

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**PROGRAMACIÓN DE ÓRDENES A UN ROBOT LEGO
MEDIANTE INTERFAZ NATURAL**

**LEONEL ESTEBAN MEZA CASTRO
PATRICIA ALEJANDRA MUÑOZ PÉREZ**

NOVIEMBRE 2013

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**PROGRAMACIÓN DE ÓRDENES A UN ROBOT LEGO
MEDIANTE INTERFAZ NATURAL**

**LEONEL ESTEBAN MEZA CASTRO
PATRICIA ALEJANDRA MUÑOZ PÉREZ**

INFORME FINAL
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

NOVIEMBRE 2013

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

PROGRAMACIÓN DE ÓRDENES A UN ROBOT LEGO MEDIANTE INTERFAZ NATURAL

**LEONEL ESTEBAN MEZA CASTRO
PATRICIA ALEJANDRA MUÑOZ PÉREZ**

Profesor Guía: Iván Mercado Bermúdez
Profesor Co-referente: Wenceslao Palma Muñoz

Carrera: Ingeniería en Ejecución Informática

NOVIEMBRE 2013

“Agradezco en especial a mis padres, amigos, profesores y compañero de proyecto. Principalmente a mis padres, quienes durante toda mi etapa de formación profesional me brindaron todo su apoyo, sacrificio y amor estando siempre presentes.”

Patricia Alejandra Muñoz Pérez

“A mis padres, mi profesor guía, mis amigos quienes con su apoyo he podido sacar adelante la carrera, mis padres siempre han estado ahí presentes para apoyarme en todo lo que siempre necesite y nunca me dejaron solo. A mi compañera de proyecto, por compartir su sabiduría, sus conocimientos y así poder sacar adelante nuestro proyecto.”

Leonel Esteban Meza Castro

Resumen

Para comenzar este proyecto se realiza en base a dos estudios, Robótica e Interfaz Natural (Kinect). Durante el desarrollo del proyecto se analizaron las herramientas a utilizar como también su funcionamiento. Se descartó el primer caso de estudio que era el Robophilo, debido a problemas de escasa documentación y problemas con la parte interna del robot, por ello se escogió como segundo caso de estudio, ocupar Lego Mindstorms, porque es mucho más práctico para el tema de la programación. Para continuar con el estudio, se realizaron mini-prototipos tanto para Legos NXT y Kinect. En la sección de diseño se construyó un brazo robótico, que consta de tres servos que proporcionan los movimientos necesarios para simular los movimientos de un brazo humano. Seguidamente, para el caso de estudio de la Kinect, se probaron los SDK disponibles; Kinect for Windows y OpenNi, con ambas herramientas se desarrollaron ciertas pruebas de comparación. Finalmente se procedió a diseñar e implementar los prototipos que controlan el robot y Kinect, para este último se desarrolló un software que permite la captura y almacenamientos de los movimientos humanos, generando ciertas órdenes que serán implementadas por el robot, siendo este el objetivo final del proyecto.

Palabras Claves: Kinect, servo, sensores, manipulador, Lego Mindstorm, Lejos, NXT

Abstract

The project is based on two studies: on robotics and on the Natural Interface (Kinect). While the project was being developed, the tools to use were analyzed. The first study case (Robophilo) was discarded due to vague documentation issues and problems in the firmware. Then, Lego Mindstorm was chosen as the second case to use, because it is more practical for programming purposes and it was considered that there is a large amount of information about this tool. To continue this investigation, we created mini-prototypes of Lego NXT. We developed a robotic arm that was formed by three servos, giving the essentials movements to simulate an human arm; later in the case of the natural interface study, we realized a research of all their SDK available that a Kinect has, we did certain tests with the kits that were available, and compared the use of the Microsoft's kit and OpenNi. Finally we designed and implemented the prototypes that control the robot, and with the Mockups, we created the interfaces to define the presentation of the prototype.

Key Words: Kinect, servo, sensor, manipulator, Lego Mindstorm, Lejos, NXT

Índice

Resumen	iii
1. Introducción	1
1.1 Motivación	2
1.2 Objetivos	3
1.2.1Objetivo General.....	3
1.2.2Objetivos Específicos	3
2. Planificación	4
2.1 Plan de Trabajo	4
3. Análisis de Riesgo	5
4. Estudio de Factibilidad	7
4.1Factibilidad Técnica	7
4.1.1 Kit de desarrollos disponibles.....	7
4.1.2 Hardware para producción.....	7
4.2 Factibilidad Operativa	8
4.3 Factibilidad Jurídica	8
5. Estado del Arte	9
5.1 Estudio de Lego Mindstorm	9
5.1.1 Componentes del Lego Mindstorm	9
5.2 Estudio de la Kinect	10
5.2.1Componentes de la Kinect.....	11
5.3 SDK Disponibles para Kinect	12
5.3.1 El SDK de Kinect for Windows	12
5.3.2 OpenNI/Sensor Kinect.....	13
5.3.3 Comparación entre Kinect for Windows SDK / OpenNI	14
6. Diseño de la Solución	15
6.1 Modelo de Proceso	15
6.2 Elección de las Herramientas	15
6.3 Descripción de las Principales Clases y sus Métodos	16
6.3.1 Clase Interfaz	16
6.3.2 Clase Tracker	17
6.3.3 Clase Global.....	18
6.4 Diseño del Prototipo del Software Kinect	18
6.4.1 Prototipo Kinect Versión 1.	18

6.5 Diseño de Prototipos Robot Lego Mindstorm	19
6.5.1 Modelo Prototipo Versión 1	19
6.5.2 Modelo Prototipo Versión 2	19
6.5.3 Modelo Prototipo Versión 3	20
6.5.4 Modelo Prototipo Versión 4	20
6.5.5 Modelo Prototipo Versión Final	20
6.6 Prototipo de Software Diseñado para NXT	21
6.6.1 Prototipo de Software en “JAVA”	21
6.6.2 Prototipo de Software en “C#”	22
7. Pruebas de Prototipos	23
7.1 Prueba de Prototipo Consola	23
7.2 Pruebas de Prototipos en Windows Form	23
7.3 Prueba de Prototipo Final	24
7.4 Prueba de Prototipo Final: Funcionalidad específica	24
8. Conclusión	25
9. Referencias	26
ANEXOS	27

Lista de Figuras

Figura 1 Muestra el Software de Diseño de Lego	27
Figura 2 Muestra Primer prototipo de Robot	27
Figura 3 Muestra motor central del Primer prototipo de Robot	28
Figura 4 Vista posterior del Primer prototipo de Lego.....	28
Figura 5 Brazo del Primer prototipo de Lego.....	29
Figura 6 Prototipo 2 de Lego.....	29
Figura 7 Vista Frontal Segundo Prototipo de Lego.....	30
Figura 8 Tercer Prototipo de Lego	30
Figura 9 Tercer Prototipo de Lego, vista trasera	31
Figura 10 Prototipo Final de Robot Lego.....	31
Figura 11 Formato de texto final	32
Figura 12 Formato de texto tentativo	32
Figura 13 Interfaz prototipo tentativo de Kinect	33
Figura 14 Datos entregados por Primer Prototipo de Consola de Kinect.....	33
Figura 15 Datos entregados por Segundo Prototipo de Consola de Kinect	34
Figura 16 Primer Prototipo en Windows Form de Kinect.....	34
Figura 17 Prototipo Final de Software, pestaña Captura de movimientos	35
Figura 18 Prototipo Final Software, pestaña Ejecución de órdenes	35

Lista de Tablas

Tabla 2 Muestra los valores de la tabla de riesgos	5
Tabla 3 Muestra el estudio de los riesgos.....	5
Tabla 4 Tabla de Mitigación de Riesgos	6

1. Introducción

La informática forma parte del estudio y desarrollo de la robótica como también lo es la mecánica y la electrónica. Además la informática es parte importante de muchas de las tecnologías actuales tales como, celulares, computadores, videojuegos, televisores etc. El objetivo de este estudio es utilizar dos tecnologías directamente relacionadas con la informática; que son la robótica y la interfaz natural.

Estas herramientas han tenido una gran aceptación y accesibilidad de los usuarios durante estos últimos años, esto es debido a que hoy en día los avances de la robótica han permitido que se fabriquen robots cada vez más económicos para el bolsillo de las personas; en el caso de la interfaz natural: Kinect es un dispositivo para videojuegos que se dio conocer en el año 2009 como un mando de la consola Xbox 360, con la principal capacidad de utilizar NUI(*natural user interface*) para poder jugar, la característica más importante es que se puede jugar sin ocupar un joystick, y que en año 2011 fue liberado oficialmente el SDK(*software development kit*) para el desarrollo de aplicaciones para PC por medio de Windows 7 y 8.

En primer lugar, es dar a conocer el propósito o idea principal, en la sección siguiente de motivación; indicando el grado de conocimiento que se pretende lograr con el proyecto. Luego se procederá a plantear el objetivo general y los objetivos específicos. En la unidad siguiente se dará a conocer la metodología de trabajo, para los casos de estudios se necesitará implementar un plan de trabajo, el cual será útil para manejar los tiempos y tareas correspondientes. Además se planteará el plan de riesgo del proyecto, representados en una tabla que mide el impacto y la probabilidad del riesgo, seguidamente del estudio de factibilidad del proyecto. Por último se desarrollará el estudio de las herramientas; Robótica y la Kinect. Dando énfasis a los diseños de prototipos, seguido de las pruebas adecuadas para verificar que se logre los objetivos planteados. Para el caso del presente estudio, se necesita de mucha recopilación de información, análisis, observación y pruebas de ensayo y error. A continuación se da a conocer al lector la motivación del proyecto.

1.1 Motivación

La oportunidad de realizar un proyecto que abarque el estudio del uso de una tecnología relativamente nueva, Kinect, la que fue lanzada al mercado en el año 2009 por Microsoft. El origen de esta herramienta radica en el concepto de interacción, que habla de todos los intercambios entre persona y computador (*Beacker y Buxton, 1987*) para cumplir tareas específicas [1].

Los avances de los videojuegos permitieron el desarrollo de este dispositivo, ya que era imprescindible crear una nueva manera de jugar. El impacto que tuvo la Kinect a nivel mundial fue muy grande, ya que fue muy superior a la competencia y además tuvo una gran aceptación de la gente. Cabe destacar, que primero fue creada para la consola Xbox360, y que posteriormente en el año 2011 se liberaron los drivers del dispositivo para desarrollar software con él, no tan sólo fue aprobada como algo comercial sino que da la oportunidad a muchos programadores e investigadores poder explotar su potencial.

Otra tecnología que se ocupará para el presente estudio es la robótica. La palabra robot es de origen checo y proviene del término *robot*, que significa trabajo forzado, y *Robonick* que es esclavo, término que aparece en el año 1921. En la parte de la robótica existen varias interrogantes, una de ellas es ¿Pueden los robots mejorar la vida de las personas?, esta interrogante apareció y se planteó desde la aparición de las máquinas con aspecto humano, hace bastante tiempo.

Actualmente los avances de la robótica se reflejan en la medicina, la parte doméstica, la agricultura, en ayuda a las personas con problemas especiales, en la parte educativa etc. Dando respuesta a esta interrogante que los robots si pueden mejorar la vida de la persona cubriendo diferentes necesidades. Por ello existen diversos tipos de robots, como lo son los robots con características móviles, humanas, y manipulables. En este estudio se intentará combinar características de robots humanoides e industriales, es decir, un robot con cualidades humanoide que se pueda programar en base a un método de programación industrial.

La programación de robots industriales, utiliza dos técnicas que son la programación gestual y programación textual. Se dará énfasis a la programación gestual, la cual se subdivide en dos clases, programación por aprendizaje directo y programación por medio de un dispositivo de enseñanza, siendo la última, la técnica a utilizar en este proyecto.

En la actualidad en el área de investigación, los trabajos que se han hecho con el dispositivo Kinect han sido con fines académicos, educativos y hasta de ocio. Algunos proyectos relacionados son: “El guiado gestual de un robot humanoide mediante Kinect”, desarrollado el 2011, como un proyecto de tesis en la facultad de Barcelona, el cual se utiliza un robot humanoide llamado Bioloid, este proyecto logra mostrar el manejo directo entre el robot y la Kinect [2]. Los otros proyectos se basan en noticias de la robótica actual, en febrero de este año 2012, salió el proyecto Kinect te ayuda a estacionar, esta aplicación se desarrolló en la Universidad de Sídney, el objetivo es asistir a los conductores al momento de estacionarse, donde la Kinect está instalada en la parte posterior del vehículo, y los distintos sensores realizaban el trabajo.

También salió el perro robot para ciegos con Kinect, este fue diseñado por la compañía de Japón NSK, se señala que la persona toma el mango del perro y lo presiona, y ahí es cuando el robot avanza y da indicaciones de voz de su entorno.

Finalmente es claro el desarrollo de estas aplicaciones, y no deja de sorprender a los usuarios, por lo que se seguirá empleando el uso de la Kinect como herramienta para nuevos proyectos. Ya teniendo claro el estudio que se llevará a cabo, se podrá definir la principal problemática que abarcará el proyecto, esta será poder realizar un software que utilice Kinect como dispositivo de enseñanza. Este será capaz de capturar y almacenar las instrucciones, que luego serán interpretadas de forma asíncrona por el robot.

1.2 Objetivos

1.2.1 Objetivo General

Implementar un prototipo de software que utilice Kinect (interfaz natural), para programar una secuencia de movimientos que pueda ser almacenada y ejecutada por un lego robot.

1.2.2 Objetivos Específicos

Una vez definido el objetivo general, del estudio, éste se descompone de los siguientes objetivos específicos:

- Crear un prototipo de software, implementado con Kinect, que permita la captura y almacenamiento de movimientos de un brazo humano.
- Crear un prototipo de software, utilizando Lego Robot, que permita la ejecución de las instrucciones almacenadas.
- Realizar pruebas validación de los prototipos, para el brazo robótico y el software de la Kinect. Ambos prototipos se incorporan al caso real de estudio, que es dibujar sobre una superficie plana.

2. Planificación

2.1 Plan de Trabajo

En consecuencia de los objetivos planteados anteriormente la realización del proyecto fue dividido en cinco etapas:

Etapas de análisis y recopilación de información: En esta etapa se seleccionará la información dependiendo del grado de importancia. El criterio a utilizar es basarse en los objetivos planteados del proyecto, para ello se realizará un análisis de toda la información, destacando ciertas fuentes de acuerdo al tema elegido.

Etapas de estudio y resumen de información: Esta etapa es consecutiva a la anterior, una vez ya analizado y recopilado la información a utilizar, se realizará un estudio de las herramientas. Es importante destacar que los proyectistas deberán utilizar sus capacidades intelectuales para el entendimiento de términos específicos, apoyándose ya sea libros, internet o personas conocedoras del tema. Luego se realizarán resúmenes, para la redacción del estudio.

Etapas de diseño de prototipos e implementación: Esta etapa se basa en el método evolutivo, por lo que los prototipos son susceptibles a constantes cambios, con lo cual se diseñarán distintos mini-prototipos para lograr un prototipo final estable y bien diseñado.

Etapas de pruebas: Una vez diseñado los mini-prototipo a utilizar, se deberá realizar ciertas pruebas de validación, esto permite asegurar que lo que estamos construyendo se ha lo correcto. Esta etapa se caracterizará por ser flexible, debido a que el método de implementación es de ensayo y error.

Etapas de documentación: Una vez ya planteado el problema a desarrollar y teniendo las ideas ya claras, se realizará la tarea de documentar. La documentación consta de dos partes; la primera es plasmar los estudios realizados en base a la Robótica y Kinect y la segunda parte es plasmar el diseño de prototipos y pruebas ya realizadas.

3. Análisis de Riesgo

IMPACTO	Alto	4	5	5
	Medio	3	3	5
	Bajo	1	2	4
		Bajo	Medio	Alto
PROBABILIDAD				

Tabla 1 Muestra los valores de la tabla de riesgos

De acuerdo a los análisis y estudios anteriores, se presentan una serie de riesgos para el desarrollo del proyecto. Estos riesgos implican tanto al desarrollador, como las herramientas y desarrollo del proyecto. Para valorizar los riesgos, se le asignara una calificación dentro de una escala de 5 valores: insignificante (1), bajo (2), medio (3), moderado (4) y alto (5). Se propone la siguiente tabla para valorizar los riesgos. La calificación dependerá de la combinación del nivel de impacto y probabilidad del riesgo. A continuación, a través de la siguiente tabla, se enumeraran los riesgos encontrados. A estos riesgos se les asignara una o varias categorías y se calificarán sus probabilidades e impactos en una escala de bajo, medio y alto.

Riesgo	Categoría	Probabilidad	Impacto	Calificación
Incompatibilidad de las tecnologías a ocupar.(a)	Tecnológico	Baja	Alto	Moderado
Inconsistencia en el estudio de las tecnologías.(b)	Investigación	Baja	Alto	Moderado
Manejo inestable de los tiempos.(c)	Investigación	Media	Media	Media
Fallas de Hardware y pérdida de datos.(d)	Tecnológico	Media	Alta	Alto
El sistema no cumpla con los objetivos planteados.(e)	Diseño	Baja	Alto	Moderado
Problemas técnicos con el dispositivo de captura y Robophilo.(f)	Tecnológico	Bajo	Alto	Moderado
Diseño de la creación de mini-prototipos.(g)	Tecnológico	Bajo	Media	Bajo
Problemas técnicos con el Hardware de la Kinect.(h)	Tecnológico	Alta	Medio	Alto

Tabla 2 Muestra el estudio de los riesgos

La tabla 4 representa el estudio de los riesgos del proyecto, donde el único riesgos más Altos fue D. dado que la falla del hardware dentro de las tecnologías a utilizar, puede resultar catastrófico, a la hora de perder algún dato; también se reflejan los riegos moderados que son A, B, E, F, G, H esta calificación no es menor que la anterior, por el ejemplo una mala interpretación o estudio puede causar demora y un esfuerzo mayor para los desarrollares del sistema. Según el estudio no se presenta ninguna probabilidad de riesgo baja, debido que las situaciones de riesgo son altamente considerables.

Se sabe que luego de mediar algún riesgo, se debe mitigarlo o tratar de prevenirlo la siguiente tabla representa la mitigación de los riesgos ya antes analizados.

Riesgo	Calificación	Medidas de Mitigación
Incompatibilidad de las tecnologías a ocupar(a).	Moderado	Realizar un estudio de las tecnologías, tener un plan b en el caso de que alguna falle
Inconsistencia en el estudio de las tecnologías (b).	Moderado	Realizar un esfuerzo mayor, ya sea en recopilación de información, verificar las fuentes confiables.
Manejo inestable de los tiempos (c).	Media	Construcción de una carta Gantt, Registrar los hitos del proyecto.
Fallas de Hardware y pérdida de datos (d).	Alto	Realizar respaldo de la información, en varias máquinas de uso.
El sistema no cumpla con los objetivos planteados (e).	Moderado	Seguir con la planificación, y estar preparados para posibles cambios dentro del proyecto
Problemas técnicos con el dispositivo de captura y Robophilo (f).	Moderado	Probar los dispositivos, asegurándonos que funcionan de manera correcta.
Diseño de la creación de mini-prototipos (g).	Moderado	Se deben diseñar varias pruebas antes de definir el prototipo final
Problemas técnicos con el hardware de la Kinect (h).	Moderado	Se debe tener en cuenta la posición del dispositivo y la luz ambiental

Tabla 3 Tabla de Mitigación de Riesgos

4. Estudio de Factibilidad

Después de presentar el objetivo del prototipo de diseño y los riesgos que puedan presentar el proyecto, es pertinente realizar un estudio de factibilidad. Con este estudio de factibilidad se determinaran los requerimientos técnicos, económicos y operativos para el desarrollo del sistema y las posibles limitantes legales que podrían ser críticos para el desarrollo del proyecto.

4.1 Factibilidad Técnica

La factibilidad técnica consistió en recopilar información sobre las posibles herramientas que se usaran para la construcción del prototipo, así como la evaluación de los dispositivos de hardware implicados en la ejecución de los mini-prototipos.

4.1.1 Kit de desarrollos disponibles

El SDK de Microsoft para su dispositivo Kinect que incluyen librerías .NET para el manejo de los sensores de la misma, estas librerías pueden ser usadas con C++ y C# tanto en aplicaciones en plataforma Windows o aplicaciones basadas en XNA (juegos para Windows y Xbox360). Consume más CPU que OpenNI/NITE. El SDK OpenNI de PrimeSense, para todas las plataformas (Windows, Linux y Mac OSX) en lenguaje C/C++.

De acuerdo al estudio realizado al él SDK de Microsoft presenta pequeñas limitaciones pero que no influyen a proceso del proyecto, para asegurarse del funcionamiento se realizaron ciertas pruebas, en base a la herramientas que trae, como el caso de la herramienta de rastreo el esqueleto tracking, al igual que el SDK el de OpenNI. Sin embargo ambos debiesen suplir las necesidades básicas de interacción con los sensores de movimiento y captura de profundidad. Como requerimiento de las herramientas, se necesita una detección de las posiciones de las extremidades (superiores e inferiores). Ambas herramientas de desarrollo suplen esta necesidad (*body tracking*).

4.1.2 Hardware para producción

El hardware disponible para la producción del prototipo debe soportar Windows 7 más las herramientas de desarrollo y prueba de los respectivos kit de desarrollo. Para ello se considera como prestaciones mínimas en la máquina de desarrollo:

- Procesador Intel de doble núcleo de 2,3 GHz mínimo.
- Memoria RAM de 2GB mínimo.
- Tarjeta de video de dedicada de 512 MB mínimo.

Con estas características se asegura un ambiente de desarrollo y de prueba estable según los requerimientos de los SDK. Además se necesitará para la conexión entre el PC y el prototipo del robot, utilizar un adaptador de USB Bluetooth, este permite a su computador de escritorio o portátil conectarse sin cables a cualquier dispositivo Bluetooth. Permite a los equipos y dispositivos un rango de trabajo de más de 10 metros con una transferencia de 2.1 Kbps en velocidad de datos. Cumple con Bluetooth 2.0 y especificaciones 2.0. Con este adaptador será posible conectar el robot Lego con el PC que contendrá las instrucciones desde el sensor Kinect.

4.2 Factibilidad Operativa

Dado como se planteó el estudio de los kit de la Kinect, esta posee la detección de todas las extremidades del cuerpo para el desarrollo del sistema, bajo ningún punto de vista debiese ser un impedimento la falta de alguna extremidad para el uso de la aplicación. De acuerdo a lo investigado, los SDKs no presentan (de momento) problemas con la falta de alguna extremidad para la detección del cuerpo. Así pues, sin importar la condición física del usuario, este será capaz de usar la aplicación.

Además la creación de mini-prototipos Legos, no presentan ningún riesgo en base a las piezas que se presenta, no existiría ningún daño físico, son piezas netamente inofensivas, por lo que no impide la creación de los prototipos, pudiéndose llevar a cabo el diseño de la solución.

4.3 Factibilidad Jurídica

Dado que se usaran solo herramientas libres y gratuitas, no habría problemas por uso de programas reproducidos ilegalmente (piratería). Esto debido a que las herramientas nombradas son oficiales y en el caso de OpenNI, es de código abierto para la comunidad.

También para el caso de las herramientas de LEJOS NXT, posee librería abierta la cual queda libre de todo tema judicial, por lo que el uso de los legos se ve abierto por toda la comunidad investigativa o para personas que desean aprender por sí sola.

Con el estudio realizado en base a la factibilidad se puede concluir que el proyecto sí es factible, debido a que la parte técnica y operativa no presenta complicaciones a la hora de realizar el proyecto, ya que las herramientas y tecnologías están al alcance para hacer uso del desarrollo.

5. Estado del Arte

5.1 Estudio de Lego Mindstorm

Lego Mindstorm es un juego de robótica para niños creado por la empresa Lego, el cual posee elementos básicos para crear robots; como la capacidad de poder unir piezas, y la programación de acciones en forma interactiva. Este puede ser usado para construir un modelo de sistema integrado con partes electromecánicas controladas por un computador. Prácticamente se puede construir de todo con las piezas disponibles desde robots humanoides hasta robots móviles.

La forma de programar el NXT (procesador) es a través de un software intuitivo de Lego en el que solamente hay que tomar y arrastrar acciones a una secuencia de movimiento. La otra forma de programación es a través del lenguaje Java, implementada para programadores expertos por ser un lenguaje de alto nivel. Otro lenguaje utilizado es C#, también es un lenguaje de alto nivel, este tiene una biblioteca llamada NXT.NET la cual permite crear aplicaciones para Windows que permiten controlar el Robot y además da la posibilidad de ocupar la conexión Bluetooth para establecer la comunicación entre el PC y el NXT.

El funcionamiento de Lego Mindstorm es similar al cuerpo humano, porque primero posee los sensores que recogen la información para la cual fueron creados (sensor de sonido, escucha; el de ultra sonido puede ver; el de tacto, siente; etc.), una vez que los sensores reciben la información, se procede a enviar instrucciones al procesador NXT, esto es a partir de la programación que se tenga, luego se toman decisiones a partir de movimiento que serán enviados a los distintos servos.

5.1.1 Componentes del Lego Mindstorm

Lego Mindstorm posee una cantidad de piezas para armar, pero existen ciertos componentes básicos que posee entre ellos tenemos [3]:

- El NXT es un micro computador inteligente, el cual puede ser programado para recibir información de los sensores y activar los servos-motores.
- El Sensor de Tacto es un interruptor que puede ser presionado y soltado, al ser presionado envía una señal al NXT en forma de 0 y 1, si el sensor no está presionado es 0, si lo presionan es 1.
- El Sensor de Sonido, como su nombre lo menciona, detecta los niveles del sonido en decibeles (dB), es decir, la cual mide la suavidad o la intensidad.
- El Sensor de Luz da al robot la capacidad de distinguir entre la luz y la oscuridad, con esto puede ver la intensidad de luz en una habitación o medir la intensidad de luz en superficies de colores. El sensor de luz convierte los colores a escala de grises para así poder distinguir qué color se está viendo.
- El Sensor de Ultrasonido incorporado al robot, tiene la habilidad de ver y poder reconocer objetos, evadir obstáculos, medir distancias y detectar movimiento.
- Los Servo Motores proporcionan al robot la capacidad de moverse. Una vez que el robot este en movimiento. Para el caso de que un servo se mueva en otra dirección, los

demás motores alinean su velocidad automáticamente, para que el robot se mueva sin problemas.

- El Sensor de Rotación, los servos tienen un sensor de rotación integrado. La retroalimentación de la rotación permite que el NXT pueda controlar los movimientos con mucha precisión. El sensor de rotación integrado mide la rotación del motor en grado (precisión de $\pm 1^\circ$) o rotaciones completas.

5.2 Estudio de la Kinect

Cuando se habla de interacción, se habla de todos los intercambios entre persona y computador [Beacker y Buxton, 1987] para cumplir tareas específicas. Estas interacciones han evolucionado durante toda la era de la computación; comenzando con la interacción más básica por medio de líneas de comandos, donde el usuario ingresa a la computadora ordenes predefinidas por medio de teclado. Luego evoluciono a la manipulación directa, donde las instrucciones por órdenes de comandos se reemplazan por la manipulación de objetos y acciones por medio de interfaces WIMP (Windows, Icons, Menús and Pointers). Finalmente se tiene la interacción natural que por medio del lenguaje o acciones naturales (reconocimiento de patrones, reconocimiento de voz, interacción *multitouch*) se puede dar órdenes a un computador [1].

Actualmente dentro del mundo de los videojuegos existen tres grandes empresas que han desarrollado consolas de alta tecnología, estas son: Nintendo, Microsoft y Sony. En los últimos años, estos tres fabricantes han tenido una dura lucha para crear la consola con las mejores características tanto gráficas, como en la jugabilidad. Nintendo fue el primero que tomo la delantera, lanzando la consola Wii el año 2006. En un intento por superar la competencia, Sony lanzó Play Station Move en septiembre del 2010; este dispositivo es un mando para la consola Play Station 3 que ocupa sensores de movimiento y una esfera en su extremo superior que se ilumina. Luego Microsoft no se quedó atrás, en Noviembre de ese mismo año llegó al mercado un dispositivo “Kinect”, este novedoso sistema no requiere controles que estén en contacto con la persona (“*joystick*”) para dirigir a los personajes de la pantalla, sino que ocupa como control el cuerpo de la persona que está jugando [5].

Este dispositivo proyecta una luz infrarroja la cual se muestra la profundidad con tonalidad de color amarillo, luego está es procesada por el chip PS1080-A2 el procede a descomponer la imagen en los parámetros necesarios para su tratamiento, es decir, se centra en la búsqueda de formas similares a las del cuerpo humano y el cálculo de cómo se podrán mover dentro del entorno; tomando en cuenta las zonas con las que puede colisionar y los movimientos que puede realizar. Las cámaras de la Kinect poseen un sistema de detección que les permite identificar a seis personas que estén en el campo de visión, pero cuando tiene que trazar los esqueletos (“*Skeletal tracking*”), solo puede realizar esta operación con dos personas. Además la Kinect posee un micrófono estéreo que le permite captar conversaciones en línea y reconocer comandos de voz. El micrófono que tiene la Kinect es un micrófono Multiarray, es decir, son muchos micrófonos colocados en fila alrededor de una superficie, captando todos ellos el sonido que les llega en todas las direcciones [8].

5.2.1 Componentes de la Kinect

En la sección anterior se señaló el funcionamiento de la Kinect, pero para que realice las funciones se necesita tener en cuenta cuales son los componentes más importantes de esta. Algunos de ellos son controladores, procesadores de imágenes, sensores entre otros [11].

- **Wolfson Stereo ADC WM8737G: (rojo):** Conversor analógico digital de sonido estéreo de baja potencia diseñado para aplicaciones portables.
 - **Fairchild N-Channel PowerTrench MOSFET (naranja):** MOSFET de N-canales diseñado especialmente para mejorar la eficiencia total de convertidores DC/DC usando tanto síncronos como controladores convencionales de interconectores PWM. Se optimizo para puertas de baja carga, baja rDS y alta velocidad de conexión.
 - **NEC uPD720114 (amarillo):** Controlador Hub USB2.0
 - **H1026567 XBOX1001 X851716-005 GEPP (verde):** Tarea Desconocida.
 - **Marvell AP102 (azul):** SoC con controlador para cámara Hynix H5PS5162FF 512 megabit DDR2 SDRAM.
 - **Hynix H5PS5162FF 512 megabit DDR2 SDRAM (rosa):** Lista de componentes de izquierda a derecha figura 25:
 - **AnalogDevices AD8694 (rojo):** Salida operacional amplificadora CMOS rail-to-rail simple, dual o cuadrática de bajo coste y bajo ruido.
 - **TI ADS7830I (naranja):** 8-bit, 8-channel Conversor analógico-digital por muestreo con interfaz I2C.
 - **Allegro Microsystems A3906 (amarillo):** Controlador de motor DC simple/doble paso a paso de bajo voltaje.
 - **ST Microelectronics M29W800DB:** 8 Mbit (1Mb x8 or 512Kb x16) Memoria flash NV.
 - **PrimeSense PS1080-A2 (azul):** SoC sensor procesador de imágenes.
- Las Cámaras y cañón infrarrojo de la Kinect.
- **Izquierda:** (IR CMOS) Microsoft / X853750001 / VCA379C7130. Resolución: 320x240.
 - **Central:** (Color CMOS) VNA38209015. Resolución 640x480.
 - **Derecha:** (IR Proyector) OG12 / 0956 / D306 / JG05A.

5.3 SDK Disponibles para Kinect

5.3.1 El SDK de Kinect for Windows

SDK oficial de Microsoft para crear aplicaciones con Kinect sobre la Plataforma Windows 7 y Windows Embedded Standard 7, actualmente en su versión 1.5 lanzada en mayo del 2012, presenta mejoras en el seguimiento esquelético, se agregaron nuevos idiomas al reconocimiento de voz tales como español, francés, italiano, entre otros; se agregaron nuevas herramientas para hacer el desarrollo de aplicaciones más rápido. Además se mejoró la sincronización entre las imágenes de color y profundidad y se agregó el SDK para el rastreo de la cara. Se presenta las siguientes características [1]:

- Soporte para lenguajes C++, C#, o Visual Basic (Utilizando Microsoft Visual Studio 2010).
- Provee herramientas de ayuda y ejemplos para hacer que el desarrollo más rápido, además de proporcionar el código fuente de estos.
- Mejoras en el Rastreo esquelético (Skeleton Tracking).
- Permite el uso del sensor de profundidad para detectar objetos hasta un mínimo de 40 centímetros de distancia, entregando más información sobre los valores de profundidad detectados.
- Provee componentes para reconocimiento de voz más exacto.

Requerimientos Hardware:

- Procesador 32-bit (x86) o 64-bit (x64).
- Procesador Dual-core 2.66 GHz o superior.
- USB 2.0 dedicado.
- 2 GB RAM

Requerimientos Sistema Operativo:

- Windows 7 ó Windows Embedded Standard 7

Librería de código libre creada a partir de la utilización de ingeniería inversa sobre la Kinect a los 3 días de su lanzamiento, actualmente es mantenida por una comunidad abierta de usuarios llamada OpenKinect. Esta librería provee los drivers para el acceso a la cámara USB del sensor Kinect permitiendo procesar la información captada por esta:

- RGB e Imágenes de Profundidad
- Motores
- Acelerómetro
- LED.
- Actualmente no soporta audio.

Libfreenect posee interfaces (Wrappers) para múltiples lenguajes:

- C (using a synchronous API)
- python
- actionscript
- C#
- Java (JNA)
- Lisp
- C++

5.3.2 OpenNI/Sensor Kinect

Framework basado en *OpenNI (Open Natural Interaction)*, es una organización sin fines de lucro enfocada en el desarrollo de tecnologías para la interacción natural con dispositivos, uno de los principales participantes es Prime Sense), que provee una infraestructura genérica basada en API's de código abierto para acceder a los dispositivos de interacción natural, permitiendo acceder específicamente a los servicios provistos por el sensor Kinect. Adicionalmente Prime Sense desarrolla el *middleware* NITE el cual no es de código abierto y permite acceder a funcionalidades avanzadas como seguimiento del esqueleto en tiempo real y reconocimiento de gestos [1].

SensorKinect utiliza el driver avin2 para conectar el sensor Kinect para utilizarlo con el framework provisto por OpenNI, permitiendo acceder a distintos módulos:

- Sensores 3D (en algunos dispositivos)
- Cámaras RGB
- Cámaras IR (Infrarrojas)
- Dispositivos de Audio

Soporta los siguientes lenguajes:

- C++
- C#
- Python
- Visual Basic
- Además el sensor Kinect se puede ejecutar en Windows, Linux y MAC OS-X

5.3.3 Comparación entre Kinect for Windows SDK / OpenNI

Kinect for Windows SDK

Ventajas:

- Soporta audio
- Soporta manejo de motor/tilt.
- Puede realizar el Rastreo de cuerpo completo:
- No se necesita calibrar los sensores cada vez que se quiere realizar un rastreo.
- Detecta cabeza, manos, pies y clavícula.
- Soporta múltiples sensores.
- Documentación detallada para programadores de distinto nivel.

Desventajas:

- Solo rastrea el cuerpo completo y el cuerpo sentado, no posee modo para rastrear solo las manos.
- Rastreo de cuerpo completo:
- Solo calcula las posiciones de las articulaciones, no las rotaciones.
- Consume más CPU que OpenNI/NITE
- No posee sistema de reconocimiento de gestos.
- Solo soporta Windows 7 (x86 - x64).

Sensor Kinect OpenNI/NITE

Ventajas:

- Incluye soporte para rastreo de manos y gesticulaciones.
- Rastreo de cuerpo completo:
- Calcula la rotación de las articulaciones.
- Consume menos CPU que el rastreador de Microsoft Kinect SDK.
- Soporta los sensores de PrimeSense y ASUS WAPI Xtion.
- Provee soporte para la conexión de múltiples sensores.
- Es Multi-Plataforma Windows (7, Vista y XP), Linux and Mac OSX.
- Soporte para reproducir/grabar desde/hacia el Disco Duro.
- SDK provee eventos para detectar cuando un Nuevo usuario entra al rango de visión del sensor.

Desventajas:

- no soporta audio.
- no soporta manejo del motor/tilt (a menos que se utilice un driver adicional).
- Rastreo de cuerpo completo:
 - Carece de rotaciones para la cabeza, mano, pies y clavículas.
 - Necesita pose de calibración para comenzar el rastreo.

Dentro de SDK no posee eventos cuando un nuevo frame de video o profundidad está disponible.

6. Diseño de la Solución

6.1 Modelo de Proceso

El modelo de proceso escogido para el desarrollo del proyecto es el modelo evolutivo basado en construcción de prototipos. Este modelo permite la iteración de las distintas fases del desarrollo de software sin la necesidad de repetir algunas fases anteriores. El hecho de que el modelo de proceso se base en prototipos, permite la construcción de programas desechables de acuerdo a mini funcionalidades.

6.2 Elección de las Herramientas

Durante el desarrollo de este estudio se dejó en claro cada una de las características de las tecnologías involucradas tanto en sus capacidades, como sus ventajas y desventajas. Para el diseño de la solución es necesario la elección de las herramientas a utilizar ya sea el robot como los lenguajes de programación asociados. Para la creación del robot se decidió elegir Lego Mindstroms NXT para desarrollar los futuros prototipos. Uno de los motivos para la elección de esta tecnología fue por la capacidad de poder diseñar un prototipo único que se adaptara a las necesidades del proyecto. Entre ellas, la capacidad de imitar algún movimiento del cuerpo y la adaptabilidad que presenta.

Con respecto a la Kinect, en base a las pruebas realizadas a los SDK disponibles, se tomó la decisión de crear un mini-prototipo para cada SDK y así probar sus capacidades, ya que tanto los driver de OpenNi como los de Microsoft ofrecen las mismas características con respecto al Skeleton Tracking. Por lo tanto, las dos herramientas serán sometidas a un estudio más profundo con respecto al desarrollo del prototipo y así poder elegir el SDK definitivo.

También se definieron los lenguajes de programación utilizar; los cuales son Java o C# para el Robot NXT, y en el caso de la Kinect es C#. Otro aspecto importante con respecto a la elección de los instrumentos; es que la tecnología es altamente sensible a los cambios; es decir, que durante un año a otro se han mejorado considerablemente los SDK, sacando nuevas versiones “Beta” tanto para el SDK de Windows como para SDK de OPEN NI . Dando más claridad a lo que se escogió por los proyectistas. Para la conexión del robot con el software, se utilizará el dispositivo Bluetooth y además de la biblioteca de C# Mindsquall, que soporta la conexión.

Mindquall es una librería.Net que permite controlar al robot Lego Mindstorm NXT o NXT 2.0 vía USB o Bluetooth [13].

Características.

- Soporta todas las características del NXT 2.0, incluyendo el sensor de color.
- Soporta conexión USB en caso de que no exista Bluetooth
- Funciona con el procesador más antiguo que es el Lego RCX
- Puede ser usado con .Net 2.0, .Net 4.0 y Visual Studio 2010 en adelante.

Finalmente, en la etapa de implementación del software, se probaron estas herramientas y se decidió por utilizar SDK de Windows 1.7 e implementado en lenguaje C#, debido a la existencia de gran documentación en la web, esto permite que los desarrolladores tengan una mayor fiabilidad a la hora de programar.

6.3 Descripción de las Principales Clases y sus Métodos

Tras el estudio se decidió ocupar el SDK oficial de Microsoft el cual actualmente está en la versión 1.7, con el cual se obtuvieron herramientas suficientes para empezar con la creación de prototipos descartables siguiendo el modelo evolutivo, estos prototipos deberán cumplir las mini-funcionalidades que serán verificadas a la hora de realizar las pruebas pertinentes. La plataforma de trabajo será Microsoft Visual C# 2010. La programación se realiza en base a los pilares de Orientación Objeto, por lo que es necesario de la creación del modelo de clases.

6.3.1 Clase Interfaz

La Clase Interfaz es hereda de la “Clase Form”, por lo que contiene los atributos necesarios para la creación de los botones, el temporizador, los cuadros de textos y las etiquetas. Esta clase contiene los métodos que hacen funcionar cada botón del programa como también el temporizador. Los Principales métodos de la Clase Interfaz son:

- **private void Interfaz_Load (object sender, EventArgs e):** Éste método es el principal que carga la interfaz se encarga de obtener los puertos COM y asignarlos a la lista de puertos, e inicializar el temporizador que regula cada cuanto tiempo se toma los datos.
- **private void Temporizador_Tick (object sender, EventArgs e):** Éste método está encargado de trabajar con los datos que calcula el objeto tracker, el objeto tracker guarda la información en variables globales, las cuales son accedidas por el método Temporizador_Tick, realiza un cálculo de diferencia de ángulos y el resultado lo muestra en la interfaz través de los cuadros de texto. Este mismo método llama a otro método llamado escribir_en_archivo.
- **private void BotonComenzar_Click (object sender, EventArgs e):** El método encargado de inicializar el sensor al presionar el botón mediante el método Start (); para que el objeto Tracker empiece a trabajar en el reconocimiento esquelético, además muestra en un cuadro de texto la palabra “Rastreando”.
- **private void BotonTerminar_Click (object sender, EventArgs e):** método encargado de terminar la ejecución del sensor mediante su método Stop (); con el cual se detiene el trabajo del objeto Tracker, almacena los datos para posteriormente ser guardados y además muestra en un cuadro de texto la frase “Rastreo Terminado”.
- **private void BotonGuardar_Click (object sender, EventArgs e):** este método se encarga de mostrar un dialogo de guardado de archivos en el cual se solicita el nombre del archivo a guardar y la ubicación en la que se quiere guardar, en este archivo se almacena la información recopilada por la Kinect.
- **private void button4_Click (object sender, EventArgs e):** este método se encarga de mostrar un cuadro de dialogo de apertura de archivos, en el cual se deberá seleccionar el archivo con las ordenes a cargar. El contenido de los archivos se mostrará en un cuadro de texto.

- **private void button5_Click (object sender, EventArgs e):** este método es un botón que se encarga de crear el objeto motor para la ejecución de órdenes. En este método se lee el arreglo que contiene las órdenes previamente cargadas y dependiendo de su contenido se envían las distintas órdenes a los motores correspondientes antes mencionados. También en este y otros métodos se manejan las excepciones correspondientes para prevenir errores en la ejecución del programa.

6.3.2 Clase Tracker

La clase Tracker, hace referencia a clase que maneja el “Skeleton Tracking”, de la cual se instancia un objeto el que tiene como función rastrear el esqueleto y obtener su información. Esta clase posee sus propios métodos, y tiene la información necesaria para la conectar la Kinect y detectar el sensor para comenzar el rastreo. Los métodos principales de la Clase Tracker son:

- **Public void SensorSkeletonFrameReady (object sender, SkeletonFrameReadyEventArgs e):** Contiene las variables necesarias para guardar la información de las coordenadas de cada articulación. Primero se encarga de detectar si hay un esqueleto en pantalla, si es así, realiza el seguimiento. Se crean objetos de tipo “Joint” los cuales almacenaran la información de una articulación. La funcionalidad del método Joint, es calcular las coordenadas de las extremidades del cuerpo. Por ejemplo: `Joint mano = skeleton.Joints [JointType.HandRight]`, se crea un objeto de tipo Joint el cual tendrá la información de la mano derecha. Para el caso de realizar el cálculo de los ángulos es necesario verificar en base a condiciones, que parte central del cuerpo esta rastreada. La condición `if (centroHombro.TrackingState == JointTrackingState.Tracked)`, sí se cumple la condición se guarda la información de las articulaciones o Joints en variables, ejemplo: `Mano_X = (double) mano.Position.X * 100`. Al hacer cada uno de estos cálculos con las coordenadas de todas las articulaciones necesarias (mano, codo y hombro), se establecen los puntos de referencia y se realiza el cálculo del ángulo actual de la articulación con el método `Atan2` de la librería `System.Math`, la cual recibe dos puntos del plano y devuelve el resultado expresado en radianes los cuales se deben traspasar a grados. Una vez realizado el cálculo con cada articulación la información obtenida se guarda en variables globales.

6.3.3 Clase Global

Esta clase fue creada para poder almacenar las variables globales las cuales son las que almacenan los datos de los ángulos y los movimientos para poder ser mostrados en la interfaz y además tiene un método el cual será descrito a continuación.

`public static void` Almacenado Movimientos (`double` movimiento_hombro, `double` movimiento_codo, `double` giro_hombro): Método usado para escribir la información de los ángulos y movimientos obtenida del objeto tracker en un archivo de texto plano llamado "lego3.txt". El método ocupa la librería (System.IO) para poder trabajar en archivos de texto. El formato de texto del archivo corresponde al prototipo de software del robot que lee sólo una rutina, la escritura se realiza sólo si se cumplen ciertas condiciones de las cuales son verificadas, para evitar que se escriban movimientos muy pequeños, y errores de lectura del sensor que provocan que se detecte un movimiento muy grande.

6.4 Diseño del Prototipo del Software Kinect

6.4.1 Prototipo Kinect Versión 1.

Como se mencionó en las secciones anteriores, el objetivo del proyecto es utilizar la Kinect como dispositivo de enseñanza, para esto es necesario utilizar la capacidad de rastrear el esqueleto de la Kinect para detectar alguna parte del cuerpo y obtener todos los datos necesarios. Algunos datos obtenidos con el dispositivo son: Las coordenadas del codo, hombros y las manos, el movimiento desde un punto hacia a otro, y el tiempo que se demoró. Las coordenadas que maneja el software son X, Y, Z con respecto a las extremidades del cuerpo (codo, hombros y manos).

Otra funcionalidad que presenta éste prototipo, es la generación del archivo de texto de formato TXT. El cual lee el motor que puede ser (A, B, C), seguidamente de un salto de línea, además de los grados en radianes que van entre (-60° a 180°), con estos datos se pretende generar el archivo de texto, la información obtenida serán las órdenes que leerá el Robot Lego. En otras palabras, el usuario deberá situarse frente a la Kinect y realizar movimientos, la Kinect se encargará de registrarlos y el prototipo que la controla, se encargará de procesar los datos y generar el archivo de texto correspondiente. Una vez generado el archivo, en un primer caso el computador deberá enviar las órdenes una por una al robot para que este las ejecute, si se logra esto, el segundo caso será que el computador envíe todas las órdenes al robot y este se encargue de ejecutarlas una por una. Este envío de datos será a través de USB en una primera instancia y después será implementado con Bluetooth. En el primer caso el robot dependerá más del computador que en el segundo.

6.5 Diseño de Prototipos Robot Lego Mindstorm

6.5.1 Modelo Prototipo Versión 1

Una vez obtenido el acceso al laboratorio, se procedió a diseñar mentalmente, y armar el robot a modo de prueba y error, la estimación de tiempo fue de una semana para realizar el primer modelo de robot. Este modelo posee característica, pruebas realizadas y criterios de descartes. Las características del prototipo del robot. Éste cuenta con una base en la cual se encuentra instalado un servomotor en forma horizontal, el cual va conectado hacia el brazo a través de un mecanismo de engranajes, el NXT (procesador) va montado en la parte trasera, y el brazo robótico va en la parte superior. El problema que presenta este modelo es la inestabilidad del brazo, debido a la poca firmeza.

Las pruebas realizadas al prototipo versión 1, fueron pruebas del software de Lego Mindstorms. En base a una pequeña prueba de programación con el Software de Lego, se le indicó ciertas rutinas de movimientos, lo cual terminó en un fracaso absoluto al no poder realizarlas. Se probó la rotación del hombro y el movimiento del brazo los cuales los realizaban con dificultad, y a veces no era capaz de realizarlos. Este modelo se descartó debido a fallas de diseño, ya que no se consideró el peso del brazo, al ser muy pesado los engranajes que se dedicaban a hacerlo girar, se trancaban y no cumplían su función, este modelo fue desarmado y se pensó en un nuevo diseño.

6.5.2 Modelo Prototipo Versión 2

Una vez descartado el primer modelo, se prosiguió a diseñar el segundo prototipo de Robot, después de un pequeño análisis de las falencias del primer modelo, se llegó a la conclusión de que la estructura que debía soportar el brazo debería ser más resistente y que a su vez le proporcionaría un eje de rotación de aproximadamente trescientos grados, después de este análisis de la anatomía del brazo humano, se diseñó y armó el nuevo modelo el cual será descrito a continuación.

El modelo de prototipo versión 2, posee las siguientes características: La ubicación del motor que hace girar al hombro, éste paso de estar de forma horizontal a vertical. La conexión hacia el brazo con respecto al motor era directa, sin engranajes, por lo cual cumple la función de hacerlo girar. Uno de los principales problemas que se tuvo con este modelo era el peso del brazo, al igual que en el modelo versión 1, el brazo robótico tendía a inclinarse. Además se hicieron pruebas con prototipos programados en java que realizaban pequeños movimientos para probar la resistencia del brazo, se constató de este modelo cumple su objetivo, pero el problema del peso del brazo persistía.

Las pruebas realizadas a este modelo fueron también realizadas con el Software de Lego, en la cual se probaron cada uno de los servos con los movimientos indicados a través del software. Estas fueron exitosas por lo cual se precedió a testarlo con el prototipo 1 de software programado en Java, al igual que en la prueba anterior se verificó el correcto funcionamiento de los motores. Al comprobar el cumplimiento de los movimientos, se estableció que este modelo sería la base para futuras modificaciones.

6.5.3 Modelo Prototipo Versión 3

Como se mencionó anteriormente en la búsqueda de seguir perfeccionando el robot, se modificó añadiéndose dos botones (sensores de tacto) para que pudiera funcionar con el nuevo prototipo de software, con el cual podrá hacer tres secuencias de movimientos distintas. Además se reforzó la base que sostiene el motor que hace girar el hombro y se agregó un contrapeso al brazo. Con estas modificaciones se añadió más estabilidad frente a movimientos bruscos y una mayor funcionalidad en el caso de la lectura y procesamiento de movimientos.

La principal característica de este modelo es que con cada uno de los botones, se realiza una secuencia de movimientos diferente, estas secuencias se encuentran almacenadas en un archivo de texto de formato TXT, el cual es leído por el computador y los datos son enviados al robot. Con el modelo versión 3 se realizaron pruebas con el prototipo 1 de software programado en Java, al verificar su correcto funcionamiento se procedió a ocupar por primera vez el prototipo versión 2. Después de una serie de ajustes al código fuente del prototipo 2, se verificó el correcto funcionamiento del modelo versión 3. Este modelo de robot es el que actualmente se está ocupando en las tareas de programación y las pruebas de prototipos, cumple la función para la cual fue diseñado, pero no se descartan futuras modificaciones.

6.5.4 Modelo Prototipo Versión 4

El último modelo implementado es la versión 4, al robot se le adquiere una nueva pieza de engranaje, esta pieza fue fundamental en aspecto físico del robot, porque gracias a ella se logró más estabilidad para el brazo robótico. Además se aumentaron los grados de movimientos, llegando a tener hasta 360° de libertad al girar el hombro. La principal característica de este modelo es que cumple con todas las funcionalidades ya implementadas en los modelos anteriores, pero ahora la ejecución de las rutinas es mucho mayor debido a la nueva pieza añadida. Se crearon tres rutinas distintas, con este modelo la primera rutina fue sacarle provecho al máximo el motor del hombro, realizando tres rutinas de movimientos distintas y de forma repetitiva.

6.5.5 Modelo Prototipo Versión Final

El robot ha pasado por una serie de modificaciones con respecto a versiones anteriores. El brazo fue puesto de forma horizontal sobre una base con ruedas, esta base le otorga mayor estabilidad y permanece en su posición firmemente. El motor que simulaba ser el hombro fue cambiado de ubicación desde la punta del brazo hacia el centro de este. El cambio tuvo como principal motivo dar al robot la capacidad de girar sobre sí mismo sin la limitación de los cables. Este motor al estar en el centro del brazo, es conectado con la base con ruedas sobre la cual girará.

El motor que le daba al brazo la capacidad de giro simulando la rotación del hombro fue cambiado de su posición inicial que era en forma perpendicular al brazo (forma de L), a una posición simulando a un brazo extendido. Por lo que el robot paso de ser un tronco humano y un brazo sobresaliendo de un lado a ser solamente un brazo. Estos cambios fueron realizados en base a adaptarlo a una funcionalidad en específica.

Estas modificaciones permiten al robot mover dos motores sin límite de grados ni limitación por el largo de los cables, pero el motor que corresponde al giro del hombro solo se puede mover con una libertad de 360°, si gira más de esto se verá limitado por el largo del cable. El nuevo robot podrá sostener en la punta del brazo un lápiz con el cual tendrá la capacidad de dibujar a partir de órdenes captadas anteriormente por la Kinect. Esto corresponde a una adaptación para que cumpla con una nueva funcionalidad la cual es dibujar.

6.6 Prototipo de Software Diseñado para NXT

Con el fin de poder verificar el correcto funcionamiento de los modelos de robot creados, se comenzó a desarrollar el primer prototipo de software el cual debía poder leer un archivo de texto, ver su contenido y enviar las órdenes adecuadamente al Robot.

6.6.1 Prototipo de Software en “JAVA”

Para desarrollar el primer prototipo se empezó a utilizar Eclipse índigo con las librerías lejos, esta es una herramienta que ocupa en java para programar el NXT. Luego de unas pequeñas líneas de código y basándose en un programa que hacia rotar un motor a partir de un sensor de sonido, se pudo crear el primer prototipo en Java.

El principal objetivo de este prototipo es leer un archivo de texto con todas las instrucciones necesarias para cada uno de los tres motores que componen el brazo robótico NXT, este documento de texto se compone del nombre del motor (A, B o C) y la cantidad en grados que se requiere que se muevan, ya sean positivos o negativos, el prototipo se encarga de leer los datos línea por línea y guardarlos en un Array que guarda datos de tipo String, una vez leídos los datos se recorre el Array leyendo todos los movimientos y ejecutándolos a medida que se va avanzando. A continuación se mostrará el formato de la información que contiene el texto.

Para asegurar el correcto funcionamiento del prototipo, se realizaron pruebas con el modelo de robot versión 1, en las cuales se tuvieron buenos resultados al funcionar correctamente los motores, solo hubo que arreglar unos detalles sobre la fuerza de los motores, al estar por defecto en 100, los motores se movían con violencia y provocaban que el Robot se desarmara. La solución a este problema fue ajustar la fuerza de los motores.

6.6.2 Prototipo de Software en “C#”

Con el fin de crear un prototipo de software desarrollado en C#, el cual pudiera controlar tanto el Robot como la Kinect, se decidió descartar el prototipo en Java y comenzar a trabajar con el robot en C#. Utilizando las bibliotecas Mindsquall, las cuales permiten controlar el robot mediante USB o Bluetooth. Se empezó a desarrollar un prototipo de consola que enviaba una orden al robot, la cual era mover un motor. Una vez logrado esto se comenzó a modificar el prototipo llegando a implementar la misma funcionalidad de la versión en Java que es leer un archivo y ejecutar las órdenes.

Para esto se diseñó que el prototipo leyera la información y la guardara en un Arreglo (ArrayList) y después este arreglo fuera recorrido y se ejecutaran las órdenes en su interior. Una vez terminado este prototipo, este se integrara al software que controla la Kinect que dará como resultado un prototipo final que controlará ambas tecnologías.

6.7 Prototipo Final Integración de Robot Lego y Kinect

La última versión del software consta con una variedad de mejoras en la interfaz de usuario. Los cambios efectuados en ésta, son la ubicación de los botones principales, que ofrecen más facilidad en la conexión con los dos dispositivos que controla el programa que son Kinect y el Robot Lego. El uso del programa se ha simplificado con respecto a las versiones anteriores. Se han agregado nuevos ítems, y se ha ordenado la información de manera tal que el usuario pueda visualizar mejor las funcionalidades del programa. En la Interfaz del programa, se presenta tres secciones principales: El panel de conexión, el panel de captura de movimientos y el panel de ejecución de instrucciones. El panel de conexión establece la conexión con la Kinect mediante botones, establece la conexión con el robot mediante Bluetooth seleccionando previamente el puerto COM correspondiente otorgado en una lista despegable con los puertos disponibles, todas estas conexiones están validadas en caso de que se produzca algún error.

La captura de movimientos es una sección que requiere que la conexión con la Kinect esté activa, permite capturar los datos obtenidos del dispositivo y ofrece la opción de guardar estos datos en un archivo de texto si son satisfactorios mediante un cuadro de diálogo. Esta sección permite guardar tantas secuencias como sean necesarias. Para mejorar la experiencia del usuario con esta funcionalidad, se agregó al software la capacidad de mover al robot de forma online para poder visualizar con mayor facilidad cuales son los movimientos que se van a guardar. En el proceso de captura de movimientos, se eliminó el control del ángulo Z, este ángulo determinaba cuando el brazo se iba a levantar o bajar, este ángulo tenía una serie de complicaciones al ser captado por la Kinect, ya que bajo ciertas circunstancias y parámetros del entorno visual del dispositivo, se producían errores de lectura. Para substituir el uso del ángulo Z, se decidió ocupar la mano izquierda como referencia para levantar el lápiz o bajarlo. Para que el robot levantara el lápiz se requiere que la mano izquierda esté levantada y para bajarlo se requiere bajar la mano. Esta funcionalidad trabaja con variables de estado, estas variables cambian dependiendo de la posición de la mano izquierda.

Por último está la sección de ejecución de instrucciones. Esta sección se encarga de cargar un archivo de texto seleccionado mediante un cuadro de dialogo, obtener la información contenida en este y mediante un botón enviar las órdenes al robot previamente conectado. Al momento de realizar la conexión con la Kinect mediante el botón correspondiente se puede dar el caso que la Kinect no esté conectada, si esto sucede se desplegara un cuadro de mensaje advirtiendo de ésta situación. Por el contrario, si la conexión es exitosa, se mostrará un mensaje notificando que se realizó la conexión con éxito.

7. Pruebas de Prototipos

7.1 Prueba de Prototipo Consola

Las pruebas de verificación realizadas a este prototipo correspondieron a corroborar que la información entregada por el sensor sea correcta, es decir, se indicó que el sensor obtuviera los datos de la mano derecha y estos fueran mostrados por consola indicando a que coordenada correspondía, para esto uno de los integrantes se tuvo que parar en frente de la Kinect y empezó a mover la mano, se constató que los valores entregados por el objeto tracker eran correctos.

Las otras pruebas realizadas a este prototipo correspondieron a verificar el correcto funcionamiento del cálculo de los ángulos, el movimiento y la escritura en archivo. Ocupando el mismo procedimiento de la prueba anterior. Se verificó el correcto cálculo de los ángulos y el movimiento en un intervalo de tiempo, con respecto a la escritura de archivos, surgieron problemas en la manera de abrir el archivo, problema solucionado en versiones posteriores.

7.2 Pruebas de Prototipos en Windows Form

Una vez verificado el funcionamiento de los prototipos de consola, se prosiguió a desarrollar algo más elaborado, esto consistió en traspasar el código del programa de consola a una aplicación con interfaz en Windows Form también programado en C# aplicando Orientación a Objetos, es decir, separando el programa en clases e instanciando objetos.

En el primer prototipo desarrollado con Windows Form, se verifico que los botones cumplieran la función para la cual estaban creados, es decir, que el botón comenzar encendiera la Kinect y el terminar la apagara, también se verifico de igual forma que los prototipos anteriores, que el programa pudiera rastrear el brazo y calcular los ángulos correspondientes. Debido a que en este prototipo todavía no se implementaba en temporizador, los cálculos de los ángulos y movimientos se realizaban a 30 fps. Se probó la escritura de archivos con las condiciones y apertura de archivos correspondientes. Para verificar su funcionalidad, un integrante tuvo que ser rastreado, el prototipo respondió correctamente al calcular los ángulos y los movimientos en el intervalo de tiempo definido en el temporizador y la correcta escritura de los movimientos en el archivo.se agregaron condiciones para que no se escribieran datos muy grandes o datos erróneos.

7.3 Prueba de Prototipo Final

Una vez implementado el prototipo final, se realizaron pruebas las cuales consistieron en realizar una captura de movimientos para poder ejecutados por el robot desde el mismo software. El resultado de las primeras pruebas mostraron que el software envía todas las órdenes juntas y es probable que las órdenes lleguen desfasadas, la solución a esto fue implementar un temporizador que enviara las órdenes cada 1 segundo. En las pruebas se generaron una serie de errores con respecto al no manejo de excepciones. Estos errores fueron solucionados al manejar las excepciones que se pudieran generar. Después de una serie de mejoras, se logró realizar pruebas exitosas, es decir, se logró captar una secuencia de movimientos, guardarlos en un archivo de texto para después dar la función de poder cargar estos datos y poder enviárselos al robot.

7.4 Prueba de Prototipo Final: Funcionalidad específica

Al realizar con éxito la última prueba en la cual se corroboró el correcto funcionamiento del software con el Robot, se procedió a modificar al robot para darle una funcionalidad específica la cual es dibujar. Al terminar estas modificaciones al igual que con modelos anteriores se procedió a probar su funcionalidad, el robot fue adaptado para poder ser puesto encima de una hoja de block en la cual pudiera moverse y poder a través de un lápiz colocado en la punta del brazo, dibujar circunferencias y líneas curvas a partir de las instrucciones captadas desde el brazo e interpretadas por el software.

Las pruebas consistieron en captar órdenes tanto online como offline y probar la capacidad del robot de dibujar con la mayor similitud a los movimientos del brazo. El resultado de las pruebas se realizó de forma exitosa, ya que el robot pudo dibujar sobre el papel con los movimientos captados por la Kinect. Estas pruebas se ejecutaron en dos formas usando el modo online es decir, los movimientos captados fueron ejecutados al mismo tiempo que eran almacenados y se comprobó que estos eran los mismos movimientos que se registraron y se ejecutaron posteriormente, y se probó el modo offline captando los movimientos y ejecutándolos posteriormente. Se requirió hacer unos pequeños ajustes en el brazo robótico para que pudiera sostener mejor el lápiz para que realizara los movimientos con mayor facilidad.

8. Conclusión

El estudio se basó en dos áreas importantes que fueron Kinect (Interfaz Natural) y robótica (Lego Mindstrom). Claramente, la comercialización de ambas tecnologías, es, hoy en día conocida por una gran mayoría de personas. En el caso de Microsoft y su competencia, con respecto a los periféricos. Microsoft lidera este mercado desde que empezó a comercializar la Kinect, con la liberación de los drivers y kit de desarrollo, se amplió aún más la gama de productos que se pueden desarrollar tanto para el entretenimiento o estudios científicos.

La importancia de la robótica radica en el diseño y la construcción, ya que obteniendo un buen diseño y un correcto ensamblaje de piezas se puede lograr el objetivo deseado de la mejor manera, aunque no se deben descartar futuras mejoras a la hora de rediseñar. El proceso de las pruebas es fundamental, ya que gracias a este se verifica si el objetivo es cumplido o no, es decir, permite visualizar si los prototipos fueron bien diseñados. Para sí conservar las características de este o descartarlo, y pensar en otro diseño.

Una vez terminado el estudio de ambas tecnologías se logró visualizar de mejor manera como sería la implementación del proyecto, obteniendo resultados favorables en la parte de programación. Algunos de los aspectos positivos del proyecto, son que no se encontró ningún percance a nivel de hardware, y que además hubo una buena compatibilidad del equipo el cual debía ocupar el software asociados a las tecnologías. También se implementó el método online (directa), que fue un plus en el momento de registrar los datos con la Kinect, porque ayudo a visualizar de mejor manera los movimientos que se estaban guardando.

La parte más compleja fue el manejo y procesamiento de los datos entregados por la Kinect, debido a que hubieron muchos datos que se necesitaba ser filtrados, es decir, varias veces se producían errores de lecturas por parte de la Kinect, lo que provocaba que estos datos se transformaran en información basura. Una de las cosas que no se pudo mejorar fue tener un envío más preciso de los datos, ya que la librería no controlaba los tiempos de envío, solamente se enviaba la información, de manera que los datos se podían perder.

Teniendo en cuenta que el objetivo principal del proyecto se logró de forma exitosa, siempre quedan cosas por hacer o mejorar, por esto se planteó algunas futuras mejoras al proyecto, una de ellas es la incorporación del control de voz, esto habría permitido un mejor uso del software en las condiciones en que una persona sola estaría ocupando el programa. La otra modificación sería implementar en la interfaz, una visualización del Skeletal Tracking para el caso de la captura de movimiento. En resumen, todas estas ideas habrían permitido un software más completo.

Finalmente de esta forma se expande la gama de problemas que se pueden resolver con tecnologías computacionales. Por ejemplo, ayuda a personas con discapacidad motriz, ayuda a la industria de automotriz, ayuda para medicina, terapias para enfermedades como el autismo, educación, entretenimiento y muchos otros.

9. Referencias

- [1] Wilschrey, J *Simulador de percusión utilizando interacción natural*, Página 12-15, 2012.
- [2] Pfeiffer, S. *Guiado gestual de un robot mediante un sensor Kinect*. Barcelona, 2011.
- [3] Beland, C. *LEGO Mindstorms . The Structure of an Engineering (R)evolution*, 2000.
- [4] Edubrick. *Guía de Robótica*. 2006.
- [5] Jarret Webb, J. A, *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apress. 2006.
- [6] Microsoft. *Microsoft Corporation*. <http://www.microsoft.com>, Consultado el 13 de Noviembre, 2012.
- [7] J. Ruiz del Solar y R. Salazar (2009) *Introducción a la Robótica*, Universidad de Chile: Departamento de Ingeniería Electrónica.
- [8] Gonzalez.B,Blanco.J,Cambeiro.A,<http://sabia.tic.udc.es/gc/trabajos%20201112/Kinect/comoFunciona.html>, Consultado el 20 de septiembre del 2012
- [9] Lego Mindstorm *Manual de Usuario* <http://mindstorms.lego.com/en-us/Default.aspx>, Consultado el 23 de Junio 2013.
- [10] Alumnos de tecnología industrial, <http://tin1bach.blogspot.com/2011/02/que-es-lego-mindstorms.html>, Consultado el febrero del 2011.
- [11] Microsoft Fully Unveils Kinect for Xbox 360 Controller-Free Game Device. [Ennews.com](http://ennews.com). Consultado el 14 de noviembre de 2010.
- [12] Microsoft Developer Network <http://social.msdn.microsoft.com/Forums/en-US/kinectsdknuiapi/thread/631aff06-cb4c-42ff-84ee-d895c94c5da4/>, consultado 11 de agosto del 2013
- [13] NielsK.Handest, Lego Mindsqualls, Diciembre 2006, <http://www.mindsqualls.net/Introduction.aspx>.

ANEXOS

Lista de Figuras

Diseño de Prototipos Robot

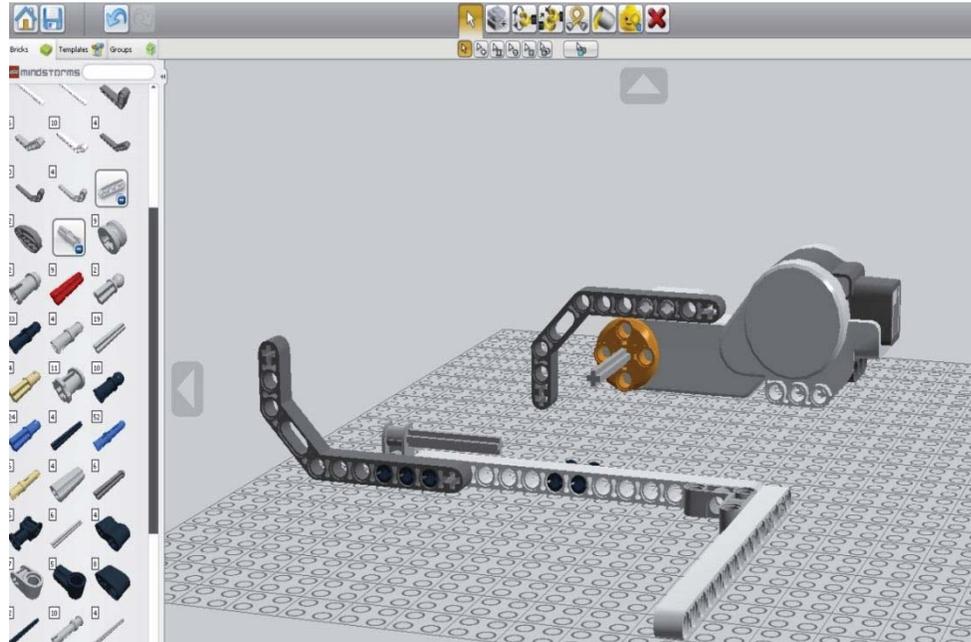


Figura 1 Muestra el Software de Diseño de Lego



Figura 2 Muestra Primer prototipo de Robot

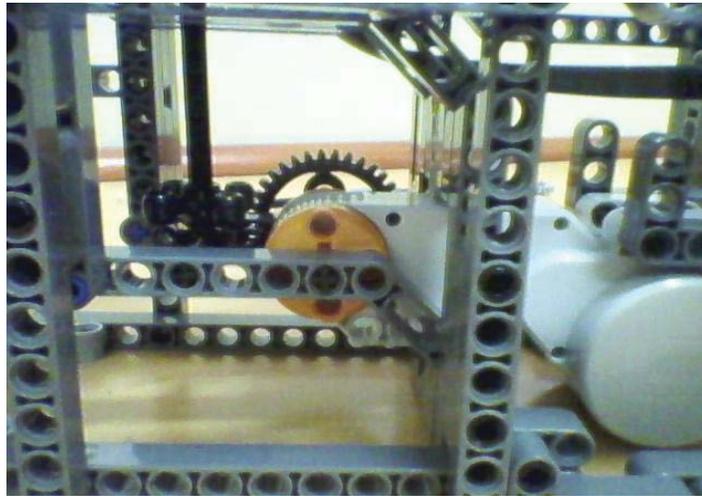


Figura 3 Muestra motor central del Primer prototipo de Robot



Figura 4 Vista posterior del Primer prototipo de Lego

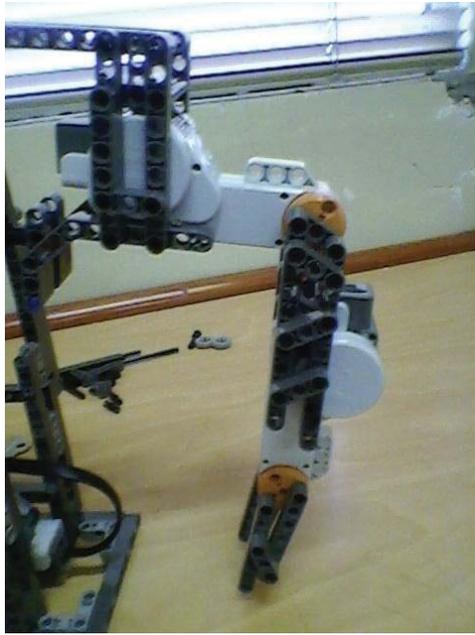


Figura 5 Brazo del Primer prototipo de Lego



Figura 6 Prototipo 2 de Lego



Figura 7 Vista Frontal Segundo Prototipo de Lego

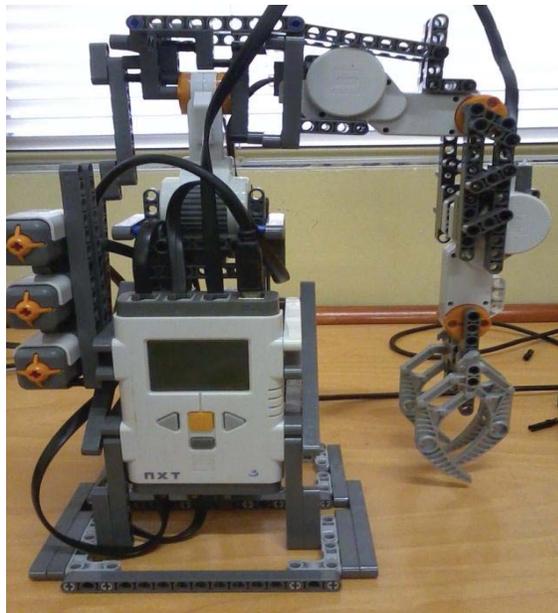


Figura 8 Tercer Prototipo de Lego

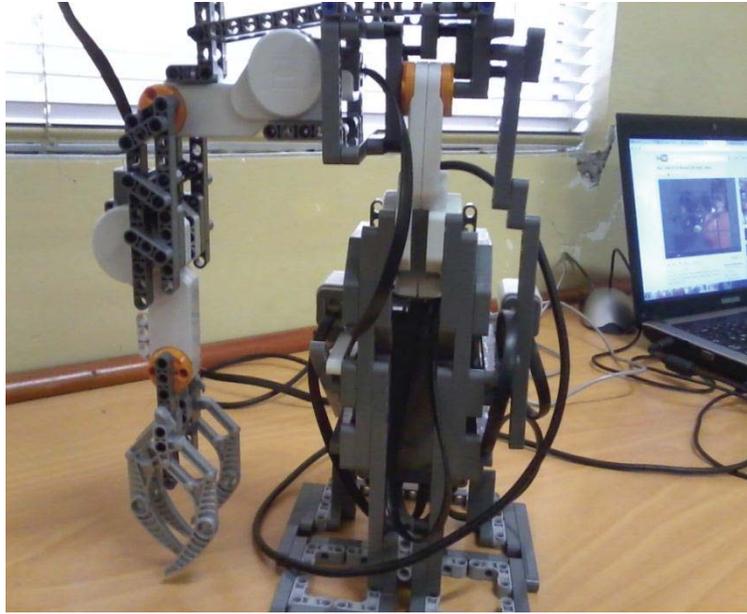


Figura 9 Tercer Prototipo de Lego, vista trasera

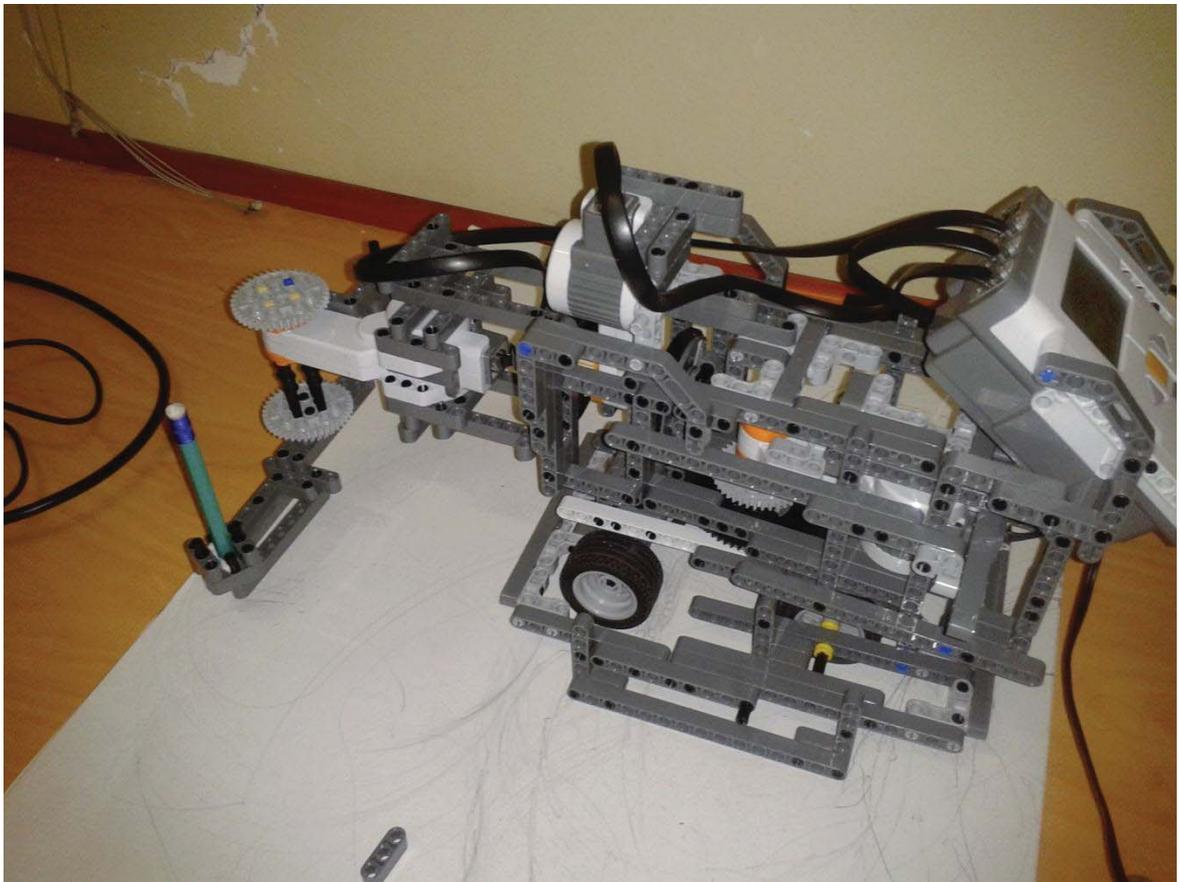
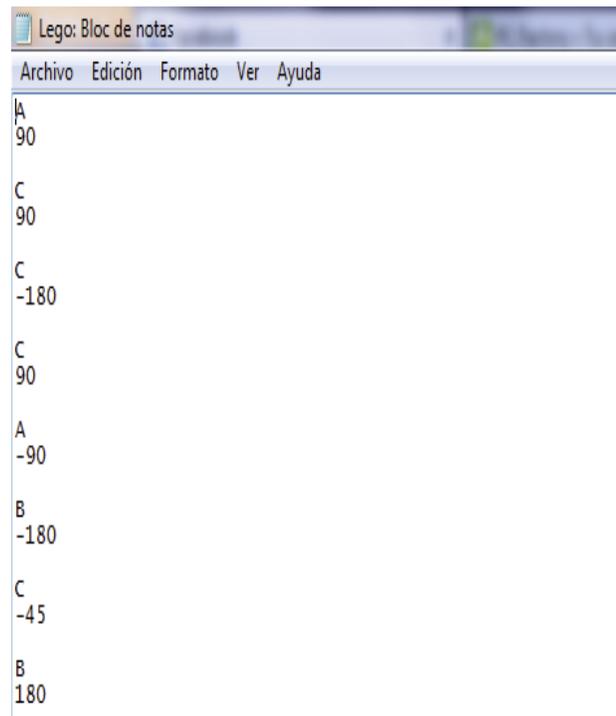


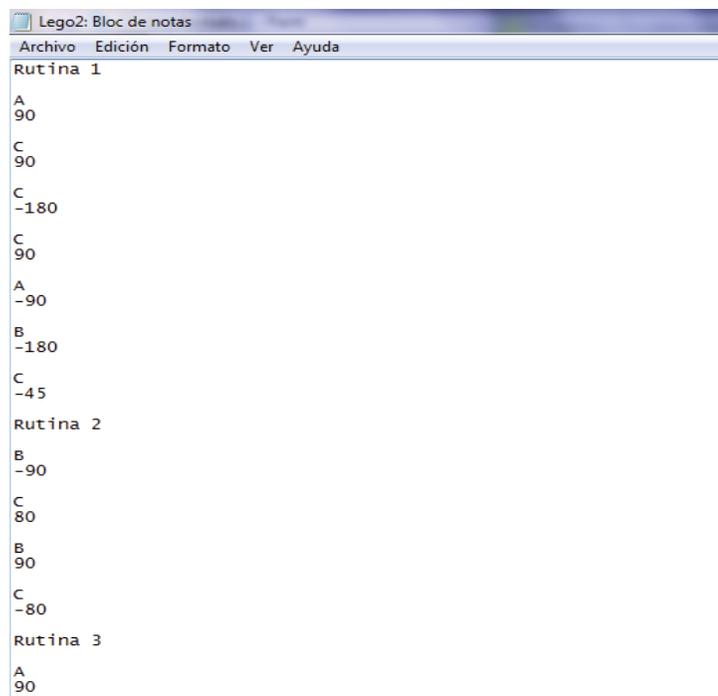
Figura 10 Prototipo Final de Robot Lego

Diseño de la Implementación del software Kinect



```
Lego: Bloc de notas
Archivo Edición Formato Ver Ayuda
A
90
C
90
C
-180
C
90
A
-90
B
-180
C
-45
B
180
```

Figura 11 Formato de texto final



```
Lego2: Bloc de notas
Archivo Edición Formato Ver Ayuda
Rutina 1
A
90
C
90
C
-180
C
90
A
-90
B
-180
C
-45
Rutina 2
B
-90
C
80
B
90
C
-80
Rutina 3
A
90
```

Figura 12 Formato de texto tentativo

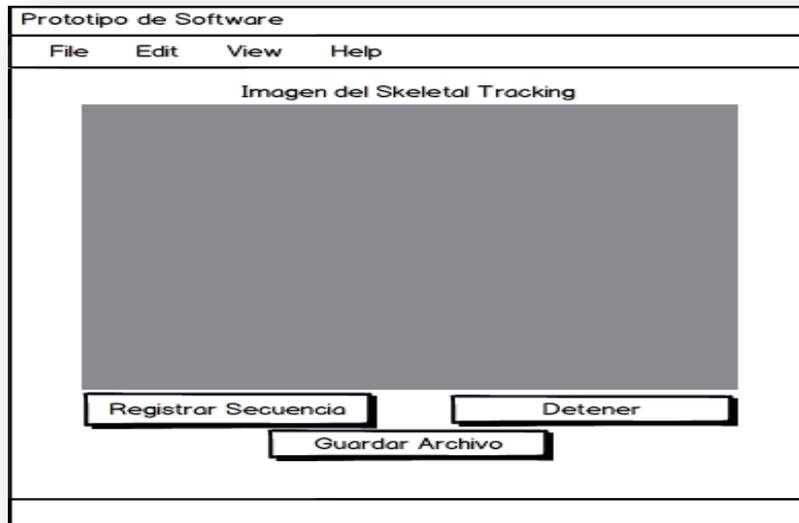


Figura 13 Interfaz prototipo tentativo de Kinect

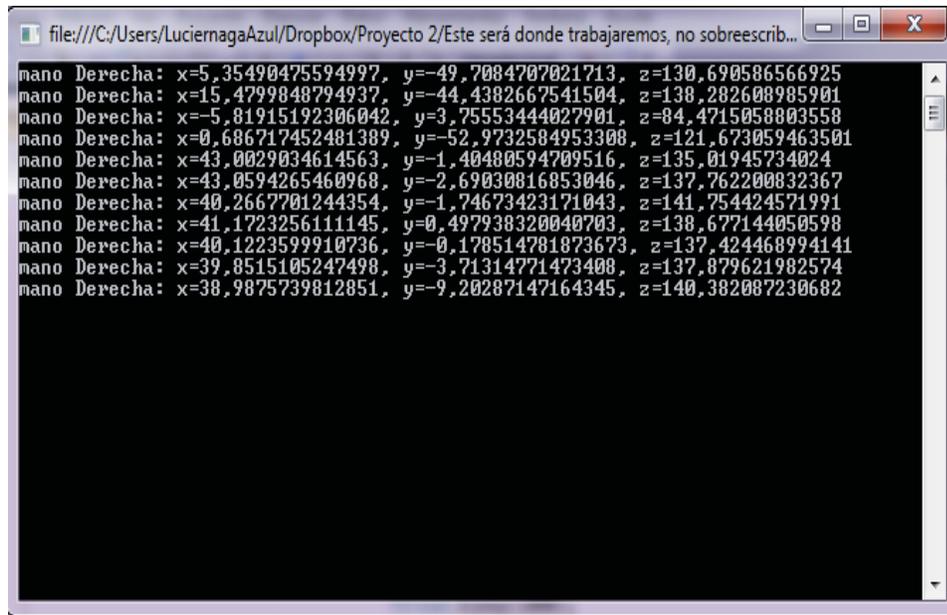


Figura 14 Datos entregados por Primer Prototipo de Consola de Kinect

```
file:///C:/Users/LuciernagaAzul/Dropbox/Proyecto 2/Este será donde trabajaremos, no sobrescrib...
ang actual codo =5,99089507836179
movimiento =5,99089507836179
ang actual codo =72,4381807468153
movimiento =66,4472856684535
ang actual codo =74,713839696464
movimiento =2,27565894964866
ang actual codo =-85,286162218665
movimiento =-160,000001915129
ang actual codo =1,9199595293512
movimiento =87,2061217480162
ang actual codo =72,3510470743112
movimiento =70,43108754496
ang actual codo =-85,0160051595897
movimiento =-157,367052233901
```

Figura 15 Datos entregados por Segundo Prototipo de Consola de Kinect

The screenshot shows the 'Proyecto Kinect' application window. At the top, it says 'KINECT for Windows'. Below that are two buttons: 'Comenzar' and 'Detener'. The 'Estado del Sensor' is 'Sensor Activado'. The interface displays data for four joints: Hombro, Codo, Giro Brazo, and Codo Horizontal. Each joint has two input fields: 'Angulo actual' and 'Movimiento'.

Joint	Angulo actual	Movimiento
Hombro	39	42
Codo	37	28
Giro Brazo	12	26
Codo Horizontal	11	99

Figura 16 Primer Prototipo en Windows Form de Kinect

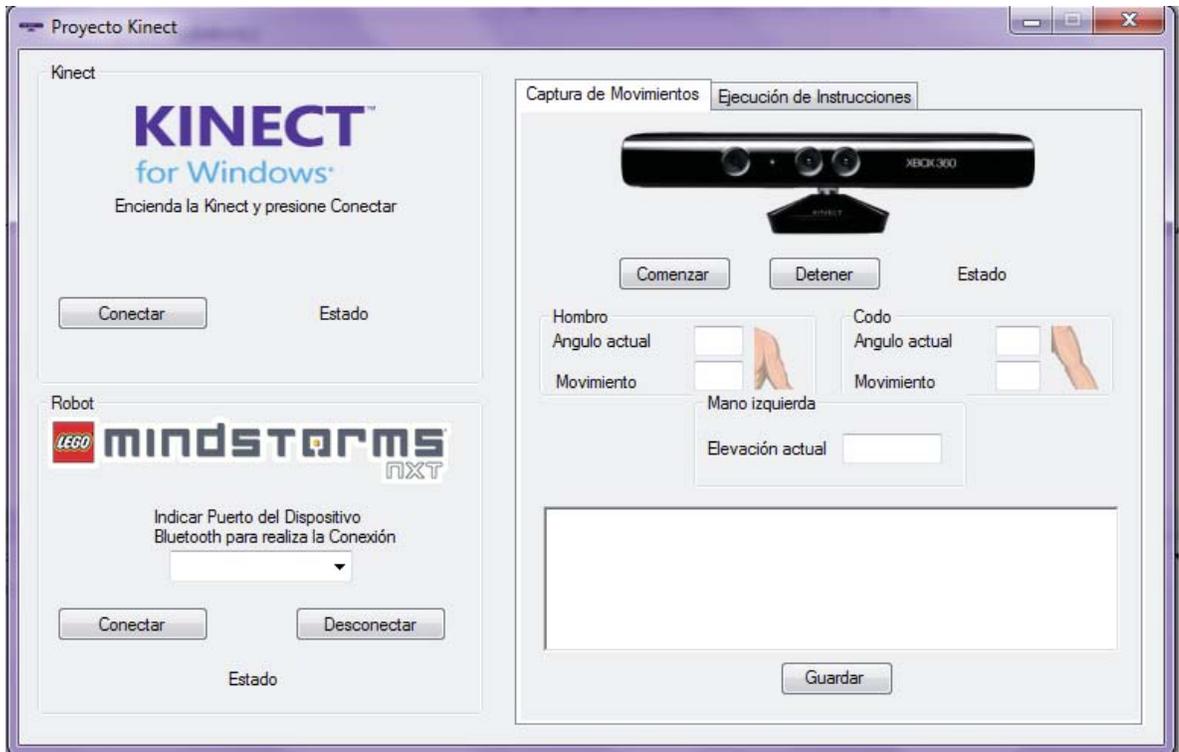


Figura 17 Prototipo Final de Software, pestaña Captura de movimientos

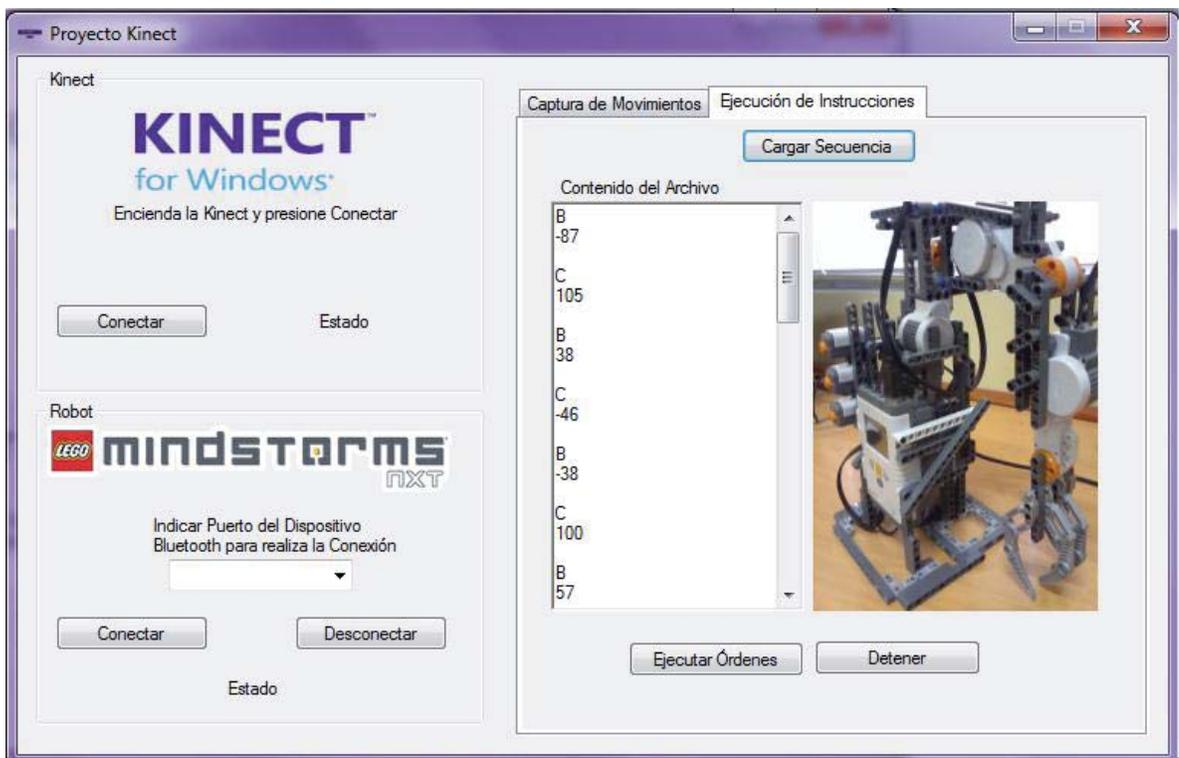


Figura 18 Prototipo Final Software, pestaña Ejecución de órdenes