



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

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

TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA TELEOPERADO PARA UN BRAZO ROBÓTICO “ROBOTIC ARM EDGE” CON SENSOR KINECT PARA WINDOWS A TRAVÉS DE UNA RED WIRELESS

AUTOR: VILLALBA PORTILLA, FERNANDO JAVIER

**DIRECTOR: ING. TIPÁN, EDGAR
CODIRECTOR: ING. IBARRA, ALEXANDER MSc.**

SANGOLQUÍ, AGOSTO 2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICADO

Ing. Edgar Tipán
Ing. Alexander Ibarra

CERTIFICAN

Que el trabajo titulado: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA TELEOPERADO PARA UN BRAZO ROBÓTICO “ROBOTIC ARM EDGE” CON SENSOR KINECT PARA WINDOWS A TRAVÉS DE UNA RED WIRELESS”, realizado por el Sr. Fernando Javier Villalba Portilla con C.I. 1718083841, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas – ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

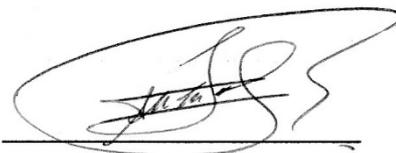
Debido a que se trata de un trabajo de investigación recomiendan su publicación.

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 Fernando Javier Villalba que lo entreguen al Ingeniero Luis Orozco, en su calidad de Coordinador de Carrera.

Sangolquí, 18 de Agosto 2015



Ing. Edgar Tipán
DIRECTOR



Ing. Alexander Ibarra MSc.
CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

DECLARACIÓN DE RESPONSABILIDAD

DECLARAMOS QUE:

El proyecto de grado titulado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA TELEOPERADO PARA UN BRAZO ROBÓTICO “ROBOTIC ARM EDGE” CON SENSOR KINECT PARA WINDOWS A TRAVÉS DE UNA RED WIRELES”, 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. Por consecuente este trabajo es de nuestra autoría.

Sangolquí, 18 de Agosto 2015



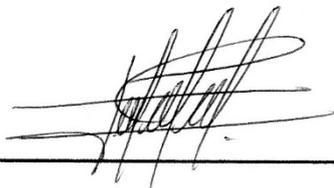
Fernando Javier Villalba Portilla

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Autorizamos a la Universidad de la Fuerzas Armadas – ESPE publicación, en la biblioteca virtual de la Institución, el proyecto de grado titulado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA TELEOPERADO PARA UN BRAZO ROBÓTICO “ROBOTIC ARM EDGE” CON SENSO KINECT PARA WINDOWS A TRAVÉS DE UNA RED WIRELESS” cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, 18 de Agosto 2015



Fernando Javier Villalba Portilla

DEDICATORIA

A ti DI-S JIREH, inmensa es tú Gloria, cada día me has dado nuevas energías, fuerzas, y me has entregado por herencia todo lo que ha pisado la planta de mi pie. Todo mérito y Gloria pertenecen a Dios.

Fernando Villalba

“Así que no temas, porque yo estoy contigo;
no te angusties, porque yo soy tu Dios.
Te fortaleceré y te ayudaré;
te sostendré con mi diestra victoriosa”

Isa 41:10

A mi madre Carmita Portilla mujer valiente y esforzada que cada día me empujo a seguir y no detenerme sin importar cuan altos sean los obstáculos. Dios te bendiga madre amada.

A mi hermano Dennis Portilla inspiración de fuerza y perseverancia, en el cual Dios ha puesto su mano para guiarme, apoyarme. Esfuérzate, se valiente que Dios tiene grandes cosas para tu vida.

Fernando Villalba

AGRADECIMIENTO

A Dios en primer lugar por abrir caminos en medio del desierto, caminos en el cual ha puesto bendición a través de mi familia, Pablo Portilla, Mirian Portilla que gracias a su apoyo incondicional me mostraron con su ejemplo, el esfuerzo que se requiere para cumplir las metas, a través del arduo trabajo y sabiduría en cada decisión.

Agradezco a Dios por el corazón desinteresado de la Dra. Cecilia Arauz y sus hijas Cristina, Daniela, Silvana Domínguez quienes me mostraron que no es necesario llevar el mismo apellido para ser familia, gracias por la confianza, apoyo que pusieron en mí.

Agradezco a Dios por la bendición de amigos Andrés Sosa, Leonardo Mantilla, quienes me acompañaron incondicionalmente el proceso estudiantil, empujando en los momentos difíciles y riendo en las más grandes victorias ahora que el camino profesional nos ha dividido quedan valiosos recuerdos de una amistad desinteresada, y que sin importar el paso del tiempo se conservará.

Fernando Villalba

Agradezco al Ing. Edgar Tipán cuyo ejemplo y sabiduría han guiado todo el proceso de la universidad a través del recorrido de cuatro asignaturas en las cuales han guiado y marcado vida profesional.

Agradezco al Ing. Alexander Ibarra cuya mano amiga supo guiar mis pasos desde el Colegio con su ejemplo, dedicación esfuerzo, sabiduría, y grandes conocimientos que no han sido guardados, sino al contrario han rendido frutos en quienes hemos sido sus estudiantes

Fernando Villalba

ÍNDICE

RESUMEN.....	1
ABSTRACT	2
CAPÍTULO I	3
1. INTRODUCCIÓN	3
1.1 ANTECEDENTES.....	3
1.2 JUSTIFICACIÓN E IMPORTANCIA	5
1.3 ALCANCE	7
1.4 OBJETIVOS.....	9
1.4.1 OBJETIVO GENERAL.....	9
1.4.2 OBJETIVOS ESPECÍFICOS	9
CAPÍTULO II	10
2. MARCO TEÓRICO	10
2.1 ROBÓTICA INDUSTRIAL.....	10
2.1.1 CLASIFICACIÓN SEGÚN SU FUNCIONALIDAD	11
2.1.2 CLASIFICACIÓN SEGÚN SU MORFOLOGÍA	13
2.2 INTERFAZ DE USUARIO.....	16
2.2.1 GUIA GEDIS.....	17
2.3 INTERFAZ NATURAL DE USUARIO.....	19
2.4 RED INALÁMBRICA WIRELESS.....	22
2.4.1 DIRECCIÓN IP	22
2.4.2 ENRUTADORES	23
2.4.3 PROTOCOLO TCP	24
2.5 RASPBERRY PI.....	25
2.5.1.1 RASPBERRY PI PUERTOS GPIO	28
2.5.1.2 RASPBERRY CONFIGURACIÓN LAN.....	29
2.6 ARDUINO.....	30
2.7 COMUNICACIÓN SERIE ARDUINO VS RASPBERRY.....	32
2.8 ACELEÓMETROS.....	33
2.9 ENCODER	34
2.10 DRIVER DE LOS MOTORES.....	35
2.11 DESCRIPCIÓN DEL BRAZO	36
CAPÍTULO III	37

3.	DISEÑO DE HARDWARE Y SOFTWARE	37
3.1	ESQUEMATIZACIÓN Y DESCRIPCIÓN DEL BRAZO ROBÓTICO.....	37
3.1.1	MOTORES DEL BRAZO ROBÓTICO	37
3.1.2	OPERACIÓN DEL TRABAJO	38
3.2	INSTRUMENTACIÓN DEL BRAZO ROBÓTICO	39
3.3	CONSIDERACIONES DEL USO DE KINECT	41
3.3.1	CONSIDERACIONES FÍSICAS	42
3.3.2	CONSIDERACIONES DE SOFTWARE	43
3.4	SISTEMA DE COMUNICACIÓN KINECT-BRAZO ROBÓTICO A PC .	45
3.5	DISEÑO DEL HARDWARE DEL SISTEMA DE CONTROL.....	47
3.6	LEVANTAMIENTO DE PLANOS	50
3.6.1	DISEÑO ELECTRÓNICO.....	50
3.6.2	DISEÑO DE TARJETA.....	51
3.7	DISEÑO DE LA INTERFAZ A TRAVÉS DE VISUAL C#-SHARP	52
	CAPÍTULO IV.....	54
4.	IMPLEMENTACIÓN DE HARDWARE Y SOFTWARE.....	54
4.1	IMPLEMENTACIÓN DE LA INSTRUMENTACIÓN DEL BRAZO.....	54
4.2	IMPLEMENTACIÓN DEL HARDWARE	
	DEL SISTEMA DE CONTROL	57
4.3	IMPLEMENTACIÓN DEL SOFTWARE	
	DEL SISTEMA DE CONTROL	58
4.3.1	CONTROL BRAZO	58
4.3.2	CONTROL ANTE BRAZO.....	61
4.3.3	CONTROL BASE.....	64
4.3.4	CONTROL MUÑECA.....	65
4.3.5	CONTROL GRIPPER	65
4.4	IMPLEMENTACIÓN DE LA COMUNICACIÓN INALÁMBRICA	66
4.3.1	KINECT-PC	67
4.3.2	PC-RASPBERRY	71
4.5	IMPLEMENTACIÓN KIT E DESARROLLO DE	
	APLICACIONES SDK	74
4.5.1	MÉTODO FUERA DE RANGO	76
4.5.2	EVENTO RECONOCIMIENTO DE ESQUELETO	78
4.5.3	FUNCIÓN DIBUJAR Y TRANSMITIR DATOS	80

4.5.4	MEDIR ÁNGULO EN ARTICULACIONES CORPORALES	81
4.6	IMPLEMENTACIÓN DE LA INTERFAZ DE USUARIO.....	83
4.6.1	CARÁTULA.....	83
4.6.2	PÁGINA PRINCIPAL.....	83
	CAPÍTULO V.....	84
5.	PRUEBAS Y RESULTADOS	84
5.1	PRUEBAS	84
5.1.1	ALCANCE DE TRANSMISIÓN Y RECEPCIÓN DE DATOS VÍA WIRELESS.....	84
5.1.2	PRUEBA DE SENSORES PARA CALIBRACIÓN	87
5.1.3	ALCANCE DE VISUALIZACIÓN DEL.....	90
5.1.4	USUARIO A TRAVÉS DE SENSOR KINECT	90
5.1.5	PRUEBA DE DECODIFICACIÓN DE LA TRAMA DE DATOS RECEPTADOS EN RASPBERRY	94
5.2	RESULTADOS	94
	CAPÍTULO VI.....	96
6.	CONCLUSIONES Y RECOMENDACIONES	96
6.1	CONCLUSIONES.....	96
6.2	RECOMENDACIONES.....	97
	REFERENCIAS	98

ÍNDICE DE FIGURAS

Figura 1 Robot Cartesiano, fuente (PÉREZ & SÁNCHEZ, 2013)	14
Figura 2 Robot Cilíndrico, fuente (PÉREZ & SÁNCHEZ, 2013).....	14
Figura 3 Robot Cilíndrico, fuente (PÉREZ & SÁNCHEZ, 2013).....	15
Figura 4 Robot antropomórfico, fuente (PÉREZ & SÁNCHEZ, 2013).....	15
Figura 5 Robot Scara, fuente (PÉREZ & SÁNCHEZ, 2013).....	16
Figura 6 Metodología de desarrollo de Interfaz, fuente (Granollers, 2015)...	17
Figura 7 Distribución de elementos dentro de pantalla, fuente Propia.....	19
Figura 8 Distribución de sensores y actuadores en Kinect, fuente propia.....	20
Figura 9 Arquitectura SDK Kinect. Fuente Propia	22
Figura 10 Enrutador Dlink, fuente Propia	23
Figura 11 Trama de datos enviados por TCP, fuente (Castillo, 2015)	25
Figura 12 Raspberry PI, fuente Propia	26
Figura 13 General Purpose Entrada/Salida GIO, fuente (PE, 2014).....	28
Figura 14 Configuración Wifi externa Raspberry, fuente (Mario, 2015).....	29
Figura 15 Campos configurados, (Bejarano, 2015).....	30
Figura 16 Arduino Duemilanove, fuente propia.....	31
Figura 17 Comunicación Serie Arduino-Raspberry, fuente Propia.....	33
Figura 18 Ejes de sensibilidad Acelerómetro, (México, 2015).....	34
Figura 19 ADXL 335, fuente Propia	34
Figura 20 Cambio de giro Motores, fuente (Carletti, 2015).....	35
Figura 21 Caja reductora Brazo Robótico, fuente (Usuario, 2008).....	37
Figura 22 Espacio de Trabajo Horizontal, fuente (Edge, 2008)	39
Figura 23 Figura 24 espacio de trabajo vertical, fuente (Edge, 2008).....	39
Figura 24 Ubicación sensor ADXL 335 en antebrazo, fuente propia	40
Figura 25 Ubicación sensor ADXL 335 en Brazo, fuente propia.....	40
Figura 26 Diseño-ubicación de encoder en la base, fuente propia.....	41
Figura 27 Diseño encoder, inventor, fuente propia	41
Figura 28 Cable de alimentación Kinect, fuente propia	43
Figura 29 Instalador Librería SDK.....	43
Figura 30 Reconocimiento de puntos de movilidad, fuente (Kleyner, 2013). 44	44
Figura 31. Agregar Librería de Kinect a WPF, fuente propia.....	44
Figura 32 Identificación de la librería instalada a través del SDK.....	45
Figura 33 Esquema Envío / Recepción de datos Kinect, fuente propia.....	46
Figura 34 Esquema envío/Recepción de datos Raspberry.....	46
Figura 35 Esquema de Red, Dirección Ip de cada equipo.....	47
Figura 36 Esquema del ciclo Raspberry-Arduino	48
Figura 37 Control de cambio de giro del motor, (Eduardo, 2015).....	49
Figura 38 Diseño Electrónico de la conexión entre elementos de la tarjeta.. 50	50
Figura 39 Esquema PCB de la tarjeta Controladora, fuente propia.....	51
Figura 40 Vista 3D del diseño completo de la tarjeta, fuente propia.....	51
Figura 41 Presentación de la Pantalla Inicial de la interfaz.....	53
Figura 42 Navegación entre los botones de comando.....	53
Figura 43 Encoder para Base implementado en Impresora 3D	54

Figura 44 Encoder Impreso junto a la base de la estructura	55
Figura 45 Sensor ubicado en el Brazo, fuente propia.....	55
Figura 46 Sensor, ubicado en el antebrazo, fuente propia	55
Figura 47 Estructura con Instrumentación incorporada.....	56
Figura 48 Lectura Sensor Antebrazo valor mínimo aceptable.....	56
Figura 49 Lectura Sensor Antebrazo valor máximo aceptable.....	57
Figura 50 Placa revelada según diseño, fuente propia.....	57
Figura 51 Placa con elementos soldados, fuente propia.....	58
Figura 52 Placa sobrepuesta sobre arduino Duemilanove, fuente propia.....	58
Figura 53 Flujograma de Control Brazo, fuente propia.....	59
Figura 54 Flujograma de lectura de sensores, fuente propia	60
Figura 55 Flujograma Control Ante Brazo.....	62
Figura 56 Flujograma Lectura de Sensores.....	63
Figura 57 Flujograma, control Base, fuente propia.....	64
Figura 58 Flujograma Control muñeca, fuente propia.....	65
Figura 59 Flujograma Control gripper, fuente propia	66
Figura 60 Paso 1 para configurar Ruteador, fuente propia.....	66
Figura 61 Paso 2 para configurar el ruteador, fuente propia	67
Figura 62 Configuración de red PC para Kinect, fuente propia.....	68
Figura 63 Flujograma Cliente visual c#, fuente propia.....	69
Figura 64 Flujograma servidor, fuente propia.....	70
Figura 65 Comunicación establecida, Pc-Raspberry vía Wireless	71
Figura 66 Flujograma Servidor, python, fuente propia.....	72
Figura 67 Implementación de la red Raspberry a medio Wireless	73
Figura 68 Flujograma cliente, python, fuente propia	73
Figura 69 Agregar Librería Kinect, fuente propia.....	74
Figura 70 Método para emplear la esqueletización, fuente propia.....	79
Figura 71 Método para dibujar las partes del esqueleto y enviar datos.....	81
Figura 72 Método para medir los ángulos internos de las articulaciones.....	82
Figura 73 Presentación Carátula de la aplicación	83
Figura 74 Presentación Interfaz HMI, fuente propia.....	83
Figura 75 Ping distancia directa 1[m], fuente Propia.....	84
Figura 76 Ping distancia con obstáculo 1[m], fuente propia.	85
Figura 77 Ping distancia con obstáculo 3[m], fuente propia	85
Figura 78 Ping distancia directa 5[m], fuente propia	86
Figura 79 Ping distancia con Obstáculo 5[m], fuente propia.....	86
Figura 80 Ping distancia con Obstáculo 7[m], fuente propia,	87
Figura 81 Lectura Sensor Antebrazo ángulo sensor eje x	87
Figura 82 Posición Antebrazo ángulo de 112 eje x, fuente propia.....	88
Figura 83 Lectura Sensor Antebrazo ángulo sensor eje x	88
Figura 84 Posición Antebrazo ángulo de 180 eje x, fuente propia.....	88
Figura 85 Lectura Sensor Brazo ángulo sensor eje x.....	89
Figura 86 Lectura Sensor Brazo ángulo sensor eje x, fuente propia.....	89
Figura 87 Detección de usuario 1[m], fuente propia.....	90
Figura 88 Detección de usuario 1.25 [m], fuente propia.....	91

Figura 89 Detección de usuario 1.5 [m], fuente propia.....	91
Figura 90 Detección de usuario 2 [m], fuente propia	92
Figura 91 Detección de usuario 2.5 [m], fuente propia.....	92
Figura 92 Detección de usuario 3 [m], fuente propia	93
Figura 93 Detección de usuario 3.5 [m], fuente propia.....	93
Figura 94 Ping de envío/Recepción de datos, fuente propia.....	94
Figura 95 Seteo de Posiciones	95
Figura 96 Movimiento. Brazo robótico según setPoint.....	95

ÍNDICE DE TABLAS

Tabla 1 Características del sensor KINECT.....	21
Tabla 2 Características Raspberry PI.....	27
Tabla 3 Características puerto GPIO.....	28
Tabla 4 Características Arduino.	31
Tabla 5 Área de trabajo Brazo Robótico	36
Tabla 6 Características principales Brazo Robótico.....	38
Tabla 7 Consideraciones óptimas de funcionamiento Kinect.....	42
Tabla 8 Lógica Digital para cambio de Giro, fuente propia	49
Tabla 9 Valores medidos Vs valores Deseados sensor Brazo.....	61
Tabla 10 Librería Microsoft.Kinect; funciones que se puede acceder.....	74
Tabla 11 Librerías necesarias para la implementación	75
Tabla 12 Funciones para desarrollar el método fuera de rango.....	76
Tabla 13 Funciones para desarrollar el método Reconocer esqueleto.....	78
Tabla 14 Funciones para desarrollar el método.....	80

RESUMEN

El presente documento detalla el diseño y la implementación de un Sistema Teleoperado; el mismo que tiene por objetivo controlar los movimientos de un brazo robótico “Robotic Arm Edge” y el cual consta de un sensor de visión Kinect que permite la detección de los movimientos del usuario, a través del reconocimiento de las articulaciones empleando esqueletización, el envío de datos será vía Wireless para poder acceder al sistema de forma remota.

El proyecto surge de la necesidad de implementar un sistema que permita realizar el control de equipos sin tener contacto directo con los mismos, puesto que hay varios campos que resultan nocivos como Industria Radiactiva, o riesgosa como la minería, o simplemente resulta difícil trasladarse de un lugar a otro para realizar dicha actividad, de esta forma se la puede realizar de forma remota. El sistema consta de una tarjeta programable Raspberry la misma que permite la adquisición de datos de control y envío de los valores detectados por los sensores implementados en el brazo Robótico. La interfaz natural de usuario NUI será el nexo de comunicación visual entre equipos y dispositivos implementados en el sistema, la NUI se la desarrolla en el programa Visual C# con el afán de poder tener fácil acceso a las librerías provistas por Microsoft para desarrollo de aplicación en base a Kinect. El brazo robótico no posee de Instrumentación de detección de movimientos, motivo por el cual se implementará una tarjeta controladora y una tarjeta que permita el monitoreo de los datos presentados por los sensores.

PALABRAS CLAVES:

KINECT

ESQUELETIZACIÓN

WIRELESS

RASPBERRY

INTERFAZ NATURAL DE USUARIO NUI

ABSTRACT

This work is about the design and implementation of a Tele-Operated System which its main purpose is to control the movements of a Robotic Arm Edge. This arm is constituted with a vision sensor Kinect that allows to detect the user's movements by recognizing his joints at using skeletonization or roadmap methods. That data will be sent by wireless access so it is possible to access to a remote system.

So, this work fulfills the necessity of implementing a system that could control certain devices without direct contact as it is in certain areas that are injurious for people as in Radioactive Industries, mining sectors or just in the case it is complicated to move to different areas to do those activities. Therefore, this system has a programmable card Raspeberry that allow the acquisition of data control as the delivery of detected variables by the implemented sensors in the robotic arm. The Robotic User Interfaces (RUI) will be the connection of visual communication between equipment and devices that already are in the system. The RUI is developed by the Visual C# program in order to have easy access to libraries provided by Microsoft for development in apps based on Kinect. The robotic arm does not have the instrumentation for movement detection, for that reason it is necessary to implement a controller card and another that helps the monitory of data presented by the sensors.

KEYWORDS:

KINECT

SKELETONIZATION

WIRELESS

RASPBERRY

NATURAL USER INTERFACE NUI

CAPÍTULO I

1. INTRODUCCIÓN

1.1 ANTECEDENTES

Existen muchas circunstancias en las cuales no es conveniente emplear personas para la realización de algunas labores tales como el manejar elementos radioactivos, o realizar maniobras de investigación de suelos en minería sin riesgo para el operador, por esta razón se ha desarrollado herramientas que puedan reemplazar la mano humana, dentro de estos equipos se encuentran los robots tele dirigidos. (Terrones, 2014)

La influencia de la automatización y la robótica en la sociedad moderna es una realidad, ya sea desde el marco de los sistemas de producción de cada país y sus aspectos económicos, hasta el punto de vista social en lo que respecta a hábitos e incremento del confort y calidad de vida. La investigación y desarrollo en este campo es de vital importancia y marca claramente la diferencia entre países desarrollados y países en vía de desarrollo, es así que la robótica aporta a la sociedad mediante sistemas tele operados la protección y seguridad de un usuario al momento que se tiene que realizar trabajos en zonas inseguras. (Diplomado en Robótica Industrial, 2014)

Kinect es un dispositivo para las consolas Xbox 360, está compuesta por 3 sensores de profundidad una cámara RGB que se encargan de reflejar los diferentes movimientos realizados por una persona, Kinect combina la cámara RGB, cámara de infrarrojos para diferenciar profundidad, Kinect, normalmente, es usado para percepción en 3D de los movimientos humanos, sin embargo, sus usos pueden ir más allá, pues también se puede utilizar para aplicaciones robóticas. Para facilitar el desarrollo de aplicaciones utilizando el Kinect, se ha liberado el código de la librería en LibFreenect para Linux, que

es una plataforma de programación compatible con esta tecnología, Microsoft de igual manera ha liberado la librería Kinect SDK para desarrollo de aplicaciones con Kinect en Visual Studio. (Ingenet, 2014)

Cabe mencionar que el sensor de Kinect es categorizado como un sistema de cámara de profundidad, las cuales están cambiando la visión de los robots y la forma en la que las máquinas perciben el mundo reemplazando la monovisión, estereovisión y otros tipos de indicadores de posición, tales como el láser, los sensores ultrasónicos y de radar.

En computación, una interfaz natural de usuario o NUI, es el término común usado por diseñadores y desarrolladores de interfaces de computadora para referirse a una interfaz de usuario que es efectivamente invisible, o se vuelve invisible con las interacciones aprendidas sucesivamente por sus usuarios. La palabra natural se utiliza porque la mayoría de las interfaces de computadora utilizan dispositivos artificiales de control, cuyo funcionamiento se tiene que aprender. Una NUI se basa en que un usuario puede rápidamente hacer la transición de principiante a experto. Mientras que la interfaz requiere de aprendizaje, por lo tanto, "Natural" se refiere a un objetivo en la experiencia del usuario, que un usuario siente "como algo natural" mientras interactúa con la tecnología. (Sarajevo, 2014)

El brazo robótico ROBOTIC ARM EDGE es de índole educativa, para investigación y desarrollo de prototipos demostrativos de los sistemas teleoperados, también se puede considerar un juguete didáctico, que posee 4 grados de libertad. El movimiento de cada articulación se lo realiza mediante los actuadores propios del Brazo Robótico a través de sistemas reductores con engranes, este es de tipo antropomórfico y su estructura es de plástico y utiliza motores de corriente continua.

Para esta aplicación se plantea el incorporar la instrumentación externamente, cada sensor tiene el objetivo de evaluar la posición de cada articulación del brazo robótico, para poder establecer las respectivas condiciones de control necesarias con el fin de alcanzar una determinada posición.

Se plantea la realización de la investigación y utilización del desempeño de visión artificial mediante Kinect, evaluando el reconocimiento de las articulaciones corporales a través de librerías de esqueletización propias de la librería de Microsoft y programadas mediante la plataforma de Visual c#. Además del reconocimiento visual de movimientos se busca realizar el envío y recepción de datos mediante una red inalámbrica entre la interface humano-máquina que se conecta con el Kinect y una tarjeta programable Raspberry PI. El control y monitoreo de los datos se realizara a través de una tarjeta programable Arduino que será conectada a la Raspberry PI a través de una conexión USB, lo que hace que el sistema sea modular tanto en monitoreo de los sensores así como en parámetros de control para periféricos de salida que para esta aplicación es un brazo robótico "ROBOTIC ARM EDGE".

1.2 JUSTIFICACIÓN E IMPORTANCIA

El proyecto surge de la necesidad de la interacción Usuario-Brazo Robótico sin utilizar dispositivos de mando manual, solo mediante el reconocimiento visual de movimientos de extremidades superiores del operador como medio de comunicación brindando facilidades de operación a las personas que posean discapacidad motriz inferior como por ejemplo, o que simplemente por aspectos de seguridad deban operarlo a distancia.

Entre algunas de las aplicaciones de un brazo robótico están: Cirugías Tele-Operadas, manipulación de herramientas o sustancias nocivas para un

operador. El uso de este tipo de tecnología tiene por objetivo el evitar enfermedades ocupacionales causadas por el contacto directo de la mano o brazo humano, accidentes por análisis e investigación de suelos para desarrollo en campos de minería, por lo que este proyecto representa una introducción a la manipulación de un brazo robótico a partir de los movimientos corporales detectados mediante un reconocimiento visual.

El presente trabajo busca investigar el desempeño de los sistemas de control mediante una red inalámbrica, así como la investigación de la interrelación HUMANO-MÁQUINA a través de un sistema de mando por medio de visión artificial a través de un sensor KINECT. La gran variedad de los sensores incorporados en el Kinect hacen que la comunicación entre el ser humano y computadoras sean más fáciles de desarrollar, alcanzando metas de control más avanzadas a través de mandos visuales o mediante comandos de voz. Considerando el estudio de sistemas TELEOPERADOS mediante un sistema de mando visual se busca que el sistema sea modular y adaptable a cualquier sistema de control, que para este caso se implementa sobre un brazo robótico educativo que posee 4 grados de libertad, cuya estructura es de funcionalidad y movilidad limitada.

El desempeño de cada sistema se lo evaluará con la funcionalidad de cada actuador incorporado en el brazo robótico monitoreado por sensores externos así como controlados por una tarjeta programable encargada de recibir los valores de punto de selección de posición y realizar la respectiva acción de control, considerando los valores de movimientos pedidos por el sistema de Visión Kinect.

1.3 ALCANCE

En vista que el brazo robótico “ROBOTIC ARM EDGE” es para uso académico, cuenta con algunas limitantes como: no cuenta con instrumentación, tiene 4 grados de libertad, limitado en fuerza, estructura de plástico, velocidad de respuesta limitada, por lo que se realizará el análisis de su estructura mecánica para delimitar su zona de trabajo y sus limitaciones de operación y trabajo como velocidad y carga.

También se consideran las limitaciones de movilidad de las extremidades del cuerpo humano en la relación de movimientos detectados por el sistema de visión Kinect, motivo por el cual se evalúa la sensibilidad del sensor en medir el ángulo de apertura de cada articulación en relación a la anterior o a la posterior según corresponda el ángulo

Para el control en lazo cerrado del brazo robótico “ROBOTIC ARM EDGE” se colocarán sensores de forma no invasiva sin cambiar el funcionamiento interno del mecanismo, por lo que se analizarán métodos indirectos para la medición de la posición de cada articulación y se usaran como actuadores los motores DC con los que cuenta actualmente.

Los sensores de Kinect transmitirán datos a una computadora la cual procesará la información y enviará las señales inalámbricas mediante una red Wireless hacia un sistema de control digital, el mismo que permitirá analizar el estado de los sensores haciendo un Reconocimiento de la posición del Brazo Robótico, ejerciendo una acción de control de posición apropiada y devolviendo los valores presentados al computador para ser visualizados a través de una Interfaz Gráfica.

La interfaz Gráfica nos permitirá visualizar la posición del Brazo Robótico en las diferentes coordenadas según la zona de operación, se evalúa la relación medida Kinect vs ángulo censado, obteniendo el respectivo lazo de control.

Para la programación del proyecto se hará uso del Software Microsoft Visual Express 2010 C#-Sharp y sus librerías de uso libre para el público, así como las librerías del SDK del Kinect para Windows, de libre distribución para desarrolladores de sistemas.

La programación para el sistema de control del brazo robótico, así también como la presentación Visual en tiempo real de los movimientos se realizará mediante una tarjeta programable Raspberry Pi, de libre arquitectura y sistema embebido Linux "Raspbian", la misma que usará un Editor de Desarrollo Integrado "IDE" Python.

Debido a la ausencia de entradas análogas en la tarjeta se emplea una tarjeta Arduino de arquitectura abierta, para lectura de la posición de cada articulación a través de sensores análogos, así como las respectivas acciones de control en lazo cerrado.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Diseñar e implementar un sistema teleoperado de un brazo robótico “Robotic Arm Edge” con sensor Kinect para Windows a través de una red Wireless”

1.4.2 OBJETIVOS ESPECÍFICOS

- Desarrollar el programa para el control del sensor Kinect, mediante el kit de desarrollo de software de uso público desarrolladas por Microsoft.
- Obtener la posición 3D del brazo del operador mediante el uso de la librería Skeleton Tracking de Microsoft.
- Transmitir y Recibir datos presentados por los sensores y actuadores del sistema mediante una red Inalámbrica “Wireless”.

CAPÍTULO II

2. MARCO TEÓRICO

2.1 ROBÓTICA INDUSTRIAL

Existen en la actualidad dos criterios diferentes entre Robot Industrial y manipulador, uno de ellos surge del mercado Japonés el cual considera que un robot industrial es cualquier dispositivo mecánico dotado de articulaciones móviles destinado a la manipulación.

La definición más común es la adoptada por la Asociación de Industrias Robóticas la cual dice: "Un **robot industrial** es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas" (Gonzales, 2002)

En la actualidad se maneja dos conceptos que convergen en un fin común entre manipulador y Robot Industrial:

Manipulador: mecanismo formado generalmente por elementos articulados conectados en serie, cuyo objetivo es el agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico.

Robot: manipulador automático, reprogramable, multifuncional, capaz de posicionar y orientar piezas o dispositivos especiales, siguiendo trayectorias programables, para la ejecución de varias tareas. Normalmente posee un elemento de sujeción o herramienta al final de su muñeca. Su unidad de

control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su manipulador final.

La maquinaria rígida dio paso al desarrollo de Robot en base al aumento y mejoramiento de controladores programables, así como el desempeño de servos en bucle cerrado, tales elementos nos permiten establecer con exactitud la posición real de los elementos del Robot mediante dispositivos electrónicos de medición de factores Físicos, sensores.

2.1.1 CLASIFICACIÓN SEGÚN SU FUNCIONALIDAD

MANIPULADORES: Son sistemas mecánicos multifuncionales, con un sistema de control de simple funcionalidad, que permite gobernar el movimiento de sus elementos, de los siguientes modos:

- Manual: Cuando el operario controla directamente la tarea del manipulador puede ser mediante un tablero de control, dispositivo inteligente o de mecánica simple.
- De secuencia fija: cuando se repite su tarea programada de forma invariable, manteniendo su estructura física, y variando su operatividad.
- De secuencia variable: Se pueden alterar algunas funciones de trabajo sean de su estructura física, o de su tarea programada.

ROBOT DE REPETICIÓN O APRENDIZAJE: Son manipuladores que se limitan a repetir una secuencia de movimientos, previamente ejecutada por un operador humano, haciendo uso de un controlador manual. En este tipo de robots, el operario en la fase de enseñanza, se vale de un sistema de programación que puede ser: teclados, joysticks, o un desplazamiento directo de las articulaciones del robot. Los robots de aprendizaje son los más

conocidos, hoy día, en los ambientes industriales y el tipo de programación que incorporan, recibe el nombre de "gestual".

ROBOT CON CONTROL POR COMPUTADORA: Son manipuladores controlados por un computador, que habitualmente suele ser un microordenador. En este tipo de robots, el programador no necesita mover realmente el elemento de la máquina cuando la programa para determinada tarea. El control por computador dispone de un lenguaje específico, compuesto por varias instrucciones adaptadas al robot, es una programación secuencial que se rige a la prioridad de movimientos programados con anterioridad a esta programación se le denomina textual y se crea sin la intervención del manipulador. Su principal ventaja es su funcionalidad por aquello se han posesionado del mercado con facilidad

ROBOT INTELIGENTES: Son manipuladores capaces de relacionarse con el mundo que les rodea a través de sensores, sistemas de visión, y según los datos obtenidos tomar decisiones en tiempo real (auto programable).

De momento, son muy poco conocidos en el mercado y se encuentran en fase experimental. La visión artificial, el sonido de máquina y la inteligencia artificial, son las ciencias que más están estudiando para su aplicación en los robots inteligentes.

ROBOT SOMBRA: Son manipuladores capaces de identificar movimientos corporales a través de sistemas de visión artificial y realizarlos según los datos obtenidos. Además de relacionarse con el mundo que les rodea a través de sensores son capaces de tomar decisiones en tiempo real a través de los comandos preestablecidos mediante un sistema de visión. La principal funcionalidad de estos sensores es poder incorporarlos en ambientes hostiles para el ser humano tales como extracción de minerales y piedras de valor, excavaciones, perforaciones, traslado de objetos de tamaño considerable; mediante sistemas inalámbricos los brazos robóticos son

empleados en maniobras de mucho cuidado tal como son las cirugías clínicas, posicionamiento y traslado de materiales radiactivos o sustancias nocivas para el ser humano.

ROBOT TELEOPERADOS: estos pueden ser manipuladores robóticos o sensores con cierto grado de movilidad, los mismos que a través de un sistema de comunicación inalámbrica es capaz de cumplir las órdenes efectuadas por un sistema de mando. La necesidad del desarrollo de los sistemas teleoperados nace en la premisa de eliminar la limitante de movilidad a través de cable y poder alcanzar acciones de control a grandes distancias tal como se lo ha podido apreciar en sistemas teleoperados enviados al espacio los mismos que reciben las ordenes de control a través de un satélite.

Las áreas actuales de investigación y desarrollo de los sistemas son:

- El manipulador y mandos del robot móvil.
- Las arquitecturas remotas de los sistemas teleoperados
- Control e integración del sistema sensorial.
- La visualización gráfica de las imágenes sobrepuestas.
- Acciones de control mediante sistemas Multisensoriales
- Micromecanismos

2.1.2 CLASIFICACIÓN SEGÚN SU MORFOLOGÍA

Cartesiano: Realiza movimientos definidos mediante articulaciones lineales en cada uno de sus ejes.

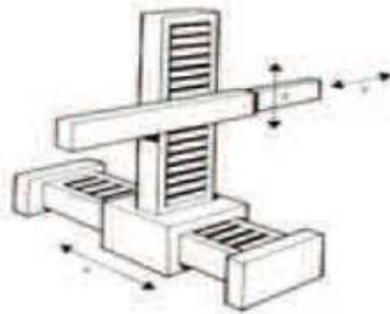


Figura 1 Robot Cartesiano, fuente (PÉREZ & SÁNCHEZ, 2013)

Cilíndrico: Posee un movimiento rotacional sobre su base y dos movimientos lineales, uno vertical y el segundo lateral

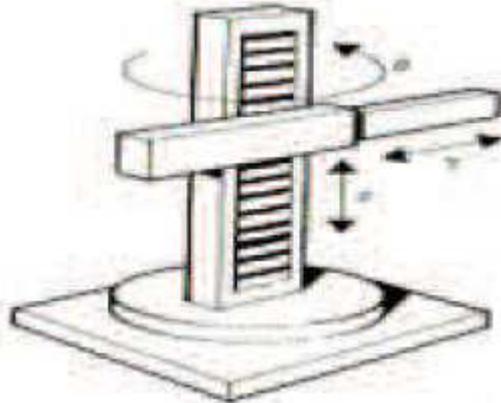


Figura 2 Robot Cilíndrico, fuente (PÉREZ & SÁNCHEZ, 2013)

Polar: posee dos articulaciones rotacionales y un desplazamiento lateral

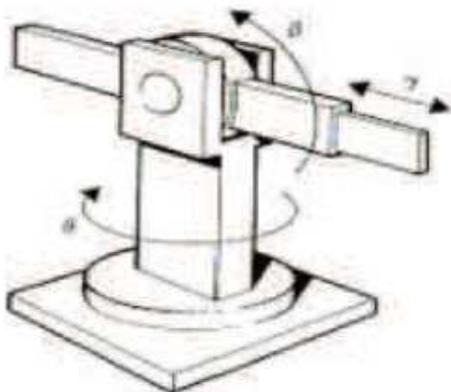


Figura 3 Robot Cilíndrico, fuente (PÉREZ & SÁNCHEZ, 2013)

Antropomórfico: posee una articulación rotacional y dos angulares. Este tipo de sistema Robótico hace similitud al brazo humano

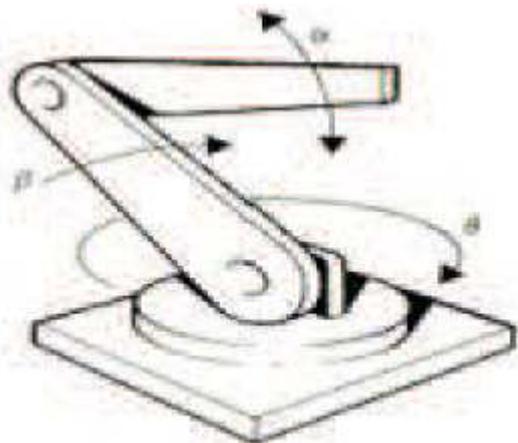


Figura 4 Robot antropomórfico, fuente (PÉREZ & SÁNCHEZ, 2013)

Scara: Posee dos articulaciones superiores con movimiento rotacional, y una articulación al final de la estructura que realiza un desplazamiento lateral.

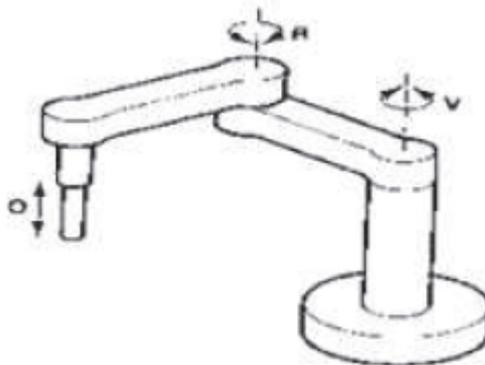


Figura 5 Robot Scara, fuente (PÉREZ & SÁNCHEZ, 2013)

2.2 INTERFAZ DE USUARIO

Las interfaces de usuario son las conexiones que se puede establecer de manera física y con un nivel de utilidad entre un dispositivo o un sistema y como se relacionan con el usuario. La interfaz de usuario es una conexión donde se realiza el intercambio de información entre usuario y máquina. La interfaz de usuario se debe exponer al máximo para no hacerla solo atractiva, sino funcional y sencilla, cuya interacción con la persona sea fácil de entender y manipular.

La interfaz de usuario incluye la relación con dispositivos externos a través de múltiples medios de comunicación que en unos casos puede ser tangibles mediante cables, o abstractos mediante señales electromagnéticas, los elementos de hardware puede ser: Kinect, Raspberry pi, Arduino, los mismos que poseen driver para su comunicación con la PC, lo que permite el reconocimiento de dispositivos, su programación y comunicación con otras aplicaciones ya sea por estos medios o por otras vías de comunicación con el computador.

La función principal de la interfaz es el arranque del sistema, apagado, control de las diferentes variables del sistema, información del estado, para lo cual el sistema debe ser interactivo. Es importante reconocer que la interfaz es un medio para distribuir tareas y realizar operaciones complejas, reduciendo la carga del operario.

2.2.1 GUIA GEDIS

Es una metodología propuesta para diseño de ventanas de interacción HUMANO-MÁQUINA, a través de la guía ergonómica de diseño de interfaces de Supervisión "GEDIS", la misma que ha sido enfocada en ambientes de supervisión computarizada y centralizada.

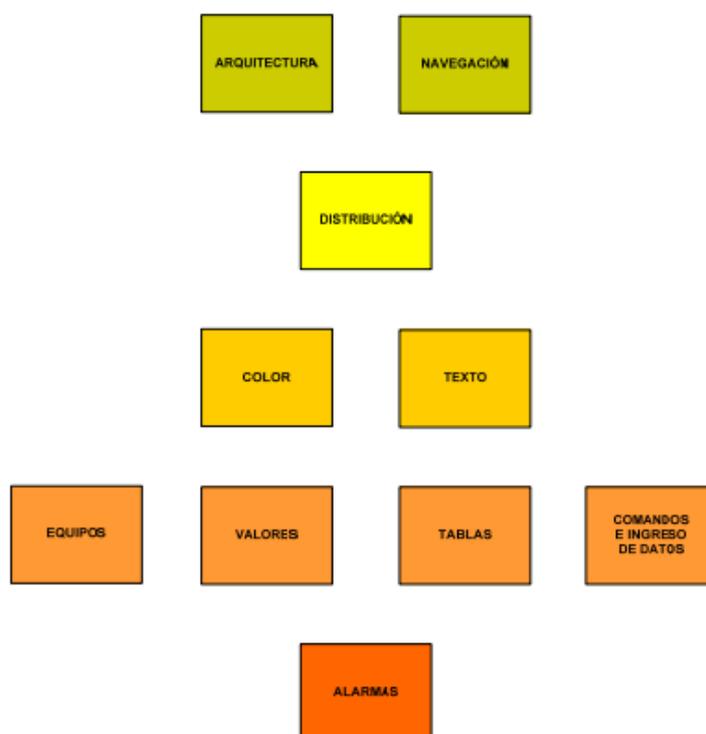


Figura 6 Metodología de desarrollo de Interfaz, fuente (Granollers, 2015)

Se debe en primer lugar considerar los elementos básicos del sistema tales como: Arquitectura, Navegación, así como los estándares de colores, fuentes. La arquitectura se relaciona con el mapa, así como el orden jerárquico entre ventanas, mientras la Navegación establece el árbol del sistema, que establece los criterios de navegación entre ventanas.

El segundo paso de la metodología es la distribución de ventanas, en el cual se establece un estándar para cada pantalla, en el cual se debe considerar ubicación del título, hora, fecha, logo de la empresa, ubicación de alarmas, ubicación de tablas. En la distribución de elementos en la pantalla se debe considerar aspectos físicos relacionados con la vista y la mente, tal como es el movimiento del ojo de izquierda a derecha, la información superior es la más importante, el centro de la pantalla de alta importancia.

La navegación consiste en la facilidad que tiene el usuario de trasladarse de una pantalla a otra, ya sea por botones, listas desplegables, hipervínculos, los botones básicos que se recomienda: adelante, atrás, inicio, alarmas, etc. El uso de Colores es fundamental para el desarrollo de pantallas pues debe mantener una relación conservadora, convencional y consistente; en esta etapa se debe definir el estatus de los equipos sean marcha, paro, falla, etc. Así como colores de texto, títulos, etiquetas, alarmas, se debe limitar el uso de colores de 4 a 7 por pantalla asegurándose que sean diferenciables entre sí. No se recomienda el uso de combinaciones como Rojo-Azul, Rojo-Verde, Amarillo-Blanco, Verde-Azul, el color debe utilizarse para indicar calidad más no cantidad; los colores de fondo deben ser neutros tales como gris, arena, azul.

Es importante regular el uso de textos para informar al operador respecto al estado del proceso entre las principales características se debe considerar el tipo de fuente, la alineación, el espaciado, los acrónimos y abreviaturas.

Para el diseño de la plantilla no se debe utilizar más de tres fuentes, no usar más de tres tamaños de la misma fuente, el tamaño de la fuente debe ser tal se puede leer a distancia por el operador. Procurar combinar mayúsculas y minúsculas; es indispensable contrastar el color de texto con el color de fondo para poder tener una mejor apreciación del texto y así evitar aglutinamientos.

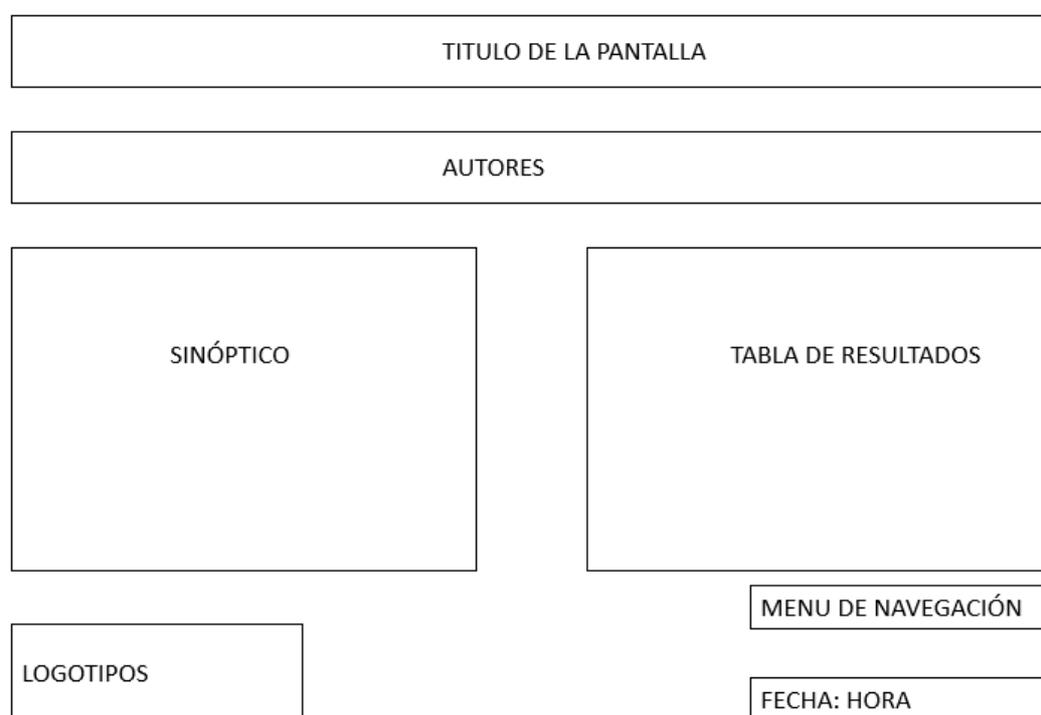


Figura 7 Distribución de elementos dentro de pantalla, fuente Propia

2.3 INTERFAZ NATURAL DE USUARIO

Las interfaces de usuario fueron desarrolladas como respuesta a la necesidad de facilitar la interacción humano-máquinas. La evolución de las interfaces de usuario nos ha llevado hasta las interfaces gráficas, lo que ha permitido que la computadora y los dispositivos móviles se posicionen como herramientas de la vida cotidiana. La concentración de la evolución en el desarrollo de hardware y software más poderoso y con mayores prestaciones,

ha llevado a la investigación para el desarrollo de interfaces de usuario más poderosas, fáciles de usar y con la característica de que éstas sean las que se adapten al usuario y no viceversa. Esta búsqueda ha llevado a la investigación de la Interacción Natural (IN) y de las Interfaces Naturales de Usuario (NUI).

El objetivo de la NUI es permitir al usuario interactuar con la computadora como si ésta se tratase de otro humano, evitando la necesidad de saber el manejo previo del dispositivo, concentrándose en el uso de las habilidades naturales del ser humano, tales como gesticulación, movimientos específicos, e incluso el uso de la voz, para ser utilizados como datos de entrada a la interfaz. Gracias al sensor Kinect, la consola Xbox 360 de Microsoft permite la interacción con videojuegos sin más que con movimientos del cuerpo y voz, eliminando el uso del control de la consola.



Figura 8 Distribución de sensores y actuadores en Kinect, fuente propia

El Kinect entrega imágenes de profundidad que permiten interpretar el ambiente y estimar el movimiento del usuario con la ayuda de bibliotecas de software presentadas por Windows a través del SDK. Comprende de un arreglo de sensores que permiten la interacción entre el programa y el usuario y este posee: cámara de color, emisor infrarrojo, sensor infrarrojo de

profundidad, arreglo de micrófonos, y led de estado, ubicados como muestra la figura 8.

En la tabla a continuación se describen las principales características del Kinect.

Tabla 1
Características del sensor KINECT.

DETALLE	CARACTERÍSTICA
Cámara de color	soporta hasta 30 cuadros por segundos
Resolución	640x480 pixeles, el máximo de resolución es de 1280x960. La resolución soportada es de 640x480, 320x240 y 80x60 pixeles, la información obtenida se muestra utilizando gradientes para distinguir la profundidad entre objetos tridimensionales
Ángulos de visualización	43 grados en vertical y 57 en horizontal.
Sensores infrarrojos	Sensores de profundidad que permite obtener la distancia entre un objeto y la cámara, empleando métodos de luz estructurada
Micrófonos	Cuenta con cuatro elementos que captan la dirección de la onda de sonido de acuerdo a su ubicación.

Fuente (ComuSoft.com, 2015).

La arquitectura del kit de desarrollo de software (SDK) para Kinect, se conforma por tres partes principales, la primera se basa en su funcionamiento en el control del hardware, de los sensores o cámaras con las que cuenta Kinect, la segunda el control del arreglo de micrófonos, la conexión USB y los motores que permiten la alineación del dispositivo, y finalmente el desarrollo de la aplicación a través de una plataforma de trabajo.

El nivel central refiere a la comunicación, en la cual almacena los datos que los sensores detectan. Finalmente se tiene la interacción entre la aplicación en c# y el usuario.

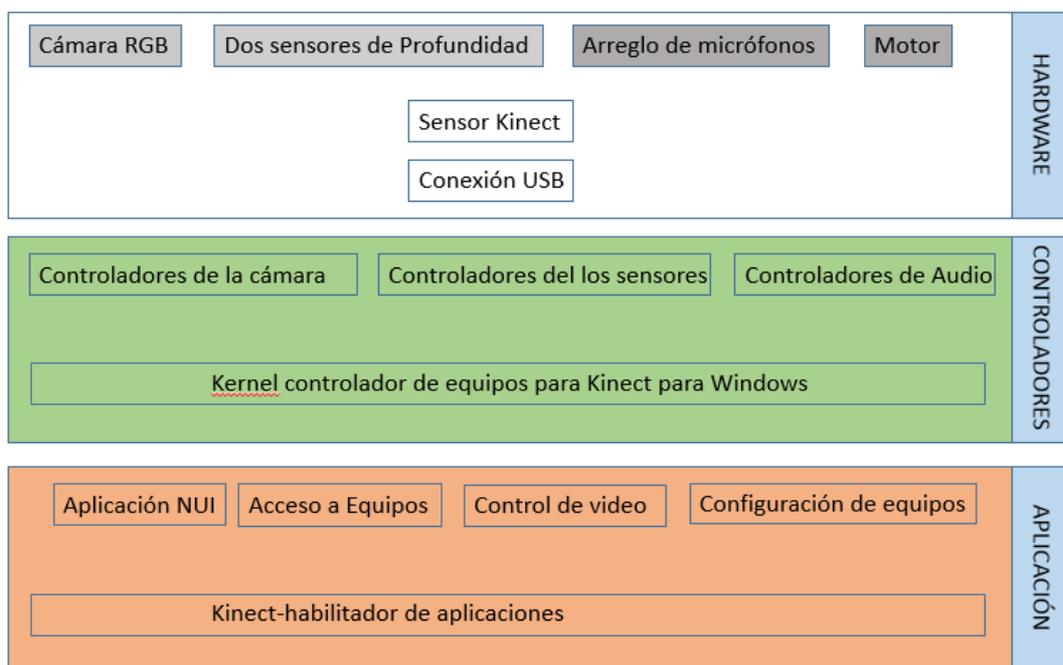


Figura 9 Arquitectura SDK Kinect. Fuente Propia

2.4 RED INALÁMBRICA WIRELESS

Las redes LAN (redes de área local) inalámbricas le permiten conectar sus computadoras sin alambres ni cables. Las redes LAN inalámbricas usan ondas de radio que permiten que los dispositivos móviles se conecten y comuniquen dentro de un área determinada.

2.4.1 DIRECCIÓN IP

Conocido con el acrónimo para "Internet Protocol", es un número irrepetible que consiste en un conjunto de 4 dígitos del 0 al 255, para dar

direcciones a redes no conectadas a internet, se reservan direcciones privadas, estas direcciones pueden ser usadas por cualquier persona pero no pueden ser ruteadas a internet Global.

- **192.168.0.0** - 256 clases C o 65,536 direcciones
- **172.16.0.0** - 256 clases B o 4,194,304 direcciones
- **10.0.0.0** - una clase A o 2,097,152 direcciones

2.4.2 ENRUTADORES

El enrutador permite comunicar diversos equipos en una misma red capaz de intercambiar información presente en la red, el enrutador generalmente es el intermediario entre la red local y la red privada del proveedor de internet, motivo por el cual los enrutadores utilizan dos tipos de direcciones IP, una pública que es la que nos entrega el proveedor de internet que puede ser estática es decir que nunca varía, o una dinámica que cambia aleatoriamente en función de las necesidades del proveedor. Y el segundo tipo que vendría ser una red privada que nos servirá para centralizar la comunicación entre los equipos.



Figura 10 Enrutador Dlink, fuente Propia

Para el desarrollo de la aplicación se desarrolla la red para dos nodos, uno que es el servidor en el cual se encuentra conectado el Brazo Robótico que

representa el equipo a controlar inalámbricamente y el segundo que pertenece al Kinect que representa el cliente conectado al computador.

En vista que se requiere enviar y recibir los datos en tiempo real para un manejo aceptable del brazo robótico se considera la utilización del protocolo TCP a través de *sockets*; los *sockets* son un tipo especial de fichero que utiliza un proceso para pedir un servicio de red al sistema en el que se está aplicando, la dirección de *socket* está conformada por tres partes (protocolo, dirección local, proceso local), considerando que el sistema teleoperado es bidireccional se debería emplear dos procesos consecutivos en el cual consta de: (protocolo, dirección local, proceso local, dirección externa, proceso externo)

2.4.3 PROTOCOLO TCP

El objetivo del Protocolo de Control de Transmisión “TCP” es proveer un flujo de bytes confiable, sobre una red que probablemente tenga pérdidas de datos, en el transporte parte un flujo de bytes y los manda como datagramas de IP , para obtener el servicio de TCP tanto el emisor como el receptor, tienen puntos terminales de la conexión (*sockets*). La dirección de un *socket* es la dirección IP y la puerta de enlace que es un valor de 16 bits, se puede usar *sockets* de cada extremo, se puede usar *sockets*. (Castillo, 2015)



Figura 11 Trama de datos enviados por TCP, fuente (Castillo, 2015)

Las conexiones de TCP son punto a punto y full dúplex, no preservan los límites de mensajes, cuando una aplicación manda datos a TCP, TCP puede mandarlos inmediatamente o almacenarlos, en TCP se mandan datos en segmentos, un segmento tiene un encabezamiento de 20bytes, TCP cuando recibe los datos tiene un acuse de recibo. Los problemas que se pueden generar son pérdidas de datos durante el acuse de recibo, los segmentos llegan fuera de orden, congestión de la red. (Castillo, 2015)

2.5 RASPBERRY PI

Es un ordenador cuyo tamaño es similar al de una tarjeta de crédito, el mismo que puede interactuar mediante periféricos de entrada tal como se lo hace con un Ordenador normal (mouse, teclado), se puede visualizar su contenido a través de una pantalla que puede ser conector HDMI o mediante una TFT.

Posee un sistema onChip que contiene un procesador ARM que corre a una velocidad 700 MHz, un procesador gráfico VideoCore IV y una memoria RAM de hasta 512 MB. La tarjeta nos permite acceder a un sistema operativo

mediante una tarjeta SD la misma que comparte y divide según necesidades de la tarjeta tanto para almacenamiento temporal como para procesos.



Figura 12 Raspberry PI, fuente Propia

Cuenta con una serie de conexiones o puertos de entrada y salida de propósito general o GPIO, que nos permite hacer todo tipo de interfaces de hardware, como el control de otros dispositivos, El sistema operativo preinstalado en la SD que es de libre distribución y generalmente son con base Linux como es el caso de Raspbian que a su vez es derivado de la versión Linux. Por su alta versatilidad y facilidad de programar posee múltiples accesorios adicionales para mejor la comunicación o mejorar la utilidad del equipos para diversas aplicaciones tal como es el caso de la cámara o el reemplazo del monitor mediante una tarjeta TFT que se accede a una red inalámbrica mediante un adaptador WIF por USB.

Tabla 2
Características Raspberry PI

DETALLE	CARACTERÍSTICA
Procesador	Broadcom BCM2835 SoC full HD (igual que el modelo B)
RAM	512 MB SDRAM 400 MHz
Almacenamiento:	Tarjeta microSD, de capacidad variable según requerimientos de usuario
USB	Cuatro puertos USB 2.0
Energía	600mA hasta 1.8A a 5V
Pines GPIO	40

(raspberrypi.org, 2015)

2.5.1.1 RASPBERRY PI PUERTOS GPIO

GPIO (General Purpose Input/Output), es un Sistema de propósito general el cual se puede usar como entradas y salidas para usos múltiples, la GPIO representa la comunicación entre Raspberry y el mundo exterior. Todos los pines son tipo “unbuffered”, es decir, que no disponen de buffers de protección para corrientes o voltajes, no todos los pines tienen la misma función.



Figura 13 General Purpose Entrada/Salida GIO, fuente (PE, 2014)

Tabla 3
Características puerto GPIO

DETALLE	CARACTERÍSTICA
PINES DE ALIMENTACIÓN	Pines de 5v, 3v (limitados a 50mA) y tierra GND, según conexión puede servir como fuente de alimentación. O puede recibir alimentación exterior
DNC (DO NOT CONNECT)	En futuras implementaciones son utilizadas para otros fines, pero actualmente sirve y son marcadas como GND
GPIO NORMALES	Son conexiones configurables para programables
GPIO ESPECIALES	Se encuentran algunos pines destinados a una interfaz UAR, con conexiones TXD y RXD,

(PE, 2014)

2.5.1.2 RASPBERRY CONFIGURACIÓN LAN

Se tiene dos principales vías de configuración WIFI, una es la interfaz gráfica de usuario en la cual solo escaneamos las redes cercanas y una vez identificada la red buscada, se selecciona conectar.



Figura 14 Configuración Wifi externa Raspberry, fuente (Mario, 2015)

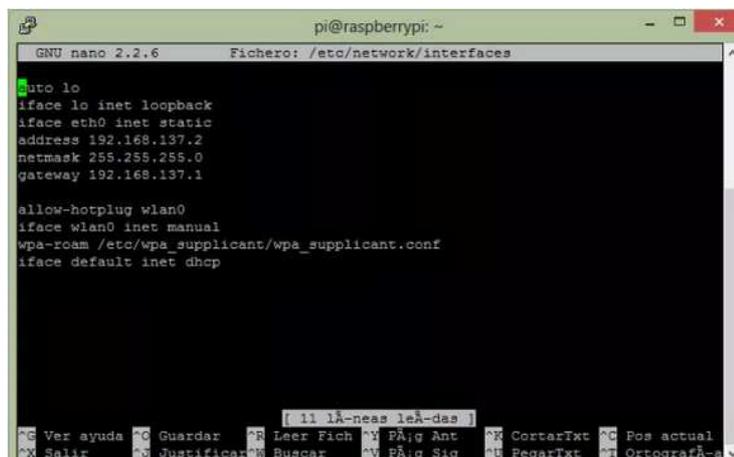
Un segundo método es la conexión con el empleo de una línea de comandos que primero se debe actualizar la tarjeta con los últimos drivers para lo cual se emplea el siguiente comando:

“sudo apt-get update sudo apt-get upgrade sudo apt-get autoremove” (Godot.Be, 2014)

Luego se crea un respaldo antes de empezar hacer cambios

“sudonano/etc/wpa_supplicant/wpa_supplicant.conf/etc/wpa_supplicant/wpa_supplicant.conf.bak”

Para finalmente abrir la interface a configurar “sudo nano /etc/network/interfaces”, en el cual una vez abierto es necesario reemplazar dhcp por static, y asignamos la IP al equipo.



```

pi@raspberrypi: ~
GNU nano 2.2.6 Fichero: /etc/network/interfaces
auto lo
iface lo inet loopback
iface eth0 inet static
address 192.168.137.2
netmask 255.255.255.0
gateway 192.168.137.1

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
  
```

Figura 15 Campos configurados, (Bejarano, 2015)

2.6 ARDUINO

Es una tarjeta electrónica, programable, de código abierto y estructura flexible hardware y software, la plataforma de programación varía según el modelo de tarjeta, Arduino Development Environment se basada en “processing “, la placas pueden ensamblarse a mano o encargadas pre ensambladas, los diseños de referencia del hardware están disponibles en la internet bajo licencia “Open-source”, por lo que se puede adaptar a las necesidades del desarrollador. Arduino principalmente encapsula todas las condiciones de programación del microcontrolador y las encierra en un paquete fácil de utilizar y programar, el código de programación tiene base en JAVA motivo por el cual se lo puede utilizar en diferentes Sistemas Operativos; arduino está basado en placas Atmega128 y Atmega328 de atmel.

Las tarjetas son de diversas características dependiendo de la aplicación o necesidades del desarrollador entre las principales del modelo básico es decir “Arduino Uno” y caso similar “Arduino Duemilanove”.

Tabla 4
Características Arduino.

DETALLE	CARACTERÍSTICA
Microcontrolador	Atmega 328P
Voltaje de Operación	5V
Voltaje de Entrada (recomendado)	7-12V
Voltaje de Entrada (limite)	6-20V
Salidas Digitales	14 de los cuales 6 son de PWM
Entradas Analogas	6
Corriente DC por pines entrada/salida	20mA
Corriente DC por pines 3.3V	50mA
Memoria Interna	32KB (atmega 328P), del cual 0.5K es usado por software
SRAM	2KB (Atmega 328P)
EEPROM	1KB(Atmega 328P)
Velocidad del reloj	16 MHz
Longitud Horizontal	68.6 mm
Longitud Vertical	53.4 mm
peso	25 g

(Cuartiles, 2015)

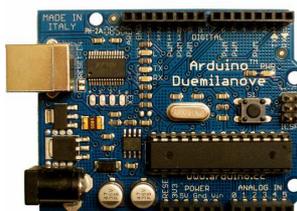


Figura 16 Arduino Duemilanove, fuente propia

2.7 COMUNICACIÓN SERIE ARDUINO VS RASPBERRY

Un puerto es el nombre que se denomina a la interfaz físico-virtual, que permiten comunicarse entre dos ordenadores o dispositivos. Un puerto serie envía la información mediante una secuencia de bits, para lo cual la comunicación serial emplea dos conectores el transmisor TX (Transmisor) y RX (Receptor), y un conector de referencia de Voltajes. Habitualmente la comunicación serie se emplea cuando es necesario transmitir grandes volúmenes de datos, que por lo general todos los ordenadores disponen del popular USB (Universal Serial Bus), la comunicación serie se la conoce adicionalmente como UART (universally asynchronous receiver/transmitter), que es una unidad que poseen ciertos procesadores, encargados de realizar la conversión de los datos a una secuencia de bits y transmitirlos o recibirlos a una velocidad determinada. El voltaje de señal que recorre es TTL (transistor-transistor lógico), eso significa que la comunicación se la realiza mediante variaciones de señal entre 0v donde (V_{cc} suele ser 3.3V y 5V), pero otros sistemas emplean variaciones de voltaje (-13v-+13v) como es el caso típico del RS232.

Todas las placas arduino disponen de una unidad UART la misma que trabaja a TTL 0/5V por lo que son directamente compatibles con la conexión USB.

En Raspberry lo más adecuado es trabajar la programación desde IDE Python, por lo que el primer paso desde programación es instalar las librerías de comunicación serie a través de la línea de comandos que ejecuta la siguiente instrucción: "sudo apt-get install python-serial". Una vez instalado desde internet la librería se debe identificar el puerto de comunicación con arduino para lo cual en la línea de comandos ejecutamos "ls /dev/tty*", y al

conectar la tarjeta arduino podremos identificar el puerto de comunicación entre ambos dispositivos que para el caso `/dev/tty/USB0`

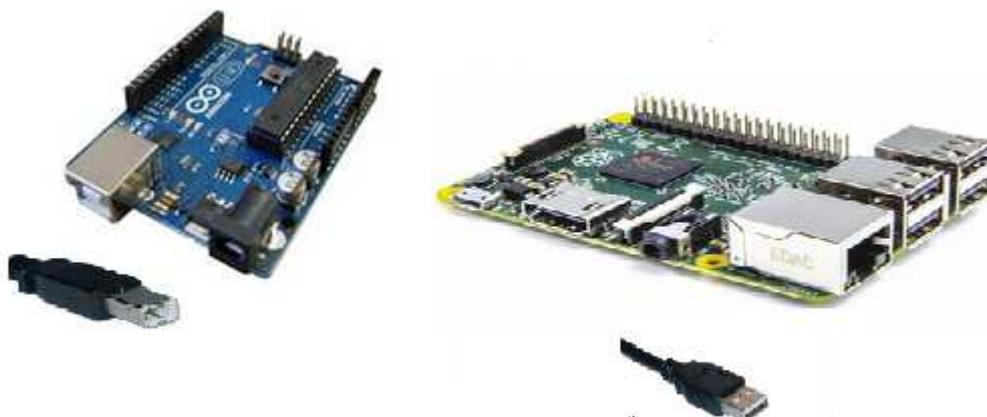


Figura 17 Comunicación Serie Arduino-Raspberry, fuente Propia

2.8 ACELEÓMETROS

Son dispositivos electromecánicos que destacan las fuerzas de aceleración ya sea estática o dinámica en función de la gravedad. El modelo `adxl335` posee 3 ejes, en el cual dependiendo la inclinación del sensor, mide la variación de la fuerza censada en función de la gravedad, generalmente los acelerómetros tienen placas capacitivas que se encuentran unidas por resortes, como estas placas se mueven en relación el uno con el otro la capacitancia entre ellos varia por lo que se puede determinar una posición de inclinación para cada eje, por lo que otra de sus funcionalidades es el poder emplearlo como giroscopio.

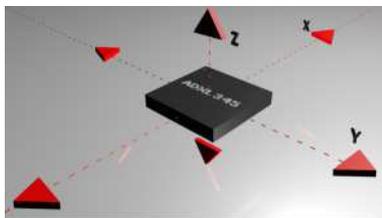


Figura 18 Ejes de sensibilidad Acelerómetro, (México, 2015)

El modelo adxl335 entrega valores análogos de cada eje, pero modelos actuales entrega la información mediante conexión modulada por ancho de pulso, los acelerómetros que poseen salida análoga entregan un valor de voltaje correspondiente a la variación de capacitancia de cada uno de sus ejes, la alimentación es de 3.3V, el rango de sensibilidad está en tres grados.



Figura 19 ADXL 335, fuente Propia

2.9 ENCODER

Principalmente el encoder se compone de un disco conectado a un eje giratorio, por lo general el disco está hecho de plástico con divisiones translucidas y otras divisiones totalmente oscuras que evitan el paso de la luz, las áreas bloqueadas se encuentran divididas de forma radial. Conforme el disco gira se emite un haz de luz el mismo que es detectado por un sensor foto-transistor que permite, a medida que la luz cruza, se generen pulsos digitales. El grado de precisión es alto lo que permite controlar la posición de giro del rotor

Se puede encontrar diversos tipos de encoders, que se emplean en diferentes aplicaciones pero el más común es el encoder óptico, consta de una fuente emisora de luz, un disco giratorio y un sensor foto detector, las señales que se producen por la activación y desactivación de la luz permiten al controlador determinar el ángulo exacto del eje.

2.10 DRIVER DE LOS MOTORES

El driver controlador de motores es un integrado formado por 4 circuitos para manejar las cargas de potencia media, el controlador puede manejar corrientes hasta de 600mA con una tensión entre 4.5 y 36V. Cada circuito tiene una configuración Darlington haciendo que el sistema sea robusto para controlar el sentido de giro de los motores empleándolo como un puente H. La salida del integrado tiene un diseño mediante diodos que le permite conectarse directamente con la carga. El circuito de salida se activa por pares por medio de una señal de activación de los transistores internos con un nivel TTL de 5v. Por medio de un control apropiado de las señales de entrada, se puede tener el manejo de potencia de un puente H, permitiendo un control bidireccional del motor.

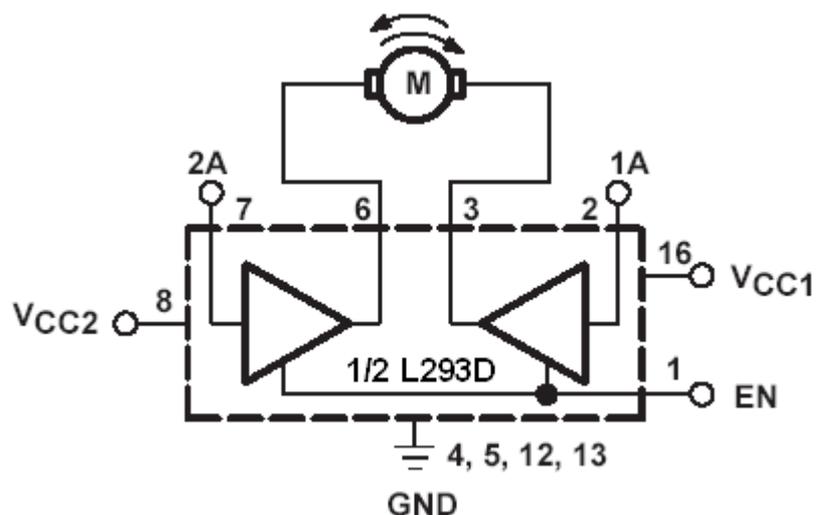
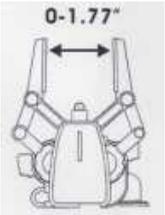
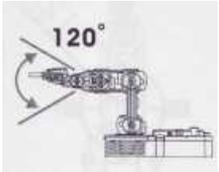
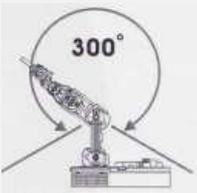
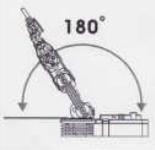


Figura 20 Cambio de giro Motores, fuente (Carletti, 2015)

2.11 DESCRIPCIÓN DEL BRAZO

Tabla 5
Área de trabajo Brazo Robótico

Pinza	Con motor y caja reductora mediante engranes para movimientos laterales	
Base	Movimiento libre de 270 grados	
Muñeca	Movimiento libre de 120 grados	
Brazo	Movimiento libre de 300 grados	
Antebrazo	Movimiento libre de 180 grados	

CAPÍTULO III

3. DISEÑO DE HARDWARE Y SOFTWARE

Se busca diseñar un sistema independiente tanto en Hardware como en Software en el cual se pueda readecuar a otro tipo de sistemas sin necesidad de causar variaciones en la estructura física del Brazo Robótico. Se procura realizarlo de forma inalámbrica con el objetivo de alcanzar mayores distancias a través de una red local o en un futuro se pueda acondicionar el envío y recepción de datos desde internet.

Con el uso de Dispositivos de visión artificial y acciones de control de forma teledirigida se diseña un sistema que pueda facilitar actividades a personas que posean dificultades de trasladarse de un lugar a otro.

3.1 ESQUEMATIZACIÓN Y DESCRIPCIÓN DEL BRAZO ROBÓTICO

3.1.1 MOTORES DEL BRAZO ROBÓTICO

Cada articulación del brazo robótico tiene un motor de bajo consumo el mismo que varía su torque y movimiento mediante una caja reductora tal como se presenta en la figura 22.

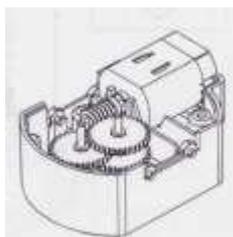


Figura 21 Caja reductora Brazo Robótico, fuente (*Usuario, 2008*)

El movimiento del motor proviene de la acción del campo magnético vs el campo eléctrico, al fluir corriente eléctrica a través de un conductor reacciona con el campo magnético produciendo fuerza de modo que esa fuerza es directamente proporcional al producto entre el campo magnético por la corriente y la longitud del conductor.

$$F = \beta * i * l$$

Del presente proyecto cada articulación posee un motor con caja reductora pero ninguno de ellos posee cambios ni de estructura ni de componentes, a excepción de la pinza que posee una caja reductora de movimiento lateral a diferencia de las anteriores que tienen un movimiento rotacional.

Tabla 6
Características principales Brazo Robótico.

DETALLE	CARÁCTERÍSTICAS
VOLTAJE DE OPERACIÓN	3-6 V DC
CORRIENTE SIN CARGA	255 A
VELOCIDAD	12340 RPM VARIACION DEL 12%
TORQUE CON BLOQUEO	60cm*g
LARGO	35 mm
ANCHO	25 mm
ALTURA	32.5 cm

3.1.2 OPERACIÓN DEL TRABAJO

Debido al peso, movimiento y estructura se ha delimitado el rango de trabajo de cada articulación con el objetivo de poder tener un control más

eficiente de cada articulación: el brazo robótico posee una pinza que puede abrirse y cerrarse mediante un motor con caja reductora lateral que permite un cambio de giro de horizontal a vertical, cuenta con una rotación del hombro de 165 grados, antebrazo 100, brazo 90 grados, muñeca una rotación libre de 120 grados. Por tal motivo tiene un alcance vertical de 35 cm y horizontal un alcance de 30 cm. La pinza tiene una longitud de agarre de 0 a 4 cm.

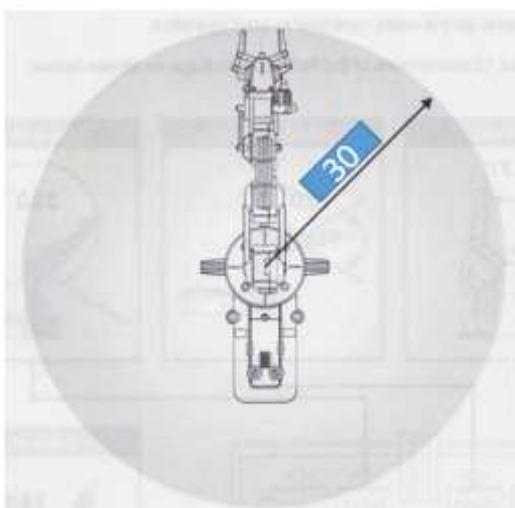


Figura 22 Espacio de Trabajo Horizontal, fuente (Edge, 2008)

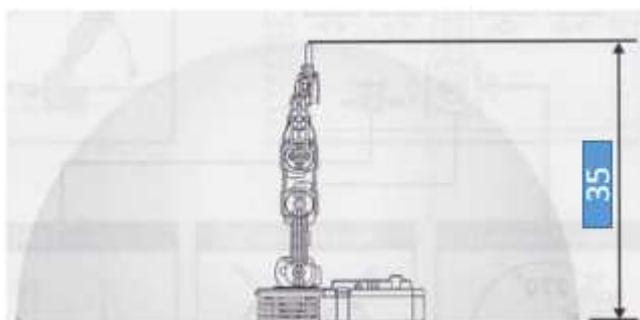


Figura 23 Figura 24 espacio de trabajo vertical, fuente (Edge, 2008)

3.2 INSTRUMENTACIÓN DEL BRAZO ROBÓTICO

El brazo robótico no consta de instrumentación que indique el estado en el que se encuentra, motivo por el cual se añade dos tipos de sensores: el primero, un acelerómetro ADXL335 que mediante la variación de la posición

en uno de sus nos entrega de forma análoga el ángulo de inclinación de la articulación en la que se encuentre, el sensor será colocado en el antebrazo y brazo del robot en los soportes que unen las dos cajas reductoras, se procura colocarlo en el centro de los soportes para evitar variaciones en el giro a causa del centro de gravedad.



Figura 24 Ubicación sensor ADXL 335 en antebrazo, fuente propia



Figura 25 Ubicación sensor ADXL 335 en Brazo, fuente propia

El segundo sensor se considera al uso de sistema de encoder, el cual está conformado por un disco dentado encoder junto a un sensor foto-transistor, que permiten el conteo de pulsos, cada pulso equivale aproximadamente a 3

grados de giro. Se ubicaran dos sistemas de este tipo, uno en la base y el segundo en la muñeca.

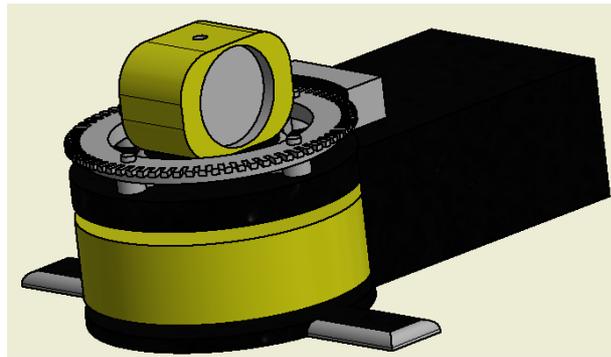


Figura 26 Diseño-ubicación de encoder en la base, fuente propia

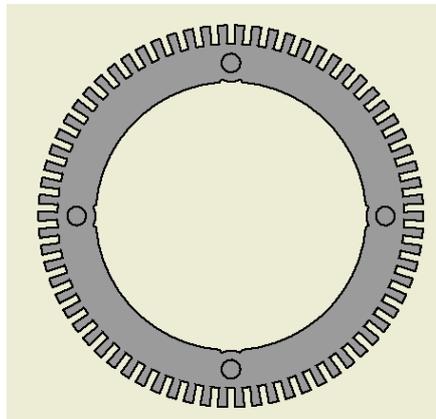


Figura 27 Diseño encoder, inventor, fuente propia

3.3 CONSIDERACIONES DEL USO DE KINECT

Kinect es una tecnología nueva que gracias a las bondades que ofrece al campo de los videojuegos, permite brindar al usuario una relación con un ambiente virtual y para nuevas aplicaciones relacionarse con el medio tecnológico que le rodea como si el Kinect fuera una consola de mando, la librería de conexión al PC fue lanzada en Junio del 2011 mediante la

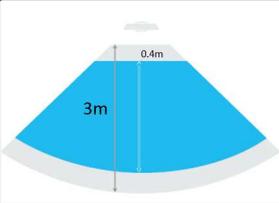
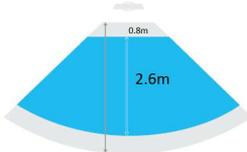
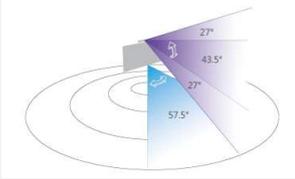
plataforma de Windows 7, pero diversos desarrolladores han readecuado las librerías para poder manejarlo desde linux.

3.3.1 CONSIDERACIONES FÍSICAS

Los datos capturados por el sensor Kinect y procesados por el controlador, son dan una relación entre métodos empleados para análisis de movimientos humanos, razón por la cual se debe considerar la posición de desempeño del sistema y la variación de los resultados obtenidos bajo las características óptimas de funcionamiento.

Tabla 7
Consideraciones óptimas de funcionamiento Kinect

DETALLE

RANGO DE VISIÓN DE PROFUNDIDAD	Limite físico total	0.4m a 3m	
	Zona de trabajo efectiva	0.8m a 2.6m	
PROFUNDIDAD Y CÁMARA RGB	Horizontal	57.5 Grados	
	Vertical	43.5 Grados -27 a +27 grados de posicionamiento del motor del sensor	

Fuente (MacCormick, 2015)

Para poder conectar el Kinect al pc se debe hacerlo mediante un cable USB con fuente de alimentación exterior



Figura 28 Cable de alimentación Kinect, fuente propia

3.3.2 CONSIDERACIONES DE SOFTWARE.

Primero es indispensable instalar SDK de Kinect para Windows el mismo que en la página oficial lo obtenemos según la versión del computador, una vez instalado el SDK cuando conectamos el Kinect se instala automáticamente los controladores de los sensores en el computador

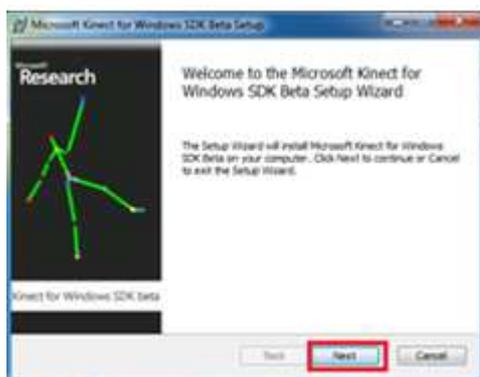


Figura 29 Instalador Librería SDK

La detección de esqueleto a través de la librería Skeleton que es una librería propia para visual c#, significa seguimiento de esqueleto, el algoritmo nos permite identificar en primer lugar los puntos de articulaciones y su posicionamiento en el espacio de trabajo.

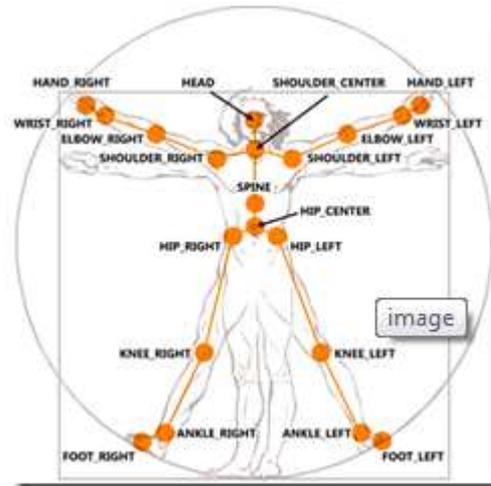


Figura 30 Reconocimiento de puntos de movilidad, fuente (Kleyner, 2013)

Para realizar la aplicación se empleará WPF (Windows Presentation Foundation), el mismo que es propio de Microsoft en visual c#, WPF es un sistema de presentación visual para crear aplicaciones cliente de Windows. La utilidad de WPF es dividir la capa de diseño de la capa de programación, con el que primero realizaremos el diseño a través de comandos en lenguaje XAML, para la programación se debe agregar la librería de Kinect en las referencias para poder acceder a los respectivos comandos de esqueletización.

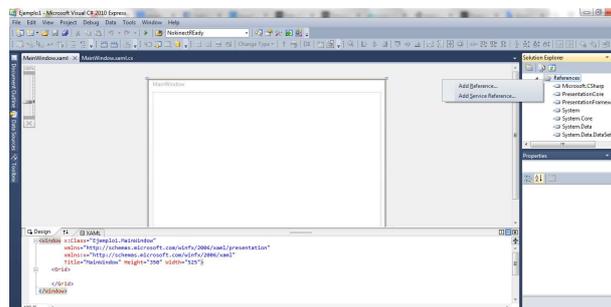


Figura 31. Agregar Librería de Kinect a WPF, fuente propia

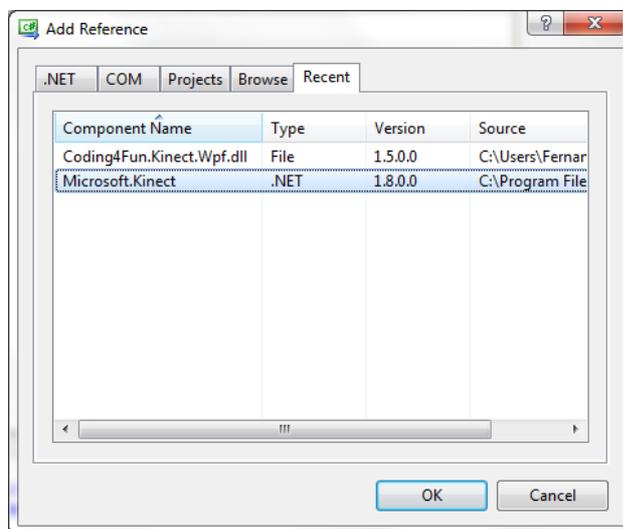


Figura 32 Identificación de la librería instalada a través del SDK

Las aplicaciones en Windows Presentation Foundation "WPF" se categorizan a Microsoft.NET framework 3.5, para el SDK 1.8 que es la última versión. La librería de Kinect lanzada por Windows se recomienda Microsoft.NET framework 4.0.

3.4 SISTEMA DE COMUNICACIÓN KINECT-BRAZO ROBÓTICO A PC

La comunicación inalámbrica no tiene medio físico de propagación, más bien al contrario utiliza ondas electromagnéticas para envío y recepción de datos, por tal motivo tiene un alcance limitado. El Kinect para poder enviar los datos por un medio inalámbrica debe estar conectada a la misma red que el brazo robótico, el envío de datos desde Kinect se lo realiza a través del ambiente de desarrollo integrado propio de Microsoft que para la aplicación se ha escogido Visual c#. Visual C# posee librerías de comunicación utilizando sockets mediante un protocolo TCP, para que la comunicación sea bidireccional tanto en envío de órdenes de control como en monitoreo de datos entregados por el sensor siendo fundamental que en la aplicación se maneje mediante criterio el diseño de Cliente-Servidor. El servidor siempre está en

escucha del puerto especificado, y el cliente debe conocer la Ip donde se quiere entregar los datos.

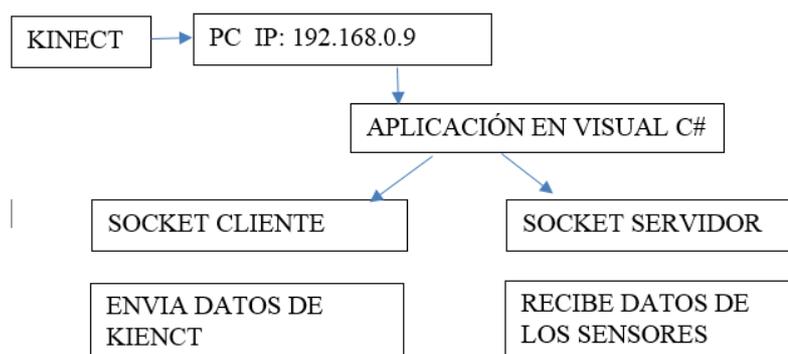


Figura 33 Esquema Envío / Recepción de datos Kinect, fuente propia

Raspberry no posee conexión inalámbrica, pero si posee de un puerto Ethernet, por lo que mediante cable UPT con conectores RJ45, se conecta a un ruteador, en el ruteador se crea la red inalámbrica, y se coloca una Dirección IP a la tarjeta, el ambiente de desarrollo integrado para realizar la aplicación en la raspberry se ha seleccionado Python, y mediante sockets se establece la comunicación a través de la red inalámbrica con la Pc conectada al Kinect.

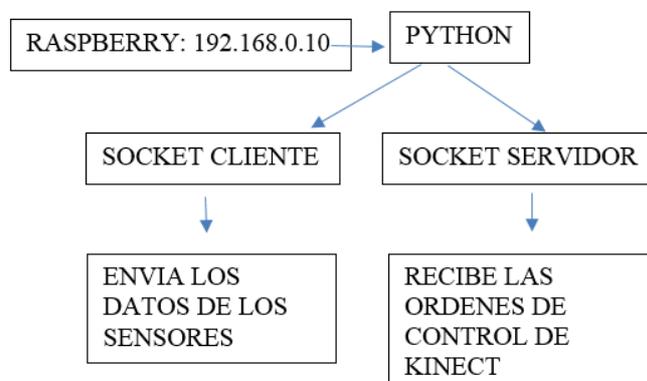


Figura 34 Esquema envío/Recepción de datos Raspberry

La comunicación es inalámbrica, raspberry es encargada de recibir los datos enviados desde Kinect a través del enrutador, todos los equipos deben estar dentro de una misma red la cual se la puede configurar en el ruteador.



Figura 35 Esquema de Red, Dirección Ip de cada equipo

3.5 DISEÑO DEL HARDWARE DEL SISTEMA DE CONTROL

Raspberry es una tarjeta programable de alto nivel pero no posee entradas análogas, razón por la cual se decidió que toda la acción de control recaiga sobre una tarjeta Arduino, la versión Duemilanove posee 7 entradas Análogas y 13 entradas/salidas digitales, conexiones suficientes para monitorear y controlar el Brazo robótico. La función de arduino es recibir los datos de control enviados por el puerto serial a través de Raspberry, y entregar a través del puerto serial los datos leídos de los sensores instalados en el brazo robótico.

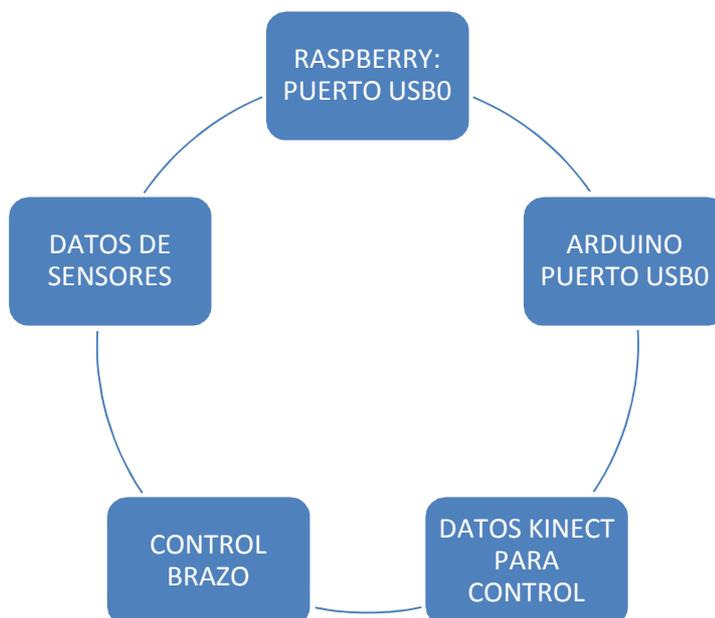


Figura 36 Esquema del ciclo Raspberry-Arduino

Considerando que los actuadores son motores de bajo consumo con caja reductora se debe realizar las acciones de cambio de giro, para control de posición, mediante el integrado L293D se realizará la acción de control de posición mediante cambios de giro, para poder realizar el cambio de giro se debe habilitar el pin de “enable”, y mediante cambios digitales de alto a bajo o viceversa se realiza el cambio de polaridad del motor a través del integrado.

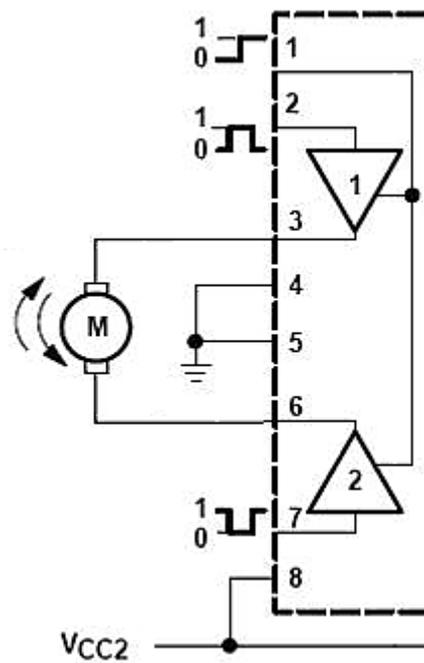


Figura 37 Control de cambio de giro del motor, (Eduardo, 2015)

Tabla 8 Lógica Digital para cambio de Giro, fuente propia

EN	A1	A2	FUNCIÓN
1	0	1	Giro derecha
1	1	0	Giro Izquierda
1	0	0	Alto
1	1	1	Alto

3.6 LEVANTAMIENTO DE PLANOS

3.6.1 DISEÑO ELECTRÓNICO

A través del programa de simulación proteus se puede considerar el uso de pines necesarios para la conexión tanto de los controladores como de los sensores que irán sujetos a través de la placa arduino.

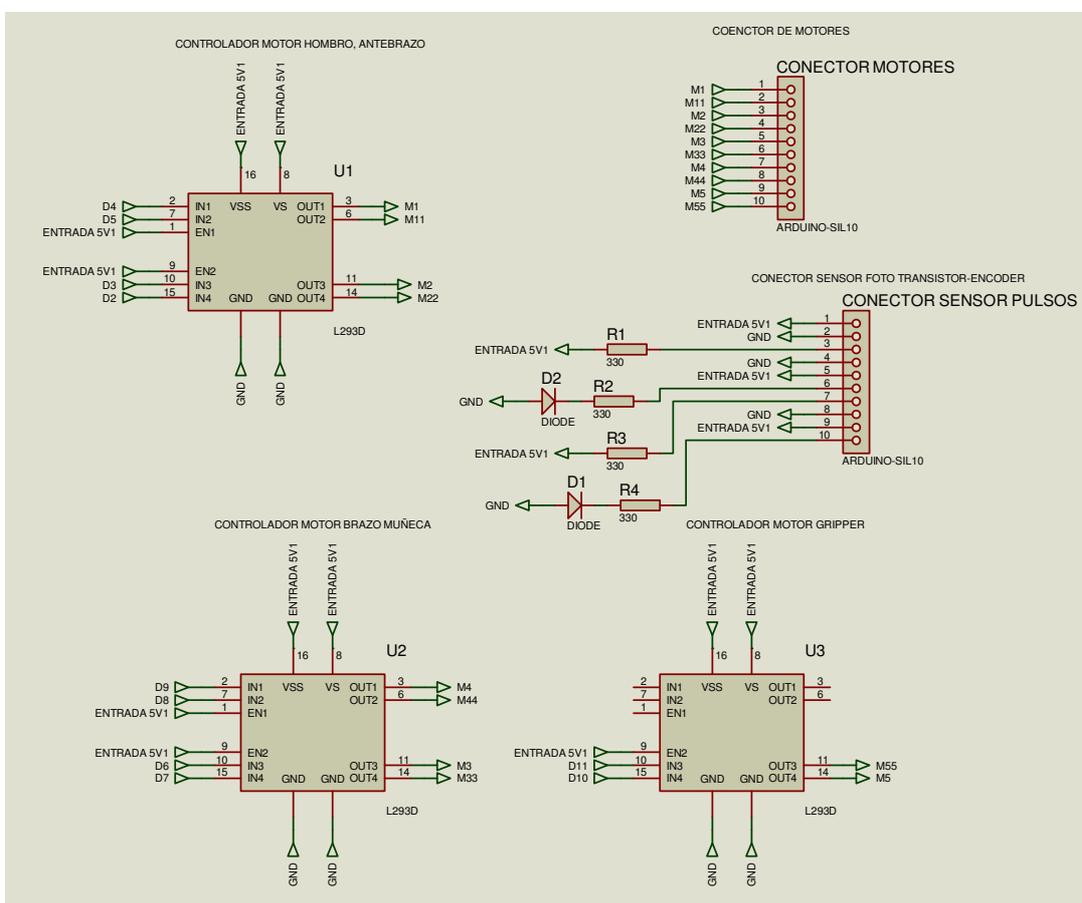


Figura 38 Diseño Electrónico de la conexión entre elementos de la tarjeta

3.6.2 DISEÑO DE TARJETA

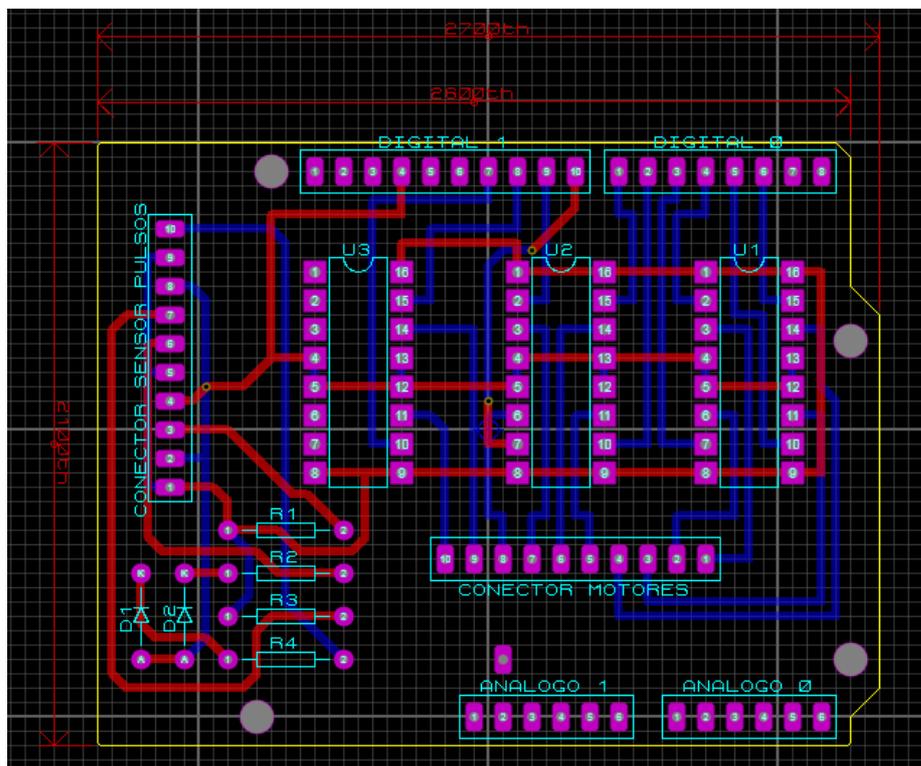


Figura 39 Esquema PCB de la tarjeta Controladora, fuente propia

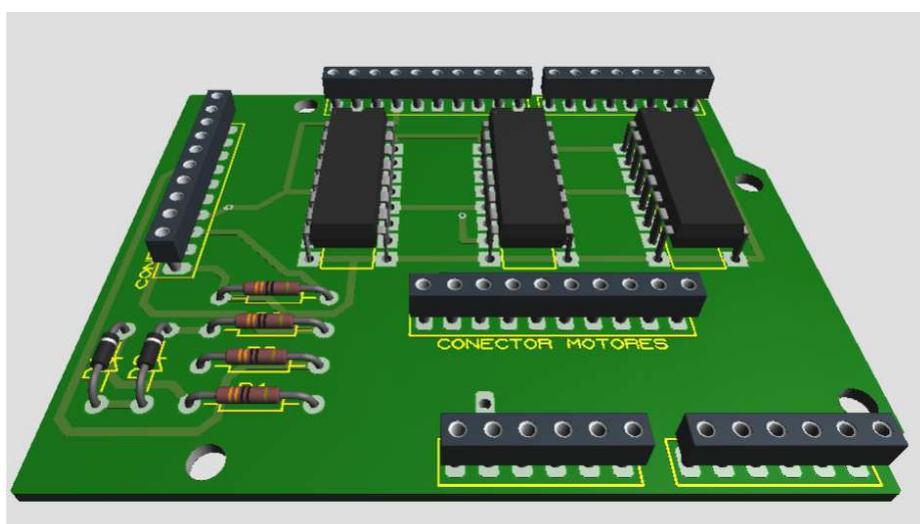


Figura 40 Vista 3D del diseño completo de la tarjeta, fuente propia

3.7 DISEÑO DE LA INTERFAZ A TRAVÉS DE VISUAL C#-SHARP

Con respecto a la normativa industrial para desarrollo de las pantallas se ha considerado la guía Gedis para diseño y distribución de los elementos dentro de la pantalla. El ambiente de desarrollo C# Sharp posee la ventaja seccionar la interfaz de diseño de la interfaz de programación, vinculando ambos campos en una misma aplicación, para el diseño de la interfaz se considera la programación mediante XAML “eXtensible Application Markup Language”, cuya traducción es lengua extensible de formato para aplicaciones, también conocido como lenguaje de marcado da la pauta para que nuestra aplicación pueda adaptarse a un diseño web pues tiene como base una programación similar a HTML.

Windows Presentation Foundation “WPF” es un subsistema de presentación unificado para Windows, se compone de un motor de visualización y un marco de código administrado. Se basa en código administrado pero utiliza lenguaje de marcado, y todas las aplicaciones en lenguaje de marcado XAML son compatibles con C#



Figura 41 Presentación de la Pantalla Inicial de la interfaz



Figura 42 Navegación entre los botones de comando

CAPÍTULO IV

4. IMPLEMENTACIÓN DE HARDWARE Y SOFTWARE

A continuación se presenta la implementación de los diseños establecidos en el capítulo anterior, además el desarrollo de la lógica de programación para control y monitoreo de cada articulación empleando arduino apoyándose en el envío y recepción de datos mediante Wireless a través de Raspberry PI. Se indica la lógica de programación de cada método necesario para el reconocimiento visual del esqueleto del usuario usando el sensor Kinect.

4.1 IMPLEMENTACIÓN DE LA INSTRUMENTACIÓN DEL BRAZO

Gracias al diseño de encoder realizado en inventor dicha pieza se la exporta a un formato .stl, el mismo que nos da la facilidad de realizar el objeto en una impresora 3D.



Figura 43 Encoder para Base implementado en Impresora 3D

El encoder se lo adjunta a la base del brazo Robótico con la finalidad de obtener la posición de giro de la base, cada apertura representa aproximadamente 3 grados de giro, por tanto cada pulso del encoder detectado por el sensor foto-transistor representa 3 grados de rotación.



Figura 44 Encoder Impreso junto a la base de la estructura

Los sensores acelerómetros nos permiten obtener el ángulo de inclinación de cada articulación en la que se encuentre predispuesto.

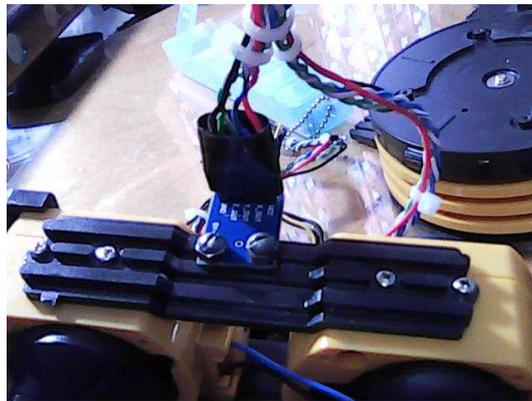


Figura 45 Sensor ubicado en el Brazo, fuente propia

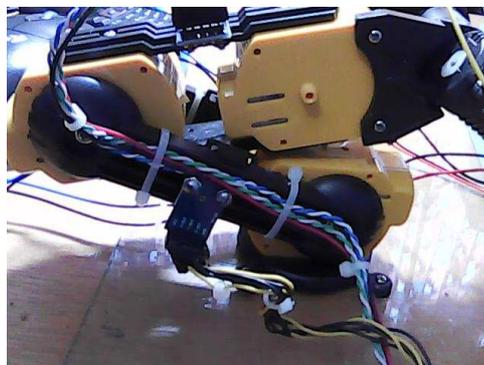


Figura 46 Sensor, ubicado en el antebrazo, fuente propia

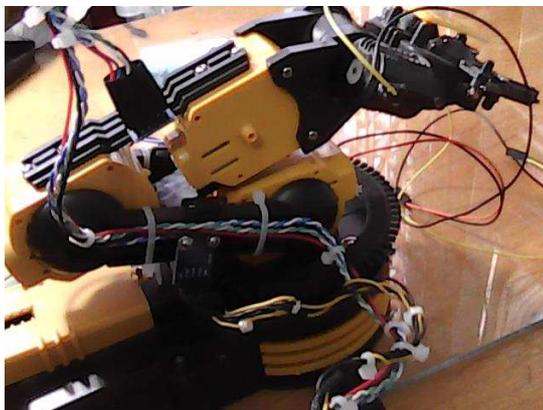


Figura 47 Estructura con Instrumentación incorporada

Para el antebrazo el sensor se encuentra colocado de forma vertical por tal motivo se tiene más variación de valores en el eje x que en los demás eje, consideramos la zona de trabajo según su estructura se ha seleccionado para el antebrazo un rango de valores aceptables desde 70 a 135 grados.

```

COM3
x: 112.46| y: 165.53 | z: 83.91
x: 112.46| y: 170.84 | z: 86.19
x: 113.75| y: 171.38 | z: 86.19
x: 112.46| y: 170.84 | z: 86.19
x: 111.80| y: 166.87 | z: 84.67
x: 112.46| y: 165.53 | z: 83.91
x: 111.54| y: 166.87 | z: 84.74
x: 112.46| y: 167.28 | z: 84.67
x: 111.80| y: 166.87 | z: 84.67
x: 112.46| y: 167.28 | z: 84.67
x: 111.80| y: 166.87 | z: 84.67
x: 112.73| y: 167.28 | z: 84.60
x: 112.73| y: 167.28 | z: 84.60
x: 112.46| y: 167.28 | z: 84.67
x: 111.80| y: 166.87 | z: 84.67
Autoscroll
No line ending
9600 baud
  
```

Figura 48 Lectura Sensor Antebrazo valor mínimo aceptable

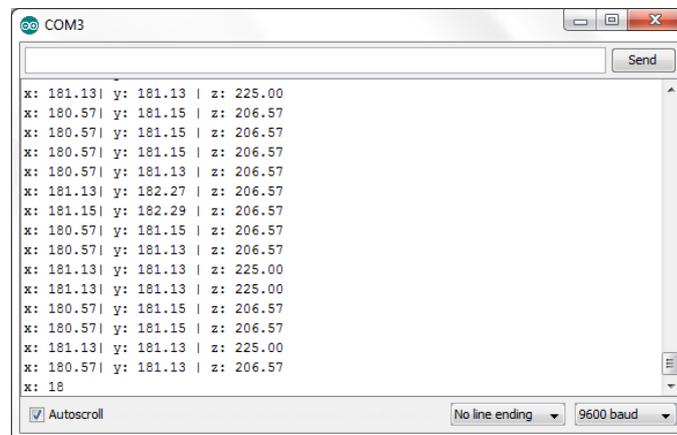


Figura 49 Lectura Sensor Antebrazo valor máximo aceptable

4.2 IMPLEMENTACIÓN DEL HARDWARE DEL SISTEMA DE CONTROL

Se procede a la elaboración de la placa, soldadura de elementos, y ubicación en brazo robótico, para su posterior conexión con los dispositivos de monitoreo y control.

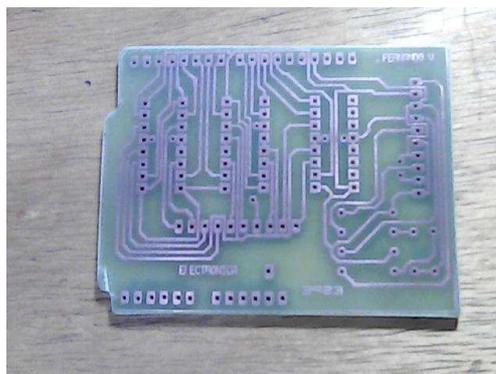


Figura 50 Placa revelada según diseño, fuente propia

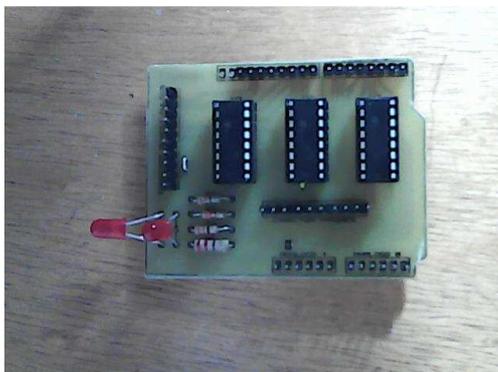


Figura 51 Placa con elementos soldados, fuente propia

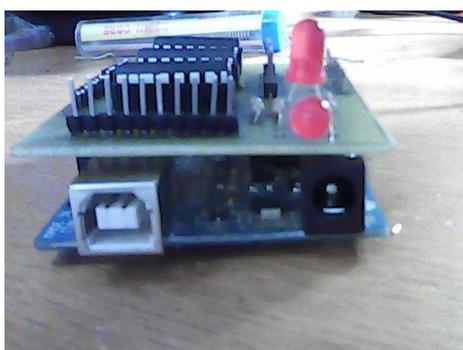


Figura 52 Placa sobrepuesta sobre arduino Duemilanove, fuente propia

4.3 IMPLEMENTACIÓN DEL SOFTWARE DEL SISTEMA DE CONTROL

Arduino es el encargado de recibir los datos de setPoint para realizar las respectivas acciones de control y enviar los valores de los sensores a través de la comunicación serial con Raspberry, cada articulación posee su propia acción de control en relación a la evaluación de datos y la ubicación determinada por el kinect.

4.3.1 CONTROL BRAZO

Primero se debe definir la función de lectura de las entradas análogas del sensor ADXL335, este tipo de sensor nos entrega una señal por cada eje; se

debe hacer la respectiva conversión de unidades crudas a unidades lógicas, mediante ecuaciones para poder obtener el rango de valores de ángulo que se desea medir, para la articulación del antebrazo se ha seleccionado un rango de valores desde 60 grados hasta 120 grados sobre la horizontal.

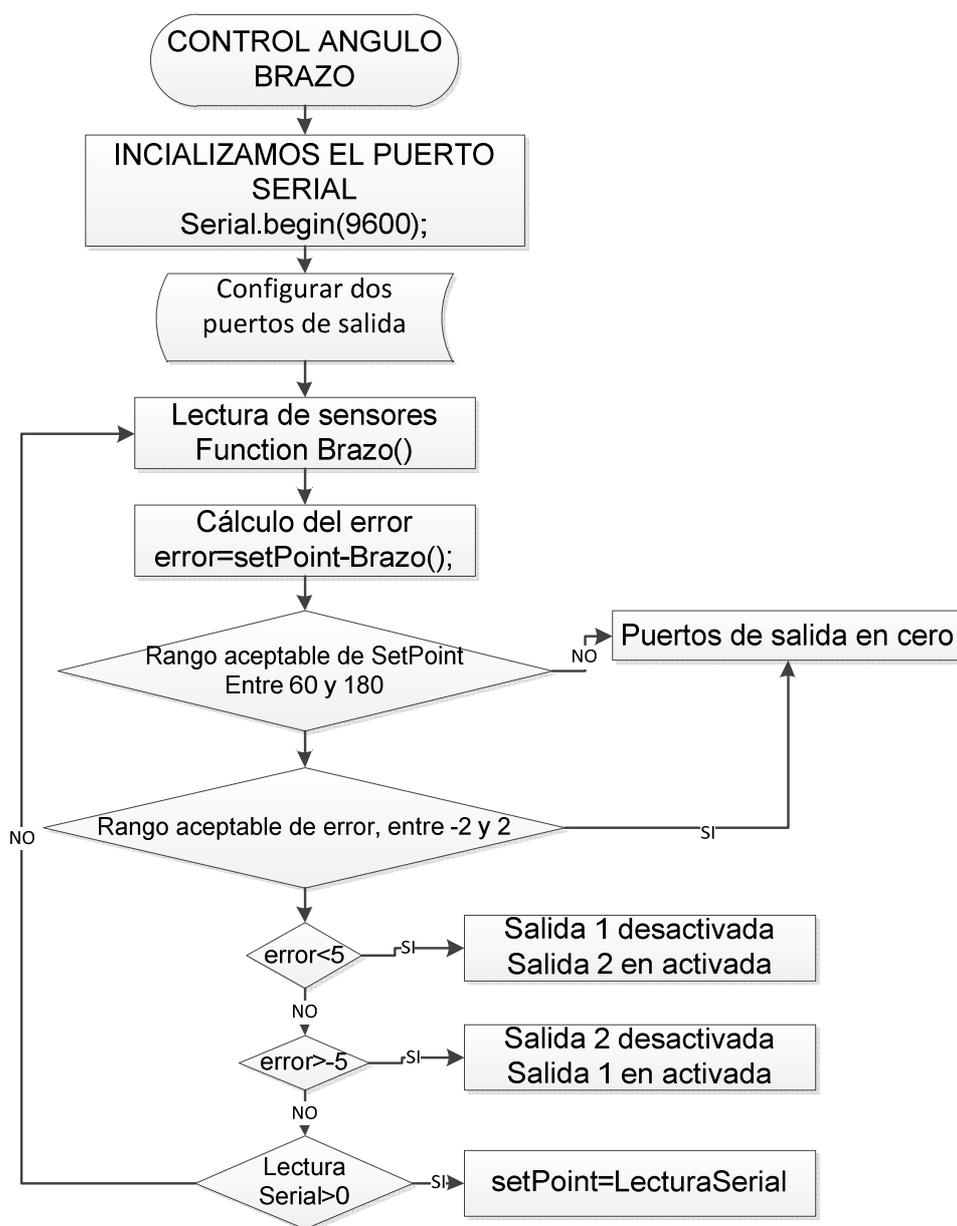


Figura 53 Flujoograma de Control Brazo, fuente propia

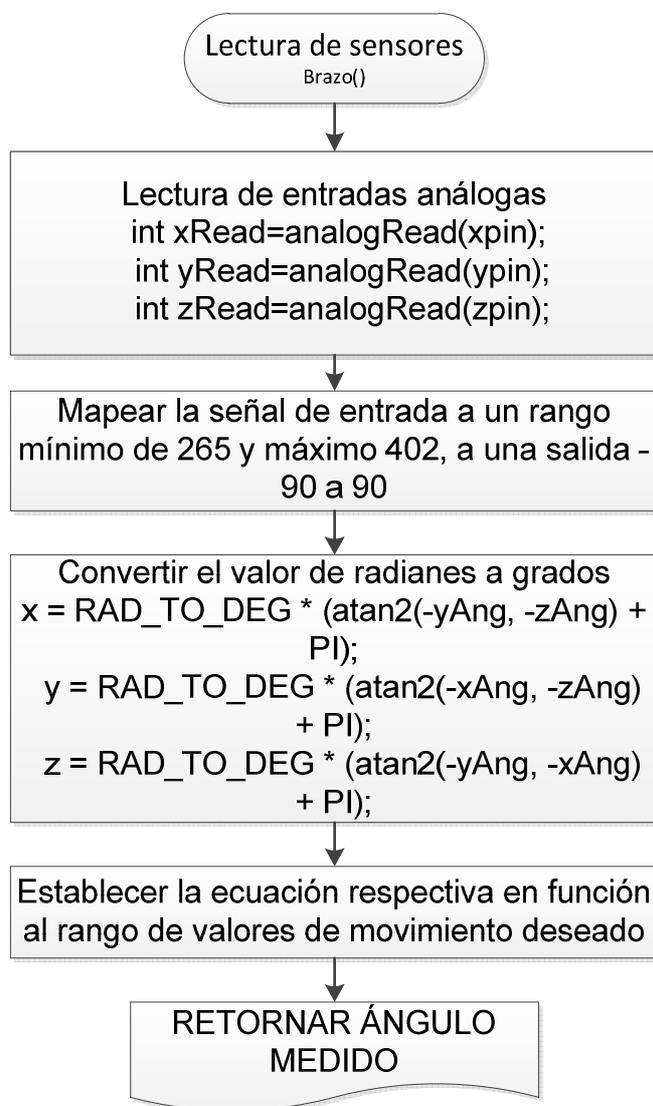


Figura 54 Flujograma de lectura de sensores, fuente propia

El rango de valores entregados por el sensor es necesario pre escalar para obtener datos medibles conforme a nuestro sistema y más no conforme al sensor

Tabla 9
Valores medidos Vs valores Deseados sensor Brazo

Valores Sensor	Valores Deseados
360	180
260	90
241	60

Para obtener el rango de valores deseados se forma una ecuación considerando los valores de la tabla 9

$$m = \frac{60 - 180}{241 - 360}$$

$$m = 1.008$$

$$(x - 360) * 1.008 = 180 - y$$

$$y = 180 - 1.008 * (360 - x)$$

4.3.2 CONTROL ANTE BRAZO

Basados en el mismo dispositivo del control del antebrazo, se ha seleccionado un rango de 60 a 180 grados.

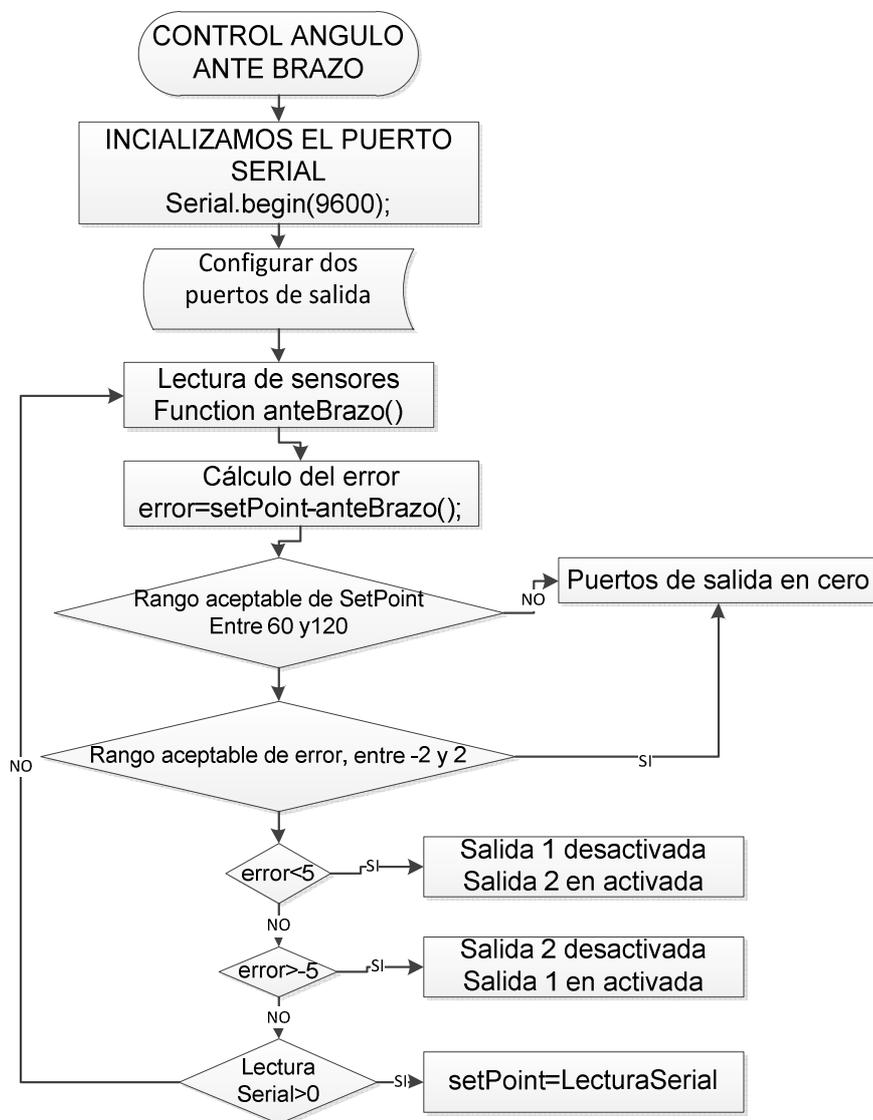


Figura 55 Flujograma Control Ante Brazo

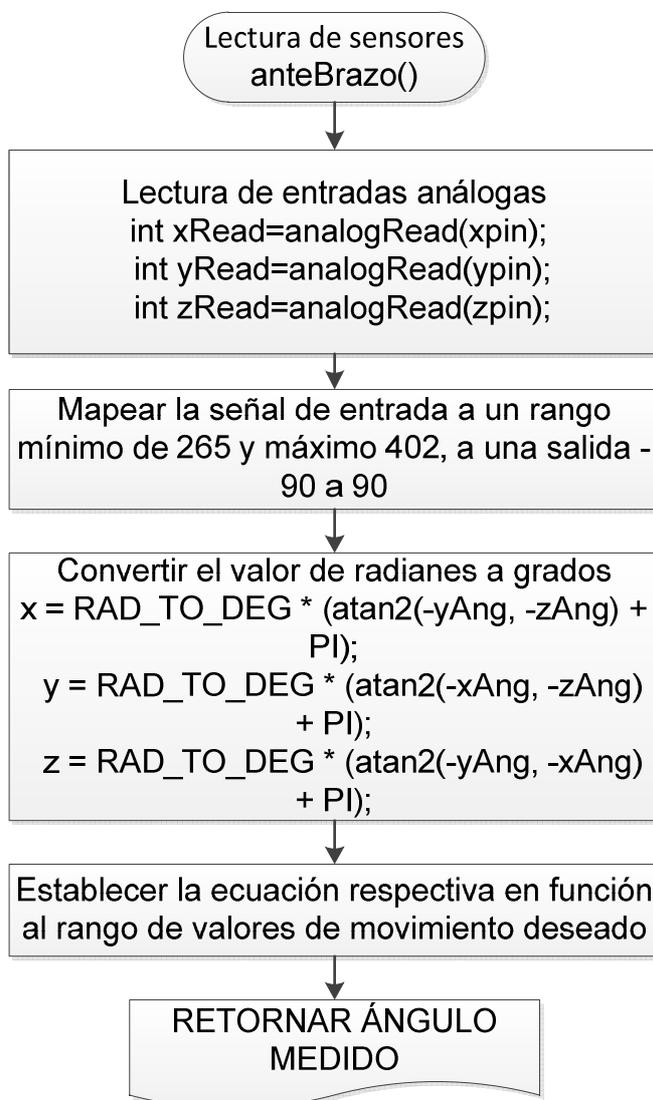


Figura 56 Flujograma Lectura de Sensores

El rango de valores entregados por el sensor varía en función a su posición como se desea obtener un rango de valores conforme a nuestras necesidades del sistema de control se va a pre escalar en función a una ecuación lineal.

4.3.3 CONTROL BASE

Primero se debe definir la función de lectura de pulsos del encoder, a través de entradas digitales; se debe hacer la respectiva conversión de pulsos para poder obtener el rango de valores de ángulo que se desea medir.

La articulación de la base se ha seleccionado un rango de valores desde 30 grados hasta 120 grados sobre la vertical.

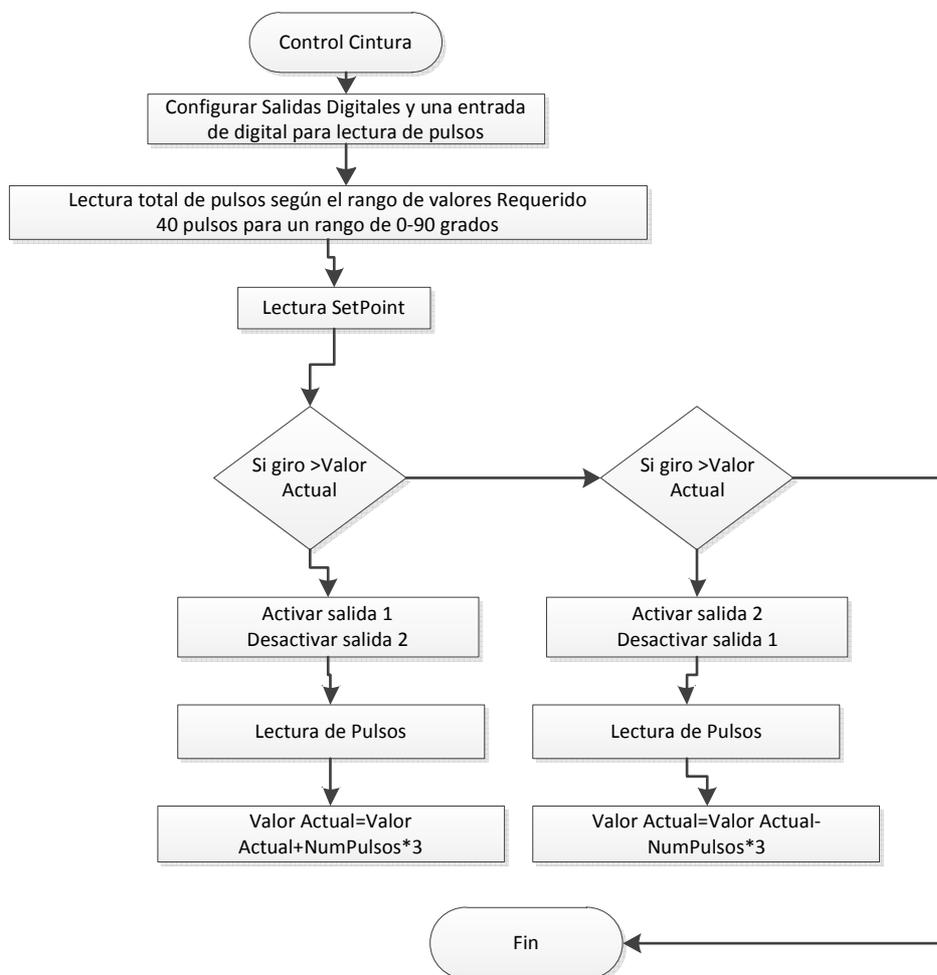


Figura 57 Flujograma, control Base, fuente propia

4.3.4 CONTROL MUÑECA

Basados en el mismo dispositivo del control de la base, se ha seleccionado un rango de 60 a 120 grados.

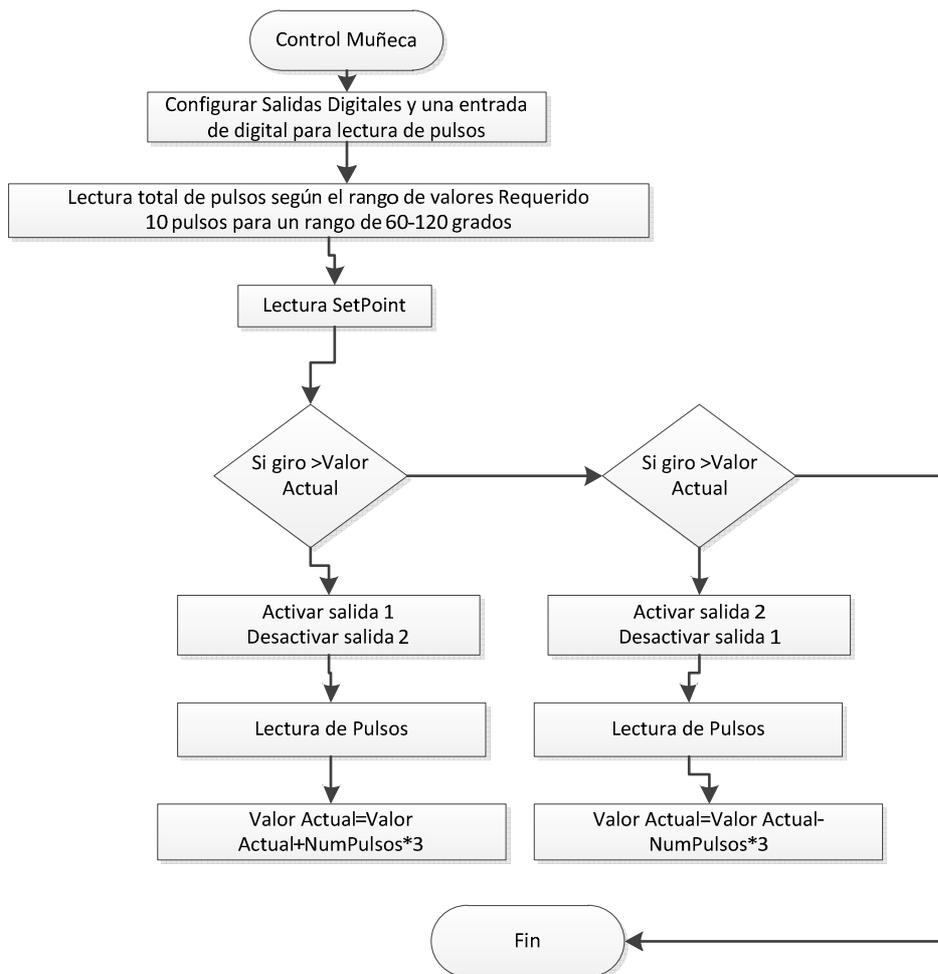


Figura 58 Flujograma Control muñeca, fuente propia

4.3.5 CONTROL GRIPPER

El sistema de control es un lazo abierto On/Off, controlado por tiempo a partir de señales del Kinect y banderas internas del sistema arduino.

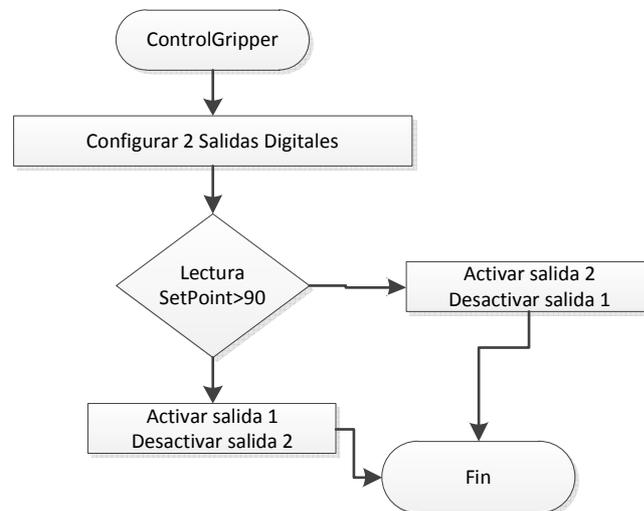


Figura 59 Flujoograma Control gripper, fuente propia

4.4 IMPLEMENTACIÓN DE LA COMUNICACIÓN INALÁMBRICA

Lo primero es configurar los equipos y enlazarlos en la misma red, para ello se configura el ruteador, creando una red propia para centralizar toda la actividad en una misma red, para configurar el ruteador se lo debe conectar con el computador y en el navegador se ingresa a la red que indica el dispositivo para nuestro caso el equipo se configura en el navegador a través de la red 192.168.0.1.



Figura 60 Paso 1 para configurar Ruteador, fuente propia

Siguiendo los pasos de configuración se crea una red, en la que no se unirán los equipos.



Figura 61 Paso 2 para configurar el ruteador, fuente propia

4.3.1 KINECT-PC

El Kinect se conecta directamente mediante al PC mediante cable USB, pero el PC para enviar los datos requiere configurar tanto su hardware Wireless, como su software programado en Visual c#

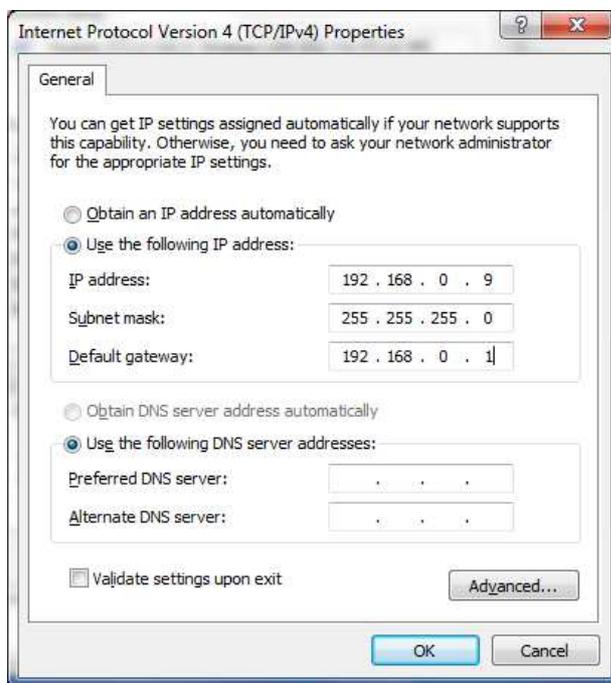


Figura 62 Configuración de red PC para Kinect, fuente propia

Para enviar los datos identificados por el Kinect la aplicación debe programar un método como cliente, puesto que el servidor siempre está escuchando el puerto.

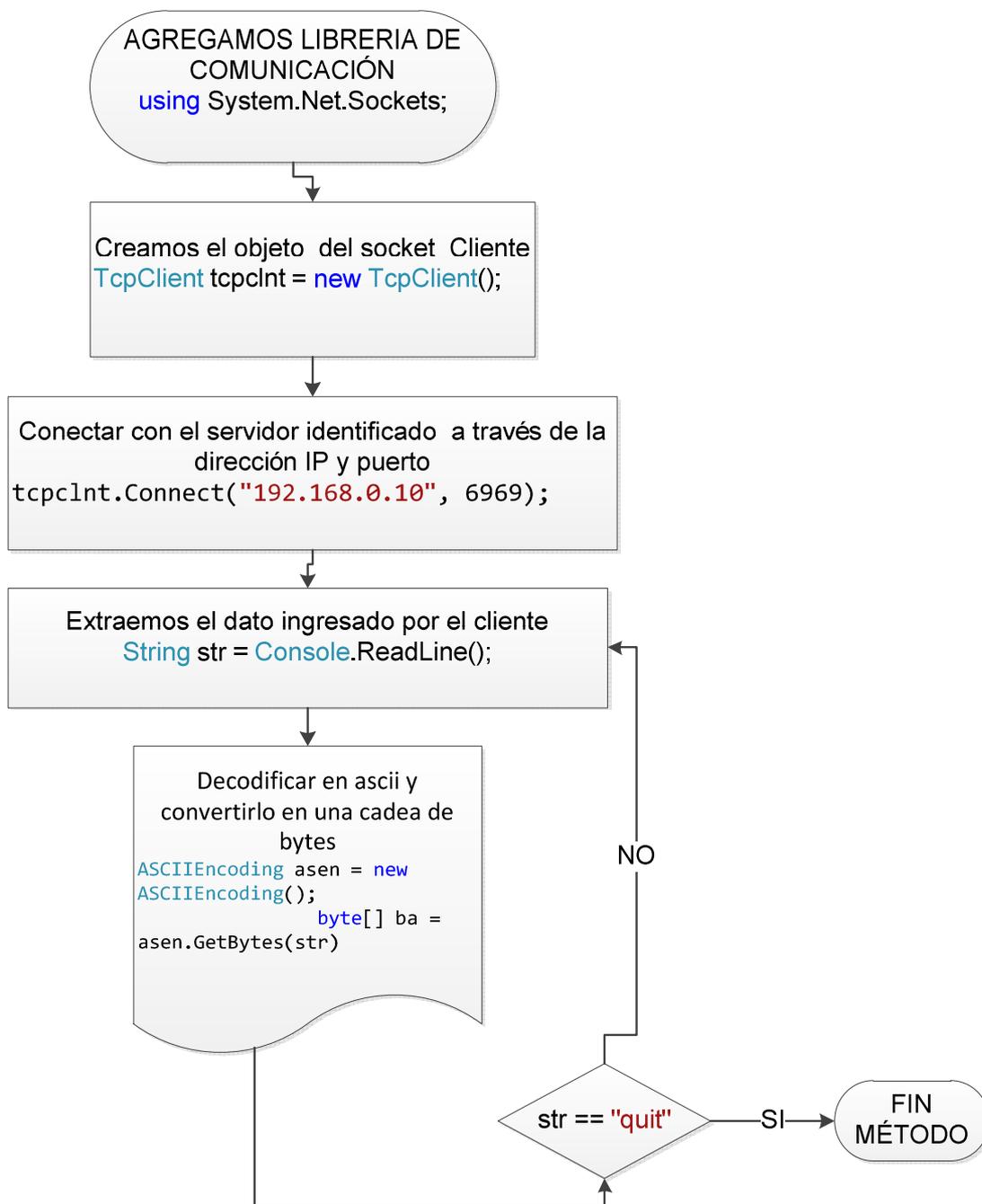


Figura 63 Flujograma Cliente visual c#, fuente propia

Para recibir los datos de los sensores se debe programar una función como servidor para que la aplicación siempre este en escucha de los datos recibidos.

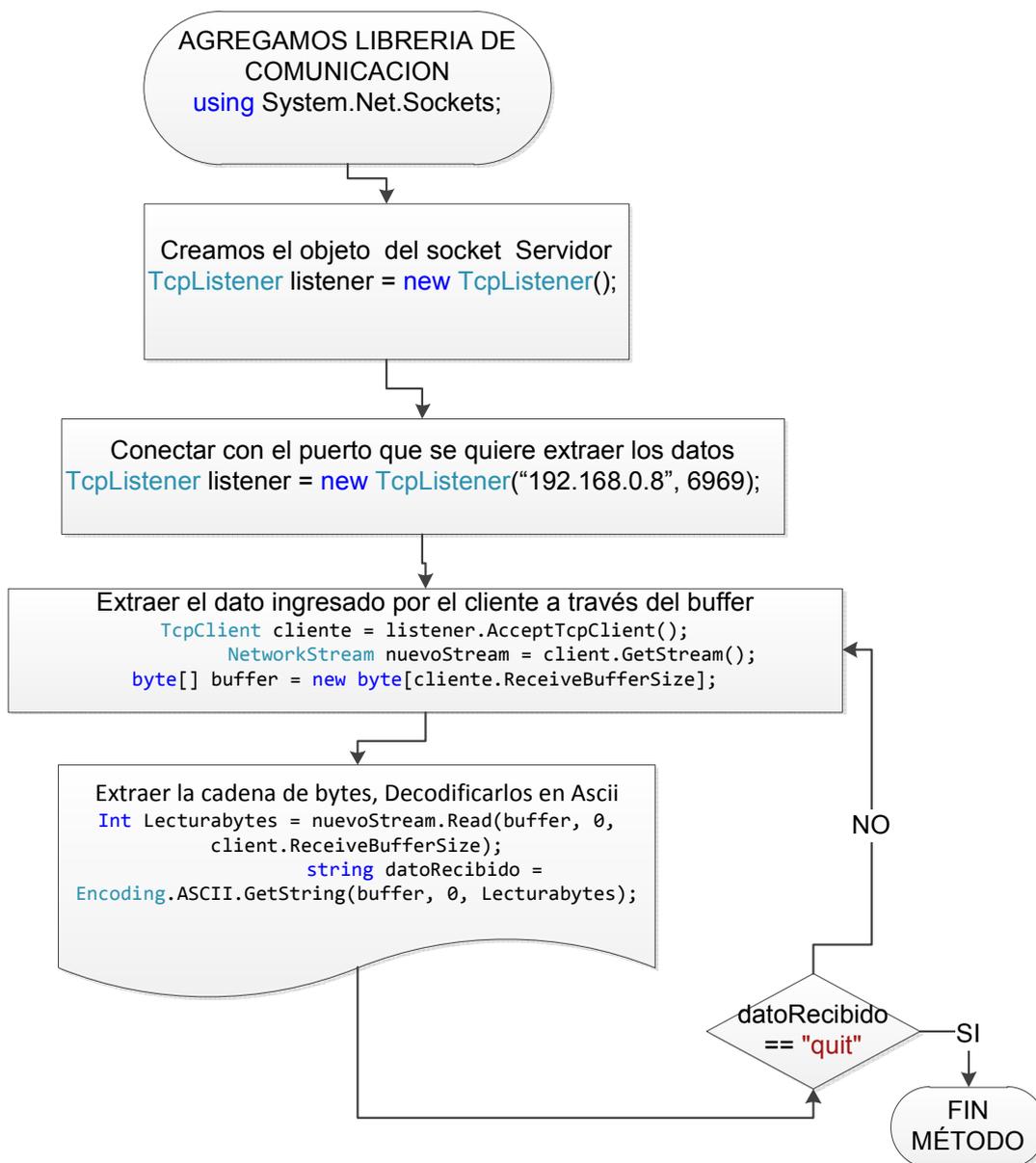
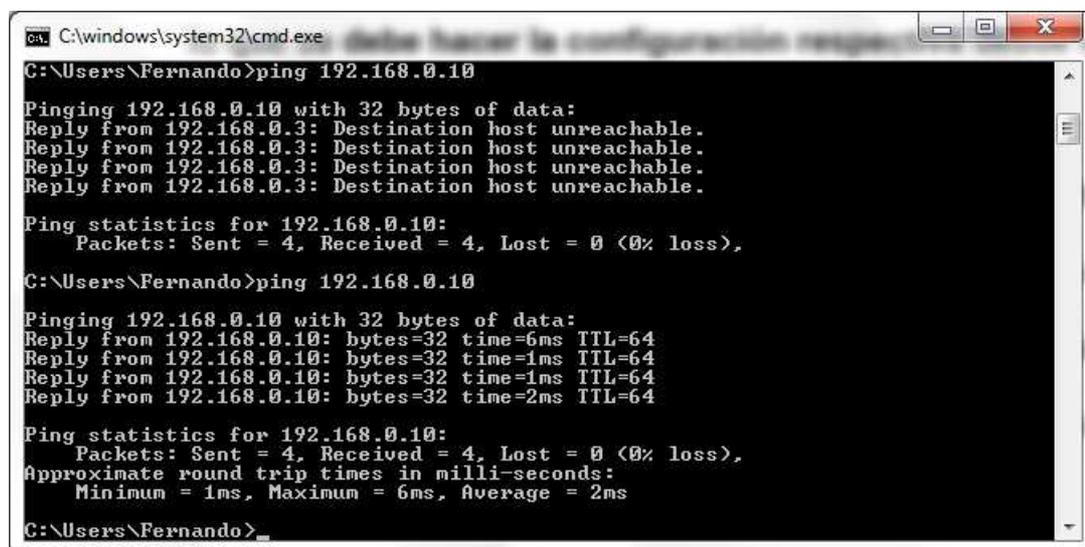


Figura 64 Flujograma servidor, fuente propia

4.3.2 PC-RASPBERRY

Raspberry posee un puerto Ethernet cuya velocidad es de hasta 100Mbps, tiene un sistema Auto-sense que permite seleccionar el tipo de configuración sea por cable directo o cable cruzado, para poder acceder a los datos de la red raspberry debe estar enlazado en la misma red mediante la IP de red, raspberry tiene su base en Linux por lo que se debe hacer la configuración respectiva desde línea de comandos

Para acceder a la configuración de red se ingresa la instrucción: `sudo nano /etc/network/interfaces`, en el archivo que se despliegue se cambia la red a static y se asigna la dirección IP correspondiente a Raspberry 192.168.0.10. Para asegurar que la configuración fue exitosa desde el PC se digita `ping 192.168.0.10`



```

C:\windows\system32\cmd.exe
C:\Users\Fernando>ping 192.168.0.10

Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.3: Destination host unreachable.

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users\Fernando>ping 192.168.0.10

Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.10: bytes=32 time=6ms TTL=64
Reply from 192.168.0.10: bytes=32 time=1ms TTL=64
Reply from 192.168.0.10: bytes=32 time=1ms TTL=64
Reply from 192.168.0.10: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 6ms, Average = 2ms

C:\Users\Fernando>

```

Figura 65 Comunicación establecida, Pc-Raspberry vía Wireless

Para acceder a los datos de la red el ambiente de desarrollo para programar es Python, en el cual para recibir los datos enviados por el cliente se debe realizar un programa que permita leer los valores como Servidor:

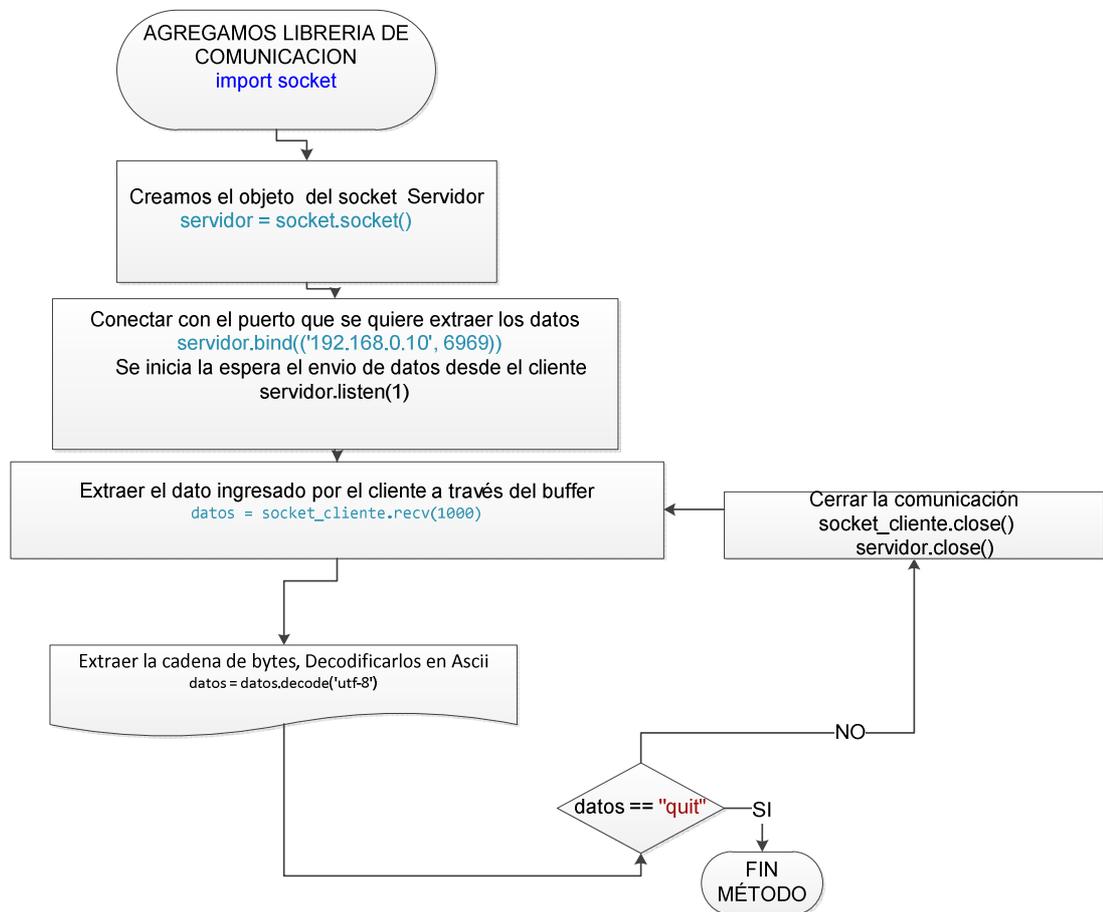


Figura 66 Flujograma Servidor, python, fuente propia



Figura 67 Implementación de la red Raspberry a medio Wireless

Y para enviar los datos recibidos por el sensor se debe enviar los datos como cliente hacia el Servidor que se encuentra en la PC.

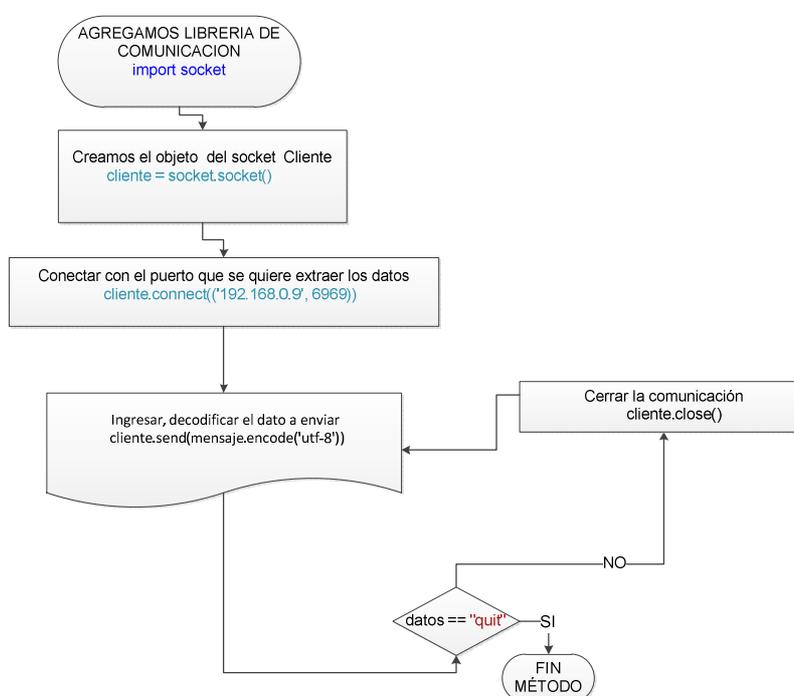


Figura 68 Flujograma cliente, python, fuente propia

4.5 IMPLEMENTACIÓN KIT E DESARROLLO DE APLICACIONES SDK

Una vez instalada la librería de Kinect en Windows, se puede utilizar sus funciones, se ha seleccionado visual C#, la librería más actual disponible desde Microsoft es sdk 1.8.

Primero seleccionar la librería instalada y reconocida por Microsoft Visual C#, y agregamos la referencia de Kinect-Microsoft. Así se puede tener acceso a las funciones y datos presentados por Kinect.

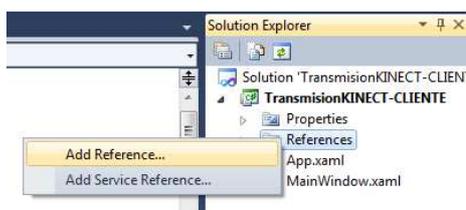


Figura 69 Agregar Librería Kinect, fuente propia

Las librerías propias de Kinect a utilizar son:

Tabla 10

Librería Microsoft.Kinect; funciones que se puede acceder

LIBRERÍA	UTILIDAD
<code>using Microsoft.Kinect;</code>	Funciones de audio
	Funciones de video
	Sensores de profundidad
	Detectores de movimiento

Tabla 11
 Librerías necesarias para la implementación

LIBRERÍA	UTILIDAD
<code>using System.Windows;</code>	Permite ingresar a las funciones propias de Windows Presentation Foundation
<code>using System.Windows.Media ;</code>	Habilita funciones de contenido Media "videos, audio, gráficos"
<code>using System.Net;</code>	Función que permite acceder a direcciones IP
<code>using System.Text;</code>	Contiene clases que permiten representar un dato tipo String en ASCII
<code>using System.Net.Sockets;</code>	Contiene clases que permiten relacionar la interface del programador con la red, a través de las clases en capsuladas de Cliente, Servidor , Oyente
<code>using System;</code>	Representa clases, tipos de arreglos, parámetros
<code>using System.Collections.Generic;</code>	Contiene Interfaces y clases que definen una conexión genérica, que permite al usuario crear tablas, de acceso dinámico
<code>using System.Windows.Controls;</code>	Crea elementos que permiten al usuario interactuar con la aplicación
<code>using System.Windows.Documents;</code>	Permite acceder a la programación del XML
<code>using System.Windows.Input ;</code>	Da soporte a la aplicación, incluyendo datos de entrada como mouse, teclado
<code>using System.Windows.Media.Imaging;</code>	Codifica o decodifica el mapa de bits ingresado por una imagen
<code>using System.Windows.Shapes;</code>	Provee acceso a las librerías que pueden ser usadas por el lenguaje de marcado XAML

Visual C# es un plataforma de programación propia de Microsoft pero con lenguaje de programación de alto nivel lo que nos permite tener una programación estructurada en base de funciones, con el objetivo de reutilizar código ya programado en aplicaciones extensas.

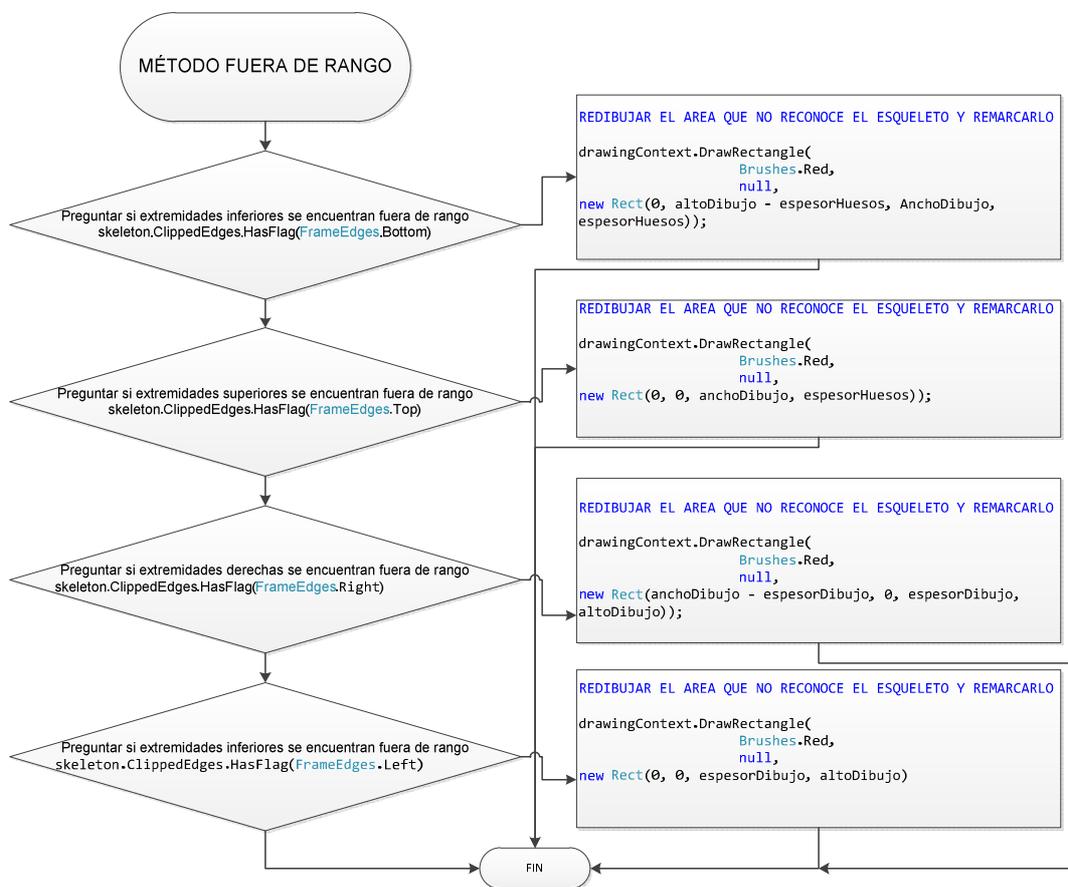
A continuación se describe la lógica de programación para cada función necesaria en la aplicación.

4.5.1 MÉTODO FUERA DE RANGO

Tabla 12

Funciones para desarrollar el método fuera de rango

FUNCIÓN	DETALLE
FrameEdges	Clase que identifica si alguna parte del cuerpo del usuario se encuentra fuera de rango
Brushes.Red	Clase que permite marcar el objeto con color Rojo



Kinect a través de la clase skeleton propia de visual c# identifica el centro del cuerpo del usuario y lo representa con un círculo azul, que servirá de referencia para la gráfica de las demás articulaciones.

4.5.2 EVENTO RECONOCIMIENTO DE ESQUELETO

Tabla 13
Funciones para desarrollar el método Reconocer esqueleto

FUNCIÓN	DETALLE
<code>Skeleton</code>	Clase que entrega todos los datos del reconocimiento del usuario
<code>SkeletonFrame</code>	Clase que entrega la estructura reconocido por el sensor posición, sector, articulaciones
<code>DrawingContext</code>	Describe el contenido visual mediante gráficos del sensor
<code>SkeletonTrackingState</code>	Entrega los posibles estados del skeleton

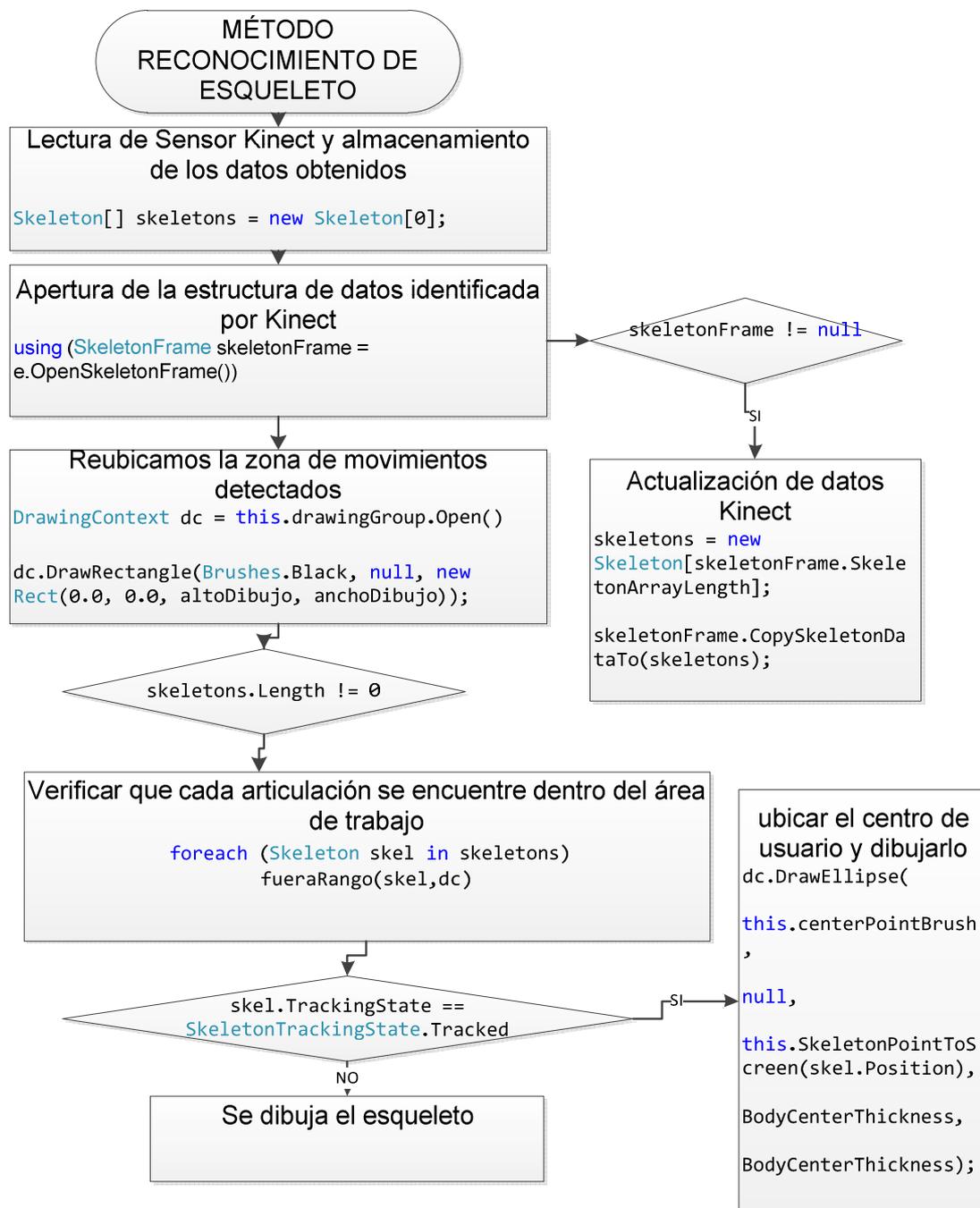


Figura 70 Método para emplear la esquetización, fuente propia

4.5.3 FUNCIÓN DIBUJAR Y TRANSMITIR DATOS

Tabla 14
Funciones para desarrollar el método

FUNCIÓN	DETALLE
<code>Skeleton</code>	Clase que entrega todos los datos del reconocimiento del usuario
<code>JointType</code>	Clase que entrega la posición de cada articulación
<code>DrawingContext</code>	Describe el contenido visual mediante gráficos del sensor
<code>Math</code>	Clase con funciones matemáticas
<code>Convert</code>	Concatenar datos
<code>asen.GetBytes</code>	Transformar en Bytes
<code>Thread</code>	Clase para Obtener y controlar un estado
<code>ASCIIEncoding</code>	Codificación ASCII
<code>JointTrackingState</code>	Clase para seguir y ubicar las articulaciones del skeleton

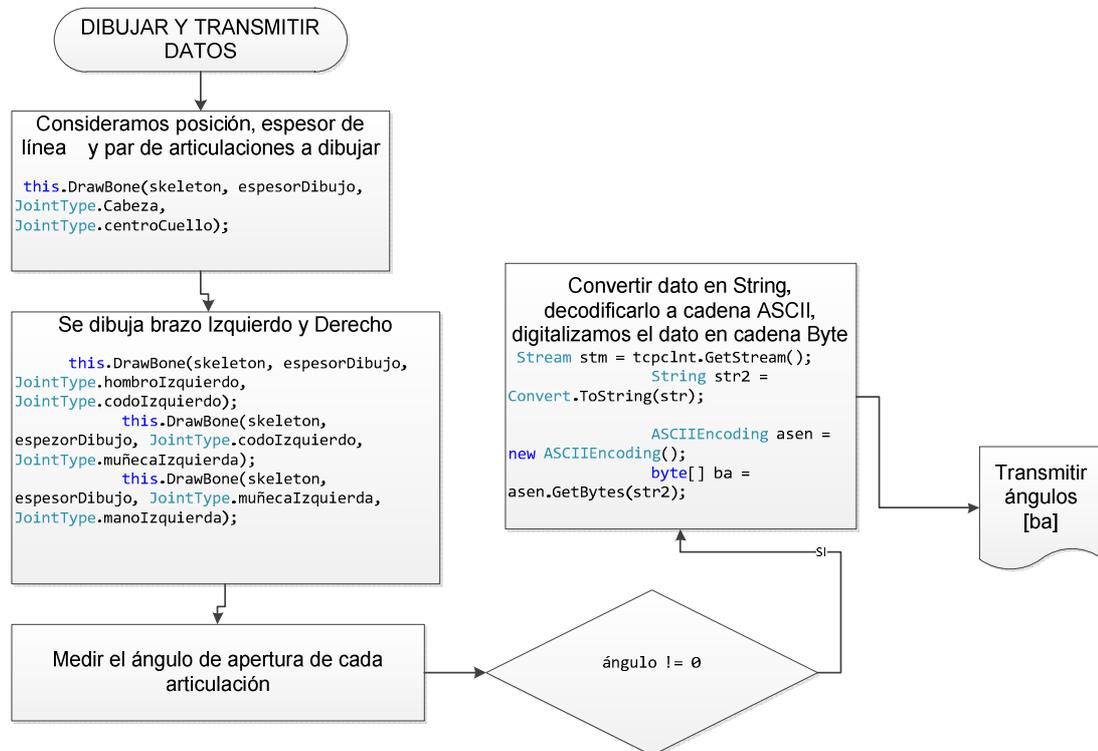


Figura 71 Método para dibujar las partes del esqueleto y enviar datos

4.5.4 MEDIR ÁNGULO EN ARTICULACIONES CORPORALES

Para medir la apertura entre dos huesos identificados por el sensor Kinect se debe considerar el posicionamiento vectorial en los ejes x,y,z, de cada unión reconocida por el sensor Kinect.

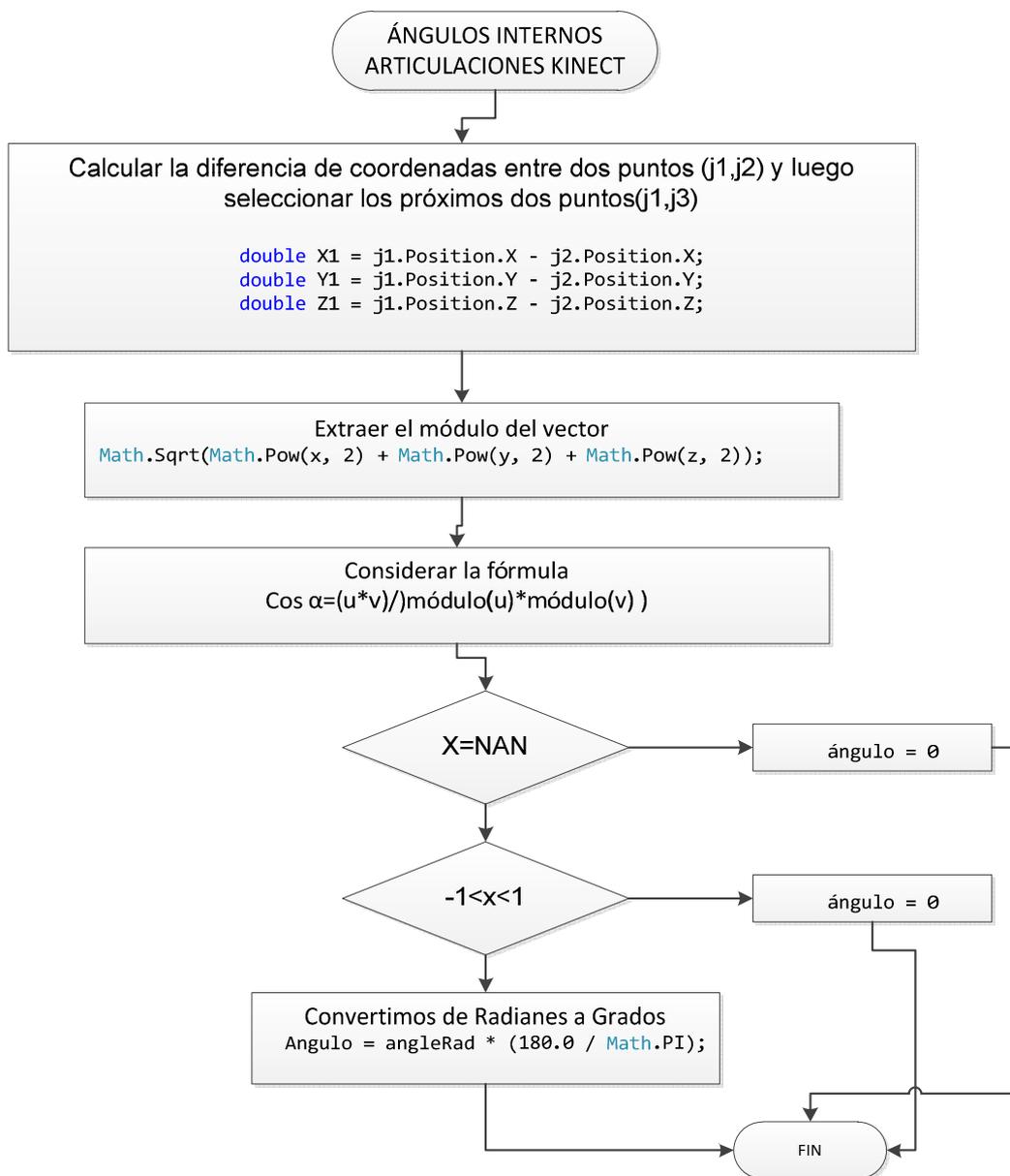


Figura 72 Método para medir los ángulos internos de las articulaciones

4.6 IMPLEMENTACIÓN DE LA INTERFAZ DE USUARIO

4.6.1 CARÁTULA



Figura 73 Presentación Carátula de la aplicación

4.6.2 PÁGINA PRINCIPAL

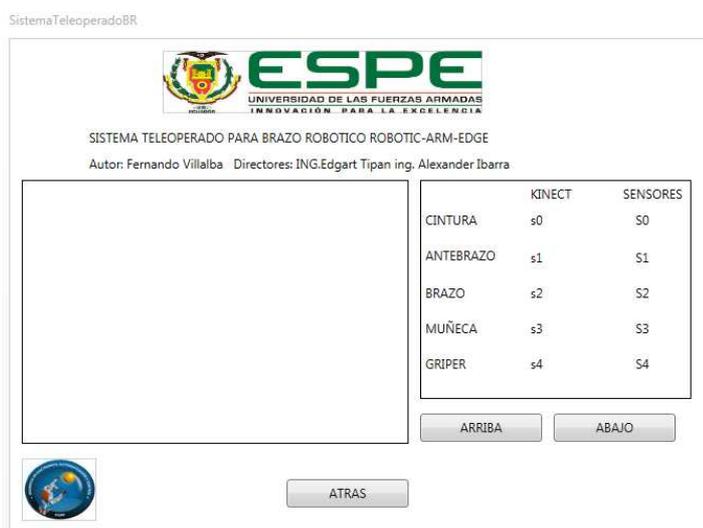


Figura 74 Presentación Interfaz HMI, fuente propia

CAPÍTULO V

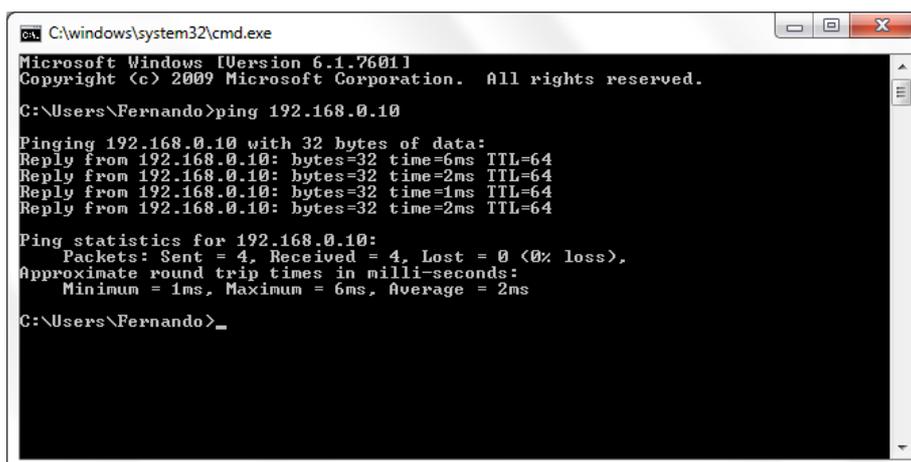
5. PRUEBAS Y RESULTADOS

5.1 PRUEBAS

5.1.1 ALCANCE DE TRANSMISIÓN Y RECEPCIÓN DE DATOS VÍA WIRELESS

Para comprobar la comunicación inalámbrica entre los dos equipos mediante la ventana de comandos de Windows se utiliza el comando PING hacia la dirección destino con la que deseamos comunicarnos, para la aplicación la dirección destino es la 192.168.0.10 perteneciente a raspberry, y la del PC es la 192.168.0.9

Probamos por diferentes distancias tanto en línea directa punto a punto, así como con distancias a través de obstáculos.



```
C:\windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Fernando>ping 192.168.0.10

Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.10: bytes=32 time=6ms TTL=64
Reply from 192.168.0.10: bytes=32 time=2ms TTL=64
Reply from 192.168.0.10: bytes=32 time=1ms TTL=64
Reply from 192.168.0.10: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 6ms, Average = 2ms

C:\Users\Fernando>_
```

Figura 75 Ping distancia directa 1[m], fuente Propia

```

C:\windows\system32\cmd.exe
Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.10: bytes=32 time=9ms TTL=64
Reply from 192.168.0.10: bytes=32 time=2ms TTL=64
Reply from 192.168.0.10: bytes=32 time=1ms TTL=64
Reply from 192.168.0.10: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 9ms, Average = 3ms

C:\Users\Fernando>ping 192.168.0.10

Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.10: bytes=32 time=4ms TTL=64
Reply from 192.168.0.10: bytes=32 time=3ms TTL=64
Reply from 192.168.0.10: bytes=32 time=4ms TTL=64
Reply from 192.168.0.10: bytes=32 time=4ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 4ms, Average = 3ms

C:\Users\Fernando>

```

Figura 76 Ping distancia con obstáculo 1[m], fuente propia.

```

C:\windows\system32\cmd.exe
Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.10: bytes=32 time=7ms TTL=64
Reply from 192.168.0.10: bytes=32 time=3ms TTL=64
Reply from 192.168.0.10: bytes=32 time=6ms TTL=64
Reply from 192.168.0.10: bytes=32 time=3ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 7ms, Average = 4ms

C:\Users\Fernando>ping 192.168.0.10

Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.10: bytes=32 time=81ms TTL=64
Reply from 192.168.0.10: bytes=32 time=4ms TTL=64
Request timed out.
Reply from 192.168.0.10: bytes=32 time=3ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 81ms, Average = 29ms

C:\Users\Fernando>

```

Figura 77 Ping distancia con obstáculo 3[m], fuente propia

```

C:\windows\system32\cmd.exe
Pinging 192.168.0.10 with 32 bytes of data:
General failure.
General failure.
Reply from 192.168.0.10: bytes=32 time=16ms TTL=64
Reply from 192.168.0.10: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 16ms, Average = 9ms

C:\Users\Fernando>ping 192.168.0.10

Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.10: bytes=32 time=3ms TTL=64
Reply from 192.168.0.10: bytes=32 time=6ms TTL=64
Reply from 192.168.0.10: bytes=32 time=3ms TTL=64
Reply from 192.168.0.10: bytes=32 time=3ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 6ms, Average = 3ms

C:\Users\Fernando>

```

Figura 78 Ping distancia directa 5[m], fuente propia

```

C:\windows\system32\cmd.exe
Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.10: bytes=32 time=81ms TTL=64
Reply from 192.168.0.10: bytes=32 time=4ms TTL=64
Request timed out.
Reply from 192.168.0.10: bytes=32 time=3ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 81ms, Average = 29ms

C:\Users\Fernando>ping 192.168.0.10

Pinging 192.168.0.10 with 32 bytes of data:
General failure.
General failure.
Reply from 192.168.0.10: bytes=32 time=16ms TTL=64
Reply from 192.168.0.10: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 16ms, Average = 9ms

C:\Users\Fernando>

```

Figura 79 Ping distancia con Obstáculo 5[m], fuente propia

```

C:\windows\system32\cmd.exe
Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.10: bytes=32 time=4ms TTL=64
Request timed out.
Reply from 192.168.0.10: bytes=32 time=36ms TTL=64
Reply from 192.168.0.10: bytes=32 time=19ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 36ms, Average = 19ms

C:\Users\Fernando>ping 192.168.0.10

Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.10: bytes=32 time=18ms TTL=64
Reply from 192.168.0.10: bytes=32 time=4ms TTL=64
Reply from 192.168.0.10: bytes=32 time=2ms TTL=64
Reply from 192.168.0.10: bytes=32 time=4ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 18ms, Average = 7ms

C:\Users\Fernando>

```

Figura 80 Ping distancia con Obstáculo 7[m], fuente propia,

5.1.2 PRUEBA DE SENSORES PARA CALIBRACIÓN

Probar los sensores moviendo manualmente las articulaciones como tanto en brazo como en antebrazo tenemos un acelerómetro de 3 ejes el mismo que nos entrega el ángulo de giro según su eje de rotación.

```

COM3
x: 112.46| y: 165.53 | z: 83.91
x: 112.46| y: 170.84 | z: 86.19
x: 113.75| y: 171.38 | z: 86.19
x: 112.46| y: 170.84 | z: 86.19
x: 111.80| y: 166.87 | z: 84.67
x: 112.46| y: 165.53 | z: 83.91
x: 111.54| y: 166.87 | z: 84.74
x: 112.46| y: 167.28 | z: 84.67
x: 111.80| y: 166.87 | z: 84.67
x: 112.46| y: 167.28 | z: 84.67
x: 111.80| y: 166.87 | z: 84.67
x: 112.73| y: 167.28 | z: 84.60
x: 112.73| y: 167.28 | z: 84.60
x: 112.46| y: 167.28 | z: 84.67
x: 111.80| y: 166.87 | z: 84.67

```

Figura 81 Lectura Sensor Antebrazo ángulo sensor eje x

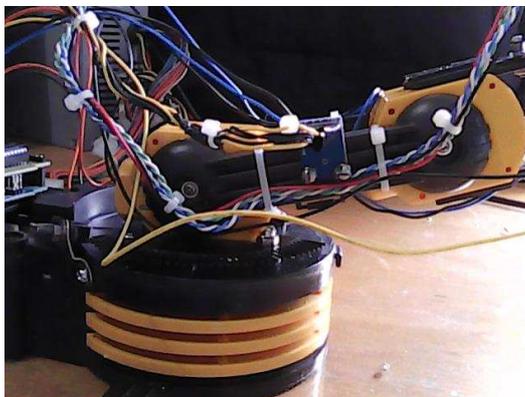


Figura 82 Posición Antebrazo ángulo de 112 eje x, fuente propia

```
COM3
Send
x: 181.13 | y: 181.13 | z: 225.00
x: 180.57 | y: 181.15 | z: 206.57
x: 180.57 | y: 181.15 | z: 206.57
x: 180.57 | y: 181.15 | z: 206.57
x: 180.57 | y: 181.13 | z: 206.57
x: 181.13 | y: 182.27 | z: 206.57
x: 181.15 | y: 182.29 | z: 206.57
x: 180.57 | y: 181.15 | z: 206.57
x: 180.57 | y: 181.13 | z: 206.57
x: 181.13 | y: 181.13 | z: 225.00
x: 181.13 | y: 181.13 | z: 225.00
x: 180.57 | y: 181.15 | z: 206.57
x: 180.57 | y: 181.15 | z: 206.57
x: 181.13 | y: 181.13 | z: 225.00
x: 180.57 | y: 181.13 | z: 206.57
x: 18
```

Autoscroll No line ending 9600 baud

Figura 83 Lectura Sensor Antebrazo ángulo sensor eje x

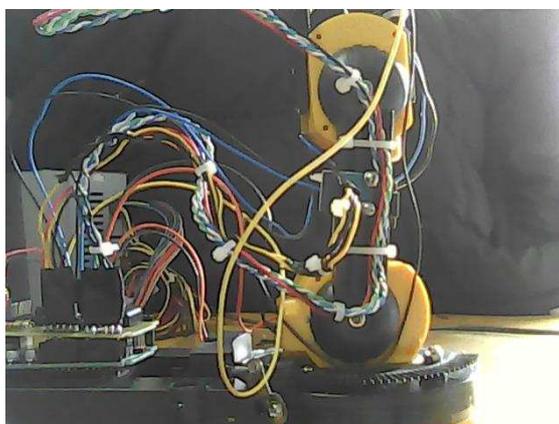


Figura 84 Posición Antebrazo ángulo de 180 eje x, fuente propia

El sensor del brazo se encuentra ubicado de forma horizontal razón por la cual el sensor tiene mayor variación de movimientos en el eje x que en los demás.

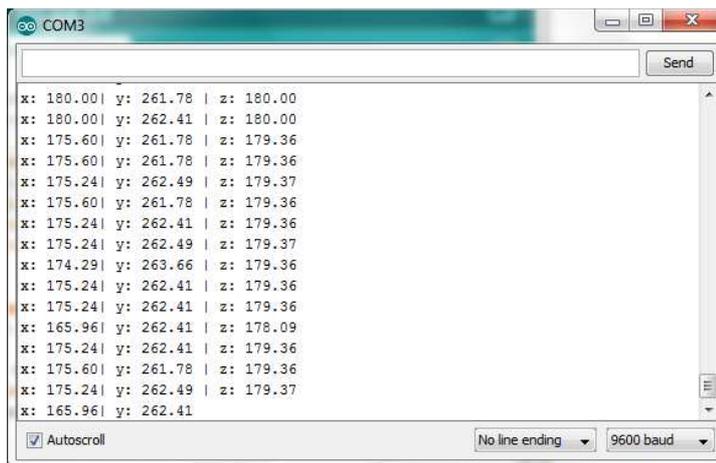


Figura 85 Lectura Sensor Brazo ángulo sensor eje x

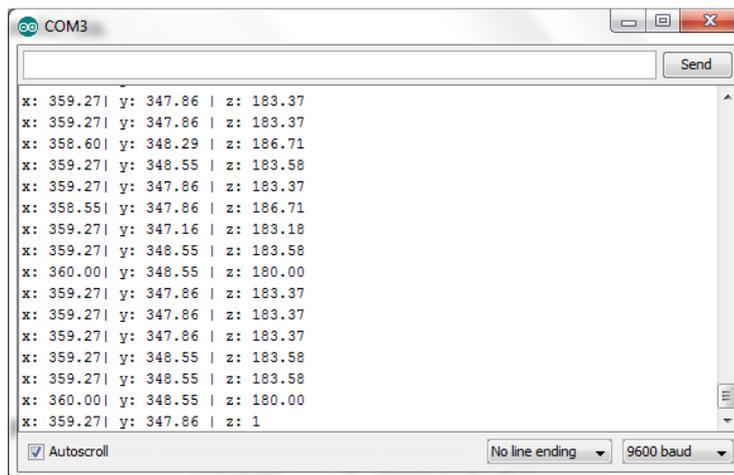


Figura 86 Lectura Sensor Brazo ángulo sensor eje x, fuente propia

5.1.3 ALCANCE DE VISUALIZACIÓN DEL

5.1.4 USUARIO A TRAVÉS DE SENSOR KINECT

El rango de reconocimiento de objetos de Kinect es aproximadamente 3[m] de longitud, consideremos que aunque el sensor detecte al usuario a corta distancia el rango de reconocimiento de cada una de sus articulaciones podría variar, de la misma forma si el usuario se encuentra a una distancia considerable aunque detecte el sensor este no podrá hacer un buen reconocimiento de cada una de sus articulaciones.

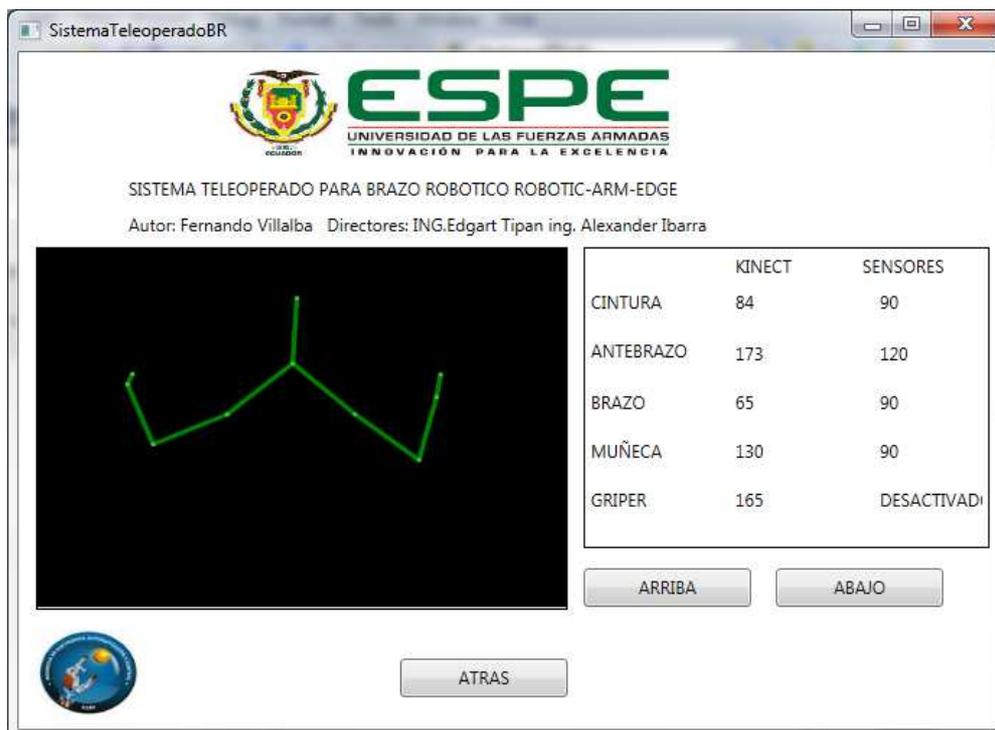


Figura 87 Detección de usuario 1[m], fuente propia

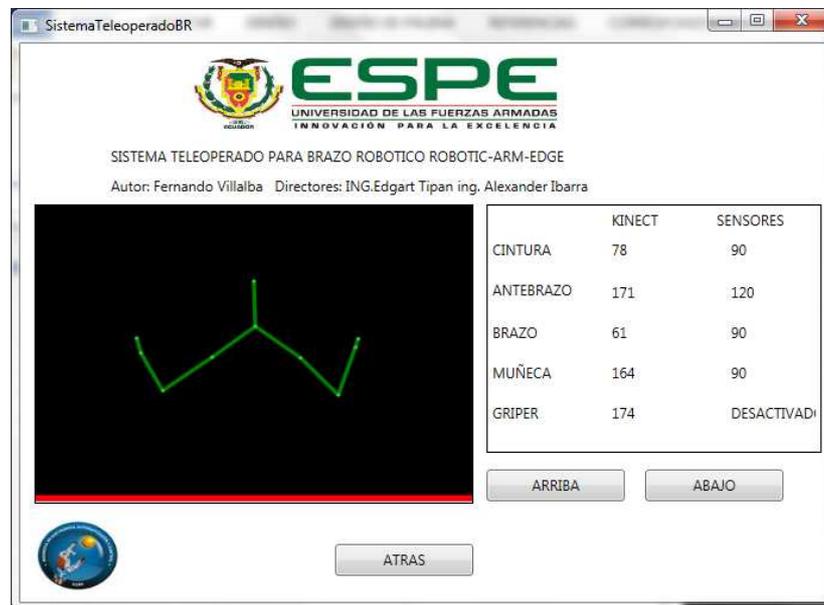


Figura 88 Detección de usuario 1.25 [m], fuente propia

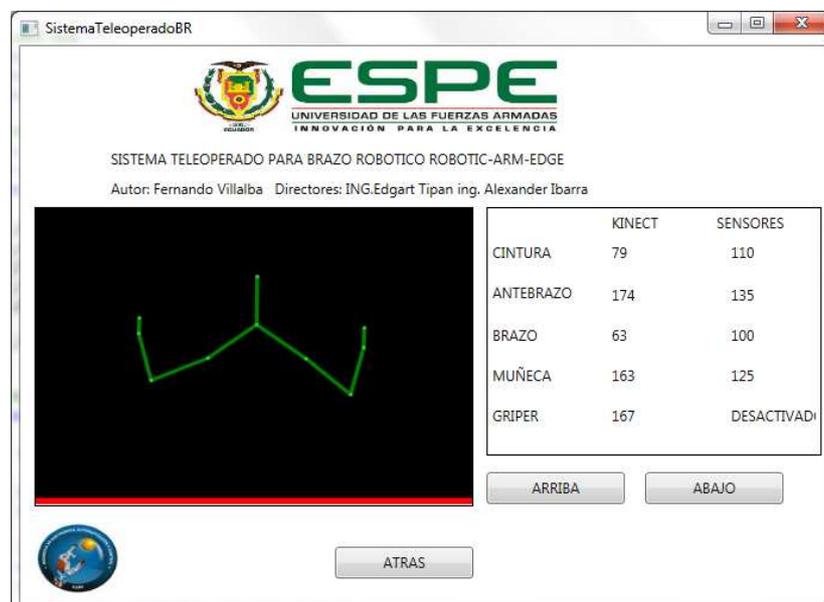


Figura 89 Detección de usuario 1.5 [m], fuente propia

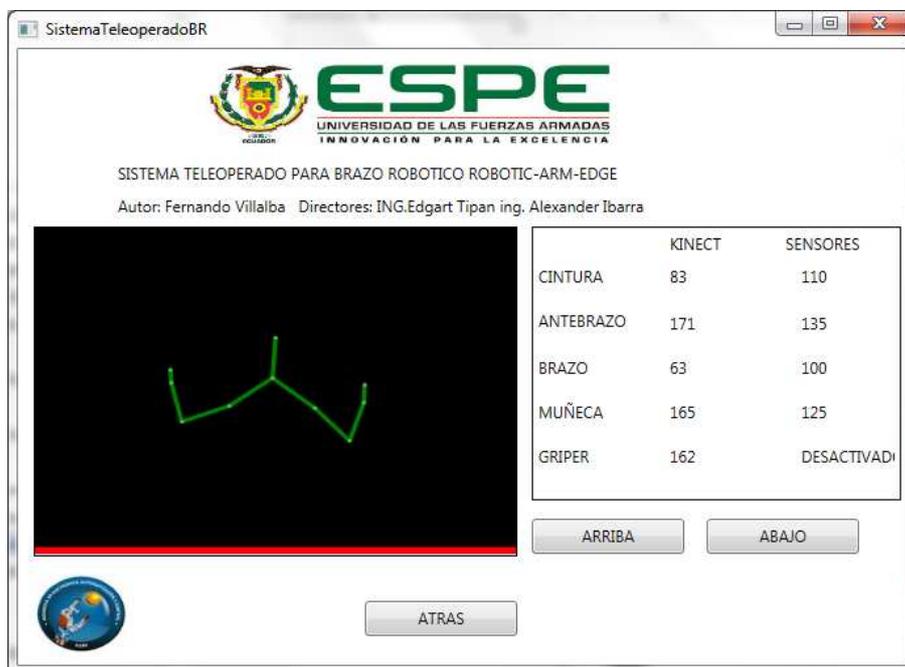


Figura 90 Detección de usuario 2 [m], fuente propia

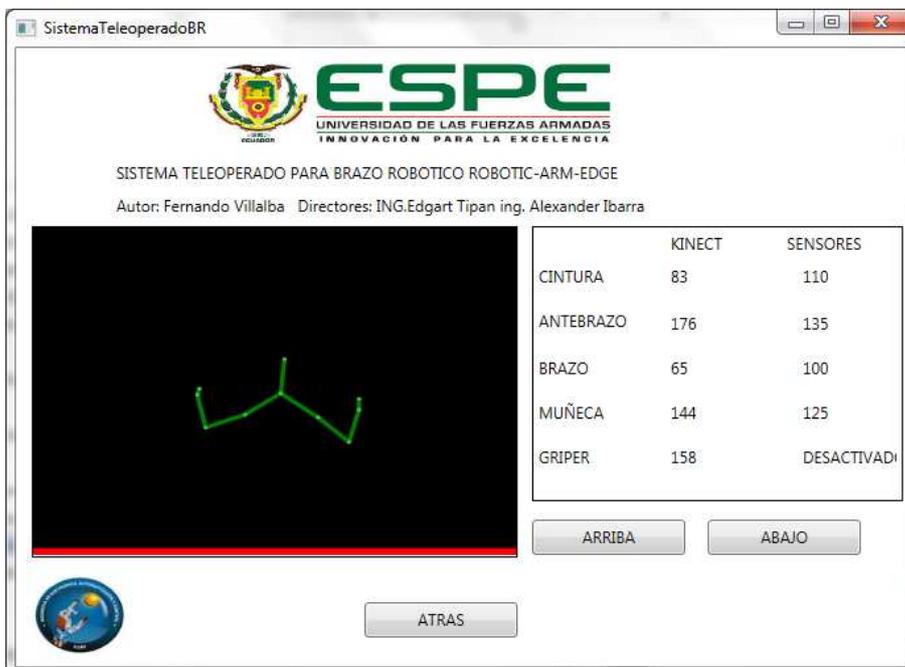


Figura 91 Detección de usuario 2.5 [m], fuente propia

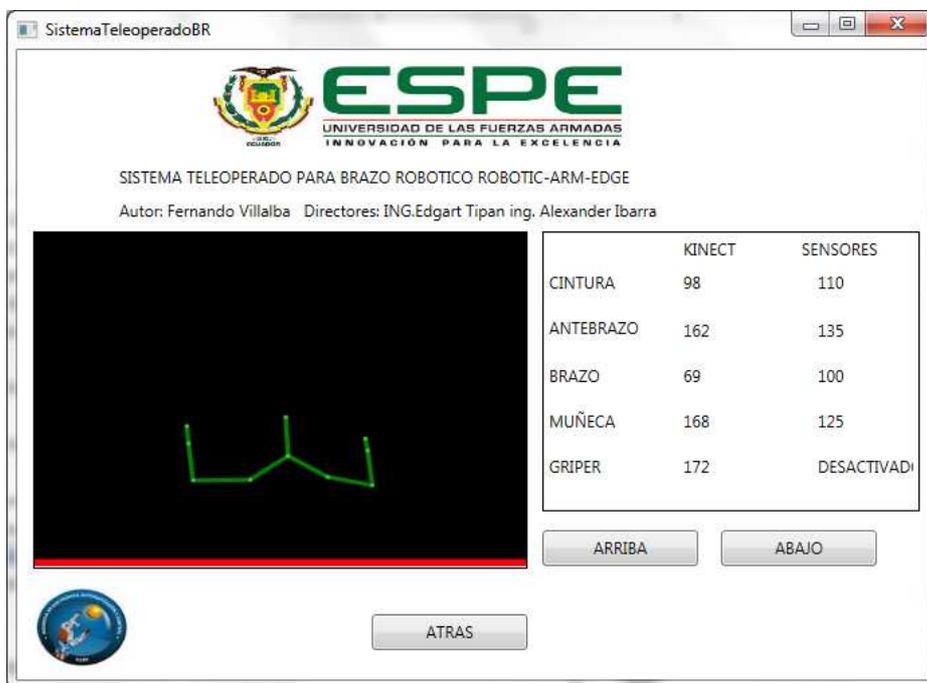


Figura 92 Detección de usuario 3 [m], fuente propia

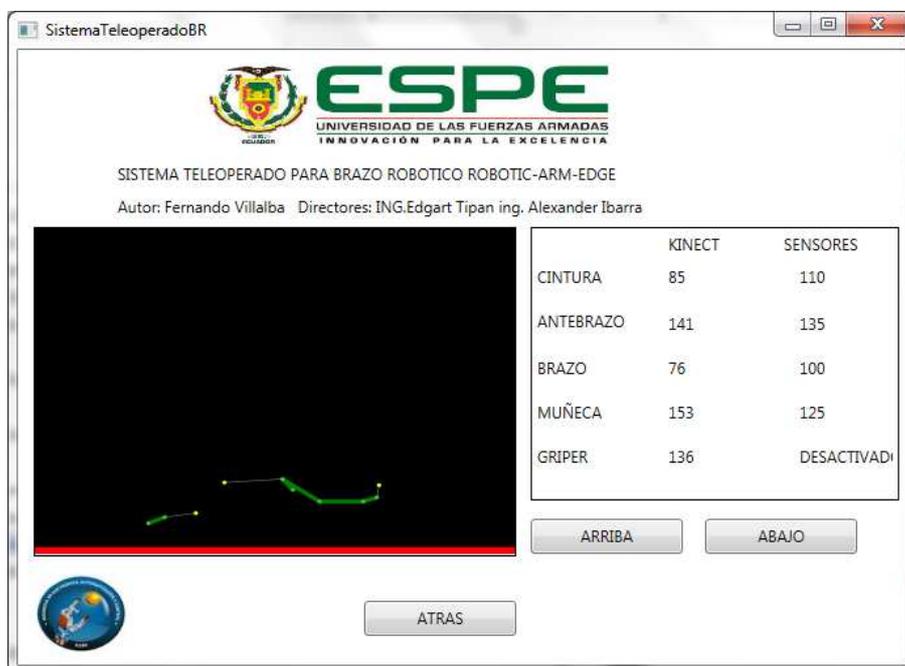


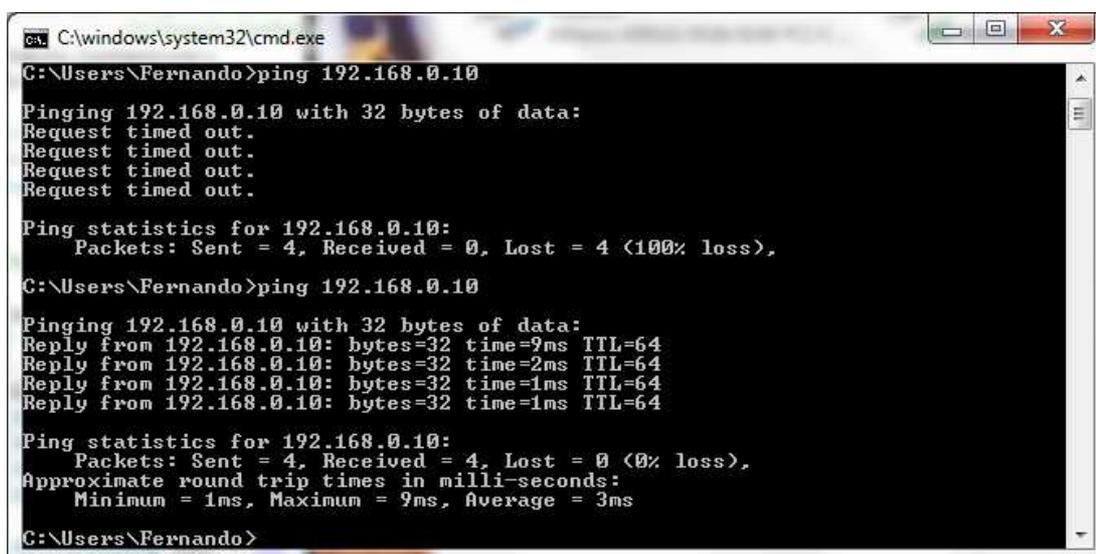
Figura 93 Detección de usuario 3.5 [m], fuente propia

5.1.5 PRUEBA DE DECODIFICACIÓN DE LA TRAMA DE DATOS RECEPTADOS EN RASPBERRY

El envío de datos a través de una red, se realiza mediante tramas, por tal motivo hay que crear el paquete de datos que deseamos transmitir, cada ángulo medido de la articulación, se las junta y separamos con un carácter distintivo para cuando llegue al receptor haga el respectivo desglose de datos que necesita.

Forma del paquete de datos:

setPointBrazo+'a'+setPointAnteBrazo+'b'+setPointBase+'c'+setPointMuñeca+'d', activación/desactivación Gripper



```

C:\windows\system32\cmd.exe
C:\Users\Fernando>ping 192.168.0.10
Pinging 192.168.0.10 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Fernando>ping 192.168.0.10
Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.10: bytes=32 time=9ms TTL=64
Reply from 192.168.0.10: bytes=32 time=2ms TTL=64
Reply from 192.168.0.10: bytes=32 time=1ms TTL=64
Reply from 192.168.0.10: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 9ms, Average = 3ms

C:\Users\Fernando>

```

Figura 94 Ping de envío/Recepción de datos, fuente propia

5.2 RESULTADOS

El sistema asimila las ordenes de mando enviadas por el sensor Kinect, las mismas que tienen un retardo de dos segundos a causa de la pérdida de

datos en el envío vía Wireless, otro factor que influye es la validación de datos aceptados para el sistema, caso que no sean aptos el control los rechaza hasta tener una nueva orden de setPoint, este rechazo ocasiona breves pero sentidos retardos para el seguimiento del usuario.



Figura 95 Seteo de Posiciones

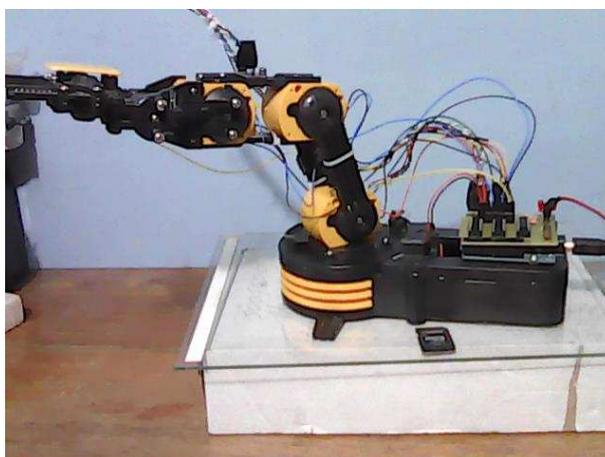


Figura 96 Movimiento. Brazo robótico según setPoint

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- Se diseña e implementa una aplicación en Visual c# capaz de realizar el reconocimiento visual de los movimientos de las articulaciones del usuario a través del sensor Kinect.
- La aplicación en visual C# permite medir los ángulos de apertura de cada articulación superior del usuario, y su variación en el tiempo es inmediata.
- Se aprecia que cada punto medido por Kinect tiene un espacio definido en un eje de coordenadas x,y,z, lo que nos da la facilidad de obtener un vector de cada articulación del usuario.
- La aplicación en Visual C# permite enviar los valores medidos por el sensor Kinect, además permite recibir los valores de los sensores incorporados en el brazo robótico a través de un medio inalámbrico Wireless.
- Mediante sockets se puede establecer la comunicación bidireccional, permitiendo enlazar en red dos sistemas operativos diferentes, así como comunicar dos programas de distinto fabricante.
- Se determinó la trama de datos que deseamos enviar por la red o recibirla según el caso, puesto que si no se los decodifica de forma adecuada tienden a leer datos erróneos por el desplazamiento de los valores en la trama que puede ser por pérdida de datos, o datos almacenados en el buffer de salida de red.
- Se determinó que una forma eficiente de comunicar dos o más programas de diferentes características o compartir datos entre

sistemas operativos de diferente proveedor es mediante programación vía sockets.

- Raspberry resultó eficiente para adquisición de los datos enviados por el Kinect pero la ausencia de entradas análogas obligaron a la incorporación de una tarjeta programable arduino.

6.2 RECOMENDACIONES

- Para hacer aplicaciones en Kinect se debe considerar la versión de librería SDK que se va ocupar.
- Si se va emplear librerías de Esqueletización mediante Kinect, se debe considerar si el desarrollo será en visual c# o mediante visual c++, puesto que las librerías varían entre ellos, debido al formato de desarrollo de cada uno.
- Para detección al envío de datos a través de la comunicación inalámbrica hay que verificar que ambos equipos estén unidos en la misma red.
- El envío y recepción de datos empleando sockets a través de la configuración cliente servidor es eficiente pero entre cada envío o recepción se debe esperar un tiempo prudencial entre cada lectura.

REFERENCIAS

- Bejarano, M. (20 de Julio de 2015). Frabuesa Pi Colombia. Obtenido de <http://www.frambuesapi.co/2013/09/09/tutorial-4-parte-2-conexion-del-raspberry-pi-directamente-al-pc/>
- C.V., L. I. (10 de Julio de 2015). LBA Industrial Mining y Cía S. de R.L. de C.V. Obtenido de <http://www.lbaindustrial.com.mx/que-es-un-encoder/>
- Campos, A. (7 de Julio de 2011). Microsoft Developer España. Obtenido de Reto SDK de Kinect: Desarrolla con Kinect: <http://blogs.msdn.com/b/esmsdn/archive/2011/07/07/sdk-de-kinect-desarrolla-con-kinect.aspx>
- Carletti, E. (15 de Junio de 2015). Robots, pasión por la robótica argentina. Obtenido de http://robots-argentina.com.ar/MotorCC_L293D.htm
- Castillo, A. (15 de Marzo de 2015). Herramientas WEB par ala enseñanza de protocolos de comunicación. Obtenido de <http://neo.lcc.uma.es/evirtual/cdd/tutorial/transporte/tcp.html>
- Cuartiles, D. (10 de Mayo de 2015). Arduino. Obtenido de Genuino: <https://www.arduino.cc/en/Main/arduinoBoardUno>
- Danny. (27 de Julio de 2014). Godot.Be. Obtenido de <https://securehomes.esat.kuleuven.be/~decockd/wiki/bin/view.cgi/RaspberryPi/SetupRaspberryPiWithWemoDevices>
- Diplomado en Robótica Industrial. (17 de Septiembre de 2014). Obtenido de <http://bogota.ecci.edu.co/index.php/diplomados/981-diplomado-en-robotica-industrial>
- Edge, M. U. (2008). Robotic Arm Edge wired control robotic arm kit. Asembly and Instruction Manual, 33.
- FOUNDATION, R. P. (15 de Marzo de 2015). raspberrypi.org. Obtenido de <https://www.raspberrypi.org/products/model-b-plus/>
- Gonzales, V. R. (3 de Febrero de 2002). Robots industriales. Obtenido de http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/industrial.htm

- Granollers, P. P. (4 de MARZO de 2015). DISEÑO DE PANTALLA. Obtenido de <http://www.epsevg.upc.edu/hcd/material/lecturas/pantalla.pdf>
- <http://www.5hertz.com>. (10 de Julio de 2015). Obtenido de <http://5hertz.com/tutoriales/?p=228>
- Ingenet, I. (17 de Septiembre de 2014). Obtenido de <http://bitacora.ingenet.com.mx/2012/12/¿te-has-preguntado-como-funciona-el-sensor-de-kinect/>
- Kleyner, C. M. (2013). Diseño e implementación de un sistema para análisis de movimiento humano usando sensor Kinect. Cuenca, Azuay, Ecuador.
- MacCormick, J. (20 de Junio de 2015). How does the Kinect work. Obtenido de <http://bitacora.ingenet.com.mx/2012/12/¿te-has-preguntado-como-funciona-el-sensor-de-kinect/>
- Mario, P. (2 de Junio de 2015). Geeky Theory. Obtenido de <https://geekytheory.com/tutorial-raspberry-pi-configurar-wif/>
- Ojeda, L. T. (15 de Julio de 2015). Arduino. Obtenido de <http://arduino.cl/que-es-arduino/>
- PE, I. (12 de Agosto de 2014). Como hacer.eu. Obtenido de <http://comohacer.eu/gpio-raspberry-pi/>
- PÉREZ , L., & SÁNCHEZ , E. (2013). DISEÑO DEL CONTROLADOR Y TELEOPERACIÓN DEL BRAZO ROBÓTICO "ROBOTIC ARM EDGE" POR MEDIO DE UN DISPOSITIVO ANDROID PC. SANGOLQUÍ: ESPE.
- Sarajevo, M. (17 de Septiembre de 2014). Natural User Interface. Obtenido de <http://www.cedim.edu.mx/sobre-cedim/noticias/nui-natural-user-interface/>
- Sebastián, J. (15 de Marzo de 2015). ComuSoft.com. Obtenido de <http://www.comusoft.com/todo-sobre-el-kinect-de-xbox-360-caracteristicas-y-precio>

Terrones, T. S. (17 de Septiembre de 2014). Robots Manipuladores y su importancia en la industria. Obtenido de http://www.tlalpan.uvmnet.edu/oiid/download/Robots%20manipuladores_04_ING_IMECA_PII_E%20E.pdf

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

Sangolquí, 18 de Agosto 2015

ELABORADO POR:



Fernando Javier Villalba Portilla

CI:1718083841

AUTORIDADES:



Ing. Victor Proaño

**DIRECTOR DE LA CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL (E)**