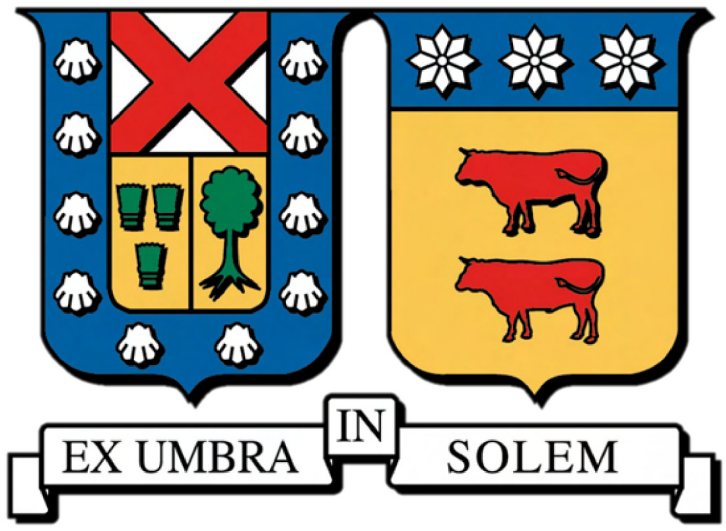
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INGENIERÍA CIVIL MECÁNICA



IMPLEMENTACIÓN DE UN SISTEMA DE MANUFACTURA ADITIVA DE ALTO VOLUMEN Y VELOCIDAD UTILIZANDO UN COBOT

ANGEL IVÁN VALDÉS AZÓCAR

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MECÁNICO

PROFESOR GUÍA: Dip. Ing. UDO RHEINSCHMIDT

PROFESOR CO-REFERENTE: Dr. Ing. LUIS PÉREZ POZO

JUNIO-2023

Resumen

En el contexto del presente trabajo se desarrolla un sistema de manufactura aditiva con el cual se busca crear piezas a partir de plástico reciclado fundido a gran volumen. Para esto se trabaja con un *extruder* de alto volumen para fundir el triturado de plástico, un brazo robótico (UR5) para guiar con las trayectorias que conforman al objeto 3D deseado y un brazo compensador, debido a que una problemática inicial es el peso del extrusor y la carga útil del cobot que son 5[kg], por ende el brazo compensador es crucial en el estudio y desarrollo, ya que soporta el peso total del dispositivo de extrusión de alto volumen, mientras que el UR5 sólo sirve de guía. Estos 3 componentes más una mesa caliente conforman la impresora 3D de alto volumen.

Las problemáticas en el estudio fueron en primera instancia aprender como funciona un brazo robótico a partir de cursos online que proporciona la misma empresa "Universal Robots". Luego de conocer su funcionamiento, se llega a la conclusión que es factible que pueda seguir trayectorias complejas y es aquí donde se explora el software "RoboDk" junto a los *slicers* "Ultimaker Cura" y "Slic3r". Estos se trabajaron durante el presente trabajo para encontrar y acomodar las configuraciones de manera digital para que calcen a las condiciones de trabajo real. Donde al tener un brazo compensador se tiene restricciones del movimiento del brazo robótico. Por lo cual se encuentra la solución de utilizar una bisagra que proporciona grados de libertad que disminuyen los esfuerzos en el robot, si bien se obtienen resultados positivos con esta solución se recomienda que sea temporal y en un futuro utilizar una unión más apropiada para el trabajo. Así entonces, se realizan pruebas para recrear un cilindro de 50 [mm] de diámetro, 3 [mm] de espesor y 15 [mm] de alto.

Los resultados al solucionar las problemáticas son favorables y se acercan a las medidas del cilindro que fue parte del estudio. Y se recomienda que la mesa este nivelada y paralela a las trayectorias horizontales del *extruder*, además de utilizar inyección de aire para aumentar la dureza del material fundido al ser depositado. Finalmente, se concluye que es factible crear una impresora 3D con el sistema total, es decir el cobot (UR5), extrusor de alto volumen y brazo compensador. Además, una recomendación de una nueva configuración del sistema para aprovechar al máximo los movimientos y disminuir los esfuerzos al cobot. Terminando con una guía de uso y mantenimiento del sistema.

Abstract

In the context of this work, an additive manufacturing system is developed, with which it is sought to create parts from recycled plastic, melted in large volume. For this, we work with a high-volume extruder to melt the plastic shredding, a robotic arm (UR5)to guide with the trajectories that make up the desired 3D object and a compensating arm, because an initial problem is the weight of the extruder and the payload of the cobot 5[kg], therefore the compensating arm is crucial in the study and development, since it supports the total weight of the high-volume extrusion device, while the UR5 only serves as a guide. These 3 components plus a hot table make up the high-volume 3D printer.

The problems in the study were in the first instance to learn how a robotic arm works from online courses provided by the same company "Universal Robots". After knowing its operation, it is concluded that it is feasible that it can follow complex trajectories, and it is here where the "RoboDk" software is explored alongside the "Ultimaker Cura" and "Slic3r" slicers. These were worked during the present work to find and accommodate the configurations in a digital way to adapt to the real working conditions. By having a compensating arm you have restrictions on the movement of the robotic arm. Which is why the solution is found to use a hinge that provides degrees of freedom that decrease the efforts in the robot, although positive results are obtained with this solution, it is recommended that it be temporary and in the future use a more appropriate union for the work. Thus, tests are carried out to recreate a cylinder 50 [mm] in diameter, 3 [mm] thick and 15 [mm] high.

The results when solving the problems are favorable and are close to the measurements of the cylinder that was part of the study. It is also recommended that the table is level and parallel to the horizontal trajectories of the extruder, in addition to using air injection to increase the hardness of the molten material when deposited. Finally, it is concluded that it is feasible to create a 3D printer with the total system, that is, the cobot (UR5), high volume extruder and compensating arm. In addition, a recommendation of a new configuration of the system to make the most of the movements and decrease the efforts to the cobot. Ending with a guide to use and maintenance of the system.

Agradecimientos

Siendo está la última página en escribir, lo cual cierra un ciclo importante en mi vida y que me forjó a ser lo que ahora soy. Y es por esto que es necesario dar agradecimientos a todas las personas que hicieron que este camino fuera más llevadero y ser parte importante en cada suceso a lo largo de ésta carrera.

Primero dar las gracias a mi familia que fue, es y será parte importante en mi desarrollo como persona y prontamente como profesional. A mi mamá por nunca dudar de mis capacidades y siempre dar su apoyo a lograr lo que quisese, de siempre ver el lado positivo de las cosas y alentándome a seguir. A mi papá por recordarme siempre que podía llegar lejos y que ahora lo estoy logrando, de enseñarme que a veces con poco se puede hacer más y marcar la diferencia a pesar de lo duro que puede llegar a ser la vida. A mi hermano por siempre estar a mi lado, ayudándome cuando necesitaba ayuda, por darme consejos y alentándome a lograr a lo que ahora voy a ser, un profesional en mi carrera, por siempre estar preocupado de si necesitaba algo y gracias a su ayuda pude tener una vida más llevadera en la etapa universitaria. Es por esto y mucho más que quiero dar las gracias inconmensurables a mi familia por siempre estar presente para mi y confiar a que lo podía lograr, de ser el primer profesional de la familia. Muchas gracias.

También dar las gracias a mi novia que es pilar importante en este proceso a ser profesional, por siempre estar ahí y buscar soluciones cuando no las veía. De alentarme a que era capaz de lograr resolver los desafíos que se me presentase y de siempre apoyarme. Muchas gracias.

Dar las gracias a mis amigos, que fueron y son parte importante en la vida universitaria, gracias a ellos el pasar por la carrera no fue tan duro. Las risas, las preocupaciones y el estudio siempre se van a recodar como anécdotas de la amistad que empezó en la universidad. Les deseo lo mejor y que la amistad que nació en la universidad persista en el mundo laboral. También a los amigos y personas que me encontré durante este camino. Muchas gracias.

Las gracias al equipo de trabajo para conformar el laboratorio ECO-FAB-LAB. Al profesor Udo Rheinschmidt por tener la genial idea de un laboratorio que recicle plástico con un fin educativo, por siempre estar positivo a los problemas que se presentaron durante el trabajo y buscar soluciones para lograr resultados. A los compañeros, que fueron parte del inicio de ésta idea y que ahora somos parte de la base a futuros trabajos para conformar el laboratorio. Muchas gracias.

Tabla de Contenidos

1 Introducción **1**

2 Objetivos y Alcances **2**

3 Marco Teórico **3**

3.1 Manufactura Aditiva . . . 3
3.2 CAD . . . 5
3.2.1 Autodesk Inventor . . . 5
3.3 Slicer . . . 5
3.3.1 Definición . . . 5
3.3.2 Slicer para Impresión 3D . . . 5
3.3.3 Tipos de slicer para impresión 3D . . . 6
3.3.4 G-code . . . 6
3.4 Cobots . . . 7
3.4.1 Tipos de Cobots . . . 7
3.5 Manufactura Aditiva con cobots . . . 8
3.6 RoboDk . . . 9
3.6.1 Ventajas y desventajas de la programación fuera de línea o simulación . . . 10

4 Materiales y Métodos **11**

4.1 Universal Robots 5 (UR5) . . . 11
4.1.1 Seguridad . . . 13
4.1.2 Espacio de trabajo de UR5 . . . 16
4.1.3 Montaje . . . 16
4.1.4 Herramienta . . . 17
4.1.5 Normas . . . 18
4.1.6 Iniciar Cobot . . . 20
4.1.7 PCH o TCP . . . 23

4.1.8 Mesa de Trabajo 27
4.1.9 Creación de programa básico en UR5 30
4.1.10 Guardado 36
4.1.11 Conectar PC con UR5 38
4.2 RoboDK 39
4.2.1 Conexión de Robodk a UR5 39
4.2.2 TCP o PCH en Robodk 43
4.2.3 Dibujo en RoboDK 45
4.2.3.1 Inkspace 45
4.2.3.2 DXF to G-Code Converter 48
4.2.3.3 Creación de programa en robodk para ser simulado y listo para ser enviado al Cobot 49
4.3 Impresión 3D 56
4.3.1 Configuración actual Cobot-Extruder 56
4.3.2 Movimientos permitidos por el brazo compensador 58
4.3.3 Solución a problema de movimiento de brazo compensador 60
4.3.4 TCP o PCH del Extrusor de alto volumen junto con la unión 62
4.3.5 Creación de programa para impresión 3D 64
4.3.5.1 Creación de código G con slicer Ultimaker Cura 64
4.3.5.2 Realizar programa de impresión 3D en RoboDk 70
4.3.5.3 Creación de código G con slicer Slic3r 73

5 Resultados **78**
5.1 Movimientos del robot con solución en la conexión Cobot-Extrusor-Brazo compensador 78
5.2 Dibujos creados por cobot a partir de programa en RoboDk con plumón de pizarra como herramienta 79
5.3 Impresión de Cilindro de diámetro 50 [mm] a partir de programa en RoboDK con extrusor de alto volumen y brazo compensador 80

6 Conclusiones y Recomendaciones **86**

7 Anexo **92**
7.1 Guía de uso y mantenimiento del sistema 92

Lista de Tablas

4.1 Especificación [22] 12
4.2 Movimiento [22] 13
4.3 Características Físicas [22] 13
4.4 Medidas PCH de extrusor de alto volumen en configuración final 63

Lista de Figuras

3.1 Herramientas asistidas por computadora [5]. 4
3.2 Movimiento lineal G-Code (elaboración propia) 7
3.3 Interfaz Robodk 9

4.1 Cobot UR5 [21]. 12
4.2 Espacio seguro de UR5 [21]. 14
4.3 Espacio de trabajo [21]. 16
4.4 Orificios de montaje con pernos M8. medidas en mm [21]. 17
4.5 Brida de salida de la herramienta, donde se monta la herramienta. Todas las medidas están en mm [21]. 18
4.6 Pantalla UR5 [21]. 21
4.7 Pantalla inicial encedida UR5 [21]. 22
4.8 Configurar UR5 [21]. 22
4.9 Encendido/Apagado UR5 [21]. 23
4.10 PCH o TCP (Creación Propia). 24
4.11 Configuración de TCP (Creación Propia). 25
4.12 4 puntos para TCP (Creación Propia). 25
4.13 Configuración de TCP (Creación Propia). 26
4.14 Posiciones brazo UR5 para TCP (Creación Propia). 26
4.15 Puntos tomados por UR5 para la obtención de TCP (Creación Propia). 27
4.16 Plano virtual (Creación Propia). 28
4.17 Pasos plano virtual (Creación Propia). 28
4.18 Pasos plano real (Creación Propia). 29
4.19 La herramienta en los puntos 2 y 3 (Creación Propia). 29
4.20 Plano creado por UR5 (Creación Propia). 30
4.21 Programa nuevo (Creación Propia). 31
4.22 Pestaña programa nuevo (Creación Propia). 31
4.23 Tipos de movimientos (Creación Propia). 32

4.24 Perfil de velocidad [21]. 33
4.25 Editor de programa (Creación Propia). 34
4.26 Punto de paso (Creación Propia). 34
4.27 Posición Robot UR5 mostrado en el interfaz (Creación Propia). 35
4.28 Wait o esperar (Creación Propia). 35
4.29 Programa básico final (Creación Propia). 36
4.30 Guardar como (Creación Propia). 37
4.31 Guardar archivos (Creación Propia). 37
4.32 Configurar Red (Creación Propia). 38
4.33 Conexión a UR5 desde PC con "Filezilla" (Creación Propia). 39
4.34 Cambiar opciones de adaptador (Creación Propia). 40
4.35 Activar IPv4 (Creación Propia). 41
4.36 Activar IPv4 (Creación Propia). 42
4.37 Interfaz RoboDK con archivo de UR5 (Creación Propia). 43
4.38 Conexión de RoboDK con el brazo UR5 (Creación Propia). 43
4.39 Coordenadas de TCP con plumón como herramienta (Creación Propia). 44
4.40 Plumón instalado en el cobot UR5 (Creación Propia). 44
4.41 Programa Inkscape 1.2 (Creación Propia). 46
4.42 Imagen cargada en programa Inkscape 1.2 (Creación Propia). 46
4.43 Imagen vectorizada en programa Inkscape 1.2 (Creación Propia). 47
4.44 Guardado de archivo DFX (Creación Propia). 47
4.45 Imagen vectorizada representada en AutoCAD (Creación Propia). 47
4.46 Imagen vectorizada en programa DXF to G-Code Converter (Creación Propia). . . . 48
4.47 Guardado de imagen vectorizada en formato código G (Creación Propia). 49
4.48 PCH tocando la superficie deseada (Creación Propia). 50
4.49 Obtener posición real de UR5 (Creación Propia). 50
4.50 Agregar target (Creación Propia). 51
4.51 Copiar las coordenadas de Target a Frame (Creación Propia). 52
4.52 Imagen en código g en RoboDK (Creación Propia). 53
4.53 Opciones principales de ajustes (Creación Propia). 53
4.54 Compensación de la herramienta a 0° respecto de X (Creación Propia). 55
4.55 Compensación de la herramienta a 90° respecto de X (Creación Propia). 55
4.56 Configuración actual de Cobot-Extruder y brazo compensador (Creación Propia). . 57
4.57 Configuración inicial de Cobot-Extruder y brazo compensador (Creación Propia). . 58
4.58 Plano brazo compensador serie 3 [25]. 59
4.59 Movimientos permitidos por brazo compensador serie 3 [25]. 59

4.60 Cruce de Coordenadas de Cobot y Brazo compensador (Creación propia). 60
4.61 Unión inicial (Creación propia). 61
4.62 Unión final (Creación propia). 61
4.63 Coordenadas base muñeca 3 (Creación propia). 62
4.64 Imagen representativa de toma de medidas TCP Extrusor (Creación propia). 63
4.65 Cilindro de 50 [mm] de diámetro en Inventor (Creación propia). 64
4.66 Opción de alta resolución para importar objeto en archivo STL(Creación propia). . 65
4.67 Comparativa entre cilindro exportado en alta resolución y baja resolución(Creación propia). 65
4.68 Interfaz slicer Ultimaker Cura con cilindro (Creación propia). 66
4.69 Opción para centrar la pieza u objeto (Creación propia). 66
4.70 Opciones de calidad de las capas(Creación propia). 67
4.71 Opciones de Paredes (Creación propia). 68
4.72 Opciones de cama de la pieza (Creación propia). 68
4.73 Opciones de velocidad (Creación propia). 69
4.74 Opciones de modos especiales (Creación propia). 69
4.75 Capas generadas por Ultimaker Cura con las opciones anteriores (Creación propia). 70
4.76 Ajustes para programa de impresión 3D (Creación propia). 71
4.77 Ajustes de eventos de programa de impresión 3D, velocidad máxima (Creación propia). 72
4.78 Ejecutar programa en el Cobot real (Creación propia). 72
4.79 Interfaz slic3r (Creación propia). 73
4.80 opciones de lineas y perímetro (Creación propia). 73
4.81 opciones falda y ala (Creación propia). 74
4.82 opciones de velocidad Slic3r (Creación propia). 74
4.83 Capas creadas por Slic3r para el cilindro de 50 [mm] de diámetro (Creación propia). 75
4.84 Opción para modificar las capas desde Robodk(Creación propia). 76
4.85 Opción para modificar las capas desde Robodk(Creación propia). 76

5.1 Dibujo realizado por Cobot y comparativa con imagen digital(Creación propia). . . 79
5.2 Otros dibujos creados por cobot y comparativa con la imagen digital 80
5.3 Primera impresión con pellets(Creación propia). 81
5.4 Primeras Impresiones con plástico reciclado triturado (Creación propia). 82
5.5 Pruebas de impresión para encontrar la velocidad óptima (Creación propia). 82
5.6 Impresión del cilindro con mesa desnivelada y sin inyección de aire (Creación propia). 83
5.7 Impresión del cilindro con inyección de aire (Creación propia). 84
5.8 Medidas de altura y espesor de la impresión de cilindro con inyección de aire 84

6.1 Configuración recomendada del sistema (creación propia). 88
6.2 Cabezal articulado plano 3arm [25]. 89

Nomenclatura

Unidad	Significado
cm^2	Área en centimetros cuadrado
in	Abreviatura de inch (pulgadas)
in/s	Velocidad en pulgadas por segundo
kg	Abreviatura de kilogramo
lb	Abreviatura de libra
mm	abreviatura de milímetros
mm/s	Velocidad en milímetros por segundo
Nm	Abreviatura de Newton por metros para términos de torque
rpm	Abreviatura de revoluciones por minuto
°/s	Velocidad en grados por segundo
Siglas	**Significado**
CAD	Computer-Aided Design (Diseño Asistido por Computadora)
CNC	Computer Numerical Control (Control Numérico por Computadora)
DXF	Drawing Exchange Format (Formato de Intercambio de Dibujo)
IP	Internet Protocol (Protocolo de internet)
ISO	International Organization for Standardization
PCH	Punto Central de Herramienta
PLA	Plástico Poliácido Láctico (Ácido Poliláctico)
STL	Standard Tessellation Language (Lenguaje de Teselado Estándar)
TCP	Tool Center Point (Punto Central de la Herramienta)
UR5	Universal Robots 5
Término	**Significado**
Cobot	robot colaborativo o brazo robótico colaborativo
Filezilla	Transferir archivos entre computadora y un servidor remoto
Frame	Sistema de coordenadas para definir la posición y orientación de un objeto
ECO-FAB-LAB	Laboratorio de Fabricación Ecológica de manufactura aditiva
M6	Tornillo de diámetro métrico de 6 milímetros
M8	Tornillo de diámetro métrico de 8 milímetros
G-Code (código G)	Lenguaje de programación para controlar máquinas CNC
RoboDk	Software de simulación y programación de robots
Software	Programas y datos que permiten a los pc's realizar tareas y ejecutar aplicaciones
Target	Punto de destino en el espacio tridimensional al cual se debe mover el robot

Capítulo 1

Introducción

La ingeniería busca ocupar los recursos de manera sostenible y sustentable, con el fin de prolongar y reducir el uso de los variados recursos naturales. En consecuencia cada vez se avanza con nuevas tecnologías que ayuden a concretar estas ideas como lo es la impresión 3D, la cual forma objetos a través de la impresión aditiva lo que se traduce como ocupar lo justo en el lugar adecuado. Esto viene de la mano de la robótica que a través de la ocupación de datos puede analizarlos y llevarlo a cabo para distintos fines. Resulta, por tanto, interesante desarrollar de alguna forma una impresora 3D de gran tamaño con la ayuda de un brazo robótico (UR5) el cual será el guía para ir incorporando material con el fin de ir creando la figura deseada.

Es el objetivo de este trabajo, por tanto, utilizando las herramientas que provee el mismo brazo robótico UR5 como también el software "Robodk", trabajando en paralelo con los *slicers* necesarios para indicar las configuraciones necesarias para un buen acabado de la pieza a imprimir, es realizar pruebas y comprobar la factibilidad de crear una impresora 3D de alto volumen tomando en cuenta un sistema que viene dado por el mismo UR5, el extrusión de alto volumen y un brazo compensador el cual soportará el peso total para evitar esfuerzos en el brazo robótico y finalizando con la creación de una guía de uso y mantenimiento para poder ocupar dicho sistema para imprimir de manera segura objetos que se deseen con material reciclado de plástico.

Capítulo 2

Objetivos y Alcances

A diferencia de un robot industrial tradicional, un Cobot está diseñado para trabajar e interactuar con personas y puede programarse previamente o en tiempo real de una forma sencilla, sin necesidad de una alta cualificación o formación previa.

El objetivo general de este trabajo de título es implementar un sistema de manufactura aditiva empleando un Cobot, lo que permitirá fabricar piezas de mayor tamaño y en un menor tiempo comparado con impresoras 3D convencionales.

Para realizar este trabajo de título se deben completar los siguientes objetivos específicos:

- Realizar un estudio del estado del arte acerca de la manufactura aditiva robotizada y sistemas auxiliares.
- Estudiar el software utilizado para transformar modelos CAD 3D a código G.
- Calibrar el TCP (punto central de herramienta) entre el soporte del Cobot y el punto de impresión.
- Realizar pruebas de funcionamiento, sincronización y precisión del sistema desarrollado.
- Elaborar guías de uso y mantenimiento para asegurar el correcto funcionamiento del sistema.

Capítulo 3

Marco Teórico

3.1 Manufactura Aditiva

La impresión 3D viene dada por un conjunto de tecnologías que son complementos entre sí para crear un objeto de diferentes tamaños, formas y estructuras, debido a la fundición del material y depositándolo capa por capa, de manera secuencial y siguiendo un patrón definido y exacto por el diseño digital. Este método es totalmente diferente a las técnicas ya conocidas cómo lo es el mecanizado, el cuál requiere quitar material del objeto [1, 2].

Para los movimientos necesarios para la creación de los objetos 3D, es necesario de una maquina que pueda mover el material en los 3 ejes (x,y,z) como lo hacen las impresoras y actualmente empresas han implementado utilizar cobots. Estos últimos se ha logrado demostrar que tienen una alta precisión al seguir las rutas complicadas de manera confiable [3].

Además de los movimientos físicos se debe tener conocimientos sobre la creación, modificación y análisis de los modelos que se requieran crear. De allí que la ingeniería cuenta con la ayuda de herramientas de soporte en todos los procesos que involucren diseño, análisis y fabricación de un producto. Y estas herramientas vienen dadas por las tecnologías asistidas como: CAD, CAE, CAM y CAPP [4]. Como se observa en la figura 3.1 el CAD es el software que más se utiliza en la manufactura aditiva debido a su capacidad de modelar distintos objetos y otorgar al usuario variadas herramientas para complementar sus creaciones [5].

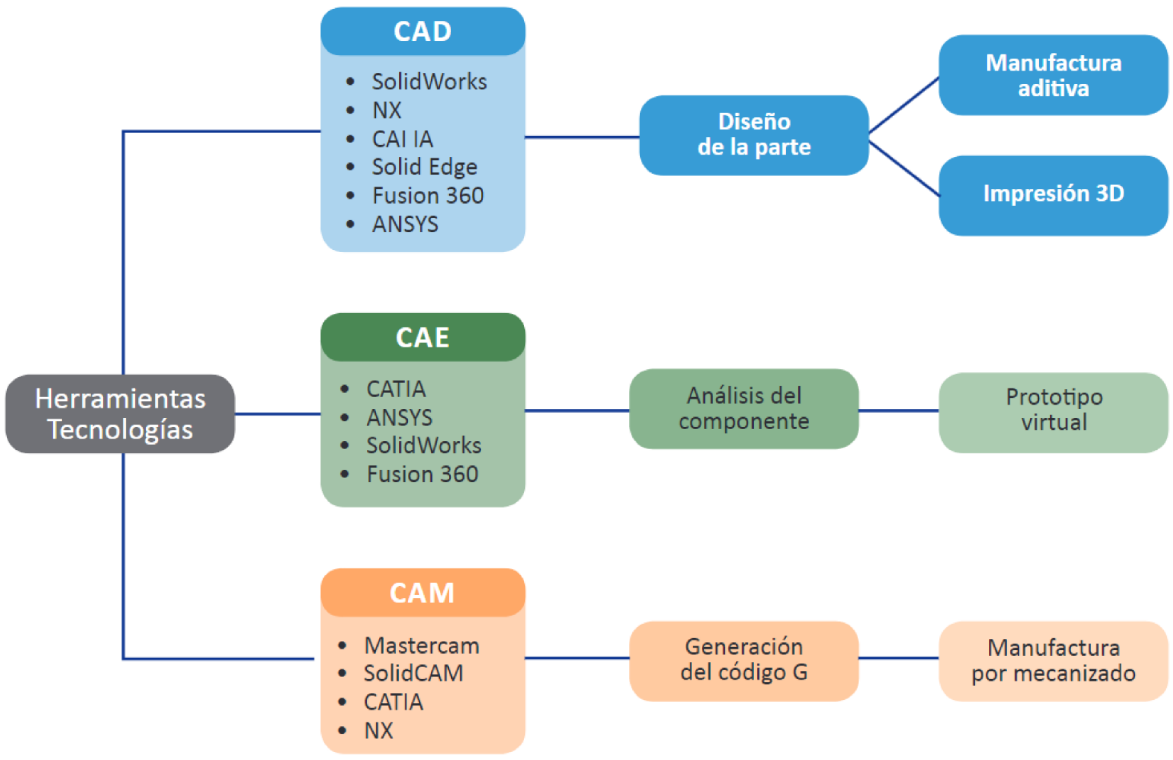


Figura 3.1: Herramientas asistidas por computadora [5].

Finalmente, se debe llevar la información del modelo digital a la máquina de manera que ésta pueda seguir la ruta y crear con la adición de material el objeto deseado. Para esto se trabajan con coordenadas espaciales las cuales se transfieren por el código G, con la velocidad y aceleración deseada. Para ello existen *softwares* que pueden tomar un modelo de CAD (Diseño Asistido por Computadora) y transformarlo a dicho código, estos últimos son los *slicers*. Los cuales calculan las interacciones del objeto 3D a movimientos, es decir convierte un modelo 3D a un archivo G-code [6].

Por lo tanto se puede decir que la manufactura aditiva o impresión 3D tiene 3 etapas definidas, las cuales son:

i. Modelado: En esta etapa se realiza el objeto con la ayuda de las herramientas CAD, de manera digital para la creación del objeto deseado.

ii. Exportación de archivo 3D: En este punto se exporta el archivo digitalizado con la geometría deseada en un archivo leíble para un *slicer*, el cual comúnmente es un archivo "STL"(*Stereolithography*). Estos archivos contienen la geometría general del objeto el cual no puede ser cambiado pero si su escala.

iii. Exportación de archivo de corte: En esta etapa final se indica las capas que deberá recorrer la impresora con la lectura del G-code. Este código es creado con la ayuda del *slicer* e indicaciones especificas que se requieran para la impresión deseada, en donde el G-code contiene los movimientos, velocidad, aceleración, temperatura.

3.2 CAD

Por siglas Diseño Asistido por Computadora es una tecnología diseñada para la documentación técnica y diseño digital. Esta tecnología abarca desde herramientas de modelación geométrica hasta aplicaciones a medida de análisis y/u optimizar el objeto. Ejemplos de estas aplicaciones incluye análisis de tensión-deformación de componentes, respuesta dinámica de distintos mecanismos, la transferencia de calor y abarca una medida del control numérico de programación. Para el estudio de este trabajo se utilizo el software proporcionado por Autodesk, el cual es Autodesk Inventor [7].

3.2.1 Autodesk Inventor

Es un software que entrega herramientas de diseño mecánico para modelos en 3D con documentación y simulación de productos de calidad profesional. Además combina potentes capacidades de diseño perimétrico, directo, de variadas formas y con las métricas según normas.

3.3 Slicer

3.3.1 Definición

Un *slicer* es una herramienta fundamental en el ámbito de la visualización de datos y el análisis de la información. Se trata de una funcionalidad que permite filtrar y segmentar los datos de manera interactiva, facilitando la exploración y comprensión de conjuntos de datos complejos. Los *slicers* proporcionan a los usuarios la capacidad de seleccionar y mostrar datos específicos, lo que contribuye a una toma de decisiones más informada y precisa en diversas áreas, como el análisis de negocios, la investigación científica y el diseño de políticas públicas [8].

3.3.2 Slicer para Impresión 3D

Un *slicer* para la impresión 3D es un *software* utilizado para convertir un modelo tridimensional en capas planas, conocidas como *'slices'* o rebanadas, que se imprimirán secuencialmente para construir el objeto en tres dimensiones. Este programa realiza operaciones como el cálculo de trayectorias de impresión, la generación de soportes y la configuración de parámetros de impresión, permitiendo una preparación y optimización eficiente del modelo para su producción física [9].

3.3.3 Tipos de slicer para impresión 3D

Existen diferentes tipos de slicers para la impresión 3D, los cuales facilitan al usuario preparar y optimizar los modelos digitales para su impresión, los cuales son:

- *Slicers* de escritorio: Son software o programas de código abierto o comercial que se ejecutan en computadoras de escritorio. Algunos ejemplos populares son "Ultimaker Cura", "PrusaSlicer", "Simplify3D" y "Slic3r". Estos *slicers* ofrecen una amplia gama de configuraciones y opciones para ajustar los parámetros de impresión [10].
- *Slicers* basados en la nube: Son herramientas en línea que permiten subir tus modelos 3D y configurar los parámetros de impresión en un entorno basado en la web. Algunos ejemplos son "OctoPrint", "AstroPrint" y "3DPrinterOS". Estos *slicers* basados en la nube brindan la conveniencia de acceder y controlar la impresión desde cualquier lugar con conexión a internet [11].

3.3.4 G-code

El código G es un conjunto de instrucciones utilizadas en la programación de máquinas herramienta de control numérico por computadora (CNC). Estas instrucciones, escritas en un lenguaje específico, permiten controlar el movimiento de la herramienta y las acciones de mecanizado durante el proceso de fabricación. El código G define parámetros como la posición de inicio y final, la velocidad de avance, la trayectoria de la herramienta, el cambio de herramientas y otras operaciones necesarias para producir una pieza con precisión y eficiencia en una máquina [12].

Los comandos más comunes en la impresión 3D son el movimiento lineal G0 y G1, los cuales son los encargados de mover los ejes (x,y,z). Los parámetros XYZ mostrados en la figura 3.2 definen la ubicación en la que la impresora deberá moverse para realizar el trazado del modelo 3D. El E define la cantidad de material extruido en milímetros y el F define el velocidad de avance, normalmente en mm por segundo. En las impresoras 3D actuales G0 y G1 realizan prácticamente lo mismo y sólo existe la convención de que G0 se utiliza para movimientos de viaje y G1 para movimientos en el cual se extruye el material fundido [13].

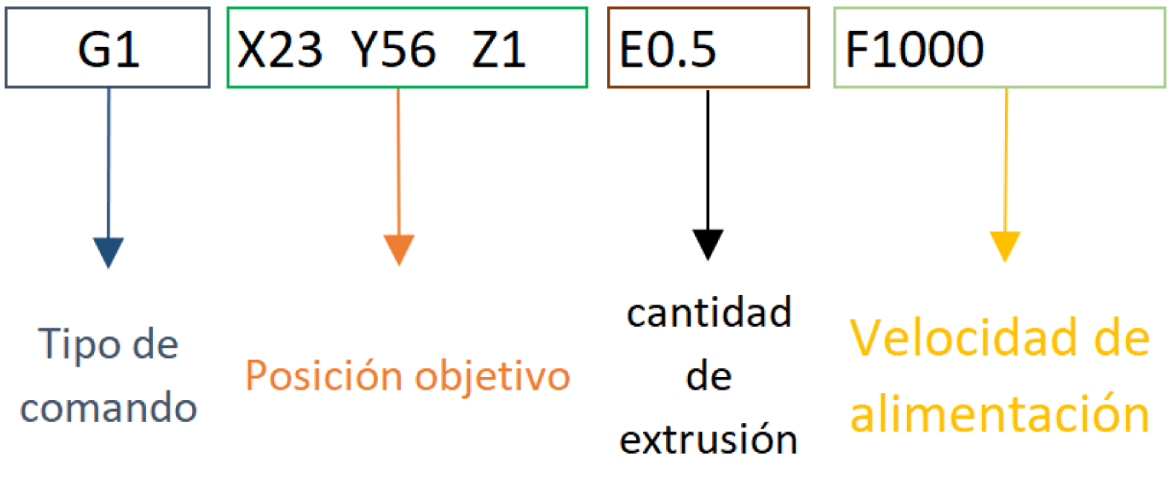


Figura 3.2: Movimiento lineal G-Code (elaboración propia)

3.4 Cobots

Los llamados robots colaborativos, son brazos robóticos diseñados para trabajar en colaboración directa y segura con humanos en un entorno de producción o trabajo conjunto. A diferencia de los robots industriales convencionales, los cobots están equipados con sensores y características de seguridad avanzadas que les permiten operar de manera segura en proximidad a los humanos, sin necesidad de barreras físicas de separación. Estos robots están diseñados para realizar tareas repetitivas, monótonas o peligrosas, al tiempo que interactúan de forma segura y flexible con los operarios, permitiendo una colaboración estrecha entre humanos y máquinas en entornos de trabajo compartidos [14]. Lo que los diferencia de los robots industriales aparte de su tamaño y capacidad es la velocidad, donde comúnmente se mueven a una velocidad máxima de 1.5 m/s (que es la mitad de los industriales), no obstante la velocidad regular en el mercado es de 1 m/s o menos, esto debido a que trabajará con humanos a poca distancia y en el cuál sólo el operario debe arrancar el cobot [15].

3.4.1 Tipos de Cobots

En el mercado actual, existen diferentes tipos de cobots diseñados para adaptarse a diversas aplicaciones y entornos de trabajo. A continuación, se mencionan algunos de los tipos comunes de cobots disponibles [16]:

- Cobots de manipulación: Estos cobots están diseñados principalmente para tareas de manipulación y manejo de materiales. Pueden ser utilizados para levantar, transportar y colocar objetos en una variedad de entornos industriales.

- Cobots de ensamblaje: Estos cobots están diseñados específicamente para realizar tareas de ensamblaje, como unir componentes, realizar soldaduras o ensamblar productos finales. Son altamente precisos y pueden adaptarse a diferentes configuraciones y requerimientos de ensamblaje.

- Cobots de paletizado: Estos cobots se utilizan en aplicaciones de paletizado y despaletizado, donde se requiere la manipulación y organización eficiente de cargas en paletas. Pueden trabajar en colaboración con operarios para agilizar y optimizar los procesos de logística y manipulación de carga.

- Cobots de inspección: Estos cobots están diseñados para realizar tareas de inspección y control de calidad. Pueden utilizar sensores y cámaras para verificar la calidad de los productos, realizar mediciones y detectar defectos o anomalías durante el proceso de fabricación.

- Cobots de atención al cliente: Algunos cobots están diseñados específicamente para interactuar con los clientes en entornos minoristas o de servicio. Pueden proporcionar información, responder preguntas frecuentes, guiar a los clientes o realizar transacciones básicas.

Cabe mencionar que la lista anterior es un ejemplo de tipos de cobots en el mercado. Ya que está en constante evolución y desarrollo de nuevos tipos y aplicaciones a medida que la tecnología y necesidades en la robótica colaborativa avanzan.

3.5 Manufactura Aditiva con cobots

La manufactura aditiva utilizando cobots se refiere a la aplicación de la tecnología de impresión 3D en combinación con robots colaborativos en los procesos de fabricación. En este enfoque, los cobots desempeñan un papel crucial debido a su alta precisión y movimientos complejos, estos pueden sustituir a una impresora convencional y realizar los movimientos con las lecturas entregadas por un g-code. Estos robots colaborativos trabajan en estrecha colaboración con los operarios, optimizando la eficiencia y la precisión de la manufactura aditiva, al tiempo que mantienen altos estándares de seguridad en el entorno de trabajo compartido como se menciono en los puntos anteriores. La integración de cobots en la manufactura aditiva permite una mayor automatización, un mayor rendimiento y la capacidad de producir componentes complejos con alto nivel de detalle y repetibilidad, abriendo nuevas posibilidades en la producción de piezas personalizadas, prototipos rápidos y producción en serie, ya que permite a los usuarios imprimir material en alto volumen y velocidad lo que genera una sustitución de la manufactura a gran escala utilizando la adición de material para crear grandes piezas [9].

3.6 RoboDk

En las secciones anteriores se ha hablado sobre la impresión 3D y la utilización de cobots como reemplazo de las impresoras convencionales. Por lo que cabe mencionar que debe existir una conexión entre el robot y el diseño digital, en donde este diseño es representado por un código G. Por lo que se debe tener en cuenta el traspaso de esta información al robot colaborativo y es por esto que se necesita el software "Robodk".

Este software es una herramienta avanzada de simulación y programación fuera de línea utilizada en la robótica principalmente. El cual con una interfaz gráfica intuitiva permite a los usuarios simular y programar el movimiento de robots como se muestra en la figura 3.3, tanto industriales como colaborativos, en un entorno virtual. "RoboDK" facilita la creación de trayectorias, la programación de tareas complejas y la generación del código al robot, tomando el código G del modelo digital y transcribiéndolo en un código compatible con diferentes marcas y modelos de robots, los cuales tiene en su biblioteca más de 200 robots de 50 fabricantes entre los cuales están: ABB, Fanuc, KUKA, Yaskawa, Staubli y Universal Robots. Además, permite la importación de modelos CAD y la configuración de entornos de trabajo para una simulación precisa. Con características como la detección de colisiones, la programación de visión artificial y la generación de código optimizado. Este software se ha convertido en una herramienta de gran utilidad para las empresas y personas que quieren adoptar una nueva forma de manufactura como lo es la impresión 3D. Su capacidad para simular y programar *offline* agiliza el desarrollo de aplicaciones robóticas, reduciendo los tiempos de puesta en marcha y mejorando la eficiencia en los procesos de fabricación [17].

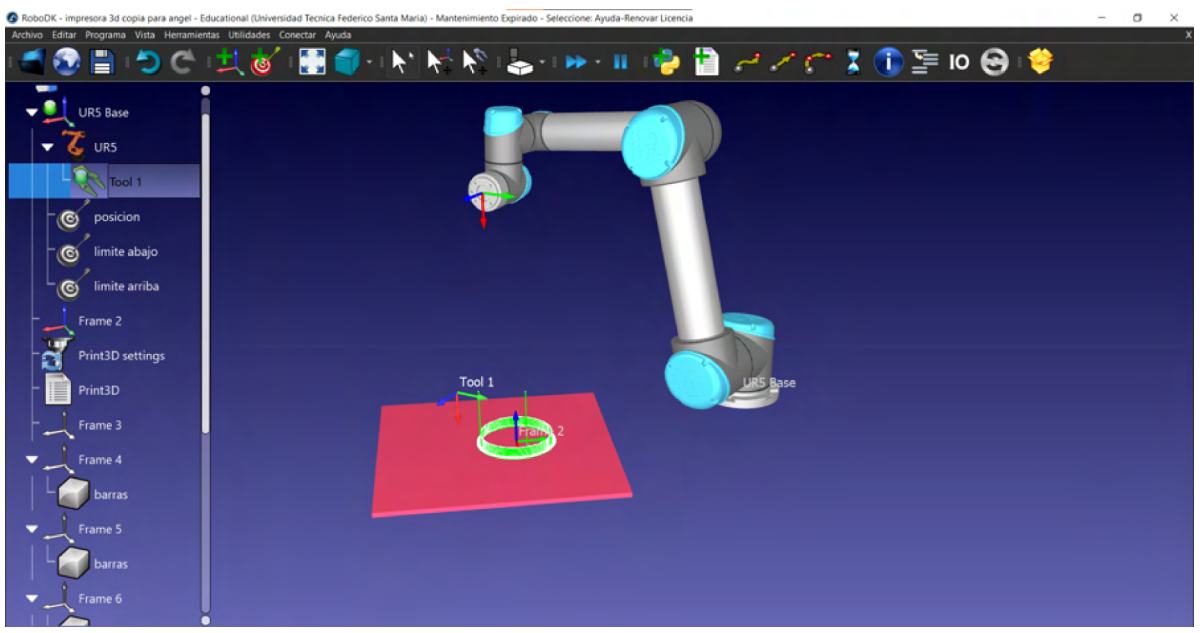


Figura 3.3: Interfaz Robodk

3.6.1 Ventajas y desventajas de la programación fuera de línea o simulación

Las ventajas vienen dadas debido que para garantizar que los algoritmos de control avanzados estén funcionando de manera correcta y óptima para el trabajo deseado, sé es necesario antes de mover el cobot real sea posible ver una simulación digital de manera que el usuario pueda observar los movimientos y asegurar que el espacio de trabajo no se verá comprometido como también las distintas partes que conforman la impresión 3D robotizada [17].

Estás ventajas son:

- Es intuitivo, ya que se puede mover en un entorno 3D con arrastre y técnicas de caída.
- Ofrece variedades de soluciones a un problema, ya que permite resolverla de forma visual.
- La programación es mas eficiente, debido que el robot está detenido antes de generar un nuevo programa.

Las desventajas vienen dadas por:

- La simulación virtual no es absolutamente precisa cuando se trata de representar la realidad.
- El proceso toma un tiempo extra, ya que se está desarrollando primero la simulación antes de ejecutarlo.
- Puede generar que el enfoque se esté dando solamente en el problema de la simulación, en lugar de los problemas reales de producción.

Capítulo 4

Materiales y Métodos

4.1 Universal Robots 5 (UR5)

Los robots colaborativos son diseñados para trabajar compartiendo el mismo espacio con las personas con las medidas de seguridad necesarias e incluso realizar tareas en donde se deba interactuar entre ellos. Estos se crearon en 1996 por profesores de la *Nothwestern University* en Illinois, EE. UU [18]. Luego la Empresa KUKA en una colaboración con el Centro Aeroespacial Alemán se desarrolló el primer robot colaborativo de uso industrial, el LBR 3 en el año 2004 [19]. Sin embargo, no fue hasta 2010 donde que el diseño, producción y su uso empezó a extenderse. Tras el desarrollo del LBR 3 por la Empresa KUKA, grandes empresas del sector empezaron a desarrollar y lanzar sus propios modelos de robots colaborativos, como lo son FANUC y Universal Robots, donde este último lanzo por primera vez el cobot llamándolo UR5 [20].

En la figura 4.1 se muestra un robot UR5 de 6 ejes, este modelo es anterior al modelo UR5e con una carga máxima de 5 kg con 15 funciones avanzadas de seguridad las cuales son regulables y probado de acuerdo con las normas ISO 13849:2008 PL d [22] .

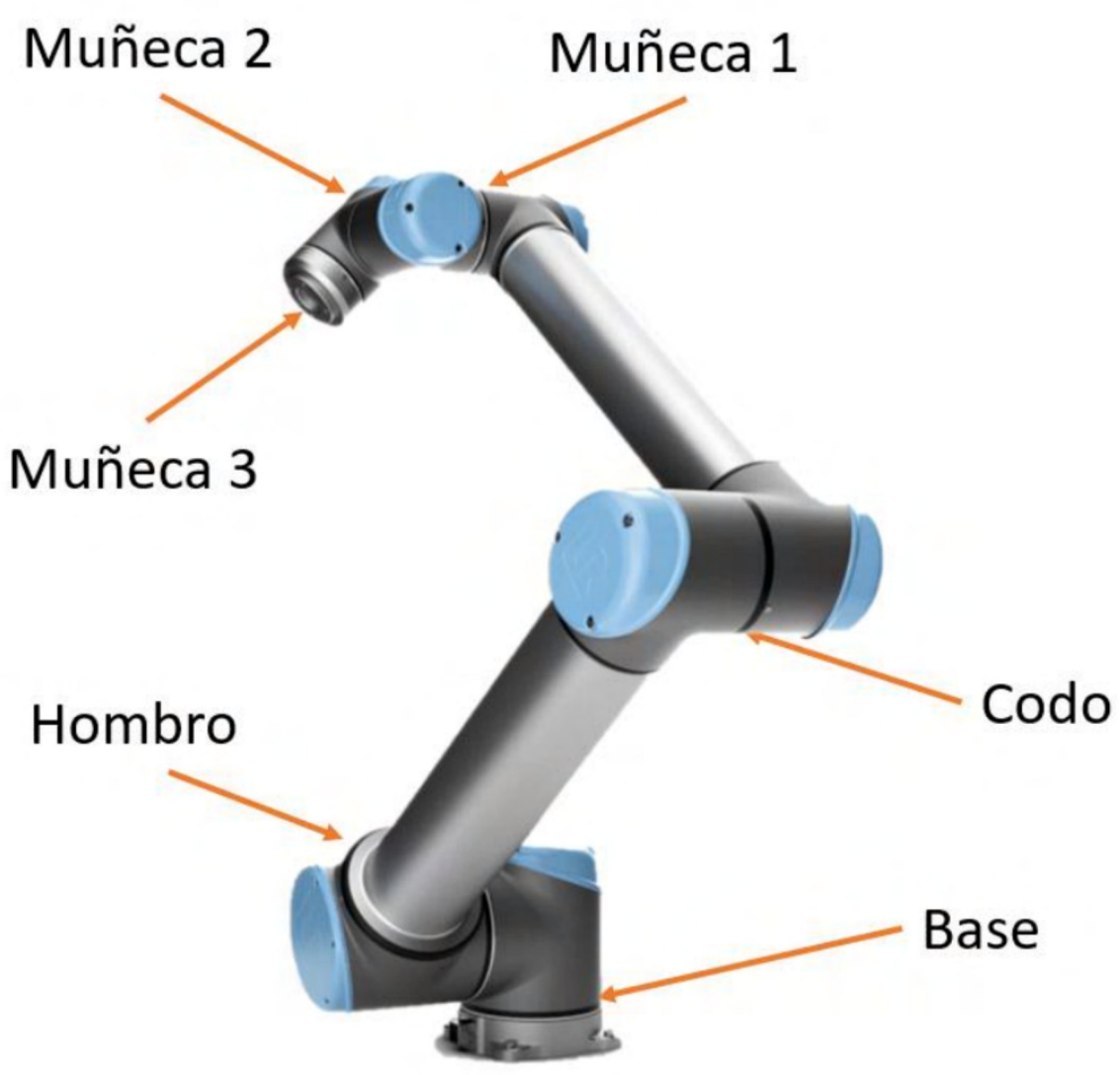


Figura 4.1: Cobot UR5 [21].

Al momento de utilizar el robot se debe conocer las especificaciones mostradas en la tabla 4.1, las cuales indican la carga útil que el robot puede soportar, el alcance máximo que tiene el brazo extendido de 850 [mm], los grados de libertad las cuales viene dado de las 6 articulaciones mostradas en la figura 4.1 las cuales pueden girar +- 360 grados cada una como se indica en la tabla 4.2 esto quiere decir que la articulaciones al momento de llegar al grado +360° debe devolverse a -360° lo mismo viceversa y para controlar el robot se tiene una pantalla táctil con soporte la cual permite al operario ocupar las diferentes funciones incluidas en el robot y además verlo de manera gráfica en el monitor.

Tabla 4.1: Especificación [22]

Carga útil	5 kg / 11 lb
Alcance	850 mm / 33,5 in
Grados de libertad	6 articulaciones giratorias
Programación	Interfaz gráfica del usuario Polyscope con pantalla táctil de 12” con soporte

Tabla 4.2: Movimiento [22]

Movi. del eje del brazo Robot.	Radio de acción	Velocidad máxima
Base	+-360°	+-180°/s
Hombro	+-360°	+-180°/s
Codo	+-360°	+-180°/s
Muñeca 1	+-360°	+-180°/s
Muñeca 2	+-360°	+-180°/s
Muñeca 3	+-360°	+-180°/s
Herramienta típica		1 m/s - 39,4 in/s

Las características físicas de UR5 presentadas en la tabla 4.3 indican principalmente que su peso es de 18,4 [kg] lo cual es ligero comparado con su volumen, longitud del cable del brazo robótico de 6 metros el cual permite operar el robot a una distancia considerable de la fuente de poder y material de aluminio y plásticos de PP por lo que se debe tener cuidado sobre las cargas externas al robot debido a que se podría dañar considerablemente.

Tabla 4.3: Características Físicas [22]

Tamaño de la caja de control (an. x al. x la.)	475 x 423 x 268 mm/ 18,7 x 16,7 x 19,6 in
Peso	15 kg / 33,1 lb
Materiales	Acero

El brazo robótico UR5 está principalmente enfocado para empresas pequeñas y medianas con una producción de menor medida, aunque también es posible que encuentren un lugar en empresas industriales de gran flujo. Además, el costo económico de adquirir estos tipos de robots son mucho menor comparados de robots industriales, lo cuales están hechos de materiales más pesados y reducen su movilidad para un entorno que requiere estar en constante cambio. Un punto es que el robot al manipular objetos o aparatos de cargas menores a uno industrial implique que no se necesite una pantalla protectora entre el cobot y el operario. Esto es debido, a que en el caso que hubiera un impacto con una persona o con algún elemento del entorno, la inercia no es suficiente para causar daños considerables, ya que como se mencionó la carga útil del UR5 es de 5 kilogramos, lo cuál es soportable para una persona común y reduce los riesgos de accidentes [23].

4.1.1 Seguridad

Para determinados espacios de trabajos se debe prestar atención a los posibles peligros de enganche, esto debido a las propiedades físicas del brazo robótico. Se define una zona para movimientos radiales cuando la junta de la muñeca 1 (ver figura 4.1) está a una distancia de al menos 750 [mm]

de base del robot. La otra zona está a 200 [mm] de la base del robot, cuando el movimiento se produce en dirección tangencial. (ver figura 4.2). [21]

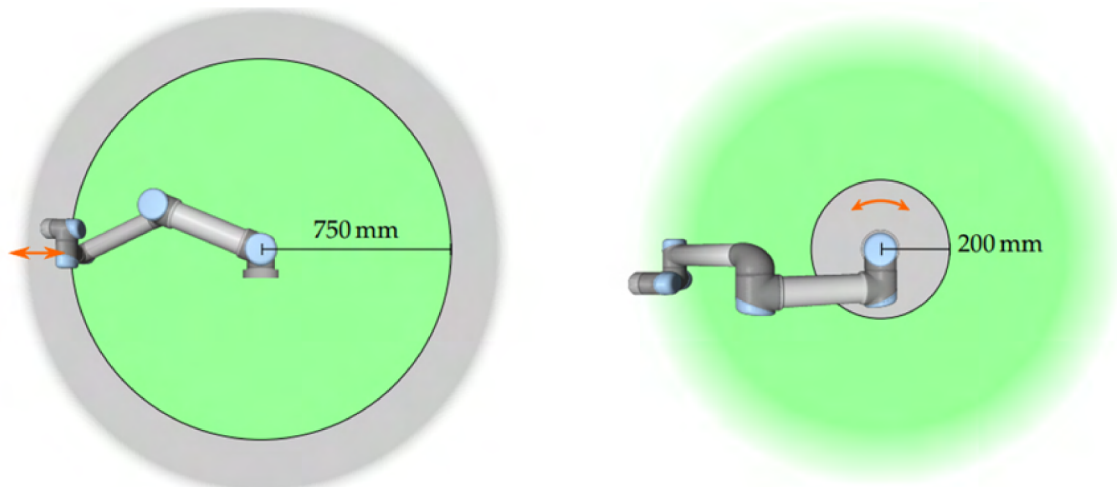


Figura 4.2: Espacio seguro de UR5 [21].

Con lo anterior este tipo de consideraciones se debe tener en cuenta debido a que si se pone en marcha el brazo UR5 puede colisionar o pasar a topar algún elemento externo o al operario más cercano. Por lo que se debe tener limitaciones de espacio en donde solo trabaje el robot para no afectar el trabajo de este y posibles daños irremediables. Por esto se recomienda que antes del primer encendido del brazo UR5 se realice una evaluación de los riesgos. Una evaluación de riesgos es que se averigüe la configuración de seguridad adecuada, así como la necesidad de pulsadores de parada de emergencias adicionales u otras medidas de protección necesarias para la aplicación del robot [21].

Algunas funciones que están relacionadas a la seguridad se diseñaron expresamente para aplicaciones robóticas colaborativas. Estas se pueden configurar a través de la configuración de seguridad y son especialmente importante cuando se tratan de riesgos específicos en la evaluación de riesgos realizadas por el integrador [21]:

- Limitación de fuerza y potencia: Se utiliza para reducir la fuerza de sujeción y la presión ejercida por el robot en la dirección del desplazamiento en caso de colisión entre el robot y el operador.

- Limitación de momento: Se utiliza para para reducir los altos niveles transitorios de energía y fuerza de impacto en caso de colisión entre el robot y el operador, mediante la reducción de velocidad del robot.

- Limitación de la posición de juntas y PCH: Se utiliza especialmente para reducir los riesgos relacionados con ciertas partes del cuerpo. Por ejemplo, para evitar movimientos hacia la cabeza o el cuello durante la configuración del programa.

- Limitación de la orientación de herramienta y PCH: Se utiliza especialmente para reducir los riesgos relacionados con ciertas zonas y funciones de la herramienta y la pieza. Por ejemplo, para evitar que los bordes afilados apunten al operador.
- Limitación de velocidad: Se utiliza especialmente para asegurar una velocidad baja del brazo robótico. Por ejemplo, para darle tiempo al operador para que evite el contacto con el brazo robótico.

Cuando se realice una evaluación de riesgos en una aplicación robótica colaborativa, es especialmente importante cubrir [21]:

- La gravedad de las colisiones potenciales individuales.
- La probabilidad de que ocurran colisiones potenciales individuales.
- La posibilidad de evitar las colisiones potenciales individuales.

Universal Robots incluye riesgos potenciales e importantes que ha identificado y que los integradores deben tener presentes. Tenga en cuenta que una instalación robótica concreta puede darse otros riesgos importantes [21]:

1. Atrapamiento de dedos entre el pie y la base del robot (ver figura 4.1).
2. Atrapamiento de dedos entre muñeca 1 y muñeca 2 (ver figura 4.1).
3. Piel penetrada por juntas y bordes afilados en la herramienta o su conector.
4. Piel penetrada por puntas y bordes afilados en obstáculos que haya cerca de la guía del robot.
5. Cardenales causados por golpes del robot.
6. Torceduras o fracturas óseas debidas a golpes entre una carga pesada y una superficie dura.
7. Consecuencias debidas a tornillos flojos que sujetan el brazo robótico o la herramienta.
8. Objetos que se caen de la herramienta, por ejemplo, por un mal agarre o una interrupción del suministro eléctrico.
9. Errores debido a que hay distintos botones de parada de emergencia para diferentes máquinas.
10. Errores debido a que se han producido cambios no autorizados en los parámetros de configuración de la seguridad.

4.1.2 Espacio de trabajo de UR5

Como se menciono en la sección 4.1 el robot ocupa 850 [mm] desde la junta a la base. Entonces, al momento de elegir la zona donde se instalará el robot, es fundamental tener en cuenta el volumen cilíndrico justo encima y debajo de la base del UR5 como se observa en la figura 4.3. Por lo que es importante evitar acercar la herramienta a este volumen cilíndrico en la medida de lo posible, debido a que provocaría que las juntas del robot se movieran rápido, aunque la herramienta lo hiciera despacio, lo que causa que el UR5 trabaje de forma ineficiente y dificultando la realización la evaluación de riesgos.

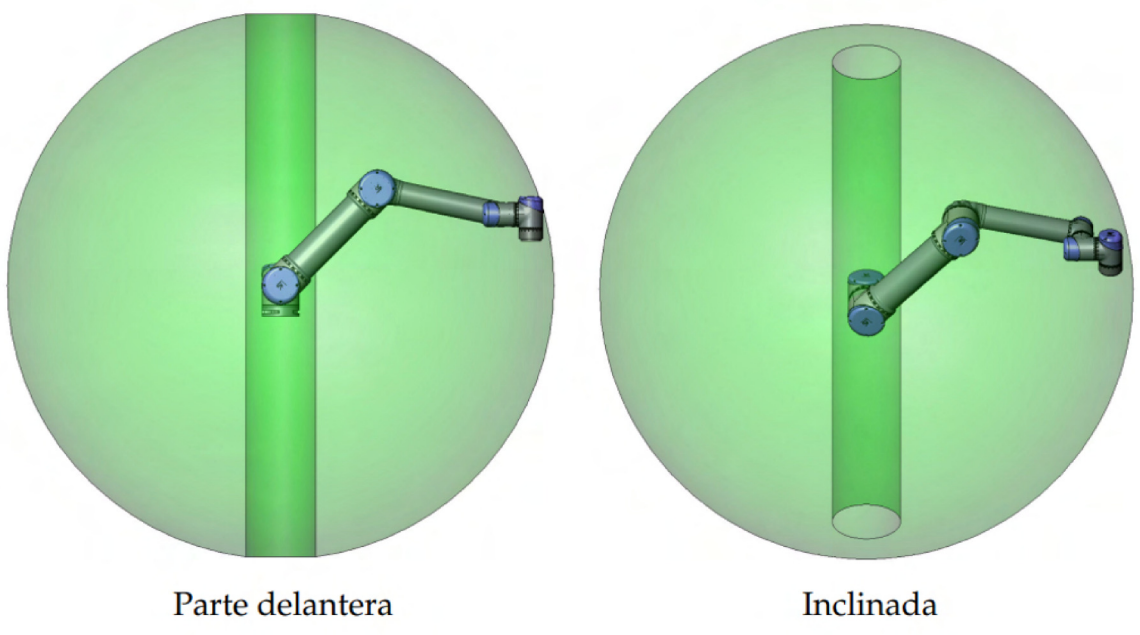


Figura 4.3: Espacio de trabajo [21].

4.1.3 Montaje

Para montar el brazo robótico se utiliza pernos M8 en los cuatros orificios de 8.5 [mm] de la base. El fabricante recomienda apretar estos pernos con un par de torsión de 20 [Nm]. Para un montaje preciso, se suministran dos orificios de diámetro 8 [mm] para utilizar como clavija. La figura 4.4 se muestra dónde taladrar los orificios y montar los tornillos [21].

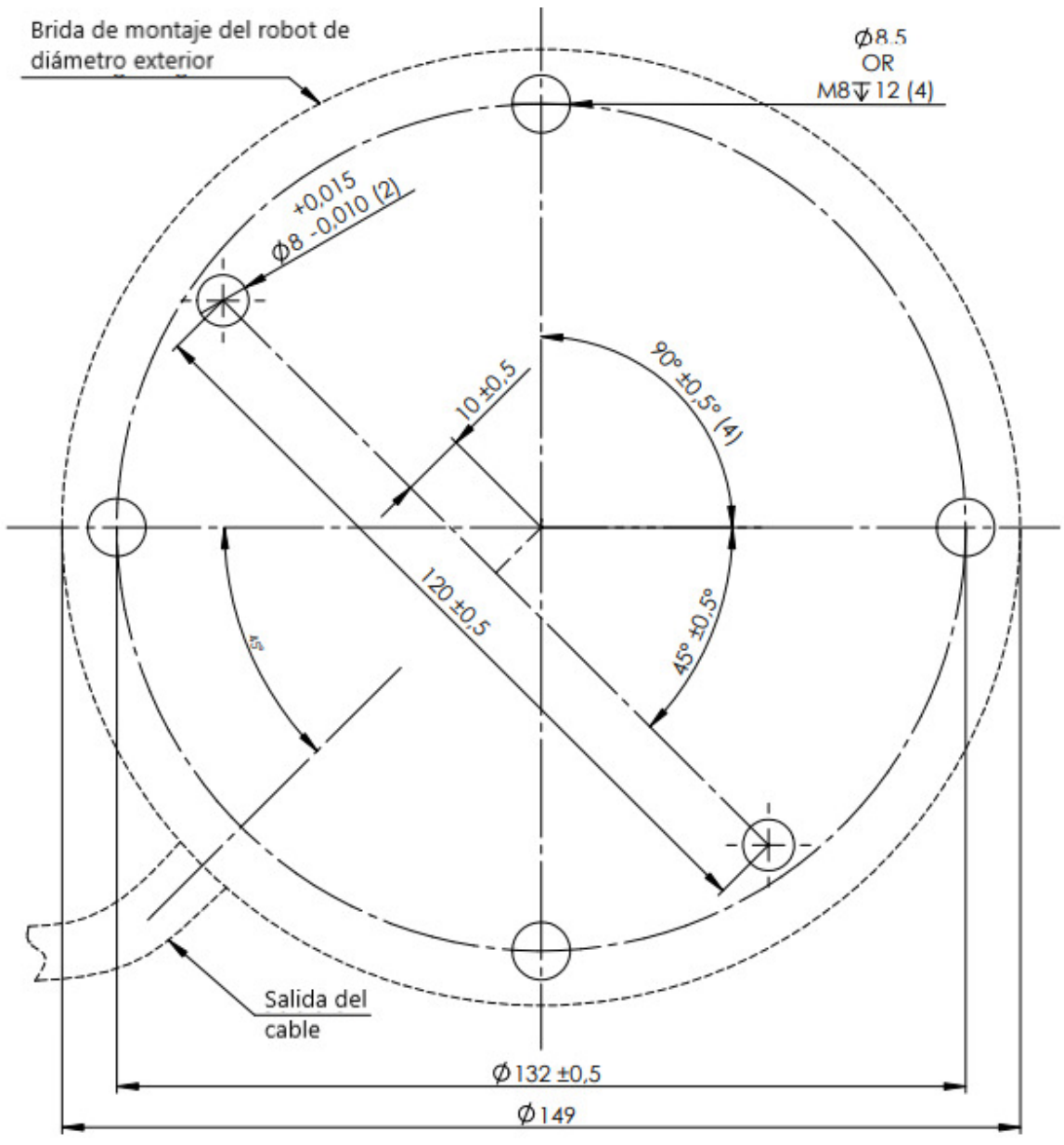


Figura 4.4: Orificios de montaje con pernos M8. medidas en mm [21].

La superficie donde se debe instalar el robot debe ser resistente, lo bastante fuerte para soportar al menos 10 veces el par de torsión total de la junta de la base y al menos 5 veces el peso del UR5. Se debe tener en cuenta que no debe haber vibraciones, debido a que afectará el funcionamiento óptimo y puede activar las paradas de emergencias. Además, si el robot se monta sobre una plataforma móvil, la aceleración de la base de montaje móvil debe ser muy baja. Ya que una aceleración alta puede hacer que el robot se pare por pensar que ha chocado con algún elemento externo [21].

4.1.4 Herramienta

El robot cuenta con una brida donde se coloca la herramienta, la cual consta con cuatro orificios de rosca M6 para acoplarla al UR5. Los pernos deben ajustarse con 9 [Nm]. En caso de que se desee volver a colocar de forma muy precisa la herramienta, se suministra el orificio de diámetro de 6 [mm] para utilizar como clavija. En la figura 4.5 se muestra el esquema de los orificios con

roscas para montar los tornillos de la brida.

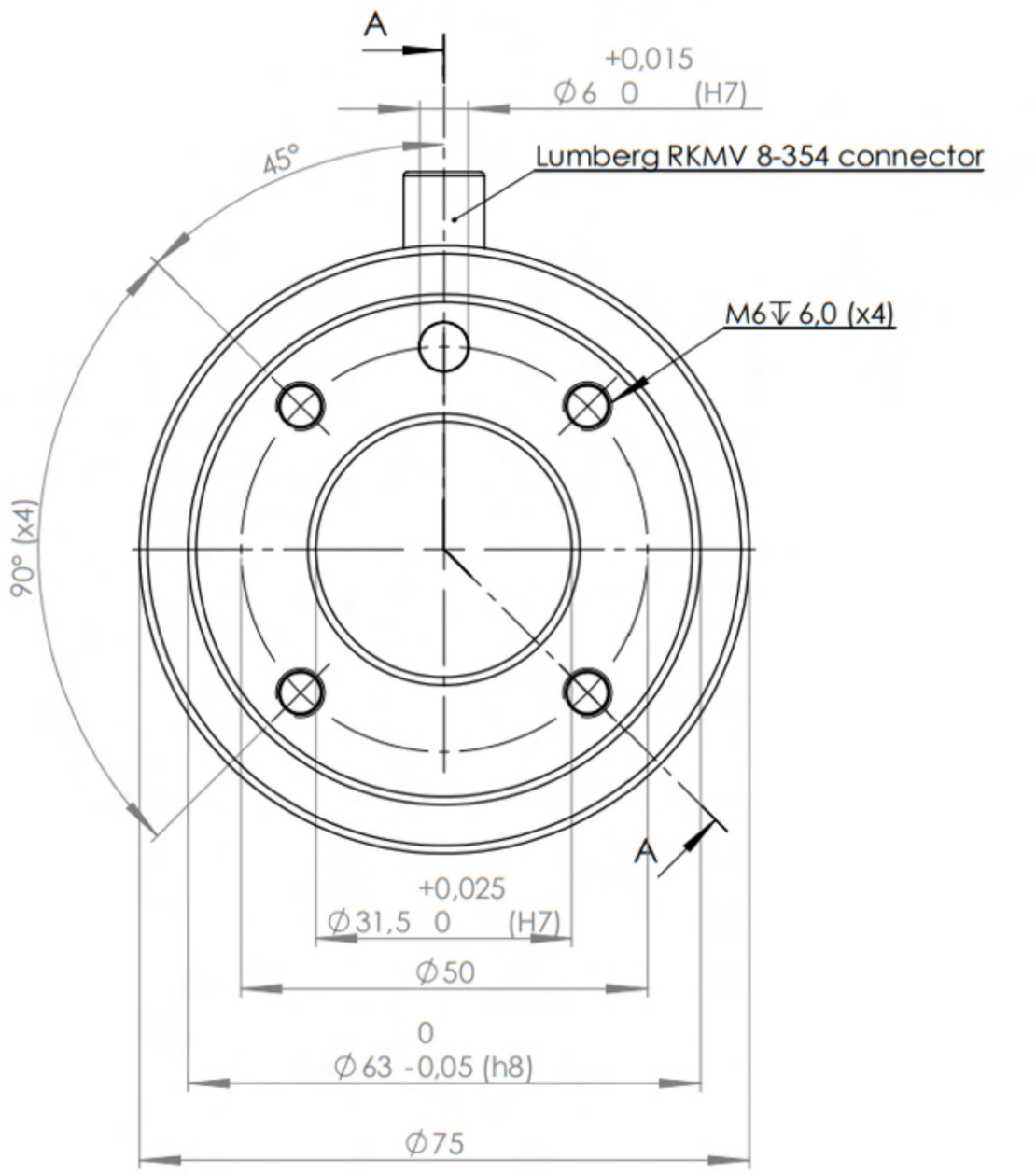


Figura 4.5: Brida de salida de la herramienta, donde se monta la herramienta. Todas las medidas están en mm [21].

4.1.5 Normas

Una norma no es una ley a la cual no se puede infligir. Mas bien es un documento que se desarrolla por partes interesados de un sector determinado, en el que se definen ciertos requisitos normales de seguridad y para el rendimiento del producto o un grupo de productos [21].

Los brazos robóticos si bien son una tecnología reciente, existen organismos de estandarización como lo es la ISO (*International Standard Organization*) que ya han desarrollado normativas respecto a los tipos de robots colaborativos que existen y las medidas de seguridad pertinentes que se deben seguir dependiendo de la aplicación de los brazos robóticos. Las normativas en concreto son ISO 10218-1:2011, ISO 10218-2:2011, ISO 13849:2008 PL d [21, 22].

ISO 13849:2008 PL d: Es una norma británica que implementa el Reino Unido, la cual es idéntica a la norma ISO 13849: 2006 incorporando el corrigendum de febrero [24]. La cual proporciona requisitos de seguridad y orientación sobre los principios para el diseño y la integración de partes de los sistemas de control relacionadas con la seguridad (SRP/SC), incluido el diseño de software. Para estas partes de SCRP/CS, especifica características que incluyen el nivel de desempeño requerido para llevar a cabo las funciones de seguridad. Se aplica a SRP/CS, independientemente del tipo de tecnología y energía utilizada (eléctrica, hidráulica, neumática, mecánica, etc). Para todo tipo de maquinaria. No especifica las funciones de seguridad o los niveles de rendimiento que se utilizarán en un caso particular [24].

ISO 13849-1:2006 proporciona requisitos para SRP/CS que utilizan sistemas electrónicos programables. No obstante, se pueden utilizar los principios dados, como categorías o niveles de rendimiento [24].

ISO 10218-1:2011 es una norma se aplica al fabricante del robot. La cual separa las características que se pueden encontrar en los robots robóticos en las siguientes [21]:

- Parada de seguridad monitorizada: Cuando la interacción del robot-operario no es constante, esta característica se utiliza para que la persona sólo necesite entrar puntualmente al espacio del robot. Así entonces, durante no se encuentre en el espacio de trabajo estará en funcionamiento de forma autónoma. Además, cuando el robot está trabajando, se frena completamente mediante la acción de los frenos de este cuando una persona se encuentra dentro de la zona que se delimito como trabajo. Lo cual permite que el robot se detenga sin esperar que se realice una parada de emergencia.

- Guía manual: Es un modo con el cual se puede instruir de forma manual al robot un determinado camino. Existe un mecanismo que es capaz de liberar el robot y que permite moverlo de forma manual hasta una posición determinada, esta posición es almacenada como un waypoint, con esto se pueden crear distintos tipos de aplicaciones de posicionado o pick and place. Para el caso de los robots el control de fuerza y velocidad es necesario para evitar daños y accidentes a su alrededor

- Monitorización de la velocidad y separación: Para obtener información sobre el entorno en el que trabaja el robot, se necesita de distintos tipos de sensores, ya sean láseres o sistemas de visión que sean capaces de detectar si el robot se va a colisionar con algún elemento externo o persona cercana. Con esto se puede variar la velocidad con la cual se mueva el brazo robótico al aproximarse a algún obstáculo. Este modo es necesario cuando la interacción de robot-humano es constante, la cuál se puede configurar para que se habitué a la aplicación en el que se desarrolle el robot.

- Limitaciones de potencia y fuerza: Como los robots colaborativos trabajan con potencias notoriamente menores a los robots industriales, este modo es el que va dirigido a ellos ya que los anteriores pueden ser aplicados a los robots industriales. No requiere de ningún tipo de visión ni sensor que controle el entorno de trabajo ya que se detiene en el momento en el que se produce un contacto con el objeto.

Según *British Standard* [24], esta normativa se encuentra en revisión, por lo que es posible que pronto estos tipos de operaciones en colaboración entre robot y humano se modifique o amplié. Para este proyecto se utiliza un robot UR5 de "Universal Robots" y se puede considerar que se esta trabajando con la aplicación colaborativa de Limitaciones de potencia y fuerza, ya que, en caso de contacto o algún movimiento brusco se detendrá.

La segunda parte (ISO 10218-2) se aplica al integrador del robot, pues trata de la instalación y el diseño de la aplicación robótica. Los redactores de la norma pensaron de forma implícita en los robots industriales tradicionales, protegidos mediante vallas y cortinas de luz. Los robots de UR están diseñados con límites de fuerza y potencia habilitados en todo momento [21][24].

4.1.6 Iniciar Cobot

Para controlar el robot se tiene una pantalla táctil como se muestra en la figura 4.6, esta consta de dos botones frontales uno de parada de emergencia de color rojo el cual tiene como función parar el robot si hay un mal funcionamiento o si por alguna razón se activa la parada de emergencia poder desactivarla una vez solucionado el problema. El segundo botón de color gris y luz es el encendido de pantalla y poder así controlar el robot, también funciona para apagar el robot si la pantalla táctil no responde.

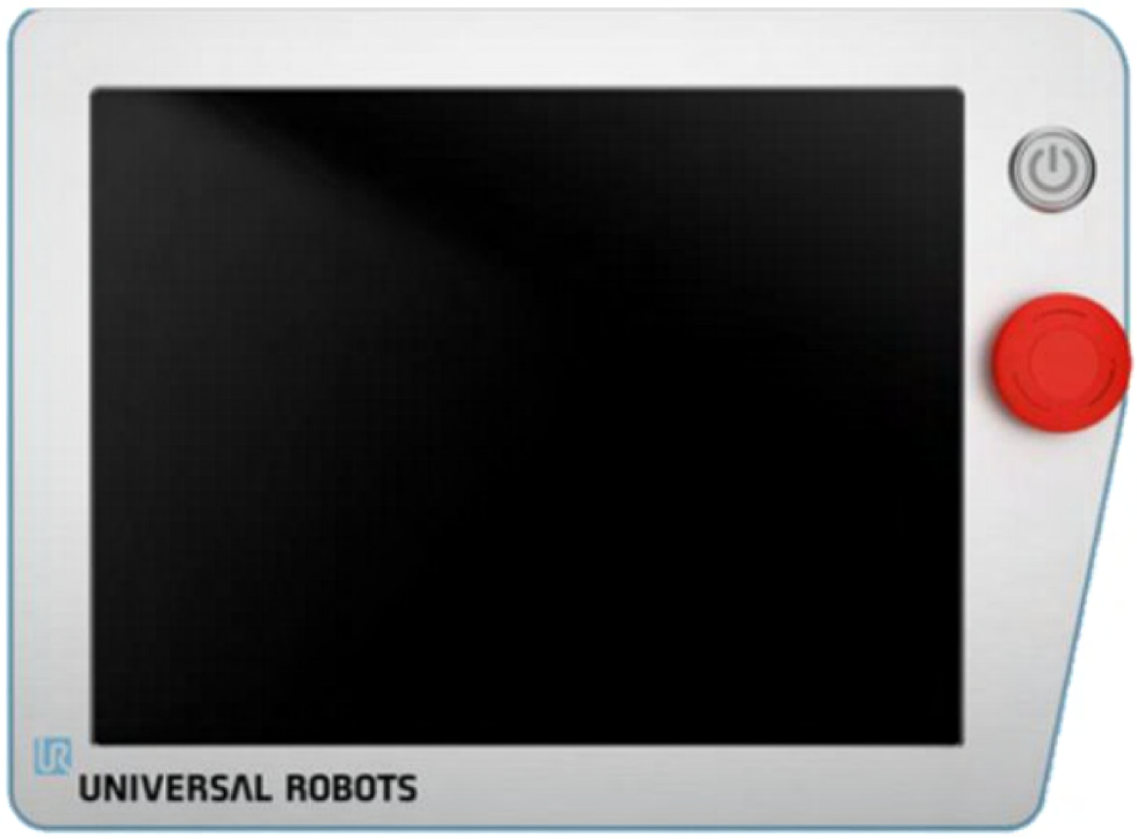


Figura 4.6: Pantalla UR5 [21].

Para iniciar el robot se debe hacer clic a configurar robot como se observa en la figura 4.7, luego se abre una pantalla como se observa en la figura 4.8 y hacer clic en Iniciar Robot y pasará a la pantalla como tal de encendido como se observa en la figura 4.9 y hacer clic en encender y listo para mover el robot. Además, existe una opción para una carga activa la cual indica la carga de la herramienta, esto permite al robot saber cuanto esta moviendo y no activar la parada de emergencia a encontrar una perturbación en la fuerza ejercida por la masa de la herramienta.

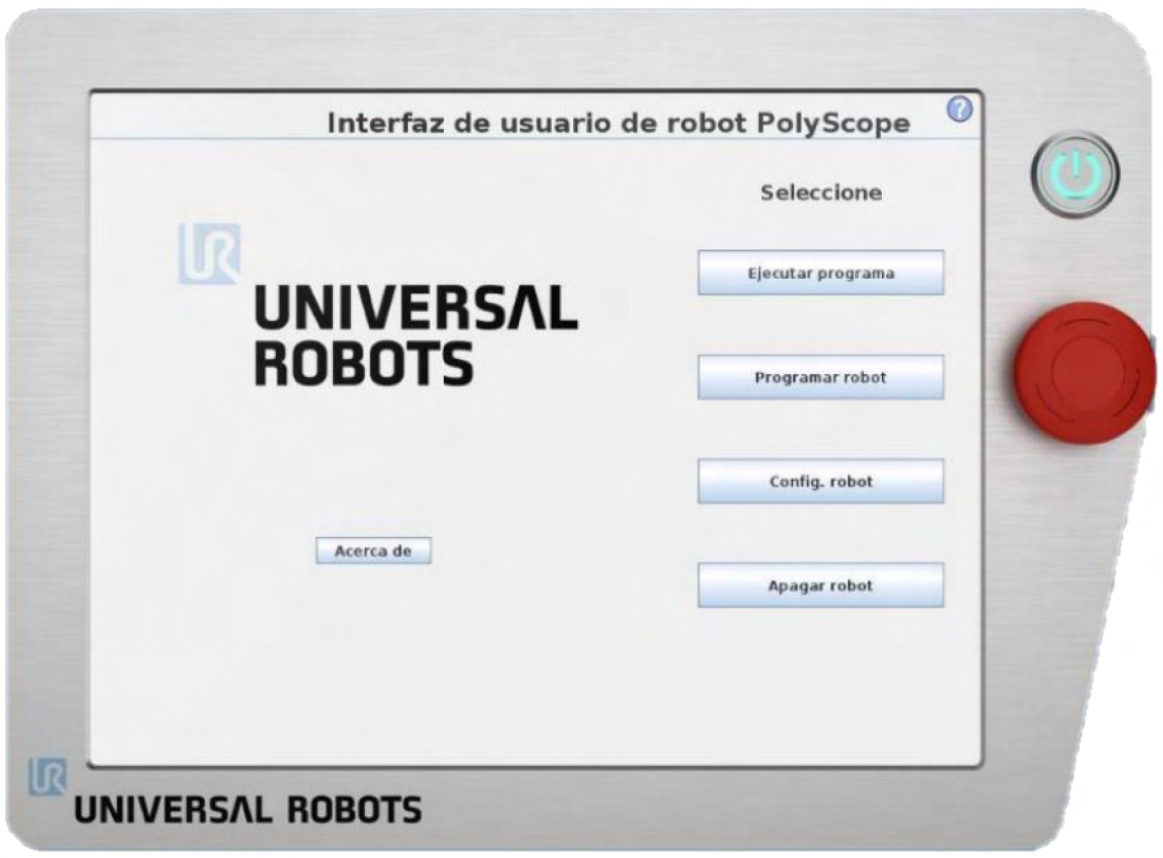


Figura 4.7: Pantalla inicial encedida UR5 [21].



Figura 4.8: Configurar UR5 [21].



Figura 4.9: Encendido/Apagado UR5 [21].

4.1.7 PCH o TCP

El TCP o PCH es el Punto Central de la Herramienta como se observa en la figura 4.10. Este punto ayuda al brazo a saber dónde se encuentra la base de la herramienta y a través de las coordenadas del PCH puede moverse de tal manera que la punta del extruder puede estar dirigido en todo momento como por ejemplo en la impresión 3D la salida del plástico fundido estar en contacto con el suelo.

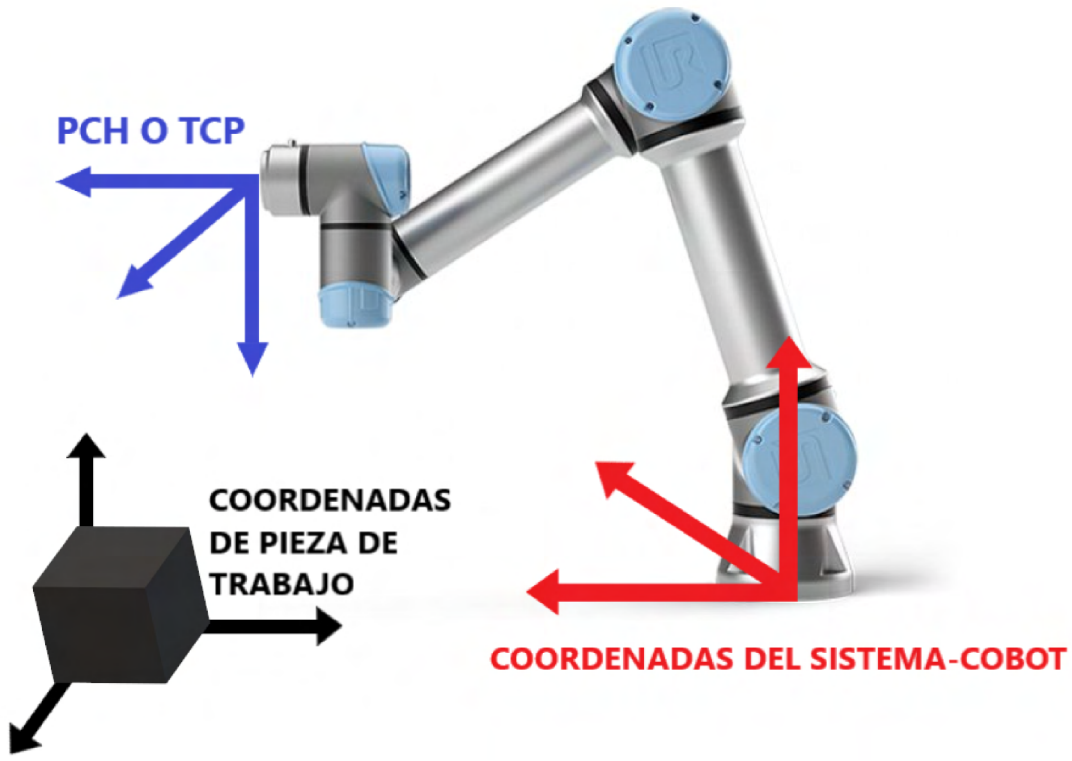


Figura 4.10: PCH o TCP (Creación Propia).

Para configurarlo en el robot se debe ir a Instalación como se muestra en la figura 4.11 y hacer clic en configuración de PCH. Una vez ahí en la parte derecha de la pantalla se muestra gráficamente el PCH donde el origen se encuentra en la parte central de la muñeca 3. Para configurarlo se debe cliquear en Posición y pedirá cuatro puntos como se muestra en la figura 4.12. Aquí se debe mover el brazo de tal manera que la punta de la herramienta llegue a un mismo punto, pero en diferentes posiciones (4 posiciones), para mover el brazo fácilmente y de manera rápida en la pantalla en la parte trasera tiene un botón el cual permite mover el robot libremente, en donde libera los frenos para guiar al robot con las manos.



Figura 4.11: Configuración de TCP (Creación Propia).



Figura 4.12: 4 puntos para TCP (Creación Propia).

En la figura 4.13 se muestra el punto que se seleccionó para obtener el TCP el cual es una punta metálica fija en una mesa. Para este ejemplo se utilizo un extruder de menor tamaño para realizar los movimientos de manera rápida y segura.



Figura 4.13: Configuración de TCP (Creación Propia).

En la figura 4.14 se muestran las diferentes posiciones que se realizo con el brazo UR5 para que el mismo cobot calculara de manera automática el TCP exacto, es decir que el punto central de la herramienta se encuentra en la punta del extruder, por donde saldría el plástico fundido.

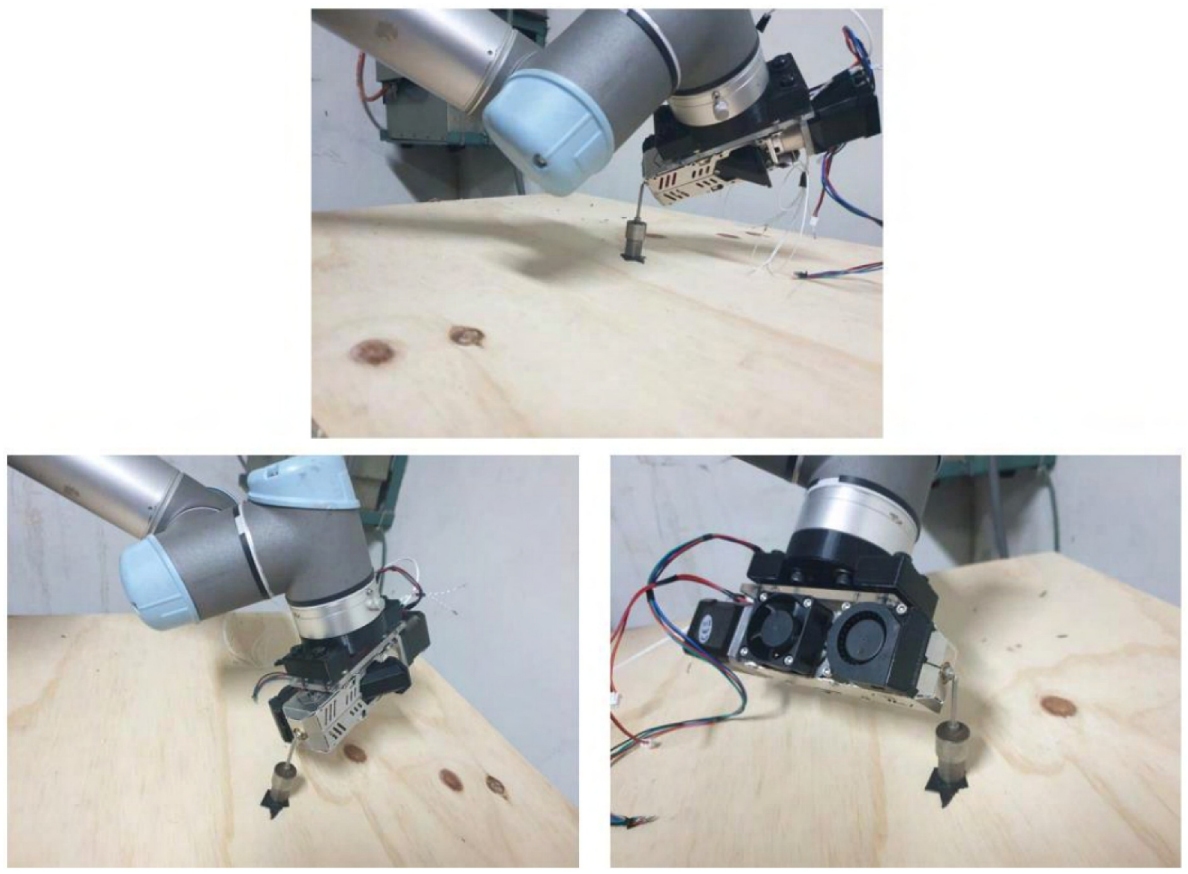


Figura 4.14: Posiciones brazo UR5 para TCP (Creación Propia).

Una vez hechos los cuatro puntos nos aseguramos de que estos se encuentren con una luz verde, lo

cual indica que las posiciones coinciden en un punto y que el PCH está calculado como se observa en la figura 4.15.

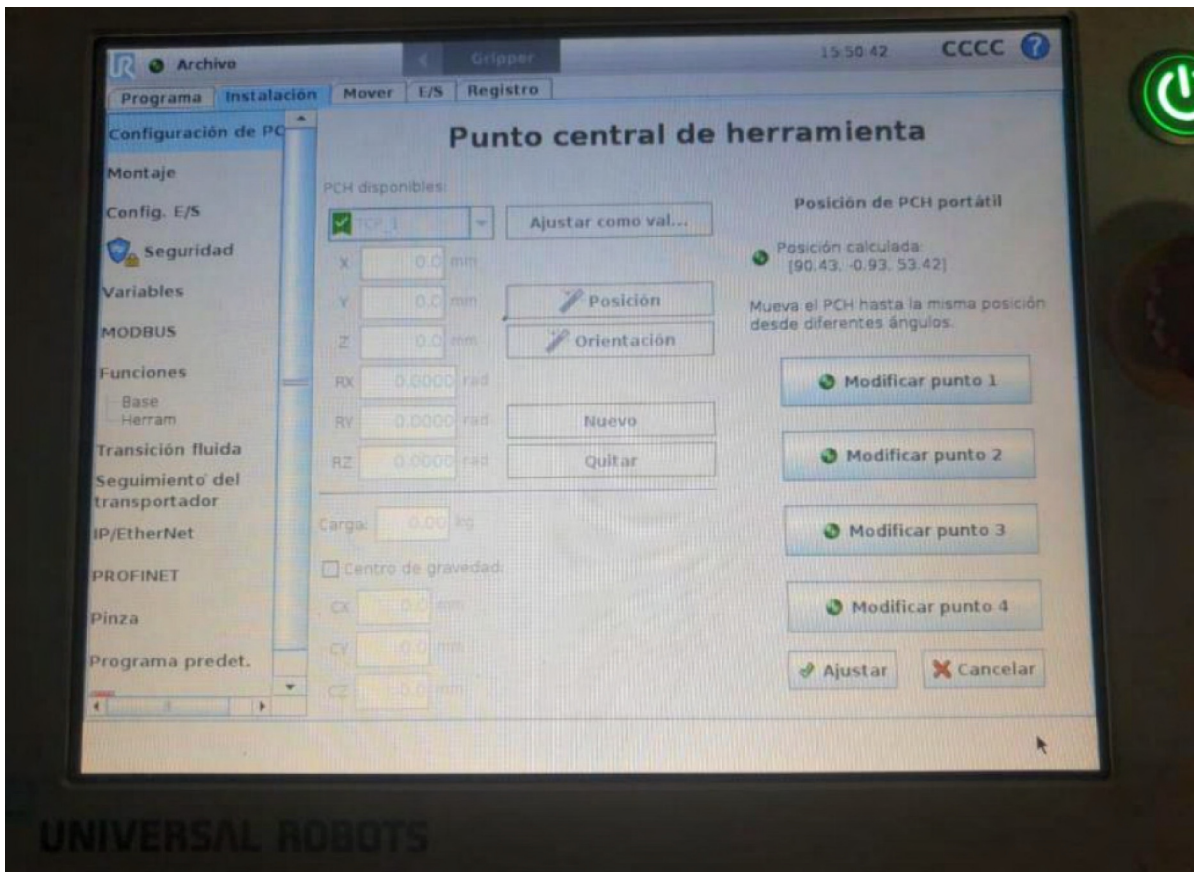


Figura 4.15: Puntos tomados por UR5 para la obtención de TCP (Creación Propia).

4.1.8 Mesa de Trabajo

Para crear una mesa de trabajo se debe crear un plano el cual el robot reconocerá como una sección que no debe traspasar. Para esto se debe dirigir a funciones (ver figura 4.16) y se mostrará 3 funciones en donde se debe cliquear en plano. Luego aparece una ventana con pasos en donde se muestra el plano y sus ejes con 3 puntos (ver figura 4.17). El cual el primero indica el origen, el segundo el eje X positivo y el tercero el eje Y positivo.

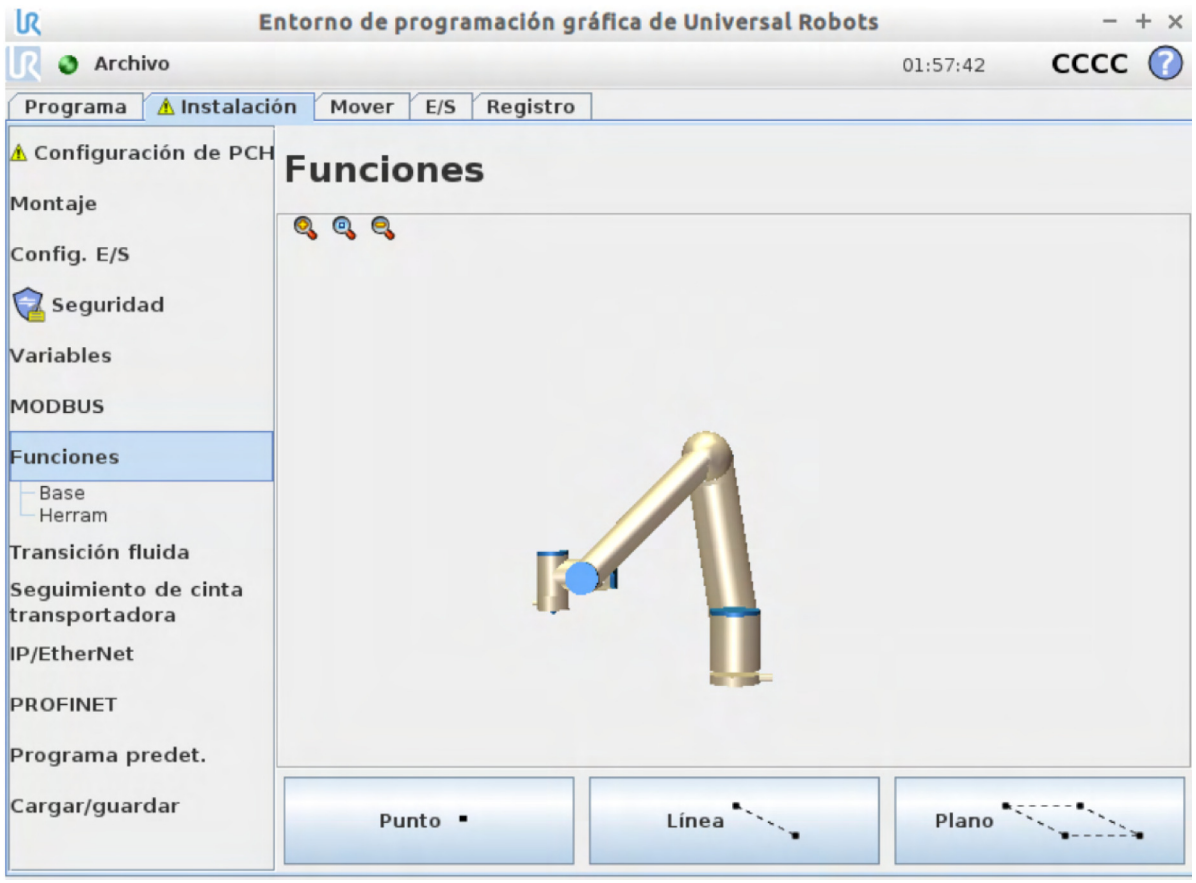


Figura 4.16: Plano virtual (Creación Propia).

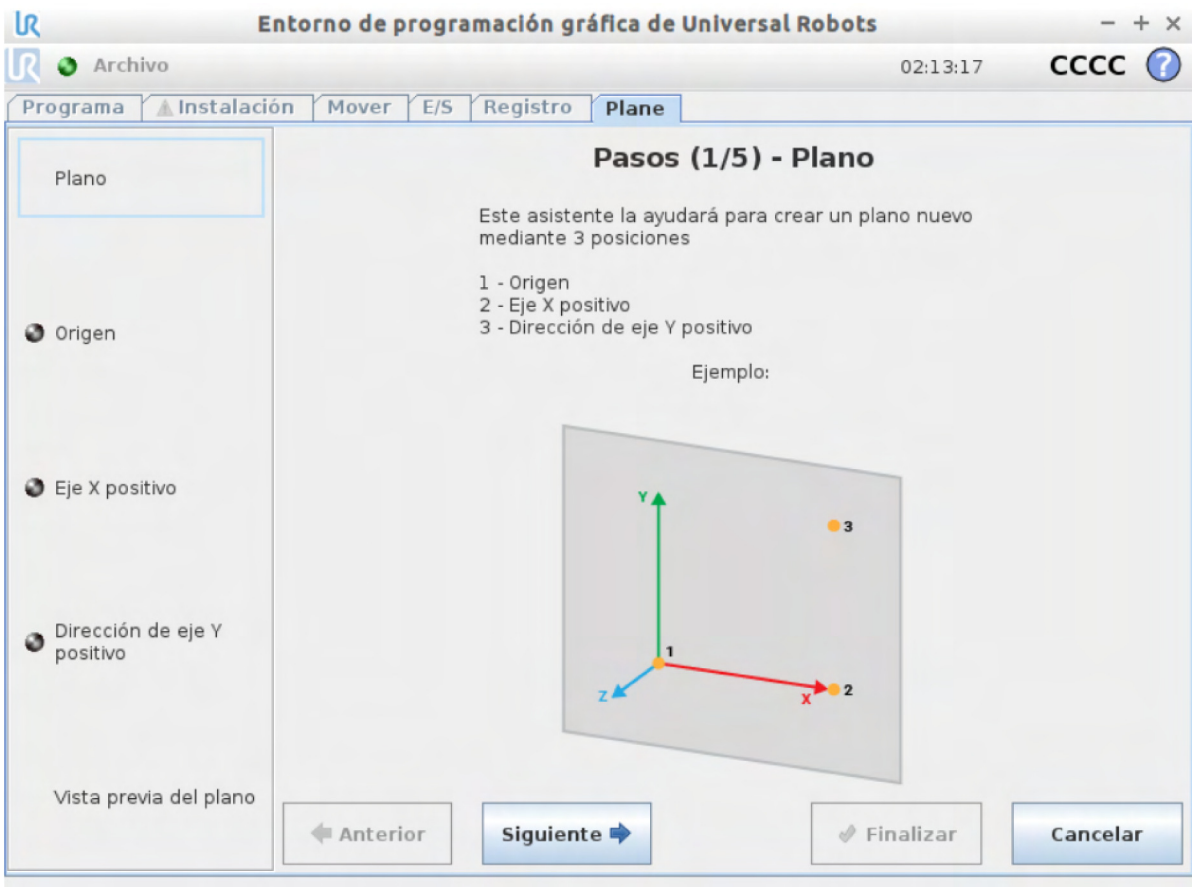


Figura 4.17: Pasos plano virtual (Creación Propia).

En la figura 4.18 se muestra la mesa de trabajo real, para simular el plano virtual con el cual se le comunicará al cobot la posición espacial del plano de trabajo. Donde 1 es el origen de las coordenadas del plano, 2 un punto en el eje X y 3 es un punto en el plano desplazado en Y.



Figura 4.18: Pasos plano real (Creación Propia).

En la figura 4.19 se muestra al UR5 en los puntos seleccionados en la figura 4.18. Así entonces el software incorporado en el cobot calculará el plano que contiene la mesa de trabajo.



Figura 4.19: La herramienta en los puntos 2 y 3 (Creación Propia).

Una vez identificados los 3 puntos físicamente por el robot, pulsar finalizar y notar que existe una mesa virtual en el gráfico de la pantalla como muestra en la figura 4.20. El plano de la mesa se mostrará en la sección de funciones con el nombre que se quiera renombrar.

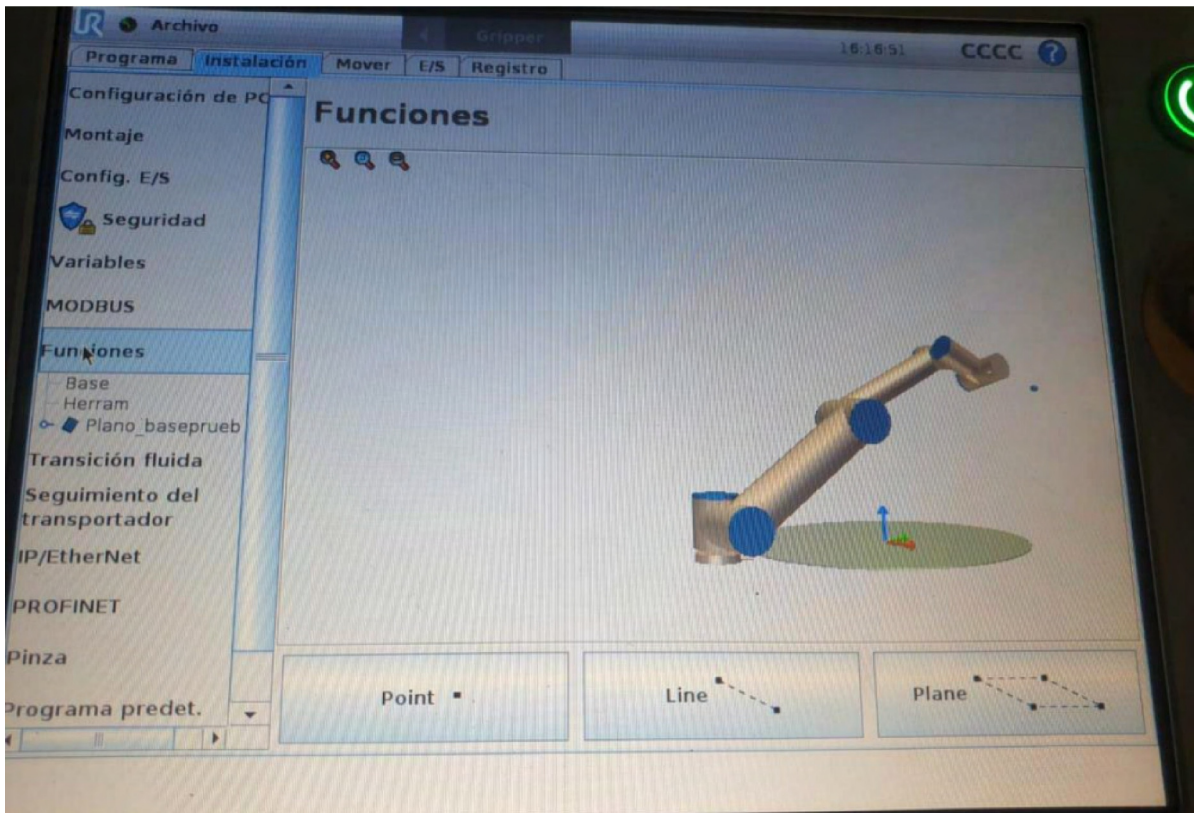


Figura 4.20: Plano creado por UR5 (Creación Propia).

4.1.9 Creación de programa básico en UR5

El robot permite crear programas a través de códigos que tiene el mismo. Para crearlo se hace clic en Programa y luego en programa vacío o programa nuevo como se muestra en la figura 4.21, se abrirá la pestaña (ver figura 4.22) en donde se muestra en la parte izquierda los códigos que se van a requerir y en la parte derecha las opciones de los comandos.



Figura 4.21: Programa nuevo (Creación Propia).

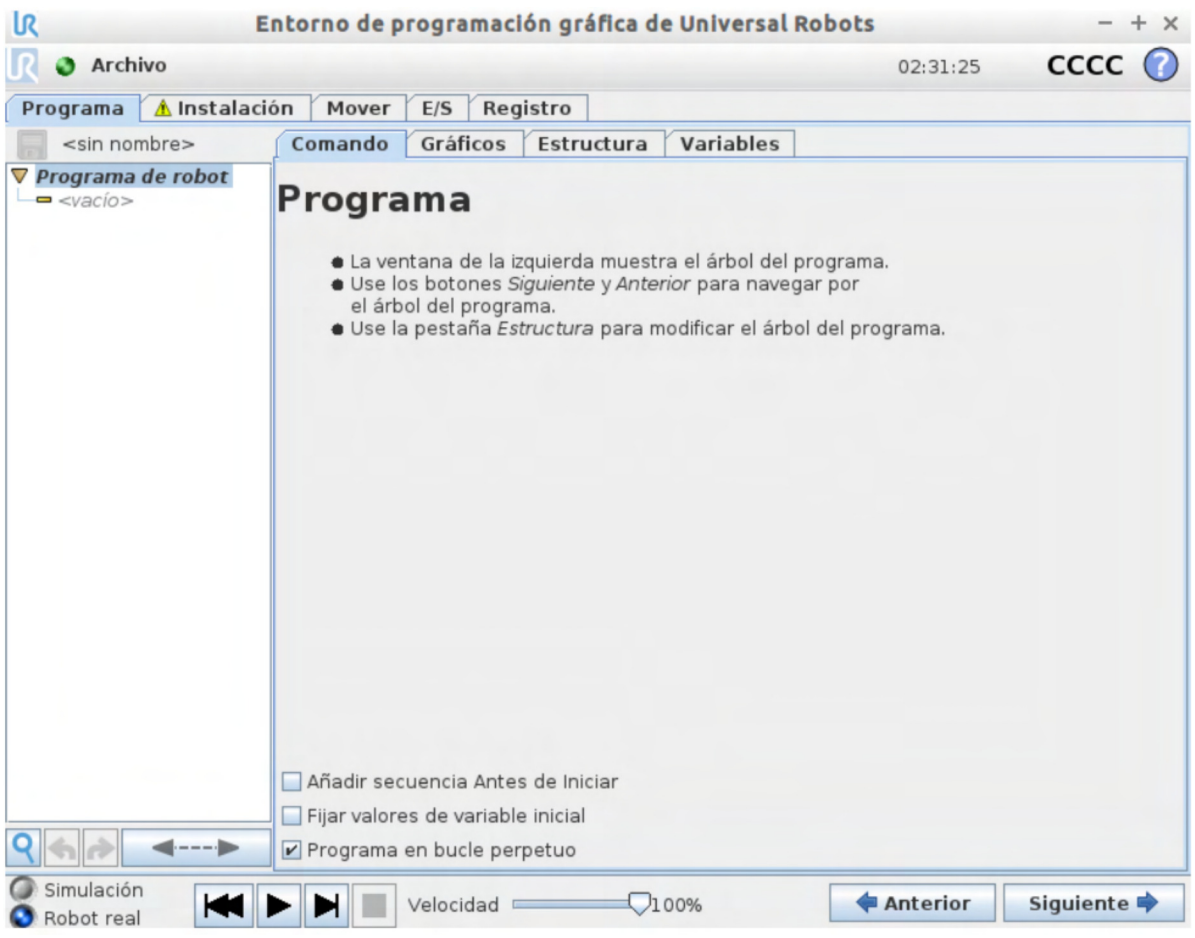


Figura 4.22: Pestaña programa nuevo (Creación Propia).

Antes de realizar el código se debe tener en cuenta que el UR5 se mueve con 3 tipos de movimientos como se muestra en la figura 4.23. Estos movimientos sirven para distintos trabajos que se requieran realizar como mover objetos, soldar o combinación de ambas, etc. Para el ejemplo se utilizará

MoveL.

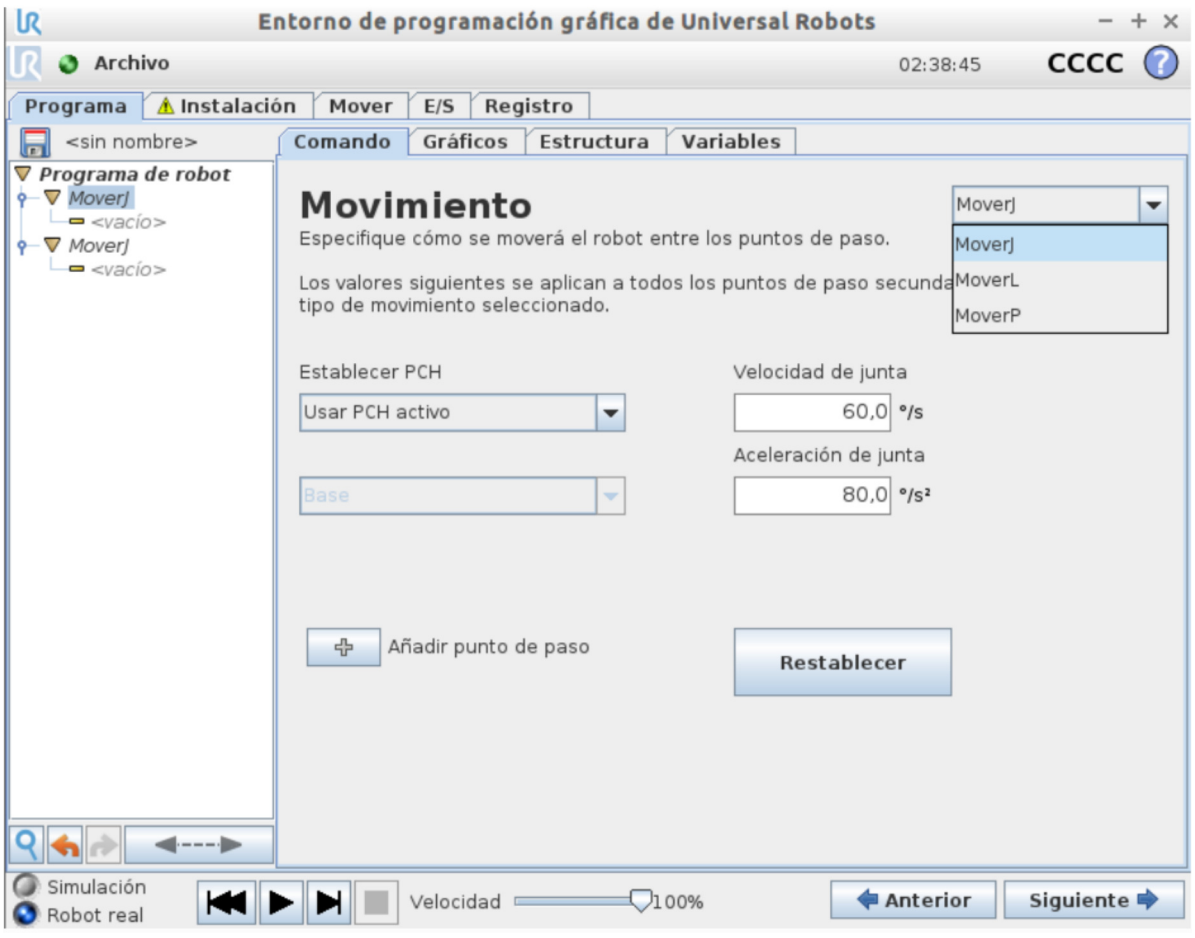


Figura 4.23: Tipos de movimientos (Creación Propia).

Tipos de movimientos:

- MoveJ: Es un movimiento no lineal cuando la ruta del trayecto del brazo no es de importancia, donde cada junta articulada se controla para llegar al mismo tiempo a la posición deseada con un movimiento de trayectoria curva de la herramienta. Los parámetros compartidos que se aplican a Movej son la velocidad de la junta y aceleración de la junta máximas utilizadas, especificadas en *grados/s* y *grados/s*^{2},respectivamente [21].

- MoveL: Es para trayectorias rectas (para recoger, colocar, etc), es decir movimientos precisos. Lo que significa que cada junta realiza un movimiento más complicado para mantener la herramienta en una trayectoria recta. Los parámetros compartidos que se pueden configurar para MoveL son la velocidad de la herramienta y la aceleración de la herramienta, especificadas en *mm/s* y *mm/s*^{2}, respectivamente [21].

- MoveP: Movimiento de proceso a velocidad constante con transiciones circulares. Ideal para soldaduras o para aplicar pegamento, debido a que permite una distribución homogénea. El tamaño del radio de transición esta predeterminado con un valor compartido entre todos los puntos de paso, en el cual un valor mas pequeño resultará en un movimiento brusco en la

trayectoria mientras que al contrario un valor mayor resultará en un movimiento más suave en la trayectoria [21].

La curva se divide en tres segmentos: aceleración, crucero y desaceleración como se muestra en la figura 4.24. El nivel de la fase crucero se obtiene del ajuste de velocidad del movimiento,mientras que la pendiente de las fases de aceleración y desaceleración se obtiene del parámetro de aceleración [21]

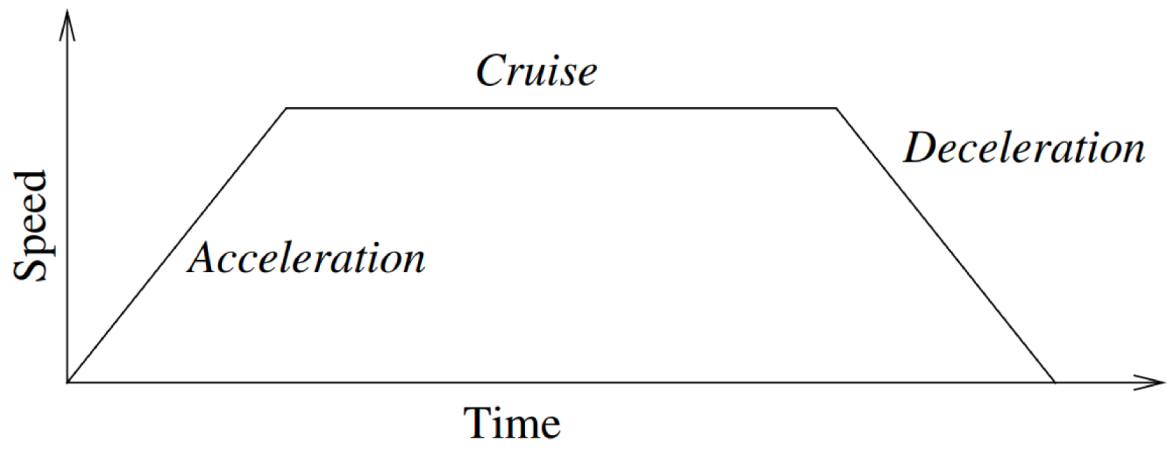


Figura 4.24: Perfil de velocidad [21].

Asi entonces se puede realizar un programa básico, donde el movimiento será con MoveL, para mantener la herramienta en una trayectoria recta. Para esto se realiza un programa básico, se cliquea en Estructura y aparece una pestaña donde se observa el editor de estructura de programa, en donde se muestra los comandos básicos (ver figura 4.25). Luego se dirige a mover que es para indicar el movimiento del robot, el cual será en MoveL y luego a punto de pasos, aparecerá las opciones de esta última como se ve en la figura 4.26 para luego ir a ajustar punto de paso, esto indica donde se quiere que el robot se encuentre cuando el código esté en ese comando. Entonces una vez ahí se ve en pantalla como se muestra en la figura 4.27 y se mueve el robot a las coordenadas deseadas a través de los botones táctiles o con el botón físico que se encuentra atrás de la pantalla para liberar los frenos y moverlo libremente con las manos. Una vez en la posición deseada apretar en *'Ok'* y se guardará ese primer comando. Luego se dirige nuevamente a estructura y se cliquea *'wait'* o esperar como se observa en la figura 4.28 y se dirige a comandos en donde se puede configurar el tiempo que espera el robot antes de moverse al punto siguiente, para el ejemplo se escogió 2 segundos. Así entonces se crearon 2 puntos más, en donde el robot se moverá del punto 1 al punto 2 con una espera de 2 segundos y del punto 2 al punto 3 con una espera de 1 segundo como se observa en la figura 4.29. Finalmente, se debe tocar en la parte inferior de la pantalla (ver figura 4.29) en donde aparece el símbolo de *'play'* para hacer correr el programa en el robot y además se puede variar el porcentaje de la velocidad máxima moviendo el variador de velocidad posicionado en la parte inferior central de la pantalla (ver figura 4.29), con esto se puede regular la

velocidad para que el robot no se traslade a una velocidad considerable entre los puntos deseados. El comando creado se realizará en bucle por lo que del punto 3 se moverá al punto 1 hasta que se indique que pare con el botón de *'play'*.

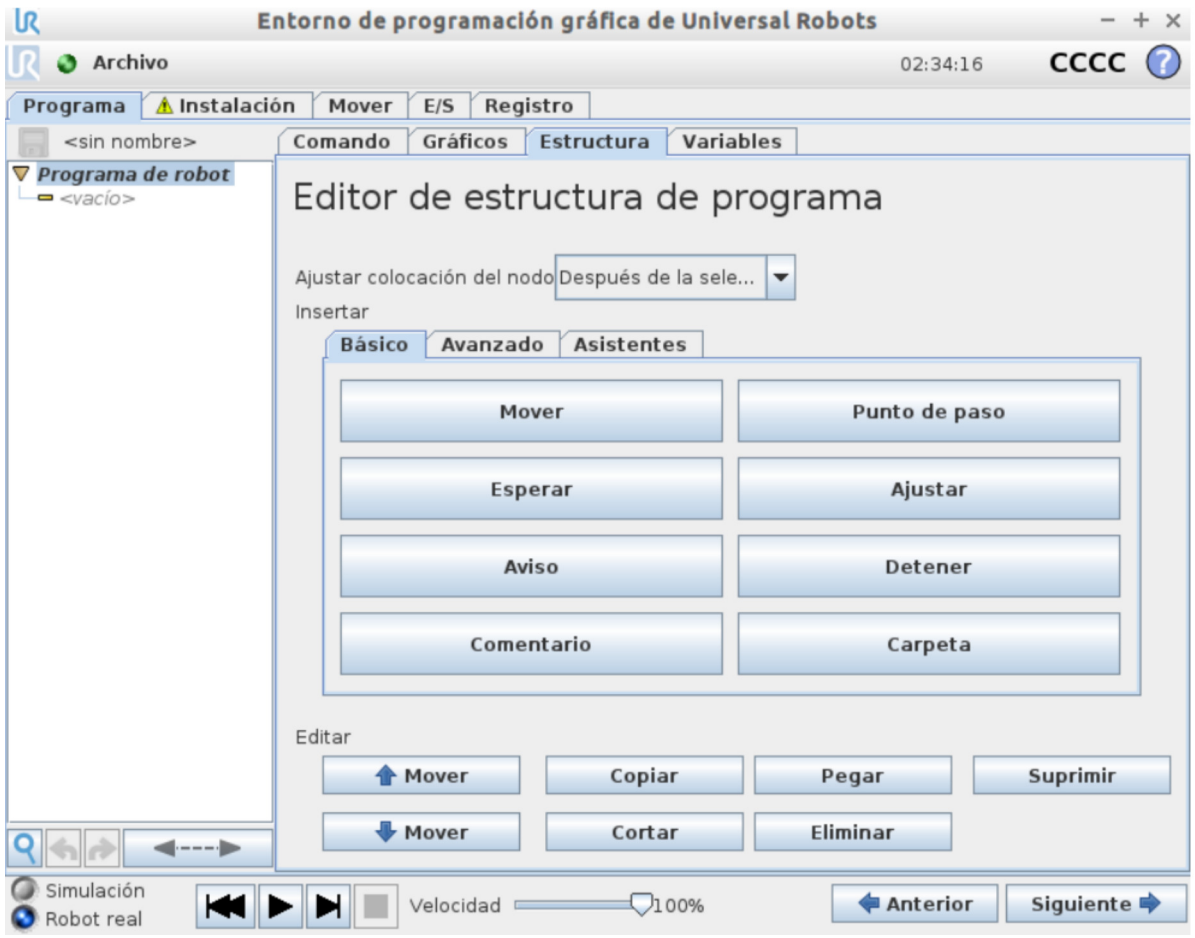


Figura 4.25: Editor de programa (Creación Propia).

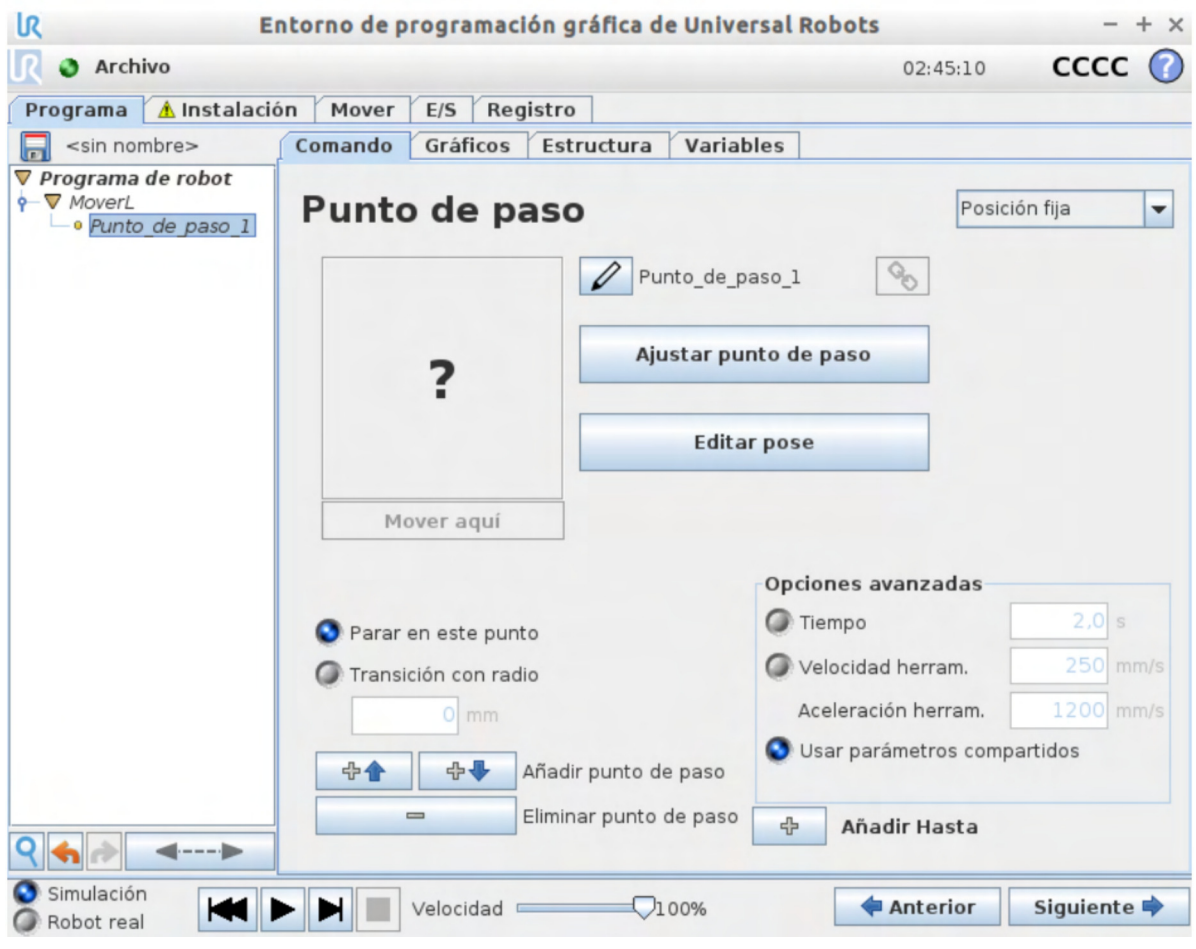


Figura 4.26: Punto de paso (Creación Propia).

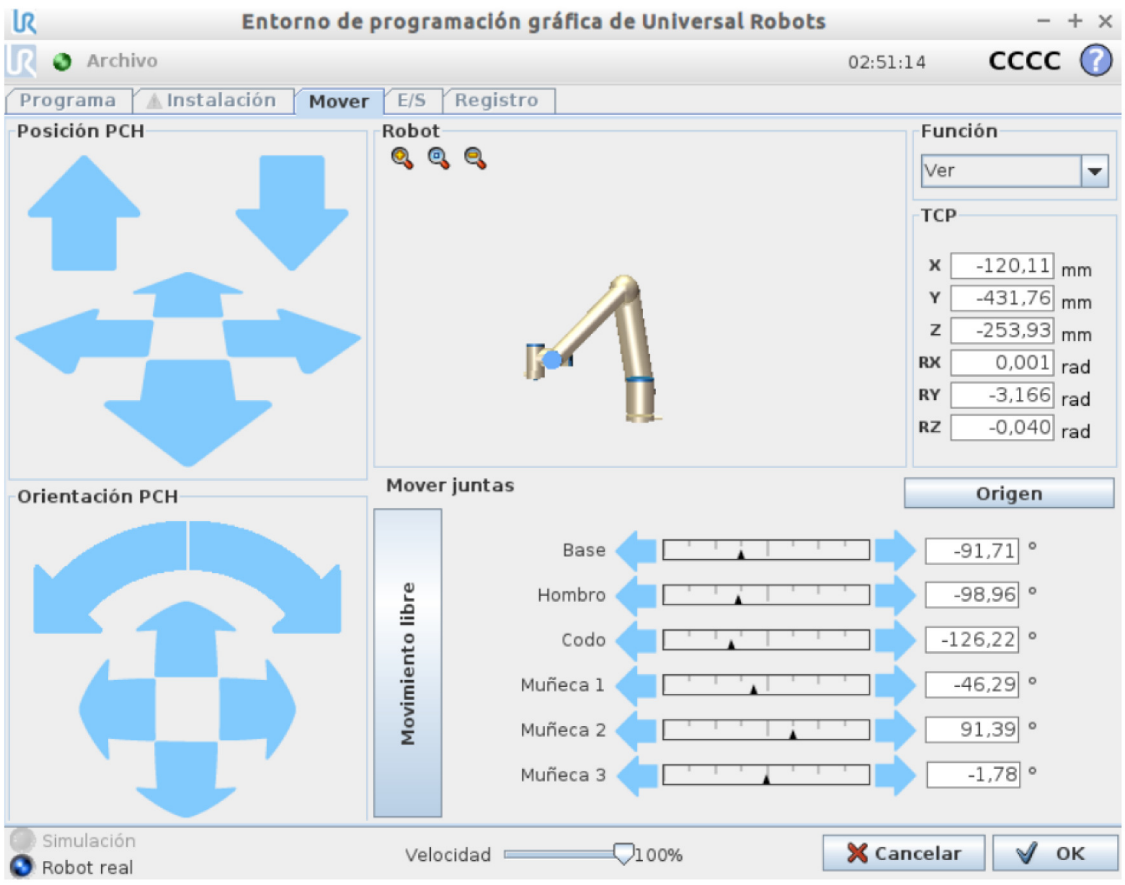


Figura 4.27: Posición Robot UR5 mostrado en el interfaz (Creación Propia).

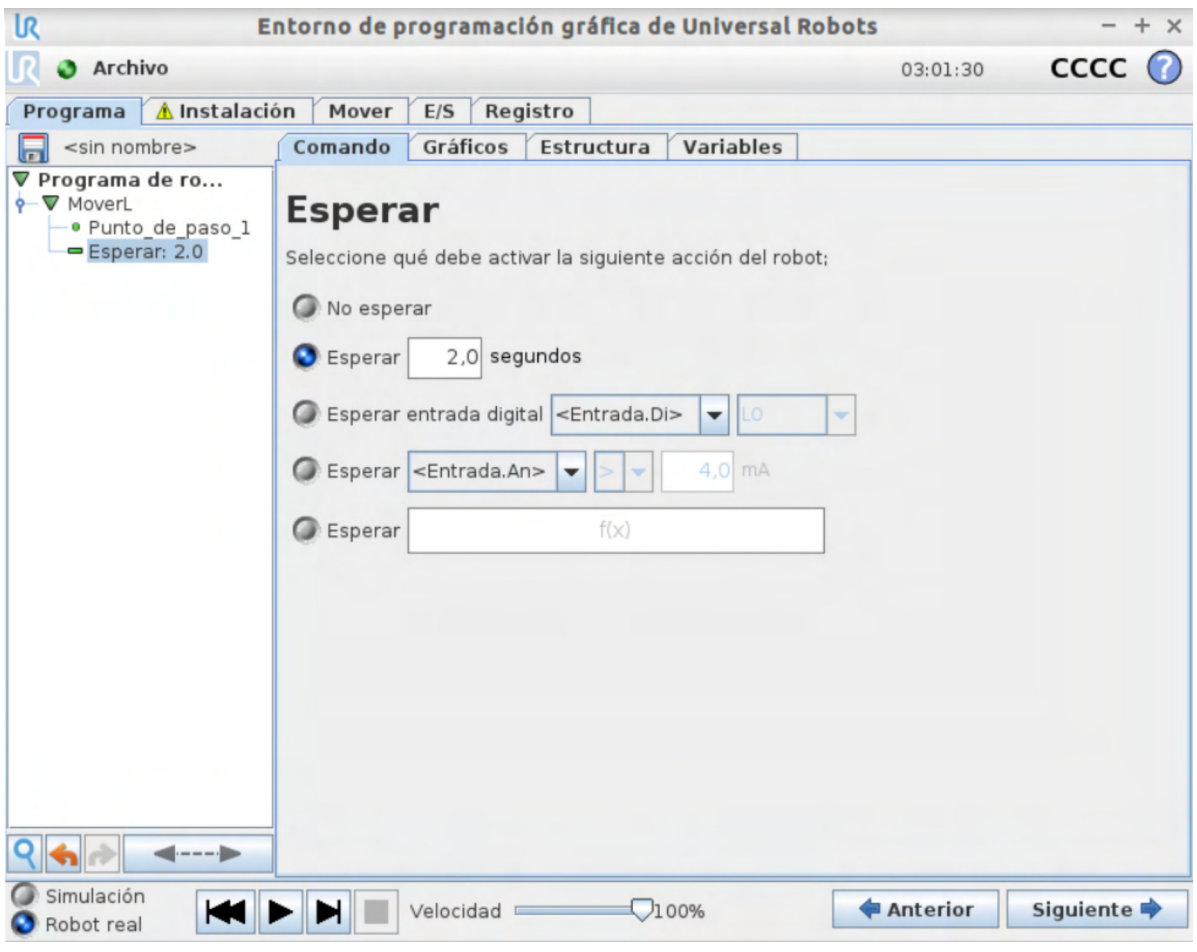


Figura 4.28: Wait o esperar (Creación Propia).

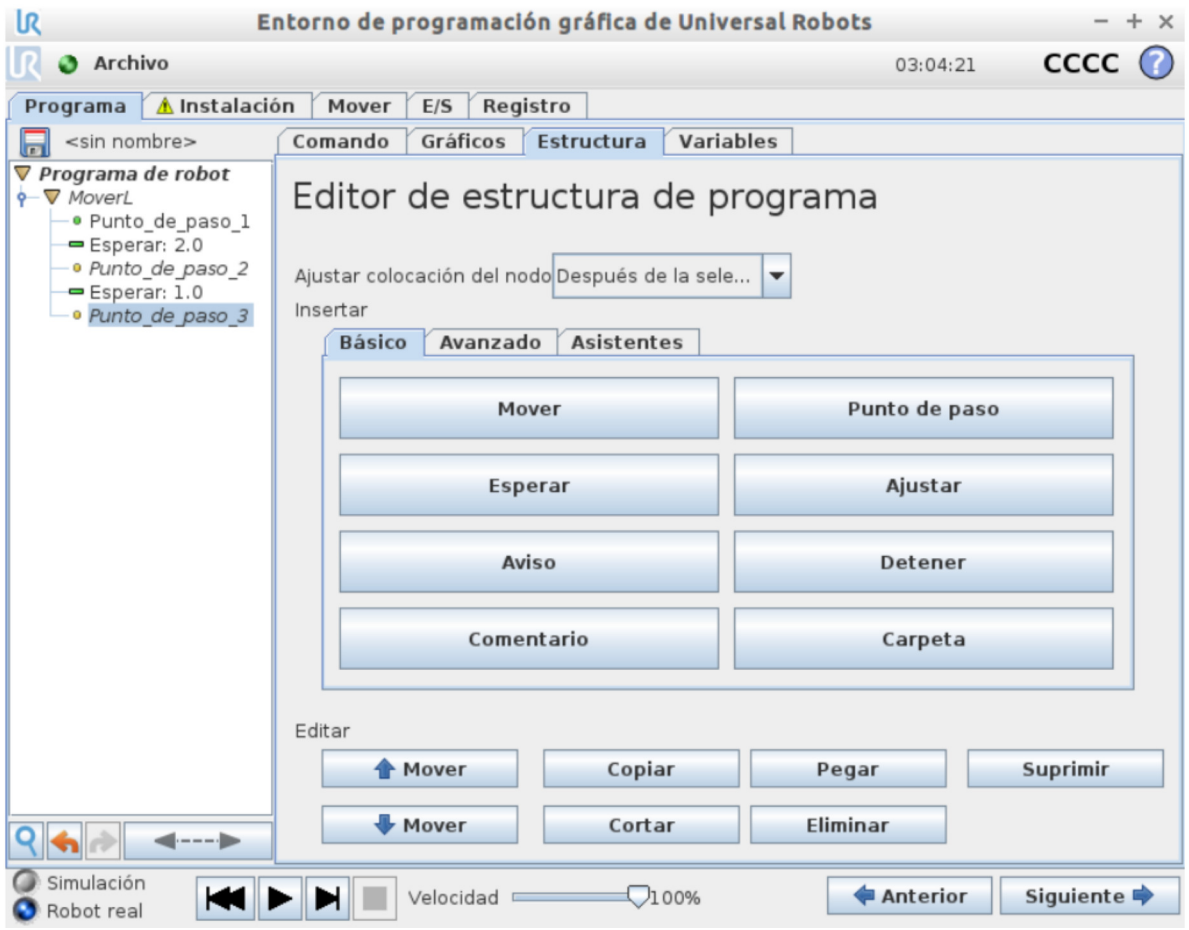


Figura 4.29: Programa básico final (Creación Propia).

4.1.10 Guardado

Tanto el PCH como el comando creado se puede guardar en el robot y así no realizarlo cada vez que se encienda. Para esto se debe apretar en Archivo y aparecerá opciones (ver figura 4.30), luego se toca en guardar como y aparecerá una ventana como se muestra en la figura 4.31, aquí se coloca el nombre deseado para el archivo y cliquear en guardar.

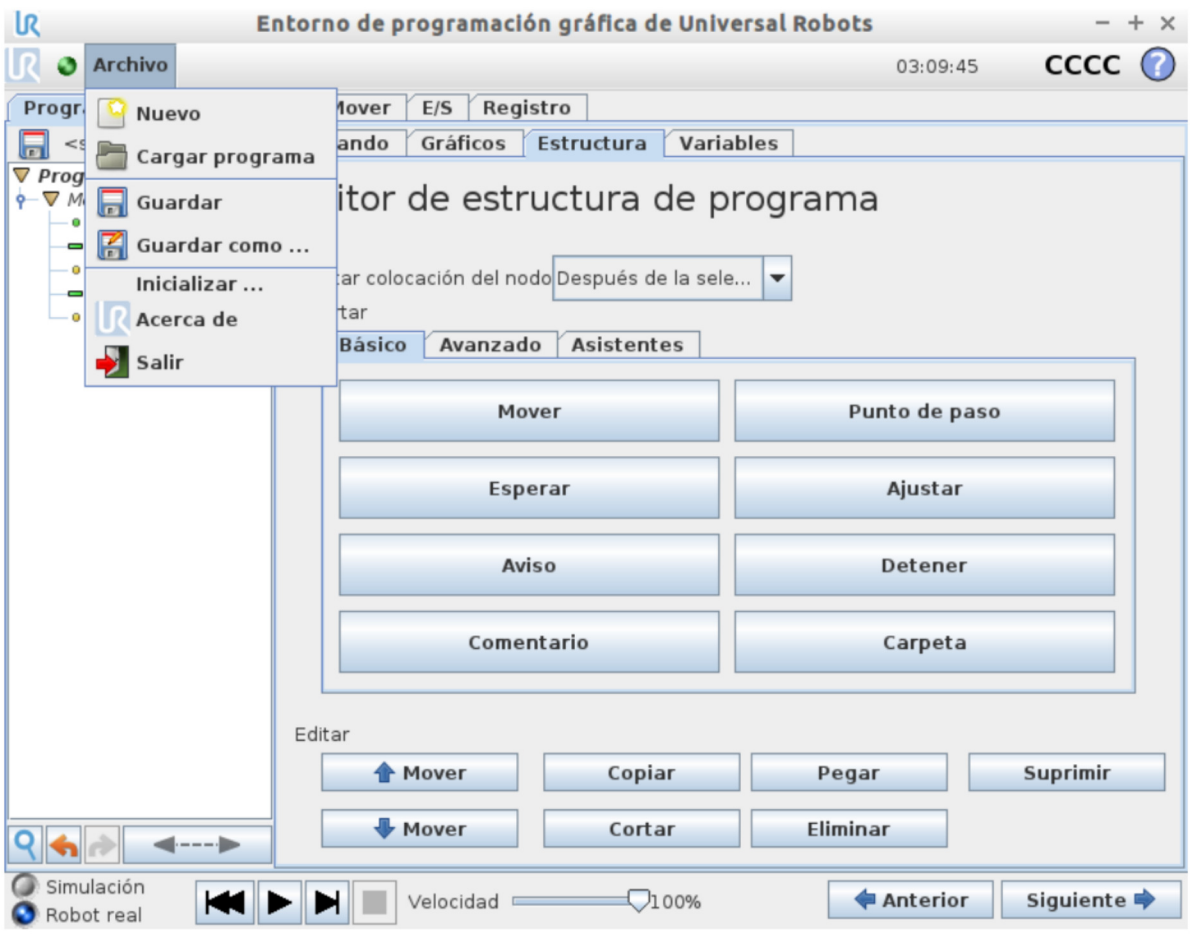


Figura 4.30: Guardar como (Creación Propia).

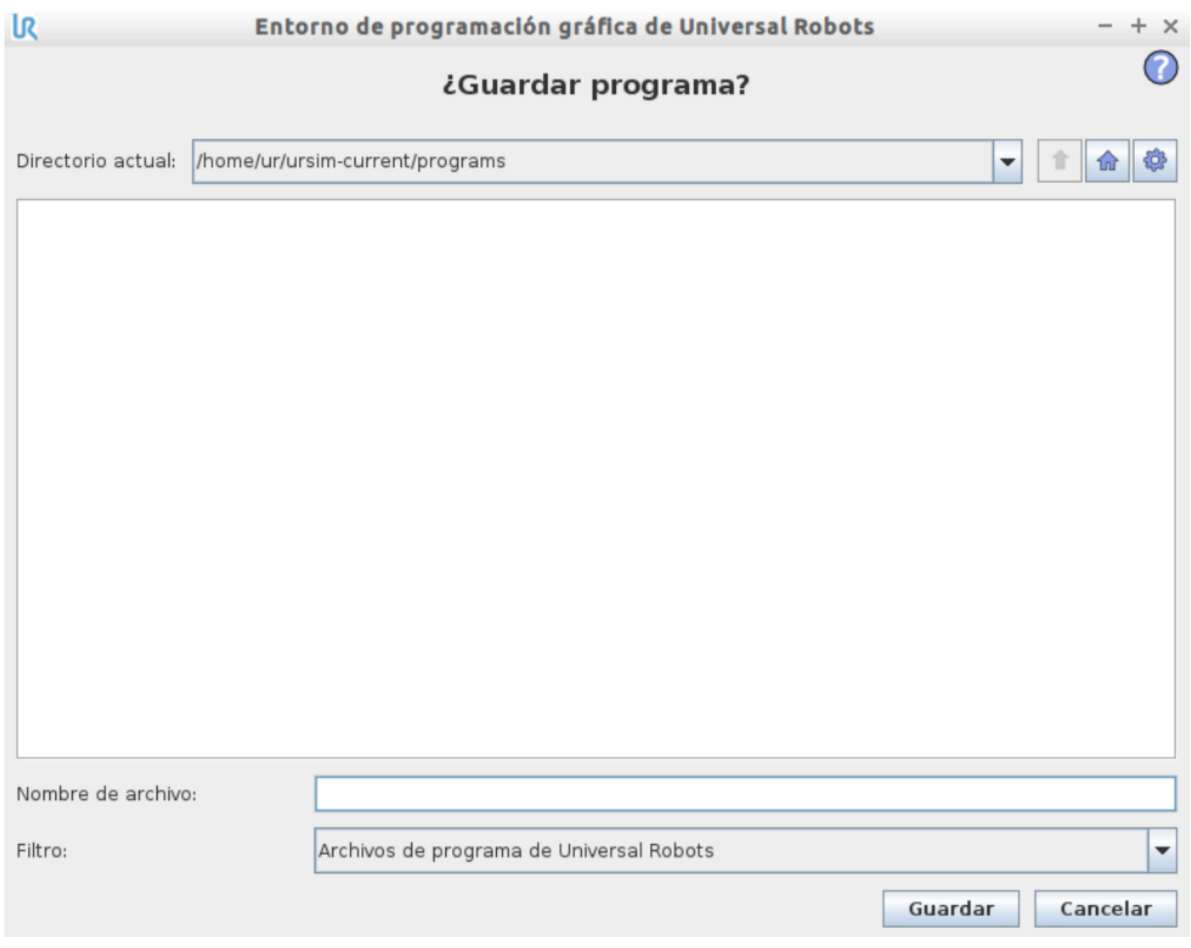


Figura 4.31: Guardar archivos (Creación Propia).

4.1.11 Conectar PC con UR5

Para conectar el robot al PC se necesita de un cable ethernet y el programa Filezilla, el cual permite entrar al robot y ver todos los archivos de UR5. Para realizar esta conexión se debe cambiar la red del robot, para esto se debe ir a configurar robot, luego Red y se selecciona la Dirección estática y se cambia la dirección IP a 172.16.17.10, la máscara de Red a 255.255.255.0 y puerta de enlace predeterminada a 0.0.0.0 y se finaliza dando click a aplicar como se observa en la figura 4.32.



Figura 4.32: Configurar Red (Creación Propia).

Luego en "Filezilla" se coloca el servidor que se aplico al robot que es 172.16.17.10, en nombre de usuario *'root'*, contraseña *'easybot'* y puerto 22. Así entonces, se da click a conectar y una vez conectado se debe observar como la figura 4.33.

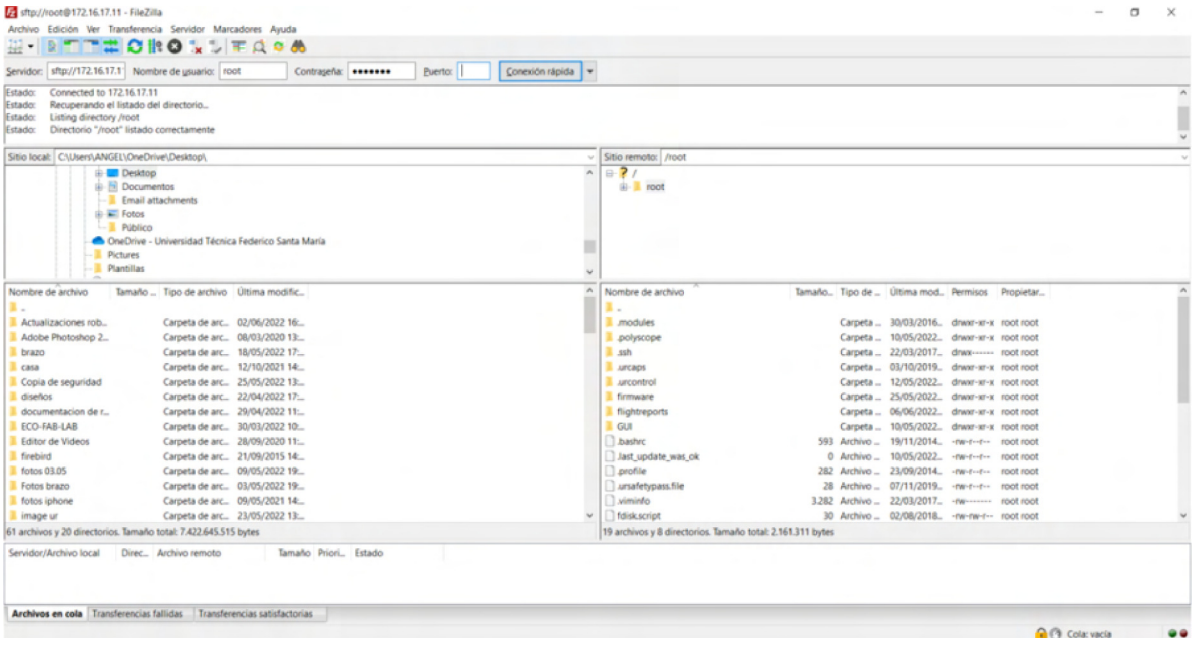


Figura 4.33: Conexión a UR5 desde PC con "Filezilla" (Creación Propia).

En la figura se observan todos los archivos que tiene integrado el cobot, esto permite cargar archivos de programas desde el pc al brazo robótico, guardar carpetas de importancia en el pc para tener una copia si se reinicia o actualiza el UR5 y poder integrarla nuevamente si se da el caso que se haya eliminado, etc.

4.2 RoboDK

Para lograr que el robot se mueva siguiendo una trayectoria deseada sin la necesidad de saber todos los puntos de la trayectoria y llegar hacer un programa como en la sección 4.1.9, es que existen *softwares* que trabajan con brazos robóticos y facilitan al usuario que pueda crear movimientos a partir de figuras deseadas para imprimir en 3d. Para esto se ocupa el software de "RoboDK" ocupado en la industria de manufactura robotizada, el cual requiere licencia para ocupar sus distintas funciones entre las cuales la impresión 3d. Para las pruebas se comunicó con el departamento de arquitectura que tienen en su poder licencias estudiantiles de "Robodk", las cuales se utilizó para primeramente realizar dibujos con el robot UR5 y obtener resultados de movimiento de éste respecto a distintas figuras antes de ocupar la impresión 3D.

4.2.1 Conexión de Robodk a UR5

Para conectar a "Robodk" al cobot primero se debe forzar la ip de la computadora, para esto se debe ir a Estado de red y luego a Cambiar opciones de adaptador y se abrirá una ventana como se observa en la figura 4.34.

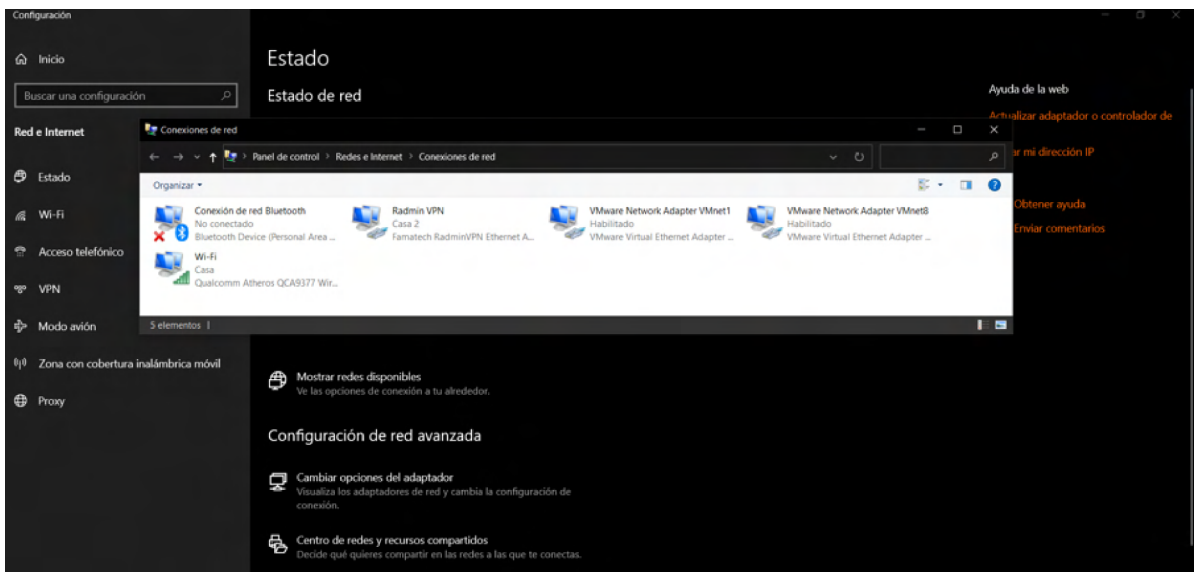


Figura 4.34: Cambiar opciones de adaptador (Creación Propia).

Luego ir a 'Radmin VPN' dar click derecho a propiedades, activar la opción protocolo de internet versión 4 e ir a propiedades como se muestra en la figura 4.35. En propiedades activar la opción de Usar la siguiente dirección IP y colocar lo siguiente:

- Dirección IP: 172.16.17.11 Esto permitirá que la computadora esté en el rango de dirección de IP del brazo UR5 definido en la sección 4.1.11 en configurar red del robot.
- Mascara: 255.255.255.0 Al igual que la sección anterior 4.1.11, debe estar en la misma máscara de subred.
- Puerta de enlace predeterminada: no colocar nada.

Una vez colocado estos valores como se muestra en la figura 4.36 se da click a aceptar y una vez se conecte el cable eterneth a la computadora tomará la IP descrita anteriormente lo que permitirá que tanto el brazo como el pc estén en la misma dirección.

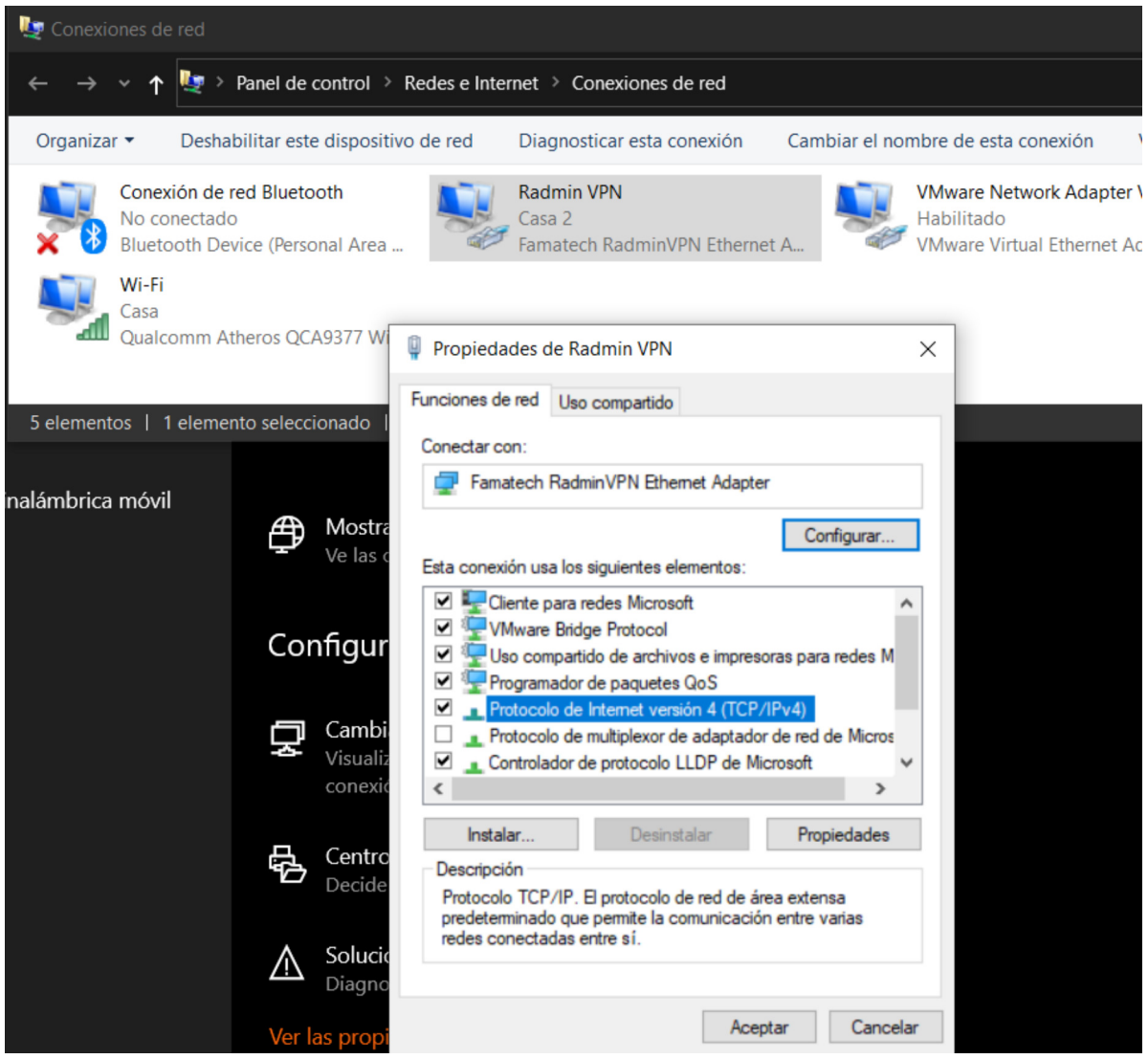


Figura 4.35: Activar IPv4 (Creación Propia).

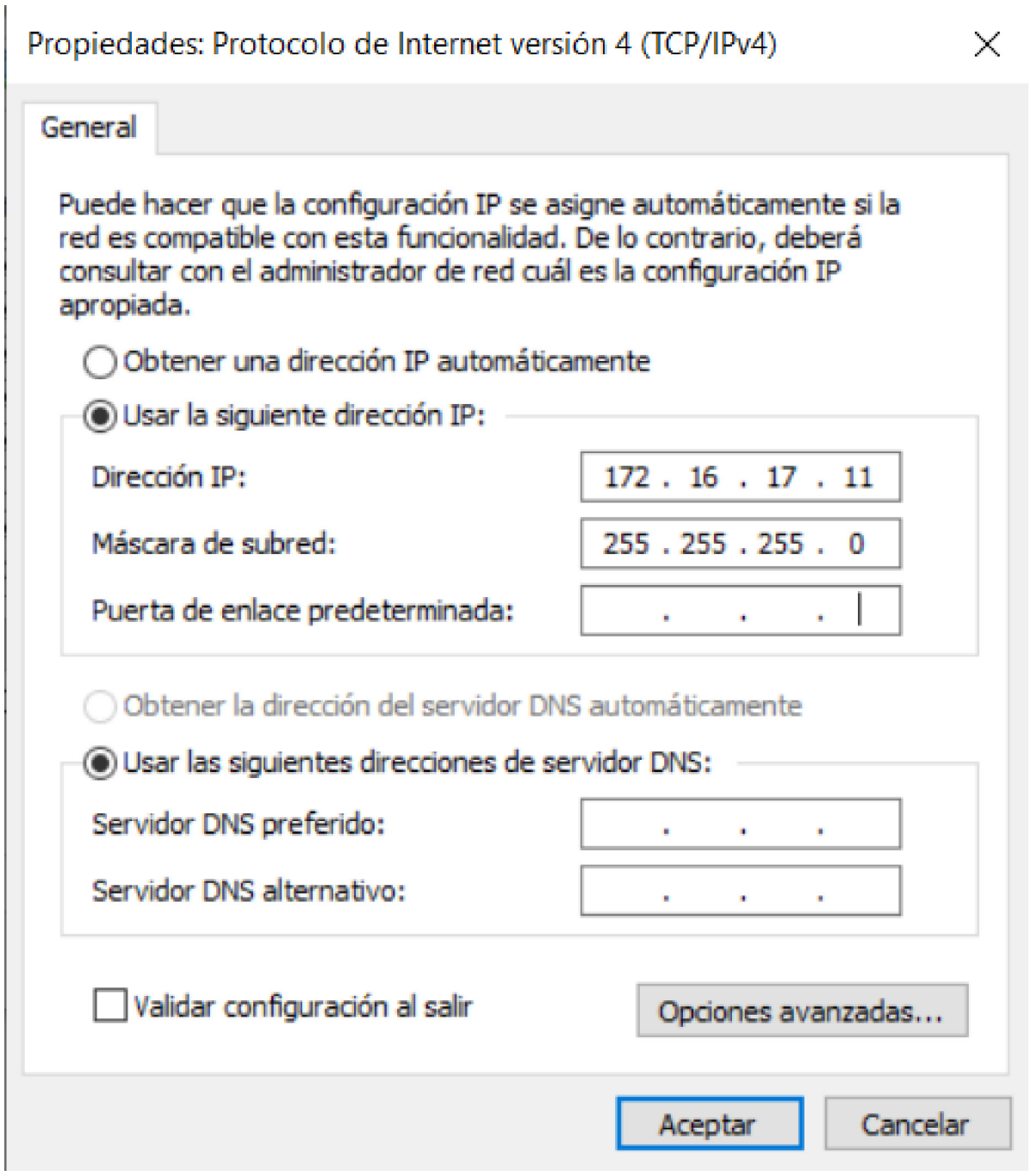


Figura 4.36: Activar IPv4 (Creación Propia).

Finalmente, para hacer la conexión del software "RoboDK" con el brazo UR5 se debe descargar el archivo que contiene las informaciones físicas del brazo robótico, el cual se obtiene de manera gratuita en la página oficial de "Robodk". Una vez descargado el archivo del brazo UR5 se abre el *software* y se hace click en archivo, luego en abrir, se busca el archivo descargado de "RoboDk" del cobot UR5 y se hace click en abrir, así entonces se mostrará el robot como se observa en la figura 4.37. Una vez mostrado el robot virtualmente se debe hacer click a conectar ubicada en las opciones en la parte superior del *sofwtare*, luego a conectar robot, se abre una ventana en la parte izquierda y se debe colocar la dirección que se le impuso a la computadora 172.16.17.11 y mantener el puerto por defecto. Se cliquea en conectar y debe aparecer *'Ready'* y color verde, lo cual indica que se conectó correctamente al brazo robótico como se muestra en la figura 4.38, por lo que el robot puede ser movido desde el pc. Notar que no se puede mover libremente el robot con el botón trasero que permite moverlo manualmente. Para esto se debe desconectar de "Robodk" haciendo doble click en desconectar.

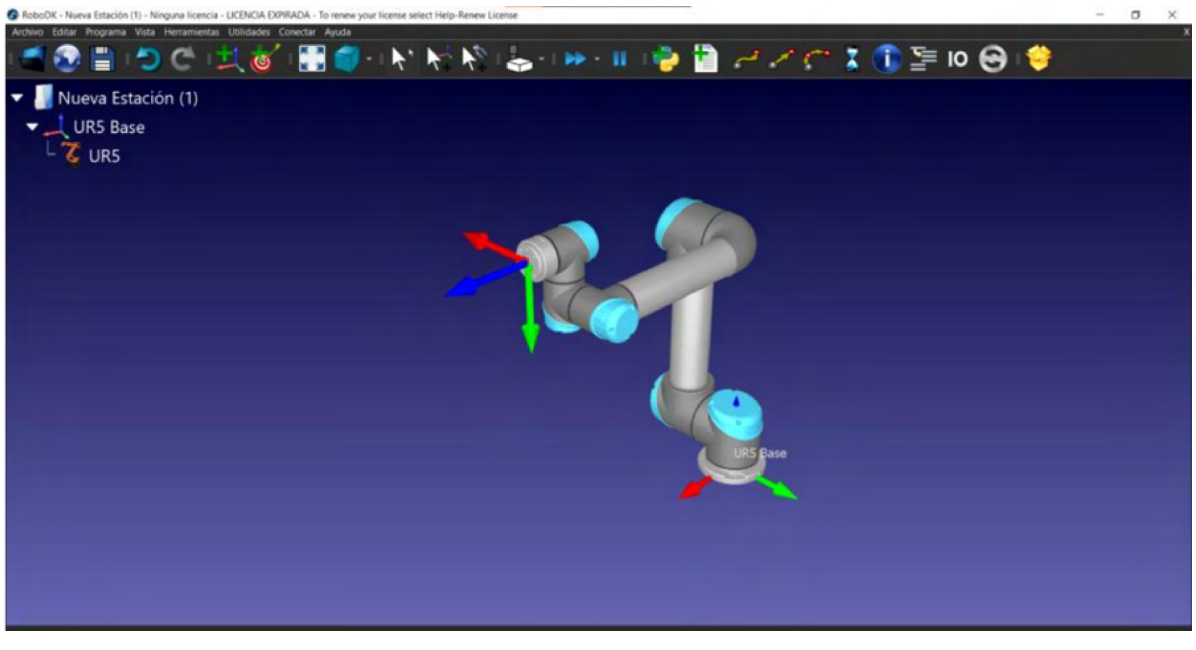


Figura 4.37: Interfaz RoboDK con archivo de UR5 (Creación Propia).

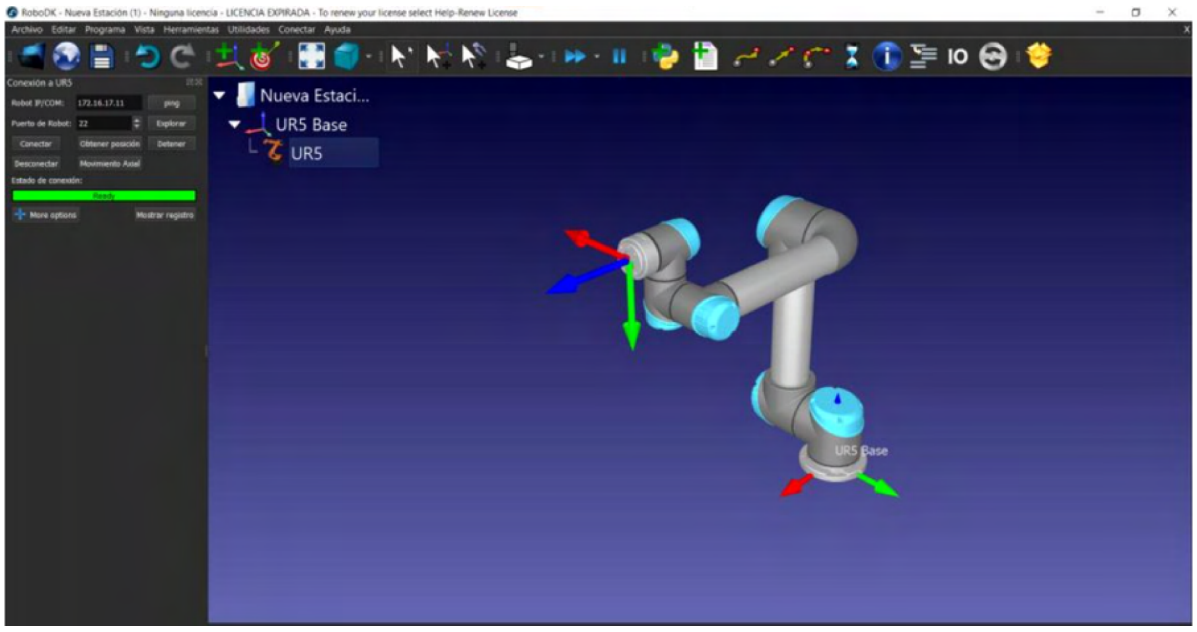


Figura 4.38: Conexión de RoboDK con el brazo UR5 (Creación Propia).

4.2.2 TCP o PCH en Robodk

Como se utilizará un plumón para dibujar se debe definir el PCH del mismo, para esto se debe cliquear con el botón derecho sobre donde dice UR5 y se cliquea en añadir herramienta (TCP). Luego doble clic en *'Tool 1'* y abre una ventana en la parte derecha de la pantalla como se muestra en la figura 4.39 en donde se coloca las coordenadas obtenidas por los pasos de encontrar el PCH con las herramientas del robot enseñadas en la sección 4.1.7. En la figura 4.40 se muestra una imagen del plumón puesto como herramienta en el brazo robótico .

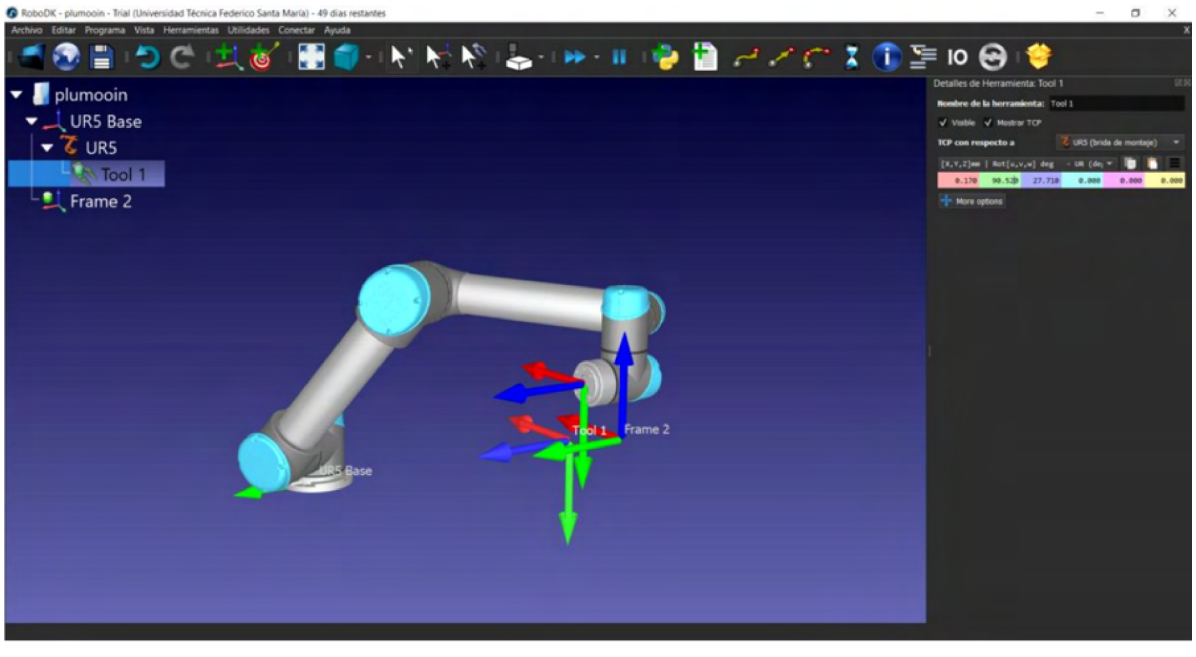


Figura 4.39: Coordenadas de TCP con plumón como herramienta (Creación Propia).



Figura 4.40: Plumón instalado en el cobot UR5 (Creación Propia).

4.2.3 Dibujo en RoboDK

Para poder dibujar con el robot una imagen cualquiera en formato principalmente '.jpg', el cual es un formato común en imágenes o archivos 2D, y además fácil de descargar de internet un dibujo simple o uno complejo. Es por esto que se necesita un programa que estudie la imagen y transforme las lineas del dibujo en formato CAD y luego con el *software* CAD revisar si existen imperfecciones y guardarlo en formato '.DXF' con el cual se puede trabajar para hacer el código G que es el código con el cuál trabajan los brazos robóticos e impresoras 3D como se explica en el capitulo 3 sección 3.3.4.

4.2.3.1 Inkspace

Para poder transformar una imagen '.jpg' a CAD se necesita el programa "INKSCAPE 1.2" como se muestra en la figura 4.41, luego se dirige a nuevo documento y se carga la imagen deseada como se muestra en la figura 4.42 se va a *'Update preview'* para ver como van quedando las lineas al momento de pasarla a CAD. Para que tome todas las lineas deseada se debe ir ajustando con los detalles que se muestran en la parte derecha, una vez tomadas todas como se me muestra en la figura 4.43 se dirige a archivo y se cliquea en guardar como, se coloca un nombre y el formato del archivo debe ser 'AutoCAD DFX R12' como se muestra en la figura 4.44. Así entonces se obtiene la imagen vectorizada de una imagen cualquiera de internet como se muestra en la figura 4.45.

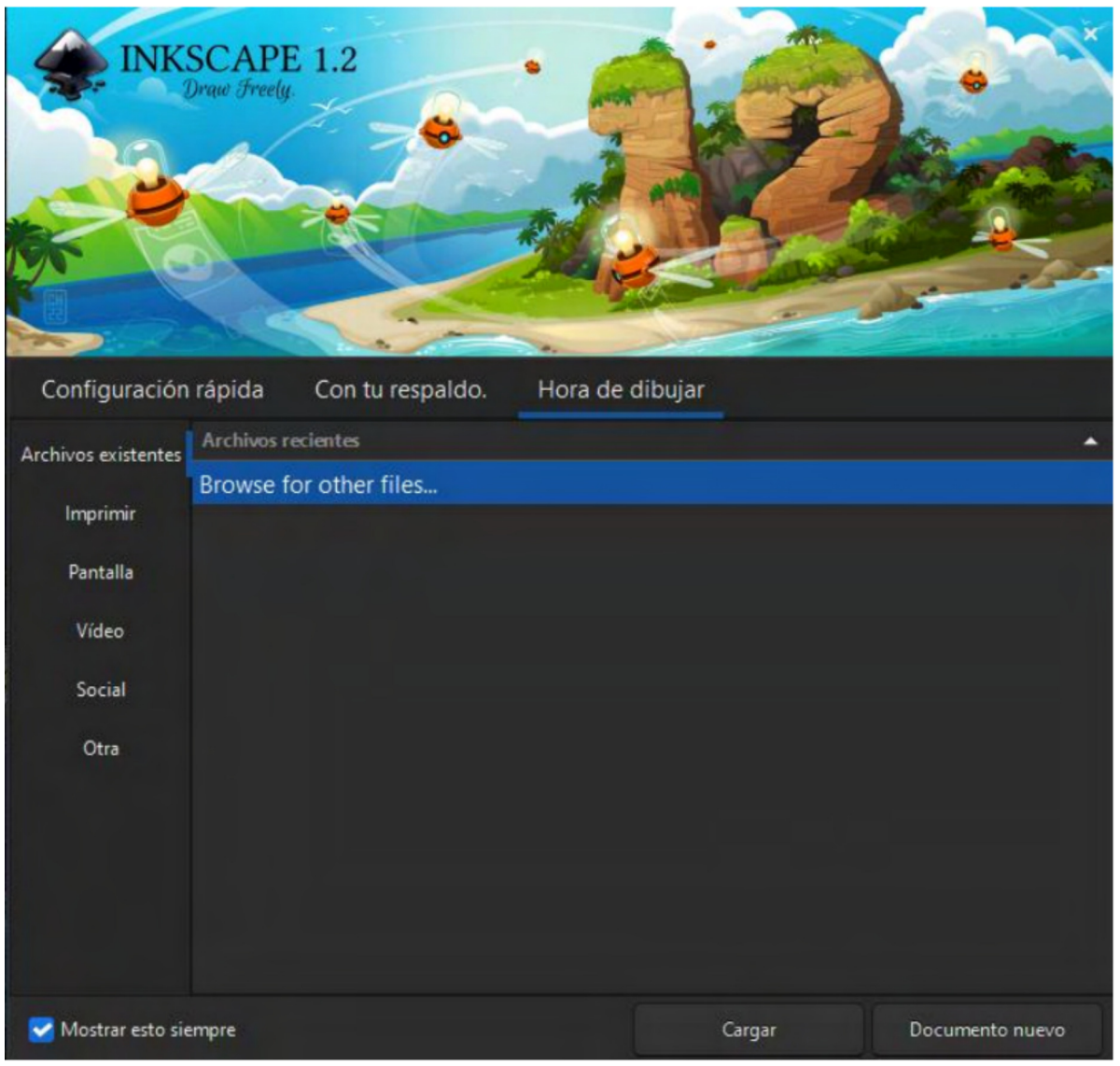


Figura 4.41: Programa Inkscape 1.2 (Creación Propia).

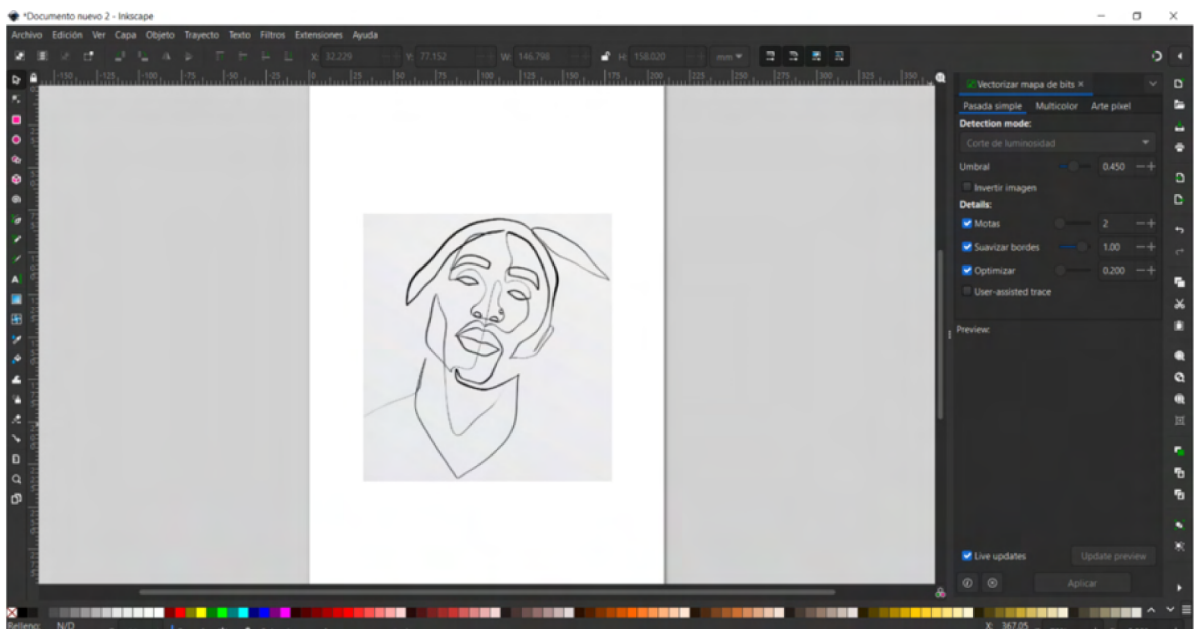


Figura 4.42: Imagen cargada en programa Inkscape 1.2 (Creación Propia).

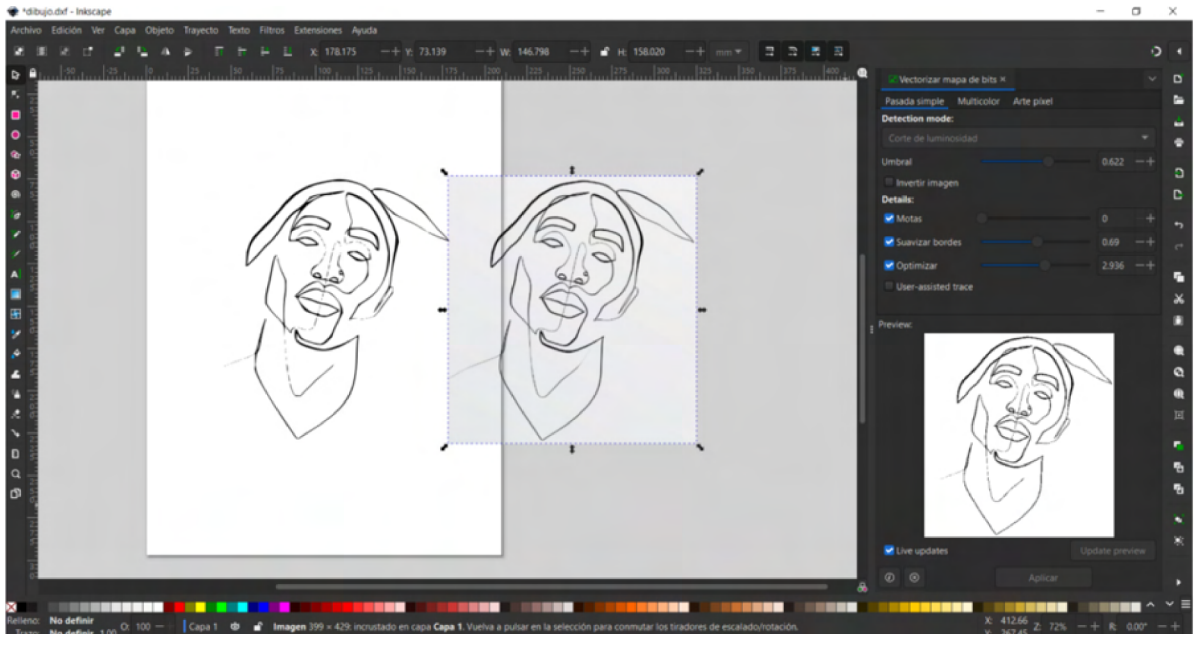


Figura 4.43: Imagen vectorizada en programa Inkscape 1.2 (Creación Propia).

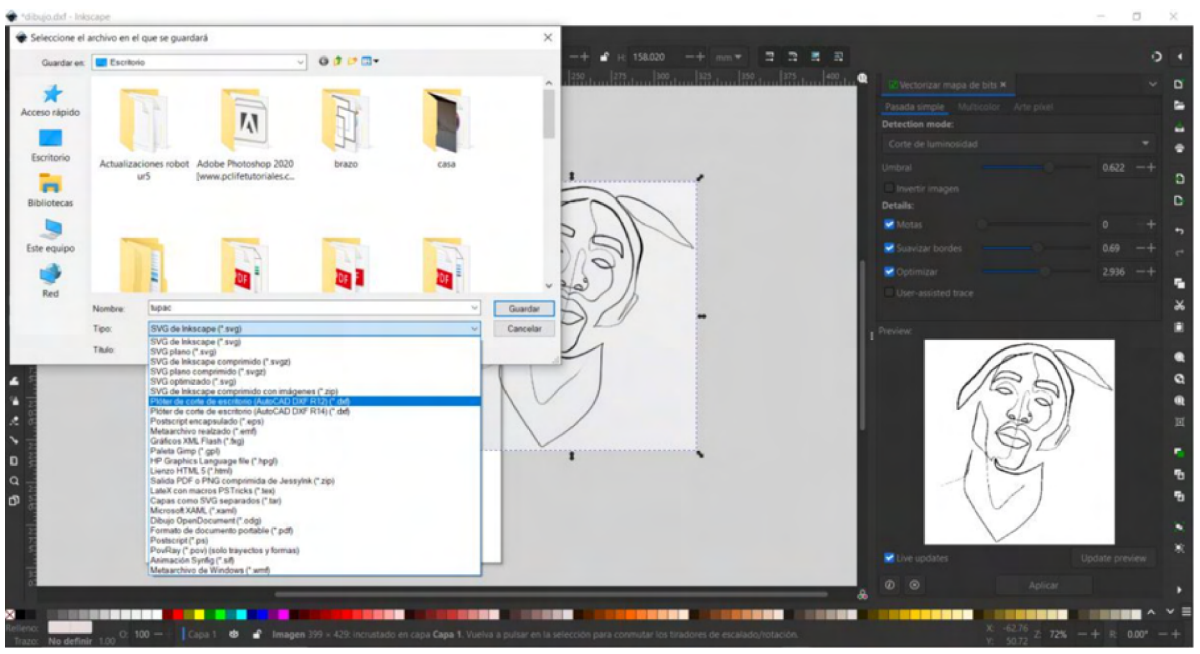


Figura 4.44: Guardado de archivo DFX (Creación Propia).

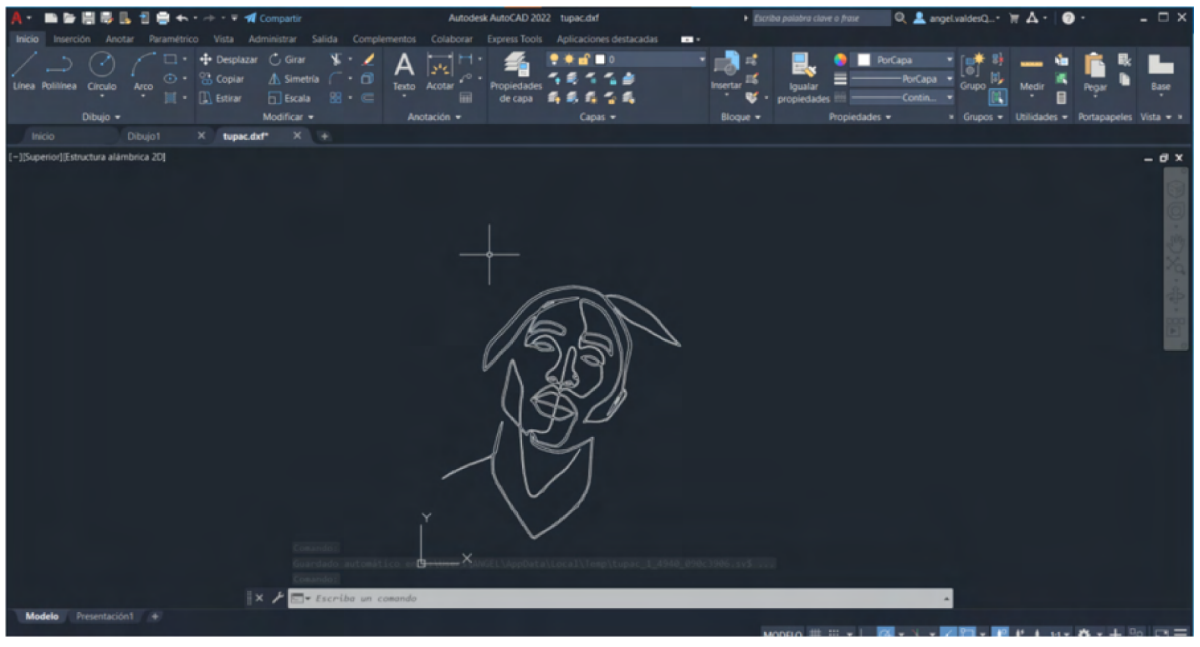


Figura 4.45: Imagen vectorizada representada en AutoCAD (Creación Propia).

4.2.3.2 DXF to G-Code Converter

Ahora bien, se tiene un archivo vectorizado de una imagen '.jpg' descargada de la web. Como se ha mencionado anteriormente las impresoras y cobot pueden leer archivos en código G y es por esto que se necesita de un software que pueda leer los vectores de la imagen y transformarlo en un código indicando las coordenadas que se debe mover para crear la forma deseada.

El programa "DXF to G-Code Converter" permite desde un archivo '.DXF' obtener un código G para utilizarlo en "Robodk". Así entonces, una vez instalada se va a *'file'*, luego a *'open'* y se busca el archivo guardado previamente vectorizada y se abre como se muestra en la figura 4.46. Se observa que se tiene la figura ya vectorizada y se puede configurar de tal manera que se reduzcan las líneas innecesarias para que al momento de dibujar el robot no haga movimientos extras. Una vez listo el dibujo deseado se cliquea en *'export y optimize and export shapes'* y abre una ventana (ver figura 4.47) donde el formato del archivo viene en código G. Asi entonces se tiene un archivo en dicho código para que el robot tenga los movimientos necesarios a seguir donde la trayectoria es la misma que conforma las lineas de la imagen.

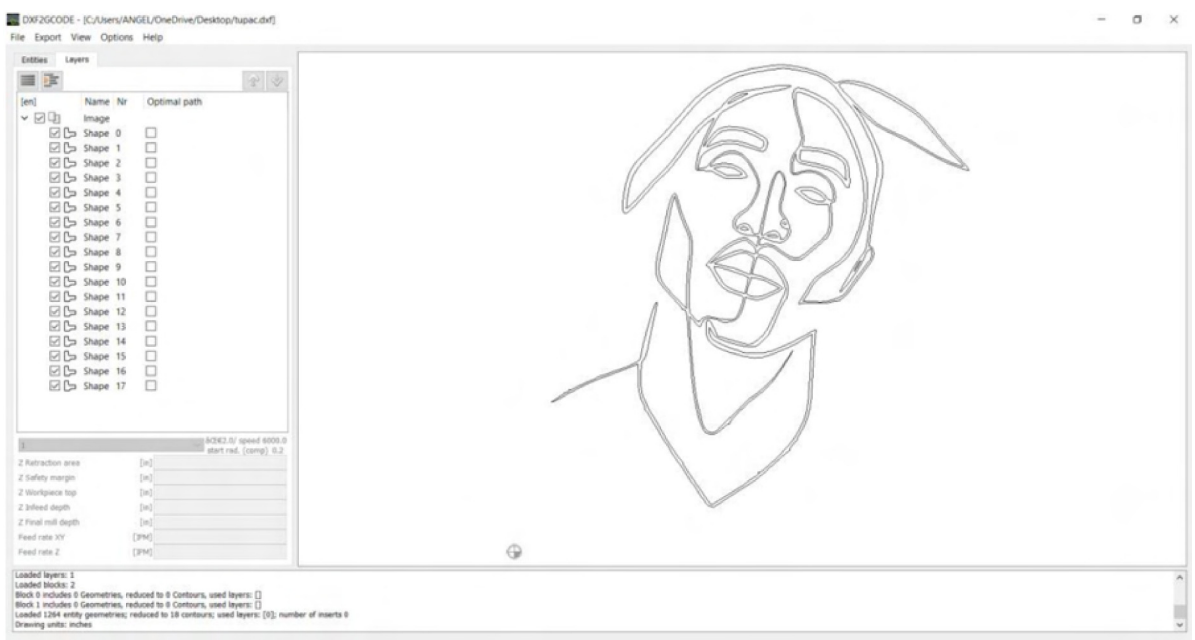


Figura 4.46: Imagen vectorizada en programa DXF to G-Code Converter (Creación Propia).

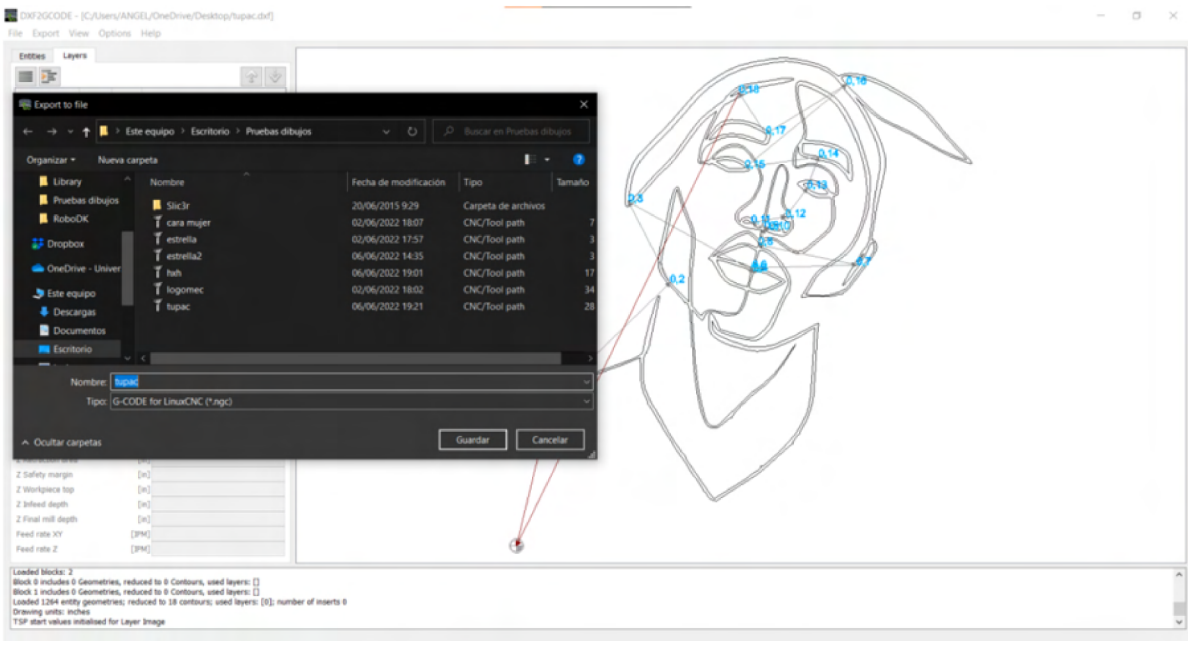


Figura 4.47: Guardado de imagen vectorizada en formato código G (Creación Propia).

4.2.3.3 Creación de programa en robodk para ser simulado y listo para ser enviado al Cobot

Una vez ya obtenido el archivo en código G de la figura deseada a dibujar, primero se debe tener el PCH indicado de la herramienta. Para el caso de ejemplo se utilizó un plumón y se obtuvo según lo explicado en la sección 4.2.2, luego se debe conocer la superficie en donde se moverá el brazo robótico, el cuál será una pizarra. Para esto se recomienda llevar el robot con la punta del plumón en este caso tocando la superficie, como se observa en la figura 4.48, ya sea manualmente con el botón para liberar los frenos o moviéndolo con los servomotores. Cabe mencionar que debe ser lo más perpendicular posible a la superficie y en una posición cómoda para el robot de modo que al momento de seguir la trayectoria no sufra de colisiones entre sus mismas partes.



Figura 4.48: PCH tocando la superficie deseada (Creación Propia).

Luego de haber obtenido la posición correcta se debe conectar el robot al *'software'* "RoboDK" explicado en la sección 4.2.1 y una vez conectado se debe dar click a obtener posición en la opción central debajo del puerto del robot como se muestra en la figura 4.49.

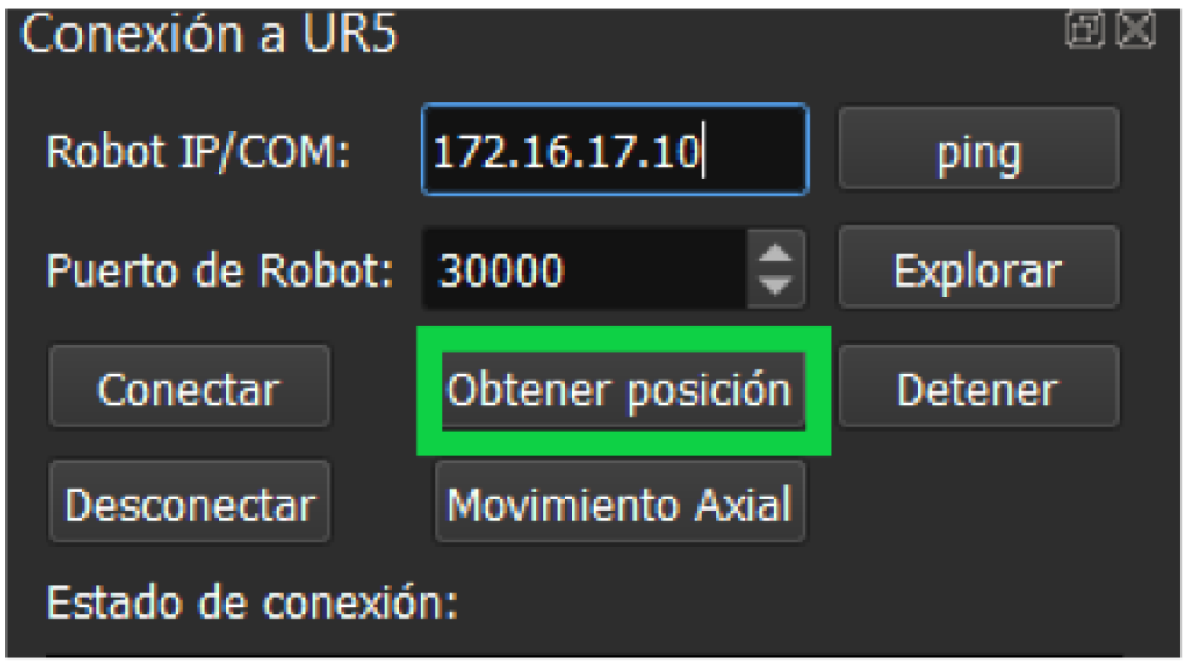


Figura 4.49: Obtener posición real de UR5 (Creación Propia).

Esta opción permite tener las coordenadas espaciales del robot en esa posición, es decir que nos muestra la coordenada que se debe utilizar para la mesa de trabajo. Así entonces, la recomendación para obtener el valor numérico de la posición es agregar un *'Target'* que se encuentra en las opciones superiores en el software "RoboDK" con un icono de un dardo dando al blanco con un signo "+" de color verde y se da click, instantáneamente se agregará como se muestra en la figura 4.50.

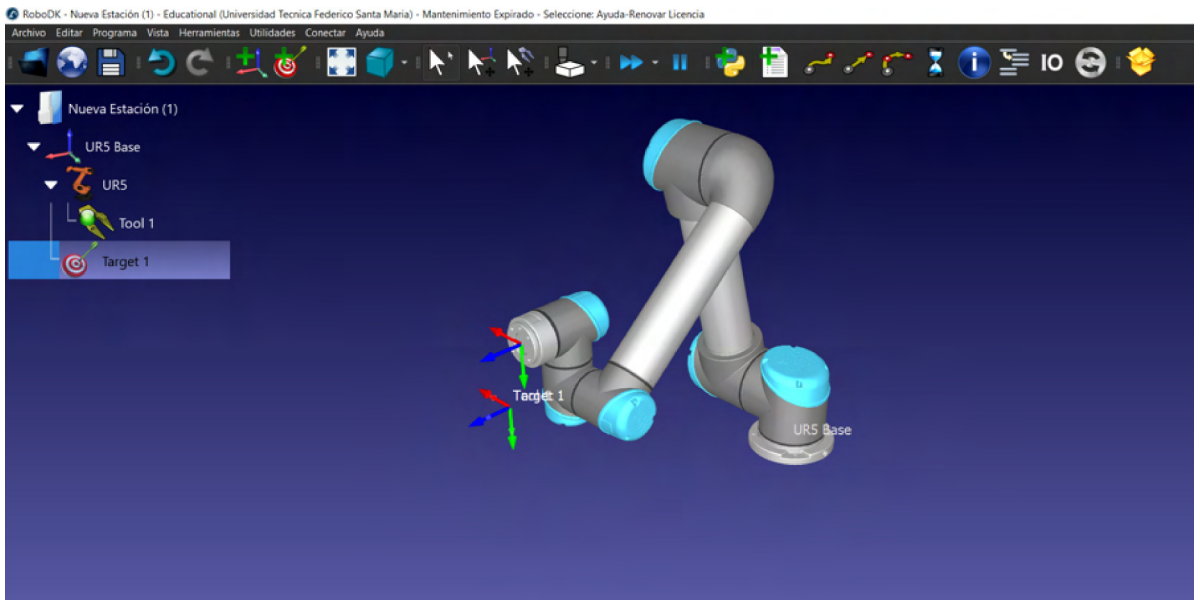


Figura 4.50: Agregar target (Creación Propia).

Así entonces, se tiene guardado la posición de la mesa que para este ejemplo se utilizo una pizarra. Ahora bien se debe tener una referencia de coordenadas de la pieza de trabajo como se mostró en la figura 4.10 de la sección 4.1.7. Para esto al igual que el *'Target'* existe una opción de agregar un *'Frame'* el cual se encuentra al lado con un icono de coordenadas y un símbolo "+". Se le da click y se agrega una coordenada de trabajo que el software reconocerá como mesa de trabajo para el robot agregado. Este *'frame'* no tendrá los valores de la mesa de trabajo que se necesita, es por esto que se guarda en un *'target'*. Para agregar las coordenadas al nuevo *'frame'* se debe dar click derecho a en el *'target'* que se agrego e ir a opciones, se abrirá una ventana en el lado derecho y luego hacer doble click en el *'frame'* agregado donde también se abrirá una ventana en el lado derecho como se muestra en la figura 4.51

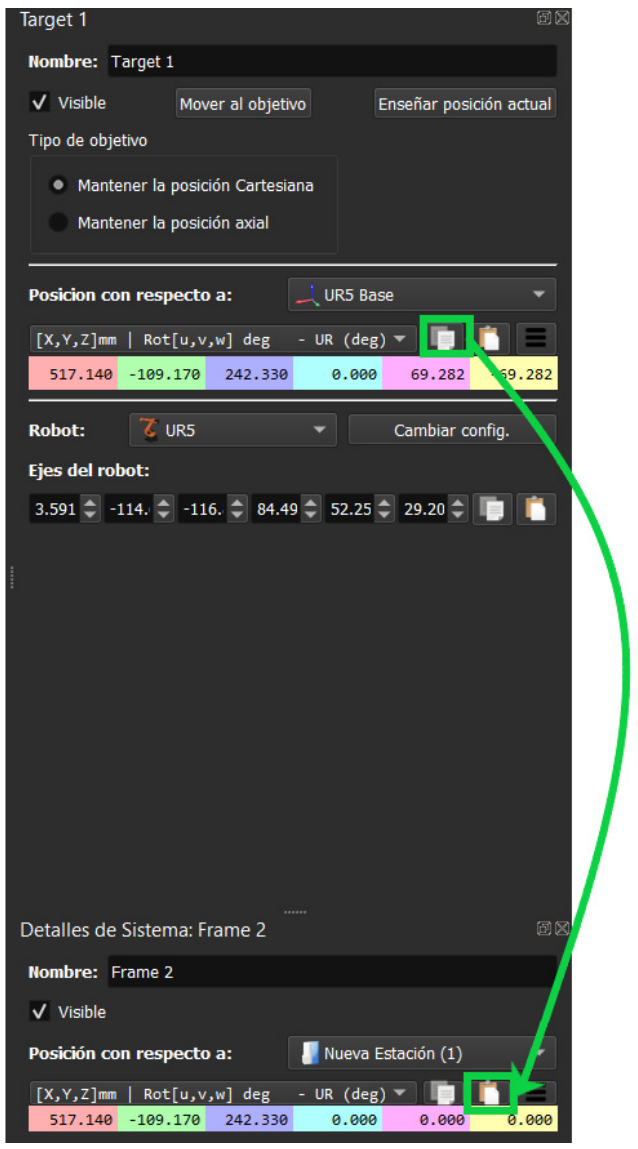


Figura 4.51: Copiar las coordenadas de Target a Frame (Creación Propia).

Como se muestra se puede copiar las coordenadas con los iconos marcados para que sea más rapido, pero tener en cuenta que los grados en el frame 2 deben ser cero. Los grados son los 3 últimos números, así el codigo cargado no estará rotado en algún eje.

Una vez ya con el eje de referencia de trabajo se agrega el archivo en código g de la imagen a dibujar directamente trasladando el archivo al software "RoboDK". Cuando se agrega se abrirá autómaticamente una ventana de ajustes como se muestra en la figura 4.52

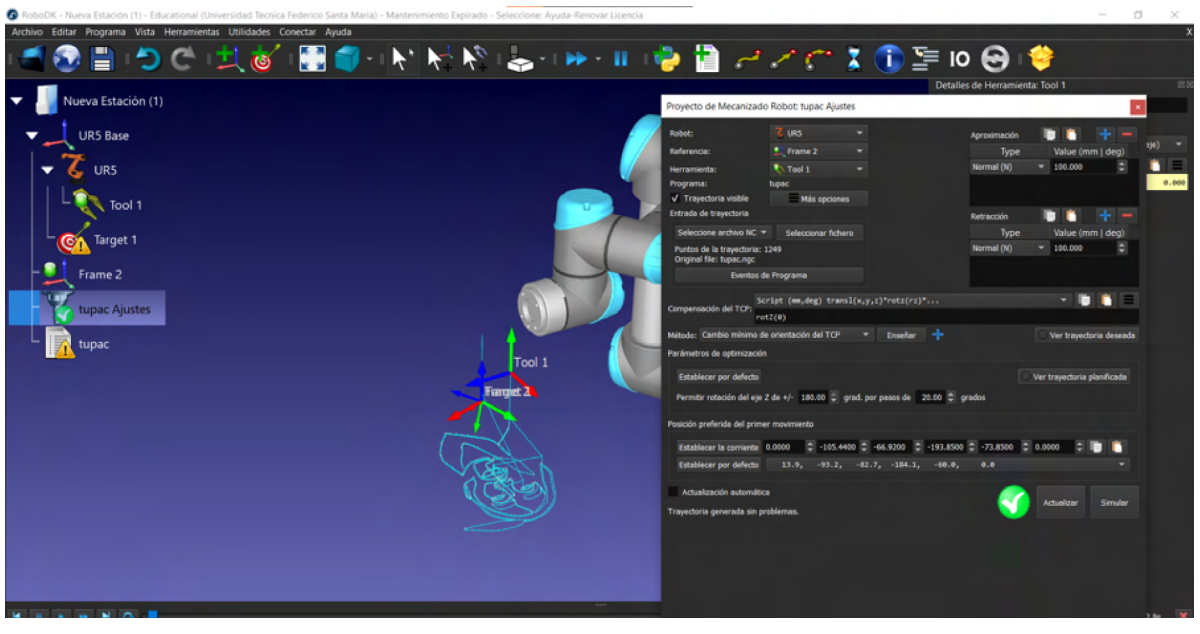


Figura 4.52: Imagen en código g en RoboDK (Creación Propia).

En esta ventana es donde se ajustan los movimientos y velocidad máxima del robot. En donde principalmente se enfocará el estudio para este ejemplo en la opciones: establecer corriente, enseñar, compensación de la herramienta, actualizar y simular. Como se muestra en la figura 4.53.

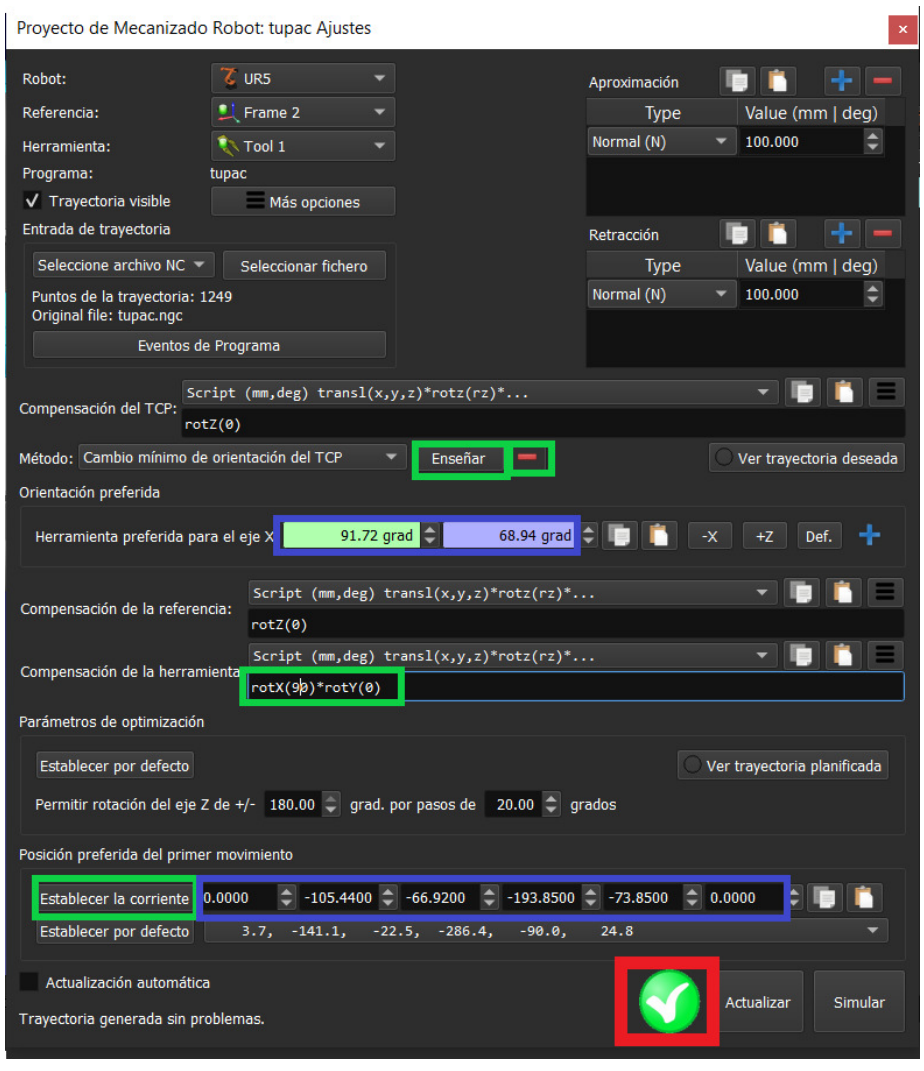


Figura 4.53: Opciones principales de ajustes (Creación Propia).

Las opciones encerradas en un rectángulo verde indican las opciones que se deben o pueden modificar para un correcto funcionamiento, las de color azul son las calculadas por el mismo software

y el cuadrado de color rojo indica que la simulación se puede realizar sin problema. Ahora bien primero se debe explicar porqué la importancia de estas opciones encerradas en color verde.

- Establecer corriente: Esta opción le indica al software de donde tiene que empezar a tomar el primer movimiento. Esto es importante debido a que para la impresión 3d o para el dibujo no se quiere que el robot se mueva, sobre todo la muñeca número 3 (ver figura 4.1 de sección 4.1) debido a que tendrá una herramienta que siempre tiene que trasladarse en una misma posición.

- Enseñar: Esta opción genera que el software calcule la mejor opción de orientación de la herramienta para la posición establecida, lo que ayuda a reducir movimientos en la muñeca 3 y que se obtengan mejores resultados.

- Compensación de la Herramienta: Esta opción al momento de simular es muy importante ya que rota la herramienta dependiendo los grados que se necesiten para el punto de contacto. Es particularmente útil cuando el punto de interés o punto de acción se encuentra en una ubicación diferente de la punta de la herramienta. Por ejemplo, si se tiene una herramienta con una longitud adicional o una geometría complicada, la compensación de herramienta permite especificar la ubicación exacta donde se produce el punto de acción de la herramienta. Un ejemplo se puede observar en la figura 4.54 a 0° respecto de X donde se observa que el punto de acción entre la herramienta y la mesa de trabajo no esta siendo perpendicular y afectará notablemente los resultados ya que la punta del plumón no tocará la pizarra. Al contrario si se modifica a 90°como se muestra en la figura 4.55 se puede observar que se tiene un resultado mas positivo a lo que se requiere, ya que el plumón esta perpendicular a la mesa de trabajo. Hay que tener en cuenta que no siempre se tendrá que rotar un solo eje, dependerá de como este posicionado el robot, la pieza de trabajo y las restricciones de movimiento.

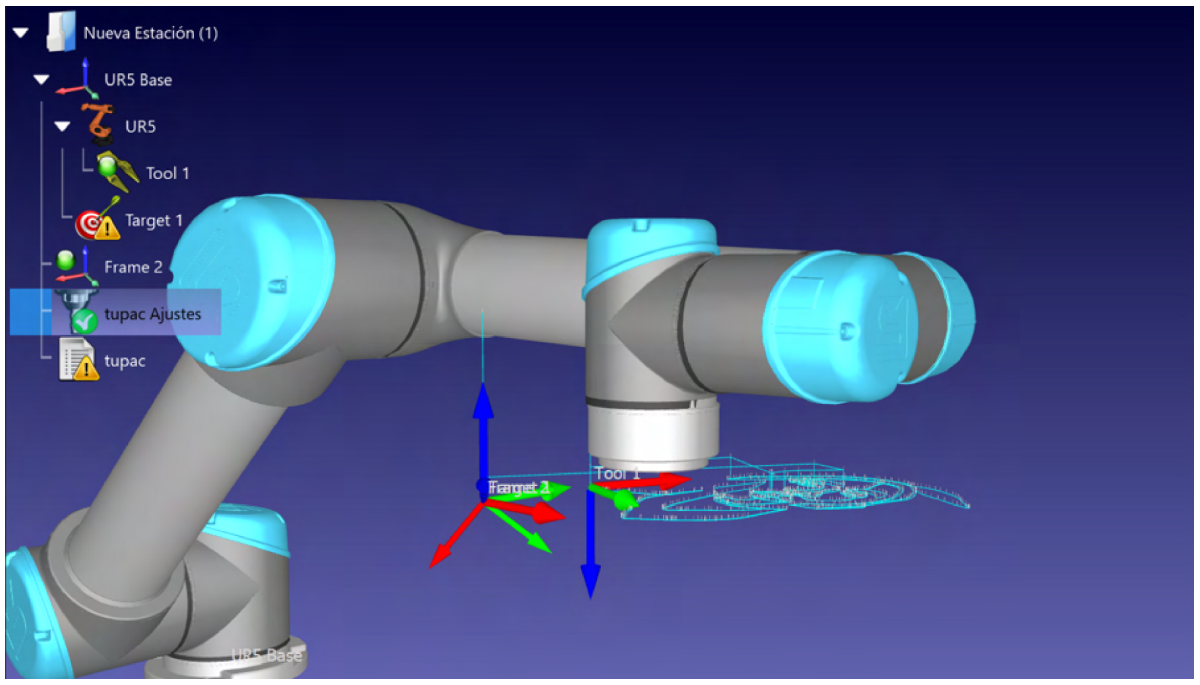


Figura 4.54: Compensación de la herramienta a 0° respecto de X (Creación Propia).

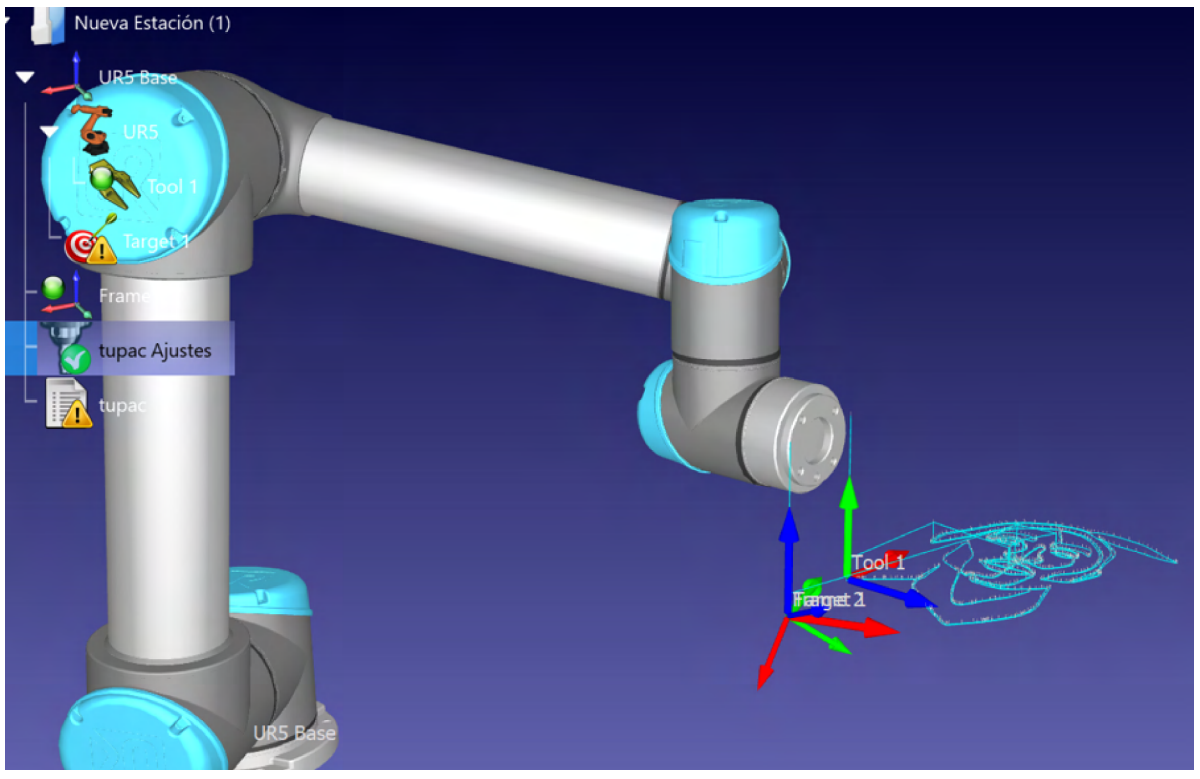


Figura 4.55: Compensación de la herramienta a 90° respecto de X (Creación Propia).

- Actualizar: Esta opción permite que cada vez que se modifica algún parámetro en los ajustes, se actualice el programa y verifique si es posible hacer o no los movimientos. Esto último se verifica con el símbolo verde mostrado en la figura 4.53, en caso de que no se pueda realizar será de color rojo con una equis.

- Simular: Esta opción es el último paso antes de pasar el programa al robot y sirve para confirmar con la ayuda gráfica que permite el software a través de los movimientos digitales del robot observar si cumplen con los requisitos necesarios para cumplir la tarea en el robot real. Si no se cumplen verificar las opciones anteriores y corregir.

Finalmente, ya corregido y visualizado la simulación se puede ejecutar en el robot el programa creado en "Robodk". Como recomendación se debe estar presente en todo momento de la ejecución del programa, ya que pueden existir factores externos que pueden complicar al robot y crear daños o que el trabajo no se realice, como por ejemplo que el suelo no este nivelado y que con ciertos movimientos del robot generen vibraciones fuertes que activen una parada de emergencia.

4.3 Impresión 3D

Ahora bien, recopilando todo lo aprendido en las secciones anteriores del Capitulo 4 se puede indagar en la manufactura aditiva, que para el caso de este estudio es la impresión 3D. Para este tipo de impresión es importante tener consideraciones antes de realizar prácticas, las cuales vienen dadas primero por el tipo de material, para nuestro caso es plástico, debido a que establecerá el rango utilizado en la temperatura para su adecuada fundición. También consideraciones de velocidad, tanto del Cobot como del tornillo sinfín del extruder, para esto no existe un modelo definido y constará del estudio, pruebas y experiencias compartidas que guiarán a un buen resultado. Una mesa rugosa con temperatura que permita que el plástico se asiente y no se corra al momento de pasar a otra capa también es importante para un buen resultado. Y por último un factor que se debe tener en cuenta es que se va depositar gran cantidad de material fundido encima una de otra, esto dará como resultado que por el peso y la facilidad de moldeo por la temperatura, las capas tiendan a caer.

Todas las consideraciones fueron compartidas y aprendidas a lo largo del estudio y que se darán a conocer más detalladamente a continuación.

4.3.1 Configuración actual Cobot-Extruder

En la figura 4.56 se muestra como esta configurada las distintas partes que conforman la manufactura aditiva de alto volumen del laboratorio Eco-Fab-Lab, las cuales son: Brazo robótico UR5 (Cobot), Extruder de alto volumen, Brazo compensador y Cama caliente (cama de impresora 3D Ender de laboratorio Eco-Fab-Lab). Cabe mencionar que las primeras pruebas de movimiento entre las 3 partes móviles (cobot, extruder y brazo compensador) fue una configuración distinta a la actual.



Figura 4.56: Configuración actual de Cobot-Extruder y brazo compensador (Creación Propia).

Como se muestra en la figura 4.57 el cobot se encuentra por encima del brazo compensador al contrario a la configuración final, pero más importante aún es la unión del brazo robótico a la placa que une los elementos. Esto durante las pruebas fue un problema como se explicará más adelante.



Figura 4.57: Configuración inicial de Cobot-Extruder y brazo compensador (Creación Propia).

4.3.2 Movimientos permitidos por el brazo compensador

En la figura 4.58 se muestra el plano de brazo compensador serie 3 de la empresa 3arm. Este brazo articulado permite que el cobot no sostenga los 11 [kg] del extruder, ya que el UR5 solo permite hasta 5 [kg] lo que se tiene que solo guíe al extruder de alto volumen sin la necesidad de sostener la masa de éste. El brazo compensador puede equilibrar de 8 a 15 [kg] con una extensión máxima de 918 [mm] y movimientos verticales de hasta 535 [mm] desde el punto más bajo al más alto.

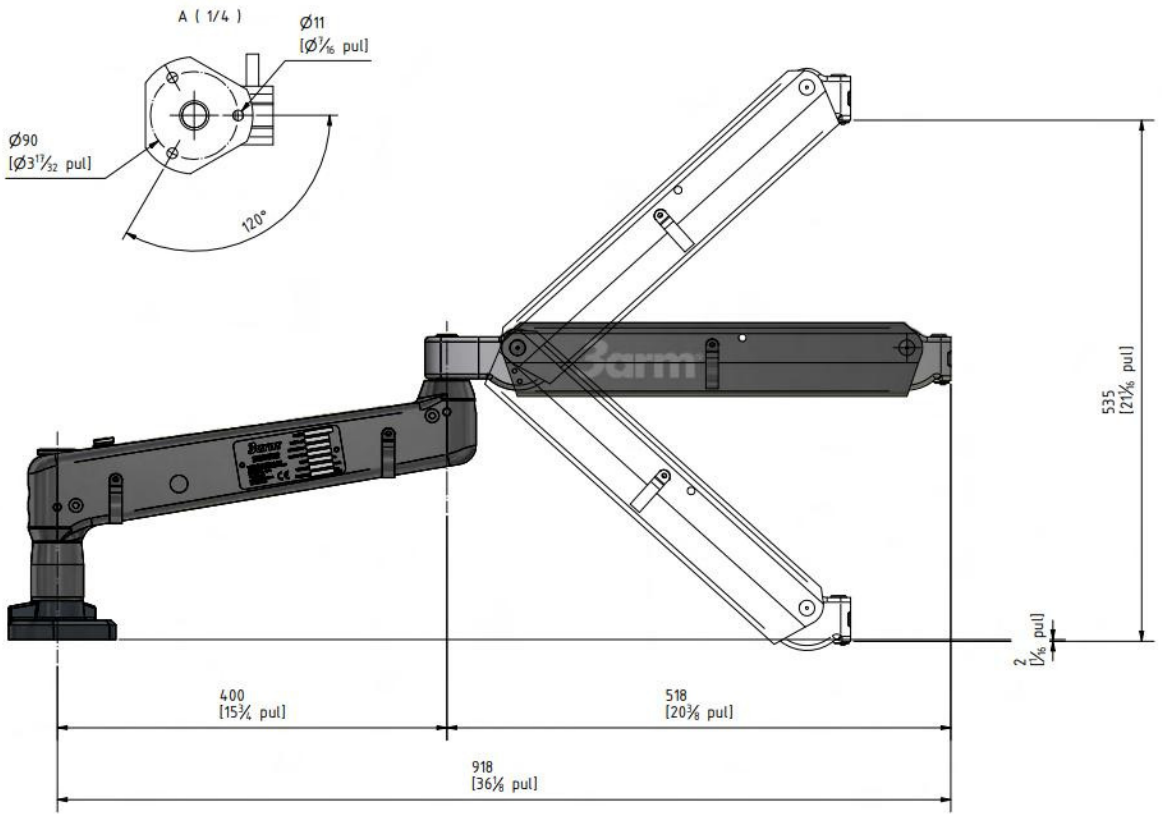


Figura 4.58: Plano brazo compensador serie 3 [25].

Ahora bien como se mencionó anteriormente el brazo es articulado, en la base y codo, mientras que el cabezal es plano con movimientos verticales para tener equilibrado a la herramienta compensada. Este punto es importante ya que el cabezal seguirá la ruta, como se muestra en la figura 4.59, de manera tangencial creando una trayectoria en coordenadas polares. Debido a que forma un arco dificulta los movimientos del cobot, ya que este último se mueve en coordenadas cartesianas al cargar un programa para trayectorias de impresión 3D como se explicará más adelante.



Figura 4.59: Movimientos permitidos por brazo compensador serie 3 [25].

4.3.3 Solución a problema de movimiento de brazo compensador

Como se mencionó en la sección 4.3.1 en la configuración inicial se tenia a la muñeca 3 unida a la placa de forma directa, pero como se explicó en la sección 4.3.2 se tiene 2 coordenadas distintas que restringen los movimientos del UR5 como se observa en la figura 4.60.

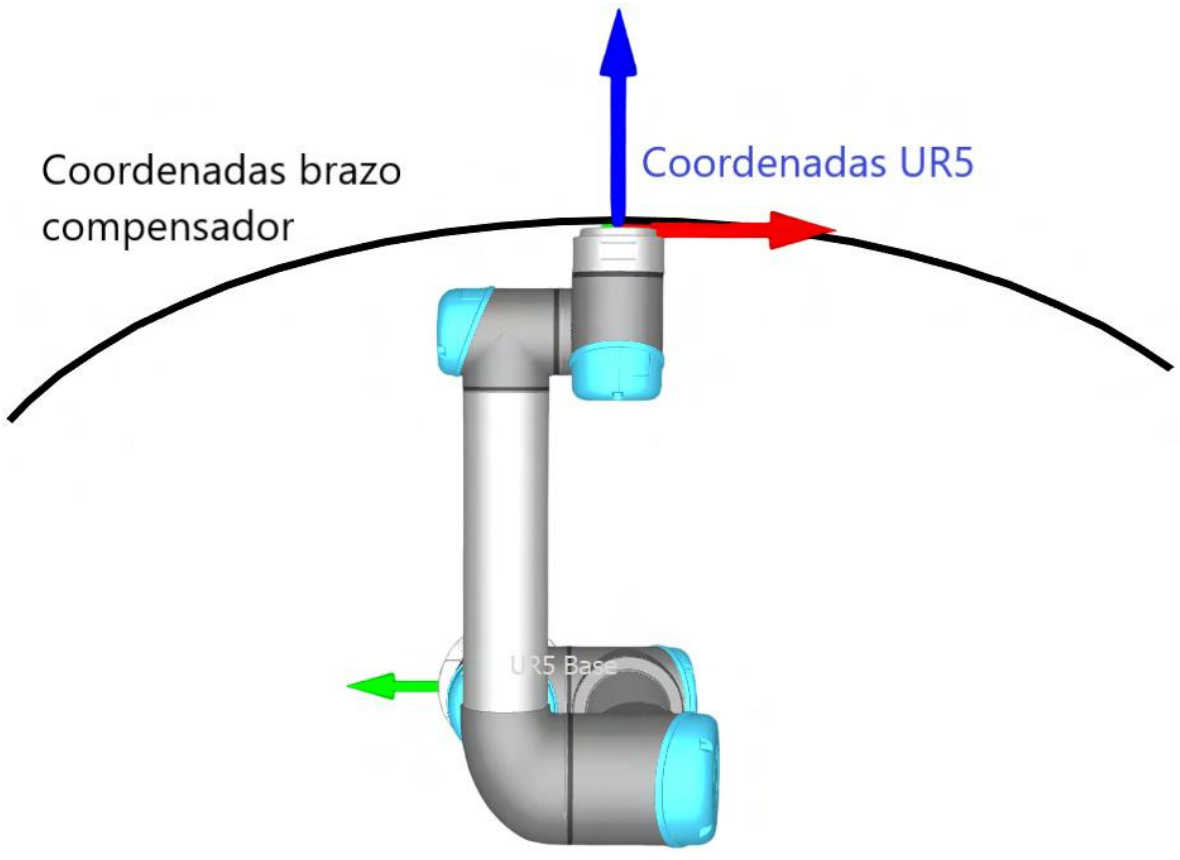


Figura 4.60: Cruce de Coordenadas de Cobot y Brazo compensador (Creación propia).

Cabe mencionar que los movimientos en el eje Z, no fueron el mayor problema, ya que el brazo robótico se movía de manera óptima sin mayores dificultades. Pero los movimientos en los planos X e Y provocaban que el sistema del cobot parará los movimientos y entraba en parada de emergencia, debido a las fuerzas involucradas al querer mover el brazo robótico a una dirección pero el brazo compensador empujaba hacia sus propias coordenadas.

En la figura 4.61 se observa la unión inicial del cobot a la placa, como se venia explicando estaba unida de forma directa y se producían fuerzas al brazo robótico al querer mover en el plano X e Y de manera recta.



Figura 4.61: Unión inicial (Creación propia).

Es por esto que se encontró una solución sencilla y rápida, pero sobre de bajo costo que ayudo al robot moverse libremente a pesar de las restricciones del brazo compensador. Como se muestra en la figura 4.62 esta solución constó de unir el UR5 a la placa con una bisagra modificada de tal manera que el movimiento de ésta se ubicará en el centro de la base de la muñeca 3.

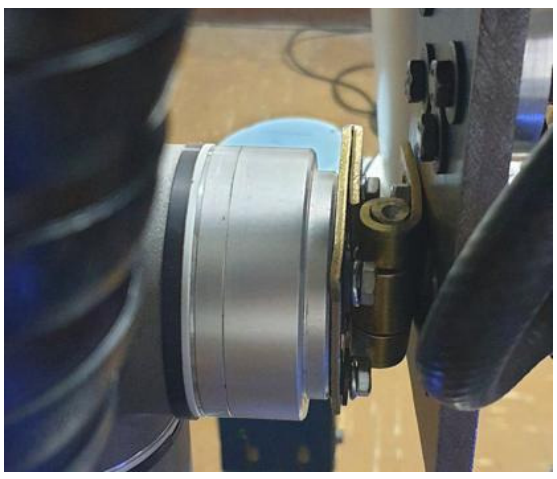


Figura 4.62: Unión final (Creación propia).

Esta unión hizo que el robot tenga más grados de libertad y por ende poder hacer pruebas de programas 3D con el extruder de alto volumen de manera segura y sin que el cobot arrojase para de emergencia por fuerzas externas.

4.3.4 TCP o PCH del Extrusor de alto volumen junto con la unión

Para poder realizar un correcto programa de impresión 3D al brazo robótico se debe tener en cuenta el punto central de la herramienta (PCH), en este caso es el *extruder* de alto volumen. Pero como se mencionó en las secciones anteriores se montó un brazo compensador el cual se lleva el peso de éste y el robot sólo guía los movimientos. Sin embargo, esta triple unión dificulta que el mismo *software* pueda calcular el PCH, explicado en la sección 4.1.7, debido a que no se puede rotar la muñeca número 3 y tener distintas poses para que se calcule automáticamente.

En consecuencia se realiza manualmente, midiendo desde el origen en los 3 ejes (x,y,z) de la muñeca número 3 como se muestra en la figura 4.63. Y para saber el punto de referencia en el robot real, se debe tomar la conexión integrada que trae el cobot para herramientas que se conectan a los *'input'* y *'output'* del mismo, que tiene forma de cilindro que sobresale de la muñeca. Esta conexión esta en el eje Y en la parte negativa, por lo tanto ese será el punto de referencia a tomar para medir desde el origen de los ejes a la punta (boquilla) del extrusor.

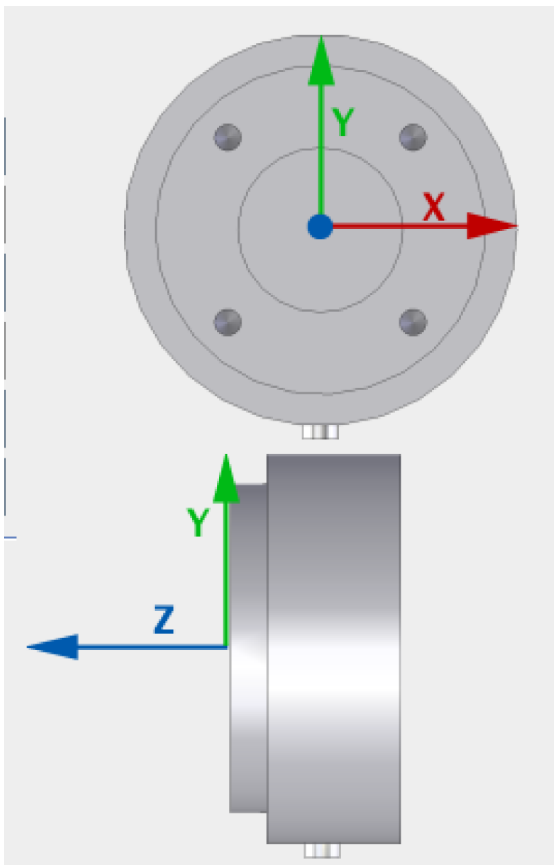


Figura 4.63: Coordenadas base muñeca 3 (Creación propia).

Como se observa en la figura 4.64 se toma la medida desde el centro de la muñeca 3 al eje central del extrusor en dirección del eje Z teniendo en cuenta la distancia entre la placa y el centro de la muñeca 3, es decir la distancia que ocupa la bisagra (a), además del espesor de la placa que son 10 [mm] (b) y por último la distancia de la placa al eje central (c). Luego por la configuración final de la impresora 3D de alto volumen y la unión que se hizo de tal manera que quedará lo más

centrado posible, obteniendo que no hay un mayor desplazamiento en el eje Y, por lo que se mide desde el centro de la muñeca 3 a la boquilla del *extruder* en dirección a X negativo (d), Obteniendo las siguientes medidas como se muestra en la tabla 4.4:

Tabla 4.4: Medidas PCH de extrusor de alto volumen en configuración final

Eje	Medida	Unidad
-X	209.5	mm
Y	0	mm
Z	121	mm

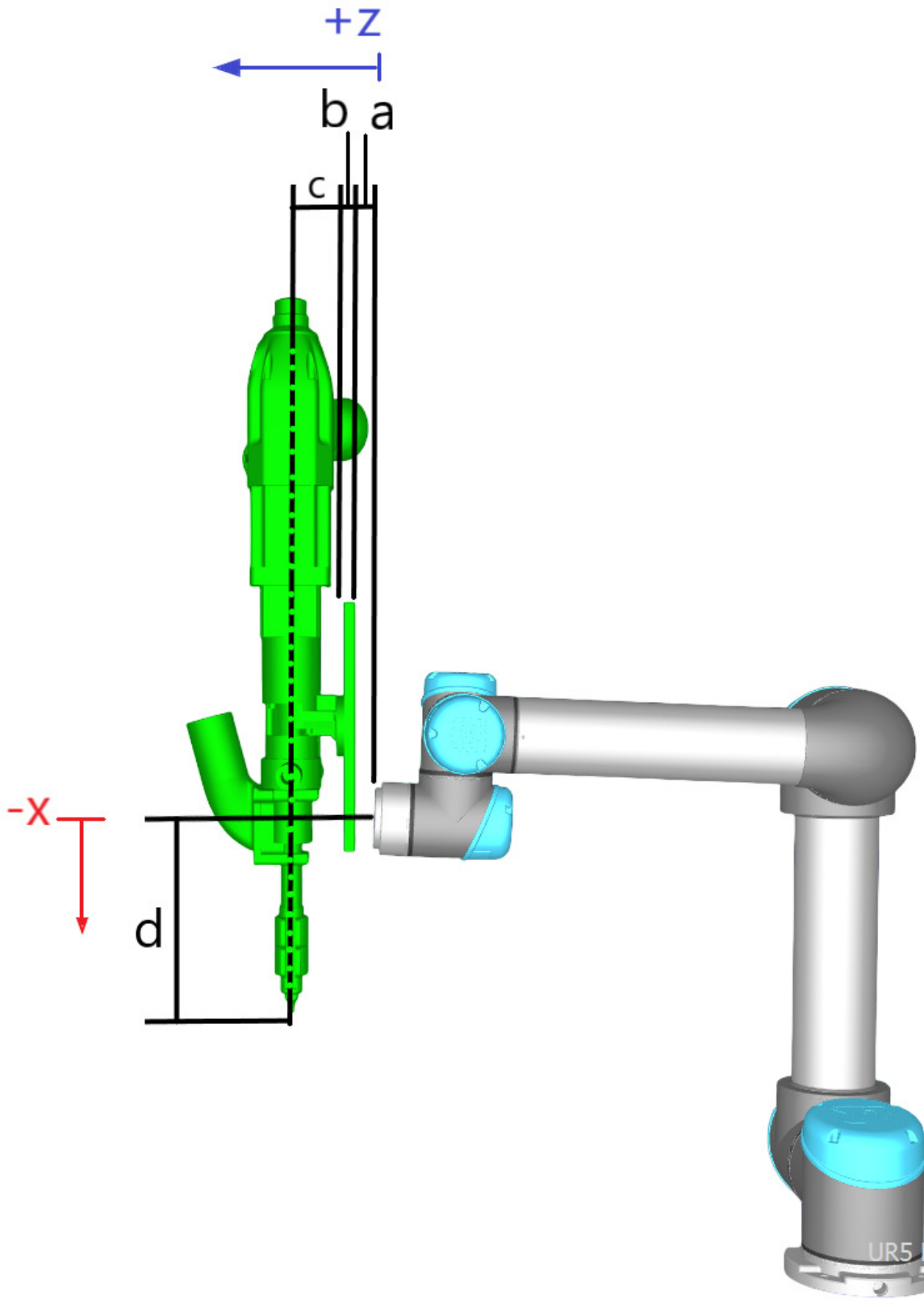


Figura 4.64: Imagen representativa de toma de medidas TCP Extrusor (Creación propia).

4.3.5 Creación de programa para impresión 3D

Ahora bien, teniendo en consideración los puntos anteriores de la sección 4.3 que son principalmente problemáticas físicas resueltas las cuales ayudarán a que el programa creado funcione de manera óptima, se puede pasar al siguiente paso, el cual es crear las trayectorias que el cobot deba seguir para obtener una figura u objeto en las tres dimensiones. Para obtener dichas trayectorias hay que tener en cuenta en la creación de la figura que para este estudio se realizo en el *software* "Inventor", el cual permite realizar objetos tridimensionales. Luego se debe, al igual que el programa de dibujo explicada en la sección 4.2.3, realizar el código G con la ayuda de un *software* que para este estudio es "Ultimaker Cura" utilizado en la mayoría de impresoras 3D, el cual entrega el código detallado dependiendo de las opciones que se necesiten. Finalmente, llevar dicho programa a "RoboDk" el cual obtendrá las trayectorias deseadas para conformar el objeto impreso, hay que tener en cuenta que este programa no tendrá en consideración los movimientos permitidos por el brazo compensador por lo que se deben hacer simulaciones antes de ejecutarlo en el brazo robótico UR5.

4.3.5.1 Creación de código G con slicer Ultimaker Cura

En la figura 4.65 se muestra un cilindro creado en inventor con un diámetro de 50 [mm], espesor de 3 [mm] y una altura de 15 [mm].

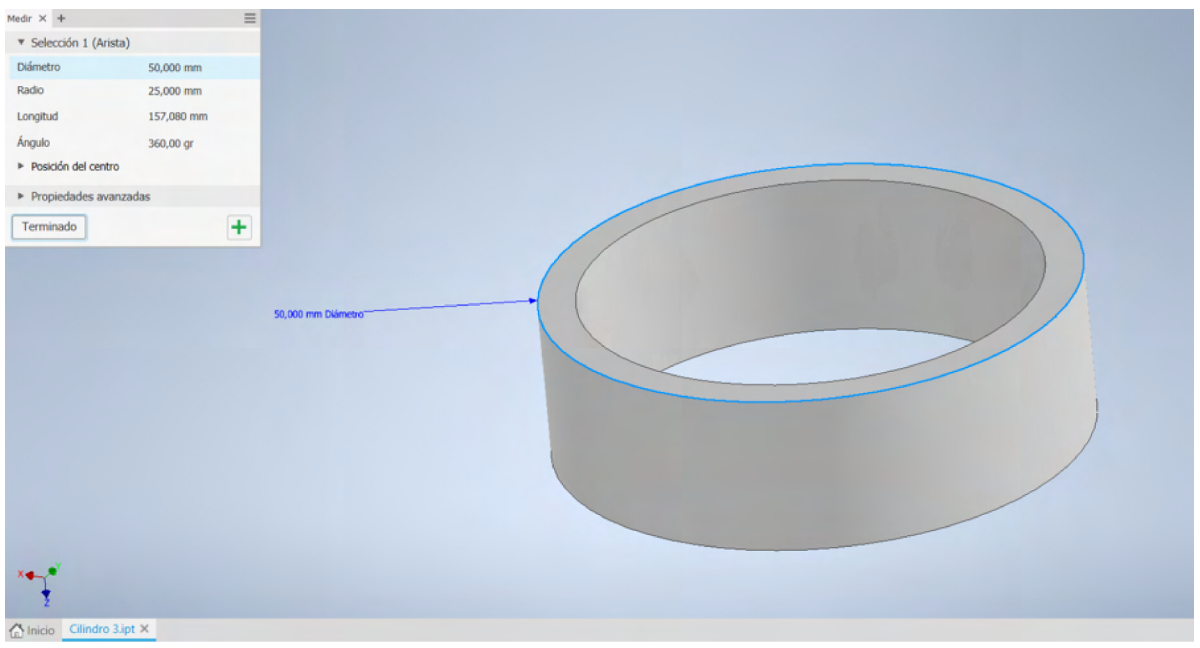


Figura 4.65: Cilindro de 50 [mm] de diámetro en Inventor (Creación propia).

Estas medidas, sobretodo el espesor y la altura tienen relación debido a que la boquilla del extrusor de alto volumen tiene un diámetro de 3 [mm], entonces con una pasada se obtendrá el espesor y por tanto con 5 vueltas se debe completar la altura de 15 [mm].

Se debe tener en consideración que al momento de exportar el archivo debe ser recomendablemente

en STL y con la opción de alta resolución como se muestra en la figura 4.66. Esto es importante ya que al momento de llevar el archivo en STL al *slicer* "Ultimaker Cura", será sensible a las trayectorias y si es de baja calidad para trazados curvos tomará trayectorias rectas pequeñas pero notorias al contrario de una alta resolución como se observa en la figura 4.67 donde el cilindro del lado izquierdo es exportado en una alta resolución mientras que el cilindro del lado derecho en una baja resolución.

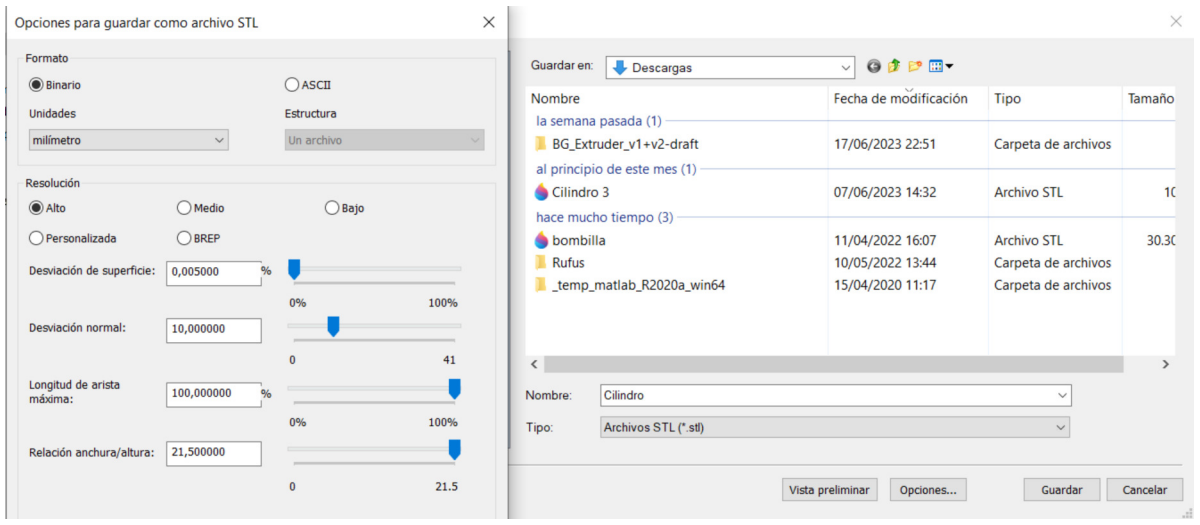


Figura 4.66: Opción de alta resolución para importar objeto en archivo STL(Creación propia).

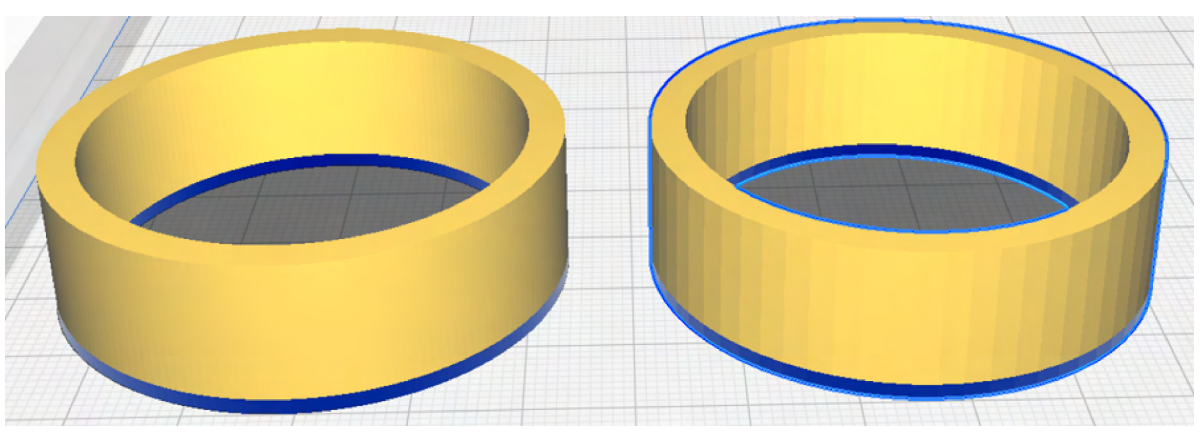


Figura 4.67: Comparativa entre cilindro exportado en alta resolución y baja resolución(Creación propia).

Las lineas rectas cortas tendrá resultados negativos en el cobot, ya que se deberá mover de manera recta en trazos pequeños y cambiar de dirección. Estos cambios rápidos se ven reflejados en como funciona el brazo robótico, debido a que tendrá frenados y por ende movimientos en el sistema que perjudicarán la vida útil del cobot como también los resultados de la impresión 3D.

Asi entonces, se debe importar el archivo de STL en alta resolución a *slicer* "Ultimaker Cura" y se observa como se muestra en la figura 4.68.

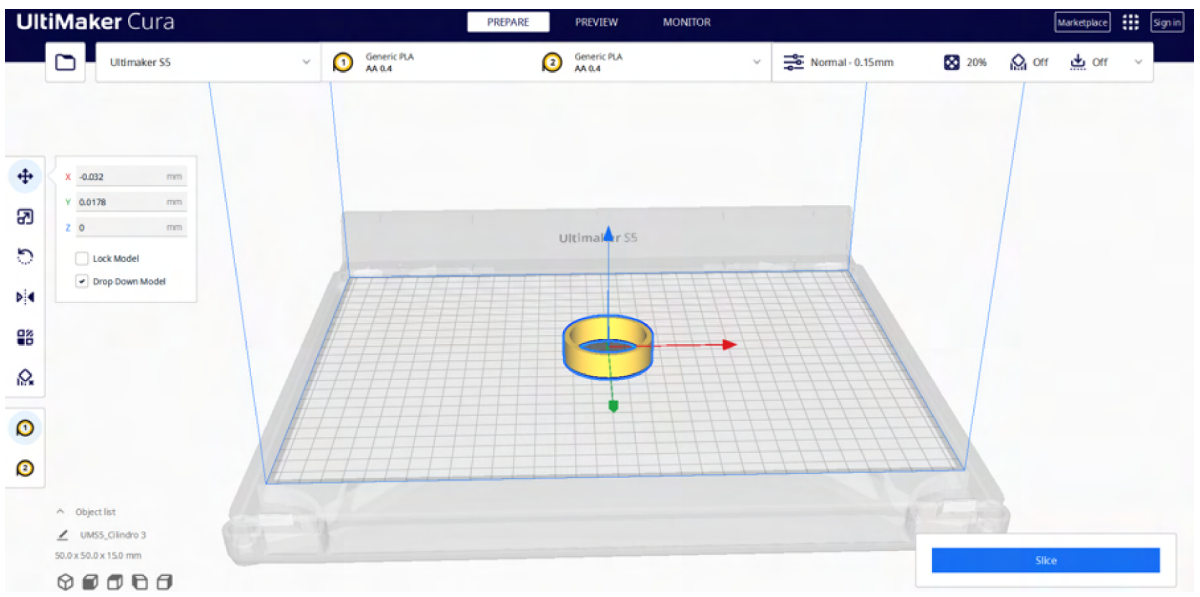


Figura 4.68: Interfaz slicer Ultimaker Cura con cilindro (Creación propia).

Al momento de importar un archivo y ocupar la mesa por defecto, esta trae el origen en la esquina izquierda inferior por lo que afectara al código G que se crea, debido a que tomará las coordenadas desplazadas del origen. Es por esto que para mayor comodidad se debe centrar el origen a la mesa en la opción de configuración, luego a impresoras y dando click en configuración de la máquina. Se abrirá una ventana como se observa en la figura 4.69 y habilitar la opción origin center. Notar que también se puede modificar las dimensiones de la mesa, pero para este estudio es suficiente el espacio.

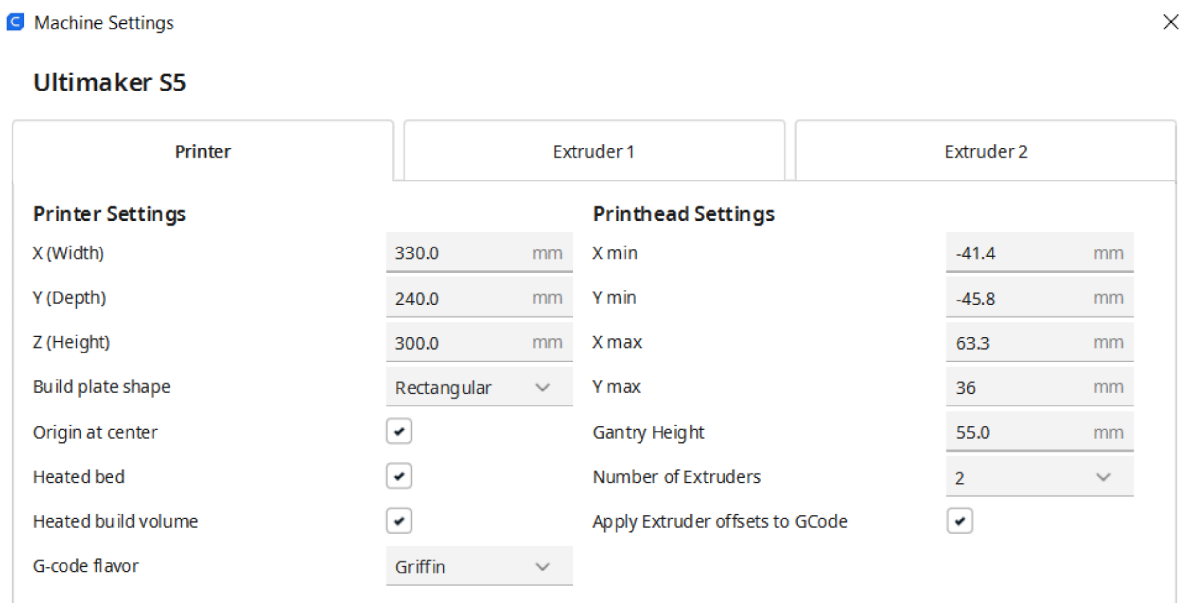


Figura 4.69: Opción para centrar la pieza u objeto (Creación propia).

Para las configuraciones de la calidad de las capas se tiene las opciones de *'Quality'* como se muestra en la figura 4.70, donde se enfocará en 2 opciones. La primera es *'Layer Height'* en el cual se informa al *slicer* de cuanto deben ser las alturas de las capas, para este estudio como el diámetro de la boquilla es de 3 [mm] la altura debe ser mínima de 3 milímetros, debido a que dependerá de como fluya el material si se requiere dar más altura o no. La segunda opción importante es *'Initial*

Layer Height', que tiene como función dar a conocer la altura de la capa inicial al código G, se recomienda que sea cercana en un rango de 5 [mm] menor al diámetro de la boquilla o el mismo valor para asegurar un buen asentamiento del material a la mesa. Para el ancho de linea esta la opción *'Line Width'* la cual esta influenciada por el ancho de la boquilla, que para este estudio es de 3 [mm]. Las otras opciones se auto-calcularán por el mismo *software*

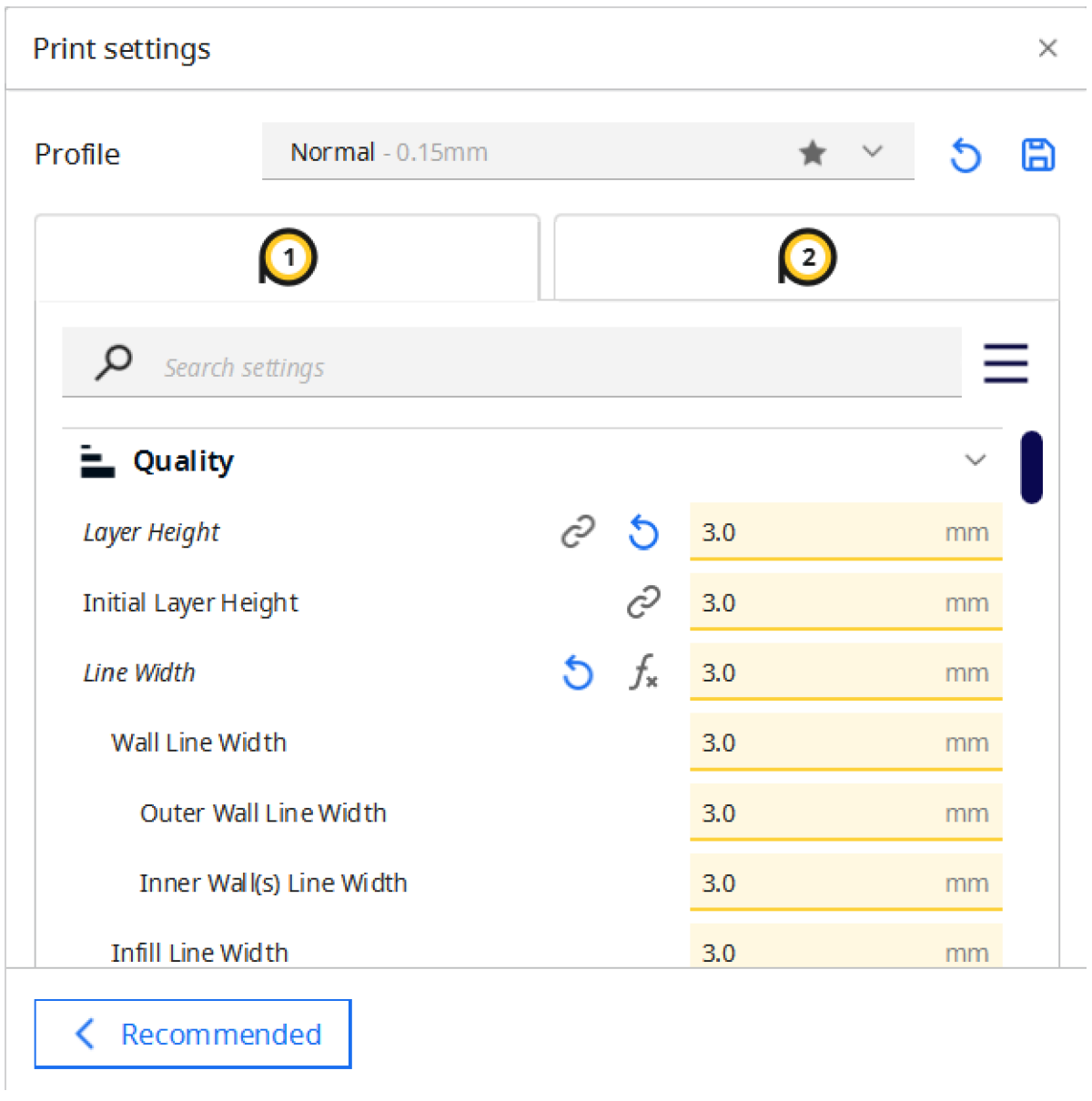


Figura 4.70: Opciones de calidad de las capas(Creación propia).

Otra configuración importante son las paredes como se muestra en la figura 4.71. En esta configuración es importante que la opción *'Wall Line Count'* sea 1, ya que esto indica cuantas veces pasará el cobot por la misma capa y como esta configurado para que el espesor del cilindro sea el mismo que el diámetro de la boquilla del extrusor de alto volumen es necesario indicar o comprobar que sea 1.

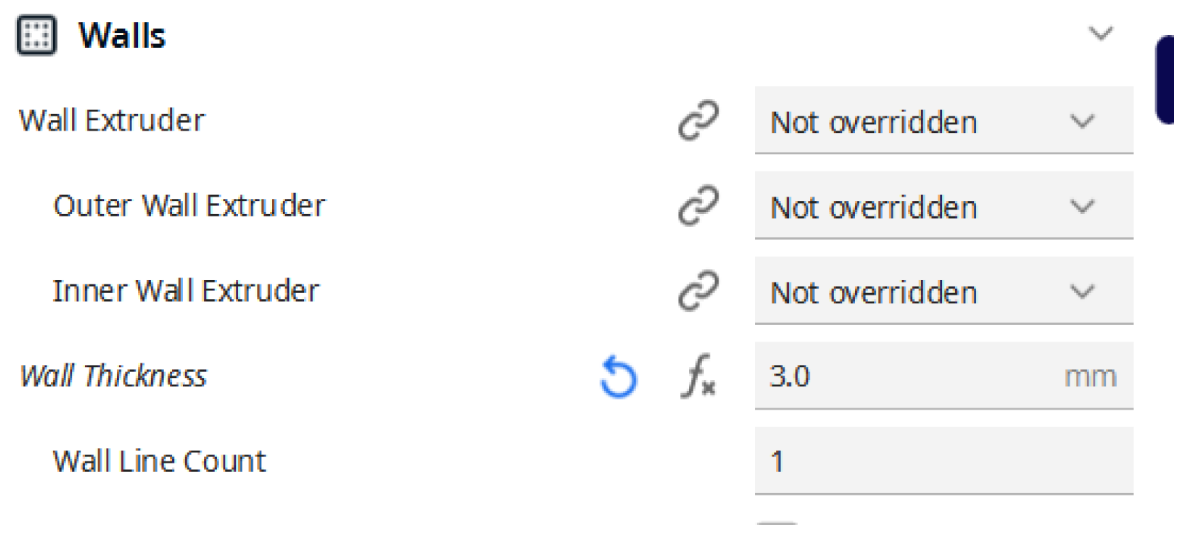


Figura 4.71: Opciones de Paredes (Creación propia).

Para que el *slicer* no genere una cama a la pieza al imprimir, se debe ingresar el valor cero en las opciones que se muestran en la figura 4.72 con esto se asegura que el código G no tome como trayectorias en la parte inferior y superior del cilindro. Esta opción es opcional pero para este estudio para evitar movimientos extras y reducir el gasto de material se excluyo utilizar esta opción.

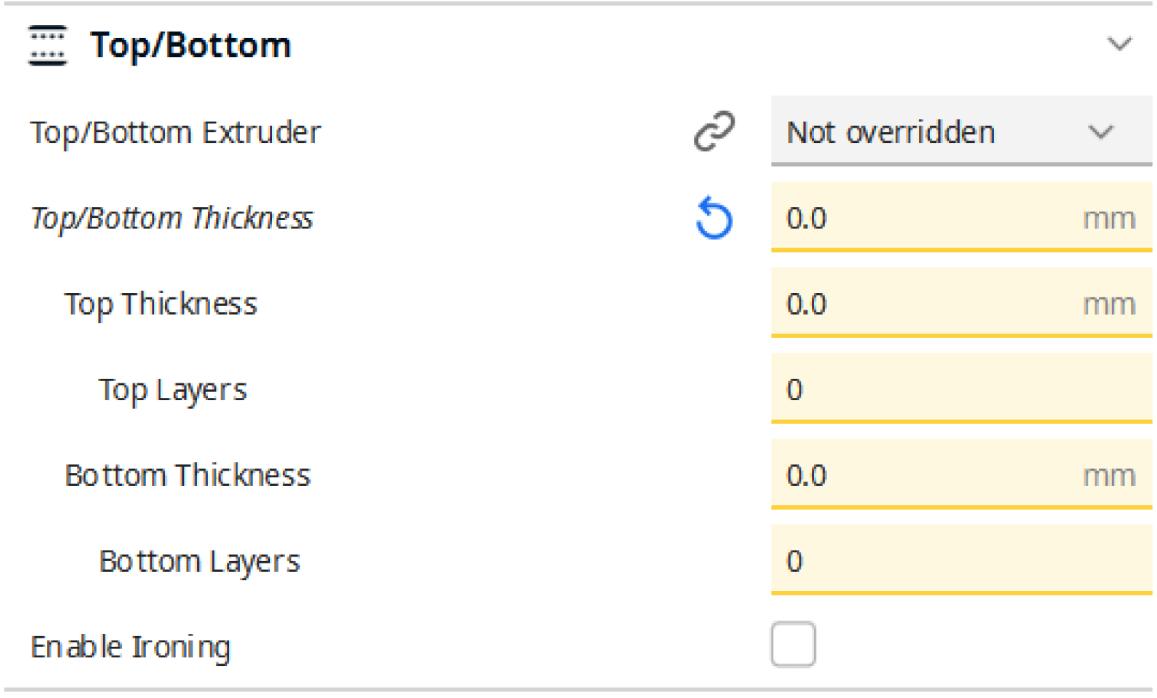


Figura 4.72: Opciones de cama de la pieza (Creación propia).

Para las opciones de velocidad por pruebas de este trabajo se recomienda que sea constante en todas opciones. Y como se muestra en la figura 4.73 se movió a una velocidad de 20 $[mm/s]$, esto porque se obtuvieron resultados con mejor calidad a esa velocidad al realizar distintas pruebas sin afectar en las vibraciones del robot y movimientos bruscos en el sistema para velocidades altas, además de al momento de depositar el material a velocidades altas no se obtiene el ancho de pared deseado y para velocidades bajas el material se acumula.

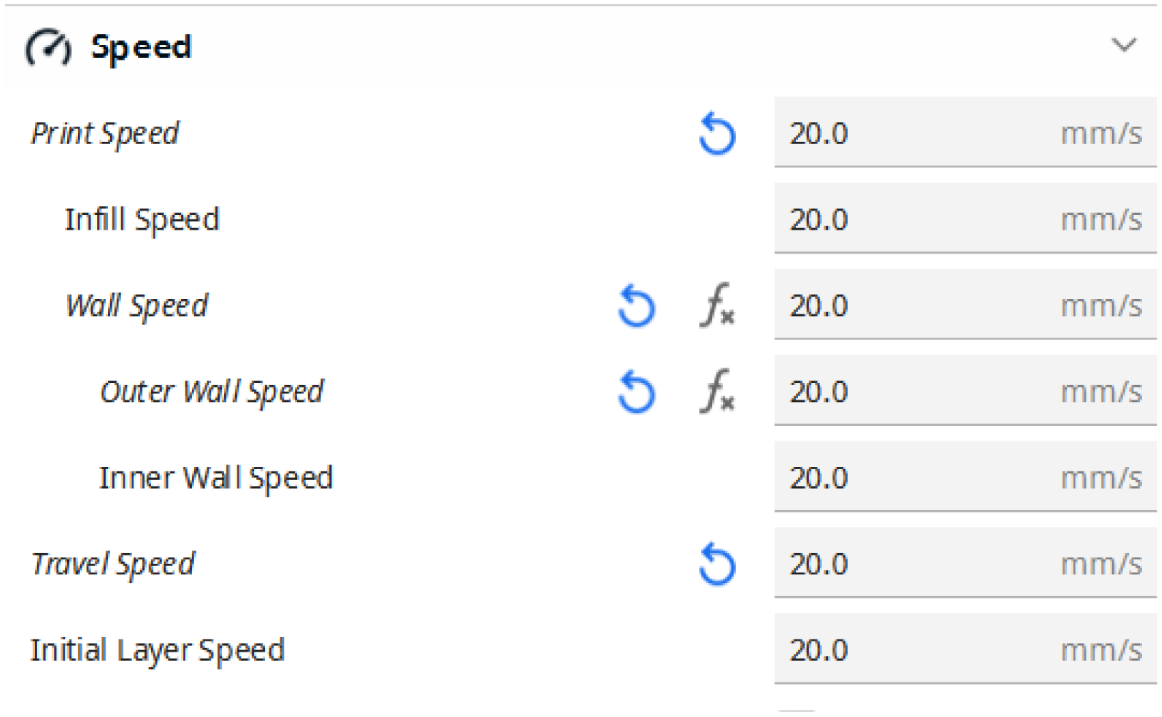


Figura 4.73: Opciones de velocidad (Creación propia).

Finalmente una opción importante es la de *'Spiralize Outer Countour'* como se muestra en la figura 4.74, el cual indica al *slicer* que la impresión es sin retraerse y lo hará de forma de espiral, esto reduce el tiempo a imprimir y se acomoda a las condiciones actuales de trabajo para las pruebas, ya que el flujo de material es continuo. Y la opción de *'Smooth Spiralized Countours* para suavizar los contornos del espiral y obtener mejores resultados al momento de imprimir.

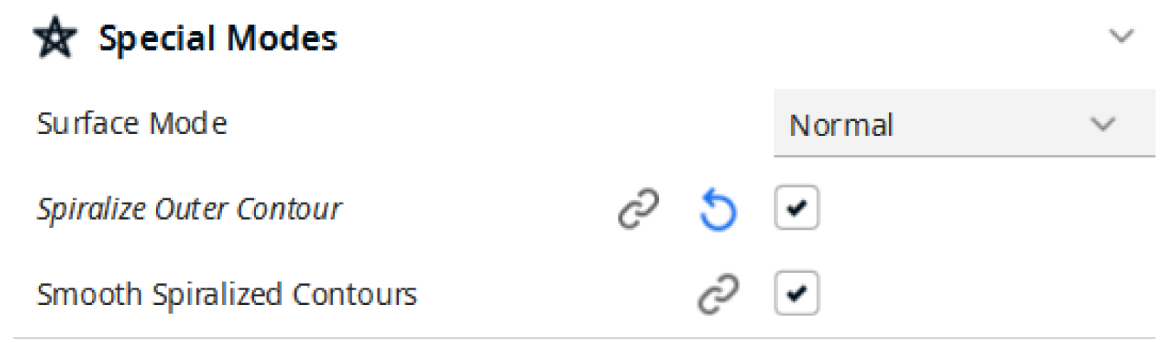


Figura 4.74: Opciones de modos especiales (Creación propia).

Una vez considerado las opciones explicadas anteriormente se puede realizar las rebanadas (*'Slice'*) en el *software* e ir a *'preview'* en la parte superior del interfaz para observar como quedarían las capas para crear en este caso el cilindro como se muestra en la figura 4.75. Se puede observar que se generaron 5 capas como se esperaba en forma de espiral desde la capa inicial hasta la final, también se muestra el tiempo que tomará con los datos seleccionados dando como tiempo 0 [mi], lo que significa que durará en imprimir menos que 1 minuto.

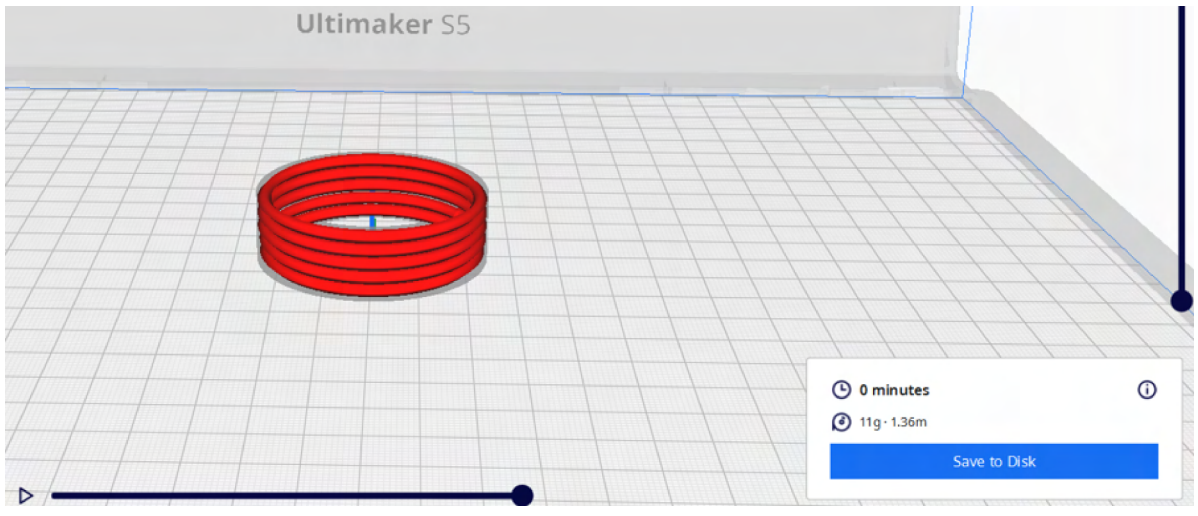


Figura 4.75: Capas generadas por Ultimaker Cura con las opciones anteriores (Creación propia).

Así entonces, se puede exportar el código G en la opción archivo, exportar y procurar indicar en el tipo de archivo sea *'G-code'*.

4.3.5.2 Realizar programa de impresión 3D en RoboDk

La realización del programa en "Robodk" es similar en lo explicado en la sección 4.2.3, pero teniendo en consideración la mesa de trabajo real que se ocupará. Para este estudio como se menciono en las secciones pasadas, se utiliza una mesa caliente de la impresora 3D "Ender" del mismo laboratorio Eco-Fab-lab. Las consideraciones que se deben tomar es:

- Nivelar la mesa: Se debe mover el brazo en cuatro puntos en las esquinas opuestas y acercar de tal manera de que la boquilla del extrusor de alto volumen quede separada de forma que una hoja de papel pueda pasar ajustada.
- Referencia de mesa de trabajo: Luego obtenida la mesa nivelada se debe escoger la parte centrada en donde se moverá el robot en la mesa caliente y estar conectado a "Robodk". Y como se explico en la sección 4.2.3.3 se da click en obtener posición para saber las coordenadas de la mesa caliente respecto del brazo robótico UR5, crear un *'target'* para obtener los valores de las coordenadas y luego vincular esos valores a la referencia de mesa de trabajo (*'Frame'*) como se explico en la misma sección.

Así entonces, se envía el código G, creado en la sección anterior, arrastrándolo a el programa adaptado según las consideraciones mencionadas en los puntos anteriores. Al igual que la creación de programa para dibujos, se abrirá una ventana como se muestra en la figura 4.76. En esta ventana se debe considerar las mismas opciones explicadas en la sección 4.2.3.3 con la diferencia debido a la configuración del sistema de la impresora 3D de alto volumen (Cobot, Extrusor de alto

volumen y brazo compensador) la compensación de la herramienta en este caso debe ser -90° en el eje X y -90° en el eje Y. Además, otra opción extra que se debe considerar son los eventos de programa el cual se utiliza para agregar al código una personalización que se requiera en momentos específicos durante la ejecución de un programa. Estos eventos se pueden utilizar para realizar acciones adicionales, tomar decisiones o interactuar con otros dispositivos o sistemas durante la ejecución del programa.

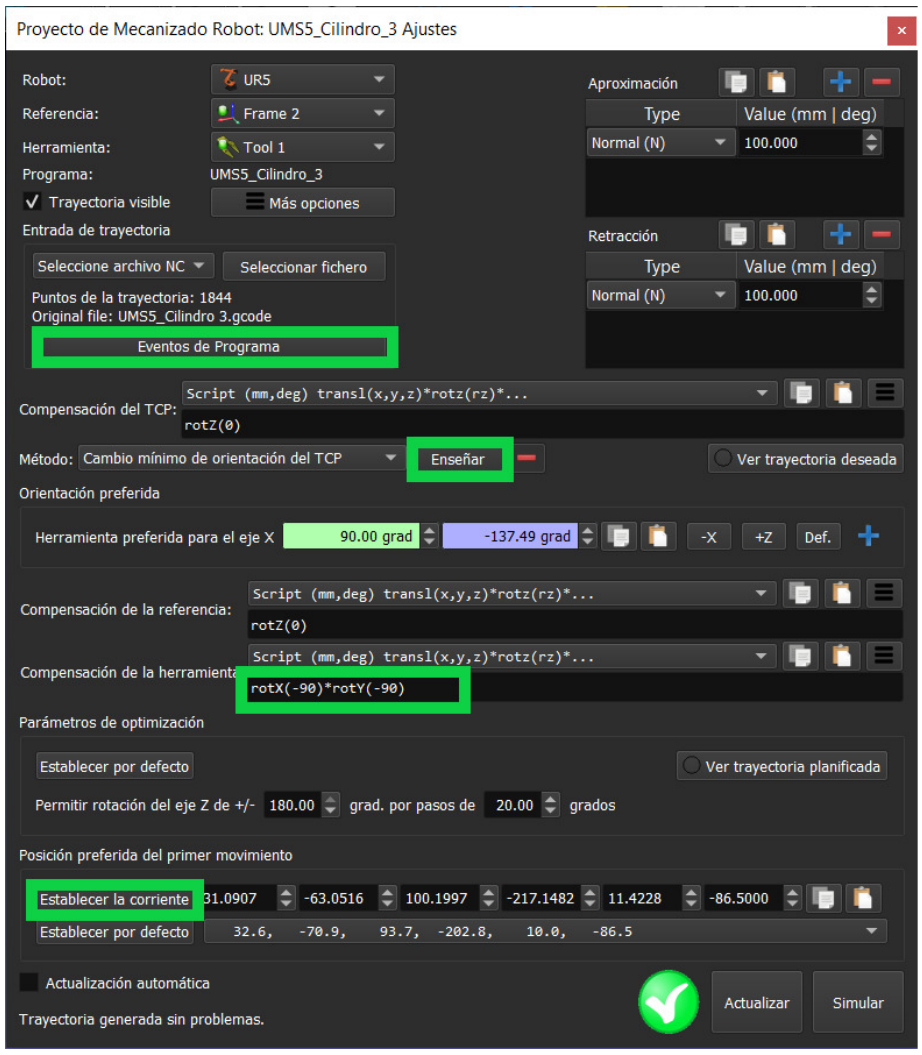


Figura 4.76: Ajustes para programa de impresión 3D (Creación propia).

Para este estudio se tomará en cuenta la velocidad de movimiento rápida como se muestra en la figura 4.77 en $[mm/s]$, esto le indica al robot como moverse en casos que no este ejecutando el programa, es decir se tiene el programa de impresión 3D que inicia en una coordenada, pero el robot se encuentra posicionado en otro espacio, por lo que al momento de ejecutar el programa indicará que se dirija a la coordenada de inicio. Es en este intervalo que el programa por defecto tiene 1000 $[mm/s]$, lo cual moverá al robot muy rápido perjudicando al sistema por lo que se recomienda bajar este valor para este caso a la misma velocidad de impresión que son 20 $[mm/s]$ así entonces cuando se posicione en la coordenada de inicio lo hará a una velocidad discreta.

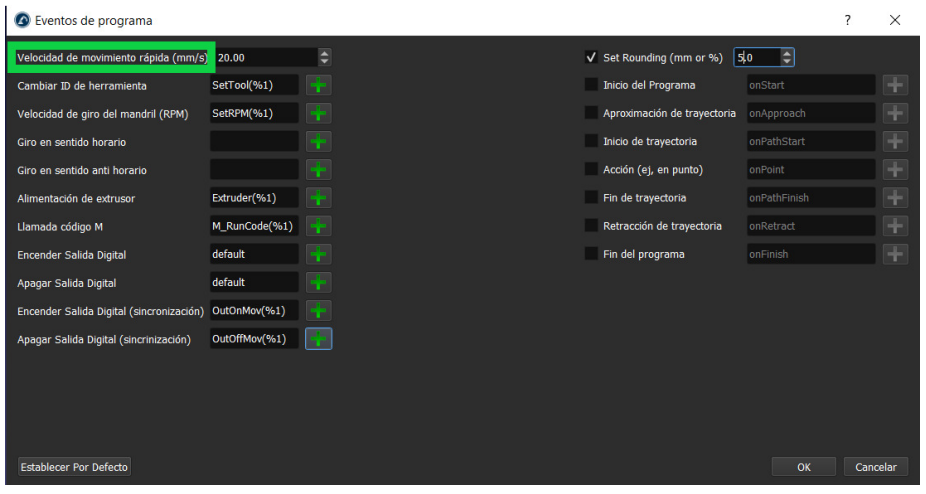


Figura 4.77: Ajustes de eventos de programa de impresión 3D, velocidad máxima (Creación propia).

Teniendo las consideraciones anteriores y verificando que las trayectorias se generaron sin problemas al observar el circulo verde, como se muestra en la figura 4.76 en la parte inferior derecha, se puede pasar a simular y de forma visual verificar que se cumple con las tareas a realizar para la impresión 3D.

Finalmente enviar el programa al robot dando click derecho al programa creado en "Robodk" como se muestra en la figura 4.78, habilitando la opción de "Ejecutar en el robot" y por último habilitar la opción de 'Ejecutar'.

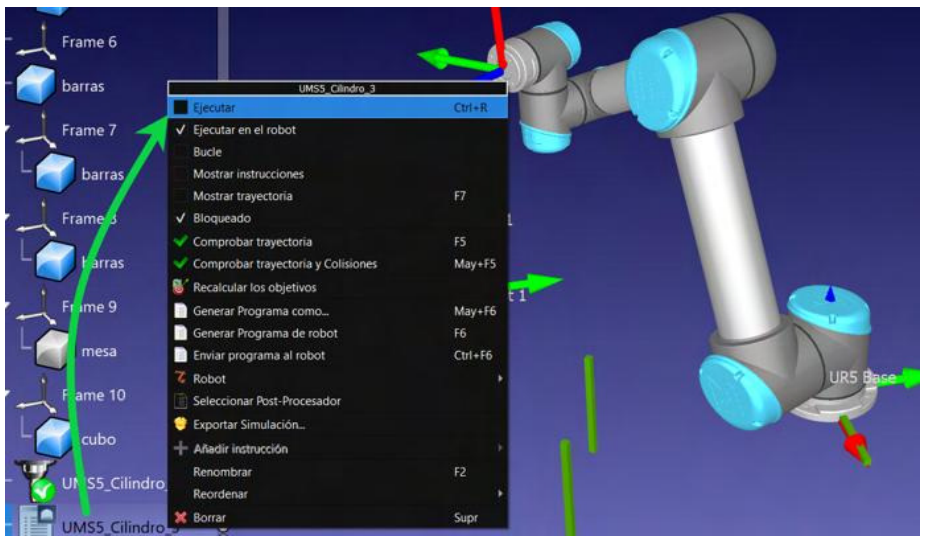


Figura 4.78: Ejecutar programa en el Cobot real (Creación propia).

Se debe tener en cuenta que para modificar las opciones relacionadas a las capas de la impresión, es necesario cambiarlas desde el *slicer* "Ultimaker Cura" y repetir el proceso explicado a lo largo de la sección 4.3.5.

4.3.5.3 Creación de código G con slicer Slic3r

Existe un *slicer* que funciona junto a "RoboDK", el cual se llama "Slic3r". Este es más básico que el "Ulimaker Cura" y más antiguo, ya que su ultima actualización es del año 2015. Pero su ventaja es que se puede modificar las capas directamente desde el "Robodk" sin la necesidad de estar creando un nuevo código G para enviarlo a "Robodk" y repetir el proceso. En la figura 4.79 se muestra el interfaz de este "Slic3r con el cilindro de este estudio.

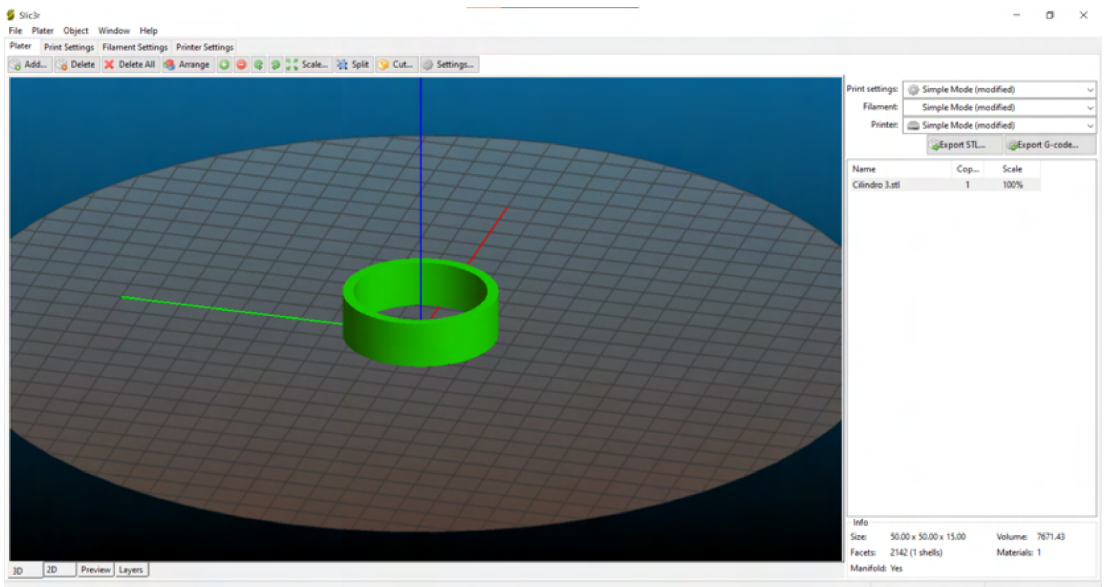


Figura 4.79: Interfaz slic3r (Creación propia).

Para las configuraciones primero se debe ir a configuraciones de impresión (*'Print Settings'*) e ir a *'layers and perimeters'* como se muestra en la figura 4.80 se debe modificar la altura de capa (*'Layer height'*) el cual debe ser como se ha mencionado de 3 [mm] y la altura de la primera capa (*'Initial layer height'*) la cual también debe ser de 3 dependiendo de los resultados en la impresión con el cobot real. Otra opción es *'Spiral vase'* con esto al igual que con "Ultimaker Cura", se realizara el cilindro de forma en espiral, es decir con extrusión del material continuo. Y por último las opciones de *'Top'* y *'Bottom'* en cero, ya que no se requiere que se tenga superficies superiores e inferiores.

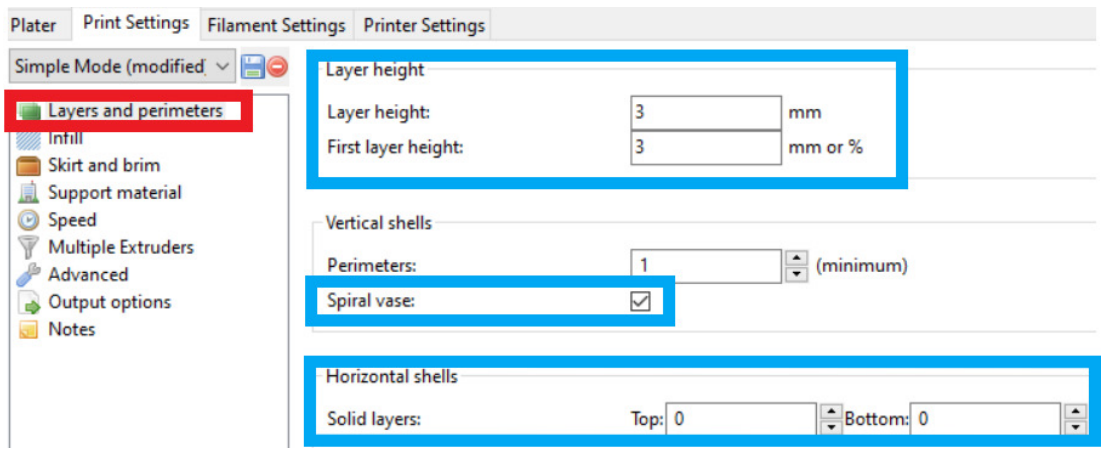


Figura 4.80: opciones de lineas y perímetro (Creación propia).

Normalmente los *slicer* generan una primera capa alrededor de la pieza a imprimir llamada *'skirt'* (falda), esto con el propósito de purgar el material y al momento de realizar la impresión se deposite de manera uniforme. Para este estudio, como se realizaban variadas pruebas y evitar un gasto de material extra se decidió omitir esta opción como se muestra en la figura 4.81 colocando cero en la opción de *'skirt height'* (altura de falda).

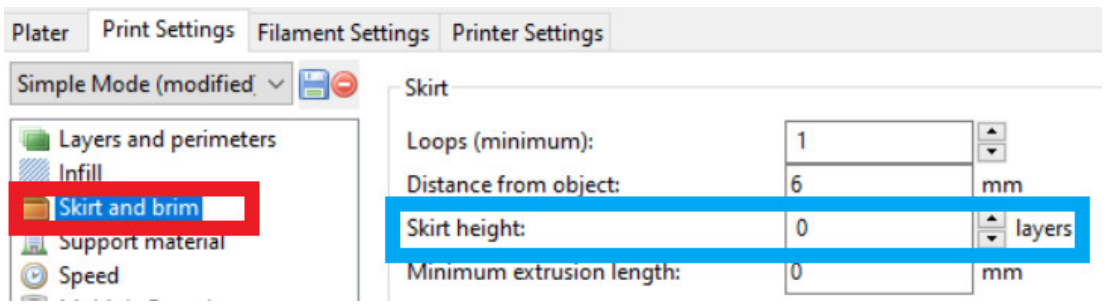


Figura 4.81: opciones falda y ala (Creación propia).

Como última opción en configuraciones de impresión es la velocidad, como se menciono en las secciones anteriores se realizaron pruebas a variadas temperaturas y la velocidad seleccionada es de 20 $[mm/s]$ de manera constante por lo que se deben cambiar a esta velocidad en las opciones de *'speed'* como se muestra en la figura 4.82 mientras que las demás estarán por defecto.

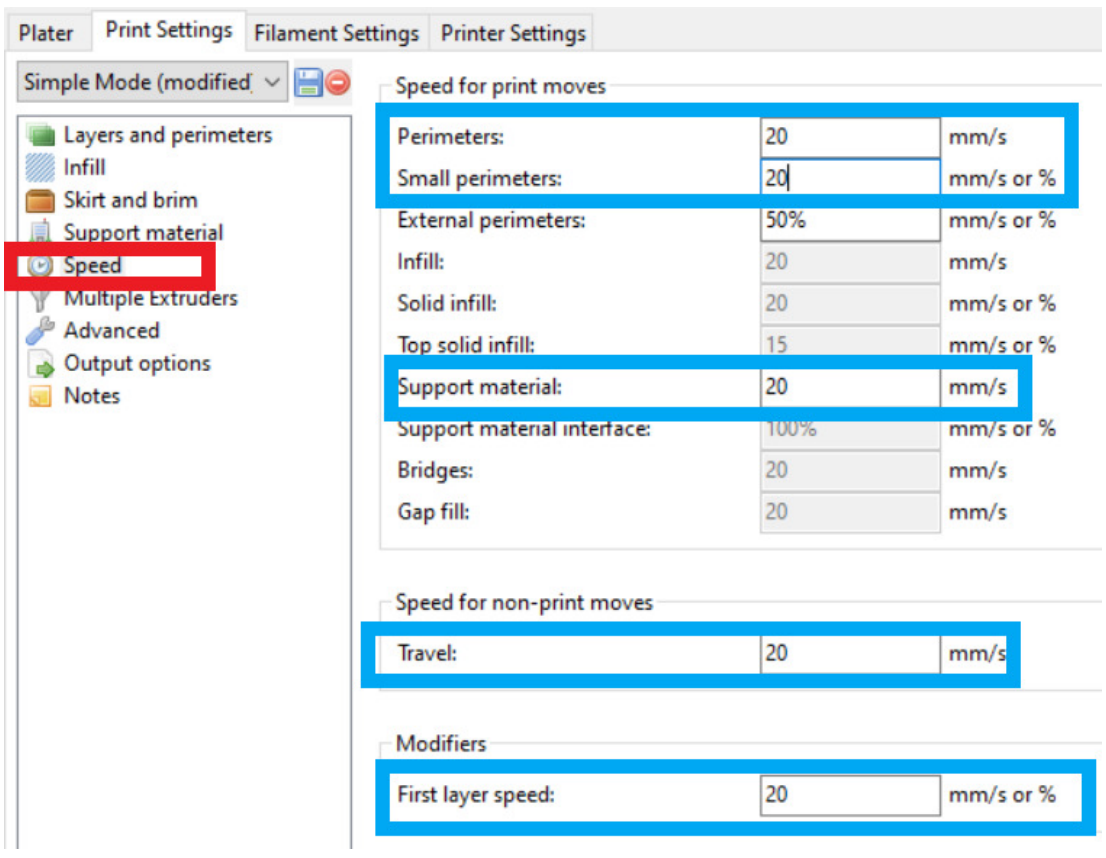


Figura 4.82: opciones de velocidad Slic3r (Creación propia).

Por último ajustar el diámetro de la boquilla en la opción *'Printer Setting'* y en *'Nozzle diameter'* cambiar a 3 [mm].

En la figura 4.83 se muestra las capas creada por el "Slic3r" y al igual que "Ultimaker Cura" y como es de esperar debe generar 5 capas de 3 [mm] para completar la altura de 15 [mm]. Finalmente, se debe exportar en código G para enviarlo a "Robodk" para seguir la configuración de la impresión 3D.

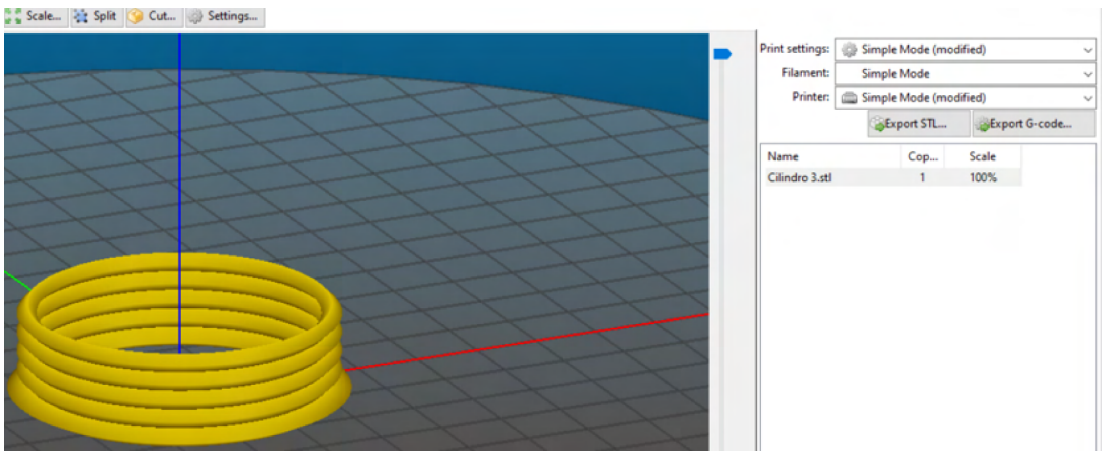


Figura 4.83: Capas creadas por Slic3r para el cilindro de 50 [mm] de diámetro (Creación propia).

Ahora para cargar el programa se hace de la misma manera que lo explicado para Ultimaker Cura teniendo en cuenta las mismas consideraciones de velocidad y ajustes de impresión explicado en la sección 4.3.5.2 con la diferencia que con slic3r se puede modificar desde Robodk las configuraciones de las capas, haciéndolo más cómodo y rápido.

Para poder realizarlo desde "Robodk", es necesario instalar la carpeta de slic3r descargada en la carpeta *'others'* de la carpeta *'Robodk'* ubicada en la carpeta del equipo *'OS (C:)'*. Luego en "Robodk" al instalar el código G creado por "Slic3r" se abre la venta y se puede seleccionar la opción imprimir objeto 3D como se muestra en la figura 4.84.

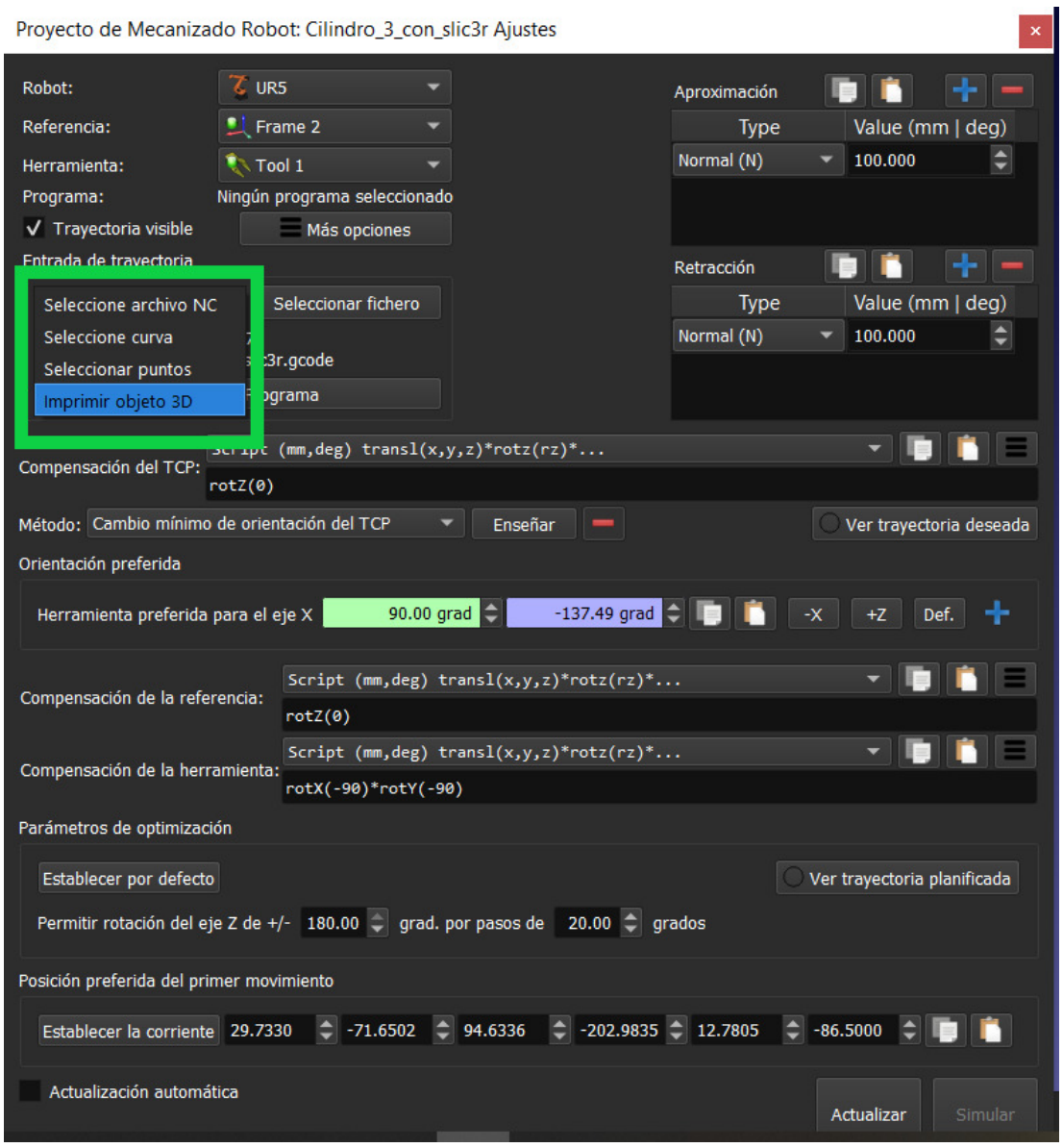


Figura 4.84: Opción para modificar las capas desde Robodk(Creación propia).

Al seleccionar opciones de impresión 3D se abrirá una ventana como se muestra en la figura 4.85, la cual muestra las configuraciones de impresión para ser modificadas directamente de "RoboDK", evitando pasos extras y repetitivos. Al modificar algún valor en las configuraciones se cierra la ventana y ya esta modificado en el código G y basta con actualizar el programa en "Rododk" para utilizar estos nuevos valores en la impresión 3D.

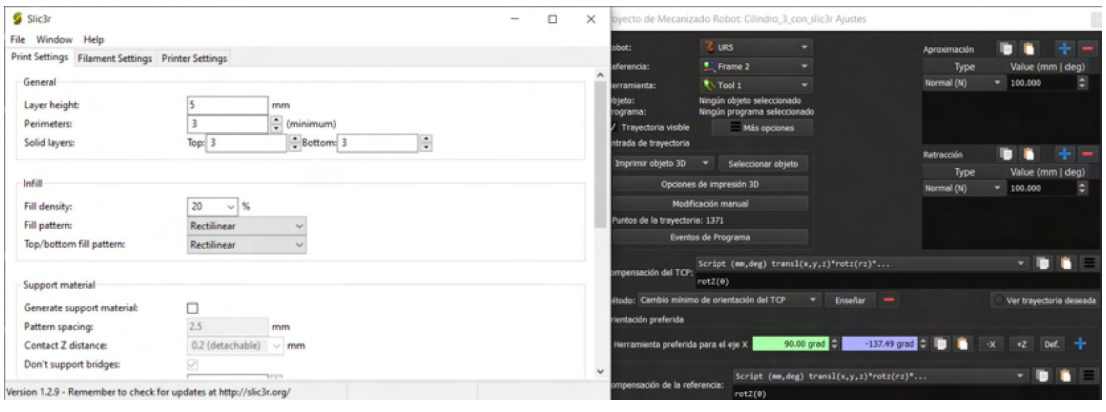


Figura 4.85: Opción para modificar las capas desde Robodk(Creación propia).

La ejecución en el robot es igual a lo explicado para "Ultimaker Cura". La diferencia entre estos

2 *slicers'* es que "Slic3r" trabaja directamente con "Robodk" y hace más rápido los pasos al querer modificar tanto las capas como velocidades de impresión, en cambio "Ultimaker Cura" se necesita cambiar desde el *slicer* para luego repetir el proceso de cargar el programa en "Robodk" y ajustar las condiciones de trabajo. No obstante, la ventaja de este último es que es utilizado en variadas impresoras 3D haciendo que sea mas accesible encontrar recomendaciones de uso, además es más familiar para personas que utilizan impresoras 3D convencionales como la utilizada en el laboratorio y por último que está en constante actualizaciones que se acomodan a las necesidades de los usuarios por lo que "Slic3r" si bien tiene la ventaja de ser más rápido los cambios desde "Robodk", no es tan accesible a encontrar recomendaciones por otros usuarios y al ser un *slicer* que su última actualización fue hace 8 años aproximadamente puede traer consigo bajas en mejoras que se necesitan en la actualidad, ya que la impresión 3D es más accesible que en el año 2015 y por ende las necesidades aumentaron y mutaron por lo que este *slicer* para piezas más complejas podría no ser suficiente.

Capítulo 5

Resultados

5.1 Movimientos del robot con solución en la conexión Cobot-Extrusor-Brazo compensador

Al inicio se utilizó una conexión directa del sistema (Cobot, Extrusor de alto volumen y brazo compensador) donde los movimiento permisible horizontales eran mínimos debido a que el robot sigue una trayectoria recta mientras que el brazo compensador una trayectoria en forma de arco, esto sumado a que el cabezal de este último es fijo no permite mantener la posición de la muñeca 3 del cobot y por ende existe parada de emergencia. La solución fue crear un juego entre el brazo y la conexión con los otros 2 componentes (extrusor de alto volumen y brazo compensador) permitiendo liberar el movimiento del cabezal del brazo compensador al mover el brazo robótico horizontalmente. Esta conexión fue sencilla y de bajo costo el cuál tuvo resultados positivos para la impresión 3D con las restricciones de trabajo explicadas. Lo que se debe considerar es que como existe un juego y es una bisagra común y corriente, es decir que no ésta hecha para este fin permite movimientos verticales y horizontales por su mismo juego por lo que genera vibraciones que podrían ser considerables a altas velocidades. No obstante, a las condiciones que se trabajo en este estudio el funcionamiento fue óptimo permitiendo obtener resultados favorables. Aún así para futuros trabajos se debe mejorar la conexión con componentes más apropiados y de forma que se pueda ocupar de manera óptimo los movimientos del brazo robótico UR5 cambiando algunas configuraciones como se dará a conocer en recomendaciones más adelante en este estudio.

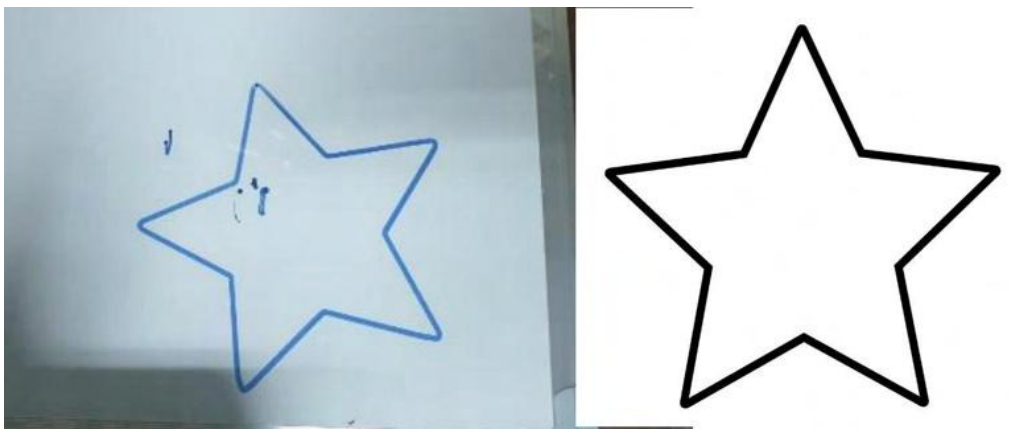
5.2 Dibujos creados por cobot a partir de programa en RoboDk con plumón de pizarra como herramienta

En la figura 5.1 se muestra el resultado del dibujo explicado en la sección 4.2.3, en el cual se puede observar, a partir de la comparativa con la imagen digital utilizada para la creación de vectores para realizar el código G, que son muy similares y que el cobot siguió a la perfección las lineas de la imagen. Sin embargo, se observa que las lineas son más toscas debido a que el robot sólo siguió la trayectoria más no el grosor de las lineas en ciertas partes del dibujo. No obstante, los resultados indican que el cobot es capaz de seguir lineas complejas a pequeñas distancias, ya que la pizarra utilizada tiene una dimensión de 250 [mm] de alto y ancho de 350 [mm], y se dibujo en una sección de aproximadamente 219 [cm^2]. Además, cabe mencionar que la herramienta utilizada para hacer las pruebas fue un plumón pegado con cinta, ajustado lo más posible y por ende sólo siguió la trayectoria sin tomar en cuenta profundidades.

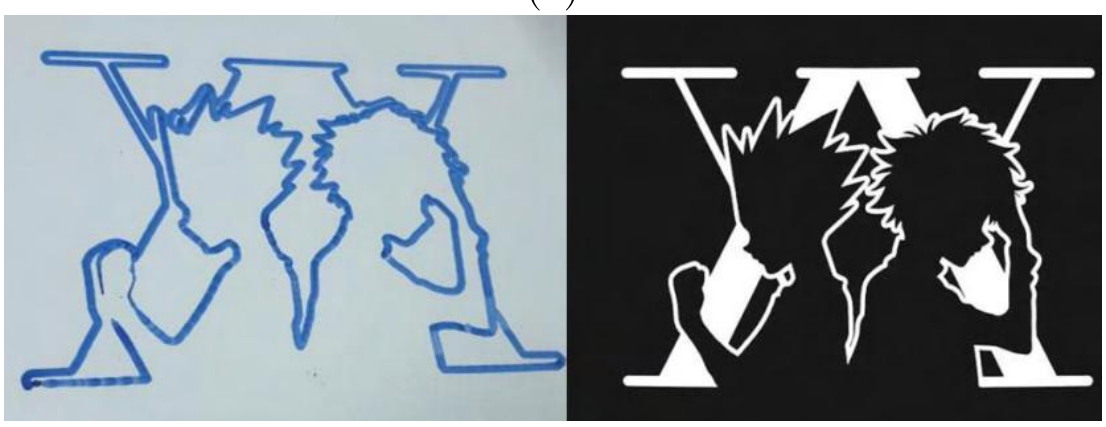


Figura 5.1: Dibujo realizado por Cobot y comparativa con imagen digital(Creación propia).

En la figura 5.2 se muestran otras pruebas de dibujos, la (a) de una estrella que como se observa la precisión de las rectas y curvas son de excelente calidad. Mientras que la imagen (b) si bien es muy parecida a la imagen digital tiene cruces de líneas debido a que el plumón estaba muy cerca de la pizarra y las líneas obtenidas eran gruesas, por lo que para espacios pequeños se juntaban estas trayectorias, no obstante los resultados siguen siendo muy parecidos.



(a)



(b)

Figura 5.2: Otros dibujos creados por cobot y comparativa con la imagen digital

Para los resultados obtenidos al dibujar con el brazo robótico UR5, se puede analizar que con unos pequeños ajustes se pueden obtener resultados más precisos, pero la idea general es que las trayectorias almacenadas en un código G y luego ser interpretada por el *software* "RoboDK" para hacer el enlace con el cobot real son de alta confiabilidad, por lo que estas pruebas son satisfactorias para pasar al siguiente paso que es crear una pieza en tres dimensiones. Ya que en 2 dimensiones como lo son los dibujos, se puede interpretar como una capa de una impresión 3D, siendo así una cantidad finita de dibujos 2D sobrepuestos uno encima de otro para crear una impresión 3D. Lo anterior es un análisis a partir de los resultados, pero como se explico en la sección 4.3.5 los programas utilizados son distintos salvo el *software* "Robodk" para crear el código G, pero la idea central es la misma.

5.3 Impresión de Cilindro de diámetro 50 [mm] a partir de programa en RoboDK con extrusor de alto volumen y brazo compensador

Una vez obtenido resultados positivos del manejo del robot para seguir trayectorias por programas creadas por el *software* "RoboDk" a partir de un código G, se puede pasar a la impresión 3D. Para

este estudio como se ha mencionado en reiteradas secciones se utiliza un extrusor de alto volumen. Para la primera prueba se utilizo *pellets* de plástico comercial para probar tanto el funcionamiento del extrusor como los movimientos del cobot y como se muestra en la figura 5.3 se obtuvo un resultado no muy convincente, pero viene dado por ser la primera prueba y reajustar valores de velocidad, tanto del robot como del extrusor (rpm), como alturas de capas y temperatura en la boquilla principalmente. Para esta prueba se coloco un *'bottom'* en el *slicer*, lo que genera una superficie en la parte inferior del cilindro, el cual se decidió quitar ya que genera un malgasto de plástico para las pruebas y evita observar con más detalles las capas.



Figura 5.3: Primera impresión con pellets(Creación propia).

Como el enfoque principal del laboratorio Eco-Fab-Lab es utilizar plástico reciclado se decidió que las siguientes pruebas fueran con plástico triturado, que para este estudio son las tapas de las botellas que son principalmente de material PLA utilizada en impresoras 3D convencionales. Para este tipo de material se utilizó temperaturas en el rango de 195°C a 120°C, estas variaciones fue debido a que el triturado no era uniforme y como es de esperar el flujo no es continuo, por lo que se iba variando la temperatura para obtener un flujo mas uniforme, además que el extrusor tiene un rango de error entre la temperatura deseada y la temperatura real, ya que no es constante y fluctúa. Como se muestra en la figura 5.4 se ven distintos cilindros impresos con triturado de tapas de botella, los cuales no tienen la forma deseada la que es de un cilindro y más bien el material se empieza a recoger hacia al centro creando una especie de cono.



Figura 5.4: Primeras Impresiones con plástico reciclado triturado (Creación propia).

En la figura 5.5 se muestran variadas pruebas pero con 2 vueltas del cilindro, es decir 2 capas. Esto debido a que se quería encontrar la velocidad en el cual el diámetro del filamento fuera lo más constante posible y sin ser alterado por la velocidad del cobot. Dando como resultado una velocidad para este estudio de 20 $[mm/s]$. Además se puede observar que los filamentos tienen relieves, esto es debido a que como se mencionó el flujo no es constante por ende crea que se agrupe el material en ciertas secciones.



Figura 5.5: Pruebas de impresión para encontrar la velocidad óptima (Creación propia).

En la figura 5.6 se muestra un buen resultado de filamento y deposición del material, creando capas que visiblemente se ven continuas. No obstante también se observa que en una sección del

cilindro el material no se está depositando del todo bien, esto es debido a que la mesa no esta bien nivelada. Es decir, que las trayectorias horizontales del extruder no son paralelas a la mesa caliente de la impresora 3D Ender por lo que crea que el material se deposite de manera incorrecta. Eso sumándolo al flujo no constante de material, que genera variaciones en el diámetro del filamento. Otro punto importante es que se encontró que en todas las piezas impresas, el cilindro tendía a hundirse creando una especie de cono como se menciono anteriormente, por lo que se hizo un análisis y se determino que debido al alto flujo de material depositado encima de otro y a una alta temperatura crea, que por el peso y ser más moldeable, empuja a la capa inferior desplazándose hacia el centro. Es por esto y por recomendaciones de personas que utilizan impresoras 3D, se informo que comúnmente traen ventiladores que van enfriando el filamento depositado y así asegurando que éste quede fijo en su posición con una dureza más alta.



Figura 5.6: Impresión del cilindro con mesa desnivelada y sin inyección de aire (Creación propia).

Así entonces tomando en cuenta los análisis anteriores y con la ayuda de una pistola de calor, que para este caso solo se inyecto aire a temperatura ambiente, se fue siguiendo la ruta manualmente de las capas inyectando aire al filamento depositado. Teniendo como resultado como se muestra en la figura 5.7 un cilindro como se esperaba, si bien no esta del todo formado como tal, se corrigió el hundimiento del material con gran éxito. Lo que indica que para futuras pruebas y

trabajos se considere la utilización de inyección de aire en el sistema. Se corrigió en gran medida la desnivelación de la mesa pero aún asi falta por corregir. Para esto se debe ajustar el sistema (Cobot, Extrusor de alto volumen y mesa caliente), ya que se ajustó lo más que se pudo pero por las condiciones del mismo no se consiguió mejor resultado. Como se muestra en figura 5.8 la altura es aproximadamente 13 [mm], lo que viene dado debido a que por falta de flujo de material se detuvo antes la impresión, pero las alturas de capas se obtuvieron resultados cercanos a los 3 [mm] como se esperaba en la teoría. Así entonces, el flujo también es un factor importante en los resultados, por lo que se debe considerar para futuros trabajos conseguir que sea más constante.



Figura 5.7: Impresión del cilindro con inyección de aire (Creación propia).



(Altura aproximada de 13 [mm])



(Espesor aproximado de 3 [mm])

Figura 5.8: Medidas de altura y espesor de la impresión de cilindro con inyección de aire

Como análisis general de los resultados obtenidos en la impresión 3D, se puede decir que si bien falta por hacer, los resultados hasta este camino son de gran valor, debido a que en cada prueba se encontraron resultados esperados y con oportunidad de mejora. En donde primero fue deshacer la superficie, para luego pasar a las pruebas de velocidad con el triturado de plástico encontrando las velocidades y temperaturas de trabajo adecuadas para después encontrar la importancia de la nivelación de la mesa y por último encontrar que para obtener un resultado con mayor calidad se debe utilizar una inyección de aire para enfriar el filamento depositado para hacerlo más firme y no se hunda.

Otro punto importante pero no mencionado es el error que se tiene al utilizar una bisagra en la conexión entre el cobot y el extrusor. Esto genera que los movimientos del robot no se traspasen como tal a la boquilla, ya que existe un juego entre ambos debido a las restricciones de movimiento del brazo compensador y más enfocado al cabezal de éste que es rígido y por ende sigue la trayectoria de arco, no permitiendo que la herramienta se quede fija en una posición al momento de imprimir. No obstante, como se observo en los resultados este error no es tan considerable ya que se tienen impresiones muy cercanas a lo deseado.

Capítulo 6

Conclusiones y Recomendaciones

Al termino de este estudio y escritura del trabajo de titulo se lograron los objetivos propuestos al inicio de éste, donde al comienzo con las partes divididas pero con una idea general se alcanzó llegar a una propuesta inicial que se acomodará a los requerimientos preliminares, a través de la investigación, pruebas, errores en el camino y ayuda de terceros que no tenían un conocimiento sobre lo que se quería lograr, pero si el entendimiento sobre las distintas partes por separado, a lo cual se convergió a lo que se trabajó durante este estudio.

En primera instancia, a través de pruebas con el cobot se verificó la gran precisión que pueden llegar a tener estos tipos de brazos colaborativos, donde pueden seguir una trayectoria casi a la perfección con mínimos errores, cabe mencionar que estos errores no sólo dependen del robot, ya que dependerá del usuario si tiene el conocimiento de indicar lo que realmente quiere al robot como también el espacio de trabajo como el suelo, de como esta sujeto a este, si existe mucho juego que cause vibraciones y si hay fuerzas externas que empujen al robot, lo que causará problemáticas al resultado. En está etapa se llegaron a conclusiones favorables, debido a que el UR5 seguía trayectorias complejas y de manera repetitiva.

El *software* "RoboDk" fue de gran utilidad en este estudio ya que fue pilar importante en la creación de los movimientos del cobot para crear el cilindro modelado, donde en paralelo también se utilizó *slicers* que funcionaban como comunicador que transformaba el modelo a un archivo de corte llamado G-code en el cual indica como debe moverse el brazo robótico (altura inicial, altura de capa, soportes, relleno, velocidad, etc). Como objetivo fue estudiar estos *slicer* y buscar que puntos son relevantes al momento de imprimir y sobre todo que influyen a un movimiento del cobot, donde se hicieron variadas pruebas hasta llegar a valores donde los resultados fueron cada vez más similares a lo que se quería obtener, en este caso un cilindro. Estos resultados fueron de ayuda para orientar el trabajo a las necesidades reales del sistema de impresión en el laboratorio, además

con cada prueba se fueron obteniendo datos duros que van a hacer la base a futuros trabajos en la impresión 3D en el laboratorio ECO-FAB-LAB.

Un punto importante es el TCP o PCH, que le indica tanto al cobot como al *software* "RoboDk" a que distancia se encuentra el punto central de la herramienta, tomando como referencia la base del brazo robótico, siendo este la boquilla del extrusor de alto volumen del laboratorio. Para esto, no fue posible ocupar el mismo cerebro del UR5 para que se auto-calcule por lo que se realizó de manera manual e intentado que un eje no varié para no tener una medición adicional, por lo que se debe tener en cuenta los errores de medición y juego en las uniones que pueden afectar en los resultados finales en la impresión, no obstante si bien existen los pequeños errores anteriores, tanto las pruebas como movimientos fueron de una precisión aceptable tomando en cuenta las distintas restricciones del sistema.

La unión del brazo compensador al sistema resulto ser uno de los otros puntos complejos durante el estudio, ya que como se explico este seguía una trayectoria distinta a lo que el robot se mueve. Por lo que la solución con la bisagra fue una manera rápida y barata de salir del problema, sin embargo debe ser temporal, debido a que se debe tener una conexión que disminuya el juego que hasta ahora existe y que trae consigo que el TCP varié mientras el cobot se mueve.

Finalmente, analizando el estudio en este trabajo se puede concluir que como sistema general es factible realizar una impresora 3D de alto volumen con un cobot (UR5), en el cual con la ayuda de *software* "Robodk" y *slicers* se pueden crear trayectorias digitales para ser ejecutadas en el brazo robótico real. Se debe tener en cuenta los comentarios anteriores, los cuales indican que puntos son relevantes tener un cuidado, como lo es el TCP y la unión del brazo compensador al sistema cobot-extrusor de alto volumen. Además de estudiar soluciones al flujo del material triturado, ya que influye directamente a la impresión 3D, como lo es la altura de las capas. Donde un bajo flujo hará que la altura de capa sean menores y un alto flujo puede causar que el material se aglomere en un sitio afectando las dimensiones de la pieza impresa.

Como recomendación al sistema general, al realizar este estudio y al hacer variadas pruebas, se puede recomendar la siguiente configuración como se observa en la figura 6.1. Esta configuración tiene como misión disminuir la variación de TCP, ya que como se observa, sólo se tomaría el eje Z como distancia 'd', mientras que los ejes X e Y estarían centrados al extrusor de alto volumen. Además permitiría que el cobot tenga más grados de libertad debido a que estaría paralelo al brazo compensador disminuyendo las restricciones de movimiento con éste. Los cuadrados rojos indican las uniones, donde para el robot-extrusor la recomendación en lo posible es que tenga un movimiento permisible giratorio en el eje Z, ya sea con algún tipo de rodamiento u otro componente que permita esta rotación, debido a que puede tener fuerzas que influyan al brazo robótico a

detenerse. Mientras que para la unión brazo compensador-extrusor se recomienda utilizar un cabezal articulado, como se muestra en la figura 6.2 de la misma empresa que crea estos tipos de brazo compensadores, que funcione como muñeca y teniendo como objetivo que el cabezal pueda seguir las trayectorias del cobot sin restringir los movimientos del sistema. Esta recomendación puede llevar más tiempo en concretar debido a distintos factores como lo es la instalación del robot en el techo o en una estructura en altura y los costos que estas traen, como la compra de los materiales para la estructura y creación de unión cobot-extrusor, como también el costo del cabezal para el brazo compensador.

Otra recomendación que resultó ser muy favorable en los resultados, es para futuros trabajos incorporar la inyección de aire en la base del extrusor de manera que el material fundido depositado sea enfriado y por ende la dureza aumente, con el fin que el peso de la capa superior no aplaste a las capas inferiores y termine afectando a las dimensiones de la pieza deseada.

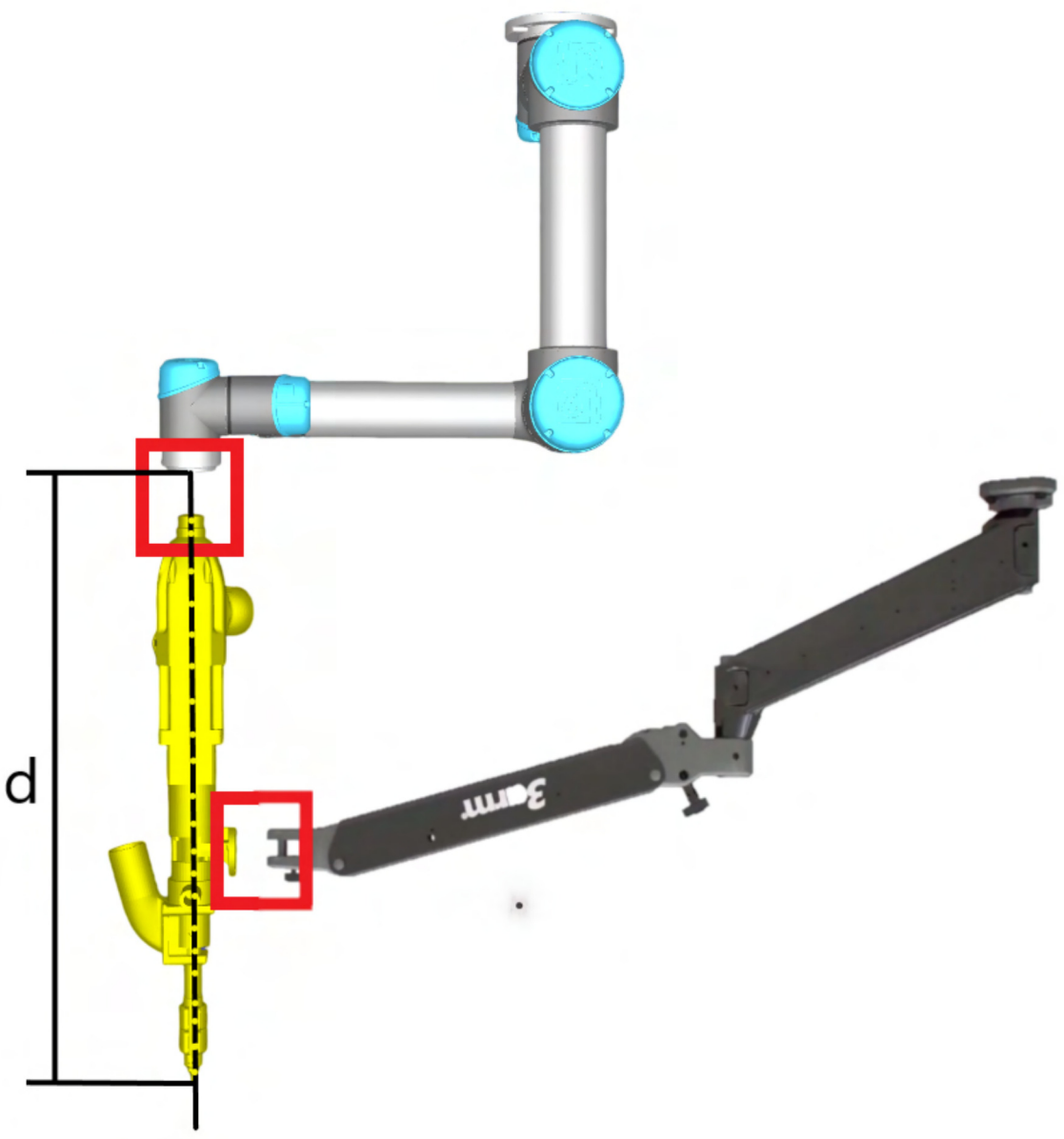


Figura 6.1: Configuración recomendada del sistema (creación propia).

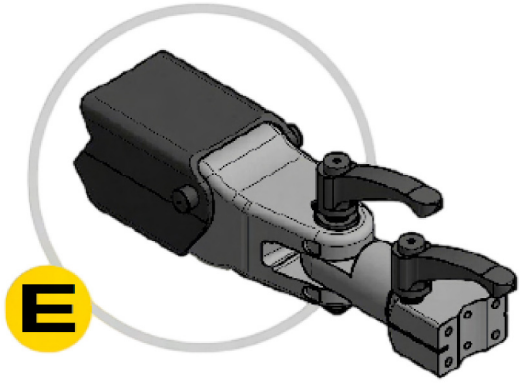


Figura 6.2: Cabezal articulado plano 3arm [25].

Referencias

[1] Cade Cobots. Fabricación aditiva (fa) integrada en soluciones de robótica colaborativa. Recuperado 27 de agosto de 2023 a `https://l1nq.com/9ft05`, agosto 2021.

[2] KUKA. Fabricación aditiva e impresión 3d industrial. Recuperado 27 de agosto de 2023 a `https://l1nq.com/tWmvv`, noviembre 2018.

[3] Hsiao T. Reyes-Uquillas D. Compliant human–robot collaboration with accurate path-tracking ability for a robot manipulator. Recuperado 28 de agosto de 2023 a `https://doi.org/10.3390/app11135914`, julio 2021.

[4] L Ken Keys. System life cycle engineering and df'x'. *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, 13(1):83–93, 1990.

[5] Lira I Sánchez N. La manufactura aditiva como potencializador de los sitemas productivos. Recuperado 28 de agosto de 2023 a `https://encr.pw/ak4sp`, mayo 2020.

[6] Luis LLamas. Qué es el slicer y cuál es su función en la impresión 3d. Recuperado 28 de agosto de 2023 a `https://l1nq.com/Jgmwf`, noviembre 2019.

[7] Mikell Groover and EWJR Zimmers. *CAD/CAM: computer-aided design and manufacturing*. Pearson Education, 1983.

[8] Microsoft Corporation. Slicer (segmentación de datos). Recuperado 29 de agosto de 2023 a `https://encr.pw/P235N`, septiembre 2019.

[9] Ian Gibson, David W Rosen, Brent Stucker, Mahyar Khorasani, David Rosen, Brent Stucker, and Mahyar Khorasani. *Additive manufacturing technologies*, volume 17. Springer, 2021.

[10] Romero E. Slicers o laminadores más usados en impresión 3d. Recuperado 29 de agosto de 2023 a `https://l1nq.com/lCUtd`, Marzo 2023.

[11] descubrearduinocom. Slicer 3d: los mejores softwares de corte para impresión 3d. Recuperado 29 de agosto de 2023 a `https://encr.pw/6QIGP`, abril 2021.

[12] Deon de Beer Andrew C. Development of a stereolithography (stl) slicing and g-code generation algorithm for an entry level 3-d printer. Recuperado 29 de agosto de 2023 a `https://l1nq.com/sPyEM`, diciembre 2018.

[13] CNC Kitchen. G-code basics for 3d printing. Recuperado 29 de agosto de 2023 a `https://l1nq.com/eBqOQ`, agosto 2021.

[14] Peshkin M. Colgate J. Cobots, industrial robot: An international journal. Recuperado 30 de agosto de 2023 a `https://l1nq.com/CSN2D`, 1999.

[15] R. Osorio-Comparán G. Lefranc, I. López and M. Peña. Cobots in automation and at home. Recuperado 30 de agosto de 2023 a `https://l1nq.com/YEULR`, 2020.

[16] Shimon Y Nof. *Handbook of industrial robotics*. John Wiley & Sons, 1999.

[17] A. Garbev and A. Atanassov. Comparative analysis of robodk and robot operating system for solving diagnostics tasks in off-line programming. Recuperado 30 de agosto de 2023 a `https://encr.pw/RI7Mf`, 2020.

[18] Van J. Mechanical advantge. Recuperado 20 de agosto de 2022 a `https://encr.pw/LUJ0a`, diciembre 1996.

[19] Institute of Robotic and Mechatronics. History of the dlr lwr. Recuperado 20 de agosto de 2022 a `https://l1nq.com/bOpDr`, diciembre 2021.

[20] Universal Robots. How universal robots sold the first cobot. Recuperado 20 de agosto de 2022 a `https://l1nq.com/bIFof`, diciembre 2021.

[21] Universal Robots. Manual de usuario ur5/cb3 versión 3.2. Recuperado 20 de agosto de 2022 a `https://encr.pw/pFIX7`, abril 2016.

[22] Universal Robots. Ficha técnica ur5/cb3. Recuperado 20 de agosto de 2022 a `https://l1nq.com/ma92s`, abril 2017.

[23] Robotiq. Getting started with collaborative robots. Recuperado 20 de agosto de 2022 a `https://encr.pw/ZZegB`, octubre 2018.

[24] BRITISH STANDARD and BSEN ISO. Safety of machinery—safety-related parts of control systems—. 2003.

[25] 3arm. Brazo compensador serie 3. Recuperado 17 de junio de 2023 a `https://l1nq.com/67oEL`, abril 2019.

Capítulo 7

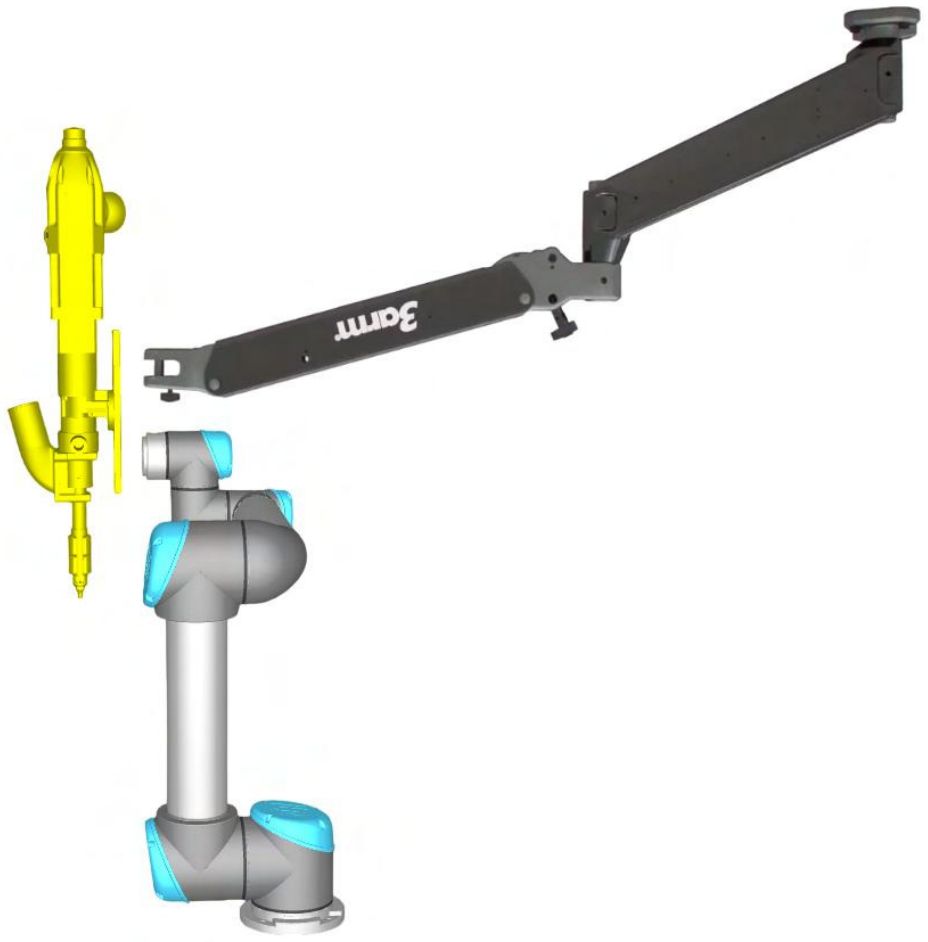
Anexo

7.1 Guía de uso y mantenimiento del sistema

El siguiente anexo, es la guía de uso y mantenimiento para el correcto uso del sistema versión 1, para el laboratorio ECO-FAB-LAB, de la carrera Ingeniería civil Mecánica de la Universidad Técnica Federico Santa María. Se incorporó como anexo debido a que la mayoría de los datos fueron expuestos en este trabajo y se comprimió de manera que sirva como manual para futuros estudiantes que trabajen en el laboratorio.

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INGENIERÍA CIVIL MECÁNICA



GUÍA DE USO Y MANTENIMIENTO PARA EL CORRECTO USO DEL SISTEMA. VERSIÓN 1

ECO-FAB-LAB

MEMORISTA ANGEL VALDÉS AZÓCAR

PROFESOR GUÍA: UDO RHEINSCHMIDT

2023

Tabla de Contenidos

1 Guía de uso del Sistema **1**

1.1 Encendido brazo robótico UR5 1
1.2 TCP o PCH 4
1.3 Tipos de movimiento del Cobot (UR5) 7
1.4 Guardado 8
1.5 Conectar PC con UR5 9
1.6 RoboDK 11
1.6.1 Conexión de Robodk a UR5 11
1.7 Impresión 3D 14
1.7.1 Configuración actual Cobot-Extruder 15
1.7.2 TCP o PCH del Extruder de alto volumen junto con la unión 15
1.7.3 Creación de programa para impresión 3D 17
1.7.4 Creación de código G con slicer Ultimaker Cura 18
1.7.5 Realizar programa de impresión 3D en RoboDk 23
1.7.6 Creación de código G con slicer Slic3r 29

2 Mantenimiento del Sistema **34**

2.1 Universal robot UR5 34
2.1.1 Especificaciones 35
2.1.2 Montaje 36
2.1.3 Herramienta 36
2.2 Sistema 37
2.2.1 Unión Sistema 37
2.2.2 Refrigeración de extruder de alto volumen 38
2.2.3 Mover UR5 con PCH del Extruder 38
2.2.4 Mover UR5 libre 39
2.2.5 Importancia de simular antes de ejecutar 40
2.2.6 Velocidades altas en secciones pequeñas 40

2.3 Actualizaciones Softwares 40
2.3.1 Actualización software Universal Robots 5 40
2.3.2 Actualización software Robodk y licencia educacional 42
2.3.3 Actualización softwares slicers 43

Capítulo 1

Guía de uso del Sistema

1.1 Encendido brazo robótico UR5

Para controlar el robot se tiene una pantalla táctil como se muestra en la figura 1.1, esta consta de dos botones frontales uno de parada de emergencia de color rojo el cual tiene como función parar el robot si hay un mal funcionamiento o si por alguna razón se activa la parada de emergencia poder desactivarla una vez solucionado el problema. El segundo botón de color gris y luz es el encendido de pantalla y poder así controlar el robot, también funciona para apagar el robot si la pantalla táctil no responde.

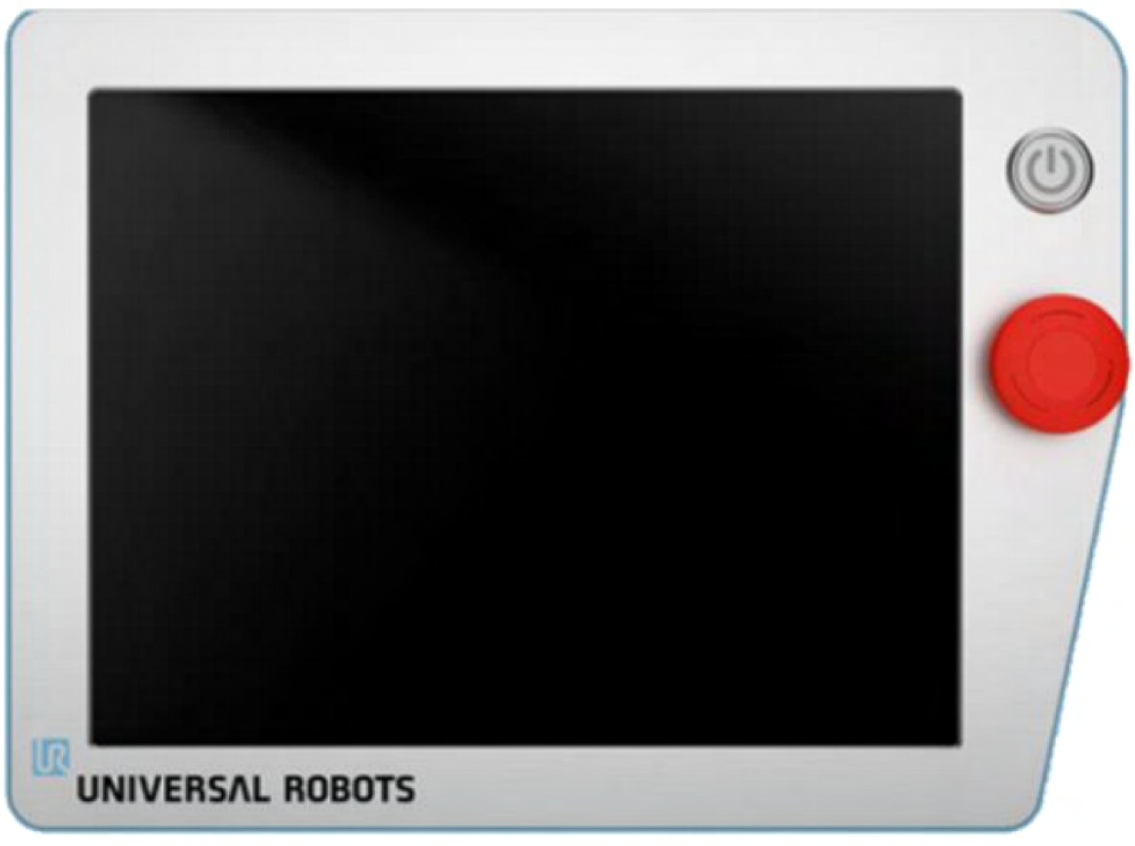


Figura 1.1: Pantalla UR5.

Para iniciar el robot se debe hacer clic a configurar robot como se observa en la figura 1.2, luego se abre una pantalla como se observa en la figura 1.3 y hacer clic en Iniciar Robot y pasará a la pantalla como tal de encendido como se observa en la figura 1.4 y hacer clic en encender y listo para mover el robot. Además, existe una opción para una carga activa la cual indica la carga de la herramienta, esto permite al robot saber cuanto esta moviendo y no activar la parada de emergencia a encontrar una perturbación en la fuerza ejercida por la masa de la herramienta.



Figura 1.2: Pantalla inicial encedida UR5.



Figura 1.3: Configurar UR5.

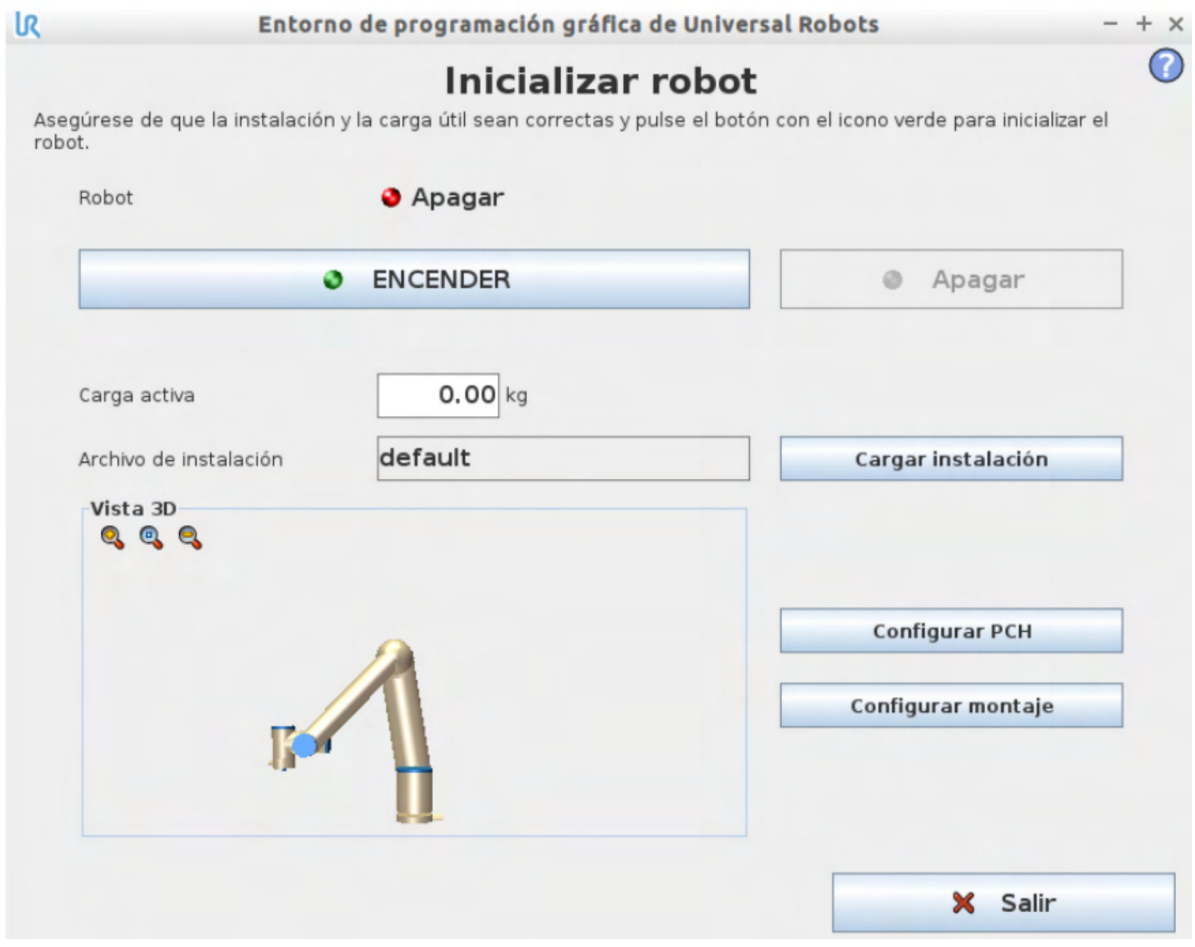


Figura 1.4: Encendido/Apagado UR5.

1.2 TCP o PCH

El TCP o PCH es el Punto Central de la Herramienta como se observa en la figura 1.5. Este punto ayuda al brazo a saber dónde se encuentra la base de la herramienta y a través de las coordenadas del PCH puede moverse de tal manera que la punta del extruder puede estar dirigido en todo momento como por ejemplo en la impresión 3D la salida del plástico fundido estar en contacto con el suelo.

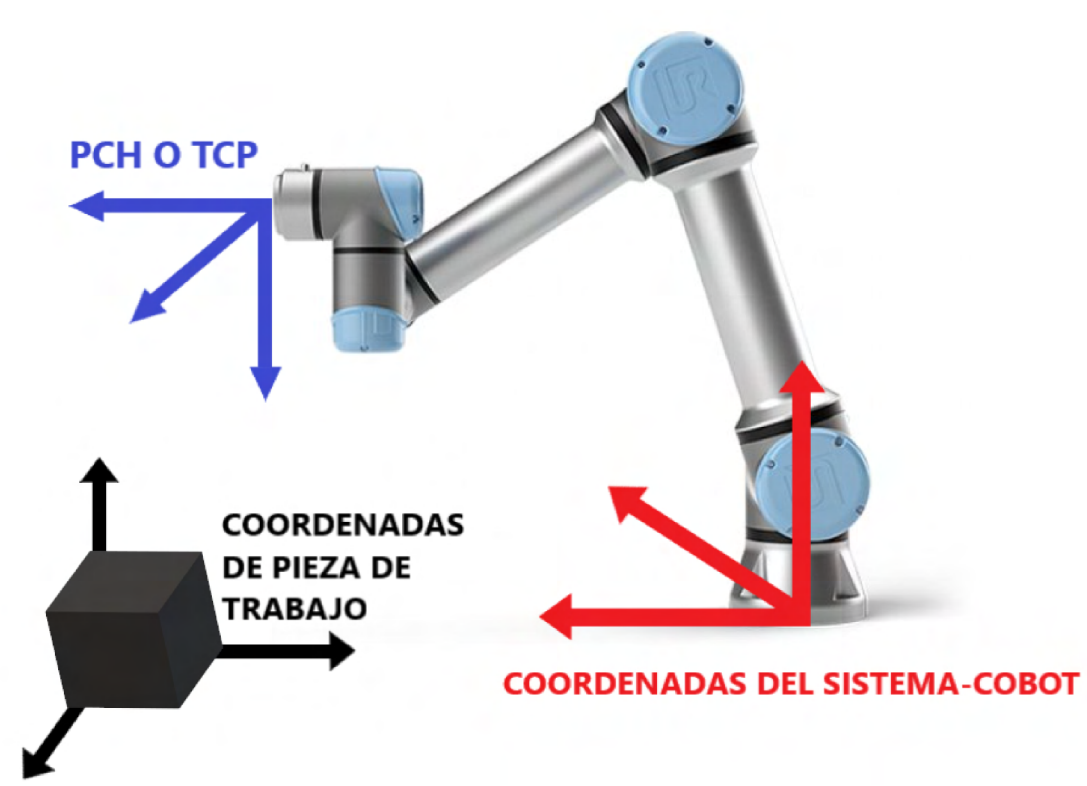


Figura 1.5: PCH o TCP.

Para configurarlo en el robot se debe ir a Instalación como se muestra en la figura 1.6 y hacer clic en configuración de PCH. Una vez ahí en la parte derecha de la pantalla se muestra gráficamente el PCH donde el origen se encuentra en la parte central de la muñeca 3. Para configurarlo se puede hacer de manera manual, colocando la distancia desde el centro de la muñeca 3 en los 3 ejes hasta la herramienta, y de manera automatica donde se debe cliquear en Posición y pedirá cuatro puntos como se muestra en la figura 1.7. Aquí se debe mover el brazo de tal manera que la punta de la herramienta llegue a un mismo punto, pero en diferentes posiciones (4 posiciones), para mover el brazo fácilmente y de manera rápida en la pantalla en la parte trasera tiene un botón el cual permite mover el robot libremente, en donde libera los frenos para guiar al robot con las manos.



Figura 1.6: Configuración de TCP.

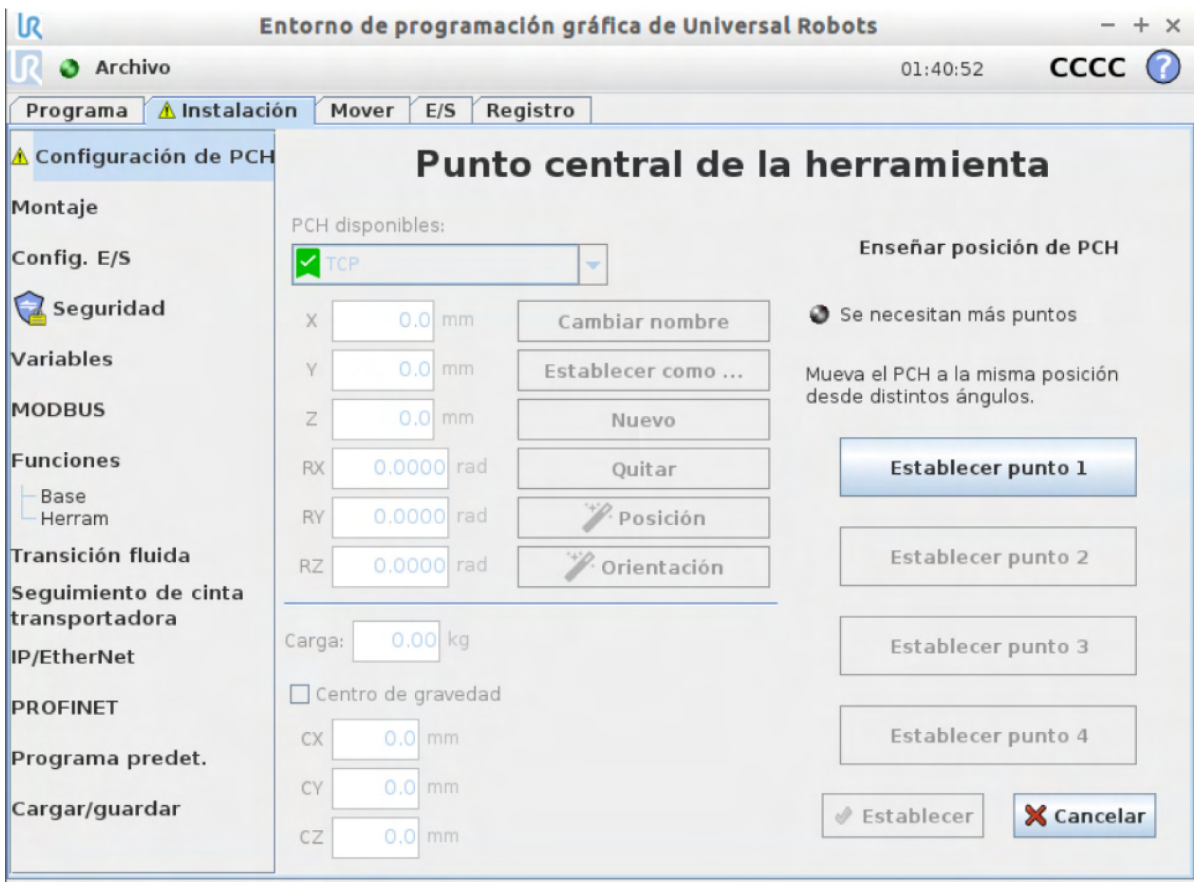


Figura 1.7: 4 puntos para TCP.

En la figura 1.8 se muestra el punto que se seleccionó para obtener el TCP el cual es una punta metálica fija en una mesa. Para este ejemplo se utilizo un extruder de menor tamaño para realizar los movimientos de manera rápida y segura.



Figura 1.8: Configuración de TCP.

En la figura 1.9 se muestran las diferentes posiciones que se realizo con el brazo UR5 para que el mismo cobot calculara de manera automática el TCP exacto, es decir que el punto central de la herramienta se encuentra en la punta del extruder, por donde saldría el plástico fundido.

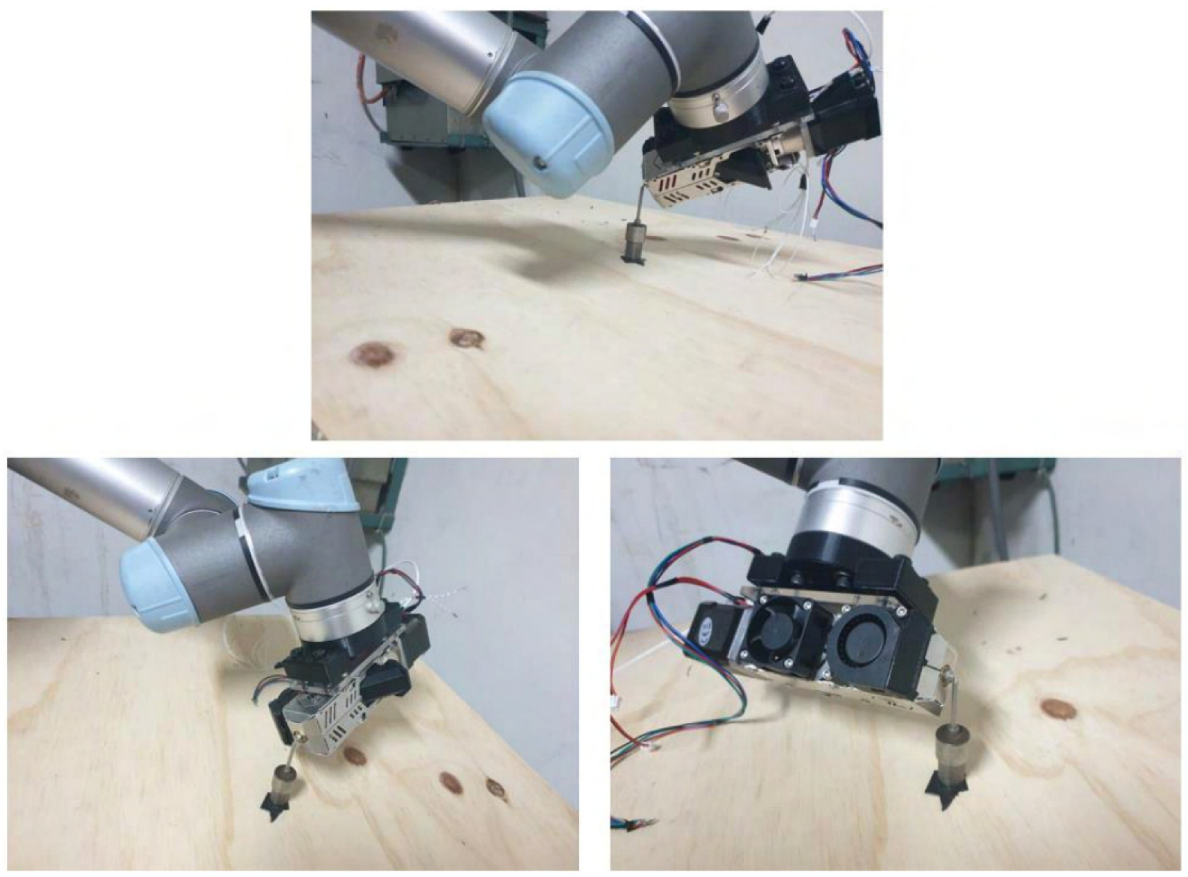


Figura 1.9: Posiciones brazo UR5 para TCP.

Una vez hechos los cuatro puntos nos aseguramos de que estos se encuentren con una luz verde, lo

cual indica que las posiciones coinciden en un punto y que el PCH está calculado como se observa en la figura 1.10.

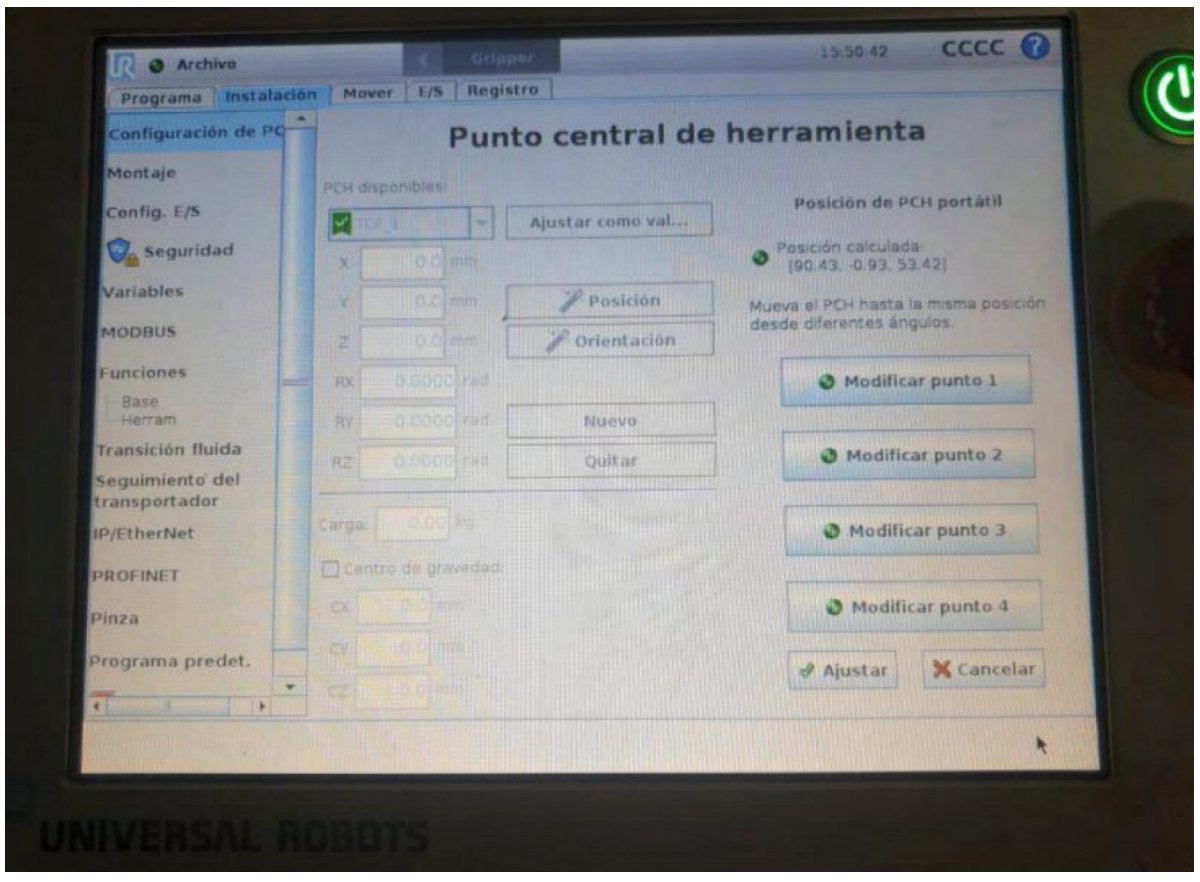


Figura 1.10: Puntos tomados por UR5 para la obtención de TCP.

1.3 Tipos de movimiento del Cobot (UR5)

Tipos de movimientos:

- MoveJ: Es un movimiento no lineal cuando la ruta del trayecto del brazo no es de importancia, donde cada junta articulada se controla para llegar al mismo tiempo a la posición deseada con un movimiento de trayectoria curva de la herramienta. Los parámetros compartidos que se aplican a Movej son la velocidad de la junta y aceleración de la junta máximas utilizadas, especificadas en grados/s y grados/s^2,respectivamente .

- MoveL: Es para trayectorias rectas (para recoger, colocar, etc), es decir movimientos precisos. Lo que significa que cada junta realiza un movimiento más complicado para mantener la herramienta en una trayectoria recta. Los parametros compartidos que se pueden configurar para MoveL son la velocidad de la herramienta y la aceleración de la herramienta, especificadas en mm/s y mm/s^2, respectivamente.

- MoveP: Movimiento de proceso a velocidad constante con transiciones circulares. Ideal para soldaduras o para aplicar pegamento, debido a que permite una distribución homogénea. El

tamaño del radio de transición esta predeterminado con un valor compartido entre todos los puntos de paso, en el cual un valor mas pequeño resultará en un movimiento brusco en la trayectoria mientras que al contrario un valor mayor resultará en un movimiento más suave en la trayectoria.

La curva se divide en tres segmentos: aceleración, crucero y desaceleración como se muestra en la figura 1.11. El nivel de la fase crucero se obtiene del ajuste de velocidad del movimiento,mientras que la pendiente de las fases de aceleración y desaceleración se obtiene del parámetro de aceleración.

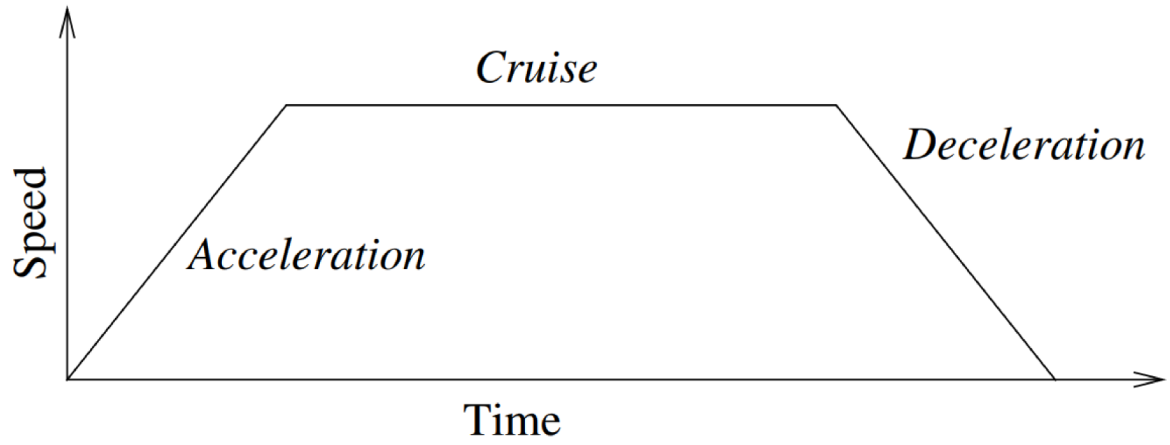


Figura 1.11: Perfil de velocidad.

1.4 Guardado

Tanto el PCH como el comando creado se puede guardar en el robot y así no realizarlo cada vez que se encienda. Para esto se debe apretar en Archivo y aparecerá opciones (ver figura 1.12), luego se toca en guardar como y aparecerá una ventana como se muestra en la figura 1.13, aquí se coloca el nombre deseado para el archivo y cliquear en guardar.

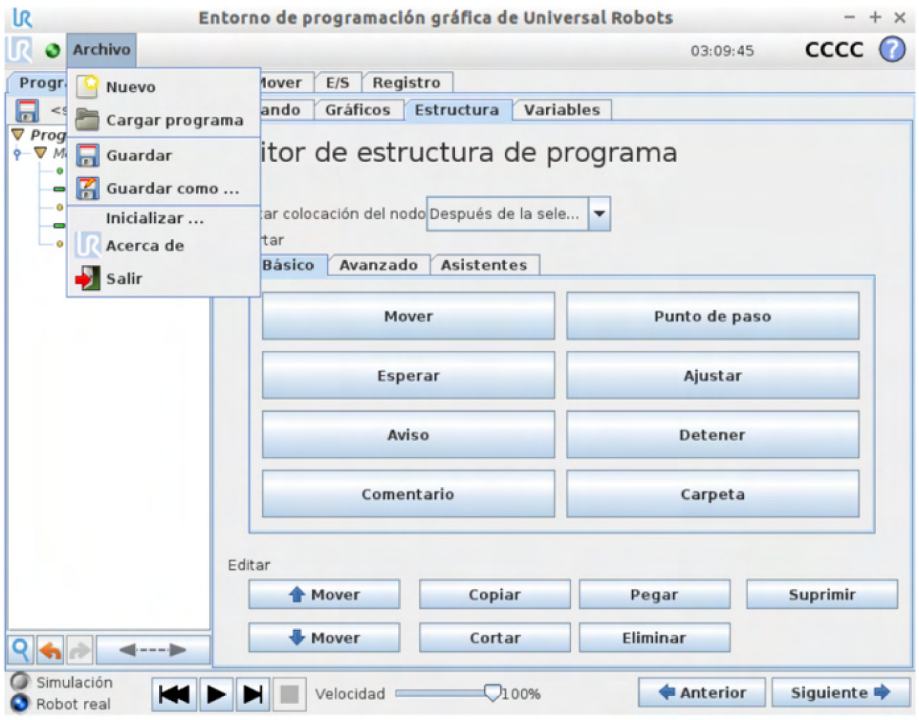


Figura 1.12: Guardar como.

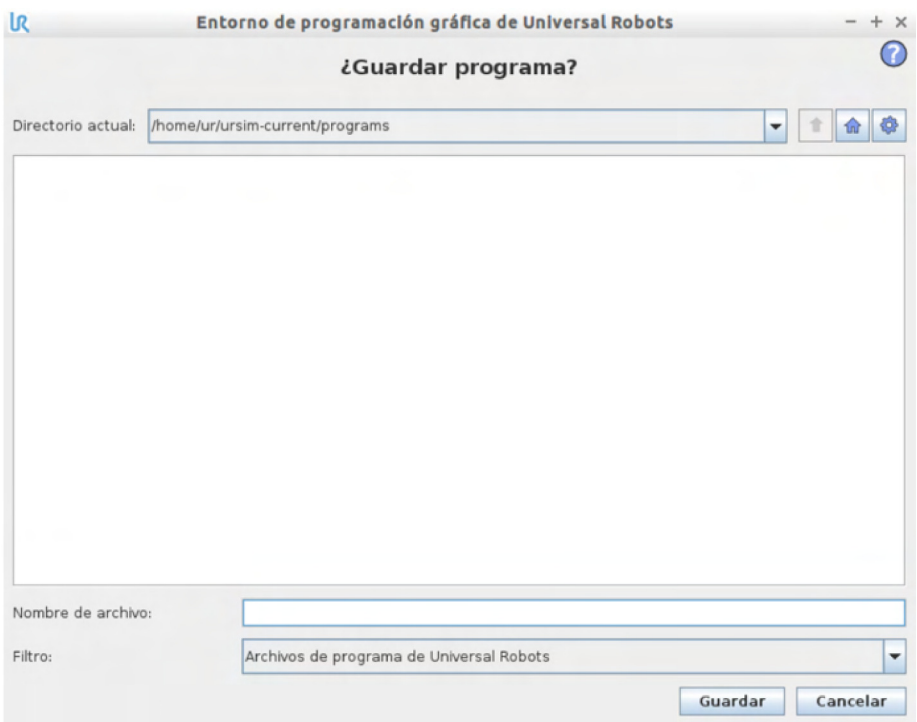


Figura 1.13: Guardar archivos.

1.5 Conectar PC con UR5

Para conectar el robot al PC se necesita de un cable ethernet y el programa Filezilla, el cual permite entrar al robot y ver todos los archivos de UR5. Para realizar esta conexión se debe cambiar la red del robot, para esto se debe ir a configurar robot, luego Red y se selecciona la Dirección estática y se cambia la dirección IP a 172.16.17.10, la máscara de Red a 255.255.255.0 y puerta de enlace predeterminada a 0.0.0.0 y se finaliza dando click a aplicar como se observa en la figura 1.14.



Figura 1.14: Configurar Red.

Luego en filezilla se coloca el servidor que se aplico al robot que es 172.16.17.10, en nombre de usuario “root”, contraseña “easybot” y puerto 22. Así entonces, se da click a conectar y una vez conectado se debe observar como la figura 1.15.

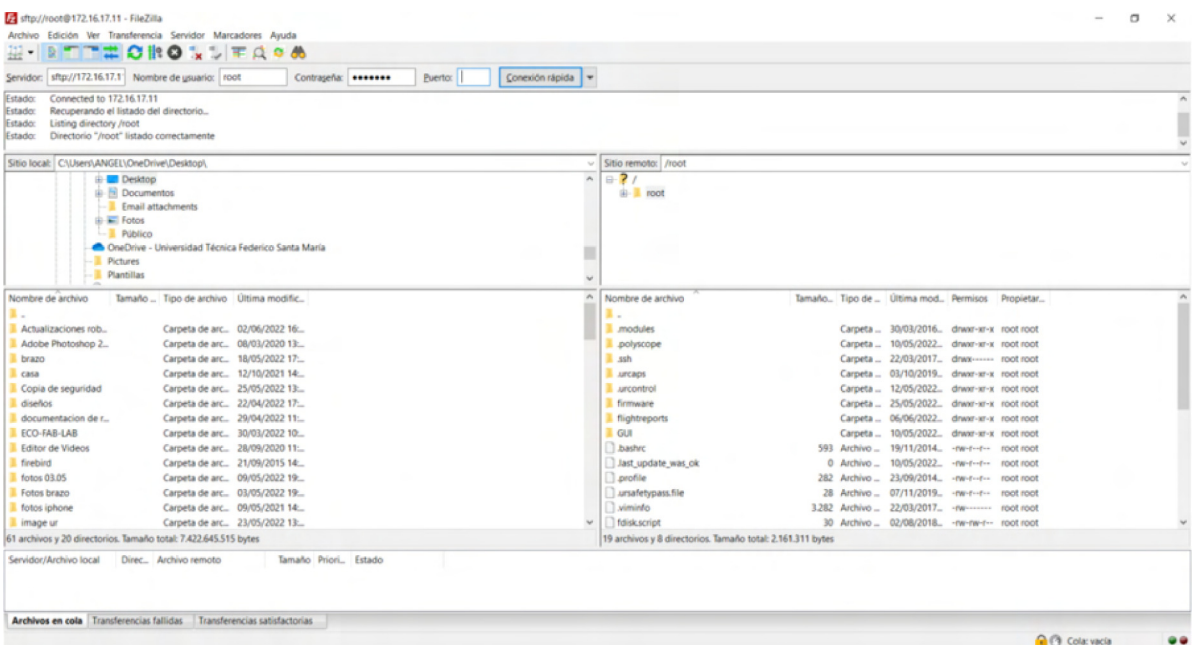


Figura 1.15: Conexión a UR5 desde PC con Filezilla.

En la figura se observan todos los archivos que tiene integrado el cobot, esto permite cargar archivos de programas desde el pc al brazo robótico, guardar carpetas de importancia en el pc para tener una copia si se reinicia o actualiza el UR5 y poder integrarla nuevamente si se da el caso que se

haya eliminado, etc.

1.6 RoboDK

Para lograr que el robot se mueva siguiendo una trayectoria deseada sin la necesidad de saber todos los puntos de la trayectoria y llegar hacer un programa, es que existen softwares que trabajan con brazos robóticos y facilitan al usuario que pueda crear movimientos a partir de figuras deseadas para imprimir en 3d. Para esto se ocupa el software de RoboDK ocupado en la industria de manufactura robotizada, el cual requiere licencia para ocupar sus distintas funciones entre las cuales la impresión 3d. Para las pruebas se comunicó con el departamento de arquitectura que tienen en su poder licencias estudiantiles de robodk, las cuales se utilizó para primeramente realizar dibujos con el robot UR5 y obtener resultados de movimiento de éste respecto a distintas figuras antes de ocupar la impresión 3D.

1.6.1 Conexión de Robodk a UR5

Para conectar a Robodk al cobot primero se debe forzar la ip de la computadora, para esto se debe ir a Estado de red y luego a Cambiar opciones de adaptador y se abrirá una ventana como se observa en la figura 1.16.

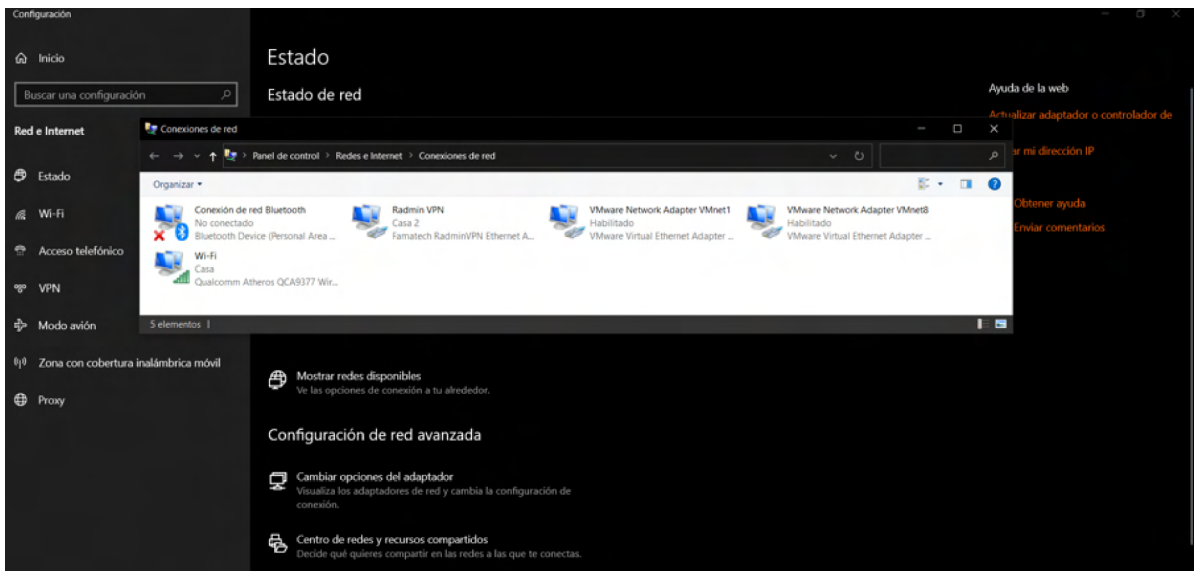


Figura 1.16: Cambiar opciones de adaptador.

Luego ir a Radmin VPN dar click derecho a propiedades, activar la opción protocolo de internet versión 4 e ir a propiedades como se muestra en la figura 1.17. En propiedades activar la opción de Usar la siguiente dirección IP y colocar lo siguiente:

- Diercción IP: 172.16.17.11 Esto permitirá que la computadora esté en el rango de dirección de IP del brazo UR5 definido en la sección 1.5 en configurar red del robot.
- Mascara: 255.255.255.0 Al igual que la sección anterior 1.5, debe estar en la misma máscara de subred.
- Puerta de enlace predeterminada: no colocar nada.

Una vez colocado estos valores como se muestra en la figura 1.18 se da click a aceptar y una vez se conecte el cable eterneth a la computadora tomará la IP descrita anteriormente lo que permitirá que tanto el brazo como el pc estén en la misma dirección.

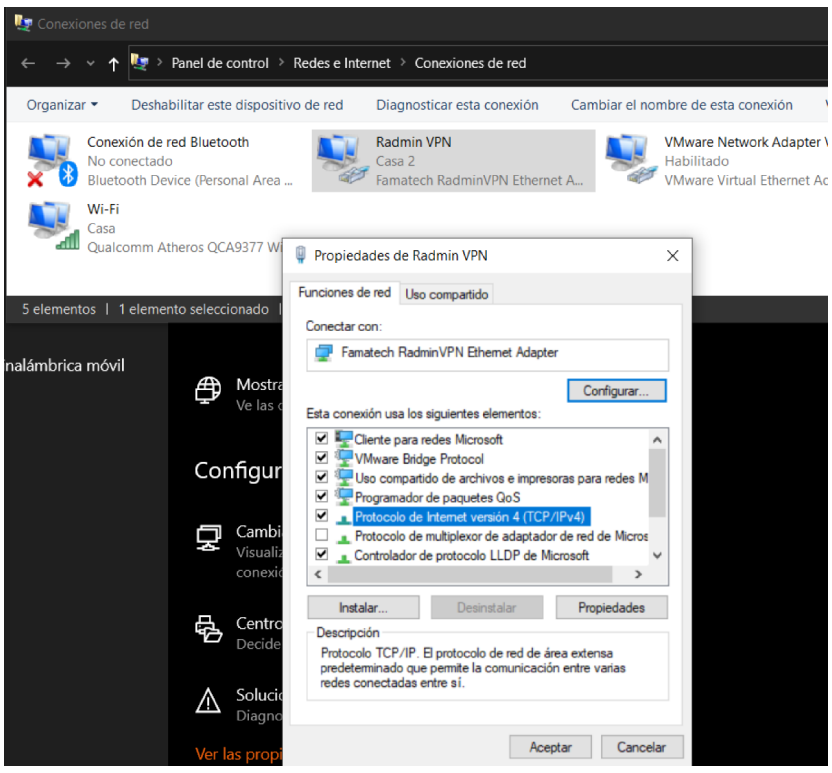


Figura 1.17: Activar IPv4.

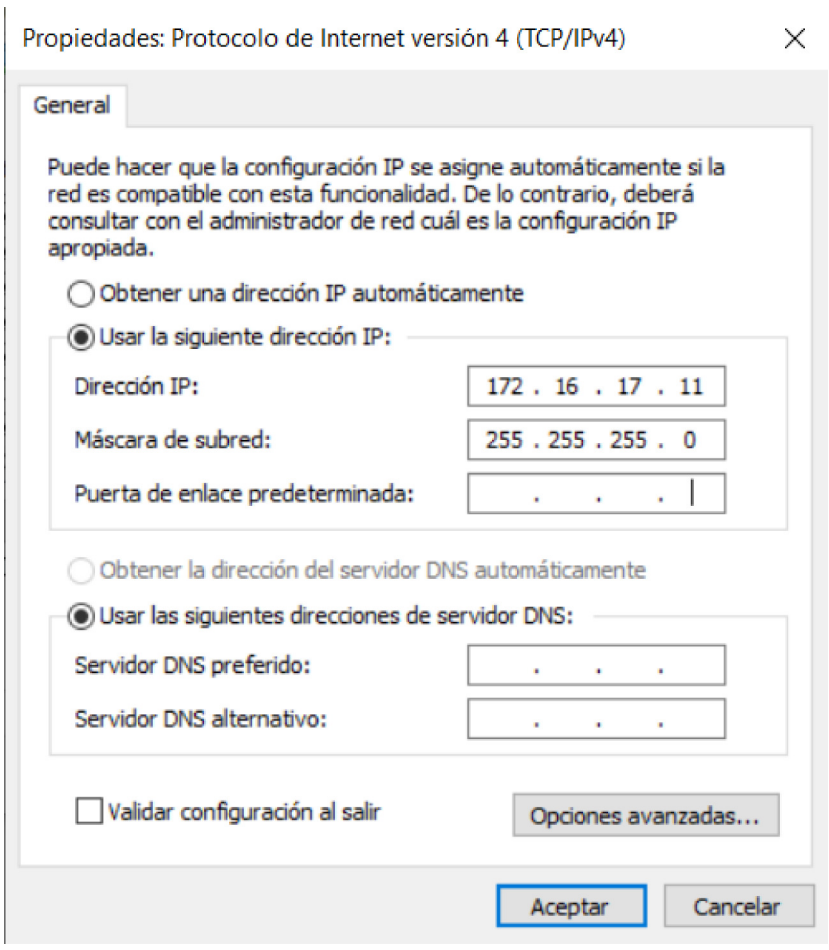


Figura 1.18: Activar IPv4.

Finalmente, para hacer la conexión del software RoboDK con el brazo UR5 se debe descargar el archivo que contiene las informaciones físicas del brazo robótico, el cual se obtiene de manera gratuita en la página oficial de Robodk. Una vez descargado el archivo del brazo UR5 se abre el software y se hace click en archivo, luego en abrir, se busca el archivo descargado de RoboDk del cobot UR5 y se hace click en abrir, así entonces se mostrará el robot como se observa en la figura 1.19. Una vez mostrado el robot virtualmente se debe hacer click a conectar ubicada en las opciones en la parte superior del sofwtare, luego a conectar robot, se abre una ventana en la parte izquierda y se debe colocar la dirección que se le impuso a la computadora 172.16.17.11 y mantener el puerto por defecto. Se cliquea en conectar y debe aparecer Ready y color verde, lo cual indica que se conectó correctamente al brazo robótico como se muestra en la figura 1.20, por lo que el robot puede ser movido desde el pc. Notar que no se puede mover libremente el robot con el botón trasero que permite moverlo manualmente. Para esto se debe desconectar de Robodk haciendo doble click en desconectar.

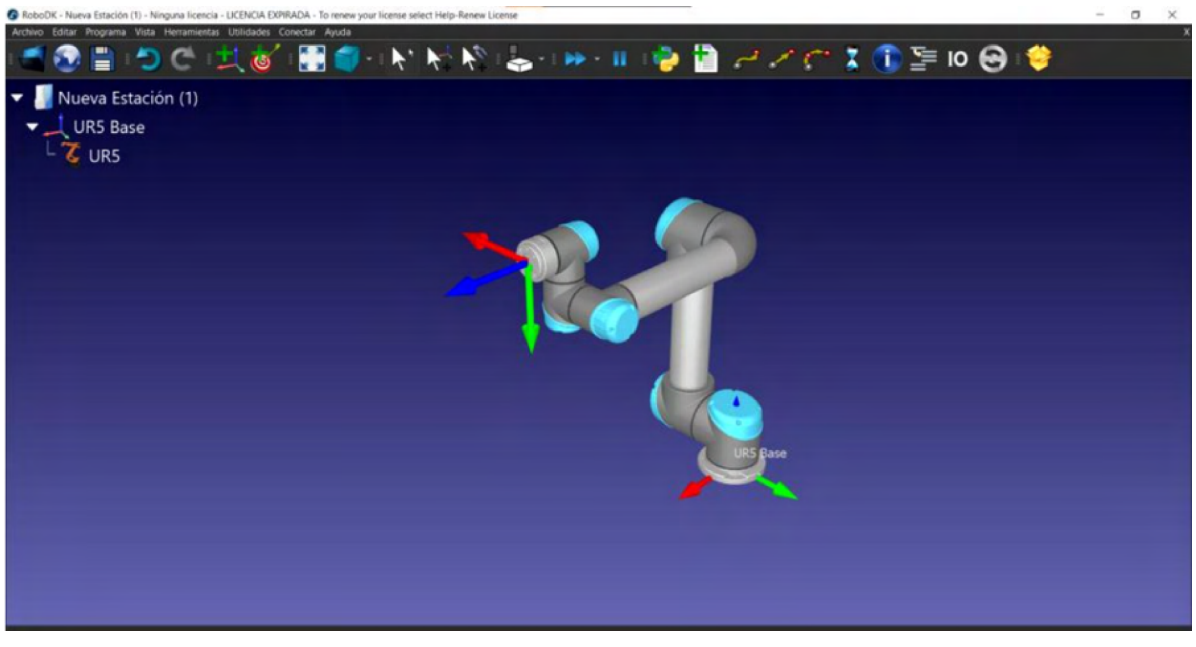


Figura 1.19: Interfaz RoboDK con archivo de UR5.

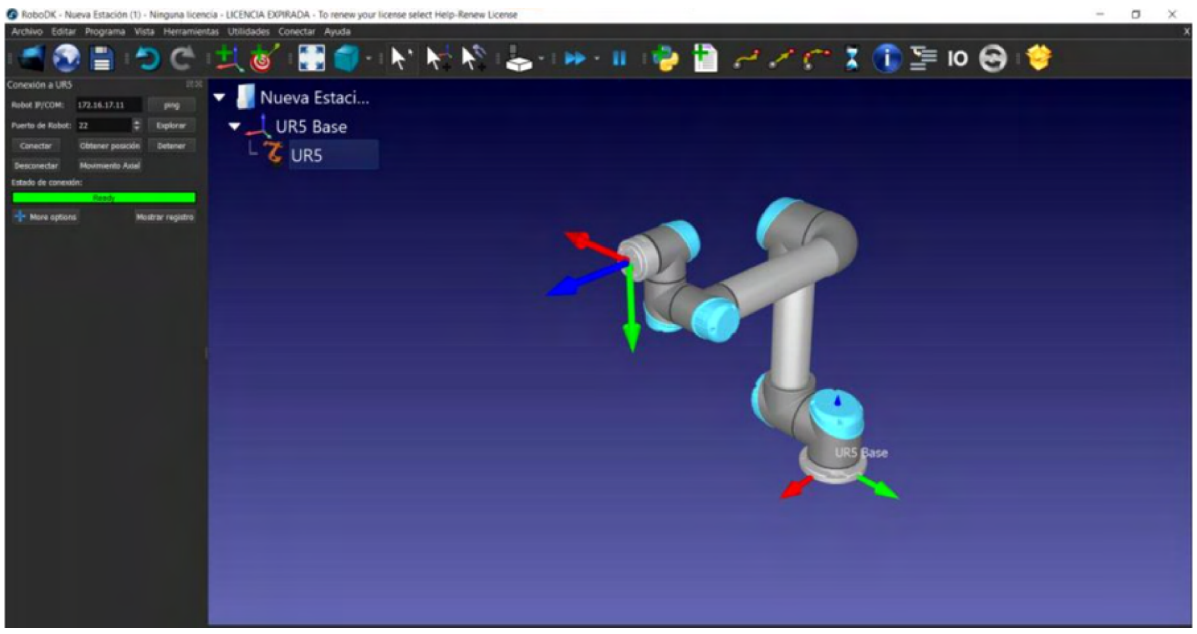


Figura 1.20: Conexión de RoboDK con el brazo UR5.

1.7 Impresión 3D

Para este tipo de impresión es importante tener consideraciones antes de realizar prácticas, las cuales vienen dadas primero por el tipo de material, para nuestro caso es plástico, debido a que establecerá el rango utilizado en la temperatura para su adecuada fundición. También consideraciones de velocidad, tanto del Cobot como del tornillo sinfín del extruder, para esto no existe un modelo definido y constará del estudio, pruebas y experiencias compartidas que guiarán a un buen resultado. Una mesa rugosa con temperatura que permita que el plástico se asiente y no se corra al momento de pasar a otra capa también es importante para un buen resultado. Y por último un factor que se debe tener en cuenta es que se va depositar gran cantidad de material fundido encima

una de otra, esto dará como resultado que por el peso y la facilidad de moldeo por la temperatura, las capas tiendan a caer.

1.7.1 Configuración actual Cobot-Extruder

En la figura 1.21 se muestra como esta configurada las distintas partes que conforman la manufactura aditiva de alto volumen del laboratorio Eco-Fab-Lab, las cuales son: Brazo robótico UR5 (Cobot), Extruder de alto volumen, Brazo compensador y Cama caliente (cama de impresora 3D Ender de laboratorio Eco-Fab-Lab).



Figura 1.21: Configuración actual de Cobot-Extruder y brazo compensador.

1.7.2 TCP o PCH del Extruder de alto volumen junto con la unión

Para poder realizar un correcto programa de impresión 3D al brazo robótico se debe tener en cuenta el punto central de la herramienta (PCH), en este caso es el extruder de alto volumen. Pero como se mencionó en las secciones anteriores se montó un brazo compensador el cual se lleva el peso del extruder y el brazo sólo guía los movimientos. Sin embargo, esta triple unión dificulta que el mismo software que puede calcular el PCH, explicado en la sección 1.2, debido a que no se puede rotar la muñeca número 3 y tener distintas poses para que se calcule automáticamente.

Es por esto que se debe realizar manualmente, midiendo desde el origen en los 3 ejes (x,y,z) de la muñeca número 3 como se muestra en la figura 1.22. Y para saber el punto de referencia en el robot real, se debe tomar la conexión integrada que trae el cobot para herramientas que se conectan a los input y output del mismo, que tiene forma de cilindro que sobresale de la muñeca. Esta conexión esta en el eje "y" en la parte negativa, por lo tanto ese será el punto de referencia a tomar para medir desde el origen de los ejes a la punta (boquilla) del extruder.

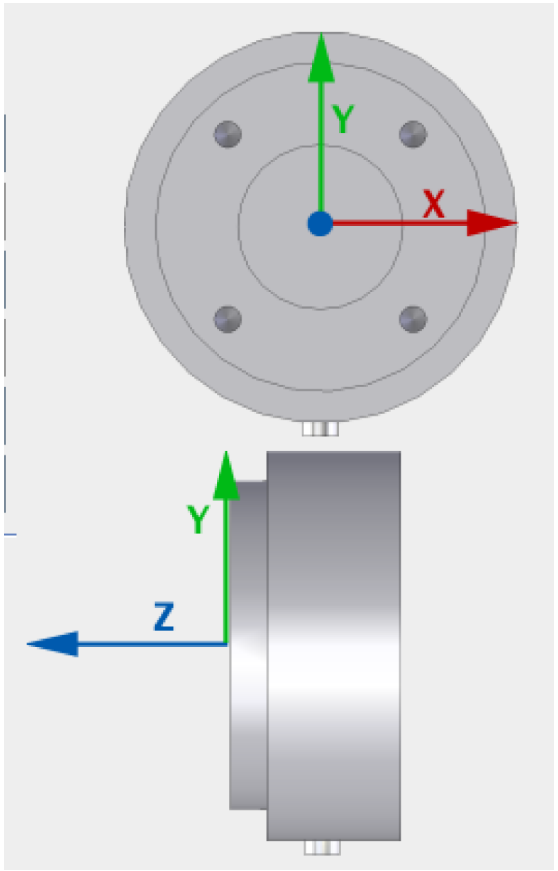


Figura 1.22: Coordenadas base muñeca 3.

Como se observa en la figura 1.23 se toma la medida desde el centro de la muñeca 3 al eje central del extruder en dirección del eje z teniendo en cuenta la distancia entre la placa y el centro de la muñeca 3, es decir la distancia que ocupa la bisagra (a), además del espesor de la placa que son 10 [mm] (b) y por último la distancia de la placa al eje central del extruder (c). Luego por la configuración final de la impresora 3D de alto volumen y la unión que se hizo de tal manera que quedará lo más centrado posible se obtuvo que no había un mayor desplazamiento en el eje y, por lo que se mide desde el centro de la muñeca 3 a la boquilla del extruder en dirección a x negativo (d), Obteniendo las siguientes medidas como se muestra en la tabla 1.1:

Tabla 1.1: Medidas PCH de extruder de alto volumen en configuración final

Eje	Medida	Unidad
X	-209.5	mm
Y	0	mm
Z	121	mm

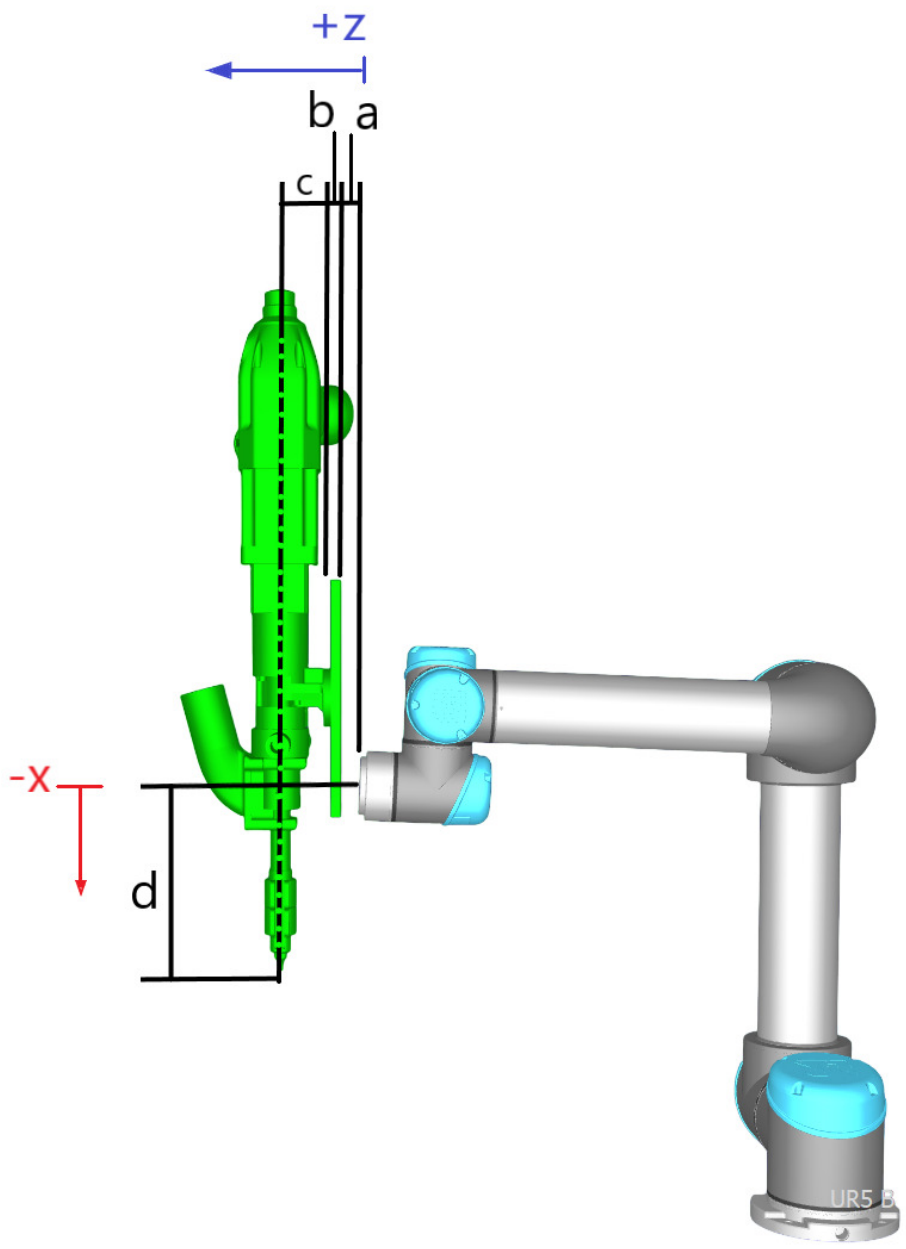


Figura 1.23: Imagen representativa de toma de medidas TCP Extruder.

1.7.3 Creación de programa para impresión 3D

Para obtener las trayectorias de un objeto 3D hay que tener en cuenta en la creación de la figura, el cual se crea en software Inventor, el cual permite realizar objetos tridimensionales. Luego se debe realizar el código g con la ayuda de un slicer que para este trabajo se ocupa el Ultimaker Cura utilizado en la mayoría de impresoras 3D y Slic3r, los cuales entregan un código detallado dependiendo de las opciones que se necesiten. Finalmente, llevar dicho programa a el software RoboDk el cual obtendrá las trayectorias deseadas para conformar el objeto impreso, hay que tener en cuenta que este software no tendrá en consideración los movimientos permitidos por el brazo compensador por lo que se deben hacer simulaciones antes de ejecutarlo en el brazo robótico UR5.

1.7.4 Creación de código G con slicer Ultimaker Cura

Se debe tener en consideración que al momento de exportar el archivo debe ser recomendablemente en STL y con la opción de alta resolución como se muestra en la figura 1.24. Esto es importante ya que al momento de llevar el archivo en STL al slicer Ultimaker Cura, será sensible a las trayectorias y si es de baja calidad para trazados curvos tomará trayectorias rectas pequeñas pero notorias al contrario de una alta resolución como se observa en la figura 1.25 donde el cilindro del lado izquierdo es exportado en una alta resolución mientras que el cilindro del lado derecho en una baja resolución.

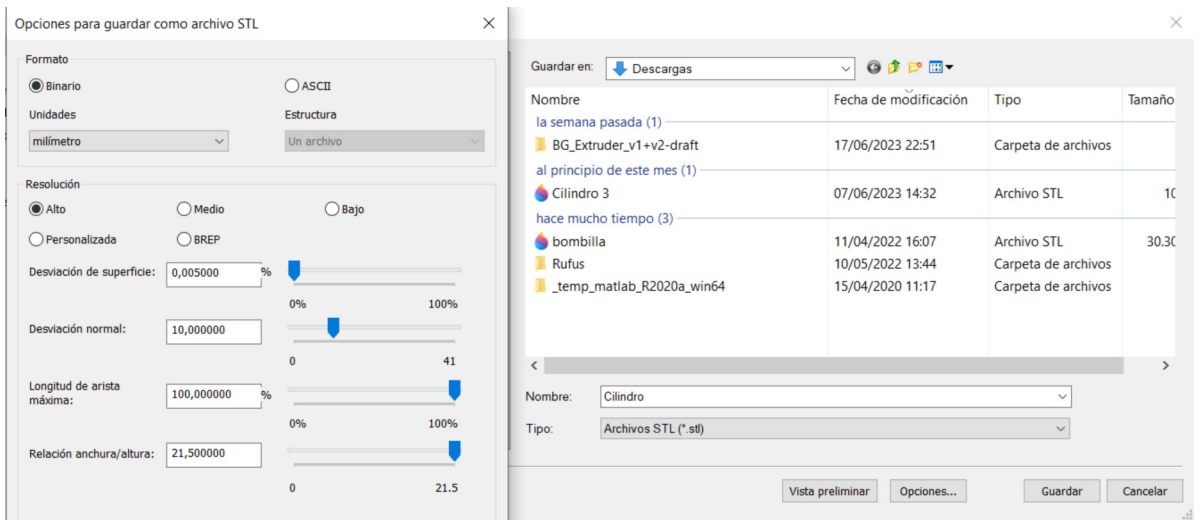


Figura 1.24: Opción de alta resolución para importar objeto en archivo STL.

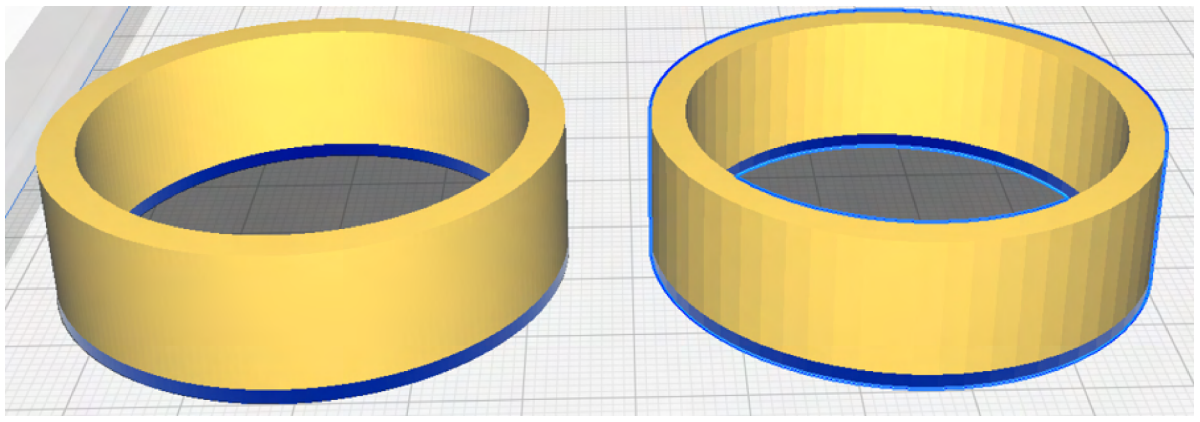


Figura 1.25: Comparativa entre cilindro exportado en alta resolución y baja resolución.

Asi entonces, se debe importar el archivo de STL en alta resolución a slicer Ultimaker Cura y se observa como se muestra en la figura 1.26.

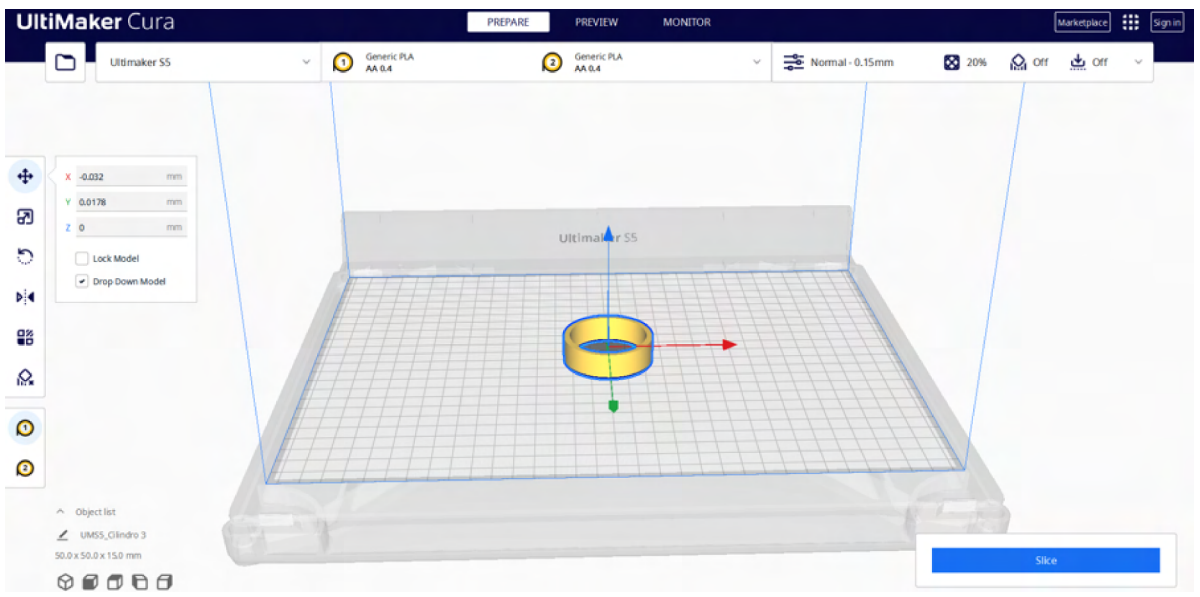


Figura 1.26: Interfaz slicer Ultimaker Cura con cilindro.

Al momento de importar un archivo y ocupar la mesa por defecto, esta trae el origen en la esquina izquierda inferior por lo que afectara al código G que se crea, debido a que tomará las coordenadas desplazadas del origen. Es por esto que para mayor comodidad se debe centrar el origen a la mesa en la opción de configuración, luego a impresoras y dando click en configuración de la máquina. Se abrirá una ventana como se observa en la figura 1.27 y habilitar la opción origin center. Notar que también se puede modificar las dimensiones de la mesa, pero para este estudio es suficiente el espacio.

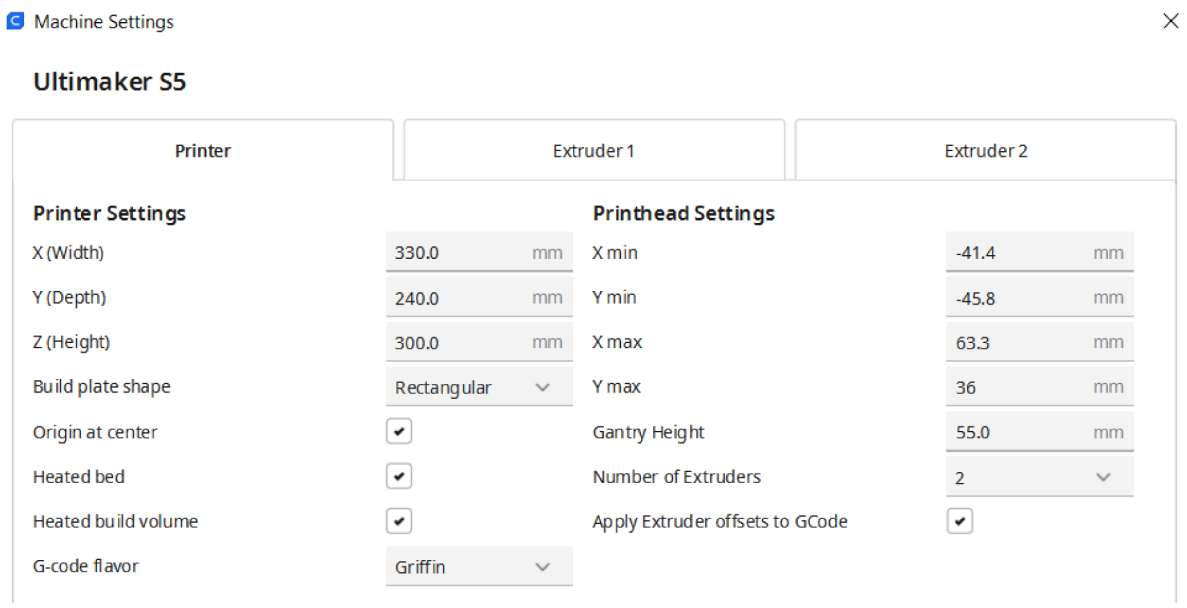


Figura 1.27: Opción para centrar la pieza u objeto.

Para las configuraciones de la calidad de las capas se tiene las opciones de Quality como se muestra en la figura 1.28, donde se enfocará en 2 opciones. La primera es Layer Height en el cual se informa al slicer de cuanto deben ser las alturas de las capas, para este estudio como el diámetro de la boquilla es de 3 [mm] la altura debe ser mínima de 3 milímetros, debido a que dependerá de como fluya el material si se requiere dar más altura o no. La segunda opción importante es Initial Layer

Height, que tiene como función dar a conocer la altura de la capa inicial al código G, se recomienda que sea cercana en un rango de 5 [mm] menor al diámetro de la boquilla o el mismo valor para asegurar un buen asentamiento del material a la mesa. Para el ancho de linea esta la opción Line Width la cual esta influenciada por el ancho de la boquilla, que para este estudio es de 3 [mm]. Las otras opciones se auto-calcularán por el mismo slicer.

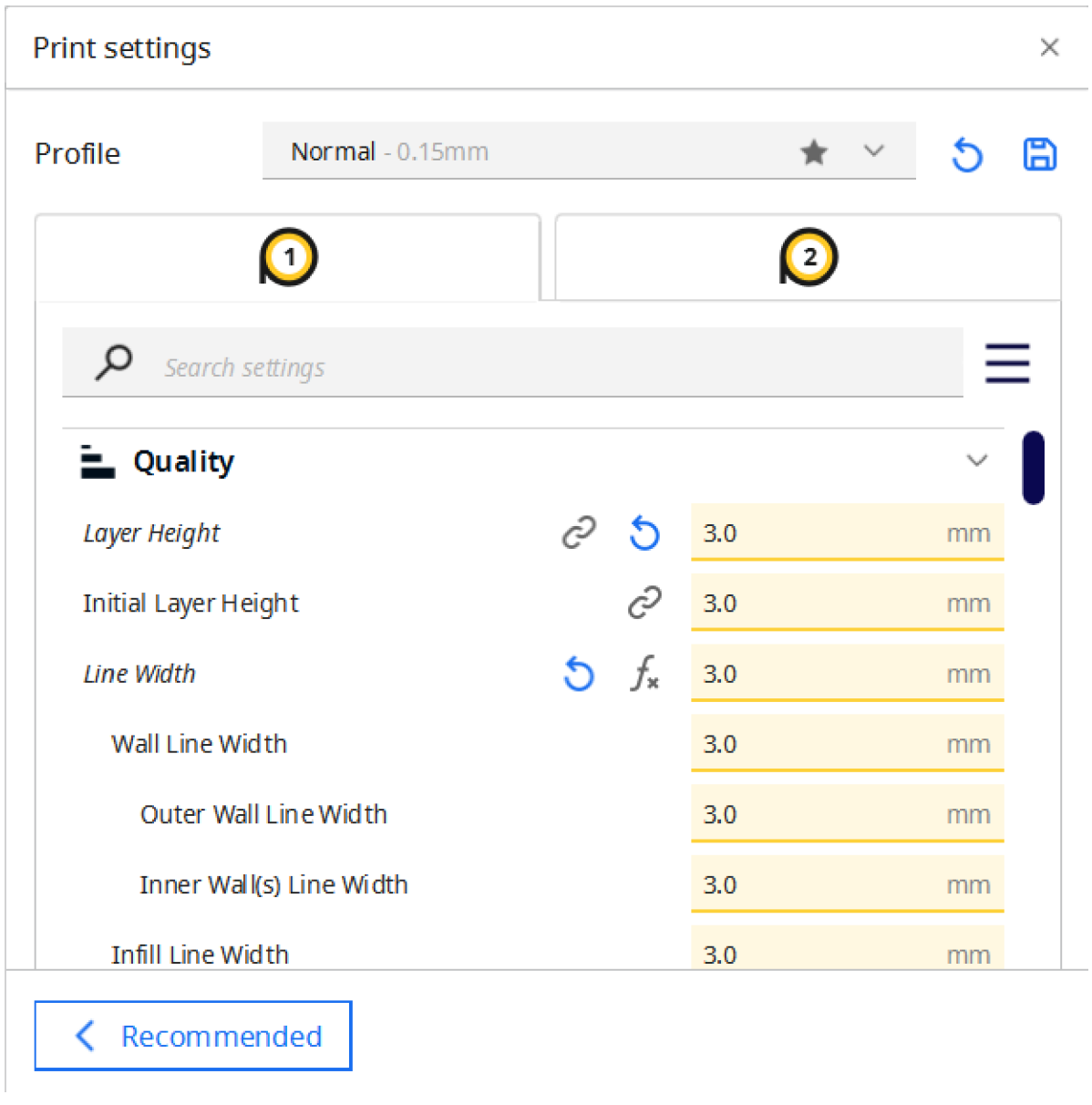


Figura 1.28: Opciones de calidad de las capas.

Otra configuración importante son las paredes como se muestra en la figura 1.29. En esta configuración es importante que la opción Wall Line Count sea 1, ya que esto indica cuantas veces pasará el cobot por la misma capa y como esta configurado para que el espesor del cilindro sea el mismo que el diámetro de la boquilla del extruder de alto volumen es necesario indicar o comprobar que sea 1.

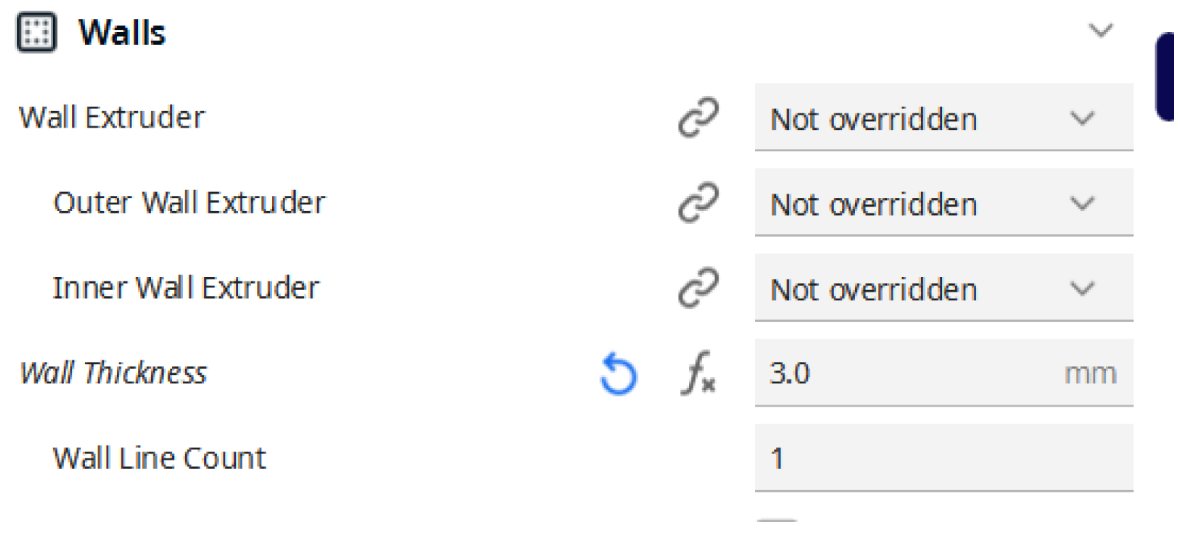


Figura 1.29: Opciones de Paredes.

Para que el slicer no genere una cama a la pieza al imprimir, se debe ingresar el valor cero en las opciones que se muestran en la figura 1.30 con esto se asegura que el código G no tome como trayectorias en la parte inferior y superior del cilindro. Esta opción es opcional.

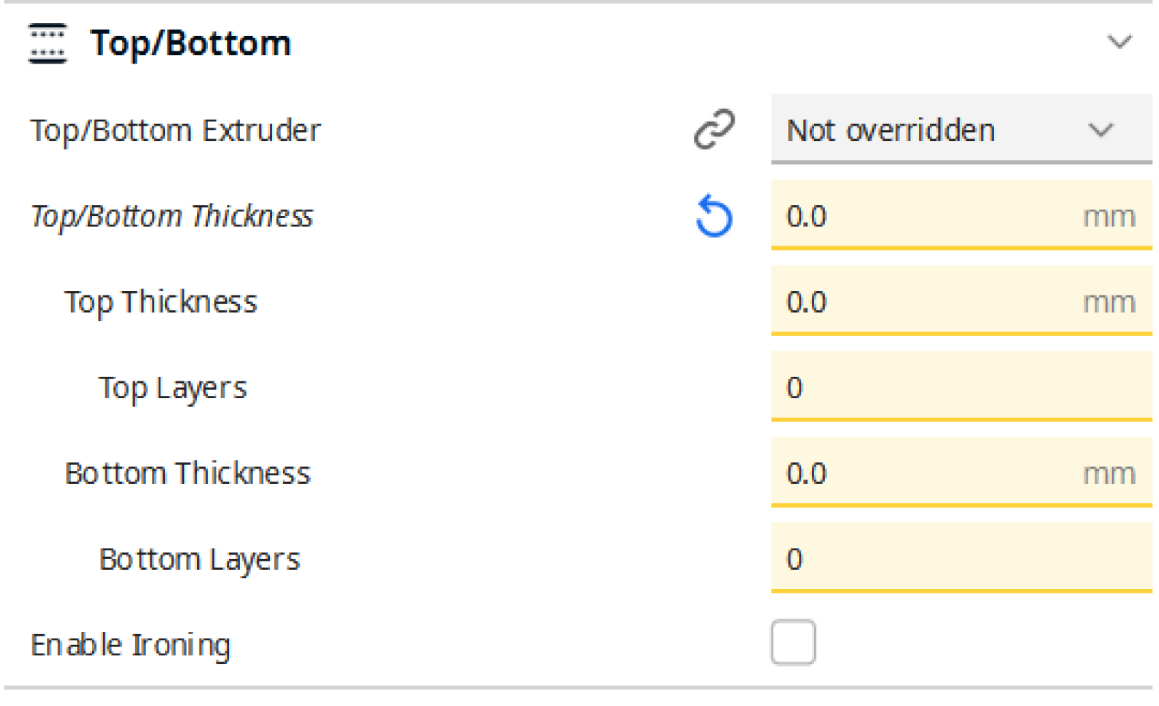


Figura 1.30: Opciones de cama de la pieza.

Para las opciones de velocidad se recomienda que sea constante en todas opciones como se muestra en la figura 1.31.

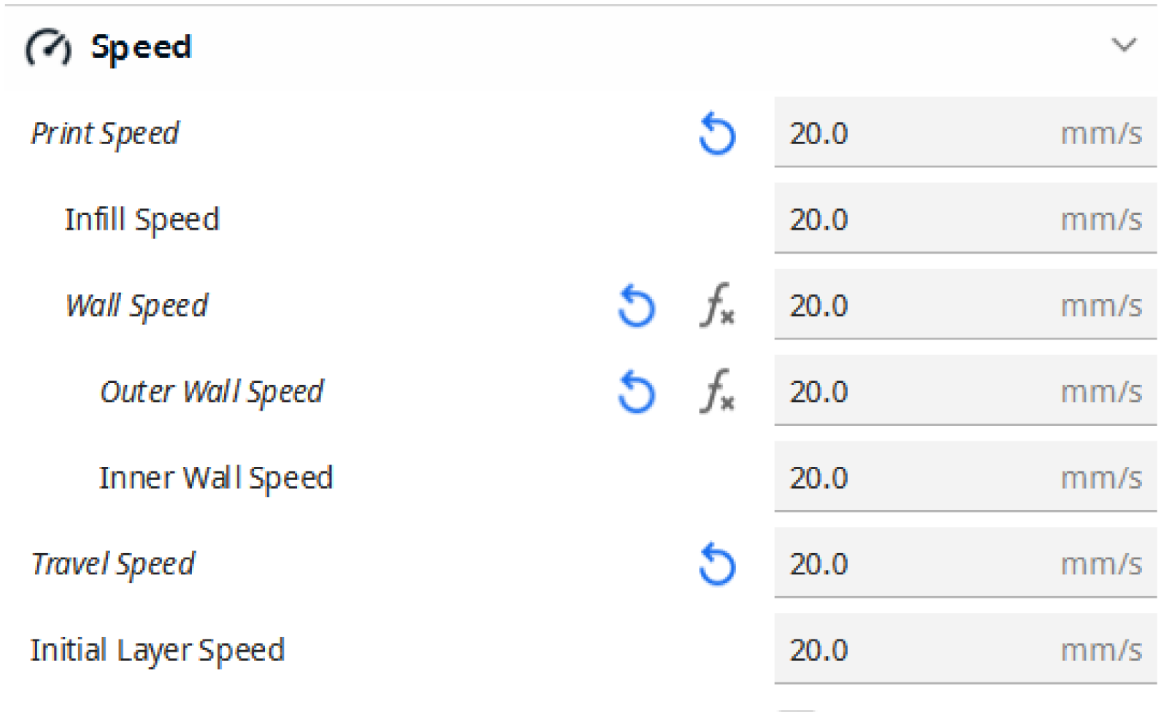


Figura 1.31: Opciones de velocidad.

Finalmente una opción importante es la de Spiralize Outer Countour como se muestra en la figura 1.32, el cual indica al slicer que la impresión es sin retraerse y lo hará de forma de espiral, esto reduce el tiempo a imprimir y se acomoda a las condiciones actuales de trabajo para las pruebas, ya que el flujo de material es continuo. Y la opción de Smooth Spiralized Countours para suavizar los contornos del espiral y obtener mejores resultados al momento de imprimir.

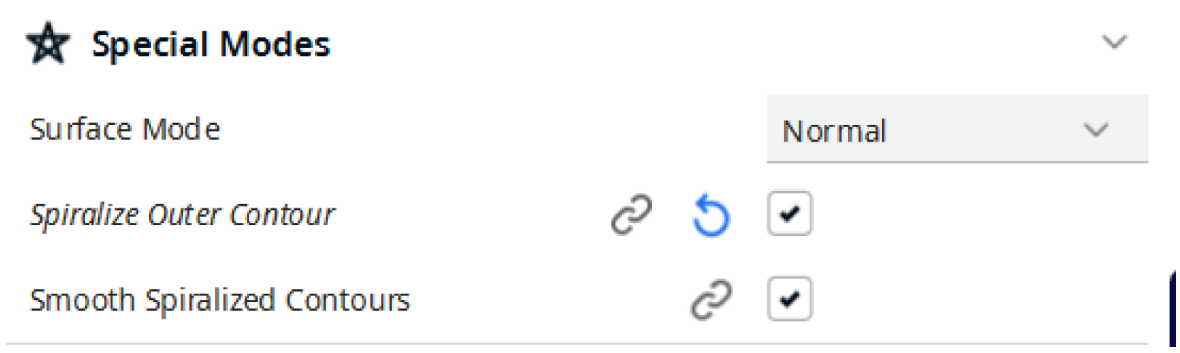


Figura 1.32: Opciones de modos especiales.

Una vez considerado las opciones explicadas anteriormente se puede realizar las rebanadas (Slice) en el software e ir a preview en la parte superior del interfaz para observar como quedarían las capas para crear en este caso el cilindro como se muestra en la figura 1.33. Se puede observar que se generaron 5 capas como se esperaba en forma de espiral desde la capa inicial hasta la final, también se muestra el tiempo que tomará con los datos seleccionados dando como tiempo 0 [min], lo que significa que durará en imprimir menos que 1 minuto.

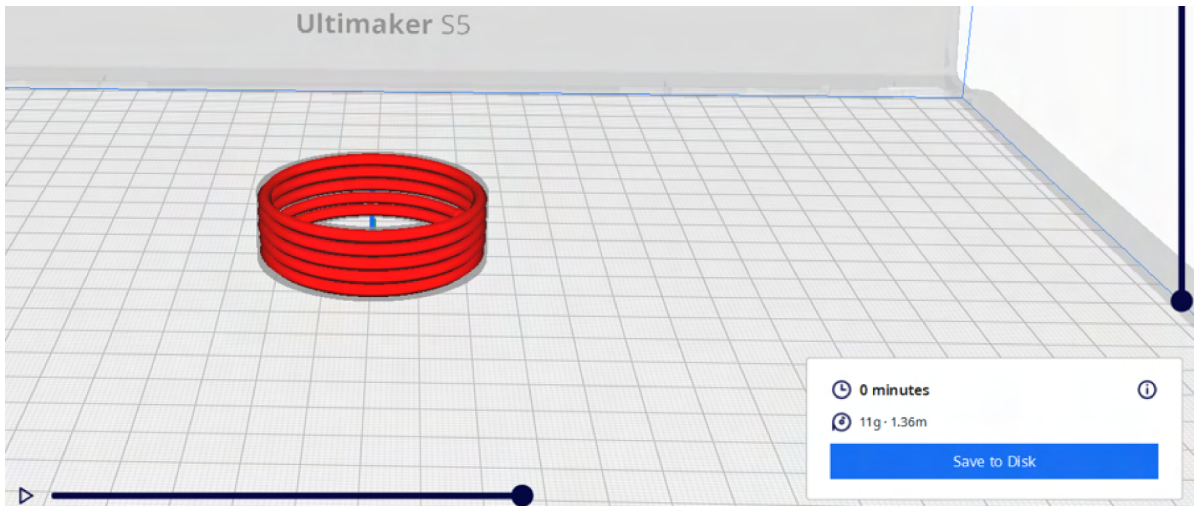


Figura 1.33: Capas generadas por Ultimaker Cura con las opciones anteriores.

Así entonces, se puede exportar el código G en la opción archivo, exportar y procurar indicar en el tipo de archivo sea gcode.

1.7.5 Realizar programa de impresión 3D en RoboDk

Para la realización del programa en robodk hay que tener consideración sobre la mesa de trabajo real que se ocupará. Para este estudio como se menciono en las secciones pasadas, se utiliza una mesa caliente de la impresora 3D ender del mismo laboratorio Eco-Fab-lab. Las consideraciones que se deben tomar es:

- Nivelar la mesa: Se debe mover el brazo en cuatro puntos en las esquinas opuestas y acercar de tal manera de que la boquilla del extruder de alto volumen quede separada de forma que una hoja de papel pueda pasar ajustada.
- Referencia de mesa de trabajo: Luego obtenida la mesa nivelada se debe escoger la parte centrada en donde se moverá el robot en la mesa caliente y estar conectado a Robodk.

Luego de haber obtenido la posición correcta se debe conectar el robot al software RoboDK y una vez conectado se debe dar click a obtener posición en la opción central debajo del puerto del robot como se muestra en la figura 1.34.

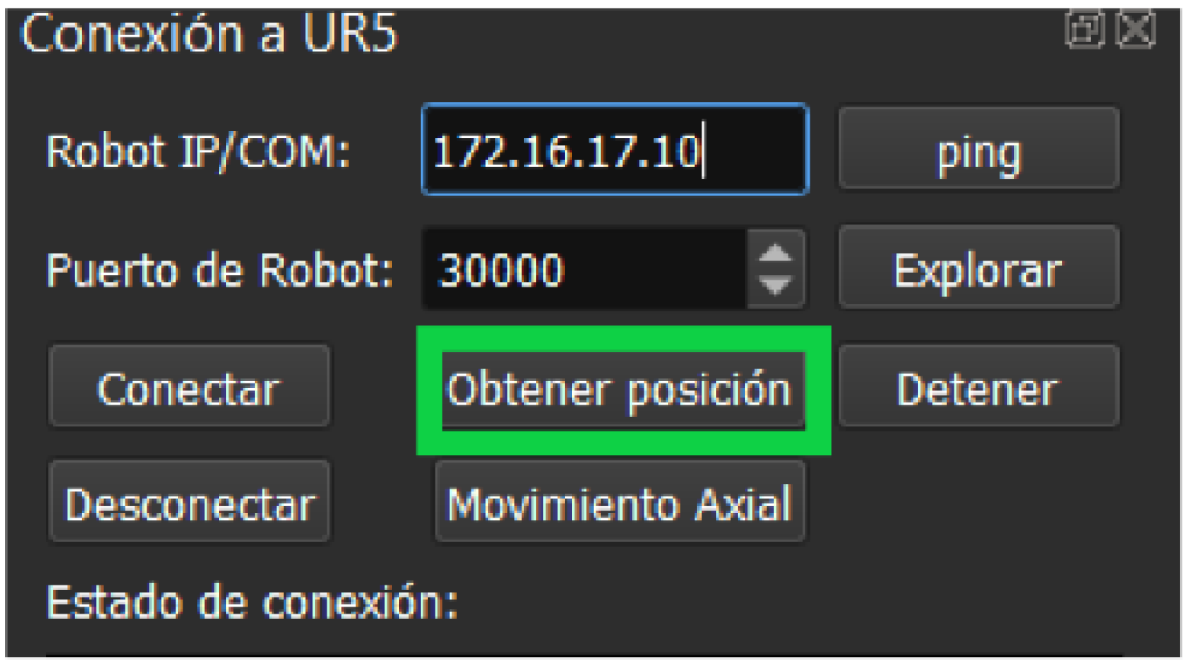


Figura 1.34: Obtener posición real de UR5.

Esta opción permite tener las coordenadas espaciales del robot en esa posición, es decir que nos muestra la coordenada que se debe utilizar para la mesa de trabajo. Así entonces, la recomendación para obtener el valor numérico de la posición es agregar un "Target" que se encuentra en las opciones superiores en el software RoboDK con un icono de un dardo dando al blanco con un signo "+" de color verde y se da click, instantáneamente se agregará como se muestra en la figura 1.35.

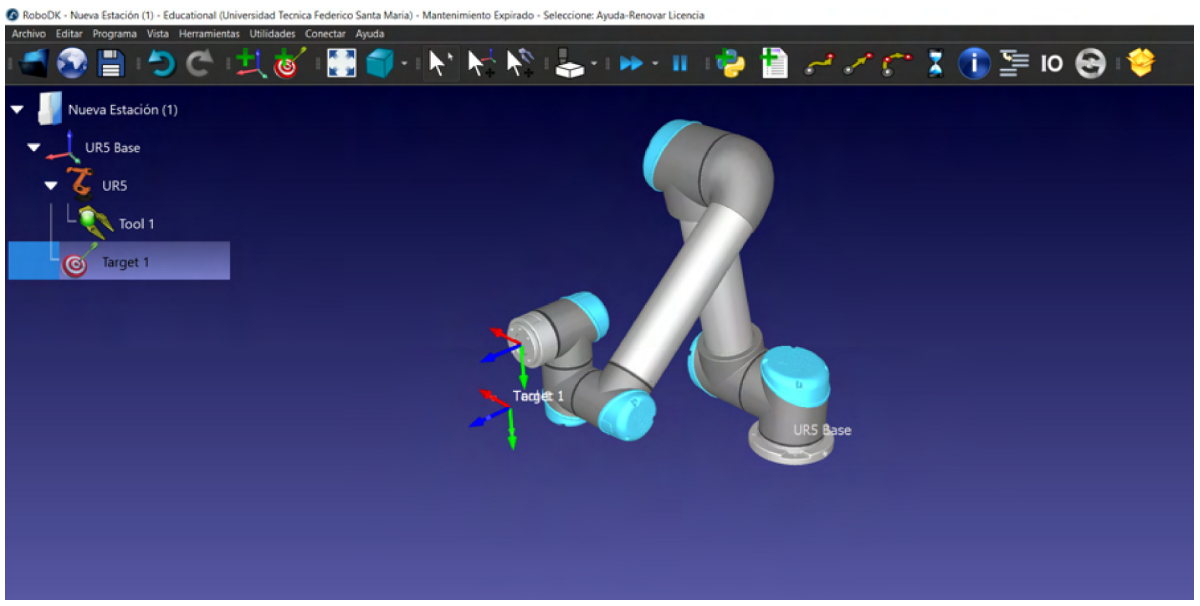


Figura 1.35: Agregar target.

Así entonces, se tiene guardado la posición de la mesa que para el sistema es el espacio del espesor de una hoja entre la cama caliente y la boquilla del extruder. Ahora bien se debe tener una referencia de coordenadas de la pieza de trabajo de la sección. Para esto al igual que el "Target" existe una opción de agregar un "Frame" el cual se encuentra al lado con un icono de coordenadas y un símbolo "+". Se le da click y se agrega una coordenada de trabajo que el software reconocerá como mesa de trabajo para el robot agregado. Este frame no tendrá los valores de la mesa de

trabajo que se necesita, es por esto que se guarda en un target. Para agregar las coordenadas al nuevo frame se debe dar click derecho a en el target que se agrego e ir a opciones, se abrirá una ventana en el lado derecho y luego hacer doble click en el frame agregado donde también se abrirá una ventana en el lado derecho como se muestra en la figura 1.36

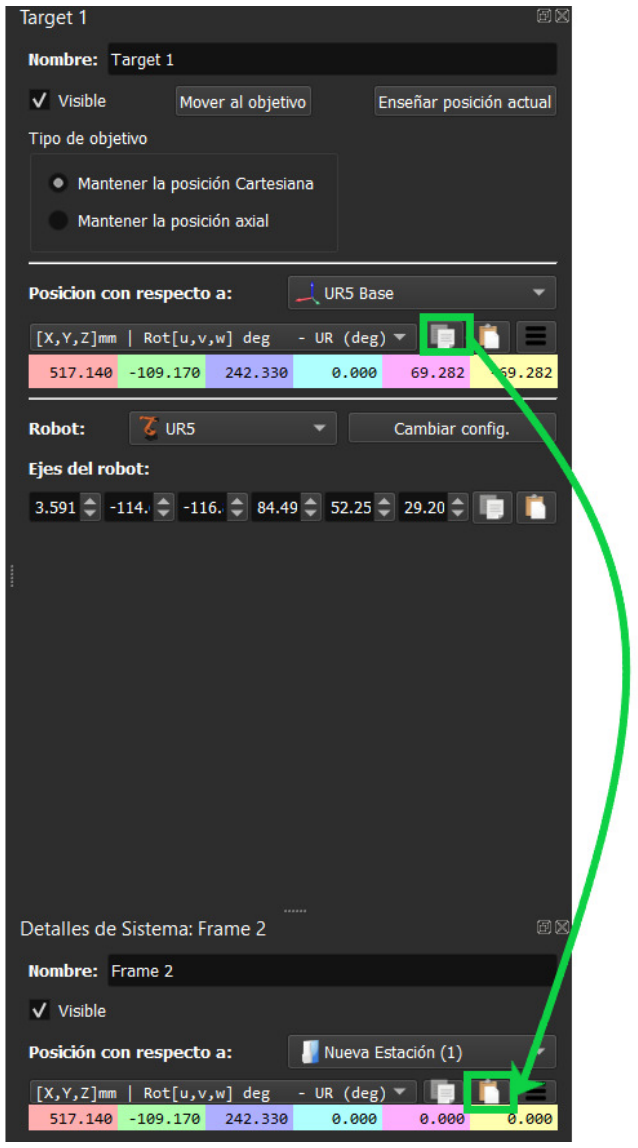


Figura 1.36: Copiar las coordenadas de Target a Frame.

Como se muestra se puede copiar las coordenadas con los iconos marcados para que sea más rápido, pero tener en cuenta que los grados en el frame 2 deben ser cero. Los grados son los 3 últimos números, así el código cargado no estará rotado en algún eje.

Así entonces, se envía el código G, creado en la sección anterior, arrastrándolo a el programa adaptado según las consideraciones mencionadas en los puntos anteriores. Se abrirá una ventana como se muestra en la figura 1.37. En esta ventana es donde se ajustan los movimientos y velocidad máxima del robot. En donde principalmente el enfoque son en la opciones: establecer corriente, enseñar, compensación de la herramienta,eventos del programa, actualizar y simular. Como se muestra en la figura.

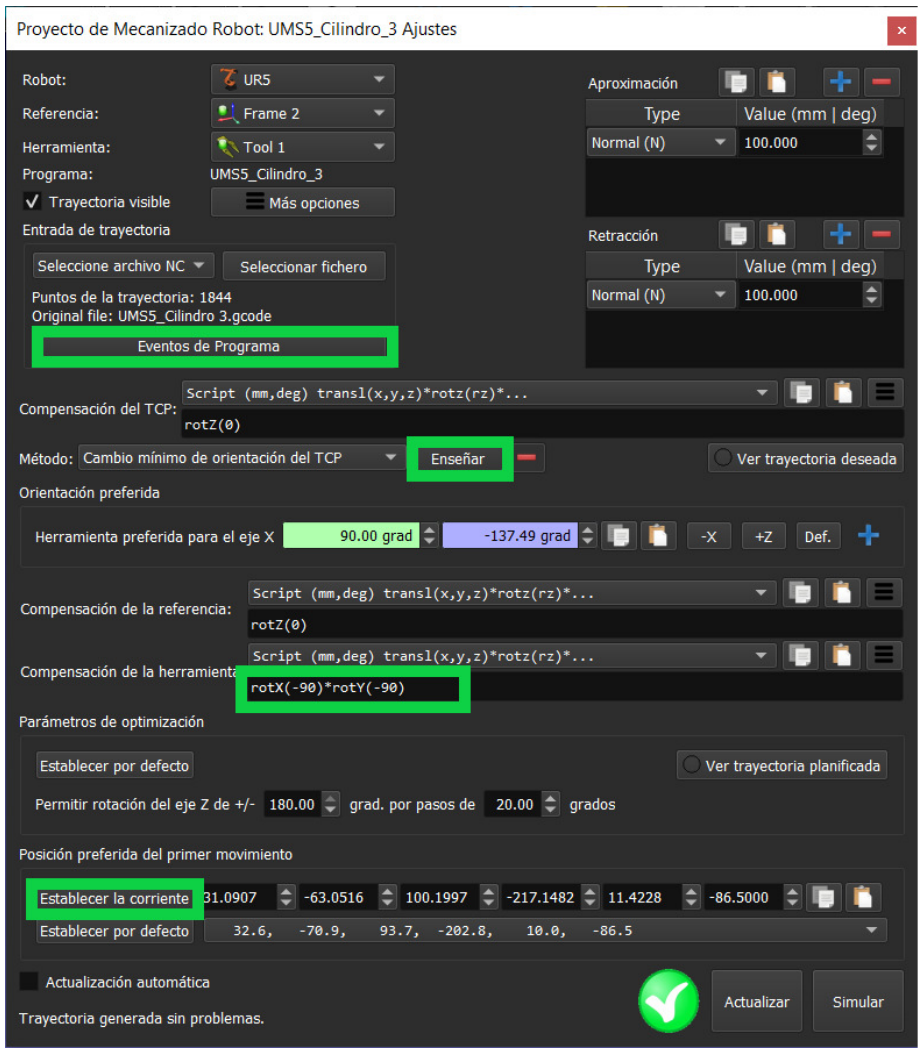


Figura 1.37: Ajustes para programa de impresión 3D.

Las opciones encerradas en un rectángulo verde indican las opciones que se deben o pueden modificar para un correcto funcionamiento y el cuadrado de color rojo indica que la simulación se puede realizar sin problema. Ahora bien primero se debe explicar porqué la importancia de estas opciones encerradas en color verde.

- Establecer corriente: Esta opción le indica al software de donde tiene que empezar a tomar el primer movimiento. Esto es importante debido a que para la impresión 3d no se quiere que el robot se mueva, sobre todo la muñeca número 3 debido a que tendrá una herramienta que siempre tiene que trasladarse en una misma posición.

- Enseñar: Esta opción genera que el software calcule la mejor opción de orientación de la herramienta para la posición establecida, lo que ayuda a reducir movimientos en la muñeca 3 y que se obtengan mejores resultados.

- Compensación de la Herramienta: Esta opción al momento de simular es muy importante ya que rota la herramienta dependiendo los grados que se necesiten para el punto de contacto. Es particularmente útil cuando el punto de interés o punto de acción se encuentra en una ubicación diferente de la punta de la herramienta. Por ejemplo, si se tiene una herramienta con una longitud adicional o una geometría complicada como lo es el extruder de alto volumen,

la compensación de herramienta permite especificar la ubicación exacta donde se produce el punto de acción de la herramienta. Un ejemplo se puede observar en la figura 1.38 a 0° respecto de X e Y donde se observa que el punto de acción entre la herramienta y la mesa de trabajo no esta siendo perpendicular lo que afecta considerablemente al robot ya que el sistema es rigidido y no permite rotación en la muñeca 3. Al contrario si se modifica a -90° en X e Y como se muestra en la figura 1.39 se puede observar que se tiene un resultado mas positivo a lo que se requiere, ya que la muñeca 3 no esta siendo girada ni rotada, es decir esta perpendicular a la mesa de trabajo. Hay que tener en cuenta que no siempre se tendrá que rotar a estos valores los ejes, dependerá de como este posicionado el robot, la pieza de trabajo y las restricciones de movimiento.

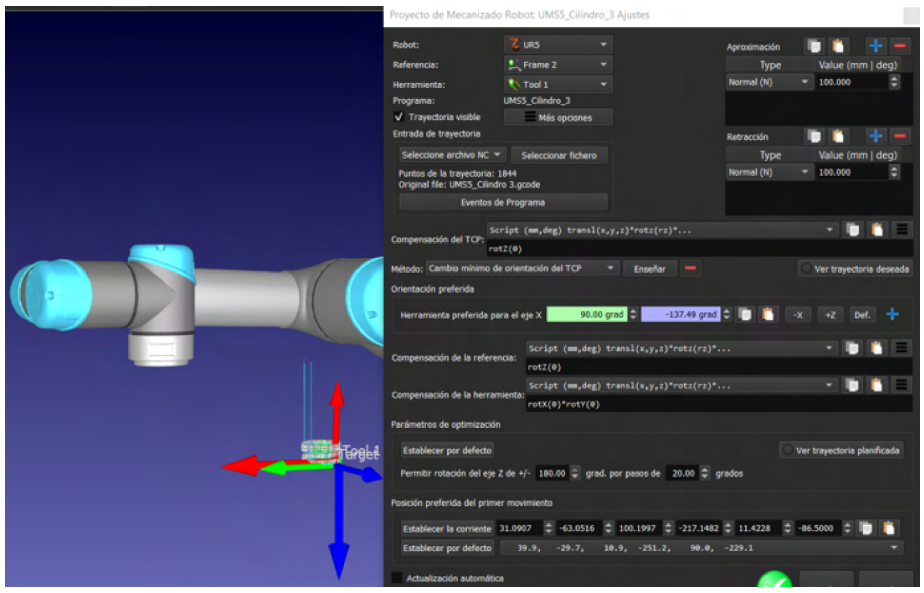


Figura 1.38: Compensación de herramienta a 0° en x e y.

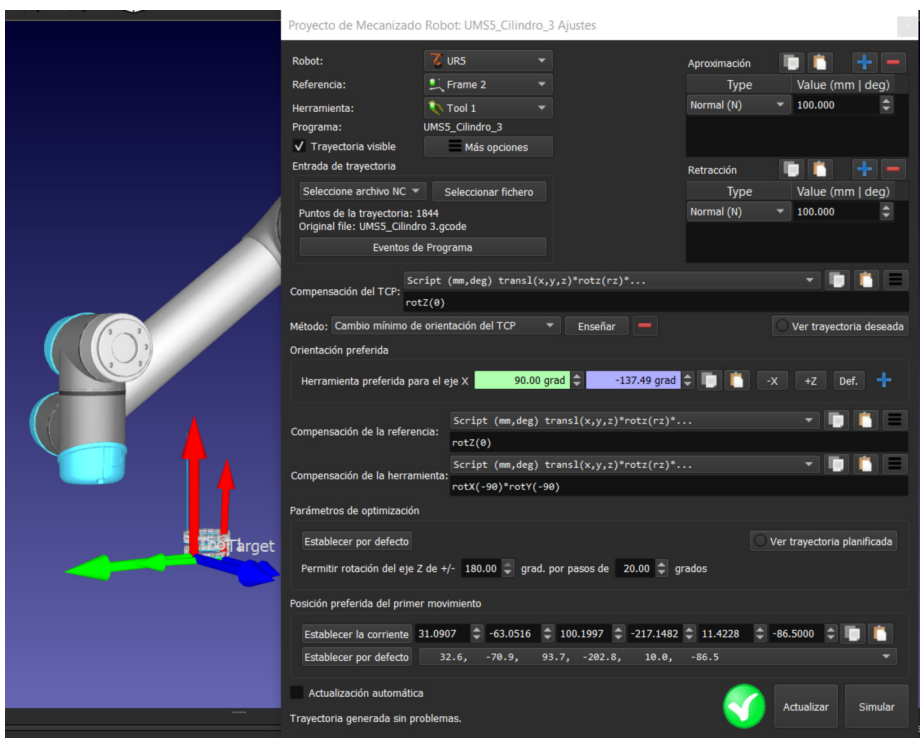


Figura 1.39: Compensación de herramienta a -90° en x e y.

- Eventos del programa: se utiliza para agregar al código una personalización que se requiera en

momentos específicos durante la ejecución de un programa. Estos eventos se pueden utilizar para realizar acciones adicionales, tomar decisiones o interactuar con otros dispositivos o sistemas durante la ejecución del programa.

- Actualizar: Esta opción permite que cada vez que se modifica algún parámetro en los ajustes, se actualice el programa y verifique si es posible hacer o no los movimientos. Esto último se verifica con el símbolo verde, en caso de que no se pueda realizar será de color rojo con una equis.

- Simular: Esta opción es el último paso antes de pasar el programa al robot y sirve para confirmar con la ayuda gráfica que permite el software a través de los movimientos digitales del robot observar si cumplen con los requisitos necesarios para cumplir la tarea en el robot real. Si no se cumplen verificar las opciones anteriores y corregir.

Se tomará en cuenta la velocidad de movimiento rápida como se muestra en la figura 1.40 en [mm/s], esto le indica al robot como moverse en casos que no este ejecutando el programa, es decir se tiene el programa de impresión 3D que inicia en una coordenada, pero el robot se encuentra posicionado en otro espacio, por lo que al momento de ejecutar el programa indicará que se dirija a la coordenada de inicio. Es en este intervalo que el programa por defecto tiene 1000 [mm/s], lo cual moverá al robot muy rápido perjudicando al sistema por lo que se recomienda bajar este valor para este caso a la misma velocidad de impresión, así entonces cuando se posicione en la coordenada de inicio lo hará a una velocidad discreta.

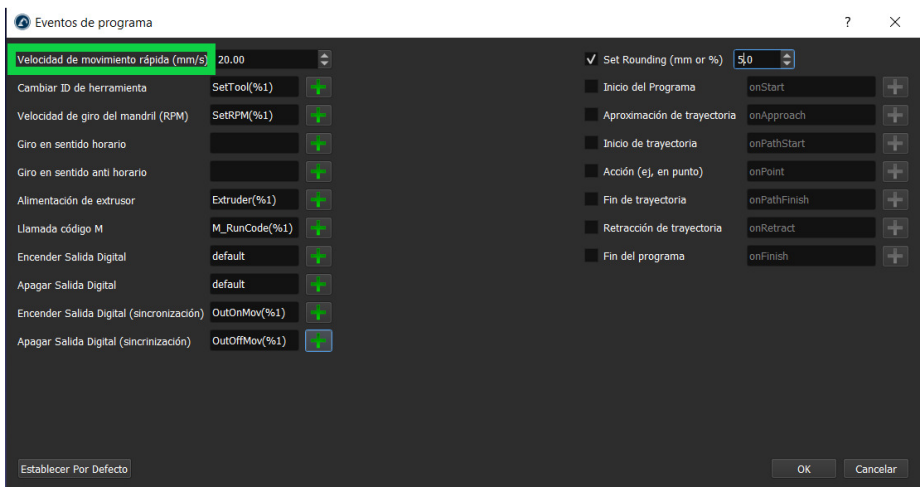


Figura 1.40: Ajustes de eventos de programa de impresión 3D, velocidad máxima.

Teniendo las consideraciones anteriores y verificando que las trayectorias se generaron sin problemas al observar el circulo verde, como se muestra en la figura 1.37 en la parte inferior derecha, se puede pasar a simular y de forma visual verificar que se cumple con las tareas a realizar para la impresión 3D.

Finalmente enviar el programa al robot dando click derecho al programa creado en Robodk como se muestra en la figura 1.41, habilitando la opción de "Ejecutar en el robot" y por último habilitar la opción de "Ejecutar".

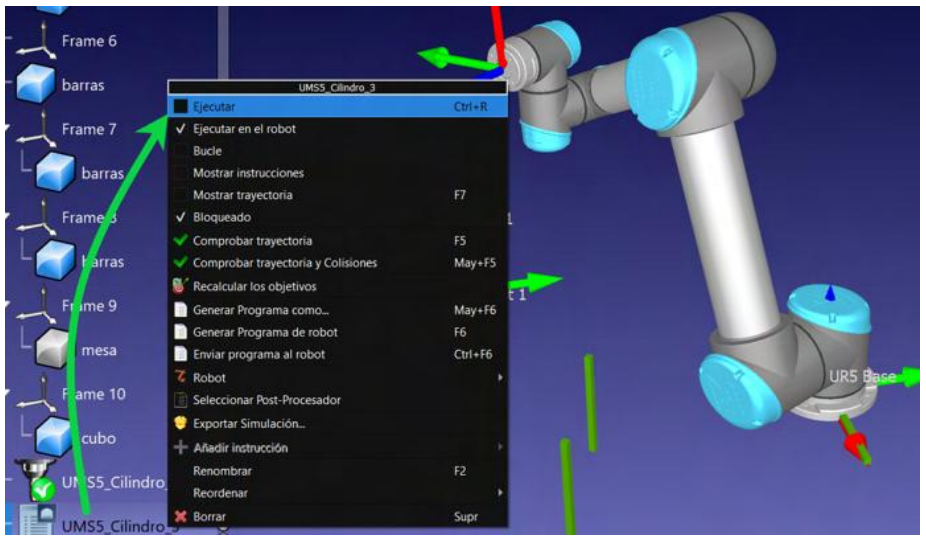


Figura 1.41: Ejecutar programa en el Cobot real.

Se debe tener en cuenta que para modificar las opciones relacionadas a las capas de la impresión, es necesario cambiarlas desde el slicer Ultimaker Cura y repetir el proceso explicado a lo largo de la sección 1.7.3.

1.7.6 Creación de código G con slicer Slic3r

Existe un slicer que funciona junto a RoboDK, el cual se llama slic3 su ventaja es que se puede modificar las capas directamente desde el Robodk sin la necesidad de estar creando un nuevo código G para enviarlo a Robodk y repetir el proceso. En la figura 1.42 se muestra el interfaz de este slic3r.

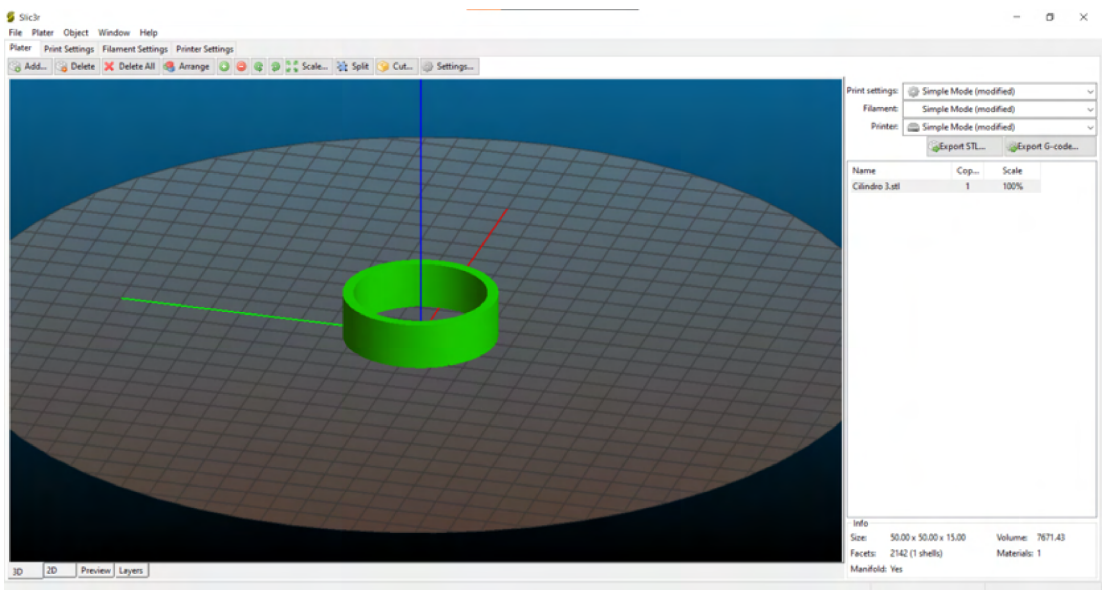


Figura 1.42: Interfaz slic3r.

Para las configuraciones primero se debe ir a configuraciones de impresión (Print Settings) e ir a layers and perimeters como se muestra en la figura 1.43 se debe modificar la altura de capa (Layer height) el cual debe ser como se ha mencionado de 3 [mm] y la altura de la primera capa (Initial layer height) la cual también debe ser de 3 dependiendo de los resultados en la impresión con el cobot real. Otra opción es Spiral vase con esto al igual que con Ultimaker Cura, se realizara el cilindro de forma en espiral, es decir con extrusión del material continuo. Y por último las opciones de Top y Bottom en cero, ya que no se requiere que se tenga superficies superiores e inferiores.

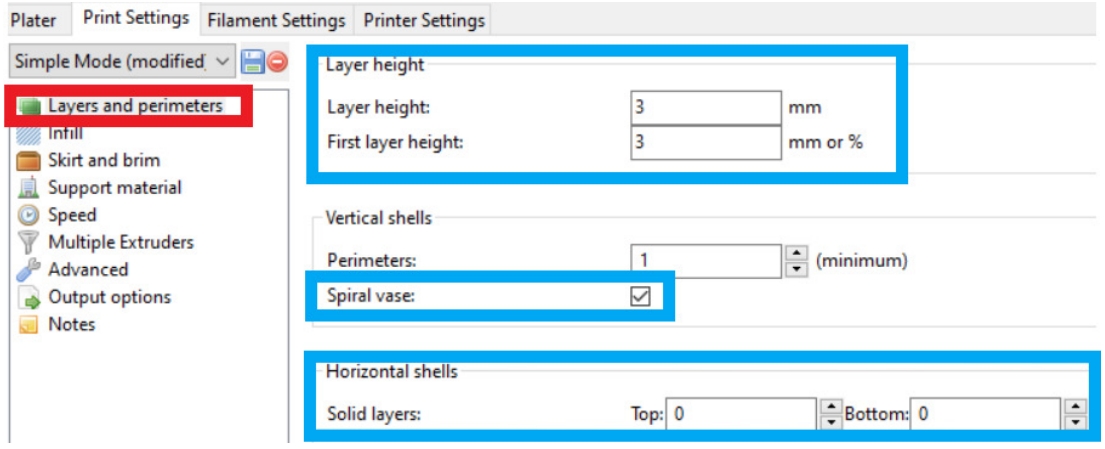


Figura 1.43: opciones de lineas y perímetro.

Normalmente los slicer generan una primera capa alrededor de la pieza a imprimir llamada skirt (falda), esto con el propósito de purgar el material y al momento de realizar la impresión se deposite de manera uniforme. Para este estudio, como se realizaban variadas pruebas y evitar un gasto de material extra se decidió omitir esta opción como se muestra en la figura 1.44 colocando cero en la opción de skirt height (altura de falda).

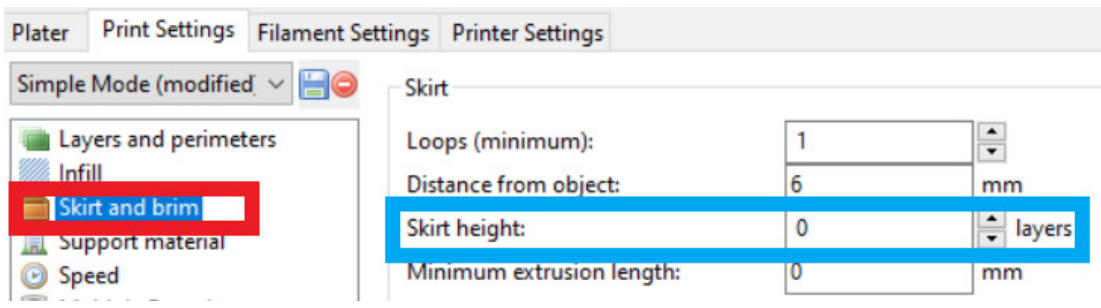


Figura 1.44: opciones falda y ala.

Como última opción en configuraciones de impresión es la velocidad, como se menciono en las secciones anteriores debe ser constante por lo que se deben cambiar a esta velocidad en las opciones de speed como se muestra en la figura 1.45 mientras que las demás estarán por defecto.

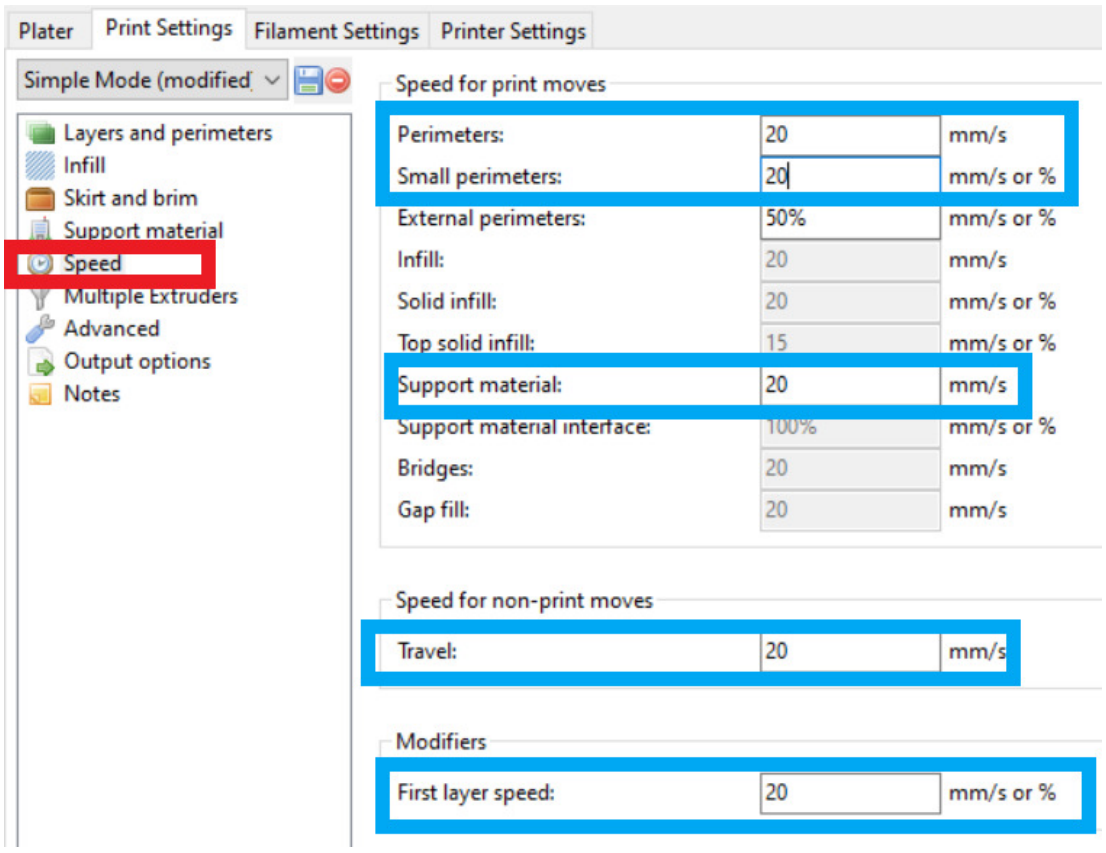


Figura 1.45: opciones de velocidad Slic3r.

Por último ajustar el diametro de la boquilla en la opción Printer Setting y en Nozzle diameter cambiar a 3 [mm].

En la figura 1.46 se muestra las capas creada por el slic3r. Finalmente, se debe exportar en código G para enviarlo a Robodk para seguir la configuración de la impresión 3D.

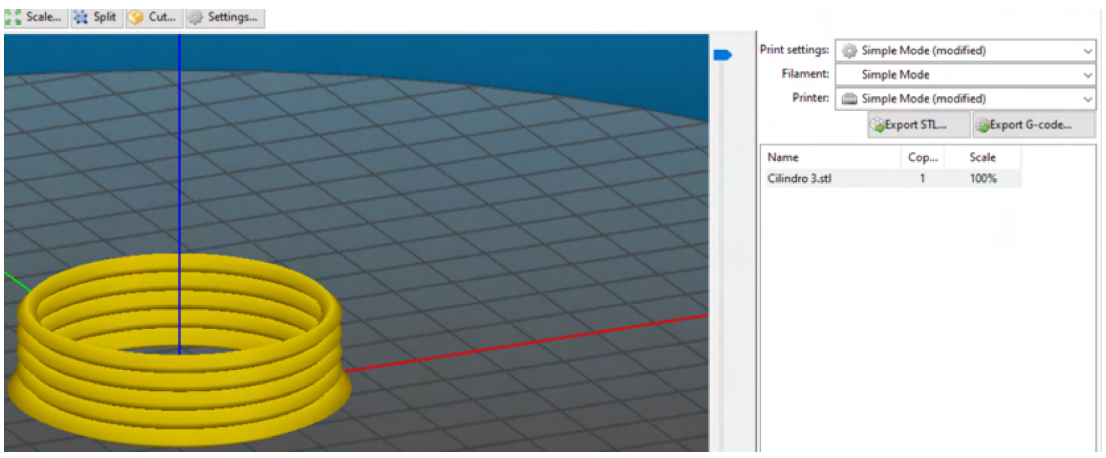


Figura 1.46: Capas creadas por Slic3r.

Ahora para cargar el programa se hace de la misma manera que lo explicado para Ultimaker Cura teniendo en cuenta las mismas consideraciones de velocidad y ajustes de impresión explicado anteriormente con la diferencia que con slic3r se puede modificar desde Robodk las configuraciones de las capas, haciéndolo más cómodo y rápido.

Para poder realizarlo desde Robodk, es necesario instalar la carpeta de slic3r descargada en la carpeta "others" de la carpeta "Robodk" ubicada en la carpeta del equipo "OS (C:)". Luego en

Robodk al instalar el código G creado por Slic3r se abre la venta y se puede seleccionar la opción imprimir objeto 3D como se muestra en la figura 1.47.

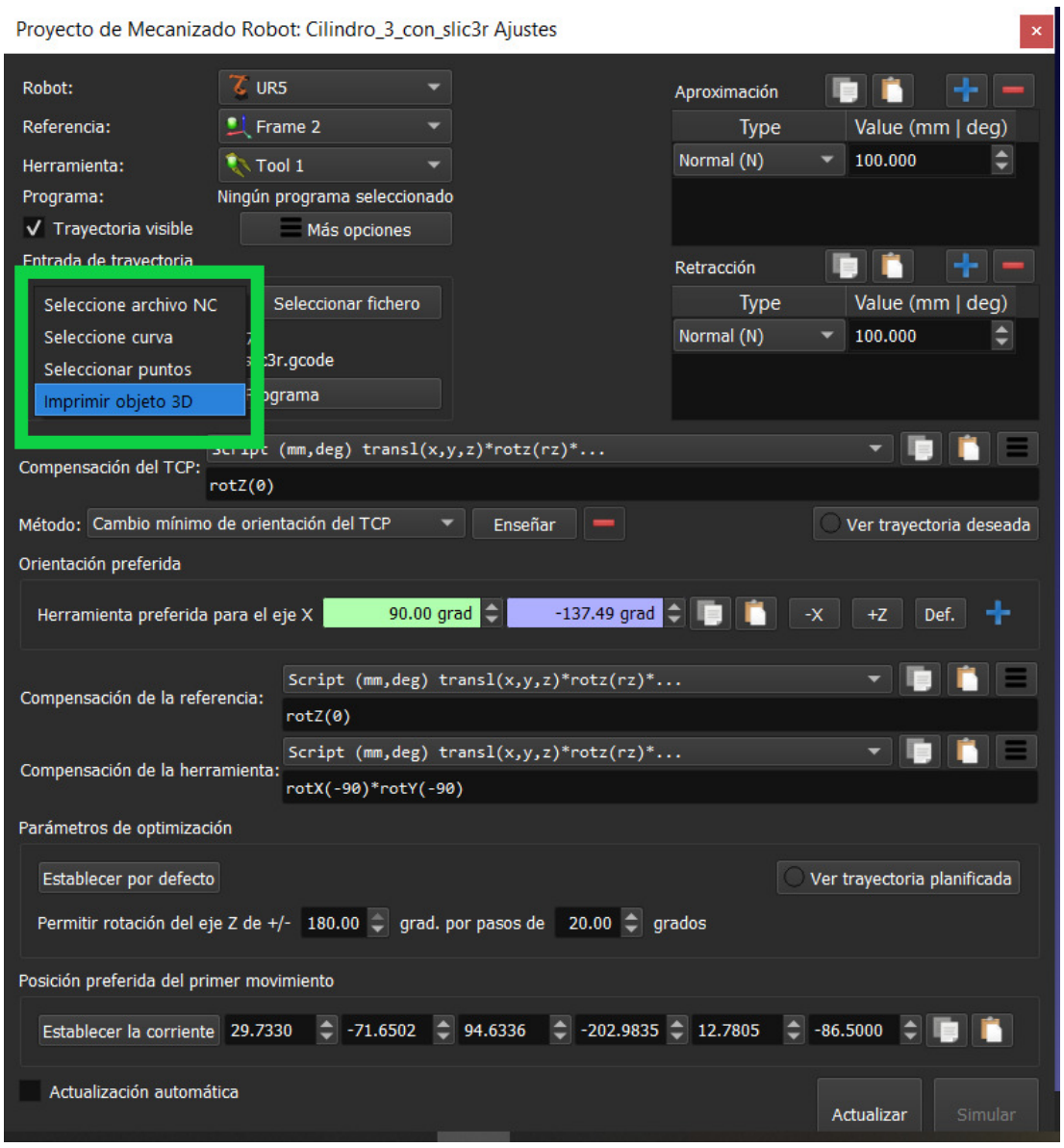


Figura 1.47: Opción para modificar las capas desde Robodk.

Al seleccionar opciones de impresión 3D se abrirá una ventana como se muestra en la figura 1.48, la cual muestra las configuraciones de impresión para ser modificadas directamente de RoboDK, evitando pasos extras y repetitivos. Al modificar algún valor en las configuraciones se cierra la ventana y ya esta modificado en el código G y basta con actualizar el programa en Rododk para utilizar estos nuevos valores en la impresión 3D.

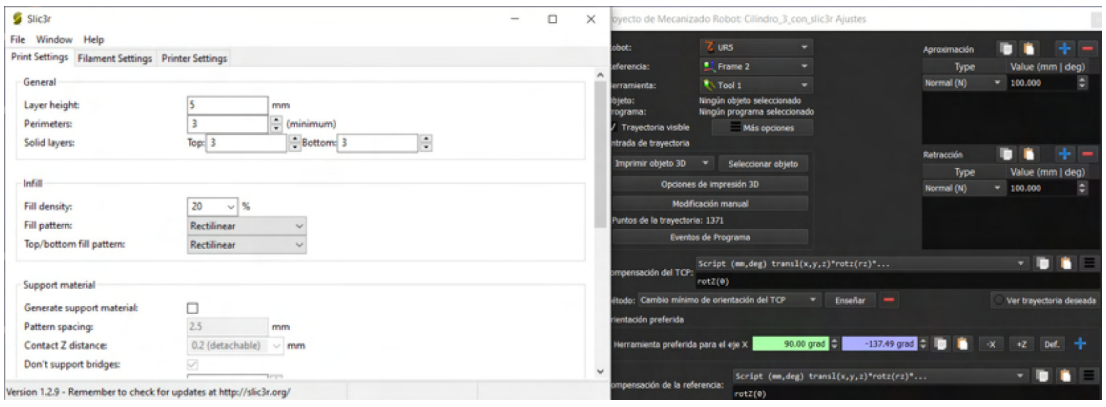


Figura 1.48: Opción para modificar las capas desde Robodk.

La ejecución en el robot es igual a lo explicado para Ultimaker Cura. La diferencia entre estos 2 slicers es que slic3r trabaja directamente con robodk y hace más rápido los pasos al querer modificar tanto las capas como velocidades de impresión, en cambio Ultimaker Cura se necesita cambiar desde el Slicer para luego repetir el proceso de cargar el programa en Robodk y ajustar las condiciones de trabajo.

Capítulo 2

Mantenimiento del Sistema

2.1 Universal robot UR5

En la figura 2.1 se muestra un robot UR5 de 6 ejes, este modelo es anterior al modelo UR5e con una carga máxima de 5 kg con 15 funciones avanzadas de seguridad las cuales son regulables y probado de acuerdo con las normas ISO 13849:2008 PL d .

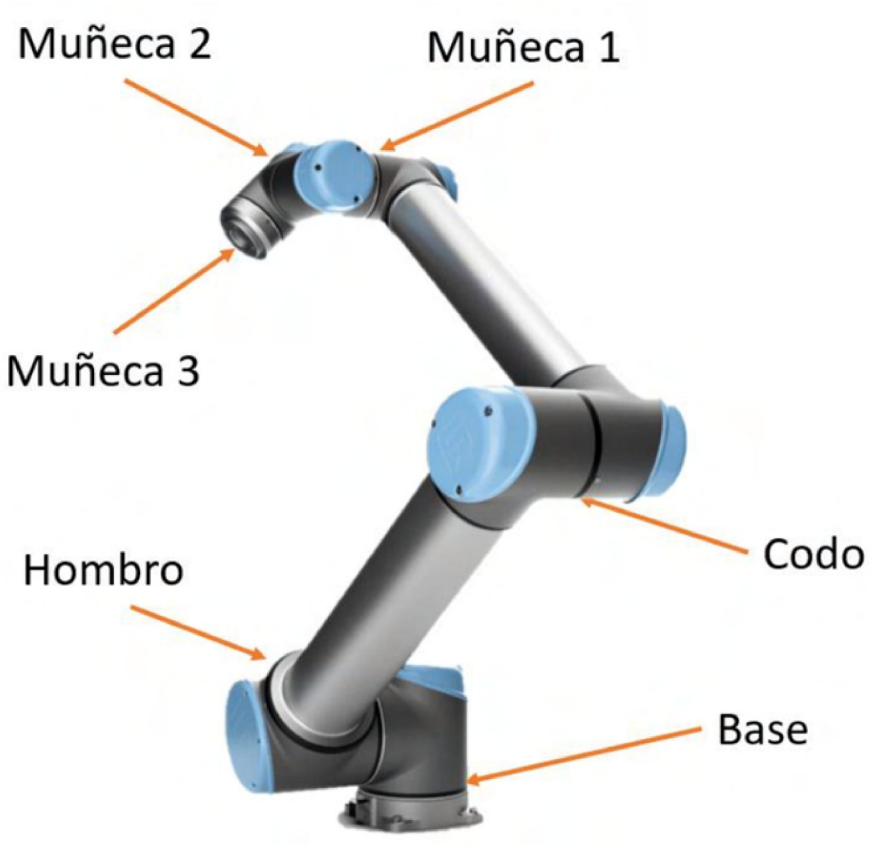


Figura 2.1: Cobot UR5.

2.1.1 Especificaciones

La carga útil que el robot puede soportar, el alcance máximo que tiene el brazo extendido de 850 [mm], los grados de libertad las cuales viene dado de las 6 articulaciones mostradas en la figura 2.1 las cuales pueden girar +- 360 grados cada una como se indica en la tabla 2.2 esto quiere decir que la articulaciones al momento de llegar al grado +360° debe devolverse a -360° lo mismo viceversa y para controlar el robot se tiene una pantalla táctil con soporte la cual permite al operario ocupar las diferentes funciones incluidas en el robot y además verlo de manera gráfica en el monitor.

Tabla 2.1: Especificación

Carga útil	5 kg / 11 lb
Alcance	850 mm / 33,5 in
Grados de libertad	6 articulaciones giratorias
Programación	Interfaz gráfica del usuario Polyscope con pantalla táctil de 12” con soporte

Tabla 2.2: Movimiento

Movi. del eje del brazo Robot.	Radio de acción	Velocidad máxima
Base	+-360°	+-180°/s
Hombro	+-360°	+-180°/s
Codo	+-360°	+-180°/s
Muñeca 1	+-360°	+-180°/s
Muñeca 2	+-360°	+-180°/s
Muñeca 3	+-360°	+-180°/s
Herramienta típica		1 m/s - 39,4 in/s

Las características físicas de UR5 presentadas en la tabla 2.3 indican principalmente que su peso es de 18,4 [kg] lo cual es ligero comparado con su volumen, longitud del cable del brazo robótico de 6 metros el cual permite operar el robot a una distancia considerable de la fuente de poder y material de aluminio y plásticos de PP por lo que se debe tener cuidado sobre las cargas externas al robot debido a que se podría dañar considerablemente.

Tabla 2.3: Características Físicas

Tamaño de la caja de control (an. x al. x la.)	475 x 423 x 268 mm/ 18,7 x 16,7 x 19,6 in
Peso	15 kg / 33,1 lb
Materiales	Acero

2.1.2 Montaje

Para montar el brazo robótico se utiliza pernos M8 en los cuatros orificios de 8.5 [mm] de la base. El fabricante recomienda apretar estos pernos con un par de torsión de 20 Nm. Para un montaje preciso, se suministran dos orificios de diámetro 8 [mm] para utilizar como clavija. La figura 2.2 se muestra dónde taladrar los orificios y montar los tornillos.

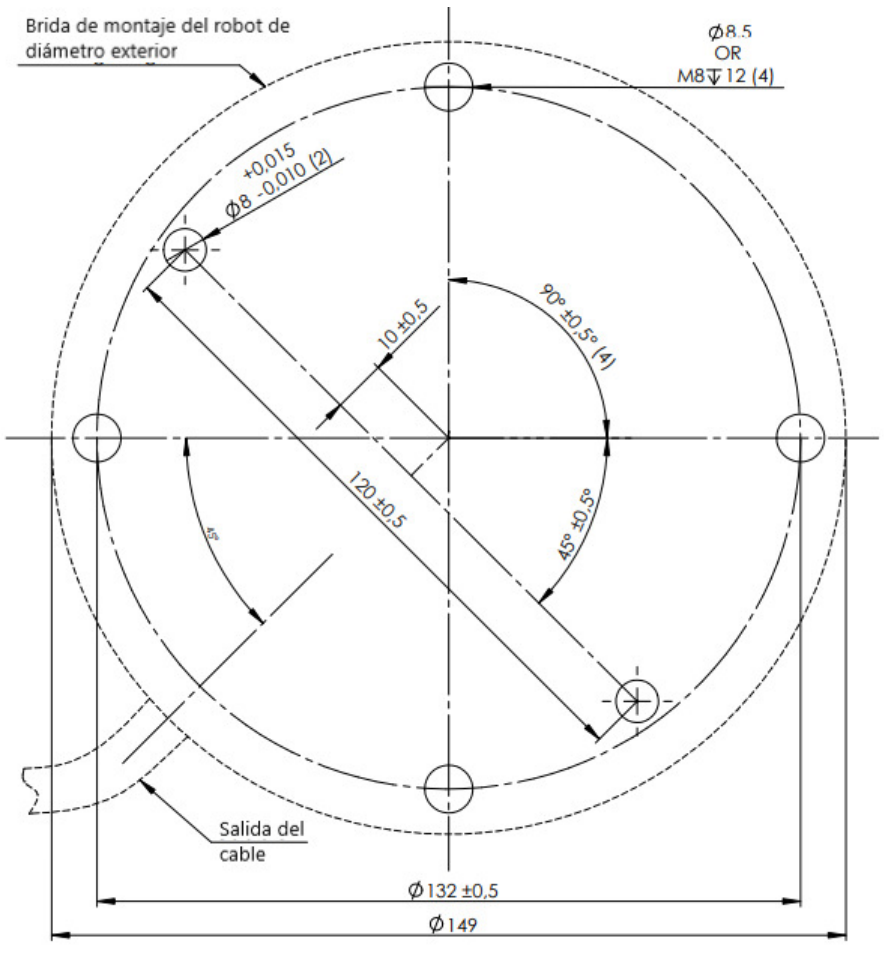


Figura 2.2: Orificios de montaje con pernos M8. medidas en mm

La superficie donde se debe instalar el robot debe ser resistente, lo bastante fuerte para soportar al menos 10 veces el par de torsión total de la junta de la base y al menos 5 veces el peso del UR5. Se debe tener en cuenta que no debe haber vibraciones, debido a que afectará el funcionamiento optimo y puede activar las paradas de emergencias. Además, si el robot se monta sobre una plataforma móvil, la aceleración de la base de montaje móvil debe ser muy baja. Ya que una aceleración alta puede hacer que el robot se pare por pensar que ha chocado con algún elemento externo.

2.1.3 Herramienta

El robot cuenta con una brida donde se coloca la herramienta, la cual consta con cuatro orificios de rosca M6 para acoplarla al UR5. Los pernos deben ajustarse con 9 Nm. En caso de que se

desee volver a colocar de forma muy precisa la herramienta, se suministra el orificio de diámetro de 6 [mm] para utilizar como clavija. En la figura 2.3 se muestra el esquema de los orificios con roscas para montar los tornillos de la brida.

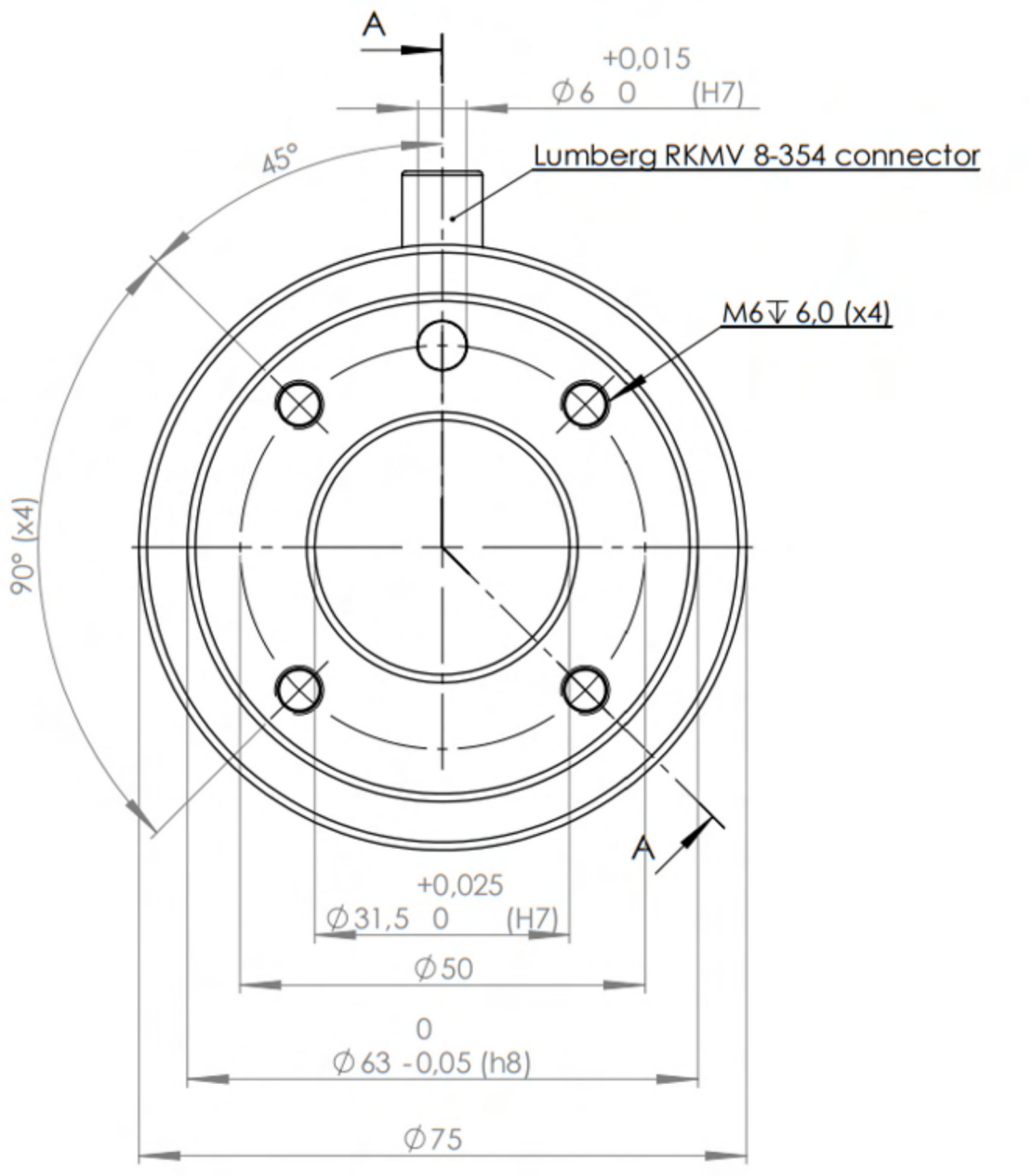


Figura 2.3: Brida de salida de la herramienta, donde se monta la herramienta. Todas las medidas están en mm .

2.2 Sistema

2.2.1 Unión Sistema

Para la unión se tiene una placa de acero y una bisagra para permitir el movimiento de los 3 elementos (UR5, extruder de alto volumen y brazo compensador) como se observa en la figura 2.4. Esta unión se debe tener cuidado de colocar de forma correcta los pernos adecuados, la manera óptima es colocar primero los pernos a la muñeca 3 del brazo robótico para luego colocarlo a la placa de acero, ambos con golillas. Esta conexión debe quedar lo más nivelada posible e intentando

que queden paralelo entre sí. Esto para evitar esfuerzos sobre el robot y esfuerzos cortantes que puedan provocar que se sobre exija los pernos y se corten.

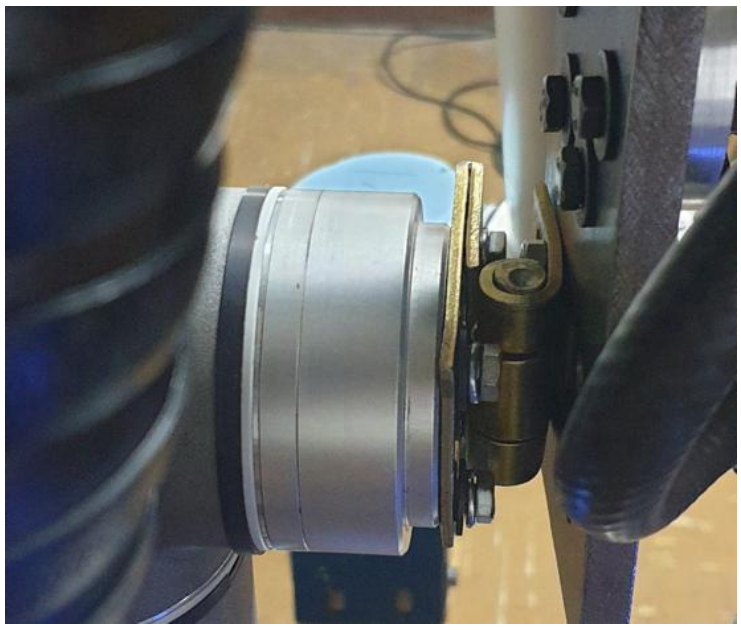


Figura 2.4: Unión.

2.2.2 Refrigeración de extruder de alto volumen

Para evitar que se sobre caliente el extruder por las altas temperaturas en la boquilla se es necesario que se enfríen las partes que no se deben calentar. Esto debido a que existe una conexión directa con el brazo en las partes cercanas a la boquilla, si bien es cierto que no se calienta lo suficiente pero para una cantidad considerable de tiempo en funcionamiento estas temperaturas pueden ocasionar que el cobot resulte perjudicado. Ya que sus mismos servomotores generan calor y existe la posibilidad que se caliente y que el sistema integrado de UR5 apague el cobot para impedir que siga funcionando.

Esta refrigeración se realiza por una bomba de agua de acuario conectada a los puntos de refrigeración que tiene integrado el extruder. se conectan a presión procurando que el agua no este con partículas contaminantes que puedan perjudicar la vida útil tanto de la bomba como los canales de refrigeración del extruder.

2.2.3 Mover UR5 con PCH del Extruder

Para prevenir que al mover el cobot con el sistema integrado arroje parada de emergencia o se mueva de manera que la muñeca 3 gire y genere esfuerzos en la unión se debe mover desde "Posición PCH"(ver figura 2.5), esto hará que el brazo robótico se mueva manteniendo la posición de la herramienta, es decir perpendicular a la mesa caliente, otra manera es moverlo desde cada

una de sus partes en la opción "Mover juntas". Esto agregado a que se recomienda mover el cobot a una velocidad reducida para no tener vibraciones del sistema, para esto se puede reducir el porcentaje de velocidad, por lo que se recomienda moverlo a un porcentaje de 50% o menor.

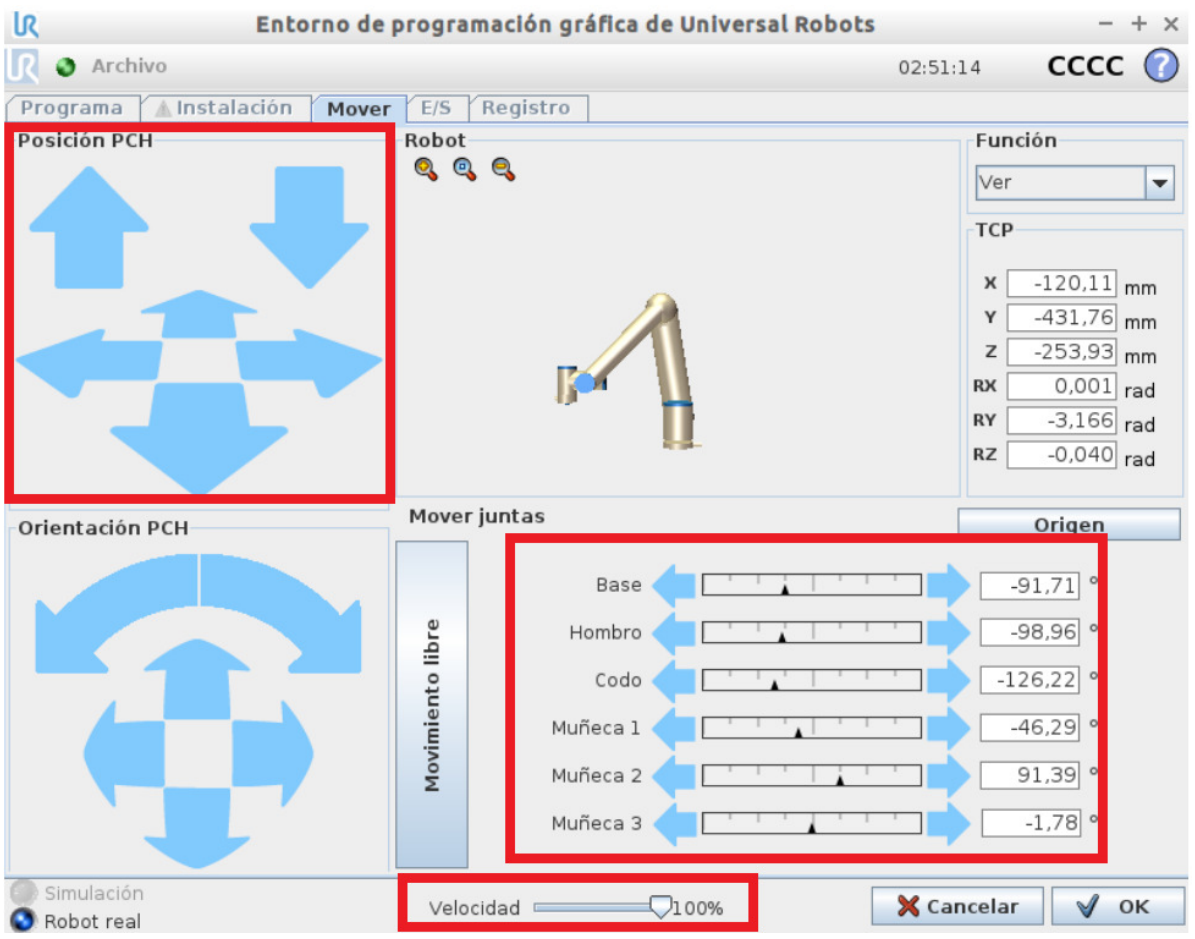


Figura 2.5: Interfaz de movimientos recomendados para mover Cobot en el sistema.

2.2.4 Mover UR5 libre

Cuando se requiera mover el cobot, sin el sistema (sólo), a una posición deseada como por ejemplo a la unión con el extruder existe un botón físico al reverso de la pantalla digital. Al presionar esté botón se debe tener cuidado que el robot no tenga fuerzas externas ya que se liberarán los frenos y puede causar un movimiento brusco del brazo robótico que puede perjudicar a una persona cercana o al mismo robot como también algún objeto en el espacio de trabajo. Teniendo esto en cuenta se podrá mover de manera manual a donde se requiera de manera más rápida.

No se debe mover libre cuando el sistema este conectado, ya que el robot puede tener fuerzas externas mínimas pero importantes que al momento de liberar los frenos, puede generar un movimiento brusco perjudicando el área de trabajo.

2.2.5 Importancia de simular antes de ejecutar

Para prevenir que el programa cargado de impresión 3D siempre se acomode a las limitaciones de trabajo, como lo es la restricción de rotar el extruder. Se debe siempre tener en consideración que al momento de ejecutar antes se debe simular y verificar que la herramienta no se gire y siempre se mantenga perpendicular a la mesa de trabajo. Para poder acomodar el programa a las limitaciones se debe leer la sección 1.7.5 del capitulo 1, así entonces se previene que el cobot tenga exigencias de esfuerzos y que su vida útil sea perjudicada por un mal funcionamiento.

2.2.6 Velocidades altas en secciones pequeñas

Las velocidades altas en secciones pequeñas resulta exigente para el cobot, ya que debe acelerar y desacelerar en un corto tiempo. Esto muchas veces viene dado, en el contexto de impresión 3D, en una mala resolución de un objeto 3D digital. Esto produce que los slicer tomen las curvas como líneas rectas pequeñas y que al momento de crear y ejecutar el programa el cobot se mueva de manera que los frenados para el cambio de dirección generen vibraciones que perjudicarán al resultado y que para un tiempo prolongado puede parar el brazo robótico por altas variaciones en los servomotores y que el sistema del mismo se proteja. Es por esto que se debe verificar en exportar objetos en alta resolución y observar que el slicer no genere lineas rectas en donde debería existir una curva.

2.3 Actualizaciones Softwares

2.3.1 Actualización software Universal Robots 5

Para los robots de la serie CB, la actualización debe realizarse de manera de incremento, es decir tomar la versión actual del robot e ir subiendo una por una hasta llegar a la actualización final del software entregado por Universal Robots. El primer paso entonces, es comprobar la versión de PolyScope y, a partir de esta, actualizar a la siguiente versión disponible hasta llegar a la última versión del software de Polyscope.

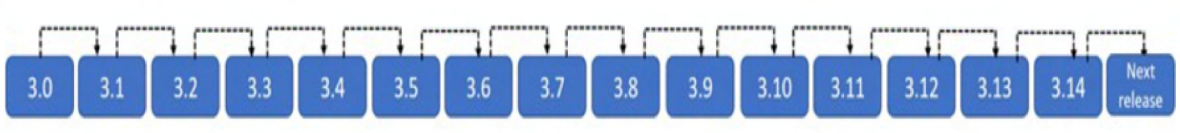


Figura 2.6: Orden recomendado por UR para actualizacines de softwares serie CB.

Para saber el software actual del robot se debe ir a "about" como se muestra en la figura 2.7 donde se abrirá una venta como se muestra en la figura 2.8 con un ejemplo de la versión actual que tiene el sistema.

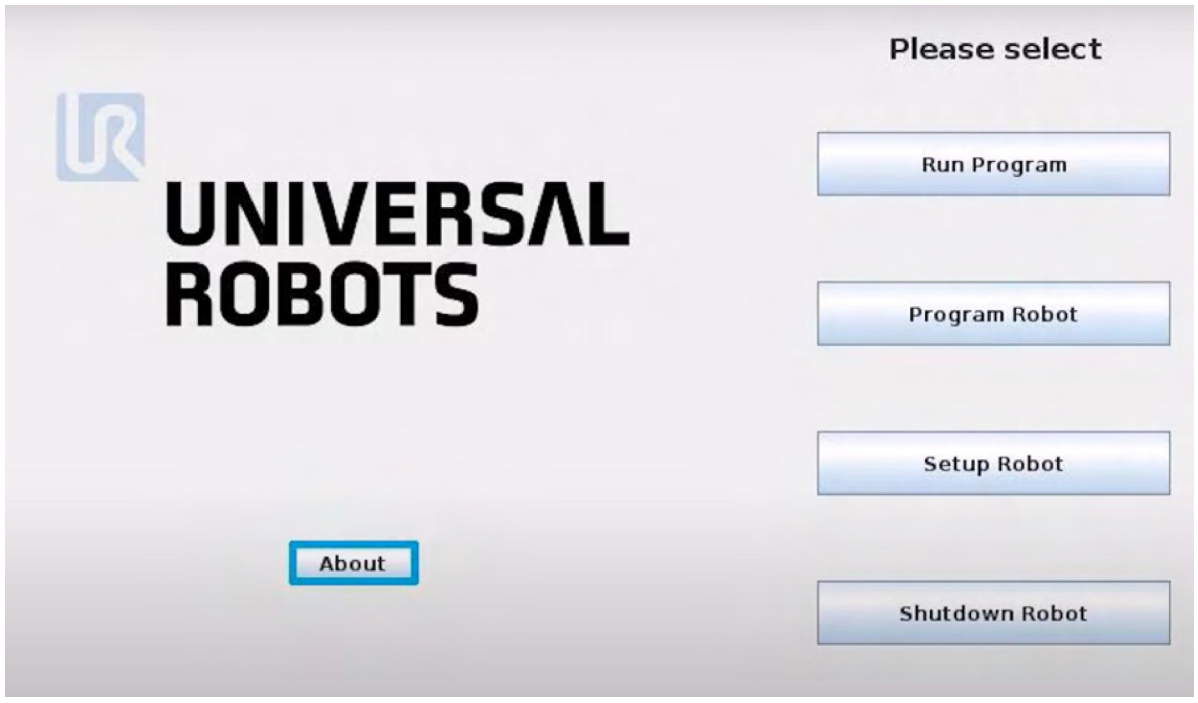


Figura 2.7: Acerca del sistema actual del cobot.

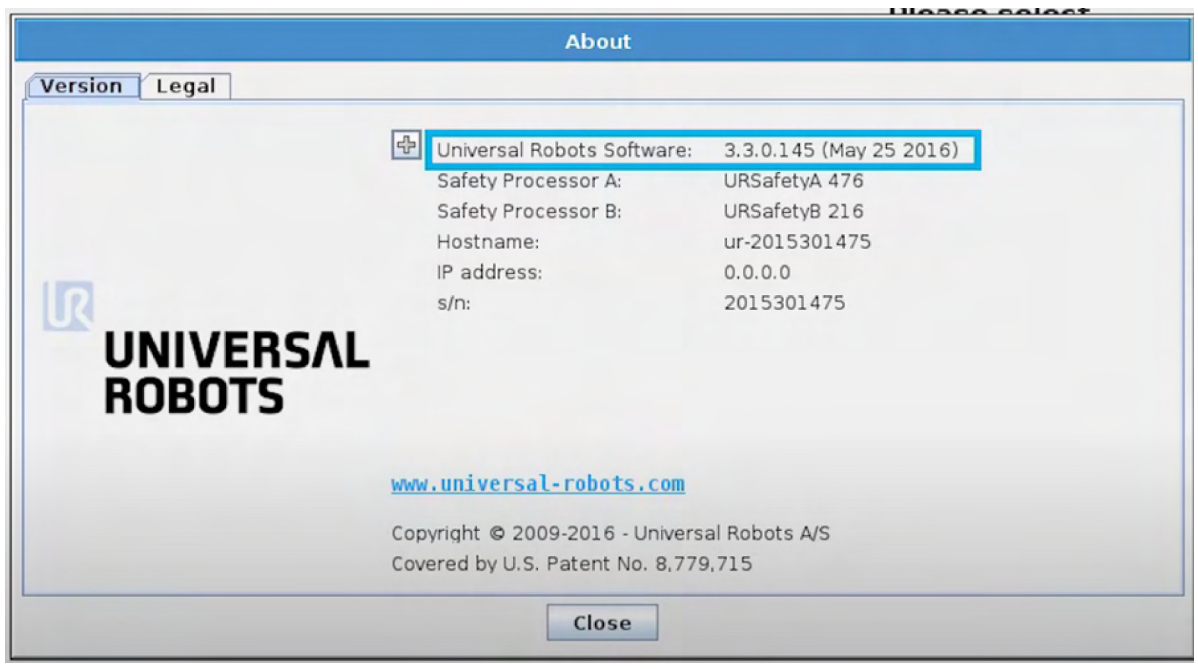


Figura 2.8: Ejemplo de versión actual del sistema.

Luego de comprobado el sistema se debe ir a las actualizaciones entregadas en la página web de Universal Robots en el siguiente link https://www.universal-robots.com/articles/ur/documentation/legacy-download-center/, con esto se deben descargar todas las versiones de las actualizaciones a partir de la versión actual del cobot. Luego de descargadas se deben transferir a un pendrive USB y conectarlo a la entrada usb integrada en la parte lateral de la pantalla táctil.

Una vez conectado el pendrive, se dirige a configuraciones y luego a Update Robot. Aquí debería

aparecer todos los archivos cargados de las versiones de los softwares como se muestra en la figura 2.9. Se debe seleccionar la versión siguiente a la actual, donde aparecerá una ventana para actualizar a dicha versión dar click y se actualizará automáticamente. Finalmente revisar la versión actual del robot para verificar la actualización se haya realizado. Para más información y ayuda pueden revisar el siguiente vídeo explicativo https://l1nq.com/VgQT1.

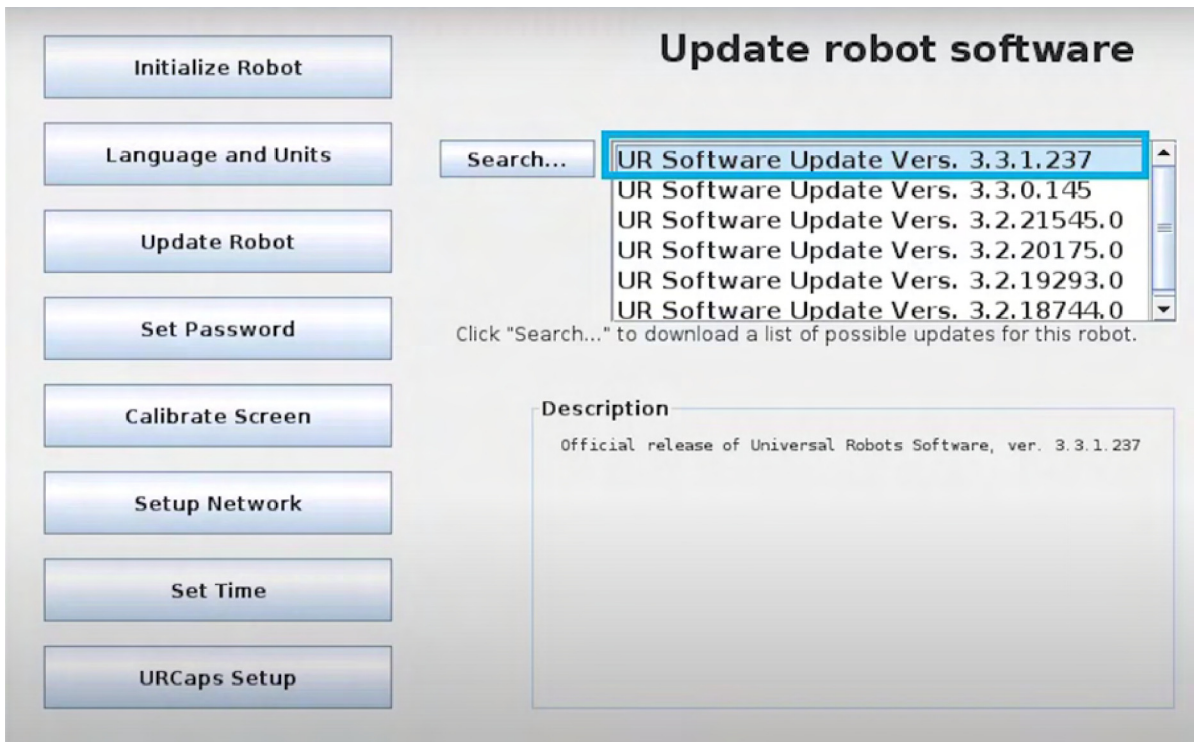


Figura 2.9: Archivos con las actualizaciones.

Nota 1: Siempre después de instalar la actualización, se debe encender los motores para que la actualización completa se complete antes de pasar a la siguiente actualización. Omitir este paso puede tener influencia en la finalización final del software.

Nota 2: Internet Explorer en algunos casos detectará automáticamente el archivo .urup como un archivo .xml debido al manejo MIME "inteligente" de los tipos de archivo. Si este es el caso, asegúrese de seleccionar Guardar como tipo: Todos los archivos y luego ingrese manualmente el nombre de archivo y la extensión correctos, es decir. actualizar.3.xxxurup

2.3.2 Actualización software Robodk y licencia educacional

Para la actualización del software RoboDk, se puede descargar directamente de la página oficial de RoboDK, pero teniendo en cuenta que el laboratorio ECO-FAB-LAB, tiene licencia estudiantil. Esta licencia tiene como condición de mantenimiento hasta marzo del año 2023, si bien es permanente para actualizaciones después de esta fecha no funcionará dicha licencia. Para esto es necesario tener la última versión antes de marzo, el cual es la versión 5.5.1, el cual se puede descargar del siguiente link https://robodk.com/downloads-v/Install-RoboDK-64-v5.5.1.exe .

Para ocupar la licencia educacional se debe ir a ayuda en las opciones en la parte superior de la ventana de RoboDk, luego dar click a licencia y se abrirá una ventana como se muestra en la figura 2.10, se debe seleccionar red, verificar que se este conectado a la red de internet y colocar la siguiente licencia: 4186-12F9-7D9D-D5C3-6D83-1496-A056-0A81. Una vez puesta debería reconocerla y estar lista para su uso.

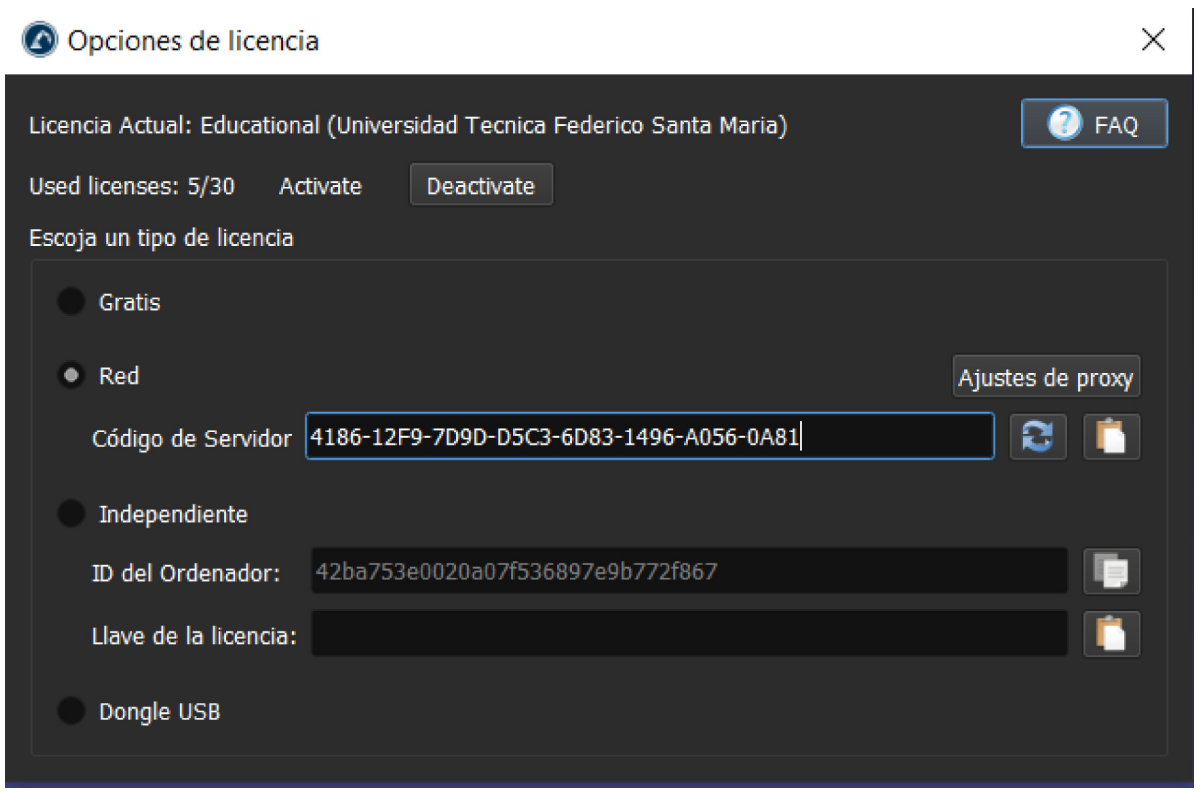


Figura 2.10: Licencia educacional para RoboDK.

Nota: Para personas que recién esten instalando el software en su computadora la empresa entrega 30 dias de licencia gratis.

2.3.3 Actualización softwares slicers

Las actualizaciones de los slicers, no tienen mayor problema. se hacen directamente de sus páginas oficiales. Para Ultimaker Cura con el siguiente link: https://ultimaker.com/software/ultimaker-cura/ , y para slic3r con el siguiente link: https://slic3r.org/ .