



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico mediante visión y robótica colaborativa.

Trabajo Fin de Máster

Máster Universitario en Ingeniería Industrial

AUTOR/A: Cubells Teodoro, Pedro Luis

Tutor/a: Valera Fernández, Ángel

Cotutor/a: Blanes Noguera, Juan Francisco

CURSO ACADÉMICO: 2023/2024

AGRADECIMIENTOS

“Aprovecho esta oportunidad para expresar mi más sincero y profundo agradecimiento por el cariño, apoyo y paciencia que me ha proporcionado mi familia a lo largo de todos estos años de aprendizaje. Quiero agradecer también a compañeros, amigos y profesores como Ángel por compartir valiosos conocimientos y prestar oportunidades tan valiosas. Tampoco puedo pasar por alto agradecer a todos mis compañeros y compañeras de equipo en el AI2, su apoyo y compañía ha sido invaluable. Por último, y no menos importante, agradecer a Paco por brindarme la oportunidad de participar en el proyecto y por el apoyo recibido.”

Resumen

El siguiente proyecto presenta el diseño, desarrollo e implementación de una aplicación industrial en el sector de la cerámica basada en robótica colaborativa y visión artificial. La aplicación consiste en perforar una esfera cerámica en toda su superficie, evitando las zonas donde se encuentran tallados motivos florales.

La justificación de este trabajo reside en la importancia actual de los sistemas robóticos y en la colaboración humano-robot fomentada por la Industria 4.0, generando aumentos en la productividad y eficiencia de las empresas. El aumento del uso de robots colaborativos en la industria es un indicativo de la continua búsqueda de mejora y optimización, reduciendo tiempos de producción y permitiendo la creación de nuevas estrategias de control.

Para desarrollar este proyecto es necesario, en primer lugar, obtener una forma de sujeción que permita acceder con total libertad y de forma completa a la cerámica esférica. Esta también debe tener cierta capacidad para mover la pieza para asegurar el acceso completo por parte del robot. También se han de diseñar soportes tanto para una cámara de visión como para la herramienta encargada de perforar, una Dremel 3000.

Para poder saber dónde se ha de perforar, es necesario usar un sistema de visión artificial que permita procesar la posición de la pieza. Para ello se usará una cámara OAK-D Pro, pues es capaz de distinguir entre zonas perforables y no perforables, a la vez de poder comunicar la información de forma precisa al centro de procesado.

Por último, es importante poseer un nodo que procese la información de la cámara y mande las señales pertinentes al robot UR 5 disponible y a la base de sujeción de la cerámica. Este dispositivo es una Raspberry Pi 5 con OS preparado de Luxonics, fabricante de la cámara. Para crear el código necesario se usará Python como lenguaje de programación y forma de comunicación entre todas las partes.

En resumen, para realizar el proyecto se usará un robot colaborativo UR 5, se creará una base para sujetar y manipular la pieza cerámica, una cámara OAK-D Pro para obtener imágenes y decidir dónde se ha de perforar y una Raspberry Pi 5 como sistema para procesar la información de la cámara y enviar

instrucciones a la base y al robot. Se espera obtener una solución que cumpla todos los requerimientos del proyecto.

Palabras Clave: Control de robots, Programación de robots, Robótica colaborativa, Visión 3D, Automatización, Aplicación industrial, Control por ordenador.

Resum

El següent projecte presenta el disseny, desenvolupament i implementació d'una aplicació industrial en el sector de la ceràmica basada en robòtica col·laborativa i visió artificial. L'aplicació consisteix a perforar una esfera ceràmica en tota la seua superfície, evitant les zones on es troben tallats motius florals.

La justificació d'este treball residix en la importància actual dels sistemes robòtics i en la col·laboració humà-robot fomentada per la Indústria 4.0, generant augments en la productivitat i eficiència de les empreses. L'augment de l'ús de robots col·laboratius en la indústria és un indicatiu de la contínua busca de millora i optimització, reduint temps de producció i permetent la creació de noves estratègies de control.

Per a desenvolupar este projecte és necessari, en primer lloc, obtenir una forma de subjecció que permeta accedir amb total llibertat i de manera completa a la ceràmica esfèrica. Esta també ha de tindre una certa capacitat per a moure la peça per a assegurar l'accés complet per part del robot. També s'han de dissenyar suports tant per a una càmera de visió com per a la ferramenta encarregada de perforar, una Dremel 3000.

Per a poder saber on s'ha de perforar, és necessari usar un sistema de visió artificial que permeta processar la posició de la peça. Per a això s'usarà una cambra OAK-D Pro, perquè és capaç de distingir entre zones perforables i no perforables, alhora de poder comunicar la informació de manera precisa al centre de processament.

Finalment, és important posseir un node que processe la informació de la cambra i mane els senyals pertinents al robot UR 5 disponible i a la base de subjecció de la ceràmica. Este dispositiu és una Raspberry Pi 5 amb US Preparat de Luxonics, fabricadora de la cambra. Per a crear el codi necessari s'usarà Python com a llenguatge de programació i forma de comunicació entre totes les parts.

En resum, per a realitzar el projecte s'usarà un robot col·laboratiu UR 5, es crearà una base per a subjectar i manipular la peça ceràmica, una cambra OAK-D Pro per a obtenir imatges i decidir on s'ha de perforar i una Raspberry Pi 5 com a sistema per a processar la informació de la cambra i enviar

instruccions a la base i al robot. S'espera obtenir una solució que complisca tots els requeriments del projecte.

Paraules Clau: Control de robots, Programació de robots, Robòtica col·laborativa, Visió 3D, Automatització, Aplicació industrial, Control per ordinador

Abstract

The following project presents the design, development and implementation of an industrial application in the ceramics sector based on collaborative robotics and artificial vision. The application consists of drilling a ceramic sphere across its entire surface, avoiding areas where floral motifs are carved. The justification for this work lies in the current importance of robotic systems and the human-robot collaboration fostered by Industry 4.0, generating increases in productivity and efficiency of companies.

The increase in the use of collaborative robots in the industry is indicative of the continuous search for improvement and optimization, reducing production times and allowing the creation of new control strategies.

To develop this project, it is necessary, first of all, to obtain a form of fastening that allows complete and complete access to the spherical ceramic. This must also have some ability to move the piece to ensure complete access by the robot. Supports must also be designed for both a vision camera and the tool in charge of drilling, a Dremel 3000.

In order to know where to drill, it is necessary to use an artificial vision system that allows processing the position of the piece. For this purpose, an OAK-D Pro camera will be used, as it is able to distinguish between pierceable and non-perforable areas, while being able to communicate the information accurately to the processing center.

Finally, it is important to have a node that processes the information from the camera and sends the relevant signals to the available UR 5 robot and the ceramic holding base. This device is a Raspberry Pi 5 with OS prepared by Luxonics, manufacturer of the camera. To create the necessary code, Python will be used as a programming language and form of communication between all parties.

In short, to carry out the project will be used a collaborative robot UR 5, a base to hold and manipulate the ceramic part, an OAK-D Pro camera to obtain images and decide where to drill and a Raspberry Pi 5 as a system to process the information from the camera and send instructions to the base and the robot. It is expected to obtain a solution that meets all the requirements of the project.

Keywords: Robot control, Robot programming, Collaborative robotics, 3D vision, Automation, Industrial application, Computer control.

Índice general

Índice general	7
Índice de ilustraciones.....	10
Índice de tablas	13
1. Introducción	14
1.1 Contexto y justificación.....	14
1.2 Objetivos.....	15
1.3 Estructura del trabajo.....	15
2. Fundamentos teóricos.....	18
2.1 Industria 4.0.....	18
2.2 Robots Colaborativos de Universal Robots.....	19
2.3 OAK-D Pro de Luxonis.....	21
2.4 Raspberry Pi 5 de Raspberry Pi.....	22
3. Base y soportes.....	29
3.1 Creación de la base.....	29
3.1.1 Consideraciones previas al diseño.....	29
3.1.1.1 Puntos clave a resolver	29
3.1.2 Diseño de la base.....	30
3.1.2.1 Primera versión	31
3.1.2.2 Segunda versión	33
3.1.3 Desarrollo de la versión seleccionada	35
3.1.3.1 Primera iteración	35
3.1.3.2 Segunda iteración.....	37
3.1.3.3 Tercera iteración	39
3.1.4 Prototipo.....	40
3.1.4.1 Primer prototipo.....	40
3.1.4.2 Prototipo final.....	45

3.2 Diseño soporte Dremel.....	55
3.3.1 Primera versión.....	55
3.3.2 Segunda versión	59
3.3.3 Tercera versión	62
3.3 Diseño soporte OAK-D PRO	63
4. Controlador y comunicación	66
4.1 Consideraciones previas a la implementación	66
4.1.1 Selección del sistema empotrado.....	66
4.1.2 Lenguaje de programación	67
4.1.3 Herramientas de compilación.....	68
4.2 Montaje de controladores.....	68
4.2.1 DRV8825	68
4.2.2 Montaje de placa para controladores	72
4.3 Bibliotecas usadas	75
4.3.1 Para motores - Gpiozero.....	75
4.3.2 Para OAK-D Pro	75
4.3.3 Para UR 5	78
4.4 Construcción del programa	78
5. Control y maniobras brazo robot.....	82
5.1 Consideraciones previas	82
5.2 Desarrollo	84
5.3 Configuración y código definitivo	86
6. Funcionamiento y análisis de resultados	91
6.1 Configuración final.....	91
6.2 Análisis de resultados	93
6.3 Conclusiones del análisis	97
7. Conclusión	98
7.1 Objetivos logrados.....	98
7.2 Líneas de mejora	99
7.3 Reflexión final	99
Bibliografía.....	101
Presupuesto general del proyecto.....	104
Resumen del presupuesto	104

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico
mediante visión y robótica colaborativa.

Presupuesto detallado.....	104
A. Planificación del proyecto	109
A.1 Planificación de tareas del proyecto	110

Índice de ilustraciones

Ilustración 1. Pieza cerámica a agujerear en distintos estados de su proceso.	16
Ilustración 2. Esquema gráfico del planteamiento del proyecto.	17
Ilustración 3. Imagen de un teach pendant perteneciente a un UR 5. Fuente: Automation Distribution.	21
Ilustración 4. Esquema visual sobre DepthAI API. Fuente: Luxonics.	22
Ilustración 5. Reunión en 2012. De izquierda a derecha: Martin Cartwright, Profesor Alan Mycroft, Pete Lomas, David Braben, Eben Upton Jack Lang and Alex Bradbury. Fuente: Raspberry Pi Foundation...	22
Ilustración 6. Primer prototipo Raspberry Pi (2006, con el mismo rendimiento que BBC Micro de 1980s). Fuente: Raspberry Pi Foundation.	23
Ilustración 7. Raspberry Pi 5.	23
Ilustración 8. Placas mencionadas: a la izquierda Raspberry Pi Compute Module 4S y a la derecha Raspberry Pi Pico 2. Fuente: Raspberry Pi Foundation.	24
Ilustración 9. Raspberry Pi 5.	24
Ilustración 10. Configuración pines GPIO Raspberry Pi 5. Fuente: Raspberry Pi Foundation.	25
Ilustración 11. Obtención por pantalla al ejecutar "gpio readall". Fuente: https://solectroshop.com/	26
Ilustración 12. Obtención por pantalla al ejecutar "pinout". Fuente: https://solectroshop.com/	27
Ilustración 13. Estados de la cerámica: izquierda sin tratar, centro tratada y derecha es un prototipo.	30
Ilustración 14. Primer prototipo de base.	31
Ilustración 15. Segunda iteración de primer prototipo.	32
Ilustración 16. Pieza montada en segunda iteración de primer prototipo.	33
Ilustración 17. Primera iteración de un apoyo.	35
Ilustración 18. Primera iteración de la parte superior de la base.	36
Ilustración 19. Ejemplo modelado con las piezas anteriores.	36
Ilustración 20. Segunda iteración de los apoyos.	37
Ilustración 21. Segunda iteración de la parte superior de la base.	38
Ilustración 22. Segundo ejemplo modelado con las piezas anteriores.	38
Ilustración 23. Tercera iteración de los apoyos.	39
Ilustración 24. Tercera iteración de la parte superior de la base.	40
Ilustración 25. Parte superior del primer prototipo impreso.	41
Ilustración 26. Primer prototipo de los apoyos.	41
Ilustración 27. Cálculos realizados para obtener el punto óptimo donde posicionar el eje.	42
Ilustración 28. Casquillos: izquierda modelado y derecho impreso.	42
Ilustración 29. Primer prototipo de rodillo montado.	43
Ilustración 30. Colocación y ajuste de los rodillos previo a montaje.	43
Ilustración 31. Primer prototipo montado con distintas vistas.	44
Ilustración 32. Prototipo preparado para realizar diversas pruebas.	44
Ilustración 33. Modelado de apoyo final de la base.	46

Ilustración 34. Modelado de la parte superior de la base final.....	47
Ilustración 35. Machos de rosca M6x1.00. Fuente: ManoMano.es	47
Ilustración 36. Ejemplo de rodamiento 626-2Z. Fuente: Rs-online.....	48
Ilustración 37. Muestra de la base atornillada correctamente.	48
Ilustración 38. Diseño de casquillo con sección extraída plana.	49
Ilustración 39. Modelado de casquillo con sección extraída en ángulo.....	50
Ilustración 40. Visualización de ejes colocados en su posición correspondiente.....	50
Ilustración 41. Motor Wantai 42BYGHW811i usado. Fuente: Openimpulse.com.....	51
Ilustración 42. Diseño del soporte a alojar los motores Wantai de la base.	52
Ilustración 43. Acopladores de distinto diámetro usados. Fuente: Amazon.es	52
Ilustración 44. Montaje final de la base.	53
Ilustración 45. Montaje probando el comportamiento de la espuma aislante.	54
Ilustración 46. Solución alcanzada con el filtro de aire de espuma.....	54
Ilustración 47. Montaje de la base definitivo. En la espuma se puede apreciar el residuo de la cerámica.	55
Ilustración 48. Herramienta Dremel usada. Fuente: Amazon.es.....	56
Ilustración 49. Eje extensor de Dremel. Fuente: Dremel.com.....	56
Ilustración 50. Primera versión del soporte para el eje extensible.	57
Ilustración 51. Forma de obtener la geometría observada en el soporte.....	57
Ilustración 52. Zona del eje extensible reforzada.....	58
Ilustración 53. Refuerzos colocados rigidizar más el soporte.....	58
Ilustración 54. Parte complementaria al soporte anterior que retiene al eje.....	59
Ilustración 55. Preimera versión de las piezas impresas.	59
Ilustración 56. Comparación entre soportes (arriba el nuevo y abajo la primera versión).....	60
Ilustración 57. Mejora no implantada.	61
Ilustración 58. Segunda versión de la pieza complementaria al soporte.	61
Ilustración 59. Segunda versión del soporte del eje extensible.	62
Ilustración 60. Tercera y última versión del soporte del eje extensible.	62
Ilustración 61. Soporte para OAK-D Pro.	63
Ilustración 62. Vista lateral de soporte OAK-D Pro.....	64
Ilustración 63. Montaje completo obtenido a través de impresión 3D desde distintos ángulos.	64
Ilustración 64. Controlador DRV8825 con disipador. Fuente: electrocomponentes.es.	69
Ilustración 65. Esquema de los pines del controlador. Fuente: luisllamas.es	69
Ilustración 66. Esquema visual sobre cómo funciona un motor a pasos enteros.	70
Ilustración 67. Muestra visual de cómo se obtienen señales senoidales a través de un sistema digital.	71
Ilustración 68. Placa para conectar 4 controladores DRV8825.	72
Ilustración 69. Conexiones necesarias para controlar el sistema.....	73
Ilustración 70. Fuente de alimentación RS-3005D. Fuente: Rs-online.com	74
Ilustración 71. Parte del código que crea las conexiones con la cámara.....	76
Ilustración 72. Definición de propiedades se unen todos los nodos creados.	76
Ilustración 73. Cola de la retrasnmisión.	77
Ilustración 74. Interior del bucle de la función camera.....	77
Ilustración 75. COmparación entrela imagen sin recortar (abajo) y la imagen recortada (arriba).	78

Ilustración 76. Esquepa para empoezar las comunicaciones siendo el servidor.....	79
Ilustración 77. Posiciones de perforación y delimitaciones.	80
Ilustración 78. Línea que envía una coordenada.....	80
Ilustración 79. Diagrama con el funcionamiento del código explicado anteriormente explicado.	81
Ilustración 80. Esquema que resumen los tipos de comunicaciones que poseen los robots de UR.Fuente:Universal Robots.....	83
Ilustración 81. Imagen del simulador offline de un UR 5 CB.	84
Ilustración 82. Frame de un vídeo junto un aumento a una perforación no realizada correctamente con escamado.	85
Ilustración 83. Localización del TCP en Polyscope de UR 5e.	86
Ilustración 84. Posición del centro de gravedad y obtención del peso del montaje.	86
Ilustración 85. Variables y posiciones encontradas en el código.	88
Ilustración 86. Parte del código que se ejecuta antes de empezar el bucle principal del robot.	89
Ilustración 87. Configuración final del proyecto.....	91
Ilustración 88. Modo modo libe para guiar el robot con las flechas.	92
Ilustración 89. Posición del centro del plano respecto la base del robot.....	93
Ilustración 90. Momento donde se produce mal contacto (izquierda) y daños recibidos en la cerámica (derecha).	94
Ilustración 91. Cerámica con escamados cerca de la zona perforada.....	95
Ilustración 92. Buena calidad de perforaciones conseguidas a poca distancia.	95
Ilustración 93. Observación del plano del centro de la cerámica estando de frente al robot.	96
Ilustración A94. Diagrama de Gantt.	112

Índice de tablas

Tabla 1. Comparación entre los materiales disponibles.....	53
Tabla 2. Pasos de microstepping en función de pines activados en DRV8825 y A4988	71
Tabla 3. Pines asignados a cada variable.	74
Tabla 4. Presupuesto Básico	104
Tabla 5. Presupuesto detallado	104
Tabla A6. Cronograma seguido.....	110

Capítulo 1

1. Introducción

1.1 Contexto y justificación

Cada día es más habitual encontrar robots en tareas donde hasta el momento no era posible usarlos, bien por el coste de la instalación completa o bien por la complejidad de la tarea a realizar. Es por este motivo que la aparición de robots colaborativos ha permitido abaratar y simplificar los altos costes de instalación puesto que se tratan de un tipo de robot fácil de programar y utilizar. Además, estos robots tienen instaladas medidas de control para garantizar la seguridad cuando se trabaja en conjunto con humanos.

Por este motivo, cada vez más empresas buscan implementar estos robots en sus instalaciones para mejorar la producción sin afectar a la seguridad, como por ejemplo Audi y Nissan proponiendo fábricas donde robot y humano trabajan a la par mejorando la flexibilidad y la productividad.

Este proyecto se trata de una aplicación similar, pero a menor escala, pues la justificación viene originada por la voluntad de una empresa de comprobar si se puede implementar un sistema de perforación automático en el sector cerámico mediante visión y robótica colaborativa.

Esta cerámica se trata de una esfera de unos 115 mm de diámetro la cual tiene tallados unos motivos florales y cuenta con una sección cortada en la cual hay un orificio grande para insertar un perfume. El objetivo de este proyecto es perforar de forma rápida y precisa una serie de orificios en la superficie completa de la pieza, evitando las zonas con los motivos florales.

Debido a la forma y posición de los orificios es necesario diseñar un soporte instrumentado que permita el acceso completo a la pieza cerámica. Además, para realizar estos orificios se usará una Dremel 3000 usando una broca de diámetro similar, creando los soportes necesarios para que se puede implementar en la muñeca del robot UR 5 usado.

Para poder distinguir de forma automática las zonas talladas es necesario implementar un sistema de visión artificial que permita de alguna forma diferenciar dónde se ha de perforar y donde no. Para ello se adquirirá una cámara de visión 3D que guíe el movimiento de la pieza en la base creada a la vez de seleccionar dónde se ha de perforar.

En resumen, este proyecto nace de la necesidad de las empresas de modernizarse y aprovechar la tecnología actual para implementar una colaboración humano-robot que aumente tanto la producción como el bienestar de los trabajadores.

1.2 Objetivos

El presente trabajo tiene como objetivo principal diseñar y desarrollar un sistema que permita agujerear una cerámica esférica usando un brazo colaborativo de Universal Robots mediante la comunicación TCP entre una cámara OAK-D Pro, un soporte donde se manipulará la cerámica, una Dremel y el robot ya mencionado. Esta comunicación usa como intermediario una Raspberry Pi 5, que es la encargada de procesar los datos de la cámara y tomar decisiones sobre el estado de la cerámica. Para lograr dicho objetivo, se han planteado una serie de requerimientos principales para definir el proyecto:

- Comportamiento preciso: Asegurar que todos los agujeros se realizaran en las posiciones deseadas de forma que no se dañe la pieza, además de obtener un soporte que permita alcanzar todas las partes de la cerámica.
- Repetibilidad: Asegurar que los resultados obtenidos se puedan replicar con la menor varianza posible, asegurando que el proceso se puede realizar múltiples veces y, por lo tanto, es un proceso industrial automatizable.
- Velocidad de perforación adecuada: Realizar las perforaciones en las piezas a una velocidad mayor o igual a la velocidad obtenida al hacer el proceso a mano.
- Obtener perforaciones de calidad: Automatizar el proceso debe permitir una mejora de calidad debido a que este se realiza bajo condiciones controladas y usando un sistema que permite repetir las operaciones con gran precisión.
- Modificabilidad y reusabilidad: Facilitar la implementación del sistema diseñado de forma que permita la modificación del sistema. Además, se busca capacidad de usar componentes modulares para poder reemplazar y mejorar de forma continua el sistema.

1.3 Estructura del trabajo

El trabajo se dividirá en distintos capítulos para explicar con claridad y coherencia los aspectos clave del proyecto.

En este primer capítulo se ha expuesto una introducción al tema, añadiendo contexto a la importancia de la aplicación de robots colaborativos y estableciendo una estructura básica como guía para abordar el proyecto. Durante el capítulo 2, se proporciona unas bases teóricas y explicaciones abarcando los conceptos fundamentales relacionados con las distintas partes del proyecto, introduciendo la tecnología que se ha usado para obtener una solución al planteamiento.

La creación de los prototipos y la implementación de las partes se realiza durante los capítulos 3, 4 y 5. Este sistema se ha estructurado en 3 partes principales las cuales corresponden con los tres capítulos citados anteriormente.

Comenzando por la primera parte, el capítulo 3 desarrolla la creación del sistema de sujeción de la pieza cerámica para asegurar una perforación exitosa y de otros soportes. Partiendo de una pieza cerámica sin cocinar, es necesario obtener algún sistema que permita la fijación de esta pieza esférica a la vez que permita el acceso completo a todas las partes de la pieza sin entorpecer el movimiento del robot. También se ha de tener en cuenta la una sección plana que se encuentra en la parte inferior de esta la cual tiene un agujero que permite acceder al interior de la cerámica, pues se ha de tener en cuenta en caso de utilizar algún apoyo lateral por si se pierde el contacto por este, además de tener en cuenta el residuo en forma de polvo que deja la cerámica al no estar cocinada y que entra en contacto con las superficies donde apoya.

En el caso de los otros soportes, se tratan de soportes para fijar la herramienta usada para perforar y para mantener la cámara de visión unida al brazo robot para poder visualizar y definir dónde se han de realizar las perforaciones.



Ilustración 1. Pieza cerámica a agujerear en distintos estados de su proceso.

En el capítulo 4 se profundiza en el montaje de la parte electrónica y la programación que permite la correcta comunicación entre las partes de la solución desarrollada. Se detalla cómo se realiza la conexión entre los motores y el sistema usado para procesar la información, así como la programación realizada para ello. También se desarrolla el funcionamiento de la cámara y cómo se tratan los datos obtenidos por ella para obtener la imagen que se desea, explicando cual es el criterio para perforar o no en función de la imagen obtenida y la posición del dibujo tallado en la pieza.

En el capítulo 5, se explora la implementación del sistema con el brazo robot seleccionado. En este se explica cómo se obtiene la solución final, los problemas aparecidos durante el desarrollo y se enseña el funcionamiento del código y la configuración realizada necesarias para realizar el programa y que opere todo según especificaciones.

La evaluación de los resultados se realiza en el capítulo 6, examinando la calidad de las perforaciones, la velocidad de perforado y el comportamiento del montaje realizado.

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico mediante visión y robótica colaborativa.

Por último, se presentan las conclusiones, resumiendo lo logrado, discutiendo implicaciones de los resultados y proponiendo futuras recomendaciones y mejoras para investigaciones posteriores relacionadas con este tema.

A continuación, se muestra un diagrama mostrando de las distintas conexiones y partes que conforman este proyecto:

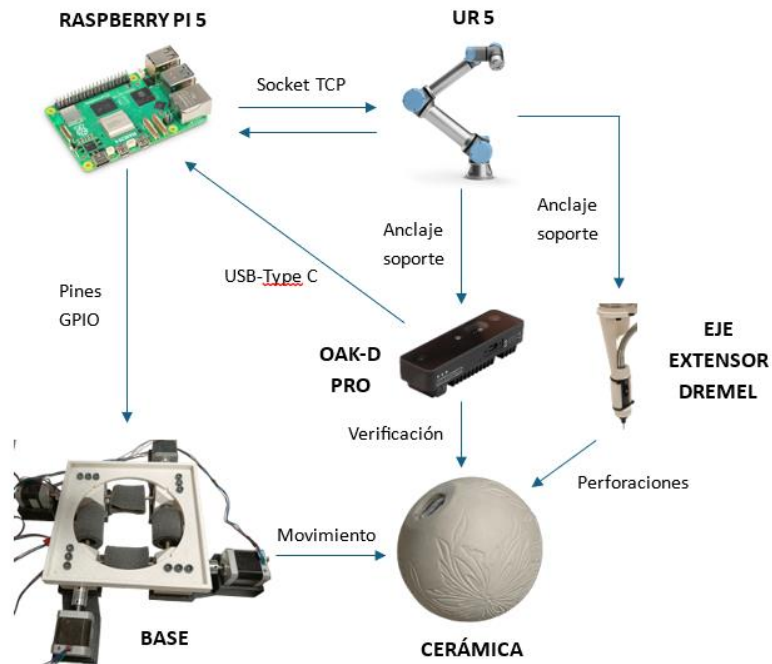


Ilustración 2. Esquema gráfico del planteamiento del proyecto.

Capítulo 2

2. Fundamentos teóricos

2.1 Industria 4.0

Industria 4.0 es nombre por el que se denomina la cuarta revolución industrial, que se caracteriza por usar técnicas avanzadas de producción (automatización) junto con tecnologías inteligentes, como son la robótica, inteligencia artificial, analítica, internet, computación en la nube, etc. Esta revolución se inició la década de 1950 con las primeras apariciones de sistemas de control numérico, progresando conforme iban surgiendo nuevas tecnologías como la robótica y Big data.

La Industria 4.0 se basa en diversos principios en los que muchas compañías y empresas se basan para implementarlo. Estos son:

- Interoperabilidad: Este principio se basa en la interconexión de los sistemas, máquinas y personas. La comunicación entre estos sistemas permite un flujo de datos continuo a alta velocidad que sienta las bases de esta revolución, debido que esto permite que, por ejemplo, sensores, actuadores y programas informáticos puedan comunicarse de forma casi inmediata para poder prever algún problema y ayudar a tomar decisiones para optimizar la eficiencia y la productividad.
- La nube: Debido al gran flujo de información, este principio es clave para el funcionamiento de la industria. Poder almacenar y acceder a datos desde cualquier lugar permite crear gemelos digitales (copias de sistemas virtuales del mundo físico) además de analizar en cualquier momento el funcionamiento del sistema, además de poder crear copias de seguridad para que, en caso de mal funcionamiento o de error, se pueda diagnosticar o usar una versión anterior y continuar operando.
- Asistencia al humano: Toda la información y el procesado de esta es vital para asistir a las personas tomar decisiones informadas y resolver urgencias a través de obtener información comprensible. Además, esto añade diversas posibilidades, como la posibilidad de que los sistemas realicen tareas desagradables o inseguras para los humanos o de tener la capacidad de tomar decisiones y desarrollar tareas de forma autónoma (solo en excepciones de interferencias se relega a un nivel superior). También se integran en esta parte los robots

colaborativos, creados para que puedan operar a la par con humanos de forma segura e intuitiva.

- Modificabilidad y reusabilidad: Facilitar la implementación del sistema diseñado de forma que permita la modificación del sistema. Además, se busca capacidad de poseer un sistema modular permitiendo poder reemplazar componentes y mejorar de forma continua el sistema.
- Realidad virtual y simulaciones: Este principio se fundamenta en la gran capacidad de procesamiento de los equipos actuales. Añadiendo el desarrollo de visión e inteligencia artificiales, esta capacidad de procesamiento permite clasificar e intercambiar de información junto a crear simulaciones para predecir cualquier cambio antes de implementarlo o prevenir algún tipo de fallo o error.
- Ciberseguridad: Este es un pilar fundamental de la nueva revolución industrial, pues toda la información circulante y almacenada se encuentra dentro de la misma red. Esto hace que esta sea susceptible a ser robada para poder obtener ventajas fuera del marco legal de las empresas, creando una amenaza importante para aquellas que contienen datos comprometidos de otras empresas o personas particulares.

2.2 Robots Colaborativos de Universal Robots

La definición de robot colaborativo (también llamado “cobot”) es la de un robot industrial articulado que se caracteriza por tener unas dimensiones pequeñas o medianas que permiten el trabajo en paralelo con humanos sin necesitar un cercado de seguridad. Esto es debido a que fueron concebidos con la finalidad de operar en tareas automatizadas junto a operarios. A continuación, se exponen las características principales de los cobots, las cuales sirven como diferenciadores respecto a otros robots industriales:

- **Tamaño**: Los robots colaborativos de UR son de un tamaño menor en comparación de otros fabricantes. Esto limita su capacidad máxima de carga, pero para muchas aplicaciones no es necesario tener una excesiva capacidad, además de que se van creando nuevos modelos como el UR30 con una carga útil de 30 kg.
- **Programación**: Estos robots están diseñados para ser más sencillos que otros modelos, facilitando así su implementación y por consiguiente mejorando el confort de los empleados que no deben realizar tareas repetitivas. Además, incorporan una visualización intuitiva 3D que ayuda a determinar los movimientos y trayectorias del robot antes de ponerlo en funcionamiento junto un modo que permite guiar el robot a la posición deseada con la mano y guardar la posición.
- **Versatilidad**: Universal Robots diseña los robots para ser versátiles. Debido a la fácil programación e implementación de los robots, pueden ser usados para una gran variedad de tareas, incluyendo soldadura, ensamblaje, tallados, etc.

- Seguridad: Los cobots de Universal Robots están diseñados de forma que se puede obtener una representación del espacio de alrededor del robot. Esto junto a los sensores y su bajo peso permite que se puedan detener fácilmente y por ello es posible reducir las medidas de seguridad externas.

Universal Robots fue fundada en 2005 por Esben Østergaard en Dinamarca y es una de las primeras que introdujo los robots colaborativos en la industria de manufacturación. Esta empresa revolucionó la industria en 2008 con el lanzamiento del UR5, un robot de dimensiones medianas que permitía un trabajo colaborativo humano-robot añadiendo componentes de seguridad a un precio inferior al de la media de sus competidores. Con el paso del tiempo, Universal Robots ha avanzado desarrollando una variedad de robots de distintos tamaños y características con mejoras en precisión, capacidad de carga, utilidades y facilidad de programación y comunicación. El éxito de dicha empresa demuestra la capacidad de democratización de una parte emergente de la industria como es la robótica, permitiendo que el sector se adapte a las necesidades de cada empresa de forma efectiva y rentable. Hoy en día, presentan una posición influyente en la industria permitiendo la continua innovación y asistencia a los clientes.

Como muchos otros robots, los robots de Universal Robots presentan una tableta (teach pendant) que sirve como puerto de entrada principal del robot. En esta tableta es posible programar mediante interfaces gráficas de usuario (GUI) intuitivas. En esta tableta encontramos el entorno de programación de UR, teniendo una configuración que permite crear programas a través de bloques, diagramas de flujo e hilos en paralelo facilitando su uso y comprensión. El entorno de programación cuenta con plantillas y funciones simplificadas que facilitan la creación de programas al no tener que incluir el código directamente. Además, el robot cuenta con varios sistemas de comunicación que permiten la interacción entre distintos componentes (manipuladores, sensores, sistemas de visión, etc) para poder extender aún más las posibilidades de funcionalidades e interacción del robot con su entorno.

Otra de las funciones más destacables de un cobot de UR es la capacidad de programar el robot a través de la manipulación directa por un humano. Esta forma de programar permite usar un modo en el que el robot solo compensa la fuerza de la gravedad y a través del manejo con la mano se pueden programar puntos e incluso trayectorias que se registrarán y se creará de forma automática un código replicando los movimientos. Todo esto ayuda a realizar una programación intuitiva y fácil de comprender.

Otra herramienta que proporciona Universal Robots es URSim, un entorno de simulación que permite programar un UR virtual para depurar y probar código y programas antes de su implementación. Este entorno tiene diversos modelos de robot los cuales se pueden ejecutar de forma parecida a un gemelo virtual, y su uso es intuitivo, pues que la aplicación trata de una simulación de un teach pendant con todas sus funciones con cada modelo de robot. URSim se puede ejecutar tanto Linux (o Windows usando una máquina virtual) o encapsulado para un sistema Docker.



Ilustración 3. Imagen de un teach pendant perteneciente a un UR 5. Fuente: Automation Distribution.

2.3 OAK-D Pro de Luxonis

Luxonis es una empresa que se centra en la unión entre inteligencia artificial y cámaras de visión espacial. Sus productos se centran en el diseño y desarrollo de hardware, firmware, software e interfaces para usuarios usando inteligencia artificial y aprendizaje computacional, ayudando a clientes de todo el mundo a obtener conceptos y prototipos usando y procesando información espacial (como por ejemplo la profundidad).

Fundada en 2011 por 4 personas (Shawn McLaughlin, Brian Weinstein, Vlad Vyskocil, and Brandon Gilles), cuentan con repertorio considerable de modelos de cámaras que permiten integrar la computación y la sensorización espacial a través de la visión usando tanto Python como C++.

Un ejemplo del modelo que se va a usar en el proyecto trata de OAK-D Pro. Esta cámara cuenta con un proyector de luz infrarroja que mejora la percepción de la profundidad, sobre todo en ambientes con baja luz. Cuenta también con visión nocturna con una cámara central de 12 MP con 2 cámaras laterales usadas para obtener una mayor calidad de profundidad (altamente configurables con capacidad de filtrado, post-procesado y alineamiento RGB). Permitiendo usar modelos de inteligencia artificial y visión computacional, esta cámara es ideal para los requerimientos del proyecto, puesto que es capaz de obtener imágenes 4K a 30 fps o 1080p a 60 fps; además de ser capaz de tener seguimiento a objetos 2D y 3D. Luxonis también cuenta con módulos para crear prototipos y en sistemas fijos junto a diversos accesorios que permiten una mejor manipulación y uso de los productos.

Luxonis cuenta con DepthAI API, desarrollada para permitir que los usuarios desarrollen y usen “conductos” (pipelines) para usar en los bloques acelerados de productos que ofrece la empresa. Este sistema es altamente configurable y funciona a través de la interconexión de nodos. También es posible usar DepthAI Viewer, una herramienta que permite modificar, personalizar y verificar cada cámara del producto de forma inmediata con un interfaz sencillo, sin tener que usar código.

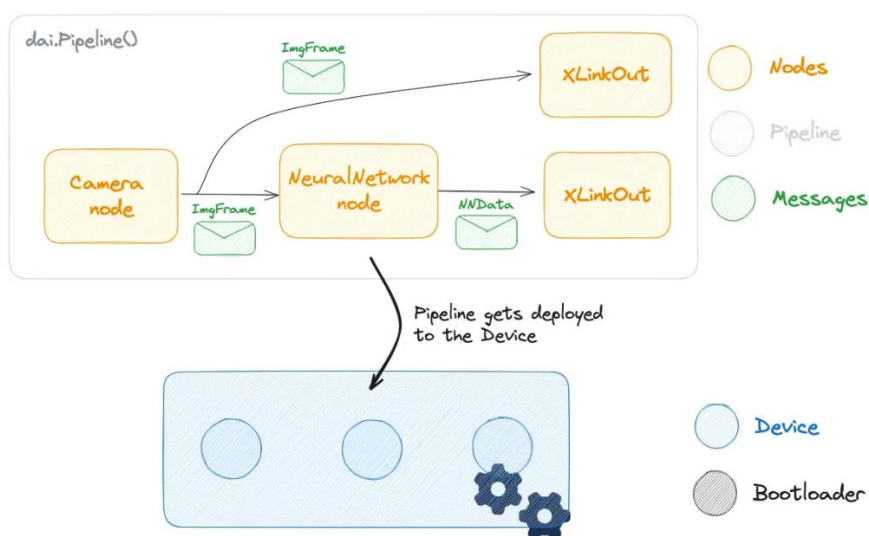


Ilustración 4. Esquema visual sobre DepthAI API. Fuente: Luxonics.

2.4 Raspberry Pi 5 de Raspberry Pi

Raspberry Pi es una empresa dedicada a crear placas computacionales para promover la informática en colegios y lugares del mundo en vías de desarrollo. Puesto que las placas son económicas, cuentan con un tamaño bastante reducido (similar al de una tarjeta de crédito) y se pueden usar como un ordenador basado en un sistema operativo Linux (Raspberry Pi OS) se han usado como base de muchos proyectos en todas las partes del mundo.

Fundada en 2009 por como una asociación caritativa regulada por la comisión de Inglaterra y Gales, no fue hasta 2011 que se realizó el lanzamiento oficial de su primera placa desarrollada en diciembre de este año.



Ilustración 5. Reunión en 2012. De izquierda a derecha: Martin Cartwright, Profesor Alan Mycroft, Pete Lomas, David Braben, Eben Upton Jack Lang and Alex Bradbury. Fuente: Raspberry Pi Foundation.

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico mediante visión y robótica colaborativa.

A pesar de que el lanzamiento se realizara en 2011, el desarrollo de las placas comenzó en 2006, creando prototipos inspirados en la BBC Micro.

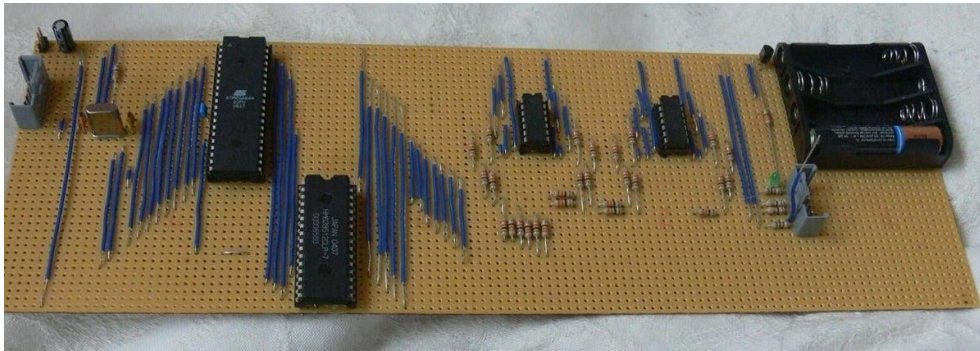


Ilustración 6. Primer prototipo Raspberry Pi (2006, con el mismo rendimiento que BBC Micro de 1980s). Fuente: Raspberry Pi Foundation.

Como ejemplo de un modelo actual se muestra el modelo usado en el proyecto, una Raspberry Pi 5 pues fue proporcionada para ello. Una Raspberry Pi 5 se trata de un dispositivo (una placa concretamente) de desarrollo de código abierto que mejora el rendimiento y la capacidad de procesamiento respecto a los dispositivos predecesores.



Ilustración 7. Raspberry Pi 5.

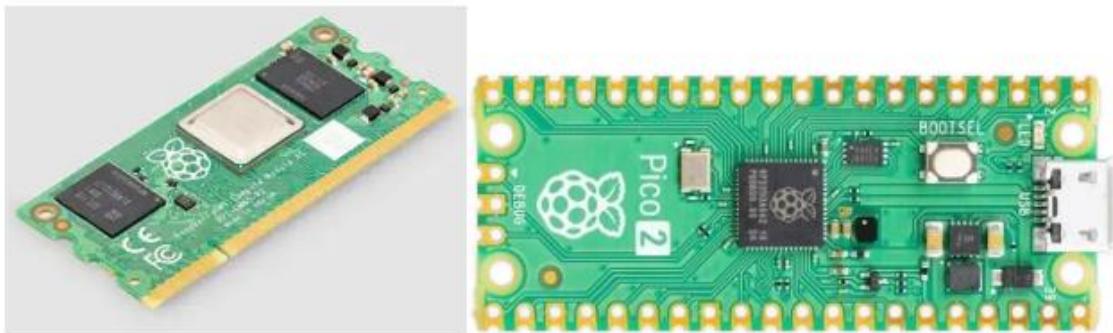
Esta placa cuenta con las siguientes características:

- **Procesador:** Broadcom BCM2712 Quad-core Arm Cortex-A76 a 2,4 GHz
- **GPU:** VideoCore VII de 800 MHz con salida de pantalla dual 4Kp60 a través de HDMI
- **Memoria:** 2 GB, 4GB o 8GB LPDDR4X (4 Gb en este caso)
- **Almacenamiento:** Ranuras microSD y opciones para unidades SSD M.2.
- **Conectividad:** Wi-Fi 5 (doble banda), Bluetooth 5.0, Gigabit Ethernet (PoE+), USB 2.0/3.0, Pines GPIO

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico mediante visión y robótica colaborativa.

Estas placas cuentan también con diversos accesorios como carcasas, fuente de alimentación de 27 V, disipadores con ventilación forzada (o sin), etc.

Además de las Raspberry Pi principales (las numeradas), también se encuentran las Raspberry Pi Zero, las cuales son versiones más compactas y con menos potencia, y versiones de esta con partes ya integradas (como algunas con wifi y bluetooth, otras con el cabezal de GPIO soldado para fácil acceso). Otro ejemplo son las Raspberry Pi Pico, siendo similar a un Arduino Nano no siendo un “ordenador completo”. Los últimos productos constan de Raspberry Pi Pico 2 (mejora respecto la anterior) y Raspberry Pi Compute Module 4S, siendo este último un módulo preparado con un conector SODIMM compatible con Compute Module 3 siendo diseñado para integrarse en otro producto (TV, placa de desarrollo, equipamiento industrial, etc).



*Ilustración 8. Placas mencionadas: a la izquierda Raspberry Pi Compute Module 4S y a la derecha Raspberry Pi Pico 2.
Fuente: Raspberry Pi Foundation.*

La Raspberry Pi 5 tiene diversas formas de comunicarse con otros dispositivos. Entre ellos cuenta con diversos puertos de conexión como se ha mencionado anteriormente, siendo destacables el Ethernet, Wi-Fi y salidas de video que permiten 4K a 60 hercios; pero la más importante para este proyecto trata de los pines GPIO, mostrándose a continuación.



Ilustración 9. Raspberry Pi 5.

Estos son unos pines de propósito general entrada/salida teniendo varios grados de configurabilidad.

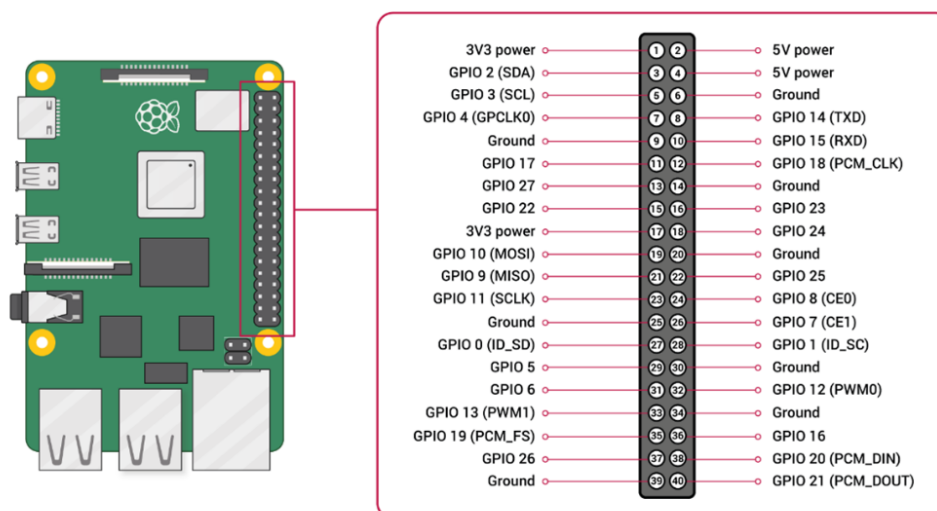


Ilustración 10. Configuración pines GPIO Raspberry Pi 5. Fuente: Raspberry Pi Foundation.

Cualquiera de los pines se puede configurar (en software) como un pin de entrada o de salida, pero hay que tener en cuenta que estos no se encuentran en orden numérico, como se muestra en la imagen anterior. Además, algunos de estos pines tienen funciones específicas reservadas para usos avanzados, como los pines GPIO 0 (pin físico 27) y GPIO 1 (pin físico 28).

A continuación, se muestran características de estos:

- **Voltajes:** Dos pines de 5 V y otros dos de 3,3 V están presentes en la placa (número 2 y 4; número 1 y 17 respectivamente), así como algunos pines de tierra (GND = 0 V, número 6, 9, 14, 20, 25, 30, 34, 39). Estos no son configurables, pero los restantes son pines de 3V3 de uso general (es decir, toleran 3V3 como entradas y como salidas proporcionan 3V3).
- **Entradas:** Un pin GPIO reservado como entrada se puede interpretar como alto (3V3) o como bajo (0V). Esto se facilita a través del uso de resistencias internas pull-up o pull-down. Para todos los pines estas resistencias se pueden configurar a través de software excepto los pines GPIO 2 y GPIO 3 (número 3 y 5), puesto que poseen resistencias fijas.
- **Salidas:** Un pin GPIO designado como salida puede proporcionar 3V3 o 0 V configurándose en alto o bajo, respectivamente.
- **PWM:** Son pines que permiten la modulación de ancho de pulso, permitiendo así transmitir información a través de un canal de comunicaciones o controlar una cantidad de energía que se envía a una carga o dispositivo (por ejemplo, un motor eléctrico de pasos). El software se encuentra disponible en todos los pines, pero el hardware exclusivamente en GPIO 12, GPIO 13, GPIO 18 y GPIO 19 (número 32, 33, 12 y 35).
- **SPI:** EL Bus SPI (Serial Peripheral Interface) es un tipo de comunicación considerado estándar, usado para la transferencia de información entre circuitos integrados electrónicos. Para SPI 0 existen: MOSI (GPIO 10); MISO (GPIO 9); SCLK (GPIO 11); CE0 (GPIO 8), CE1 (GPIO 7). Para SPI

1 se tienen: MOSI (GPIO 20); MISO (GPIO 19); SCLK (GPIO 21); CE0 (GPIO 18); CE1 (GPIO 17); CE2 (GPIO 16).

- **I2C:** Circuito Integrado Interno (Inter-Integrated Circuit) es usado internamente para la comunicación entre diferentes partes de un circuito, como por ejemplo entre un controlador y circuitos periféricos integrados. Para ello se tiene: datos (GPIO 2); reloj (GPIO 3), datos EEPROM (GPIO 0), reloj EEPROM (GPIO 1).
- **Serie:** Son puertos para comunicación UART (Universal Asynchronous Receiver-Transmitter), usando transmisión TX (GPIO 14) con un de recepción RX (GPIO 15).

En general, existen dos formas de numerar los pines, siendo esto modo GPIO o modo BCM. Esto rebosa importancia a la hora de programar, puesto cuando se controlan los GPIO con Python (u otro lenguaje) se ha de indicar en qué modo se usarán dichos pines (GPIO.BOARD o GPIO.BCM).

Para el caso del uso en GPIO.BOARD, es necesario referirse a los pines por su número impreso en la placa, siendo este el número que aparece redondeado en la imagen anterior (fila izquierda con impares y fila derecha pares).

En el caso de GPIO.BCM, es necesario referirse a los pines por el número "Broadcom SOC channel". Esto corresponde a la nomenclatura utilizada en la numeración del procesador y los BOARD del conector. En la imagen anterior, los BCM corresponden a aquellos que tienen precedente en el nombre "GPIO".

Para ver los pines en la Raspberry, es posible usar comandos en la consola. Ejecutando GPIO readall, se leen todos los pines pertenecientes a GPIO con una tabla informativa como aparece en la imagen a continuación:

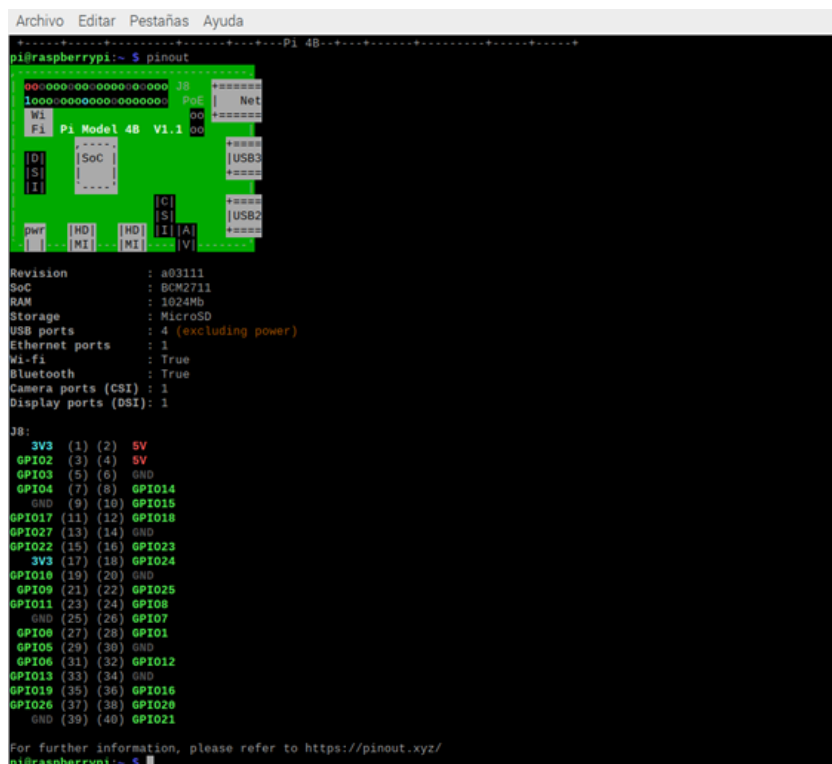
```

pi@raspberrypi:~/tmp $ cd
pi@raspberrypi:~/tmp $ gpio readall
.....Pi 4B.....
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 8 | 3.3v | | | 1 | 2 | | | 5v | | |
| 3 | 9 | SDA.1 | IN | 1 | 3 | 4 | | | 5v | | |
| 4 | 7 | SCL.1 | IN | 1 | 5 | 6 | | | 0v | | |
| 4 | 7 | GPIO.7 | IN | 1 | 7 | 8 | 1 | IN | Tx0 | 15 | 14 |
| | | | | | 9 | 10 | 1 | IN | Rx0 | 16 | 15 |
| 17 | 0 | 0v | | | | | | | | | | |
| 17 | 0 | GPIO.0 | IN | 0 | 11 | 12 | 0 | IN | GPIO.1 | 1 | 18 |
| 27 | 2 | GPIO.2 | IN | 0 | 13 | 14 | | | 0v | | |
| 22 | 3 | GPIO.3 | IN | 0 | 15 | 16 | 0 | IN | GPIO.4 | 4 | 23 |
| | | | | | 17 | 18 | 0 | IN | GPIO.5 | 5 | 24 |
| 10 | 12 | 3.3v | | | | | | | | | | |
| 9 | 13 | MOSI | IN | 0 | 19 | 20 | | | 0v | | |
| 11 | 14 | MISO | IN | 0 | 21 | 22 | 0 | IN | GPIO.6 | 6 | 25 |
| | | | | | 23 | 24 | 1 | IN | CE0 | 10 | 8 |
| 0 | 30 | SCLK | IN | 0 | 25 | 26 | 1 | IN | CE1 | 11 | 7 |
| | | | | | 27 | 28 | 1 | IN | SCL.0 | 31 | 1 |
| 5 | 21 | 0v | | | | | | | | | | |
| 6 | 22 | SDA.0 | IN | 1 | 27 | 28 | 1 | IN | | | |
| 13 | 23 | GPIO.21 | IN | 1 | 29 | 30 | | | 0v | | |
| 19 | 24 | GPIO.22 | IN | 1 | 31 | 32 | 0 | IN | GPIO.26 | 26 | 12 |
| 26 | 25 | GPIO.23 | IN | 0 | 33 | 34 | | | 0v | | |
| | | | | | 35 | 36 | 0 | IN | GPIO.27 | 27 | 16 |
| | | | | | 37 | 38 | 0 | IN | GPIO.28 | 28 | 20 |
| | | | | | 39 | 40 | 0 | IN | GPIO.29 | 29 | 21 |
|-----|-----|-----|-----|---|-----|---|-----|-----|-----|-----|
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
.....Pi 4B.....
pi@raspberrypi:~/tmp $

```

Ilustración 11. Obtención por pantalla al ejecutar "gpio readall". Fuente: <https://solectroshop.com/>.

Otro ejemplo se trata de pinout, mostrando el siguiente esquema visual:



```
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ pinout
-----Pi 4B-----
00000000000000000000 JB =====
10000000000000000000 PoE | Net
-----
[Wi] Pi Model 4B V1.1 00 =====
-----
[IO] [SOC] =====
[SI] [SOC] =====
[II] [SOC] =====
-----
[OW] [HD] [HD] [II] [A] =====
[MI] [MI] [MI] [VI] =====
-----
Revision      : a03111
SoC           : BCM2711
RAM           : 1024MB
Storage       : MicroSD
USB ports     : 4 (excluding power)
Ethernet ports : 1
Wi-fi        : True
Bluetooth    : True
Camera ports (CSI) : 1
Display ports (DSI) : 1

JB:
  3V3 (1) (2) 5V
  GPIO2 (3) (4) 5V
  GPIO3 (5) (6) GND
  GPIO4 (7) (8) GPIO14
  GND (9) (10) GPIO15
  GPIO17 (11) (12) GPIO18
  GPIO27 (13) (14) GND
  GPIO22 (15) (16) GPIO23
  3V3 (17) (18) GPIO24
  GPIO18 (19) (20) GND
  GPIO9 (21) (22) GPIO25
  GPIO11 (23) (24) GPIO8
  GND (25) (26) GPIO7
  GPIO8 (27) (28) GPIO1
  GPIO5 (29) (30) GND
  GPIO6 (31) (32) GPIO12
  GPIO13 (33) (34) GND
  GPIO19 (35) (36) GPIO16
  GPIO26 (37) (38) GPIO28
  GND (39) (40) GPIO21

For further information, please refer to https://pinout.xyz/
pi@raspberrypi:~$
```

Ilustración 12. Obtención por pantalla al ejecutar "pinout". Fuente: <https://solectroshop.com/>.

Para controlar GPIO en Python, es necesario importar una librería de código escrita previamente.

El modo más sencillo de controlar estos pines consiste en usar comandos sobre la consola. Una vez abierta la consola, se ejecuta el comando "ls /sys/class/gpio/". /sys es un directorio virtual en el que las referencias a los archivos son manejadas por el kernel (núcleo del SO). Además, se usan controladores de clase GPIO.

Tras hacer esto, se observa que ya se encuentran algunos archivos en este. Para poder manipular GPIO, es necesario realizar el comando "echo xx > /sys/class/gpio/export" siendo "xx" el número de GPIO deseado y usando "export" para incluirlo en el directorio virtual.

Una vez hecho esto, hay que declarar si se quiere el pin de entrada o de salida. De forma predeterminada se configura como entrada (in). Para obtener una salida, habría que utilizar el siguiente comando "echo out > /sys/class/gpio/gpioxx/direction" siendo "xx" el número de GPIO.

Una vez definido como salida, se usa 1 o 0 para poner en "alto" o "bajo" respectivamente usando "echo" como en el siguiente ejemplo: "echo 1 > /sys/class/gpio/gpio21/value". Para deshabilitar el acceso al pin, es necesario usar un comando similar para exportarlo, pero usando "unexport": "echo xx > /sys/class/gpio/unexport".

Cabe destacar que a través del acceso "directorio/sys/class/gpio" también se puede acceder a los pines en otras plataformas, pues esta funcionalidad viene proporcionada por el kernel de Linux.

Existen formas más intuitivas para usar, puesto que la comunicación directa con kernel puede no ser muy efectiva. Debido a esto, existen bibliotecas y programas que permiten acceder a los pines de forma más eficiente. Un ejemplo se trata de gpio-. Instalando WiringPi es posible ejecutar gpio usando "gpio

-v" en la consola. Wiringpi usa el sistema BCM, pero es más sencillo de usar. Por ejemplo, para configurar el pin GPIO 12 (número 32) como salida solo se ha de escribir "gpio -g mode 12 out" en la consola. Después, para cambiar a modo alto o bajo, solo es necesario elegir 1 o 0 en el siguiente comando "gpio -g write 12 1".

Capítulo 3

3. Base y soportes

Durante la primera sección del capítulo, se va a abordar el proceso de desarrollo de la base de la cerámica, pues es una de las partes más importantes del proyecto debido a su complejidad. Los procesos de desarrollo de la base se han realizado de manera iterativa como se puede comprobar en el anexo B. Debido a esto, este capítulo es el más extenso.

El diseño y configuración de la base han sido aspectos fundamentales en el desarrollo del proyecto, pues de ello dependía cómo se procede con la programación y el montaje del sistema. Además, se explicará el funcionamiento de esta, aunque el control se realice con una Raspberry Pi 5. En los capítulos siguientes se detallará el funcionamiento desde el punto de vista del controlador.

También se expondrá el proceso de creación de los otros soportes realizados para poder acoplar las herramientas al robot a través del diseño en 3D. Se tratan de dos soportes: uno para la cámara OAK-D Pro y otro para la Dremel usada para realizar las perforaciones.

3.1 Creación de la base

3.1.1 Consideraciones previas al diseño

En esta parte se tratarán los aspectos clave del diseño de la base, siendo el núcleo del proyecto. También es importante la correcta selección de los componentes para asegurar la compatibilidad entre todos los dispositivos que se van a usar.

3.1.1.1 Puntos clave a resolver

La pieza por agujerear se trata de una esfera cerámica (sin cocinar) de unos 115 mm de diámetro con motivos florales tallados además de tener una sección “cortada” (que sirve como base de apoyo en superficies lisas).



Ilustración 13. Estados de la cerámica: izquierda sin tratar, centro tratada y derecha es un prototipo.

Debido a estas características tan peculiares, es necesario cubrir una serie de puntos:

- **Manipulación:** Debido a la fragilidad y al material, se debe crear una forma de sujeción que permita el movimiento preciso y libre del sistema.
- **Resistencia:** Además de sustentar la pieza, se debe asegurar que la pieza no se mueva para obtener un agujereado preciso y limpio. También debe soportar la presión mientras se realiza la operación.
- **Presupuesto:** Debido a ser un proyecto tipo experimental, será necesario pasar por diversas iteraciones y versiones hasta conseguir una configuración que se acomode a las necesidades. Es por ello por lo que es necesario que el sistema sea modular para mejorar partes conforme surgen nuevas necesidades del sistema además de para poder cubrir las nuevas ideas durante las iteraciones de la plataforma base.
- **Residuo:** La cerámica deja un polvo residual que mancha y genera pérdidas de tracción en diversos materiales, además de ensuciar el lugar de trabajo, por lo que se ha de diseñar una base de acuerdo con esto.
- **Peso:** Al tratarse de un prototipo, es importante que la estructura que se cree para la sujeción y manipulación de la cerámica sea fácil de transportar y manipular puesto que se van a crear diversas iteraciones y modelos.

3.1.2 Diseño de la base

Debido a lo anterior, al final se decidió usar máquinas de impresión 3D con filamento PLA como material seleccionado. Esto se debe a su fácil manipulación y a su poco peso, puesto que, al tener la resistencia suficiente para aguantar las fuerzas que se implementan, se comporta de forma predecible y resiste sin problemas. Además, gracias a su flexibilidad permite realizar piezas de moderada complejidad como soluciones a problemas que surgen durante el diseño del sistema.

Se usarán 2 modelos de máquinas encontradas en los laboratorios de AI2.

3.1.2.1 Primera versión

Como primera idea, se tenían 2 posibilidades de operar el sistema: una en la que la pieza estuviera inmóvil u otra en la que el brazo robot sujetara y moviera la pieza para realizar las perforaciones necesarias.

Debido a las características de la cerámica, la segunda opción no resultaba viable, ya que se debería crear un prototipo en el que se debería usar aire comprimido para sujetar la pieza por el agujero de la sección a través de unas pinzas asegurándola, siendo complejo debido a su fragilidad y a la no uniformidad de las piezas por el uso de distintas baterías de moldes. Además, se consideró más eficiente tener el sistema de visión conectado al brazo robot, ya que las verificaciones se podrían realizar de forma más rápida sin tener que mover el brazo a posiciones no naturales para verificar la integridad y la correcta perforación de la pieza.

Es debido a esto que se optó como primera instancia a crear una plataforma cilíndrica con un cilindro en el interior que encajara en el agujero de la cerámica para estabilizar. Además, se dispone de una hendidura donde se encaja un soporte semiesférico que evitaría que la pieza se mueva cuando se perfore.

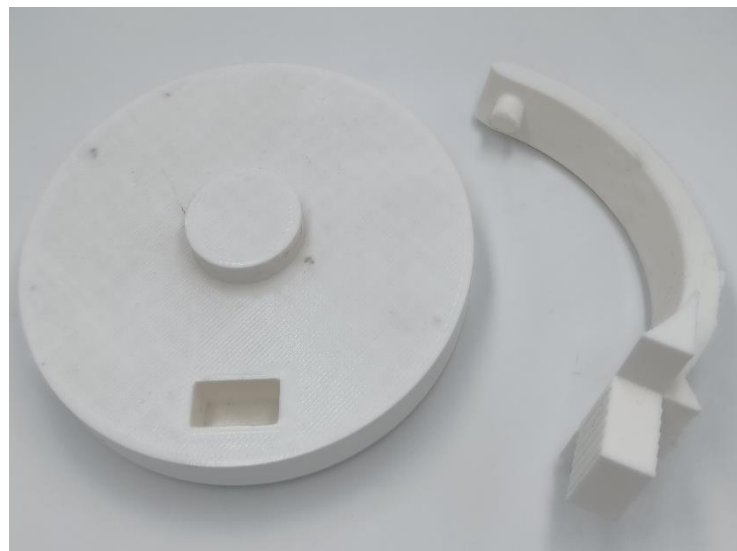


Ilustración 14. Primer prototipo de base.

Una vez se obtuvo este prototipo, se observó que para realizar las perforaciones en la parte inferior de la pieza se tendrían problemas al no tener un acceso claro. Es por ello por lo que se decidió crear una segunda versión del prototipo el cual solo tiene una sección del cilindro mayor con una zona alrededor del cilindro interior que sobresale para actuar como soporte de la pieza además de dejar acceso a la mayor parte posible de la cerámica. A su vez, se mejoró el soporte para que la pieza no se moviera al agujerearla ampliando horizontalmente una sección para asegurar el menor movimiento posible.

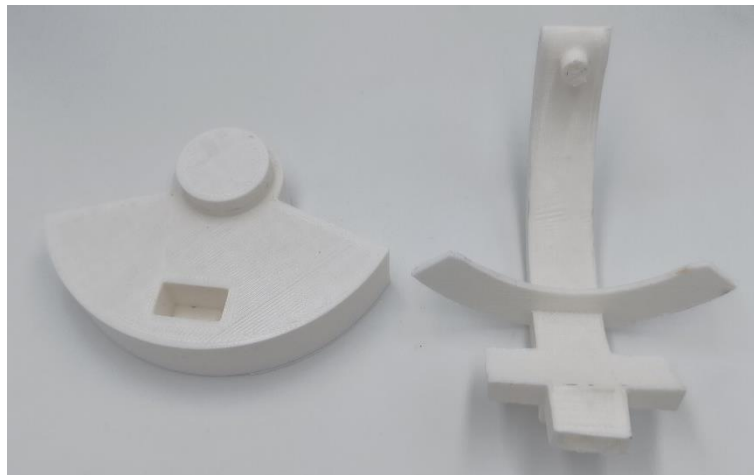


Ilustración 15. Segunda iteración de primer prototipo.

Tras diversas pruebas, se pudo comprobar que este sistema presentaba diversos inconvenientes.

En primer lugar, no se podían obtener las tolerancias adecuadas para evitar que la cerámica se moviera durante la perforación, además de que las cerámicas no podían ajustarse correctamente debido a las diferencias de cada una por ser creadas en distintos moldes de una batería.

Otro inconveniente reside en la propia forma de la base. Debido a esta la base debería estar en voladizo para poder acceder a los puntos inferiores de la cerámica, ya que si está en contacto con una superficie resulta imposible posicionar el robot para realizar las perforaciones, a no ser que se alargara verticalmente la base para crear distancia (esto no es conveniente pues por la fuerza de perforación se crearía un momento que tumbaría el sistema).

Además, se puede observar que por la forma compleja y curvada de la base resulta muy complicado poder girar y maniobrar la pieza con algún motor sustentado. En un primer momento se planteó la posibilidad de poder rotar la pieza con el brazo robot (usando alguna pinza) pero fue rápidamente descartado por la falta de practicidad. También se acumula el problema del peso del motor en el sistema, que como se ha mencionado anteriormente generaría un momento que desequilibraría la base a no ser que estuviera empotrada en voladizo.

Es por todo ello que se decidió descartar este sistema, ya que anclar la base a la estación donde está el robot o elevarla para acceder a las zonas inferiores no era factible, además de que la base diseñada no permitía mover la cerámica con la libertad y precisión deseadas.



Ilustración 16. Pieza montada en segunda iteración de primer prototipo.

3.1.2.2 Segunda versión

Debido a las conclusiones obtenidas anteriormente, se decidió rediseñar el sistema para hacerlo más robusto y funcional. Gracias a la recomendación de mi jefe Paco, que mencionó cómo se realizaba la operación manual en la empresa, se propuso un sistema en que la cerámica se encuentre centrada y suspendida entre unos rodillos. Esta configuración cubre muchas de las necesidades mencionadas anteriormente, pero es necesario pulir el sistema para que funcione correctamente.

Como primer paso, era crucial elegir qué configuración se elegiría. Debido a que una configuración estable se consigue con mínimo 3 puntos de apoyo, se plantearon diversas opciones.

La primera opción era crear la base con 3 puntos de apoyo, formando así una base triangular. La idea principal es encajar la cerámica en el centro, quedando en suspensión en unos rodillos que servirían como sustento.

Uno de los problemas que se encontró con este diseño es que debido al ángulo en el que se colocan los lados (suponiendo todos los lados iguales sería de 60 grados), se tendría que usar rodillos con demasiado diámetro para que tuviera contacto con la pieza, ya que los lados deberían ser bastante largos para que esta encajara en el interior de forma centrada. Además, debido al ángulo en que se deberían poner los lados, el cálculo de los giros y radianes que debería girar cada rodillo para moverse en la dirección adecuada se hacía muy complejo, añadiendo que, al no estar en pares, los rodillos no motrices ejercerían fricción sobre la cerámica a no ser que se realizara un cálculo concreto para el movimiento de cada uno, dificultando así más el proyecto.

Tras descartar la opción de usar 3 puntos de apoyo, se decidió realizar un prototipo con 4 puntos de apoyo.

Esta configuración presenta una serie de ventajas y desventajas. Las ventajas con las siguientes:

- Estabilidad: Al usar 4 lados, la cerámica cuenta con más puntos de apoyo que facilitan la estabilidad y el apoyo de esta cuando se están realizando perforaciones, evitando movimientos que puedan dañar la pieza o la broca usada.

- Control: Debido a que se usan 4 rodillos, estos se pueden configurar de diferente manera dependiendo de la funcionalidad requerida del sistema. Se pueden configurar como 2 motores que mueven a través de una serie de poleas un par de rodillos cada uno o se pueden usar 4 motores independientes para cada rodillo para obtener un control independiente (o programar la paridad a la hora de mandar las señales al controlador).
- Fácil configuración: Debido a la flexibilidad del sistema, sobre todo usando 4 motores independientes, es posible crear y programar configuraciones de movimiento para mover la pieza en la dirección deseada o, en caso de que ocurra, desatascar la cerámica, ya que debido a su forma puede quedar en un punto muerto donde uno de los rodillos no alcanza a la pieza por la sección "cortada" que hace que no sea completamente esférica.
- Simplicidad de diseño: Debido a la simplicidad que presenta usar un cuadrado para el sistema, no es necesario diseñar las patas de la base con ciertos ángulos para evitar que los ejes de los rodillos se solapen en el interior de la pieza

Por otro lado, se aprecian las siguientes desventajas:

- Materiales: Debido a tener más puntos de apoyo, es necesario usar una mayor cantidad de material y crear piezas mayores (armazón de la parte superior de la base y el diámetro de los rodillos) que puede que no quepan en máquinas de impresión 3D disponibles. También en función de la configuración elegida, será necesario usar más o menos material para crear los soportes o mecanismos necesarios.
- Configuración: A pesar de ser un sistema modulable, modificar la configuración requiere algunos cambios estructurales. Por ejemplo, si se quiere realizar el sistema con solo 2 motores, es necesario elevar bastante el punto de apoyo pues es necesario usar un sistema de poleas a distinta altura (para cada par de rodillos) para mover de forma coordinada el par de motores opuestos que se necesita. En cambio, si se quiere realizar el control independiente (o a pares con programación) con 4 motores se tendrá que crear una serie de bases dependiendo del modelo de motor usado e implementarlos en el sistema.
- Fricción y deslizamiento: Debido a tener más puntos de apoyo, es necesario resolver el problema de la fricción. En la configuración con cuatro rodillos, la posición a 90 grados de los rodillos no motrices al momento de rotar la pieza provoca fricción que puede llegar a levantar la pieza y provocar que se pierda precisión en el movimiento de esta. El problema de esta fricción reside en encontrar el equilibrio entre fricción y deslizamiento, ya que si se reduce demasiado la fricción la pieza no giraría a pesar de rotar los rodillos.

Tras discernir sobre todas las implicaciones de realizar el sistema con distintos puntos de apoyo, se optó por usar cuatro. Esto es debido a que los inconvenientes de realizar el sistema con cuatro puntos de apoyo son fácilmente corregibles menos uno, en el que se dedicó una parte de la planificación exclusivamente a él. Al tener total libertad de usar los materiales que se quieran, incluyendo motores, se pudieron resolver los problemas de la forma más sencilla posible. En cambio, los inconvenientes

encontrados con el sistema de tres puntos de apoyo son más laboriosos de resolver, sobre todo la parte del diámetro de los rodillos, ya que se obtenían diámetros excesivos.

Como se observó que un sistema de cuatro puntos de apoyo resolvía gran parte de las necesidades requeridas, no se planteó ni se estudió usar una configuración con más cantidad de puntos de apoyo, puesto que esto crearía más problemas, sobre todo a la hora de realizar cálculos y configurar los ejes de los rodillos.

3.1.3 Desarrollo de la versión seleccionada

Una vez determinada la configuración que se va a usar se procedió a elaborar los primeros bocetos.

3.1.3.1 Primera iteración

Como primer paso se decidió crear los apoyos de la base. Estos son unos apoyos de 45 x 45 x 55 mm con una hendidura para encajar el eje del rodillo. Esta hendidura ha de ser lo suficientemente amplia para que el eje pueda entrar y luego encaje libre.

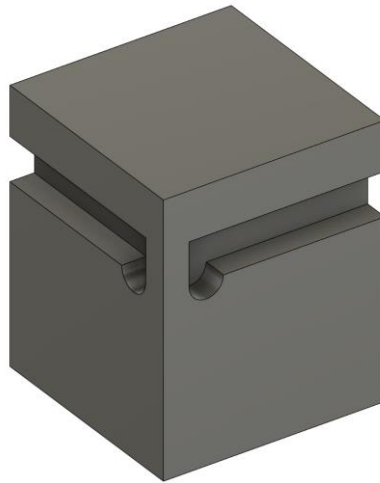


Ilustración 17. Primera iteración de un apoyo.

Una vez diseñados, se planteó la configuración de la parte superior. Esta parte está formada por un cuadrado de 190 x 190 mm el cual tiene en el interior un área vacía que permite colocar la cerámica, apoyando así en los rodillos inferiores. Este diseño también cuenta con una pared envolvente que rigidiza el sistema y unas paredes que seccionan el soporte en 4 partes. Esto es debido a que, en esta configuración, se pensó colocar los rodillos motrices en la parte superior del soporte, usando así los inferiores simplemente como apoyos. Los motores se colocarían en las esquinas del soporte y tendrían una pequeña zona donde el rodillo tendría contacto con la cerámica.

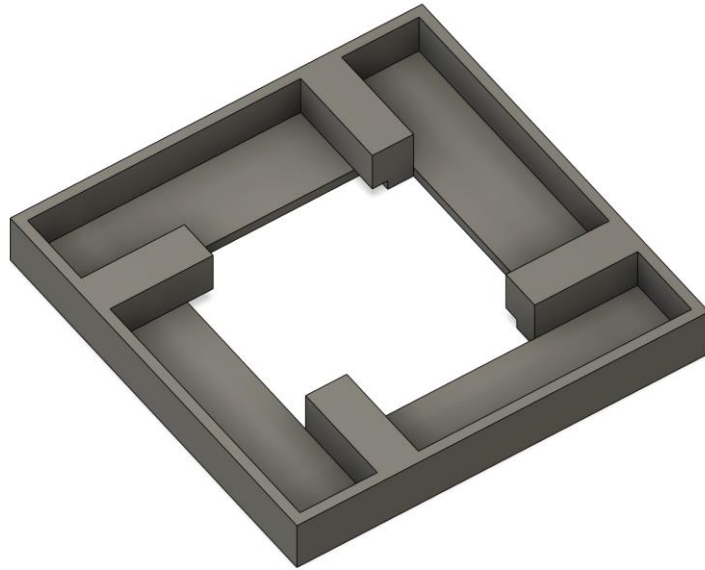


Ilustración 18. Primera iteración de la parte superior de la base.

A continuación, se muestra cómo queda el prototipo realizado a través de Autodesk Fusion como primera iteración.

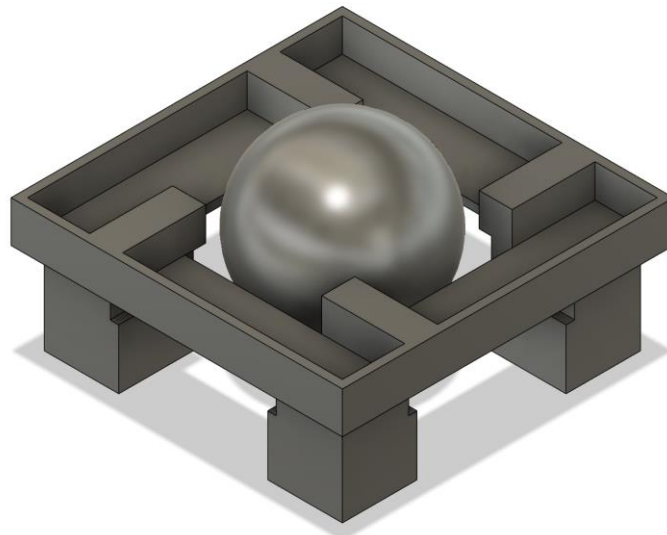


Ilustración 19. Ejemplo modelado con las piezas anteriores.

Este prototipo presenta unas ciertas desventajas:

- Rodillos superiores: En la parte superior, se puede observar que el sistema no tiene preparado los alojamientos para poner rodamientos o crear una sujeción correcta de los ejes que conforman los rodillos motrices del sistema.

- Rodillos inferiores: Se observa que el alojamiento de los rodillos inferiores no contempla el uso de rodamientos para poder encajar los ejes. Esto es un gran problema pues si no hay rodamientos no hay garantía de obtener un movimiento suave debido al peso de la pieza. Por otro lado, si se usara este diseño con rodamiento se corre el peligro de desencajar debido a que estos no estarían bien sujetos a la pata donde residen. Además, se tuvo que usar un diámetro de eje pequeño para poder acercar lo máximo posible el eje del rodillo hacia el centro de la pieza para no tener un gran cambio de diámetro desde el eje hasta el punto en el que haría contacto con la cerámica.

Es por ello por lo que se decidió crear una segunda iteración del proyecto para arreglar algunos inconvenientes y tratar de crear un prototipo para comprobar que el diseño es viable y útil para la tarea requerida.

3.1.3.2 Segunda iteración

En esta segunda iteración se realizaron algunos cambios para mejorar el prototipo.

Respecto a las patas de la base, se diseñó el alojamiento del rodamiento para usar un eje de 10 mm de diámetro, además de aumentar las dimensiones a 50 x 50 x 55 mm. Estos cambios se realizaron para asegurar una mayor estabilidad del sistema.

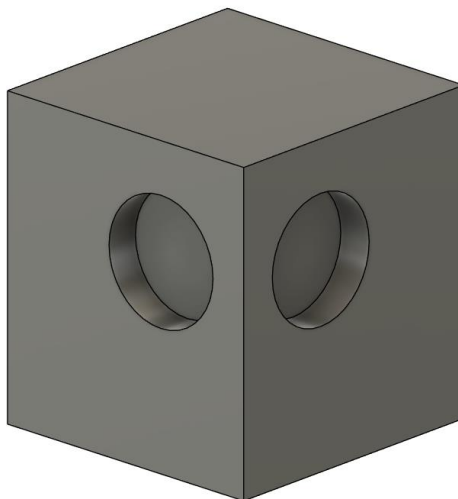


Ilustración 20. Segunda iteración de los apoyos.

En la parte superior de la base también se realizaron algunos cambios. Como se puede observar, las paredes que delimitan las secciones interiores son más altas y se alarga más hacia el interior, permitiendo añadir el alojamiento de un rodamiento para que el movimiento con el motor sea suave y preciso.

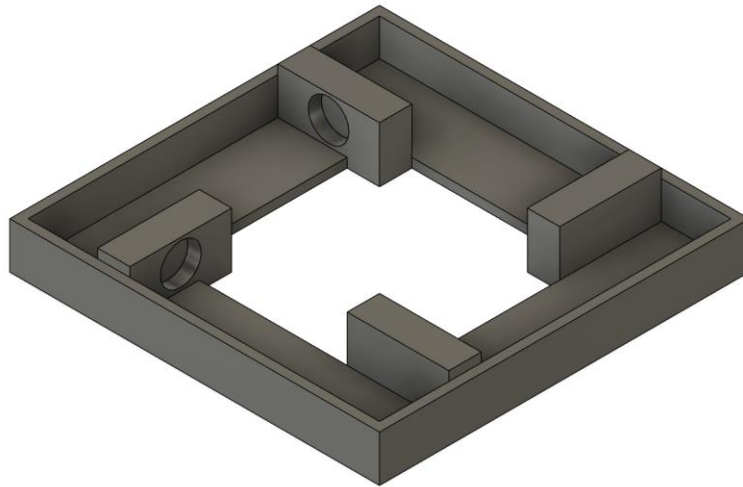


Ilustración 21. Segunda iteración de la parte superior de la base.

A continuación, se muestra cómo queda el prototipo completado.

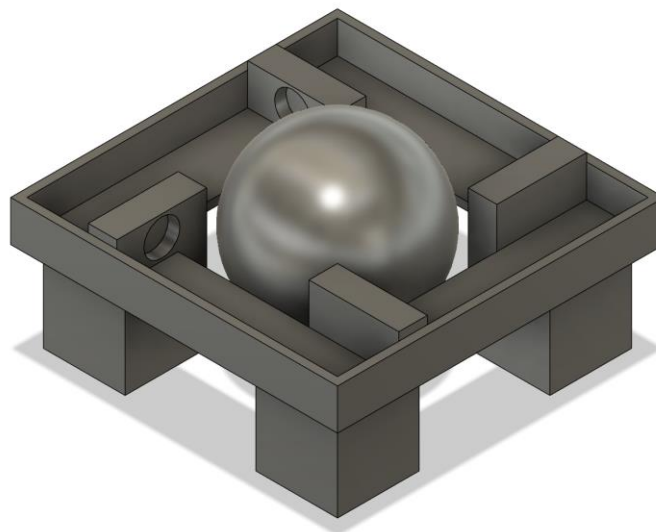


Ilustración 22. Segundo ejemplo modelado con las piezas anteriores.

Tras completar el prototipo y discernir sobre este, se observa que presenta varios inconvenientes, los cuales son:

- Rigidizado: Debido a la configuración, lo único que mantendría las patas del soporte en su sitio correcto sería los rodillos que se colocan entre ellos. Además, esto plantea un nuevo problema, pues la parte superior no tiene ningún anclaje que asegure que no se produzca ningún tipo de deslizamiento con las patas y, además, se mantenga centrado en la posición de operación, ya que al tener rodillos motrices se podría levantar y separar de las patas.

- Rodillos motrices: En este sistema, los rodillos motrices se encuentran en la zona superior. Esto deriva en problemas de contacto, ya que al tener un rodamiento en un alojamiento fijo puede que no haga la suficiente presión para transferir el movimiento del rodillo a la cerámica. Además, se debería añadir un sistema que permita variar la presión que ejerce (por ejemplo, unos muelles en el alojamiento que permitan mover hacia fuera los rodillos) para poder insertar correctamente y sin daños la cerámica.

En un primer momento se creó un diseño similar en el que se añadían agujeros en la parte superior como en los apoyos de la base para poder atornillarse y evitar movimientos indeseados, pero, tras analizar mejor la configuración, se decidió descartar la idea de usar rodillos superiores debido a los inconvenientes que presentaban. Es debido a ello por lo que los rodillos inferiores pasaron a ser motrices y se decidió eliminar el seccionamiento de la parte superior de la base.

3.1.3.3 Tercera iteración

Tras decidir descartar los rodillos superiores, se decidió crear un prototipo físico para comprobar de forma manual que el sistema era viable usando la configuración determinada.

Los apoyos de las bases sufrieron modificaciones mínimas, solo unos agujeros para poder atornillar la parte superior y rigidizar el sistema. Estos agujeros se realizaron de 12 mm de diámetro pues se disponía de los tornillos para realizar el prototipo de forma rápida.

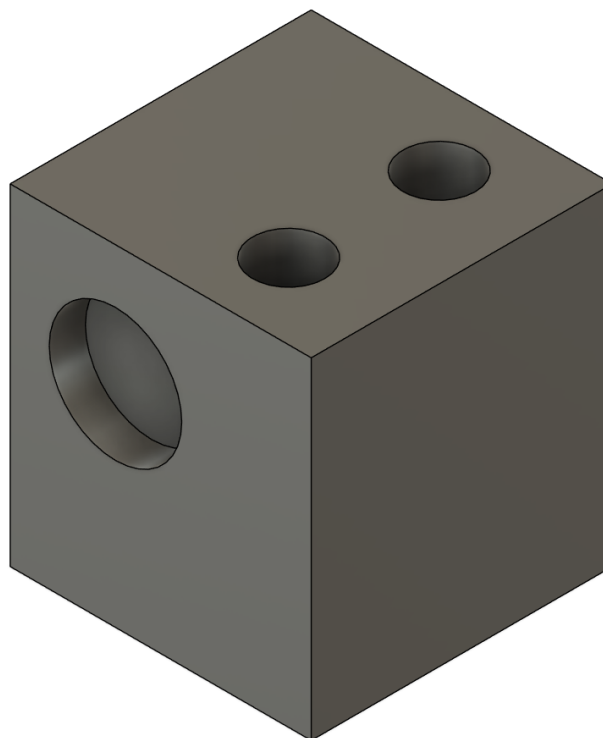


Ilustración 23. Tercera iteración de los apoyos.

Por otro lado, la parte superior de la base sufrió algunos cambios. Se cambió las dimensiones del cuadrado a 200 x 200 mm y se quitó el área de un cuadrado de 115 x 115 mm del interior para ubicar

la cerámica. Se eliminaron las secciones que delimitaban el interior además de añadir 2 agujeros en cada esquina en sentido horario para poder atornillar y asegurar esta parte a las patas de apoyo, evitando así movimientos no deseados.

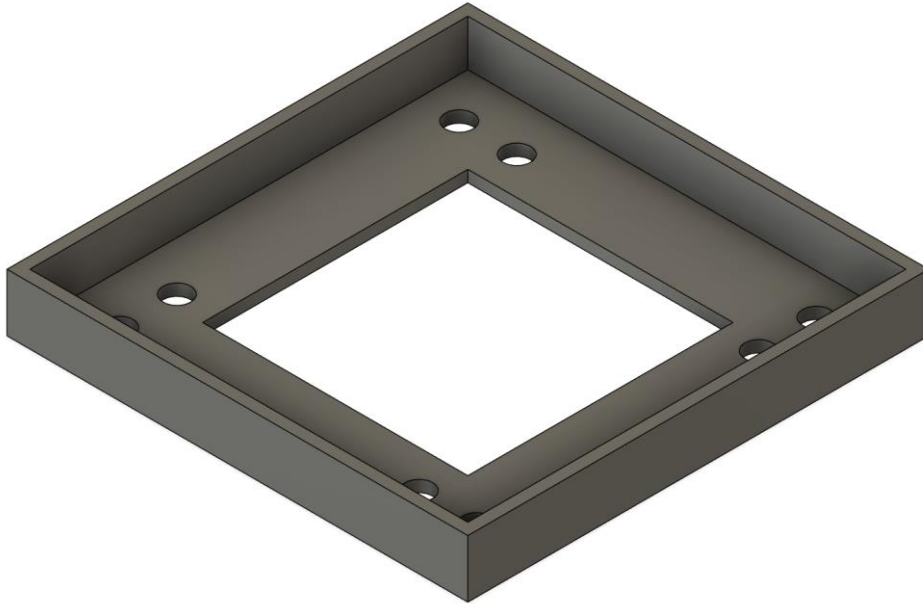


Ilustración 24. Tercera iteración de la parte superior de la base.

Tras analizar esta tercera versión, se decidió que se iba realizar un prototipo para comprobar el correcto funcionamiento de este, dando por bueno el diseño del sistema y añadiendo mejoras al sistema sin cambiar el diseño base de este.

3.1.4 Prototipo

3.1.4.1 Primer prototipo

Una vez definido el prototipo que se va a crear, se dispuso a importar los archivos a STL para poder ser impresos correctamente.

Primero, se imprimió la parte superior del sistema sin problema. Debido a las tolerancias que se obtienen al realizar una impresión en 3D, fue necesario perforar con una broca los agujeros ya creados para obtener el diámetro deseado de forma ajustada.

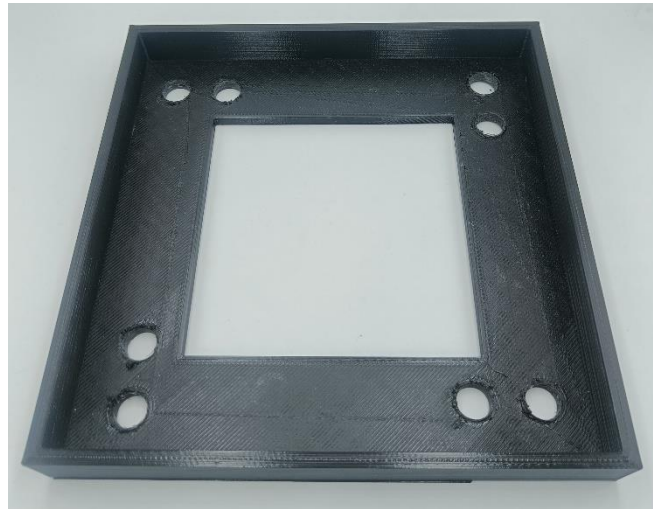


Ilustración 25. Parte superior del primer prototipo impreso.

Por otro lado, los apoyos de la base fueron creados y fue necesario realizar algunas modificaciones. En primer lugar, hubo que llevar los agujeros ciegos a tolerancias usando una broca de 12 mm. Una vez realizado, fue necesario usar una serie de machos para crear la rosca de 12 mm para asegurar la parte superior del soporte con los apoyos y evitar movimientos. Además, se realizó un agujero pasante en el alojamiento del rodamiento de 10,5 mm para asegurar que pueda pasar el eje de los rodillos y que pueda girar libre, además de añadir un agujero ciego en el alojamiento del rodamiento del lado colindante para poder apoyar el eje en el otro apoyo y así que quede sostenido.



Ilustración 26. Primer prototipo de los apoyos.

Una vez preparado el soporte, se procedió a crear los rodillos. Estos están formados por 3 partes: eje, casquillos y silicona.

El eje se trata de un eje tubular de diámetro 10 mm y de 186 mm de longitud. Este fue seleccionado para realizar una prueba de perforación, ya que en un principio sería capaz de resistir gran cantidad de fuerza ejercida en él.

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico mediante visión y robótica colaborativa.

Los casquillos se tratan de piezas diseñadas para imprimir en 3D cuya función es la de servir de transición entre el diámetro del eje hasta el diámetro que fue calculado como necesario para tener un contacto seguro con la cerámica, como se muestra a continuación.

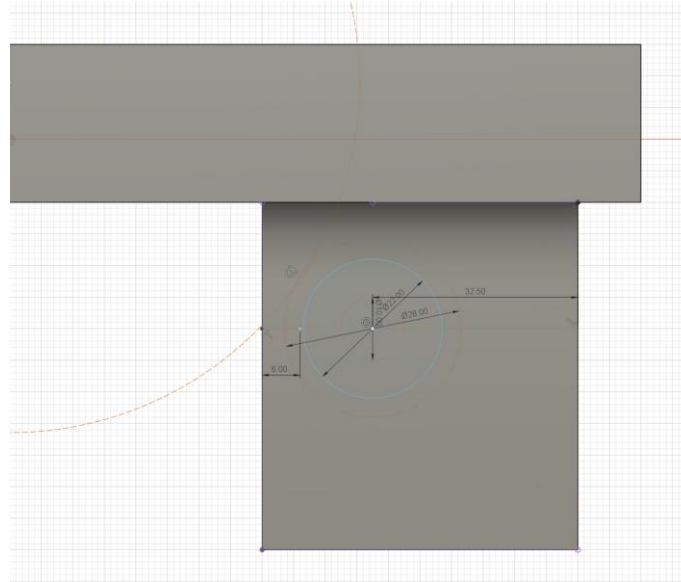


Ilustración 27. Cálculos realizados para obtener el punto óptimo donde posicionar el eje.

Estas piezas cuentan con un diámetro interior de 10 mm (para obtener un buen ajuste) y un diámetro exterior de 20,8 mm.



Ilustración 28. Casquillos: izquierda modelado y derecho impreso.

Por último, el componente restante es la silicona, que se trata de un tubo de silicona de grado industrial de 16 mm de diámetro interior y 21 mm de diámetro exterior. Esta se posiciona entre dos casquillos ya

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico mediante visión y robótica colaborativa.

mencionados anteriormente para que no se pueda mover, además de introducir una pequeña parte de estos casquillos en la silicona para evitar deslizamientos y asegurar que se mantenga la posición correcta. La silicona no se encuentra en el centro del eje, si no que el centro de esta se encuentra a 5 cm de uno de sus lados. Esto se debe a que se quiere que el eje sobresalga de los apoyos para poder manipular los rodillos de forma manual y comprobar la viabilidad del diseño.



Ilustración 29. Primer prototipo de rodillo montado.

Una vez se tienen los 3 componentes listos, se procedió a realizar el montaje. Debido a cómo está dispuesto el diseño, primero se ha de insertar el eje de los rodillos en los agujeros pasantes de los apoyos y una vez realizado se atornilla la parte superior de la base a los apoyos como se muestra a continuación.

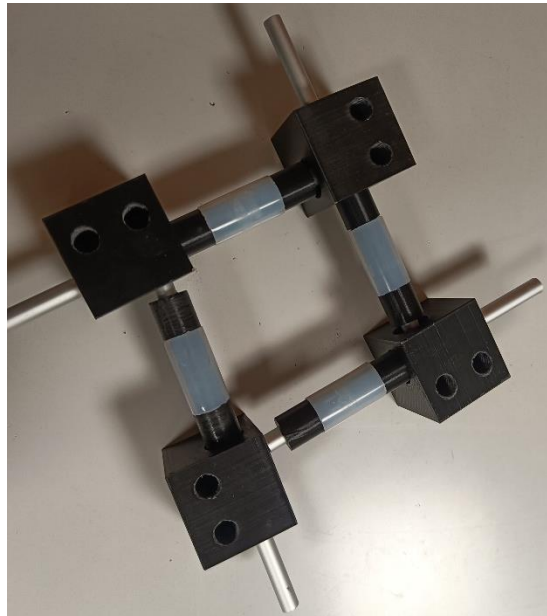


Ilustración 30. Colocación y ajuste de los rodillos previo a montaje.

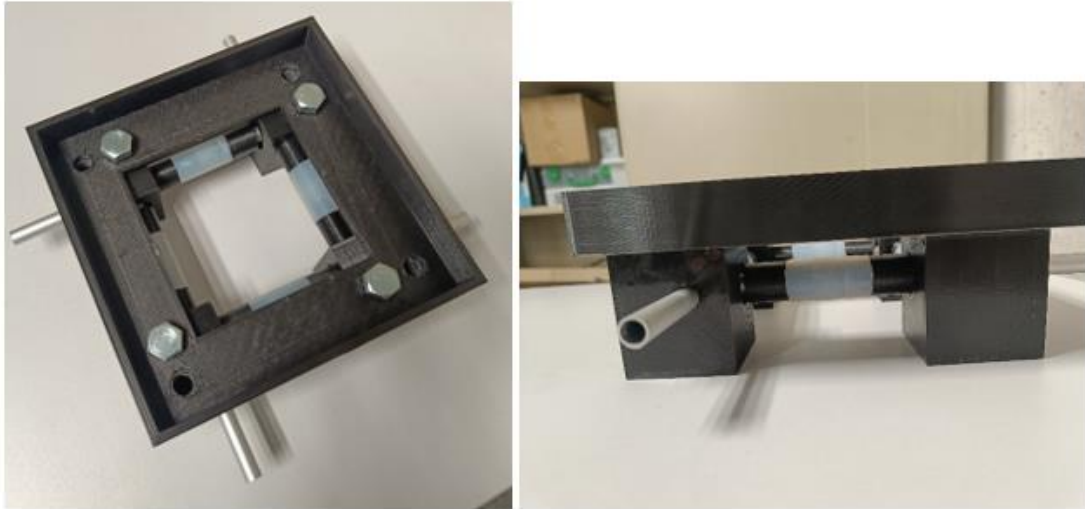


Ilustración 31. Primer prototipo montado con distintas vistas.

Tras este montaje, se procedió a realizar las pruebas para comprobar cómo se comportaba el prototipo ante fuerzas y sobre todo cómo transfería movimiento a la cerámica, comprobando sobre todo la posibilidad de encontrar puntos muertos donde la pieza no deslizará y/o perdiera el contacto con alguno de los rodillos, provocando que quedara estancada en una posición.

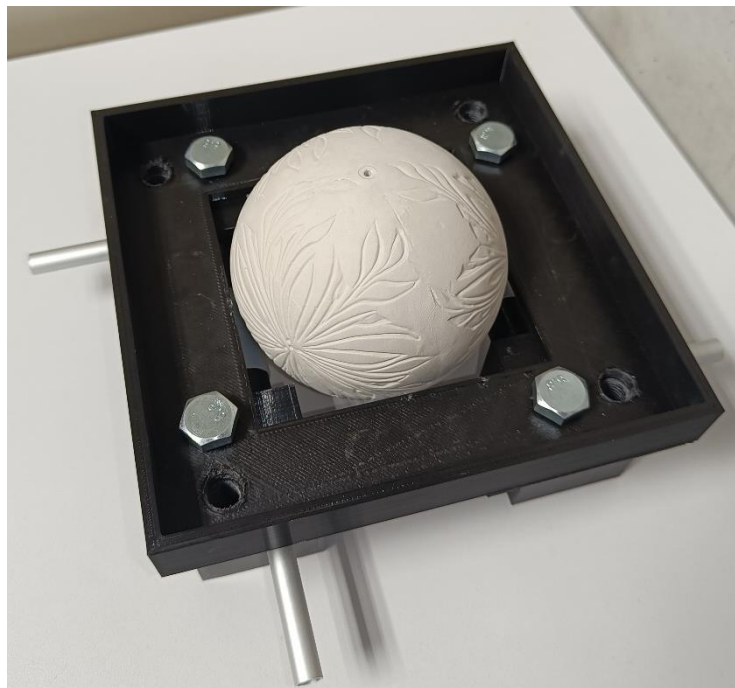


Ilustración 32. Prototipo preparado para realizar diversas pruebas.

Tras diversas pruebas (movimiento manual de los rodillos, diversas posiciones de arranque de la cerámica, uso de varias cerámicas, etc), se detectaron algunos problemas que se habían de resolver:

- Deslizamiento: Debido al residuo que deja el material cerámico de la pieza, esta desliza en los rodillos perdiendo tracción, ya que el polvo se deposita en la silicona. Esto produce que se pierda la adherencia solo por el hecho de tener contacto con la cerámica.

- Puntos muertos: Se detectó que en determinadas posiciones la pieza cerámica perdía el contacto con alguno de los rodillos. Esto se produce cuando la sección cortada donde tiene el agujero principal se encuentra cerca o en la posición donde se encuentra un rodillo.
- Contacto no deseado: Debido a la posición de la cerámica en la base, en determinados momentos esta entra en contacto con la esquina interior de un apoyo, pudiendo marcar la pieza. Además, debido a que el apoyo en los rodillos se encuentra más cerca de la línea del ecuador de la pieza, esta está demasiado baja y al realizar fuerza sobre ella en ocasiones colisiona con las paredes de la parte superior de la base, marcando la pieza.
- Geometría variable: Al ser obtenidas a través de una batería de moldes, las cerámicas no son totalmente iguales ni esféricas, lo que provoca algunos puntos en que no se obtenga tracción.
- Casquillos: Los casquillos que se utilizaron se crearon con una tolerancia muy justa para no permitir movimiento. Esto puede ser contraproducente pues para introducir los casquillos en el eje se tuvo que utilizar fuerza y lubricante que se evapora rápido. Esto produjo que los ejes sufrieran pequeñas deformaciones (algunas solo en la forma y otras se arquearon) que creaban problemas debido que el eje rota describiendo un arco.
- Tornillos excesivos: Debido al uso de tornillos de 12 mm, se vio excesivo el diámetro puesto que

Tras obtener estos resultados, se propuso crear el prototipo final que se usará en el proyecto.

3.1.4.2 Prototipo final

El prototipo final se comenzó a partir de las bases asentadas que se obtuvieron durante todo este proceso de investigación y creación.

En primer lugar, se modificaron los apoyos de acuerdo con lo observado anteriormente. Los cambios que se realizaron son:

- Chaflanado: Debido a la posición de los rodillos, la pieza tenía un punto de contacto cercano a su ecuador. Es por ello por lo que se decidió crear un chaflán, en la esquina que del apoyo que se encuentra en la parte interior de la base creando espacio suficiente para alojar sin problemas la cerámica. A su modo, esto ayuda a aligerar el peso y a no consumir tanto material, permitiendo una impresión más rápida de las piezas.
- Diámetro agujeros: Se decidió revisar y reducir el tamaño de los agujeros donde roscan los tornillos que mantienen la parte superior de la base y el apoyo juntos. Se optó por usar tornillos de 6 mm de diámetro, además de añadir un tercer agujero para asegurar que la estructura estuviera correctamente rigidizada y pudiera resistir movimientos horizontales en caso de producirse.

- Rodamiento y agujero pasante: Se decidió cambiar el tamaño del alojamiento de los rodamientos pues se cambió el diámetro del eje de los rodillos (más tarde se explicarán los motivos). Esto permitió además acercar más el rodillo al extremo del apoyo, pues al estar ambos rodamientos en caras contiguas si estos eran demasiado grandes o profundos podían solaparse y crear conflicto pues uno evitaría que se alojase otro. Debido a esto, también se redujo el diámetro del agujero pasante de uno de los rodamientos a unos 6,5 mm para asegurar que no se produjeran rozamientos en el interior de la pieza.
- Posicionamiento para impresión 3D: Debido a que la calidad de impresión de los bordes y partes circulares cambian en función de la orientación que tienen respecto la orientación de la impresora 3D, fue necesario rotar la pieza 180° de forma que el alojamiento del rodamiento que tiene en su interior el agujero pasante quede expuesto en el plano x-z, permitiendo una mayor resolución y acabado en la pieza obtenida.

Tras implementar estos cambios, el diseño del apoyo quedó de la siguiente forma:

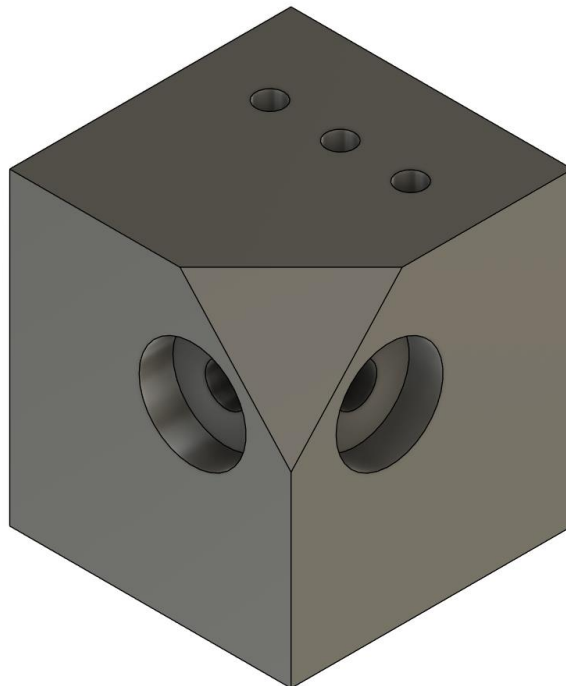


Ilustración 33. Modelado de apoyo final de la base.

En cuanto al diseño de la parte superior de la base, este cambió un poco. Como ya se ha mencionado con anterioridad, se usaron 3 tornillos de 6 mm como anclaje. Para ello se diseñaron agujeros de 5 mm para posteriormente roscarlos. Además, debido al contacto intermitente de la pieza cerámica con los bordes del soporte, se decidió crear un agujero de 155 mm de diámetro de forma que se eliminaran las aristas cercanas a la cerámica para evitar dicho contacto no deseado, quedando así el diseño:

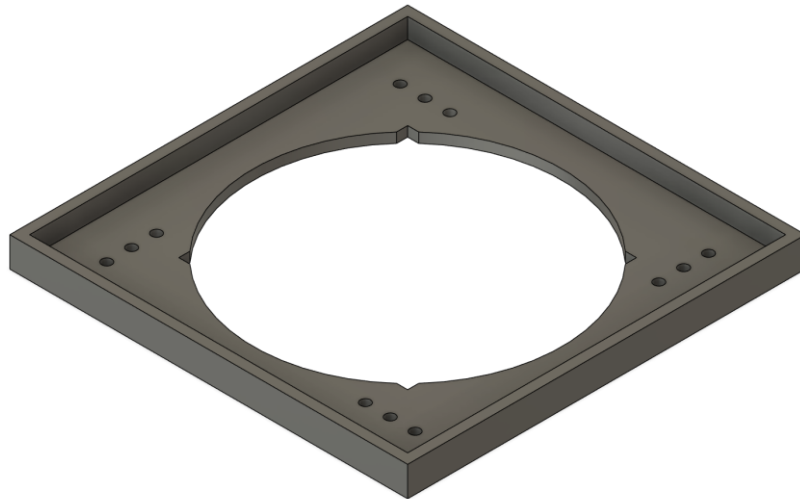


Ilustración 34. Modelado de la parte superior de la base final.

Una vez obtenidas las piezas impresas, se procedió a mecanizar las partes necesarias.

En primer lugar, se llevaron a tolerancias los 3 agujeros ciegos para poder crear la rosca con machos. Esto fue necesario pues al estar la pieza con el agujero pasante en el plano x-z, estos agujeros no tenían el acabado ni las dimensiones necesarias para poder hacer el roscado correctamente. Una vez llevados los agujeros a tolerancias, se procedió a realizar el roscado a través de un macho de 3 etapas como el que se muestra a continuación:



Ilustración 35. Machos de rosca M6x1.00. Fuente: ManoMano.es

Tras esto, se aseguró que los alojamientos de los rodamientos tuvieran las dimensiones necesarias pues las cotas usadas en el diseño diferían con las dimensiones obtenidas en la pieza. Por último, se aseguró que el chaflanado realizado mantuviera las dimensiones especificadas.

Una vez se dio el visto bueno a las piezas mecanizadas, se procedió al montaje de los rodamientos aptos para altas velocidades. en estas. Para ello simplemente se encajaron los rodamientos en el alojamiento ya llevado a tolerancias.

Estos rodamientos se tratan de "626-2Z". Se tratan de rodamientos de acero de ranura profunda apantallados con un diámetro interior de 6 mm, uno exterior de 19 mm y un grosor de 6 mm. Con una carga nominal dinámica de 2.336 kN y una estática de 558N, cuentan con especificaciones más que suficientes para la operación.



Ilustración 36. Ejemplo de rodamiento 626-2Z. Fuente: Rs-online.

Una vez completados los apoyos, se pasó al mecanizado de la parte superior de la base.

Debido al dimensionamiento previo de los agujeros, sólo fue necesario realizar el roscado para 6 mm usando los mismos machos usados para los apoyos de la base.



Ilustración 37. Muestra de la base atornillada correctamente.

Como se ha comentado en la parte de los apoyos, se realizó un cambio de diámetro tanto en los ejes de los rodillos como en la parte que tiene contacto con la pieza.

En primer lugar, tras realizar pruebas se concluyó que un diámetro de 6 mm de eje sería suficiente para aguantar la presión que se ejerce al realizar las perforaciones en la pieza cerámica, pues esta fuerza se divide entre cuatro rodillos y, como se explicará a continuación, es necesario realizar poca fuerza para no dañar la pieza y obtener buenos resultados. Debido a esto se redujo el diámetro del agujero para el eje a 6,10 mm para poder mover el casquillo a la posición deseada sin problema.

Una vez se obtuvo esta conclusión, se planteó una nueva forma de pasar del diámetro del eje al diámetro necesario para el contacto con la pieza. Para ello se plantearon 2 diseños de casquillos para

evitar que los ejes se deformaran. Para ello, se realizó un sistema el cual permite cerrar los casquillos como si se trataran de una abrazadera. Con esta idea, se propusieron 2 diseños que fueron probados.

El primero se trata de un casquillo de 22 mm de largo con una sección rectangular extraída, junto con un poco de material en el lado contrario para facilitar el movimiento y evitar que se rompa por ser demasiado rígido. Para ello se realizó una geometría más compleja que la usada anteriormente.



Ilustración 38. Diseño de casquillo con sección extraída plana.

Como se observa, este diseño cuenta con la sección extraída además de estar formado por dos diámetros. El exterior de 28 mm sirve para encapsular un tornillo de métrica 4 junto a su rosca por el que se cerrará el casquillo para fijarlo al eje a través de un agujero pasante y el alojamiento para la cabeza del tornillo y de la tuerca, además de servir como tope para la sección de tubo de silicona que se coloca entre dos de estos casquillos. Debido a que el tubo de silicona usado tiene un diámetro interior de 22 mm, se realizó el diámetro menor del casquillo de 22,2 mm para asegurar un encaje correcto.

Al probar esta iteración se comprobó que la superficie de contacto del tubo exterior no sobresalía lo suficiente respecto el diámetro mayor del casquillo, provocando contacto con este y perdiendo tracción. Además, al ser la sección extraída completamente recta, al cerrar el casquillo en el eje se creaba un saliente que deformaba la silicona creando problemas pues el contacto con la pieza cerámica no era suave.

Debido a esto se realizó una segunda versión en la que se implementaron diversos cambios. En primer lugar, se modificó la forma de la sección extraída a una con un ángulo de inclinación de -3 grados para evitar el relieve mencionado anteriormente.

En segundo lugar, se redujo el diámetro del diámetro mayor de la pieza a 24 mm para así evitar que la pieza contacte con el casquillo. A pesar de tener el diámetro reducido, esta parte sirve perfectamente para evitar que la sección de silicona usada no se mueva de su posición, además de poder usar el mismo tornillo de métrica 4 en su agujero pasante.

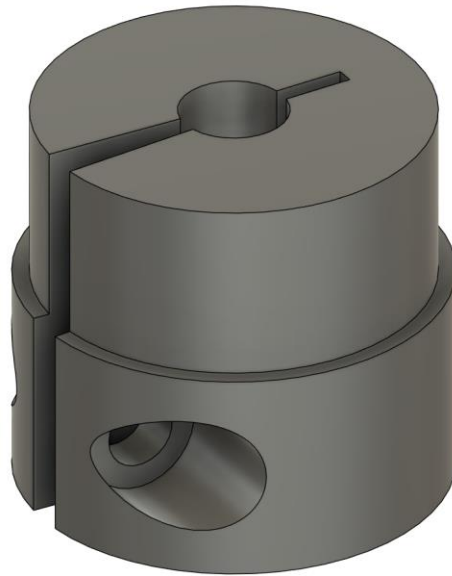


Ilustración 39. Modelado de casquillo con sección extraída en ángulo.

Una vez se imprimió el diseño y se comprobó su funcionalidad, se montaron los rodillos como se muestra a continuación, usando secciones de unos 26 mm de largo. Se usaron secciones más cortas pues se determinó que el punto de apoyo es pequeño y se necesita menos material.



Ilustración 40. Visualización de ejes colocados en su posición correspondiente.

Posteriormente, había que decidir cómo se moverían estos rodillos. En un primer momento, se planteó la posibilidad de usar motores pequeños pues el par necesario para mover la cerámica es pequeño al tener un peso reducido. Como primera instancia, se planteó el uso de motores con encoder pues el robot UR5 tiene una función que permite controlar la posición de este tipo de motores. El problema reside en que sólo se podría controlar la posición de dos de ellos, teniendo que usar los motores en pares además de tener que buscar motores que sean compatibles.

Es por ello por lo que se decidió optar por motores stepper, debido a que su movimiento está fijado en pasos y es posible modular de forma precisa el movimiento deseado a través de la programación.

Se eligió usar 4 motores modelo "Wantai 42BYGHW811". Esto se debe a que en el laboratorio se encontraba una máquina de impresión 3D que estaba en desuso y de la que se podían reciclar partes. Es por ello por lo que poseen unas características más que suficientes para el prototipo. A continuación, se muestra una imagen del motor:



Ilustración 41. Motor Wantai 42BYGHW811i usado. Fuente: Openimpulse.com

Este modelo cuenta con un eje de 5 mm y un giro de 1,8 grados por paso con un consumo nominal de 2,5 amperios, siendo estas cualidades excesivas, por lo que se usarán controladores para facilitar el uso de los motores. El funcionamiento de estos controladores se explicará en el apartado de comunicaciones.

Para poder situar los motores en la posición adecuada fue necesario imprimir unas piezas las cuales se encajan en la parte inferior de los apoyos y sirven para nivelar el eje del rodillo y el eje de los motores. La pieza es la siguiente:

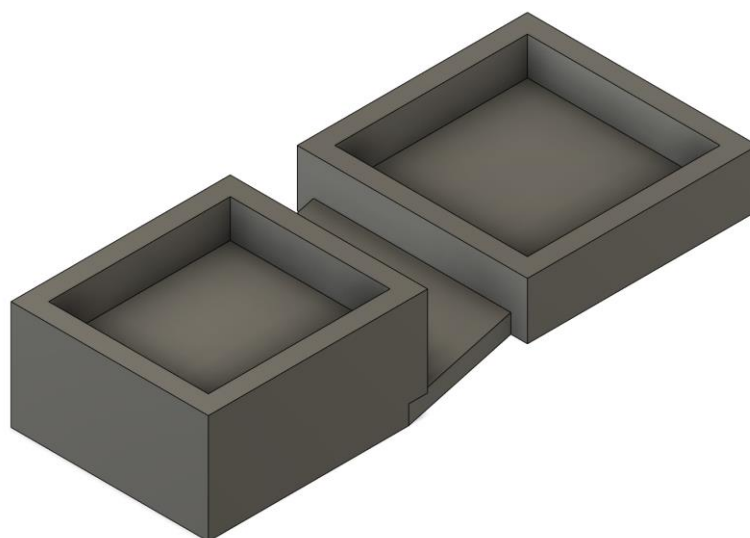


Ilustración 42. Diseño del soporte a alojar los motores Wantai de la base.

Esta cuenta con 2 partes cuyos centros están separados 93 mm en un ángulo de 83,5 grados, con muro de 1 cm de sustento para evitar movimientos horizontales. El mayor se trata de la parte que se encaja en los apoyos de la base, siendo de 51 x 51 mm para que encaje correctamente y con una altura de 5 mm respecto el suelo para igualar las alturas.

La segunda parte está formada por otro cuadrado de 48 x 48 mm con una elevación de 19 mm para ajustar correctamente la altura del eje del motor al eje del rodillo, puesto que este sobresale unos 3 cm respecto la pared del apoyo. Para realizar la conexión entre los ejes es necesario utilizar un acoplador que permita el movimiento libre y absorba las vibraciones y desalineaciones que se puedan producir. Para este caso, se tratan de acopladores de 5 a 6 mm permitiendo conectar ambos ejes.



Ilustración 43. Acopladores de distinto diámetro usados. Fuente: Amazon.es

El sistema quedaría de la siguiente manera:

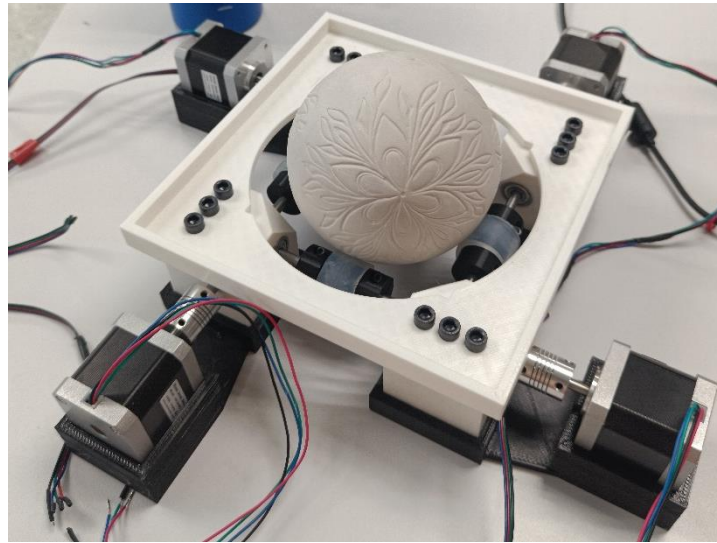


Ilustración 44. Montaje final de la base.

Tras unas últimas pruebas, se comprobó que la pieza continuaba patinando a pesar de haber bajado el punto de contacto entre los rodillos y la pieza para mejorar el rozamiento a través de la gravedad, por lo que se decidió utilizar otra aproximación.

Se decidió por añadir a la silicona algún material que permitiera el asentamiento correcto de la pieza, a la vez de permitir un deslizamiento útil (para que la pieza pueda rodar sin problemas sin que se levante cuando los rodillos contiguos giren) y que tenga el agarre necesario para poder mover la cerámica. Estas características son complejas de obtener pues son opuestas, pero se decidió realizar pruebas con 3 tipos de materiales.

Se decidió realizar una serie de pruebas con las cerámicas para cualificar el comportamiento de cada material en diferentes. A continuación, se muestra una tabla con los resultados obtenidos.

Tabla 1. Comparación entre los materiales disponibles.

	Espuma filtro de aire	Espuma aislante tuberías	Papel de burbujas espumado
Asentamiento	Medio	Bueno	Medio
Deslizamiento	Bueno	Bueno	Bueno
Tracción	Bueno	Malo	Malo
Impacto residuo	Pequeño	Alto	Mediano
Implementación	Difícil	Fácil	Difícil

A continuación, se muestra el ejemplo de la prueba de la espuma aislante de tuberías:

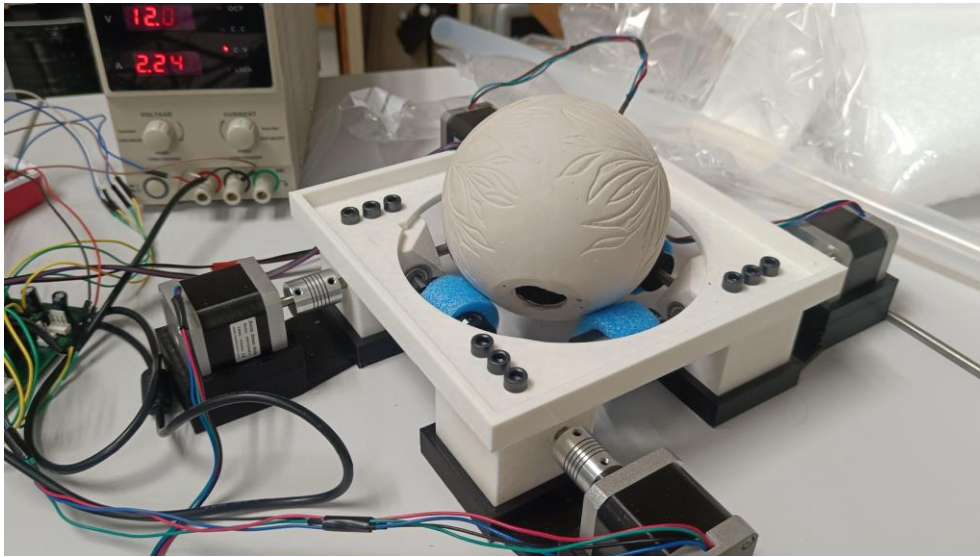


Ilustración 45. Montaje probando el comportamiento de la espuma aislante.

Finalmente, se decidió usar el filtro de aire de espuma pues fue el que mejores resultados obtuvo. Debido a su porosidad el residuo no se acumulaba y no impedía el movimiento de la pieza, además de la textura de la espuma permite un buen asentamiento de la cerámica a la vez de tener una tracción suficiente para mover la pieza sin problemas y un deslizamiento que genera que los rodillos perpendiculares no influyan en el movimiento. Para poder instalar la espuma en el rodillo fue necesario usar un pegamento instantáneo (en este caso Loctite) de forma que sólo las fibras exteriores de la espuma entraran en contacto con la silicona y el pegamento, pues si el pegamento penetraba en esta (la espuma es muy porosa) se creaban cúmulos y pegotes de pegamento que se endurecían y no permitían que el movimiento de la pieza cerámica fuera suave, pues esta se elevaba cuando tenía contacto con las partes rígidas.



Ilustración 46. Solución alcanzada con el filtro de aire de espuma.

Tras todo este desarrollo, finalmente se obtuvo la base en la cual se sustenta la pieza cerámica, se realizan los movimientos y permite el avance del proyecto.

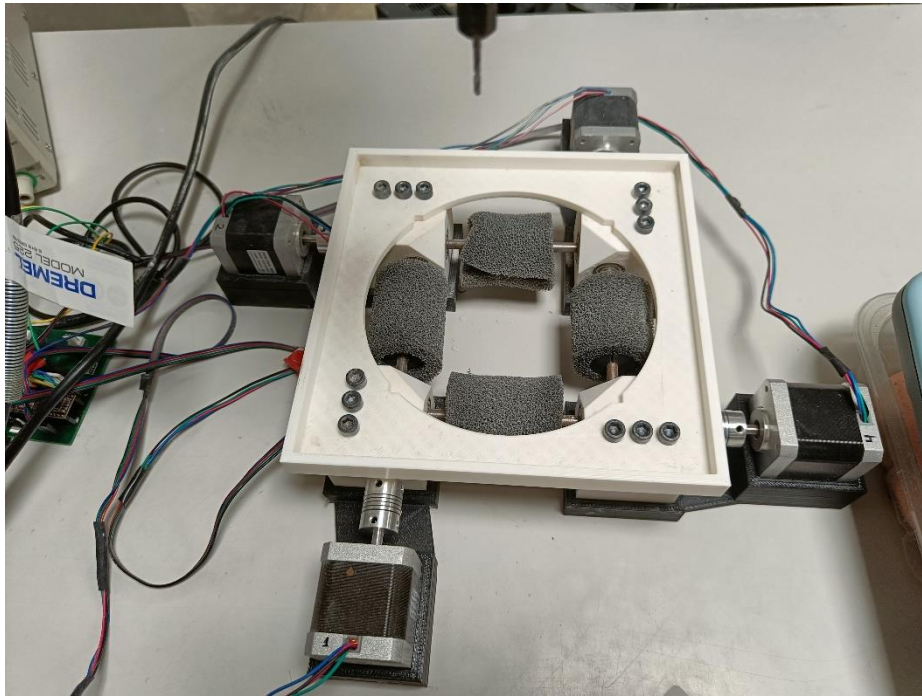


Ilustración 47. Montaje de la base definitivo. En la espuma se puede apreciar el residuo de la cerámica.

3.2 Diseño soporte Dremel

Por último, como sistema de perforación se decidió usar una Dremel 3000. Este sistema podría haber sido simplemente un motor con un adaptador para colocar una broca para perforar, pero se optó por una Dremel pues la configurabilidad era más sencilla al tener ya modos y velocidades estables a pesar de los inconvenientes que más adelante se exploran.

Para poder fijar la herramienta al robot UR 5, se realizaron varios diseños hasta obtener la versión final de este.

3.3.1 Primera versión

La herramienta seleccionada para perforar se trata de una Dremel 3000. Puesto que la Dremel tiene un peso y volumen considerable, se optó por usar el extensor que proporciona la herramienta y colocar la Dremel en una zona elevada permitiendo completa movilidad al robot mientras perfora.



Ilustración 48. Herramienta Dremel usada. Fuente: Amazon.es

Esto planteó otro reto pues, debido a la forma del eje extensible de la Dremel, es necesario encapsularlo para evitar movimientos indeseados a la hora de perforar o debido a las vibraciones producidas al encender la herramienta.



Ilustración 49. Eje extensor de Dremel. Fuente: Dremel.com.

Esta primera versión se obtuvo primero midiendo la parte principal del eje extensible creando una geometría para alojar esta pieza.

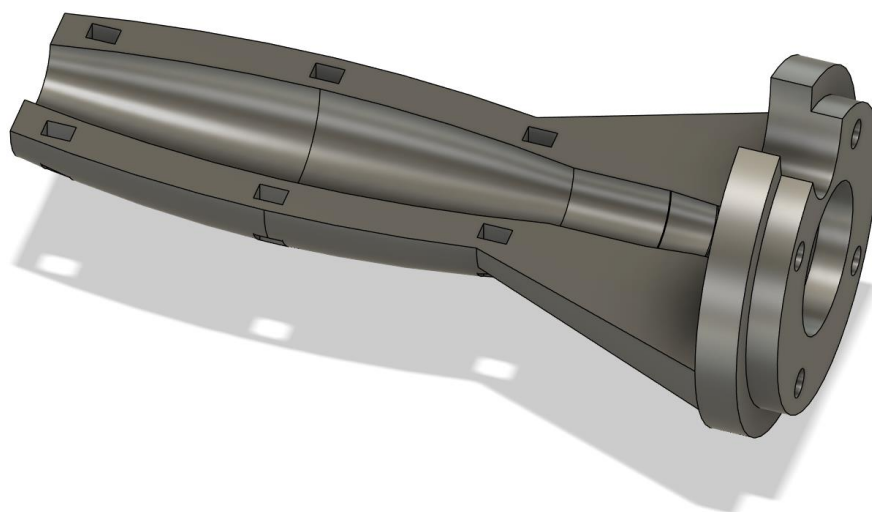


Ilustración 50. Primera versión del soporte para el eje extensible.

La pieza cuenta con 3 partes diferenciadas. La primera se trata del alojamiento principal, que cuenta con una longitud de 108 mm y un diámetro máximo de 26 mm en la parte central y de 16 mm en el extremo libre donde se encuentra la boquilla de la broca. Esta pieza se formó a través de la revolución de una sección de 108 mm y 299 mm de diámetro para obtener el alojamiento con una pared de un 10 mm de grosor para asegurar la rigidez.

Cabe destacar que también en esta parte se encuentran tres agujeros pasantes de 6 x 4 en cada lado de la pieza que sirven para encajar una pieza superior que fija el eje extensible al soporte.

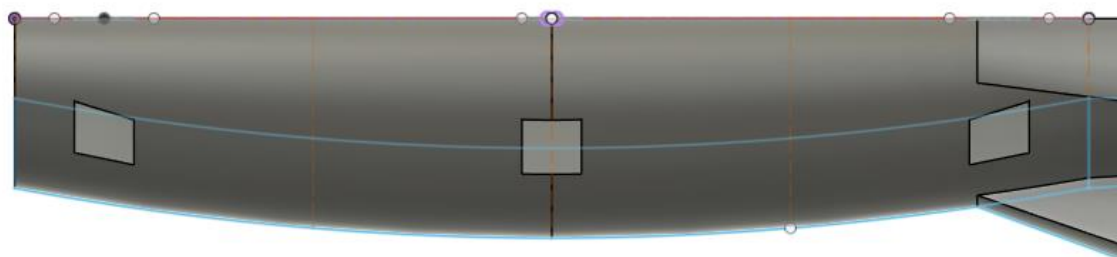


Ilustración 51. Forma de obtener la geometría observada en el soporte.

La segunda parte consta de la parte del soporte que se conectará a la muñeca del robot UR 5. Este cuenta con 2 cilindros (unos de 8,5 mm y otro de 6,5 mm). Este primero cuenta con un diámetro de 75 mm con una sección extraída correspondiente a media elipse con 20 mm de eje menor y 37,3 de eje mayor. Esta parte se realizó como guía para el cable del eje extensor, para así evitar que se mueva e interfiera en los movimientos y en las perforaciones.

El segundo cilindro cuenta con un diámetro de 63 mm en el cual se encuentran agujeros de 6,5 mm situados a una distancia de 25 mm a 45 grados de la línea de simetría de la pieza. Esta es la parte la cual tiene contacto con la muñeca y donde se alojan los tornillos y arandelas que fijan el soporte.

Por último, tenemos la tercera parte, que se trata de la conexión ente las dos partes anteriores. Esta cuenta con un diámetro interior de 16 mm que se alarga 20 mm hasta que se reduce a 14 cm para acomodar la parte reforzada el eje extensible.



Ilustración 52. Zona del eje extensible reforzada.

Demás, esta parte cuenta con 3 nervios que sirven como sustento de la pieza, pudiendo así aguantar el peso en caso de movimientos en voladizo, además de estabilizar la pieza para evitar movimientos por las vibraciones al perforar y al estar la Dremel encendida.

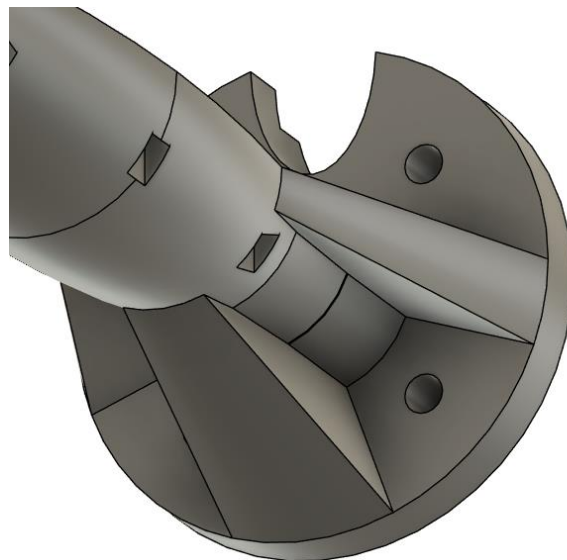


Ilustración 53. Refuerzos colocados rigidizar más el soporte.

Por otro lado, también fue necesario diseñar la parte superior que permite sujetar la parte de la punta del eje extensible. Esta está diseñada para que encajara gracias a unos ganchos situados en la parte final de las patillas extruidas.

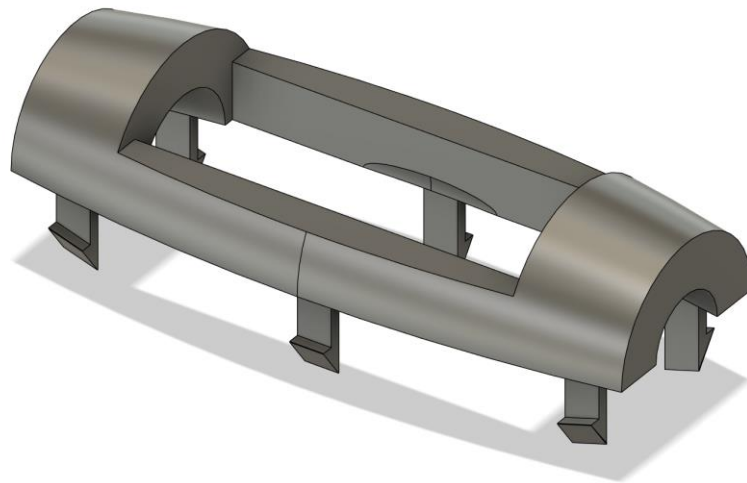


Ilustración 54. Parte complementaria al soporte anterior que retiene al eje.

Como se puede observar, el diseño es similar a la primera parte de la pieza anterior. La diferencia principal reside en la apertura de la zona central. Esta se trata de un rectángulo de 25,5 x 70 mm que forma el agujero pasante. Además, se retiró también el material de la mitad superior de la pieza para permitir manipular mejor la cabeza del eje extensible. A continuación, se muestra una imagen con el prototipo creado:



Ilustración 55. Primera versión de las piezas impresas.

Como se puede observar, la parte superior del soporte fue impresa con deformaciones en las patas extruidas, por lo que se tuvo que modificar el diseño pues la precisión de la impresora 3D no permitía tal nivel de detalle.

3.3.2 Segunda versión

Al poner en funcionamiento el prototipo anterior se observó que este era demasiado corto para poder albergar correctamente la parte reforzada del eje extensible de la herramienta. Esto generaba una tensión en la dirección del refuerzo metálico que desplazaba la punta de la herramienta de forma que

no quedaba totalmente centrada, además de generar tensión en el acoplamiento con la muñeca del robot.

Es por ello por lo que se decidió modificar este soporte haciendo que la tercera parte de este (la parte que conecta el alojamiento principal y el punto donde el soporte se atornilla al robot) sea más larga. Esta se alargó hasta los 88 mm (anteriormente era de 10 mm) para asegurar que la zona reforzada del eje flexible tenga espacio suficiente para asentarse y no crear tensión. También se modificó el cambio de diámetro en la parte donde está el refuerzo del eje extensible para que sea más grande y así poder alojarlo por completo.

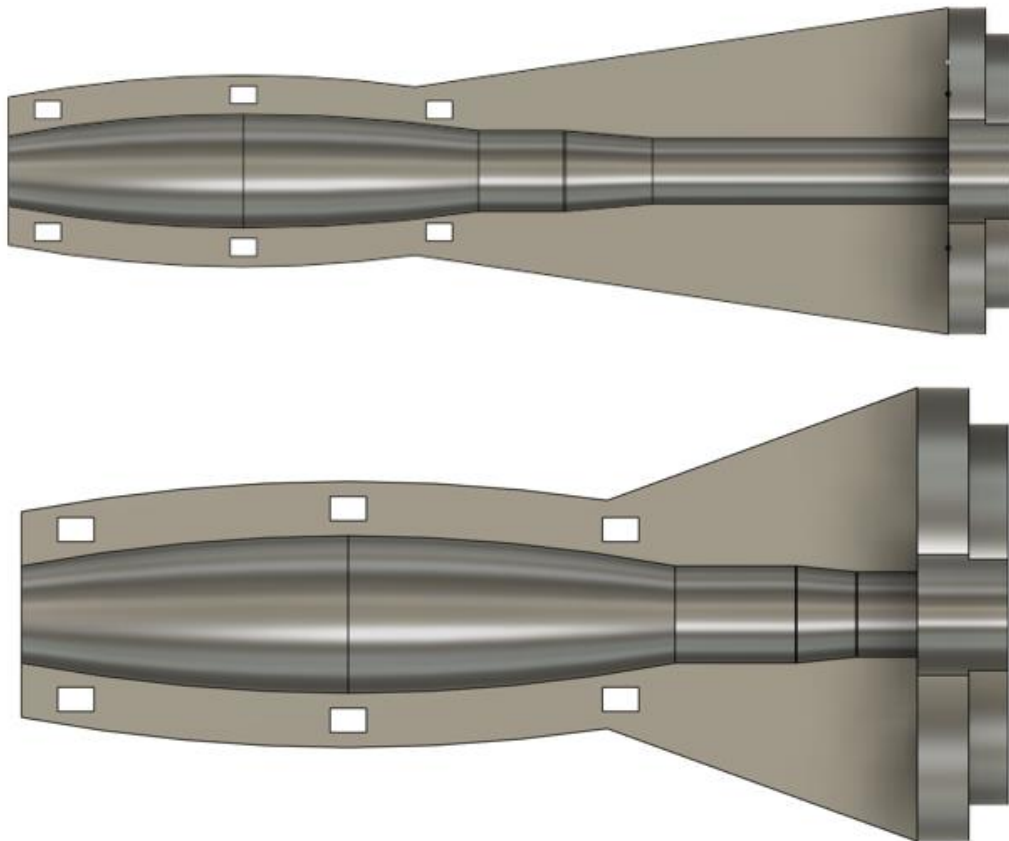


Ilustración 56. Comparación entre soportes (arriba el nuevo y abajo la primera versión).

Además, también hubo que modificar los nervios que rigidizan la pieza pues al tener más longitud se tuvo que reducir el grosor

Se planteó un cambio en la zona de anclaje para hacerlo encajar en la muñeca del robot creando un “negativo” de esta, como se puede observar a continuación. Además, contaría con una sección extraída mayor para poder dar más libertad al cable en caso de entrar en contacto con esta parte del soporte.

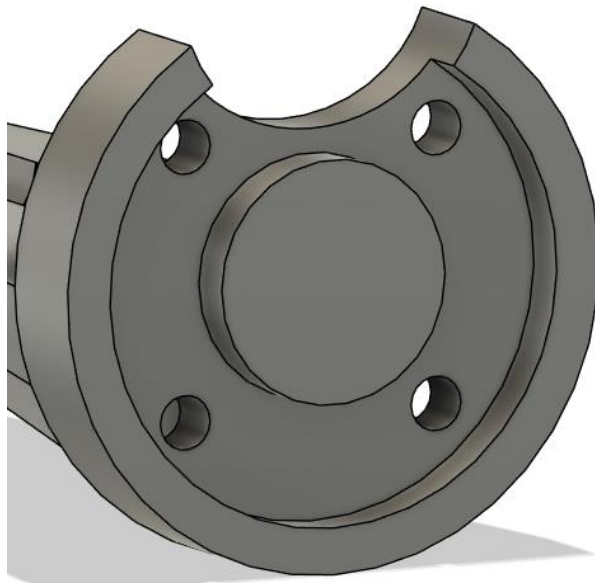


Ilustración 57. Mejora no implantada.

Este cambio no fue realizado pues en una versión posterior fue necesario otro cambio.

Por otro lado, se modificaron las patas extruidas de la parte superior del soporte para poder encajarlo de forma correcta y poder realizar el acople asegurándose con bridas para una mayor estabilidad. También se agrandó hasta los 23 mm de diámetro la parte que alojaría la zona reforzada del eje extensible para asegurar un encaje completo en el montaje final.

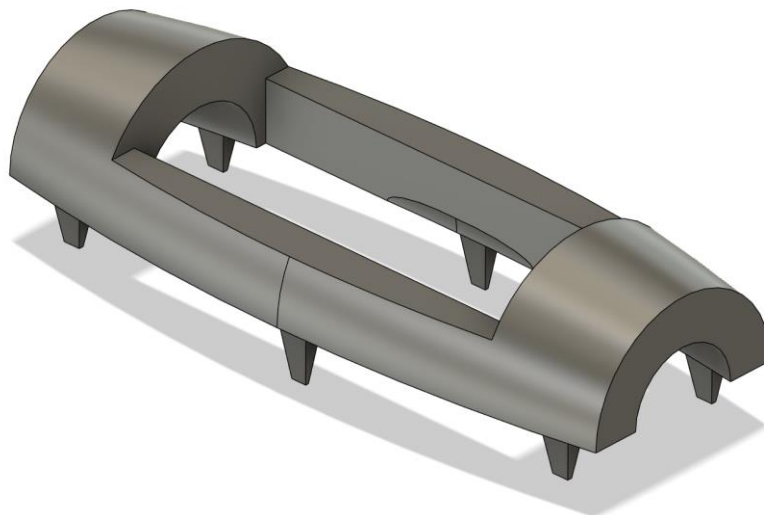


Ilustración 58. Segunda versión de la pieza complementaria al soporte.

Quedando las piezas finales de la siguiente forma:

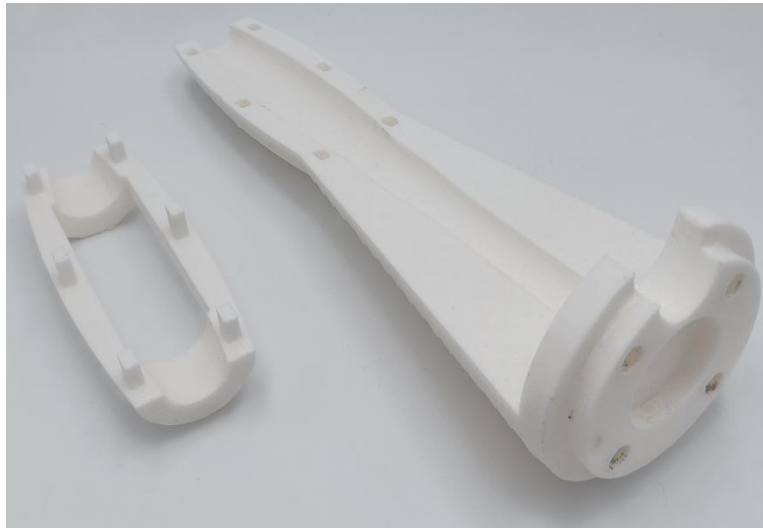


Ilustración 59. Segunda versión del soporte del eje extensible.

3.3.3 Tercera versión

La segunda versión del soporte funcionaba de forma exitosa, pero al avanzar el proyecto fue necesario modificarlo. Debido a la necesidad de implementar una cámara a la muñeca del robot, fue necesario modificar la parte que ancla el soporte del eje extensible al robot para permitir atornillar el soporte de la cámara que posteriormente se expondrá.

Debido a que el soporte del eje extensible original medía 15 mm, al añadir el grosor del soporte de la cámara los tornillos no llegaban a anclar de forma segura pues las arandelas necesarias para distribuir la fuerza y no dañar el material provocaban que apenas roscaran.

Para ello solo se modificó esta parte, reduciendo su grosor a 7,5 mm para permitir el correcto roscado quedando de la siguiente forma:

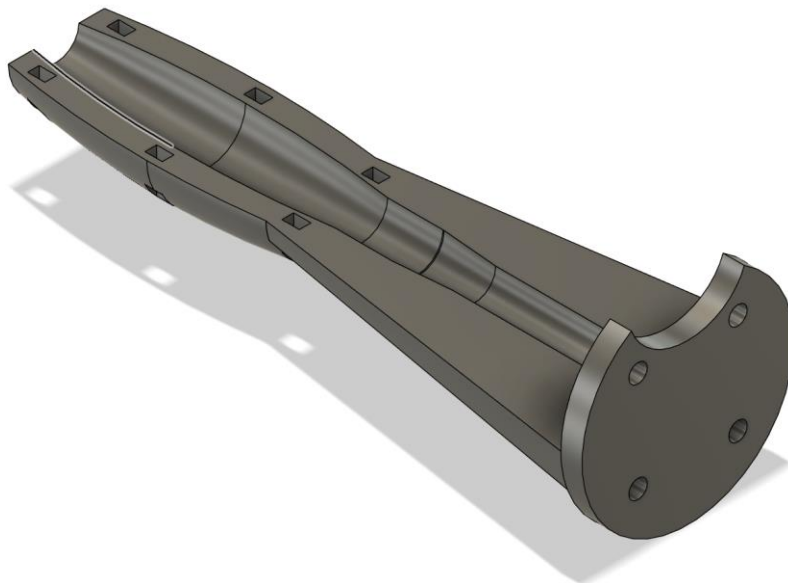


Ilustración 60. Tercera y última versión del soporte del eje extensible.

3.3 Diseño soporte OAK-D PRO

Por otro lado, para asegurar que la cámara OAK-D Pro tuviera una buena visión del proyecto, fue necesario colocarla en una posición donde se aprecie la cerámica al completo.

Para ello, se decidió situarla a unos pocos cm de distancia de la muñeca del robot UR 5, pues se puede colocar este último en una posición que permite el correcto visionado y análisis de la pieza.

Este soporte se consta de dos partes principales como se puede observar a continuación:

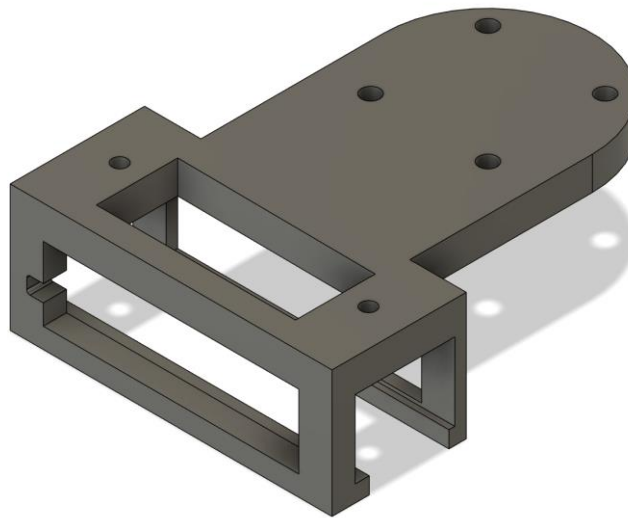


Ilustración 61. Soporte para OAK-D Pro.

La primera parte consta del círculo de 63 mm de diámetro que corresponde con las dimensiones de la muñeca del UR 5. Además, cuenta con agujeros de 5,5 mm para poder roscar con métrica 6 situados en un radio de 25 mm a 45 grados respecto la línea vertical de simetría.,

La segunda parte se trata del alojamiento de la cámara. El centro de esta se encuentra a 73,5 mm del centro de la primera parte. Esto es para asegurar que la imagen obtenida se centre en la cerámica y no en la herramienta que se encuentra por debajo de la cámara. Se trata de un volumen de 40,5 x 97 x 38,2 mm el cual se ha retirado parte del material para poder insertar la cámara por el lateral. La pieza cuenta unos agujeros situados en la parte superior para insertar los tornillos que sirven de fijación a la cámara para asegurar que no se mueve. Además, en la parte inferior se diseñaron unos bordes para que la cámara apoye en estos como mayor medida de seguridad.

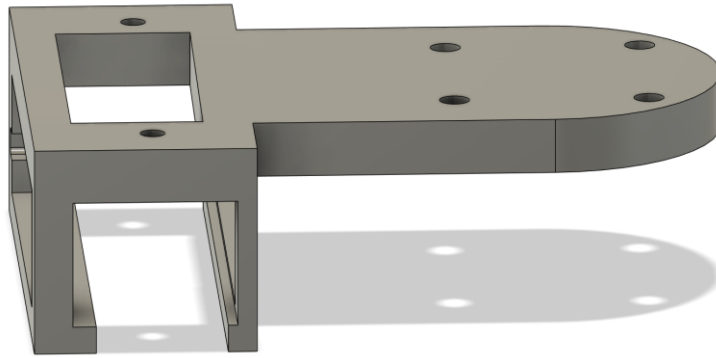


Ilustración 62. Vista lateral de soporte OAK-D Pro.

Como se puede observar, también tiene una apertura en la zona frontal situado donde se encuentran los puertos de comunicación para las conexiones están expuestos permitiendo conectar la cámara apartando los cables de la herramienta principal. También se puede apreciar que existe una apertura en la parte superior, sirviendo esta para ayudar a refrigerar la cámara pues las aletas de esta se encuentran situadas ahí.

Podemos observar cómo queda todo el montaje en las siguientes imágenes:

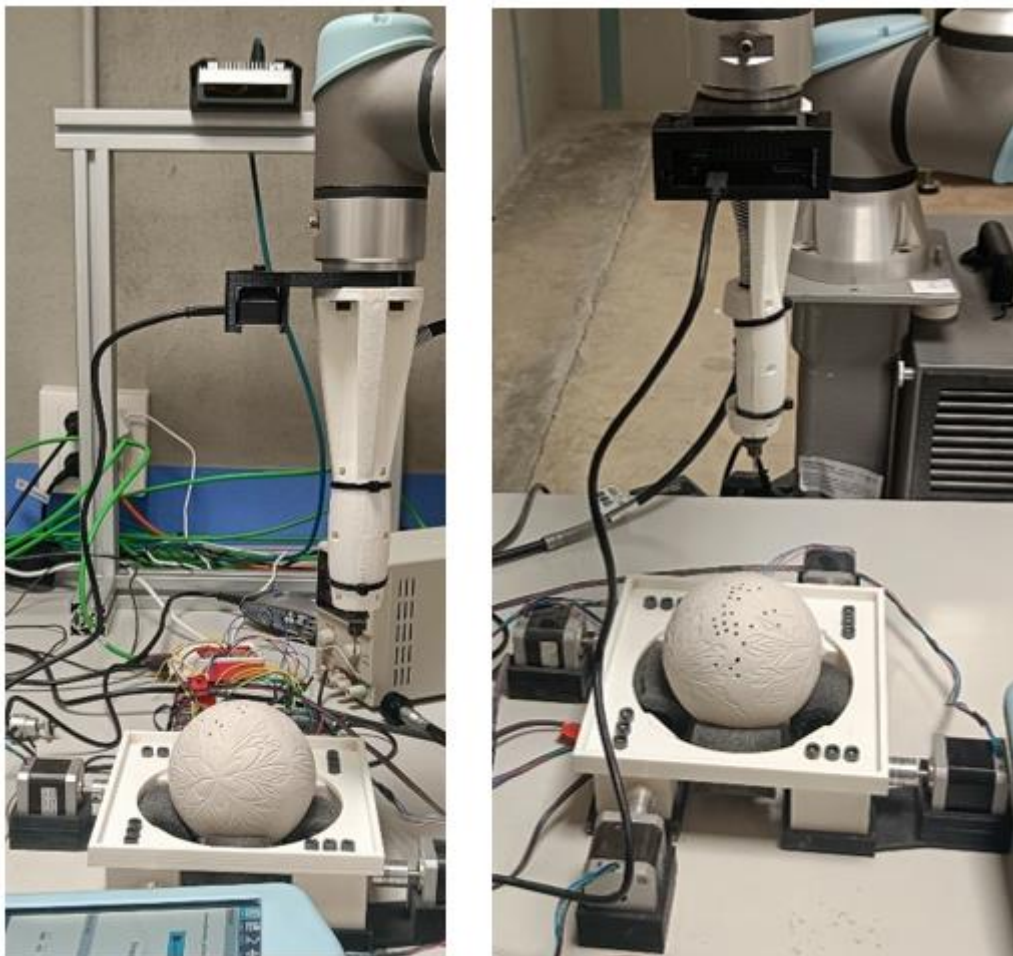


Ilustración 63. Montaje completo obtenido a través de impresión 3D desde distintos ángulos.

Capítulo 4

4. Controlador y comunicación

En este capítulo, se presenta el control del sistema. Durante este, se explica el sistema de comunicación visto desde la Raspberry Pi 5 comenzando por la comunicación entre la Raspberry y la base, presentando las bases y librerías que permiten dicho control y el movimiento coherente de la pieza cerámica.

Dentro de este contexto, se profundiza en el proceso seguido para poder realizar el control de la base, entrando en detalle en los tipos de comunicación posibles de una Raspberry Pi 5, cómo se opera y cómo es posible realizar el control de motores de forma más sencilla usando controladores intermediarios.

Finalmente se explica la implementación de la visión a través de la cámara OAK-D Pro, entrando en detalle sobre cómo se crean las líneas de comunicación a través de Python y cómo es posible procesar la imagen sin necesidad de entrenar una IA.

4.1 Consideraciones previas a la implementación

4.1.1 Selección del sistema empotrado

Para poder tener una comunicación exitosa entre el Robot UR 5, la cámara OAK-D Pro y la base donde apoya la pieza, es necesario tener un nodo que concentre y procese toda la información recibida, que a su vez debe enviar información e instrucciones a las partes necesarias para poder realizar el proceso o movimiento deseado.

Los dispositivos empotrados cumplen dicha función, diseñados específicamente para llevar a cabo tareas concretas, además de contar con características que permiten un rendimiento óptimo bajo ciertas circunstancias.

Estos sistemas están diseñados con hardware y software dedicado para obtener el mayor rendimiento posible. Con un tamaño compacto, bajo consumo de energía, permiten obtener un rendimiento superior en comparación con sistemas de propósitos generales asegurando eficiencia y rapidez en

tiempo real usando una cantidad de recursos menor. Cuentan también con gran robustez para garantizar el continuo funcionamiento para aplicaciones que así se desee.

Como se ha explicado en el Capítulo 2, el modelo seleccionado para este proyecto es una Raspberry Pi 5 de 4 Gb de RAM debido a:

- **Rendimiento:** Cuenta con un procesador Broadcom BCM2712 Quad-core Arm Cortex-A76 a 2,4 GHz que ofrece un gran rendimiento ideal para el procesamiento continuo de datos, teniendo capacidad suficiente para controlar otros sistemas al mismo tiempo.
- **Modularidad:** Debido a las reducidas dimensiones del sistema y de su consumo de 12 W, esta placa proporciona un funcionamiento confiable a su vez de facilidad de sustitución en caso de tener alguna avería, puesto que estos sistemas son económicos y altamente personalizable.
- **Conectividad:** Una Raspberry Pi 5 cuenta con una gran cantidad de opciones de conectividad, como puertos USB, Ethernet, Wifi, HDMI; que permite establecer comunicaciones exitosas y robustas en redes. Otro aspecto importante es la capacidad de interactuar con sistemas externos usando los pines GPIO para adaptar el sistema y enviar y recibir señales.

Para este proyecto, se propuso usar un SO propio de Luxonics (Empresa de la cámara a usar) debido a que al hacerse se implementa el sistema operativo necesario (RPi OS Linux) teniendo preinstalado DepthAI (se explicará de que se trata más adelante). La naturaleza en código abierto de Linux permite usar múltiples herramientas y bibliotecas de forma gratuita, siendo esta una clara ventaja frente a usar Windows. También ofrece protección contra ataques de malware pues Linux es menos vulnerable en comparación con Windows, protegiendo datos y el correcto funcionamiento de aplicaciones industriales cruciales.

4.1.2 Lenguaje de programación

Una vez seleccionado el sistema empotrado, se ha de plantear qué tipo de lenguaje será usado durante el desarrollo del sistema. Muchos proyectos industriales utilizan diversos lenguajes de programación en función de su propósito o complejidad, por lo que determinar cuál es el más indicado para este caso es importante.

En el desarrollo para aplicaciones industriales, los lenguajes requieren capacidad de manejar la complejidad y robustez necesaria para la industria, por lo que existen diversos candidatos a ser utilizados. C, C++, Java, Erlang, Python son algunos ejemplos de posibilidades. Todos estos son robustos y a prueba de fallos, por lo que son esenciales en entornos donde se requiera minimizar la posibilidad de error o fallos. C++ y Erlang ofrecen un gran rendimiento, permitiendo procesar rápidamente una gran cantidad de datos. Por otro lado, Java y C ofrecen escalabilidad, permitiendo adaptarse a cambios en la demanda y el entorno.

Para este proyecto, se estipuló que Python era la elección más apropiada, puesto que se es un lenguaje muy usado en aplicaciones similares. Debido a su profunda integración en “Machine learning”, Python se usa en todo tipo de aplicaciones relacionadas, como reconocimiento de lenguaje, “deep learning”, IA, etc. Debido además a su gran contenido en librerías gratuitas con todo tipo de funcionalidades, es el lenguaje ideal para el proyecto.

Además, gran parte de las pruebas, programas y aplicaciones proporcionadas por Luxonics para OAK-D Pro se encuentran programadas en Python, por lo que la implementación de este lenguaje es ideal además de poder seleccionar bibliotecas gratuitas que sean de ayuda.

4.1.3 Herramientas de compilación

Para poder procesar este código, es necesario usar un compilador. En este caso, se trata de Geany. Geany es una herramienta de código abierto que funciona en múltiples sistemas operativos como Linux, Windows, FreeBSD, macOS, etc. Solo en Windows faltan algunas características.

Una de las características más destacables de este compilador es la compatibilidad con distintos tipos de archivos incluyendo C, Java, PHP, HTML, Pascal, Python, etc. Además, presenta una interfaz flexible que facilita la configuración y la automatización de los sistemas que permite crear proyectos complejos evitando problemas de compatibilidad entre lenguajes o proyectos.

Otra ventaja es la gran cantidad de plugins que se pueden implementar para personalizar la experiencia del usuario en función de las necesidades requeridas. Estos se pueden escribir en C, Lua o incluso Python (para algunos es necesario usar otro plugin de compatibilidad) permitiendo crear algunos en caso de necesidad.

4.2 Montaje de controladores

Para controlar los motores durante el montaje de la base, es necesario usar señales PWM para mover de forma precisa estos. Como la creación de este tipo de señales puede ser compleja, se recomienda usar controladores, sobre todo al usar motores paso a paso como en el caso del proyecto.

Suponiendo un motor Nema 17 (1,2 A nominal y 1,5 Ohm por fase) aplicando 1,8 V para obtener esa intensidad nominal no se movería el motor. Para este caso, el fabricante indica que ha de conectarse a 12 voltios, según la ley de Ohm se obtendrían 8 A por la bobina, cosa que dañaría el motor.

Es por ello por lo que los controladores cuentan con limitadores de intensidad para poder conectar los motores a tensiones nominales superiores posible por su resistencia e intensidad máxima admisible. ·Estos limitadores interrumpen la señal creando una señal PWM de forma que el valor promedio de la intensidad es el valor nominal de la intensidad nominal de la bobina del motor, denominado "Chopping".

Debido a esto se usarán controladores para facilitar la operación de los motores seleccionados.

4.2.1 DRV8825

Estos se tratan de los DRV8825. Son unos controladores para motores paso a paso con mayor capacidad de corriente que su predecesor, A4988. Estos son ideales para el control de impresoras 3D y máquinas CNC, siendo compatible con el módulo del anterior mencionado controlador A4988.

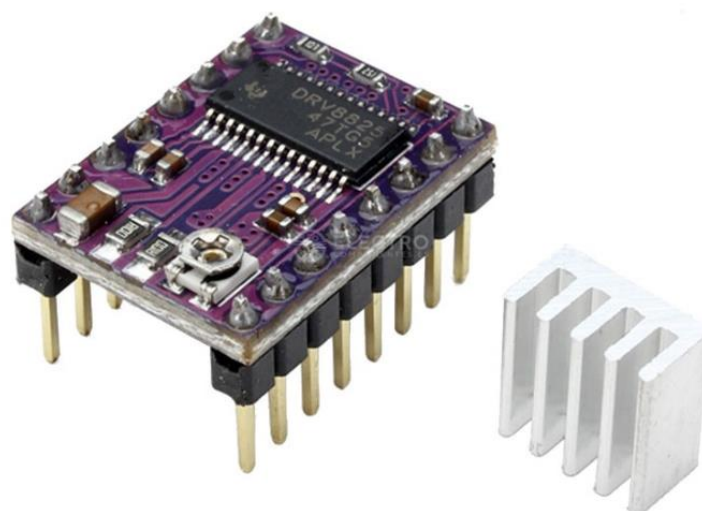


Ilustración 64. Controlador DRV8825 con disipador. Fuente: electrocomponentes.es.

Este cuenta con limitación de corriente ajustable, protección ante sobre corriente y sobre temperatura. Funciona en un rango entre 8,2 y 45 V DC, entregando aproximadamente 1,5 A por fase sin necesidad de disipador (el disipador junto a ventilación forzada permite llegar a 2,5 A, siendo un mayor voltaje dañino para el controlador por el amperaje y la temperatura alcanzada. Una característica clave de este controlador es la posibilidad de modificar el tamaño del paso a través de 3 pines, pudiendo obtener micro pasos de hasta 32 veces menor a un paso normal del motor.

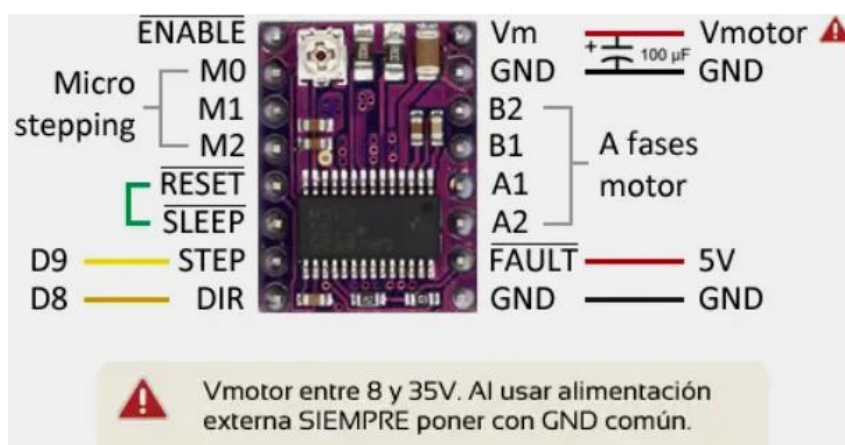


Ilustración 65. Esquema de los pines del controlador. Fuente: luisllamas.es

Como se puede observar en la imagen anterior, el controlador tiene un esquema simple de conexión a pesar de contar con bastantes pines. Antes de conectar el controlador, es imprescindible calibrar el dispositivo de forma adecuada para asegurar un comportamiento adecuado y seguro evitando dañar así las unidades. Para ello se deben seguir los siguientes pasos:

1. Conectar el controlador a la tensión de alimentación sin tener carga conectada ni microstepping (se explicará posteriormente qué es).
2. Medir con un multímetro la tensión entre la tierra (GND) y el potenciómetro (la parte del controlador que cuenta con una cabeza de tornillo estrella para calibrar).

3. Ajustar el valor obtenido girando dicha cabeza de tornillo hasta obtener el valor adecuado (en función del amperaje nominal del motor).
4. Una vez realizado, desconectar la alimentación, conectar el motor y volver a alimentar el controlador.
5. Con el motor conectado, verificar y ajustar de forma más precisa la corriente realizando el mismo proceso del punto 2 y 3.
6. Desconectar dicho controlador de la alimentación y conectar a la placa que se vaya a usar.

Dependiendo del modelo, el valor adecuado para la calibración varía. Esto se debe a que cada modelo de controlador tiene unas resistencias distintas. Para el caso del DRV 8825, se usará la siguiente fórmula como aproximación: $I_{max} = 2 * V_{ref}$, siendo V_{ref} la tensión medida en el paso 2.

La corriente a la que se ha de calibrar el sistema es mínimo del 60% y de forma recomendable del 100% de la corriente nominal del motor que se va a utilizar. En caso de utilizar microstepping, el valor medido en el amperímetro se trata del el 71% de la corriente nominal del motor, por lo que se ha de tener en cuenta al buscar el valor deseado.

El microstepping trata de una técnica que permite obtener pasos inferiores al paso nominal del motor. Cuando se usa un motor sin microstepping, los controladores aplican una frecuencia de 2 fases aplicando permanentemente el 71% de la corriente del limitador a cada bobina como se muestra a continuación.

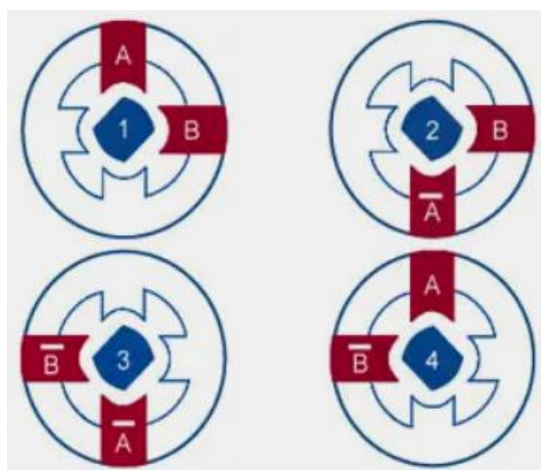


Ilustración 66. Esquema visual sobre cómo funciona un motor a pasos enteros.

Fuente: <https://www.luisllamas.es/motores-paso-paso-arduino-driver-a4988-drv8825/>

Ocurre una situación distinta cuando se realiza microstepping. El controlador llega a usar el 100% de la corriente en un determinado paso, pero esta cantidad varía en cada paso. Suponiendo una relación de 1/4 de paso, en el primer paso de la secuencia de giro se tendría la bobina A al 100% y la bobina B al 0%, en el segundo paso la bobina A al 92% y la bobina B al 38%, en el tercer paso la bobina A al 71%, y así sucesivamente.

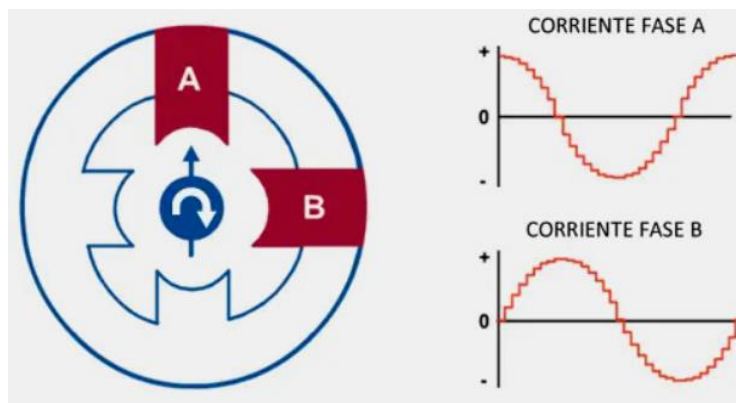


Ilustración 67. Muestra visual de cómo se obtienen señales senoidales a través de un sistema digital.

Para controlar el microstepping, es necesario poner en alto o en bajo los pines M0, M1 y M2 en función de lo que se quiera. A continuación, se muestra una tabla con la configuración para A4988 y DRV8825.

Tabla 2. Pasos de microstepping en función de pines activados en DRV8825 y A4988

Resolución		Pines M0, M1 y M2		
A4988	DRV8825	MODE0	MODE1	MODE2
Full step	Full step	Low	Low	Low
1/2 step	1/2 step	High	Low	Low
1/4 step	1/4 step	Low	High	Low
1/8 step	1/8 step	High	High	Low
-	1/16 step	Low	Low	High
-	1/32 step	High	Low	High
-	1/32 step	Low	High	High
1/16 step	1/32 step	High	High	High

Para el conectar y obtener el correcto funcionamiento del controlador se ha de hacer lo siguiente:

- La fuente de alimentación del motor ha de estar conectada a GND y a VMOT.
- Las bobinas del motor paso a paso se han de conectar a A1, A2 y B1, B2.

- La clavija GND debe estar conectada a tierra del microcontrolador y FAULT debe conectarse a 5 V.
- Step y DIR se conectan a un pin para poder llevar alto o bajo nivel según programado.
- RESET y SLEEP deben puentearse, de lo contrario el controlador no se enciende.
- La clavija ENABLE se puede desconectar, pues si se activa el controlador se apaga.
- M0, M1, M2 conectar en caso de que se vayan a usar en función del microstepping deseado.

4.2.2 Montaje de placa para controladores

Como se ha explicado anteriormente, el para poder controlar los motores instalados en la base es necesario usar controladores para facilitar la comunicación y la implementación de estos en el sistema. Para ello, se hizo uso de una placa proporcionada por el departamento de electrónica la cual permite conectar 4 controladores (uno para cada motor). Esto permite simplificar considerablemente la instalación, pues al estar en paralelo permite usar solo un único cable de alimentación de los controladores, una alimentación para los motores y una tierra común, además de poder extender cable para los pines de interés de cada controlador. También cuenta con condensadores necesarios, pues según el fabricante (Pololu) es recomendable usar condensadores de mínimo 47 picofaradios debido a que DRV8825 utiliza un condensador cerámico de baja energía, provocando que sea susceptible a los picos de voltaje destructivos de LC, especialmente cuando se utilizan cables de alimentación de más de unos pocos centímetros.

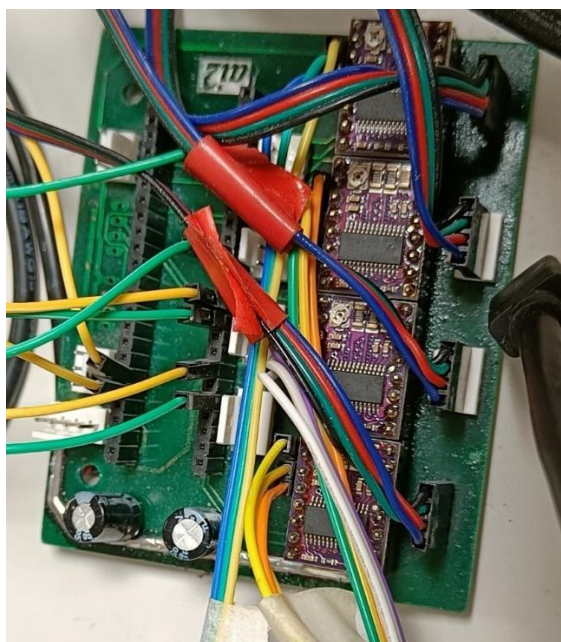


Ilustración 68. Placa para conectar 4 controladores DRV8825.

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico mediante visión y robótica colaborativa.

Para conectar los controladores a dicha placa sólo hay que introducir los pines de estos en sus correspondientes huecos a y ya están implementados. Como se observa en la imagen anterior, a la derecha se encuentran los pines para conectar las fases de los motores. A la izquierda de cada controlador se encuentran los pines para M0, M1 M2. Mas a la izquierda se tienen los pines de DIR y STEP de cada controlador, cada uno siendo de un color y estando unidos a pares para identificar cual es el correspondiente a cada controlador.

Para la alimentación se soldó un cable con 3 cables los cuales sirven para la alimentación de los controladores, la de los motores y tierra. Para la alimentación de los controladores, se conectó el cable correspondiente al pin 2 de la Raspberry puesto que proporciona de forma segura 5 V, como se aprecia a continuación:

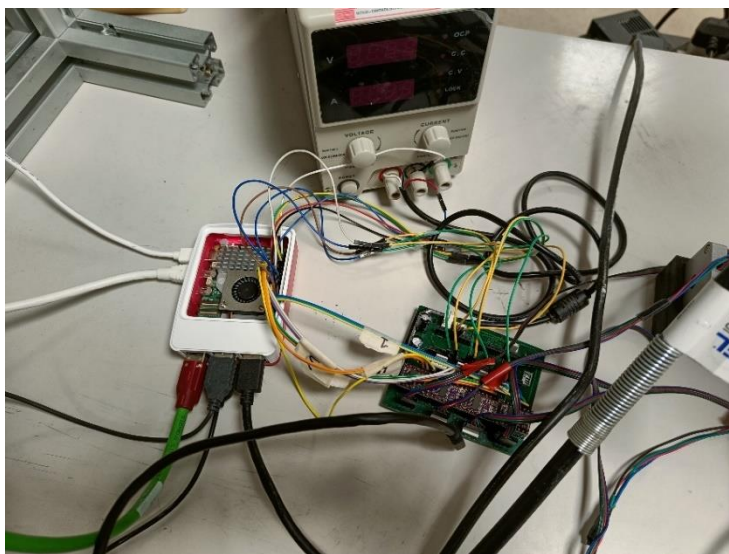


Ilustración 69. Conexiones necesarias para controlar el sistema.

Como se puede observar, también se ha conectado la alimentación de los motores a la fuente de alimentación y la toma a tierra de esta. Cabe destacar que la tierra de la Raspberry Pi 5 se ha conectado a la misma toma de tierra para que funcione el sistema correctamente.

La fuente de alimentación usada se trata de una RS-3005D. Se trata de una fuente capaz de generar voltajes de hasta 30 V con corrientes de hasta 5 amperios con un consumo de 150 W. Es una fuente ideal para diversas aplicaciones como producción, laboratorio en universidades, y otras aplicaciones debido a su precisión y fiabilidad.



Ilustración 70. Fuente de alimentación RS-3005D. Fuente: Rs-online.com

Una vez el sistema está preparado hay que realizar las conexiones pertinentes a la comunicación entre la placa que contiene los controladores y la Raspberry Pi 5.

Para poder conectar la placa de los controladores a la Raspberry Pi 5 es necesario usar los GPIO disponibles en el sistema, como se observa en la imagen anterior (LA DE PLACA).

A continuación, se muestra una tabla con la asignación de cada variable y su pin correspondiente:

Tabla 3. Pines asignados a cada variable.

Variable	Pin
DIR1_pin	20
STEP1_pin	21
DIR2_pin	12
STEP2_pin	16
DIR3_pin	14
STEP3_pin	15
DIR4_pin	25
STEP4_pin	8
M10	26
M11	19
M12	13
M20	6
M21	5
M22	0
M30	11
M31	9
M32	10
M40	22
M41	27
M42	17

En el apartado siguiente se explicará cómo se han asignado estas variables a los pines asociados.

Para la conexión de la cámara OAK-D Pro simplemente fue necesario conectar un cable a la entrada USB de la Raspberry Pi 5 y el otro extremo conectarlo a la cámara a través de USB-C.

4.3 Bibliotecas usadas

Para poder realizar el control de los motores, la conexión y el tratamiento de los datos de la cámara y la comunicación con el UR 5 es necesario usar las bibliotecas disponibles necesarias que permitan la coordinación de todas las partes.

4.3.1 Para motores - Gpiozero

La biblioteca en la que se centra el diseño para mover los motores se trata de gpiozero. Esto se debe a que la biblioteca usada hasta Raspberry Pi 4 se trataba de RPi.GPIO, pero por la arquitectura de Raspberry Pi 5 y actualizaciones del sistema, la biblioteca no estaba actualizada ni preparada para usarse. Cabe destacar que el funcionamiento de gpiozero es muy similar al de RPi.GPIO, aunque este último es algo más cómodo debido a su forma de invocar funciones.

Esta biblioteca fue creada por Raspberry Pi Foundation en 2013 y ha ido evolucionando durante el tiempo mejorando el soporte a las aplicaciones actuales. Esta cuenta con soporte a varios componentes GPIO, como son LEDs, botones, sensores, motores, etc. Este soporte consta de funciones y variables ya creadas para solo tener que activar, desactivar o configurar mínimamente facilitando la manipulación.

4.3.2 Para OAK-D Pro

Para poder conectar la cámara y transmitir los datos es necesario utilizar diversas bibliotecas, siendo varias propias para este propósito.

Estas bibliotecas son: cv2, depthai (como dai), threading blobconverter.

La biblioteca principal de este caso se trata de depthai, puesto que está creada por Depth para implementar IA en distintos sectores como en el sector sanitario. En este caso, Luxonics lo proporciona para realizar el control de sus productos, estando disponible para Python, C++ y Java.

Cv2 es una biblioteca usada para leer imágenes, manipular y mostrar imágenes en pantalla. Creada en 2000 admite muchos algoritmos y está disponible para múltiples lenguajes de programación como Java, C++, Python, etc.

Therading es también una biblioteca importante en este proyecto, pues esta permite ejecutar en paralelo partes de código que se definan.

A través de Depthai se realizan la conexión pipeline de la cámara al controlador. Una vez creada, hay que definir y crear los nodos que actúan como fuentes de información y las salidas, usando los comando que se encuentran a continuación:

```
# Create pipeline
pipeline = dai.Pipeline()
# Define sources and outputs
camRgb = pipeline.create(dai.node.ColorCamera)
monoLeft = pipeline.create(dai.node.MonoCamera)
monoRight = pipeline.create(dai.node.MonoCamera)

edgeDetectorLeft = pipeline.create(dai.node.EdgeDetector)
edgeDetectorRight = pipeline.create(dai.node.EdgeDetector)
edgeDetectorRgb = pipeline.create(dai.node.EdgeDetector)

xoutEdgeLeft = pipeline.create(dai.node.XLinkOut)
xoutEdgeRight = pipeline.create(dai.node.XLinkOut)
xoutEdgeRgb = pipeline.create(dai.node.XLinkOut)
xinEdgeCfg = pipeline.create(dai.node.XLinkIn)

edgeLeftStr = "edge left"
edgeRightStr = "edge right"
edgeRgbStr = "edge rgb"
edgeCfgStr = "edge cfg"

xoutEdgeLeft.setStreamName(edgeLeftStr)
xoutEdgeRight.setStreamName(edgeRightStr)
xoutEdgeRgb.setStreamName(edgeRgbStr)
xinEdgeCfg.setStreamName(edgeCfgStr)
```

Ilustración 71. Parte del código que crea las conexiones con la cámara.

En la imagen anterior se observa cómo se han creado unas conexiones para cada una de las 3 cámaras de las que dispone OAK-D Pro, se crea la conexión con el nodo para detección de bordes y para la conexión que permitirá transmitir las imágenes por pantalla de cada una de estas cámaras.

Una vez realizado esto, se pasa a seleccionar las propiedades de cada cámara (las calidades de video para la cámara RGB es de 1080p mientras que las laterales son de 720p).

A pesar de que se tienen las conexiones necesarias para usar las 3 cámaras, se decidió usar solo la central pues es la que más calidad de video presenta. Es por ello se selecciona el mayor tamaño de output size frame para obtener el mayor tamaño de relación de aspecto posible para la cámara RGB.

```
# Properties
camRgb.setBoardSocket(dai.CameraBoardSocket.CAM_A)
camRgb.setResolution(dai.ColorCameraProperties.SensorResolution.THE_1080_P)

monoLeft.setResolution(dai.MonoCameraProperties.SensorResolution.THE_720_P)
monoLeft.setCamera("left")
monoRight.setResolution(dai.MonoCameraProperties.SensorResolution.THE_720_P)
monoRight.setCamera("right")

edgeDetectorRgb.setMaxOutputFrameSize(camRgb.getVideoWidth() * camRgb.getVideoHeight())

# Linking
monoLeft.out.link(edgeDetectorLeft.inputImage)
monoRight.out.link(edgeDetectorRight.inputImage)
camRgb.video.link(edgeDetectorRgb.inputImage)

edgeDetectorLeft.outputImage.link(xoutEdgeLeft.input)
edgeDetectorRight.outputImage.link(xoutEdgeRight.input)
edgeDetectorRgb.outputImage.link(xoutEdgeRgb.input)

xinEdgeCfg.out.link(edgeDetectorLeft.inputConfig)
xinEdgeCfg.out.link(edgeDetectorRight.inputConfig)
xinEdgeCfg.out.link(edgeDetectorRgb.inputConfig)
```

Ilustración 72. Definición de propiedades se unen todos los nodos creados.

Por último, se enlaza todo lo que se ha creado para así obtener de forma definitiva la conexión con la cámara.

Una vez obtenida, para poder transmitir por una pantalla solo es necesario obtener la cola de retransmisión:

```
# Output/input queues
edgeRgbQueue = device.getOutputQueue(edgeRgbStr, 1, False)
edgeCfgQueue = device.getInputQueue(edgeCfgStr)
```

Ilustración 73. Cola de la retransmisión.

Cuando se obtiene, es posible obtener los datos en pantalla del Frame creado (una matriz con los códigos de 0 a 255 de colores con los que se construye la imagen) y esta se puede representar en pantalla usando la biblioteca cv2, a través del comando cv2.imshow("nombre de retransmisión", "Frame a retransmitir"). Para este caso en concreto, se decidió recortar el Frame para solo operar en la zona de interés, que corresponde con el rango de filas y columnas dentro de la matriz de Frame general que se han seleccionado (esta parte corresponde a la zona donde se ubica la pieza cerámica al estar el robot en el punto de verificación). Tras esto, se añade a la imagen el filtro de Sobel para obtener el gradiente de intensidad de los píxeles de cada imagen y se envía a la pantalla. A continuación, se puede observar el código usado:

```
edgeRgb = edgeRgbQueue.get()
global edgeRgbFrame
edgeRgbFrame = edgeRgb.getFrame()

edgeRgbFrame = edgeRgbFrame[350:750, 750:1150]

#cv2.rectangle(edgeRgbFrame, (750, 350), (1150, 750), (255, 0, 0), 2)

# Show the frame
cv2.imshow(edgeRgbStr, edgeRgbFrame)

key = cv2.waitKey(1)

cfg = dai.EdgeDetectorConfig()
sobelHorizontalKernel = [[1, 0, -1], [2, 0, -2], [1, 0, -1]]
sobelVerticalKernel = [[1, 2, 1], [0, 0, 0], [-1, -2, -1]]
cfg.setSobelFilterKernels(sobelHorizontalKernel, sobelVerticalKernel)
edgeCfgQueue.send(cfg)
```

Ilustración 74. Interior del bucle de la función camera.

En las siguientes imágenes se ofrece una comparación entre obtener el frame completo y la selección de la zona de interés de trabajo a través de seleccionar unas filas y columnas concretas de la matriz completa:

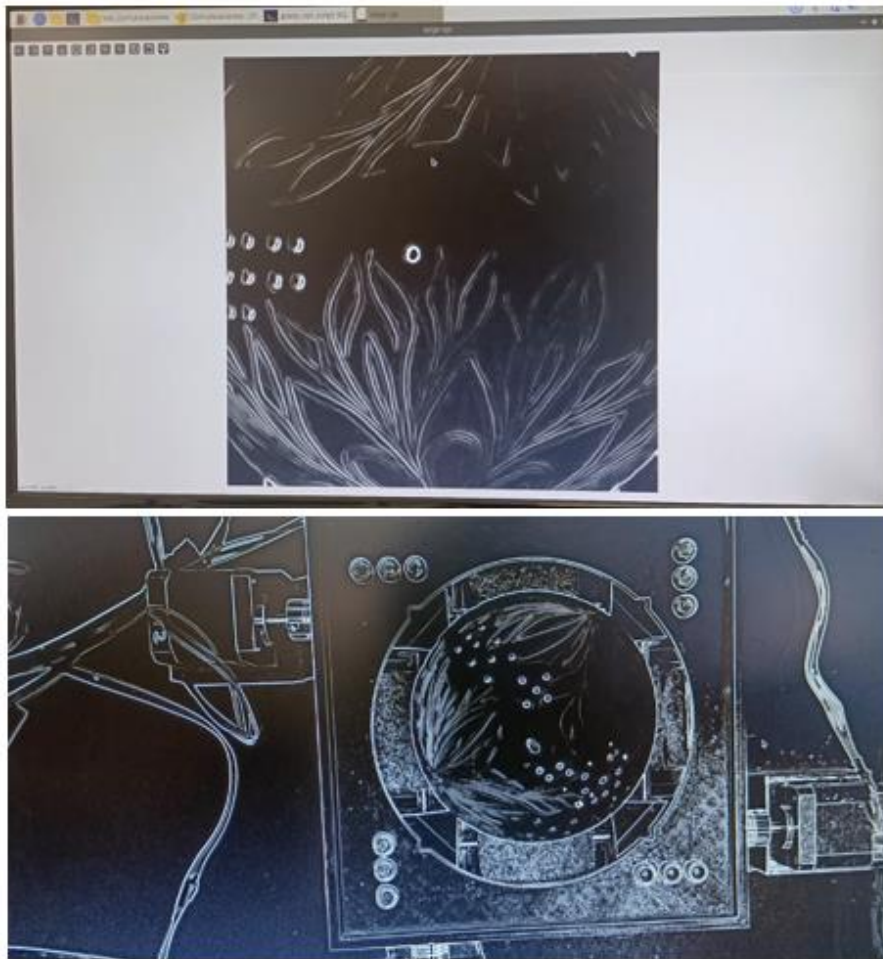


Ilustración 75. Comparación entre la imagen sin recortar (abajo) y la imagen recortada (arriba).

4.3.3 Para UR 5

Para la comunicación con el brazo robot UR 5 se utiliza un sistema TCP a través de sockets. Para ello se utiliza la librería propia de Python “socket”. La API sistema permite establecer comunicación accediendo a los servicios IPC proporcionados.

Resumiendo este sistema, se trata de un sistema que crea un flujo de bytes ordenado y fiable que evita la pérdida de información. Para ello, se crea una conexión entre servidor y cliente a través de una conexión IP mandando bytes de forma fiable y en orden.

4.4 Construcción del programa

Crear un software funcional es crítico para el funcionamiento del proyecto. Para ello es necesario que la construcción y la compilación de este sean prioridades al considerar los requerimientos del código. Al requerir construcción se implica la necesidad de organizar y configurar los componentes del código; mientras que, al referirse a compilación, se hace referencia a crear un código ejecutable o en un formato preparado para hacer funcionar el hardware necesario.

LA estructura del código se basa en 3 partes principales: la iniciación y asignación de variables, la función main y la función cámara.

En la iniciación, se crean y asignan las variables y conexiones necesarias para el funcionamiento del sistema. En esta parte se asignan los pines GPIO usando “gpiozero.OutputDevice(X)”, siendo “X” el pin elegido, eligiendo “Output” o “Input” en función si se quiere un pin de entrada o salida. También se crea la conexión con el brazo robot UR 5 a través de sockets, usando la librería “socket” como se muestra a continuación:

```
# Creación de SOCKET
host = 'localhost'

data_payload = 2048

backlog = 5

HOST="158.42.125.169" # Dirección del host
PORT=3000 # Puerto a usar

# Se crea socket de TCP
sock=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server_address = (HOST, PORT)
sock.bind(server_address)

sock.listen(backlog)
client, address = sock.accept()
data = client.recv(data_payload)
```

Ilustración 76. Esquema para empoezar las comunicaciones siendo el servidor.

Tras esto, se define el número de pasos que se quiere realizar con los motores stepper y se añade el retardo necesario para que no se sobrepongan envíos a la hora de mandar señales a los motores. Por último, se crean las conexiones para obtener información de la cámara. Para ello se realiza la creación de una “pipeline” y se crean los nodos necesarios para poder usar la IA para usar un filtro de Sobel (filtro para hacer resaltar los bordes en una imagen) y seleccionar la resolución de la imagen. Cabe destacar que el código relacionado con la conexión de la cámara se obtuvo de un ejemplo de la biblioteca de ejemplos de Luxonics, por lo que las modificaciones de esta parte son mínimas.

Tras esto se pasa a la segunda parte del código. Esta está formada por una función llamada “camara”. Esta función es importante pues tras definirla se va a usar a través de la biblioteca “threading” para que funcione en paralelo a la función main, para así realizar el bucle con las imágenes más actuales posibles y no volver a perforar en una zona ya perforada. Esta función trata crear la cola de ejecución de la cámara, y en un bucle obtener la imagen más reciente, reducir el tamaño de la imagen a la sección deseada (en este caso, la zona donde se encuentra la cerámica), mostrar la imagen por pantalla para verificar el correcto funcionamiento y aplicar el filtro. En el caso de apretar la tecla “0”, se cierra el bucle y la función.

Por último, se encuentra la función main. En primer lugar, se encuentran dos “if” que verifican si es la primera iteración del bucle para dejar un tiempo para que la cámara cree la imagen y enfoque correctamente. Tras esto, se encuentran una serie de “if” que verifican si en unas posiciones determinadas existen píxeles con un valor mayor a 40, siendo esto indicativo de la existencia de un borde o agujero previo. Estas posiciones se encuentran localizadas en una configuración donde en el centro de la imagen se encuentra un cuadrado de 32 x 32 píxeles. Este cuadrado es el cuadrado central de un cuadrado mayor dividiendo el espacio en nueve secciones, como se muestra a continuación donde se ha perforado en el centro de estas 9 secciones.



Ilustración 77. Posiciones de perforación y delimitaciones.

Con cada comprobación, se añade 1 a una variable llamada “Tot” para saber el total de agujeros y se suma 1 a la variable de la sección correspondiente para posteriormente verificar si existe un borde previo en esa zona. Tras esto se guarda el valor de “Tot” en otra variable llamada “Cont”.

Para saber dónde se han de realizar los agujeros o no, se verifica si las variables propias de cada sección son distintas de 0 con una cadena de “if”. Si alguna lo es, se envían las coordenadas a perforar al robot usando bytes con codificación ascii y se resta 1 a “Tot” si es mayor que 0. En caso contrario, se manda un mensaje indicando que existe un borde previo a la pantalla de comandos. La forma de enviar esta información tiene una estructura específica como se puede apreciar:

```
client.send(bytes(f"(-0.01,0.01,0.06172,0,3.089,-0.061,{Tot})",encoding='ascii'))
```

Ilustración 78. Línea que envía una coordenada.

Se indica que son bytes en codificación ascii, pero además de los seis valores que indican la posición respecto la base del robot UR de la punta de la herramienta del UR llamada TCP (siendo coordenadas X, Y, Z, RX, RY, RZ) también se envía el valor actual de la variable “Tot”. El motivo de esto se explicará durante la exposición del código generado para el robot UR.

No es necesario esperar a que se ejecute la acción para enviar la siguiente posición pues UR cuenta con un sistema que guarda las variables y conforme se van recibiendo se realizan las operaciones en el orden recibido.

Una vez hecho esto se verifica si se han realizado los agujeros necesarios comparando “Tot” y “Cont”. “Tot” debe ser 0 y “Cont” será el número de agujeros realizados en esa iteración. Tras esto, se esperan 9,2 segundos por “Cont” para asegurar que no se mueve la cerámica mientras se está perforando y se pone una variable llamada “iv” a 0.

Por último, queda el movimiento de la cerámica usando los motores y GPIO. Primero se comprueba que iv es igual a “0”. Si se cumple, se pone una variable llamada “mov” a 0 y se realiza un bucle

“while(True)” donde se activan y desactivan los pines necesarios para mover los 2 rodillos pares que mueven la pieza hacia abajo, mostrando las partes superiores y donde se suma 1 a “mov”. Este movimiento se realiza hasta que se detecta un píxel con un valor mayor de 30 en el borde superior del rectángulo que encapsula las superficies o hasta que la variable “mov” es igual a 3 para limitar el movimiento.

Después de esto, se realiza la misma operación, pero para mover la pieza hacia la izquierda mostrando la zona derecha de esta. En caso de que una vez realizado esto, todos los píxeles del borde derecho no tengan un valor superior a 30, se asigna 1 a “iv”, 0 a “mov”, se espera un segundo y se reanuda el bucle. En caso opuesto, se asigna 3 a “iv” y se verifica que mientras el borde derecho y el borde superior tengan algún píxel con valor mayor que 30, que se mueva la pieza hacia abajo hasta que mov sea igual a 3. Esto se realiza para que, en caso de que el cuadrado donde se verifican las secciones se encuentre en una zona donde hay tallado, pueda desplazar la pieza hasta poder perforar.

Tras esto, se ha explicado correctamente todo el código desarrollado para la comunicación desde la Raspberry Pi 5 y los distintos componentes del proyecto. A continuación, se muestra un diagrama con el resumen de lo explicado.

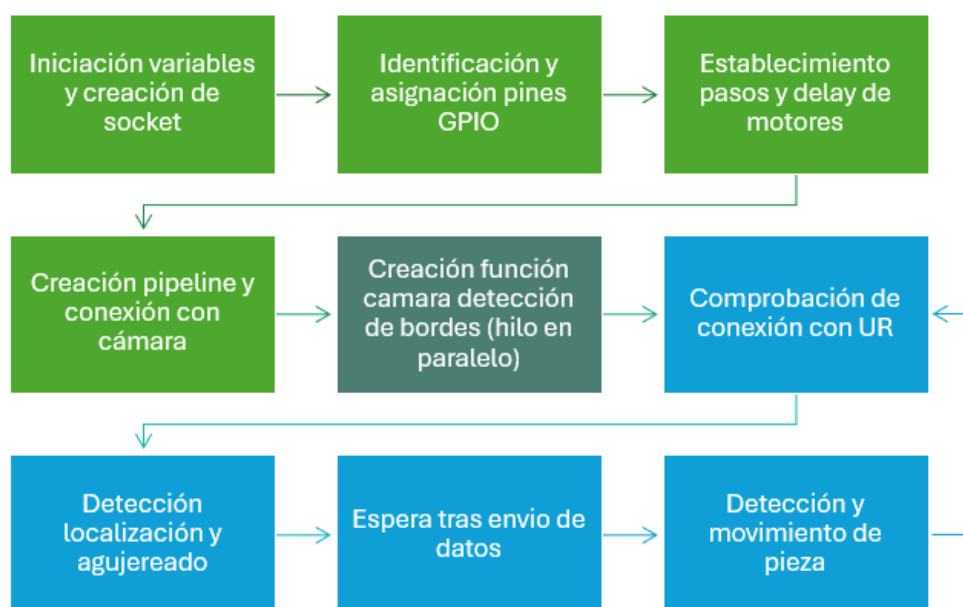


Ilustración 79. Diagrama con el funcionamiento del código explicado anteriormente explicado.

Capítulo 5

5. Control y maniobras brazo robot

Este capítulo está centrado en la comunicación desde el punto de vista del brazo robot UR 5 y la programación que debe ejecutar para realizar las perforaciones correctamente. La primera parte se centra en el desarrollo de los programas iniciales donde se iban implementando los cambios que se realizaban a la base de la cerámica conforme avanzaba el proyecto.

La segunda parte está centrada en el código final, en el funcionamiento del robot y los parámetros seleccionados para este.

5.1 Consideraciones previas

UR cuenta con distintas maneras de establecer comunicaciones utilizando interfaces varias para comunicarse con dispositivos externos:

- Comunicación mediante sockets: Este es uno de los métodos más empleados debido a la facilidad de uso. Gran parte de los servidores remotos disponibles de UR emplean TCP/IP, aceptando varios comandos desde un programa cliente siendo este otro robot un programa externo. Además, el lenguaje URScript permite abrir y cerrar sockets junto a enviar y recibir distintos tipos de formatos de información.
- Servidor Dashboard: Este proporciona acceso a todo tipo de funciones como por ejemplo para cargar programas, apagar y encender el robot, proporcionar información sobre el estado de este, etc. Estos comandos llegan a la interfaz a través de la conexión TCP/IP, permitiendo también configurar niveles de acceso del usuario.
- XML-RPC: Este sistema no es en sí una comunicación robot a robot, pero es útil para ejecutar funciones que no están implementadas a través de URScript. Se puede considerar como un programa que consta en un lenguaje de programación durante su ejecución, complementando las capacidades de URScript con funcionalidades de otros lenguajes. A través de estas invocaciones es posible realizar cálculos complejos y recibir datos estructurados.

- Interfaz Primaria/Secundaria: El controlador de UR proporciona servidores para enviar datos del estado del robot y recibir comandos de URScript. Mientras que los datos de estado del robot y mensajes adicionales los transmite la interfaz primaria, la interfaz secundaria solo es capaz de transmitir datos de estado del robot. Estos datos son la principal fuente de comunicación entre la interfaz a nivel usuario y el controlador del robot, permitiendo controlar el robot de forma remota sin usar un programa as una velocidad de actualización de 10 Hz.
- Interfaz a tiempo real: Este sistema funciona de forma similar al anterior, siendo la principal diferencia la velocidad de actualización, llegando a los 50 Hz.
- RTDE (Real-Time Data Exchange): Este sistema es el sustituto del sistema anterior, puesto que permite transmitir datos personalizados y aceptar puntos personalizados y registros. Además, permite el desarrollo de aplicaciones en lenguajes como Python y C++ a través de API de la biblioteca.

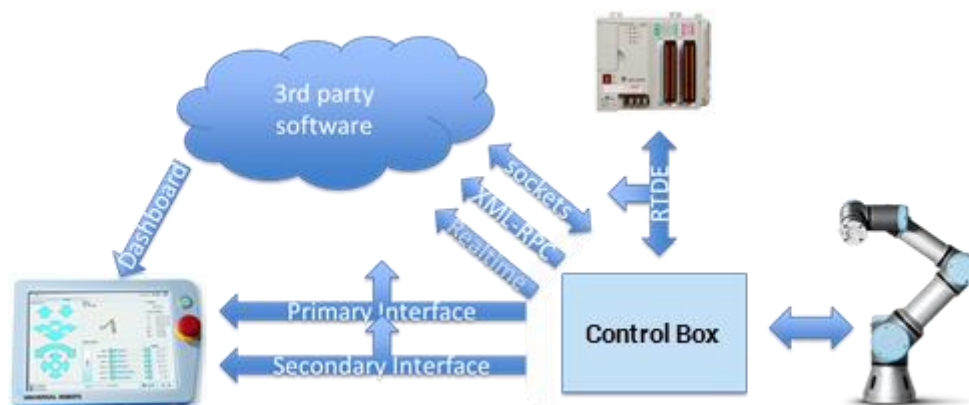


Ilustración 80. Esquema que resumen los tipos de comunicaciones que poseen los robots de UR. Fuente: Universal Robots

Todas estas herramientas facilitan la programación e incorporación de los cobots en las empresas ajustándose así al mercado pues permite personalizar soluciones de forma rápida y sencilla, pues estos robots tienen un enfoque dirigido al usuario.

Además, UR cuenta con una cierta variedad de robots incluyendo kits de automatización de aplicaciones con componentes certificados UR+ (más de 340 kits). Debido a esto se tiene la posibilidad de seleccionar qué modelo es el más indicado para cada aplicación.

Para el caso de este proyecto, el más indicado se trata del modelo UR 3e, puesto que sus dimensiones reducidas no impiden obtener los resultados deseados. En cambio, debido a diversas circunstancias no fue posible usar este modelo. En primer lugar, en el laboratorio donde se realizaron las pruebas y la investigación de este proyecto no hay disponible ningún UR 3 ni UR3e. En segundo lugar, al ser proporcionada como herramienta de perforación una Dremel, esto provoca que el desplazamiento de la punta de la herramienta sea demasiado grande al compararse con las dimensiones del UR3 (en caso de crear una herramienta con un motor en la muñeca del robot sería posible, pero no es el caso). Debido a esto se decidió usar los modelos de robot disponibles en el laboratorio, un UR 5 y un UR 5e.

5.2 Desarrollo

En un primer momento se tenía disponible el robot UR 5. Mientras se realizaba el diseño de la primera iteración de la base, se usó una máquina virtual para poder simular el UR 5 para comprobar que las medidas y la localización de la base permitiera el perforado de la pieza cerámica.

Esta máquina virtual se trata de Oracle VM Virtualbox, usando Ubuntu (Linux) para crear el sistema operativo virtual para ejecutar la versión 3.15 de URSim. Esta versión fue obtenida a través de la página de Universal Robots. Este simulador contiene una interfaz igual al encontrado en los robots, permitiendo el igual manejo como si se tratara del robot real. Además, se tiene la ventaja de poder importar los archivos para no tener la necesidad de trabajar directamente en el robot.

A través de esta simulación, se realizaron varias pruebas y códigos verificando la posibilidad y la localización de la cerámica para seleccionar el punto donde se ubicaría a la hora de perforar.

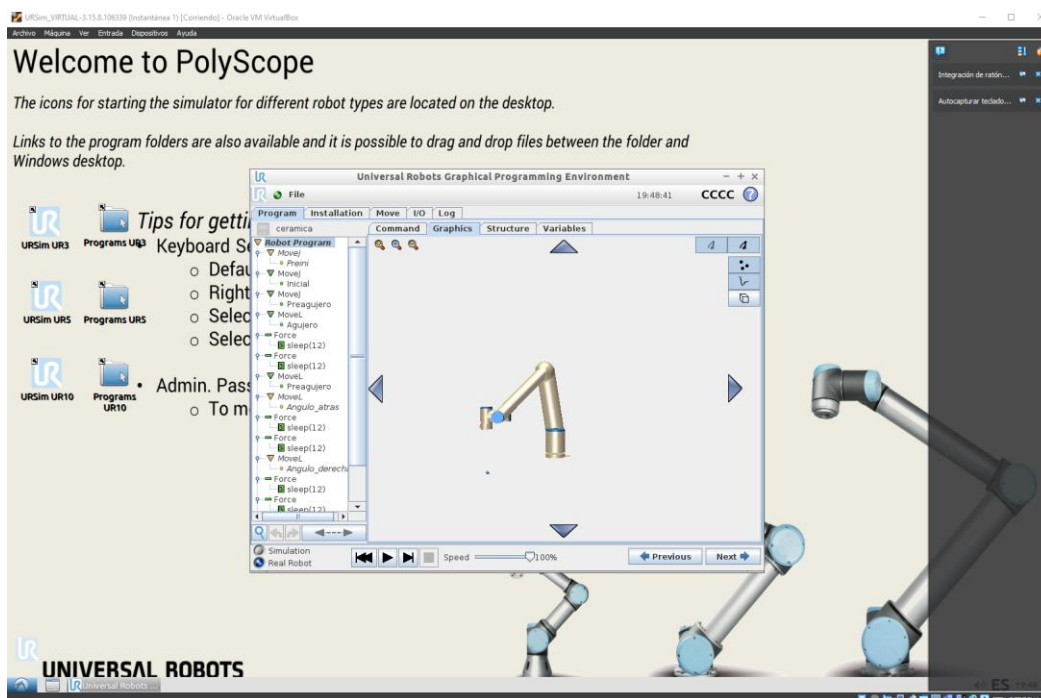


Ilustración 81. Imagen del simulador offline de un UR 5 CB.

A través de este sistema se comprobaron varias configuraciones para comprobar el funcionamiento del sistema.

Una vez se estableció un sistema, se realizó una prueba con el robot real. Esta prueba resultó no ser exitosa en un primer momento debido a algunos problemas. En primer lugar, el robot no operaba correctamente usando la plantilla disponible en Polyscope (programa propio de UR para facilitar la programación en el teach pendant) para realizar los movimientos de fuerza. Tras diversos intentos, se consiguió realizar a través de código URScript de forma manual. Esto confirmó que el robot no operaba correctamente cuando se pedía que ejerciera alguna fuerza, siendo que solo ejercía fuerzas mayores de 15 N y sólo en dirección vertical ascendente (algunas ocasiones se realizaba descendente).

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico mediante visión y robótica colaborativa.

Esto ocasiona problemas pues para realizar perforaciones en la pieza cerámica se requiere menos cantidad de fuerza, pues al ser cerámica sin tratar es fácil de dañar y de crear un escamado alrededor de la perforación realizada. Además, al realizar pruebas con menores fuerzas no se alcanzaba a perforar en el caso de que la herramienta se desplazara correctamente.

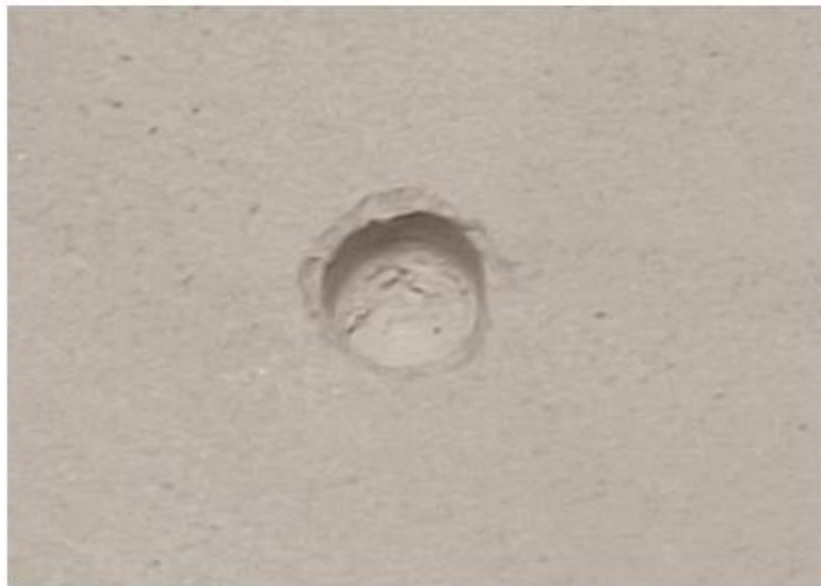
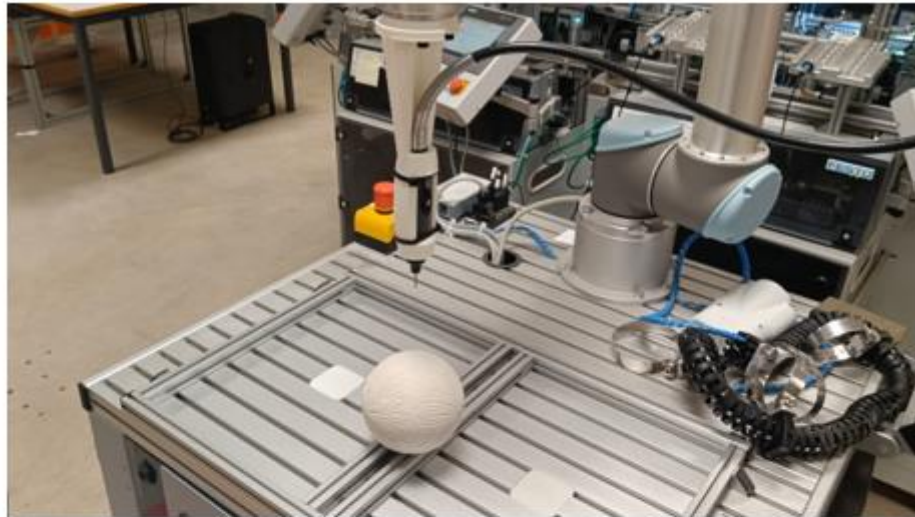


Ilustración 82. Frame de un vídeo junto un aumento a una perforación no realizada correctamente con escamado.

Tras la comprobación, se decide usar el robot contiguo, un UR 5e. Este opera con una versión más reciente de Polyscope, siendo esta la 5. A partir de este momento se procede a usar el robot para realizar las pruebas, sin necesidad de usar el simulador pues se encontraba disponible en ese momento.

Al realizar pruebas de perforación, se comprobó que una fuerza de 5 N a 2mm/s es suficiente para perforar la cerámica de forma limpia y precisa.

5.3 Configuración y código definitivo

Una vez se realizó la instalación correcta de la Dremel junto a la cámara OAK-D Pro, se midió la distancia hasta la punta de la broca obteniendo 275 mm desde la base de la muñeca del UR 5e.

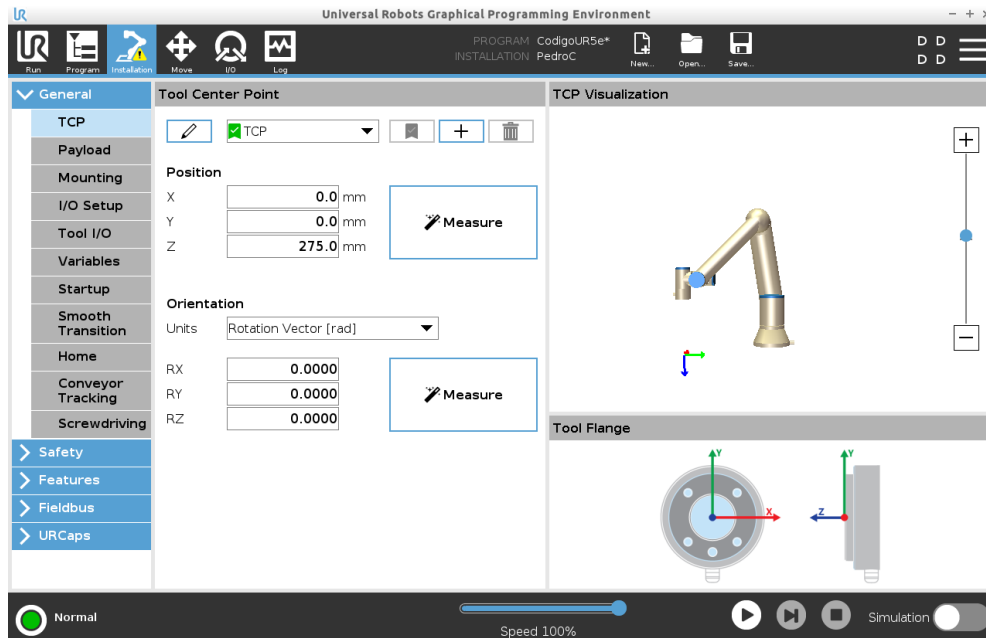


Ilustración 83. Localización del TCP en Polyscope de UR 5e.

También se realizó el auto calibrado para obtener el centro de gravedad del montaje y así mover con más precisión el brazo robot, obteniendo los siguientes resultados:

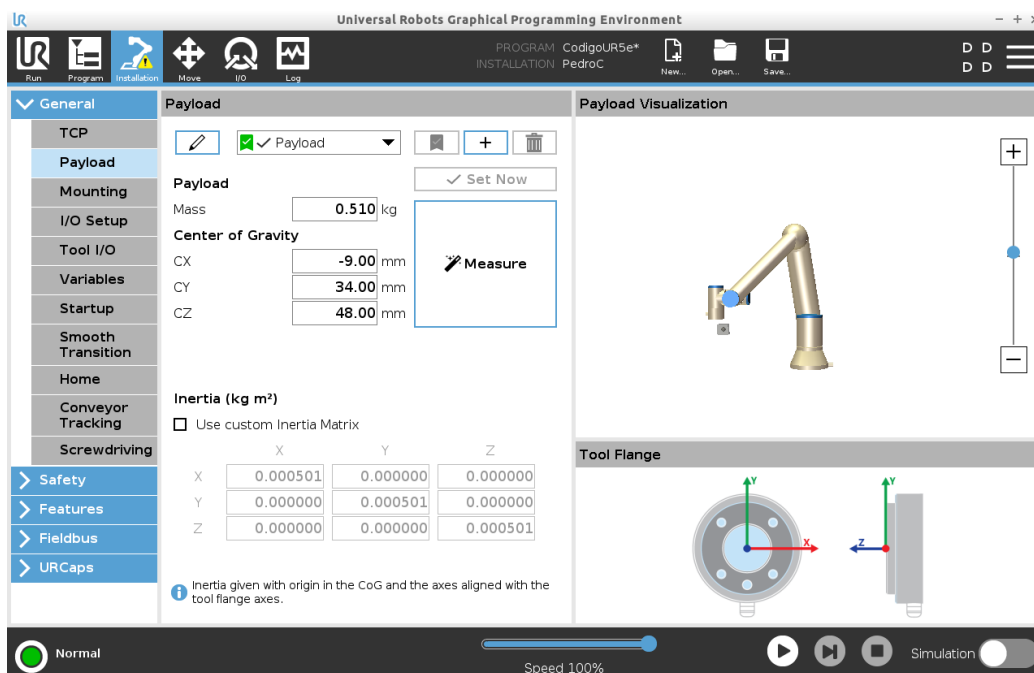


Ilustración 84. Posición del centro de gravedad y obtención del peso del montaje.

No se establecieron planos de seguridad para este proyecto pues no se consideró necesario al tener pocos movimientos y siendo lentos y controlados.

Para poder realizar las operaciones requeridas, se establecieron distintos puntos importantes. Estos se tratan de:

- “Safe_point”: Como su nombre indica, se trata de la posición en la que se coloca el robot cuando no se quiere operar más, escondiendo la punta de la herramienta.
- “P_preparación”: Esta variable se trata del punto de la posición anterior a comenzar las perforaciones, sirviendo como un punto “pre-acción” para garantizar la seguridad.
- “Punto_angulo”: Esta variable la variable usada para comprobar que las coordenadas enviadas a través de la comunicación por socket son precisas y cuadran perfectamente con las zonas delimitadas en la programación de la cámara.
- “P_verificación”: Esta variable se trata del punto en el cual se ha de posicionar el robot para obtener una imagen clara de la cerámica colocada en la base.
- “Centro_ceramica”: Se trata de un plano en cuyo centro se encuentra en el centro de la pieza cerámica situada en la base usada como referencia para los puntos anteriores.

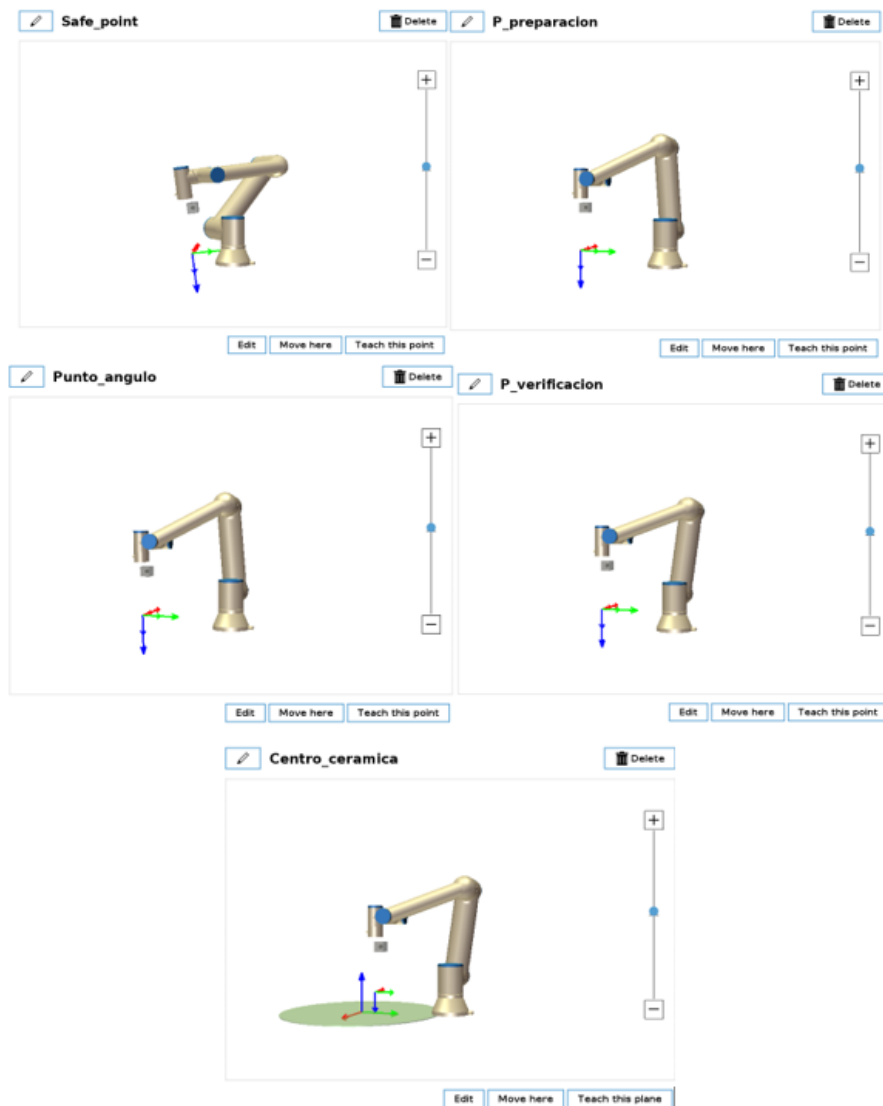


Ilustración 85. Variables y posiciones encontradas en el código.

Una vez obtenido los puntos de interés, se procede a realizar el código. Este utiliza las plantillas propuestas desde Polyscope para facilitar la programación, permitiendo una programación más visual y sencilla.

En primer lugar, se programan las acciones necesarias antes de empezar, inicializando variables, llevando el brazo robot a la posición segura e iniciando la conexión por socket con la Raspberry Pi 5. Una vez se obtiene la verificación de conexión (Raspberry Pi 5 como servidor y UR 5e como cliente), se lleva el robot a la posición de verificación, como se muestra a continuación:

```
1 X Variables Setup
2 ▼ BeforeStart
3 socket_close("socket_raspberry")
4 inicia:=0
5 socket_close("socket_cam")
6 MoveJ
7   Safe_point
8 open:=socket_open("158.42.125.169",3000,"socket_raspberry")
9 Loop open≠ False
10   open:=socket_open("158.42.125.169",3000,"socket_raspberry")
11   sendToServer:="send to server"
12   socket_send_string(sendToServer,"socket_raspberry")
13   MoveJ
14   P_verificacion
```

Ilustración 86. Parte del código que se ejecuta antes de empezar el bucle principal del robot.

A continuación, comienza el programa del robot que se ejecuta en bucle. En primer lugar, se verifica a través de un “if” si la variable “inicia” es igual a 0. Si es el caso, esto indica que es la primera iteración y, por consiguiente, se ha de posicionar el robot en la posición de verificación y esperar unos pocos segundos para que la cámara pueda auto enfocar de forma correcta.

Una vez verificado, se asigna un 0 a una variable llamada “Tot” y se asigna el valor 1 a la variable “inicia”. Tras esto, se realiza una escucha para comprobar si se han enviado datos desde el socket creado anteriormente. Esta línea de código indica varias cosas. En primer lugar, indica el nombre asignado al socket al crear la conexión anteriormente. En segundo lugar y mas importante, se ha de indicar el número de datos que se va a recibir, debido a que las comas separan los bloques de datos que se envían (en este caso, las 6 coordenadas y la variable “Tot” de Raspberry Pi 5). Una vez hecho esto, se crea un bucle en el que mientras el primer valor de “receiveFromServ” sea distinto de 7 se seguirá comprobando hasta obtener los datos. Cabe destacar que a pesar de recibir 7 datos, se crea un vector con 8 datos, pues se le añade en la primera posición el número total de bloques de datos recibidos (por eso se comprueba si “receiveFromServ[0] es distinto de 0”).

En el caso de recibir un vector el cual en su tercera posición tiene un valor unitario, el robot debe ir a la posición segura y parar por completo el programa. Esta es una forma de apagar la conexión asegurando una ejecución limpia del programa. Cabe destacar que las unidades en las que trabaja el robot se tratan de las unidades internacionales, por lo que las coordenadas se encuentran en metros y los giros en radianes (se pueden añadir rotaciones RPY pero en formato base trabaja con radianes). Es debido a esto por lo que se unas este valor, pues es imposible de obtener en coordenadas para perforar al estar en un punto demasiado lejano de la zona de trabajo.

A continuación, se encuentra otro “if” que funciona de la siguiente forma:

- ❖ Verifica si la octava posición (la referente a la variable Tot enviada) es distinta de cero.
 - En caso de ser así, se verifica si es mayor o igual que 0 y se verifica este “if” de forma continua mientras está dentro del bucle.

- En caso de ser afirmativo, se guarda el valor de "Tot" enviado a través del socket en la variable "Tot" creada anteriormente
- Se mueve el brazo robot a la posición de preparación.
- Se verifica si la primera posición del vector leído es igual a 7 y si es el caso se guarda cada posición en una variable distinta, siendo "x", "y", "z", "rx", "ry" y "rz".
- Se guardan estas variables en una nueva variable llamada "Punto_agujero".
- Tras esto se realiza un movimiento lineal a "Punto_agujero" y se realizan dos comandos de fuerza. Uno trata de un movimiento a lo largo del eje Z del TCP (punta de la herramienta) usando una fuerza de 5 N a 2 mm/s durante 6,5 segundos. La otra trata del mismo movimiento, pero en sentido contrario usando la misma fuerza, pero a una velocidad de 32 mm/s durante medio segundo (esto se realiza para que se desplace la misma distancia que en el movimiento de fuerza anterior pero más rápido para mejorar el tiempo de perforación).

Tras esto, se crea otro "if" donde se comprueba si la variable "Tot" del UR es igual a 0. Si es el caso, se verifica si la posición en la que se encuentra el TCP se trata de la posición de verificación y si no es el caso se realiza un movimiento lineal hasta ella. Aquí termina la implementación del código, puesto que el bucle acaba y queda esperando a recibir otra comunicación a través del socket creado.

6. Funcionamiento y análisis de resultados

En el capítulo actual se presenta el montaje final del sistema. Además, se analizan los resultados obtenidos, observando la calidad de los agujeros, el tiempo necesitado, la precisión obtenida tanto al perforar como en los movimientos, etc.

6.1 Configuración final

El proyecto completo queda montado de la siguiente forma:



Ilustración 87. Configuración final del proyecto.

Una de las partes importantes se encuentra en situar correctamente la base en una posición donde el robot puede acceder con la punta de la herramienta programada. Esta posición se puede obtener de distintas formas. La primera consiste en utilizar el guiado usando las manos. UR cuenta con un modo en el que los motores del robot sólo compensan la fuerza de la gravedad. En este modo es posible bloquear planos o rotaciones para tener más precisión en el movimiento. Usando este modo, es posible posicionar el robot guiándolo con la mano en el punto donde se desea tener el centro de la cerámica

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico mediante visión y robótica colaborativa.

con esta precisión. Como los puntos de perforación están programados respecto a este punto, se obtiene una gran flexibilidad para mover la base.

La segunda forma consiste en asignar la posición a través del movimiento del robot usando el modo asignar posición (a través de las flechas y la referencia en el sistema) como se muestra a continuación.

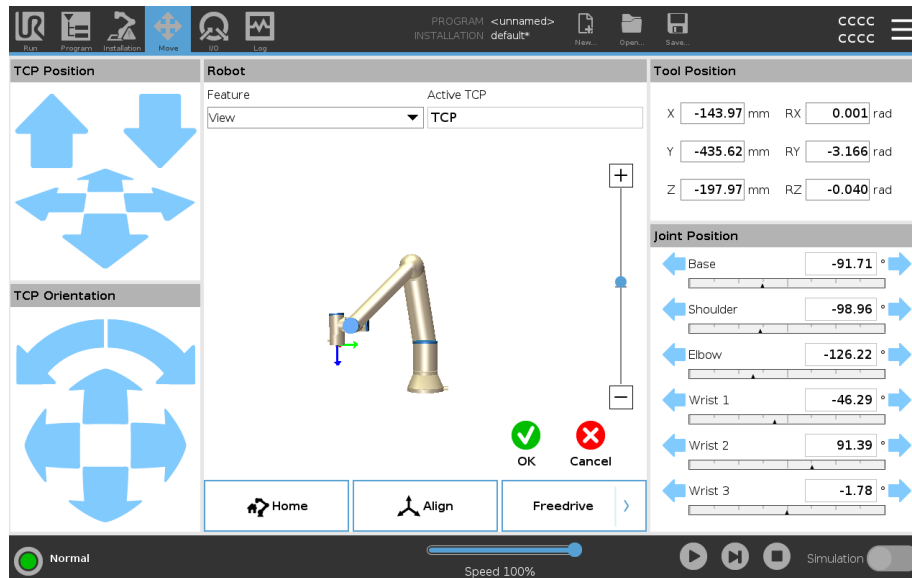


Ilustración 88. Modo modo libre para guiar el robot con las flechas.

Idealmente, lo mejor es usar ambas formas (la primera para posicionar el centro y la segunda para obtener el centro con mayor precisión, ya que se puede medir la ubicación más o menos de forma precisa del centro de la pieza usando una regla.

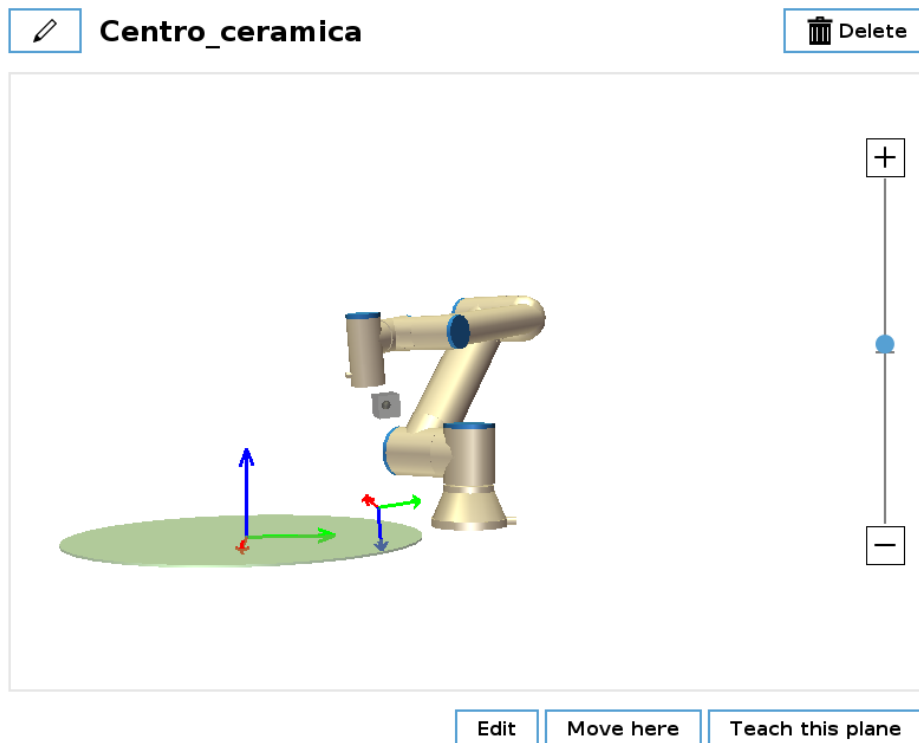


Ilustración 89. Posición del centro del plano respecto la base del robot.

Respecto la posición de la herramienta de perforación, para poder realizar las pruebas es necesario elevar esta herramienta entre 0,5 y 1 metro por encima del plano de perforación debido a la limitación del radio de flexión que puede tolerar el eje extensible de la herramienta. Además, ha de estar posicionada en un lugar cercano donde permita el movimiento libre del robot sin que el cableado interfiera.

Respecto al montaje de las conexiones con los motores y la Raspberry, se puede observar una gran cantidad de cables mezclados. Esto es debido a que los cables usados para realizar las pruebas son de pequeñas dimensiones e imposibilitan el agrupado y ordenado de estos. Se comprobó también que el cable que comunica la Raspberry Pi 5 y la cámara OAK-D Pro no entorpece la visión ni estorba durante los movimientos del brazo robot mostrando un correcto diseño del soporte.

6.2 Análisis de resultados

Durante las primeras pruebas realizadas en el robot UR5 CB, se obtuvieron resultados variados.

Debido al mal funcionamiento del modo fuerza del robot, los resultados obtenidos de las perforaciones no fueron satisfactorios. La primera prueba realizada culminó con una marca en la pieza producida por el contacto de la sujeción de la broca con la superficie a perforar, como se muestra a continuación:

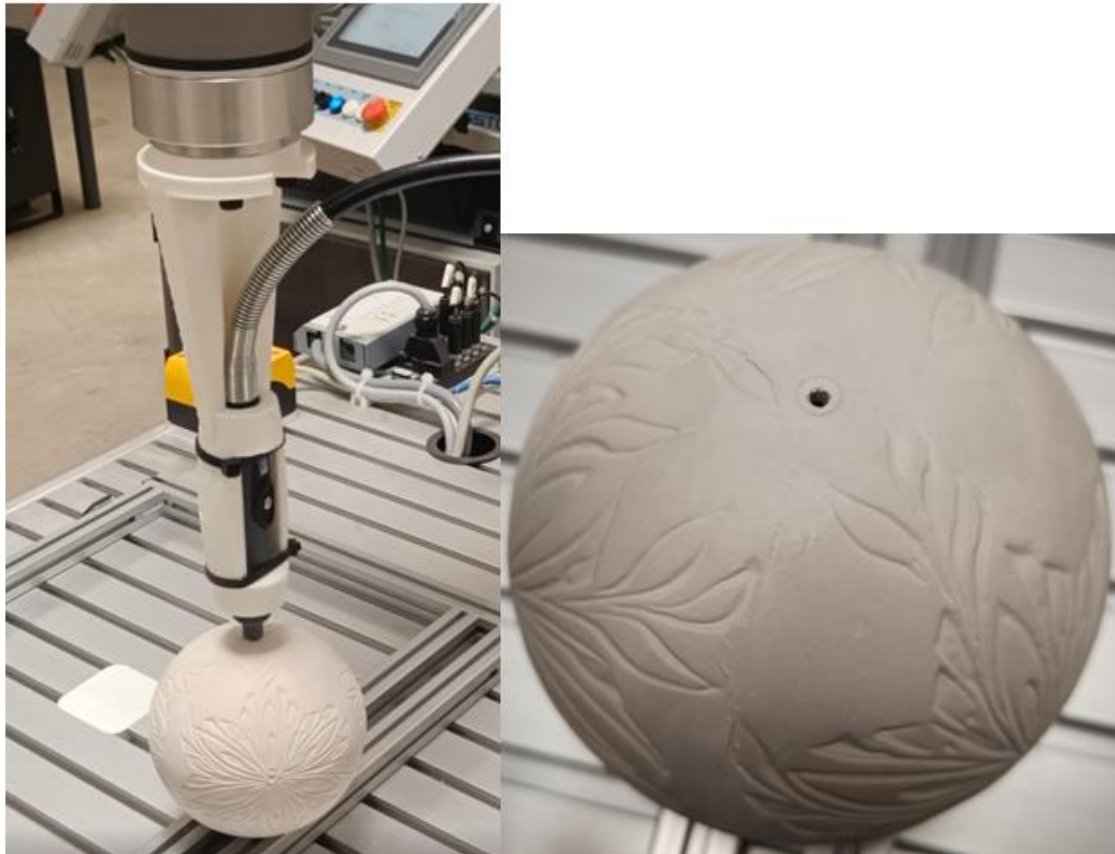


Ilustración 90. Momento donde se produce mal contacto (izquierda) y daños recibidos en la cerámica (derecha).

Durante las siguientes pruebas con el robot UR 5e obtuvieron resultados mucho más satisfactorios, mejorando tras cada cambio. En un primer lugar, tras implementar la espuma de filtro de aire en los rodillos se obtuvieron movimientos más precisos, permitiendo obtener perforaciones de mejor calidad. A pesar de ello, las perforaciones no cumplían el estándar establecido pues se realizaron las perforaciones a demasiada velocidad, dañando los bordes como se muestra:



Ilustración 91. Cerámica con escamados cerca de la zona perforada

En la iteración final, se obtuvieron resultados satisfactorios respecto las perforaciones. En el apartado de calidad, se obtuvieron buenos resultados evitando lo mínimo posible realizar perforaciones con el borde dañado como se muestra a continuación:



Ilustración 92. Buena calidad de perforaciones conseguidas a poca distancia.

Cabe destacar que en el acabado de las perforaciones influyen tanto los rpm de la herramienta como el desgaste de esta, obteniendo resultados notablemente mejores al usar una broca nueva o con el filo de la punta recuperado.

Otro punto donde el proyecto fue un éxito es en el de tiempo de perforación. Gracias al ajuste realizado durante las pruebas para obtener el mejor equilibrio entre fuerza y velocidad de la herramienta se obtuvo una solución que no compromete calidad con cantidad.

En general, se necesitan unos 8,5 segundos para realizar cada perforación, pues se incluye el tiempo desde que la herramienta está situada en la posición de preparación hasta que termina el agujero y vuelve a ella de nuevo. Emplear una velocidad de avance de la pieza mayor a los 2 mm/s crea perforaciones inconsistentes, pues la cantidad de agujeros con el borde dañado aumenta a pesar de usar el modo de mayor rpm de la herramienta.

Se puede mejorar más la calidad de perforación para no producir apenas daños en los bordes de los agujeros, pero se emplea el doble de tiempo siendo esto un problema puesto uno de los objetivos del proyecto es obtener una solución que permita realizar las perforaciones en el menor tiempo posible.

En el apartado de detección de los puntos donde no se debe agujerear se han obtenido resultados mixtos. Debido a la lógica creada durante el desarrollo del código que decide en que posiciones y localizaciones se debe perforar, se emplean un par de segundos parado para dar tiempo a estabilizar la imagen y detectar las nuevas zonas a perforar. A pesar de que la detección de bordes funciona y se puede considerar un éxito el sistema de selección, es necesario pulir algunos aspectos. Una parte en la que no se consiguió alcanzar el objetivo como se esperaba fue cuando la pieza se encuentra en tal posición que el cuadrado donde se verifica si se puede perforar está ubicado en la zona con el tallado. Esto provoca que el sistema quede parado un segundo y después gire la pieza en el sentido positivo del eje X (el rojo grande de la siguiente figura):

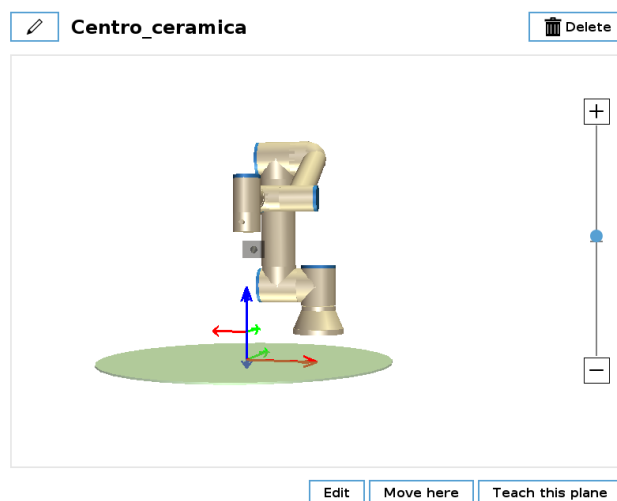


Ilustración 93. Observación del plano del centro de la cerámica estando de frente al robot.

Una vez se cumplen las condiciones ya mencionadas durante el apartado de la programación, el sistema deja de rotar la cerámica. El problema reside en que este método de “desbloqueo” de la pieza provoca que la perforación completa de la cerámica tarde una cantidad de tiempo aleatoria.

También cabe destacar que en el momento de analizar las imágenes obtenidas a través de la cámara OAK-D Pro es necesario apagar la Dremel, puesto que la rotación del eje extensible provoca vibraciones que interfieren en la obtención correcta de imágenes, no pudiendo leer con examinar

6.3 Conclusiones del análisis

El análisis del montaje demuestra que el proyecto es viable en distintos apartados. Gracias a la versatilidad del robot es posible implantar este sistema en cualquier lugar una vez definido el centro de la cerámica. También, debido a que el robot es el componente más caro de todo el proyecto y la base resulta fácil de crear usando impresión 3D, es posible añadir múltiples bases en la que un solo robot opere, permitiendo amortizar de una mejor forma la inversión requerida de añadir un robot a la industria.

Cabe destacar que debido al sistema elegido se ha conseguido implementar una solución que permite sustituir componentes de forma rápida y sienta de un coste reducido.

En resumen, el proyecto valida la utilidad de usar un robot para esta aplicación sirviendo como base para proyectos similares en los que se requiera un sistema de visión para perforar esferas o figuras similares.

Capítulo 7

7. Conclusión

Debido al constante cambio y evolución que actualmente se presenta en muchos aspectos de la industria, la necesidad de colaboración entre humano-robot y de automatizar procesos para hacerlos más competitivos ha provocado demanda de proyectos para estudiar si ciertas propuestas son viables. Este trabajo se ha desarrollado centrándose en el diseño de uno de estos proyectos, aprovechando las capacidades de comunicación de un robot UR 5e y la versatilidad de dicho robot junto plataformas de informatización de bajo coste y nuevas tecnologías con IA en visión disponibles para obtener una solución que presente ventajas a una parte de la industria.

El desarrollo de la implantación técnica del sistema ha sido acompañado de una investigación I+D evaluando tecnologías disponibles y distintos enfoques para crear una solución viable al proyecto planteado cumpliendo unos objetivos. Esto ha permitido predecir, evaluar y abordar distintos desafíos que han surgido al implementar ideas para obtener soluciones que cumplieran con las características deseadas.

7.1 Objetivos logrados

En el desarrollo del sistema se ha buscado implementar una solución que abarque la mayor cantidad de objetivos planteados durante las fases iniciales del proyecto. Los objetivos que se han alcanzado son:

- Comportamiento preciso: Se puede asegurar que la mayoría de los agujeros se van a realizar en las zonas deseadas sin crear agujeros dobles que fracturan ni dañen la pieza debido a su proximidad.
- Repetibilidad: Debido a la programación realizada, se ha obtenido cierto grado de repetibilidad que no dependen de la pieza pudiendo crear patrones de agujeros.
- Velocidad de perforación adecuada: Las velocidades de perforación y de actuación del robot está acorde a lo buscado.

- Obtener perforaciones de calidad: Las perforaciones obtenidas cumplen con las especificaciones deseadas dentro de un margen de tiempo razonable para el tipo de operación deseada.
- Modificabilidad y reusabilidad: La facilidad de intercambiar componentes o de usar múltiples veces el mismo montaje permite que el proyecto sea reutilizable además de poder modificar las partes según surjan necesidades nuevas.

7.2 Líneas de mejora

A pesar de haber cumplido la gran parte de los objetivos establecidos, es necesario que un proyecto vaya evolucionando pues durante el desarrollo se detectan áreas que presentan oportunidades de mejora significativas que ayudan a obtener una solución más eficiente y eficaz.

En el caso de este proyecto, unas de las mejoras que se deben implementar es el cambio de la herramienta que se utiliza para perforar. Debido a la geometría del eje extensible de la Dremel, el TCP de esta se encuentra bastante alejado de la muñeca y esto no es deseable para evitar vibraciones y pérdida de precisión. De forma ideal, crear una herramienta con una posición de TCP más cercana y centrada para así poder modificar la distancia la base del robot e incluso poder usar un modelo más económico que cumpla las necesidades es lo más deseable. Además, esto produce que el sistema sea más modular, pues es más sencillo de implantar y de sustituir en caso de mal funcionamiento.

Otro aspecto que mejorar está relacionado con el proceso de agujereado. Durante este se produce una gran cantidad de residuo en forma de polvo que se deposita en las zonas contiguas del agujero en proceso, creando problemas pues la cámara no detecta correctamente si las zonas cercanas tienen bordes o no (detectando en este caso que existe un borde y por lo tanto no va a perforar). Una forma de solucionar esto constaría de un pequeño tubo de plástico (ya sea conectado cerca del TCP o en la propia base) para poder usar aire a presión y así limpiar el residuo que se acumula.

Un aspecto crítico que se debe mejorar es la programación del movimiento y detección de puntos a perforar. Aprovechando al máximo las capacidades de una OAK-D Pro, es posible generar una IA y entrenarla para que detecte en qué puntos ha de perforar y donde no. A través de este entrenamiento, es posible obtener movimientos más precisos pues la cámara puede detectar en qué dirección rotar la pieza cerámica para continuar el camino y verificar que se ha terminado de perforar la pieza.

7.3 Reflexión final

Al concluir este proyecto de investigación y desarrollo y realizar la presentación del sistema obtenido, es necesario reflexionar sobre el camino y decisiones que se han tomado. El proyecto ha sido un desafío en el que se ha aprendido varias cosas importantes.

En primer lugar, la rápida evolución de la tecnología y su adaptabilidad son notables durante el desarrollo del proyecto, puesto que conforme se iba avanzando en este, se tenía más claro que la colaboración entre humano y robot es inevitable para obtener un aumento de la producción.

También se muestra que, en caso de no tener una solución prefabricada o implementada, a través de la tecnología es posible investigar y crear una solución que se adapte a las necesidades tanto humanas como económicas de las empresas interesadas. Esto queda demostrado pues durante el desarrollo del proyecto ha surgido la necesidad de adaptarse a las circunstancias y de modificar en varias ocasiones la dirección tomada en el desarrollo.

Por lo tanto, ha quedado demostrado que la tecnología es una de las herramientas más poderosas y necesarias para la mejora continua de la sociedad y su bienestar, incentivando a futuras generaciones a continuar, mejorar e innovar proyectos que permitan un desarrollo mayor y mejor de las ventajas en la sociedad que se poseen en la actualidad.

Bibliografía

- (2020). En H. Fairhead, & M. James, *Raspberry Pi IoT In Python Using GPIO Zero* (Segunda ed., pág. 260). I/O Press.
- a.chamarthy. (Septiembre de 2023). *Control UR5e with Python/ Execute a .script file with Python*. Obtenido de <https://forum.universal-robots.com/t/control-ur5e-with-python-execute-a-script-file-with-python/30523/2>
- Chobey, A. (2018). Obtenido de Stack Overflow: <https://stackoverflow.com/questions/49236146/python-cv2-edge-and-contour-detection>
- Docs.Python. (2024). Obtenido de <https://docs.python.org/3/library/threading.html>
- EDS robotics. (2024). Obtenido de <https://www.edsrobotics.com/blog/pilares-industria-4-0/>
- Electrogeek. (2024). Obtenido de <https://www.electrogeekshop.com/como-controlar-un-motor-de-pasos-con-el-conductor-drv8825-y-arduino/>
- Gpiozero. (2024). Obtenido de https://gpiozero.readthedocs.io/en/stable/migrating_from_rpigpio.html
- Luxonics. (2024). Obtenido de <https://docs.luxonis.com/hardware/platform/deploy/to-rpi>
- OpenCV. (2024). Obtenido de https://docs.opencv.org/4.x/dc/da5/tutorial_py_drawing_functions.html
- OpenCV-Python. (2024). Obtenido de <https://cjjouanne.github.io/OpenCV-Python/>
- Protosupplies. (2024). Obtenido de <https://protosupplies.com/product/drv8825-high-current-stepper-motor-driver/>
- Qahwaji, R., & Green, R. (8 de Enero de 2001). *Detection of Closed Regions in Digital Images*. Obtenido de https://www.researchgate.net/publication/220085316_Detection_of_Closed_Regions_in_Digital_Images
- RaspberryTips*. (2024). Obtenido de <https://raspberrytips.com/raspberry-pi-history/>
- Universal Robots. (2024). Obtenido de <https://www.universal-robots.com/articles/ur/programming/urscript-dynamic-force-control/>

Universal Robots. (2024). Obtenido de <https://www.universal-robots.com/es/blog/comunicacion-inteligente-robots/>

Universal Robots. (2024). Obtenido de <https://www.universal-robots.com/articles/ur/interface-communication/tcpip-socket-communication-via-urscript/>

Universal Robots. (2024). *Remote Control Via TCP/IP*. Obtenido de <https://www.universal-robots.com/articles/ur/interface-communication/remote-control-via-tcpip/>

Universal Robots. (s.f.). *ClientInterfaces_Primary*. Obtenido de https://s3-eu-west-1.amazonaws.com/ur-support-site/16496/ClientInterfaces_Primary.pdf

Vilros. (s.f.). Obtenido de <https://vilros.com/pages/raspberry-pi-5-pinout>

Presupuesto general del proyecto

Resumen del presupuesto

Tabla 4. Presupuesto Básico

Recurso	Tipo	Tipo de presupuesto	Trabajo	Tasa estándar	
Presupuesto Mano de Obra			528 horas	18.480,00 €	
Ingeniero	Trabajo	Mano de obra	528 horas	18.480,00 €	20,00 €/hora
Presupuesto Equipamiento			668 horas	925,25 €	
Portátil Lenovo	Material	Equipamiento	256 horas	192,00 €	60 €/ms
Robot UR 5	Material	Equipamiento	116 horas	398,75 €	275,00 €/ms
Raspberry Pi 5	Material	Equipamiento	168 horas	73,50 €	35 €/ms
OAK_D Pro	Material	Equipamiento	128 horas	96,00 €	60,00 €/ms
Impresora 3D	Material	Equipamiento		45,00 €	35,00 €
Dremel	Material	Equipamiento		120 €	120 €
Presupuesto total			1196 horas	925,25 €	

Presupuesto detallado

Tabla 5. Presupuesto detallado.

Tarea	Trabajo	Duración	Costo
1. Organización	104 horas	13 días	1.859,00 €
<i>División de partes</i>	32 horas		572,00 €
<i>Documentación programas modelado</i>	24 horas		429,00 €
<i>Documentación UR</i>	48 horas		858,00 €
2. Diseño soporte Dremel	56 horas	7 días	1.001,00 €

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico mediante visión y robótica colaborativa.

<i>Diseño de 1ª Versión</i>	40 horas		715,00 €
<i>Diseño de 2ª Versión</i>	16 horas		286,00 €
3. Diseño Base Cerámica	396 horas	60 días	8.586,50 €
<i>Diseño de 1ª versión</i>	64 horas		1.144,00 €
<i>1ª Iteración</i>	40 horas		715,00 €
<i>2ª Iteración</i>	24 horas		429,00 €
<i>Diseño de 2ª versión</i>	156 horas		3.170,50 €
<i>1ª Iteración</i>	56 horas		1.001,00 €
<i>Diseño de apoyos</i>	32 horas		572,00 €
<i>Diseño de parte superior</i>	24 horas		429,00 €
<i>2ª Iteración</i>	100 horas		2.169,50 €
<i>Diseño de apoyos</i>	24 horas		429,00 €
<i>Diseño de parte superior</i>	16 horas		286,00 €
<i>Impresión de partes</i>	16 horas		301,00 €
<i>Mecanizado de piezas</i>	12 horas		420,00 €
<i>Diseño de casquillos</i>	8 horas		143,00 €
<i>Montaje de rodillos</i>	8 horas		280,00 €
<i>Montaje de base</i>	4 horas		140,00 €
<i>Pruebas con base</i>	12 horas		170,50 €
<i>Diseño final</i>	176 horas		4.272,00 €
<i>Diseño de apoyos</i>	40 horas		715,00 €
<i>Diseño de parte superior</i>	24 horas		429,00 €
<i>Impresión de partes</i>	8 horas		295,00 €
<i>Diseño 1º casquillo brida</i>	16 horas		286,00 €
<i>Diseño 2º casquillo brida</i>	8 horas		143,00 €
<i>Impresión de partes</i>	12 horas		435,00 €
<i>Mecanizado de piezas</i>	16 horas		560,00 €
<i>Construcción de rodillos</i>	8 horas		280,00 €
<i>Diseño soporte motores e impresión</i>	24 horas		429,00 €
<i>Montaje motores</i>	12 horas		420,00 €
<i>Recubrimiento rodillos</i>	8 horas		280,00 €
4. Código Raspberry Pi 5	376 horas	34 días	5.147,50 €
<i>Montaje y calibrado controladores</i>	16 horas		560,00 €
<i>Código con pruebas de motores</i>	80 horas		1.417,50 €
<i>Código prueba OAK-D Pro</i>	120 horas		1.447,50 €
<i>Desarrollo código control</i>	160 horas		1.722,50 €
5. Código Robot UR 5	72 horas	6 días	1.023,00 €
<i>Desarrollo código UR</i>	72 horas		1.023,00 €
6. Test y análisis	192 horas	12 días	2.187,00 €
<i>Depuración código y pruebas finales</i>	192 horas		2.187,00 €

Detalle de los recursos

- Renting Ordenador portátil Lenovo

Modelo: Lenovo ThinkPad E15 RAM 16 GB SDD 256 GB

Precio medio: 60 € al mes

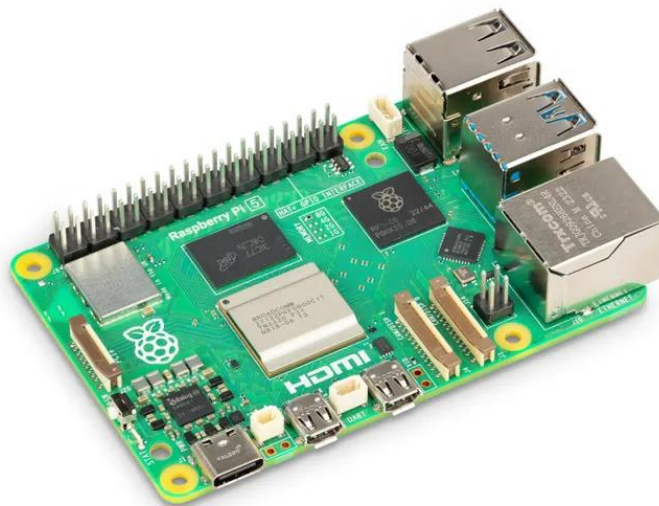


Portátil Lenovo ThinkPad E15. Fuente: Lenovo.

- Compra Sistema de computación

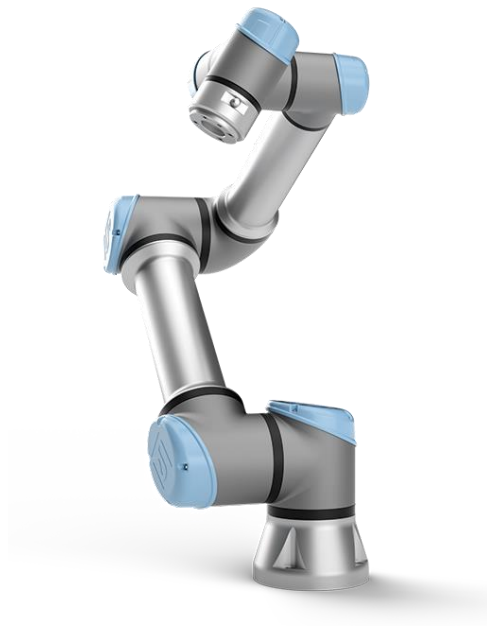
Modelo: Raspberry Pi 5

Precio medio: 120 €



Sistema Raspberry Pi 5. Fuente: Raspberry Pi.

- Renting Robot UR5
Modelo: UR5e
Ubicación: Instituto de Automática e Informática Industrial
Precio medio: 550 al mes



Robot colaborativo UR5e. Fuente: Universal Robots.

- Renting camara de visión
Modelo: OAK_D Pro
Ubicación: Instituto de Automática e Informática Industrial
Precio medio: 60 al mes



OAKD-Pro. Fuente: Luxonix.

- Compra Dremel 3000
Modelo: Dremel 3000
Precio medio: 120 €



Dremel 3000. Fuente: Amazon.es

- Impresoras 3D
Precio medio: 15 euros por bobina



Filamento para impresión 3D. Fuente: Amazon.es

Apéndice A

Planificación del proyecto

A.1 Planificación de tareas del proyecto

Para desarrollar con éxito el proyecto, se estructuró de forma que se pudieran planificar tareas más sencillas. Para ello, se ha utilizado software de Dirección de proyectos, MS Project. En la tabla B.2 se encuentra la planificación detallada del proyecto:

Tabla A6. Cronograma seguido.

Tabla 7. Cronograma detallado.

Tarea	Trabajo	Duración	Comienzo	Fin
1. Organización	104 horas	13 días	lun 29/01/24	jue 15/02/24
<i>División de partes</i>	32 horas	4 días	lun 29/01/24	vie 02/02/24
<i>Documentación programas modelado</i>	24 horas	3 días	vie 02/02/24	mié 07/02/24
<i>Documentación UR</i>	48 horas	6 días	mié 07/02/24	jue 15/02/24
2. Diseño soporte Dremel	56 horas	7 días	jue 15/02/24	lun 26/02/24
<i>Diseño de 1ª Versión</i>	40 horas	5 días	jue 15/02/24	jue 22/02/24
<i>Diseño de 2ª Versión</i>	16 horas	2 días	jue 22/02/24	lun 26/02/24
3. Diseño Base Cerámica	396 horas	60 días	lun 26/02/24	lun 20/05/24
<i>Diseño de 1ª versión</i>	64 horas	8 días	lun 26/02/24	jue 07/03/24
<i>1ª Iteración</i>	40 horas	5 días	lun 26/02/24	lun 04/03/24
<i>2ª Iteración</i>	24 horas	3 días	lun 04/03/24	jue 07/03/24
<i>Diseño de 2ª versión</i>	156 horas	22 días	jue 07/03/24	lun 08/04/24
<i>1ª Iteración</i>	56 horas	7 días	jue 07/03/24	lun 18/03/24
<i>Diseño de apoyos</i>	32 horas	4 días	jue 07/03/24	mié 13/03/24
<i>Diseño de parte superior</i>	24 horas	3 días	mié 13/03/24	lun 18/03/24
<i>2ª Iteración</i>	100 horas	15 días	lun 18/03/24	lun 08/04/24
<i>Diseño de apoyos</i>	24 horas	3 días	lun 18/03/24	jue 21/03/24
<i>Diseño de parte superior</i>	16 horas	2 días	jue 21/03/24	lun 25/03/24
<i>Impresión de partes</i>	16 horas	2 días	lun 25/03/24	mié 27/03/24
<i>Mecanizado de piezas</i>	12 horas	3 días	mié 27/03/24	lun 01/04/24
<i>Diseño de casquillos</i>	8 horas	1 día	lun 01/04/24	mar 02/04/24
<i>Montaje de rodillos</i>	8 horas	2 días	mar 02/04/24	jue 04/04/24
<i>Montaje de base</i>	4 horas	1 día	jue 04/04/24	vie 05/04/24
<i>Pruebas con base</i>	12 horas	1 día	vie 05/04/24	lun 08/04/24
<i>Diseño final</i>	176 horas	30 días	lun 08/04/24	lun 20/05/24
<i>Diseño de apoyos</i>	40 horas	5 días	lun 08/04/24	lun 15/04/24
<i>Diseño de parte superior</i>	24 horas	3 días	lun 15/04/24	jue 18/04/24

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico mediante visión y robótica colaborativa.

<i>Impresión de partes</i>	8 horas	2 días	jue 18/04/24	lun 22/04/24
<i>Diseño 1º casquillo brida</i>	16 horas	2 días	lun 22/04/24	mié 24/04/24
<i>Diseño 2º casquillo brida</i>	8 horas	1 día	mié 24/04/24	jue 25/04/24
<i>Impresión de partes</i>	12 horas	3 días	jue 25/04/24	mar 30/04/24
<i>Mecanizado de piezas</i>	16 horas	4 días	mar 30/04/24	lun 06/05/24
<i>Construcción de rodillos</i>	8 horas	2 días	lun 06/05/24	mié 08/05/24
4. Diseño soporte motores e impresión	24 horas	3 días	mié 08/05/24	lun 13/05/24
<i>Montaje motores</i>	12 horas	3 días	lun 13/05/24	jue 16/05/24
<i>Recubrimiento rodillos</i>	8 horas	2 días	jue 16/05/24	lun 20/05/24
5. Código Raspberry Pi 5	376 horas	34 días	lun 20/05/24	vie 05/07/24
<i>Montaje y calibrado controladores</i>	16 horas	4 días	lun 20/05/24	vie 24/05/24
<i>Código con pruebas de motores</i>	80 horas	10 días	vie 24/05/24	vie 07/06/24
<i>Código prueba OAK-D Pro</i>	120 horas	10 días	vie 07/06/24	vie 21/06/24
<i>Desarrollo código control</i>	160 horas	10 días	vie 21/06/24	vie 05/07/24
6. Código Robot UR 5	72 horas	6 días	vie 05/07/24	lun 15/07/24
<i>Desarrollo código UR</i>	72 horas	6 días	vie 05/07/24	lun 15/07/24
7. Test y análisis	192 horas	12 días	lun 15/07/24	mié 31/07/24
<i>Depuración código y pruebas finales</i>	192 horas	12 días	lun 15/07/24	mié 31/07/24

En la próxima figura se presenta un diagrama de Gantt. Este sirve para planificar proyectos y contiene cierta información:

- La duración de real de un proceso se muestra como una línea azul.
- La duración planificada se obtiene con un color azul claro.
- Se tienen unas barras grises que actúan como resumen de un conjunto de tareas.
- Un porcentaje de 100% indica que se ha completado por completo una tarea.

Como se observa en la siguiente figura lo planeado y lo sucedido concuerdan pues sólo se observa un color azul claro. Además, las tareas en negrita se tratan de los resúmenes, aunque no se aprecie en el color de las barras.

Diseño y desarrollo de un sistema automático de perforación en el sector cerámico mediante visión y robótica colaborativa.

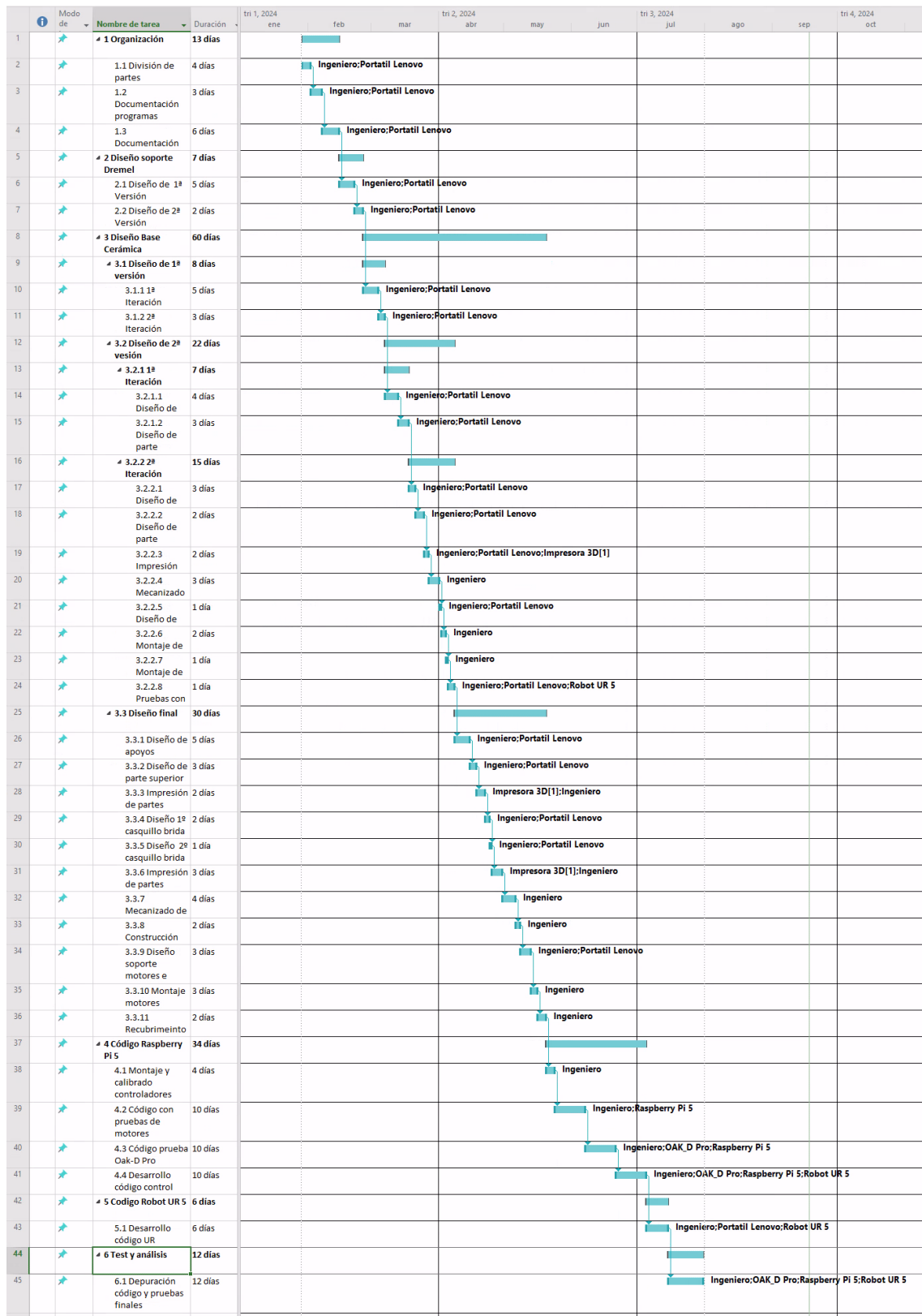


Ilustración A94. Diagrama de Gantt.