

**DESARROLLO DE PLANEACIÓN Y SEGUIMIENTO DE TRAYECTORIAS PARA  
UN VEHÍCULO AUTÓNOMO**

**KEVIN STEVEN SARMIENTO GAMBOA  
JAVIER EDUARDO VASQUEZ LAGOS  
OSCAR LEONARDO LAGUNA SUAREZ**

**UNIVERSIDAD PILOTO DE COLOMBIA  
FACULTAD INGENIERÍA MECATRÓNICA  
BOGOTÁ D.C.  
2015**

**DESARROLLO DE PLANEACIÓN Y SEGUIMIENTO DE TRAYECTORIAS  
PARA UN VEHÍCULO AUTÓNOMO**

**KEVIN STEVEN SARMIENTO GAMBOA  
JAVIER EDUARDO VASQUEZ LAGOS  
OSCAR LEONARDO LAGUNA SUAREZ**

**MONOGRAFÍA DE TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE  
INGENIERO MECATRÓNICO**

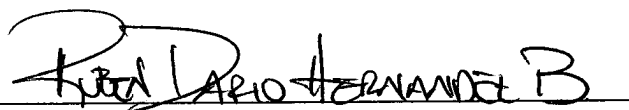
**Profesor Orientador**

**MS.C RUBÉN DARÍO HERNÁNDEZ BELEÑO PH.D ©**

**UNIVERSIDAD PILOTO DE COLOMBIA  
FACULTAD INGENIERÍA MECATRÓNICA  
BOGOTÁ D.C.  
2015**

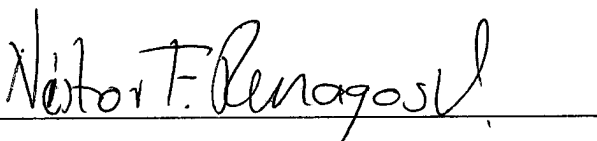
Nota de aceptación:

El trabajo de grado, titulado  
"DESARROLLO DE  
PLANEACIÓN Y  
SEGUIMIENTO DE  
TRAYECTORIAS PARA UN  
VEHÍCULO AUTÓNOMO"  
elaborado y presentado por los  
estudiantes **Kevin Sarmiento,**  
**Javier Vasquez** y **Oscar**  
**Laguna**, como requisito parcial  
para optar al título de Ingeniero  
Mecatrónico, fue aprobado por  
el Jurado Calificador



---

Firma del presidente del  
jurado



---

Firma del jurado



---

Firma del jurado

Bogotá D.C. 14 de Diciembre de 2015

## DEDICATORIA

Dedico este trabajo a mi padre Carlos Sarmiento y a mi madre Jeaneth Gamboa quienes a lo largo de mi vida me han acompañado en mi formación como persona integral; sus consejos y enseñanzas reflejan lo que soy actualmente. De igual manera dedico este trabajo a mi hermano Sebastian Sarmiento quien fue un apoyo incondicional y me acompañó durante este proceso.

Kevin Sarmiento G.

Dedico este trabajo a mi madre ya que ella fue el apoyo incondicional en la totalidad del desarrollo, además el acompañamiento en mi carrera por parte de ella ya que cada avance en el ejercicio académico fue posible gracias a su incondicionalidad, sabiduría, guía moral y el amor disponible en cualquier momento; cualidades que caracterizan una persona digna de un reconocimiento, digna de poder llamarse a sí misma, madre.

Javier Vasquez L.

Yo le dedico este trabajo principalmente a mi querida madre por guiarme y acompañarme por este, un trayecto más en mi vida que con esmero estoy culminando. También a mi hermano que siempre estuvo para colaborarme en sus tiempos libres para ayudarme con este arduo proyecto. Es uno de muchos escalones que aún faltan en la vida los cuales caminaré con seguridad y mucha voluntad.

Oscar Laguna S.

## **AGRADECIMIENTO**

Agradezco primordialmente a Dios ya que me brindó sabiduría, salud y capacidades para desarrollar el proyecto de manera exitosa y por permitirme alcanzar una meta más de mi proyecto de vida. A mis padres y hermano quienes fueron, son y serán siempre mi apoyo incondicional, ya que gracias a su amor y consejos, me han permitido desarrollarme como persona integral. Al ingeniero Rubén Hernández, por estar siempre dispuesto a colaborarnos y brindarnos su apoyo, sus conocimientos y experiencia fueron un gran apoyo en el desarrollo del proyecto de grado. De igual manera a la Universidad Piloto de Colombia en especial a la facultad de ingeniería mecatrónica, por la formación académica brindada.

Kevin Sarmiento G.

Agradezco a mi madre ya que ella fue el sustento económico y emocional de todo mi trabajo, agradezco el colocar mi bienestar y emociones primero que las de ella, agradezco todos los sacrificios hechos creyendo que valía la pena hacerlo, agradezco el convencimiento ciego hasta el día de hoy en que seré alguien merecedor de todas las acciones de ella. Agradezco inmensamente a la guía del ingeniero Rubén Hernández ya que gracias a él fueron posibles los resultados obtenidos, agradezco la confianza depositada en nosotros, el creer que éramos las personas indicadas para este trabajo y los futuros ingenieros capaces de afrontar cualquier adversidad, incluso cuando desfallecíamos. Agradezco la manera de guiarnos ya que esta fue más que de estudiante a profesor, esta fue como la de un amigo incondicional dispuesto a brindar su ayuda a cualquier momento, más que un amigo un hermano en el conocimiento y la vida.

Javier Vasquez L.

Yo le agradezco al cuerpo de docentes que me instruyeron, me guiaron en mi formación como profesional y como persona. Aquellas personas que se esmeran día a día por cumplir con su deber de enseñar, para que en un futuro no muy lejano ese trabajo se vea reflejado en futuras generaciones.

Muchas Gracias.

Oscar Laguna S.

# CONTENIDO

	<b>Pág.</b>
1 INTRODUCCIÓN .....	19
1.1 PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA .....	19
1.1.1 Descripción del problema .....	19
1.1.2 Formulación del problema .....	20
1.2 JUSTIFICACIÓN.....	20
1.2.1 Justificación social (Internacional).....	20
1.2.2 Justificación social (Nacional) .....	21
1.3 ANTECEDENTES.....	21
1.3.1 Historia de los vehículos autónomos.....	21
1.3.2 Vehículos autónomos en el sector militar.....	33
1.4 OBJETIVOS.....	35
1.5 ALCANCES Y LIMITACIONES .....	36
1.6 LÍNEA DE INVESTIGACIÓN DEL PROGRAMA.....	36
1.7 DISEÑO METODOLÓGICO .....	37
1.8 CONTRIBUCIONES .....	37
2 FUNDAMENTOS TEÓRICOS .....	38
2.1 DESCRIPCIÓN DE LA IMPLEMENTACIÓN .....	38
2.2 SISTEMA DE LOCALIZACIÓN (GPS).....	39
2.3 SISTEMA DE PERCEPCIÓN .....	45
2.3.1 Percepción sensorial.....	45
2.3.2 Percepción visual .....	48
2.4 SISTEMA DE CONTROL.....	51
2.5 PLANEAMIENTO DE TRAYECTORIA O RUTA.....	52
3 VISIÓN ARTIFICIAL.....	59
3.1 CONCEPTOS TEÓRICOS GENERALES.....	59

3.1.1	Imagen digital .....	59
3.1.2	Espectro visible .....	60
3.1.3	Ojo humano.....	60
3.1.3.1	Anatomía del ojo humano .....	61
3.1.3.2	Funcionamiento del ojo .....	62
3.1.4	Cámara .....	64
3.1.5	Similitudes entre el ojo humano y la cámara fotográfica .....	64
3.2	CONFIGURACIÓN BÁSICA DE UN SISTEMA DE VISIÓN ARTIFICIAL .....	66
3.3	REPRESENTACIÓN DE LAS IMÁGENES EN LOS COMPUTADORES DIGITALES.....	67
3.3.1	Imágenes binarias .....	68
3.3.2	Imágenes en escala de grises.....	68
3.3.3	Imágenes indexadas .....	69
3.3.4	Imágenes a color (RGB).....	70
3.4	ETAPAS BÁSICAS EN EL PROCESO DE VISIÓN ARTIFICIAL.....	71
3.5	VISIÓN ARTIFICIAL FRENTE A LA VISIÓN HUMANA.....	74
3.6	FILTRADO DE IMÁGENES DIGITALES.....	75
3.6.1	Técnicas de filtrado .....	75
3.6.1.1	Filtrado en el dominio del espacio.....	75
3.6.1.2	Filtrado en el dominio de la frecuencia .....	76
4	MODELADO MATEMÁTICO .....	77
4.1	MODELO CINEMÁTICO DEL VEHÍCULO.....	78
4.1.1	Modelo cinemático “bicicleta” .....	80
4.1.1.1	Ecuaciones características del sistema .....	83
4.2	SISTEMA DE DIRECCIÓN .....	89
4.2.1	Configuración del sistema de dirección.....	89
4.2.2	Restricción del modelo .....	90
4.3	VALIDACIÓN DEL MODELO CINEMÁTICO EN MATLAB® .....	92
4.3.1	Implementación.....	92
4.3.2	Parámetros del sistema.....	94

4.3.2.1	Posición y orientación inicial del vehículo en el plano.....	94
4.3.3	Verificación del modelo .....	96
5	INGENIERÍA DEL PROYECTO.....	99
5.1	ARQUITECTURA DEL SISTEMA.....	99
5.2	SELECCIÓN DE PLATAFORMA.....	100
5.3	ELECTRÓNICA EMBARCADA.....	101
5.3.1	Sistema electrónico de procesamiento.....	101
5.3.2	Actuadores .....	102
5.3.2.1	Actuadores eléctricos.....	103
5.3.3	Electrónica de potencia .....	106
5.3.4	Sensores .....	109
5.3.4.1	Sensores de movimiento y detección .....	109
5.3.5	Sistemas electrónicos implementados .....	116
5.4	Sistema de localización .....	119
5.4.1	Espacio coordinado de trabajo.....	119
5.4.2	Detección del vehículo en el espacio coordinado .....	122
5.4.2.1	Posición del vehículo .....	122
5.4.2.2	Orientación del vehículo.....	124
5.4.3	GPS y sistema de localización .....	125
5.5	PLANEACIÓN DE LA TRAYECTORIA.....	126
5.6	CONTROL DE ACTUACIÓN DEL SISTEMA.....	131
5.6.1	Algoritmo de seguimiento de trayectorias .....	131
5.6.2	Control de velocidad PID.....	135
5.7	PROGRAMACIÓN PRIMERA APLICACIÓN .....	139
5.8	DETECCIÓN DE ÁREA NAVEGABLE Y PEATONES .....	143
5.8.1	Detección de área navegable.....	143
5.8.2	Reconocimiento de peatones.....	148
5.9	PROGRAMACIÓN SEGUNDA APLICACIÓN.....	153
5.10	PROGRAMACIÓN CONTROL MANUAL.....	157
6	INTEGRACIÓN Y RESULTADOS .....	159



6.1	ESTRUCTURA MECÁNICA .....	159
6.1.1	Componentes de la plataforma .....	161
6.2	PRIMERA APLICACIÓN.....	162
6.2.1	Configuración de la plataforma .....	164
6.2.2	Interfaz gráfica.....	164
6.2.3	Primera trayectoria.....	165
6.2.4	Segunda trayectoria .....	168
6.2.5	Tercera trayectoria .....	169
6.2.6	Control de velocidad.....	170
6.3	SEGUNDA APLICACIÓN .....	171
6.3.1	Configuración de la plataforma .....	172
6.3.2	Interfaz gráfica.....	173
6.3.3	Detección de líneas.....	173
6.3.4	Detección de peatones.....	174
6.3.5	Sensor ultrasónico.....	178
6.4	CONTROL MANUAL .....	179
6.5	IMPLEMENTACIÓN DE LA SEGUNDA APLICACIÓN EN UN ENTORNO REAL 180	
	CONCLUSIONES .....	184
7	TRABAJOS FUTUROS .....	186
	ANEXO A.....	187
	ANEXO B.....	194
	ANEXO C.....	195
	ANEXO D.....	196
	ANEXO E.....	197
	ANEXO F .....	198
	ANEXO G .....	199
	ANEXO H.....	200
	ANEXO I .....	211
	ANEXO J .....	216

ANEXO K.....	217
ANEXO L.....	219
ANEXO M.....	220
BIBLIOGRAFÍA.....	222

## LISTA DE FIGURAS

<b>Figura 1.</b> Furgoneta usada por Dickmanns .....	22
<b>Figura 2.</b> Robot equilibrista de Dickmanns.....	23
<b>Figura 3.</b> Detección de líneas del camino .....	23
<b>Figura 4.</b> Chevrolet Navlab .....	24
<b>Figura 5.</b> Prototipo fabricado por Mercedes Benz .....	25
<b>Figura 6.</b> Vehículo del equipo Stanford .....	25
<b>Figura 7.</b> Desafío Darpa Urban .....	26
<b>Figura 8.</b> Área navegable del vehículo Google .....	27
<b>Figura 9.</b> Detección de obstáculos en la vía .....	28
<b>Figura 10.</b> Reconocimiento de peatón .....	28
<b>Figura 11.</b> Tren de vehículos autónomos de Volvo .....	29
<b>Figura 12.</b> Vehículo eléctrico autónomo Nissan .....	30
<b>Figura 13.</b> Vehículo con piloto automático Tesla.....	31
<b>Figura 14.</b> Detección de vehículos tesla .....	31
<b>Figura 15.</b> Vehículo autónomo Audi .....	32
<b>Figura 16.</b> Guardium UGV .....	33
<b>Figura 17.</b> Detección de soldado.....	34
<b>Figura 18.</b> Detección de patrones de huella.....	34
<b>Figura 19.</b> Detección de túneles.....	35
<b>Figura 20.</b> Metodología Implementada.....	37
<b>Figura 21.</b> Proceso general de la navegación autónoma. ....	39
<b>Figura 22.</b> Distancia del radio de señal .....	41
<b>Figura 23.</b> Intersección de radios de cobertura .....	41
<b>Figura 24.</b> Navegación por GPS .....	42
<b>Figura 25.</b> Ejes de referencia .....	42
<b>Figura 26.</b> Ángulo de orientación .....	43
<b>Figura 27.</b> Parametrización de imagen .....	43
<b>Figura 28.</b> Configuración de comunicación .....	44
<b>Figura 29.</b> Señal infrarroja.....	47
<b>Figura 30.</b> Señal del sensor ultrasónico .....	48
<b>Figura 31.</b> Visión interna .....	49
<b>Figura 32.</b> Visión estereoscópica .....	49
<b>Figura 33.</b> Detección de colores.....	50
<b>Figura 34.</b> Detección de bordes .....	50
<b>Figura 35.</b> Detección de objetos.....	51
<b>Figura 36.</b> Grafo visible .....	53

<b>Figura 37.</b> Diagrama de VORONI .....	54
<b>Figura 38.</b> Diagrama de espacio libre .....	54
<b>Figura 39.</b> Diagrama de conectividad.....	55
<b>Figura 40.</b> Diagrama de celdas .....	55
<b>Figura 41.</b> Diagrama de conectividad.....	56
<b>Figura 42.</b> Trayectoria Spline .....	56
<b>Figura 43.</b> Punto de inicio .....	57
<b>Figura 44.</b> Punto de destino .....	57
<b>Figura 45.</b> Trayectoria general .....	58
<b>Figura 46.</b> Representación de una imagen digital .....	59
<b>Figura 47.</b> Espectro electromagnético (luz visible).....	60
<b>Figura 48.</b> Anatomía del ojo humano .....	61
<b>Figura 49.</b> Formación de la imagen en el ojo .....	63
<b>Figura 50.</b> Córtex visual .....	63
<b>Figura 51.</b> Sensor fotosensible CCD y CMOS .....	64
<b>Figura 52.</b> Anatomía del ojo humano frente a la cámara fotográfica.....	65
<b>Figura 53.</b> Sistemas de un equipo de visión artificial .....	66
<b>Figura 54.</b> Subsistemas de un equipo de visión artificial.....	67
<b>Figura 55.</b> Imagen Binaria .....	68
<b>Figura 56.</b> Imagen en escala de grises .....	69
<b>Figura 57.</b> Imagen Indexada .....	69
<b>Figura 58.</b> Imagen a color (RGB) .....	70
<b>Figura 59.</b> Etapas en el proceso de Visión Artificial .....	71
<b>Figura 60.</b> Niveles de procesamiento.....	73
<b>Figura 61.</b> Matriz de filtrado en el dominio espacial .....	76
<b>Figura 62.</b> Coordenadas del vehículo respecto al sistema global de referencia ...	78
<b>Figura 63.</b> Configuración Ackerman.....	79
<b>Figura 64.</b> Geometría básica del modelo Ackerman .....	79
<b>Figura 65.</b> Modelo cinemático tipo Bicicleta .....	81
<b>Figura 66.</b> Geometría básica del modelo Bicicleta .....	82
<b>Figura 67.</b> Coordenadas de rueda delantera y trasera.....	84
<b>Figura 68.</b> Coordenadas de las ruedas para el modelo tipo bicicleta.....	85
<b>Figura 69.</b> Configuración del sistema de dirección.....	90
<b>Figura 70.</b> Restricción angular del modelo .....	91
<b>Figura 71.</b> Variables del Modelo.....	92
<b>Figura 72.</b> Implementación del modelo cinemático .....	93
<b>Figura 73.</b> Bloques integradores (posición inicial del vehículo).....	95
<b>Figura 74.</b> Bloque integrador (orientación inicial del vehículo).....	95
<b>Figura 75.</b> Diagrama de bloques del modelo cinemático .....	96

<b>Figura 76.</b> Simulación No.1 (Giro izquierda) .....	97
<b>Figura 77.</b> Simulación No.2 (Línea Recta) .....	97
<b>Figura 78.</b> Simulación No.3 (Giro Derecha) .....	98
<b>Figura 79.</b> Arquitectura del sistema.....	99
<b>Figura 80.</b> Integración de control.....	101
<b>Figura 81.</b> Motor faulhaber 12v .....	103
<b>Figura 82.</b> Señal para sentido de giro .....	104
<b>Figura 83.</b> Circuito interno de un servomotor .....	105
<b>Figura 84.</b> Partes de un servomotor .....	105
<b>Figura 85.</b> Speed Control .....	106
<b>Figura 86.</b> Regulador 7805K .....	106
<b>Figura 87.</b> Diagrama de bloques del regulador .....	107
<b>Figura 88.</b> Esquemático de circuito .....	108
<b>Figura 89.</b> Diseño de circuito impreso .....	108
<b>Figura 90.</b> Encoder óptico Pololu .....	109
<b>Figura 91.</b> Circuito del encoder óptico.....	110
<b>Figura 92.</b> Señal sin calibrar.....	111
<b>Figura 93.</b> Señal calibrada .....	111
<b>Figura 94.</b> Datos de la señal .....	112
<b>Figura 95.</b> Montaje del encoder.....	113
<b>Figura 96.</b> Caracterización del encoder .....	113
<b>Figura 97.</b> Sensor hc-sr04.....	115
<b>Figura 98.</b> Diagrama de funcionamiento .....	115
<b>Figura 99.</b> Configuración de primera aplicación .....	117
<b>Figura 100.</b> Configuración de segunda aplicación .....	118
<b>Figura 101.</b> Error de posicionamiento del GPS .....	119
<b>Figura 102.</b> Cámara dispuesta en un punto elevado.....	120
<b>Figura 103.</b> Coordenadas captadas por la cámara .....	121
<b>Figura 104.</b> Resolución y distancias captadas por la cámara .....	121
<b>Figura 105.</b> Marcadores de posicionamiento del vehículo .....	122
<b>Figura 106.</b> Identificación de contornos de los marcadores de posicionamiento .....	123
<b>Figura 107.</b> Centroides del vehículo .....	123
<b>Figura 108.</b> Orientación del vehículo (diferentes posiciones).....	124
<b>Figura 109.</b> GPS actual y sistema de localización .....	125
<b>Figura 110.</b> Planeamiento de trayectoria punto a punto.....	126
<b>Figura 111.</b> Segmentos de la trayectoria definida por funciones.....	127
<b>Figura 112.</b> Trayectoria cambio de carril mediante Spline cúbica .....	129
<b>Figura 113.</b> Trayectorias definidas .....	130
<b>Figura 114.</b> Algoritmo de control .....	132

<b>Figura 115.</b> Coordenadas de la trayectoria .....	133
<b>Figura 116.</b> Orientación y corrección de dirección .....	133
<b>Figura 117.</b> Velocidad .....	134
<b>Figura 118.</b> Datos de posición y orientación del vehículo .....	134
<b>Figura 119.</b> Diagrama de polos y ceros .....	135
<b>Figura 120.</b> Respuesta del sistema por PID Ziegler-Nichols .....	136
<b>Figura 121.</b> Respuesta del sistema por PID lugar de las raíces.....	138
<b>Figura 122.</b> Diagrama de estabilidad de polos y ceros .....	139
<b>Figura 123.</b> Proceso de extracción de características para detección de líneas en la vía .....	143
<b>Figura 124.</b> Imagen de la vía a escala de grises.....	144
<b>Figura 125.</b> Detección de contornos de la vía con operador Canny.....	145
<b>Figura 126.</b> Ajuste del umbral de la imagen .....	145
<b>Figura 127.</b> Conversión hacia el espacio paramétrico .....	147
<b>Figura 128.</b> Detección de líneas de la vía .....	147
<b>Figura 129.</b> Proceso de extracción de características para detección de peatones .....	148
<b>Figura 130.</b> Desarrollo del algoritmo en imagen de referencia.....	150
<b>Figura 131.</b> Desarrollo del descriptor HOG .....	151
<b>Figura 132.</b> Detección de peatón en tiempo real.....	151
<b>Figura 133.</b> Distancias de detección del peatón.....	152
<b>Figura 134.</b> Detección de peatón en tiempo real.....	153
<b>Figura 135.</b> Plataforma adoptada.....	159
<b>Figura 136.</b> Plataforma adoptada (vistas) .....	160
<b>Figura 137.</b> Vista superior .....	161
<b>Figura 138.</b> Componentes electrónicos de la plataforma .....	162
<b>Figura 139.</b> Configuración primera aplicación .....	163
<b>Figura 140.</b> Configuración primera aplicación (ambiente real).....	163
<b>Figura 141.</b> Configuración de la plataforma para primera aplicación .....	164
<b>Figura 142.</b> Interfaz primera aplicación .....	165
<b>Figura 143.</b> Ejecución trayectoria cambio de carril.....	166
<b>Figura 144.</b> Trayectoria cambio de carril determinada mediante algoritmo.....	167
<b>Figura 145.</b> Trayectoria Spline vs trayectoria simulada.....	167
<b>Figura 146.</b> Ejecución trayectoria giro derecha .....	168
<b>Figura 147.</b> Ejecución trayectoria giro izquierda .....	169
<b>Figura 148.</b> RPM de acuerdo a dirección de las ruedas delanteras.....	170
<b>Figura 149.</b> Velocidad del vehículo respecto al ángulo de dirección.....	171
<b>Figura 150.</b> Configuración segunda aplicación .....	171
<b>Figura 151.</b> Escenario de implementación .....	172

<b>Figura 152.</b> Configuración segunda aplicación .....	172
<b>Figura 153.</b> Interfaz segunda aplicación .....	173
<b>Figura 154.</b> Detección del área navegable.....	174
<b>Figura 155.</b> Identificación de peatón .....	175
<b>Figura 156.</b> Identificación de múltiples peatones .....	175
<b>Figura 157.</b> Identificación de múltiples peatones con error de posición .....	176
<b>Figura 158.</b> Contorno de persona a una distancia lejana.....	176
<b>Figura 159.</b> Contorno de persona a una distancia media.....	177
<b>Figura 160.</b> Contorno de persona a una distancia cercana.....	177
<b>Figura 161.</b> Proceso de detención del vehículo de acuerdo a contorno HOG ....	178
<b>Figura 162.</b> Obstáculo repentino .....	178
<b>Figura 163.</b> Gráfica de PWM y sensor ultrasónico .....	179
<b>Figura 164.</b> Interfaz control manual.....	180
<b>Figura 165.</b> Detección de peatones en la vía .....	180
<b>Figura 166.</b> Representación de la aplicación HOG .....	182
<b>Figura 167.</b> Representación de líneas en un ambiente nocturno .....	182
<b>Figura 168.</b> Representación de la aplicación detección de líneas en la vía .....	183

## LISTA DE TABLAS

Tabla 1. Sensores.....	46
Tabla 2. Similitudes del ojo humano frente a la cámara fotográfica.....	65
Tabla 3. Ventajas de la visión artificial frente a la visión humana .....	74
Tabla 4. Nomenclatura del sistema.....	83
Tabla 5. Características de las plataformas.....	100
Tabla 6. Especificaciones de los sistemas embebidos .....	102
Tabla 7. Características eléctricas del regulador 7805k .....	107
Tabla 8. Características del sensor hc-sr04.....	116
Tabla 9. Especificaciones de la cámara IP WiFi Foscam FI8918W .....	120
Tabla 10. Características dimensionales de la plataforma.....	161



## RESUMEN

La robótica móvil ofrece una amplia variedad de aplicaciones, gran parte de estas están enfocadas a la industria del sector automotor, tal como la navegación autónoma y la asistencia en la conducción. El presente trabajo monográfico tiene como objetivo la implementación del control enfocado a la planeación y seguimiento de trayectorias, además de esto se expone el desarrollo de una aplicación para la detección del área navegable y peatones presentes en la vía. En primera instancia se referencian los desarrollos más representativos a lo largo de la historia en este campo, con el fin de conocerlos e interpretarlos.

En capítulos posteriores se presentan los fundamentos teóricos necesarios para la ejecución del proyecto, partiendo de los métodos a implementar; de acuerdo a esto se modela la cinemática de la plataforma seleccionada. Con el fin de ejecutar una trayectoria deseada se hace necesaria la simulación de un GPS para lo cual se incursiona en la visión artificial. Partiendo de esto se expone el desarrollo ingenieril pertinente para las dos aplicaciones mencionadas, finalmente se validan los desarrollos en ambientes predeterminados dando a conocer los resultados obtenidos.

**Palabras clave:** Trayectorias, área navegable, peatones, cinemática, GPS, visión artificial.

## ABSTRACT

Mobile Robotics offers a wide variety of applications, great part of these are focused on the automotive industry, such as autonomous navigation and driving assistance. The present monograph aims to focus the implementation of planning and trajectory tracking control in addition to this the development of an application for the detection of navigable area and present pedestrians on the road is exposed. In the first instance the most representative developments are referenced throughout history in this field, in order to know and interpret.

In later chapters, the theoretical foundations necessary for project implementation are presented, based on the methods to be implemented; accordingly the kinematics of the chosen platform is modeled. In order to execute a wished path, the simulation of a GPS becomes necessary, for which ventures into artificial vision. Departing from this the relevant engineering development of the two applications is exposed; finally the developments are validated in predetermined environments announcing the obtained results.

**Keywords:** Paths, navigable area, pedestrians, kinematics, GPS, artificial vision.

# 1 INTRODUCCIÓN

La robótica móvil ha incursionado a gran escala en la creación de vehículos autónomos, los cuales han tenido un gran impacto en la industria y en el sector militar. Los robots autónomos sirven de inspiración para la creación de nuevas aplicaciones que permitan asistir y generar confort al conductor, generando vías inteligentes o sistemas vehiculares netamente autónomos, dando como resultado grandes avances en aspectos como el reconocimiento de terrenos, estacionamiento automático, reconocimiento de obstáculos (peatones, otros vehículos, señales de tránsito), entre otros.

El presente documento muestra el desarrollo de dos aplicaciones características de los vehículos autónomos, mediante la implementación mecatrónica (basado en sensores, sistemas embebidos y sistemas de visión computacional) en un prototipo a escala, con características similares a los automóviles convencionales.

La primera aplicación (principal) presenta el desarrollo de un algoritmo para seguimiento de trayectorias (previamente estipuladas), en un ambiente predeterminado libre de obstáculos. En primera instancia el algoritmo se implementa en el software de simulación de Matlab® y posteriormente se valida mediante el desarrollo de un sistema de localización simulado por medio de una cámara externa (proporcionando al vehículo la localización y orientación respecto a un sistema de referencia). En la segunda aplicación, se muestra la detección del área navegable mientras el vehículo transita sobre un camino predefinido y reconoce los peatones presentes en la vía, por medio de una cámara embarcada en el vehículo, realizando la detención de su movimiento en caso dado de que detecte un obstáculo a una distancia predefinida.

La programación de los algoritmos y procesamiento digital de imágenes se desarrollan en lenguaje C++ usando las librerías de OpenCV en el sistema embebido pcDuino/Arduino, junto con ello se desarrolla una interfaz gráfica la cual permite al usuario visualizar la ejecución de la trayectoria realizada por el vehículo junto con los parámetros proporcionados por el sistema de localización simulado y de igual forma los algoritmos de procesamiento de imagen para la identificación de vías y peatones.

## 1.1 PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA

### 1.1.1 Descripción del problema

El desarrollo de la robótica móvil a través del tiempo, ha tenido un avance significativo en la aplicación de robots autónomos, esto sirve como referente, en la búsqueda de crear sistemas que tornen los vehículos cada vez más

independientes, es decir, que la necesidad de ser intervenidos por factor humano sea escasa o nula. En la actualidad se han desarrollado aplicaciones en vehículos autónomos orientados a generar respaldo y asistencia al momento de conducir, pero hasta ahora este tipo de tecnología es utilizada en países desarrollados como Estados Unidos, Alemania, Países Bajos, entre otros<sup>1</sup>.

De acuerdo a ello se desarrolla en una plataforma robótica terrestre (con características semejantes a las de un vehículo real) aplicaciones que permiten ejecutar una trayectoria de forma acertada, para que pueda desplazarse correctamente desde un punto inicial a un punto final, establecido en un ambiente predeterminado. Ya que uno de los mayores inconvenientes es, “*conseguir que un robot se mantenga localizado mientras se desplaza a través de su entorno*”<sup>2</sup>. Para esto desarrolla un método que localiza al vehículo y le permite ejecutar la trayectoria satisfactoriamente.

De igual manera con la intención de generar asistencia en la conducción se aplica el procesamiento digital de imágenes para la identificación de líneas del camino y peatones en la vía.

### **1.1.2 Formulación del problema**

El problema se centra en el desarrollo de algoritmos que permitan a un robot móvil terrestre realizar el seguimiento de trayectorias en un ambiente predeterminado y de igual manera contribuir a la asistencia de la conducción mediante la detección del área navegable y los posibles peatones que se encuentren presentes en la vía.

## **1.2 JUSTIFICACIÓN**

La importancia del desarrollo del proyecto radica en la necesidad de aplicar los conocimientos de la robótica móvil enfocados principalmente en el desarrollo de aplicaciones en el campo automotor, orientado hacia minimizar la intervención humana en la conducción y generar tareas de asistencia al conductor. De igual forma se busca que por medio de este tipo de desarrollos puedan ser aplicados en el futuro y puedan reducir los índices de accidentalidad relacionados con el tránsito con la implementación de tecnologías de bajo costo.

### **1.2.1 Justificación social (Internacional)**

La seguridad a la hora de manejar es importante para el conductor como también para el peatón, según la OMS, en la actualidad los accidentes relacionados con el

---

<sup>1</sup> TICbeat. Alemania y Países Bajos permitirán probar coches autónomos en sus carreteras. [en línea] <<http://www.ticbeat.com/innovacion/alemania-paises-bajos-permitiran-probar-coches-autonomos-en-sus-carreteras/>>

<sup>2</sup> Fernández Caramés, Carlos. Técnicas de navegación para un robot móvil utilizando sistemas de razonamiento espacial. Universidad de Salamanca. 2012. p. 4

transito ocupan el octavo lugar como la mayor causa de muerte prematura en la población mundial<sup>3</sup> y aún con los desarrollos obtenidos actualmente se están presentando inconvenientes relacionados a la funcionalidad de estos vehículos. Según Guerrero<sup>4</sup>, las estadísticas muestran que aun con los desarrollos obtenidos, los carros autónomos también tienen la posibilidad de chocar, lo cual genera una reflexión acerca de los métodos que se puede implementar con el fin de orientar en su trayectoria a los vehículos autónomos.

## 1.2.2 Justificación social (Nacional)

En Colombia, de acuerdo a un estudio realizado por el Grupo Centro de Referencia Nacional sobre Violencia, la clase más representativa en accidentes de tránsito es el “choque entre vehículos” seguido del “atropellamiento”, estas dos formas enmarcan el 77.61% de los casos fatales y el 89.61% de las lesiones<sup>5</sup>.

De igual forma en 2013 se registraron 52 casos fatales y 1590 no fatales relacionados a choque entre automóviles, concluyendo que el peatón posee un alto riesgo de accidentalidad ya que abarca el 29.27% de las muertes en accidentes de tránsito<sup>6</sup>. Estos indicadores podrían minimizar mediante la implementación a futuro de sistemas autónomos que tengan la percepción del ambiente y puedan controlar la distancia entre vehículos y de igual forma tengan la capacidad de reconocer los peatones y obstáculos existentes en la vía.

## 1.3 ANTECEDENTES

### 1.3.1 Historia de los vehículos autónomos

A lo largo de la historia la robótica móvil se ha enfrentado a diversos desafíos, desde los primeros robots móviles creados como experimentos mecánicos hasta los vehículos autónomos que se están generando hoy en día; la idea de un vehículo sin conductor se ha propuesto desde hace mucho tiempo, tanto así que el primer vehículo sin conductor fue presentado por la empresa fabricante de automóviles *White Company* en los años veinte, siendo este de los primeros prototipos en la historia<sup>7</sup>.

---

<sup>3</sup> Organización mundial de la salud. Estadísticas sanitaria mundiales 2014: una mina de información sobre salud pública mundial, 2014.

<sup>4</sup> Guerrero, Juan Felipe. Estadísticas muestran que lo carros autónomos también se estrellan. 2015. [en línea] < <http://www.enter.co/cultura-digital/autotecnologia/cuatro-accidentes-de-autos-autonomos-reportados-desde-septiembre/>>

<sup>5</sup> Vargas Castillo, Diego Alfonso. Comportamiento de muertes y lesiones por accidente de transporte, Colombia, 2013. p. 185-186.

<sup>6</sup> Ibid., p. 188.

<sup>7</sup> Bejarano, Pablo. La historia de los coches autónomos es más larga de lo que se podía esperar. 2013. [en línea] <<http://blogthinkbig.com/historia-de-los-coches-autonomos/>>

El carro presentado estaba configurado con actuadores mecánicos controlados por radiofrecuencia el cual fue noticia en los diarios de esa época denotándolo como: “Un coche de motor fantasma se aparecerá en las calles de Milwaukee hoy. Sin conductor; encenderá su propio motor, tirará de su embrague, girará su volante, tocará su claxon y quizás incluso replique al policía de la esquina”<sup>8</sup>, desde esa época los avances en todos los campos de la robótica han sido extraordinarios ya que cada día se generan tecnologías aplicables a la meta total de un vehículo sin tripulante, capaz de definir caminos y tomar decisiones de manejo tal como lo haría un ser humano.

En la línea de tiempo, se encuentra un avance importante para la robótica móvil fue un logro hecho en Alemania por la Universidad de Múnich en la cual el profesor Dickmanns, experto en inteligencia artificial implementó métodos como la visión sacádica (movimientos rápidos de visión), cálculos probabilísticos y computación paralela lo cual facilita mucho el control del vehículo, además de lo anterior implementó procesadores con múltiples núcleos lo que lo llevó a la creación de una furgoneta mercedes como se observa en la (figura 1), capaz de viajar a 96km/h<sup>9</sup>.

**Figura 1.** Furgoneta usada por Dickmanns



**Fuente:** Ernst Dickmanns, el desconocido padre alemán de los coches inteligentes. URL: [http://www.eldiario.es/hojaderouter/tecnologia/Ernst\\_Dickmanns-vehiculo-autonomo-inteligente\\_0\\_382511814.html](http://www.eldiario.es/hojaderouter/tecnologia/Ernst_Dickmanns-vehiculo-autonomo-inteligente_0_382511814.html)

Después de este logro Dickmanns se convirtió en investigador de la Universidad Bundeswehr de Múnich y años más tarde integrando sus conocimientos en visión artificial con los primeros procesadores Intel, logró hacer un robot equilibrista el cual se movía a la derecha o izquierda para devolver a la posición de reposo un bastón situado encima de él, como se puede ver en la (figura 2).<sup>10</sup>

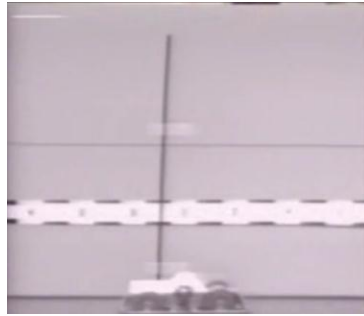
---

<sup>8</sup> Ibid., [en línea]

<sup>9</sup> Ernst Dickmanns, el desconocido padre alemán de los coches inteligentes. El diario. [en línea] < [http://www.eldiario.es/hojaderouter/tecnologia/Ernst\\_Dickmanns-vehiculo-autonomo-inteligente\\_0\\_382511814.html](http://www.eldiario.es/hojaderouter/tecnologia/Ernst_Dickmanns-vehiculo-autonomo-inteligente_0_382511814.html) > Fecha de consulta 17/05/15

<sup>10</sup> Ibid., [en línea]

**Figura 2.** Robot equilibrista de Dickmanns



**Fuente:** Ernst Dickmanns, el desconocido padre alemán de los coches inteligentes [Video]. Disponible en: [http://www.eldiario.es/hojaderouter/tecnologia/Ernst\\_Dickmanns-vehiculo-autonomo-inteligente\\_0\\_382511814.html](http://www.eldiario.es/hojaderouter/tecnologia/Ernst_Dickmanns-vehiculo-autonomo-inteligente_0_382511814.html)

Este avance lo llevó a realizar los primeros filtros para el procesamiento de la imagen y detección de objetos para después implementarlos en su camioneta, la implementación de estos nuevos algoritmos de visión dinámica lo llevaron a generar una versión mejorada de su anterior proyecto, el cual ya tenía la capacidad de detectar las líneas del camino y seguir la trayectoria del mismo, esto era posible gracias a que se establecía un punto en el centro de la imagen y otro que detectara la línea del camino, con estos dos datos se calculaba la distancia entre los dos puntos y se efectuaba la acción de corrección en la dirección si era necesario, tal como se observa en la (figura 3).<sup>11</sup>

**Figura 3.** Detección de líneas del camino



**Fuente:** Ernst Dickmanns, el desconocido padre alemán de los coches inteligentes [Video]. Disponible en: [http://www.eldiario.es/hojaderouter/tecnologia/Ernst\\_Dickmanns-vehiculo-autonomo-inteligente\\_0\\_382511814.html](http://www.eldiario.es/hojaderouter/tecnologia/Ernst_Dickmanns-vehiculo-autonomo-inteligente_0_382511814.html)

El investigador siguió creando algoritmos cada vez más complejos lo que permitió finalmente robustecer los algoritmos de visión computacional para aumentar su

---

<sup>11</sup> Ibid., [en línea]

capacidad de no solo detectar las líneas del camino sino también los obstáculos en el mismo tales como otros coches, botes de basura, entre otros. La detección de estos obstáculos se efectúa para calcular la distancia de la camioneta hasta dichos obstáculos y esquivarlos de no ser vehículos, de lo contrario se mantenía a una distancia promedio del objetivo; con el desarrollo de este algoritmo se dio la idea de la posibilidad de adelantar en los tramos posibles, para lo cual comparaba los dos carriles buscando vehículos, de no encontrarlos tomaba la decisión de adelantar o no, de lo contrario calculaba la distancia entre la camioneta y los dos vehículos y cambiaba de carril.

Casi en la misma época Navlab en la Universidad Carnegie Mellon desarrollaba otro prototipo de vehículo autónomo con la ayuda de GPS, el primer prototipo que se realizó solo tenía la capacidad de recorrer caminos delimitados por líneas a una velocidad de treinta kilómetros por hora, las tecnologías usadas para el desarrollo del primer prototipo eran bastante limitadas ya que contaban con procesadores de ocho bits lo que hacía de la captura de la imagen fuera un problema<sup>12</sup>.

Después del segundo modelo PANS financió la investigación y regaló los dispositivos necesarios para realizar el control óptimo del vehículo, que para este caso se desarrolló una van Chevrolet como se muestra en la (figura 4), este vehículo se equipó con un GPS, giroscopio, encoder, y una laptop para efectuar el control de todas las variables, después de este prototipo se generaron tres más haciendo grandes avances en el posicionamiento por GPS y el reconocimiento del entorno así como el área navegable y obstáculos estáticos como también móviles<sup>13</sup>.

**Figura 4.** Chevrolet Navlab



**Fuente.** Jochem T., Pomerleau D., Kumar B., Armstrong J. PANS: A Portable Navigation Platform, 1995. URL: [http://www.cs.cmu.edu/~tjochem/nhaa/navlab5\\_details.html](http://www.cs.cmu.edu/~tjochem/nhaa/navlab5_details.html)

Después de esto Mercedes apoyó la investigación de vehículos autónomos con el fin de reducir el tráfico en las ciudades, el prototipo fue apodado "VAMP" el cual

<sup>12</sup> Jochem Todd, Pomerleau Dean, Kumar Bala, Armstrong Jeremy. PANS: A Portable Navigation Platform, 1995. [en línea] < [http://www.cs.cmu.edu/~tjochem/nhaa/navlab5\\_details.html](http://www.cs.cmu.edu/~tjochem/nhaa/navlab5_details.html) > Fecha de consulta 20/05/15

<sup>13</sup> Ibid., [en línea]



contaba con una velocidad máxima de cien kilómetros por hora implementando las mismas tecnologías de los modelos anteriores como se observa en la (figura 5).<sup>14</sup>

**Figura 5.** Prototipo fabricado por Mercedes Benz



*Fuente.* Pete. *Self-Drive Cars and You: A History Longer than You Think*, 2014. URL: <http://www.velocetoday.com/self-drive-cars-and-you-a-history-longer-than-you-think/>

Posterior a ello, DARPA (Defense Advanced Research Projects Agency) genera un desafío de vehículos autónomos como incentivo a la creación de los mismos. El desafío consiste en llegar de un punto a otro con algunos puntos intermedios sin la intervención humana en ningún aspecto, el desafío se dio como primera instancia en el desierto de Mojave en el año 2004 el cual tenía un trayecto de 142 millas y participaron un total de quince equipos, de los cuales ningún vehículo logró completar el circuito, sin embargo en 2005 Stanford Racing Team (figura 6), finalizó el recorrido de 131,2 millas en 6 horas 54 minutos<sup>15</sup>.

**Figura 6.** Vehículo del equipo Stanford



**Fuente.** Stanford Racing Team. *Stanford Racing Team's Entry In The 2005 DARPA Grand Challenge*. URL: <http://www.cs.stanford.edu/people/dstavens/darpa/Stanford.pdf>

<sup>14</sup> Pete. *Self-Drive Cars and You: A History Longer than You Think* 2014. [en línea] <http://www.velocetoday.com/self-drive-cars-and-you-a-history-longer-than-you-think/> >

<sup>15</sup> Stanford Racing Team. *Stanford Racing Team's Entry In the 2005 DARPA Grand Challenge*.

En 2007 se creó el reto urbano llamado DARPA Urban Challenge como se muestra en la (figura 7), el cual simulaba entornos urbanos. Los ejercicios consistían en misiones simuladas de transporte de suministros y al igual que el desafío anterior se debía tener en cuenta el ambiente dinámico entorno<sup>16</sup>.

**Figura 7.** Desafío Darpa Urban



**Fuente.** Hundelshausen, F. Team AnnieWAY's Autonomous System for the 2007 DARPA Urban Challenge. URL: [https://www.unibw.de/tas/institut/mitarbeiter/alumni/prof\\_von\\_hundelshausen/publications/kammel-jfr08-annieway-system.pdf](https://www.unibw.de/tas/institut/mitarbeiter/alumni/prof_von_hundelshausen/publications/kammel-jfr08-annieway-system.pdf)

Al finalizar el desafío DARPA, la Universidad de Stanley junto con la compañía Google se fusionaron en un proyecto realmente ambicioso que tenía como meta final un vehículo autónomo capaz de desenvolverse en el entorno real de la ciudad teniendo en cuenta todas las variables que podrían presentarse en la conducción cotidiana de un ser humano, los integrantes del equipo ganador del desafío DARPA junto con quince ingenieros más se dieron a la tarea no sólo de perfeccionar las tecnologías y métodos implementados sino también de proponer nuevos sistemas.

Finalmente en el año 2014, Google presenta un automóvil que se conduce solo innovando y planteando un modelo radical no solo de la concepción que se tenía de los vehículos autónomos sino también del diseño del automóvil, dicho automóvil es un sedán pequeño con capacidad para dos tripulantes y capaz de viajar a ciento treinta kilómetros por hora. No solo tuvieron en cuenta la meta principal de que se condujera solo, sino también la comodidad y estética del carro.

En el control de movimiento usaron actuadores hidráulicos controlados con una computadora que además de controlar, analiza cada una de las variables a su alrededor, la autonomía del proyecto radica principalmente en el posicionamiento con GPS el cual no solo les brinda los datos o imágenes del camino, por medio del procesamiento de imagen y la reconstrucción del entorno. La computadora detecta

---

<sup>16</sup> Hundelshausen, Felix von. Team AnnieWAY's Autonomous System for the 2007 DARPA Urban Challenge. Journal of Field Robotics. 2008. p. 616.

el punto de llegada deseado y traza la ruta más óptima teniendo en cuenta la distancia menor entre la ubicación del automóvil y el punto de llegada a la hora de efectuar el movimiento. Google plantea diversas soluciones a los problemas cotidianos de manejo tales como peatones, ciclistas, semáforos, e incluso otros vehículos<sup>17</sup>.

Mediante la reconstrucción y detección de objetos en el entorno, Google identifica los vehículos a su alrededor así como sus direcciones, además de esto identifica el carril navegable, es decir el sentido de cada carril y el área por el cual puede transitar en su propio carril como se observa en la (figura 8), teniendo en cuenta obras en la vía o carros estacionados, para lo cual encierra el volumen de cada patrón en rectángulos para el caso de los vehículos, para las señales de tránsito emplea la imagen de cada señal, para objetos desconocidos diferentes a los anteriores muestra un rectángulo amarillo y para los conos emplea triángulos de color naranja como se evidencia en la (figura 9)<sup>18</sup>.

**Figura 8.** Área navegable del vehículo Google



**Fuente.** Levy Karyne. *Here's What It Looks Like To Drive In One Of Google's Self-Driving Cars On City Streets [Video]*. Disponible en:<http://www.businessinsider.com/google-self-driving-cars-in-traffic-2014-4>

Como si fuera poco Google pensó en los acontecimientos imprevistos pero sumamente posibles a la hora de manejar como lo son los peatones y los ciclistas, para lo cual implementa un método similar al anteriormente explicado con un factor adicional, la computadora encierra el peatón o el ciclista de color rojo teniéndolo como prioridad a la hora de efectuar una decisión de evasión o algo similar para preservar la integridad del ser humano.

<sup>17</sup> FayerWayer. Google ha creado un automóvil que se maneja solo. 2014. [en línea] <<https://www.fayerwayer.com/2014/05/google-muestra-un-nuevo-auto-que-se-maneja-solo-disenado-completamente-por-ellos/>> Fecha de consulta 22/05/15.

<sup>18</sup> Ibid., [en línea]

**Figura 9.** Detección de obstáculos en la vía



**Fuente.** Levy Karyne. *Here's What It Looks Like To Drive In One Of Google's Self-Driving Cars On City Streets [Video]*. Disponible en: <http://www.businessinsider.com/google-self-driving-cars-in-traffic-2014-4>

Una vez detectado, analiza si el peatón o ciclista está obstruyendo su área navegable para lo cual se generan dos posibilidades, la primera es evitarlo y continuar con la ruta, esto se da cuando es una avenida de más de dos carriles y no hay ningún otro vehículo impidiendo esta acción, de lo contrario, de ser un solo carril y el área navegable ha sido invadida, disminuye su velocidad manteniendo una distancia prudente o deteniéndose finalmente tal como se muestra en la (figura 10)<sup>19</sup>.

**Figura 10.** Reconocimiento de peatón



**Fuente.** Levy Karyne. *Here's What It Looks Like To Drive In One Of Google's Self-Driving Cars On City Streets [Video]*. Disponible en: <http://www.businessinsider.com/google-self-driving-cars-in-traffic-2014-4>

<sup>19</sup> Ibid., [en línea]

Por otro lado Volvo ofrece una idea nueva de vehículos autónomos donde se optimizan las distancias recorridas, aspecto a mejorar de algunos prototipos anteriormente nombrados, esta propuesta consiste en generar una misma ruta para varios vehículos sin restricción de distancias y efectuando un control más simple, la idea de Volvo consiste en generar un tren de vehículos autónomos, es decir, que el tren de vehículos es dirigido por un camión el cual actúa como el centro de control y puede ser asistido por el ser humano, y detrás de él se encuentran una cadena de carros los cuales reciben información de la ruta usando tecnologías de mando maestro y esclavo.

Volvo desarrolló con éxito esta aplicación, la cual tuvo como prueba una ruta en las avenidas de España donde se vinculó al camión de mando tres autos de turismo con diversas características los cuales fueron un XC60, un V60 y un S60 como se observa en la (figura 11), durante la prueba se manejó una velocidad de 85 km/h y con una distancia entre los coches de 6 metros, aunque durante el ensayo se probaron otras distancias, entre 5 y 15 metros.

La idea de Volvo con esta implementación de tren de carros autónomos es reducir costos de producción y la optimización del tráfico en las grandes ciudades<sup>20</sup>.

**Figura 11.** Tren de vehículos autónomos de Volvo



**Fuente.** ABC. Volvo prueba en España su sistema de piloto automático para coches. 2012. URL: <http://www.abc.es/20120529/tecnologia/abci-coche-piloto-automatico-201205290811.html>

En la industria automotriz existen grandes marcas con el mismo objetivo de un carro sin conductor, Nissan plantea un modelo que además de manejarse automáticamente en las carreteras hará las funciones de parqueo automático, todo

<sup>20</sup> ABC. Volvo prueba en España su sistema de piloto automático para coches. 2012. [en línea] < <http://www.abc.es/20120529/tecnologia/abci-coche-piloto-automatico-201205290811.html> > Fecha de consulta 20/05/15

esto aplicado a una idea revolucionaria para la industria automotriz y para el medio ambiente, el prototipo será aplicado a un carro eléctrico como se muestra en la (figura 12), lo cual disminuirá radicalmente las emisiones de  $CO_2$  causadas por la combustión de gasolina, diesel o gas. Sin embargo, este modelo solo será lanzado hasta el 2020 con el fin de perfeccionar todas las tecnologías y garantizar la seguridad a sus clientes<sup>21</sup>.

**Figura 12.** Vehículo eléctrico autónomo Nissan



**Fuente.** HÍBRIDOS Y ELÉCTRICOS, Ecotecnología del vehículo. El coche eléctrico autónomo de Nissan, en 2016. URL: <http://www.hibridosyelectricos.com/articulo/tecnologia/coche-electrico-autonomo-nissan-2016/20140719000725007531.html>

Seguidamente, se encuentra que uno de los primeros en incursionar en los carros híbridos fue Tesla, el cual generó en 2014 una modalidad de piloto automático el cual se activa desde la consola de mando permitiéndole al conductor descansar por un momento en los trayectos largos y monótonos, el piloto automático no se guía por GPS como los prototipos antes mencionados, simplemente cuenta con una serie de sensores ópticos e infrarrojos para la detección de las líneas del camino y la distancia de los automóviles en la vía, esto respaldado por una serie de cámaras que cubren un ángulo de visión de trescientos sesenta grados<sup>22</sup>; el vehículo se puede observar en la (figura 13).

---

<sup>21</sup> HÍBRIDOS Y ELÉCTRICOS, Ecotecnología del vehículo. El coche eléctrico autónomo de Nissan, en 2016. 2014. [en línea] < <http://www.hibridosyelectricos.com/articulo/tecnologia/coche-electrico-autonomo-nissan-2016/20140719000725007531.html> > Fecha de consulta 22/05/15

<sup>22</sup> DIARIOMOTOR. Tesla quiere y puede lanzar un coche autónomo, pero no les dejan, y se conforman con un Model S que cambia solo de carril. 2014. [en línea] < <http://www.diariomotor.com/2014/10/10/tesla-autopilot-semi-autonomo/> >

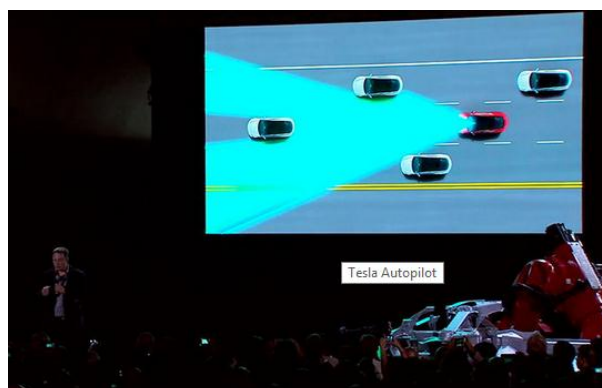
**Figura 13.** Vehículo con piloto automático Tesla



**Fuente.** DIARIOMOTOR. Tesla quiere y puede lanzar un coche autónomo, pero no les dejan, y se conforman con un Model S que cambia solo de carril. 2014. URL: <http://www.diariomotor.com/2014/10/10/tesla-autopilot-semi-autonomo/>

Las cámaras entregan datos en tiempo real de vehículos y además hace la reconstrucción del entorno en el que se desenvuelve de tal manera que el usuario puede ver en la pantalla de comando el escenario por el cual está transitando, las cámaras y los sensores hacen una interacción para evitar chocar con los demás vehículos como se ilustra en la (figura 14)<sup>23</sup>.

**Figura 14.** Detección de vehículos tesla



**Fuente.** DIARIOMOTOR. Tesla quiere y puede lanzar un coche autónomo, pero no les dejan, y se conforman con un Model S que cambia solo de carril. 2014. URL: <http://www.diariomotor.com/2014/10/10/tesla-autopilot-semi-autonomo/>

El coche Shelley es un Audi TTS (figura 15), que Volkswagen comenzó a desarrollar desde el 2010 para carreras, se desarrolló con la colaboración de Stanford Racing Team de la Universidad de Stanford<sup>24</sup>.

<sup>23</sup> Ibid., [en línea]

<sup>24</sup> MICROSIERVOS. Un coche autónomo gana en circuito a un piloto amateur. 2015. [en línea] <<http://www.microsiervos.com/archivo/tecnologia/coche-autonomo-gana-circuito-a-piloto-amateur.html>>  
Fecha de consulta 26/05/15

Obteniendo como resultado un auto capaz de ir a doscientos kilómetros por hora sin conductor alguno probado en la ascensión al Pikes Peak, donde batió el récord del piloto oficial de Audi por 0.4 segundos, resultados impresionantes para un vehículo autónomo teniendo en cuenta la velocidad del auto y de igual manera la velocidad necesaria de procesamiento para ejercer el control<sup>25</sup>.

**Figura 15.** Vehículo autónomo Audi



**Fuente:** Microsiervos. *Un coche autónomo gana en circuito a un piloto amateur.* 2015. URL: <http://www.microsiervos.com/archivo/tecnologia/coche-autonomo-gana-circuito-a-piloto-amateur.html>

La idea de la automatización de los carros cada vez se hace más fuerte, hasta el punto en que las grandes compañías no solo ven la automatización como un método para salvar vidas o avanzar en conocimientos e investigación, sino como un lujo a la hora de crear sus nuevas propuestas para ser lanzadas al mercado. Un buen ejemplo de lo anterior es BMW®, quien lanzó al mercado en la exposición de vehículos en las vegas el nuevo F015 Luxury, el cual no es solo un buen vehículo ya que además de esto se conduce solo, integrando las tecnologías antes nombradas como el reconocimiento del área navegable y la modalidad de adelantar en una vía recta.

El nuevo carro también tiene la posibilidad de ser conducido manualmente, lo cual es un factor importante ya que para algunas personas el tener un carro autónomo le quita la emoción al conducir, adicionando a los sistemas autónomos tiene la opción de parqueo automático en paralelo o en reversa y como si fuera poco cuando está estacionado el conductor mediante su control de bloqueo manual puede llamar a su auto desde cualquier parte del parqueadero y el vehículo llega hasta el frente de él y abre sus puertas para ser abordado<sup>26</sup>.

---

<sup>25</sup> Ibid., [en línea]

<sup>26</sup> LR. La república. Mercedes, BMW y Toyota presentan los carros autónomos del CES. 2015. [en línea] < [http://www.larepublica.co/mercedes-bmw-y-toyota-presentan-los-carros-aut%C3%B3nomos-del-ces\\_207501](http://www.larepublica.co/mercedes-bmw-y-toyota-presentan-los-carros-aut%C3%B3nomos-del-ces_207501) >



Las alianzas comerciales no se han hecho esperar así que Toyota® ofreció a la compañía Google® abastecerla de los autos necesarios para sus investigaciones en cuanto a vehículos autónomos, a cambio de un sistema de navegación para sus carros, lo cual tuvo como resultado el Lexus LS 600h el cual cuenta con una serie de cámaras de alta gama las cuales hacen una reconstrucción de obstáculos en el camino brindando datos de distancia entre objetivos y evaluar las posibilidades de adelantar, esta opción está disponible para la conducción manual, pero si el conductor lo desea puede poner estas mismas funciones en conducción automática, en este caso el vehículo ejecuta las decisiones de adelantar por sí solo<sup>27</sup>.

### 1.3.2 Vehículos autónomos en el sector militar

La industria militar ofrece una gran cantidad de desarrollos, otra de las grandes aplicaciones de vehículos autónomos es el GUARDIUM UGV como se observa en la (figura 16), usado para sustituir a los centinelas en terrenos hostiles. El guardium es un vehículo usado en muchos terrenos de combate como el desierto o en terrenos de trópico, al igual en las prisiones de alta seguridad el guardium ofrece una gran ayuda ya que las cámaras son enlazadas a un centro de control y ya que estas tienen detección de movimiento y de objetivos ejerce una gran labor de centinela<sup>28</sup>.

**Figura 16.** *Guardium UGV*



**Fuente.** *'Guardium' unmanned ground vehicle patrols Gaza border.* URL: [http://www.israelnationalnews.com/News/News.aspx/162982#.Vf2NLd9\\_Okp](http://www.israelnationalnews.com/News/News.aspx/162982#.Vf2NLd9_Okp)

---

<sup>27</sup> Hipertextual. Toyota y Audi mostraran sus vehículos autónomos en el CES. 2013. [en línea] <<http://hipertextual.com/2013/01/toyota-audi-vehiculos-autonomos-ces> >

<sup>28</sup> Dvaruh. Guardium Unmanned Ground Vehicle (UGV). 2014. [en línea] <<http://techmafia.net/robotics/2014/06/13/guardium-unmanned-ground-vehicle-ugv.html> > Fecha de consulta 26/05/15

Los algoritmos de programación de filtros de imagen de este vehículo militar están diseñados para tareas específicas útiles en el campo de combate, la primera aplicación es la detección de una persona o soldado, ya sea aliado o enemigo, el soldado a cargo de monitorear las cámaras, decide de acuerdo a los historiales de misiones si es posiblemente un aliado herido o un enemigo camuflado como se evidencia en la (figura 17)<sup>29</sup>.

**Figura 17.** Detección de soldado



**Fuente.** 'Guardium' unmanned ground vehicle patrols Gaza border. URL: [http://www.israelnationalnews.com/News/News.aspx/162982#.Vf2NLd9\\_Okp](http://www.israelnationalnews.com/News/News.aspx/162982#.Vf2NLd9_Okp)

La segunda aplicación de estas cámaras es la detección de huellas en el camino, esto genera la gran ventaja de rastrear al enemigo o encontrar tropas extraviadas en escenarios difíciles como el desierto, como se ilustra en la (figura 18)<sup>30</sup>.

**Figura 18.** Detección de patrones de huella



**Fuente.** 'Guardium' unmanned ground vehicle patrols Gaza border. URL: [http://www.israelnationalnews.com/News/News.aspx/162982#.Vf2NLd9\\_Okp](http://www.israelnationalnews.com/News/News.aspx/162982#.Vf2NLd9_Okp)

<sup>29</sup> Woman Power Fights Hamas with 'Unmanned Ground Vehicles'. 2012. [en línea] <[http://www.israelnationalnews.com/News/News.aspx/162982#.Vf3wqt9\\_Okq](http://www.israelnationalnews.com/News/News.aspx/162982#.Vf3wqt9_Okq) >

<sup>30</sup> Ibid., [en línea]

La guerra de Irak trajo consigo una gran variedad de modalidades de combate o de camuflaje, los soldados iraquíes usaron la excavación de túneles y aprovechamiento de cuevas para resguardarse de los ataques aéreos y llegar sin ser detectados a diversos puntos, debido a ello se genera la tercera aplicación del vehículo la cual es la detección de cuevas, labor que puede ser monitoreada por un soldado para la ejecución de decisiones tácticas como se observa en la (figura 19)<sup>31</sup>.

**Figura 19.** Detección de túneles



**Fuente:** Israeli Robotic hour Rises in operation?. URL: <https://xi0.info/israeli-robotic-hour-rises-operation.html>

#### 1.4 OBJETIVOS

Desarrollar un prototipo de un vehículo autónomo terrestre, capaz de realizar la planeación y seguimiento de trayectorias en un entorno determinado.

Para llegar al objetivo general se plantean los siguientes objetivos específicos:

- Determinar y validar el modelo cinemático del vehículo el cual considere las restricciones físicas del mismo.
- Implementar un algoritmo para el seguimiento de trayectorias en un ambiente controlado.
- Desarrollar una aplicación de visión computacional para transmitir los datos de posición y orientación del vehículo en tiempo real.

---

<sup>31</sup> Israeli Robotic hour Rises in operation? 2008. [en línea] < <https://xi0.info/israeli-robotic-hour-rises-operation.html> >

- Realizar el análisis de los dispositivos electrónicos embarcados en el vehículo.
- Implementar la electrónica necesaria para procesar, ejecutar y detectar los parámetros que se encuentran en el ambiente del vehículo, teniendo en cuenta los obstáculos presentes cuando el vehículo está ejecutando sus desplazamientos.
- Validar el sistema propuesto en un ambiente recreado.

### **1.5 ALCANCES Y LIMITACIONES**

El vehículo está compuesto por un sistema de procesamiento embebido (Pcduino/Arduino), dispositivos eléctricos, cámaras, sensores y un computador para la visualización de las tareas a ejecutar. La sinergia de estos componentes permiten la implementación de algoritmos basados en el control de dirección y desarrollos de visión computacional, con el objeto de realizar seguimiento de trayectorias (previamente estipuladas) considerándolas libre de obstáculos.

De esta manera se logra que el vehículo se desplace de forma autónoma hacia el camino de puntos predefinidos realizando el trayecto deseado. Por otro lado, en una segunda aplicación el vehículo es capaz de reconocer peatones y demarcar las líneas presentes en la vía. El usuario puede observar la ejecución del movimiento, procesamiento digital de imagen e información del sistema en tiempo real mediante la implementación de una interfaz gráfica.

Ya que se desarrolló un proyecto funcional a bajo costo dado a factores económicos, las limitaciones del proyecto en primera instancia radican en las dimensiones del vehículo (escala 1:10) y la calidad de los dispositivos implementados, de igual forma el procesamiento digital de imágenes elaborado es dependiente de la iluminación del ambiente lo cual hace que se tengan ciertas restricciones para el desarrollo de las aplicaciones. Debido a que el desarrollo de los algoritmos de visión computacional, requieren de gran capacidad de procesamiento, la velocidad de transmisión y recepción de datos es un limitante en los resultados obtenidos.

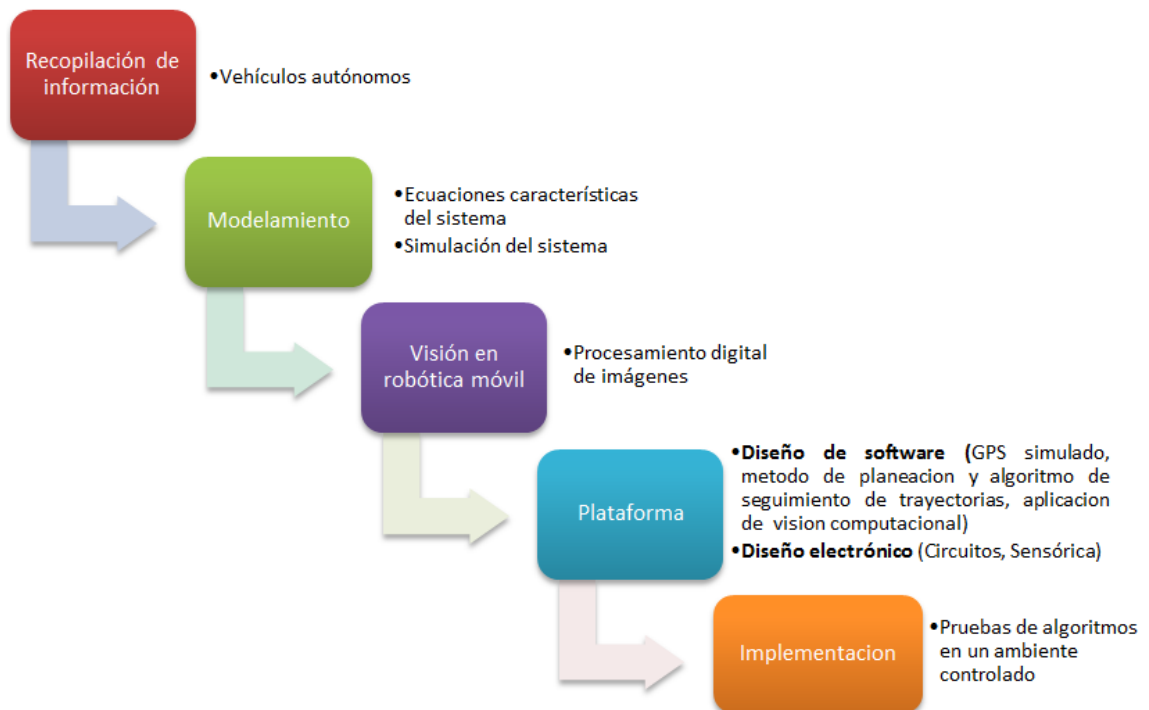
### **1.6 LÍNEA DE INVESTIGACIÓN DEL PROGRAMA**

El proyecto se enfoca en la línea de investigación del programa de robótica y biomecatrónica, ya que representa los sistemas móviles autónomos y a su vez este proyecto de opción de grado está vinculado a la investigación de vehículos autónomos propuesta y desarrollada por el profesor Rubén Dario Hernández dentro del semillero de investigación.

## 1.7 DISEÑO METODOLÓGICO

La metodología implementada en el proyecto se realizó mediante la aplicación de una serie de procesos, los cuales se ilustran en la (figura 20), los cuales permitieron el desarrollo de los objetivos propuestos. De igual forma las condiciones del área de trabajo, el intercambio de conocimientos y participación activa, permitiendo dar solución a los inconvenientes presentados durante el desarrollo del proyecto.

*Figura 20. Metodología Implementada*



*Fuente: Elaboración propia*

## 1.8 CONTRIBUCIONES

Dentro del desarrollo del presente trabajo, se logró durante el desarrollo del proyecto publicar un artículo, el cual se expone en el *anexo A*.

- Kevin S. Sarmiento G., Oscar L. Laguna S., Javier E. Vásquez L., Ruben D. Hernández B. Seguimiento de Trayectorias para un vehículo Autónomo. En: **IV Congreso Internacional de Ingeniería Mecatrónica y Automatización, Medellín 2015.**

## 2 FUNDAMENTOS TEÓRICOS

El presente capítulo presenta los fundamentos teóricos para el conocimiento y desarrollo de vehículos autónomos, desarrollando la contextualización sobre la localización por GPS, percepción sensorial y visual, control y planeamiento de rutas. El conocimiento de estos conceptos permitirá la aplicación y comprensión de las diversas prácticas que exige la robótica móvil enfocada al planeamiento y ejecución de trayectorias.

### 2.1 DESCRIPCIÓN DE LA IMPLEMENTACIÓN

El desarrollo de la robótica ha traído consigo avances significativos en la aplicación de áreas como: control, programación, inteligencia artificial, percepción e instrumentación que sirven de base para el avance en diferentes campos de la industria. En el desarrollo del proceso de estructuración y ejecución del robot es necesario el entendimiento de estas áreas, de igual manera conocer los principios de la navegación autónoma y los componentes necesarios del prototipo a implementar<sup>32</sup>.

El principio de la navegación autónoma es descrito según Cárdenas<sup>33</sup>, como el desplazamiento entre dos lugares sin depender de la intervención humana. La navegación tiene una serie de componentes los cuales se dividen en tres partes importantes, tal como se describe en el diagrama de la (figura 21)<sup>34</sup>.

La primera parte está compuesta por la *percepción*, en ella se recopila toda la información del ambiente físico captada por los sensores y dispositivos de visión computacional, a partir de ello se realizan los respectivos acondicionamientos de señal e interpretación de las variables entregadas.

La segunda parte está integrada por la *localización*, en la que intervienen métodos que se pueden desarrollar mediante GPS o procesamiento de imagen, además de esto en la planeación de una ruta se debe tener en cuenta la cinemática del vehículo con el fin de que la trayectoria final sea posible de ejecutar por el mismo.

En la etapa de control de movimiento se tiene en cuenta la percepción sensorial y visual además de la ruta deseada, las variables del entorno son entregadas por los etapa de percepción y la trayectoria para ejecutar el movimiento, este proceso es

---

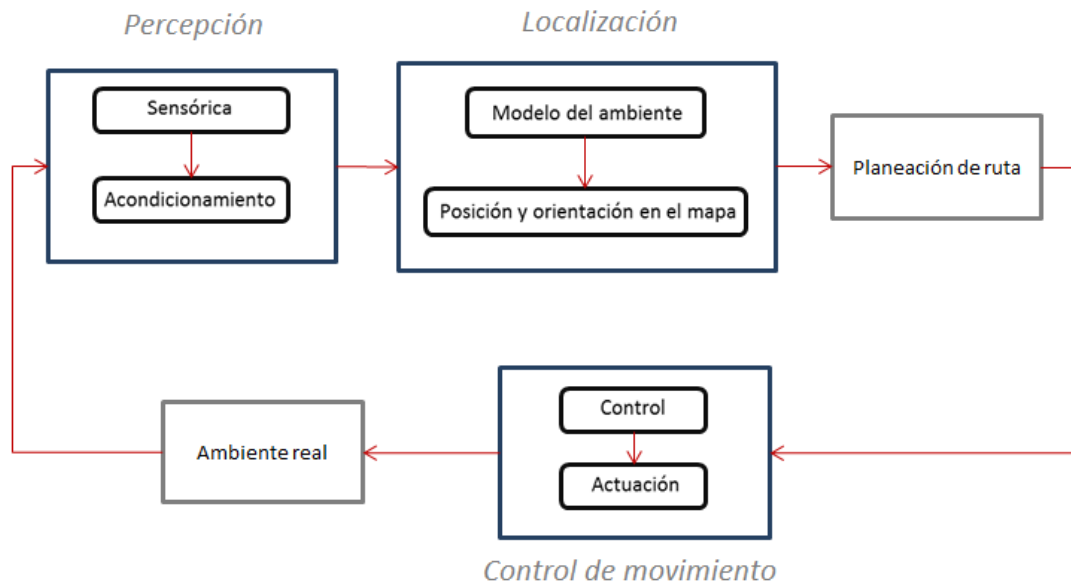
<sup>32</sup> Bermúdez, Giovanni. Robots móviles: teoría aplicaciones y experiencias. 2002. p.10.

<sup>33</sup> Cárdenas Rueda, Miguel Ángel. Localização Híbrida para Um Veículo Autônomo em Escala Usando Fusão de Sensores. Campinas, 2012.

<sup>34</sup> FernandezLancha, Miguel. Fernandez Sanz, David. Valsamedia Plasencia, Carlos. Planificación de trayectorias para un robot móvil. 2009-2010. p. 21.

ejecutado en tiempo real con el fin de realizar acciones correctivas para el seguimiento de la ruta.

**Figura 21.** Proceso general de la navegación autónoma.



**Fuente.** Elaboración propia

## 2.2 SISTEMA DE LOCALIZACIÓN (GPS)

El sistema de localización GPS o (Global Position System) es usado para posicionar un vehículo o un objeto en un determinado espacio, este sistema integra tecnologías satelitales que en conjunto da como resultado que determinado dispositivo sea localizado.

El GPS integra la cartografía, arte de crear mapas, lo cual permite visualizar en un entorno conocido la ubicación brindada por el sistema de posicionamiento global, ya que la cartografía ofrece diversas variables como lo son mapas de elevación, ríos, avenidas de los centros urbanos, topografía corrientes aéreas y marítimas; también existen diversos GPS que adaptan o no estos mapas como GPS militares que integran de acuerdo a su necesidad mapas topográficos, navales o aéreos, vehiculares que integran en su mayoría mapas urbanos dando como resultado diversas aplicaciones.

El sistema de navegación global satelital funciona por tres factores importantes: la cartografía según la aplicación, al menos cuatro satélites de posicionamiento y un

receptor de la señal emitida por los satélites en el vehículo, de ser el caso de una aplicación de GPS urbano. En la actualidad existen 24 satélites geoestacionarios ubicados aproximadamente a 20200 km de distancia de la tierra, estos satélites están distribuidos en seis planos orbitales generando una cobertura global del sistema; los datos finales de posición son generados por la interacción de los satélites y al receptor son entregados en medidas de latitud, longitud, orientación, altitud y tiempo dando la posibilidad de ubicarlos en los mapas antes mencionados<sup>35</sup>.

La manera de saber la posición del vehículo es mediante el método de triangulación el cual consiste en emitir varias señales desde puntos distintos para compararla y obtener la ubicación del receptor; cada satélite emite una señal a la cual se le conoce la frecuencia ( $f$ ) y la longitud de onda ( $\lambda$ ) lo cual da la velocidad ( $v$ ) como se presenta en la ecuación (1).

$$v = f * \lambda \quad (1)$$

Con el fin de conocer cuál señal es emitida por los diferentes satélites cada uno de ellos maneja una frecuencia distinta. Con la comunicación entre emisor y receptor se conoce el tiempo ( $t$ ) de llegada de la señal, con estos dos valores se halla la distancia ( $D$ ) entre el satélite y el receptor como se ilustra en la ecuación (2).

$$D = v * t \quad (2)$$

Conociendo esta distancia ( $a$ ) y la relación de la tierra al satélite ( $h$ ) es fácil hallar mediante el teorema de Pitágoras el radio ( $r$ ) de la circunferencia como se presenta en la ecuación (3) en este radio se encuentra el receptor como se ven en la (figura 22).

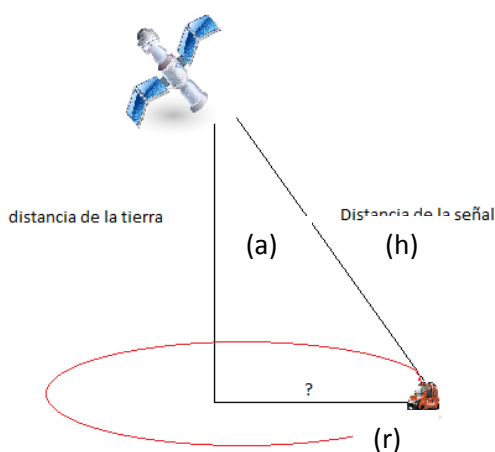
$$r^2 = h^2 - a^2 \quad (3)$$

---

<sup>35</sup> Informática hoy. Que es el GPS y como funciona. [en línea] < <http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-el-GPS-y-como-funciona.php> > Fecha de consulta 25/09/2015.



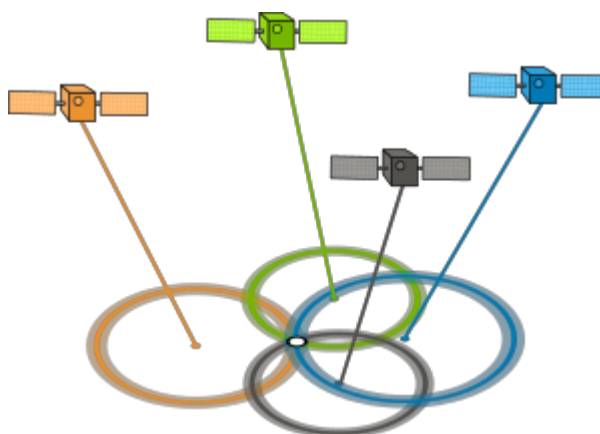
**Figura 22.** Distancia del radio de señal



**Fuente.** Elaboración propia

Teniendo este dato por cada satélite se hace una comparación entre las señales de recepción las cuales describen una circunferencia creada por el radio antes calculado, la posición se determina por el único punto de intersección de los radios de los cuatro o más satélites utilizados como se ilustra en la (figura 23).

**Figura 23.** Intersección de radios de cobertura



**Fuente.** 'paesta' how GPS revolutionized seismic research.  
URL <http://www.paesta.psu.edu/article/how-gps-revolutionized-seismic-research>

Finalmente se obtiene la posición en datos de longitud, latitud y orientación los cuales se ubican en el mapa, este proceso se ejecuta en tiempo real para visualizar el desplazamiento del receptor en el mapa dispuesto para ello, como se observa en la (figura 24).

**Figura 24. Navegación por GPS**

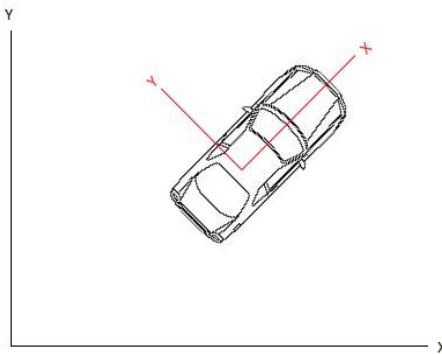


**Fuente.** Sygic GPS 13.4.2. <http://www.taringa.net/comunidades/advance/8510996/Aporte-Sygic-GPS-13-4-2-Full-Maps-Downloader-MEGA.html>

En algunas aplicaciones de localización en interiores se hace necesario implementar un sistema de localización ya que para el GPS global sería imposible brindar datos precisos de la ubicación del robot, teniendo en cuenta que se realizarán movimientos en interiores, además de lo anterior el GPS global carece de precisión en lugares con edificaciones altas y continuas ya que posee un error de aproximadamente 15 metros, lo cual se traduce en un error de posición.

Un sistema de localización parte del mismo principio que un GPS global pero aplicado de manera diferente, para esto se debe tener en cuenta el uso de una cámara que visualiza la totalidad del espacio de trabajo del vehículo, un robot móvil parametrizado para ser localizado por la cámara (colores o formas definidas). Para una buena simulación no solo es necesario saber en qué punto de la imagen está el vehículo sino también la orientación del robot respecto a un eje coordenado de referencia, para lograrlo primero se le da un eje coordenado al vehículo y otro al escenario donde efectúa los movimientos como se muestra en la (figura 25).

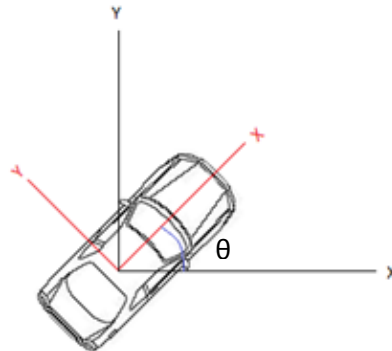
**Figura 25. Ejes de referencia**



**Fuente.** Elaboración propia

Para conocer el ángulo ( $\theta$ ) de orientación del vehículo se traslada el eje coordenado del mismo al punto (0.0) del eje de referencia, como se muestra en la (figura 26).

**Figura 26.** Ángulo de orientación

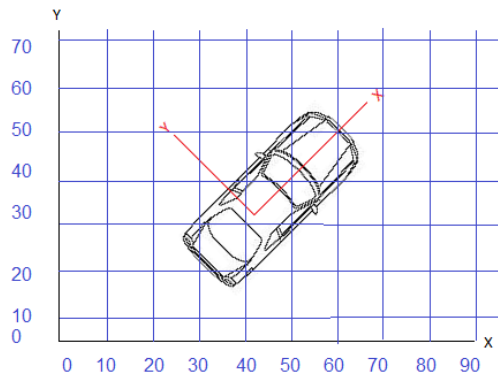


**Fuente.** Elaboración propia

Para la localización en el escenario se efectúa una parametrización de la imagen generando un plano coordenado dentro de la misma, las medidas de las imágenes están dadas en píxeles, como por ejemplo (640 X 480) el cual es un tamaño de imagen que maneja una matriz de 640 píxeles de ancho por 480 de largo, conociendo las medidas del área de trabajo en metros se pueden convertir los píxeles en unidades de distancia mediante el método de interpolación que se describe en la ecuación (4), esto caracterizará el área de trabajo como se muestra en la (figura 27).

$$y = y_0 + (x - x_0) \frac{(y_b - y_a)}{(x_b - x_a)} \quad (4)$$

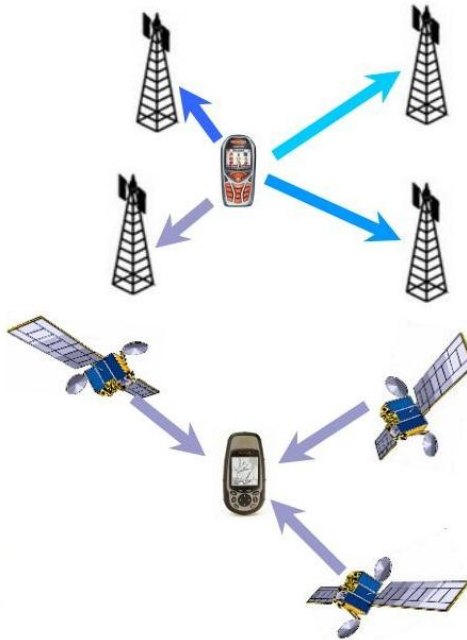
**Figura 27.** Parametrización de imagen



**Fuente.** Elaboración propia

Dentro de los LPS (local position system) se encuentran gran variedad de métodos los cuales se implementan en interiores, dentro de ellos se encuentran señales de radio, WiFi o Bluetooth. Estas señales son implementadas mediante la interacción del emisor y las antenas de recepción las cuales manejan un protocolo a la inversa del utilizado por el GPS tal como se ve en la (figura 28).

**Figura 28.** Configuración de comunicación



**Fuente.** Instituto de automática industrial. URL <http://www.car.upm-csic.es/lopsi/static/publicaciones/docencia/Apuntes%20RF-LPS.pdf>

El posicionamiento por señales RF se maneja mediante la configuración ilustrada en la (figura 27) dicha señal es interpretada por el receptor y comparada con los demás receptores para descifrar la posición del emisor; la proximidad del emisor a cada receptor es dada por la intensidad de señal que capta cada uno de ellos; la señal RF es óptima para interiores ya que ignora los materiales de construcción presentes en el entorno y posee poca interferencia de otras señales RF de baja intensidad.

El protocolo GSM hace referencia al posicionamiento mediante señales de telefonía móvil la cual actúa mediante la medición de los tiempos de respuesta entre el emisor (móvil) y el receptor (antena de telefonía), esta medición en conjunto con varios receptores brinda el posicionamiento del móvil; la

implementación en interiores reduce el error de posicionamiento que posee este método en exteriores el cual es de 50 a 200m, para interiores este error se ve reducido a 2,5m.

Las señales WiFi, Bluetooth y Wmax actúan bajo la misma configuración de localización emisor receptor, este tipo de señales posee gran velocidad y rango de cobertura, el bajo coste de los elementos hace que este método sea implementado en gran cantidad de aplicaciones que no requieran tanta precisión ya que posee un error de 1,5m de localización.

### **2.3 SISTEMA DE PERCEPCIÓN**

El sistema de percepción es la etapa de mayor importancia en el desarrollo de cualquier tipo de implementación robótica, dado que las informaciones obtenidas permitirán al robot o al controlador comprender y tomar decisiones acerca del entorno en que se encuentre.

Dentro de los sistemas de percepción se encuentran dos tipos: El primer sistema es conocido como percepción remota, esta clase de percepción hace alusión a un centro de mando donde llega toda la información y un operario o usuario que se encarga de tomar las decisiones de acuerdo a las variables observadas; El segundo sistema es basado en la intelección la cual es la integración de dispositivos electrónicos que proporcionen la mayor información posible acerca del entorno en que se encuentre para que un cerebro electrónico la pueda interpretar y genere las decisiones a partir de los datos entregados<sup>36 37</sup>.

De igual forma, los sistemas de percepción tienen dos grandes variables las cuales están basadas en los sistemas sensorial y visual. Estas variables funcionan de manera distinta pero con el mismo propósito, responder la pregunta ¿qué me rodea?, en aplicaciones de robots autónomos y sobre todo en robots móviles el sistema de percepción responde la pregunta antes formulada a cada instante de la ejecución de movimiento con el fin de minimizar errores en la ejecución de este, además de esto brindara información esencial que se debe tener en cuenta según el propósito del robot móvil, como por ejemplo el planeamiento de trayectorias.

#### **2.3.1 Percepción sensorial**

La percepción sensorial permite medir variables físicas del escenario en el cual se encuentra el robot, por ejemplo la temperatura, la velocidad, la distancia entre

---

<sup>36</sup> Semar. Percepción remota. [en línea] <<http://www.semar.gob.mx/s/informacion-sector/ciencia/ermexs/percepcion-remota.html> > Fecha de consulta 25/09/2015.

<sup>37</sup> Julio Martínez Muñoz. Percepción remota "Fundamentos de la teledetección espacial" 2005. p. 9.

objetos, entre otras variables cuantificables con el uso de sensores como se muestra en la (tabla 1).

**Tabla 1. Sensores**

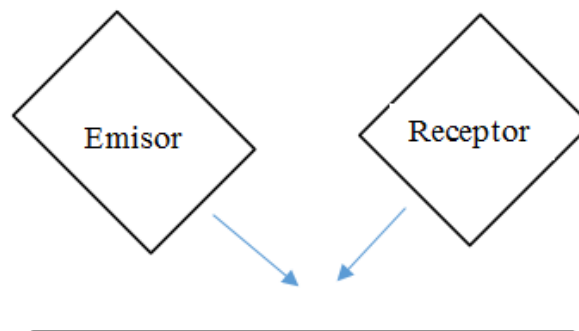
Magnitud	Descripción	Transductor
Posición lineal y angular	Se utilizan para medir la posición de un objeto previamente paramentado	Potenciómetro
Desplazamiento y deformación	El desplazamiento o deformación es medido por la perturbación del sensor calibrado previamente	Transformador diferencial de variación lineal
Velocidad lineal y angular	Mediante la perturbación o la medición de parámetros se ejecuta la medición de las velocidades respectivas	Dinamo tacométrica
Aceleración	Se genera la perturbación del sensor la cual es medida en magnitud de aceleración	Acelerómetro
Presión	La presión ejerce una fuerza deformante la cual es medida por el sensor	Membranas
Temperatura	El sensor posee un material termoeléctrico el cual mide la variación de temperatura	Termopar
Visión artificial	La cámara de vídeo es paramentada para captar variables específicas del entorno	Cámaras de video
Sensor acústico (presión sonora)	Capta las variaciones sonoras del entorno	micrófono
Sensores de acidez	Mide el PH del entorno en el que se encuentra	ISFET
Sensor de luz	Mide la incapacidad de rayos UV en el entorno	fotodiodo
		Fotorresistencia
		Fototransistor

**Fuente.** Elaboración propia

Dentro de la amplia gama de sensores existen dos clasificaciones importantes las cuales son: internos y externos. Los sensores internos son aquellos que miden variables físicas del vehículo o prototipo móvil, variables físicas como: velocidad, posición, aceleración, estas magnitudes son medidas mediante encoders, acelerómetros o giroscopios que nos entregarán datos del comportamiento que tiene el robot para las diversas variables nombradas. Los sensores externos miden variables del exterior ajenas al robot como por ejemplo temperatura del ambiente, humedad, distancia de objetos o luminosidad<sup>38</sup>.

El sensor interno más usado es el encoder; los encoders funcionan por la emisión y recepción de rayos infrarrojos, mediante la interrupción de estos rayos o la recepción de señal infrarroja con cierta frecuencia se puede determinar la velocidad. La señal infrarroja es emitida por un led infrarrojo, esta señal se refleja en una superficie y es capturada por el receptor como se muestra en la (figura 29).

**Figura 29.** Señal infrarroja



**Fuente.** Elaboración propia

El emisor y el receptor poseen un ángulo determinado por el fabricante de acuerdo a especificaciones de distancia de lectura o sensibilidad del sensor, la refracción de la señal infrarroja se controla colocando una cinta de color generalmente blanco y negro, ya que el color blanco refleja la totalidad de la luz mientras el color negro la absorbe, creando de esta manera el corte o saturación del transistor infrarrojo, cuando la luz es reflejada el receptor permite el flujo de corriente y dependiendo de la configuración del transistor se genera un uno o un cero lógico en la salida del transistor creando el pulso alto o bajo de la señal.

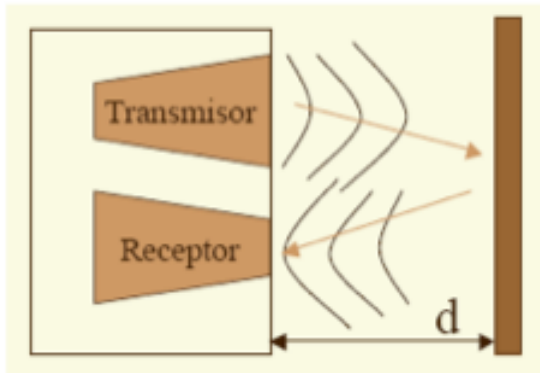
---

<sup>38</sup> Torres Medina, Fernando. Percepción en Robótica, 2004. Universidad de Alicante. "La robótica en los albores del siglo XXI"

De igual manera los sensores externos más utilizados son los ultrasónicos ya que permiten determinar la distancia de un objeto u obstáculo, este sensor envía una señal de ultrasonido la cual rebota en el objeto dando como resultado la distancia del mismo, como se puede observar en la (figura 30), esta distancia se calcula por la velocidad de propagación del sonido y el tiempo de retorno de esta, con estas dos variables (velocidad del sonido en el aire y tiempo de recepción del eco de la señal) es fácilmente calculada la distancia mediante la ecuación (5).

$$D = \frac{1}{2} v * t \quad (5)$$

**Figura 30.** Señal del sensor ultrasónico



**Fuente.** Limoncello digital. URL: <http://www.limoncellodigital.com/2012/03/que-son-los-sensores-ultrasonidos-y.html>

### 2.3.2 Percepción visual

En el campo de la robótica móvil la percepción visual juega un papel muy importante en el reconocimiento del entorno, ya que con esta es posible determinar valores que los sensores no brindan totalmente. La percepción visual conlleva el tratamiento de imágenes mediante cámaras. Las cámaras de acuerdo a su cantidad y algoritmos programados pueden desempeñar diversas funciones como visión externa, visión interna o incluso reconstrucción del entorno.

La visión interna es la configuración de una o más cámaras embarcadas en el robot móvil, la configuración de visión interna está enfocada a percibir datos del entorno en el cual se está moviendo el robot, datos tales como reconocimiento de objetos, profundidades, líneas, entre otros como se ven en la (figura 31).



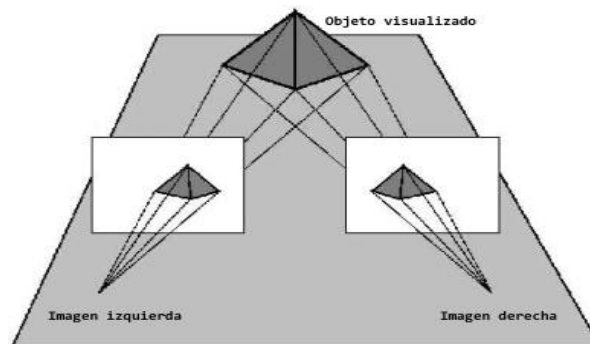
**Figura 31. Visión interna**



**Fuente.** Los autos sin conductor ya son un hecho en nevada URL: <http://www.robotnik.es/robots-moviles/summit-xl/>

Mediante la visión estereoscópica es posible reconstruir el entorno ya que al poder tener la comparación de dos cámaras enfocadas en el mismo punto se pueden determinar con mayor exactitud profundidades u obstáculos como se observa en la (figura 32).

**Figura 32. Visión estereoscópica**

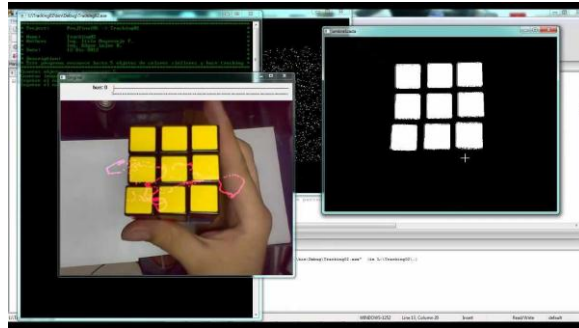


**Fuente.** La tecnología 3D. URL: <http://blog.panasonic.es/tecnologia/tecnologia-3d-a-fondo/>

Las configuraciones antes nombradas harán uso de algunas o todas las herramientas que ofrecen los filtros de imagen, los filtros de imagen son de gran utilidad para parametrizar variables del entorno real como lo son los filtros de color, líneas, referencias de imagen. El uso de dichos filtros es libre para cualquier tipo de aplicación que se desee ejecutar. Los filtros de color son de gran ayuda cuando se quiere parametrizar una cosa en específico como un objeto o incluso un robot, en el ambiente real hay colores que siempre están fijos, en los cuales los filtros de color son también de gran ayuda como el caso de semáforos o señales de tránsito. El filtro de color funciona mediante el reconocimiento de la frecuencia de cada color ya que para el ser humano como para las cámaras el

reconocimiento de los colores se da gracias a la frecuencia de la luz que refleja cada objeto, con esto se pueden aislar diversas frecuencias o colores como se ilustra en la (figura 33).

**Figura 33.** Detección de colores



**Fuente.** Seguimiento y detección de objeto por color y forma con OpenCV [video]. Disponible en: <https://www.youtube.com/watch?v=HBwLIf-6UDQ>

La detección de líneas o bordes se efectúa mediante el cambio brusco de colores o método de degradación de color que representa la variación de intensidad del mismo, por ejemplo si hay un cambio brusco en la intensidad de color será detectado como borde de lo contrario no será detectado, en ocasiones se presentan bordes falsos con el método de degradación por lo cual se complementa con la detección laplaciana la cual se usa para encontrar bordes muy finos ya que cada ondulación es representada como un cero y los bordes con unos como se puede observar en la (figura 34).

**Figura 34.** Detección de bordes

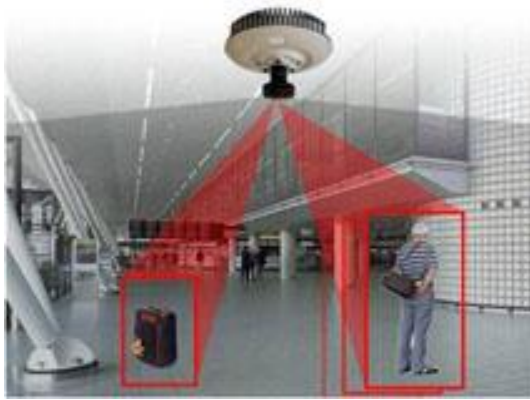


**Fuente.** Área docente. URL: <http://areadocente.blogspot.com.co/2012/10/deteccion-de-bordes-en-oftalmologia.html>

La técnica de referencia de imagen se usa en casos muy puntuales donde solo se desea detectar un objeto o persona en específico, el proceso consiste en guardar

diversas imágenes del mismo objeto y compararlas constantemente, el porcentaje de reconocimiento es graduable para efectos de mayor o menor precisión tal como se muestra en la (figura 35).

**Figura 35.** Detección de objetos



**Fuente.** Sosedí. URL: <http://www.sosedí.com.mx/servicios/video-analitico.php>

## 2.4 SISTEMA DE CONTROL

El control de un robot es la parte más importante de todo el proceso ya que en este se tienen en cuenta las variables obtenidas por los diferentes métodos de percepción, el control ejecutará acciones de acuerdo a los valores entregados de los sensores ya sean electrónicos o cámaras en el caso de la percepción visual. Los métodos de control más usados son PID y el cumplimiento de parámetros<sup>39</sup>.

El control PID (proporcional integral derivativo) hace alusión la captura de un dato y la generación de una respuesta de estabilización de acuerdo a estos datos; los factores fundamentales de un control PID son una referencia la cual dirá el valor deseado al cual se desea llegar, una referencia actual la cual no es más que el estado en tiempo real del dispositivo controlado y por último la señal de salida que es el resultado de la comparación entre estas dos variables con las correcciones necesarias para lograr el objetivo de estabilización<sup>40</sup>.

El control PID permite ejecutar acciones de acuerdo a las ganancias de cada parte de este (proporcional, integral, derivativo) dichas acciones corresponden a atenuación de señal, tiempo de estabilización y número de oscilaciones; cada una de estas se pueden variar de acuerdo a la respuesta deseada, la fórmula general de un PID está determinada por la ecuación (5).

<sup>39</sup> Controladores PID. Control automático 1, 2002. Universidad Nacional de Quilmes. p. 1-2

<sup>40</sup> The Ziegler Nichols Method. Video. [en línea] < <https://www.youtube.com/watch?v=nvAQHSe-Ax4> >. Fecha de consulta 05/10/2015.

$$u(t) = k_p e(t) + \frac{k_p}{T_i} \int_0^t e(\tau) d\tau + k_p T_d \frac{de(t)}{dt} \quad (5)$$

El control mediante este método no solo tiene que ser PID, también se puede presentar según el tipo de control que se desee implementar, ya sea un controlador PI o PD. El control PI (control proporcional integral) brinda una corrección de acuerdo al error actual, la corrección es como su nombre lo indica “proporcional al error medido”, este control tiene en cuenta los errores anteriores, para ello es necesario que exista un error con el fin que la respuesta sea diferente de cero, la expresión matemática para un control PI está determinada por la ecuación (6)<sup>41</sup>.

$$u(t) = k_p e(t) + \frac{k_p}{T_i} \int_0^t e(\tau) d\tau \quad (6)$$

Un control PD está determinado por la misma parte proporcional más una derivativa la cual aporta una predicción del error, esto actúa disminuyendo el tiempo de estabilización, el control derivativo permite una mayor precisión en estado estable ya que la acción de corrección se efectúa antes de que el error sea demasiado grande; la ecuación (7) determina lo anterior.

$$u(t) = k_p e(t) + k_p T_d \frac{de(t)}{dt} \quad (7)$$

El control por cumplimiento de parámetros no es más que una programación donde se evalúan eventos, si un evento se cumple desencadena una serie de acciones las cuales pueden dar paso a otra condición para desatar el siguiente suceso: este tipo de control es muy frecuente en el campo de la automatización industrial donde brazos robóticos cumplen la labor de tareas repetitivas o paso a paso.

## 2.5 PLANEAMIENTO DE TRAYECTORIA O RUTA

Dentro del planeamiento de trayectorias existen dos grandes tipos: deseadas y autónomas; las rutas deseadas son aquellas que se determinan para que se ejecuten siempre de la misma manera, estas son fabricadas mediante diferentes métodos como el punto a punto o linealmente; el método de punto a punto se hace con el fin de generar una mayor precisión en la ejecución de la ruta ya que la trayectoria final estará conformada por varios puntos, al estar compuesta por

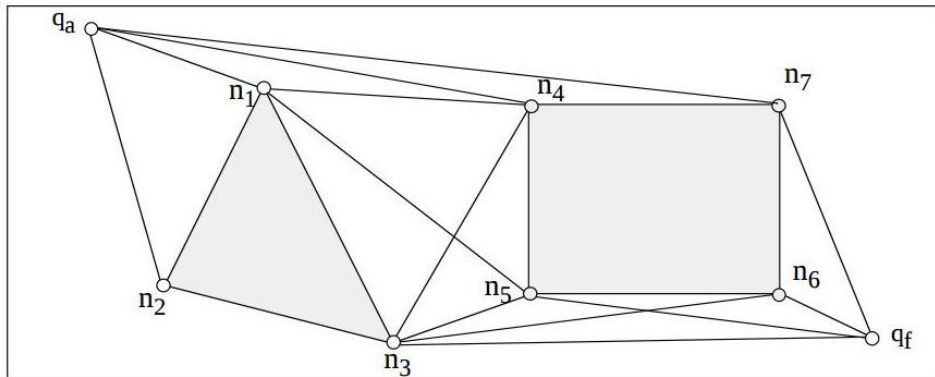
---

<sup>41</sup> Control PID Clásico. [en línea] < [http://csd.newcastle.edu.au/SpanishPages/clase\\_slides\\_download/C07.pdf](http://csd.newcastle.edu.au/SpanishPages/clase_slides_download/C07.pdf)  
> Fecha de consulta 05/10/2015.

puntos el robot interpreta la ruta final como la composición de muchas mini rutas lo cual es de gran ayuda, ya que en cada tramo del recorrido el robot evaluará y corregirá los movimientos precisos para llegar a su destino. El método lineal corresponde al trazo de líneas rectas las cuales el robot tendrá que tratar de seguir y hacer las correcciones pertinentes en la marcha minimizando el error teniendo como referencia la línea que describe la ruta<sup>42</sup>.

Dentro de los métodos de planificación de trayectorias existen algunos que parten de la visualización del entorno en el que se desarrolla el movimiento con el fin de visualizar posibles obstáculos, uno de estos métodos es el de grafos de visibilidad, el cual parte de una solución geométrica para el trazo de la trayectoria asignándole formas geométricas a los obstáculos y sabiendo el punto de inicio y final de la trayectoria, las rutas posibles serán la unión entre dos puntos si y sólo si entre ellas no hay ningún obstáculo, la unión aleatoria de los puntos de inicio y final da como resultado todas las posibles trayectorias como se muestra en la (figura 36)<sup>43</sup>.

**Figura 36.** Grafo visible



**Fuente.** Planificación de Trayectorias para Robots Móviles capítulo 2. Pg. 37

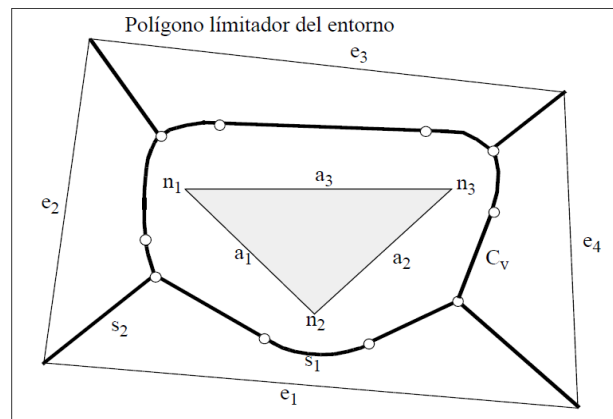
Como se observa en la (figura 36) existen dos obstáculos (un triángulo y un cuadrado) entre los puntos de inicio ( $q_a$ ) y final ( $q_f$ ) la unión de varias líneas en un solo punto es denominado nodo visible, la unión de nodos visibles teniendo en cuenta el punto de partida y de llegada da como resultado una ruta.

<sup>42</sup> Interpolación y extrapolación lineal SECUNDARIA. Video. [en línea] < <https://www.youtube.com/watch?v=pzDfi2jVDGs> > Fecha de consulta 05/10/2015

<sup>43</sup> Seco Granja, Fernando. Sistemas de localización en interiores basados en radiofrecuencia. [en línea] < <http://www.car.upm-csic.es/lopsi/static/publicaciones/docencia/Apuntes%20RF-LPS.pdf> > Fecha de consulta 29/10/2015.

A diferencia de los grafos de visibilidad, el diagrama de VORONI plantea algo totalmente opuesto para la generación de una ruta, la cual parte de la evasión total de los obstáculos generando una distancia mínima que rodea los mismos, esta distancia estará directamente enlazada a la cinemática del vehículo respondiendo a la pregunta ¿cuánto espacio necesito para moverme? El diagrama de VORONI asegura la posibilidad de llegar a cualquier punto alrededor del obstáculo mediante trechos de segmentos de recta o parábolas como se muestra en la (figura 37).

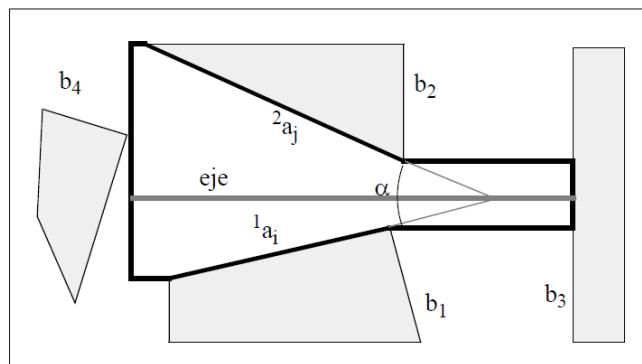
**Figura 37.** Diagrama de VORONI



**Fuente.** Planificación de Trayectorias para Robots Móviles capítulo 2. Pg. 40

Al igual que VORONI el modelado de espacio libre plantea que se debe navegar a una distancia considerable entre los obstáculos, para lo cual se identifican las aristas que lo conforman y cuales tienen cercanía al vehículo, cuando dos aristas se ven enfrentadas el espacio entre ellas, será el espacio libre para la ejecución del movimiento, de esta manera se plantea una zona libre de obstáculos para el desarrollo de la trayectoria como se muestra en la (figura 38).

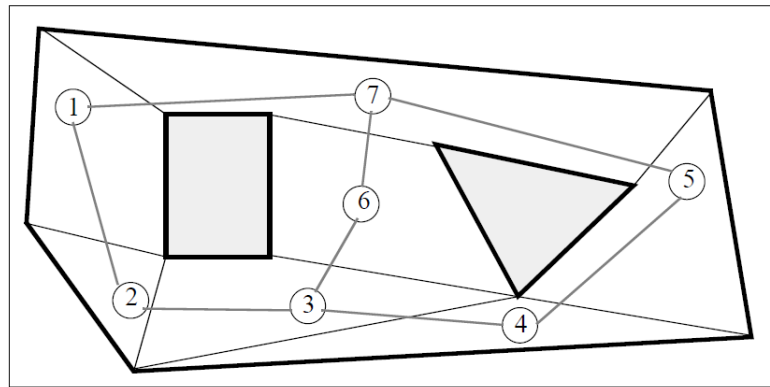
**Figura 38.** Diagrama de espacio libre



**Fuente.** Planificación de Trayectorias para Robots Móviles capítulo 2. Pg. 51

Otro método de trazo de trayectorias consiste en descomposición de celdas el cual parte del principio de conectividad entre dos puntos sin interrupción por obstáculos entre los mismos, para esto se genera un diagrama de conectividad el cual tendrá en cuenta el espacio libre entre los obstáculos y el escenario, de acuerdo a esto el diagrama de conectividad posicionará puntos entre los espacios libres como se ve en la (figura 39).

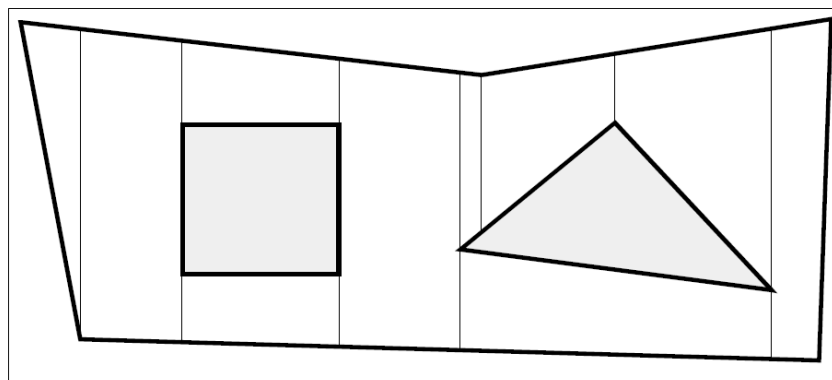
**Figura 39.** Diagrama de conectividad



**Fuente.** Planificación de Trayectorias para Robots Móviles capítulo 2. Pg. 53

Una vez trazado el diagrama de conectividad se buscan las celdas que satisfacen este diagrama mediante el método de descomposición trapezoidal el cual consiste en trazar líneas paralelas al eje y teniendo en cuenta los espacios libres y dicho diagrama de conectividad como se muestra en la (figura 40).

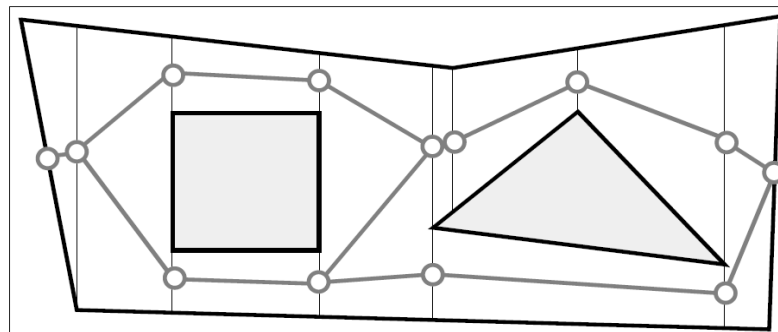
**Figura 40.** Diagrama de celdas



**Fuente.** Planificación de Trayectorias para Robots Móviles capítulo 2. Pg. 55

El grafo de conectividad final se da en la ubicación de los puntos del diagrama de conectividad en los puntos medios de las celdas encontradas como se muestra en la (figura 41).

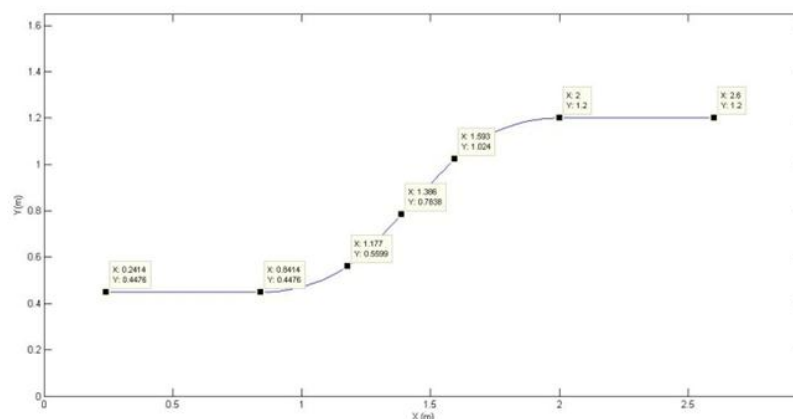
**Figura 41.** Diagrama de conectividad



**Fuente.** Planificación de Trayectorias para Robots Móviles capítulo 2. Pg. 56

Una vez conocida la ruta que se debe describir, existen métodos que satisfacen la necesidad de minimizar errores en la ejecución, por ejemplo Spline que se basa en la interpolación de puntos, el cual recopila la trayectoria descrita por una función, de allí es posible conocer el punto de inicio y el punto final de la ruta. Con el fin de recrear la ruta mediante puntos, el Spline interpola los mismos y los ubica sobre la recta que describiría la trayectoria si fuera lineal, la trayectoria final estará compuesta por puntos que describen un camino como se ilustra en la (figura 42)<sup>44</sup>.

**Figura 42.** Trayectoria Spline



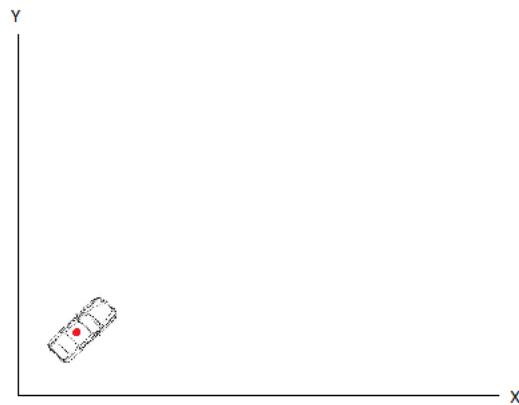
**Fuente.** Elaboración propia

<sup>44</sup> Planificación de Trayectorias para Robots Móviles, Capítulo 2.- 2.4



El trazo de trayectorias punto a punto es uno de los más utilizados ya que evalúa y corrige en tiempo real dirección y orientación, otros métodos también ejecutan este mismo proceso pero el método punto a punto genera pequeñas trayectorias entre ellos, minimizando el error de posición; el primer punto siempre será el de inicio del vehículo como se muestra en la (figura 43).

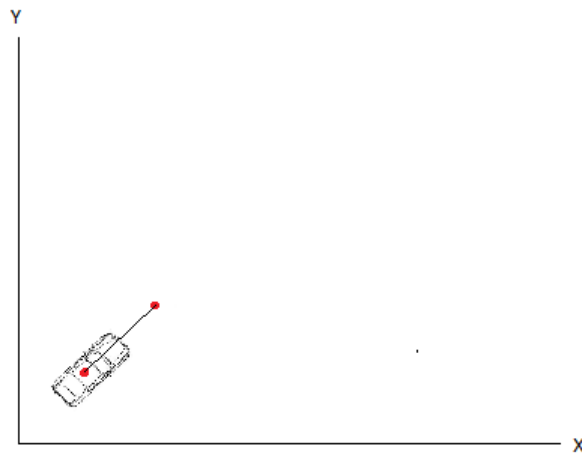
**Figura 43.** Punto de inicio



**Fuente.** Elaboración propia

El segundo paso es trazar el siguiente punto el cual es el destino de llegada, como se muestra en la (figura 44) para posicionar el siguiente punto se debe tener en cuenta las restricciones físicas del carro, trazando el segundo punto da como resultado una mini ruta entre estos haciendo acciones de corrección de dirección y/o velocidad para llegar a su destino.

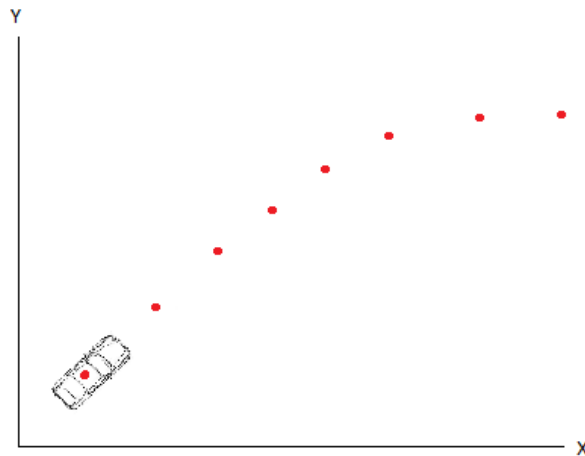
**Figura 44.** Punto de destino



**Fuente.** Elaboración propia

Cada vez que el vehículo llega a su destino, este cambia al próximo punto generando de esta manera mini rutas que cumplen una trayectoria general como se muestra en la (figura 45).

**Figura 45.** Trayectoria general



**Fuente.** Elaboración propia

Por otro lado las rutas autónomas evalúan dos puntos, el de inicio y el de llegada; aunque el concepto es bastante sencillo la complejidad radica en la cantidad de variables a tener en cuenta para el trazo de la ruta; por ejemplo si el movimiento se ejecutará en un ambiente controlado solo se deberá tener en cuenta la cinemática del robot y la orientación inicial y final deseada de ser el caso. El caso contrario es cuando el vehículo se desenvolverá en escenarios reales donde se tienen en cuenta no solo factores de la naturaleza como fricción, humedad, vientos, tipo de terreno sino también la posibilidad de encontrar obstáculos en el camino, los cuales el robot debe detectar y de acuerdo a ello corregir su ruta.

Todo lo anterior es programado mediante algoritmos que tienen en cuenta la percepción, la ruta que se debe cumplir, y las restricciones físicas del robot móvil; en conjunto se genera el trazo y corrección de trayectorias.

### 3 VISIÓN ARTIFICIAL

La Visión Artificial o también llamada Visión por Computador, es una rama de la inteligencia artificial, la cual es una técnica basada en la captura de la información visual del entorno físico, con el fin de modelar una imagen digital suministrada por un sensor, de manera tal que un ordenador sea capaz de procesarlas y determinar sus propiedades, para poder usarlas como base para la toma de decisiones. De forma tal que es una tecnología la cual combina las cámaras de video y el uso de los computadores con el objeto de asemejar la utilidad de la visión humana.

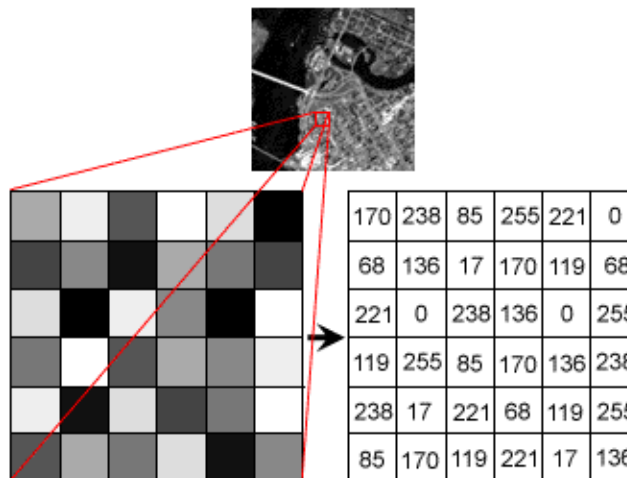
La visión artificial compone un conjunto de procesos destinados a realizar análisis de imágenes y su utilidad en el proyecto radica en la implementación de aplicaciones basadas en identificación de objetos, establecimiento de relaciones espaciales, determinación de posición de los objetos y sus coordenadas en el espacio<sup>45</sup>.

#### 3.1 CONCEPTOS TEÓRICOS GENERALES

##### 3.1.1 Imagen digital

Es una representación bidimensional de una imagen, la cual consiste en la colección ordenada de valores (matriz) constituidos por filas y columnas que representan cada uno de los pixeles o puntos de color de la imagen (figura 46), de igual manera se pueden visualizar por medio de un monitor, papel o cualquier otro dispositivo electrónico relacionado.

**Figura 46.** Representación de una imagen digital



**Fuente.** Grossmont College Intro to Geographic Information Systems. URL: <http://www.grossmont.edu/people/judd-curran/intro-to-gis/gisoutline.aspx>

<sup>45</sup> Visión artificial. Aspectos de un Sistema de Visión Artificial. 2005. p. 1.

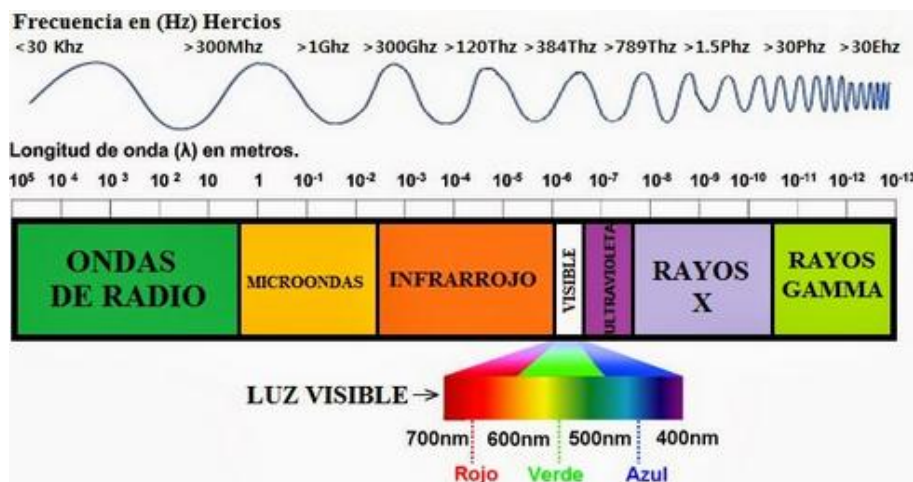
La imagen digital puede ser obtenida a través de distintos tipos de conversión analógica/digital como lo son cámaras digitales, escáneres, o por medios computacionales, por ejemplo, los programas de dibujo.

### 3.1.2 Espectro visible

El espectro visible es una parte del conjunto de ondas electromagnéticas, la cual el ojo humano es capaz de percibir y procesar mediante el cerebro; la radiación electromagnética comprendida en dicho rango se denomina luz visible<sup>46</sup>.

La (figura 47) presenta la disposición del rango de luz visible por el ser humano dentro del espectro electromagnético. De acuerdo a ello el ser humano es capaz de percibir radiaciones de longitudes de onda dentro del margen de 380 nanómetros hasta los 780 nanómetros. El sentido de la vista transforma dichas amplitudes y frecuencias en características como color y brillo<sup>47</sup>.

**Figura 47.** Espectro electromagnético (luz visible)



**Fuente.** Electrónica especializada: Medios y equipos .URL: <http://americamilitar.com/temas-generales-de-las-ffmm/1999-electronica-especializada-medios-y-equipos.html#gsc.tab=0>

### 3.1.3 Ojo humano

El ojo humano es un órgano el cual tiene una forma aproximadamente esférica de unos 2.5 cm de diámetro y su función consiste en transformar la luz visible en impulsos, los cuales son enviados al cerebro a través del nervio óptico. El ojo es el elemento transductor mientras que el cerebro procesa dicha información.

<sup>46</sup> Espectro electromagnético. Espectro visible. [en línea] < [http://educativa.catedu.es/44700165/aula/archivos/repositorio/3000/3233/html/21\\_espectro\\_visible.html](http://educativa.catedu.es/44700165/aula/archivos/repositorio/3000/3233/html/21_espectro_visible.html) >

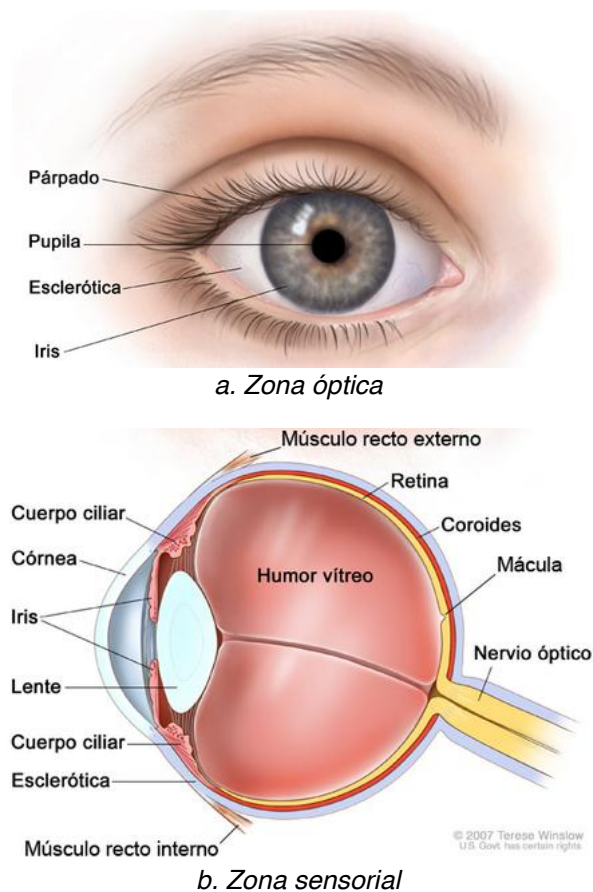
<sup>47</sup> Ibid., [en línea]

### 3.1.3.1 Anatomía del ojo humano

El ojo humano está formado por una zona óptica y una zona sensorial. La (figura 48) presenta la composición anatómica del ojo humano, en la que se puede observar en primera instancia la zona óptica (figura 48a), la cual está compuesta por el párpado, el cual es el músculo mucoso situado delante del bulbo ocular y su función es mantener la integridad de la superficie del ojo.

De igual manera se encuentra la pupila, la cual es el orificio situado en la parte central del iris que regula la cantidad de luz que entrante a la retina. Por otra parte la esclerótica es una membrana de color blanco compuesta por una serie de fibras de tejido de colágeno, su función es mantener la forma del ojo. Finalmente se encuentra el iris el cual es la membrana pigmentada la cual tiene en su centro la pupila y su función principal es controlar la cantidad de luz que penetra en el ojo<sup>48</sup>.

**Figura 48.** Anatomía del ojo humano



**Fuente.** Imagui. *El ojo humano y sus partes.* URL: <http://www.imagui.com/a/el-ojo-humano-y-sus-partes-dibujo-cLLrdb05X>

<sup>48</sup> Historia y biografías. La vista: órganos de los sentidos humanos vista, oído, gusto, olfato, tacto. [en línea] <<http://historiaybiografias.com/sentido1/>>

En la (figura 48b) se presenta las partes que constituyen la zona sensorial del ojo, los partes (principales) son en primer lugar el nervio óptico el cual es el nervio encargado de recoger las sensaciones que provienen de la retina para enviarlas al cerebro. De igual manera se encuentra la retina, que es el tejido que cuando percibe la incidencia luz, produce una serie de reacciones químicas e impulsos eléctricos que son enviados al nervio óptico<sup>49 50</sup>.

Por otra parte se encuentra el cristalino, el cual es el lente de enfoque del ojo, que mediante la interacción de los músculos que lo rodean permite enfocar a diferentes distancias los objetos de interés. De igual manera se encuentra la mácula, la cual es una mancha amarilla localizada en la retina, su función principal es el detalle y la finesa en las imágenes, por ejemplo leer o detallar pequeños objetos y finalmente el cuerpo ciliar, que es un músculo situado a continuación del iris y se caracteriza por modificar la curvatura del cristalino<sup>51</sup>.

### **3.1.3.2 Funcionamiento del ojo**

La visión es el sentido más importante que tiene el ser humano, este proceso es llevado a cabo mediante la interacción de dos elementos básicos: el ojo y el cerebro. Dado a que el sentido de la vista es primordial a la hora de interactuar en el ambiente, es necesario conocer como es el funcionamiento y formación de las imágenes.

Este complejo funcionamiento se da en primera instancia cuando los rayos de luz inciden en el ojo a través de la córnea, a continuación pasan a través de la pupila la cual se expande o se contrae de acuerdo a la intensidad; entre más luz penetre la pupila será más pequeña y entre mayor ausencia de luz aumentará de tamaño.

Seguidamente el cristalino realiza el proceso de enfoque de la imagen y la retina recibe en sus paredes la imagen invertida, tal como se observa en la (figura 49). Los conos y los bastones allí localizados transforman dicha información luminosa en impulsos eléctricos, los cuales viajan a través de fibras nerviosas hacia el nervio óptico<sup>52</sup>.

---

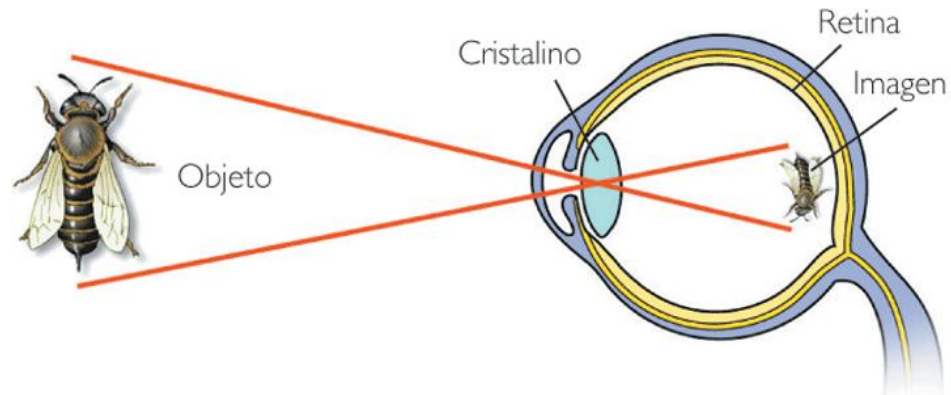
<sup>49</sup> Ibid., [en línea]

<sup>50</sup> Avilés, Asturias. Óptica, el ojo humano. IES la Magdalena. p. 1.

<sup>51</sup> Ibid., p. 2.

<sup>52</sup> Vaqué Alcázar, Carlos, De Andrés. El ojo humano, ¿Cómo funciona? [en línea] <  
<http://cienciainquieta.com/2013/12/02/el-ojo-humano-como-funciona/>> Fecha de consulta 26/10/15

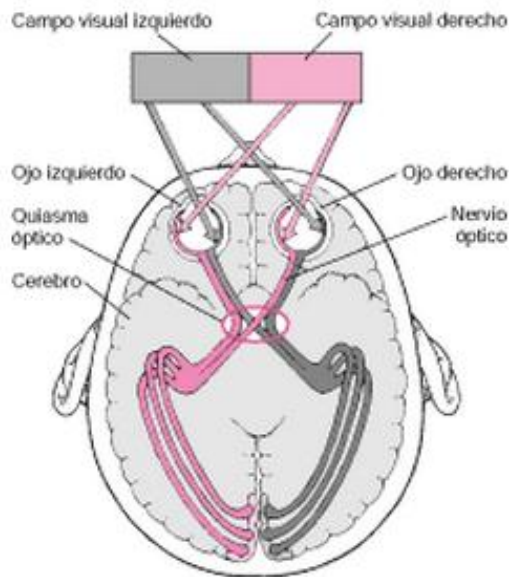
**Figura 49.** Formación de la imagen en el ojo



**Fuente.** Un fotorreceptor altamente especializado. URL: <http://lacienciaysusdemonios.com/2010/09/28/el-disenador-poco-inteligente-ii-tragaluces/>

Las señales eléctricas captadas por el nervio óptico, llegan a la parte del lóbulo occipital del cerebro, donde se localiza el córtex visual (figura 50), en el que la información ojo izquierdo llega al hemisferio derecho y viceversa. En esta parte el cerebro endereza la imagen invertida, y realiza la extracción de características de la imagen<sup>53</sup>.

**Figura 50.** Córtex visual



**Fuente.** Platero, Carlos. *Introducción a la visión artificial*. Dpto. Electrónica, Automática e informática Industrial. pág. 23

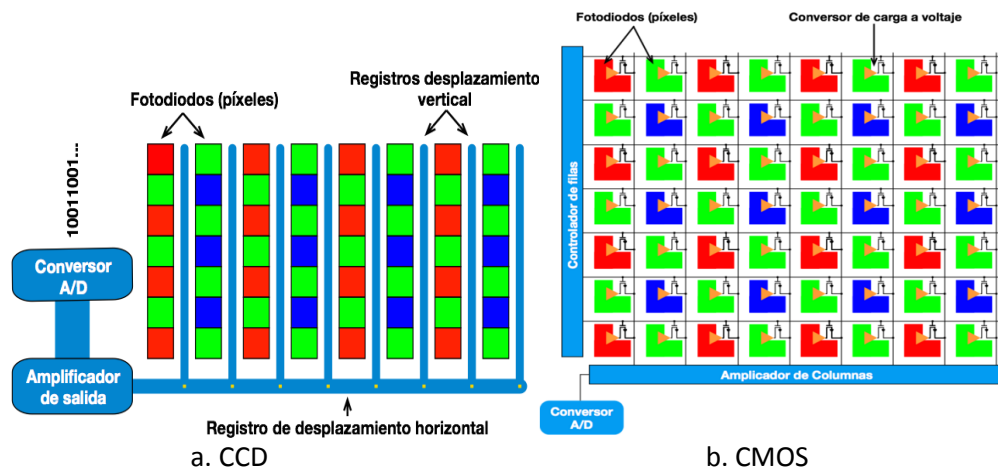
<sup>53</sup> Ibid., [en línea]

### 3.1.4 Cámara

Las cámaras en la visión artificial cumplen un papel esencial, ya que permiten capturar imágenes con el fin de transmitirla a un sistema computacional con el objeto de que pueda ser interpretada y visualizada. Las cámaras actualmente basan su funcionamiento en la aplicación (mayormente utilizada) de la tecnología CMOS (Complementary Metal-Oxide-Semiconductor) y CCD (Charge Coupled Devide), cuya función es transformar las señales de fotones en imágenes.

Como se puede observar en la (figura 51), el sensor de la cámara está compuesto por una serie de fotodiodos, los cuales tienen la función de captar los fotones y convertirlas en señales eléctricas. El sensor de la cámara cubre un área o matriz de píxeles donde llegan los fotones, por tal, el número de píxeles es determinado por la cantidad de fotodiodos existentes en el sensor<sup>54</sup>.

**Figura 51.** Sensor fotosensible CCD y CMOS



**Fuente.** Evolución de los sensores digitales en fotografía, de CCD a CMOS

Una vez de que los fotodiodos envían la señal eléctrica, el amplificador de salida convierte dicha señal en un voltaje análogo, que después será digitalizada por medio de un convertidor (A/D), proceso realizado en un ordenador<sup>55</sup>.

### 3.1.5 Similitudes entre el ojo humano y la cámara fotográfica

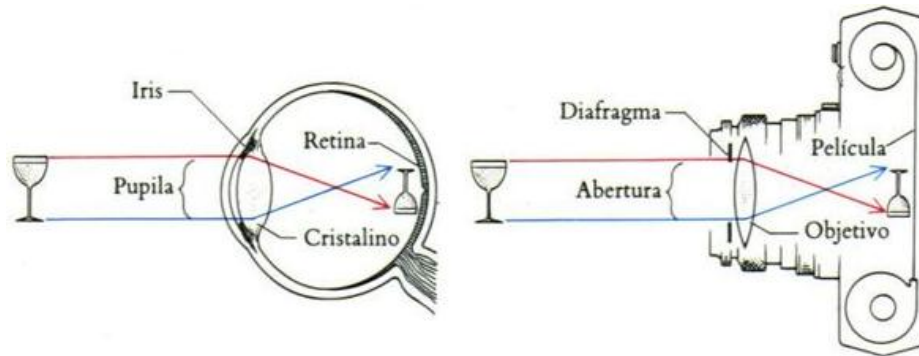
El funcionamiento del ojo humano es semejante al de la cámara fotográfica, dado a que los procesos de obtención de la imagen tienen puntos de comparación relevantes y el resultado final es el mismo, como se puede observar en la (figura 52).

<sup>54</sup> Visión artificial. Aplicación práctica de la visión artificial en el control de procesos industriales. 2012. p. 9

<sup>55</sup> Evolución de los sensores digitales en fotografía, de CCD a CMOS. 2015.



**Figura 52.** Anatomía del ojo humano frente a la cámara fotográfica



**Fuente.** Imágenes digitales. URL: <http://www.monografias.com/trabajos-pdf5/manual-administracion-e-informatica-modulos-1-y-2/manual-administracion-e-informatica-modulos-1-y-2.shtml>

Es claro que existen diferencias en relación a la calidad y de la forma en como se muestra la imagen, por un lado el cerebro muestra la imagen final, mientras que en la cámara es necesario un sistema visual para proyectarla. En la (tabla 2) se exponen las principales similitudes entre el sistema visual humano respecto al de la cámara fotográfica en función de la obtención de la imagen.

**Tabla 2.** Similitudes del ojo humano frente a la cámara fotográfica

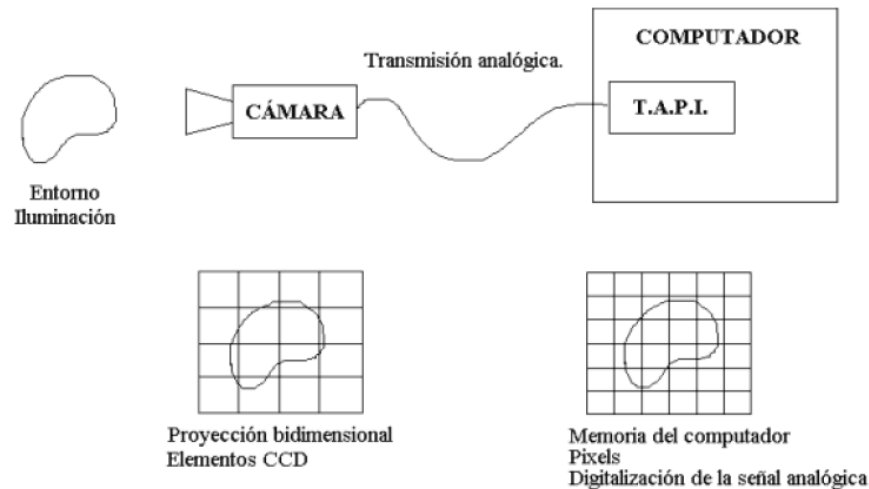
	Ojo humano	Cámara
<b>Comparativo</b>	La córnea y el cristalino son las capas que están sobre el ojo, ayudan a enfocar la luz	El lente es el componente que regula el foco y regula la nitidez de la imagen.
	El iris es el tejido pigmentoso que regula la cantidad de luz que entra al ojo.	El diafragma es un sistema mecánico o electrónico que gradúa el paso de intensidad de luz.
	La retina es la capa más profunda del ojo, contiene las células nerviosas que forman la imagen al cerebro.	La película fotográfica es donde se imprimen las imágenes provenientes del sensor de la cámara.

**Fuente.** Elaboración propia

### 3.2 CONFIGURACIÓN BÁSICA DE UN SISTEMA DE VISIÓN ARTIFICIAL

El desarrollo de la visión artificial está compuesto por dos partes físicas principales, la primera de ellas es el sistema de formación de las imágenes y en segundo lugar el sistema de procesamiento de las mismas. De tal manera que el objeto de aplicar la visión artificial es modelar una imagen bidimensional para que un computador pueda procesarla para el requerimiento a desarrollar, tal como se puede observar en la (figura 53).

**Figura 53.** Sistemas de un equipo de visión artificial



**Fuente.** Platero, Carlos. *Introducción a la visión artificial. Dpto. Electrónica, Automática e informática Industrial. pág. 26*

En el desarrollo de dicho proceso intervienen una serie de subsistemas que dan como resultado la aplicación de la visión artificial. El primer sistema está constituido por un subsistema de iluminación el cual es el conjunto de implementos que producen radiación electromagnética que inciden sobre los objetos a visualizar, es decir, el uso de implementos como lámparas, pantallas fotográficas, luz natural, entre otras<sup>56</sup>.

De igual manera se compone por un subsistema de captación, que son aquellos transductores que convierten la radiación luminosa en señales eléctricas, fundamentalmente se habla en esta parte del uso de las cámaras y de su aplicación incluso en el espectro no visible. Finalmente, se compone de un subsistema de adquisición, que es la señal eléctrica procedente de la cámara y por medio de la tarjeta de adquisición del ordenador forman la señal de video, para

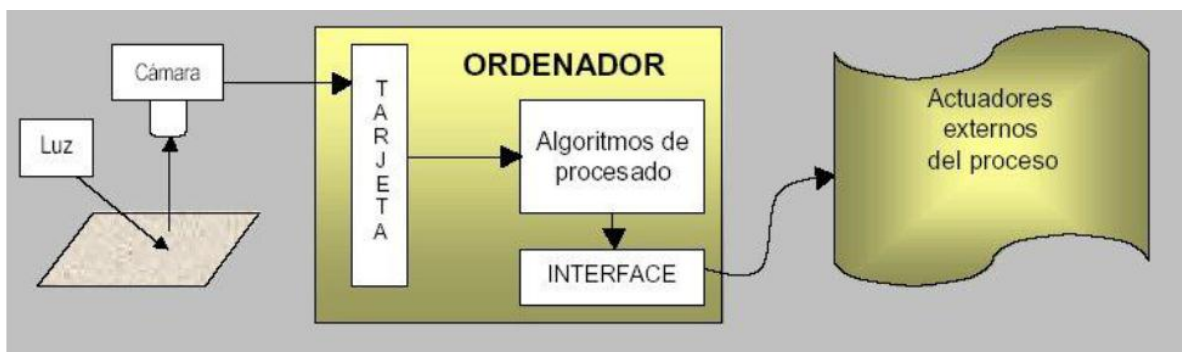
<sup>56</sup> Platero, Carlos. *Introducción a la Visión Artificial. 2009. p.25.*

que puedan ser tratadas; a la formación de dichas imágenes se les denomina también como *frame*<sup>57</sup>.

Por otra parte, el sistema físico de procesamiento se encuentra compuesto por dos subsistemas, el primero de ellos generalmente es un computador o un grupo de computadores, acorde a la necesidad de la aplicación. En esta parte el ordenador recibe la representación de la imagen digital y de acuerdo a la composición de los algoritmos programados procesa dicha información hasta alcanzar el tipo de información deseada<sup>58</sup>.

Finalmente, el segundo subsistema está integrado por los periféricos, que es el conjunto de elementos que reciben toda la información procesada del ordenador, la cual puede ser utilizada para la representación visual de datos del proceso o para ser fuente de datos en la actuación del robot<sup>59</sup>. La configuración completa del sistema de visión artificial se ilustra en la figura 54.

**Figura 54.** Subsistemas de un equipo de visión artificial



**Fuente.** Platero, Carlos. *Introducción a la visión artificial*. Dpto. Electrónica, Automática e informática Industrial. pág. 26

### 3.3 REPRESENTACIÓN DE LAS IMÁGENES EN LOS COMPUTADORES DIGITALES

En el proceso de visión artificial, las imágenes se adquieren a través de la cámara de video, empleando la tarjeta de memoria de adquisición con el objeto de que sean procesadas por el ordenador, de tal manera que la señal es de carácter bidimensional. Es por ello que dentro de los desarrollos de procesamiento de imagen existen diferentes tipos de imágenes digitales, con las cuales se realizan los respectivos tratamientos de información en el ordenador.

<sup>57</sup> Ibid., p.25

<sup>58</sup> Ibid., p.26

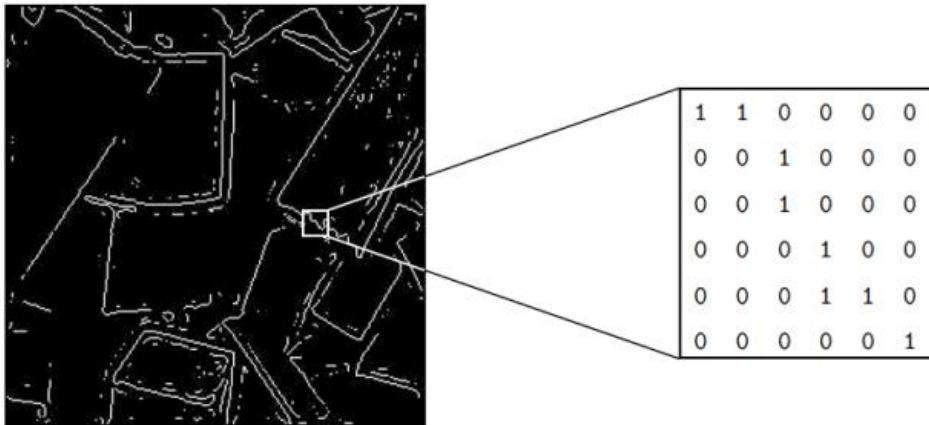
<sup>59</sup> Ibid., p.26

### 3.3.1 Imágenes binarias

Las imágenes binarias son un tipo de imagen digital en la que los valores de los píxeles adquieren un valor discreto, de manera tal que solo existen dos valores posibles (1 y 0), en la que 1 representa el color blanco y el 0 representa el color negro tal como se puede observar en la (figura 55).

En el procesamiento digital de imágenes son muy útiles, ya que al usar el mínimo de datos necesarios para representar una imagen permite el máximo aprovechamiento de capacidad computacional, es decir, que la velocidad en que se procesan es mucho más eficiente<sup>60</sup>.

**Figura 55.** Imagen Binaria



**Fuente.** Flores Pulido, Leticia. *Procesamiento digital de imágenes y video*. URL: [https://aicitel.files.wordpress.com/2011/08/clase3\\_pdi.pdf](https://aicitel.files.wordpress.com/2011/08/clase3_pdi.pdf)

### 3.3.2 Imágenes en escala de grises

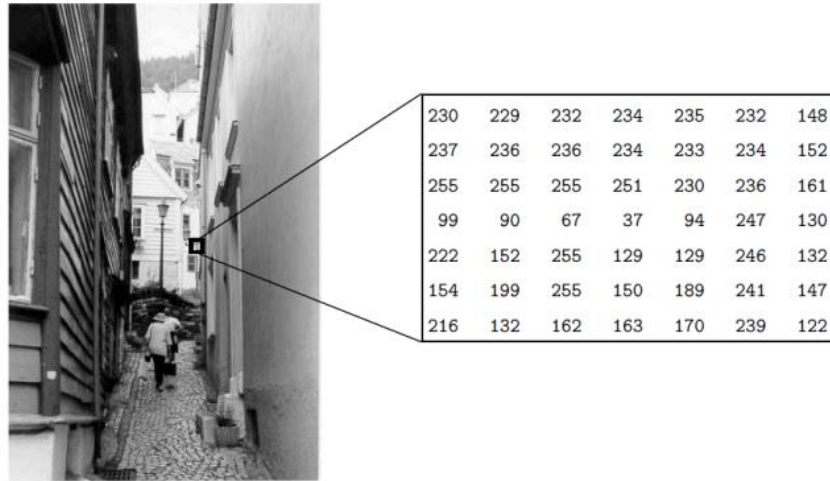
Las imágenes en escala de grises, están compuestas por intensidades de gris en la que los píxeles son representados en el rango de 8 bits, es decir, 256 valores de intensidad. La imagen varía desde (0) para representar el color negro, hasta (255) que representa el color blanco, tal como se expone en la (figura 56).

Es muy utilizada en el procesamiento de imagen dado a que sus rangos (8 bits) permiten la manipulación de información para realizar todo tipo de aplicación (segmentación, filtros, binarización, etc.) en una imagen digital<sup>61</sup>.

<sup>60</sup> E. de la fuente F.M. trespaderne. Imágenes binarias. Capítulo 4.

<sup>61</sup> Glosario Gráfico. Un diccionario de artes gráficas, diseño y materias a fines. [en línea] < [http://glosariografico.com/escala\\_grises](http://glosariografico.com/escala_grises) > Fecha de consulta 27/10/15.

**Figura 56.** Imagen en escala de grises



**Fuente.** Flores Pulido, Leticia. *Procesamiento digital de imágenes y video*. URL: [https://aicitel.files.wordpress.com/2011/08/clase3\\_pdi.pdf](https://aicitel.files.wordpress.com/2011/08/clase3_pdi.pdf)

### 3.3.3 Imágenes indexadas

El modo de imagen indexada se caracteriza por usar una paleta exacta de colores, con un máximo de 256 tonos. La estructura de la imagen cuenta con valores enteros para cada píxel y cada uno almacena un mapa de color característico, con un rango de 0 a 1 tal como se puede observar en la (figura 57)

**Figura 57.** Imagen Indexada



**Fuente.** Flores Pulido, Leticia. *Procesamiento digital de imágenes y video*. URL: [https://aicitel.files.wordpress.com/2011/08/clase3\\_pdi.pdf](https://aicitel.files.wordpress.com/2011/08/clase3_pdi.pdf)

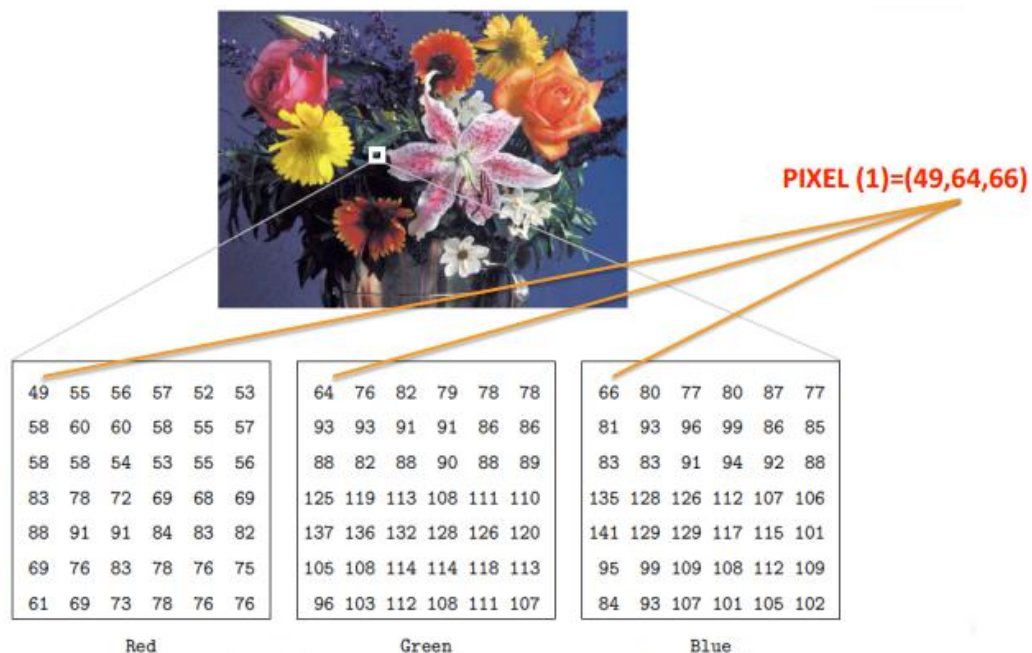
La particularidad de este tipo de imagen es que reduce la gama cromática, con lo que el tamaño del archivo y necesidad de procesamiento se reducen considerablemente<sup>62</sup>.

### 3.3.4 Imágenes a color (RGB)

Las imágenes RGB, es la representación de las imágenes a color, en el que cada uno de sus pixeles están formados por tres valores de intensidad de color específicas, que representa el rojo, verde y azul respectivamente, tal como se observa en la (figura 58).

Dado a que cada componente de color dentro del píxel, puede tomar un valor numérico dentro del rango de 0 a 255, se generan más de 16 millones de representación de color diferentes, dentro de una misma imagen<sup>63</sup>.

**Figura 58.** Imagen a color (RGB)



**Fuente.** Flores Pulido, Leticia. *Procesamiento digital de imágenes y video*. URL: [https://aicitel.files.wordpress.com/2011/08/clase3\\_pdi.pdf](https://aicitel.files.wordpress.com/2011/08/clase3_pdi.pdf)

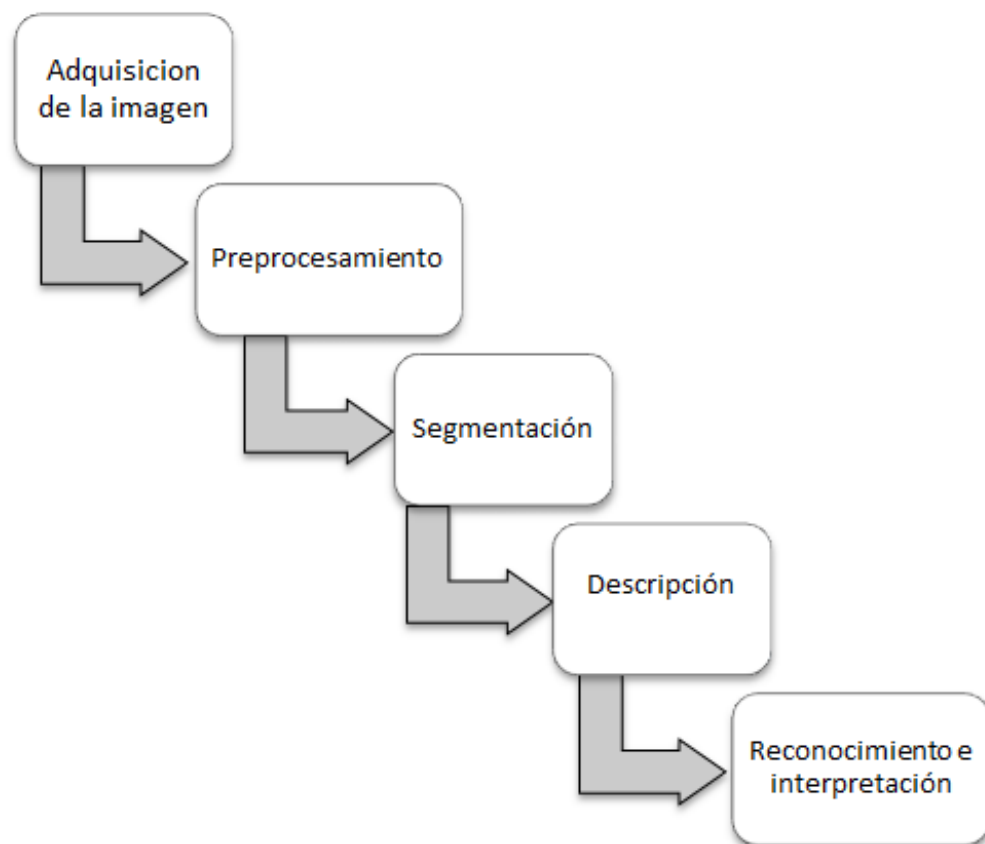
<sup>62</sup> Hablamos de comunicación multimedia. Modos o modelos de color LAB e indexado: que son y usos específicos. 2010.

<sup>63</sup> Hablamos de comunicación multimedia. Modos o modelos de color RGB y CMYK: que son y usos específicos. 2010.

### 3.4 ETAPAS BÁSICAS EN EL PROCESO DE VISIÓN ARTIFICIAL

El desarrollo de aplicaciones de Visión Artificial, están compuestas por una serie de etapas (figura 59) las cuales tienen el objeto de extraer, caracterizar e interpretar la información procedente de imágenes del mundo real. No necesariamente todas las etapas se involucran en una aplicación, dado que cada una de ellas posee un nivel de complejidad y funcionalidad distinta. En el gráfico se muestra un encadenamiento del proceso, pero ello no implica que no exista una interacción entre distintas fases; se realiza con el fin de facilitar la comprensión del proceso<sup>64</sup>.

*Figura 59. Etapas en el proceso de Visión Artificial*



*Fuente. Elaboración propia*

La primera etapa dentro del proceso de visión artificial está compuesta por la construcción del sistema de formación de las imágenes, es decir, se realiza la

---

<sup>64</sup> Aguilar Carrera, Gustavo German. Procesamiento digital de imágenes utilizando filtros morfológicos. 1995. p. 1-2.

adquisición de la imagen por medio del uso de un sensor, la cual posteriormente será digitalizada con el fin de poder realizar futuras aplicaciones con ella<sup>65</sup>. Una vez obtenida la imagen se involucra la siguiente etapa la cual es el preprocesado, cuyo objetivo es mejorar la calidad de la información de la imagen adquirida, en la que se incluyen aspectos como procesos de filtrado, supresión de ruido, atenuación de imperfecciones, mejoramiento de contraste, realce y restauración, entre otras<sup>66</sup>.

Seguidamente se encuentra la etapa de segmentación en la que se divide la imagen en regiones con diferentes fines, es decir, se extraen ciertas características de la imagen determinando zonas de interés para posteriormente ser analizadas. En esta parte se encuentran los procesos de detección de bordes, esquinas, umbralización, crecimiento de regiones, etc.<sup>67</sup>

Una vez dividida la imagen en zonas particulares se procede a realizar la descripción de la misma, la cual conlleva a extraer características de información tal como textura, color, tamaño o forma<sup>68</sup>.

La identificación de la información de los objetos caracterizados se realiza en la etapa del reconocimiento. Por último en la etapa de interpretación se evalúa la información del objeto de estudio y se pone en contexto con el mundo real, realizando la interpretación de la escena con el objeto de realizar aplicaciones basadas netamente en la información procesada<sup>69</sup>.

En general, el procesamiento de las imágenes involucran algunas o todas las etapas anteriormente mencionadas, de acuerdo a ello existe una clasificación por niveles, en las que se agrupan determinadas etapas del proceso de visión y tienen un papel importante dentro del desarrollo de una determinada aplicación. Se dividen en tres niveles, bajo medio y alto, como se puede observar en la (figura 60).

---

<sup>65</sup> Ibid., p.4

<sup>66</sup> Ibid., p.5

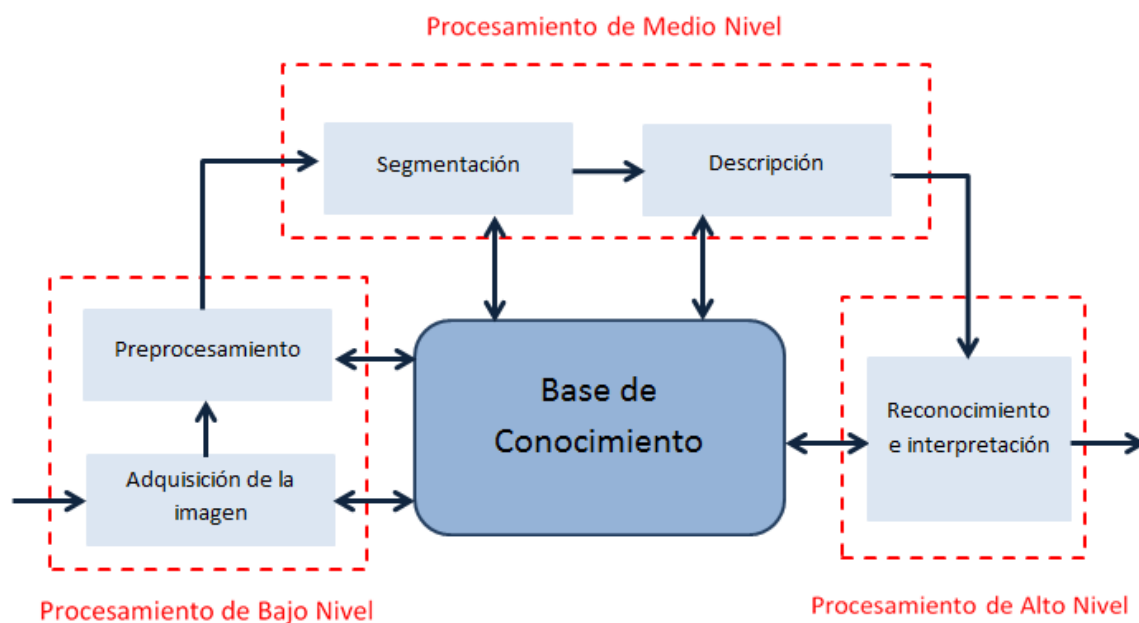
<sup>67</sup> Ibid., p.6

<sup>68</sup> Ibid., p.7

<sup>69</sup> Ibid., p.7



**Figura 60.** Niveles de procesamiento



**Fuente.** Elaboración propia

El procesamiento de bajo nivel está compuesto por las etapas de adquisición de la imagen y su respectivo pre-procesamiento. En el procesamiento de medio nivel se encuentran las etapas de segmentación de la imagen y la descripción de sus datos. Por último, se encuentra el procesamiento de alto nivel en la que se involucran las etapas de reconocimiento de los respectivos objetos e interpretación dentro de la escena. De acuerdo a ello el conocimiento como base en la aplicación de la visión artificial es importante ya que para realizar las aplicaciones es necesario conocer la problemática y saber que métodos o fases involucrar en su desarrollo.

De acuerdo a ello, se puede afirmar que el presente proyecto contempla la aplicación de los tres niveles de procesamiento, dado a que se implementan (en los dos primeros niveles) algoritmos para la detección de colores, contornos, y centroide de una figura. Adicionalmente se aplican algoritmos para la detección de líneas y figuras en base a características propias de la imagen en tiempo real.

Junto con ello se involucra el tercer nivel de procesamiento en el que se realiza un sistema de control y actuación basado en la información procesada de las imágenes captadas por una cámara de video. La sinergia de estos procesos da como resultado la aplicación de métodos de visión computacional orientados principalmente a generar desarrollos útiles en la detección de parámetros y control en vehículos autónomos.

### 3.5 VISIÓN ARTIFICIAL FRENTE A LA VISIÓN HUMANA

La visión es el sentido más importante que posee el ser humano, dado a que tiene más de dos millones de terminaciones nerviosas las cuales proporcionan el 75% de la información que los seres humanos perciben en el cerebro. El sistema de visión humano es muy complejo y eficaz, pero en ciertas circunstancias puede presentar algunas falencias; las ilusiones o información inconsistente respecto al mundo real pueden afectar la calidad de la visión.

Dado que la visión artificial todavía no ha creado sistemas o algoritmos tan complejos que emulen la visión humana, en determinados procesos a nivel tecnológico pueden desarrollar tareas mucho más efectivas como lo son acciones repetitivas, caracterizar información de una imagen, determinar la trayectoria de un vehículo en una pista, entre otras. Las principales ventajas de la visión artificial se muestran en la (tabla 3)<sup>70</sup>.

**Tabla 3.** Ventajas de la visión artificial frente a la visión humana

	<b>Visión Humana</b>	<b>Visión artificial</b>
<b>Comparativo</b>	Tiene una velocidad de respuesta de 0.06 segundos	Tiene una velocidad de respuesta de 0.00001 segundos
	Tiene la capacidad de reconocer 10 o 20 niveles de gris	Tienen una definición superior
	Posee poca precisión apoyándose en instrumentos de medida	Tienen gran precisión en la medición dependiendo de los componentes del sistema.
	Factores como la fatiga y las distracciones, le impiden tener consistencia.	Mantienen su nivel de rendimiento constante a lo largo de su vida útil.
	No puede ser expuesta a entornos peligrosos, (temperatura, contaminación, químicos, etc.)	Puede trabajar en ambientes peligrosos ya sea radiactivos, químicos, polución etc.

**Fuente.** Elaboración propia

<sup>70</sup> Visión artificial e interacción sin mandos. 2010. Asignatura de gráficos en computación. [en línea] <<http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/3D/VisionArtificial/index.html> >

## **3.6 FILTRADO DE IMÁGENES DIGITALES**

Los filtros digitales establecen su funcionamiento en la operación de convolución de una imagen y una función de filtro, con el objeto de modificar o mejorar las características propias de la imagen.

### **3.6.1 Técnicas de filtrado**

Se basan en métodos que permitan mejorar o realzar las características propias de una imagen, es decir, compone las técnicas de preprocesado (suavizado, detección de bordes, realce, eliminación de ruido, transformada de Fourier, etc.) los cuales permiten caracterizar de forma selectiva la información contenida en una imagen. Las técnicas de filtrado permiten interpretar la información o mejorar las propiedades de una imagen, de acuerdo a ello, es una parte esencial en el desarrollo de la visión artificial.

Dependiendo de la aplicación, es posible realizar distintos tipos de filtrado sobre una imagen, de acuerdo a ello el proceso de filtrado se puede llevar a cabo en el dominio del espacio (trabajando directamente sobre los píxeles de una imagen), o en el dominio de la frecuencia (operaciones realizadas mediante la transformada de Fourier)<sup>71</sup>.

#### **3.6.1.1 Filtrado en el dominio del espacio**

El filtrado espacial es la operación que se aplica a una imagen con el objeto de caracterizar detalles específicos para la interpretación y procesamiento posterior en el desarrollo de la visión artificial.

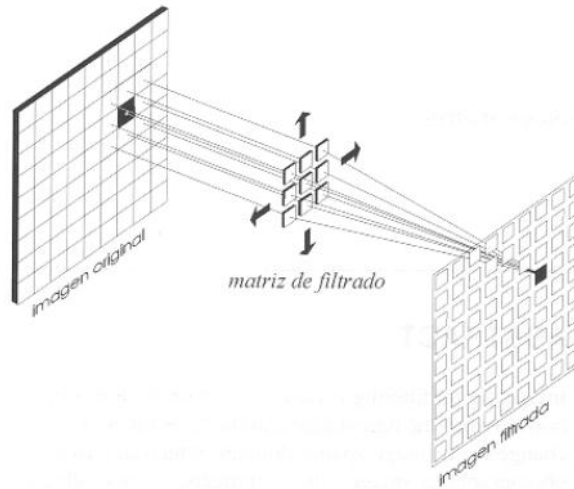
El filtrado espacial basa su funcionamiento en la operación directa sobre la imagen, es decir, consiste en la aplicación sobre cada píxel de la imagen de una matriz de filtrado de tamaño  $N \times N$ , que mediante operaciones de convolución genera un nuevo valor, el cual corresponderá al valor del píxel de la nueva imagen, tal como se puede observar en la (figura 61), en el que al costado izquierdo se encuentra la imagen original, en el centro la matriz de filtrado y al costado derecho imagen filtrada. Los filtros de este tipo de pueden clasificar como lineales y no lineales<sup>72</sup>.

---

<sup>71</sup> Visión Artificial. Preprocesado (realizado y filtrado) de imágenes digitales. Universidad rey Juan Carlos. Tema 2.

<sup>72</sup> Filtrado Espacial. 2005. Universidad Nacional de Quilmes. Ing. en Automatización y Control Industrial. Visión Artificial. p. 1.

**Figura 61.** Matriz de filtrado en el dominio espacial



**Fuente.** Revista de teledetección. Departamento de ingeniería Cartográfica, Geodésica y Fotogrametría. Universidad de Jaén. pp. 2

### 3.6.1.2 Filtrado en el dominio de la frecuencia

Este tipo de filtro basa su funcionamiento en el procesamiento de la imagen en el dominio de la frecuencia, es decir, trabaja sobre el espectro o distribución de frecuencias que componen la imagen, calculando la transformada de Fourier.

Para realizar este proceso, se aplica el teorema de la convolución, el cual en primera instancia calcula la transformada de Fourier de la imagen a tratar, posterior a ello se multiplica por el filtro específico y finalmente se aplica la transformada inversa de Fourier para obtener la imagen mejorada. Este tipo de filtro es aplicable para efectos de suavizado, atenuación, mejora en la detección de bordes etc. de acuerdo al nivel de frecuencias características de la imagen<sup>73</sup>.

---

<sup>73</sup> C. Pinilla, A. Alcalá, F Ariza. Revista de Teledetección. Filtrado de imágenes en el dominio de la frecuencia. p.2

## 4 MODELADO MATEMÁTICO

El modelo matemático permite identificar las características del vehículo, tal como se presenta en Hernández<sup>74</sup>, en el que incorpora las propiedades físicas del movimiento, representando su (posición y orientación); parámetros determinados a partir de un sistema global de referencia, sin tener en cuenta las fuerzas que puedan afectar su movimiento. De esta manera, para analizar el movimiento espacial del vehículo en función del tiempo, de acuerdo a parámetros geométricos y con el objeto de determinar la posición y orientación correctamente, se desarrolló el modelo cinemático del sistema.

Algunos desarrollos de investigación se enfocan en el análisis de las propiedades dinámicas del sistema basado en el desarrollo de relaciones geométricas implementados en <sup>75</sup> y <sup>76</sup>, teniendo en cuenta que los neumáticos del vehículo no presenten deslizamiento sobre la superficie de trabajo, para lo cual se han implementado modelos matemáticos de mayor complejidad que centran su investigación en el fenómeno de derrape del vehículo<sup>77</sup>, dado que el sistema propuesto se desarrolla a bajas velocidades de actuación, el modelo dinámico no fue tomado en cuenta en el desarrollo matemático.

La configuración espacial del vehículo en el plano se realizó teniendo en cuenta los parámetros usados en la ecuación (8):

$$X = \begin{bmatrix} X_c \\ Y_c \\ \theta \end{bmatrix} \quad (8)$$

Las coordenadas  $(X_c, Y_c)$  representan el punto de centro de gravedad del vehículo, identificándose en el espacio cartesiano y la orientación del robot ( $\theta$ ) indica la dirección del robot con respecto al sistema de referencia.

A partir de esta definición, se establece un sistema de coordenadas global cartesiano bidimensional con el objeto de poder identificar el vehículo en el

---

<sup>74</sup> Hernández Rubén, Bernardes Giovani, Vaqueiro Janito, Siqueira Pablo. Planeación y seguimiento de trayectorias de un vehículo terrestre con base en el control de dirección en un ambiente real, 2014. p.409.

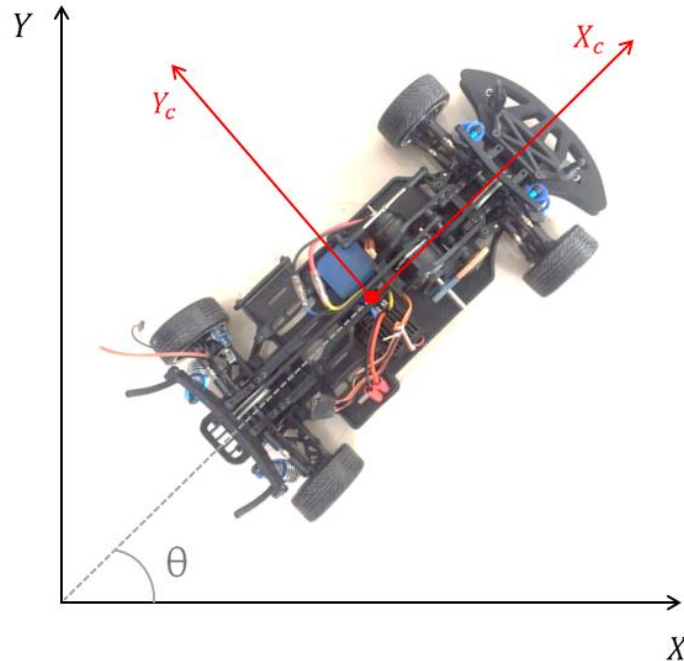
<sup>75</sup> Massera Filho, Carlos Alberto. Controle e Tomada de Decisao de Veículos Autonomos nos Limites da Estabilidade. 2015. p. 15-20.

<sup>76</sup> Snider, Jarrod M. Automatic Steering Methods for Autonomous Automobile Path Tracking. 2009. pp. 28-31.

<sup>77</sup> De Pavia, Azinaheira Jose, Bueno Samuel. Controle De Trajetória Para Veículos Terrestres De Exterior. 2010.

espacio, es decir, que se representa en el plano horizontal  $(X,Y)$ , como se muestra en la (figura 62).

**Figura 62.** Coordenadas del vehículo respecto al sistema global de referencia



**Fuente:** Elaboración propia

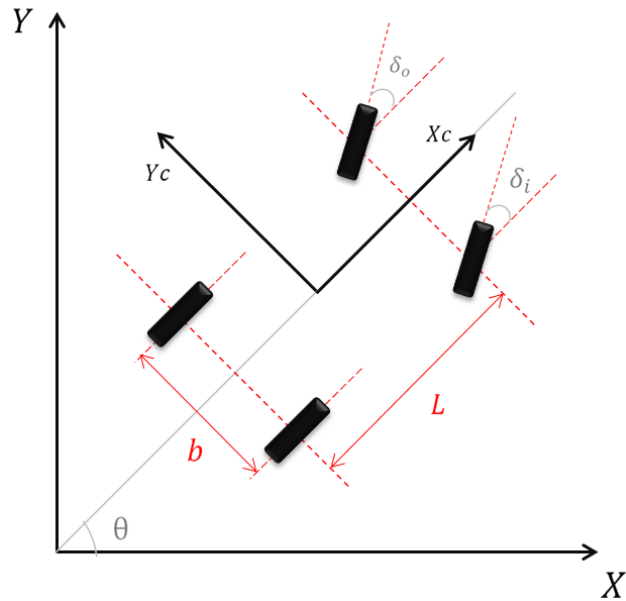
#### 4.1 MODELO CINEMÁTICO DEL VEHÍCULO

El modelo cinemático permite describir analíticamente el movimiento espacial del robot teniendo en cuenta la relación entre la posición y la orientación respecto a un sistema de coordenadas. La configuración que se utiliza generalmente para modelar este tipo de vehículo es la arquitectura Ackerman y es válido para ser usado en el prototipo a escala, “la ventaja de este modelo se encuentra en que evita el deslizamiento de las llantas bajo ciertas condiciones”<sup>78</sup>.

Este sistema posee dos ruedas traseras con un eje fijo y dos ruedas delanteras las cuales son operadas mediante una barra para controlar la dirección. Los componentes característicos de este modelo están integrados por un ángulo de dirección ( $\delta_o$ ) ( $\delta_i$ ), un ángulo de orientación ( $\theta$ ), la separación longitudinal entre las ruedas ( $L$ ), la separación lateral entre las ruedas ( $b$ ), y el punto de centro de gravedad del vehículo ( $X_c, Y_c$ ), en relación a un sistema de referencia global  $(X, Y)$  como se ilustra en la figura 63.

<sup>78</sup> Castillo Oscar, Duran Eyner. Localización de un robot móvil de configuración Ackerman usando el filtro extendido de kalman. p. 2.

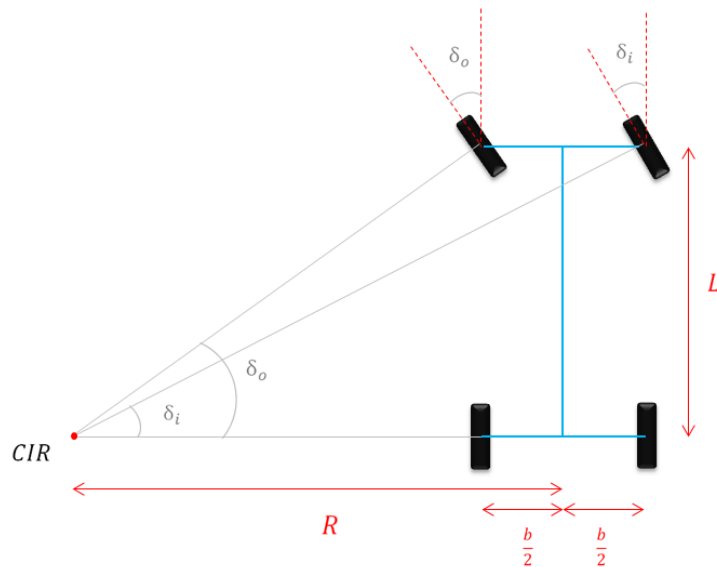
**Figura 63. Configuración Ackerman**



**Fuente:** Elaboración propia

A partir de la geometría es posible observar el CIR (Centro Instantáneo de Rotación) ilustrado en la (figura 64), que es el punto donde se encuentran las perpendiculares de todas las ruedas y por el cual el robot gira alrededor en un instante determinado

**Figura 64. Geometría básica del modelo Ackerman**



**Fuente:** Elaboración propia

A partir de este modelo se obtienen las siguientes relaciones geométricas representadas en la ecuación (9) y (10), las cuales describen el valor de la dirección de las ruedas delanteras.

$$\tan \delta_o = \frac{L}{R - \frac{b}{2}} \quad (9)$$

$$\tan \delta_i = \frac{L}{R + \frac{b}{2}} \quad (10)$$

Al igualar el radio de curvatura  $R$  en la ecuación (9) y (10) se obtiene la siguiente expresión:

$$\frac{L}{\tan \delta_i} - \frac{b}{2} = \frac{L}{\tan \delta_o} + \frac{b}{2} \quad (11)$$

Las ecuaciones (12) y (13) representan el despeje correspondiente, para obtener posteriormente la ecuación (14), la cual relaciona la dirección entre las ruedas delanteras.

$$\frac{L}{\tan \delta_i} - \frac{L}{\tan \delta_o} = b \quad (12)$$

$$L(\cot \delta_i - \cot \delta_o) = b \quad (13)$$

$$\cot \delta_i - \cot \delta_o = \frac{b}{L} \quad (14)$$

#### 4.1.1 Modelo cinemático “bicicleta”

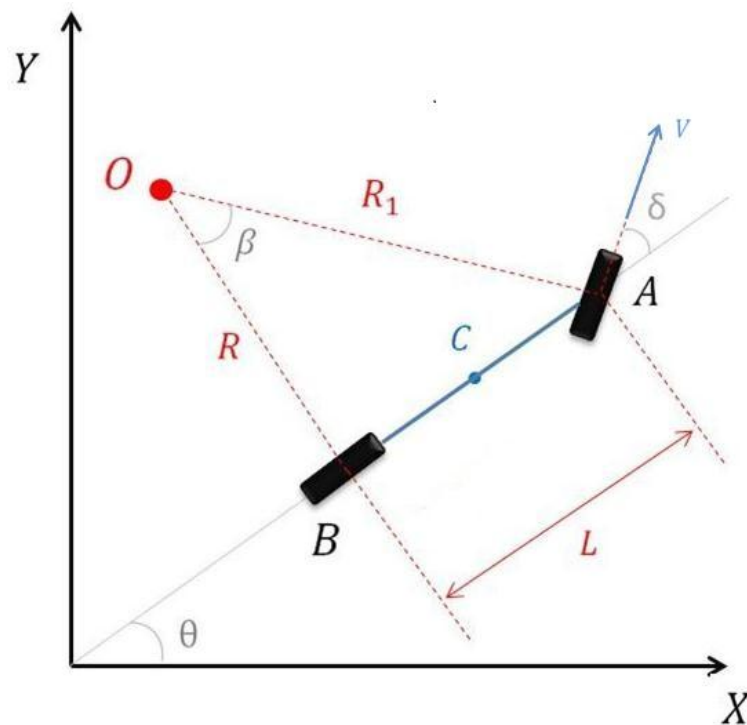
Con el objeto de reducir el modelo matemático y garantizar una correcta simulación, “una simplificación de este modelo comúnmente usado en el



*seguimiento de trayectorias es el modelo de bicicleta*<sup>79</sup>. Para este modelo (figura 65) se considera que se pueden reemplazar las cuatro ruedas del vehículo por solo dos, una rueda delantera representada por el punto  $A$  y una rueda trasera denotada por el punto  $B$ , los ángulos de dirección de las ruedas delanteras se sustituyen a un único ángulo  $\delta$  y se conserva el punto de centro de gravedad del vehículo representado por el punto  $C$ , de igual forma la separación longitudinal entre la ruedas es representado por  $L$ . Este modelo reduce las relaciones geométricas y “*aproxima las características del modelo Ackerman a velocidades razonablemente bajas con ángulos de dirección moderados*”<sup>80</sup>.

La velocidad del vehículo es representada por  $V$  y genera un ángulo  $\beta$  respecto al eje longitudinal del vehículo, el cual es conocido en la literatura como ángulo de deslizamiento. De igual forma existe un segundo ángulo el cual es resultado de la diferencia entre el ángulo de dirección de apuntamiento de la rueda y el valor real de dirección de giro de la rueda ( $\beta - \delta$ ) definido como ángulo de patinaje. Este análisis aplica en casos en que los vectores de velocidad de apuntamiento de la rueda no coinciden con el valor real de giro<sup>81</sup>.

**Figura 65.** Modelo cinemático tipo Bicicleta



**Fuente:** Elaboración propia

<sup>79</sup> Ibid., p.2

<sup>80</sup> Ibid., p.2

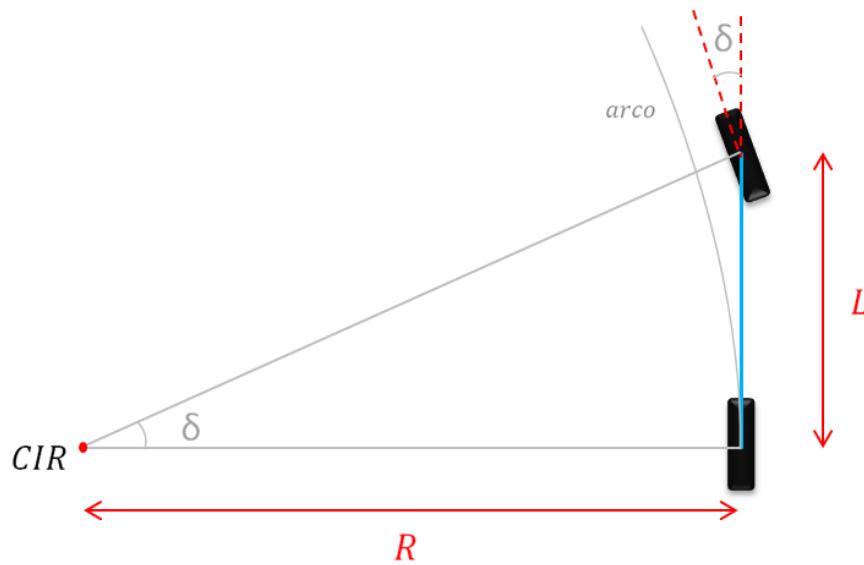
<sup>81</sup> Beal, Craig E. Applications of Model Predictive Control to Vehicle Dynamics for Active Safety and Stability. 2011. p. 38-46.

A partir de dicho análisis y considerando que el modelo propuesto se ejecuta a velocidades bajas (inferiores a 1 m/s), de acuerdo a lo expuesto por Rajamani<sup>82</sup> es válido realizar una simplificación, en la cual se asume que los vectores de velocidad en A y B coinciden con la orientación de las ruedas delantera y trasera respectivamente, de tal manera que el ángulo de patinaje se considera nulo, definiendo entonces que el ángulo de deslizamiento será igual al ángulo de dirección del vehículo tal como se expresa en la ecuación (15).

$$\beta = \delta \quad (15)$$

De esta manera es posible observar un modelo simplificado como se ilustra en la (figura 66), que de igual manera describe el CIR (Centro Instantáneo de Rotación) del robot teniendo en cuenta las reducciones previamente mencionadas.

**Figura 66.** Geometría básica del modelo Bicicleta



**Fuente:** Elaboración propia

Con el modelo ahora se puede obtener la relación geométrica expresada en la ecuación (16):

$$\text{Tan}_{\delta} = \frac{L}{R} \quad (16)$$

<sup>82</sup> Rajamani, Rajesh. Vehicle Dynamics and Control. Springer. 2006. p. 64-66.

#### 4.1.1.1 Ecuaciones características del sistema

Para definir las ecuaciones características del sistema del modelo de bicicleta es importante tener presente la nomenclatura expresada en la (tabla 4).

*Tabla 4. Nomenclatura del sistema*

Nomenclatura	
Simbología	Descripción
$\theta$	Orientación global del vehículo
$\delta$	Ángulo de dirección
$L$	Separación longitudinal entre las ruedas
$(X, Y)$	Coordenadas del sistema de referencia global
$(X_c, Y_c)$	Coordenadas de centro de gravedad del vehículo
$(x_f, y_f)$	Coordenadas globales - rueda delantera
$(x, y)$	Coordenadas globales - rueda trasera
$V$	Velocidad lineal del vehículo

*Fuente: Elaboración propia*

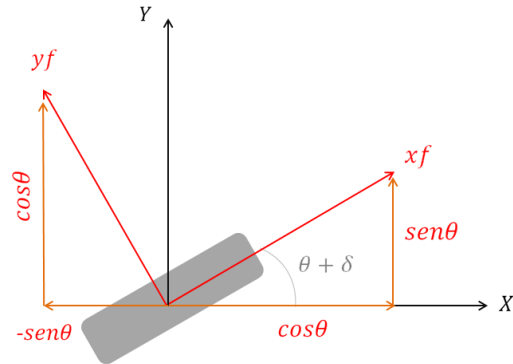
Las ruedas de un robot son un elemento de gran importancia ya que proporcionan la capacidad de movilidad. Para determinar las ecuaciones características tanto de la rueda delantera como de la trasera es necesario tener en cuenta, que el sistema tiene restricciones no holonómicas, es decir, que el robot móvil no puede desplazarse lateralmente; “*un sistema holonómico, controla cada uno de sus grados de libertad de forma independiente*”<sup>83</sup>.

De tal manera que las ecuaciones de restricción no holonómicas están orientadas a la limitación de movilidad de la rueda lateralmente (eje  $y$ ). La (figura 67) representa en el plano cartesiano las coordenadas de las ruedas del vehículo, el primero de ellos (figura 67a) muestra las coordenadas para la rueda delantera, en

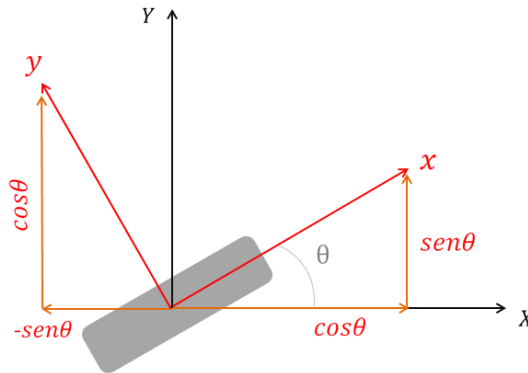
<sup>83</sup> G. Ejarque, R. Haarth, M. Diestefano. Mecanismo de control de rotación de una rueda esférica. Facultad de ingeniería. Capítulo 2.

la que su rotación es afectada tanto por la orientación del vehículo  $\theta$  como por la propia dirección de la rueda  $\delta$  y el otro esquema (figura 67b) ilustra las coordenadas de la rueda trasera en la que su rotación es afectada únicamente por la orientación del vehículo.

**Figura 67.** Coordenadas de rueda delantera y trasera



a. Rueda delantera



b. Rueda trasera

**Fuente:** Elaboración propia

De acuerdo a las coordenadas planteadas respecto al sistema global de referencia y junto a los desarrollos de A. De Luca<sup>84</sup>, las ecuaciones (17) y (18) representan las ecuaciones de restricción para las ruedas delantera y trasera respectivamente.

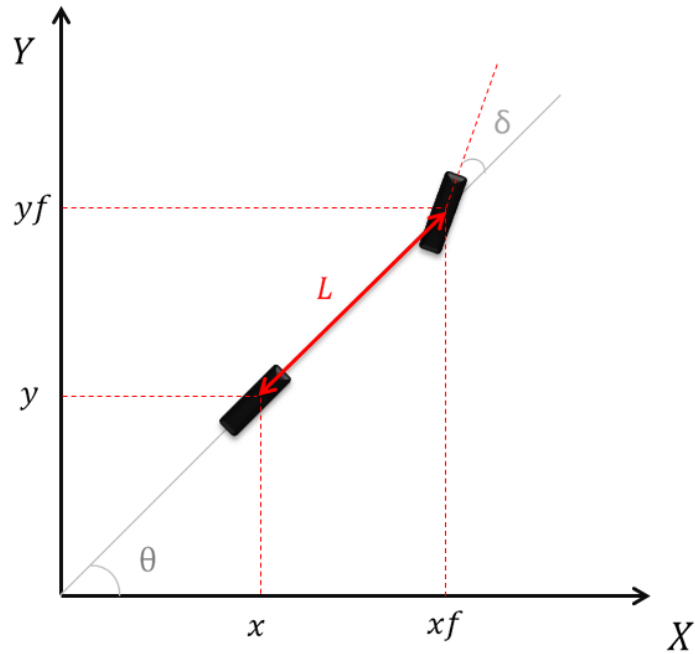
$$\dot{x}_f \text{sen}(\theta + \delta) - \dot{y}_f \text{cos}(\theta + \delta) = 0 \quad (17)$$

$$\dot{x} \text{sen}(\theta) - \dot{y} \text{cos}(\theta) = 0 \quad (18)$$

<sup>84</sup> Alessandro De Luca, Giuseppe Oriolo. Feedback Control of a Nonholonomic Car-Like Robot. Universita di roma "La Sapienza". INRIA, Sophia-Antipolis. p. 11-14.

De esta forma es posible representar las coordenadas de las ruedas del modelo tipo bicicleta respecto al sistema global de referencia tal como se muestra en la (figura 68).

**Figura 68.** Coordenadas de las ruedas para el modelo tipo bicicleta



**Fuente:** Elaboración propia

Con el objeto de desarrollar la ecuación (17), las coordenadas globales para la rueda delantera,  $(x_f, y_f)$  pueden ser expresadas como se muestra en la ecuación (19) y (20) respectivamente.

$$x_f = x + L \cos(\theta) \quad (19)$$

$$y_f = y + L \sin(\theta) \quad (20)$$

Reemplazando  $x_f$  y  $y_f$  en la ecuación (17) tenemos que:

$$\frac{d(x + L \cos(\theta))}{dt} \sin(\theta + \delta) - \frac{d(y + L \sin(\theta))}{dt} \cos(\theta + \delta) = 0 \quad (21)$$

Los términos pueden ser expresados de otro modo, tal como se expone en la ecuación (22).

$$(\dot{x} - \dot{\theta}L\text{sen}(\theta)) \text{sen}(\theta + \delta) - (\dot{y} + \dot{\theta}L\text{cos}(\theta)) \text{cos}(\theta + \delta) = 0 \quad (22)$$

Resolviendo se obtiene la ecuación (23).

$$\dot{x}\text{sen}(\theta + \delta) - \dot{\theta}L\text{sen}(\theta)\text{sen}(\theta + \delta) - \dot{y}\text{cos}(\theta + \delta) - \dot{\theta}L\text{cos}(\theta)\text{cos}(\theta + \delta) = 0 \quad (24)$$

Por medio del uso de razones trigonométricas, se reemplazan los términos correspondientes, tal como se muestra en la ecuación (25).

$$\begin{aligned} \dot{x}\text{sen}(\theta + \delta) - \dot{y}\text{cos}(\theta + \delta) - \dot{\theta}L\text{sen}(\theta)(\text{sen}(\theta)\text{cos}(\delta) + \text{cos}(\theta)\text{sen}(\delta)) \\ - \dot{\theta}L\text{cos}(\theta)(\text{cos}(\theta)\text{cos}(\delta) - \text{sen}(\theta)\text{sen}(\delta)) = 0 \end{aligned} \quad (25)$$

Resolviendo se obtiene la ecuación (26).

$$\begin{aligned} \dot{x}\text{sen}(\theta + \delta) - \dot{y}\text{cos}(\theta + \delta) - \dot{\theta}L\text{sen}^2(\theta)\text{cos}(\delta) - \dot{\theta}L\text{sen}(\theta)\text{cos}(\theta)\text{sen}(\delta) \\ - \dot{\theta}L\text{cos}^2(\theta)\text{cos}(\delta) + \dot{\theta}L\text{cos}(\theta)\text{sen}(\theta)\text{sen}(\delta) = 0 \end{aligned} \quad (26)$$

Cancelando los términos correspondientes y simplificando, se obtiene la ecuación (27) y (28):

$$\dot{x}\text{sen}(\theta + \delta) - \dot{y}\text{cos}(\theta + \delta) - \dot{\theta}L\text{sen}^2(\theta)\text{cos}(\delta) - \dot{\theta}L\text{cos}^2(\theta)\text{cos}(\delta) = 0 \quad (27)$$

$$\dot{x}\text{sen}(\theta + \delta) - \dot{y}\text{cos}(\theta + \delta) - \dot{\theta}L\text{cos}(\delta)(\text{sen}^2(\theta) + \text{cos}^2(\theta)) = 0 \quad (28)$$

Teniendo en cuenta que  $\text{sen}^2(\theta) + \text{cos}^2(\theta) = 1$ , se realiza la operación correspondiente obteniendo de esa manera la ecuación (29):

$$\dot{x}\text{sen}(\theta + \delta) - \dot{y}\text{cos}(\theta + \delta) - \dot{\theta}L \text{cos}(\delta) = 0 \quad (29)$$

La restricción no holonómica de la ecuación (18) “se satisface con  $\dot{x} = v \text{cos}(\theta)$  y  $\dot{y} = v \text{sen}(\theta)$  y cualquier múltiplo escalar del mismo”<sup>85</sup>. Este escalar corresponde a la velocidad longitudinal del vehículo  $v$  por tanto las ecuaciones (30) y (31) expresan la relación de la velocidad en  $\dot{x}$  y  $\dot{y}$  respectivamente:

$$\dot{x} = v \text{cos}(\theta) \quad (30)$$

$$\dot{y} = v \text{sen}(\theta) \quad (31)$$

Despejando  $\dot{\theta}$  de la ecuación (29) se obtiene la ecuación (32):

$$\dot{\theta} = \frac{\dot{x}\text{sen}(\theta + \delta) - \dot{y}\text{cos}(\theta + \delta)}{L \text{cos}(\delta)} \quad (32)$$

La ecuación (33) se obtiene reemplazando las ecuaciones (30) y (31) en (32).

$$\dot{\theta} = \frac{v \text{cos}(\theta) \text{sen}(\theta + \delta) - v \text{sen}(\theta) \text{cos}(\theta + \delta)}{L \text{cos}(\delta)} \quad (33)$$

---

<sup>85</sup> Cardenas Rueda, Miguel Ángel. Localização Híbrida para Um Veículo Autônomo em Escala Usando Fusão de Sensores. Campinas, 2012. p. 24.

Por medio del uso de razones trigonométricas se reemplazan los términos correspondientes, obteniendo de esta manera la ecuación (34):

$$\dot{\theta} = \frac{v \cos(\theta)(\sin(\theta) \cos(\delta) + \cos(\theta) \sin(\delta)) - v \sin(\theta)(\cos(\theta) \cos(\delta) - \sin(\theta) \sin(\delta))}{L \cos(\delta)} \quad (34)$$

Resolviendo se obtiene la ecuación (35):

$$\dot{\theta} = \frac{v \cos^2(\theta) \sin(\delta) + v \cos(\theta) \sin(\theta) \cos(\delta) + v \sin^2(\theta) \sin(\delta) - v \sin(\theta) \cos(\theta) \cos(\delta)}{L \cos(\delta)} \quad (35)$$

Cancelando los términos correspondientes y simplificando, se obtiene la ecuación (36) y (37):

$$\dot{\theta} = \frac{v \cos^2(\theta) \sin(\delta) + v \sin^2(\theta) \sin(\delta)}{L \cos(\delta)} \quad (36)$$

$$\dot{\theta} = \frac{v \sin(\delta) (\cos^2(\theta) + \sin^2(\theta))}{L \cos(\delta)} \quad (37)$$

Teniendo en cuenta que  $\sin^2(\theta) + \cos^2(\theta) = 1$ , se reemplaza el término correspondiente obteniendo de esa manera la ecuación (38):

$$\dot{\theta} = \frac{v \sin(\delta)}{L \cos(\delta)} \quad (38)$$



Expresando el resultado simplificado, la ecuación (39) expresa el valor de  $\dot{\theta}$ .

$$\dot{\theta} = \frac{v \tan(\delta)}{L} \quad (39)$$

De manera tal que las ecuaciones de movimiento que describen el modelo cinemático del vehículo están dadas por la ecuación (40):

$$\begin{bmatrix} \dot{X}_c \\ \dot{Y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \frac{v \tan(\delta)}{L} \end{bmatrix} \quad (40)$$

## 4.2 SISTEMA DE DIRECCIÓN

El sistema de dirección tiene por objeto orientar las ruedas delanteras con el fin de que el vehículo pueda desarrollar la trayectoria deseada por el conductor.

### 4.2.1 Configuración del sistema de dirección

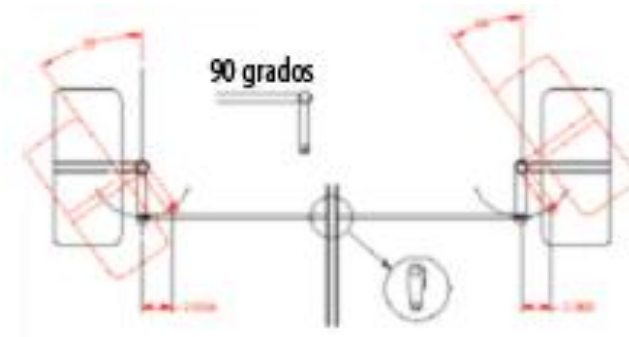
El sistema de dirección del vehículo tiene como base el modelo Ackerman, a través de la geometría de este modelo las ruedas delanteras tienen la característica de girar con ángulos diferentes. La (figura 69) presenta un boceto de dos tipos de sistemas, el primero de ellos (figura 69a) muestra un esquema con el brazo del eje en un ángulo de 90 grados y el otro esquema (figura 69b), siguiendo el modelo Ackerman, con el brazo del eje con un ángulo de 105 grados.

*“El modelo de 90 grados hace que las dos ruedas delanteras giren paralelamente con un mismo ángulo y su principal desventaja es la pérdida de desempeño y deslizamiento en el momento de la curva”<sup>86</sup>. Caso contrario en el modelo de Ackerman, que garantiza que el ángulo de dirección de las ruedas delanteras sean diferentes en el momento de realizar la curva como se muestra en la (figura 69b).*

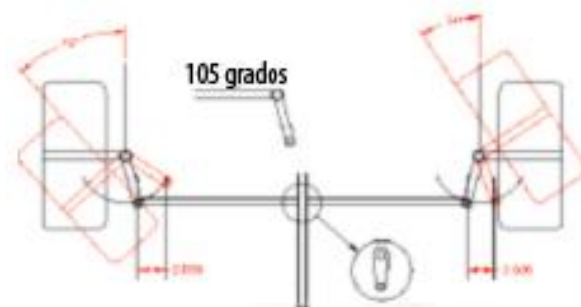
---

<sup>86</sup> Hernández Rubén. Op. cit., p. 408.

**Figura 69.** Configuración del sistema de dirección



a. Dirección convencional



b. Dirección ackerman

**Fuente:** Hernández Rubén, Bernárdez Giovani, Vaqueiro Janito, Siqueira Pablo. *Planeación y seguimiento de trayectorias de un vehículo terrestre con base en el control de dirección en un ambiente real*, 2014. p. 407

#### 4.2.2 Restricción del modelo

Dada la configuración del vehículo, existe una limitante mecánica en las ruedas delanteras al momento de girar, por tal es necesario tener en cuenta el valor del ángulo que puede generar la rueda al realizar los giros de izquierda o derecha, con el fin de que los resultados de la simulación no demuestran comportamientos irreales y correspondan a los esperados. De acuerdo a mediciones realizadas sobre el vehículo, se asume que el valor máximo de giro de la rueda es 30° (grados) respecto a su eje central.

Por tal el valor de dirección estará dentro del rango que se expresa en la ecuación (41)

$$-\delta_{max} < \delta < \delta_{max} \quad (41)$$

Siendo  $\delta_{max} = 30$

La (figura 70) presenta la configuración de dirección del modelo implementado, el primero de ellos (figura 70a) presenta la disposición de las ruedas totalmente alineadas formando así un ángulo de 0 grados de giro, seguidamente se presenta la configuración para la dirección orientada hacia el lado derecho (figura 70b) en el que el valor máximo de giro ( $-\delta_{max}$ ) se toma respecto a la rueda derecha y en el otro esquema (figura 70c) se expone el giro de las llantas orientado hacia el lado izquierdo para el cual el valor máximo de giro ( $\delta_{max}$ ) se toma respecto a la rueda izquierda.

**Figura 70.** Restricción angular del modelo



a. Dirección centrada "0 grados"



b. Dirección (giro derecha) "-30 grados"



c. Dirección (giro izquierda) "30 grados"

**Fuente:** Elaboración propia

### 4.3 VALIDACIÓN DEL MODELO CINEMÁTICO EN MATLAB®

Con el fin de corroborar los desarrollos matemáticos previamente realizados y con el objetivo de simular el comportamiento del vehículo en función del tiempo, se realiza la implementación de las ecuaciones características del sistema en un software de simulación.

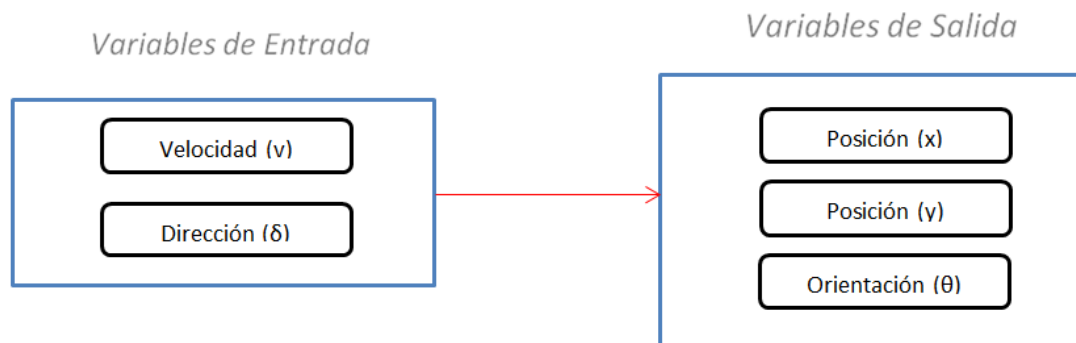
El programa seleccionado para dicha implementación fue el software de Matlab®, dado que es un software de desarrollo matemático el cual integra diferentes aplicaciones en el área de la ingeniería, permitiendo la implementación de algoritmos y la visualización gráfica de los mismos<sup>87</sup>.

De igual forma con el objetivo de analizar las variables y las trayectorias realizadas por el vehículo se desarrolla la implementación del modelo cinemático en el entorno de programación visual Simulink/ Matlab®, dado a que es una herramienta de simulación que permite visualizar el comportamiento de modelos basados en la programación de diagrama de bloques<sup>88</sup>.

#### 4.3.1 Implementación

La (figura 71) expone las variables a tener en cuenta para la implementación del modelo en la simulación, en primera instancia se definen las variables de entrada, que son la velocidad y dirección, estas permiten que el vehículo genere un desplazamiento con una dirección determinada, por otro lado, se evalúan las variables de salida, que son la posición resultante del vehículo en el espacio  $(x, y)$  y la orientación de acuerdo al sistema coordenado del espacio de trabajo.

**Figura 71.** Variables del Modelo



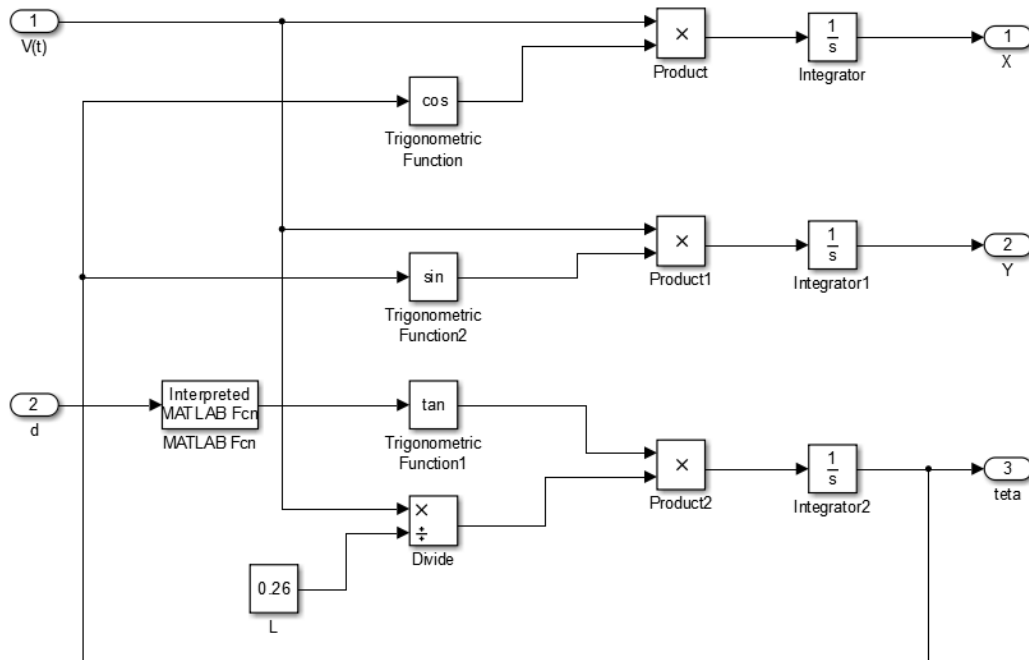
**Fuente:** Elaboración propia

<sup>87</sup> MathWorks. The Language of Technical Computing. [en línea] <<http://www.mathworks.com/products/matlab/>>

<sup>88</sup> MathWorks. Simulation and Model-Based Design. [en línea] <<http://www.mathworks.com/products/simulink/>>

Teniendo presente las variables de entrada y salida del sistema, es posible implementar la ecuación (40) en Simulink/Matlab®, tal como se puede observar en la (figura 72).

**Figura 72.** Implementación del modelo cinemático



**Fuente:** Elaboración propia

El diagrama está compuesto por un bloque con designación "L", el cual representa la separación longitudinal entre las ruedas y su valor constante está dado en metros "m".

En la implementación del modelo se tuvo en cuenta la restricción de giro de las ruedas delanteras, para ello se implementó la función "grados", la cual tiene por objeto validar el rango de restricción del ángulo, con el fin de que la simulación aproxime los resultados acordes con las limitaciones físicas de giro de las ruedas delanteras del vehículo (anteriormente explicadas), es decir, que esta función realiza la corrección en caso de que se ingrese al sistema valores no pertenecientes al rango determinado. Esta función lee el valor de dirección deseada y evalúa tres posibles situaciones, en las que si el valor está dentro de los rangos devuelve el mismo parámetro pero si esta fuera de los mismos entrega al sistema el valor máximo permitido.

A continuación se muestra el pseudocódigo correspondiente a la función grados

**Programa:** grados

**Entrada:** u

**Salida:** y

**Variables:** angulomax: racional  
angulomin: racional

**Algoritmo**

```
    Leer u
    SI u > angulomax ENTONCES
        y = angulomax
    SI NO
        SI u < angulomin ENTONCES
            y = angulomin
        SI NO
            y = u
        FIN SI
    FIN SI
Finprograma
```

#### 4.3.2 Parámetros del sistema

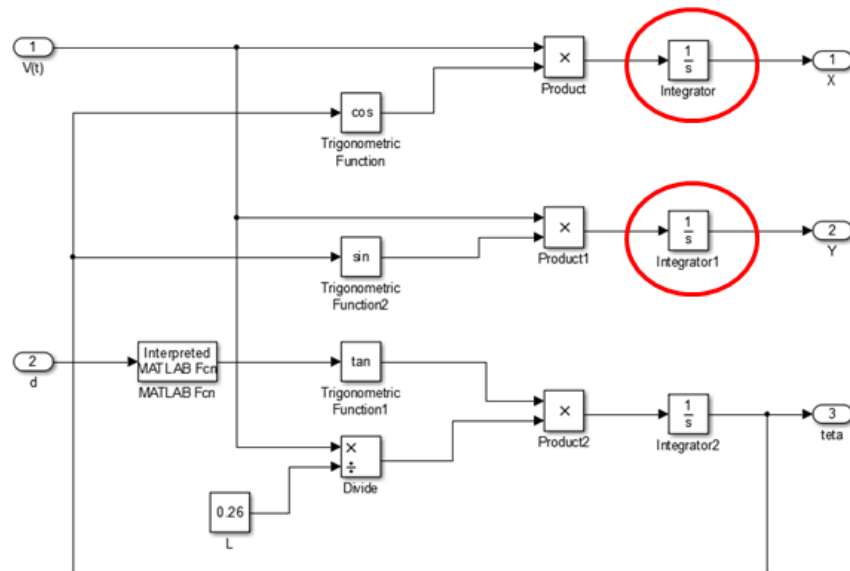
Es importante tener en cuenta que para entender el funcionamiento del modelo y las simulaciones, es necesario saber que de qué manera se determinaron las condiciones del sistema para posteriormente validar los resultados.

Los parámetros iniciales de la simulación, se ajustan de acuerdo a las condiciones que se requiera. Por tal existen dos condiciones esenciales que ubican el vehículo dentro del espacio de trabajo. La primera de ellas es la ubicación en el espacio coordinado  $(x, y)$  y la segunda la orientación del vehículo  $(\theta)$ .

##### 4.3.2.1 Posición y orientación inicial del vehículo en el plano

La posición inicial del vehículo en el espacio de trabajo se determina a partir de evaluar las condiciones de las variables  $(X)$  y  $(Y)$ , para ello, en el modelo el diagrama de bloques implementado, se modifican los estados de los bloques integradores denotados en la (figura 73), de manera tal que el valor numérico ingresado en cada bloque corresponderá a la coordenada inicial  $(X_c, Y_c)$  del vehículo respecto al sistema global de referencia.

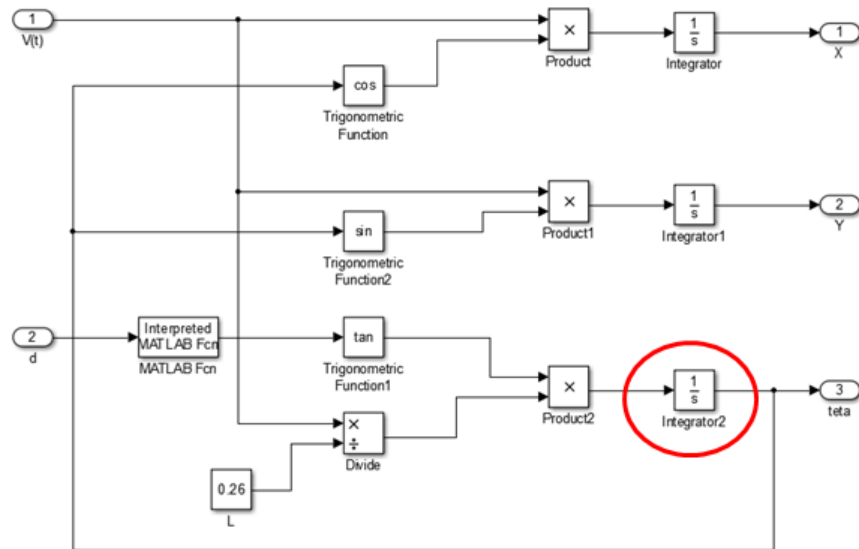
**Figura 73. Bloques integradores (posición inicial del vehículo)**



**Fuente:** Elaboración propia

Esta consideración de igual forma aplica para la orientación del vehículo. Para ello, se modifica el diagrama de bloques denotado en color rojo como se observa en la (figura 74). El valor se ingresa al sistema en unidades de *radián*.

**Figura 74. Bloque integrador (orientación inicial del vehículo)**

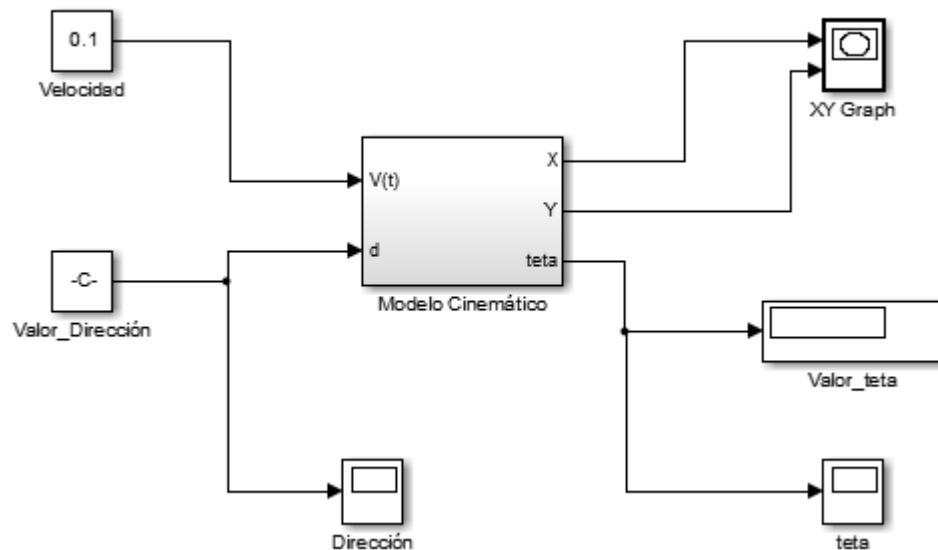


**Fuente:** Elaboración propia

### 4.3.3 Verificación del modelo

La validación del modelo cinemático (figura 75), se realizó implementando las ecuaciones características en un subsistema, seguidamente se asumieron los parámetros de velocidad y valor constante que tendrá la dirección del vehículo. De igual forma para poder observar la trayectoria descrita por el vehículo se caracterizan en una ventana de graficación los datos  $(X, Y)$  y se visualiza el valor de orientación.

**Figura 75.** Diagrama de bloques del modelo cinemático



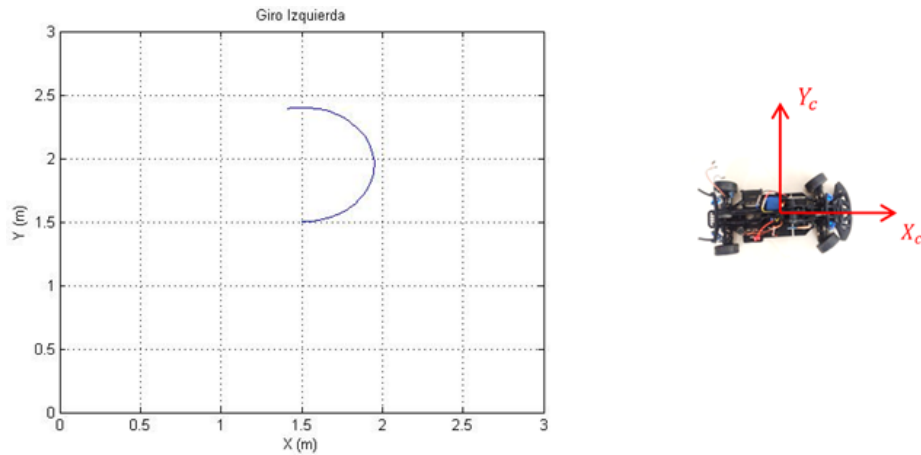
**Fuente:** Elaboración propia

Para validar los desplazamientos básicos del vehículo de acuerdo al modelo cinemático del mismo, se realizan una serie de simulaciones asumiendo un tiempo de ejecución y velocidad. Los valores de dirección son modificados en cada simulación con el fin de observar la trayectoria que describe el vehículo, verificando que las restricciones de dirección y desplazamiento sean correctas.

La primera simulación (figura 76) muestra la trayectoria que describe el vehículo en una condición inicial de  $(X = 1.5, Y = 1.5, \theta = 0^\circ)$  y se asume el valor constante de dirección  $\delta = 30$ , el vehículo gira en sentido contrario a las manecillas del reloj describiendo una trayectoria en forma de circunferencia.



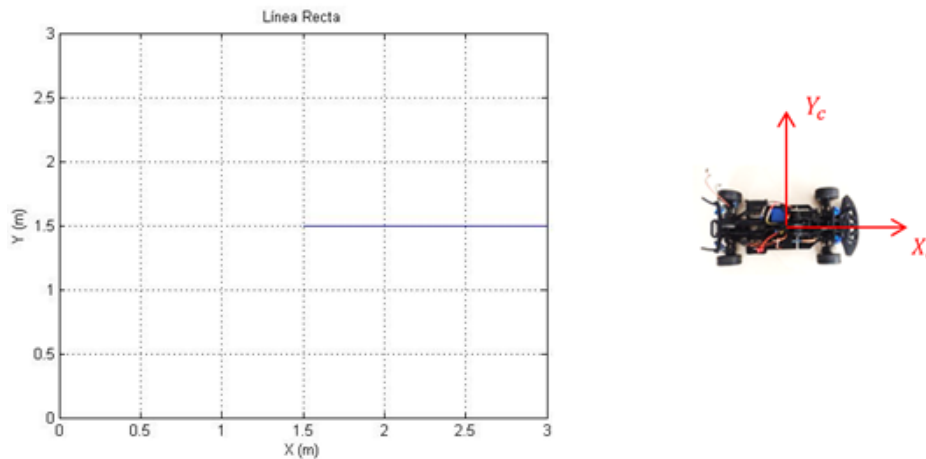
**Figura 76. Simulación No.1 (Giro izquierda)**



**Fuente:** Elaboración propia

La segunda simulación (figura 77) muestra la trayectoria que describe el vehículo en una condición inicial de  $(X = 1.5, Y = 1.5, \theta = 0^\circ)$  y se asume el valor constante de dirección  $\delta = 0$ , el vehículo se desplaza en línea recta.

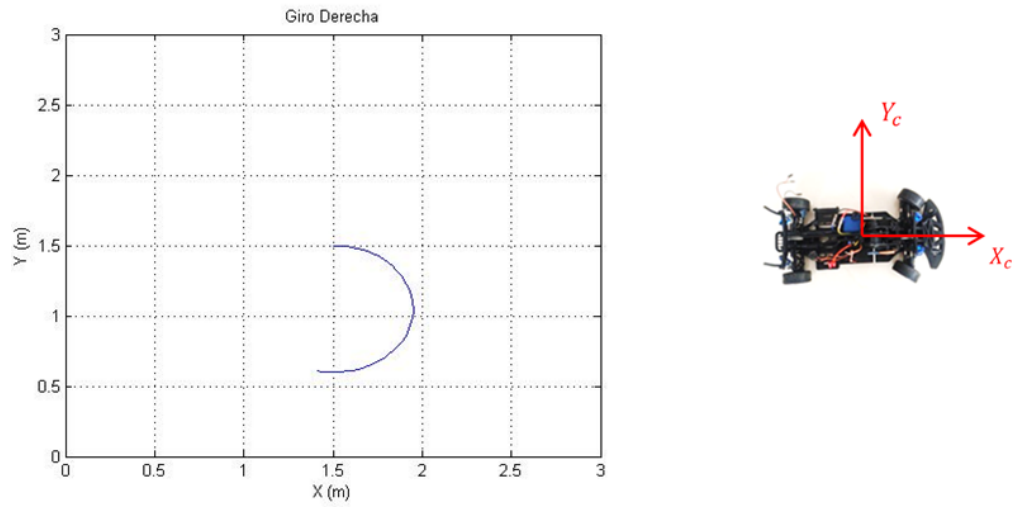
**Figura 77. Simulación No.2 (Línea Recta)**



**Fuente:** Elaboración propia

La tercera simulación (figura 78) muestra la trayectoria que describe el vehículo en una condición inicial de  $(X = 1.5, Y = 1.5, \theta = 0^\circ)$  y se asume el valor constante de dirección  $\delta = -30$ , el vehículo gira en sentido de las manecillas del reloj describiendo una trayectoria en forma de circunferencia.

**Figura 78. Simulación No.3 (Giro Derecha)**



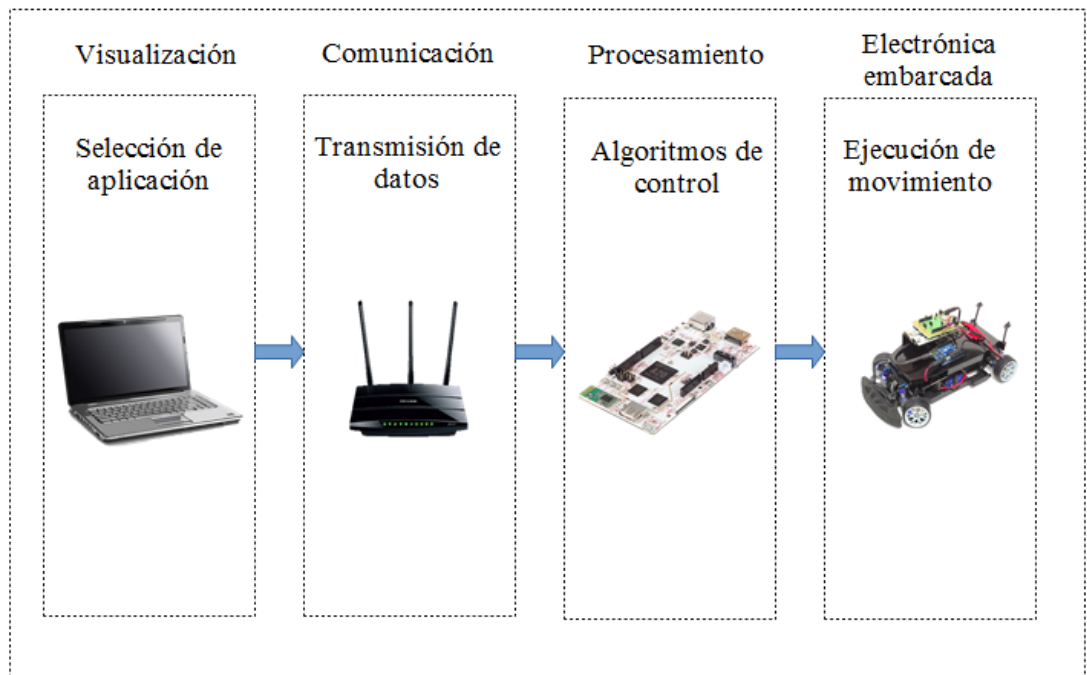
**Fuente:** *Elaboración propia*

## 5 INGENIERÍA DEL PROYECTO

### 5.1 ARQUITECTURA DEL SISTEMA

El funcionamiento del sistema se desarrolla mediante la interacción de una serie de etapas las cuales permiten la comunicación de los dispositivos y ejecución de las aplicaciones propuestas. El sistema está compuesto por cuatro etapas principales, que se componen por la visualización, comunicación, procesamiento y electrónica embarcada, tal como se observa en la (figura 79).

*Figura 79. Arquitectura del sistema*



*Fuente. Elaboración propia*




El ordenador permite al usuario visualizar la información en tiempo real de las variables captadas y procesadas por el sistema embebido (cámaras, sensores etc.), la comunicación entre estos dos dispositivos (computador y sistema de procesamiento) se logra mediante la implementación de un Router el cual genera un punto de red WiFi y permite la transmisión de datos.

De acuerdo a la información del sistema de procesamiento, se desarrollan las acciones de actuación del vehículo, tal como lo son el seguimiento de trayectorias en el ambiente predeterminado o de igual manera la identificación de peatones y líneas de la vía.

## 5.2 SELECCIÓN DE PLATAFORMA

Para la elaboración del proyecto se planteó la necesidad de implementar una plataforma a escala con características similares a un vehículo convencional real. Teniendo en cuenta las restricciones mecánicas de la dirección del vehículo, sistema de tracción diferencial y la no holonomicidad para generar la ejecución de los movimientos; de igual forma, la posibilidad de aplicar los métodos de control necesarios. Además se busca que el vehículo sea eléctrico para que sea adaptable a las tecnologías automotoras actuales, para esto se tuvo en cuenta diversas opciones como se observa en la tabla 5.

*Tabla 5. Características de las plataformas*

CARACTERÍSTICAS	<b>SNOW PANTER NISSAN SKYLINE ELÉCTRICO</b> 	<b>1/18 RUCKUS 4WD MONSTER TRUCK RTR</b> 	<b>CARRO RC 1/5 GASOLINA</b> 
TIPO DE ALIMENTACIÓN	Eléctrico	Eléctrico	Gasolina
ESCALA	1:10	1:18	1:5
SPEED CONTROL	Si	Si	No
TIPO DE TRACCIÓN	4x4	4x4	4x2
PRECIO	360.000 \$	619.000\$	1.500.000 \$

*Fuente. Elaboración propia*

De la tabla se puede concluir que la mejor opción es la plataforma Snow panter Nissan skyline eléctrico, debido a que el tipo de alimentación es eléctrica lo cual infiere el uso de baterías que permite una sencilla manipulación a comparación de los vehículos a gasolina, de igual manera el tipo de alimentación y la implementación de un speed control es una gran ventaja a la hora de ejercer un control de velocidad. La escala de la plataforma es 1:10 a diferencia de los otros vehículos, esto dará una gran ventaja a la hora de ejecutar los movimientos en las dimensiones del escenario implementado, finalmente el tipo de tracción 4x4 con el propósito de que la plataforma tuviera características similares a las de un vehículo urbano.

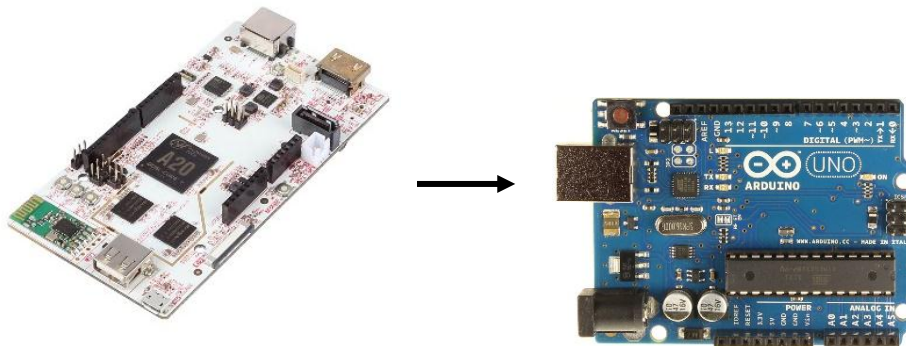
### 5.3 ELECTRÓNICA EMBARCADA

Todo robot móvil parte de unas necesidades de procesamiento y detección acordes a la tarea a desempeñar, de ahí surge la integración de dispositivos para satisfacer estas necesidades; el sistema de control embarcado en el vehículo permite el seguimiento de trayectorias, la lectura de variables físicas es ejecutada por los sensores seleccionados para la medición de velocidad y distancias entre el vehículo respecto a un objeto u obstáculo; dichos componentes conforman la electrónica embarcada.

#### 5.3.1 Sistema electrónico de procesamiento

Para el sistema de procesamiento se selecciona una interacción entre un PcDuino v3 el cual se encarga del procesamiento de imagen necesario para el sistema de localización y un Arduino Uno el cual responde a los comandos del PcDuino para ejecutar los movimientos de acuerdo al control efectuado por el mismo, haciendo que un controlador sea el maestro y otro el esclavo como se muestra en la (figura 80).

*Figura 80. Integración de control*



*Fuente. Elaboración propia*

El pcDuino es un sistema embebido óptimo para el procesamiento y control en tiempo real, este es compatible con los sistemas operativos Ubuntu y Android ICS, al tener la capacidad de manejar los sistemas operativos antes mencionados se da la posibilidad de ejecutar el procesamiento de imagen, control e interacción con el Arduino Uno, mediante el software QT Creator el cual fue implementado para el desarrollo.

La selección de este dispositivo se dio gracias a características específicas que posee: un procesador dual Core A20, 1GB de memoria RAM, comunicación de red de 150 mbps, dos puertos seriales, y una alimentación de 5V a 2000 mA; en comparación a otros sistemas embebidos que se encuentran en el mercado que ofrecen gran variedad de posibilidades como lo son: el Raspberry pi y el

Beaglebone Black, en la (tabla 6) se presenta una comparación entre las características de los sistemas anteriormente nombrados.

**Tabla 6.** Especificaciones de los sistemas embebidos

<b>CONTROLADOR</b>	<b>PCDUINO</b>	<b>RASPBERRY PI</b>	<b>BEAGLEBONE BLACK</b>
<b>CPU</b>	DUAL CORE A20	700 MHz Low Power ARM1176JZ-F	1 GHz ARM Cortex-A8
<b>GPU</b>	OpenVG 1.1, Mali 400 Dual Core	Dual Core VideoCore IV	PowerVR SGX530
<b>MEMORIA</b>	1GB	512MB SDRAM	512 MB DDR3
<b>USB</b>	USB 2.0	Conector USB X2	USB 2.0
<b>HDMI</b>	si	si	si
<b>SISTEMA OPERATIVO</b>	Ubuntu 12.04.Android	Linux	Linux, Android

**Fuente.** Elaboración propia

Como se observa en la (tabla 5), el pcDuino es superior frente a los otros controladores en aspectos como el procesamiento y la capacidad de la memoria, haciéndolo el controlador de bajo costo más óptimo para el procesamiento de imágenes, además de esto ofrece la capacidad de comunicación con el arduino Uno por puerto serial; para la configuración e instalación del pcDuino se recomienda revisar el *Anexo H*.

El usar un segundo controlador proporciona mayor capacidad de procesamiento a la hora del tratamiento de imágenes por parte del pcDuino. El arduino Uno cuenta con una velocidad de procesamiento de 12 MHz, puertos digitales y análogos para el control de los actuadores que ejecutan los movimientos propuestos por el controlador principal.

### 5.3.2 Actuadores

Los actuadores se definen como los dispositivos que ejecutan un movimiento, estos dispositivos pueden ser neumáticos, hidráulicos o eléctricos dependiendo de

su aplicación. Estos mecanismos poseen diversas características en su forma de desarrollo de la acción, dentro de la robótica los actuadores más usados son los motores y los servomotores que efectúan un movimiento mecánico y pueden ser controlados por otros dispositivos electrónicos.

### 5.3.2.1 Actuadores eléctricos

La plataforma seleccionada se compone de un motor-reductor y un servomotor los cuales desarrollan los movimientos de desplazamiento y dirección respectivamente del vehículo.

#### a. Motor-reductor

Los motor-reductores son utilizados en casos donde se busca un mayor torque dejando de lado la velocidad angular, la conversión de velocidad angular se da por una caja reductora con una configuración de piñones determinada para esta tarea, en aplicaciones de robótica existe una gran gama de motor-reductores que varían en torques, velocidad angular y tamaño: teniendo en cuenta estos parámetros el motor implementado por el fabricante para el vehículo es el Faulhaber Dc 12v Motor 2342l012cr como se ilustra en la (figura 81) este cuenta con una caja reductora que transforma 8100rpm a 120 rpm además cuenta con un torque de 1.72 N/M y unas medidas de 30 mm de diámetro y 85 mm de largo.

**Figura 81.** Motor faulhaber 12v



**Fuente.** Grab cad URL. <https://grabcad.com/library/faulhaber-dc-motor-gear-box-1>

Para el PID se modeló la planta “motor” mediante la función de transferencia característica de un motor DC dada por la ecuación (42).

$$tf = \frac{k}{(js + b)(ls + r) + k^2} \quad (42)$$

De la ecuación, “k” es igual a la fuerza contra-electromotriz medida en voltios a una velocidad constante sobre las RPM en ese instante de tiempo, “j” es el momento de inercia, “b” es la constante de fricción de coulomb, “l” la inductancia y “r” la resistencia del motor. Las anteriores variables fueron tomadas de las

especificaciones del fabricante del motor (información expuesta en el *Anexo L*), reemplazando los valores se obtiene la ecuación (43)

$$tf = \frac{2.44}{(5.7s + 6 * 10^{-5})(6.5s + 1.3k) + 5.9536} \quad (43)$$

Operando se obtiene la ecuación (44)

$$tf = \frac{2.44}{0.3705s^2 + 359.1s + 5.9536} \quad (44)$$

Se divide la ecuación (3) en 0.3705 para obtener la ecuación (45)

$$tf = \frac{6.6}{s^2 + 969.2s + 16.06} \quad (45)$$

## b. Servomotor

El segundo actuador está definido por un servomotor, que se caracteriza por el movimiento rotacional controlado, tiene la capacidad de llegar a determinado ángulo entre 0 y 180 grados generalmente, en la actualidad existen algunos servomotores con la capacidad de variar su ángulo entre 0 y 210 grados. El giro está determinado por un sistema electrónico de control que responde a una señal de PWM, esta señal es un tren de pulsos los cuales determinan el sentido de giro, para llegar a un determinado ángulo se emite un tren de pulsos con una frecuencia determinada para cada ángulo; para una posición neutral se emite una señal de 1.5 ms, para un giro en sentido horario se emiten señales menores a 0.7 ms y para sentido anti horario se emite una señal superior a 1.7 ms como se muestra en la (figura 82).

**Figura 82.** Señal para sentido de giro

Motor en sentido horario (ejemplo 0,7 ms):



Motor neutral (1,5ms):



Motor en sentido antihorario (ejemplo 1,8ms):

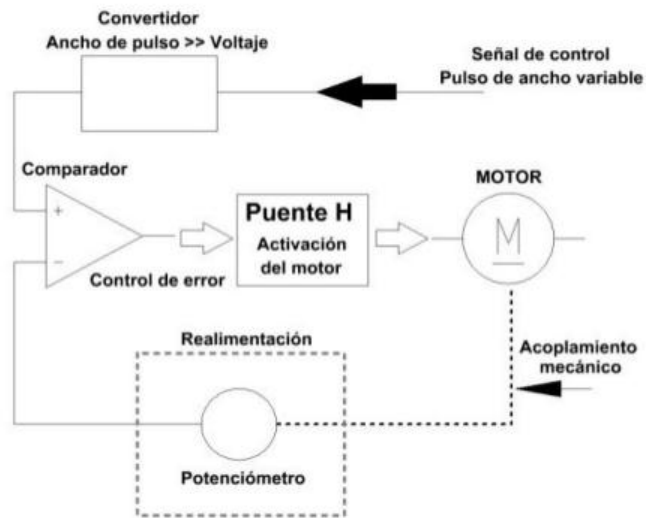


**Fuente.** Elaboración propia



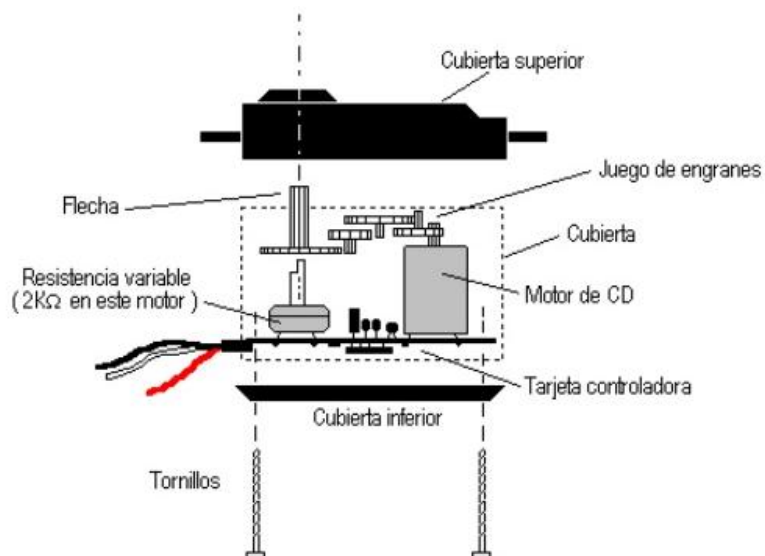
El servomotor está compuesto por un sistema interno de control como se muestra en la (figura 83) y engranajes como se muestra en la (figura 84), esto finalmente determina el giro de las llantas de acuerdo al control de dirección. Los datos técnicos del servomotor se exponen en el *Anexo J*.

**Figura 83.** Circuito interno de un servomotor



**Fuente.** Partes de un servomotor. URL <http://es.slideshare.net/CristianDavis/partes-de-un-servomotor>

**Figura 84.** Partes de un servomotor



**Fuente.** Partes de un servomotor URL. <http://es.slideshare.net/CristianDavis/partes-de-un-servomotor>

### 5.3.3 Electrónica de potencia

La electrónica de potencia juega un papel fundamental en el desarrollo de cualquier robot ya que de esta dependerá el suministro adecuado de la energía de todos los implementos a utilizar, dentro de los elementos de potencia utilizados está el Speed Control wd-1040-brushed (figura 85), el cual es incorporado por el fabricante a la plataforma (en el *Anexo K* se mencionan los datos técnicos del Speed Control).

**Figura 85.** Speed Control



**Fuente.** Brushed. URL <http://www.lelong.com.my/hobbywing-wp1040-waterproof-brushed-esc-rc-car-kokstore-154972669-2015-09-Sale-P.htm>

El Speed Control se alimenta mediante una batería de 1000 mA a 12 V, el mecanismo regula la frecuencia del motor haciendo que aumente o disminuya la velocidad, la variación de frecuencia se efectúa por medio de una señal PWM la cual es interpretada para ejecutar la variación de velocidad para cada valor del mismo. La alimentación de los sensores y sistemas de control se da por la implementación de una batería lipo, Para regular la tensión se implementa el dispositivo regulador de potencia 7805K como se ilustra en la (figura 86) el cual posee características específicas como se muestra en la tabla 7.

**Figura 86.** Regulador 7805K



**Fuente.** Elaboración propia

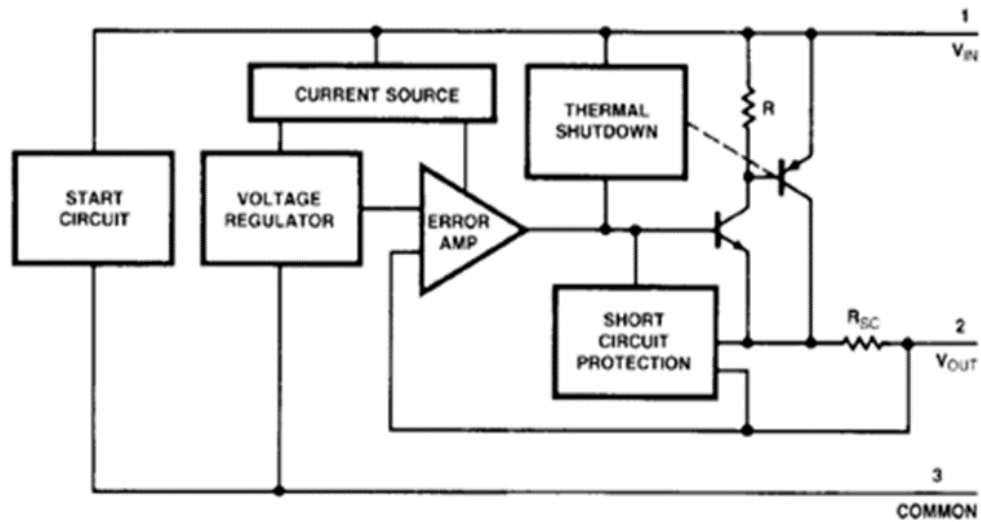
**Tabla 7.** Características eléctricas del regulador 7805k

símbolo	característica	Condición	Límites			Unidad
			Min	Typ	Max	
$V_{out}$	Voltaje salida	$I_{out}$	4.85	5.0	5.25	V
$\Delta V_{out}$	Linea regulada	$V_{in}= 8.5 - 25$		10	50	mV
	Carga regulada	$V_{in}= 7.5 - 25$		10	50	mV
$I_Q$	Coeficiente corriente	$I_{out}=0$		3.0	10	mA
$V_n$	frecuencia de salida	$10 \text{ Hz} \leq f \leq 100 \text{ kHz}$		40		$\mu V_{RMS}$
$V_{dd}$	Voltaje de descarga	$I_{out}= 5.0 \text{ A}$		2.3		V
		$I_{out}= 3.0 \text{ A}$		2		V
$I_{os}$	límite de corto circuito			7.0	12.0	$A_{pk}$

**Fuente.** All datasheet. URL. <http://pdf1.alldatasheet.com/datasheet-pdf/view/130555/FAIRCHILD/UA78H05A.html>

La (figura 87) ilustra el diagrama de bloques de la estructura del anterior regulador.

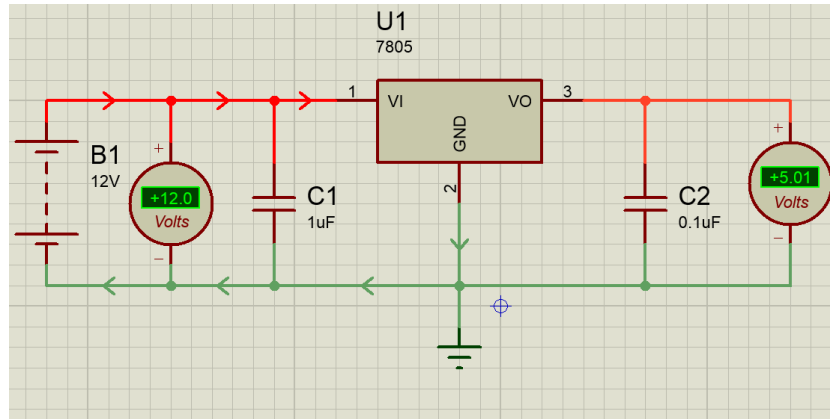
**Figura 87.** Diagrama de bloques del regulador



**Fuente.** All datasheet. URL. <http://pdf1.alldatasheet.com/datasheet-pdf/view/130555/FAIRCHILD/UA78H05A.html>

El circuito utilizado se simula previamente en el software Proteus® como se aprecia en la (figura 88); este circuito tiene la capacidad de manejar altas tensiones. Para la implementación a desarrollar se manejará 1 A, por lo cual no es necesario acoplar un dispositivo regulador de temperatura como lo sugiere el fabricante.

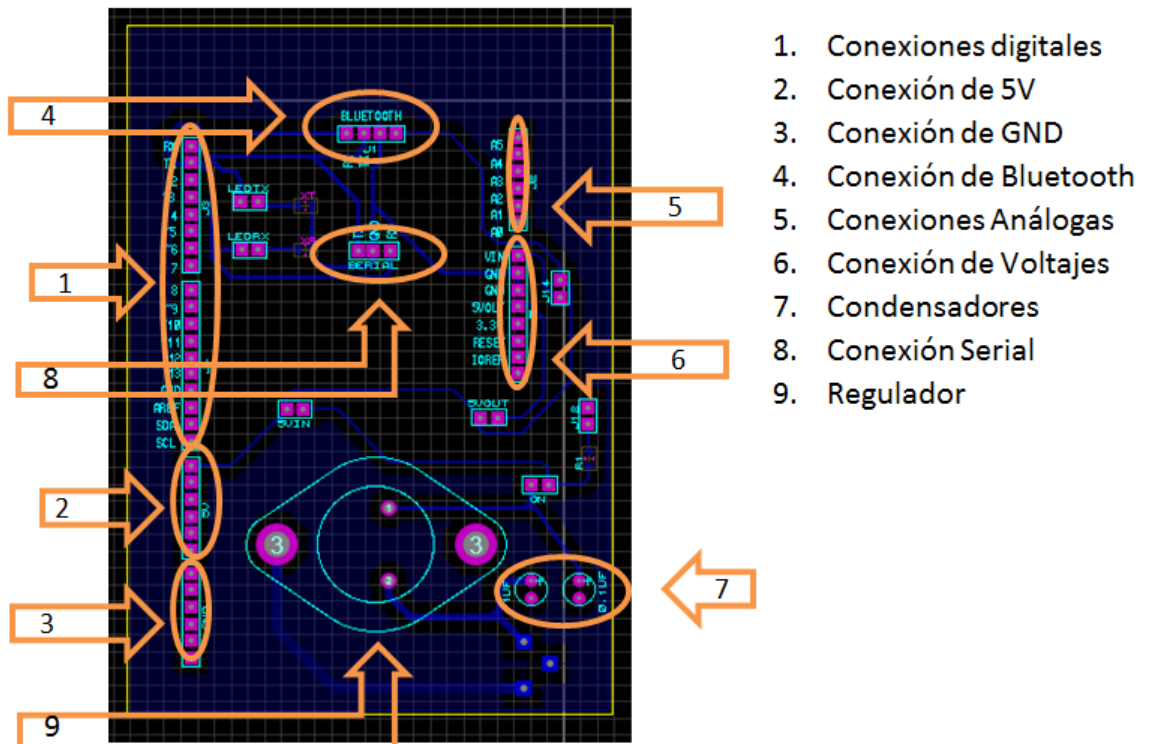
**Figura 88.** Esquemático de circuito



**Fuente.** Elaboración propia

Los condensadores C1 y C2, actúan como filtros tanto en la entrada como la salida para garantizar un mínimo cambio de voltaje en ambas mallas. Una vez simulada la regulación se procede a diseñar el circuito final como se aprecia en la (figura 89), el cual se acoplara al pcDuino y tendrá varios puntos de alimentación para los sensores ubicados en el vehículo.

**Figura 89.** Diseño de circuito impreso



**Fuente.** Elaboración Propia (diseñado en Ares de Proteus)

### 5.3.4 Sensores

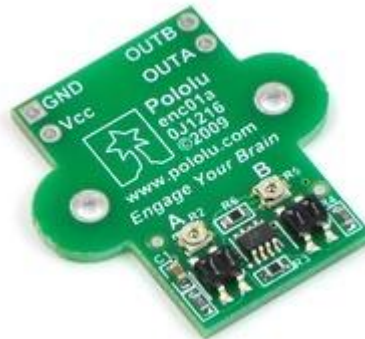
Los sensores son los dispositivos que interpretan las variables físicas del entorno de ejecución del movimiento o del propio vehículo, de esta manera se accede a la información fundamental del estado de las variables censadas a cada instante de tiempo, esto en conjunto con los dispositivos de control ejecutan una acción de corrección de acuerdo a la referencia estipulada para esta variable.

#### 5.3.4.1 Sensores de movimiento y detección

##### a. Encoder óptico Pololu

Con el fin de censar la velocidad del vehículo se implementó un encoder óptico como se ve en la (figura 90), es un sensor infrarrojo el cual funciona por medio de interrupciones, el encoder óptico implementado cuenta con dos canales (A, B) los cuales detectan el flanco de subida y de bajada dado por los colores blanco y negro, los colores blanco y negro poseen características de color opuestas, la capacidad de reflexión de luz de cada uno de ellos es diferente haciendo que los sensores ópticos detectan este cambio; el sensor tiene un voltaje mínimo de trabajo y un máximo siendo respectivamente 3.3V y 5V; la distancia óptima de medición entre el sensor y la cinta de colores es de 5mm para la lectura óptima de la señal, la histéresis del sensor son 5mm haciendo que a 1cm la señal se a intermitente o nula.

**Figura 90.** Encoder óptico Pololu



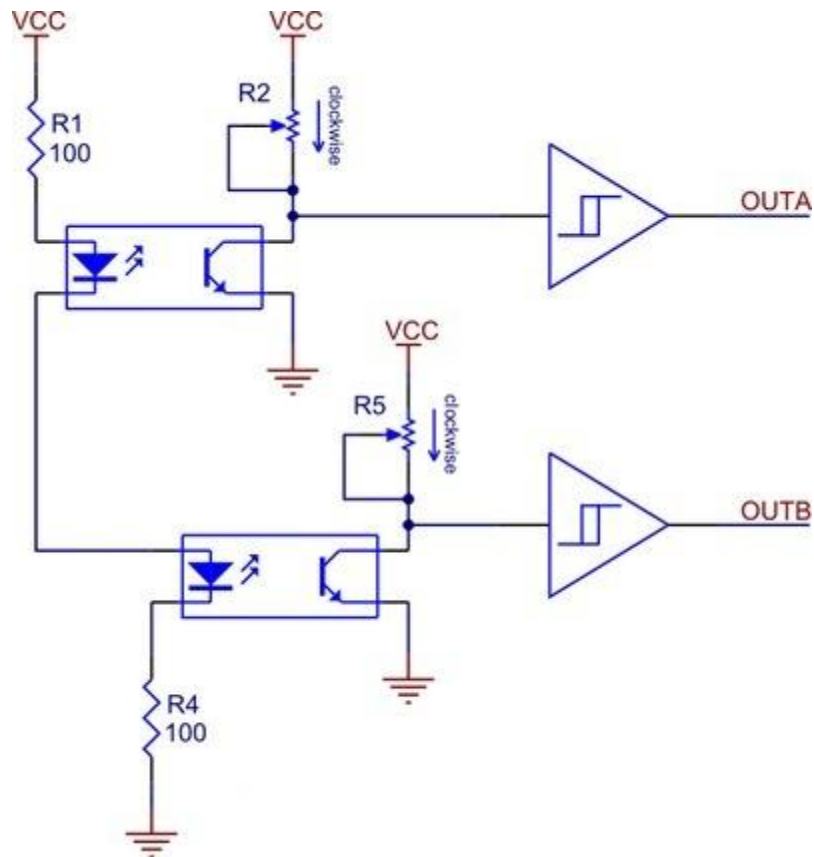
**Fuente.** Pololu URL <https://www.pololu.com/product/1217>

El voltaje de trabajo para el proyecto fue de 5V para tener un mayor rango de calibración de los sensores puesto que cuentan con una resistencia variable que modifica la resolución de los infrarrojos siendo estas R2 y R5 como se muestra en la (figura 91).

El sensor funciona mediante un transistor óptico conformado por un emisor y un receptor, al reflejarse la señal en la superficie dada por el emisor, el receptor o

base del transistor coloca el transistor en saturación permitiendo el paso de la energía entre colector y emisor, la configuración del transistor más el corte y saturación del mismo da como resultado un cero lógico para el controlador

**Figura 91.** Circuito del encoder óptico

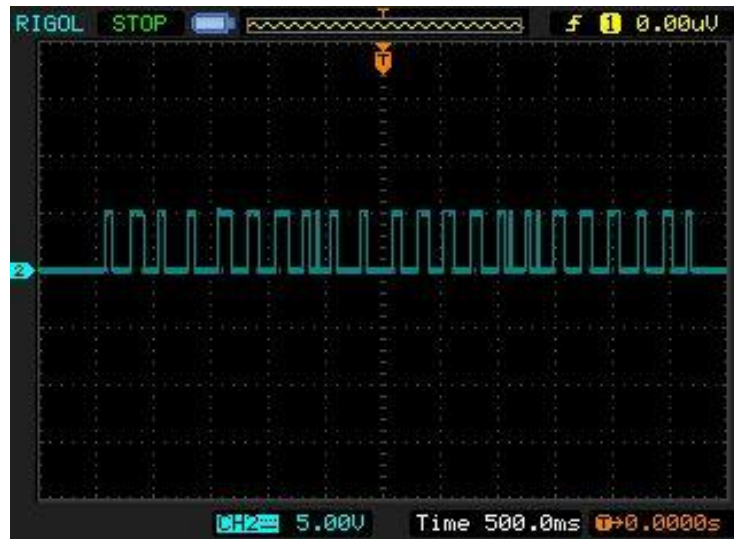


**Fuente.** Pololu. URL. <https://www.pololu.com/product/1217>

La calibración de la señal de salida está dada por los potenciómetros que varían el ancho de pulso del sensor, para calibrarlos se colocó el vehículo con una velocidad constante y se varió el potenciómetro de la señal del canal A que se implementó. La calibración debe estar dada por un 50%, es decir, el flanco de subida y el de bajada deben ser iguales en magnitud lo cual corresponde a un valor de 13k para la resistencia R2.

En primera instancia se observa la señal sin calibrar (figura 92), al variar el potenciómetro hasta el valor mencionado se observa el pulso de subida y de bajada de igual magnitud.

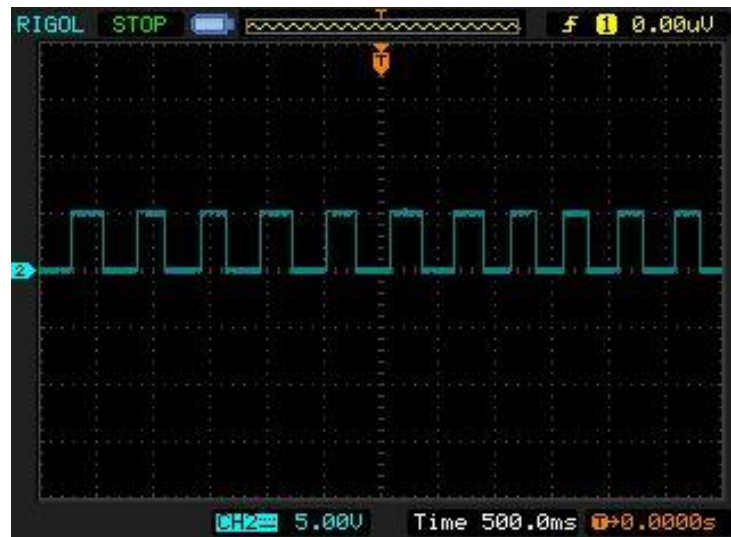
**Figura 92.** Señal sin calibrar



**Fuente.** Elaboración propia

En la (figura 92) se observa la señal inicial obtenida, como se observa en la imagen los flancos de subida y de bajada no son iguales haciendo que la señal de salida no sea la adecuada para realizar las lecturas futuras de RPM.

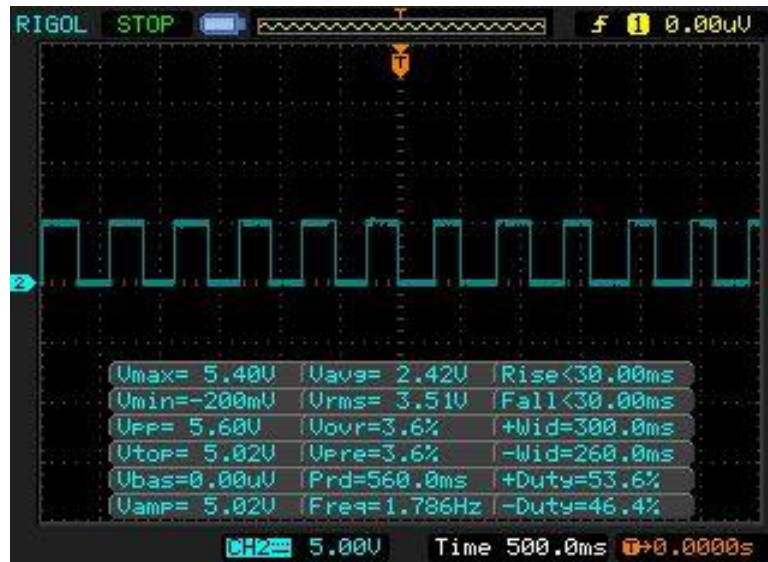
**Figura 93.** Señal calibrada



**Fuente.** Elaboración propia

Tras la calibración de los potenciómetros se obtuvo la señal (figura 93) en la cual se observa que el flanco de subida y de bajada son iguales, por lo tanto la señal obtenida será óptima para las lecturas deseadas.

**Figura 94.** Datos de la señal



**Fuente.** Elaboración propia

En la (figura 94), tras la calibración de la señal se tomaron los datos característicos de ésta siendo los más importantes el voltaje máximo que es 5.4 voltios, voltaje mínimo de 200mv lo cual es muy apropiado para que el controlador lea el flanco de bajada ya que se interpreta el flanco de bajada como una reducción de energía al 10% del valor del flanco de subida.

La caracterización de la cinta del vehículo fue determinada teniendo en cuenta el perímetro de la llanta dada por la ecuación (46).

$$p = 2 * r * \pi \quad (46)$$

Donde ( $p$ ) es igual al perímetro de la llanta y ( $r$ ) el radio de la misma. Reemplazando los valores se obtienen las ecuaciones (47), (48)

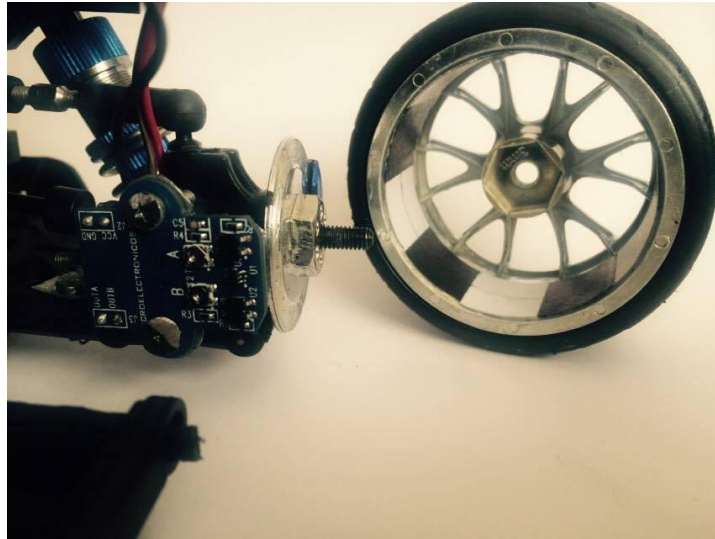
$$p = 2 * 3cm * \pi \quad (47)$$

$$p = 18.84cm \quad (48)$$

Por lo cual se eligió una caracterización de diez cambios de flanco cada uno de 1.8 cm como se muestra en las (figuras 95, 96)



**Figura 95.** Montaje del encoder



**Fuente.** Elaboración propia

**Figura 96.** Caracterización del encoder



**Fuente.** Elaboración propia

Teniendo la caracterización del encoder se calcula los datos de revoluciones por minuto (R.P.M), esta medida hace referencia a cuántas vueltas da en un minuto. Se implementó mediante programación en un micro-controlador un contador de un segundo el cual a su vez cuenta durante este intervalo de tiempo los cambios de flanco, de allí se implementa la ecuación (49)

$$rpm = \frac{\eta * t}{\kappa} \quad (49)$$

Siendo RPM revoluciones por minuto, “ $\eta$ ” el número de cambios de flanco contados en un segundo, “ $t$ ” sesenta segundos que equivalen a un minuto y “ $\kappa$ ” el número de pasos de la caracterización de la llanta, remplazando se obtiene la ecuación (50).

$$rpm = \frac{\eta * 60s}{10} \quad (50)$$

Para la velocidad se hallan las variables que satisfacen la ecuación (51).

$$V = \frac{D}{T} \quad (51)$$

Donde V es la velocidad, D distancia y T el tiempo en segundos. Para hallar la distancia se implementó la ecuación (52).

$$D = \frac{\eta * d * \pi}{360} * \varepsilon \quad (52)$$

Siendo  $\eta$  el número de pulsos  $d$  el diámetro de la rueda y  $\varepsilon$  el valor en mm del cambio de flanco.

Remplazando se obtiene la ecuación (53).

$$D = \frac{\eta * 3 * \pi}{360} * 1.8 \quad (53)$$

#### **b. Sensor ultrasónico HC-SR04**

Otro de los sensores implementados para el desarrollo del proyecto es el sensor ultrasónico HC-SR04 el cual se ilustra en la (figura 97), dicho sensor se utilizó para

medir la distancia entre el vehículo y un objeto predeterminado utilizando la reflexión del sonido sobre él.

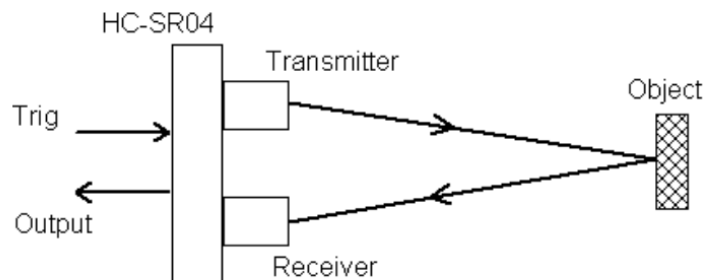
**Figura 97.** Sensor hc-sr04



**Fuente.** Hc-sr04 URL. <https://www.aimagin.com/hc-sr04-ultrasonic-sensor.html>

El sensor funciona con un emisor de ultrasonido y un receptor, además de un sistema de control que caracteriza la señal, el sensor emite una frecuencia de 40Hz la cual viaja por medio del aire y se refleja en un objeto siendo captada por el receptor como se observa en la (figura 98), internamente se inicializa un pulso alto y al recibir la señal reflejada se cambia a un pulso bajo, de acuerdo al ancho de pulso es posible determinar el tiempo que transcurrió entre la emisión y la recepción de la señal.

**Figura 98.** Diagrama de funcionamiento



**Fuente.** Hc-sr04. [http://www.accudiy.com/download/HC-SR04\\_Manual.pdf](http://www.accudiy.com/download/HC-SR04_Manual.pdf)

La distancia estará determinada por la ecuación (54).

$$D = \frac{tr * vs}{2} \quad (54)$$

Donde  $D$  es la distancia,  $tr$  es el tiempo de reflexión de la señal y  $vs$  es la velocidad del sonido en el aire.

Remplazando se obtiene la ecuación (55)

$$D = \frac{tr * 340}{2} \quad (55)$$

El sensor cuenta con unas características de funcionamiento específicas como se muestra en la (tabla 8).

**Tabla 8.** Características del sensor hc-sr04

Voltaje	5V DC
Corriente	15 mA
Frecuencia	40 Hz
Rango máximo	4 m
Rango mínimo	2 cm
Señal del receptor	10 us ttl pulse
Señal del emisor	Nivel de señal y rango proporcional a la entrada
Dimensiones	45*20*15 mm

**Fuente.** Elaboración propia

### 5.3.5 Sistemas electrónicos implementados

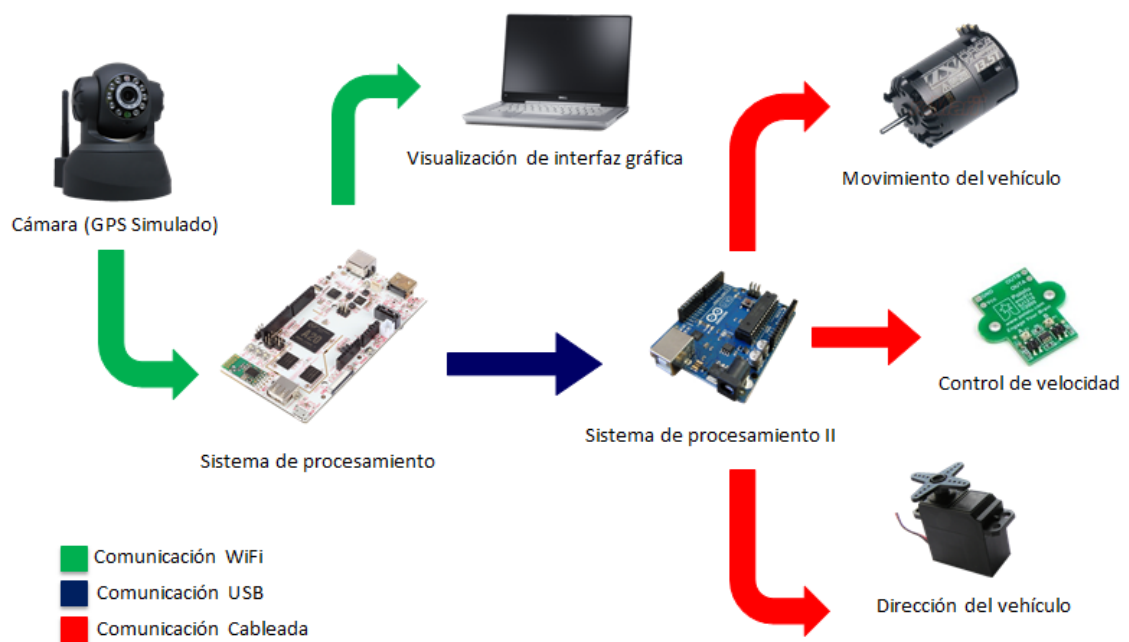
La configuración de los sistemas electrónicos depende de la aplicación que se desea manejar, por ejemplo para la primera aplicación de seguimiento de trayectorias se implementa la configuración ilustrada en la (figura 99)

Para la primera aplicación se implementa un sistema de localización dado por una cámara ubicada a una distancia determinada del suelo, esta no solo visualiza la totalidad del escenario de ejecución de movimiento sino también brinda datos de posición y orientación del escenario antes nombrado, la cámara implementada para el sistema de localización se comunica vía WiFi al controlador principal el cual tiene la tarea de hacer el procesamiento de imagen adecuado para la realizar el sistema de localización, de acuerdo a eso ejerce un control de dirección para corregir errores en la trayectoria a ejecutar previamente seleccionada; la selección

de trayectorias se efectúa en un computador que visualiza el entorno gráfico del pcDuino mediante una aplicación de Google® llamada VNC; la configuración respectiva de la visualización del pcDuino en el computador se expone en el *Anexo I*.

Dicha aplicación funciona mediante la configuración de una IP asignada al pcDuino generando la comunicación inalámbrica entre ellos, en el ordenador se ejecuta la aplicación desarrollada en el programa QT Creator para realizar la selección de la ruta deseada.

**Figura 99.** Configuración de primera aplicación



**Fuente.** Elaboración propia

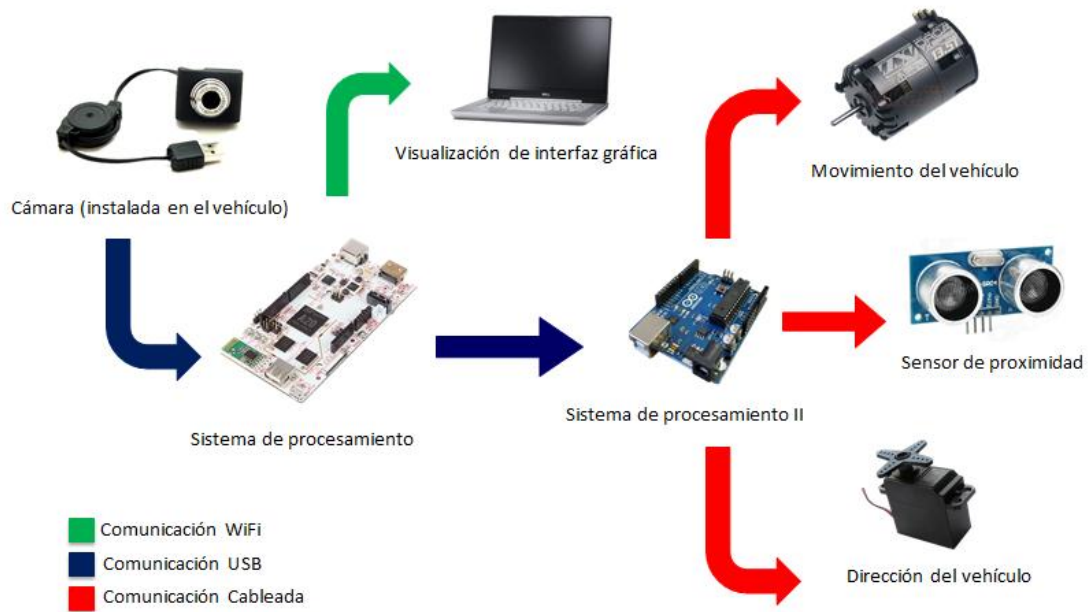
Una vez iniciada la ejecución de la trayectoria entra en acción el segundo controlador, el Arduino Uno recibe órdenes del pcDuino vía USB, para el control de los actuadores conectados de forma cableada al arduino, respondiendo al valor de PWM necesario para hacer las correcciones.

Para el seguimiento de trayectorias se planteó en primera instancia un control de dirección con velocidad constante, sin embargo se plantearon otras dos trayectorias en las cuales se expone el control de velocidad, de allí parte la necesidad de implementar un encoder conectado a el segundo controlador, para el control de velocidad se tiene como referencia la dirección esto con el fin de simular

cada vez más un control para entornos reales donde se disminuye la velocidad a la hora de tomar las curvas.

Para la ejecución de la segunda trayectoria se plantea la configuración mostrada en la (figura 100).

**Figura 100.** Configuración de segunda aplicación



**Fuente.** Elaboración propia

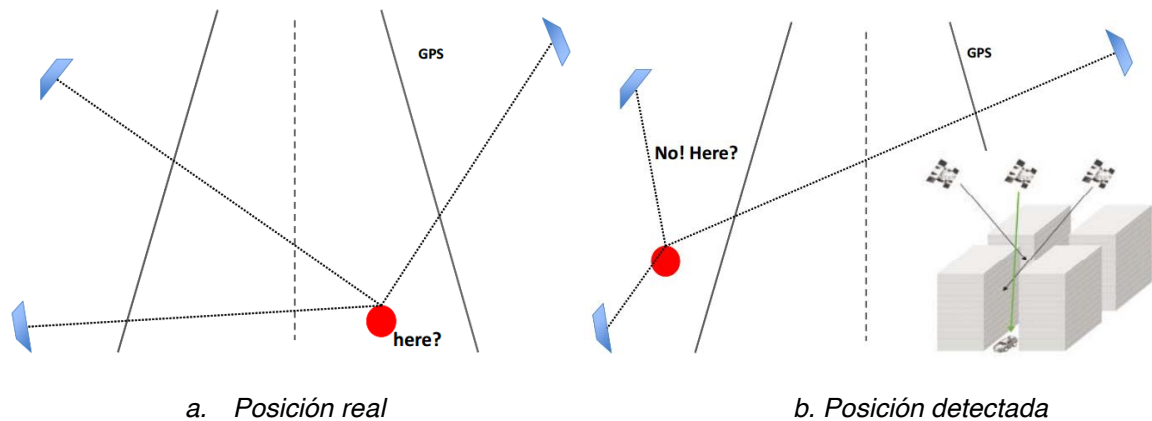
El primer dispositivo de percepción es una cámara embarcada en la parte frontal del carro conectada vía USB al controlador principal, que se encarga de efectuar los filtros para la detección de líneas y peatones respectivamente, la detección de líneas está enfocada a la detección del área navegable en los caminos, la detección de personas se plantea como una asistencia a conductores, donde se plantea una detección de peatones y la reducción de la velocidad del vehículo de acuerdo a la proximidad de estos.

Por tal se tiene la visualización del sistema donde al igual que en la primera aplicación se ejecuta el programa y se muestran variables de interés como la identificación del peatón y la proximidad del mismo en relación del contorno que lo denota. De igual manera se cuenta con un sistema secundario de control para el proceso de ejecución del movimiento (Arduino Uno), sin embargo esta configuración cuenta con sensores ultrasónicos que actúan como sistema de respaldo para la cámara, simulando un paro de emergencia en caso de que el primer sistema de percepción falle o se presente un obstáculo repentino, ejecutando la detención del vehículo de forma inmediata.

## 5.4 SISTEMA DE LOCALIZACIÓN

El sistema de posicionamiento global (GPS) es un sistema que permite determinar la posición de un objeto (vehículo, persona, etc.). Dado a que el GPS puede presentar lecturas erróneas debido a errores en la calibración de los satélites, al estado de algunas capas de la atmósfera (troposfera, ionosfera etc.), el ruido, la reflexión de las señales causadas por edificios, entre otros factores, hacen que se obtengan errores de aproximadamente 1 a 15 metros<sup>89</sup>, como se puede observar en la (figura 101), en el que a campos abiertos la lectura tiene una precisión aceptable (figura 101a), pero en campos con edificios (figura 101b) la lectura es errónea, por tal, dado a que el proyecto se desarrolla en interiores la recepción de datos puede ser mínima.

**Figura 101.** Error de posicionamiento del GPS



**Fuente.** Vaqueiro Ferreira, Janito. *Challenges for Achieving Autonomus Vehicles. Laboratorio de movilidad autónoma.*

Es por ello que se desarrolló un sistema de localización, que basa su funcionamiento en la implementación de una cámara dispuesta en un punto elevado, de manera tal que permite visualizar el área de trabajo, que junto con técnicas de procesamiento de imagen permiten identificar el vehículo, determinando su posición y orientación en tiempo real con gran precisión.

### 5.4.1 Espacio coordinado de trabajo

Al igual que en el GPS convencional, en primera instancia se implementó un sistema que permite visualizar el área de trabajo, es decir, el espacio coordinado en el que se desarrollan las tareas de actuación del vehículo. De manera tal que se implementó la cámara IP WiFi Foscam FI8918W ya que combina de manera eficiente la calidad de video digital y conectividad de red, permitiendo facilidad de

<sup>89</sup> Fuentes de error GPS. [en línea] < <http://www.alsitel.com/tecnico/gps/errores.htm> > Fecha de consulta 01/11/15.

manejo mediante conexión a cualquier red local. Otras especificaciones importantes se mencionan en la (tabla 9).

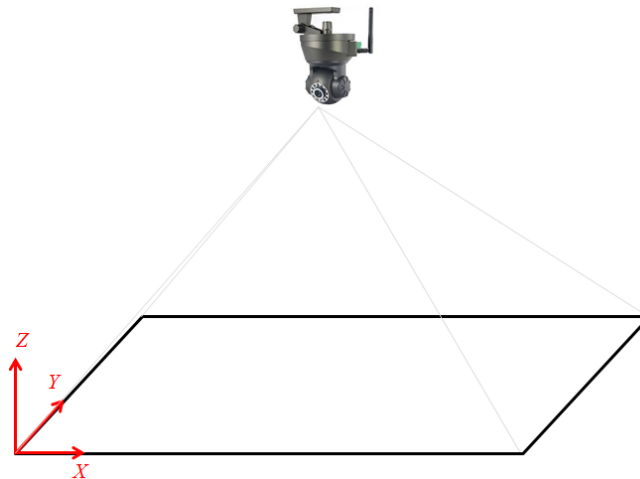
**Tabla 9.** Especificaciones de la cámara IP WiFi Foscam FI8918W

<b>Sensor de imagen</b>	1/5" sensor CMOS
<b>Resolución máxima</b>	640 x480 píxeles
<b>Resolución utilizada</b>	320 x 240 píxeles
<b>Tipos de lentes</b>	Lentes de vidrio
<b>Lentes</b>	f: 2.8mm
<b>Iluminación Min.</b>	0 lux
<b>Frame Rate de la imagen</b>	30fps

*Fuente.* Elaboración propia

La (figura 102) muestra la implementación realizada, en la que se situó la cámara en un punto elevado. El área visual captada por la cámara se define como el espacio de trabajo y se establece como el área predeterminada en donde el vehículo describe las trayectorias específicas. De igual manera se establecen los ejes coordenados (de color rojo) los cuales permiten validar la posición y orientación del vehículo respecto al sistema de referencia global.

**Figura 102.** Cámara dispuesta en un punto elevado



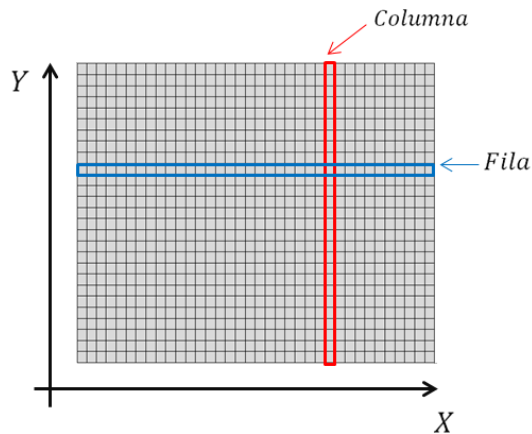
*Fuente.* Elaboración propia

Las imágenes digitales captadas por el sistema de procesamiento consisten en una representación bidimensional (la adición de una tercera dimensión es innecesaria, debido a que no existen variaciones de altitud significativas durante el



desarrollo de las trayectorias), de manera tal que cada uno de los puntos de color o píxeles de la imagen tienen una coordenada  $(X, Y)$ , es decir una posición de fila y columna respectivamente como se observa en la (figura 103).

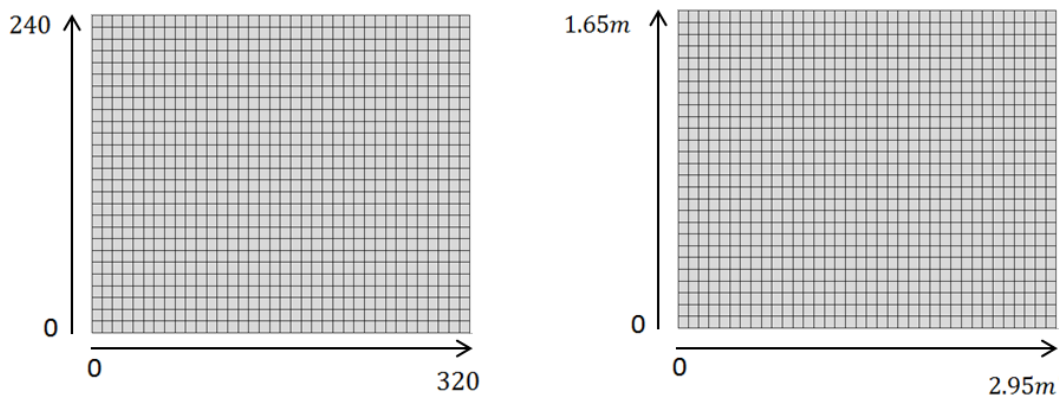
**Figura 103.** *Coordenadas captadas por la cámara*



**Fuente.** *Elaboración propia*

La resolución predeterminada de la cámara es de  $640 \times 480$ , por tal, para proporcionar una tasa de imágenes por segundo lo suficientemente rápida se utilizó una resolución de  $320 \times 240$  lo cual permite un procesamiento de 30 (fps) "frames per second", a comparación de la resolución predeterminada que proporciona 15 (fps). De igual manera se estandarizaron las medidas en metros ( $m$ ) del área captada por el foco de la cámara, la cual es de aproximadamente ( $2.95 \times 1.65 m$ ), de manera tal que por medio de la implementación de interpolación lineal, se estableció que a cada píxel de la imagen le corresponde una distancia determinada, como se ilustra en la (figura 104).

**Figura 104.** *Resolución y distancias captadas por la cámara*



**Fuente.** *Elaboración propia*

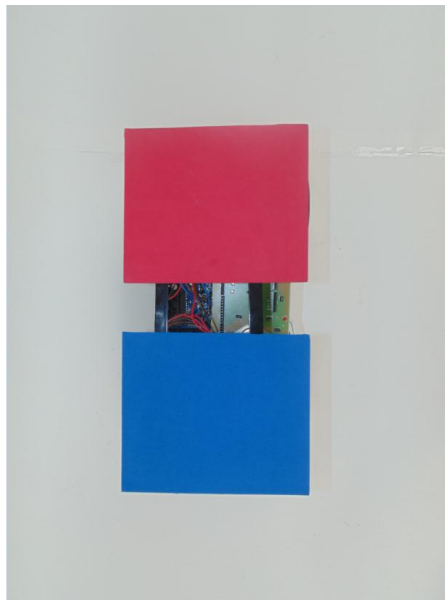
## 5.4.2 Detección del vehículo en el espacio coordenado

La identificación del vehículo dentro del espacio predeterminado, se realizó mediante la aplicación de herramientas de visión artificial, las cuales contemplan la extracción de las características de la imagen basada en el reconocimiento de color, el proceso de detección de contornos y centroide de una figura.

La ventaja que presenta la implementación de una cámara como observador del sistema global es que por medio del reconocimiento del color, es posible realizar una identificación precisa del objeto identificando su punto centro, el cual será denotado como la coordenada de posición en un instante determinado, además el algoritmo implementado permite determinar la orientación del vehículo.

En primera instancia, el vehículo tiene implementado dos marcadores representados por dos cuadrados de diferentes colores, en donde el cuadrado de color rojo representa la parte frontal del vehículo y el cuadrado de color azul representa la parte trasera, tal como se puede observar en la (figura 105).

*Figura 105. Marcadores de posicionamiento del vehículo*

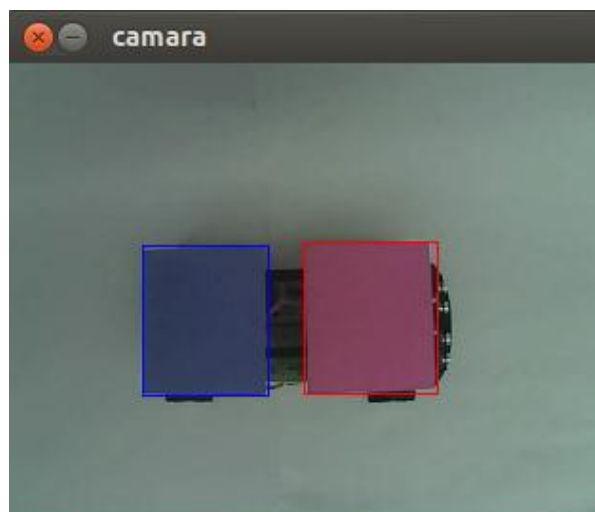


*Fuente. Elaboración propia*

### 5.4.2.1 Posición del vehículo

Con el objeto de saber las coordenadas espaciales del vehículo respecto al sistema global de referencia, en primera instancia se implementó un algoritmo que permitiera identificar los dos marcadores de color instalados en el vehículo, una vez identificados el programa caracteriza los contornos del área de cada cuadro posicionándolos en el espacio coordenado, tal como se observa en la (figura 106).

**Figura 106.** Identificación de contornos de los marcadores de posicionamiento



**Fuente.** Elaboración propia

Como se observa en la (figura 107), para establecer las coordenadas de centro de gravedad del vehículo, en primera instancia se determinó el centroide de cada marcador (figura 107a), de acuerdo a ello, se realiza una comparación entre los puntos medios del centro de cada cuadro y se establece que la distancia media que separa los dos centroides es la coordenada central del vehículo (figura 107b). Este dato es una coordenada  $X_c, Y_c$  (longitud y latitud) y varía de acuerdo a la posición del vehículo en el espacio de trabajo.

**Figura 107.** Centroide del vehículo



a. Centroide de marcadores

b. Centroide global

**Fuente.** Elaboración propia

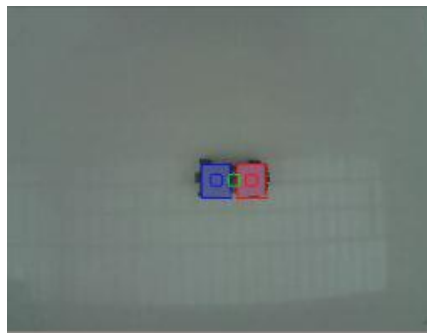
### 5.4.2.2 Orientación del vehículo

La orientación del vehículo se determinó usando los datos proporcionados por cada uno de los centroides de los dos marcadores, en el que se tienen en cuenta los parámetros mencionados en la ecuación (55):

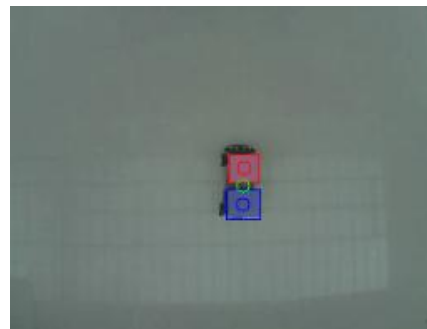
$$\theta = \tan^{-1} \frac{y^* - y}{x^* - x} \quad (55)$$

Donde  $(x^*, y^*)$  representa las coordenadas del marcador de la parte delantera del vehículo y  $(x, y)$  representa las coordenadas de la parte trasera. De manera tal que el algoritmo implementado permite definir los valores de orientación comprendidos entre  $(0^\circ - 360^\circ)$  "giro completo" del vehículo lo que permite validar el algoritmo de control de seguimiento de trayectorias. La (figura 108), presenta la orientación correspondiente en diferentes configuraciones espaciales del vehículo dentro del espacio coordenado de trabajo. Como se puede observar el valor de orientación del vehículo es dependiente de la posición de los marcadores lo cual garantiza una correcta lectura y procesamiento de información.

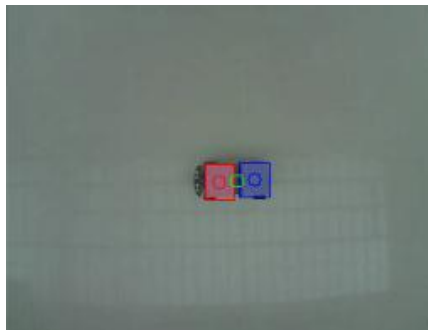
**Figura 108.** Orientación del vehículo (diferentes posiciones)



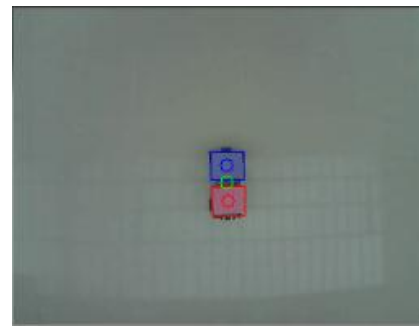
a. Orientación  $(0^\circ)$



b. Orientación  $(90^\circ)$



c. Orientación  $(180^\circ)$



d. Orientación  $(270^\circ)$

**Fuente.** Elaboración propia

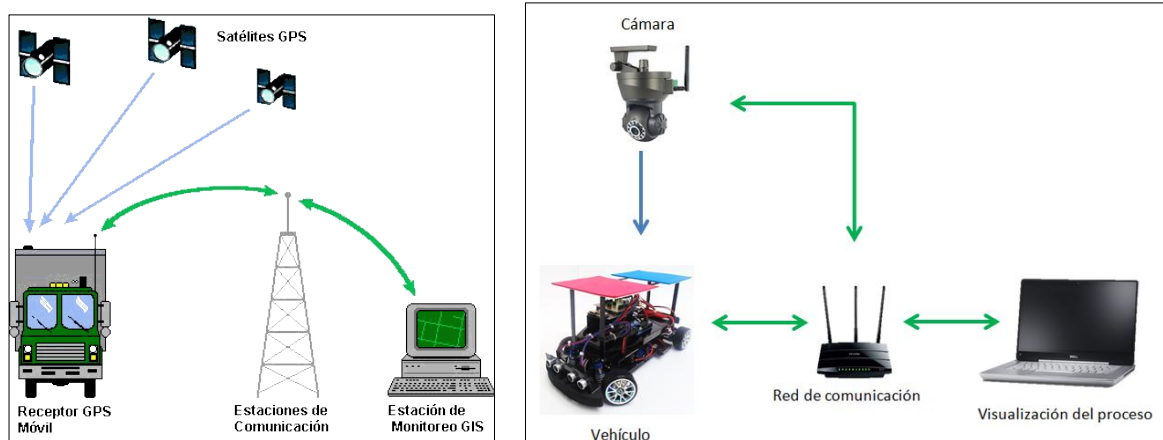
### 5.4.3 GPS y sistema de localización

De acuerdo al sistema implementado es posible realizar una comparación respecto al sistema GPS utilizado actualmente.

La (figura 109), presenta el proceso de comunicación para la detección del vehículo para los dos casos. En el GPS convencional (figura 109a) la red de satélites captan las señales emitidas por el receptor, los cuales por medio de un proceso de triangulación determinan la posición del vehículo (entre otras variables), de manera tal que la estación de comunicación capta los datos suministrados por el receptor y envía los resultados a una estación de monitoreo remoto, por lo cual la persona es capaz de saber la posición del vehículo en tiempo real.

El sistema de localización (figura 109b) está compuesto por una cámara que capta el área visual de actuación del vehículo y los marcadores de color, estas señales son enviadas al dispositivo de control (instalado en el vehículo) por medio de una red de comunicación inalámbrica. El área visual del espacio de trabajo y los procesos de caracterización de visión artificial se pueden observar en un ordenador que se comunica de igual forma inalámbricamente con el dispositivo de control el cual determina la posición y orientación en tiempo real.

**Figura 109.** GPS actual y sistema de localización



*a. GPS actual*

*b. sistema de localización*

**Fuente.** Elaboración propia

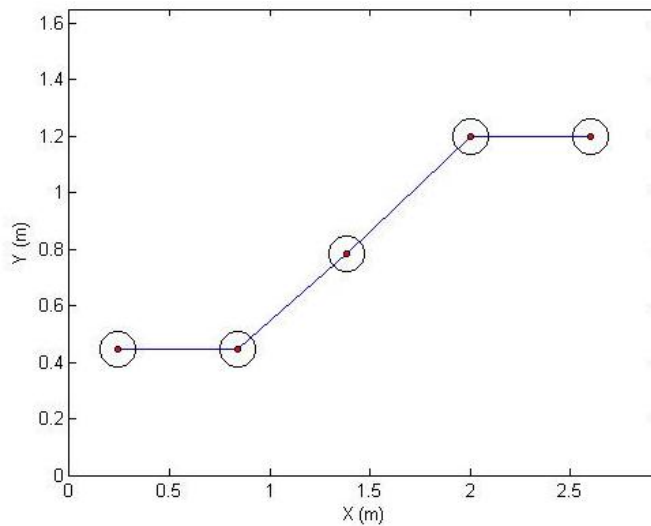
## 5.5 PLANEACIÓN DE LA TRAYECTORIA

La trayectoria proyectada para que el vehículo pueda desplazarse dentro del plano coordenado de trabajo se compone de una serie de  $n$  puntos descritos secuencialmente.

Se considera en primera instancia que la trayectoria es libre de obstáculos, de manera tal que se genera un camino compuesto por la unión del conjunto de rectas definidas, por las cuales se desea que el vehículo pase en un instante determinado. De igual manera es necesario tener presente las dimensiones del área de ejecución de la trayectoria, con el fin de generar un camino que el vehículo sea capaz de describir.

De los métodos existentes de planeación de trayectorias, el método que más se ajusta al proyecto es la generación de trayectoria punto a punto, es decir, se establecen una serie de waypoints dentro del espacio coordenado con el fin de que el vehículo planee y describa la trayectoria formada por la unión de todos los puntos. Para ello, como se observa en la (figura 110) se representan los puntos seleccionados con el cual se busca que el vehículo describa un cambio de dirección de giro izquierda y derecha, que visualmente en el plano se observa como un cambio de carril.

**Figura 110.** Planeamiento de trayectoria punto a punto



**Fuente.** Elaboración propia

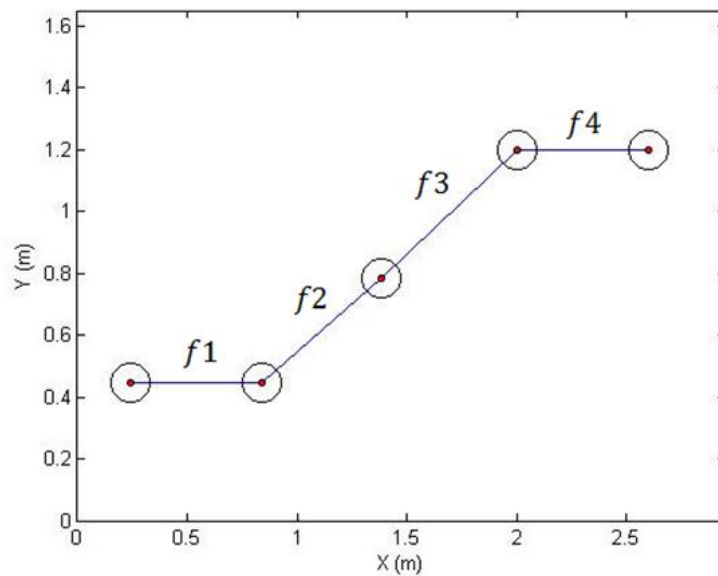
Dado a que el vehículo tiene restricciones no holonómicas (definición descrita en el capítulo 4), para que el vehículo describa con precisión la trayectoria es necesario tener en cuenta que el giro de las ruedas pueda describir los cambios de dirección presentes en la gráfica. Por tal para la obtención de una trayectoria mejorada se implementa la interpolación cúbica Spline.

El Spline cúbico es un polinomio descrito por partes, es empleado para realizar interpolación entre segmentos descritos por puntos definidos, cuyo objetivo es suavizar los cambios rectilíneos entre puntos, dando como resultado curvas suaves, los cuales aproxima el comportamiento de la trayectoria descrita por el vehículo al dar giros toscos. El escenario propuesto se compone de una trayectoria definida por la consecución de líneas rectas unidas por puntos, de manera tal que se asume que la trayectoria se compone de  $n + 1$  parejas de datos, como se describe en la ecuación (56)

$$(X_0, Y_0), (X_1, Y_1) \dots, (X_n, Y_n) \quad (56)$$

Cada segmento de la trayectoria se define como una función, tal como se observa en la (figura 111), de manera que cada función se describe como un polinomio de orden 3.

**Figura 111.** Segmentos de la trayectoria definida por funciones



**Fuente.** Elaboración propia

Cada función se representa como se muestra en la ecuación (57)

$$f_k(x) = a_{0,k} + a_{1,k}x + a_{2,k}x^2 + a_{3,k}x^3 \quad (57)$$

Donde  $k = 1, \dots, n$

Teniendo en cuenta que  $n$  representa el número de parejas de datos.

En adición a ello, es necesario tener en cuenta una serie de condiciones que se necesitan para la solución del Spline. La primera de ellas es que existan condiciones de interpolación como se define en la ecuación (58)

$$f_k(x_k - 1) = y_k - 1 \quad (58)$$

De forma similar se establece una función de continuidad de los nodos representados en la ecuación (59)

$$f_k(x_k) = y_k \quad (59)$$

También es necesario tener en cuenta que se establecen las condiciones de suavidad, es decir, se evalúan las derivadas de cada función, como se expone en la ecuación (60) y (61)

$$f'_k(x_k) = f'_{k+1}(x_k) \quad (60)$$

$$f''_k(x_k) = f''_{k+1}(x_k) \quad (61)$$

De acuerdo a ello, se procede a realizar la solución del Spline de acuerdo a la trayectoria implementada, para ello se utiliza el software Matlab®. A continuación se expone el pseudocódigo utilizado.

**Programa:** Spline cubica

**Variables:** puntos coordenados en x  
puntos coordenados en y

**Algoritmo**

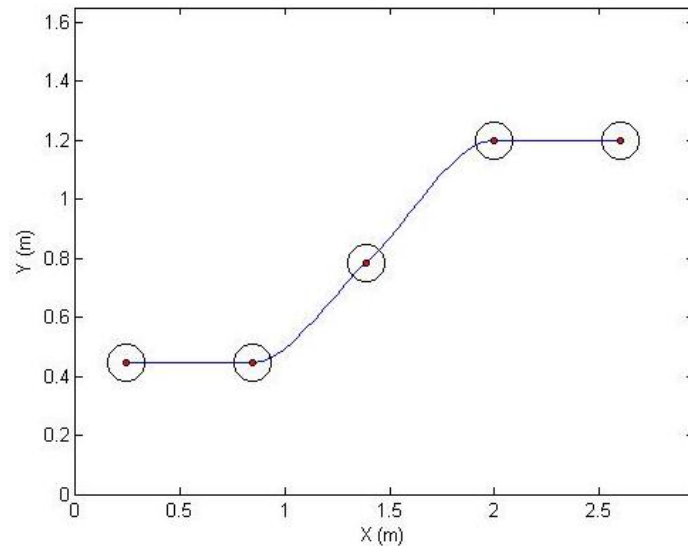
Leer puntos coordenados en x y en y  
Definición de variable final e inicial  
Declaración de tiempo de muestreo  
Declaración de la función interp1  
Asignación de método 'cubic'  
Graficación de tiempo de muestreo vs interp1  
Asignación de puntos coordenados en la imagen

Finprograma



De acuerdo a ello como se observa en la (figura 112), se obtiene la trayectoria resultante mediante Spline, en el que se observa el suavizado de curvas describiendo los mismos puntos iniciales, en el que el vehículo una vez identifique dichos waypoints, planea el camino para describir cada uno (línea de color azul).

**Figura 112.** Trayectoria cambio de carril mediante Spline cúbica

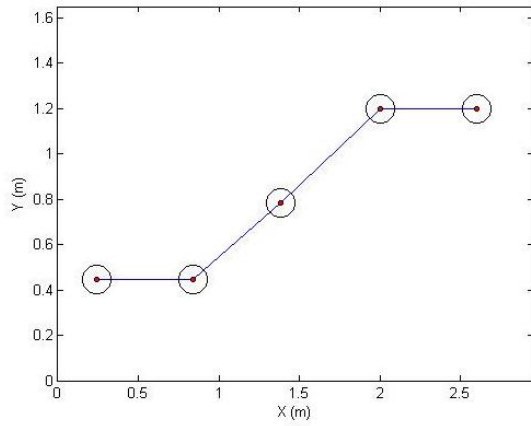


**Fuente.** Elaboración propia

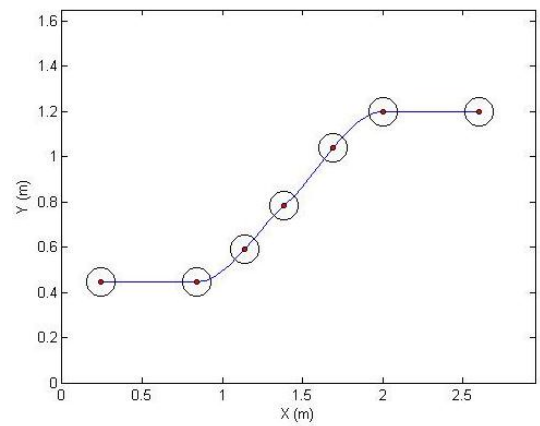
Una vez obtenida la trayectoria mediante Spline, se plantean otras dos trayectorias con el objeto de validar los movimientos del vehículo dentro del espacio de trabajo (giro izquierda y giro derecha), las cuales se obtienen realizando el mismo procedimiento anterior. En adición a ello se seleccionan los waypoints de cada trayectoria Spline y se compara respecto a la trayectoria punto a punto generada en un principio, como se ilustra en la (figura 113).

Como se puede observar, la trayectoria definida por Spline describe de una manera correcta la trayectoria posible que puede describir el vehículo, por tal, se definen las trayectorias (figura 113b, 113d, 113f) resultantes para que el vehículo por medio del control implementado valide el algoritmo de seguimiento de trayectorias, en la primera, asumiendo una velocidad constante y en las otras dos, aplicando un algoritmo de control de velocidad (que se explicaran dentro del desarrollo de este capítulo).

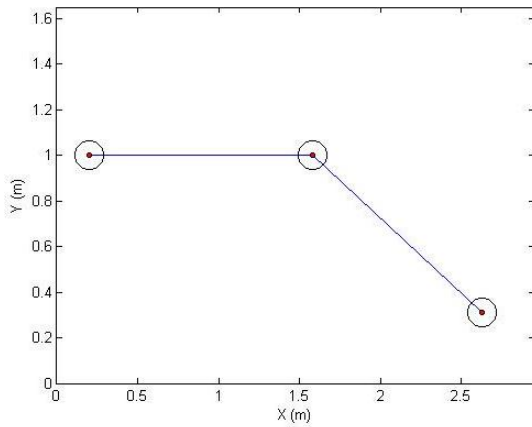
**Figura 113. Trayectorias definidas**



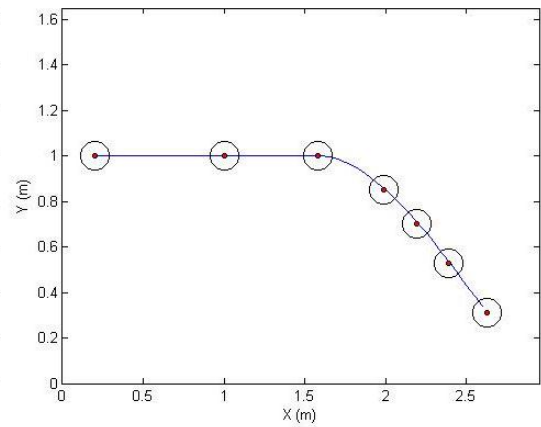
**a. Cambio de carril (punto a punto)**



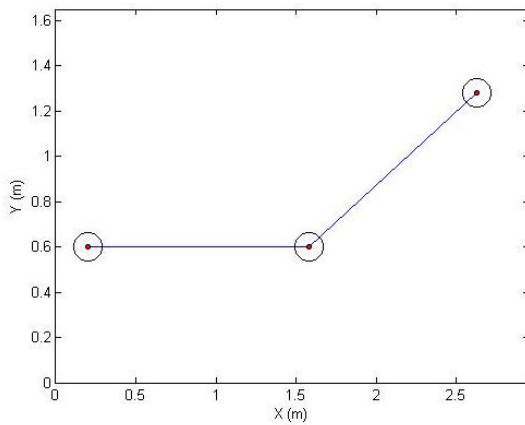
**b. Cambio de carril (Spline)**



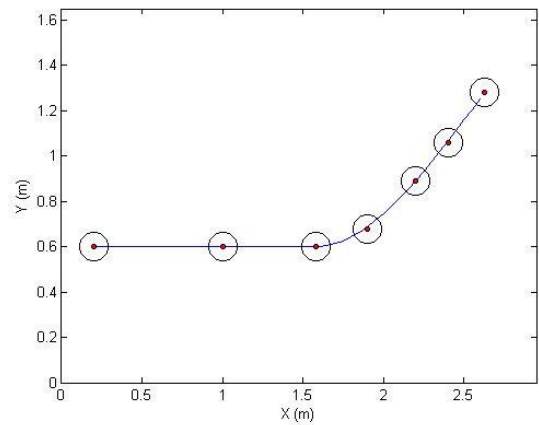
**c. Giro derecha (punto a punto)**



**d. Giro derecha (Spline)**



**e. Giro Izquierda (punto a punto)**



**f. Giro izquierda (Spline)**

**Fuente. Elaboración propia**

## 5.6 CONTROL DE ACTUACIÓN DEL SISTEMA

### 5.6.1 Algoritmo de seguimiento de trayectorias

La implementación del algoritmo tiene por objeto, orientar al vehículo hacia la posición deseada, dados los puntos que definen la trayectoria y las condiciones iniciales del sistema. En primera instancia se considera una velocidad constante para el vehículo durante la ejecución de su movimiento. Se parte del problema de que el robot tendrá que llegar a un punto ( $p$ ) definido en el sistema coordenado de trabajo, el cual se define en la ecuación (62).

$$p = (x^* \ y^*) \quad (62)$$

Con el objetivo de que el vehículo se desplace al punto indicado teniendo en cuenta coordenadas inicial y final, se tiene en cuenta la orientación global del sistema actual respecto al deseado, representado en la ecuación (63):

$$\theta^* = \tan^{-1} \frac{y^* - y}{x^* - x} \quad (63)$$

La siguiente consideración es evaluar una de las ecuaciones características del sistema determinada anteriormente y representada en la ecuación (64)

$$\dot{\theta} = \frac{v \tan(\delta)}{L} \quad (64)$$

El control de dirección radica en realizar la diferencia angular entre  $\theta^*$  y  $\dot{\theta}$  retornando una diferencia en el intervalo entre  $(-\pi, \pi)$  [1], de manera tal que el vehículo ajusta la dirección de las llantas delanteras con el fin de llegar a la coordenada deseada. Con el objeto de reducir el error y mejorar la respuesta del sistema se implementó un controlador proporcional en el valor de la corrección angular como se observa en la ecuación (65):

$$\delta = K_h * \theta^* \quad (65)$$

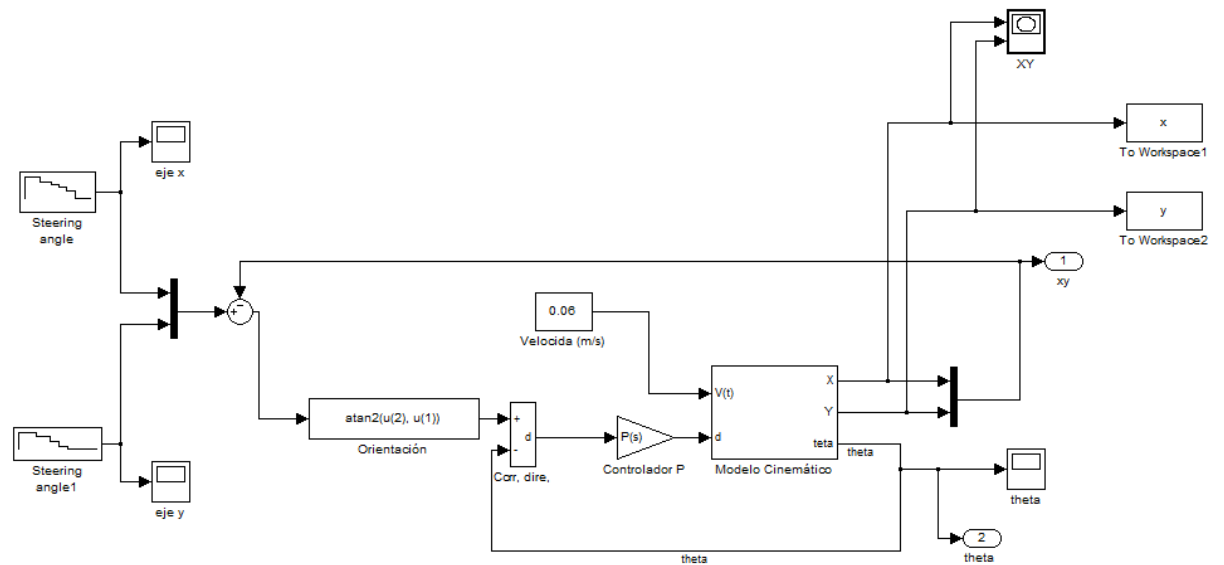
Siendo  $K_h > 0$

Donde  $\theta^*$  representa el valor de corrección de dirección del intervalo comprendido entre  $(-\pi, \pi)$ . De igual manera es necesario considerar que el vehículo tiene un limitante mecánico en su dirección por tal se establece que las llantas delanteras tienen un giro (externo) máximo de  $n$  grados respecto a su centro, por tanto se tienen en cuenta las consideraciones de la ecuación (66):

$$\begin{aligned} \text{llanta der} &= \delta_{\max} \\ \text{llanta izq} &= -\delta_{\max} \end{aligned} \tag{66}$$

De acuerdo a los parámetros mencionados se procede a desarrollar todo el sistema de control en Simulink de Matlab®, con el objetivo de simular el comportamiento del vehículo en la ejecución de la trayectoria, teniendo en cuenta de igual manera la inclusión del modelo cinemático desarrollado en el capítulo 4, tal como se ilustra en la (figura 114).

**Figura 114.** Algoritmo de control

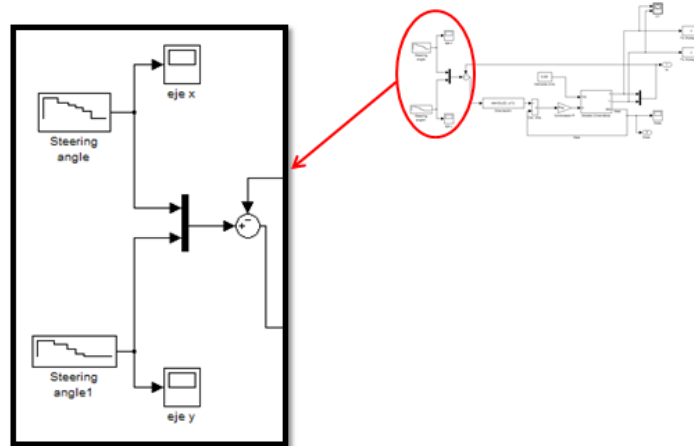


**Fuente.** Elaboración propia

Como se puede observar, el algoritmo desarrolla un sistema de control en lazo cerrado, dado a que el sistema necesita realizar la comparación entre los valores de posición y orientación actual respecto a las coordenadas deseadas, de acuerdo al error de estas variables se ajusta la dirección del vehículo con el fin de que describa la trayectoria satisfactoriamente. En primera instancia se definen los puntos de la trayectoria, es decir, cada uno de los puntos por los cuales el vehículo

deberá pasar en un orden determinado. En la (figura 115) se representan las coordenadas  $(X, Y)$  de la trayectoria.

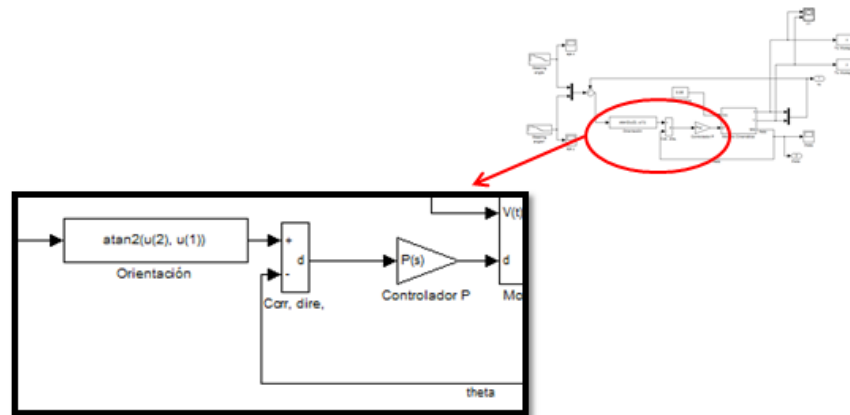
**Figura 115.** Coordenadas de la trayectoria



**Fuente.** Elaboración propia

Como se observa en la figura 116, el sistema toma en cuenta la orientación actual y de referencia, de acuerdo a ello envía el resultado de dicha diferencia a un bloque denominado “*corrección de dirección*” el cual entrega al sistema el valor de dirección correspondiente al giro de las ruedas delanteras. Finalmente se observa el controlador proporcional implementado, el cual amplifica el valor resultante para que la respuesta del cambio de dirección sea más precisa.

**Figura 116.** Orientación y corrección de dirección

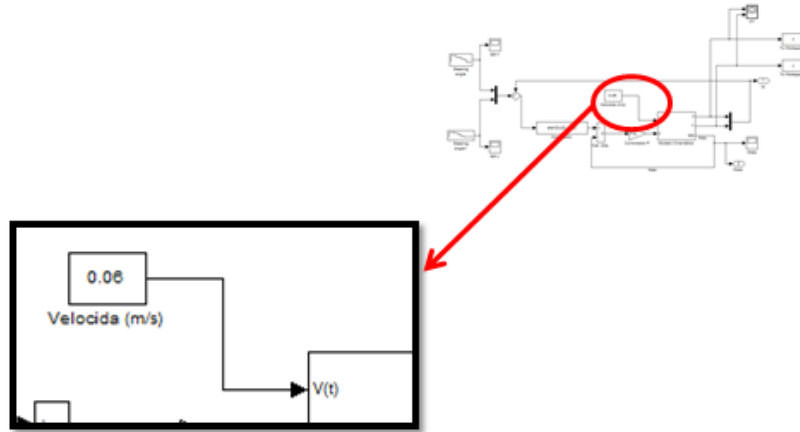


**Fuente.** Elaboración propia

De igual manera para las simulaciones de la trayectoria definida por el vehículo se asume una velocidad constante como se observa en la (figura 117), de acuerdo a

los parámetros de los sensores y condiciones del sistema, se realiza la prueba con una velocidad de  $6 \text{ cm/s}$ .

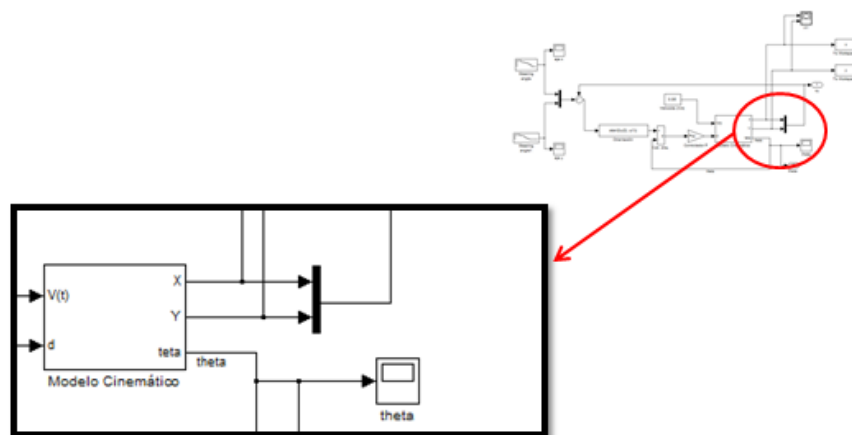
**Figura 117.** Velocidad



**Fuente.** Elaboración propia

Por último se observan las variables de salida del modelo cinemático (posición y orientación actuales), los cuales en el ambiente físico se obtienen mediante el sistema de localización, y son las variables que retroalimentan el sistema para realizar el lazo de control, como se observa en la (figura 118).

**Figura 118.** Datos de posición y orientación del vehículo



**Fuente.** Elaboración propia

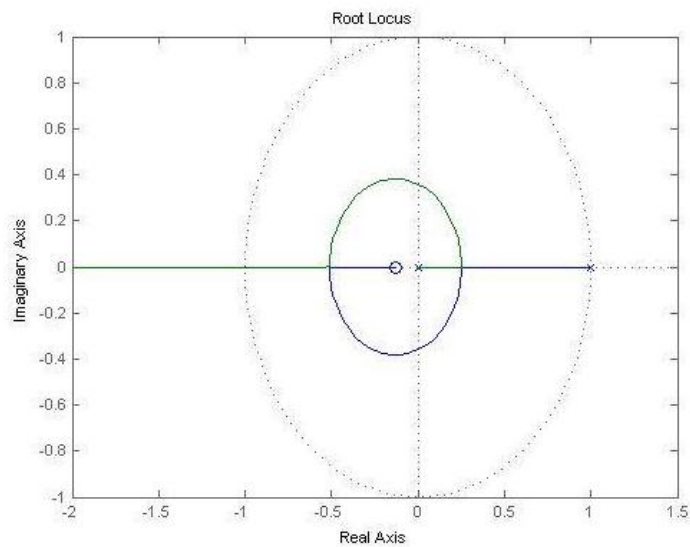
## 5.6.2 Control de velocidad PID

Para calcular un control sobre la velocidad se planteó en primera instancia un PID por Ziegler-Nichols en discreto como se ve en la ecuación (67)

$$k(z) = k_p + \left(\frac{z-1}{Tz}\right)kd + ki\left(\frac{Tz}{z-1}\right) \quad (67)$$

Para el desarrollo se determinó la ganancia  $km$  tal que el sistema empiece a escalar polos de malla cerrada sobre el circuito unitario, también se determina  $Wn$  de la gráfica de polos como se observa en la (figura 119).

**Figura 119.** Diagrama de polos y ceros



**Fuente.** Elaboración propia

De la gráfica se obtiene un  $km = 9.1e^{-3}$  y un  $wn = 153$ ; con estos valores se determinan las ganancias dadas por  $ki$ ,  $kd$ ,  $kp$  las cuales se caracterizan por las ecuaciones (68), (69), (70)

$$k_p = 0.6km \quad (68)$$

$$k_d = kp \frac{\pi}{4} wn \quad (69)$$

$$k_i = kp \frac{wn}{\pi} \quad (70)$$

Se reemplazan las ganancias en la ecuación de PID y se multiplica por la planta (función de transferencia del motor), a su vez el resultado se multiplica por un escalón para observar la respuesta como se ilustra en la (figura 120) Para la

simulación se implementó el software Matlab® en donde se describe el siguiente pseudocódigo.

**Programa:** PID Ziegler-Nichols

**Variables:**

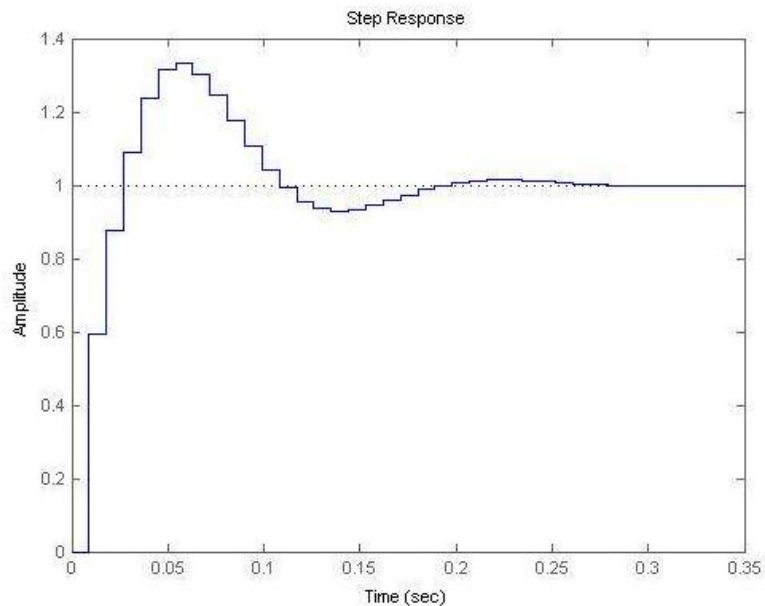
Tiempo de muestreo  
Numerador  
Denominador

**Algoritmo:**

Aplicación de la transformada Z para la planta  
Calculo de grafica de polos y ceros  
Identificación de las variables  $k_m$  y  $\omega_n$  de la grafica  
Calculo de las ganancias  $k_i$ ,  $k_p$ ,  $k_d$   
Remplazo de variables en la formula general de un PID discreto  
Multiplicación de la planta por el PID  
Aplicación de un escalón para visualizar el comportamiento

Fin de programa

**Figura 120.** Respuesta del sistema por PID Ziegler-Nichols



**Fuente.** Elaboración propia

De la gráfica se puede observar una estabilización en 0.3 segundos lo cual es óptimo para el control que se desea implementar, sin embargo también se muestra una respuesta subamortiguada al sistema por lo cual se implementa un PID por lugar de las raíces, con el fin de generar una respuesta crítica del sistema.



El PID por lugar de las raíces maneja la misma fórmula general de PID discreto como se muestra en la ecuación (67), la variación radica en la obtención de las ganancias  $k_i$ ,  $k_p$ ,  $k_d$ . La obtención de la ganancia  $k_i$  está determinada por la ecuación (71) para un error estacionario de 0.2

$$\lim_{s=0} p(s)k(s) \quad (71)$$

De la ecuación  $p(s)$  hace referencia a la planta y  $k(s)$  a la formula general del PID.

Remplazando se obtiene la ecuación (32).

$$\lim_{s=0} \frac{6.6(kds^2 + kps + ki)}{s^2 + 969.2s + 16.06} = 0.2 \quad (72)$$

Despejando la ganancia  $k_i$  tiene como resultado la ecuación (73)

$$k_i = 0.2 \quad (73)$$

Las ganancias  $k_p$  y  $k_d$  están determinadas por las ecuaciones (74), (75)

$$k_d = \frac{k_i * (a * f - d * c) + d * alpha - a * beta}{d * b - a * e} \quad (74)$$

$$k_p = \frac{k_i * (e * c - f * b) - e * alpha + b * beta}{d * b - a * e} \quad (75)$$

De donde  $a$ ,  $b$ ,  $c$ ,  $d$ ,  $e$ ,  $f$ ,  $alpha$  y  $beta$  están determinados por las ecuaciones (76), (77), (78), (79), (80), (81), (82), (83) respectivamente.

$$a = (\sigma^2 - \sigma - wd^2) * Ta \quad (76)$$

$$b = \sigma^2 - wd^2 - 2 * \sigma + 1 \quad (77)$$

$$c = (\sigma^2 - wd^2) * Ta^2 \quad (78)$$

$$d = (2 * \sigma * wd - wd) * Ta \quad (79)$$

$$e = 2 * \sigma * wd - 2 * wd \quad (80)$$

$$f = (2 * \sigma * wd) * Ta^2 \quad (81)$$

$$alpha = mk * (a * \cos(\theta) - d * \sin(\theta)) \quad (82)$$

$$beta = mk * (a * \sin(\theta) + d * \sin(\theta)) \quad (83)$$

Y se da del ángulo de la respuesta en frecuencia de la planta. Una vez obtenidos los valores de ganancia se reemplazan en la fórmula del PID y se multiplica la

ecuación final por la planta, para hallar la respuesta del controlador se multiplica por un escalón como se muestra en la (figura 121). Para la simulación se utilizó el siguiente pseudocódigo en Matlab®

**Programa:** PID lugar de las raíces

**Variables:**

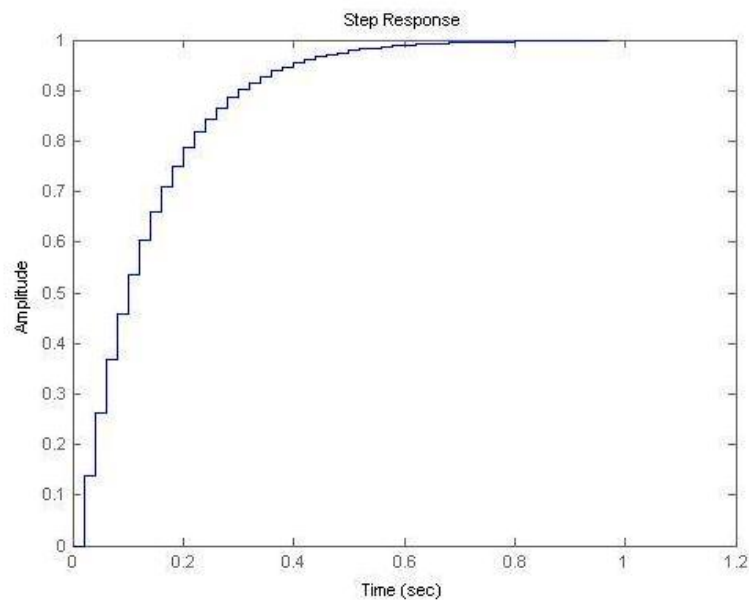
Tiempo de muestreo  
Numerador  
Denominador  
Variable ki

**Algoritmo:**

Aplicación de la transformada Z para la planta  
Calculo de grafica de polos y ceros  
Calculo para las variables sigma y wn  
Calculo del ángulo theta  
Calculo de las variables a, b, c, d, e, f, alpha y beta según parámetros  
Calculo de las ganancias kp y kd  
Reemplazo de variables en la formula general de un PID discreto  
Multiplicación de la planta por el PID  
Aplicación de un escalón para visualizar el comportamiento

Fin de programa

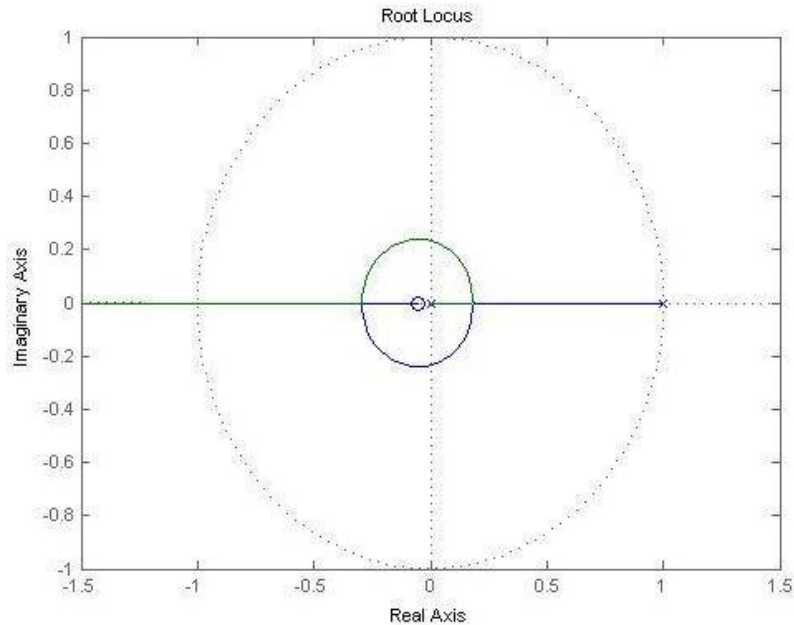
**Figura 121.** Respuesta del sistema por PID lugar de las raíces



**Fuente.** Elaboración propia

De la gráfica se observa cómo se estabiliza en tiempo igual a 1 segundo, a diferencia del anterior controlador con una respuesta críticamente amortiguada, la cual es óptima para el control de velocidad deseado, al igual para corroborar la estabilidad del sistema se realizó la gráfica de polos y ceros donde se demuestra la estabilidad del sistema dado que el polo tiende a anular al cero como se muestra en la (figura 122).

**Figura 122.** Diagrama de estabilidad de polos y ceros



**Fuente.** Elaboración propia

## 5.7 PROGRAMACIÓN PRIMERA APLICACIÓN

En el software QT Creator se desarrolló toda la programación correspondiente al sistema embebido, la programación de la primera aplicación contempla el desarrollo del sistema de localización, la asignación de las tres trayectorias, y la implementación del algoritmo de seguimiento de trayectorias.

A continuación se muestra el pseudocódigo correspondiente a la programación del pcDuino.

**Programa:** Primera aplicación

### **Algoritmo:**

- Incluir librerías de OpenCV
- Incluir librerías para comunicación serial
- Incluir librerías para creación de interfaz grafica
- Definición de variables globales

Definir comunicación serial con arduino  
 Establecer condición de enlace  
 Definir el puerto serial de comunicación  
 Creación de apuntadores para tratamiento de imágenes  
 Función boton on\_serial  
     Si -> el estado del puerto = 0;  
         Crear objeto de tipo serial  
         Asignar puerto correspondiente  
         Abrir la comunicación de lectura y escritura  
     Si -> no se detecta comunicación serial  
         Emitir mensaje de alerta  
         Cerrar comunicación existente  
     Sí No  
         Emitir mensaje de comunicación establecida  
         Asignar configuración de comunicación serie correspondiente  
     Fin SI  
 Fin función botón on\_serial  
  
 Función botón off\_serial  
     Si botón off\_serial es activo  
         cerrar comunicación serial  
         emitir mensaje puerto liberado  
     Fin SI  
 Fin función botón off\_serial  
  
 Función salir  
     Cerrar ventanas graficas existentes  
 Fin función salir  
  
 Función Speed  
     Enviar arduino PWM de activación del Speed  
 Fin función Speed  
  
 Función aplicación (inicio)  
     Capturar la dirección IP de la cámara  
     Redimensionamiento de la imagen  
     Declaración de variables  
     Clonar imagen original  
     Definir puntos iniciales de cada trayectoria  
     Mientras (1)  
         Aplicación -> activo  
         Lectura de imagen redimensionada  
         Asignación de variables internas  
         Asignación del espacio de color  
         Selección de rangos del color rojo  
         Selección de rangos del color azul

```

Determinar área de color rojo
Determinar área de color azul
Selección de contornos del color rojo
Selección de contornos del color azul
Graficar centroide del recuadro rojo
Graficar centroide del recuadro azul
Leer puntos centros de recuadros de color (posición)
Interpolar medidas del centro del vehículo (metros y pixeles)
Hallar arco-tangente de centro rojo respecto azul (orientación)
Implementar corrección angular
Definir restricción para ruedas delanteras
Mapeo de dirección respecto a función de restricción
Establecer error porcentual de posición
Definición de velocidad
//valor de velocidad constante para trayectoria cambio de carril
//no se envía valor de velocidad para trayectoria giro derecho e izquierdo, la
velocidad está dada por el encoder implementado en el arduino

Si (primera trayectorias es seleccionada)
    Definición de los puntos de la trayectoria Spline cambio de carril
    Enviar valor de velocidad
    Enviar valor de dirección
Fin Si
Si (segunda trayectorias es seleccionada)
    Definición de los puntos de la trayectoria Spline giro derecha
    Enviar valor de dirección
Fin Si

Si (tercera trayectorias es seleccionada)
    Definición de los puntos de la trayectoria Spline giro izquierda
    Enviar valor de dirección
    Fin Si
Fin mientras
Función reset
Cerrar aplicación
Fin función reset
Fin aplicación
Fin de programa

```

En el *Anexo B* se expone el diagrama de flujo del programa primera aplicación. Dado que el Arduino es el controlador esclavo, también se encuentra programado para recibir las señales provenientes del pcDuino. Para la primera aplicación se implementaron dos códigos; en el primero Arduino ejecuta una velocidad constante y la dirección se ajusta conforme al sistema de localización (válido para la trayectoria cambio de carril) y en el segundo se implementó el PID de velocidad

y la dirección se ajusta conforme al sistema de localización (válido para la trayectoria giro izquierda -derecha). A continuación se muestra el pseudocódigo correspondiente a la programación de Arduino para la primera trayectoria.

**Programa:** Primera aplicación (trayectoria cambio de carril)

**Algoritmo:**

Incluir librería de servo

Incluir EEPROM

Declaración de variables dirección y velocidad

Definir comunicación serial

Definición de puertos de entrada y salida

Definición de comunicación para variables internas

Leer puerto serie

    Si (dirección está entre los rangos de lectura)

        Mapear valor de entrada respecto al PWM de activación del servomotor  
        enviar valor mapeado

    Fin Si

    Si (velocidad está entre los rangos de lectura)

        Mapear valor de entrada respecto al PWM de activación del speed  
        enviar valor mapeado

    Fin Si

Fin de programa

En el *Anexo C* se expone el diagrama de flujo del programa primera aplicación (trayectoria cambio de carril). A continuación se muestra el pseudocódigo correspondiente a la programación de Arduino para la segunda y tercera trayectoria.

**Programa:** Primera aplicación (trayectoria giro derecha e izquierda)

**Algoritmo:**

Incluir librería de servo

Declaración de variables

Definir comunicación serial

Definición de puertos de entrada y salida

Definición de comunicación para variables internas

Leer puerto serie

    Si (dirección está entre los rangos de lectura)

        Mapear valor de entrada respecto al PWM de activación del servomotor  
        enviar valor mapeado

    Comparación estado actual (encoder) con referencia en la misma escala (servo)

    Hallar el valor  $k_i$ ,  $k_p$ ,  $k_d$  mediante fórmulas de acuerdo al error

    Multiplicación de ganancias  $k_i$ ,  $k_p$ ,  $k_d$  obtenidas de la simulación

    Envío del valor de PWM (speed control)

Fin de programa

En el *Anexo D* se expone el diagrama de flujo del programa primera aplicación (Trayectoria giro izquierda y derecha)

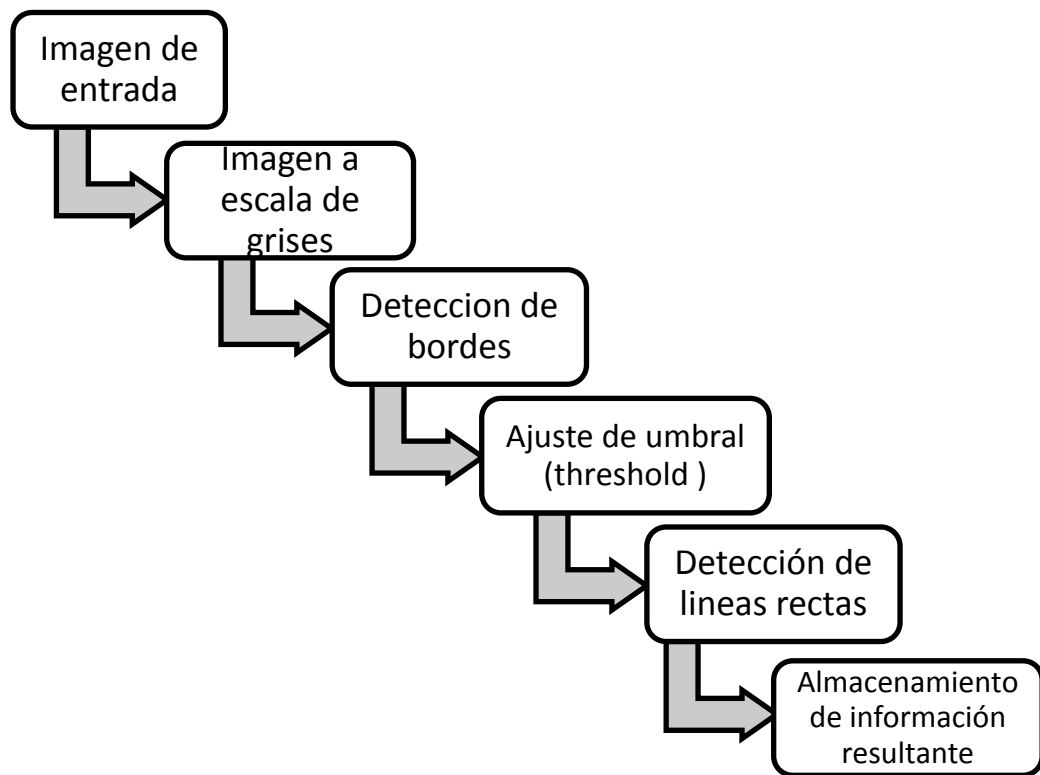
## 5.8 DETECCIÓN DE ÁREA NAVEGABLE Y PEATONES

La segunda aplicación propuesta es generar aplicaciones de apoyo en la conducción basada en el reconocimiento de líneas en la vía, y de igual forma la detección de peatones en tiempo real. Para ello se desarrolló mediante aplicaciones de visión computacional, técnicas para la extracción de información y junto con el uso de librerías propias de OpenCV®, se generaron los algoritmos necesarios para la identificación de estos parámetros.

### 5.8.1 Detección de área navegable

En primera instancia es necesario tener presente que el área navegable está descrita como el espacio por el cual el vehículo puede transitar de forma segura, para este caso se desarrolla un algoritmo capaz de identificar las líneas del camino mientras el vehículo se desplaza en el ambiente. Para ello es necesario realizar diferentes aplicaciones basadas en el desarrollo de la visión artificial como se puede observar en la (figura 123).

**Figura 123.** Proceso de extracción de características para detección de líneas en la vía



**Fuente.** Elaboración propia

En primera instancia, el sistema recibe las señales captadas por la cámara, las cuales se describen como la proyección bidimensional de las imágenes tomadas en un instante de tiempo. Cada imagen tomada por el sistema es sometida al proceso de extracción de características, es decir, el algoritmo desarrolla el reconocimiento de estos parámetros en tiempo real.

La obtención de características, continúa el proceso mediante la conversión de la imagen a un solo canal de color, es decir, a escala de grises, como se puede observar en la (figura 124), ya que esto permite mejorar el rendimiento en relación a procesamiento de máquina y simplifica los cálculos necesarios para desarrollos más complejos. Sobre esta imagen se continúa el desarrollo de aspectos aplicados al reconocimiento de líneas, para ello el algoritmo desarrolla seguidamente la detección de contornos de la imagen, entendiéndose contorno como la variación del nivel de intensidad entre los píxeles cercanos, es decir, pretende reconocer la región de los objetos presentes en una misma imagen.

**Figura 124.** Imagen de la vía a escala de grises



**Fuente.** Elaboración propia

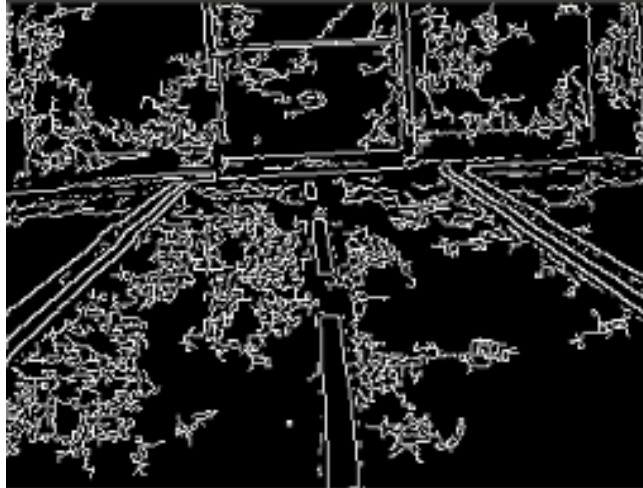
Para la localización de los contornos se utilizó el operador *Canny*, el cual a diferencia de otros métodos comúnmente utilizados, permite caracterizar de forma definida los contornos evitando la ruptura de los bordes, es decir, define mayor cantidad de bordes presentes en la imagen. La efectividad de este operador radica en que utiliza la derivada de Gauss, lo cual garantiza que la imagen sea menos vulnerable al ruido, permitiendo un mejor reconocimiento del contorno, el operador se define mediante la ecuación (84)

$$\nabla(f(x, y) * h_o(x, y)) = f(x, y) * \nabla(h_o(x, y)) = f(x, y) * DroG(x, y) \quad (84)$$



Donde  $DroG$  es una derivada suavizada, el resultado de la implementación del operador Canny se observa en la (figura 125)

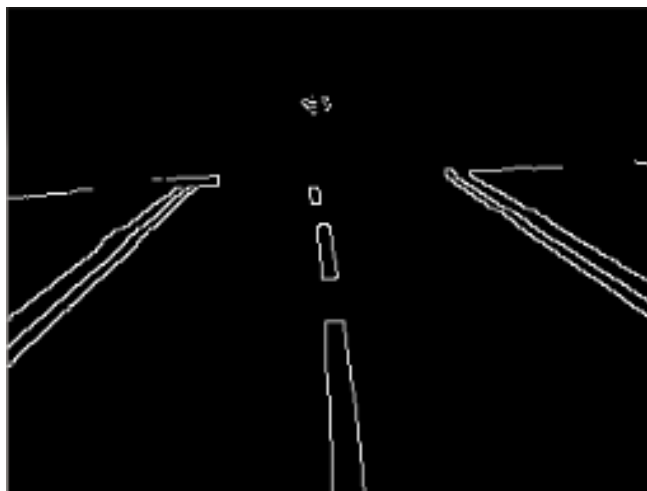
**Figura 125.** Detección de contornos de la vía con operador Canny



**Fuente.** Elaboración propia

Como se puede observar en la (figura 125), las tareas de limitación de regiones demarcan zonas que no son de interés, de manera tal que se aplicó el método de ajuste de umbral (*threshold*), que basan su procesamiento en un nivel de segmentación más robusto, demarcando los elementos que pertenecen a un área y elimina los que se encuentran por fuera de él, los rangos del umbral fueron predeterminados de acuerdo a las características del ambiente, tal como se expone en la (figura 126).

**Figura 126.** Ajuste del umbral de la imagen



**Fuente.** Elaboración propia

La finalidad de la ajustar los límites del umbral es analizar la pertinencia de los píxeles existentes de la imagen, denotando los valores que son de interés. Dado a que el proceso se realiza sobre una imagen binarizada (0 y 1) se realiza la comparación de los rangos del nivel de gris de la imagen " $g$ ", respecto al rango preestablecido del umbral " $U$ ". La pertinencia de un pixel se puede determinar mediante la ecuación (85), en el que  $Tg$  representa el valor resultante del cálculo de pertinencia del píxel.

$$Tg = \begin{cases} 0 & \text{si } g < U \\ 1 & \text{si } g > U \end{cases} \quad (85)$$

Finalmente, una vez delimitada las características propias de la imagen, se implementa la transformada de Hough, el cual es un algoritmo que extrae características de alto nivel como puede ser líneas, elipses u otro tipo de curva parametrizada<sup>90</sup>. La ejecución de este algoritmo requiere una imagen previamente parametrizada mediante detección de bordes.

Este desarrollo contempla el uso de la transformada para la localización de líneas rectas en la imagen. De manera tal que en primera instancia se considera un pixel de coordenadas en el espacio  $(X_i, Y_i)$ , representado como un borde por el que pasa una recta definida por la ecuación (86)

$$y_i = ax_i + b \quad (86)$$

Realizando la transformación en el espacio paramétrico (figura 127), las infinitas rectas que pasan por el punto coordinado corresponden a infinitos valores de  $b$ , por tanto se puede definir la ecuación (87)

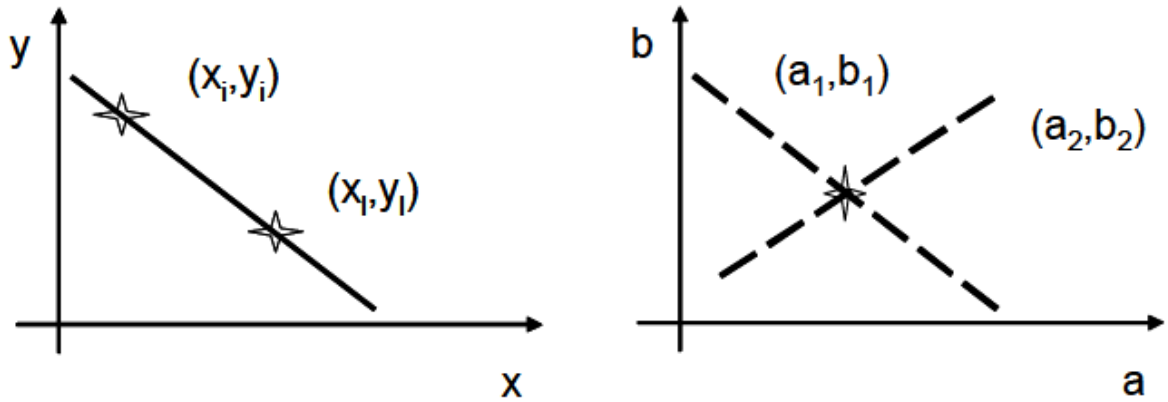
$$b = -ax_i + y_i \quad (87)$$

De manera tal que si dos píxeles corresponden a la misma línea recta, son representados en el espacio paramétrico.

---

<sup>90</sup> Apuntes de visión Artificial. Segmentación, detección de líneas rectas. p.153.

**Figura 127.** Conversión hacia el espacio paramétrico



**Fuente.** Apuntes de visión Artificial. Segmentación, detección de líneas rectas. Departamento de electrónica, Automática e Informática industrial.

Mediante la implementación de esta transformada es posible observar el resultado de la detección de líneas de la vía (figura 128), evidenciando un alto grado de precisión considerando que son imágenes captadas por video, la línea roja representa la línea de la vía lo que permite tener una mejor identificación el área navegable, generando un respaldo en la conducción e introduciendo un desarrollo para aplicaciones de control futuras.

**Figura 128.** Detección de líneas de la vía



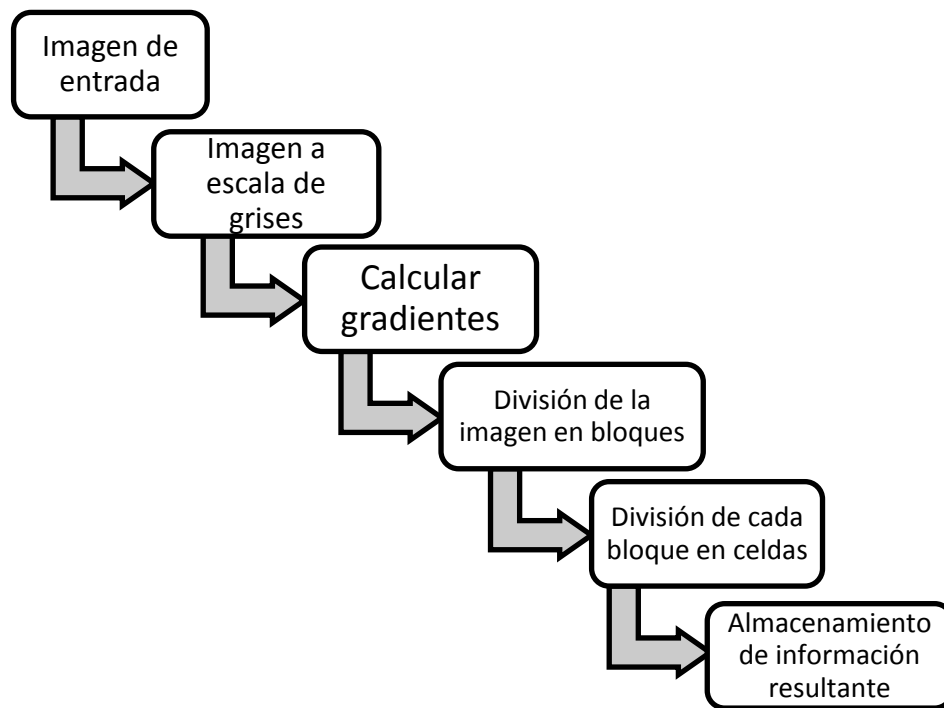
**Fuente.** Elaboración propia

## 5.8.2 Reconocimiento de peatones

En la actualidad, muchas personas pierden la vida en accidentes de tránsito, de manera tal que se implementó un algoritmo el cual permitiese identificar un peatón en la vía, con el fin de generar asistencia a la conducción buscando reducir las cifras de accidentalidad. El desarrollo del algoritmo se basa en el uso de librerías de OpenCV útiles para el procesamiento de los parámetros en identificación de figuras humanas.

La técnica utilizada para la localización de los peatones se basa en los histogramas de gradientes orientados (Histogram of Oriented Gradients (HOG)). Se trata de un método capaz de realizar detección de personas en secuencias de video en tiempo real, este método fue implementado por Navneet Dalal y Bill Triggs en el Instituto Nacional de Investigación en Informática y Automática en 2005<sup>91</sup>. La técnica consiste en generar un contorno sobre la figura de interés, para lograr ello se tiene en cuenta el proceso expuesto en la (figura 129).

**Figura 129.** Proceso de extracción de características para detección de peatones



**Fuente.** Elaboración propia

---

<sup>91</sup> N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Computer Vision and Pattern Recognition, 2005. vol. 1, p. 886–893.

La elección de este método radica en que proporciona una gran robustez frente a condiciones de iluminación, cambios de contorno y escalas de la imagen a detectar. En primera instancia el HOG determina la imagen proveniente del sistema de visión representado por la cámara de video.

De acuerdo a ello transforma la imagen a escala de grises con el fin de operar con mayor facilidad los cálculos para la detección de características. La complejidad del algoritmo radica en que después de ello se calcula la dirección de los valores del gradiente y la magnitud a lo largo de los ejes  $(x, y)$ , representada en las ecuaciones (88) y (89) respectivamente. El operador gradiente aplicado sobre un pixel, retorna un valor el cual indica el nivel de luminosidad presente en la imagen.

$$\theta = \tan^{-1} \frac{I_x}{I_y} \quad (88)$$

$$|G| = \sqrt{I_x^2 + I_y^2} \quad (89)$$

Posterior a ello se procede con la división de la imagen en bloques, de un área determinada, cada bloque se compone de una cantidad de pixeles determinado. De la misma forma cada bloque se subdivide en regiones más pequeñas denominadas celdas, en las que el algoritmo determina el histograma de gradientes orientados de cada una, ordenándose en vectores con valores definidos (figura 130).

De manera tal que se considera una imagen  $A$  de dimensiones  $M \times N$ , un tamaño de bloques  $B_M \times B_N$  y un tamaño de celda  $C_M \times C_N$ . En primera instancia se define el ancho y el alto de la imagen en celdas  $M_C \times N_C$  definidos en la ecuación (90)

$$M_C = \frac{M}{C_M}, \quad N_C = \frac{N}{C_N} \quad (90)$$

El número de bloques distribuidos en la imagen  $N_{B_M}$  y  $N_{B_N}$  se definen en la ecuación (91)

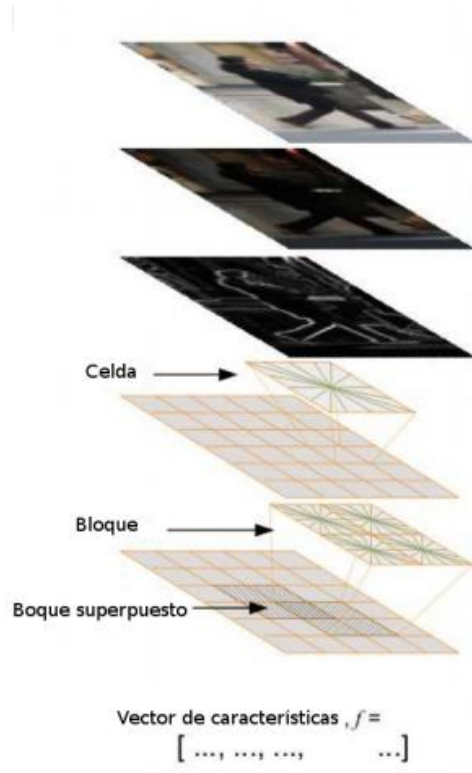
$$N_{BM} = 1 + M_C - B_M, \quad N_{BN} = 1 + N_C - B_N \quad (91)$$

De manera tal que el número total de celdas  $T_C$  y bloques  $T_B$  se expresan en la ecuación (92)

$$T_C = M_C * N_C, \quad T_B = N_{BM} * N_{BN} \quad (92)$$

En el proceso de caracterización del HOG (figura 130), cuando se calculan las orientaciones de los píxeles en la imagen, los autores recomiendan normalizar el resultado, es decir, establecer el parámetro según se requiera la aplicación. Es por ello que se utilizan los descriptores HOG los cuales realizan la identificación en base al clasificador SVM, el cual consiste en un módulo de aprendizaje y clasificación de acuerdo a parámetros establecidos (en el algoritmo desarrollado se utilizaron los parámetros por defecto del clasificador proporcionados por OpenCV).

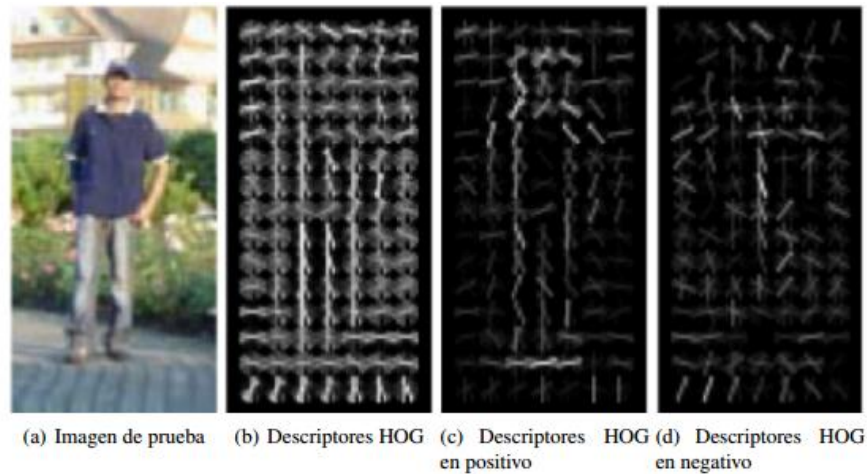
**Figura 130.** Desarrollo del algoritmo en imagen de referencia



**Fuente.** Txarterina, Jon. Detección de personas. Universidad del país Vasco

El descriptor HOG proporciona la información de los cambios de intensidad, permitiendo reconocer las fronteras de un objeto. La extracción de las características del descriptor HOG se expone en la (figura 131).

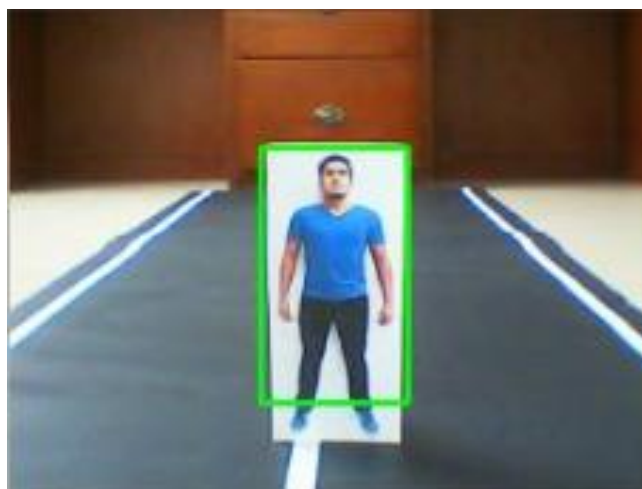
**Figura 131.** Desarrollo del descriptor HOG



**Fuente.** Txarterina, Jon. Detección de personas. Universidad del país Vasco

El cálculo del algoritmo HOG requiere un proceso computacional bastante alto, dado a que opera cada celda de la imagen, determinando las fronteras para su posterior reconocimiento. Teniendo en cuenta esta serie de parámetros, se implementó el algoritmo obteniendo como resultado la identificación de peatones en la vía, delimitándolas dentro de un cuadro de color verde tal como se observa en la (figura 132).

**Figura 132.** Detección de peatón en tiempo real

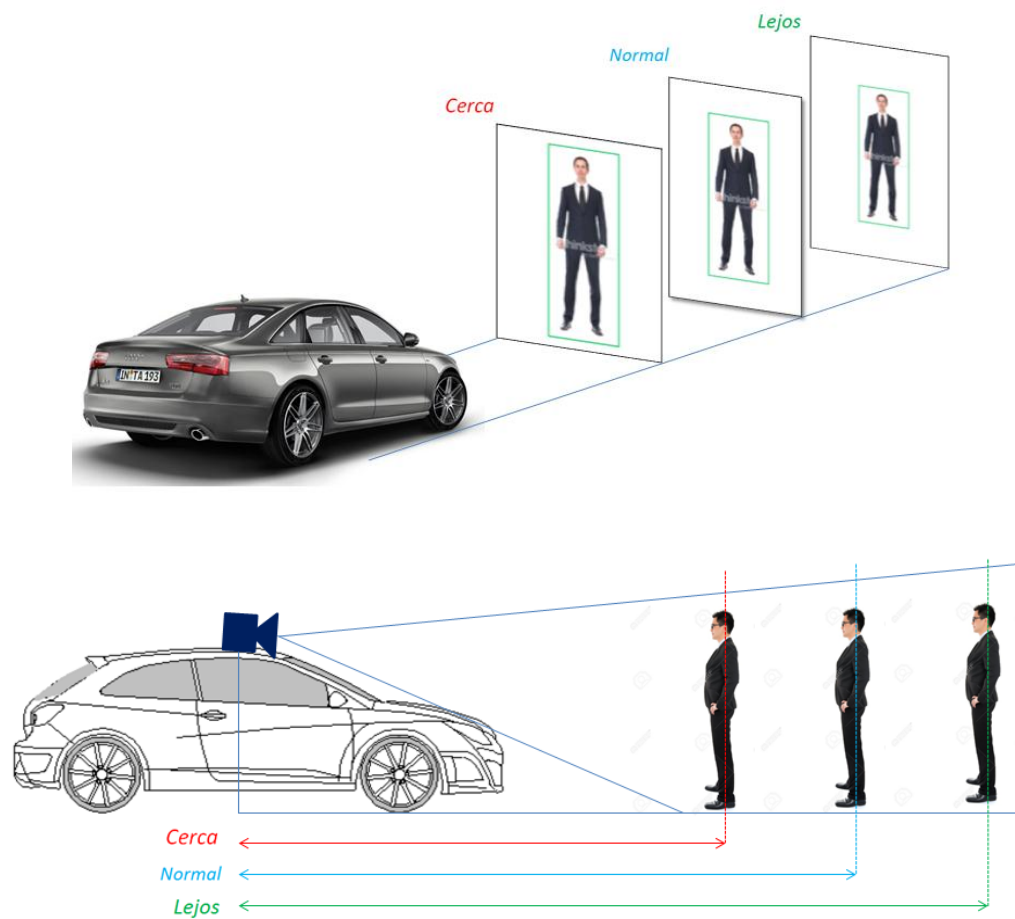


**Fuente.** Elaboración propia

Dado a que el algoritmo desarrollado, busca generar futuros métodos de apoyo en la conducción, se implementó mediante el sistema de control, una aplicación que permita detener el vehículo en caso de que la persona esté muy cercana al mismo.

Para ello se generaron tres distancias de referencia (figura 133), las cuales se establecieron de acuerdo al tamaño del recuadro que contiene la persona, es decir, el tamaño del recuadro indica la cercanía del vehículo respecto al peatón, estableciendo que a mayor tamaño del recuadro, menor será la distancia que separa a ambos.

**Figura 133.** Distancias de detección del peatón



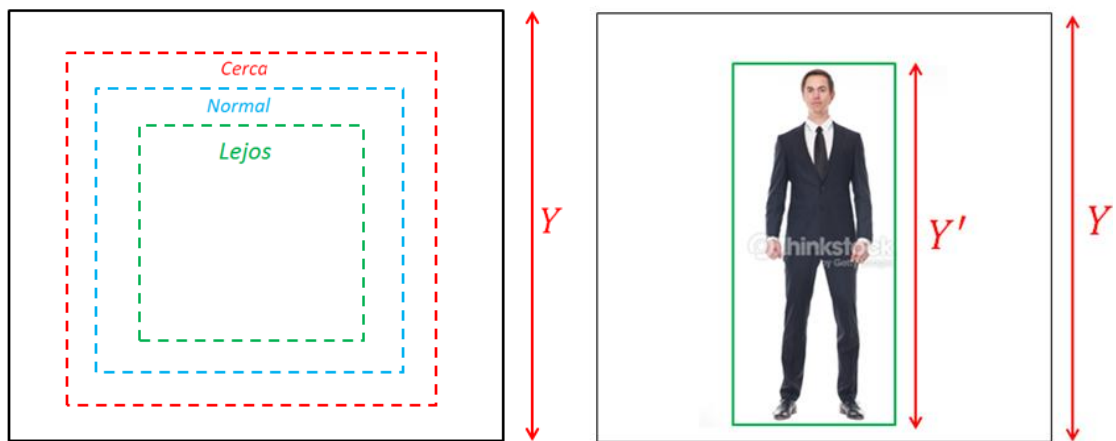
**Fuente.** Elaboración propia

Los rangos de detección se determinaron de acuerdo a medidas tomadas en el ambiente predeterminado. De acuerdo a ello se establecieron tres dimensiones



dentro del recuadro total definido como el área visual captada por la cámara (figura 134a), de manera tal que la acción de detención del vehículo se desarrolla cuando la dimensión de altura del recuadro que contiene el peatón representado por  $Y'$ , excede los parámetros de dimensión establecidos del recuadro denominado “cerca” indicando que la distancia entre el vehículo y el peatón es demasiado próxima (figura 134b). Adicional a ello se establece que la velocidad del vehículo disminuirá conforme a la cercanía del peatón, esto se obtuvo relacionando la distancia  $Y'$  respecto a la señal PWM correspondiente a la velocidad, de manera tal que el vehículo se detendrá progresivamente de acuerdo al aumento en la dimensión del recuadro que contiene al peatón.

**Figura 134.** Detección de peatón en tiempo real



**Fuente.** Elaboración propia

## 5.9 PROGRAMACIÓN SEGUNDA APLICACIÓN

La programación de la segunda aplicación contempla el desarrollo del algoritmo para la detección de líneas en la vía y la detección de peatones en tiempo real, de igual manera la programación se desarrolla en QT Creator® para el sistema embebido.

A continuación se muestra el pseudocódigo correspondiente a la programación del pcDuino.

**Programa:** Segunda aplicación

**Algoritmo:**

Incluir librerías de OpenCV

Incluir librerías para comunicación serial

Incluir librerías para creación de interfaz grafica  
Incluir librería HOG

Definición de variables globales  
Definir comunicación serial con arduino  
Establecer condición de enlace  
Definir el puerto serial de comunicación  
Función botón on\_serial

```
Si -> el estado del puerto = 0;  
    Crear objeto de tipo serial  
    Asignar puerto correspondiente  
    Abrir la comunicación de lectura y escritura  
Si -> no se detecta comunicación serial  
    Emitir mensaje de alerta  
    Cerrar comunicación existente  
    Sí No  
        Emitir mensaje de comunicación establecida  
        Asignar configuración de comunicación serie correspondiente  
Fin SI
```

Fin función botón on\_serial

Función botón off\_serial

```
Si botón off_serial es activo  
    cerrar comunicación serial  
    emitir mensaje puerto liberado
```

Fin función botón off\_serial

Función botón parar

```
Si botón parar es activo  
    Enviar PWM a arduino de detención del vehículo (stop)
```

Fin función botón parar

Función adelante

Enviar arduino velocidad constante

Fin función adelante

Función salir

Cerrar ventanas graficas existentes

Fin función salir

Función Speed

Enviar arduino PWM de activación del Speed

Fin función Speed

Función aplicación

```
Capturar la dirección de la cámara USB
```

Redimensionamiento de la imagen  
Declaración de variables  
Clonar imagen original  
Declarar HOGDescriptor(reconocimiento de persona)

Mientras (1)

    Aplicación -> activo  
    Lectura de imagen redimensionada  
    Crear dos ventanas graficas  
    Transformar imagen RGB a escala de grises (GRAY)  
    // Detección de líneas  
    Aplicación del método Canny para detección de contornos  
    Definición de umbral correspondiente  
    Definición de vectores  
    Declaración de HoughLines  
    SI (tamaño i = 0 < tamaño de línea)  
        Declarar variables  
        Hallar sen y cos de variables  
        Dimensionar los puntos verticales  
        Dimensionar los puntos horizontales  
        Demarcar línea de acuerdo a orientación de los puntos

    SI NO

        Demarcar línea en puntos predefinidos

    Fin SI

    // Detector de personas  
    Declarar detección hog  
    Declaración de vectores  
    Dimensionar elemento  
    Para (tamaño de recuadro < tamaño de persona)  
        Incrementar tamaño de cuadro  
        Demarcar recuadro de color verde  
    Para (recuadro > tamaño de persona)  
        Dimensión en eje x  
        Dimensión en eje y  
        Demarcar recuadro de color verde  
    Enviar arduino alto del recuadro  
    Enviar velocidad  
    Enviar dirección (centro)  
    Graficar en primera ventana HOG  
    Graficar en segunda ventana detección de líneas

Fin mientras

Función reset

Cerrar aplicación

Fin función reset

Fin aplicación

Fin de programa

En el *Anexo E* se expone el diagrama de flujo del programa segunda aplicación.

Dado que el Arduino es el controlador esclavo, también se encuentra programado para recibir las señales provenientes del pcDuino. El código implementado tiene por objeto recibir las variables de la dimensión del cuadro que contiene el peatón y las variables proporcionadas por los sensores ultrasónicos. De manera tal que a una distancia preestablecida el vehículo se detendrá para evitar chocar con el obstáculo, ya sea por la lectura de la cámara o la distancia emitida por los sensores. A continuación se muestra el pseudocódigo correspondiente a la programación de Arduino para la segunda aplicación.

**Programa:** Segunda aplicación (Arduino)

**Algoritmo:**

Incluir librería de servo

Incluir EEPROM

Declaración de variables

Definir comunicación serial

Definición de puertos de entrada y salida

Definición de comunicación para variables internas

Definir los trigger iniciales para ultrasónicos

Tiempo de lectura de los sensores

Adquisición de medida y conversión a cm

Leer puerto serie

Definir variables cm

    Si (distancia de sensor 1 <15cm)

        Detención del vehículo (stop)

    Fin SI

    Si (distancia de sensor 2 <15cm)

        Detención del vehículo (stop)

    Fin SI

    Si (tamaño del recuadro > valor predefinido)

        Detención del vehículo (stop)

    Fin SI

    Si (velocidad está entre los rangos de lectura)

        Mapear valor de entrada respecto al PWM de activación del speed

        enviar valor mapeado

    SI (dimensión del recuadro < valor predefinido)

        Mapear valor de recuadro respecto a velocidad

        Enviar valor mapeado

    Fin SI

Fin de programa

En el *Anexo F* se expone el diagrama de flujo del programa segunda aplicación (Arduino)

## 5.10 PROGRAMACIÓN CONTROL MANUAL

Una tercera aplicación, es la implementación del control manual sobre el vehículo, esto con el fin de corroborar los movimientos básicos como lo son velocidad y dirección. Además de ello con esta aplicación alternativa se puede controlar el vehículo y guiarlo al lugar que se desee. A continuación se muestra el pseudocódigo correspondiente a la programación del pcDuino.

**Programa:** control manual

**Algoritmo:**

Incluir librerías de OpenCV

Incluir librerías para comunicación serial

Incluir librerías para creación de interfaz grafica

Definición de variables globales

Definir comunicación serial con arduino

Establecer condición de enlace

Definir el puerto serial de comunicación

Función botón on\_serial

Si -> el estado del puerto = 0;

    Crear objeto de tipo serial

    Asignar puerto correspondiente

    Abrir la comunicación de lectura y escritura

Si -> no se detecta comunicación serial

    Emitir mensaje de alerta

    Cerrar comunicación existente

    Sí No

        Emitir mensaje de comunicación establecida

        Asignar configuración de comunicación serie correspondiente

    Fin SI

Fin función botón on\_serial

Función botón off\_serial

    Si botón off\_serial es activo

        cerrar comunicación serial

        emitir mensaje puerto liberado

    Fin SI

Función aplicación

    Función control manual

    Activar estado de comunicación serial

```
Fin función control manual
Asignación de sliders
Slider de velocidad
Slider de dirección
Si (slider es activo)
    Enviar arduino valor de velocidad y dirección correspondiente
Fin Si
Fin aplicación
Fin de programa
```

En el *Anexo G* se expone el diagrama de flujo del programa Control manual, realizado en el pcDuino.

La programación de arduino, es correspondiente con el código de la primera aplicación para la primera trayectoria.

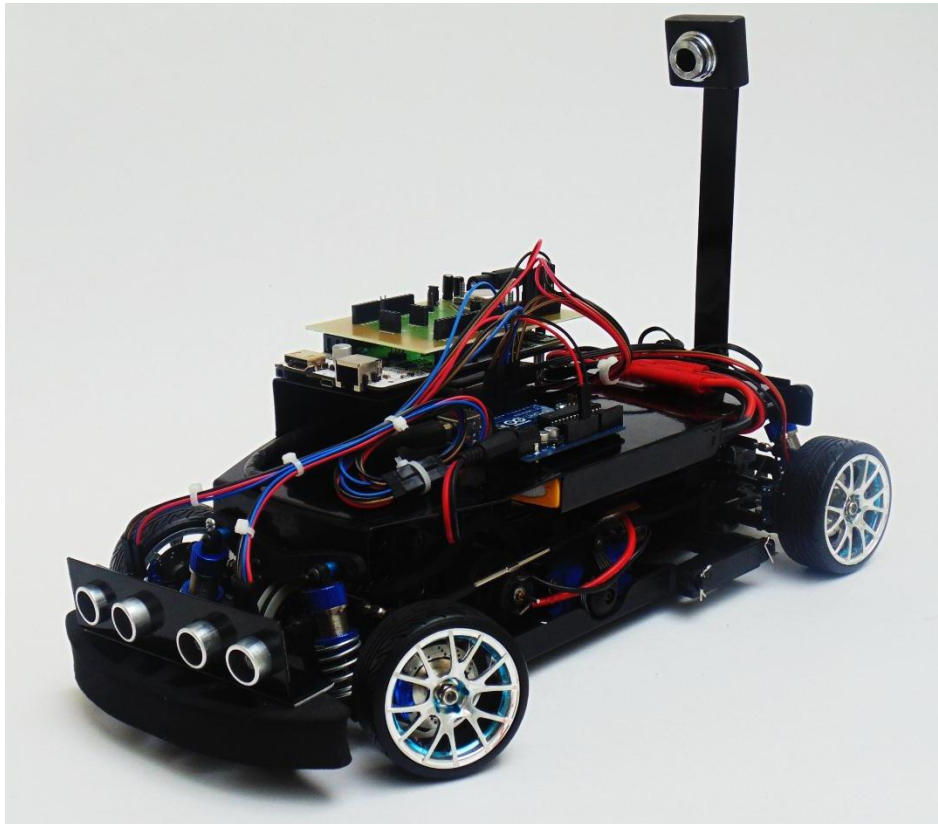
## 6 INTEGRACIÓN Y RESULTADOS

En este capítulo se exponen los resultados obtenidos de la implementación tecnológica realizada para la plataforma en escala (1:10), de igual manera se desarrollaran las pruebas de las aplicaciones propuestas desde un principio; seguimiento de trayectorias en un ambiente predeterminado (considerándolo libre de obstáculos) y reconocimiento de peatones y líneas en la vía.

### 6.1 ESTRUCTURA MECÁNICA

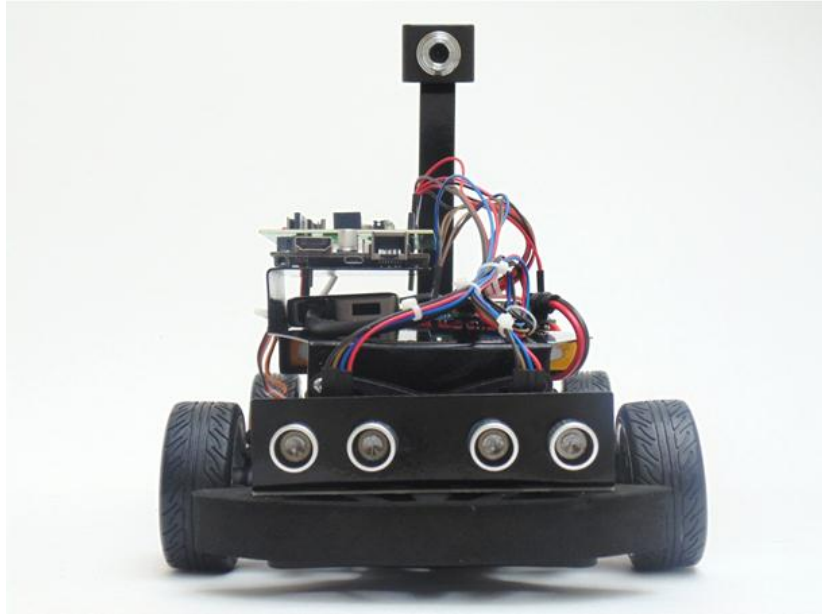
La plataforma utilizada para el proyecto está basada en un vehículo en escala reducida (1:10) con características aproximadas a un vehículo comercial. Se implementó el carro 1/10th SCALE RC 4WD HIGH PERFORMANCE CAR (figura135, 136), dado que su estructura mecánica cumple con los requerimientos funcionales y cinemáticos necesarios para realizar las pruebas de las aplicaciones propuestas.

*Figura 135. Plataforma adoptada*

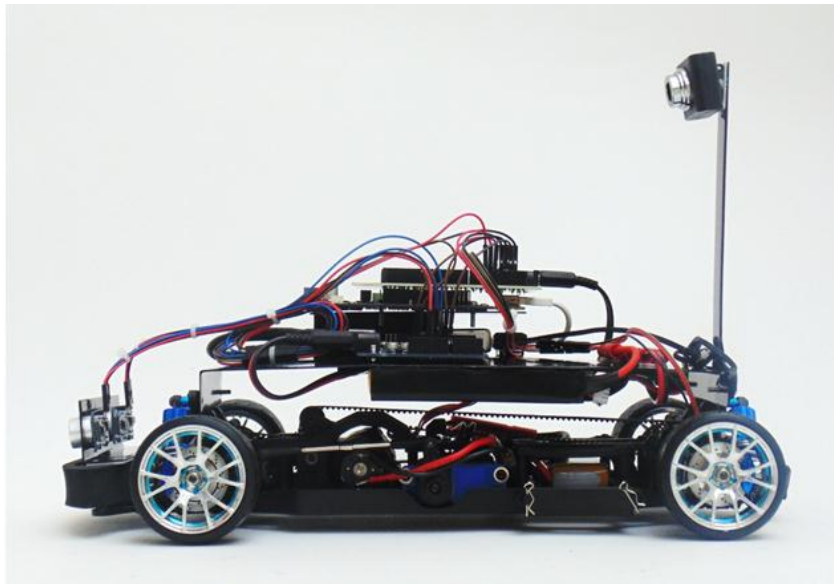


*Fuente. Elaboración propia*

**Figura 136.** Plataforma adoptada (vistas)



*a. Vista frontal*



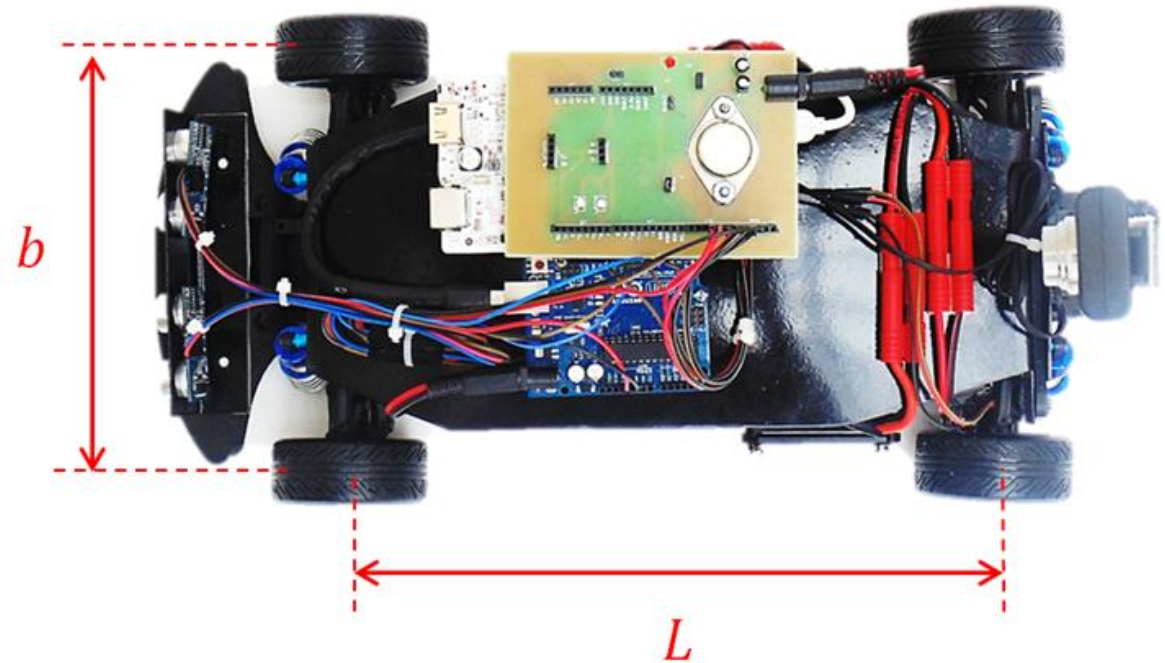
*b. Vista Lateral*

**Fuente.** *Elaboración propia*

En la (tabla 10) se muestran las principales características dimensionales de la plataforma adoptada (figura 137), las cuales fueron tomadas en cuenta para la realización de los respectivos sistemas de control.



Figura 137. Vista superior



Fuente. Elaboración propia

Tabla 10. Características dimensionales de la plataforma

Separación longitudinal entre las ruedas (L)	26 cm
Separación lateral entre las ruedas (b)	17.5 cm
Ángulo de dirección de ruedas delanteras ( $\delta_o$ ) ( $\delta_i$ )	-30 grados a 30 grados
Diámetro de rueda	6 cm

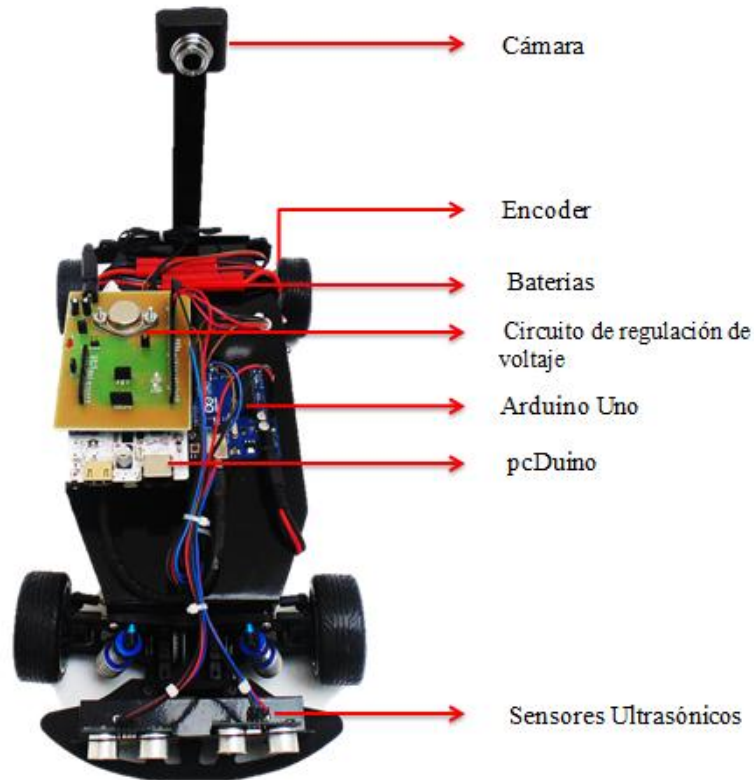
Fuente. Elaboración propia

### 6.1.1 Componentes de la plataforma

La plataforma está compuesta por una serie de sistemas electrónicos los cuales permiten la actuación de movimiento del vehículo y la comunicación con el sistema remoto (computador). Como se puede observar en la (figura 138), la parte correspondiente al sistema de fuente de energía, sensor de velocidad y sistema

visual se encuentran ubicados en la parte trasera. El sistema de procesamiento se encuentra ubicado en la parte central del vehículo y la sensorica de proximidad se dispone en la parte frontal.

**Figura 138.** Componentes electrónicos de la plataforma

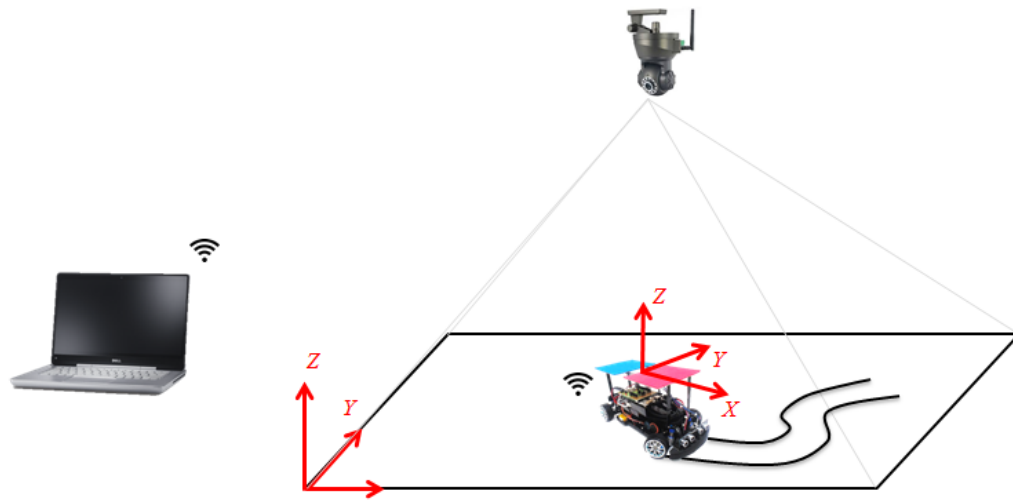


**Fuente.** Elaboración propia

## 6.2 PRIMERA APLICACIÓN

Se presenta el desarrollo de algoritmos basados en el control de dirección y desarrollos de visión computacional aplicables para un vehículo autónomo terrestre en escala (1:10), con el objeto de realizar seguimiento de trayectorias (previamente estipuladas), considerándolas libre de obstáculos. Estos algoritmos son elaborados en base a las características cinemáticas del vehículo y un sistema de localización (desarrollado por medio de procesamiento de imagen). De esta manera se logró (figura139), que el vehículo se desplace de forma autónoma hacia el camino de puntos predefinidos realizando el trayecto deseado, observando la ejecución del movimiento y la información del sistema mediante la implementación de una interfaz gráfica.

**Figura 139.** Configuración primera aplicación



**Fuente.** Elaboración propia

En la (figura 140) se expone el área predeterminada de trabajo cuyas dimensiones son de (3.70x2.85m); seleccionada para la realización de las pruebas de seguimiento de trayectorias, La cámara está dispuesta a 2.5m a nivel del suelo y el foco de la cámara capta un área de (2.95 x1.65m).

**Figura 140.** Configuración primera aplicación (ambiente real)



**Fuente.** Elaboración propia

### 6.2.1 Configuración de la plataforma

Para la realización de las pruebas correspondientes a la primera aplicación, se adoptó en la plataforma la configuración presentada en la (figura 141), con el objeto de ubicar los marcadores de color de manera segura, para hacer efectivo el proceso de reconocimiento de los mismos y efectuar las trayectorias de manera correcta.

*Figura 141. Configuración de la plataforma para primera aplicación*



*Fuente. Elaboración propia*

### 6.2.2 Interfaz gráfica

La interfaz (figura 142) se compone en primera instancia del sistema de localización, el cual informa los datos de posición y orientación del vehículo, de igual forma cada vez que el vehículo pasa sobre una coordenada de la trayectoria, el programa ajusta el próximo punto de destino por el cual el vehículo debe pasar.

Por otra parte dentro de la información del sistema se observa el error de posición de la coordenada central del vehículo respecto al punto de destino de la trayectoria, el ángulo de dirección de las ruedas delanteras y el valor de PWM correspondiente a la velocidad del vehículo. Finalmente la interfaz ofrece la posibilidad de seleccionar la trayectoria de interés.

**Figura 142.** Interfaz primera aplicación



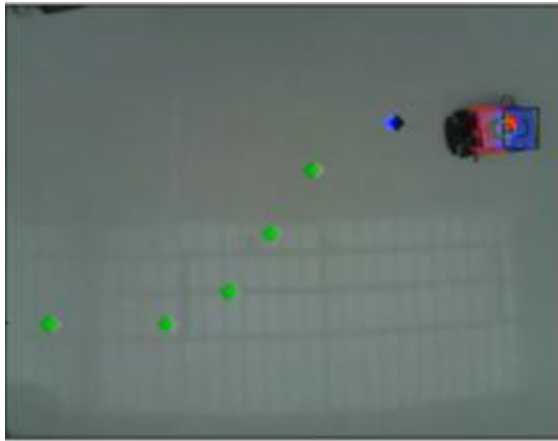
**Fuente.** Elaboración propia

### 6.2.3 Primera trayectoria

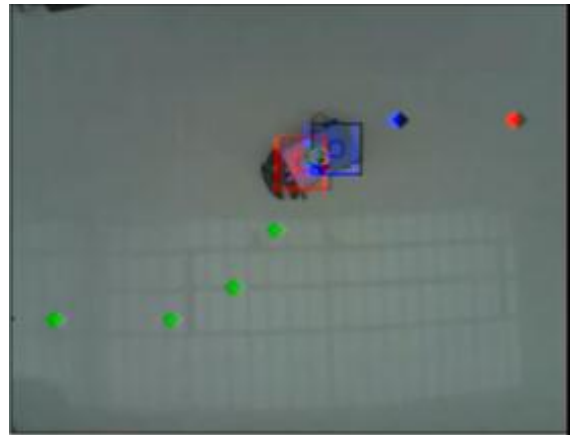
Para ejecutar el programa en primera instancia, en la interfaz se establece la comunicación entre los dispositivos (botón ON\_SERIAL), después de ello se procede a la activación del Speed Control (botón speed), posteriormente se selecciona la trayectoria (botón 1re trayectoria) y se procede al inicio de la aplicación (botón INICIO), como se puede observar en la (figura 143), la interfaz abre la ventana donde se observa el espacio coordenado de trabajo

El vehículo parte de la posición ( $X = 2.6$ ,  $Y = 1.2$ ,  $\theta = 180^\circ$ ) con  $\delta = 0$  (figura 143a) y realiza el recorrido a velocidad constante. Cada vez que pasa sobre un punto de la trayectoria, el siguiente destino se marca de color azul, y así seguidamente hasta que complete la trayectoria en su totalidad. De manera tal que el vehículo ajusta la dirección de las ruedas delanteras con el fin del pasar sobre los puntos que definen la trayectoria planeada, tal como fue propuesto en el control de dirección; lo que permite validar el algoritmo de seguimiento de trayectorias.

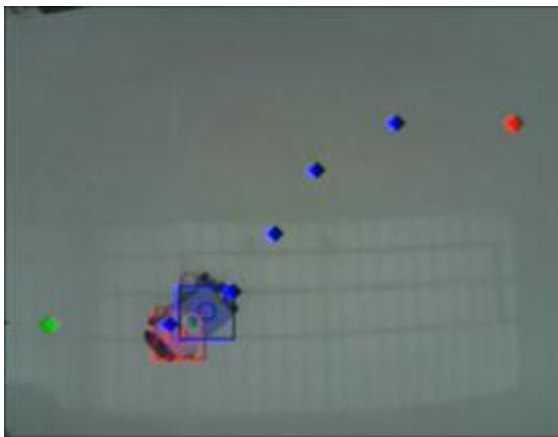
**Figura 143.** Ejecución trayectoria cambio de carril



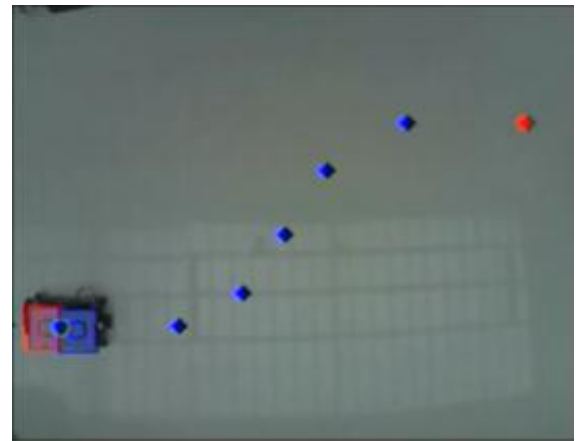
a. Toma 1



b. Toma 2



c. Toma 3



d. Toma 4

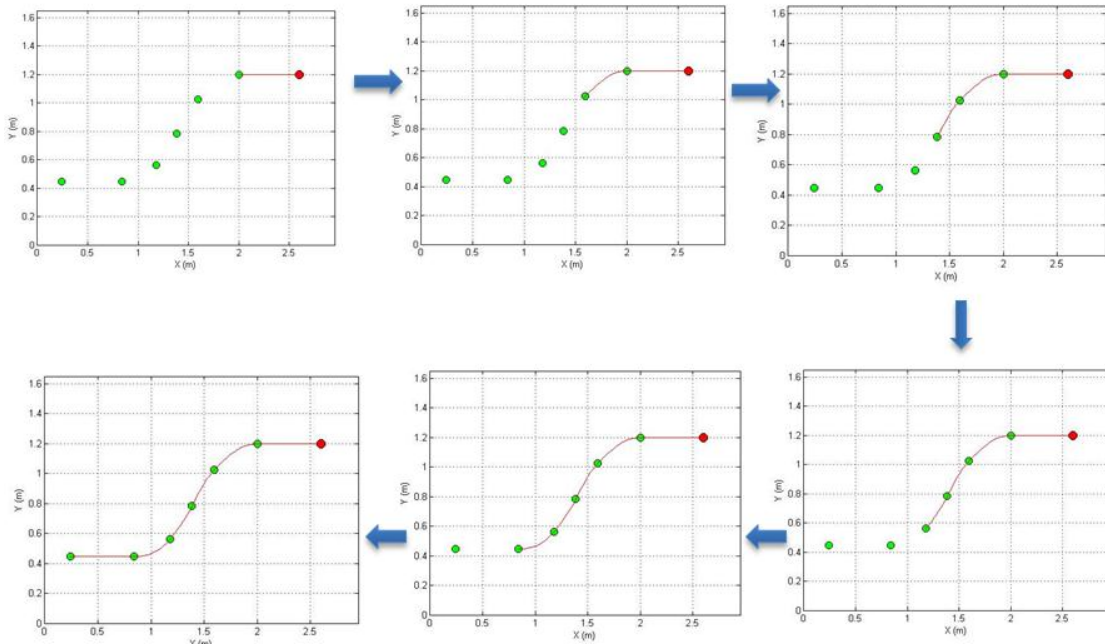
**Fuente.** Elaboración propia

De acuerdo al algoritmo de seguimiento de trayectorias implementado, es posible simular la trayectoria que describe el vehículo con las condiciones iniciales antes mencionadas (figura 144). Como se puede observar la trayectoria simulada corresponde al comportamiento real del vehículo, generando un error de posición de aproximadamente 0 – 3%. Ver video.

Video Implementación:

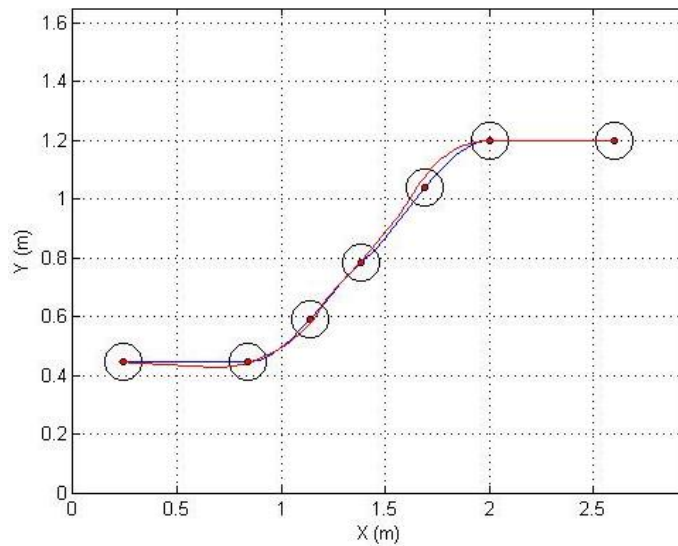
(<https://www.dropbox.com/s/d0rdbv2dlezzikn/VIDEO%20TESIS.mp4?dl=0>).

**Figura 144.** Trayectoria cambio de carril determinada mediante algoritmo



**Fuente.** Elaboración propia

**Figura 145.** Trayectoria Spline vs trayectoria simulada



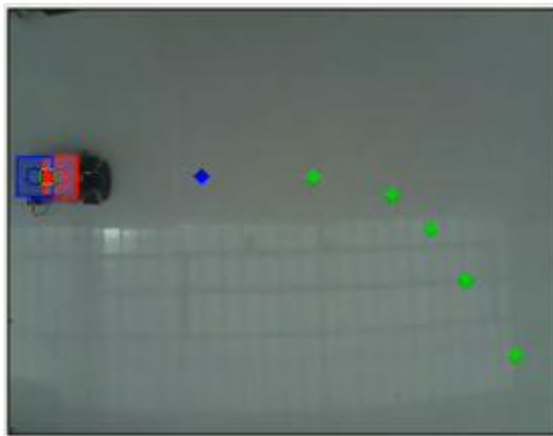
**Fuente.** Elaboración propia

Como se puede observar en la (figura 145), comparando la trayectoria Spline (color azul) vs la trayectoria generada por el control implementado (color rojo), las tendencias de aproximación son muy cercanas, de manera tal que el control implementado describe correctamente la trayectoria planeada por el vehículo.

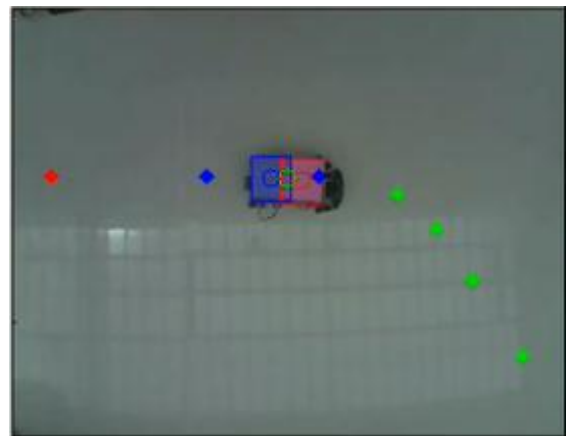
## 6.2.4 Segunda trayectoria

En la interfaz se establece la comunicación entre los dispositivos (botón ON\_SERIAL), después de ello se procede a la activación del Speed Control (botón speed), posteriormente se selecciona la trayectoria (botón 2da trayectoria) y se procede al inicio de la aplicación (botón INICIO).

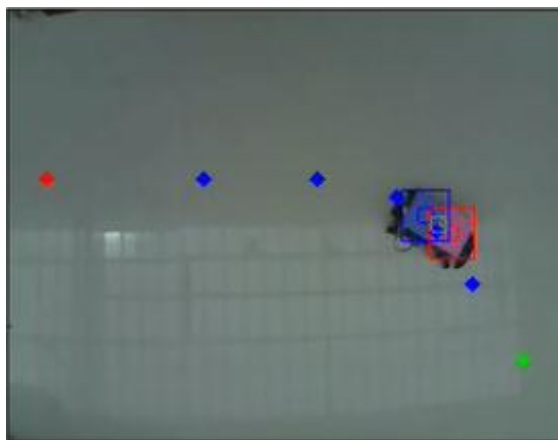
**Figura 146.** Ejecución trayectoria giro derecha



a. Toma 1



b. Toma 2



c. Toma 3



d. Toma 4

a. **Fuente.** Elaboración propia

Como se puede observar en la (figura 146) el vehículo parte de la posición ( $X = 0.2$ ,  $Y = 1$ ,  $\theta = 0^\circ$ ) con  $\delta = 0$  (figura104a), el vehículo describe la parte de la trayectoria de línea recta, a mayor velocidad (9 - 11 cm/s) y en las curvas describe una velocidad menor (4 - 5 cm/s), esto se logra mediante la implementación del encoder el cual junto con la programación de Arduino (PID)

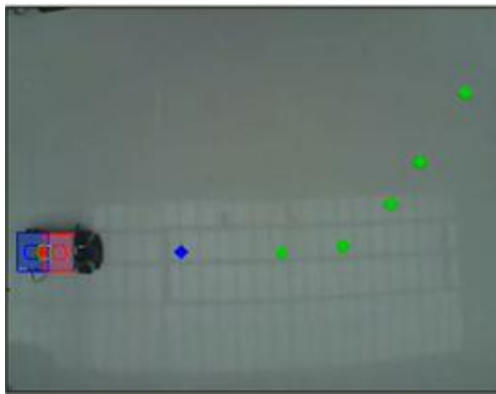


controlan la velocidad del vehículo dependiendo del grado de orientación de las ruedas delanteras. De manera tal que el algoritmo de seguimiento de trayectorias es aplicable en este caso únicamente para la dirección del vehículo, dado que ya no se asume velocidad constante. Por tanto se observa que el vehículo describe de manera correcta la trayectoria con un error de posición aproximadamente 0 – 4%

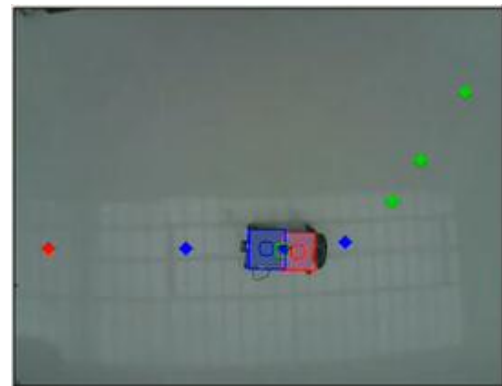
### 6.2.5 Tercera trayectoria

Se desarrolla de igual manera los pasos de activación para la ejecución de la aplicación, seleccionando para este caso (botón 3ra trayectoria). El vehículo describe los desplazamientos correspondientes.

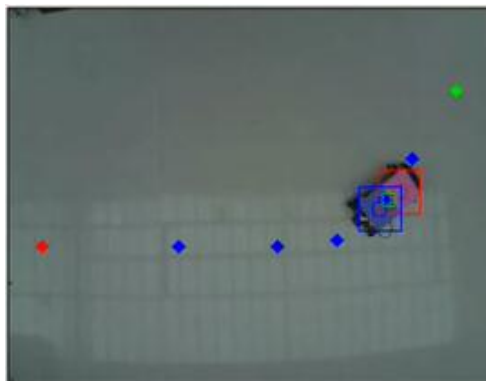
**Figura 147.** Ejecución trayectoria giro izquierda



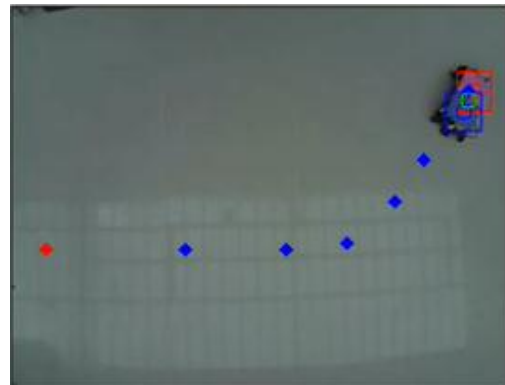
b. Toma 1



b. Toma 2



b. Toma 3



b. Toma 4

**Fuente.** Elaboración propia

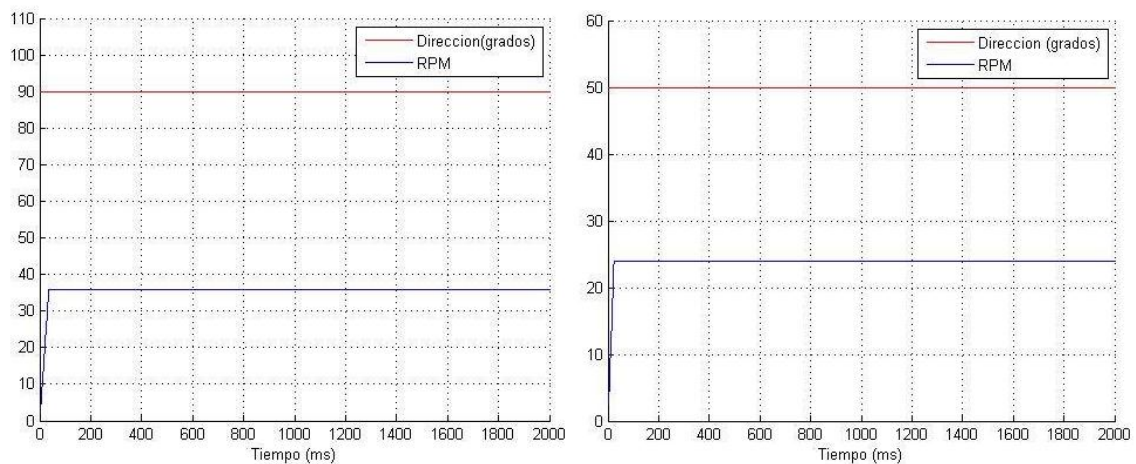
Como se puede observar en la (figura 147) el vehículo parte de la posición ( $X = 0.2, Y = 0.6, \theta = 0^\circ$ ) con  $\delta = 0$  (figura 105a), el cual describe un

comportamiento similar que en la segunda trayectoria, dado que de igual manera se controla la velocidad de acuerdo al ángulo de las ruedas delanteras, de manera tal que el vehículo describe los puntos que conforman la trayectoria de manera acertada validando el control implementado.

### 6.2.6 Control de velocidad

El control PID de velocidad toma como referencia la dirección de las llantas delanteras, de esta manera mediante la programación implementada, a  $90^\circ$  en el servomotor ( $\delta = 0$ ), el encoder debe captar una lectura de 36 RPM, de manera tal que el vehículo debe aumentar su velocidad, hasta alcanzar la lectura deseada, por tal, al momento de desarrollar una trayectoria en línea recta se describe una velocidad alta, como se muestra en la (figura 148a). De igual manera cuando el valor corresponde a los valores máximos de ángulo de giro ( $\delta = 30, -30$ ) que en el servomotor son ( $120^\circ - 50^\circ$ ) respectivamente, la lectura del encoder deberá corresponder a 24 RPM de manera tal que el vehículo en las curvas disminuye su velocidad tal como se observa en la (figura 148b).

**Figura 148.** RPM de acuerdo a dirección de las ruedas delanteras



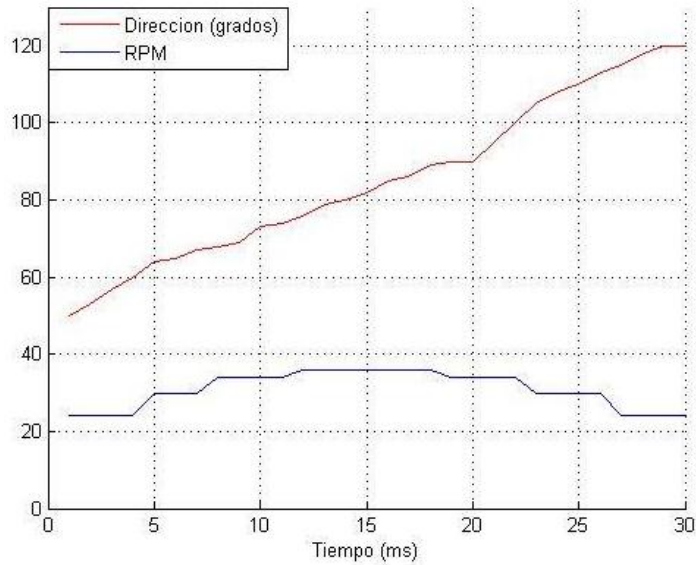
a. Dirección en línea recta

b. Dirección en curvas

**Fuente.** Elaboración propia

De igual manera como se observa en la (figura 149), con el fin de verificar que de acuerdo a la dirección de las llantas delanteras varíe la velocidad del vehículo, se realizó un muestreo en el que se giraron de extremo a extremo ( $50^\circ - 120^\circ$ ) la orientación de las llantas, de manera tal que al momento de llegar a la dirección central ( $90^\circ$ ), la velocidad corresponderá al máximo valor y será mínima en los giros pronunciados.

**Figura 149.** Velocidad del vehículo respecto al ángulo de dirección

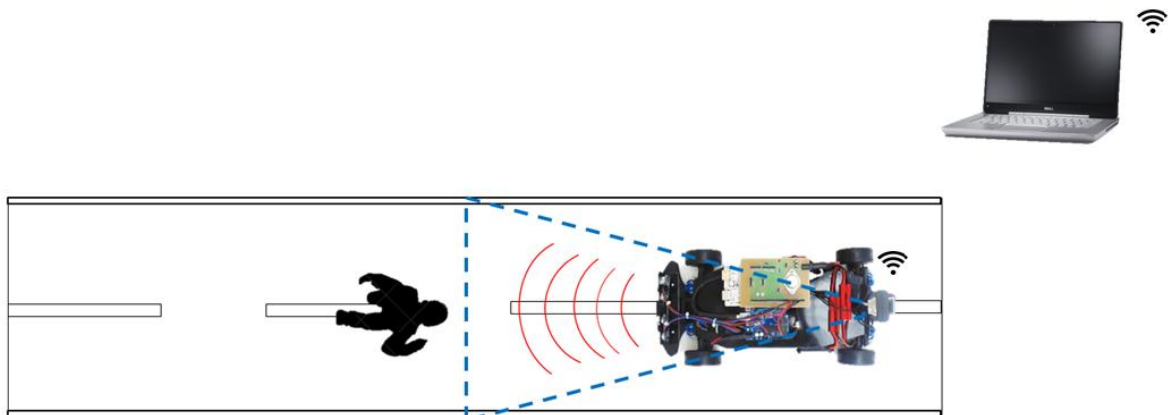


**Fuente.** Elaboración propia

### 6.3 SEGUNDA APLICACIÓN

Se expone el desarrollo de la segunda aplicación teniendo en cuenta los parámetros estipulados inicialmente, como lo son la detección de personas y el área navegable. De igual manera el vehículo se comunica de forma remota con el sistema de visualización, tal como se observa en la (figura 150).

**Figura 150.** Configuración segunda aplicación



**Fuente.** Elaboración propia

Para la óptima ejecución de la aplicación, se tuvo en cuenta que el vehículo está fabricado a escala (1:10) por lo cual se implementaron peatones simulados

mediante imágenes y una carretera de la misma escala. La simulación se desarrolló en un ambiente predeterminado como se ilustra en la (figura 151).

**Figura 151.** Escenario de implementación



**Fuente.** Elaboración propia

### 6.3.1 Configuración de la plataforma

Para la segunda aplicación se implementó la configuración ilustrada en la (figura 152), la cual tiene en cuenta sensores de ultrasonido en la parte frontal del vehículo como también una cámara instalada, que visualiza el frente del entorno de desarrollo de la conducción, asemejando la percepción visual de un conductor en la vida real.

**Figura 152.** Configuración segunda aplicación



**Fuente.** Elaboración propia

### 6.3.2 Interfaz gráfica

En la interfaz (figura 153) se proponen tres maneras de efectuar la visualización; detección de peatones (HOG), detección del área navegable (líneas) y la interacción de las dos aplicaciones (aplicación).

Una vez seleccionado el tipo de visualización que se desea, se observa un control de movimiento el cual desempeña la función de desplazamiento rectilíneo en dirección delantera (adelante) un paro manual (detener). Los dos recuadros (parte central) representan cada una de las aplicaciones cuando el sistema está en funcionamiento

**Figura 153.** Interfaz segunda aplicación



**Fuente.** Elaboración propia

### 6.3.3 Detección de líneas

La detección de líneas está orientada a la identificación del área navegable lo cual le permitirá al conductor identificar las posibilidades de área para el desarrollo de la conducción como se muestra en la (figura 154)

**Figura 154.** Detección del área navegable



**Fuente.** Elaboración propia

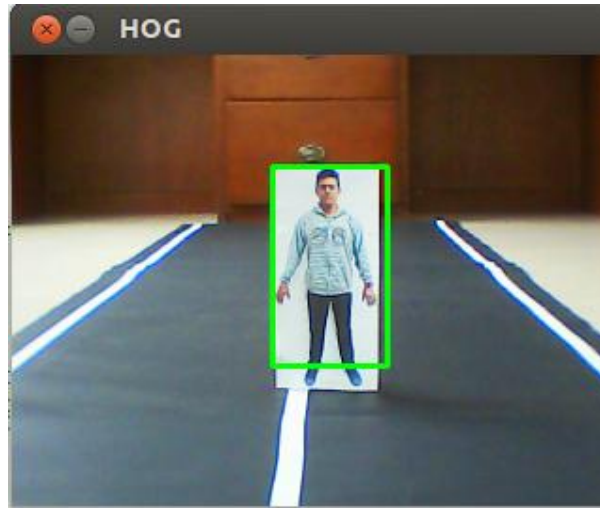
Como se observa en la (figura 154) las líneas de color rojo delimitan el área de conducción junto con las líneas de separación de carril, la detección de las líneas fluctúa de acuerdo a la intensidad de la luz del ambiente. En la conducción en ambientes reales tanto como en escenarios simulados es de vital importancia la demarcación precisa de áreas haciendo referencia a un contraste notorio de color como se aprecia en la imagen (blanco y negro) el área de visión de la cámara está enfocada netamente a la carretera ya que otras líneas que posean características ideales de detección en el ambiente también podrían ser visualizadas.

#### **6.3.4 Detección de peatones**

Esta aplicación está enfocada a la asistencia a la hora de efectuar la conducción, con el fin de salvaguardar la vida de los peatones y del conductor, tal como se propuso en el desarrollo, la aplicación identifica a los peatones y los encierra en un contorno, de acuerdo a este contorno (en unidades de píxeles) se implementa un control directo de velocidad hasta el punto de detener el vehículo por completo cuando el peatón está demasiado cerca.

En la (figura 155) se ilustra la identificación de un peatón y la manera de cómo se identifica en el espacio visual (cuadro de color verde).

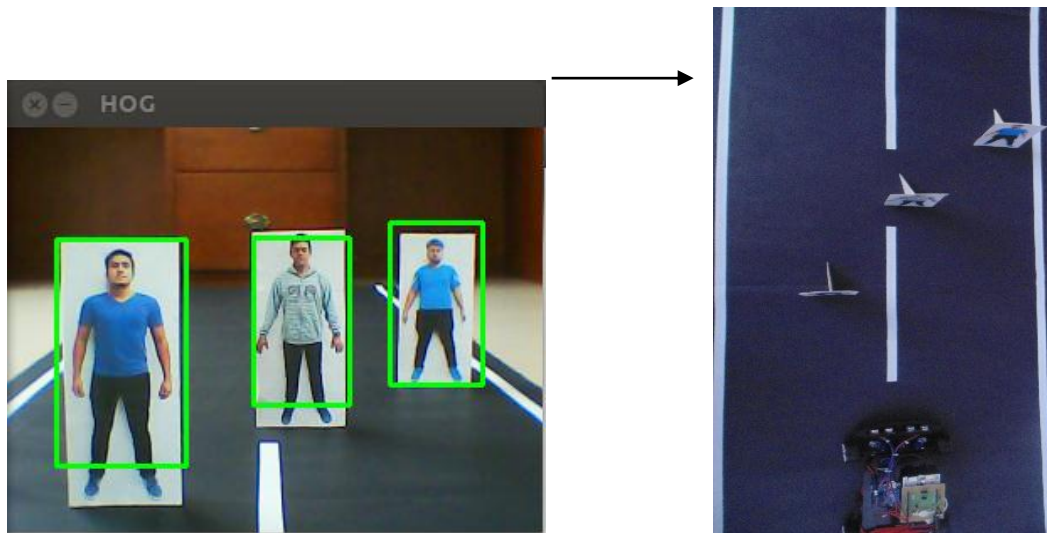
**Figura 155. Identificación de peatón**



**Fuente.** Elaboración propia

De igual manera también es aplicable para la identificación de múltiples peatones en la vía, sin embargo la detención del vehículo o el control de velocidad se efectúa bajo el contorno del peatón más cercano como se ilustra en la (figura 156).

**Figura 156. Identificación de múltiples peatones**

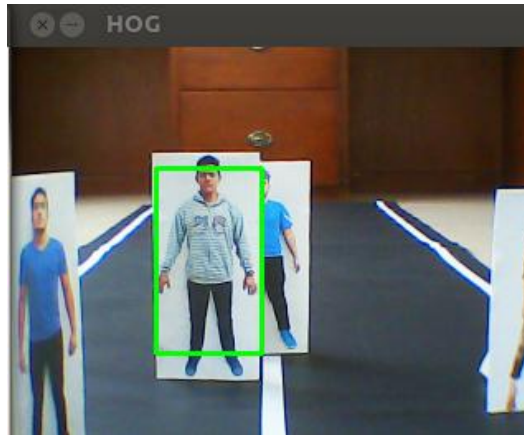


**Fuente.** Elaboración propia

El número de personas que son detectadas está determinado por la capacidad de visualización del entorno captado por la cámara, además de esto también influye la detección de cuerpo completo de la persona y la postura de la misma la cual está determinada por imágenes de referencia las cuales dictan la captura de las

extremidades del cuerpo humano, esto sustentado por el hecho que un peatón en movimiento está caracterizado por la identificación de estas, como se ilustra en la (figura 157).

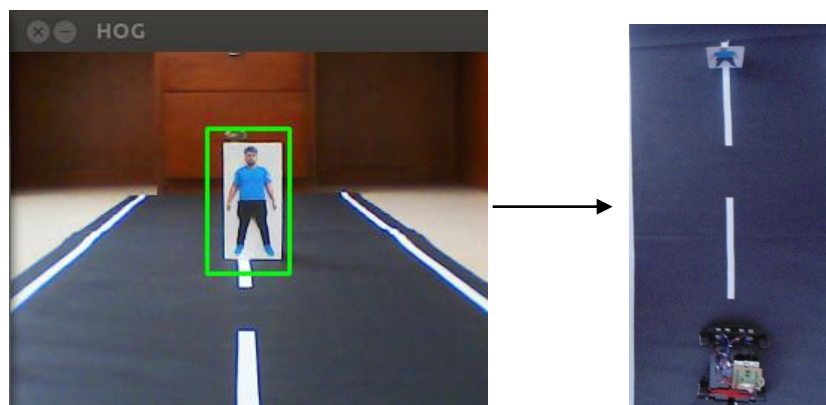
**Figura 157.** Identificación de múltiples peatones con error de posición



**Fuente.** Elaboración propia

El control de velocidad tiene como referencia el alto del contorno de la persona, para la detección de un contorno entre los valores de 96 a 140, siendo 96 un contorno de una persona a una distancia lejana y 140 a una distancia cercana. De acuerdo a ello se mapean estos valores de tal manera que un valor de contorno de 140 será un valor de PWM de 75 lo cual es la detención total del vehículo y un contorno de 96 será un valor de 135 lo cual hace referencia a un valor de PWM para una velocidad moderada alta. La detección de una persona a una distancia lejana se ilustra en la (figura 158) con lo cual corresponde un valor de PWM para velocidad moderada alta.

**Figura 158.** Contorno de persona a una distancia lejana

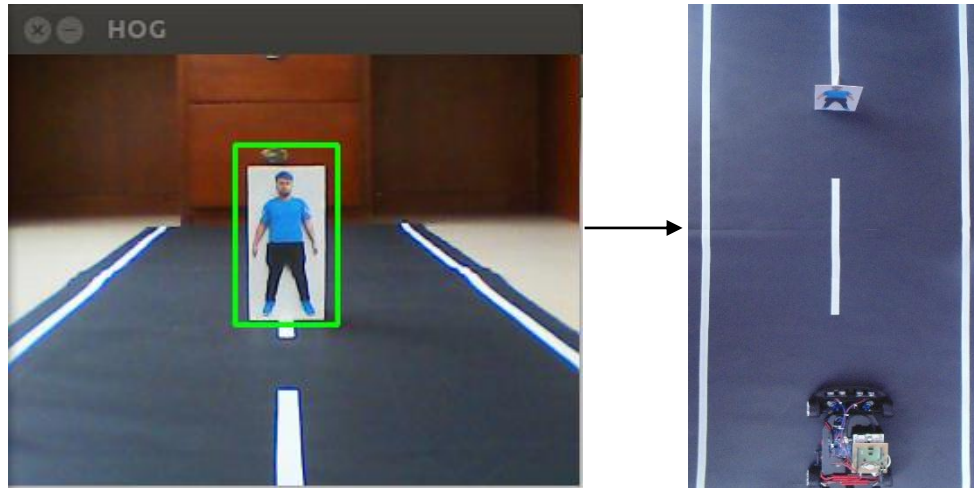


**Fuente.** Elaboración propia



La detección de una persona a una distancia media se ilustra en la (figura 159) con un valor de PWM para velocidad media

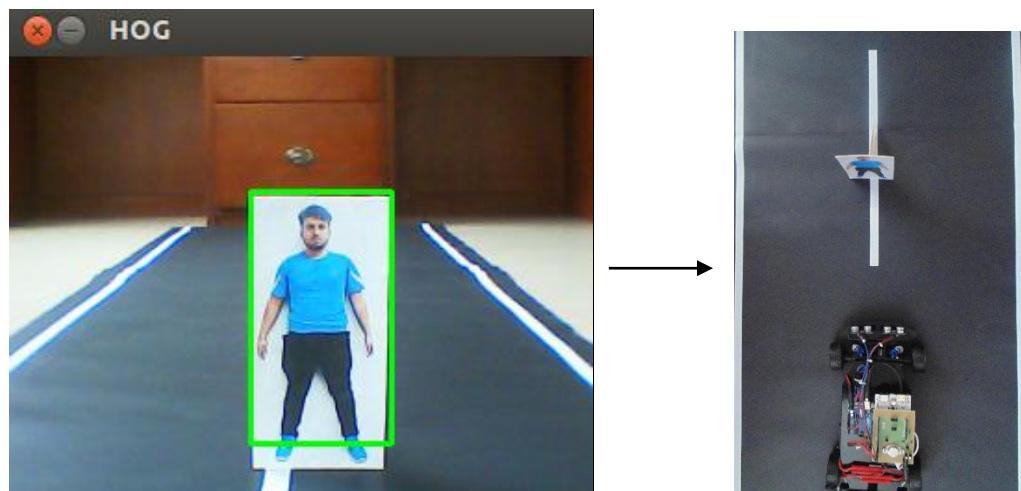
**Figura 159.** Contorno de persona a una distancia media



**Fuente.** Elaboración propia

La detección de una persona a una distancia cercana se ilustra en la (figura 160) con un valor de PWM para velocidad detención.

**Figura 160.** Contorno de persona a una distancia cercana

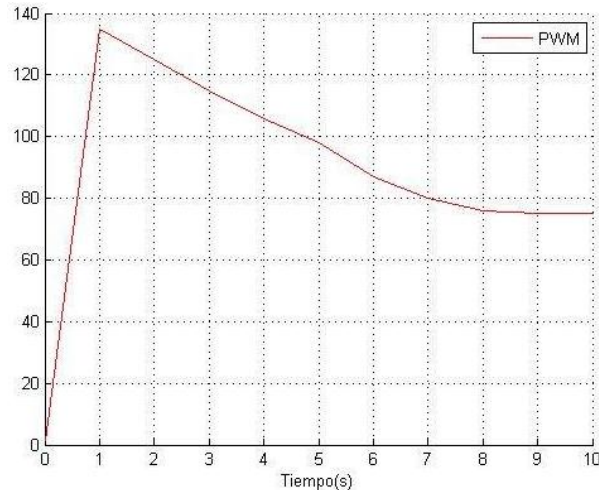


**Fuente.** Elaboración propia

La prueba realizada se efectúa situando el vehículo y el peatón a una distancia lejana, al momento de iniciar la aplicación el vehículo describe una velocidad alta, dado a que detecta la dimensión reducida del cuadro que denota el peatón, a medida que la distancia que separa a ambos es menor, el PWM disminuirá hasta

el punto de generar la acción de detención del vehículo, la cual se da alrededor de los 8 segundos (figura 161)

**Figura 161.** Proceso de detención del vehículo de acuerdo a contorno HOG

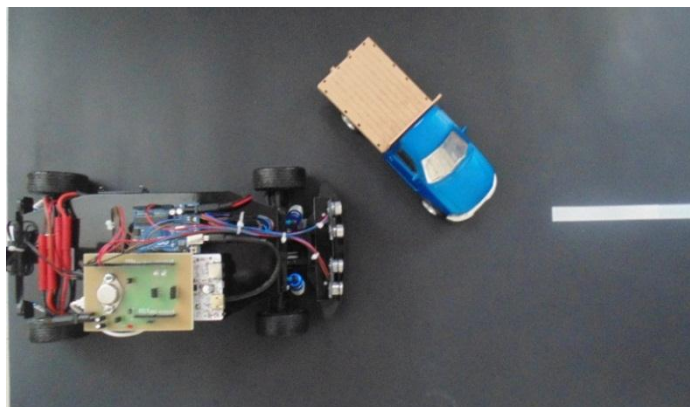


**Fuente.** Elaboración propia

### 6.3.5 Sensor ultrasónico

La implementación de los sensores ultrasónicos se dio como respaldo a la detección de peatones ya que de acuerdo a la implementación de HOG se tienen como referencia unas imágenes predeterminadas para efectuar la detección, sin embargo las imágenes que no corresponden a la identificación asertiva de peatones siguen representando un riesgo para la conducción, además de los peatones no reconocidos existen obstáculos físicos, como por ejemplo otros vehículos o animales en la vía como se ilustra en la (figura 162).

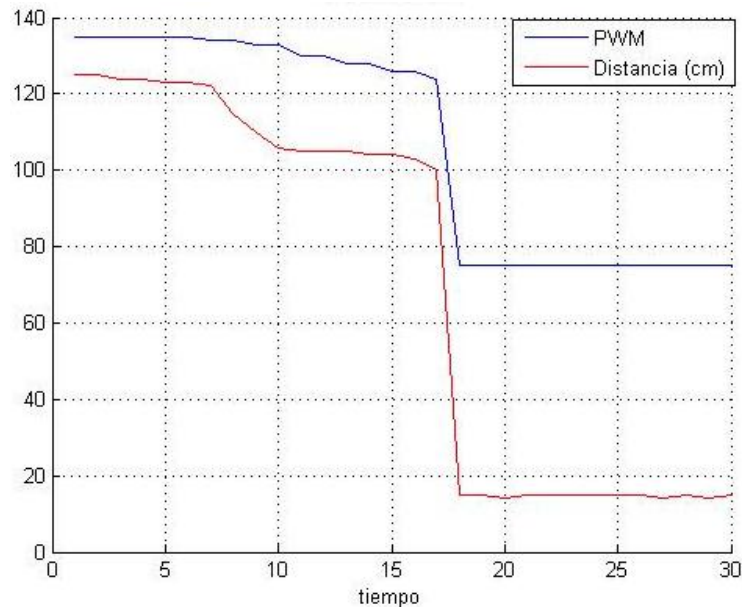
**Figura 162.** Obstáculo repentino



**Fuente.** Elaboración propia

En la (figura 163) se muestra el valor de PWM y la lectura en cm de uno de los sensores, cuando la distancia es mayor a 15 cm el sensor no interviene en el control desarrollado para la detección de peatones haciendo que el valor de PWM varíe libremente de acuerdo al control principal, sin embargo cuando se presenta un obstáculo repentino (en este caso a los 16 seg.) el sensor interviene y coloca el valor de PWM en 75, lo cual corresponde a la detención total del vehículo.

**Figura 163.** Gráfica de PWM y sensor ultrasónico



**Fuente.** Elaboración propia

#### 6.4 CONTROL MANUAL

La implementación de un control manual parte de la necesidad realizar la conducción libre en el escenario de ejecución de los movimientos, esto con el fin de corroborar la cinemática modelada para el vehículo, mediante esta aplicación se observa en el ambiente físico las restricciones no holonómicas que presenta la plataforma, tal como lo dicta el modelo cinemático.

Para el control se desarrolla una interfaz como se ilustra en la (figura 164) la cual cuenta con dos slider, uno para el control de velocidad del vehículo, el segundo está enfocado a la variación de dirección, actuando en conjunto da como resultado la navegación manual de la plataforma

**Figura 164.** Interfaz control manual

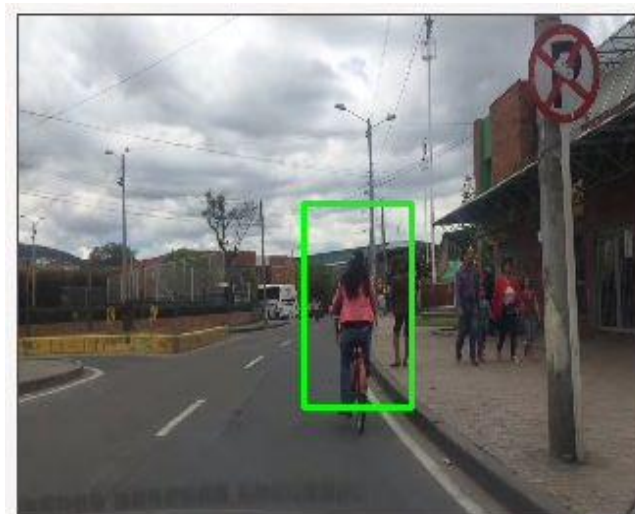


**Fuente.** Elaboración propia

## 6.5 IMPLEMENTACIÓN DE LA SEGUNDA APLICACIÓN EN UN ENTORNO REAL

A continuación se muestran algunas imágenes (figura 165), de videos que se grabaron desde un carro a través de un dispositivo móvil por carretera, en el cual se ilustra la detección del peatón.

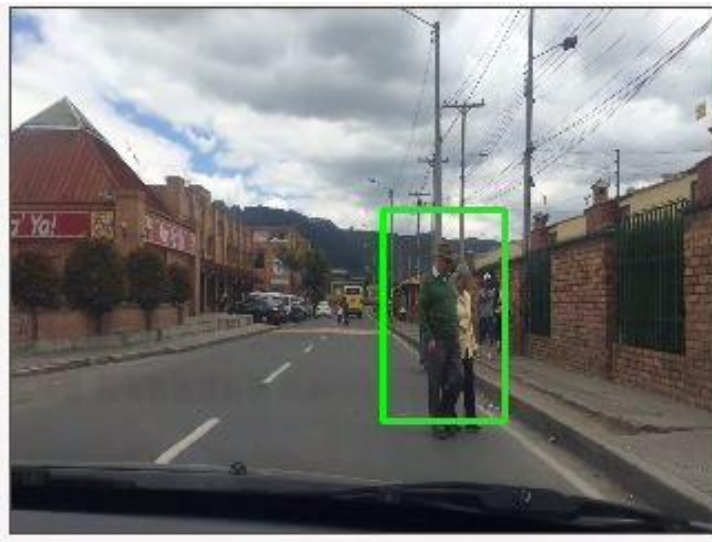
**Figura 165.** Detección de peatones en la vía



a. Primera toma



*b. Segunda toma*



*c. Tercera toma*

**Fuente.** *Elaboración Propia*

En la (figura 166) se puede observar la identificación de los peatones en tiempo real, mediante la implementación de la interfaz propuesta para el desarrollo de la segunda aplicación.

**Figura 166.** Representación de la aplicación HOG



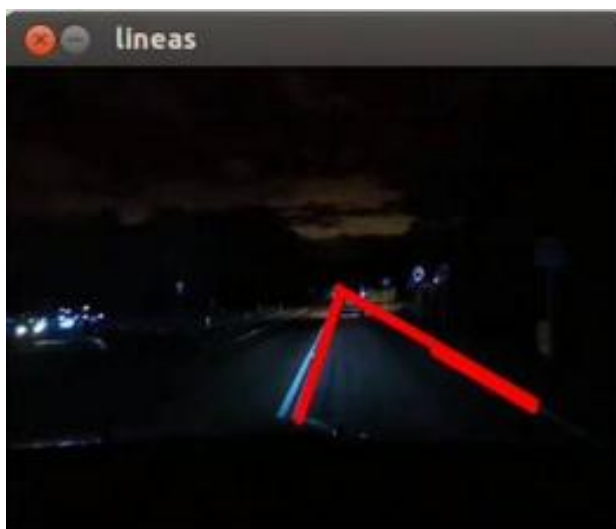
**Fuente.** Elaboración Propia

De igual manera, se muestra la detección del área navegable en un ambiente nocturno (figura 167).

**Figura 167.** Representación de líneas en un ambiente nocturno



a. Primera toma



b. Segunda toma

**Fuente:** Elaboración Propia

La (figura 168) expone el desarrollo de la interfaz, mostrando el área navegable en tiempo real.

**Figura 168.** Representación de la aplicación detección de líneas en la vía



**Fuente:** Elaboración Propia

Videos implementación:

(<https://www.dropbox.com/s/jt7891hh46wngij/AMBIENTE%20%20REAL.mp4?dl=0>)

## CONCLUSIONES

- El trabajo presenta el desarrollo de un vehículo a escala, capaz de navegar de forma autónoma a puntos preestablecidos, ejecutando una trayectoria planeada mediante Spline cúbica y realizando el seguimiento de la misma. El objetivo fue alcanzado por medio del desarrollo de algoritmos, aplicaciones de visión computacional, programación en C++ y la implementación de los dispositivos electrónicos óptimos para la ejecución del control planeado.
- El resultado de las simulaciones fue veraz y aproximado a la realidad, dado al desarrollo del modelo matemático y validación del modelo cinemático del vehículo, lo que permitió tener en cuenta una serie de parámetros esenciales para la definición de la trayectoria y ejecución del movimiento.
- La implementación del algoritmo desarrollado en el vehículo presentó un comportamiento razonable comparado con las simulaciones de planeamiento y seguimiento de trayectorias.
- La aplicación de este algoritmo es válida y eficaz para velocidades bajas dado que al implementar comunicación inalámbrica se presenta un retraso en la señal, lo que impide una correcta lectura y ejecución del movimiento a velocidades elevadas. Esto se debe a que el vehículo se compone de una arquitectura de bajo costo, lo cual genera limitantes en velocidades de procesamiento y ejecución de movimiento.
- La implementación de sensores aportó al control propuesto la percepción necesaria de variables físicas tanto del vehículo como del entorno; con ellas se desarrolló el soporte necesario en velocidad y proximidad de objetos para la asertiva ejecución de las aplicaciones.
- El algoritmo de extracción de características HOG presenta resultados óptimos en la identificación de personas en un sistema de tiempo real, brindando la información suficiente para generar sistemas de control de actuación en escenarios predeterminados.



- La efectividad de la identificación del área navegable y visualización de la misma se da gracias a desarrollos de visión computacional enfocados a la detección de líneas en las áreas destinadas para el tránsito vehicular. La eficacia del método va ligada a las condiciones de luminosidad del ambiente y los rangos de segmentación predeterminados.
- los algoritmos de visión computacional implementados para la identificación de peatones y líneas en la vía, debido a su robustez requieren de una capacidad de procesamiento de máquina elevada, por lo cual la visualización en tiempo real efectuada en el sistema embebido presenta limitaciones en la transmisión de datos

## 7 TRABAJOS FUTUROS

- Optimizar las aplicaciones de visión computacional y la calidad de los dispositivos de procesamiento con el fin de perfeccionar la simulación de un GPS, permitiendo generar un control de mayor precisión en vehículos a escala con velocidades de actuación más elevadas.
- Desarrollar un sistema de procesamiento más robusto y eficiente que permita aplicar los algoritmos de reconocimiento de peatones y líneas en la vía en plataformas de escala real, con el fin de generar futuros sistemas de control orientados a la navegación autónoma y reducción de accidentes de tránsito.

## ANEXO A

# Seguimiento de Trayectorias para un Vehículo Autónomo

## Trajectory Tracking for an autonomous vehicle

Kevin S. Sarmiento G.

Facultad de Ingeniería Mecatrónica  
Universidad Piloto de Colombia  
Carrera 9 N° 45ª - 44 Bogotá, Colombia  
Ksarmiento83@upc.edu.co

Javier E. Vásquez L.

Facultad de Ingeniería Mecatrónica  
Universidad Piloto de Colombia  
Carrera 9 N° 45ª - 44 Bogotá, Colombia  
Jvasquez29@upc.edu.co

Oscar L. Laguna S.

Facultad de Ingeniería Mecatrónica  
Universidad Piloto de Colombia  
Carrera 9 N° 45ª - 44 Bogotá, Colombia  
Olaguna01@upc.edu.co

Ruben D. Hernández B.

Facultad de Ingeniería Mecatrónica  
Universidad Piloto de Colombia  
Carrera 9 N° 45ª - 44 Bogotá, Colombia  
Ruben-hernandez1@upc.edu.co

**Abstract** — El siguiente artículo presenta la aplicación de algoritmos basados en el control de dirección y desarrollos de visión computacional aplicables para un vehículo autónomo terrestre en escala (1:10), con el objeto de realizar seguimiento de trayectorias (previamente estipuladas), considerándolas libre de obstáculos. Estos algoritmos son elaborados en base a las características cinemáticas del vehículo y la simulación de un GPS (desarrollado por medio de procesamiento de imagen). De esta manera se logra que el vehículo se desplace de forma autónoma hacia el camino de puntos predefinidos realizando el trayecto deseado, observando la ejecución del movimiento y la información del sistema mediante la implementación de una interfaz gráfica.

**Keywords** — *Robótica Móvil, Control de Dirección, Planeamiento de trayectorias.*

### I. INTRODUCCIÓN

La robótica móvil ha evolucionado en diversas aplicaciones para satisfacer las necesidades en la optimización de tiempos en tareas determinadas por medio del control, inteligencia artificial entre otros; brindando el apoyo y la seguridad a la hora de ejecutarlas [1-2]. Su principal énfasis se basa en los problemas de operación conocidos como desplazamiento, que involucran los parámetros de reconocimiento y planeación de rutas en ambientes complejos no estructurados.

De forma que los vehículos robóticos terrestres se distinguen principalmente por sus aspectos de configuración de tracción, dirección y grado de actuación, estos pueden ser sub-actuados

hasta completamente no-holonómicos, es decir, que tiene menos grados de libertad controlables y que no puede desplazarse lateralmente. De acuerdo con estos parámetros, los sistemas de control son proyectados satisfaciendo sus características físicas y mecánicas [2]. Sin embargo, el control de vehículos con ruedas presenta dificultades debido a las restricciones no holonómicas propias del modelo cinemático y además de la saturación de los actuadores. Algunas soluciones a este problema son abordados en [3-4]. En el primero de ellos, la velocidad de referencia del robot no puede ser cero, por tanto no puede realizar tareas de posicionamiento. En [3], se obtiene una ley de control estable, otros enfoques similares a este tipo de sistemas, se logran con controladores difusos [4]. El problema de estabilización de posición, significa posicionar el vehículo en el destino final, comenzando en cualquier punto con referencia a la trayectoria, teniendo en cuenta la localización como la orientación con respecto al sistema de coordenadas global.

De acuerdo a lo anterior, se realiza una planeación y ejecución de trayectorias haciendo uso de diferentes métodos y dispositivos. El vehículo seleccionado posee características similares a las de un vehículo real (escala 1:10) usado en competencias profesionales de vehículos a escala, por tal se considera para su análisis matemático la configuración tipo ackerman, el cual puede ser simplificado al modelo de bicicleta (una rueda delantera y una trasera). Este modelo permite explicar por medio de ecuaciones basadas en relaciones geométricas las propiedades físicas del vehículo [5]. Una vez obtenidas las ecuaciones que definen el sistema se implementaron en Simulink de Matlab®, con el objetivo de

validar el comportamiento del vehículo y así mismo analizar las restricciones físicas. Una vez hecha la validación del modelo, se continuó con la implementación y simulación del control de dirección del vehículo teniendo en cuenta las variables de posición y orientación suministradas por el GPS simulado. Este algoritmo permite desplazar el robot a un punto determinado del sistema pasando por una trayectoria previamente definida.

En el escenario en el que se ejecutará el movimiento, se situó una cámara a cierta distancia del suelo tomando el área de ejecución de la trayectoria, mediante filtros de detección de color y posicionamiento de centroide se implementó la simulación de un GPS, que determina la orientación y posición del vehículo dentro del área de trabajo en el cual se ejecutará el seguimiento de trayectorias.

Seguido de esta sección de introducción, el artículo está dividido de la siguiente manera: La sección 2 presenta la estructura mecánica adoptada, en la sección 3 es muestra el modelo matemático del vehículo, en la sección 4 se enseña el sistema propuesto, en la sección 5 el planeamiento de trayectoria propuesto, la sección 6 presenta el algoritmo desarrollado. Demostrado los resultados de validación y simulación en la sección 7; finalmente, en la sección 8 se concluye el desarrollo propuesto, seguido la relación de la bibliografía presentada.

## II. ESTRUCTURA MECÁNICA

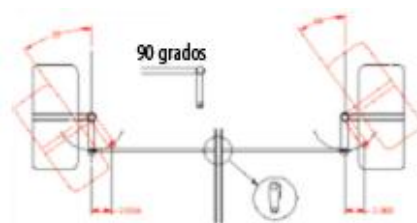
La plataforma utilizada para el proyecto está basada en un vehículo en escala reducida (1:10) y con características aproximadas a un vehículo comercial. Se implementó el carro 1/10th SCALE RC 4WD HIGH PERFORMANCE CAR (Fig. 1), dado que su estructura mecánica cumple con los requerimientos funcionales y cinemáticos necesarios para realizar las pruebas de seguimiento de trayectorias, en un ambiente predeterminado.



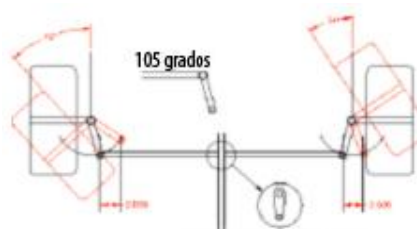
Fig. 1. Plataforma Adoptada.

Este vehículo posee un sistema diferencial (tracción en las cuatro ruedas) y sus dimensiones son apropiadas para realizar

todo el montaje electrónico necesario para el desarrollo funcional del proceso. Además, el sistema de dirección del vehículo tiene como base el modelo Ackerman, a través de la geometría de este modelo las ruedas delanteras giran con ángulos diferentes. La figura 2 presenta un boceto de dos tipos de sistemas, el primero de ellos (Figura 2a) muestra un esquema con el brazo del eje en un ángulo de 90 grados y el otro esquema (Figura 2b), siguiendo el modelo Ackerman, con el brazo del eje con un ángulo de 105 grados. El modelo de 90 grados hace que las dos ruedas delanteras giren paralelamente con un mismo ángulo y su principal desventaja es la pérdida de desempeño y deslizamiento en el momento de la curva. Caso contrario en el modelo de Ackerman, que garantiza que el ángulo de dirección de las ruedas delanteras sean diferentes en el momento de realizar la curva como se muestra en la figura 2b.



a. Dirección convencional



b. Dirección Ackerman.

Fig. 2. Configuración del sistema de dirección.

## III. MODELO MATEMÁTICO

El modelo matemático permite explicar por medio de ecuaciones basadas en relaciones geométricas las propiedades físicas del vehículo, sin considerar las fuerzas que intervienen en su movimiento [5]. En primera instancia se establece un sistema de coordenadas global cartesiano bidimensional, es decir, que se representa en el plano horizontal (x,y) en el que se tiene en cuenta los parámetros usados en la ecuación 1:

$$X = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (1)$$

La identificación en el espacio cartesiano está dado por las coordenadas  $(x,y)$  y la orientación global del vehículo está representado por  $(\theta)$ .

### A. Modelo cinemático

El modelo cinemático está basado en el modelo de Ackerman, la ventaja de este modelo se encuentra en que no considera el deslizamiento de las llantas bajo ciertas condiciones a velocidades bajas [6]. Teniendo esto en cuenta, se parte de que el sistema posee dos ruedas traseras con un eje fijo y dos ruedas delanteras las cuales son operadas mediante una barra para controlar la dirección (fig. 2). Los componentes característicos de este modelo están definidos por un ángulo de dirección  $(\delta_o)$   $(\delta_i)$ , un ángulo de orientación  $(\theta)$ , la separación longitudinal entre las ruedas  $(L)$ , la separación lateral entre las ruedas  $(b)$ , y el punto de centro de gravedad del vehículo  $(X_c, Y_c)$ , en relación a un sistema de referencia global  $(X, Y)$ , como se muestra en la figura 3.

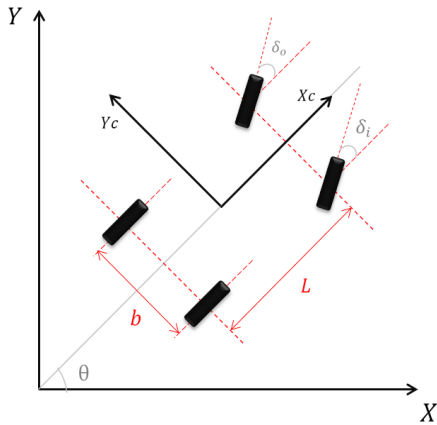


Fig. 3. Configuración Ackerman.

Dado que el vehículo operara a velocidades bajas, este modelo matemático puede ser simplificado con el modelo de la bicicleta [7], el cual posteriormente permitiría la simulación para en el seguimiento de trayectorias. El modelo de la bicicleta considera que se pueden reemplazar las cuatro ruedas del vehículo por solo dos (una rueda trasera y una rueda delantera), de igual forma los ángulos de dirección de las ruedas delanteras se sustituyen a un único ángulo y se conserva el punto de centro de gravedad del vehículo.

Este modelo (fig. 4) reduce las relaciones geométricas y aproxima las características del modelo Ackerman a velocidades razonablemente bajas con ángulos de dirección moderados [8]. Por tal se puede ahora observar un modelo simplificado que describe el CIR (Centro Instantáneo de Rotación) del robot teniendo en cuenta las reducciones previamente mencionadas.

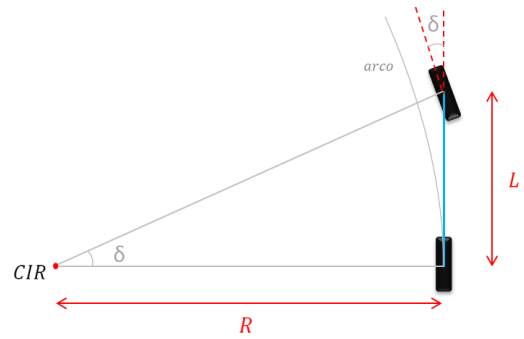


Fig. 4. Geometría del modelo Bicicleta.

A partir del modelo se obtiene la siguiente relación geométrica:

$$\tan \delta = \frac{L}{R} \quad (2)$$

En donde es posible determinar el radio de curvatura del vehículo, dependiente del valor del ángulo de dirección  $(\delta)$ , como se observa en la figura 5.

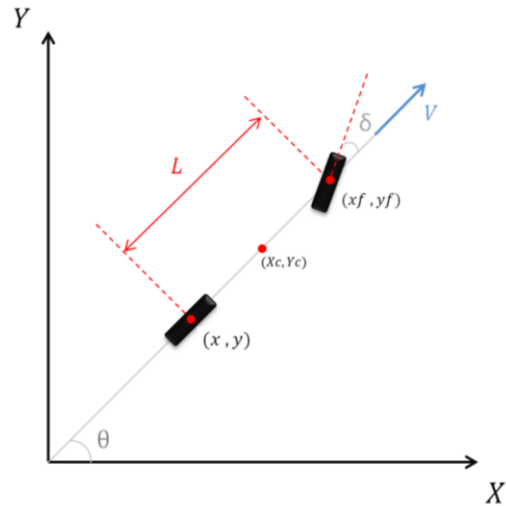


Fig. 5. Modelo cinemático de la bicicleta

Para definir las ecuaciones características del vehículo, es necesario determinar las restricciones no holonómicas para las ruedas delanteras y traseras [9-10].

$$\dot{x}f \sin(\theta + \delta) - \dot{y}f \cos(\theta + \delta) = 0 \quad (3)$$

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0 \quad (4)$$

Para el cual se tiene en cuenta que  $(x,y)$  define las coordenadas globales de la rueda trasera y  $(x_f, y_f)$  las

coordenadas de la rueda delantera. De igual manera es posible expresar las coordenadas globales de la rueda delantera de la siguiente forma:

$$x_f = x + L \cos(\theta) \quad (5)$$

$$y_f = y + L \sin(\theta) \quad (6)$$

A partir desarrollar las ecuaciones (5) y (6) en (3) y teniendo en cuenta que la ecuación (4) se satisface con  $\dot{x} = \cos(\theta)$  y  $\dot{y} = \sin(\theta)$  y cualquier escalar del mismo, se evalúa para la correspondiente velocidad del vehículo [11], de tal manera que las ecuaciones cinemáticas del sistema pueden ser expresadas como:

$$\dot{x} = v \cos(\theta) \quad (7)$$

$$\dot{y} = v \sin(\theta) \quad (8)$$

$$\dot{\theta} = \frac{v \tan(\delta)}{L} \quad (9)$$

#### IV. SISTEMA PROPUESTO

El proceso de comunicación para hacer efectivo el desarrollo de la trayectoria se da mediante la interacción de una cámara IP, un ordenador y un sistema de actuación, respaldados mediante las simulaciones previamente realizadas.

La cámara cumple la función de GPS simulado, es decir, determina la latitud, longitud y orientación del vehículo, con el fin de ubicarlo dentro del plano coordenado de trabajo, para lograr dicho desarrollo la cámara envía frames por segundo al ordenador y allí por medio del uso del software Qt se caracterizan dichos parámetros, mediante la detección de colores. También dentro del ordenador se realiza el desarrollo del algoritmo de control de dirección.

Una vez caracterizados las posiciones del robot y habiendo implementado el algoritmo se envían las señales tanto de velocidad y dirección al vehículo por medio de comunicación bluetooth, al sistema electrónico. Por tal el vehículo ejecutará la trayectoria definida, ajustando la dirección, teniendo en cuenta las señales provenientes del ordenador hasta realizar la ejecución total del movimiento.

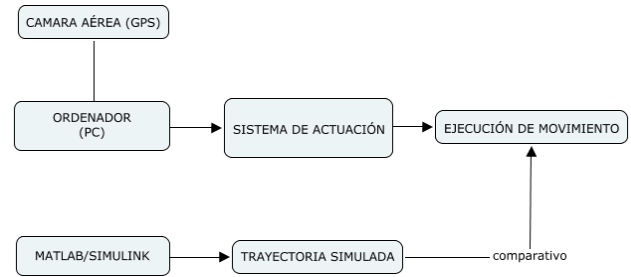


Fig. 6. Arquitectura del sistema.

#### V. PLANEACIÓN DE LA TRAYECTORIA

La planeación de la trayectoria fue definida punto a punto, sin embargo, para realizar dicha trayectoria es necesario tener presente las dimensiones del área de trabajo, captada por la cámara instalada en el techo que simula el GPS. Teniendo en cuenta este parámetro se define una trayectoria considerando las ecuaciones características del sistema, la cual se desarrolló teniendo en cuenta una velocidad constante y se enviaron pulsos para cambiar el sentido de dirección en un instante de tiempo determinado, es decir, giro izquierda y derecha, que visualmente en el plano xy se observa como un cambio de carril.

Una vez obtenido el recorrido ideal, se definen una serie de puntos, por tal, la trayectoria está definida como la composición sucesiva de una serie de puntos por los cuales el vehículo tendrá que pasar en un orden determinado.

El vehículo tendrá una posición y orientación inicial, una vez definidos la secuencialidad de los puntos que conformarán la trayectoria, el robot automáticamente debe reconocer el primer punto al que debe dirigirse, al llegar a las coordenadas del primer punto, planea el siguiente y así continuamente hasta definir la trayectoria simulada (ideal).

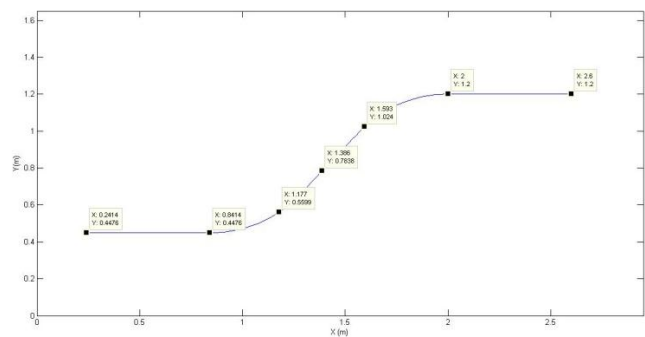


Fig. 7. Trayectoria simulada y definición de puntos.

## VI. ALGORITMO DESARROLLADO

La implementación del algoritmo tiene por objeto, orientar al vehículo hacia la posición deseada, dados los puntos que definen la trayectoria y las condiciones iniciales del sistema [11]. En primera instancia se considera una velocidad constante para el vehículo durante la ejecución de su movimiento. Se parte del problema de que el robot tendrá que llegar a un punto ( p ) definido en el sistema coordenado de trabajo

$$p = (x^* \ y^*) \quad (10)$$

Con el objetivo de que el vehículo se desplace al punto indicado teniendo en cuenta coordenadas inicial y final, se tiene en cuenta la orientación global del sistema actual respecto al deseado definida por:

$$\theta^* = \tan^{-1} \frac{y^* - y}{x^* - x} \quad (11)$$

La siguiente consideración es evaluar una de las ecuaciones características del sistema determinadas anteriormente:

$$\dot{\theta} = \frac{v \tan(\delta)}{L} \quad (12)$$

El control de dirección radica en realizar la diferencia angular entre  $\theta^*$  y  $\theta$  retornando una diferencia en el intervalo entre  $(-\pi, \pi)$  [11], de manera tal que el vehículo ajusta la dirección de las llantas delanteras con el fin de llegar a la coordenada deseada. Con el objeto de reducir el error y mejorar la respuesta del sistema se implementó un controlador proporcional en el valor de la corrección angular definido por:

$$\delta = K_h * \theta^* \quad (13)$$

Siendo  $K_h > 0$

Donde  $\theta^*$  representa el valor de corrección de dirección del intervalo comprendido entre  $(-\pi, \pi)$ .

De igual manera hay que considerar que el vehículo tiene un limitante mecánico en su dirección por tal se establece que las llantas delanteras tienen un giro (externo) máximo de n grados respecto a su centro, por tanto se considera que:

$$\text{llanta der} = \delta_{\max}$$

$$\text{llanta izq} = -\delta_{\max}$$

De acuerdo a los parámetros mencionados se procede a desarrollar todo el sistema en Simulink de Matlab®, con el objetivo de simular el comportamiento del vehículo en la ejecución de la trayectoria.

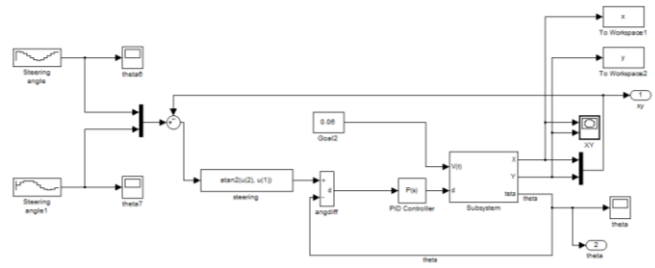


Fig. 8. Algoritmo implementado en Simulink-Matlab®.

De igual forma, para aplicar el algoritmo en el vehículo se utilizó el software libre Qt en el sistema operativo Linux, el cual es una librería utilizada para desarrollar aplicaciones de interfaz gráfica en C++ [12]. En esta aplicación se implementó el código, en el cual se aplicó el desarrollo de visión computacional necesario para realizar la identificación de colores (GPS simulado), obtención de datos y comunicación con el sistema electrónico embarcado en el vehículo.

## VII. RESULTADOS

Para la trayectoria definida, se puede observar que la respuesta del vehículo busca minimizar el error de dirección para así seguir la trayectoria deseada; la ganancia del controlador proporcional es  $K_p = 4$  el cual satisface que el error estacionario sea del 0.1%, el resultado obtenido se presenta en la figura 9.

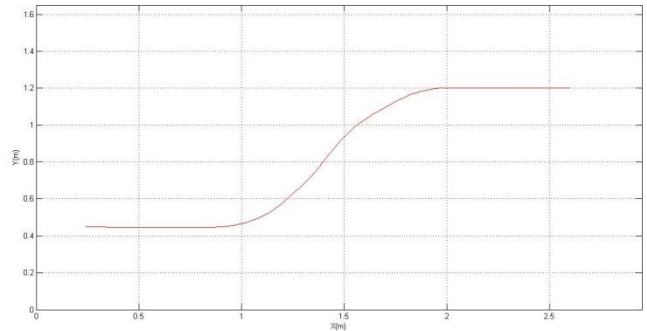


Fig. 9. Trayectoria (real) del vehículo.

En la figura 10 se puede observar el comparativo de la trayectoria deseada (color azul) frente a la resultante mediante el control implementado (color rojo), como se puede observar el vehículo define gran parte de la trayectoria y mediante los ajustes de la ganancia los errores de orientación y referencia son minimizados.

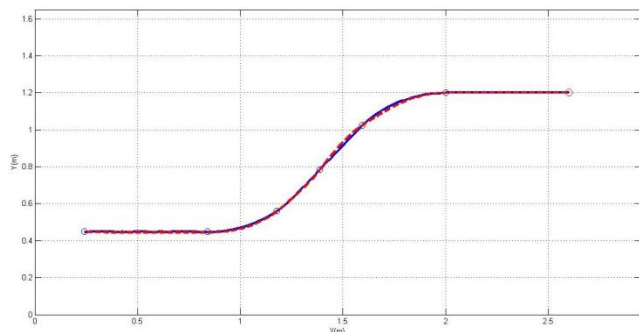


Fig. 10. Comparativo trayectoria real vs. simulada.

De igual forma se obtiene la interfaz gráfica desarrollada en Qt la cual permite observar los datos del GPS simulado y el recorrido del vehículo mientras realiza la definición de la trayectoria deseada.

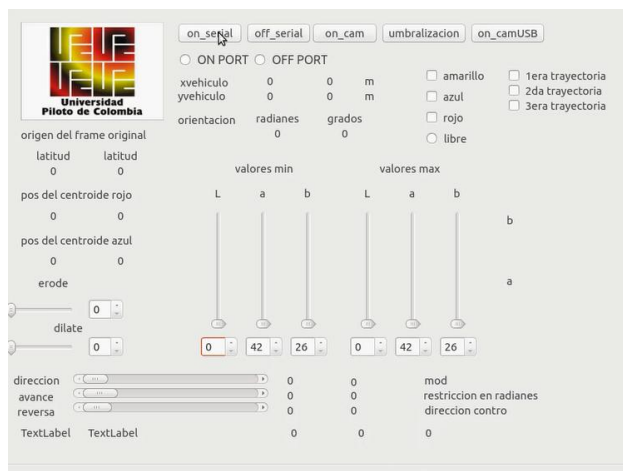


Fig. 11. Interfaz desarrollada.

En la figura 12 se presenta la ejecución de la trayectoria hecha por el vehículo en tiempo real. Como se observar el punto rojo descrito en las imágenes representa el punto de partida y los puntos verdes son las coordenadas previamente seleccionadas. Estos puntos son de referencia, es decir, son de apoyo visual para que se pueda observar la forma del camino y verificar la ejecución de la trayectoria.

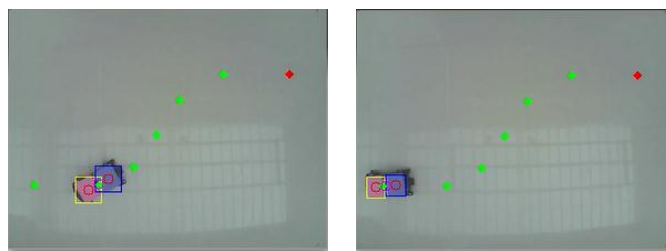
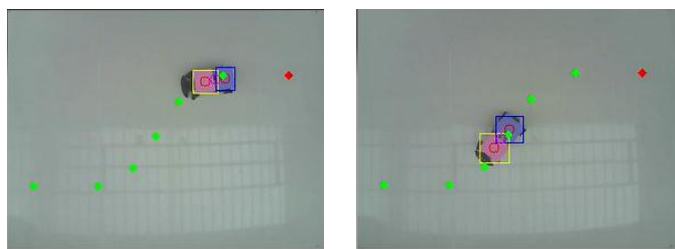


Fig. 12. Ejecución de la trayectoria en Tiempo Real.

## VIII. CONCLUSIONES

El trabajo presenta el desarrollo de un vehículo a escala, capaz de navegar de forma autónoma a puntos preestablecidos definiendo una trayectoria deseada. Esta implementación fue lograda por medio de aplicaciones de visión computacional, programación en C++ e implementación de dispositivos electrónicos.

El resultado de las simulaciones fue veraz y aproximado a la realidad, dado al desarrollo efectivo del modelo matemático del vehículo, lo que permitió tener en cuenta una serie de parámetros esenciales para la definición de la trayectoria y ejecución del movimiento.

El algoritmo fue implementado de forma correcta, ya que el vehículo presento un comportamiento razonable de acuerdo a las simulaciones de la trayectoria propuesta.

La aplicación de este algoritmo es válida y eficaz para velocidades bajas dado que al implementar comunicación inalámbrica se presenta un retraso en la señal, lo que impediría una correcta lectura y ejecución del movimiento.

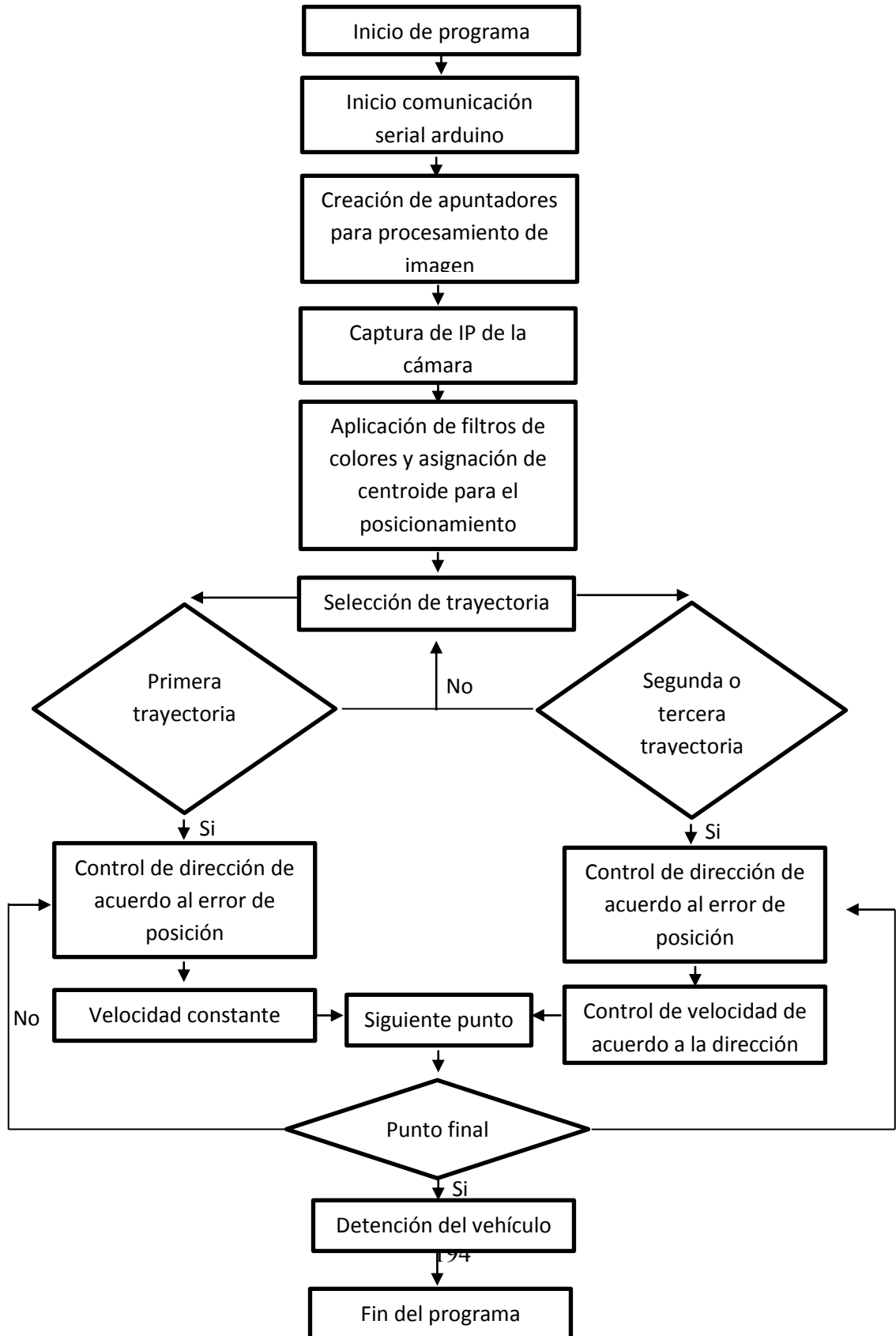
## REFERENCIAS

- [1] BERMUDEZ, Giovanni. Robots móviles: teoría aplicaciones y experiencias. 2002 Tecura 10. [Citado en 20 de mayo, 2015]. Disponible en: <http://tecnura.udistrital.edu.co/ojs/index.php/revista/article/view/225/223>
- [2] Morin, P. and Samson, C. "Motion control of wheeled mobile robots". Chapter 34 of "Handbook of Robotics", Springer Verlag, 2008.
- [3] Doh-hyum Kim, Jum-Ho Oh, (1998). "Globally Asymptotically Stable Tracking Control of Mobile Robots". Int. Conf. on Control Application IEEE pp. 1297-1301.
- [4] Kanayama Y, Kimura Y, Miyazaki F, Noguchi T. "A Stable Tracking Control Method for an Anutonomous Mobile Robot". Int. Conf. on Robotics and Automation. pp 384-389.

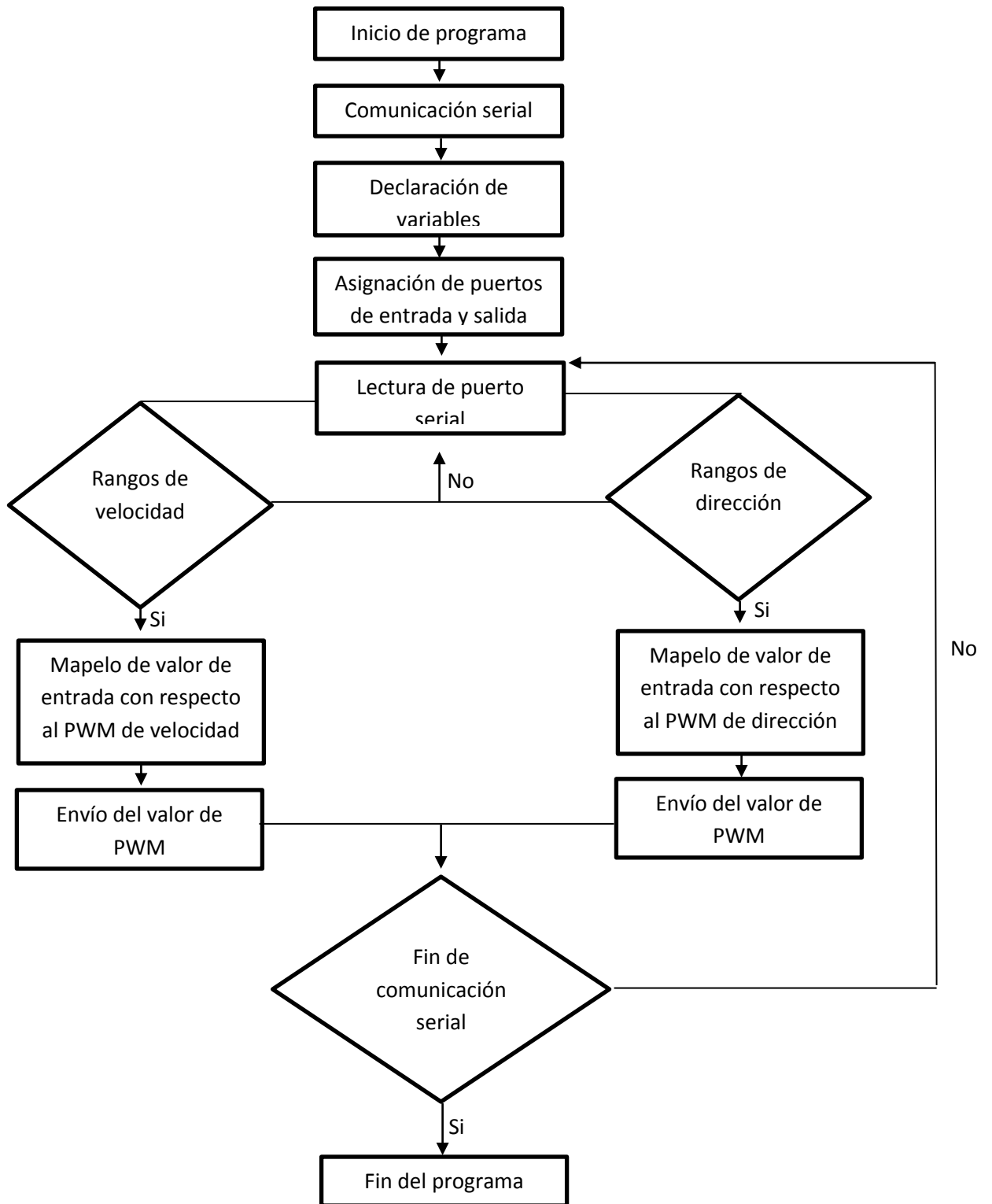


- [5] MARISCAL GARCIA, Efraín. Planeación y seguimiento de trayectorias de robots móviles en una simulación de un ambiente real. 2005. Universidad Autónoma Indígena de México.
- [6] FERNÁNDEZ CARAMÉS, Carlos. Técnicas de navegación para un robot móvil utilizando sistemas de razonamiento espacial. Salamanca, 2012. Universidad de Salamanca. Departamento de informática y automática. [Citado en 15 de mayo, 2015]. Disponible en: <http://gro.usal.es/web/thesis/tesisCF.pdf>
- [7] CASTILLO Oscar, Duran Eyner. Localización de un robot móvil de configuración Ackerman usando el filtro extendido de kalman. Universidad Nacional de ingeniería. [Citado en 25 de junio, 2015]. Disponible en: <http://robotstudio.wdfiles.com/local--files/academics/RobotsAutonomos.pdf>
- [8] G EJARQUE, R HAART, M DISTEFANO. Mecanismo de control de rotación de una rueda esférica. Instituto de automática y electrónica industrial (IAEI). Universidad nacional de Cuyo. Facultad de ingeniería. Argentina.
- [9] DE LUCA, A. ORIOLO G. C. SAMSON. Feedback Control of a Nonholomic Car-Like Robot. This is the fourth chapter of the book: Robot Motion Planning and Control.
- [10] CARDENAS RUEDA, Miguel Ángel. Localização Híbrida para Um Veículo Autônomo em Escala Usando Fusão de Sensores. Campinas, 2012. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica.
- [11] CORKE PETER. Robotics, Vision and Control. Fundamental Algorithms in MATLAB®. Springer. School of Engineering Systems. Faculty of Built Environment and Engineering.
- [12] MERELO RUS. Desarrollo de aplicaciones con Qt. 2007/08. Dpto. de lenguajes y sistemas informáticos. Escuela técnica superior de ingeniería informática.

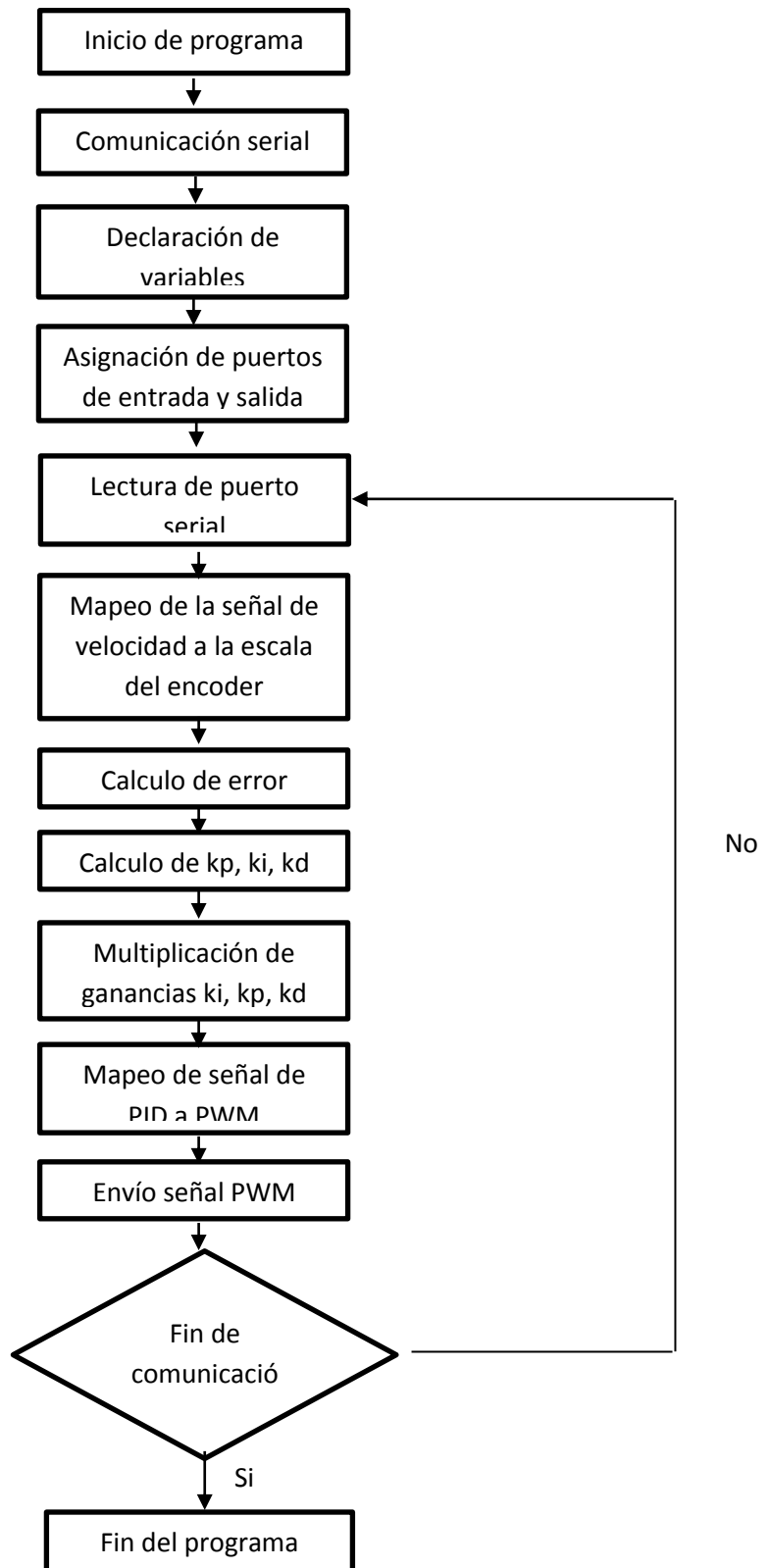
**ANEXO B**  
**DIAGRAMA DE FLUJO PRIMERA APLICACIÓN (PCDUINO)**



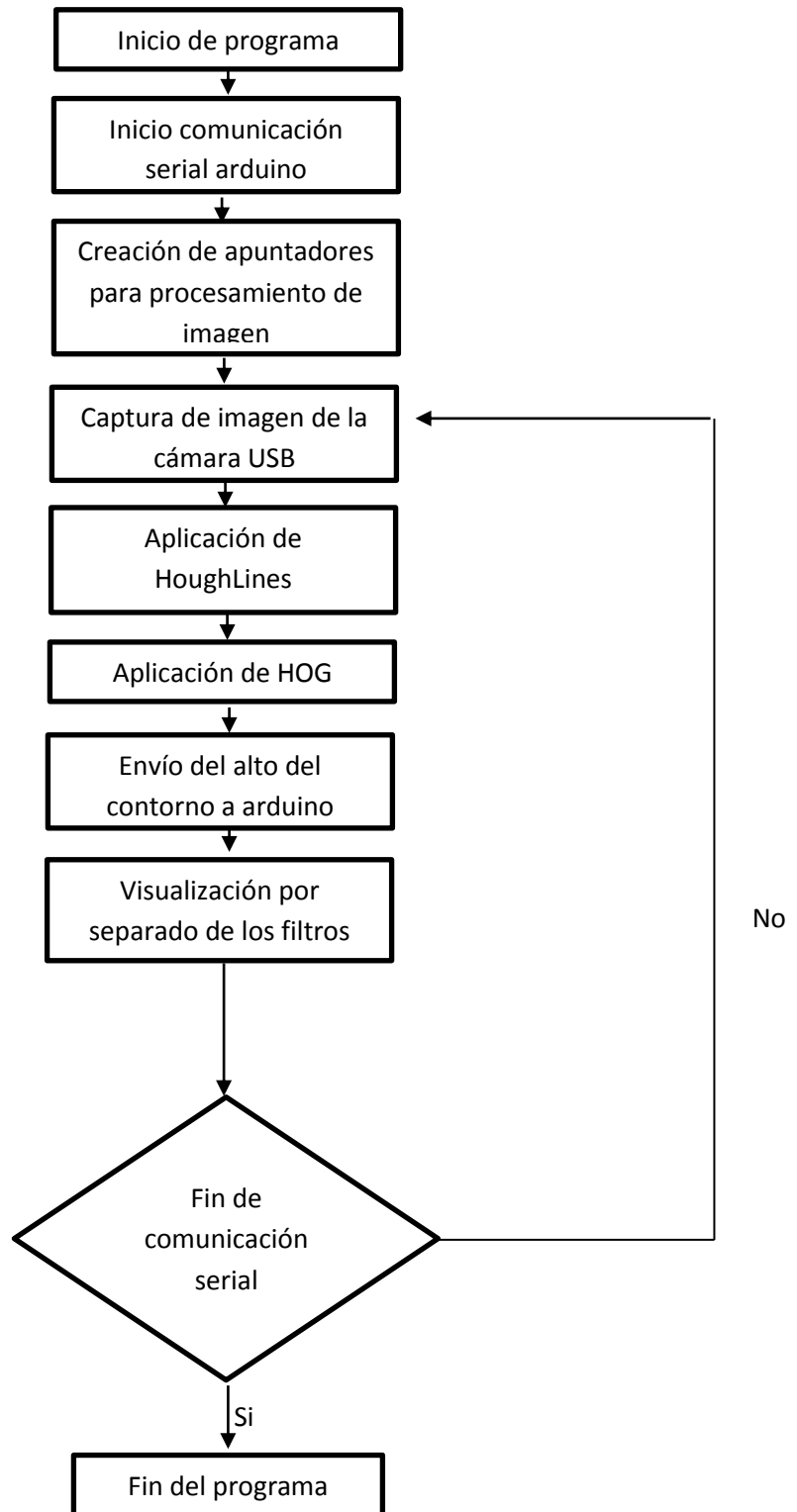
**ANEXO C**  
**DIAGRAMA DE FLUJO PRIMERA APLICACIÓN (TRAYECTORIA CAMBIO DE CARRIL)**



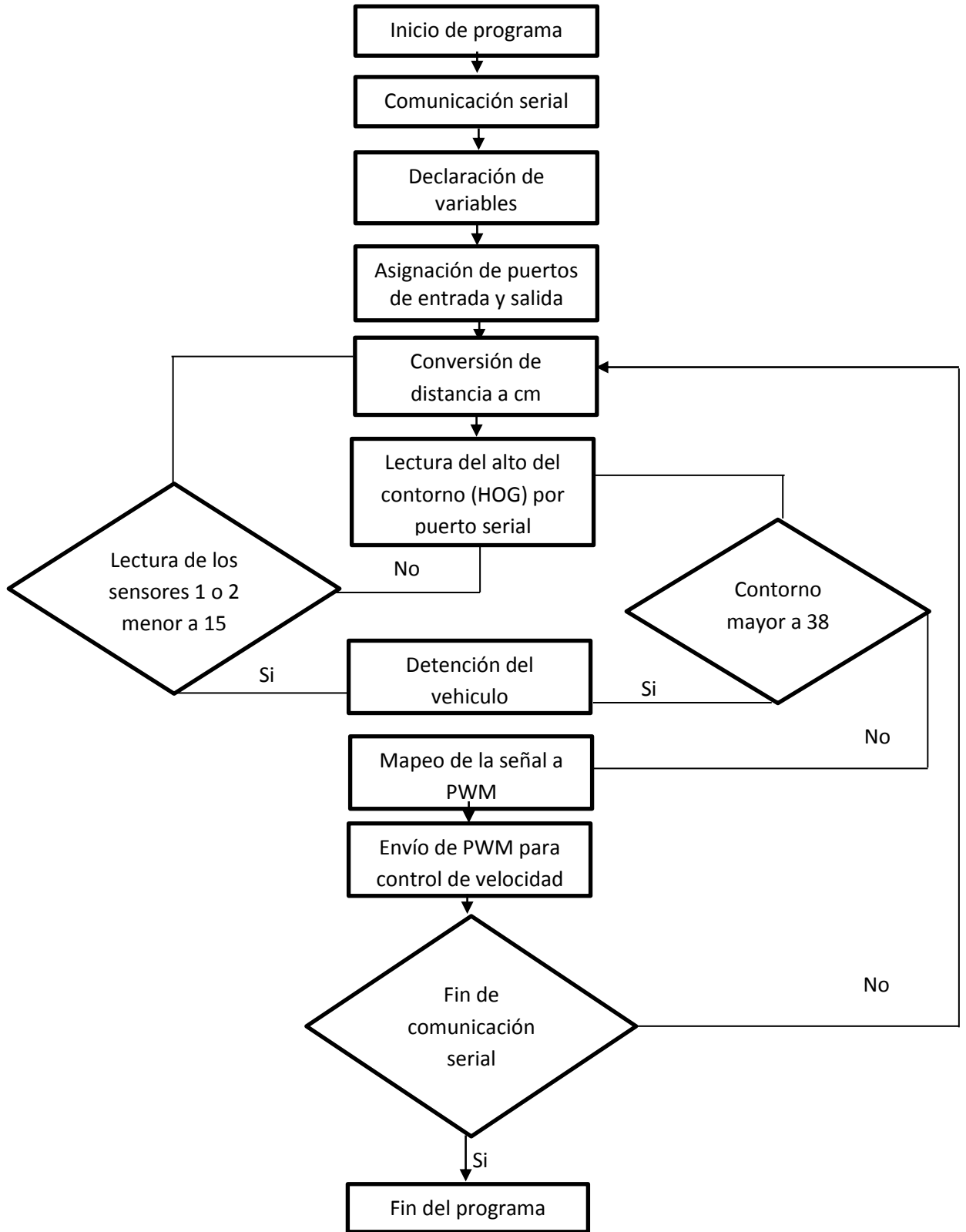
**ANEXO D**  
**DIAGRAMA DE FLUJO PRIMERA APLICACIÓN (TRAYECTORIA GIRO IZQUIERDA Y GIRO DERECHA)**



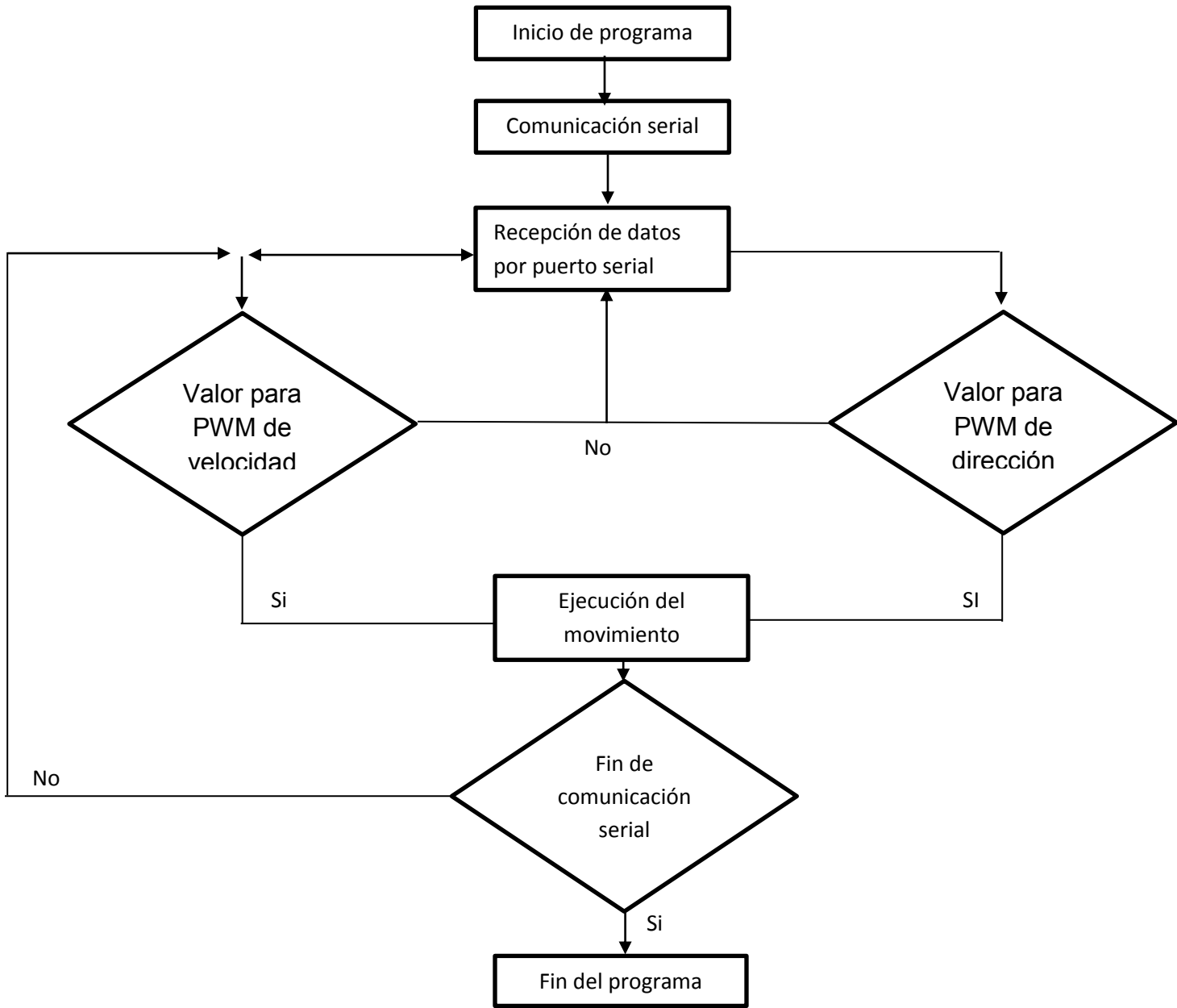
**ANEXO E**  
**DIAGRAMA DE FLUJO SEGUNDA APLICACIÓN (PCDUINO)**



**ANEXO F**  
**DIAGRAMA DE FLUJO SEGUNDA APLICACIÓN (ARDUINO)**

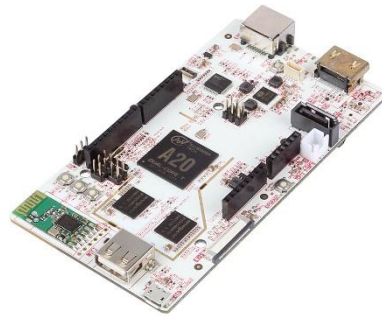


**ANEXO G**  
**DIAGRAMA DE FLUJO CONTROL MANUAL (PCDUINO)**



## ANEXO H INSTALACIÓN DEL KERNEL DE PCDUINO Y CONFIGURACIÓN DEL BOOT

*Figura 1. Pcdiuno 3*



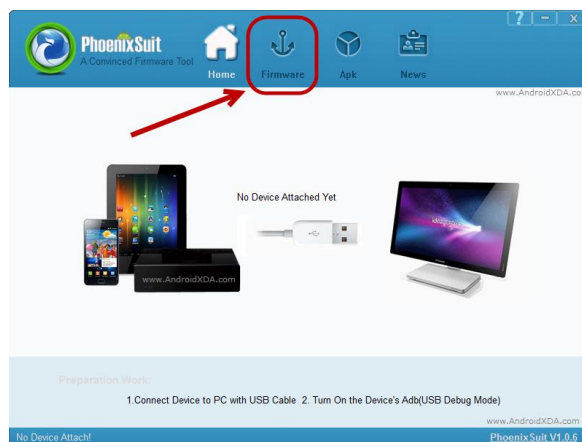
*Fuente. Pcdiuno 3 URL [http://www.linksprite.com/wp-content/uploads/2014/11/IMG\\_0242.jpg](http://www.linksprite.com/wp-content/uploads/2014/11/IMG_0242.jpg)*

### 1 PASOS DE INSTALACIÓN DEL KERNEL

Para comenzar dicha instalación se descargan los siguientes programas y archivos para su respectiva instalación.

PhoenixSuit: Es un programa el cual permite la actualización del firmware del pcDuino el cual se implementara para la actualización de la board. Se abre el entorno de PhoenixSuit seleccionando la opción Firmware como se aprecia en la (figura 2).

*Figura 2. Entorno de PhoenixSuit*



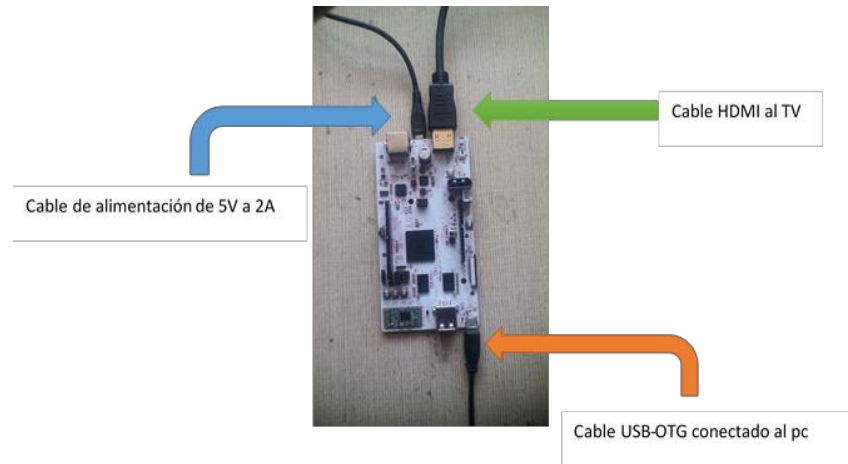
Fuente. Elaboración Propia

1.1 El paso siguiente es obtener la imagen del sistema operativo de la página del siguiente link [http://www.linksprite.com/?page\\_id=855](http://www.linksprite.com/?page_id=855)



1.2 Después de obtener la imagen del sistema operativo se lleva a cabo las siguientes conexiones para ejecutar su debida instalación como se observa en la (figura 3).

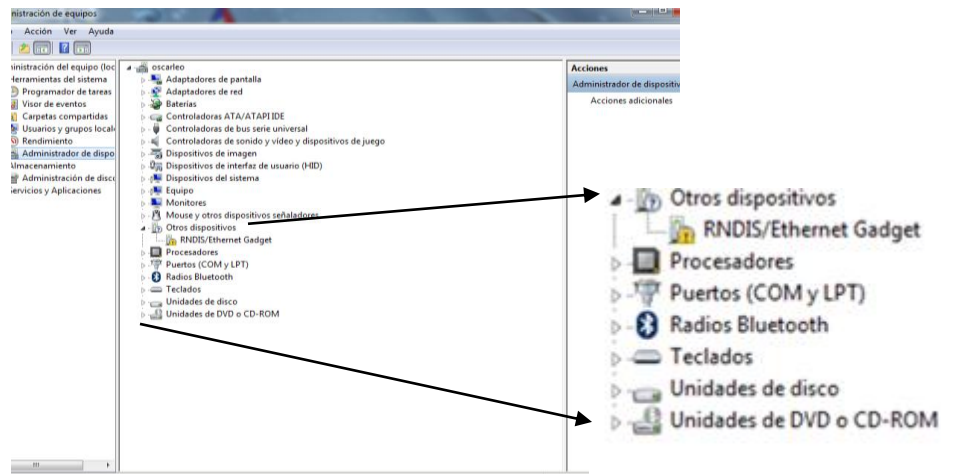
**Figura 3. Conexión del Pcdduino**



**Fuente. Elaboración Propia**

1.3 Para la instalación de los drivers, se debe dirigir a propiedades del PC -> administrador de dispositivos y se selecciona el dispositivo desconocido como se observa en la (figura 4).

**Figura 4. Panel de Administrador de Dispositivos**



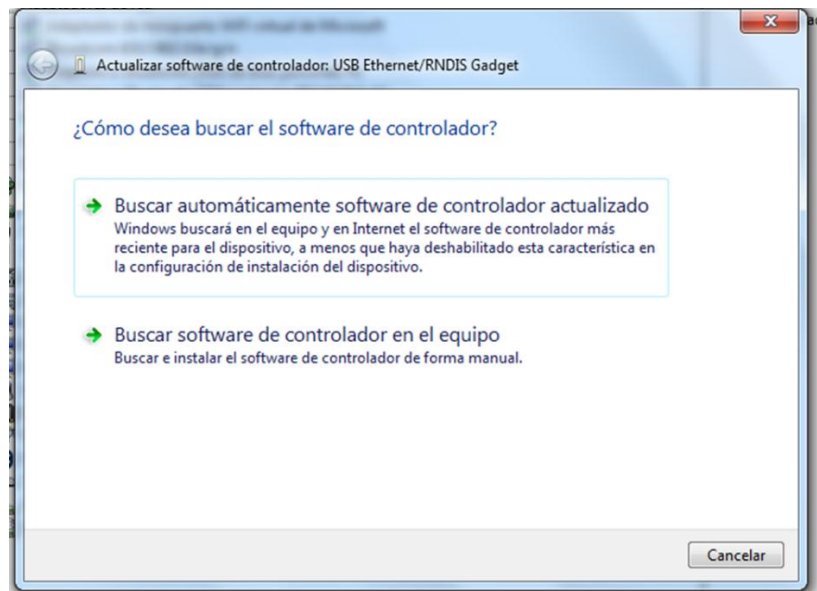
**Fuente. Elaboración Propia**

1.4 A continuación se oprime click derecho y se selecciona actualizar software del controlador.

Nota: el driver del driver del Pduino quedara en este link <https://www.dropbox.com/sh/x87ox682akbns7j/AAB5PXg5upQi8EyBT2UZ04bha?dl=0>

1.5 Después de hacer el proceso anterior, se continúa con escoger la siguiente opción como se observa en la (figura 5).

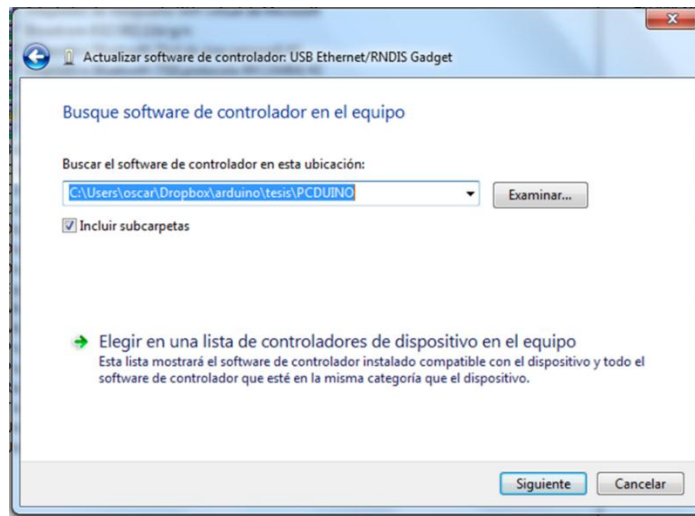
**Figura 5.** Opciones de búsqueda del controlador



**Fuente.** Elaboración Propia

- Se escoge la opción de *Buscar software de controlador en el equipo*.
- A continuación aparecerá la siguiente ventana como se aprecia en la (figura 6).

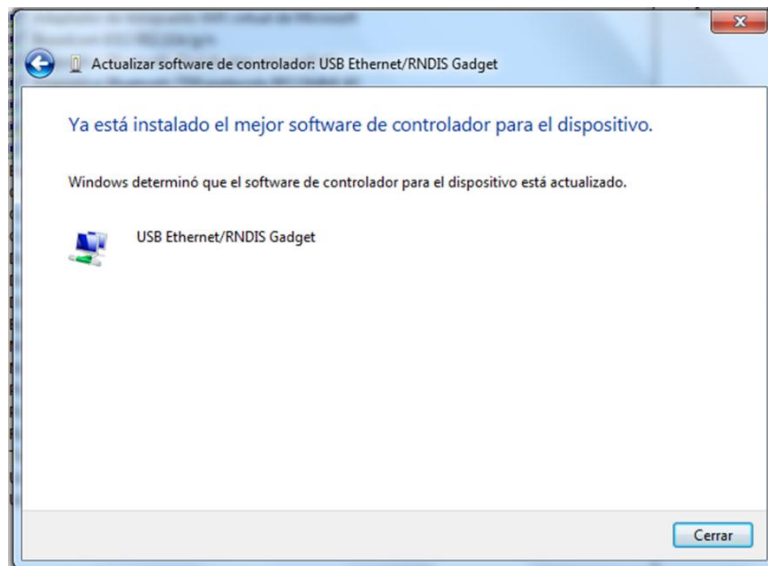
**Figura 6. Escoger directorio del Software**



**Fuente. Elaboración Propia**

1.6 Se procede a ingresar la dirección donde se descargó el controlador y se realiza su instalación, se oprime el botón siguiente y se instalara el driver dando como resultado el aviso que se ve en la (figura 7).

**Figura 7. Instalación del controlador**



**Fuente. Elaboración Propia**

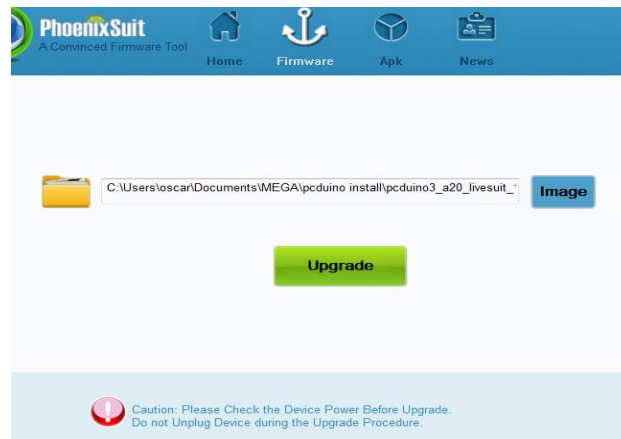
1.7 Para continuar con el proceso de instalación se continúa con Phoenix Suite.

1.8 Se verifica la opción firmware y obtención de la imagen descargada.

1.9 Se ingresa la dirección de donde está ubicada dicha imagen, a continuación se busca la dirección a donde fue descargada la imagen y se procede a seleccionar el sistema operativo anteriormente descargado como se aprecia en la (figura 8).

- Ej: `C:\Users\oscar\Documents\MEGA\pcDuino install\pcduino3_a20_livesuit_1404_20150107.img`

**Figura 8.** Entorno para cargar la imagen del sistema Operativo

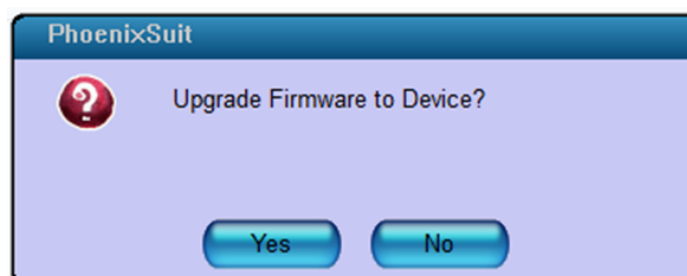


**Fuente.** Elaboración Propia

1.10 Después de asignar la dirección de la imagen se procede a seleccionar la opción upgrade.

1.11 Aparecerá este aviso en el cual se da click a Yes como se observa en la (figura 9)

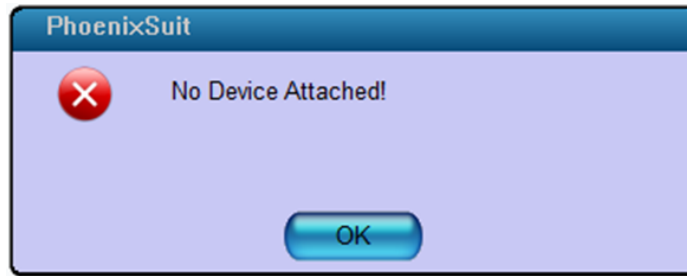
**Figura 9.** Upgrade Firmware



**Fuente.** Elaboración Propia

1.12 Aparecerá el siguiente aviso, se selecciona la opción Ok como se observa en la (figura 10)

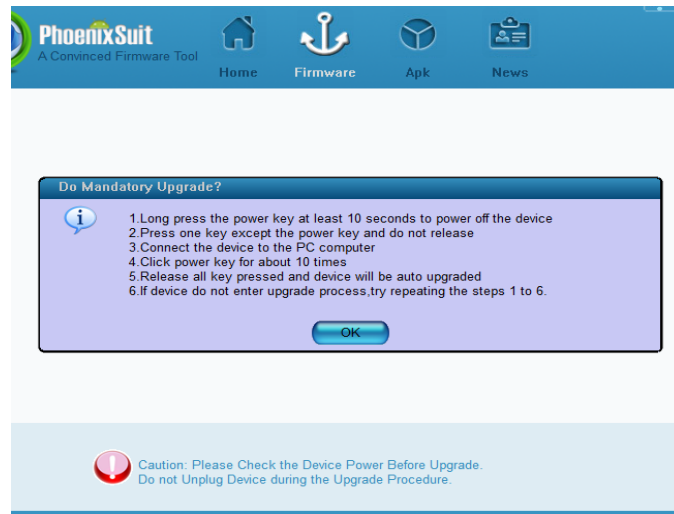
**Figura 10.** No hay Dispositivo Disponible



**Fuente.** Elaboración Propia

1.13 Después saldrá este aviso al cual se selecciona Ok como se observa en la (figura 11)

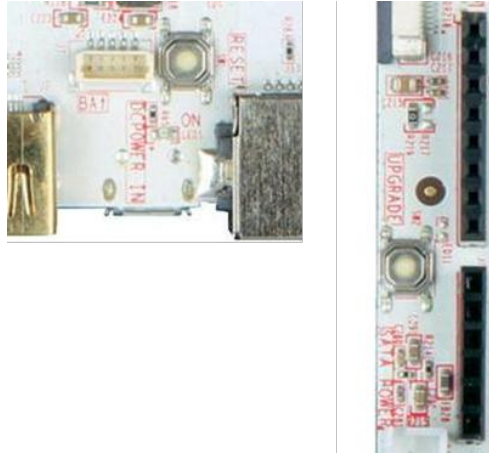
**Figura 11.** Opción para seleccionar el Dispositivo



**Fuente.** Elaboración Propia

1.14 Ahora se oprime los dos botones al mismo tiempo como se aprecia en la (figura 12), los cuales permiten que el pcDuino ingrese el modo Upgrade.

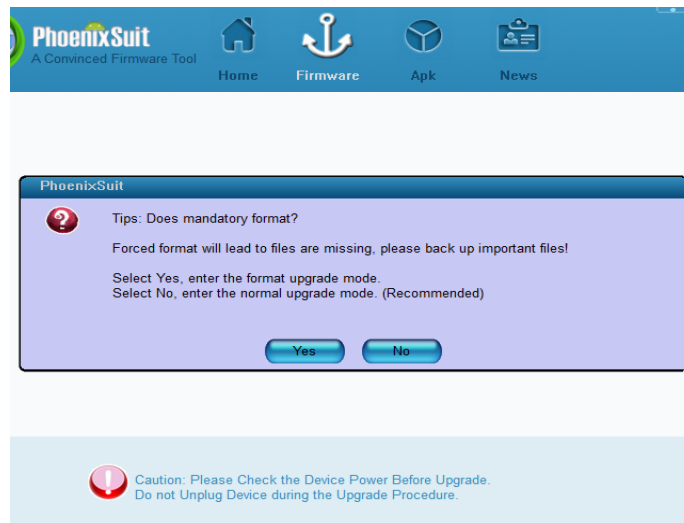
**Figura 12. Botones de inicialización**



**Fuente. Elaboración Propia**

1.15 Tras la anterior acción se abrirá la siguiente ventana la cual da la opción para el debido formato del pcDuino como se observa en la (figura 13).

**Figura 13. Proceso de Formato**

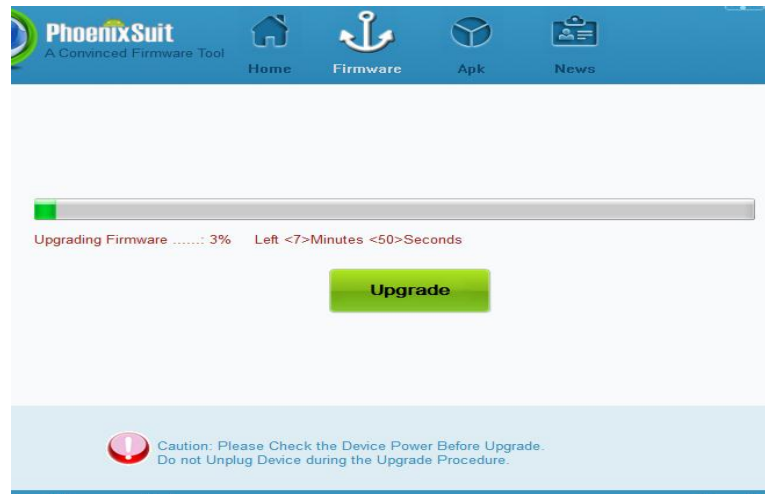


**Fuente. Elaboración Propia**

1.16 A la cual se le deberá seleccionar *yes* para su debido proceso de formato seguido de su instalación.

1.17 A continuación el proceso comenzara dando visualmente una barra de progreso como se observa en la (figura 14), el cual se espera a que dicho proceso finalice.

**Figura 14.** Barra de Progreso



**Fuente.** Elaboración Propia

1.18 Se espera a que instale y el pcDuino se reinicie automáticamente, seguido el pcDuino hará su proceso normal de inicialización.

Arrancará el pcDuino como se observa en la (figura 15).

**Figura 15.** Inicialización del Pcdduino



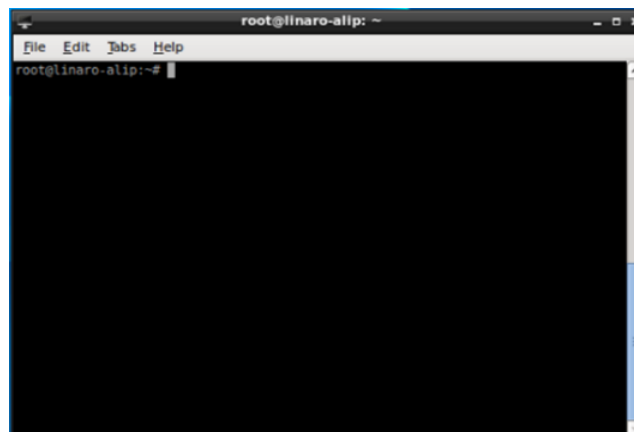
**Fuente.** Elaboración Propia

## 2 Configuración del Boot de pcDuino

Se empieza primero por conectarlo al WI-FI para que su kernel se configure bien

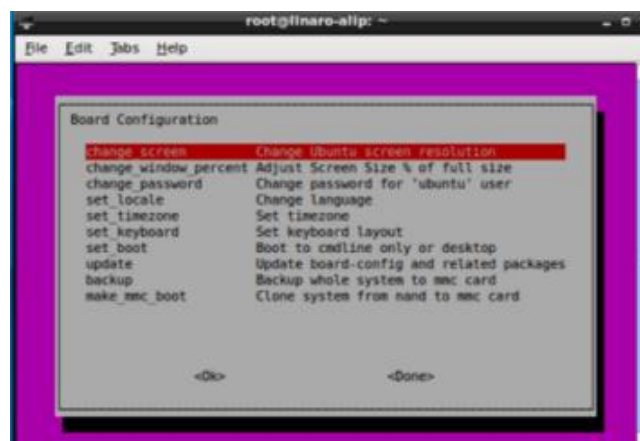
- Con el siguiente comando escrito en el terminal (figura 16), *sudo board-config.sh*, que podrá ser ingresado por medio de un teclado y un mouse conectados al pcDuino por medio USB. Nos permitirá ingresar al menú de configuración (figura 17).

**Figura 16.** Terminal de Pcdiuno



**Fuente.** Elaboración Propia.

**Figura 17.** Menú de Configuración de Pcdiuno

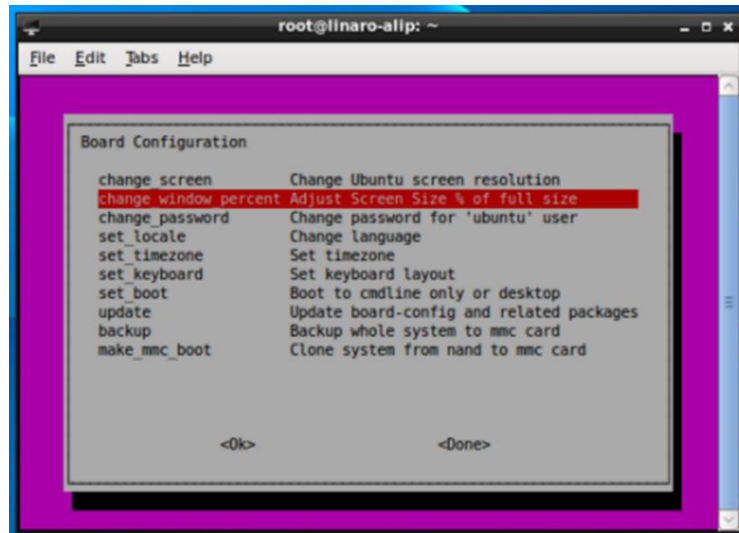


**Fuente.** Elaboración Propia

- De las opciones que allí aparecen se escoge *Change\_Windows\_percent* el cual permite darle en factor de proporción una escala a la visualización como se observa en la (figura 18) y se procede a seleccionarla.



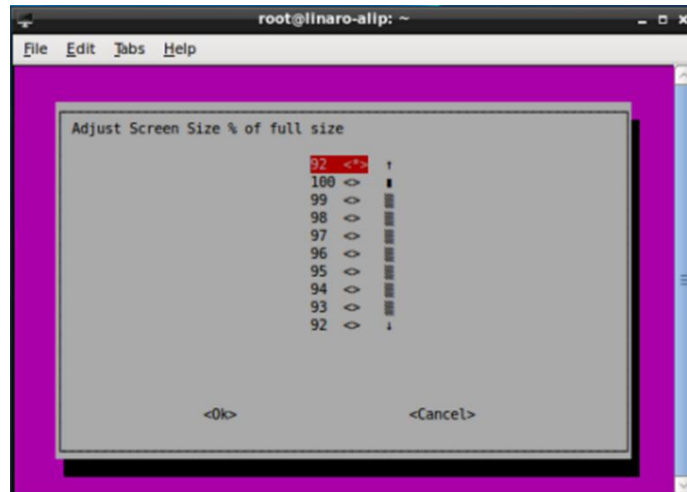
**Figura 18.** Change Windows Percent



**Fuente.** Elaboración Propia

- A continuación se observa la ventana, la cual ilustra dicha escala en valores como se observa en la (figura 19).

**Figura 19.** Valores en Escala de Pantalla

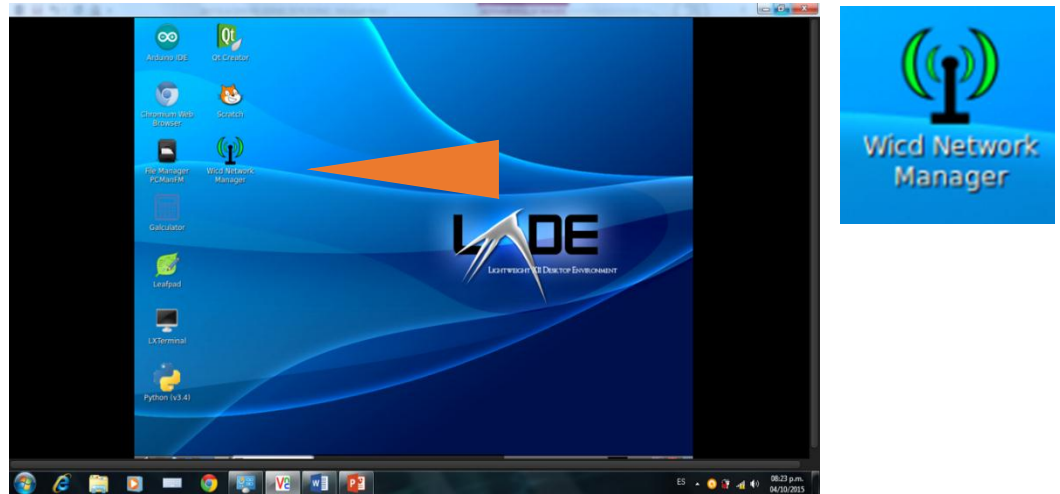


**Fuente.** Elaboración Propia

- Se procede escoger el 92 para la opción en que la pantalla se ajuste a la resolución a la que esté siendo proyectada.
- Seleccionando <OK> se define automáticamente la selección.

- El paso siguiente será conectar al pcDuino a internet, el cual tiene una aplicación denominada Wicd Network Manager como se observa en la (figura 20).

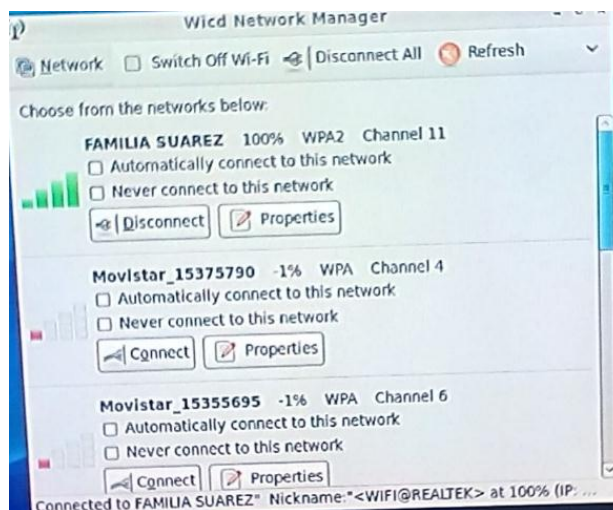
**Figura 20.** Wicd Network Manager



**Fuente.** Elaboración Propia

- Cuando se ejecuta aparece este menú (figura 21) y se procede a conectarse a la red Wifi.

**Figura 21.** Menu Wicd Network Manager



**Fuente.** Elaboración Propia

## ANEXO I

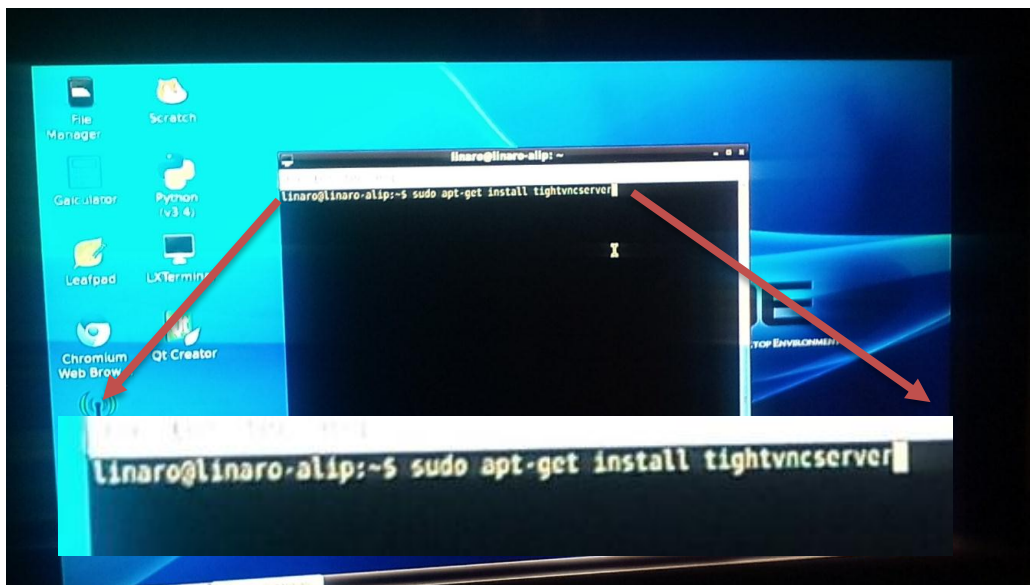
### INSTALACIÓN Y CONFIGURACIÓN DE TIGHTVNC SERVER EN PC DUINO

#### INTRODUCCIÓN SOBRE TIGHTVNC SERVER

Tightvncserver es una aplicación por la cual permite visualizar escritorios remotos a través de una conexión ya sea por una red local o inalámbrica. Esta aplicación se ejecuta en los sistemas operativos Linux, en Windows solo se pueden utilizar para visualizar el escritorio como se explica en este documento.

- La aplicación fue instalada por medio de una conexión HDMI entre el TV y el pcDuino. Se abre un terminal y se procede a escribir el comando `sudo apt-get install tightvncserver` –y como se observa en la (figura 1).

*Figura 1. Instalación de Tightvncserver*

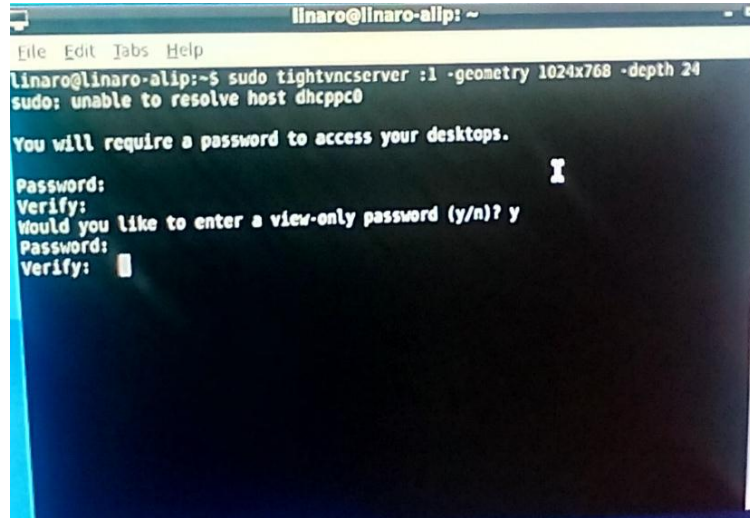


*Fuente. Elaboración Propia*

- Al escribir el anterior comando se le dará enter, y comenzará a instalarse la aplicación. Cuando finalice la instalación, se procede a la creación de la interfaz Gráfica.
- **Creando un servidor gráfico:** El cual permitirá la conexión remota al pcDuino, con el siguiente comando, `sudo tightvncserver: 1 -geometry 1024x768 -depth 24` como se observa en la (figura 2).

La configuración de la resolución de 1024x768 puede variar por la que se acomode más al dispositivo de donde fue visualizado el escritorio remoto.

**Figura 2.** Servidor Gráfico



```
linaro@linaro-alip: ~  
File Edit Tabs Help  
linaro@linaro-alip:~$ sudo tightvncserver :1 -geometry 1024x768 -depth 24  
sudo: unable to resolve host dhcpcd0  
  
You will require a password to access your desktops.  
Password:   
Verify:   
Would you like to enter a view-only password (y/n)? y  
Password:   
Verify: 
```

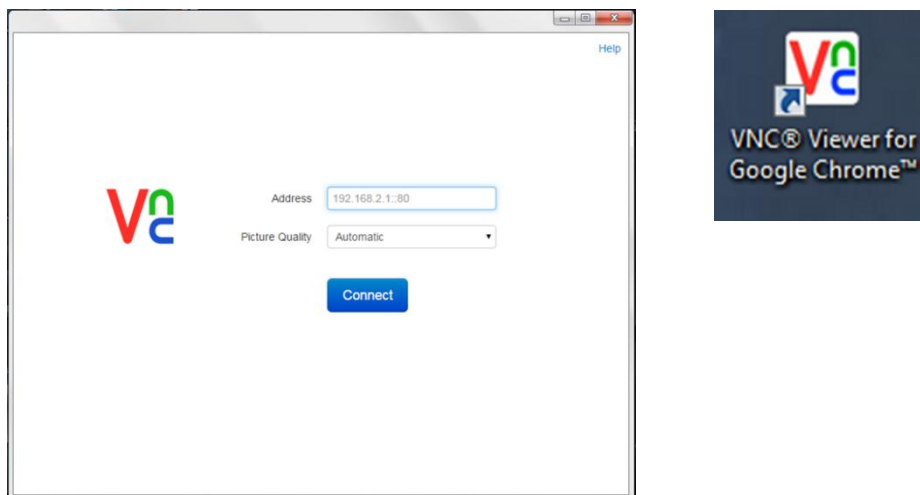
**Fuente.** Elaboración Propia

- Después de introducir dicho comando se le da enter.
- Se requirió crear una contraseña.
- Se selecciona enter. Aparecerá que se verifique.
- La cual se copia de nuevo para verificar.
- En "would you like to enter" se asigna la letra y (que requerirá una contraseña para visualizar la pantalla).
- Selecciona la misma contraseña asignada anteriormente
- Se procede a verificarla.

## 2.1 CONEXIÓN AI SERVIDOR GRÁFICO

Se procede a ejecutar el programa VNC Viewer el cual se puede observar su aplicación y su icono en las (figura 3)

**Figuras 3.** Aplicación e icono de VNC Viewer for Google Chrome

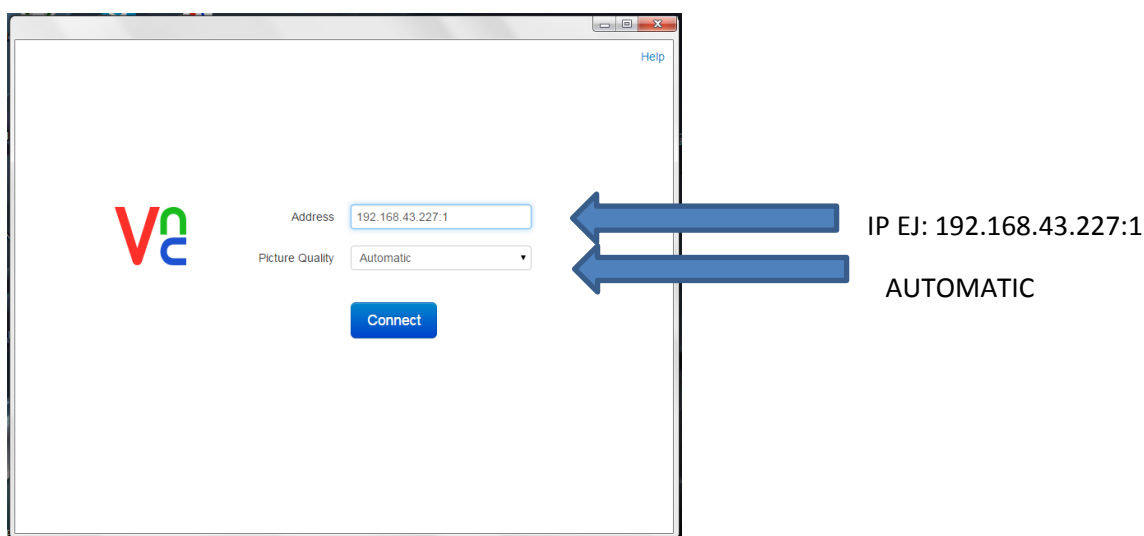


**Fuente.** Elaboración Propia

Aparecerán dos espacios denominados cada uno como **ADDRESS Y PICTURE QUALITY**.

En ADDRESS se escribe la dirección IP que le fue asignado al dispositivo por el modem ej. (192.168.XXX.XXX) la cual se la asigna al dispositivo cuando se es conectado, y en PICTURE QUALITY se deja Automatic como se aprecia en (figura 4).

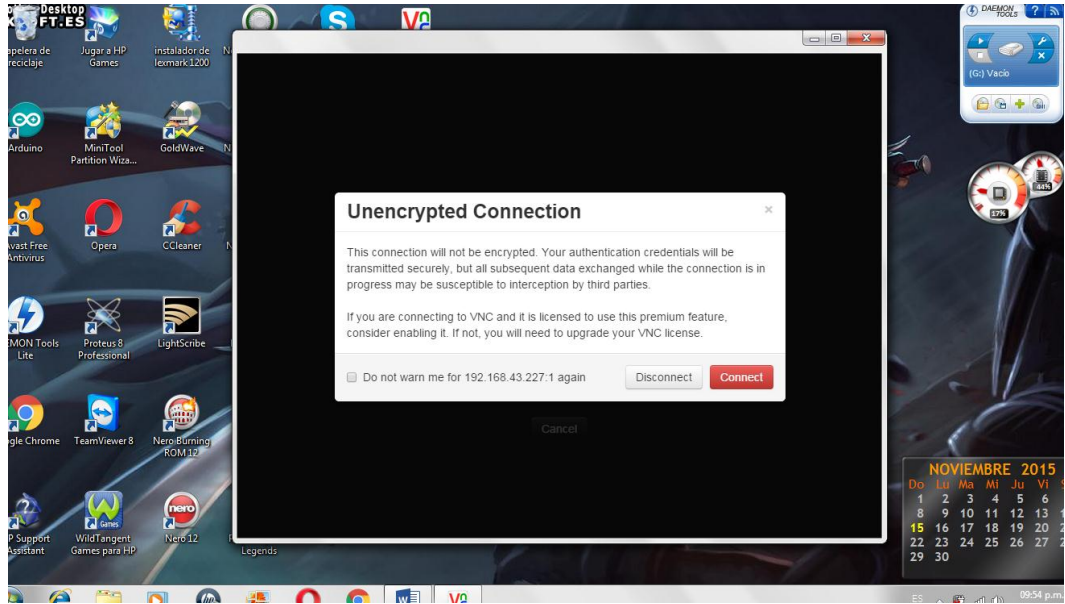
**Figura 4.** Asignación de IP



**Fuente.** Elaboración Propia

El paso a seguir será presionar Connect y aparecerá lo siguiente (Figura 5).

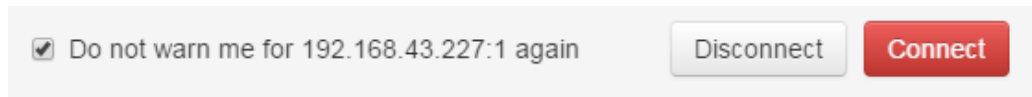
**Figura 5. Conexión Encriptada**



**Fuente. Elaboración Propia**

Se selecciona check box para que recuerde la conexión (figura 6)

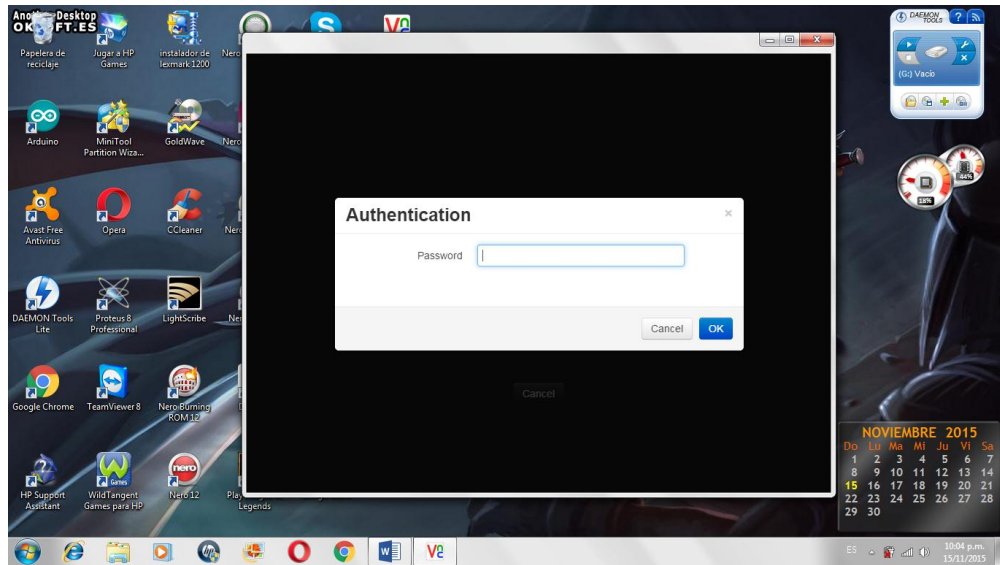
**Figura 6. Selección de Autoguardado**



**Fuente. Elaboración Propia**

Se procede a presionar Connect. A continuación aparecerá una nueva ventana (figura 7), la cual pedirá se escriba la contraseña que se le indico en la creación del servidor, se presiona enter para finalizar.

**Figura 7. Petición de contraseña**



*Fuente. Elaboración Propia.*

- Dando como resultado la (figura 8).

**Figura 8. Visualización de VNC Viewer (en Windows)**



*Fuente. Elaboración Propia*

## ANEXO J DATOS TÉCNICOS DE SERVO-MOTOR MG995

### Descripción:

En este anexo se procederá a dar a conocer la información técnica del servomotor implementado. El cual tiene un rango de trabajo entre 0 y 180°. Lo cual la forma de programarlo es muy práctico, ya que te da la facilidad de controlarlo con un microcontrolador o con arduino.

Representación del servo (figura 1).

*Figura 1. Servomotor*



*Fuente. URL*

[http://www.geelmed.com/documents/products/producto\\_000000132DOC.pdf?d=1312813326](http://www.geelmed.com/documents/products/producto_000000132DOC.pdf?d=1312813326)

### DATOS TÉCNICOS

- Dimensiones: 1.57" x 0.79" x 1.44" (40x20x36.5mm)
- Peso 1.7 oz (48gr)
- Velocidad de operación (4.9V sin carga) 0.17 seg / 60 grados
- Velocidad de operación (6.0V sin carga) 0.17 seg / 60 grados
- Torque de parada (4.9 V) (13Kg/cm)
- Torque de parada (6.0 V) (15Kg/cm)
- Rango de Temperatura -30° a +60 grados °C

*Fuente. URL*

[http://www.geelmed.com/documents/products/producto\\_000000132DOC.pdf?d=1312813326](http://www.geelmed.com/documents/products/producto_000000132DOC.pdf?d=1312813326)



## ANEXO K

### DATOS TÉCNICOS DE WP-1040 BRUSHED, WP-1040 BRUSHED-CRAWLER AND BOAT.

El siguiente dispositivo (figura 1) es un circuito de potencia que por medio de la conversión de una señal digital (~PWM) genera la potencia necesaria para alimentar el motor.

*Figura 1. WP-1040 BRUSHED*



*Fuente. URL <http://www.redcatracing.com/wp-1040-brushed.jpg>*

#### FEATURES:

1. Water-proof and dust-proof for all weather races.
2. Small size with built-in capacitor module
3. Automatic throttle range calibration, easy to use.
4. Multiple protections: Low voltage cut-off protection for Lipo or NiMH battery / Over-heat protection / Throttle signal loss protection.
5. Easily programmed with the jumpers.

**Tabla1. SPECIFICATIONS WP-1040 BRUSHED, AND WP-1040 BRUSHED-CRAWLER AND BOAT.**

FWD Cont. / Burst Current		Forward: 40A / 180A
BWD Cont. / Burst Current		Backward 20A / 90A
Input		2-3S Lipo, 5-9 Cells NiMH/NiCD
Cars Applicable		1:10 on-road Buggy, SCT, Truggy 1:10 Crawler, Tank and Boat
Motor Limit	2 Lipo or 6 NiMH	540 or 550 size motor $\geq$ 12T RPM < 30000 7.2 V
	3 Lipo or 9 NiMH	540 or 550 size motor $\geq$ 18T RPM < 20000 7.2 V
Resistance		FWD: 0.002 Ohm; BWD: 0.004 ohm
Built-in BEC		2A/5V (Linear mode BEC)
PWM Frequency		1 KHz
Dimension		(46.5 x 34 x 28.5) mm
Weight		WP-1040 BRUSHED: 65 gr WP-1040 BRUSHED-CRAWLER: 70 gr

**Fuente.** User Manual of Water-Proof Brushless Speed Controller (RTR Version)

## ANEXO L

### DATOS TÉCNICOS DE MOTOR FAULHABER

Este motor (figura 1) cuenta con una buena gama de revoluciones las cuales viene equipada con una caja reductora la cual permitió el torque necesario para la velocidad deseada.

**Figura 1. MOTOR FAULHABER**



**Fuente.** URL <https://grabcad.com/library/faulhaber-dc-motor-gear-box-1>

**Tabla 1. Características**

Modelo	2342L012CR
Longitud	42 mm
Diámetro	30 mm
Diámetro de eje	6 mm
Longitud de eje	35 mm
Potencia	17W
Voltaje de Funcionamiento	12 V
Torque	17.54 Kg-cm
Velocidad sin carga	8100 RPM
Velocidad con carga (reducción de engrane)	120 RPM
Corriente sin carga	75 mA
Corriente con carga	1500 mA
Relación de Engranajes	1/64
Material caja Reductora	Metal

**Fuente.** URL <http://www.electrotekmega.com/producto/motor-reductor-faulhaber/>

## ANEXO M

### SOFTWARE UTILIZADOS

#### QT

QT es un IDE (entorno de desarrollo integrado) multiplataforma que permite a los desarrolladores utilizar una herramienta creada para la compilación y construcción de código basado en lenguaje máquina. Este compilador soporta código de Tipo C++, QML y HTML Script.

Qt le permite al usuario creación de proyectos basados en interfaces gráficas para el diseño de Widgets, los cuales son totalmente funcionales.

**Fuente:** URL [https://wiki.qt.io/Category:Tools::QtCreator\\_Spanish](https://wiki.qt.io/Category:Tools::QtCreator_Spanish)

#### LINUX

Es uno de los términos conocidos para referirse a un núcleo o kernel, su desarrollo es un ejemplo prominente de software libre, eso da a entender que puede ser modificado y redistribuido libremente por cualquier persona.

Un sistema operativo consiste en varios programas que necesita el ordenador para poder comunicar y recibir instrucciones de los usuarios. La parte más importante de un sistema es su núcleo. Linux es el núcleo. Linux se diseñó para que fuera un sistema multitarea y multiusuario.

El núcleo Linux apareció en 1991, cuando un estudiante finlandés Linus Torvalds anuncio una primera versión de un núcleo que reemplazo uno anterior denominado Minix.

**Fuente:** <https://www.debian.org/releases/stable/mips/ch01s02.html.es>

#### OPENCV

OpenCV es un conjunto de librerías utilizadas en el área de visión artificial, la cual fue desarrollada por Intel. Es de uso multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Está comprendida entre más de 500 funciones que operan en una gran gama de áreas en el proceso de visión, como reconocimiento de objetos, calibración de cámaras y visión robótica.

**Fuente:** URL <https://es.wikipedia.org/wiki/OpenCV>

## **MATLAB**

Es una herramienta de software matemático que ofrece un IDE con un lenguaje de programación propio. Entre sus funciones básicas están la del manejo de matrices, representación de datos y funciones matemáticas, algoritmos, permite la creación de interfaces gráficas de usuario (GUI).

Es un software muy usado en universidades y centros de investigación. Han aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL.

**Fuente:** URL <https://es.wikipedia.org/wiki/MATLAB>

## BIBLIOGRAFÍA

- [1] TICbeat. Alemania y Países Bajos permitirán probar coches autónomos en sus carreteras. 2015. Disponible en: <http://www.ticbeat.com/innovacion/alemania-paises-bajos-permitiran-probar-coches-autonomos-en-sus-carreteras/>
- [2] Fernández Caramés, Carlos. Técnicas de navegación para un robot móvil utilizando sistemas de razonamiento espacial. Salamanca, 2012. Universidad de Salamanca. Departamento de informática y automática. Disponible en: <http://gro.usal.es/web/thesis/tesisCF.pdf>.
- [3] Organización mundial de la salud. Estadísticas sanitaria mundiales 2014: una mina de información sobre salud pública mundial, 2014. Disponible en: [http://apps.who.int/iris/bitstream/10665/112817/1/WHO\\_HIS\\_HSI\\_14.1\\_spa.pdf](http://apps.who.int/iris/bitstream/10665/112817/1/WHO_HIS_HSI_14.1_spa.pdf).
- [4] Guerrero, Juan Felipe. Estadísticas muestran que los carros autónomos también se estrellan. 2015. Disponible en <http://www.ener.co/cultura-digital/autotecnologia/cuatro-accidentes-de-autos-autonomos-reportados-desde-septiembre/>. Fecha de consulta 15/05/15
- [5] Vargas Castillo, Diego Alfonso. Comportamiento de muertes y lesiones por accidente de transporte, Colombia, 2013. Disponible en: <http://www.medicinalegal.gov.co/documents/10180/188820/FORENSIS+2013+4-+accidentes+de+transporte.pdf/51867e30-9ab5-4a15-8363-f2232d2c86ae>. Fecha de consulta 15/05/15
- [6] Bejerano Pablo. La historia de los coches autónomos es más larga de lo que se podía esperar. Al avance de los años 80 y 90 le preceden algunos inventos ingeniosos surgidos décadas atrás. 2013. Disponible en: <http://blogthinkbig.com/historia-de-los-coches-autonomos/>
- [7] Ernst Dickmanns, el desconocido padre alemán de los coches inteligentes. El diario. Disponible en: [http://www.eldiario.es/hojaderouter/tecnologia/Ernst\\_Dickmanns-vehiculo-autonomo-inteligente\\_0\\_382511814.html](http://www.eldiario.es/hojaderouter/tecnologia/Ernst_Dickmanns-vehiculo-autonomo-inteligente_0_382511814.html). Fecha de consulta 17/05/15
- [8] Jochem Todd, Pomerleau Dean, Kumar Bala, Armstrong Jeremy. PANS: A Portable Navigation Platform, 1995. The Robotics Institute, the Carnegie Mellon University, Pittsburgh. Disponible en: [http://www.cs.cmu.edu/~tjochem/nhaa/navlab5\\_details.html](http://www.cs.cmu.edu/~tjochem/nhaa/navlab5_details.html). Fecha de consulta 20/05/15
- [9] Pete. Self-Drive Cars and You: A History Longer than You Think 2014. Disponible en: <http://www.velocetoday.com/self-drive-cars-and-you-a-history-longer-than-you-think/>

- [10] Stanford Racing Team. Stanford Racing Team's Entry In the 2005 DARPA Grand Challenge. Disponible en: <http://www.cs.stanford.edu/people/dstavens/darpa/Stanford.pdf>
- [11] Hundelshausen, Felix von. Team AnnieWAY's Autonomous System for the 2007 DARPA Urban Challenge. Journal of Field Robotics. 2008. Disponible en: [https://www.unibw.de/tas/institut/mitarbeiter/alumni/prof\\_von\\_hundelshausen/publications/kammel-jfr08-annieway-system.pdf](https://www.unibw.de/tas/institut/mitarbeiter/alumni/prof_von_hundelshausen/publications/kammel-jfr08-annieway-system.pdf)
- [12] FayerWayer. Google ha creado un automóvil que se maneja solo. 2014. Disponible en: <https://www.fayerwayer.com/2014/05/google-muestra-un-nuevo-auto-que-se-maneja-solo-disenado-completamente-por-ellos/>. Fecha de consulta 22/05/15
- [13] ABC. Volvo prueba en España su sistema de piloto automático para coches. 2012. Disponible en: <http://www.abc.es/20120529/tecnologia/abci-coche-piloto-automatico-201205290811.html>. Fecha de consulta 20/05/15
- [14] HÍBRIDOS Y ELÉCTRICOS, Ecotecnología del vehículo. El coche eléctrico autónomo de Nissan, en 2016. 2014. Disponible en: <http://www.hibridosyelectricos.com/articulo/tecnologia/coche-electrico-autonomo-nissan-2016/20140719000725007531.html>. Fecha de consulta 22/05/15
- [15] DIARIOMOTOR. Tesla quiere y puede lanzar un coche autónomo, pero no les dejan, y se conforman con un Model S que cambia solo de carril. 2014. Disponible en: <http://www.diariomotor.com/2014/10/10/tesla-autopilot-semi-autonomo/>
- [16] MICROSIERVOS. Un coche autónomo gana en circuito a un piloto amateur. 2015. Disponible en: <http://www.microsiervos.com/archivo/tecnologia/coche-autonomo-gana-circuito-a-piloto-amateur.html>. Fecha de consulta 26/05/15
- [17] LR. La república. Mercedes, BMW y Toyota presentan los carros autónomos del CES. 2015. Disponible en: [http://www.larepublica.co/mercedes-bmw-y-toyota-presentan-los-carros-aut%C3%B3nomos-del-ces\\_207501](http://www.larepublica.co/mercedes-bmw-y-toyota-presentan-los-carros-aut%C3%B3nomos-del-ces_207501)
- [18] Hipertextual. Toyota y Audi mostraran sus vehículos autónomos en el CES. 2013. Disponible en: <http://hipertextual.com/2013/01/toyota-audi-vehiculos-autonomos-ces>
- [19] Dvaruh. Guardium Unmanned Ground Vehicle (UGV). 2014. Disponible en: <http://techmafia.net/robotics/2014/06/13/guardium-unmanned-ground-vehicle-ugv.html>. Fecha de consulta 26/05/15
- [20] Woman Power Fights Hamas with 'Unmanned Ground Vehicles'. 2012. Disponible en: [http://www.israelnationalnews.com/News/News.aspx/162982#.Vf3wqt9\\_Okq](http://www.israelnationalnews.com/News/News.aspx/162982#.Vf3wqt9_Okq).

- [21] Israeli Robotic hour Rises in operation?. 2008. Disponible en: <https://xi0.info/israeli-robotic-hour-rises-operation.html>
- [22] BERMUDEZ, Giovanni. Robots móviles: teoría aplicaciones y experiencias. 2002. p.10
- [23] CARDENAS RUEDA, Miguel Ángel. Localizaçãõ Híbrida para Um Veículo Autônomo em Escala Usando Fusão de Sensores. Campinas, 2012.
- [24] FERNANDEZ LANCHAS, Miguel. FERNANDEZ SANZ, David. VALSAMEDA PLASENCIA, Carlos. Planificación de trayectorias para un robot móvil. 2009-2010. p. 21
- [25] Informática hoy. Que es el GPS y como funciona. Disponible en: <http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-el-GPS-y-como-funciona.php>. Fecha de consulta 25/09/2015
- [26] Semar. Percepción remota. Disponible en: <http://www.semar.gob.mx/s/informacion-sector/ciencia/ermexs/percepcion-remota.html>. Fecha de consulta 25/09/2015
- [27] Julio Martínez Muñoz. Percepción remota “Fundamentos de la teledetección espacial”. Disponible en: <http://siga.cna.gob.mx/SIGA/Percepcion/Fundamentos%20de%20teledetecci%C3%B3n%20espacial.PDF>. Fecha de consulta 25/09/2015
- [28] Torres Medina Fernando. Percepción en Robótica. Universidad de Alicante. Fecha de consulta 25/09/2015
- [29] Controladores PID. Disponible en: <http://www.eng.newcastle.edu.au/~jhb519/teaching/caut1/Apuntes/PID.pdf>. Fecha de consulta 05/10/2015
- [30] *The Ziegler Nichols Method*. Disponible en: <https://www.youtube.com/watch?v=nvAQHSe-Ax4>. Fecha de consulta 05/10/2015
- [31] Control PID Clásico. Disponible en: [http://csd.newcastle.edu.au/SpanishPages/clase\\_slides\\_download/C07.pdf](http://csd.newcastle.edu.au/SpanishPages/clase_slides_download/C07.pdf). Fecha de consulta 05/10/2015
- [32] *Interpolación y extrapolación lineal SECUNDARIA*. Disponible en: <https://www.youtube.com/watch?v=pzDfi2jVDGs>. Fecha de consulta 05/10/2015
- [33] Seco Granja Fernando. Instituto de Automática Industrial – CSIC. Sistemas de localización en interiores basados en radiofrecuencia Disponible en:



<http://www.car.upm-csic.es/lopsi/static/publicaciones/docencia/Apuntes%20RF-LPS.pdf>. Fecha de consulta 29/10/2015

[34] Planificación de Trayectorias para Robots Móviles capítulo 2, 2,4

[35] Visión artificial. Aspectos de un Sistema de Visión Artificial. 2005. Universidad Nacional de Quilmes – Ing. En Automatización y Control Industrial.

[36] Espectro electromagnético. Espectro visible. Disponible en: [http://educativa.catedu.es/44700165/aula/archivos/repositorio/3000/3233/html/21\\_espectro\\_visible.html](http://educativa.catedu.es/44700165/aula/archivos/repositorio/3000/3233/html/21_espectro_visible.html)

[37] Historia y biografías. La vista: órganos de los sentidos humanos vista, oído, gusto, olfato, tacto. Disponible en: <http://historiaybiografias.com/sentido1/>

[38] Avilés, Asturias. Óptica, el ojo humano. IES la Magdalena. Disponible en: <http://web.educastur.princast.es/proyectos/fisquiweb/Apuntes/Apuntes2Fis/Ojo.pdf>

[39] Vaqué Alcázar, Carlos, De Andrés. El ojo humano, ¿Cómo funciona? Disponible en: <http://cienciainquieta.com/2013/12/02/el-ojo-humano-como-funciona/>

[40] Visión artificial. Aplicación práctica de la visión artificial en el control de procesos industriales. 2012.

[41] Evolución de los sensores digitales en fotografía, de CCD a CMOS. 2015. Disponible en: <http://dgpfotografia.com/2015/06/24/evolucion-de-los-sensores-digitales-en-fotografia-de-ccd-a-cmos/>

[42] Platero, Carlos. Introducción a la Visión Artificial. Dpto. Electrónica, Automática e Informática Industrial.

[43] E. de la fuente F.M. trespaderne. Imágenes binarias. Disponible en: <http://www.librovision.eii.uva.es/framesContenidos.html>

[44] Glosario Grafico. Un diccionario de artes gráficas, diseño y materias a fines. Disponible en: [http://glosariografico.com/escala\\_grises](http://glosariografico.com/escala_grises)

[45] Hablamos de comunicación multimedia. Modos o modelos de color LAB e indexado: que son y usos específicos. 2010. Disponible en: <http://docs.gimp.org/es/gimp-images-in.html>

[46] Hablamos de comunicación multimedia. Modos o modelos de color RGB y CMYK: que son y usos específicos. 2010. Disponible en: <http://www.comunicacion-multimedia.info/2010/05/modos-o-modelos-de-color-rgb-y-cmyk-que.html>

- [47] Aguilar Carrera, Gustavo German. Procesamiento digital de imágenes utilizando filtros morfológicos. Diciembre 1995. Escuela Politécnica Nacional
- [48] Visión artificial e interacción sin mandos. 2010. Asignatura de gráficos en computación.
- [49] Visión Artificial. Tema 2. Preprocesado (realzado y filtrado) de imágenes digitales. Universidad rey Juan Carlos.
- [50] Filtrado Espacial. 2005. Universidad Nacional de Quilmes. Ing. en Automatización y Control Industrial. Visión Artificial.
- [51] C. Pinilla, A. Alcalá, F Ariza. Revista de Teledetección. Filtrado de imágenes en el dominio de la frecuencia. Universidad de Jaén. Departamento de ingeniería Cartográfica, Geodésica y Fotogrametría.
- [52] Hernández Rubén, Bernárdez Giovani, Vaqueiro Janito, Siqueira Pablo. Planeación y seguimiento de trayectorias de un vehículo terrestre con base en el control de dirección en un ambiente real, 2014. Disponible en: <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/8287/5874>
- [53] Massera Filho, Carlos Alberto. Controle e Tomada de Decisao de Veículos Autonomos nos Limites da Estabilidade. 2015. ICMC-USP, pp. 15-20.
- [54] Snider, Jarrod M. Automatic Steering Methods for Autonomous Automobile Path Tracking. 2009. Carnegie Mellon University, pp. 28-31.
- [55] De Pavia, Azinaheira Jose, Bueno Samuel. Controle De Trajetória Para Veículos Terrestres De Exterior. 2010.
- [56] Castillo Oscar, Duran Eyner. Localización de un robot móvil de configuración Ackerman usando el filtro extendido de kalman. Universidad Nacional de ingeniería, pp. 2.
- [57] Beal, Craig E. Applications of Model Predictive Control to Vehicle Dynamics for Active Safety and Stability. 2011, pp. 38-46.
- [58] Rajamani, Rajesh. Vehicle Dynamics and Control. Springer. 2006, pp. 64-66.
- [59] G. Ejarque, R. Haarth, M. Diestefano. Mecanismo de control de rotación de una rueda esférica. Facultad de ingeniería. Universidad Nacional de Cuyo. Disponible en <http://taee.euitt.upm.es/actas/2010/papers/2010S11A04.pdf>
- [60] Alessandro De Luca, Giuseppe Oriolo. Feedback Control of a Nonholonomic Car-Like Robot. Universita di roma "La Sapienza". INRIA, Sophia-Antipolis. Departamento di informática e Sistemistica, pp. 11-14.

- [61] Cardenas Rueda, Miguel Ángel. Localização Híbrida para Um Veículo Autônomo em Escala Usando Fusão de Sensores. Campinas, 2012. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica.
- [62] MathWorks. The Language of Technical Computing. Disponible en: <http://www.mathworks.com/products/matlab/>
- [63] MathWorks. Simulation and Model-Based Design. Disponible en: <http://www.mathworks.com/products/simulink/>
- [64] Fuentes de error GPS. Disponible en: <http://www.alsitel.com/tecnico/gps/errores.htm>
- [65] Apuntes de visión Artificial. Segmentación, detección de líneas rectas. Departamento de electrónica, Automática e Informática industrial.
- [66] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 886–893, IEEE, 2005.