



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTOR: VEINTIMILLA CAMPOVERDE, MANUEL ALEJANDRO

TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL PARA EL ROBOT MÓVIL P3-AT POR MEDIO DE GESTOS MANUALES

DIECTOR: ING. GORDILLO, RODOLFO

CODIRECTOR: ING. AYALA, PAÚL

SANGOLQUÍ, MAYO 2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y

CONTROL

CERTIFICADO

Ing. Rodolfo Gordillo

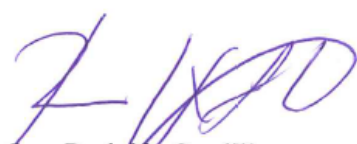
Ing. Paúl Ayala

CERTIFICAN

Que el trabajo titulado “Diseño e implementación de un sistema de control para el robot móvil P3-AT por medio de gestos manuales”, realizado por Manuel Alejandro Veintimilla Campoverde, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas ESPE.

Debido a que se trata de un trabajo de investigación recomiendan su publicación. El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a Manuel Alejandro Veintimilla Campoverde que lo entregue al Ingeniero Luis Orozco, en su calidad de Coordinador de la Carrera.

Sangolquí, 14 de Mayo de 2015



Ing. Rodolfo Gordillo
DIRECTOR



Ing. Paúl Ayala
CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE**INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y
CONTROL****DECLARACIÓN DE RESPONSABILIDAD****MANUEL ALENAJANDRO VEINTIMILLA CAMPOVERDE****DECLARO QUE:**

El proyecto de grado denominado “Diseño e implementación de un sistema de control para el robot móvil P3-AT por medio de gestos manuales”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie, de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 14 de Mayo de 2015



Manuel Alejandro Veintimilla Campoverde

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Yo, Manuel Alejandro Veintimilla Campoverde

Autorizo a la Universidad de las Fuerzas Armadas ESPE, la publicación en la biblioteca virtual de la Institución del trabajo “Diseño e implementación de un sistema de control para el robot móvil P3-AT por medio de gestos manuales”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Sangolquí, 14 de Mayo de 2015



Manuel Alejandro Veintimilla Campoverde

DEDICATORIA

A mis padres Manuel (+) y Yolanda,
a mis hermanos Gabriela Yolanda, Ana Belén y Pablo Andrés a mis sobrinos
Valentina y Hunter Armando.

AGRADECIMIENTO

Gracias a Dios Todopoderoso por bendecirme con una hermosa familia que me ha apoyado en todo momento.

A la Universidad de las Fuerzas Armadas ESPE, Departamento de Eléctrica y Electrónica, a todos los docentes del departamento y en especial a los Ingenieros Rodolfo Gordillo y Paúl Ayala, quienes me ayudaron en el desarrollo del presente trabajo.

También agradezco a mis mejores amigos David Allán y Fabián Izquierdo quienes me han brindado su apoyo incondicional, desde el inicio de nuestra amistad.

ÍNDICE DE CONTENIDOS

CERTIFICADO	II
DECLARACIÓN DE RESPONSABILIDAD	III
AUTORIZACIÓN.....	IV
DEDICATORIA	II
AGRADECIMIENTO	VI
RESUMEN.....	XIII
ABSTRACT	XIV
CAPÍTULO I.....	1
INTRODUCCIÓN	1
1.1 Antecedentes	1
1.2 Justificación e importancia	2
1.3 Alcance del proyecto	2
1.4 Objetivos	3
1.4.1 General	3
1.4.2 Específicos	3
1.5 Descripción general del proyecto.....	3
CAPÍTULO II	5
MARCO TEÓRICO	5
2.2 Robots móviles.....	6
2.3 Entorno de operación.....	7

2.4	Aplicaciones de los robots móviles.....	9
2.4.1	Aplicaciones industriales	9
2.4.2	Aplicaciones militares	11
2.4.3	Aplicaciones de servicio	13
2.4.4	Aplicaciones de investigación y desarrollo	15
2.5	Sensor de profundidad Kinect	16
2.5.1	Entorno de desarrollo	17
2.5.2	Fundamentos de cámara.....	18
2.5.3	Emisor y receptor infrarrojo (IR).....	19
2.5.4	Funciones de audio	22
2.5.5	Funcionamiento de “Skeleton Traking”	23
CAPÍTULO III.....		28
DISEÑO DEL SISTEMA		28
3.1	Arquitectura del sistema.....	28
3.1.1	Cliente.....	29
3.1.2	Servidor	34
3.2	Descripción de hardware y requerimientos	38
3.2.1	Comportamiento de robot	39
3.2.1.1	Comportamiento paseo	39
3.2.1.2	Comportamiento distancia	40
3.2.1.3	Comportamiento conducir	40
3.2.2	Movimientos y gestos admitidos.....	41
3.3	Software de programación	41
3.3.1	ARIA (Advanced Robotics Interface for Applications)	42
3.3.2	Kit de desarrollo de software para Windows(SDK).....	43
3.4	Especificaciones de diseño	43
3.4.1	Diseño de la red de comunicación	43
CAPÍTULO IV		51
ANÁLISIS Y PRUEBAS DEL SISTEMA		51
4.1	Pruebas de comunicación	51
4.2	Adquisición de datos	52

CAPÍTULO V.....	58
CONCLUSIONES Y RECOMENDACIONES.....	58
5.1 Conclusiones	58
5.2 Recomendaciones	59
REFERENCIAS BIBLIOGRÁFICAS	61

ÍNDICE DE FIGURAS

Figura 2. 1	Entrono de operación.....	7
Figura 2. 2	Robots móviles en entornos interno y externo	8
Figura 2. 3	Entornos según con objetos alrededor del robot.....	8
Figura 2. 4	Inspector Bot robot que inspecciona tuberías.....	10
Figura 2. 5	Robot de transporte de carga	10
Figura 2. 6	Tractor autónomo y cosechadora autónoma.....	11
Figura 2. 7	Robot Goliat (modelo SdKfz 302).....	12
Figura 2. 8	Robot soldado y avión no tripulado.....	13
Figura 2. 9	Robot Rudy del centro médico de la Universidad de California.....	14
Figura 2. 10	Robot QRIO, robot de entretenimiento	14
Figura 2. 11	Robot humanoide ASIMO.....	15
Figura 2. 12	Robot araña, se moviliza por paredes y realiza vigilancia	15
Figura 2. 13	Componentes del sensor Kinect	18
Figura 2. 14	Rango de movimiento horizontal y vertical	19
Figura 2. 15	Fotografía con resolución de 640 x 480 píxeles.....	19
Figura 2. 16	Patrón de " puntos aleatorios"	20
Figura 2. 17	Emisor y receptor infrarrojo	20
Figura 2. 18	Chip PrimeSense.....	21
Figura 2. 19	Procesamientos de datos de profundidad.....	22
Figura 2. 20	Procesamiento de datos de profundidad.	22
Figura 2. 21	Objeto "cuerpo humano"	24
Figura 2. 22	Segmentación del cuerpo humano	25
Figura 2. 23	Puntos en las articulaciones.....	26
Figura 2. 24	Tres vistas calculadas por el Kinect	26
Figura 3. 1	Robot propuesto, Pioneer P3-AT	28
Figura 3. 2	Arquitectura de Comunicación.....	29
Figura 3. 3	Transmisión de datos al cliente	30
Figura 3. 4	Diagrama del cliente.....	31
Figura 3. 5	Conexión con servidor.....	32
Figura 3. 6	Envío de modos de operación.....	32
Figura 3. 7	Proceso asíncrono para el envío de datos	33
Figura 3. 8	Comunicación entre el servidor y el robot.....	34
Figura 3. 9	Computadora y controles del P3-AT	35
Figura 3. 10	Diagrama del servidor	36
Figura 3. 11	Subproceso de inicialización	37
Figura 3. 12	Subproceso comportamientos.....	38
Figura 3. 13	Subproceso conducir	40
Figura 3. 14	Arquitectura ARIA	42

Figura 3. 15	Red de comunicación	44
Figura 3. 16	Inicio de conexión	45
Figura 3. 17	Interfaz de comportamientos	46
Figura 3. 18	Dirección IP y puerto de comunicación	46
Figura 3. 19	Mensaje de bienvenida	47
Figura 3. 20	Imagen a color obtenida por el sensor	47
Figura 3. 21	Imagen de visualización de articulaciones y ángulos	48
Figura 3. 22	Panel de Comportamientos	48
Figura 3. 23	Botones salir y detener	49
Figura 3. 24	Mensaje de falló la conexión con el sensor	50
Figura 4. 1	Servidor conectado	51
Figura 4. 2	Mensaje de bienvenida del servidor	52
Figura 4. 3	Línea trazada entre las manos.....	53
Figura 4. 4	Angulo entre las manos	53
Figura 4. 5	Ángulos enviados cada 330 ms.	55
Figura 4. 6	Ángulos para cada periodo	57
Figura 4. 7	Adquisición de Datos del servidor	57

ÍNDICE DE TABLAS

Tabla 3. 1 Descripción de Requerimientos	39
Tabla 3. 2 Descripción de modo de operación	49
Tabla 4. 1 Ángulos enviados al servidor	54
Tabla 4. 2 Tiempo de envío de datos	56

RESUMEN

El elevado desarrollo de la robótica en la actualidad permite al ser humano, interactuar mucho más con máquinas de forma autónoma, por ello se establece un sistema donde el hombre aplique la interfaz natural de usuario dando un paso adelante en la comunicación hombre-máquina. Se construyó una relación entre un usuario y un robot móvil a través de una red cliente servidor inalámbrica, con lo cual el robot P3-AT, se mueve con los gestos indicados por la persona ya sea que se necesite explorar o simplemente trasladar el robot. El usuario debe desarrollar la destreza para manipular el robot siguiendo el manual de usuario e iniciar con el simulador, así con este procedimiento se evita tener errores desde la conexión con el robot hasta la ejecución de los comportamientos. Los comportamientos se ejecutan con botones en la interfaz del cliente, y cada uno de forma exclusiva. Los resultados muestran que al hacer movimientos pausados y continuos el robot se puede conducir sin problemas, pues la programación valida y verifica que la información recibida sea correcta, sin embargo el usuario debe conocer los movimientos admitidos antes de manipular el robot.

Palabras Clave:

- Hombre-máquina
- Inalámbrica
- Interfaz
- P3-AT
- Simulador

ABSTRACT

The high development of robotics now allows humans to interacting with machines more independently, so a system where men apply the natural user interface taking a step forward in human-machine communication is established. A relationship between a user and a mobile robot through a client-server network with which the P3-AT robot moves with gestures indicated by the person, whether it is needed to explore or simply move the robot was built. User must develop the ability to manipulate the robot following the manual and start the simulator, so this procedure avoids having errors from the connection to the robot to the execution of behaviors. Behaviors are executed with buttons on the client interface, each uniquely. The results show that by making unhurried and continuous movements the robot can be driven smoothly, because the program validates and verifies that the information received is correct, but the user must know the movements admitted before handling the robot.

Keywords:

- **Human-machine**
- **Interface**
- **P3-AT**
- **Simulator**
- **Wireless**

CAPÍTULO I

INTRODUCCIÓN

1.1 Antecedentes

La Robótica, es hoy un área de extensa investigación, ocupando una importancia relevante en el campo industrial, y que concierne a los procesos de innovación productiva y modernización. Teniendo en cuenta las nuevas tecnologías que se encuentran en la actualidad, ya sean en control automático como en comunicaciones.

Siendo parte de la electrónica y automatización, la robótica tiene un amplio espacio de investigación, cuya importancia es relevante tanto en el ámbito industrial como en los procesos de innovación productiva y modernización. Sustituye las tareas peligrosas por otras que presentan condiciones más ventajosas en cuanto a salud, seguridad, flexibilidad, calidad y productividad.

Aunque la robótica hoy en día ha presentado muchos avances, aún no cuenta con un nivel de desarrollo elevado, en especial la robótica autónoma, donde es realmente difícil realizar movimientos que son cotidianos para el hombre, como correr, caminar y tomar objetos.

Se plantea entonces el diseño e implementación de un sistema de telecontrol del robot móvil P3-AT a través de gestos manuales, los cuales serán captados por medio del sensor Kinect desarrollado por Microsoft para su plataforma de juegos Xbox.

1.2 Justificación e importancia

La ejecución del Proyecto se justifica por las siguientes razones:

- Porque permite la apertura a nuevos campos de investigación en robótica y telecontrol, implementando nuevas y actuales tecnologías.
- Por la aplicación en sí del proyecto, ya que no habrá la necesidad de controlar el robot a través de teclado o joystick.
- Porque se podrá utilizar una tecnología para la identificación procesamiento de gestos corporales, debido a su bajo costo.
- Porque permite conocer e investigar en un lenguaje de programación no utilizado convencionalmente como C#, para la programación del sensor.
- El aporte técnico de este proyecto se presta para dar servicio a aplicaciones militares y de investigación en zonas de difícil acceso para el ser humano.
- Con este aporte, se podrá dar un servicio más técnico al reconocimiento de movimientos corporales tanto para el uso industrial, militar e incluso educativo, ya que se podría aplicar tanto para robots móviles como también en todo el campo comprendido por la robótica.

1.3 Alcance del proyecto

Se investigará y documentará todo lo concerniente a la programación del robot móvil y del sensor Kinect, para este último, se utilizará su propio software de licencia libre, el cual ha sido desarrollado para el lenguaje de programación C#.

Se desarrollará una red cliente-servidor inalámbrica para la comunicación entre el robot y la computadora que procesará los gestos y movimientos corporales receptados a través del sensor Kinect, el servidor será implementado en el robot atendiendo los requerimientos del usuario.

La aplicación del proyecto permitirá a un usuario controlar el movimiento del robot a través de una interfaz, sin la necesidad de usar un teclado o joystick, dentro de ésta se procesaran diversos comportamientos como el avance, retroceso, giros y parada del robot.

Se realizaran pruebas tanto de comunicación como del control para verificar la viabilidad del sistema.

1.4 Objetivos

1.4.1 General

Diseñar un sistema de control para el robot móvil P3-AT por medio de gestos manuales para analizar la utilización de sensores de profundidad aplicados en la robótica.

1.4.2 Específicos

- Investigar y documentar los fundamentos y funciones que presenta la cámara RGB y el sensor de profundidad.
- Diseñar una red cliente-servidor para el envío y recepción de datos entre el robot móvil y la computadora.
- Desarrollar un manual de usuario para facilitar el uso del prototipo.
- Realizar las respectivas pruebas de comunicación, envío y recepción de datos para el sistema.

1.5 Descripción general del proyecto

El siguiente trabajo sirve de base para proyectos a futuro, debido a su configuración se puede utilizar en actividades exploratorias y militares más concretamente; está diseñado para que a través de una red informática cliente-servidor el robot móvil interactúe con una persona quien va a comandar los movimientos del mismo.

Dentro de la red el cliente estará conectado a un sensor, que permitirá receptar los distintos gestos de una persona, los mismos que se transmitirán hacia el robot,

no solo se enviaran estos gestos corporales sino que también se enviara diferentes comandos para realizar tareas diversas como eventos de parada, o avanzar a cierta posición ya sea de frente o en sentido opuesto, todas estos comportamientos serán descritos en detalle en la configuración de comportamientos del robot.

En concreto la función principal del proyecto será la de manejar el robot móvil por medio de gestos realizados con las manos, dentro de esta área estará comprendido el avanzar de frente, parada, giros y retroceder.

Cada conducta del robot, desde el manejo por gestos manuales así como las introducidas por comandos deberán ser exclusivas, es decir, no se podrán realizar dos comportamientos a la vez para evitar errores graves, que afectarían la funcionalidad del robot o incluso la de todo el proyecto.

El servidor finalmente será el encargado de receptar todos los mensajes enviados por el cliente a través de la red, principalmente debe evitar que efectúen errores en la ejecución de las órdenes del cliente y también permitir la fluidez en la manipulación del robot, impidiendo los choques o roces con objetos en el campo.

CAPÍTULO II

MARCO TEÓRICO

2.1 Robots móviles y sensor de profundidad

En la actualidad cada vez nos vemos sumergidos en una ola de avances tecnológicos, mientras algunos tienen grandes aplicaciones en la ciencia y tecnología, otros trascienden de una manera diferente en la vida cotidiana, si se habla de los sensores, cada cierto tiempo se encuentra más utilidades en un solo dispositivo, pues ya no es nuevo que estos instrumentos hoy por hoy incluyan cámara o controladores avanzados.

La factibilidad del uso de un sensor de profundidad, es la apertura a una gran cantidad de posibilidades para el estudio de esta tecnología en la electrónica, ya que nos permite realizar aplicaciones como el mapeo en tercera dimensión, captura de gestos corporales, interacción virtual directa entre máquina y usuario; estos aparatos permiten que nos comuniquemos con una computadora directamente a través de nuestras expresiones corporales, por ende se hace más sencillo la manipulación de máquinas más complejas que hoy se van haciendo cada vez mas de uso común como robots móviles y antropomórficos, por ejemplo en el campo automovilístico ya se están usando para hacer de los carros, medios de transporte totalmente autónomos, es decir se conducen por sí solos.

Esta época de desarrollo científico lleva a reunir los conocimientos ya adquiridos, para resolver las problemáticas más complejas de la actualidad, con lo que se plantea ¿Como se relacionan los robots con los sensores de profundidad?, pues el objetivo general de este trabajo ayuda a disipar esta interrogante,

permitiendo la manipulación de robots móviles por medio de gestos corporales, específicamente gestos manuales, evitando el envío de comandos e instrucciones para hacer que estos dispositivos cumplan las órdenes del hombre, incluso no se limita a estos instrumentos móviles sino más bien abre un campo de investigación para todo tipo de robots y tecnologías afines.

2.2 Robots móviles

Los robots móviles son dispositivos de transporte automático, es una plataforma mecánica que está dotada de un sistema de locomoción capaz de navegar a través de un determinado ambiente de trabajo, dotado de cierto nivel de autonomía para su desplazamiento portando cargas, sus aplicaciones pueden ser muy variadas y siempre están relacionadas con tareas que normalmente son riesgosas o nocivas para la salud humana, ya sea en el campo de agricultura, transporte o exploraciones solitarias o colectivas, junto con otros vehículos. [1]

La definición de robot móvil está ligada a esa capacidad para alcanzar uno o varios objetivos, con una pequeña intervención del ser humano; en referencia a la denominación de vehículo auto guiado se establece a las estructuras móviles limitadas a seguir caminos o vías preestablecidas. [2]

Debido a que los robots móviles tienen que operar en grandes ambientes no estructurados se ven inmersos en incertidumbres en cuanto a posición e identificación respecta, es decir, que si un robot tiene que trasladarse de un lugar a otro no se puede precisar la trayectoria por ejemplo, pues esto no sucede en el caso de un manipulador industrial que pueden operar hasta en centésimas de milímetro.

2.3 Entorno de operación

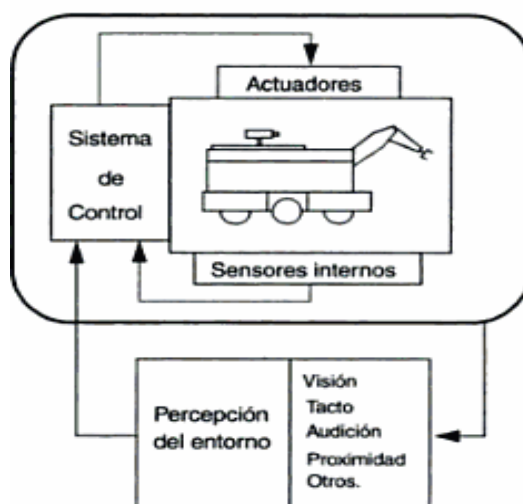


Figura 2. 1 Entorno de operación, tomado de [3]

Un robot móvil tiene la capacidad de relacionarse con su entorno por medio de su sistema sensorial, hoy en día podemos explorar una gran cantidad de tecnologías sensoriales como odometría, ultrasonido, infrarrojo, entre otro, la figura 2.1 muestra como el sistema de percepción sensorial permite que el robot adapte automáticamente su comportamiento en función de la variaciones que se producen en su entorno, para esto los robots usan bucles de realimentación del sensorial del entorno. [3]

La capacidad para tomar acciones y así lograr uno o varios objetivos es parte integral en la operación de un robot móvil, y su comportamiento dependerá de la información entregada por el sistema de sensores. La característica del entorno de operación radica en que restringe al robot en cuanto a su alcance y objetivos.

Si se toma en cuenta el área de trabajo de los robots, estos entornos pueden ser externos o internos. Es interno cuando es el área de trabajo está definida por paredes y techos, además de tener iluminación artificial principalmente. En cambio en un ambiente externo el área de trabajo no está estrictamente delimitada, y la iluminación generalmente es natural.



Figura 2. 2 Robots móviles en entornos interno y externo, tomado de [4] , [5].

El entorno del área de operación se ve afectado también por los objetos que se encuentran a su alrededor, si el robot está en un ambiente donde los objetos que lo rodean son estáticos, pues no cambian de posición ni de forma, lo que permite relacionarlos con figuras geométricas conocidas como cilindros, cubos, entre otros, por ello se hace más fácil la distinción de estos objetos. Por el contrario si nos encontramos en un ambiente donde los objetos no se pueden asociar por su forma o posición, es decir que cambian con el paso del tiempo, esto impedirá el rápido reconocimiento del entorno, haciendo impredecible la reacción de los robots.



Figura 2. 3 Entornos según con objetos alrededor del robot, tomado de [2].

2.4 Aplicaciones de los robots móviles

La robótica móvil representa un campo importante en la industria, pues a ha generado un significativo aporte en el ámbito de la industria manufacturera, en agroindustria y en el sector de servicio.

Las actuales investigaciones plantean el desarrollo de robots con más autonomía y mejores niveles de destreza, permitiendo tener sistemas de producción de gran nivel y automatizados, realizar tareas de servicio y de auxilio que podrían ser riesgosas para el ser humano. Si tomamos en cuenta el campo en el que se desempeña el robot móvil se pueden clasificar en distintos tipos.

2.4.1 Aplicaciones industriales

El requerimiento de robots industriales cada día va en aumento, con estos el sector industrial puede reducir sus costes de producción, aumento de productividad, mejoras en la calidad de producción y eliminación de condiciones peligrosas en el trabajo. Con estas inversiones tecnológicas los mercados son más competitivos, contribuyendo a que las empresas integren más y mejores tecnologías a los sistemas de producción.

La implementación de un robot en un ambiente industrial requiere de un detallado estudio previo, ya que es necesario analizar las ventajas e inconvenientes que conlleva la introducción de un robot en el proceso.

La inmensa mayoría de los robots industriales se componen de un brazo articulado, a través del cual desempeñan su tarea en una cadena de producción. Este tipo de robots no forma parte de este estudio pues se encuentran en otro ámbito de la robótica [6].

a. Limpieza industrial

- Inspección y limpieza de conductos
- Inspección y limpieza de barcos
- Limpieza de fachadas
- Limpieza de aviones



Figura 2. 4 Inspector Bot robot que inspecciona tuberías, tomado de [2].

b. Transporte de mercancías

- Vehículos de guiado automático (AGV)
- Vehículos guiados por láser (LGV)
- Sistema automatizado de transporte de materiales (AMTS)



Figura 2. 5 Robot de transporte de carga, tomado de [2]

c. Agricultura

- Tractor autónomo
- Cosechadora autónoma



Figura 2. 6 Tractor autónomo y cosechadora autónoma, tomado de [2].

2.4.2 Aplicaciones militares

En el sector militar la robótica móvil está en auge, pues su importancia se enfoca en que no se pierdan vidas en combate, esta idea se ha ido imponiendo con mayor presencia a partir de principios del siglo XX, donde se produjeron las peores guerras de la historia.

Actualmente la idea de infantería se va renovando continuamente, ya que no se enfatiza en la fuerza bruta de ataque con grandes ejércitos, sino que se utiliza grupos relativamente pequeños con objetivos específicos, sufriendo la menor cantidad de bajas.

Toda investigación militar se interesa en la robótica, con la finalidad de reemplazar la infantería en las tareas específicas en este campo de la milicia, y evitando poner en riesgo la vida de las tropas en actividades como espionaje desde aire, activación y desactivación de minas.

El principal interés en la robótica por parte del sector militar, se dirige a conseguir grandes avances en este sector, lo que repercutiría en un gran ahorro; en la parte económica en cuanto al bajo costo de mantenimiento de un ejército y por otro lado evitar pérdida de vidas humanas.

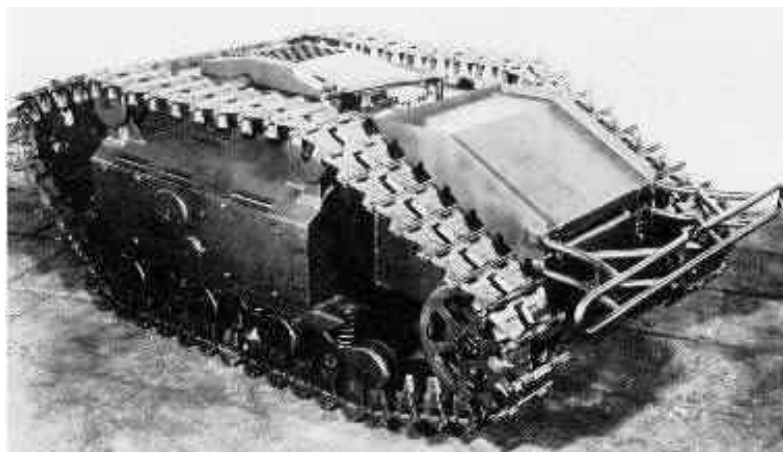


Figura 2. 7 Robot Goliat, tomado de [7]

A partir del éxito de estos robots, se fueron desarrollando nuevas ideas para aplicaciones militares, al mismo tiempo el sistema de guerra se iba adaptando en cada situación.

Uno de los referentes actuales de la robótica aplicada a la milicia, son las aeronaves no tripuladas, que en sus inicios eran usadas para la exploración y detección de objetivos, pero en la actualidad se utilizan aeronaves no tripuladas con gran control de posicionamiento, y con armamento integrado. Entre los robots militares más usados están:

- Robots Soldado
- Robots de reconocimiento aéreo
- Robot explorador o de reconocimiento terrestre
- Robots de desactivación de explosivos
- Robots suicidas o minas móviles.



Figura 2. 8 Robot soldado y avión no tripulado, tomado de [8], [9].

2.4.3 Aplicaciones de servicio

Los ayudantes mecánicos del hombre como los robots de servicio, están capacitados para razonar, ante un conjunto de situaciones o contexto. Este sector de servicios, sea personal o colectivo, es una de las áreas de aplicación más novedosa, es probable que en los próximos años este sector llegue a ser solicitado a un nivel comparable solo con el sector industrial actual [2].

Los robots incluidos en este tema son aquellos que realizan actividades al servicio de los humanos o para el mantenimiento de infraestructuras y equipos, excluidas las operaciones de fabricación. De manera más concreta, se puede indicar que los robots de servicio operan en sectores como: construcción, medicina, submarino, nuclear, limpieza, agricultura, doméstico y de oficina, militar y seguridad, ocio y entretenimiento.

Un robot de servicio puede operar parcial o totalmente con autonomía, realizando actividades útiles para el bienestar del ser humano, excluyendo las operaciones de manufactura.

En cuanto a la robótica de servicios, el estudio anual *WorldRobotics*, proporciona cifras a nivel mundial de ventas y expectativas para los próximos años. Si diferenciamos los robots de servicios, entre robots para uso profesional y robots para uso doméstico y entretenimiento, hasta finales de 2007 se han vendido cerca de 5,5 millones de robots de servicios. Tomando en cuenta que hasta hace

pocos años, el robot más conocido era el robot manipulador industrial que se utiliza en el campo de la manufactura, en el año 2007 el total de robots industriales operativos no llegaba al millón de unidades y hablando de robots instalados desde el año 1995 hasta ahora [10].

Los principales robots de servicio son:

- Medicina
- Domésticos y de oficina
- Entretenimiento
- Industria de construcción
- Servicios de comunidad



Figura 2. 9 Robot Rudy, tomado de [2].



Figura 2. 10 Robot QRIO, robot de entretenimiento, tomado de [2]

2.4.4 Aplicaciones de investigación y desarrollo

La investigación y desarrollo de la robótica ha ido creciendo de manera acelerada desde sus inicios a mediados del siglo XX, los robots ahora pueden realizar actividades más complejas sin problemas lo que implica una ventaja al invertir en esta tecnología.

Las líneas de investigación actuales son más amplias, permitiendo a los robots resolver problemas de mayor magnitud. Entre estos se pueden mencionar desde autómatas de limpieza para el hogar, hasta los que realizan tareas peligrosas para el ser humano como desactivación de explosivos o manipulación de sustancias peligrosas.



Figura 2. 11 Robot humanoide ASIMO, tomado de [11].



Figura 2. 12 Robot araña C-Bot, tomado de [12].

2.5 Sensor de profundidad Kinect

El sensor Kinect es un periférico que ofrece grandes posibilidades de investigación interactiva y el desarrollo de tareas de realidad aumentada (AR), con interfaces naturales de usuario (NUI). Lo característico de este equipo es que contiene un conjunto de sensores, es de bajo costo, tiene procesamiento integrado y actualmente ya cuenta con un software de desarrollo (SDK) libre, el cual está orientado a la investigación de aplicaciones industriales y comerciales que están fuera del ámbito de los videojuegos, aplicación para la cual el Kinect fue creado.

Kinect fue presentado por Microsoft como el dispositivo complementario para la consola de videojuegos Xbox 360®. Es un controlador de juego que utiliza un sensor de movimiento para controlar todo el cuerpo, su interfaz natural reconoce gestos, comandos de voz y objetos o imágenes, su lanzamiento comercial se realizó en el 2010. La utilización de este dispositivo supone grandes avances en la tecnología que presenta diferentes oportunidades para la investigación en diversas áreas tanto científicas como médicas y hasta espaciales [13].

Tanto el sensor Kinect como el SDK, nos permiten realizar a una amplia gama de aplicaciones como captura de video en tiempo real, realizar análisis y modelación en tercera dimensión, reconocimiento de voz humana y desarrollo de aplicaciones de manos libres, que se puedan controlar mediante la voz.

Podemos construir una serie de aplicaciones para el mundo real que dependiendo del área de desempeño nos abren las puertas de diferentes estudios:

- **Salud:** Se puede construir diferentes aplicaciones para la salud, tales como la medición de ejercicio, los pacientes de control, sus movimientos corporales, y entre otras.
- **Robótica:** Se puede utilizar como un sistema de navegación para los robots, ya sea mediante el seguimiento de los gestos humanos, comandos de voz, o por los movimientos corporales humanos.

- **Educación:** Se puede construir diversas aplicaciones para los estudiantes y niños para educar y ayudar a que aprendan las materias, por medio de sus gestos y comandos de voz.
- **Sistema de seguridad:** Kinect se puede usar para el desarrollo de los sistemas de seguridad donde se puede realizar un seguimiento de los movimientos del cuerpo humano o de la cara y enviar las notificaciones.
- **Realidad virtual:** Con la ayuda de la tecnología 3D y de seguimiento de gestos humanos, muchas aplicaciones de realidad virtual se puede construir utilizando el sensor de Kinect.
- **Entrenamiento:** Se puede utilizar como un entrenador mediante la medición de los movimientos de las articulaciones del cuerpo humano, proporcionando retroalimentación en directo a los usuarios, si las articulaciones se mueven de una manera apropiada mediante la comparación de los movimientos con los datos almacenados previamente.
- **Militar:** Se puede usar para manipular aviones no tripulados inteligentes para espiar a las líneas enemigas.

2.5.1 Entorno de desarrollo

Inicialmente fue conocido bajo el nombre de *Project Natal*, este periférico para videojuegos que prescinde de mandos gracias a un sensor de detección de movimientos, está previsto que sea utilizable en ordenadores con sistema operativo, razón por la cual se utiliza en este proyecto [14].

El sensor de Kinect es un dispositivo horizontal conectado a un pivote, diseñado para estar en una posición longitudinal. El dispositivo tiene una cámara RGB, sensor de profundidad y una matriz de micrófonos bidireccional, conjuntamente capturan el movimiento de los cuerpos en 3D, además de ofrecer reconocimiento facial y aceptar comandos de voz. Dentro de este sensor se pueden identificar los siguientes componentes:

- Cámara VGA a color
- Emisor Infrarrojo (IR)
- Sensor Infrarrojo
- Motor
- Conjunto de micrófonos

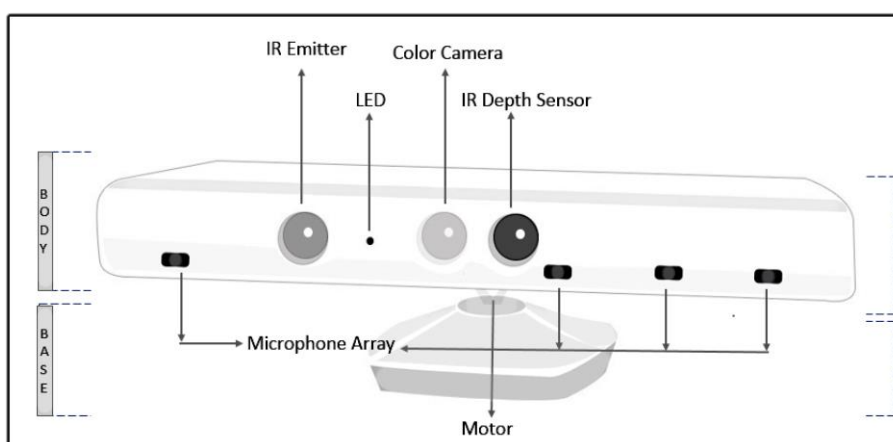


Figura 2. 13Componentes del sensor Kinect, tomado de [13].

2.5.2 Fundamentos de cámara

Esta cámara a color es responsable de la captura y transmisión de los datos de vídeo, su función es detectar los colores rojo, azul y verde de la fuente. El flujo de datos devueltos por la cámara es una sucesión de cuadros de imágenes fijas, el flujo a color de Kinect es compatible con una velocidad de 30 cuadros por segundo (FPS) a una resolución de 640 x 480 píxeles y una resolución máxima de 1280 x 960 píxeles de hasta 12 cuadros por segundo. El valor de fotogramas por segundo puede variar dependiendo de la resolución utilizada para el video.

El límite del rango visual del sensor está entre 1.2 metros y 3.5 metros de distancia, con un ángulo de vista de 57° horizontalmente y un ángulo de 43° verticalmente, mientras que el pivote puede orientarse hacia arriba o abajo ampliando hasta 27°.

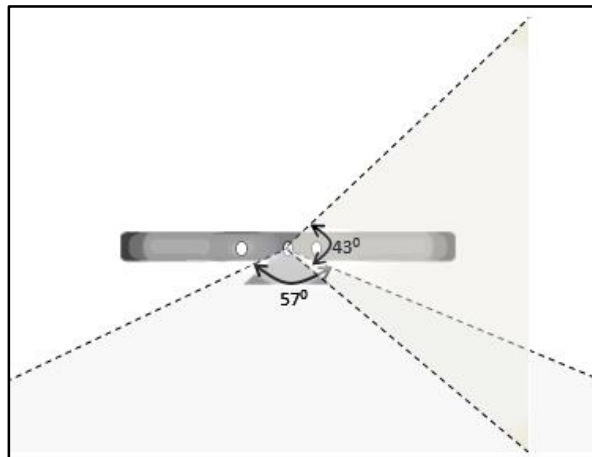


Figura 2. 14 Rango de movimiento horizontal y vertical, tomado de [13].



Figura 2. 15 Fotografía con resolución de 640 x 480 píxeles, tomado de [13].

2.5.3 Emisor y receptor infrarrojo (IR)

Los sensores de profundidad Kinect consisten en un emisor infrarrojo y un sensor de profundidad IR. El emisor de es un proyector que emite constantemente la luz infrarroja en un patrón de "puntos aleatorios" sobre todo en frente del dispositivo.



Figura 2. 16 Patrón de " puntos aleatorios", tomado de [15].

Estos puntos son normalmente invisibles para el espectador, pero es posible capturar su información utilizando un sensor de profundidad IR, la luz de los puntos se refleja en diferentes objetos, y el receptor los lee desde los objetos y los convierte en información mediante la medición de la distancia entre el sensor y el objeto desde donde se leyó el punto IR.

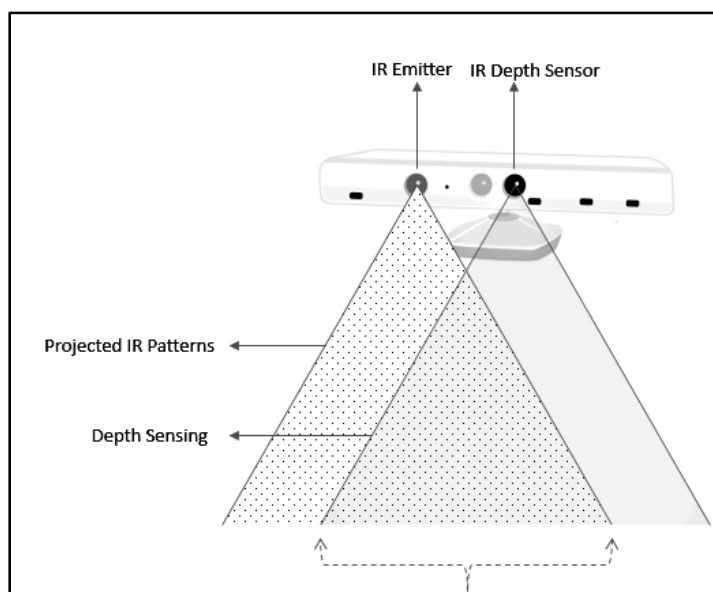


Figura 2. 17 Emisor y receptor infrarrojo, tomado de [13]

El sensor Kinect tiene la capacidad de capturar una imagen 3D primitiva de los objetos en frente, sin importar las condiciones de iluminación de la habitación. Utiliza un emisor infrarrojo y un receptor de profundidad infrarrojo que es un sensor CMOS monocromo (semiconductor complementario de óxido de metal). La columna vertebral detrás de esta tecnología es del chip *PrimeSense*, y la siguiente figura muestra cómo funciona esto:

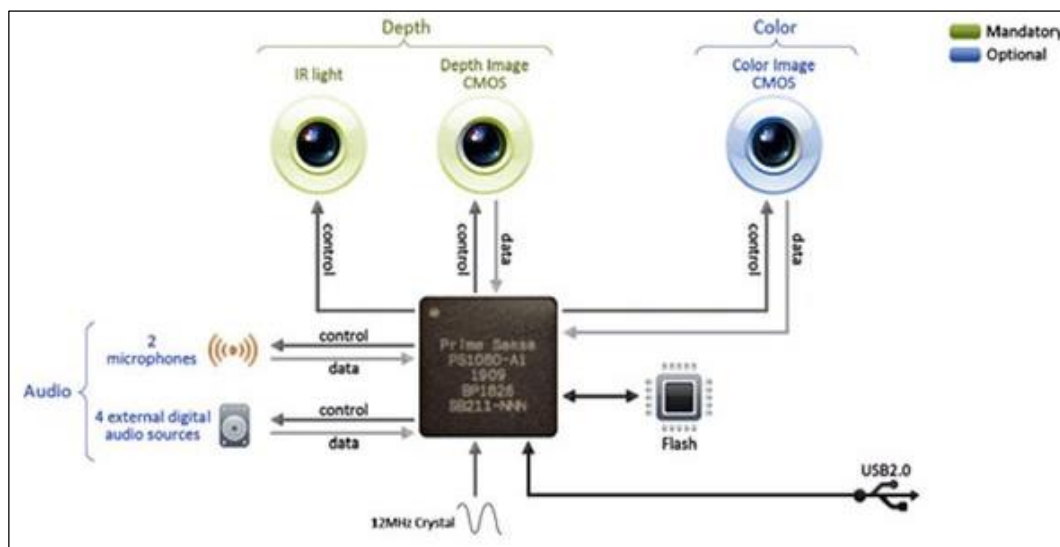


Figura 2. 18 Chip *PrimeSense*, tomado de [16].

Cuando hay necesidad de capturar datos de profundidad, se realiza el siguiente procedimiento:

1. El chip *PrimeSense* envía una señal al emisor para encender la luz infrarroja.
2. Luego envía otra señal al recetor IR para iniciar la captura de datos de profundidad.
3. Mientras tanto el emisor IR empieza a expedir una luz infrarroja invisible a los ojos humanos, hacia los objetos en frente del dispositivo.
4. El sensor de profundidad IR comienza a leer los datos inferidos del objeto basado en la distancia de los puntos de luz individuales emitidos.
5. Inmediatamente los datos pasan al circuito integrado *PrimeSense*.
6. El circuito analiza los datos capturados, y crea una imagen de profundidad por cuadro y se convierte en un flujo de datos de salida mostrándose como una imagen.

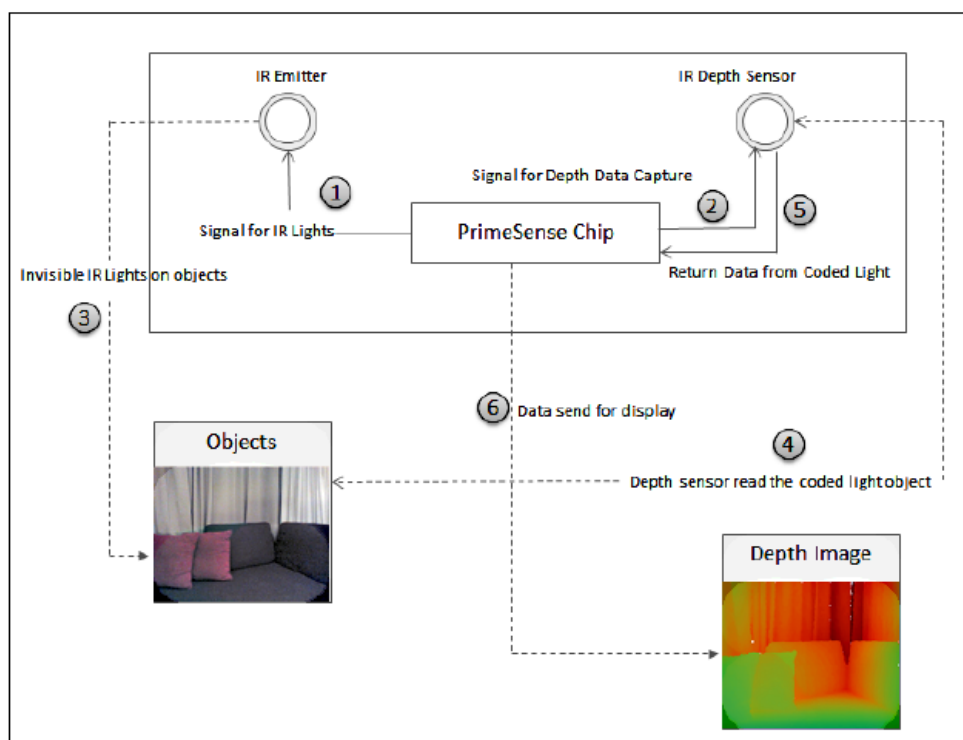


Figura 2. 19 Procesamientos de datos de profundidad, tomado de [13].

2.5.4 Funciones de audio

El dispositivo Kinect presenta un gran apoyo para el audio con la ayuda de un conjunto de micrófonos. El conjunto consta de cuatro micrófonos diferentes que se colocan en un orden lineal, tres de ellos se extienden en el lado derecho y el último se coloca en el lado izquierdo en la parte inferior del sensor.

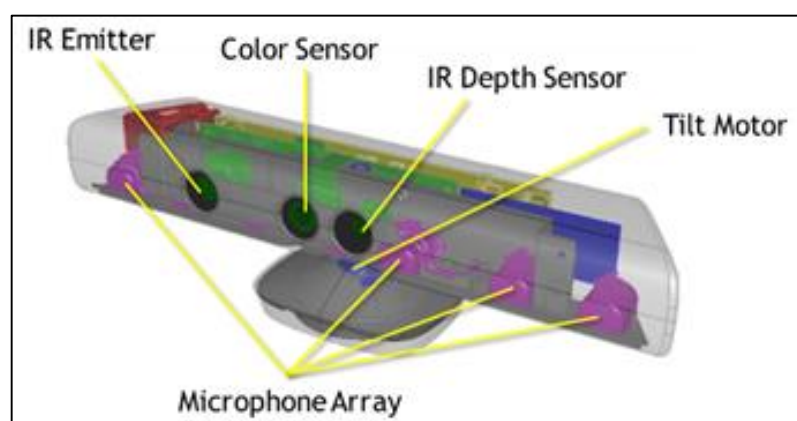


Figura 2. 20 Procesamiento de datos de profundidad, tomado de [17].

El propósito de la matriz de micrófonos no es sólo para dejar que el dispositivo capture el sonido, sino también para localizar la dirección de la onda de audio. Las principales ventajas de tener una matriz de micrófonos sobre un solo micrófono es que la captura y el reconocimiento de voz son más eficaces con la supresión mejorada del ruido, cancelación de eco, y la tecnología de formación de haz. Esto permite al sensor ser un micrófono de alta bidireccional que puede identificar la fuente del sonido y reconocer la voz independientemente del ruido y el eco presente en el medio ambiente.

Si hacemos una analogía con el ser humano este conjunto de micrófonos funcionan como nuestros oídos, al tener uno a cada lado de la cabeza nuestro cerebro calcula aproximadamente de donde viene el sonido, a partir de las diferencias de fase de las ondas que llegan a cada oído.

2.5.5 Funcionamiento de “Skeleton Traking”

El sensor Kinect devuelve los datos de profundidad de los que podemos identificar fácilmente los píxeles que representan a los individuos. El “skeleton traking” no sólo es sobre la detección de las articulaciones mediante la lectura de la información de los individuos; más bien, se trata de la lectura del movimiento del cuerpo completo. En tiempo real se plantea que el reconocimiento es difícil y desafiante debido a las diferentes formas, tamaños, alturas y posturas del ser humano.

Para superar estos problemas y para realizar una localización de diferentes articulaciones independientemente de la pose del cuerpo, Kinect utiliza el “*rendering pipeline*”¹. El algoritmo de reconocimiento de poses humanas, utiliza varios modelos base que varían con diferentes alturas, tamaños, ropa y otros factores, los datos se recogen de la base de personajes con diferentes tipos de posturas, tipos de pelo y la ropa, y en diferentes rotaciones y puntos de vista. Los datos conseguidos se etiquetan con partes individuales del cuerpo y se comparan con los datos de profundidad entrantes para identificar a qué parte del cuerpo

¹ La “*rendering pipeline*” típicamente, recibe la representación de una escena tridimensional como entrada y genera una imagen en 2 dimensiones como salida.

pertenece, el “*rendering pipeline*” procesa los datos en varios pasos para realizar na detección de partes del cuerpo humano a partir de los datos de profundidad [13].

El sensor Kinect puede identificar el rango de píxeles de un personaje desde los datos de profundidad. En los pasos iniciales del proceso de “*rendering pipeline*”, el sensor identifica el objeto “cuerpo humano”, que no es sino datos de profundidad iniciales que es lo mismo que cualquier otro objeto capturado por el sensor, hasta este punto el sensor no sabe si se trata de un cuerpo humano o alguna otra cosa.

Cuando se inicia el reconocimiento de un cuerpo humano, los sensores hacen coincidir cada píxel individual de los datos entrantes con los datos modelados, esta concordancia se lleva a cabo dentro del sensor con una muy alta velocidad de procesamiento. Los datos modelados son etiquetados y tienen algunos valores asociados para que coincida con los datos de entrada. Esta correlación está basada en la probabilidad de que la información entrante coincida con aquella obtenida por algoritmos.

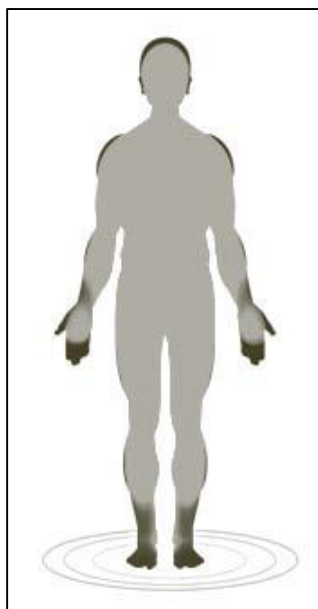


Figura 2. 21 Objeto “cuerpo humano”, tomado de [13]

Lo que sigue en el reconocimiento inmediato de poses, es etiquetar las partes del cuerpo mediante la creación de segmentos, esta creación segmentos se realiza haciendo coincidir los datos probables con similitudes. Kinect utiliza una estructura de árbol de decisiones para que coincida con los datos de un determinado tipo de cuerpo humano, este procedimiento se conoce como “*Decision Forrest*”² [18].

Todos los nodos de este árbol son diferentes datos etiquetados del modelo ser humano, con nombres de las partes del cuerpo. Eventualmente, todos los datos de cada píxel pasan a través de este árbol para que coincida con las partes del cuerpo, el proceso completo de coincidencia de datos se ejecuta una y otra vez. Siempre que coincidan los datos, el sensor comienza marcándolos y a crear los segmentos del cuerpo.

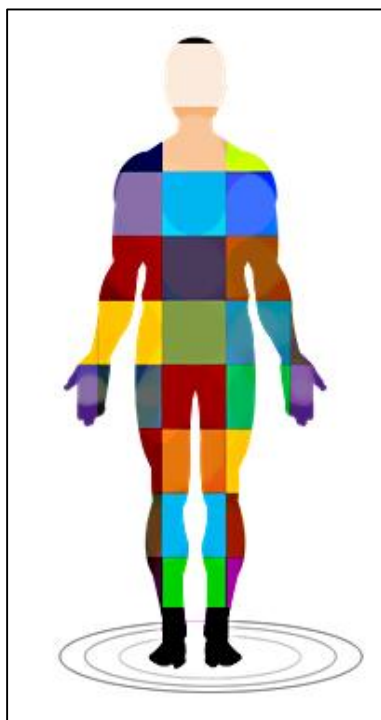


Figura 2. 22 Segmentación del cuerpo humano, tomado de [13]

Una vez que se identifican las diferentes partes del cuerpo, el sensor posiciona los puntos de articulaciones con los datos coincidentes probables. Con los puntos

² Una *Decision Forrest* es una colección de árboles de decisión formados de forma independiente.

de articulaciones identificados y el movimiento de las mismas, el sensor puede seguir el movimiento del cuerpo completo.

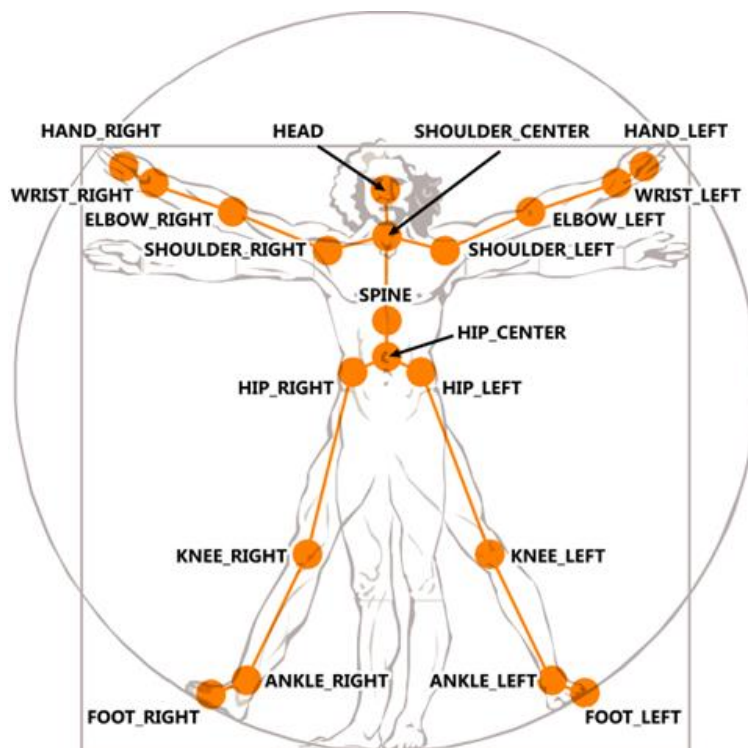


Figura 2. 23 Puntos en las articulaciones, tomado de [19].

Las posiciones de articulaciones se miden por tres coordenadas (X, Y, y Z), donde X e Y definen la posición de la articulación y Z representa la distancia desde el sensor. Para obtener las coordenadas adecuadas, el sensor calcula las tres vistas de la misma imagen: vista frontal, vista izquierda, y la vista desde arriba, por lo que el sensor define la propuesta del cuerpo en 3D.

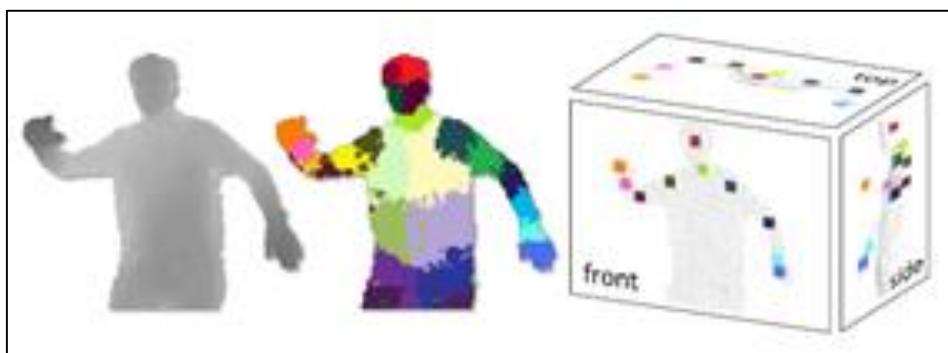


Figura 2. 24 Tres vistas calculadas por el Kinect, tomado de [20].

En síntesis son varios algoritmos y actividades que se realiza para usar el “skeleton tracking”:

- Los datos de profundidad entran en el proceso de “rendering pipelines” y los hace coincidir con los datos etiquetados del “decision forrest” y genera los segmentos corporales deducidos.
- Una vez que todas las piezas están identificadas basados en los datos etiquetados, el sensor identifica las articulaciones del cuerpo.
- El sensor calcula entonces la vista 3D desde la parte superior, frontal y la izquierda de las articulaciones propuestas.
- Finalmente, el sensor inicia el seguimiento del movimiento del esqueleto y del cuerpo humano en base a los puntos de articulación propuestos y la vista 3D.

CAPÍTULO III

DISEÑO DEL SISTEMA

3.1 Arquitectura del sistema

Existen dos partes principales en este proyecto, el cliente y el servidor, el cliente-PC está conectado a un sensor de Kinect. Con la ayuda de este sensor se reconocerán varios gestos diferentes. El cliente está conectado a través de *Wi-Fi* a un servidor que procesa los comandos y los movimientos, en este proyecto el servidor es una pc montada sobre la plataforma del robot móvil ya que la comunicación inalámbrica de los robots no está disponible. Esta aplicación se puede utilizar en cualquiera de los robots móviles del laboratorio pues el proyecto se enfoca en el movimiento de los robots y no en las características individuales de cada uno, es decir se podrá utilizar tanto en los robots Pioneer 3-DX como con el Pioneer P3-AT, a excepción de elementos propios de cada modelo como *gripper* o *bumpers*.



Figura 3. 1 Robot propuesto, Pioneer P3-AT, tomado de [21]

Como ya se ha descrito anteriormente el proyecto consta de cliente y servidor, en el siguiente gráfico se muestra más en detalle como está planteado el sistema para una apropiada ejecución.

El usuario escoge el modo de conducir, el sensor percibe los movimientos y los transmite al cliente, luego éste calcula el respectivo ángulo entre las manos, y lo envía como texto al servidor por medio de la red previamente configurada, el servidor realiza comparaciones con la información enviada y ejecuta los movimientos.

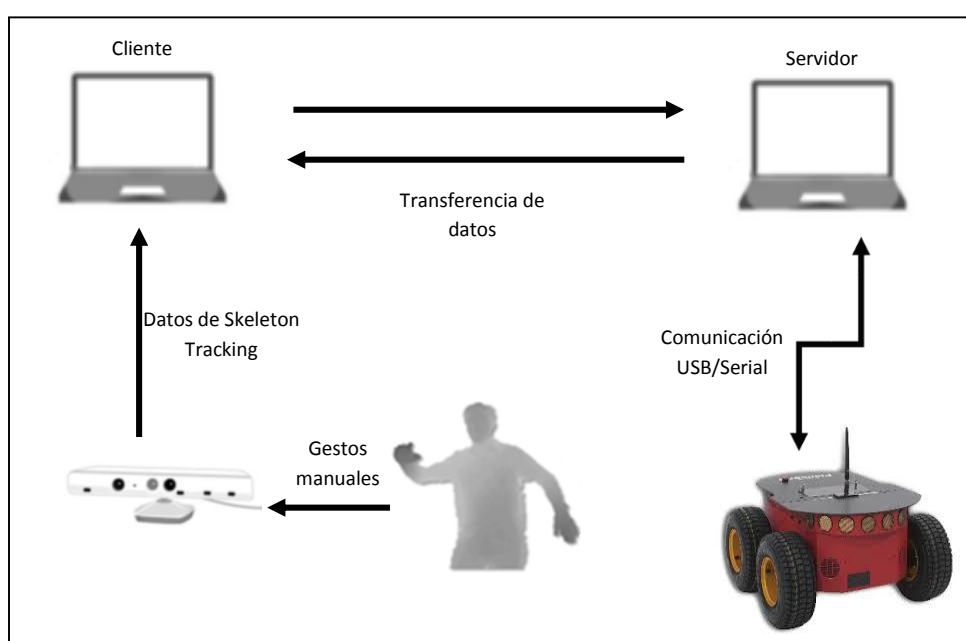


Figura 3. 2 Arquitectura de Comunicación

3.1.1 Cliente

En esta parte el cliente se desempeña a través de una interfaz de usuario, con las respectivas órdenes y/o comandos a ejecutar, esta interfaz será explicada más adelante. Como primer punto importante a mencionar, está el lenguaje de programación utilizado para crear la interfaz del cliente, la aplicación cliente se desarrolló en Windows 7 con Visual Studio 2010 y está escrita en C #, aunque el SDK del sensor nos permite trabajar también con C++, se escogió el primero por las siguientes razones:

- La comunidad de programadores y desarrolladores que trabaja con C# en aplicaciones para el sensor Kinect es más amplia que aquellos que utilizan el lenguaje C++.
- No se requieren librerías externas.
- Permite hacer una programación más sencilla al momento de crear una aplicación de red, ya que provee una mejor interfaz de trabajo.

Con la ayuda del sensor de Kinect es posible capturar vídeo RGB y audio, pero la característica principal es el seguimiento de partes del cuerpo de los seres humanos en el espacio 3D, esto significa que un modelo de esqueleto de la persona se puede producir si se realiza un seguimiento por el sensor. El sensor Kinect utiliza infrarrojos que chocan en los objetos y lee los puntos reflejados con una cámara de infrarrojos, esto no funciona bien si el brillo del obstáculo es demasiado alto, por lo que el sensor de Kinect se debe utilizar en interiores.

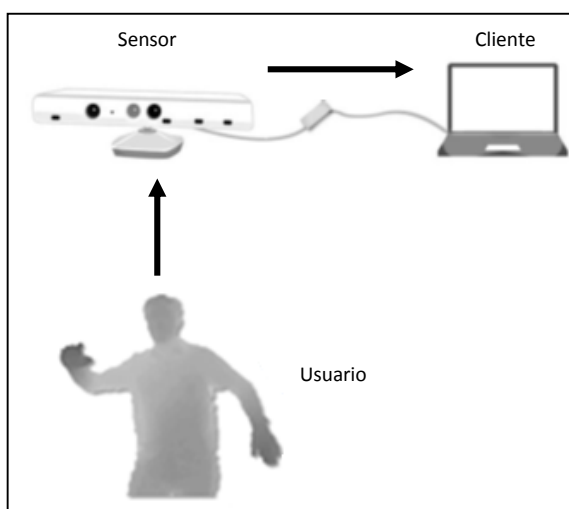


Figura 3. 3 Transmisión de datos al cliente

La figura 3.3 muestra que al ponerse una persona frente al sensor, este comienza inmediatamente a detectar los movimientos, realiza un modelo del esqueleto de la persona en frente y posiciona las articulaciones, estos datos son transmitidos al cliente para realizar operaciones.

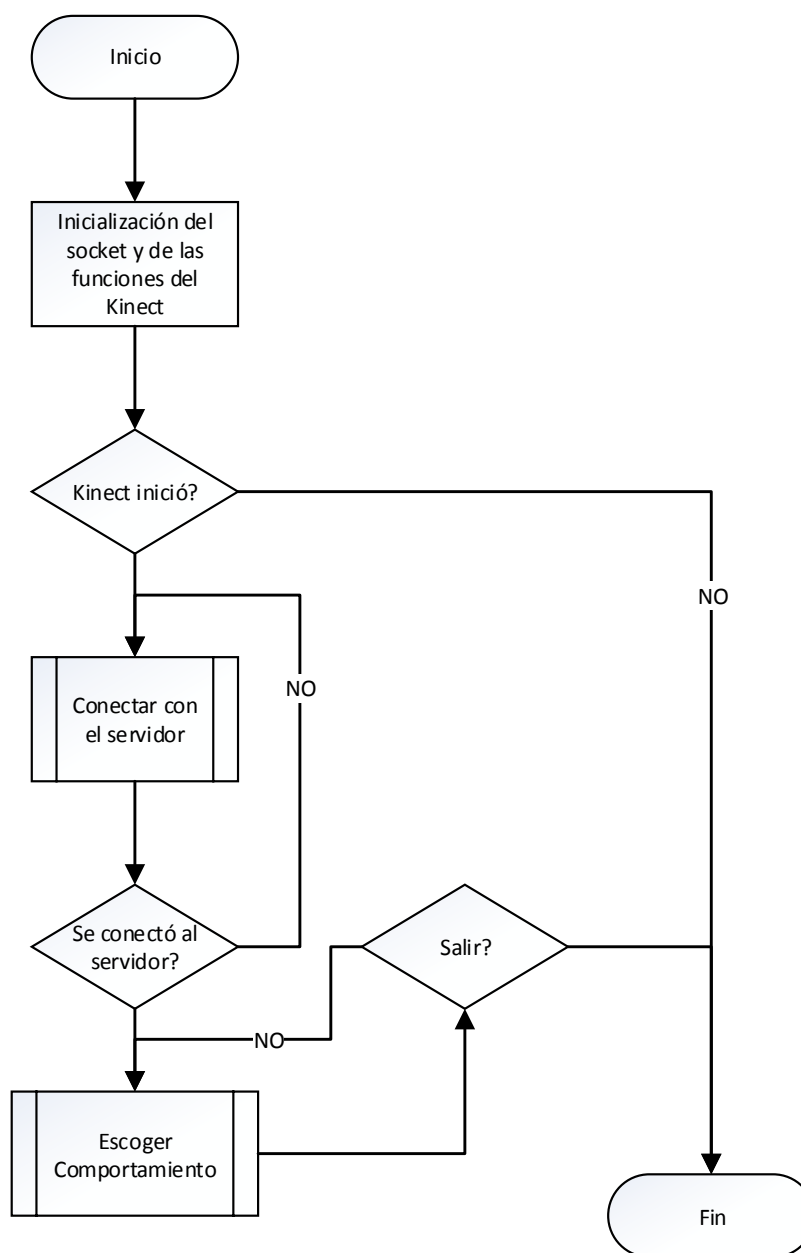


Figura 3. 4 Diagrama del cliente

La interfaz de usuario se encarga de inicializar las funciones del Kinect y del socket de comunicación, la figura 3.3 muestra el diagrama de las operaciones del cliente, se busca conectar con el servidor, luego se puede escoger un modo de operación hasta que el usuario opte por salir, en tal caso se finaliza la aplicación.

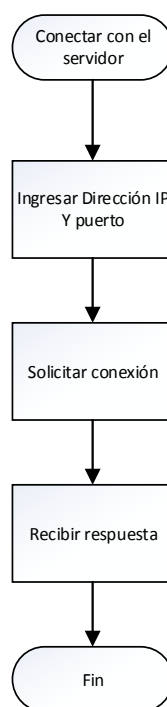


Figura 3. 5 Conexión con servidor

Después de ingresar la dirección IP y el puerto de comunicación el cliente espera por la réplica del servidor, si se concreta la conexión entonces se muestra el mensaje recibido. Si se selecciona el modo conducir, el método lanza un proceso asíncrono, para enviar el ángulo de rotación del robot.

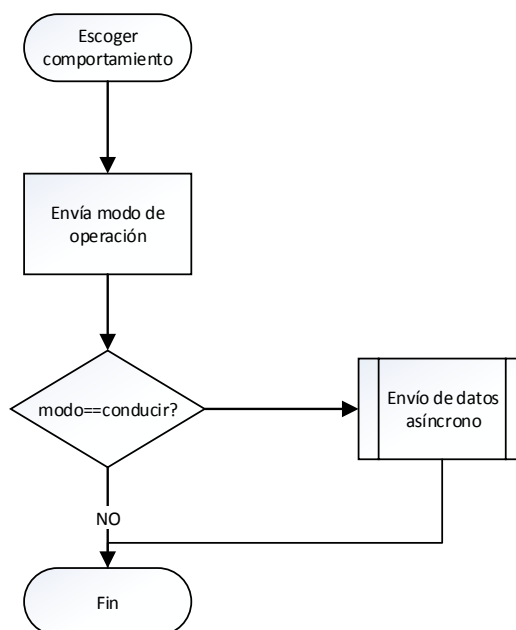


Figura 3. 6 Envío de modos de operación

Una vez que se calcula el ángulo el proceso asíncrono toma ese dato y lo envía por el socket hacia el servidor, luego actualiza el ángulo tomado de las posiciones de las manos y se vuelve a enviar, mientras no se cancele la acción mandar el ángulo el proceso se mantiene.

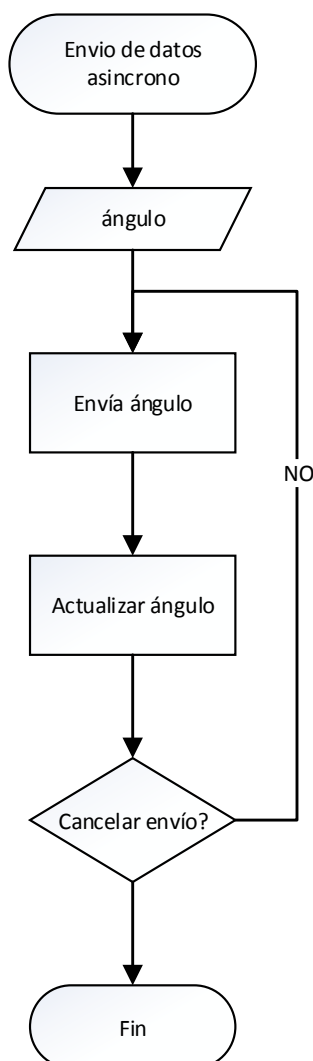


Figura 3. 7 Proceso asíncrono para el envío de datos

Todo este proceso se mantiene independiente del programa principal, así se ahorra tiempo de procesamiento, y es más eficiente el uso de la memoria del computador.

Es necesario mencionar los requerimientos tanto de hardware como de software para el cliente, que permiten la programación y aplicación del proyecto:

- El sistema operativo en el que se desempeña el trabajo debe ser Windows 7 o superior.
- Microsoft Kinect(SDK)
- Visual Studio
- .NET Framework 4.0

Se puede decir que .NET Framework hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. En síntesis su propuesta es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones.

3.1.2 Servidor

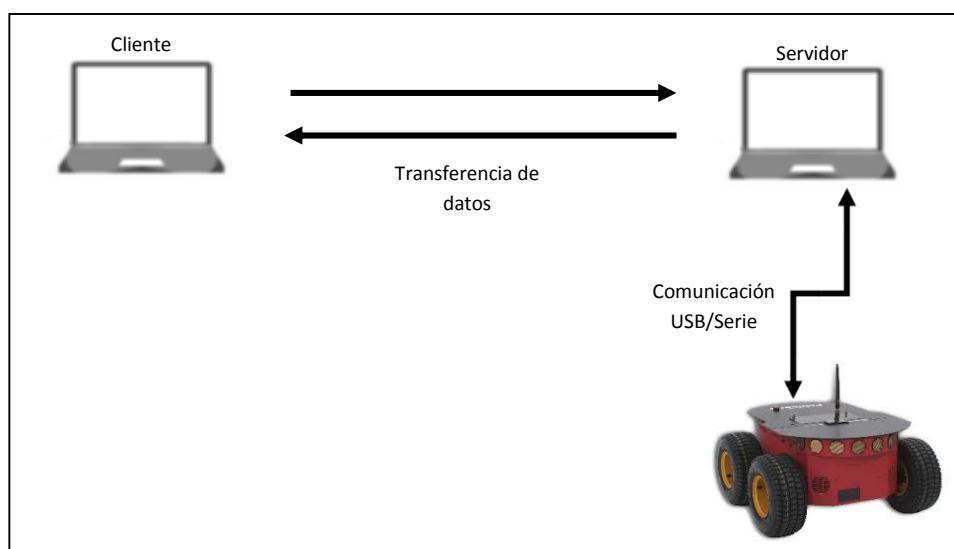


Figura 3. 8 Comunicación entre el servidor y el robot

La aplicación del servidor se desarrolla en el robot Pioneer P3-AT, lastimosamente no cuentan con una antena para comunicación inalámbrica, de tal manera que es necesario montar una PC tipo laptop en la plataforma del robot, en esta computadora se instala el servidor del proyecto, y se comunica con el robot

por medio de puerto USB, cabe aclarar que para dicha comunicación es necesario utilizar un cable de transmisión USB a serial, porque la plataforma del Pioneer no cuenta con este puerto.

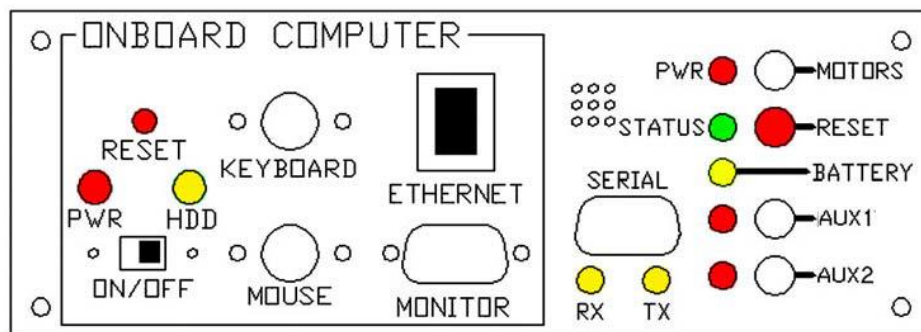


Figura 3. 9 Computadora y controles del P3-AT, tomado de [22].

La aplicación del servidor con la programación de sus comportamientos en lenguaje C++, se puede realizar en el sistema operativo que se desee, en este caso se usó Windows, y se debe incluir la librería ARIA³, exclusiva para la programación de los robots Pioneer.

Una vez que se logra la comunicación entre el robot y la PC servidor, ya podemos interactuar con el cliente y así completar la comunicación total, desde el usuario hasta el robot.

El P3-AT se moverá dependiendo de las acciones que envíe el cliente, cada uno de los comportamientos está configurado en el programa servidor, cuando se pida realizar una acción, ésta se realizará excluyendo a las demás.

³ Advanced Robotics Interface for Applications (Interfaz Robótica Avanzada para Aplicaciones)

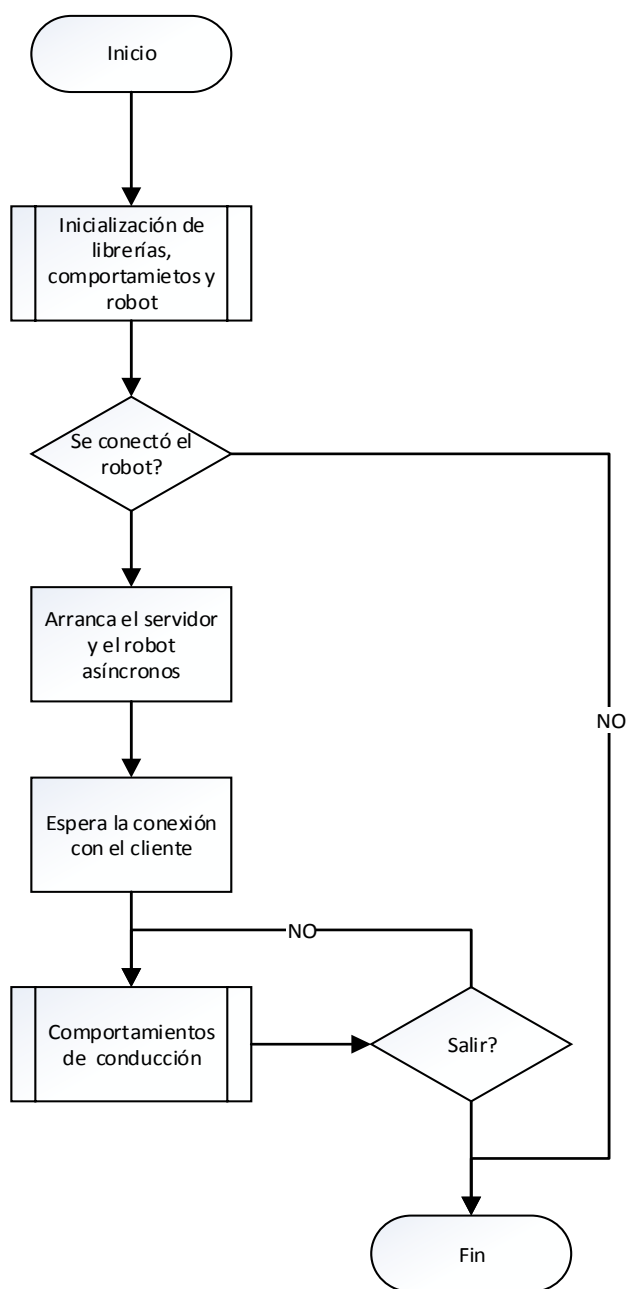


Figura 3. 10 Diagrama del servidor

Como se ve en la figura 3.10, lo primero que se realiza es la inicialización de las librerías y se crea el grupo de acciones configuradas para cada comportamiento del robot. Al final el servidor siempre está esperando una respuesta del cliente si se envía “salir” termina la aplicación.

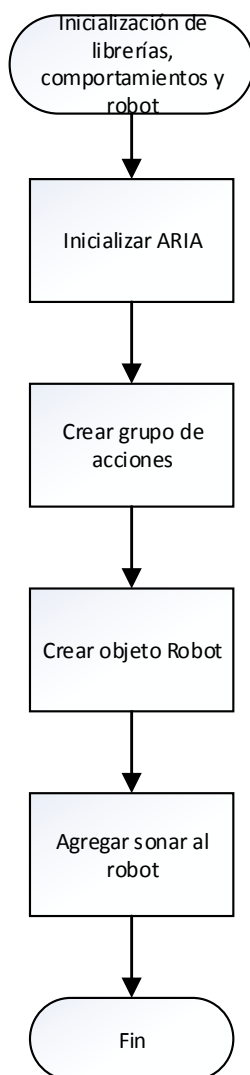


Figura 3. 11 Subproceso de inicialización

El subproceso inicialización carga las librerías necesarias y crea los objetos robot y sonar, este último es agregado al robot y sirven para evitar las colisiones en cada comportamiento.

La selección de comportamientos se realiza en un lazo donde cada dato recibido se compara para determinar la acción solicitada por el cliente, y se mantiene en este lazo hasta que se pide salir.

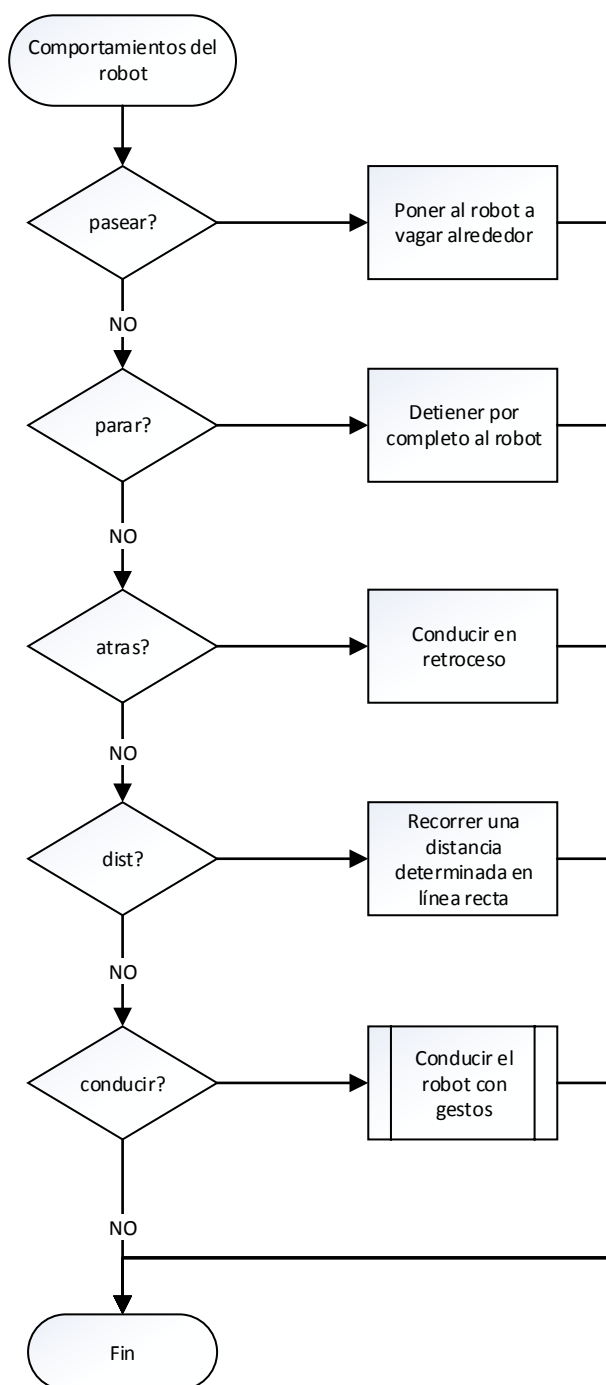


Figura 3. 12 Subproceso comportamientos

3.2 Descripción de hardware y requerimientos

Los dispositivos que están implicados en este proyecto deben cumplir con ciertos requerimientos para poder ejecutar debidamente todo el proceso de la aplicación, para lo cual se debe diferenciar tanto los requerimientos del cliente

como los del servidor. Todos los requerimientos de cada dispositivo se muestran en la siguiente tabla:

Tabla 3. 1 Descripción de Requerimientos

Cliente/Servidor	Requerimientos
Servidor	<ul style="list-style-type: none"> • Librería ARIA • Visual Studio(C++)
Cliente	<ul style="list-style-type: none"> • Microsoft Kinect SDK • Visual Studio(C#) • NET Framework • Sistema Operativo Windows 7/8

3.2.1 Comportamiento de robot

En este proyecto se ha propuesto una serie de comportamientos para el robot, a continuación se muestra el conjunto de comportamientos que se permitirán para el robot móvil.

3.2.1.1 Comportamiento paseo

A medida que el comportamiento paseo, realiza los movimientos, las funciones del programa se aseguran que el robot no colisione con ningún obstáculo. Además se deben evitar los obstáculos que están más lejos, finalmente cuando no hay ningún objeto frente al robot, éste se mueve con una velocidad constante de 250 mm/s.

3.2.1.2 Comportamiento distancia

Permite conducir el robot en una distancia específica, el robot se desplaza hacia adelante si el parámetro es positivo y hacia atrás si es negativo.

3.2.1.3 Comportamiento conducir

Este comportamiento se ejecuta directamente desde el cliente, para lo cual se requiere de dos parámetros, ángulo y velocidad. Este comportamiento es el tema principal del proyecto.

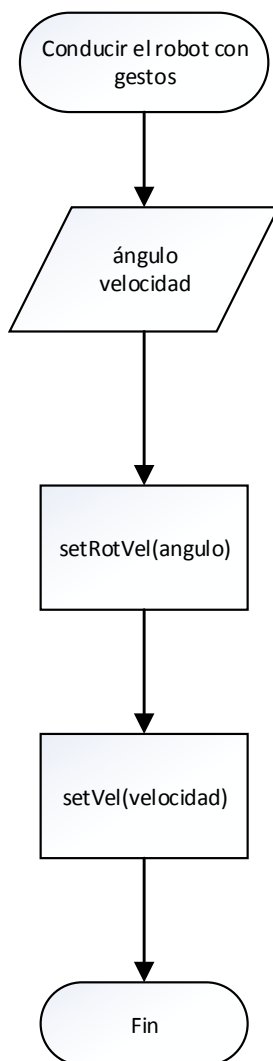


Figura 3. 13 Subproceso conducir

En el proceso de conducir por gestos, el dato que se recibe principalmente es el ángulo, la velocidad es constante para evitar que el robot acelere indiscriminadamente, esta es una forma de limitar los posibles errores que el usuario puede cometer cuando se conduce el robot.

3.2.1.4 Comportamiento atrás

Este modo de operación permite al robot la conducción en retroceso, con una velocidad constante de 150 mm/s., así se evita que el usuario conduzca el robot hacia un obstáculo.

Es importante mencionar que estos comportamientos se pueden ejecutar directamente con el robot o en el modo simulador usando la herramienta *MobileSim*⁴, en esta forma se ejecutan los comportamientos por medio del teclado.

3.2.2 Movimientos y gestos admitidos

El robot debe ser controlado por gestos manuales y existen tres tipos de gestos admitidos para el usuario:

- a) **Parar:** Si la posición vertical tanto de la mano derecha como de la mano izquierda, están por debajo del centro de la cadera, el robot se detendrá.
- b) **Conducir:** Se levanta las manos por encima de la cadera para proceder con el comando de conducir, se calcula el ángulo entre la mano derecha y la mano izquierda, luego la distancia entre las manos corresponde a la velocidad del robot. Para mayor velocidad la distancia es más amplia y para una velocidad menor la apertura es más pequeña.
- c) **Retroceso:** Cuando sea necesario conducir hacia atrás solo será necesario subir los antebrazos y cruzarlos.

3.3 Software de programación

En este capítulo se revisa las librerías específicas que se necesitan para la aplicación del proyecto, como la *ARIA Library* y *Kinect SDK*, si bien es cierto que

⁴ Simulador de robots móviles

la programación se realiza en C# y C++ para el cliente y el servidor respectivamente, no es necesario describir ambos lenguajes de programación sino más bien las librerías utilizadas por estos, ya que estas contienen la información necesaria para los procesos implicados en este proyecto.

3.3.1 ARIA (Advanced Robotics Interface for Applications)

El software está diseñado para todas las plataformas de *MobileRobots*, es un entorno de desarrollo en código abierto basado en C++, que proporciona una interfaz robusta de cliente, para una variedad de sistemas robóticos inteligentes, incluyendo microcontrolador de la plataforma y sistemas de accesorios.

ARIA / ARNetworking es la plataforma ideal para la integración de su propio software de control del robot, ya que maneja cuidadosamente los detalles de nivel más bajo de interacciones cliente-servidor, incluyendo la creación de redes y comunicaciones en serie, comando y servidor de procesamiento de paquetes de información, el tiempo de ciclo, y “*multithreading*”⁵, así como el apoyo de una variedad de accesorios y controles, como un buscador de escaneo láser de gama, giroscopios de movimiento, el sonar, y muchos otros.

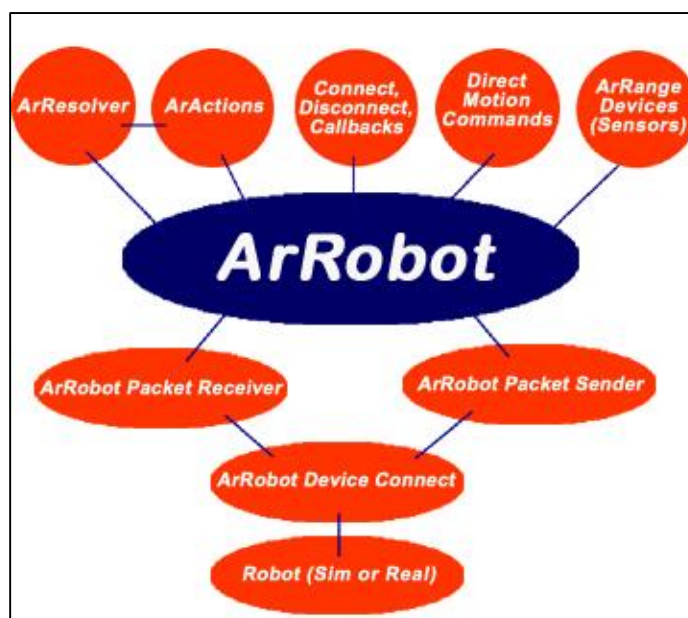


Figura 3. 14 Arquitectura ARIA, tomado de [22]

⁵ Capacidad de ejecutar eficientemente múltiples subprocesos.

Una vez instaladas las librerías de ARIA, el servidor puede llamar a las subclases de `ArAction` para desarrollar los comportamientos del robot, por ejemplo `ArActionStop` pone la velocidad del robot en cero.

3.3.2 Kit de desarrollo de software para Windows(SDK)

El SDK de Kinect para Windows es el conjunto de bibliotecas que nos permite programar aplicaciones en una variedad de plataformas de desarrollo de Microsoft que utilizan el sensor de Kinect como entrada. Con él, se puede programar aplicaciones WPF⁶, aplicaciones WinForms. Los desarrolladores pueden utilizar el SDK con el sensor de Kinect de Xbox. Para utilizar las capacidades de modo cerca de Kinect, requerimos que el hardware sea el oficial para aplicaciones de Windows, además, el sensor Kinect para Windows está requerido para despliegues comerciales.

3.4 Especificaciones de diseño

La comunicación de los programas y la ejecución de acciones está diseñada para seguir una serie de pasos organizados donde primero estableceremos una comunicación entre cliente y servidor a través de sockets⁷, una vez enviada una instrucción al servidor, este reenviará un mensaje de reconcomiendo o de error cual sea el caso, en el servidor se encontrarán las instrucciones prediseñadas para que sean aplicadas por el robot, se sobreentiende también que en el caso de necesitarse parámetros de entrada como velocidad y dirección estos son enviados con la orden de comportamiento ya especificadas.

3.4.1 Diseño de la red de comunicación

En esta aplicación en el cliente hay tareas que requieren una gran cantidad de tiempo de cálculo. Para una aplicación reactiva, es necesario el uso de *threads* (subprocesos que se ejecutan de manera asíncrona del programa principal) para tareas que requieren mucho tiempo.

⁶ Windows presentation foundation

⁷ Método de comunicación de un programa cliente y uno servidor dentro de una red.

El lenguaje C # admite varias opciones para poner en práctica los *threads*, uno de ellos es la clase *BackgroundWorker*. El control *BackgroundWorker* fue introducido en el .NET Framework 2.0 y permite realizar operaciones costosas o duraderas en un *thread* diferente al de la interfaz, de manera que la aplicación sigue respondiendo al usuario mientras este trabajo se procesa en segundo plano.

Al iniciar el programa se puede ingresar los comportamientos por comandos a través de botones en la interfaz del cliente, pero se debe tomar en cuenta que mientras se hacen los gestos de movimientos también se podría ingresar los comportamientos a través de botones esto puede producirá un error, por tanto es necesario sincronizar y validar el ingreso de datos para evitar estos errores. Para recibir el contenido desde el servidor, se utilizan *threads* para que esto no interfiera con la ejecución principal del cliente que es obtener los datos de movimiento.

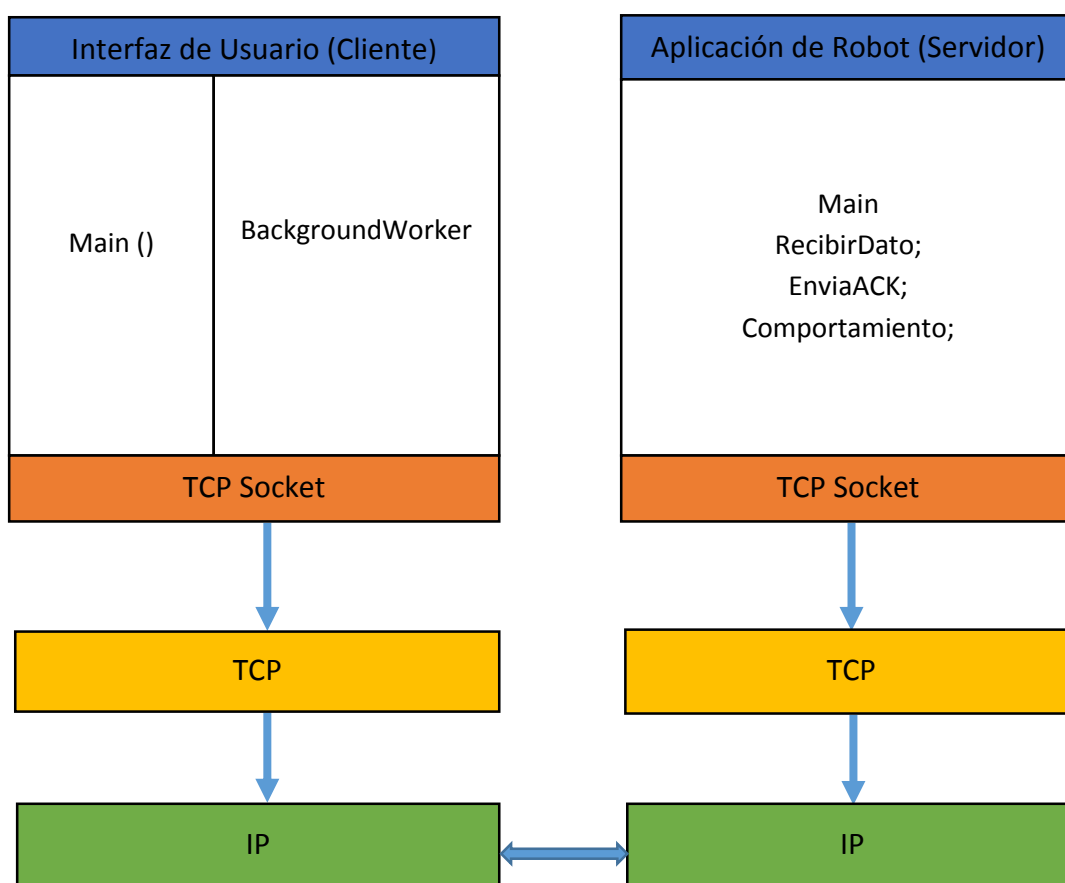


Figura 3. 15 Red de comunicación

3.5 Interfaz gráfica de usuario

La interfaz gráfica es el programa que interactúa directamente con el usuario y que le permite realizar todas las tareas definidas en este proyecto, aquí se valida todo aquello que el cliente emite al servidor, para que este último se dedique específicamente a ejecutar los comportamientos del robot descritos anteriormente.

El cliente debe empezar la conexión con el servidor, el mismo que debe estar previamente inicializado y en comunicación con el robot o el simulador, el usuario ingresa la dirección IP y el puerto de comunicación, así el servidor envía un mensaje “Hola Cliente” reconociendo la conexión con el usuario, como se muestra en la figura 3.16.



Figura 3. 16 Inicio de conexión

En la sección de configuración se puede observar el título y los campos a ingresar, seguido de respuesta de reconocimiento del servidor, en ese instante se puede seleccionar la pestaña “Manejo”, donde se opta por los distintos comportamientos.

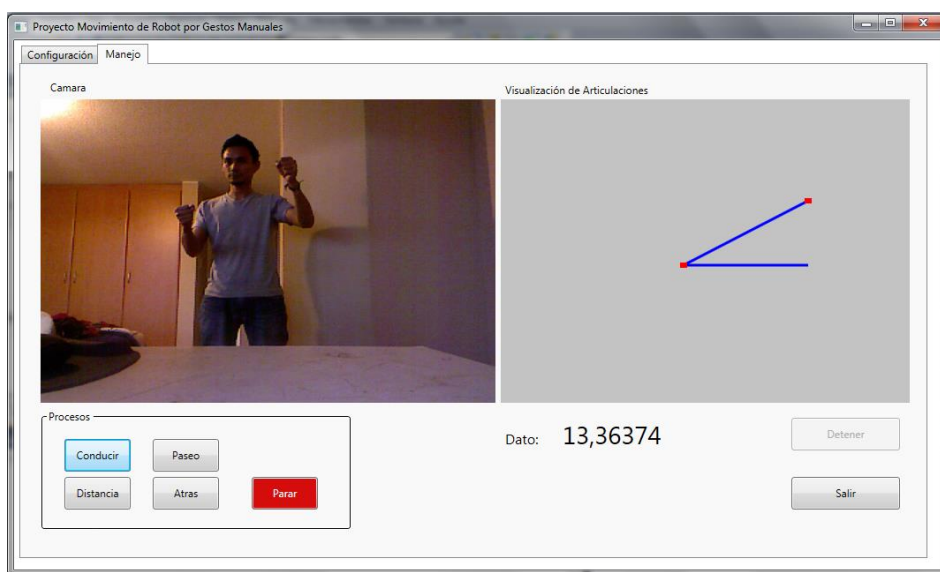


Figura 3. 17 Interfaz de comportamientos

La pestaña manejo permite escoger los comportamientos a ejecutar y se visualiza los gestos realizados, el dato enviado al servidor, la ilustración del “*skeleton tracking*” con el ángulo obtenido entre las manos del usuario y los botones para cada comportamiento.

3.5.1 Componentes de configuración

El primer componente es la caja de texto de la dirección IP, donde se introduce la dirección asignada al servidor.

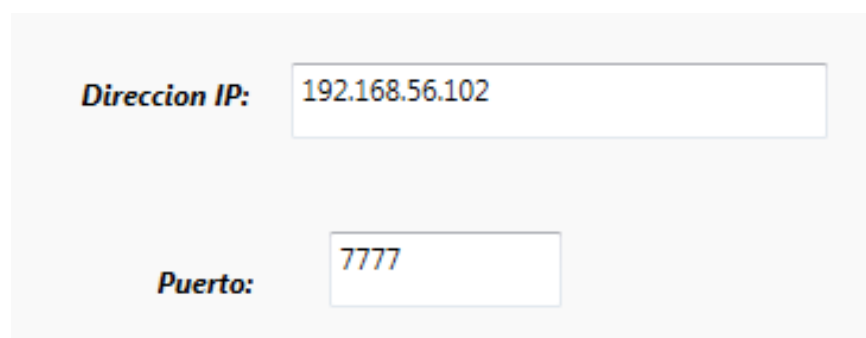


Figura 3. 18 Dirección IP y puerto de comunicación

El puerto de comunicación es importante, porque a través de éste se transfieren los datos entre cliente y servidor, se pueden usar varios puertos en una misma red para funciones diferentes. El botón conectar permite establecer la conexión con el servidor con la dirección IP y el puerto, ingresados en las cajas de texto. Cuando se establece la conexión se recibe un mensaje como el que se muestra en la figura 3.19:

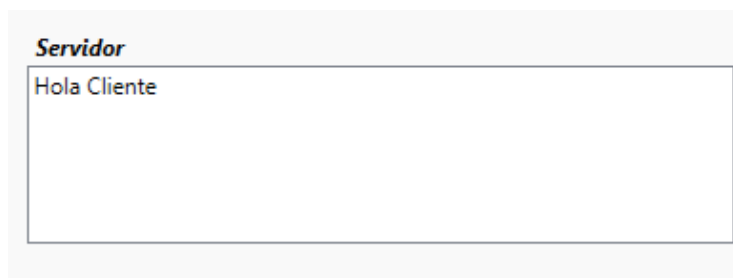


Figura 3. 19 Mensaje de bienvenida

3.5.2 Componentes de manejo

La sección de manejo consta de una ventana para el video donde se muestra la imagen obtenida por la cámara a color del sensor, con una resolución de 640 x 480 píxeles.

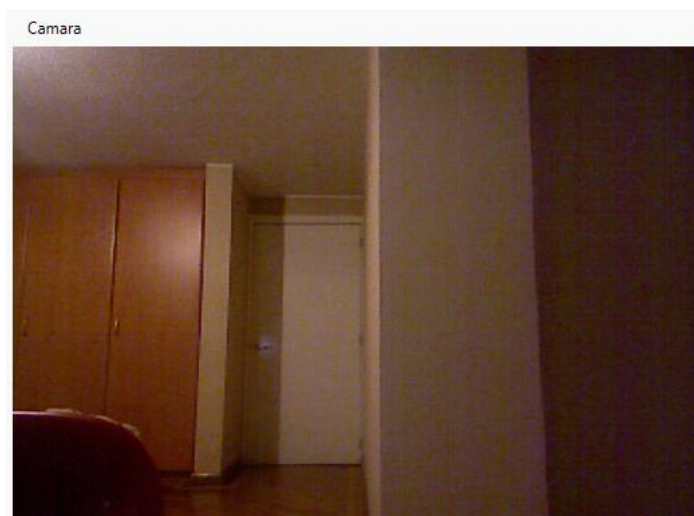


Figura 3. 20 Imagen a color obtenida por el sensor

La siguiente imagen permite visualizar las articulaciones detectadas por el sensor infrarrojo del Kinect, se grafica un punto para cada mano el ángulo entre ambas.

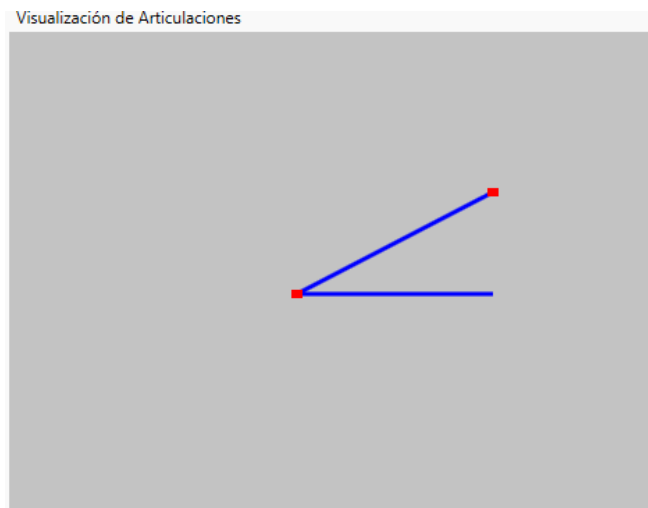


Figura 3. 21 Imagen de visualización de articulaciones y ángulos

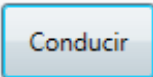
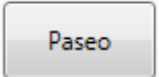
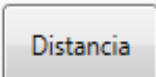
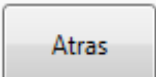

El panel de procesos contiene los botones de cada comportamiento, lo más importante es tener en cuenta que estos modos se activan de forma exclusiva.



Figura 3. 22 Panel de Comportamientos

La siguiente tabla muestra la descripción de cada comportamiento optado por cada botón y su correspondiente método:

Tabla 3. 2 Descripción de modo de operación

Botón	Método	Descripción
	conducir	Permite establecer la conducción del robot por gestos manuales.
	pasear	El robot móvil se mueve alrededor sin rumbo específico evitando obstáculos.
	dist	Ordena al robot moverse una distancia determinada en línea recta.
	atrás	Permite que mover el robot en retroceso
	parar	Detiene al robot sea cual sea el comportamiento

Por ultimo encontramos los botones de salir y detener, “salir” sirve para terminar la conexión y abandona la aplicación, “detener” es para que el usuario deje de enviar datos desde el Kinect al servidor.

**Figura 3. 23** Botones *salir* y *detener*

Existen posibles errores que se pueden encontrar, esto sucede cuando no se procede correctamente en la conexión del hardware; tanto el Kinect como el robot deben estar conectados para iniciar la comunicación de la red.

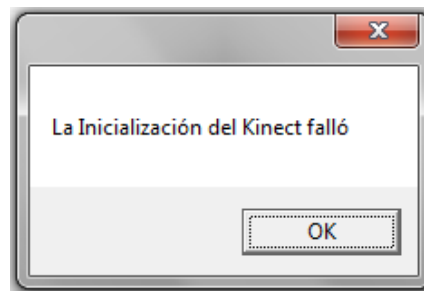


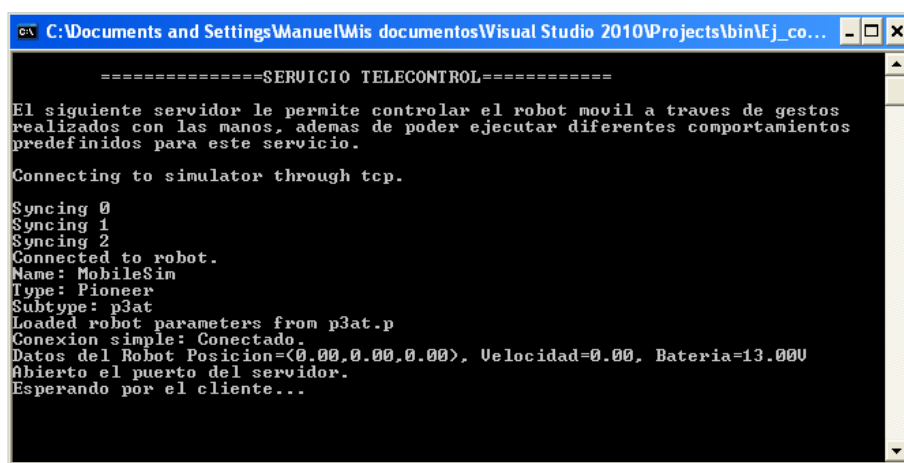
Figura 3. 24 Mensaje de falló la conexión con el sensor

CAPÍTULO IV

ANÁLISIS Y PRUEBAS DEL SISTEMA

4.1 Pruebas de comunicación

El sistema está diseñado para que ejecute el servidor, estando éste conectado al robot o al simulador, luego se espera a que el cliente se comunice para iniciar la aplicación. Si no es posible la comunicación con el robot se lanza una excepción informando un error y saliendo del programa.



```
====SERVICIO TELECONTROL====
El siguiente servidor le permite controlar el robot movil a traves de gestos
realizados con las manos, ademas de poder ejecutar diferentes comportamientos
predefinidos para este servicio.
Connecting to simulator through tcp.
Syncing 0
Syncing 1
Syncing 2
Connected to robot.
Name: MobileSim
Type: Pioneer
Subtype: p3at
Loaded robot parameters from p3at.p
Conexion simple: Conectado.
Datos del Robot Posicion=(0.00,0.00,0.00), Velocidad=0.00, Bateria=13.00U
Abierto el puerto del servidor.
Esperando por el cliente...
```

Figura 4. 1 Servidor conectado

Una vez que se establece la comunicación el programa se dispone a esperar al usuario, la figura 4.1 nos muestra la información obtenida del robot móvil como su posición velocidad y carga de baterías, la comunicación se establece principalmente entre el servidor que opera directamente el robot y el cliente que envía las instrucciones de movimiento ya sea por comandos o por gestos manuales. Estas pruebas están realizadas tanto para el simulador como para el robot P3-AT.

Para poder iniciar la aplicación completa es necesario que el servidor esté conectado con robot y el cliente debe estar conectado con el sensor, una vez hecho esto ya se puede manejar el robot.

El sistema está configurado para que al momento de realizar la conexión desde el cliente, el servidor envía una señal de reconocimiento diciéndole al usuario “Hola Cliente”, con esto se sabe que la comunicación ya se estableció con el servidor.



Figura 4. 2 Mensaje de bienvenida del servidor

Al pasar a la pestaña de conducción en la interfaz podemos controlar el robot de dos maneras, ya sea por comandos o por gestos manuales, para evitar que se envíe un comando mientras se maneja por gestos el programa impide que ingrese algún comando mientras no se finalice la conducción.

4.2 Adquisición de datos

Como ya se ha mencionado el programa controla el robot por medio de comandos o por gestos manuales el cual es la aplicación principal, para el correcto funcionamiento, el robot toma los valores de velocidad y ángulo determinados por la posición de las manos. El rango de ángulos que se puede ingresar va de -90° a 90° , aunque no es necesario llegar a estos valores máximos pues el robot gira con una velocidad de rotación y no con un ángulo directamente.

Para poder obtener el ángulo es necesario obtener las posiciones de las manos, luego se traza una línea horizontal imaginaria, desde la mano izquierda hasta la posición de X_D de la mano derecha, esta línea será la referencia para marcar el ángulo de giro de las manos y posteriormente del robot, la posición Y en ambas manos es igual así se mantiene una línea horizontal, por tanto $Y_I = Y_D$.

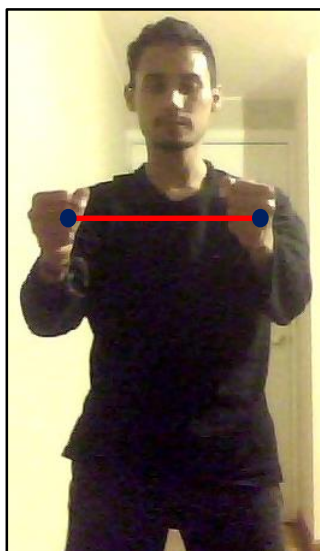


Figura 4. 3 Línea trazada entre las manos

Una vez que se tiene la línea de referencia se traza una línea dada por la posición de las manos como se muestra en la figura 4.3 la cual determina el ángulo que será enviado al robot.

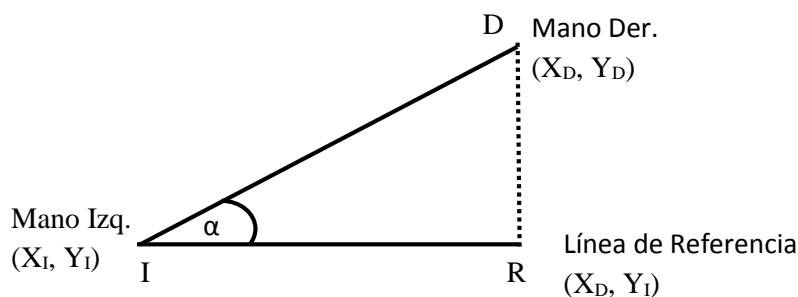


Figura 4. 4 Angulo entre las manos

Para obtener el ángulo α se utiliza la definición trigonométrica de coseno, donde el coseno de un ángulo está determinado por el cociente entre el cateto adyacente y la hipotenusa (3). Ahora obtenemos la dimensión de la hipotenusa y del cateto, como se usa en (1) y (2):

Hipotenusa: \overline{ID}

Cateto: \overline{IR}

$$\overline{ID} = \sqrt{(X_D - X_I)^2 + (Y_D - Y_I)^2} \quad (1)$$

$$\overline{IR} = \sqrt{(X_D - X_I)^2} \quad (2)$$

$$\alpha = \cos^{-1} \frac{\overline{IR}}{\overline{ID}} \quad (3)$$

Tabla 4. 1 Ángulos enviados al servidor

<i>Muestra</i>	α	$\alpha/2$	$\alpha/3$	$\alpha/5$
1	03.1233	00.0140	00.6020	00.1307
2	05.1826	03.4693	00.7434	01.6580
3	19.2739	07.9349	01.4244	04.0390
4	33.0039	10.2589	06.1634	05.6904
5	42.2090	13.8514	08.5950	07.2268
6	51.3436	17.4088	12.3703	08.4278
7	54.13.54	20.4567	14.7429	09.3981
8	53.8911	22.9922	18.3775	11.0203
9	48.4466	24.5446	19.5532	12.0223
10	35.1111	26.1011	19.7501	13.3193
11	04.5604	27.7545	18.1034	12.8708
12	03.2308	28.6312	12.1036	09.2864
13	-06.4376	27.6133	08.9370	06.0075
14	-18.8650	20.2385	06.4740	03.3045
15	-26.7977	13.0157	04.5478	01.9365
16	-35.1878	09.2853	02.4528	00.6450
17	-41.4424	05.3992	-00.4931	-00.5881
18	-42.4420	02.9769	-03.1681	-02.3996
19	-43.1771	-01.4422	-08.4258	-05.8059
20	-40.5505	-05.8563	-12.8674	-06.4522
21	-19.5209	-08.7922	-14.6944	-05.6465

La prueba se realizó desde una posición donde el ángulo es cercano a cero, luego se gira lentamente los brazos hacia arriba para tomar ángulos positivos y después se gira hacia abajo para tomar los ángulos negativos, cada adquisición se hizo dividiendo el ángulo obtenido para un número determinado, cada envío de datos se realiza en intervalos de 330 ms., es decir tres datos cada segundo aproximadamente.

El robot tiene en su programación interna un método para girar *setRotVel()*, pero este dato es la velocidad de rotación y no el ángulo de rotación que se está obteniendo, por ello la tabla 4.1 muestra algunos resultados de los valores dados al método *setRotVel()*, para escoger el mejor desempeño.

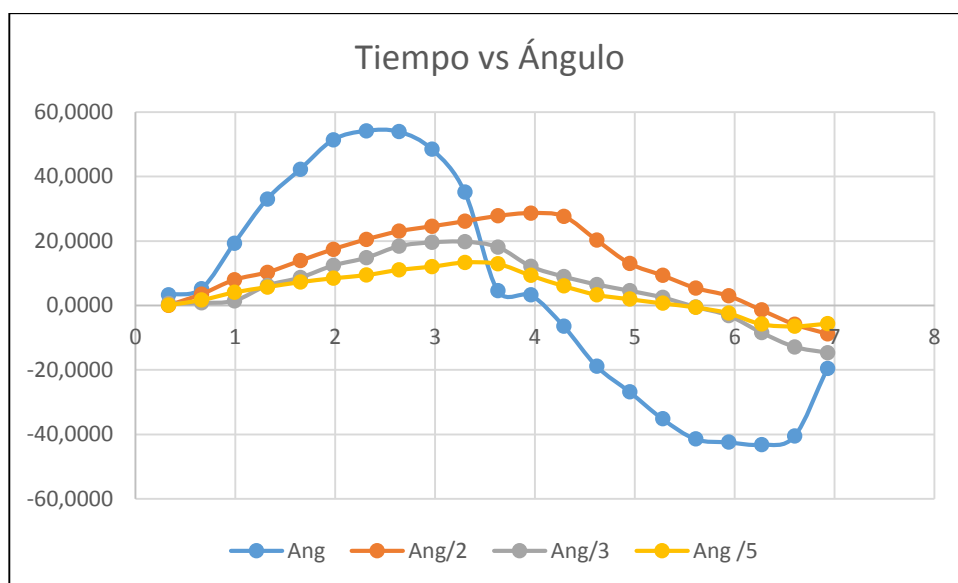


Figura 4. 5 Ángulos enviados cada 330 ms.

Como se ve en grafico si se envía directamente el ángulo dado por el sensor, el robot haría giros demasiado bruscos, lo cual puede llegar a ser peligroso es por eso que se opta por dividir el ángulo para que la tasa de cambio sea menor, que si se envía el dato sin cambios. Para esta configuración es preferible escoger un flujo de datos que no de cambios bruscos, por ello es que *Ang/2* cumple con estos requerimientos mínimos.

En la siguiente prueba se configura con qué rapidez, se envía los datos de ángulo, es decir cada que tiempo es mejor enviar un ángulo para el robot.

Tabla 4. 2 Tiempo de envío de datos

<i>Muestra</i>	<i>330ms</i>	<i>500ms</i>	<i>850ms</i>
1	00.4467	00.4867	00.1268
2	02.0106	03.0914	09.2490
3	05.0559	07.4857	19.9281
4	06.7102	10.8825	26.2130
5	08.5294	14.1236	27.1920
6	10.9955	16.3220	25.3283
7	13.2329	18.4788	19.5553
8	14.6516	21.1672	08.3808
9	17.12.24	23.9648	00.8509
10	19.8128	19.4517	-04.3296
11	21.4603	12.6098	-11.4312
12	22.6249	06.7816	-16.3923
13	22.5282	03.1301	-18.2499
14	18.8682	-03.6475	-20.4273
15	12.8252	-08.5307	-20.5705
16	07.0713	-13.7752	-21.1969
17	03.8917	-17.7752	-07.1468
18	00.4199	-19.9402	01.8802
19	-02.6605	-23.7915	02.1005
20	-06.9185	-22.3670	05.3992
21	-11.6036	-08.4643	12.1036

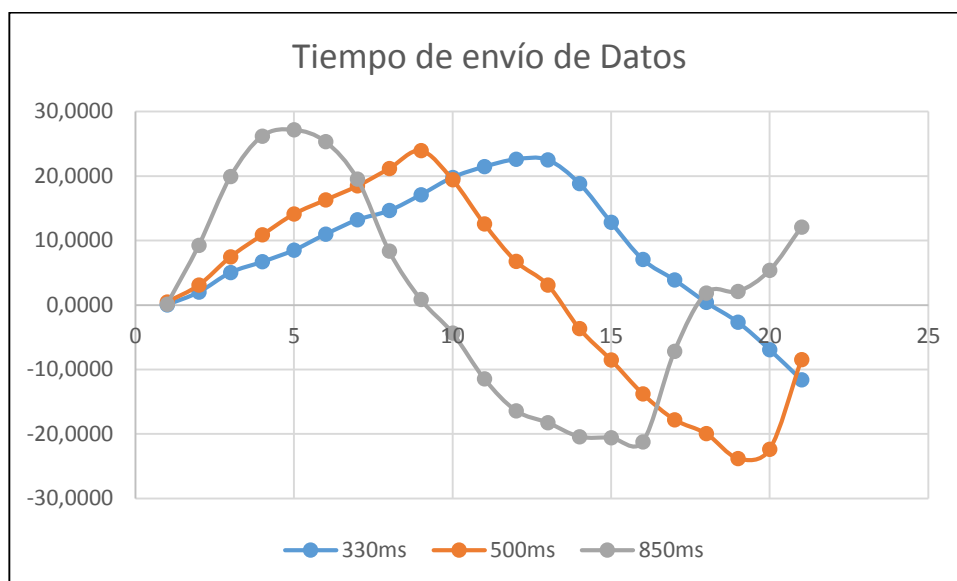


Figura 4. 6 Ángulos para cada periodo

Los mejores resultados con respecto al envío de datos se obtiene con 330 ms., ya que la tasa de cambio de ángulo es más estable y no surgen cambios tan rápidos que hagan que el robot gire bruscamente.

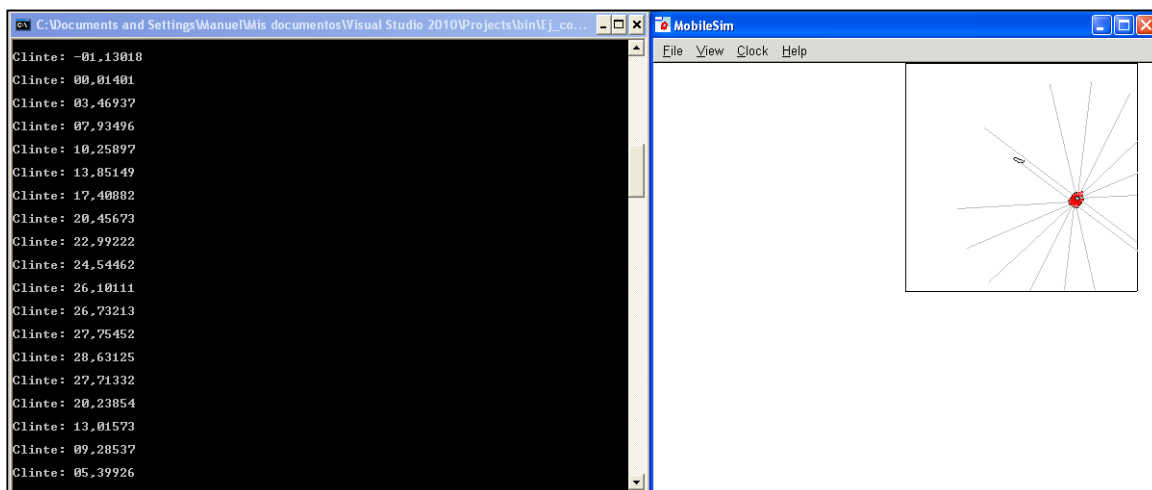


Figura 4. 7 Adquisición de Datos del servidor

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- El diseño del sistema permitió el manejo del robot móvil con gestos manuales, y a través de botones en la interfaz del cliente, con los que se controla acciones aún más específicas.
- La utilidad de los sensores de profundidad, es muy amplia, no solo se puede manejar un robot móvil por medio de gestos sino que también, se pueden diseñar sistemas donde se manipule robots en áreas de riesgo de forma sencilla, al imitar los movimientos humanos.
- La comunicación entre cliente y servidor se estableció correctamente, y es necesario que el envío de datos del sensor se haga en un subproceso asíncrono de la aplicación principal, así se pudo evitar que el modo “conducir” interfiera con los procesos de la aplicación.
- La cámara y el sensor de profundidad son eficientes, pero sin embargo tiene limitantes ya que si el usuario se encuentra muy cerca de él o muy lejos, los datos obtenidos serán erróneos, haciendo que robot se mueva de forma muy irregular y errática.
- El ángulo de rotación tuvo que ser reducido al 50%, pues la función que permite girar al robot admite una velocidad de rotación y no directamente un

ángulo específico, así el ángulo α se divide para dos, porque con este valor se obtienen mejores resultados en cuanto a la respuesta del sistema.

- La utilización de la clase *ArActionGroup*, facilita la programación de los comportamientos del robot haciendo que todos ellos se desarrollen en un solo grupo de acciones, permitiendo que el servidor pueda pasar de un comportamiento a otro inmediatamente, con el llamado de funciones.
- Una ventaja de agrupar los comportamientos es que se puede ampliar la interacción del cliente con el robot programando movimientos adicionales, que complementen las utilidades de la aplicación.

5.2 Recomendaciones

- Es necesario que el cliente se conecte a la red después que el servidor se inicialice, así como también verifique que el sensor y el robot estén conectados al cliente y al servidor respectivamente, para evitar posibles errores de conexión.
- Es preferible utilizar el simulador MobileSim antes que el robot, para aprender a realizar correctamente los movimientos con los que se manipula el P3-AT.
- Es recomendable valerse de una velocidad de traslación relativamente baja, para utilizar la aplicación en espacios reducidos, a medida que el usuario aprenda a controlar el robot con mayor destreza.
- Al momento de ejecutar los movimientos se recomienda no realizar giros de más de 90 grados, sino más bien girar con un ángulo inferior, retroceder y luego terminar el giro, esto evita problemas en la comunicación entre servidor y el robot, debido a la cantidad de información procesada.

- Para utilizar la función “conducir” se recomienda situarse a una distancia entre 1 y 2 metros enfrente del sensor para evitar lecturas erróneas en la adquisición de datos.
- Se debe recordar que los comportamientos se ejecutan de manera exclusiva, por tanto es preferible que antes de activar un nuevo modo primero ejecute la parada del robot y luego el movimiento, sobre todo al momento de cambiar de “conducir” a otro modo, porque el robot seguirá ejecutando el último dato que haya enviado, hasta que se inicie el nuevo comportamiento.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Javier González and Aníbal Ollero, *Estimación de la posición de un robot móvil*. España, 1996.
- [2] Gabriela Andaluz, "Modelación Identificación y Control de Robots Móviles," Escuela Politecnica Nacional, Quito, Tesis de Ingeniería en Electrónica y Control 2011.
- [3] Aníbal Ollero, *ROBÓTICA Manipuladores y robots móviles*. Barcelona, España: Marcombo, 2001.
- [4] Víctor González. (2015) Robots Móviles. [En línea]. http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/15/Tema_5.5.htm
- [5] Antonio Bueno. (2015) Unidad didáctica: "Control y Robótica". [En línea]. http://www.portaleso.com/usuarios/Toni/web_robot_3/robot_indice.html
- [6] Julio Larriba, "Interfaz de teleoperación y control para plataforma robótica móvil," Escuela Técnica Superior de Ingenieros Industriales, Madrid, 2009.
- [7] (2010, Mayo) Las mejores historias de la Historia. [En línea]. <http://laestanteriadearriba.blogspot.com/2010/05/el-robot-goliat-aleman-de-la-ii-gm.html>
- [8] Chris Cole and Jim Wright. (2010, Enero) Drone Wars UK. [En línea]. <http://dronewars.net/aboutdrone/>
- [9] Carlos Zahumenszky. (2014, Enero) Gizmodo. [En línea]. <http://es.gizmodo.com/en-2040-uno-de-cada-cuatro-soldados-estadounidenses-se-1506802417>
- [10] (2014) Robotica de Servicios. [Online]. <http://www.roboticadeservicios.com>
- [11] Lee Ann Obringer and Jonathan Strickland. (2015) How Stuff Works. [En línea]. <http://science.howstuffworks.com/asimo.htm>
- [12] Niklas Galler. (2006, Julio) C-BOT. [En línea]. <http://www.kh-berlin.de/projekt-detail/project/detail/c-bot-11.html>
- [13] Abhijit Jana, *Kinect for Windows SDK Programming Guide*. Birmingham, U.K.: Packt Publishing Ltd, 2012.

- [14] Jarrett Webb and James Ashley, *Beginning Kinect Programming with the Microsoft Kinect SDK*. Nueva York, N. Y.: Apress, 2012.
- [15] Matthew Fisher. (2014) Matt's Webcorner. [En línea]. <https://graphics.stanford.edu/~mdfisher/Kinect.html>
- [16] Anand Muglikar. (2013, Mayo) Accessing IR Video Stream from PrimeSense 3D Sensor using OpenNI and OpenCV. [En línea]. <http://stomatobot.com/primesense-3dsensor-ir-stream/>
- [17] Microsoft. (2015) Kinect for Windows Sensor Components and Specifications. [En línea]. <https://msdn.microsoft.com/en-us/library/jj131033.aspx>
- [18] John MacCormick. (2015) How does the Kinect work? [En línea]. <http://www.cs.bham.ac.uk/~vvk201/Teach/Graphics/kinect.pdf>
- [19] Lukas Häfele. (2012, Marzo) Gesture Control. [En línea]. <http://www.gesture-control.com.eu/>
- [20] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, and Mark Finocchio. (2011) Real-Time Human Pose Recognition in Parts from Single Depth. [En línea]. <http://research.microsoft.com/pubs/145347/BodyPartRecognition.pdf>
- [21] Cyberbotics Ltd. (2015) Webots User Guide. [En línea]. <http://www.cyberbotics.com/dvd/common/doc/webots/guide/section8.3.html>
- [22] MobileRobots Inc., Pioneer 3 Operations Manual with MobileRobots Exclusive Advanced Robot Control & Operations Software, 2006.

ACTA DE ENTREGA

El proyecto fue entregado al Departamento de Eléctrica y Electrónica y reposa en la Universidad de las Fuerzas Armadas – ESPE, desde:

Sangolquí, 14 DE MAYO de 2015

ELABORADO POR:



MANUEL ALEJANDRO VEINTIMILLA CAMPOVERDE
110434879-0

AUTORIDAD



Ing. Luis Orozco MSc.
DIRECTOR DE LA CARRERA DE ELÉCTRICA Y ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL