos 3 métodos para segmentar regiones: la unión de regiones, la división de regiones y el crecimiento de regiones. La unión de regiones básicamente consiste en dividir la imagen en muchas regiones pequeñas, por ejemplo por número de píxeles y luego unir las siempre que cumplan un criterio de homogeneidad en la región unida. La división de regiones es lo opuesto, partimos de regiones grandes y mientras no cumpla el criterio de homogeneidad, se divide hasta que las regiones cumplan con el criterio. Las técnicas anteriores generalmente se utilizan unidas. El crecimiento de regiones es una versión de la unión de regiones donde las regiones iniciales de la imagen son cada uno de los píxeles, la segmentación se inicia en un solo punto.

### 2.2.9.1.3 Segmentación de texturas

Para segmentar una imagen de textura buscamos una característica que defina la textura, que tendrá valores diferentes en cada una de las texturas de la imagen.

Para segmentar se calcula una imagen de textura y luego se le aplican las técnicas convencionales de segmentación a esta imagen transformada.

Hay métodos de análisis de texturas que se pueden dividir en:

- Estadísticos

Los métodos estadísticos obtienen medidas a partir del histograma. El histograma es la distribución de niveles de gris y a partir de él se caracterizan las texturas con distribuciones de intensidad concretas. Una vez obtenido el histograma se pueden obtener, entre otros, la media, los momentos del histograma y la entropía.

Para determinar como están distribuidos los niveles de gris en el espacio se utiliza la matriz de co-ocurrencia, que mide la cantidad de píxeles que se encuentran a una determinada distancia y tienen el mismo nivel de gris.

A partir de esta matriz se calculan:

$$\text{Entropía: } H = \sum_{i=1}^N \sum_{j=1}^N P_{ij} \log P_{ij}$$

$$\text{Inercia: } H = \sum_{i=1}^N \sum_{j=1}^N (i - j)^2 P_{ij}$$

$$\text{Energía: } H = \sum_{i=1}^N \sum_{j=1}^N [P_{ij}]^2$$

- Estructurales

Los métodos estructurales suponen que la textura está formada por una estructura que se repite, denominada primitiva. Se intenta detectar esa primitiva y estudiar su distribución.

- Transformadas

Este método utiliza la transformada de Fourier que contiene información de las texturas. A partir de la transformada de Fourier de la región que posee la textura se calculan dos parámetros: la suma de los valores a lo largo de un círculo y a lo largo de un cierto ángulo.

#### 2.2.9.1.4 Segmentación por agrupación de píxeles

Esta técnica, también denominada clustering, clasifica los píxeles estadísticamente, sin tener en cuenta su situación espacial. Es decir, contempla información de las regiones o bordes, sólo información de intensidad de cada punto.

Existen 2 métodos para clasificar los píxeles:

- Método supervisado: se define el número de clases en que se quiere clasificar los píxeles y se seleccionan ciertas muestras de cada clase.
- Método no supervisado: este método es automático ya que el clasificador es quien elige el número de clases.

Para la clasificación se pueden utilizar diversas características de cada punto, como por ejemplo: nivel de gris, bordes, líneas, ángulos, formas.

Existen variados algoritmos de clasificación como Fuzzy C-Means: número de clases prefijado, K-Nearest Neighbor: mayoría de los k-vecinos, Clasificación jerárquica ascendente: tantas clases como píxeles y Clasificación jerárquica descendente: una sola clase para empezar.

El proceso para esta técnica de segmentación es extraer las características y luego hacer la clasificación multivariable automática.

## 2.2.10 Reconocimiento de patrones

Una vez se hayan segmentado los distintos objetos de la imagen viene la etapa de reconocimiento o clasificación de estos objetos.

Los pasos para el clasificar los patrones son: selección de características, diseño del clasificador, entrenamiento del clasificador y evaluación del rendimiento.

Para determinar qué características son las más discriminantes y cuantas se necesitan para una correcta clasificación se utilizan métodos como la relación de Fisher.

En la figura 2.16 se muestra lo que se busca, las funciones discriminantes que nos permitan separar entre las muestras de cada clase.

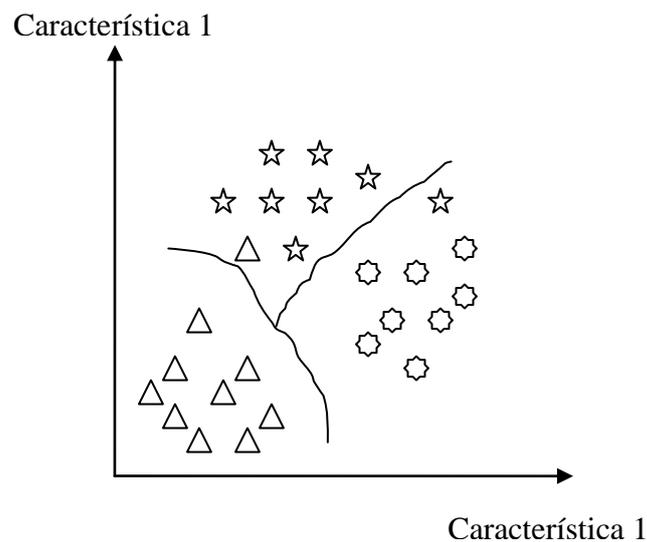


Figura 2.16: Diseño del clasificador.

## 2.3 Inteligencia Artificial

### 2.3.1 Agentes inteligentes

La Inteligencia Artificial Distribuida (IAD) es un subcampo de la Inteligencia Artificial (IA) que trata de los comportamientos inteligentes colectivos, producto de la cooperación de diversas entidades denominadas agentes. [WIKL 06]

La IAD “clásica” estudia las técnicas, coordinación y distribución del conocimiento y acciones en un entorno de múltiples agentes. Tiene dos áreas principales de estudio: la resolución cooperativa de problemas distribuidos (DPS), donde un conjunto de nodos coopera dividiendo y compartiendo el conocimiento de un problema y en el desarrollo de la solución; y los sistemas multiagente (MAS), que estudian la coordinación de la conducta inteligente entre agentes inteligentes autónomos. [WIKL 06]

La IAD “autónoma” y “comercial” se centra en los agentes inteligentes. Podemos distinguir tres áreas: teoría de agentes, que intenta responder a la pregunta qué es un agente y definir los formalismos matemáticos utilizados para razonar y representar las propiedades de los agentes; arquitecturas de agentes, que trata las arquitecturas de software y hardware utilizadas para reflejar las propiedades teóricas de los agentes; y lenguajes de agentes, que son los sistemas de software utilizados para la programación y experimentación con agentes. También podríamos hablar de tipología de agentes, que distingue los diversos campos de aplicación. [WIKL 06]

### **2.3.1.1 ¿Qué es un Agente?**

La pregunta ¿qué es un agente? resulta tan embarazosa como la pregunta ¿qué es la inteligencia? El problema es que, aunque el término sea ampliamente utilizado por mucha gente, desafía los intentos de establecer una definición única y universalmente aceptada. Esto no debe ser necesariamente un problema: si se desarrollan con éxito muchas aplicaciones interesantes y útiles, entonces apenas importa que no se establezca un acuerdo sobre detalles terminológicos potencialmente triviales. Sin embargo, si se intenta definirlo, existe el peligro de que la palabra “agente” pueda convertirse en un término mal empleado. [WIKL 06]

Se pueden encontrar una gran cantidad de definiciones sobre el concepto agente, sin embargo se ha conseguido resumir y extraer las más representativas y con mayor peso en área de la inteligencia artificial:

El Agente según Maes [MAES 95]

“Los agentes autónomos son sistemas computacionales que habitan en entornos dinámicos complejos, percibiendo y actuando autónomamente en ese entorno, y realizan un conjunto de metas o tareas para las que han sido diseñados.”

### El Agente según KidSim [KID 94]

“Definiremos un agente como una entidad software persistente dedicada a un propósito específico. La ‘persistencia’ distingue a los agentes de las subrutinas; los agentes tienen sus propias ideas sobre como ejecutar tareas, sobre sus agendas. Con ‘propósito específico’ se distinguen los agentes de las aplicaciones multifunción, que son típicamente más pequeños.”

### El Agente según IBM [IBMA 06]

”Los agentes inteligentes son entidades software que llevan a cabo un conjunto de operaciones en beneficio de un usuario u otro programa con algún grado de independencia o autonomía, y haciendo esto, emplean algún conocimiento o representación de las metas y deseos del usuario”

### El Agente según Wooldridge y Jennings [AAT 98]

“Es un hardware o más comúnmente un sistema software basado en computador que disfruta de las siguientes propiedades:

- *Autonomía*: los agentes operan sin la intervención directa de personas u otros, y tienen algún tipo de control sobre sus actuaciones y estado interno.
- *Habilidad social*: los agentes interactúan con otros agentes, posiblemente humanos, vía algún tipo de lenguaje de comunicación de agentes.
- *Reactividad*: los agentes perciben el entorno o ambiente, lo que representa la palabra físicamente, un usuario vía una interfaz de usuario, una colección de otros agentes, Internet, o quizás todos ellos combinados, y responde rápidamente a cambios que ocurren en dicho entorno.
- *Pro-actividad*: los agentes no actúan simplemente en respuesta a su entorno, sino que son capaces de exhibir ‘comportamiento dirigido hacia el objetivo’, tomando la iniciativa”

#### 2.3.1.2 Características

La mayoría de los agentes poseen las siguientes características: comunicación, inteligencia y autonomía.

- **Comunicación**: El agente puede comunicarse con el usuario, con otros agentes y con otros programas. Con el usuario se comunica con un interfaz amigable, mediante el que

personaliza sus preferencias. Algunos agentes permiten comunicarse en lenguaje natural, algo típico de los chatbots<sup>4</sup>.

- **Inteligencia:** El grado de inteligencia varía mucho de unos agentes a otros, que suelen incorporar módulos con tecnologías procedentes de la Inteligencia Artificial. Los más sencillos se limitan a recoger las preferencias del usuario, quien debe personalizarlos. Un ejemplo son los agentes inteligentes basados en tecnología de redes neuronales especializados en identificar mensajes de correo electrónico sospechosos de contener spam<sup>5</sup>. En una primera fase el usuario debe marcarlos como spam, el agente va aprendiendo a identificar los rasgos que caracterizan a estos mensajes y posteriormente los filtra.
- **Autonomía.** Un agente no sólo debe ser capaz de hacer sugerencias al usuario sino de actuar. En el ejemplo anterior, el agente que filtra el spam no puede estar continuamente alertando al usuario en cada mensaje de correo que llega sobre la posibilidad de que sea un mensaje no deseado y su verdadera utilidad surge cuando elimina de forma autónoma dichos mensajes.

En la figura 2.17 se puede observar el comportamiento típico de un agente básico. El ambiente le provee de datos de entrada, el cual el agente toma como percepciones a través de sus sensores. Estas percepciones del ambiente pueden ser interpretadas de diferentes maneras, pudiendo aplicar una acción, acorde con la percepción, a través de sus actuadores o efectores.

---

<sup>4</sup> **Chatbots:** Programa computacional creado para desarrollar procesamiento de lenguaje natural, pudiendo éste mantener una conversación con un humano.

<sup>5</sup> **Spam:** Es el hecho de enviar mensajes electrónicos (habitualmente de tipo comercial) no solicitados y en cantidades masivas.

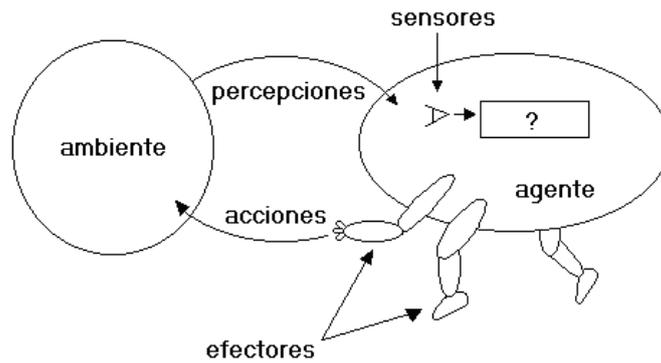


Figura 2.17: Comportamiento básico de una agente inteligente.

### 2.3.1.3 Clasificación.

A continuación, en la tabla 2.3 se especifican las características de los diferentes tipos de agentes, destacando sus ventajas y desventajas aplicadas en ciertas circunstancias dadas.

	CARACTERÍSTICAS	VENTAJAS	DESVENTAJAS	SON RECOMENDABLES CUANDO:
<b>AGENTES REACTIVOS SIMPLES.</b>	<ul style="list-style-type: none"> <li>- Acción-reacción: a un estímulo, el agente responde.</li> <li>- Ignoran acciones históricas, se preocupan de las actuales.</li> </ul>	<ul style="list-style-type: none"> <li>- Fácil de implementar.</li> <li>- Computacionalmente muy económicas.</li> <li>- Rápida capacidad de reacción.</li> </ul>	<ul style="list-style-type: none"> <li>- Inteligencia limitada.</li> <li>- Solo pueden captar percepciones si el entorno es observable.</li> <li>- Ámbito de aplicación limitada.</li> <li>- No son convenientes en ambientes muy dinámicos.</li> </ul>	<ul style="list-style-type: none"> <li>- El entorno es totalmente observable y es en tiempo real.</li> <li>- La tarea es bien específica.</li> </ul>
<b>AGENTES BASADOS EN MODELOS</b>	<ul style="list-style-type: none"> <li>- Poseen un estado interno que depende de la historia percibida.</li> <li>- Actualiza su estado para conocer aspectos no observables.</li> </ul>	<ul style="list-style-type: none"> <li>- Posee información del entorno que no puede ver.</li> <li>- Mantiene un estado interno que refleja alguno de los aspectos no observables.</li> </ul>	<ul style="list-style-type: none"> <li>- Para cada uno de los estados se necesitan acciones distintas.</li> <li>- Su conocimiento no es siempre suficiente para determinar decisiones correctas.</li> </ul>	<ul style="list-style-type: none"> <li>- Existe información del entorno para aplicarle.</li> </ul>
<b>AGENTES BASADOS EN OBJETIVOS</b>	<ul style="list-style-type: none"> <li>- Se basa en conseguir una meta propuesta.</li> <li>- Puede elegir las acciones que logren el objetivo.</li> </ul>	<ul style="list-style-type: none"> <li>- Afina la decisión agregando el elemento de "alcance de meta".</li> <li>- Es más flexible ya que el conocimiento que soporta su decisión está representado explícitamente y puede modificarse y así adaptarse.</li> </ul>	<ul style="list-style-type: none"> <li>- Si no se posee una meta clara los resultados pueden ser insatisfactorios.</li> </ul>	<ul style="list-style-type: none"> <li>- El objetivo es claro.</li> <li>- El ambiente permite que los actuadores del agente logren el objetivo.</li> </ul>
<b>AGENTES BASADOS EN UTILIDAD</b>	<ul style="list-style-type: none"> <li>- Posee una función de utilidad que da un nivel de felicidad al realizar una acción.</li> <li>- Esta función permite tomar decisiones más complejas.</li> </ul>	<ul style="list-style-type: none"> <li>- Torna eficiente una decisión al elegir las acciones que mas convienen dentro de varias posibles, mediante una función de utilidad.</li> </ul>	<ul style="list-style-type: none"> <li>- Hay que estudiar todas las acciones posibles y calcular su utilidad.</li> </ul>	<ul style="list-style-type: none"> <li>- Las metas o resultados requieren de alguna característica adicional, como velocidad, seguridad, confort, detalle, etc.</li> </ul>
<b>AGENTES QUE APRENDEN</b>	<ul style="list-style-type: none"> <li>- Comienzan sin saber mucho del entorno.</li> <li>- Posee elementos de aprendizaje, actuación, críticas y generador de problemas.</li> <li>- Terminan comprendiendo al entorno y usarlo adecuadamente.</li> </ul>	<ul style="list-style-type: none"> <li>- Permite que el agente opere en medios inicialmente desconocidos y que sea más competente que si sólo utilizase un conocimiento inicial.</li> </ul>	<ul style="list-style-type: none"> <li>- Complejo.</li> <li>- Requiere de gran sincronización de sus elementos internos para realizar acciones validas.</li> </ul>	<ul style="list-style-type: none"> <li>- No existe mucha información inicial referente al entorno.</li> <li>- Se requiere de especialización de tareas.</li> <li>- El entorno es muy dinámico.</li> </ul>

Tabla 2.3: Clasificación de los agentes.

## CAPITULO III – ANALISIS Y DISEÑO

## Capítulo III – Análisis y Diseño

### 3.1 Introducción

El presente capítulo consta del desarrollo de los tres módulos que conforman la parte software del sistema completo: módulo de visión, módulo de estrategia y módulo de comunicación con el agente. Su construcción se llevó a cabo mediante una arquitectura de 3 capas, donde se aplicaron los incrementos descritos anteriormente (ver Capítulo 1, 1.4 - Metodología a Utilizar).

En este capítulo se describe el análisis y el diseño de los módulos de visión, estrategia y comunicación, basado en reglas de la categoría Small Size de la competencia Robocup. Para lo anterior se utilizó notación UML.

### 3.2 Planificación y especificación de requerimientos

Esta fase se corresponde con las actividades de especificación de requisitos tradicional, ampliada con una definición de casos de uso de alto nivel, viendo las necesidades del sistema de una manera global y general.

#### 3.2.1 Requerimientos

Los requerimientos del sistema, pueden ser divididos en dos categorías:

- Requerimientos funcionales, los cuales se basan en las reglas de la liga de robots pequeños (también conocida como Liga F180), ya que definen el ambiente sobre el cual el sistema se desenvuelve.
- Requerimientos no funcionales que indican el desempeño necesario para lograr que el sistema sea preciso, robusto y en tiempo real.

Cabe destacar, que a continuación se muestra los requisitos funcionales, no funcionales y planillas combinadas, detallando a qué módulo del sistema pertenece tal análisis.

### 3.2.1.1 Requerimientos Funcionales

Módulo	Ref #	Función	Categoría
Visión	R1	Inicializar el módulo de visión	Evidente
	R2	Capturar imágenes del campo de juego	Evidente
	R3	Segmentar la imagen para filtrar el contenido de colores	Oculto
	R4	Determinar posiciones de los objetos	Evidente
	R5	Determinar dirección del jugador	Evidente
	R6	Generar datos de entrada para el módulo de estrategia	Oculto
Estrategia	R7	Detectar movimientos y su dirección, de jugadores y pelota.	Oculto
	R8	Detectar velocidad de jugadores y pelota.	Oculto
	R9	Escoger la estrategia según indica operador	Evidente
	R10	Determinar dirección y velocidad a tomar por un jugador.	Oculto
	R11	Envío de instrucciones al módulo de comunicación.	Oculto
Comunicación	R12	Inicializar el módulo de comunicación	Evidente
	R13	Enviar datos seriales	Oculto

Tabla 3.1: funcionalidades del sistema

### 3.2.1.2 Requerimientos no funcionales

	Atributos	Detalle y restricciones
Visión	Facilidad de uso	Interfaz de usuario visual fácil de comprender
	Tiempo de respuesta	El proceso de captura debe ser rápido y debe tomar a lo más 1 segundo. Una vez capturada la imagen el proceso de segmentación de la misma no debe superar 1 segundo. El proceso de extracción de datos debe ser inferior a 1 segundo.
	Metáfora de interfaz	Ventanas de imágenes y botones
	Plataformas del Sistema Operativo	Linux (kernel con soporte de video4linux 2)
	Lenguaje de implementación	C++
Estrategia	Facilidad de uso	Ésta viene dada por la simplicidad de la estructura de datos, que serán entregados a este módulo por el módulo de visión.
	Tiempo de respuesta	El proceso completo desde que se obtienen los datos de las imágenes, hasta que el módulo infiere una decisión, no debería superar un segundo.
	Plataformas del Sistema Operativo	Linux
	Lenguaje de implementación	C++

Comunicación	Tiempo de respuesta	<i>El proceso de conexión debe tardar menos de 1 segundo. A su vez, el envío de datos se debe efectuar también en menos de 1 segundo.</i>
	Plataformas del Sistema Operativo	Linux
	Lenguaje de implementación	C++

Tabla 3.2: Atributos del sistema de visión.

### 3.2.1.3 Planilla combinada

	Ref #	Función	Categoría	Atributo
Visión	R1	<i>Inicializar el módulo de visión</i>	<i>Evidente</i>	<i>Metáfora de interfaz Tiempo de respuesta</i>
	R2	<i>Capturar imágenes del campo de juego</i>	<i>Evidente</i>	<i>Metáfora de interfaz Tiempo de respuesta Facilidad de uso</i>
	R3	<i>Segmentar la imagen para filtrar el contenido de colores</i>	<i>Oculto</i>	<i>Tiempo de respuesta</i>
	R4	<i>Determinar posiciones de los objetos</i>	<i>Evidente</i>	<i>Tiempo de respuesta</i>
	R5	<i>Determinar dirección del jugador</i>	<i>Evidente</i>	<i>Tiempo de respuesta</i>
	R6	<i>Generar datos de entrada para módulo de IA</i>	<i>Oculto</i>	<i>Tiempo de respuesta</i>
Estrategia	R7	<i>Detectar movimientos y su dirección, de jugadores y pelota.</i>	<i>Oculto</i>	<i>Facilidad de uso.</i>
	R8	<i>Detectar velocidad de jugadores y pelota.</i>	<i>Oculto</i>	<i>Facilidad de uso.</i>
	R9	<i>Escoger la estrategia según indica operador</i>	<i>Oculto</i>	<i>Facilidad de uso Tiempo de respuesta</i>
	R10	<i>Determinar dirección y velocidad a tomar por un jugador.</i>	<i>Oculto</i>	<i>Tiempo de respuesta.</i>
	R11	<i>Envío de instrucciones al módulo de comunicación.</i>	<i>Oculto</i>	<i>Tiempo de respuesta Facilidad de uso</i>
Com.	R12	<i>Inicializar el módulo de comunicación</i>	<i>Evidente</i>	<i>Tiempo de respuesta</i>
	R13	<i>Enviar datos seriales</i>	<i>Oculto</i>	<i>Tiempo de respuesta</i>

Tabla 3.3: Tabla combinada entre requisitos funcionales y no funcionales del sistema.

### 3.2.2 Actores del sistema

Los actores del sistema sirven para identificar las interacciones que tiene el sistema con el exterior [SCO 01]. Pueden representar:

- Un rol que un usuario puede jugar con respecto al sistema.
- Una entidad, como otro sistema o base de datos que reside fuera del sistema.

En este proyecto, los actores identificados son: la cámara de video, quien provee de información en tiempo real al sistema; el operador que inicia, configura y detiene el sistema; el robot, que es quien recibe la información para actuar en el medio y retroalimentar el sistema (ver figura 3.1).

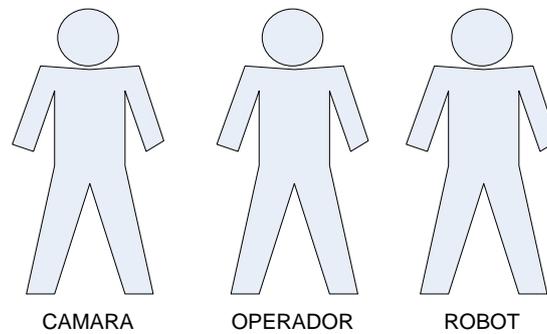


Figura 3.1: Actores del sistema: Cámara de video, operador del sistema y robot.

### 3.2.2.1 Descripción de los casos de uso de alto nivel

<b>Caso de uso</b>	<i>Iniciar sistema de visión</i>
<b>Actores</b>	<i>Operador, Cámara</i>
<b>Tipo</b>	<i>Primario</i>
<b>Descripción</b>	<i>El operador conecta la cámara al computador para capturar las imágenes e inicia el módulo de visión, si la cámara no esta conectada el sistema no se inicia. El módulo lee desde el archivo de segmentación, la última configuración de colores realizada y la carga en memoria, el usuario reconfigura el archivo de segmentación utilizando los rangos de colores según las condiciones de luz actuales.</i>

Tabla 3.4: Caso de uso alto nivel – Iniciar sistema de visión.

<b>Caso de uso</b>	<i>Inicio de una estrategia de movimiento</i>
<b>Actores</b>	<i>Operador</i>
<b>Tipo</b>	<i>Primario</i>
<b>Descripción</b>	<i>Luego de iniciado el sistema, el operador debe escoger entre iniciar una estrategia automática o una manual. Si escoge una automática, debe escoger una de las estrategias implementadas en el sistema. Una ejecución automática muestra la visión artificial en tiempo real y una manual solo sirve para probar la comunicación y los motores de los robots.</i>

Tabla 3.5: Caso de uso alto nivel – inicio de una estrategia de movimiento.

<b>Caso de uso</b>	<i>Captura de imágenes</i>
<b>Actores</b>	<i>Cámara</i>
<b>Tipo</b>	<i>Primario</i>
<b>Descripción</b>	<i>Una vez iniciada cualquier estrategia automática, la cámara tiene vital importancia. Esta comienza una constante captura de imágenes (30 fps) para que estas sean analizadas por la estrategia y en base a eso, determinar una acción.</i>

Tabla 3.6: Caso de uso de alto nivel – captura de imágenes.

<b>Caso de uso</b>	<i>Envío de acción</i>
<b>Actores</b>	<i>Operador</i>
<b>Tipo</b>	<i>Módulo de estrategia y dispositivo de comunicación serial</i>
<b>Descripción</b>	<i>Si el usuario escoge una estrategia manual, entonces no hay intervención de ningún otro módulo, más que el de comunicación. El operador puede configurar un set de datos a enviar al robot, el módulo envía a través del puerto serial los datos a la BS2, dando movimiento al robot.</i>

Tabla 3.7: Caso de uso de alto nivel – Envío de acción.

### 3.2.2.2 Descripción de los casos de uso expandidos

<b>Nombre del caso de uso</b>	<i>Iniciar el sistema de visión</i>	
<b>Propósito</b>	<i>Tener el sensor (cámara) dispuesto para su uso en el sistema en general y configurar la calibración de colores.</i>	
<b>Tipo</b>	<i>Primario</i>	
<b>Referencia cruzada</b>	<i>R1</i>	
<b>Resumen</b>	<i>El operador conecta la cámara para la captura de imágenes e inicia el módulo de visión, configurando el archivo de</i>	
<b>Actores</b>	<i>Operador, cámara.</i>	
<b>Actor</b>		<b>Sistema</b>
<i>1. El caso de uso comienza cuando el operador conecta la cámara al</i>		
<i>2. El operador inicia la configuración del módulo de visión con una interfaz gráfica.</i>		
		<i>3. El módulo lee la configuración desde el archivo de segmentación</i>
<i>4. El operador configura el filtro de los colores.</i>		<i>5. Se almacena la configuración para poder ser utilizada, actualizando el archivo de segmentación.</i>
<i>6. El operador inicia el módulo de visión artificial mediante interfaz gráfica.</i>		<i>7. El módulo valida la inicialización mostrando la imagen capturada o el mensaje de error.</i>

Tabla 3.8: Caso de uso expandido – Iniciar sistema de visión.

<b>Nombre del caso de uso</b>	<i>Inicio de estrategia de movimiento.</i>	
<b>Propósito</b>	<i>Dar a conocer la aplicación de las diferentes estrategias automáticas programadas.</i>	
<b>Tipo</b>	<i>Primario</i>	
<b>Referencia cruzada</b>	<i>R9</i>	
<b>Resumen</b>	<i>El operador escoge una de las estrategias programadas e inicia el modo automático.</i>	
<b>Actores</b>	<i>Operador.</i>	
<b>Actor</b>		<b>Sistema</b>
<i>1. El operador escoge una de las estrategias programadas en el sistema.</i>		
<i>2. El operador inicia la estrategia presionando “iniciar” en la interfaz.</i>		<i>3. El sistema refresca la simulación 2D y comienza la estrategia escogida.</i>

Tabla 3.9: Caso de uso expandido – Inicio de estrategia de movimiento.

<b>Nombre del caso de uso</b>	<i>Captura de imágenes.</i>	
<b>Propósito</b>	<i>Proporcionar datos de entrada al módulo de estrategia.</i>	
<b>Tipo</b>	<i>Primario</i>	
<b>Referencia cruzada</b>	<i>R2, R3, R4, R5, R6</i>	
<b>Resumen</b>	<i>Las imágenes capturadas desde el dispositivo de captura son procesadas extrayéndole información relevante y enviándola al módulo de estrategia.</i>	
<b>Actores</b>	<i>Cámara</i>	
<b>Actor</b>	<b>Sistema</b>	
<i>1. La cámara realiza la captura de imágenes.</i>	<i>2. Obtiene las imágenes y realiza segmentación sobre estas.</i>	
	<i>3. Determinar posiciones de los objetos</i>	
	<i>4. Generar datos de entrada para el módulo de estrategia</i>	

Tabla 3.10: Caso de uso expandido – Captura de imágenes.

<b>Nombre del caso de uso</b>	<i>Envío de acción.</i>	
<b>Propósito</b>	<i>Enviar los datos correctamente a la BS2 a través del puerto serial</i>	
<b>Tipo</b>	<i>Primario</i>	
<b>Referencia cruzada</b>	<i>R12, R13</i>	
<b>Resumen</b>	<i>El módulo debe ser capaz de enviar datos recibidos del módulo de estrategia a la BS2 para que esta los envíe al robot y se interpreten como instrucciones.</i>	
<b>Actores</b>	<i>Robot</i>	
<b>Actor</b>	<b>Sistema</b>	
	<i>1. Obtiene los datos y los envía a la BS2 a través del puerto serial.</i>	
<i>2. El robot recibe los datos, que son enviados desde el puerto serial del computador, a través de una antena de radio frecuencia.</i>		

Tabla 3.11: Caso de uso expandido – Envío de acción.

### 3.2.2.3 Diagrama de caso de uso

La figura 3.2 muestra el diagrama de casos de uso del sistema:

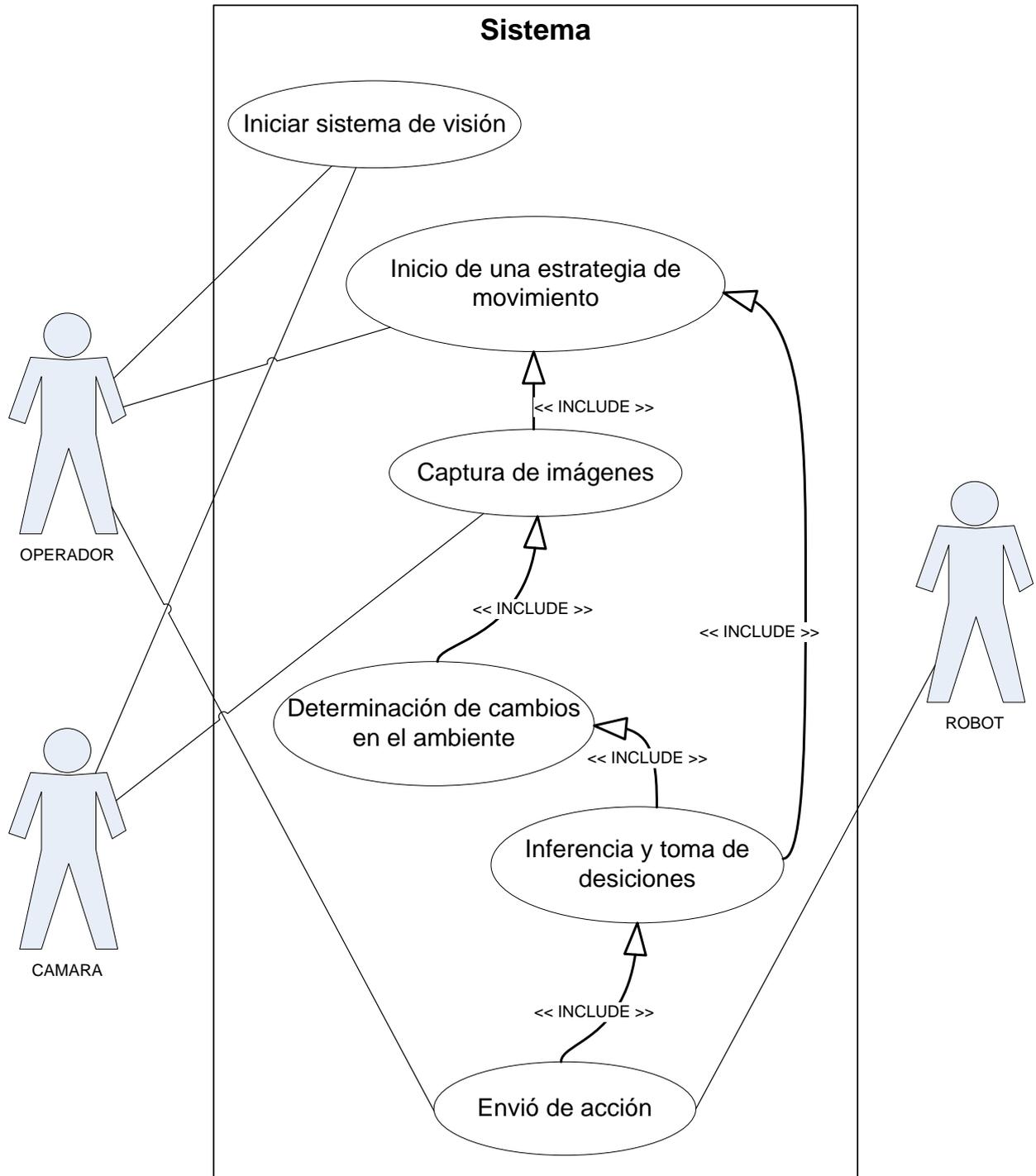


Figura 3.2: Diagrama de casos de uso del sistema

### 3.2.2.4 Planificación de Casos de uso según Ciclos de Desarrollo

A continuación, en la figura 3.3, se especifica la forma en que se dividieron las tareas de construcción de los casos de uso, por cada uno de los 3 incrementos propuestos en el desarrollo de este trabajo:

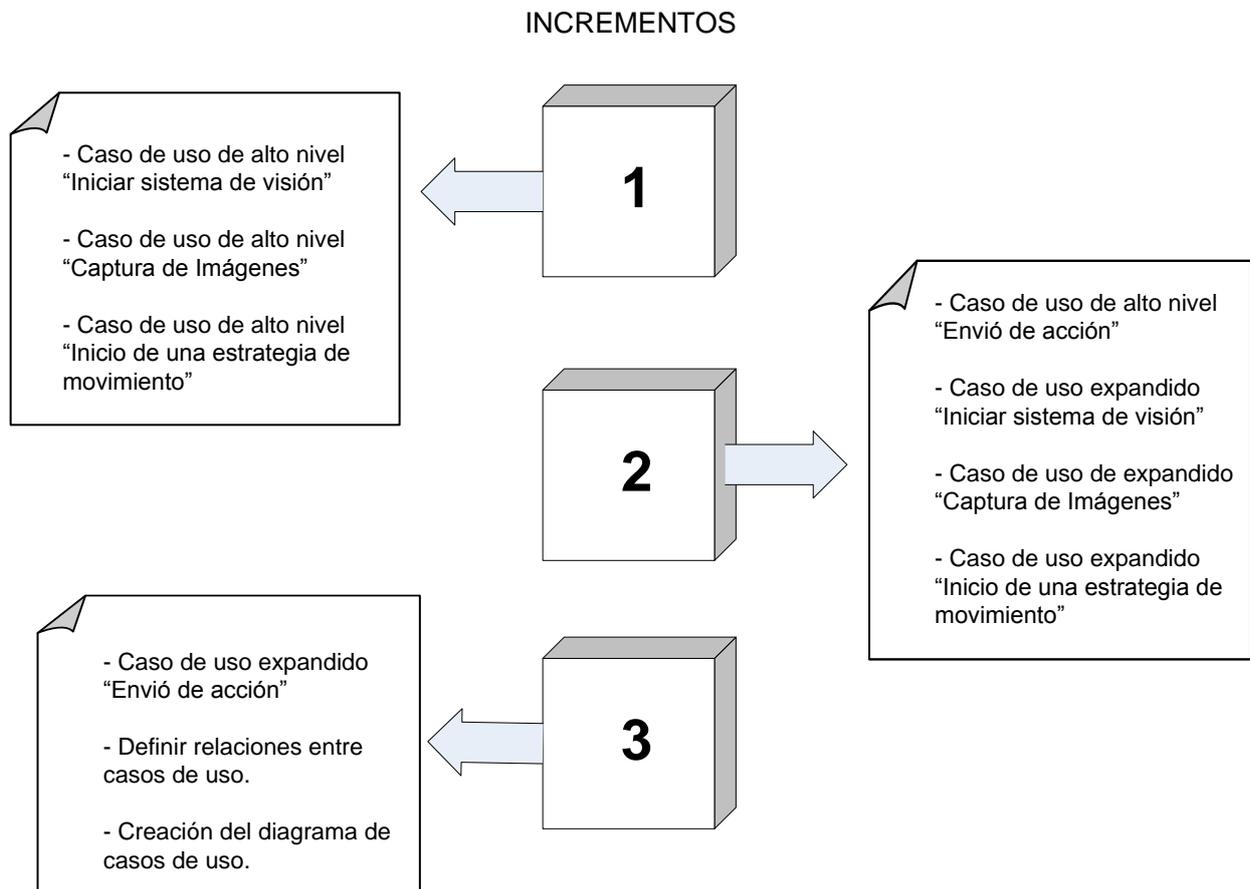


Figura 3.3: Detalle de la división del trabajo en la creación de casos de uso, para los 3 incrementos propuestos.

## 3.3 Análisis del Sistema

En la fase de análisis de un ciclo de desarrollo se investiga sobre el problema, sobre los conceptos relacionados con el subconjunto de casos de uso que se esté tratando. Se intenta llegar a una buena comprensión del problema, sin entrar en cómo va a ser la solución en cuanto a detalles de implementación, sino más a los conceptos que describen lo que se quiere hacer.

### 3.3.1 Modelo conceptual

En el modelo conceptual se tiene una representación de conceptos del mundo real, no de componentes software. El objetivo de este diagrama es aumentar la comprensión del problema, reflejando conceptos y relaciones básicas del sistema a implementar.

Conceptos identificados en el dominio del problema:

- *Cámara*: indispensable actor y proveedor de la entrada del sistema. Captura las imágenes del ambiente.
- *Agente*: el agente es el coordinador del sistema, solicitando información de entrada desde la cámara y la estrategia, para luego entregar una decisión de acción al robot.
- *Estrategia de juego*: Dependiendo de la disposición de los robots en la imagen, se aplica una de las estrategias de juego programadas, se crea una acción y se le pasa al agente.
- *Robot*: recibe vía inalámbrica los acción que se decidió. Solo recibe órdenes y las ejecuta.

En la figura 3.4 se puede ver el modelo conceptual del sistema:

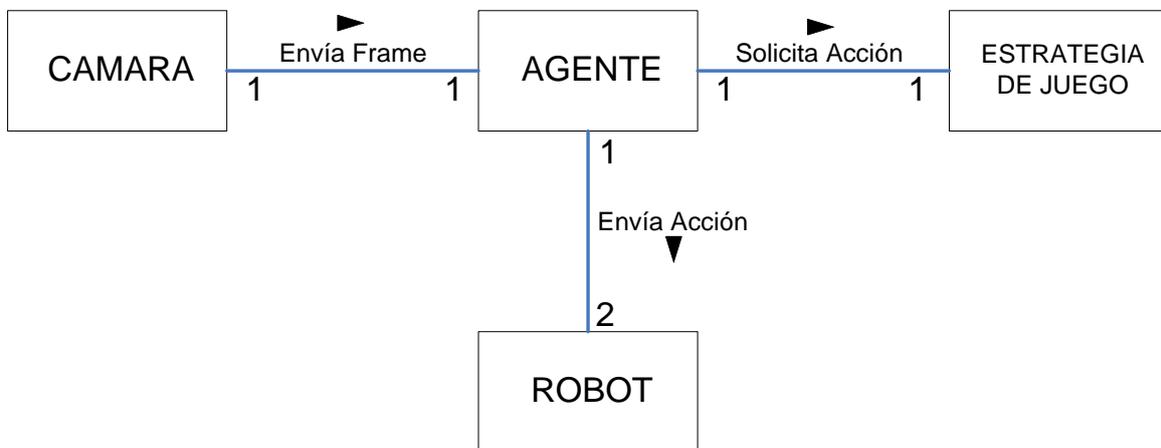


Figura 3.4: Modelo conceptual de sistema.

### 3.3.2 Diagramas de secuencia del sistema

Los diagramas de secuencia de las figuras 3.5 a la 3.8 muestran el comportamiento del sistema, visto éste como una caja negra.

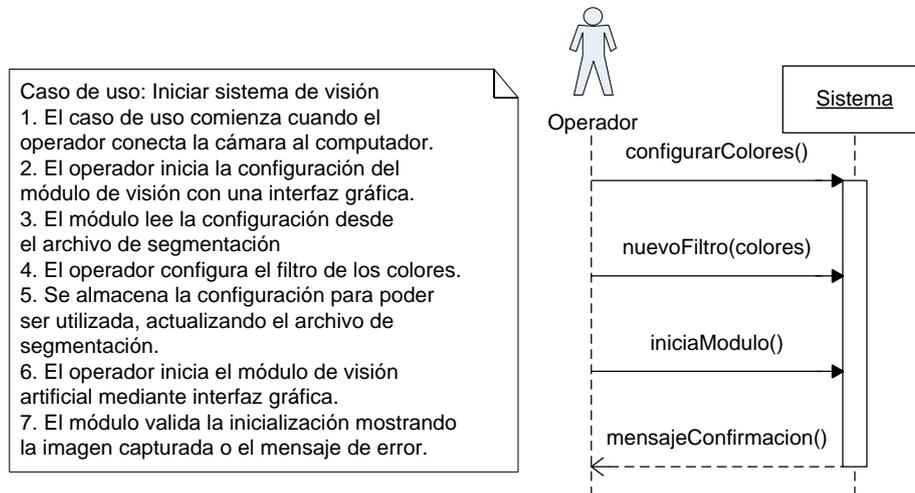


Figura 3.5: Diagrama de secuencia – Iniciar sistema de visión

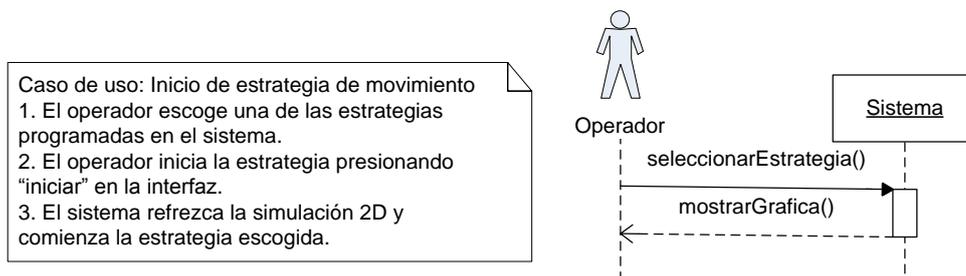


Figura 3.6: Diagrama de secuencia – Inicio de estrategia de movimiento

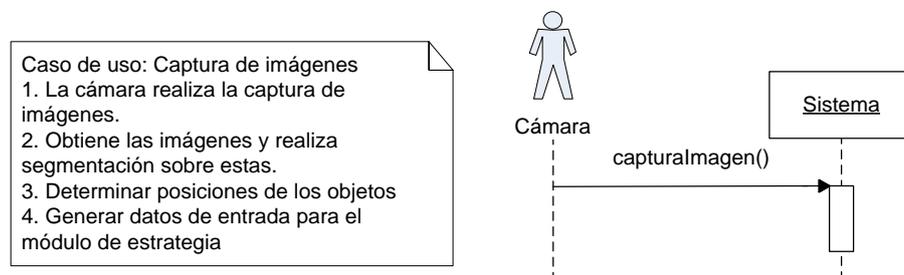


Figura 3.7: Diagrama de secuencia – Captura de imágenes

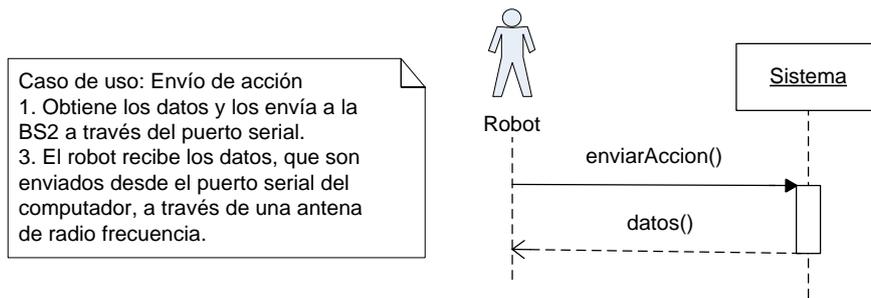


Figura 3.8: Diagrama de secuencia – Envío de acción

### 3.4 Diseño del Sistema

En esta etapa se presenta una solución lógica para satisfacer los requerimientos del sistema. Aquí, se muestran los casos de uso reales del sistema, el diagrama de paquetes, el diagrama arquitectónico y los diagramas de clases del sistema, tanto en su versión simplificada como completa. Los diagramas de clases serán divididos por módulos para una mejor comprensión de los mismos.

#### 3.4.1 Casos de uso reales

En la figura 3.9 se muestra la pantalla para el monitoreo del sistema.

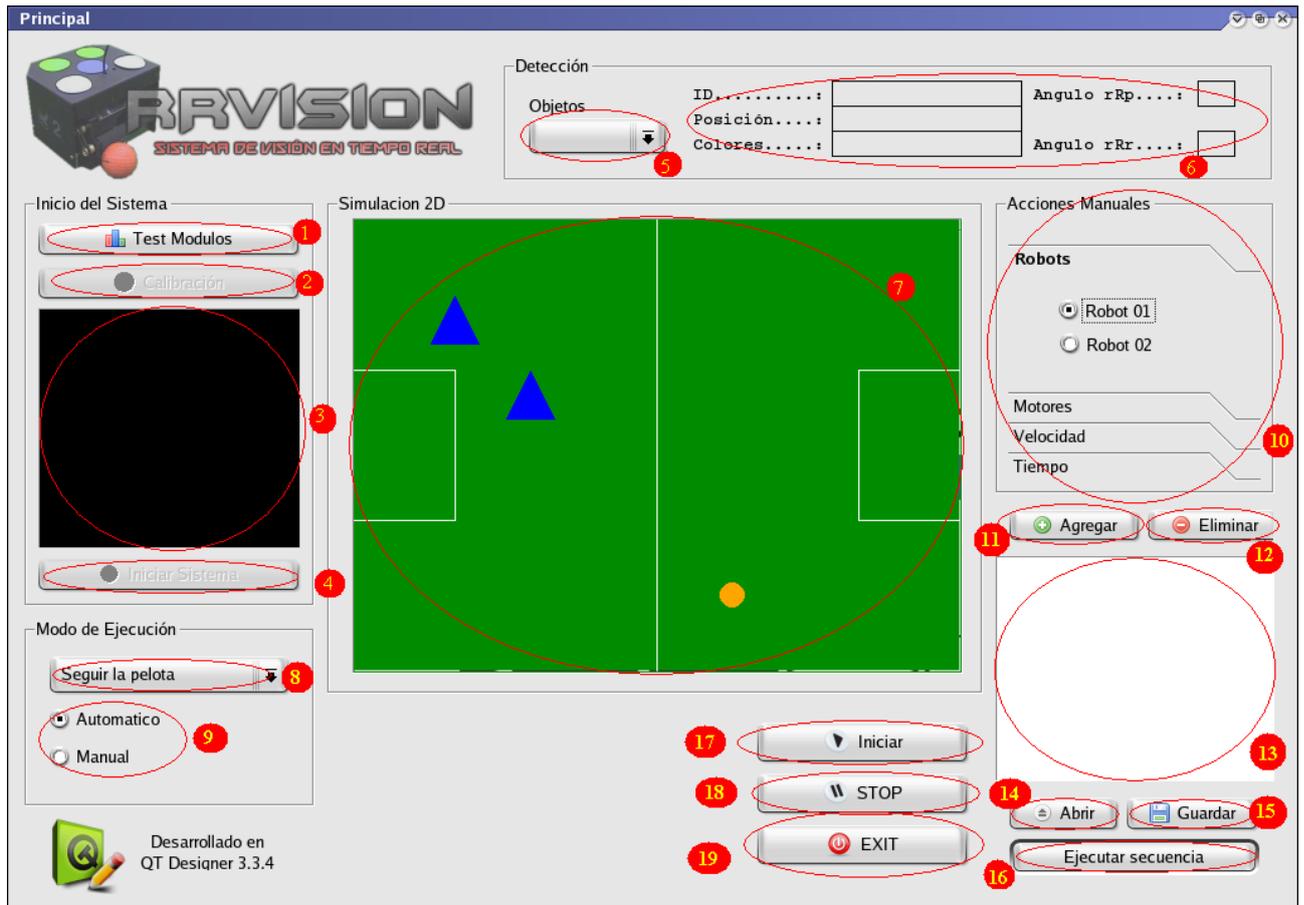


Figura 3.9: Pantalla principal.

A continuación, en la tabla 3.12 se muestran los casos de uso reales del sistema.

Nro	Componente	Función
1	Botón "Test módulos"	Verifica que el módulo de captura este listo
2	Botón "Calibración"	Inicia una ventana de configuración o calibración de los colores
3	Área de texto "Detalle de Ejecución"	Muestra información de las acciones realizadas
4	Botón "Iniciar sistema"	Inicia el módulo de visión. Carga la configuración
5	Lista desplegable "Objetos"	Selecciona el objeto a detallar
6	Campos de texto	Muestran información de cada objeto de la cancha.
7	Campo de juego	Simulación 2D de la cancha de fútbol
8	Lista desplegable "Estrategias"	Selecciona la estrategia a utilizar
9	Opción "Manual - automático"	Corresponde al modo de envío de instrucciones
10	Lista "Acciones manuales"	Corresponde a los datos de la instrucción: robot a mover, dirección y velocidad de los motores, tiempo de ejecución de la acción
11	Botón "Agregar"	Agrega una instrucción a la lista
12	Botón "Eliminar"	Elimina una instrucción de la lista
13	Area de texto	Muestra las instrucciones disponibles
14	Botón "Abrir"	Carga instrucciones desde un archivo de texto

15	Botón “Guardar”	Almacena la lista de instrucciones en un archivo de texto
16	Botón “Ejecutar secuencia”	Ejecuta la secuencia mostrada en el área de texto (13)
17	Botón “Iniciar”	Inicia el modo automático o envía la instrucción manualmente, esto depende de la opción seleccionada en 9
18	Botón “Detener”	Detiene el modo automático
19	Botón “Exit”	Termina los módulos y cierra la aplicación

Tabla 3.12: Casos de uso reales del sistema.

### 3.4.2 Diagrama Arquitectónico

Se utilizó una arquitectura de tres niveles, como se muestra en la figura 3.10, que contempla además de la interfaz de usuario, la persistencia de los datos.

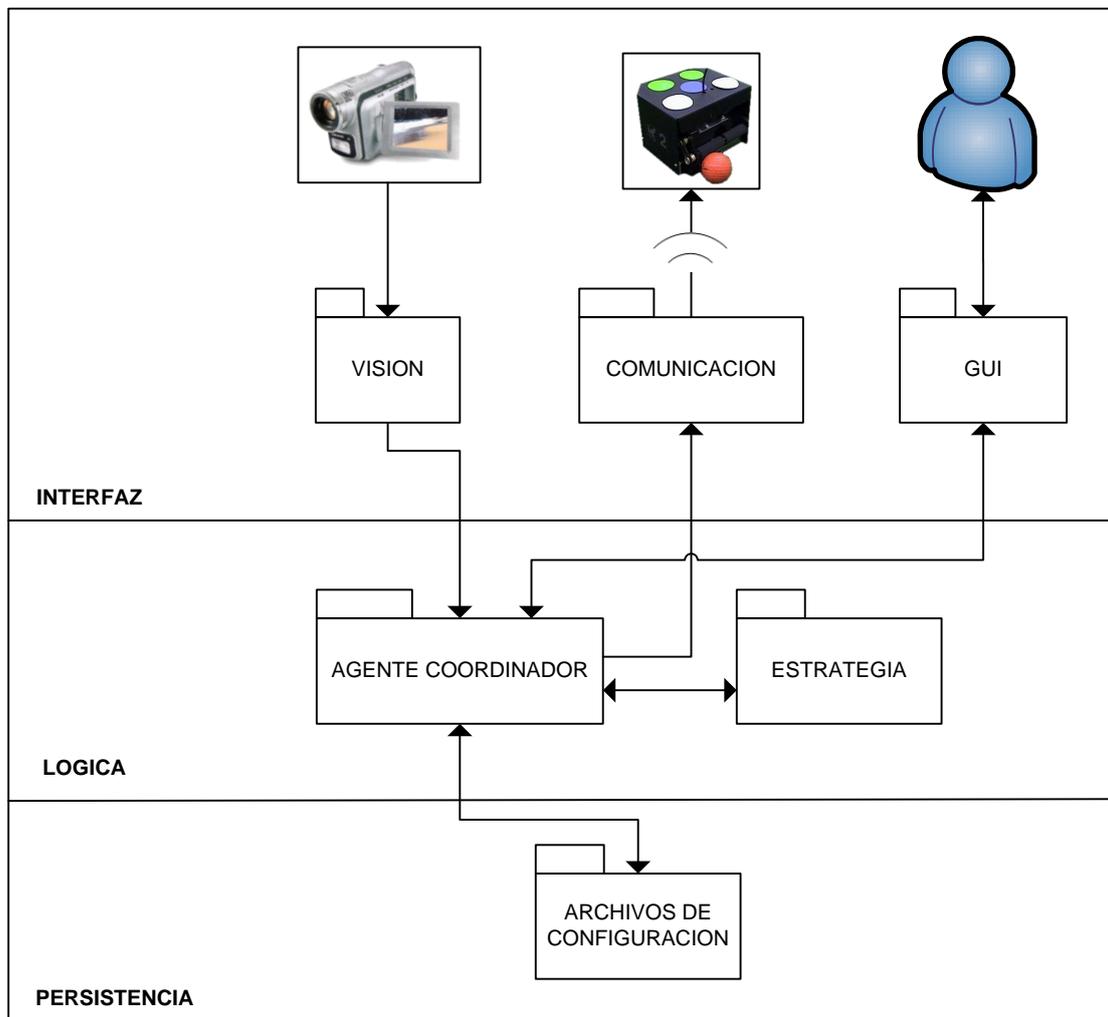


Figura 3.10: Diagrama arquitectónico del sistema.

### 3.4.3 Diagrama de paquetes

El diagrama de la figura 3.11 muestra la subdivisión hecha por módulos o paquetes. Los paquetes más importantes son Visión, Estrategia y Comunicación.

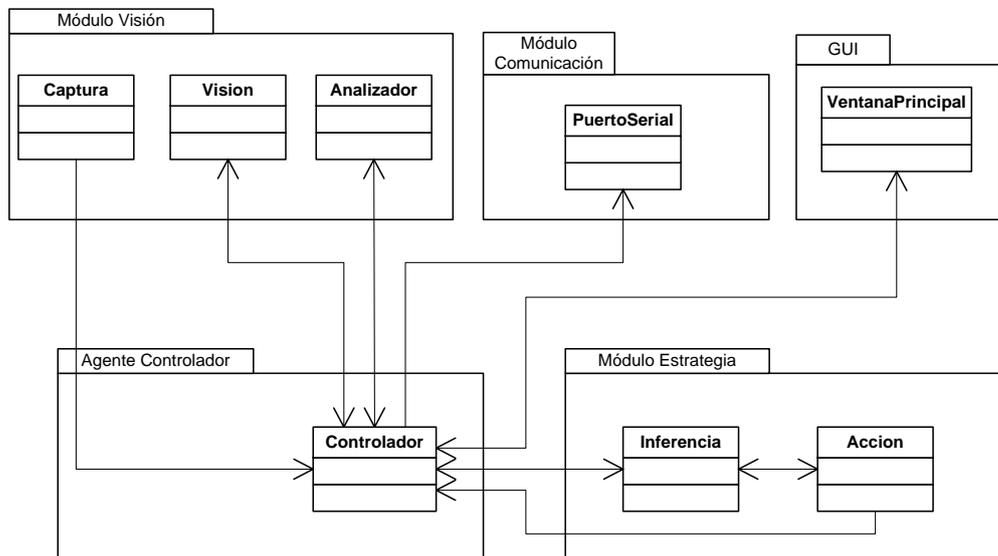


Figura 3.11: Diagrama de paquetes del sistema.

### 3.4.4 Diagramas de clases por módulos

Los diagramas de clases se dividen en paquetes o módulos y se presentan a continuación. Estos diagramas representan la especificación para las clases software de la aplicación.

#### 3.4.4.1 Diagrama de clases simplificado módulo visión

El diagrama de clases simplificado se muestra en la figura 3.12.

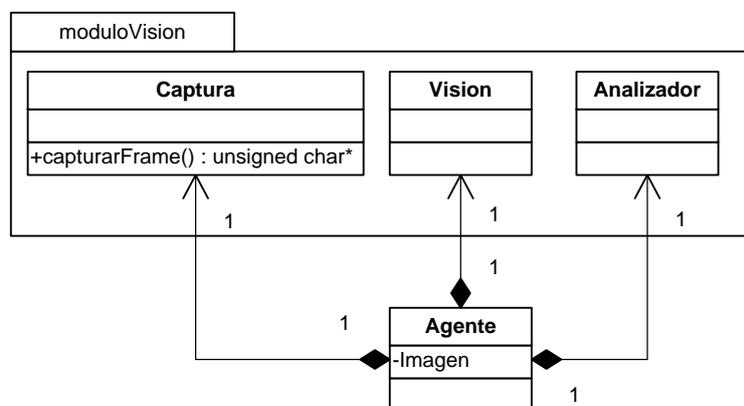


Figura 3.12: Diagrama de clases simplificado - Módulo de visión.

### 3.4.4.2 Diagrama de clases completo módulo visión

La versión completa del diagrama de clases del módulo de visión se presenta en la figura

3.13.

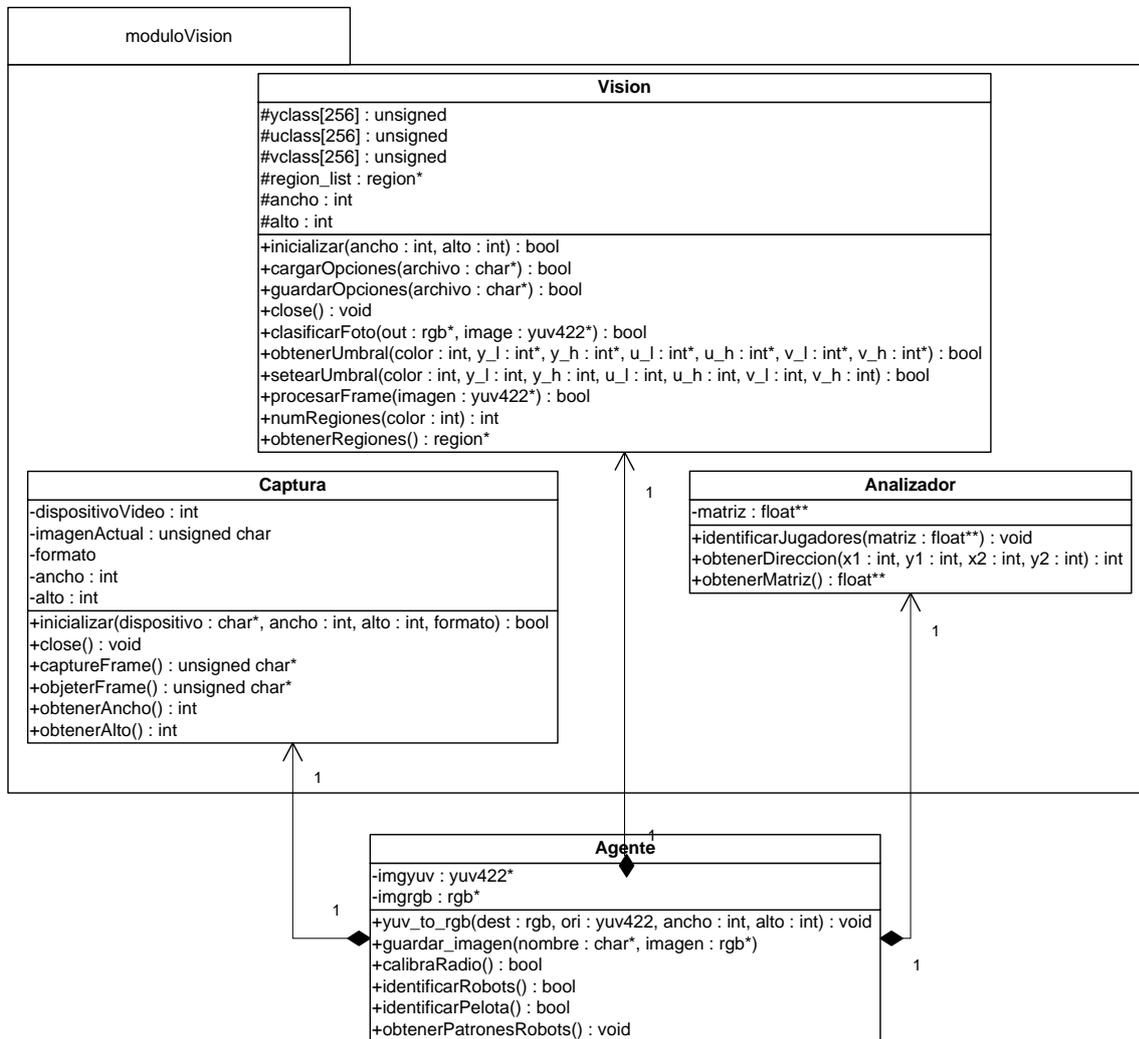


Figura 3.13: Diagrama de clases completo - Módulo de visión.

### 3.4.4.3 Diagrama de clases simplificado módulo estrategia

La figura 3.14 muestra el diagrama de clases en su versión simplificada.

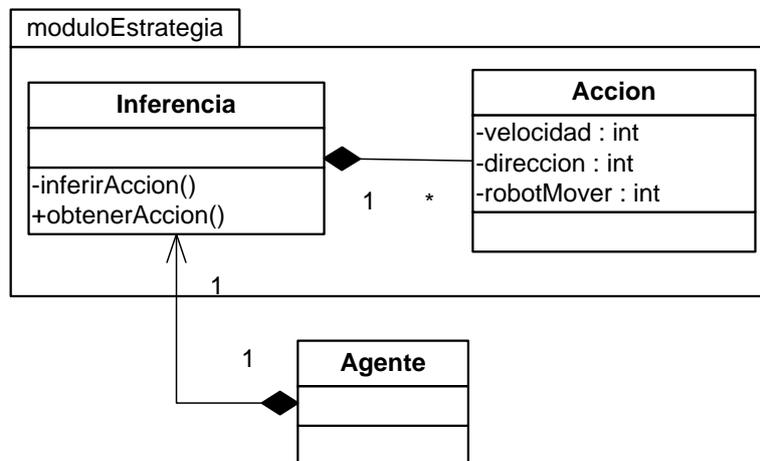


Figura 3.14: Diagrama de clase simplificado – Módulo de estrategia.

### 3.4.4.4 Diagrama de clases completo módulo estrategia

La figura 3.15 muestra el diagrama de clases del módulo de estrategia en su versión completa. Como se puede ver la clase Inferencia tiene como métodos las 3 estrategias a implementar.

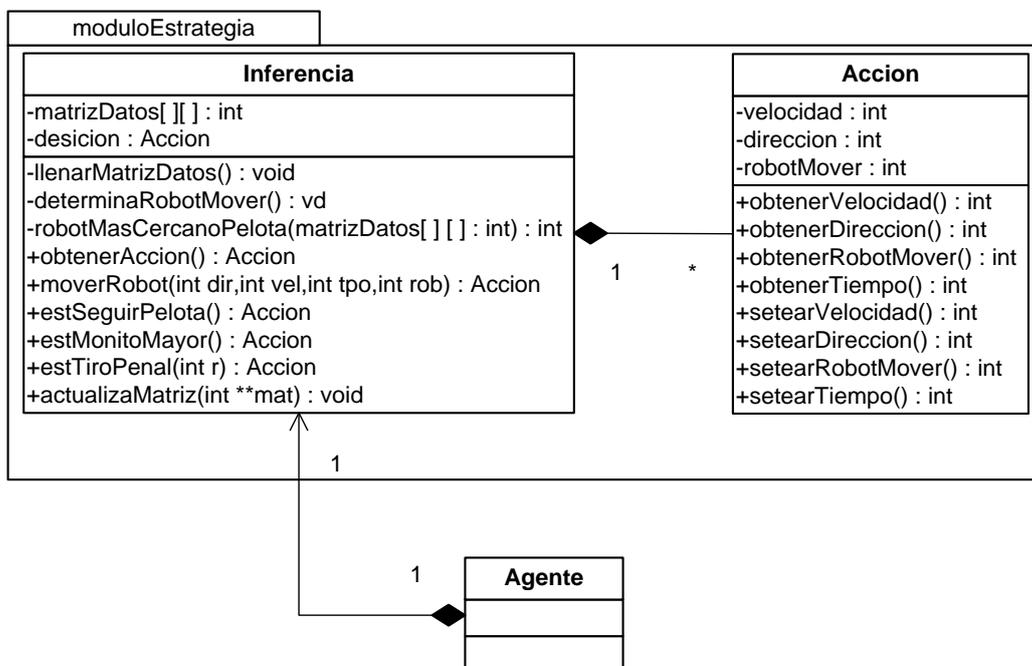


Figura 3.15: Diagrama de clases completo – Módulo de estrategia.

### 3.4.4.5 Diagrama de clases simplificado módulo comunicación

El diagrama de clases simplificado de la comunicación serial se muestra en la figura 3.16.

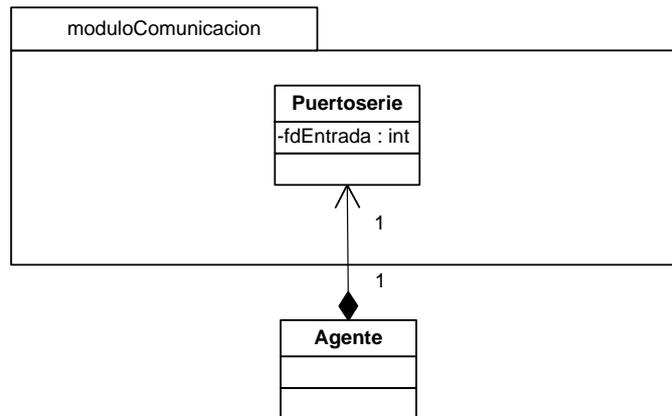


Figura 3.16: Diagrama de clases simplificado - Módulo de comunicación.

### 3.4.4.6 Diagrama de clases completo módulo comunicación

El diagrama de clases del módulo de comunicación en su versión completa se presenta en la figura 3.17.

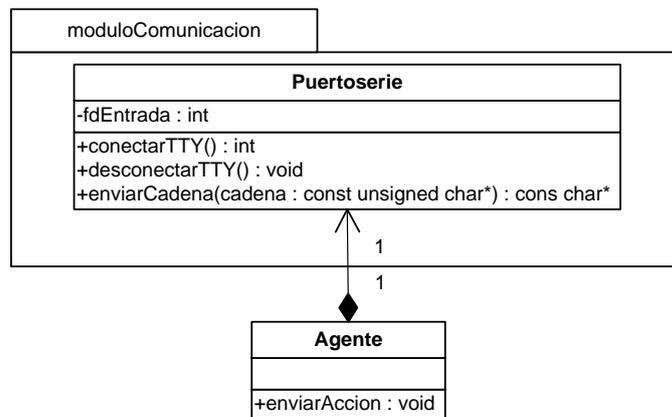


Figura 3.17: Diagrama de clases completo - Módulo de comunicación.

## **CAPÍTULO IV – IMPLEMENTACION Y PRUEBAS**

## Capítulo IV – Implementación y Pruebas

### 4.1 Introducción

El presente capítulo da a conocer detalles de la implementación del sistema propuesto, tanto hardware como software. Además se incluyen las pruebas realizadas a ambas áreas (hardware y software), terminando con ello el desarrollo del sistema a través de las etapas de la metodología escogida.

La estructura del presente capítulo consta primero del desarrollo hardware del sistema, como son los robots y la estructura que sostiene la visión. Posteriormente el detalle del desarrollo software del sistema como lo son los módulos de visión, comunicación y estrategia. Por último, se encuentran las pruebas realizadas a los módulos y sus resultados.

### 4.2 Construcción del robot jugador

Este punto explica la construcción de los agentes robóticos, los cuales se desenvolverán en un ambiente muy similar al de la competencia RoboCup, esto es: sobre una alfombra verde demarcada por líneas blancas.

Por otro lado los robots serán construidos bajo las normas indicadas por la competencia, en la categoría F180, con excepción del dribbling, el cual no fue implementado por problemas que se indican en el punto 4.2.1.4.

#### 4.2.1 Características del ambiente y los robots según Robocup

Las características del ambiente, es decir la alfombra verde, se deben llevar a un plano a escala, debido a las limitaciones de espacio, las cuales impiden simular una cancha de más de 15 mts<sup>2</sup> y una estructura metálica de más de 2.5 metros de alto.

##### 4.2.1.1 El Campo de Juego

El campo de juego de esta categoría debe ser rectangular y tener 4,9 metros de largo por 3,4 metros de ancho. La superficie debe ser de color verde y demarcada con líneas blancas de 10 milímetros de ancho [ROB 06], dispuestas según se detalla en la figura 4.1:

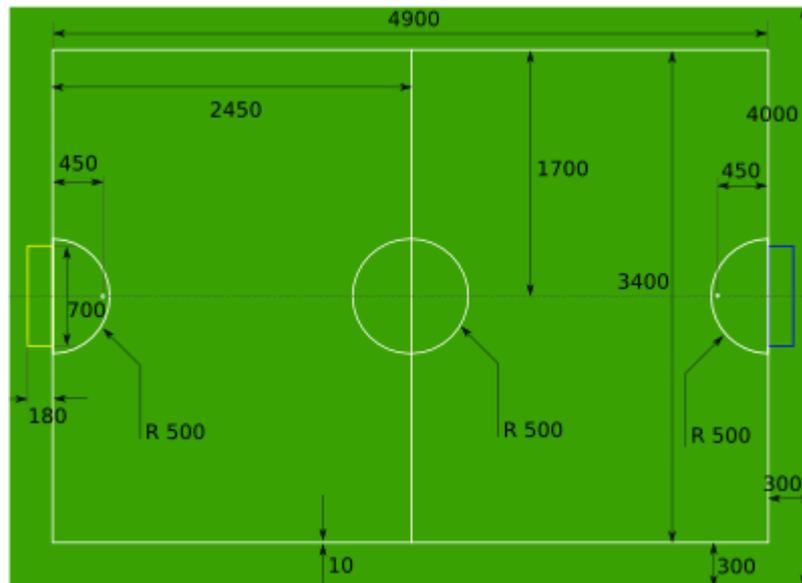


Figura 4.1: Medidas de la cancha expresadas en milímetros.

#### 4.2.1.2 La pelota

La pelota es por defecto color naranja, su peso es de 46 gramos y tiene aproximadamente 43 milímetros de diámetro. La pelota no puede ser cambiada durante un partido sin consentimiento del árbitro, a menos que sufra daños que obligue cambiarla [ROB 06].

#### 4.2.1.3 El robot

La primera regla que se indica, es que el robot no debe poseer en su construcción nada que sea peligroso para él, para otros robots o para los humanos [ROB 06]. Resguardando las 3 leyes de la robótica de Asimov.

El robot debe tener máximo 18 centímetros de diámetro. Si el equipo ocupa un sistema de visión global, cada robot del equipo debe medir 15 centímetros de alto o menos. En caso de tener visión local, todos pueden medir máximo 22,5 centímetros [ROB 06].

#### 4.2.1.4 El dribbling del robot

El robot contempla la construcción de un mecanismo de disparo y retención de la pelota llamado dribbling del robot, el cual debe posicionarse en la parte delantera de éste. Su objetivo es atraer la pelota para luego dispararla y darle una dirección determinada.

El dispositivo de dribbling del robot debe ejercer un movimiento hacia delante para atraerla, cuando la pelota hace contacto con él, y un movimiento hacia atrás para dispararla. El dispositivo debe estar posicionado perpendicular al piso y vertical o parcialmente vertical a la pelota, como se indica en la figura 4.2. No se permiten dispositivos de dribbling a los lados [ROB 06]. Este dispositivo no se implemento en nuestro proyecto.

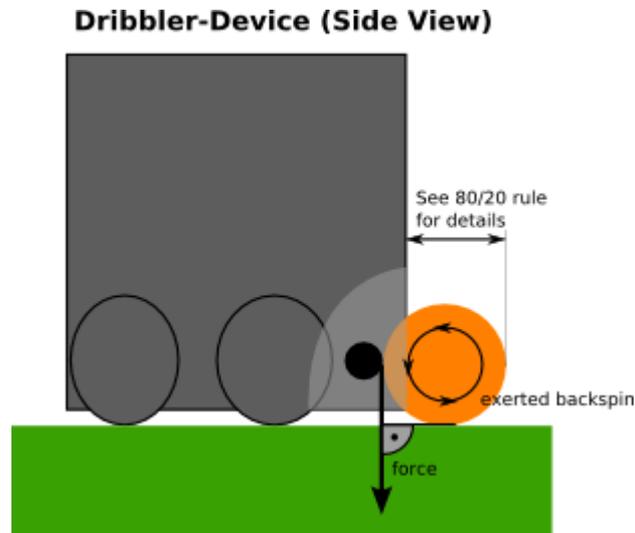


Figura 4.2: Posición del dispositivo de dribbling del robot.

#### 4.2.1.5 Colores y Marcadores

Antes de empezar el juego se realiza la convención de colores que adoptarán los equipos durante el partido (equipo amarillo y el equipo azul). Este marcador debe estar en el centro de la parte superior del robot, junto con otros marcadores según la interpretación dada por el equipo [ROB 06]. Los equipos deben utilizar los colores oficiales (amarillo, azul, verde claro, rozado claro y cian) a menos que ambos equipos convengan lo contrario (ver figura 4.3).

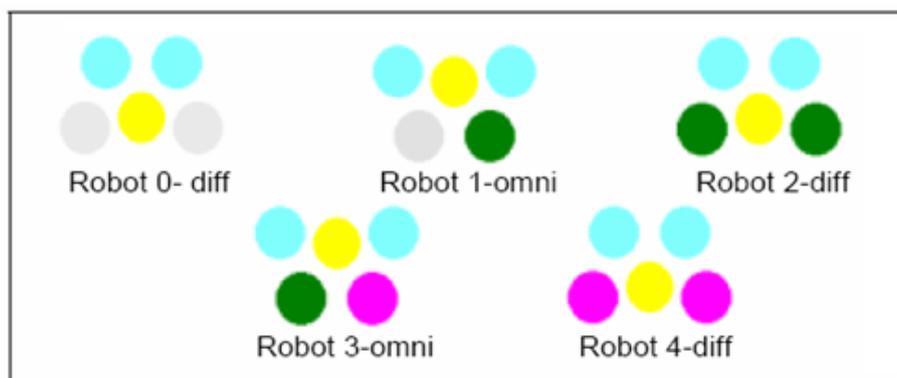


Figura 4.3: Ejemplo de distribución de marcadores de colores para equipo amarillo.

#### 4.2.1.6 Comunicación inalámbrica

Los robots pueden ocupar comunicación inalámbrica mediante computadores o redes presentes alrededor del campo de juego. Según las decisiones del comité de la categoría, no está permitida la comunicación Bluetooth (ver Capítulo 2 – Bases Teóricas, sección 2.1.2.3 Bluetooth).

### 4.2.2 Construcción del prototipo

El proceso de construcción del prototipo de robot, pasa por diferentes etapas incrementales por las cuales se agregan componentes, obteniendo un resultado totalmente funcional. Las etapas contemplan el ensamble de los motores y la base, el acoplamiento del segundo nivel y los circuitos y por ultimo el tercer nivel y la cabeza. Cabe destacar que su construcción comprende la participación de diferentes áreas como la mecánica, eléctrica y el diseño industrial.

#### 4.2.2.1 Ensamblaje de motores y estructura base

Para la construcción de la base del robot, se dispone de 2 micas plásticas de 20 cms. de diámetro similares a los discos compactos, los cuales servirán tanto para la base como para los siguientes niveles. Estas micas son perforadas (ver figura 4.4) y unidas por 2 sostenedores de base, los cuales sostienen la base con el siguiente nivel. De esta manera se hace encajar el sistema de motores y ruedas traseras, dando además el soporte para el siguiente nivel.

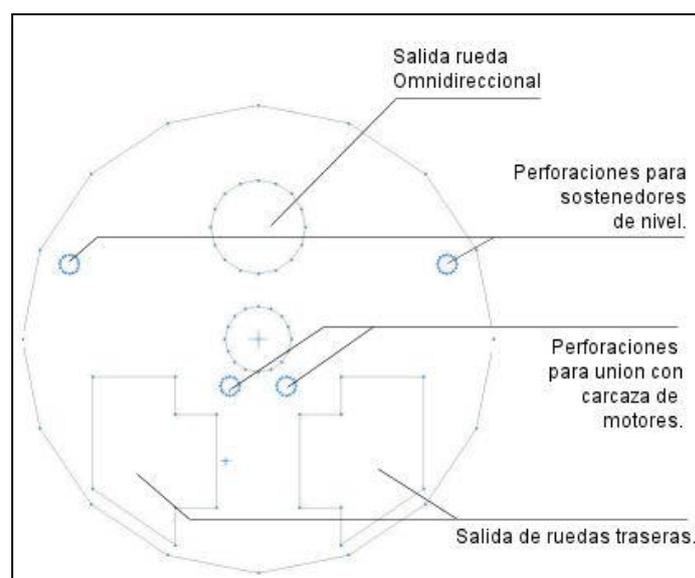


Figura 4.4: Disposición de las perforaciones en la base (modelo 2D).

El sistema mecánico consta de dos motores DC independientes, los cuales se alinean de forma que mediante engranajes internos puedan dar movimiento autónomo a las ruedas. Los engranajes no son visibles y están empaquetados en una estructura plástica que sostiene a los motores.

La decisión de montar en la parte delantera una rueda omnidireccional (ver figura 4.5-1), obedece principalmente a que ésta permite un movimiento libre en todas las direcciones, aminorando considerablemente el problema del rose. Problema que sí presenta una rueda convencional (ver figura 4.5-2), que solo muestra un buen desempeño al realizar movimientos rectilíneos.

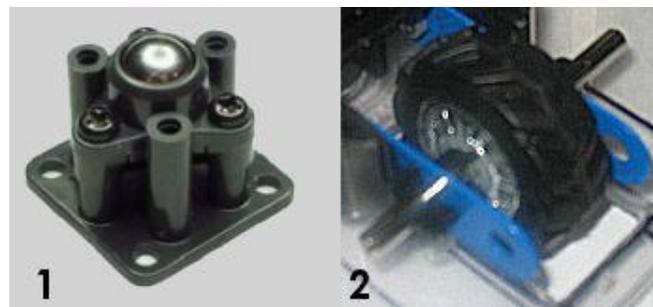


Figura 4.5: Rueda Omnidireccional (1) y rueda convencional (2).

La construcción de la base termina con el ensamblaje de las micas plásticas y la estructura que sostiene a los motores, estos unidos mediante dos pernos de 1.5 pulgadas. Por último, la instalación de la rueda omnidireccional en la parte delantera de la mica plástica, quedando la base del robot terminada y lista para sostener el siguiente nivel, como se muestra en la figura 4.6.



Figura 4.6: Aspecto de la estructura base.

#### 4.2.2.2 Ensamblaje del segundo nivel y circuitos

En el segundo nivel irá la placa Basic Stamp con todos los controladores necesarios para el correcto funcionamiento del robot. Para esto se dispone de una nueva mica plástica, donde se instala la Basic Stamp. Luego la mica es perforada primero para la entrada de los cables de los motores y segundo para la ubicación de la antena, la cual quedará a un costado (ver figura 4.7).

Luego que la mica del segundo nivel esté lista, ésta se instala sobre los separadores metálicos de la base uniéndose junto a otros 2 separadores, los cuales acoplan los niveles y al mismo tiempo ejercen presión para que las micas no se rompan o cedan con el peso.

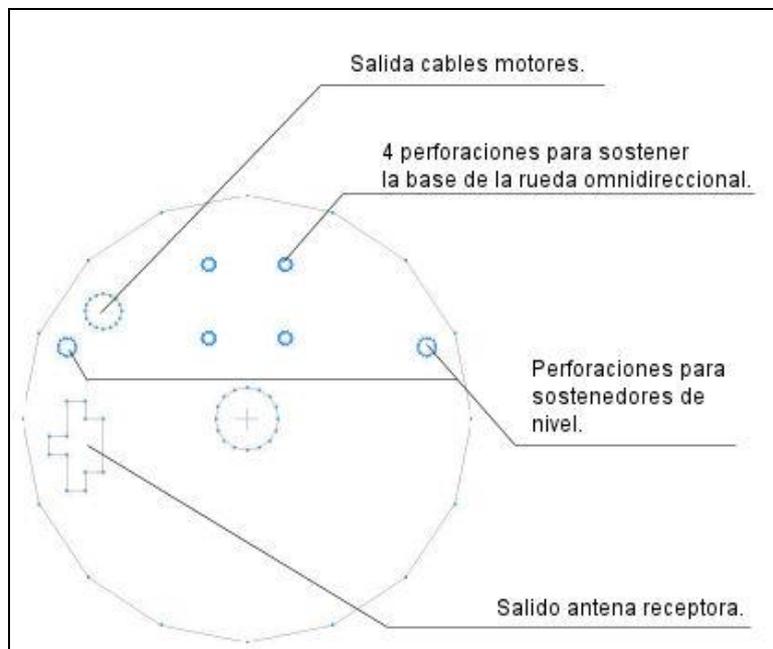


Figura 4.7: Disposición perforaciones en la mica del segundo nivel (modelo 2D).

#### 4.2.2.3 Ensamblaje del tercer nivel y cabeza

El tercer nivel es algo simple y solo contempla la instalación de la última mica sobre los separadores metálicos del segundo nivel y sobre ella las cartulinas con los colores distintivos del jugador. Cabe destacar que esta última mica posee también el cuerpo del robot, encapsulando sus circuitos en un plástico oscuro pero transparente.



Figura 4.8: Aspecto del robot luego de instalado el tercer nivel.

### 4.3 Construcción de la Estructura Metálica

El sistema comprende el posicionamiento de una cámara sobre el campo de juego, la cual pueda observar el movimiento de la pelota y los agentes robóticos que en ella se desenvuelven. La cámara deberá estar sostenida por una estructura metálica, que la posicione a una altura de 2 metros aproximadamente, para así obtener una imagen total del campo de juego.

#### 4.3.1 Diseño Inicial

Como propuesta inicial nació la idea de construir una estructura que permitiera la regulación de su altura, pues esto determina la calibración del enfoque de la cámara sin utilizar el zoom integrado, ya sea éste óptico o digital (en ambos casos puede haber leves pérdidas de precisión). El soporte que sostendría la cámara, se ubicaría en la parte superior, horizontal al campo de juego, sosteniendo a la cámara en la parte central del mismo. Por último, se adhieren 2 tubos fluorescentes en las barras correspondientes al ancho de la estructura.

Toda la estructura estaría compuesta por 14 barras de fierro angular de 50 x 50 x 2. Cuatro barras para la base (ancho = 100 cms, largo = 135 cms), cuatro para los soportes (alto = 190 cms) y cuatro para la parte superior (ídem base). Además se adhieren 2 barras para sostener la cámara en el centro.

#### 4.3.1.1 Aspecto del diseño inicial

Un software de modelamiento de objetos 3d (Anivis AC3D), permitió la obtención de mallas tridimensionales y renderizados con la maqueta de la estructura inicial, la cual se muestra a continuación en siguiente figura 4.9:

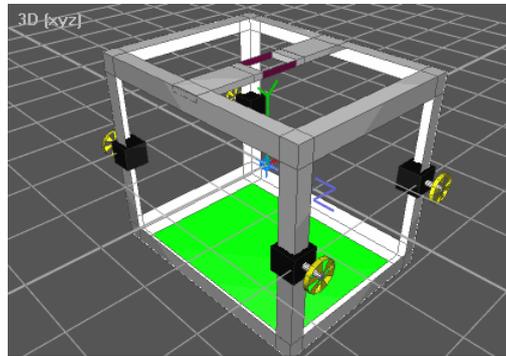


Figura 4.9: Renderizado de la estructura metálica según diseño inicial.

#### 4.3.1.2 Problemas con el diseño inicial de la estructura metálica

La estructura debe poseer ciertas características que faciliten su uso y optimicen la captura de imágenes desde la cámara. Por lo anterior, se busca que la estructura sea construida con materiales ligeros además de tener regulación y poseer un mecanismo de ensamblaje para un cómodo traslado.

En síntesis, los problemas detectados con esta estructura son:

1. El uso de 14 barras provoca un excesivo aumento en el peso.
2. No se logra transportabilidad ni facilidad de ensamblaje.
3. El método de regulación puede no ser preciso.
4. No logra ser una alternativa económica.

#### 4.3.2 Diseño final estructura metálica

El diseño final de la estructura comparte con la anterior el uso de una barra sostenedora para la cámara y los 2 tubos fluorescentes para la iluminación. Por otro lado contempla las siguientes características:

5. Uso de 4 barras de fierro angular (50 x 50 x 2) y 3 barras tubo (20 x 20 x 1,5) para la construcción de la estructura completa.

6. Las 4 barras fierro angular usadas para la base (ancho = 100cms, largo = 135 cms).
7. Las 2 barras tubo usadas como soporte lateral (alto = 190 cms) y 1 usada como sostenedor para la cámara (largo = 135 cms).
8. Base y soportes totalmente desmontables.
9. Peso adecuado y fácil transporte.
10. Construcción económicamente viable.
11. No soporta regulación de los soportes laterales (adecuada al uso del proyecto).

#### 4.3.2.1 Aspecto final de la estructura metálica

En las figuras 4.10, 4.11 y 4.12 se presentan las medidas y el aspecto final que tendrá la estructura metálica.

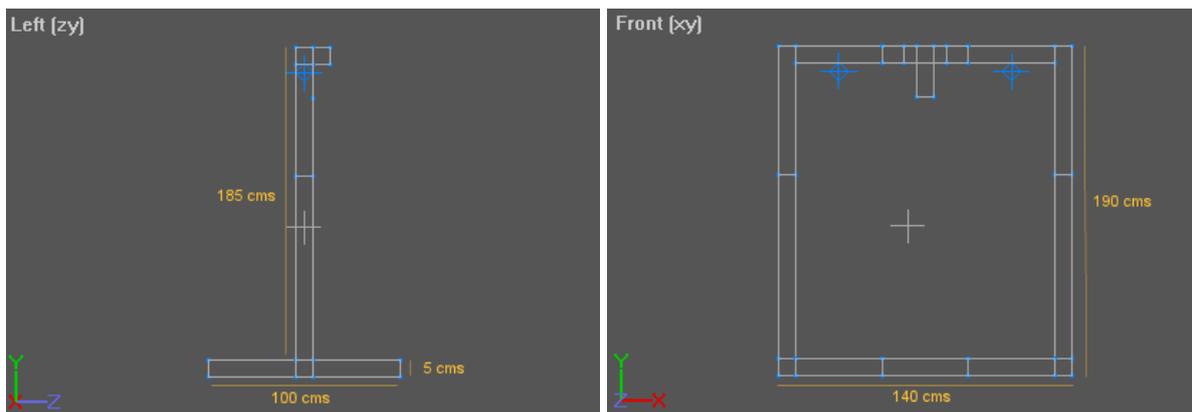


Figura 4.10: Vista lateral y frontal de la estructura (medidas en centímetros).

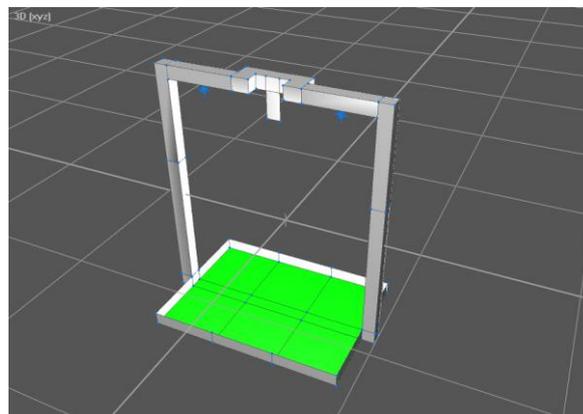


Figura 4.11: Renderizado estructura metálica según diseño final.



Figura 4.12: Aspecto de la estructura según el modelo final.

## 4.4 Control de Movimiento

Para que un robot interactúe con el ambiente, debe poder convertir señales eléctricas en movimiento. Además, para un control más preciso del movimiento se requiere constantemente cambiar las señales enviadas a los actuadores, dejando el trazado del circuito con poco tiempo para atender otras tareas. Aquí tienen importancia el módulo de inteligencia que determina las acciones a seguir por el robot, el módulo de comunicación inalámbrica que se encarga de transmitirle esas acciones al robot y el módulo interno del robot que es el que recibe las instrucciones y activa los motores para ejecutar las acciones.

En este punto se comprenderá cómo se envía los datos al módulo interno del robot y cómo funciona dicho módulo. También se explica el proceso que se realizó para comunicar el computador con el agente de forma inalámbrica (radiofrecuencia).

#### 4.4.1 Tecnología utilizada

Para la comunicación entre el computador y cada Agente robótico móvil se utilizan antenas transmisoras y receptoras de la empresa Parallax, concretamente se utilizan los modelos de antenas de radio frecuencia Parallax 433.92 MHz RF Transmitter (#27980) y Parallax 433.92 MHz RF Receiver (#27981).

A su vez, para el control de los motores se utiliza un Micro Dual Serial Motor Controller (SMC02B), el cual posee un pequeño tamaño y permite controlar 2 motores independientes.

Para transmitir información desde el computador a la Basic Stamp 2, que tiene una antena transmisora, se utiliza un cable serial.

Para utilizar las antenas y el controlador de motores se utilizaron microcontroladores Basic Stamp 2, las cuales poseen un tablero protoboard para crear los circuitos necesarios. Tanto la antena como el controlador de motores tienen pines que se pueden conectar directamente a la BS2.

#### 4.4.2 Características del transmisor y receptor de radiofrecuencia

- Tasas de transferencia de datos de alta velocidad (1200 ~ 19.2k baudios dependiendo del regulador usado).
- La ubicación de los pines permite mayor facilidad de trabajo con los tableros (por ej. La BS2).
- Es de fácil uso ya que utiliza instrucciones SEROUT/SERIN PBASIC.
- Permite alargar la vida de la batería con el modo bajo consumo.
- Alcance de 150 metros (o mayor dependiendo de las condiciones).



Figura 4.13: Transmisor (#27980), 4 pines.



Figura 4.14: Receptor (#27981), 5 pines.

Estos transmisores y receptores son conectados a microcontroladores para la utilización de los datos en el movimiento de los motores y en el procesamiento del computador. El microcontrolador usado es una Basic Stamp 2 HomeWork.

Tanto el transmisor como el receptor utilizado están hechos para que el usuario pueda enviar fácilmente datos seriales, control del robot u otra información inalámbricamente.

La tabla 4.1 muestra los pines utilizados del transmisor, su nombre y la función que desempeña. En la tabla 4.2 se muestran los pines utilizados en el receptor.

Pin	Nombre	Función
2	DATA	Salida de datos
3	5v	Energía – Conectar a +5v DC
4	GND	Tierra – 0 v

Tabla 4.1: Pines utilizados del transmisor (#27980).

Pin	Nombre	Función
3	DATA	Entrada de datos
4	5v	Energía – Conectar a +5v DC
5	GND	Tierra – 0 v

Tabla 4.2: Pines utilizados del receptor (#27981).

Los otros pines no se utilizan ya que el PND sirve para dejar el transmisor/receptor en un estado de poca intensidad, en este estado el módulo no puede transmitir/recibir una señal. A su vez el RSSI es un indicador de la intensidad de la señal, que sirve para proveer de un voltaje analógico proporcional a la fuerza de la señal recibida.

Al iniciar la comunicación entre los módulos se debe enviar un pulso de sincronización para restablecer la conexión entre estos. Este método, mantiene un alto estado de transmisión durante la comunicación, por lo que es mucho más eficiente que enviar varios caracteres correlativos.

Para Basic Stamp 2 la línea de código de calibración es la siguiente:

- PULSOUT 8, 1200

#### 4.4.3 Circuito de cada transmisor y receptor

Para el circuito necesario para la transmisión de datos entre el computador y el Agente se utilizan 3 pines de cada antena: el GND que corresponde a la tierra (vss en la Basic Stamp), el DATA que es el pin por el cual se envían o reciben los datos y el pin 5v que es el pin que se conecta a 5 volts de energía (vdd en la Basic Stamp), tal como se muestra en la figura 4.15 donde se indica el pin y su conexión en la Basic Stamp.



Figura 4.15: El circuito del transmisor a la izquierda y a la derecha el circuito del receptor.

El pin DATA puede ser conectado a cualquier pin de entrada/salida de la Basic Stamp, pero hay que tomar en cuenta que al codificar el programa para el microcontrolador se debe especificar el pin en el que se conectó el mecanismo transmisor/receptor.

#### 4.4.4 Características del controlador de motores

- Tamaño reducido, 2.286 cms x 1.143 cms
- Puede controlar dos motores de forma independiente
- 127 velocidades adelante y atrás, además de freno.
- Tiene una frecuencia de 600 Hz para el control 2 motores
- Velocidad serial entre 1200-19200 (detectada automáticamente)

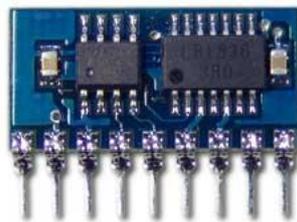


Figura 4.16: Micro Dual Serial Motor Controller (SMC02B).

La función de cada uno de los nueve pines del controlador de motores se muestra en la tabla 4.3. Los números están organizados de izquierda a derecha según la figura 4.16.

Pin	Función
1	Fuente de alimentación del motor (1.8 – 9.0v)
2	Tierra (0v)
3	Fuente de alimentación lógica (2.5 – 5.5v)
4	Entrada serial de control
5	Reset
6	Motor 1, salida positiva
7	Motor 1, salida negativa
8	Motor 0, salida negativa
9	Motor 0, salida positiva

Tabla 4.3: Pines del SMC02B y su función

#### 4.4.5 Circuito del Micro Dual Serial Motor Controller

El controlador está hecho para conectarse directamente a un tablero protoboard. Los pines 1-3 son para la energía. El pin 2 (GND) se conecta a la tierra de la BS2 y al negativo de la batería del motor, el pin 1 se conecta al positivo de la batería del motor. El pin 3 se conecta a la fuente de alimentación lógica, el vdd de la BS2. El pin 5 es para resetear la operación del controlador de motores al estado inicial (todos los motores apagados y esperando una instrucción serial), se conecta en un pin de salida de la BS2. El pin 4 sirve para enviar datos seriales de 8 bits a una tasa entre 1200 y 19200 baudios, se conecta a un pin de salida de la BS2. Los pines 6-9 son para conectar los motores, tal como se muestra en la figura 4.5.

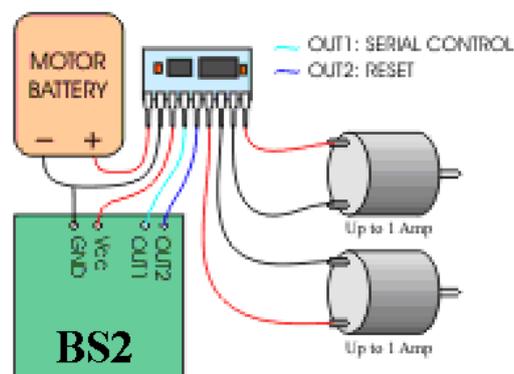


Figura 4.16: Esquema de conexión de los motores, la BS2, la batería y el controlador.

El controlador usa un nivel lógico de transmisión serial, no invertida, con una tasa entre 1200 y 19200 baudios, transmite 8 bits a la vez sin bit de paridad y con un bit de parada.

#### 4.4.6 Protocolo de comunicación

Un protocolo es un conjunto de reglas que deben ser respetadas para que pueda ser realizado un proceso de comunicación. Es necesario establecer estas reglas para que la comunicación entre la IA y los robots se lleve a cabo de forma confiable.

Se debe considerar que la IA decide la posición y orientación futura de cada uno de los robots. Como el movimiento de cada robot es distinto, el módulo de comunicación debe encargarse de que cada robot reciba sólo la información que le corresponde.

También es importante conocer la cantidad de información que cada robot recibe y para ello basta considerar que para que el robot llegue a su posición futura debe mover sus motores a una determinada velocidad y activar su dirección.

La comunicación se establece a través de un pulso de sincronización por lo que se utiliza sólo el envío de la señal, sin confirmación. Además, el robot debe reaccionar rápido a los cambios del medio, esto quiere decir que una confirmación haría que un movimiento quede obsoleto.

Como el ambiente es cambiante, el sistema de visión constantemente captura el mundo del robot y envía información al módulo de inteligencia artificial, si no llega una instrucción se enviará otra instrucción con una nueva acción si el mundo ha cambiado. Con esto se justifica el uso de un protocolo de envío de datos solamente, por el ambiente cambiante en que cada acción será distinta y por la rapidez que se requiere para reaccionar.

#### 4.4.7 Envío de instrucciones

El envío de datos desde el computador a la Basic Stamp 2 se logra a través de la comunicación serial. A su vez, la Basic Stamp tiene cargado un módulo que lee desde el puerto serial y lo envía a través de antenas de radio frecuencia. Un esquema físico del envío de instrucciones se expresa en la figura 4.17.

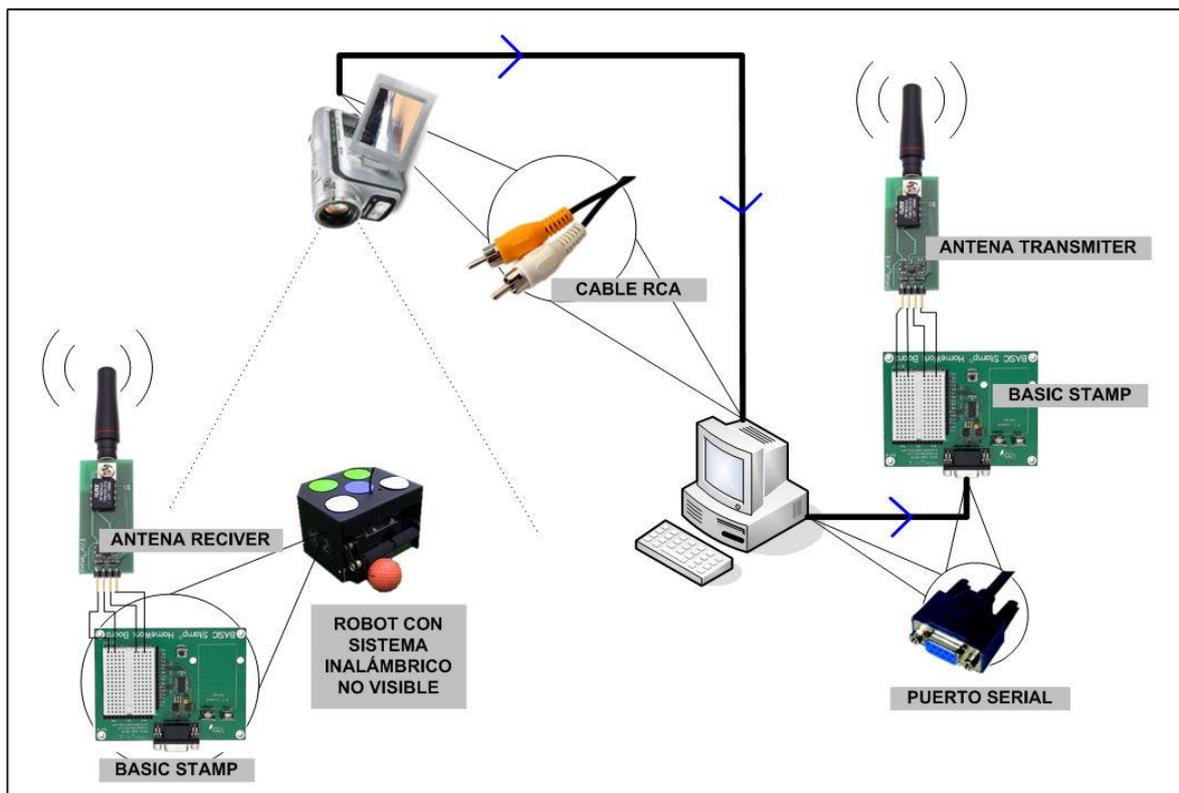


Figura 4.17: Esquema del flujo de datos.

#### 4.4.8 Módulo interno del robot

El módulo que posee el robot es muy sencillo ya que sólo recibe la velocidad de cada rueda, la dirección de estas y un tiempo de ejecución de la instrucción. En la tabla 5.1 se muestra un ejemplo de códigos para cada motor. Así por ejemplo, si se quiere girar se puede activar el motor izquierdo adelante (0) y el motor derecho atrás (3), la velocidad se puede fijar en el rango 0-127.

<b>Motor</b>	<b>Adelante</b>	<b>Atrás</b>	<b>Velocidad</b>
Izquierdo	0	1	0 a 127
Derecho	3	2	0 a 127

Tabla 4.4: Códigos para controlar los motores.

#### 4.4.9 Algoritmos para el control de motores

##### 4.4.9.1 Algoritmo del transmisor

El programa del transmisor recibe el robot a mover, la velocidad, la dirección y el tiempo de ejecución de la instrucción. El algoritmo se detalla a continuación:

```

Begin
Do
  Leer robot, velocidades, dirección, tiempo
  If robot = 0 then
    Mover_robot0 (velocidades, dirección, tiempo)
  Else
    Mover_robot1 (direccion, velocidad)
  End if
Loop
End
    
```

Tabla 4.5: Algoritmo del Transmisor.

##### 4.4.9.1.1 Código BS2 para el transmisor

Se mostrará el código más importante del transmisor, se trata de la lectura desde el puerto serial y luego el envío de datos a un robot a través del dispositivo transmisor.

La siguiente instrucción muestra la lectura desde el puerto serial, se envían como datos el robot a mover, la dirección (derecha o izquierda), la velocidad (izquierda o derecha) y el tiempo.

```
SERIN 16,84,[DEC robot,DEC dd,DEC di,DEC vd,DEC vi,DEC tiempo]
```

La instrucción que a continuación se presenta muestra la forma de enviar datos a través del dispositivo transmisor. El 15 corresponde al pin de salida (data), el símbolo “!” corresponde al código del robot, luego están la distancia, la velocidad y el tiempo.

```
SEROUT 15, 16468, [ "!", dd, di, vd, vi, tiempo.HIGHBYTE, tiempo.LOWBYTE]
```

#### 4.4.9.2 Algoritmo que controla a los robots

El programa que se carga al robot recibe, mediante la antena, la velocidad y la dirección de cada motor, luego activa los motores con los parámetros recibidos. El algoritmo se detalla a continuación:

```
Begin
Do
  Read velocidades, direccion
  Mover_motores(direccion, velocidad)
Loop
End
```

Tabla 4.6: Algoritmo control de motores.

##### 4.4.9.2.1 Código BS2 para el robot

Se mostrará el código más importante de un robot (receptor), se trata de la lectura desde la antena receptora y el movimiento de los motores.

La siguiente instrucción muestra la lectura desde la antena. El número 3 corresponde al pin de entrada (data), con WAIT("!") espera por el robot con código “!”, cuando llegue recibe la dirección (derecha, izquierda), la velocidad (izquierda, derecha) y el tiempo.

```
SERIN 3, 16468, [WAIT("!"), dd, di, vd, vi, tiempo.HIGHBYTE, tiempo.LOWBYTE]
```

Las próximas instrucciones contienen la forma de mover los motores con los datos recibidos. El 14 corresponde al pin de salida (data), el tercer parámetro corresponde a la dirección (0 y 3 hacia delante, 1 y 2 hacia atrás).

```
SEROUT 14,84,[$80,0,dd,vd]
SEROUT 14,84,[$80,0,di,vi]
PAUSE tiempo
SEROUT 14,84,[$80,0,dd,0]
SEROUT 14,84,[$80,0,di,0]
```

## 4.5 Implementación de la visión artificial

### 4.5.1 Librería de visión CMVision

Los tres acercamientos más comunes a la segmentación utilizan los colores, los bordes o límites, la textura o posiblemente una combinación de ellos (ver Capítulo II - Bases Teóricas – sección 2.2 Visión artificial). Dadas las limitaciones y las capacidades de procesamiento de imagen disponibles esta librería utiliza la segmentación del color. Este método demostró que satisface mejor los requerimientos de tiempo real sin necesidad de utilizar la aceleración de hardware. Ha demostrado que trabaja de forma robusta en usos autónomos de robótica y es mucho más simple de utilizar que los otros acercamientos que utilizan el borde o la textura que además procesa el color.

El espacio de color más común es el RGB que representa los colores rojo, verde y azul. La librería se intentó implementar procesando los espacios de color RGB, una proyección del RGB llamada YRGB fraccionario y YUV que es usado por los estándares de video NTSC y PAL. En la figura 4.18 se muestra el resultado del uso de la librería.

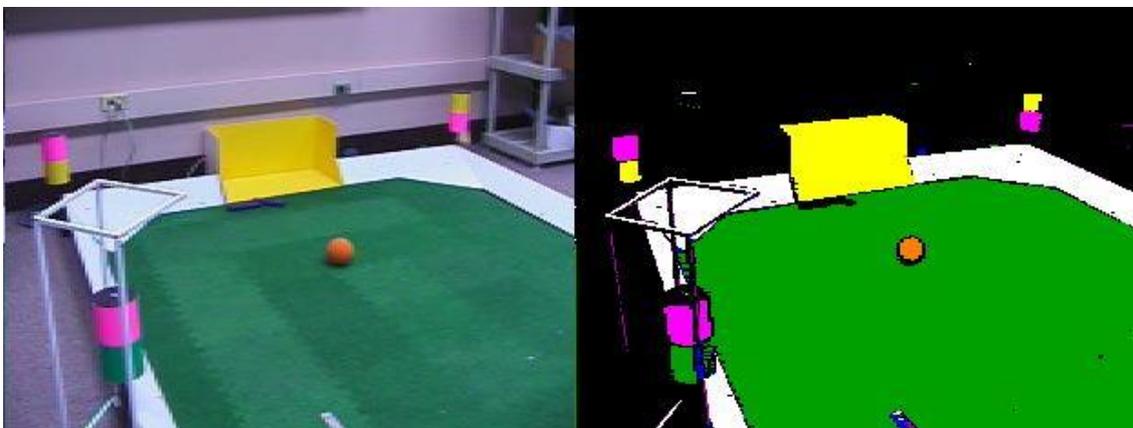


Figura 4.18: A la izquierda la imagen de video original, y a la derecha la imagen con colores clasificados por los umbrales YUV predefinidos.

CMVision trabaja con formato YUV ya que es muy robusto y rápido al trabajar directamente con el hardware.

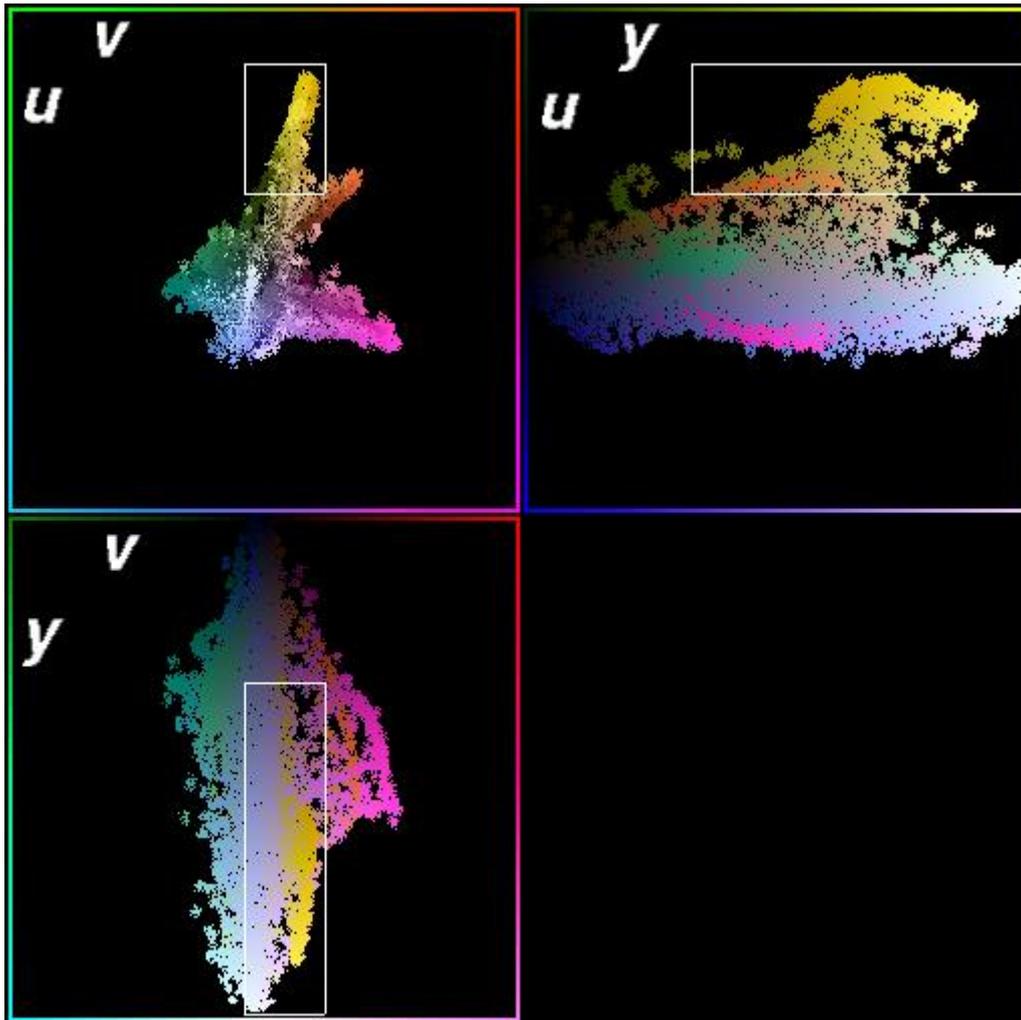


Figura 4.19: Un histograma YUV de un campo de fútbol de RoboCup, con un umbral definido para un color amarillo.

Puede procesar hasta 32 colores con una resolución de 640x480 y a un ritmo de 30 o 60 Hz sin hardware especializado.

Se utiliza esta librería por lo eficiente en el procesamiento de la imagen y por el bajo costo que implica al ser libre y al no requerir un hardware especializado. Se ha utilizado en los cuadrúpedos de la Sony (Aibo), los robots Minnow, la categoría F180 de la RoboCup y seguimiento de animales.

Para poder ocupar la librería se necesita una capturadora de video con el driver de video *video4linux 2* bajo un kernel de las series 2.2.x (comenzando con el 2.2.6) o superior.

## 4.5.2 Reconocimiento de patrones

Como inicio e introducción al desarrollo del sistema de visión, se presenta la solución propuesta a la etapa de reconocimiento de patrones en un marco o espectro observable. La visión artificial es la encargada de proveer de información visual del entorno donde éste se encuentre. En este caso los robots necesitan saber cómo se comporta el ambiente y quiénes interactúan en el, para poder tomar decisiones acertadas.

### Elementos

Ahora veremos los componentes mínimos necesarios para el funcionamiento de la visión artificial:

- **Superficie Plana Verde:** Simula el campo de fútbol y es el entorno por donde los robots se desenvolverán. Se encuentra delimitada por franjas blancas y los arcos con los colores diferentes para simular equipos contrarios.
- **Cámara de Video:** De alta resolución, posicionada sobre la superficie verde a una altura aun no determinada, para captar la visión panorámica del campo de juego. Es el sensor que captura el movimiento en el ambiente y determina la decisión que pueda tomar el robot que se encuentra ahí.
- **Robot:** El(los) robot(s) estará(n) en la superficie plana de color verde. Poseen en la parte superior una base negra, plana y con círculos de diferentes colores que representan un equipo y una posición para cada jugador (defensa, delantero, etc.)
- **Pelota:** Se trata de una pelota similar a una pelota de golf, de color rojo. Esta deberá ser interceptada por el robot y deberá reconocerla por medio de la cámara, para determinar trayectoria, velocidad, etc.



Figura 4.20: Elementos del Problema.

#### 4.5.2.1 Descripción del problema

El problema es determinar donde está cada jugador y que tipo de jugador es: delantero, defensa, arquero, medio campista. Todo esto usando los colores que cada robot lleva en su parte superior. Una primera aproximación es tener 2 simulaciones de la parte superior plana con colores de los robots e identificar el tipo de jugador que es. Más específico es determinar en que posición dentro del entorno se encuentra con respecto al otro. Se debe determinar a que robot o jugador pertenece cada color capturado por la cámara.

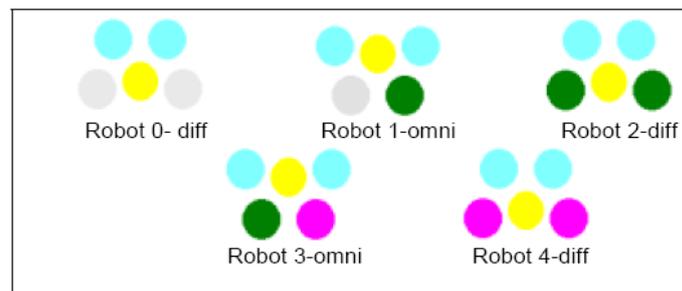


Figura 4.21: Ejemplo de colores superiores para cada Robot.

#### 4.5.2.2 Convenciones

La cámara no observa al robot íntegramente, sino que ve sólo su parte superior donde está el patrón de colores ya descrito. Entonces el patrón se guía de la siguiente forma:

- En el centro está el color común que identifica al equipo
- Los colores adyacentes al central determinan el tipo de jugador
- Existe un color que determina la orientación del robot
- Estos colores están sobre una superficie negra (en el mismo robot) para facilitar la captura de colores (el negro absorbe luz disminuyendo el ruido)
- Se calcula un radio de aceptación para determinar si un color pertenece a un jugador o a otro. Esto es por si dos jugadores se aproximan mucho pudiendo haber confusión o solapamiento
- A cada color se le da un valor numérico que lo identifique como único (código).
- Se crea una matriz con cantidad de filas igual al número de jugadores detectados. Y con 8 columnas:

- c0: posición X del patrón (circulo central).
- c1: posición Y del patrón (circulo central).
- c2: código del 1<sup>er</sup> color encontrado en el radio del patrón.
- c3: código del 2<sup>do</sup> color encontrado en el radio del patrón.
- c4: código del 3<sup>er</sup> color encontrado en el radio del patrón
- c5: código del 4<sup>to</sup> color encontrado en el radio del patrón
- c5: tipo de jugador al que corresponde la combinación de colores encontrado.
- c6: contador de colores encontrados (para ver cuantos ha detectado la cámara).

		Posicion X	Posicion Y	1er Color reconocido	2do Color reconocido	3er Color reconocido	Tipo	Contador
Jugadores {	0							
	1							
		0	1	2	3	4	5	6

Figura 4.22: Ejemplo de matriz que maneja 2 jugadores y 3 colores para estos.

#### 4.5.2.3 Algoritmo aplicado

A continuación se presenta el algoritmo aplicado para determinar a que jugador pertenecen los colores capturados y el tipo de jugador que se reconoció:

1. Agregar las posiciones del plano correspondientes al centro de cada robot a la matriz.
2. Agregar ceros a la última columna, esto es porque esa columna se usa como contador.
3. Calcular la distancia entre el color central y los demás colores. Si el color analizado está en el rango de aceptación, entonces se agrega como color en la matriz de jugadores, sino se sigue buscando con otro jugador. Cabe destacar que el radio de aceptación se calcula automáticamente al iniciar la aplicación.
4. Distancia:  $(x_2 - x_1)^2 + (y_2 - y_1)^2$  aquí se evita la raíz para optimizar el procedimiento del algoritmo en el uso de la CPU.

5. Sumar los colores. Esto porque cada color tiene su representación numérica (código), que están alrededor del círculo central de cada jugador. El número es el equivalente a la posición del jugador.
6. Con lo anterior ya hemos determinado que tipo de jugador es y la posición que tiene (en píxeles) dentro del campo de juego.
7. Para determinar la posición de un jugador respecto de otro basta con comparar las coordenadas x para ver si un robot está a la izquierda o a la derecha del otro y las coordenadas y para ver si esta arriba o abajo del otro.

### **4.5.3 Posicionamiento del robot**

En esta sección se estudiará el problema de posicionamiento de los robots, que forma parte de la visión artificial y que ayuda a que el robot tenga una orientación, esto es, que se sepa hacia donde está apuntando y si tiene que girar para ir a otro lugar específico.

#### **4.5.3.1 Descripción del problema**

La precondition para obtener el posicionamiento de un robot, es que deben estar calculadas las posiciones de cada color en el campo de juego, correspondiente al robot estudiado. Con esta información se debe determinar mediante un cálculo de ángulos, la dirección a la que apunta el robot.

#### **4.5.3.2 Consideraciones importantes**

Se debe tener en cuenta varios aspectos antes de buscar la solución al problema de posicionamiento. Estos se detallan a continuación:

- La fotografía tomada por la cámara se muestra en píxeles que son los puntos en la imagen. Estos están numerados desde (0,0), que corresponde a la esquina superior izquierda, hasta el píxel (N, M), que corresponde al tamaño de la imagen (N=ancho y M=alto).
- De lo anterior se decide trabajar con la función coseno para determinar el ángulo del robot (ver figura 4.23) y el ángulo del robot respecto a la pelota (ver figura 4.24). Estos ángulos están en el rango  $[-180, 180]$  para determinar el movimiento izquierda o derecha del robot.

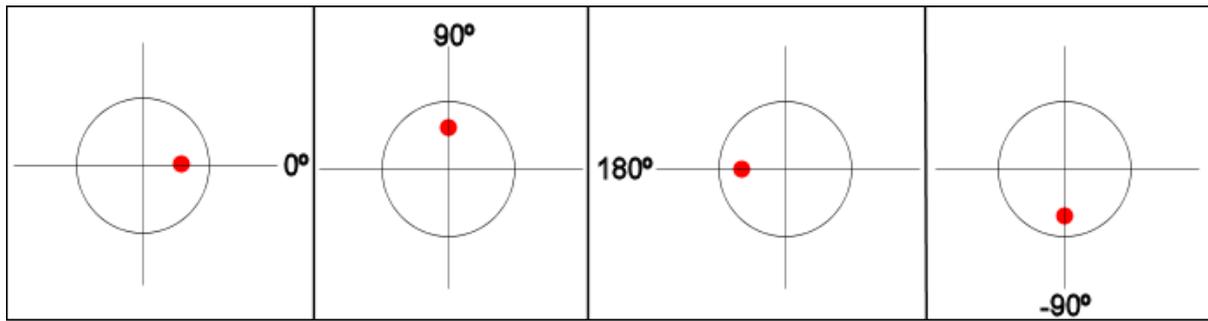


Figura 4.23: Diferentes ángulos que puede tomar el robot con respecto de si mismo.

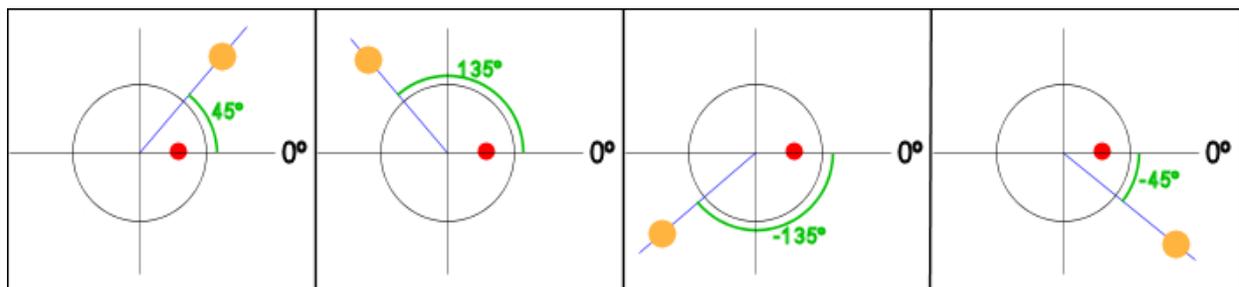


Figura 4.24: Diferentes ángulos del robot (en  $0^\circ$ ) respecto de la pelota.

- Se tendrán dos colores representativos de la posición del robot: el del centro que identifica al equipo y sirve de punto (0,0), y otro color ubicado en la parte delantera del robot indicando la dirección a la que apunta. En la figura 4.25 se muestra un ejemplo: el color azul identifica el equipo al que pertenece y el color amarillo sería el que indica la dirección a la que apunta el frene del robot.

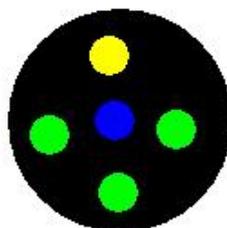


Figura 4.25: Visual superior del robot.

### 4.5.3.3 Algoritmo aplicado

A continuación se modela una solución al problema de posicionamiento, en el cual se busca el ángulo de inclinación del robot:

1. Dados 2 puntos  $(x1, y1)$  y  $(x2, y2)$  como se muestra en la figura 4.26. Se comienza calculando la distancia de los catetos, el cateto “a” se calcula con la diferencia entre  $x2$  y  $x1$ , a su vez el cateto “b” se calcula con la diferencia entre  $y1$  e  $y2$ .
2. Calcular la distancia de la hipotenusa “c”. Esto se puede calcular con el teorema de Pitágoras:

$$\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

3. Calcular el ángulo despejando la ecuación:

$$\cos(\alpha) = \frac{a}{c}$$

Despejando nos queda:

$$\alpha = \arccos\left(\frac{a}{c}\right)$$

4. Opcional: Si el resultado entregado es en radianes se puede transformar a grados. La nueva fórmula quedaría:

$$\alpha = \frac{\arccos\left(\frac{a}{c}\right) * 180}{\pi}$$

5. Para dejar el ángulo entre  $[-180,180]$  lo ajustamos determinando a que cuadrante pertenece y que ángulo es en realidad. Primero le quitamos 180 y luego preguntamos por las diferencias obtenidas anteriormente: si la diferencia de  $y2$  e  $y1$  es negativa, se le cambia signo al ángulo obtenido.

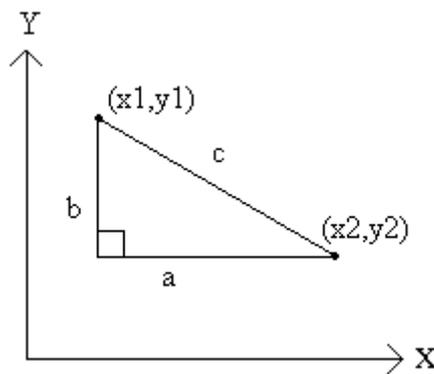


Figura 4.26: Ejemplo de dos puntos en el plano.

## 4.6 Implementación de estrategias

El desarrollo del framework de visión, se ve potenciado por la construcción de distintas estrategias programadas, las cuales proporcionan una amplia gama de ideas sobre las múltiples aplicaciones de la visión artificial.

### 4.6.1 Seguir la pelota

Esta estrategia consiste en la persecución de la pelota, por parte del agente más cercano a ella. Para determinar el movimiento del robot debemos considerar 2 casos: movimientos rectilíneos (trayectoria en línea recta) y movimientos curvilíneos (trayectoria en curvas).

#### 4.6.1.1 Movimiento rectilíneo

Todos los cálculos relacionados con las magnitudes que describen los movimientos rectilíneos podemos hacerlos con estas dos ecuaciones:

$$d = e_o + v_o \cdot t + \frac{1}{2} \cdot a \cdot t^2$$

$$v_f = v_o + a \cdot t$$

**Donde:**

- d es el desplazamiento del robot
- $e_o$  es la posición inicial
- t es el intervalo de tiempo que estamos considerando
- $v_o$  es la velocidad inicial (al principio de nuestro intervalo de tiempo)
- $v_f$  es la velocidad final (al final de nuestro intervalo de tiempo)
- a es la aceleración

Si el movimiento es constante, como es nuestro caso, la aceleración es cero y la ecuación queda así:

$$d = v_o \cdot t$$

$$v_f = v_o$$

Conocemos la distancia entre el robot (origen) y la pelota (destino), también conocemos la velocidad constante. La ecuación final para determinar el tiempo está dada por:

$$t = d / v \text{ (pixel/ms)}$$

#### 4.6.1.2 Movimiento curvilíneo

Dentro de estos movimientos podemos incluir a los movimientos circulares, elípticos, parabólicos, etc. La ecuación simplificada para representar un movimiento circular está dada por:

$$w = \frac{\theta}{t}$$

**Donde:**

- w      velocidad angular promedio
- $\theta$     desplazamiento angular, radianes  $\theta$
- t      tiempo

#### 4.6.1.3 Algoritmo aplicado

Para determinar la velocidad, la dirección y el tiempo, se presenta un algoritmo simple que utiliza la teoría anteriormente mostrada.

- Se tiene como parámetros de entrada la distancia entre los jugadores y la pelota, el ángulo del jugador, el ángulo del jugador respecto a la pelota, la posición de la pelota y el tipo de cada jugador.
- Se calcula la diferencia entre el ángulo del jugador respecto a la pelota, esto se hace para conocer la relación existente entre el ángulo y la magnitud del giro a realizar.
- Verificar a través del ángulo si el robot está más cerca de llegar por la derecha o por la izquierda a la pelota.
- Si está justo al frente el robot debe moverse en línea recta hacia la pelota.
- Si está más cerca por la derecha, debe mover el motor derecho más lento para permitir el giro. Mientras menos diferencia haya entre los ángulos menos debe ser la diferencia de velocidades y viceversa.

- Si está más cerca por la izquierda debe mover el motor izquierdo más lento para permitir el giro. Mientras menos diferencia haya entre los ángulos menos debe ser la diferencia de velocidades y viceversa.
- Calcular el tiempo de acuerdo a las fórmulas anteriormente descritas.

#### 4.6.1.4 Consideraciones de movimiento y estimación

Para esta estrategia, inicialmente se pensó en incorporar un algoritmo o método de inferencia de movimiento, que permitiera conocer la posición de la pelota, a partir de la información de su posición en un tiempo  $t$  y  $t-1$ , determinando así la posición de la pelota en un tiempo  $t+1$ . Esto permitiría la obtención de coordenadas futuras, con lo que podríamos direccionar al agente a posiciones no lógicas de seguimiento, sino de intercepción.

El algoritmo o método que cumple con estas exigencias es el Filtro de Kalman, el cual es un método estadístico utilizado para predicción y estimación de variables en el tiempo.

Finalmente el filtro de kalman no se pudo implementar por los siguientes motivos:

- Al ser un método estadístico, necesariamente se trabaja con una muestra, la cual es proporcionada por un estudio previo del comportamiento de las variables conocidas. Este estudio no es posible realizarlo en un tiempo acotado.
- Generalmente el número de variables resultantes es una, lo cual da un espectro esperable de soluciones. Sin embargo este proyecto, por diversas características<sup>6</sup>, posee más de una variable, complicando enormemente el método.
- El método necesita poseer variables iniciales para comenzar el filtrado estadístico, lo cual este sistema no tiene, pues el movimiento es netamente estocástico. Esto se soluciona con un espacio muestral o historial de posiciones, las cuales podrían estimar el estado inicial para usar el filtro.

Para mas información sobre el filtro de kalman, ver Anexo A.

---

<sup>6</sup> Los agentes tienen limitaciones hardware que agregan complejidad al sistema.

## **4.6.2 Tiro penal**

Esta estrategia consta de la ejecución de un tiro penal por parte de uno de los agentes y el intento de contención de este tiro, por parte de otro agente.

Para la ejecución de esta estrategia se tomó en cuenta la posición inicial de la pelota, el ángulo que genera la pelota en movimiento y la dirección y velocidad que tiene que tomar el arquero para atajar el disparo.

### **4.6.2.1 Algoritmo aplicado**

El algoritmo recibe los datos de posición de la pelota y el movimiento, para luego calcular la dirección, velocidad y tiempo de movimiento del arquero.

- Se tienen la posición del robot arquero y la pelota.
- Se le da la indicación al robot pateador que ejecute el penal.
- Se comienza a detectar el movimiento en la pelota. Si no hay movimiento quiere decir que el robot que patea aun no llega a la pelota, sino entonces ya llego y hay que calcular su dirección.
- Se estima su dirección analizando datos de la pelota en dos instantes de tiempo distintos y en posiciones distintas (se movió).
- Estimada la dirección, se calcula la velocidad necesaria y la dirección que tendrá que tomar el arquero para atajar el tiro.

Cabe destacar que el movimiento que realiza el robot arquero es solo hacia delante o atrás, pues la disposición de las ruedas no permite un movimiento más óptimo y rápido.

### **4.6.3 Perseguir compañero**

Esta estrategia tiene por objetivo que uno de los agentes realice movimientos aleatorios o programados y el otro lo siga. Esta estrategia no hace uso de la pelota.

A continuación se presenta el algoritmo aplicado.

#### **4.6.3.1 Algoritmo aplicado**

Para determinar la velocidad, la dirección y el tiempo, se presenta un algoritmo simple que utiliza la misma teoría de la estrategia de seguir la pelota.

- Se tienen como parámetros los datos de cada robot como distancia de uno respecto del otro y el tipo de jugador.
- Calcular el ángulo entre los jugadores para obtener la magnitud del giro, en caso de que sea necesario.
- Si está en línea recta, sólo tiene que avanzar.
- Si no está en línea recta debe verificar si el ángulo dice que está más cerca por la derecha o por la izquierda.
- Dependiendo el ángulo y la distancia se calcula la velocidad y el tiempo de ejecución de una acción. Para esto se utilizan las fórmulas descritas en la estrategia de seguir la pelota.

En general esta estrategia es similar a la de seguir la pelota, sólo que ahora no sigue a la pelota sino que al otro agente.

## 4.7 Pruebas

Esta sección contempla la realización de las pruebas al sistema y sus respectivos módulos. El detalle de las mismas se muestra a continuación.

### 4.7.1 Pruebas de caja negra módulo de visión

#### 4.7.1.1 Prueba al la conectividad de la cámara con el sistema

**Descripción:** la prueba intenta vislumbrar que tan efectiva y estable es la conexión de la cámara con el sistema, sin que ocurran imprevistos que afecten el funcionamiento del sistema en general.

**Condición de Ejecución:** La cámara debe estar conectada al sistema sin indicadores de salida (*display off*) por al menos 10 minutos.

**Datos de entrada:** imagen captada por la cámara.

**Resultado esperado:** la cámara debe mostrar sin inconvenientes la imagen del campo de juego, sin indicadores de salida por al menos 10 minutos.

**Resultado entregado:** la cámara logra mostrar la imagen sin inconvenientes pero no completamente limpia, pues aparecen indicadores de salida que se interpretan como basura. Además permanece solo 5 minutos encendida, luego se apaga y hay que volver a encenderla y dejarla en modo *display off*.

#### 4.7.1.2 Prueba a la imagen captada, sin sensibilidad de luz

**Descripción:** el objetivo de la prueba es conocer el comportamiento de la captura, frente a la nula exposición de luz externa (luz que no sea la integrada en la estructura).

**Condición de Ejecución:** Cámara conectada al sistema e interfaz ejecutándose para la captura de imágenes.

**Datos de entrada:** tres imágenes capturadas en su estado de segmentación.

**Resultado esperado:** tres imágenes segmentadas sin diferencias sustanciales en su estructura de colores.

**Resultado entregado:** no hubo cambios sustanciales en las 3 imágenes.

#### **4.7.1.3 Prueba a la imagen captada, con sensibilidad de la luz**

**Descripción:** el objetivo de la prueba es conocer el comportamiento de la captura, frente a diferentes cambios en la exposición de luz externa a la de la estructura.

**Condición de Ejecución:** Cámara conectada al sistema e interfaz ejecutándose para la captura de imágenes.

**Datos de entrada:** tres imágenes capturadas en su estado de segmentación.

**Resultado esperado:** tres imágenes segmentadas sin diferencias sustanciales en su estructura de colores.

**Resultado entregado:** las tres imágenes entregadas reflejaron cambios sustanciales que provocaron la caída del sistema. La exposición de luz externa desde diferentes ángulos refleja en los objetos y distorsiona el color segmentado, provocando salida no deseada.

### **4.7.2 Pruebas de caja negra módulo de estrategia**

#### **4.7.2.1 Prueba a la entrega de datos con un robot y la pelota**

**Descripción:** se pretende mostrar el resultado dado por la estrategia, según un conjunto de datos entregados como entrada, correspondientes a las posiciones de la pelota y un robot en el campo de juego.

**Condición de Ejecución:** Cámara conectada al sistema e interfaz ejecutándose en modo automático.

**Datos de entrada:** 10 imágenes de la pelota y un robot en diferentes posiciones (ángulos) en el campo de juego. Los ángulos del robot con respecto a la pelota están comprendidos entre: 0°, 45°, 90° y 180°, en ambos sentidos.

**Resultado esperado:** el resultado debe ser la dirección correcta a emprender por el robot y la velocidad de cada motor aplicado a sus ruedas respectivas.

**Resultado entregado:** la prueba entregó resultados correctos en el 80% de los casos (ver tabla 4.7). El 20% de errores fue entregado cuando el ángulo respectivo del robot estaba cercano a 180° con respecto a la pelota.

Ángulo del Robot Respecto de la pelota.	Resultado Esperado		Resultado entregado (dir, vMD, vMI)	
	Dirección	Velocidad		
		MD		MI
0° - 5°	Adelante	110	110	Adelante, 110, 110
40° - 55°	Derecha	110	70-90	Derecha, 110, 85
85° - 95°	Derecha	110	40-70	Derecha, 110, 43
175° - 180°	Derecha	110	20-40	Derecha, 110, 97
0° - -5°	Adelante	110	110	Adelante, 110, 110
-40° - -55°	Izquierda	70-90	110	Izquierda, 81, 110
-85° - -95°	Izquierda	40-70	110	Izquierda, 66, 110
-175 - -180	Izquierda	20-40	110	Izquierda, 105, 110

Tabla 4.7: Ángulos de entrada y resultados obtenidos, prueba a la estrategia “seguir la pelota”.

### 4.7.3 Pruebas de caja negra módulo de comunicación

#### 4.7.3.1 Prueba a la recepción de los datos con interferencia.

**Descripción:** el objetivo de esta prueba es probar la consistencia del flujo de información de las antenas transmisoras y su entrega a las receptoras, según una distancia dada, con interferencia de por medio (paredes, objetos, etc.).

**Condición de Ejecución:** Interfaz ejecutándose en modo manual, robots y antenas conectadas y alimentadas con energía.

**Datos de entrada:** envió de 10 instrucciones por cada distancia aplicada.

**Resultado esperado:** que el envío de las 10 instrucciones sea recepcionado 10 veces en el robot.

**Resultado entregado:** los resultados arrojaron que la interferencia junto con una mayor distancia aplicada, son causales de retardo en la entrega de datos y la no entrega de la totalidad de ellos.

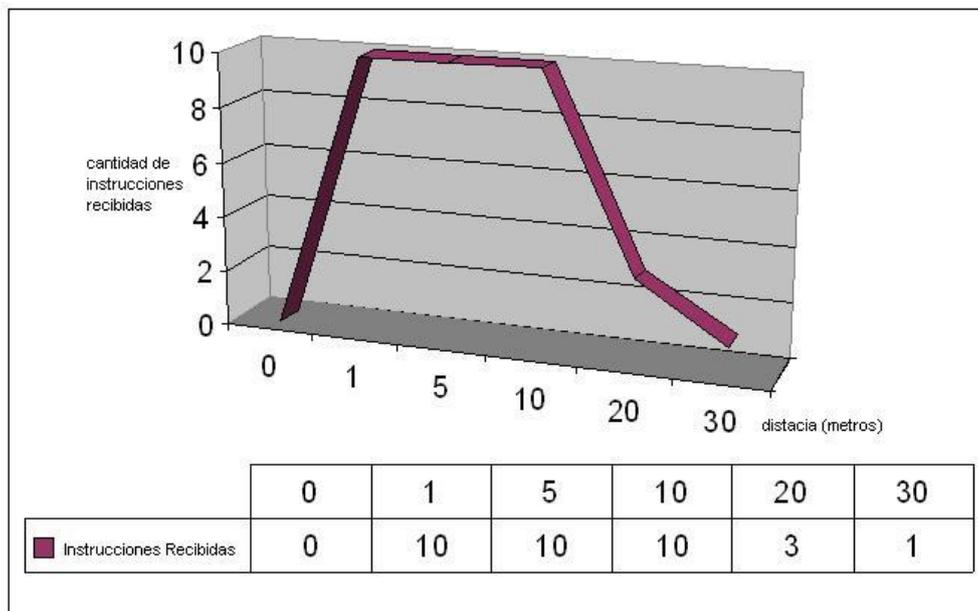


Figura 4.27: Grafico de resultados prueba de recepción de datos con interferencia.

#### 4.7.3.2 Prueba a la recepción de los datos sin interferencia.

**Descripción:** el objetivo de esta prueba es probar la consistencia del flujo de información de las antenas transmisoras y su entrega a las receptoras, según una distancia dada, sin interferencia de por medio.

**Condición de Ejecución:** Interfaz ejecutándose en modo manual, robots y antenas conectadas y alimentadas con energía.

**Datos de entrada:** envío de 10 instrucciones por cada distancia aplicada.

**Resultado esperado:** el envío de las 10 instrucciones sea recepcionado 10 veces en el robot.

**Resultado entregado:** los resultados arrojaron la recepción de todos los envíos independiente de la distancia aplicada. La excepción es en 0 metros donde no recibió ninguno de los 10 envíos.

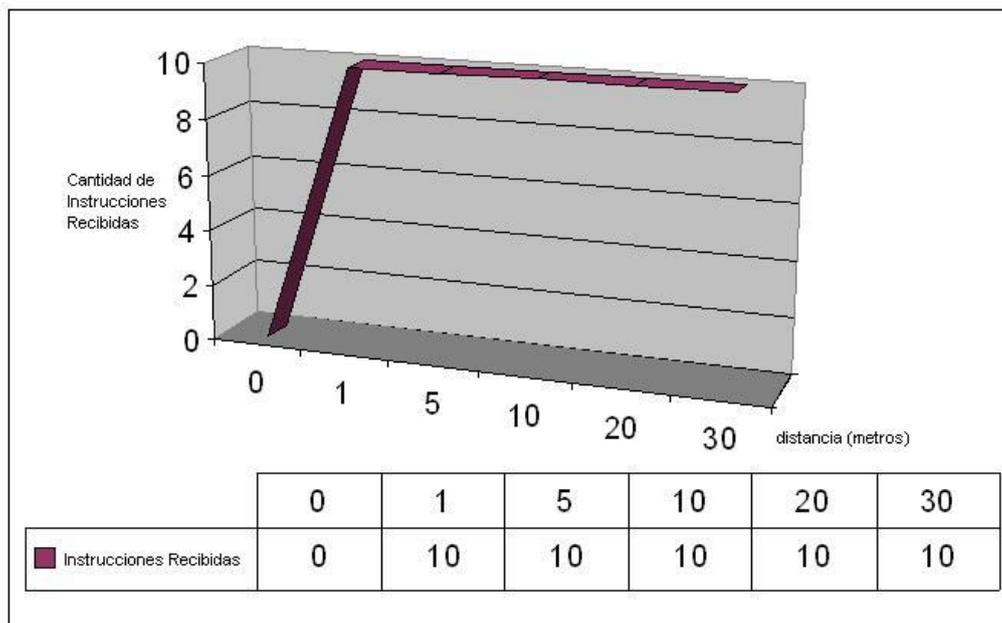


Figura 4.28: Grafico de resultados prueba de recepción de datos sin interferencia.

### 4.7.4 Pruebas de caja blanca

En este tipo de pruebas se consideraron las funciones más conflictivas y críticas del sistema, las cuales nos permiten comprobar la integridad del sistema, dada la relación que tienen con los demás módulos.

#### 4.7.4.1 Clase Analizador – Función calcular

Esta función se encarga de determinar a que robot pertenece el color pasado como parámetro. Su código puede verse en la tabla 4.7 y su análisis de caminos en la figura 4.29:

```
void Analizador::calcular (int **m, int x, int y, int color){
    int i=0;
    int d=(x-m[i][0])*(x-m[i][0])+(y-m[i][1])*(y-m[i][1]);
    while (d > radio) {
        i++;
        d=(x-m[i][0])*(x-m[i][0])+(y-m[i][1])*(y-m[i][1]);
    }
    if (m[i][7]<4){
        m[i][m[i][7]+2]=color;
        m[i][7]++;
    }
}
```

Tabla 4.8: Código función “calcular”

##### 4.7.4.1.1 Camino básico

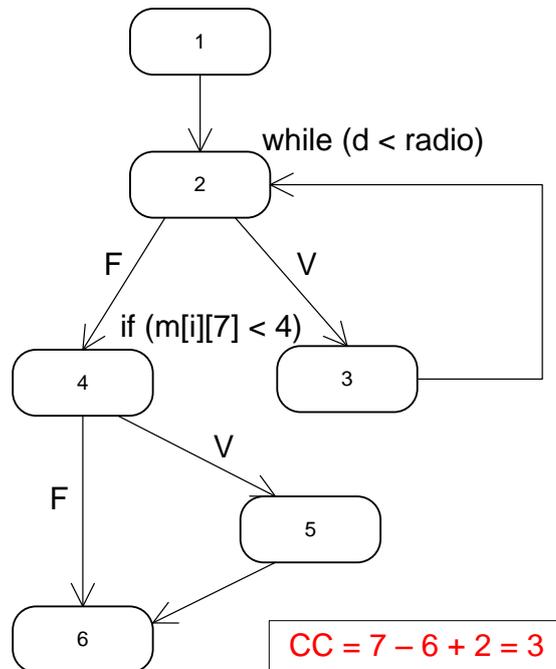


Figura 4.29: Grafo asociado a los caminos básicos (función calcular).

Como  $x_1, x_2, y_1, y_2$  representan 2 puntos en el plano cartesiano, serán siempre números positivos donde  $0 \leq x \leq 640$  y  $0 \leq y \leq 480$ , ya que la imagen capturada es de 640x480. Vamos a suponer además que el radio es 100 y que la matriz tiene los siguientes datos:

121	100	0	0	0	0	0	0
100	100	0	0	0	0	0	0

Vamos a considerar que el robot 1 está en la posición (121,100) y el robot 2 está en la posición (100,100), Se expone este caso porque es el más conflictivo que se podría dar, cuando los robots están uno al lado del otro. El radio de aceptación considera la distancia desde el centro robot hasta el borde del mismo, pero los demás colores no se encuentran en el borde sino más adentro, ese radio tiene una tolerancia mayor.

#### 4.7.4.1.2 Caminos a probar

$$C1 = 1 - 2 - 3 - 2 - 4 - 5 - 6$$

$$C2 = 1 - 2 - 4 - 6$$

$$C3 = 1 - 2 - 4 - 5 - 6$$

#### 4.7.4.1.3 Tabla de casos de prueba

A continuación se presenta la tabla 4.8 de casos de prueba para probar todos los caminos resultantes de las etapas anteriores. En el camino C2 hay que considerar que todos los colores estén asignados y su fila es [121, 100, 2, 1, 2, 3, 0, 4]. En C3 consideremos que la matriz que tenemos es la original.

Caminos probados	Dato de entrada x, y, color	Resultado esperado en la matriz	Resultado real
C1	105, 105, 1	[100, 100, 1, 0, 0, 0, 0, 1]	[100, 100, 1, 0, 0, 0, 0, 1]
C2	126, 100, 2	[121, 100, 2, 1, 2, 3, 0, 4]	[121, 100, 2, 1, 2, 3, 0, 4]
C3	116, 100, 3	[121, 100, 3, 0, 0, 0, 0, 1]	[121, 100, 3, 0, 0, 0, 0, 1]

Tabla 4.9: Casos de prueba función calcular.

#### 4.7.4.2 Clase Inferencia: Función detVelocidadDireccion

Esta función se encarga de determinar la velocidad, la dirección y el tiempo que se debe mover el robot (ver tabla 4.9)

```

void inferencia::detVelocidadDireccion() {
    int i = mDatosActual[0][4] - mDatosActual[0][3];
    int t = (int)sqrt(mDatosActual[0][0]);
    decision.setDireccion(ATRAS);
    if((i >= -10 && i <= 10)){
        decision.setVelocidadM1(110);
        decision.setVelocidadM2(110);
        decision.setTiempo((int)(t*1.5)); //t=d/v
    }else{
        if((i>10 && i<170) || (i<-190 && i>-350)){//Derecha
            if(i<0) i=360+i;
            decision.setVelocidadM1((int)(110 - ((95./159.)*(i-10))));
            decision.setVelocidadM2(110);
            decision.setTiempo((int)(t*1.7));
        }else{
            if((i>-170 && i<-10) || (i>190 && i<350)){//izquierda
                i=i<0?-i:360-i;
                decision.setVelocidadM1(110);
                decision.setVelocidadM2((int)(110-fabs((95./159.)*(i-10))));
                decision.setTiempo((int)(t*1.7));
            }else{//debe girar en 180
                decision.setVelocidadM1(10);
                decision.setVelocidadM2(110);
                decision.setTiempo(1300);
            }
        }
    }
}
}
}
}

```

Tabla 4.10: Código función “detVelocidadDiureccion”

Camino básico

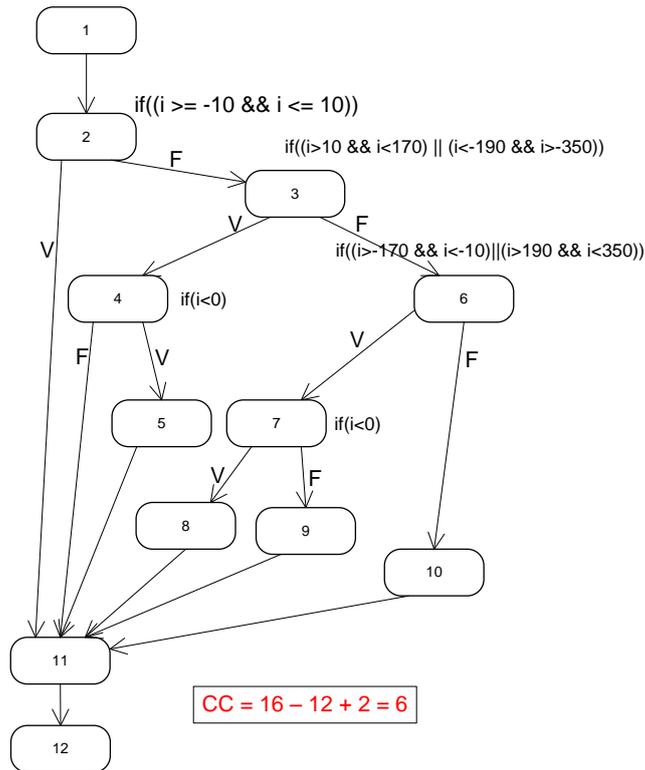


Figura 4.30: Grafo asociado a los caminos básicos (función detVelocidadDireccion).

4.7.4.2.1 Caminos a probar

C1 = 1 – 2 – 11 – 12

C2 = 1 – 2 – 3 – 4 – 11- 12

C3 = 1 – 2 – 3 – 4 – 5 – 11- 12

C4 = 1 – 2 – 3 – 6 – 7 – 8 – 11 – 12

C5 = 1 – 2 – 3 – 6 – 7 – 9 – 11 – 12

C6 = 1 – 2 – 3 – 6 – 10 – 11 – 12

4.7.4.2.2 Tabla de casos de prueba

A continuación se presenta la tabla 4.10 de casos de prueba para probar todos los caminos resultantes de las etapas anteriores. Los datos de entrada en este caso es una matriz con la distancia entre el robot y la pelota, la posición de la pelota (x, y), el ángulo del robot y el ángulo del robot respecto de la pelota. El resultado que entrega es la velocidad de cada motor (entre 0 y 127) y el tiempo de ejecución.

Caminos probados	Dato de entrada x, y, color	Resultado esperado en la matriz	Resultado real
C1	[90000, 320, 240, 50, 51]	110, 110, 450	110, 110, 450
C2	[90000, 320, 240, 50, 90]	110, 92, 510	110, 92, 510
C3	[90000, 320, 240, 130, -80]	110, 25, 510	110, 25, 510
C4	[90000, 320, 240, 50, -10]	80, 110, 510	80, 110, 510
C5	[90000, 320, 240, -150, 50]	90, 110, 510	90, 110, 510
C6	[90000, 320, 240, -85, 90]	10, 110, 1300	10, 110, 1300

Tabla 4.11: Casos de prueba función detVelocidadDireccion.

## **CAPITULO V - CONCLUSIONES**

## Capítulo V – Conclusiones

### 5.1 Conclusiones Generales

La primera generación de robots F180 presentada en este proyecto es el producto de un semestre de trabajo. En este tiempo se ha mejorado cada módulo hasta el punto de no presentar fallas detectables fácilmente como se ha demostrado. A continuación se presentan las conclusiones para cada módulo: visión artificial, estrategia y comunicación respectivamente.

#### 5.1.1 Respetto de la visión artificial

De la visión se puede decir que es una etapa compleja y muy sensible. El desarrollo de sus módulos implica la atención de factores ajenos a la programación, como la calidad de la cámara de video, la iluminación y la combinación o set de colores que se utilice para la identificación de los robots. Estos aspectos se deben tomar muy en cuenta, pues de ellos depende el nivel de ruido que afecte las capturas y al sistema en general.

En el caso de la cámara de video, se recomienda aquella con características mínimas de calidad como: modo *display off* completo, auto-foco automático rápido, resolución mínima de 640x480, mínima sensibilidad a la luz o a cambios de ella y que permita el modo de *ahorro de energía* apagado.

Con respecto a la iluminación, ésta no debe provenir de otras fuentes indirectas, que no sean las proporcionadas por la implementación. El motivo, es que estas fuentes provocan cambios sustanciales en la segmentación de los colores, haciendo que el sistema realice malas capturas del ambiente y los robots.

En relación a los patrones de colores empleados para la distinción de los robots, es recomendable escoger colores secundarios como lo son el cian y el magenta. Dado que los colores primarios son capturados con una leve baja (ruido) en su calidad.

### 5.1.2 Respecto de la estrategia

En cuanto a la estrategia se puede decir que es un módulo flexible, ya que permite incorporar diferentes tipos de estrategias, cambiando el funcionamiento de los robots. Por el momento se encuentran implementadas 3 de ellas, las cuales corresponden a seguir la pelota, tiro penal y perseguir al compañero.

Existen variadas formas para entregar decisiones de movimiento a los robots, las cuales pasan por cálculos físicos de cinemática y algoritmos de estimación, los cuales pueden dar mayor precisión y anticipación al movimiento.

En relación a los algoritmos de estimación podemos mencionar la implementación del filtro de kalman, para estimar el movimiento de la pelota en un tiempo  $t+1$  en base al tiempo  $t$  y  $t-1$ . Este algoritmo no pudo ser implementado por factores que se explican en el Capítulo X – Sección X.X, sin embargo es una alternativa muy recomendable en posteriores implementaciones, con un equipo completo.

### 5.1.3 Respecto de la comunicación

El módulo de comunicación no supone que los dispositivos de transmisión y recepción, realizan el intercambio de datos de manera efectiva. Por lo anterior es necesario contar con dispositivos confiables y rápidos, que aseguren la entrega fiable de información. Una solución a este problema podría ser el desarrollo de un protocolo que asegure el envío y recepción de los datos de manera rápida y fiable. Sin embargo, en el caso de este proyecto, se cuenta con dispositivos de radiofrecuencia que poseen un mecanismo de comprobación de paridad integrado, lo cual los exime de posibles errores o cambios que tergiversen los datos.

En el desarrollo del proyecto aparecieron detalles importantes, referentes a la comunicación y control de motores en el microcontrolador, que reducen la usabilidad y escalabilidad del sistema. Estos detalles hacen imposible realizar acciones paralelas en la programación de los dispositivos. Por lo anterior, el sistema queda limitado a realizar una sola tarea y no realizar otra hasta que ésta termina, i.e.: no se puede interrumpir una acción si se requiere.

Una solución a la problemática anterior sería intentar migrar los robots a otro tipo de microcontrolador que si permita este detalle.

## **5.2 Trabajos posteriores**

La intención del trabajo presentado es la conformación a largo plazo de un equipo que represente a la Universidad del Bío-Bío en la categoría Small Size de la competencia Robocup. Se pretende que se desarrolle, a partir de la base presentada en este trabajo, una copia o mejora de los robots construidos, una mejora paulatina de cada módulo y lo mas importante: el desarrollo de una estrategia sólida y completamente enfocada a un encuentro de fútbol de robots, con ambos equipos operativos.

Se recomienda también mejorar la electrónica y el diseño del robot para optimizar el movimiento del mismo. Además de incorporar un pateador frontal que lo dote de la capacidad de disparar la pelota, dar pases y driblar.

Como trabajo posterior a largo plazo se recomienda la formación de un grupo multidisciplinario de investigación y desarrollo, para la conformación de un equipo de robots f180 small size. Esta idea viene a tomar relevancia dada la importante atención de diferentes docentes de la Universidad del Bio Bio, en este proyecto.

## Bibliografía

- [LAR 99] Larman, Craig. 1999. UML Y PATRONES: introducción al análisis y diseño orientado a objetos. México, Prentice Hall. 536p.
- [PRA 97] Prasad R., 1997, Reseña de las Comunicaciones Personales Inalámbricas Perspectivas de Microondas, IEEE Communications Magazine, Volumen 9 , Páginas 104-108.
- [WHI 95] Steele R., Whitehead J., Wong W., 1995, Aspectos del Sistema Radio Celular, IEEE Communications Magazine, Volumen 5 , Páginas 80-86
- [SCO 01] Scott, Kendall, 2001, UML Explained, Addison-Wesley.
- [ROB 06] ROBOCUP. 2006. Bremen Germany. [En línea] <[www.robocup.org](http://www.robocup.org)> [consulta: 14 de noviembre 2006].
- [WIK 06] Wikipedia 2006. [En línea] <[www.wikipedia.es](http://www.wikipedia.es)> [consulta: noviembre 2006].
- [KEI 04] Keith Jack, 2004, Video Demystified, Cuarta Edición
- [IUM 06] Introducción a UML - <http://www-gris.det.uvigo.es/~avilas/UML/UML.html> [Visitada el día 12 – octubre – 2006]
- [VIC 06] Visión Computacional - [http://en.wikipedia.org/wiki/Computer\\_vision](http://en.wikipedia.org/wiki/Computer_vision) [Visitada noviembre 2006]
- [EDC 06] Especificaciones del Color - [http://www.f64digital.com/old/tutoriales/esp\\_color\\_I.html](http://www.f64digital.com/old/tutoriales/esp_color_I.html) [Visitada noviembre 2006]
- [TDC 06] Teoría del Color - [http://es.wikipedia.org/wiki/Teor%C3%ADa\\_del\\_color](http://es.wikipedia.org/wiki/Teor%C3%ADa_del_color) [Visitada en Octubre del 2006]
- [IMI 06] Introducción al Microcontrolador - <http://usuarios.lycos.es/sfriswolker/pic/uno.htm> [Visitada en 12 octubre del 2006]
- [MIC 06] Microcontrolador - <http://es.wikipedia.org/wiki/Microcontrolador> - [visitada el 12 octubre del 2006]

- [MIE 06] Microcotrollers - <http://en.wikipedia.org/wiki/Microcontroller> - [visitada el 12 octubre del 2006]
- [BLU 06] Comunicación Inalámbrica – BlueTooth - <http://www.enterate.unam.mx/Articulos/2004/octubre/bluetooth.htm> - [visitada 3 noviembre 2006]
- [WIF 06] Comunicación Inalámbrica – WIFI - <http://www.aulaclie.es/articulos/wifi.html> [ visitada 3 noviembre 2006]
- [INF 06] Red por infrarrojos - [http://es.wikipedia.org/wiki/Red\\_por\\_infrarrojos](http://es.wikipedia.org/wiki/Red_por_infrarrojos) - [visitada 4 noviembre 2006]
- [IDA 06] Infrared Data Association - [http://es.wikipedia.org/wiki/Infrared\\_Data\\_Association](http://es.wikipedia.org/wiki/Infrared_Data_Association) - visitada 4 noviembre]
- [PRP 06] Pruebas de Programas - <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm#s22> – [visitada el 5 de noviembre del 2006]
- [WIKL 06] Wikileatning 2006 <[www.wikilearning.org](http://www.wikilearning.org)> [Consulta: noviembre 2006]
- [MAES 95] Society of Mexican American Engineers and Scientists – MAES - <<http://www.maes-natl.org/>> [Consulta: noviembre 2006].
- [ENT 04] Internet, Cómputo y telecomunicaciones. Universidad nacional autónoma de México. <[www.enterate.unam.mx](http://www.enterate.unam.mx)> [Consultada: Noviembre 2006]
- [KID 94] Smith, David Canfield - Chipre, Allen – Spohrer, Jim – 1994, KidSim: programming agents without a programming language, EEUU, ACM Press.
- [AAT 98] N. R. Jennings and M. Wooldridge, 1998. Applications of Agent Technology, EEUU, Springer-Verlag.
- [IBMA 06] IBM Research - Intelligent Agents Project <[www.research.ibm.com/iagents/](http://www.research.ibm.com/iagents/)> [Consultada: noviembre 2006]
- [PRE 98] Pressman, Roger 1998. Ingeniería del Software: Un enfoque práctico Cuarta Edición. España, MacGraw-Hill. 581p.
- [KAL 03] Greg Welch and Gary Bishop, An Introduction to the Kalman Filter, May 23, 2003, EEUU.

## Abreviaturas

**BS2:** Basic Stamp 2

**UML:** Lenguaje de modelamiento unificado

**IA:** Inteligencia artificial

**IAD:** Inteligencia artificial distribuida

## ANEXOS

## Anexo A: Filtro de Kalman

### Filtro de Kalman

En 1960, R. E. Kalman publica un documento describiendo una solución recursiva al problema de filtrado lineal para datos discretos. Desde entonces, debido a los avances en la computación digital, el filtro de Kalman ha sido un tema de investigación y uso extenso, particularmente en el área de navegación autónoma o asistida [KAL 03].

El filtro de Kalman es un conjunto de ecuaciones matemáticas que proporcionan una eficiente solución computacional (recursiva) del método mínimos cuadrados. El filtro es de gran alcance puesto que: apoya estimaciones de estados pasados, presentes e incluso futuros, y puede llegar a ser preciso aun cuando la naturaleza del sistema modelado es desconocida [KAL 03].

Entrega el estimador óptimo de  $X$  dado un conjunto de medidas  $Z$  sólo si se cumplen las siguientes condiciones:

- El sistema es lineal
- El ruido asociado con el modelo del proceso y las mediciones es blanco y Gaussiano.

### El proceso para estimar

El filtro intenta estimar el estado  $x \in R^n$  de un proceso controlado en tiempo discreto que se representa por la ecuación de diferencia estocástica lineal

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, \text{ (Modelo; predicción)}$$

con una medida  $z \in R^n$  que es

$$z_k = Hx_k + v_k \text{ (Modelo de mediciones; corrección, actualización de mediciones)}$$

Donde

$x_k$ : estado en tiempo  $k$

$u_k$ : señal de entrada al sistema. (acciones de control)

$z_k$ : mediciones en tiempo  $k$  (datos de los sensores)

Las variables aleatorias  $w$  y  $v$  representan el ruido del proceso y de la medida (respectivamente) y se asumen independientes.

$w$  corresponde al ruido del proceso con covarianza  $Q$

$v$  corresponde al ruido en la medición con covarianza  $R$

Se asume que poseen las siguientes distribuciones:

$$p(w) \sim N(O, Q)$$

$$p(v) \sim N(O, R)$$

## Los orígenes del filtro

Se define:

$\hat{x}_k^- \in R^n$  como el estimador a priori del estado, i.e. función del conocimiento del proceso previo al paso  $k$ .

$\hat{x}_k \in R^n$  como el estimador a posteriori del estado, i.e. función de los pasos hasta  $k$  y de la medición efectuada en el paso  $k$ .

Los errores a priori y posteriori se definen como:

$$e_k^- \equiv x_k - \hat{x}_k^-,$$

$$e_k \equiv x_k - \hat{x}_k.$$

Las matrices de covarianza a priori y posteriori están dadas por:

$$P_k^- = E[e_k^- e_k^{-T}], \quad P_k = E[e_k e_k^T].$$

Para comenzar se debe encontrar una ecuación que exprese un estimador a posteriori del estado como una combinación lineal del estimador a priori del estado y la diferencia entre la medida actual y la predicción de la medida:

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-)$$

La diferencia  $(z_k - H\hat{x}_k^-)$  en la ecuación anterior corresponde a la innovación de la medición, o al residuo. El residuo refleja la diferencia entre medida predicha y la medida  $H\hat{x}_k^-$  y la medida  $z_k$ .

La matriz  $K$  de  $n \times m$  de la ecuación se obtiene para minimizar la covarianza del error a posteriori. Esta minimización se puede obtener primero reemplazando la ecuación en la expresión del error  $e_k$ , luego en la expresión de la covarianza del error a posteriori, obtener la derivada del resultado con respecto a  $K$ , se iguala el resultado con cero y se despeja  $K$ .

Es importante señalar que:

- Si la covarianza del error en la medición se aproxima a cero, la ganancia  $K$  otorga mayor peso a la innovación de la medición, i.e. se confía más en la medición. Específicamente:

$$\lim_{R_k \rightarrow 0} K_k = H^{-1}$$

- Si el estimador a priori de la covarianza del error se aproxima a cero, la ganancia  $K$  otorga menor importancia a la innovación de la medición. Específicamente:

$$\lim_{P_k^- \rightarrow 0} K_k = 0$$

El filtro de Kalman estima un proceso usando una forma de control de regeneración o retroalimentación: el filtro estima el estado de proceso en algún momento y después obtiene la regeneración bajo la forma de medidas (ruidosas). Como tal, las ecuaciones para el filtro de Kalman recaen en dos grupos: las ecuaciones de actualización del tiempo y ecuaciones de actualización de medidas. Las ecuaciones de actualización del tiempo son responsables de proyectar adelante (a tiempo) las estimaciones de la covariación del estado actual y del error para obtener las estimaciones a priori para el próximo paso. Las ecuaciones de actualización de medida son responsables de la regeneración, es decir, para incorporar una nueva medida en la estimación a priori para obtener una estimación posteriori mejorada [KAL 03].

Las ecuaciones discretas específicas para las actualizaciones del tiempo están dadas por:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

$$P_k^- = AP_{k-1}A^T + Q$$

Las ecuaciones discretas específicas para las actualizaciones de la medida son:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

$$P_k = (I - K_k H)P_k^-$$

### Funcionamiento del filtro de Kalman

A continuación un completo cuadro de operación del filtro de Kalman, utilizando las fórmulas anteriores.

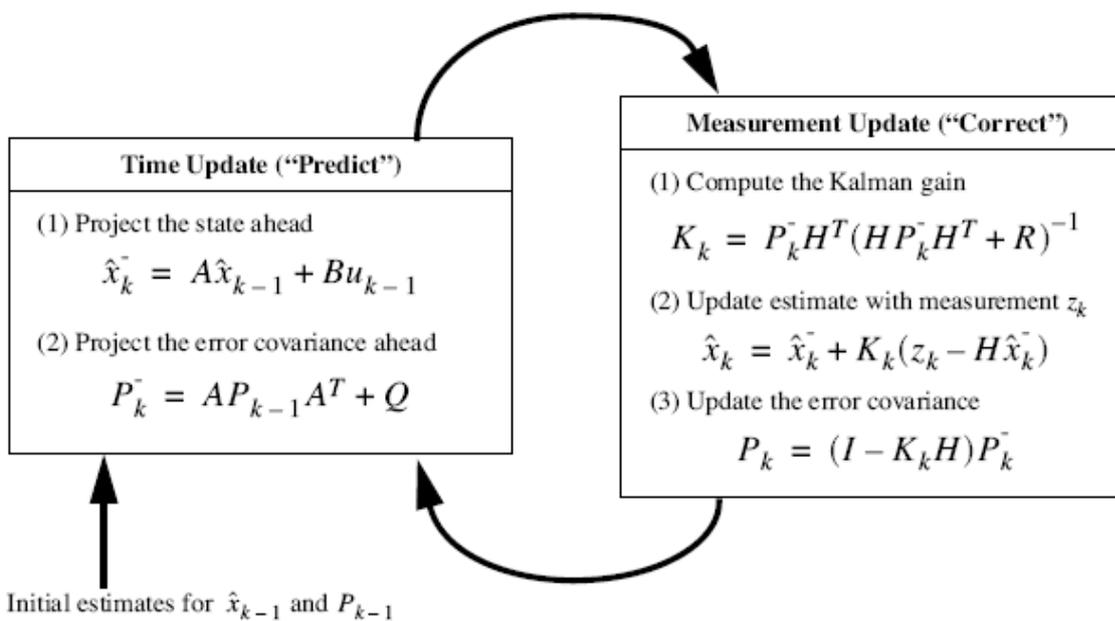


Figura M,n : Ciclo del algoritmo filtro de kalman.

Por último hay q considerar que:

6.  $R$  usualmente se mide antes de poner el filtro en operación. Esto es generalmente posible.
7. Determinar  $Q$  es, en general, más difícil.
8. También es posible ajustar los parámetros  $R$  y  $Q$  del filtro, empleando otro filtro de Kalman.

$P_k$  y  $K_k$  se estabilizarán rápidamente y encontrarán, en general, un valor de régimen permanente. Si esto ocurre, podrían determinarse antes de la ejecución del filtro y dejarse como constantes [KAL 03].