



DESARROLLO DE UN BRAZO ROBÓTICO VIRTUAL CON CINCO GRADOS DE LIBERTAD MANIPULADO DESDE UN DISPOSITIVO MÓVIL

MANUEL ALEJANDRO BURBANO SOLARTE

UNIVERSIDAD DEL VALLE
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
FACULTAD DE INGENIERÍA
SANTIAGO DE CALI
JUNIO 2019



DESARROLLO DE UN BRAZO ROBÓTICO VIRTUAL CON CINCO GRADOS DE LIBERTAD MANIPULADO DESDE UN DISPOSITIVO MÓVIL

MANUEL ALEJANDRO BURBANO SOLARTE

Trabajo de grado Ingeniería Electrónica
Universidad del Valle

Director:

Director: Ms.C. José Tomas Buitrago Molina

UNIVERSIDAD DEL VALLE
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
FACULTAD DE INGENIERÍA
SANTIAGO DE CALI
JUNIO 2019

Dedicatoria

El presente trabajo lo dedico principalmente a Dios, por ser el inspirador y darnos fuerza para continuar en este proceso de obtener uno de los anhelos más deseados.

A mi padre Miguel Antonio Burbano, por su amor, trabajo, sacrificio y paciencia en todos estos años. Gracias a ello me han permitido llegar a cumplir hoy un sueño más.

A mi hermana Fernanda Burbano por estar siempre presente, acompañándome desde la distancia con su apoyo y oración para poder sortear las dificultades que se presentaron.

A mi esposa Sandra Bolaños Martínez por su amor, cariño, comprensión y quien día a día me apoyo incondicionalmente para poder seguir adelante y no decaer a pesar de todos los inconvenientes que se presentaron en el camino.

A mis familiares quien de una u otra manera estuvieron pendiente de mí, apoyándome de cualquier forma para poder seguir luchando por este sueño que hace tiempo inicio y hoy termina con éxito. En especial a mis tías Martha, Zoila y Rita y mis tíos Jesús y Jaime. A mis primas Diana, Meily, Ximena y Dayana y mi primo Gabriel porque siempre estuvieron presentes con sus buenos deseos y dándome moral para seguir adelante. Así mismo a mi madre, abuelos y abuelas que estuvieron conmigo cuando inicié y hoy (QEPD) no están, pero sé que de cualquier forma todo su amor y cariño me siguió acompañando a cada momento.

A mis amigos Jhon, Stevenson, Danny y todos aquellos compañeros que fueron un gran apoyo en el transcurso de la carrera y apoyo para poder continuar con la realización de este trabajo.

Finalmente, a los maestros de la Escuela de Ingeniería Eléctrica y Electrónica, aquellos que marcaron cada etapa de nuestro camino universitario, y que me ayudaron en asesorías y dudas presentadas en la elaboración de la tesis en especial apoyo al director de tesis José Tomas Buitrago y al director de programa Asfur Barandica ya que sin su constante apoyo no habría podido continuar adelante. Todas las personas que me han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que nos abrieron las puertas y compartieron sus conocimientos. **MANUEL ALEJANDRO BURBANO SOLARTE**

RESUMEN

Este documento presenta un brazo robótico industrial de 5 grados de libertad, implementado en el software de diseño mecánico SOLIDWORKS y virtualizado a través de su exportación a lenguaje VRML. La manipulación de sus articulaciones se realiza a través de comunicación serial con tecnología BLUETOOTH que conecta con una aplicación móvil en una estructura de cliente-servidor.

La aplicación móvil se desarrolla en el IDE Android Studio, la interfaz gráfica construida en XML, compone los controles de las articulaciones en barras de deslizantes. El código principal en lenguaje JAVA hace uso de las funciones de alto nivel de Android para implementar la comunicación BLUETOOTH y despliega el uso de hilos de procesos.

La integración de sistemas se realiza en el entorno de la herramienta matemática SIMULINK de MATLAB, apoyado en el INSTRUMENT CONTROL TOOLBOX y en el SIMULINK 3D ANIMATION, así como en la librería ROBOTICS de Peter Corke que se usa para el cálculo de la cinemática directa del robot.

Palabras clave: *Bluetooth, Realidad virtual, Android, protocolos, Aplicaciones móviles, comunicación serial, grados de libertad, robot virtual.*

Abstract

This document presents an industrial robotic arm of 5 degrees of freedom implemented in the mechanical design software SOLIDWORKS, and exported to VRML language and virtualized. The manipulation of its joints is through serial communication with BLUETOOTH technology that is connected to a mobile application in a client-server structure.

The mobile application is developed in the IDE Android Studio, the graphical interface constructed in XML, including the controls of the joints in progress bars. The main code was programmed in JAVA language uses of the high-level functions of Android to enable BLUETOOTH communication and deploys the use of process threads.

The systems integration is performed in the MATLAB SIMULINK mathematical tool environment, supported by the INSTRUMENT CONTROL TOOLBOX and in the SIMULINK 3D ANIMATION, as well as in the ROBOTICS toolbox of Peter Corke that calculates the direct kinematics of the robot.

Key words: *Bluetooth, Virtual reality, Android, protocols, Mobile applications, serial communication, degrees of freedom, virtual robot.*

CONTENIDO

1. INTRODUCCIÓN.....	10
2. ANTECEDENTES	11
3. PLANTEAMIENTO DEL PROBLEMA	15
4. JUSTIFICACIÓN	16
5. OBJETIVOS	17
5.1. Objetivo general	17
5.2. Objetivos específicos	17
6. MARCO REFERENCIAL	18
6.1 Definición de un robot.....	18
6.2 El brazo manipulador.....	18
6.3 Manipulador o brazo mecánico.....	19
6.4 Características más destacables de un robot industrial.....	19
6.5 Plataforma de simulación virtual.....	20
6.6 Cinemática de un robot.....	20
6.6.1 Cinemática directa.....	22
6.6.2 Algoritmo denavit – hartenberg	23
6.6.3 Cinemática inversa	23
6.7 MATLAB	24
6.8 Robotics de peter corke.....	24
6.9 Solidworks	25
6.10 Android studio.....	26
6.11 Bluetooth	27
7. DESARROLLO DEL PROYECTO	28
7.1 Apuntes metodológicos	28
7.2 Esquema general del proyecto	30
7.3 Modelo mecánico.....	31
7.4 Conversión del diseño mecánico a archivo de realidad virtual	32

7.5 Virtualización del brazo robótico	33
7.6 Creación de la aplicación android	34
7.6.1 Bluetooth	35
7.6.2 Hilos en java	35
7.6.3 Interfaz gráfica	36
7.6.4 UUID (Universal unique identifier)	38
7.7 Instrument control toolbox	38
7.8 Estableciendo conexión	39
7.8.1 Emparejamiento	39
7.8.2 Agregar un puerto	39
7.8.3 El toolbox de comunicación serial	39
7.9 Función de procesamiento en simulink	40
7.10 Obtención de parámetros denavit-hartenberg	41
7.11 Cinemática directa en librería robotics	42
8.RESULTADOS OBTENIDOS	45
9. RECOMENDACIONES	46
10. CONCLUSIONES	47
11. BIBLIOGRAFÍA	47

TABLA DE ILUSTRACIONES

Figura 1. Partes de un brazo robótico.....	19
Figura 2. Modelo de un brazo robótico mediante Solidworks, Simscape, MATLAB.....	20
Figura 3. Estructura de una matriz homogénea.	23
Figura 4. Relación entre los parámetros de las cinemáticas directa e inversa	24
Figura 5. Carátula del manual de librería Robotics.	25
Figura 6. Esquema de estructura general del proyecto.....	30
Figura 7. Robot	31
Figura 8. Robot virtual en entorno SolidWorks.....	32
Figura 9. Robot virtual en formato VRML en el entorno 3D World Editor.....	33
Figura 10. Entorno de Android Studio.....	35
Figura 11. Interfaz gráfica de aplicación final	37
Figura 12. Estructura con función de procesamiento en simulink	41
Figura 13. Herramienta medir de solidworks	42
Figura 14. Gráfica del modelo matemático en robotics.....	43
Figura 15. Paleta en Simulink de Robotics.....	44
Figura 16. Arreglo en simulink de bloques de la librería robotics.....	44
Figura 17. Bloque de Cinemática directa.....	45
Figura 18. Simulación del entorno completo: Bloques de simulink, brazo robótico virtual y gráfica de modelado matemático con librería Robotics	45

LISTA DE TABLAS

Tabla 1. Diferentes configuraciones de los brazos robóticos.....	21
Tabla 2. Estadísticas de mercado mundial de smartphones en Agosto de 2018	27
Tabla 3. Parámetros Denavit-Hartenberg del brazo robótico virtual.....	41

1. INTRODUCCIÓN

La robótica se dedica al diseño, construcción, operación, disposición estructural, manufactura y aplicación de los robots. En la historia de la producción humana, las máquinas han dado un enorme aporte en la necesidad de generar movimientos mecánicos que requieren de fuerza por fuera de lo humanamente posible. Sin embargo, existen movimientos rutinarios que requieren, además de fuerza, cierta complejidad, control y sensibilidad. Es, en este sentido, cuando el concepto de robot entra en escena. Su aplicabilidad se abre paso entre todas las actividades del hombre que exigen procesos mecánicos. Primero, como apoyo, en actividades que son riesgosas para la vida humana y posteriormente en reemplazo, cada vez que exista la tecnología para generar acciones de creciente complejidad.

Dado que la variedad de aplicaciones de la robótica aumenta progresivamente y se sale del mismo ámbito mecánico para entrar en tareas de reconocimiento y clasificación, como la inteligencia artificial. El ámbito que se trabaja en este trabajo se centra en robots manipuladores, que se restringen a la solución de ejecución de movimientos mecánicos de distinta complejidad. Dentro de este grupo, es recurrente su utilización industrial, y dado que se intentan reproducir movimientos que reemplazan el trabajo humano, su morfología toma como base la morfología de un humano. De esta manera es común encontrarse con brazos robóticos industriales, que además de ejecutar las mismas acciones, pueden realizarlas en un menor tiempo y con mayor precisión en ciertos casos.

En la industria se presentan frecuentemente movimientos periódicos, en este caso, es fundamental que un mecanismo robótico presente autonomía y evite la asistencia de un operador, así que se recurre a un programa de cómputo. Sin embargo, también se presentan casos en los cuales son necesarios el reconocimiento, el cuidado, y evaluación de situaciones que presentan una complejidad inalcanzable para un robot o muy costosa de alcanzar, y, por el contrario, son fácilmente realizables por un operario. De esta dicotomía surgen las formas de controlar un robot.

Desde el punto de vista social, las aplicaciones móviles han impactado en todo nivel industrial convirtiéndose en un común denominador en casi todos los aspectos del ser humano y su entorno.

La simulación es un procedimiento habitual dentro de la ingeniería y la ciencia. Permite reproducir el esquema científico de ensayo y error un sinnúmero de

veces, cambiando distintas variables o simulando sólo una fracción del esquema general lo que permite gran flexibilidad para crear conocimiento.

Por tal motivo, en este trabajo se propone la implementación de un prototipo virtual de un brazo robótico con cinco grados de libertad cuyos movimientos sean manipulados desde un dispositivo móvil.

2. ANTECEDENTES

En la evolución de los robots industriales y de servicio para realizar una o varias tareas específicas, hay una estrecha relación con los últimos cambios económicos mundiales, con efecto sobre la automatización de los procesos de manufactura y necesidades del ser humano o de la industria [1]. Por tal motivo, la industria en años recientes ha empezado a utilizar robots para la automatización de procesos repetitivos, peligrosos, los cuales requieren algún tipo de fuerza, llamados robots industriales manipuladores o mecánicos. [2].

Hoy en día los robots se están desarrollando para su uso tanto en la educación como para otras tareas que tienen interacción con el ser humano. A estos últimos se les llama robots colaborativos que están siendo insertados en el trabajo y vida cotidiana en general. Los robots colaborativos funcionan de la misma manera que los manipuladores con la diferencia que tienen menor riesgo, un sistema de control y sensores para controlar los movimientos de su entorno para salvaguardar la integridad de las personas y poder trabajar junto a él.

Existen algunas implementaciones basadas en la simulación virtual como lo son “Modelado y simulación de la trayectoria del brazo robótico de 5 ejes”: en este artículo realizado por Simona Sofia Duicu, Luminita Popa, se plantea un modelo de brazo robótico de cinco ejes modelado con el software SolidWorks y su trayectoria simulada con el software Matlab y Simulink. Este tipo de robot es rápido, preciso, confiable y fácil de programar. El brazo robot tiene sensores y está diseñado para manejar objetos sin intervención humana. El robot presentado tiene cinco grados de libertad, todos sus movimientos están girando. [3].

“Simulador de movimiento de cinco grados de libertad basado en entorno virtual”: en este artículo realizado por LingWen, Yu Zhang, LufengLuo, Cong Zhang, se plantea la simulación del movimiento de un manipulador de cinco grados de libertad basada en entornos de realidad virtual. Para el cálculo de la trayectoria de movimiento del manipulador se establecieron los modelos matemáticos y los

modelos tridimensionales y la simulación interactiva en tiempo real estudio basado en la plataforma EON Studio, proporcionando una base teórica para los múltiples grados de libertad, la entrada de múltiples canales y el movimiento mecánico de percepción múltiple de la investigación virtual basada en simulación interactiva en el entorno EON. [4].

“Modelado y simulación de un robot RV 2AJ de 5 ejes utilizando simmechanics”: en este artículo realizado por Mohammad Afif Ayoba, Wan Nurshazwani Wan Zakaria, Jamaludin Jalani, MohdRazali Md Tomari plantea la fiabilidad y precisión de un modelo del brazo robótico Mitsubishi RV-2AJ de 5 ejes. El modelo CAD del robot se desarrolló utilizando SolidWorks, mientras que el entorno de simulación de múltiples cuerpos se demostró mediante el uso de la caja de herramientas de SimMechanics en MATLAB. Los resultados de la simulación cinemática inversa e inversa propusieron que el modelo establecido se asemeja al robot real con una precisión del 98,99%. [5]

Existen proyectos que vinculan las ventajas de la tecnología móvil con los sistemas mecatrónicos en un ambiente educacional o industrial.

En [6] se desarrolla un entorno de comunicación entre un pc y un dispositivo móvil que ejecuta una aplicación para controlar un robot. Se utiliza tecnología Bluetooth para la comunicación. El pc está conectado al robot el cual cuenta con su propio sistema operativo. El desarrollo está totalmente ligado al robot diseñado por la empresa SCARA, de manera que su aporte se vincula a las utilidades de dicha empresa. En ese sentido, hay ciertas limitantes en cuanto a los recursos, el robot, y los sistemas operativos que se manejan, pues se restringen a una marca un diseño y un modo específico de desarrollo.

El proyecto brinda un ejemplo operativo de una aplicación basada en android para manipular un robot, su entorno consta de dos botones para cada articulación a manipular; uno para aumentar y otro para disminuir el desplazamiento con unos marcos de referencia configurados de antemano.

En [7] se desarrollan dos aplicaciones basadas en un entorno Android, una de ellas trata el tema de control de un robot a través de Bluetooth. El sentido del trabajo va dirigido a apoyar las actividades de estudiantes de ingeniería electrónica y mecatrónica, por tanto, tiene un objetivo educacional y didáctico. El robot que se intenta controlar es un carro modelo y ha sido diseñado y fabricado en la facultad de ingeniería de la Universidad Veracruzana de México. El programa de la aplicación toma como referencia un ejemplo de código libre para los

desarrolladores android llamado “Bluetooth Chat” que podría ser tomado como guía para este nuevo trabajo. Su entorno se basa en el acelerómetro que lleva un Smartphone en su interior para detectar la posición del móvil. La desventaja de este método es la restricción de su manipulación a los cuatro grados de libertad en que se puede inclinar el móvil.

En [8], el sistema mecatrónico a controlar es un humanoide fabricado por la empresa francesa AldebaranRobotics, de manera que se reconocen partes y grados de articulaciones más complejos con movimientos de mayor maniobrabilidad. Se proporciona aquí una forma de utilizar el acelerómetro para manipular el humanoide seleccionando el cambio de un grado de libertad a la vez, de esta forma, se puede aumentar el número de grados de libertad controlados mediante la variación de la posición del móvil.

En [9] se enfatiza en la aplicación móvil y la estructura de comunicación, sino también hay un diseño y desarrollo de un brazo robótico de bajo costo y con un objetivo educacional. De esta manera este proyecto por su integralidad y completitud, se presenta como un excelente ejemplo a seguir. Se implementa además dentro de los modos de operación, unos cuantos recorridos automáticos. Su interfaz parecida a algunas mencionadas presenta botones para aumentar o disminuir el movimiento de la articulación en una dirección configurada de antemano.

En [10] se constituye una tesis para obtener el grado de maestría en tecnología de cómputo; se produce en el Instituto Politécnico Nacional en México. Comunica un dispositivo móvil a través de bluetooth con un ensamble robótico en hardware comandado por arduino que constantemente envía información acerca del posicionamiento de los servomotores. Se da énfasis a la electrónica del ensamblaje. Muestra poco sobre la programación y el desarrollo de la aplicación móvil.

En [11] se centra en desarrollar el control mediante medios inalámbricos de un brazo robótico cuyo ensamble y diseño hace parte del proyecto y se basa en el “Robotic Arm Edge”. Este modelo de brazo robótico sirve de prototipo por sus características y dimensiones, pero está por fuera del ámbito industrial; la empresa que lo fabrica se dedica al desarrollo de prototipos educacionales y juguetes. Se elaboran dos interfaces, una para pc y otra para un teléfono móvil. Para la interfaz y la comunicación a través del móvil, se utiliza la aplicación TouchOs disponible en el Playstore para android, esto facilita el diseño de la interfaz, pero restringe la comunicación al protocolo Open Sound Control basado en UDP y a la estructura

de comunicación que configura dicha aplicación. En este sentido se requiere de un router que permita conexión inalámbrica wi-fi entre los tres terminales: el módulo del robot, el pc y el teléfono móvil.

En [12] se da mayor peso a la electrónica del diseño del brazo robot. En cuanto a la aplicación móvil, se reduce mucho trabajo de programación y diseño debido a que se utiliza el entorno de desarrollo APP INVENTOR que posee la característica de ser muy didáctico, pero impide alcanzar un mayor nivel de profundidad en el desarrollo de aplicaciones.

En [13] se describe cómo manipular un carro robótico el cual tiene unos motores de DC, a través una aplicación Android. La comunicación se realiza por medio Bluetooth y un microcontrolador Arduino. Los movimientos del vehículo se da según los comandos recibidos como adelante, atrás, izquierda y derecha y se visualizan en la aplicación móvil.

En [14] se implementa una aplicación móvil educativa para enseñar a manipular un brazo robótico de dos grados de libertad usando realidad aumentada. La comunicación se realiza de manera serial a través del Bluetooth. Además, se utiliza Matlab/Simulink utilizando un control PID. La aplicación recibe cada uno de los grados de la articulación y los muestra de manera virtual y el usuario recibe una retroalimentación logrando hacer una comparación entre la trayectoria deseada y la real.

3. PLANTEAMIENTO DEL PROBLEMA

El uso de herramientas digitales puede vincular una realidad virtual que integra los gráficos por computadora, la tecnología multimedia, la inteligencia artificial, la red informática y la tecnología de sensores en una tecnología interdisciplinaria [4]. La tecnología de simulación virtual es una de las aplicaciones más importantes de la tecnología de realidad virtual en el campo CAD / CAM. Es ampliamente utilizada en el diseño de mecanismos electromecánicos. La combinación de tecnología de realidad virtual y máquina refleja la nueva aplicación de la era digital en los campos industriales modernos. El brazo mecánico es una parte importante de la máquina industrial. Es un mecanismo de movimiento co-dependiente que consiste en el hombro, el brazo, la muñeca, el cuerpo y el mecanismo de caminar. Actualmente, muchos programas de CAD, como Pro-E, UG, Solidworks, etc., pueden completar el modelado tridimensional de una máquina, y realizar la simulación de movimiento mecánico. Pero la simulación de movimiento de este software tiene desventajas en el rendimiento interactivo en tiempo real. Es significativo estudiar la simulación de diferentes prototipos de brazos con el fin de realizar algún movimiento específico para una tarea puntual que requiera hacer una persona y en muchas de ellas es sabido se pone en riesgo la integridad de la persona.

Por tal razón, es indispensable seguir trabajando en pro de prototipos totalmente confiables aptos para ser manipulados por cualquier operario que desee hacer una tarea concreta sin afectarlo físicamente a él. En primer lugar, implementado de forma virtual, para de manera posterior poder ser llevado a su desarrollo y aplicación física. Además, que a su vez dichos movimientos puedan ser manipulados vinculando las nuevas tecnologías como los dispositivos móviles las cuales involucran el diario vivir del ser humano.

Se formula la siguiente pregunta: **¿Cómo y mediante qué herramientas es viable desarrollar un prototipo de brazo robótico virtual que permita simular características reales y ejecute diferentes movimientos de una tarea específica a través de la tele-operación de un dispositivo móvil?**

4. JUSTIFICACIÓN

El presente trabajo se realiza con un propósito didáctico para el aprendizaje y manipulación de un prototipo virtual de un brazo robótico que desee simular una actividad específica a realizar. La elección del tema referente al diseño, construcción y programación del prototipo, tiene bases en los conocimientos y necesidades que se presentan hoy día en la industria y fuera de ella. La problemática que se pretende satisfacer, es la necesidad de generar diseños propios u originales de sistemas de brazo robótico virtual antes de ser llevados a su construcción física, que puedan ser empleados en la enseñanza y en la investigación.

El brazo robótico virtual brindará apoyo a aquellas personas que cuentan con alguna discapacidad motriz, generando un impacto individual aumentando su autoestima y consiguiendo que puedan realizar acciones como tomar un objeto, abrir una puerta o ejecutar simplemente algunas de ellas rutinarias superando esas barreras del entorno y logrando además inclusión en la sociedad la cual en muchas ocasiones las discrimina.

Es importante aprovechar las aplicaciones móviles para fortalecer el vínculo hombre-máquina ya sea por necesidad o comodidad. Además, esta aplicación sirve como una herramienta didáctica educativa permitiendo que los estudiantes se familiaricen poco a poco con este tipo de desarrollos a nivel prototipo y posteriormente puedan ser implementados de forma real en las empresas e industrias.

5. OBJETIVOS

5.1 Objetivo general

Desarrollar un brazo robótico virtual con cinco grados de libertad manipulado desde un dispositivo móvil

5.2 Objetivos específicos

1. Realizar una revisión documental sobre las diferentes aplicaciones móviles existentes orientadas a la manipulación de los movimientos de un brazo robótico.
2. Desarrollar el brazo robótico virtual mediante un software de diseño mecánico.
3. Definir los parámetros de la interfaz de usuario sobre el dispositivo móvil.
4. Implementar una aplicación móvil que interactúe con el brazo robótico.

6. MARCO REFERENCIAL

6.1 Definición de robot

Pensando en la aceptación general del robot, es una máquina programable que imita las acciones y/o apariencia de una criatura inteligente, generalmente un ser humano. Tener habilidades de un robot, una máquina debe ser capaz de:

- 1) Aprender y percibir: obtener información del ambiente circundante.
- 2) Cumplir diferentes tareas: locomotoras o de manejo, tales como mover o manipular objetos.
- 3) Para ser "reprogramable": puede realizar diversas tareas (misiones), usando diferentes programas.
- 4) Para trabajar de forma autónoma o para interactuar con humanos seres.

Las partes esenciales en un sistema robotizado son las siguientes:

- El controlador.
- Los actuadores.
- Los reguladores.
- Elemento terminal.
- Sensores.

6.2 El brazo o manipulador

La estructura mecánica del manipulador puede ser tan variada como los fabricantes que las hacen. Pero generalmente se pueden distinguir cuatro partes principales en el manipulador:

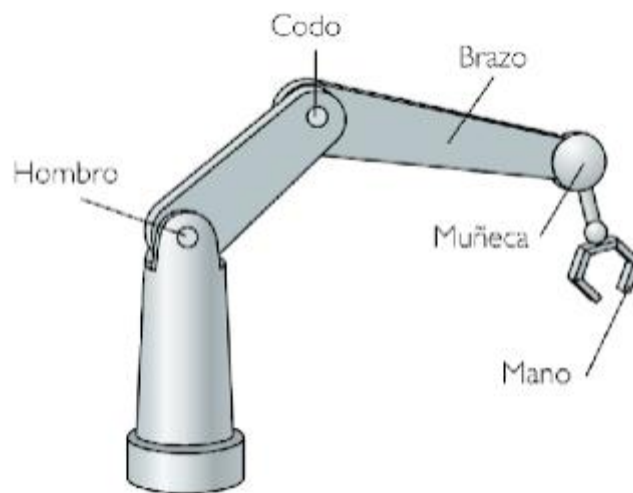
- El pedestal.
- El cuerpo.
- El brazo.

- El antebrazo.

Las articulaciones entre las distintas partes rígidas del brazo pueden ser giratorias (como las del brazo humano) o deslizantes (si hay traslación de las partes). El número de elementos del brazo y sus articulaciones determinan una característica propia de cada robot. Al número de movimientos espaciales independientes entre sí se le denomina grados de libertad.

La figura 1 muestra un ejemplo de las partes que componen un brazo robótico.

Figura 1. Partes de un brazo robótico.



6.3 Manipulador o brazo mecánico.

El manipulador consta de un conjunto de herramientas interrelacionadas que permiten los movimientos del elemento terminal del brazo del robot. Consta de una base para sujetarse a una plataforma rígida (como el suelo), un cuerpo donde se suele integrar la mayor parte del hardware interno que lo hará funcionar (circuitaría, placas impresas, etc.), un brazo para permitir un gran movimiento en 3 dimensiones y un antebrazo para hacer también movimientos en 3 dimensiones, aunque diferenciales (muy pequeños) y de mucha precisión, tal que puede llegar a los nanómetros, es decir, a 1×10^{-6} mm

6.4 Características más destacables de un robot industrial.

- Aumentar la productividad.
- Evitar la realización de trabajos pesados y repetitivos para el ser humano.

- Acoplarse rápidamente por sustitución de la mano de obra obteniendo, así, una mayor duración de las herramientas, más precisión en los trabajos realizados, menos pérdida de material y reducido mantenimiento.
- Realización de tareas en condiciones y ambientes peligrosos para el ser humano (hostiles, a muy altas o muy bajas temperaturas, etc.).

6.5 Plataforma de simulación virtual.

Aquella herramienta con la cual se puede realizar un prototipo virtual capaz de emular un brazo real dependiendo de las características y su aplicación. Existen diferentes tipos de plataformas para la simulación de un brazo robótico, entre ellas destacan Autodesk Inventor, Solidworks, EON, MATLAB, entre otros. En la figura 2 se observa un ejemplo de una simulación de un brazo robótico [3].

Figura 2. Modelo de un brazo robótico mediante Solidworks, Simscape, MATLAB



6.6 Cinemática de un robot.

La cinemática es una ciencia que estudia la mecánica del movimiento de un objeto haciendo abstracción de las fuerzas que interactúan. Se toma como fundamento la geometría y se recurre al álgebra lineal para interpretar de manera matemática, la configuración de los eslabones de un robot con respecto a su estructura completa.

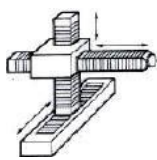
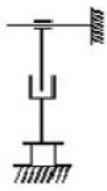
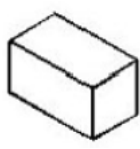
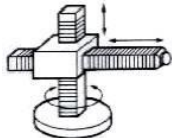
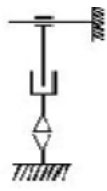
Entre las formas desarrolladas para la definir la ubicación de un robot con respecto a un plano de referencia. Se enumeran la cinemática directa y la cinemática inversa, ambas se complementan mutuamente. Su naturaleza se diferencia en el enfoque desde el cual se desarrollan. La primera, arranca desde el cálculo de la posición final, específicamente la posición del mecanismo conocido como efector final, el cual ejerce la tarea de manera directa, ya sea a manera de tenaza, taladro,

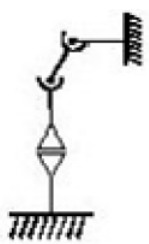
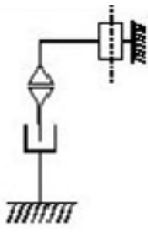
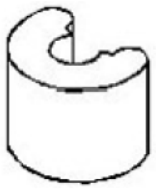
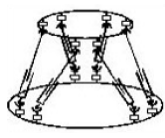
soldadura etc. La segunda, inicia el cálculo de la posición de las respectivas articulaciones tomando en cuenta una posición conocida del efector final. En el análisis de movimiento, generalmente se recurre a ambas para comprobar los cálculos y llegar a resultados similares.

La clasificación de movimientos de los brazos robóticos compone formas rotacionales o lineales, con las cuales se categoriza cada articulación o también llamada grado de libertad. El conjunto de las articulaciones forma la cadena cinemática, que genera un margen de movimientos específicos.

En la tabla 1 [9] se muestra las principales configuraciones de brazos robóticos. A su lado se provee de información relevante, en cuanto a su geometría y aplicaciones.

Tabla 1. Diferentes configuraciones de los brazos robóticos. (Fuente: [9])

Geometría	Diagrama del sólido libre	Espacio accesible	Ejemplo	Comentarios
Cartesiana 				Simplicidad, alta precisión y rigidez. Aplicaciones: -Máquinas CNC -Impresión 3D
Cilíndrica 				Simplicidad, alta precisión y rigidez. Aplicaciones: -Traslación horizontal de piezas (Pick&Place)
Esférica 				Primer brazo con capacidad para trabajar en planos inclinados. Aplicaciones: Soldaduras por puntos. Manipulación de partes y fundiciones en cadenas de montaje.

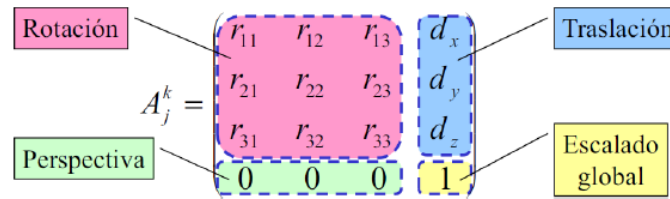
Angular o antropomórfica 				Versión más completa de brazo robótico. Aplicaciones: Paletizado. soldadura por arco y puntos, pintura etc.
Tipo SCARA 				Versión mejorada del diseño cilíndrico, mayor espacio interior. Aplicaciones: -Traslación horizontal de piezas (Pick&Place)
Paralela 				El endeffector es la misma base a la cual se pueden acomodar los elementos. Aplicaciones: Simulaciones de vuelo, micro manipuladores

6.6.1 Cinemática directa.

Para comenzar el análisis, se inicia con la cinemática directa. Primero se establecen los frames o ejes de referencia para cada articulación presente en el robot. Se aplica el algoritmo de Denavit-Hartenberg, mediante el cual se obtienen matrices de transformación homogénea para cada grado de libertad. Cada matriz homogénea tendrá la información de la posición, orientación, perspectiva y escala de sus ejes coordenados correspondientes o frames con respecto a ejes anteriores o de referencia.

La estructura de la matriz de transformación homogénea se presenta en la figura 3

Figura 3. Estructura de una matriz homogénea. (Fuente:[16])



6.6.2 Algoritmo Denavit – Hartenberg.

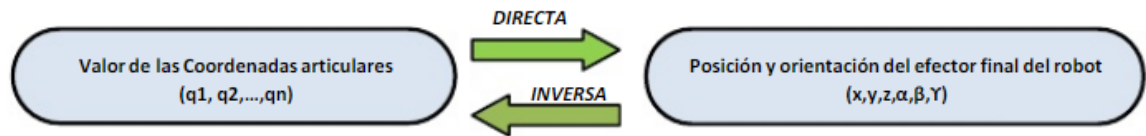
Este algoritmo es ampliamente utilizado para obtener el modelo cinemático directo de un brazo robot. Dicho algoritmo es un método matricial que permite establecer sistemas de coordenadas para cada eslabón del brazo robot. Así, mediante transformaciones de rotación y traslación se pueden conocer las coordenadas cartesianas (X_n , Y_n , Z_n) de la herramienta, referido a un sistema de coordenadas fijo con base en los ángulos de todos los elementos del brazo robot. Luego se calculan los parámetros del robot y sus variables de articulación, para generar la matriz cinemática. Explicaciones amplias sobre el algoritmo de Denavit-Hartenberg se encuentran en diversas fuentes, como [1], [15], [16].

6.6.3 Cinemática inversa.

La cinemática inversa de un manipulador es un término usado para denotar el cálculo de los valores articulares (ángulos de las juntas) del manipulador, necesario para posicionar un punto en el espacio referenciado al sistema de coordenadas global del manipulador. Para este caso, se calcularon los valores de θ_1 , θ_2 y θ_3 basados en el punto (P_x , P_y y P_z).

El proceso del desarrollo de la cinemática del robot se basa en el problema de hallar la posición y orientación del efector final del robot (último eslabón) para los valores de las coordenadas articulares del robot (cinemática directa) y viceversa (cinemática inversa). En la figura 4 se observan los parámetros y la relación entre las dos cinemáticas.

Figura 4. Relación entre los parámetros de las cinemáticas directa e inversa. (Fuente: [16])



6.7 MATLAB.

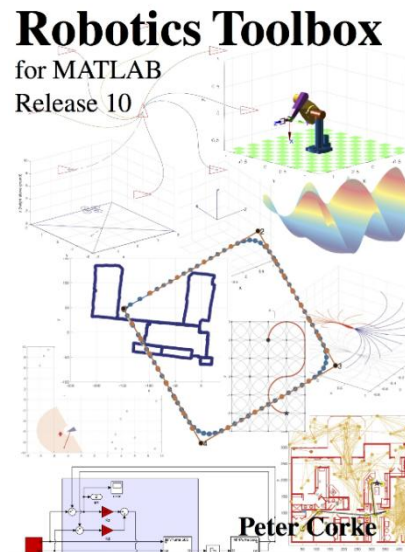
Cuyas siglas llevan al significado de “Matrix Laboratory”, es un software que implementa funciones de cálculo matricial para brindar soluciones matemáticas a distintas áreas de la ingeniería. En ese sentido, el equipo de Mathworks [22] ha incursionado a través de sus toolboxes en visión artificial, sistemas de control, comunicaciones, redes neuronales etc.

Dado el contenido matricial que requiere el cálculo de la cinemática directa del brazo robótico, Matlab se convierte en la primera opción para trabajar con las transformaciones homogéneas que presenta el trabajo. Un primer inconveniente que se presenta al trabajar con esta plataforma es que se requiere cierta capacitación, pues Matlab posee su propio lenguaje de programación, aunque sea de alto nivel, además de la programación gráfica que se ofrece a través de simulink incluyendo todas sus librerías y cómo usarlas. Sin embargo, existe extensa documentación accesible desde el mismo espacio de trabajo, así como un conjunto de comandos para las distintas actividades de la ingeniería como calcular promedios, trabajar con variables simbólicas o aplicar transformadas.

6.8 Robotics de Peter Corke.

Peter Corke desarrolló una librería o toolbox para robótica llamado “Robotics System Toolbox” que incluye algunas de las operaciones que normalmente se usan en robótica industrial y es ofrecida de manera libre en su página oficial junto a la documentación necesaria para su manejo. Esta librería es muy mencionada en la literatura y constituye una respuesta completa para las necesidades en desarrollo robótico. Cuenta además con una sección para trabajar con bloques en el entorno Simulink de MATLAB. El cálculo de la cinemática directa e inversa se nos facilita mucho a través de esta librería, y con sus bloques de Simulink nos permite acceder a los cálculos a la vez que se integran las distintas etapas de procesamiento y comunicación del proyecto.

Figura 5. Carátula del manual de librería Robotics. (Fuente: [25])



6.9 Solidworks.

Acerca de una definición de Solidworks [17]: “SOLIDWORKS es un software de diseño CAD 3D (diseño asistido por computadora) para modelar piezas y ensamblajes en 3D y planos en 2D. Sus productos ofrecen la posibilidad de crear, diseñar, simular, fabricar, publicar y gestionar los datos del proceso de diseño.”

A través de Solidworks se puede crear un modelo en gráficos de computadora del brazo robótico a partir de las dimensiones y características mecánicas de un modelo industrial real. La posibilidad que brinda este software de exportar en distintos formatos el objeto 3D, facilita la selección de un formato que se ajuste a la integración.

6.10 Android studio.

La documentación disponible de la página de Android ofrece una descripción de la composición de este software [18]:

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA . Algunas De las funciones de Android son las siguientes:

- *Un sistema de compilación basado en Gradle flexible*
- *Un emulador rápido con varias funciones*
- *Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android*
- *InstantRun para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK*
- *Integración de plantillas de código y GitHub para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código*
- *Gran cantidad de herramientas y frameworks de prueba*
- *Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.*
- *Compatibilidad con C++ y NDK*
- *Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine*

La participación en el mercado de teléfonos móviles inteligentes del sistema operativo Android es la más alta para el 2018 como lo muestra la siguiente tabla de GartnerInc, uno de los principales consultores en tecnologías de la información.

Ventas mundiales de smartphones a usuarios finales por sistema operativo en 2018 (miles de unidades)

Tabla 2. Estadísticas de mercado mundial de smartphones en Agosto de 2018(Fuente: [24])

Sistema Operativo	Unidades en 2018	Cuota de mercado de 2018 (%)	Unidades en 2017	Cuota de mercado de 2017 (%)
Android	329,503.4	88.0	321,848.2	87.8
iOS	44,715.1	11.9	44,314.8	12.1
Otros SO	112.1	0.0	433.1	0.1
Total	374,330.6	100.0	366,596.1	100.0

Estas características posicionan al Android Studio como el principal entorno de desarrollo de aplicaciones para dispositivos móviles, así como, plantean la necesidad, para los desarrolladores, de profundizar y actualizarse en este entorno.

En este proyecto, se trabaja entonces con Android Studio para el desarrollo de la aplicación, su interfaz gráfica propia para la manipulación de los grados de libertad del brazo robótico, así como la configuración de una comunicación a través de Bluetooth.

6.11 Bluetooth.

Es una tecnología de comunicación inalámbrica a corta distancia que ha tomado un nuevo impulso con la introducción de tecnologías móviles. Continúa evolucionando y mejorando sus prestaciones. La última versión es la 5.0. Se describe el estado del arte en [19]:

“Bluetooth es un método para la comunicación que usa señales de radio de corto alcance a fin de eliminar los cables a la hora de interconectar equipos de cómputo y dispositivos en general. Esta tecnología tuvo su inicio en 1994 por L. M Ericsson. El estándar fue nombrado bluetooth gracias al rey de la mitología nórdica que lleva el mismo nombre. El Bluetooth SIG (SpecialInterest Grupo) fue fundado por Ericsson, IBM, Intel, Nokia y Toshiba en febrero de 1998 a fin de desarrollar una especificación abierta de conectividad inalámbrica de corto alcance. En la

actualidad más de 1900 compañías pertenecen al SIG, entre ellas 3Com, Motorola, Lucent, etc.

Bluetooth pretende proveer ventajas significativas sobre otras tecnologías de transferencia de datos, como IrDA y HomeRF. A diferencia de IrDA Bluetooth no requiere contacto directo con otro dispositivo para transferir datos. Además de ello provee velocidades más altas de transmisión, teniendo un tope máximo de transmisión de datos de 780kb/s y hasta 432.6kb/s de datos simétricos y seguridad garantizando así la integridad de la información que es transferida. Bluetooth está diseñado para ser implementado a bajo coste, siendo esta característica la de mayor impacto ante la popularización de bluetooth. La principal fortaleza de la tecnología bluetooth es su habilidad de mantener simultáneamente transferencia de voz y datos. Es capaz de soportar un canal de datos asíncrono y hasta tres canales sincrónicos de voz y un canal mixto de transferencia de voz y datos. También ofrece el servicio de descubrimiento automático de dispositivos en su alcance siendo esto muy útil para dispositivos móviles.

Para el actual proyecto, se desarrolló una comunicación con un perfil de tipo SPP [19]: "Bluetooth Core Versión especificación 2.1 + EDR es totalmente compatible con 1.2, y fue adoptada por el Bluetooth SIG (Bluetooth SpecialInterestGroup) el 26 de julio de 2007. La función de titular de la 2.1 es Secure Simple Pairing (SSP): se mejora la experiencia de emparejamiento de dispositivos Bluetooth, mientras que el aumento del uso y la fuerza de seguridad"

De esta manera, los dispositivos Bluetooth que se usan en este proyecto deben ser de una versión 2.1 en adelante.

7. DESARROLLO DEL PROYECTO

7.1 Apuntes metodológicos.

La caracterización del presente trabajo es de desarrollo tecnológico, sin embargo, su producto es totalmente en software. No hay implementación hardware. En ese sentido, la esencia del método se encuentra en la revisión bibliográfica tanto de antecedentes, trabajos similares entre proyectos de grado y artículos, como de la documentación de las herramientas del software utilizado. Esto, combinado con los conocimientos teóricos adquiridos en el curso del programa académico, así como durante la elaboración del proyecto.

En cuanto al desarrollo como tal, se plantea en primer lugar la construcción del objeto a controlar: un brazo robótico virtual de cinco grados de libertad. Para ello se toman en cuenta los modelos de empresas de fabricación de robótica industrial

y posteriormente se utiliza software de diseño mecánico para crear el objeto en 3D, y poder virtualizarlo.

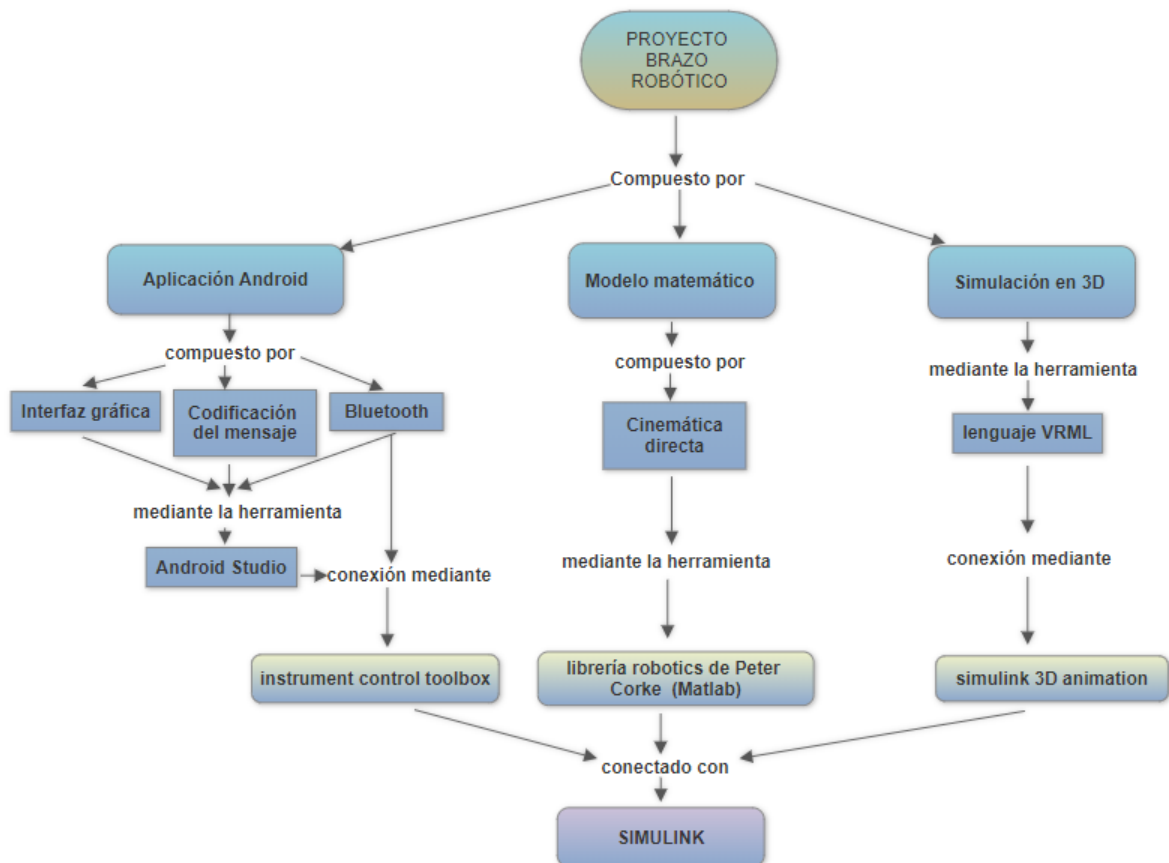
En segundo lugar, se diseña una aplicación con una interfaz apropiada para la manipulación de cinco variables. Se debe configurar para permitir una comunicación inalámbrica.

En tercer lugar, se utiliza la versión de programación gráfica de Matlab, para crear la respectiva integración de modo que se puedan recibir los mensajes enviados desde el dispositivo móvil hacia el robot virtual y se añade la etapa de procesamiento.

7.2 Esquema general del proyecto.

Este es un proyecto de integración, de manera que programas, lenguajes y en general software con distintas estructuras deben conectarse, de manera que se requiere un entorno que permita una fácil comunicación de las tecnologías.

Figura 6. Esquema de estructura general del proyecto(Fuente: propia)



A través de la herramienta simulink y sus diferentes toolbox se puede integrar las tres grandes partes que aparecen en la figura 7. El “Instrument Control Toolbox” proporciona la posibilidad de integrar señales que provienen de dispositivos bluetooth como es el caso del móvil del proyecto. Y así mismo proporciona la posibilidad de virtualizar a través del “Simulink 3D animation”.

En cuanto a la parte de simulación en 3D, dentro del software más usado para el diseño de sólidos y figuras en tres dimensiones se encontró Solidworks y Autodesk Inventor, entre ellos se seleccionó Solidworks por su posicionamiento en el mercado.

7.3 Modelo mecánico.

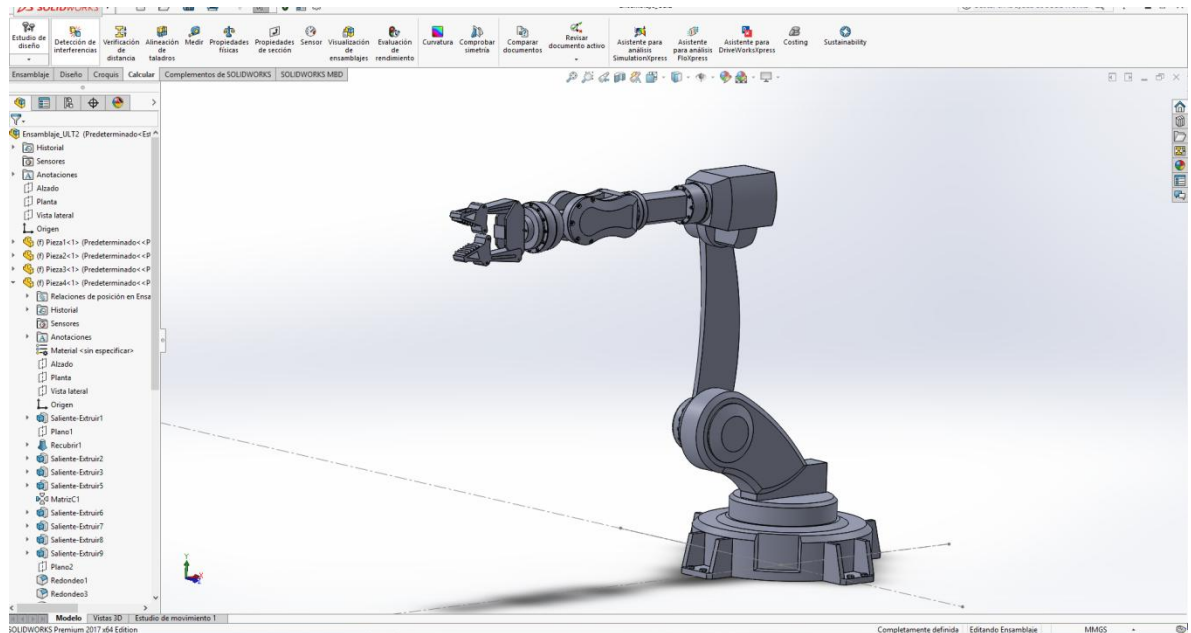
Se escogió, un modelo en 3D genérico [20], sin marca específica, diseñado para gráficos de computadora y basado en dimensiones de un robot de 6 grados de libertad de ABB [21], ya que se utiliza en la red para análisis y de manera didáctica y cumple con los requisitos planteados de manera que sirve como base para la virtualización.

Figura 7. Robot 3D [20]



El principio de extrusión en Solidworks permite crear de manera rápida y sencilla un sólido. Solidworks brinda, al inicio, tres opciones de archivo para iniciar el trabajo. Las distintas partes del robot se hicieron en archivos de tipo pieza: la base, el brazo, el antebrazo, etc. Al final se crea un nuevo archivo en modo ensamblaje, en el cual se importan todos los archivos con las piezas diseñadas y el programa permite juntarlas. El resultado es una estructura que se puede mover a partir de las articulaciones que se han fusionado. La figura 8 muestra el prototipo en el entorno SolidWorks.

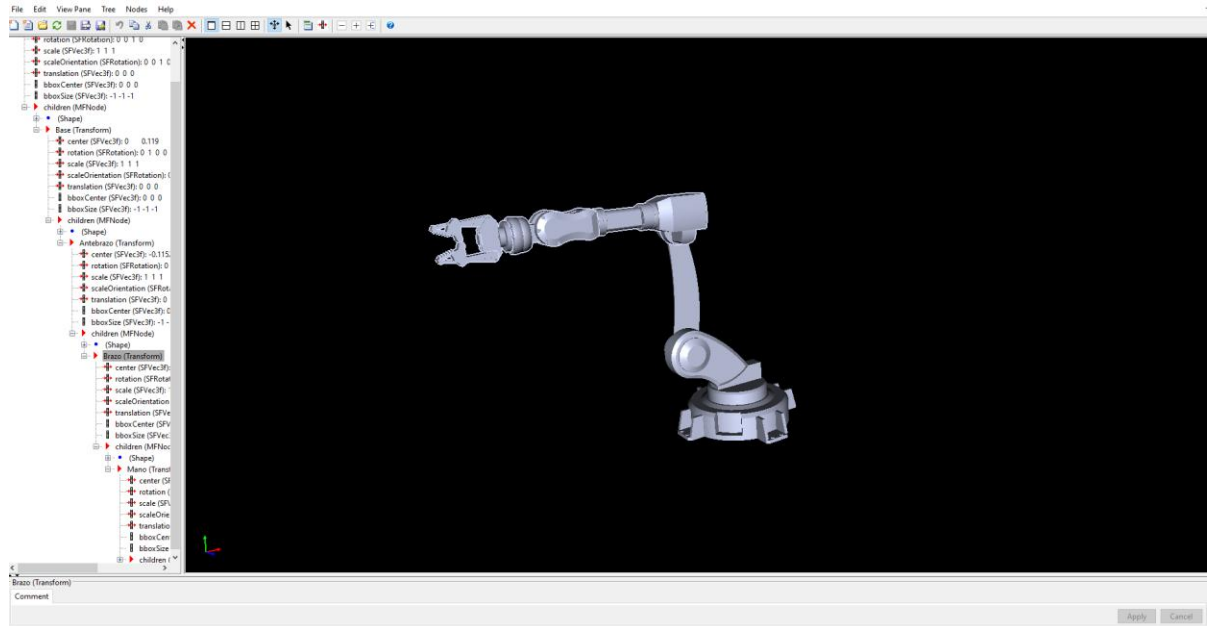
Figura 8. Robot virtual en entorno SolidWorks. (Fuente: Propia)



7.4 Conversión del diseño mecánico a archivo de realidad virtual.

Solidworks permite guardar el archivo de ensamblaje en muchos formatos, entre ellos como archivo VRML, con extensión (.vrl). Al guardar se debe ir a opciones avanzadas para asegurarse de que la versión de vrml es la 2.0 o VRML97, además se debe asegurar de que el sistema de coordenadas con el cual se va a importar el proyecto es el escogido o el que tiene por defecto según la necesidad del caso. La figura 9 muestra el robot virtual en formato VRML en el entorno 3D World Editor.

Figura 9. Robot virtual en formato VRML en el entorno 3D World Editor. (Fuente: propia)



7.5 Virtualización del brazo robótico.

Dado que Simulink es el entorno con el cual se unen todos los procesos, se utiliza el SIMULINK 3D ANIMATION para integrar la virtualización en el sistema. El bloque que se utiliza para este efecto al interior del Toolbox tiene el nombre de “Sink” el cual genera un entorno para visualizar el objeto de realidad virtual. Cuando el bloque está vacío se abre una ventana de configuraciones en la cual se puede buscar el archivo con extensión (.wrl) que se genera desde Solidworks. Cuando ya está enlazado el archivo, existe un botón para editar el archivo. La edición de este archivo es un procedimiento necesario para los fines del proyecto, ya que de esta manera se pueden definir las variables de entrada para controlar los grados de libertad y la forma del movimiento que tendrá la virtualización. Matlab viene con editores de archivos vrml entre ellos v-builder y Open 3D WorldEditor. Por defecto la edición se abre en Open 3D World, una plataforma suficiente para las tareas planteadas.

Al editar el archivo VRML se abre una estructura de árbol a la izquierda y la visualización del objeto virtual a la derecha. En la parte inferior se pueden generar los cambios en los valores que contiene cada rama del árbol. Esta es la estructura básica del editor mencionado.

Un nodo es el elemento básico en el lenguaje VRML. Al importar el archivo de ensamblaje en formato VRML, Solidworks crea un nodo de tipo “transform” para

cada parte ensamblada, generando de esta forma en el archivo vrml editable, como una lista de tantos nodos como piezas tenía el ensamble.

En primer lugar, se deben definir los nodos, esto implica darles un nombre a través del comando DEF, si no se efectúa esta tarea, no puede acceder desde el exterior a los nodos de los objetos virtuales.

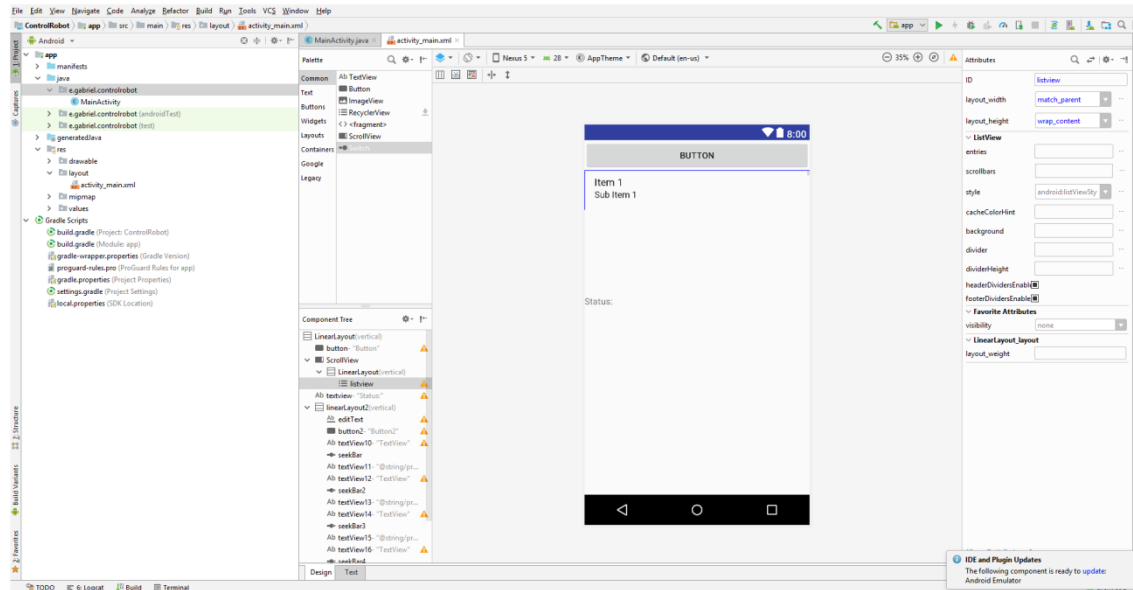
Luego de la marcación, se debe crear una jerarquía en los nodos de manera que lo que suceda a unos tenga una influencia sobre los otros, pues Solidworks genera un archivo con nodos independientes unos de otros. Esto se realiza cambiando el nodo a un lugar en la estructura del árbol que sea dependiente de otro. En VRML se maneja una estructura de nodo padre y nodo hijo. La acción se propaga a través de los distintos nodos, de manera que, al mover la base, todos los demás nodos de la estructura, que son todos hijos del nodo base, deberán moverse con ella.

Al guardar los cambios en la edición del archivo VRML se puede volver a la ventana de configuraciones del bloque SINK, a la derecha de la ventana aparecerá la estructura de árbol del archivo VRML cargado y se tiene la opción de adicionar un chulo en unas cajas de cada ítem de esa estructura. Debido a que las articulaciones del robot son todas rotacionales, entonces el ítem que se señala en cada nodo será el ROTATION, al aceptar esta configuración el bloque SINK cargará la virtualización y tendrá cinco entradas desde las cuales se pueden manipular cinco rotaciones de las partes del objeto virtual.

7.6 Creación de la aplicación android.

Se seleccionó como software de diseño para la aplicación móvil “Android Studio” debido a la gran cantidad de dispositivos móviles que utilizan sistemas Android. Una de las grandes ventajas de este sistema para los desarrolladores es que posee funciones de alto nivel que permiten abstraerse de los elementos de hardware que usualmente posee un teléfono móvil, de manera que se puede maniobrar con ellos sin dificultad a través del lenguaje de programación Java. La figura 11 muestra el entorno para la creación de la aplicación móvil.

Figura 10. Entorno de Android Studio (Fuente: propia)



7.6.1 Bluetooth.

Se utilizaron las funciones que vienen implementadas en java para aplicaciones Android, en las cuales se establece una estructura cliente-servidor. El móvil puede operar como cliente o como servidor, para ello se inicializa un adaptador de Bluetooth, y posteriormente se crea o se “escucha” un canal (dependiendo del papel que se asuma dentro de la estructura) y se genera también un socket para la transmisión de los datos. Se requiere entonces un método para la conexión y otro método para el envío de datos. El papel que asume en la arquitectura de comunicación del dispositivo a través de la aplicación es el de cliente.

7.6.2 Hilos en java.

Cuando se tiene una presentación en la pantalla del móvil, existe un proceso principal que se genera en consecuencia, sin embargo, usualmente se requieren procesos distintos que están asociados con el funcionamiento de la aplicación, pero no directamente con la presentación que se obtiene en la pantalla, es decir, procesos en segundo plano. Para la implementación de estos procesos se utilizan

los hilos (Thread). Y se utiliza el concepto de herencia, dentro del esquema de “programación orientada a objetos” de Java, para que cada método que se desarrolla en el código y que vaya a utilizarse como un proceso herede características de la clase Thread. De esta manera, los métodos utilizados tanto para la conexión como para el intercambio de datos se desarrollan en hilos de procesos y por tanto heredan de la clase Thread.

Uno de los inconvenientes al trabajar con hilos, es que la comunicación entre ellos no se puede hacer directamente a través de variables, para ello se utiliza una especie de mensajero llamado “handler”, de manera que se hace necesario crear otro objeto que herede dichas características y que permita la conexión entre los distintos procesos que se requieren en el manejo del bluetooth.

7.6.3 Interfaz grafica.

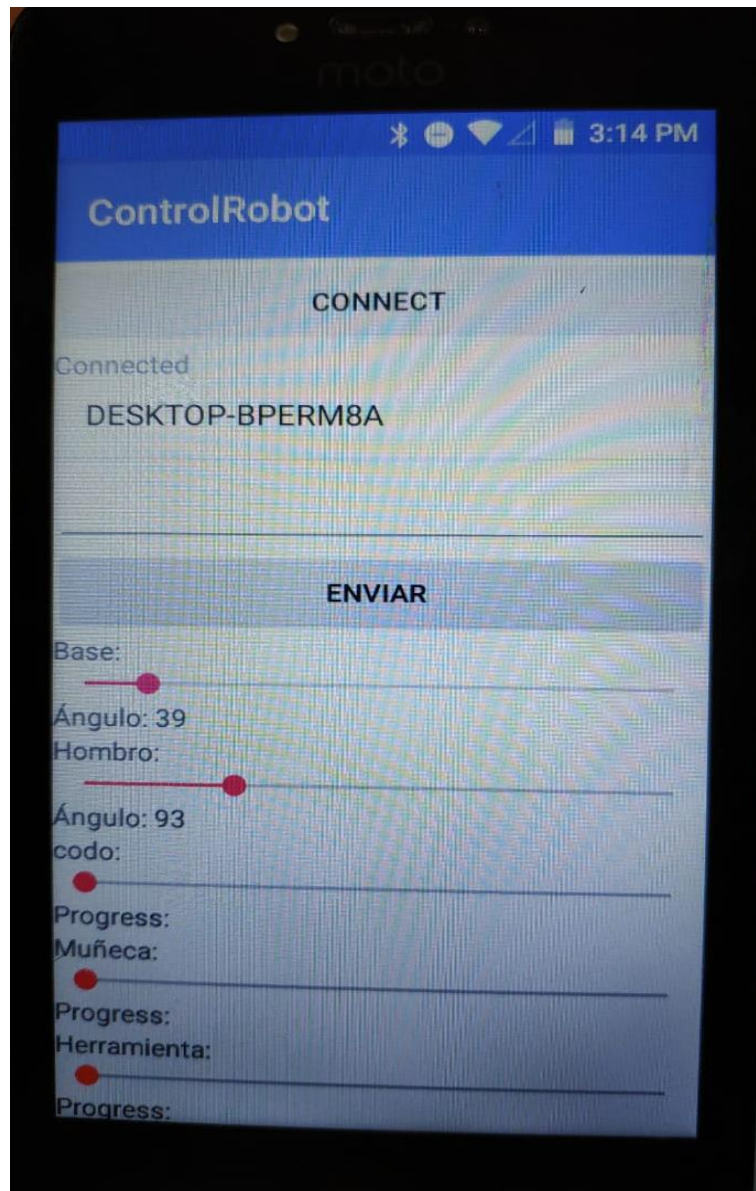
Android permite trabajar la interfaz gráfica de manera independiente a través de lenguaje XML esto brinda muchas facilidades, así mismo posee dos modalidades: texto y diseño, en la segunda de ellas se puede arrastrar y soltar objetos dentro de la presentación lo que ofrece una cómoda estructura.

La interfaz gráfica es sencilla. Consta de una sola presentación o layout. En la parte superior, se tiene un botón cuya acción despliega un listado de los dispositivos emparejados, en la lista se debe seleccionar el dispositivo donde se encuentre la simulación del robot, al presionar sobre la opción el dispositivo intentará conectarse, debajo aparece un texto que indica el estado de la conexión en inglés, ej. connecting, connection failed, connected, etc. Al principio esto es todo lo que se puede ver, sin embargo, en el momento en que la conexión sea exitosa, en la parte inferior aparecerán los controles de los cinco grados de libertad del robot.

Se utiliza para cada control el widget Seekbar de Android, que es una especie de barra de progreso que se puede manipular arrastrándola. De esta forma se genera un enlace más vívido entre la manipulación de los controles por el usuario y el movimiento que se produce en la articulación específica del robot virtual. En la parte superior de las barras aparece el nombre de la articulación que se manipula y en la parte inferior aparece el ángulo de rotación para la articulación que representaría la posición actual en la barra de progreso.

Además, se agregó un espacio en el cual se puede introducir texto y asimismo ofrecer control sobre las articulaciones del robot virtual, el formato deben ser las letras a, b, c, d o e en minúsculas, según el grado de libertad y posteriormente escribir el ángulo en grados al cual se desea mover la articulación y con tres dígitos. Ej. a030, b180. Posteriormente se presiona el botón “enviar”. La figura 11 muestra la interfaz desarrollada.

Figura 11. Interfaz gráfica de aplicación final. (Fuente: propia)



7.6.4 UUID (Universal unique identifier)

En protocolos de comunicación el UUID permite diferenciar con quién hay comunicación y sus características lo convierten en un código que difícilmente pueda repetirse, de esta manera se pueden identificar los dispositivos en un modo de comunicación inalámbrico que puede ser muy abierto. Es fundamental para que la conexión sea exitosa utilizar el UUID correcto.

En la aplicación desarrollada se trabaja con comunicación serial, también conocida como Bluetooth SPP (Secure Simple Pairing). En este tipo de comunicación se maneja un UUID específico: "00001101-0000-1000-8000-00805F9B34FB". Si se utiliza cualquier otro UUID la comunicación fallará.

7.7 Instrument control toolbox.

Además, esta toolbox, Matlab cuenta con la aplicación "test and measurement" que proporciona una interfaz muy práctica, al momento de establecer comunicación con algún instrumento. Esta aplicación implementa a través de una interfaz gráfica las funciones del toolbox, ahorrando mucho código en la etapa de pruebas. Según la documentación, Matlab es compatible con la comunicación Bluetooth con perfil SPP.

Por otro lado, posee una paleta de bloques en simulink, entre ellos un grupo para la comunicación serial. Dado que la integración de todo el sistema se da en SIMULINK se recurre al uso de estos bloques.

Dentro de la arquitectura trabajada, para hacer uso de la aplicación "test and measurement" ideal para las pruebas iniciales, así como en la utilización de las funciones en el "workspace", se requirió que el móvil trabajara como servidor y el pc en el que se corría matlab, trabajara como cliente. Sin embargo, para el uso del toolbox en la parte de los bloques de Simulink, fue necesario invertir la arquitectura y ubicar al pc como servidor, y al móvil como cliente. De otra forma la comunicación fallará.

7.8 Estableciendo conexión.

7.8.1 Emparejamiento.

Antes de usar MATLAB se debe garantizar el emparejamiento entre el móvil y el PC, esto se realiza a través de las aplicaciones que vienen por defecto tanto el móvil como en el PC. El emparejamiento se propone desde el PC luego de intentar agregar un elemento bluetooth en la ventana de configuración del sistema operativo. Para que el móvil sea detectable el bluetooth debe estar encendido en primer lugar y además debe tener activada la opción de ser visible para otros dispositivos, de esta manera el PC intentará emparejarse enviando un código de verificación que debe ser el mismo recibido en el celular, luego de aceptar los mensajes de alerta en ambos dispositivos, se puede pasar a agregar los puertos.

7.8.2 Agregar un puerto.

Es necesario a través del software administrador del bluetooth que se gestione la vinculación de puertos para la comunicación, ya sea un bluetooth interno o externo. Este es un procedimiento sencillo pero importante, pues en la configuración del bloque de comunicación serial será necesario establecer el puerto y si no ha sido adherido con antelación Matlab no lo detectará, para que el programa no genere errores, el procedimiento de agregar puertos debe realizarse antes de abrir Matlab, de otro modo será necesario reiniciarlo.

Debido a que se trabajará el pc en modo servidor y el móvil en modo cliente, el tipo de puerto a configurar debe ser entrante.

7.8.3 El toolbox de comunicación serial.

Se trabaja esencialmente con dos bloques: "configuration" y "serial receive". Se arrastra desde la paleta al espacio de trabajo de simulink y al intentar abrirlo aparece una ventana de configuración, en ella hay que establecer el puerto que se utilizará en la conexión además del formato de salida de los datos debido a que el bloque cuenta con una variable de salida llamada "data". También se puede establecer el tamaño del dato y el período de muestreo.

En cuanto al tamaño del dato, es importante configurar un vector de 4 elementos ya que desde la aplicación se ha configurado la codificación en un literal y luego tres dígitos que representan el ángulo en grados. El formato del dato de salida queda configurado en "uint18" esto debido a que Simulink no maneja caracteres, de manera que el dato saldría en código ascii. Si se toma un bloque "display" de la paleta y se posiciona a la salida del bloque SERIAL RECEIVE enviando por ejemplo el código "a123" en el DISPLAY se visualizaría el siguiente vector [97 49 50 51].

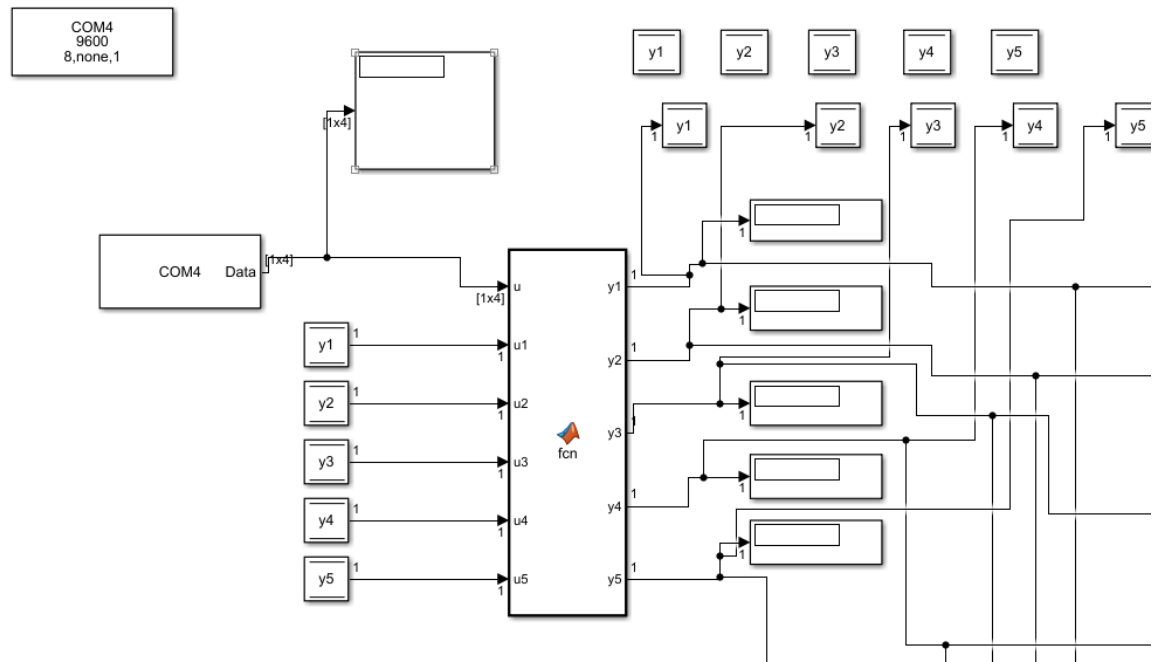
Al aplicar y aceptar la configuración del bloque SERIAL RECEIVE, Matlab nos pregunta si se permite la creación del bloque CONFIGURATION en él sólo se debe especificar de nuevo el puerto que se utilizará para la comunicación, el resto de parámetros de la comunicación serial como la tasa y bit de paridad etc., se puede dejar por defecto.

7.9 Función de procesamiento en simulink.

Para decodificar el mensaje enviado en el vector de 1x4, en primera instancia se pasa de nuevo el mensaje de código ascii a caracteres a través de la función char(), y aplicamos un switch-case examinando el primer carácter de la cadena para 5 posibles casos a, b, c, d, e conociendo así de cual articulación será el dato, posteriormente, se pasa la parte del vector que contiene la información del ángulo a números con la función str2double() y luego de una conversión a radianes enviamos el dato a la salida correspondiente de las cinco disponibles. La función de procesamiento implementada aparece en el anexo 1.

Se requiere un bloque DATA STORE MEMORY para crear variables globales en SIMULINK de otro modo, la función daría reset a todas las salidas que no sean afectadas imposibilitando establecer una configuración de posición al robot virtual. De manera que a cada salida del bloque de la función de procesamiento se ubican un bloque de DATA STORE WRITE, y adicionalmente se abren cinco nuevas entradas a la función para que sirva esto de retroalimentación.

Figura 12. Estructura con función de procesamiento en simulink (Fuente: propia)



7.10 Obtención de parámetros denavit-hartenberg.

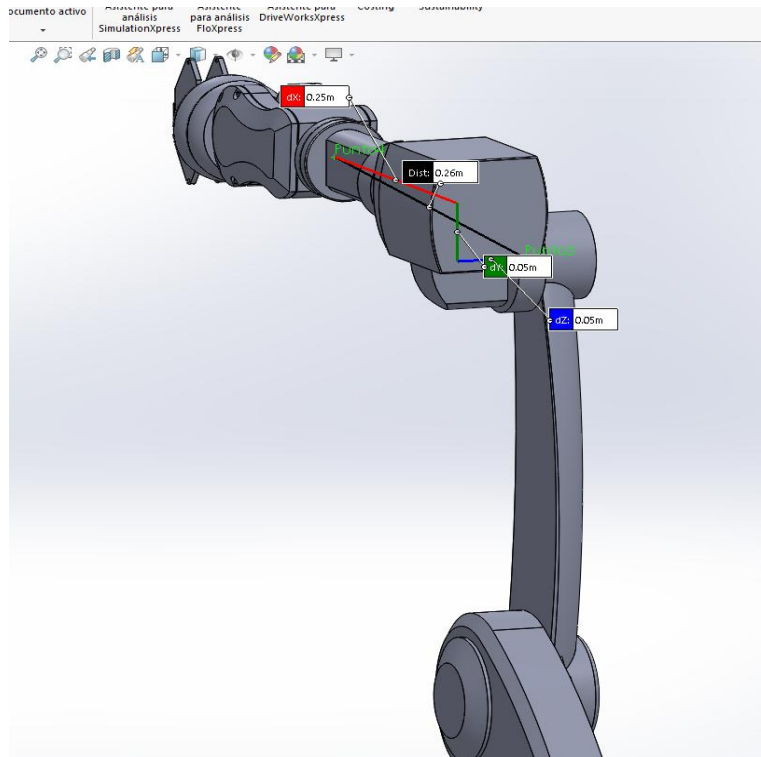
Para el cálculo de la cinemática directa, se usa el algoritmo Denavit-Hartenberg, para generar la tabla de parámetros (ver Tabla 3)

Tabla 3. Parámetros Denavit-Hartenberg del brazo robótico virtual. (Fuente: Propia)

Articulación	Parámetros			
	"a"(m)	"d"(m)	Alpha" α "	Theta" θ "
1	0,16	0,16	$\pi/2$	0
2	0,35	0	0	$\pi/2$
3	0,25	0,05	$\pi/2$	0
4	0,01	0,12	$\pi/2$	π
5	0,15	-0,01	$\pi/2$	0

Para la obtención de estos parámetros se utilizó la herramienta "medir" ubicada en la barra de herramientas del software Solidworks. En la figura 14 se ilustra el procedimiento.

Figura 13. Herramienta medir de solidworks [Fuente:propia]



7.11 Cinemática directa en librería robotics.

Para iniciar, si no se tiene instalada la librería, en la barra de navegación de Matlab se busca la ubicación de la librería y se abre el archivo startup_rvc.m. Luego de compilar este archivo, se puede trabajar con la librería.

Se utiliza el siguiente código en el espacio de trabajo de Matlab para crear el modelo matemático:

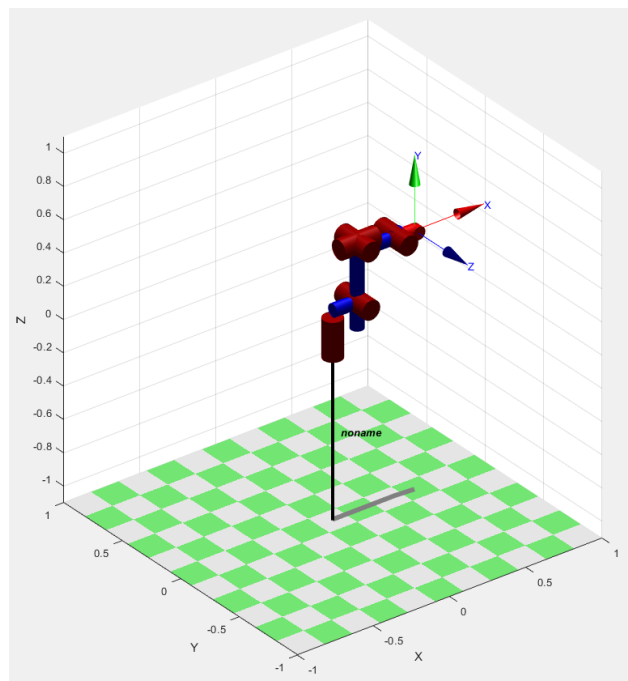
```
L1=Link([0 0.16 0.16 pi/2 0 0]);
L2=Link([0 0 0.35 0 0 pi/2]);
L3=Link([0 0 0 pi/2 0 0]);
L4=Link([0 0.25 0.05 pi/2 0 pi]);
L5=Link([0 0.01 0.12 0 0 pi/2]);
robot_virtual=SerialLink([L1 L2 L3 L4 L5]);
T=robot_virtual.fkine([120 30 20 20 20]);
```

```
y=transl(T);robot_virtual.plot([0 0 0 0 0])
```

El comando Link representa cada enlace entre articulaciones del robot y dentro de sus argumentos se ingresan los parámetros de Denavit_Hartenberg. El comando "SerialLink" une todos los enlaces, generando el modelo matemático del robot completo. El comando "fkine" realiza el cálculo completo de la cinemática directa tomando como argumento los ángulos de todas las articulaciones contempladas. El comando "transl" nos convierte la transformada homogénea que resulta del cálculo de la cinemática directa en un vector con la posición cartesiana del efector final del brazo robótico. El comando "plot" activa una visualización gráfica con la que cuenta la librería para mostrar el procedimiento.

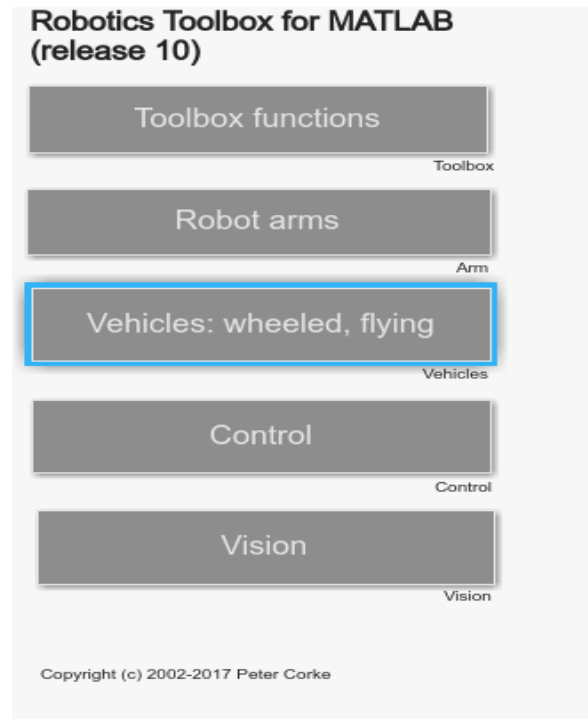
La siguiente imagen es la graficación del modelo matemático que ofrece la librería robotics al compilar el objeto robot, con grados aleatorios para las articulaciones.

Figura 14. Gráfica del modelo matemático en robotics.
(Fuente: propia)



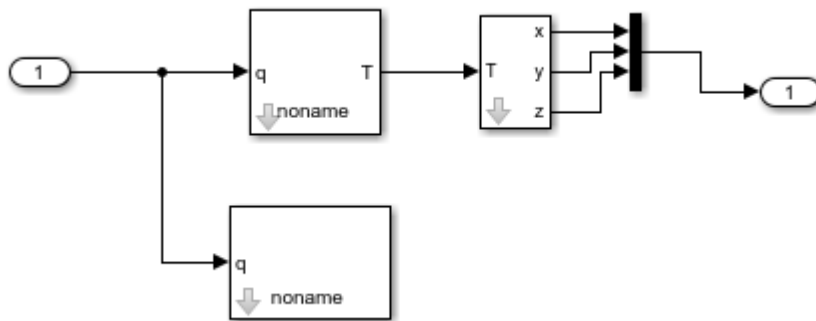
Una vez creado el objeto se trabaja con simulink que es la plataforma que permite la integración. Por tanto, se invoca el conjunto de bloques de la librería a través del comando "roblocks", el resultado es la figura 16.

Figura 15. Paleta en Simulink de Robotics.



En la paleta están los bloques con las funciones de cinemática directa, conversión de transformación homogénea a vector y graficación. Bajo el siguiente esquema se transforma un vector de entrada con 5 diferentes ángulos para los grados de libertad en un vector de salida que indica las coordenadas cartesianas, como se muestra en la figura 17.

Figura 16. Arreglo en simulink de bloques de la librería robotics. (Fuente: propia)



Al final se comprime todo el sistema anterior en un solo bloque y se conecta a una pantalla de salida que permite visualizar las coordenadas del efector final.

Figura 17. Bloque de Cinemática directa.(Fuente: propia)

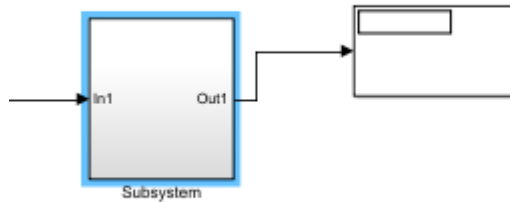
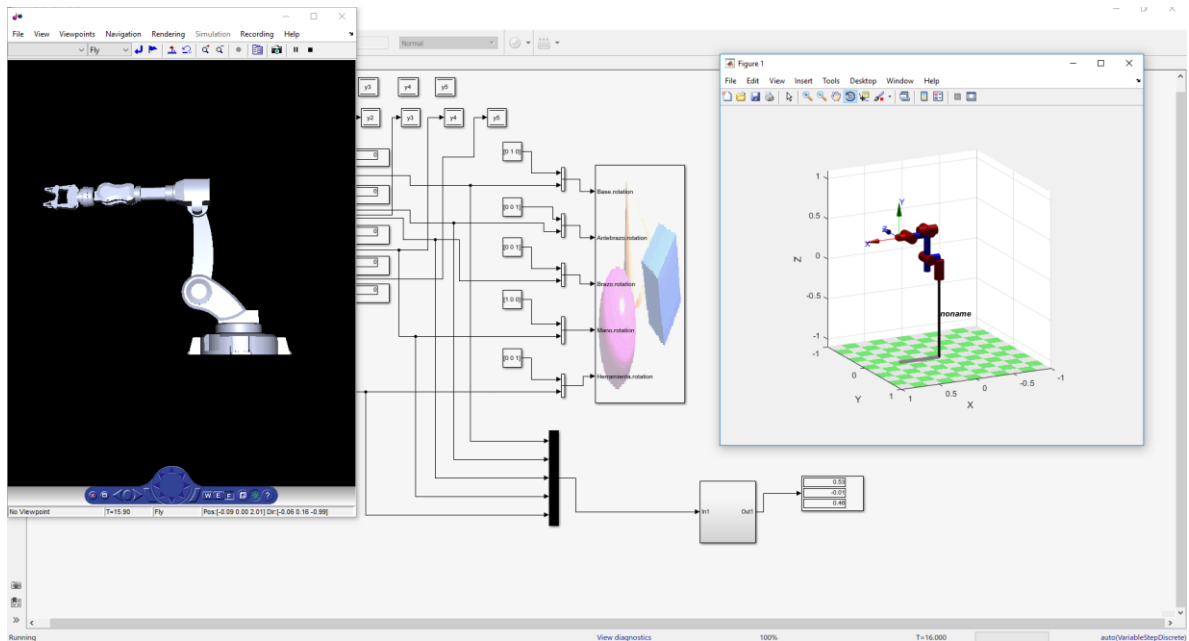


Figura 18. Simulación del entorno completo: Bloques de simulink, brazo robótico virtual y gráfica de modelado matemático con librería Robotics (Fuente propia)



8. RESULTADOS OBTENIDOS

Fue posible hacer la integración de todo el sistema, las dificultades las constituyeron los empalmes, cada entorno, tecnología, software o hardware presentan ciertos estándares. Un ejemplo de dicha situación fue la tecnología

Bluetooth, en un principio se trabajó con un dispositivo Bluetooth externo versión 2.0, después de varios intentos fallidos de emparejamiento, se descubrió en el estado del arte que sólo se podría lograr la comunicación con una versión de Bluetooth de 2.1 en adelante, debido a la incorporación a partir de esta versión del perfil SPP. El mismo inconveniente se presentó en un par de computadoras portátiles marca DELL con tecnología Bluetooth muy antigua.

Se presentaron muchos problemas en la comunicación serial, uno de ellos fue utilizar el UUID correcto para dicho perfil. Al principio se probaron UUID particulares, sólo la profundización en la teoría produjo la respuesta. Otro problema de comunicación fue la alternancia en la estructura cliente-servidor que maneja el INSTRUMENT CONTROL TOOLBOX de MATLAB, tanto en las funciones que se desarrollan en código como en la aplicación de pruebas TEST AND MEASUREMENT, el PC debe actuar como cliente para que el proceso sea funcional, sin embargo, en el entorno de los bloques del mismo toolbox en SIMULINK, es necesario que el PC adopte la postura de servidor en la arquitectura.

La respuesta del robot virtual ante el mando ejecutado en la aplicación móvil fue menor a un segundo antes de la implementación de la cinemática directa. Los cálculos matriciales llevaron a un retraso considerable de alrededor de tres segundos para la versión que incluía la cinemática directa. Esto está totalmente relacionado con la capacidad de proceso de la unidad de cómputo que se tenga a disposición para la simulación. Sin embargo, se ilustra de esta manera, que el principal “cuello de botella” en cuanto a retardos e imposibilidad de un acercamiento a un modelo de tiempo real, se encuentra en los cálculos realizados para la cinemática directa.

Enlace vídeo de funcionamiento:

<https://bit.ly/2XY8WJq>

9. RECOMENDACIONES

Se plantearon posibles variaciones y desarrollos conforme el trabajo avanzaba, la posibilidad de incluir más información que proviniese del robot virtual hacia la aplicación y la posibilidad de establecer más controles desde la aplicación hacia el robot virtual. Y poco a poco intentar migrar la simulación desde el PC hasta el móvil, empezando por la animación hasta entrar en los cálculos más sencillos.

El trabajo propone una continuación en mejores formas de implementación de la cinemática inversa, incluyendo los materiales del robot, medidas, masa, entre otros para proponer el modelado dinámico y de esta manera indagar en las posibles estrategias de control. Además, aprovechar las bondades de Android para proponer una interfaz más robusta y con diferentes formas de visualización, incluyendo algunas formas de restricción para las variaciones de los ángulos de algunas articulaciones.

10. CONCLUSIONES

Se realizó una revisión documental sobre las diferentes aplicaciones móviles orientadas a la manipulación de sistemas mecatrónicos que permitió evaluar los mecanismos gráficos para la manipulación, así como métodos de comunicación y ejemplos de programación.

Se construyó un brazo robótico virtual mediante un software de diseño mecánico, útil para las intenciones de virtualización y de cálculo de la cinemática directa.

Se desarrolló una aplicación móvil que permite la comunicación y maniobra de las articulaciones del brazo robótico virtual en sus cinco grados de libertad, mediante la configuración de parámetros relacionados con su ángulo de giro y con una velocidad de respuesta que, a partir de la observación, deja un retardo de poca visibilidad.

Se comprueba la potencialidad del desarrollo de aplicaciones móviles de la mano de la ingeniería. En la medida en que los dispositivos móviles continúen aumentando su capacidad de proceso, memoria y almacenamiento, es posible que aplicaciones posean cálculos cada vez más complejos, dando paso a simuladores más completos y didácticos, accesibles a cualquier estudiante y en los cuales se estrecha la interrelación de los conceptos robóticos con elementos tan personales y cotidianos como lo son estos dispositivos.

11. BIBLIOGRAFÍA

- [1] Craig, J. (2006). "Robótica". Pearson Education, 3a edition
- [2] J. Angeles and J. Angeles. (2002) "Fundamentals of robotic mechanical

systems”. Springer, vol. 2.

- [3] Simona Sofia Duicu, Luminita Popa (2011) “Modelado y simulación de la trayectoria del brazo robótico de 5 ejes”: [Online]. Available: <https://pdfs.semanticscholar.org/9d22/aaa3c8ad417bb20cfbf1c75731eb55fd ee14.pdf>
- [4] Ling Wen, Yu Zhang, Lufeng Luo, Cong Zhang (2013) “Simulador de movimiento de cinco grados de libertad basado en entorno virtual” [Online]. Available: http://file.scirp.org/Html/4-1860082_28254.htm
- [5] Mohammad Afif Ayoba, Wan Nurshazwani Wan Zakaria, Jamaludin Jalani, Mohd Razali Md Tomari (2015) “Modelado y simulación de un robot RV 2AJ de 5 ejes utilizando simmechanics”. [Online]. Available: https://www.researchgate.net/publication/283837618_Modeling_and_simulation_of_a_5-axis_RV_2AJ_robot_using_simmechanics
- [6] M. Arteche, “Programación de un robot industrial Scara mediante un dispositivo móvil,” pp. 1–39, 2012.
- [7] J. M. Cervantes, “Aplicaciones Android para el control de sistemas mecatrónicos,” *4º Congr. Virtual Int. sobre Tecnol. Educ. y Soc.*, pp. 1–20, 2015.
- [8] G. Juan, “Teleoperación del robot NAO mediante dispositivos móviles Android,” 2012.
- [9] Marc Ruiz Vinyeta, Ricard Llauger Roca.(2017) Trabajo de Grado: “Control de un brazo articulado mediante dispositivo móvil”. Universidad de Cataluña.
- [10] J. L. Hernández Soriano, “Control de un brazo robótico mediante un dispositivo de cómputo móvil vía bluetooth,” Mar. 2017.
- [11] E. J. Sánchez Sánchez and L. A. Pérez Llerena, “Diseño del controlador y tele-operación del brazo robótico “Robotic Arm Edge” por medio de un dispositivo Android y PC,” 2013.
- [12] J. Medina, N. Castro, and E. Mejía, “Aplicación móvil para el control de un brazo robot,” *Rev. Iberoam. Prod. Académica y Gestión Educ. Apl.*, p. 23, 2016.
- [13] A. Maity, “Android Application Based Bluetooth Controlled Robotic Car,” *Int. J. Intell. Inf. Syst.*, vol. 6, no. 5, p. 62, 2017.

- [14] M. Hernández-Ordoñez, M. A. Nuño-Maganda, C. A. Calles-Arriaga, O. Montaña-Rivas, and K. E. Bautista Hernández, “An Education Application for Teaching Robot Arm Manipulator Concepts Using Augmented Reality,” *Mob. Inf. Syst.*, vol. 2018, pp. 1–8, 2018.
- [15] Reyes, F. (2012). “MATLAB Aplicado a Robótica y Mecatrónica”. Alfaomega Grupo Editor, S.A. de C.V.
- [16] Spong, M., & Vidyasagar, M. (1989). “Robot dynamics and control” John Wiley & Sons, Inc
- [17] SolidBi, “SOLIDWORKS - Qué es y para qué sirve,” 2017. [Online]. Available: <https://solid-bi.es/solidworks/>.
- [18] Android Studio, “Conoce Android Studio | Android Developers.” [Online]. Available: <https://developer.android.com/studio/intro/?hl=es-419>.
- [19] L. Avila and C. Reyes, “Revisión estado del Arte de la tecnología Bluetooth,” *Rev. Investig. y Desarro. en TIC*, vol. 3, no. 2, pp. 1–7, 2017.
- [20] TURBOSQUID, “3d model industrial robot modeled.” [Online]. Available: <https://www.turbosquid.com/3d-models/3d-model-industrial-robot-modeled/839886>.
- [21] Asea Brown Boveri, “Especificaciones del producto. Robot articulado IRB 6400RF,” p. 46, 2006.
- [22] MATHWORKS, “Virtual Reality Modeling Language (VRML) - MATLAB & Simulink.” [Online]. Available: <https://www.mathworks.com/help/sl3d/vrml.html>.
- [23] “Procesos y subprocesos | Android Developers.” [Online]. Available: <https://developer.android.com/guide/components/processes-and-threads?hl=es-419#Threads>.
- [24] Gartner, “Gartner Says Huawei Secured No. 2 Worldwide Smartphone Vendor Spot, Surpassing Apple in Second Quarter 2018.” [Online]. Available: <https://www.gartner.com/en/newsroom/press-releases/2018-08-28-gartner-says-huawei-secured-no-2-worldwide-smartphone-vendor-spot-surpassing-apple-in-second-quarter>.
- [25] “[petercorke.com] – Resources for robotics education: code, books and MOOCS.”

Anexos:

Anexo 1. Código Matlab de función de procesamiento.

```
function [y1,y2,y3,y4,y5] = fcn(u,u1,u2,u3,u4,u5)
x=char(u);
n=4;
```

```
y1=u1;
y2=u2;
y3=u3;
y4=u4;
y5=u5;
```

```
switch x(1)
case 'a'
```

```
    z=str2double(x(2:n));
    y1=(pi*real(z))/180;
```

```
case 'b'
```

```
    z=str2double(x(2:n));
    y2=(pi*real(z))/180;
```

```
case 'c'
```

```
    z=str2double(x(2:n));
    y3=(pi*real(z))/180;
```

```
case 'd'
```

```
    z=str2double(x(2:n));
    y4=(pi*real(z))/180;
```

```
case 'e'
```

```
    z=str2double(x(2:n));
```

```

y5=(pi*real(z))/180;
otherwise

end
end

```

Anexo 2. Protocolos de comunicación usados.

RFCOMM

Puerto de serie inalámbrico o protocolo de reemplazo de cable que proporciona una emulación para nuestro caso a un puerto serie entre el celular y el pc el cual está conectado al dispositivo bluetooth. Como existen más dispositivos alrededor, se establece la conexión mediante un identificador UUID.

Los protocolos de la pila Bluetooth se pueden dividir en 2 categorías diferentes: protocolos de base y protocolos adoptados. Se muestran en la figura1.

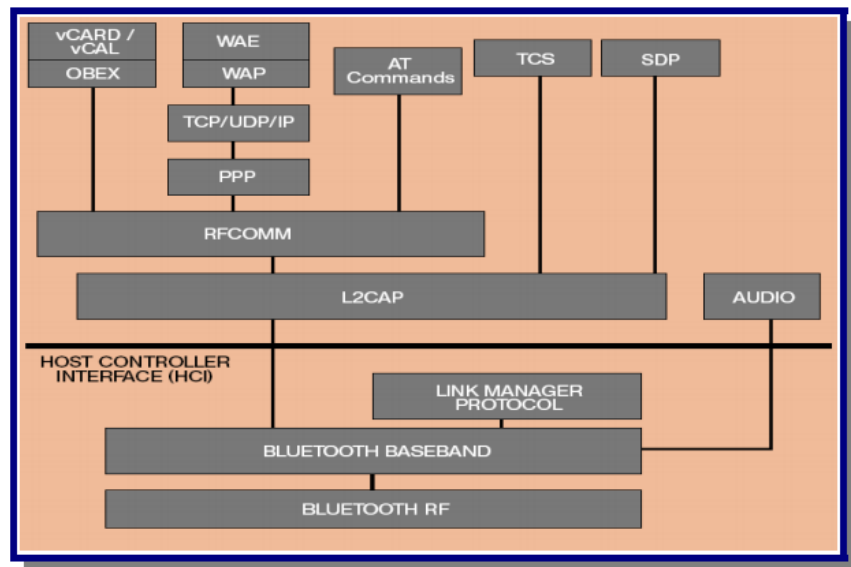


Figura 6. Pila de protocolos de la pila Bluetooth.

Este protocolo que emula las conexiones por puerto serie, dispone de 30 puertos (1 al 30) y solo puede gestionar 6 conexiones como máximo a la vez.

Es una equivalencia bluetooth se puede dar entre las redes cableadas normales de la siguiente forma:

- “Blueooth Radio” sería el cable en si mismo
- L2CAP sería ethernet
- RFCOMM sería TCP/IP

Por lo cual, a las aplicaciones que se comunican usando bluetooth, solo debe preocupar únicamente RFCOMM, igual que un navegador, le preocupa tcp/ip, y no si por debajo hay bluetooth, ethernet, o una red wireless 802.11.

COMUNICACIÓN SERIAL

El Instrument Control Toolbox de Matlab está configurado para trabajar en modo SSP (Puerto de Serie Síncrono). En este modo se utiliza un UUID específico. La secuencia utilizada fue la siguiente: 00001101-0000-1000-8000-00805F9B34FB la cual se mencionó anteriormente y se configuró en la aplicación Android, estableciendo una estructura cliente (celular) – servidor (pc).

El protocolo serial opera mediante tres condiciones digitales básicas: inicio de transmisión (IT), paridad (P) y fin de transmisión (FT).

Una de las configuraciones más usadas para un puerto serial es:

8 bits de datos

1 bit de parada

Sin bit de paridad

1 bit de inicio

Velocidad de 9600 BAUD

En la figura 2 se muestra el Bloque de configuración de comunicación serial usado en simulink.

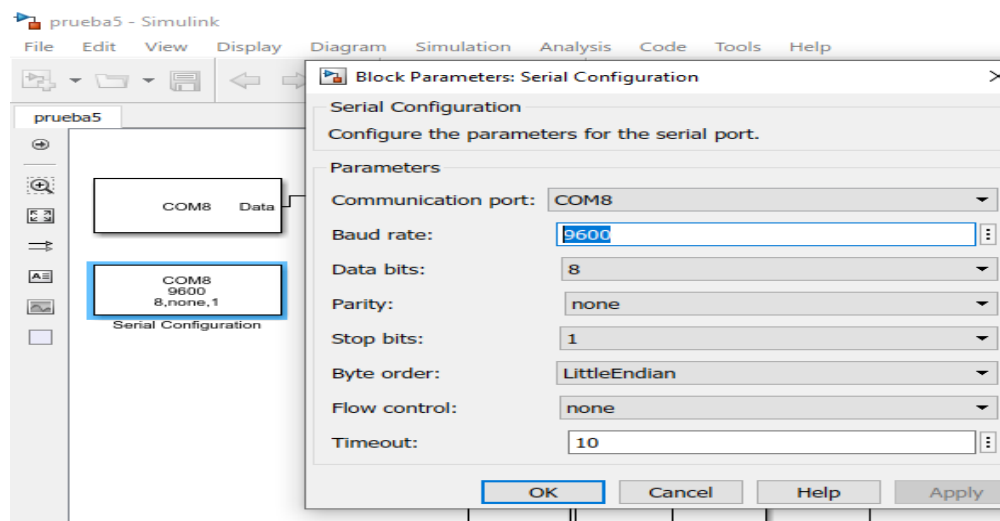


Figura 2. Bloque de configuración de comunicación serial en simulink.

- FORMATOS DE TRAMAS

La Figura 3, muestra un tipo de trama, esta consta de 11 bits que se transmiten cada uno a una velocidad de 9600 bits/s. Es decir, que a cada bit le toma $1/9600$ s transmitirse para cada flanco de subida del generador de BAUD. La tasa de transferencia de esta configuración serial de $9600/11 = 872.72$ Bits/segundo

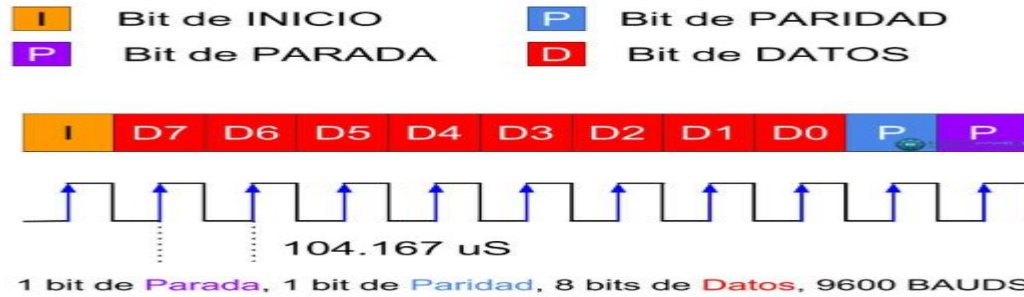


Figura 3. Trama de comunicación serial con la siguiente configuración: 1 bit de parada, 1 bit de inicio, 8 bits de datos a 9600 BAUDs.

La velocidad de transferencia es de 9600 bits/s, el retardo de cada bit enviado es de 104.167 microsegundos. Cada trama envía 10 bits (sin bit de paridad), nuestro vector de 4 bits con la información para mover una articulación a determinado ángulo es de $4 \times 10 \times 104.167$ microsegundos resultando en un 4.1678 ms.

Anexo 3. Mediciones en tiempos de retraso.

Se toman cinco mediciones de tiempo mediante un cronómetro de la respuesta entre la orden de movimiento desde el Smartphone hasta la respuesta en la virtualización. Se hacen mediciones con y sin el cálculo de la cinemática, el cual constituye el proceso que más inserta retraso.

Se utiliza botón de enviar con una orden de rotación de 180° . En este tipo de mando, se envía una sola orden con una sola posición. De manera que la virtualización efectúa un solo efecto. Se muestra la tabla 1 con sus respectivos tiempos.

Tabla 4. Mediciones tomadas para un ángulo de 180°

No. De medida.	Sin cálculo de cinemática	Con cálculo de cinemática
1	0.58	1.35
2	0.46	1.29
3	0.39	1.58
4	0.44	1.05
5	0.45	1.15
promedio	0.46	1.08

En este caso se nota que la respuesta bajó un 57.4% aproximadamente al aplicar el bloque del cálculo de la cinemática.

Se utiliza seekbar hasta su máximo: 360°. En este tipo de mando, se va desplazando por la pantalla a través de la barra hasta su máximo, de manera que se van enviando diferentes órdenes en la medida en que se va arrastrando la barra. El tiempo entonces se toma cuando la virtualización responde a la última orden, es decir, da un giro de 360° volviendo al final a la posición original. Se muestra la tabla 2 con sus respectivos tiempos.

Tabla 5. Mediciones tomadas para un ángulo de 360°

No. De medida.	Sin cálculo de cinemática	Con cálculo de cinemática
1	0.69	4.27
2	0.7	3.79
3	0.79	3.49
4	0.8	3.82
5	0.86	4.3
promedio	0.76	3.93

Anexo 4. Variables en unidades de ingeniería.

Las variables de entrada corresponden a **posición en grados (ángulos)**, los cuales se manipulan desde el dispositivo móvil y con lo cual se obtiene una salida de la **posición** del robot virtual visualizada de forma animada y también en coordenadas cartesianas en **metros (distancia)**.

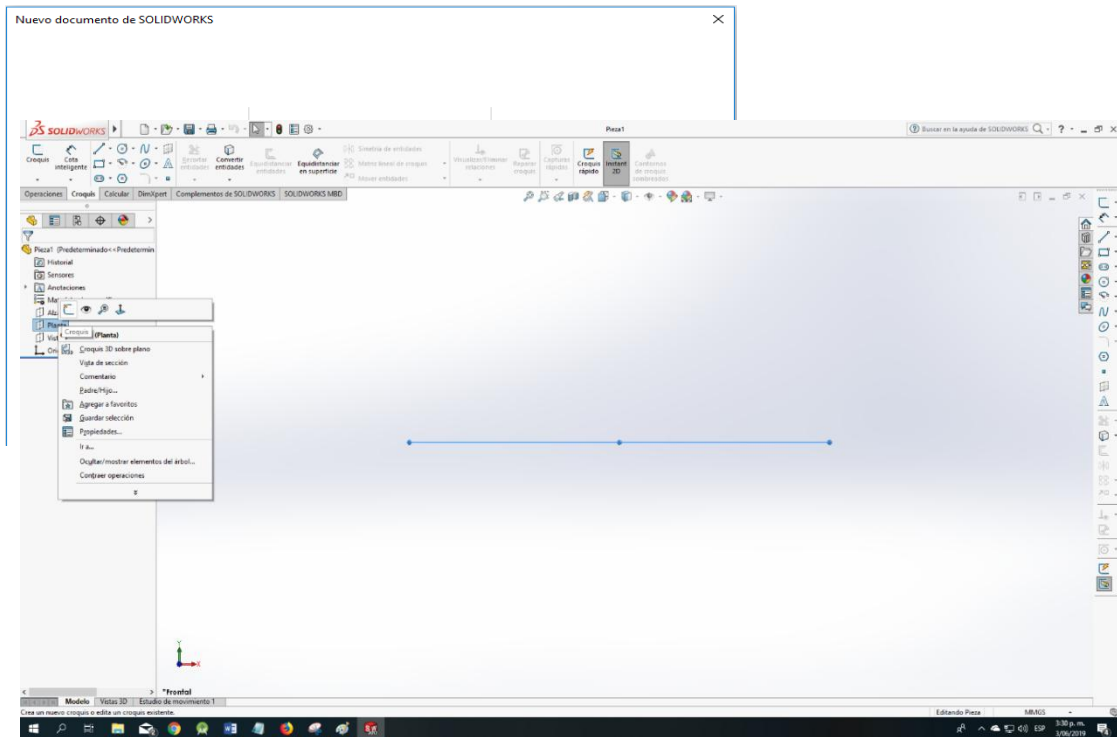
Anexo 5. Desarrollo en Solid-works.

TRABAJO EN PIEZAS

Se inicia el programa Solidworks.

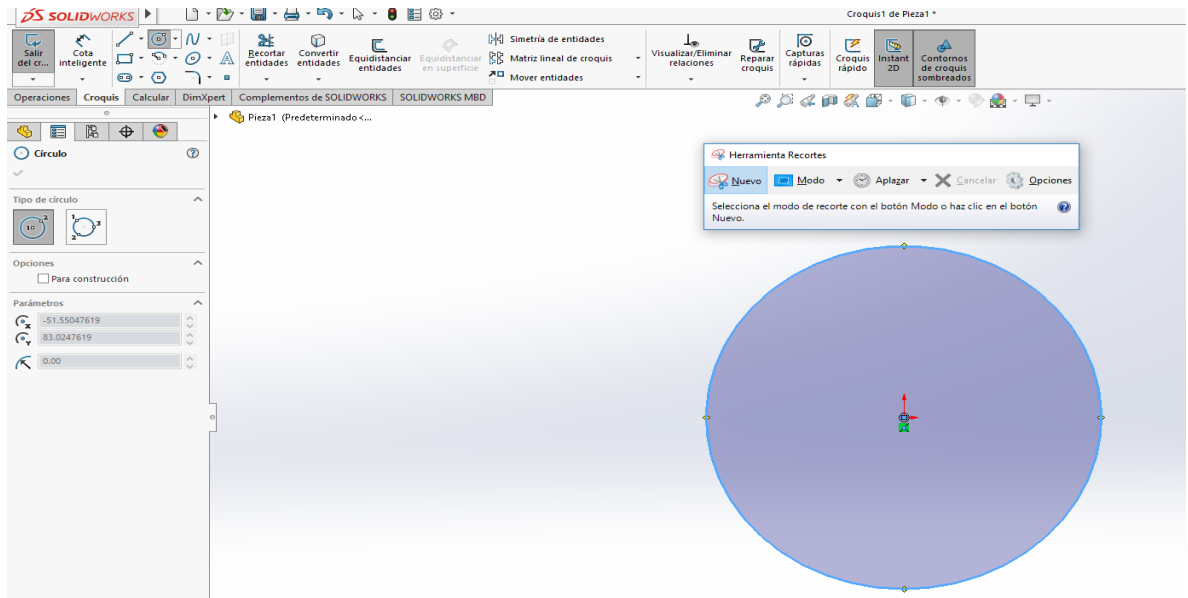
Cada vez que se trabaja con un archivo nuevo, el programa pregunta si es una pieza, un ensamblaje o un dibujo:

Se selecciona la opción de pieza.

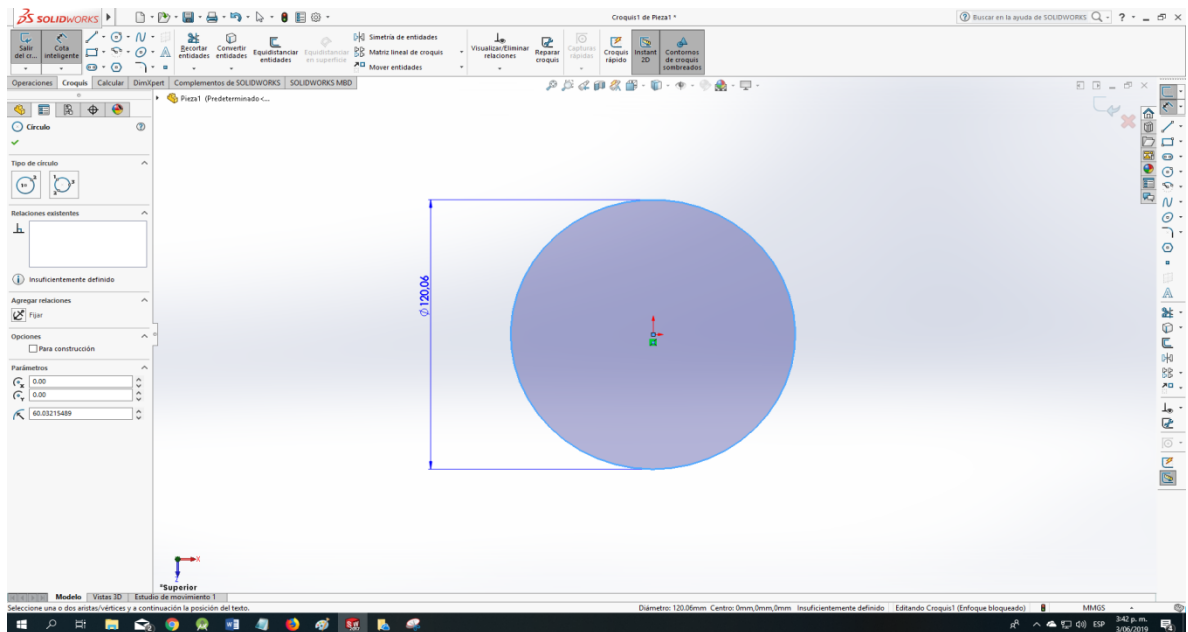


En el nuevo documento, seleccionamos planta y luego a través del clic derecho se selecciona croquis 3d en plano.

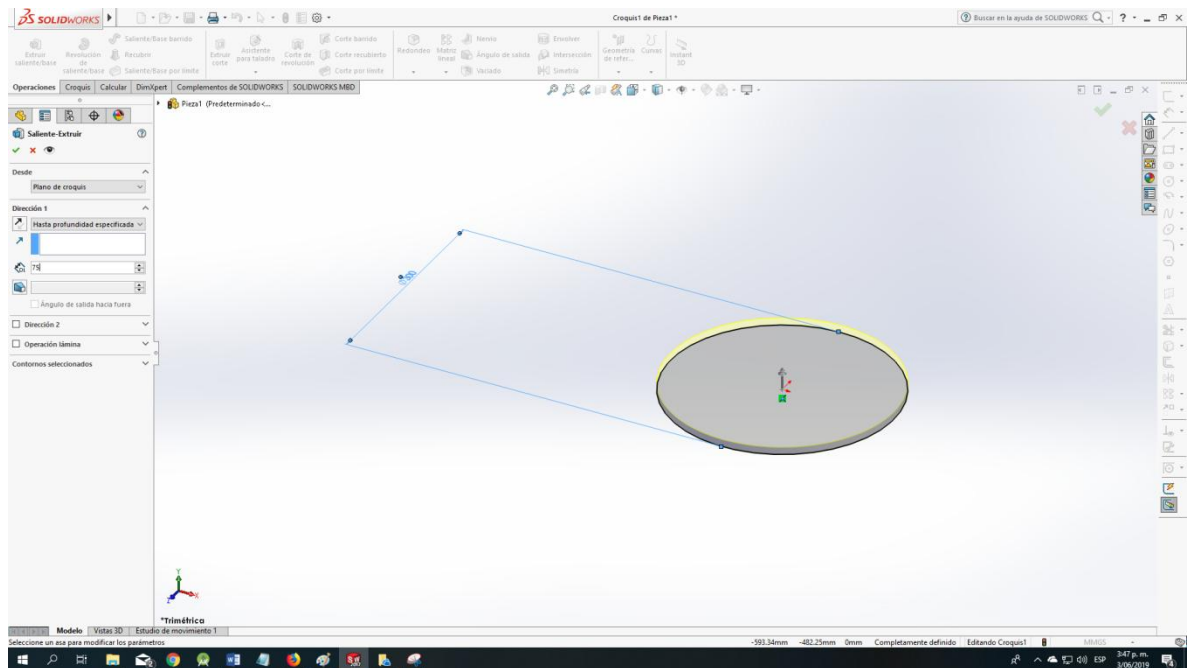
En la pestaña de croquis se selecciona el botón de círculo. Se da clic en el centro del croquis, se amplía el círculo y luego se da clic para fijar su tamaño.



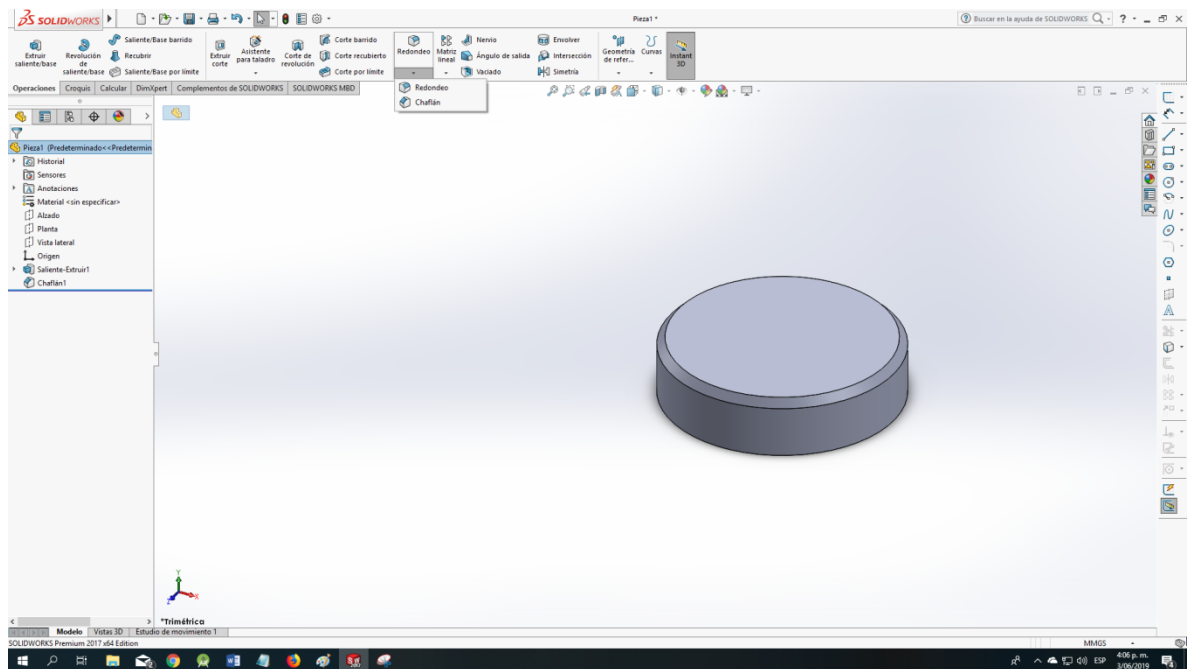
Se utiliza el botón de cota inteligente para establecer la dimensión del diámetro del círculo de manera exacta.



En la pestaña “Operaciones” se da clic en el botón “extruir base/Saliente”, insertando en el recuadro izquierdo una altura correspondiente.

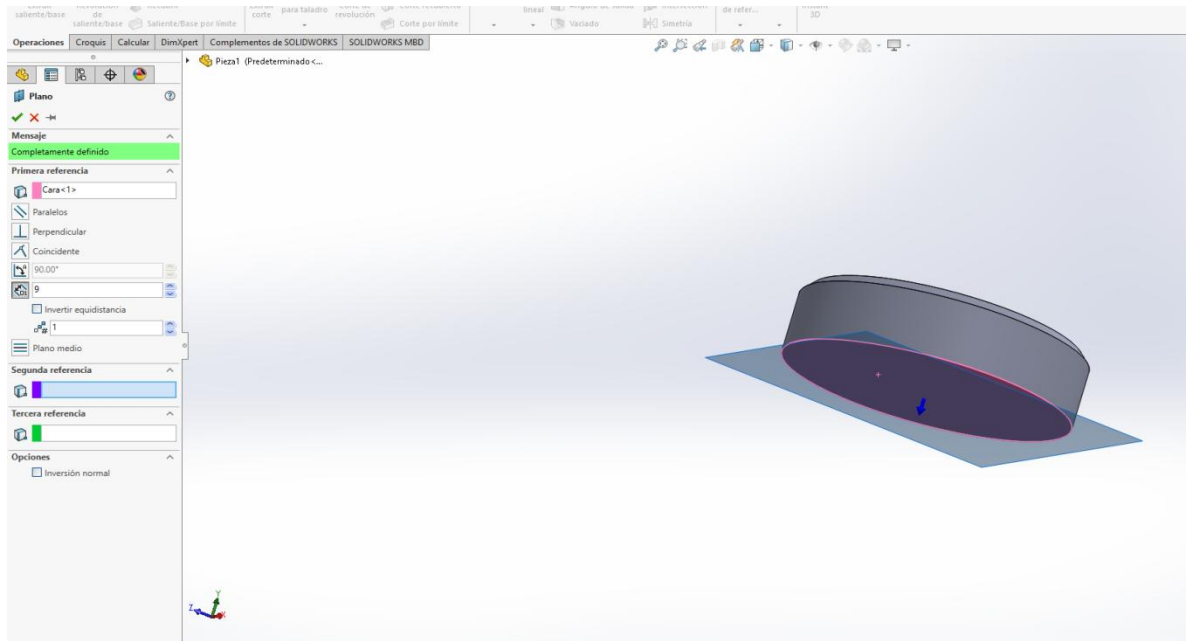


Se da clic en el botón “chaflán” de la pestaña “Operaciones”, y se inserta una medida específica.

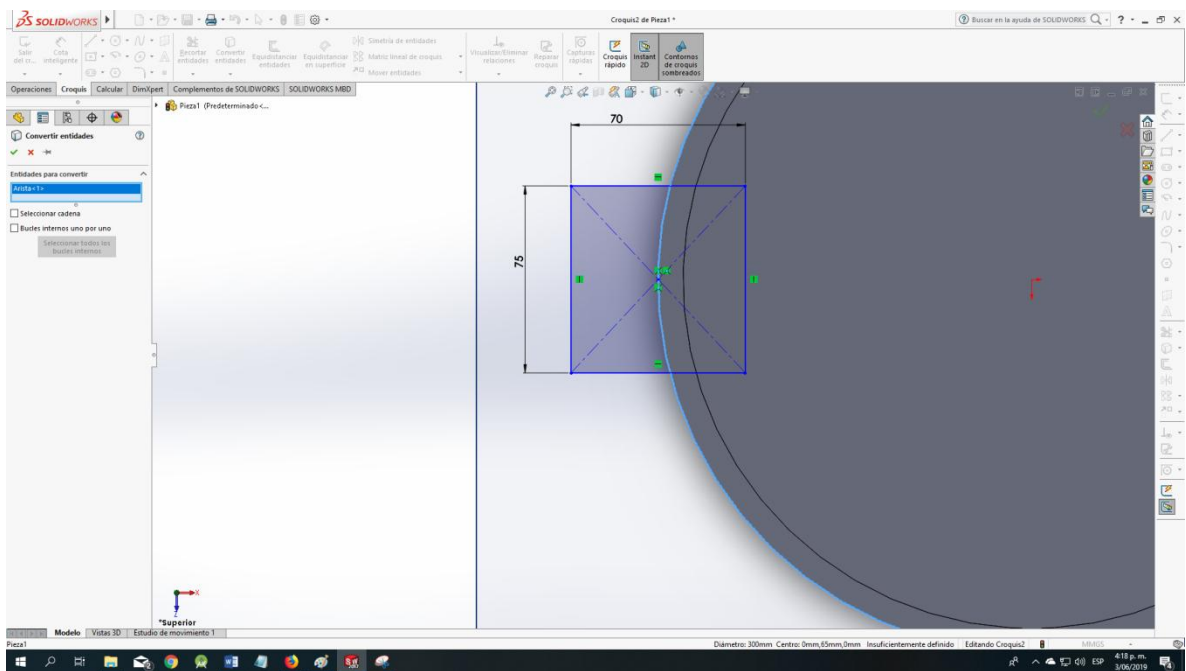


Se selecciona la opción “Plano de referencia” ubicado en el botón “Geometría de referencia” en la pestaña de “Operaciones”. Y se escoge el plano inferior, se

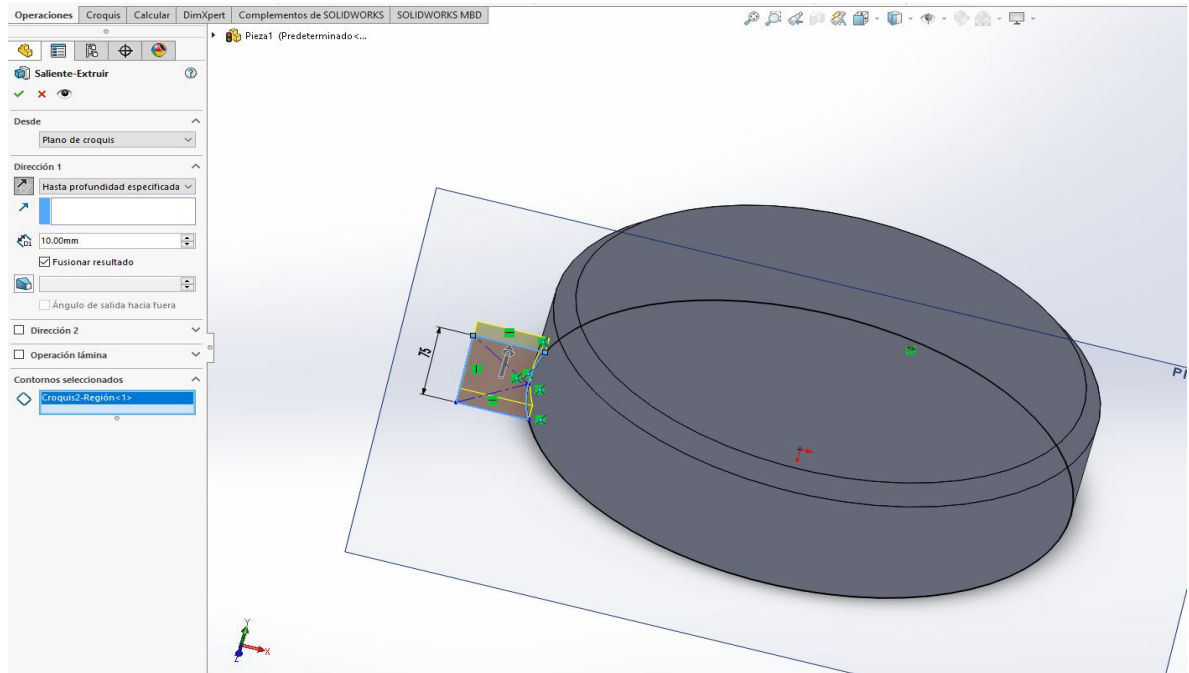
configura una distancia, y a través de clic derecho se convierte en croquis, esto para trabajar en las bases.



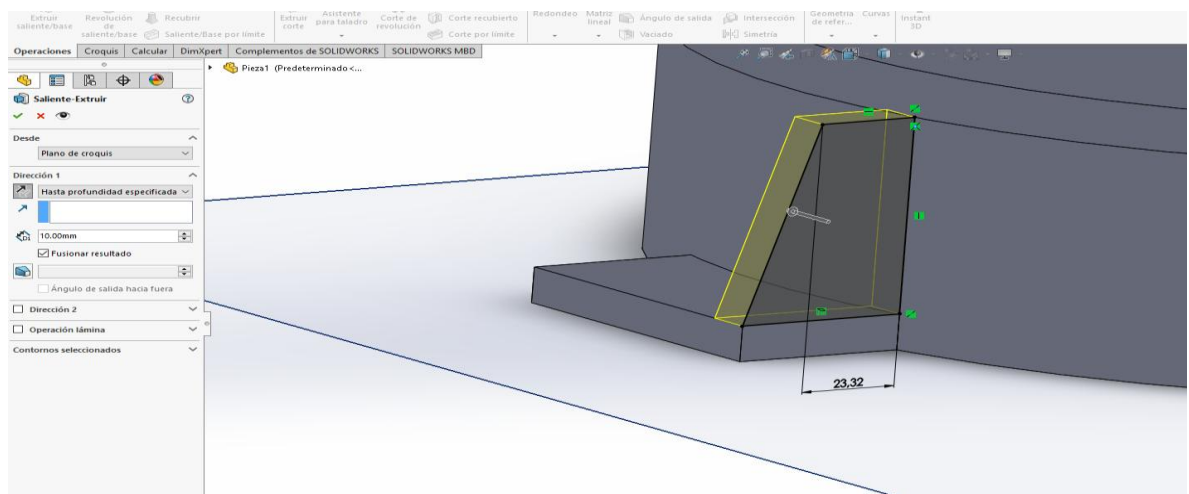
Se selecciona la opción “Rectángulo de centro” de la pestaña “Croquis”, ubicamos un rectángulo en la parte lateral de la figura. Se utiliza la herramienta “convertir unidades” y se selecciona la línea curva que intersecta el rectángulo, para luego, mediante la herramienta “recortar unidades”, darle la forma final.



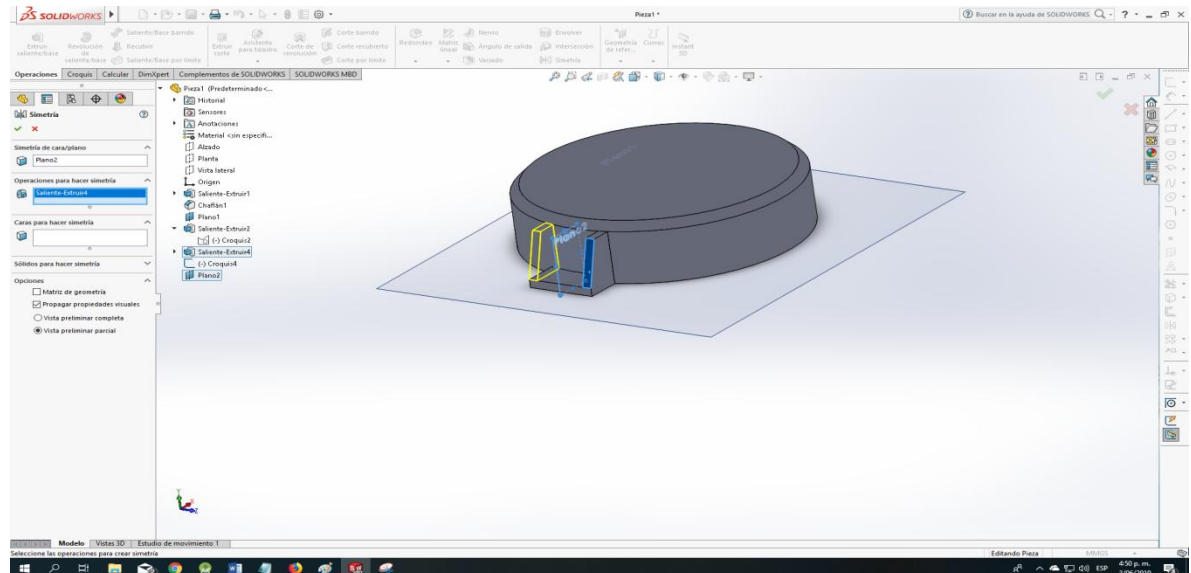
En el botón “Visualizar relaciones” seleccionamos “Definir croquis completamente” y posteriormente, en la pestaña “Operaciones” se utiliza de nuevo la herramienta “extruir base/Saliente”.



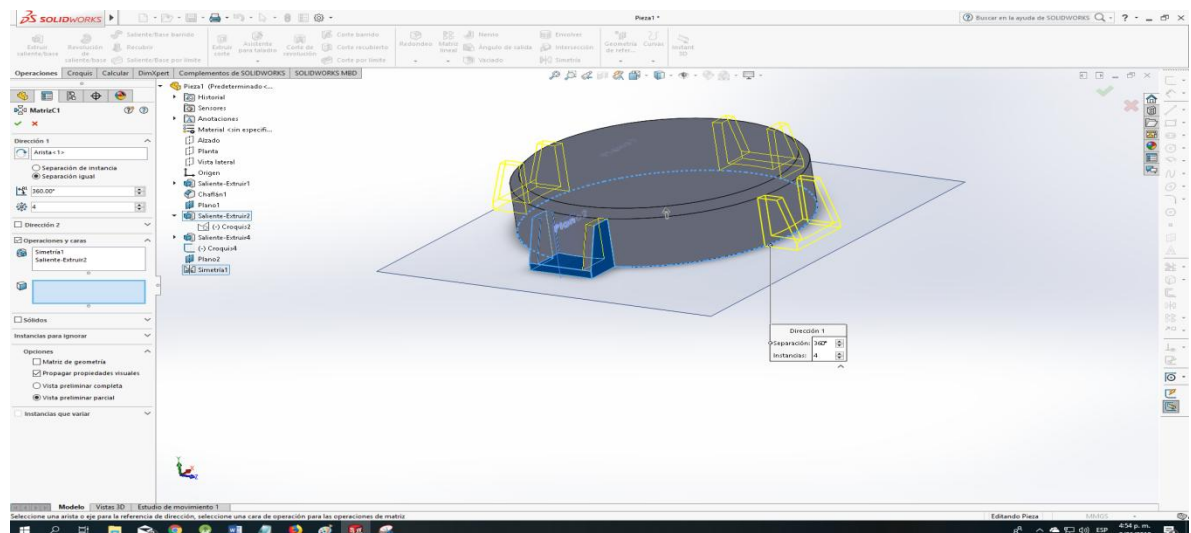
Se escoge una de las paredes del rectángulo como croquis, a través del clic derecho. Se selecciona el botón “línea” de la pestaña “croquis”, y se crea un perímetro con la siguiente figura, dando las medidas exactas mediante la herramienta de cota inteligente. Se establece como croquis en la opción “Visualizar relaciones”. Y se extruye con la herramienta mencionada anteriormente.



Se selecciona la opción “plano” del botón “Geometría de referencia” de la pestaña “Operaciones”. Se escogen los dos planos laterales del rectángulo para establecer un plano de referencia en la mitad. Posteriormente, se utiliza el botón de “simetría” de la pestaña “Operaciones”. Y se selecciona los elementos que participarán en la simetría ubicados en la estructura de árbol a la izquierda.



En la pestaña “Operaciones” se selecciona la opción “Matriz circular” del botón “Matriz lineal”. Se escoge la circunferencia inferior de la base. Se escogen los elementos que participaran de la matriz en la estructura de árbol ubicada a la izquierda, o dando clic en la figura. Se establece el número de veces que se repetirá y se repartirá la estructura.



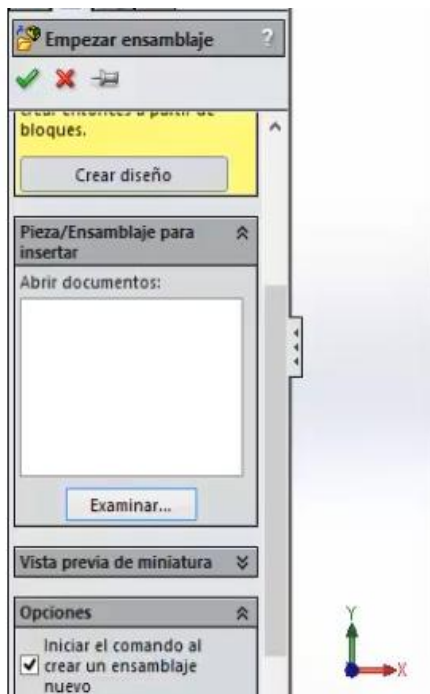
De esta manera, y mediante las herramientas utilizadas creamos un ejemplo de estructura básica para el robot.

ENSAMBLAJE

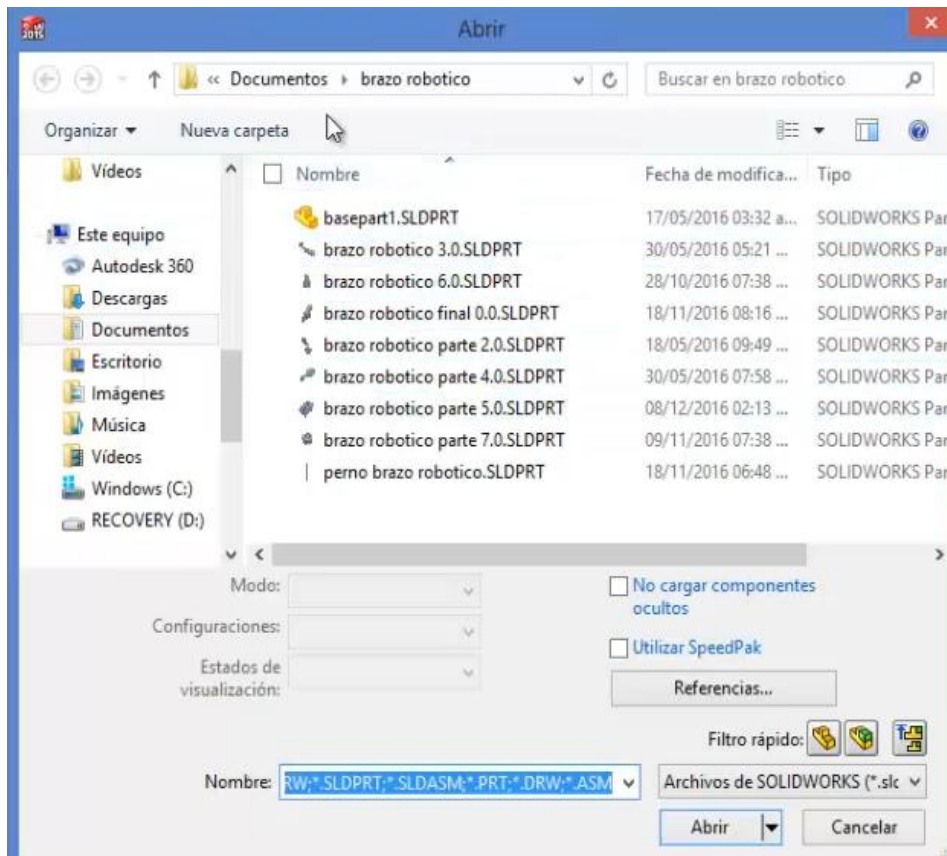
Se inicia un nuevo documento.

Se selecciona la opción de ensamblaje, debido a que se integrarán todas las piezas creadas anteriormente.

En el recuadro de la izquierda llamado “pieza/ensamblaje para insertar” se



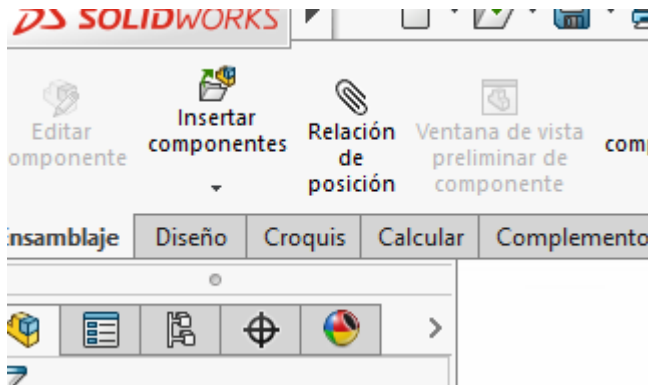
presiona “examinar” y se busca la primera pieza a ingresar en el ensamblaje.



Para ingresar las otras piezas se utiliza el botón “insertar componentes” de la barra de herramientas.



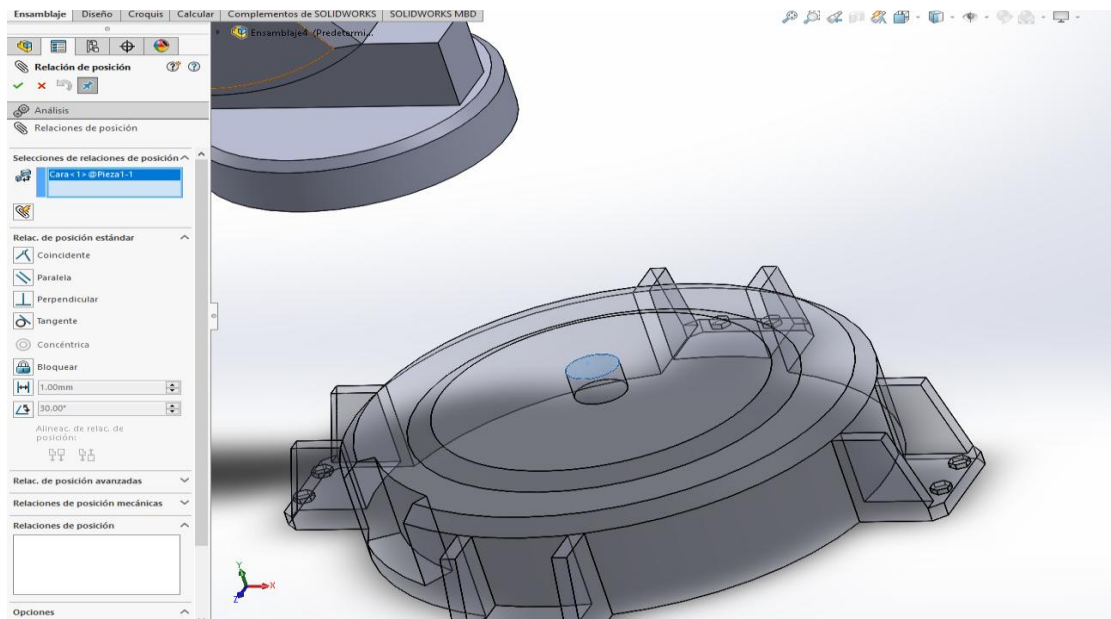
Cada que se inserta un componente, aparece asociado al puntero del mouse, se puede posicionar en el espacio de trabajo con un clic. Al principio, los componentes se posicionan de manera aleatoria, un tanto separados uno de otro, para luego unirlos. Sin embargo, al insertar la base del brazo robot se da clic en el botón aceptar para que su posición sea el origen de las coordenadas, esto será importante posteriormente a la hora de exportar el sistema de coordenadas y encontrar los ejes de rotación de las articulaciones.



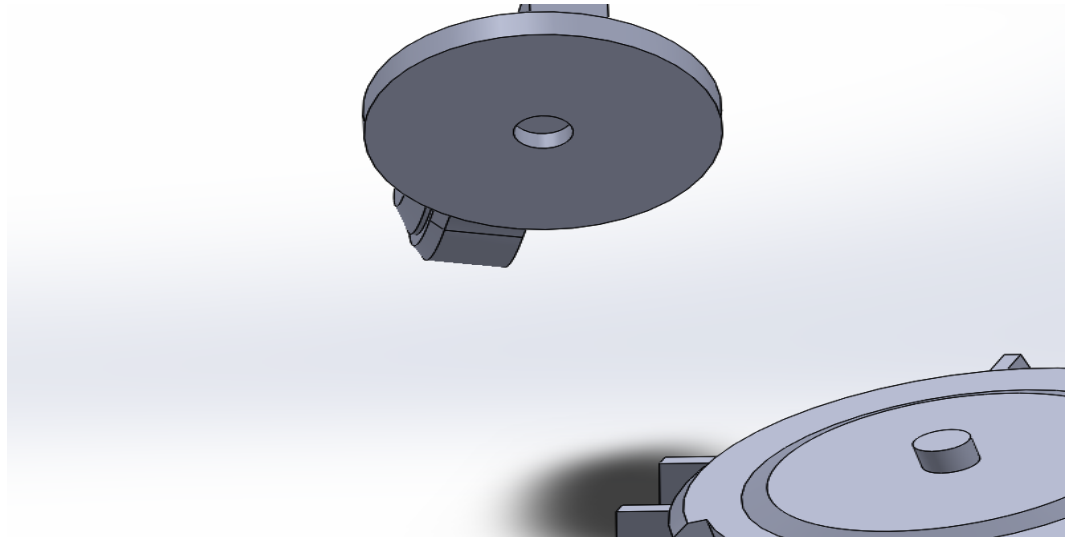
Se utiliza el botón “Relación de posición” en la barra de herramientas.

K

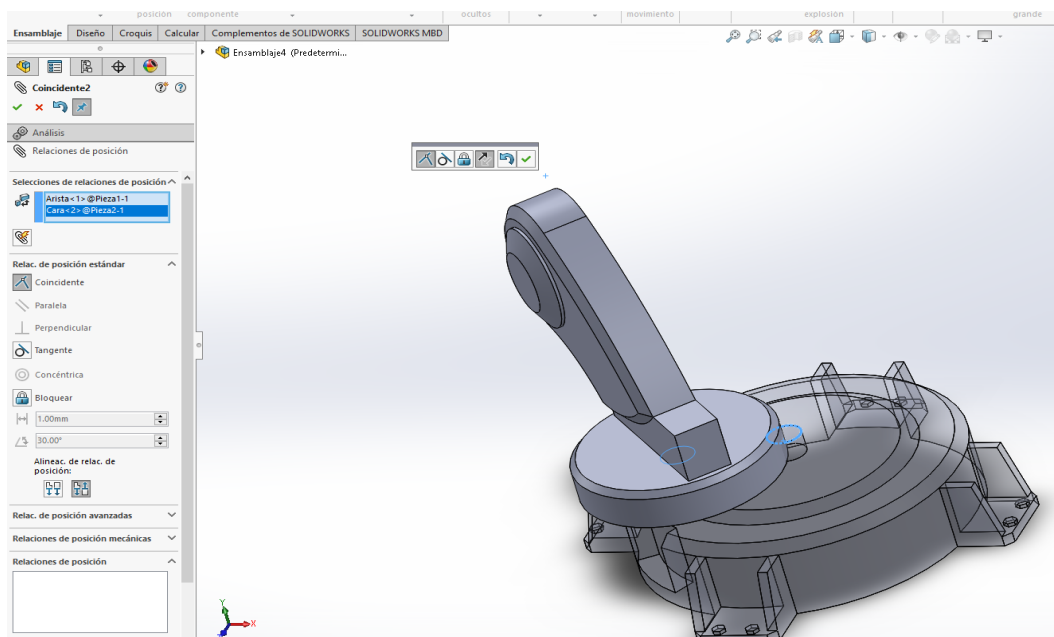
Se selecciona la superficie de ensamble del primer componente que se va a unir.



Se selecciona el cilindro de ensamble del segundo componente que se va a unir y se da clic en la opción de coincidentes en el recuadro que aparece.

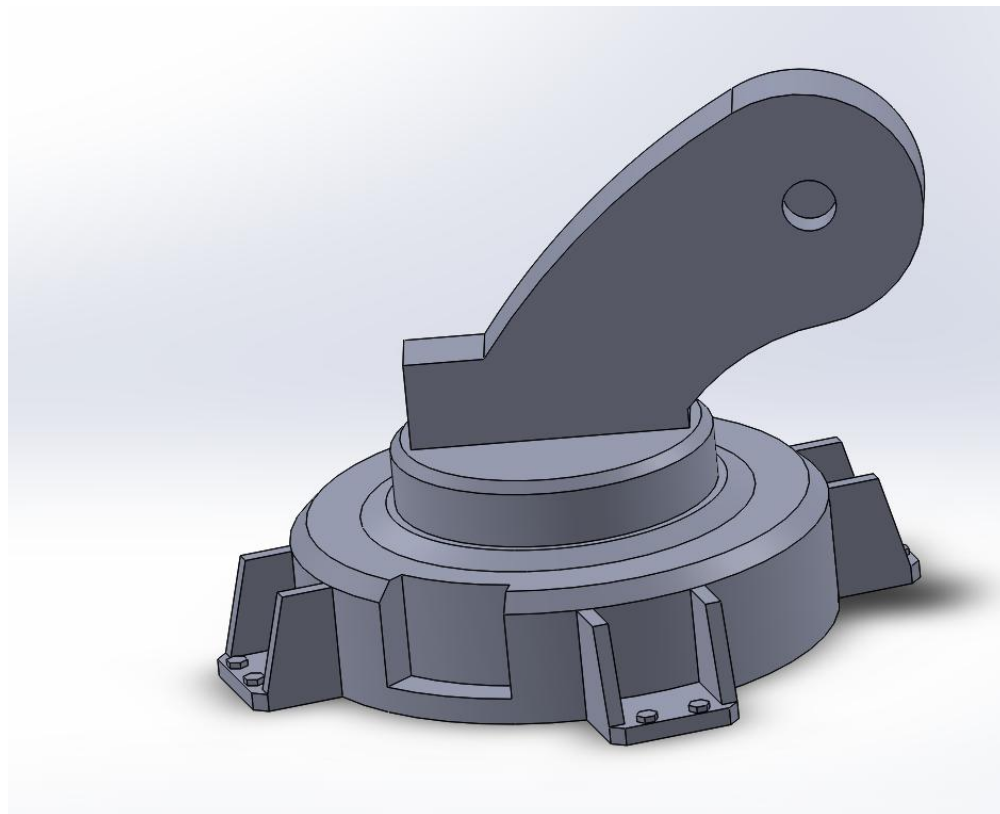
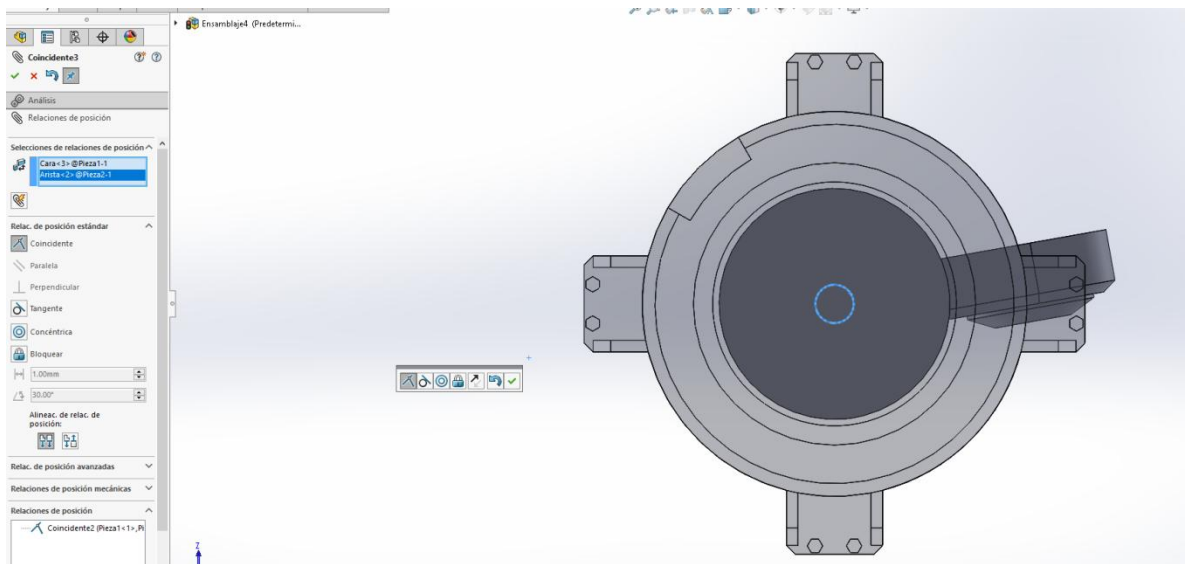


Se aceptan como coincidentes.



Se selecciona la superficie cilíndrica del primer componente y la superficie cilíndrica del segundo componente.

Al aceptar la nueva relación de posición, se completa el primer ensamble:

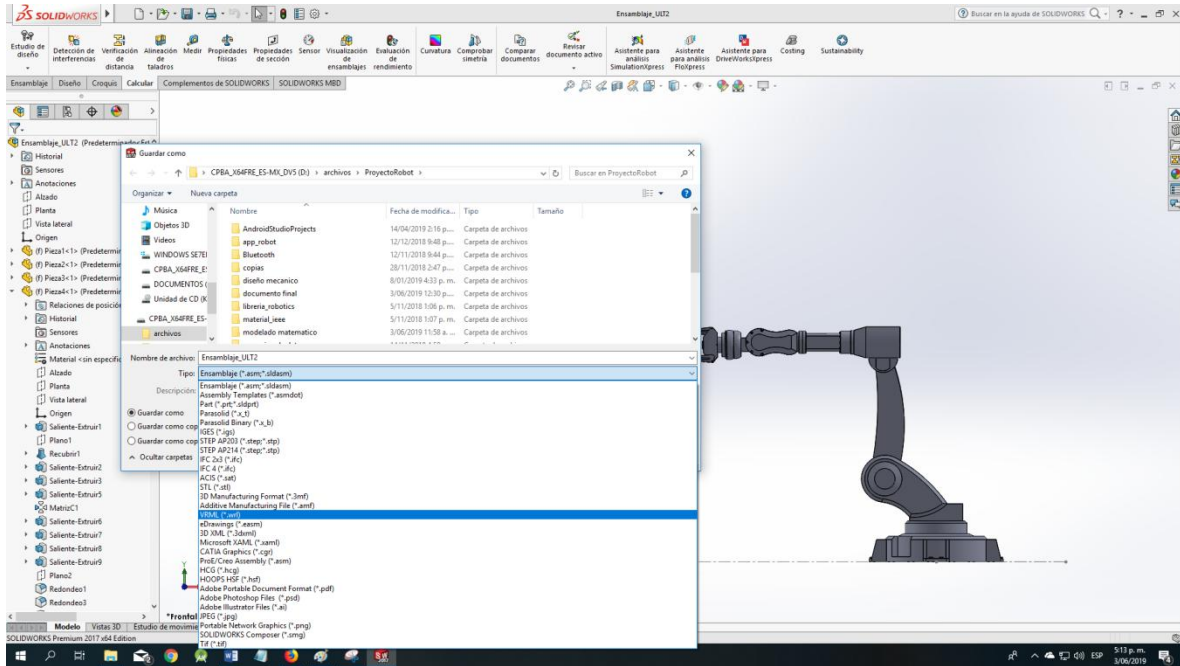


Este mismo procedimiento se repite en todos los componentes.

Anexo 6. Exportación del brazo virtual desde solid-works hacia Matlab.

Se abre el documento de ensamblaje en el programa Solidworks.

Se da clic en el menú “archivo” y luego en la opción “guardar como”. En la parte inferior del recuadro que aparece, se da clic en la flecha de la opción “tipo” y se selecciona en VRML(.wrl) como aparece en la siguiente imagen.



Se selecciona la ruta donde quedará guardado el archivo y listo. Ya ha sido exportado. Ahora dicho archivo puede abrirse con cualquier editor de lenguaje VRML.

Anexo 7. Parámetros cinemáticos – Variables de Ingeniería- Validación cinemática y/o dinámica.

Las variables de entrada corresponden a **posición en grados (ángulos)**, los cuales se manipulan desde el dispositivo móvil y con lo cual se obtiene una salida de la **posición** del robot virtual visualizada de forma animada y también en coordenadas cartesianas en **metros (distancia)**.

Los parámetros de Denavit-Hartenberg usados para el cálculo de la cinemática directa. Los parámetros se extrajeron a través de la herramienta “medir” de solidworks.

Command Window

```
noname:: 5 axis, RRRRR, stdDH, slowRNE
```

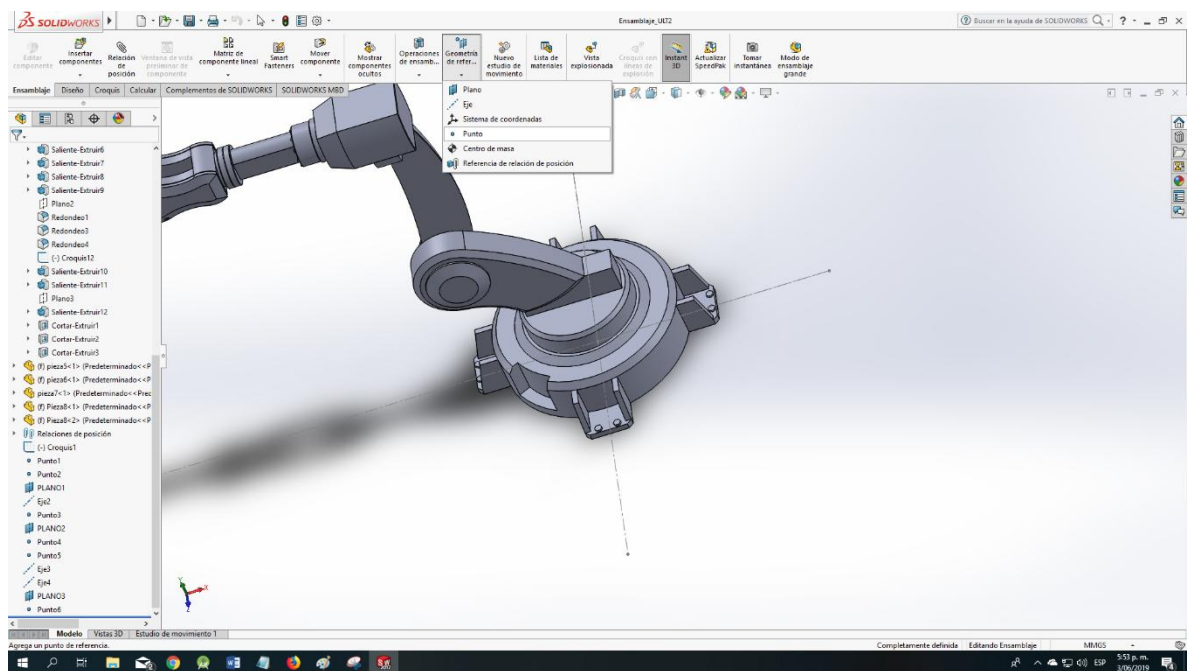
j	theta	d	a	alpha	offset
1	q1	0.16	0.16	1.5708	0
2	q2	0	0.35	0	1.5708
3	q3	0	0	1.5708	0
4	q4	0.25	0.05	1.5708	3.14159
5	q5	0.01	0.12	0	1.5708

fx >>

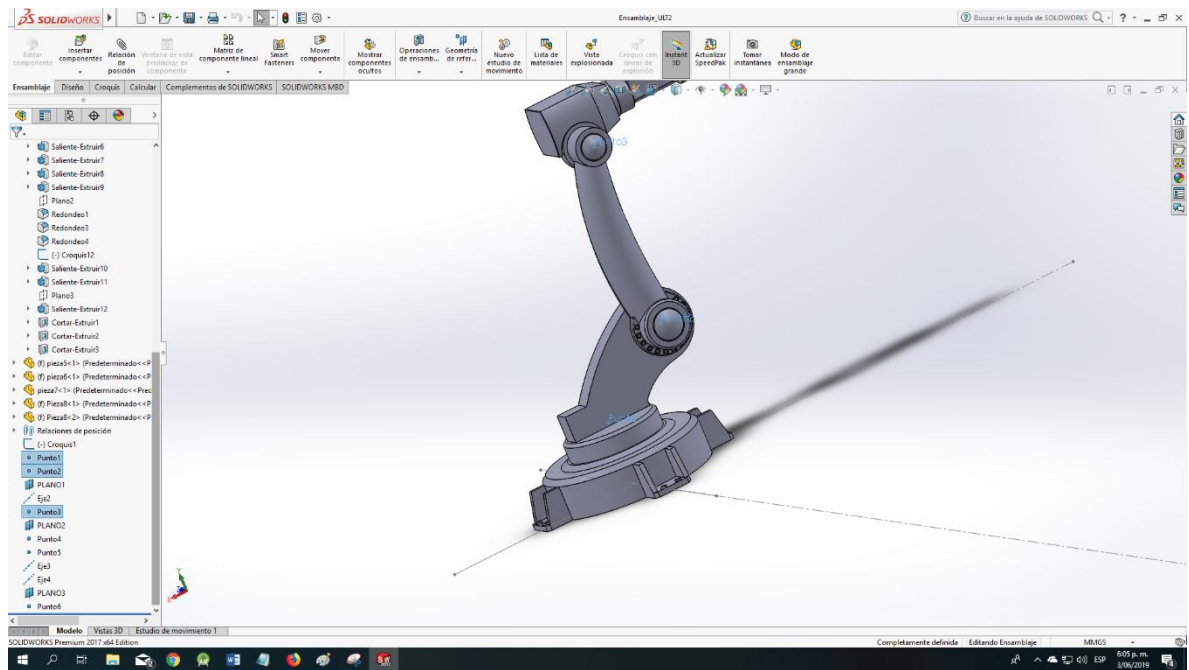
Como no se está modelando la dinámica del robot se excluyen parámetros de masa, momentos, fuerzas, velocidades del motor.

Ya que se tiene un modelo virtual, los parámetros deberían salir de dicho modelo. Por tanto, se utiliza el software Solidworks para esta necesidad.

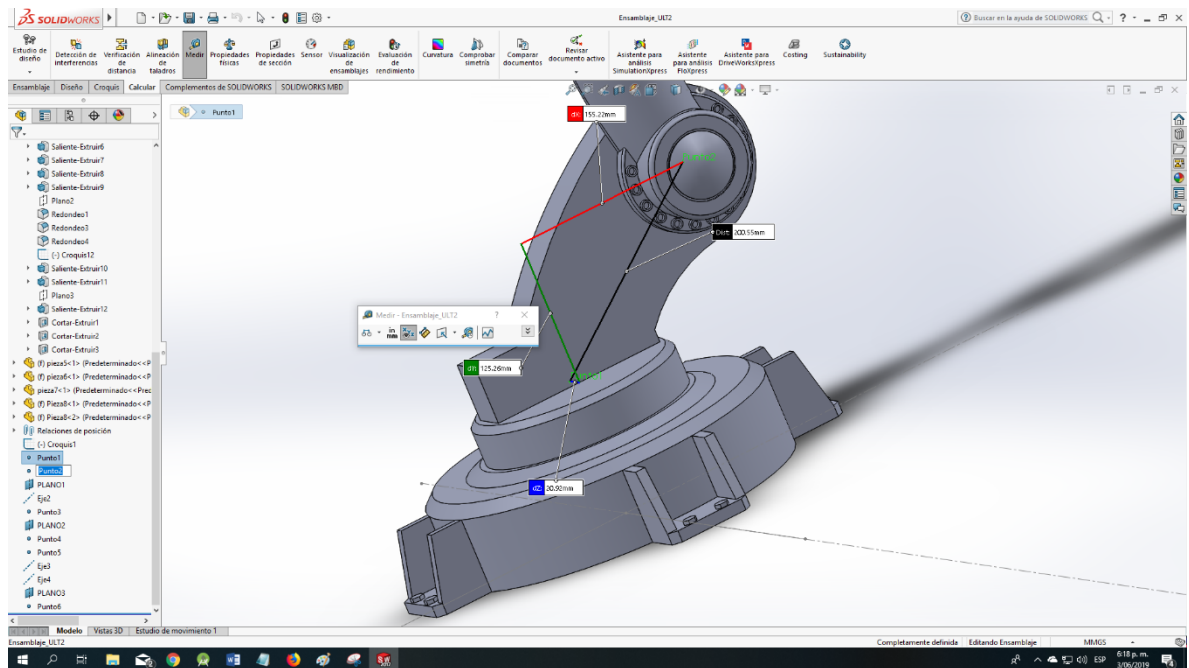
Se utiliza la opción “punto” del botón “Geometría de referencia” en la pestaña “ensamblaje” como aparece en la siguiente figura.



Y se colocan puntos en todos los ejes a través de los cuales se efectúa la rotación de cada componente del brazo. Como se puede ver en la figura con las cruces en azul.



Al dar clic en el botón “medir” de la pestaña “Calcular” se accede a una herramienta con la cual se seleccionan los puntos de ejes de un par de articulaciones y se establecen las distancias en coordenadas cartesianas como se puede apreciar en la siguiente figura.



Los parámetros Denavit-Hartenberg son mediciones entre distancias y ángulos entre las articulaciones, de manera que entre la articulación los puntos 1 y 2 podremos establecer las siguientes distancias que aparecen consignadas en la figura, y de ahí ya tendremos nuestros primeros parámetros.

Cada sistema de coordenadas se establece sobre las siguientes reglas.

θ : Es el ángulo de la articulación desde el eje x del punto 1 hasta el eje x del punto 2, medido respecto del eje z del punto 1, usando la regla de la mano derecha.

d : Es la distancia medida desde el origen, a lo largo del eje z del punto 1 hasta la intersección de este eje con el eje x del punto 2.

a : Es la distancia de separación entre los orígenes de los sistemas de referencia del punto 1 y el punto 2, medida a lo largo del eje x del punto 2, hasta la intersección con el eje z del punto 1. (o la distancia más corta entre los ejes z del punto 1 y z del punto 2, cuando estos no se interceptan)

α : Es el ángulo que separa los ejes z del punto 2 y z del punto 1, medido respecto del eje x del punto 2.

La validación en este caso se realizó de manera visual.

Recomendación:

Con respecto a los parámetros dinámicos, queda como trabajo futuro.