

ESCUELA DE INGENIERÍA EN SISTEMAS

Tema:

"DESARROLLO DE UNA INTERFAZ NATURAL DE USUARIO UTILIZANDO EL SENSOR KINECT PARA LA INTERACCIÓN ENTRE UN HUMANO Y UNA PLATAFORMA ROBÓTICA EDUCATIVA"

Disertación de Grado previa la obtención del título de
Ingeniero de Sistemas y Computación

Línea de Investigación:

INTELIGENCIA ARTIFICIAL, ROBÓTICA, DOMÓTICA Y SISTEMAS EXPERTOS

Autor:

ÁLVARO ANDRÉS CÁRDENAS SALAZAR

Director:

Ing. Msc. DARÍO JAVIER ROBAYO JÁCOME

Ambato - Ecuador

Diciembre – 2015

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE AMBATO

HOJA DE APROBACIÓN

Tema:

"DESARROLLO DE UNA INTERFAZ NATURAL DE USUARIO UTILIZANDO EL SENSOR KINECT PARA LA INTERACCIÓN ENTRE UN HUMANO Y UN PLATAFORMA ROBÓTICA EDUCATIVA"

Línea de Investigación:

INTELIGENCIA ARTIFICIAL, ROBÓTICA, DOMÓTICA Y SISTEMAS EXPERTOS

Autor:

ÁLVARO ANDRÉS CÁRDENAS SALAZAR

Darío Javier Robayo Jácome Ing. Mg. f._____

CALIFICADOR

John Osvaldo Ortega Castro Ing. Mg. f._____

CALIFICADOR

Santiago Alejandro Acurio Maldonado Ing. Mg. f._____

CALIFICADOR

Teresa Milena Freire Aillón Ing. Mg. f._____

DIRECTOR ESCUELA DE INGENIERÍA DE SISTEMAS

Hugo Rogelio Altamirano Villarroel Dr. f._____

SECRETARIO GENERAL DE LA PUCESA

Ambato – Ecuador

Diciembre - 2015

DECLARACIÓN DE LA AUTENTICIDAD Y RESPONSABILIDAD

Yo, Alvaro Andrés Cárdenas Salazar portador de la cédula de ciudadanía Nº. 0603554726 declaro que los resultados obtenidos en la investigación que presento como informe final, previo la obtención del título de Ingeniero de Sistemas y Computación son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola y exclusiva responsabilidad legal y académica.

Alvaro Andrés Cárdenas Salazar

C.I: 0603554726

DEDICATORIA

A Dios creador de todas las cosas, él me ha dado fortaleza para continuar en los momentos difíciles, de igual forma dedico esta disertación a mi madre y a mi tía por su amor, trabajo y sacrificio durante todos estos años.

RESUMEN

La presente disertación tiene como objetivo desarrollar una Interfaz Natural de Usuario, para la interacción entre un humano y una plataforma robótica educativa. Se utilizó el método experimental que ayudó a recolectar información necesaria para la elaboración de la aplicación, permitiendo la identificación de soluciones óptimas, para el desarrollo del aplicativo se utilizó la metodología de prototipos, la misma que permite el mejoramiento continuo, las principales herramientas usadas para el desarrollo fueron: el sensor Kinect de XBOX 360, el SDK oficial de Microsoft para desarrolladores, Visual Studio Express 2013 para Escritorio, el lenguaje de programación Visual C# con un proyecto WPF (Windows Presentation Fundation) y la plataforma Robótica Educativa RoboBuilder, la aplicación desarrollada demostró ser una alternativa eficiente, eficaz y de fácil uso para la tele-operación del robot por parte del usuario.

Palabras Claves: interfaz natural de usuario (nui), visual c#, sensor kinect, sdk de kinect, visual studio expres 2013.

ABSTRACT

The objective of this project is to develop a natural user interface for the interaction between a human and an educational robotics platform. The experimental method was used, which helped to collect necessary information in order to create the application allowing for the identification of optimal solutions. To develop the application, prototyping methodology was used, which allows for continuous improvement. The main tools that were used for the development were: XBOX 360 Kinect sensor, Microsoft SDK for developers, Visual Studio Express 2013 for Desktop, Visual C++ programming language with a WPF (Windows Presentation Foundation) project and the educational robotics platform RoboBuilder. The developed application has proved to be an efficient, effective and easy to use alternative for the user's teleoperation of the robot.

Key words: natural user interface (NUI), Visual C#, Kinect sensor, Microsoft SDK, Visual Studio Express 2013.

ÍNDICE DE CONTENIDOS

Preliminares

DECLARACIÓN DE LA AUTENTICIDAD Y RESPONSABILIDAD..... III

DEDICATORIA IV

RESUMEN..... V

ABSTRACT.....VI

ÍNDICE DE CONTENIDOS VII

INTRODUCCIÓN II

CAPITULO I: FUNDAMENTOS TEÓRICOS..... 3

1.1 Antecedentes 3

1.2 Problema..... 4

1.2.1 Descripción del Problema 4

1.2.2 Preguntas básicas 5

1.3 Justificación..... 6

1.3.1 Justificación de Relevancia..... 7

1.3.2 Justificación Técnica..... 7

1.3.3 Justificación Económica..... 8

1.4 Objetivos. 8

1.4.1 Objetivo General..... 8

1.4.2 Objetivos Específicos..... 8

1.5 Meta..... 9

1.6 Fundamentos Teóricos..... 9

1.6.1 ¿Qué es la interfaz de usuario?..... 9

1.6.2 Interfaz Natural de Usuario 10
1.6.3 Robótica 14
1.6.4 Puerto de Comunicación Serial Y Paralela 19
1.6.5 Puertos virtuales 22
CAPITULO II: METODOLOGÍA. 23
2.1 Metodología de investigación 23
2.1.1 Métodos Generales de Investigación 23
2.1.2 Método Específico de Investigación 24
2.1.3 Técnicas de recolección de la información 24
2.1.4 Instrumentos 25
2.2 Metodología de Desarrollo 25
2.2.1 Fase 1: Selección y Análisis 26
2.2.2 Fase 2: diseño rápido del prototipo. 34
2.2.3 Fase 3: Codificación del prototipo 36
CAPITULO III.: RESULTADOS 59
3.1 Fase 1: Movimiento Individual de los Brazos 59
3.1.1 Movimiento lateral del brazo derecho 63
3.1.2 Movimiento hacia adelante del brazo derecho: 65
3.1.3 Movimiento hacia arriba de la extremidad derecha 67
3.1.4 Movimiento lateral del brazo izquierdo 68
3.1.5 Movimiento hacia adelante del brazo izquierdo: 71
3.1.6 Movimiento hacia arriba de la extremidad izquierda 73
3.2 Fase 2: Movimiento en conjunto de los Brazos 74
3.2.1 Movimiento lateral de las dos extremidades superiores 75
3.2.2 Movimiento hacia adelante de las extremidades superiores. 76
3.2.3 Movimiento hacia arriba de las extremidades superiores 78

CAPITULO IV.: ANÁLISIS Y VALIDACIÓN DE RESULTADOS 80

4.1 Análisis de Resultados 80

4.1.1 Análisis de la Distancia del Sensor 82

4.1.2 Análisis de los valores de Servo-Motores y Ángulos Corporales 83

4.2 Validación de resultados 84

CAPITULO V: CONCLUSIONES Y RECOMENDACIONES. 88

5.1 Conclusiones 88

5.2 Recomendaciones 89

BIBLIOGRAFÍA 91

GLOSARIO 95

ANEXOS 101

ÍNDICE DE GRÁFICOS

ILUSTRACIONES

ILUSTRACIÓN 1.1: EVOLUCIÓN DE LA INTERFAZ DE USUARIO 10

ILUSTRACIÓN 1.2: PANTALLA MULTITOUCH 11

ILUSTRACIÓN 1.3: COMANDOS DE VOZ 12

ILUSTRACIÓN 1.4: SENSOR DE FUERZA 13

ILUSTRACIÓN 1.5: ROBOT ASIMO 15

ILUSTRACIÓN 1.6: ROBOT ASIMO 1986 16

ILUSTRACIÓN 1.7: ROBOT NAO DE 25 GRADOS DE LIBERTAD 17

ILUSTRACIÓN 1.8: PLATAFORMA ROBÓTICA EDUCATIVA LEGO MINDSTORMS 18

ILUSTRACIÓN 1.9: COMUNICACIÓN SERIE Y PARALELA 19

ILUSTRACIÓN 1.10: PUERTO PARALELO 20

ILUSTRACIÓN 1.11: PUERTO SERIAL 21

ILUSTRACIÓN 2.1: COMPONENTES DEL SENSOR KINECT 29

ILUSTRACIÓN 2.2: DIFERENCIAS ENTRE KINECT XBOX 360 Y KINECT PARA WINDOWS 30

ILUSTRACIÓN 2.3: ADAPTADOR DE CORRIENTE PARA KINECT DE XBOX 360 31

ILUSTRACIÓN 2.4: DISTRIBUCIÓN SERVOMOTORES 33

ILUSTRACIÓN 2.5: REPRESENTACIÓN GRÁFICA DEL SISTEMA 34

ILUSTRACIÓN 2.6: POSICIÓN INICIAL DE ROBOBUILDER 35

ILUSTRACIÓN 2.7: DIAGRAMA DE FLUJO APLICACIÓN 37

ILUSTRACIÓN 2.8: VENTANA PRINCIPAL DE LA APLICACIÓN 38

ILUSTRACIÓN 2.9: REFERENCIAS AGREGADAS 39

ILUSTRACIÓN 2.10: LIBRERÍAS UTILIZADAS EN EL PROYECTO 39

Ilustración 2.11: Librería TCD.Kinect utilizada en el código XAML de la Aplicación. 40

Ilustración 2.12: Representación de un pixel formato RGB 43

Ilustración 2.13: Cámara RGB 44

Ilustración 2.14: Representación de los puntos infrarrojos 46

Ilustración 2.15: Esqueletos 3D 47

Ilustración 2.16: Esqueleto Digital con el nombre de las Articulaciones . 49

Ilustración 2.17: Representación gráfica de las coordenadas de una articulación 50

Ilustración 2.18: Representación de los puntos y vectores de las extremidades superiores del usuario 51

Ilustración 2.19: Ángulos y Articulaciones ejes X Y Brazo Derecho 52

Ilustración 2.20: Ángulos y articulaciones ejes Y Z Brazo Derecho 53

Ilustración 2.21: Ángulos y Articulaciones ejes X Y Brazo Izquierdo 54

Ilustración 2.22: Ángulos y articulaciones ejes Y Z Brazo Derecho 54

Ilustración 3.1: Movimiento lateral brazo derecho 64

Ilustración 3.2: Movimiento lateral del brazo derecho plataforma robótica educativa 65

Ilustración 3.3: Movimiento hacia adelante brazo derecho 66

Ilustración 3.4: Movimiento hacia adelante del brazo derecho plataforma robótica educativa 66

Ilustración 3.5: Movimiento hacia arriba del brazo derecho del usuario. . 67

Ilustración 3.6: Movimiento hacia arriba del brazo derecho de la plataforma robótica. 68

Ilustración 3.7: Movimiento lateral brazo izquierdo del usuario 70

Ilustración 3.8: Movimiento lateral del brazo derecho plataforma robótica educativa. 71

Ilustración 3.9: Movimiento hacia adelante brazo derecho 72

Ilustración 3.10: Movimiento hacia adelante del brazo derecho plataforma robótica educativa. 72

ILUSTRACIÓN 3.11: MOVIMIENTO HACIA ARRIBA DEL BRAZO DERECHO 73

ILUSTRACIÓN 3.12: MOVIMIENTO HACIA ARRIBA DEL BRAZO IZQUIERDO DE LA PLATAFORMA ROBÓTICA 74

ILUSTRACIÓN 3.13: MOVIMIENTO LATERAL DE LOS BRAZOS 75

ILUSTRACIÓN 3.14: MOVIMIENTO LATERAL DE LOS BRAZOS DE LA PLATAFORMA ROBÓTICA..... 76

ILUSTRACIÓN 3.15: MOVIMIENTO HACIA DELANTE DE LOS BRAZOS 77

ILUSTRACIÓN 3.16: MOVIMIENTO HACIA DELANTE DE LOS BRAZOS DELA PLATAFORMA ROBÓTICA..... 77

ILUSTRACIÓN 3.17: MOVIMIENTO HACIA ARRIBA DE LOS BRAZOS 78

ILUSTRACIÓN 3.18: MOVIMIENTO HACIA ARRIBA DE LOS BRAZOS DE LA PLATAFORMA ROBÓTICA: 79

ILUSTRACIÓN 4.1: ANÁLISIS DE RESULTADOS DEL SISTEMA..... 81

ILUSTRACIÓN 4.2: DISTANCIA DE 1 METRO ENTRE EL USUARIO Y EL SENSOR..... 82

ILUSTRACIÓN 4.3: DISTANCIA INFERIOR A UN METRO ENTRE EL USUARIO Y EL SENSOR KINECT..... 83

ILUSTRACIÓN 4.4: PREGUNTA UNO 85

ILUSTRACIÓN 4.5: PREGUNTA DOS 85

ILUSTRACIÓN 4.6: PREGUNTA TRES..... 86

ILUSTRACIÓN 4.7: PREGUNTA CUATRO..... 86

TABLAS

TABLA 2.1: SENSOR KINECT REQUERIMIENTOS DE HARDWARE Y SOFTWARE 27

TABLA 2.2: SDK DE KINECT PARA WINDOWS, CONCEPTOS Y CARACTERÍSTICAS PRINCIPALES. 31

TABLA 2.3: DECLARACIÓN DE VARIABLE MIKINECT..... 41

TABLA 2.4: CÓDIGO FUENTE PARA ACTIVAR CÁMARA RGB DE KINECT..... 44

TABLA 2.5: CÓDIGO FUENTE PARA ACTIVAR CÁMARA DE PROFUNDIDAD DE KINECT.. 48

TABLA 2.6: CÁLCULOS SERVOMOTORES..... 58

TABLA 3.1: CÓDIGO FUENTE CALCULO DE ÁNGULOS Y VECTORES 60

TABLA 3.2: CÓDIGO FUENTE MOVIMIENTOS BRAZO DERECHO PLATAFORMA ROBÓTICA 63

TABLA 3.3: CÓDIGO FUENTE MOVIMIENTOS BRAZO IZQUIERDO PLATAFORMA ROBÓTICA 69

TABLA 4.1: HARDWARE Y SOFTWARE UTILIZADO EN LAS PRUEBAS 81

TABLA 4.2: POSICIONES SERVO-MOTORES Y ÁNGULOS CORPORALES EXTREMIDAD IZQUIERDA 83

TABLA 4.3: POSICIONES SERVO-MOTORES Y ÁNGULOS CORPORALES EXTREMIDAD DERECHA 84

INTRODUCCIÓN

En el presente trabajo de investigación titulado “Desarrollo de una Interfaz Natural de Usuario Utilizando el Sensor Kinect para la Interacción entre un Humano y una Plataforma Robótica Educativa”, tiene como objetivo utilizar las nuevas tecnologías para permitir al usuario la tele-operación de RoboBuilder mediante el movimiento de los brazos.

En el primer capítulo se detalla los antecedentes, el problema que originó la presente investigación los objetivos, y la meta alcanzada. La fundamentación teórica para la realización del proyecto y los objetivos que se van alcanzar también son abordados en este capítulo.

En el segundo capítulo se detalla la metodología de Investigación usada, cada una de las fases que fueron parte del desarrollo, así como las técnicas de recolección y análisis de la información recabada para la implementación del aplicativo.

En el capítulo tres se muestran los resultados obtenidos, evidenciados con imágenes donde se muestra claramente la funcionalidad de la aplicación, y como mediante el

uso de esta el usuario puede tele-operar la plataforma robótica educativa RoboBuilder.

En el capítulo cuatro se analizan y validan los resultados mostrados en el capítulo tres, mediante gráficos estadísticos.

En el capítulo cinco se muestra las conclusiones y recomendaciones del presente proyecto investigativo, previo a la obtención del título de Ingeniero de Sistemas y Computación.

CAPÍTULO I.

FUNDAMENTOS TEÓRICOS

1.1 Antecedentes

La revolución tecnológica en los últimos años con el desarrollo de nuevos dispositivos ha traído nuevas formas de interactuar con ellos, la Interfaz Natural de Usuario ha permitido que los usuarios tengan mejores experiencias en el uso de estos dispositivos. Es común ver que este tipo de interfaces son utilizadas en pantallas táctiles de diferentes dispositivos como celulares, tablets, computadores personales, etc., además por consolas de video juegos como XBOX, Nintendo y Play Station, que utilizan sensores para el reconocimiento de gestos.

El área de la robótica también hace uso de este nuevo paradigma de interacción, así vemos como en países desarrollados buscan nuevas utilidades a estas tecnologías, un ejemplo concreto es la investigación que realiza la NASA, científicos de esta agencia del gobierno americano en una de sus muchas experimentaciones han propuesto una alternativa para el control de robots en entornos hostiles para los humanos, incluso en el espacio exterior. Para lograr este objetivo han utilizado un auricular Oculus Rif y Kinect, siendo el experimento un éxito controlando un brazo robótico de alta precisión. (Rosenberg , 2013)

Pero esta no es la única investigación que se ha realizado, muchos aficionados, académicos e investigadores, han experimentado creando una Interfaz Natural de Usuario (NUI) que permite controlar robots menos avanzados orientados a realizar diferentes tareas (Ackerman, 2011). La experimentación con robots en el ámbito educativo se realiza utilizando plataformas robóticas que en comparación con otro tipo de robots son más accesibles tanto en costo como en disponibilidad en el mercado. Son de fácil ensamblaje y permiten codificar diferente tipo de aplicaciones con niveles de dificultad variable.

1.2 Problema

El problema está enfocado en encontrar una alternativa innovadora y que sea de fácil aprendizaje para el usuario la tele-operación de la plataforma robótica educativa RoboBuilder.

1.2.1 Descripción del Problema

Este trabajo investigativo tiene como finalidad hacer uso de las nuevas tecnologías y utilizarlas tomando en cuenta el contexto cultural del país, la Interfaz Natural de Usuario NUI, es una aplicación informática innovadora, que permite al usuario interactuar con un dispositivo electrónico de manera natural, haciendo uso de gestos propios del ser humano como mover las manos, brazos, piernas etc. todo esto para cumplir con una tarea o función específica. (Ackerman, 2011)

Estos movimientos que son considerados naturales o comunes facilitan enormemente el uso de los dispositivos, y mejoran la experiencia general del usuario. Los robots en su mayoría son controlados por personal altamente especializado en esta área, mediante dispositivos tradicionales, lo que se constituye en una limitante para el desarrollo de mecanismos robóticos más libres. (Ackerman, 2011). Razón por la cual la esta tecnología es de difícil acceso para todas las personas.

1.2.2 Preguntas básicas

¿Cómo mejora la funcionalidad de un robot una Interfaz Natural de Usuario?

La funcionalidad que un robot experimenta con una Interfaz Natural de Usuario es significativa, ya que no se limita a una rutina de movimientos sino se encuentra supeditada a los movimientos del usuario dándole más libertad al robot.

¿Cómo se controla un robot mediante una Interfaz Natural de Usuario?

El control se da mediante la captura de movimientos utilizando el sensor Kinect, y una aplicación que procese los movimientos capturados.

¿Qué hardware y software son necesarios para el Desarrollo de una Interfaz Natural de Usuario para la interacción Humano - Robot?

En el desarrollo de una Interfaz Natural de Usuario para la interacción Humano Robot, es necesario el sensor Kinect, el entorno de desarrollo Visual Studio .Net y el SDK de Kinect.

1.3 Justificación

La Interfaz Natural de Usuario (NUI), en los últimos años ha tenido un crecimiento acelerado aprovechando la disponibilidad de las nuevas tecnologías para el reconocimiento de gestos, imágenes y sonidos, su uso se ha popularizado por las grandes ventajas que presenta al usuario, ya que le permite acceder fácilmente a diversos dispositivos electrónicos sin la necesidad de un aprendizaje o conocimiento avanzado.

Los casos más comunes de uso de la Interfaz Natural de Usuario se da en los juegos de video, celulares y computadores con pantalla multitáctil, Microsoft es la empresa pionera en poner a disposición de los desarrolladores un kit oficial de herramientas para aprovechar la tecnología del sensor de movimiento Kinect, no solo en el campo del entretenimiento doméstico, sino también permite utilizar esta tecnología en el desarrollo de aplicaciones de todo tipo aprovechando la Interfaz Natural de Usuario y la detección de movimientos corporales.

Una plataforma robótica educativa, permite realizar investigaciones para encontrar nuevas funcionalidades a los robots, el desarrollo de una Interfaz Natural de Usuario para la interacción entre una persona y un robot, le dota a este de ciertas características más cercanas a las humanas, ya que el robot no estará limitado por una rutina de movimientos programados sino que imitara los movimientos del usuario.

1.3.1 Justificación de Relevancia

Se tendrá máxima acogida ya que la innovación tecnológica es una de las características destacadas de la Escuela de Ingeniería en Sistemas de la PUCESA, es por este motivo que el presente trabajo de investigación propone una nueva alternativa para que el usuario interactúe un robot mediante movimientos corporales, dejando de lado los medios tradicionales de interacción como: control remoto joystick, etc. Además de adaptar la tecnología disponible al entorno cultural y social del País.

1.3.2 Justificación Técnica

En la implementación de una interfaz natural de usuario para la interacción entre un humano y una plataforma se desarrollara poniendo en práctica todos los conocimientos adquiridos a lo largo de la carrera de Ingeniería de Sistemas en la Pontificia Universidad Católica del Ecuador Sede Ambato, utilizando los entornos de

programación: Visual Studio .Net SDK de Kinect y la plataforma robótica RoboBuilder 5710K.

1.3.3 Justificación Económica

El desarrollo de una interfaz natural de usuario para la interacción entre un humano y una plataforma robótica educativa cuenta con los recursos necesarios para su elaboración

1.4 Objetivos.

1.4.1 Objetivo General

- Desarrollar una Interfaz Natural de Usuario utilizando el sensor Kinect para la Interacción entre un Humano y una Plataforma Robótica Educativa.

1.4.2 Objetivos Específicos

- Estudiar los entornos de desarrollo Visual Studio, SDK de Kinect para diseñar la Interfaz Natural de Usuario.

- Analizar el funcionamiento de la plataforma robótica RoboBuilder y los protocolos de comunicación con un computador.

- Desarrollar una aplicación con una Interfaz Natural de Usuario para el control de la plataforma robótica RoboBuilder.

1.5 Meta

Desarrollo de una Interfaz Natural de Usuario, utilizando el sensor Kinect para la interacción entre un humano y la plataforma robótica educativa RoboBuilder, la misma que al no poseer: un acelerómetro y giroscopio, para mantener el equilibrio únicamente permite la tele operación de las extremidades superiores.

1.6 Fundamentos Teóricos

1.6.1 ¿Qué es la interfaz de usuario?

Interfaz de usuario se define como: el vínculo por medio del cual los usuarios interactúan con las computadoras, mediante protocolos y técnicas, por medio de la interfaz se solicitan comandos o acciones a los usuarios y el computador devuelve resultados. (Rivera, 2000)

La interfaz debe ser de fácil aprendizaje. El usuario final es la persona más importante en el desarrollo de la interfaz. A lo largo de la evolución de la interfaz se han presentado grandes cambios (véase ***Ilustración 1.1***), comenzando con el procesamiento

por lotes, la interfaz de línea de comandos (CLI), la interfaz gráfica de usuario (GUI), y ahora la interfaz natural de usuario (NUI). (Amuno, 2009)

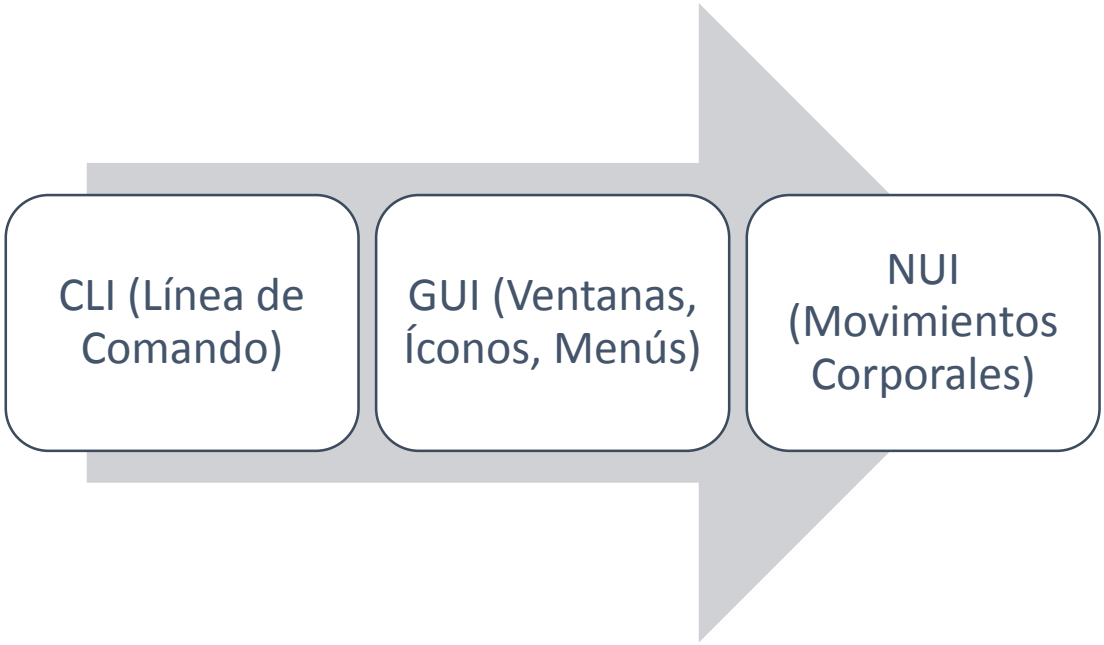


Ilustración 1.1*:* Evolución de la Interfaz de Usuario

Fuente: *(Rees, 2010)*

1.6.2 Interfaz Natural de Usuario

Una interfaz natural de usuario es aquella que permite interactuar con un dispositivo obviando el uso de periféricos tradicionales como el teclado, mouse, joystick etc. para la interacción utiliza movimientos corporales que son detectados por sensores especializados, pantallas táctiles voz humana etc.

Esta interfaz utiliza diferentes medios electrónicos para la interacción, los más utilizados son:

1.6.2.1 Pantalla Multitáctil

Celulares y computadoras son los dispositivos electrónicos donde está presente esta tecnología, la interacción con el usuario se da a través de sensores capacitivos, capaces de detectar una ligera pulsación hasta el arrastre de un dedo, pero estos sensores no son capaces de diferenciar entre un dato, no pueden determinar si es un dedo, una oreja u otra parte del cuerpo la que está tocando la pantalla (véase ***Ilustración 1.2***), para esto requieren un sistema de detección multitoque. (NEOTEO, 2009)



Ilustración 1.2: Pantalla MultiTouch

Fuente: *(Electronics Weekly, 2012)*

1.6.2.2 Comandos de voz

Aunque no es una tecnología nueva los actuales algoritmos han permitido que estos sistemas sean más sofisticados, el desarrollo de estas aplicaciones permiten realizar tareas como el dictado de mensajes de texto, abrir aplicaciones, etc. Smartphones como Iphone y su sistema Siri permiten al usuario interactuar mediante comandos de voz con el dispositivo. (Bejarano, 2013), esta innovación también están presente en el sistema Android (véase ***Ilustración 1.3***).

Cada vez son más comunes las aplicaciones que utilizan esta tecnología. El Sensor Kinect también posee una funcionalidad de reconocimiento de voz, que aunque no tiene una gran fiabilidad permite realizar tareas con comandos de voz.



Ilustración 1.3: Comandos de Voz

Fuente: *(Barrios, 2014)*

1.6.2.3 Sensores

Para (Peña, 2012), un sensor es: "un dispositivo para detectar y señalar una condición de cambio. Con frecuencia, una condición de cambio, se trata de la presencia o ausencia de un objeto o material (detección discreta). También puede ser una cantidad capaz de medirse, como un cambio de distancia, tamaño o color (detección analógica). Los sensores posibilitan la comunicación entre el mundo físico y los sistemas de medición y/o de control, tanto eléctricos como electrónicos, utilizándose extensivamente en todo tipo de procesos industriales y no industriales para propósitos de monitoreo, medición, control y procesamiento". Los sensores son fundamentales en cualquier sistema electrónico, la automatización de procesos es una de las funcionalidades de estos dispositivos, (véase ***Ilustración 1.4***)



Ilustración 1.4: Sensor de Fuerza

Fuente: *(Vernier, s.f.)*

1.6.3 Robótica

Isaac Asimov fue el precursor de la palabra Robótica para referirse a la tecnología utilizada para los robots. La palabra es de origen checo, hace referencia a trabajo forzado (robota), y han humanoide mecánico (robotnik). (Barea, 2011)

Las definiciones de robot son diversas y dependen de la característica del robot:

- Para (Barea, 2011): "Es un dispositivo reprogramable y multifuncional diseñado para mover materiales, piezas, herramientas o dispositivos especializados a través de movimientos programados" Robot Institute of America, 1979.

- Según (Webster, 2014) es: "Un dispositivo automático que realiza funciones que normalmente se considera son o debieran ser realizadas por humanos"

- Según la (RAE, 2014) es: "Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas sólo a las personas."

En conclusión un robot es un dispositivo electrónico formado por distintos componentes que pueden realizar una o muchas tareas específicas que son programadas previamente, dependiendo de su diseño son utilizados para cumplir trabajos diversos en la industria, la investigación espacial, en trabajos riesgosos para los humanos. Muchos prototipos desarrollados son similares a los humanos en su forma y en las funciones que pueden cumplir.

1.6.3.1 Robots con Estructura Humanoide

La ciencia robótica está teniendo grandes avances alrededor del mundo, el aporte de diversas áreas como: mecánica, electrónica, informática, hacen que la evolución de esta ciencia nos muestre robots cada vez más avanzados como el robot ASIMO de la multinacional HONDA, que es el resultado de casi tres décadas de investigación (Honda, 2011).



Ilustración 1.5: Robot ASIMO

Fuente: *(Honda, 2011)*

Un robot humanoide tiene la estructura anatómica similar a la de un humano, y el objetivo principal de estos mecanismos robóticos, es imitar con la mayor precisión

posible los movimientos de las personas, esto supone un gran reto ya que son muchas las limitaciones que se presentan.

Estos robots están orientados a realizar actividades en ambientes complejos, es decir que tienen múltiples obstáculos, y mantener el equilibrio es el principal reto, es por esta razón que el mayor esfuerzo investigativo se centra en este punto, históricamente estos robots tuvieron su inicio como mecanismos bípedos (Spyros G, 2009), donde la única estructura eran las extremidades inferiores, (véase Ilustración 1.6).



Ilustración 1.6: Robot ASIMO 1986

Fuente: *(Honda, 2011)*

1.6.3.2 Grados de Libertad de un Robot

La anatomía humana está compuesta por un conjunto de articulaciones que permiten mover cada una de las extremidades del cuerpo, así también está constituido un robot independientemente de si es humanoide o no, ya que la estructura básica es: un cuerpo y extremidades.

Los grados de libertad en robot representan los desplazamientos y giros que las extremidades pueden hacer, mediante cálculos matemáticos se puede determinar la posición de estos desplazamientos, generalmente los componentes electrónicos que funcionan como articulaciones pueden hacer un único movimiento, el número de estos actuadores coincide el número de grados de liberta del robot (Wilson, 2014).

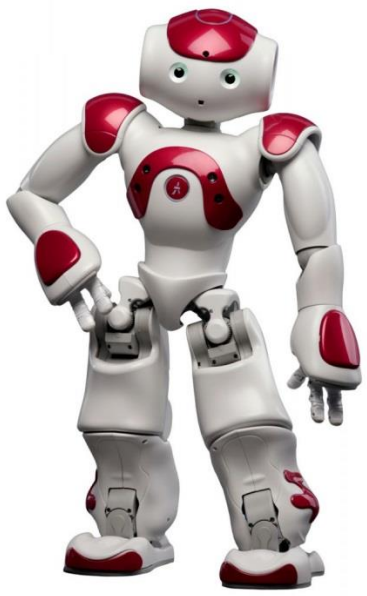


Ilustración 1.7: Robot NAO de 25 Grados de Libertad

Fuente: *(Aldebaran, 2006)*

Mientras más grados de libertad tenga un robot humanoide, más precisos serán sus movimientos, y estos se asemejaran más a los movimientos naturales de un ser humano (véase ***Ilustración 1.7***).

1.6.3.3 Plataforma Robótica Educativa

Una plataforma robótica educativa, es un conjunto de hardware, generalmente de bajo costo, que permite realizar investigaciones en el campo de la robótica, estos kits educativos permiten construir robots de diferentes tipos, porque cuentan con: servomotores, engranes, tornillos, etc., y una placa electrónica programable (véase (LEGO, 2010).



Ilustración 1.8: Plataforma Robótica Educativa LEGO Mindstorms

Fuente: *(LEGO, 2010)*

1.6.4 Puerto de Comunicación Serial Y Paralela

Los puertos de comunicación serial y paralela son elementos que permiten que el computador se pueda comunicar con otros dispositivos como: computadores, routers, módems quemadores de PIC, etc (véase ***Ilustración 1.9***). A pesar de no que en la actualidad no todas las computadoras traen puerto serial o paralelo, existen métodos de simulación para intercambiar información binarizada.

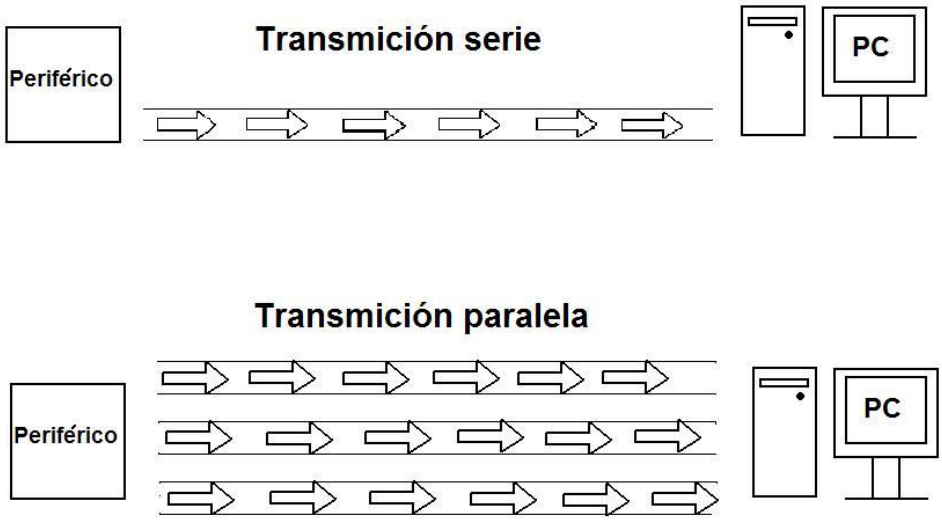


Ilustración 1.9: Comunicación Serie y Paralela

Fuente: *(AMADOR, 2012)*

1.6.4.1 Comunicación Paralela

La comunicación paralela fue utilizada desde sus inicios para la conexión con impresoras, este puerto está conformado por veinticinco pines repartidos en cuatro zonas.

Los pines entre 2 y 9, son los pines de salida, los pines 10, 11, 12, 13 y 15, son utilizados para la entrada de datos, los pines 1, 15, 16 y 17, son utilizados para controlar los distintos periféricos conectados al puerto. Los pines del 18 al 25 son utilizados como tierra (véase ***Ilustración 1.10***). (Universidad del Táchira, 2003).

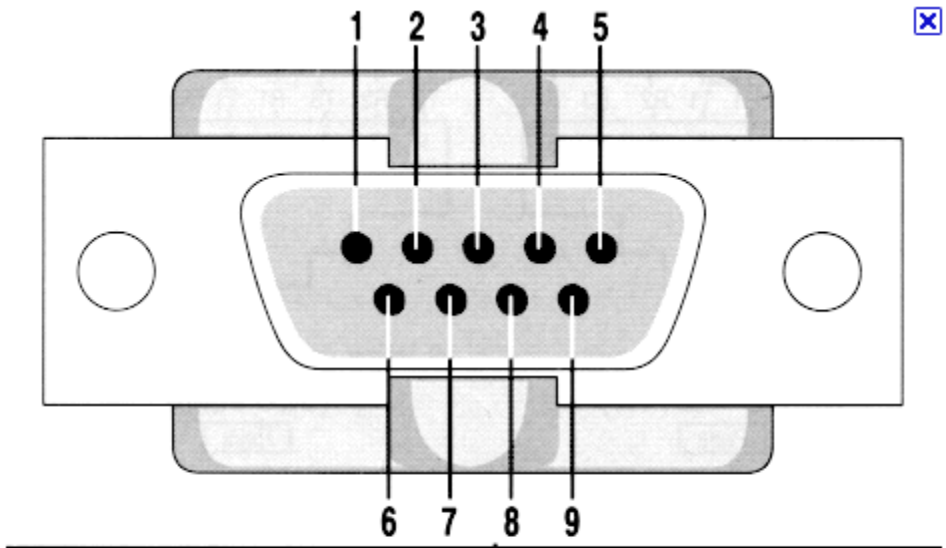


Ilustración 1.10: Puerto Paralelo

Fuente: *(Educación España, 2009)*

1.6.4.2 Comunicación Serial

El puerto serial se comunica a través del protocolo RS-232, la primera versión de este puerto era de 25 pines, luego se simplifico a 9 pines por lo que se denominó a este puerto DB9. Donde los pines que hacen posible la comunicación son: RxD (pin 2) utilizado para la recepción de datos, TxD (pin 3), y el pin GnD (pin 5) que es el pin de tierra (véase ***Ilustración 1.11***). (Universidad del Táchira, 2003)



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

Ilustración 1.11: Puerto Serial

Fuente: *(danirebollo, s.f.)*

Este puerto a pesar de presentar una velocidad de transferencia de datos rápida, por tener muchas líneas de comunicación se atenúa la señal a grandes distancias, la

distancia máxima para una óptima comunicación es de 15 metros, además es más costoso que el puerto paralelo porque tiene más cantidad de líneas de comunicación. (Pérez, 2006)

El estándar RS-232, tiene una conexión sincrónica y asincrónica, la diferencia radica en la existencia de una línea que sincronice todo el sistema, cuando únicamente existe la línea de datos se produce una comunicación asincrónica. (Pérez, 2006)

En una comunicación asincrónica, existen señales que ayudan a gestionar coordinadamente el intercambio de información entre dispositivos.

1.6.5 Puertos virtuales

Para la conexión desde computadores que no disponen de puerto serial o puerto paralelo, existe la posibilidad de conexión mediante el puerto USB, con un adaptador, este adaptador viene con un Driver suministrado por el fabricante. Una vez instalado el Driver se crea un puerto virtual en la computadora, así se puede utilizar el puerto USB como serial o paralelo.

CAPÍTULO II.
METODOLOGÍA

2.1 Metodología de investigación

Durante el desarrollo de una interfaz natural de usuario para la interacción entre un humano y una plataforma robótica, se hará uso de distintos métodos y técnicas de investigación, que permitirán indagar distintas fuentes bibliográficas tanto físicas como digitales con conceptos y definiciones relevantes.

2.1.1 Métodos Generales de Investigación

Los métodos generales de investigación que se utilizarán en el desarrollo del proyecto son:

2.1.1.1 Método de Análisis y Síntesis

El análisis de la bibliografía es fundamental para construir nuevos conocimientos a partir de los conocimientos previos. Sintetizar la información bibliográfica, permite organizar todos la información de manera relevante y organizada.

2.1.1.2 Método deductivo

Con el método deductivo la información recabada será general, la misma que permitirá deducir un concepto concreto que será aplicado en el desarrollo del tema.

2.1.2 Método Específico de Investigación

En el presente proyecto se utiliza el siguiente método de investigación como método específico.

2.1.2.1 Método experimental

El método experimental consta de un conjunto de actividades metódicas y técnicas que permiten obtener toda la información necesaria sobre el tema a investigar, optimizando los recursos humanos, económicos y tecnológicos disponibles. El investigador provoca situaciones controladas o deliberadas como variables de estudio, que son manipuladas para determinar su efecto en el tema de estudio

2.1.3 Técnicas de recolección de la información

La técnica utilizada para reunir toda la información necesaria en el proceso de desarrollo del presente proyecto es la recopilación documental y bibliográfica, esta técnica permite compendiar los más importantes estudios e investigaciones realizados,

esto ayuda a conocer las diversas teorías relacionadas con el tema por parte de otros investigadores o instituciones.

2.1.4 Instrumentos

Para el manejo de la información es necesario la utilización de distintos instrumentos entre los cuales están:

2.1.4.1 Las Fichas

Las fichas son instrumentos que permiten la identificación y registro de las fuentes de información, además facilitan la organización y clasificación de manera eficiente a través de los ficheros. Los tipos de fichas utilizados son: La ficha bibliográfica que permite tener una visión general y ordenada de las fuentes bibliográficas. La ficha de información electrónica donde se registra la información obtenida en internet, CD-ROM.

2.2 Metodología de Desarrollo

En el proceso de desarrollo del sistema propuesto, se utilizará el modelo de prototipos, este modelo se basa en la prueba y el error, esto es muy útil para ir probando las

diferentes alternativas y clarificando las ideas para llegar a cristalizar los objetivos. El desarrollo está dividido en seis fases que se detalla a continuación.

2.2.1 Fase 1: Selección y Análisis

La primera fase de desarrollo consiste en: seleccionar y hacer un análisis detallado de todos los elementos de Software y Hardware que van a formar parte del sistema. Los elementos de Software seleccionados son: SDK Kinect 1.8, Visual Studio Express 2013, Driver RoboBuilder. Los elementos de Hardware seleccionados son: Computadora, Sensor Kinect de XBOX 360 y la plataforma robótica Robobuilder.

El SDK de Kinect 1.8 es la versión más actual y estable para que el sensor Kinect de XBOX 360 pueda interactuar con el computador, ofrece un driver que al ser conectado el sensor, este es reconocido inmediatamente. Para la decodificación el SDK ofrece al desarrollador métodos que permiten manipular las diferentes funcionalidades del sensor. A pesar de existir la versión del sensor Kinect para computadora, esta versión es escasa de encontrar en el mercado y tiene un costo superior, las dos versiones ofrecen las mismas funcionalidades, existen diferencias menores que no afectan en el desarrollo del proyecto.

El SDK es compatible con cualquier versión de Visual Studio a partir de la versión 2010, para el desarrollo de esta aplicación se ha seleccionado la versión 2013 Express para escritorio a más de ser una versión gratuita, a partir de esta versión íntegra todos los lenguajes de programación disponibles, y la creación de diferentes tipos de

proyectos como: Consola, Formularios de Windows y WPF (Windows Presentation Fundation), este último tipo de proyecto facilita la codificación de aplicaciones con interfaces modernas es por esta razón que se codificará con esta opción de desarrollo el aplicativo.

La plataforma robótica educativa RoboBuilder es 100% compatible para ser programada en cualquier lenguaje de Visual Studio, el fabricante tiene disponible una librería que facilita la comunicación entre el computador y RoboBuilder, el costo de esta plataforma robótica es inferior a otras que presentan iguales características.

2.2.1.1 Sensor Kinect

Tabla 2.1:
Sensor Kinect Requerimientos de Hardware Y software

Kinect – Generalidades	
¿Qué es?	Es un sensor que captura el movimiento del usuario, fue desarrollado bajo el nombre de proyecto Natal y es distribuido por Microsoft para la consola Xbox 360.
Requerimientos mínimos de hardware:	Procesador de doble núcleo de 2,66 GHz. Bus USB 2.0. 2 Gb de RAM.

Requerimientos de software:	Windows 7 o Windows 8 (32, 64 Bits). Visual Studio 2010 o superior (cualquier versión). .NET Framework 4.0. SDK para Kinect.

Fuente: (Microsoft, 2011).

El sensor Kinect está conformado por tres cámaras, la primera es un **Proyector de Infrarrojo** que en conjunto con la cámara infrarroja, actúan como sensores de profundidad para detectar en tres dimensiones la posición (eje Z), para lograrlo emite haces de luz infrarroja invisibles para el ojo humano, la **cámara infrarrojo** que captura los puntos que obtienen el proyector infrarrojo que es la tercera cámara , la cámara **RGB**, que captura el movimiento (ejes X y Y) y permite manipular la imagen para obtener sus datos, es similar a una cámara VGA.

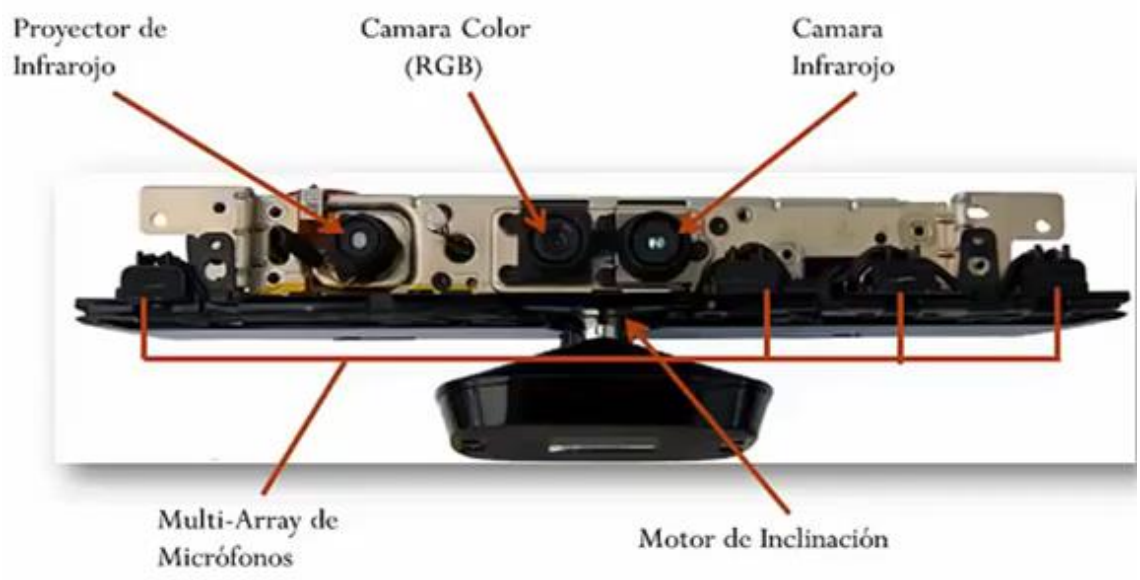


Ilustración 2.1: Componentes del sensor Kinect

Fuente: *(Microsoft, 2011)*

Este sensor además cuenta con un **Multi - Array** de micrófonos en la parte inferior, para la supresión de eco y ruido ambiente, mediante un **acelerómetro** de tres ejes detecta la orientación del sensor. El **motor de inclinación** que se encuentra conectado con la base, permite tener un ángulo de inclinación de 27 grados hacia arriba y hacia abajo para mejorar el enfoque al usuario. Kinect tiene un alcance máximo de captura de 4 metros y un mínimo óptimo de 80 centímetros en la versión Kinect Xbox 360 (véase ***Ilustración 2.1***)

En el mercado están disponibles 3 versiones del sensor que son: Kinect para: XBOX 360, para Pc V1, para Pc V2 y XBOX ONE. En el mercado hay mayor disponibilidad

de la versión XBOX 360 (véase ***Ilustración 2.2Ilustración 2.2:*** *Diferencias entre Kinect XBOX 360 y Kinect para Windows*), esta versión será utilizada para el desarrollo del proyecto.



Ilustración 2.2: Diferencias entre Kinect XBOX 360 y Kinect para Windows

Fuente: Autor

La versión XBOX 360, para ser utilizada en una PC, necesita de un adaptador de corriente que proporciona 12V para su funcionamiento (véase ***Ilustración 2.3***). Mediante el SDK de Kinect los drivers son instalados en el computador.



Ilustración 2.3: Adaptador de corriente para Kinect de Xbox 360

Elaborado Por: Autor

2.2.1.2 Sdk de Kinect para Windows

Tabla 2.2:
SDK de Kinect para Windows, conceptos y características principales.

SDK de Kinect para Windows	
¿Qué es un SDK?	Es un Kit de Desarrollo de Software, está formado por librerías, herramientas etc. Orientado a al desarrollo de aplicaciones para un sistema o hardware específicos.
SDK de Kinect Para Windows	Para el desarrollo de aplicaciones utilizando el sensor Kinect, está disponible el SDK oficial de Microsoft, también los SDK de código abierto.
SDK de Kinect Oficial	

<table>
<tr><td rowspan="3">Características</td><td>Sistemas Operativos Compatibles</td><td>Lenguajes de Programación soportados</td></tr>
<tr><td>Windows 7
Windows 8</td><td>C++
C#
Visual Basic</td></tr>
<tr><td colspan="2">Soporte continuo, drivers oficiales que garantizan una comunicación estable entre el sensor Kinect y el computador.</td></tr>
<tr><td>IDE</td><td colspan="2">El SDK oficial de Kinect tiene integración con Microsoft Visual Studio.Net a partir de la distribución 2010, y con las versiones Exprés y Completa, con el Framework 4.0 o superior.</td></tr>
</table>

Fuente: (Microsoft, s.f.)

2.2.1.3 Plataforma Robótica RoboBuilder

RoboBuilder es una plataforma robótica educativa DIY (constrúyelo tú mismo por sus siglas en ingles). Tiene 16 servo motores, una placa de control, engranes, tornillos, estructura metálica, funciona con una batería 8.4V Ni-MH, se conecta al computador mediante el puerto serial DB9. Todos estos componentes pueden ser ensamblados de 3 maneras distintas: Huno que es un humanoide, Doggy y Dino. Para el proyecto se ensamblara la plataforma robótica RoboBuilder en modo Huno.

2.2.1.4 Huno:

Huno es un robot humanoide que está formado por 16 servomotores, tiene 16 grados de libertad, que otorgan una capacidad media de movimiento hacia adelante, atrás, arriba, abajo. El kit incluye un CD de instalación, con: los drivers y las siguientes aplicaciones: MotionBuilder y RBCUpgradeTool, wCKProgramer. Los fabricantes ponen a disposición de los desarrolladores la librería RobobuilderLib, que facilita la interacción de Visual Studio con el robot para ser programado en los lenguajes: C#, Visual Basic y C++.

Los servomotores están distribuidos de la siguiente manera:

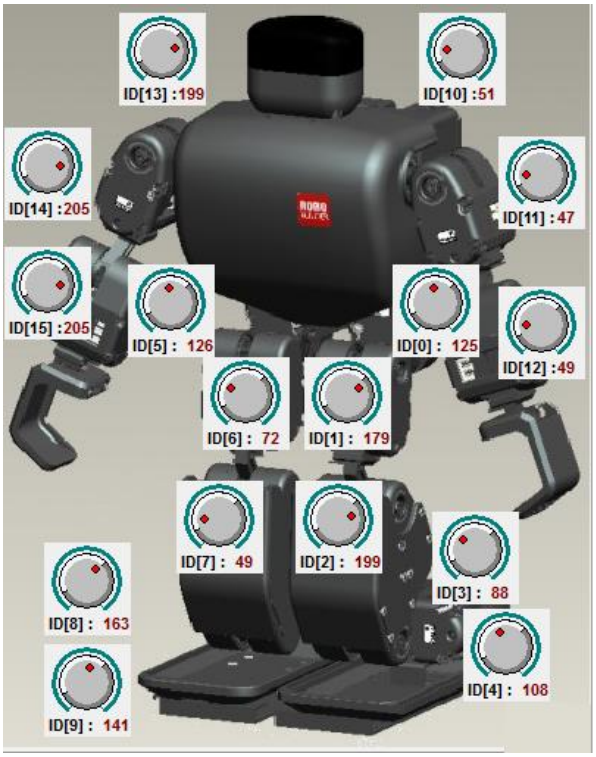


Ilustración 2.4: Distribución Servomotores

Elaborado Por: Autor

Mediante la Interfaz Natural de Usuario y la conexión serial RS-232 establecida con la librería RoboBuilderLib, hay un control total de cada uno de los 16 servos que tienen un ángulo de rotación que va desde 0 hasta 250 grados, cada uno de los servos está identificado mediante un ID y los valores no son iguales en los servos.

2.2.2 Fase 2: diseño rápido del prototipo.

En esta fase se determina como van a interactuar cada uno de los elementos que conforman el sistema. El sistema está formado por los siguientes elementos que interactúan entre sí (véase ***Ilustración 2.5***): Un usuario, un sensor Kinect, un computador donde se encuentra la aplicación que corresponde a la Interfaz Natural de Usuario, y la plataforma robótica educativa Kinect.

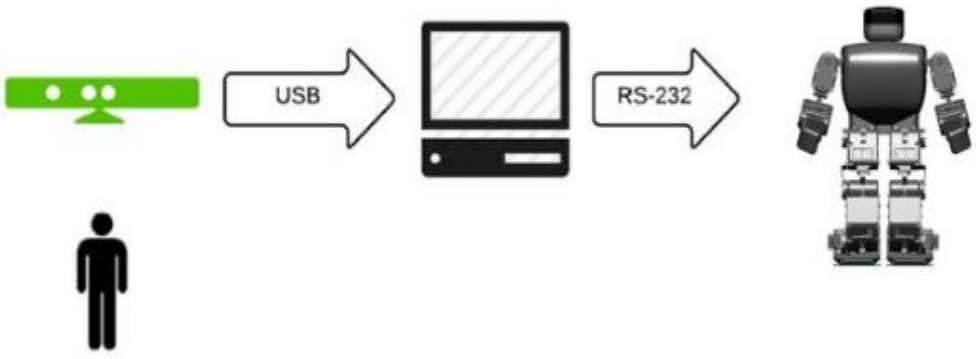


Ilustración 2.5: Representación Gráfica del Sistema

Elaborado Por autor

La idea general del sistema es: capturar los movimientos del usuario por medio del sensor Kinect, procesarlos en la aplicación y enviarlos a la plataforma robótica, (véase ***Ilustración 2.6Ilustración 2.6:*** *Posición Inicial de RoboBuilder*), el análisis del proceso de interacción se basa en un 90% en el código fuente de la aplicación donde el SDK de Kinect es un componente esencial así como la librería RobobuilderLib.

La librería RobobuilderLib desarrollada por el fabricante de la plataforma robótica educativa, hace posible la conexión entre el computador y el robot por medio del puerto serial RS32 y ubica a los servomotores en la posición inicial y los mueve de acuerdo a los valores asignados.

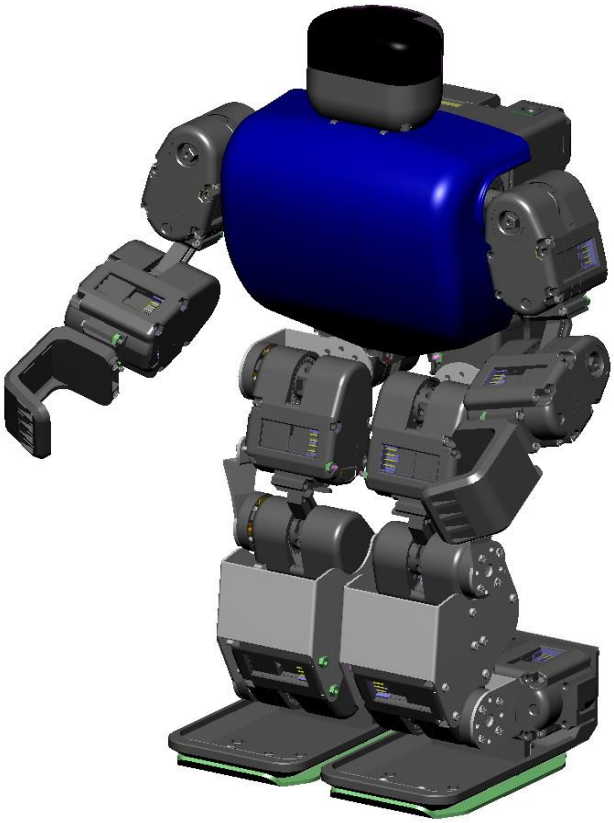


Ilustración 2.6: Posición Inicial de RoboBuilder

Elaborado por: Autor

Para el desarrollo del prototipo el SDK de Kinect proporciona métodos que permiten comunicarse con el sensor, estos métodos van desde la conexión e inicialización del sensor, hasta la captura de datos como: imágenes a color, de profundidad y coordenadas de las articulaciones del usuario.

Luego de realizar cálculos matemáticos, se determina la magnitud de los vectores que representan a los huesos de las extremidades superiores, y los ángulos que estos vectores forman, estos valores angulares se transforman en valores equivalentes determinados para cada uno de los servomotores de las extremidades superiores del robot.

2.2.3 Fase 3: Codificación del prototipo

Luego de un análisis general de cada uno de los elementos de Hardware y Software del sistema, el siguiente paso es integrarlos.

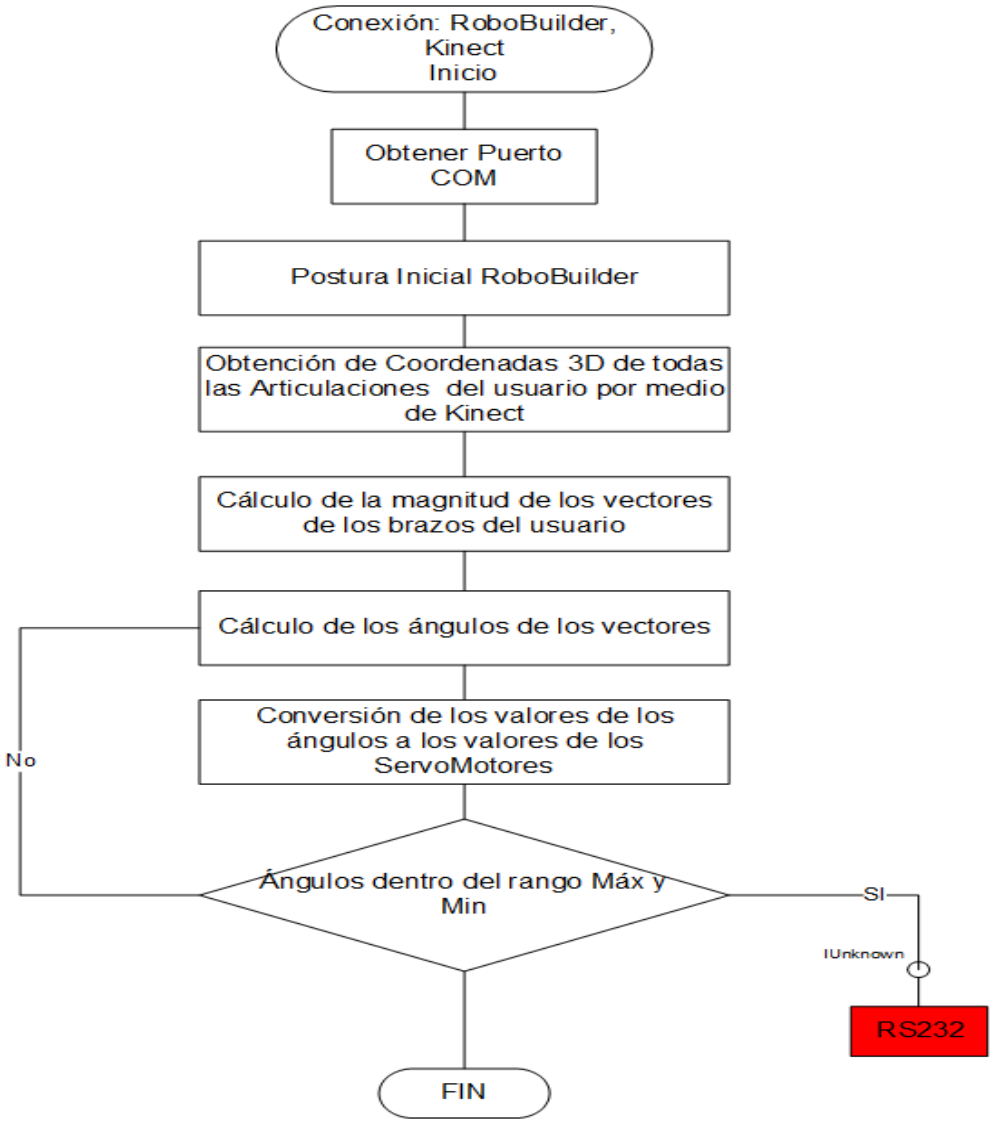


Ilustración 2.7: Diagrama de Flujo Aplicación

Elaborado por: Autor

El proyecto será un WPF (Windows Presentation Fundation), la interfaz de usuario está definida los siguientes elementos: “Image” que permite visualizar un stream de video en formato RGB para comprobar que el sensor está capturando los movimientos del usuario, el elemento “kinect:SkeletonPainter3D” propio de la librería TCD.Kinect, donde se visualiza el avatar en un ambiente 3D, este representación permite comprobar que el sensor de profundidad está capturando correctamente las coordenadas de las

articulaciones del usuario. En la parte superior de la ventana principal consta un link a un ventana secundaria donde esta información referente al uso de la aplicación (véase ***Ilustración 2.8***).

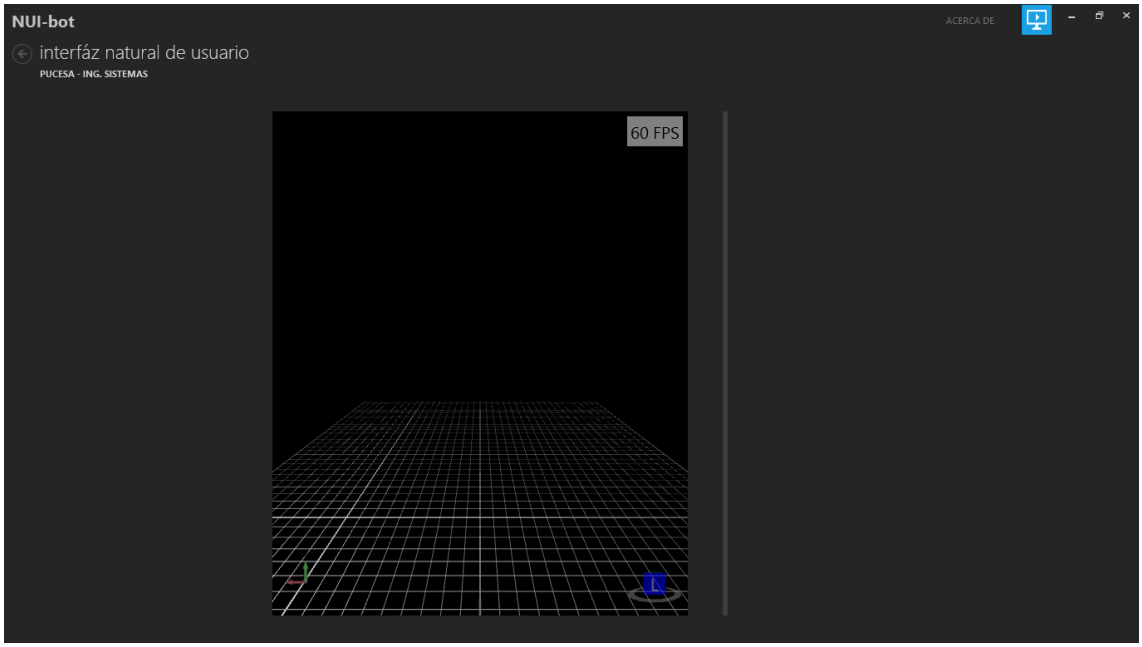


Ilustración 2.8: Ventana Principal de la Aplicación

Elaborado por: Autor

Al proyecto se referenciara los archivos .DLL: Microsoft.Kinect RobobuilderLib y TCD: Kinect, estas referencias permiten utilizar las librería de Kinect y Robobuilder en el proyecto, y acceder a cada uno de los métodos (véase ***Ilustración 2.9***).

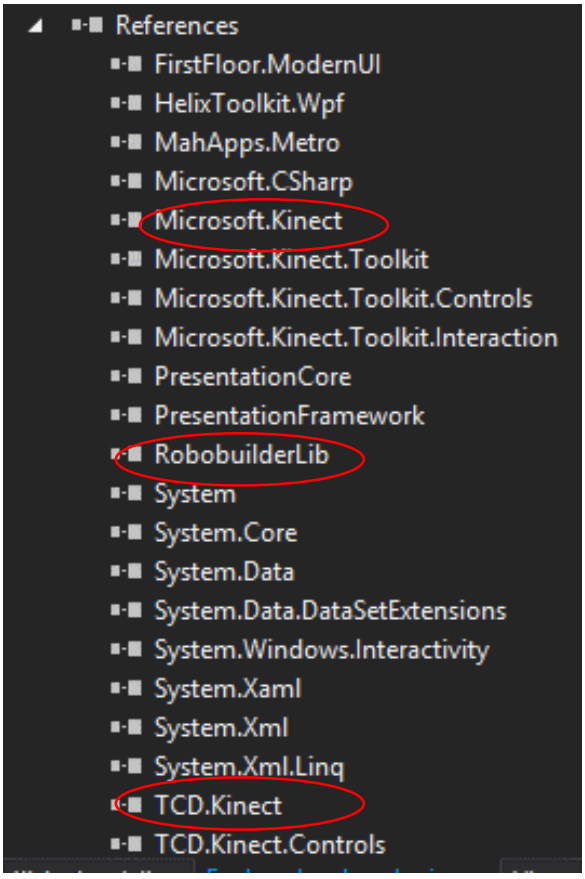


Ilustración 2.9: Referencias Agregadas

Elaborado por: Autor

En el código C# de la aplicación, se utilizan las librerías (véase ***Ilustración 2.10***).

```
19
20  using RobobuilderLib;
21  using Microsoft.Kinect;
22  using Microsoft.Kinect.Toolkit;
23  using Microsoft.Kinect.Toolkit.Controls;
24
25
```

Ilustración 2.10: Librerías Utilizadas en el Proyecto

Elaborado por: Autor

La librería TCD.Kinect, esta referencia en el código XAML de la aplicación (véase ***Ilustración 2.11***), la cual es un complemento al proyecto instalado mediante La Consola del Administrador de paquetes de Visual Studio, que permite instalar paquetes NuGet. Este paquete permite visualizar en un ambiente 3D el avatar del usuario en forma de esqueleto mediante la captura de datos de profundidad.

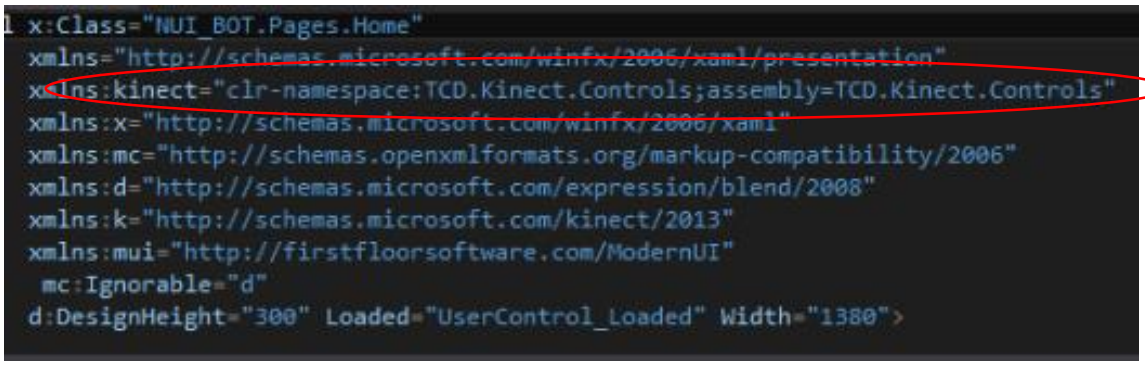


Ilustración 2.11: Librería TCD.Kinect utilizada en el código XAML de la Aplicación.

Elaborado por: Autor

2.2.3.1 Paquetes NuGet

NuGet es una extensión de Visual Studio, está presente en las distribuciones de pago y en las distribuciones Express desde la versión 2010, esta extensión permite instalar librerías y herramientas de desarrollo, que pueden ser actualizadas constantemente.

Para que la representación gráfica del esqueleto del usuario en un ambiente 3D es gracias a la librería TCD.Kinect disponible en Nuget, esta librería permite optimizar el código que dibuja las líneas que representan a las articulaciones del usuario. La

instalación se la realiza mediante la consola de instalación con el comando: “Install-Package TCD.Kinect” (Sonatype, 2013).

2.2.3.2 Inicialización del sensor Kinect:

El SDK de Kinect permite utilizar los 3 tipos de cámaras que posee el sensor, para la obtención de las coordenadas de las articulaciones del usuario, con la Librería Microsoft.Kinect podemos acceder a: las clases, propiedades y métodos que permitirán procesar cada uno de los datos obtenidos por medio del sensor.

Para interactuar con el sensor la clase KinectSensor tiene una instancia, que permite la conexión con las funciones de Kinect, en la clase MainWindow, se declara una variable que se iniciara en el evento Load (véase *Tabla 2.3*).

Tabla 2.3:
Declaración de Variable miKinect

Código Fuente

public partial class MainWindow : Window
{
KinectSensor miKinect;
public MainWindow();
{
InitizlizeComponent();
}

Fuente: *(Microsoft, s.f.)*

El evento Load se ejecutará al iniciar el proyecto, aquí se almacena en la variable miKinect (véase *Tabla 2.3*). un objeto de la clase KinectSensor.KinectSensors[0], este objeto permite llamar a las diferentes funciones del sensor, como las cámaras, micrófonos o acelerómetro.

El array llama al elemento 0, que representa al Kinect que se encuentra conectado al computador, esto permite trabajar con más de un sensor Kinect. El sensor Kinect inicia con el siguiente código: miKinect.Start(), (Microsoft, s.f.).

2.2.3.3 Cámara RGB

El control Image situado en el formulario permite visualizar los datos capturados por la cámara RBG, mediante la propiedad ColorStream.Enable la cámara RGB del sensor Kinect se habilita, y captura un flujo de datos, con una resolución de 640 x 480 pixeles y a una velocidad de 30 frames por segundo (FPS).

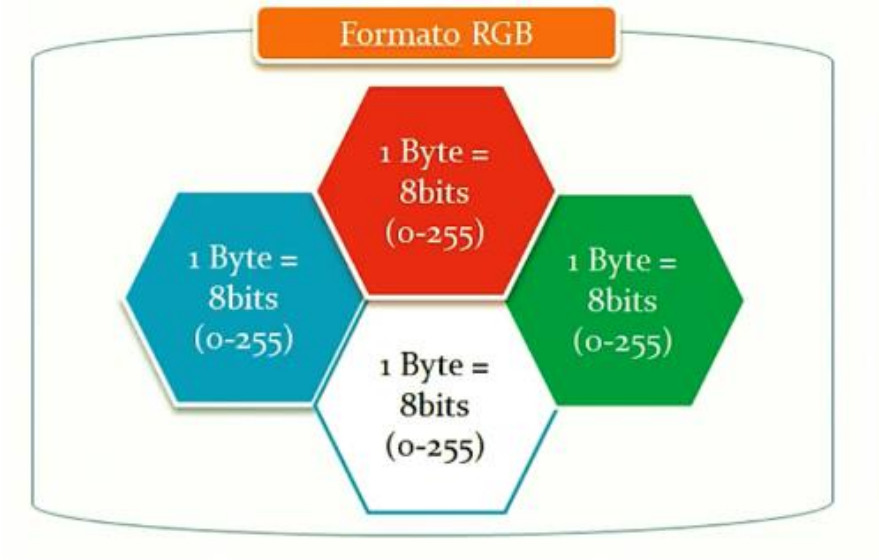


Ilustración 2.12: Representación de un pixel formato RGB

Elaborado por: Autor

Un pixel de formato RGB está formado por 32 bits o 4 bytes, (véase ***Ilustración 2.12***), donde están representados los colores: azul, rojo, verde y el canal alfa o capa de transparencia, cada uno de estos bytes toman valores entre 0 y 255, dando las diferentes tonalidades a las imágenes.



Ilustración 2.13: Cámara RGB

Fuente: (Microsoft, s.f.)

Mediante un controlador de eventos (EventHandler), captura el flujo de datos y lo almacena en la propiedad "e" que es creada por defecto, el nombre de esta propiedad puede ser cambiado, los pixeles se almacenan en un arreglo para ser manipulados dentro del código fuente.

Tabla 2.4:
Código Fuente para activar cámara RGB de Kinect

Código Fuente
Abrir Flujo de Datos
kinect.VideoStream.Open(ImageStreamType.Video, 2, ImageResolution.Resolution640x480, ImageType.Color);
EventHandler:

```
kinect.VideoFrameReady += new
EventHandler<ImageFrameReadyEventArgs>(kinect_VideoFrameReady);
```

Método kinect_VideoFrameReady

```
void kinect_VideoFrameReady(object sender, ImageFrameReadyEventArgs e)
{
PlanarImage Image = e.ImageFrame.Image;
image1.Source = BitmapSource.Create(Image.Width, Image.Height, 96, 96,
PixelFormats.Bgr32, null, Image.Bits, Image.Width * Image.BytesPerPixel);
}
```

Fuente: (Jana, 2012)

2.2.3.4 Cámara de Profundidad

La cámara de profundidad permite detectar la distancia de los objetos con respecto al sensor (véase ***Ilustración 2.14***), estos objetos deben estar a una distancia máxima de cuatro metros y una distancia mínima de ochenta centímetros.



Ilustración 2.14: Representación de los puntos infrarrojos

Fuente: *(Microsoft, s.f.)*

El proyector infrarrojo emite millones de puntos hacia los objetos que están frente al sensor, estos puntos no son perceptibles para el ojo humano, mediante la cámara infrarroja, los puntos son capturados y así se mide la distancia de los objetos. La resolución de la cámara infrarroja es de 640 x 480 pixeles a 30 FPS (frames por segundo).

2.2.3.5 Detección de Esqueletos

La NUI (Interfaz Natural de Usuario por sus siglas en inglés), permite interactuar con aplicaciones por medio de movimientos del cuerpo. Para lograr esto, Kinect, por medio de su SDK, permite procesar un flujo de datos llamado Skeleton Stream. En una combinación de las cámaras RGB y de profundidad el flujo de imágenes representarán a un esqueleto humano (véase ***Ilustración 2.15***).

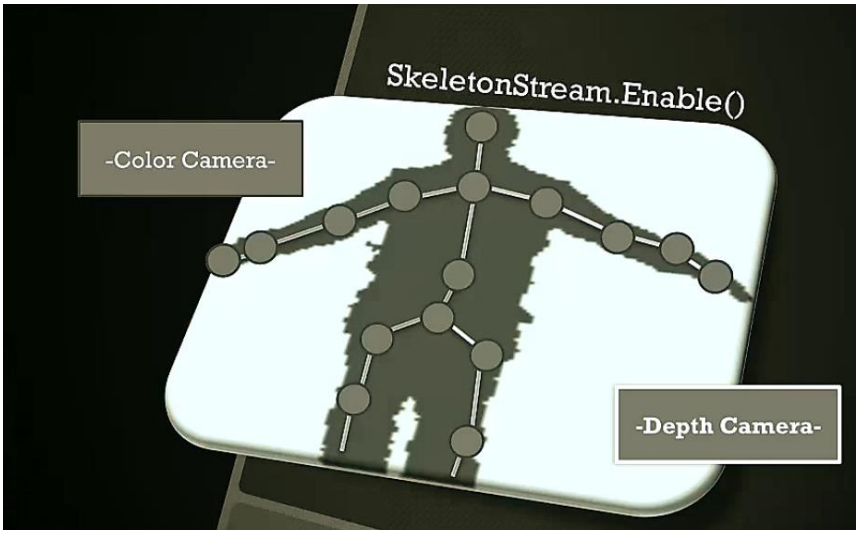


Ilustración 2.15: Esqueletos 3D

Elaborado por: Autor.

La discriminación que el sensor hace es dependiendo del tamaño de los objetos, así los mas grande son los más relevantes, y mediante patrones determina si estos objetos representan a una persona, o son muebles. Luego de este proceso el sensor representa un esqueleto digital parecido al humano, además interpreta cada uno de los movimientos que realice el usuario.

El esqueleto humano adulto está formado aproximadamente por 206 huesos, y alrededor de 360 articulaciones, esto representa un gran problema para ser programado por la gran cantidad de datos, el SDK de Kinect simplifica el número de huesos a 19, conectados por 20 articulaciones, siendo para el sensor Kinect las articulaciones las más relevantes (véase ***Ilustración 2.16:*** *Esqueleto Digital con el nombre de las Articulaciones*).

Tabla 2.5:
Código Fuente para activar cámara de Profundidad de Kinect

Código Fuente
Abrir flujo de Datos:
kinect.DepthStream.Open(ImageStreamType.Depth, 2, ImageResolution.Resolution320x240, ImageType.Depth);
EventHandler:
kinect.DepthFrameReady += new EventHandler<ImageFrameReadyEventArgs>(kinect_DepthFrameReady);
Método kinect_VideoFrameReady

void kinect_DepthFrameReady(object sender, ImageFrameReadyEventArgs e)
{
PlanarImage Image = e.ImageFrame.Image;
byte[] convertedFrame = convertDepthFrame(Image.Bits);
image2.Source = BitmapSource.Create(Image.Width, Image.Height, 96, 96,
PixelFormats.Bgr32, null, convertedFrame, Image.Width * 4);
}

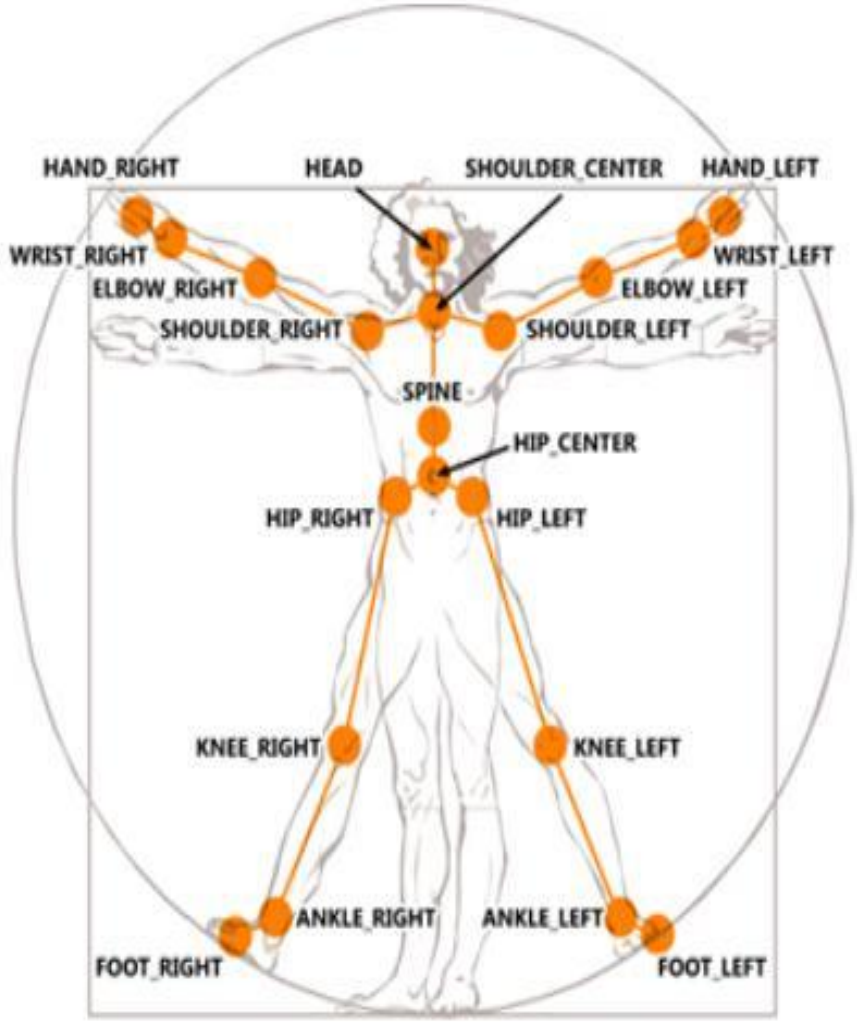


Ilustración 2.16: Esqueleto Digital con el nombre de las Articulaciones

Fuente: *(Microsoft, 2012)*

Cada una de estas articulaciones tiene una ubicación (véase ***Ilustración 2.16***), la representación de esta ubicación es un grupo de coordenadas en 3 dimensiones (véase ***Ilustración 2.17***). Estos datos al ser capturados son utilizados para realizar diferentes tipos de cálculos como: longitud entre articulaciones, ángulos corporales etc.

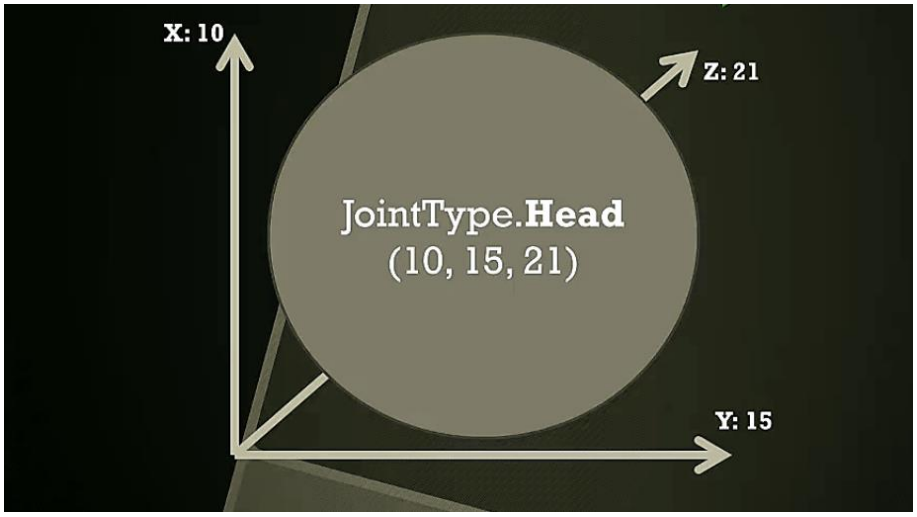


Ilustración 2.17: Representación gráfica de las coordenadas de una articulación

Elaborado por: Autor

Los pasos para capturar y procesar estos datos son:

1. Habilitar el flujo de datos de esqueletos.
2. Capturar este flujo de datos.
3. Crear un arreglo que almacenará los datos del esqueleto.
4. Copiar los datos de ese esqueleto en el arreglo de esqueletos.
5. Recorrer cada uno de los frames de esqueletos con un bucle.
6. Verificar que se detecte correctamente un esqueleto.
7. Verificar que tenga una buena calidad de captura.
8. Elegir las articulaciones de este esqueleto que se van a utilizar.
9. Utilizar alguna propiedad de esta articulación (posición).
10. Obtener resultados numéricos de esta captura.

2.2.3.6 Cálculos

Con los datos obtenidos con el sensor Kinect, se forman vectores que van desde el hombro hasta el codo, y desde el codo hasta la mano, estos vectores tridimensionales forman ángulos corporales, mediante cálculos matemáticos se determina el valor de los mismos (véase ***Ilustración 2.18***).

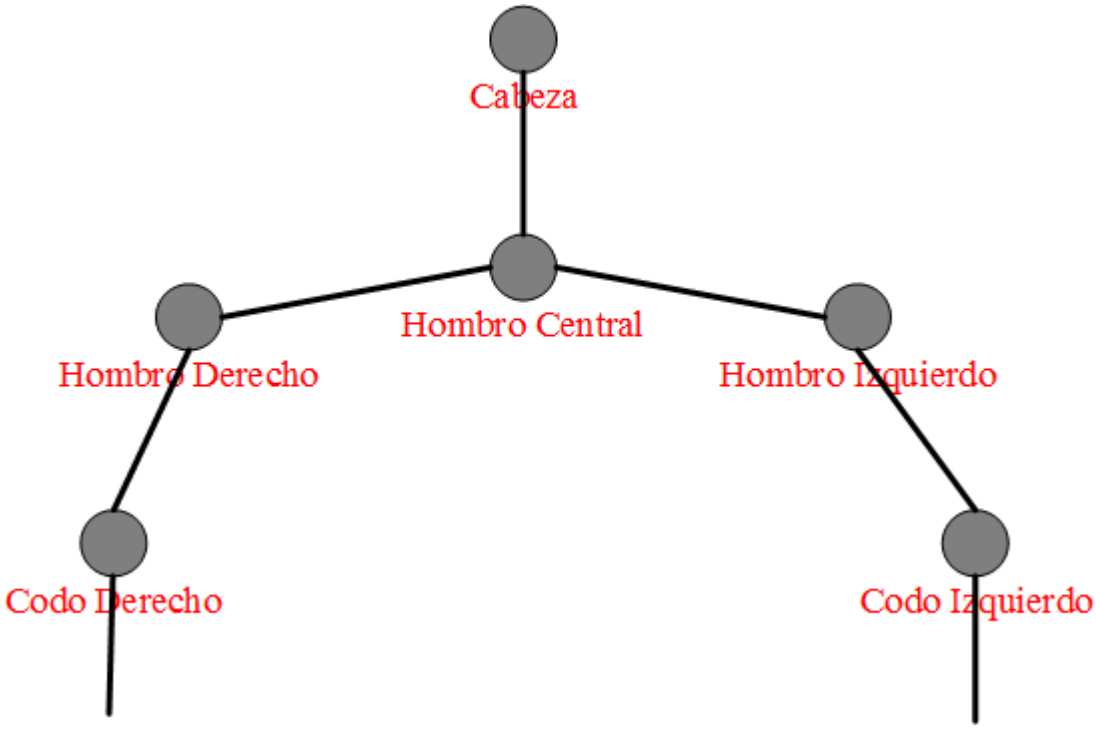


Ilustración 2.18: Representación de los puntos y vectores de las extremidades superiores del usuario

Elaborado por: Autor

2.2.3.7 Ángulos Corporales

Como se muestra en la imagen los ángulos formados por la extremidad superior derecha en los ejes X y Y son: el ángulo Hd y Cd. El ángulo Hd lo forman los vectores entre los puntos Hombro Central y Hombro Derecho y Hombro Derecho con Codo Derecho. El ángulo Cd lo forman los vectores entre los puntos Hombro Derecho (véase ***Ilustración 2.19***).

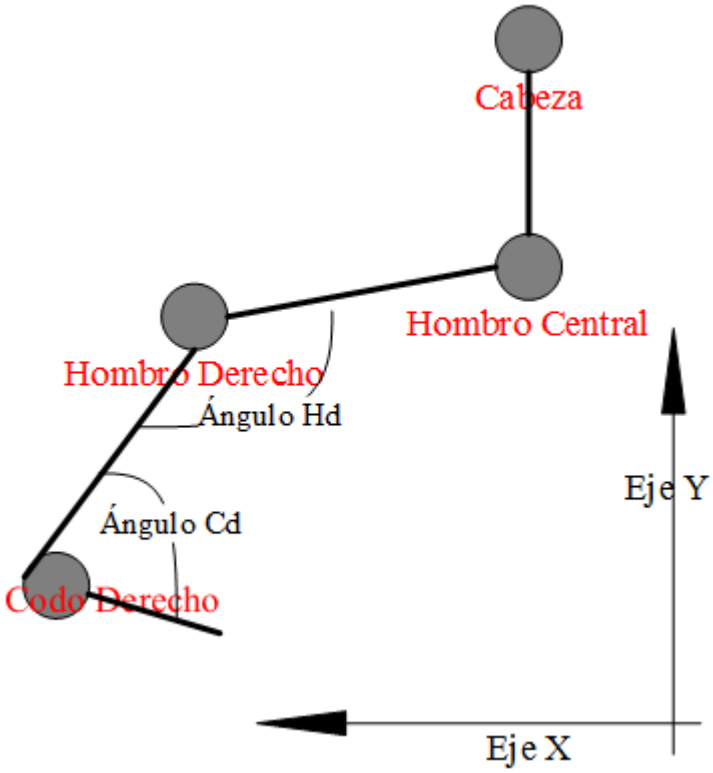


Ilustración 2.19: Ángulos y Articulaciones ejes X Y Brazo Derecho

Elaborado por: Autor

En los ejes Y, Z los vectores forman el ángulo, Hdz, como se muestra en la imagen, el ángulo Hdz lo forman los vectores (véase ***Ilustración 2.20***).

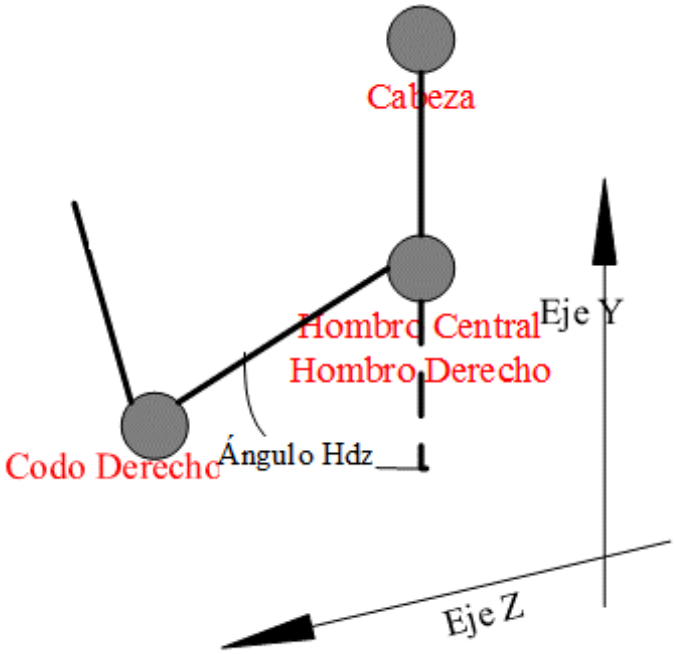


Ilustración 2.20: Ángulos y articulaciones ejes Y Z Brazo Derecho

Elaborado por: Autor

En la extremidad superior izquierda los ángulos se forman igual que en la extremidad superior derecha como se ve en la imagen, el ángulo Hi, formado por los vectores que se encuentran entre los puntos: Hombro central y hombro Izquierdo, y Hombro Izquierdo con Codo Izquierdo (véase ***Ilustración 2.21***).

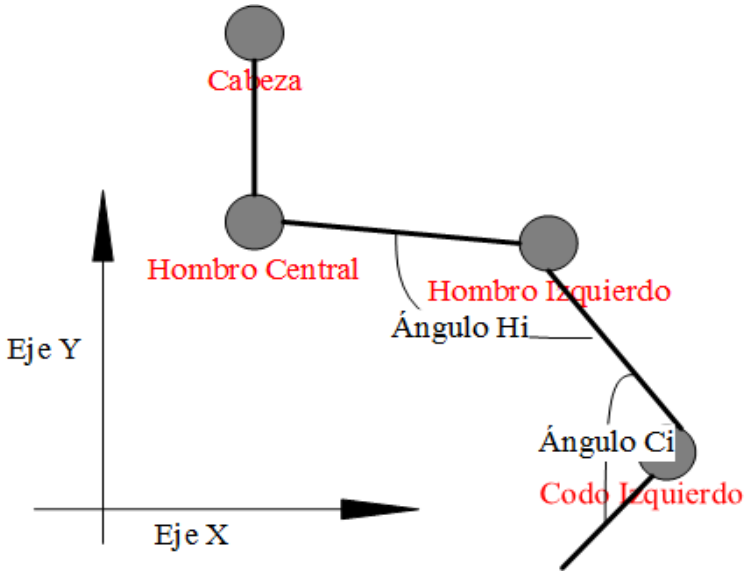


Ilustración 2.21: Ángulos y Articulaciones ejes X Y Brazo Izquierdo

Elaborado por: Autor

En los ejes Y, Z el ángulo formado es: Hiz, formado por los vectores entre los puntos Hombro Izquierdo y Central con el tronco del usuario, como se muestra en la imagen (véase ***Ilustración 2.22***).

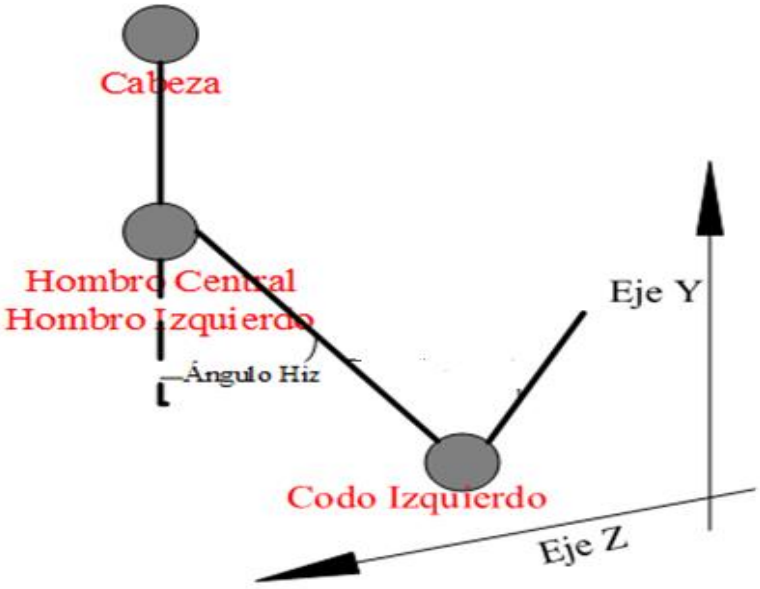


Ilustración 2.22: Ángulos y articulaciones ejes Y Z Brazo Derecho

Fuente: Autor

2.2.3.8 Operaciones con los Vectores

Con los vectores representados gráficamente las operaciones que se van a realizar para conocer el valor de los ángulos son las siguiente: Producto Punto o Escalar, Cálculos de los ángulos conociendo las magnitudes de los vectores.

Como muestran las imágenes los vectores tendrán coordenadas cartesianas en los ejes X, Y, Z, estos valores son capturados por Kinect.

El producto Escalar o Producto Punto tiene la siguiente fórmula para vectores con coordenadas cartesianas:

$$\vec{A} \cdot \vec{B} = \overrightarrow{A_x}.\overrightarrow{B_x} + \overrightarrow{A_y}.\overrightarrow{B_y} + \overrightarrow{A_z}.\overrightarrow{B_z}$$

La magnitud o módulo del vector formado entre los puntos Hombro – Codo y del vector formado por los puntos Codo - Mano está dado por la siguiente formula:

$$\overrightarrow{V_1} = \sqrt{\overrightarrow{A_x}^2 + \overrightarrow{A_y}^2 + \overrightarrow{A_z}^2}$$

$$\overrightarrow{V_2} = \sqrt{\overrightarrow{B_x}^2 + \overrightarrow{B_y}^2 + \overrightarrow{B_z}^2}$$

La fórmula para el cálculo del ángulo del codo es:

$$\text{ACodo} = \cos^{-1} \frac{\vec{A} \cdot \vec{B}}{\overrightarrow{V_1} * \overrightarrow{V_2}}$$

Este valor debe ser convertido de radianes a grados con la siguiente fórmula:

$$GradCodo = 180 - \frac{\text{ACodo} * 180}{\pi}$$

Los ángulos del Hombro Derecho en los ejes X, Y están determinados por la siguiente fórmula:

$$AHombroXY = \cos^{-1} \frac{\overrightarrow{-A_x}}{\overrightarrow{V_1}}$$

Los ángulos del Hombro Izquierdo en los ejes X, Y están determinados por la siguiente fórmula:

$$AHombroXY = \cos^{-1} \frac{\overrightarrow{A_x}}{\overrightarrow{V_1}}$$

La conversión de radianes a grados usa la siguiente fórmula:

$$AGrados = \frac{AHombroXY * 180}{\pi}$$

Para calcular el ángulo formado en el eje Z es calculado por la siguiente formula:

$$AHombroZ = \cos^{-1} \frac{\overrightarrow{A_z}}{\overrightarrow{V_1}}$$

El valor obtenido debe ser convertido de radianes a grados con la siguiente fórmula:

$$AGrados = \frac{AHombroZ * 180}{\pi}$$

2.2.3.9 Conversión de ángulos

Los ángulos obtenidos deben ser transformados a los rangos de cada servo motor, estos valores deben ser enteros, en la tabla se podrá apreciar las fórmulas matemáticas para la transformación de los valores.

Tabla 2.6:

Cálculos ServoMotores

<table>
<tr><th rowspan="2">ID Motor</th><th rowspan="2">Nombre Motor</th><th colspan="2">Valores</th><th rowspan="2">Fórmula para Conversión</th></tr>
<tr><th>Valor Máximo</th><th>Value</th></tr>
<tr><td>10</td><td>Hombro Izquierdo Z</td><td>270</td><td>0.89</td><td>$LSZP = ((Vmax - ángule) * value) + 10$</td></tr>
<tr><td>11</td><td>Hombro Izquierdo X Y</td><td>-</td><td>-</td><td>Si ángulo mayor a 0
$LSXP = 128 - ángule$
Caso Contrario
$LSXP = 128 + ángule$</td></tr>
<tr><td rowspan="4">12</td><td rowspan="4">Codo Izquierdo</td><td colspan="3">Sí ángulo mayor que 90</td></tr>
<tr><td>180</td><td>0.944443</td><td>$LEP = 125 - ((Vmax - ángule) * value)$</td></tr>
<tr><td colspan="3">Caso contrario</td></tr>
<tr><td></td><td>0.444443</td><td>$LEP = angle * value$</td></tr>
<tr><td>13</td><td>Hombro derecho Z</td><td>-</td><td>0.89</td><td>$RSZP = (angle * value) + 10$</td></tr>
<tr><td>14</td><td>Hombro Derecho XY</td><td>-</td><td>0.855555</td><td>Si ángulo mayor que 0
$RSXP = 128 + (angle * value)$
Caso contrario
$RSXP = 128 - (angle * value)$</td></tr>
<tr><td>15</td><td>Codo Derecho</td><td>180</td><td>0.84444443</td><td>$REP = ((Vmax - angle) * value) + 128$</td></tr>
</table>

CAPÍTULO III.

RESULTADOS

Los resultados obtenidos se presentan con las funcionalidades de las extremidades superiores, en la pantalla principal se encuentra la captura con datos de profundidad y color.

3.1 Fase 1: Movimiento Individual de los Brazos

Para comparar los movimientos entre el flujo de datos de profundidad, de la cámara a colores y los datos enviados a la plataforma robótica RoboBuilder se analizan los movimientos: lateral, arriba, adelante, y abajo, de cada una de las extremidades por separado, y en conjunto

Tabla 3.1:

Código Fuente Cálculo de Ángulos y Vectores

Código Fuente Cálculo de Ángulos y Vectores

```
private void CalculateAngles(Joint Hand, Joint Elbow, Joint Shoulder,  string side)
      {
         double[] Vse = new double[] { 0, 0, 0 };
         double[] Veh = new double[] { 0, 0, 0 };

         //Obtiene el punto en un espacio de coordenadas del Frame
         DepthImagePoint _Hand = GetDephtPoint(Hand);
         DepthImagePoint _Elbow = GetDephtPoint(Elbow);
         DepthImagePoint _Shoulder = GetDephtPoint(Shoulder);



         // Calcula el vector entre el hombro y el codo
         Vse[0] = _Shoulder.X - _Elbow.X;
         Vse[1] = _Shoulder.Y - _Elbow.Y;
         Vse[2] = _Shoulder.Depth - _Elbow.Depth;

         // Calcula el vector entre el codo y la mano
         Veh[0] = _Elbow.X - _Hand.X;
         Veh[1] = _Elbow.Y - _Hand.Y;
         Veh[2] = _Elbow.Depth - _Hand.Depth;

         // Producto escalar entre los dos vectores del el brazo
         double dotProd = (Vse[0] * Veh[0] + Vse[1] * Veh[1] + Vse[2] * Veh[2]);

         // Calcula la magnitud del vector de hombro - Codo
         double SizeVse = Math.Sqrt(Math.Pow(Vse[0], 2) + Math.Pow(Vse[1], 2)
+ Math.Pow(Vse[2], 2));

         // Calcula la magnitud del vector Codo - Mano
         double SizeVeh = Math.Sqrt(Math.Pow(Veh[0], 2) + Math.Pow(Veh[1], 2)
+ Math.Pow(Veh[2], 2));

         //Calcular ángulo Codo
         if (Hand.TrackingState == JointTrackingState.Tracked &&
Elbow.TrackingState == JointTrackingState.Tracked && Shoulder.TrackingState
== JointTrackingState.Tracked)
         {
            double angleRad = Math.Acos(dotProd / (SizeVse * SizeVeh));
            _AngleElbow = 180 - (angleRad * 180 / Math.PI);
            ManageCounts(0, side);
         }
```

```
        else if (Elbow.TrackingState == JointTrackingState.NotTracked &&
Shoulder.TrackingState == JointTrackingState.NotTracked &&
Hand.TrackingState == JointTrackingState.Tracked)
        {
            _AngleElbow = 180;
            ManageCounts(0, side);
        }

        //Calcular angulo Hombro X Y
        if (Elbow.TrackingState == JointTrackingState.Tracked &&
Shoulder.TrackingState == JointTrackingState.Tracked)
        {
            double angleRad = 0;
            double angleDeg = 0;
            if (side == "Right")
            {
                angleRad = Math.Acos((-Vse[0]) / SizeVse);
                angleDeg = angleRad * 180 / Math.PI;
            }
            else
            {
                angleRad = Math.Acos(Vse[0] / SizeVse);
                angleDeg = angleRad * 180 / Math.PI;
            }

            if (_Elbow.Y >= _Shoulder.Y)
            {
                _AngleShoulderY = -angleDeg;
            }
            else
            {
                _AngleShoulderY = angleDeg;
            }

            //Excepción
            if (side == "Right")
            {
                if (_Elbow.X <= _Shoulder.X)
                {
                    _AngleShoulderY = 90;
                }
            }
            else if (_Elbow.X >= _Shoulder.X)
            {
                _AngleShoulderY = 90;
            }
            ManageCounts(1, side);
        }
```

```
        else if (Elbow.TrackingState == JointTrackingState.NotTracked &&
Shoulder.TrackingState == JointTrackingState.NotTracked &&
Hand.TrackingState == JointTrackingState.Tracked)
        {
            _AngleShoulderY = 90;
            ManageCounts(1, side);
        }

        //Calcular ángulo Hombro Z
        if (Elbow.TrackingState == JointTrackingState.Tracked &&
Shoulder.TrackingState == JointTrackingState.Tracked)
        {
            double angleRad = Math.Acos(Vse[2] / SizeVse);
            double angleDeg = angleRad * 180 / Math.PI;

            if (_Elbow.Y <= _Shoulder.Y)
            {
                _AngleShoulderZ = 90 - angleDeg;
            }
            else
            {
                _AngleShoulderZ = angleDeg + 90;
            }
            //Excepción
            if (_AngleShoulderY < 30 && _AngleShoulderY > -30)
            {
                angleRad = Math.Acos(Veh[2] / SizeVeh);
                angleDeg = angleRad * 180 / Math.PI;
                if (_Hand.Y <= _Elbow.Y)
                {
                    _AngleShoulderZ = 90 + (90 - angleDeg);
                }
                else
                {
                    _AngleShoulderZ = 180 + angleDeg;
                }

            }
            ManageCounts(2, side);
        }
        //Más excepciones
        else if (Elbow.TrackingState == JointTrackingState.NotTracked &&
Shoulder.TrackingState == JointTrackingState.NotTracked &&
Hand.TrackingState == JointTrackingState.Tracked)
        {
            _AngleShoulderZ = 90;
            ManageCounts(2, side);
        }
```

```
    }
```

Elaborado por: Autor.

3.1.1 Movimiento lateral del brazo derecho

El brazo derecho tiene la funcionalidad de imitar los movimientos del usuario, teniendo en cuenta que el robot tiene 16 grados de libertad estos movimientos no son exactos.

Tabla 3.2: Código Fuente Movimientos Brazo Derecho Plataforma Robótica

```
    /// <summary>
     /// ID15
     /// </summary>
     /// <param name="angle"> </param>

private void REA_Send(int angle)
     {
        //RightElbow conversion
        int Vmax = 180;
        double value = 0.84444443;
        double REP = ((Vmax - angle) * value) + 128;
        int aux = (int)REP;
        w.wckMovePos(15, aux, 1);
      }

     /// <summary>
     /// ID14
     /// </summary>
     /// <param name="angle"> </param>
     private void RSXA_Send(int angle)
     {
        //RightShoulderY conversion
        double value = 0.855555;
        double RSXP = 0;

        if (angle > 0)
        {
           RSXP = 128 + (angle * value);
        }
        else
```

```
        {
            RSXP = 128 - (angle * value);
        }

        int aux = (int)RSXP;
        w.wckMovePos(14, aux, 1);
    }

    /// <summary>
    /// ID13
    /// </summary>
    /// <param name="angle"> </param>
    private void RSZA_Send(int angle)
    {
        double value = 0.89;
        double RSZP = (angle * value) + 10;
        int aux = (int)RSZP;
        w.wckMovePos(13, aux, 1);
    }
```

Elaborado por: Autor.

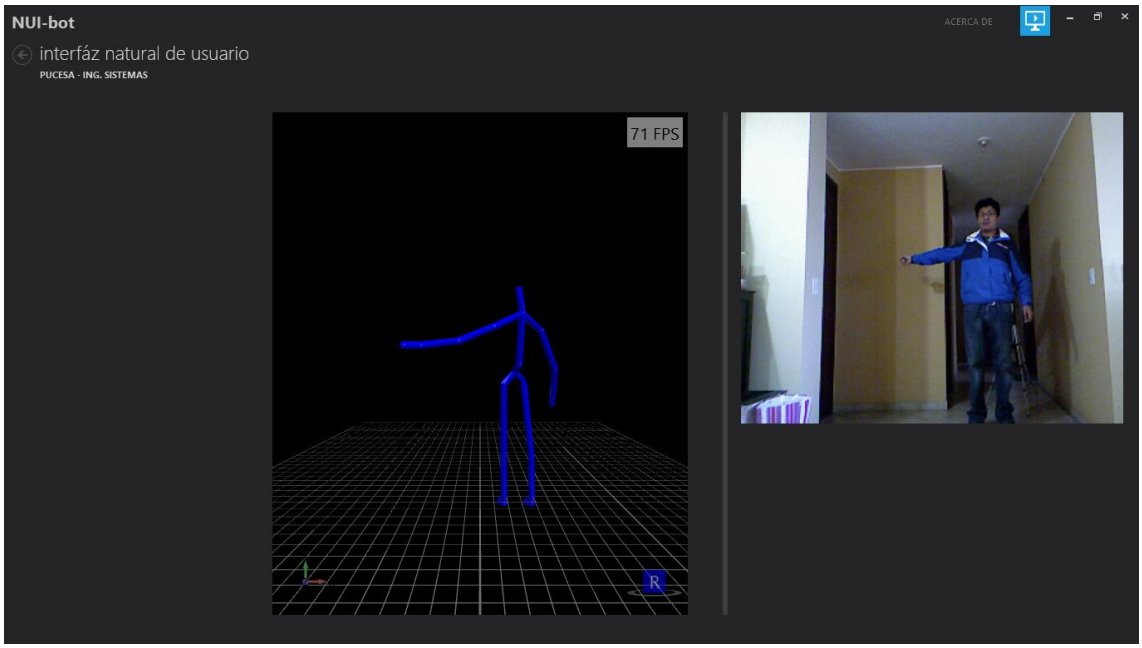


Ilustración 3.1: Movimiento lateral brazo derecho

Elaborado por: Autor.



Ilustración 3.2: Movimiento lateral del brazo derecho plataforma robótica educativa

Elaborado por: Autor

3.1.2 Movimiento hacia adelante del brazo derecho:

Se muestra los datos capturados de profundidad y color del brazo derecho del usuario representado en la aplicación, además se muestra el movimiento del brazo derecho hacia delante de la plataforma robótica educativa RoboBuilder.

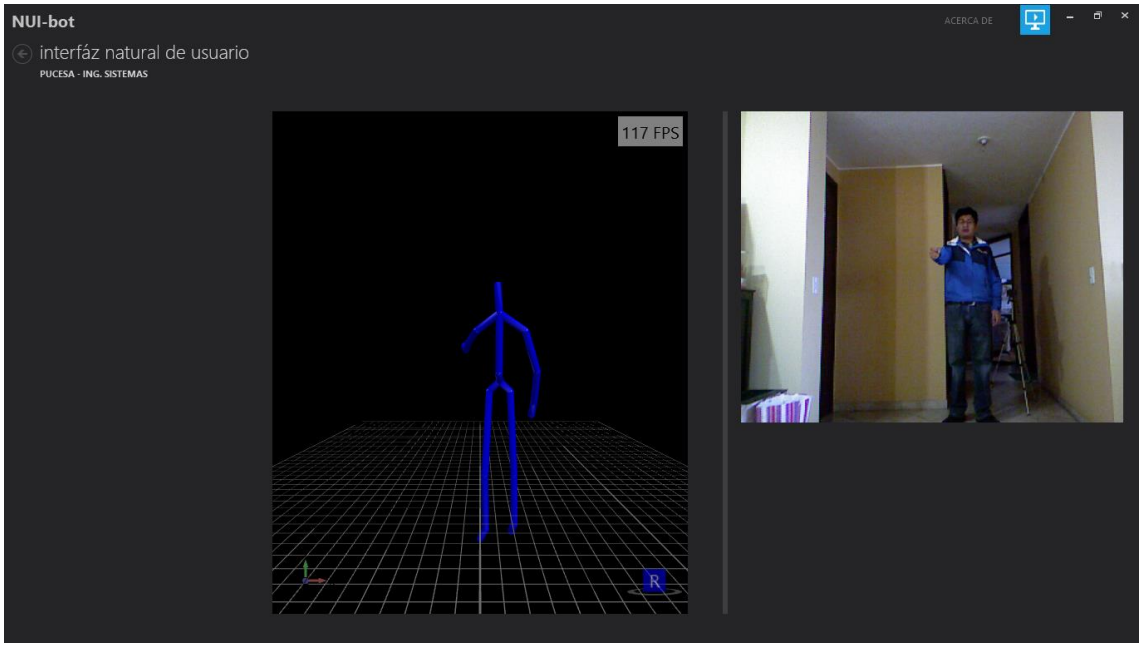


Ilustración 3.3: Movimiento hacia adelante brazo derecho

Elaborado por: Autor.



Ilustración 3.4: Movimiento hacia adelante del brazo derecho plataforma robótica educativa

Elaborado por: Autor

3.1.3 Movimiento hacia arriba de la extremidad derecha

Se muestra el movimiento hacia arriba del brazo derecho, se muestra los datos de profundidad y color en la aplicación, así como también el movimiento de la extremidad de la plataforma robótica.

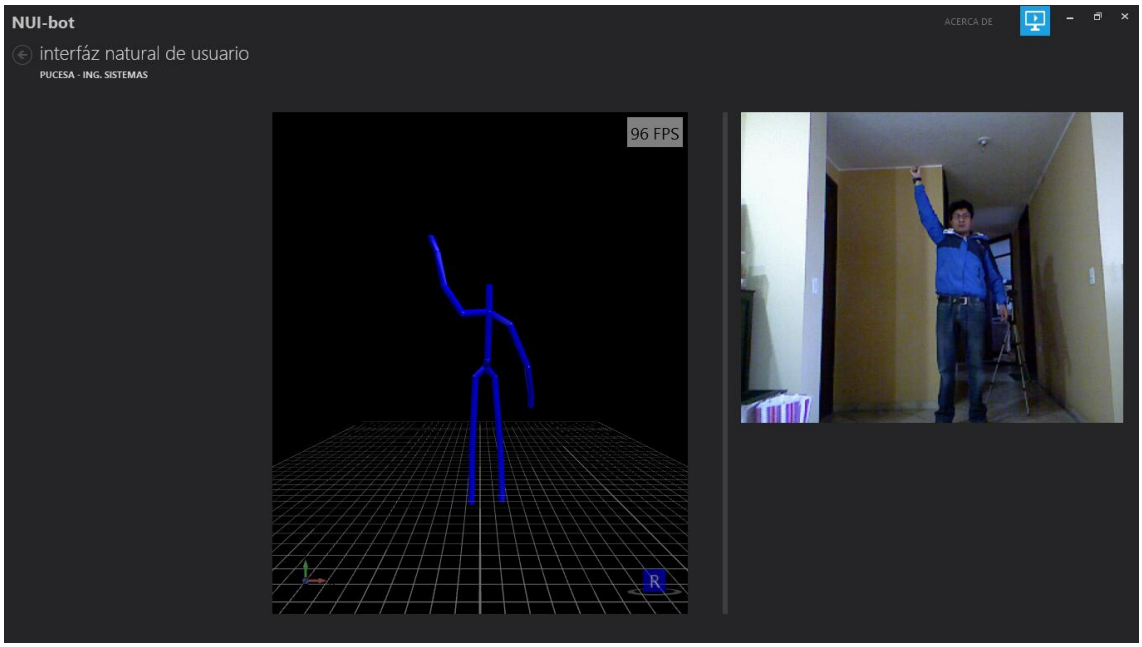


Ilustración 3.5: Movimiento hacia arriba del brazo derecho del usuario.

Elaborado por: Autor



Ilustración 3.6: Movimiento hacia arriba del brazo derecho de la plataforma robótica.

Elaborado por: Autor

3.1.4 Movimiento lateral del brazo izquierdo

El brazo izquierdo al igual que el brazo derecho tiene la funcionalidad de imitar los movimientos del usuario, teniendo en cuenta que el robot tiene 16 grados de libertad estos movimientos no son exactos.

Tabla 3.3: Código Fuente Movimientos Brazo Izquierdo Plataforma Robótica

```
//// <summary>
/// ID12
/// </summary>
/// <param name="angle"> </param>
private void LEA_Send(int angle)
{
    //LeftElbow conversion
    int Vmax = 0;
    double value = 0;
    double LEP = 0;
    if (angle > 90)
    {
        value = 0.944443; Vmax = 180;
        LEP = 125 - ((Vmax - angle) * value);
    }
    else
    {
        value = 0.444443;
        LEP = angle * value;
    }
    int aux = (int)LEP;
    w.wckMovePos(12, aux, 1);
}

/// <summary>
/// ID11
/// </summary>
/// <param name="angle"></param>
private void LSXA_Send(int angle)
{
    //LeftShoulderX conversion
    double LSXP;

    if (angle >= 0)
    {
        LSXP = 128 - angle;
    }
    else
    {
        LSXP = 128 + angle;
    }

    int aux = (int)LSXP;
    w.wckMovePos(11, aux, 1);

}

/// <summary>
```

```
/// ID10
/// </summary>
/// <param name="angle"> </param>
private void LSZA_Send(int angle)
{
   //LeftShoulderZ conversion
   int Vmax = 270;
   double value = 0.89;
   double LSZP = ((Vmax - angle) * value) + 10;
   int aux = (int)LSZP;
   w.wckMovePos(10, aux, 1);
}
```

Elaborado por: Autor.

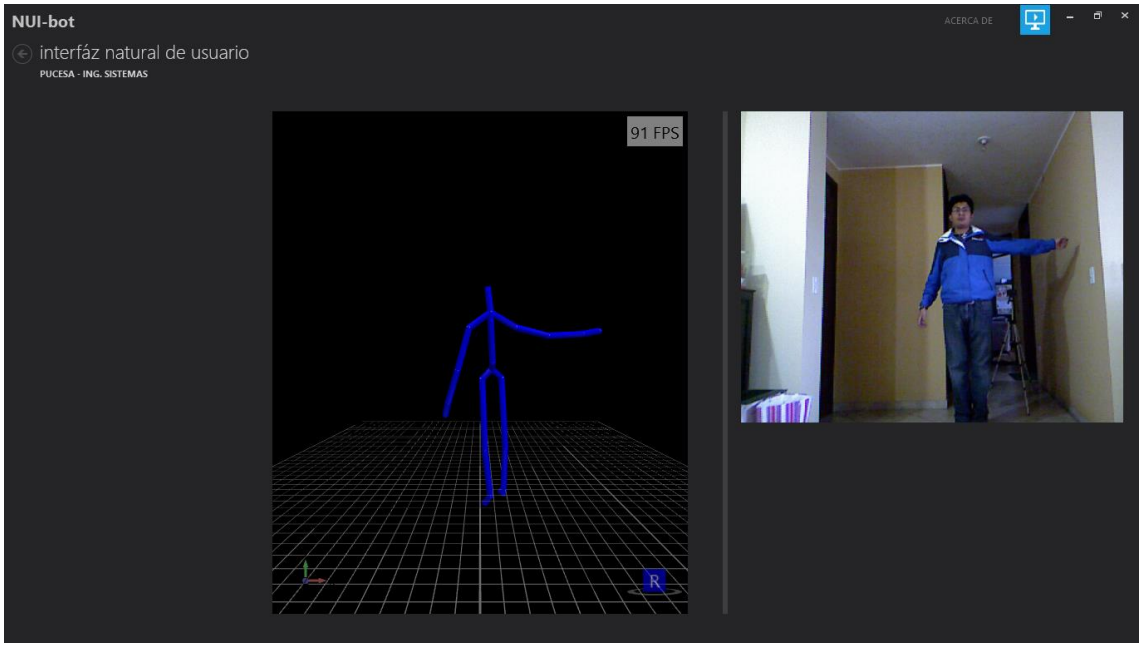


Ilustración 3.7: Movimiento lateral brazo izquierdo del usuario

Elaborado por: Autor.

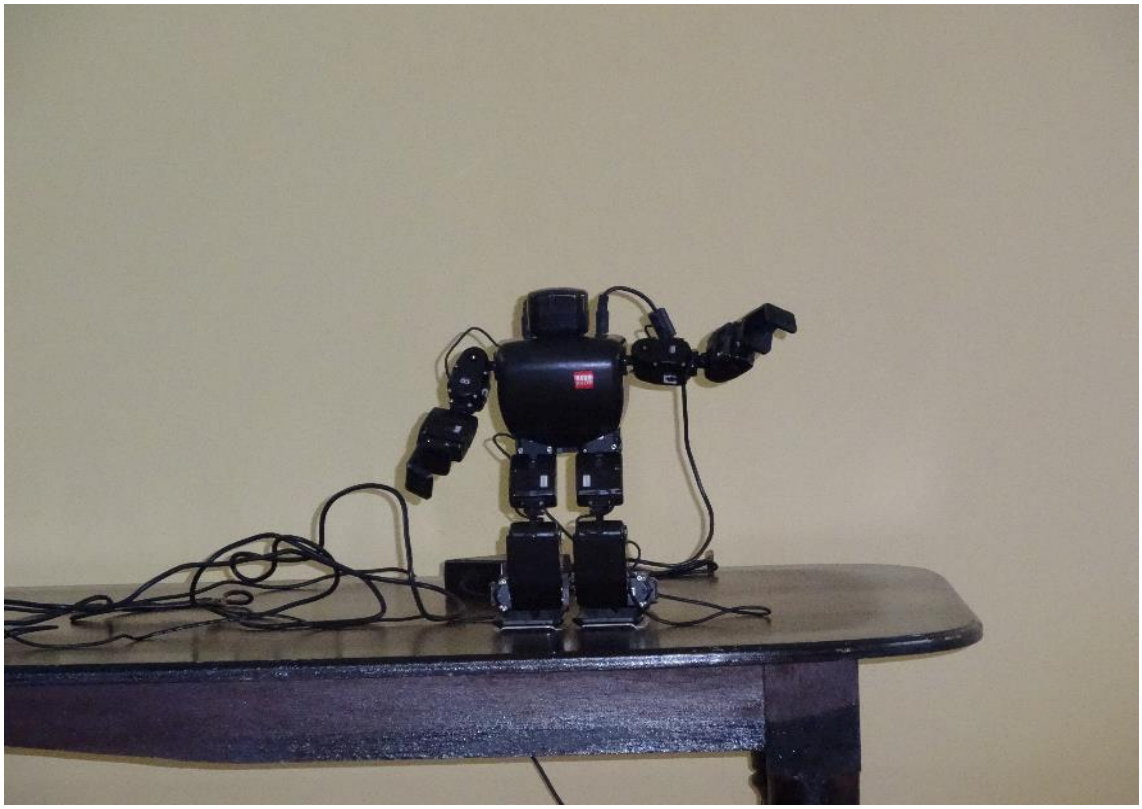


Ilustración 3.8: Movimiento lateral del brazo derecho plataforma robótica educativa.

Elaborado por: Autor

3.1.5 Movimiento hacia adelante del brazo izquierdo:

Al igual que con el brazo derecho, se muestra los datos de profundidad y color stream en la aplicación y el movimiento de la extremidad izquierda de la plataforma robótica educativa hacia adelante.

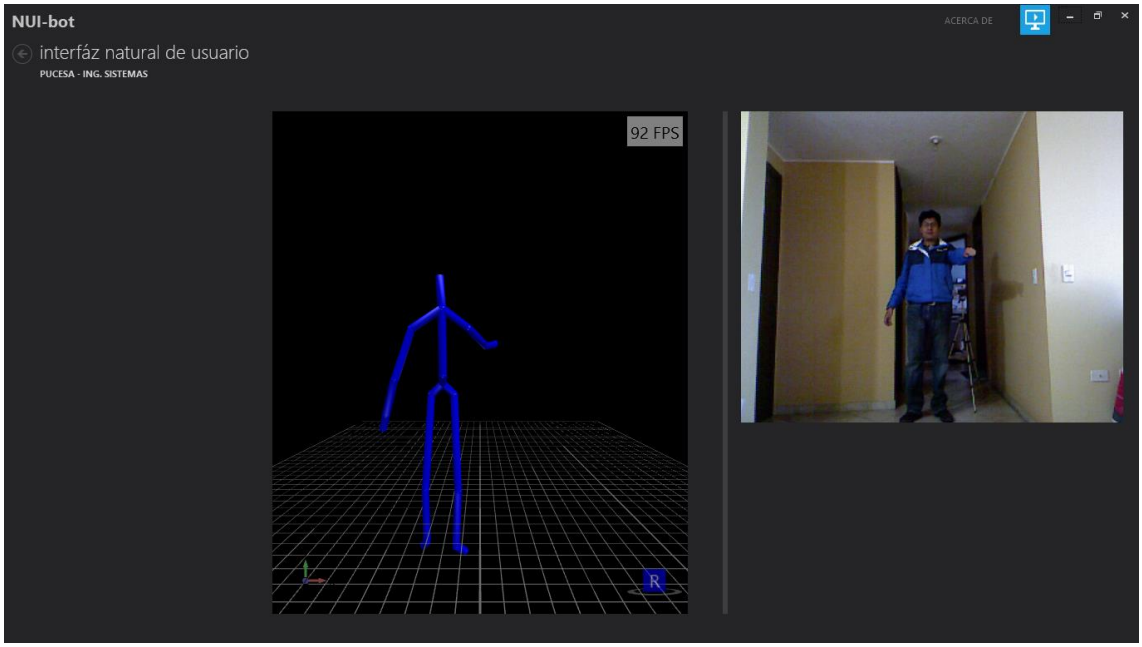


Ilustración 3.9: Movimiento hacia adelante brazo derecho

Elaborado por: Autor.

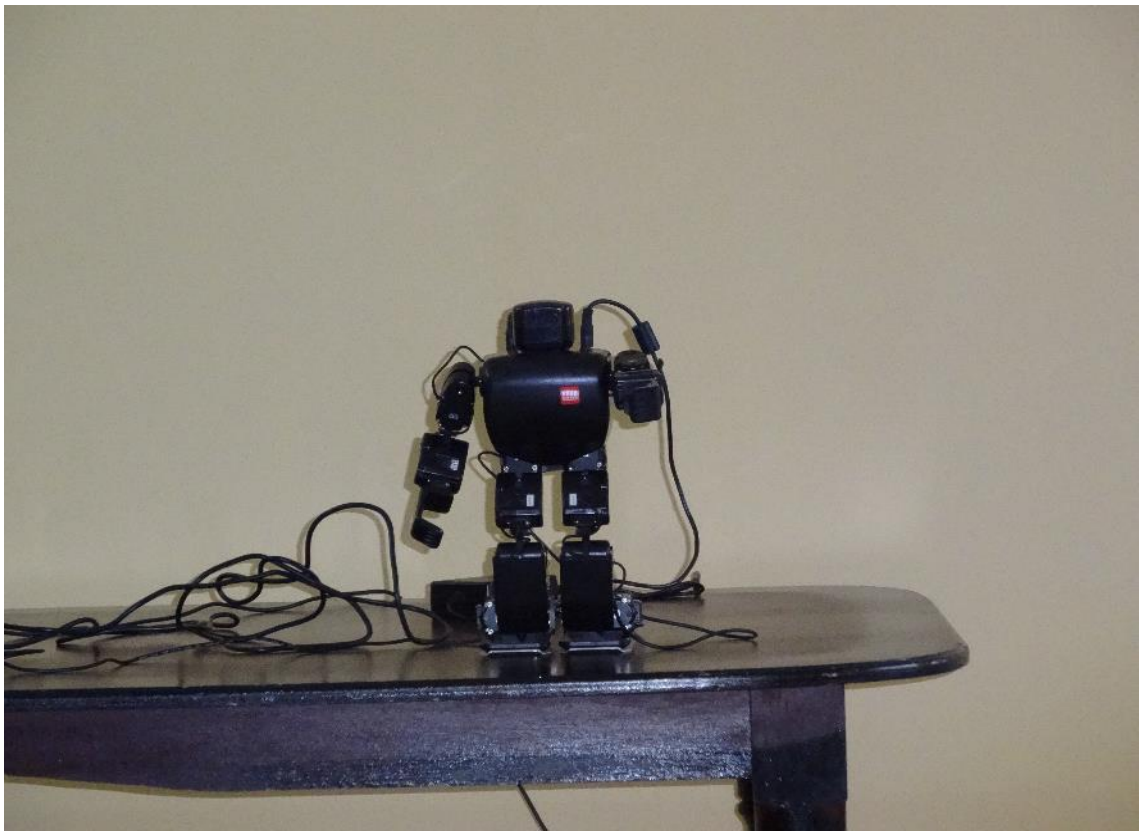


Ilustración 3.10: Movimiento hacia adelante del brazo derecho plataforma robótica educativa.

Elaborado por: Autor

3.1.6 Movimiento hacia arriba de la extremidad izquierda

Al igual que en la extremidad derecha, se muestra el movimiento hacia arriba del brazo izquierdo, los datos de profundidad y color stream se muestra en la aplicación, así como también el movimiento de la extremidad izquierda de la plataforma robótica educativa.

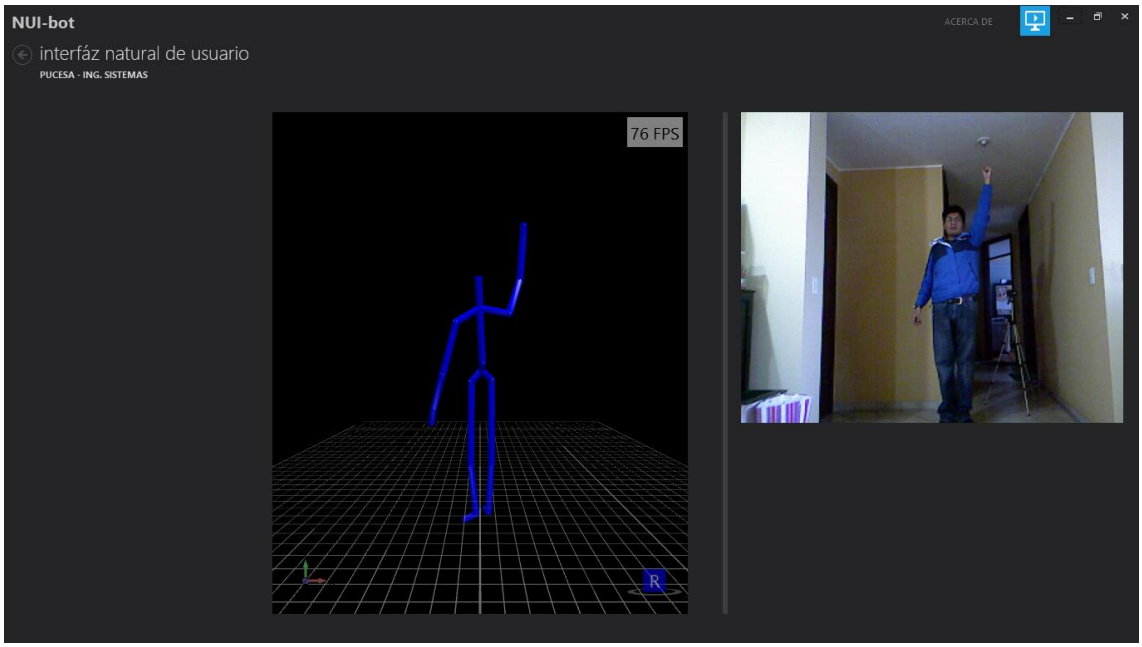


Ilustración 3.11: Movimiento hacia arriba del brazo derecho

Elaborado por: Autor



Ilustración 3.12: Movimiento hacia arriba del brazo izquierdo de la plataforma robótica

Elaborado por: Autor

3.2 Fase 2: Movimiento en conjunto de los Brazos

El movimiento en conjunto de las extremidades superiores es similar al movimiento individual, los datos de profundidad y color stream, se representan en la aplicación, y estos a su vez se transmiten a la plataforma robótica, replicando estos movimientos. Los 16 grados de libertad limitan los movimientos de la plataforma, y la imitación no es 100% igual.

3.2.1 Movimiento lateral de las dos extremidades superiores

Se muestra el movimiento lateral de las dos extremidades, se muestra los datos de profundidad y color stream, además se muestra el movimiento de las extremidades superiores de la plataforma robótica.

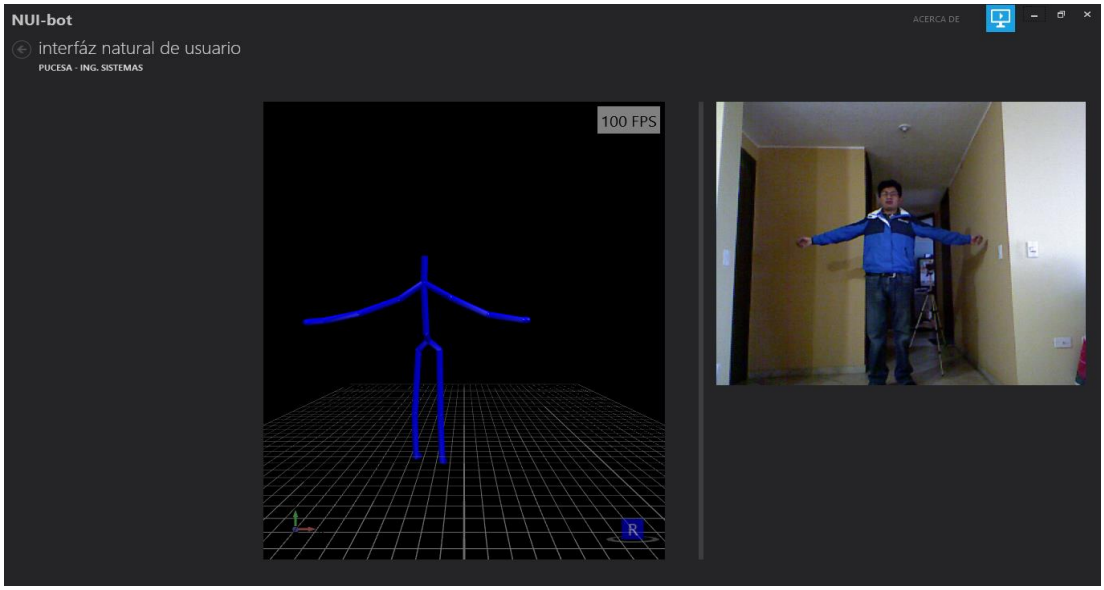


Ilustración 3.13: Movimiento lateral de los brazos

Elaborado por: Autor.

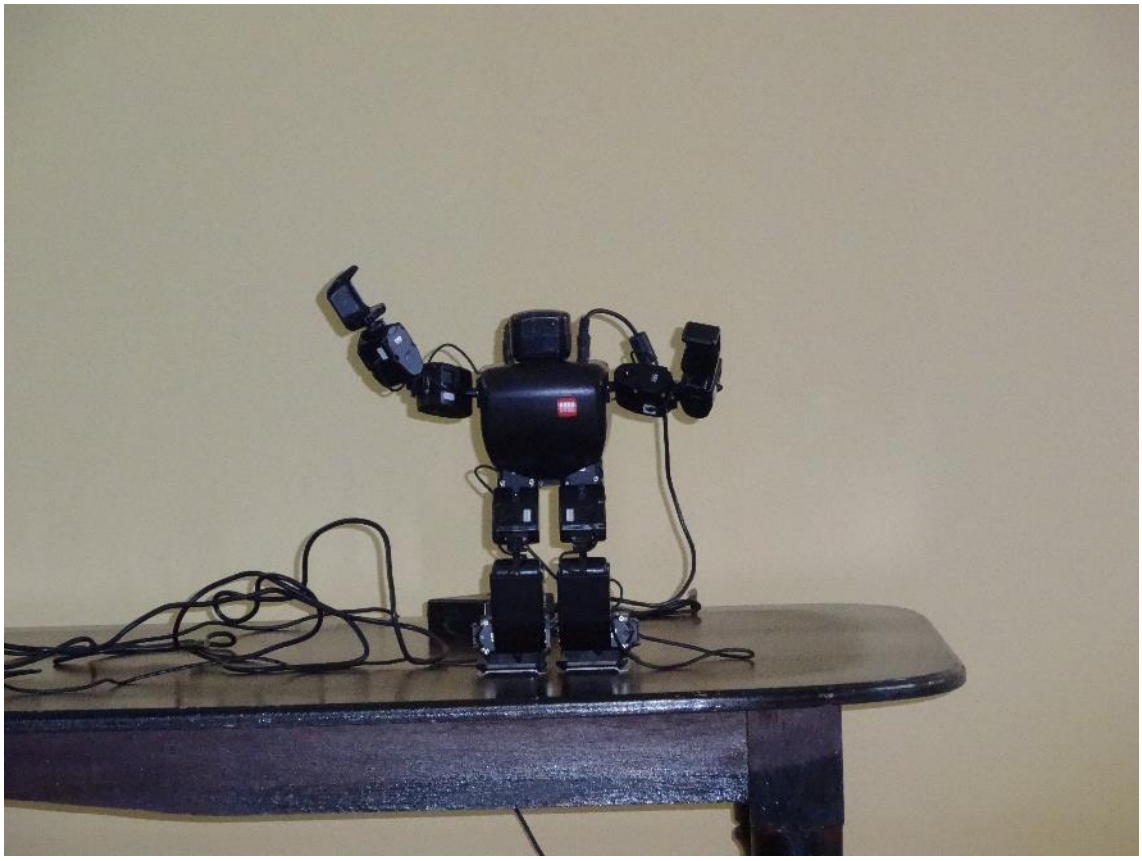


Ilustración 3.14: Movimiento lateral de los brazos de la plataforma robótica

Elaborado por: autor.

3.2.2 Movimiento hacia adelante de las extremidades superiores.

Se representa la captura de los datos de profundidad y color stream en la aplicación, y estos datos se replican en las extremidades superiores de la plataforma robótica educativa.

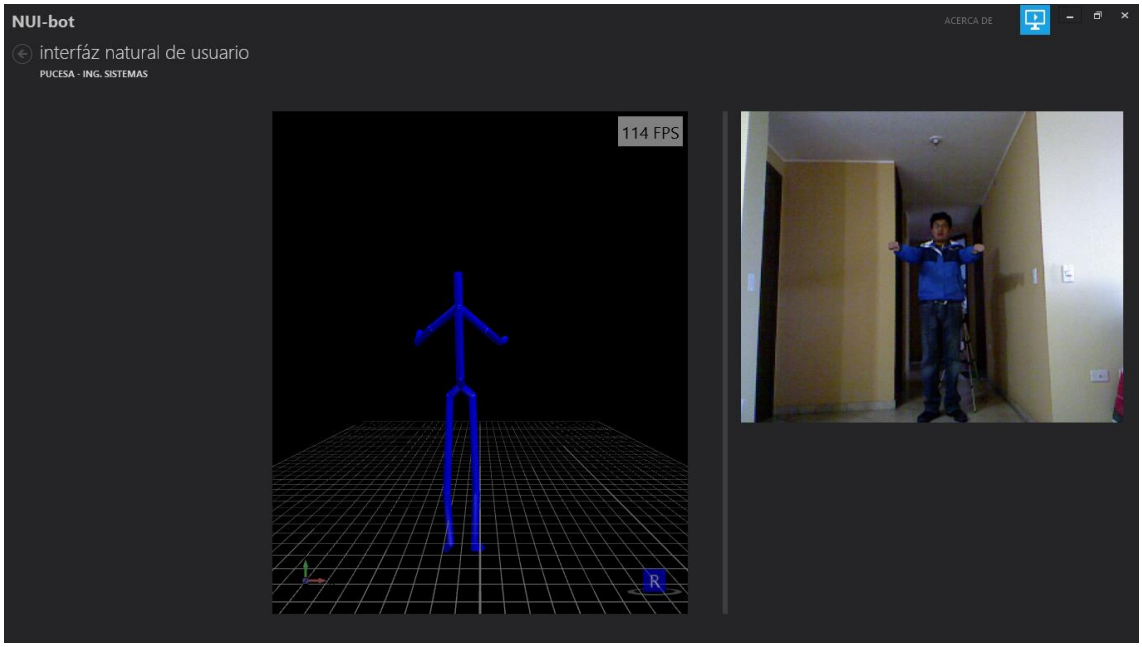


Ilustración 3.15: Movimiento hacia delante de los brazos

Elaborado por: Autor

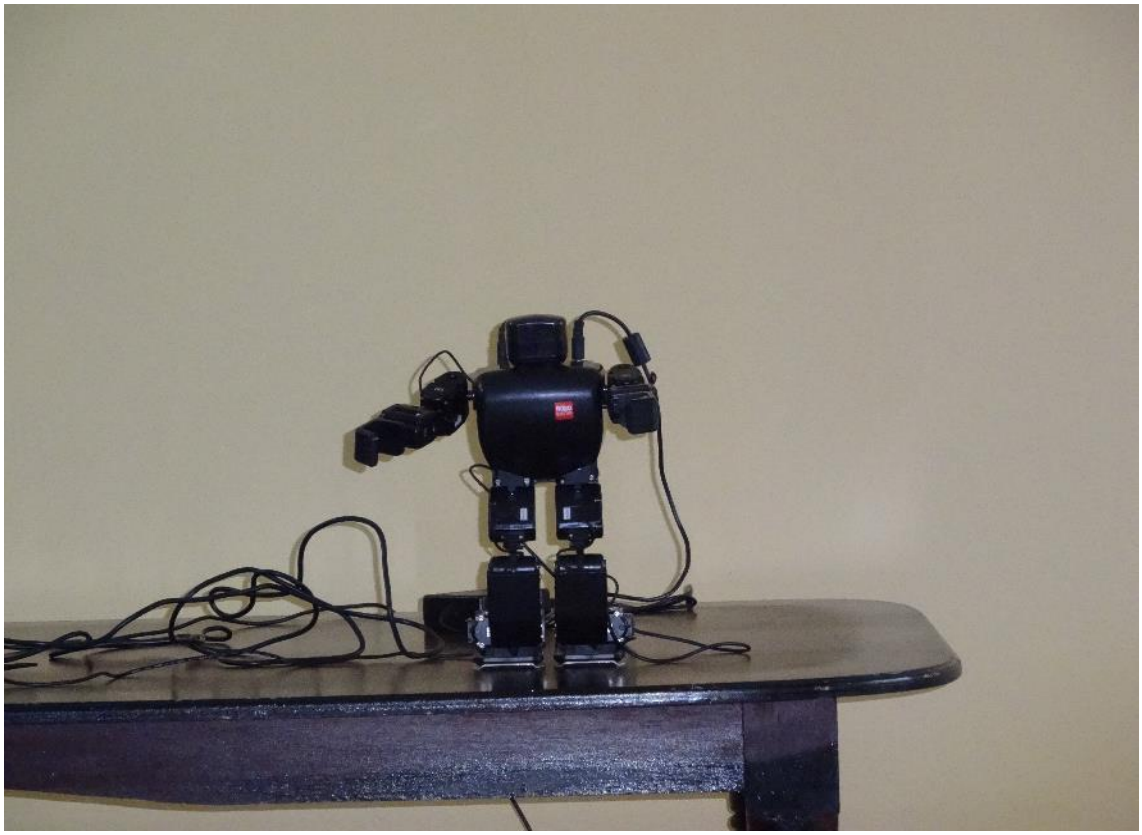


Ilustración 3.16: Movimiento hacia delante de los brazos dela plataforma robótica.

Elaborado por: Autor.

3.2.3 Movimiento hacia arriba de las extremidades superiores

Se muestra la captura de los datos de profundidad y color stream en la aplicación, y los movimientos de las extremidades superiores de la plataforma robótica educativa.

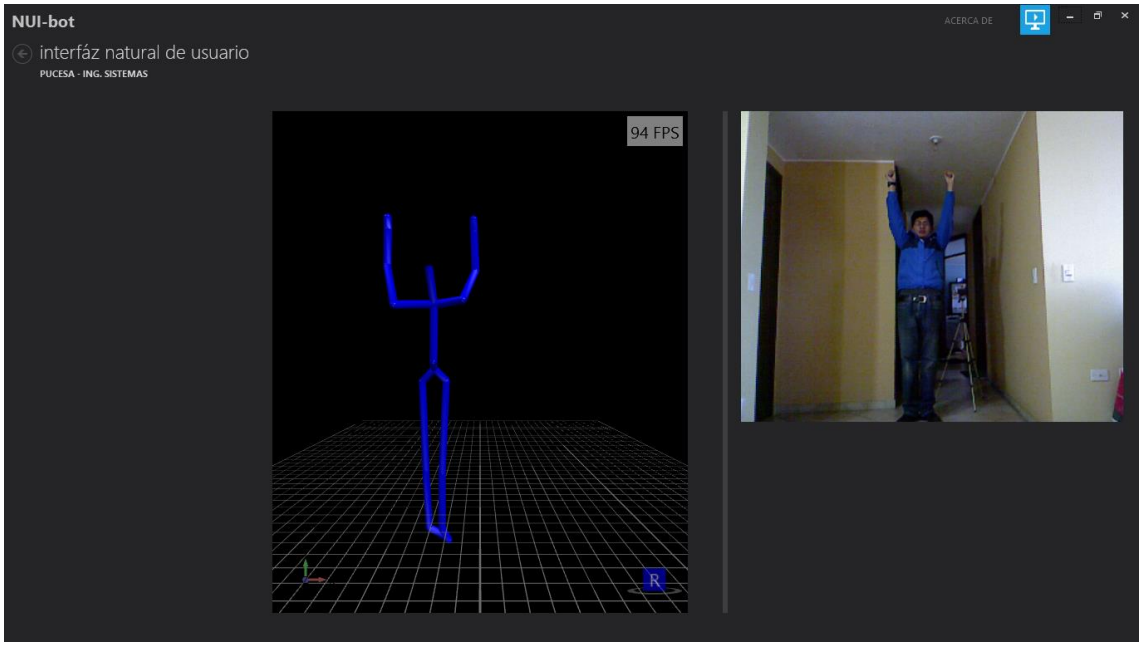


Ilustración 3.17: Movimiento hacia arriba de los brazos

Elaborado por: Autor.



Ilustración 3.18: Movimiento hacia arriba de los brazos de la plataforma robótica:

Elaborado por: Autor.

CAPÍTULO IV.

ANÁLISIS Y VALIDACIÓN DE RESULTADOS

4.1 Análisis de Resultados

Con el presente proyecto, se ha podido demostrar que el uso del sensor Kinect integrado a una aplicación con Interfaz Natural de Usuario, facilita el control en tiempo real de las extremidades superiores de la plataforma robótica educativa RoboBuilder como se muestra en la imagen (véase ***Ilustración 4.1***).

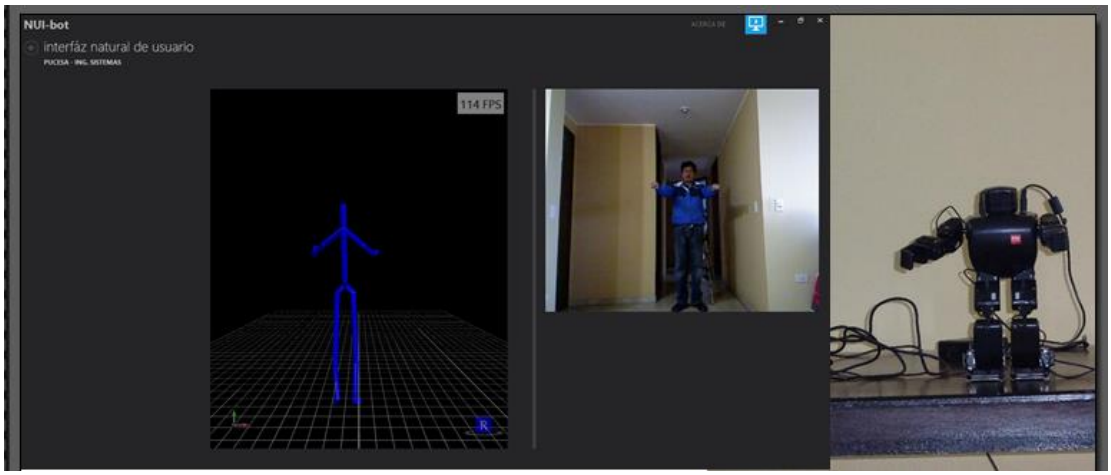


Ilustración 4.1: Análisis de Resultados del Sistema

Elaborado por: Autor.

Estas pruebas se realizaron en un computador con las características que se muestran en la siguiente tabla:

Tabla 4.1: Hardware y Software Utilizado en las Pruebas

Hardware:	
Componente	Descripción
CPU	Intel Core I5, 1.60GHZ 2,3CHZ
Memoria RAM	6 Gb DDR3L
Puerto USB	3.0
Disco Duro	500GB
Software:	

Sistema Operativo	Windows 8.1
Visual Studio	Express 2013 para Escritorio de Windows

Elaborado por: Autor.

4.1.1 Análisis de la Distancia del Sensor

Para una captura optima de los movimientos del usuario, la distancia con respecto al sensor debe ser de por lo menos un metro (véase, **Ilustración 4.2**), si la distancia es menor a un metro la captura de los movimientos del usuario no será posible (véase, **Ilustración 4.3**).

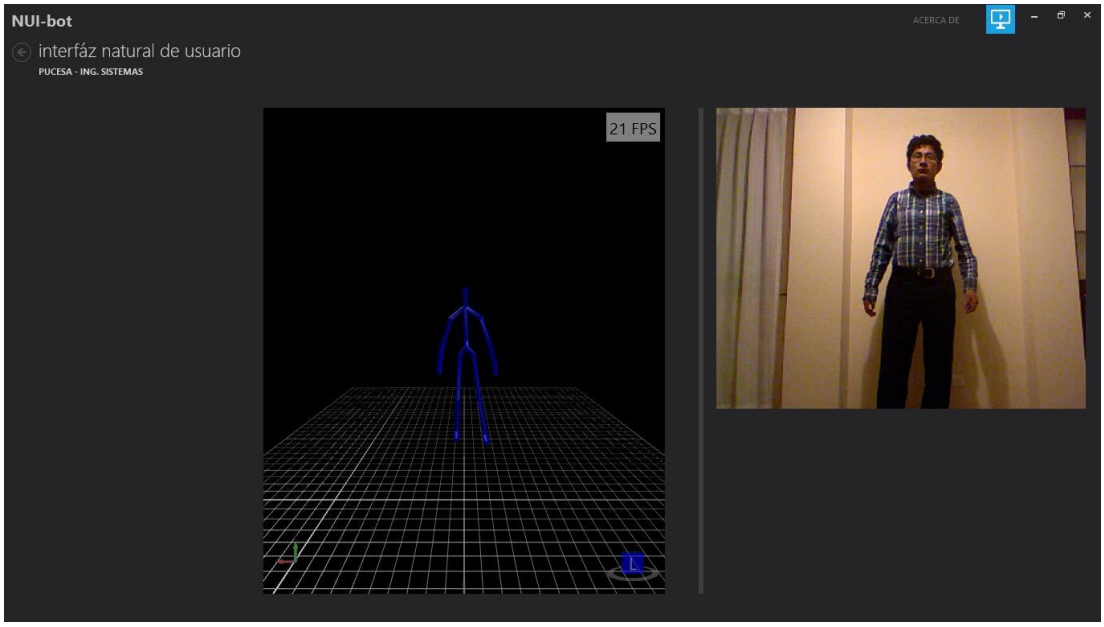


Ilustración 4.2: Distancia de 1 metro entre el usuario y el Sensor

Elaborado por: Autor.

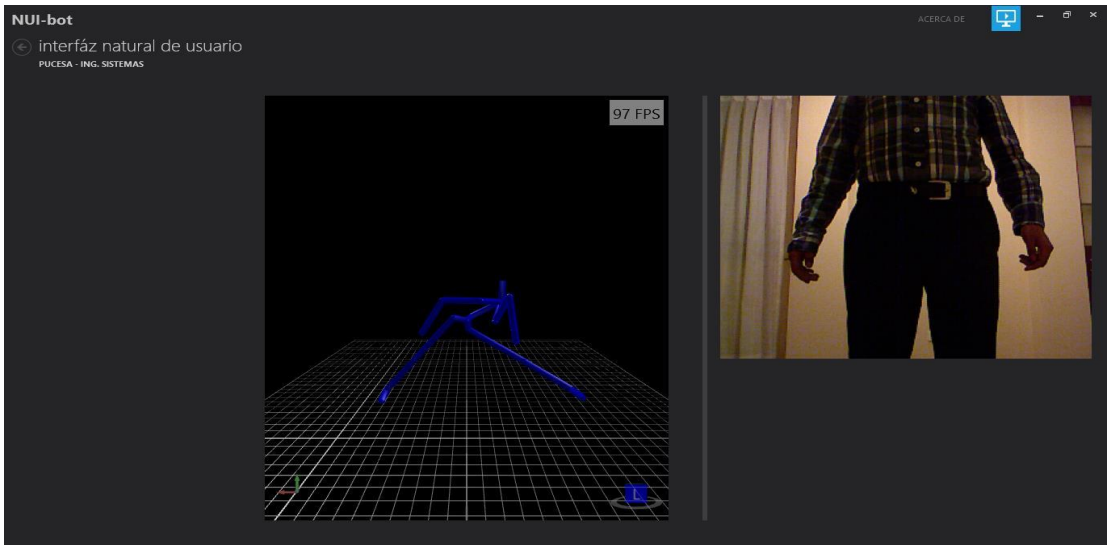


Ilustración 4.3: Distancia inferior a un metro entre el usuario y el sensor Kinect

Elaborado por: Autor.

4.1.2 Análisis de los valores de Servo-Motores y Ángulos Corporales

Los ángulos corporales obtenidos luego del cálculo realizado mediante la aplicación toman los valores correspondientes a las posiciones de los servo-motores, tanto para la extremidad derecha (véase, **Tabla 4.3**), como la extremidad izquierda (véase, **Tabla 4.2**).

Tabla 4.2: Posiciones Servo-Motores y Ángulos Corporales Extremidad Izquierda

Posición servo10	Ángulo Corporal	Posición servo11	Ángulo Corporal	Posición servo12	Ángulo Corporal
73	145	70	171	76	-58

98	142	99	126	85	-26
138	158	102	199	97	-29

Elaborado por: Autor.

Tabla 4.3: Posiciones Servo-Motores y Ángulos Corporales Extremidad Derecha

Posición servo13	Ángulo Corporal	Posición servo14	Ángulo Corporal	Posición servo15	Ángulo Corporal
146	151	133	153	146	-6
159	138	145	201	157	-20
188	151	169	168	160	-48

Elaborado por: Autor.

Los ángulos corporales pueden tener valores negativos, mientras que las posiciones de los servo-motores únicamente aceptan valores positivos.

4.2 Validación de resultados

Para la validación de resultados se han seleccionado 5 usuarios no expertos que han probado el aplicativo, la prueba consistió en utilizar la aplicación durante unos minutos, en primera instancia el usuario ejecuta movimientos sencillos para familiarizarse con la aplicación, para finalmente realizar movimientos más naturales.

Para validar la satisfacción se ha pedido a los usuarios llenar la encuesta (*véase **Anexo E)*, de la cual se han obtenido los siguientes resultados:

La media de satisfacción para la pregunta 1 es: 4,2 (véase ***Ilustración 4.4***).

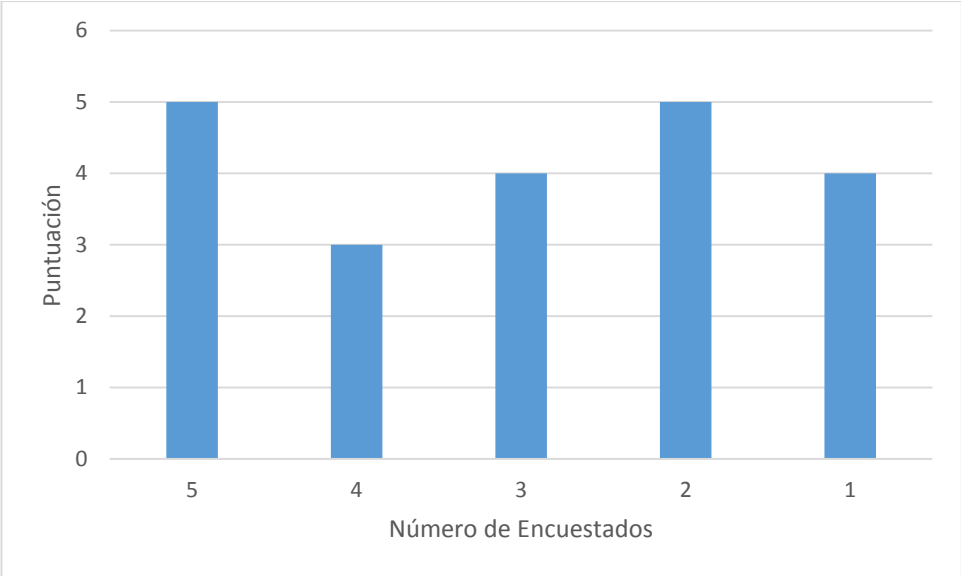


Ilustración 4.4: Pregunta Uno

Elaborado por: Autor.

La media para la pregunta 2 es: 4(véase **Ilustración 4.5**).

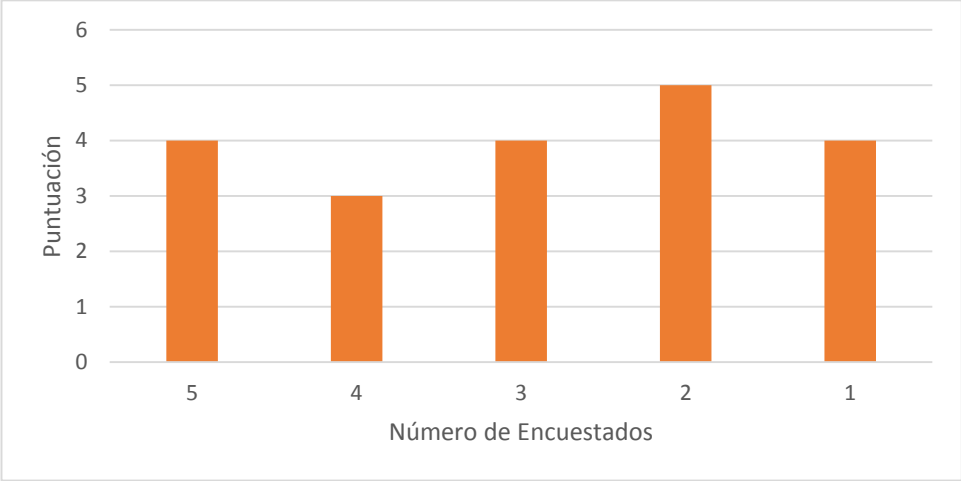


Ilustración 4.5: Pregunta Dos

Elaborado por: Autor.

La media para la pregunta 3 es de: 3,8(véase **Ilustración 4.6**).

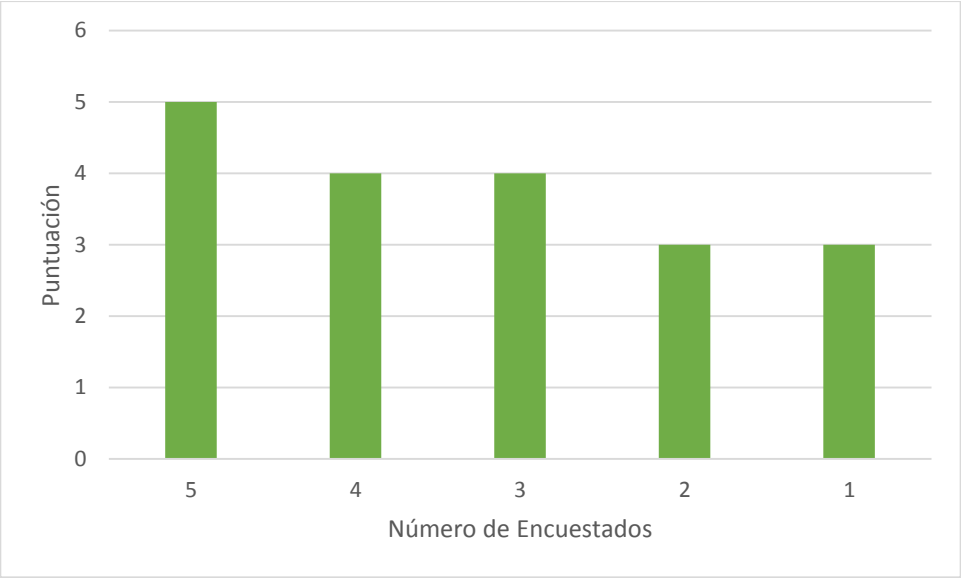


Ilustración 4.6: Pregunta Tres

Elaborado por: Autor.

La media para la pregunta 4 es de: 4 (véase **Ilustración 4.7**).

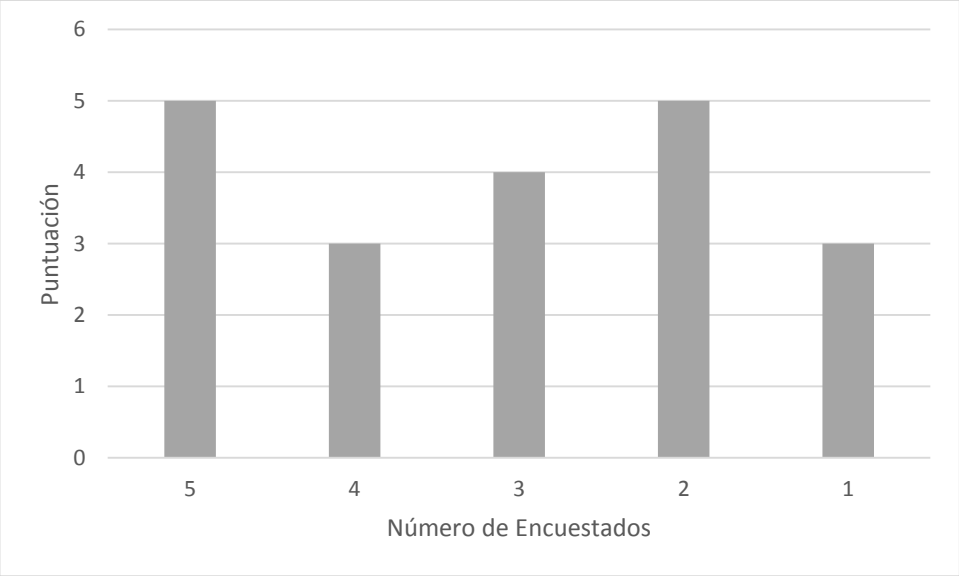


Ilustración 4.7: Pregunta Cuatro

Elaborado por: Autor.

La media de satisfacción general es de 4, que de acuerdo al criterio de evaluación es Bueno.

CAPÍTULO V.

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Una Interfaz Natural de Usuario (NUI), facilita la tele operación de las extremidades superiores de la plataforma robótica educativa RoboBuilder, ya que puede ser utilizada por la mayoría de usuarios.

Los movimientos de las extremidades superiores no son tan flexibles como los movimientos del usuario, esto se debe a que la plataforma robótica RoboBuilder tiene 16 grados de libertad.

Los modelos matemáticos para el cálculo de los valores de las extremidades no son genéricos, y resulta más fácil realizar estos cálculos con principios físicos, donde se representan a las extremidades como vectores y se puede calcular los ángulos corporales.

Los rangos que Kinect maneja, impiden que los usuarios caminen y estos movimientos sean imitados por la plataforma robótica, además RoboBuilder en su versión 5710k, no posee un acelerómetro ni un giroscopio, esta carencia impide que la plataforma robótica se equilibre, por lo que no se incluyó el movimiento de las extremidades inferiores.

El SDK de Kinect facilita la interacción del sensor con la computadora, independientemente de la versión, los paquetes NuGet disminuyen las líneas de código y el resultado resulta mucho más óptimo. La utilización de la librería RoboBuilderLib, facilita la comunicación serial entre la aplicación y la plataforma robótica por medio de la comunicación RS232.

5.2 Recomendaciones

La aplicación debe ser utilizada con un solo usuario frente al sensor, aunque el SDK de Kinect puede detectar máximo 2 esqueletos, se necesitaría más recursos de hardware para procesar un flujo de datos más grande, lo que imposibilitaría que la aplicación se ejecute en un computador estándar.

Los usuarios de la aplicación no deben ser niños pequeños porque, el sensor Kinect no funciona correctamente, ya que su estatura anatómica no está dentro de los patrones pre-programados que realizan el mapeo del esqueleto del usuario.

Es mejor que los usuarios no usen chompas gruesas, o en el caso de las mujeres faldas o vestidos, esto dificulta el procesamiento del flujo de datos de los usuarios que están frente al sensor.

Utilizar una plataforma robótica que incluya un acelerómetro y un giroscopio para agregar la funcionalidad de las extremidades inferiores y el robot no pierda el equilibrio, además de una banda caminadora para el usuario, para que el robot pueda caminar y el usuario se mantenga a una distancia óptima del sensor.

Utilizar el sistema en ambientes donde no existan objetos de gran tamaño como: muebles, sillas, etc. Este tipo de elementos dificultan la detección de esqueletos por parte del sensor. La iluminación debe ser adecuada, para que el sensor no omita información del usuario en el streaming de datos.

BIBLIOGRAFÍA

Ackerman, E. (7 de Marzo de 2011). *Top 10 Robotic Kinect Hacks*. Obtenido de http://spectrum.ieee.org/automaton/robotics/diy/top-10-robotic-kinect-hacks

Aldebaran. (2006). *Aldebaran SoftBank Group*. Obtenido de Aldebaran SoftBank Group: https://www.aldebaran.com/en/humanoid-robot/nao-robot

AMADOR, A. (24 de Octubre de 2012). *ianahiamadorqp*. Obtenido de ianahiamadorqp: http://ianahiamadorqp.blogspot.com/2012/10/elementos-de-la-comunicacion-de-datos.html

Amuno, A. (s.f.). *Evolution Of User Interface From Batch Computing, Command Line, Graphical User Interface To Metro Interface*. Obtenido de http://amuno.hubpages.com/hub/User-Interface

Barea, R. (s.f.). Obtenido de Introducción a la Robotica: http://virtual.itspa.edu.mx/Robotica/archivos/IntroduccionRobotica.pdf

Barrios, M. (22 de Noviembre de 2014). *mundotech*. Obtenido de mundotech: http://mundotech.net/2013/04/actualizacion-de-google-search-acepta-comandos-en-espanol/

Bejarano, P. (1 de Diciembre de 2013). *Cómo funciona un sistema de reconocimiento de voz* . Obtenido de http://www.eldiario.es/turing/reconocimiento-voz-biometria_0_201230680.html

danirebollo. (s.f.). *danirebollo*. Obtenido de danirebollo: http://www.danirebollo.es/index.php/blog/38-reprogramar-pic18f2550-pickit2

Educación España. (Diciembre de 2009). *educacion.es*. Obtenido de educacion.es: http://recursostic.educacion.es/secundaria/tecnologia/controladora/contenido/Descripcion/Esquemas_Electricos/Circuito_control/circuito_control.htm

Electronics Weekly. (20 de Marzo de 2012). *Electronics Weekly*. Obtenido de Electronics Weekly: http://www.electronicsweekly.com/electro-ramblings/digital-life/multi-touch-evolution-takes-ipad-style-interface-beyond-consumers-2012-03/

Honda. (Noviembre de 2011). *Asimo Honda* . Obtenido de http://asimo.honda.com/asimo-history/

Jana, A. (2012). *Kinect for Windows SDK Programming Guide.* PACKT.

LEGO. (2010). *LEGO Mindstorms*. Obtenido de LEGO Mindstorms: http://www.lego.com/en-us/mindstorms/products/31313-mindstorms-ev3

Microsoft. (Junio de 2012). *SDN Magazine*. Obtenido de SDN Magazine: https://msdn.microsoft.com/en-us/magazine/jj159883.aspx

Microsoft. (s.f.). *Kinect for Windows SDK*. Obtenido de http://msdn.microsoft.com/en-us/library/hh855347.aspx

Microsoft. (s.f.). *Kinect Sensor*. Obtenido de http://msdn.microsoft.com/en-us/library/hh438998.aspx

Multi-touch touchscreen (Pantallas Táctiles). (29 de Octubre de 2009). Obtenido de http://www.neoteo.com/multi-touch-touchscreen-pantallas-tactiles/

O ME TECHNOLOGY. (06 de Marzo de 2013). *O ME TECHNOLOGY*. Obtenido de O ME TECHNOLOGY: https://ometechnology.wordpress.com/2013/03/06/the-improved-kinect-version/

Peña, R. E. (5 de Julio de 2012). *Sensores-Tipos-y-Caracteristicas*. Obtenido de http://es.scribd.com/doc/99149696/PDF-Sensores-Tipos-y-Caracteristicas

Pérez, I. E. (s.f.). INGENIERIA EN MICROCONTROLADORES.

Rees, D. (15 de Noviembre de 2010). *Instructional Design Fusions*. Obtenido de Instructional Design Fusions: https://instructionaldesignfusions.wordpress.com/2010/11/15/natural-user-interfaces-as-natural-learning-tools/

Rivera, C. (23 de Mayo de 2000). *Definición de Interfaz de Usuario*. Obtenido de http://www.fismat.umich.mx/~crivera/tesis/node11.html

Rosenberg , A. (31 de Diciembre de 2013). *Oculus Rift and Kinect used to power robot arm for NASA*. Obtenido de http://www.digitaltrends.com/gaming/oculus-rift-kinect-used-power-robot-arm-nasa/

Spyros G, T. (2009). *Human and Nature Minding Automation.* Springer Netherlands .

Universidad del Táchira. (2003). UNET. San Cristobal, Tachira, Venezuela. Obtenido de http://www.unet.edu.ve/~ielectro/Guia%20de%20puertos.pdf

Vernier. (s.f.). *vernier-iberica*. Obtenido de vernier-iberica: http://www.vernier-iberica.com/dfs.html

Wang, R. (2013). *Augmented Reality with Kinect.* PACKT.

Wilson, i. (2014). *Implementation of Robot Systems.* Elsevier Science.

GLOSARIO

Acelerómetro

Son dispositivos diseñados para detectar cambios en la fuerza resultante de caídas, la inclinación, movimiento, posición, golpes y vibraciones., 42, 55, 88, 89

Ángulos

Un ángulo es la parte del plano comprendida entre dos semirrectas que tienen el mismo punto de origen o vértice, 49, 62, 64, 65, 66, 68, 69, 70, 71, 88

Array

Es una zona de almacenamiento continuo, que contiene una serie de elementos del mismo tipo, los elementos de la matriz., 42

ASIMO

es un robot humanoide (androide) presentado por la compañía japonesa Honda en el año 2000., xii, 29, 30

Avatar

se denomina avatar a una representación gráfica, generalmente humana, que se asocia a un usuario para su identificación. Los avatares pueden ser fotografías o dibujos artísticos, y algunas tecnologías permiten el uso de representaciones tridimensionales., 50, 53

CLI

La interfaz de línea de comandos, traducción del inglés command-line interface o CLI –la cual es, en realidad, una transcripción incorrecta de Interfaz de línea de órdenes, por el falso amigo command (orden/instrucción)– es un método que

permite a los usuarios dar instrucciones a algún programa informático por medio de una línea de texto simple., 23

Driver

Es un programa informático que permite al sistema operativo interaccionar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz (posiblemente estandarizada) para utilizar el dispositivo., 39, 109

Frames

Es una imagen particular dentro de una sucesión de imágenes que componen una animación., 55, 59, 63

GUI

La interfaz gráfica de usuario, conocida también como GUI (del inglés graphical user interface) es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz., 23

Joystick

Es un dispositivo de entrada usado para interaccionar con un videojuego, 21, 24

LEGO

es una línea de juguetes de robótica para niños fabricado por la empresa LEGO, que posee elementos básicos de las teorías robóticas, como la unión de piezas y la

programación de acciones en forma interactiva. Este robot fue comercializado por primera vez en septiembre de 1998., xii, 32, 91

Librería

Es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca., 46, 47, 48, 50, 51, 53, 88

Módems

Es el dispositivo que convierte las señales digitales en analógicas (modulación) y viceversa (demodulación), 33

NAO

es un robot humanoide programable y autónomo, desarrollado por Aldebaran Robotics, una compañía de robótica francesa con sede en París., xii, 31

Nintendo

Es una de las primeras empresas multinacionales en el mercado de los videojuegos, 17

NuGet

es un gestor de paquetes de código libre y abierto para la plataforma de desarrollo de Microsoft, 53, 88

NUI

en inglés natural user interface o NUI) es aquella en las que se interactúa con un sistema, aplicación, etc. sin utilizar sistemas de mando o dispositivos de entrada de las GUI como sería un ratón, teclado alfanumerico, lápiz óptico, Touchpad,

joystick etc. y en su lugar, se hace uso de movimientos gestuales, 18, 20, 23, 59, 87

Pines

se denomina pin, a la terminal o patilla de cada uno de los contactos metálicos de un conector o de un componente fabricado de un material conductor de la electricidad, 34, 35

Pixel

Es la menor unidad homogénea en color que forma parte de una imagen digital., xiii, 56

Play Station

Es una videoconsola de sobremesa, lanzada por Sony Computer, 17

RGB

Es la composición del color en términos de la intensidad de los colores primarios de la luz., xiii, xv, 41, 50, 55, 56, 57, 59

RoboBuilder, xii, 22, 39, 45, 48, 72, 74, 87, 88

Plataforma Robótica Educativa, 22

Routers

Es un dispositivo que proporciona conectividad a nivel de red o nivel tres en el modelo OSI, 33

SDK

Es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto, por ejemplo

ciertos paquetes de software, frameworks, plataformas de hardware, computadoras, videoconsolas, sistemas operativos, etc., xv, 20, 22, 39, 41, 43, 44, 45, 48, 49, 54, 59, 60, 88, 91, 107

Servomotores

Es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición, 32, 46, 48, 49

Siri

Siri es una aplicación con funciones de asistente personal a veces con su propia personalidad para iOS., 26

Stream

Es la distribución digital de multimedia a través de una red de computadoras de manera que el usuario consume el producto, generalmente archivo de video o audio, en paralelo mientras se descarga., 50, 79, 80, 82, 83, 85

Vectores

Es una magnitud física definida por un punto del espacio donde se mide dicha magnitud., xiii, 49, 64, 65, 66, 67, 68, 88

VGA

Sistema gráfico de pantallas para PCs desarrollado por IBM. VGA se convirtió en un estándar de facto para PCs. Las tarjetas gráficas VGA estándares traían 256 KB de memoria de video. - See more at

http

//www.alegsa.com.ar/Dic/VGA.php#sthash.ucA9TzpQ.dpuf, 41

WPF

es una tecnología de Microsoft, presentada como parte de Windows Vista. Permite el desarrollo de interfaces de interacción en Windows tomando características de aplicaciones Windows y de aplicaciones web., 50

XAML

Es el lenguaje de formato para la interfaz de usuario para la Base de Presentación de Windows (WPF por sus siglas en inglés) y Silverlight(wpf/e), el cual es uno de los "pilares" de la interfaz de programación de aplicaciones .NET en su versión 3.0, xiii, 53

XBOX, xii, 17, 42, 43

Consola de VideoJuegos de Microsoft, 17

ANEXOS

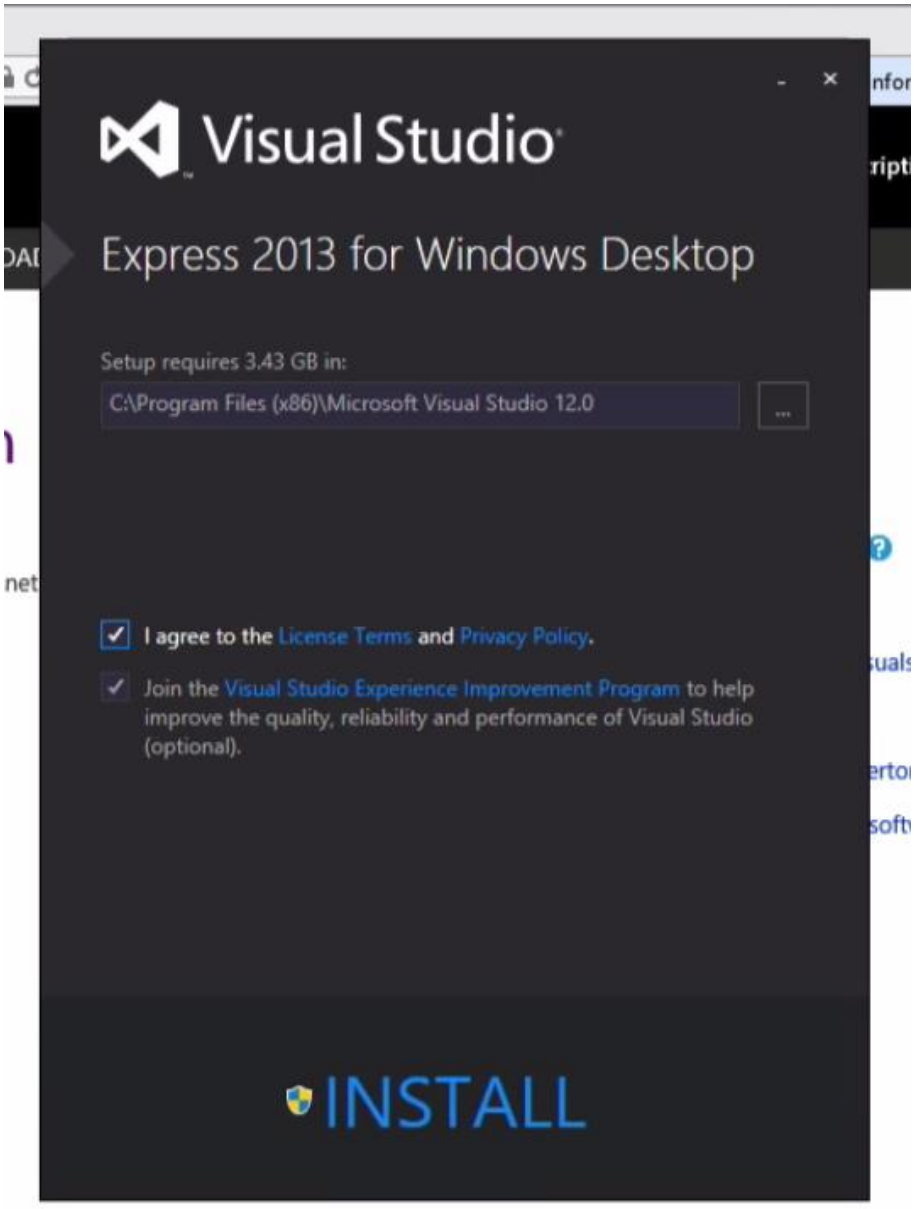
Anexo A

Instalación de Visual Studio 2013 Express para Escritorio

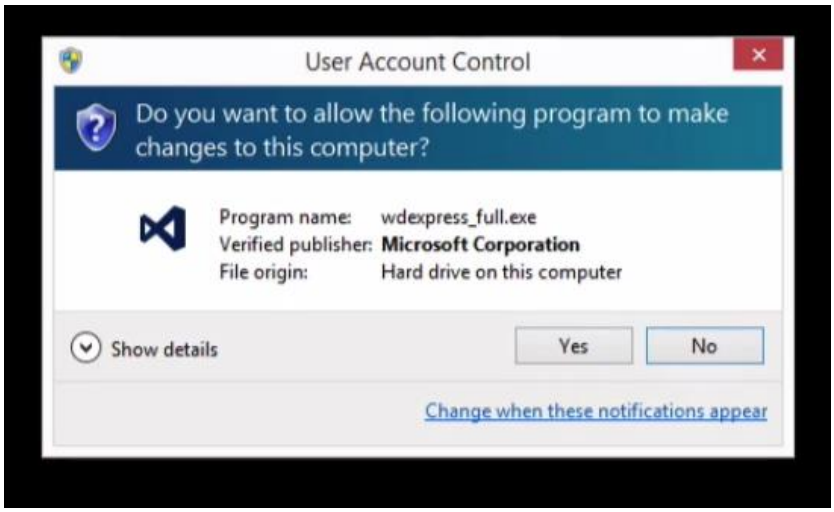
Luego de descargar la imagen ISO, donde esta el instalador de Visual Studio Express 2013 para Escritorio, el programa de instalación iniciara con esta ventana.



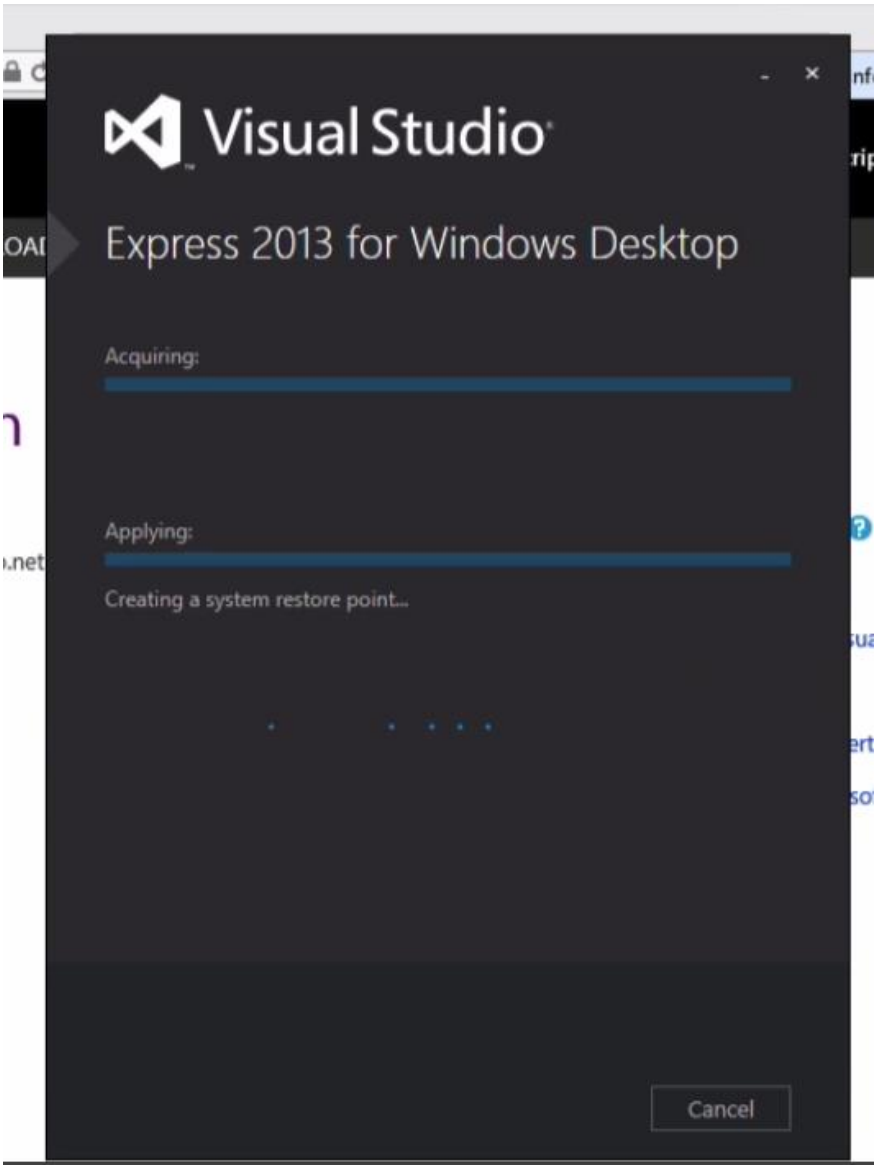
Para continuar con la instalación se debe dar clic en la opción “I agree to the Lincense Terms and Privacy Policy, y asi se habilita el botón de “INSTALL”.


Visual Studio
Express 2013 for Windows Desktop
Setup requires 3.43 GB in:
C:\Program Files (x86)\Microsoft Visual Studio 12.0
...
I agree to the License Terms and Privacy Policy.
Join the Visual Studio Experience Improvement Program to help improve the quality, reliability and performance of Visual Studio (optional).
INSTALL

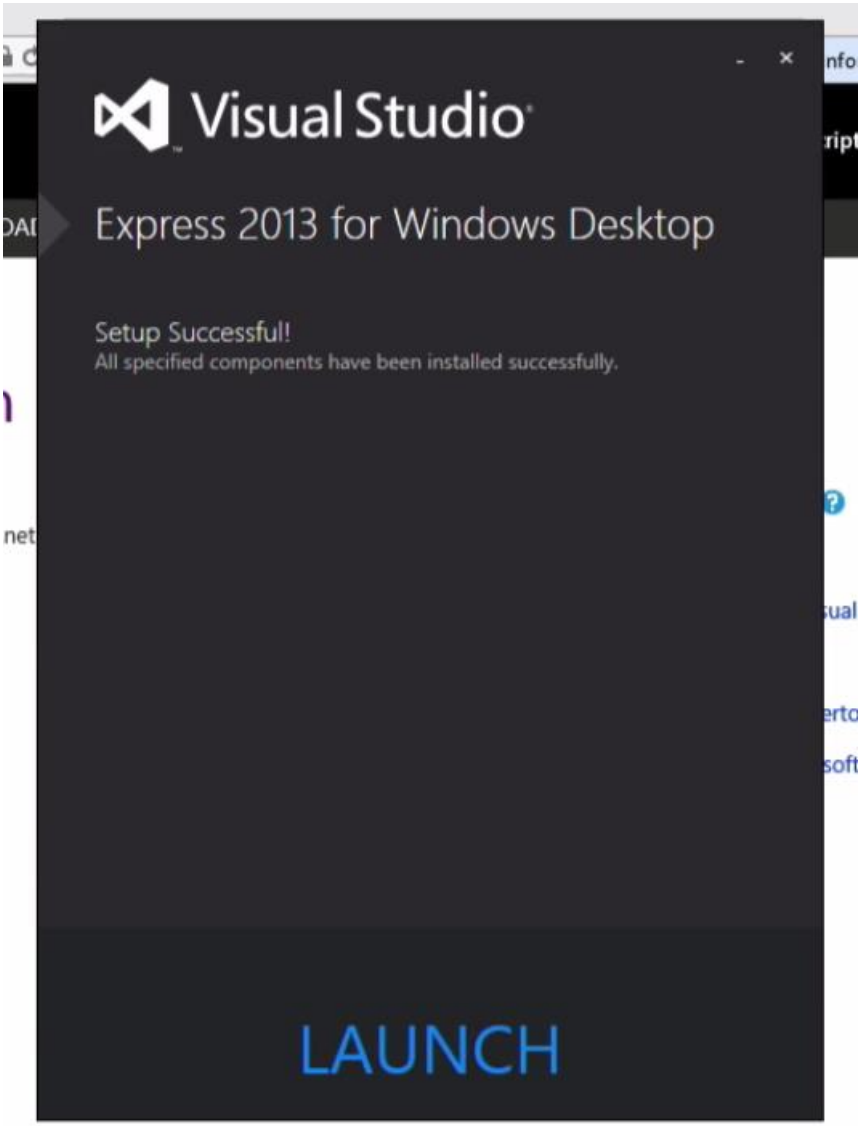
Al dar clic en el botón “INSTALL”, se abre el cuadro de dialogo, pidiendo autorización para instalar la aplicación, dar clic en “YES” para continuar.



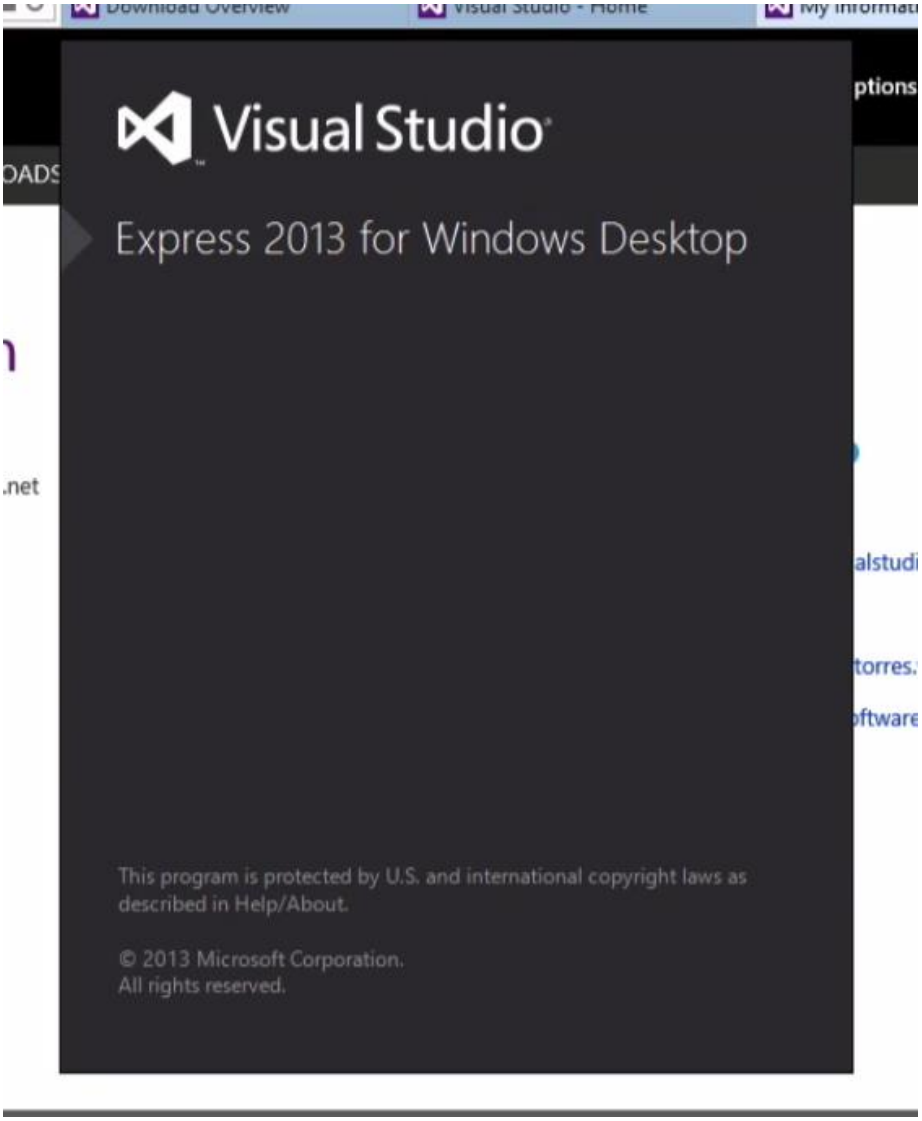
La instalación continúa automáticamente hasta llegar al final



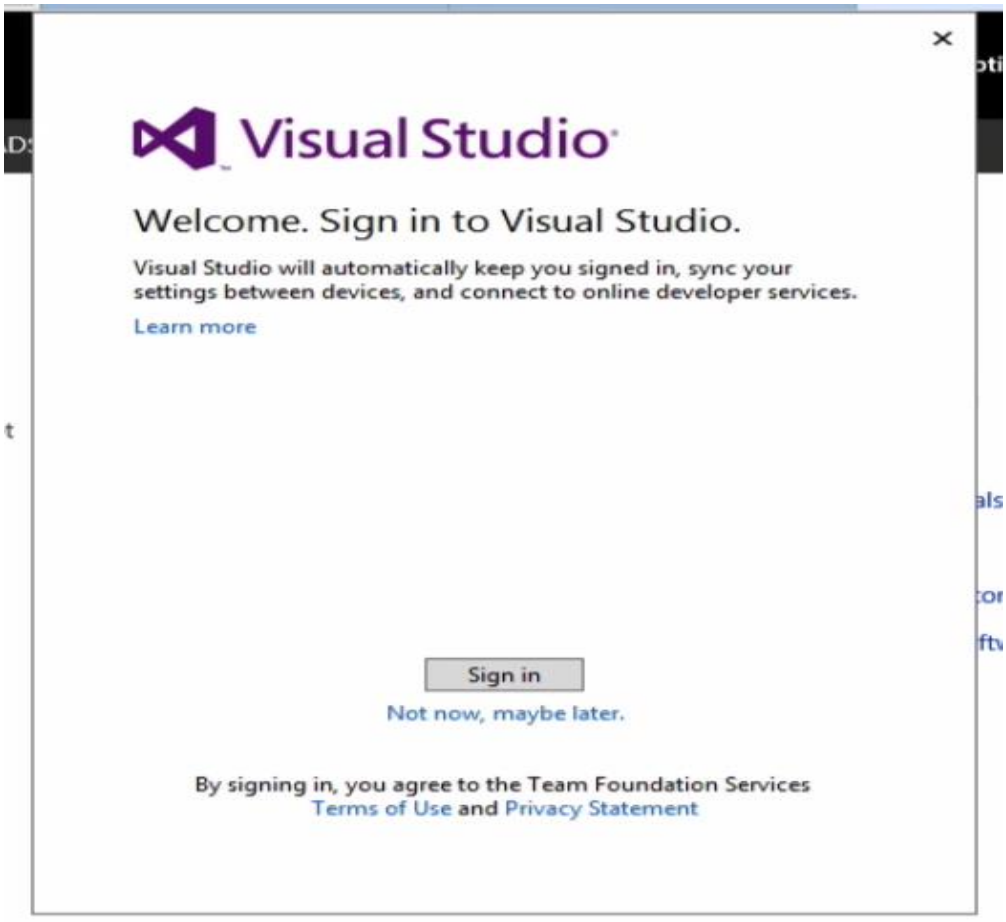
La instalación dura algunos minutos, al finalizar se habilita el botón “LAUNCH”



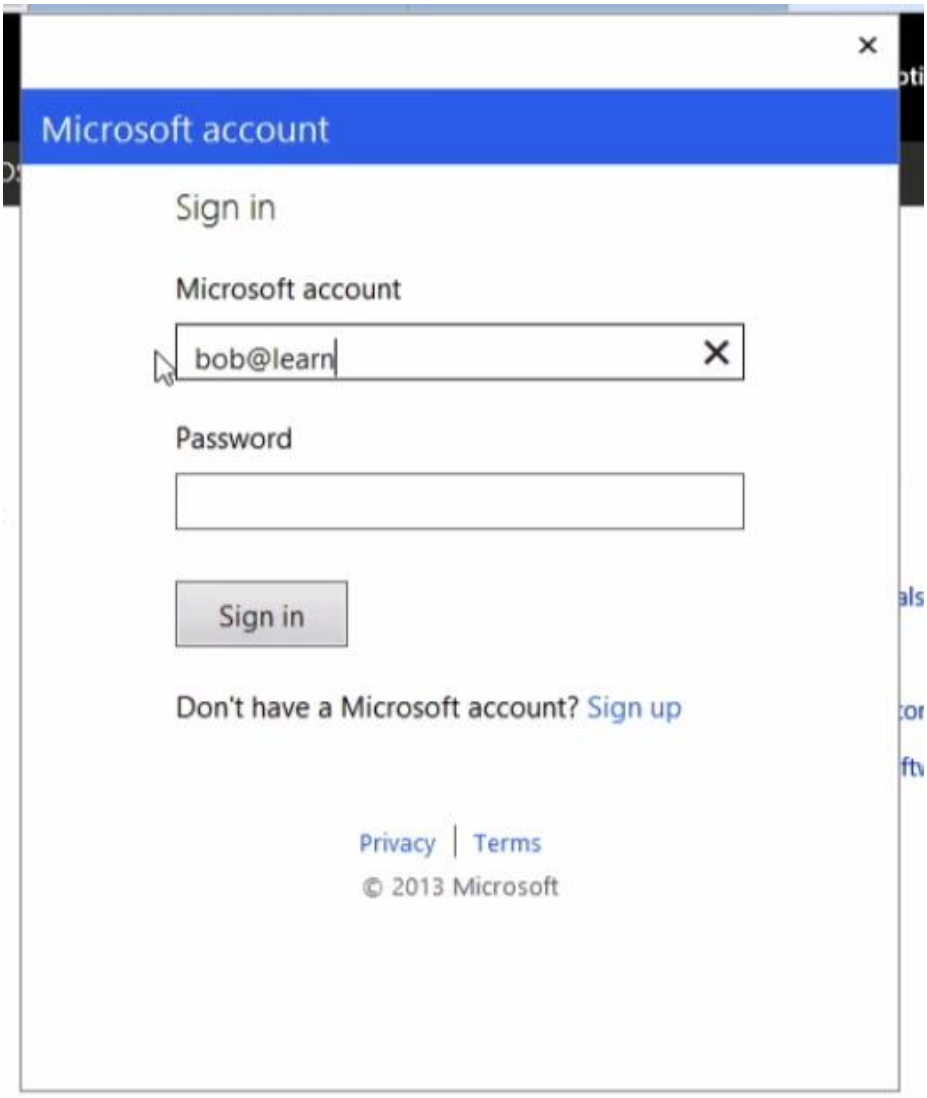
Al hacer clic en el botón “LAUNCH” inicia la aplicación



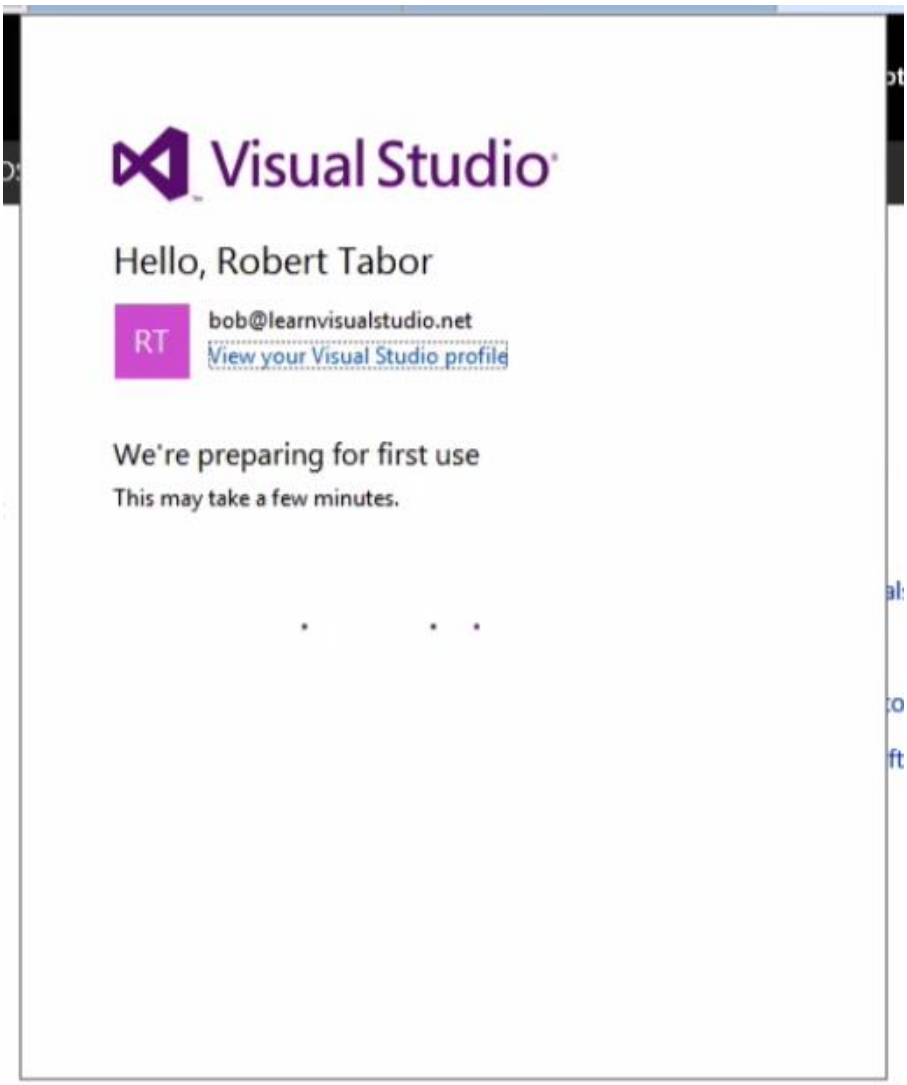
Para el primer uso se debe iniciar la sesión con un correo de Microsoft



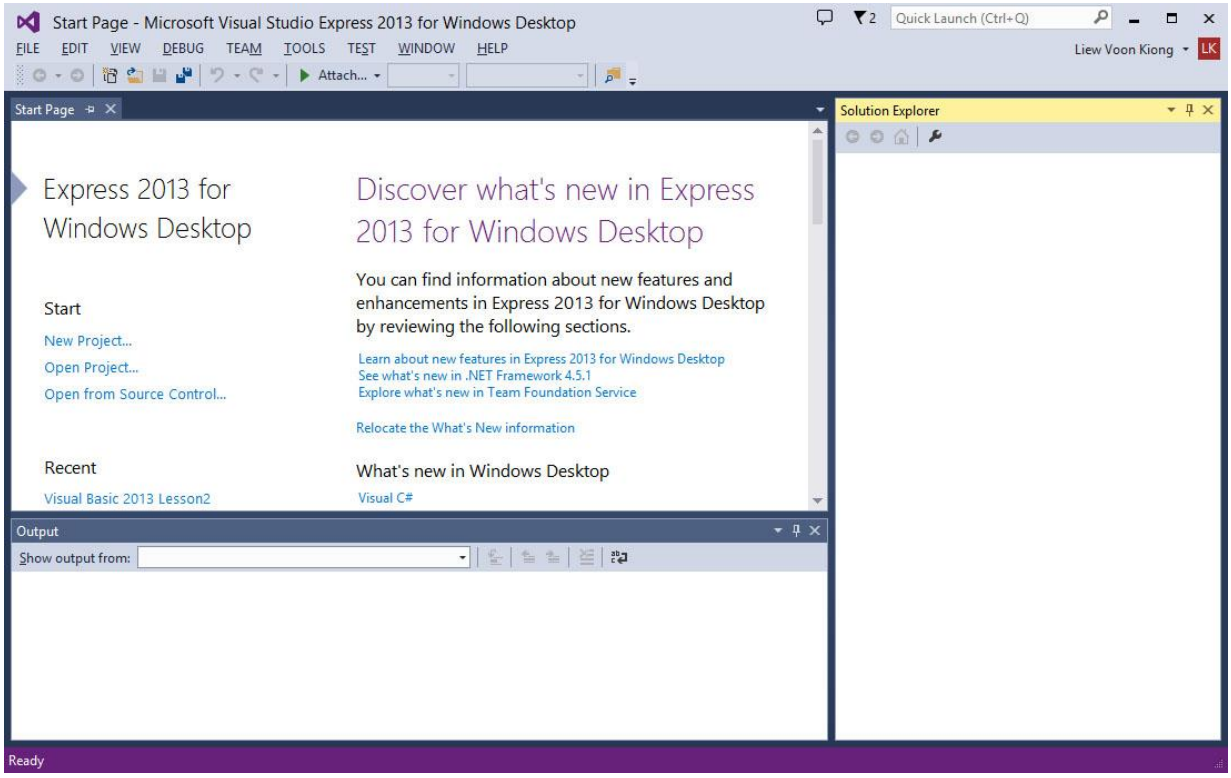
Llenar los datos de inicio de sesión



Luego de iniciar la sesión, la aplicación comienza a preparar para el primer uso, tarda unos minutos.



Luego de la configuración automática que hace Visual Studio, se abre la pantalla de inicio de Visual Studio Express 2013

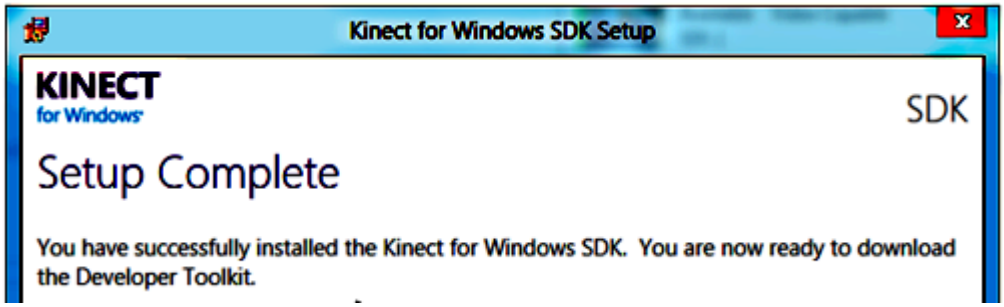


Anexo B

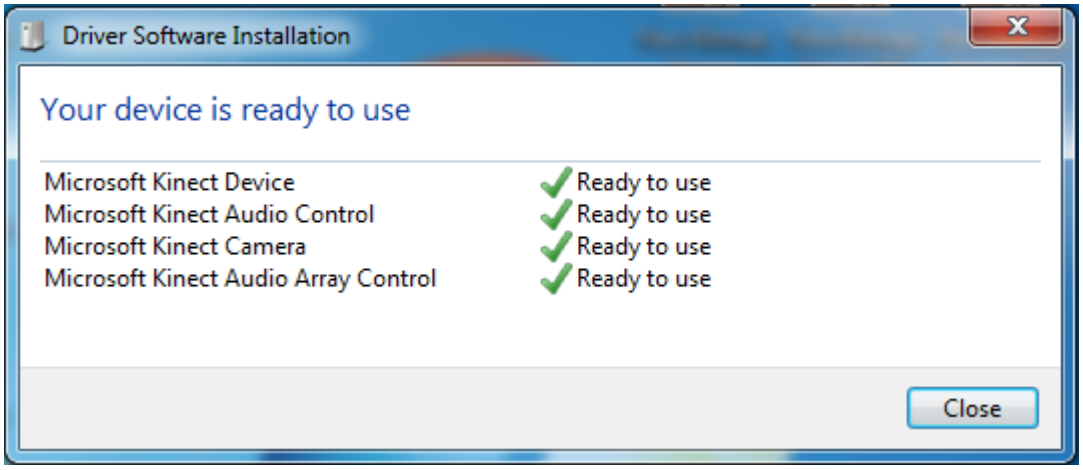
Instalación del SDK de Kinect

Luego de descargar el instalador del SDK oficial de Kinect versión 1.8, se debe dar clic derecho e iniciar la instalación con permisos de Administrador del sistema, a continuación se despliega la pantalla de inicio de la instalación, donde se debe aceptar los términos legales y dar clic en el botón “INSTALL”

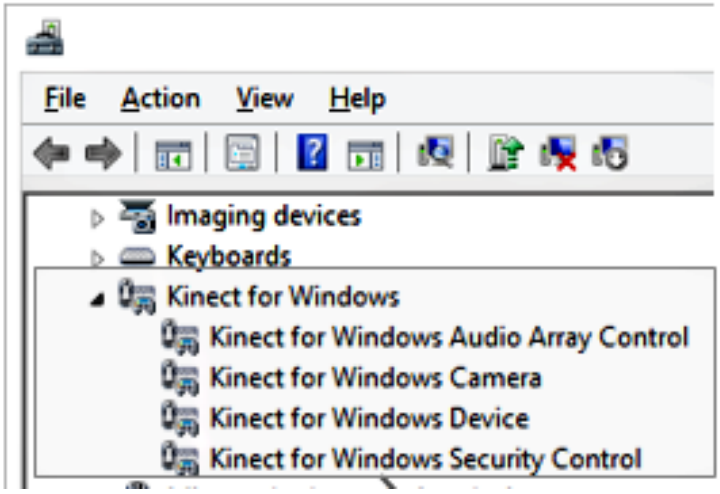




Si la instalación finaliza correctamente conecte el sensor al computador y aparecerá el siguiente cuadro de dialogo, indicando que el sensor está listo para ser utilizado:



Para verificar que el sensor ya es detectado por el sistema, ingrese a panel de control y seleccione Administrador de Dispositivos



Anexo C

Instalación de driver para el adaptador USB-TU-S9 marca TRENED.

Mediante este adaptador se puede convertir el conector USB en un puerto RS232.

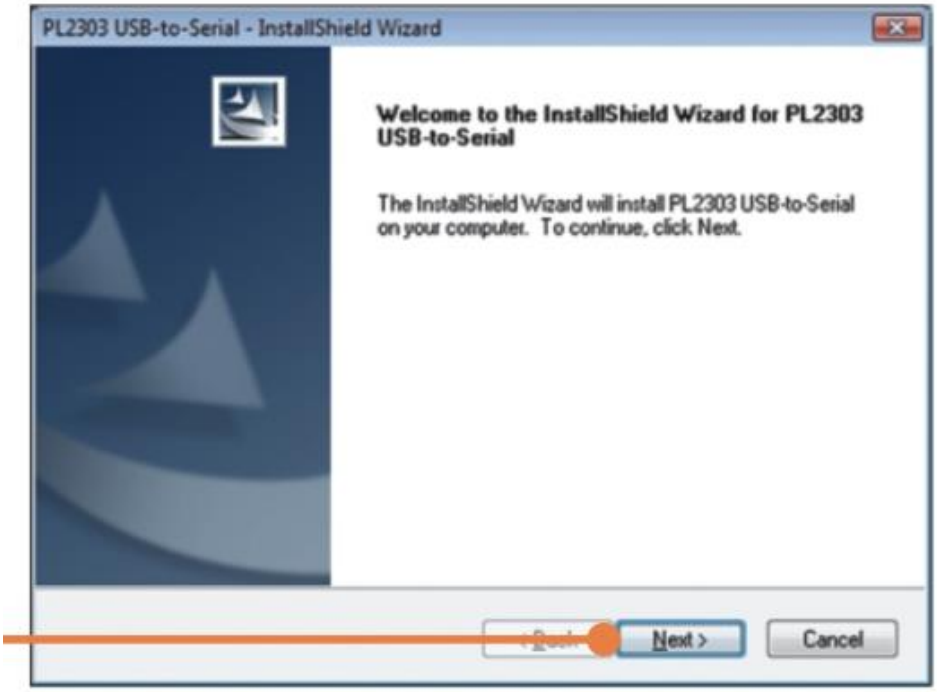
Abrir el CD, y ejecutar el archivo Autorun.exe



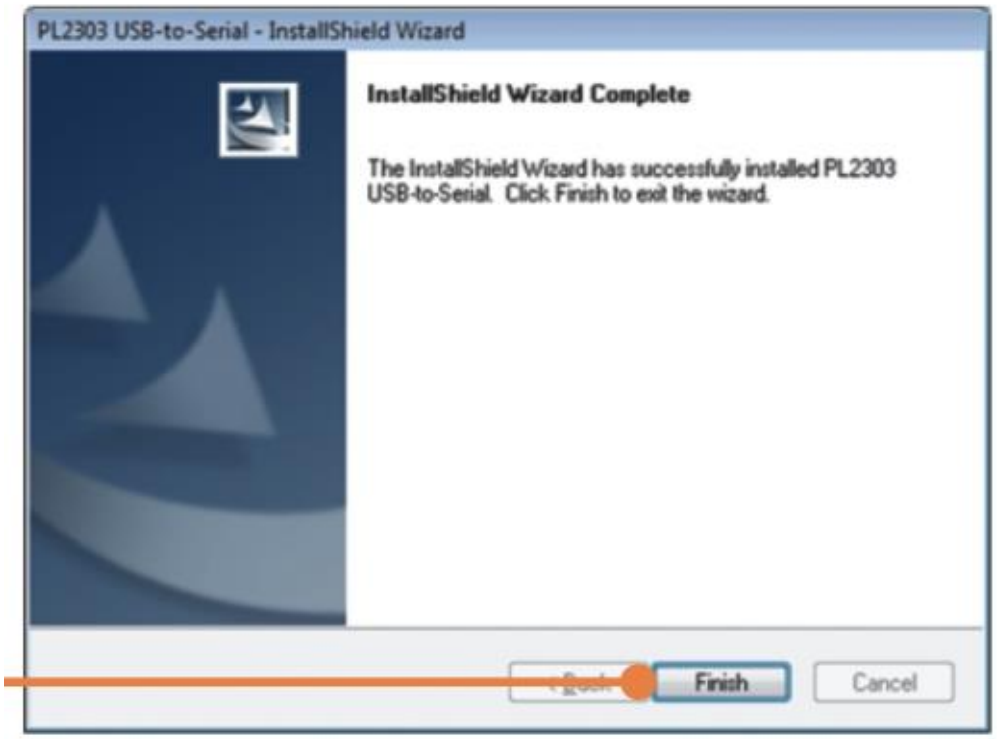
Haga clic en “Install Driver”



Haga clic en Next



Haga clic en Finish



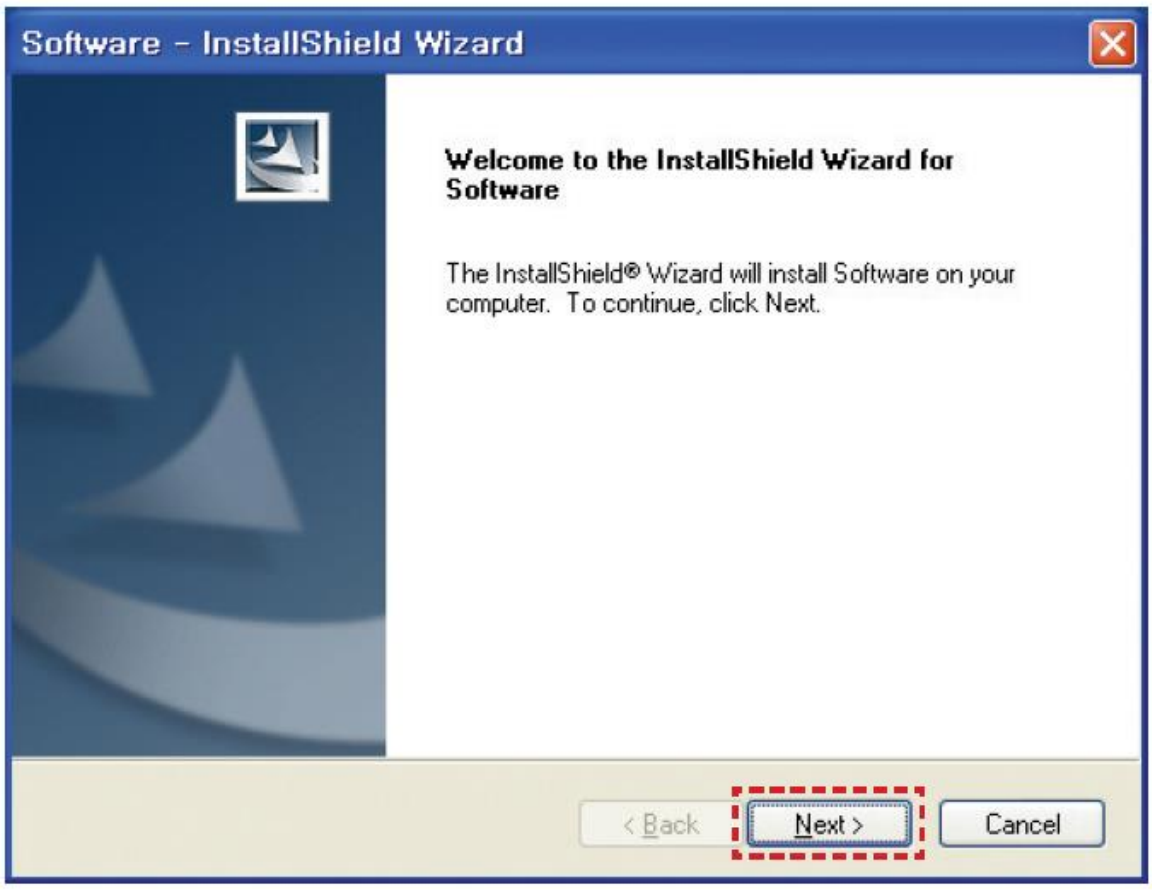
Conecte el adaptador a un puerto USB, los drivers se instalaran automaticamente

Anexo D

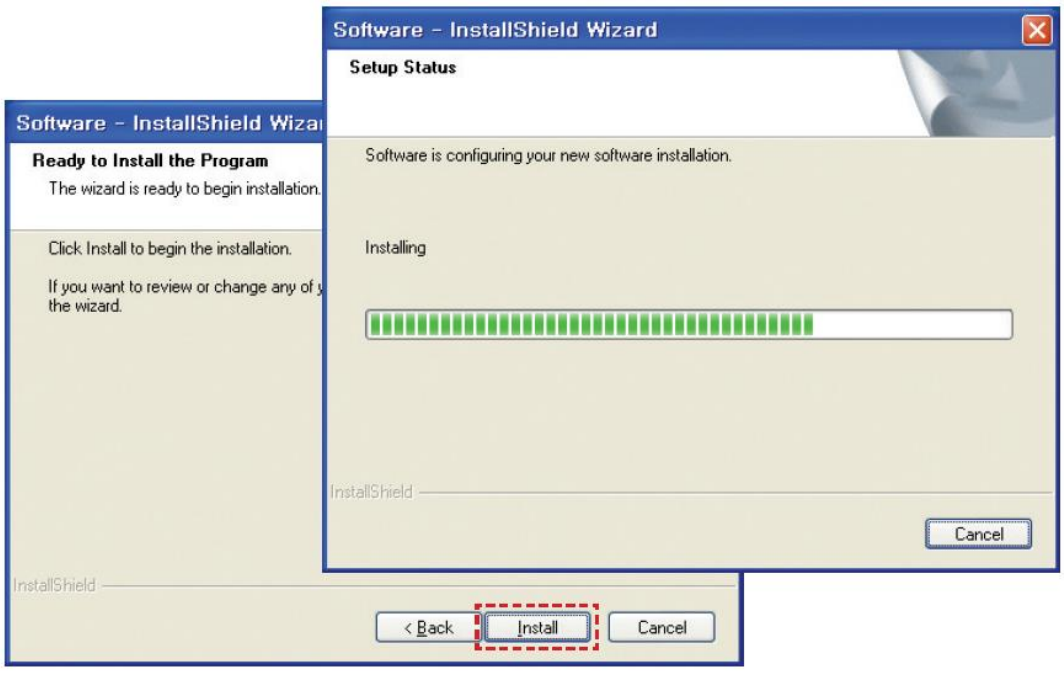
Instalación del Software para la plataforma robótica educativa.

Para que la plataforma robótica educativa pueda ser utilizada se debe instalar el software que incluye el kit.

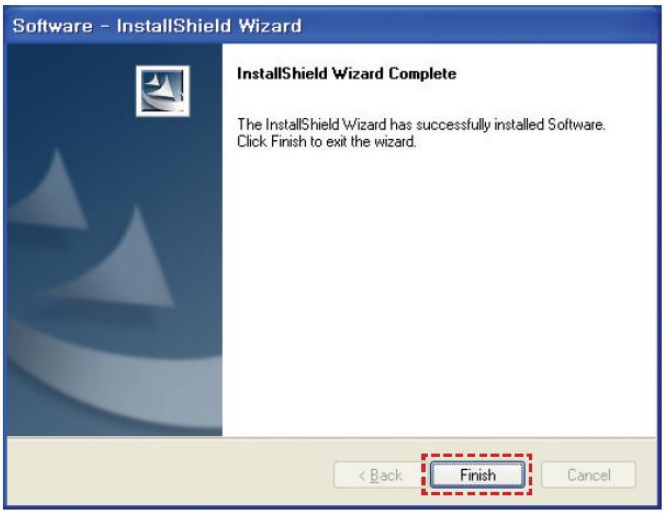
Ejecutar el archivo Autorun.exe del cd de instalación, y clic en Next



Clic en Install para iniciar el programa de instalación



Cuando la instalación haya terminado hacer clic en Finish



Anexo E:

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

SEDE AMBATO

Encuesta De Satisfacción

Objetivo: Identificar el grado de satisfacción de los usuarios al utilizar una Interfaz Natural de Usuario, para la interacción con la plataforma robótica educativa RoboBuilder.

Instrucciones: Marque con una X del 1 al 5 según sea el caso.

Pregunta	***1 Deficiente***	***2 Regular***	***3 Malo***	***4 Bueno***	***5 Muy Bueno***
Al mover un solo brazo, ¿Cuál ha sido el grado de respuesta?					
Al mover los dos brazo, ¿Cuál ha sido el grado de respuesta?					
La posición de las extremidades del robot en qué grado se asemeja a la posición de sus extremidades.					
¿Cuál es el grado de dificultad al momento de utilizar el sistema para interactuar con la plataforma robótica. ?					