



Universidad de Valladolid

**ESCUELA DE
INGENIERÍAS INDUSTRIALES**

Grado en Ingeniería Electrónica Industrial y Automática

**Simulador de movimientos de herramientas
quirúrgicas en operaciones de cirugía
laparoscópica con robots**

Alumno: Alejandro Sancho Gil

Tutor: Juan Carlos Fraile Marinero

Abril de 2022

Agradecimientos

Lo primero, agradecer a mis padres y a mi tía por el gigantesco apoyo y ayuda que siempre me han brindado y por todo lo que me han enseñado.

Sobre todo, a mi padre, porque siempre ha estado ahí, tanto en los momentos buenos como en los malos durante toda la carrera. Sin él, nada de esto podría haber sido posible y absolutamente todo ha sido gracias a él.

Además, quería agradecer a todos mis amigos de la carrera que siempre han confiado en mí, me han ayudado en incontables momentos sin dudarlos y me han apoyado muchísimas veces, me sentiré eternamente en deuda con ellos.

Igualmente, a todos mis amigos fuera de la carrera y familiares que siempre han tenido fe en mí, han sido los que me han dado muchas fuerzas durante toda la carrera, siempre les valoraré muchísimo y también les estaré eternamente agradecidos.

También, agradecer a mi tutor Juan Carlos Fraile por las directrices que me ha dado, facilitar el desarrollo de este trabajo y ayudar a tener los mejores resultados posibles en este proyecto.

Resumen

En este TFG, se aborda la creación de un entorno quirúrgico con el robot ABB IRB 120, utilizando el software de simulación *RobotStudio*. Para ello, se ha diseñado el modelo de un *Pelvitainer*, y se ha realizado un software que permite simular y planificar los movimientos de las herramientas quirúrgicas en operaciones de cirugía laparoscópica con robots. Utilizaremos el brazo robótico *ABB IRB 120* para simular los movimientos del instrumental quirúrgico en una operación de cirugía laparoscópica.

Palabras claves

Robots, Cirugía laparoscópica, Planificación de movimientos, RobotStudio, Pelvitainer.

Abstract

In this dissertation, the creation of a surgical environment with the *Robot ABB IRB 120* is addressed, using the simulation software *RobotStudio*. For this purpose, the model for a *Pelvitainer* and a software programme, that allows both to simulate and to plan the movements of the surgical material in a robot-performed operation of laparoscopic surgery, have been designed. We will use the robotic arm *ABB IRB 120* to simulate the movements of the surgical material in an operation of laparoscopic surgery

Keywords

Robots, Laparoscopic Surgery, Movement planning, RobotStudio, Pelvitainer.

Índice de la memoria

1. Introducción y objetivos	1
1.1 Objetivos del proyecto.....	1
1.2 El instrumento <i>Pelvitainer</i>	2
1.3 Estructura de la memoria	4
2. Breve introducción a la robótica aplicada a la cirugía laparoscópica.....	5
2.1 La cirugía laparoscópica	5
2.1.1 Historia de la cirugía laparoscópica	7
2.1.2 Ventajas de la cirugía laparoscópica	12
2.2 Robots aplicados a la cirugía laparoscópica.....	13
2.2.1 Robot da Vinci	13
2.2.2 Hugo RAS de Medtronic.....	14
2.3 Beneficios de aplicar la robótica a la cirugía laparoscópica.....	16
2.4 Estudios sobre la cirugía laparoscópica y robótica aplicada a esta	18
3. Desarrollo del entorno de simulación	19
3.1 Robot ABB IRB 120	19
3.2 Modelado del <i>Pelvitainer</i> y los demás elementos de la estación	22
3.2.1 Modelado en 3D del <i>Pelvitainer</i>	22
3.2.2 Modelado en 3D de las herramientas quirúrgicas.....	24
3.2.3 Modelado en 3D de los soportes de las herramientas	27
3.2.4 Modelado en 3D del vallado	30
3.3 Creación de la estación de trabajo para simular el robot en un entorno de cirugía laparoscópica	32
3.3.1 <i>Pelvitainer</i> en la estación de trabajo	35
3.3.2 Herramientas quirúrgicas y su almacén en la estación de trabajo.....	37
3.3.2.1 Herramientas negras y sus soportes	37
3.3.2.2 Herramientas blancas y sus soportes.....	41
3.3.3 Vallado en la estación de trabajo	43
4. Estructura del programa desarrollado para simular el robot en un entorno de cirugía laparoscópica.....	44
4.1 Parámetros declarados previos a los módulos	44
4.1.1 Variables <i>pose</i> y numéricas.....	44
4.1.2 Variables de los objetos de trabajo y puntos de los elementos 3D de la estación.....	44

4.1.3 Array de <i>robtargets</i> aux.....	45
4.1.4 <i>Jointtargets</i>	46
4.1.5 <i>Tooldatas</i> auxiliares y <i>Pinz</i>	46
4.1.6 <i>Robtargets aux</i> y <i>robtarget</i> salir.....	47
4.1.7 Variables numéricas ‘A’ y ‘N’	47
4.1.8 Variable numérica ‘t’	48
4.2 Funciones usadas frecuentemente	49
4.3 Módulos de la aplicación: Definición y características	50
4.3.1 Módulos que servirán como interfaz de usuario	50
4.3.1.1 Módulo <i>Main_simu</i>	50
4.3.1.2 Módulo <i>Opciones</i>	52
4.3.1.3 Módulo <i>Aplicacion</i>	53
4.3.1.4 Módulo <i>PrimerMov</i>	53
4.3.1.5 Módulo <i>Dentro</i>	55
4.3.1.6 Módulo <i>Fuera</i>	56
4.3.2 Módulos encargados de los movimientos principales de las herramientas	58
4.3.2.1 Módulo <i>PerpendicularZona</i>	58
4.3.2.2 Módulo <i>PerpendicularMesa</i>	59
4.3.2.3 Módulo <i>DesplazaDentroAng</i>	60
4.3.2.4 Módulo <i>EntradaConAngulo</i>	63
4.3.3 Módulos encargados de permitir parámetros correctos.....	65
4.3.3.1 Módulo <i>Permitir_validos</i>	65
4.3.3.2 Módulo <i>Limite_profundidad</i>	65
4.3.4 Módulos utilizados para la opción demo	67
4.3.4.1 Módulo <i>Demo</i>	67
4.3.4.2 Módulos <i>DemoBlanca</i> y <i>DemoNegra</i>	68
4.3.4.3 Módulo <i>Circunferencia</i>	69
4.3.4.4 Módulo <i>Barrido</i>	70
4.3.5 Módulos encargados de que no choque la herramienta con el <i>Pelvitainer</i> ...	73
4.3.5.1 Módulo <i>Hacia_Orificio</i>	73
4.3.5.2 Módulo <i>Orificio_a_base</i>	73
4.3.5.3 Módulo <i>Salir_Pelvitainer</i>	73
4.3.6 Módulos para tomar y dejar la herramienta.....	75
4.3.6.1 Módulos <i>TomarHerraBlanca</i> y <i>TomarHerraNegra</i>	75
4.3.6.2 Módulos <i>DejarHerraBlanca</i> y <i>DejarHerraNegra</i>	75

4.3.6.3 Módulos <i>AbrirPinza</i> y <i>CerrarPinza</i>	75
5. Funcionamiento de la aplicación desarrollada	76
5.1 Funcionamiento del menú inicial	78
5.1.1 Demo.....	78
5.1.2 Herramienta negra o blanca.....	79
5.2 Funcionamiento del menú herramienta.....	80
5.2.1 Salir en el menú herramienta	80
5.2.2 Herramienta	80
5.2.3 Aplicación	81
5.3 Funcionamiento del menú orificio	83
5.3.1 Opción perpendicular a la mesa	84
5.3.2 Opción perpendicular al orificio	85
5.3.3 Opción de entrada al orificio con ángulo	90
5.3.4 Salir en el menú orificio	99
5.4 Funcionamiento del menú movimiento	101
5.4.1 Opción de desplazamiento por ángulos.....	102
5.4.2 Opción de cambiar la profundidad a la que está introducida la herramienta	108
5.4.3 Salir en el menú movimiento	116
5.5 Funcionamiento del menú <i>Demo</i>	118
5.5.1 Demostración con la herramienta blanca en el menú <i>Demo</i>	118
5.5.2 Demostración con la herramienta negra en el menú <i>Demo</i>	120
5.5.3 Opción “Barrido” en el menú <i>Demo</i>	121
5.5.4 Opción “Circunferencia” en el menú <i>Demo</i>	123
5.5.5 Opción “Salir” en el menú <i>Demo</i>	125
6. Límites	126
6.1 Límites de profundidad.....	126
6.2 Ángulos límite.....	128
6.2.1 Ángulos límite con entrada de la herramienta perpendicular al orificio	129
6.2.1.1 Ángulos límite con entrada de la herramienta negra perpendicular al orificio.....	129
6.2.1.2 Ángulos límite con entrada de la herramienta blanca perpendicular al orificio.....	132
6.2.2 Ángulos límite con entrada de la herramienta perpendicular a la mesa.....	132
6.2.2.1 Ángulos límite con entrada de la herramienta negra perpendicular a la mesa	133

6.2.2.2 Ángulos límite con entrada de la herramienta blanca perpendicular a la mesa	133
6.2.3 Ángulos límite de los orificios 4, 5 y 6.....	134
6.2.4 Entrada con ángulo.....	135
6.2.3.1 Herramienta negra.....	135
6.2.3.2 Herramienta blanca	136
7. Conclusiones y líneas futuras.....	138
8. Bibliografía.....	139
9. Índices de las ilustraciones y de las tablas.....	142
9.1 Índice de ilustraciones	142
9.2 Índice de tablas	147
10. Anexos	149
10.1 Planos	149

1. Introducción y objetivos

Este proyecto consiste en la planificación de movimientos de herramientas quirúrgicas en operaciones de cirugía laparoscópica con robots. En concreto, utilizaremos el brazo robótico *ABB IRB 120* para simular los movimientos del instrumental quirúrgico en una operación de cirugía laparoscópica, mediante la realización de una simulación, utilizando el software *RobotStudio*.



Ilustración 1 - Robot ABB IRB 120 (ABB, s.f.)

1.1 Objetivos del proyecto

El objetivo de este proyecto es realizar una simulación en la que el brazo robótico *ABB IRB 120* realice movimientos con herramientas quirúrgicas utilizadas en cirugía laparoscópica.

Para ello, se realizará el modelado en 3D de los elementos necesarios para poder simular este entorno quirúrgico. Entre estos elementos, se mostrará un especial interés en modelar el *Pelvitainer* a escala real, instrumento que se detallará más adelante.

Cuando se implementen estos modelos 3D en la estación con el brazo robótico, se buscará encontrar una programación lo más efectiva posible para simular los movimientos quirúrgicos que se realizan con el instrumental de cirugía laparoscópica (cámara laparoscópica, pinzas...). Todo este instrumental quirúrgico se caracteriza por su “gran longitud”, lo que unido a que debe “insertarse” en el abdomen humano por un único punto, denominado trocar, hace que el movimiento de este instrumental quirúrgico

sea complejo, ya que se trata de un movimiento de pivoteo alrededor del punto de fulcro.

Se desea realizar una estación de trabajo utilizando el software *RobotStudio*, en el que el usuario pueda simular los movimientos y trayectorias con robot, del instrumental quirúrgico utilizado en operaciones de cirugía laparoscópica.

Otro propósito para el desarrollo de este TFG es que sirva de base como material de docencia en robótica.

1.2 El instrumento *Pelvitainer*

En el ámbito médico, los cirujanos realizan entrenamiento de operaciones de cirugía laparoscópica utilizando el *Pelvitainer* (ver ilustración 2). Con este dispositivo físico se simula el abdomen del paciente, y a través de sus orificios (trocares), el cirujano entrena y perfecciona sus habilidades para el manejo del instrumental laparoscópico.

Para realizar nuestra simulación se ha realizado el modelado 3D del *Pelvitainer*, que simula el abdomen humano. Este elemento será de vital importancia en nuestro proyecto, porque será donde se realicen todos los movimientos de las herramientas laparoscópicas con el robot. Una unidad de este *Pelvitainer* se encuentra en el Instituto *ITAP*, en la Escuela de Ingenierías Industriales de Valladolid.



Ilustración 2 - Imagen del Pelvitainer tomado en el departamento de ITAP

Este *Pelvitainer* ha sido creado por *Karl Storz*. *Karl Storz* es una compañía de elaboración en la sección de la endoscopia, brinda sistemas para la medicina veterinaria y humana, del mismo modo que para usos manufactureros. (STORZ, s.f.).

La necesidad que tuvieron de crearla fue porque, para el mundo de la cirugía, hace falta mucha práctica, bastante estudio y experiencia. Las metodologías de laparoscopia presentan un aprieto enorme, por ello hace falta mucho entrenamiento en el que el *Pelvitainer* puede prestar una gran ayuda.

Otro de los motivos por el que surge la necesidad de utilizar el *Pelvitainer* es porque hay estudios que demuestran que aprender cirugía visualizándola no es suficiente y realizar operaciones reales sin experiencia previa es arriesgado. También, hay que tener en cuenta que cada operación tiene su dificultad, por lo que el entrenamiento es obligatorio. Otro argumento a tener en cuenta es que hay un progresivo empuje para bajar los costos y durabilidad de las operaciones, con lo cual urge más la preparación.

Vistas estas dificultades *Karl Storz*, provee este instrumento de formación, con la finalidad de crear un entorno de aprendizaje para que el usuario pueda entrenar y mejorar en la cirugía. (Storz, Karl, 2020)

El *Pelvitainer* consta de 9 orificios, que simularán las aperturas del paciente en el torso. En estos orificios se introducirán las herramientas quirúrgicas durante la operación.

El robot ABB accederá a estos orificios con unas herramientas modeladas en 3D, que se mostrarán más adelante. El final de la herramienta tendrá una forma cilíndrica con un diámetro menor al de los orificios del *Pelvitainer*. Una vez que el final de la herramienta entre en el orificio se realizarán movimientos orientados a la cirugía. El *Pelvitainer* se modelará en 3D a escala real.

La ejecución del proyecto se llevará a cabo en la siguiente estación de *RobotStudio* que detallaremos en futuros apartados:

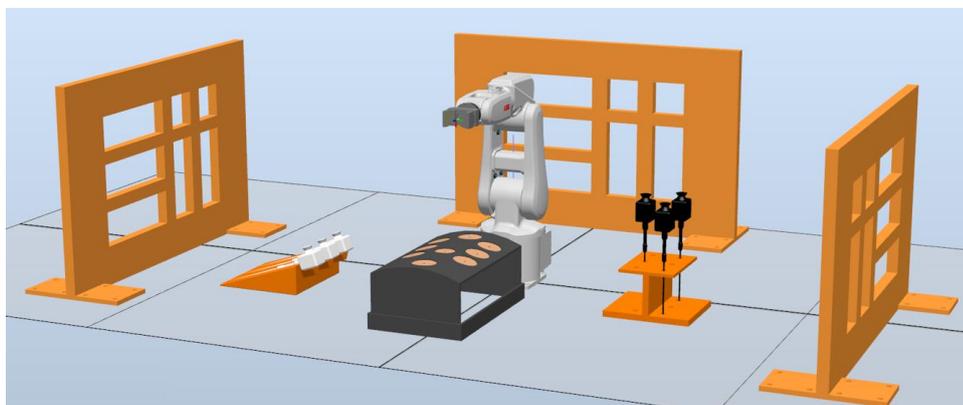


Ilustración 3 - Estación de trabajo de RobotStudio

1.3 Estructura de la memoria

En esta memoria del TFG abordamos en primer lugar una breve introducción a la cirugía laparoscópica. A continuación, en el capítulo 3, se aborda el modelado de los distintos elementos del escenario quirúrgico, y la creación y desarrollo de la estación de trabajo que simula el robot quirúrgico. En el tema 4 se aborda la estructura del programa del robot desarrollado, para simular movimientos y trayectorias del material quirúrgico movido por el robot, para simular una intervención en el abdomen humano, utilizando el *Pelvitainer*. En el capítulo 5 se indica el funcionamiento de la aplicación informática desarrollada. Algunas limitaciones de dicha aplicación se muestran en el capítulo 6. Finalmente, se indican las conclusiones y bibliografía.

2. Breve introducción a la robótica aplicada a la cirugía laparoscópica

2.1 La cirugía laparoscópica

La cirugía laparoscópica es una opción que es menos invasiva que la cirugía habitual, en la cual, se usa una cámara diminuta cuyo nombre es laparoscopio que sirve para poder visualizar el vientre adentro. (Clínica Universidad de Navarra, s.f.).

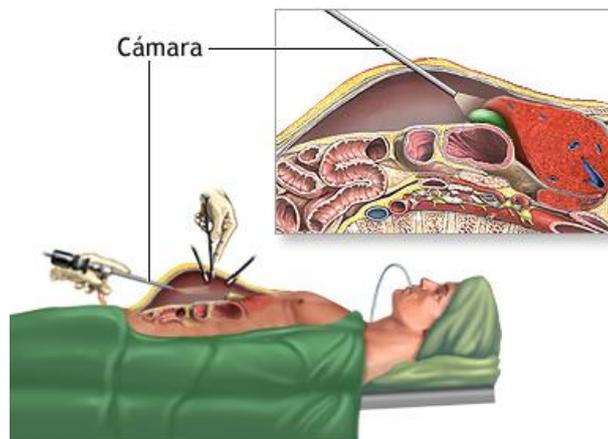


Ilustración 4 - Intervención quirúrgica utilizando cirugía laparoscópica (Zieve, 2021)

Un laparoscopio es un cilindro fino y dúctil que, en el final, tiene una luminaria y una diminuta cámara para poder visualizar el video. (American Cancer Society, 2019). Consta de un visor, que permite ver lo que capta la cámara del extremo de la herramienta.



Ilustración 5 - Imagen de un laparoscopio con el visor marcado (Olympus, s.f.)

Se puede utilizar el visor de diferentes formas, una de ellas es mediante la electrónica que, gracias a ella se puede visualizar el extremo de la herramienta en una pantalla y realizar funciones a partir de ella, como por ejemplo *zoomear*.

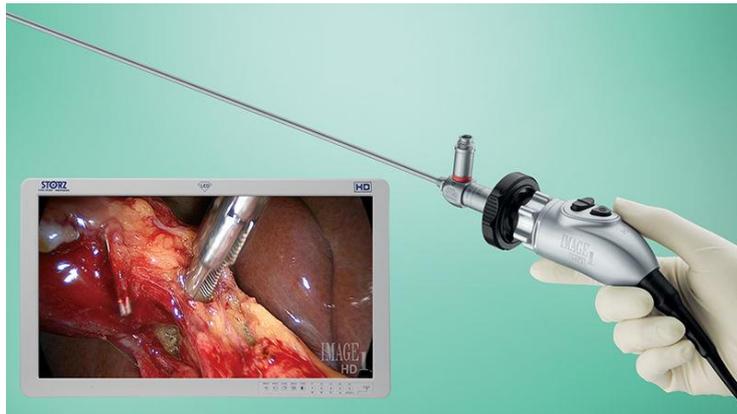


Ilustración 6 - Visión de lo que capta la cámara a través de una pantalla (Karl Storz, s.f.)

También se puede aproximar el ojo al visor para visualizar lo que capte la cámara:



Ilustración 7 -Ejemplo de profesional usando el visor (Reeve, 1998)

2.1.1 Historia de la cirugía laparoscópica

Según el artículo *'Una breve historia de la endoscopia, laparoscopia y cirugía laparoscópica'* de Spaner, la palabra *'laparoscopia'* procede del griego *'laparo'* que significa *'flancos, cada uno de los espacios laterales del abdomen que se encuentran entre las costillas falsas y los huesos superiores de la pelvis'* y *'scopia'* que es *'herramienta de vigilancia o atención'* (Etimologías de Chile, 2022). En el artículo dice que la palabra laparoscopia es el método por el que se inspecciona la lámina que envuelve el área interna del abdomen con un endoscopio.

Ciertos historiógrafos apropian a Albukasim, un quirúrgico árabe, (sobre el año 1000) la primera examinación de una cavidad interior. Recurrió al reflejo de la luminiscencia usando espejos para poder inspeccionar la parte baja del útero.

Aproximadamente, 7 siglos más adelante hubo progresos notorios en esta habilidad, el encargado de este progreso fue Philipp Bozzini. Este visibilizó, con buenos resultados, la uretra, utilizando una herramienta formada por una cámara de luminiscencia duplicada. En esta herramienta el origen de la luminiscencia era una candela que se refractaba en un espejo y se podía ver las vísceras a partir de diferentes cavidades de la persona.

Desormeaux, sobre 1863, fue el primero en crear un tubo de endoscopio con buenos resultados y se empleó para explorar la vesícula y la uretra.

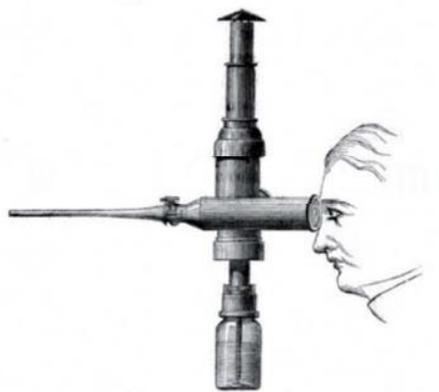


Ilustración 8 - El endoscopio de Desormeaux (Charrière, s.f.) (Antonio García Ruiz, 2016)

Maximilian Nitze, en el año 1877, transformó los endoscopios previos y les otorgó lentes, y, sobre todo, un canal quirúrgico en el que era posible meter herramientas para dilataciones uretrales o extracciones. Gracias a Thomas Edison con el invento de la bombilla, en 1880, se adecuan pequeñas

bombillas al final de la herramienta. Esto genera una solución al problema de originar luminiscencia en estas intervenciones. Aunque, había un nuevo inconveniente, el calor que desprendían estas bombillas.

En 1901, George Kelling efectuó la primera laparoscopia para poder examinar las conclusiones del neumoperitoneo en los órganos, insertando un cistoscopio en el abdomen de un perro. Hans Christian Jacobaeus empleó el método de Kelling en personas e informó 45 investigaciones en el abdomen, por una técnica a la que llamó 'laparoscopia'. Esta persona proporcionó la observación de la cirrosis hepática, metástasis cancerosa y peritonitis tuberculosa entre otras.

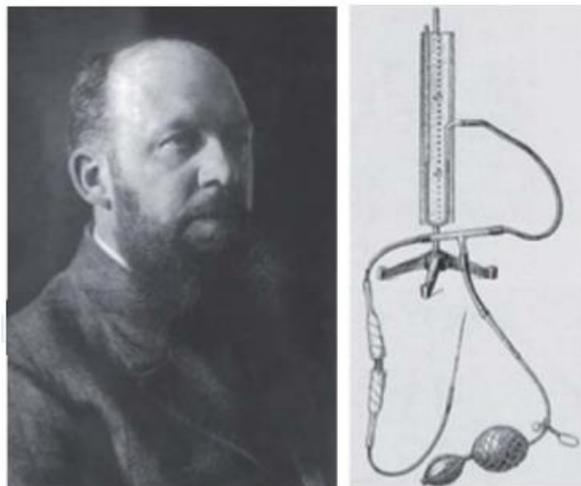


Ilustración 9 - George Kelling y su laparoscopio diseñado (Antonio García Ruiz, 2016)

El primer uso de laparoscopia en EE. UU se dio en el año 1911 con Bernheim. Éste empleó una bombilla y un proctoscopio introducido en el abdomen para poder visualizar el vientre, la vejiga y el hepático. Sobre 1918, Otto Gotz, crea un método para meter aire en las oquedades humanas. En años siguientes, Zollikofer, entre 1920 y 1930, fue el primero en utilizar el dióxido de carbono para la ejecución de este método. Janos Veress en 1938 mejoró el invento de Otto y dio lugar a la conocida 'aguja de Veress' traída para la producción del neumoperitoneo.



Ilustración 10 - Aguja de Veress (Tagumedica, s.f.)

Kurt Semm, que era ginecólogo de Ucrania, en 1966 era conocido por hacer buenos métodos operatorios, también diseñó varias herramientas, como por ejemplo el insuflador electrónico de neumoperitoneo, con la finalidad de realizar cirugía laparoscópica. Por ello, se le concibe como ‘padre de la laparoscopia’ y en ese entonces la laparoscopia entra en el mundo de la ginecología.

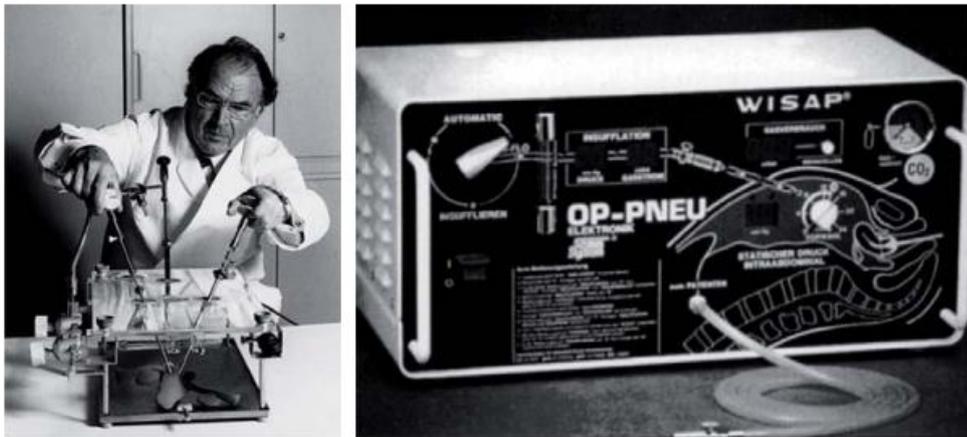


Ilustración 11 - Kurt Semm y su invento, el insuflador electrónico de neumoperitoneo (Antonio García Ruiz, 2016)

Schindler y Wolf, de Alemania, sobre 1930, crearon los endoscopios flexibles. Este endoscopio era semirrígido y se podía visualizar el objetivo gracias a las muchas lentes que tenía.

Otro notorio progreso fue en 1986, el video entró en el mundo de la endoscopia. El hecho de usar vídeos supuso un gran avance, dado que las intervenciones lograban ser más efectivas al poderse realizar movimientos más acertados, además permitía mejorar la docencia de este ámbito.

El factor de juntar una videocámara al laparoscopio marcó un antes y después en el mundo quirúrgico. Gracias a ello, en Francia, en 1987, Philippe

Mouret llevó a cabo la primera colecistectomía usando la cámara de video en el laparoscopio. Mühe, en el 86, fue el que lo hizo primero, pero sin la ayuda del vídeo, utilizando un endoscopio que modificó él y le dio como nombre 'galloscopio'.



Ilustración 12 - Imagen de Philippe Mouret (Wikipedia, 2020)



Ilustración 13 - Eric Mühe y su "galloscopio" (Reynolds, 2001) (Antonio García Ruiz, 2016)

En 1987, Mouret expuso un trabajo acerca de esta colecistectomía laparoscópica. Dos años después en 1989, lo llevan a cabo Dubois, Reddick y Olsen. Reddick y Olsen dieron a conocer las buenísimas resoluciones de las 200 colecistectomías laparoscópicas que llevaron a cabo. Estos magníficos resultados, despertaron la curiosidad de todos los quirurgos por esta técnica de cirugía.

El apogeo, que obtiene la cirugía laparoscópica en la ginecología desde la década de los 70 y en los quirurgos convencionales en los años 80, consigue volver a atraer la curiosidad de los urólogos. Los urólogos pospusieron el progreso de la cirugía laparoscópica dado que no tenían un órgano de acceso simple. Los primeros experimentos de laparoscopia en la urología empiezan con acercarse al riñón, porque era el órgano que tenía mejor cavidad para acceder. Con lo cual, en 1969, Bartel llevó a cabo una retroperitoneoscopia, sin embargo, no disponía de la suficiente área de trabajo del que se requería en la cavidad. Wickham, 10 años después, aplica dióxido de carbono a esta cavidad para ensancharla y lleva a cabo una ureterólisis, pero no causó curiosidad ni fue aceptado.

En los finales de los años 80, los avances tecnológicos realizados en la grabación y visualización del vídeo, a la vez que la transferencia de imágenes, logró que la cirugía laparoscópica fuera más conocida y que ésta evolucionara más.

La primera colecistectomía laparoscópica en Norteamérica, se dió gracias a Mckernan en 1988. Casi a la vez, como hemos mencionado antes, Reddick y Olsen realizan las primeras intervenciones quirúrgicas de la vesícula mediante el uso de laparoscopia.



Ilustración 14 - Barry McKernan a la izquierda y Eddie Joe Reddick a la derecha (Antonio García Ruiz, 2016)

Desde ese entonces y a lo largo del tiempo, muchos ámbitos quirúrgicos que tienen en común la realización de la cirugía en el abdomen han ido acogiendo la laparoscopia por las numerosas ventajas que ofrecen y la evolución de la tecnología.

En el año 2000, la robótica se empieza a adaptar a la cirugía y esto aportó mejoras a algunos de los inconvenientes que tenía la cirugía laparoscópica. Mejorando así, la cirugía laparoscópica al aplicarle robótica alrededor del mundo. (Pérez Marínez, 2014), (Guía del cirujano, s.f.) y (Antonio García Ruiz, 2016)

2.1.2 Ventajas de la cirugía laparoscópica

La cirugía laparoscópica ofrece numerosas ventajas: (Calvo, 2019)

- Las incisiones que se realizan son de mucho menor tamaño que la cirugía convencional.
- El paciente experimentará una dolencia más pequeña.
- Tras finalizar la cirugía laparoscópica, las cicatrices son mucho más pequeñas y dejan una mejor imagen a diferencia de otras cirugías.
- Las incisiones, al ser más pequeñas, dan lugar a que haya menos peligro de sufrir un percance.
- Con el uso del laparoscopio, el profesional podrá tener una mejor visualización de las zonas sobre las que se va a trabajar.

La importancia de la cirugía laparoscópica es que, como hemos mencionado este ámbito es menos lesivo que otro tipo de operaciones quirúrgicas y además, al ser más preciso, es menos dañino. Esto ayuda a que el paciente vuelva antes a la vida normal y se recupere más rápido.

Estos datos urgen a conocer más este tipo de cirugía y a intentar expandirla lo máximo posible.

2.2 Robots aplicados a la cirugía laparoscópica

A continuación, se muestran algunos de los robots aplicados a la cirugía laparoscópica:

2.2.1 Robot da Vinci

El Robot da Vinci es de las últimas progresiones de las cirugías levemente invasivas. Cuenta con una visión 3D con buenísima calidad, también dispone de herramientas articuladas *Endowrist*. Además, posee un sistema de control fácil de utilizar que da lugar a que el quirurgo pueda realizar operaciones de dificultad alta con una cirugía mínimamente invasiva

Se originó en Silicon Valley, a partir de patentes militares y elaborado por la compañía proveniente de *California Intuitive Surgical Inc.* y, en el año 1999, salió al mercado. Gracias a este robot, la cirugía robótica progresó bastante a nivel mundial, dando resultados que van mejorando continuamente con la evolución de las consecutivas plataformas robóticas.

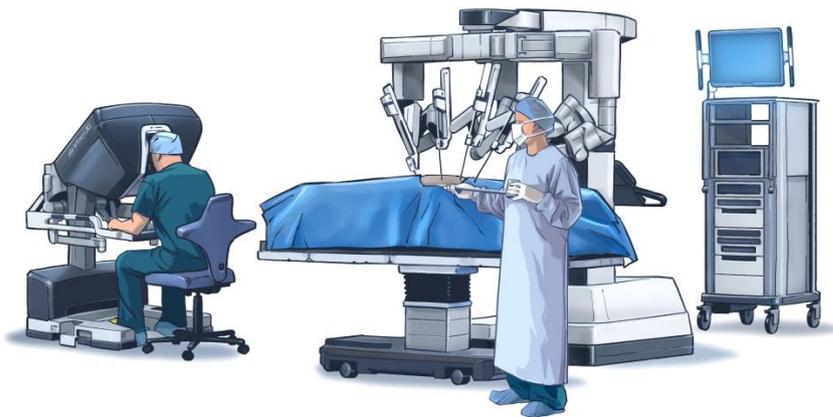


Ilustración 15 – Sistema robótico da Vinci (Urología aplicada, s.f.)

La cirugía robotizada da Vinci resulta ser una combinación de las virtudes de la cirugía laparoscopia y la abierta. El quirurgo tiene una visibilidad extensa y con buena resolución. Igualmente, puede operar en partes del cuerpo con un acceso complejo y todo por medio de aperturas de escasos centímetros.

El hecho de que sea preciso, de mínima invasión y con poco riesgo se convierte en un beneficio para el profesional y para la persona a operar.

Gracias a este robot el profesional tiene una técnica menos arriesgada para disección anatómica y la reconstrucción. Esto tiene muchas ventajas para el paciente, el cual sufre una operación con menos daños, con una hendidura

pequeña, un sangrado reducido y una mejoría, después de la operación, más rápida, lo que da lugar a una vida común más veloz. (Abex, s.f.)

2.2.2 Hugo RAS de Medtronic

Es una incorporación reciente al mundo de la robótica aplicada a la cirugía laparoscópica. Normalmente, muchos hospitales no se podían permitir implementar robótica quirúrgica en sus salas de operación por el alto coste que tenía. Medtronic, aporta a una solución a este problema con el sistema Hugo RAS.

Es una “plataforma modular de múltiples cuadrantes diseñada para una amplia gama de procedimientos de tejidos blandos”. Junta una visión 3D, “*Touch Surgery Enterprise*” y herramientas utilizadas en la medicina. Es un resultado de administración y grabación de videos de cirugía apoyado en la nube. Cuenta con *hardware* y *software* de apoyo, afines a este sistema, que se especializan en optimar, dar servicio y tener aptitudes de programas de robótica. También, el programa permite realizar rotaciones de 360 grados, cualidad que resulta bastante útil. (García, 2022)



Ilustración 16 - Hugo RAS de Medtronic (Hm Hospitales, 2021)

Se utilizó por primera vez en urología y ginecología en India y Latinoamérica (Hm Hospitales, 2021). Este sistema fue utilizado por primera vez en España por la *Fundació Puigvert* en dos operaciones. En concreto, se realizó una prostatectomía radical y una nefrectomía parcial el 14 de febrero con unas buenas conclusiones. Además, la *Fundació Puigvert* ha sido el segundo centro médico en utilizarlo. Un trabajador de este centro dice que, gracias a este

sistema, los tumores difíciles de tratar se pueden operar de forma parcial sin necesidad de extirpar el órgano. A mayores, debido a que los resultados han sido tan buenos, los profesionales no tienen ningún problema en repetir el uso de este sistema. (Pérez, 2022)

El sistema Hugo RAS no se utiliza en todo el mundo. Esto dependerá del consentimiento, autorización o accesibilidad en el mercado que determinen las condiciones reglamentarias de los países. Dentro de la Unión Europea, el sistema Hugo Ras posee la marca CE. En Canadá, Hugo RAS tiene el certificado de dispositivo médico. También, Hugo RAS está aceptado en Australia. Hugo RAS en EE. UU no se encuentra en el mercado (García, 2022).

2.3 Beneficios de aplicar la robótica a la cirugía laparoscópica

El mundo de la medicina es imprescindible hoy en día y ha ayudado mucho al mundo durante la historia. Este proyecto tiene la finalidad de brindar la mayor ayuda posible a esta rama. En el Grado en Ingeniería Electrónica Industrial y Automática, se obtienen varios conocimientos sobre la rama de la ingeniería, una de ellas es la robótica, que son los conocimientos que aplicaremos a este proyecto. Con lo cual, como se ha mencionado anteriormente, aplicaremos la robótica en la cirugía laparoscópica, que es parte de la rama de la medicina.

Como hemos comentado, la cirugía laparoscópica y la robótica industrial son dos campos que ofrecen muchas virtudes hoy en día. Por ello, combinar estos campos puede dar lugar a buenos resultados.

En la cirugía laparoscópica la precisión es un factor muy importante a tener en cuenta, por eso, la robótica en este campo puede ser muy útil. Los brazos robóticos poseen una gran precisión y con ellos se pueden hacer movimientos muy milimétricos. Como ya hemos comentado previamente, el brazo robótico que utilizaremos es el *ABB IRB 120*. La interfaz de usuario creada en el proyecto permite que se pueda sacar provecho a estas cualidades del brazo robótico.

Otro beneficio que aporta el uso del brazo robótico en la cirugía laparoscópica es que puede hacer falta menos profesionales. Es decir, el brazo robótico podría sustituir a un profesional sanitario en una operación. Por ejemplo, si en urgencias hiciera falta, repentinamente, una operación de un paciente recién traído por la ambulancia, al ser la situación tan imprevista, podría faltar personal porque no estaba planificada. Esa falta de personal podría suplirlo el brazo robótico, éste podría actuar como el brazo de un profesional en caso de que el personal necesite utilizar más herramientas.

A la vez el personal sanitario, está más abierto al uso de las nuevas tecnologías y siente curiosidad sobre este mundo porque, se sabe que, cuanto más se disponga de innovación tecnológica, mejores resultados pueden adquirir. También, la cirugía laparoscópica, de por sí, es un resultado de la evolución que ha habido en la tecnología estos últimos tiempos. (Carlos Gutiérrez-Banda, 2016).

Esta cirugía que resulta poco invasiva ha supuesto una revolución en el campo de la cirugía. Asimismo, el uso de estas tecnologías se ha implantado en otro tipo de cirugías tales como, cirugía torácica, cardiovascular, uróloga...

Al ser útil la tecnología en tantos campos de la cirugía, los profesionales se sienten más motivados al implantarla. Es un hecho que los robots irán evolucionando más y más a medida que pase el tiempo y, por lo tanto, acabarán siendo más utilizados. Al mismo tiempo, las salas donde se realizan estas operaciones irán cambiando para que se pueda hacer un mejor uso de la tecnología y se la pueda aprovechar más, dando así mejores resultados. La cirugía laparoscópica presenta limitaciones y dificultades, la robótica para esta cirugía se ha creado para superar estos obstáculos y así resultar menos peligrosa y más factible. (Ramos, 2006).

2.4 Estudios sobre la cirugía laparoscópica y robótica aplicada a esta

Se han realizado estudios sobre las complicaciones que puede tener la cirugía laparoscópica. Alberto Pérez-Lanzac junto a otros profesionales hicieron una investigación que resultó en que el 14% de cirugías laparoscópicas renales en menores de 16 años tuvieron dificultades. Se llevaron a cabo otras investigaciones con conclusiones similares como la de Drăghici y otros profesionales en Rumania, sobre el acontecimiento de la peritonitis postoperatoria en personas que se realizó la cirugía laparoscópica. Dio como resultado que el 0.56% de las operaciones tenían esta incidencia y que, en el 0.96% de los casos, hacía falta volver a intervenir.

Estas conclusiones de los estudios enseñan que, aunque la cirugía laparoscópica presenta inconvenientes, tiene resultados más ventajosos que los que se dan en la cirugía abierta.

Por otro lado, Alberto Muñoz-Calero, presidente de la AEC (Asociación Española de Cirujanos) ha declarado que 8 de cada 10 operaciones en España se realizan utilizando la cirugía laparoscópica (Infosalus, 2010). Lo cual, incita a pensar que está resultando efectiva la cirugía laparoscópica dado que es una opción a la que se recurre bastante.

Tenemos también los estudios de Brinder y Rassweiler acompañado de sus respectivos profesionales. Estos defienden los beneficios del uso de la robótica en la cirugía, probado en el artículo de Buzad y otros profesionales, realizado en el 'Instituto de Cirugía Robótica de Texas' sobre la ejecución de colecistectomía vía cirugía robótica analizando rendimiento y coste. La conclusión fue: (Pérez Marínez, 2014)

- Disminución de dificultades durante la operación.
- No había necesidad de realizar transfusiones de sangre.
- Uso menor de la anestesia.
- Reducción de tiempo en el hospital.
- Recuperación rápida, que da lugar a que las personas vuelvan antes a la vida normal.

3. Desarrollo del entorno de simulación

3.1 Robot ABB IRB 120

El brazo robótico *ABB IRB 120* será el robot encargado de realizar los movimientos de cirugía laparoscópica con las herramientas, éste se ha mostrado en la Ilustración 1. Este brazo robótico es una opción de confianza y beneficiosa para crear un elevado potencial productivo, a su vez, con un contenido coste. (ABB, s.f.)

El robot *ABB IRB 120* es un producto rápido, liviano y posee una buena exactitud a la hora de realizar trayectorias o acceder a puntos para el uso que se requiera. Sus motores tienen un incremento de velocidad bueno para que no falle en su ejecución (Eurobots, s.f.). A continuación, se muestran algunas características de él:

Características del ABB 120	
Tipo	Articulado
Número de ejes	6 ejes
Funciones	<i>Pick and place</i>
Carga máxima	3 kg
Radio de acción	580 milímetros
Repetibilidad	0,01 milímetros
Peso	25 kg

Tabla 1 - Características del brazo robótico ABB IRB 120 (Direct Industry, s.f.)

Su diseño admite que se pueda montar en una gran variedad de zonas con diferentes ángulos, podría ser en una línea de montaje al lado de otros robots o incluso sobre ciertos tipos de maquinaria.

Además, tiene un uso de potenciales aplicaciones tales como, empaquetado, instalación, manejo de materiales, atención de máquinas, medicina... (Shen, s.f.)

En la ilustración 17, mostrada a continuación, podemos ver a la izquierda, cuál es cada articulación del robot y su sentido de giro tanto positivo como negativo. A la derecha, tenemos las dimensiones del *ABB IRB 120* expresadas en milímetros.

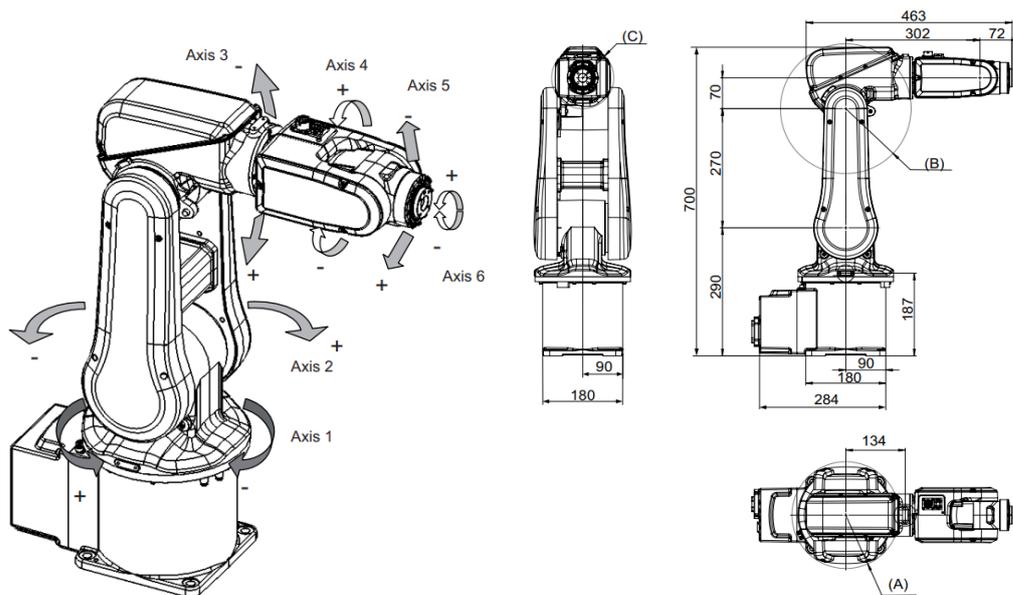


Ilustración 17 - Ejes y dimensiones del ABB IRB 120 (ABB Robotics, 2021)

A la derecha de la Ilustración 17, vemos que tenemos los parámetros A, B y C que indican un radio, su significado es el siguiente:

- A: Radio mínimo de giro del eje 1, $R=121$ mm
- B: Radio mínimo de giro del eje 3, $R=147$ mm
- C: Radio mínimo de giro del eje 4, $R=70$ mm

En la siguiente tabla podemos observar la velocidad de movimiento y el rango de trabajo de cada articulación:

	Velocidad de movimiento ($^{\circ}$ / s)	Rango de trabajo del eje ($^{\circ}$)
Articulación 1	250	-165 a 165
Articulación 2	250	-110 a 110
Articulación 3	250	-110 a 70
Articulación 4	320	-160 a 160
Articulación 5	320	-120 a 120
Articulación 6	420	-400 a 400

Tabla 2 - Velocidad y rango de trabajo de cada eje del IRB 120 (Eurobots, s.f.)

Estas dimensiones y parámetros que hemos visto recientemente sobre el robot darán lugar a que no pueda alcanzar cualquier posición que se le comande. Cuando no pueda llegar el robot a ciertos puntos diremos que estará ante una singularidad. En otras palabras, el robot alcanzará una singularidad cuando las seis articulaciones no logren alcanzar una

configuración determinada. Entonces, el rango de trabajo permitido del robot es el siguiente:

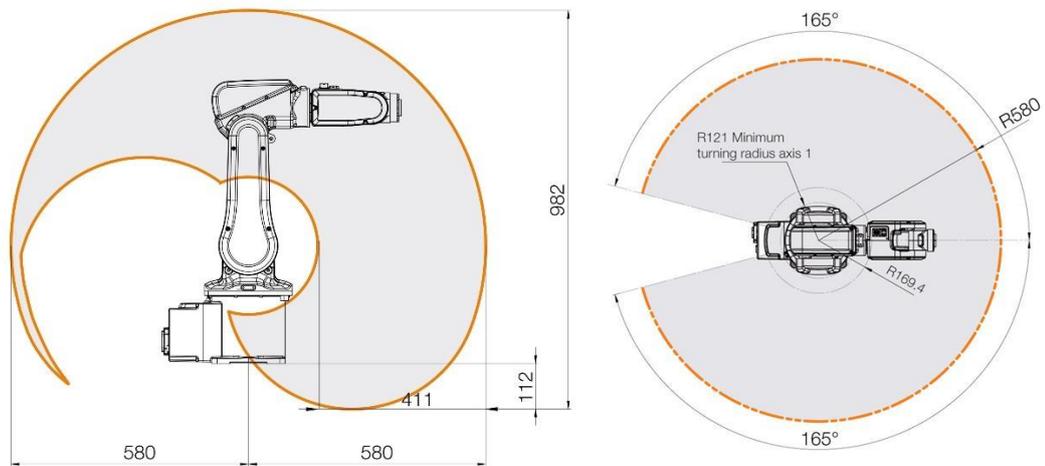


Ilustración 18 - Área de trabajo del IRB 120 (ABB Robotics, 2021)

Estas medidas también están expresadas en milímetros. Podemos ver que, dentro del radio de 580 milímetros, que se muestra a la derecha de la Ilustración 18, el robot dará un buen comportamiento con no muchas singularidades.

También, al robot se le puede combinar con *software* para que sea compaginable con otras *apps*, por ejemplo, de conectividades, de soldar o casos más complejos como el multiproceso entre otros. (ABB Robotics, 2021)

3.2 Modelado del *Pelvitainer* y los demás elementos de la estación

El *Pelvitainer* y otros elementos que se utilizarán en la estación se modelarán en 3D gracias a la aplicación de *Autodesk Inventor Professional 2022*.



Ilustración 19 - Logo de la aplicación de Autodesk Invento (Autodesk, s.f.)

Autodesk Inventor Professional brinda opciones capacitadas para realizar diseños mecánicos en 3D, y también su acreditación y su ejecución. Tiene un buen rendimiento con una profesional mezcla de opciones de diseño regulado, parametrizado, inmediato y que se pueden utilizar como se desee. (Autodesk, s.f.).

3.2.1 Modelado en 3D del *Pelvitainer*

A partir del *Pelvitainer* de la Ilustración 2 se han tomado las medidas de la forma más precisa posible y se ha modelado en 3D buscando que sea lo más fiel a la realidad:

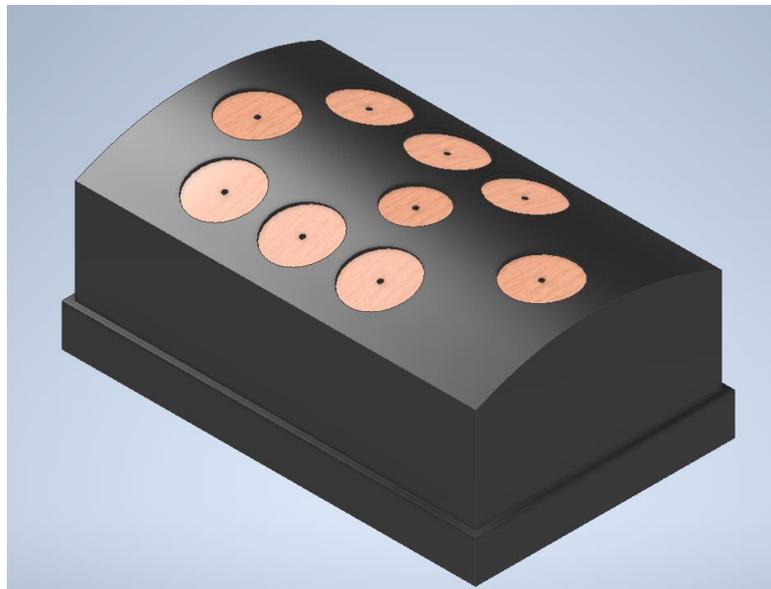


Ilustración 20 - Pelvitainer modelado en 3D en Autodesk Inventor

Las dimensiones del *Pelvitainer* en milímetros son las siguientes:

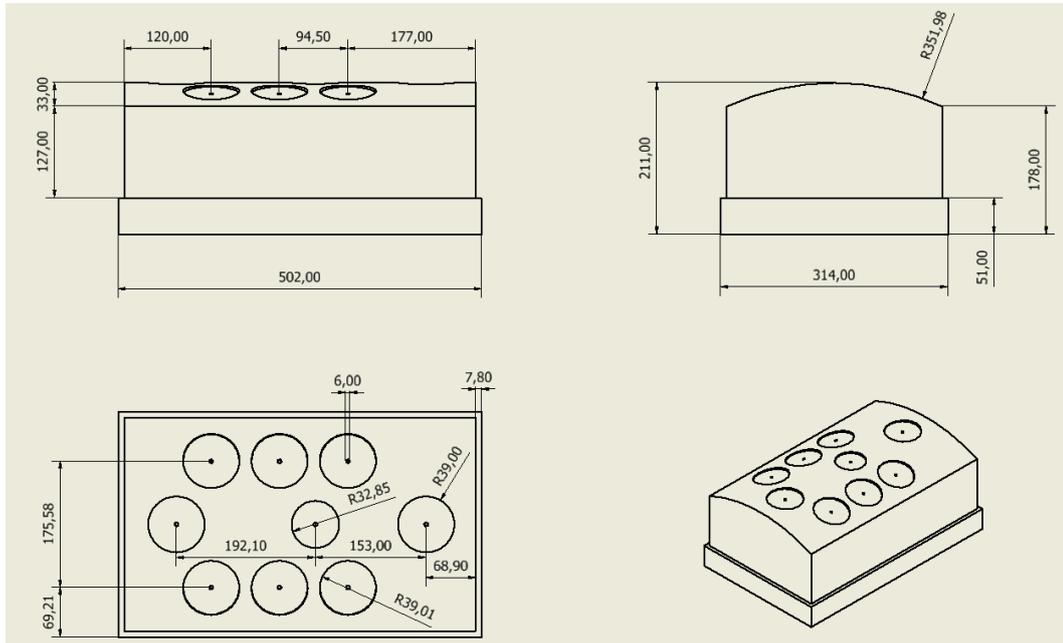


Ilustración 21 - Dimensiones del Pelvitainer

Cuando practiquemos movimientos de cirugía laparoscópica dentro del *Pelvitainer*, será necesario ver, dentro de él, si los movimientos realizados son efectivos con la herramienta que utilizemos. Así que, para poder visualizar mejor las herramientas en el *Pelvitainer*, omitiremos las paredes laterales quedando así de la siguiente forma:

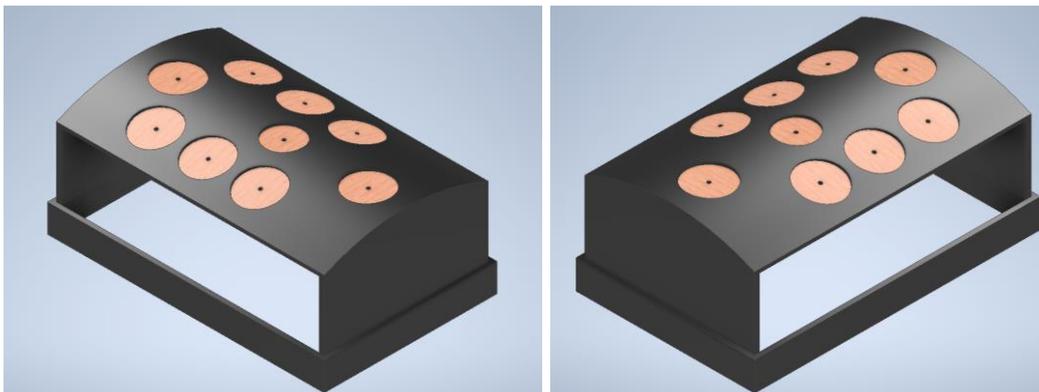


Ilustración 22 - Pelvitainer sin las paredes laterales

3.2.2 Modelado en 3D de las herramientas quirúrgicas

Como se ha dicho anteriormente, para operar en el *Pelvitainer* se utilizarán herramientas. Modelaremos dos tipos de herramientas, en una se utilizará un agarre lateral y en otra un agarre superior. Para distinguir los dos tipos de herramienta, dotaremos a cada una de un color diferente. La herramienta de agarre lateral será de color negro y la herramienta de color blanco será de agarre superior.

En la Ilustración 23 se muestran las dos herramientas de cirugía laparoscópica diseñada. Estas se acoplarán al robot, y serán utilizadas para simular los movimientos característicos de las herramientas laparoscópicas que se realizan durante las operaciones de cirugía.

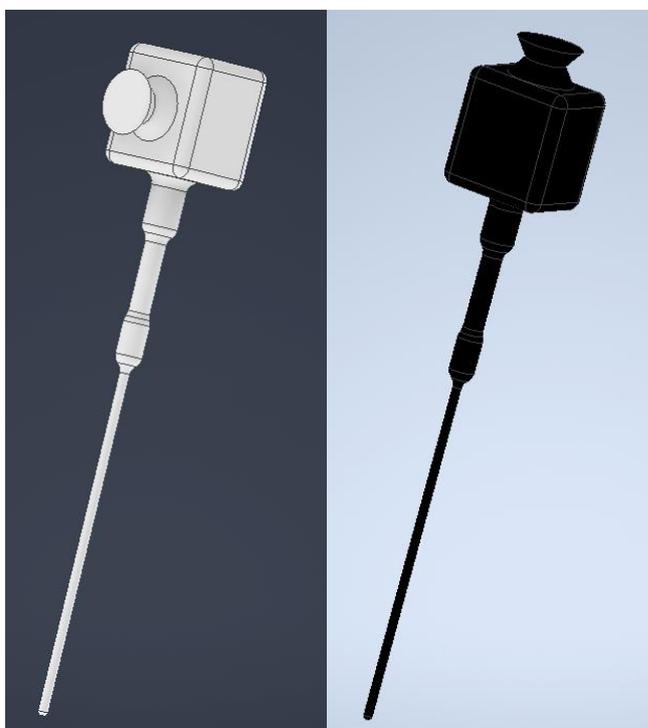


Ilustración 23 - Herramienta blanca y herramienta negra que se utilizará en la aplicación

Al tener estos dos tipos de herramientas modeladas, podremos implementar el número que queramos de cada una de ellas en el entorno de simulación. Las dimensiones de cada una de estas herramientas en milímetros son las siguientes:

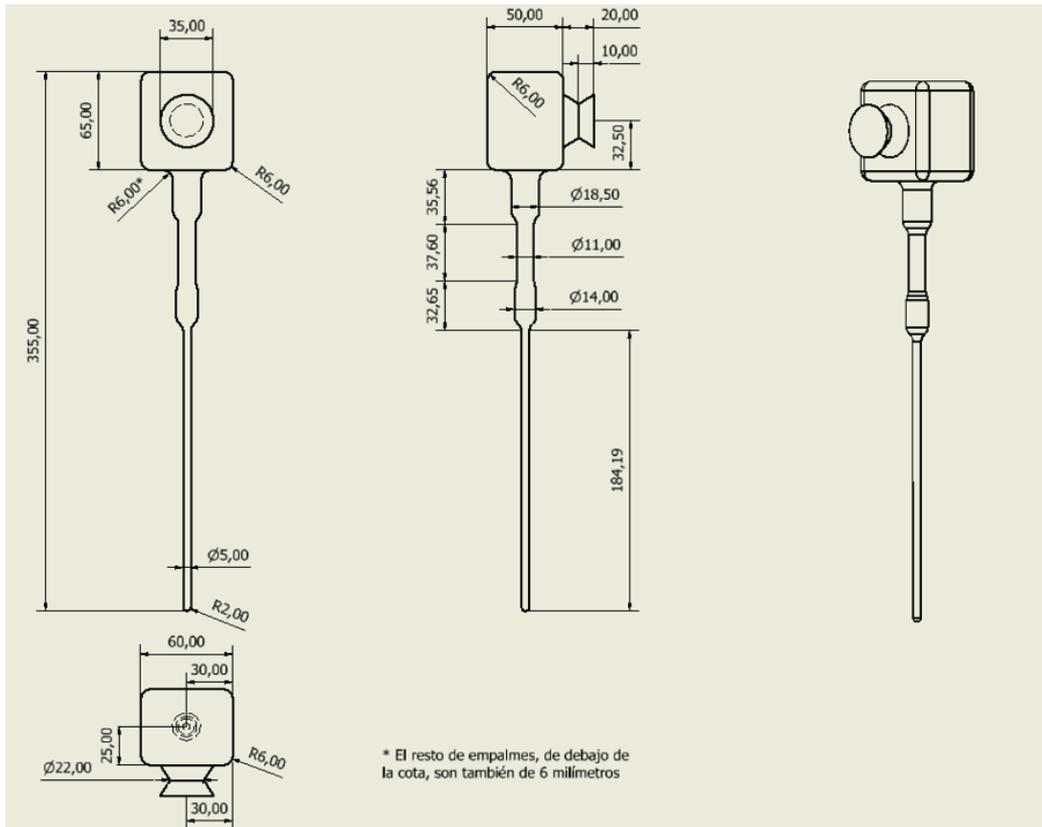


Ilustración 24 - Dimensiones de la herramienta blanca

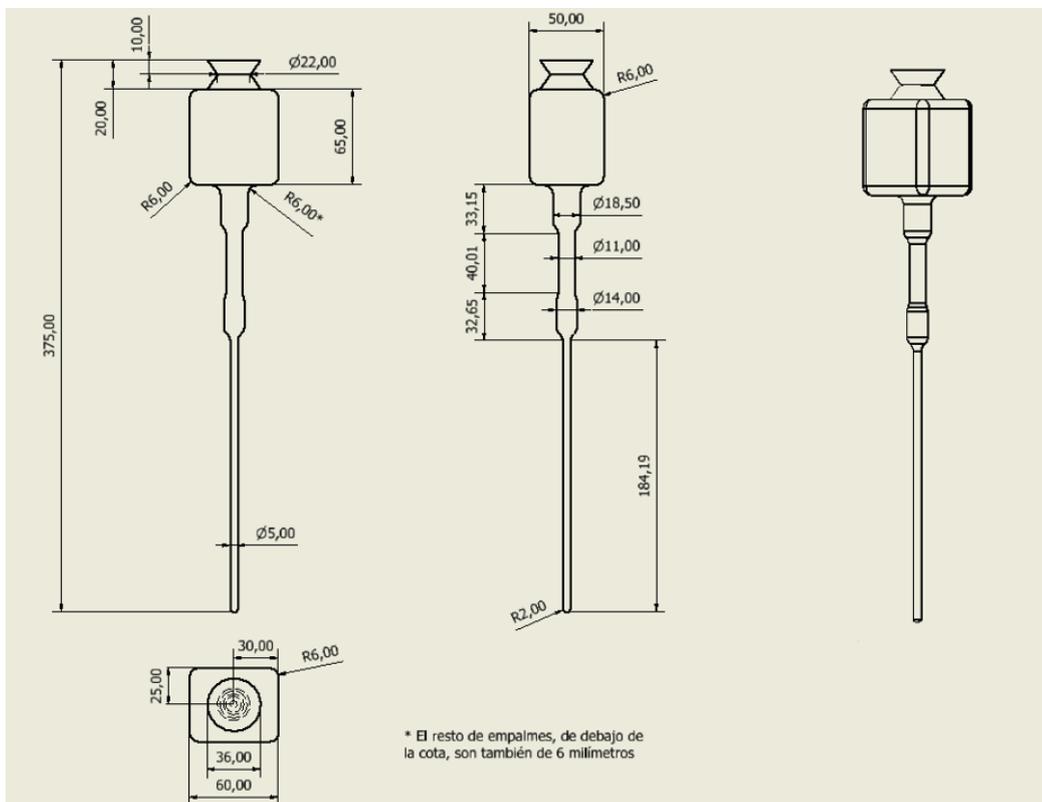


Ilustración 25 - Dimensiones de la herramienta negra

Observamos que en ambas herramientas tenemos dos conos truncados unidos por su base menor. Estos dos prismas juntados, imitan el visor del laparoscopio:

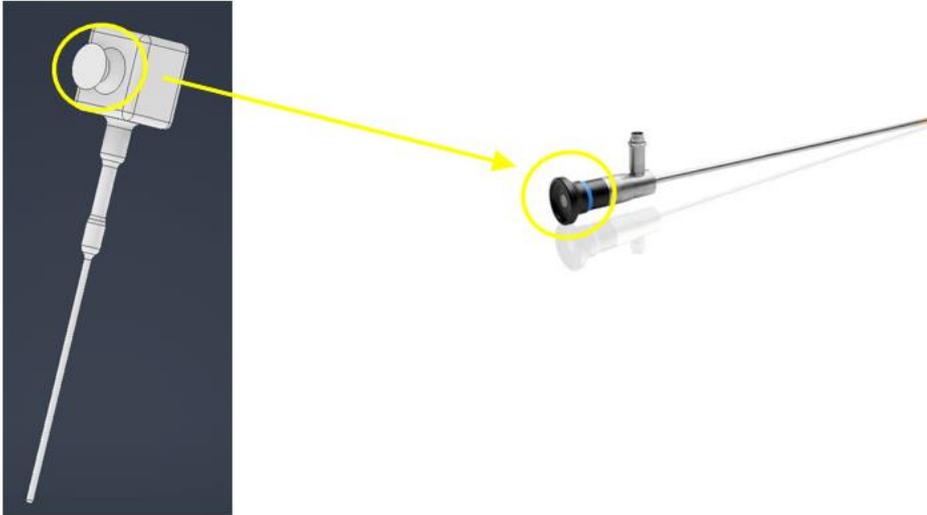


Ilustración 26 – Visor del laparoscopio emulado en la herramienta 3D

3.2.3 Modelado en 3D de los soportes de las herramientas

En el entorno de simulación, estas herramientas necesitarán estar sujetas en algún apoyo para que el robot lo pueda coger fácilmente. Con lo cual, se modelarán dos apoyos diferentes, uno dedicado a las herramientas negras y otro a las blancas.

Comenzamos, creando el apoyo dedicado a las herramientas negras. Se ha decidido modelarlo de la siguiente forma:

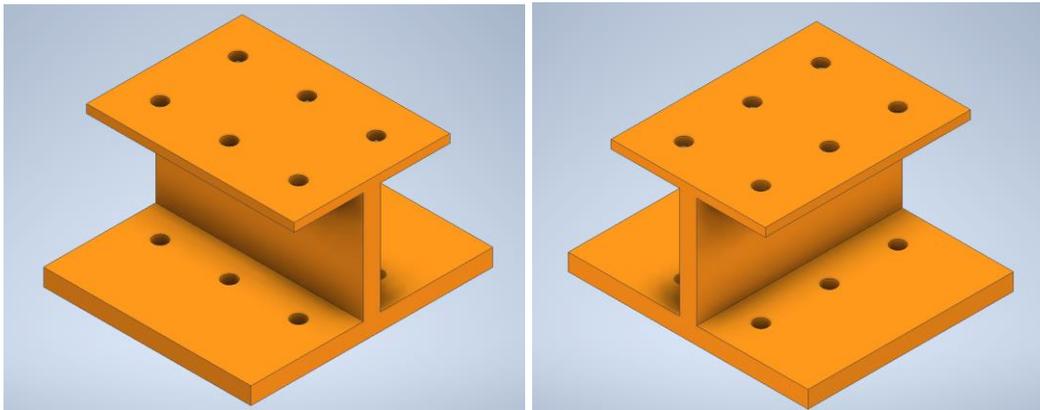


Ilustración 27 - Apoyo para las herramientas negras

La herramienta para colocarse en el soporte atravesará dos orificios, de forma que parezca que la herramienta este de pie. Este apoyo admite el máximo de seis herramientas.

Con este apoyo las herramientas podrán colocarse perpendicularmente al plano, sobre el que se apoyan los objetos de la estación de trabajo, y le será más fácil al robot coger la herramienta mediante un agarre lateral.

Las dimensiones, dadas en milímetros, de este apoyo, son las siguientes:

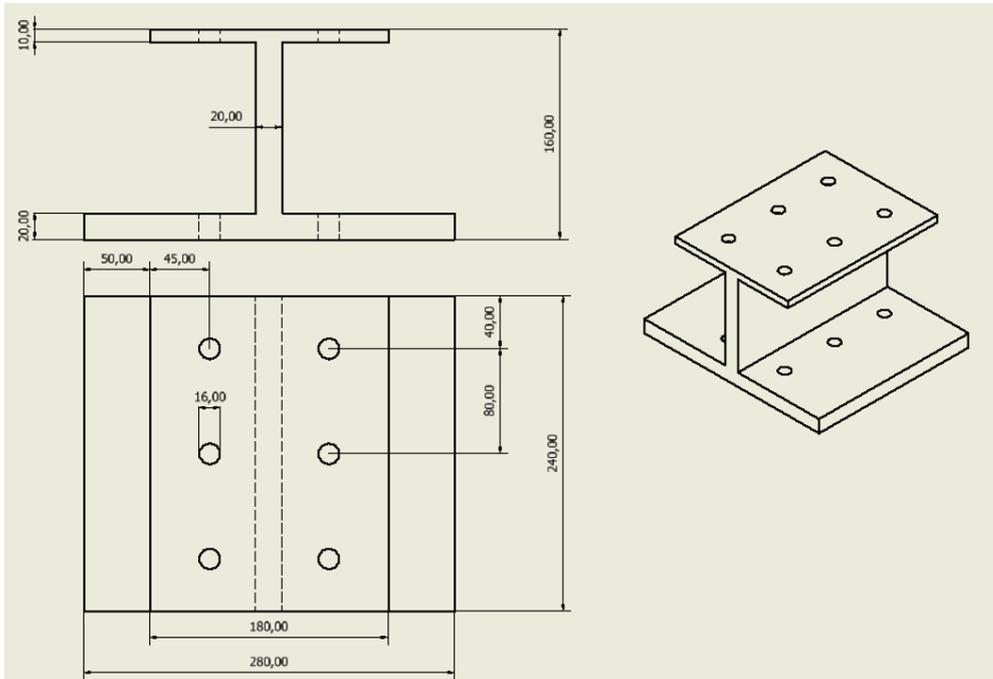


Ilustración 28 - Dimensiones del apoyo para las herramientas negras

Al igual que con las herramientas negras, se creará un apoyo que se adecue lo más posible a las herramientas blancas. El apoyo para las herramientas blancas será el siguiente:

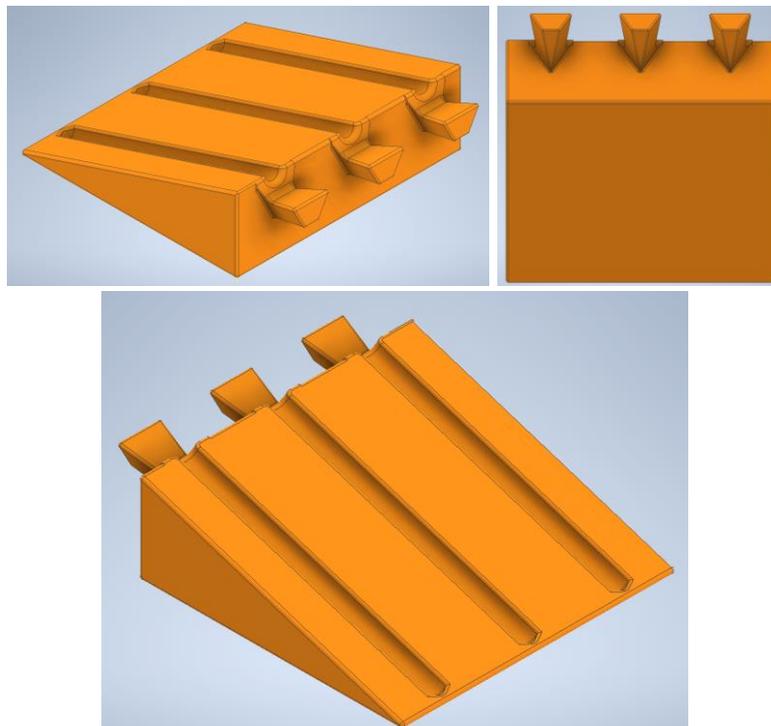


Ilustración 29 - Apoyo utilizado para las herramientas blancas

La “cabeza” de la herramienta blanca se apoyará en la zona que sirve de base, quedando el resto de la herramienta en la cavidad. Este apoyo permite depositar un máximo de tres herramientas.

En el apoyo, la parte donde reposa la cabeza de la herramienta, será la más cercana al brazo robótico. De esta forma, el robot podrá coger la herramienta blanca utilizando el agarre superior fácilmente. Además, este apoyo no presenta el inconveniente de que haya que tener cuidado con que la herramienta no entre en contacto con este. Porque la herramienta está tumbada y no se encuentra encajada.

Las dimensiones de este apoyo en milímetros son las siguientes:

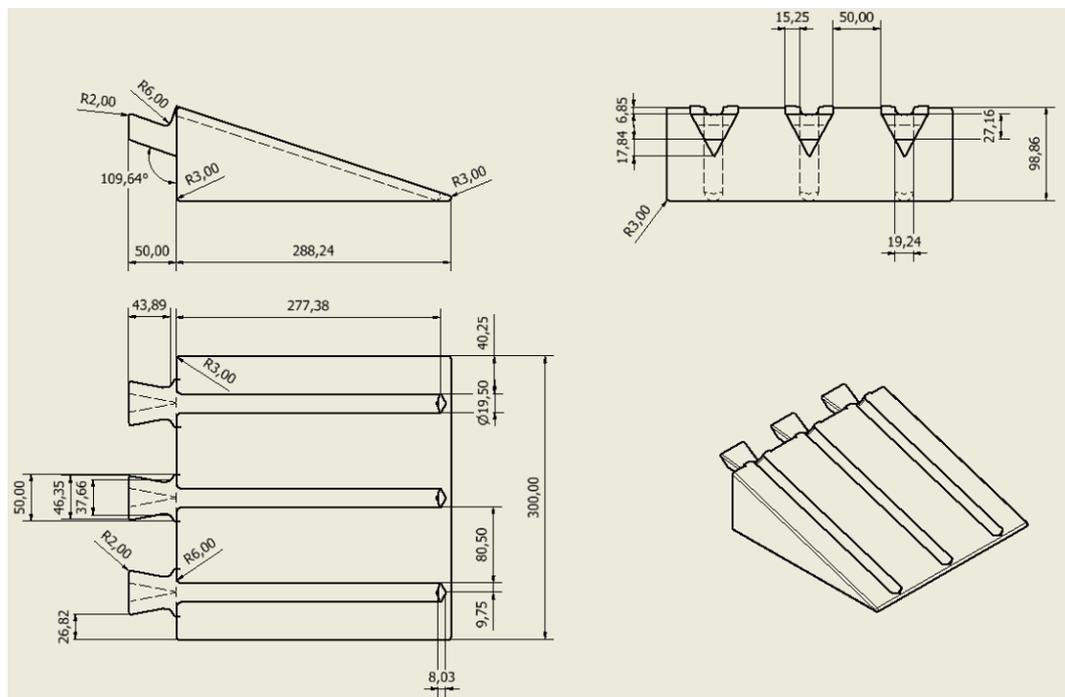


Ilustración 30 - Dimensiones para el apoyo de herramientas blancas

En resumen, para la estación en la que trabajará el robot, contaremos con el robot ABB IRB 120, el Pelvitainer y las herramientas con sus respectivos soportes.

3.2.4 Modelado en 3D del vallado

A mayores, se ha decidido añadir un vallado para mejorar el aspecto del entorno de simulación. El vallado, modelado en 3D también, tendrá la siguiente forma:

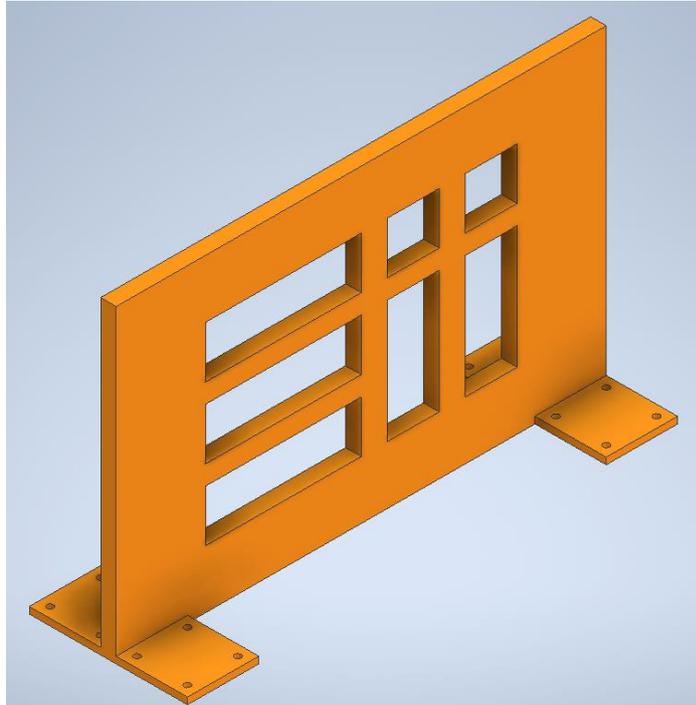


Ilustración 31 - Modelado en 3D del vallado

Podemos observar que el vallado tiene las letras “EII” del logo de la Escuela de Ingenierías Industriales de Valladolid:

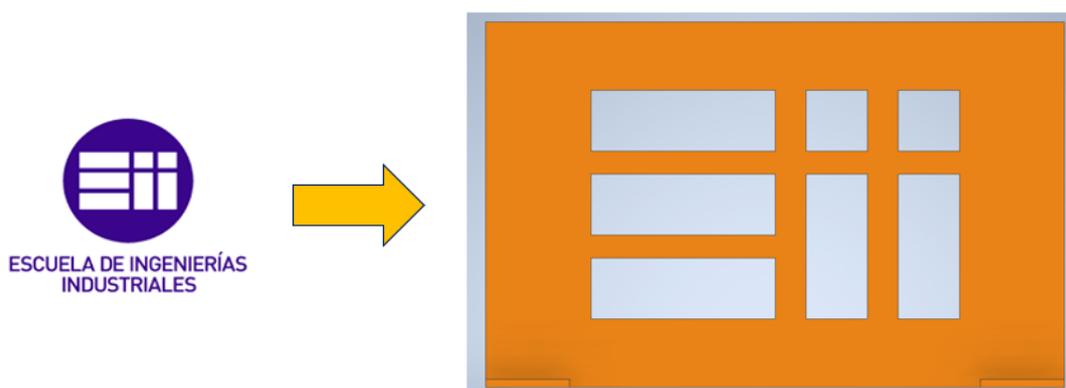


Ilustración 32 - Siglas de Escuela de Ingenierías Industriales aplicada al modelo 3D (Universidad de Valladolid, s.f.)

Al igual que con las herramientas, podríamos implementar este objeto las veces que queramos en la estación de la simulación. Este vallado cuenta con las siguientes dimensiones en milímetros:

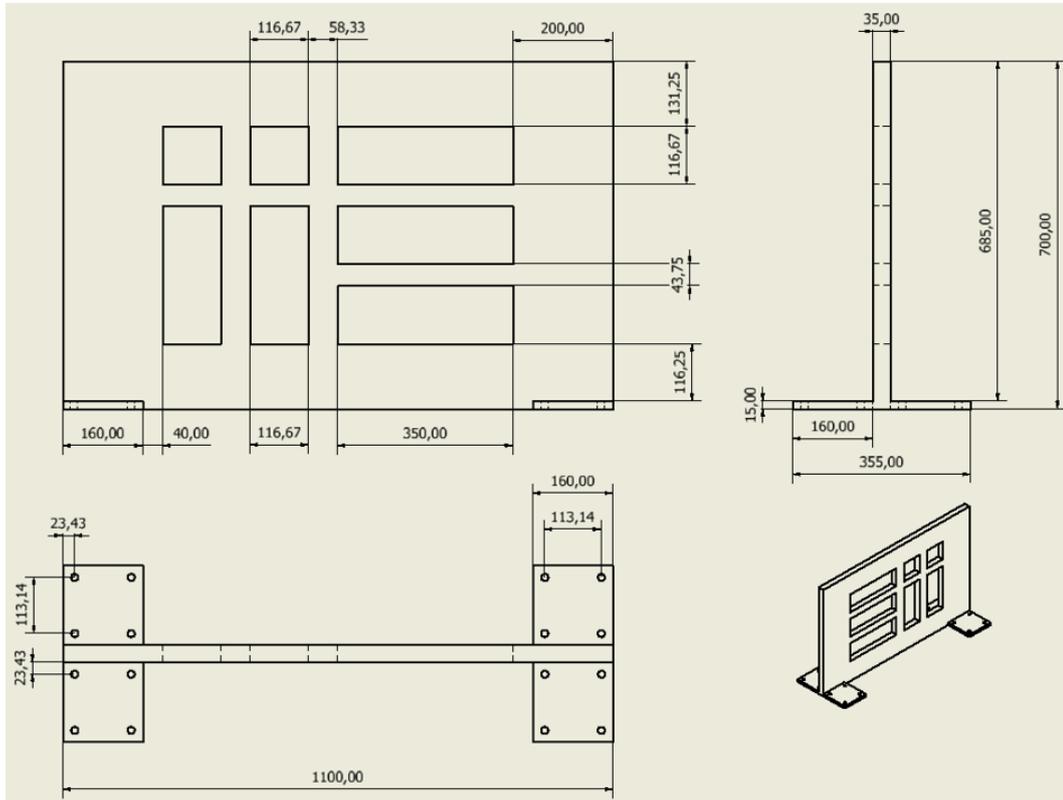


Ilustración 33 - Dimensiones del vallado

Para ver con más detalle las ilustraciones de las dimensiones de los modelos 3D, al final de la memoria se encuentra los planos de cada uno de estos.

Finalizado el modelado 3D de los diferentes objetos que se han mostrado, se implementarán en una estación de trabajo de *RobotStudio*. Se les exportará de la aplicación *Autodesk Inventor* como archivos *.stl* exceptuando el *Pelvitainer*, que se exportará como *.sat* para que sea lo más fiel al existente en la vida real.

3.3 Creación de la estación de trabajo para simular el robot en un entorno de cirugía laparoscópica

Utilizaremos la aplicación de *RobotStudio* para crear la estación de trabajo en la que simularemos los movimientos del robot *ABB IRB 120*.

RobotStudio es un software de *ABB* que sirve para simular y programar fuera de línea. Admite que el robot se pueda programar en un ordenador en el trabajo sin que se detenga la fabricación. Se fundamenta en *ABB VirtualController*, que es una imitación precisa del programa que se usa en la vida real para la ejecución de sus robots en rendimiento (*ABB*, s.f.). El lenguaje de programación que se utilizará en esta aplicación es *RAPID*.



Ilustración 34 - Logo de la aplicación de *RobotStudio* (*Microsoft*, 2016)

Combinando los objetos 3D que se han mencionado en el apartado anterior, tenemos como resultado la estación, que ya vimos en la Ilustración 3, con el siguiente aspecto:

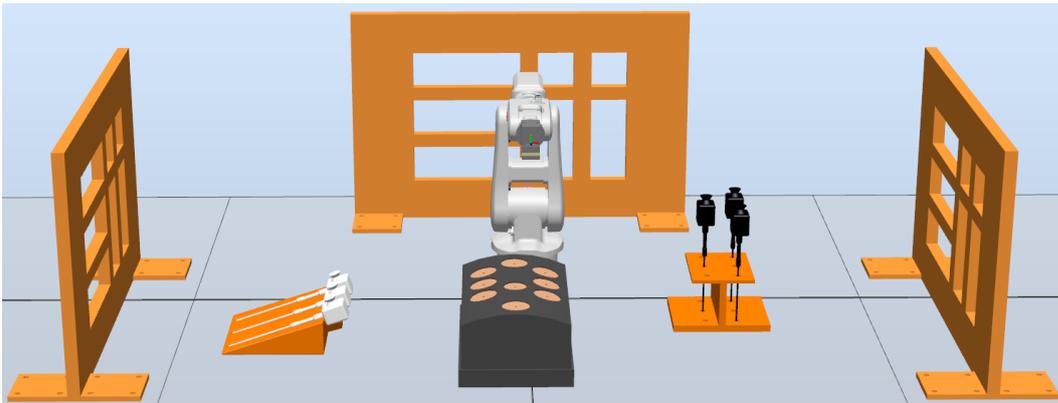


Ilustración 35- Estación resultante al implementar los objetos modelados en 3D

Se puede observar que, en frente del brazo robótico *ABB IRB 120*, se ha posicionado el *Pelvitainer*. A cada lado del robot se ha colocado un apoyo diferente, uno para las herramientas negras y otro para las blancas. Se ha decidido posicionar tres herramientas en cada apoyo. Alrededor, de estos objetos mencionados, se han colocado tres vallados formando parte un perímetro rectangular.

A su vez, *RobotStudio* nos puede mostrar un listado de los diferentes componentes de la estación. Para nuestro caso, tenemos el siguiente listado:

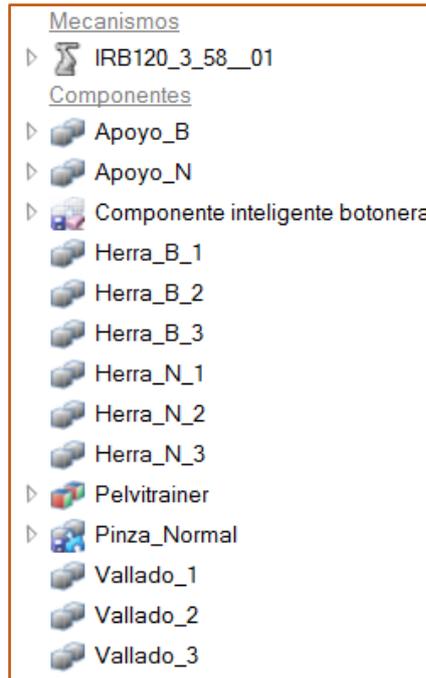


Ilustración 36 - Mecanismos y componentes en la estación

El robot IRB 120, la botonera y la componente de la pinza que forma parte del robot ya existían en la aplicación. Los demás componentes, como se puede deducir con su nombre, son los elementos ya comentados.

El *Pelvitainer* y los grupos de herramientas de color negro y blanco con sus respectivos soportes contarán con un *wobjdata* cada uno y los *robtargets* que sean necesarios.

Un *wobjdata* describe el objeto de trabajo sobre el que se realizará las tareas que se requieran y podrá tener uno o más *robtargets*. Un *robtarget* se utiliza para definir un *POSE*: posición (X, Y, Z) y orientación (cuaternio: q_1, q_2, q_3, q_4), y será una posición a las que pueda llegar el robot y se pueda realizar movimientos en torno a este.

RobotStudio nos puede mostrar también el listado de los objetos de trabajo de la estación con sus respectivos puntos:

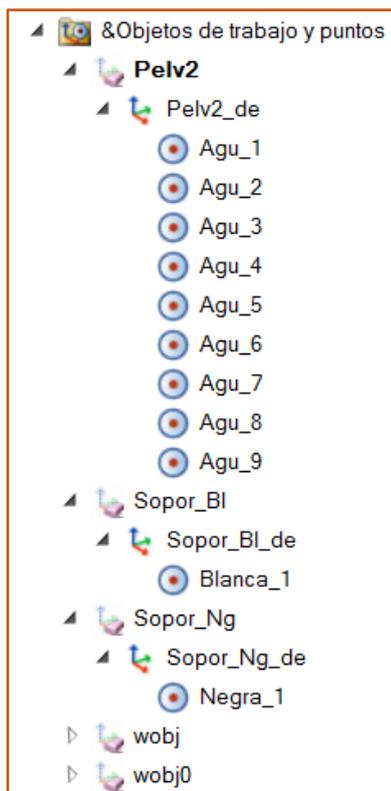


Ilustración 37 - Objetos de trabajo y puntos de la estación

Observamos que están los objetos de trabajo y puntos del *Pelvitainer*, también los objetos de trabajo de los soportes de cada grupo de herramientas con un punto para cada soporte. Sobre estos, entraremos en detalle más adelante. También vemos los objetos de trabajo de *wobj* y *wobj0* que ya existían en la aplicación.

Antes de continuar la explicación de este apartado, hace falta aclarar lo que significan los colores de los ejes de los *robtargets* y *wobjdatas*:

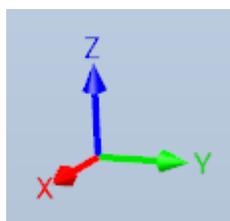


Ilustración 38 - Color para cada eje

En un *robtarget* o *wobjdata*, el eje X será de color rojo, el eje Y de color verde y el eje Z de color azul. Dicho patrón será así siempre.

3.3.1 Pelvitrainer en la estación de trabajo

Como ya hemos visto, frente al robot, tenemos el *Pelvitrainer*. Se han realizado pruebas con el *Pelvitrainer* ubicándolo en distintas posiciones alrededor del robot. La que mejor resultado ha proporcionado, desde el punto de vista de lograr la mejor accesibilidad del robot a las distintas posiciones del *Pelvitrainer* (orificios), es la que se muestra en la Ilustración 35.

Esta posición está caracterizada por el sistema de referencia “Pelv2” que se muestra señalado en la Ilustración 39. Las coordenadas POSE de este sistema de referencia, respecto al sistema de referencia *Wobj0* del robot se indican en la siguiente tabla:

Coordenadas (POSE) del sistema de referencia “Pelv2” asociado al Pelvitrainer						
Posición			Orientación (cuaternio)			
x	y	z	q1	q2	q3	q4
673,276	-40,7703	210,226	-6.17E-07	-0.00020604	0.999995	0.00299397

Tabla 3 - Coordenadas del sistema de referencia “Pelv2” asociado al Pelvitrainer

El POSE (posición + orientación) mostrado en la tabla anterior, corresponde al sistema de referencia del *Pelvitrainer* que se muestra en la Ilustración 39.

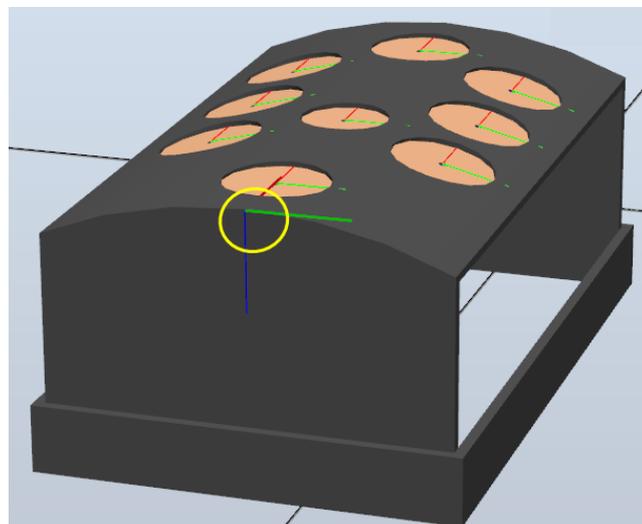


Ilustración 39 - Ubicación del sistema de referencia “Pelv2” asociado al Pelvitrainer

Se ha creado un *robtarger* para cada posición-orificio-agujero del *Pelvitrainer*. En la Ilustración 40 se puede ver el sistema de referencia asociado a cada orificio del *Pelvitrainer* y, a su vez, la enumeración que seguirán estos orificios.

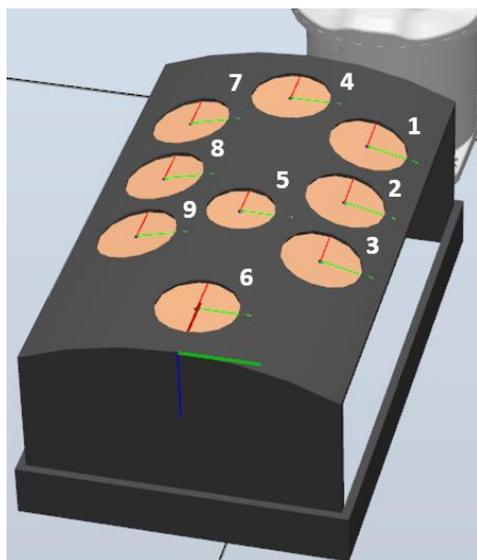


Ilustración 40 - Enumeración de los orificios del Pelvitainer

En el sistema de referencia asociado a cada orificio del *Pelvitainer*, el plano XY ha sido creado paralelo a cada orificio. El eje Z, de cada robtarget del *Pelvitainer*, ha sido creado hacia dentro del *Pelvitainer* y perpendicular a su respectivo plano XY. Los POSE de cada orificio respecto al sistema de referencia *Pelv2* del robot son los siguientes:

POSE de los orificios del <i>Pelvitainer</i> , referido al sistema de referencia "Pelv2"							
Orificio	Posición			Orientación (cuaternio)			
	X (mm)	Y (mm)	Z (mm)	q ₁	q ₂	q ₃	q ₄
1	366,05	86,09	14,983	0,9915	0,13	0,0009	-0,007
2	271,55	86,09	14,983	0,9915	0,13	0,0009	-0,007
3	177,045	86,09	14,983	0,9915	0,13	0,0009	-0,007
4	414,01	-1,64	3,22	1	0,003	0	0,0002
5	221,95	-1,64	3,22	1	0,003	0	0,0002
6	68,895	-1,688	3,22	1	0	0	0
7	366,045	-89,49	13,96	0,992	-0,124	0,00002	0,00017
8	271,545	-89,49	13,96	0,992	-0,124	0,00002	0,00017
9	177,045	-89,49	13,96	0,992	-0,124	0,00002	0,00017

Tabla 4 - Posición y orientación de los orificios del Pelvitainer

3.3.2 Herramientas quirúrgicas y su almacén en la estación de trabajo

Como se ha mencionado antes, a cada lado del robot tendremos un soporte de herramientas con sus respectivas herramientas.

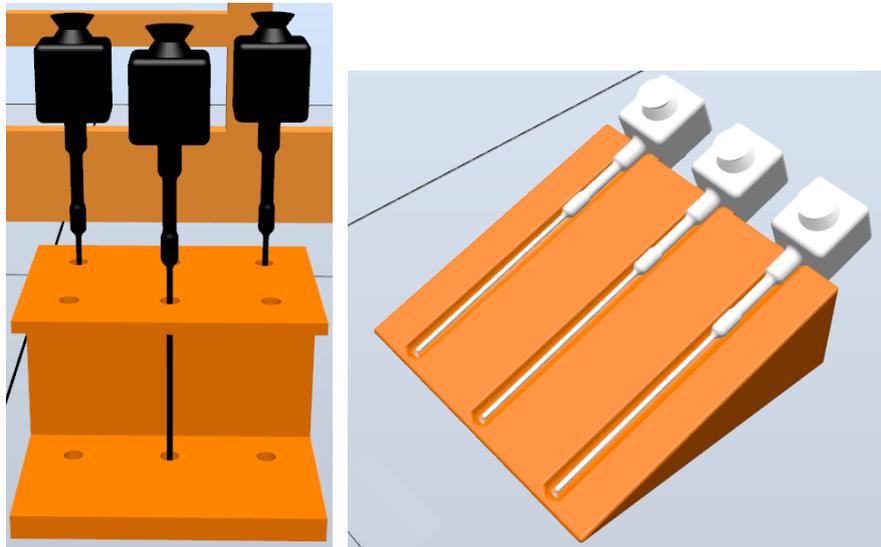


Ilustración 41 – Herramientas utilizadas en la simulación

Igual que el *Pelvitainer*, se han barajado distintas posiciones y las localizaciones resultantes han dado buenos resultados en cuanto a coger y dejar la herramienta sin llegar a ninguna singularidad.

3.3.2.1 Herramientas negras y sus soportes

La posición del soporte de las herramientas negras está caracterizada por el sistema de referencia “*Sopor_Ng*”. Las coordenadas POSE de este sistema de referencia, respecto al sistema de referencia *Wobj0* del robot se indican en la siguiente tabla:

Coordenadas del S.R. “ <i>Sopor_Ng</i> ” asociado al soporte de herramientas negras						
Posición			Orientación (cuaternio)			
x	y	z	q1	q2	q3	q4
277,805	417,658	20,203	0,498614	-0,501385	0,500663	0,4993

Tabla 5 - Coordenadas del S.R. “*Sopor_Ng*” asociado al soporte de herramientas negras

El POSE (posición + orientación) mostrado en la tabla anterior, corresponde al sistema de referencia del soporte de herramientas negras que se muestra redondeado en amarillo en la Ilustración 42.

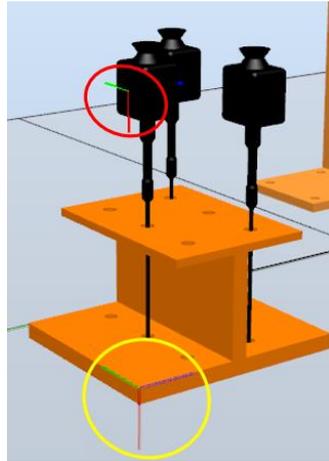


Ilustración 42 - Wobjdata del soporte de las herramientas negras y robtarget de la herramienta negra

Se ha creado un *robtarget* para este objeto de trabajo. En la Ilustración 42 está señalado con una circunferencia de color rojo. El punto está situado en la cara lateral de la herramienta negra. En todas las simulaciones solo se utilizará esa herramienta, no se usarán las otras dos. El *POSE* del *robtarget* de la herramienta respecto al sistema de referencia *Sopor_Ng* es el siguiente:

POSE de la herramienta negra, referido al sistema de referencia "Sopor_Ng"						
Posición			Orientación (cuaternio)			
x	y	z	q1	q2	q3	q4
-307,069	120,606	68,997	1	0	0	0

Tabla 6 - Posición y orientación del punto de la herramienta negra

La orientación está cuidada para que cuando el robot acceda a ese punto, obtenga un agarre lateral fácilmente. Cuando coja esa herramienta, el robot quedará con la herramienta agarrada de la siguiente forma:

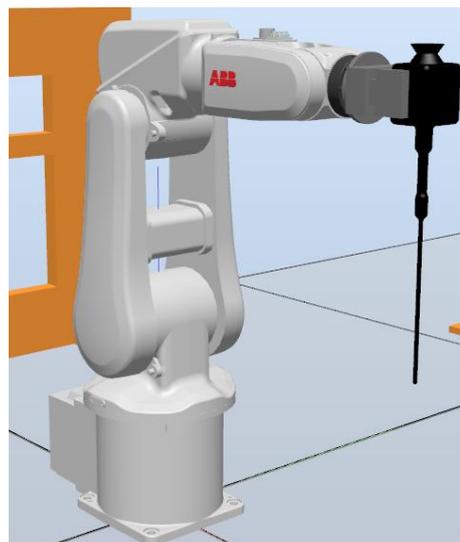


Ilustración 43 - Agarre lateral con la herramienta negra

Necesitaremos implementar un *tooldata* para esta herramienta negra para poder acceder con el final de la herramienta a los *robtargets* de los orificios. Un *tooldata* define la posición y orientación del TCP de la herramienta que vaya a tener el brazo robótico, También define las características físicas de la carga de la herramienta.

Para obtener una posición y una orientación precisa del final de esta herramienta, definiremos este parámetro en la estación. Haremos uso de la opción 'Crear dato de herramienta':

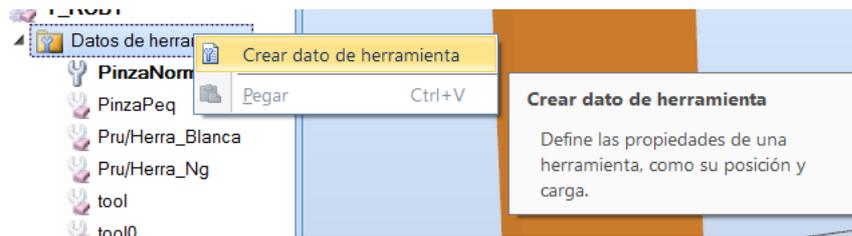


Ilustración 44- Opción de crear un tooldata en la estación de RobotStudio

En la que nos preguntará por los siguientes parámetros:

Crear dato de herramienta	
Otros datos	
Nombre	Herra_Ng
Herramienta sostenida por el rc	True
Sistema de coordenadas de herramienta	
> Posición X, Y, Z	Values...
> Rotación rx, ry, rz	Values...
Datos de carga	
Peso	1
> Centro de gravedad x, y, z	Values...
> Inercia	...
Propiedades de sincronización	
Tipo de almacenamiento	TASK PERS
Tarea	T_ROB1 (irb120_profibus_original)
Nombre de módulo	

Ilustración 45 - Parámetros a preguntar en la opción 'Crear dato de herramienta'

En la estación indicaremos la posición más precisa y centrada del final de la herramienta e introduciremos una orientación óptima para evitar singularidades y que tenga la mayor disponibilidad posible para llegar a los orificios. La posición y orientación del *tooldata* de la herramienta negra son las siguientes:

Tooldata de la herramienta negra						
Posición			Orientación (cuaternio)			
x	y	z	q1	q2	q3	q4
323,872	-0,644	84,317	0,7071	0	0,7071	0

Tabla 7 -Posición y orientación del tooldata de la herramienta negra

El *tooldata* de la herramienta negra en la estación de trabajo, le señalamos en la ilustración siguiente:

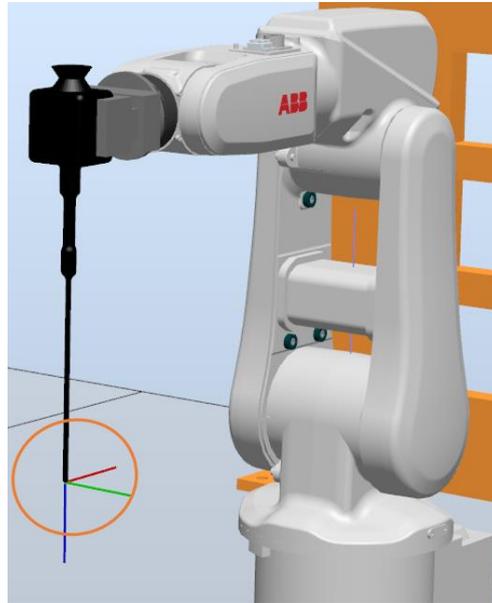


Ilustración 46 - Tooldata de la herramienta negra en la estación

Parametrizado este final de la herramienta, cuando se le comande realizar un movimiento de aproximación a los *robtargets*, de los orificios del *Pelvitainer*, los ejes X, Y y Z del *tooldata* y del *robtarjet* buscarán superponerse. Un ejemplo de esto lo tenemos en la siguiente ilustración:

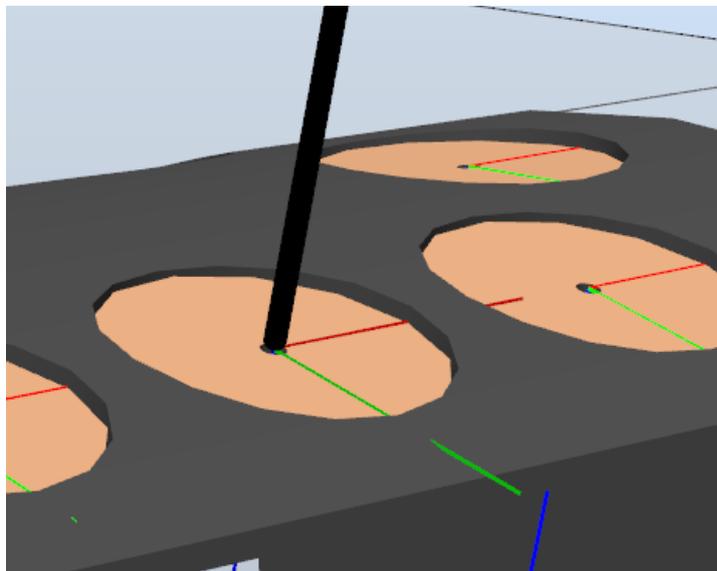


Ilustración 47 - Tooldata y robtarjet superpuestos en el orificio 2

Podemos notar el *tooldata* en el orificio 2 porque la longitud de los ejes del *tooldata* es mayor que las de los *robtargets* en la estación de trabajo.

3.3.2.2 Herramientas blancas y sus soportes

La posición del soporte de las herramientas blancas está caracterizada por el sistema de referencia "Sopor_BI". Las coordenadas POSE de este sistema de referencia, respecto al sistema de referencia *Wobj0* del robot se indican en la siguiente tabla:

Coordenadas del S.R. "Sopor_BI" asociado al soporte de herramientas blancas						
Posición			Orientación (cuaternio)			
x	y	z	q1	q2	q3	q4
441,428	-579,815	0,7554	0,501446	0,501705	-0,498779	0,49806

Tabla 8 - Coordenadas del S.R. "Sopor_BI" asociado al soporte de herramientas blancas

El POSE mostrado en la tabla anterior, corresponde al sistema de referencia del soporte de herramientas blancas que se muestra redondeado en amarillo en la Ilustración 48:

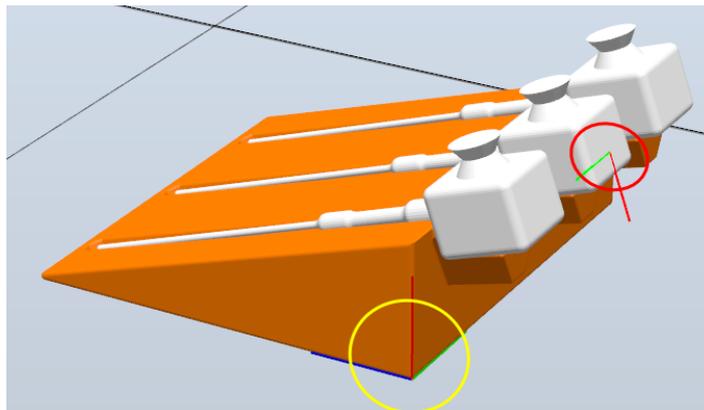


Ilustración 48 - *Wobjdata* del soporte de las herramientas negras y *robtarget* de la herramienta negra

Se ha creado un *robtarget* para este objeto de trabajo. En la Ilustración 48 está señalado con una circunferencia de color rojo. El punto está situado en la cara superior de la herramienta blanca. En todas las simulaciones solo se utilizará esa herramienta, no se usarán las otras dos. El POSE de la herramienta respecto al sistema de referencia *Sopor_BI* es el siguiente:

POSE de la herramienta blanca, referido al sistema de referencia "Sopor_BI"						
Posición			Orientación (cuaternio)			
x	y	z	q1	q2	q3	q4
114,78	150,531	-64,613	0,0031	0,1664	0,000754	-0,9861

Tabla 9 - Posición y orientación de la herramienta blanca

La orientación está cuidada para que cuando el robot acceda a ese punto, obtenga un agarre superior fácilmente. Cuando coja esa herramienta, el robot quedará con la herramienta agarrada de la siguiente forma:

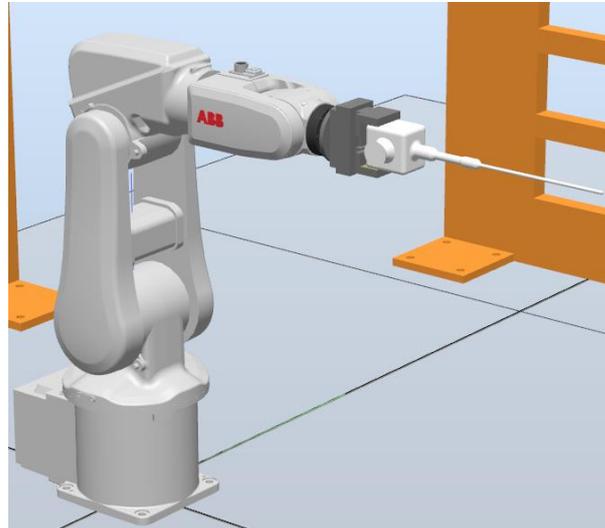


Ilustración 49 - Agarre lateral con la herramienta blanca

Igual que con la herramienta negra, necesitaremos implementar un *tooldata* para este agarre superior con la herramienta blanca.

Para obtener una posición y orientación precisa del final de esta herramienta, definiremos este parámetro en la estación. Realizaremos el mismo procedimiento que se ha realizado para crear el *tooldata* de la herramienta negra.

Igual que la herramienta negra, en la estación indicaremos la posición más precisa y centrada del final de la herramienta e introduciremos una orientación óptima para evitar singularidades y que tenga la mayor disponibilidad posible para llegar a los orificios. La posición y orientación del *tooldata* de la herramienta blanca son las siguientes:

Tooldata de la herramienta blanca						
Posición			Orientación (cuaternio)			
x	y	z	q1	q2	q3	q4
-0,193	-0,138	419,252	0	0	0	1

Tabla 10 - Posición y orientación del tooldata de la herramienta blanca.

El *tooldata* resultante de la herramienta blanca en la estación de trabajo, le señalamos en la ilustración siguiente:

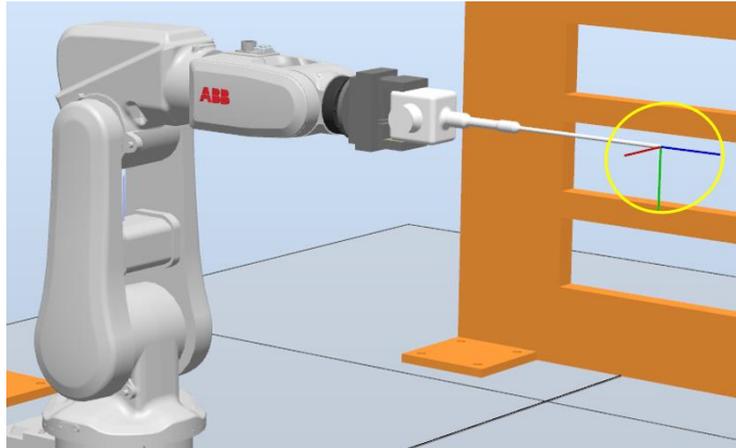


Ilustración 50 -Tooldata de la herramienta blanca en la estación

Como en el caso del *tooldata* de la herramienta negra, cuando se le comande realizar un movimiento de aproximación a los *robtargets*, de los orificios del *Pelvitainer*, los ejes X, Y y Z del *tooldata* y del *robtarjet* buscarán superponerse.

El robot IRB 120 tiene “poco” alcance debido a sus pequeñas dimensiones. Por esa razón, y teniendo en cuenta la longitud (grande) de las herramientas, no todos los puntos del *Pelvitainer* serán accesibles en todos los casos. Por ello, la **herramienta blanca solo podrá acceder a los orificios 1, 2 y 7**. Sin embargo, con la **herramienta negra podrá acceder a todos los orificios**.

3.3.3 Vallado en la estación de trabajo

El vallado al ser únicamente para mejorar el aspecto del entorno de simulación, no se le crearán objetos de trabajo ni, como consecuencia, puntos de trabajo.

Se implementarán tres vallados, uno a cada lado del robot y el otro justo detrás de él. Los dos vallados que se encuentran a cada lado del robot también cercarán a las herramientas con sus respectivos soportes. Se puede observar su posición final en la Ilustración 35.

4. Estructura del programa desarrollado para simular el robot en un entorno de cirugía laparoscópica

Dentro de este apartado, explicaremos cómo se ha estructurado el programa que permite que la simulación funcione. Se describirán los módulos del programa y sus variables importantes.

4.1 Parámetros declarados previos a los módulos

En el programa, antes de la programación de los módulos, se ha tenido que declarar diversos parámetros que serán utilizadas en el programa. No se mostrarán todos, solo las más relevantes.

4.1.1 Variables *pose* y numéricas

Se declararán numerosas variables *pose*. La variable *pose* se usa para cambiar de un eje de coordenadas a otro. Esta variable representa cómo se mueve y rota un sistema de referencia respecto a otro sistema de referencia. Dentro de los módulos veremos cómo se da uso a estas variables.

Se utilizan numerosas variables numéricas con la finalidad de ser usadas en bucles, para las opciones de un menú, etc.

4.1.2 Variables de los objetos de trabajo y puntos de los elementos 3D de la estación

También tenemos declarados el objeto de trabajo del *Pelvitainer* así como la posición y orientación de los orificios del *Pelvitainer* en *Rapid*. Mostramos solamente la posición de los *robtargets* de los orificios en el programa:

```
LOCAL CONST robtarget Ag_{9}:=  
[[[366.045393257,86.088024039,14.982760776],[0.991494578,0.129960623,0.000890996,-0.006924169],  
[[271.545385209,86.088081545,14.9827611],[0.991494578,0.129960623,0.000890996,-0.006924169],  
[[177.045377574,86.087139069,14.982755436],[0.991494578,0.129960623,0.000890996,-0.006924169],  
[[414.044576032,-1.636319327,3.216263727],[0.999995497,0.002993972,-0.000000617,0.000206044],  
[[221.944559788,-1.636481789,3.216262713],[0.999995497,0.002993972,-0.000000617,0.000206044],  
[[68.895,-1.688,3.164],[1,0,0,0],  
[[366.044748821,-89.491028972,13.962389636],[0.992248919,-0.124266065,0.000023515,0.00016587]  
[[271.544740773,-89.490971466,13.96238996],[0.992248919,-0.124266065,0.000023515,0.00016587],  
[[177.044732726,-89.49091396,13.962390284],[0.992248919,-0.124266065,0.000023515,0.00016587],
```

Ilustración 51 - Imagen de la posición y orientación de los orificios del *Pelvitainer* en *Rapid*

Para optimizar código, se ha decidido tener los 9 *robtargets* en un solo *array* de *robtargets*. Gracias a esto podremos programar de una forma más eficiente.

No se han mostrado todos los campos de los *robtargets* del *Pelvitainer*. Se ha mostrado solo los campos de la posición en X, Y y Z y los cuaternios q_1 , q_2 , q_3 y q_4 que se han mostrado en la Tabla 4 pero con todos sus decimales, por si se quería visualizar con más detalle estos valores. No se ha mostrado el objeto de trabajo del *Pelvitainer* porque ya se ha visto en la Tabla 3 y, en el código, no hay tanta diferencia respecto a los decimales de la tabla.

También, como variables declaradas tenemos, tanto el objeto de trabajo del soporte de herramientas negras, como el objeto de trabajo del soporte de las herramientas blancas. En consecuencia, tenemos los puntos de cada soporte: el punto por el que se cogerá la herramienta negra y el punto sobre el que se cogerá la herramienta blanca. A su vez, están también los *tooldata* de la herramienta negra y la herramienta blanca. Los valores de estos parámetros se han mostrado en el apartado 3.3.2 de la memoria.

4.1.3 Array de *robtargets* aux

También tenemos otro *array* de *robtargets* que es el siguiente:

```
local VAR robtarget aux_{128};
```

Ilustración 52 - Array de *robtargets* auxiliar

Durante la simulación, una vez dentro del *Pelvitainer*, el final de la herramienta accederá a diversas posiciones. Cada una de estas posiciones quedarán guardadas en este *array*. Por ejemplo, en mitad de una simulación, en la que estamos realizando movimientos por ángulos con el final de la herramienta, la posición en la que estemos se guardará, por ejemplo, en el *aux_{56}*, la siguiente posición que comandemos se guardará como *aux_{57}* y así sucesivamente.

El tamaño de este *array* es 128 porque el programa no permite una dimensión mayor. En caso de que se llegará al último valor del *array*, *aux_{128}*, se volvería a empezar por *aux_{1}* y así las veces que hicieran falta.

4.1.4 Jointtargets

En la declaración de variables, también se han declarado *jointtargets*. Las variables *jointtarget* son posiciones que se le comandará al robot mediante la configuración de los ejes del brazo robótico. Los *jointtargets* declarados en el programa son los siguientes:

```
local CONST jointtarget Punto_Base:=[[0,0,0,0,0,90],[1e9,1e9,1e9,1e9,1e9,1e9]];
local CONST jointtarget Punto_BaseA:=[[30,0,0,0,0,90],[1e9,1e9,1e9,1e9,1e9,1e9]];
local CONST jointtarget Punto_Base_1:=[[0,0,0,0,0,0],[1e9,1e9,1e9,1e9,1e9,1e9]];
local CONST jointtarget Punto_Base_2:=[[0,0,0,159,0,180],[1e9,1e9,1e9,1e9,1e9,1e9]];
```

Ilustración 53 - Jointtargets utilizados en el programa

Las unidades numéricas del primer campo están dadas en grados. Por ejemplo, para comprender mejor este parámetro, nos fijamos en la variable *Punto_Base_2* y observamos que tenemos en el primer campo los valores: [0,0,0,159,0,180]. Eso significa que la articulación 1, 2, 3 y 5 del brazo robótico se quedarán en su orientación original (0°), sin embargo, la articulación 4 girará 159 grados y la articulación 6 girará 180 grados. Estos *jointtargets* se usarán para dejar al robot con o sin la herramienta en una posición inicial segura o para evitar que entre en contacto con el *Pelvitainer* durante la simulación.

4.1.5 Tooldatas auxiliares y Pinz

A parte de los *tooldata* creados para la herramienta negra y blanca. Hará falta hacer uso de otros también:

```
LOCAL PERS tooldata Pinz:=[TRUE,[[0,0,83],[1,0,0,0]],[0.3,[0,0,1],[1,0,0,0],0,0,0]];
LOCAL PERS tooldata ini_:=[TRUE,[[0,0,83],[1,0,0,0]],[0.3,[0,0,1],[1,0,0,0],0,0,0]];
LOCAL PERS tooldata Herra_:=[TRUE,[[0.193,-0.138,419.252],[0,0,0,1]],[1,[0,0,1],[1,0,0,0],0,0,0]];
```

Ilustración 54 - Algunos de los tooldata utilizados

El *tooldata* llamado *Pinz* está referido a la pinza del robot. Se utilizará este *tooldata* para realizar movimientos cuando éste no tenga en su poder ninguna herramienta.

Observamos que hay otro *tooldata* llamado *ini_*, la finalidad de éste es que la herramienta no deje rastro al coger cualquiera de las herramientas. Es decir, únicamente cuando se cojan las herramientas, no se realizará con el *tooldata Pinz* sino con el *tooldata ini_*, que tendrán idénticos valores. Al no estar este parámetro en la estación, lograremos que se omita el rastro que deja el TCP de la pinza, durante la simulación. Para los demás movimientos que se realicen sin tener en poder ninguna herramienta seguiremos utilizando *Pinz*

debido a que no deja rastro como cuando coge las herramientas por primera vez. En conclusión, *ini_* es un parámetro auxiliar.

El *tooldata Herra* es otra variable auxiliar con un propósito parecido al del *tooldata ini_*. Cuando el robot tiene en su poder la herramienta negra o blanca, y se usan sus respectivos *tooldatas*, dejan rastro. Entonces, si usamos el *tooldata Herra* evitamos ese rastro de las herramientas. Dependiendo de qué herramienta se esté sujetando, el *tooldata Herra_* podrá tener el valor del *tooldata* de la herramienta blanca o negra.

4.1.6 *Robtargets aux* y *robtarget salir*

También, tenemos los siguientes *robtargets*:

```
local VAR robtarget aux1:=[[300,0,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
local VAR robtarget aux2:=[[300,0,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
local VAR robtarget aux3:=[[300,0,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
local VAR robtarget salir:=[[300,0,100],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

Ilustración 55 - Otros *robtargets* usados en el programa

Los *robtargets aux1*, *aux2* y *aux3* son parámetros auxiliares que se utilizarán para cuando se trabajen con variables *pose* que tomarán el valor de otros *robtargets*. Da igual el valor que tengan cuando comience la simulación porque, durante ella, cambiarán.

El *robtarget salir* es otro que tomará otro valor diferente al que tiene al empezar la simulación. Este parámetro se creó para que la herramienta, a la hora de salir del *Pelvitainer*, no entre en contacto con este. Tomará el valor que se le dé en el módulo *Salir_Pelvitainer*, que veremos más adelante, así, cuando salga la herramienta, ésta no tendrá una orientación tan pronunciada que pueda comprometer al *Pelvitainer*.

4.1.7 Variables numéricas 'A' y 'N'

Estas variables durante el programa tendrán un papel importante.

La variable 'A' decidirá en qué plano se realizarán los desplazamientos por ángulos, si 'A' vale 1 significará que se desplazará por el plano XZ. Mientras que si 'A' vale 2 se desplazará en el plano YZ. Por último, si 'A' vale 3 se realizará una rotación sobre el eje Z.

La variable 'N' se utilizará para definir sobre qué sistema de referencia se está trabajando, si sobre el sistema de referencia perpendicular al orificio o

perpendicular a la mesa. En caso de que se tenga como valor 1, significará que se está trabajando sobre el sistema de referencia perpendicular a la mesa y, si tiene como valor 2, significará que estamos en el sistema de referencia perpendicular a la mesa. Para comprender mejor estos casos se aconseja leer el apartado 5.3 de la memoria.

4.1.8 Variable numérica 't'

También cabe destacar que tenemos una variable numérica 't', esta variable será la encargada de acceder a los elementos del *array* de *robtargets*, *aux*. Cuando el programa empiece la variable 't' estará a 0, y pasará a tener valor 1 cuando se haya realizado una entrada con ángulo con la herramienta en el *Pelvitainer* o se haya realizado el primer desplazamiento por ángulos.

Una vez que esta variable valga 1, cada vez que se haga un movimiento dentro del *Pelvitainer* con la herramienta, ésta aumentará su valor una unidad. En otras palabras, por cada movimiento el valor de 't' se incrementará una unidad.

La posición del final de la herramienta que se da por cada valor de 't' se guardará en el *array* de *robtargets* *aux*. Por ejemplo, si estamos en un valor de 't' igual a 60, la posición que tenga el final de la herramienta se guardará en el *aux_{60}*.

Cuando se salga del orificio la variable 't' volverá a ser 0, y cuando se introduzca la herramienta de nuevo en otro orificio se volverá a dar el proceso comentado anteriormente.

4.2 Funciones usadas frecuentemente

Antes de comentar los módulos de la aplicación, es necesario comentar algunas de las funciones que se utilizarán más a menudo para que así se pueda comprender mejor el programa más adelante:

- **MoveJ:** El brazo robótico llevará el *tooldata*, indicado en esta función, con un movimiento no lineal al *robtarget* de un objeto de trabajo también especificado. Este movimiento se realizará a una velocidad configurable.
- **MoveL:** Idéntico al *MoveJ*, con la única diferencia de que el movimiento será lineal.
- **MoveAbsJ:** A partir de un *jointtarget* previamente definido, con esta función, el robot alcanzará la posición determinada por la configuración de los ejes. Esta configuración de los ejes se ha determinado articulación a articulación en la declaración del *jointtarget*. La velocidad a la que se realice se puede configurar. Como en las anteriores funciones, es necesario dar un *tooldata* para realizar el movimiento.
- **TPWrite:** Función para mostrar por pantalla al usuario un conjunto de palabras y/o valores numéricos que hagan falta ser conocidos o conocidas.
- **TPReadNum:** Función que necesita una variable numérica de entrada, en la que el usuario, durante la ejecución, establecerá su valor. A su vez, irá acompañado de un texto que contendrá lo que el programador vea necesario.
- **TPReadFk:** Función encargada de generar un menú por pantalla, que necesita una variable numérica como argumento de entrada. Se podrá escribir el texto que sea conveniente para las opciones a elegir y se mostrará antes que éstas. A cada opción se le puede asignar el nombre que se desee y dependiendo de la opción elegida, la variable numérica de entrada terminará teniendo un valor diferente.
- **OrientZYX:** Es una función con 3 campos de entrada. La función sirve para aplicar una rotación a un punto que podrá ser en el eje X, Y y/o Z. Podremos aplicarle al punto una rotación en un eje solamente, o en 2 a la vez, o incluso, en los 3 a la vez. El primer campo está dedicado a

la rotación en eje Z, el segundo el eje Y y el tercero en el eje X. Los parámetros de entrada estarán en grados.

4.3 Módulos de la aplicación: Definición y características

Ya comentadas algunas de las variables que se usarán, pasaremos a explicar los módulos que dan vida al programa. Dentro de la opción de *Rapid* de *RobotStudio*, el programa nos enseña un listado de estos módulos:

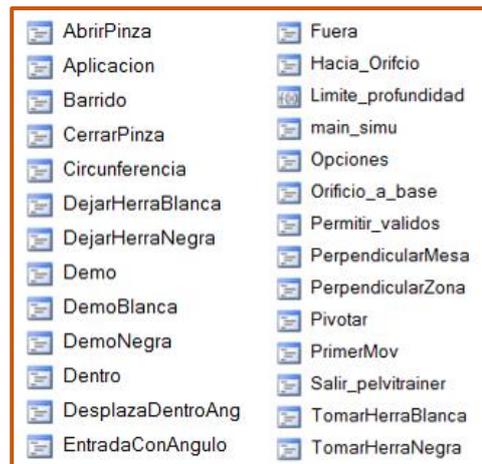


Ilustración 56 - Módulos del programa

A continuación, se definirán los módulos mostrados y se hará énfasis en los que sean más importantes. No se comentarán los módulos en el orden que aparece en la Ilustración 56. Como la mayoría de los módulos se ejecutan varias veces durante todo el código, ha sido complicado establecer un orden para explicarlos. Así que se ha decidido agruparles por las utilidades que tienen en común y describir los módulos grupo a grupo.

A mayores, en caso de que no se logre comprender totalmente algunos módulos se recomienda ver el apartado 5 de la memoria para ver, durante la simulación, los aportes que da el programa.

4.3.1 Módulos que servirán como interfaz de usuario

4.3.1.1 Módulo *Main_simu*

Cuando comienza la simulación, se ejecuta el módulo *M_Main* que contiene la función *main_simu* y ésta provoca que se ejecuten las demás, siguiendo la secuencia programada. El módulo *main_simu* es el siguiente:

```

PROC main_simu()

!Menú inicial de la simulación

TPERase;
AbrirPinza;
IF ini=0 THEN
    TPWrite "Hola, este es el menú principal.";
ENDIF
TPWrite "Herramienta blanca: Se utiliza con un agarre superior, solo podrá acceder";
TPWrite "a los orificios 1,2 y 7, pero permite realizar más movimientos en estos.";
TPWrite "Herramienta negra: Se utiliza con un agarre lateral,";
TPWrite "permite acceder a todos los orificios. ";

IF ini=0 THEN
    TPWrite "¿Qué herramienta deseas usar, ¿la herramienta blanca o negra?";
    TPReadFK bar,"También puedes usar la demo si lo deseas.",
        "H. Blanca","H. Negra","Demo",stEmpty,stEmpty;
ENDIF
ini:=0;
ini_:=Pinz;

IF bar=1 THEN
    Herra_:=Herra_Blanca;
    TomarHerraBlanca;
    Opciones;
ELSEIF bar=2 THEN
    Herra_:=Herra_Ng;
    TomarHerraNegra;
    Opciones;
ELSEIF bar=3 THEN
    Demo;
ENDIF

ENDPROC

```

Ilustración 57 - Código del módulo main_simu

Principalmente, este módulo se encarga de generar el menú inicial de la Ilustración 82 y direccionar el programa dependiendo de la opción que se haya elegido. Este menú se ha generado con la función *TPReadFK*.

Podemos ver que ciertos *TPWrite* están bajo la condición de *ini=0*. Esto significa que cuando se acceda al menú inicial (*ini=0*) se visualizarán todos los *TPWrite* de este módulo. Pero, al realizar un cambio de herramienta, en el momento que ocurra de la simulación (*ini* con un valor diferente a 0), solo se mostrarán los *TPWrite* fuera de las condiciones *IF*.

Asignamos el valor del *tooldata* de *Pinz* a *ini_* para que no deje rastro el TCP del *ABB IRB 120* al coger la herramienta por primera vez.

En caso de que decidamos coger la herramienta blanca o negra, se le asignará al *tooldata Herra_* el valor del *tooldata* de la herramienta que el robot vaya a agarrar. Con esto, lograremos evitar que la trayectoria, que siga la herramienta, deje rastro. A continuación, se hará uso de los módulos

TomarHerraBlanca o *TomarHerraNegra*. Después, se hará una llamada al módulo *Opciones* y comenzará a ejecutarse este.

Por último, si seleccionamos la opción '*Demo*', se ejecutará el módulo *Demo*.

4.3.1.2 Módulo *Opciones*

En este módulo, que carece de argumentos de entrada, se encargará de que se nos muestre por pantalla, gracias a la función *TPReadFK*, el menú de la Ilustración 84 o el de la Ilustración 86 mediante condiciones *IF*. Estas condiciones dependerán de si se ha seleccionado la opción '*Herramienta*' en el menú de la Ilustración 84 o en el de la Ilustración 121. Este módulo se encargará de direccionar las llamadas a los módulos que sean necesarios a partir de las opciones seleccionadas y mediante condiciones *IF*.

En caso de que elijamos la opción '*Aplicación*' se hará una llamada al módulo *Aplicación* y se ejecutará.

En caso de que escojamos la opción '*Herramienta*' se depositará la herramienta, que esté agarrada, en su respectivo soporte con los módulos *DejarHerraBlanca* y *DejarHerraNegra*.

```
ELSEIF op=3 THEN

    ini:=1;
    IF bar=1 THEN
        DejarHerraBlanca;
        bar:=2;
    ELSE
        DejarHerraNegra;
        bar:=1;
    ENDIF
    main_simu;
```

Ilustración 58 - Código que se utilizará en caso de seleccionar '*Herramienta*' en el módulo *Opciones*

Después se ejecutará el módulo principal *main_simu*, pero al haberse establecido la variable *ini* igual a 1, el módulo principal solo servirá para mostrar un mensaje por pantalla, coger la herramienta y volver ejecutar el módulo *Opciones*. Con lo cual, quedaremos en el menú de la Ilustración 86 del módulo *Opciones*.

Estos dos posibles menús, estarán en un bucle *while*, en el que la única manera de salir de él es seleccionando la opción '*Salir*', que, mediante la

función *GOTO*, nos devolverá al menú inicial de la aplicación mediante la llamada que se hará al módulo *main_simu*.

4.3.1.3 Módulo *Aplicacion*

Este proceso sin argumentos de entrada se activará al seleccionar la opción 'Aplicación' en el menú de la Ilustración 84 o la Ilustración 86 o la opción 'Orificio' del menú de la Ilustración 121.

En este módulo se le preguntará al usuario sobre qué orificio trabajar. Se le aplicará el módulo *Permitir_validos* en caso de que se introduzca un orificio erróneo. Con el *MoveJ*, el final de la herramienta se dirigirá al orificio y, con la llamada que se ha hecho al módulo *Hacia_Orificio*, no entrará en contacto con el *Pelvitainer* al acercarse al orificio.

Mediante el uso de condiciones *if* tendremos los menús de la Ilustración 91, la Ilustración 99 o la Ilustración 100 que dependerá del orificio al que nos acerquemos y de la herramienta que el robot sujete. Como los demás menús, los de este módulo se generan a partir de las funciones *TPWrite* o *TPReadFK*.

```
IF i=4 OR i=5 OR i=6 THEN
  TPErase;
  TPWrite "¿Deseas introducir la herramienta con ángulo, ";
  TPRReadFK N,"o sin ángulo? ",
  "Sin ángulo",stEmpty,"Ángulo","Salir",stEmpty;
ELSE
  IF i<>3 THEN
    TPErase;
    TPWrite "¿Deseas introducir la herramienta con ángulo, ";
    TPRReadFK N,"perpendicularmente a la mesa o al orificio? ",
    "A la mesa","Al orificio","Ángulo","Salir",stEmpty;
  ELSE
    TPErase;
    TPWrite "¿Deseas introducir la herramienta con ángulo, ";
    TPRReadFK N,"perpendicularmente a la mesa o al orificio? ",
    stEmpty,"Al orificio","Ángulo","Salir",stEmpty;
  ENDIF
ENDIF
```

Ilustración 59 - Fragmento de código del módulo *Aplicacion*, que muestra los posibles menús que pueden aparecer

Tras seleccionar cualquiera de las opciones de los menús mostrados, pasaremos al siguiente módulo *PrimerMov*.

4.3.1.4 Módulo *PrimerMov*

Módulo, sin argumento de entrada, que, dependiendo de la opción seleccionada en el módulo *Aplicación*, anteriormente comentado, se encargará de realizar las llamadas a los módulos necesarios.

Principalmente el módulo está formado por 4 condiciones *if*, cada condición es una opción del menú proveniente del módulo *Aplicación*.

Para el caso de las opciones perpendicular a la mesa y al orificio se aplicará la función *Limite_profundidad*, para que se introduzca un valor correcto de profundidad. Posteriormente, se les aplicará sus respectivos módulos a estas dos opciones *PerpendicularMesa* y *PerpendicularZona*.

```
!Perpendicular a la mesa de trabajo
IF N=1 THEN
  TPErase;
  TPRReadNum dist1,"¿Cuántos milímetros deseas que avance?";
  dist1:=Limite_profundidad(dist1);
  y:=0;
  PerpendicularMesa Ag_{i},dist1;
  v:=v+1;

!Perpendicular al orificio
ELSEIF N=2 THEN
  TPErase;
  TPRReadNum dist1,"¿Cuántos milímetros deseas que avance?";
  dist1:=Limite_profundidad(dist1);
  y:=0;
  PerpendicularZona Ag_{i},dist1;
  v:=v+1;
```

Ilustración 60 - Código en caso de seleccionar que la herramienta entre perpendicularmente a la mesa o al orificio

En el caso de la opción de entrada con ángulo se le preguntará al usuario los parámetros necesarios para esta acción. Los parámetros se pedirán mediante *TPReadNum* y *TPReadFK* y finalmente se hará llamamiento al módulo *EntradaConAngulo*.

```
TPReadFK A,"¿En qué plano deseas trabajar?",
  "Plano XZ","Plano YZ",stEmpty,stEmpty,stEmpty;
TPReadNum r,"¿Cuántos grados quieres que se desplace?";
TPReadFK z_ent,"¿Deseas que gire en el eje Z?",
  "Sí","No",stEmpty,stEmpty,stEmpty;

IF z_ent=1 THEN
  TPRReadNum z_ang,"¿Cuántos grados deseas que gire?";
ENDIF

IF i<>4 AND i<>5 AND i<>6 THEN
  TPRReadFK rec,"¿Deseas que entre con ángulo respecto a la mesa o al orificio?",
  "A la mesa","Al orificio",stEmpty,stEmpty,stEmpty;
ENDIF

TPReadNum dist1,"¿Cuántos milímetros deseas que avance?";
```

Ilustración 61 - Código en caso de seleccionar que la herramienta entre con ángulo

En caso de que hayamos seleccionado ‘Salir’ en el módulo *Aplicación* se ejecutará el módulo *Fuera*. Si se ha seleccionado una opción diferente a ‘Salir’ se ejecutará el módulo *Dentro* después de que se realice el primer movimiento.

4.3.1.5 Módulo *Dentro*

Este proceso, que no tiene argumentos de entrada, se encargará de mostrar el menú de la Ilustración 123 y direccionar el programa hacia los módulos adecuados a la opción seleccionada.

Si se selecciona ‘Por ángulos’ se le pedirá al usuario los parámetros necesarios mediante *TPReadFk* y se ejecutará su propio módulo, *DesplazaDentroAng*. Ejecutada esta opción, la variable numérica *N* pasará a ser siempre de valor 3 hasta que se salga del módulo *Dentro*.

```
IF l=2 THEN

  !Seleccionar plano
  TPErase;
  IF t=0 THEN
    TPRReadFK A,"¿En qué plano deseas trabajar?",
    "Plano XZ","Plano YZ",stEmpty,stEmpty,stEmpty;
  ELSE
    TPRReadFK A,"¿En qué plano deseas trabajar?",
    "Plano XZ","Plano YZ","Eje Z",stEmpty,stEmpty;
  ENDIF

  IF ind=1 AND t>0 THEN
    ind:=ind+1;
  ENDIF
  dist1:=y;
  TPRReadNum r,"¿Cuántos grados quieres que se desplace?";
  DesplazaDentroAng r,dist1,Ag_{i};
```

Ilustración 62 - Código para la opción del cambio de orientación, en el módulo *Dentro*

En caso de que elijamos la opción ‘Cambiar h’, dependiendo del módulo que se haya ejecutado al introducir la herramienta por primera vez, se volverá a ejecutar ese mismo para cambiar la profundidad.

Sin embargo, si previamente a seleccionar esta opción, se ha hecho ya un desplazamiento por ángulos, siempre se ejecutará el módulo *EntradaConAngulo* porque la variable numérica ‘*N*’ será igual a 3. A pesar de que se esté ejecutando el módulo *EntradaConAngulo* para este caso, en la simulación la herramienta cambiará la distancia introducida manteniendo la orientación que tenía.

```

ELSEIF l=3 THEN
  TPErase;
  IF N=1 THEN

    TPReadNum dist1,"¿Cuántos milímetros deseas que avance?";
    PerpendicularMesa Ag_{i},dist1;

  ELSEIF N=2 THEN

    TPReadNum dist1,"¿Cuántos milímetros deseas que avance?";
    PerpendicularZona Ag_{i},dist1;
  ELSEIF N=3 THEN

    TPReadNum dist3,"¿Cuántos milímetros deseas que avance?";
    EntradaConAngulo r,dist3,Ag_{i};
    dist1:=y;
  ENDIF

```

Ilustración 63 - Código para la opción del cambio de profundidad, en el módulo Dentro

El menú del módulo *Dentro* estará en un bucle *while*, la única manera de finalizar el bucle es con la opción 'Salir'. Al seleccionar esta opción, el robot sacará la herramienta cuidadosamente gracias al *robtarger salir* que se generó en el módulo *Salir_Pelvitainer*. Una vez que ya esté fuera la herramienta del *Pelvitainer*, saldremos del bucle *while* de este módulo *Dentro* y se ejecutará el módulo *Fuera*.

4.3.1.6 Módulo *Fuera*

Al ejecutarse este proceso sin parámetros de entrada, se activará el módulo *Orificio_a_base* para que el robot, con la herramienta que esté sujetando, vaya a una posición base sin que choque con el *Pelvitainer* desde el orificio en el que estaba.

Mediante *TPWrite* y *TPReadFK* tendremos el menú de la Ilustración 121. En este módulo, mayoritariamente se usan condiciones *IF* para distinguir cada opción. Si seleccionamos la opción 'Orificio' se ejecutará el módulo de *Aplicación*.

En caso de que escojamos la opción 'Herramienta' se depositará la herramienta que esté agarrada en su respectivo soporte con los módulos *DejarHerraBlanca* o *DejarHerraNegra*. El código es similar al de la Ilustración 58 y el resto de las acciones que quedan por ocurrir, para esta opción de 'Herramienta', serán idénticas a las de esa ilustración.

Si elegimos la opción 'Salir' se dejará la herramienta que esté sujeta con los módulos *DejarHerraBlanca* O *DejarHerraNegra*. Realizada esta acción se ejecutará el proceso principal *main_simu* y estaremos en la misma situación que al comenzar la simulación.

```
ELSEIF out=4 THEN
    IF bar=1 THEN
        DejarHerraBlanca;
    ELSE
        DejarHerraNegra;
    ENDIF
    main_simu;
ENDIF
```

Ilustración 64 - Código para la opción 'Salir', en el módulo Fuera

4.3.2 Módulos encargados de los movimientos principales de las herramientas

En todos los bucles *FOR* que veremos, en este apartado 4.3.2, a continuación, vemos que el paso está impuesto por una variable *signo* o *signoz*. Los valores de este paso serán, o bien positivos, o bien negativos dependiendo si la variable de entrada de distancia o ángulo es positiva o negativa.

4.3.2.1 Módulo *PerpendicularZona*

```
LOCAL PROC PerpendicularZona(robtarget punto,num dist1)
```

Este módulo tiene como argumentos de entrada el *robtarget* del orificio, sobre el que se va a trabajar, y la variable numérica, que definirá la profundidad a la que se va a introducir la herramienta.

Con este módulo conseguiremos que la herramienta se introduzca perpendicularmente al *robtarget* del orificio. La parte clave de este módulo que hace que se lleve a cabo esta tarea es la siguiente:

```
FOR dist_ FROM 0 TO dist1 STEP signo DO
  ConFL\off;
  pose1.rot:=punto.rot;
  pose1.trans:=punto.trans;
  pose2:=[[0,0,y+dist_],[1,0,0,0]];
  pose3:=PoseMult(pose1,pose2);
  aux_{t+1}.rot:=pose3.rot;
  aux_{t+1}.trans:=pose3.trans;
  moveL aux_{t+1},v7000,fine,Herra_\WObj:=Pelv2;
ENDFOR
```

Ilustración 65 - Bucle encargado de introducir la herramienta perpendicularmente al orificio

Es un bucle *FOR*, en el que generaremos el movimiento característico de este módulo mediante las variables *pose*. Inicialmente, hacemos que el *pose1* tenga la orientación y posición del *robtarget* del orificio sobre el que se esté trabajando.

Luego, el *pose2* describirá solo el desplazamiento en Z, con un valor que irá aumentando por cada iteración del bucle hasta alcanzar el valor de la profundidad que tenemos a partir de la variable de entrada *dist1*.

Calculamos el producto de *pose1* y *pose2*, y tendremos un punto con la orientación y posición que tenía el *robtarget* del orificio, pero desplazado en su propio eje Z los milímetros que defina la variable *dist1*.

4.3.2.2 Módulo *PerpendicularMesa*

```
LOCAL PROC PerpendicularMesa(robotarget punto,num dist1)
```

Como el anterior, este módulo tiene como argumentos de entrada el *robotarget* del orificio sobre el que se va a trabajar y la variable numérica que definirá la profundidad a la que se va a introducir la herramienta.

Ejecutando este módulo, la herramienta entrará perpendicularmente al plano de la mesa. Este módulo es prácticamente idéntico al anterior, la única diferencia reside en el bucle *FOR* encargado de efectuar el movimiento de la herramienta.

```
FOR dist_ FROM 0 TO dist1 STEP signo DO

    ConFL\off;
    pose1.rot:=Ag_{6}.rot;
    pose1.trans:=punto.trans;
    pose2:=[[0,0,y+dist_],[1,0,0,0]];
    pose3:=PoseMult(pose1,pose2);
    aux_{t+1}.rot:=pose3.rot;
    aux_{t+1}.trans:=pose3.trans;
    moveL aux_{t+1},v100,fine,Herra_\WObj:=Pelv2;

ENDFOR
```

Ilustración 66 - Bucle encargado de introducir la herramienta perpendicularmente a la mesa

Si nos fijamos, como única diferencia, la orientación establecida para el *pose1* es diferente a la del módulo anterior. En lugar de que la orientación descrita por el *pose1* sea la orientación del *robotarget* del orificio sobre el que se esté trabajando, será la orientación del *robotarget* del orificio 6.

Si nos fijamos en el *robotarget* del orificio 6 del *Pelvitainer*, en la estación de trabajo de *RobotStudio*, podemos fijarnos que su eje Z se encuentra totalmente perpendicular al plano de la mesa. Por lo tanto, el resultado final, del producto entre *pose1* y *pose2*, al que se le aplicará un *MoveL*, tendrá una orientación perpendicular a la mesa.

En conclusión, tenemos un *pose1* con una posición que será la del *robotarget* en el que se está operando y una orientación en la que el eje Z estará perpendicular a la mesa. Un *pose2* que, únicamente, define el desplazamiento en Z.

El producto resultante de estos dos será un punto con la orientación del *pose1*, pero desplazado, respecto a la posición anterior que tenía el *pose1*, una distancia *dist1* en su eje Z.

4.3.2.3 Módulo *DesplazaDentroAng*

```
LOCAL PROC DesplazaDentroAng(num R,num y,robtarget punto)
```

Módulo encargado de desplazar el final de la herramienta mediante ángulos, estando ya el final de la herramienta dentro del *Pelvitainer*. Este módulo tiene como variables de entrada la variable numérica que describe el valor en grados del ángulo que recorrerá el final de la herramienta, la variable numérica que define la distancia a la que estará introducida la herramienta y el *robtarget* del orificio en el que se realice esta acción.

La parte principal de este módulo, que lleva a cabo su acción característica, es la que se muestra a continuación:

```
FOR R_ FROM 0 TO R STEP signo DO

  Confl\Off;
  pose1.trans:=punto.trans;

  IF t=0 THEN
    IF N=1 THEN

      IF A=1 THEN
        pose1.rot:=OrientZYX(0,R_,0)*Ag_{6}.rot;
      ELSEIF A=2 THEN
        pose1.rot:=OrientZYX(0,0,R_)*Ag_{6}.rot;
      ENDIF

    ELSEIF N=2 THEN
      IF A=1 THEN
        pose1.rot:=OrientZYX(0,R_,0)*punto.rot;
      ELSEIF A=2 THEN
        pose1.rot:=OrientZYX(0,0,R_)*punto.rot;
      ENDIF
    ENDIF
  ENDIF

  IF t>0 THEN

    IF A=1 THEN
      pose1.rot:=OrientZYX(0,R_,0)*aux_{t}.rot;
    ELSEIF A=2 THEN
      pose1.rot:=OrientZYX(0,0,R_)*aux_{t}.rot;
    ELSEIF A=3 THEN
      pose1.rot:=OrientZYX(R_,0,0)*aux_{t}.rot;
    ENDIF

  ENDIF

  pose2:=[[0,0,y],[1,0,0,0]];
  pose3:=PoseMult(pose1,pose2);

  aux_{t+1}.trans:=pose3.trans;
  aux_{t+1}.rot:=pose3.rot;

  MoveL aux_{t+1},v100,fine,Herra_\WObj:=Pelv2;

ENDFOR
```

Ilustración 67 - Bucle encargado de cambiar la orientación de la herramienta

Observamos que tenemos dos situaciones distintas, una para cuando 't' es igual 0 y otra para cuando 't' es mayor que 0. Como se observa en el código, la situación en la que estemos repercutirá principalmente en el valor de la orientación del *pose1*.

La situación cuando 't' es igual a 0 se da cuando, después de haberse introducido la herramienta por primera vez en el *Pelvitainer* perpendicularmente al plano de la mesa o al orificio, se realiza un desplazamiento por ángulos. Dependiendo de las condiciones *IF* impuestas por las variables numéricas 'A' (plano sobre el que se realizará el ángulo) y 'N' (S.R. perpendicular a la mesa o al orificio) tendremos un diferente resultado.

A continuación, se muestra un ejemplo para comprender mejor la parte del código de esta función:

Si tras introducir la herramienta blanca perpendicularmente al agujero 80 milímetros en el orificio 2, queremos realizar un desplazamiento por ángulos de 15 grados en el plano XZ, el procedimiento del código será el siguiente:

Primero, *pose1* tendrá la posición del *robtarger* del orificio 2. La orientación de *pose1* será la del *robtarger* del orificio 2 al que se le aplicará un *OrientZYX* de 15 grados en el campo de Y. La posición y orientación del *pose1* en la estación de trabajo aplicando este *OrientZYX* mencionado sería la siguiente:

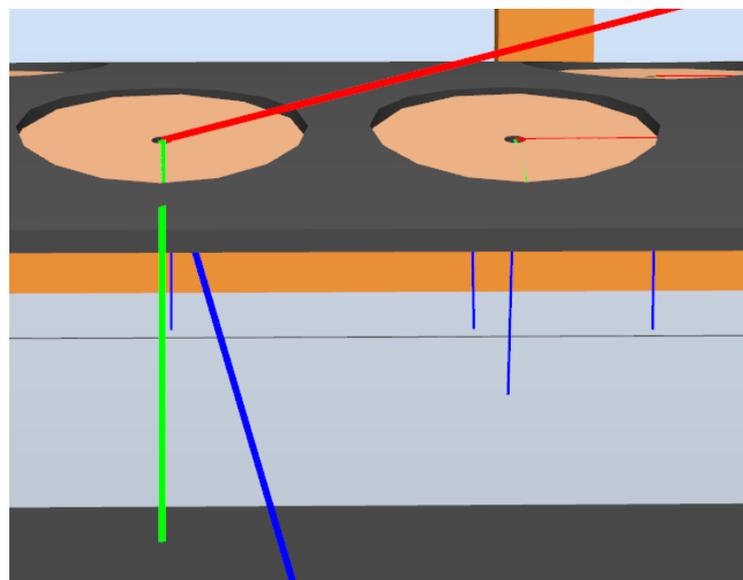


Ilustración 68 - Rotación en Y en *robtarger* de orificio 2

Si comparamos la orientación de este *pose1* con la del *robtarger* del orificio 1, podemos ver que son notoriamente diferentes. Al realizar una rotación en el eje Y, la orientación de los ejes X y Z ha cambiado.

Pose2 exclusivamente definirá un desplazamiento en Z de 80 milímetros. Entonces, al hacer un producto de *pose1* y *pose2*, quedaría un *pose1* al que se le ha aplicado un desplazamiento en Z de 80 milímetros:

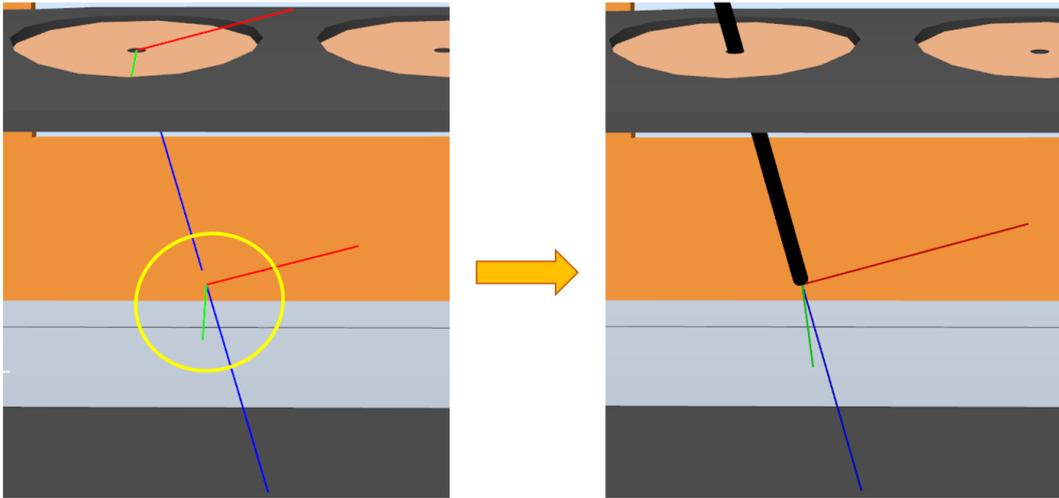


Ilustración 69 - Pose1 desplazado 80 milímetros en el eje Z

En la ilustración de la izquierda, observamos que el resultado es igual que *pose1*, pero con el desplazamiento en Z aplicado de 80 milímetros. En la ilustración de la derecha, vemos el resultado, durante la simulación, de realizar un movimiento lineal al resultado del producto de *pose1* y *pose2*. Observamos, cómo el *tooldata* de la herramienta negra se encuentra en el punto calculado.

La situación cuando 't' es mayor que 0 se da, o bien después de haber entrado la herramienta con ángulo, o bien después de la situación descrita para 't' igual a 0.

Para esta situación, en cuanto a la programación, la diferencia es que, al definir el *pose1* en lugar de usar la rotación del *robtarger* del orificio, se usa la rotación de la posición anterior alcanzada por la herramienta, *aux_{t}*.

Esto significa que, los cálculos que hemos visto se aplican a la última posición y orientación alcanzada del *tooldata* de la pieza. El resto del bucle realizaría el mismo procedimiento que el visto para la primera situación.

4.3.2.4 Módulo *EntradaConAngulo*

```
LOCAL PROC EntradaConAngulo(num R,num dist3,robtarjet punto)
```

Esta función tiene dos usos, el primero es hacer que la herramienta entre al orificio con un ángulo predeterminado y el segundo es realizar un cambio en la profundidad cuando la herramienta tenga una orientación distinta a perpendicular a la mesa o al orificio. Los argumentos de entrada serán la orientación que tendrá la herramienta, la profundidad y el *robtarjet* del orificio sobre el que se esté trabajando.

Para el primer uso, teniendo en cuenta las elecciones hechas en la Ilustración 61, se ejecutará el módulo *Pivotar*. Al ser un módulo que únicamente se ejecuta en este módulo, se comentará su función como un subapartado del módulo *EntradaConAngulo*.

4.3.2.4.1 Módulo *Pivotar*

```
local PROC Pivotar(robtarjet punto,num R)
```

Los argumentos de entrada de este módulo son el *robtarjet* en el que se está trabajando y el ángulo que pivoteará.

Como indica su nombre, este módulo estará dedicado a realizar el pivoteo. El pivoteo en los planos XZ o YZ se realizará esencialmente con el siguiente código de este módulo:

```
aux.trans:=punto.trans;  
  
FOR Rz FROM 0 TO R step signo DO  
  IF A=1 THEN  
    aux.rot:=OrientZYX(0,Rz,0)*aux3.rot;  
  ELSEIF A=2 THEN  
    aux.rot:=OrientZYX(0,0,Rz)*aux3.rot;  
  ENDIF  
  
  moveL aux,v50,fine,Herra_\WObj:=Pelv2;  
  
ENDFOR
```

Ilustración 70 - Código encargado del pivoteo

El parámetro *aux3*, será el S.R. perpendicular a la mesa o al orificio, depende de los parámetros que hayamos seleccionado. Básicamente, para una posición en el *robtarjet* del orificio, le aplicamos una rotación, con *OrientZYX*,

a *aux3*, en X o en Y, que también depende de los parámetros que hayamos seleccionado.

Luego, en caso de que queramos que la herramienta pivotee alrededor del eje Z, se ejecutará otro bucle en el que se le aplica al punto resultante *aux*, de la Ilustración 70, una rotación en Z, con *OrientZYZ*. Al punto resultante de este pivoteo en Z se le llamará *aux2*. Entonces, terminando con este módulo, a *pose1* se le dará el valor de *aux* o *aux2*, esto dependerá de si se ha escogido rotar en Z. Con esto comentado, ya se habría simulado el pivoteo y finalizaría el módulo *Pivotar*.

Faltaría que se introdujera la herramienta para la orientación alcanzada con el pivoteo, de eso se encargará el módulo *EntradaConAngulo*. Así que, volviendo al módulo *EntradaConAngulo*, después de la llamada al módulo *Pivotar*, tenemos el siguiente bucle *FOR*:

```
FOR dist_ FROM 0 TO dist3 STEP signo DO

  Confl\off;
  pose1.trans:=punto.trans;
  pose2:=[[0,0,y+dist_],[1,0,0,0]];

  pose3:=PoseMult(pose1,pose2);

  aux_{t+1}.trans:=pose3.trans;
  aux_{t+1}.rot:=pose3.rot;
  MoveL aux_{t+1},v100,fine,Herra_\WObj:=Pelv2;

ENDFOR
```

Ilustración 71 - Bucle del módulo *EntradaConAngulo*

Tenemos *pose1* con orientación definida en el módulo *Pivotar* y la posición del orificio. *Pose2* describe el desplazamiento en Z. Con lo cual al hacer el producto de *pose1* y *pose2* tendremos *pose1* desplazado, en su eje Z, los milímetros comandados.

Recordamos que el segundo uso que tiene este módulo es cambiar la profundidad para un desplazamiento por ángulos ya realizado por el final de la herramienta.

No se ejecutará el módulo *Pivotar* y se ejecutará directamente el bucle visto en la Ilustración 71. Al no ejecutarse *Pivotar*, no tendremos una orientación para *pose1* definida en el módulo *EntradaConAngulo*. Entonces, la orientación del *pose1* será la definida en el módulo *DesplazaDentroAng*. Definida ya la rotación de *pose1*, se ejecutaría el bucle de la Ilustración 71 de la misma forma y con la misma lógica que hemos visto anteriormente.

4.3.3 Módulos encargados de permitir parámetros correctos

4.3.3.1 Módulo *Permitir_validos*

Este proceso, sin argumentos de entrada, se ejecutará automáticamente si el usuario introduce un valor numérico erróneo al seleccionar el orificio.

Si tenemos la herramienta negra, se le avisará al usuario, en caso de que haya comandado un valor de orificio mayor que 9 o menor que 1, se le recomendará introducir un valor entre 1 y 9. Lo mismo ocurre con la herramienta blanca, si damos un valor de orificio distinto de 1, 2 o 7, se le avisará al usuario que no es válido y se le pedirá introducir uno de los tres valores correctos.

Esta situación se dará tantas veces como el usuario introduzca un valor numérico erróneo. Básicamente se utilizan bucles *while*, y se hace uso de las funciones *TPWrite* para informar al usuario y *TPReadNum* para solicitar el nuevo valor. Un fragmento, que sirve como ejemplo para este módulo es el siguiente:

```
WHILE bar=2 AND (i>9 OR i<1) DO
  TPErase;
  TPWrite "Error, le recordamos que con la herramienta negra";
  TPWrite "solo se puede acceder a los orificios del 1 al 9.";
  TPRReadNum i,"¿A qué orificio quieres que se desplace?";
ENDWHILE
```

Ilustración 72 - Fragmento de código del módulo *Permitir_validos*

4.3.3.2 Módulo *Limite_profundidad*

A diferencia de los demás módulos, éste es una función, por lo tanto, al finalizar su ejecución devolverá una variable.

Esta función se encargará de que, en caso de que se introduzca un valor erróneo de profundidad, al meter la herramienta por primera vez perpendicularmente a la mesa o al orificio, el programa le diga al usuario la profundidad límite. Posteriormente, le volverá a pedir el valor numérico que definirá cuánto se introduzca la herramienta. El argumento de entrada de esta función es la variable numérica de la profundidad que tendrá la herramienta.

Al estar esta cuestión en un bucle *while* se le preguntará al usuario, este valor, las veces que sea hasta que se introduzca un valor correcto. Para preguntar e informar al usuario se utiliza *TPReadNum* y *TPWrite* respectivamente. Para distinguir cada caso, es decir, el orificio en el que se encuentre y la barra que se utilice, se hace uso de condiciones *IF* y de la función *TEST* respectivamente. Además, se distinguirá respecto a qué S.R. entra la herramienta con otra condición *IF* aplicada a la variable 'N'.

Se muestra un fragmento del código de esta función para ver la estructura que se ha estado comentando:

```
ELSEIF N=1 THEN
  !Perpendicular a la mesa
  TEST bar
  CASE 1:
    IF i=1 OR i=2 OR i=7 THEN
      WHILE dist1>170 DO
        TPErase;
        TPWrite "Error, la profundidad máxima permitida son 170 milímetros";
        TPClear dist1,"¿Cuántos milímetros deseas que avance?";
      ENDWHILE
    ENDIF
```

Ilustración 73 - Fragmento de código de la función Límite_profundidad

El valor que se devolverá al terminar la función es la distancia a la que se terminará introduciendo la herramienta.

```
RETURN dist1;
```

Ilustración 74 - Parámetro que se devolverá al finalizar la función Limite_profundidad

Se ha necesitado devolver este parámetro porque es necesario para los siguientes módulos que impliquen realizar movimientos con la herramienta.

4.3.4 Módulos utilizados para la opción demo

4.3.4.1 Módulo *Demo*

Este proceso, que carece de argumentos de entrada, entrará en ejecución, cuando se haya seleccionado la opción “*Demo*” en el menú inicial, que proviene del proceso principal *main_simu*.

Este módulo tendrá la finalidad de mostrar por pantalla, al usuario, el menú de la *demo* con sus cinco opciones. También, se encargará de direccionar el programa al módulo correspondiente a la opción seleccionada en este menú. A diferencia de los demás menús, para este caso no usamos la función *TPReadFK*, sino que hacemos uso exclusivo de *TPWrite* y *TPReadNum*.

```
TPWrite "Esta es la demo, puedes elegir entre las 5 siguientes opciones:";
TPWrite "1- Demostración con herramienta blanca";
TPWrite "2- Demostración con herramienta negra";
TPWrite "3- Barrido";
TPWrite "4- Circunferencia";
TPWrite "5- Salir";
TPReadnum dem, "¿Qué opción deseas?";
```

Ilustración 75 - Código del menú que sale por pantalla al entrar en la opción *Demo*

A partir del valor numérico que dé el usuario, para la opción que se vaya a usar de la *demo*, la función *TEST*, se encargará de realizar la llamada al módulo vinculado al número de la opción. Dentro de la función *TEST*, haremos uso de la función *TPWrite* para informar al usuario de en qué opción nos encontramos.

```
TEST dem
CASE 1:
  !Con agarre superior
  TPWrite "Opción demo con herramienta blanca, agarre superior";
  DemoBlanca;
CASE 2:
  !Con agarre lateral
  TPWrite "Opción demo con herramienta negra, agarre lateral";
  DemoNegra;
CASE 3:
  TPWrite "Opción demo barrido";
  Barrido;
CASE 4:
  TPWrite "Opción demo circunferencia";
  Circunferencia;
CASE 5:
  GOTO out;
ENDTEST
```

Ilustración 76 - Uso de la función *TEST* en el código de la *demo*

Este menú se encontrará en un bucle que se ejecutará tantas veces como se seleccione cualquiera de las cuatro primeras opciones. La única forma de

salir de este bucle, y como consecuencia, de este módulo, es seleccionando la opción cinco, salir, que hace uso de *GOTO* para finalizar este proceso.

Después del bucle, se encuentra de nuevo el proceso *main_simu*, el cual, genera por pantalla el menú inicial. En otras palabras, al salir del bucle, seleccionando la opción 5, se ejecuta directamente el proceso *main_simu*.

4.3.4.2 Módulos *DemoBlanca* y *DemoNegra*

Estos módulos *DemoBlanca* y *DemoNegra*, que no cuentan con argumentos de entrada, se ejecutarán cuando se seleccione la opción 1 o 2, respectivamente, del menú de la *demo*. En el apartado 5 de la memoria se comentará los movimientos que lleva a cabo la herramienta en estas dos posibles opciones elegidas.

Los módulos comenzarán cogiendo su respectiva herramienta con su módulo de coger herramienta y llevando el final de ésta al orificio sobre el que se vaya a trabajar.

En el caso de la *DemoBlanca*, el primer movimiento que se realiza es haciendo uso del módulo *PerpendicularZona* que ya hemos visto lo que hace. Mientras que, en el caso de la *DemoNegra*, el primer movimiento que hace la herramienta se realiza ejecutando el proceso *PerpendicularZona*, el cual, también, ya hemos visto cómo funciona.

Para el resto de los movimientos de estas dos opciones, la lógica a seguir es idéntica:

- Cuando se realice un cambio de profundidad, respetando la orientación que tenga la herramienta, se hará uso del proceso *EntradaConAngulo*.

Este proceso como parámetro de entrada tendrá el valor numérico de cuántos milímetros subirá o bajará respecto a la profundidad a la que se encontraba la herramienta. También, tendrá como parámetro de entrada el ángulo que, en estos dos módulos de la *demo*, siempre se mantendrá constante respecto al valor de la orientación que se tenía en el anterior movimiento.

- A la hora de realizar un cambio de orientación, se utilizará el proceso *DesplazaDentroAng*. Para este proceso, el parámetro de entrada que se mantendrá es el de la profundidad a la que se encuentre la

herramienta introducida. Sin embargo, cambiará el parámetro de entrada de la orientación por el valor del nuevo ángulo que tenga.

Previamente, a la ejecución de este proceso, *DesplazaDentroAng*, se indica el valor de la variable 'A' que denotará en qué plano se realizará el cambio de orientación.

Finalizados los movimientos de demostración, se llevará la herramienta a su soporte respetando la superficie de los modelos 3D de la estación. Esto será gracias al *robtarget salir* generado en el módulo *Salir_Pelvitainer* y el módulo para dejar la herramienta que se sostenga.

Se muestra una parte del código para ver la lógica que siguen estos módulos:

```
A:=2;
dist1:=y;
TPErase;
TPWrite "Ahora se moverá -15 grados en el plano YZ";
DesplazaDentroAng -15,120,Ag_{2};
t:=t+1;
WaitTime 2;

TPErase;
y:=120;
TPWrite "Se introducirá 40 mm de profundidad más,";
TPWrite "estará introducida a 160 mm de profundidad de nuevo";
EntradaConAngulo -15,40,Ag_{2};
dist1:=y;
WaitTime 2;
```

Ilustración 77 - Fragmento del código del módulo DemoBlanca

4.3.4.3 Módulo *Circunferencia*

Este proceso no tiene argumentos de entrada. Es el proceso encargado de trazar una circunferencia con el final de la herramienta blanca. Cogemos la herramienta blanca, con el módulo dedicado a esta acción, y, con un *MoveJ*, desplazamos el final de la herramienta hacia el agujero 7.

Posteriormente, utilizamos el proceso *PerpendicularMesa* para que entre, perpendicularmente al plano de la mesa de trabajo, 84 milímetros. A partir de esa posición, aplicamos un desplazamiento por ángulos con el módulo *DesplazaDentroAng* para que se desplace 15 grados en el plano YZ.

Puesta la herramienta en esa posición, se ejecutará el bucle clave de este proceso:

```

FOR a FROM 0 TO 360 STEP 2 DO

    pose4.trans:=Ag_{7}.trans;
    pose4.rot:=OrientZYX(a,15,0)*Ag_{6}.rot;
    pose5:=[[0,0,84],[1,0,0,0]];
    pose6:=PoseMult(pose4,pose5);
    aux1.trans:=pose6.trans;
    aux1.rot:=pose6.rot;
    confL\Off;
    confJ\Off;
    MoveL aux1,v100,fine,Herra_Blanca\WObj:=Pelv2;

ENDFOR

```

Ilustración 78 - Bucle encargado de trazar la circunferencia

Este bucle será el encargado de aplicar una rotación a la herramienta, a partir de la posición y orientación que quedó después de utilizarse el módulo *DesplazaDentroAng* en esta opción.

Tenemos un *pose4* con la posición del *robtarger* del orificio 7. La orientación del *pose4* es la del *robtarger* del orificio 6 (perpendicular a la mesa) con un *OrientZYX* que le aplicará una rotación al *robtarger* en Y para mantener la orientación del módulo, *DesplazaDentroAng*, que se había usado recientemente. También, con el mismo *OrientZYX* se le aplicará una rotación en Z con un ángulo que irá aumentando su valor de dos en dos hasta llegar a 360, esta rotación en Z será clave para que el final de la herramienta trace la circunferencia.

El *pose5* solo definirá un desplazamiento en Z, cuya distancia es a la que se introdujo la herramienta, 84 milímetros. Realizamos el producto entre el *pose4* y *pose5*, y obtenemos un punto, cuya orientación será la del *pose4*, pero desplazado 84 milímetros en el eje Z de este *pose4*. El valor de la rotación en Z del *pose4*, que se le aplicará al resultado del producto, irá aumentando en cada iteración del bucle y eso tendrá como consecuencia el giro de la herramienta.

4.3.4.4 Módulo *Barrido*

También sin argumentos de entrada, es el proceso encargado de trazar la figura mostrada en la Ilustración 151. Cogemos también la herramienta blanca y, con un *MoveJ*, desplazamos el final de la herramienta hacia el orificio 1.

Después, utilizamos el proceso *PerpendicularMesa* para que entre perpendicularmente al plano de la mesa de trabajo 100 milímetros. A partir de esa posición, aplicamos el bucle clave de este proceso:

```

FOR salto FROM 0 TO 315 STEP 45 DO

  FOR R_ FROM 0 TO angu STEP signo DO

    pose1.trans:=Ag_{1}.trans;
    pose1.rot:=OrientZYX(salto,0,R_)*Ag_{6}.rot;
    pose2:=[[0,0,dist],[1,0,0,0]];
    pose3:=PoseMult(pose1,pose2);
    aux.trans:=pose3.trans;
    aux.rot:=pose3.rot;
    MoveL aux,v1000,fine,Herra_Blanca\WObj:=Pelv2;
  ENDFOR

  FOR R_ FROM angu TO 0 STEP -signo DO

    pose1.trans:=Ag_{1}.trans;
    pose1.rot:=OrientZYX(salto,0,R_)*Ag_{6}.rot;
    pose2:=[[0,0,dist],[1,0,0,0]];
    pose3:=PoseMult(pose1,pose2);
    aux.trans:=pose3.trans;
    aux.rot:=pose3.rot;
    MoveL aux,v1000,fine,Herra_Blanca\WObj:=Pelv2;
  ENDFOR

  MoveL aux3,v100,fine,Herra_Blanca\WObj:=Pelv2;
ENDFOR

```

Ilustración 79 - Bucle encargado de realizar el barrido

Observamos que tenemos un bucle *FOR* y, dentro de él, otros dos. El bucle que engloba a los otros dos es el encargado de que, en cada iteración de éste, se trace un “aspa” de la figura de la Ilustración 151. El trazo del “aspa” es cada 45 grados, esto es consecuencia de que el bucle *FOR* tenga como salto 45.

Luego, en cuanto a los dos bucles *FOR*, dedicados a la variable numérica ‘*R_*’, el que tenga un salto positivo se dedicará a realizar el recorrido de ida del “aspa” que toque trazar con el final de la herramienta. Por recorrido de ida se quiere decir que, el final de la herramienta irá desde el centro de la figura de la Ilustración 151 hasta el extremo del “aspa” que esté trazando. El otro bucle, que tendrá un salto negativo, realizará el recorrido de vuelta del “aspa” que le toque. El recorrido de vuelta será como el de ida, pero al revés.

En el bucle dedicado a realizar el trazo de ida de cada “aspa”, se le asignará a *pose1* la posición del orificio 1. Mientras que la orientación que tendrá el *pose1*, será la orientación del *robtarget* del orificio 6 (perpendicular a la mesa) a la que se le aplicará un *OrientZYX*, con una rotación en el eje X que irá aumentando en cada iteración del bucle terminando por trazar el “aspa”.

Cabe mencionar que el *OrientZYX*, que se le aplica a la orientación del *pose1*, también le aplicará una rotación en el eje Z que dependerá de qué “aspa” se estará trazando. Este parámetro, que marca la orientación en Z, está regido, como hemos visto por el bucle *FOR* que engloba a los otros dos.

Pose2 solo definirá un desplazamiento en Z, cuya distancia la definirá el parámetro *dist*. Entonces, cuando se haga el producto de *pose1* con *pose2*, obtendremos un punto, con la orientación del punto del orificio 6 a la que se le aplicará una rotación en el eje X y Z, y, por último, un desplazamiento en el eje Z resultante.

4.3.5 Módulos encargados de que no choque la herramienta con el *Pelvitainer*

4.3.5.1 Módulo *Hacia_Orificio*

A la hora de seleccionar, correctamente, el orificio sobre el que se va a trabajar se ejecutará este proceso de forma automática. Se encargará de que la herramienta, sujeta por el brazo robótico en una posición base, se dirija al orificio sin que choque con el *Pelvitainer*.

Especialmente se utilizan las funciones *MoveJ* y *MoveAbsJ* con *offsets* que dependerán del orificio al que se vaya a aproximar el final de la herramienta. Los diferentes *offsets* que se le aplique a cada orificio, estarán condicionados por las funciones *if* aplicadas a las variables numéricas que están asociadas al tipo de barra y/o número del orificio. Este proceso no tiene argumentos de entrada.

4.3.5.2 Módulo *Orificio_a_base*

Se ejecutará automáticamente cuando la herramienta vaya a dejar el *robtaret* sobre el que esté trabajando, o bien para cambiar de orificio, o bien para cambiar de herramienta. Este proceso asegurará que la herramienta no entre en contacto con el *Pelvitainer*.

Principalmente, se hace uso de las funciones *MoveJ* y *MoveAbsJ* con diferentes parámetros de *offset* que dependen del orificio del que provenga la herramienta.

También, como el proceso anterior, los diferentes *offsets* que se le aplique a cada orificio, dependerán de las funciones *if* asociadas a las variables numéricas de orificio y/o barra. Igual que el anterior proceso, éste no tiene argumentos de entrada.

4.3.5.3 Módulo *Salir_Pelvitainer*

```
LOCAL PROC Salir_pelvitainer(num dist,robtaret punto)
```

La finalidad de este módulo es crear un *robtaret* que ayude a salir a la herramienta del *Pelvitainer* sin que entre en contacto con él.

Tras varios experimentos, se ha llegado a la conclusión de que la mejor manera de sacar la herramienta es con la orientación del eje Z del *robtarget* sobre el que se esté trabajando.

Con lo cual, como argumentos de entrada del proceso, necesitaremos el *robtarget* del orificio en el que estemos y la variable numérica que describe la distancia a la que se introdujo por primera vez la herramienta.

Esta función será llamada cuando vaya a entrar la herramienta al orificio, por primera vez mediante cualquiera de las 3 opciones: perpendicular al orificio, a la mesa o con ángulo. El código de esta función es el siguiente:

```
ConFL\off;
pose7.rot:=punto.rot;
pose7.trans:=punto.trans;
pose2:=[[0,0,dist],[1,0,0,0]];
pose3:=PoseMult(pose7,pose2);
salir.rot:=pose3.rot;
salir.trans:=pose3.trans;
```

Ilustración 80 - Código del módulo Salir_Pelvitainer

Creamos el *robtarget salir* a partir de aplicarle al *robtarget* del orificio un desplazamiento positivo en su eje Z de una distancia que definirá la variable numérica de entrada.

Entonces, cuando se ejecute el módulo, la herramienta pasará a estar introducido a la distancia a la que estaba la primera vez que entró y con la orientación del eje Z del *robtarget* del orificio en el que se encuentra. Así, cuando salga del *Pelvitainer* seguirá la dirección del eje Z del *robtarget*.

4.3.6 Módulos para tomar y dejar la herramienta

4.3.6.1 Módulos *TomarHerraBlanca* y *TomarHerraNegra*

Estas dos funciones se encargarán de que la pinza del brazo robótico agarre, dependiendo del módulo que sea, o bien la herramienta negra, o bien a herramienta blanca de su respectivo soporte. En estos movimientos se tendrá cuidado de no entrar en contacto con la superficie del soporte de la herramienta que se vaya a coger. Básicamente se hace uso de las funciones *MoveJ* y *MoveL* para realizar esta tarea y se les aplica *offsets* para cuidar los modelos 3D de la estación.

Igualmente, la función *MoveAbsJ* se usa para que el brazo robótico al principio y al final de este proceso se encuentre en una posición segura y preparado para ejecutar los siguientes módulos que se requieran.

Como en estas dos funciones el robot tiene que agarrar una herramienta, se hace uso de los procesos *AbrirPinza* y *CerrarPinza*. No ha sido necesario implementar argumentos de entrada a este proceso.

4.3.6.2 Módulos *DejarHerraBlanca* y *DejarHerraNegra*

La estructura es bastante similar a la de los procesos *TomarHerraBlanca* y *TomarHerraNegra*. La diferencia es que se dejará, o bien la herramienta negra, o bien la herramienta blanca en su respectivo soporte. Se realizará con el mayor cuidado posible de que la herramienta no choque con su soporte.

Igualmente se hace uso de los procesos *AbrirPinza* y *CerrarPinza*. Las funciones *MoveJ*, *MoveL* y *MoveAbsJ* tendrán también un papel crucial para lograr esta tarea efectivamente. Se hará uso de *offsets* también en estas funciones para depositar las herramientas. Igual que el anterior, no tiene argumentos de entrada.

4.3.6.3 Módulos *AbrirPinza* y *CerrarPinza*

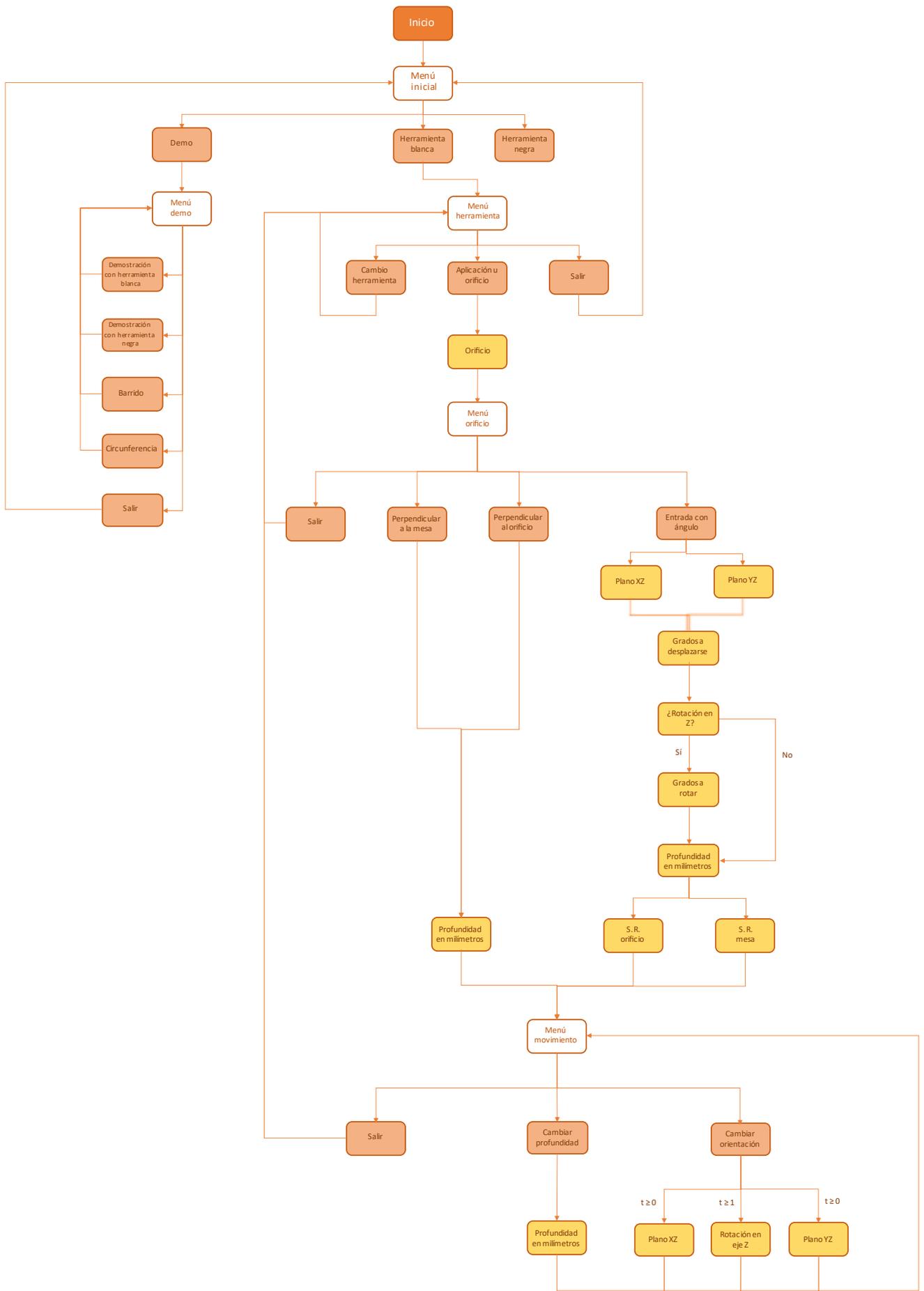
Como su nombre indica, estos procesos se encargan de abrir o cerrar la pinza del brazo robótico que se encuentra en la articulación 6. Básicamente se usarán para dejar o coger la herramienta con la que se vaya a trabajar. Ambos procesos no poseen argumentos de entrada

5. Funcionamiento de la aplicación desarrollada

Inicialmente, se mostrará el diagrama de la aplicación que se ha desarrollado. El color de los recuadros tendrá un significado en particular:

- El recuadro de color naranja marca el comienzo de la aplicación.
- Los recuadros que tienen un color rosáceo serán las opciones que podrá elegir el usuario durante la simulación.
- Los recuadros que tienen un color amarillento será los parámetros que el usuario deberá introducir dependiendo de la opción marcada.
- Los recuadros de fondo blanco nos indica que es un menú al que se puede acceder seleccionando determinadas opciones, no será algo que nos muestre el programa.

Durante este apartado iremos fraccionando este diagrama para poder explicarlo con más detalle y explicaremos lo que hace cada fragmento durante la simulación.



Normalmente, los diferentes menús que se irán mostrando por pantalla, se borrarán tras haber realizado una selección en ellos. Se borrarán con la finalidad de no acumular texto en la ventana de operador.

5.1 Funcionamiento del menú inicial

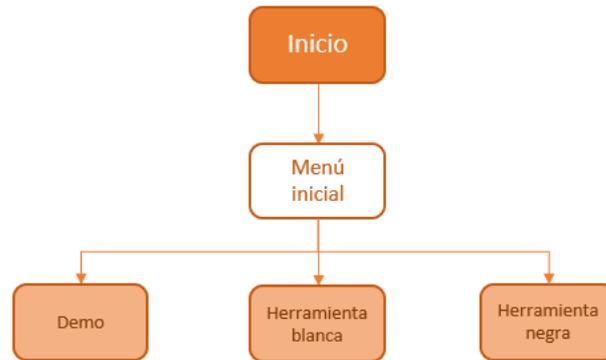


Ilustración 81 - Menú inicial en el diagrama

Al iniciar la simulación, lo primero que se le preguntará al usuario será lo siguiente:

```
Hola, este es el menú principal.  
Herramienta blanca: Se utiliza con un agarre superior, solo podrá acceder  
a los orificios 1,2 y 7, pero permite realizar más movimientos en estos.  
Herramienta negra: Se utiliza con un agarre lateral,  
permite acceder a todos los orificios.  
Qué herramienta deseas usar, ¿la herramienta blanca o negra?  
También puedes usar la demo si lo deseas.
```



Ilustración 82 - Menú principal

El programa inicialmente dará las opciones al usuario de coger la herramienta negra, blanca o utilizar la *demo*. En el menú principal se informará al usuario de que con la herramienta blanca se utiliza un agarre superior, sólo podrá acceder a los orificios 1, 2 y 7, y permitirá realizar más movimientos dentro de ellos. También se informará que, con la herramienta negra, la cual se utiliza con un agarre lateral, se podrá acceder a todos los orificios, pero se podrá realizar menos movimientos dentro de él.

5.1.1 Demo

La opción de la demo será la última que comentemos de las opciones que aparecen en la Ilustración 82. El motivo de esto es porque se comprenderán

mejor los movimientos que se hagan en la *demo* tras haber conocido la variedad de opciones que ofrece el programa en la aplicación.

5.1.2 Herramienta negra o blanca

Si seleccionamos 'H. Blanca' o 'H. Negra' entraremos en un entorno en el que el usuario podrá manejar los movimientos que desee a su gusto. A este entorno lo denominaremos como 'Aplicación'.

Depende de si seleccionamos las opciones de herramienta blanca o herramienta negra cogerá o bien una herramienta blanca o bien una herramienta negra. Esto dará lugar al siguiente menú:

5.2 Funcionamiento del menú herramienta

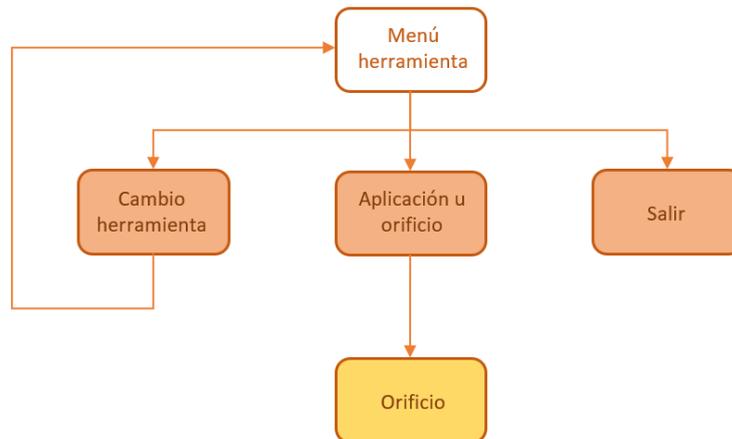


Ilustración 83 - Menú herramienta en el diagrama

Una vez el brazo robótico haya cogido una de dichas herramientas, aparecerán las siguientes opciones por pantalla:

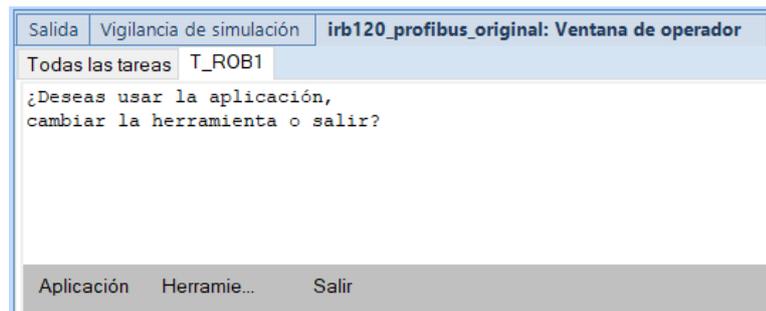


Ilustración 84 - Menú herramienta

5.2.1 Salir en el menú herramienta

Si seleccionamos salir, el robot dejará la herramienta, que haya cogido, en su respectivo soporte y volveremos al menú inicial.

5.2.2 Herramienta

Si seleccionamos esta opción, el robot cambiará la herramienta del color que tenga por otra del color contrario. Es decir, si al principio de la simulación hemos cogido una herramienta de color negro y seleccionamos la opción de 'Herramienta', el brazo robótico dejará la herramienta de color negro y cogerá la herramienta de color blanco o viceversa.

Además, durante la acción del cambio de la herramienta, saldrá por pantalla el siguiente mensaje:

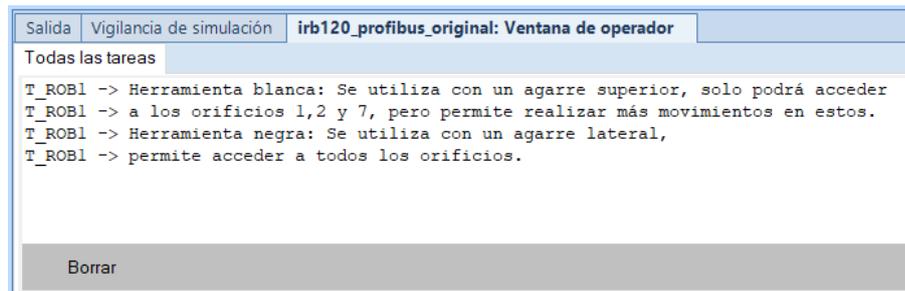


Ilustración 85 - Mensaje durante el cambio de herramienta

Esta información es similar a la que aparece al comenzar la simulación en el menú de la Ilustración 82, solo que se han omitido algunas frases. La finalidad de esto es recordar al usuario las cualidades que tiene la herramienta de cada color.

Una vez realizada esta opción y el robot tenga su nueva herramienta, saldrán las siguientes opciones por pantalla:

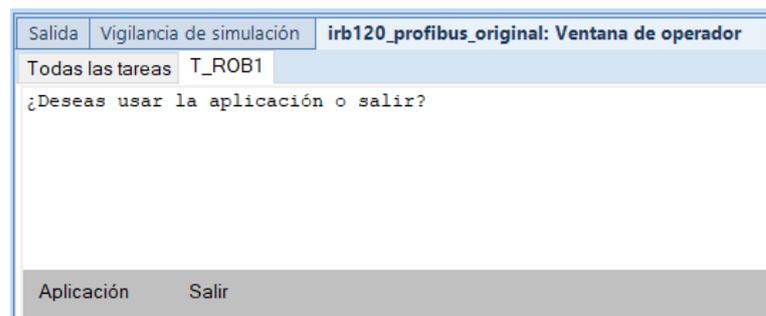


Ilustración 86 - Menú después de seleccionar cambio de herramienta.

Se trata del mismo menú que la Ilustración 84, solo que la opción de la 'Herramienta' está omitida, dado que se ha considerado redundante volver a preguntar una acción que ya se ha hecho. Sin embargo, cuando se realicen más acciones, posteriores a ésta, volverá a aparecer disponible la opción del cambio de herramienta. El efecto que causaría seleccionar cualquiera de las opciones de la Ilustración 86 ('Aplicación' o 'salir') sería el mismo que el de seleccionarlas en la Ilustración 84.

5.2.3 Aplicación

Una vez seleccionada esta opción, se nos mostrará lo siguiente por pantalla:

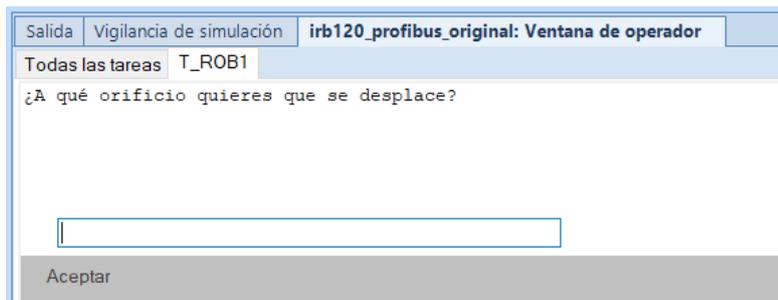


Ilustración 87 – Petición al usuario después de seleccionar 'Aplicación'

El programa nos preguntará a qué orificio queremos acceder con el final de la herramienta que tengamos. En caso de que estemos con la herramienta blanca e introduzcamos un número de orificio que no sea el 1, 2 o el 7 nos saldrá el siguiente mensaje de error:

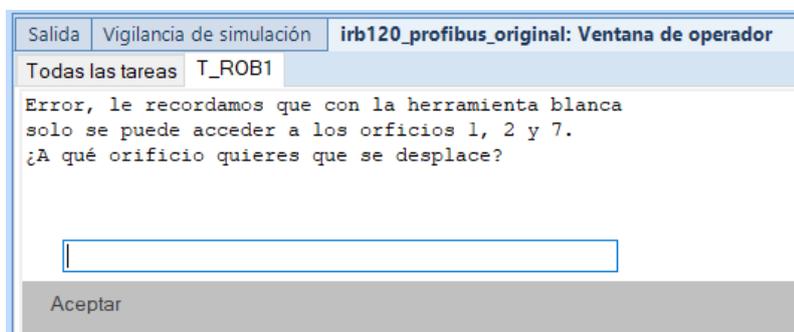


Ilustración 88 - Mensaje de error con herramienta blanca al seleccionar orificio

Hasta que no comandemos el orificio 1, 2 o 7 nos aparecerá dicho mensaje de error. En caso de que tengamos la herramienta negra e introduzcamos un número de orificio mayor que 9 o menor que 1, tendremos el siguiente mensaje de error:

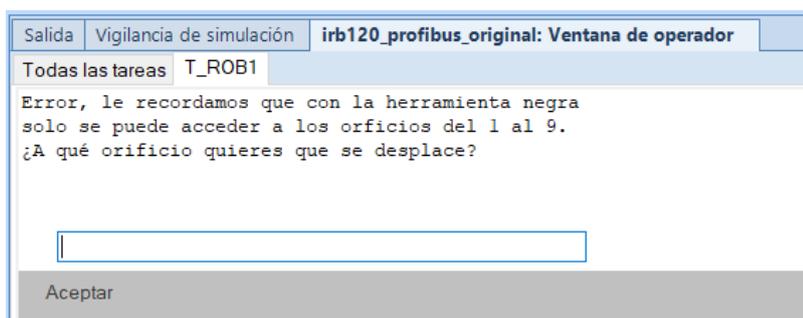


Ilustración 89- Mensaje de error con herramienta negra al seleccionar orificio

Este mensaje seguirá apareciendo hasta que introduzcamos un número de orificio entre 1 y 9. Una vez introducido en el cuadro de diálogo el orificio adecuado, estaremos en el siguiente menú, el menú orificio.

5.3 Funcionamiento del menú orificio

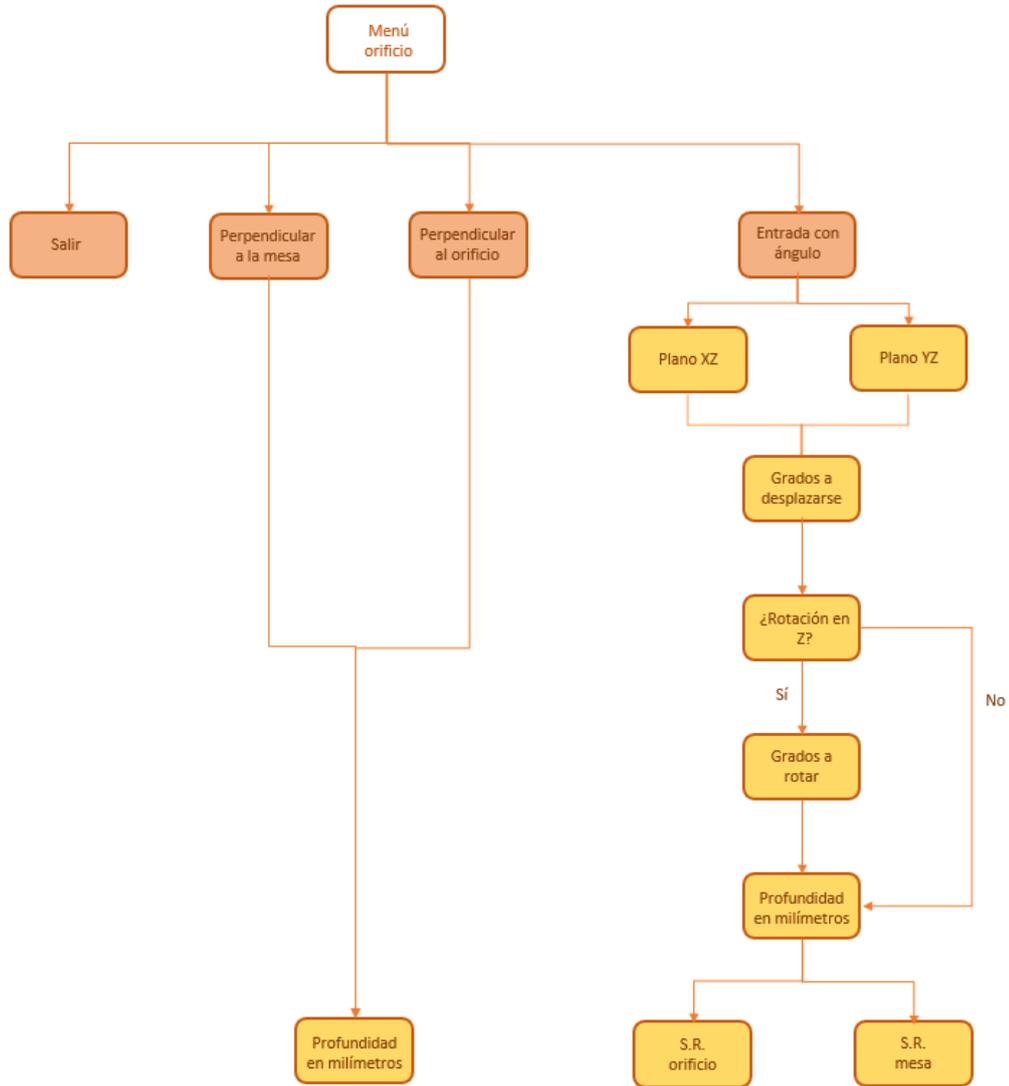


Ilustración 90 - Menú orificio en el diagrama

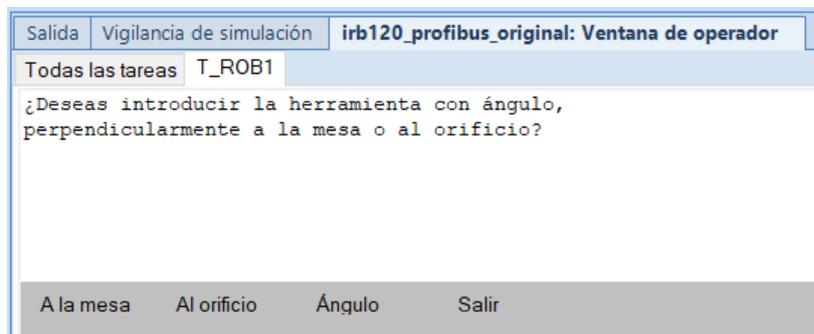


Ilustración 91 - Menú inicial para comandar movimientos dentro del Pelvitainer

El programa nos preguntará si queremos introducir la herramienta perpendicularmente al orificio, al plano de la mesa, con ángulo o si queremos salir.

5.3.1 Opción perpendicular a la mesa

Si seleccionamos esta opción la herramienta se introducirá con la orientación del eje Z del sistema de referencia de la mesa. Si suponemos que tanto como el robot, el *Pelvitainer* y las herramientas con su respectivo soporte están apoyados en una mesa, el sistema de referencia de esta mesa y su orientación sería el siguiente:

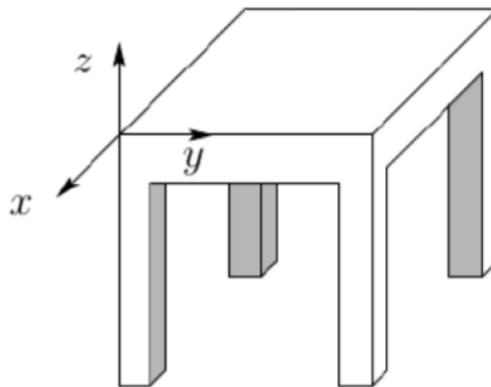


Ilustración 92- Ejemplo de orientación del eje z de la mesa

Entonces, la herramienta se introducirá siguiendo la orientación negativa del eje Z mostrado en la Ilustración 92. Con la herramienta negra, esta opción no estará disponible para el orificio 3 pero sí que lo estará para el resto de los orificios. Al haber seleccionado esa opción, se nos preguntará lo siguiente:

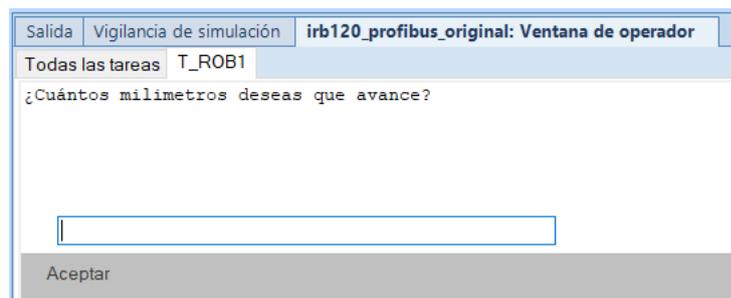


Ilustración 93 - Petición al usuario de distancia para introducir la herramienta.

El valor que introduzcamos será siempre en milímetros. Esta distancia que introduzcamos será la cantidad de milímetros que entrará la herramienta en el orificio deseado del *Pelvitainer*. A continuación, se muestra un ejemplo de cómo quedaría la herramienta negra habiendo comandado que se introdujera 100 milímetros en esta opción en el orificio 1:

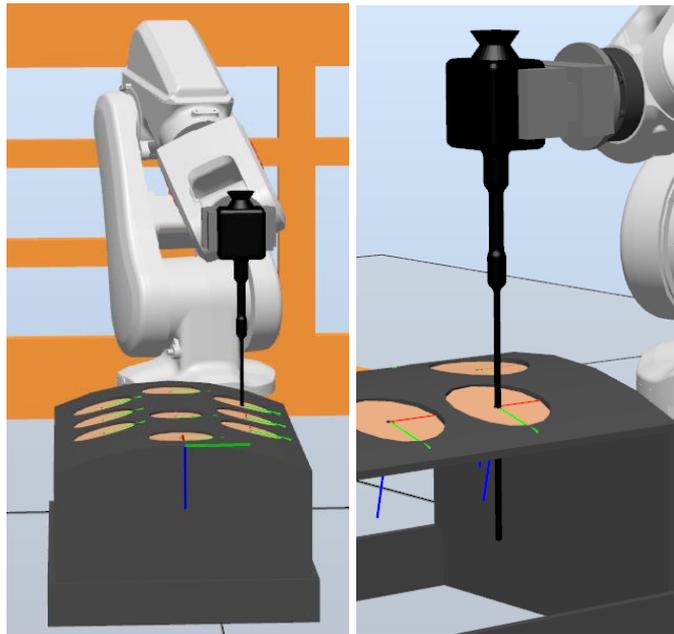


Ilustración 94 - Ejemplo de herramienta introducida perpendicular a la mesa

Observamos que se respeta la orientación del eje Z, y la herramienta está completamente perpendicular al plano sobre el que están apoyado los elementos 3D (robot, *Pelvitainer*, etc.) de la estación.

5.3.2 Opción perpendicular al orificio

Con una lógica parecida a la de la opción anterior, la herramienta se introducirá siguiendo la orientación del eje Z del *robtarget*. En otras palabras, la herramienta se introducirá perpendicularmente al plano XY del *robtarget*.

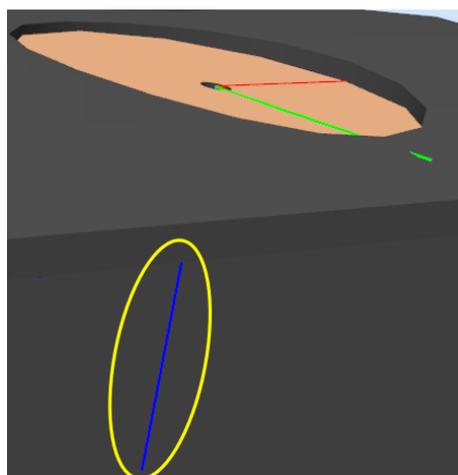


Ilustración 95 - Eje z del *robtarget* del orificio 1

Entonces, cuando el robot introduzca la herramienta perpendicularmente al orificio, se limitará a seguir la orientación del eje azul de la Ilustración 95. Una

vez se haya seleccionado esta opción, nos aparecerá la misma cuestión por pantalla que la de la Ilustración 93. Igual que en la Ilustración 93 las unidades que se darán al programa serán siempre en milímetros. A diferencia de la opción 'perpendicular a la mesa', esta opción sí que estará disponible para el orificio 3 y los demás con la herramienta negra.

A continuación, tenemos un ejemplo de esta opción para el orificio 7 con la herramienta blanca introducida una distancia de 70 milímetros:

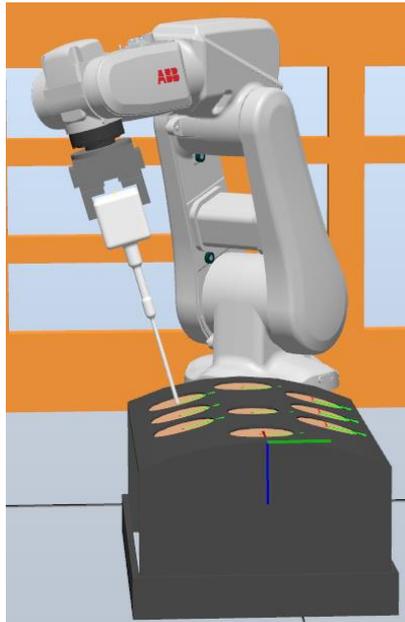


Ilustración 96 - Ejemplo de simulación de la opción perpendicular al orificio

Podemos ver que la herramienta está introducida perpendicularmente a la superficie del orificio. Además, si miramos desde otro ángulo este ejemplo, podemos observar que la herramienta blanca sigue el eje azul del *robtarger*:

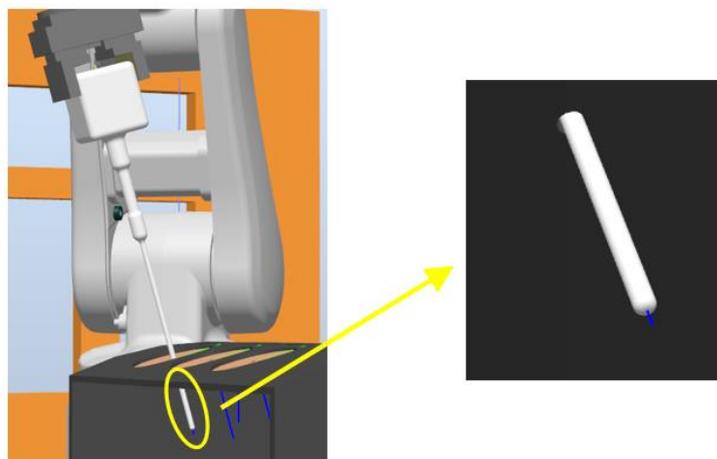


Ilustración 97 - Visualización del final de la herramienta en la opción perpendicular al orificio

Tras haber comentado las opciones de perpendicular a la mesa o al orificio, otra forma de comprender mejor la diferencia entre estas orientaciones es con la siguiente ilustración:

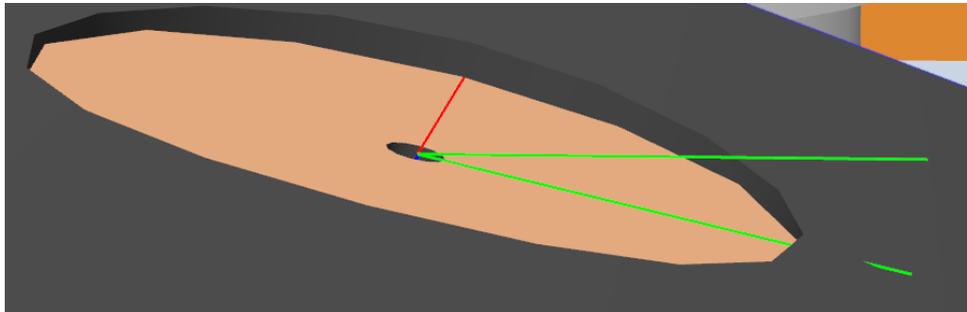


Ilustración 98 - Sistemas de referencia para perpendicular a la mesa y al orificio superpuestos

Observamos que hay 2 *robtargets* superpuestos, uno de ellos está con el plano XY paralelo al plano donde están apoyados los modelos 3D (la mesa) y el otro plano XY está paralelo al agujero de la superficie. Así que, si seleccionamos '*perpendicular a la mesa*', se trabajará con la orientación de los ejes de coordenadas del *robtarget* cuyo plano XY es paralelo a la superficie de la mesa. En cambio, si seleccionamos '*perpendicular al orificio*' trabajará con la orientación de los ejes de coordenadas del otro *robtarget*.

En ambas opciones de entrada, con la herramienta blanca solo podremos acceder a los orificios 1, 2 y 7. Además, en caso de tener la herramienta negra y seleccionar trabajar en el orificio 3, nos aparecerá el siguiente menú.

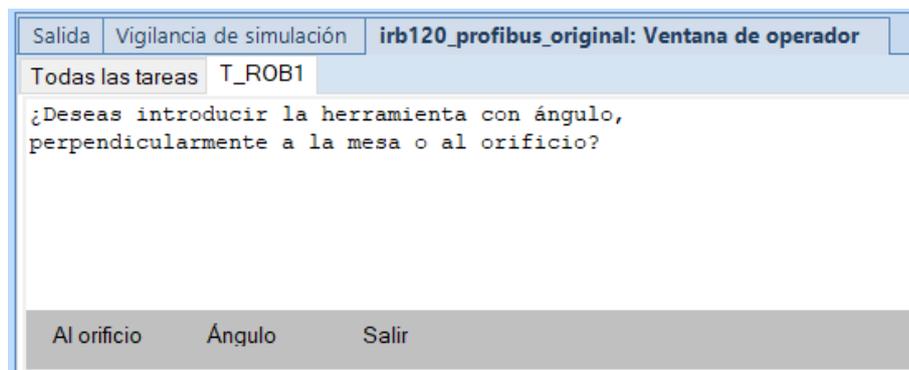


Ilustración 99 - Menú al dirigir la herramienta negra al orificio 3

Como podemos ver, se ha omitido la opción entrar perpendicularmente a la mesa. Esto se debe a que el robot alcanza una singularidad al introducir la herramienta más de 3 milímetros con esta orientación.

Estas dos opciones de entrada comentadas son exclusivas para los orificios 1, 2, 7, 8 y 9, para los orificios 4, 5 y 6 tendremos un menú diferente al de la Ilustración 91. En otras palabras, si nos pregunta el programa a qué orificio

deseamos llevar la herramienta y seleccionamos el orificio 4, 5 o 6 se nos muestra el siguiente menú:

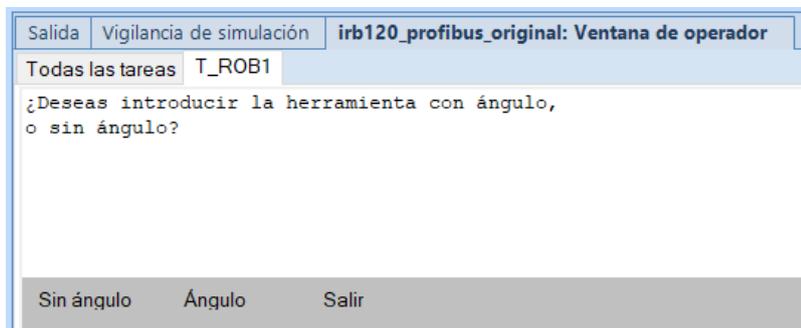


Ilustración 100 - Menú para los orificios 4, 5 y 6

Comparándolo con el menú de la Ilustración 91, en la Ilustración 100 no aparecen las opciones de perpendicular a la mesa o perpendicular al orificio. En su lugar, aparece la opción 'Sin ángulo', esto es debido a que usar una orientación perpendicular a la mesa o al orificio es la misma, porque los orificios 4, 5 y 6 no se encuentran con ángulo de inclinación a diferencia del resto.

Entonces, si seleccionamos la opción 'Sin ángulo', nos aparecerá la misma pregunta que la Ilustración 93, con lo cual, la distancia habrá que introducirla en milímetros. Si decimos que la herramienta se desplace al orificio 5, seleccionamos esta opción y le decimos que se introduzca 130 milímetros tendríamos el siguiente resultado:

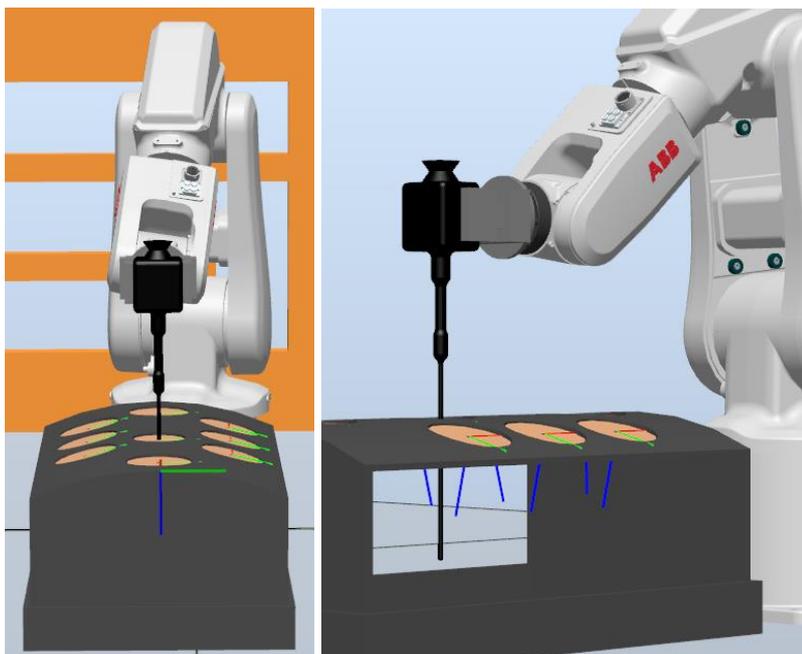


Ilustración 101 – Ejemplo para entrada de herramienta 'Sin ángulo'

Se puede observar que se introduce perpendicularmente a la mesa que es justo la misma orientación que el eje Z del *robtarger* del orificio.

El resto de las opciones que aparecen en la Ilustración 100 son idénticas a las de la Ilustración 91, y realizan la misma función.

A mayores, en ambas opciones (perpendicular a la mesa y al orificio), se limitará al usuario la distancia a la que se puede introducir la herramienta. Es decir, si el usuario introduce una distancia más grande de lo que es capaz de introducir el robot con la herramienta, el programa lo rechazará.

Los límites de profundidad, para cuando introduzcamos por primera vez la herramienta, perpendicularmente al orificio o a la mesa de trabajo, se encuentran en la Tabla 11 y Tabla 12 respectivamente. Con lo cual, si introducimos una profundidad más grande de la permitida, el programa no lo ejecutará y nos informará de cuál es la máxima distancia posible.

Por ejemplo, tenemos la herramienta negra y la llevamos al orificio 4, si decimos que se introduzca 170 milímetros, el programa nos dirá los siguiente:

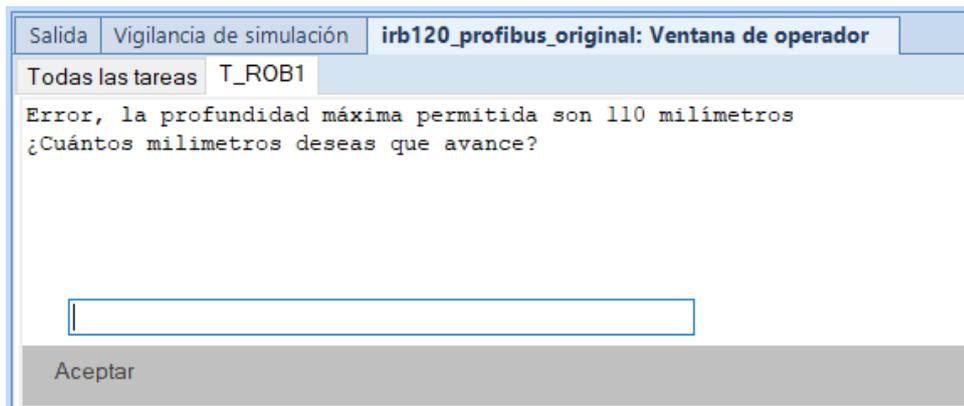


Ilustración 102 - Mensaje por pantalla al introducir la herramienta negra 170 milímetros en el orificio 4

El programa no efectuará la orden de introducir la herramienta 170 milímetros. Si en el cuadro de diálogo introducimos un valor mayor de 110 milímetros el programa nos sacará el mensaje por pantalla de la Ilustración 102. Este mensaje por pantalla saldrá cada vez que introduzcamos un valor mayor al máximo, se podría dar un número ilimitado de veces. Lógicamente, si introducimos un valor menor a 111 milímetros, el programa lo permitirá y se podrá avanzar en la aplicación sin problemas.

Esta situación se producirá solamente cuando el robot introduzca la herramienta perpendicularmente a la mesa o al orificio por primera vez. Es decir, si después de introducir la herramienta, realizamos un desplazamiento

por ángulos y luego cambiamos la profundidad de la herramienta, el programa no nos avisará de cuáles son los valores límites.

Esto es debido a que, las situaciones son infinitas, porque para cada milímetro de profundidad hay un amplio de número grados que le podamos aplicar. Con lo cual, hacer un estudio sobre la profundidad para cada milímetro, con cada herramienta, en cada orificio y para cada ángulo en diferentes planos llevaría una cantidad de tiempo incalculable.

5.3.3 Opción de entrada al orificio con ángulo

Esta opción, a diferencia que las anteriores, permite hacer que la herramienta entre al orificio con una orientación determinada por el usuario. Dentro de esta opción la aplicación nos pedirá una serie de parámetros, el primero que nos pedirá será el siguiente:

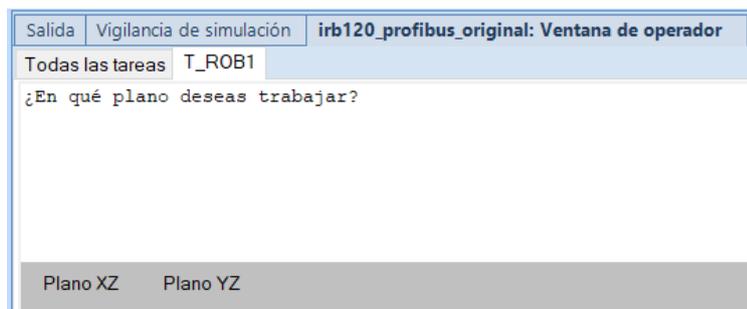


Ilustración 103 - Primera elección de la opción con ángulo

Antes de comentar los efectos del menú de la Ilustración 103 de la aplicación, explicaremos qué es un pivoteo. En un pivoteo, el final de la herramienta siempre estará situado en el *robtarger* del orificio en el que se trabaje. En otras palabras, el final de la herramienta estará en un punto fijo. La acción en la que la herramienta se incline manteniendo ese punto fijo se llamará pivoteo. Más adelante veremos ejemplos de pivoteo.

Siguiendo con el menú de la ilustración 103, lo primero que habrá que hacer es seleccionar el plano en el que se va a trabajar, o bien en el plano XZ o bien en el plano YZ. Para visualizar mejor estos planos en 3D tenemos la siguiente ilustración:

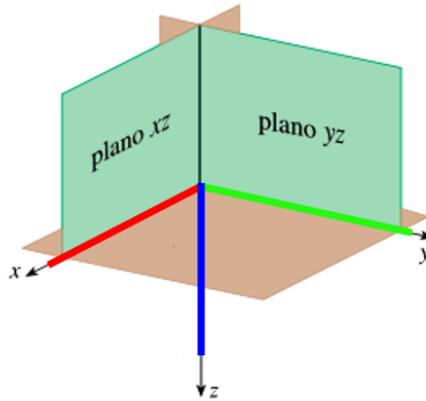


Ilustración 104 - Plano XZ e YZ sobre los ejes de un rotarget

Observamos que los ejes de coordenadas de la ilustración tienen los mismos colores que el de un *rotarget*, esto nos ayudará a visualizar mejor los planos. Entonces cuando seleccionemos el 'Plano XZ' en el menú, la herramienta realizará un pivoteo por donde está señalado el 'plano xz' en la Ilustración 104. Lo mismo ocurriría si seleccionamos el 'Plano YZ' en el menú, realizará un pivoteo por el 'plano yz'.

Una vez hayamos seleccionado uno de esos dos planos, el programa nos preguntará con cuántos grados queremos que haga el pivoteo en el plano seleccionado:

¿Cuántos grados quieres que se desplace?

Aceptar

Ilustración 105 - Petición de grados para el plano seleccionado

Para comprender mejor el pivoteo en los planos XZ e YZ, se muestran dos ejemplos en los que únicamente se ha realizado un pivoteo en cada plano:

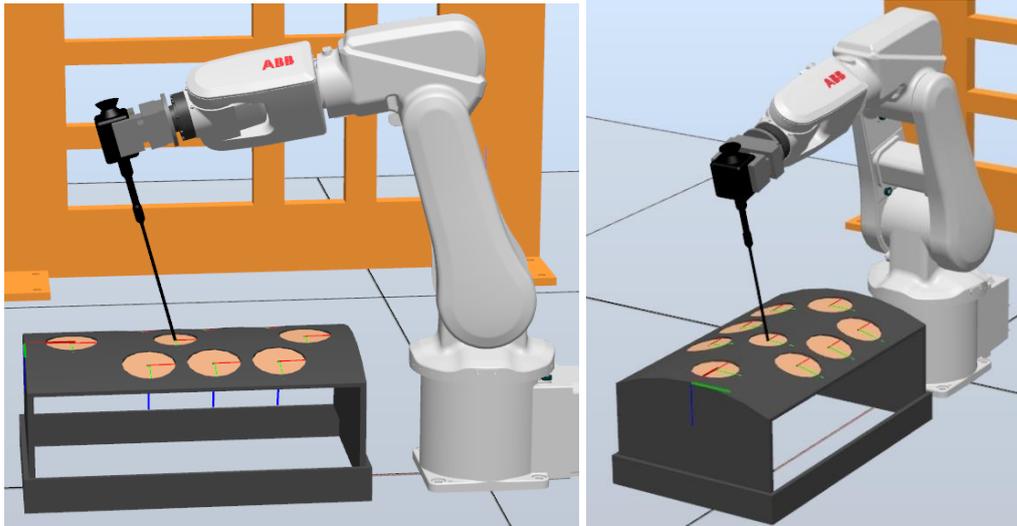


Ilustración 106 - Ejemplo pivoteo en el plano XZ

En la Ilustración 106, estando el final de la herramienta negra en el orificio 5, se ha realizado un pivoteo en el plano XZ de 15 grados. Se puede observar que, a pesar de la inclinación, el final de la herramienta sigue en el orificio 5. Tenemos otro ejemplo similar al anterior, pero en el plano YZ:

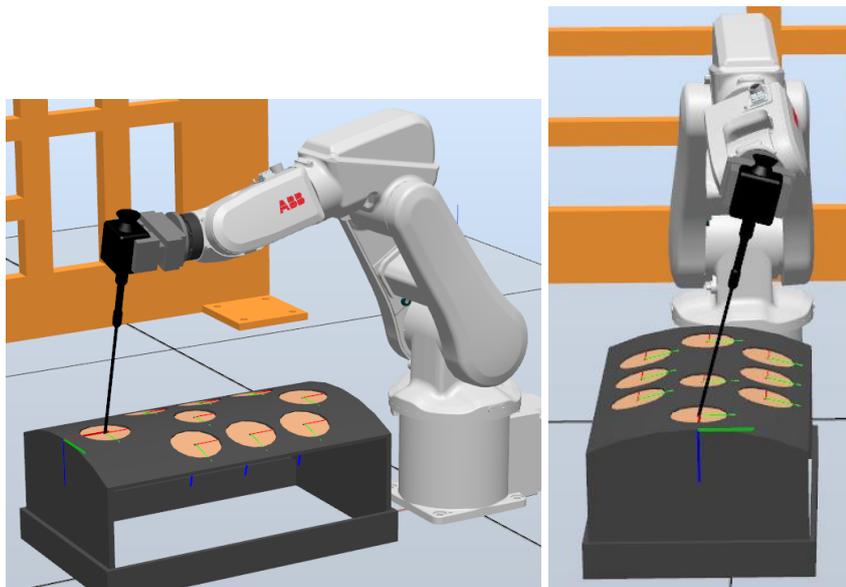


Ilustración 107 - Ejemplo pivoteo en el plano YZ

En la Ilustración 107, con el final de la herramienta en el orificio 6 se ha realizado un pivoteo de 15 grados en el plano YZ. Igual que en la Ilustración 106, el final de la herramienta se mantiene en el orificio señalado a pesar de la inclinación que tiene.

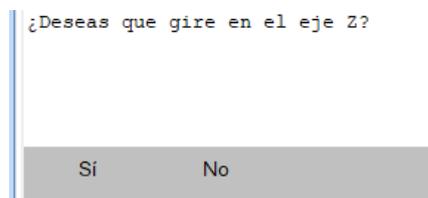
Además, en el cuadro de diálogo de la Ilustración 105 se pueden introducir valores negativos. En caso de que introduzcamos un valor negativo para el

plano XZ, la herramienta pivotará en el sentido opuesto al de la Ilustración 106. Mientras que si tenemos un valor negativo en el cuadro de diálogo de la Ilustración 105 para el plano YZ, igual que con el plano XZ, la herramienta pivotará en el sentido contrario al de la Ilustración 107.

Como otra forma de entender las diferencias entre el plano YZ y el plano XZ, si miramos al *Pelvitainer* desde la posición en la que está la articulación 1 del brazo robótico, moverse en el plano YZ sería realizar un pivoteo hacia la derecha o la izquierda. Mientras que en el plano XZ sería realizar un pivoteo hacia adelante o hacia atrás.

En concreto, si seguimos desde el punto de vista de la articulación 1 del brazo robótico, para valores positivos del plano XZ pivotará hacia adelante y de forma análoga, para valores negativos pivotará hacia atrás. Para el plano YZ, cuando introduzcamos un valor positivo, pivotará hacia la izquierda y, por lo tanto, si damos un ángulo negativo para este plano, pivotará hacia la derecha.

Tras haber introducido los grados deseados en el plano XZ o YZ, como siguiente pregunta, dentro de la opción de entrada con ángulo, tenemos la siguiente cuestión:

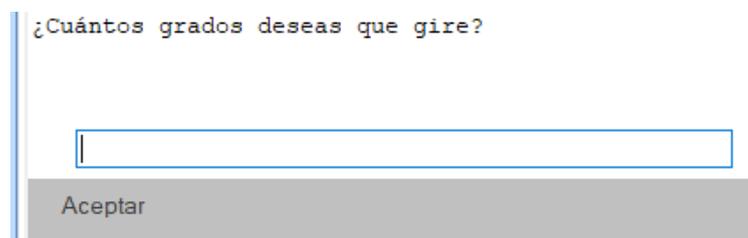


¿Deseas que gire en el eje Z?

Sí No

Ilustración 108 - Pregunta al usuario si quiere realizar rotación

Cuando la herramienta haya realizado un pivoteo en el plano XZ o YZ, podremos realizar una rotación si lo deseamos. Si decimos que no, pasaremos a la pregunta de la Ilustración 113. Y si decimos que sí, se nos preguntará de nuevo, como en la Ilustración 105, cuantos grados deberá rotarse de la siguiente forma:



¿Cuántos grados deseas que gire?

Aceptar

Ilustración 109 - Pregunta de con cuántos grados realizar la rotación

Para comprender mejor cómo funciona la rotación o giro en el eje z, tenemos la siguiente ilustración:

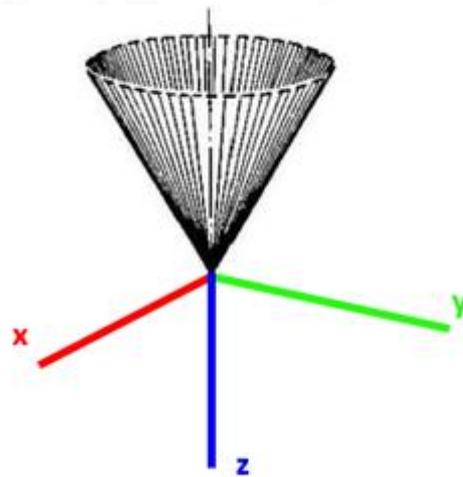


Ilustración 110 - Ejemplo de rotación completa

Lo primero a tener en cuenta en esta imagen, es que los ejes de coordenadas son de un *robotarget*. Además, se han puesto del mismo color los ejes de coordenadas de la imagen que los del *robotarget* para poder entenderlo mejor.

Si para un pivoteo como los de las Ilustración 106 o Ilustración 107, realizamos una rotación de 360 grados, la herramienta generaría una superficie de revolución. La superficie de revolución sería similar a un cono invertido como el de la Ilustración 110, porque el final de la herramienta se mantendría en el origen de coordenadas y el resto de la herramienta rotaría sobre este punto fijo.

Para ver los efectos de la rotación en esta opción de entrada con ángulo, mostramos el siguiente ejemplo:

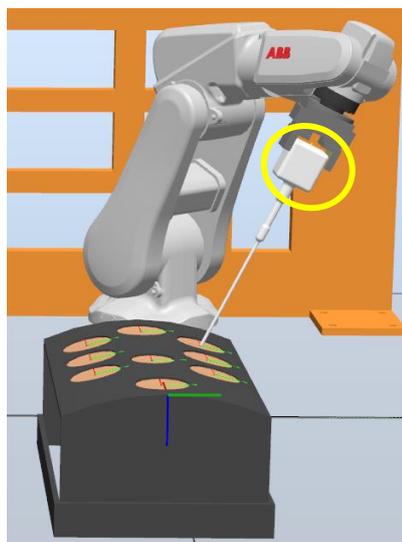


Ilustración 111 - Posición inicial con herramienta inclinada 12 grados

Partimos de la posición inicial de la Ilustración 111, en la que se realiza un pivoteo de 12 grados, en el plano YZ, en el orificio 1. La cara del prisma, de la 'cabeza' de la herramienta, que está marcada, nos ayudará a notar la rotación que realizaremos a continuación.

En la posición de la Ilustración 111, realizamos una rotación de -40 grados. La posición de la herramienta será la siguiente:

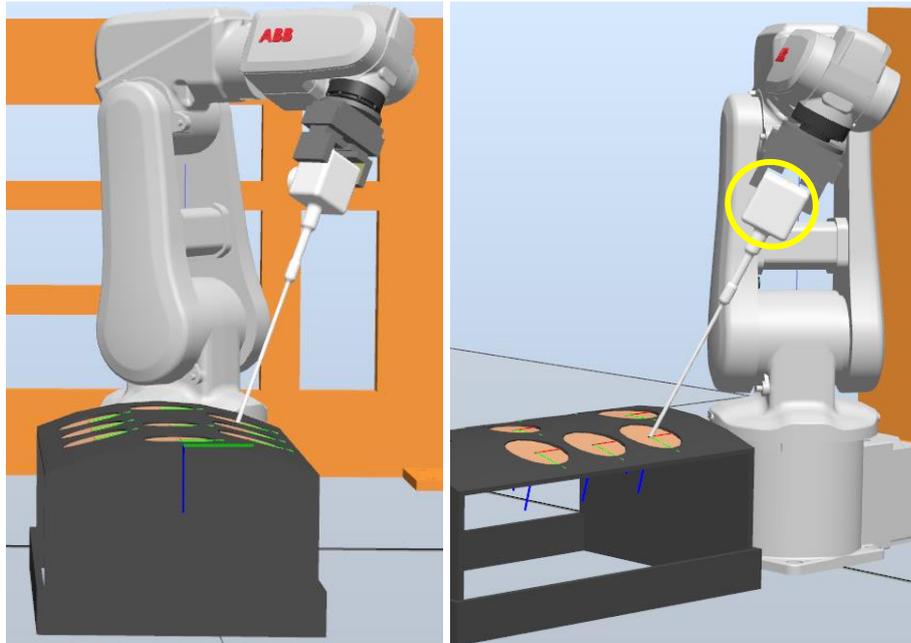


Ilustración 112 - Rotación de -40 grados aplicada a la ilustración 111

En la ilustración de la derecha vemos cómo la cara señalada ya no tiene la misma orientación que en la Ilustración 111. Si nos fijamos en el *Pelvitainer*, en las dos ilustraciones que están rodeadas, se puede ver que las dos capturas se han realizado desde diferentes ángulos y la orientación de la cara redondeada ha cambiado.

Podemos observar también que, aunque sea rotación, la acción se sigue haciendo mediante pivoteo. El final de la herramienta se sigue manteniendo en el origen de coordenadas del *robtarget* y a partir de ahí realiza la rotación.

Al igual que el cuadro de diálogo de la Ilustración 105 en el cuadro de diálogo de la Ilustración 109, se admitirán valores negativos. Al ser negativo el valor, realizará la rotación en un sentido opuesto al del valor positivo. En caso de que sea un valor positivo la rotación se hará en un sentido horario, mientras que cuando sea un valor negativo se realizará en un sentido contrario a las agujas del reloj.

Después de la cuestión de la rotación, se nos preguntará cuantos milímetros se deberá introducir la herramienta, de la misma forma que se hizo en la Ilustración 93. Por último, tendremos la siguiente pregunta:

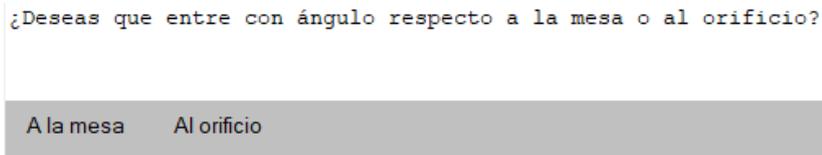


Ilustración 113 - Última petición al usuario de la opción con ángulo

El programa querrá saber si toda esta información introducida lo aplica a un sistema de referencia con el eje z orientado perpendicularmente a la mesa o al orificio. Otra forma de entenderlo es sobre cuál los dos *robtargets*, de la Ilustración 98 queremos trabajar.

En caso de que realicemos esta opción de entrada con ángulo en los orificios 4, 5 y 6 no saldrá el menú de la Ilustración 113, se omitirá completamente. Esto es debido a que estos orificios no tienen inclinación.

En esta opción en la que introduces la herramienta con un ángulo predeterminado, al tener que dar tanta información, se mostrarán las opciones marcadas por pantalla, para informar al usuario, hasta que acabe el movimiento. Por ejemplo:

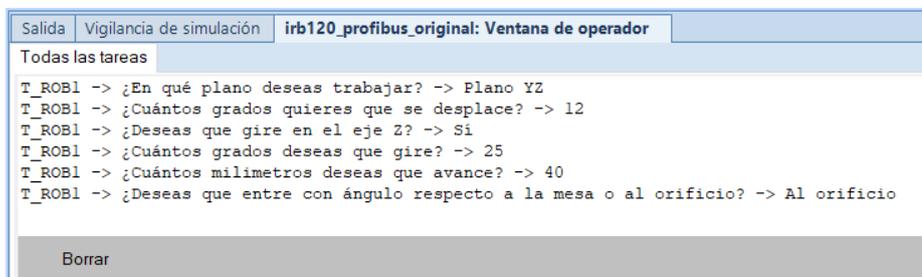


Ilustración 114 - Información en la pantalla de las opciones seleccionadas

De esta forma, el usuario podrá tener en pantalla la información que le urja sobre los datos introducidos.

Para ver cómo se realiza este conjunto de ordenes en la opción de entrada con ángulo o 'Ángulo' como se diría en la interfaz de usuario, veremos cómo se ejecuta el ejemplo explicándolo e ilustrándolo. Inicialmente tenemos la herramienta en el orificio 7:

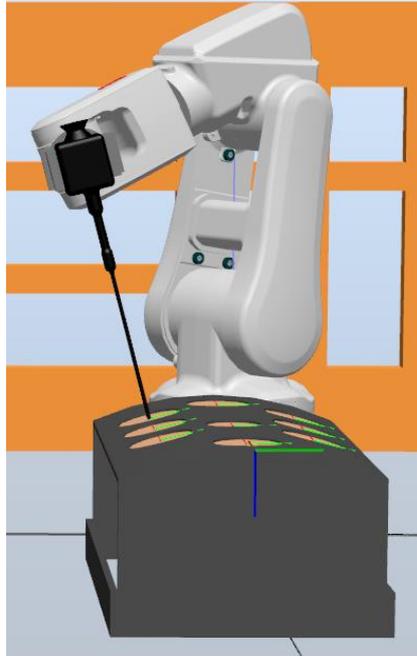


Ilustración 115 - Posición inicial del ejemplo en orificio 7

Tenemos los siguientes datos para que la herramienta entre con ángulo en el orificio 7:

Salida	Vigilancia de simulación	irb120_profibus_original: Ventana de operador
Todas las tareas		
<pre> T_ROB1 -> ¿En qué plano deseas trabajar? -> Plano YZ T_ROB1 -> ¿Cuántos grados quieres que se desplace? -> -15 T_ROB1 -> ¿Deseas que gire en el eje Z? -> Si T_ROB1 -> ¿Cuántos grados deseas que gire? -> -60 T_ROB1 -> ¿Cuántos milímetros deseas que avance? -> 100 T_ROB1 -> ¿Deseas que entre con ángulo respecto a la mesa o al orificio? -> Al orificio </pre>		
Borrar		

Ilustración 116 - Movimientos a realizar en la ejecución ejemplo

Se realizará en el siguiente orden las acciones señaladas en la Ilustración 116 acciones durante la simulación:

1. El primer movimiento que realizará el brazo robótico, con la herramienta negra, será el pivoteo en el plano YZ de -15 grados.

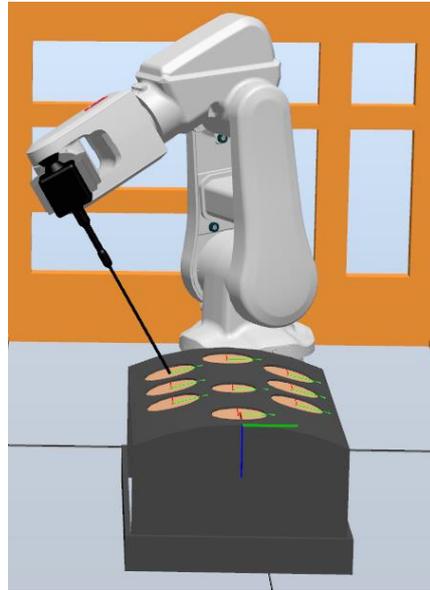


Ilustración 117 - Pivoteo de -15 grados en el plano YZ

2. Con esa orientación dada, realizará la rotación de -60 grados resultando de la siguiente forma:

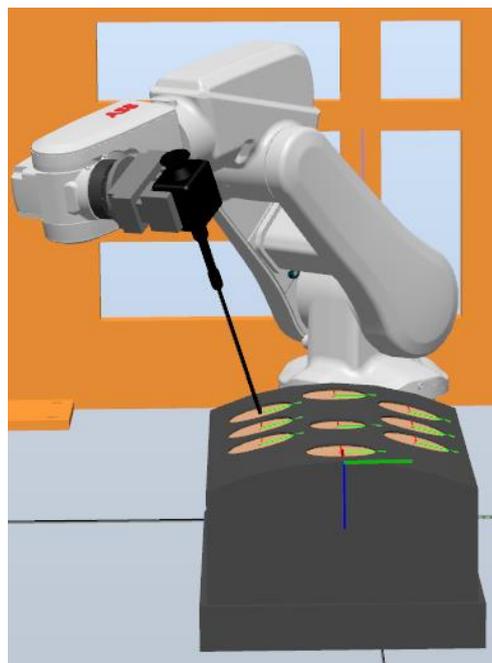


Ilustración 118 - Rotación de -60 grados a partir de la posición de la Ilustración 117.

3. Para la orientación mostrada, procederá a introducirse 100 milímetros quedando de la siguiente manera:

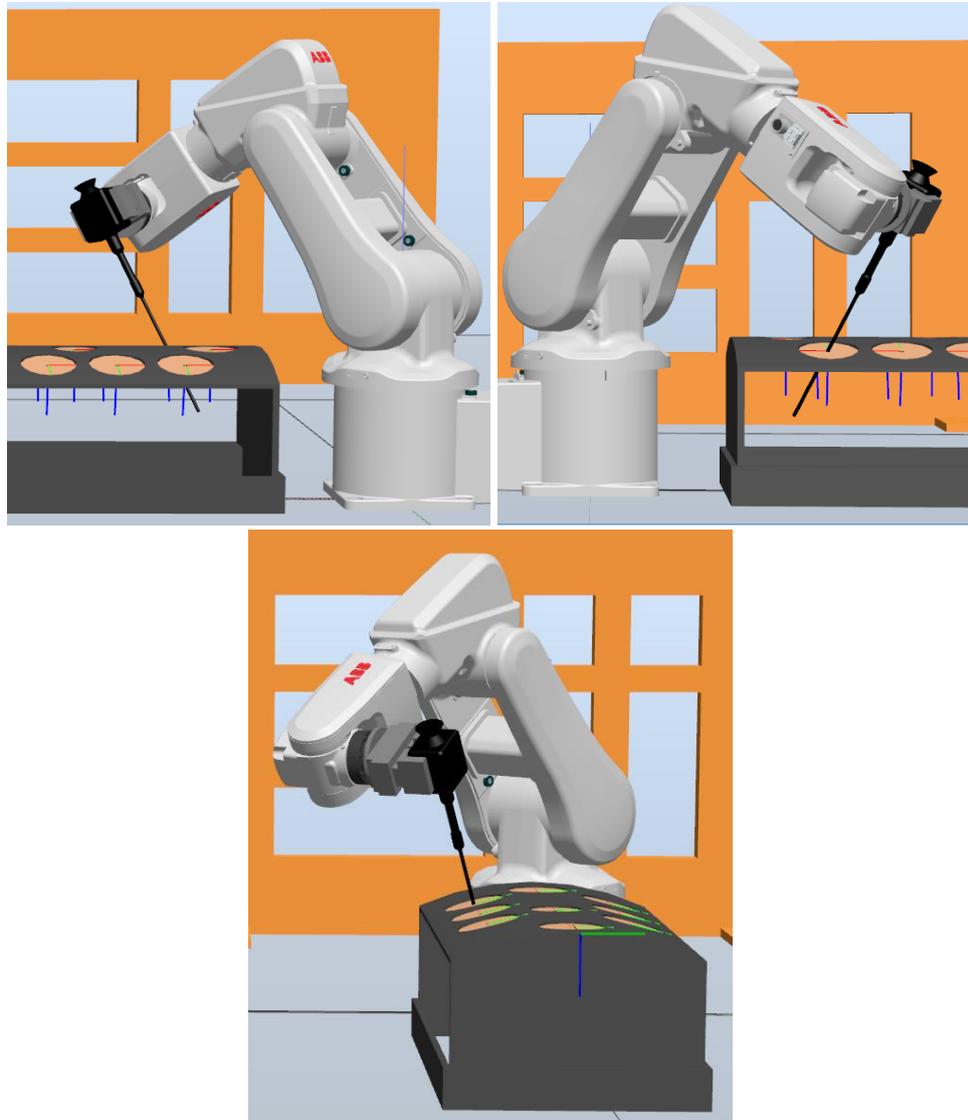


Ilustración 119 - Resultado de la ejecución para las ordenes de la ilustración 116

5.3.4 Salir en el menú orificio

Si en el menú de la Ilustración 91 marcamos la opción de ‘Salir’, el final de la herramienta se moverá del *robotarget* y no entrará en el orificio. El brazo robótico y la herramienta se colocará en un punto base de la siguiente forma:

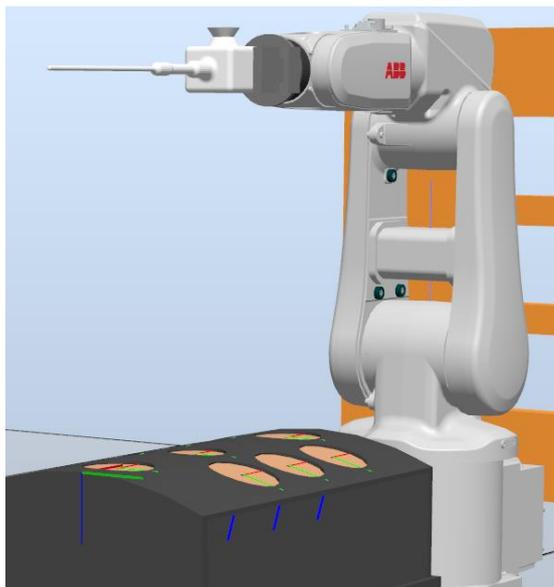


Ilustración 120 - Ejemplo de brazo robótico en posición de punto base para herramienta blanca

A su vez, aparecerá el siguiente menú por pantalla:

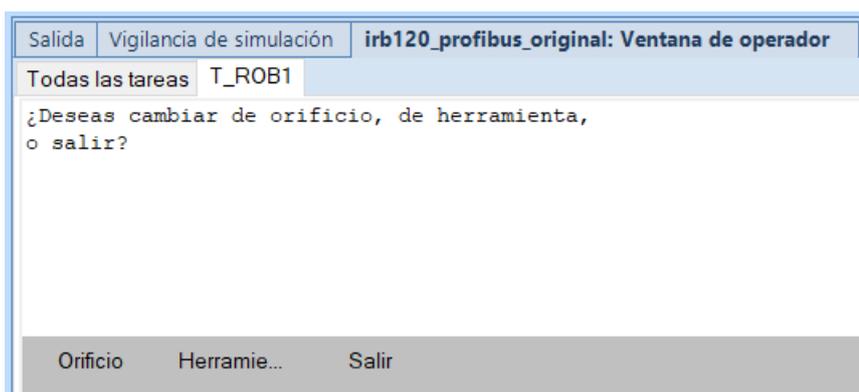


Ilustración 121 - Menú después de seleccionar 'salir' en el menú de la ilustración 21

Este menú es prácticamente idéntico al de la Ilustración 84. La única diferencia es que tenemos la opción de 'Orificio' en lugar de la de 'Aplicación'. A pesar de que tenga un nombre diferente su función es la misma, preguntar por el orificio sobre el que se quiere trabajar. El resto de las opciones de la Ilustración 121, 'Herramienta' y 'Salir', realizarán las mismas acciones que las de la Ilustración 84.

Volviendo al menú de la Ilustración 91, después de que se ejecuten completamente las opciones de entrada perpendicular a la mesa, al orificio o con ángulo, tendremos el siguiente menú.

5.4 Funcionamiento del menú movimiento

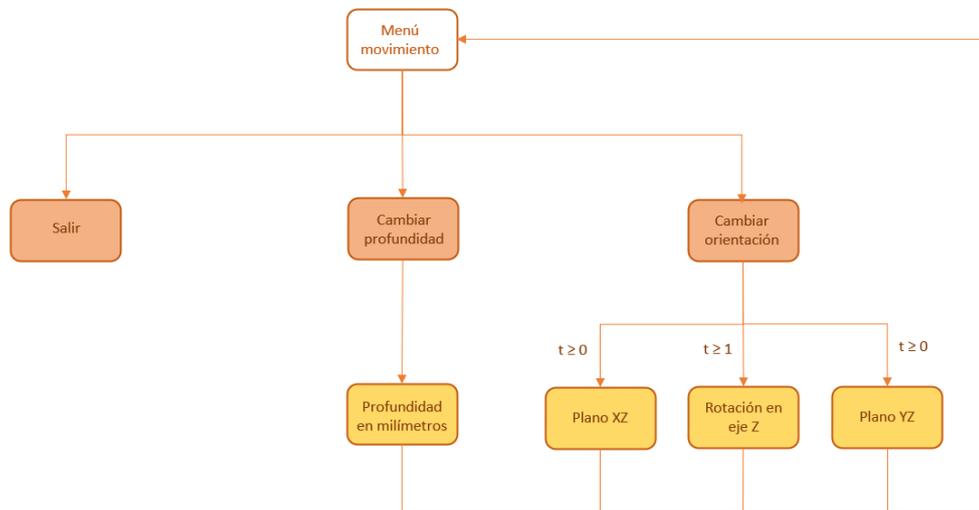


Ilustración 122 - Menú movimiento en el diagrama

Este menú sale por pantalla de la siguiente forma:

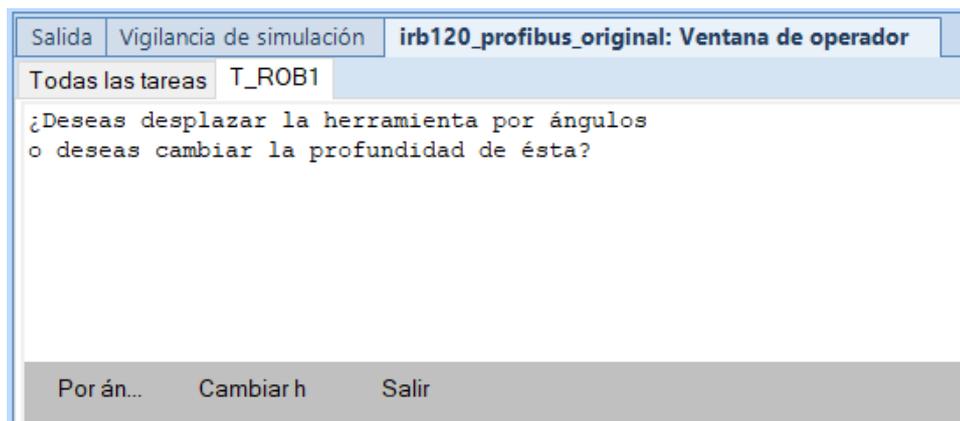


Ilustración 123 - Menú tras haber introducido la herramienta

Este menú tiene la característica de que se ejecutará en bucle a no ser que seleccionemos 'Salir'. Es decir, una vez seleccionemos 'Por ángulos' o 'Cambiar h' y se ejecute su respectiva acción, el menú que volverá a aparecer será el de la Ilustración 123. Al ser un bucle, se nos podría hacer la pregunta de la Ilustración 123 infinitas veces. La única manera de salir de este menú será eligiendo la opción 'Salir'.

A continuación, comentaremos lo que nos ofrece cada una de las opciones:

5.4.1 Opción de desplazamiento por ángulos

Esta opción se encargará de desplazar el final de la herramienta dentro del *Pelvitainer* mediante grados. Cuando presionemos esta opción, tendremos la siguiente cuestión por pantalla:

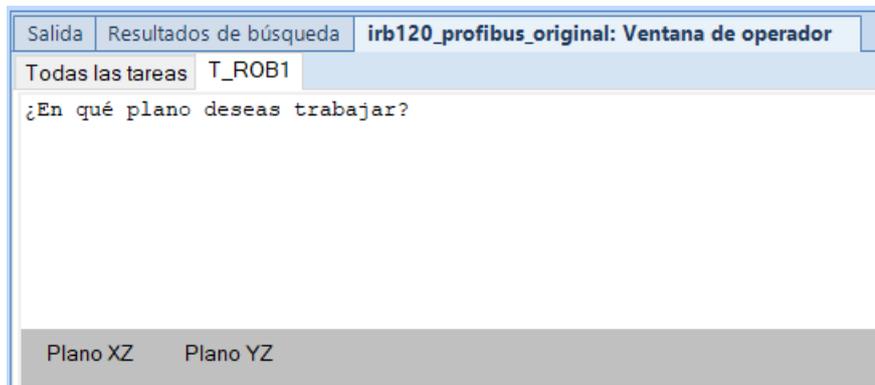


Ilustración 124 - Cuestión por pantalla tras seleccionar 'Por ángulos'

Una vez más, tendremos que elegir entre el plano XZ e YZ, se tratan de los mismos planos de los que se comentaron de la Ilustración 104. Si seleccionamos uno de esos planos, al igual que ocurría con el menú de la Ilustración 103, tendremos la siguiente cuestión:

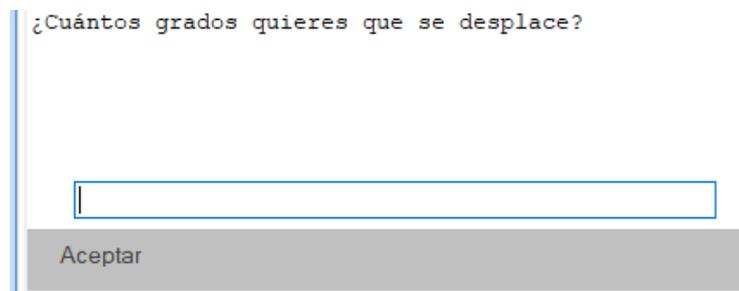


Ilustración 125 - Petición de grados para el plano seleccionado en la opción 'Por ángulos'

Tras haber introducido los grados deseados, la herramienta realizará su desplazamiento por el plano. Si recordamos, en el pivoteo la herramienta rotaba o se desplazaba por planos, manteniendo el final de la herramienta siempre fijo al origen de coordenadas del *robtarger* del orificio sobre el que se trabajara. Para la opción de 'Por ángulos' es algo similar, solo que, a diferencia del pivoteo, el final de la herramienta se podrá mover y el punto fijo de la herramienta que no se moverá, será la parte de ésta que este atravesando el orificio.

A continuación, mostramos la siguiente ilustración en la que se señala la parte de la herramienta que se quedará fija al orificio de entrada del *Pelvitainer*:

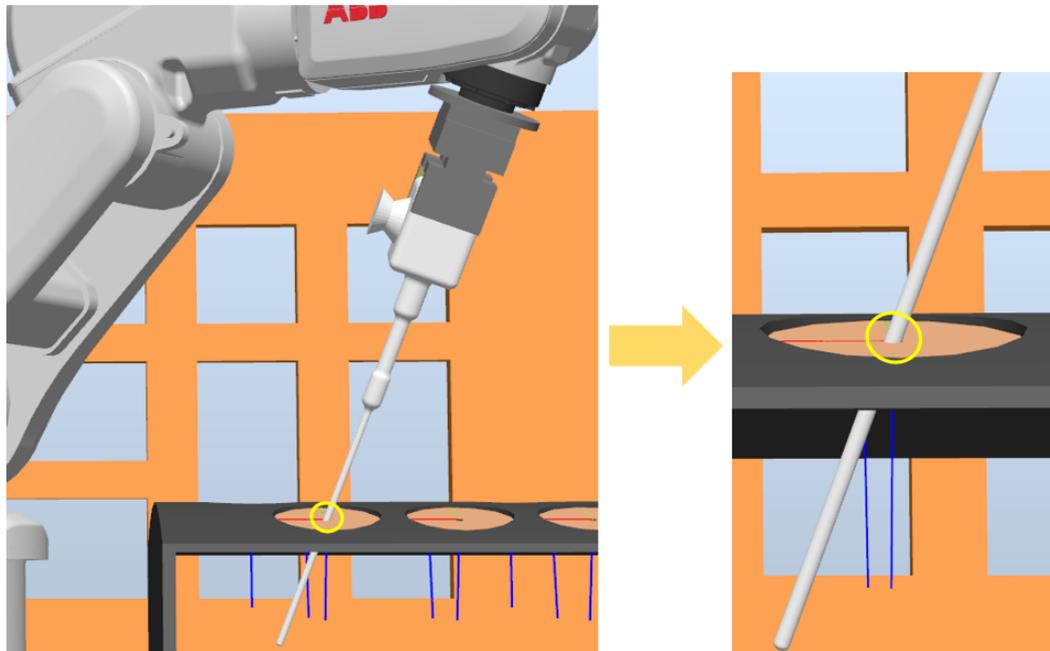


Ilustración 126 – Parte de la herramienta que no entra en contacto con la superficie del Pelvitainer

Como podemos ver, el punto fijo de la herramienta que no se moverá, durante el desplazamiento del final de la herramienta, será la parte de ésta que esté en la superficie del orificio. Da igual que se trate de un movimiento de rotación o desplazamiento en el plano XZ o YZ, la parte de la herramienta redondeada en la ilustración será inamovible.

La zona marcada en la Ilustración 126 es una señal de que la superficie del *Pelvitainer* no estará comprometida, porque la zona redondeada de la herramienta no se moverá en toda la acción. Además, esto se ha hecho conscientemente de que este proyecto está enfocado a la robótica aplicada a la cirugía laparoscópica.

Como en el pivoteo, el cuadro de diálogo de la Ilustración 125 admitirá valores negativos. Con los grados en valor negativo, la herramienta se desplazará en la dirección opuesta a la que iría para el mismo valor positivo.

Si observáramos el *Pelvitainer* desde el mismo punto de vista que la articulación 1 del brazo robótico, para un ángulo positivo en el plano YZ, el final de la herramienta se movería hacia la derecha y, por lo tanto, la ‘cabeza’ de la herramienta hacia la izquierda. Para un valor negativo, en el plano YZ ocurriría lo opuesto, el final de la herramienta se movería hacia la izquierda y, con lo cual, la ‘cabeza’ de la herramienta hacia la derecha. En el caso del plano XZ, para un valor positivo, el final de la herramienta iría hacia atrás, mientras que la ‘cabeza’ de la herramienta irá hacia adelante. Para un valor

negativo de este ángulo XZ, ocurriría los movimientos opuestos a los comentados para un valor positivo en este mismo plano.

Al igual que con la opción de 'Cambio h' cuando termine la ejecución de la opción 'Por ángulos', nos volverá a aparecer por pantalla el menú de la Ilustración 123. Al presionar por segunda vez la opción de 'Por ángulos' nos aparecerá el mismo menú de la Ilustración 124 solo que con una opción más:

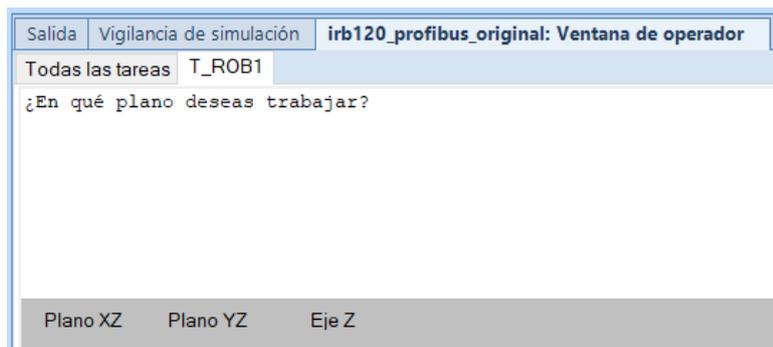


Ilustración 127 - Menú por pantalla al seleccionar por segunda vez 'Por ángulos'

Si nos fijamos, aparece la opción de 'Eje Z' esta opción nos permitirá rotar la herramienta a partir de la orientación que teníamos. En la rotación, el final de la herramienta trazará el arco de una circunferencia de una longitud que dependerá del número de grados que se le haya comandado. La lógica que sigue esta ejecución es similar al pivoteo en el eje z que realizaba en la opción entrada con ángulo.

Al igual que en los planos XZ e YZ, el final de la herramienta se moverá, mientras que la parte de la herramienta que se encuentra en el orificio no se moverá al igual que en la Ilustración 126. En el ejemplo de más adelante veremos cómo se ejecuta esta opción.

Tras pulsar la opción de 'Eje Z', tendremos un cuadro de diálogo como el de la Ilustración 109. Este cuadro de diálogo aceptará tanto valores positivos como valores negativos. Si comandamos grados positivos la herramienta girará en sentido horario, mientras que si introducimos valores negativos la herramienta girará en sentido antihorario.

Esta situación de que aparezca la opción de 'Eje Z' al presionar por segunda vez la opción de 'Por ángulos' se dará solo en dos situaciones. Estas dos situaciones serán cuando el primer movimiento que hayamos hecho al introducir la herramienta al orificio sea perpendicular al orificio o perpendicular a la mesa. Esto ocurre porque, si aplicamos una rotación a la herramienta habiendo entrado perpendicularmente al orificio o a la mesa, la

posición del final de la herramienta no cambiará. Pero, si aplicamos un desplazamiento por ángulo previamente, la posición del final de la herramienta cambiará, y ahí sí que será útil realizar una rotación en el eje Z, porque la posición del final de la herramienta cambiará.

Sin embargo, si el movimiento con el que hemos introducido la herramienta en el orificio ha sido una entrada con ángulo, sí que nos aparecerá la opción 'Eje Z' al seleccionar la opción de 'Por ángulos' por primera vez. Porque, si aplicamos una rotación después de que la herramienta haya entrado con ángulo, sí que cambiará la posición del final de la herramienta.

A continuación, mostramos un ejemplo del uso de esta opción de 'Por ángulos'. Inicialmente tenemos el final de la herramienta en el orificio 8 y la introducimos 110 milímetros perpendicularmente al agujero:

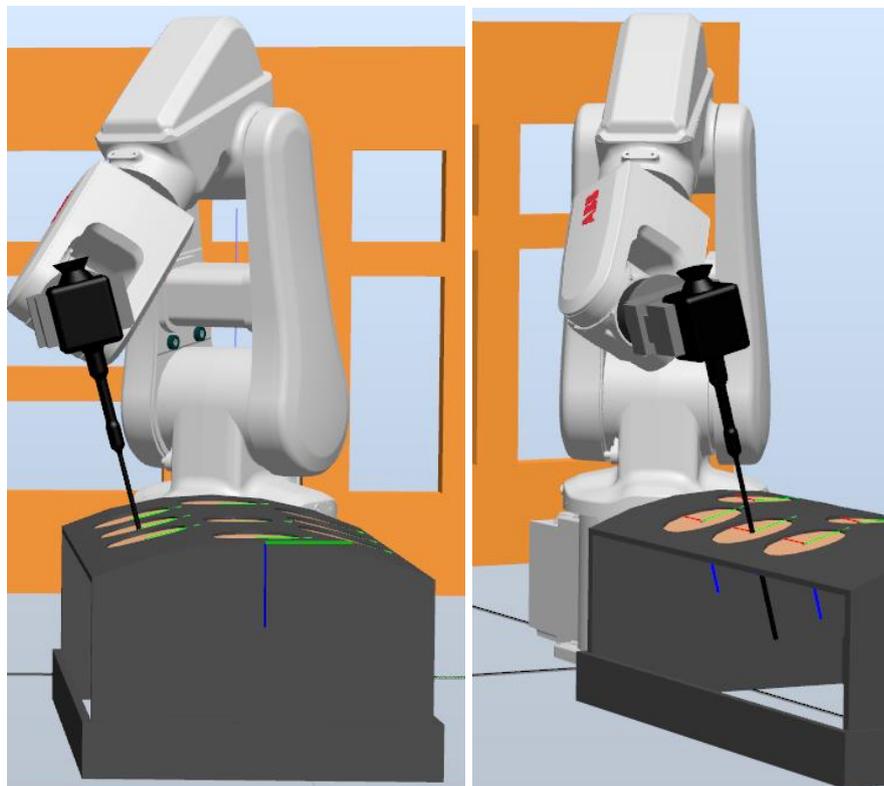


Ilustración 128 - Herramienta introducida 110 milímetros perpendicularmente al orificio

Luego, le aplicamos el primer desplazamiento por ángulos. Se desplazará en el plano YZ -12 grados, quedando la herramienta en esa posición:

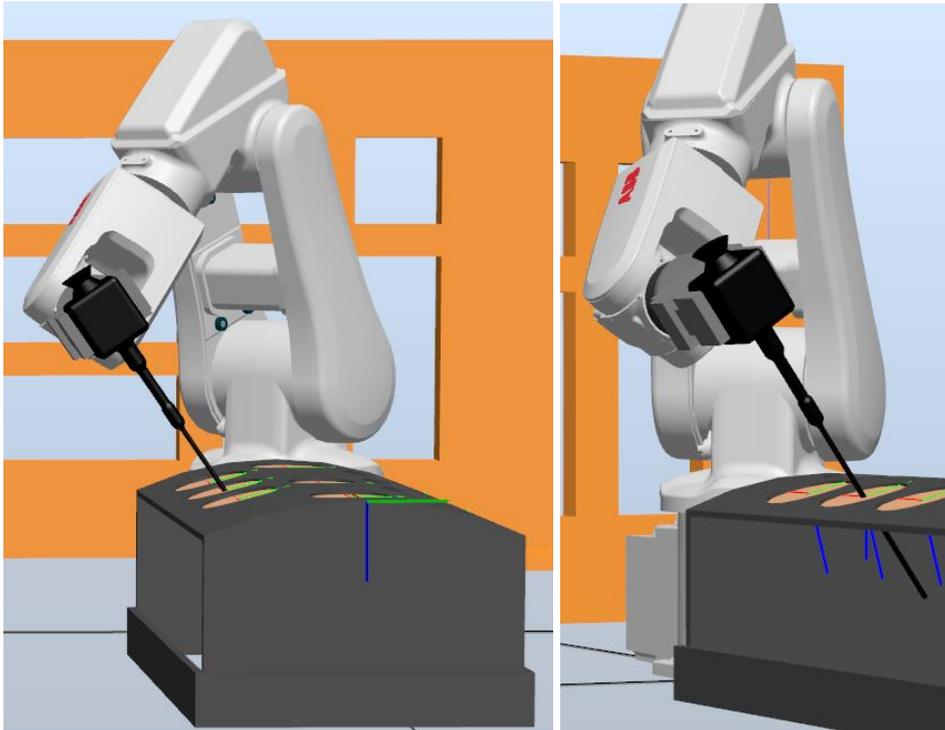


Ilustración 129 - Herramienta desplazada -12 grados en el plano YZ a partir de la Ilustración 128

Podemos fijarnos que la herramienta tiene una inclinación mayor que en la Ilustración 128. Si miramos desde la articulación 1 del brazo robótico, vemos que, comparado con la Ilustración 128, la ‘cabeza’ de la herramienta está más a la derecha y el final de la herramienta más a la izquierda.

También, se puede apreciar cómo la parte de la herramienta que se encuentra en el orificio no se ha movido de él.

Acabado este movimiento, nos volverá a aparecer el menú de la Ilustración 123, seleccionamos otra vez la opción ‘Por ángulos’. Al elegir esta opción por segunda vez nos aparecerá por primera vez la opción de ‘Eje Z’, usaremos esta opción.

Dada la posición de la Ilustración 129, aplicamos una rotación de -50 grados:

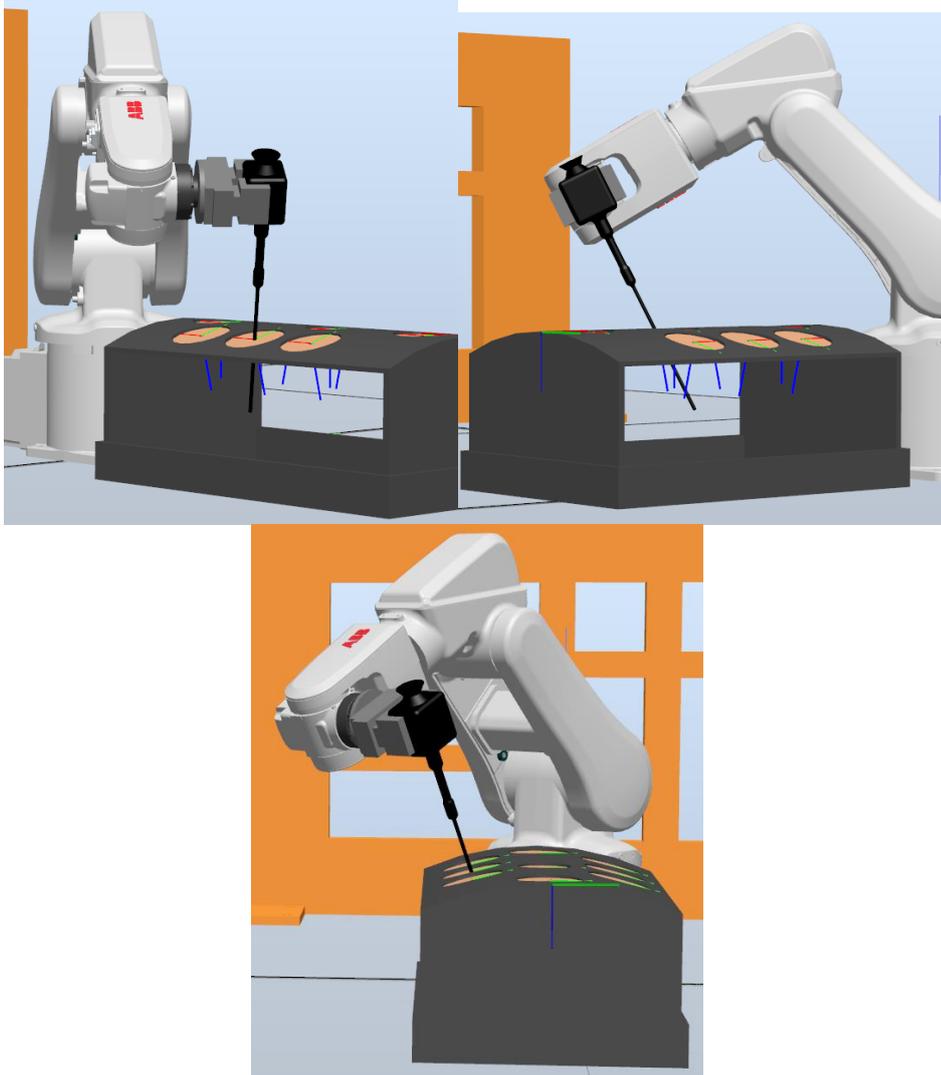


Ilustración 130 - Rotación de -50 grados aplicada a la Ilustración 129

Comparándolo con la Ilustración 129, se puede observar cómo notoriamente ha cambiado la orientación. Se puede apreciar el cambio de orientación si nos fijamos en la 'cabeza' de la herramienta. El final de la herramienta ya no tiene la misma posición que en la Ilustración 129. Una vez más, la parte de la herramienta que se encuentra en el orificio ha permanecido dentro de éste durante toda la rotación.

Alcanzada la posición de la Ilustración 130, volvemos a elegir la opción de 'Por ángulos'. Por último, realizaremos un desplazamiento de -30 grados en el plano XZ:

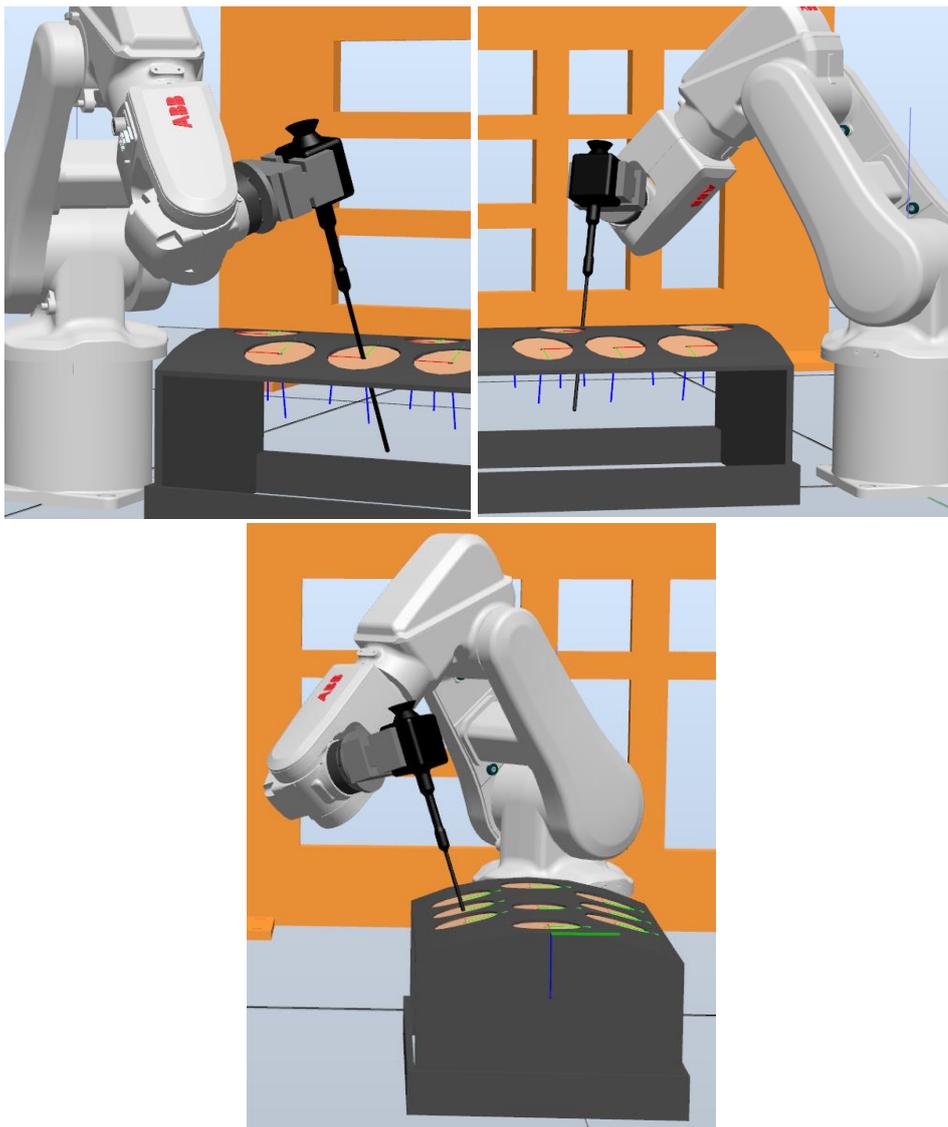


Ilustración 131 - Desplazamiento de -30 grados en el plano XZ de la Ilustración 130

Si nos fijamos en la Ilustración 131 se puede apreciar que la herramienta se ha desplazado por el plano XZ y ha cambiado su orientación. Desde el punto de vista de la articulación 1 del brazo robótico, la cabeza de la herramienta está más atrás, mientras que el final de la herramienta está más hacia adelante. Como siempre, se ha respetado la superficie del *Pelvitainer* y la parte de la herramienta que está atravesando el orificio apenas entra en contacto con él.

5.4.2 Opción de cambiar la profundidad a la que está introducida la herramienta

En el menú de la Ilustración 123, en lugar de que aparezca 'Cambiar altura' aparecerá 'Cambiar h'. Esto es debido a que la palabra 'altura' de 'Cambiar

altura' no se mostraba por completo en la interfaz de usuario. En su lugar se ha cambiado la palabra 'altura' por cómo se denota, 'h'. En inglés la palabra altura es 'height', ese es el motivo por el que se denota 'h', porque es la primera letra de la palabra.

Seleccionada esta opción, tendremos el siguiente mensaje por pantalla:

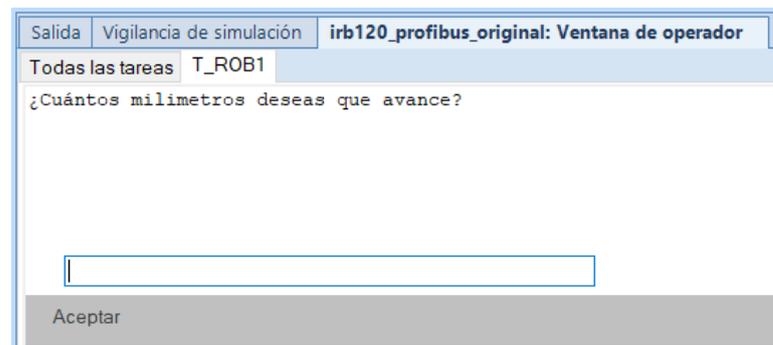


Ilustración 132 - Cuestión al usuario después de seleccionar 'Cambiar h'

Como se indica en la ilustración, las unidades que introduzcamos serán en milímetros. Si, por ejemplo, previamente, decimos al programa que se introduzca 90 milímetros la herramienta perpendicularmente a la mesa y después queremos que esté ahora introducida a 120 milímetros, en el cuadro de diálogo de la Ilustración 132, tendremos que poner 30 en lugar de 120 ($90 + 30 = 120$).

En otras palabras, en el cuadro de diálogo no se tecleará la distancia total a la que se quiere introducir, sino cuántos milímetros más queremos que se introduzca o cuántos milímetros menos queremos que suba.

Es decir, si en el cuadro de diálogo de la Ilustración 132, introducimos un valor 'x' positivo la herramienta se introducirá 'x' milímetros más de los que estaba introducido. En cambio, si en el mismo cuadro de diálogo introducimos un valor '-b' (valor negativo), la herramienta estará introducida 'b' milímetros menos. En otras palabras, para un valor negativo, la herramienta saldrá, del *Pelvitainer* con la orientación que tenía, esos 'b' milímetros.

Cabe destacar que estos cambios de distancia de la herramienta introducida en el *Pelvitainer*, se realizan manteniendo la orientación que tenía antes de realizarse este cambio. Nunca se alterará la orientación de la herramienta utilizando esta opción.

Si la herramienta está introducida perpendicularmente a la mesa y se cambia la distancia de profundidad, subirá o bajará manteniendo la orientación de perpendicular a la mesa. Lo mismo ocurrirá para una orientación perpendicular al agujero. De la misma manera, si la herramienta ha entrado con ángulo, la herramienta subirá o bajará con la orientación que tenía.

Acabada la ejecución de la opción de 'Cambiar h', nos volverá a aparecer el menú de la Ilustración 123. Entonces, si queremos volver a cambiar la distancia a la que está introducida la herramienta, volveremos a tener que presionar esta opción.

Por ejemplo, teniendo como caso inicial, la herramienta blanca introducida en el Pelvitainer 100 milímetros perpendicularmente al orificio 2:

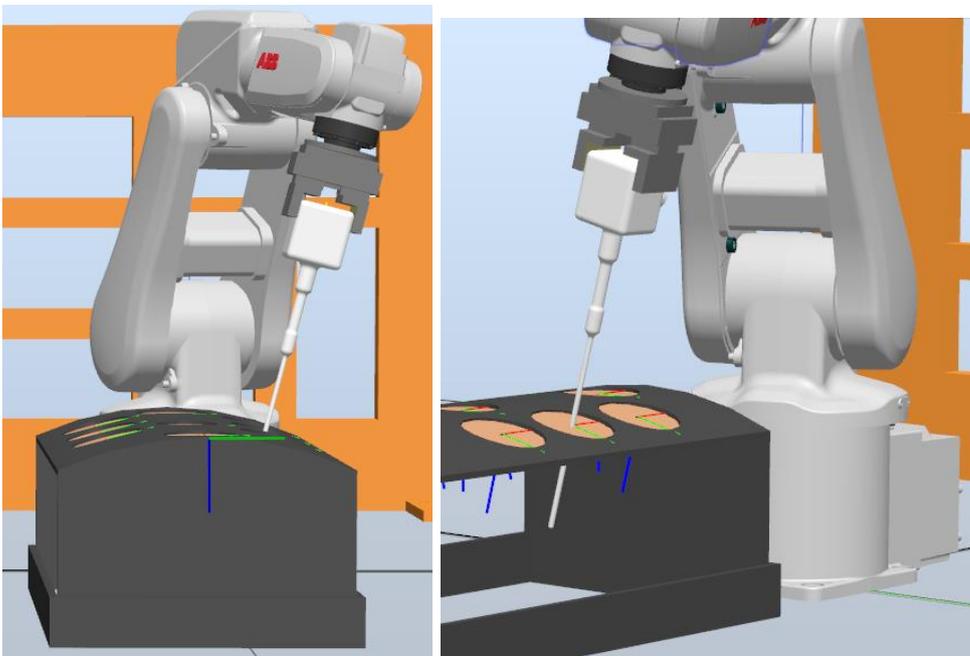


Ilustración 133- Caso inicial con herramienta introducida 100 milímetros perpendicularmente al orificio

Si elegimos la opción 'Cambiar h' y tecleamos -40 en el cuadro de diálogo:

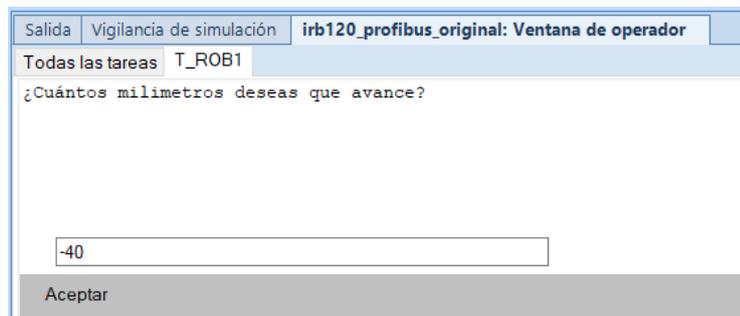


Ilustración 134 - Interfaz de usuario para ejemplo de la opción cambio de altura

Aceptamos y la herramienta subirá 40 milímetros y estará a 60 milímetros, (100 - 40 = 60):

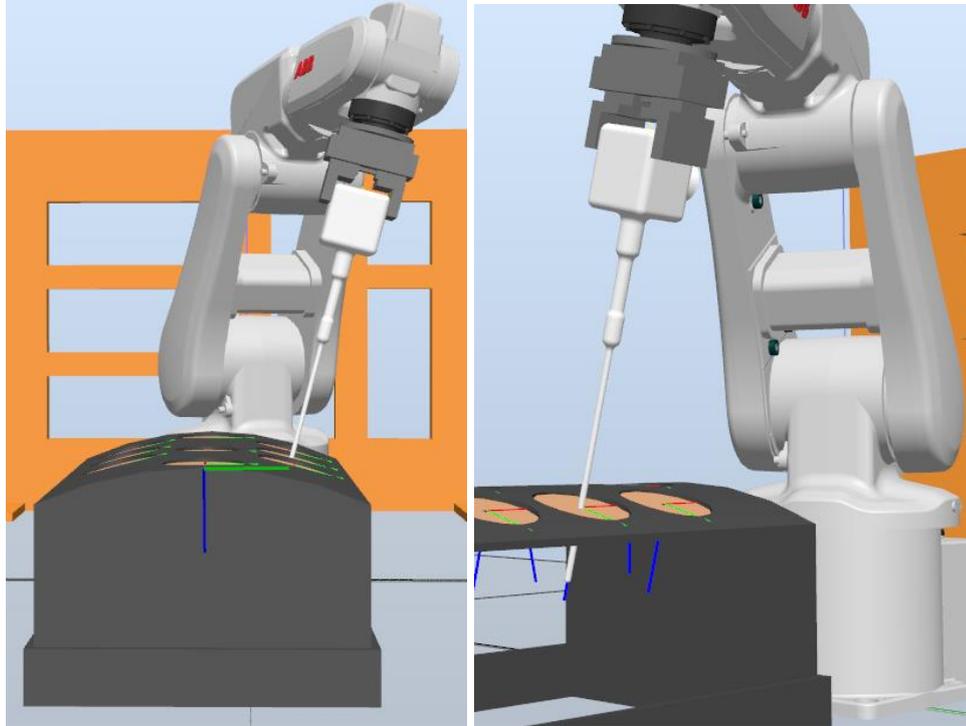


Ilustración 135 - Herramienta de la ilustración 133 subida 40 milímetros

Observamos que la herramienta ha subido 40 milímetros. Podemos fijarnos, en la ilustración de la izquierda, que la herramienta está más sacada. Otra forma de verlo es que, se puede notar que, el ángulo que forman las articulaciones 2 y 3, ha aumentado respecto a la Ilustración 133 porque ha subido la herramienta. Además, si realizamos zoom en el final de la herramienta de la ilustración de la derecha:



Ilustración 136 - Zoom del final de la herramienta de la ilustración 135

Se aprecia que, en el final de la herramienta, se ve el eje z del *robtarget* del orificio 2. Esto no ocurría en la Ilustración 133, en la que el eje z del *robtarget* del orificio 2, estaba tapado completamente. Con lo cual, esta Ilustración 136,

es otro indicativo de que la herramienta ha subido y hay una menor longitud de la herramienta introducida.

Entonces, siguiendo con el ejemplo, volvemos a presionar la opción de 'Cambiar h' en el menú de la Ilustración 123 e introducimos 100 milímetros en el cuadro de diálogo de la Ilustración 134. Al partir de la posición de la Ilustración 135, la herramienta quedará introducida 160 milímetros ($100 + 60 = 160$):

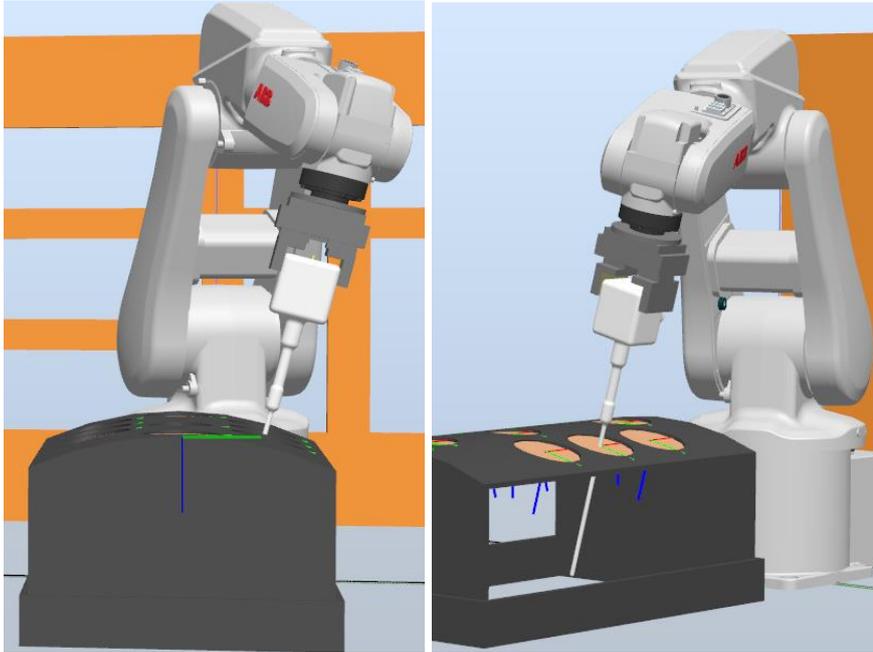


Ilustración 137 - Herramienta introducida 100 milímetros más respecto a la Ilustración 135

En la ilustración de la izquierda, podemos observar que el ángulo que forman las articulaciones 2 y 3 se ha reducido, siendo eso un indicativo de que la herramienta está introducida a más distancia. También en la ilustración de la derecha, ya no podemos ver el eje z del *robtarget* del orificio 2, otra señal de que la herramienta ha avanzado más en la orientación que tenía. Incluso podemos notar que el final de la herramienta está más cerca del plano donde se apoyan los elementos 3D de la estación.

Antes de comentar la opción de 'Salir' del menú de la Ilustración 123, mostraremos las opciones combinadas de 'Cambiar h' y 'Por ángulos' para que el usuario vea las infinitas posibilidades que hay combinando estas opciones.

Comenzaremos con la herramienta negra, situaremos el final de ésta en el orificio 1. Como primer movimiento, haremos una entrada con ángulo,

seleccionado en el menú de la Ilustración 91 con las siguientes características:

```
Salida  Vigilancia de simulación  irb120_profibus_original: Ventana de operador
Todas las tareas
T_ROB1 -> ¿En qué plano deseas trabajar? -> Plano XZ
T_ROB1 -> ¿Cuántos grados quieres que se desplace? -> 15
T_ROB1 -> ¿Deseas que gire en el eje Z? -> Sí
T_ROB1 -> ¿Cuántos grados deseas que gire? -> 30
T_ROB1 -> ¿Cuántos milímetros deseas que avance? -> 90
T_ROB1 -> ¿Deseas que entre con ángulo respecto a la mesa o al orificio? -> Al orificio
Borrar
```

Ilustración 138 - Características del movimiento entrada con ángulo del ejemplo

Pivoteará con un ángulo de 15 grados en el plano XZ, esta orientación sufrirá un giro de 30 grados y se introducirá 90 milímetros en un sistema de referencia del orificio. Primero mostramos la orientación que tenía antes de entrar en el *Pelvitainer*:

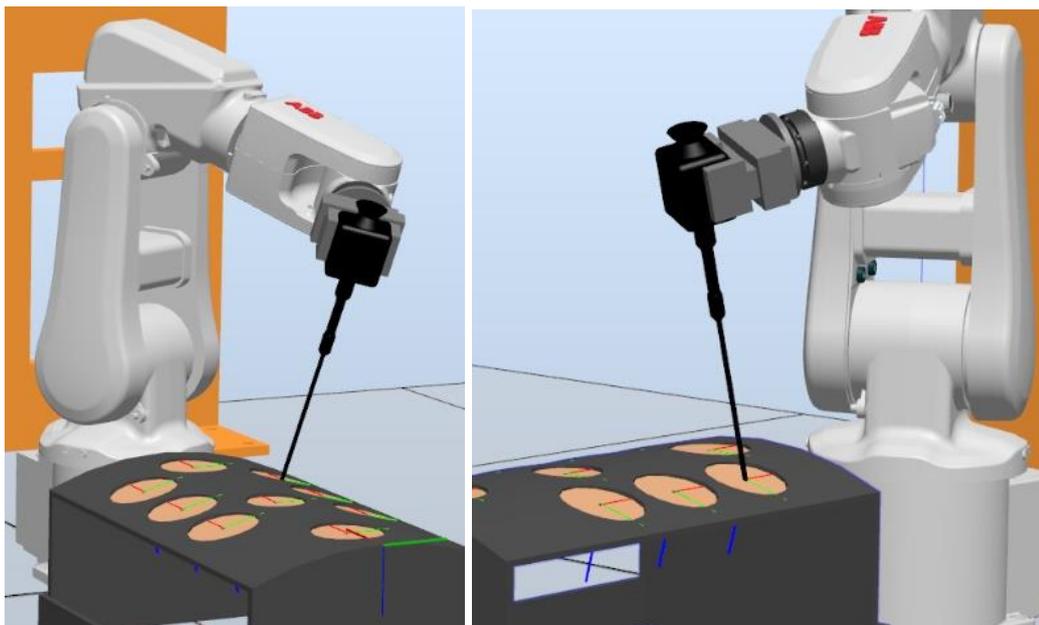


Ilustración 139 - Orientación previa de la herramienta al entrar en el *Pelvitainer*

En esta ilustración ya ha realizado el pivoteo de 15 grados en el plano XZ y ha rotado 30 grados. Luego, a partir de esa orientación entrará en el *Pelvitainer*:

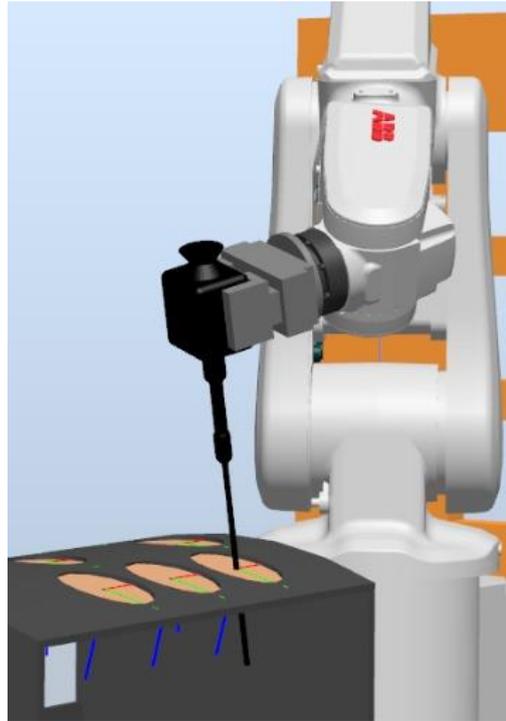


Ilustración 140 - Entrada con ángulo completa del ejemplo

Realizado este movimiento, tendremos el menú de la Ilustración 123, seleccionaremos la opción 'Cambiar h' para que la herramienta esté introducida con más profundidad en el *Pelvitainer*.

En la Ilustración 140, como hemos visto está 90 milímetros introducida, queremos que esté introducida a 130 milímetros.

Como queremos 130 milímetros, en el cuadro de diálogo de la Ilustración 132, que nos saldrá después de seleccionar la opción 'Cambiar h', introduciremos 40 ($90 + 40 = 130$) y tendremos el siguiente resultado:

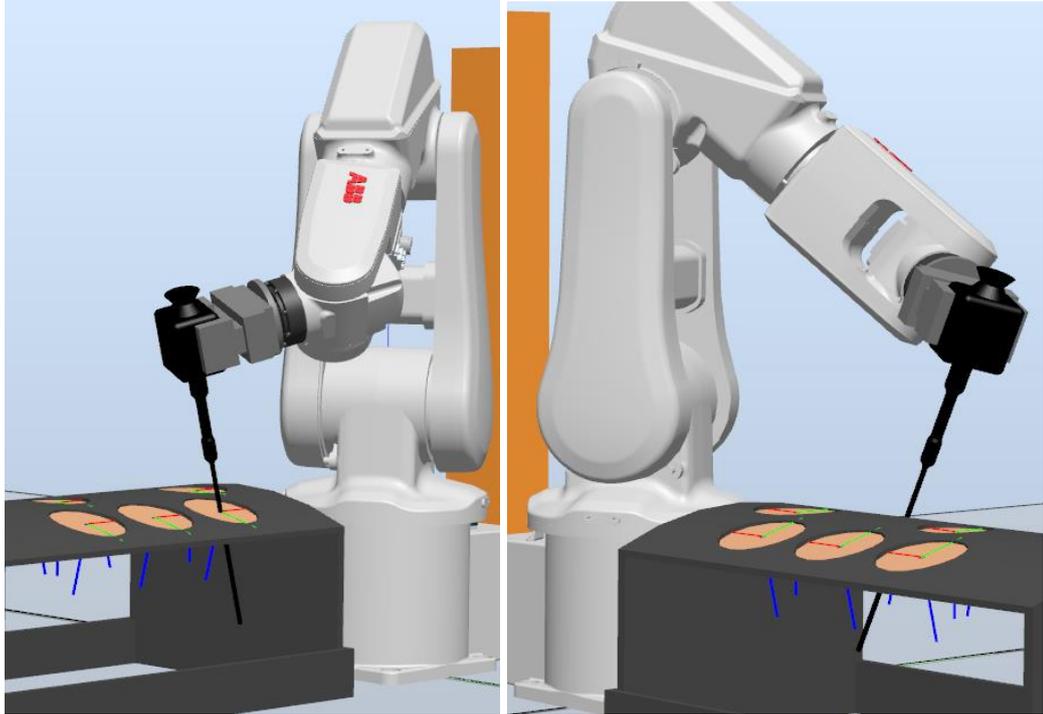


Ilustración 141 - Herramienta introducida 40 milímetros más respecto a la Ilustración 140

Podemos ver que se ha mantenido la orientación que tenía anteriormente a pesar de que la herramienta haya entrado más en el orificio. A continuación, en el menú de la Ilustración 123 marcamos la opción 'Por ángulos' y realizamos un desplazamiento en el plano YZ de -20 grados:

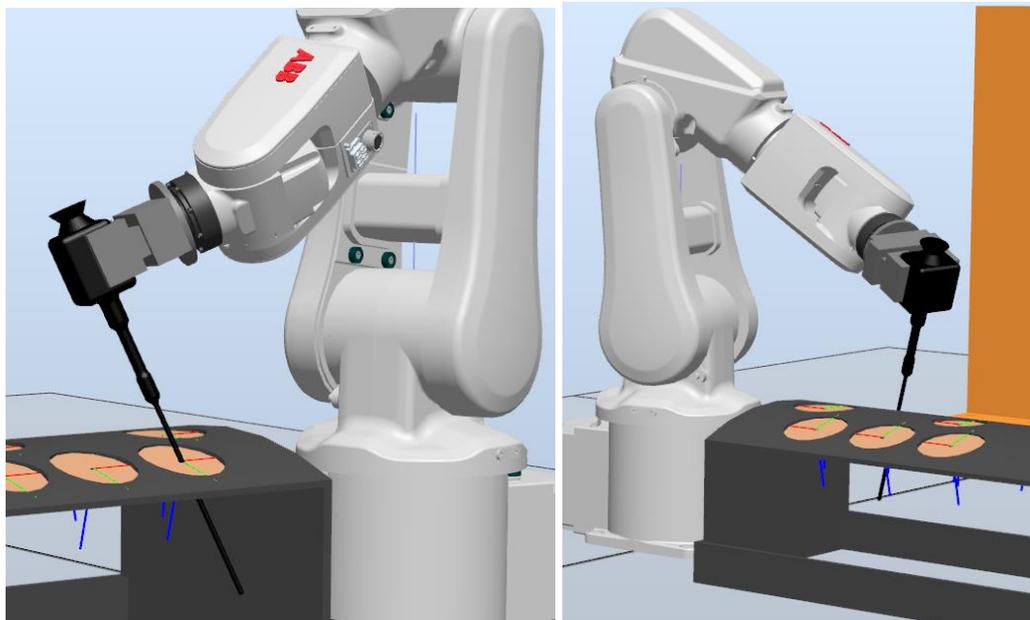


Ilustración 142 - Desplazamiento en el plano YZ de -20 grados a partir de la posición de la Ilustración 141

Notamos que se sigue respetando el orificio por el que pasa la herramienta y no se ve comprometido. Además, si miramos al *Pelvitainer* desde donde está

la articulación 1 del brazo robótico, podemos ver que la ‘cabeza’ de la herramienta se ha desplazado más a la derecha y el final de la herramienta más a la izquierda.

Para finalizar este ejemplo, realizaremos una rotación de 40 grados. Así que, en el menú de la Ilustración 123, volvemos a seleccionar ‘Por ángulos’ y seleccionamos la opción de ‘Eje Z’ y le decimos que rote 40 grados. La herramienta quedará de la siguiente forma:

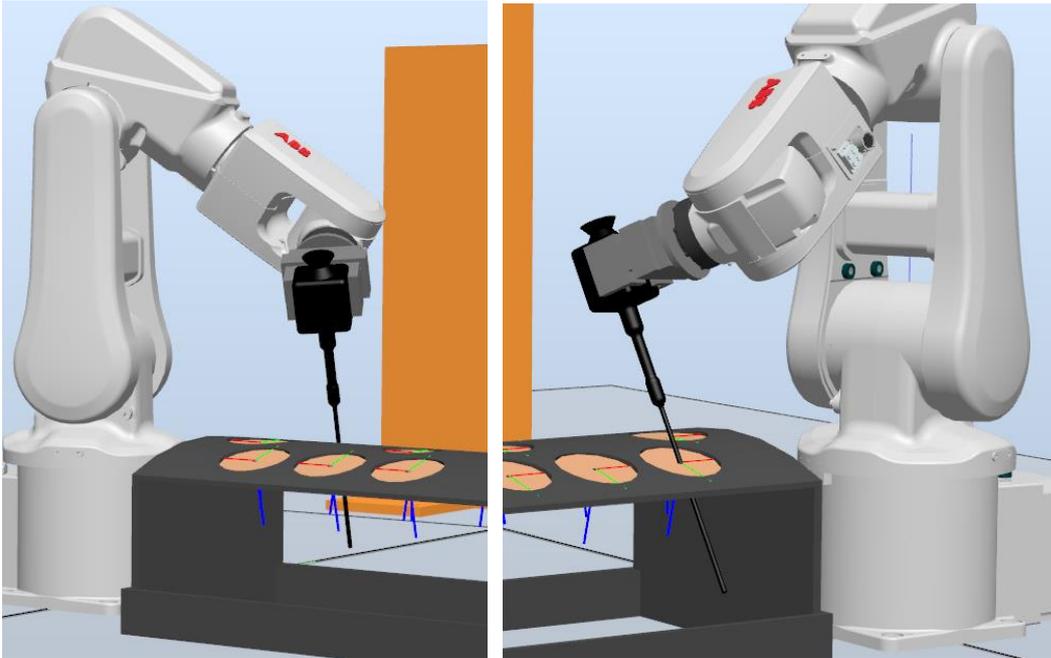


Ilustración 143 - Rotación de 40 grados aplicada a la Ilustración 142

Realizándose este giro en sentido horario se cuida que la herramienta no atraviese mucho más allá el perímetro de la circunferencia del orificio. Si nos fijamos en la ‘cabeza’ de la herramienta, podemos notar como su orientación ha cambiado por la rotación que ha sufrido.

5.4.3 Salir en el menú movimiento

En el menú de la Ilustración 123, si elegimos esta opción ocurrirá lo siguiente:

- La herramienta se colocará en la posición que tenía al entrar por primera vez en el orificio, pero con la orientación del eje Z del *robtarjet* del orificio.
- A continuación, la herramienta se moverá linealmente hasta la posición en la que el final de ésta esté en el origen de coordenadas del *robtarjet*. Siguiendo con el ejemplo mencionado de la Ilustración 133,

el final de la herramienta estaría en el origen de coordenadas del *robtarget* del orificio 2 como en la siguiente ilustración:



Ilustración 144 - Final de la herramienta en orificio 2 durante la ejecución de la opción 'Salir'

- Por último, el brazo robótico se colocará en un punto base quedando en la misma posición que en la Ilustración 120

Todas estas acciones mencionadas se realizarán automáticamente. Acabado este conjunto de actividades volveremos a tener el menú de la Ilustración 121. Es decir, la opción 'Salir' del menú de la Ilustración 91 tiene la misma finalidad que el menú de la Ilustración 123 y dejaría al robot en el mismo estado final que el de la Ilustración 120.

Volviendo al menú principal que tenemos al arrancar la simulación, el menú de la Ilustración 82, comentaremos qué ofrecerá la opción de la *demo*:

5.5 Funcionamiento del menú *Demo*



Ilustración 145 - Menú demo en el diagrama

Una vez seleccionada la opción de la *demo*, el usuario visualizará el siguiente menú por pantalla:

```
Esta es la demo, puedes elegir entre las 5 siguientes opciones:  
1- Demostración con herramienta blanca  
2- Demostración con herramienta negra  
3- Barrido  
4- Circunferencia  
5- Salir  
¿Qué opción deseas?  
  
Aceptar
```

Ilustración 146 - Menú de la demo

Cada opción ofrece las siguientes acciones:

5.5.1 Demostración con la herramienta blanca en el menú *Demo*

Con esta opción el usuario podrá ver el abanico de opciones que tiene con la herramienta de color blanco, que se usa con un agarre superior. En esta demostración, el usuario podrá ver las alternativas de movimientos que tiene

con la herramienta de color blanco para cuando lo use en la opción de la aplicación.

Cuando se active esta opción, el robot cogerá la herramienta de color blanco de su respectivo soporte. El robot moverá la herramienta y colocará el final de ésta en el orificio 2. Llegará al orificio respetando la superficie del *Pelvitainer*. Los movimientos que realizará para esta demostración son los siguientes:

- El robot empezará introduciendo la herramienta 150 milímetros perpendicularmente al plano del orificio.
- Posteriormente, se desplazará -12 grados en el plano YZ.
- Luego, realizará una rotación de 45 grados, en otras palabras, girará respecto al eje Z.
- Con esa orientación, el robot sacará la herramienta 30 milímetros. Dejando así la herramienta introducida unos 120 milímetros de profundidad.
- A continuación, se moverá 10 grados en el plano XZ.
- Dada la orientación anterior el final de la herramienta se desplazará -15 grados en el plano YZ.
- Después, se introducirá la herramienta 40 milímetros más, quedando la herramienta introducida unos 160 milímetros.
- Por último, realizará una rotación de -30 grados en el eje Z

Cada uno de estos movimientos se realizan a partir de la posición y orientación que quedó al finalizar el anterior movimiento y respetando el área del orificio por el que la herramienta entró.

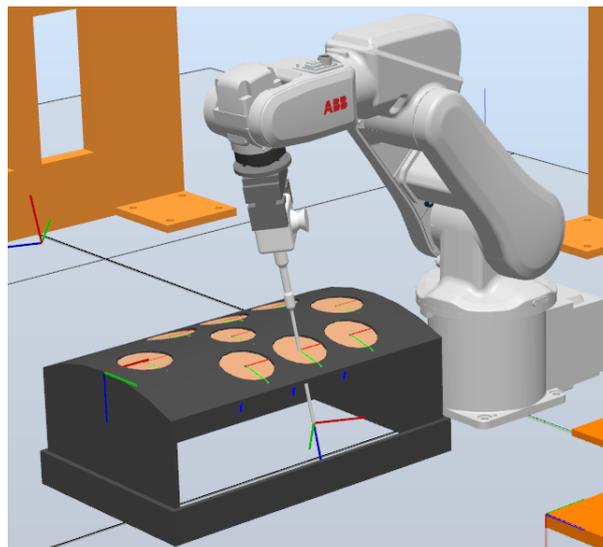


Ilustración 147 - Ejemplo de opción 1 de la demo

Además, durante la simulación de esta opción el programa irá informando del movimiento que está realizando:

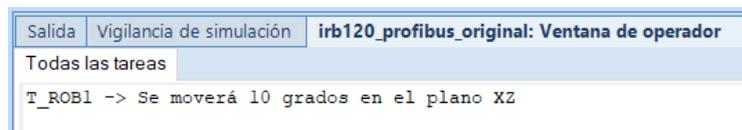


Ilustración 148- Mensaje por pantalla en la opción de 'Demostración con herramienta blanca'

Efectuados estos movimientos, el brazo robótico hará un movimiento lineal hacia la primera posición de esta opción 1, en la que la herramienta estaba introducida 150 milímetros perpendicularmente al plano del orificio. Luego el robot, llevará linealmente el final de la herramienta al *robtarget* del orificio 2. Finalmente, el robot dejará la herramienta blanca en su respectivo soporte.

5.5.2 Demostración con la herramienta negra en el menú *Demo*

Igual que la anterior opción, ésta está destinada a demostrar al usuario las posibilidades que tiene con la herramienta de color negro, con la cual se usa un agarre lateral. De esta forma, el usuario tendrá una idea de las oportunidades que tiene al utilizar dicha herramienta dentro de la opción de la aplicación, dónde él mismo comanda los movimientos y las unidades.

El robot cogerá dicha herramienta de su respectivo soporte, se dirigirá al orificio 8 y realizará la demostración en este orificio, al cual, llegará con el final de la herramienta respetando la superficie del *Pelvitainer*. Los movimientos que realizará en esta demostración son los siguientes:

- El brazo robótico comenzará introduciendo la herramienta 80 milímetros perpendicularmente a la mesa.
- Alcanzada esa posición, se desplazará 15 grados en el plano XZ.
- Manteniendo esa orientación, la herramienta subirá 20 milímetros. Con lo cual, la herramienta estará introducida 60 milímetros.
- Después, se volverá a desplazar 15 grados en el plano XZ.
- Con la orientación anterior, recorrerá -15 grados en el plano YZ.
- A continuación, la herramienta se introducirá 30 milímetros más. De esta forma la herramienta quedará introducida 90 milímetros.
- Dada esa posición, realizará una rotación de 35 grados. En otras palabras, girará 35 grados en el eje Z.
- De nuevo, se introducirá otros 15 milímetros quedando introducida la herramienta 105 milímetros.
- Por último, realizará una rotación de -25 grados, es decir, giraría en el eje Z.

Al igual que la opción 1, todos estos movimientos se realizan a partir de la posición y orientación que quedó al finalizar el anterior movimiento y respetando el área del orificio por el que la herramienta entró.

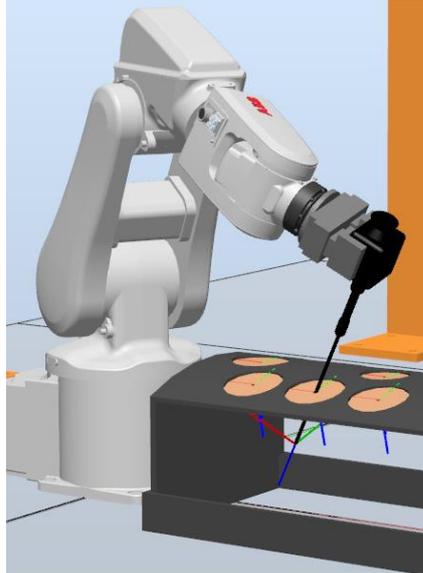


Ilustración 149- Ejemplo de opción 2 de la demo

Como en la opción anterior, se le irá informando al usuario del movimiento que está realizando el robot al momento:

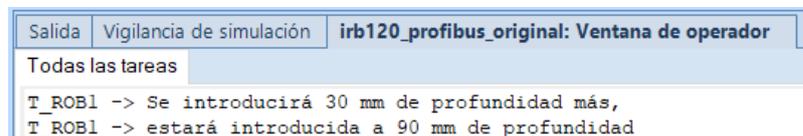


Ilustración 150- Mensaje por pantalla del movimiento de la ilustración 149

Realizados estos movimientos, el robot realizará un movimiento lineal hacia la primera posición de esta demostración, en la que la herramienta estaba introducida 80 milímetros, pero perpendicularmente al orificio. Posteriormente, el robot llevará linealmente el final de la herramienta al *robtarger* del orificio 8. Finalmente, el brazo robótico dejará la herramienta en el soporte, en su posición original.

5.5.3 Opción “Barrido” en el menú *Demo*

Seleccionada esta opción, el brazo robótico empezará cogiendo la herramienta de color blanco, que es la de agarre superior. El robot dirigirá la herramienta al orificio número 1 respetando la superficie del *Pelvitainer* de forma que no entre en contacto con ella. Una vez que el final de la herramienta haya llegado al orificio 1, la herramienta se introducirá 100 milímetros perpendicularmente al plano de la mesa. Posteriormente, realizará

el barrido con el final de la herramienta y este, dejará un rastro de la siguiente forma:

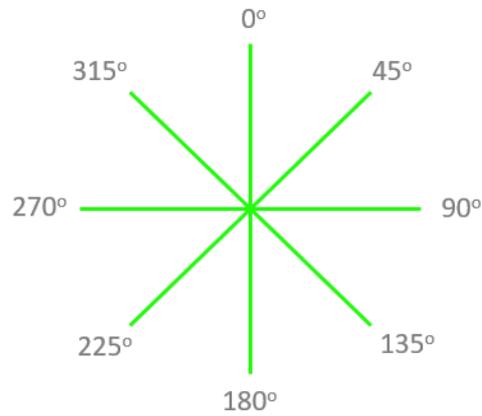


Ilustración 151 - Rastro que recorrerá el TCP en la opción de barrido

Observamos que cada 45° el final de la herramienta trazará un “aspa”, así hasta completar los 360°. El proceso entero está cuidado para que la herramienta respete el orificio de entrada. El barrido se realizará para un ángulo de 15 grados.

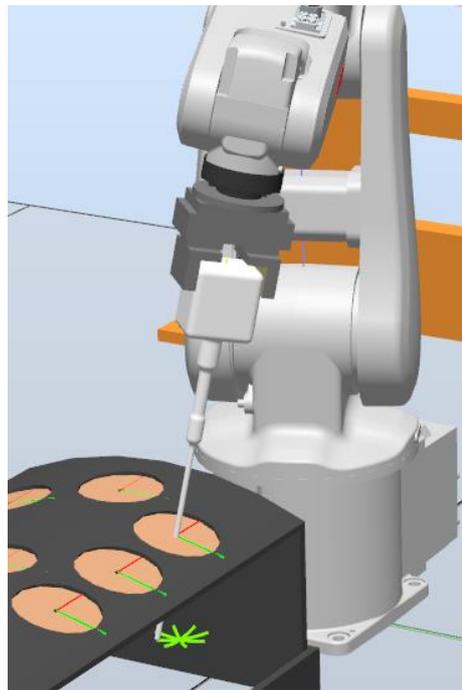


Ilustración 152 - Captura durante la simulación de la opción de barrido de la demo

De forma más detallada, el rastro que deja el TCP en la simulación de esta opción:

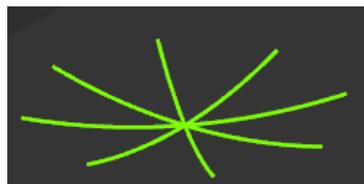


Ilustración 153 - Imagen detallada del rastro del TCP en el barrido

Saldrá por pantalla el siguiente mensaje al usuario: “Opción demo barrido”. Esto indicará que se encuentra en ejecución la opción de la demo del barrido.

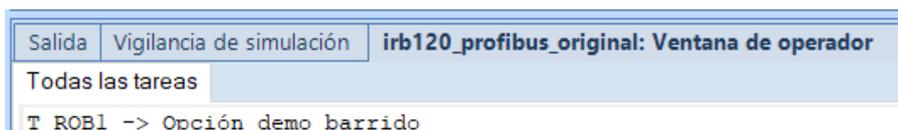


Ilustración 154 - Mensaje por pantalla en la opción de barrido de la demo

La longitud en cada trazo dependerá del ángulo que se haya establecido en el barrido, a más grados establecidos en la función, más largo será el recorrido por cada trazo. Pero, al tener que respetarse el orificio de entrada, la longitud del barrido aumentará de forma convexa, dado que siempre estará la herramienta introducida 100 milímetros. Por ejemplo, si el barrido fuera de 30 grados, el TCP dejaría el siguiente rastro:



Ilustración 155 - Rastro para 30 grados de barrido

En caso de que se deseara, se podría cambiar el ángulo de barrido, eso daría lugar a una mayor o menor longitud de barrido. Para valores menores que el establecido se garantiza su ejecución, pero para valores mayores al predeterminado podría dar fallo porque el robot podría alcanzar una singularidad. Como ya se ha mencionado, está preestablecido el barrido para un ángulo de 15 grados, el máximo ángulo con el que podría realizar el barrido sería con 30 grados. Si se intentara con un ángulo mayor a 30 grados, el robot alcanzaría una singularidad y la simulación se interrumpiría.

5.5.4 Opción “Circunferencia” en el menú *Demo*

Dentro de esta opción, el robot cogerá la herramienta de color blanco, nuevamente la de agarre superior. El robot con la herramienta agarrada se dirigirá al orificio 7 e introducirá la herramienta 84 mm perpendicularmente al plano de la mesa. Una vez que el final de la herramienta esté en esa posición final, se desplazará 15 grados en el plano XZ y desde esa posición se

realizará una circunferencia. Resultando de la siguiente manera en la simulación:

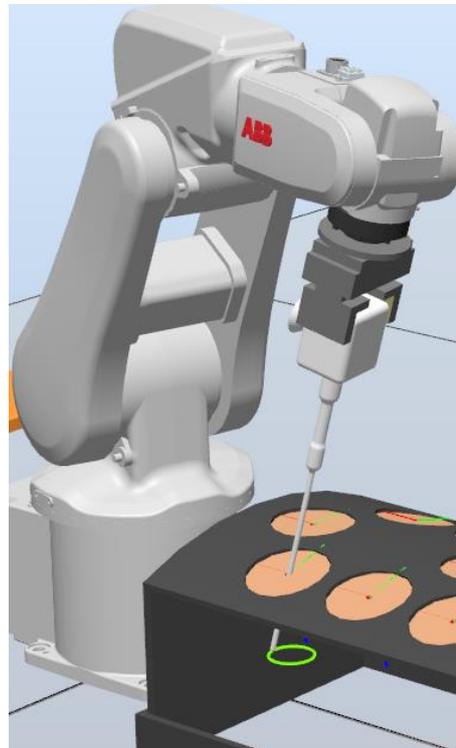


Ilustración 156 - Captura durante la simulación de la opción circunferencia

De forma más detallada, el rastro que deja el TCP en la simulación de esta opción:



Ilustración 157 - Imagen del rastro del TCP en la opción circunferencia

Saldrá por pantalla el siguiente mensaje al usuario: “*Opción demo circunferencia*”. Esto indicará que se encuentra en ejecución la opción de la demo de la circunferencia.

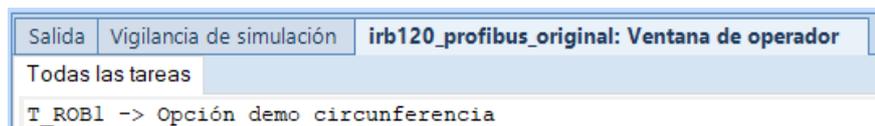


Ilustración 158 - Mensaje por pantalla durante la opción de circunferencia

Al igual que en la opción del barrido, a más grados se desee realizar la circunferencia, más radio tendrá esta. Porque para una longitud constante, como es en este caso con la herramienta introducida a 84 milímetros, si

introducimos más grados, el final de la herramienta se separará más respecto a la posición inicial que tenía al entrar perpendicularmente a la mesa.

De nuevo, de forma similar al barrido, si se deseara se podría variar los grados consiguiendo una circunferencia de diferentes radios. Para valores menores a 15 grados se garantiza su ejecución, pero para valores mayores podría dar fallo porque el robot podría alcanzar singularidades.

5.5.5 Opción “Salir” en el menú *Demo*

Si seleccionamos esta opción, saldremos del menú *demo*. Al salir, volveremos al menú inicial que se nos mostraba al principio de la simulación en la Ilustración 82. Volveremos a tener la alternativa de coger las herramientas o utilizar la *demo* en caso de que quisiéramos volver a usarla.

6. Límites

6.1 Límites de profundidad

Con la finalidad de conocer los límites, al introducir la herramienta perpendicularmente al orificio o a la mesa, para que el robot no alcance singularidades durante la simulación, se han realizado numerosos experimentos. En estos experimentos, se han ejecutado numerosas veces la acción de introducir perpendicularmente la herramienta al orificio o a la mesa hasta encontrar el máximo valor permitido para cada orificio.

Al poder introducirse la herramienta perpendicularmente a la mesa o al orificio tenemos dos casos posibles de longitud máxima. Dentro de estos dos casos, tenemos otras dos situaciones posibles, que se realice el experimento con la herramienta negra o con la blanca. Recordamos que la herramienta blanca solo puede acceder a los orificios 1, 2 y 7.

Los resultados de los experimentos de introducir la herramienta perpendicularmente al orificio lo tenemos en la siguiente tabla:

	Profundidad máxima (herramienta negra)	Profundidad máxima (herramienta blanca)
Orificio 1	170	170
Orificio 2	170	170
Orificio 3	130	Singularidad
Orificio 4	110	Singularidad
Orificio 5	30	Singularidad
Orificio 6	170	Singularidad
Orificio 7	170	170
Orificio 8	170	Singularidad
Orificio 9	170	Singularidad

Tabla 11 - Limite de profundidad perpendicular al orificio

Mientras que los resultados de los experimentos de introducir la herramienta perpendicularmente a la mesa son los siguientes:

	Profundidad máxima (herramienta negra)	Profundidad máxima (herramienta blanca)
Orificio 1	165	170
Orificio 2	50	170
Orificio 3	3	Singularidad
Orificio 4	110	Singularidad
Orificio 5	30	Singularidad
Orificio 6	170	Singularidad
Orificio 7	170	170
Orificio 8	170	Singularidad
Orificio 9	170	Singularidad

Tabla 12 - Límites profundidad perpendicular a la mesa

Como ya hemos visto, gracias a estas tablas el programa avisará al usuario de estos límites. Si el usuario introduce un valor mayor, al correspondiente de las tablas, el programa no le permitirá realizar la ejecución con ese valor y le informará del valor máximo posible. Al usuario le saltará el mensaje de error tantas veces como el usuario introduzca el valor mal. El programa admitirá el valor cuando se introduzca el valor máximo o un valor menor a él.

Este fenómeno solo ocurrirá la primera vez que se introduzca la herramienta perpendicularmente al orificio o a la mesa debido a la infinidad de situaciones que se pueden dar una vez que se hayan realizado más movimientos dentro del *Pelvitainer*.

6.2 Ángulos límite

Como hemos visto anteriormente, la herramienta al ser introducida por primera vez en el *Pelvitainer* puede ser movida por ángulos en el plano XZ o en el plano YZ, después de haberla introducido perpendicularmente a la mesa o al orificio.

Se han realizado experimentos para conocer el límite de ángulos en el primer desplazamiento por ángulos. Los valores que hay en las tablas que se muestran más adelante, se han obtenido realizando un gran número de simulaciones buscando la singularidad para cada caso. Se verán los ángulos límites tanto para valores positivos como para valores negativos y expresados en grados todos.

Como para cada milímetro hay diferentes singularidades para los desplazamientos en los planos XZ e YZ, se toman valores medios. Es decir, para cada longitud máxima, se dividirá ésta en tres partes de tamaño similar. En cada una de esas partes se tomará un valor medio y a partir de ese valor medio se empezará a medir los grados límite que se puede alcanzar en cada plano. Las singularidades de ese valor medio se interpretarán como las singularidades de toda esa parte.

Si, por ejemplo, como ocurre en muchos orificios, la profundidad máxima que puede alcanzar es 170, se dividirá en tres partes: la primera parte de 0 a 60 milímetros, la segunda parte de 60 a 120 milímetros y la tercera de 120 a 170 milímetros. En la primera parte tomaremos como valor medio 30 milímetros, en la segunda parte 90 milímetros y en la tercera parte 150 milímetros. En cada uno de esos valores medios empezaremos a buscar los límites y esos límites lo serán para toda la parte. Esta será la lógica a seguir para las tablas que veremos posteriormente.

Para valores menores a 70 milímetros no se tomarán 3 partes, se tomará únicamente una parte. Por lo tanto, solamente se tomará un valor medio y se realizarán los mismos experimentos para ese valor.

Por ejemplo, el valor máximo de profundidad para una entrada al orificio perpendicularmente a la mesa, en el orificio 2, con la herramienta negra, es 50 milímetros como se puede ver en la Tabla 12. Entonces el valor medio que se cogerá es 25 milímetros y se realizarán experimentos para ese valor únicamente.

Estas medidas son solo para cuando la herramienta se desplaza por primera vez en los planos XZ e YZ después de haberlas introducido perpendicularmente a la mesa o al orificio. El número de situaciones, que se puede dar al desplazar ángulos por segunda vez o realizar cambios de altura y luego otro desplazamiento por ángulos, son una gran cantidad por eso solo se muestran los límites de ángulos para cuando se introduce la herramienta por primera vez.

Igual que con la distancia máxima a la que se puede introducir la herramienta en el *Pelvitainer*, tendremos dos situaciones una con entrada perpendicular al orificio y otra perpendicular a la mesa. También, en estas dos situaciones podemos hacerlo con herramienta negra (agarre lateral) o con la blanca (agarre superior).

Inicialmente, no se enseñarán los límites para los orificios 4, 5 y 6, esto es debido a lo que vimos anteriormente, en estos orificios la orientación perpendicular a la mesa y al orificio son la misma. Se enseñarán los límites de estos después de mostrar los límites para una entrada perpendicular a la mesa.

6.2.1 Ángulos límite con entrada de la herramienta perpendicular al orificio

6.2.1.1 Ángulos límite con entrada de la herramienta negra perpendicular al orificio

Comenzaremos mostrando los límites de los grados para una entrada perpendicular al orificio. Primero tenemos los siguientes límites para un agarre lateral (herramienta negra):

	Orificio 1		Orificio 2	
	Ángulo límite XZ	Ángulo límite YZ	Ángulo límite XZ	Ángulo límite YZ
0-60	-22 a 50	-19 a 50 (80)	-34 a 50 (52)	-15 a 50 (89)
60-120	-13 a 50 (89)	-18 a 50 (89)	-39 a 50 (68)	-11 a 50 (89)
120-170	-6 a 50 (89)	-15 a 50 (89)	-34 a 50 (89)	-5 a 50 (89)

Tabla 13 - Límites de ángulos para herramienta negra con entrada perpendicular a los orificios 1 y 2

En esta Tabla 13, si nos fijamos en los ángulos límites del orificio 1, para una distancia de 0 a 60 milímetros, en el plano XZ pone '-22 a 50'. Esto se puede interpretar de forma que, el máximo ángulo negativo en el plano XZ a esta distancia, que puede alcanzar la herramienta negra, es -22 grados, mientras

que el máximo valor positivo es 50 grados. De la misma forma se interpretarían las demás casillas de esta tabla y las demás tablas que veremos a continuación.

En la misma Tabla 13 mostrada recientemente podemos ver un número en paréntesis al lado del límite del ángulo. Por ejemplo, de 60 a 120 milímetros, en el orificio 1, para el plano YZ, en valores positivos aparece '50 (89)'. Esto significa que la herramienta puede llegar a desplazarse por ángulos hasta los 89 grados sin llegar a una singularidad, pero para este valor la herramienta está en una posición que no es favorable para el *Pelvitainer*. La herramienta quedaría de la siguiente forma al tener 89 grados en este caso:

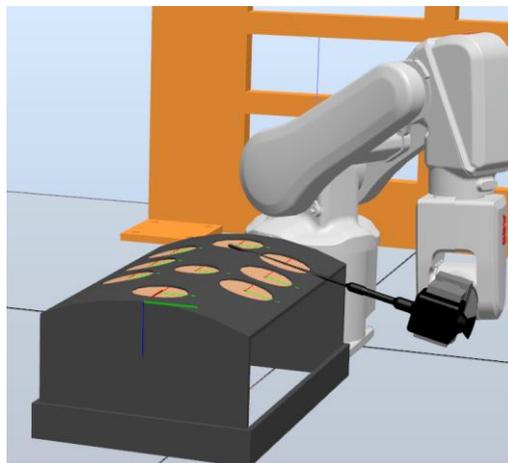


Ilustración 159 - Herramienta negra desplazada por el plano YZ a 89 grados a una distancia de 60 a 120 milímetros en orificio 1

Como podemos ver, es una posición inviable y no deseada. Sin embargo, para 50 grados queda de la siguiente forma:

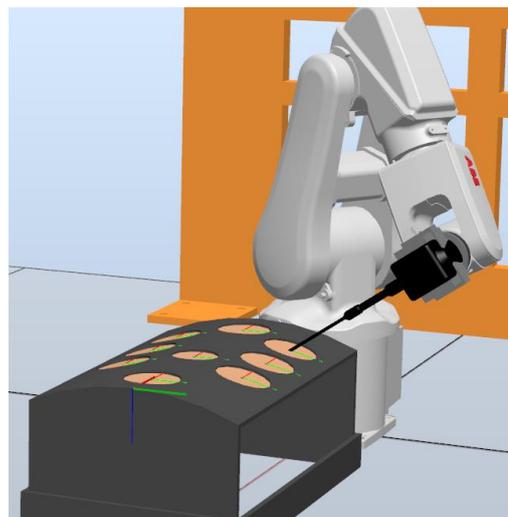


Ilustración 160 - Herramienta negra desplazada por el plano YZ a 50 grados a una distancia de 60 a 120 milímetros en orificio 1

Vemos que para 50 grados queda la herramienta en una posición más natural, que respeta la superficie del *Pelvitainer*. Con lo cual, esos 50 grados será el nuevo ángulo límite para este caso, porque para 50 grados o menos no se ve comprometida la superficie del *Pelvitainer* y para más de 50 grados sí que se ve comprometida.

Esta forma de señalar los límites, que sacrifican la seguridad del *Pelvitainer* y en su lugar poner un nuevo límite que no lo sacrifique, se utilizará en el resto de las tablas.

Orificio 3		
	Ángulo límite XZ	Ángulo límite YZ
0-43	-45 a 32	-12 a 50 (89)
44-87	-43 a 39	-9 a 50 (89)
88-130	-10 a 48	-4 a 50 (89)

Tabla 14 - Límites de ángulos para herramienta negra con entrada perpendicular al orificio 3

	Orificio 7		Orificio 8	
	Ángulo límite XZ	Ángulo límite YZ	Ángulo límite XZ	Ángulo límite YZ
0-60	-36 a 50 (70)	(-100*) -50 a 37	-41 a 46	(-100*) -50 a 46
60-120	-35 a 50 (89)	(-100*) -50 a 40	-39 a 50 (61)	(-100*) -50 a 33
120-170	-32 a 50 (89)	(-100*) -50 a 43	-36 a 50 (82)	(-100*) -50 a 35

Tabla 15 - Límites de ángulos para herramienta negra con entrada perpendicular a los orificios 7 y 8

En esta Tabla 15, nos podemos fijar que para los ángulos límite negativos del plano YZ, en el paréntesis tenemos un asterisco. Esto significa que la herramienta puede desplazarse mucho más de -100 grados, pero para valores más negativos de -100 grados la herramienta alcanza posiciones que no son viables.

Para valores menores que -100 grados tendríamos posiciones más impracticables y menos seguras que la posición de la herramienta de la Ilustración 159. Estos valores límites por debajo de -100 grados al ser tan irrelevantes, no hay necesidad de conocerlos. Este asterisco aparecerá, en las demás tablas cuando se dé esta situación y tendrán el mismo significado.

Orificio 9		
	Ángulo límite XZ	Ángulo límite YZ
0-60	-45 a 27	(-100*) -50 a 30
60-120	-43 a 39	(-100*) -50 a 29
120-170	-(41) -29 a 53	(-100*) -50 a 28

Tabla 16 - Límites de ángulos para herramienta negra con entrada perpendicular al orificio 9

6.2.1.2 Ángulos límite con entrada de la herramienta blanca perpendicular al orificio

Ahora, para la herramienta blanca con agarre superior tenemos los siguientes límites:

Orificio 1		
	Ángulo límite XZ	Ángulo límite YZ
0-60	(-100*) -50 a 15	(-97) -50 a 39
60-120	(-100*) -50 a 27	(-100*) -50 a 50 (61)
120-170	(-52) -50 a 40	(-100*) -50 a 50 (78)

Tabla 17 - Límites de ángulos para herramienta blanca con entrada perpendicular al orificio 1

Orificio 2		
	Angulo limite XZ	Angulo limite YZ
0-60	(-68) -50 a 4	(-85) -50 a 21
60-120	(-62) -50 a 13	(-100*) -50 a 48
120-180	(-62) -50 a 22	(-100*) -50 a 50 (69)

Tabla 18 - Límites de ángulos para herramienta blanca con entrada perpendicular al orificio 2

Orificio 7		
	Ángulo límite XZ	Ángulo límite YZ
0-60	(-100*) -50 a 9	-15 a 50 (89)
60-120	(-100*) -50 a 21	-34 a 50 (89)
120-170	(-100*) -50 a 33	-53 a 50 (89)

Tabla 19 - Límites de ángulos para herramienta blanca con entrada perpendicular al orificio 7

6.2.2 Ángulos límite con entrada de la herramienta perpendicular a la mesa

Visto ya los límites para la herramienta introducida perpendicularmente al orificio, pasamos a ver los límites para cuando la herramienta se pueda introducir perpendicularmente a la mesa.

6.2.2.1 Ángulos límite con entrada de la herramienta negra perpendicular a la mesa

Comenzamos viendo los límites para la herramienta negra con un agarre lateral:

	Orificio 1		Orificio 2	
	Ángulo límite XZ	Ángulo límite YZ	Ángulo límite XZ	Ángulo límite YZ
0-50	-16 a 50 (78)	-19 a 50 (89)	-10 a 51	-1 a 50 (89)
51-106	-9 a 50 (89)	-4 a 50 (89)		
107-160	-10 a 50 (89)	-2 a 50 (89)		

Tabla 20 - Límites de ángulos para herramienta negra con entrada perpendicular a la mesa en orificio 1 y 2

Recordamos que el orificio 2, para esta orientación de entrada con la herramienta negra, solo se puede introducir 50 milímetros y por eso no hay límites de ángulos de 50 a 160 milímetros en este orificio. No se muestra el orificio 3 porque, como hemos dicho, el brazo robótico alcanza una singularidad al introducir la herramienta perpendicular en este orificio.

	Orificio 7		Orificio 8	
	Ángulo límite XZ	Ángulo límite YZ	Ángulo límite XZ	Ángulo límite YZ
0-60	-25 a 50 (74)	(-100*) -50 a 22	-37 a 49	(-100*) -50 a 17
60-120	-19 a 50 (89)	(-100*) -50 a 25	-39 a 50 (64)	(-100*) -50 a 18
120-170	-17 a 50 (89)	(-100*) -50 a 35	-35 a 50 (84)	(-100*) -50 a 19

Tabla 21 - Límites de ángulos para herramienta negra con entrada perpendicular a la mesa en los orificios 7 y 8

	Orificio 9	
	Ángulo límite XZ	Ángulo límite YZ
0-60	-45 a 30	(-100*) -50 a 14
60-120	-42 a 41	(-100*) -50 a 13
120-170	(-40) -30 a 55	(-100*) -50 a 12

Tabla 22 - Límites de ángulos para herramienta negra con entrada perpendicular a la mesa en orificio 9

6.2.2.2 Ángulos límite con entrada de la herramienta blanca perpendicular a la mesa

Habiendo terminado de mostrar los límites para la herramienta negra, mostramos los límites para la herramienta blanca.

Orificio 1		
	Ángulo límite XZ	Ángulo límite YZ
0-60	(-46) -40 a 17	(-82) -50 a 53
60-120	(-49) -40 a 29	(-99) -50 a 50 (75)
120-170	-43 a 41	(-100*) -50 a 50 (89)

Tabla 23 - Límites de ángulos para herramienta blanca con entrada perpendicular a la mesa en el orificio 1

Orificio 2		
	Angulo limite XZ	Angulo limite YZ
0-60	(-55) -50 a 6	(-71) -50 a 35
60-120	(-52) -50 a 14	(-90) -50 a 50 (62)
120-180	(-53) -50 a 23	(-100*) -50 a 50 (69)

Tabla 24 - Límites de ángulos para herramienta blanca con entrada perpendicular a la mesa en el orificio 2

Orificio 7		
	Ángulo límite XZ	Ángulo límite YZ
0-60	(-100*) -50 a 16	-31 a (89) 50
60-120	(-60) -50 a 27	-34 a (89) 50
120-170	(-50) -53 a 39	(-69) -50 a 50 (89)

Tabla 25 - Límites de ángulos para herramienta blanca con entrada perpendicular a la mesa en orificio 7

6.2.3 Ángulos límite de los orificios 4, 5 y 6

Como se ha mencionado anteriormente en los orificios 4, 5 y 6 no hay distinción entre la orientación de entrada perpendicular a la mesa y al orificio. Al tener esta característica en particular no se les ha incluido en los apartados de perpendicular a la mesa o al orificio.

Como con la herramienta blanca solo podemos acceder a los orificios 1, 2 y 7, los límites que se van a mostrar son hallados con la herramienta negra.

A continuación, se muestran los límites de los ángulos para la herramienta introducida a diferentes distancias:

Orificio 4		
	Ángulo límite XZ	Ángulo límite YZ
0-37	-17 a 34	(-100*) -50 a 50 (89)
37-74	-10 a 39	(-100*) -50 a 50 (89)
74-110	-3 a 44	(-100*) -50 a 50 (89)

Tabla 26 - Límites de ángulos para herramienta negra en orificio 4

Orificio 5		
	Ángulo límite XZ	Ángulo límite YZ
0-30	25 a -36	(89) 50 a (-100*) -50

Tabla 27 - Límites de ángulos para herramienta negra en orificio 5

Orificio 6		
	Ángulo límite XZ	Ángulo límite YZ
0-60	12 a -49	(89) 50 a (-100*) -50
60-120	18 a -48	(89) 50 a (-100*) -50
120-170	25 a -49	(89) 50 a (-100*) -50

Tabla 28 - Límites de ángulos para herramienta negra en orificio 6

6.2.4 Entrada con ángulo

Los límites de ángulos mostrados recientemente eran para las opciones de perpendicular a la mesa y al orificio. En este apartado, tendremos límites de ángulos para la opción de entrada con ángulo.

Para la opción entrada con ángulo, no se han realizado las medidas de los valores máximos en los planos XZ e YZ después de haberse introducido la herramienta, porque tenemos un gran número de casos posibles.

Este gran número de casos se debe a que la herramienta puede entrar con una gran variedad de ángulos en el plano XZ o YZ aplicándole una rotación o no. Además, para esta cantidad de orientaciones posibles, la herramienta puede entrar con un rango notorio de milímetros posibles, de esta forma queda la infinidad de casos mencionada.

Entonces, se ha decidido obtener los límites de ángulos para el pivoteo inicial en los planos XZ e YZ. En otras palabras, vamos a buscar los límites de valores para el menú, de la opción entrada con ángulos, de la Ilustración 103.

6.2.3.1 Herramienta negra

Comenzaremos mostrando los límites para la herramienta negra:

	Orificio 1		Orificio 2	
	Ángulo límite XZ	Ángulo límite YZ	Ángulo límite XZ	Ángulo límite YZ
Orificio	-27 a 50 (68)	-19 a 50 (100*)	-35 a 45	-16 a 50 (100*)
Mesa	-27 a 50 (68)	-5 a 50 (100*)	-27 a 45	-2 a 50 (100*)

Tabla 29 - Límites de ángulos de pivoteo para herramienta negra en los orificios 1 y 2

Orificio 3		
	Ángulo límite XZ	Ángulo límite YZ
Orificio	-46 a 28	-14 a 50 (100*)
Mesa	0 a 28	0 a 50 (100*)

Tabla 30 - Límites de ángulos de pivoteo para herramienta negra en el orificio 3

Orificio 4		Orificio 5	
Ángulo límite XZ	Ángulo límite YZ	Ángulo límite XZ	Ángulo límite YZ
-20 a 32	(-100*) -50 a 50 (100*)	-35 a 22	(-100*) -50 a 50 (100*)

Tabla 31 - Límites de ángulos de pivoteo para herramienta negra en los orificios 4 y 5

Orificio 6	
Ángulo límite XZ	Ángulo límite YZ
-50 a 9	(-84) -50 a 50 (100*)

Tabla 32 - Límites de ángulos de pivoteo para herramienta negra en el orificio 6

	Orificio 7		Orificio 8	
	Ángulo límite XZ	Ángulo límite YZ	Ángulo límite XZ	Ángulo límite YZ
Orificio	-36 a 60	(-100*) -50 a 36	-42 a 39	(-100*) -50 a 33
Mesa	-29 a 65	(-100*) -50 a 21	-36 a 42	(-100*) -50 a 17

Tabla 33 - Límites de ángulos de pivoteo para herramienta negra en los orificios 7 y 8

Orificio 9		
	Ángulo límite XZ	Ángulo límite YZ
Orificio	-46 a 23	(-80) -50 a 30
Mesa	-47 a 26	(-96) -50 a 15

Tabla 34 - Límites de ángulos de pivoteo para herramienta negra en el orificio 9

6.2.3.2 Herramienta blanca

De la misma forma que en los casos anteriores, estudiaremos los límites del pivoteo en el plano XZ y el plano YZ para el caso de la herramienta blanca con un agarre superior:

	Orificio 1		Orificio 2	
	Ángulo límite XZ	Ángulo límite YZ	Ángulo límite XZ	Ángulo límite YZ
Mesa	-42 a 12	(-100*) -70 a 40	(-57) -50 a 2	(-56) -50 a 18
Orificio	(-100*) -40 a 10	(-84) -50 a 26	(-100*) -50 a 0	-70 (-50*) a 4

Tabla 35 - Límites de ángulos de pivoteo para herramienta blanca en los orificios 1 y 2

Orificio 7		
	Ángulo límite XZ	Ángulo límite YZ
Mesa	(-100*) -50 a 10	-22 a 50 (100*)
Orificio	(-100*) -50 a 3	-6 a 50 (100*)

Tabla 36 - Límites de ángulos de pivoteo para herramienta blanca en el orificio 7

7. Conclusiones y líneas futuras

Realizar este proyecto ha resultado ser una experiencia muy enriquecedora.

Se han ampliado los conocimientos en *RobotStudio*. Dentro de *RobotStudio* tenemos *Rapid* que es el lenguaje de programación de los robots *ABB*. He aprendido a usar nuevas funciones que ofrece este programa, y a utilizar y optimizar mejor las que ya se conocía. A su vez, *RobotStudio* tiene un entorno 3D en el que, como hemos visto, se puede manipular los objetos 3D, crear *robtargets*, crear objetos de trabajo, etc. Se ha logrado tener un mejor manejo de este entorno 3D, y se ha aprendido a usar de una forma más eficiente gracias a las horas que se han dedicado.

También, como hemos visto, se ha utilizado la aplicación *Autodesk Inventor Professional 2022*, donde se han creado todos los componentes 3D de la estación, exceptuando el brazo robótico que ya venía por defecto. Esta aplicación ofrece una gran cantidad de opciones, tanto para como los objetos 3D, como para los planos. Durante la creación de estos se han ido descubriendo nuevas opciones de la aplicación, que se han puesto en uso, y a su vez, se ha mejorado el manejo de las opciones que ya se conocían.

A la hora de realizar este proyecto se han contemplado una gran variedad de opciones que se han ido implementando en el programa. La creación de las opciones de la aplicación, que se han indicado en esta memoria, ha supuesto un gran esfuerzo. Además, hacer que todas estas opciones funcionen, coordinadamente y sin que se obstaculicen entre ellas durante la simulación, ha resultado ser bastante laborioso.

A pesar de que este programa ofrezca varias opciones se le puede añadir todavía más alternativas. Dejando el techo de posibles opciones o funciones por añadir mucho más alto.

Como línea futura de este proyecto, cabría la posibilidad de pasar de la simulación al “mundo físico”, mediante la impresión 3D de las herramientas modeladas, y con el *Pelvitainer* y el brazo robótico *ABB IRB 120*, que se encuentra en *ITAP*. La aplicación desarrollada, podría ser utilizada en este entorno físico que podría ser creado.

8. Bibliografía

- ABB. (s.f.). Recuperado el Abril de 2022, de <http://personal.biada.org/~jhorriillo/INSTRUCCIONS%20RAPID.pdf>
- ABB. (s.f.). ABB. Recuperado el Marzo de 2022, de <https://new.abb.com/products/robotics/es/robotstudio>
- ABB. (s.f.). ABB. Recuperado el Marzo de 2022, de <https://new.abb.com/products/robotics/es/robots-industriales/irb-120>
- ABB Robotics. (27 de Mayo de 2021). *Especificaciones IRB 120*. Recuperado el Marzo de 2022, de <https://abb.sluzba.cz/Pages/Public/IRC5RoboticsDocumentationRW6/Robots/Articulated%20Robots/IRB%20120/es/3HAC035960-005.pdf>
- Abex. (s.f.). Abex. Recuperado el Abril de 2022, de <https://www.abexsl.es/es/sistema-robotico-da-vinci/ques#:~:text=Nacido%20en%20el%20seno%20de,lanz%C3%B3%20al%20mercado%20en%201999.>
- American Cancer Society. (14 de Enero de 2019). *American Cancer Society*. Recuperado el Marzo de 2022, de <https://www.cancer.org/es/tratamiento/como-comprender-su-diagnostico/pruebas/endoscopia/laparoscopia.html>
- Autodesk. (s.f.). *Productos de Autodesk*. Recuperado el Marzo de 2022, de <https://www.autodesk.es/products/inventor/overview?term=1-YEAR&tab=subscription>
- Calvo, A. P. (11 de Marzo de 2019). *Dr. Alberto Parajó Calvo*. Recuperado el Marzo de 2022, de https://www.albertoparajo.com/5-ventajas-de-la-cirugia-laparoscopica_fb24512.html
- Charrière, C. &. (s.f.). *Science Museum Group*. Recuperado el Marzo de 2022, de <https://collection.sciencemuseumgroup.org.uk/objects/co91100/desormeaux-type-endoscope-paris-france-1861-1880-endoscope>
- Clínica Universidad de Navarra. (s.f.). *Clínica Universidad de Navarra*. Recuperado el Marzo de 2022, de <https://www.cun.es/enfermedades-tratamientos/tratamientos/cirugia-laparoscopica>
- Direct Industry*. (s.f.). Recuperado el Marzo de 2022, de <https://www.directindustry.es/prod/abb-robotics/product-30265-565867.html>
- Etimologías de Chile*. (28 de Marzo de 2022). Recuperado el Marzo de 2022, de <http://etimologias.dechile.net/?laparo#:~:text=La%20voz%20de%20origen%20griego,de%20la%20pelvis%20o%20caderas%22.>
- Eurobots*. (s.f.). Recuperado el Marzo de 2022, de <https://www.eurobots.es/irb-120-es.html>
- García, L. (4 de Febrero de 2022). *Empresas con Salud*. Recuperado el Abril de 2022, de <https://www.consalud.es/ecs salud/ehealth/medtronic-anuncia-procedimiento-europa-sistema-cirugia-asistida-robot->

hugo_109465_102.html#:~:text=El%20sistema%20Hugo%20RAS%2C%20la,de%20procedimientos%20de%20tejidos%20blandos.

Guía del cirujano. (s.f.). Recuperado el Marzo de 2022, de <https://guiadelcirujano.com/historia-de-la-laparoscopia/>

Hm Hospitales. (14 de Octubre de 2021). *Hm hospitales*. Recuperado el Abril de 2022, de <https://www.hmhospitales.com/prensa/notas-de-prensa/hugo-el-sistema-cirugia-asistida-por-robot-de-medtronic>

Infosalus. (4 de Noviembre de 2010). Recuperado el Marzo de 2022, de <https://www.infosalus.com/actualidad/noticia-80-cirugias-espana-realiza-ya-laparoscopia-20101104164038.html>

Karl Storz. (s.f.). *Karl Storz - Endoscope*. Recuperado el Marzo de 2022, de <https://www.karlstorz.com/es/es/documentation-vet.htm>

Litynski, G. (14 de Mayo de 2021). *MPMS*. Recuperado el Marzo de 2022, de <https://www.pamsm.org/post/the-invention-of-the-laparoscopy-1901>

Microsoft. (12 de Septiembre de 2016). *Microsoft App Store*. Recuperado el Marzo de 2022, de <https://www.microsoft.com/en-us/p/robotstudio-holographic/9nclpblbq14h#activetab=pivot:overviewtab>

Olympus. (s.f.). *Medical Expo*. Recuperado el Marzo de 2022, de <https://www.medicalexpo.es/prod/olympus-medical-europa/product-69587-554537.html>

Pérez, A. (23 de Febrero de 2022). *Redacción médica*. Recuperado el Abril de 2022, de <https://www.redaccionmedica.com/secciones/tecnologia/-el-robot-hugo-es-la-prueba-de-que-la-cirugia-esta-en-constante-evolucion-5961>

Reeve, A. (14 de Septiembre de 1998). *Alamy*. Recuperado el Abril de 2022, de <https://www.alamy.es/la-laparoscopia-vista-de-un-cirujano-utilizando-un-laparoscopio-durante-la-cirugia-un-laparoscopio-es-un-tipo-de-endoscopio-y-puede-insertarse-en-el-abdomen-del-paciente-image335386516.html>

Reynolds, W. (Marzo de 2001). *ResearchGate*. Recuperado el Marzo de 2022, de https://www.researchgate.net/figure/Prof-Dr-Med-Erich-Muehe-World-Champion-Cyclist-for-Physicians-and-Pharmacists-in-1987_fig3_12030409

Shen, S. (s.f.). *Dr. Industrial Group*. Recuperado el Marzo de 2022, de <https://es.drindustrialrobot.com/abb-irb-120-robot/>

STORZ. (s.f.). *Karl Storz - Endoscope*. Recuperado el Marzo de 2022, de <https://www.karlstorz.com/es/es/index.htm>

Storz, Karl. (2020). Soluciones de formación para uso interno. Aurangabad: EndoWorld. Recuperado el Marzo de 2022

Tagumédica. (s.f.). Recuperado el Marzo de 2022, de <https://www.tagumédica.com/producto/agujas-de-veress/>

Universidad de Valladolid. (s.f.). *Escuela de Ingenierías Industriales*. Recuperado el Abril de 2022, de eii.uva.es

Urología aplicada. (s.f.). Recuperado el Marzo de 2022, de <https://urologiaaplicada.com/cirugia-robotica/>

Wikimedia. (22 de Septiembre de 2016). Recuperado el Marzo de 2022, de https://commons.wikimedia.org/wiki/File:Endoscope_de_Desormeaux.jpg

- Wikipedia*. (21 de Abril de 2020). Recuperado el Marzo de 2022, de https://es.frwiki.wiki/wiki/Philippe_Mouret
- Zieve, D. G. (19 de Septiembre de 2021). *MedlinePlus*. Recuperado el Marzo de 2022, de https://medlineplus.gov/spanish/ency/esp_presentations/100166_2.htm
- Ruiz, A. G., Rodríguez, L. G., & García, J. C. (2016). Evolución histórica de la cirugía laparoscópica. *Revista Mexicana de Cirugía Endoscópica*, 17(2), 93-106. Obtenido de <https://www.medigraphic.com/pdfs/endosco/ce-2016/ce162i.pdf>
- Gutiérrez-Banda, C., Arellano-López, P. R., González-Ruiz, V., Bandeh-Moghaddam, H., De Santos-González, L. R., Valadez-Caballero, D., ... & Núñez-Valenzuela, D. (2016). El arte e importancia de la enseñanza en cirugía laparoscópica. *Revista Mexicana de Cirugía Endoscópica*, 17(2), 88-92. Obtenido de <https://www.medigraphic.com/pdfs/endosco/ce-2016/ce162h.pdf>
- Martínez, C. D. J. P. (2014). Historia de la cirugía laparoscópica: particularidades de su introducción y desarrollo en Cuba. *Universitas Médica*, 55(2), 200-2010.
DOI: <https://doi.org/10.11144/Javeriana.umed55-2.hclp>
- Ramos, C. M. (2006). Robótica y cirugía laparoscópica. *Cirugía Española*, 80(4), 189-194. DOI: [10.1016/S0009-739X\(06\)70956-3](https://doi.org/10.1016/S0009-739X(06)70956-3)

9. Índices de las ilustraciones y de las tablas

9.1 Índice de ilustraciones

Ilustración 1 - Robot ABB IRB 120 (ABB, s.f.) _____	1
Ilustración 2 - Imagen del Pelvitainer tomado en el departamento de ITAP __	2
Ilustración 3 - Estación de trabajo de RobotStudio_____	3
Ilustración 4 - Intervención quirúrgica utilizando cirugía laparoscópica (Zieve, 2021) _____	5
Ilustración 5 - Imagen de un laparoscopio con el visor marcado (Olympus, s.f.) _____	5
Ilustración 6 - Visión de lo que capta la cámara a través de una pantalla (Karl Storz, s.f.) _____	6
Ilustración 7 -Ejemplo de profesional usando el visor (Reeve, 1998) _____	6
Ilustración 8 - El endoscopio de Desormeaux (Charrière, s.f.) (Antonio García Ruiz, 2016) _____	7
Ilustración 9 - George Kelling y su laparoscopio diseñado (Antonio García Ruiz, 2016) _____	8
Ilustración 10 - Aguja de Veress (Tagumedica, s.f.) _____	9
Ilustración 11 - Kurt Semm y su invento, el insuflador electrónico de neumoperitoneo (Antonio García Ruiz, 2016)_____	9
Ilustración 12 - Imagen de Philippe Mouret (Wikipedia, 2020) _____	10
Ilustración 13 - Eric Mühe y su "galloscopio" (Reynolds, 2001) (Antonio García Ruiz, 2016) _____	10
Ilustración 14 - Barry McKernan a la izquierda y Eddie Joe Reddick a la derecha (Antonio García Ruiz, 2016)_____	11
Ilustración 15 - Sistema robótico da Vinci (Urología aplicada, s.f.) _____	13
Ilustración 16 - Hugo RAS de Medtronic (Hm Hospitales, 2021)_____	14
Ilustración 17 - Ejes y dimensiones del ABB IRB 120 (ABB Robotics, 2021) _____	20
Ilustración 18 - Área de trabajo del IRB 120 (ABB Robotics, 2021) _____	21
Ilustración 19 - Logo de la aplicación de Autodesk Invento (Autodesk, s.f.)_	22
Ilustración 20 - Pelvitainer modelado en 3D en Autodesk Inventor _____	22
Ilustración 21 - Dimensiones del Pelvitainer _____	23
Ilustración 22 - Pelvitainer sin las paredes laterales_____	23
Ilustración 23 - Herramienta blanca y herramienta negra que se utilizará en la aplicación _____	24
Ilustración 24 - Dimensiones de la herramienta blanca_____	25
Ilustración 25 - Dimensiones de la herramienta negra _____	25
Ilustración 26 - Visor del laparoscopio emulado en la herramienta 3D____	26
Ilustración 27 - Apoyo para las herramientas negras _____	27
Ilustración 28 - Dimensiones del apoyo para las herramientas negras _____	28
Ilustración 29 - Apoyo utilizado para las herramientas blancas _____	28
Ilustración 30 - Dimensiones para el apoyo de herramientas blancas _____	29
Ilustración 31 - Modelado en 3D del vallado_____	30
Ilustración 32 - Siglas de Escuela de Ingenierías Industriales aplicada al modelo 3D (Universidad de Valladolid, s.f.)_____	30
Ilustración 33 - Dimensiones del vallado _____	31

Ilustración 34 - Logo de la aplicación de RobotStudio (Microsoft, 2016) ____	32
Ilustración 35- Estación resultante al implementar los objetos modelados en 3D _____	32
Ilustración 36 - Mecanismos y componentes en la estación _____	33
Ilustración 37 - Objetos de trabajo y puntos de la estación _____	34
Ilustración 38 - Color para cada eje _____	34
Ilustración 39 - Ubicación del sistema de referencia "Pelv2" asociado al Pelvitainer _____	35
Ilustración 40 - Enumeración de los orificios del Pelvitainer _____	36
Ilustración 41 - Herramientas utilizadas en la simulación _____	37
Ilustración 42 - Wobjdata del soporte de las herramientas negras y robtarget de la herramienta negra _____	38
Ilustración 43 - Agarre lateral con la herramienta negra _____	38
Ilustración 44- Opción de crear un tooldata en la estación de RobotStudio _	39
Ilustración 45 - Parámetros a preguntar en la opción 'Crear dato de herramienta' _____	39
Ilustración 46 - Tooldata de la herramienta negra en la estación _____	40
Ilustración 47 - Tooldata y robtarget superpuestos en el orificio 2 _____	40
Ilustración 48 - Wobjdata del soporte de las herramientas negras y robtarget de la herramienta negra _____	41
Ilustración 49 - Agarre lateral con la herramienta blanca _____	42
Ilustración 50 -Tooldata de la herramienta blanca en la estación _____	43
Ilustración 51 - Imagen de la posición y orientación de los orificios del Pelvitainer en Rapid _____	44
Ilustración 52 - Array de robtargets auxiliar _____	45
Ilustración 53 - Jointtargets utilizados en el programa _____	46
Ilustración 54 - Algunos de los tooldata utilizados _____	46
Ilustración 55 - Otros robtargets usados en el programa _____	47
Ilustración 56 - Módulos del programa _____	50
Ilustración 57 - Código del módulo main_simu _____	51
Ilustración 58 - Código que se utilizará en caso de seleccionar 'Herramienta' en el módulo Opciones _____	52
Ilustración 59 - Fragmento de código del módulo Aplicacion, que muestra los posibles menús que pueden aparecer _____	53
Ilustración 60 - Código en caso de seleccionar que la herramienta entre perpendicularmente a la mesa o al orificio _____	54
Ilustración 61 - Código en caso de seleccionar que la herramienta entre con ángulo _____	54
Ilustración 62 - Código para la opción del cambio de orientación, en el módulo Dentro _____	55
Ilustración 63 - Código para la opción del cambio de profundidad, en el módulo Dentro _____	56
Ilustración 64 - Código para la opción 'Salir', en el módulo Fuera _____	57
Ilustración 65 - Bucle encargado de introducir la herramienta perpendicularmente al orificio _____	58
Ilustración 66 - Bucle encargado de introducir la herramienta perpendicularmente a la mesa _____	59

Ilustración 67 - Bucle encargado de cambiar la orientación de la herramienta	60
Ilustración 68 - Rotación en Y en robtarget de orificio 2	61
Ilustración 69 - Pose1 desplazado 80 milímetros en el eje Z	62
Ilustración 70 - Código encargado del pivoteo	63
Ilustración 71 - Bucle del módulo EntradaConÁngulo	64
Ilustración 72 - Fragmento de código del módulo Permitir_validos	65
Ilustración 73 - Fragmento de código de la función Límite_profundidad	66
Ilustración 74 - Parámetro que se devolverá al finalizar la función Límite_profundidad	66
Ilustración 75 - Código del menú que sale por pantalla al entrar en la opción Demo	67
Ilustración 76 - Uso de la función TEST en el código de la demo	67
Ilustración 77 - Fragmento del código del módulo DemoBlanca	69
Ilustración 78 - Bucle encargado de trazar la circunferencia	70
Ilustración 79 - Bucle encargado de realizar el barrido	71
Ilustración 80 - Código del módulo Salir_Pelvitainer	74
Ilustración 81 - Menú inicial en el diagrama	78
Ilustración 82 - Menú principal	78
Ilustración 83 - Menú herramienta en el diagrama	80
Ilustración 84 - Menú herramienta	80
Ilustración 85 - Mensaje durante el cambio de herramienta	81
Ilustración 86 - Menú después de seleccionar cambio de herramienta.	81
Ilustración 87 - Petición al usuario después de seleccionar 'Aplicación'	82
Ilustración 88 - Mensaje de error con herramienta blanca al seleccionar orificio	82
Ilustración 89- Mensaje de error con herramienta negra al seleccionar orificio	82
Ilustración 90 - Menú orificio en el diagrama	83
Ilustración 91 - Menú inicial para comandar movimientos dentro del Pelvitainer	83
Ilustración 92- Ejemplo de orientación del eje z de la mesa	84
Ilustración 93 - Petición al usuario de distancia para introducir la herramienta.	84
Ilustración 94 - Ejemplo de herramienta introducida perpendicular a la mesa	85
Ilustración 95 - Eje z del robtarget del orificio 1	85
Ilustración 96 - Ejemplo de simulación de la opción perpendicular al orificio	86
Ilustración 97 - Visualización del final de la herramienta en la opción perpendicular al orificio	86
Ilustración 98 - Sistemas de referencia para perpendicular a la mesa y al orificio superpuestos	87
Ilustración 99 - Menú al dirigir la herramienta negra al orificio 3	87
Ilustración 100 - Menú para los orificios 4, 5 y 6	88
Ilustración 101 - Ejemplo para entrada de herramienta 'Sin ángulo'	88
Ilustración 102 - Mensaje por pantalla al introducir la herramienta negra 170 milímetros en el orificio 4	89
Ilustración 103 - Primera elección de la opción con ángulo	90

Ilustración 104 - Plano XZ e YZ sobre los ejes de un rotarget _____	91
Ilustración 105 - Petición de grados para el plano seleccionado _____	91
Ilustración 106 - Ejemplo pivoteo en el plano XZ _____	92
Ilustración 107 - Ejemplo pivoteo en el plano YZ _____	92
Ilustración 108 - Pregunta al usuario si quiere realizar rotación _____	93
Ilustración 109 - Pregunta de con cuántos grados realizar la rotación _____	93
Ilustración 110 - Ejemplo de rotación completa _____	94
Ilustración 111 - Posición inicial con herramienta inclinada 12 grados _____	94
Ilustración 112 - Rotación de -40 grados aplicada a la ilustración 38 _____	95
Ilustración 113 - Última petición al usuario de la opción con ángulo _____	96
Ilustración 114 - Información en la pantalla de las opciones seleccionadas _____	96
Ilustración 115 - Posición inicial del ejemplo en orificio 7 _____	97
Ilustración 116 - Movimientos a realizar en la ejecución ejemplo _____	97
Ilustración 117 - Pivoteo de -15 grados en el plano YZ _____	98
Ilustración 118 - Rotación de -60 grados a partir de la posición de la Ilustración 117. _____	98
Ilustración 119 - Resultado de la ejecución para las ordenes de la ilustración 43 _____	99
Ilustración 120 - Ejemplo de brazo robótico en posición de punto base para herramienta blanca _____	100
Ilustración 121 - Menú después de seleccionar 'salir' en el menú de la ilustración 21 _____	100
Ilustración 122 - Menú movimiento en el diagrama _____	101
Ilustración 123 - Menú tras haber introducido la herramienta _____	101
Ilustración 124 - Cuestión por pantalla tras seleccionar 'Por ángulos' _____	102
Ilustración 125 - Petición de grados para el plano seleccionado en la opción 'Por ángulos' _____	102
Ilustración 126 - Parte de la herramienta que no entra en contacto con la superficie del Pelvitainer _____	103
Ilustración 127 - Menú por pantalla al seleccionar por segunda vez 'Por ángulos' _____	104
Ilustración 128 - Herramienta introducida 110 milímetros perpendicularmente al orificio _____	105
Ilustración 129 - Herramienta desplazada -12 grados en el plano YZ a partir de la Ilustración 128 _____	106
Ilustración 130 - Rotación de -50 grados aplicada a la Ilustración 129 _____	107
Ilustración 131 - Desplazamiento de -30 grados en el plano XZ de la Ilustración 130 _____	108
Ilustración 132 - Cuestión al usuario después de seleccionar 'Cambiar h' _____	109
Ilustración 133- Caso inicial con herramienta introducida 100 milímetros perpendicularmente al orificio _____	110
Ilustración 134 - Interfaz de usuario para ejemplo de la opción cambio de altura _____	110
Ilustración 135 - Herramienta de la ilustración 51 subida 40 milímetros _____	111
Ilustración 136 - Zoom del final de la herramienta de la ilustración 53 _____	111
Ilustración 137 - Herramienta introducida 100 milímetros más respecto a la Ilustración 135 _____	112

Ilustración 138 - Características del movimiento entrada con ángulo del ejemplo_____	113
Ilustración 139 - Orientación previa de la herramienta al entrar en el Pelvitainer_____	113
Ilustración 140 - Entrada con ángulo completa del ejemplo _____	114
Ilustración 141 - Herramienta introducida 40 milímetros más respecto a la Ilustración 140_____	115
Ilustración 142 - Desplazamiento en el plano YZ de -20 grados a partir de la posición de la Ilustración 141 _____	115
Ilustración 143 - Rotación de 40 grados aplicada a la Ilustración 142 ____	116
Ilustración 144 - Final de la herramienta en orificio 2 durante la ejecución de la opción 'Salir'_____	117
Ilustración 145 - Menú demo en el diagrama _____	118
Ilustración 146 - Menú de la demo _____	118
Ilustración 147 - Ejemplo de opción 1 de la demo _____	119
Ilustración 148- Mensaje por pantalla del movimiento de la ilustración 3_	120
Ilustración 149- Ejemplo de opción 2 de la demo _____	121
Ilustración 150- Mensaje por pantalla del movimiento de la ilustración 5_	121
Ilustración 151 - Rastro que recorrerá el TCP en la opción de barrido ____	122
Ilustración 152 - Captura durante la simulación de la opción de barrido de la demo_____	122
Ilustración 153 - Imagen detallada del rastro del TCP en el barrido _____	123
Ilustración 154 - Mensaje por pantalla en la opción de barrido de la demo	123
Ilustración 155 - Rastro para 30 grados de barrido _____	123
Ilustración 156 - Captura durante la simulación de la opción circunferencia _____	124
Ilustración 157 - Imagen del rastro del TCP en la opción circunferencia __	124
Ilustración 158 - Mensaje por pantalla durante la opción de circunferencia	124
Ilustración 159 - Herramienta negra desplazada por el plano YZ a 89 grados a una distancia de 60 a 120 milímetros en orificio 1 _____	130
Ilustración 160 - Herramienta negra desplazada por el plano YZ a 50 grados a una distancia de 60 a 120 milímetros en orificio 1 _____	130

9.2 Índice de tablas

Tabla 1 - Características del brazo robótico ABB IRB 120 (Direct Industry, s.f.)	19
Tabla 2 - Velocidad y rango de trabajo de cada eje del IRB 120 (Eurobots, s.f.)	20
Tabla 3 - Coordenadas del sistema de referencia “Pelv2” asociado al Pelvitainer	35
Tabla 4 - Posición y orientación de los orificios del Pelvitainer	36
Tabla 5 - Coordenadas del S.R. “Sopor_Ng” asociado al soporte de herramientas negras	37
Tabla 6 - Posición y orientación del punto de la herramienta negra	38
Tabla 7 - Posición y orientación del tooldata de la herramienta negra	39
Tabla 8 - Coordenadas del S.R. “Sopor_BI” asociado al soporte de herramientas blancas	41
Tabla 9 - Posición y orientación de la herramienta blanca	41
Tabla 10 - Posición y orientación del tooldata de la herramienta blanca.	42
Tabla 11 - Límite de profundidad perpendicular al orificio	126
Tabla 12 - Límites profundidad perpendicular a la mesa	127
Tabla 13 - Límites de ángulos para herramienta negra con entrada perpendicular a los orificios 1 y 2	129
Tabla 14 - Límites de ángulos para herramienta negra con entrada perpendicular al orificio 3	131
Tabla 15 - Límites de ángulos para herramienta negra con entrada perpendicular a los orificios 7 y 8	131
Tabla 16 - Límites de ángulos para herramienta negra con entrada perpendicular al orificio 9	132
Tabla 17 - Límites de ángulos para herramienta blanca con entrada perpendicular al orificio 1	132
Tabla 18 - Límites de ángulos para herramienta blanca con entrada perpendicular al orificio 2	132
Tabla 19 - Límites de ángulos para herramienta blanca con entrada perpendicular al orificio 7	132
Tabla 20 - Límites de ángulos para herramienta negra con entrada perpendicular a la mesa en orificio 1 y 2	133
Tabla 21 - Límites de ángulos para herramienta negra con entrada perpendicular a la mesa en los orificios 7 y 8	133
Tabla 22 - Límites de ángulos para herramienta negra con entrada perpendicular a la mesa en orificio 9	133
Tabla 23 - Límites de ángulos para herramienta blanca con entrada perpendicular a la mesa en el orificio 1	134
Tabla 24 - Límites de ángulos para herramienta blanca con entrada perpendicular a la mesa en el orificio 2	134
Tabla 25 - Límites de ángulos para herramienta blanca con entrada perpendicular a la mesa en orificio 7	134
Tabla 26 - Límites de ángulos para herramienta negra en orificio 4	134
Tabla 27 - Límites de ángulos para herramienta negra en orificio 5	135
Tabla 28 - Límites de ángulos para herramienta negra en orificio 6	135

Tabla 29 - Límites de ángulos de pivoteo para herramienta negra en los orificios 1 y 2	135
Tabla 30 - Límites de ángulos de pivoteo para herramienta negra en el orificio 3	136
Tabla 31 - Límites de ángulos de pivoteo para herramienta negra en los orificios 4 y 5	136
Tabla 32 - Límites de ángulos de pivoteo para herramienta negra en el orificio 6	136
Tabla 33 - Límites de ángulos de pivoteo para herramienta negra en los orificios 7 y 8	136
Tabla 34 - Límites de ángulos de pivoteo para herramienta negra en el orificio 9	136
Tabla 35 - Límites de ángulos de pivoteo para herramienta blanca en los orificios 1 y 2	136
Tabla 36 - Límites de ángulos de pivoteo para herramienta blanca en el orificio 7	137

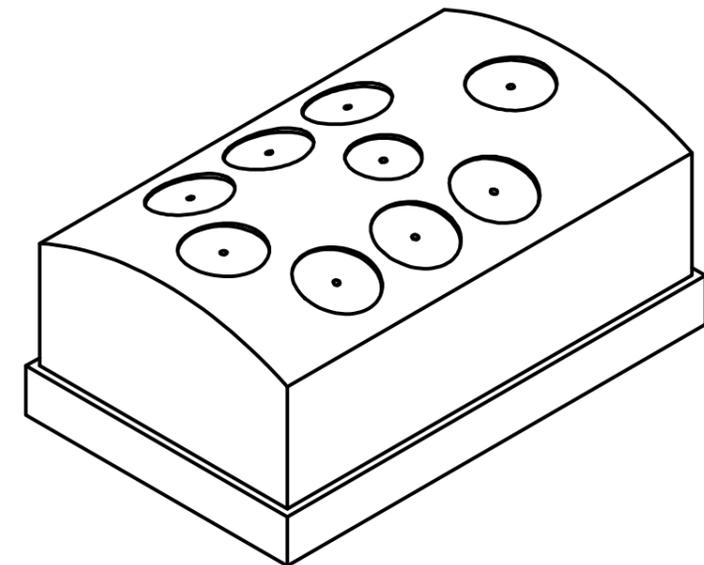
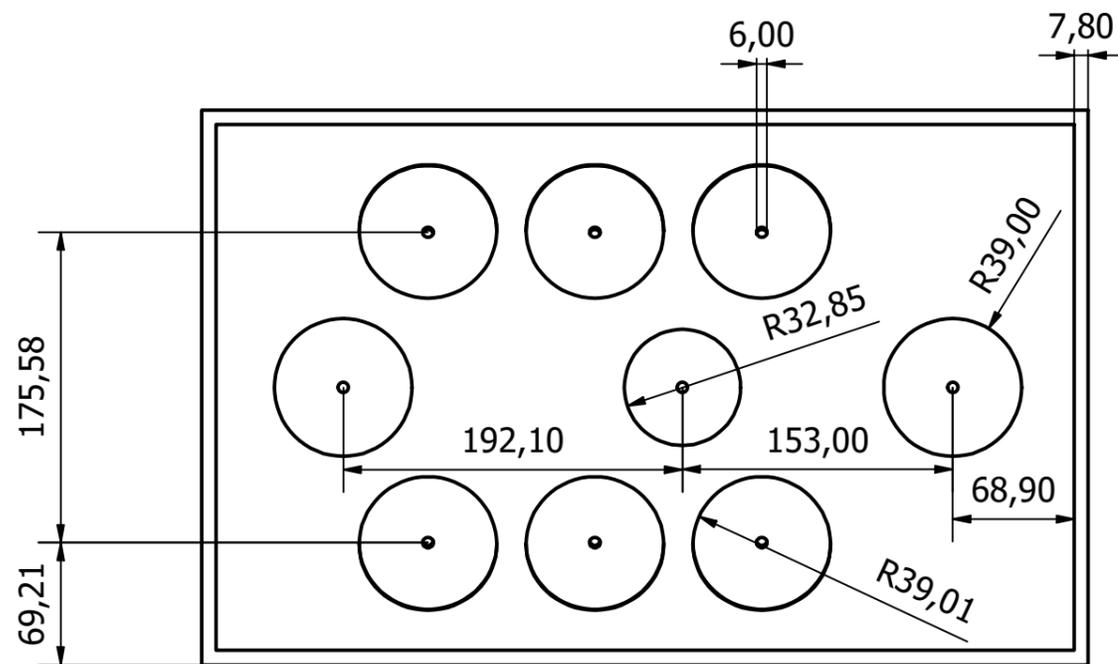
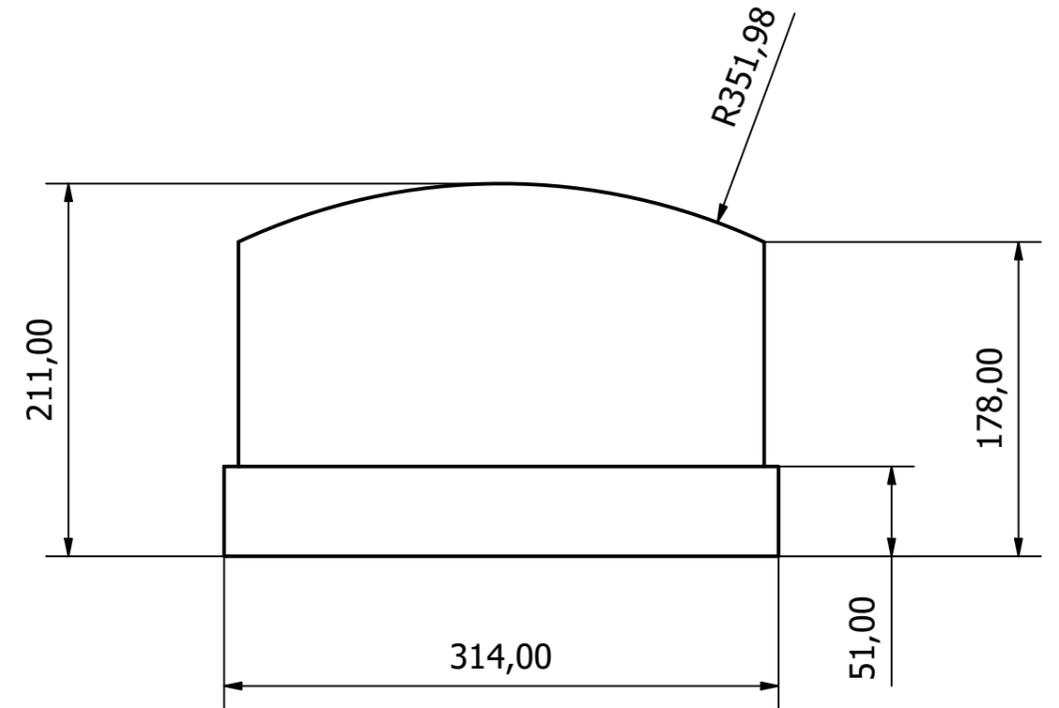
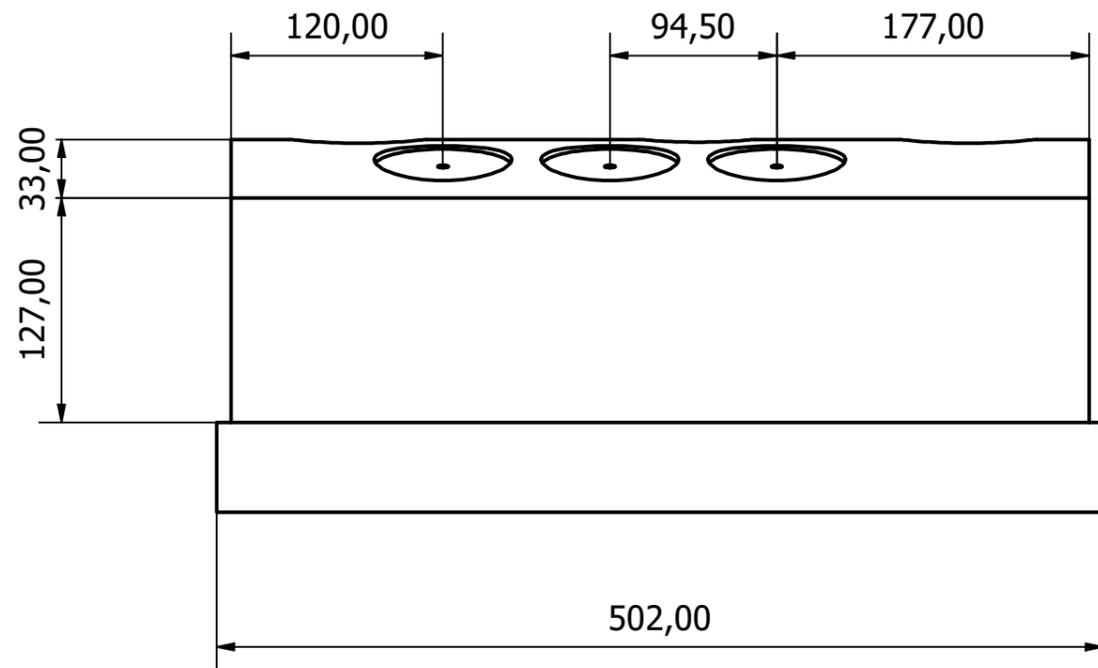
10. Anexos

10.1 Planos

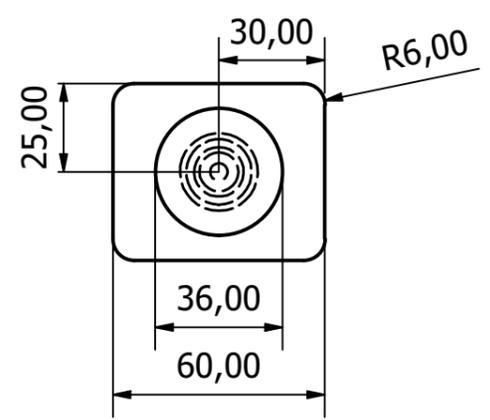
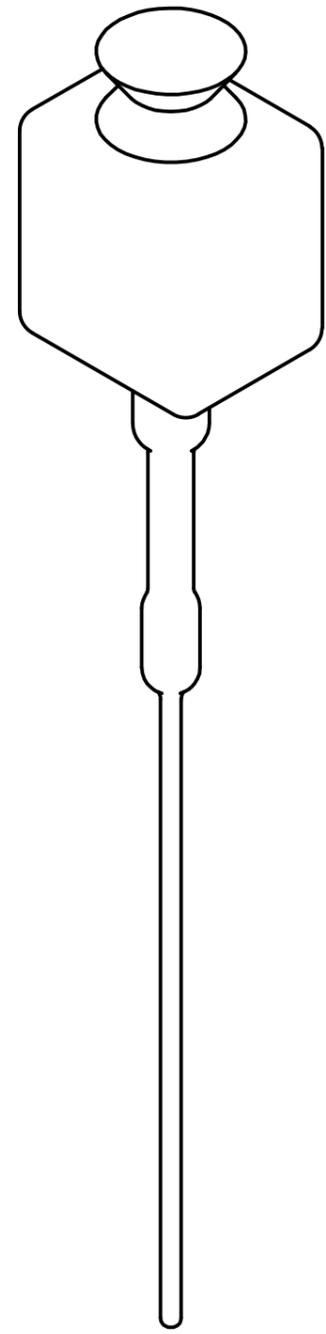
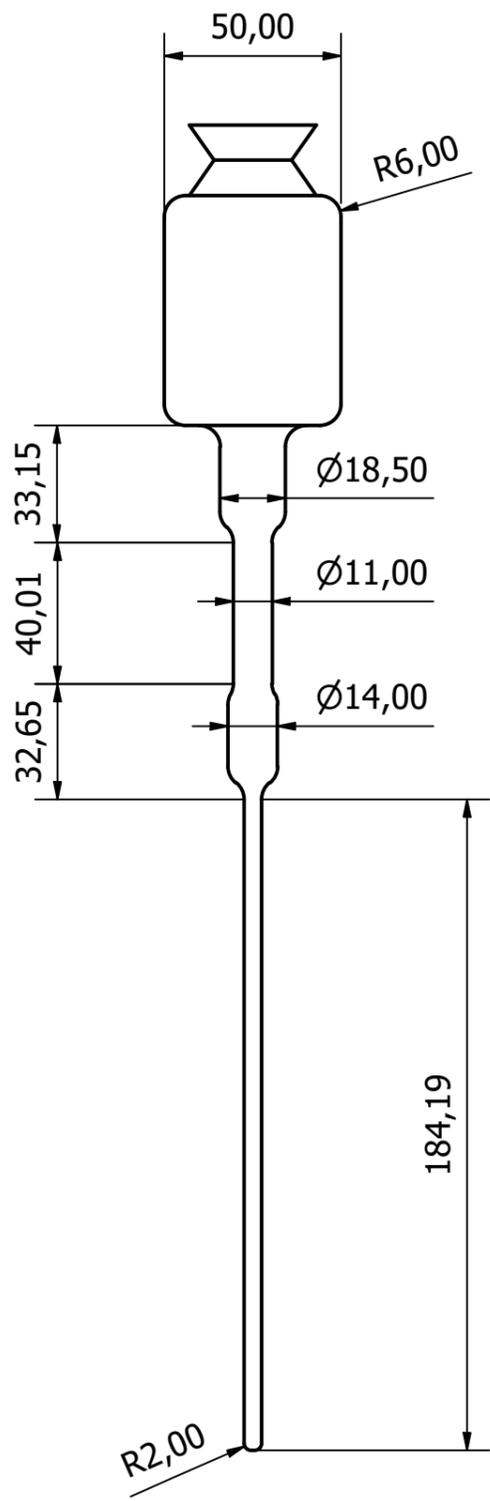
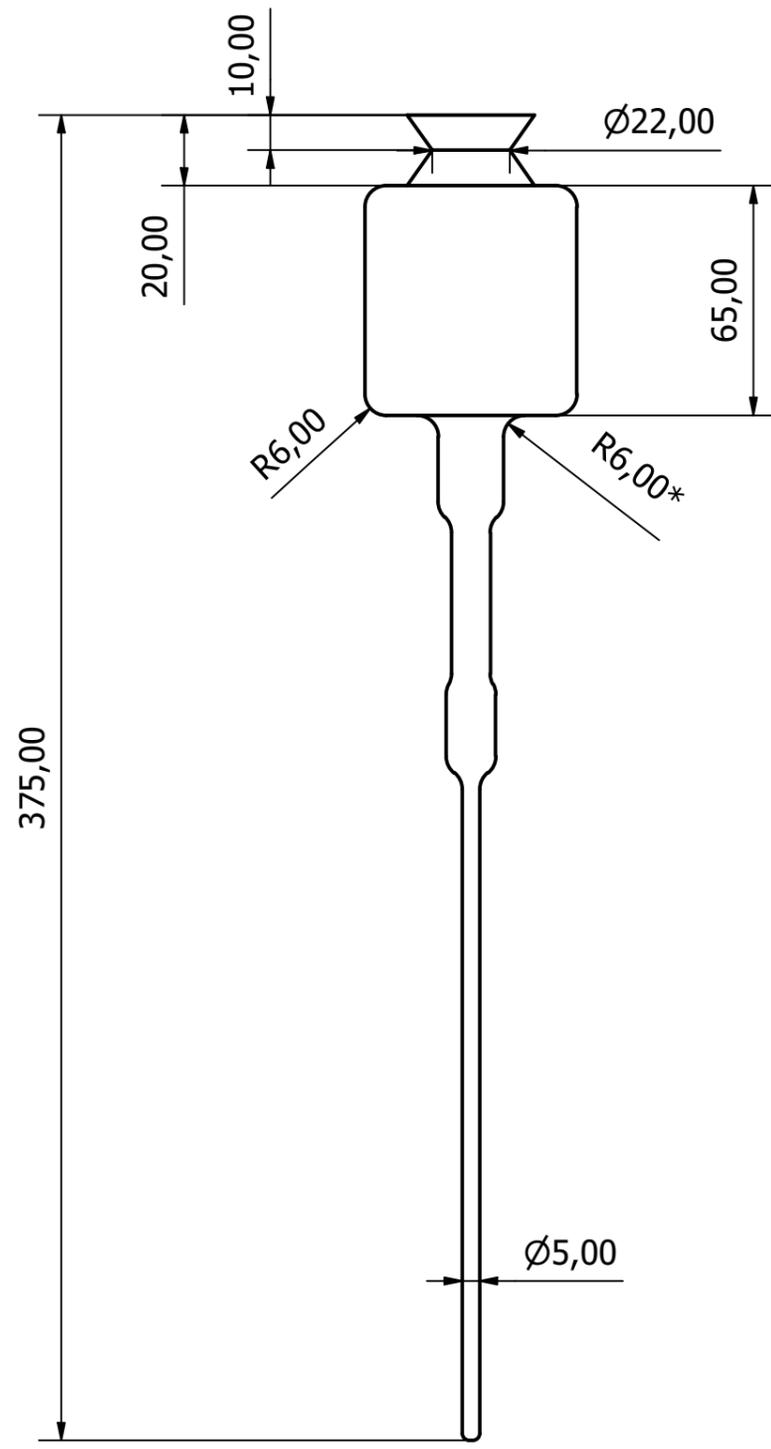
A continuación, se muestran los planos para cada uno de los elementos 3D modelados para la estación de trabajo.

Índice de planos

<i>Pelvitainer</i>	Plano 1
Herramienta negra	Plano 2
Herramienta blanca	Plano 3
Apoyo de herramientas negras	Plano 4
Apoyo de herramientas blancas	Plano 5
Vallado	Plano 6

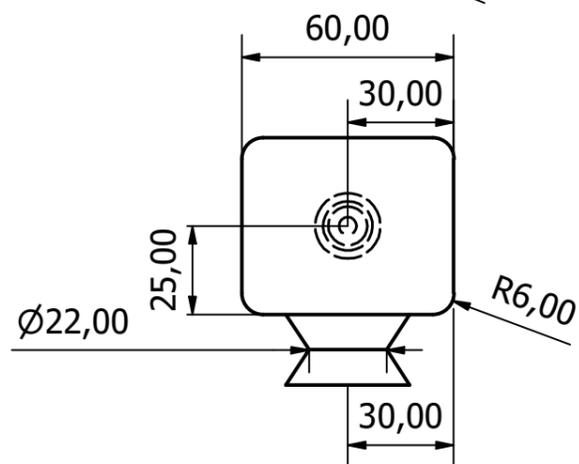
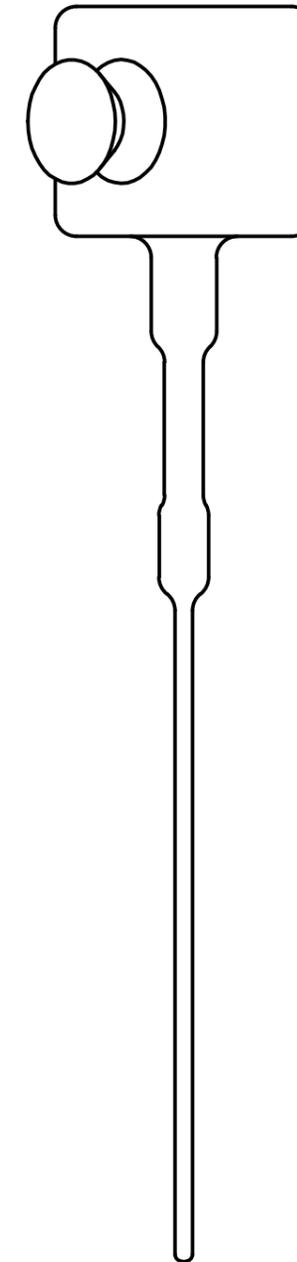
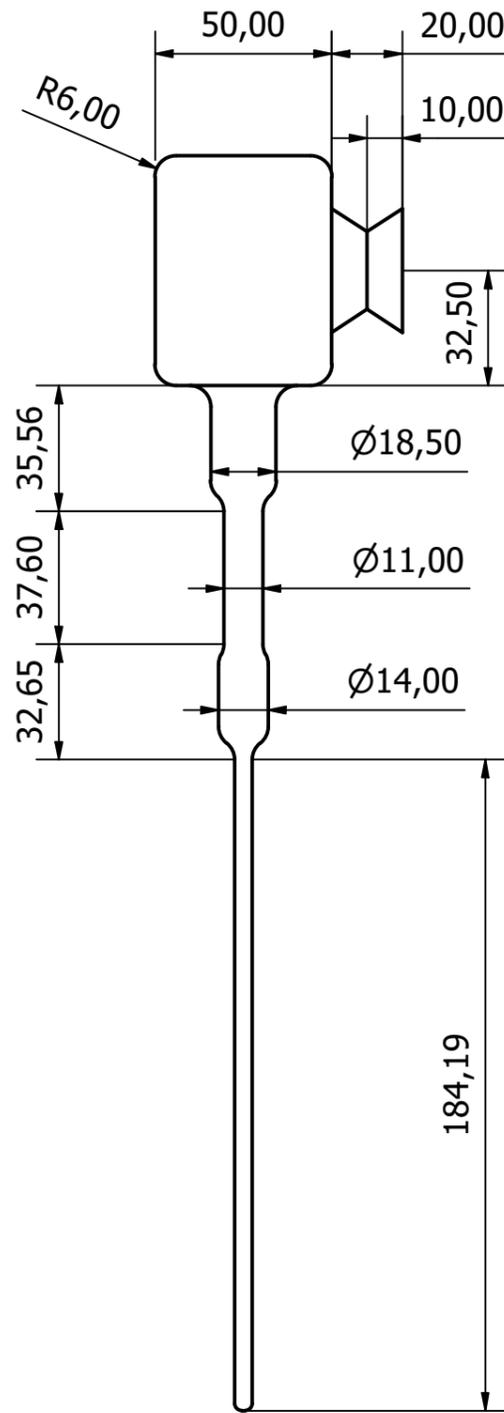
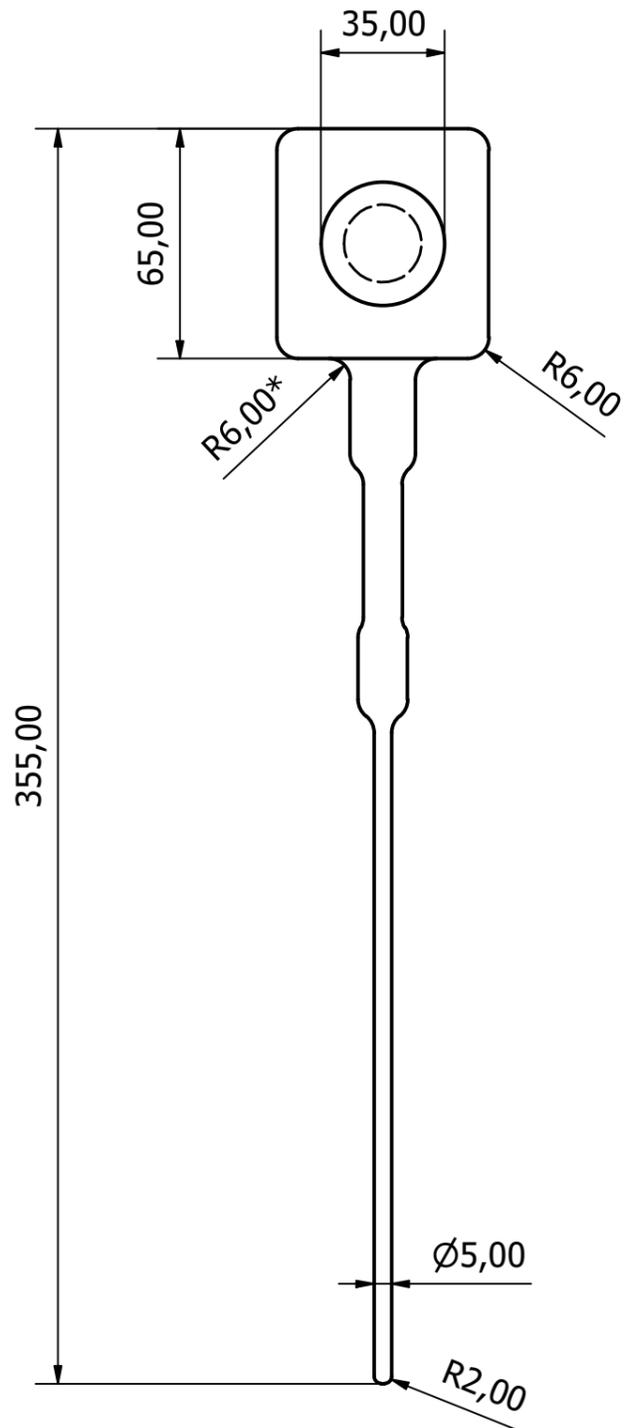


Diseño de Alejandro Sancho	Revisado por	Aprobado por	Fecha	Fecha 28/08/2021	
Grado en Ingeniería Electrónica Industrial y Automática			Pelvitainer		
			Universidad de Valladolid	Edición	Plano 1 / 6



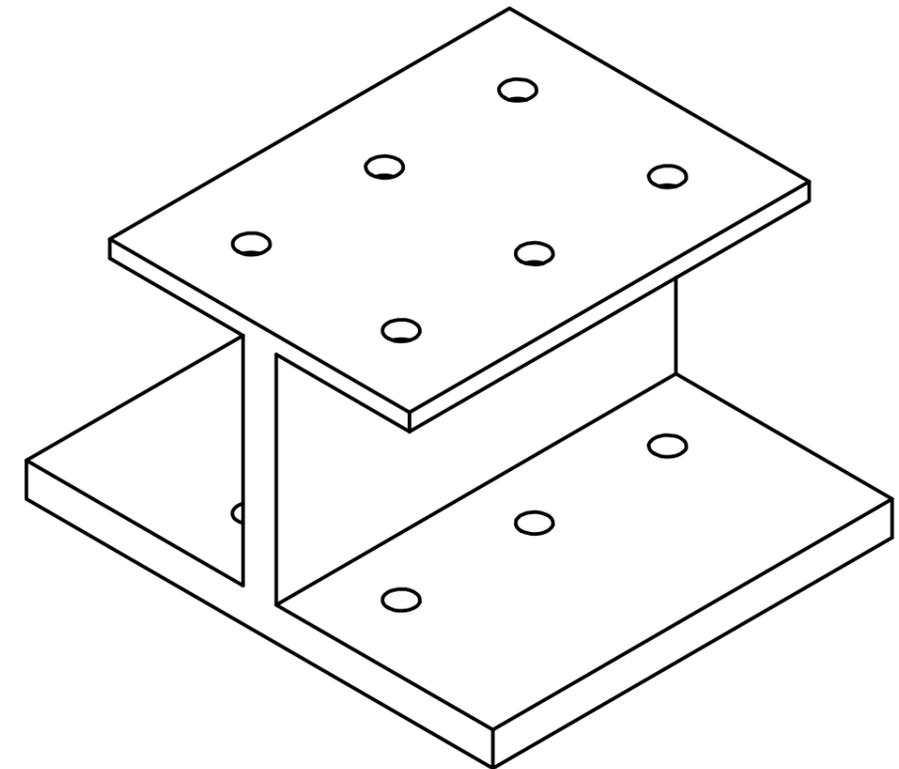
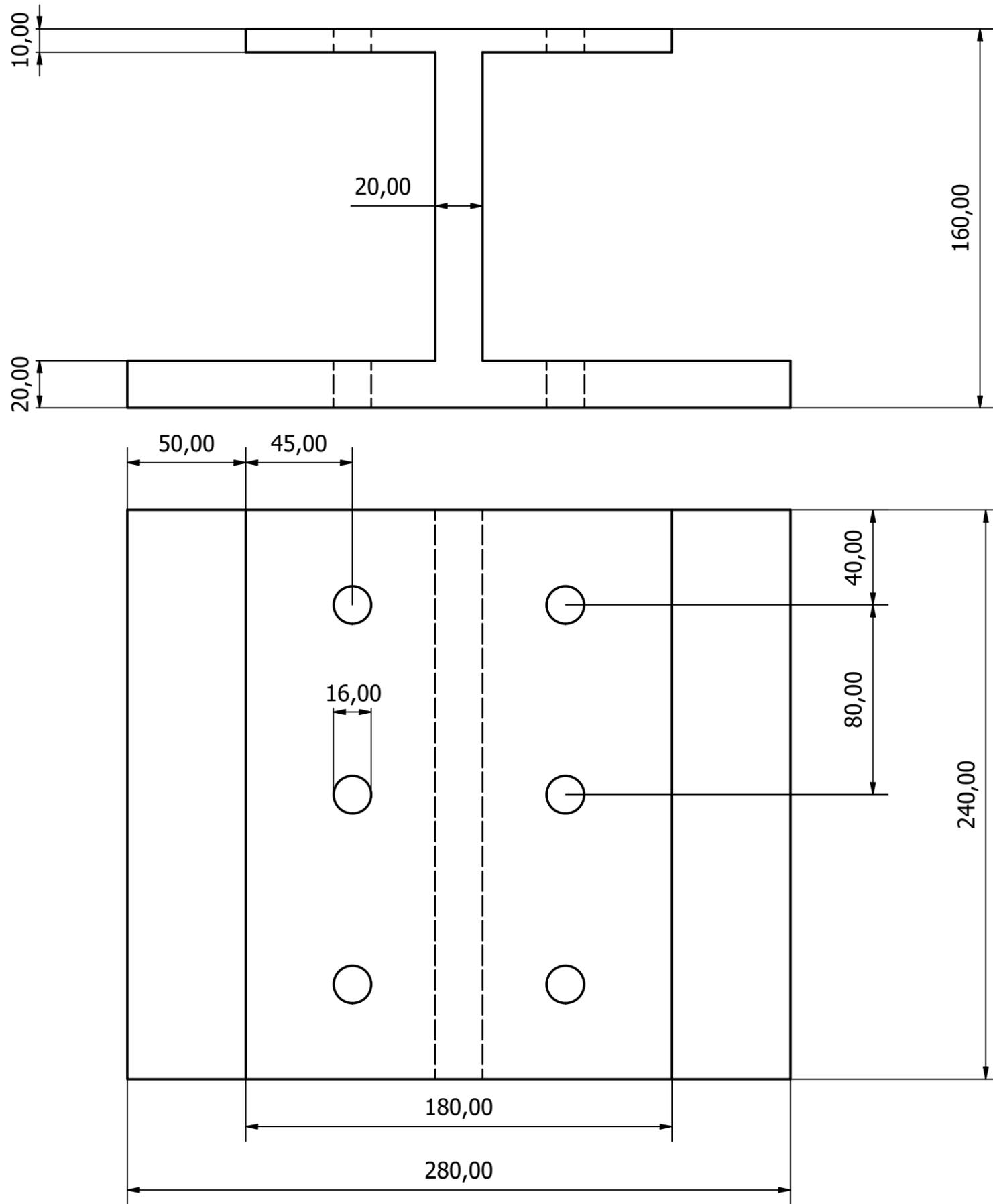
* El resto de empalmes, de debajo de la cota, son también de 6 milímetros

Diseño de Alejandro Sancho	Revisado por	Aprobado por	Fecha	Fecha 08/04/2022
Grado en Ingeniería Electrónica Industrial y Automática		Herramienta negra		
		Universidad de Valladolid	Edición	Plano 2 / 6

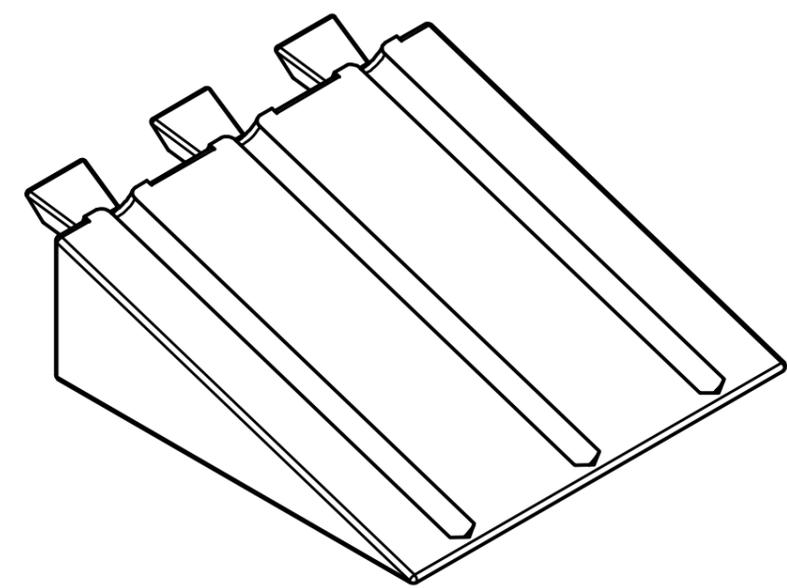
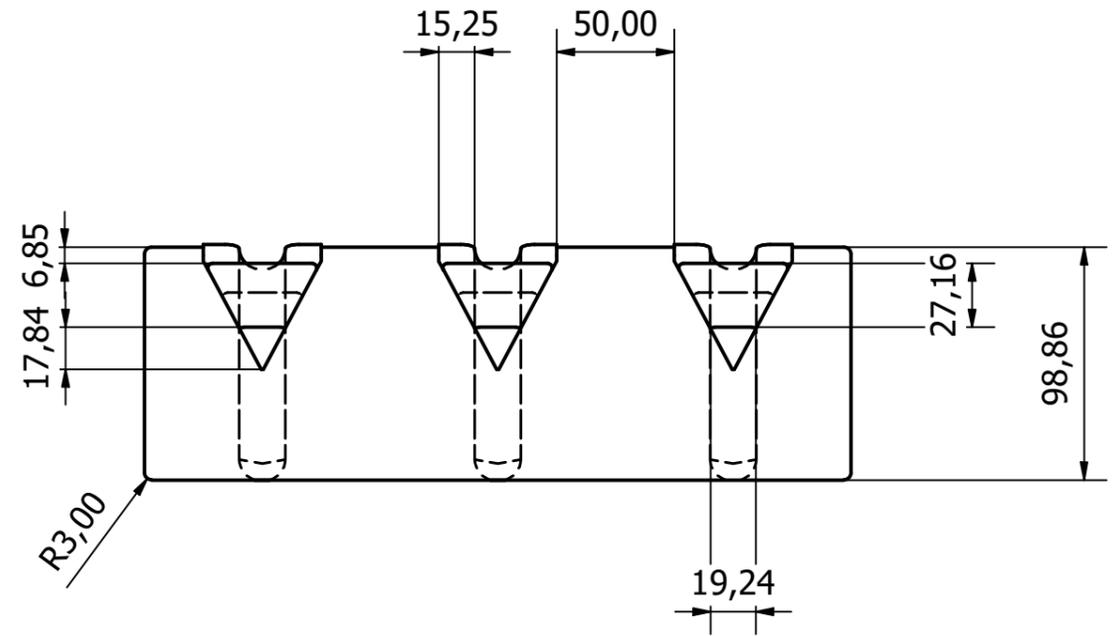
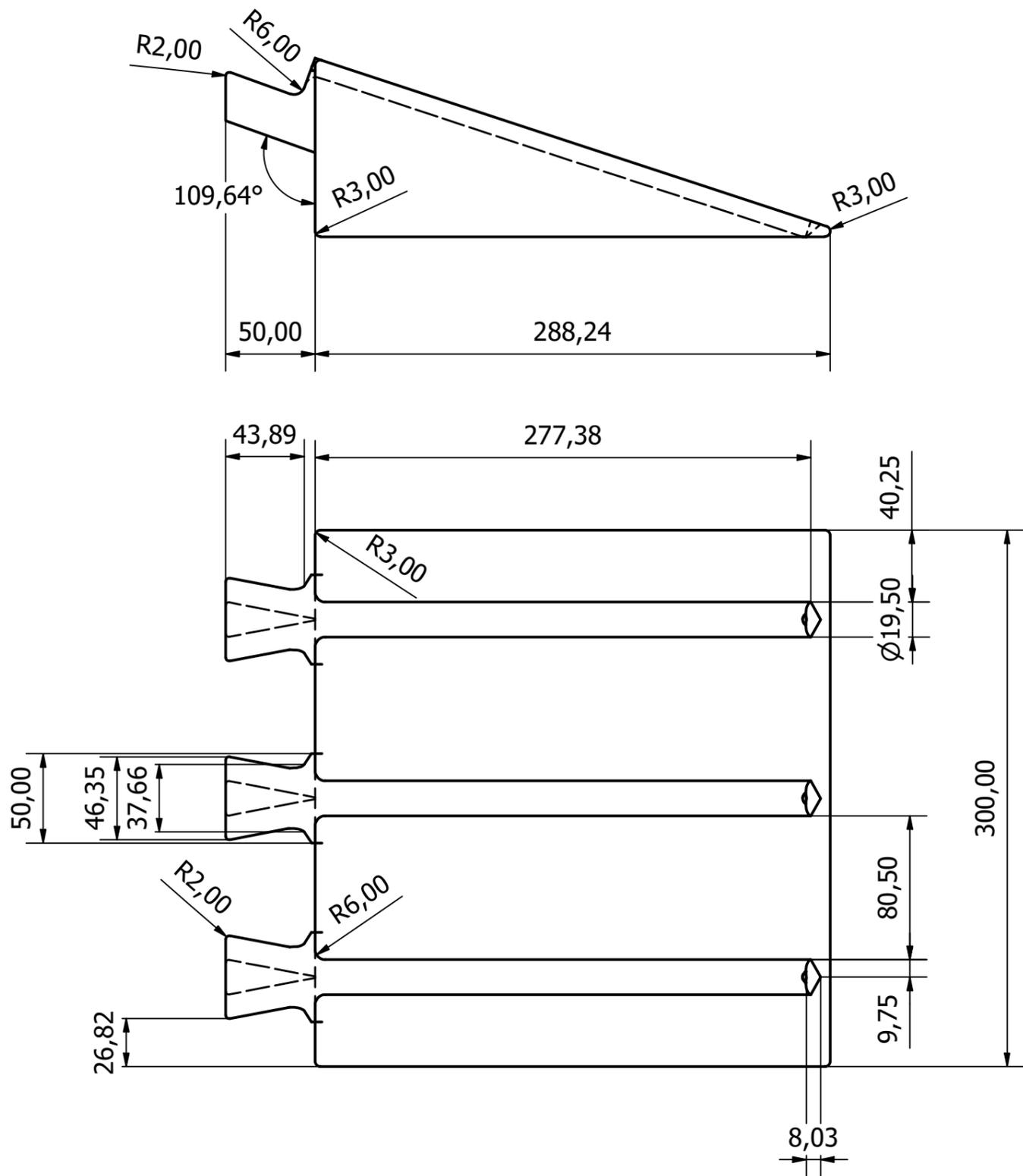


* El resto de empalmes, de debajo de la cota, son también de 6 milímetros

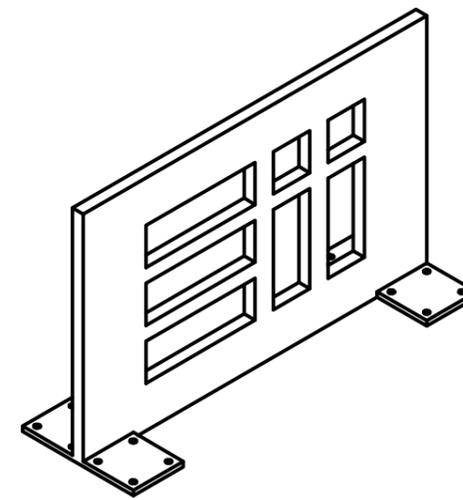
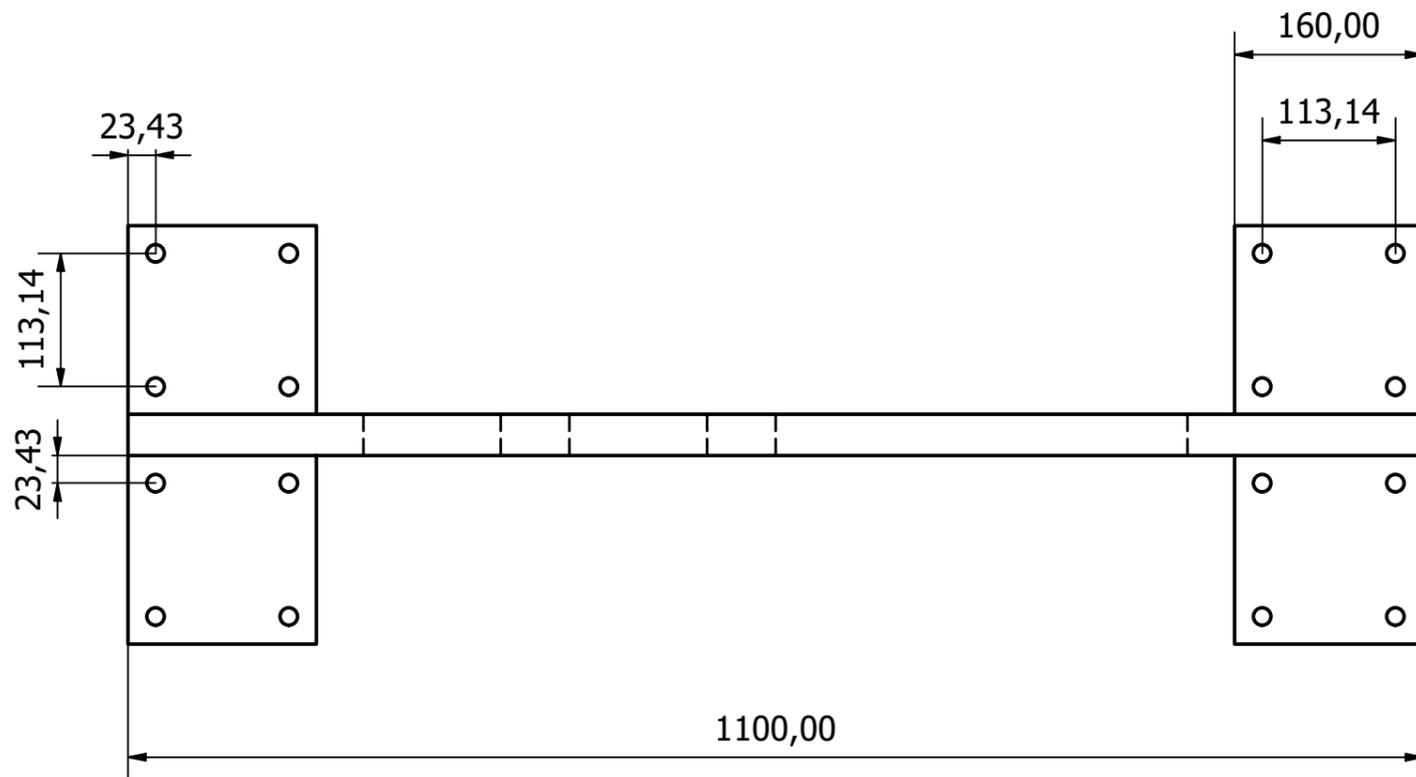
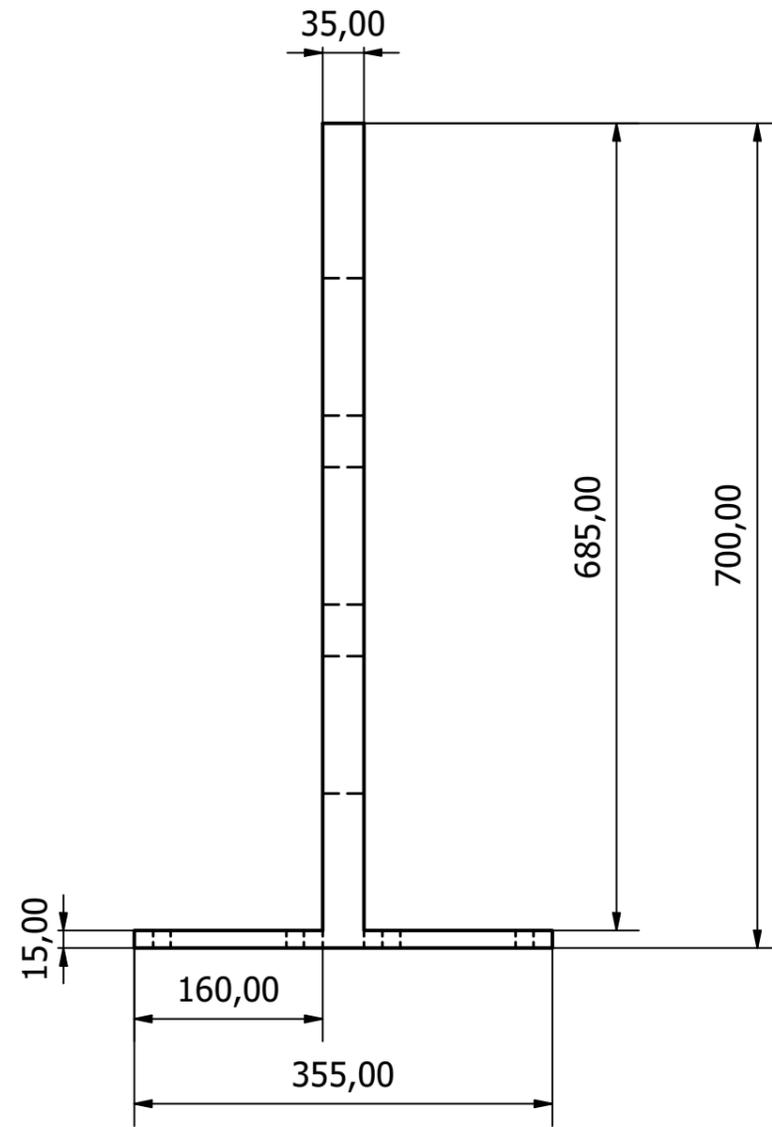
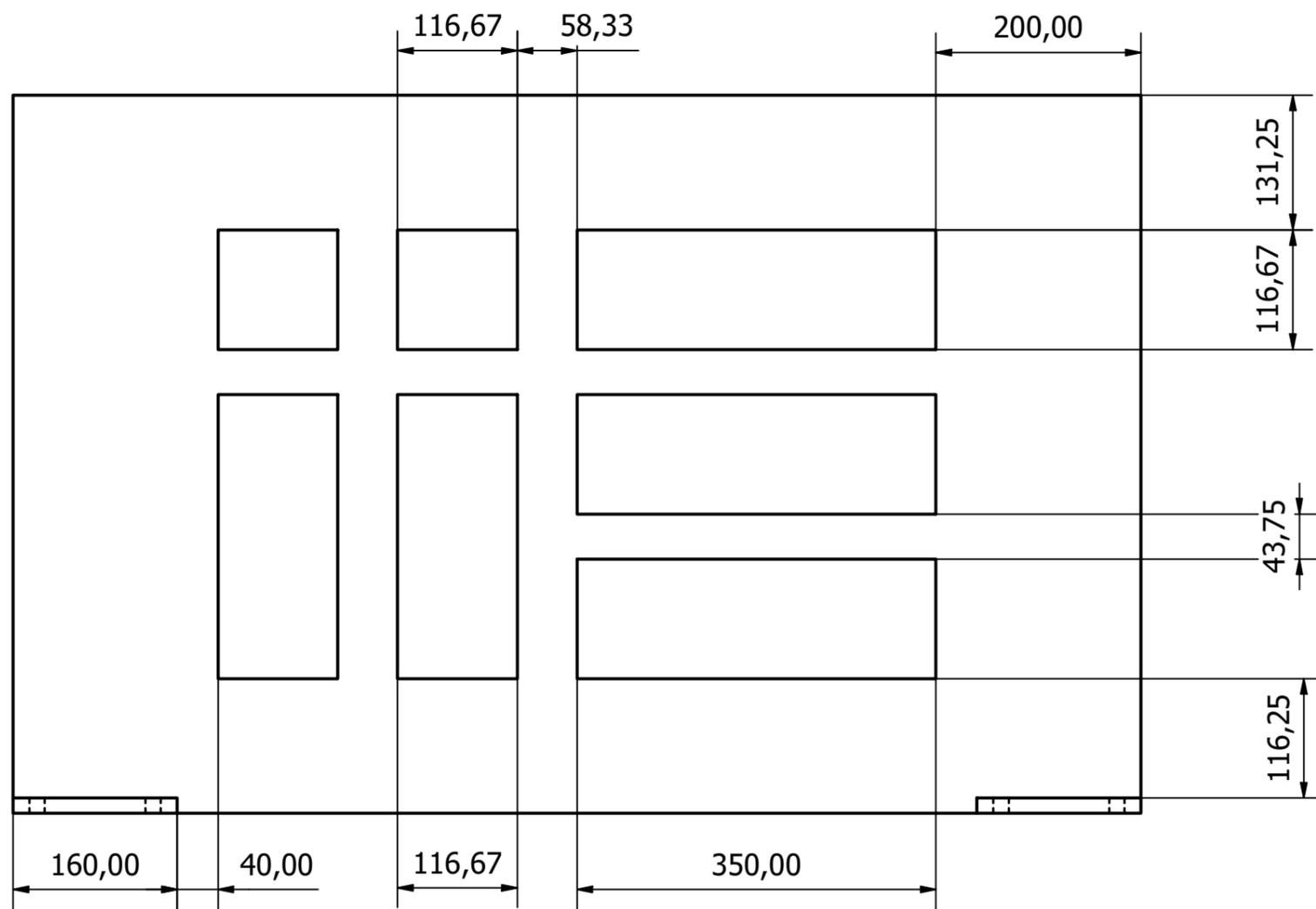
Diseño de Alejandro Sancho	Revisado por	Aprobado por	Fecha	Fecha 28/02/2022	
Grado en Ingeniería Electrónica Industrial y Automática			Herramienta blanca		
			Universidad de Valladolid	Edición	Plano 3 / 6



Diseño de Alejandro Sancho	Revisado por	Aprobado por	Fecha	Fecha 31/03/2022	
Grado en Ingeniería Electrónica Industrial y Automática		Apoyo herramientas negras			
		Universidad de Valladolid	Edición	Plano 4 / 6	



Diseño de Alejandro Sancho	Revisado por	Aprobado por	Fecha	Fecha 01/04/2022
Grado en Ingeniería Electrónica Industrial y Automática		Apoyo de herramientas blancas		
		Universidad de Valladolid	Edición	Plano 5 / 6



Diseño de Alejandro Sancho	Revisado por	Aprobado por	Fecha	Fecha 31/03/2022	
Grado en Ingeniería Electrónica Industrial y Automática			Vallado		
			Universidad de Valladolid	Edición	Plano 6 / 6