ENTREGABLE
PROYECTOS—
2023

DESARROLLO DE UN SISTEMA DE GENERACIÓN DE TRAYECTORIAS COMPLEJAS EN ROBOTS BASADO EN APRENDIZAJE POR DEMOSTRACIÓN "ROBOTRACK"

Entregable: E6.1- Informe de pruebas realizadas

Número de proyecto: 22200073
Expediente: IMDEEA/2022/9
Duración: Del 01/07/2022 al 30/09/2023
Coordinado en AIDIMME por: JOSÉ LUÍS SÁNCHEZ

AIDIMME
INSTITUTO TECNOLÓGICO

ÍNDICE

ÍNDICE 1

A. OBJETIVO DEL ENTREGABLE 1

B. ACTIVIDADES REALIZADAS 2

1. DISEÑO DEL DEMOSTRADOR 2

1.1. DEFINICIÓN DE OBJETIVOS 2

1.2. DESCRIPCIÓN DE PROCESOS POTENCIALES A UTILIZAR EN EL DEMOSTRADOR 4

CASO ENVASADO PINTURAS (PROPUESTO POR EMPRESA COLABORADORA BLATEM) 4
CASO ENVASADO ALIMENTACIÓN (PROPUESTO POR EMPRESA COLABORADORA ALNUT) 6
CASO PROCESO DE LIJADO DE PIEZAS 11

1.3. SELECCIÓN DEL PROCESO DE DEMOSTRACIÓN 12

1.4. KPIS Y MÉTODO DE LA VALIDACIÓN 12

2. MONTAJE DEL DEMOSTRADOR Y PRUEBAS 14

2.1. DEMOSTRADOR EN FASE 1 14

MONTAJE DEMOSTRADOR EN FASE 1 14
EJECUCIÓN DE LAS PRUEBAS 16
RESULTADOS FASE 1 DE LA VALIDACIÓN 29

2.2. DEMOSTRADOR EN LA FASE 2 35

MONTAJE DEMOSTRADOR EN LA FASE 2 35
EJECUCIÓN DE PRUEBAS EN LA FASE 2 39
RESULTADOS FASE 2 DE LA VALIDACIÓN 55

3. VALIDACIÓN DEL SISTEMA 58

3.1. ANÁLISIS 58

3.2. PROPUESTA DE MEJORAS 59

C. RESUMEN Y CONCLUSIONES 62

A. OBJETIVO DEL ENTREGABLE

Este entregable recoge las actividades realizadas durante la ejecución del **PT6 - Desarrollo de un demostrador**

En este paquete de trabajo se realizaron pruebas piloto de validación principalmente en las instalaciones de AIDIMME. Se implementaron los sistemas desarrollados en entornos de fabricación.
Mediante estas demostraciones se realizó la validación de los sistemas y se realizaron las modificaciones necesarias para alcanzar los objetivos previstos

En el entregable se presentan las pruebas de validación llevadas a cabo sobre la aplicación Robotrack, mediante el piloto demostrativo desarrollado para tal fin. Se presentan diferentes procesos industriales analizados como posible proceso piloto y se justifica la decisión de adoptar el proceso de lijado como piloto para validar los desarrollos realizados. Se monta la infraestructura necesaria para simular un proceso de lijado de piezas planas, y se llevan a cabo dos rondas de validación con el objetivo de hacer aflorar posibles mejoras a implementar sobre el sistema.

B. ACTIVIDADES REALIZADAS

1. DISEÑO DEL DEMOSTRADOR

1.1. Definición de objetivos

Tal y como aparece reflejado en la memoria de solicitud del presente proyecto, el objetivo general es el desarrollo de un sistema de aprendizaje por demostración para robots colaborativos, basado en sistemas de visión que permita al robot reproducir tareas complejas de forma rápida tras la demostración humana.

Este objetivo general, tras la ejecución de los paquetes de trabajo PT4 y PT5, que han dado lugar al diseño y desarrollo de la interfaz HMI que gobierna los sistemas de captura de datos y permite generar el programa final del robot mediante una demostración humana, se ha ido traduciendo en una serie de funcionalidades de la solución final.

Se han identificado una serie de funcionalidades que potencialmente pueden ser demandas por un usuario de la solución. Durante la validación llevada a cabo en la demostración se pretende validar por un lado la importancia y por otro el grado de consecución de estas funcionalidades.

A continuación, se resumen estas funcionalidades, organizadas según las fases del flujo de trabajo en la HMI de Robotrack.

1.- Fase de demostración

- Asistencia al usuario no experto en el montaje del sistema de captura de datos
- Edición y gestión de diferentes sistemas de captura de datos del proceso
- Edición y ajuste de los parámetros de configuración de los sistemas de captura de datos
- Autocomprobación de la conexión y buen funcionamiento de los sistemas de captura de datos disponibles
- Ayuda con posibles soluciones tras la detección de un mal funcionamiento del sistema de captura de datos
- Permitir la grabación de datos por parte de 1 solo usuario (no depender de la ayuda de un segundo usuario)
- Verificación al usuario de que los datos han sido grabados y están disponibles en el sistema
- Visualización inicial de los datos grabados en bruto

2.- Fase de generación de trayectorias.

- Asistencia al usuario no experto en el proceso de transformación de los datos en bruto
- Identificación de datos incoherentes o insuficientes

- Aplicación de técnicas de filtrado, reducción, suavizado, etc.
- Generación de un archivo de trayectorias con los puntos y acciones clave del programa en un lenguaje natural de usuario (tipo flujograma, etc.) que le permita entender a alto nivel el programa de robot que va a ser generado

3.- Fase de generación de programa de Robot

- Generación de un archivo de programa de robot válido para 1 robot
- Generación de un archivo de programa de robot válido para diferentes modelos y marcas de robot

4.- Fase de simulación

- Poder realizar una simulación del programa de robot generado
- Utilizar la simulación para modificar velocidades de trabajo y establecer tiempos de ciclo

5.- Fase de ejecución del programa

- Transferir el programa al robot
- Ejecutar el programa de robot
- Realizar paradas durante la ejecución
- Realizar modificaciones en el programa de robot tras observar el resultado de la ejecución
- Guardado del programa de robot

Estas funcionalidades y hacen referencia a las prestaciones de la HMI Robotrack. Para evaluar el resultado final de la solución se han definido una serie de indicadores de resultados a contrastar mediante la fase de demostración.

- Conseguir un programa de robot válido a partir de los datos capturados en la fase de demostración.
- Conseguir, a partir de una única demostración un programa válido para diferentes modelos y marcas de robots.
- Precisión del programa de robot para lograr un producto con una calidad igual a la obtenida por el operario trabajando en modo manual
- Obtener un programa de robot cuyo tiempo de ciclo reduzca el de la actividad manual original
- Tiempo empleado en la generación del programa
- Facilidad con la que usuario consigue completar todo el flujo de trabajo (desde la demostración hasta la ejecución final del programa)
- Ausencia de errores durante el proceso

1.2. Descripción de procesos potenciales a utilizar en el demostrador

Se describen a continuación tres procesos industriales que han sido tomados como referencia para analizar su viabilidad para llevar a cabo su automatización por robots colaborativos, mediante un proceso de demostración humana.

Los dos primeros son procesos llevados a cabo actualmente en las empresas colaboradoras Blatem y Alnut respectivamente, y el tercero (proceso de lijado de pizas planas) ha surgido en reuniones y discusiones particulares con el resto de las empresas colaboradoras y el equipo de trabajo de AIDIMME.

Caso envasado pinturas (propuesto por empresa colaboradora Blatem)

Descripción del proceso.

La tarea realizada por el operario consiste en coger un cubo de una pila con la mano izquierda. Moverlo ligeramente para "desatascarlo" y liberar el cubo de arriba. El cubo se coloca cobre la línea y se espera al llenado de este. Con la mano derecha se coge una tapa y se coloca sobre el cubo. En un proceso posterior se realiza el cierre de cubo y tapa.

Figura 1 – Esquema del proceso de envasado de pinturas
Fuente: elaboración propia

Viabilidad de automatización mediante robot

Desde el punto de vista de automatización de la tarea con un robot, sería posible realizarla modificando la posición de las tapas en la pila, poniéndolas boca abajo, lo que permitiría cogerlas con un "gripper" (herramienta de agarre) por el centro de esta.

Además, resultaría de utilidad un alimentador de tapas para siempre cogerla a la misma altura y posición mientras que estas se alimentan desde una pila de mayor cantidad. Este alimentador pasaría a ser repuesto por el operario, lo que permitiría liberar de la tarea de carga de la línea.

Por otro lado, se debería modificar la pila fija de cubos por algún sistema elevador de la pila que permita que el cubo superior siempre se encuentre en la misma posición (altura de trabajo) . En este caso el agarre del cubo superior se realizaría mediante herramienta por vacío por las caras interiores del mismo. Esto permitiría que la herramienta fuera la misma para ambas tareas de la línea.

Para facilitar el agarre y salida del cubo de la pila sería de interés algún tipo de sujeción del cubo que se encuentra a continuación, ya que se ha observado que en los cubos inferiores se genera una pequeña fuerza por vacío. (no mayor a 20 N. = 2 kg).

Figura 2 – Ejemplo de cubos y tapas a manipular
Fuente: elaboración propia

En este caso el operario, al igual que ocurre con las tapas, solo tendría que alimentar la pila de cubos. Para ambas pilas, se puede recurrir a un sistema de sensor y baliza para que cuando queden pocas unidades, se genere un aviso al operario. Entendiendo que este puede dedicarse a supervisar varias líneas o realizar varias operaciones.

Idoneidad para la generación de un programa mediante demostración humana

El proceso de llenado y precintado de los cubos de 15 kilos puede ser automatizado mediante la integración de un robot colaborativo, tal y como se ha descrito en el apartado anterior. Pero las características de dicho proceso no hacen posible que esta automatización pueda realizarse mediante el aprendizaje por demostración.

En este caso los movimientos realizados por el operario tanto con su brazo derecho como el izquierdo resultan viables para ser capturada la trayectoria de estos. Sin embargo, los gestos realizados con sus manos para sacar los cubos de la pila por un lado y coger la tapa por otro lado, así como voltear esta última, no es posible de capturar.

Ya que estas acciones se realizan en un punto externo a ambos componentes, los gestos de agarre realizados se llevan a cabo en un borde de la pieza y son posibles gracias a la capacidad de movimiento de las articulaciones humanas. A esto hay que sumar la imposibilidad de realizar un agarre similar mediante herramientas en el lugar que se

realiza por parte del operario.

Por tanto, se considera un proceso no viable para su implementación como piloto demostrativo.

Caso envasado alimentación (propuesto por empresa colaboradora Alnut)

Descripción del proceso.

El proceso de estuchado de producto de Alnut, puede dividirse en varios subprocesos en distintos puntos a lo largo de una línea de montaje.

Etapa 1

Al inicio de la línea se comienza con una operación de montaje de caja. En ella el operario coge una caja premontada de un contenedor de plástico, tal y como puede verse en las imágenes mostradas a continuación.

Figura 3 – Proceso de envasado (i)
Fuente: elaboración propia en empresa Alnut

Para darle la forma deseada, se realiza presión en los bordes de la plancha y la caja se monta. Para dejarla definitivamente montada se introduce la mano dentro y se empujan las solapas y separadores interiores que contiene. Se posiciona una caja alternativamente en cada lado de la línea.

Etapa 2

El siguiente proceso de la línea de estuchado de producto es el propio estuchado. En esta intervienen varias personas cargando cada una parte de la cantidad total del embalaje.

Figura 4 – Ejemplo de productos a manipular
Fuente: elaboración propia en empresa Alnut

Etapa 3

La siguiente actividad en la línea tras el estuchado del producto es el cierre de la caja y colocación en la mesa de precintado y marcado de lote.

Figura 5 – Proceso de envasado (ii)
Fuente: elaboración propia en empresa Alnut

En esta parte de la línea el operario cierra las 3 solapas de la caja e inserta la lengüeta del cierre en su posición. En este puesto se revisan las cantidades completando la caja si fuera necesario.

Etapa 4

Una vez cerrada completa y cerrada, la caja se posición en otra parte de la línea en la que se empuja a pasar por una zona en la que se imprimirán los datos necesarios de lote y/o fecha.

Figura 6 – Proceso de envasado (iii)
Fuente: elaboración propia en empresa Alnut

Etapa 5

El traspaso de la línea a la parte final la realiza el operario en cantidades aleatorias, empleando ambas manos. Según la ocasión, el operario carga una, dos incluso tres. Posteriormente las posiciona en el carril de impresión.

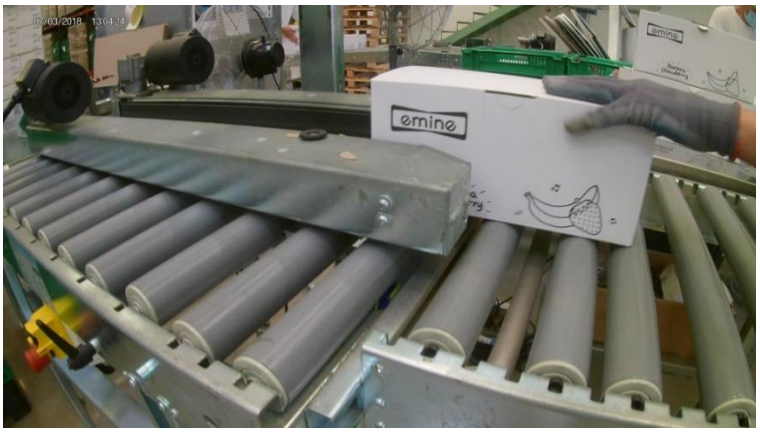


Figura 7 – Proceso de envasado (iv)
Fuente: elaboración propia en empresa Alnut

Viabilidad de automatización mediante robot

Etapa 1

Este proceso es automatizable ya que existen ejemplos en la industria de este proceso automatizado, si bien la automatización mediante robot es compleja, ya que requiere de herramientas con formas adaptadas a la caja a montar.

Etapa 2

Este proceso es complejo de automatizar ya que los artículos se encuentran ubicados en posiciones y orientaciones aleatorias dentro de la caja, y el operario escoge para el estuchado los que considera mas oportunos en cada ronda.

No hay una posición fija sobre la que trabajar. Se requeriría de sistemas de visión artificial que reconocieran la posición y orientación de los artículos, con la dificultad de obtener información sobre la altura a la que se encuentran.

Etapa 3.

Se trata de una actividad con una complejidad similar a la etapa 1. Se podría llegar a automatizar siempre que se desarrollara una herramienta de cierre de la caja adecuada.

Etapa 4

Esta parte es la que mayor complejidad tiene para su automatización, ya que requiere de una inserción precisa de la lengüeta de la caja en una ranura.

La lengüeta viene plana por lo que previamente hay que darle forma para poder insertarla, operación bastante compleja para poder realizar por demostración. A la que se une la complejidad de la herramienta para poder realizar la acción sin tocar la solapa a la hora de introducirla en la ranura.

Se considera que las etapas 2, 3 y 5 pueden llegar a ser automatizables con un estudio mas exhaustivo de las mismas.

Etapa 5

Esta etapa puede automatizarse con una herramienta de agarre adecuada (tipo pinza con ventosas), siempre que se estandaricen las posiciones de agarre y dejada de las cajas.

Idoneidad para la generación de un programa mediante demostración humana

Etapa 1

Este proceso presenta varias barreras para su aprendizaje con demostración. En primer lugar, imitar la forma en la que el operario realiza la cogida de la plancha y en segundo lugar el proceso de montaje al aplicar fuerza en los bordes opuestos.

Asimismo, el proceso de introducir la mano dentro de la caja presenta dificultad para su aprendizaje ya que las dimensiones de la caja entran en conflicto con las dimensiones de la estructura del robot o la herramienta.

Etapa 2

Las barreras para el aprendizaje por demostración en esta operación al igual que en la anterior es la operación de pick and place del artículo. Los operarios utilizan las 2 manos para tomar unas 5 o 6 unidades del producto. Ayudándose con ambas manos colocan el producto en una posición concreta dentro de la caja y en caso de ser la 2º carga de producto, tiene que hacer hueco empujando con el exterior de la mano para poder añadir las nuevas unidades.

Por otro lado, el producto se encuentra dentro de una caja en posición caótica en posición horizontal y la posición para su colocación es vertical lo que se complica dado que el propio producto tiene solo 1 parte rígida y el resto es flexible, lo que implicaría o 2 tipos de herramientas o una herramienta compleja.

Los operarios que realizan esta tarea realizan simultáneamente una tarea de inspección, consistente en revisar la correcta impresión de la fecha de consumo preferente en cada producto, desechando aquellas en las que no son perfectamente legibles. Esto lo realizan en el momento de coger el producto para introducirlo en la caja. Por tanto, no tiene sentido realizar un programa de robot mediante demostración humana en esta etapa.

Etapa 3

Se trata de un proceso con unas características similares a la etapa 1, y por tanto presenta las mismas barreras que la hacen no adecuada para un aprendizaje por demostración humana.

Etapa 4

Se trata de una actividad donde se requiere el movimiento de la pieza entre dos puntos. No importa la trayectoria seguida sino las ubicaciones. Es mucho mas fácil generar un programa de robot mediante programación tradicional. No presenta ventajas realizar un aprendizaje por demostración.

Etapa 5

Esta última etapa podría enseñarse al robot por demostración, pero condicionando la demostración a grabar la coordenada central de la parte superior de la caja (marcador para la cámara de visión artificial) y moviendo de forma unitaria.

No tiene sentido aplicar la demostración humana, si el operario no recoge y deposita las cajas de forma estandarizada siempre en las mismas ubicaciones exactas. Además, en

esta tarea lo importante son las ubicaciones de la caja, y no la trayectoria que se realiza con las mismas durante su manipulación.

Por tanto, tras el análisis realizado, no se considera el estuchado de producto un caso adecuado para realizar un programa de robot mediante demostración humana.

Caso proceso de lijado de piezas

Descripción del proceso.

La tarea que realiza el operario durante un proceso de lijado manual de piezas planas se corresponde con una trayectoria sobre la pieza de la mano que sujeta la lijadora. Es el propio operario el que decide la trayectoria a seguir sobre la pieza, la velocidad, así como el número de pasadas, en base a la geometría y material de esta.

Figura 8 – Operario lijando (i)
Fuente: elaboración propia

Figura 9 – Operario lijando (ii)
Fuente: elaboración propia

Viabilidad de automatización mediante robot

El proceso es automatizable. De hecho, los robots de lijado o pintura son habituales en diferentes sectores como el metalmecánico o el del mueble.

El programa de robot se compone de una serie de puntos sobre la superficie de la pieza a procesar sobre los que se describe la trayectoria deseada.

Idoneidad para la generación de un programa mediante demostración humana

Se trata de un proceso donde en función del tipo de pieza a procesas (tamaño, material, tipo de barniz, uso final, etc.) el operario decide como realiza las trayectorias de la operación de lijado (verticales, horizontales, circulares, ovaladas, etc.), el número de pasadas, la velocidad, la presión ejercida sobre la pieza, etc.

El operario decide en base a un "saber hacer," y es importante saber "que" hace con sus manos y "como" lo hace. Por tanto, tiene todo el sentido del mundo aplicar un proceso de demostración humana, para intentar recoger ese conocimiento del operario de lijado, e intentar traducirlo en un programa de robot que lo imite lo mejor posible.

1.3. Selección del proceso de demostración

A la vista de los tres procesos analizados en el apartado anterior, se selecciona el proceso de lijado de piezas planas como el mas adecuado para llevar a cabo un piloto demostrativo que permitiera validar los desarrollos llevados a cabo en el proyecto.

1.4. KPIs y método de la validación

En la fase de demostración se ha buscado la participación tanto de personal técnico de AIDIMME como de las empresas colaboradoras en el proyecto. Para ello, se han planificado dos fases de demostración.

En una primera fase, se llevará a cabo una demostración solo técnicos de AIDIMME. El objetivo es validar de forma exhaustiva el funcionamiento de la HMI Robotrack y poder corregir deficiencias no identificadas en etapas anteriores.

El entorno de demostración de esta primera fase emula un puesto de trabajo de lijado de piezas planas. Las trayectorias de lijado a capturar serán simples (líneas rectas) para poder verificar que la captura de datos es fiel a la realizada por el usuario. Ante la incertidumbre del funcionamiento de los programas generados en un robot real, la validación en esta primera fase se lleva a cabo con el robot ejecutando los programas en vacío sin ninguna pieza o mesa de trabajo con la que el robot pudiera llegar a interferir.

En la segunda fase, se valida la tecnología mediante la ejecución de trayectorias mas complejas y ejecutando el robot los programas generados sobre una mesa de trabajo y dos modelos de piezas diferentes. El robot ejecutará las trayectorias sin tocar las piezas de trabajo, ya que no se dispone de un entorno con sistemas de aspiración de polvo o viruta que genera el lijado de piezas.

En ambas fases, después de las demostraciones y ejecución de programas de robot presentan una etapa de cuestionarios de evaluación que rellenan las personas que realizan las demostraciones, permitiendo analizar los resultados implementar mejoras y extraer conclusiones.

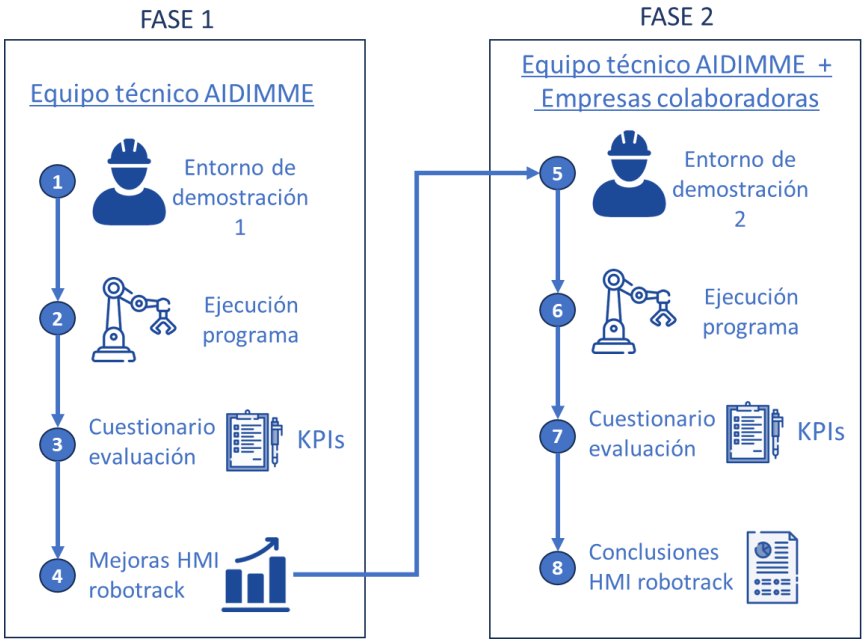


Figura 10 –Esquema del proceso de validación
Fuente: elaboración propia

2. MONTAJE DEL DEMOSTRADOR Y PRUEBAS

2.1. Demostrador en Fase 1

Montaje demostrador en Fase 1

Se utiliza una pieza de trabajo cuadrada de dimensiones de 50 x 50 cm para realizar la demostración de la trayectoria de lijado.

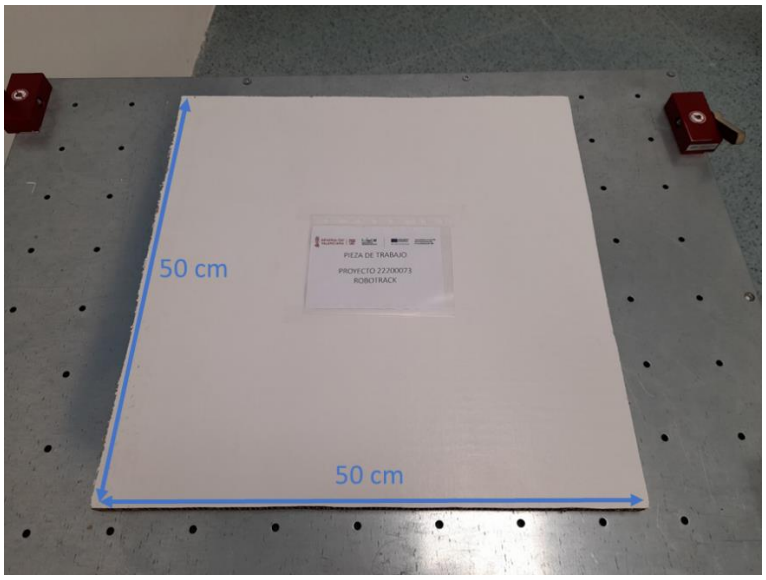


Figura 11 – Dimensiones pieza empleada en las pruebas de demostración
Fuente: elaboración propia

Se propone inicialmente que los usuarios realicen una trayectoria simple, donde tras la ejecución de esta por parte del robot se pueda comprobar que la trayectoria real realizada con las manos coincide con la trayectoria ejecutada por el robot.

Por tanto, la trayectoria base propuesta es una “X” realizada sobre las esquinas de la pieza de prueba, tal y como se muestra en las siguientes imágenes.

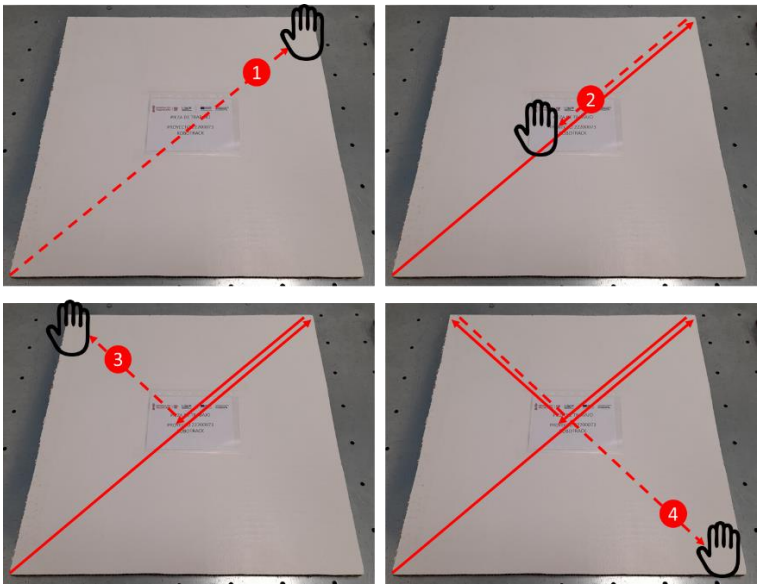


Figura 12 – Trayectoria a seguir por la mano del operario en la pieza
Fuente: elaboración propia

Se monta una mesa con la pieza de trabajo, donde llevar a cabo la demostración del proceso de lijado. Frente a la mesa de trabajo, se monta la cámara de visión, conectada a un ordenador portátil (entrono Windows) donde está instalada la aplicación Robotrack.

Figura 13 – Montaje del sistema de pruebas
Fuente: elaboración propia

En otra zona, se dispone de un robot UR5e montado sobre un vehículo autónomo de tipo AGV. Para evitar riegos de golpeo del robot contra una mesa o pieza de trabajo, esta primera validación del sistema se va a realizar ejecutando la trayectoria en el aire. En la segunda fase de validación, se montará una mesa y pieza de trabajo sobre la que el robot ejecutará las trayectorias.

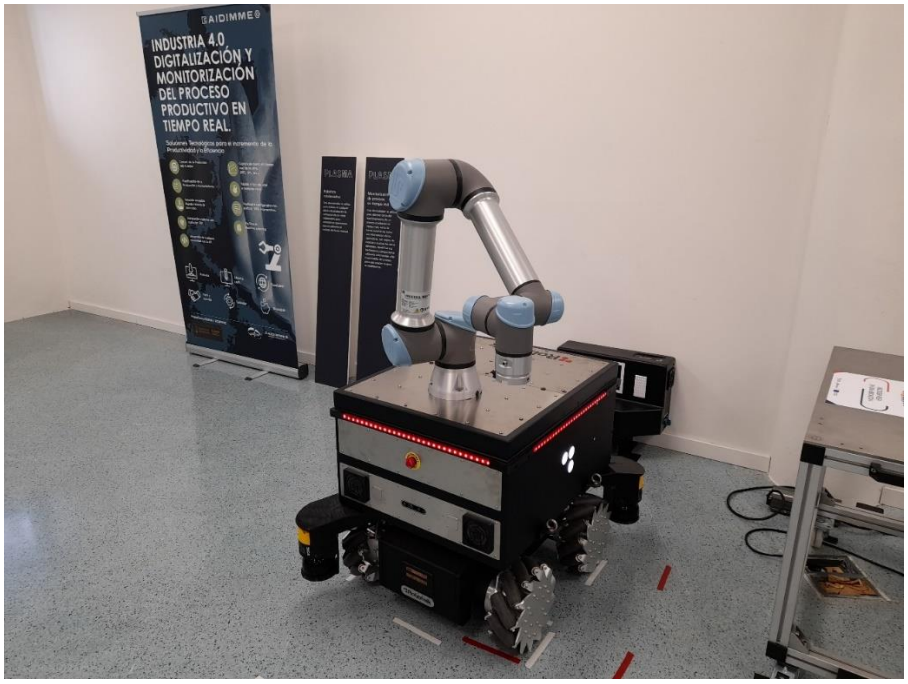


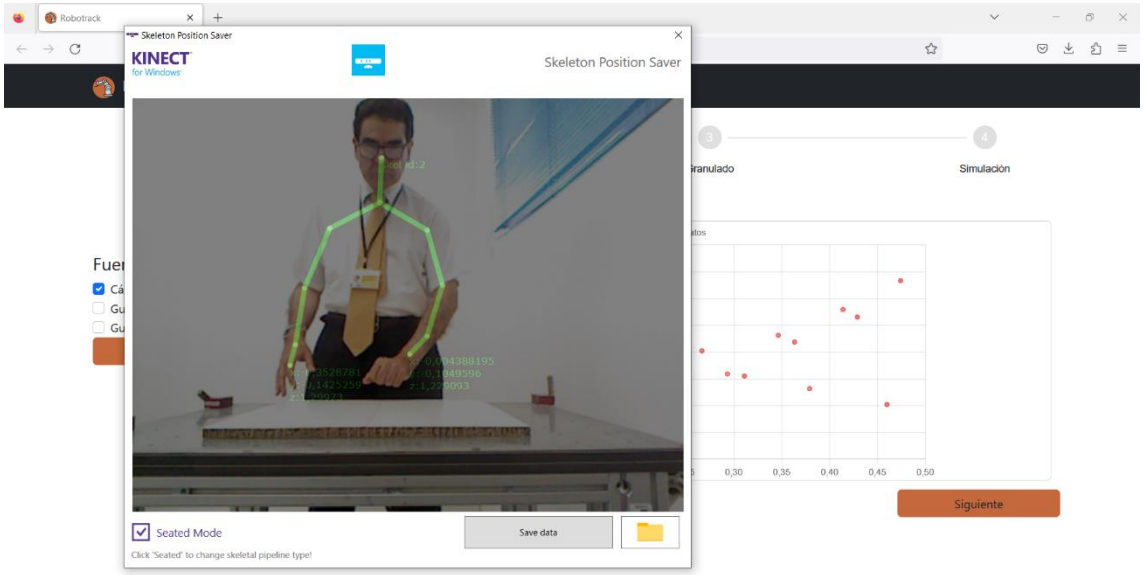
Figura 14 – Robot UR5 empleado en las pruebas
Fuente: elaboración propia

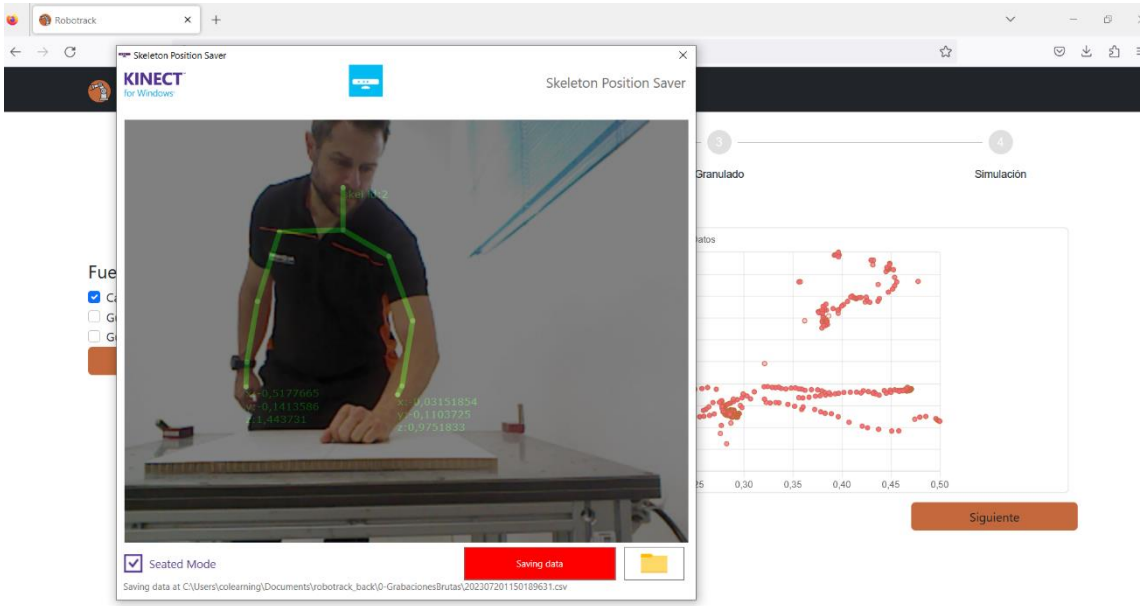
Ejecución de las pruebas

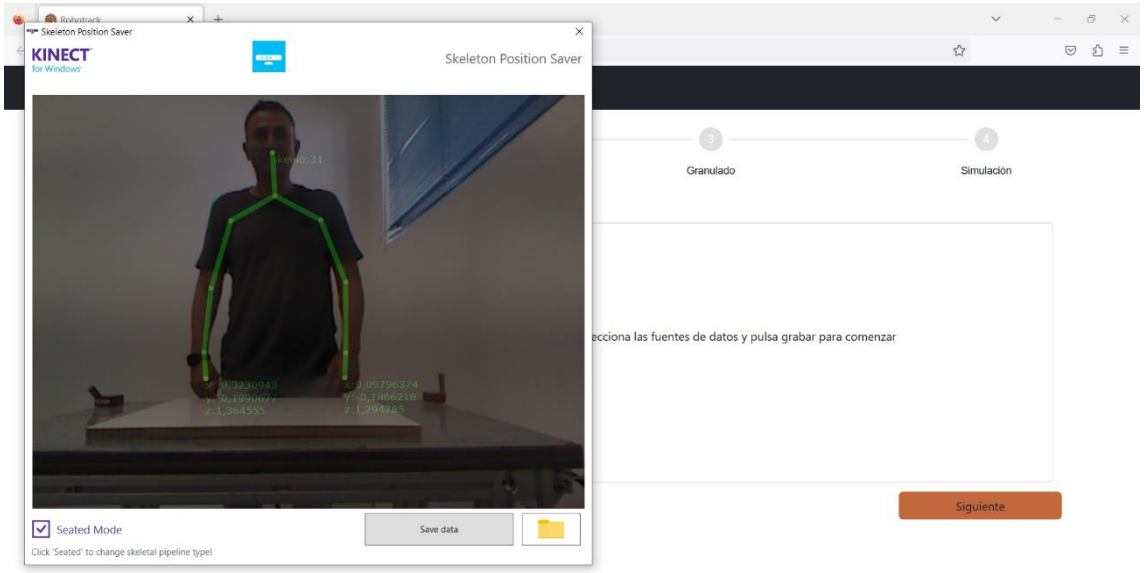
Etapa de demostración

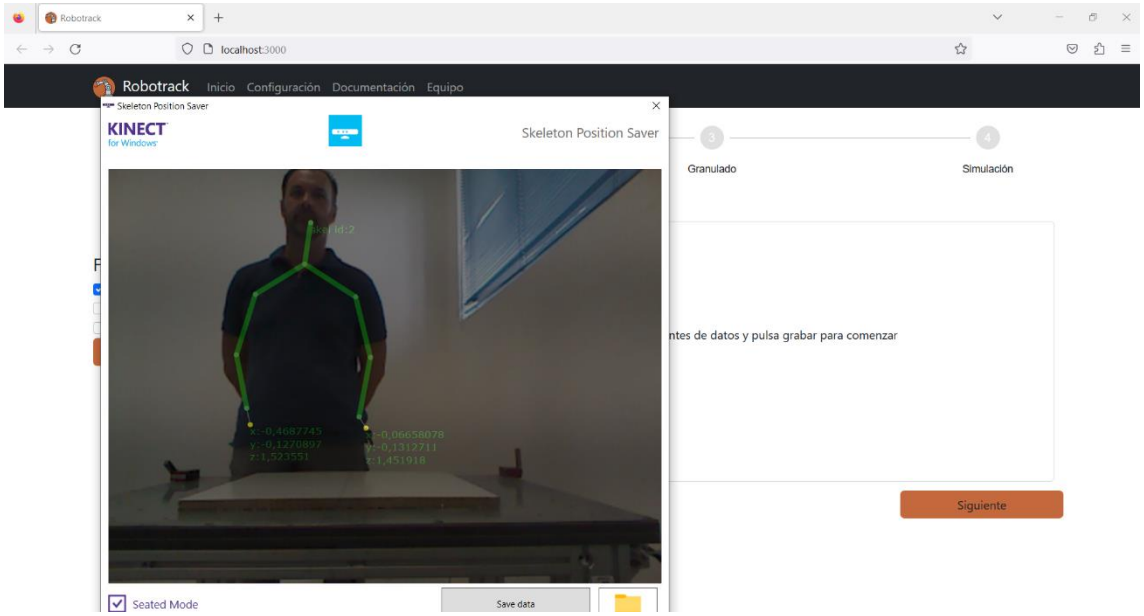
Diferente personal de AIDIMME participante en el proyecto ha llevado a cabo la fase de demostración, ejecutando la trayectoria en "X" descrita en el apartado anterior.

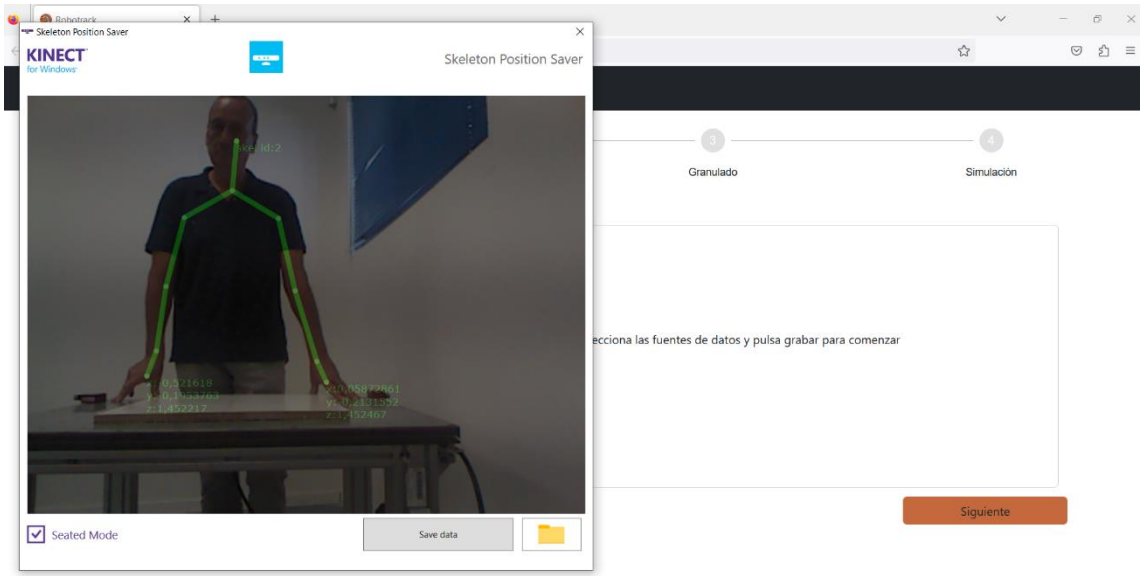
Estos usuarios han sido grabados mediante la cámara de visión, registrándose las coordenadas x, y, z del movimiento de su mano derecha sobre la pieza de trabajo. En las siguientes imágenes, se muestran capturas de pantalla de diferentes personas de AIDIMME realizando las pruebas.

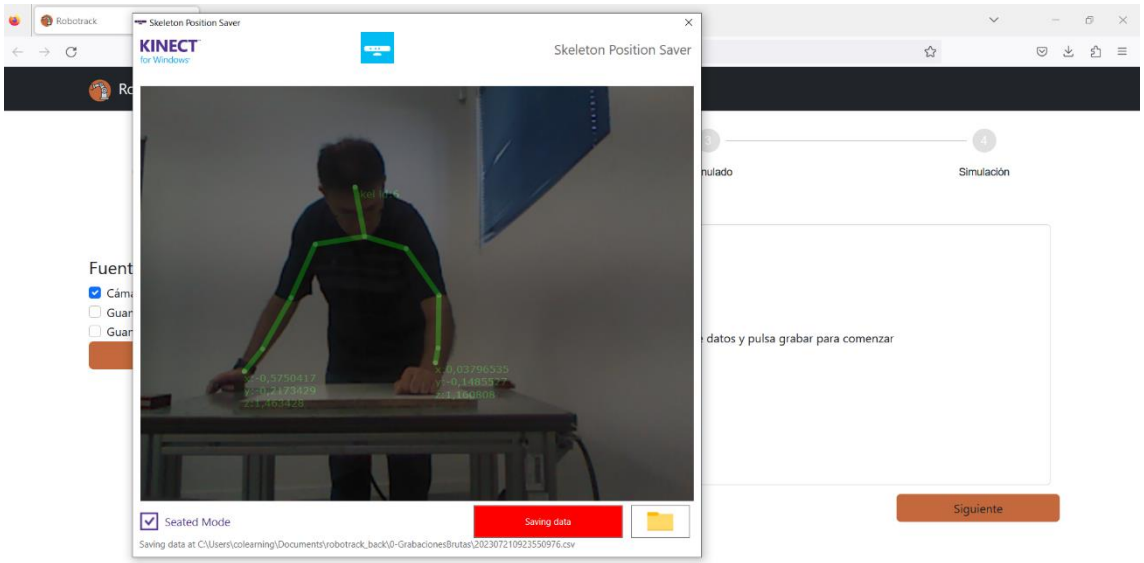

Robotrack
Skeleton Position Saver
KINECT
for Windows
Skeleton Position Saver
Granulado
Simulación
Seated Mode
Save data
Click 'Seated' to change skeletal pipeline type!
Siguiente


Robotrack
Skeleton Position Saver
KINECT
for Windows
Skeleton Position Saver
Granulado
Simulación
Seated Mode
Saving data
Saving data at C:\Users\colearning\Documents\robotrack_back\0-GrabacionesBrutas\20230720115018963 1.csv
Siguiente


Skeleton Position Saver
KINECT
for Windows
Skeleton Position Saver
Granulado
Simulación
ecciona las fuentes de datos y pulsa grabar para comenzar
Siguiente
Seated Mode
Save data
Click 'Seated' to change skeletal pipeline type!


Robotrack
localhost:3000
Robotrack
Inicio
Configuración
Documentación
Equipo
Skeleton Position Saver
KINECT
for Windows
Skeleton Position Saver
Granulado
Simulación
ntes de datos y pulsa grabar para comenzar
Siguiente
Seated Mode
Save data


Skeleton Position Saver
KINECT
for Windows
Skeleton Position Saver
Granulado
Simulación
ecciona las fuentes de datos y pulsa grabar para comenzar
Seated Mode
Save data
Siguiente


Skeleton Position Saver
KINECT
for Windows
Skeleton Position Saver
Simulación
datos y pulsa grabar para comenzar
Seated Mode
Saving data
Saving data at C:\Users\colearning\Documents\robotrack_back\0-GrabacionesBrutas\20230721092355097 6.csv
Siguiente

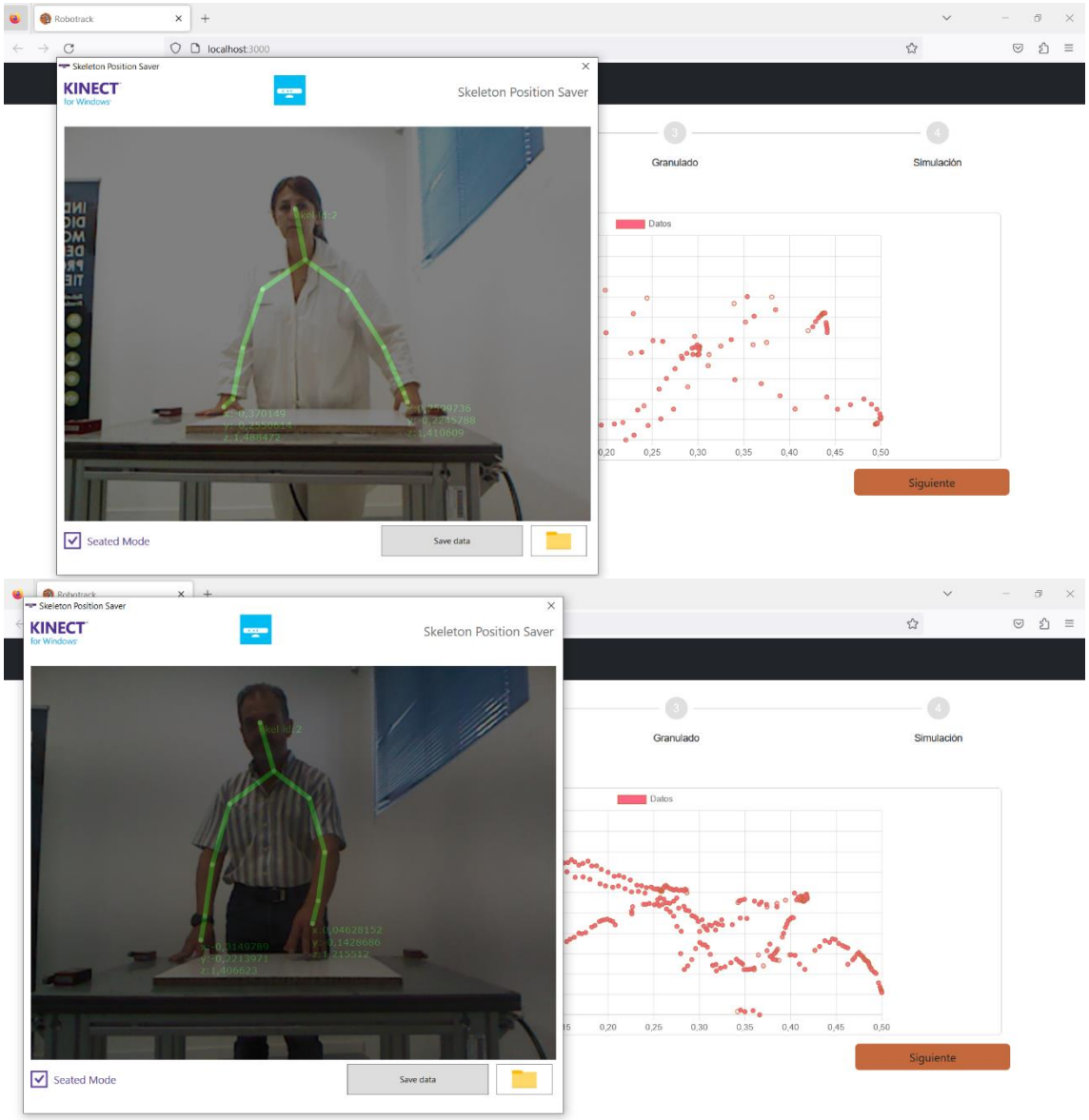
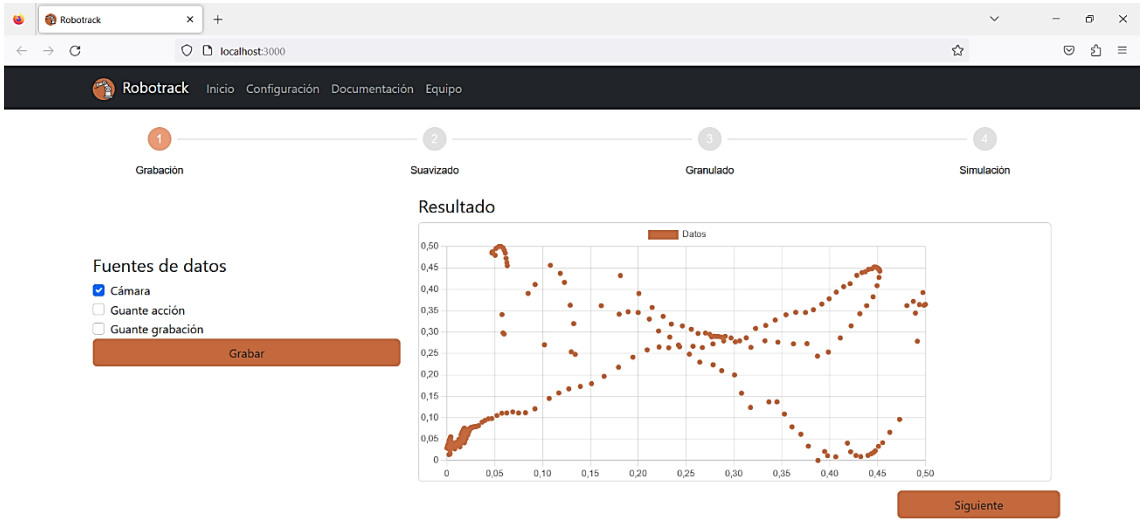


Figura 15 – Personal de AIDIMME realizando la demostración
Fuente: elaboración propia

En las siguientes imágenes se muestran los datos en bruto de alguna de las trayectorias ejecutadas durante estas pruebas, mediante un diagrama donde cada punto se representa mediante sus coordenadas x,y.


Robotrack
localhost:3000
Robotrack
Inicio
Configuración
Documentación
Equipo
Grabación
Suavizado
Granulado
Simulación
Resultado
Datos
Fuentes de datos
Cámara
Guante acción
Guante grabación
Grabar
Siguiente

Robotrack
localhost:3000
Robotrack
Inicio
Configuración
Documentación
Equipo
Grabación
Suavizado
Granulado
Simulación
Resultado
Datos
Fuentes de datos
Cámara
Guante acción
Guante grabación
Grabar
Siguiente

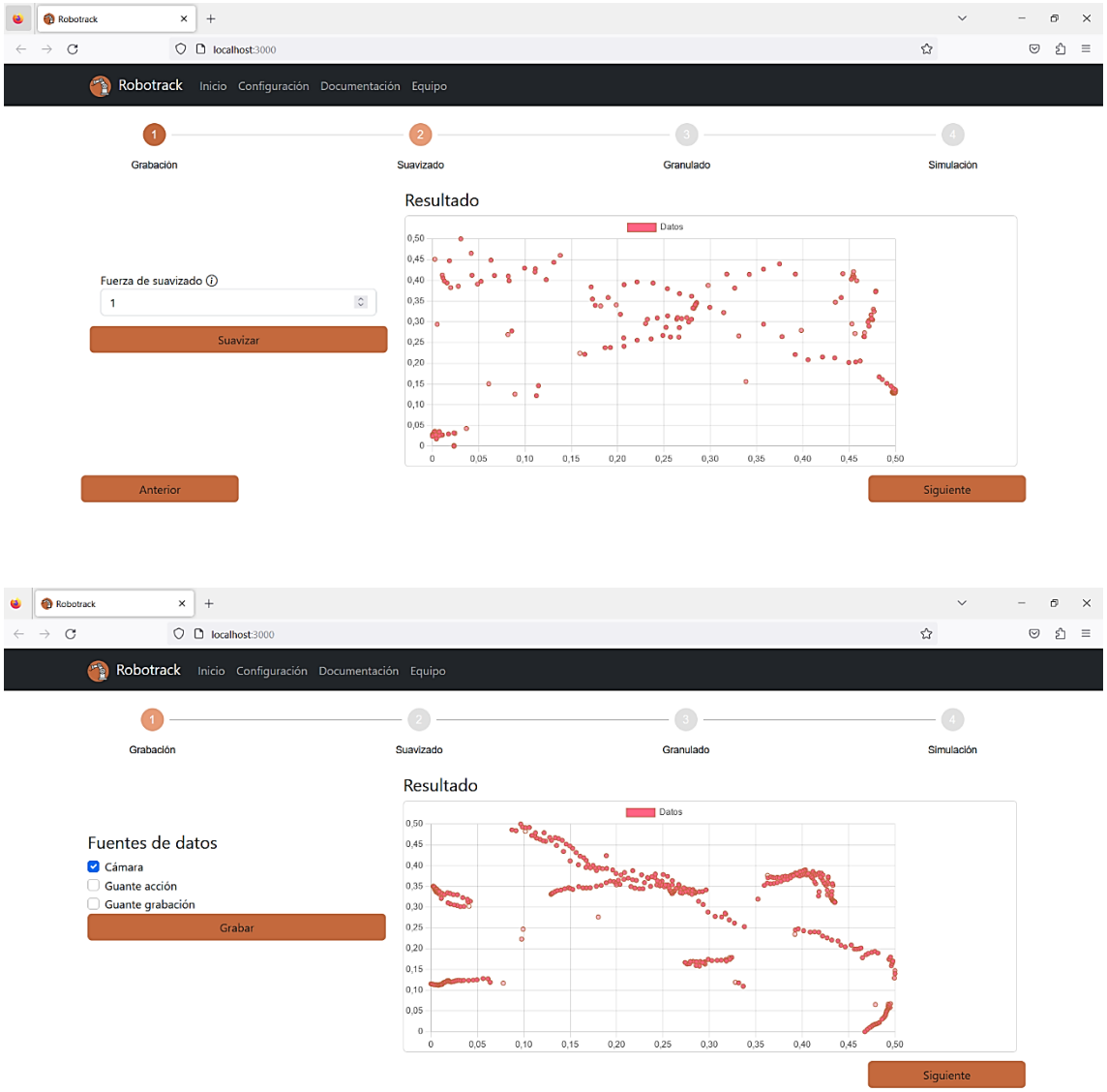


Figura 16 – Datos en bruto capturados durante la demostración
Fuente: elaboración propia

Cuando el usuario ha realizado la trayectoria de forma muy lenta se han acumulado una gran cantidad de puntos. En el caso de los usuarios que han ejecutado la trayectoria de forma muy rápida, la cantidad de puntos recogido es menor.

Etapa de suavizado de datos en bruto

Los datos capturados en bruto presentan un error inherente a la propia precisión del sistema de captura de datos empleado (cámara Kinect de Microsoft). Para mitigar este tipo de error se ejecuta un proceso de suavizado o filtrado de la información mediante un modelo de medias móviles donde el usuario puede seleccionar el número de periodos (entre 0 y 10).

El resultado de este proceso de suavizado se muestra en las siguientes imágenes.

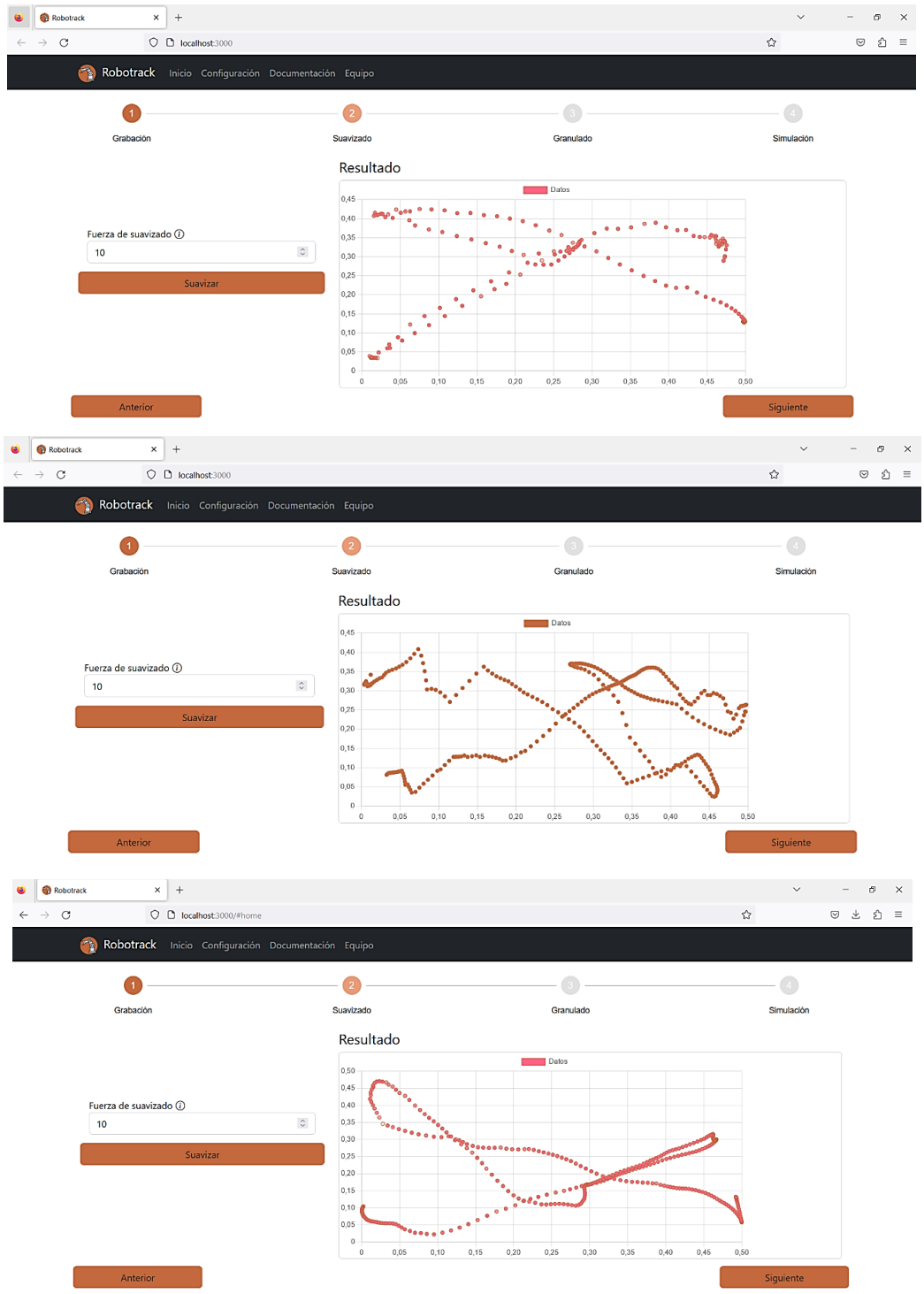
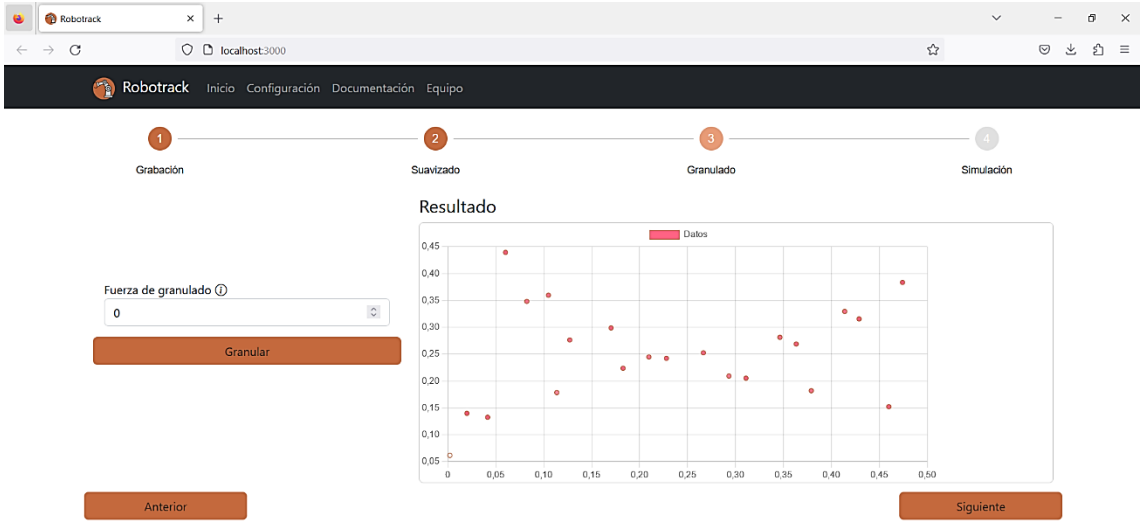


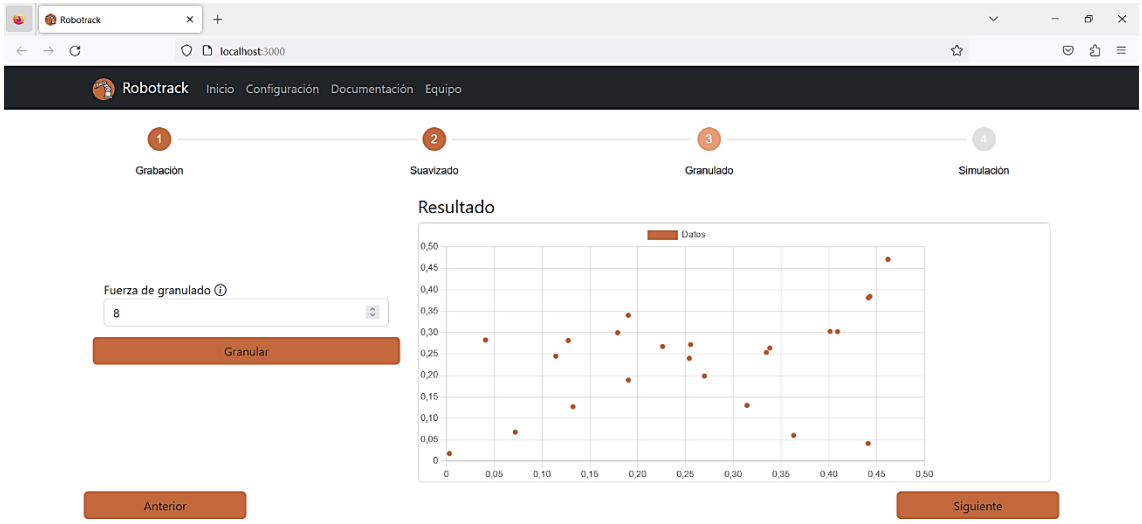
Figura 17 – Ejemplos de trayectoria tras el proceso de suavizado
Fuente: elaboración propia

Etapa de granulado

Los datos suavizados contienen una elevada cantidad de puntos muy próximos unos de otros. El robot no puede ejecutar trayectorias con puntos tan próximos entre sí. Se lleva a cabo un proceso de granulado, donde el usuario puede decidir cada cuantos centímetros deja un punto de la trayectoria, pudiendo elegir entre 1 y 10 cm.

A continuación, en las siguientes imágenes se muestra el resultado de varios ejemplos tras llevar a cabo el proceso de granulado de estos.





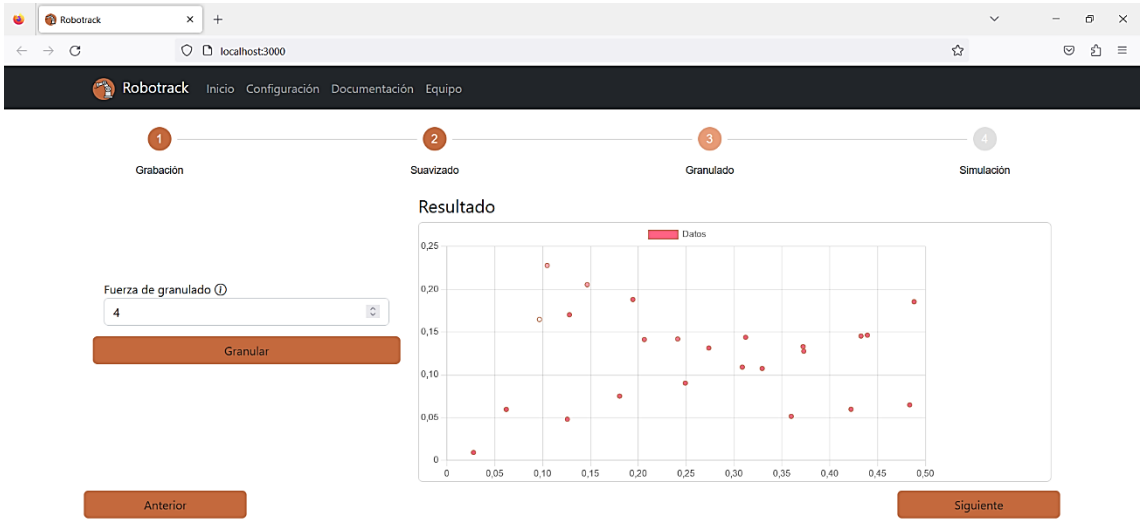
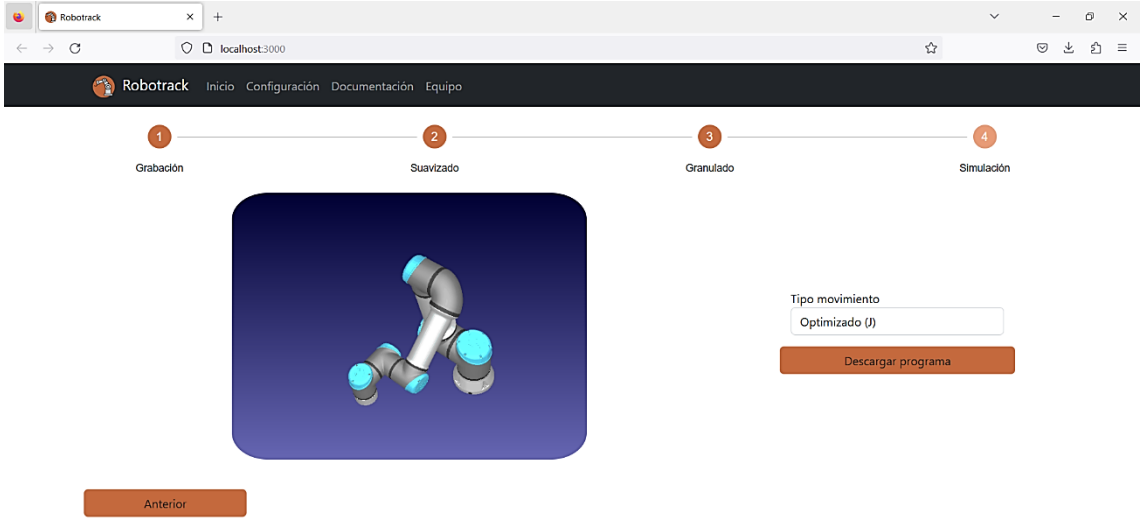


Figura 18 – Ejemplos de trayectoria tras el proceso de granulado
Fuente: elaboración propia

Etapa de simulación

A continuación, se genera video de simulación de la trayectoria del robot siguiendo la trayectoria generada tras el suavizado y granulado de los datos capturados inicialmente. En las siguientes imágenes se muestran dos capturas de pantalla de momentos de una de las simulaciones.



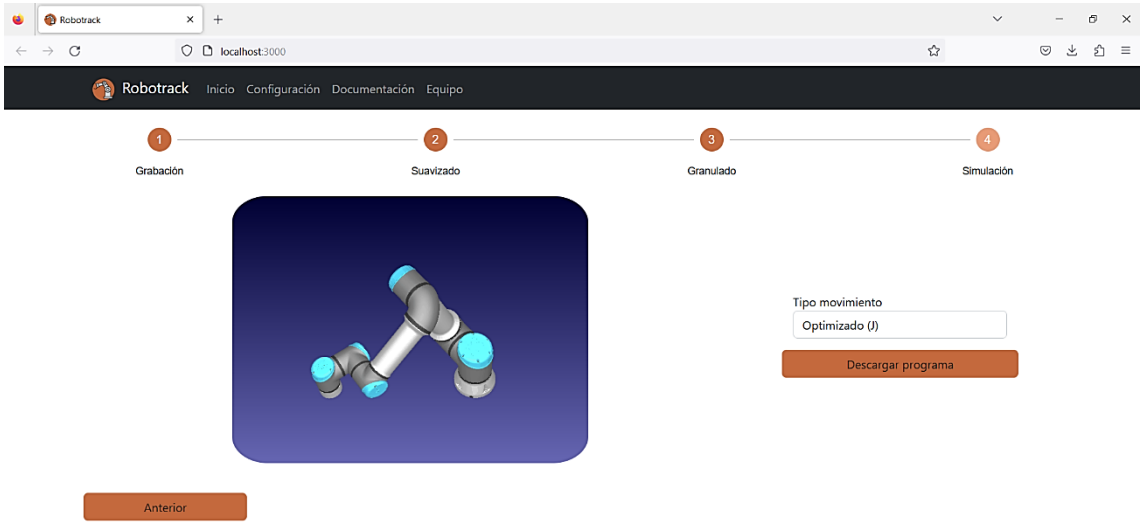


Figura 19 – Ejemplos de trayectoria tras el proceso de granulado
Fuente: elaboración propia

Si se han dejado en una misma trayectoria puntos de singularidad a los que el robot no puede alcanzar, el video de la simulación no se genera y se muestra un mensaje de error indicando el problema.

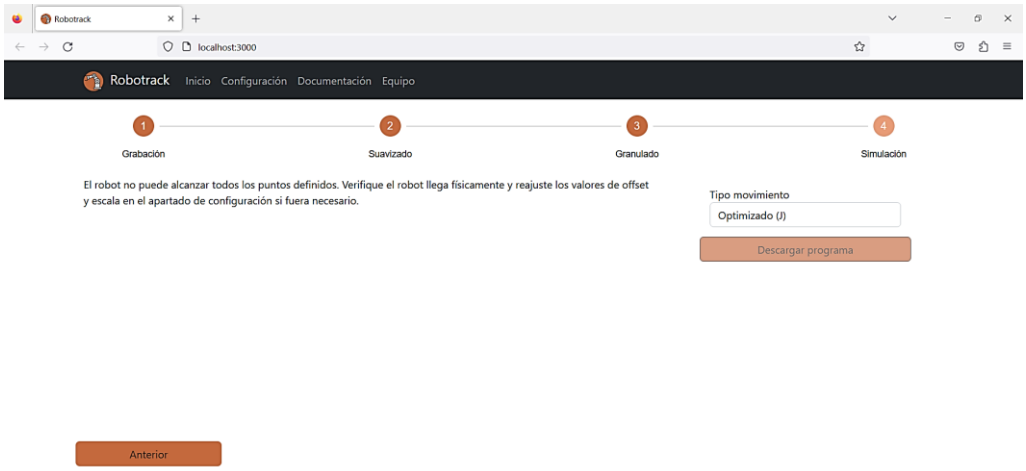


Figura 20 – Ejemplo de trayectoria que no puede ser alcanzada por el robot
Fuente: elaboración propia

Etapa de generación del programa de robot

Tras comprobar que la simulación propuesta por el sistema es correcta, se descarga del programa de robot para el modelo UR5e. En la zona de descargas aparece el fichero con el script adecuado a la sintaxis de programación del entorno de dicho robot. Se muestra

un ejemplo de uno de los URscripts generados.

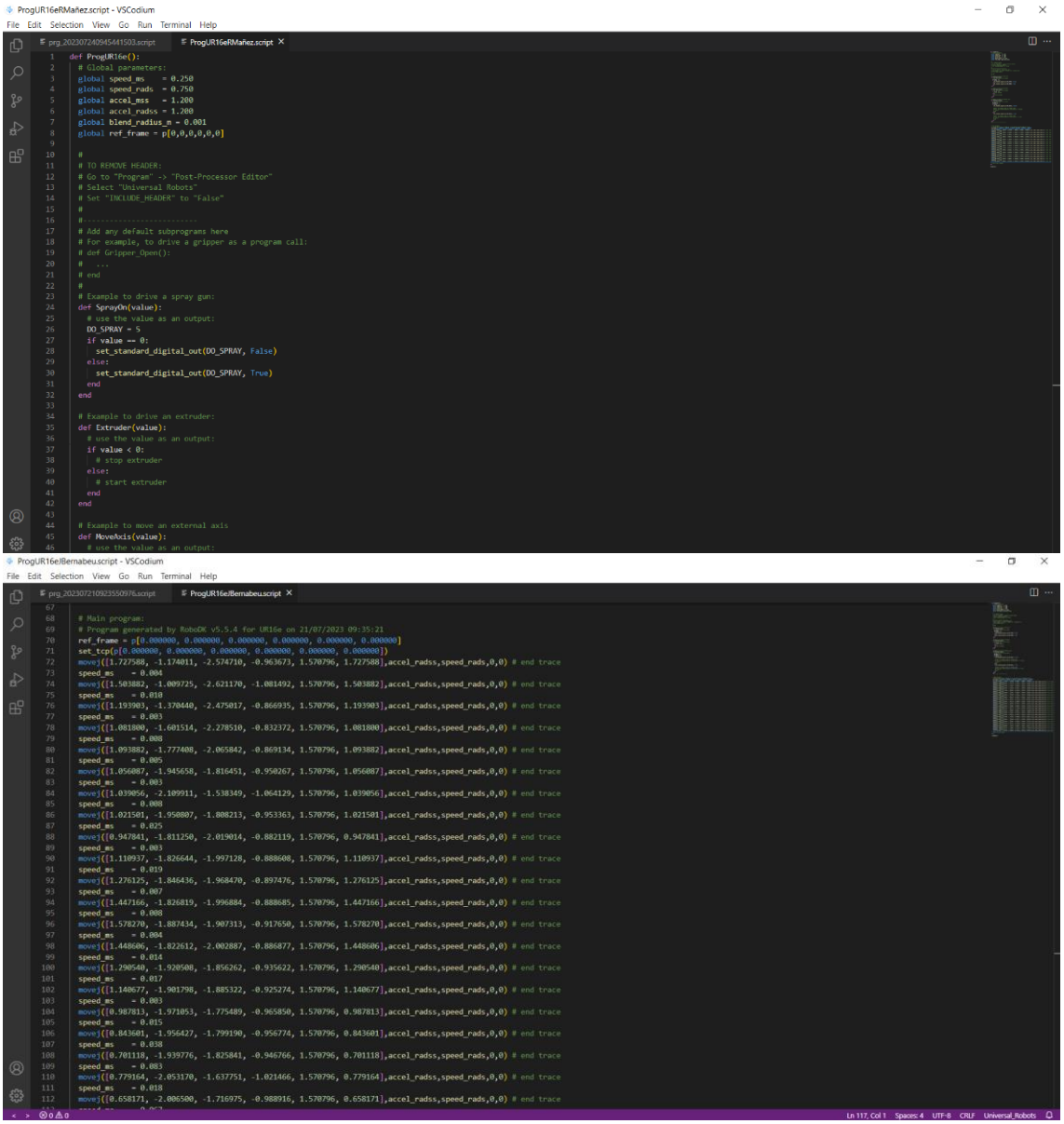


Figura 21 – Ejemplo de URscript generado
Fuente: elaboración propia

Etapa de ejecución del programa de robot

El programa descargado se guarda en un pendrive, que se carga en el robot. Desde el entorno nativo de programación del robot (Polyscope en el caso del UR5e) se abre el programa, tal y como se muestra en los siguientes ejemplos.

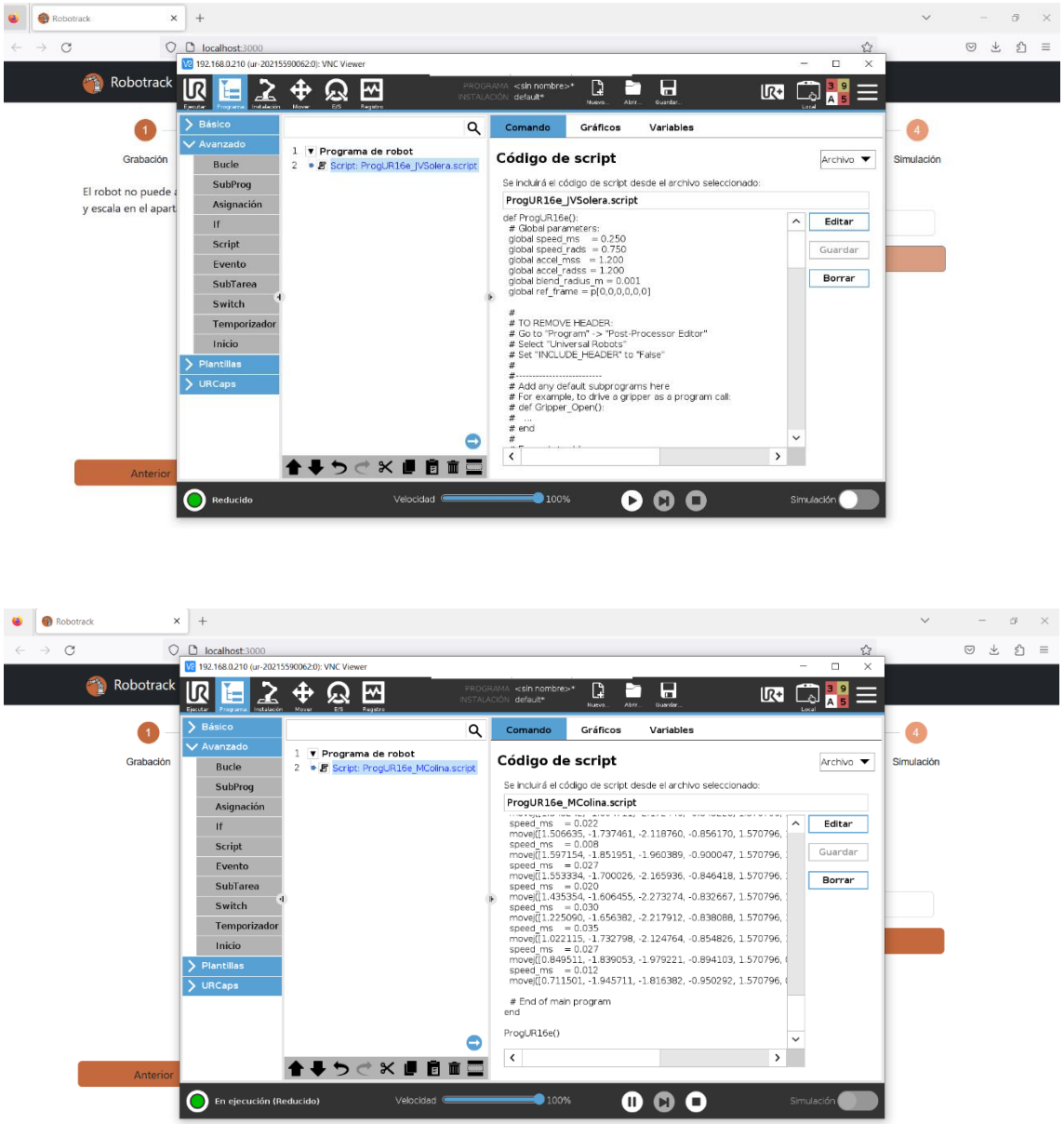


Figura 22 – Ejemplo de script cargado en Polyscope
Fuente: elaboración propia

Finalmente, se pulsa en ejecutar el programa y se comprueba como el robot sigue una trayectoria similar a la realizada por el usuario cuando estaba realizando la demostración de lijado.

Figura 23 – Ejemplo de robot ejecutando una trayectoria de lijado.
Fuente: elaboración propia

Resultados Fase 1 de la validación

Para evaluar los resultados de esta primera fase de validación se desarrolló un cuestionario donde poder valorar diferentes funcionalidades de la aplicación Robotrack e indicadores acerca de los resultados alcanzados.

Las preguntas del cuestionario relacionadas con funcionalidades de Robotrack son las siguientes.

ID	Pregunta
4A1	Asistencia al usuario no experto en el montaje del sistema de captura de datos:
4A2	Edición y gestión de diferentes sistemas de captura de datos del proceso:
4A3	Edición y ajuste de los parámetros de configuración de los sistemas de captura de datos:
4A4	Autocomprobación de la conexión y buen funcionamiento de los sistemas de captura de datos disponibles:
4A5	Ayuda con posibles soluciones tras la detección de un mal funcionamiento del sistema de captura de datos:
4A6	Permitir la grabación de datos por parte de 1 solo usuario (no depender de la ayuda de un segundo usuario):
4A7	Verificación al usuario de que los datos han sido grabados y están disponibles en el sistema:
4A8	Visualización inicial de los datos grabados en bruto:
4B1	Asistencia al usuario no experto en el proceso de transformación de los datos en bruto:
4B2	Identificación de datos incoherentes o insuficientes:
4B3	Aplicación de técnicas de filtrado, reducción, suavizado, etc.:
4B4	Generación de un archivo de trayectorias con los puntos y acciones clave del programa en un lenguaje natural de usuario (tipo flujograma, etc.) que le permita entender a alto nivel el programa de robot que va a ser generado:

4C1	Generación de un archivo de programa de robot válido para 1 robot:
4C2	Generación de un archivo de programa de robot válido para diferentes modelos y marcas de robot:
4D1	Poder realizar una simulación del programa de robot generado:
4D2	Utilizar la simulación para modificar velocidades de trabajo y establecer tiempos de ciclo:
4E1	Transferir el programa al robot:
4E2	Ejecutar el programa de robot:
4E3	Realizar paradas durante la ejecución:
4E4	Realizar modificaciones en el programa de robot tras observar el resultado de la ejecución:
4E5	Guardado del programa de robot:

Las preguntas relacionadas con la consecución de determinados objetivos son:

ID	Pregunta
5A	Conseguir un programa de robot válido a partir de los datos capturados en la fase de demostración.
5B	Conseguir, a partir de una única demostración un programa válido para diferentes modelos y marcas de robots.
5C	Precisión del programa de robot para lograr un producto con una calidad igual a la obtenida por el operario trabajando en modo manual
5D	Obtener un programa de robot cuyo tiempo de ciclo reduzca el de la actividad manual original
5E	Tiempo empleado en la generación del programa
5F	Facilidad con la que usuario consigue completar todo el flujo de trabajo (desde la demostración hasta la ejecución final del programa)
5H	Ausencia de errores durante el proceso

Se encuestó a los diez participantes de las pruebas a cerca de la importancia que otorgaban a la funcionalidad o el indicador de resultados, así como al grado de consecución de este. La valoración se realizó sobre una escala Likert 1-5, donde 1 era el valor más bajo, y 5 el valor más alto de importancia o grado de consecución.

Los usuarios que han realizado las pruebas han considerado de forma mayoritaria como importantes la mayoría de las funcionalidades descritas en el cuestionario.

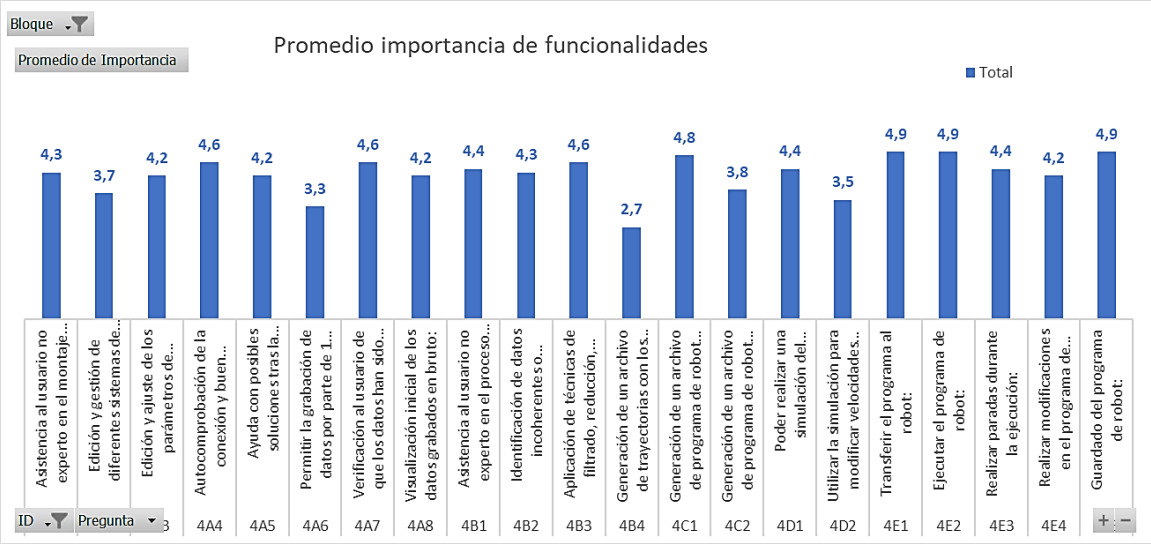


Figura 24 – Resultado promedio de la importancia de las funcionalidades
Fuente: elaboración propia

Se destacan, a continuación, aquellas con una puntuación promedio por debajo de 4 y por tanto, consideradas como menos importantes.

ID	Pregunta
4A2	Edición y gestión de diferentes sistemas de captura de datos del proceso:
4A6	Permitir la grabación de datos por parte de 1 solo usuario (no depender de la ayuda de un segundo usuario):
4B4	Generación de un archivo de trayectorias con los puntos y acciones clave del programa en un lenguaje natural de usuario (tipo flujograma, etc.) que le permita entender a alto nivel el programa de robot que va a ser generado:
4C2	Generación de un archivo de programa de robot válido para diferentes modelos y marcas de robot:
4D2	Utilizar la simulación para modificar velocidades de trabajo y establecer tiempos de ciclo:

A continuación, se muestra una gráfica que proporciona la valoración promedio acerca del grado de consecución de las funcionalidades planteadas en el cuestionario.

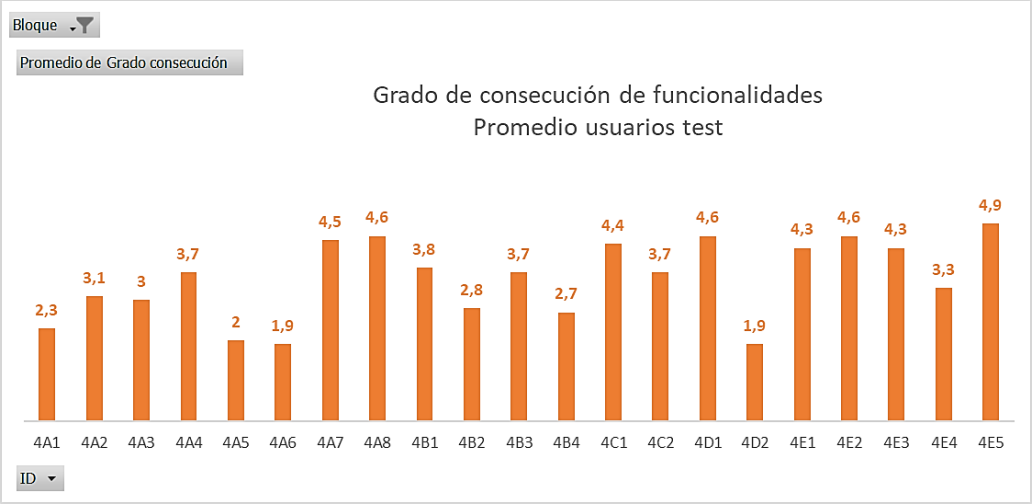


Figura 25 – Resultado promedio del grado de consecución de las funcionalidades
Fuente: elaboración propia

En la siguiente gráfica se muestra una comparativa entre la importancia otorgada a cada funcionalidad y el grado de alcance percibido por los usuarios tras la etapa de pruebas.

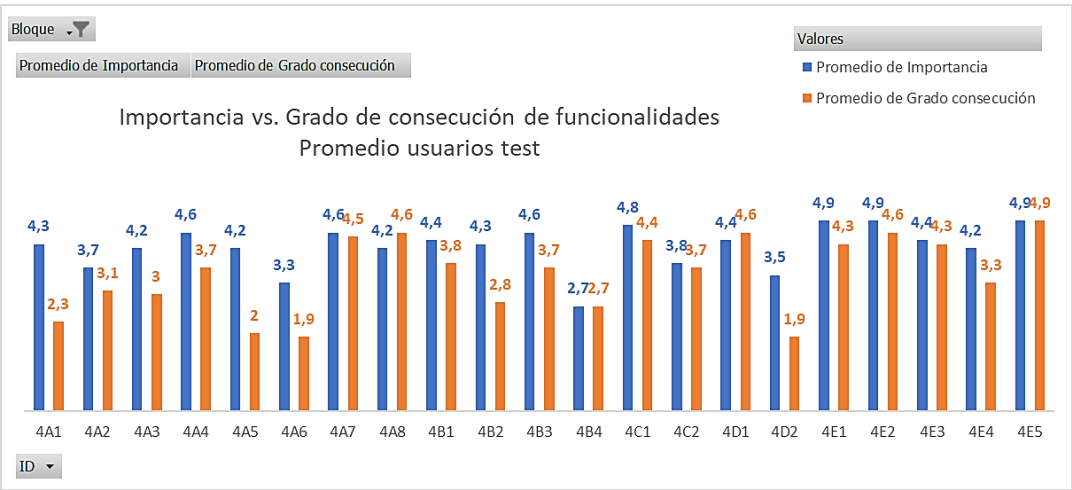


Figura 26 – Importancia y grado de consecución de las funcionalidades
Fuente: elaboración propia

En el siguiente gráfico se muestra el grado de importancia otorgado a las funcionalidades, respecto del grado de consecución de la misma en los ejes x,y. Se observa marcado en verde una serie de funcionalidades valoradas con una importancia alta y un grado de consecución también elevado. Marcadas en azul se encuentra un grupo de funcionalidades con una importancia baja y un grado de consecución también medio-bajo. Finalmente, marcadas en color rojo, se encuentran una serie de funcionalidades de importancia alta con un margen de mejora en cuanto al grado de consecución elevado.

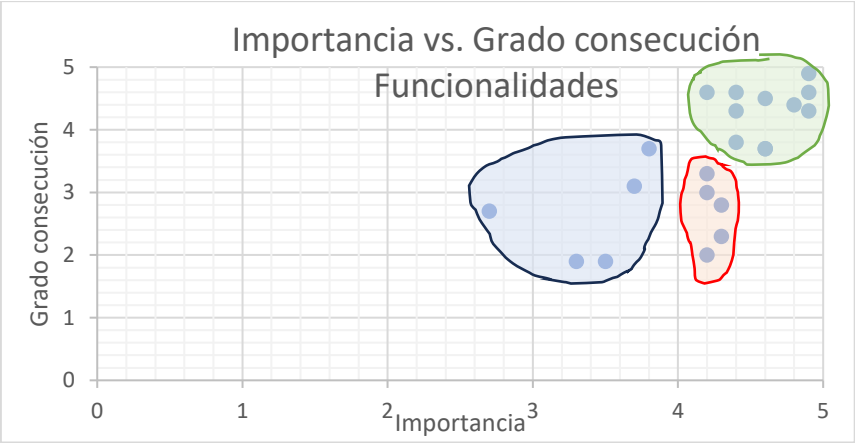


Figura 27 – Importancia vs. Grado de consecución de las funcionalidades
Fuente: elaboración propia

Estas funcionalidades resaltadas en rojo son las siguientes.

ID	Pregunta
4A1	Asistencia al usuario no experto en el montaje del sistema de captura de datos:
4A3	Edición y ajuste de los parámetros de configuración de los sistemas de captura de datos:
4A5	Ayuda con posibles soluciones tras la detección de un mal funcionamiento del sistema de captura de datos:
4B2	Identificación de datos incoherentes o insuficientes:
4E4	Realizar modificaciones en el programa de robot tras observar el resultado de la ejecución:

Se ha realizado un análisis similar para el caso de los criterios de validación de resultados del proyecto propuestos.

La importancia otorgada ha sido elevada a todos, y el grado de consecución también excepto para los dos siguientes.

ID	Pregunta
5C	Precisión del programa de robot para lograr un producto con una calidad igual a la obtenida por el operario trabajando en modo manual
5D	Obtener un programa de robot cuyo tiempo de ciclo reduzca el de la actividad manual original

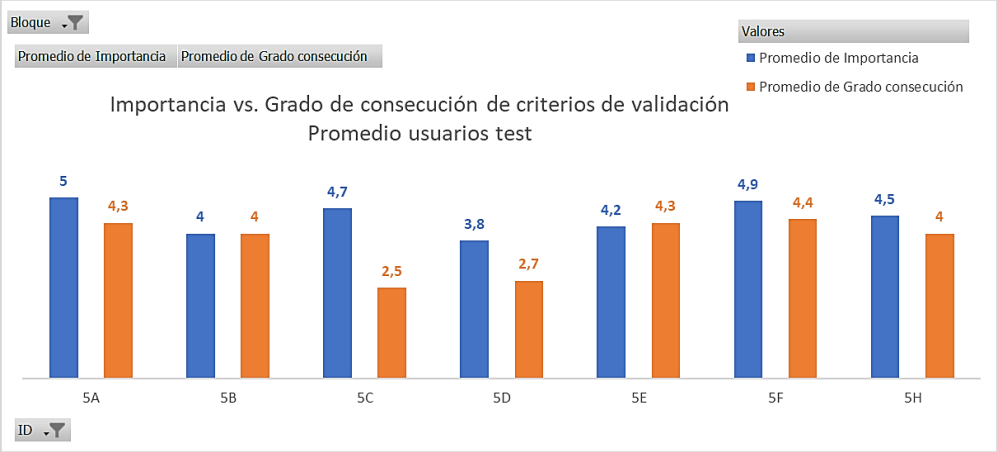


Figura 28 – Importancia y grado de consecución de los criterios de validación (resultados)
Fuente: elaboración propia

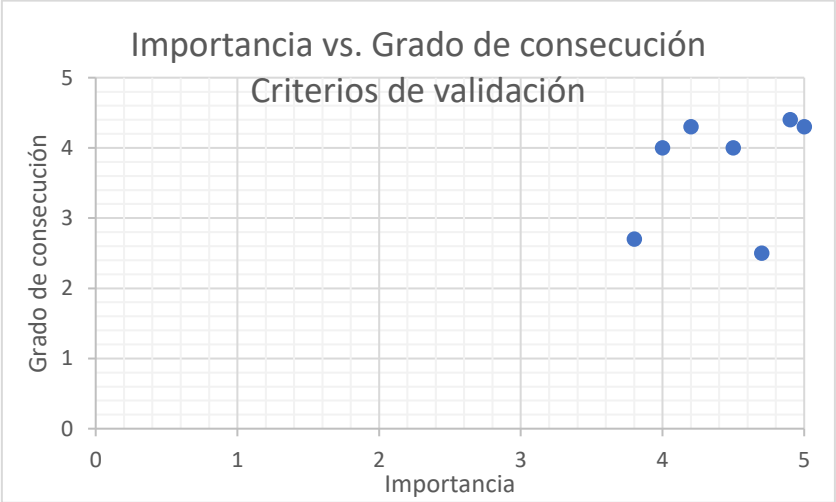


Figura 29 – Importancia vs. Grado de consecución de los criterios de validación (resultados)
Fuente: elaboración propia

2.2. Demostrador en la Fase 2

Montaje demostrador en la Fase 2

Mejoras implementadas en el sistema

Para llevar a cabo la segunda fase de ejecución de las pruebas de validación en este segundo demostrador se han implementado una serie de mejoras en la aplicación Robotrack.

Se ha habilitado una pestaña de configuración donde editar diferentes parámetros que se requieren en la generación final del programa de robot. En primer lugar, se ha habilitado una zona para introducir el origen de la zona de trabajo a partir de la cual trabará el robot, así como la altura de esta. Estos parámetros se han denominado Offset x, Offset y, Valor Z.

Todas las coordenadas de puntos de la trayectoria del robot tienen la altura fija definida en "Valor Z", y están referenciadas a partir de una distancia "Offset x", "Offset y" respecto de la base del robot.

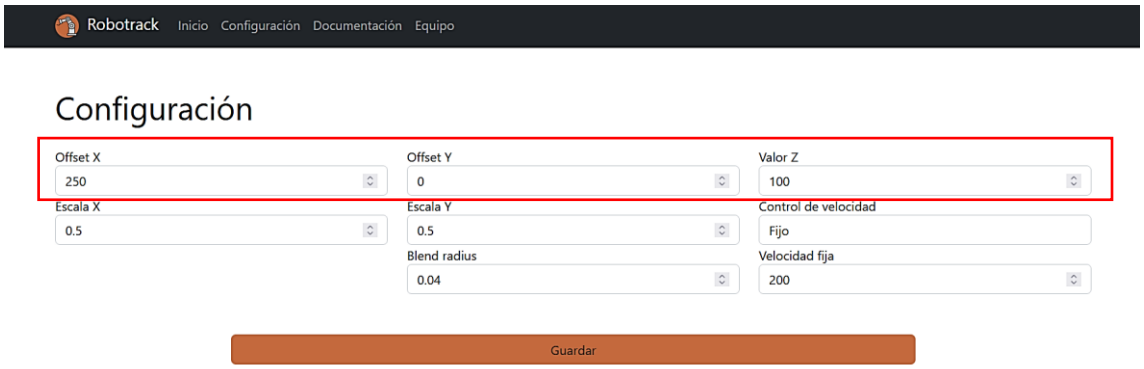


Figura 30 – Parámetros "Offset x", "Offset y", "Valor Z"
Fuente: elaboración propia

Se puede definir el tamaño de la pieza de trabajo sobre la que el robot va a ejecutar la trayectoria, mediante los parámetros "Escala X", "Escala Y". De esta manera, aunque en cada demostración se cambie la distancia o la altura de la cámara de visión a la mesa de trabajo, los datos recopilados se escalan a las dimensiones especificadas.

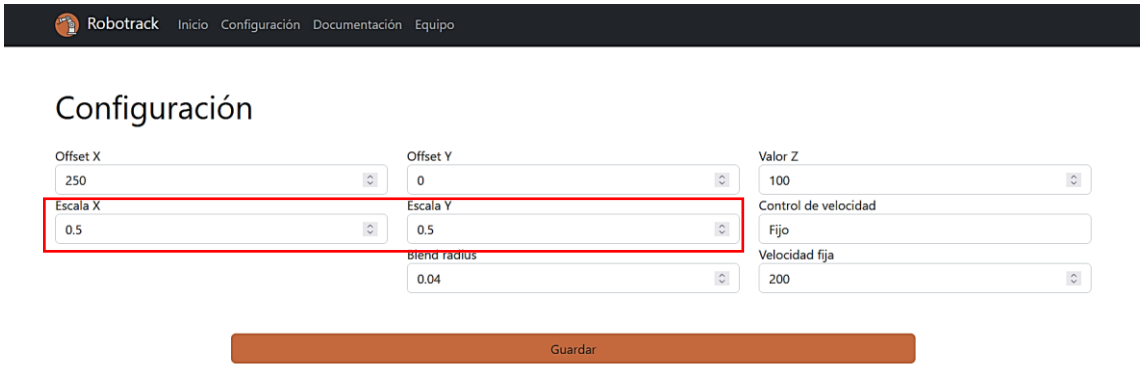


Figura 31 – Parámetros "Escala X", "Escala Y"
Fuente: elaboración propia

También se puede escoger entre dos modos de generación del programa de robot, en función de la velocidad del robot. Existe un modo donde la velocidad es proporcional a la llevada a cabo por el operario que realiza la demostración, y otro modo en el que se fija la velocidad a la que el robot ejecutará la tarea.

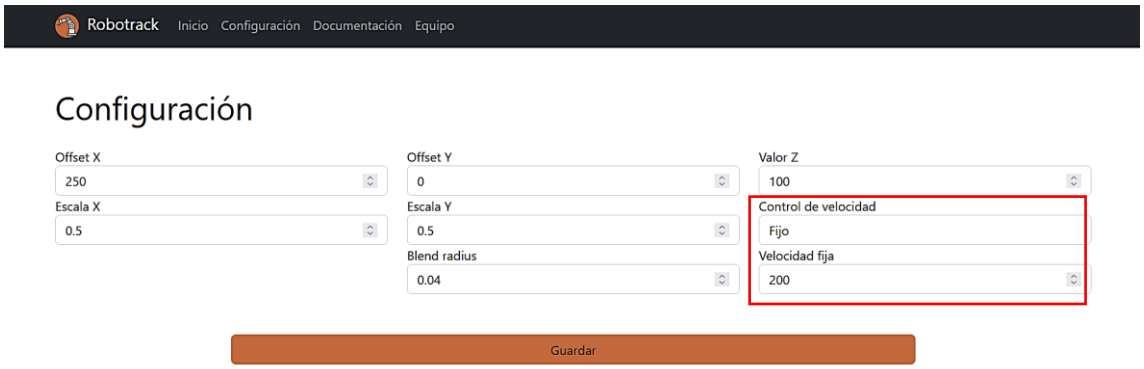


Figura 32 – Parámetros "Control de velocidad", "Velocidad fija"
Fuente: elaboración propia

Por último, tras detectar en las pruebas del entorno de demostración inicial que el robot ejecutaba paradas bruscas en cada punto de la trayectoria, se ha incorporado el parámetro "Blend radius", que permite un pequeño radio de curvatura a la hora de posicionarse sobre el punto, de manera que la trayectoria sea mas suave. Se ha definido el parámetro para que sea el usuario el que lo defina según sus necesidades.

Figura 33 – Parámetro "Blend Radius"
Fuente: elaboración propia

Modificaciones en la configuración del demostrador

Para la segunda fase de las demostraciones se utiliza la misma mesa y pieza de trabajo que en la fase 1. Para la ubicación de la cámara de visión se utiliza un trípode ajustable en altura para verificar la incidencia de la altura de la cámara sobre la calidad de los datos de trayectoria capturados.

Figura 34 – Montaje de cámara de visión y mesa de trabajo durante la demostración
Fuente: elaboración propia

Además de la pieza de trabajo de 500 x 500 mm de tamaño, se ha preparado otra pieza de 200 x 700 mm. El objetivo de disponer de dos piezas de trabajo es validar la misma trayectoria generada en la pieza original de 500 x 500 mm, en la pieza de 200 x 700 mm mediante un proceso de escalado llevado a cabo en la interfaz Robotrack.

Previamente se ha comprobado mediante el software RoboDK, que el robot UR podía ejecutar trayectorias sobre ambas superficies, ya que su alcance es reducido.

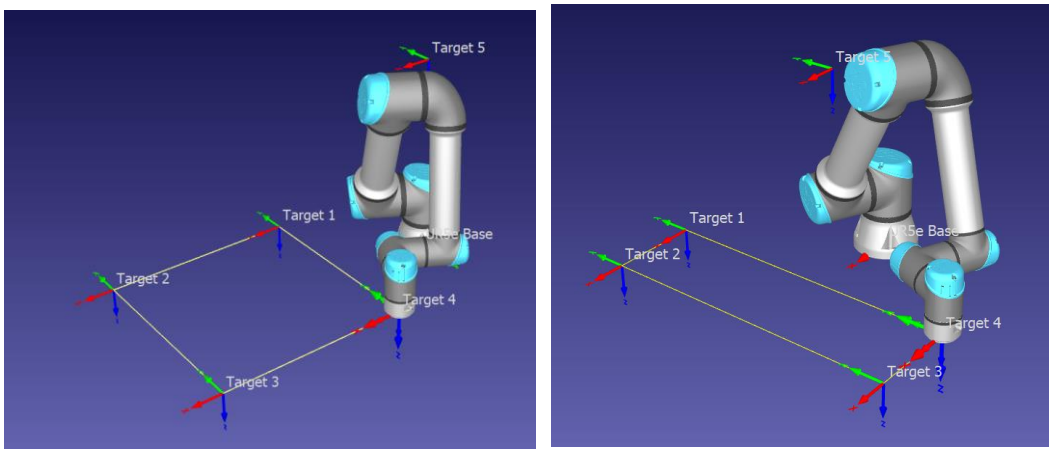


Figura 35 – Verificación de alcances del UR5e sobre ambas piezas
Fuente: elaboración propia

Se ha establecido un offset tanto en el eje x como en el eje y, para definir el punto de origen sobre el que el robot toma su origen de zona de trabajo. A partir de este origen (situado a 250 mm en el eje x, 0 mm en el eje y) se monta la mesa de trabajo donde ubicar las piezas sobre las que traba el robot.

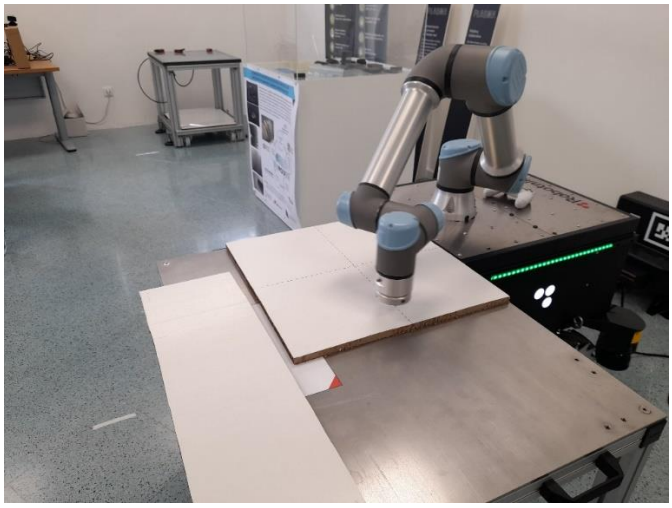


Figura 36 – Montaje de la mesa y piezas de trabajo con el robot.
Fuente: elaboración propia

La configuración de los parámetros de trabajo en la aplicación Robotrack ha sido la siguiente.

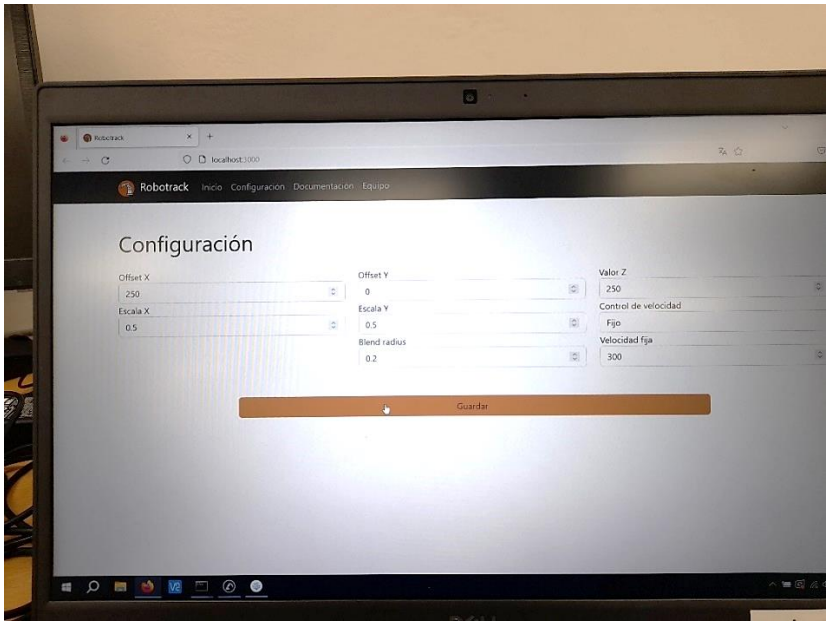


Figura 37 – Configuración de parámetros iniciales en Robotrack
Fuente: elaboración propia

Ejecución de pruebas en la Fase 2

Etapa de demostración

Se ha realizado la captura de datos de diferentes demostraciones llevadas a cabo por personal de AIDIMME, así como de personal de tres de las empresas colaboradoras del proyecto (IT8 SOFTWARE, HURTADO RIVAS y CFZ COBOTS).

Figura 38 – Ejemplo de técnico de AIDIMME siendo grabado en una demostración
Fuente: elaboración propia

Además, se han probado diferentes trayectorias sobre la pieza, empezando con las de menor complejidad (una cruz), y terminando por otras de mayor complejidad (pasadas de lija en trayectorias horizontales, verticales y circulares).

A continuación, se muestran algunas imágenes del proceso de captura de datos de las coordenadas x, y, z de la mano derecha del usuario durante diferentes demostraciones.

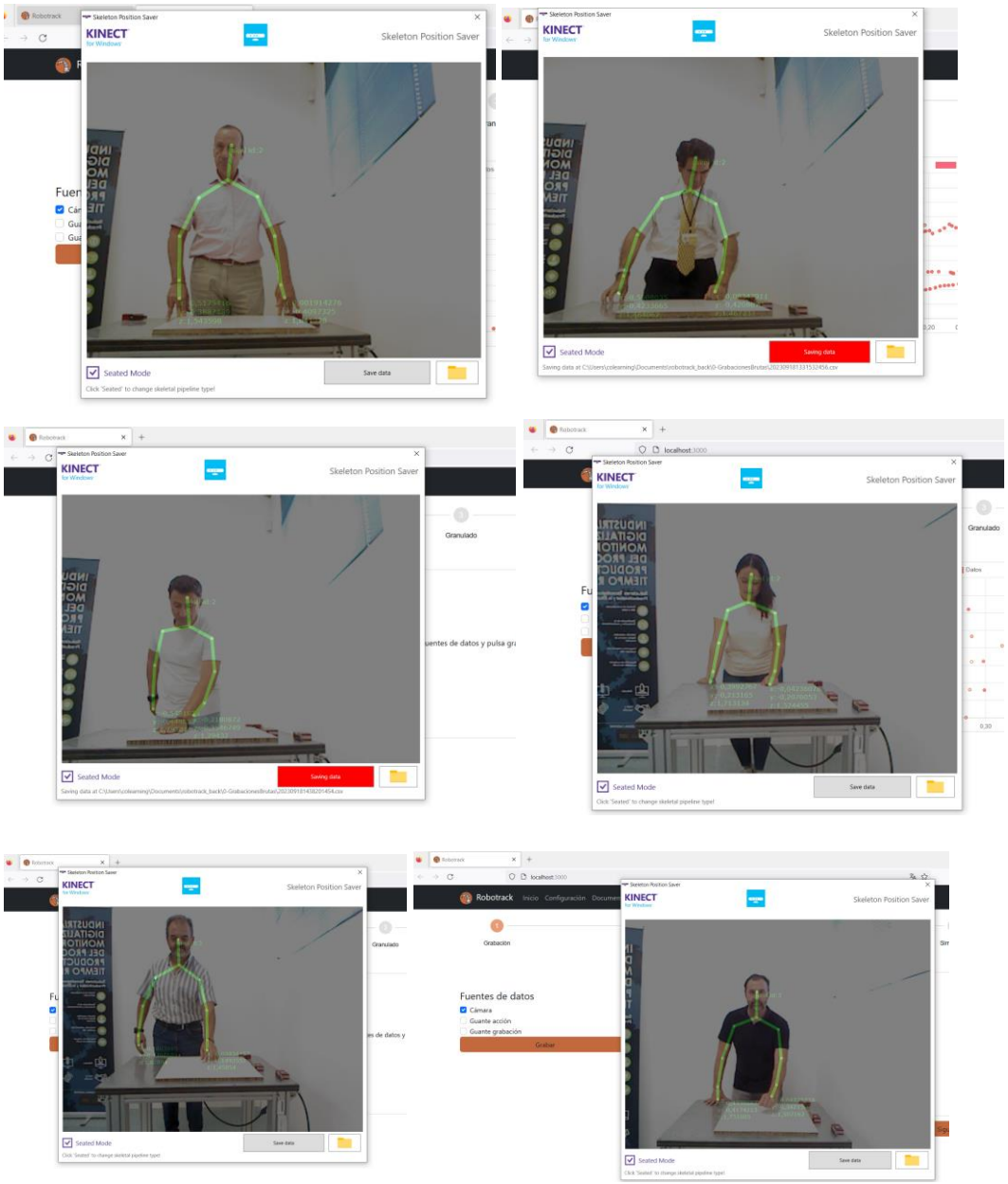
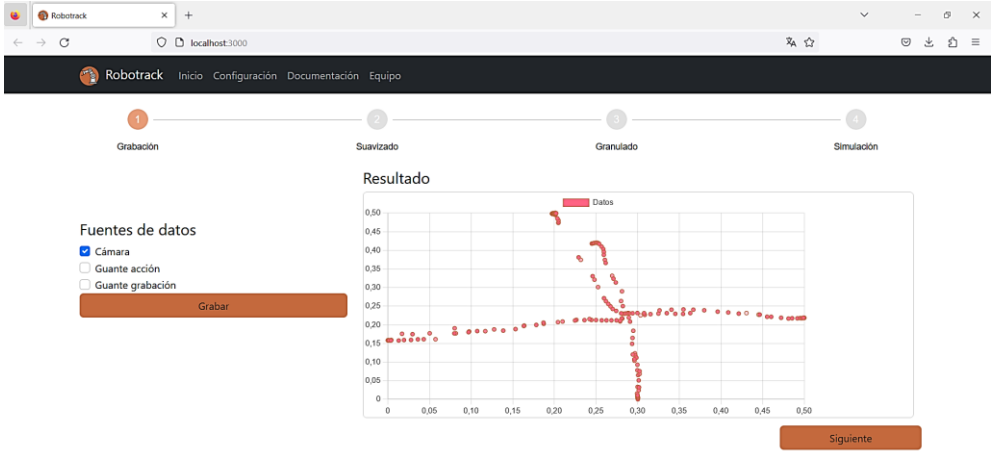
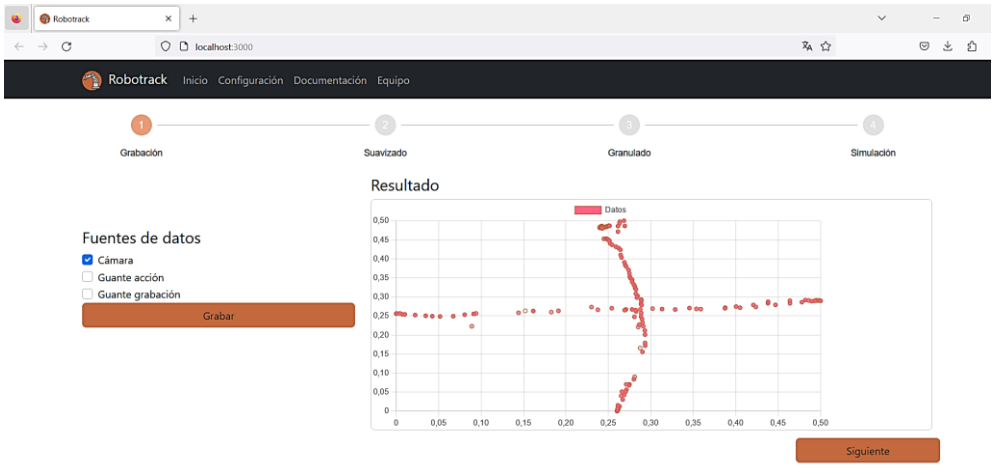
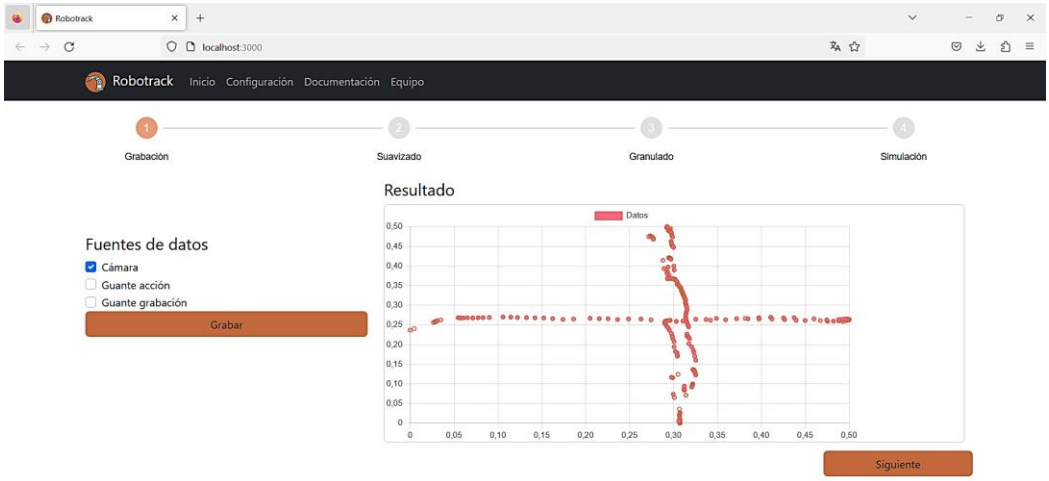


Figura 39 – Ejemplo de técnicos de AIDIMME durante la captura de datos
Fuente: elaboración propia

En una primera instancia lo que se pidió es que los participantes ejecutaran una trayectoria simple sobre la pieza, para comprobar visualmente la precisión de la ejecución del programa de robot generado. A continuación, se muestran datos en bruto registrados para diferentes usuarios.


Robotrack
localhost:3000
Robotrack Inicio Configuración Documentación Equipo
Grabación
Suavizado
Granulado
Simulación
Resultado
Datos
Fuentes de datos
Cámara
Guante acción
Guante grabación
Grabar
Siguiente


Robotrack
localhost:3000
Robotrack Inicio Configuración Documentación Equipo
Grabación
Suavizado
Granulado
Simulación
Resultado
Datos
Fuentes de datos
Cámara
Guante acción
Guante grabación
Grabar
Siguiente


Robotrack
localhost:3000
Robotrack Inicio Configuración Documentación Equipo
Grabación
Suavizado
Granulado
Simulación
Resultado
Datos
Fuentes de datos
Cámara
Guante acción
Guante grabación
Grabar
Siguiente

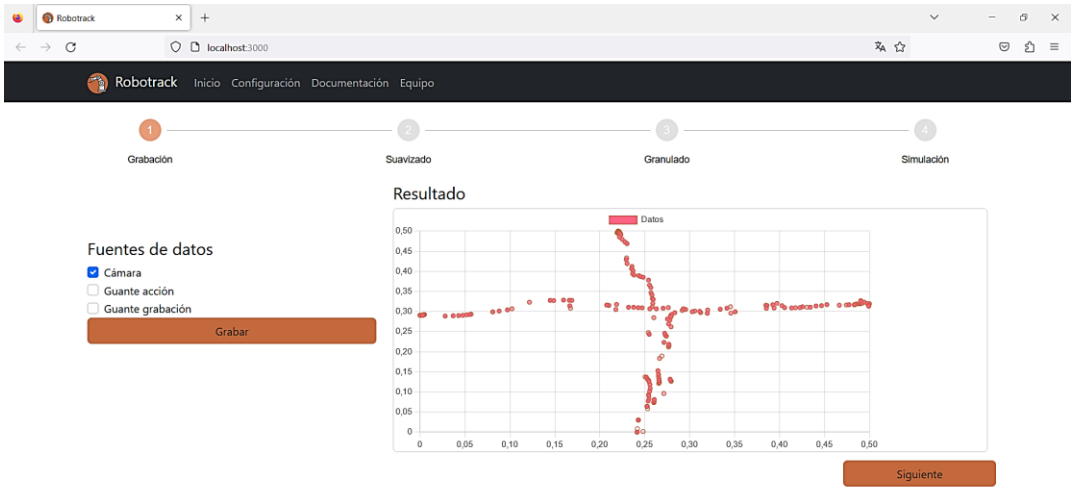


Figura 40 – Datos en bruto capturados
Fuente: elaboración propia

Etapa de suavizado y suavizado de datos en bruto

Al igual que en Fase 1 de la validación, se lleva a cabo el proceso de suavizado y granulado de los datos capturados en bruto. A continuación, se muestra un ejemplo de captura de pantalla tras haber llevado a cabo un proceso de suavizado.

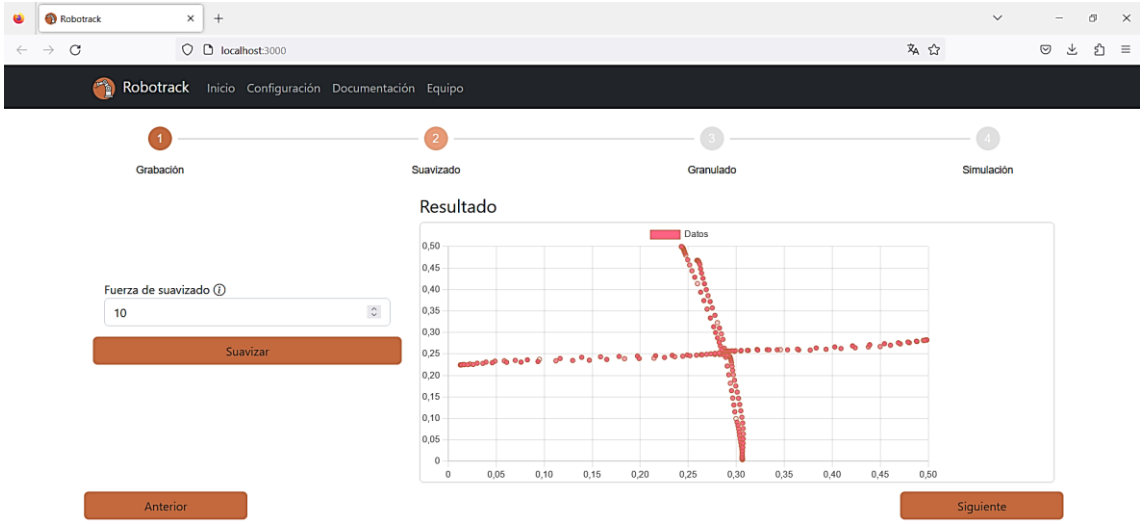


Figura 41 – Ejemplo de datos tras un proceso de suavizado
Fuente: elaboración propia

Etapa de simulación

Tras el suavizado y granulado de los datos, se genera el v ideo de simulación del robot ejecutando la trayectoria finalmente definida. Se muestra en la siguiente imagen un ejemplo de simulación generada.

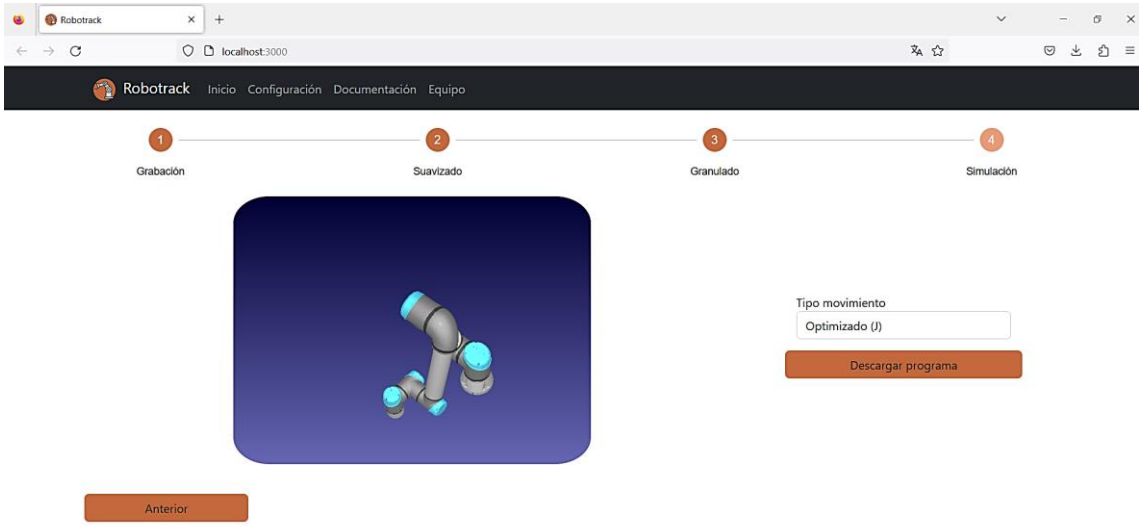


Figura 42 – Ejemplo video de simulación generado
Fuente: elaboración propia

Etapa de generación de programa de robot

Tras el video de simulación, se puede descargar el script del programa para el robot UR5e que se utiliza en el demostrador. A continuación, se muestra el inicio de un ejemplo de uno de los scripts descargados.

programa(39).script: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
def prg_20230918152511418 5():
  # Global parameters:
  global speed_ms     = 0.250
  global speed_rads   = 0.750
  global accel_mss    = 1.200
  global accel_radss = 1.200
  global blend_radius_m = 0.02
  global ref_frame = p[0,0,0,0,0,0]

  #
  # TO REMOVE HEADER:
  # Go to "Program" -> "Post-Processor Editor"
  # Select "Universal Robots"
  # Set "INCLUDE_HEADER" to "False"
  #
  #--------------------------
  # Add any default subprograms here
  # For example, to drive a gripper as a program call:
  # def Gripper_Open():
  #   ...
  # end
  #
  # Example to drive a spray gun:
  def SprayOn(value):
    # use the value as an output:
    DO_SPRAY = 5
    if value == 0:
      set_standard_digital_out(DO_SPRAY, False)
    else:
      set_standard_digital_out(DO_SPRAY, True)
    end
  end

  # Example to drive an extruder:
  def Extruder(value):
    # use the value as an output:
    if value < 0:
      # stop extruder
    else:
      # start extruder
    end
  end
```

Figura 43 – Ejemplo video de script para UR5e
Fuente: elaboración propia

Etapa de ejecución del programa de robot

Los programas generados se guardan en un pendrive, y se cargan en el robot UR5e. A continuación, se muestra un ejemplo de script cargado a partir de un programa vacío.

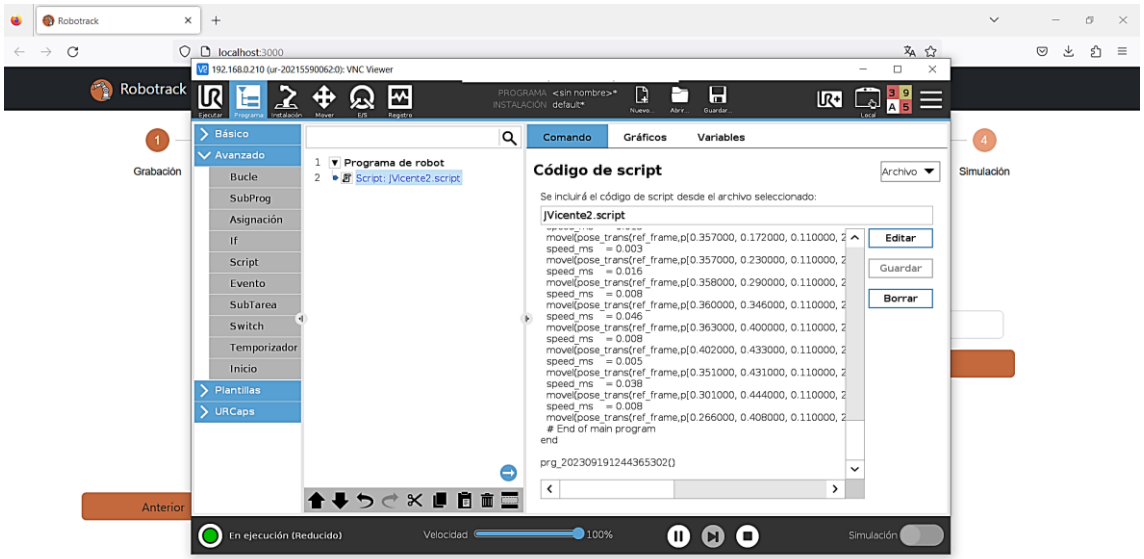


Figura 44 – Ejemplo de script cargado en Polyscope
Fuente: elaboración propia

Una vez cargado el programa, se ejecuta y se comprueba que el robot sigue una trayectoria sobre la pieza igual a la realizada por el usuario durante la fase de demostración.

Figura 45 – Ejemplo de robot ejecutando la trayectoria
Fuente: elaboración propia

Escalado programa de robots a otros tamaños de pieza

Una de las nuevas funcionalidades añadidas tras la fase de demostración inicial es la posibilidad de escalar una trayectoria generada por demostración en una pieza inicial, a las dimensiones de una segunda pieza.

Mediante la pestaña de configuración se modificaron las dimensiones de la pieza objetivo a las de una pieza de 200 x 700 mm. Con los mismos datos en bruto grabados para la pieza de 500 x 500 mm, se realizó el suavizado y granulado de los datos, tal y como se muestra en la siguiente imagen.

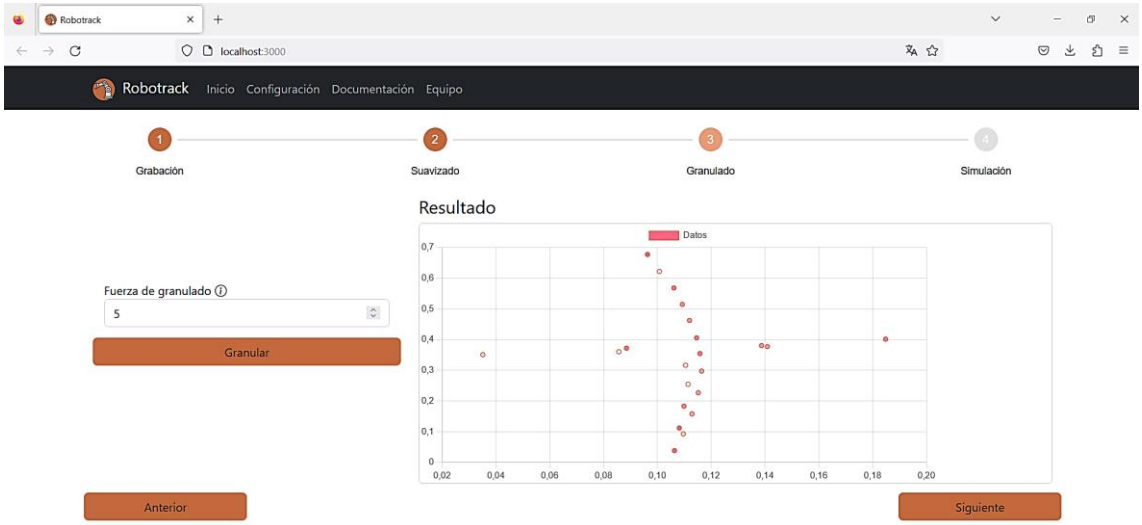


Figura 46 – Trayectoria suavizada y granulada para la nueva pieza
Fuente: elaboración propia

A continuación, se generaron las simulaciones y los programas asociados. Estos nuevos scripts se cargaron en el robot y se verificó que se ejecutaba correctamente la trayectoria original escalada sobre esta nueva pieza.

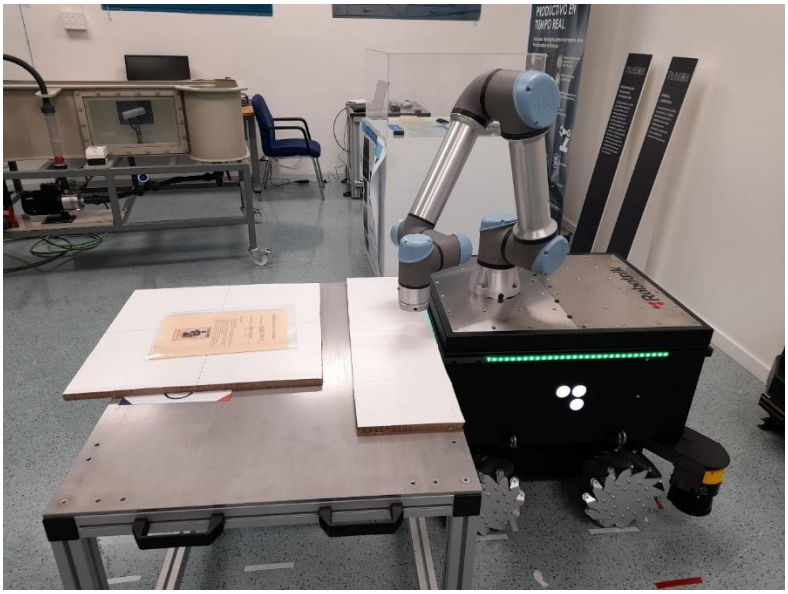


Figura 47 – Ejecución de trayectoria escalada a la nueva pieza
Fuente: elaboración propia

Pruebas con trayectorias complejas

Una vez validado el sistema para la captura y transformación de datos de trayectorias simples, se valida el sistema para trayectorias complejas, similares a las de un proceso de lijado o barnizado real. En concreto, se recrea la captura de datos para procesos de lijado mediante pasadas de lija con trayectorias en línea recta horizontal, en línea recta vertical, y mediante trayectorias circulares.

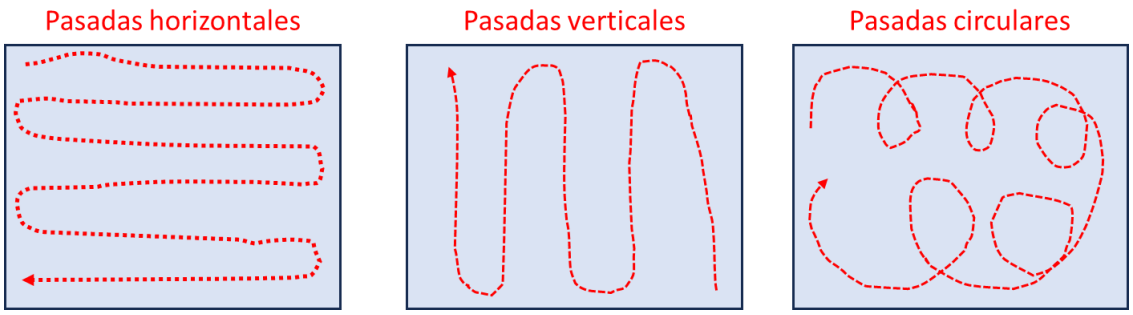


Figura 48 – Tipos de trayectorias complejas
Fuente: elaboración propia

Trayectorias mediante pasadas horizontales

A continuación, se muestran un ejemplo de trayectoria generada imitando un proceso de lijado mediante pasadas horizontales.

Figura 49 – Datos en bruto de trayectoria de pasadas horizontales
Fuente: elaboración propia

Estas trayectorias pasan por las etapas de suavizado y granulado tal y como se muestra en las siguientes imágenes.

Figura 50 – Datos suavizados de trayectoria de pasadas horizontales
Fuente: elaboración propia

Figura 51 – Datos granulados de trayectoria de pasadas horizontales
Fuente: elaboración propia

En este tipo de trayectorias mas complejas, si se granulan en exceso los datos, se pierde información de la trayectoria, tal y como se comprueba en la siguiente imagen, donde se granuló con una fuerza de 10 cm por punto.

Figura 52 – Ejemplo de granulado de datos excesivo
Fuente: elaboración propia

Al igual que en los casos anteriores, se genera la simulación de la trayectoria y el script final del robot, para validarse visualmente sobre la pieza.

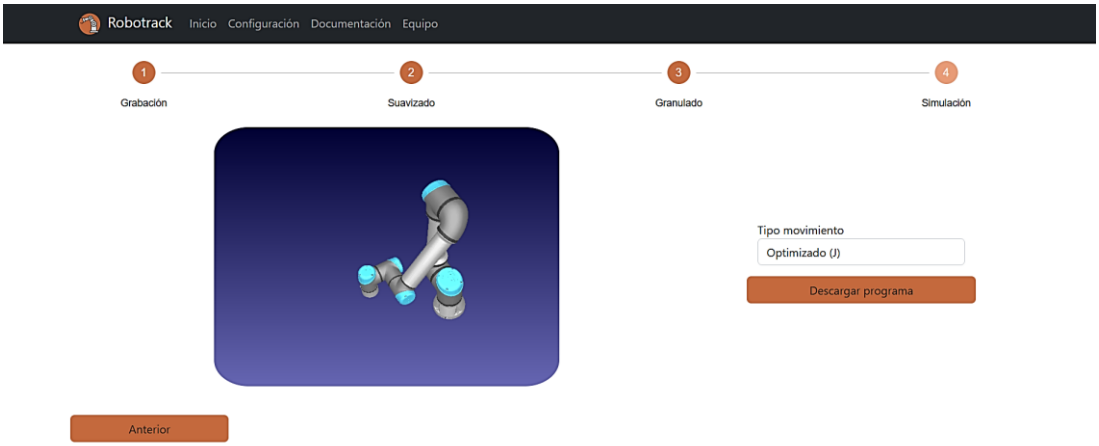


Figura 53 – Ejemplo de simulación realizada
Fuente: elaboración propia

```
programa(88).script: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
    # use the value as an output:
    DO_AXIS_1 = 1
    DI_AXIS_1 = 1
    if value <= 0:
      set_standard_digital_out(DO_AXIS_1, False)

      # Wait for digital input to change state
      #while (get_standard_digital_in(DI_AXIS_1) != False):
      #  sync()
      #end
    else:
      set_standard_digital_out(DO_AXIS_1, True)

      # Wait for digital input to change state
      #while (get_standard_digital_in(DI_AXIS_1) != True):
      #  sync()
      #end
    end
  end
  #-------------------------


  # Main program:
  # Program generated by RoboDK v5.5.4 for UR5e on 28/09/2023 08:38:52
  # Using nominal kinematics.
  speed_ms    = 0.200
  movel(pose_trans(ref_frame,p[0.441000, 0.680000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  movel(pose_trans(ref_frame,p[0.333000, 0.681000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  movel(pose_trans(ref_frame,p[0.382000, 0.588000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  movel(pose_trans(ref_frame,p[0.447000, 0.512000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  movel(pose_trans(ref_frame,p[0.344000, 0.507000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  movel(pose_trans(ref_frame,p[0.392000, 0.413000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  movel(pose_trans(ref_frame,p[0.362000, 0.316000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  movel(pose_trans(ref_frame,p[0.402000, 0.220000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  movel(pose_trans(ref_frame,p[0.301000, 0.209000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  movel(pose_trans(ref_frame,p[0.337000, 0.112000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  movel(pose_trans(ref_frame,p[0.435000, 0.088000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  movel(pose_trans(ref_frame,p[0.344000, 0.039000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  movel(pose_trans(ref_frame,p[0.283000, 0.121000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
  # End of main program
end
```

Figura 54 – Ejemplo de script generado para trayectoria de pasadas horizontales
Fuente: elaboración propia

Trayectorias mediante pasadas verticales

Con las pasadas en vertical se comprueba como la cámara de visión genera cierta distorsión de la verticalidad. La cámara de profundidad del sistema de visión induce un error de unos 4-5 centímetros entre el inicio y el fin de la línea vertical.

Figura 55 – Ejemplo de datos en bruto para trayectorias con líneas verticales
Fuente: elaboración propia

Se han probado diferentes alturas de cámara, siendo la que menor error genera aquella que se sitúa próxima a la cintura del usuario.

Los datos en bruto se suavizan y granulan siguiendo el mismo procedimiento en pruebas anteriores, obteniendo los resultados que se muestran en las siguientes imágenes.

Figura 56 – Ejemplo de suavizado con fuerza 4
Fuente: elaboración propia

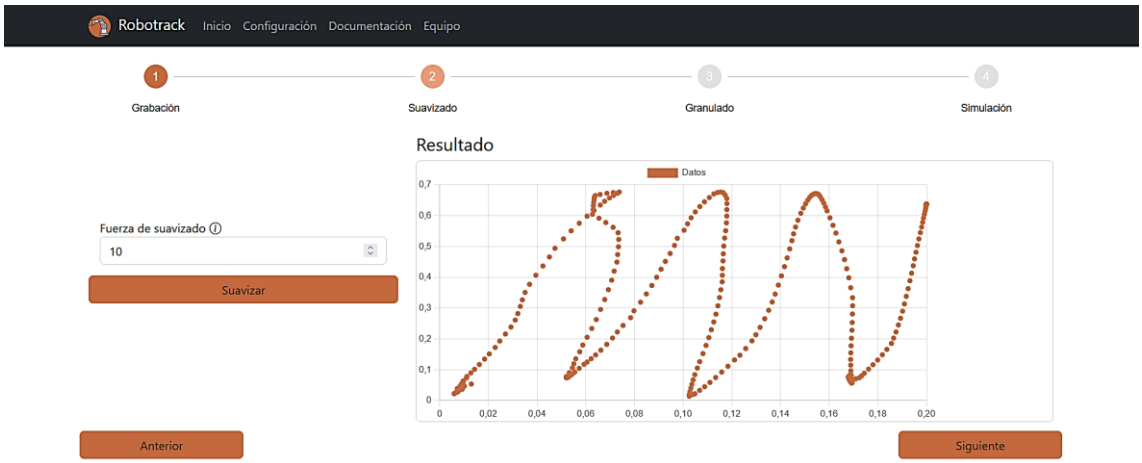


Figura 57 – Ejemplo de suavizado con fuerza 10
Fuente: elaboración propia

Figura 58 – Ejemplo de granulado con fuerza 4
Fuente: elaboración propia

Figura 59 – Ejemplo de granulado con fuerza 10
Fuente: elaboración propia

Se genera la simulación y la descarga del programa de robot, para ser testado visualmente sobre la pieza.

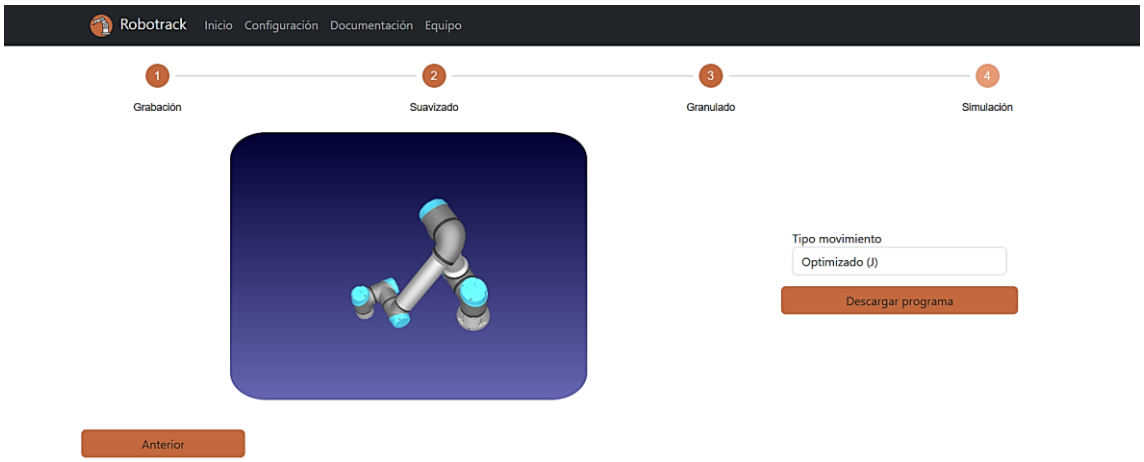


Figura 60 – Ejemplo de simulación
Fuente: elaboración propia

programa(89).script: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
# Main program:
# Program generated by RoboDK v5.5.4 for UR5e on 28/09/2023 08:46:15
# Using nominal kinematics.
speed_ms    = 0.200
movel(pose_trans(ref_frame,p[0.449000, 0.637000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.446000, 0.524000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.443000, 0.413000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.440000, 0.290000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.435000, 0.185000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.423000, 0.080000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.419000, 0.202000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.419000, 0.307000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.417000, 0.427000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.412000, 0.543000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.407000, 0.651000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.395000, 0.541000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.391000, 0.434000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.386000, 0.319000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.379000, 0.213000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.368000, 0.111000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.361000, 0.229000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.364000, 0.334000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.366000, 0.453000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.367000, 0.576000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.344000, 0.477000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.337000, 0.373000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.328000, 0.268000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.316000, 0.163000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.316000, 0.295000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.321000, 0.420000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.323000, 0.522000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.313000, 0.632000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.300000, 0.524000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.289000, 0.406000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.283000, 0.305000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.274000, 0.192000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.262000, 0.089000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
# End of main program
end
```

Figura 61 – Ejemplo de script para trayectoria con pasadas verticales
Fuente: elaboración propia

Trayectorias mediante pasadas circulares

Se capturan datos de pasadas de lijado mediante trayectorias circulares, tal y como queda recogido en la siguiente captura de pantalla.

Figura 62 – Ejemplo de datos en bruto de trayectorias circulares
Fuente: elaboración propia

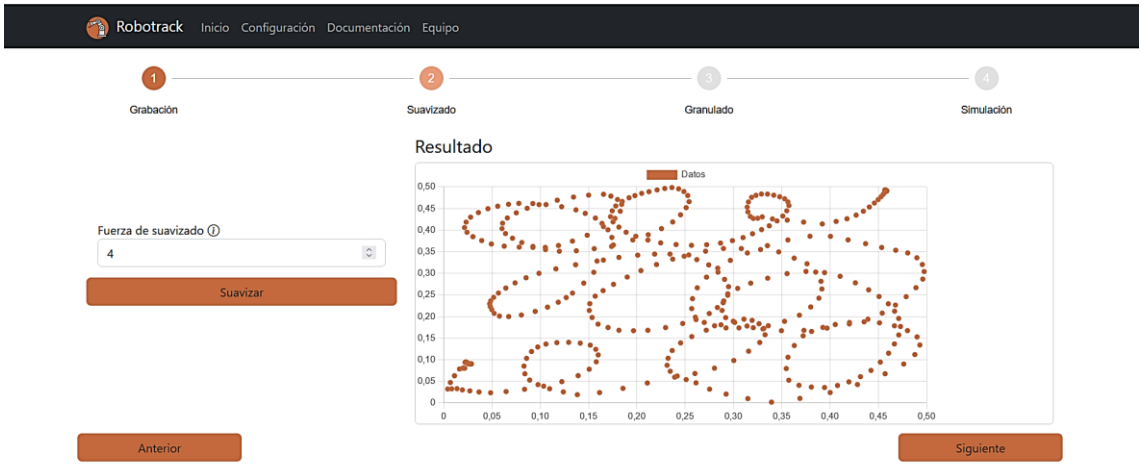


Figura 63 – Ejemplo de suavizado de datos con fuerza 4
Fuente: elaboración propia

Figura 64 – Ejemplo de suavizado de datos con fuerza 10
Fuente: elaboración propia

Figura 65 – Ejemplo de granulado de datos con fuerza 3
Fuente: elaboración propia

Figura 66 – Ejemplo de granulado de datos con fuerza 8
Fuente: elaboración propia

Se observa al igual que en el caso de otro tipo de trayectorias complejas que cuando la fuerza de granulado se acerca a 10, la trayectoria queda difuminada y el usuario no puede comprobar si es correcta o no.

Tras las etapas de suavizado y granulado se procede a la ejecución de la simulación y la generación del script de robot.

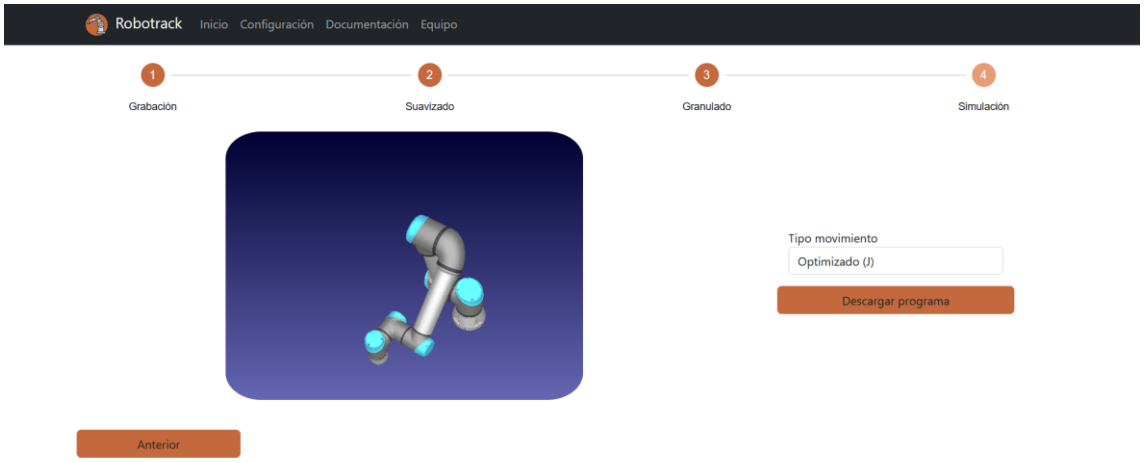


Figura 67 – Ejemplo de simulación para trayectorias circulares
Fuente: elaboración propia

programa(91).script: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
# Main program:
# Program generated by RoboDK v5.5.4 for UR5e on 28/09/2023 09:52:34
# Using nominal kinematics.
speed_ms    = 0.200
movel(pose_trans(ref_frame,p[0.707000, 0.492000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.649000, 0.426000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.569000, 0.440000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.519000, 0.371000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.443000, 0.397000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.456000, 0.483000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.456000, 0.392000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.374000, 0.359000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.323000, 0.426000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.395000, 0.473000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.385000, 0.383000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.304000, 0.372000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.319000, 0.455000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.403000, 0.423000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.409000, 0.330000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.376000, 0.249000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.302000, 0.215000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.340000, 0.288000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.427000, 0.332000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.509000, 0.323000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.534000, 0.237000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.478000, 0.180000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.409000, 0.228000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.456000, 0.303000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.540000, 0.348000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.618000, 0.315000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.622000, 0.226000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.545000, 0.180000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.520000, 0.264000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.570000, 0.344000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.646000, 0.372000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.724000, 0.339000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.725000, 0.247000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.663000, 0.187000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.570000, 0.182000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.574000, 0.274000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.657000, 0.283000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
movel(pose_trans(ref_frame,p[0.711000, 0.211000, 0.200000, 2.221441, -2.221441, -0.000000]),accel_mss,speed_ms,0, blend_radius_m)
```

Figura 68 – Ejemplo de script para UR5e con trayectorias circulares
Fuente: elaboración propia

Resultados fase 2 de la validación

Antes de esta segunda etapa de la validación se introdujeron diversas mejoras en la aplicación Robotrack. Con el objetivo de verificar si se han logrado alcanzar los beneficios de las mejoras se plantearon una serie de cuestiones a valorar por los usuarios participantes en las pruebas.

ID	Pregunta
1A	Se ha mejorado la precisión a la hora de registrar los puntos de la trayectoria real de lijado ejecutada por el usuario
1B	Es posible establecer un origen de pieza diferente en cada demostración, en función de cómo se situará la mesa y la pieza de trabajo del robot
1C	Es posible establecer la altura de trabajo del robot en cada nueva demostración
1D	Es posible adaptar el programa del robot a piezas de diferentes dimensiones (siempre dentro del alcance del robot)
1E	El robot ejecuta la trayectoria de lijado de forma "suave" y sin paradas bruscas durante su ejecución
1F	La velocidad a la que el robot ejecuta la trayectoria es proporcional a la velocidad empleada por el usuario

Figura 69 – Cuestiones acerca de la mejora alcanzada
Fuente: elaboración propia

Los resultados de los cuestionarios muestran que se considera que las cuatro primeras mejoras han sido alcanzadas, ya que presentan una puntuación promedio de 4 o superior.

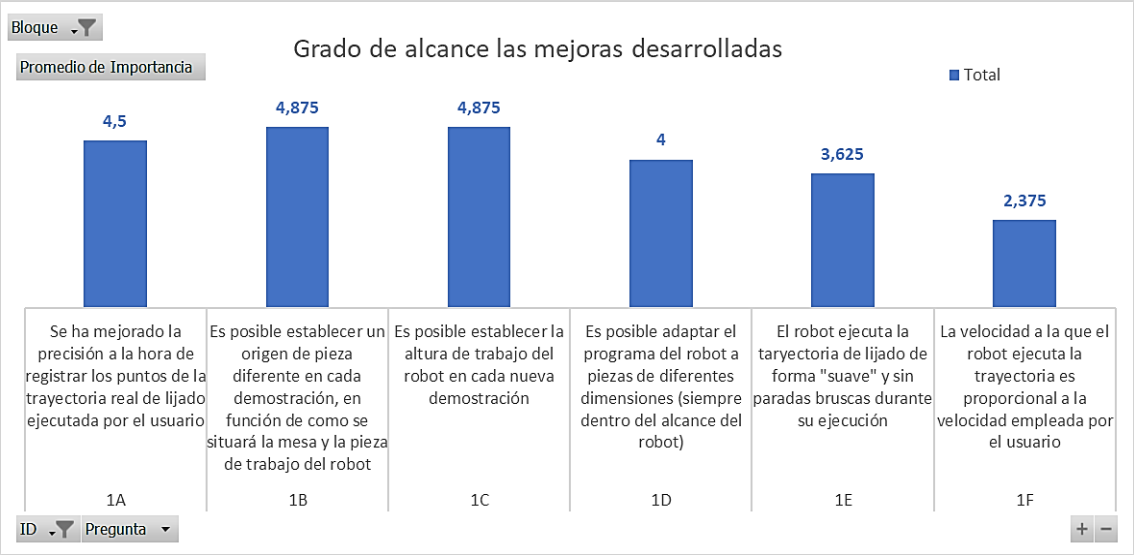


Figura 70 – Grado de alcance de mejoras desarrolladas
Fuente: elaboración propia

Las mejoras relacionadas con la "suavidad" de la trayectoria de lijado y la proporcionalidad de la velocidad proporcional a la ejecutada por el usuario presentan una puntuación más baja. Esta puntuación más baja se debe a los siguientes motivos:

- Un error de código a la hora de generar el script final, fijaba el parámetro "blend radius" a 0'001, haciendo que el robot se parara en cada punto de la trayectoria, independientemente del parámetro que escribiera en la interfaz Robotrack.

- El cálculo de la velocidad entre puntos se desvirtuaba al realizar el suavizado y el granulado, por lo que el robot ejecutaba la trayectoria de forma mucho mas lenta que el usuario.

Estos dos aspectos han sido corregidos tras la realización de las pruebas.

Se volvió a encuestar a los participantes en las pruebas de validación acerca de la importancia y grado de consecución de los objetivos, obteniendo los siguientes resultados.

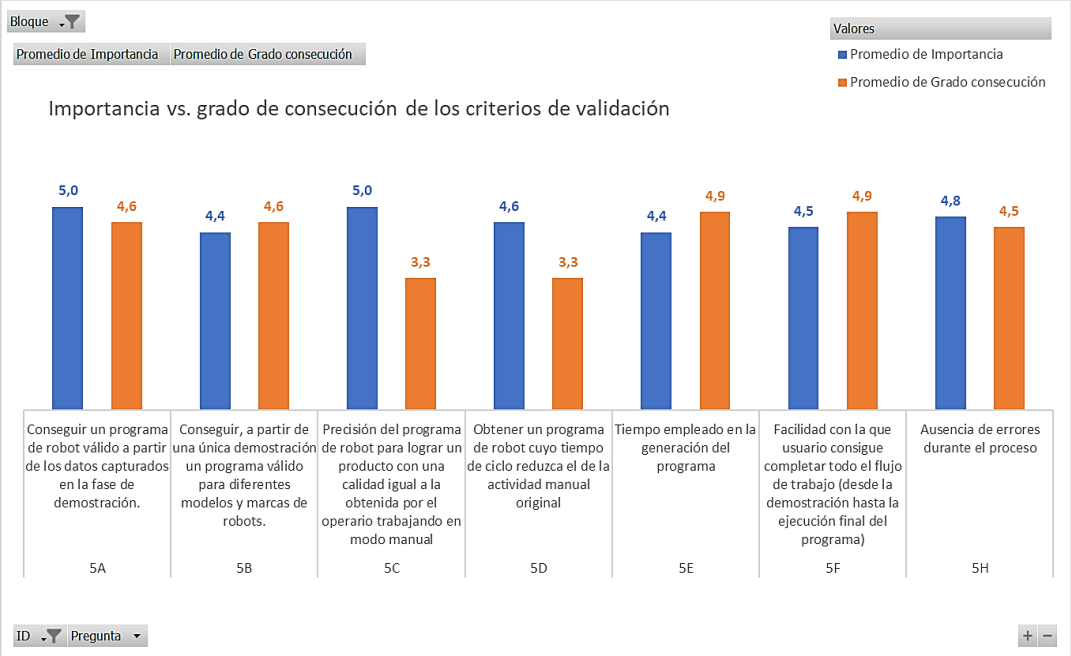


Figura 71 – Grado de alcance de KPIs – personal AIDIMME
Fuente: elaboración propia

Tras corregir los errores ligados al parámetro "Blend radius" y la velocidad de ejecución de la trayectoria por parte del robot, se realizan pruebas de validación con algunas de las empresas colaboradoras en el proyecto: IT8 SOFTWARE, HURTADO RIVAS y CFZ COBOTS.

Al igual que al personal técnico de AIDIMME, se les pide que rellenen los cuestionarios valorando los KPIs definidos.

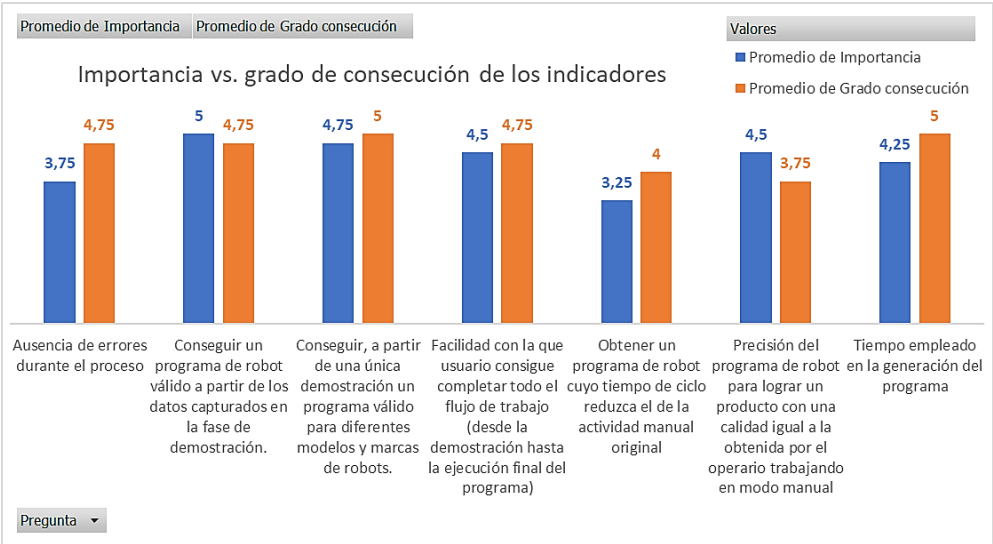


Figura 72 – Grado de alcance de KPIs – empresas colaboradoras
Fuente: elaboración propia

De forma general, dan por cumplidos todos los objetivos planteados. El KPI de menor promedio es la precisión del programa, que viene ligada por la precisión de la cámara de visión.

3. VALIDACIÓN DEL SISTEMA

3.1. Análisis

Tras las dos rondas de validación realizadas se han evaluado los KPIs identificados al inicio del proyecto como los de mayor relevancia para evaluar el resultado del proyecto. Tras las modificaciones realizadas, todos los indicadores han mejorado en la segunda ronda, respecto de la puntuación promedio obtenida en la primera ronda.

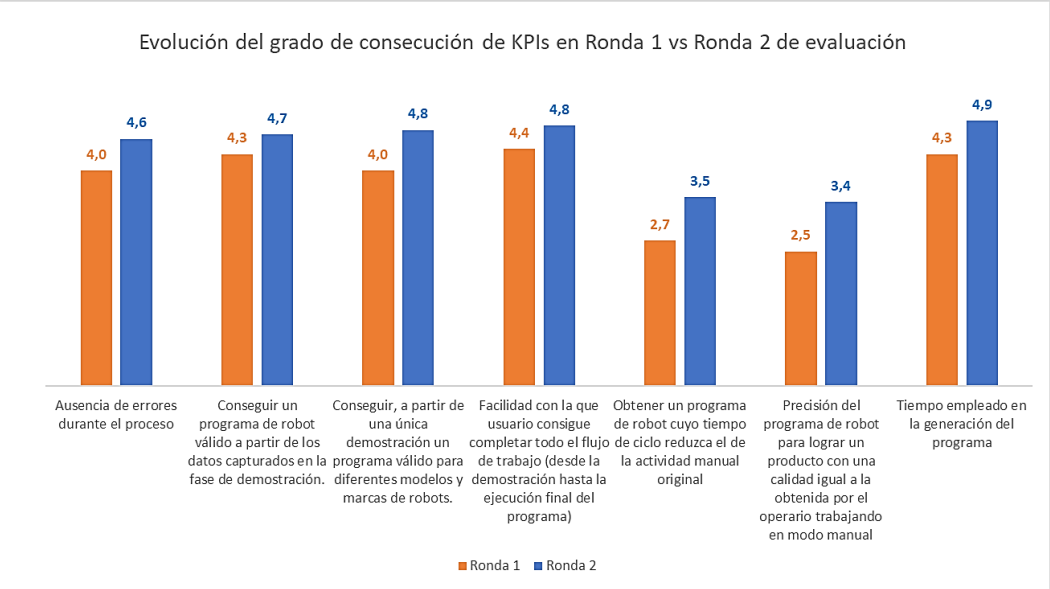


Figura 73 – Comparativa KPIs ronda 1 y ronda 2
Fuente: elaboración propia

Aunque existen dos indicadores que se han quedado por debajo de un promedio de 4, son justamente los indicadores donde mayor porcentaje de mejora se ha producido en la valoración de los usuarios.

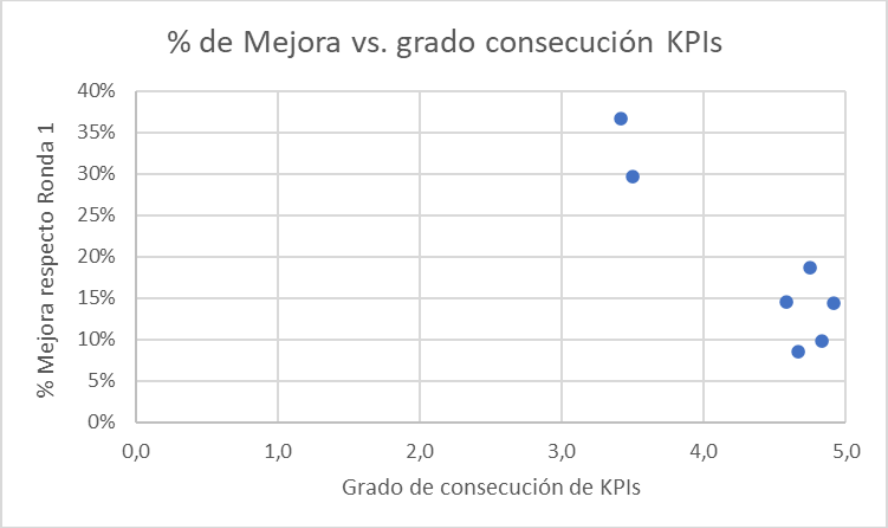


Figura 74 – Comparativa grado de consecución KPIs y el Grado de mejora
Fuente: elaboración propia

Por tanto, los resultados del proyecto se pueden dar por validados con un alto grado de puntuación por el conjunto de técnicos de AIDIMME, así como de empresas que han participado durante las pruebas de validación.

3.2. Propuesta de mejoras

Se ha identificado en algunas pruebas un error en la posición x,y capturada, respecto de la realmente ejecutada por el usuario. Esta situación se corresponde con momentos en los que el operario cruza los brazos. El sistema de visión llega a confundir durante breves lapsos de tiempo ambas manos, generando el error mencionado.

Figura 75 – Ejemplo en que usuario cruza las manos durante la demostración
Fuente: elaboración propia

Se propone formar al usuario que realice las demostraciones para que no ejecute movimientos cruzando brazos o escondiendo una de sus extremidades, de forma que el sistema de captura de datos pueda funcionar correctamente.

Por otro lado, se ha detectado que durante las demostraciones se requieren siempre de dos usuarios, uno que realiza la demostración y otro que inicia y finaliza la grabación de datos en la interfaz Robotrack. Sin la ayuda de esta segunda persona, el usuario que realiza la demostración generaría datos erróneos mientras manipula el ordenador.

Se propone mejorar la idea inicial del guante planteada en el Entregable E4.1 donde con un sistema de pulsadores, el usuario diera inicio y final a la grabación de datos se solventaría esta situación dotando de autonomía al usuario. Este guante podría disponer de más sensores que pudieran capturar datos de interés para el robot, como por ejemplo un sensor de presión para saber la fuerza que se aplica durante el lijado de piezas.

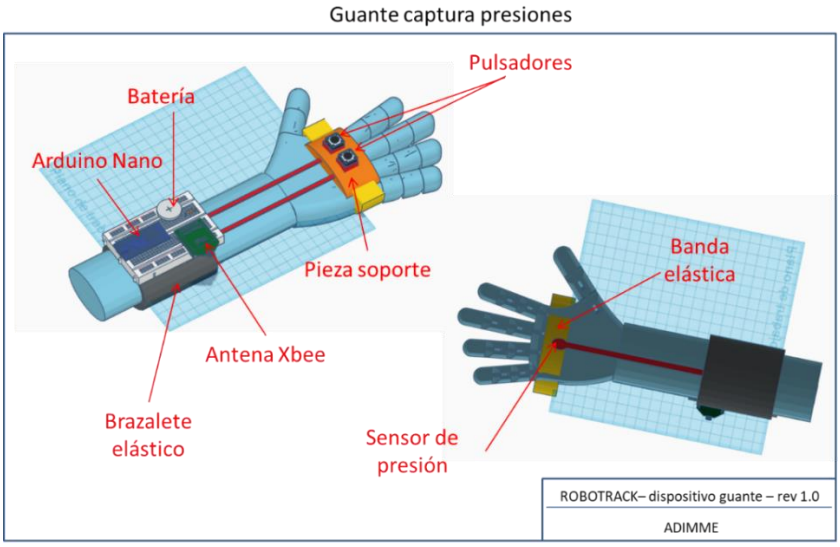


Figura 76 – Boceto conceptual del guante de captura de acciones de usuario (i)
Fuente: elaboración propia

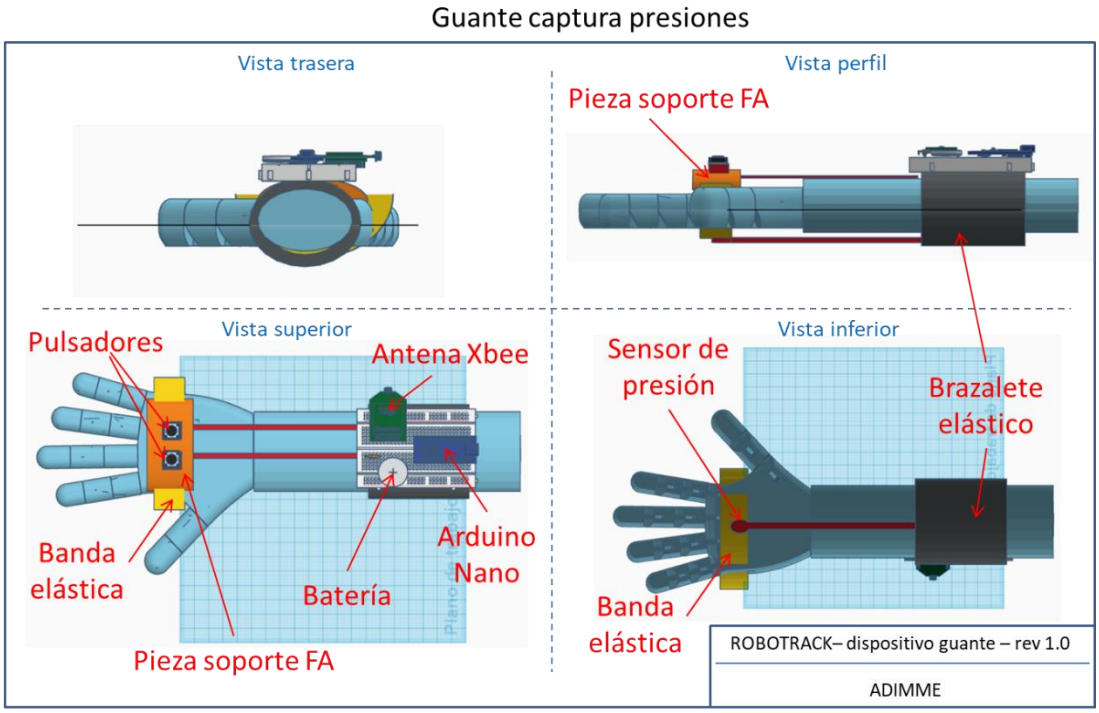


Figura 77 – Boceto conceptual del guante de captura de acciones de usuario (ii)
Fuente: elaboración propia

Se ha diseñado una placa electrónica que podría dar soporte a toda la electrónica requerida para implementar esta mejora.

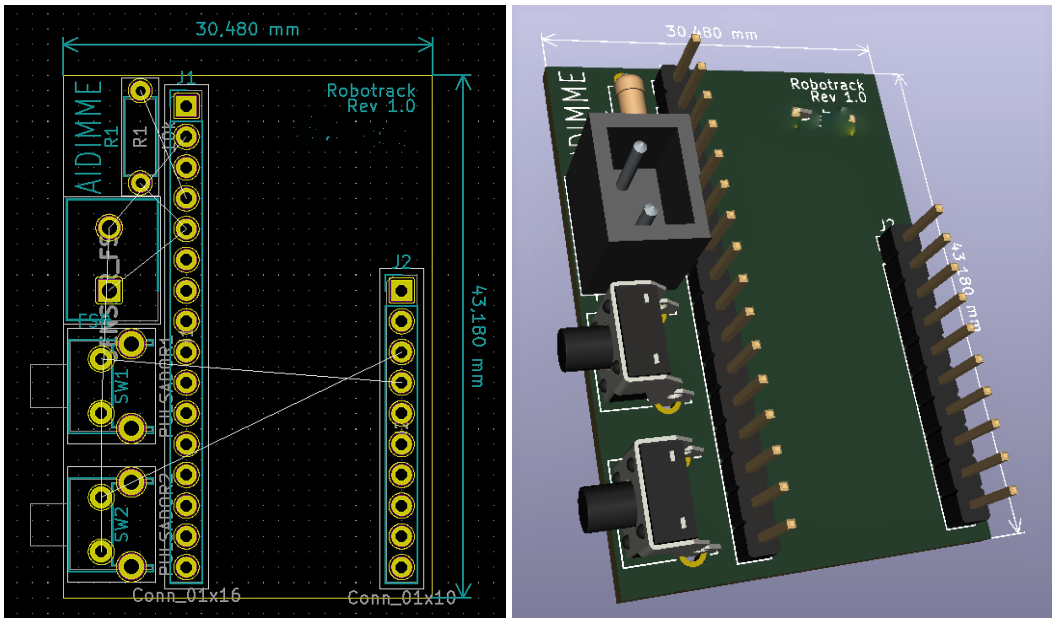
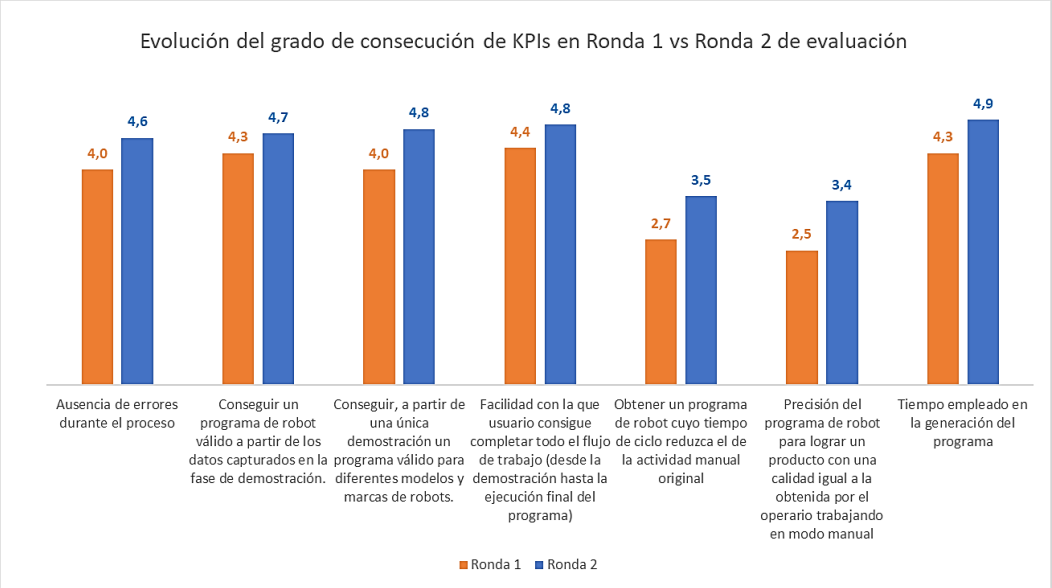


Figura 78 – Diseño de placa electrónica para guante Robotrack
Fuente: elaboración propia

C. RESUMEN Y CONCLUSIONES

Con el proyecto Robotrack se ha desarrollado un sistema que permite de manera sencilla e intuitiva generar un programa de robot en base a la demostración inicial de un operario experto en la ejecución manual a automatizar.

El resultado obtenido es la validación del sistema desarrollado en el proyecto. Se han llevado a cabo dos rondas de validación y tras las mismas, cinco de los indicadores superan una evaluación promedio de 4’5 (en una escala 1-5, donde 5 es la mejor puntuación) y los dos indicadores restantes del proyecto han sido valorados en promedio en el umbral del 3’5.



Comparativa KPIs ronda 1 y ronda 2
Fuente: elaboración propia

Las evaluaciones realizadas por las empresas colaboradoras en el proyecto han aportado información valiosa acerca de las acciones y actividades a desarrollar para acercar el prototipo inicial a las necesidades del mercado en cuando a programación de robots mediante demostración humana.

A la vista de los resultados obtenidos, el sistema Robotrack se ha mostrado como un prototipo de tecnología de aprendizaje por demostración capaz facilitar la programación de robots para procesos de lijado de piezas planas. Un solo operario en un periodo de entre 5-10 minutos puede generar, sin conocimientos previos de programación de robots, un programa válido para el modelo de robot que vaya a ser encargado de automatizar la tarea.

Con la colaboración de:

AIDIMME
INSTITUTO TECNOLÓGICO

Domicilio fiscal —
C/ Benjamín Franklin 13. (Parque Tecnológico)
46980 Paterna. Valencia (España)
Tlf. 961 366 070 | Fax 961 366 185

Domicilio social —
Leonardo Da Vinci, 38 (Parque Tecnológico)
46980 Paterna. Valencia (España)
Tlf. 961 318 559 - Fax 960 915 446

aidimme@aidimme.es
www.aidimme.es