

TIE  
043  
2012  
Ej.1

0 9 2 5

**ACCIONAMIENTO REAL A DISTANCIA DE UN  
SERVOMECANISMO MEDIANTE UN MUNDO  
VIRTUAL CONTROLADO POR INTERFAZ GESTUAL**

BIBLIOTECA - USTA  
TUNJA

//  
**DIEGO IVÁN OLIVEROS ACOSTA**

0 9 2 5

**UNIVERSIDAD SANTO TOMÁS DE AQUINO ✓  
DIVISIÓN DE INGENIERÍAS Y ARQUITECTURA ✓  
FACULTAD DE INGENIERÍA DE SISTEMAS ✓  
FACULTAD DE INGENIERÍA ELECTRÓNICA ✓  
TUNJA ✓  
2012 ✓**

**ACCIONAMIENTO REAL A DISTANCIA DE UN  
SERVOMECANISMO MEDIANTE UN MUNDO  
VIRTUAL CONTROLADO POR UNA INTERFAZ GESTUAL**

**DIEGO IVÁN OLIVEROS ACOSTA**

**Asesores:**

**Msc. Ing. LUZ SANTAMARÍA GRANADOS**  
Docente de la facultad de Ingeniería de Sistemas  
**Ms(c). Ing. EDGAR ANDRÉS GUTIÉRREZ CÁCERES**  
Docente de la facultad de Ingeniería de Electrónica

**UNIVERSIDAD SANTO TOMÁS DE AQUINO  
DIVISIÓN DE INGENIERÍAS Y ARQUITECTURA  
FACULTAD DE INGENIERÍA DE SISTEMAS.  
FACULTAD DE INGENIERÍA ELECTRÓNICA.  
TUNJA  
2012**

Nota de aceptación:

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

Tunja, 15 de marzo de 2012

Solamente el autor es responsable de las ideas y tesis expuestas en este trabajo.

---

A mis padres:  
Joaquín Oliveros Ayala y  
Nohora Cecilia Acosta Álvarez  
Por su apoyo incondicional, su amor, los  
principios y valores mostrados.

## AGRADECIMIENTOS

Al Ingeniero Juan Francisco Mendoza Moreno por su ejemplo de don de gente, inteligencia emocional y emprendimiento.

Al Ingeniero Miguel Ángel Mendoza por su dedicación y método pensamiento sistemático rígido sus ideas simples precisas y completas.

Al Ingeniero Oscar Umaña por la paciencia y don de padre.

Al Ingeniero Davis Montenegro por enseñarme que no hay porque reinventar la rueda.

Al licenciado Carlos Uribe amor y pasión por la física.

Ingeniero Luis Fredy Sosa por sus interminables talleres y el desarrollo del espíritu crítico.

Al Ingeniero Fabián Jiménez por su orden metodología y pensamiento estructurado.

A mis tutores Edgar Andrés Gutiérrez y en especial a la Ingeniera Luz Santamaría Granados su responsabilidad e incansable trabajo.

### A LA UNIVERSIDAD

En especial al padre Tiberio Polanía Ramírez, O.P. por su excelente labor como Vicerrector Académico y su motivación al desarrollo académico e investigativo de la Universidad Santo Tomas Aquino.

## CONTENIDO

Introducción .....	19
1. Planteamiento del problema.....	21
1.1 Formulación del problema .....	22
2. Justificación .....	23
3. Objetivos .....	24
3.1 General .....	24
3.2 Específicos .....	24
4. Marco referencial.....	25
4.1 Teórico .....	25
4.2 Histórico .....	26
4.3 Conceptual.....	28
4.3.1 Computación gráfica.....	28
4.3.2 Realidad virtual.....	35
4.3.3 Comunicaciones inalámbricas .....	40
4.3.4 Robótica.....	42
4.3.5 Interfaz de comunicación.....	47
5. Estado del arte.....	50
6. Diseño Metodológico.....	51
6.1 La Investigación aplicada.....	51
6.2 Pregunta de investigación: .....	51
6.3 Metodología de investigación: .....	52
6.4 Hipótesis.....	52
6.5 Variables: .....	52
6.6 Modelo en espiral[9]: .....	53
6.7 Módulo 1 Interface gestual:.....	55
6.7.1 ROPES (Rapid Object-Oriented Process for Embedded Systems).....	55
6.7.2 Programación extrema ( <i>extreme programming</i> , xp).....	56
6.8 Módulo 2 Comunicación Inalámbrica para la tele operación. ....	59
6.8.1 Diseño de red top-down :[30][31].....	59
6.9 Módulo 3 Mundo Virtual. ....	61
6.9.1 Metodología de desarrollo VRML Efectivo:[32] [33].....	61

6.10	Módulo 4 Servomecanismo.....	64
6.10.1	Método RCM (mantenimiento centrado en la fiabilidad)[34][35] .....	64
7.	Resultados.....	65
7.1	Ingeniería del software.....	65
7.1.1	Arquitectura del sistema .....	65
7.1.2	Arquitectura del software:.....	66
7.1.3	Diseño de interfaces:.....	67
7.2	Computación gráfica .....	68
7.2.1	Primera fase: Interface Gestual (desarrollo de la Metodología Xp y Ropes) 68	
7.3	Realidad virtual (Segunda fase: Mundo virtual) .....	72
7.3.1	Diseño de VRML Efectivo .....	72
7.4	Comunicación inalámbrica (Interfaz de comunicación pc a módulo inalámbrico)80	
7.4.1	Objetivo: .....	80
7.4.2	Herramientas: .....	80
7.4.3	Desarrollo:.....	80
7.4.4	Pruebas.....	81
7.4.5	Desarrollo de la metodología top-down.....	82
7.5	Robótica (Cuarta fase: servomecanismo).....	84
7.5.1	Sistema mecánico.....	84
7.5.2	Sistema Electrónico .....	85
7.6	Integración (desarrollo del primer prototipo).....	94
7.6.1	Desarrollo.....	94
7.6.2	Pruebas.....	94
8.	Conclusiones:.....	95
9.	Recomendaciones y trabajo futuro.....	96
10.	Bibliografía .....	97
11.	Anexos .....	100
12.	Asesores .....	156

## LISTA DE ILUSTRACIONES

Ilustración 1 Interface gestual kinect .....	29
Ilustración 2 Captura cámara y distribución de probabilidad de la piel .....	31
Ilustración 3 Personaje virtual 3D sobre una tarjeta de patrón real .....	33
Ilustración 4 Sistema inmersivo .....	35
Ilustración 5 Sistema no inmersivo .....	36
Ilustración 6 Entorno de realidad inmersiva .....	36
Ilustración 7 Diagrama de pines PIC18F4550 .....	43
Ilustración 8 Herramientas para análisis y diseño electrónico .....	44
Ilustración 9 Inteface de programacion PCW .....	45
Ilustración 10 Logo Microchip .....	45
Ilustración 11 Logo PICkit2 .....	46
Ilustración 12 Tarjeta arduino 1 .....	48
Ilustración 13 Diseño Metodológico .....	53
Ilustración 14 Metodología en espiral .....	54
Ilustración 15 Ciclo de vida ROPES .....	56
Ilustración 16 Customer story and task card .....	57
Ilustración 17 Ciclo de diseño e implementación de una red .....	59
Ilustración 18 Proceso de diseño VRML .....	63
Ilustración 19 Arquitectura del sistema .....	66
Ilustración 20 Arquitectura del software .....	66
Ilustración 21 Diseño de interfaz del sistema .....	67
Ilustración 22 Intervalos de control .....	68
Ilustración 23 Definición de coordenadas .....	70
Ilustración 24 Head Tracking .....	70
Ilustración 25 Traking con una pelota azul .....	71
Ilustración 26 Parachoques del robot .....	73
Ilustración 27 Antena del robot .....	73
Ilustración 28 Ruedas Robot .....	73
Ilustración 29 Chasis del robot .....	73
Ilustración 30 Vista superior del chasis .....	74
Ilustración 31 Agarres del robot .....	74
Ilustración 32 Soporte para motores .....	74
Ilustración 33 Esquema general del brazo .....	74
Ilustración 34 Pinza del robot .....	74
Ilustración 35 Diagrama general .....	74
Ilustración 36 Modelo 3D del avatar texturizado .....	75
Ilustración 37 Armadura del brazo robótico en Modo edición .....	75
Ilustración 38 Detalle del motor .....	76
Ilustración 39 Detalle base .....	76
Ilustración 40 Detalle pinza .....	76
Ilustración 41 Diagrama general .....	76
Ilustración 42 Vista lateral, trasera, superior y perspectiva ortogonal del avatar .....	77
Ilustración 43 Selección de objetos en el modo edición .....	77

Ilustración 44 Lógica para el servomecanismo.....	78
Ilustración 45 Perspectiva ortogonal del robot con emparentado.....	78
Ilustración 46 Renderizado en versión 4.9.....	79
Ilustración 47 Proceso de instalación puerto COM o USB simulado.....	81
Ilustración 48 Diseño de la red lógica.....	82
Ilustración 49 Diseño de la red física.....	82
Ilustración 50 Configuración inicial.....	84
Ilustración 51 Modificación.....	84
Ilustración 52 Soldadura.....	84
Ilustración 53 Reconstrucción de soporte.....	84
Ilustración 54 Mantenimiento.....	84
Ilustración 55 Diagrama circuital driver motores DC.....	86
Ilustración 56 Diagrama circuital.....	87
Ilustración 57 auto ruteo en ares.....	87
Ilustración 58 Diseño de impresión por capas.....	87
Ilustración 59 Capa de cobre.....	88
Ilustración 60 Capa de etiquetas.....	88
Ilustración 61 Materiales para la impresión del tóner en la placa de cobre.....	88
Ilustración 62 Proceso de remoción del papel fotográfico.....	89
Ilustración 63 Proceso de remoción del papel fotográfico Fuente: el autor.....	89
Ilustración 64 Tóner impreso sobre la placa de cobre Fuente: el autor.....	89
Ilustración 65 Proceso de quemado.....	90
Ilustración 66 Proceso de quemado con cloruro férrico Fuente: El autor.....	90
Ilustración 67 Diferencia entre el ácido nítrico y el cloruro férrico.....	90
Ilustración 68 Detalle motor con aguja hipodérmica Fuente: El autor.....	91
Ilustración 69 Motor con aguja hipodérmica.....	91
Ilustración 70 Base para taladro casera.....	91
Ilustración 71 Base para taladro.....	91
Ilustración 72 Mango de fuerza del taladro.....	91
Ilustración 73 Baquela perforada.....	92
Ilustración 74 Socket (Porta integrado).....	92
Ilustración 75 Sockets y conectores.....	92
Ilustración 76 Puntos de soldadura baquela.....	92
Ilustración 77 Pruebas de funcionalidad del circuito.....	93
Ilustración 78 Producto final.....	93

## LISTA DE TABLAS

Tabla 1 Características del microcontrolador 18F4550 .....	44
Tabla 2 Intervalos de control .....	68
Tabla 3 Lógica interface gestual .....	72
Tabla 4 Planos iniciales del avatar .....	74
Tabla 5 Fotografías base para el diseño de texturas .....	76
Tabla 6 Señal de control .....	80
Tabla 7 Modificación de código para señal de control .....	83
Tabla 8 Comandos para el robot .....	85
Tabla 9 Componentes C1L258 .....	86
Tabla 10 Código rastreo de objetos .....	131
Tabla 11 Código interface Arduino uno .....	135
Tabla 12 Código microcontrolador .....	138

## LISTA DE ANEXOS

Anexo A. Metodología RCM.....	101
Anexo B. Análisis de riesgos.....	103
Anexo C. Análisis de requisitos.....	112
Anexo D. Pruebas .....	118
Anexo E. Algoritmo Head Traking.....	124
Anexo F. Descripción de clases.....	125
Anexo G. Desarrollo Módulo interface Gestual .....	127
Anexo H. Comunicación PC a modulo inalámbrico.....	132
Anexo I. Servomecanismo .....	136
Anexo J. Datasheet L293D .....	139
Anexo K. Manual Básico de Blender .....	140
Anexo L. Artículo.....	151

## GLOSARIO

**Servomecanismo:** sistema electromecánico que se regula por sí mismo al detectar el error o la diferencia entre su propia actuación real y la deseada.[1]

**Mundo Virtual:** es una representación del mundo real, mediante sistemas que buscan emular algunas de las sensaciones visuales, táctiles, auditivas, gustativas y/o del olfato.

**Interfaz gestual:** es el hardware o software que permite hacer comunicación y adaptación entre los gestos del ser humano y el computador, es decir que la interfaz actúa como un traductor(mediador) entre los dos sistemas, volviéndolos sensibles entre sí[2].

**Píxel:** procede de la contracción de la palabra inglesa *picture element* por lo tanto no es una unidad de medida, sino que se trata en realidad de un elemento de la imagen como viene a indicar su origen.

**Vóxel:** (Del inglés volumetric pixel) es la unidad cúbica que compone un objeto tridimensional. Constituye la unidad mínima procesable de una matriz tridimensional y es, por tanto, el equivalente del píxel en un objeto 2D.[3]

**Captura:** este proceso se puede realizar mediante muchas técnicas, una cámara digital, escáner o cualquier agrupamiento de sensores o transductores ópticos.

**Pos-proceso:** una vez capturada estos datos, se puede llegar a obtener información valiosa como velocidad, caudal, volumen, calor y un sin número de propiedades físicas mediante algoritmos y técnicas matemáticas.

**Inmersión:** enfoque selectivo en la emulación de la sensación física mediante la cual se percibe el mundo real.

**Interacción virtual:** es la forma como se navega en el mundo virtual, en otras palabras el movimiento y acción dentro del mundo virtual, ya sea la locomoción común, la capacidad de volar, si se habla de primera persona o segunda en el último caso la persona puede ver un avatar que emula la presencia de el mismo en el escenario.

**Tridimensionalidad:** son las transformaciones gráficas, de los objetos de acuerdo a la posición en el espacio virtual y la iluminación correspondiente.

**Sistemas de Tele presencia:** término creado por Marvin Minsky, es un entorno que proporciona a la persona la idea de estar físicamente en otro lugar por medio de sensaciones creadas artificialmente.

**Realidad aumentada:** es el término que se usa para definir una visión directa o indirecta de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real.

**Videojuego:** es un software creado para el entretenimiento en general y basado en la interacción entre una o varias personas y un aparato electrónico que ejecuta dicho videojuego usualmente el término "video" en la palabra "videojuego" se refiere en sí a un visualizador de gráficos rasterizados.

**Micro controlador:** es un chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, memoria y unidades de E/S, es decir, se trata de una computadora completa en un solo circuito integrado.

**Circuito integrado (CI):** también conocido como chip o microchip, es una pastilla pequeña de material semiconductor, de algunos milímetros cuadrados de área, sobre la que se fabrican circuitos electrónicos generalmente mediante fotolitografía y que está protegida dentro de un encapsulado de plástico o cerámica. El encapsulado posee conductores metálicos apropiados para hacer conexión entre la pastilla y un circuito impreso.

## RESUMEN

Los sistemas tele-operados son de gran utilidad en la industria, ámbito militar, médico, investigativo así como también en el mundo comercial. Pero requieren gran capacidad de procesamiento debido al manejo de un número significativo de datos, sobretodo en labores de rescate por desastres naturales donde el ser humano no tiene acceso y además los sistemas de información son improvisados[4]. Es por esto que se hacen necesarias estrategias que hagan que la tele-operación no solo sea óptima para solventar estos problemas si no que pueda significar un desarrollo comercial.

El desarrollo de este tipo de tecnología requiere de: una buena administración; planeación, dirección y control, una adecuada selección de una metodología de desarrollo y enfrentarse a las características del trabajo interdisciplinar.

El presente libro, describe la investigación de las tecnologías utilizadas para la elaboración del sistema, expone la estrategia que minimiza el número de información como lo es la utilización de mundo virtual además muestra la técnica que ayuda al manejo intuitivo del control, llamada interfaz gestual.

Para el diseño 3D se usa Blender, para la comunicación inalámbrica el codificador PT2262 y decodificador PT2272, el manejo de las tramas de bits para el control del robot se realiza por medio del microcontrolador 18F4550 y el puente L293D. En cuanto al servomecanismo se repara el robot del grupo de investigación GINSCON de la Facultad de Ingeniería Electrónica, para el reconocimiento de imágenes se usan las librerías de OpenCV, para la comunicación PC-micro se usa el kit Arduino uno.

Este es el primer proyecto interdisciplinar como opción de grado para doble titulación, se presentó como ponencia en el Segundo Congreso Internacional de Instrumentación, Control y Telecomunicaciones junto con el artículo titulado: Accionamiento Real a Distancia de un Servomecanismo mediante un Mundo Virtual Controlado por Interfaz Gestual.

Este libro está dividido básicamente en 4 partes, la primero describe la parte referente a la planeación; los objetivos, justificación luego se realiza un apoyo conceptual de las tecnologías y metodologías manejadas, el tercero muestra el desarrollo metodológico utilizado en la ejecución de todas sus fases y finalmente unos anexos; aportes, graficas, manuales y hojas de datos.

## ABSTRACT

Tele-operated systems are useful in industry, military, medicine, researchers as well as in the commercial world. But they require high processing capacity due to handling a significant number of data, especially in rescue by natural disasters in which human beings have no access and information systems also are improvised. Due to that, strategies are necessary to render the tele-operation not only is optimal for solving these problems if it might mean a commercial development.

The development of this technology requires: good management, planning, direction and control, a proper selection of a development methodology and address the characteristics of interdisciplinary work.

This book describes research on the technologies used to develop the system, outlines the virtual world as a strategy that minimizes the number of information, also using gestural interface technique shows a way for intuitive control.

Blender is used for 3D design, wireless communication is developed by encoder and decoder PT2262 and PT2272, the management of bit patterns to control the robot is performed by the 18F4550 microcontroller and the bridge L298. As for servomechanism is made the maintenance of the caterpillar robot from the research group of Faculty Electronic Engineering (GISCON), for the recognition of images is used OpenCV libraries for communication between microcontroller and computer is used an Arduino kit.

This interdisciplinary project is the first grade as an option for double degree, was presented as a paper at the Second International Congress on Instrumentation, Control and Communications together with the article titled: "*Accionamiento Real a Distancia de un Servomecanismo mediante un mundo Virtual Controlado por Interfaz Gestual.*".

This book is basically divided into 4 chapters, the first describes the part concerning the planning, the objectives, justification, after makes conceptual support of the technologies and methodologies used, the third shows the development methodology used in the execution of all phases and finally annexes; contributions, charts, manuals and datasheets.

## PROLOGO

El proyecto de fin de carrera es una de las etapas más importantes en la culminación de la formación del Ingeniero de Sistemas, ya que le permite contribuir en la construcción del conocimiento desde las distintas áreas disciplinares, pero haciendo énfasis en la profundización de temas que le den los lineamientos para hacer la formulación de proyectos de investigación como el propuesto en este documento: "Accionamiento real a distancia de un servomecanismo mediante un mundo virtual controlado por una interfaz gestual". El proyecto hace la integración de herramientas tecnológicas del área computacional de la realidad virtual, la computación gráfica, las comunicaciones inalámbricas y la robótica, para generar el prototipo del sistema tele-operado. Este dispendioso trabajo le permitió al investigador fortalecer su proceso de aprendizaje en las técnicas y modelos para desarrollar los algoritmos que facilitaron la comunicación de la interfaz gestual con el robot y el escenario virtual.

Ms. Ing. LUZ SANTAMARÍA GRANADOS

**ACCIONAMIENTO REAL A DISTANCIA DE UN SERVOMECANISMO  
MEDIANTE UN MUNDO VIRTUAL CONTROLADO POR UNA INTERFAZ  
GESTUAL**

## INTRODUCCIÓN

La presente investigación se refiere al tema de tele operación de dispositivos robóticos con reducción del volumen de datos de control mediante mundos virtuales y aumento características de inmersión mediante interfaces gesticulares. La tele operación como su nombre lo indica es el manejo a distancia, para lo que se recurre a sistemas de gobierno de señales de radiofrecuencia. Las interfaces gestuales son aquellas que permiten mediante gestos, la comunicación humano máquina. Esta máquina, ya sea el computador o circuitos micro controlados envían las señales a los dispositivos electromecánicos o el robot que se quieren operar en un lugar remoto. La generación de un entorno virtual y el prototipo, posibilitan el estudio de las hipótesis planteadas.

Las características más relevantes son la interdisciplinariedad, el trabajo con software y hardware libre, la adecuada selección de la metodología de trabajo, la delimitación de objetivos, su contribución al trabajo en equipo y de productos tecnológicos que ayuden al fortalecimiento de las líneas de investigación interdisciplinar.

Además a nivel profesional este tipo de proyectos de enfoque gerencial abren las puertas a las labores de planeación organización y ejecución de proyectos que en el sector empresarial e investigativo serán de mucha utilidad.

El tipo de investigación que maneja este trabajo es aplicada, esta comprende básicamente 8 fases; antecedentes, hipótesis y objetivos, información utilizada, método de análisis, resultados, discusión, conclusiones y referencias bibliográficas.

En los antecedentes o estado de arte, se describen los proyectos relacionados con la tele operación, diseño 3D, imagenología[3] que son los componentes teóricos más fuertes dentro del proyecto y en lo posible se buscan proyectos que conjuguen este tipo de tecnologías en un porcentaje significativo o que incurran en el mismo campo de aplicación.

Con relación a la hipótesis, se cuestiona la posible integración de las tecnologías relacionadas con las áreas de realidad virtual, tele-operación y reconocimiento de imágenes, para el desarrollo de sistemas de tele presencia en cualquier área específica.

La primera fase de la investigación corresponde al desarrollo del prototipo que permita realizar estas evaluaciones para lo cual se plantean los objetivos que delimitan esta tarea y serán discutidos unas páginas más adelante.

La información utilizada es el punto de partida y corresponde al banco o bases de datos que se encuentra descrita en el marco teórico, es claro que este trabajo solo cubrirá la primera parte del ciclo de vida de la investigación que es el desarrollo del prototipo por lo que el proceso metodológico y el análisis se centraran en el avance de los objetivos en cuanto a la evaluación de investigación y la hipótesis serán evidenciadas a través de las variables usabilidad, compatibilidad e interoperabilidad de las herramientas utilizadas.

Los resultados, discusión, conclusiones y referencias bibliográficas se encuentran en el último capítulo.

## 1. PLANTEAMIENTO DEL PROBLEMA

Actualmente la comunidad científica trabaja en el desarrollo de sistemas óptimos para el intercambio de datos, de manera que estos garanticen la fiabilidad y entrega oportuna de la información. Para obtener sistemas que mejoran la calidad de vida, minimizar el riesgo de accidente, mejorar la producción en actividades humanas y ampliar el dominio conocimiento.

La labor de ingeniería busca mejorar la interacción entre los sistemas y el hombre, buscando que los dispositivos sean cada vez más ergonómicos, menos invasivos y reduciendo su tiempo de respuesta.

Este tipo de proyectos exigen de un trabajo interdisciplinario continuo; desde el análisis, diseño y simulación de los sistemas electrónicos que controlan el servomecanismo, así como los correspondientes a su construcción, pasando por la ingeniería de software para desarrollar la aplicación que administre los datos obtenidos, hasta la réplica desarrollada en entornos de realidad virtual que aumenta el grado de inmersión.

Los desastres naturales acaecidos recientemente, por ejemplo el terremoto de magnitud 7.0 que azotó a Haití el 12 de enero de 2010 con un resultado trágico 250.000 muertos[5], o el terremoto y tsunami sucedido el 11 de marzo en Japón[6]. Son dos escenarios que muestran la urgente necesidad de acompañar, ayudar o incluso sustituir al hombre en las tareas de salvamento bajo situaciones extremas de peligro.

Esta necesidad se convierte en reto para los profesionales en ingeniería. Además de los anteriores, la exploración de planetas, minas y mundo submarino, despiertan el interés de la ingeniería, por lo que se hace necesario suministrar de tecnología necesaria, para que el hombre pueda cumplir tales tareas.

Precisamente, la realidad virtual da respuesta a estos interrogantes por cuanto puede emular el escenario de lugares remotos como operaciones médicas, militares, antiexplosivas y otras de alto riesgo para el ser humano. Así, junto con los sistemas tele-operados, la realidad virtual inmersiva puede salir de un entorno computarizado para proveer interacción con el mundo real, logrando magníficos resultados.

Utilizar este tipo de tecnología para solucionar problemas en el ámbito regional y contribuir a la investigación de nuevas tecnologías, podría ser factor clave en su desarrollo, de esta forma se puede facilitar a los miembros de la comunidad que aprovechen el avance tecnológico, la globalización, la programación de alto nivel y los paquetes de software especializado, junto con la formación de sus profesionales y los procesos investigativos adelantados por esta comunidad[7]. Asimismo contribuirían enormemente a evitar que la región no solo se caracterice por su riqueza en la parte agrícola, turismo, artesanías, entre otros, sino que además deje el rezago tecnológico y la falta de oportunidades de progreso científico.

## **1.1 FORMULACIÓN DEL PROBLEMA**

¿Es posible crear un prototipo de accionamiento real a distancia de un servomecanismo, mediante un mundo virtual controlado por una interfaz gestual?

## 2. JUSTIFICACIÓN

El desarrollo en áreas como la telemedicina, sistemas de diagnóstico para la rehabilitación, mundos virtuales, videojuegos, robótica y los aportes al mejoramiento de la misma en pro del beneficio humano en el mundo, son áreas de investigación adelantadas por países desarrollados, a pesar que en los países en vía de desarrollo se cuenta con el talento humano y las necesidades para aplicar estas tecnologías a la propia región.

A nivel comercial, existe una gran demanda de dispositivos de realidad virtual inmersiva, un ejemplo de ello son los videojuegos, que se apoyan en software de reconocimiento de imágenes para mejorar sus productos, hacerlos más reales y ergonómicos. *“La corporación Microsoft vendió en 4 meses más de 10 millones de controladores Kinect para la consola Xbox, que permiten a los usuarios jugar directamente sin mandos. Estos resultados permitieron incluir el aparato en el Libro Guinness de los récords como el dispositivo electrónico de consumo con mayor velocidad de ventas”*[8].

La Facultad de Ingeniería Electrónica de la Universidad Santo Tomás de Tunja ha adelantado investigación acerca del procesamiento de imagen así como también prácticas de control de los servomecanismos. El software de reconocimiento de imagen y los servomecanismos permiten en conjunto, desarrollar investigación que genera aportes significativos, coherentes con las nuevas tecnologías y las necesidades actuales.

Por su parte la Facultad de Ingeniería de Sistemas de la misma universidad tiene experiencia en la creación de mundos virtuales, que junto con la tecnología mencionada aportan experiencia para implementar sistemas de tele presencia y juegos de realidad virtual inmersiva.

Aprovechando las fortalezas y los espacios mencionados anteriormente es pertinente desarrollar un prototipo que contribuya al enriquecimiento de esta área de investigación, a la creación de grupos de investigación interdisciplinarios y que contribuya al proceso de formación académico.

### **3. OBJETIVOS**

#### **3.1 GENERAL**

Desarrollar un prototipo que permita el accionamiento real a distancia de un servomecanismo mediante un mundo virtual controlado por una interfaz gestual.

#### **3.2 ESPECÍFICOS**

Diseñar un mundo 3D que sirva de medio para emular el comportamiento de un servomecanismo real.

Realizar una interface gestual para mejorar el grado de inmersión al mundo virtual.

Controlar un servomecanismo de forma remota con el fin de materializar las acciones del robot virtual.

Integrar las tecnologías mencionadas para desarrollar el prototipo y la documentación que aporte al proceso investigativo de la universidad.

## 4. MARCO REFERENCIAL

El marco referencial es la base y apoyo necesarios para entender este libro, se divide en marco teórico, histórico y conceptual. El primero ilustra la prospectiva, el segundo la retrospectiva “Quien olvida su historia está condenado a repetirla”<sup>1</sup> el último, las ideas centrales y auxiliares producto de la experiencia y comprobación científica.

### 4.1 TEÓRICO

Se describen los sucesos que a través de la historia, han venido gestando la idea de consolidar un sistema: la representación virtual de un mundo real con dispositivos de inmersión y su articulación como herramientas para la tele presencia o tele operación de dispositivos electro mecánicos o servomecanismos.

Muchas de las teorías que soportan este tipo de desarrollos no nacen a partir de fuertes investigaciones sino de películas de ciencia ficción, que hoy en día son una realidad gracias al mágico trabajo de la ingeniería, la consolidación de tecnologías por algunas empresas y el trabajo investigativo de las universidades.

En la película Hinokio[9], un niño sufre un accidente de coche con su madre, su madre muere y él sobrevive, pero se ve aprisionado a una silla de ruedas durante un tiempo. Aparte de este problema físico, después de la muerte de su madre se retrae, culpa a su padre de lo sucedido y se encierra en su habitación, sin querer saber nada del mundo exterior. Su padre, preocupado porque no se relacione con nadie, le diseña un robot que el niño puede teledirigir desde un computador en su casa. A través de este robot va a la escuela, es decir, es el robot el que va a la escuela pero el niño lo maneja desde su casa.

Esto pareciera solamente una película pero día a día el ser humano se acerca más a esta realidad, es el caso real de LyndonBaty para quien el mundo es una ambiente hostil él solamente puede permanecer en un medio controlado como un hospital o su casa y por mediante un robot desarrollado por VGo® Communications, Inc (Una empresa de comunicaciones fundada en el año 2007), asiste a clases en Texas de manera remota a través de su computador.

Existen un sin número de películas de ciencia ficción que ofrecen teorías muy interesantes y relacionadas con la temática del proyecto, que han servido a lo largo del tiempo en motor de inspiración y desarrollo de tecnologías para muchos investigadores. Otro punto importante de esta base teórica es la característica de inmersión y tele-presencia que se logra a través de la imagenología. Es el caso concreto de Andrew Maimone y Henry Fuchs,

---

<sup>1</sup> Ruiz de Santayana, Jorge Agustín Nicolás.

ambos de la Universidad de Carolina del Norte en los EE. UU. , quienes desarrollaron un sistema para Tele-presencia 3D en tiempo real con cuatro Kinects a principios de 2011.[10]

El secreto del sistema descansa en una etapa de procesamiento basada en hardware Unidad de Procesamiento Gráfico GPU, encargada de las operaciones fundamentales de procesamiento de imágenes: generación de superficies, alisado, corrección de color, entre otros, a la impresionante tasa de 100 millones de triángulos por segundo que son procesados a través de un computador con una tarjeta gráfica.

Este tipo de teorías se pueden llevar a la práctica mediante métodos de investigación tradicionales, conjugando trabajos o tecnologías, mediante la creación, el análisis y la optimización de prototipos que es finalmente en lo que se quiere incursionar.

## 4.2 HISTÓRICO

Hace una reflexión mediante un análisis cronológico o recuento histórico de finales del siglo XIX al momento de escribir este libro, diferente al estado del arte, pues este último recoge trabajos de rigor científico actualizados con un intervalo de tiempo no mayor a cinco años (para más información ver[11]).

A finales del siglo XIX, el inventor británico Alexander StanhopeSt George ideó un proyecto llamado Telectroscope con el que pretendía crear un “dispositivo de supresión de la ausencia”. Según un artículo publicado en The New York Post el 29 de marzo de 1877, sería un dispositivo colosal que “permitiría que las personas y comerciantes de ciudades tan distantes entre sí como Calcuta, Liverpool, Paris, Pekín; se vieran cara a cara”. Para lograrlo, sería necesario cavar un túnel por debajo del mar y, a través de un poderoso artefacto óptico que aprovecharía la reflectividad del selenio, se transmitirían las imágenes por tubos que se desplegarían por el túnel submarino. A pesar de las dificultades y lo estrambótico del proyecto, en 1890 el inventor empezó a cavar el túnel, aunque, obviamente, no pudo llegar a terminarlo<sup>2</sup>.

En 1958 la Philco Corporation desarrolla un sistema basado en un dispositivo visual de casco controlado por los movimientos de la cabeza del usuario.

En el inicio de los 60, Iván Sutherland y otros crean el casco visor HMD mediante el cual un usuario podía examinar, moviendo la cabeza, un ambiente gráfico. Simultáneamente Morton Heilig inventa y opera el Sensorama.

Para 1969, Myron Krueger creó ambientes interactivos que permitían la participación del cuerpo completo, en eventos apoyados por computadoras.

---

<sup>2</sup>Tratamiento digital de imágenes. [En línea] [Citado el: 26 de 05 de 2010.] <http://www.grimaldos.es/cursos/imgdig/tipos.html>.

En 1969 la NASA puso en marcha un programa de investigación con el fin de desarrollar herramientas adecuadas para la formación, con el máximo realismo posible, de posteriores tripulaciones espaciales.

En el inicio de los 70, Frederick Brooks logra que los usuarios muevan objetos gráficos mediante un manipulador mecánico.

A fines de los 70, en el Media Lab del instituto tecnológico de Massachusetts MIT, se obtiene el mapa filmado de Aspen, una simulación de vídeo de un paseo a través de la ciudad de Aspen, Colorado. Un participante puede manejar por una calle, bajarse y hasta explorar edificios.

También en los 70, Marvin Minsky acuña el término "tele-presencia", para definir la participación física del usuario a distancia.

William Gibson, al inicio de los 80, publica la novela "Neuromancer" donde la trama se desarrolla en base a aventuras en un mundo generado por computadora al que denomina ciberespacio.

En 1982, Steven Lisberger dirige la película "TRON" y en 2010 las empresas Disney lanzan la segunda parte de la película "tronlegacy". (Ideas sobre la inmersión e interacción entre el hombre y un mundo virtual)

Tom Zimmerman inventa el guante de datos (*Data glove*)

Jaron Lanier acuña el término de Realidad Virtual, concretando la variedad de conceptos que se manejaban en esa época.

En 1984, Michael Mc Greevy y sus colegas de la NASA desarrollan lentes de datos con los que el usuario puede ahora mirar el interior de un mundo gráfico mostrado en computadora.

Después de 1980 aparece el HOLODECK en la serie de TV StartTrek; este es un ambiente generado por computadora, con figuras holográficas para entretenimiento de la tripulación.

Para el inicio de los 90 los sistemas de realidad virtual emergen de los ambientes de laboratorio en búsqueda de aplicaciones comerciales.

Una de las obras pioneras en este campo es Telegarden, de Ken Goldberg, quien en 1994 creaba un dispositivo consistente en un brazo robótico que, controlado on-line, llevaba a cabo las labores de cuidado de un jardín según las órdenes del usuario.

Para el año 1995 los simuladores de vuelo, desde los más perfectos, como los que utilizaban Thomson-Militaire o Dassault, hasta los videojuegos para microordenadores son en sí aplicaciones de la realidad virtual, cuyo fin es situar a la persona en situaciones comparables a la experiencia real.

Un grupo de investigadores de IBM desarrolla un prototipo informático para la creación de realidad virtual. Este sistema generaba modelos del mundo real basados en representaciones tridimensionales y estereoscópicas de objetos físicos con los que pueden interactuar varias personas simultáneamente.

### **4.3 CONCEPTUAL**

La Academia Real de la Lengua Española[1] define el concepto como la idea que concibe o forma el entendimiento, la manera en que se determina algo en la mente después de examinar las circunstancias o experiencias. A diferencia del conocimiento especulativo considerado con independencia de toda aplicación como lo es el marco teórico o la teoría que se evaluó anteriormente y sirve como referencia o punto de partida para llegar al concepto.

En este apartado se definen algunos conceptos que son necesarios para entender la dinámica del proyecto y sobre los cuales se fundamenta.

#### **4.3.1 Computación gráfica**

Aunque algunos lo consideran un lenguaje que posibilita la expresión, es usual que se confunda, con las técnicas usadas en el campo de la ingeniería para la generación de imágenes en un computador.

La ingeniería en este campo utiliza computadores para ilustrar, integrar o cambiar la información visual y espacial, imitando el mundo real en perspectiva, geometría y los efectos de color iluminación y texturas mediante el renderizado[12].

Estas imágenes digitales están compuesta por pixeles, que son una descripción numérica del conjunto de colores que la conforman o la luminosidad asociada que en conjunto producen la sensación de color, muchas de ellas pueden simular una imagen tridimensional en cuyo caso se hace alusión al termino voxel.

Existen herramientas para retransmitir las imágenes captadas o para generar imágenes a partir de elementos primarios desde puntos pasando por líneas o figuras básicas como polígonos hasta superficies complejas como rostros. En este proyecto se aplican las herramientas de diseño de la computación grafica para emular un espacio real y un servomecanismo (mundo virtual) y también para analizar imágenes con el fin de obtener datos de control (interface gestual).

El conjunto de colores y luminosidad son captados mediante arreglos de transistores sensibles a estas radiaciones (que es el trabajo de la cámara web), la interpretación de estos datos, el análisis y la transformación de los mismos para obtener información útil es el trabajo de la computación gráfica.

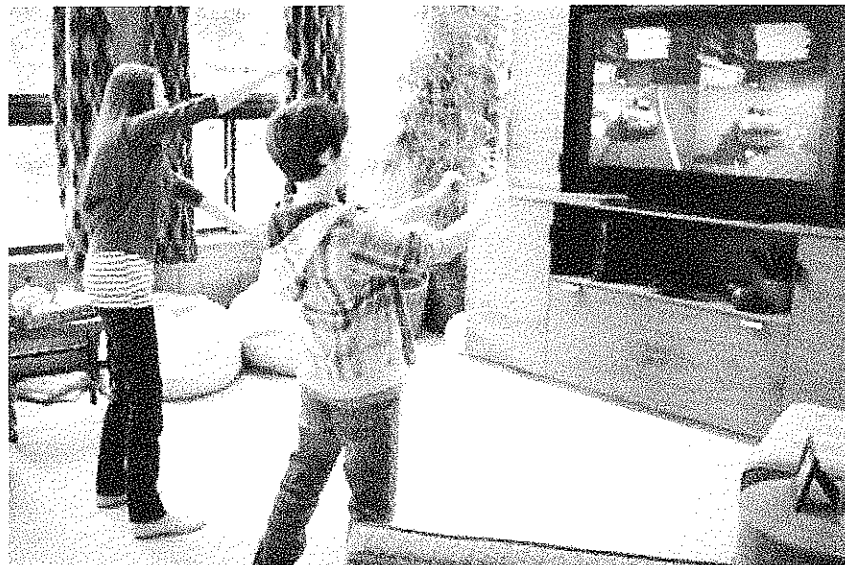
Con el fin de garantizar un ambiente de control lo más real, que posibilite la investigación del fenómeno de inmersión a través de la utilización de mundos virtuales y la combinación con técnicas de reconocimiento de imágenes para la navegación en el mismo, en el proyecto se utiliza la librería OpenCV (Open Computer Vision).

#### 4.3.1.1 Procesamiento de imágenes.

Son las estrategias matemáticas y lógicas que permiten la transformación de matrices de datos (píxeles) en información útil para un sin número de aplicaciones; diagnóstico médico identificando, zonas de alta temperatura, descalcificación, cáncer, cálculo de aéreas para realizar exámenes del terreno, analizar los recursos naturales, las fallas geológicas, ángulos, contornos y posiciones para identificar cuerpos entre otros.

Para el procesamiento de imágenes existen algoritmos ya probados por equipos de desarrollo de software, para el caso se utilizara OpenCV (**Open Computer Vision Library**), Esta librería es un proyecto en la web que busca consolidar un conjunto de las herramientas de tratamiento de imagen multiplataforma y para uso libre en la academia y el comercio[13].

#### 4.3.1.2 Interface gestual:



**Ilustración 1 Interface gestual kinect**

**Fuente: XBOX[14]**

En estos últimos años el uso de las interfaces gestuales (ver Ilustración 1) cada vez atrae más la atención de la sociedad y se convierte en una necesidad porque poco a poco van entrando en nuestra vida. La interactividad que se logra con una interface gestual se podría definir como la reacción rápida y coherente de un sistema en base a los movimientos naturales con o sin intención del usuario.

Cuando el ser humano usa simuladores de vehículos por lo general tiende a desplazar su cuerpo de forma natural de acuerdo al grado de inmersión que maneja el juego, por ejemplo en un juego de carreras, si el automóvil que se está simulando gira a la derecha es común que la persona que está jugando también lo haga porque se encuentra inmersa en esa realidad.

Para realizar una interface gestual genérica mediante la visión por computadora se requiere los siguientes elementos[15]:

- Cámara: el equivalente a la vista. Puede ser una simple cámara de luz visible o radiación electromagnética que no pueda ser percibida por el ojo humano (luz infrarroja).
- Algoritmos de reconocimiento: Se utilizan para analizar y reconocer los píxeles o voxels (*volumetric pixel*) de las imágenes que captura la cámara, con el fin de obtener una información específica por ejemplo la obtención de la posición de la cabeza (*head tracking*) para analizar el gesto o cualquier movimiento articular. Entre todas las herramientas de desarrollo
- Módulo de Información: Se envían los datos obtenidos del módulo anterior a una aplicación objetivo con un protocolo predefinido.
- Aplicación objetivo: En nuestro caso se generan cambios en una escena en tercera dimensión (3D), de acuerdo con la información proporcionada por el Módulo anterior. Cambiaría el punto de vista o realizaría cambios en las propiedades de los objetos virtuales.

Algunos ejemplos de interfaces gestuales o aplicaciones con imagenología son; el conteo de objetos en una autopista para control de tráfico, video inteligente o detección de rostros para el control de los trabajadores en una planta, sistemas de seguridad, Kinect para Xbox® entre otros. Con Kinect es posible aportar una dimensión totalmente nueva y extraordinaria a los juegos y al entretenimiento, es posible jugar con varias articulaciones del cuerpo sin tecnología adherida para la captura de información, en otras palabras los controles tradicionales desaparecen.[14].

#### **4.3.1.3 Algoritmos.**

Un algoritmo es un conjunto pre escrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad[16].

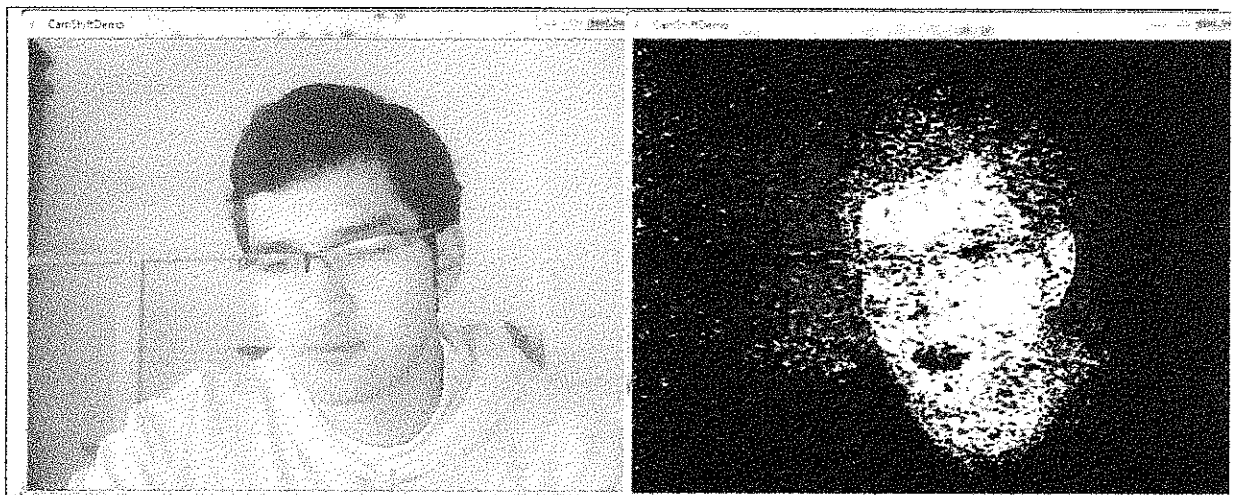
Mediante algunos algoritmos se encuentra el centro, el tamaño y la orientación de un objeto dentro de la imagen y el estudio de estas propiedades a través del tiempo se identifica el cambio de posición, es decir se puede rastrear el segmento de imagen, como lo explica OpenCV mediante el algoritmo camshift.

- **Adaptive Mean Shift Algorithm (CAMSHIFT)**

Los algoritmos de visión por computadora destinados a formar parte de una Interfaz gestual de usuario deben ser rápidos y eficientes, tal es el caso del seguimiento de la cara que debe procesarse prácticamente en tiempo real, por tal razón se ha escogido la técnica de seguimiento por color. Existen algoritmos más eficaces pero menos eficientes o complejos como los que usan correlación de colores, filtro de Kalman, predicción del contorno y otras técnicas o herramientas como las se muestran en la sección 4.3.1.5.

De acuerdo a la Figura 7. Diagrama de seguimiento de color, ver: Anexo E. Algoritmo Head Traking.

- ✦ El primer paso es escoger la ventana de búsqueda seleccionando el objeto que se quiere seguir.
- ✦ Se calcula el histograma con los matices que se encuentran en la selección, los valores se escalan en el rango  $[0, 255]$  para organizar los colores, posteriormente se obtiene la distribución de probabilidad objetivo.
- ✦ En cada captura se busca la retroproyección (back projection) del histograma del paso anterior. Esta operación consiste en generar una imagen en escala de grises, en donde cada pixel tendrá como intensidad el valor del histograma correspondiente al matiz de dicho pixel en la imagen procesada (Ilustración 2). El valor de cada pixel de esta imagen identifica la probabilidad de que dicho pixel en la imagen procesada pertenezca al objeto.



**Ilustración 2 Captura cámara y distribución de probabilidad de la piel**

**Fuente: El autor**

- ✦ Luego de calcular el centro “centroide” de la imagen de retroproyección “back projection” (Ilustración 2) mediante el algoritmo *Mean Shift* se establece el centro obtenido como nuevo centro de la ventana de búsqueda. Para el proceso de convergencia se repiten los pasos a y b.

- ⚡ Establecer una nueva ventana de búsqueda para las siguientes capturas, utilizando el centro obtenido en el paso 4 con los valores de tamaño y posición, además de otros patrones como el alabeo “roll” que permite obtener el valor de la posición de interés para el desarrollo del proyecto.
- ⚡ En el último paso se iterara el proceso como lo muestra el flujograma.

#### 4.3.1.4 Técnicas.

La técnica es un conjunto de saberes prácticos o procedimientos para obtener un resultado<sup>3</sup>. Para el procesamiento de las imágenes se mencionan algunas técnicas que permiten conseguir información adicional a partir de una imagen o una secuencia de las mismas[3]:

- **Filtrado.** Las operaciones de filtrado se llevan a cabo directamente sobre los píxeles de la imagen. En este proceso se relaciona, para todos y cada uno de los puntos de la imagen, un conjunto de píxeles próximos al píxel objetivo con la finalidad de obtener una información útil, dependiente del tipo de filtro aplicado, que permita actuar sobre el píxel concreto en que se está llevando a cabo el proceso de filtrado para, de este modo, obtener mejoras sobre la imagen y/o datos que podrían ser utilizados en futuras acciones o procesos de trabajo sobre ella.
- **El promediado.** Consiste en el análisis de las matrices de píxeles que conforman la imagen como función matemática, se promedian píxeles de distintas imágenes de una misma escena, la escena no debe variar significativamente durante un determinado número de muestras y cuanto mayor sea el número de muestras, mayor es la posibilidad de obtener una imagen que represente las demás.
- **La segmentación de bordes.** La idea básica detrás de detección de ejes es encontrar lugares en la imagen donde la intensidad cambia rápidamente, usando uno de estos dos criterios:
  - ⚡ Encontrar lugares donde la primera derivada de la intensidad es mayor que una magnitud específica.
  - ⚡ Encontrar lugares en la imagen donde la segunda derivada pasa por cero [17]
- **El análisis de regiones.** La idea básica para la detección de regiones es encontrar lugares o píxeles en la imagen donde la intensidad o color no cambia rápidamente o que se encuentra dentro de un rango específico. [18]

---

<sup>3</sup>Real Academia Española. [En línea] [Citado el: 26 de 05 de 2011.] <http://www.rae.es/rae.html>.

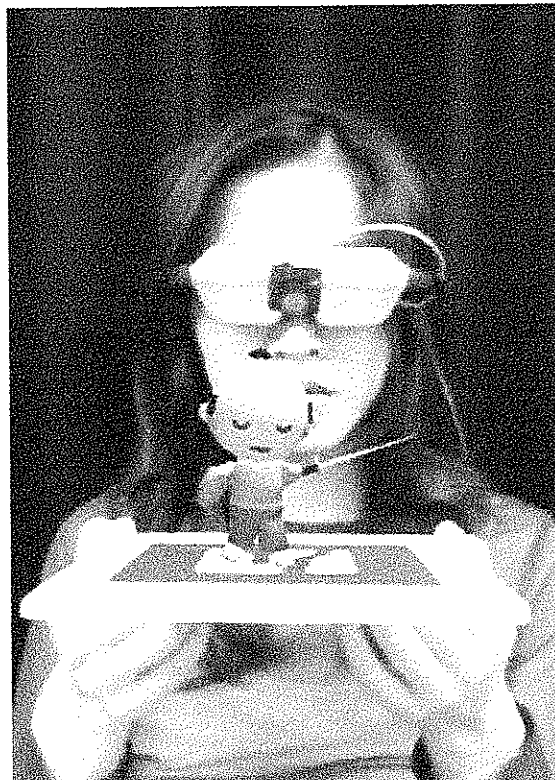
- **La detección de formas.** Es el ajuste de una imagen o parte de ella a figuras geométricas, usando las técnicas de segmentación de bordes para identificar puntos o líneas y puntos de corte, otra técnica menos usual es mediante la identificación de colores y áreas de colores pero es para formas definidas.

#### 4.3.1.5 Herramientas.

Existen algoritmos o bibliotecas para la utilización de las técnicas explicadas anteriormente con suficiente documentación en la web permiten a cualquier usuario explorarlas y adaptarlas a los requerimientos de los proyectos de procesamiento de imágenes como: Artoolkit y OpenCV.

- **ARToolKit**

ARToolKit fue desarrollada por el Dr. Hirokazu Kato con apoyo del Laboratorio Tecnológico de Interfaces Humanas (Human Interface Technology Lab, HIT Lab) y ARToolworks, Inc, Seattle [19]. Es una biblioteca de software de Licenciamiento Público General (GPL) para desarrollar aplicaciones de Realidad Aumentada (AR), es decir, para la superposición de imágenes virtuales en el mundo real[19].



**Ilustración 3 Personaje virtual 3D sobre una tarjeta de patrón real**  
Fuente:[19]

El fuerte de ARToolKit es la AR, la (Ilustración 3) es un claro ejemplo, en ella aparece un personaje virtual en tres dimensiones, este se encuentra de pie sobre una tarjeta real a la cual se le dibuja un patrón (Figura a blanco y negro de fácil identificación). Puede ser visto por el usuario con el dispositivo de visualización que está usando en los ojos Head mounted Display (HMD). Cuando el usuario mueve la tarjeta que contiene el patrón, el personaje virtual se mueve con él como si en realidad estuviese asociado al objeto real.

Una de las principales dificultades en el desarrollo de aplicaciones de AR es la de localizar el punto de vista los usuarios. Con el fin de saber cómo dibujar la imagen virtual, la aplicación tiene que saber el punto desde dónde el usuario observa el mundo real.

ARToolKit utiliza algoritmos en tiempo real, de visión por computador y cálculo de posición-orientación de la cámara, de acuerdo con cualquier patrón. Esto permite el fácil desarrollo de aplicaciones de Realidad Aumentada. Otras características importantes[19] son:

- Interfaz gráfica de usuario *graphical user interface* (GUI)
- Una biblioteca gráfica (sobre la base de GLUT)
- Un renderizado basado en OpenGL
- Soporte a Lenguaje Modelado de Realidad Virtual (VRML)

#### • OPENCV

Es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos[13].

La escogencia de esta herramienta se debe en gran medida a que tiene licencia (BSD)<sup>4</sup>, que permite a diferencia de ARToolKit que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

Open CV es multiplataforma (GNU/Linux, Mac OS X y Windows). Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica.

Además proporciona un entorno de desarrollo fácil de utilizar y eficiente, programación en código C y C++ optimizados, aprovecha las capacidades que proveen los procesadores multinúcleo. OpenCV puede utilizar el sistema de primitivas de rendimiento integradas de Intel y un conjunto de rutinas de bajo nivel específicas para procesadores Intel (IPP).

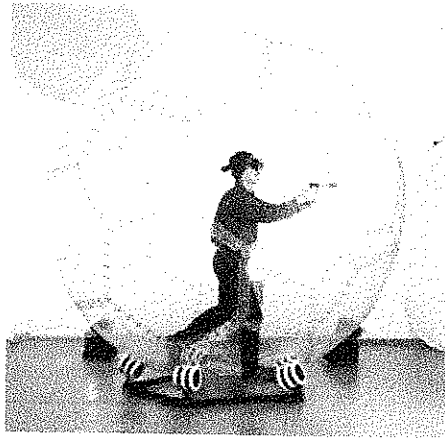
---

<sup>4</sup> Berkeley Software Distribution

En el contexto del proyecto se utiliza la librería OpenCV para capturar una zona dentro de la imagen obtenida mediante la cámara, con el fin de hacer seguimiento para que esta arroje los datos de salida necesarios y se generen las señales de control adecuadas al servomecanismo y al mundo virtual.

#### 4.3.2 Realidad virtual

Es un área que ha surgido como evolución de la computación gráfica y que aún se encuentra en desarrollo.



**Ilustración 4 Sistema inmersivo**

**Fuente:**<sup>5</sup>

La Realidad virtual es una simulación de un espacio y elementos tridimensionales interactivos usando dispositivos micro procesados en la que el usuario es introducido en un ambiente artificial, y que lo percibe como real mediante estímulos a los órganos sensoriales[17].

La realidad virtual se puede clasificar en inmersiva y no inmersiva, la primera involucra periféricos convencionales, imagen plana 2D y comando mediante teclado aunque la reacción de una persona frente a este tipo de sistemas algunas veces pareciera ser producto de un sistema altamente inmersivo como se ve en la siguiente figura.

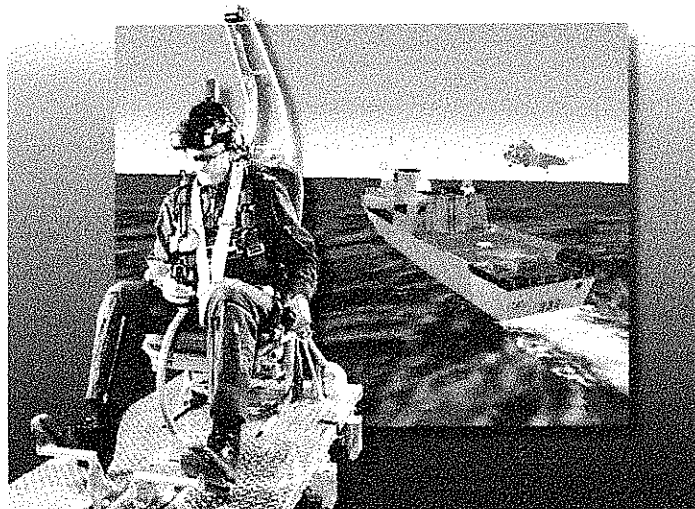
---

<sup>5</sup>Realidad virtual inmersiva. [En línea] [Citado el: 26 de 05 de 2011.] [www.rvi-immersivevirtualreality.blockspot.com](http://www.rvi-immersivevirtualreality.blockspot.com)



**Ilustración 5 Sistema no inmersivo**  
**Fuente:**<sup>6</sup>

La segunda utiliza dispositivos como guante de datos, pantallas para cada ojo o pantallas 3D, gafas vicio cascos, caminadoras, o dispositivos que emulen de forma directa sensaciones físicas como se muestra en la siguiente figura.



**Ilustración 6 Entorno de realidad inmersiva**  
**Fuente:**<sup>7</sup>

El proyecto es de carácter inmersivo debido a 2 aspectos, el primero es que el aplicativo consta de un mundo virtual y un avatar en 3D (ver sección 5.3.2.1) y el segundo que los movimientos que realiza el usuario, influyen de manera directa en el comportamiento del servomecanismo simulado (*avatar*).

<sup>6</sup> Synchronizing CRM vs. Direct Browser Access. [En línea] [Citado el: 26 de 05 de 2011.] <http://www.crmswitch.com/crm-access/crm-synch-vs-crm-web-access/>

<sup>7</sup> Que es la realidad virtual. [En línea] [Citado el: 26 de 05 de 2011.] <http://www.universidaddelsur.edu.mx>

#### 4.3.2.1 Aplicación objetivo o mundo virtual.

La aplicación objetivo es donde se materializa la información captada a través del recurso tecnológico, ya sea la interfaz gestual o con interacción física directa mediante algún dispositivo electrónico adherido al cuerpo

La plataforma para la creación del aplicativo objetivo es la siguiente:

- **Blender:**

Es un programa informático multiplataforma usado especialmente para el modelado, animación y creación de gráficos tridimensionales, compatible con todas las versiones de Windows, Mac OS X, Linux, Solaris, Free BSD e IRIX.

La Fundación de Blender tiene sus oficinas principales en Ámsterdam, Holanda. Ton Roosendaal (Desarrollador en jefe y creador del programa de animación 3D) junto con un pequeño personal, trabajan a tiempo completo sobre Blender. Los ingresos para pagar la administración del portal gratuito no salen de los usuarios si no de la tienda de internet *e-shop* y la industria editorial que organiza actividades como Siggraph y conferencias de Blender, apoya proyectos de desarrollo y documentación, mantenimiento y otros servicios. Blender es un esfuerzo de comunidad verdadero. El mantenimiento, nuevos proyectos e información para ser un experto en poco tiempo son definidos vía canales abiertos y accesibles en Blender.org.[20]

La funcionalidad de Blender puede extenderse vía Python, existen un número de scripts muy útiles que lo asisten durante todos los procesos, y es la manera como se consolidan las tecnologías abordadas en este trabajo de grado

- ✓ **Scripting (Python)**

Python [21] es un lenguaje de programación multiplataforma, libre y potente que se utiliza en una amplia variedad de aplicaciones se puede comparar con lenguajes como Tcl, Perl, Ruby, Scheme o Java. Algunas de sus características distintivas clave incluyen:

- ✚ Blender está creado en Python y por ende es posible extender sus funcionalidades mediante scripts, módulos o librerías hechos en este lenguaje.
- ✚ Sintaxis muy clara y legible
- ✚ Código abierto
- ✚ Aplicaciones de escritorio con interfaces visuales accediendo a componentes escritos en .Net (Microsoft), Qt, GTK, MFC, Swing (Java) etc
- ✚ Manejo de errores y excepciones
- ✚ Módulos fácilmente escritos en c, c++ (o java para Python, o .net para ironpython)
- ✚ Aplicaciones para celulares.

## ✓ Modelado:

Para el diseño de una escena 3D se necesita de modelos, materiales y luces. En el proceso de modelado se crea la geometría que imita la forma de un objeto del mundo real o se ajusta a objetos abstractos.

Los objetos vienen en muchas formas y tamaños, en Blender se dispone de muchas herramientas diferentes para que el modelado sea rápido y eficiente:

- ✚ Objetos (Trabajar con los objetos en su conjunto)
- ✚ Mallas (Trabajar con la malla que define la forma de un objeto)
- ✚ Curvas (Usar curvas para modelar y controlar objetos)
- ✚ Superficies Modelar una superficie NURBS)
- ✚ Texto (Herramientas de texto para colocar palabras en el espacio 3D).
- ✚ Meta Objetos (Globos y Glóbulos) Duplicaciones, (Duplicar objetos) y *Modeling Scripts*.

Dos técnicas de modelado comunes en Blender son "box modeling" y "curve modeling", la primera comienza con un cubo base y se procede con extruir caras y mover vértices para crear una malla más grande y compleja. La segunda se usa para objetos planos, como paredes y tablas de mesa, define el bosquejo utilizando curvas Bézier o Nurbs, y luego las extruye al espesor deseado. Cualquiera de los métodos está totalmente soportado en Blender utilizando sus herramientas de modelado.[22]

## ✓ Texturizado:

El proceso de texturizado en Imágenes Generadas por Computador (Computer Generated Imagery, CGI)<sup>8</sup>, es un método para añadir detalles a las superficies mediante la proyección de imágenes y patrones en las superficies. Las imágenes proyectadas y los patrones se pueden ajustar a afectar no sólo en el color, sino también la especularidad, la reflexión, la transparencia y la profundidad de 3 dimensiones, incluso falsa. Muy a menudo, las imágenes y los patrones se proyectan durante el tiempo de render, pero la asignación de texturas también se utiliza para esculpir, pintar objetos que se deforman en la animación.

En Blender, el texturizado se usa; aplicado a un material, aplicado al fondo Mundial, aplicado a un cepillo para esculpir un objeto, aplicado a modificadores entre otros.

## ✓ La iluminación:

Es un tema muy importante en la renderización, tan importante como el modelado, los materiales y las texturas. Una escena muy precisa, completamente modelada y texturizada

---

<sup>8</sup> Computer-generated [En línea] [Citado el: 26 de 10 de 2011.] imagery, [http://www.sciencedaily.com/articles/c/computer-generated\\_imagery.htm](http://www.sciencedaily.com/articles/c/computer-generated_imagery.htm)

tendrá pobres resultados sin un apropiado esquema de iluminación, mientras que un simple modelo puede convertirse en uno muy real si es iluminado con destreza. La iluminación, tristemente, es a menudo pasada por alto por artistas sin experiencia quienes comúnmente creen que como en el caso de las escenas del mundo real, son frecuentemente iluminadas por una única luz (una lámpara, el sol, etc.), una única luz hará lo mismo en un gráfico computarizado. Esto es falso, aunque en el mundo real una única fuente de luz está presente, la luz emitida por tal fuente rebota en los objetos y es irradiada nuevamente por encima de toda la escena, haciendo que las sombras se vuelvan suaves y que las regiones sombreadas no parezcan tan oscuras, sino parcialmente iluminadas.

### ✓ La Animación

Es lo que hace que un objeto se mueva, o cambie de forma con el tiempo. Los objetos pueden ser animados en muchos sentidos:

Moviéndose como un objeto completo, (cambiando su posición, la orientación o el tamaño en el tiempo), deformándolas (la animación de sus vértices o puntos de control), Animación de personajes a través de la armadura (animado a deformarse por el movimiento de los huesos dentro de la malla, una interacción muy compleja y flexible que hace en forma de carácter-objetos aparecen a caminar y saltar).

Tres métodos se utilizan normalmente en el software de animación para hacer un movimiento de objetos 3D:

### ✓ Los fotogramas clave

Posiciones completas se guardan para las unidades de tiempo (frames). Una animación se crea mediante la interpolación de un objeto de forma fluida a través de los marcos. La ventaja de este método es que permite trabajar con unidades claramente visualizados. El animador puede trabajar desde una posición a otra y puede cambiar de posición creados anteriormente, o moverlos en el tiempo.

### ✓ Curvas de animación

Las curvas se interpolan a partir de fotogramas clave, y se pueden extraer para cada componente XYZ la ubicación, rotación y tamaño, así como cualquier otro atributo en Blender. Estos forman los gráficos para el movimiento, con el tiempo establecido en horizontal y el valor que figura en posición vertical. La ventaja de este método es que proporciona un control preciso sobre los resultados de la animación.

### ✓ Camino

Una curva se dibuja en el espacio 3D, y el objeto está limitado a seguirlo de acuerdo con una función de tiempo dado de la posición a lo largo del camino.

En versiones anteriores a 2.5x sólo los bloques de determinados datos tenían la capacidad de ser *keyframed*. Ahora es posible animar casi cualquier tipo de datos.

### ✓ Motor de juegos:

Blender tiene su propio motor de juego en el que le permite crear aplicaciones interactivas en 3D. El motor de juego de Blender (BGE) es una potente herramienta de alto nivel de programación. Su principal objetivo es el desarrollo del juego, pero se puede utilizar para crear cualquier software interactivo en 3D con otros fines, tales como tours interactivos en 3D de arquitectura o de investigación de la física educativa.

Hay tres tipos de bloques lógicos, Sensores, Controladores, Actuadores. Los sensores perciben cuando las cosas suceden como una colisión, la pulsación de una tecla o el movimiento del ratón. Los sensores están conectados a los controladores, los cuales los comparan y activan los actuadores.

Cuenta además con el editor de la lógica es lo que hace que cualquier cosa suceda en el juego. Está diseñado para proveer una poderosa herramienta para crear la lógica del juego a través de un entorno gráfico. Los bloques (*bricks*) representan funciones pre-programadas las cuales pueden ser ajustadas y combinadas para crear juegos y/o aplicaciones.<sup>9</sup>

### 4.3.3 Comunicaciones inalámbricas

Los medios de transporte de información por ondas electromagnéticas posibilitan la omnipresencia o ubicuidad, una característica ambiciosa pero deseada por el ser humano. Dotar a los sistemas robóticos con esta cualidad es posible mediante muchas tecnologías, una de ellas son los módulos de radio frecuencia (RF) a continuación analizaremos 3 tecnologías.

#### 4.3.3.1 Módulos RF. Zigbee<sup>10</sup>.

Dentro de varios protocolos de comunicaciones inalámbricas, el más común es el basado en el estándar de comunicaciones para redes inalámbricas IEEE\_802.15.4. Creado por Zigbee Alliance, una organización teóricamente sin ánimo de lucro, de más de 200 grandes empresas entre ellas se destacan Mitsubishi, Honeywell, Philips, Motorola, Invensys), muchas de ellas fabricantes de semiconductores.

Zigbee permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas. Es especialmente útil para redes de sensores en entornos industriales, médicos, domóticos, energías alternativas, almacenes de cadena e incluso en sistemas alámbricos.

---

<sup>9</sup> Blender Foundation [En línea] [Citado el: 26 de 10 de 2011.] [http://wiki.Blender.org/index.php/Main\\_Page](http://wiki.Blender.org/index.php/Main_Page)

<sup>10</sup> ZigBee Alliance [En línea] [Citado el: 26 de 05 de 2011.] <http://www.zigbee.org/>

Por lo general las comunicaciones Zigbee se realizan en la banda libre de 2.4GHz. A diferencia de bluetooth no utiliza espectro ensanchado por salto de frecuencia, *Frequency Hopping Spread Spectrum (FHSS)*, sino que realiza las comunicaciones a través de una única frecuencia, es decir, de un canal.

- ⚡ Es posible seleccionar un canal entre 16 posibles.
- ⚡ El alcance depende de la potencia de emisión del dispositivo, así como el tipo de antenas utilizadas (cerámicas, dipolos). Por ejemplo con antena dipolo y línea de vista, suele ser aproximadamente de 100m y en interiores de unos 30m.
- ⚡ La velocidad de transmisión de datos de una red Zigbee es de hasta 256kbps.
- ⚡ Una red Zigbee la pueden formar, teóricamente, hasta 65535 equipos.

- **Arquitectura básica de una red Zigbee.**

Una red Zigbee la forman básicamente 3 tipos de elementos. Un único dispositivo Coordinador, dispositivos Routers y dispositivos finales (end points).

- ✓ **El Coordinador.**

Es el nodo de la red que tiene la única función de formar una red. Es el responsable de establecer el canal de comunicaciones (como se menciona antes) y del PAN ID (identificador de red) para toda la red. Una vez establecidos estos parámetros, el Coordinador puede formar una red, permitiendo unirse a él a dispositivos Routers y End Points. Una vez formada la red, el Coordinador hace las funciones de Router, esto es, participar en el enrutado de paquetes y ser origen y/o destinatario de información.

- ✓ **Los Routers.**

Es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta para enrutar un paquete de información. Lógicamente un router debe unirse a una red Zigbee antes de poder actuar como Router retransmitiendo paquetes de otros routers o de Endpoints.

- ✓ **End Device**

Los dispositivos finales no tienen capacidad de enrutar paquetes. Deben interactuar siempre a través de su nodo padre, ya sea este un Coordinador o un Router, es decir, no puede enviar información directamente a otro enddevice. Normalmente estos equipos van alimentados a baterías. El consumo es menor al no tener que realizar funciones de enrutamiento.

- **IEEE 802.15.4.**

Es un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos (256kbps). También

es la base sobre la que se define la especificación de ZigBee, cuyo propósito es ofrecer una solución completa para este tipo de redes construyendo los niveles superiores de la pila de protocolos que el estándar no cubre.

#### 4.3.3.2 Z-Wave:

Es un protocolo inalámbrico por ondas de radio destinado para la comunicación entre dispositivos domésticos eléctricos. Sirve para crear una red única con los equipos eléctricos como: Luz, termostatos, alarmas, ordenadores, teléfonos, aire acondicionado y el manejo se ventanas eléctricas y persianas. A la red se le pueden conectar todos los dispositivos electrónicos para el uso doméstico. En el mercado ya están disponibles muchos productos con esta tecnología.

El sistema Z-wave trabaja con la frecuencia 868,42 Mhz y se caracteriza por tener un bajo consumo de energía (tiempo de alerta de 2,5uA). El alcance de los espacios cerrados es de 45 metros y en los espacios libres 150 metros.<sup>11</sup>

#### 4.3.3.3 Módulos de comunicación inalámbrica PT2262 y PT2272

El circuito integrado PT2262 es un control remoto del codificador se combina con la utilización de la tecnología CMOS PT2272. Que codifica los datos de una pines de dirección en una serie wave de código adecuado para RF o la modulación de rayos infrarrojos. PT2262 tiene un máximo de 12 bits de pines de dirección de los tres estados que proporciona hasta 531.441 (o  $3^6$ ) los códigos de dirección, por lo tanto, reduciendo drásticamente cualquier colisión de código y el código no autorizado de exploración posibilidades.

- **Características:** Es tecnología CMOS con un bajo consumo de potencia, gran inmunidad al ruido, tiene hasta 12 direcciones de pines tri-estado y hasta 6 para datos, un rango de operación de 4V a 15V, sus salidas son tipo latch y está disponible en empaquetado DIP y SO.
- **Aplicaciones:** Las aplicaciones más comunes para este circuito integrado son; sistemas de seguridad para carros, controles de puertas para garaje, ventiladores de control remoto, juguetes de control remoto, automatización y seguridad en el hogar y control remoto industrial.

#### 4.3.4 Robótica

Diseño y construcción de máquinas capaces de realizar tareas repetitivas o dañinas para el hombre.

---

<sup>11</sup> Conexiones inalámbricas. [En línea] [Citado el: 10 de 10 de 2011.] <http://www.fakro.es/att/foldery/Z-WAVE.pdf>

#### 4.3.4.1 Robot

- **Autómatas:** Un autómatas es una estructura electro mecánica dotada de un algoritmo que le permite realizar tareas programadas. También puede considerarse autómatas una definición algorítmica que realice operación de datos.
- **Mecánica Servos:** Un servomotor (también llamado servo) es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición. Los servos se utilizan frecuentemente en sistemas de radio control y en robótica, pero su uso no está limitado a estos. Es posible modificar un servomotor para obtener un motor de corriente continua que, si bien ya no tiene la capacidad de control del servo, conserva la fuerza, velocidad y baja inercia que caracteriza a estos dispositivos.

#### 4.3.4.2 Sistema Electrónico

El robot dispone de 3 elementos en cuanto al aspecto electrónico se refiere, un módulo de recepción de señal inalámbrica (sección 4.3.3 ), un circuito micro controlado encargado de recibir los datos del módulo anterior y traducirlo en órdenes para el circuito de potencia.

- **Microcontroladores**

Son dispositivos semiconductores programables de alta integración, sus prestaciones son similares pero limitadas si se comparan con las de cualquier computador personal, la característica principal es su alto nivel de especialización.

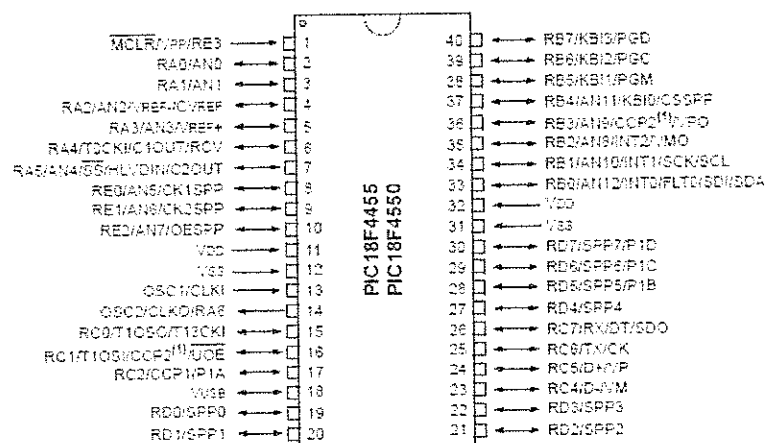


Ilustración 7 Diagrama de pines PIC18F4550

Fuente: Microchip Technology Inc®.

✓ **El pic 18F4550:**

El PIC 18F4550, pertenece a los microcontroladores PIC18 de gama alta. Posee una arquitectura *reduced instruction set computer* (RISC) de 16 bits longitud de instrucciones y 8 bits de datos. La tabla muestra en resumen las características fundamentales de este microcontrolador.

CARACTERÍSTICAS GENERALES	PIC18F4550
Frecuencia de reloj	Hasta 48MHz
Memoria ROM (bytes)	32.768
Memoria RAM de Datos (bytes)	2.048
Memoria EEPROM Datos (bytes)	256
Interrupciones	20
Líneas de E/S	35
Temporizadores	4
Módulos de Comparación/Captura/PWM (CCP)	1
Módulos de Comparación/Captura/PWM mejorado (ECCP)	1
Canales de Conversión A/D de 10 bits	13 Canales
Canales de Comunicación Serie	MSSP.EUSART
Canal USB	1
Puerto Paralelo de Transmisión de Datos (SPP)	1
Comparadores analógicos	2
Juego de instrucciones	75 (83 ext.)
Encapsulados	QFN 40 pines PDIP40 pines TQFP 40 pines

**Tabla 1 Características del microcontrolador 18F4550**

Fuente: El autor

✓ **Herramientas:**

Existen herramientas de programación de alto nivel para programar en corto tiempo lo que lo hace muy útil a la hora de analizar, simular, diseñar programar prototipos.

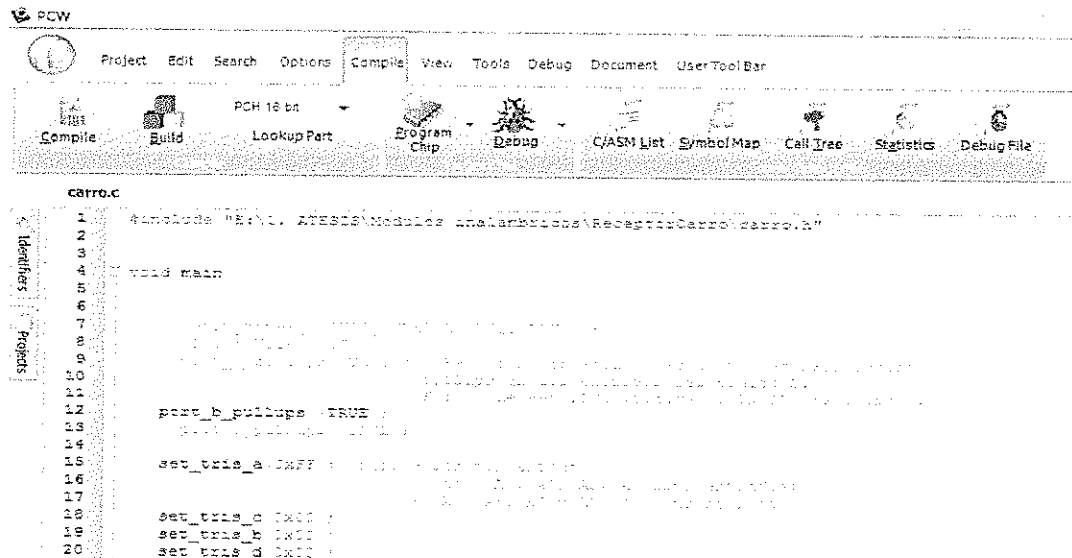


**Ilustración 8 Herramientas para análisis y diseño electrónico**

Fuente: El autor

- **Proteus ISIS professional:** es un programa para el diseño de esquemas electrónicos, orientada para fines educativos y domésticos. Contiene aplicaciones como aplicaciones: Isis, Ares y ProSpice

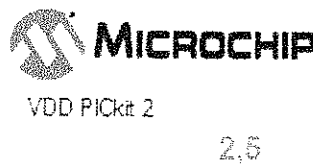
- **PCW Compiler for microchip PIC®<sup>12</sup>**: *Custom Computer Services CCS* es una herramienta de software integrada para construcción de código. Ofrece una suite para desarrollar y depurar aplicaciones embebidas se ejecutan en Microchip PIC ® MCU y dsPIC® DSC. Además de eso es tiene una interface intuitiva que facilita la escritura de código Ilustración 9.



**Ilustración 9 Inteface de programacion PCW.**  
Fuente: El autor

- **PROGRAMADOR PIC USB<sup>13</sup>**

PICKIT 2 PLUS es una herramienta para la programación de microcontroladores PIC de MICROCHIP, el cual permite programar microcontroladores de las familias 10fx ,12fx, 16fx, 18fx , 24fx, dsp30fx y 33fx, además soporta dispositivos vía ICSP.



**Ilustración 10 Logo Microchip**

<sup>12</sup> Custom Computer Services, Inc [Citado el: 26 de 05 de 2011.]<http://www.ccsinfo.com/>

<sup>13</sup> Manual de usuario.[Citado el: 26 de 09 de 2011.]  
[http://www.umicroelectronic.260mb.com/document/MANUAL\\_USUARIO.pdf](http://www.umicroelectronic.260mb.com/document/MANUAL_USUARIO.pdf)

PICKIT 2 PLUS es una adaptación del programador Pickit2, el cual se le agregó un módulo de programación para microcontroladores de zócalo de encapsulado Dip universal, muy fácil de usar a la hora de programar el dispositivo.

**PICKit™ 2**

**Ilustración 11 Logo PICKit2**

❖ *Características generales:*

- ⚡ Debugger Express compatible con MPLAB.
- ⚡ Analizador lógico de 2 canales.
- ⚡ Puerto serial virtual UART TX - RX.
- ⚡ Interfaz USB 2.0 compatible con 1.0 y 1.1.
- ⚡ No requiere alimentación externa.
- ⚡ Zócalo para programación tipo encapsulado DIP de dispositivos.
- ⚡ Conector para programación en circuito vía ICSP.
- ⚡ Interfaz UART – analizador lógico con puntas de prueba y conector independiente.

• **Puente H**

Es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, avance y retroceso. Son ampliamente usados en robótica y como convertidores de potencia. Los puentes H están disponibles como circuitos integrados, pero también pueden construirse a partir de componentes discretos.

✓ **El integrado L293D:**

Incluye cuatro circuitos para manejar cargas de potencia media, en especial pequeños motores y cargas inductivas, con la capacidad de controlar corriente hasta 600 mA en cada circuito y una tensión entre 4,5 V a 36 V.

Los circuitos individuales se pueden usar de manera independiente para controlar cargas de todo tipo y, en el caso de ser motores, manejar un único sentido de giro. Pero además, cualquiera de estos cuatro circuitos sirve para configurar la mitad de un puente H.

El integrado permite formar, entonces, dos puentes H completos, con los que se puede realizar el manejo de dos motores. En este caso el manejo será bidireccional, con frenado rápido y con posibilidad de implementar fácilmente el control de velocidad.

Para más información ver Anexo J. Datasheet L293D

### 4.3.5 Interfaz de comunicación

La mayoría de los Microcontroladores contienen módulos para la implementación de interfaces de comunicación tales como, SPI, SCI, I2C y USB. Además, algunas familias proveen interfaces Ethernet y CAN para el diseño de sistemas avanzados. Todos estos dispositivos vienen provistos de Librerías específicas de Diseño que facilitan la implementación del módulo.

Este proyecto usa un puerto USB simulado, a través de la Arduino:

#### 4.3.5.1 Descripción del sistema USB.

El USB es un bus punto a punto: dado que el lugar de partida es el host (PC o hub), el destino es un periférico u otro hub. No hay más que un único host (PC) en una arquitectura USB.

Los PC estándar tienen más de una interface USB, lo que implica que, para permitir más de dos periféricos simultáneamente, es necesario un hub. Algunos periféricos incluyen un hub integrado, por ejemplo, el teclado USB, al que se le puede conectar un Mouse USB.

Los periféricos comparten la banda de paso del USB. El protocolo se basa en el llamado paso de testigo (token). El ordenador proporciona el testigo al periférico seleccionado y seguidamente, éste le devuelve el testigo en su respuesta.

Este bus permite la conexión y la des-conexión en cualquier momento sin necesidad de apagar el equipo, los primeros aspectos de este protocolo son:

- **Interfaz física.**

A nivel eléctrico, el cable USB transfiere la señal y la alimentación sobre 4 hilos, a nivel de alimentación, el cable proporciona la tensión nominal de 5 V. Es necesario definir correctamente el diámetro del hilo con el fin de que no se produzca una caída de tensión demasiado importante en el cable. Una resistencia de terminación instalada en la línea de datos permite detectar el puerto y conocer su configuración (1,5 o 12 Mbits/s).

A nivel de señal, se trata de un par trenzado con una impedancia característica de  $90 \Omega$ , la velocidad puede ser tanto de 12 Mbits/s como de 1,5 Mbits/s. La sensibilidad del receptor puede ser de, al menos, 200mV y debe poder admitir un buen factor de rechazo de tensión en modo común. El reloj se transmite en el flow de datos, la codificación es de tipo NRZI, existiendo un dispositivo que genera un bit de relleno (bit stuffing) que garantiza que la frecuencia de reloj permanezca constante. Cada paquete va precedido por un campo de sincronismo.

En cuanto al consumo, cada sección puede proporcionar una determinada potencia máxima siendo el PC el encargado de suministrar la energía. Además, el periférico puede estar autoalimentado (selfpowered).

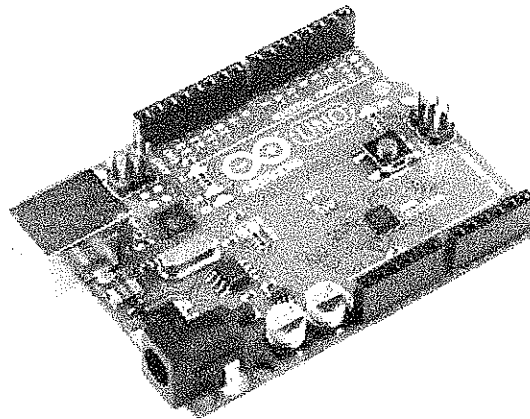
Control de consumo. El ordenador gestiona el consumo, teniendo capacidad de poner en reposo (suspend) o en marcha a un periférico USB. En reposo, este reduce su consumo (si puede), quedándose la parte USB funcional. [23]

#### 4.3.5.2 ¿Qué es Arduino?<sup>14</sup>

Es un dispositivo electrónico modular (Ilustración 12), abierto para la creación de prototipos basado en software y hardware flexibles y fáciles de usar. La tarjeta de Arduino puede mediante combinación de sensores y actuadores tomar y enviar información al entorno a través de sus pines.

El microcontrolador en la placa Arduino se programa mediante el lenguaje de Arduino (basado en Wiring<sup>15</sup>) y el entorno de desarrollo Arduino (basado en Processing<sup>16</sup>). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un computador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software (Python)

Las placas pueden ser hechas a mano o compradas montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta, libre de adaptación las necesidades.



**Ilustración 12 Tarjeta arduino 1**  
Fuente: El autor

Esta tecnología ofrece ventajas frente a los microcontroladores convencionales en cuanto a tres aspectos importantes.

- ⚡ Lenguaje de programación de alto nivel.
- ⚡ Documentación, ejemplos y soporte.
- ⚡ Arquitectura modular y prefabricada.

<sup>14</sup> Arduino.[Citado el: 26 de 05 de 2011.] <http://www.arduino.cc/es/>

<sup>15</sup> Wiring.[Citado el: 26 de 05 de 2011.] <http://wiring.org.co/>

<sup>16</sup> Processing.[Citado el: 26 de 05 de 2011.] <http://processing.org/>

Una arquitectura que permite ahorrar tiempo en diseño y fabricación de circuitos impresos y construcción de código (HW y SW), además de tener la documentación pertinente para conocer todos los detalles.

Esta tecnología se utilizó ya que se tenía la oportunidad de aprender algo nuevo y por el tiempo que se ahorra en proyectos de ingeniería con el uso de la misma.

## 5. ESTADO DEL ARTE

Existen un sin número de proyectos que cuentan con tecnología relacionada como lo son laboratorios virtuales de aprendizaje, laboratorios virtuales de robótica, tele operación por internet, robótica asistida por tele operación y realidad virtual, pero son muy escasos los que reúnen los desarrollos y tecnologías de tratamiento de imágenes, mundos virtuales y tele operación robótica.

Un ejemplo es la empresa sudafricana Robótica quien desarrollo un robot, teleoperado que *puede conectarse con un mundo virtual en línea por medio de una computadora para descargar nuevos programas, juegos o socializar con otros robots y sus propietarios.*[24], en este caso no utiliza cámaras de video para su manipulación.

El proyecto natal Kinect para Xbox 360, o simplemente Kinect es una tecnología que utiliza el tratamiento de imágenes para la interacción con mundos virtuales, tiene una gran cantidad de juegos, aún sigue incrementando la calidad y la cantidad implementando no solo reconocimiento gestual sino también comandos por voz.

En la Facultad de Ciencias de la Universidad de Salamanca, la Ingeniería Técnica en Informática de Sistemas, el señor Alberto Rodríguez Valle elaboró un proyecto titulado: "Teleoperación de un Brazo Robótico Mediante Realidad Aumentada", en este trabajo se estudia una biblioteca de realidad aumentada y se emplea en el desarrollo e implementación de una aplicación que conduce el movimiento de un brazo robótico.

Intel y Softkinetic-Optrima trabajan conjuntamente en una tecnología de reconocimiento gestual para el televisor, de modo que el espectador pueda ejecutar ciertos comandos sólo con mover las manos. El aparato permitiría que los televidentes navegaran en los menús, cambiaran de canal o regularan el volumen al mover sus brazos en patrones predeterminados. Softkinetic-Optrima ya trabaja con un casino para desarrollar tragamonedas que se puedan operar con gestos, amén de las posibles aplicaciones médicas y (¡por supuesto!) militares.[25]

Otras empresas como Cisco trabaja a diario en el sistema de video 3D en tiempo real y sonido envolvente en general en sistemas de telepresencia, es el caso del reciente modelo Cisco TelePresence 1100, pensado para ser utilizado por dos personas en una sala de conferencia multiuso, la calidad de la comunicación llega al extremo de la alta definición[26].

La Facultad de Ingeniería Universidad Militar Nueva Granada, ha realizado investigaciones relacionadas es el caso de la titulada "sistemas robóticos tele operados por" Alexander Cerón Correa, en la que realiza una revisión general sobre los sistemas robóticos teleoperados, su estructura, componentes principales, clasificación, tipos de sensores, sistemas de locomoción y aplicaciones y mencionan algunos trabajos realizados Colombia y en el mundo.

## 6. DISEÑO METODOLÓGICO

Para poder diseñar una metodología que se ajuste a las necesidades de un proyecto interdisciplinar es importante en primera medida definir el tipo de investigación que guía todo el proceso (investigación aplicada), luego cada una de las metodologías que mejor se ajusten a las necesidades de desarrollo particular.

El proyecto se realizó teniendo en cuenta el tipo de investigación aplicada porque esta garantiza un producto práctico, específico y original. Sin embargo sus procedimientos están pensados para llegar a la solución de manera general.

Cuando nos enfrentamos a un problema que involucra varias disciplinas y temas, la mejor forma de solucionarlo es aplicar el adagio popular "Divide y vencerás"<sup>17</sup>.

Además es necesario apoyar esos subtrabajos con metodologías que se ajusten a las necesidades particulares, bajo un mismo modelo natural (incremental e iterativo). Siempre respetando la metodología y pensando de manera modular cuando menos en el análisis de requerimientos.

### 6.1 LA INVESTIGACIÓN APLICADA

La estructura del documento evidencia el desarrollo del tipo de investigación aplicada y consiste en llevar a cabo trabajos originales realizados para adquirir nuevos conocimientos; sin embargo, está dirigida fundamentalmente hacia un objetivo práctico específico<sup>18</sup>. En este caso la originalidad radica en la integración de las tecnologías; realidad virtual, reconocimiento de imágenes, tele-operación. Cuando menos a nivel de la universidad, objetivo práctico específico es definido en la (sección 3 Objetivos) y la forma de consecución se explica en el desarrollo del modelo metodológico (sección 7 Resultados). El trabajo investigativo se evidencia en todo su proceso, la exploración, selección y aprendizaje de herramientas involucradas con cada una de las fases.

En cuanto al objetivo práctico específico se basa en la resolución de la siguiente pregunta de investigación:

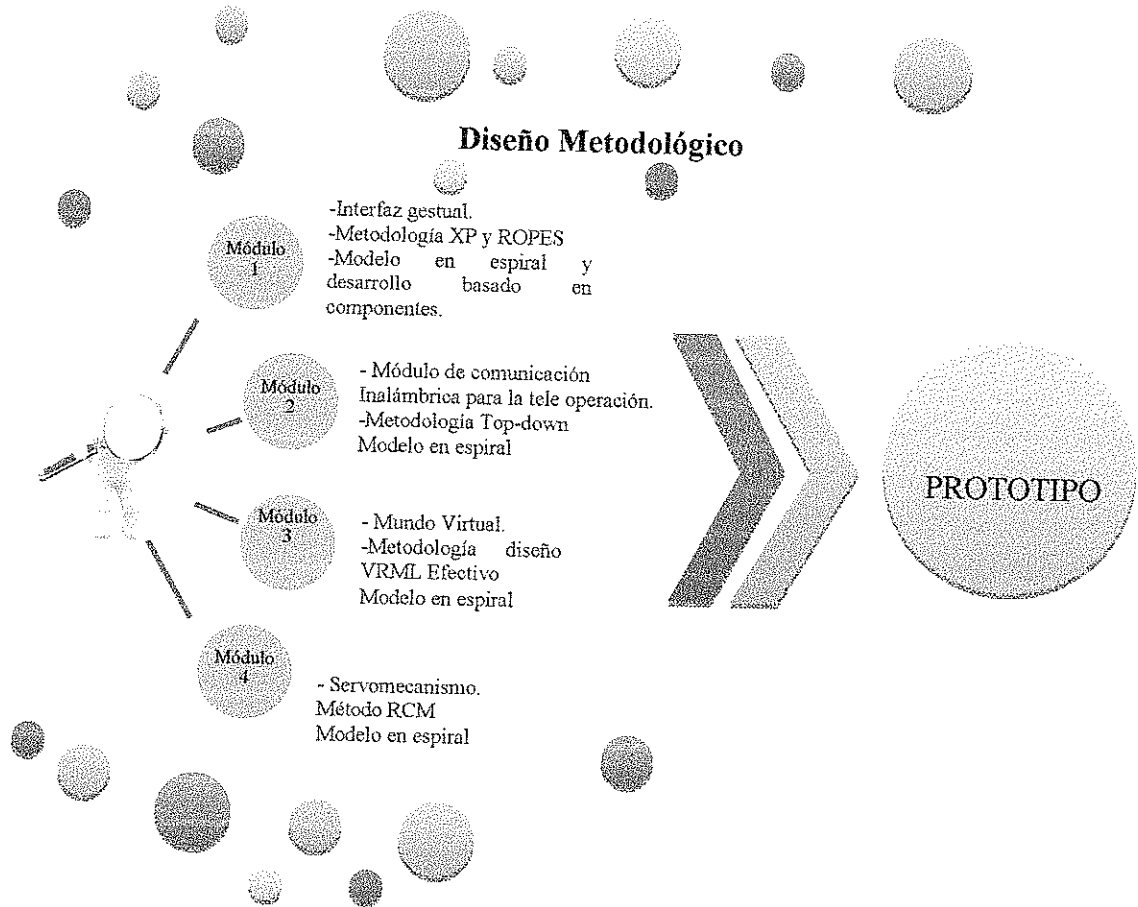
### 6.2 PREGUNTA DE INVESTIGACIÓN:

¿Es posible crear un prototipo de accionamiento real a distancia de un servomecanismo mediante un mundo virtual controlado por una interfaz gestual?

---

<sup>17</sup> Julio Cesar. Frase corta pero memorable

<sup>18</sup> OCDE. Manual de Frascati. Propuesta de Norma Práctica para encuestas de Investigación y Desarrollo experimental. 2002.



**Ilustración 13 Diseño Metodológico**  
Fuente: El autor.

## 6.6 MODELO EN ESPIRAL[9]:

El desarrollo en espiral es un modelo de ciclo de vida desarrollado por Barry Boehm en 1985, utilizado generalmente en la Ingeniería de software.

Las actividades de este modelo son una espiral, cada bucle es una actividad. Para cada actividad habrá cuatro tareas:

- **Planificación:**

Determinación de objetivos, alternativas y restricciones.

Se revisa todo lo hecho, se evalúa, y con ello se decide si se continúa con las fases siguientes y se planifica la próxima actividad.

- **Análisis de riesgo:**

Análisis de alternativas e identificación/resolución de riesgos.

- **Ingeniería:**

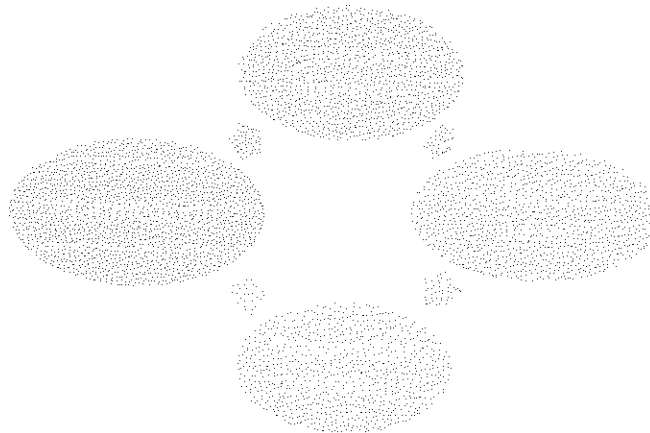
Desarrollo del producto del “siguiente nivel”

Tareas de la actividad propia y se prueba.

Análisis de alternativas e identificación resolución de riesgos.

- **Evaluación del cliente:**

Valorización de los resultados de la ingeniería.



**Ilustración 14 Metodología en espiral**

**Fuente: Autor basado en [9]**

- **Ventajas:**

El análisis del riesgo se hace de forma explícita y clara. Une los mejores elementos de los restantes modelos, reduce riesgos del proyecto, incorpora objetivos de calidad, integra el desarrollo con el mantenimiento, etc. Además es posible tener en cuenta mejoras y nuevos requerimientos sin romper con la metodología, ya que este ciclo de vida no es rígido ni estático.

- **Desventajas:**

Genera mucho tiempo en el desarrollo del sistema, modelo costoso, requiere experiencia en la identificación de riesgos.

El paradigma del modelo en espiral para la ingeniería de software es actualmente el enfoque más realista para el desarrollo de software y de sistemas a gran escala. Utiliza un enfoque evolutivo para la ingeniería de software, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel evolutivo. Utiliza la creación de prototipos como un mecanismo de reducción de riesgo, pero, lo que es más importante permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución de prototipos<sup>19</sup>.

<sup>19</sup> Modelo Espiral. [Citado el: 26 de 05 de 2011.]  
<http://www.mitecnologico.com/Main/ModeloDeEspiral>

## 6.7 MÓDULO 1 INTERFACE GESTUAL:

La metodología XP es una buena alternativa para de software sin embargo no se cuenta con un equipo de trabajo adecuado, esta metodología sugiere el trabajo en parejas se centra en la descripción de las relaciones que deben existir entre los actores del proyecto.

La metodología ROPES puede orientar correctamente la plataforma sobre que se desarrolla el este proyecto (Blender) pero esto no quiere decir que los artefactos correspondientes a la metodología sean de utilidad para el desarrollo de este Módulo. Sin embargo las siguientes características son consideradas dentro del diseño del algoritmo.

- ⊕ Concurrencia: comunicación de mensajes, manejo de eventos, validación temporal.
- ⊕ Identifica la documentación (diagramas, modelos) generada por la metodología, para describir la gestión de varios procesos dentro del sistema.
- ⊕ Comunicación de Mensajes: identifica la documentación generada para describir los mecanismos para dar soporte al intercambio de mensajes entre procesos.
- ⊕ Manejo de eventos: se identifican los documentos que describe el conjunto de pasos seguidos por el sistema con la llegada de ciertos eventos.
- ⊕ Validación temporal del sistema: se identifican los documentos que describe la planificación de los eventos en el tiempo.

En este módulo se hace un estudio y diseño de la arquitectura del sistema para poder articular la interface gestual al proyecto, por medio de ROPES se determinan las características mencionadas anteriormente y se tiene en cuenta la filosofía XP para el análisis de riesgos.

### 6.7.1 ROPES (Rapid Object-Oriented Process for Embedded Systems)

Es una de las metodologías de desarrollo para sistemas de tiempo real[27] emplea como notación UML se basa en un proceso de desarrollo iterativo (o en espiral). Está compuesto de diversas tendencias de la ingeniería del software, tales como, análisis de riesgo y calidad de software. Se organiza en cuatro grandes fases: análisis, diseño, traducción y pruebas, siendo los artefactos resultantes de cada fase, modelos o diagramas UML que se refinan o corrigen en las siguientes

En la actividad de análisis se llevan a cabo tres tipos de análisis.

- **Un análisis de requisitos:** en el que se especifican los requisitos funcionales y no funcionales del sistema.
- **Análisis de sistema:** se realiza la división de responsabilidades entre el hardware y el software, la arquitectura de alto nivel del sistema y los algoritmos de control necesarios.
- **Análisis de objetos:**
  - ⊕ Determinación los modelos estructurales de los objetos que han de componer el sistema
  - ⊕ Modelo del comportamiento mediante colaboraciones o diagramas de secuencias.

- **Diseño:** se llevan a cabo los diseños de la arquitectura, mecanismos y el detallado.
- **Traducción:** comprende la generación de código ejecutable a partir del diseño del sistema.
- **Pruebas:** se verifica la conformidad de la aplicación, sea para encontrar defectos o para observar un cierto nivel funcional. Incluye pruebas de integración y de validación.

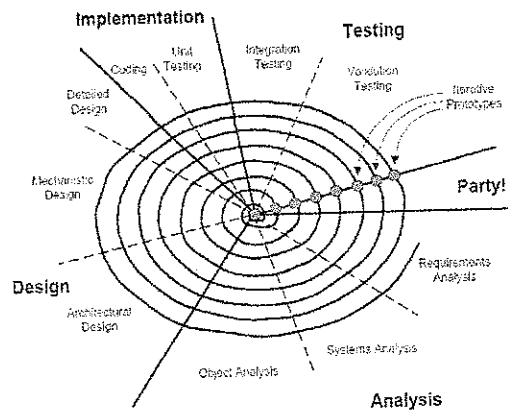


Figure 2: Iterative Lifecycle

### Ilustración 15 Ciclo de vida ROPES

Fuente:[28]

Como herramienta de soporte para la elaboración y gestión de los artefactos, su autor propone Rhapsody de I-Logix, en parte debido a que con ella se automatiza la generación del código. ROPES sin embargo no alcanza a proponer una estrategia que permita integrar el análisis de planificación en el proceso de desarrollo.

#### 6.7.2 Programación extrema (*extreme programming, xp*)<sup>20</sup>

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software creada por Kent Beck, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

A pesar de que este trabajo es un desarrollo individual orientado por los tutores, considero que la mejor forma de alcanzar objetivos es mediante la unión de trabajo, experiencia y saberes. Es por esto que describo una metodología a tener en cuenta para la realización de proyectos con características similares a este.

<sup>20</sup> Programación Extrema [Citado el: 26 de 05 de 2011.] [www.extremeprogramming.org](http://www.extremeprogramming.org), [www.xprogramming.com](http://www.xprogramming.com), [c2.com/cgi/wiki?ExtremeProgramming](http://c2.com/cgi/wiki?ExtremeProgramming)

XP se basa en realimentación continua entre todos los participantes; el cliente y el equipo de desarrollo, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Ilustración 16 Customer story and task card

Fuente: [29]

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre.

- **Las Historias de Usuario y documentación de riesgos**

Especifican los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

El tratamiento de las historias de usuario es muy dinámico y flexible, pueden romperse, reemplazarse, añadirse o ser modificadas.

Respecto de la información contenida en la historia de usuario, existen varias plantillas sugeridas pero no existe un consenso al respecto (57Ilustración 16).

- **Roles XP**

- ✓ **Programador**

El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

- ✓ **Cliente**

El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.

## 6. DISEÑO METODOLÓGICO

Para poder diseñar una metodología que se ajuste a las necesidades de un proyecto interdisciplinar es importante en primera medida definir el tipo de investigación que guía todo el proceso (investigación aplicada), luego cada una de las metodologías que mejor se ajusten a las necesidades de desarrollo particular.

El proyecto se realizó teniendo en cuenta el tipo de investigación aplicada porque esta garantiza un producto práctico, específico y original. Sin embargo sus procedimientos están pensados para llegar a la solución de manera general.

Cuando nos enfrentamos a un problema que involucra varias disciplinas y temas, la mejor forma de solucionarlo es aplicar el adagio popular "Divide y vencerás"<sup>17</sup>.

Además es necesario apoyar esos subtrabajos con metodologías que se ajusten a las necesidades particulares, bajo un mismo modelo natural (incremental e iterativo). Siempre respetando la metodología y pensando de manera modular cuando menos en el análisis de requerimientos.

### 6.1 LA INVESTIGACIÓN APLICADA

La estructura del documento evidencia el desarrollo del tipo de investigación aplicada y consiste en llevar a cabo trabajos originales realizados para adquirir nuevos conocimientos; sin embargo, está dirigida fundamentalmente hacia un objetivo práctico específico<sup>18</sup>. En este caso la originalidad radica en la integración de las tecnologías; realidad virtual, reconocimiento de imágenes, tele-operación. Cuando menos a nivel de la universidad, objetivo práctico específico es definido en la (sección 3 Objetivos) y la forma de consecución se explica en el desarrollo del modelo metodológico (sección 7 Resultados). El trabajo investigativo se evidencia en todo su proceso, la exploración, selección y aprendizaje de herramientas involucradas con cada una de las fases.

En cuanto al objetivo práctico específico se basa en la resolución de la siguiente pregunta de investigación:

### 6.2 PREGUNTA DE INVESTIGACIÓN:

¿Es posible crear un prototipo de accionamiento real a distancia de un servomecanismo mediante un mundo virtual controlado por una interfaz gestual?

---

<sup>17</sup> Julio Cesar. Frase corta pero memorable

<sup>18</sup> OCDE. Manual de Frascati. Propuesta de Norma Práctica para encuestas de Investigación y Desarrollo experimental. 2002.

### ✓ Encargado de pruebas (Tester)

El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

### ✓ Encargado de seguimiento (Tracker)

El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.

También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

### ✓ Entrenador (Coach)

Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

### ✓ Consultor

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

Gestor (Big boss)

Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

### • Proceso XP

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

- ✦ El cliente define el valor de negocio a implementar.
- ✦ El programador estima el esfuerzo necesario para su implementación.
- ✦ El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- ✦ El programador construye ese valor de negocio.
- ✦ Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la

obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

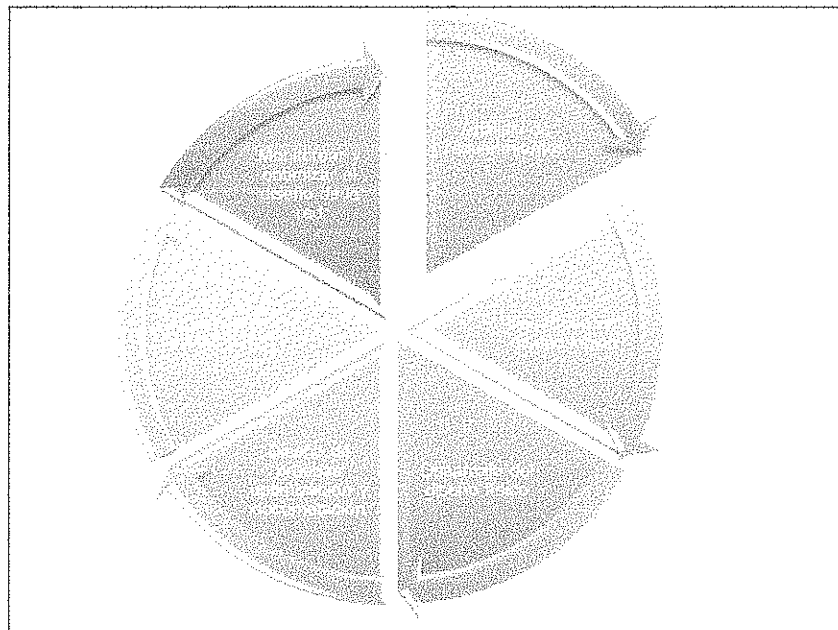
- **El ciclo de vida ideal de XP consiste de seis fases:**

- ✦ Exploración
- ✦ Planificación de la Entrega (Release)
- ✦ Iteraciones
- ✦ Producción
- ✦ Mantenimiento
- ✦ Muerte del Proyecto.

## 6.8 MÓDULO 2 COMUNICACIÓN INALÁMBRICA PARA LA TELE OPERACIÓN.

A pesar de que se trata de un radio enlace corto se usa la metodología Top-down debido a la gran experiencia de su autor y a la forma genérica en la que esta descrito el método, y la posibilidad que existe de escalar el sistema.

### 6.8.1 Diseño de red top-down :[30][31]



**Ilustración 17 Ciclo de diseño e implementación de una red**  
**Fuente: El autor basado en [30]**

Priscilla Oppenheimer ha estado en el campo de las comunicaciones de datos y sistemas de redes desde 1980, cuando obtuvo su maestría en ciencias de la información en la Universidad de Michigan. Después de muchos años como una desarrolladora de software, se convirtió en instructora técnica, enseñando a más de 3000 ingenieros de redes para más de 500 compañías. Entre ellas Apple Computer y Cisco Systems. Todo eso le ha dado la

experiencia para solucionar los problemas de diseño de redes del mundo real y desarrollar una metodología práctica de red como lo es top-down.

Esta metodología está organizada en cuatro partes que corresponden con las fases principales del diseño de una red. Es una metodología sencilla, modular y su modelo se ajusta al espiral<sup>21</sup> y por ende a las condiciones de expansión del proyecto.

- **Identificación de necesidades de sus clientes y los Objetivos:**

Comprende la fase de requisitos y análisis. Esta fase comienza con la identificación de negocio, objetivos y requisitos técnicos. La tarea de caracterización de la red existente, incluyendo la arquitectura y el rendimiento de los principales segmentos de red y dispositivos.

El último paso en esta fase es analizar el tráfico de red, incluyendo el flujo de tráfico y carga, el comportamiento del protocolo, y la calidad de servicio (QoS).

- **Diseño de red lógica**

Durante la fase de diseño de la red lógica, se desarrolla una topología. Teniendo en cuenta las características de la red; el tamaño y del tráfico. La topología puede clasificarse como simple o compleja y necesitará manejar jerarquía y modularidad. Durante esta fase, el diseñador de la red también crea un modelo de direccionamiento para la capa de red y selecciona protocolos de enrutamiento y *switching*. El diseño lógico también incluye la planificación de la seguridad, el diseño de la gestión de la red, y la investigación inicial de los proveedores de servicios relacionados y requisitos de acceso remoto.

- **Diseño de red física**

Durante la fase de diseño físico, se seleccionan las tecnologías y productos específicos que hacen posible lo que se diseñó en la fase anterior. El diseño de la red física comienza con la selección de las tecnologías y dispositivos para redes de campo, incluyendo cableado, *Ethernet switches*, puntos de conexión inalámbrica, puentes inalámbricos y routers. Selección de tecnologías y dispositivos de acceso remoto y las necesidades de flujo. Además, la investigación de los proveedores de servicios, que comenzó durante la fase de diseño lógico, debe ser completada durante esta fase.

- **Las pruebas, optimización y la documentación de su diseño de red**

Los pasos finales en el diseño de la red top-down son escribir y poner en práctica un plan de pruebas, construir un prototipo o piloto, optimizar el diseño de la red, y documentar el trabajo con una propuesta de diseño de la red. Si los resultados de las pruebas indican problemas de rendimiento, durante esa fase se deberá actualizar el diseño para incluir características tales como la optimización del tráfico, catalogación de paquetes, consultas avanzadas de ruteo y mecanismos de *switching*.

---

<sup>21</sup> Sección 5.3.6.5

## 6.9 MÓDULO 3 MUNDO VIRTUAL.

Basado en el método de diseño VRML Efectivo se construye la aplicación objetivo del proyecto, sin necesidad de llegar a la publicación.

### 6.9.1 Metodología de desarrollo VRML Efectivo:[32] [33]

Este diseño se usa porque guía el desarrollo de proyectos con realidad virtual y permite que en un futuro se lleve esta aplicación a Internet.

Un proyecto VRML, de cualquier envergadura, puede ser dividido en 7 etapas. Además su estructura permite el uso de un modelo en espiral:

#### 6.9.1.1 Especificación:

Se decide qué es lo que se va a construir, los requisitos o funcionalidades específicas.

En primer lugar hay un elemento artístico-técnico, básicamente se observa la disponibilidad de acceso a los elementos que conforman el mundo virtual y el nivel de acceso que esta información puede tener teniendo en cuenta la publicación que se desea realizar. Luego se especifica el proyecto considerando los tipos dos fundamentales:

- **Proyecto Profesional:** Es el que consiste en el modelamiento de un sitio real, por lo tanto la imaginación queda relegada
- **Proyecto Artístico:** En este caso el modelamiento está basado en las ideas que surgen de la imaginación del artista modelador, aquí el ideal estético es lo más importante, a diferencia del tipo de proyecto anterior en donde sólo se "calca" lo que existe en la realidad objetiva

Independiente de si se mezclan o no los tipos es importante documentar:

- **Descripción:** se dice en qué consiste el proyecto (profesional y/o artístico) así como el nivel de detalle a alcanzar, locaciones a modelar y finalidad del proyecto.
- **Usuarios y clientes:** se describe el perfil del usuario de ésta aplicación, así como quien encarga el proyecto. Ambas descripciones son importantes a la hora de establecer los requisitos de funcionalidad propios del proyecto.
- **Recursos necesarios:** se indican los recursos a ocupar en el desarrollo de la aplicación (software modelador, especificación de VRML, editores, software de aplicaciones gráficas, elementos gráficos, etc.).
- **Requerimientos funcionales:** aquí se establece el modo de interacción con el usuario, limitada a la interfaz que posea el usuario, en general es conveniente recomendar al usuario un visualizador específico. Una vez decidido lo anterior debe verse cómo se dibuja, si se usarán texturas para cubrir los objetos o no. se verifican las restricciones técnicas de diseño y de equipamiento computacional hardware (HW) y software (SW).

- **Planificación**

Se decide cuándo y cómo construir. El primero depende de la premura de la aplicación y el número de personas involucradas en la realización del proyecto y el otro factor dependerá de la complejidad del proyecto, se tiene en cuenta la adquisición y capacitación de las herramientas a utilizar para el diseño.

La diferencia entre proyecto simple y complejo irá dada por el nivel de detalle precisado por los usuarios, las exigencias de SW de modelado 3D, junto a HW especialmente dedicado a la computación gráfica. La complejidad del proyecto debe estar totalmente definida en la etapa de especificación, ya que en este punto (planificación), los recursos tanto HW/SW, así como de Recursos Humanos, deben estar plenamente establecidos para cada una de las etapas restantes del proyecto (muestreo, diseño, construcción, pruebas, publicación).

- **Muestreo**

Durante esta etapa se verifican todas las características acerca del objeto a modelar.

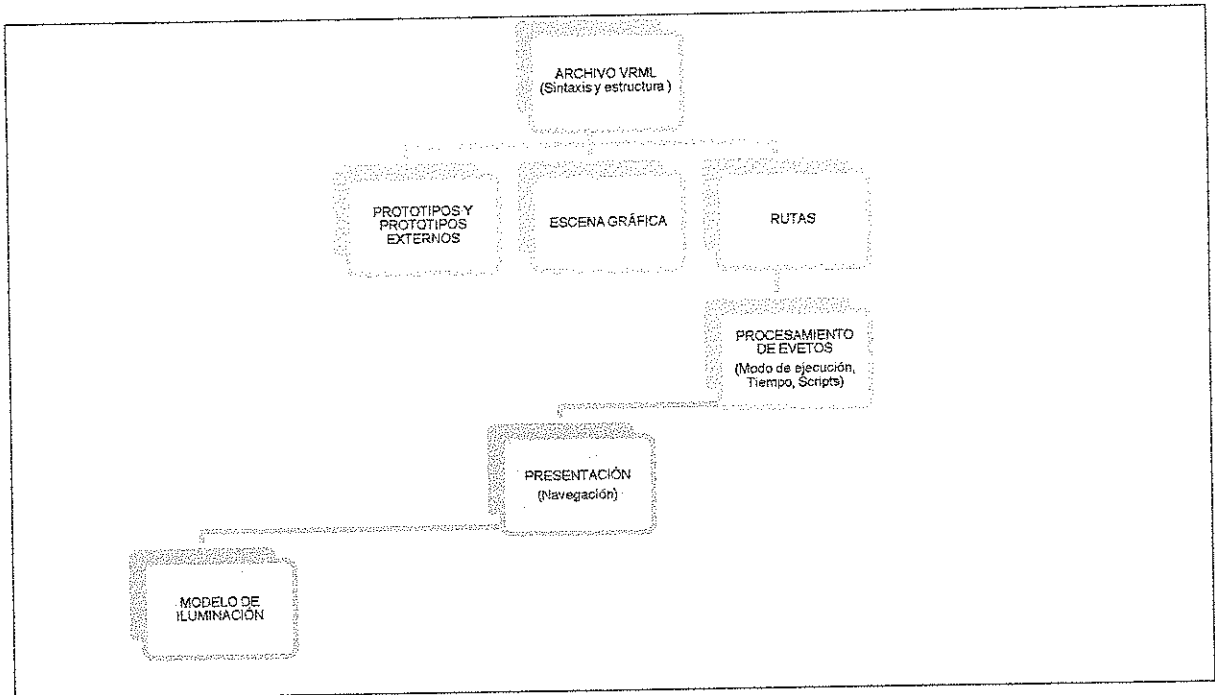
Esta forma de recopilación variará de acuerdo al objeto, se tomarían fotografías o tomas de video de diversos ángulos, las cuales posteriormente serían revisadas durante la etapa de construcción, realizar un croquis simple o recopilación de planos (elevaciones, de planta, cortes), trabajos previos en Autocad, fotografías digitales, tomas de video ubicación de los distintos elementos dentro del escenario. En fin toda la información que sea necesaria para un adecuado modelamiento de los objetos que poblarán el mundo virtual a construir.

- **Diseño:**

Antes de iniciar cualquier proceso de diseño, conviene revisar la sintaxis y estructura básica de un archivo VRML, la que se puede apreciar en la Ilustración 18.

En un mundo VRML determinado se pueden apreciar componentes estáticos y dinámicos. Los componentes estáticos son todas aquellas figuras geométricas que dan forma al objeto u objetos que se están modelando. Los componentes dinámicos en cambio son todos aquellos eventos transmitidos entre los objetos que forman el mundo virtual, a través de una ruta, que permiten a éstos presentar interacciones traducidas en movimientos, sonidos, o algún otro fenómeno perceptible para el usuario. Estos eventos pueden ser tanto internos, como externos.

Ahora es necesario comprender bien la estructura de un mundo VRML. Un mundo VRML está compuesto por una o más escenas gráficas, cada una contenida en un archivo VRML. Las escenas gráficas están compuestas por un grupo de nodos, los cuales contienen campos y eventos. Los nodos representan el componente esencial en un archivo VRML, pueden ser de distinto tipo, nodos pueden agrupar otros nodos. Los eventos son mensajes enviados entre los nodos a través de rutas, estos eventos permiten otorgar dinamismo a los componentes de un mundo VRML.



**Ilustración 18 Proceso de diseño VRML**

**Fuente: El autor basado en [33]**

En el diseño es importante tener en cuenta, identificación de objetos, especificación de atributos, identificación de eventos y comunicación entre objetos.

- **Construcción**

El proceso de construcción se puede realizar ya sea digitando el código completo mediante scripts, con programación gráfica, diagramas de bloques o “bricks”, transformando un archivo de un formato adecuado a VRML o partir de un formato estándar (ej. dxf, 3ds) teniendo en cuenta las características de los objetos virtuales, descritas en el tipo de licencia con que fueron creados.

- **Pruebas**

VRML no es un lenguaje de programación, por lo tanto no se compila antes de publicarlo, pero la detección de errores de sintaxis se puede hacer en forma de reporte a través de Cosmoplayer de Silicon Graphics. Sin embargo para Cosmoplayer el nodo Text aun no es soportado, por lo que cualquier mención de éste en un archivo no es considerada. Las pruebas permitirán apreciar si el resultado obtenido es el esperado o no y si efectivamente se ha logrado una representación reconocible con el modelo original.

- **Publicación**

Se coloca el archivo VRML en un servidor web. En esta etapa se ponen a prueba las etapas anteriores, un diseño simple, completo y preciso. Mapas de bits de gran tamaño retardan enormemente la carga del archivo, un mundo VRML debe estar completamente almacenado en la memoria del computador para su

adecuada visualización, por lo que el exceso en el tamaño produce distintos resultados como el hecho que el browser no cargue el escenario o que el computador no lo soporte por adecuadas características de rendimiento. Existen browsers que en este aspecto presentan la facilidad de poder cargar archivos VRML previamente comprimidos en formato gzip. La compresión alcanza hasta un 80%.

El proceso iterativo VRML pretende que el usuario pueda interactuar con la realidad virtual en tiempo real

## 6.10 MÓDULO 4 SERVOMECANISMO.

Utiliza el método RCM ya que además de ser un método muy claro para la ingeniería de mantenimiento, posee una estructura lo suficiente sólida para servir de metodología de diseño mecánico. En otras palabras es funcional para comenzar el Módulo o realizar labores de mantenimiento correctivo y preventivo.

En esta etapa se analiza la parte eléctrica y mecánica garantizando la fiabilidad del activo físico.

### 6.10.1 Método RCM (mantenimiento centrado en la fiabilidad)[34][35]

El método *Reliability Centered Maintenance* (RCM), mantenimiento centrado en la fiabilidad (Anexo A. Metodología RCM) tiene una amplia utilización porque permite determinar convenientemente las necesidades de mantenimiento de cualquier tipo de activo físico. También se ha definido como un método que identifica las funciones de un sistema, la forma en que pueden fallar, y establece a priori tareas de mantenimiento preventivo aplicables y efectivas, basadas siempre en consideraciones que tienen que ver con la seguridad y la economía de un sistema.

De acuerdo con la normativa SAE<sup>22</sup> JA1011(1999)[36], se deben seguir siete pasos durante el proceso:

- ✦ Identificación del entorno a analizar:
- ✦ Determinación de las funciones del elemento:
- ✦ Determinación de lo que constituirá un fallo de esas funciones:
- ✦ Identificación de las causas de los fallos funcionales:
- ✦ Identificación de los efectos de estos fallos:
- ✦ Utilización de la lógica RCM para seleccionar la táctica de mantenimiento adecuada:
- ✦ Documentación del programa de mantenimiento y depuración del mismo conforme se adquiere experiencia en la operación del elemento:

La lógica RCM se describe en Figura 1. Descripción general método RCM del Anexo A. Metodología RCM.

---

<sup>22</sup> Es una asociación global de más de 128.000 ingenieros y técnicos expertos y relacionados con las industrias aeroespacial, de automoción y vehículos comerciales ([www.sae.org](http://www.sae.org))

## 7. RESULTADOS

En este apartado se explica el desarrollo metodológico para la obtención del prototipo (servomecanismo-aplicación objetivo), las etapas de cada módulo: Ingeniería del software, Interface gestual, Mundo virtual, Comunicaciones, Robótica e Integración, se encuentran organizadas en los sub capítulos:

- ⊕ Ingeniería del software
- ⊕ Computación Gráfica
- ⊕ Realidad Virtual
- ⊕ Comunicación Inalámbrica
- ⊕ Robótica
- ⊕ Integración

### 7.1 INGENIERÍA DEL SOFTWARE

Es aquella que ofrece métodos y técnicas para desarrollar y mantener software de calidad, para llevar a cabo la aplicación práctica del conocimiento científico, al diseño y construcción de programas que den solución a los objetivos de este trabajo, independiente de las metodologías que se escojan, eso implica la documentación requerida para desarrollar, operar y mantenerlos[9].

No es sólo de la resolución de problemas con base a un diseño metodológico (sección 6 Diseño Metodológico), sino elegir la solución más apropiada, teniendo en cuenta las diferentes variables.

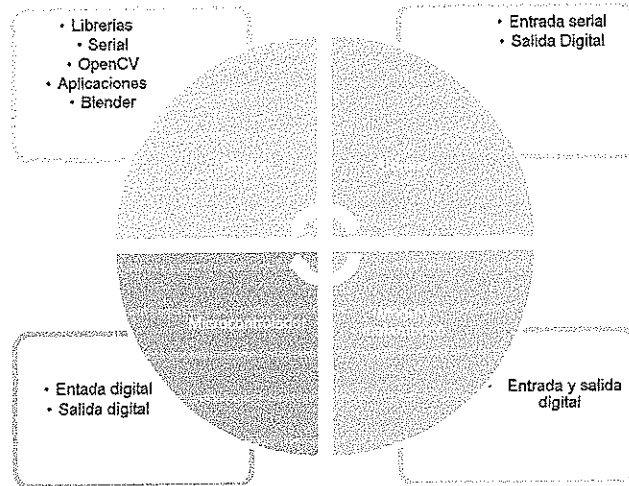
Como etapa inicial se contempló el análisis de los riesgos del proyecto de manera general para hacer la verificación de la viabilidad del mismo (ver Anexo B).

Luego se hace un análisis de los requisitos de este sistema en particular, teniendo en cuenta el propósito, su alcance, descripción, usuarios y casos de uso (ver Anexo C. Análisis de requisitos).

La definición de la estructura, funcionamiento e interacción entre las partes del software se describe a continuación:

#### 7.1.1 Arquitectura del sistema

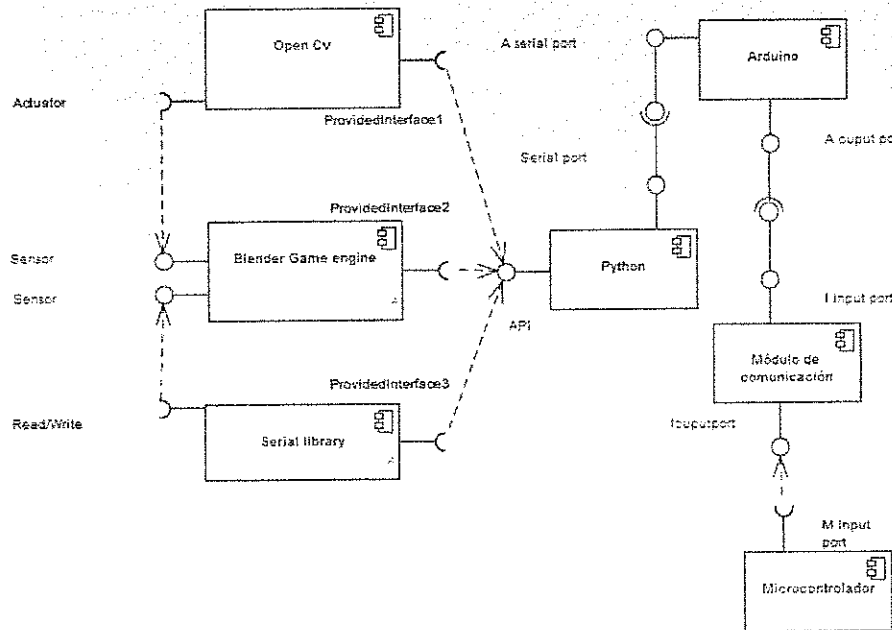
El sistema lo conforman básicamente cuatro plataformas (python, Arduino, sistema microcontrolado y en el módulo inalámbrico), (Ilustración 19) cada una de ellas maneja diferentes lenguajes de programación. La comunicación entre estas plataformas se realiza mediante interfaces que son una colección de operaciones específicas para solicitar o proporcionar un servicio (traducción de información).



**Ilustración 19 Arquitectura del sistema**  
**Fuente: El autor**

**7.1.2 Arquitectura del software:**

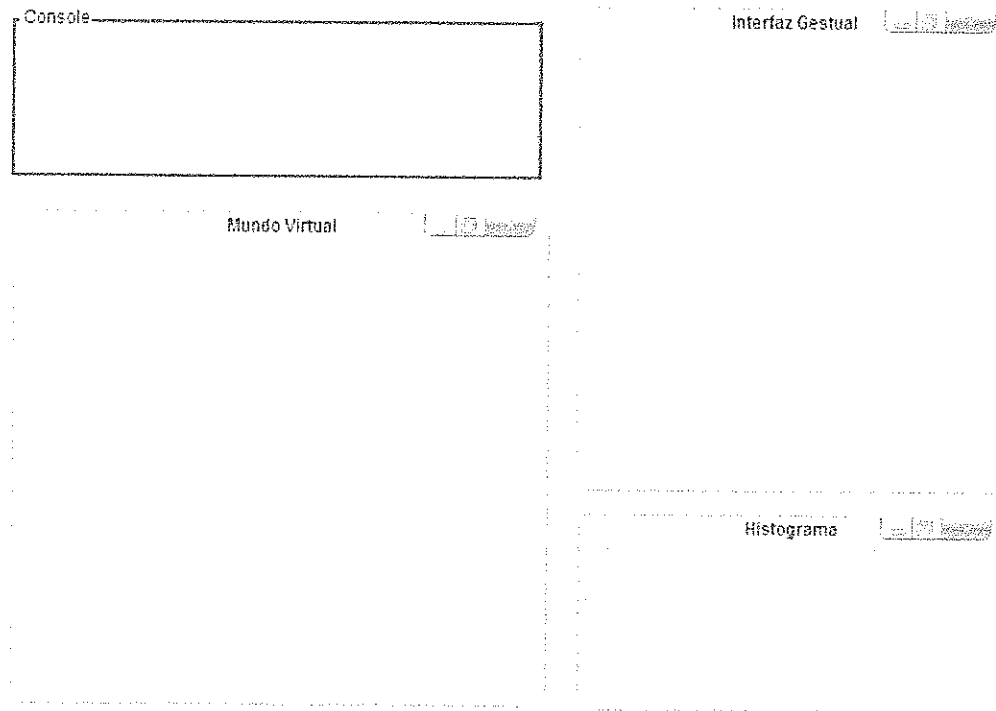
En cuanto a la arquitectura de software (Ilustración 20), está conformada por 7 componentes, dos librerías (OpenCv y Serial) que son soportadas por el motor de juegos (Game Engine) de Blender que conforma un tercer componente. Un cuarto, Python que es el lenguaje nativo de Blender por lo que funciona como plataforma para los tres módulos iniciales.



**Ilustración 20 Arquitectura del software**  
**Fuente: El autor**

Tanto el quinto como el sexto (Arduino y Modulo de comunicación) funcionan como componentes encapsulados que reciben y envían mensajes digitales con independencia. Finalmente el microcontrolador funciona en una sola vía y depende de las instrucciones del módulo de comunicación para su funcionamiento.

### 7.1.3 Diseño de interfaces:



**Ilustración 21** Diseño de interfaz del sistema.

Fuente: el autor.

Utilizando el iPlotz(R) Ilustración 21 se hace un diseño de la interfaz de usuario, básicamente consiste en la consola de comandos donde se puede evidenciar la respuesta del programa base (Phyton), una ventana de renderizado (Mundo virtual), una ventana donde aparecerá el video de la cámara (Interfaz gestual), y un por ultimo la ventana del histograma donde se hará la representación de distribución de colores de una imagen.

Una vez analizado y diseñado el sistema se procede a la elaboración de cada una de las fases del proyecto, finalmente se hace algunas pruebas para evaluar la funcionalidad tanto en las fases del proyecto como me manera general (ver Anexo D. Pruebas).

## 7.2 COMPUTACIÓN GRÁFICA

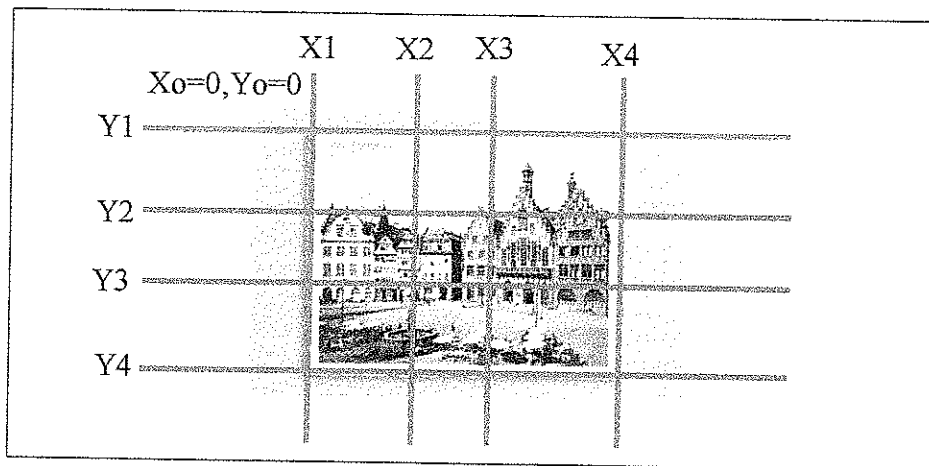
### 7.2.1 Primera fase: Interface Gestual (desarrollo de la Metodología Xp y Ropes)

En el desarrollo de la interface gestual se tuvieron en cuenta las metodologías XP y ROPES, a continuación se explica el desarrollo de las etapas necesarias dentro del diseño:

#### 7.2.1.1 Análisis de requisitos

Se especifican los requisitos funcionales y no funcionales del sistema.

- **Requisitos funcionales**



**Ilustración 22 Intervalos de control**

Fuente: El autor

	De X1 a X2	DE X2 a X3	De X3 a X4
De Y1 a Y2	Adelante Izquierda	Adelante	Adelante derecha
De Y2 a Y3	izquierda	Pausa	Derecha
De Y3 a Y4	Atrás izquierda	Atrás	Atrás derecha

**Tabla 2 Intervalos de control**

Fuente: El autor

- ✦ Capturar el video de la imagen de control
  - ✦ Generar un histograma que corresponda al área de selección
  - ✦ Guardar histograma
  - ✦ Comparar el histograma
  - ✦ Determinar la posición del histograma o posición en la que se encuentra el objeto de acuerdo a la Ilustración 22, los ejes verticales son nominados con el prefijo X y los ejes horizontales con el prefijo Y.
- El software debe ser capaz de determinar cuando el usuario se desplaza hacia la derecha, adelante, arriba y abajo, La Tabla 2 Intervalos de control muestra los intervalos asociados a los ejes.

⊕ En cuanto a la captura de imágenes es importante tener en cuenta su tamaño en pixeles (640 x 480) y los ejes que definen una malla que servirá para definir las posiciones nombradas en el ítem anterior.

- **Requisitos no funcionales:**

- ⊕ El computador que será utilizado para la aplicación es un Intel(R)Core(TM)2 Duo con procesador 1.5GHz Memoria RAM de 4G.
- ⊕ La aplicación debe ser compatible con Blender
- ⊕ Se debe usar la librería OpenCv

#### **7.2.1.2 Diseño**

La interface gestual se controla a través de la librería OpenCV. El diseño consiste en la modificación de la clase camshift mediante la adición de los métodos MovetoPort y run.

Debido a que la librería se ejecuta indefinidamente, se realiza un método run que se incluye dentro de la clase Camshift, de manera que el control de la ejecución lo realiza Blender garantizando el multiproceso.

Además, se usa en tiempo de ejecución el parámetro de posición y se ejecuta el nuevo método MovetoPort para enviar el dato de control (ver Anexo F. Descripción de clases).

#### **7.2.1.3 Traducción:**

Comprende la generación de código ejecutable a partir del diseño del sistema. Se diseña una arquitectura de software que reciba los componentes y se integran los componentes en la arquitectura.

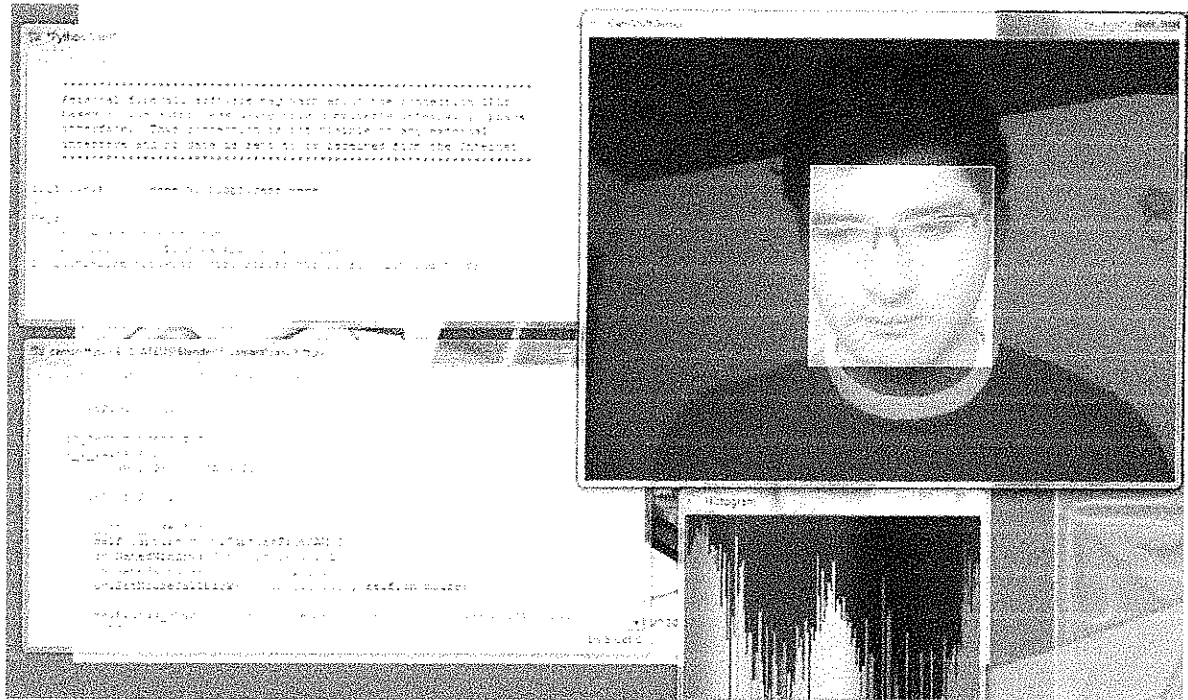
#### **7.2.1.4 Pruebas:**

Se verifica la conformidad de la aplicación, sea para encontrar defectos o para observar un cierto nivel funcional. Incluye pruebas de integración y de validación. Se efectúan pruebas exhaustivas para asegurar la funcionalidad apropiada.

En la parte superior derecha de la

Ilustración 23 se muestra el área de selección en ese momento se define la distribución de colores que van en el histograma para luego comparar su retroproyección con la nueva imagen (condiciones iniciales), con ese patrón se calcula posición y tamaño iniciales del área seleccionada.

El algoritmo diseñado conclusiones permite la identificación de la orden que se le envía al robot mediante la captura del centro del segmento de imagen seleccionada.



**Ilustración 23 Definición de coordenadas.**  
**Fuente. El autor**

En la Ilustración 25 muestra cómo se selecciona solamente el color azul, se puede evidenciar observando el histograma o la ventana inferior derecha en la ilustración.

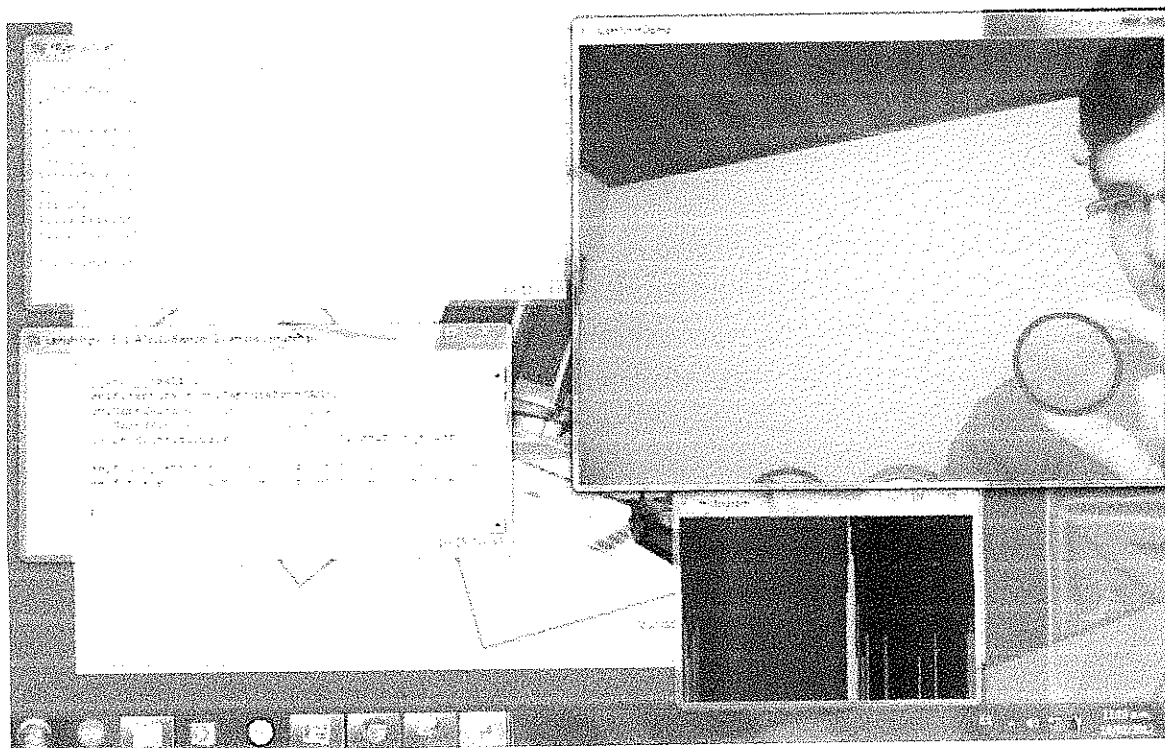


**Ilustración 24 Head Tracking**  
**Fuente: El autor**

De acuerdo a las condiciones definidas en la (Ilustración 22 Intervalos de control.) el programa retorna una cadena de texto que indica la posición en la que se encuentra, para el caso "Atrás derecha", esta información se evidencia en la ventana superior izquierda de la Ilustración 25 en letra color azul.

La ventana superior derecha "Python shell", nos indica que el programa se ejecuta con éxito. De acuerdo a nuestro modelo en espiral, este módulo está listo para la próxima iteración.

**Para simplificar la integración de esta etapa se reducen los comandos a 5 (Tabla 3 Lógica interface gestual), el comando stop controlara la ejecución, el algoritmo constantemente determinara la posición del objeto seleccionado y solo cuando este pase por la región de stop será enviado para que altere tanto el mundo virtual como el valor de control en el puerto serial.**



**Ilustración 25 Traking con una pelota azul**  
**Fuente: El autor**

Dato = Adelante		
Dato = Izquierda	SI Dato != Stop Envíe dato al Puerto Envíe dato a Blender Dato = Stop Dato = Atrás	Dato = Derecha

**Tabla 3 Lógica interface gestual**  
Fuente: El autor

### 7.3 REALIDAD VIRTUAL (SEGUNDA FASE: MUNDO VIRTUAL)

Los sistemas de tele-presencia o tele-operación mediante cámaras requieren del manejo de gran cantidad de información y están limitados por los grados de libertad que maneje el soporte de la cámara y el ángulo de visión que cubra la misma.

Con el desarrollo del diseño tridimensional, cada día hay más productos innovadores y con mayor calidad de renderizado, sin embargo son cada vez más complejos y requieren de programas especializados y gran capacidad de cómputo. Muchos de ellos no son de uso comercial.

El desarrollo de sistemas tridimensionales, mundos virtuales y videojuegos es un trabajo interdisciplinario, diseñadores gráficos, libretistas, artistas en general.

#### 7.3.1 Diseño de VRML Efectivo

Para abordar el diseño del este módulo se trabaja con una cámara estática y se diseña el entorno virtual, con el modelo a escala del robot a utilizar utilizando la metodología *virtual reality modeling language* (VRML), teniendo en cuenta la modularidad que requiere para el manejo de la información correspondiente al módulo anterior.

##### 7.3.1.1 Conceptual:

- **Descripción de los escenarios:**

La primera escena es la presentación del juego "Carrarrito lunar"

La segunda escena es un menú de selección el cual lo componen los ítems: Inicio, Salir e Información.

Es una pista de carreras en la luna donde el jugador guiará la réplica del servomecanismo, mediante la interface gestual, se usa un box para la creación del escenario, las texturas del suelo lunar y la vista espacial.

- **Libreto del videojuego:**

Es el 2020, la contaminación y los desastres naturales han hecho de este mundo un completo caos, Terremotos, tsunamis, huracanes se presentan con frecuencia, y es necesario restablecer el rumbo normal mediante remoción de desechos en el lecho marino, destrucción de las fábricas que causan contaminación, tratamiento de los escombros y haciendo labores de rescate, recoger gente en una balsa, construir barricadas, diques rompeolas, además de buscar nuevos lugares mediante la exploración espacial, donde en principio no hay posibilidad de vida. Así como también la reutilización de estos materiales de desecho y la reforestación.

Las misiones serán orientadas por 4 diosas que corresponden a los 4 elementos agua aire tierra y fuego. Azul café verde amarillo el 5 es rojo el amor.

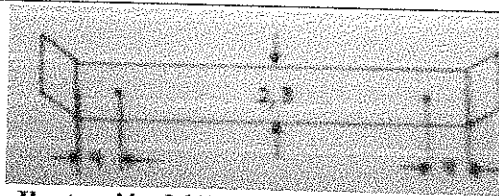
- **Descripción el objetivo del mundo virtual.**

El mundo virtual tiene como componente principal el avatar con las funciones y animaciones del robot en el escenario 3D.

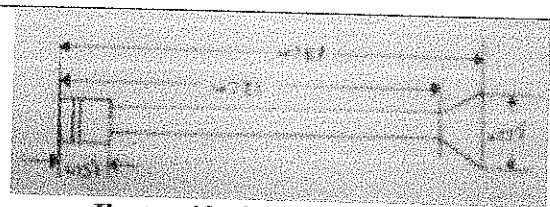
La primera misión es aprender a manejar el servomecanismo que realizará las misiones, “el cacharrito lunar” y es el objetivo de este Módulo. Para las misiones (trabajo futuro) se sugiere adicionar funciones al prototipo y consolidar la historia.

### 7.3.1.2 Planificación:

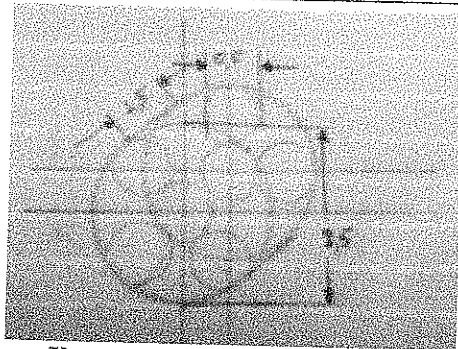
Planos de los escenarios corresponden a un proyecto artistico porque son basados en la imaginación y el robot a escala un proyecto profesional cuyos planos pueden ser elaborados a mano alzada sin tener en cuenta la proporción, lo importante es obtener las medidas.



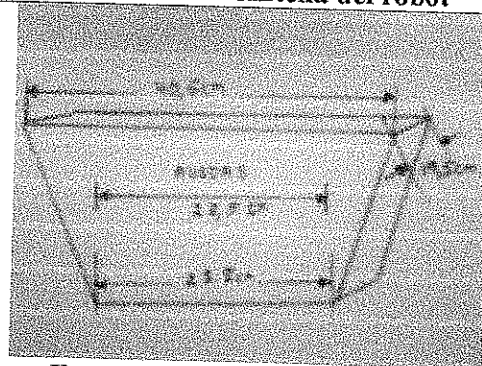
**Ilustración 26 Parachoques del robot**



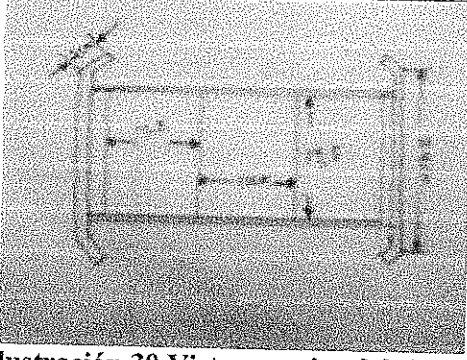
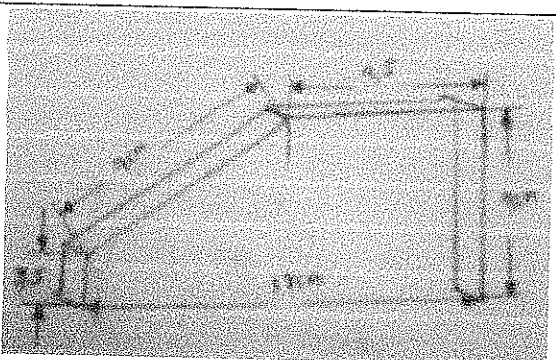
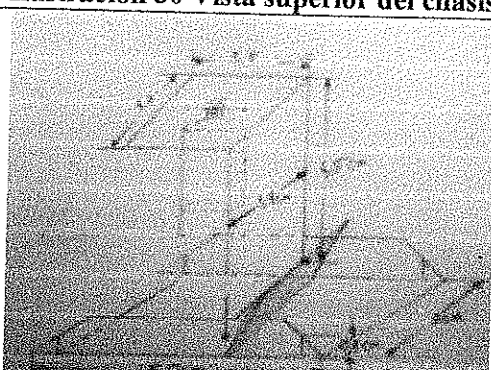
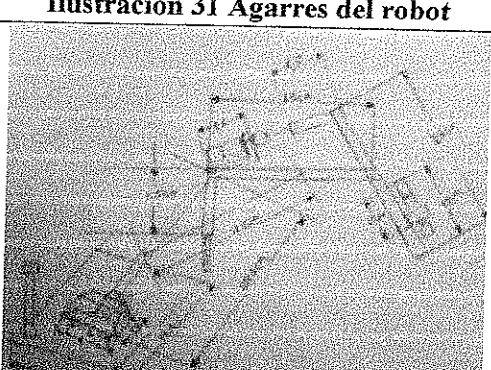
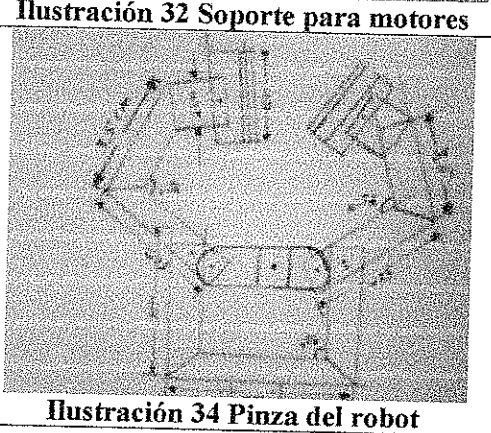
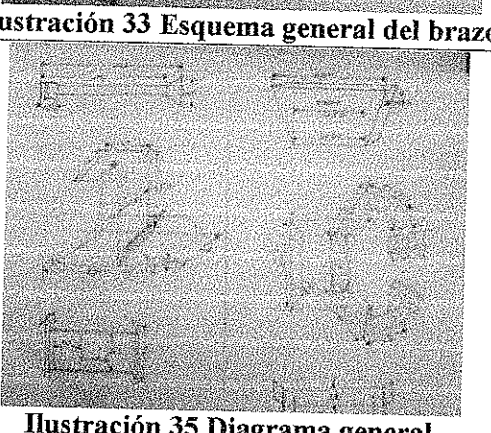
**Ilustración 27 Antena del robot**



**Ilustración 28 Ruedas Robot**



**Ilustración 29 Chasis del robot**

**Tabla 4 Planos iniciales del avatar**

**Fuente: El autor**

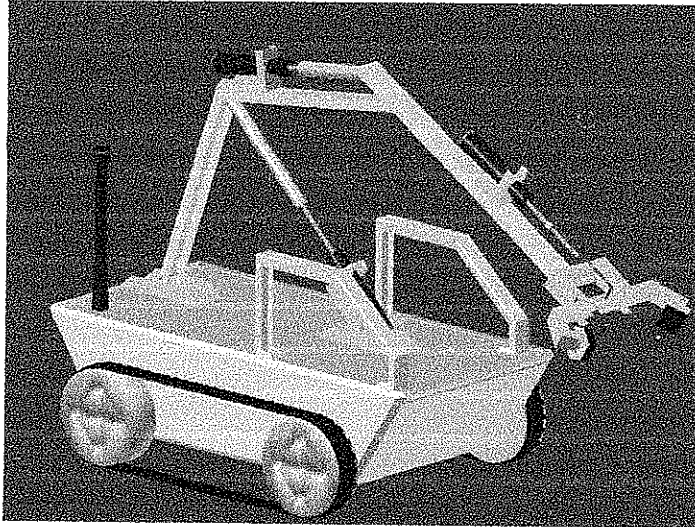
Se construye el servomecanismo o avatar en los siguientes pasos:

Modelado, texturizado, animación y finalmente programación.

En cuanto al Punto de referencia dentro del escenario, se usará una vista en tercera persona para el control del servomecanismo virtual.

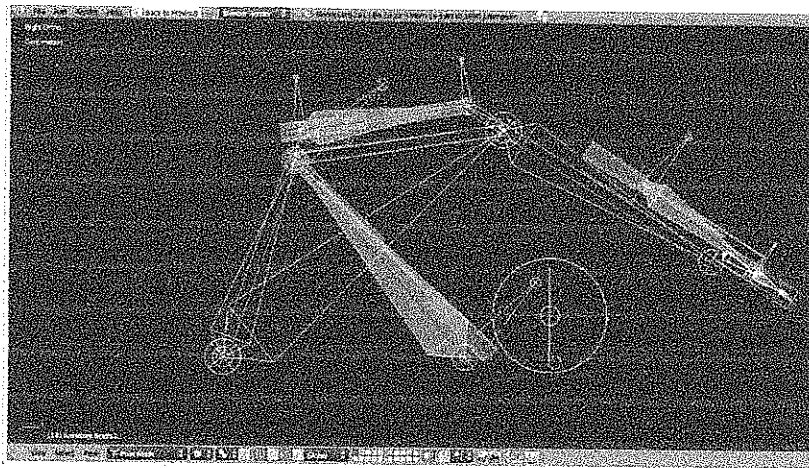
### 7.3.1.3 Diseño y muestreo:

Lo primero que se diseña es el avatar Ilustración 36, de acuerdo a los planos iniciales (Tabla 4 Planos iniciales del avatar) con las técnicas que más se ajusten al modelado de la pieza en particular y las piezas se diseñan como objetos independientes.



**Ilustración 36 Modelo 3D del avatar texturizado**  
**Fuente: El autor**

Una vez creado el avatar se procede a la identificación de la armadura, en este paso se analiza la dinámica del servomecanismo teniendo en cuenta los puntos de rotación más no el diseño de la estructura Ilustración 37. El sistema es similar a juego de pistones.



**Ilustración 37 Armadura del brazo robótico en Modo edición**  
**Fuente: El autor**

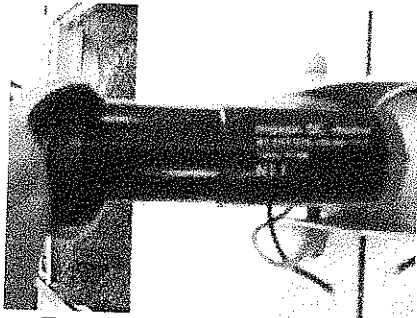
El siguiente paso será la construcción de las animaciones mediante puntos de control. Y finalmente se implementara la lógica del juego mediante los scripts y los diagramas de bloques.

- **Geometría:**

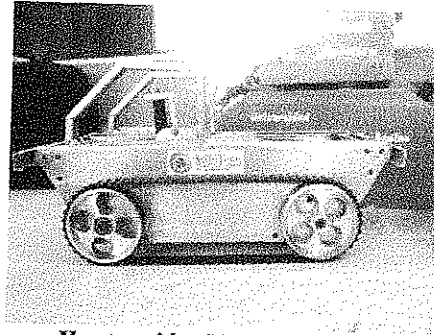
La geometría de los objetos se basa estructuras poligonales formadas a partir de puntos, teniendo en cuenta las medidas del robot a escala para el diseño del avatar. Mientras que el diseño del escenario tiene la geometría de un *box*.

- **Renderizado de los objetos:**

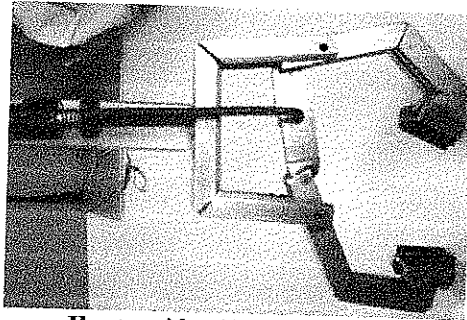
El proceso de renderizado se basa en los colores del prototipo real (Tabla 5 Fotografías base para el diseño de texturas), usa distorsiones de los colores primarios en vez de sectores de fotografía, mejorando así la uniformidad y reduciendo el tamaño de los archivos de imagen.



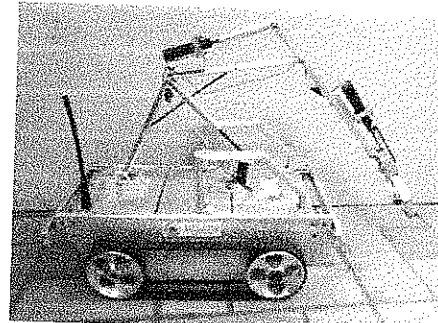
**Ilustración 38 Detalle del motor**



**Ilustración 39 Detalle base**



**Ilustración 40 Detalle pinza**



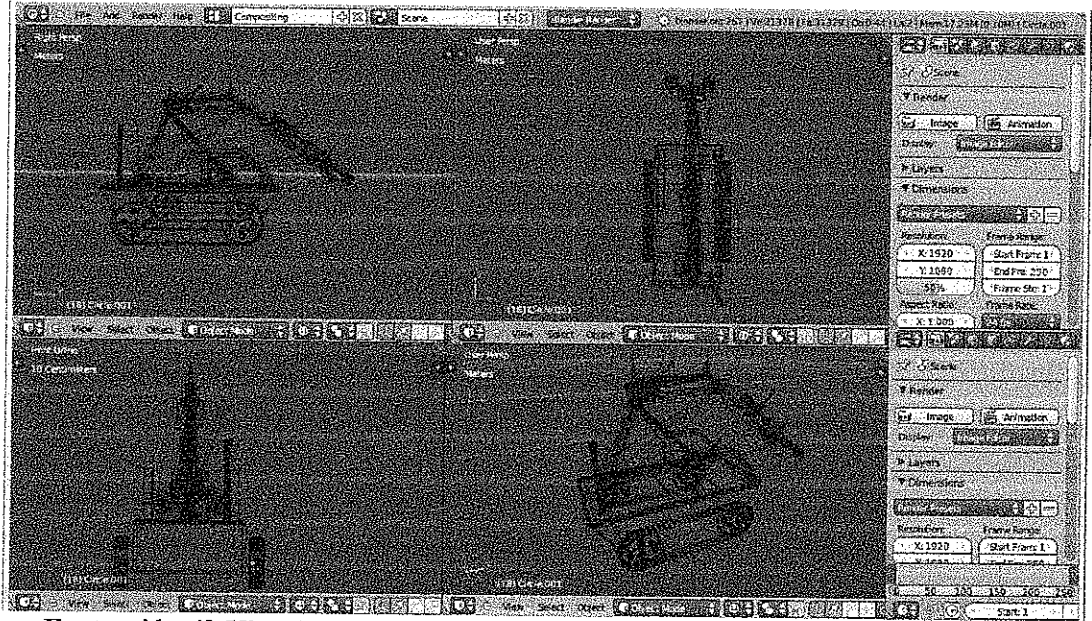
**Ilustración 41 Diagrama general**

**Tabla 5 Fotografías base para el diseño de texturas**

Fuente: El autor

- **Las técnicas:**

Para el modelado básicamente se usó *curve modeling* ya que se trabajó con medidas a escala del robot a utilizar, en el evento de crear escenarios con fallas geográficas es importante considerar el método de mallas, la figura geométrica diseñada con *curve modeling* presentó buen desempeño.

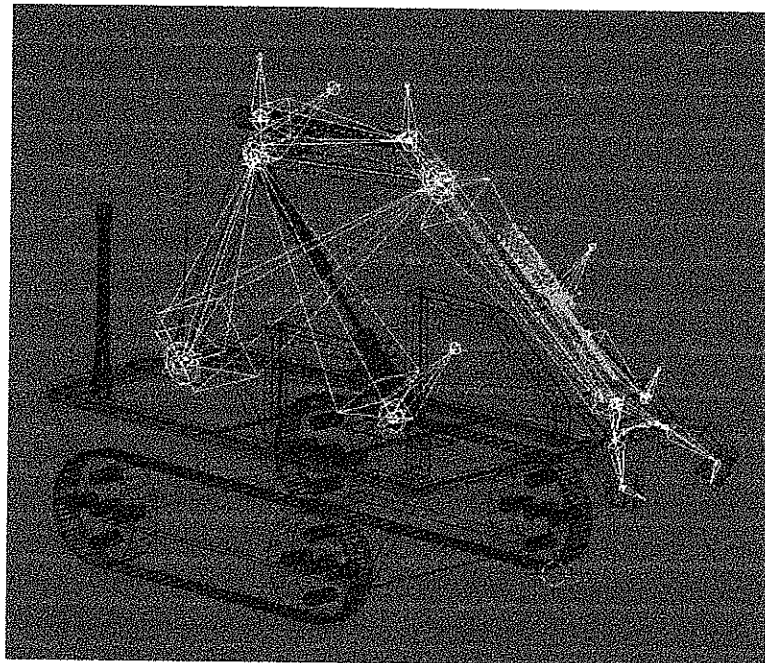


**Ilustración 42 Vista lateral, trasera, superior y perspectiva ortogonal del avatar**

Fuente: El autor

#### 7.3.1.4 Construcción:

Es importante que el diseño de las piezas se realice de manera independiente para poder realizar de forma rápida la asociación de la armadura con los objetos independientes Ilustración 43.

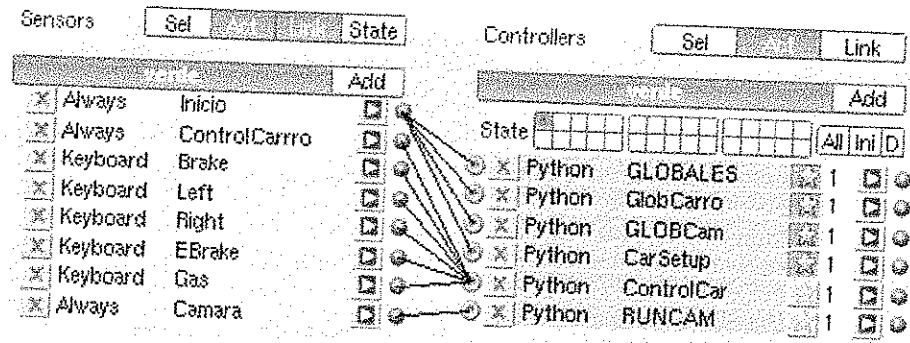


**Ilustración 43 Selección de objetos en el modo edición**

Fuente: El autor

Luego de la construcción del avatar, las piezas del avatar se convierten en único objeto exceptuando las ruedas, a este le aplicamos la programación. En la versión actual (febrero 2012) de Blender también es posible mediante la armadura.

Una vez seleccionado el objeto se asocian los scripts de configuración como se muestra en la Ilustración 44 *Logic Bricks*.

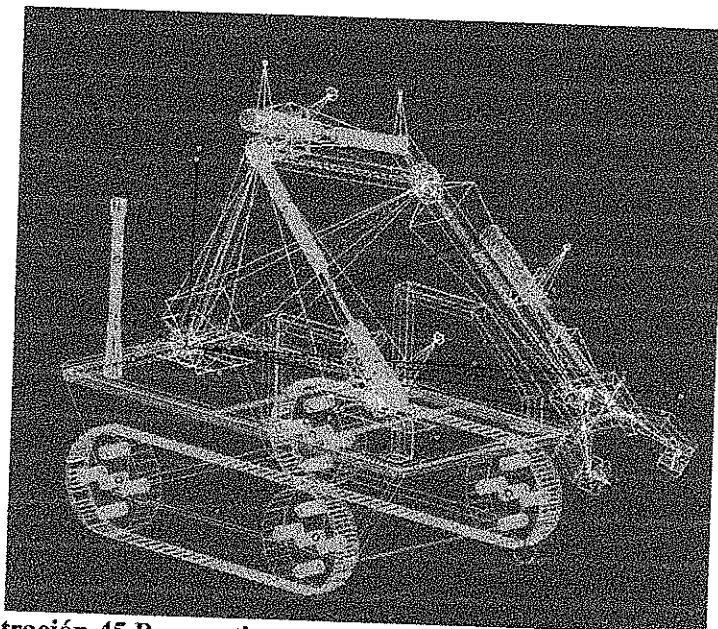


**Ilustración 44** Lógica para el servomecanismo

Fuente: El autor

De esta manera se combina la programación visual *Logic Bricks* con los scripts de Python, aumentando la funcionalidad de Blender.

La Integración de los objetos en el escenario virtual se hace de manera natural en Blender mediante la opción *file/append or link*.



**Ilustración 45** Perspectiva ortogonal del robot con emparentado

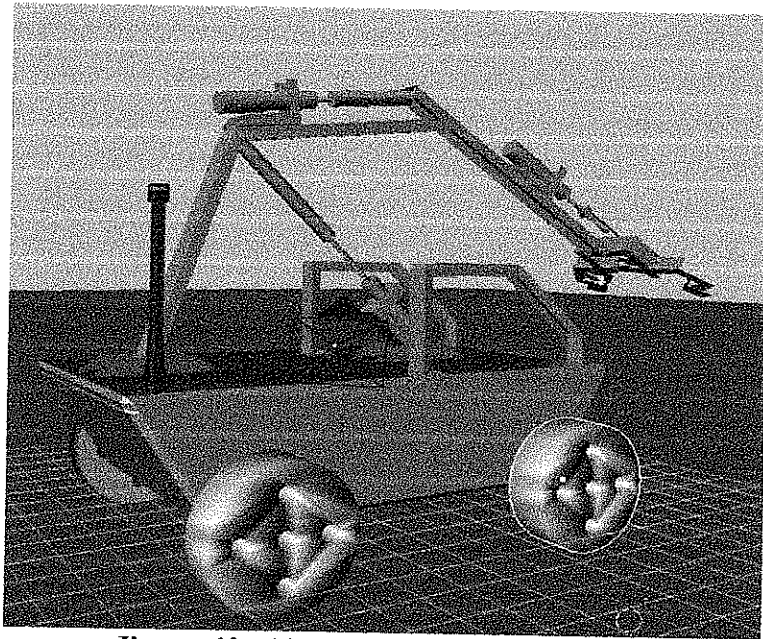
Fuente: El autor

El emparentado que se utilizó para la programación de los movimientos del avatar se encuentra en la Ilustración 45.

La programación de las propiedades físicas del servomecanismo se realiza con scripts que pueden ser obtenidos a través de la comunidad Blender o paginas relacionadas<sup>23</sup> liberadas bajo la licencia (CC) Creative Commons Attribution 3.0 Unported License<sup>24</sup>.

#### 7.3.1.5 Pruebas

En esta etapa se encontraron problemas de compatibilidad de las versiones, tanto OpenCV como la librería serial solo trabajan con la versión 4.9 de Blender, con el tiempo podrán ser integrados, por lo que no fue posible integrar la animación, para lo cual se hizo una migración de los objetos por lo que se perdieron algunas texturas y las animaciones.



**Ilustración 46** Renderizado en versión 4.9

**Fuente: El autor**

#### 7.3.1.6 Publicación:

En el momento es stand alone, pero para una segunda fase del proyecto fácilmente se puede llevar a la web mediante la conversión de los archivos a VRML para la última versión sólo es necesario llevarlo a un formato estándar (ej. dxf, 3ds), sin embargo la versión sobre la que se trabajó permite la conversión a VRML 1.0 directamente, sin necesidad de ningún paso adicional.

<sup>23</sup> Tutorials for Blender [Citado el: 26 de 05 de 2011.] [www.tutorialsforBlender3d.com](http://www.tutorialsforBlender3d.com)

<sup>24</sup> Creative commons [Citado el: 26 de 05 de 2011.] <http://creativecommons.org/licenses/by/3.0/>

## 7.4 COMUNICACIÓN INALÁMBRICA (INTERFAZ DE COMUNICACIÓN PC A MÓDULO INALÁMBRICO)

Blender garantiza mediante la librería serial que los comandos, enviados por teclado o de acuerdo a la interpretación de la interface gestual, estarán en el puerto de comunicaciones serial donde arruino se encarga de interpretarlo:

### 7.4.1 Objetivo:

Realizar una interface que permita tomar una cadena de caracteres que se encuentra en el puerto serial y enviar un dato de 4 bits o señal de control, de acuerdo a la Tabla 6, al módulo inalámbrico.

No	BINARIO	Señal de control	Letra	ASCII
0	0000			
1	0001	Adelante	W	87
2	0010	Atrás	S	83
3	0011	Derecha	D	68
4	0100	Izquierda	A	65
5	0101	Abrir pinza	O	79
6	0110	Cerrar pinza	L	76
7	0111	Subir brazo	I	73
8	1000	Bajar brazo	K	75
9	1001	Subir Antebrazo	U	85
10	1010	Bajar Antebrazo	J	74
11	1011	Adelante con Rueda derecha	E	69
12	1100	Adelante con Rueda Izquierda	Q	81
13	1101	Atrás con Rueda derecha	C	67
14	1110	Atrás con Rueda Izquierda	Z	90
15	1111			

**Tabla 6 Señal de control**

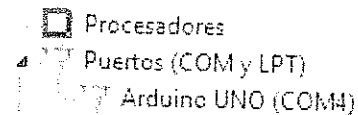
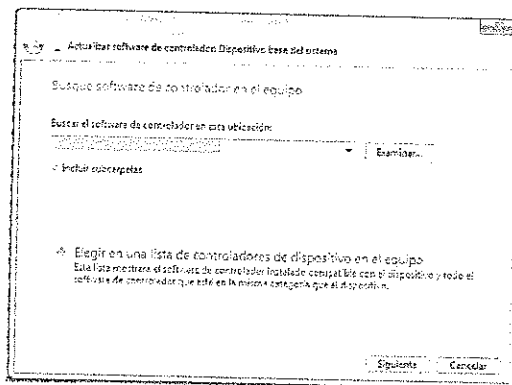
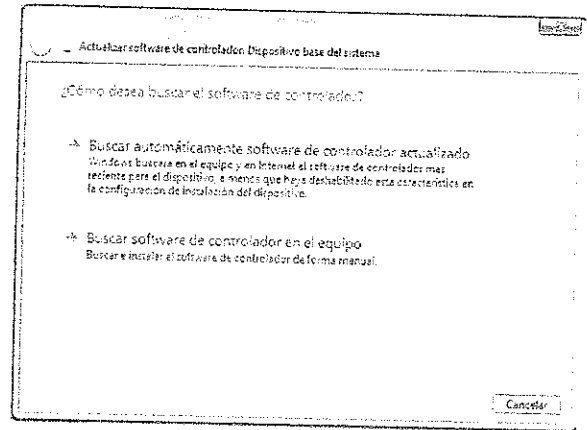
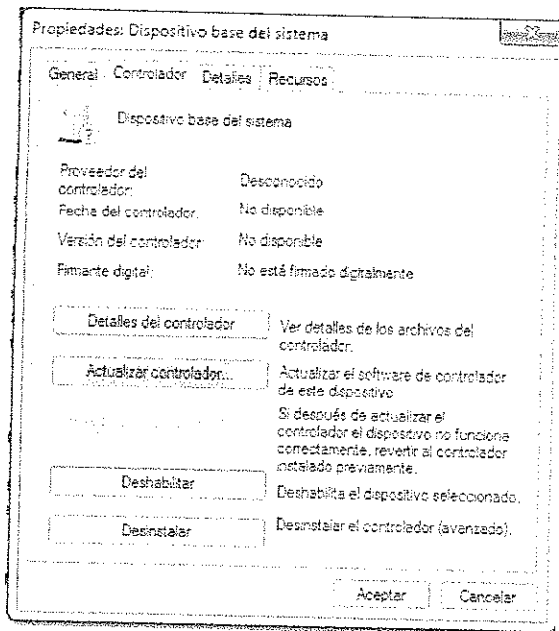
**Fuente: El autor**

### 7.4.2 Herramientas:

Para la construcción de esta interface se escogió la tarjeta arduino uno, debido a su simplicidad de codificación y su arquitectura física implementada.

### 7.4.3 Desarrollo:

El primer paso es la instalación de los drivers para que Arduino pueda simular el puerto de comunicaciones serial. Si Arduino se encuentra instalado, al conectar el cable USB aparece, la ventana de información de Windows, seleccionamos la ubicación de los drivers en la carpeta donde se encuentra Arduino y finalmente verificamos la emulación en el Arbol de dispositivos instalados, mediante la ventana de propiedades de controlador ver (Ilustración 47 Proceso de instalación puerto COM o USB simulado).



**Ilustración 47** Proceso de instalación puerto COM o USB simulado  
**Fuente:** El autor

La explicación del código se encuentra desarrollada en el Anexo G.

#### 7.4.4 Pruebas

Mediante esta etapa se determina que el tiempo mínimo de respuesta de Phyton para la comenzar a enviar y recibir los datos de control es de 1.5 segundos y que esto se debe a la instanciación de la clase serial.

### 7.4.5 Desarrollo de la metodología top-down

Para la tercera fase: Módulo de comunicación inalámbrica para la tele operación, los módulos utilizados para la transmisión de la señal de control son PT2262 y PT2272 como propuesta metodológica se presenta la metodología top-down.

#### 7.4.5.1 Identificación de necesidades de sus clientes y los Objetivos:

El objetivo es llevar un dato de 4 bits, mediante ondas de radio, hacia el circuito de control del servomecanismo, de manera que las órdenes que se impartan sean realizadas por el mismo.

El área de trabajo no supera los 20 metros.

#### 7.4.5.2 Red lógica

La Ilustración 48 describe como es se transforman o viajan los datos a través de la red.

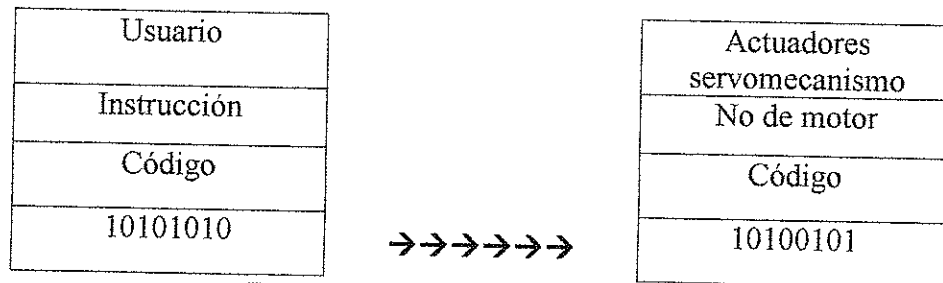


Ilustración 48 Diseño de la red lógica  
Fuente: El autor

Este es un enlace punto a punto, en la banda libre de radio frecuencia, no requiere repetidores.

#### 7.4.5.3 Red física

Muestra la cadena de comunicación de cada uno de los nodos que pertenecen a la red.

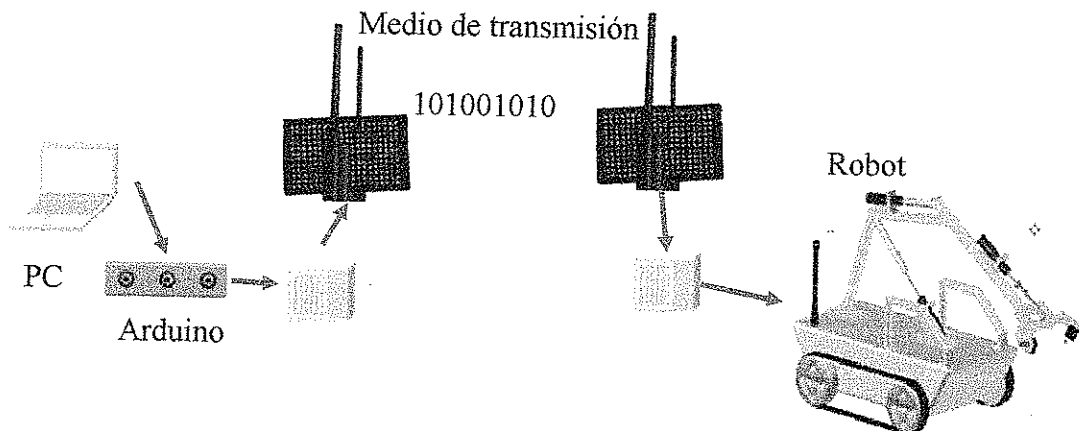


Ilustración 49 Diseño de la red física  
Fuente: El autor

#### 7.4.5.4 Las pruebas, optimización y la documentación de su diseño de red

Durante el desarrollo de la se detectaron problemas de potencia en la transmisión de Por lo que se decide eliminar algunos datos de control y ajustar el código Tabla 7.

Usualmente la tecnología inalámbrica utiliza ondas de radiofrecuencia de baja potencia y una banda específica, de uso libre o privada para transmitir, entre dispositivos.

Estas condiciones de uso sin necesidad de licencia o libertad, ha propiciado que cada vez se encuentren más equipos; hornos microondas, computadores y algunos celulares, que utilizan las ondas para conectarse en el espectro de 2.4GHz, a través de redes inalámbricas.

Es importante evaluar los problemas de potencia en la transmisión, pueden ser mejorados cambiando la banda de operación pero esto solo es posible en arquitecturas abiertas.

Para las próximas iteraciones sería interesante considerar el uso de internet satelital.

No	BINARIO	Señal de control	Letra	ASCII
0	0000			
1	0001	Adelante	W	87
2	0010	Atrás	S	83
3	0011	Derecha	D	68
4	0100	Izquierda	A	65
5	0101	Abrir pinza	O	79
6	0110	Cerrar pinza	L	76
7	0111	Subir brazo	I	73
8	1000	Bajar brazo	K	75
9	1001	Subir Antebrazo	U	85
10	1010	Bajar Antebrazo	J	74
11	1011	Adelante	E	69
12	1100	Abrir pinza	Q	81
13	1101	Cerrar pinza	C	67
14	1110	Subir Antebrazo	Z	90
15	1111			

**Tabla 7 Modificación de código para señal de control**

**Fuente: El autor**

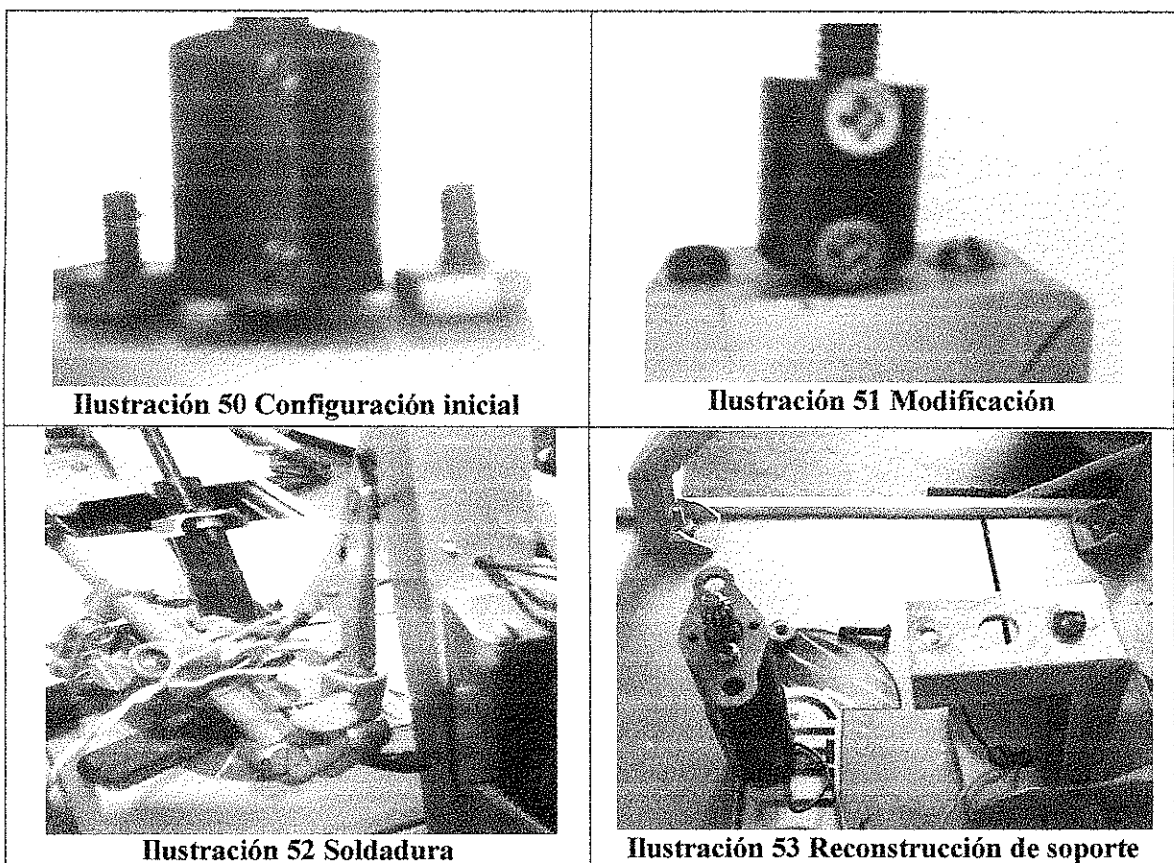
## 7.5 ROBÓTICA (CUARTA FASE: SERVOMECANISMO)

Como la palabra lo indica este módulo se encarga de la parte mecánica y electrónica, para la primera etapa se realizó un mantenimiento preventivo y correctivo tomando como referencia la lógica RCM Anexo A. Metodología RCM

Para la segunda es necesario evaluar los circuitos de potencia y control del servomecanismo, posiblemente rediseñarlos o ajustarlos garantizando su funcionalidad.

### 7.5.1 Sistema mecánico

La estructura que soporta los motores se encuentra sobre el brazo, fue hecha doblando aluminio, esos pliegues que se forman tienen riesgo de fallo.



**Ilustración 54 Mantenimiento**

Fuente: El autor

La Ilustración 54 Mantenimiento corresponde el detalle de algunas piezas clave para el funcionamiento del motor. La transmisión utilizaba inicialmente tornillos piconeros sin cabeza, lo que ocasionaba dificultad para el ajuste y por ende deslizamiento del perno. Finalmente para que fuese posible el giro de la nueva transmisión se cambió la posición de las arandelas.

Para la reparación de las fisuras y la reconstrucción de piezas, se utilizó soldadura fría.

## 7.5.2 Sistema Electrónico

En este caso no existían circuitos electrónicos, por lo que se procede a su diseño usando la metodología en espiral.

### 7.5.2.1 Análisis de requerimientos:

La siguiente tabla explica mediante las casillas marcadas con una x roja los puertos que deben ser activados cuando la palabra de control o comando se ejecute (se presente en los puertos de entrada del microcontrolador).

No	Bin	Comando	Motor									
			M1		M2		M3		M4		M5	
			BASE		Rueda Izquierda		Rueda Derecha		Codo		Pieza	
		Polaridad	+	-	+	-	+	-	+	-	DER	IZQ
		PIN PUENTE H	2(1)	7(1)	10(1)	15(1)	2(2)	7(2)	10(2)	15(2)	16(3)	15(3)
		PIN del PIC	35	36	34	33	29	30	28	27	26	25
			B2	B5	B3	B0	D6	D7	D5	D4	C7	C6
1	0001	Adelante			X			X				
2	0010	Atras				X		X				
3	0011	Derecha			X			X				
4	0100	Izquierda				X		X				
5	0101	Pieza arriba									X	
6	0110	Pieza arriba										X
7	0111	Base sube	X									
8	1000	Base Baja		X								
9	1001	Codo sube								X		
10	1010	Codo Baja							X			
11	1011	Adelante con rueda DER						X				
12	1100	Atras con rueda DER						X				
13	1101	Adelante con rueda IZQ			X							
14	1110	Atras con rueda IZQ				X						

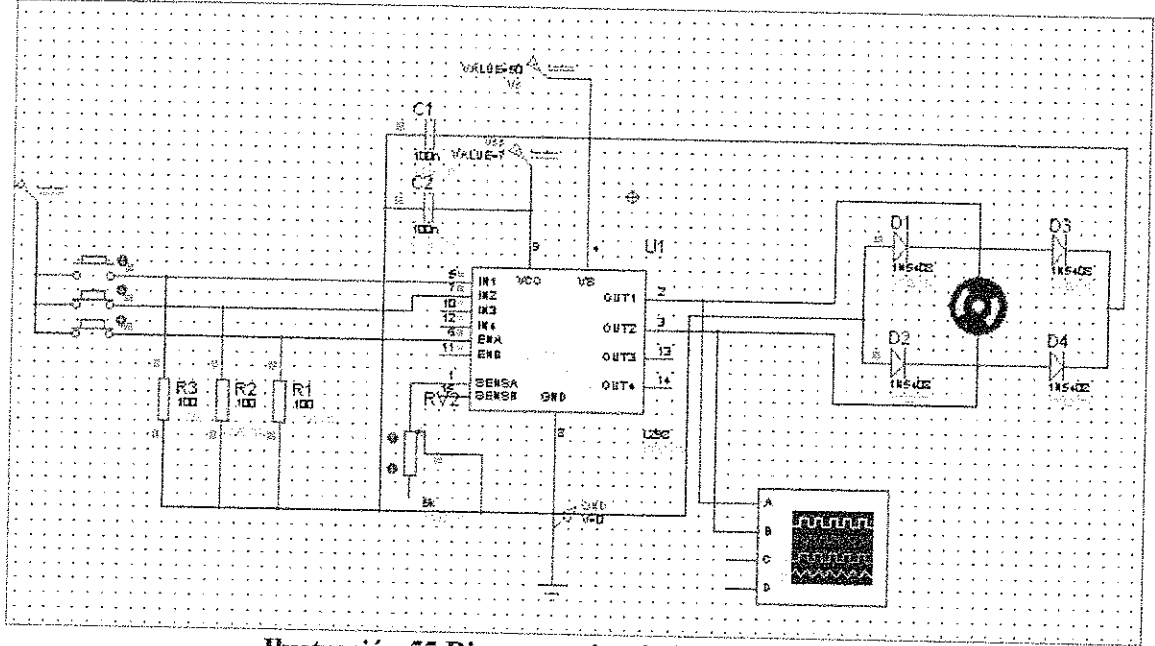
**Tabla 8 Comandos para el robot**  
Fuente: El autor

### 7.5.2.2 Diseño:

En esta etapa se describen el circuito de potencia o puente h y el diseño final con el microcontrolador.

• **Puente H**

El circuito usa los materiales descritos en la Tabla 9, su diagrama circuital está dispuesto en la Ilustración 55.



**Ilustración 55 Diagrama circuital driver motores DC**

**Fuente: El autor**

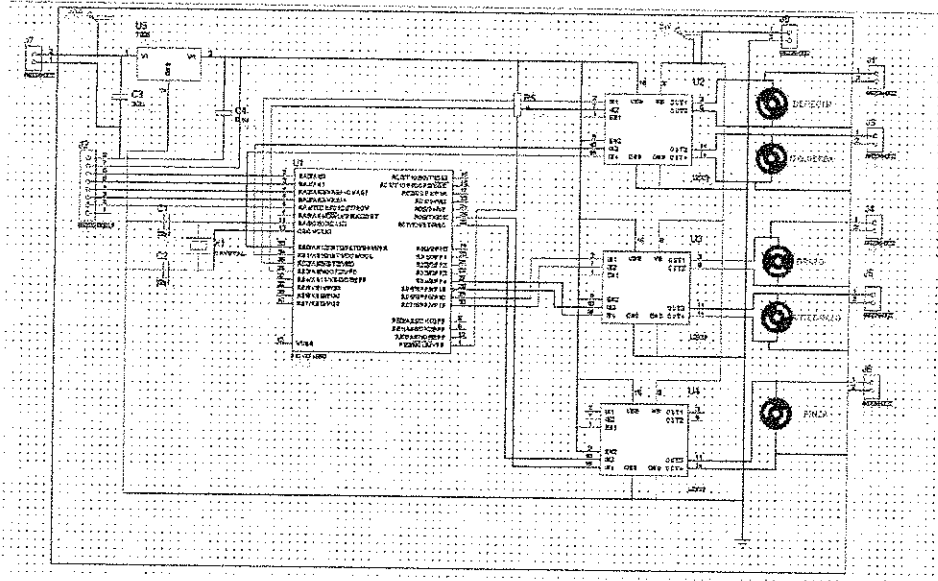
Nombre	Descripción	Simbología	Valor
Resistencias		R1, R2, R3	100K
Potenciómetro	Regulador de velocidad.	RV2	5K
Condensadores		C1	100n
Diodos	Diodo de potencia	D1, D2, D3, D4	3A
Interruptores	Simulan la entrada de control.	Ninguna	No aplica.
Fuentes	Potencia		50V
	Control		7V
Driver	CI Puente H para el control de motores y control de velocidad	L298	No aplica.

**Tabla 9 Componentes C1L258**

**Fuente: El autor**

• **Micro controlador:**

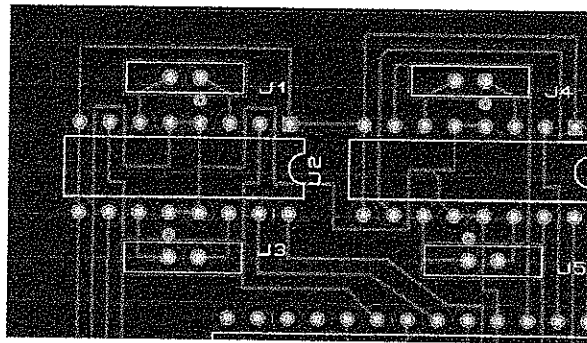
La Ilustración 56 corresponde al diagrama circuital final en ISIS, en él se encuentran distribuidos los componentes utilizados para la aplicación, además es posible simular el movimiento de los motores de acuerdo a las señales presentes en el puerto A del microcontrolador, por ejemplo mediante Dipswitch.



**Ilustración 56 Diagrama circuital**

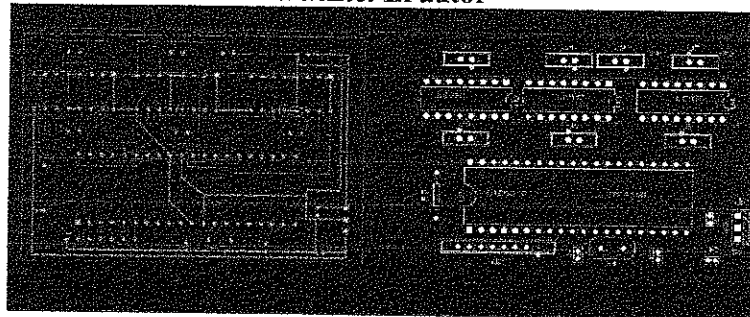
**Fuente: El autor**

Una vez se simula el sistema en ares, se procede a situar los componentes utilizados sobre una *board* virtual en el diseñador de impresos ISIS, Ilustración 57, luego de distribuyen las capas para la impresión Ilustración 58.



**Ilustración 57 auto ruteo en ares**

**Fuente: El autor**

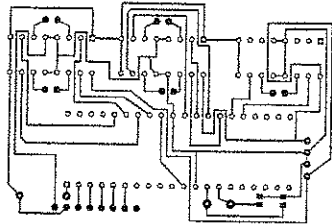


**Ilustración 58 Diseño de impresión por capas**

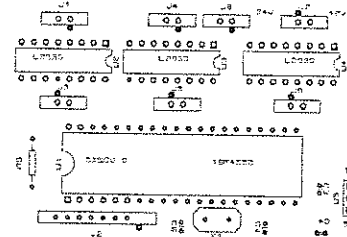
**Fuente: El autor**

### 7.5.2.3 Implementación

Se usa el principio de la serigrafía para la elaboración de la placa de cobre: Primero se alista la placa con lija fina para dejarla limpia de óxido o cualquier otra impureza, con una goma lija, que es una lija fina que deja la superficie totalmente lisa, normalmente se usa en salones de belleza o con esponjilla BOMBRIL(R), después la lavaremos con alcohol para poder imprimir el tóner.

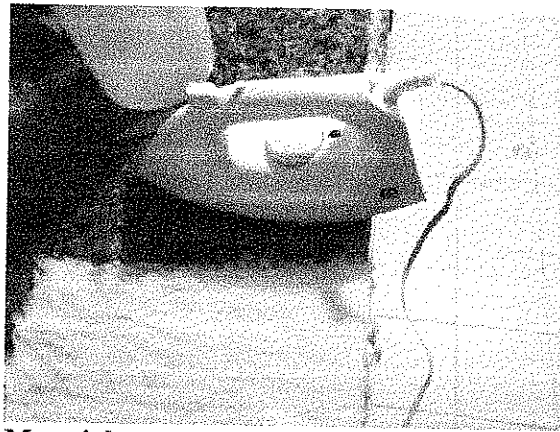


**Ilustración 59** Capa de cobre  
Fuente: El autor



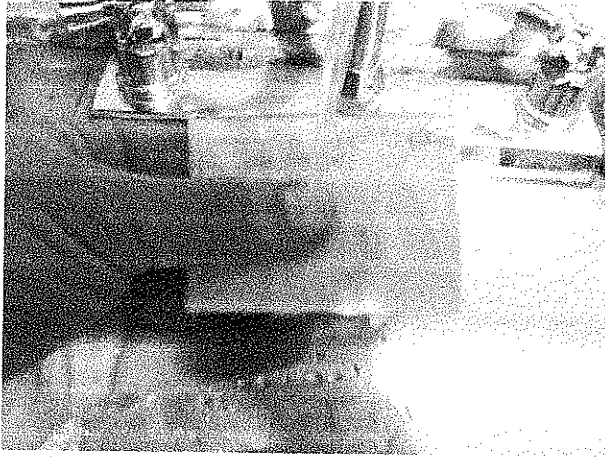
**Ilustración 60** Capa de etiquetas  
Fuente: El autor

El esquema de la figura anterior se imprime en papel glossy (papel fotográfico para impresoras de inyección de tinta), con una impresora láser y se corta a la misma medida que la placa.



**Ilustración 61** Materiales para la impresión del tóner en la placa de cobre  
Fuente: El autor

Se coge el conjunto placa papel y se pone sobre una tabla o toalla, se coloca la plancha a la temperatura más alta y se la pasa durante un rato sobre el papel, hasta que esté totalmente pegado (el tiempo depende de la máxima temperatura de la plancha unos 6 o 8 minutos y el espesor del papel)

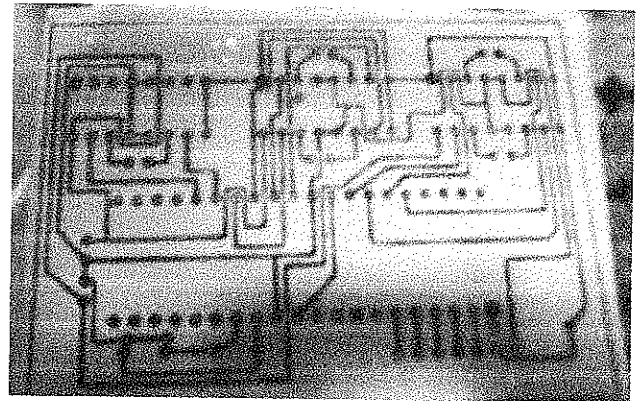


**Ilustración 62 Proceso de remoción del papel fotográfico**  
Fuente: El autor

Se deja enfriar la placa con el papel, es importante que la placa este fría, se sumerge en agua hasta que el papel se pueda retirar suavemente con las yemas de los dedos.

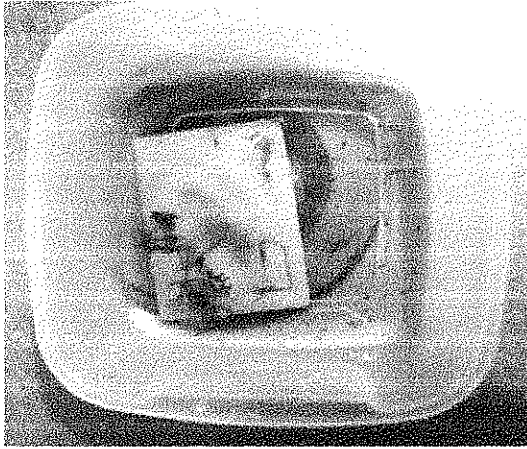


**Ilustración 63 Proceso de remoción del papel fotográfico** Fuente: el autor

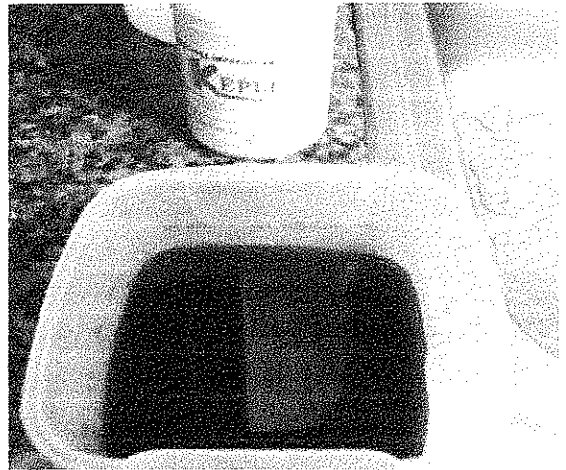


**Ilustración 64 Tóner impreso sobre la placa de cobre** Fuente: el autor

En la Ilustración 64 se muestra el producto terminado, luego se quita el cobre sobrante para dejar la placa con las pistas solamente.

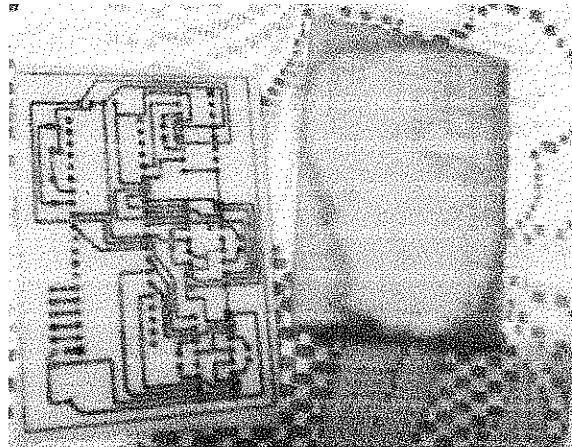


**Ilustración 65** Proceso de quemado con ácido nítrico Fuente: el autor



**Ilustración 66** Proceso de quemado con cloruro férrico Fuente: El autor

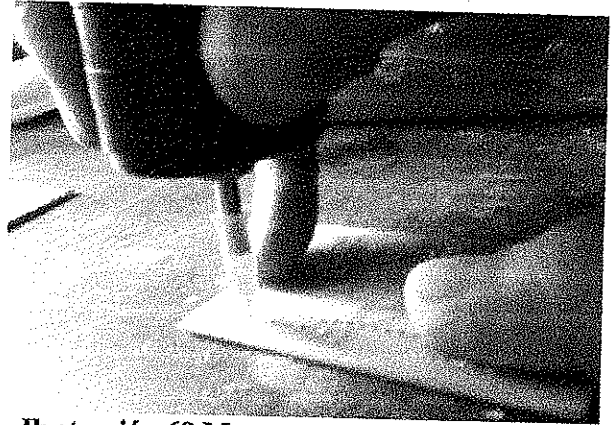
Es importante usar una solución de nítrico mezclada con agua o cloruro férrico que es menos agresivo ya que puede ocurrir lo que se muestra en la Ilustración 64.



**Ilustración 67** Diferencia entre el ácido nítrico y el cloruro férrico Fuente: El autor



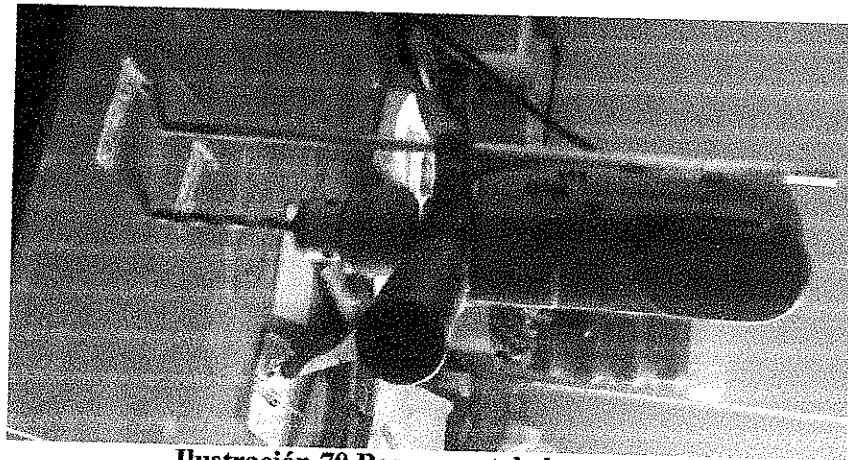
**Ilustración 68** Detalle motor con aguja hipodérmica Fuente: El autor



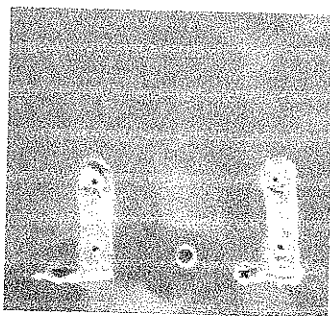
**Ilustración 69** Motor con aguja hipodérmica Fuente: El autor

Las Ilustración 68 e Ilustración 69 describen el proceso que se usó para realizar los orificios a la placa, este es un recurso muy importante cuando no se tiene *motortool* o taladro a la mano, se puede improvisar uno con un motor y una aguja hipodérmica.

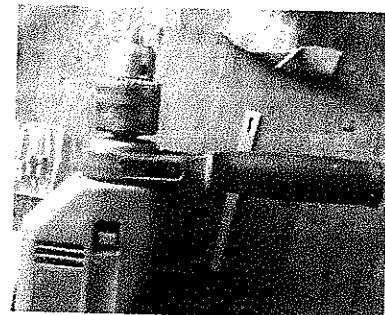
Se puede cortar transversalmente la aguja hipodérmica a medida que se desgaste, La base para taladro mejora la calidad de los orificios, solo requiere una tabla y una varilla.



**Ilustración 70** Base para taladro casera Fuente: El autor

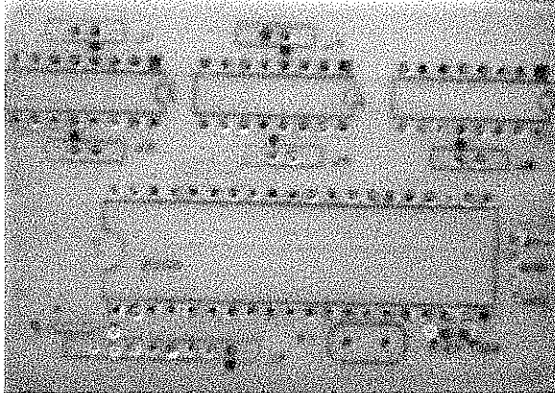


**Ilustración 71** Base para taladro Fuente: El autor



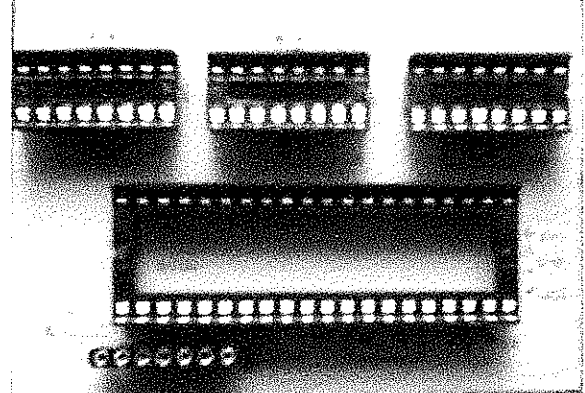
**Ilustración 72** Mango de fuerza del taladro Fuente: El autor

Una vez se tiene lista la placa Ilustración 73 se procede a instalar los sockets o bases para integrado Ilustración 74 el resultado final se puede apreciar en la Ilustración 75.



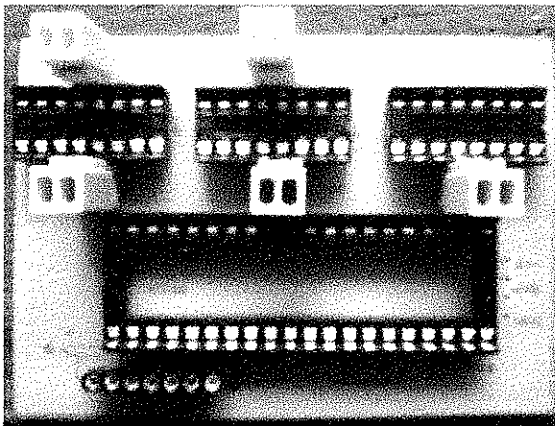
**Ilustración 73 Baquela perforada**

**Fuente: El autor**



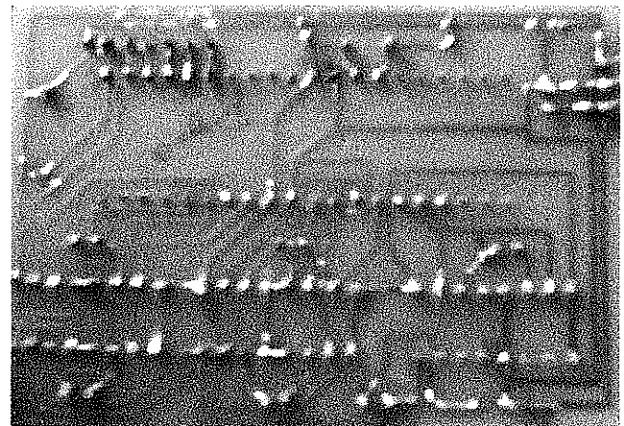
**Ilustración 74 Socket (Porta integrado)**

**Fuente: El autor**



**Ilustración 75 Sockets y conectores**

**Fuente: El autor**



**Ilustración 76 Puntos de soldadura baquela**

**Fuente: El autor**

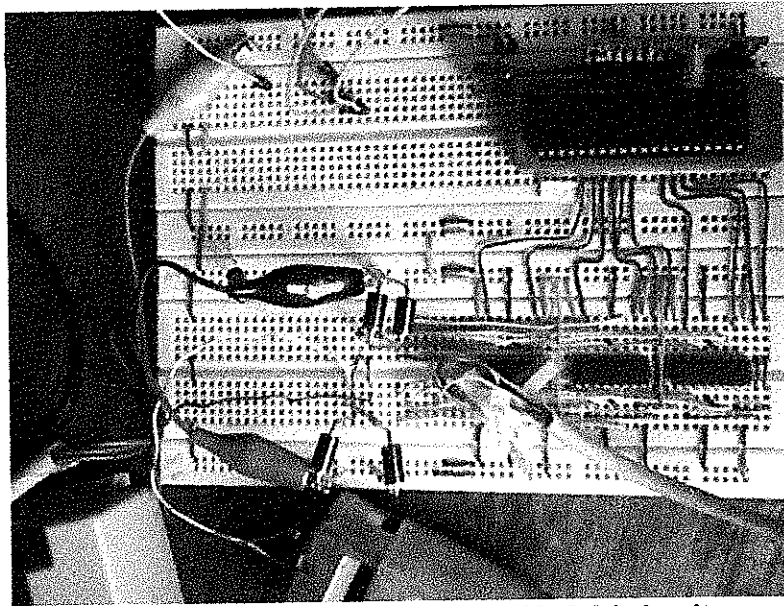
La Ilustración 78 corresponde a la tarjeta final, esta tarjeta funciona como interface entre el módulo inalámbrico y los 5 motores del servomecanismo.

#### **7.5.2.4 Pruebas**

Generalmente después de la prueba en *protoboard* (Ilustración 77) y simulación no es necesario realizar modificaciones en el diseño, sin embargo las pistas fueron creadas con el estilo T20 una dimensión recomendada para diseños profesionales, pero con el procedimiento que se utilizó es necesario hacer los caminos mucho más gruesos.

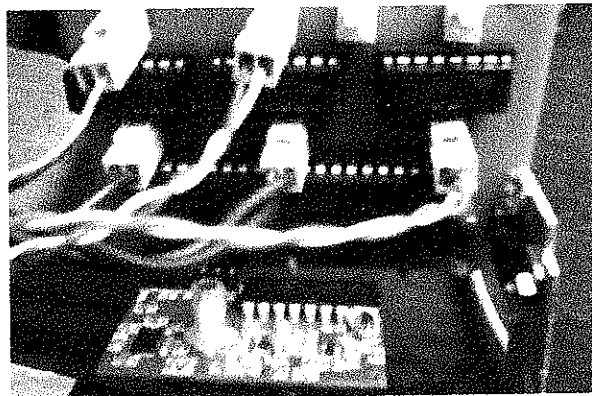
La tarjeta funcional (Ilustración 78 Producto final), cuenta con la ventaja que el microcontrolador está sobre un socket esto permite hacer el cambio de la lógica del programa para el control de los motores en cualquier momento.

Con este tamaño de tarjeta es posible controlar 6 motores con la misma distribución de espacio.



**Ilustración 77 Pruebas de funcionalidad del circuito**  
Fuente: El autor

En este caso se decide tener un solo sistema de alimentación, y utilizar el regulador L7805, se puede remplazar si se quiere tener un sistema de alimentación independiente para la etapa de control sin alterar en gran medida el lugar en la tarjeta.



**Ilustración 78 Producto final**  
Fuente: El autor

## 7.6 INTEGRACIÓN (DESARROLLO DEL PRIMER PROTOTIPO)

Teniendo en cuenta las metodologías anteriormente mencionadas, se hace sinergia con el fin de consolidar el producto de investigación, establecer mejoras y conclusiones de utilidad investigativa en el área de la tele operación apoyada en mundos virtuales inmersivos teniendo en cuenta la metodología de desarrollo basada en componentes.

### 7.6.1 Desarrollo

Para conseguir la integración es necesario agotar las siguientes sub integraciones<sup>25</sup>.

- ✦ Robot móvil y radio enlace<sup>26</sup>
- ✦ Arduino junto con Phyton<sup>27</sup>
- ✦ OpenCv y Python
- ✦ Librería serial y Phyton
- ✦ Librería serial y Blender
- ✦ Mundo virtual y puerto serial
- ✦ Mundo virtual y OpenCV
- ✦ Mundo virtual con animación
- ✦ Mundo virtual OpenCV y librería serial<sup>28</sup>

### 7.6.2 Pruebas

Blender tiene un motor especializado para videojuegos, un motor de juego es un sistema diseñado para la creación y desarrollo de los videojuegos. La funcionalidad básica proporcionada típicamente por un motor de juego incluye un motor de renderizado ("render") para gráficos en 2D o 3D, un motor para la física y detección de colisiones (la respuesta de choque), el sonido, secuencias de comandos, la animación, la inteligencia artificial, redes, transmisión de la memoria, gestión, roscado, soporte de localización, y un escenario gráfico.

Eso en conjunto significa que es posible administrar los procesos para realizar tareas a la par, entonces es cómodo capturar la imagen procesarla, enviar los datos al juego y al servomecanismo y renderizar de manera que parezca que todo se hace al tiempo.

---

<sup>25</sup> Robot movil con arduino uno y teclado PC. [En línea] [Citado el: 29 de 02 de 2012.] <http://www.youtube.com/watch?v=yiMd-WE4mGY&feature=plcp&context=C31394a2UDOEgstoPDskJOUxMKd7KBUBj15utTEBp4>

<sup>26</sup> Robot casero con control inalámbrico. [En línea] [Citado el: 29 de 02 de 2012.] <http://www.youtube.com/watch?v=gJpC1ANV2wM&feature=related>

<sup>27</sup> Robot movil inalámbrico con arduino uno y teclado PC. [En línea] [Citado el: 29 de 02 de 2012.] <http://www.youtube.com/watch?v=ZKED7AXuipw&feature=related>

<sup>28</sup> Canal de DiegoIvanOliveros. [En línea] [Citado el: 29 de 02 de 2012.] <http://www.youtube.com/user/DiegoIvanOliveros?feature=mhee>

## 8. CONCLUSIONES:

Los mundos virtuales generados a partir de planos reales sirven como mecanismo para emular la realidad, ya que funcionan como herramienta de monitoreo brindando más información que la lograda por una cámara estática.

Las interfaces gestuales son un mecanismo para hacer que el usuario se sienta inmerso en una realidad virtual, pues los movimientos o acciones naturales son interpretados por el computador mediante este recurso.

El control de un servomecanismo de forma remota, no solo sirve para materializar las acciones del robot virtual, si no que extiende la capacidad de interacción del ser humano de manera proporcional al alcance y calidad del radio enlace, en la medida en que el servomecanismo emule funciones humanas se considera un sistema de tele - presencia.

El diseño metodológico y las herramientas utilizadas para la creación de sistemas inmersivos tele operados, son útiles y funcionales. Mediante el desarrollo del proyecto se evidencia la aplicabilidad del proceso, lo que hace que este prototipo sea una instrumento base para la construcción de sistemas inmersivos tele operados.

Los aportes más significativos del proyecto son: el estudio y prueba de la herramienta que hizo posible la integración de las tecnologías mediante la investigación y uso de metodologías específicas, la utilización del API para comunicar el robot con el mundo virtual y los algoritmos desarrollados para la interface gestual.

El proceso de investigación solo puede ser concebido siempre que sea continuo, el avance que se logra mediante la experimentación debe ser tenido en cuenta y trabajado de manera permanente, de no ser así los esfuerzos apuntan en direcciones contrarias por lo que su evolución será contenida.

Las organizaciones exitosas son las que innovan, y lo hacen porque investigan de manera interdisciplinar, no se conforman con contemplar y usar tecnología sino de entenderla y generar nuevas alternativas; el trabajo con el software libre permite el estudio meticuloso sin barreras y de esta manera posibilita la innovación. Las herramientas utilizadas para la elaboración de este prototipo son libres y fueron probadas sobre sistemas privativos con arquitectura de 64 bits, de esta manera representan una plataforma cómoda para la migración al software libre.

## 9. RECOMENDACIONES Y TRABAJO FUTURO

Los grupos de investigación de las dos facultades puedan fortalecerse en el desarrollo de proyectos interdisciplinarios, tomando como referente esta investigación. Este proyecto reúne temas estudiados en las dos disciplinas como lo son tratamiento de imágenes, comunicaciones y algoritmos utilizados en la robótica, de esta manera es un ejemplo para la elaboración de proyectos interdisciplinarios.

El trabajo individual en proyectos interdisciplinarios presenta desventajas, es importante que se realicen de manera grupal. Las ventajas que presentan el compañerismo y el trabajo en equipo frente al individual, tanto para las personas involucradas como para las organizaciones son innumerables; se trabaja con menos tensión en proyectos complejos, se comparte la responsabilidad frente a soluciones evaluadas desde diferentes puntos de vista, es más gratificante, las decisiones que se toman tienen mayor aceptación, se dispone de más información, se logra una mayor integración entre las personas para poder conocer las aptitudes de los integrantes.

Es vital fortalecer los laboratorios, mejorar los espacios y dotar de medios necesarios para realizar investigaciones interdisciplinarias. Los experimentos, prácticas y trabajos de carácter científico, tecnológico o técnico; requieren además del interés por el área, contar con instrumentos de medida o equipos para la realización de experimentos, investigaciones y prácticas diversas, especialmente en el área de realidad virtual.

Algunas herramientas se hacen obsoletas con el tiempo, por eso es más importante el desarrollo de competencias que hacerse experto en el uso de ellas, sin embargo, no puedo desconocer la bondades que tiene Python y por eso recomiendo su utilización y enseñanza. Este lenguaje de programación de alto nivel y multiparadigma además de las ventajas de licencia de código abierto es una herramienta clave para el desarrollo de proyectos en líneas de investigación interdisciplinaria.

## 10. BIBLIOGRAFÍA

- [1] RAE, «Real Academia Española,» Juan Manuel Fernández Pacheco, 1713. [En línea]. Available: <http://www.rae.es/rae.html>. [Último acceso: 18 09 2011].
- [2] I. H. Garcia, «ESTÉTICA, CIENCIA Y TECNOLOGÍA,» de *ESTÉTICA, CIENCIA Y TECNOLOGÍA Creaciones electrónicas y numéricas*, Bogota, Pontificia Universidad Javeriana, 2005.
- [3] «Tratamiento digital de imágenes,» [En línea]. Available: <http://www.grimaldos.es/cursos/imgdig/tipos.html>. [Último acceso: 26 05 2010].
- [4] Intel communities, «Disaster Response Communications La comunicación durante los desastres,» 30 11 2009. [En línea]. Available: <http://communities.intel.com/community/latinamerica/scope/blog/2009/11/30/la-comunicaci%C3%B3n-durante-los-desastres>. [Último acceso: 04 01 2012].
- [5] J. L. CORDEIRO, «El cuarto terremoto del mundo,» *EL UNIVERSAL*, lunes 21 de marzo de 2011.
- [6] «En vivo,» 17 03 2010. [En línea]. Available: [http://actualidad.rt.com/ciencia\\_y\\_tecnica/electronica\\_tecnologia/issue\\_21767.html](http://actualidad.rt.com/ciencia_y_tecnica/electronica_tecnologia/issue_21767.html). [Último acceso: 21 03 2010].
- [7] «scienti,» [En línea]. Available: <http://www.colciencias.gov.co/scienti>.
- [8] «En vivo,» 11 03 2010. [En línea]. Available: [http://actualidad.rt.com/ciencia\\_y\\_tecnica/electronica\\_tecnologia/issue\\_21589.html](http://actualidad.rt.com/ciencia_y_tecnica/electronica_tecnologia/issue_21589.html). [Último acceso: 21 03 2010].
- [9] R. S. Pressman, *Ingeniería de software Un enfoque práctico*, Sexta Edición ed., Mexico: Mc Graw Hill, 2008.
- [10] BIGFR, «Biblioteca Instituto Química Física "ROCASOLANO" CSIC,» 07 06 2011. [En línea]. Available: <http://biqfr.blogspot.com/2011/06/telepresencia-3d-en-tiempo-real-con.html>. [Último acceso: 30 06 2011].
- [11] T. Mazuryk y M. Gervautz, «History, Applications, Technology and Future».
- [12] Publicado por Pontificia Universidad Católica de Chile (Escuela de Ingeniería), «Ingeniería de Construcción,» vol. Vol. 17, n° N.º 2, pp. 64-73, 2002.
- [13] vpsarev, «OpenCV,» 24 08 2011. [En línea]. Available: <http://opencv.willowgarage.com/wiki/>. [Último acceso: 08 11 2011].
- [14] «XBOX®,» [En línea]. Available: <http://www.xbox.com/es-es/kinect>. [Último acceso: 21 02 2012].
- [15] G. Bradski and A. kaebler, *OpenCV Computer vision with the openCV library*, Sebastopol: O'Reilly Media Ing, 2008.
- [16] L. Joyanes Aguilar, *Fundamentos de programación algoritmos, estructura de datos y objetos*, McGraw-Hill, 2003.
- [17] «Librería para el manejo y configuración de dispositivos de realidad virtual».
- [18] D. d. M. Benito, «DETECCIÓN AUTOMÁTICA DEL COLOR DE LA PIEL EN

- IMÁGENES BIDIMENSIONALES BASADO EN EL ANÁLISIS DE REGIONES,» 2005. [En línea]. Available: <http://www.escet.urjc.es/~jjpantrigo/PFCs/MemoriaPielFeb05.pdf>. [Último acceso: 27 06 2010].
- [19] «ARToolKit,» [En línea]. Available: <http://www.hitl.washington.edu/artoolkit/>. [Último acceso: 12 01 2012].
- [20] Blender Foundation, «blender,» [En línea]. Available: <http://www.blender.org/>. [Último acceso: 2012].
- [21] Python Software Foundation , «Python Programming Language – Official Website,» Tim Parkin, 1990-2011. [En línea]. Available: <http://python.org/>. [Último acceso: 12 12 2011].
- [22] Blender Foundation, «Welcome to Blender Wiki,» [En línea]. Available: [http://wiki.blender.org/index.php/Main\\_Page](http://wiki.blender.org/index.php/Main_Page). [Último acceso: 22 02 2012].
- [23] «COMO MANEJAR EL PUERTO USB,» [En línea]. Available: <http://www.cotecom.net/category/informacion-relevante/>.
- [24] Jairo, «<http://www.tecnolives.com/un-robot-que-utiliza-mundos-virtuales/>,» Septiembre 2009. [En línea].
- [25] «alt1040,» Pepe Flores , 18, 23:02 Febrero 2010. [En línea]. Available: <http://alt1040.com/2010/02/reconocimiento-gestual-cuando-la-tv-nos-vea-a-nosotros>. [Último acceso: 30 05 2010].
- [26] «cisco-telepresence-1100-sistema-de-telepresencia-en-alta-definicion,» 14 09 2009. [En línea]. Available: <http://www.tuexpertoit.com>. [Último acceso: 2010 06 27].
- [27] E. O. D. D. Uzcátegui, «METODOLOGÍAS DE DESARROLLO PARA SISTEMAS DE TIEMPO REAL. UN ESTUDIO COMPARATIVO,» *Universidad, Ciencia y Tecnología*, 2009.
- [28] C. E. Bruce Powel Douglass, «Rapid Object-Oriented Process for Embedded Systems,» *I-Logix*, 1999.
- [29] K. Beck, *Extreme Programming Explained. Embrace Change*, Pearson Education, 1999.
- [30] P. Oppenheimer, *Top-Down Network Design Third Edition*, Indianapolis, USA: Cisco Press, 2011.
- [31] b. P. Oppenheimer, «Top-Down Network Design,» Open Door Networks., [En línea]. Available: <http://www.topdownbook.com/>. [Último acceso: 23 02 2012].
- [32] M. Pesce, *VRML para Internet*, México: Prentice Hall, 1996.
- [33] J. E. Muñoz Labra, «METODOLOGÍA DE DESARROLLO VRML,» [En línea]. Available: <http://www.jose-emilio.com/estudios/m1metodologia.htm>. [Último acceso: 2012 02 25].
- [34] A. Crespo Márquez, P. Moreu de León y A. Sánchez Herquedas, *Ingeniería de mantenimiento Técnicas y metodos de aplicacion a la fase operativa de los equipos*, España: AENOR, 2004.
- [35] N. Cross, *Metodos de Diseño estrategias para el diseño de productos*, Mexico, D.F.:

LIMUSA, S.A, 2005.

- [36] SAE International , «SAE International,» [En línea]. Available: <http://www.sae.org/>. [Último acceso: 26 02 2012].
- [37] J. J. Esqueda Elizondo y L. E. Palafox Maestre, Fundamentos para el procesamiento de imágenes, Universidad Autónoma de baja California.
- [38] R. C. González y R. E. Woods, Tratamiento digital de imágenes, Addison wesley/ Diaz de santos.
- [39] E. A. Gutierrez Caceres, Procesamiento digital de imagenes con Matlab (R) Utilizando una web cam y comunicación serial para control de dispositivo movil, Tunja, 2010.
- [40] D. G. Pérez. [En línea]. Available: <http://gva1.dec.usc.es/~antonio/docencia/2005tci/practica/apuntes-practica-5.pdf>. [Último acceso: 27 06 2010].
- [41] [En línea]. Available: <http://interartive.org/index.php/2008/11/telepresencia/>.
- [42] M. S. Gast, 802.11 Wireless Networks. 2nd Edition. The Definitive Guide, O'Reilly Networking, 2005, p. 768.
- [43] L. Santamaría Granados y J. F. Mendoza Moreno, «Herramientas en 3D para el modelado de escenarios virtuales basados en Logo, estado del arte,» *Ciencia e Ingeniería Neogranadina*, vol. 19, n° 2, pp. 77-94, 2009.
- [44] Web3D CONSORTIUM, «Extensible 3D (X3D) Part 1: Architecture and base components,» Web3D, 2008.
- [45] T. Akiyama, Dirección, *Hinokio*. [Película]. Shuuji Uchiyama, Hidekazu Uehara y Takuji Ushiyama, 2005.

## 11. ANEXOS

## Anexo A. Metodología RCM

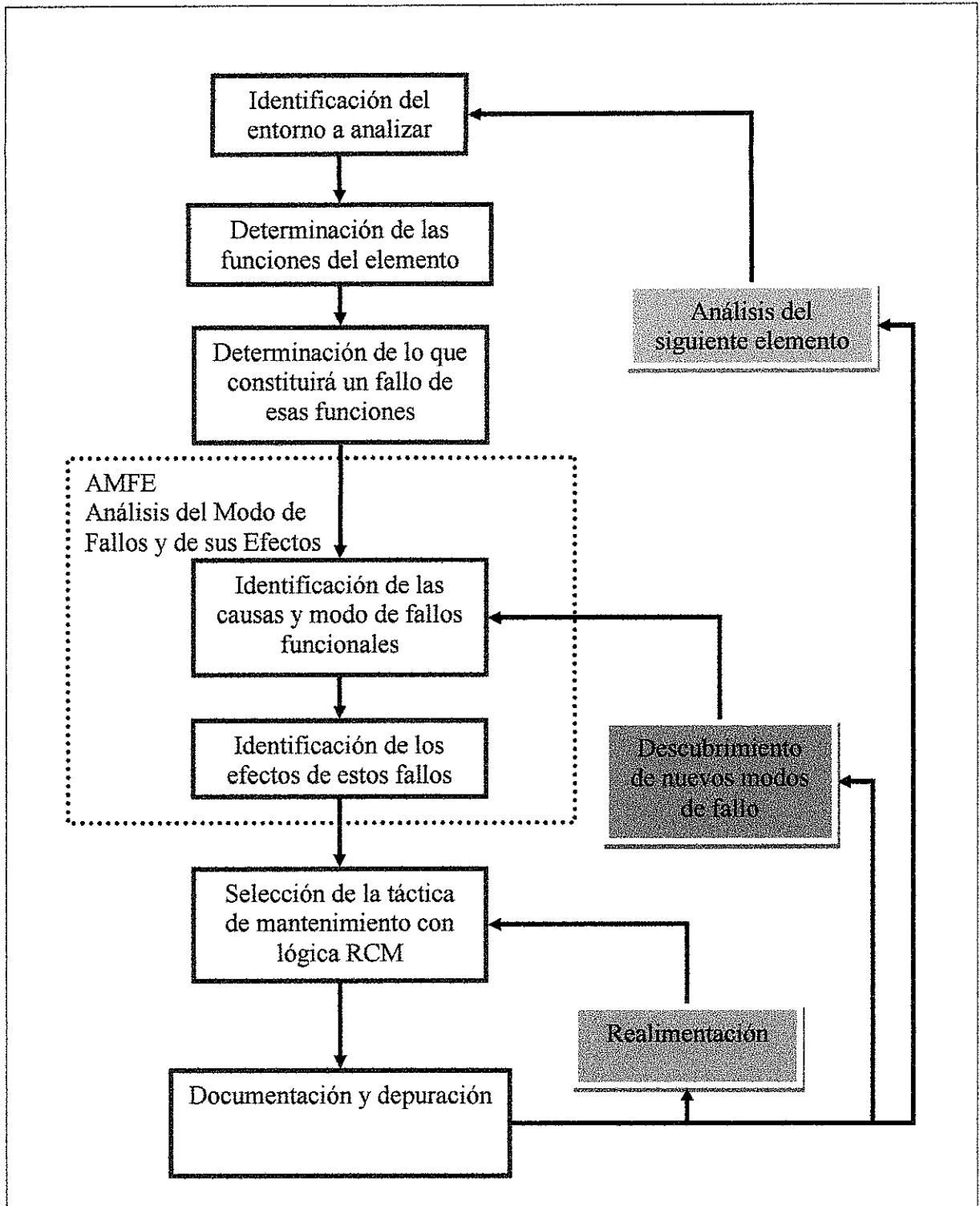


Figura 1. Descripción general método RCM  
Fuente: El autor basado en:[34]

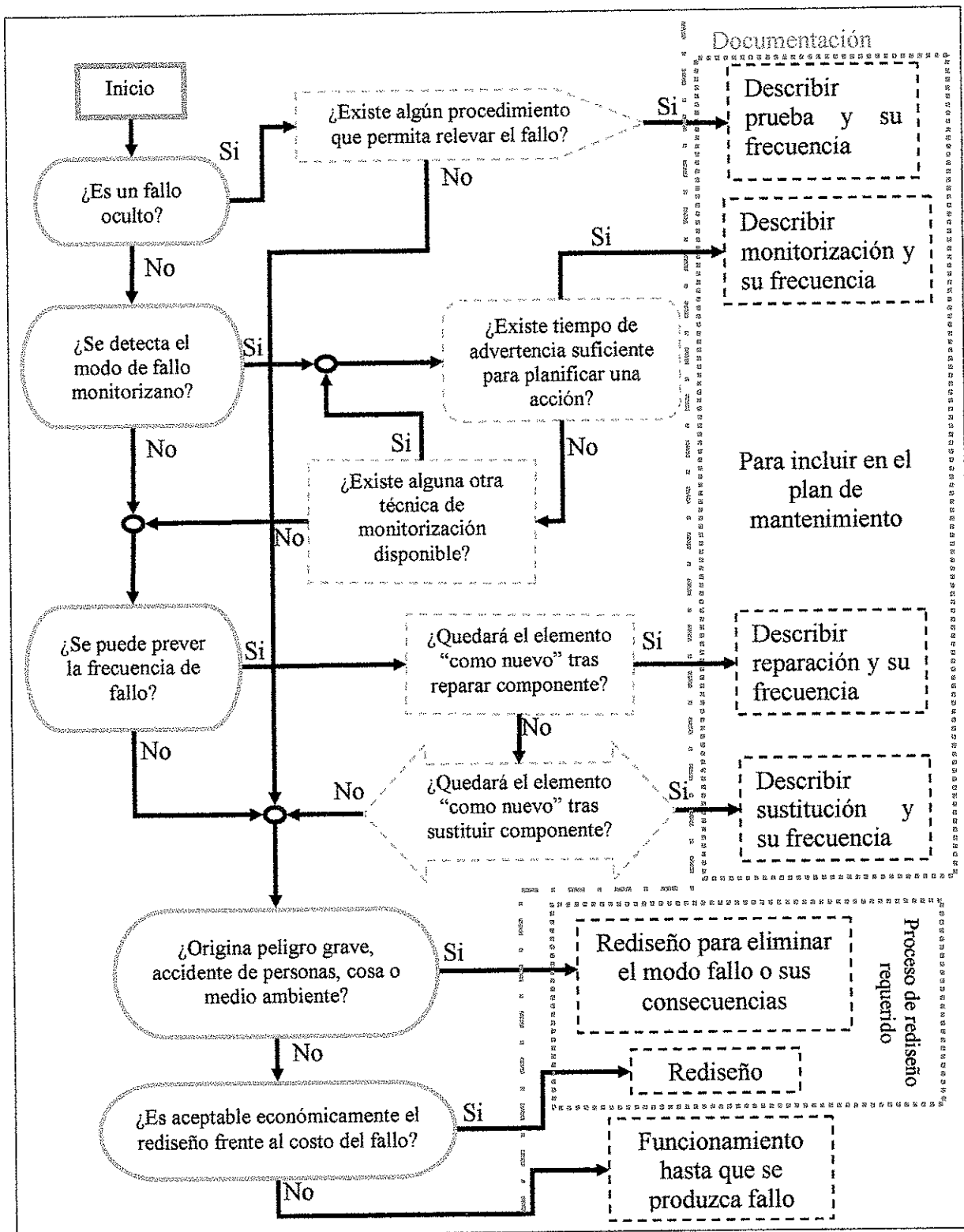


Figura 2. Descripción de la lógica método RCM

Fuente: El autor basado en:[34]

## Anexo B. Análisis de riesgos

En la ilustración 1 se encuentra el formato de *check list* de manejo de riesgos que se aplica para definir las alternativas de solución de acuerdo a los riesgos identificados en el proyecto[29].

CHECKLIST DE MANEJO DE RIESGOS

No	Pregunta	Sí / No	Próxima fecha	Comentario
1	¿xxx?	Check	AAAA/MM/DD	
		<input type="checkbox"/>		
No	Pregunta	Sí / No	Próxima fecha	Comentario

Figura 3 Checklist

Fuente: el autor

Los diseños que se utilizan para el reporte de riesgos se encuentran en las ilustraciones (2)

Reporte de Riesgos	
<b>Nombre del proyecto:</b> Accionamiento Real A Distancia De Un Servomecanismo Mediante Un Mundo Virtual Controlado Por Interfaz Gestual	
Fecha: 22 Octubre 2011	Estado Riesgo : Activo
Riesgo Numero : 1	Probabilidad : 40-50%
Nombre Riesgo : Interface OpenCvBlender	Impacto en costos: \$1'000.000
Área Funcional: Captura de movimiento	Impacto en tiempo: Semana
Responsable: Diego Iván Oliveros Acosta	Impacto desempeño: Sin cumplimiento
Meta Crítica : Revisión de integración	WBS Ref : Especific. Sist-1
<b>Enfoque de evasión de Riesgo :</b> Hay dos posibilidades de evasión 1) Utilizar tuberías 2) Usar archivos de texto para almacenar las variables de comunicación (implicaría que el proceso no se hiciera en tiempo real)	
<b>Estado del Plan de Contingencia al Riesgo :</b> Este plan de contingencia no se ha implementado y su implementación ocurriría si faltando dos semanas para la meta de revisión de integración no se ha resuelto el riesgo actual	
<b>Estado Global del Riesgo :</b> Las actividades de desarrollo de interface reducidas deben eludir el riesgo. Los planes de contingencia están firmes y pueden llevarse a cabo a corto Tiempo.	

Figura 4 Reporte de riesgos

Fuente: El autor

### CHECKLIST DE MANEJO DE RIESGOS

No	Pregunta	Si / No	Próxima fecha	Comentario
1	¿Se identificaron y evaluaron riesgos durante la fase de captura y arranque definiendo impactos en costos, ingresos, cronograma y desempeño técnico así como también sus probabilidades?			
		Check	AAAA/MM/DD	
		<input type="checkbox"/>		
No	Pregunta	Si / No	Próxima fecha	Comentario
2	¿Se desarrolló la red de tareas para evaluar problemas de riesgos e impacto en cronograma?			
		Check	AAAA/MM/DD	
		<input type="checkbox"/>		
No	Pregunta	Si / No	Próxima fecha	Comentario
3	¿Se establecieron planes de evasión y mitigación para cada Riesgo identificado (como evitar o reducir costos de impacto sobre riesgos)?			
		Check	AAAA/MM/DD	
		<input type="checkbox"/>		
No	Pregunta	Si / No	Próxima fecha	Comentario
4	¿Se definieron planes de contingencia para los riesgos seleccionados dirigidos por la Gerencia (que hacer si ocurre el riesgo)?			
		Check	AAAA/MM/DD	
		<input type="checkbox"/>		
No	Pregunta	Si / No	Próxima fecha	Comentario
5	¿Se definió presupuesto de costos para contingencia en riesgos identificados?			
		Check	AAAA/MM/DD	
		<input type="checkbox"/>		

No	Pregunta	Si / No	Próxima fecha	Comentario
6	¿La asignación de presupuesto de contingencia está directamente relacionada con los planes de mitigación y/o evasión de riesgos?			
		Check	AAAA/MM/DD	
		<input type="checkbox"/>		
No	Pregunta	Si / No	Próxima fecha	Comentario
7	¿La información de los riesgos es monitoreada para determinar el tiempo apropiado de implementación de los planes de contingencia?			
		Check	AAAA/MM/DD	
		<input type="checkbox"/>		
No	Pregunta	Si / No	Próxima fecha	Comentario
8	¿El conjunto de riesgos es reevaluado al menos quincenalmente para agregar nuevos riesgos y afinar los impactos en forma oportuna?			
		Check	AAAA/MM/DD	
		<input checked="" type="checkbox"/>		
No	Pregunta	Si / No	Próxima fecha	Comentario
9	¿La WBS* se usó como base de trabajo en la identificación de riesgos?			
		Check	AAAA/MM/DD	
		<input checked="" type="checkbox"/>		

\*La WBS o Estructura desglosada del Trabajo, es una técnica de planeación mediante la cual podemos definir y cuantificar el trabajo a realizar en todo el proyecto.

## Reporte de Riesgos.

**Nombre del proyecto:**

Accionamiento Real A Distancia De Un Servomecanismo Mediante Un Mundo Virtual Controlado Por Interfaz Gestual

**Fecha :** 22 Octubre 2011

**Estado Riesgo :** Activo

**Riesgo Numero :** 1

**Probabilidad :** 40-50%

**Nombre Riesgo :** Interface OpenCvBlender

**Impacto en costos:** \$1'000.000

**Área Funcional:** Captura de movimiento

**Impacto en tiempo:** Semana

**Responsable:** Diego Iván Oliveros Acosta

**Impacto desempeño :** Sin cumplimiento

**Meta Crítica :** Revisión de integración

**WBS Ref :** Especific. Sist-1

**Enfoque de evasión de Riesgo :**

Hay dos posibilidades de evasión

1) Utilizar tuberías

2) Usar archivos de texto para la almacenar las variables de comunicación (implicaría que el proceso no se hiciera en tiempo real)

**Estado del Plan de Contingencia al Riesgo :**

Este plan de contingencia no se ha implementado y su implementación ocurriría si faltando dos semanas para la meta de Revisión de integración no se ha resuelto el riesgo actual

**Estado Global del Riesgo :**

Las actividades de desarrollo de interface reducidas deben eludir el riesgo.  
Los planes de contingencia están firmes y pueden llevarse a cabo a corto Tiempo.

## Reporte de Riesgos

<b>Nombre del proyecto:</b>	
Accionamiento Real A Distancia De Un Servomecanismo Mediante Un Mundo Virtual Controlado Por Interfaz Gestual	
<b>Fecha :</b> 22 Octubre 2011 <b>Riesgo Numero :</b> 2 <b>Nombre Riesgo :</b> Interface Blender-Puerto USB <b>Área Funcional:</b> Captura de movimiento <b>Responsable:</b> Diego Iván Oliveros Acosta <b>Meta Crítica :</b> Revisión de integración	<b>Estado Riesgo :</b> Activo <b>Probabilidad :</b> 40-50% <b>Impacto en costos:</b> \$100.000 <b>Impacto en tiempo:</b> Semana <b>Impacto desempeño :</b> Sin cumplimiento <b>WBS Ref :</b> Especific. Sist-2
<b>Enfoque de evasión de Riesgo :</b>	
La posibilidad de evasión es utilizar puerto paralelo o DB9	
<b>Estado del Plan de Contingencia al Riesgo :</b>	
Utilizar la librería de Artoolkit y el soporte del proveedor Los costos adicionales serían los materiales necesarios para la construcción y el tiempo necesario para el cambio de máquina. Este plan de contingencia no se ha implementado y su implementación ocurriría si faltando tres semanas para la meta de Revisión de integración no se ha resuelto el riesgo actual	
<b>Estado Global del Riesgo :</b>	
Las actividades de desarrollo de interface reducidas deben eludir el riesgo. Los planes de contingencia están firmes y pueden llevarse a cabo a corto Tiempo.	

## Reporte de Riesgos

<b>Nombre del proyecto:</b> Accionamiento Real A Distancia De Un Servomecanismo Mediante Un Mundo Virtual Controlado Por Interfaz Gestual	
<b>Fecha :</b> 22 Octubre 2011 <b>Riesgo Numero :</b> 3 <b>Nombre Riesgo:</b> Fallas mecánicas del dispositivo. <b>Área Funcional:</b> Captura de movimiento <b>Responsable:</b> Diego Iván Oliveros Acosta <b>Meta Crítica :</b> Revisión de integración	<b>Estado Riesgo :</b> Activo <b>Probabilidad :</b> 40-50% <b>Impacto en costos:</b> sujeto al tipo de fallo <b>Impacto en tiempo:</b> Semana <b>Impacto desempeño :</b> Sin cumplimiento <b>WBS Ref:</b> Especific. Sist-3
<b>Enfoque de evasión de Riesgo :</b> La posibilidad de evasión es usar un producto del mercado que tenga la misma cantidad de servos.	
<b>Estado del Plan de Contingencia al Riesgo :</b> El costo adicional seria la compra del nuevo servomecanismo y el costo de desplazamiento a Bogotá. Este plan de contingencia no se ha implementado y su implementación ocurriría si faltando cuatro semanas para la meta de Revisión de integración no se ha resuelto el riesgo actual	
<b>Estado Global del Riesgo :</b> Las actividades de desarrollo de interface reducidas deben eludir el riesgo. Los planes de contingencia están firmes y pueden llevarse a cabo a corto Tiempo.	

## Reporte de Riesgos

**Nombre del proyecto:**

Accionamiento Real A Distancia De Un Servomecanismo Mediante Un Mundo Virtual Controlado Por Interfaz Gestual

**Fecha :** 22 Octubre 2011

**Riesgo Numero :** 4

**Nombre Riesgo:** Interferencia del puente h con el micro controlador.

**Área Funcional:** Captura de movimiento

**Responsable:** Diego Iván Oliveros Acosta

**Meta Crítica :** Revisión de integración

**Estado Riesgo :** Activo

**Probabilidad :** 40-50%

**Impacto en costos:** \$300.000

**Impacto en tiempo:** Semana

**Impacto desempeño :** Sin cumplimiento

**WBS Ref :** Especific. Sist-4

**Enfoque de evasión de Riesgo :**

La posibilidad de evasión es usar un producto del mercado que tenga la misma cantidad de servos.

**Estado del Plan de Contingencia al Riesgo :**

El costo adicional seria la compra del nuevo servomecanismo

Este plan de contingencia no se ha implementado y su

implementación ocurriría si faltando cuatro semanas para la meta de Revisión de integración no se ha resuelto el riesgo actual

**Estado Global del Riesgo :**

Las actividades de desarrollo de interface reducidas deben eludir el riesgo.

Los planes de contingencia están firmes y pueden llevarse a cabo a corto

Tiempo.

<b>Reporte de Riesgos</b>	
<b>Nombre del proyecto:</b>	
Accionamiento Real A Distancia De Un Servomecanismo Mediante Un Mundo Virtual Controlado Por Interfaz Gestual	
<b>Fecha :</b> 22 Octubre 2011 <b>Riesgo Numero :</b> 5 <b>Nombre Riesgo:</b> Falta de Capacidad de procesamiento para el renderizado. <b>Área Funcional:</b> Mundo Virtual <b>Responsable:</b> Diego Iván Oliveros Acosta <b>Meta Crítica :</b> Pruebas finales mundo virtual	<b>Estado Riesgo :</b> Activo <b>Probabilidad :</b> 40-50% <b>Impacto en costos:</b> \$4'000.000 <b>Impacto en tiempo:</b> Semana <b>Impacto desempeño :</b> Sin cumplimiento <b>WBS Ref :</b> Especific. Sist-5
<b>Enfoque de evasión de Riesgo :</b>	
Existen dos posibilidades de evasión la primera es que se use un equipo de mayor capacidad la Segunda que se use una superficie plana o manejar distintos escenarios.	
<b>Estado del Plan de Contingencia al Riesgo :</b>	
El costo adicional seria la compra del nuevo equipo con las prestaciones necesarias para hacer posible el renderizado. Este plan de contingencia no se ha implementado y su implementación ocurriría si faltando una semana para la meta de Revisión del mundo virtual no se ha resuelto el riesgo actual.	
<b>Estado Global del Riesgo :</b>	
Las actividades de puesta en funcionamiento del mundo virtual deben eludir el riesgo. Los planes de contingencia están firmes y pueden llevarse a cabo a corto Tiempo.	

## Reporte de Riesgos

<b>Nombre del proyecto:</b>	
Accionamiento Real A Distancia De Un Servomecanismo Mediante Un Mundo Virtual Controlado Por Interfaz Gestual	
<b>Fecha :</b> 22 Octubre 2011 <b>Riesgo Numero :</b> 6 <b>Nombre Riesgo:</b> Interface Blender(Python) –Arduino . <b>Área Funcional:</b> Mundo Virtual <b>Responsable:</b> Diego Iván Oliveros Acosta <b>Meta Crítica :</b> Revisión de integración	<b>Estado Riesgo :</b> Activo <b>Probabilidad :</b> 40-50% <b>Impacto en costos:</b> \$200.000 <b>Impacto en tiempo:</b> Semana <b>Impacto desempeño :</b> Sin cumplimiento <b>WBS Ref :</b> Especific. Sist-5
<b>Enfoque de evasión de Riesgo :</b>	
Hay dos posibilidades de evasión 1) Utilizar tuberías 2) Usar archivos de texto para la almacenar las variables de comunicación (implicaría que el proceso no se hiciera en tiempo real)	
<b>Estado del Plan de Contingencia al Riesgo :</b>	
El costo adicional seria el plan operativo Este plan de contingencia no se ha implementado y su implementación ocurriría si faltando una semana para la meta de Revisión del mundo virtual no se ha resuelto el riesgo actual.	
<b>Estado Global del Riesgo :</b>	
Las actividades de puesta en funcionamiento de la Interface Blender(Python) –Arduino deben eludir el riesgo. Los planes de contingencia están firmes y pueden llevarse a cabo a corto Tiempo.	

## Anexo C. Análisis de requisitos

En este anexo se realiza el documento de análisis de requisitos funcionales y no funcionales del sistema.

### 1. Propósito del sistema

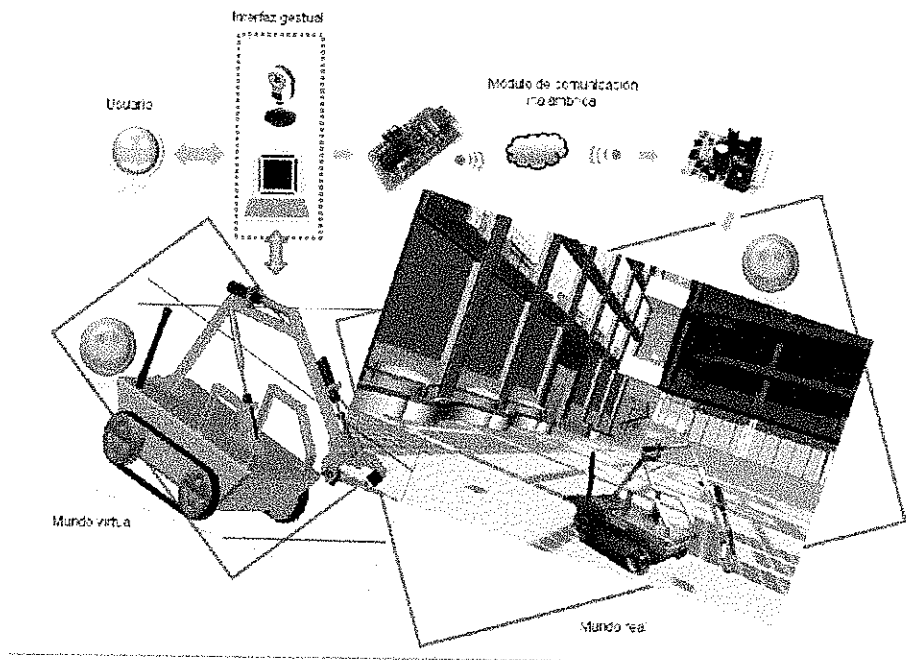
Corresponde a los objetivos del trabajo de grado.

### 2. Alcance del sistema

Se desarrollaran 4 módulos (Interface Gestual, Mundo virtual, Comunicación Inalámbrica y Servomecanismo) que analizados en conjunto servirán como punto de partida para el diseño de sistemas de tele-presencia, permitiendo el control a distancia de servomecanismos. Para eso se utiliza Blender y Scrips de Phyton con el fin de extender sus funcionalidades. La aplicación objetivo es el mundo virtual.

### 3. Descripción del sistema

El sistema mecánico está fuera de servicio, sin un circuito de control o sistema eléctrico, no existe una aplicación objetivo ni mundo virtual, no se cuenta con un software de reconocimiento de imagen o interface gestual y es necesario contar con un mecanismo que permítala comunicación de los módulos que componen el sistema de tele-presencia o tele operación.



**Figura 5 Descripción del sistema**

Fuente el autor

#### 4. Usuarios del sistema

Diseñador: Es quien realiza la aplicación objetivo, tiene acceso directo a toda la programación

Usuario final: Quien utiliza la aplicación objetivo.

#### 5. Requerimientos funcionales

<b>Nombre</b>	<b>R1- Selección del objeto</b>
<b>Depende de</b>	<b>Ninguno</b>
<b>Resumen</b>	<b>El usuario debe seleccionar un área dentro de la imagen capturada por la cámara a la que se le realizará el seguimiento</b>
<b>Entradas</b>	<b>Coordenadas inicial y final dentro del mapa de bits (X1,Y1)(X2,Y2)</b>
<b>Resultados</b>	<b>El sistema identifica el objeto que debe seguir.</b>

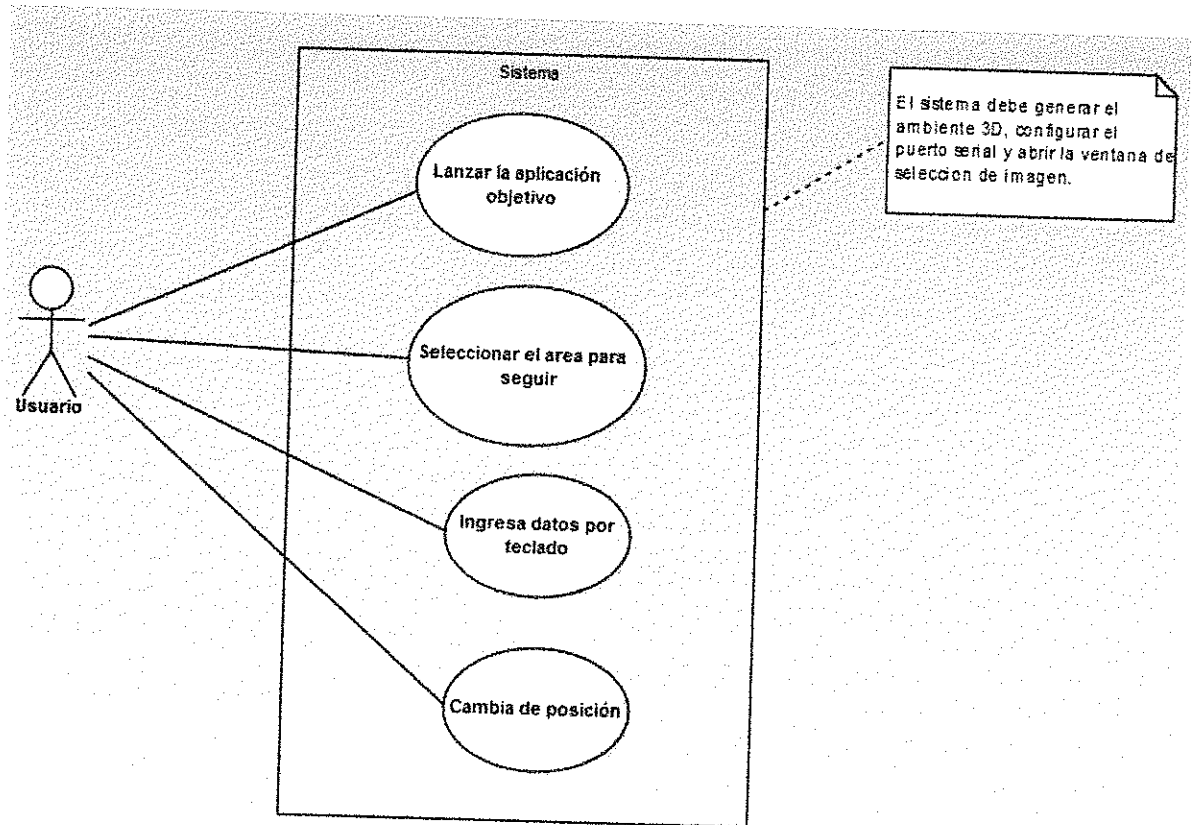
<b>Nombre</b>	<b>R2- Generar escena en 3D</b>
<b>Depende de</b>	<b>R1</b>
<b>Resumen</b>	<b>Una vez se determina la posición del usuario en la imagen se construye la escena virtual.</b>
<b>Entradas</b>	<b>Coordenadas del usuario en la imagen</b>
<b>Resultados</b>	<b>Presentación del mundo virtual</b>

#### 6.1 Casos de uso

En la figura se muestra el esquema de un diagrama de casos de uso para ingreso al sistema.

### 6.1.1 Descripción de los Casos de uso

Como se muestra en la figura, la aplicación objetivo define los siguientes casos de uso:



**Figura 1. Diagrama de casos de uso administración de usuarios**  
Fuente: El autor

1. **Lanzar la aplicación objetivo:** Implica que el programa debe rederizar el mundo virtual, mostrar el video de las imágenes capturadas por la cámara y un histograma de colores correspondiente a una zona del video.
2. **Seleccionar el área para seguir:** El usuario selecciona una zona dentro del video en donde se encuentre el objeto a seguir. Hace click sobre el borde superior izquierdo de la zona a seguir y arrastra el mouse hasta el borde inferior de la misma.
3. **Ingresa datos por teclado:** Las instrucciones; Adelante (w), atrás (s), derecha (d), izquierda (a), pinza abre (o), pinza cierra (l), codo sube (i), codo baja (k), base sube (u) y base baja (k), son ingresadas por el usuario para controlar la posición y acciones del servomecanismo tanto en el mundo virtual, como en el real.
4. **Cambio de posición:** la interface gestual deberá identificar cuando existe un cambio de posición del área seleccionada.

<b>Identificador: 1</b>	<b>Nombre descriptivo: Lanzar la aplicación objetivo</b>	
<b>Descripción</b>	El sistema deberá permitir a el actor al comienzo de la aplicación visualizar el mundo virtual según se describe en el siguiente caso de uso:	
<b>Precondición</b>	Ninguna	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Instanciar los el mundo virtual, las escenas y el avatar
<b>Postcondición</b>	Ninguna	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Ninguna
<b>Rendimiento</b>	Para que el sistema realice las acciones descritas en los pasos anteriores no tiene restricción de tiempo.	
<b>Frecuencia</b>	Este caso de uso se espera que se lleve a cabo una sola vez	
<b>Importancia</b>	Vital	
<b>Urgencia</b>	inmediatamente	
<b>Comentarios</b>	Ninguno	

<b>Identificador: 2</b>	<b>Nombre descriptivo: Seleccionar el área para seguir</b>	
<b>Descripción</b>	El sistema deberá permitir a [lista actores] en [instante en el que se puede realizar el caso de uso] [funcionalidad que define el caso de uso] según se describe en el siguiente caso de uso:	
<b>Precondición</b>	Caso # 1. Y debe existir una ventana con el video objetivo.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario selecciona una zona dentro del video en donde se encuentre el objeto a seguir.
	2	Hace click sobre el borde superior izquierdo de la zona a seguir y arrastra el mouse hasta el borde inferior de la misma.
		2a Si la zona tiene un valor positivo el sistema deberá guardar la información de la zona.
	2b Si es nulo el sistema deberá esperar	

<b>Postcondición</b>	Zona seleccionada.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	En el caso de que el usuario no seleccione un área o zona el sistema deberá esperar
<b>Rendimiento</b>	El usuario no debe percibir que esta tarea implica tiempo de procesamiento	
<b>Frecuencia</b>	Este caso de uso se espera que se lleve a cabo todo el tiempo	
<b>Importancia</b>	Importante	
<b>Urgencia</b>	puede esperar	
<b>Comentarios</b>	Ninguno	

<b>Identificador: 3</b>	<b>Nombre descriptivo: Ingresar datos por teclado</b>	
<b>Descripción</b>	El sistema deberá permitir a el actor en cualquier momento ingresar datos de control según se describe en el siguiente caso de uso:	
<b>Precondición</b>	Caso 1	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Enviar el dato al puerto
	2	Modificar las propiedades del avatar
		2 <sup>a</sup> Si la letra es w mover hacia adelante
		2b Si la letra es s mover hacia atrás
		2c Si la letra es a mover a izquierda
	2d Si la letra es d mover a derecha	
<b>Postcondición</b>	Cambio de posición del avatar, dato en el puerto.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1 y 2	En el caso de que el usuario digite esc el sistema deberá terminar el programa.
<b>Rendimiento</b>	El usuario no debe percibir que esta tarea implica tiempo de procesamiento	
<b>Frecuencia</b>	Este caso de uso se espera que se lleve a cabo todo el tiempo	
<b>Importancia</b>	Importante	
<b>Urgencia</b>	Inmediatamente	
<b>Comentarios</b>	Ninguno	

<b>Identificador: 4</b>	<b>Nombre descriptivo: Cambio de posición</b>	
<b>Descripción</b>	El sistema deberá permitir al usuario en todo momento cambiar la posición según se describe en el siguiente caso de uso:	
<b>Precondición</b>	Caso 1 y 2	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Calcular la distribución de probabilidad de encontrar la zona dentro de la imagen.
	2	Determinar la zona de mayor concentración
	3	Calcular el centro de masa de la distribución.
	4	Identificar en que porción de la imagen se encuentra
<b>Postcondición</b>	Posición del objeto	
<b>Excepciones</b>	Ninguna	
<b>Rendimiento</b>	El usuario no debe percibir que esta tarea implica tiempo de procesamiento	
<b>Frecuencia</b>	Este caso de uso se lleve a cabo todo el tiempo	
<b>Importancia</b>	Importante	
<b>Urgencia</b>	Inmediatamente	
<b>Comentarios</b>	Ninguno	

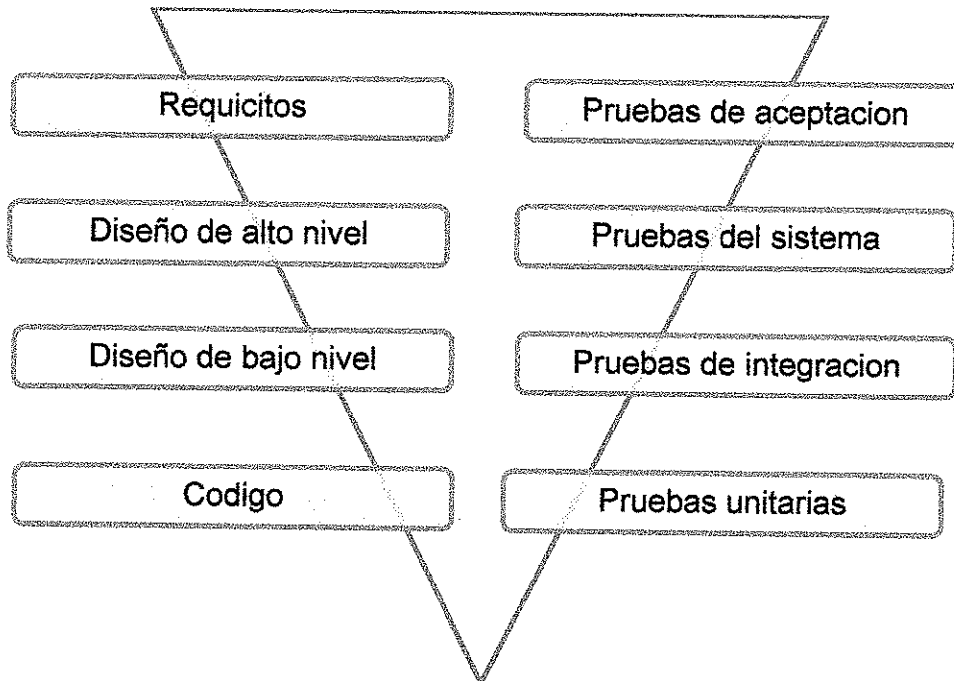
### 7. Requerimientos no funcionales

Especifica las características de HW y SW para que el prototipo sea funcional.

<b>Nombre</b>	<b>R1- Condiciones de la máquina</b>
<b>Depende de</b>	Ninguno
<b>Resumen</b>	Para que la aplicación pueda ser ejecutada se debe usar un computador con 4G de memoria RAM o superior, un procesador de 1.5GHz o superior
<b>Entradas</b>	
<b>Resultados</b>	El usuario ingreso de forma satisfactoria a la aplicación objetivo

## Anexo D. Pruebas

En esta fase del desarrollo de software se verifica cada uno de los productos que se han construido, bajo la premisa que cada parte bien elaborada garantiza el producto final, además la validación comprueba el cumplimiento de los objetivos de los requisitos y del sistema[9].



**Figura 6 modelo en V para pruebas de software**  
Fuete: el autor basado en [9]

### Descripción de pruebas.

Como se ilustra en la figura existen varios tipos de pruebas a lo largo del proceso de construcción de software a continuación se documentan algunas de ellas.

Número	1
Nombre	Interfaz gestual
Tipo	Prueba unitaria
Prioridad	Alta/Esencial x Media/Deseado Baja/Opcional
Descripción	Mediante la ejecución independiente del código phyton en la consola, verificar que se reconozca el objeto seleccionado mediante la visualización de un ovalo rojo que encierre la superficie a pesar que este se desplace.
Resultado	El ovalo sigue la figura seleccionada en el video.

Número	2
Nombre	Avatar
Tipo	Prueba unitaria
Prioridad	Alta/Esencial      Media/Deseado x      Baja/Opcional
Descripción	Evaluar el desplazamiento del avatar en el mundo virtual, paramentoros físicos y respuesta a la animación.
Resultado	Renderizado y física del avatar funcionales, en cuanto a la respuesta a la animación existe dificultad para ejecutar 2 animaciones dentro de una misma escena para un mismo objeto.
Número	3
Nombre	Puerto serial
Tipo	Prueba de unitaria
Prioridad	Alta/Esencial x      Media/Deseado      Baja/Opcional
Descripción	Mediante la ejecución del código Phyton en la consola independiente verificar que se presente el dato de selección en el puerto.
Resultado	Caracteres en el puerto de comunicaciones de acuerdo a los comandos por teclado. Phyton requiere un tiempo de 0.5s para la instanciación del puerto.
Número	4
Nombre	Avatar + Animación + mundo virtual
Tipo	Prueba de integración
Prioridad	Alta/Esencial      Media/Deseado x      Baja/Opcional
Descripción	El desplazamiento del avatar con las animaciones en el mundo virtual.
Resultado	Requiere gran capacidad de cómputo y nueva api.
Número	5
Nombre	Avatar + físicos +mundo virtual + puerto serial
Tipo	Prueba del sistema
Prioridad	Alta/Esencial      Media/Deseado x      Baja/Opcional
Descripción	Utilización de Blender Game engine para juegos e interacción con el puerto de comunicaciones.
Resultado	No presenta inconvenientes ni requiere altos recursos de máquina.

Número	6
Nombre	Avatar + físicos + animación
Tipo	Prueba integración
Prioridad	Alta/Esencial      Media/Deseado x      Baja/Opcional
Descripción	Integrar animaciones y parámetros físicos del juego.
Resultado	Requiere gran capacidad de cómputo y no es compatible entre versiones.
Número	7
Nombre	Avatar + comunicación serial + interfaz gestual + mundo virtual
Tipo	Prueba integración
Prioridad	Alta/Esencial      Media/Deseado x      Baja/Opcional
Descripción	Prueba de sistema
Resultado	Requiere gran capacidad de cómputo y no es compatible entre versiones.
Número	8
Nombre	Inicio de la aplicación
Tipo	Prueba aceptación
Prioridad	Alta/Esencial x      Media/Deseado      Baja/Opcional
Descripción	La aplicación instancia el puerto de comunicaciones, renderiza el mundo virtual, crea una ventana de captura de video y grafica el histograma.
Resultado	El usuario puede empezar a utilizar la aplicación
Número	9
Nombre	Avatar + escenario + interface gestual + comunicación serial
Tipo	Prueba de aceptación
Prioridad	Alta/Esencial x      Media/Deseado      Baja/Opcional
Descripción	El avatar responde a los comandos de teclado e interfaz gestual dentro del mundo virtual
Resultado	Renderizado y animación física del avatar.

### Evaluación de la hipótesis:

#### Prueba de interactividad con el mundo virtual

Existen 3 tipos de terreno, área de superficies extraterrestres (luna), video juegos (terrenos irregulares) y área abierta o multipropósito (superficies planas), para cada una de ellas se realizó la siguiente evaluación:

Objeto o	Descripción	Satisfactoria	Aceptable	Comentarios
Superficies planas	Se realizó una representación virtual de una superficie	X		
Terrenos irregulares	Se diseñó un mundo virtual aplicando texturas de suelo y grama en terreno con caminos protuberancias y depresiones	X		Este tipo de terrenos generan un ambiente más natural pero requiere de mas capacidad de computo
Luna	Utilizando texturas, iluminación y efectos físicos se recrea el escenario lunar	X		

#### Prueba de reconocimiento de imagen

Objeto	Descripción	Satisfactoria	Aceptable	No funciona	Comentarios
Monocromática	Se utiliza un objeto de color azul	X			Este modulo puede ser optimizado aplicando combinando técnicas de reconocimiento de imagen, característica inherente a OpenCV(librería manejada)
Poli-cromática		X			
Región facial	Colores y luminosidad	X			

### Prueba de conexión de inalámbrica

Distancia	Descripción	Satisfactoria	Aceptable	No funciona	Comentarios
20 a 40 cm		X			Esta característica es proporcional al tipo de dispositivo que se utilice, mayor potencia y calidad de transmisión implica mayor costo.
2 metros		X			
5 metros		X			
10 metros			X		

### Tele operación plataforma móvil

Operación	Descripción	Satisfactoria	Aceptable	No funciona	Comentarios
Adelante		X			
Atrás		X			
Izquierda		X			
Derecha		X			

### Tele operación brazo robótico

Operación	Descripción	Satisfactoria	Aceptable	No funciona	Comentarios
Pinza		X			
Base		X			
Bazo		X			

### Tiempo de transmisión

Objeto	Retardo	Satisfactoria	Aceptable	Comentarios
Instancia del puerto serial	1.5 segundos	X		La librería requiere un tiempo de 1.5 segundos para la instanciación del puerto serial, este tiempo solo se requiere al comienzo de la aplicación.
Modulo inalámbrico	1 segundo		X	El dato debe ser mantenido por lo menos 1 segundo en el puerto para que el modulo realice la transmisión y conversión comunicación serial - paralela

**Tiempo de carga del mundo virtual:**

Objeto	Tiempo	Satisfactoria	Aceptable	No funciona
Superficies planas	5 segundos	X		
Terrenos irregulares	7 segundos	X		
Luna	9 segundos	X		

**Versiones de Blender compatibles con las librerías asociadas:**

A partir de la versión 2.49b Blender cambia sus características debido a la actualización de la API de Python en septiembre de 2009 esto implica cambios adicionales como documentos del motor de juegos y corrección de errores. Esta nueva API comienza a desarrollarse el 24 de diciembre de 2009 y llega a la versión estable 2.57 el 13 de abril de 2011.

Versión	Descripción	Satisfactoria	Aceptable	No funciona	Comentarios
Versiones anteriores a 2.49b		X			Debido al cambio de API algunas librerías aun no se encuentran desarrolladas o son compatibles en su totalidad.
Versiones 2.5 X en adelante				X	

**Comprobación de la hipótesis**

Los resultados pueden observarse mediante las evaluaciones realizadas, de acuerdo a esto se puede reafirmar es posible integrar de las tecnologías relacionadas con las áreas de realidad virtual, teleoperación y reconocimiento de imágenes, para el desarrollo de sistemas de tele presencia en cualquier área específica de esta manera la hipótesis queda comprobada.

### Anexo E. Algoritmo Head Traking

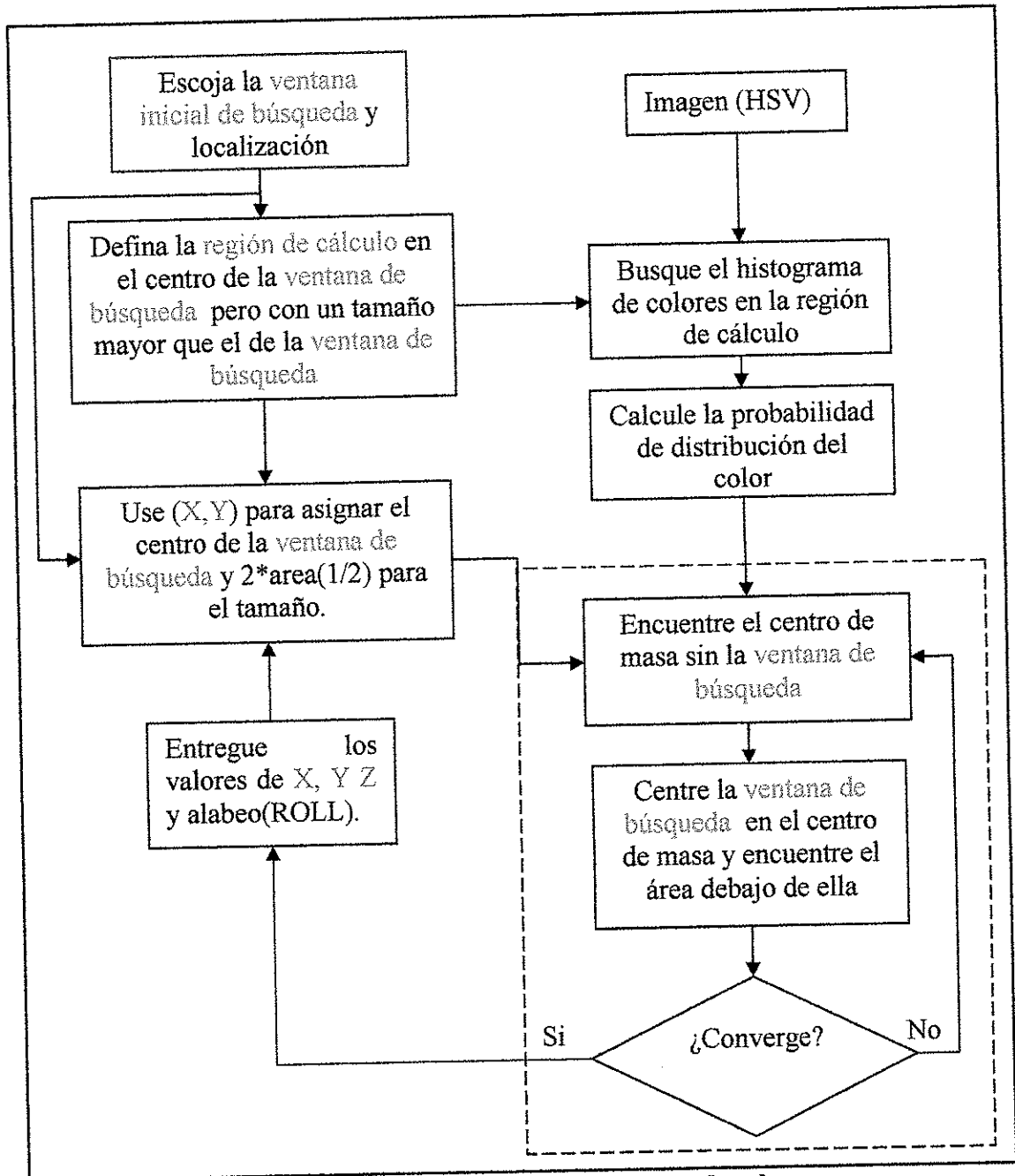
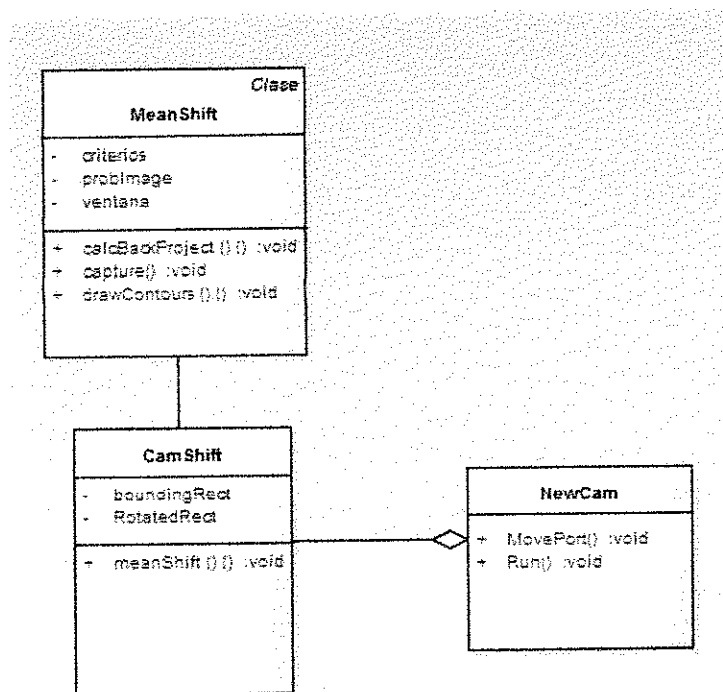


Figura 7. Diagrama de seguimiento de color  
Fuente: El autor basado en [15] pág. 298

## Anexo F. Descripción de clases

En la figura 1 se muestra el diagrama de clases que soporta el diseño de la interface gestual.



**Figura 8 Agregación de la clase Newcam**

**Fuente: El autor**

### **CamShift:**

Esta clase encuentra el centro del objeto, el tamaño y la orientación.

Python: `cv2.CamShift (probImage, ventana, criterios) → retval, ventana`

Parámetros:

`probImage` - la retroproyección del histograma objeto. “`calcBackProject ()`”

`ventana` - ventana de búsqueda inicial.

`criterios` – Entrega los criterios para la fundamental “`meanShift ()`.”

La función implementa el objeto de la clase CAMSHIFT. En primer lugar, se encuentra el centro del objeto con `meanShift ()` y luego ajusta el tamaño de la ventana y encuentra a la rotación óptima. La función devuelve la estructura rectángulo girado que incluye la posición del objeto, el tamaño y la orientación. La siguiente posición de la ventana de búsqueda puede obtener con `RotatedRect :: boundingRect ()`.

Esta función junto con el histograma se usa para rastrear objetos de colores.

## MeanShift

Encuentra un objeto en una imagen de retro proyección

Python: cv.MeanShift (probImage, ventana, criterios) → borrador

Parámetros:

probImage - la retro proyección del histograma objeto. calcBackProject ()

ventana - ventana de búsqueda inicial.

criterios - entrega los criterios el algoritmo iterativo de búsqueda.

La función implementa un algoritmo iterativo de búsqueda de un objeto. Se toma la retro proyección de un objeto y la posición inicial. Se calcula el centro de masa en la ventana de la imagen de retro proyección, y el centro de la ventana de búsqueda se desplaza hacia el centro de masa. El procedimiento se repite hasta el número especificado de iteraciones (criteria.maxCount) se hace o hasta que la diferencia entre el centro de la ventana sea menor que criteria.epsilon. El algoritmo se utiliza dentro de CamShift, el tamaño de la ventana de búsqueda o la orientación no cambian durante la búsqueda. Es posible cambiar este parámetro mediante calcBackProject ().

Se pueden obtener mejores resultados si se realiza un pre-filtro de la retro proyección y eliminar el ruido.

Es posible hacer esto mediante la recuperación de los componentes relacionados, mediante findContours (), para luego evaluar contornos con un área pequeña (contourArea ()), y los contornos restantes con drawContours ().

En la figura 6 se muestra el diagrama de clases del sistema de tele presencia.

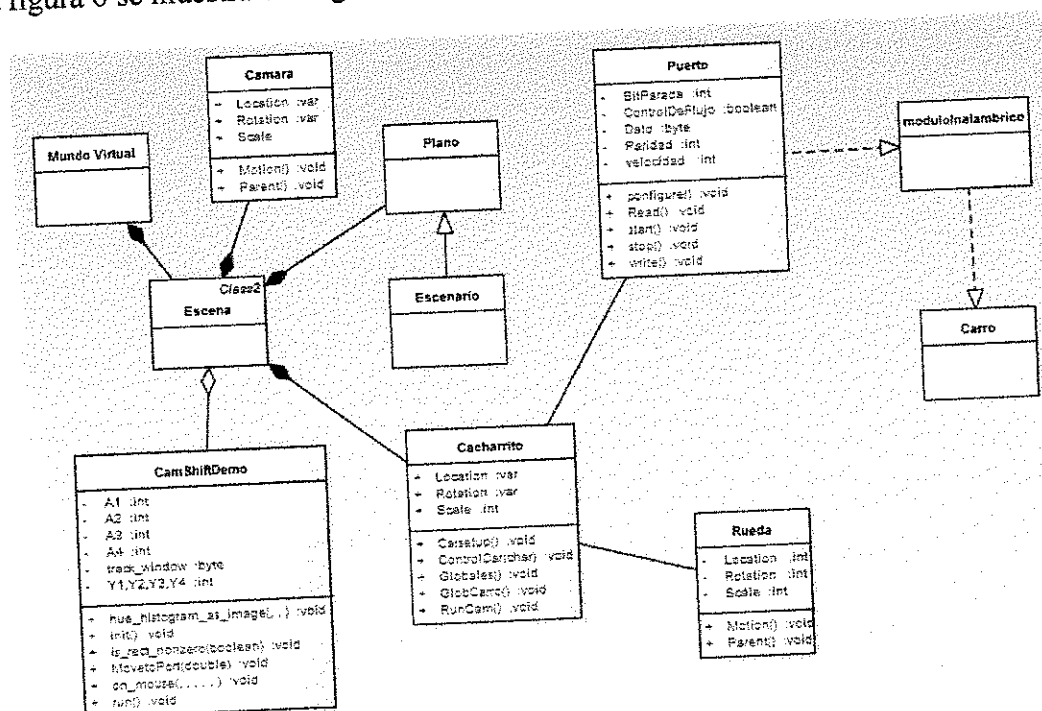


Figura 9. Diagrama de clases  
Fuente: El autor

## Anexo G. Desarrollo Módulo interface Gestual

Para que la interface gestual funcione en Blender es necesario incluir la librería GameLogic, mediante esta librería se relaciona el código de la cámara y el juego.

En la tabla 1 se especifican los métodos utilizados para el desarrollo del módulo de la interface gestual.

Método o función	Parámetros	Descripción
MovetoPort	Track_window	Track_window contiene las coordenadas donde se encuentra el nuevo objeto, de acuerdo a este valor este método envía el dato correspondiente al puerto
is_rect_nonzero	r	Determina si las coordenadas w y h del vector r son mayores que cero
getCurrentScene		Retorna la escena en un vector conformado por los objetos que se encuentran en ella
CaptureFromCAM	0	Captura una fotografía
NamedWindow	1, String	Le asigna el valor de la cadena al atributo <i>name</i> de la ventana.
CalcBackProject	Imagen, retroproyeccion, histograma	Calcula la retroproyección del histograma. Para cada tupla de pixeles en la misma posición de la selección compara con la retroproyección y devuelve la coincidencia.
CvtColor	Src(cvarr),dst(cvarr),code(int)	Convierte la imagen de entrada a otro espacio de colores mediante los parámetros imagen fuente, imagen destino, operación.
write	string	Escribe un dato en el puerto.
CreateImage	frame	Crea una imagen a partir de un frame

**Tabla 1. Lista de métodos utilizados**

**Fuente: El autor**

```

import cv2.cv as cv
import GameLogicas g
import serial
import time

def is_rect_nonzero(r):
    (_,_,w,h) = r
    return (w >0) and (h >0)

def MovetoPort(track_window):
    """traza las zonas adelante atrás derecha izquierda
    y las combinaciones para determinar los datos en el puerto"""
    A1=0
    A2=240
    A3=300
    A4=640
    Y1=0
    Y2=50
    Y3=80
    Y4=480
    (P1,P2, Ancho, Alto) = track_window
    print P1,P2, Ancho, Alto
    P1=P1+Ancho/2
    P2=P2+Alto/2
    print P1,P2
    OBCube = g.getCurrentScene().objects["OBverde"]
    #OBCube ["COM"].write(KEY)
    #OBCube ["KEY"]=KEY
    if P1>=A1 and P1<A2:
        if P2>=A1 and P2<A2:
            print("Adelante-Izquierda")
            OBCube ["COM"].write("q")
    if P2>=A2 and P2<A3:
        print("Izquierda")
        OBCube ["COM"].write("a")
    if P2>=A3 and P2<A4:
        print("Atras-Izquierda")
        OBCube ["COM"].write("z")

    if P1>=A2 and P1<A3:
        if P2>=A1 and P2<A2:
            print("Adelante")
            OBCube ["COM"].write("w")
    if P2>=A2 and P2<A3:
        print("Stop")
    if P2>=A3 and P2<A4:
        print("Atras")
        OBCube ["COM"].write("s")

    if P1>=A3 and P1<=A4:
        if P2>=A1 and P2<A2:
            print("Adelante-Derecha")

```

```

        OBCube ["COM"].write("e")
if P2>=A2 and P2<A3:
print("Derecha")
        OBCube ["COM"].write("d")
if P2>=A3 and P2<=A4:
print("Atras-Derecha")
        OBCube ["COM"].write("c")

#Fin Moveport

class CamShiftDemo():
def __init__(self):
    self.capture = cv.CaptureFromCAM(0)
    cv.NamedWindow( "CamShiftDemo", 1 )
    cv.NamedWindow( "Histogram", 1 )
    cv.SetMouseCallback( "CamShiftDemo", self.on_mouse)
    self.hist = cv.CreateHist([180], cv.CV_HIST_ARRAY, [(0,180)], 1 )
    self.backproject_mode = False

    self.drag_start = None # Asignado a (x,y) cuándo el mouse
# comienza el arrastre
self.track_window = None #Asignado arect cuando el mouse
# finaliza el arrastre

print( "LETRAS:\n"
        "     ESC-Salir del Programa\n"
        "     b - Cambia a vista backprojection \n"
        "     para iniciar el tracking, arrastre"
        "     alrededor del objeto a seguir con el mouse\n" )

def hue_histogram_as_image(self, hist, cv):
    """ Retorna una representación del histograma de colores """
    histimg_hsv = cv.CreateImage( (320,200), 8, 3)

    mybins = cv.CloneMatND(hist.bins)
    cv.Log(mybins, mybins)
    (_, hi, _, _) = cv.MinMaxLoc(mybins)
    cv.ConvertScale(mybins, mybins, 255. / hi)

    w,h = cv.GetSize(histimg_hsv)
    hdims = cv.GetDims(mybins)[0]
    for x in range(w):
        xh = (180 * x) / (w - 1)
        #Pasos de colores desde 0-180 alrededor de la imagen
        val = int(mybins[int(hdims * x / w)] * h / 255)
        cv.Rectangle( histimg_hsv, (x, 0), (x, h-val), (xh,255,64), -1)
        cv.Rectangle( histimg_hsv, (x, h-val), (x, h), (xh,255,255), -1)

    histimg = cv.CreateImage( (320,200), 8, 3)
    cv.CvtColor(histimg_hsv, histimg, cv.CV_HSV2BGR)
    return histimg

```

```

def on_mouse(self, event, x, y, flags, param):
if event == cv.CV_EVENT_LBUTTONDOWN:
    self.drag_start = (x, y)
if event == cv.CV_EVENT_LBUTTONUP:
    self.drag_start = None
    self.track_window = self.selection
if self.drag_start:
    xmin = min(x, self.drag_start[0])
    ymin = min(y, self.drag_start[1])
    xmax = max(x, self.drag_start[0])
    ymax = max(y, self.drag_start[1])
    self.selection = (xmin, ymin, xmax - xmin, ymax - ymin)

def run(self, cv):

    frame = cv.QueryFrame( self.capture )

# convertir aHSVy mantener el color
hsv = cv.CreateImage(cv.GetSize(frame), 8, 3)
    cv.CvtColor(frame, hsv, cv.CV_BGR2HSV)
    self.hue = cv.CreateImage(cv.GetSize(frame), 8, 1)
    cv.Split(hsv, self.hue, None, None, None)
# Calcular la proyección
    backproject = cv.CreateImage(cv.GetSize(frame), 8, 1)
# Ejecutar cam-shift
    cv.CalcArrBackProject( [self.hue], backproject, self.hist )
if self.track_window and is_rect_nonzero(self.track_window):
    crit = ( cv.CV_TERMCRIT_EPS | cv.CV_TERMCRIT_ITER, 10, 1)
        (iters, (area, value, rect), track_box) =
cv.CamShift(backproject, self.track_window, crit)
self.track_window = rect
# Si el mouse es presionado, esconda la selección y
# recalcula el histograma
if self.drag_start and is_rect_nonzero(self.selection):
    sub = cv.GetSubRect(frame, self.selection)
    save = cv.CloneMat(sub)
    cv.ConvertScale(frame, frame, 0.5)
    cv.Copy(save, sub)
    x,y,w,h = self.selection
cv.Rectangle(frame, (x,y), (x+w,y+h), (255,255,255))
print "self.selection=",x,y,w,h
    sel = cv.GetSubRect(self.hue, self.selection )
    cv.CalcArrHist( [sel], self.hist, 0)
    (_, max_val, _, _) = cv.GetMinMaxHistValue( self.hist)
if max_val != 0:
    cv.ConvertScale(self.hist.bins, self.hist.bins, 255. / max_val)
elif self.track_window and is_rect_nonzero(self.track_window):
    cv.EllipseBox( frame, track_box, cv.CV_RGB(255,0,0), 3,
cv.CV_AA, 0 )
        #Salto arutinadirecciónobjeto
if self.track_window:

```

```

        MovetoPort(self.track_window)
if not self.backproject_mode:
    cv.ShowImage( "CamShiftDemo", frame )
else:
    cv.ShowImage( "CamShiftDemo", backproject)
    cv.ShowImage( "Histogram", self.hue_histogram_as_image(self.hist,
cv))
    c = cv.WaitKey(7) % 0x100
if c == ord("b"):
    self.backproject_mode = not self.backproject_mode

g.demo = CamShiftDemo()#instancia la clase camshiftdemo

```

**Tabla 10 Código rastreo de objetos**

**Fuente: El autor**

## Anexo H. Comunicación PC a modulo inalámbrico

En la tabla 1 se especifican las funciones o métodos utilizados para la comunicación serial entre el computador y el modulo inalámbrico mediante la tarjeta microcontrolada de Arduino.

Funciones o métodos	Parámetros	Descripción
String	Cadena(String )	Constructor de la clase string
pinMode	Pin(int), modo(string)	Configura el pin de acuerdo al modo recibido (entrada o salida)
begin	Velocidad(int)	Inicializa el puerto con la velocidad especificada
read		Lee el valor que se encuentra en el puerto
println	Cadena(string)	Imprime la cadena en el puerto serial con salto de línea
delay	Retard(string)	Detiene el programa por una cantidad de tiempo "retardo"

**Tabla 1. Lista de métodos utilizados**

**Fuente: El autor**

```
String inString = ""; // string to hold input
String a = String("a");
String d = String('d');
String w = String("w");
String s = String("s");
String o = String("o");
String i = String("i");
String u = String("u");
String l = String("l");
String k = String("k");
String j = String("j");
String q = String("q");
String z = String("z");
String e = String("e");
String c = String("c");
int retardo =1;
int P1=11;
int P2=10;
int P3=9;
int P4=8;

void setup()
```

```

{
  // initialize the serial communication:
  Serial.begin(9600);
  // initialize the ledPin as an output:

  pinMode(P1, OUTPUT);
  pinMode(P2, OUTPUT);
  pinMode(P3, OUTPUT);
  pinMode(P4, OUTPUT);
  pinMode(13, OUTPUT);
}
void loop(){

  if (Serial.available()) {
    // read the most recent byte (which will be from 0 to 255):
    int inChar = Serial.read(); // Serial.write(Serial.read());
    inString += (char)inChar;
    if (inString >= a && inString <= z && inString != "\n") { //si se termina
la cadena
      // Serial.println(inString);
      // Serial.println( inChar);
      if(inString.equals(w)){
        Serial.println("ADELANTE");
        digitalWrite(P4,HIGH);//0001
        delay (retardo);
      }
      if(inString==s){
        Serial.println("ATRAS");
        digitalWrite(P3, HIGH);//0010
        delay (retardo);
      }
      if(inString==d){
        Serial.println("DERECHA");
        digitalWrite(P3, HIGH);//0011
        digitalWrite(P4, HIGH);
        delay (retardo);
      }
      if(inString==a){//atras
        Serial.println("IZQUIERDA");
        digitalWrite(P2, HIGH);//0100
        delay (retardo);
      }
      if(inString==o){
        Serial.println("PINZA abre");
        digitalWrite(P2, HIGH);//0101
        digitalWrite(P4, HIGH);
        delay (retardo);
      }
      if(inString==l){
        Serial.println("PINZA cierra");
        digitalWrite(P2, HIGH);//0110

```

```

digitalWrite(P3, HIGH);
  delay (retardo);
}
  if(inString==i){
Serial.println("BASE sube");
digitalWrite(P2, HIGH);//0111
digitalWrite(P3, HIGH);
digitalWrite(P4, HIGH);
  delay (retardo);
}
    if(inString==k){
Serial.println("BASE baja");
digitalWrite(P1, HIGH);//1000
delay (retardo);
}
    if(inString==u){
Serial.println("CODO sube");
digitalWrite(P1, HIGH);//1001
digitalWrite(P4, HIGH);
  delay (retardo);
}
    if(inString==j){
Serial.println("CODO baja");
digitalWrite(P1, HIGH);//1010
digitalWrite(P3, HIGH);
delay (retardo);
}
    if(inString==e){
Serial.println("Adelante con d");
digitalWrite(P1, HIGH);//1011
digitalWrite(P3, HIGH);
digitalWrite(P4, HIGH);
delay (retardo);
}
    if(inString==c){
Serial.println("Atras con d");
digitalWrite(P1, HIGH);//1100
digitalWrite(P2, HIGH);
delay (retardo);
}
    if(inString==q){
Serial.println("Adelante con Iz");
digitalWrite(P1, HIGH);//1101
digitalWrite(P2, HIGH);
digitalWrite(P4, HIGH);

delay (retardo);
}
    if(inString==z){
Serial.println("Atras con Iz");
digitalWrite(P1, HIGH);//1110
digitalWrite(P2, HIGH);
digitalWrite(P3, HIGH);

```

```
delay (retardo);  
}  
  
//fin>  
digitalWrite(P4,LOW);  
digitalWrite(P1,LOW);  
digitalWrite(P2,LOW);  
digitalWrite(P3,LOW);  
  
}inString = "";  
}  
}
```

**Tabla 11 Código interface Arduino uno**  
**Fuente: el autor**

## Anexo I. Servomecanismo

En la tabla 1 se especifican las funciones o métodos utilizados para el código del microcontrolador.

Funciones	Parámetros	Descripción
Main	(Void)	
port_b_pullups	Condición(Bool)	Esta función activa o desactiva las resistencias de pullups en las entradas del puerto B. La condición puede ser TRUE (activa) o FALSE (desactiva).
set_tris_a set_tris_a set_tris_a	NoPin(HEX)	Define los estados 1 como entradas y 0 como salidas No pines es un número en hexadecimal.
output_low	PIN_PUERTO_NUMERO	Pone en bajo el pin del puerto
output_high	PIN_PUERTO_NUMERO	Pone en alto el pin del puerto
input	PIN_PUERTO_NUMERO	Lee el pin del puerto

**Tabla 1. Lista de métodos utilizados**

**Fuente: El autor**

```
void main()
{

    port_b_pullups (TRUE);

    set_tris_a(0xFF);
    set_tris_c(0x00);
    set_tris_b(0x00);
    set_tris_d(0x00);

    output_low(PIN_B1);
    while (1)
    {

        if (input (PIN_A3)==0 && input (PIN_A2)==0 && input (PIN_A1)==0
            && input (PIN_A0)==0){
```

```

output_low(PIN_B3);
output_low(PIN_B2);
output_low(PIN_B1);
output_low(PIN_B0);
output_low(PIN_D7);
output_low(PIN_D6);
output_low(PIN_D5);
output_low(PIN_D4);
output_low(PIN_C7);
output_low(PIN_C6);
}
if (input (PIN_A3)==0 && input (PIN_A2)==0 && input (PIN_A1)==0
    && input (PIN_A0)==1)
{output_high(PIN_D7);output_high(PIN_B1);}

if (input (PIN_A3)==0 && input (PIN_A2)==0 && input (PIN_A1)==1
    && input (PIN_A0)==0)
{output_high(PIN_D6);output_high(PIN_B0);}

if (input (PIN_A3)==0 && input (PIN_A2)==0 && input (PIN_A1)==1
    && input (PIN_A0)==1)
{output_high(PIN_D6);output_high(PIN_B1);}

if (input (PIN_A3)==0 && input (PIN_A2)==1 && input (PIN_A1)==0
    && input (PIN_A0)==0)
{output_high(PIN_D7);output_high(PIN_B0);}

if (input (PIN_A3)==0 && input (PIN_A2)==1 && input (PIN_A1)==0
    && input (PIN_A0)==1)
output_high(PIN_C7);

if (input (PIN_A3)==0 && input (PIN_A2)==1 && input (PIN_A1)==1
    && input (PIN_A0)==0)
output_high(PIN_C6);

if (input (PIN_A3)==0 && input (PIN_A2)==1 && input (PIN_A1)==1
    && input (PIN_A0)==1)
output_high(PIN_B2);

if (input (PIN_A3)==1 && input (PIN_A2)==0 && input (PIN_A1)==0
    && input (PIN_A0)==0)
output_high(PIN_B3);

if (input (PIN_A3)==1 && input (PIN_A2)==0 && input (PIN_A1)==0

```

```

    && input (PIN_A0)==1)
output_high(PIN_D4);

if (input (PIN_A3)==1 && input (PIN_A2)==0 && input (PIN_A1)==1
    && input (PIN_A0)==0)
output_high(PIN_D5);

if (input (PIN_A3)==1 && input (PIN_A2)==0 && input (PIN_A1)==1
    && input (PIN_A0)==1)
output_high(PIN_D7);

if (input (PIN_A3)==1 && input (PIN_A2)==1 && input (PIN_A1)==0
    && input (PIN_A0)==0)
output_high(PIN_D6);

if (input (PIN_A3)==1 && input (PIN_A2)==1 && input (PIN_A1)==0
    && input (PIN_A0)==1)
output_high(PIN_B1);

if (input (PIN_A3)==1 && input (PIN_A2)==1 && input (PIN_A1)==1
    && input (PIN_A0)==0)
output_high(PIN_B0);

}

}

```

**Tabla 12 Código microcontrolador**

**Fuente: El autor**

# Anexo J. Datasheet L293D

## L293, L293D QUADRUPLE HALF-H DRIVERS

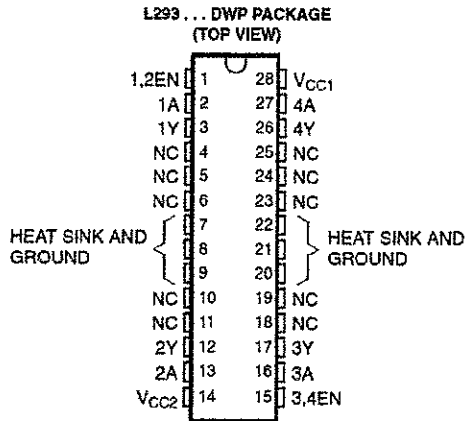
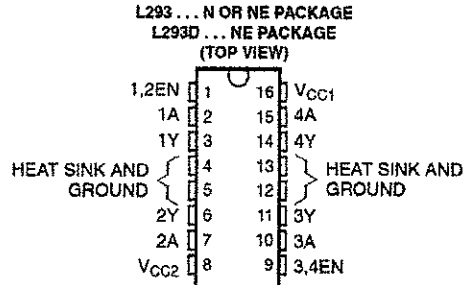
SLRS006C – SEPTEMBER 1996 – REVISED NOVEMBER 2004

- Featuring Unifrode L293 and L293D Products Now From Texas Instruments
- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- Thermal Shutdown
- High-Noise-Immunity Inputs
- Functionally Similar to SGS L293 and SGS L293D
- Output Current 1 A Per Channel (600 mA for L293D)
- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

### description/ordering information

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.



### ORDERING INFORMATION

T <sub>A</sub>	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	HSOP (DWP)	Tube of 20	L293DWP	L293DWP
	PDIP (N)	Tube of 25	L293N	L293N
	PDIP (NE)	Tube of 25	L293NE	L293NE
		Tube of 25	L293DNE	L293DNE

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at [www.ti.com/sc/package](http://www.ti.com/sc/package).



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS  
INSTRUMENTS**  
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2004, Texas Instruments Incorporated

1

## Anexo K. Manual Básico de Blender

Blender es un software libre para diseño 3D multiplataforma, esto es que puede ser usado en cualquier sistema operativo. Integra herramientas de edición de imágenes, video, animaciones y juegos.

Blender es un mundo fascinante para los amantes del diseño, la creatividad y la ingeniería 3D, este no es un manual instructivo de cómo realizar productos 3D, es una introducción al manejo de la herramienta en su versión 2.5 y algunas experiencias con el manejo de la misma, pero será necesaria la actitud y las ganas de aprender del estudiante inquietud para dominar la herramienta, solo la práctica pues hace al maestro.

- **Interfaz**

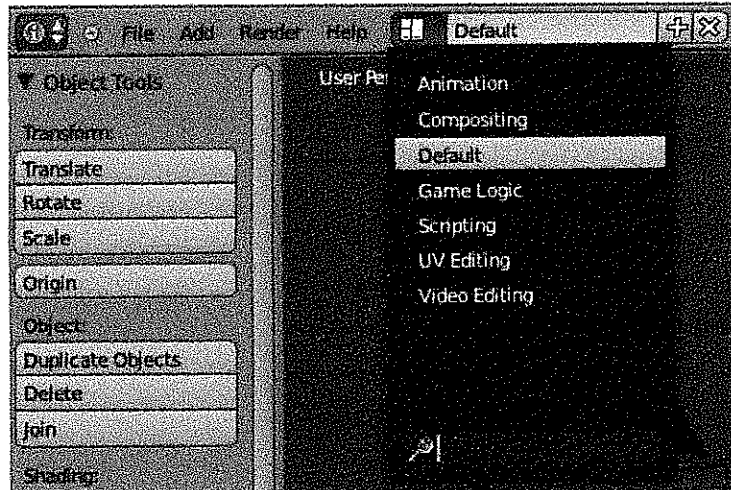
Blender utiliza un concepto dinámico de ventanas lo que lo hace un poco diferente al modelo tradicional de los demás software de diseño, pero esto hace que podamos personalizarlo.



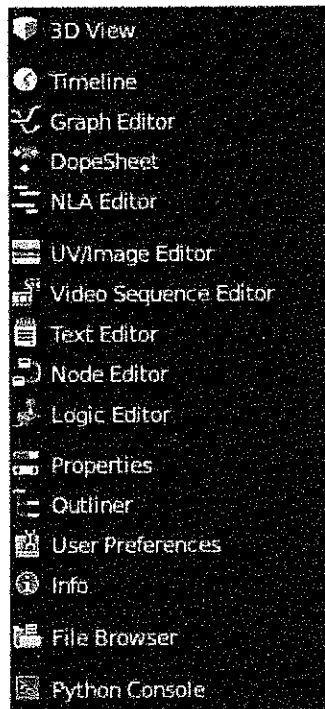
**Figura 10 Interface de Blender**

**Fuente: El autor**

Trae por defecto 7 pantallas de diseño, que organizan de manera diferente alguno de los 16 tipos de editores como se muestra a continuación, pero podemos agregar o modificarlas para obtener las que nos parezcan óptimas. Para navegar entre las 7 pantallas se pulsa control + flecha derecha.



**Figura 11 (Pantallas de diseño)**  
**Fuente: El autor**



**Figura 12 (Tipos de editores)**  
**Fuente: El autor**

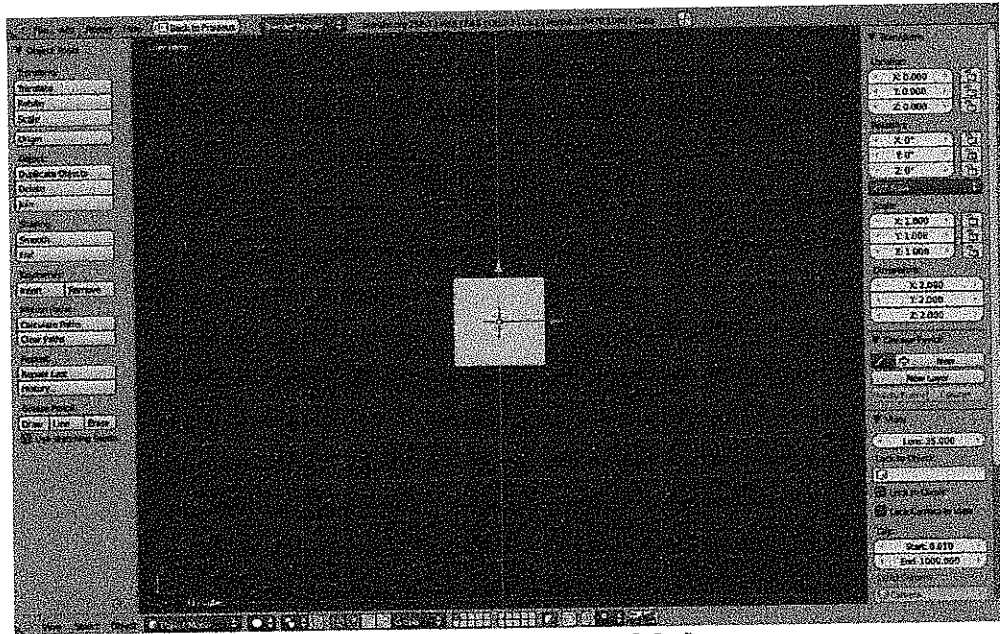
Para modificar el tamaño de una ventana basta con ubicar el puntero del mouse sobre el borde de la misma y arrastrar a la posición deseada.

Si se quiere cambiar el tipo de editor, se coloca el puntero del mouse sobre el icono que aparece a mano izquierda sobre la barra de herramientas de la ventana que se quiera modificar y posteriormente se selecciona del menú de la figura 2.

Para eliminar o crear una nueva ventana colocamos el puntero del mouse sobre el borde de la ventana, pulamos el botón derecho del mismo y aparecerán las dos opciones, en el primer caso seleccionaremos el lugar donde se dividen las dos ventanas y para el segundo la ventana sobre la cual se colocara la ventana que queremos conservar.

- **Modelado**

El modelado generalmente se realiza en la pantalla por defecto en el editor de vista 3D.



**Figura 13 Ventana de modelado**

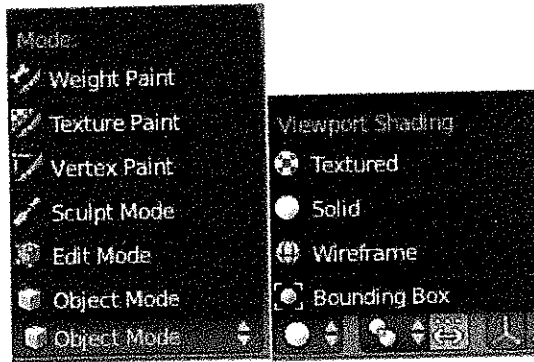
**Fuente: El autor**

Para ver cualquier menú en vista completa se usa control + flecha arriba.

Con las teclas N y T podremos desplegar los menús de la derecha e izquierda respectivamente.

La barra de herramientas se posiciona en el fondo de la ventana del editor pero se puede posicionar en la parte superior con un simple arrastre.

Existen 5 modos de edición y 4 formas de visualización de los elementos que se pueden apreciar en la figura y son de fácil selección mediante el evento de clic sobre la barra inferior.



**Figura 14 Modos de edición**

**Fuente: El autor**

Un comando muy útil es el cambio de textura a modo sólido que se hace mediante la combinación  $\text{alt} + \text{z}$ . Otro es la tecla  $\text{tab}$  o tabulador, para alternar entre los modos objeto y edición.

Por el momento para el modelado nos interesa la vista en modo sólido y malla, y los modos objeto y edición.

### **Escalado y cambio de posición:**

El escalado se realiza mediante la tecla  $s$  y el movimiento del mouse, el cambio de posición se hace mediante la tecla  $g$  y el movimiento del mouse. Si se quiere hacer movimiento en un eje en especial se usa la tecla  $g$  y luego el eje en el cual se quiere desplazar la figura, por ejemplo:  $(g + z)$ ,  $(g + y)$  ó  $(g + x)$ .

Es posible escalar o modificar tamaño y posición de vértices, lados y áreas. Mediante la selección de estos modos en la barra de herramientas o la figura completa mediante la tecla  $A$ .

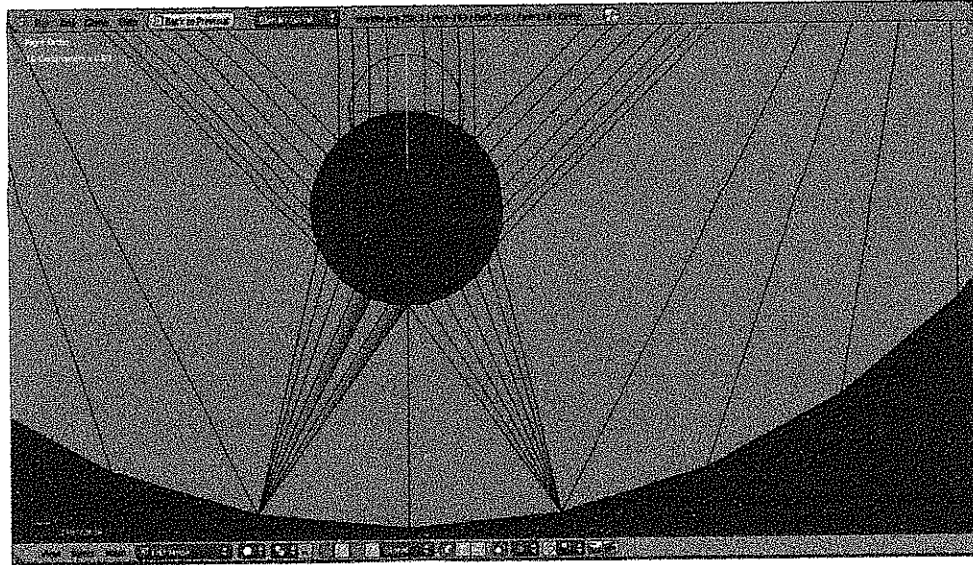


### **Extrusión y copiado:**

Se realiza mediante la tecla  $e$  en el modo edición, esta genera una copia de la selección que podemos desplazar a nuestra conveniencia pero es automáticamente asociada a la figura a diferencia de la opción  $\text{shift} + \text{d}$  en el modo objeto, que genera una copia independiente.

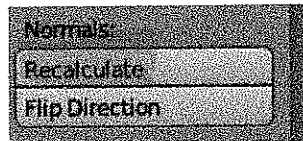
A partir de dos puntos es posible generar una línea seleccionándolos y pulsando la tecla  $F$ . de Fill llenado, esta misma lógica se usa para completar figuras mediante 2, 3 y 4 puntos.

Un ejemplo es la figura que aparece a continuación.



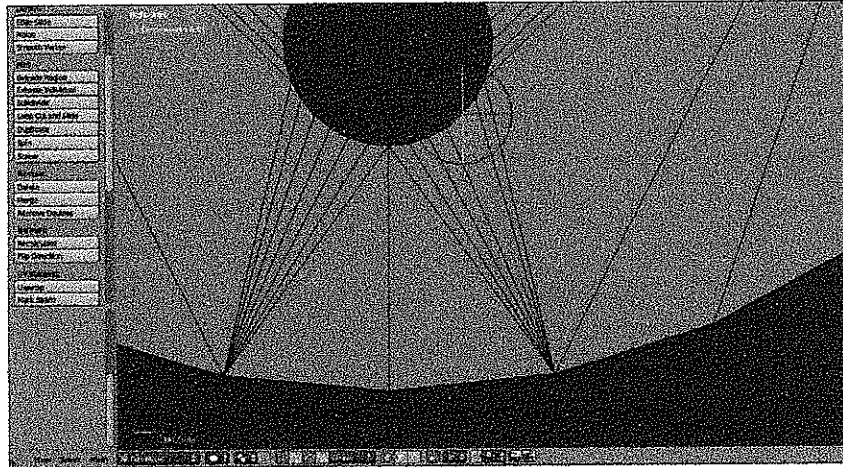
**Figura 15 Detalle rueda sin recalcu**  
**Fuente: El autor**

Si nos damos cuenta hay momentos en que se sobre ponen las áreas afectando el renderizado mediante un tono más oscuro. Para solucionar este frecuente inconveniente al usar la técnica de llenado, se utiliza la función Recalculate del menú izquierdo herramientas de objeto.



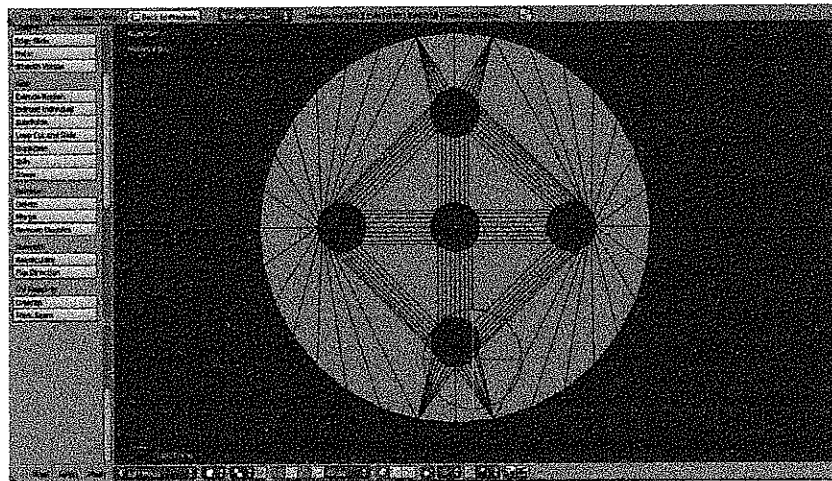
**Figura 16 Menú herramientas de objeto**  
**Fuente: El autor**

Una vez usada esta función, el aspecto se mejora como se ve a continuación en detalle de una rueda:



**Figura 17 Detalle rueda con recalculo**  
**Fuente: El autor**

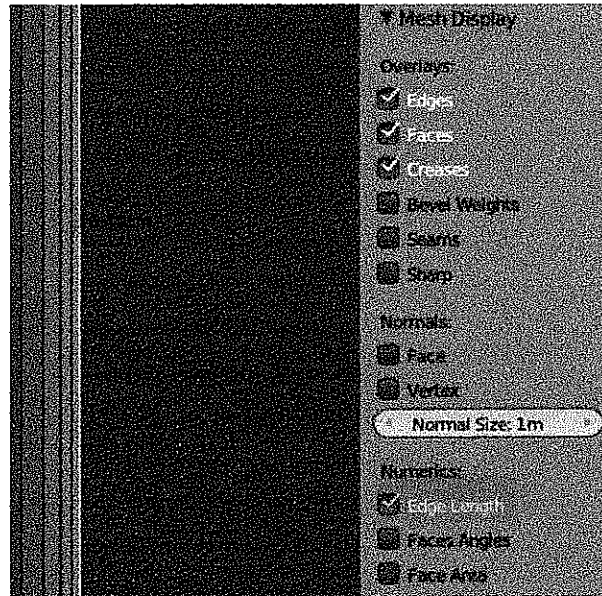
Y la figura completa se vería algo así:



**Figura 18 Rueda final**  
**Fuente: El autor**

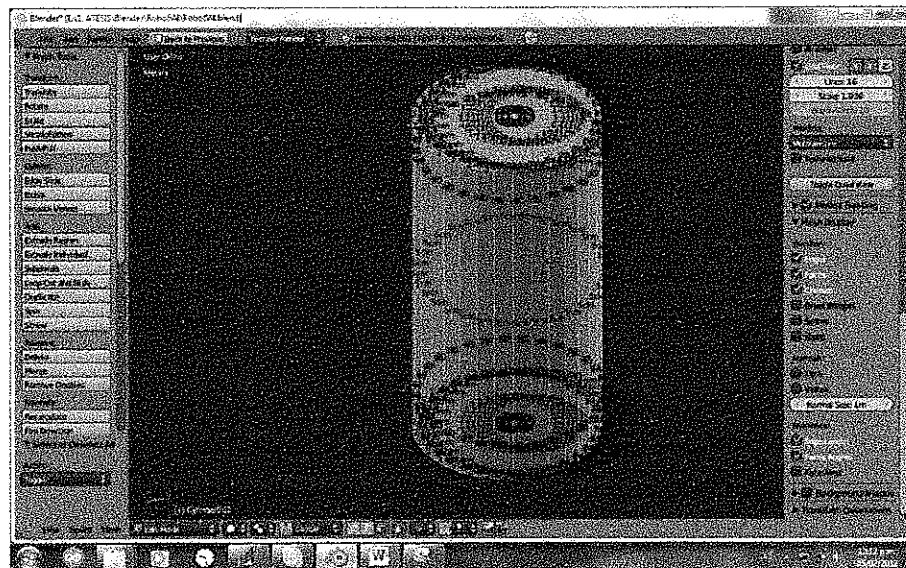
Siempre que se desee mejorar la precisión decimal y modificar las medidas a escala real es importante, primero trabajar en modo edición y segundo usar la siguiente configuración.

En modo edición puedo observar las dimensiones del objeto virtual y ajustarlo según se requiera mediante la opción *edge length* como se observa en la Figura 19.



**Figura 19 Menú edición de escala**  
**Fuentes: El autor**

Analicemos la modificación de la siguiente figura, creamos el cilindro (add/mesh/cylinder), luego pasamos a modo edición, seleccionamos vista de medidas como se menciona anteriormente, se verá algo como se muestra a continuación Figura 20:



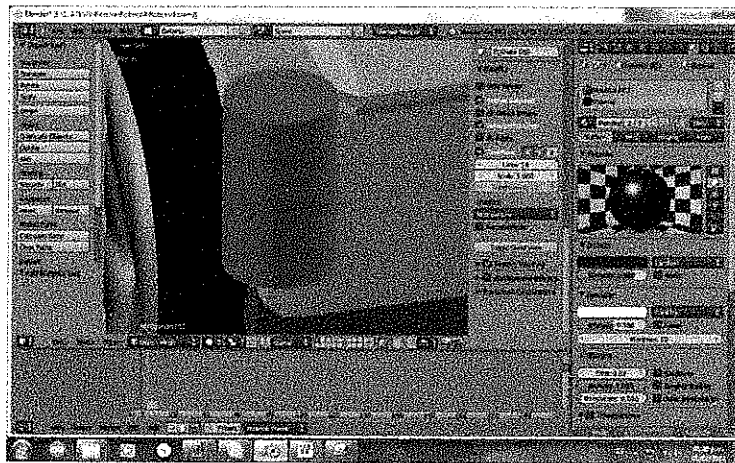
**Figura 20 Detalle de medidas**  
**Fuente: El autor**

Luego seleccionamos el eje que deseamos modificar (X,Y,Z), si se quieren modificar 2 entonces presionamos la excepción (shift + el que no queremos seleccionar), si se quieren modificar todos la letra A.

Una vez seleccionado nuestro objetivo, solo hay que tener en cuenta los siguientes pasos:

- ✚ Seleccionar medida
- ✚ Modo edición
- ✚ Seleccionar todo
- ✚ Selecciona g grap
- ✚ Seleccionar el eje
- ✚ Ajustar
- ✚ Si el tamaño no permite la edición a 3 cifras ajustar mediante zoom área
- ✚ Si se desea texturizar diferente se pinta la selección con otro color.

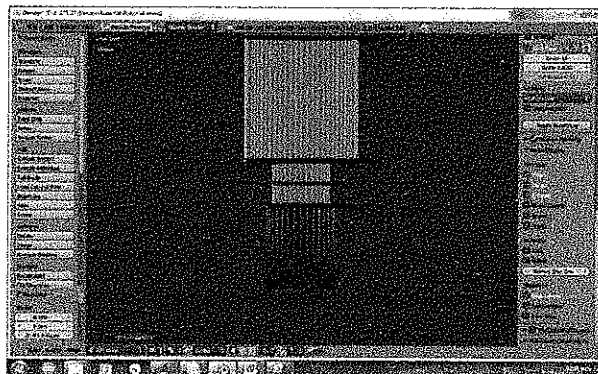
Otra ventaja de Blender frente a otros modeladores es la facilidad de texturizar, es posible darle color a una parte específica del objeto, basta con seleccionarla en el modo edición y añadir el material que se desee como de muestra en la Figura 23.



**Figura 21 Color mediante definición de material**

Fuente: El autor

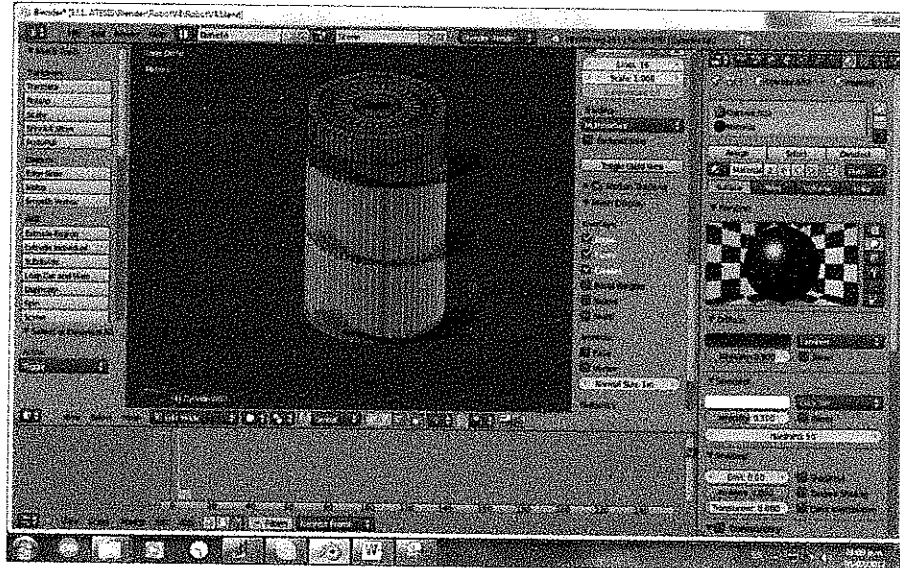
- **Materiales**



**Figura 22 Edición de medidas**

Fuente: El autor

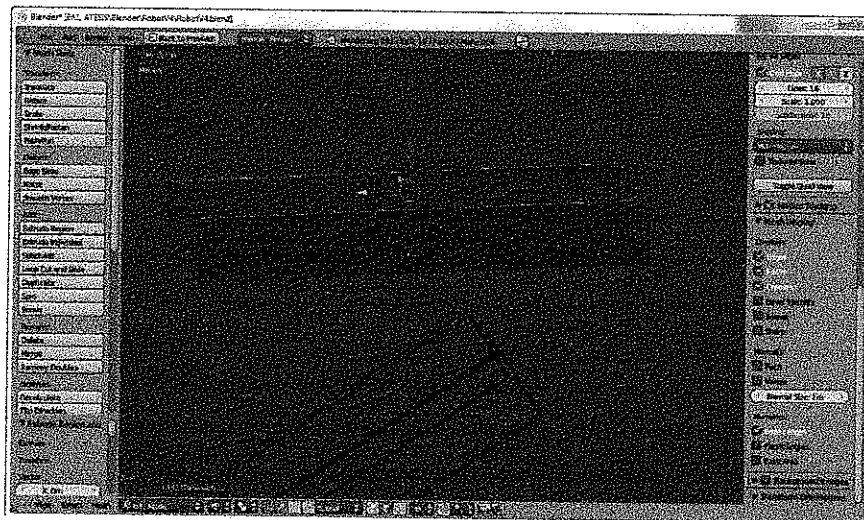
La creación de materiales básicamente consiste en la mezcla de texturas y colores mediante algoritmos matemáticos que se traduce en configuración de propiedades de las mismas, y el uso de la técnica artística de estarcido, también llamado estencil (del inglés stencil).



**Figura 23** Diferentes materiales dentro de un mismo objeto

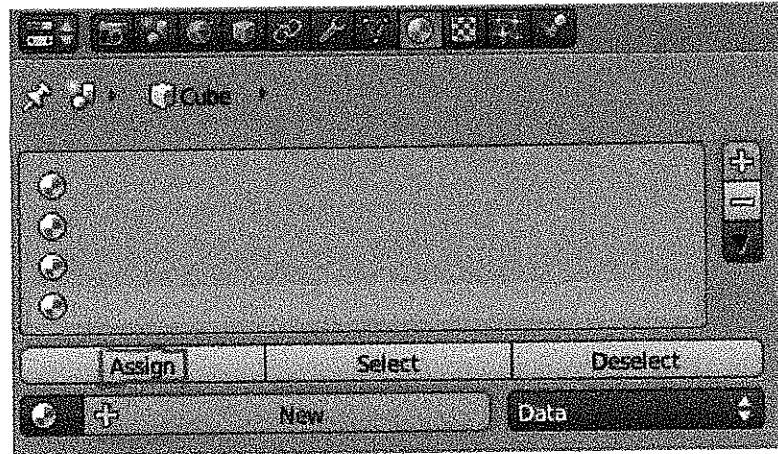
Fuente: El autor

Para crear un nuevo material utilizamos el editor de propiedades seleccionamos el objeto, en la opción material, agregar un nuevo material. Al hacer esto se despliegan todas las opciones de configuración del mismo, seleccionamos el color mediante la opción *Diffuse* del menú material Figura 23.



**Figura 24** Detalle medidas

Fuente: El autor



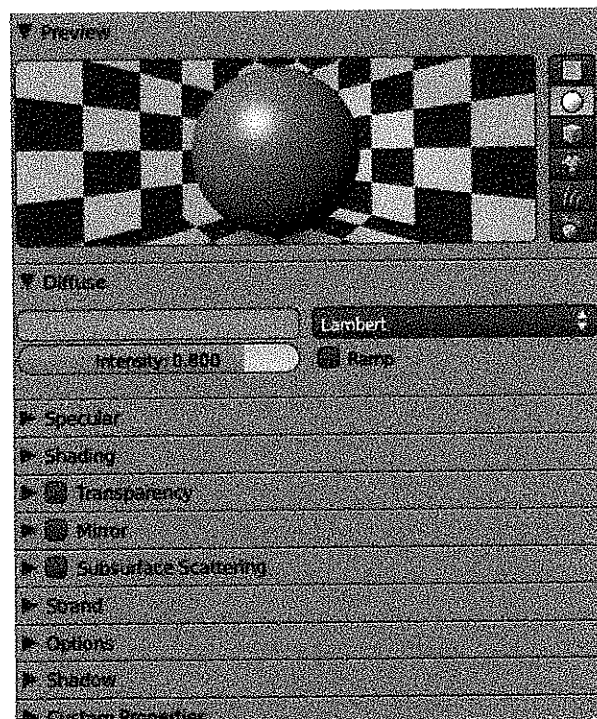
**Figura 25 Creación de materiales**

Fuente: El autor

Es importante no sobre escribir los materiales previamente creados y no olvidar asignar el nuevo material a la selección.

- **Edición de material:**

Para editar el color del material de manera rápida podemos usar la opción Diffuse, este cambiara el color del material según la selección.



**Figura 26 Panel edición de materiales**

Fuente: El autor

- **Renderizado**

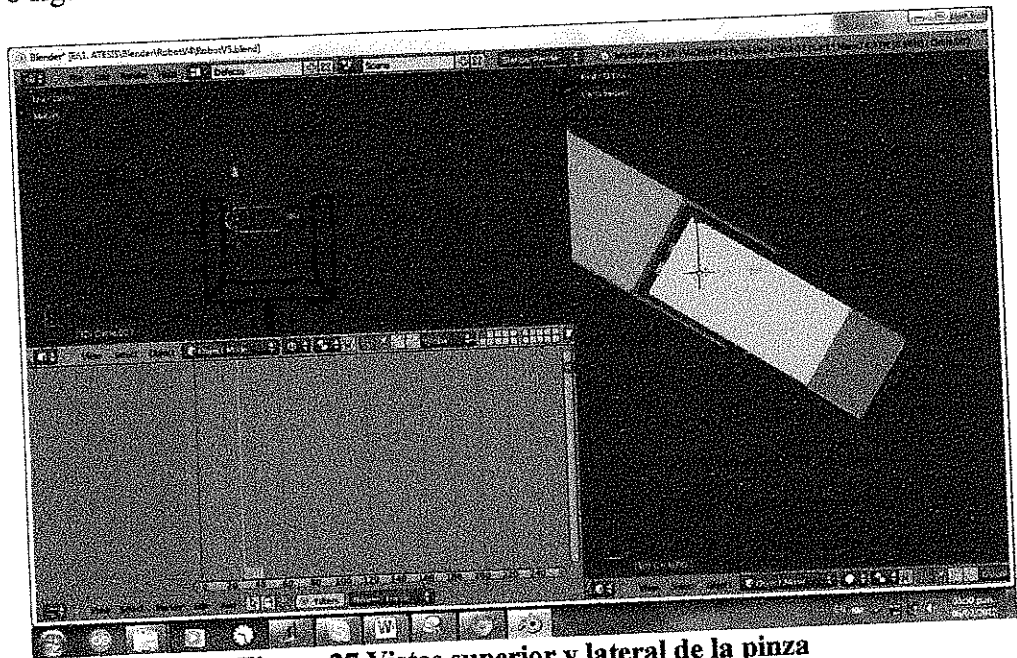
El producto final de nuestro diseño es el *renderizado* ya sea en la película, juego o imagen, por esto es importante tener en cuenta esto todo el tiempo (Tecla F12). Ya que muchas veces se modelan los objetos omitiendo estos detalles.

- **Armadura:**

Para crear una armadura usamos el teclamos barra espaciadora buscamos Add/Armature, para poderla mover en el menú de vista 3D, seleccionamos Pose Mode.

- **Animación**

Una aplicación tiene muchas escenas, una escena tiene muchos frames, un frame es una imagen que se va a renderizar, muchos frames o imágenes generan una película. Entonces la animación consiste en crear paso a paso una película, para ello se usa una escala de tiempo como aparece en la Figura 27. La mejor técnica es colocar puntos de control o figura clave.



**Figura 27 Vistas superior y lateral de la pinza**  
Fuente: El autor

Por ejemplo si se quiere hacer una animación de la apertura y cierre de la pinza de la Figura 27 una vez creada a armadura solo necesitamos dos fotogramas o frames o imágenes clave, la primera es cuando la pinza está abierta totalmente y la segunda cuando está cerrada.

**Anexo L. Artículo**

# Accionamiento Real a Distancia de un Servomecanismo mediante un Mundo Virtual Controlado por Interfaz Gestual

Diego Iván Oliveros Acosta A. Author

**Resumen** – Es la descripción general del sistema de mando a distancia de un servomecanismo, el cual mediante la implementación de un mundo virtual permite evaluar el grado de inmersión y por medio de la interpretación de gestos, con técnicas de imagenología, mejorar el manejo intuitivo y natural. Define la metodología que distribuye el trabajo en cuatro módulos, el mundo virtual, la interfaz gestual, el sistema de comunicación inalámbrica y el servomecanismo.

**Palabras clave** — Robótica, Sistemas de teleoperación, Realidad virtual, Imagenología.

**Abstract**— It is the general description of the remote control system of a servomechanism, which by implementing a virtual world to evaluate the degree of immersion. Through the interpretation of gestures, with imaging techniques improve the natural and intuitive management. Define the methodology to distribute the work in four modules, the virtual world, gestural interface, the wireless communication system and the servo.

**Index Terms**— Robotics, Teleoperated Systems, Immersive Virtual Reality, Imagenology.

## I. INTRODUCCIÓN

Actualmente la comunidad científica trabaja en el desarrollo de sistemas óptimos para el intercambio de datos, de manera que estos garanticen la fiabilidad y entrega oportuna de la información. Para obtener sistemas que mejoran la calidad de vida, minimizar el riesgo de accidente, mejorar la producción en actividades humanas y ampliar el dominio del conocimiento.

La labor de la ingeniería tiene como propósito mejorar la interacción entre los sistemas y el hombre, buscando que los dispositivos sean cada vez más ergonómicos, menos invasivos y reduciendo su tiempo de respuesta.

Este tipo de proyectos exigen de un trabajo interdisciplinario desde el análisis electrónico del servomecanismo, pasando por la ingeniería de software para desarrollar la aplicación que administre los datos obtenidos hasta la simulación desarrollada en entornos de realidad virtual.

Los desastres naturales acaecidos recientemente, por ejemplo el terremoto de magnitud 7.0 que azotó a Haití el 12 de enero de 2010 con un resultado trágico 250.000 muertos

[1], o el terremoto y tsunami sucedido el 11 de marzo en Japón [2], son dos escenarios que muestran la urgente necesidad de acompañar, ayudar o incluso sustituir al hombre en las tareas de salvamento bajo situaciones extremas de peligro.

Esta situación se convierte en un reto para los profesionales en ingeniería ya que aún no se cuenta con los mecanismos tecnológicos para salvaguardar vidas ante una calamidad. Además de las anteriores características, la exploración de planetas, minas y mundo submarino, despertaron el interés de la humanidad, por lo que se hace necesario desarrollar soluciones tecnológicas, para que el hombre pueda cumplir tales tareas.

Precisamente, la realidad virtual da respuesta a estos interrogantes por cuanto puede emular el escenario de lugares remotos como operaciones médicas, militares, antiexplosivos y otras de alto riesgo para el ser humano. Así, junto con los sistemas teleoperados, la realidad virtual inmersiva puede salir de un entorno computarizado para proveer interacción con el mundo real, logrando magníficos resultados.

Utilizar este tipo de tecnología para solucionar problemas en el ámbito regional y contribuir a la investigación de nuevas tecnologías, podría ser un factor clave en su desarrollo, de esta forma se puede facilitar a los miembros de la comunidad que aprovechen el avance tecnológico, la globalización, la programación de alto nivel y los paquetes de software especializado, junto con la formación de sus profesionales y los procesos investigativos adelantados por esta comunidad [3]. Asimismo contribuirían enormemente a evitar que la región no solo se caracterice por su riqueza en la parte agrícola, turismo, artesanías, entre otros, sino que además deje el rezago tecnológico y la falta de oportunidades de progreso científico.

El proyecto de investigación consiste en la búsqueda y selección adecuada y prueba de técnicas, herramientas, protocolos y demás recursos tecnológicos que permitan modelar el sistema teniendo en cuenta los recursos temporales, físicos, servicios humanos y el impacto social.

## II. DESCRIPCIÓN GENERAL DEL SISTEMA

### A. Módulos:

Básicamente el sistema se puede dividirse en 4 partes o módulos "fig2.ps" que se describen y articulan de la siguiente manera:

La primera parte es la encargada de recibir las instrucciones del usuario ya sea por teclado o haciendo la interpretación del lenguaje gesticular definido por el programador, mediante patrones, software de reconocimiento de imágenes captadas por una cámara.

La segunda corresponde al mundo virtual que sirve de apoyo para el control a distancia del servomecanismo este mundo puede ser utilizado como video juego, entono de simulación y todas las características derivadas del uso de los entornos virtuales, pero la de mayor interés es la posibilidad de evaluar el grado de inmersión.

La tercera es la interface de comunicación inalámbrica y los circuitos de acople tanto en la salida de datos del computador por el puerto USB. como en el circuito de potencia y control de los motores del robot.

La ultima describe el sistema mecánico que será el encargado de materializar las órdenes del usuario, dicho sistema tendrá una réplica virtual que servirá de avatar, para la navegación en el mundo virtual que se mencionó anteriormente.

### B. Diagrama general:

En la "fig1.tif" se puede observar una cara azul, esta representa el usuario del sistema. Las caras verdes son la capacidad de interacción del usuario en los dos mundos; En el mundo virtual mediante la representación del robot (avatar) y en el real en una locación distinta al centro de mando o control, por medio del servomecanismo.

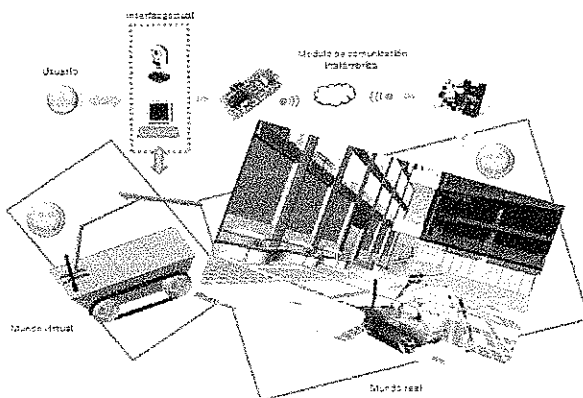


Fig. 1. Descripción el sistema Fuente: el autor.

La interfaz de comunicación (RX y TX) entre el usuario y el sistema se encuentra dentro de un cuadro con borde rojo, está compuesta básicamente por el teclado, la cámara como

elementos de emisión y la pantalla como recepción de datos por parte del usuario.

## III. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

El diseño del sistema y su implementación están soportados en metodologías que según un estudio presentan mayores ventajas o afinidad con relación a las que se utilizan en cada una de las áreas o módulos que se describen en la gráfica "fig2.ps" y que fueron mencionados con antelación.

Su ejecución es en paralelo pensada de manera modular y culmina con la integración de cada uno de los productos con el prototipo de prueba.

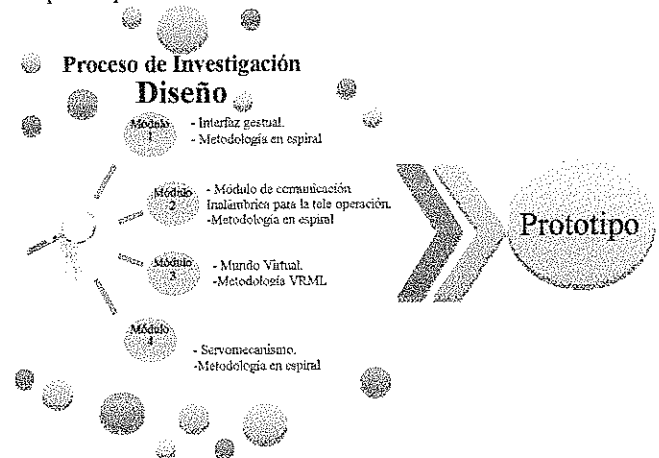


Fig.2. Describe la metodología del proceso de investigación en sus 4 fases o módulos.

### A. Interfaz gestual

Al tratarse de un producto de software o un código el modelo en espiral o metodología en espiral "fig3.ps" se utiliza en este módulo y también en la mayoría que los conforman el proyecto por las ventajas que se especifican a continuación.

Además de articular otros modelos como el iterativo y cascada de manera coherente, realiza el análisis del riesgo de forma explícita y clara, de manera que aumenta la posibilidad de controlarlo. Es decir que la característica iterativa es acorde con las exigencias de un prototipo y la incertidumbre sobre la estrategia a utilizar se minimiza con el análisis de riesgo.

Al Integrar el desarrollo con el mantenimiento es posible generar productos con características investigativas de manera constante.

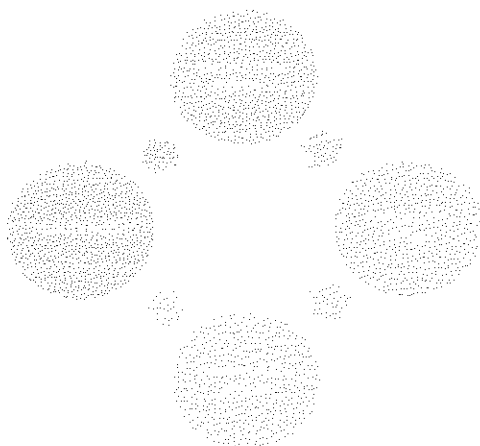


Fig. 3. Metodología en espiral: el autor basado en [11].

En la "fig3.tif" se muestran las etapas de la metodología en espiral de manera resumida:

El objetivo del módulo es generar las palabras de control, tanto para el mundo virtual como para el servomecanismo, que sean coherentes con los movimientos gesticulares a través de los algoritmos y técnicas comunes de imagenología.

El riesgo es la compatibilidad con el resto de herramientas, software o hardware involucradas en el sistema.

El desarrollo es coherente con los requisitos funcionales y no funcionales.

Finalmente la planificación es una especificación del trabajo futuro, las posibles mejoras del diseño, teniendo en cuenta los resultados obtenidos a partir de las etapas previas.

#### B. Módulo de comunicación

Con la metodología mencionada "fig3.ps" se explora la teoría de algunas de las tecnologías de radio frecuencia para llegar a determinar la solución.

Para el caso se estudian los módulos de radio frecuencia zigbee, ZigPro, z-wave, codificadores y decodificadores de control remoto integrados y se analizan las ventajas, desventajas; de adquisición, técnicas, comerciales y temporales para aplicar una de estas alternativas.

#### C. Mundo Virtual

Se utiliza la reingeniería para replicar el diseño mecánico, o para implementar la simulación con base en una metodología de diseño 3D para cualquiera de los casos se usa VRML efectivo "fig3.ps" y se genera un ambiente 3D con el modelo a escala del robot a utilizar "fig4.ps".



Fig. 3. Metodología Diseño VRML efectivo  
Fuente: el autor.

La metodología de diseño VRML sugiere 7 etapas, la ventaja de esta metodología es que orienta las aplicaciones de software a ser publicadas en la web, hoy en día las aplicaciones stand alone están mandadas a recoger.

El paso siguiente una vez se realice el diseño del escenario y el avatar animado, es construir a partir de estos productos un video juego. Aprovechando que los requerimientos ya han sido terminados en un gran porcentaje.

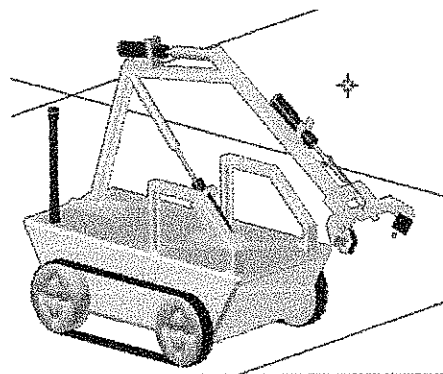


Fig. 4. Modelo 3D del robot. Fuente el autor.

El robot 3D cuenta con una plataforma móvil con posibilidad de translación y rotación en superficies planas. Esta sostiene un brazo mecánico con 3 grados de libertad.

#### D. Servomecanismo

El diseño de cada una de las piezas y sus planos se desarrollan con la herramienta de diseño 3D, teniendo en cuenta las medidas reales del robot.

Los circuitos directamente involucrados con su funcionamiento, como lo son sus drivers, los microcontroladores que hacen parte del control y la etapa de potencia se simulan en un paquete de software antes de su montaje.

La "fig5.tif" es el diagrama circuital de la simulación de los drivers de los motores y con la tabla de materiales utilizados se presentan a continuación:

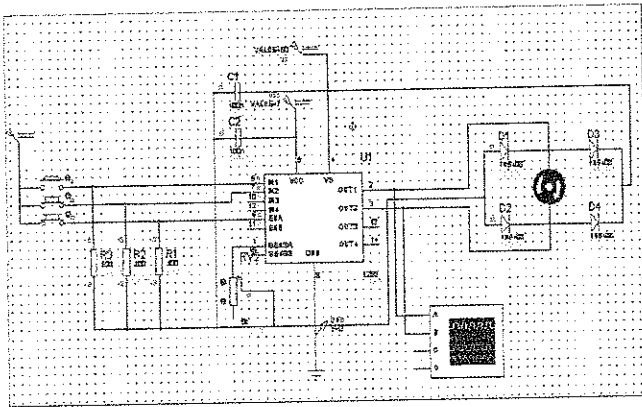


Fig. 5. Simulación drivers motores el robot. Fuentes: el autor.

El circuito mostrado en la figura anterior "fig3.ps", se utiliza para el control de giro de los motores y control de velocidad. En la parte izquierda se simulan las entradas digitales y la señal analógica controlada por el potenciómetro mediante la cual se realiza el control de velocidad. En la derecha se ilustra la etapa de potencia y el motor.

TABLA I  
COMPONENTES CI

NOMBRE	DESCRIPCIÓN	SIMBOLOGÍA	VALOR
Resistencias		R1, R2, R3	100K
Potenciómetro	Regulador de velocidad.	RV2	5K
Condensadores		C1	100n
		C2	100n
Diodos	Diodo de potencia IN5408	D1, D2, D3, D4	3A
Interruptores	Simulan la entrada de control.	Ninguna	No aplica.
Fuentes	Potencia		50V
	Control		7V
Driver	CI Puente H para el control de motores y control de velocidad	L298	No aplica.

Las unidades siglas correspondientes son; V = voltios, A = Amperios, n = Nanofaradios; K = kilo ohmios CI = Circuito Integrado.

Fuente: el autor

El modelo mecánico "fig6.ps", tiene características similares al diseño 3D "fig4.ps", trabaja superficies planas, tiene 3 grados de libertad, una piza y 2 articulaciones que utilizan un pistón mecánico.

El pistón transforma la rotación del motor mediante un tubo roscado (espárrago) a desplazamiento lineal por medio de una arandela. La pinza que tiene el mismo sistema con la variante que el desplazamiento lineal de la arandela, se usa para

desplazar las palancas que permiten su apertura y cierre. Para cualquiera de los casos es de mayor importancia la velocidad de rotación que el torque.

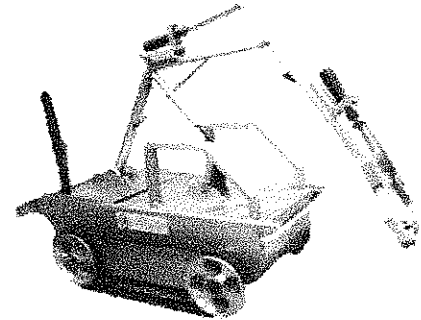


Fig. 6. Imagen del robot real. Fuente el autor.

El interior de la plataforma móvil, sirve para alojar los motores, circuitos de potencia, comunicación inalámbrica, control y baterías.

#### IV. CONCLUSIONES Y TRABAJO FUTURO

El sistema descrito anteriormente corresponde a un sistema de control a lazo abierto, este tipo de sistemas no nos dan la fiabilidad de que la señal de control llegue y actúe de acuerdo a lo que queremos o la referencia, para optimizar al sistema es necesario que se efectúe un sistema de control en doble vía que me permita hacer corrección y precisar el control.

El desarrollo en áreas como la telemedicina, sistemas de diagnóstico para la rehabilitación, mundos virtuales, videojuegos, robótica y los aportes al mejoramiento de la misma en pro del beneficio humano en el mundo, son áreas de investigación adelantadas por países desarrollados, a pesar que en los países en vía de desarrollo se cuenta con el talento humano y las necesidades para aplicar estas tecnologías a la propia región.

A nivel comercial, existe una gran demanda de dispositivos de realidad virtual inmersiva, un ejemplo de ello son los videojuegos, que se apoyan en software de reconocimiento de imágenes para mejorar sus productos, hacerlos más reales y ergonómicos. "La corporación Microsoft vendió en 4 meses más de 10 millones de controladores Kinect para la consola Xbox, que permiten a los usuarios jugar directamente sin mandos. Estos resultados permitieron incluir el aparato en el Libro Guinness de los récords como el dispositivo electrónico de consumo con mayor velocidad de ventas" [10].

La Facultad de Ingeniería Electrónica de la Universidad Santo Tomás de Tunja ha adelantado investigación cerca del

procesamiento de imagen sí como también prácticas de control de los servomecanismos. El software de reconocimiento de imagen y los servomecanismos permiten en conjunto, desarrollar investigación que genera aportes significativos, coherentes con las nuevas tecnologías y las necesidades actuales.

Por su parte la Facultad de Ingeniería de Sistemas de la misma universidad tiene experiencia en la creación de mundos virtuales, que junto con la tecnología mencionada aportan experiencia para implementar sistemas de tele presencia y juegos de realidad virtual inmersiva.

Aprovechando las fortalezas y los espacios mencionados anteriormente es pertinente desarrollar un prototipo que contribuya al enriquecimiento de esta área de investigación, a la creación de grupos de investigación interdisciplinarios y que contribuya al proceso de formación académico.

## V. REFERENCIAS

- [1]. El cuarto terremoto del mundo. CORDEIRO, JOSÉ LUIS. lunes 21 de marzo de 2011, EL UNIVERSAL.
- [2]. En vivo. ROBOTS ACUDEN EN AYUDA DE LAS VÍCTIMAS EN JAPÓN. [En línea] 17 de 03 de 2010. [Citado el: 21 de 03 de 2010.] [http://actualidad.rt.com/ciencia\\_y\\_tecnica/electronica\\_tecnologia/issue\\_21767.html](http://actualidad.rt.com/ciencia_y_tecnica/electronica_tecnologia/issue_21767.html).
- [3]. scienti. Plataforma ScienTI - Colombia. [En línea] <http://www.colciencias.gov.co/scienti>
- [4]. Tratamiento digital de imágenes. [En línea] [Citado el: 26 de 05 de 2010.] <http://www.grimaldos.es/cursos/imagdig/tipos.html>.
- [5]. González, Rafael C y Woods, Richard E. Tratamiento digital de imágenes. s.l. : Addison wesley/ Diaz de santos.
- [6]. Jairo. <http://www.teenlives.com/un-robot-que-utiliza-mundos-virtuales/>. Un robot que utiliza mundos virtuales. [En línea] Septiembre de 2009.
- [7]. Pérez, David García. Segmentación Basada en Bordes. [En línea] [Citado el: 27 de 06 de 2010.] <http://eva1.dec.usc.es/~antonio/docencia/2005tcj/practica/apuntes-practica-5.pdf>.
- [8]. Libreta para el manejo y configuración de dispositivos de realidad virtual. [aut. libro] Carlos Eric Mahecha Roa Rubén Darío García Castillo.
- [9]. En vivo. KINECT PARA XBOX, EL PLUSMARQUISTA EN VELOCIDAD DE VENTAS. [En línea] 11 de 03 de 2010. [Citado el: 21 de 03 de 2010.] [http://actualidad.rt.com/ciencia\\_y\\_tecnica/electronica\\_tecnologia/issue\\_21589.html](http://actualidad.rt.com/ciencia_y_tecnica/electronica_tecnologia/issue_21589.html).
- [10] Pressman, Roger S. Ingeniería de software Un enfoque práctico. Sexta Edición. Mexico : Mc Graw Hill, 2008.

**First A. Author.** Nacido en Tunja ciudad del departamento de Boyacá (Colombia) el 2 de enero de 1985. Este autor es egresado del Colegio Salesiano Maldonado de Tunja en el año 2002. Actualmente se encuentra en trabajo de grado para optar el título de Ingeniero Electrónico e Ingeniero de Sistemas en la Universidad Santo Tomás de Aquino. Es miembro auxiliar de del grupo de investigación basado en redes de aprendizaje sobre nuevas tecnologías (GIBRANT). En el área de realidad virtual.

## 12. ASESORES

**ING. JUAN FRANCISCO MENDOZA MORENO**  
Decano Académico de la facultad de Ingeniería de sistemas



**Ingeniero de Sistemas**  
**Especialista en Diseño y Construcción de Soluciones Telemáticas**  
**Especialista en Gerencia de Instituciones de Educación Superior**  
**Candidato a Ms. en Software Libre**  
**Correo electrónico: [jmendoza@ustatunja.edu.co](mailto:jmendoza@ustatunja.edu.co)**

**ING. JOSÉ RICARDO CASALLAS GUTIERREZ**  
Decano Académico de la facultad de Ingeniería Electrónica



**Correo Electrónico: [jcasallas@ustatunja.edu.co](mailto:jcasallas@ustatunja.edu.co)**

**Contacto: 8 – 7440404 Ext. 31050 -31051**

**Estudios:** Ingeniero Electrónico – Universidad Santo Tomás – Bogotá, Magister en Ingeniería Electrónica y de Computadores – Universidad de los Andes, Especialista en Telemática - Uniboyacá

**Trayectoria:** 15 años de experiencia en los sectores público, privado y empresarial; diseño de soluciones electrónicas; diseño, administración, operación y mantenimiento de redes de telecomunicaciones, profesor de la Facultad de Ingeniería Electrónica Usta-Tunja desde 1999.

**ING. LUIS FREDY SOSA QUINTERO**

Docente



**Correo Electrónico:** [lsosa@ustatunja.edu.co](mailto:lsosa@ustatunja.edu.co)

**Contacto:** 8 – 744 0404 Ext. 31050

**Estudios:** Ingeniero Electrónico, Especialista en Instrumentación médica, Especialista en Gerencia de Proyectos, Especialista en Gerencia de instituciones de educación superior, Magister en educación, aspirante a Doctor en Ciencias de la Educación.

**Trayectoria:** Asesor técnico en la Comisión Reguladora de Telecomunicaciones, Representante de los docentes al consejo académico General Universidad Santo Tomás. Representante de los decanos al consejo Superior Universidad Santo Tomás. Decano académico Facultad de Ingeniería Electrónica.

**JORGE GABRIEL HOYOS PINEDA**

Docente lector



**Ingeniero de Sistemas**

**Ms. en Ciencias de la Información y la Comunicaciones**

**Correo electrónico:** [jhoyos@ustatunja.edu.co](mailto:jhoyos@ustatunja.edu.co)

**ING. EDGAR ANDRÉS GUTIERREZ CACERES**

Docente Tutor

**Correo Electrónico:** [walvarez@ustatunja.edu.co](mailto:walvarez@ustatunja.edu.co)

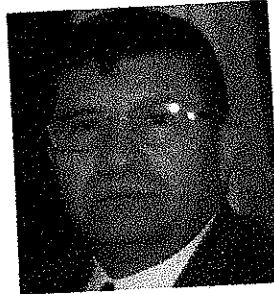
**Contacto:** 8 – 744 04 04 Ext.31050

**Estudios:** Ingeniero Electrónico, Especialista en Instrumentación Electrónica.

**Trayectoria:** Experiencia en el sector de de las telecomunicaciones con el Departamento de Policía oficina de Telemática Boyacá. Laboratorista y Docente Facultad de Ingeniería Electrónica.

**Área de interés:** Sistemas de visión artificial, Procesamiento y tratamiento de imágenes, Instrumentación Electrónica industrial, Sistemas de Programación, Telecomunicaciones

**ING. OSCAR EDUARDO UMAÑA MENDEZ**  
Docente



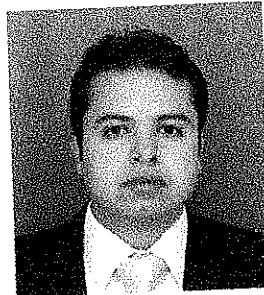
**Correo Electrónico:** [oumana@ustatunja.edu.co](mailto:oumana@ustatunja.edu.co)

**Contacto:** 8 – 7440404 Ext. 31050

**Estudios:** Ingeniero Electrónico, Magister en Administración de Empresas, Especialista en Instrumentación Electrónica.

**Trayectoria:** Experiencia en el sector siderúrgico, mantenimiento eléctrico-electrónico en instrumentación, Docente en Educación superior, Coordinador laboratorios Facultad de Ingeniería Electrónica.

**ING. FABIÁN ROLANDO JIMÉNEZ LÓPEZ**  
Docente



**Correo Electrónico:** [fjimenez@ustatunja.edu.co](mailto:fjimenez@ustatunja.edu.co)

**Contacto:** 8 – 7440404 Ext. 31050

**Estudios:** Ingeniero Electrónico – UPTC Sogamoso, Especialista en Automatización Industrial – UPTC Sogamoso, MSc en Ingeniería – Automatización y Control – UNAL Bogotá.

**Trayectoria:** Docente Facultad de Ingeniería Electrónica, Coordinador de Área de Control y Automatización, Coordinador de Investigaciones de Facultad, Perteneciente al Grupo de Investigación en Instrumentación y Control GINSCON.

**Áreas de Interés:** Control, Automatización, Instrumentación, DSPs y PDS.

**LUZ SANTAMARÍA GRANADOS**  
Docente directora



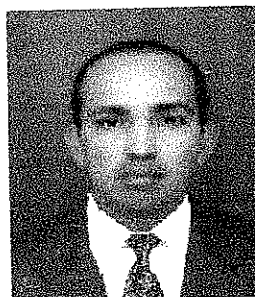
**Ingeniera de Sistemas**  
**Especialista en Telemática**  
**Ms. en Ciencias de la Información y las Comunicaciones**  
**Correo electrónico: [lsantamaria@ustatunja.edu.co](mailto:lsantamaria@ustatunja.edu.co)**

**JULIÁN ALBERTO MONSALVE PULIDO**  
Docente lector



**Ingeniero de Sistemas**  
**Msc. en Software Libre Especialidad en Desarrollo de Software**  
**Correo electrónico: [jmonsalve@ustatunja.edu.co](mailto:jmonsalve@ustatunja.edu.co)**

**ING. ADOLFO AVILA BARON**  
Docente lector



**Correo Electrónico: [avilaba@ustatunja.edu.co](mailto:avilaba@ustatunja.edu.co)**  
**Contacto: 8 – 7440404 Ext. 31050**

**Estudios: Ingeniero Electrónico, Especialista en Automatización Industrial, Especialista en Administración de Empresas, Magister en Ingeniería - Automatización**  
**Trayectoria: Once años de experiencia en docencia Universitaria.**

**FREDY ANDRÉS APONTE NOVOA**

Docente



**Ingeniero de Sistemas**

**Candidato a Ms. en Software Libre**

**Correo electrónico:** [faponte@ustatunja.edu.co](mailto:faponte@ustatunja.edu.co)

**ING. CAMILO ERNESTO PARDO BEAINY**

Docente Lector



**Correo Electrónico:** [cpardo@ustatunja.edu.co](mailto:cpardo@ustatunja.edu.co)

**Contacto:** 8 – 7440404 Ext. 31050

**Estudios:** Ingeniero Electrónico – Universidad Santo Tomás – Especialista en Instrumentación Electrónica – Universidad Santo Tomás

**Trayectoria:** Participación en Proyectos y Licitaciones del sector energético, de Ingeniería y Construcción Industrial – Abengoa, Madrid, España. Docente Universitario en las áreas de Comunicaciones, Técnicas Digitales, Teoría de Circuitos, y Tratamiento Digital de Señales - Facultad de Ingeniería Electrónica, USTA - Tunja.

**Áreas de Interés:** Sistemas SCADA, Telecomunicaciones, Tratamiento de Señales, DSPs y PDS, Redes Inalámbricas, Redes de Convergencia, Sistemas de Control, Automatización e Instrumentación de Procesos.