

# INFORME PROYECTOS— 2023

**DESARROLLO DE UN SISTEMA DE GENERACIÓN DE TRAYECTORIAS  
COMPLEJAS EN ROBOTS BASADO EN APRENDIZAJE POR  
DEMOSTRACIÓN  
“ROBOTRACK”**

**Resumen de tareas realizadas y resultados obtenidos**

**Número de proyecto:** 22200073

**Expediente:** IMDEEA/2022/9

**Duración:** Del 01/01/2022 al 30/09/2023

**Coordinado en AIDIMME por:** JOSÉ LUIS SÁNCHEZ



## ÍNDICE

ÍNDICE.....	1
1. INTRODUCCIÓN. OBJETIVOS .....	1
2. ACTIVIDADES REALIZADAS .....	3
2.1- DESARROLLO DE INTERFAZ HMI PARA CAPTURA DE SECUENCIAS DE TRABAJO Y TRANSFERENCIA DE INFORMACIÓN.....	3
2.1.1 EVALUACIÓN DE NECESIDADES DE CAPTURA DE INFORMACIÓN.....	3
2.1.2 DESARROLLO DE SOLUCIONES PARA CAPTURA DE INFORMACIÓN.....	6
2.1.3- DESARROLLO DE LA INTERFAZ HMI .....	13
2.2- DESARROLLO DEL SISTEMA SOFTWARE DE CONVERSIÓN Y TRANSFERENCIA DE TRAYECTORIAS Y ACCIONES .....	19
2.2.1- DESARROLLO DEL SISTEMA DE ANÁLISIS DE TRAYECTORIAS Y ACCIONES .....	19
2.2.2- DESARROLLO DEL SISTEMA DE GENERACIÓN DE PROGRAMA DE ROBOT .....	21
2.2.3- INTEGRACIÓN DE UN SIMULADOR DE TRAYECTORIAS .....	23
2.2.4- VALIDACIÓN DE DESARROLLOS.....	25

<u>2.3- DESARROLLO DE UN DEMOSTRADOR.....</u>	<u>30</u>
<u>2.3.1- DISEÑO DEL DEMOSTRADOR .....</u>	<u>30</u>
<u>2.3.2- MONTAJE DEL DEMOSTRADOR Y PRUEBAS.....</u>	<u>34</u>
<u>2.3.3- VALIDACIÓN DEL SISTEMA.....</u>	<u>59</u>
<u>3. RESULTADOS Y CONCLUSIONES .....</u>	<u>62</u>

## 1. Introducción. Objetivos

Hoy en día es común el uso de robots en la industria manufacturera, especialmente para realizar tareas repetitivas y en producciones elevadas. El paradigma de estas aplicaciones se puede encontrar en las líneas de montaje de automoción: soldadura, pintura, ensamblaje de piezas, entre otras. También se ha incrementado el uso de sensores externos para adquirir información adicional del entorno de trabajo. En particular, las cámaras y los sistemas de visión artificial brindan flexibilidad y aumentan la precisión de las tareas realizadas por los sistemas robóticos.

Un campo de trabajo que se está explorando es el de la colaboración entre humanos y robots, de forma que se complementen las capacidades de ambos, para lo cual se impulsa el desarrollo de nuevas herramientas para facilitar la interacción entre robots y humanos, o incluso sólo entre robots. Hasta cierto punto, la interacción en la dirección robot – humano es relativamente sencilla dadas las capacidades cognitivas de las personas; la información generada por cualquier robot se puede presentar con múltiples formatos y las personas normalmente pueden comprenderla. Sin embargo, en sentido inverso es bastante más complicado interactuar.

Actualmente, uno de los mayores problemas de la industria es la programación de robots para la ejecución de tareas en entornos altamente cambiantes, bien sea porque cambia el producto (modelo) fabricado debido a la alta rotación requerida por el mercado, bien sea porque cambia el entorno productivo debido a la reconfiguración requerida por el proceso, o bien por otras circunstancias no previstas. Los métodos de programación tradicional, entre otros problemas, requieren un amplio conocimiento técnico por parte del personal responsable y consumen un tiempo considerable. De ahí surge la necesidad de facilitar el proceso de programación de operaciones industriales en robots, campo en el que AIDIMME está trabajando desde hace varios años.

Una de las técnicas con las que se está experimentando es la denominada Aprendizaje por Demostración (LbD, en sus siglas en inglés). Este método consiste principalmente en “observar” el desempeño de una tarea, extraer información relevante y generar acciones que puedan ser interpretadas y reproducidas por el controlador del robot. Para que sea fácilmente adoptable por la industria, este proceso de conversión debe ser rápido (sin intervención humana en la medida de lo posible), fiable y preciso, de manera que las tareas observadas se reproduzcan fielmente y sin errores.

Con estas premisas, el proyecto se planteó para avanzar en el desarrollo de técnicas que permitan agilizar la programación de robots aplicando técnicas de Aprendizaje por Demostración a partir de los resultados obtenidos en un proyecto anterior (*COLEARNING 4.0 - Desarrollo de técnicas de aprendizaje para COBOTS basadas en interacción humana y aprendizaje de refuerzo. IMDEEA/2020/22*), en el cual se consiguió que el robot reprodujera movimientos y presiones realizadas previamente por una persona, aunque de forma aproximada y con un tratamiento manual de los datos capturados.

Para profundizar en esta técnica de programación automática, el **objetivo principal del proyecto** era desarrollar un sistema de aprendizaje por demostración para robots colaborativos, basado en un sistema de visión artificial y sensores externos al robot, que permita reproducir con precisión tareas complejas de forma inmediata tras la demostración humana, utilizando

algoritmos de inteligencia artificial para el modelado de trayectorias y ejecución de tareas. La validación del desarrollo que se consiga se realizaría sobre la reproducción de un proceso con elevada carga de trabajo manual que implique el control de varios parámetros del robot, además de las trayectorias (velocidad, fuerza, par, etc).

## 2. Actividades realizadas

Seguidamente se detallan las tareas técnicas realizadas en el marco del proyecto, y los resultados alcanzados.

### 2.1- DESARROLLO DE INTERFAZ HMI PARA CAPTURA DE SECUENCIAS DE TRABAJO Y TRANSFERENCIA DE INFORMACIÓN AL CONTROL DEL ROBOT

#### 2.1.1 Evaluación de necesidades de captura de información.

Se realizó una revisión de procesos y proyectos de automatización mediante robots, permitiendo identificar aquellos de los sectores madera-mueble y metalmecánico donde mayor potencial presenta la automatización mediante robots.

##### Revisión procesos en el sector madera-mueble

Algunos de los procesos más susceptibles de automatización mediante robots que se han identificado son:

- **Ensamblaje:**  
Realización de montaje de piezas y ensamblajes repetitivos. Los robots pueden ser programados para unir piezas utilizando diferentes métodos de unión, como tornillos, clavos o cola.
- **Lijado:**  
Robots utilizados en procesos de lijado pueden trabajar de manera uniforme en superficies grandes o complejas, eliminando la necesidad de trabajo manual intensivo.
- **Pintura y acabado:**  
Robots de pintura automatizados pueden aplicar capas uniformes de pintura, barniz u otros acabados a piezas de muebles.
- **Embalaje y envío:**  
Robots de embalaje pueden empaquetar los muebles de manera eficiente y segura para su transporte.
- **Inspección y control de calidad:**  
Sistemas de visión artificial y sensores embarcados en un robot pueden realizar inspecciones automatizadas para detectar defectos en las piezas y asegurar la calidad del producto final.

##### Revisión procesos sector metalmecánico

En el sector metalmecánico, también existen varios procesos que son susceptibles de ser automatizados mediante el uso de robots. Algunos de los procesos más susceptibles de automatización identificados en este sector son:

- **Soldadura:**  
Robots de soldadura pueden llevar a cabo uniones de soldadura de manera constante y precisa en componentes metálicos, incluso en piezas de formas complicadas.
- **Pulido y acabado:**  
Robots de pulido automatizados pueden dar acabado a piezas metálicas, eliminando imperfecciones y mejorando la apariencia final.
- **Ensamblaje:**  
Robots pueden ensamblar piezas metálicas mediante procesos como remachado, atornillado y unión por adhesivos.
- **Inspección y control de calidad:**  
Sistemas de visión artificial y sensores embarcados en un robot pueden realizar inspecciones automatizadas para detectar defectos en las piezas metálicas y garantizar la calidad.
- **Manipulación de materiales:**  
Robots pueden mover y transportar piezas metálicas pesadas y voluminosas de manera segura y eficiente.
- **Embalaje y envío:**  
Robots de embalaje pueden empaquetar las piezas de manera eficiente y segura para su transporte.

#### Perspectiva empresas colaboradoras en el proyecto

También se recogió la perspectiva de las empresas colaboradoras del proyecto mediante reuniones al inicio del proyecto en la que se expusieron los objetivos de este, así como el abordaje que se estaba dando.

Como resultados de estas reuniones, los procesos que las empresas destacaron como potencialmente compatibles para ser automatizados mediante un robot programado mediante una herramienta de aprendizaje por demostración como la de Robotrack son:

- Procesos donde el robot tenga que seguir una trayectoria continua a lo largo de la superficie de una pieza: como por ejemplo procesos de soldadura, lijado o pintura.
- Procesos de embalaje o paletizado.
- Procesos de montaje o ensamblaje de piezas.

#### Perspectiva técnicos de automatización de procesos

Tras reuniones de presentación de objetivos y esbozo de los primeros diseños y desarrollos del proyecto Robotrack a diez técnicos de AIDIMME con experiencia en automatización de procesos y programación de robots, se les emitió un breve cuestionario cuyo objetivo era identificar procesos donde fuera factible la aplicación de los resultados del proyecto.

La pregunta formulada fue la siguiente: *“Valore la importancia o idoneidad de aplicar una*

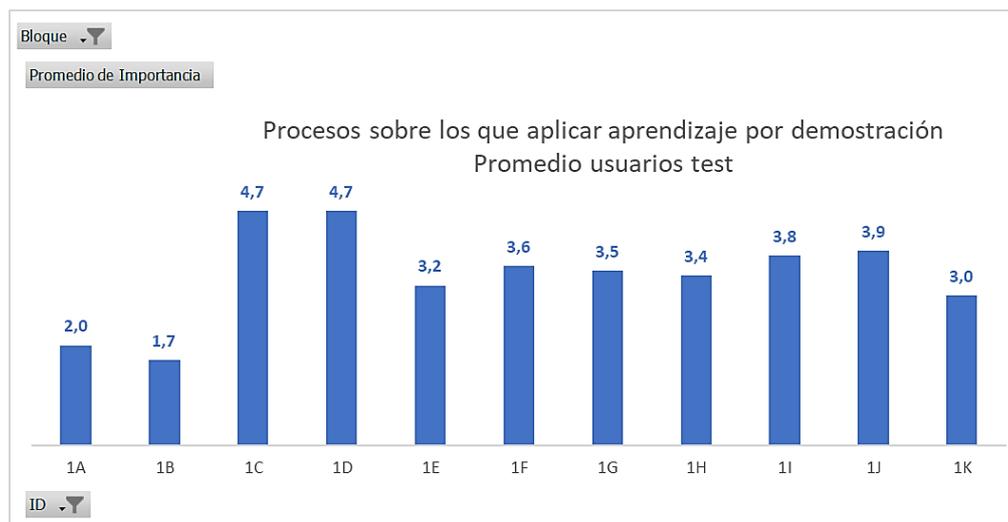
automatización con robots, mediante un proceso de aprendizaje por demostración para los siguientes procesos **(Valorar entre 1-5)**".

Se plantearon las siguientes opciones para valorar entre 1-5, siendo el 5 un proceso muy adecuado y el 1 un proceso nada adecuado. Las opciones de procesos planteadas fueron más amplias de las identificadas inicialmente, con el objetivo de no descartar ningún proceso.

ID	Pregunta
1ª	Corte de piezas:
1B	Mecanizado de piezas:
1C	Lijado de piezas:
1D	Acabado de piezas (pintura):
1E	Ensamblaje / montaje de piezas:
1F	Soldadura de piezas:
1G	Inspección de calidad (apoyado en sistema de visión u otros sensores):
1H	Empaquetado de piezas:
1I	Movimiento /manipulación de piezas entre dos zonas de trabajo:
1J	Alimentación de piezas a otras máquinas:
1K	Paletizado / despaletizado:

Preguntas formuladas sobre procesos a automatizar mediante demostración  
Fuente: elaboración propia

El resultado promedio de las valoraciones fue el que se muestra en la siguiente gráfica.



Resultado promedio de procesos sobre los que aplicar aprendizaje por demostración  
Fuente: elaboración propia

Como se puede comprobar, hubo consenso en identificar los procesos de lijado y pintura como los más adecuados para la aplicación de los resultados del proyecto, coincidiendo con

lo detectado en la revisión bibliográfica y lo indicado por algunas de las empresas participantes en el proyecto.

## 2.1.2 Desarrollo de soluciones para captura de información

### Identificación de parámetros relevantes

En un aprendizaje por demostración para generar un programa de robot, es importante registrar una variedad de parámetros y datos que permitan transferir al robot las acciones realizadas por un operario experto. Estos parámetros pueden variar según el tipo de tarea y el entorno en el que se esté realizando el aprendizaje.

Se identificaron algunos de los parámetros clave que podrían registrarse:

- **Posición y orientación de las manos del operario:**  
Coordenadas espaciales y orientación de las manos o extremidades del operario durante la demostración.
- **Velocidad y aceleración:**  
Velocidades y aceleraciones de las manos del operario durante la demostración. Control de la velocidad en diferentes puntos de la trayectoria.
- **Fuerza y presión:**  
Fuerzas y presiones ejercidas sobre las superficies o piezas durante la demostración. Ajustes de fuerza aplicados en diferentes momentos de la tarea.
- **Tiempo:**  
La secuencia temporal de los movimientos y acciones realizadas durante la demostración. Duraciones de diferentes etapas de la tarea.
- **Condiciones del entorno:**  
Información sobre el entorno en el que se realiza la tarea, como la disposición de objetos o obstáculos, la iluminación y las condiciones físicas.
- **Interacciones humanas:**  
Cualquier interacción entre el operario humano y el robot durante la demostración. Información sobre cómo el operario maneja la herramienta o interactúa con el entorno.
- **Eventos y desencadenantes:**  
Eventos que pueden desencadenar acciones específicas durante la tarea. Información sobre cuándo se aplican ciertos movimientos o ajustes.
- **Contexto y explicaciones:**  
Explicaciones verbales o anotaciones visuales proporcionadas por el operario para comprender la tarea. Contexto sobre la intención detrás de ciertos movimientos o

decisiones.

Al igual que se hizo con los procesos que potencialmente podían ser automatizados mediante una solución de aprendizaje por demostración, se ha testado la opinión de diez técnicos de AIDIMME expertos en robótica y automatización de procesos para identificar aquellos parámetros del proceso de demostración que se consideran de mayor relevancia y que deben ser capturados por dispositivos externos al robot.

La pregunta formulada fue la siguiente: *“Valore la importancia de capturar información de (Valorar entre 1-5)”*.

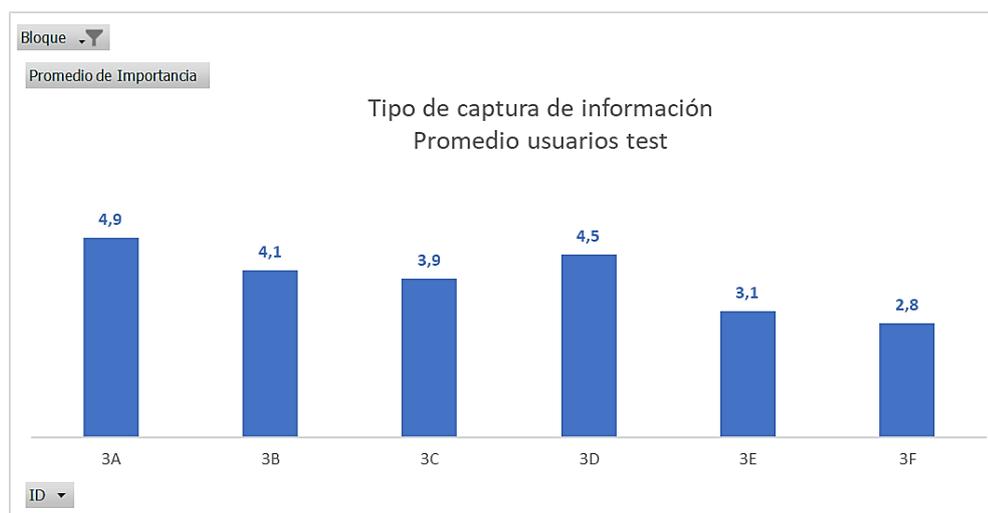
Las opciones de procesos planteadas fueron las siguientes:

ID	Pregunta
3A	Movimiento de las manos del operario:
3B	Acciones o gestos concretos de las manos (abrir, cerrar, girar, apretar, etc.):
3C	Movimiento de la pieza:
3D	Movimiento de una herramienta de trabajo:
3E	Condiciones del entorno de trabajo (luminosidad, temperatura, nivel de ruido etc.):
3F	Condiciones de trabajo de otras máquinas o elementos del proceso (velocidad de una cinta, contador de piezas, etc.):

Preguntas formuladas sobre los parámetros a capturar durante el proceso de demostración.

Fuente: elaboración propia

El resultado promedio de las valoraciones fue el que se muestra en la siguiente gráfica.



Resultado promedio de parámetros que se considera importante capturar

Fuente: elaboración propia

El parámetro con que hay prácticamente consenso absoluto es el movimiento de las manos del operario. Como alternativa al movimiento de las manos se encuentra el movimiento de

la herramienta de trabajo, o las acciones concretas de las manos como abrir, cerrar, girar, etc.

De todos los posibles parámetros a registrar durante el proceso de demostración humana, los desarrollos del presente proyecto se centran en el registro del movimiento y acciones de las manos.

### Identificación sistemas para la captura de la posición de las manos

#### Kinect de Xbox 360

La cámara Kinect para Xbox 360 es un accesorio desarrollado por Microsoft para su consola de videojuegos Xbox 360. Lanzado en 2010, Kinect permite a los jugadores interactuar con los juegos sin necesidad de utilizar un controlador físico. En lugar de eso, la cámara Kinect utiliza una combinación de sensores, cámaras y software para rastrear los movimientos del cuerpo y reconocer comandos de voz.



Cámara Kinect de Xbox (i)

Fuente: <https://es.wikipedia.org/wiki/Kinect>

#### Intel Realsense

La cámara Intel RealSense es una serie de cámaras y sensores desarrollados por Intel que están diseñados para capturar información tridimensional y permitir la interacción entre humanos y computadoras de una manera más natural y envolvente. Estos dispositivos utilizan una combinación de cámaras RGB (captura de color), sensores de profundidad y software de procesamiento para crear un entorno tridimensional del mundo real y rastrear los movimientos y gestos de los usuarios.



Cámara Intel Realsense

Fuente: <https://es.rs-online.com/web/p/camaras-de-profundidad/1720981>

Las cámaras Intel RealSense se utilizan en una variedad de aplicaciones, que van desde la realidad virtual y aumentada hasta la robótica y la automatización industrial.

#### Guante senso DK3

Senso Glove es un guante inalámbrico que permite reemplazar a los controladores tradicionales de videojuegos. Con los sensores IMU, se pueden reproducir los movimientos

de las manos. Dispone de motores de vibración LRA que le permiten obtener retroalimentación háptica en aplicaciones y juegos.

Senso Glove es compatible con aplicaciones y juegos SteamVR, pero con la ayuda del SDK de código abierto para Unity y Unreal Engine.



Guantes Senso Glove  
Fuente: <https://senso.me/dev>

### Identificación sistemas para la captura de posición de un objeto o pieza

Una segunda opción en vez de capturar información del movimiento del operario, es capturar información de la posición de la pieza o herramienta de trabajo que va a ser manipulada.

Para capturar información de las coordenadas  $x$ ,  $y$ ,  $z$  de un objeto en un entorno tridimensional, se requiere el uso de sensores o tecnologías de seguimiento de movimiento que puedan medir la posición en el espacio. Las opciones identificadas para ello son las siguientes:

- Sensores de Profundidad y Cámaras 3D. Estas cámaras pueden ofrecer una precisión en el rango de centímetros.
- Sistemas de Captura de Movimiento. La precisión de los sistemas de captura de movimiento puede variar según el número de cámaras, la resolución de las cámaras y la calidad de los marcadores utilizados. En configuraciones estándar, se pueden lograr precisiones en el rango de milímetros a centímetros.
- Giroscopios y Acelerómetros. Los sensores de giroscopio y acelerómetro pueden proporcionar información sobre el movimiento y la orientación del objeto, pero debido a la acumulación de errores en el tiempo, la precisión puede disminuir. La precisión puede variar desde unos pocos grados hasta fracciones de grado, dependiendo de la calidad de los sensores y el sistema de filtrado utilizado.
- Tecnologías de Realidad Virtual y Aumentada. Los sistemas más avanzados pueden ofrecer precisiones en el rango de milímetros a centímetros.

- Sistemas de Posicionamiento Global (GPS). La precisión del GPS varía según la cantidad de satélites visibles y la calidad del receptor GPS. En condiciones ideales, el GPS puede proporcionar precisiones en el rango de metros.

### Pruebas y desarrollo de sistemas

#### Sistemas para la captura de la posición de las manos

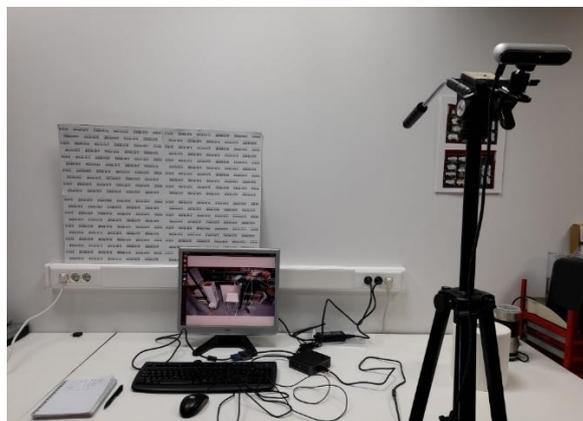
##### Basado en cámara Intel

Se ha desarrollado una aplicación para la cámara “Intel Realsense Depth Camera D455” que permita capturar la posición de las manos de un operario durante la demostración.

La aplicación desarrollada parte de experiencias previas del equipo de trabajo de AIDIMME empleando el Kit de desarrollo (SDK) “Cubemos Skeleton Tracking”. Se trata de un sistema IA en concreto de Deep Learning para el reconocimiento y seguimiento de personas en un ambiente. Permite un máximo de 5 personas y 18 puntos por persona.

Permite grabar la información identificada de los cuerpos y sus articulaciones en ficheros de texto con formato CSV para su posterior tratamiento.

En la siguiente imagen se muestra el montaje de la cámara, conectada al ordenador, monitor, teclado y ratón que permiten al usuario interactuar con el sistema.

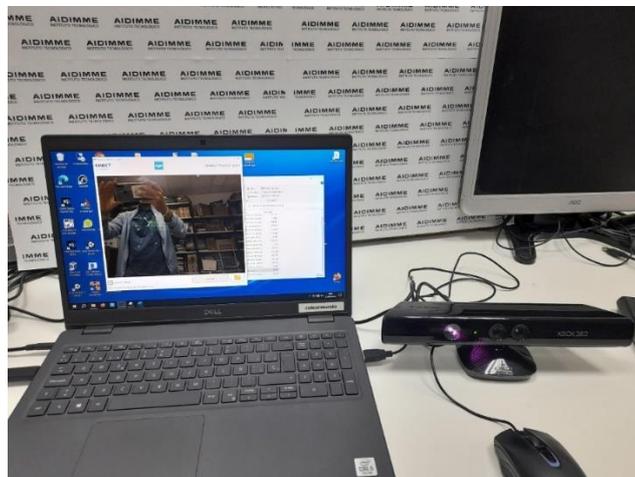


Montaje de la cámara Intel Realsense  
Fuente: elaboración propia

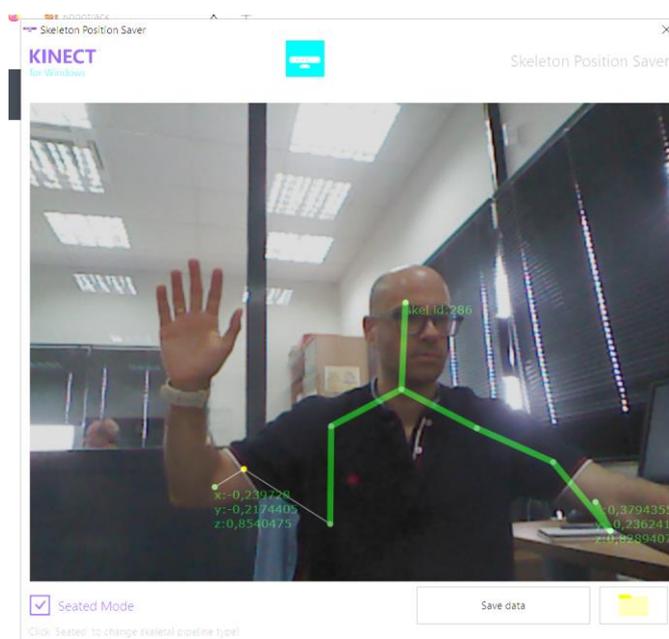
##### Basado en cámara Kinect

A partir del Kinect SDK se ha desarrollado una aplicación que realiza el “tracking” de las articulaciones del cuerpo humano, registrando las posiciones x, y, z de cada una ellas, basada en el “skeleton basis” del SDK.

La aplicación desarrollada permite guardar los datos en un fichero .csv para su posterior tratamiento por la HMI del proyecto.



Montaje de la cámara Kinect  
Fuente: elaboración propia



Aplicación de prueba de cámara Kinect  
Fuente: Skeleton Position Saver

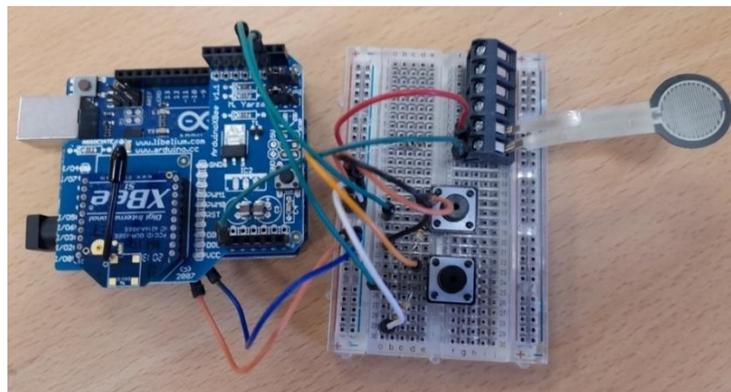
El número de articulación va de la 0 a la 19, y cuando no hay lectura de una articulación el sistema devuelve 0.

### Desarrollo de sistemas para la captura de acciones con las manos

Tal y como se ha identificado en apartados anteriores además de la posición de las manos (capturando posiciones x, y, z de estas), puede ser necesario recoger información de aspectos como la fuerza que se realiza con la mano o el momento concreto en que se agarra, suelta o mueve una pieza.

Para ello, se ha diseñado un dispositivo que permita mediante un sensor medir la fuerza de presión que ejerza la mano del operario durante la fase de demostración. Además, se dispone de dos pulsadores de señal digital, que permitan capturar alguna acción puntual cuyo momento temporal el usuario quisiera registrar.

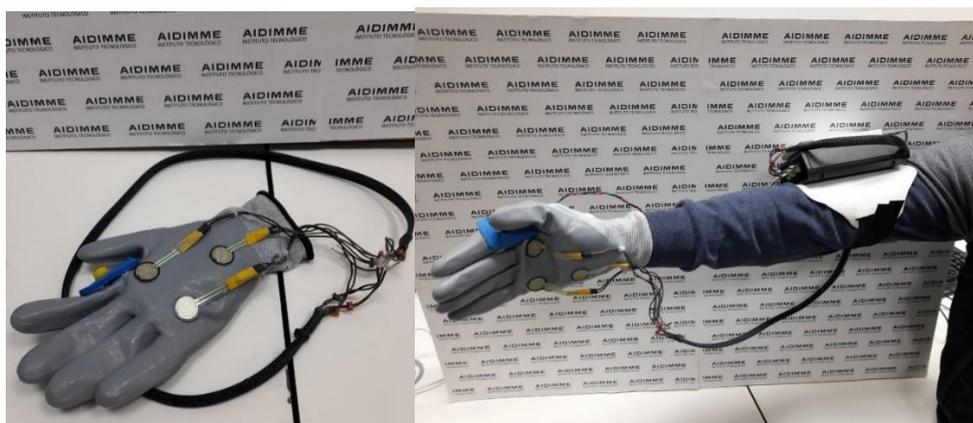
Se montó un prototipo inicial basado en la placa Arduino, con un sensor de fuerza FSR y dos pulsadores, así como una antena Xbee para la transmisión inalámbrica de datos.



Prototipo inicial del guante de captura de acciones de usuario  
Fuente: elaboración propia

Los datos registrados se guardan en un fichero .txt donde se registra el timestamp, el valor del sensor de fuerza, y los dos pulsadores digitales.

Se montó un prototipo de dispositivo en un guante añadiendo dos sensores de fuerza más y una batería, tal como se muestra en la imagen.



Prototipo inicial instalado en el guante de captura de acciones de usuario  
Fuente: elaboración propia

### 2.1.3- Desarrollo de la interfaz HMI

#### Definición de requisitos y funcionalidades

Dentro de la interfaz HMI del proyecto Robotrack el usuario debe de poder realizar 6 tareas principales: (1) Capturar (grabar) datos del proceso que se va a automatizar mediante el robot, (2) Revisar los datos grabados del proceso, (3) Crear una trayectoria optimizada a partir de los datos grabados, (4) Generar un programa de robot válido, (5) Generar una simulación de la trayectorias y acciones a llevar a cabo por el robot, y (6) Transferir el programa al robot.



Tareas a ejecutar dentro de HMI Robotrack  
Fuente: elaboración propia

Para cada tarea, las funcionalidades que inicialmente se identificaron necesarias para la interfaz Robotrack son las siguientes

#### (1) Capturar datos del proceso

- US1.1.- Iniciar un nuevo proceso de aprendizaje por demostración
- US1.2.- Seleccionar sistemas de captura de datos a utilizar
- US1.3.- Iniciar la grabación de datos
- US1.4.- Finalizar la grabación de datos
- US1.5.- Guardar el archivo/s de datos

#### (2) Revisar datos grabados en bruto

- US2.1.- Revisar información básica de la grabación

#### (3) Crear trayectoria optimizada

- US3.1.- Escribir los parámetros del algoritmo que genera la nueva trayectoria
- US3.2.- Visualizar el resultado de la trayectoria generada
- US3.3.- Validación de la trayectoria generada

#### (4) Generar programa de robot

- US4.1.- Visualizar que se ha generado el archivo Python genérico
- US4.2.- Visualizar que se ha generado el script de programa de robot específico
- US4.3.- Validación del programa de programa de robot

#### (5) Generar simulación

- US5.1.- Visualizar que se ha generado el archivo de la simulación
- US5.2.- Visualizar la simulación
- US5.3.- Validación del programa de programa de robot

#### (6) Transferir programa a robot

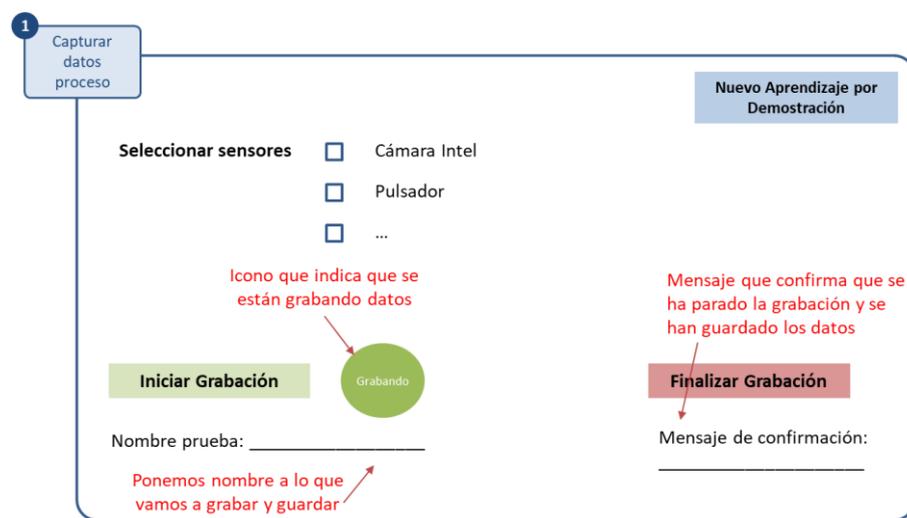
- US6.1.- Transferir programa al robot

### Diseño interfaz HMI

A continuación, se exponen algunas mockups del diseño inicial de la HMI para cumplir con las funcionalidades indicadas.

#### i. Capturar datos del proceso

Se plantea una pantalla inicial donde seleccionar el dispositivo de captura de datos empleado, así como botones o pulsadores para iniciar y finalizar la grabación de datos.

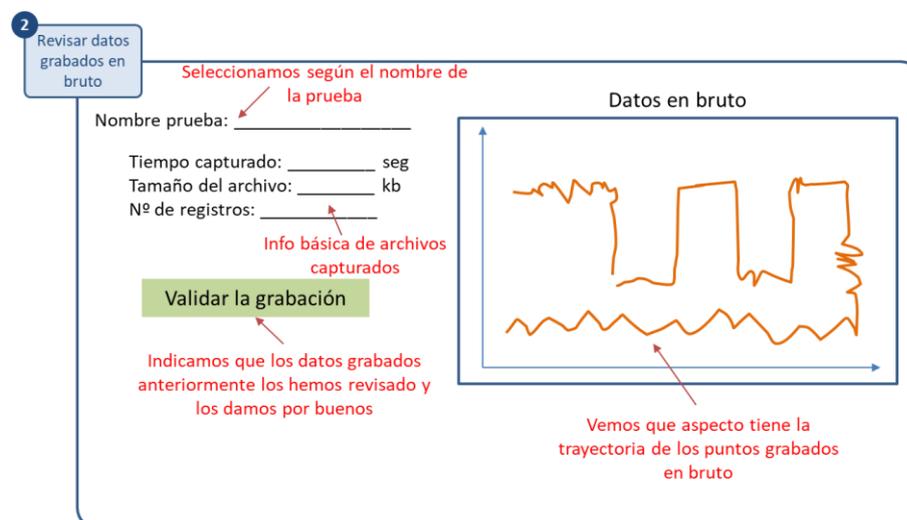


Mockup del proceso de captura de datos

Fuente: elaboración propia

#### ii. Revisar datos grabados en bruto

Se propone una pantalla donde visualizar los datos grabados por el usuario en forma de gráfica donde ver la trayectoria de las manos mientras el operario ejecuta la actividad.

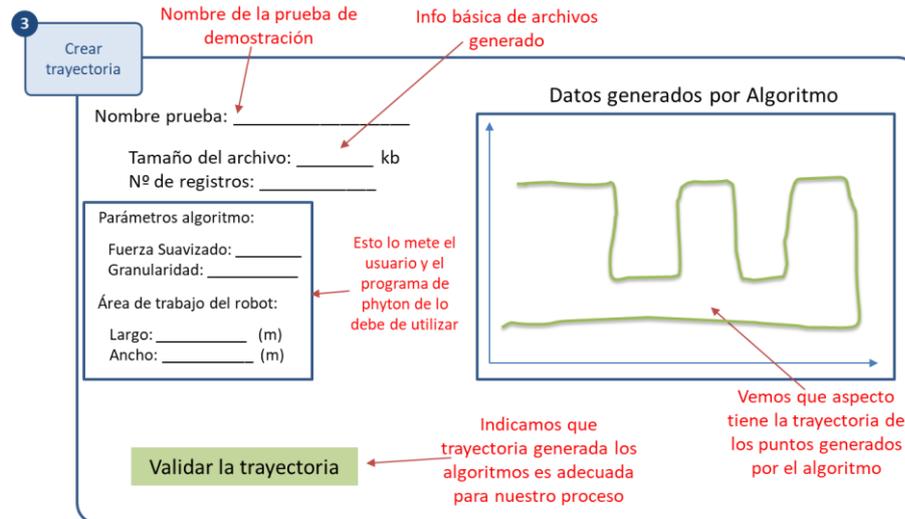


Mockup del proceso de revisión de datos grabados

Fuente: elaboración propia

iii. Crear trayectoria

Se propone una pantalla donde generar una trayectoria optimizada, suavizada que elimine el ruido inicial de los datos.

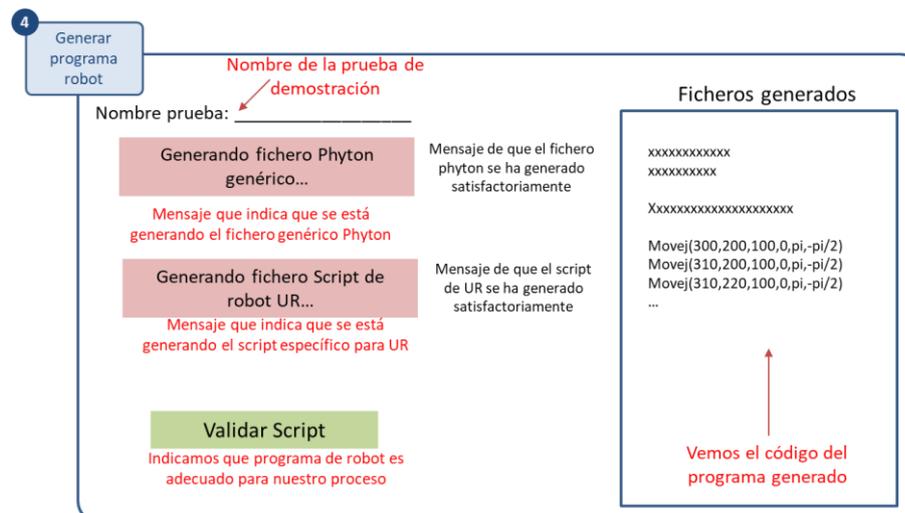


Mockup del proceso de generación de trayectoria de robot

Fuente: elaboración propia

iv. Generar programa robot

Se propone una pantalla con pulsadores donde iniciar la generación del script final del programa de robot, y poder visualizar el propio código generado.



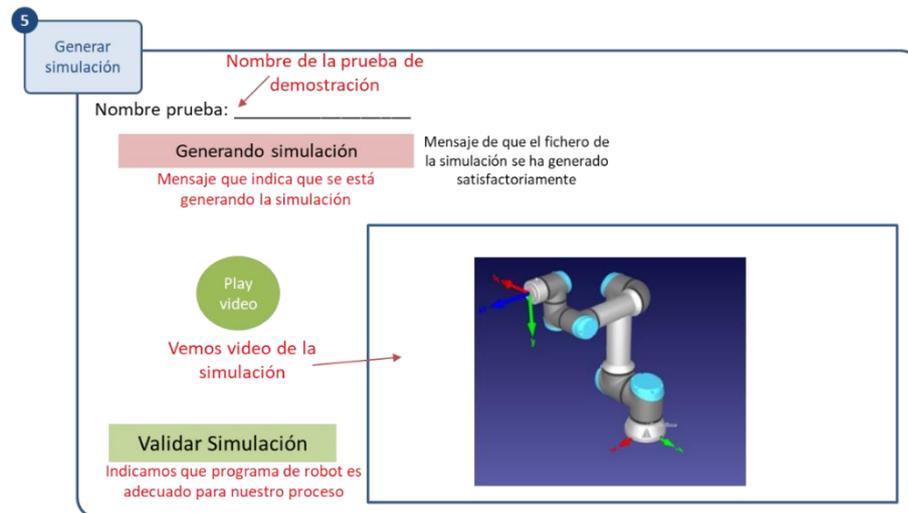
Mockup del proceso de generación del programa de robot

Fuente: elaboración propia

v. Generar simulación

Se propone una pantalla con pulsadores donde iniciar la generación de la simulación de la

trayectoria del programa de robot, y poder visualizarla.

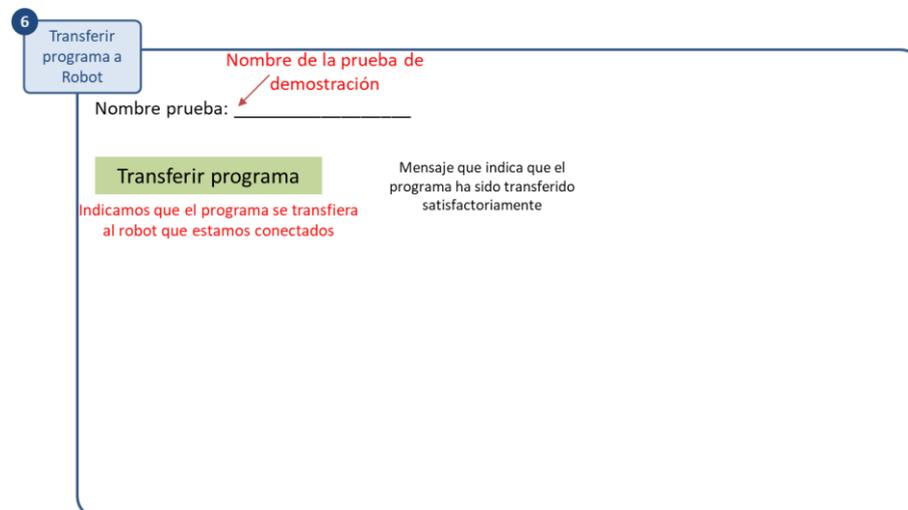


Mockup del proceso de generación de simulación de trayectoria

Fuente: elaboración propia

vi. TRANSFERIR PROGRAMA A ROBOT

Se propone una pantalla donde poder iniciar la transferencia del programa al robot.



Mockup del proceso de transferencia del programa de robot

Fuente: elaboración propia

**Desarrollo interfaz HMI**

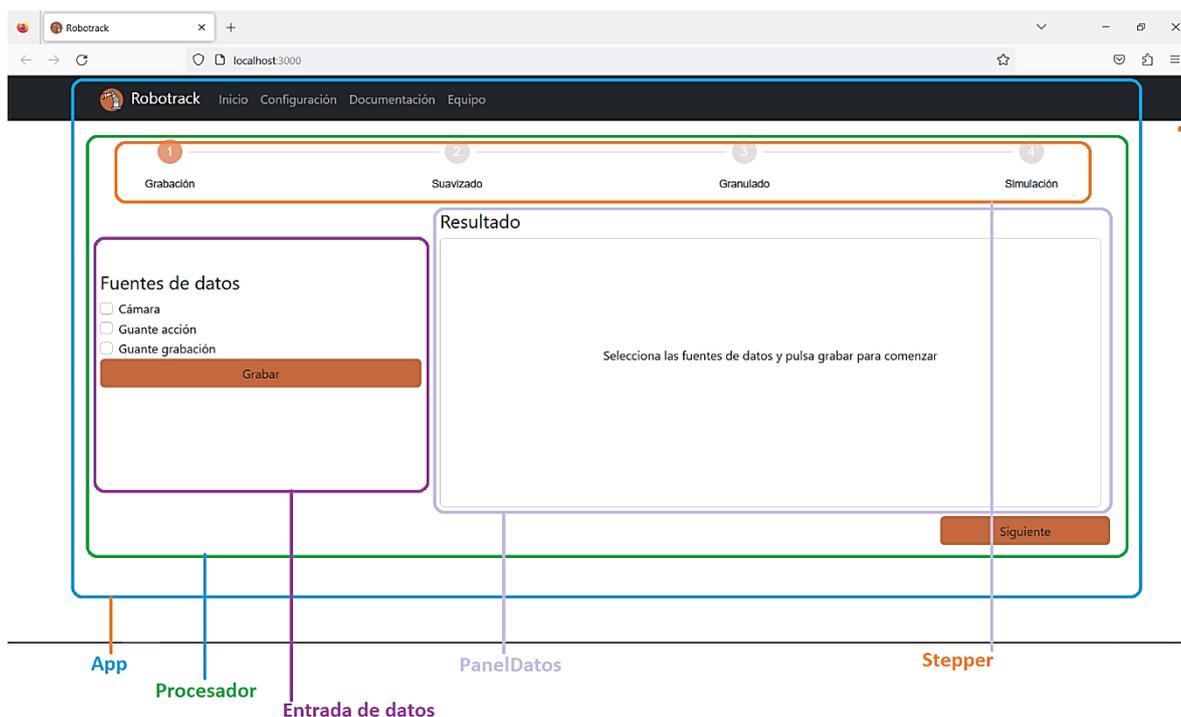
El desarrollo del proyecto requiere de la creación de un sistema que sea capaz de permitir al usuario una interacción simple pero efectiva con el flujo de trabajo necesario para la creación del programa robot. Bajo esta premisa, se ha optado por la creación de un sistema de aplicación web basada en 2 subsistemas conocidos habitualmente como frontend y backend.

### Frontend

El frontend se ejecuta en un navegador, y por tanto, utiliza los lenguajes que este conoce, es decir, el lenguaje de marcado HTML5, las hojas de estilo en cascada CSS y el lenguaje de programación JavaScript, abreviado habitualmente como JS.

La aplicación desgrana el proceso de creación del programa del robot en 4 fases dentro del componente de procesador: Grabación, Suavizado, Granulado y Simulación. Las 3 primeras poseen un componente de entrada de datos propio para cada caso, y comparten un componente de visualización de datos llamado PanelDatos, mientras que la última, dada su naturaleza diferente, posee un panel para visualizar la simulación del movimiento generada por el sistema y un panel para la configuración y descarga del programa de robot generado en el proceso.

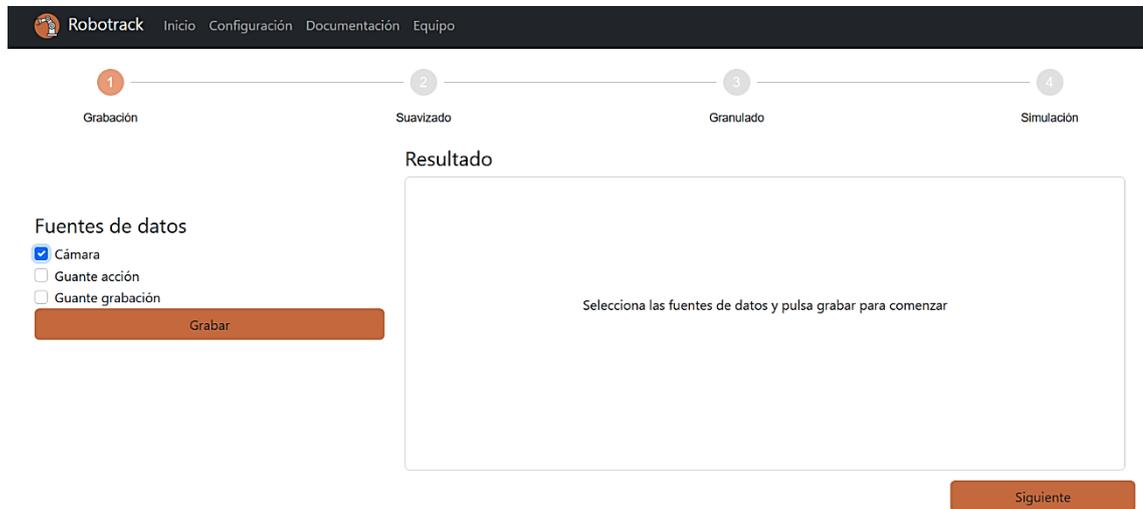
Se muestra a continuación una imagen que desgrana la jerarquía de componentes existente en la interfaz desarrollada.



Jerarquía de componentes de la aplicación  
Fuente: elaboración propia

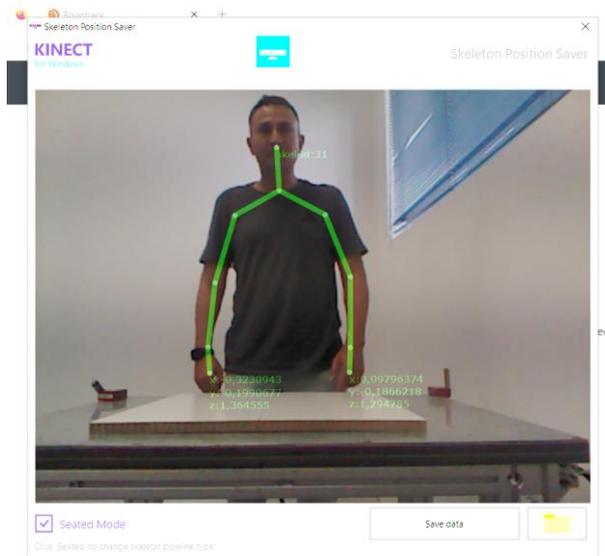
### Integración de sistemas de captura de datos en el Frontend

Desde la pantalla inicial de la aplicación, el usuario de esta puede seleccionar el sistema de captura de datos que va a ser utilizado. En el ejemplo de la siguiente figura se ha seleccionado la cámara que captura la posición de la mano derecha.



Selección sistema de captura de datos en la HMI  
Fuente: elaboración propia

Cuando se pulsa en el botón “Grabar” se ejecuta el programa de la cámara Kinect. Se abre una ventana nueva donde se ve el usuario y las coordenadas de posición de varias articulaciones del cuerpo.



Funcionamiento del sistema de captura de posición de las manos  
Fuente: elaboración propia

Cuando se pulsa el botón “Save data” los datos de posición de las articulaciones se comienzan a grabar. Cuando se pulsa “Finalizar” se termina la grabación y los datos se guardan. En la parte derecha de la interfaz, aparece un gráfico x-y de los datos grabados sobre la posición de la mano derecha del usuario.



Visualización de los datos de posición de la mano derecha

Fuente: elaboración propia

### Backend

El caso del backend generalmente se dispone de mayor libertad en la elección de tecnologías de desarrollo, ya que casi cualquier lenguaje puede satisfacer las necesidades de escucha y respuesta de peticiones HTTP necesarias para el desarrollo de este tipo de sistemas. Además, muchos lenguajes tienen frameworks de desarrollo backend que facilitan enormemente las tareas de desarrollo. Gracias a esto, generalmente el equipo de desarrollo puede elegir entre tecnologías como PHP con Laravel o Symfony, Java con Spring, JavaScript con Express o Python con Django o Flask entre otros.

Sin embargo, para este proyecto en particular, es necesario que el backend se integre con el programa RoboDK para generar tanto las simulaciones de funcionamiento del robot como el propio código de robot en sí mismo. RoboDK incluye la integración con el lenguaje de programación Python mediante la librería `robodk`, lo que obliga a usar este lenguaje para el desarrollo.

Gracias a Flask, el código backend se puede estructurar de manera simple en un solo archivo utilizando unas estructuras conocidas en Python como decoradores para definir las rutas y los tipos de peticiones aceptadas por cada endpoint. Para cada toma de datos, el Backend generará un token único y generará un archivo de datos utilizando como nombre ese token para cada paso del proceso, de esta forma el frontend puede modificar en cualquier momento los elementos generados accediendo a través del token identificador.

## 2.2- DESARROLLO DEL SISTEMA SOFTWARE DE CONVERSIÓN Y TRANSFERENCIA DE TRAYECTORIAS Y ACCIONES

### 2.2.1- Desarrollo del sistema de análisis de trayectorias y acciones

El sistema de análisis de trayectorias y acciones está compuesto por los pasos de suavizado y granulado de los datos capturados del movimiento de la mano derecha durante el proceso de demostración humana.

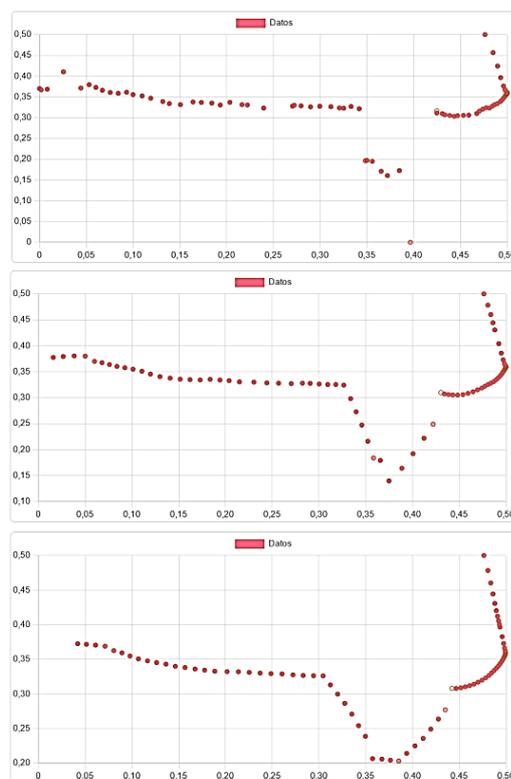
### Suavizado de datos

El objetivo de esta etapa es eliminar las posibles imperfecciones que por distintas razones puedan aparecer en la trayectoria ejecutada, como la imprecisión de la cámara, el propio movimiento del operario o cualquier otro posible defecto producido en el proceso de toma de datos, así como distribuir uniformemente los puntos tomados para facilitar la tarea al robot.

Para ello, se ha programado un algoritmo basado en medias móviles, que permite en base al número de periodos que el usuario de la interfaz desee, obtener una trayectoria suavizada que elimine los errores propios del proceso de captura de datos.

El usuario puede elegir la fuerza con la que se va a aplicar el suavizado mediante un formulario. El frontend entonces envía una petición HTTP al backend en la que incluirá el dato de fuerza y el token de datos. El backend crea un archivo de datos suavizado aplicando una media móvil a los datos originales, extrayendo la cantidad de puntos sobre los que promediar a partir del valor de fuerza proporcionado por el frontend. Por último, el backend devuelve los datos suavizados al frontend que los muestra al usuario. Éste tendrá la posibilidad de suavizar con otro valor de fuerza o bien pasar a la etapa siguiente.

En la siguiente imagen se muestra un ejemplo de trayectoria en bruto registrada por el sistema de captura de datos, así como el resultado de la fase de suavizado aplicando el algoritmo de medias móviles para cinco y diez periodos respectivamente.



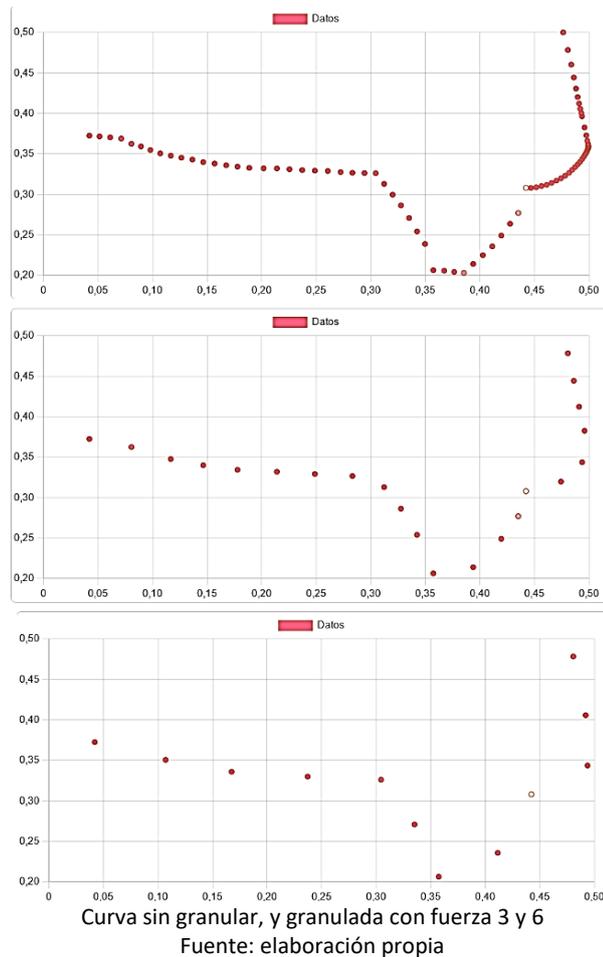
Curva sin suavizar, suavizada con 5 puntos y con 10

Fuente: elaboración propia

### Granulado de datos

El proceso de granulado reduce el número de puntos presentes en la muestra dejando los más representativos para correcta ejecución de la trayectoria. El objetivo de este proceso es reducir los parones que puede sufrir el robot al llegar a cada punto incluido en la muestra, así como simplificar el programa de robot.

El proceso es muy similar al caso anterior, el frontend manda una petición al backend, este recibe el token y la fuerza de granulado y se ocupará de crear un nuevo archivo. En este caso, el sistema buscará eliminar aquellos puntos que no tengan la distancia suficiente del punto anterior, obteniendo esta distancia de la fuerza de granulado. En la imagen se puede ver el resultado de aplicar un granulado de fuerza 3 y 6 al conjunto de datos introducido anteriormente.



## 2.2.2- Desarrollo del sistema de generación de programa de robot

Para el buen funcionamiento de Robotrack se requiere la definición de algunos parámetros cuyo objetivo es ajustar el funcionamiento de la aplicación a distintas situaciones y

casuísticas, que permitirán la generación final del programa de robot. En esta sección se revisa el funcionamiento de estas variables y el efecto que produce su modificación.

**Offset.** Las variables de offset, permiten ajustar la distancia desde la base del robot a la que va a operar el robot. El robot toma como punto (0,0,0) su propia base, por lo que, si no hay un ajuste de distancia, este fallará al chocarse consigo mismo. La variable “*offsetx*” permite señalar el desplazamiento en el eje x necesario para marcar el origen de coordenadas real y por su parte la variable “*offsety*” realiza lo propio en el eje y.

**Escala.** Las variables de escala permiten ajustar las diferencias de perspectivas existentes entre la toma de datos mediante el dispositivo de captura. De esta forma, la variable “*escalax*” realizará un escalado lineal con el valor indicado de todos los puntos en el eje x, por su parte la variable “*escalay*” se ocupa del otro eje.

**Valor Z.** El valor Z indicará al robot la altura a la que debe trabajar.

**Factor de velocidad.** Robotrack ajustará la velocidad de ejecución del robot dinámicamente dependiendo de la velocidad a la que haya realizado el operario que grabó los movimientos. Sin embargo, es posible ajustar el factor base desde el que se calcula esta velocidad de movimiento en cada punto.

**Blend Radius.** Se trata de un parámetro que marca un radio de curvatura a la hora de aproximarse a cada punto de la trayectoria desde el punto anterior de la misma. Este radio de curvatura hace que los movimientos del robot sean más suaves y fluidos. Si este parámetro está a cero, el robot se para en cada punto de la trayectoria.

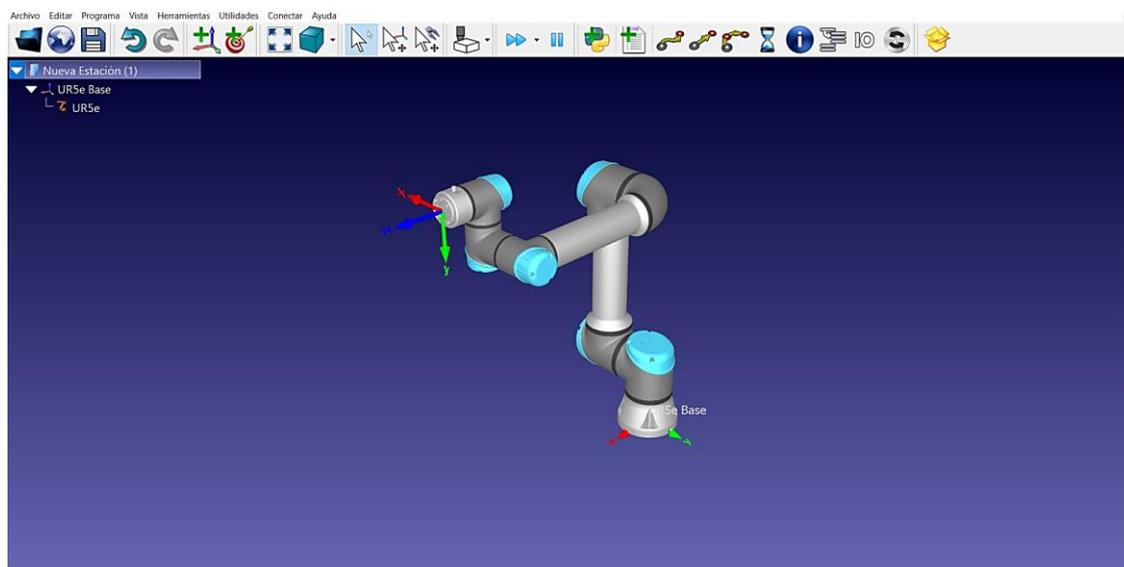
Todos estos parámetros de configuración son tenidos en cuenta a la hora de generar el script final en el lenguaje de programación propio del robot que se vaya a utilizar, y están definidos en el backend de la aplicación.

Para que el sistema sea capaz de generar tanto una simulación como el programa final del robot, es necesario valerse de RoboDK (<https://robodk.com/es/>), un software de simulación y programación de robots industriales desarrollado por RoboDK Inc.

El software dispone de una conexión con Python mediante una API externa. Para conectarse con esta API de manera sencilla, se utiliza la librería RoboDK de Python.

La librería RoboDK de Python está formada por un conjunto de herramientas y funciones diseñado para facilitar la interacción entre el software RoboDK y el lenguaje de programación Python. Esta librería actúa como un puente, permitiendo a los desarrolladores controlar y automatizar tareas en RoboDK utilizando scripts de Python. Entre sus principales funciones, se encuentra la capacidad de cargar modelos de robots y herramientas, definir y modificar trayectorias, simular movimientos del robot y generar programas para distintos controladores de robots. Asimismo, permite la integración con otros softwares y hardware, gracias a su capacidad para recibir y enviar información a través de la API.

Cuando se inicia el backend, Python inicia a su vez una instancia de RoboDK con la que comunicarse y cargará un robot en escena. En el caso del presente proyecto se han creado estaciones de trabajo con robots UR5 y UR16, ya que son los robots de AIDIMME planteados para ser utilizados durante el proyecto.



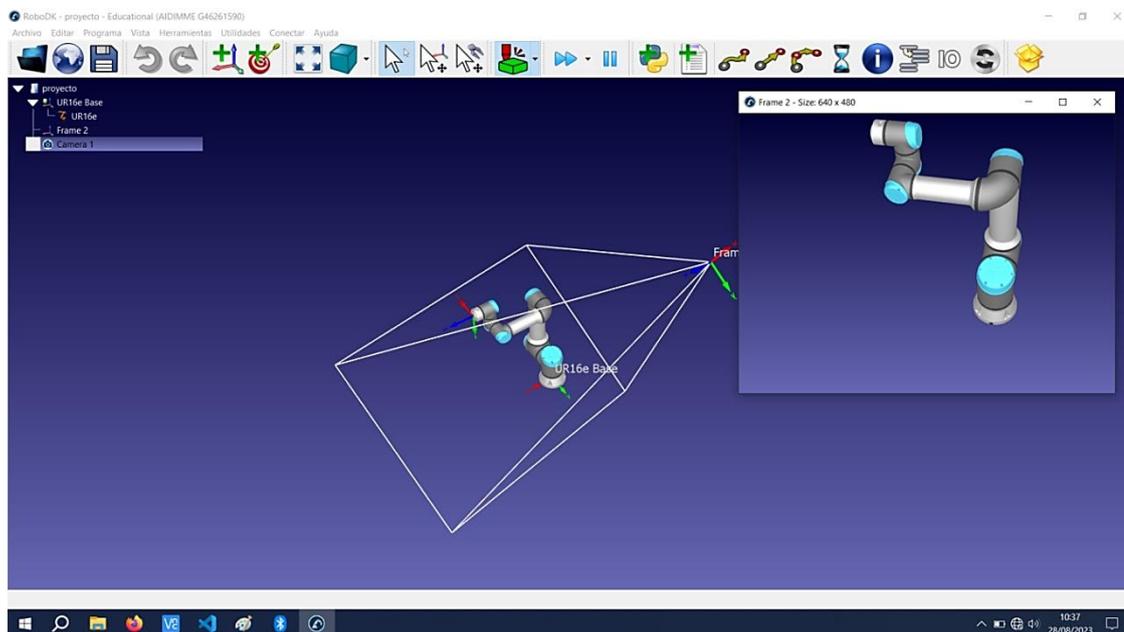
Ejemplo de estación de trabajo creada para un robot UR5  
Fuente: elaboración propia

El endpoint /generar\_programa recibe una petición POST en la que se incluyen en el cuerpo de la petición un archivo JSON indicando el token identificador de la grabación de datos y el tipo de movimiento que se desea utilizar

### 2.2.3- Integración de un simulador de trayectorias

Para realizar la simulación de la ejecución, se va a utilizar una vez más, el software RoboDK y la librería robotdk de Python. La API de RoboDK no incluye ninguna funcionalidad propia para generar vídeos de una simulación, es por esto por lo que ha sido necesario programar esta parte utilizando Python.

Para obtener el efecto deseado se ha incluido un nuevo eje de coordenadas en el proyecto, y en este se ha incluido una cámara. Este elemento de RoboDK permite obtener una imagen 2D del entorno 3D desde el punto de coordenadas en el que se ha fijado la cámara, tal como se muestra en la siguiente imagen.



Nuevo eje de coordenadas, cámara y visualización  
Fuente: elaboración propia

El elemento de cámara incluye la capacidad de generar una instantánea en un momento dado y guardarla como archivo png.

Cuando el Backend recibe una petición al endpoint /simular, este se asegurará de que exista la cámara en el proyecto, y de no ser así, incluirá una nueva. Tras esto, Python ordena a RoboDK que realice movimientos a todos los puntos indicados por el conjunto de datos. Después de cada movimiento indicará al elemento de cámara que tome una instantánea que guardará en una carpeta destinada a tal efecto. Las instantáneas se numerarán en orden ascendente de forma que se mantenga la secuencia de ejecución en el orden correcto.

Una vez terminado el proceso de simulación, Python accederá a la carpeta de instantáneas y obtendrá todas aquellas que tienen el token solicitado y las incluirá en un nuevo archivo respetando el orden de generación. Este archivo será de tipo GIF, es decir, animado. Cuando el archivo está conformado, el backend lo enviará al frontend que tendrá la responsabilidad de mostrarlo al usuario.

#### Comunicación y transferencia de programa

Para la transferencia del programa generado a través de los desarrollos realizados, no se ha identificado en la API del software RoboDK una función que permita realizar la transferencia y ejecución de este al robot. Si se puede realizar desde un programa nativo generado desde el entorno de programación de RoboDK, pero no a través de las funciones de la API.

Esta situación no supone ningún problema, ya que el script generado se puede guardar en un pendrive, conectarlo al robot y cargarlo de forma manual sin ningún tipo de problema.

### 2.2.4- Validación de desarrollos

A continuación, se muestra el resultado de diferentes pruebas de validación llevadas a cabo para verificar el funcionamiento de los desarrollos descritos en los apartados anteriores, así como los ligados a la captura de datos durante la fase de demostración humana.

Estas pruebas suponen una fase previa e inicial a las que se desarrollan en el paquete de trabajo PT6, materializadas en un demostrador basado en un proceso industrial analizado previamente.

#### Sistema de captura de datos

Se accede a la aplicación del proyecto, donde la pantalla principal permite seleccionar el sistema de captura de datos a utilizar.



Pantalla inicial de la aplicación Robotrack  
Fuente: elaboración propia

Se selecciona la cámara que captura la posición (coordenadas x, y) de las articulaciones del usuario que realiza la demostración.



Selección del sistema de captura de datos  
Fuente: elaboración propia

Se activa la aplicación desarrollada para la cámara Kinect. El usuario realiza un movimiento en diagonal con la mano derecha, quedando los puntos registrados tal y como se muestra en la siguiente figura.



Resultado de la grabación de los datos capturados por la cámara Kinect  
Fuente: elaboración propia

### Sistema de análisis de trayectorias y acciones

Al pulsar sobre el botón “Siguiente” se accede a la pantalla de suavizado.



Selección de una fuerza de suavizado “4”  
Fuente: elaboración propia

Se selecciona una fuerza de suavizado de “4”. Al pulsar sobre el botón “Suavizar” se aplica un modelo de medias móviles de cuatro periodos y muestra la nueva serie de datos en pantalla, tal y como se muestra en la siguiente figura.



Resultado de haber aplicado el proceso de suavizado  
Fuente: elaboración propia

Al pulsar sobre el botón “Siguiete” se accede a la pantalla de granulado.



Selección de una fuerza de granulado “5”  
Fuente: elaboración propia

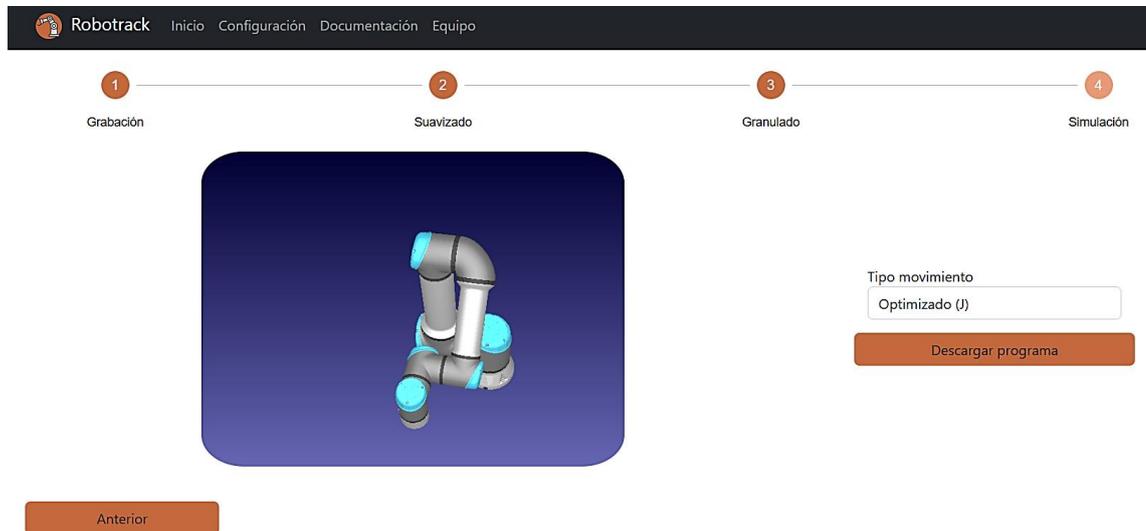
Se selecciona una fuerza de granulado de “5”. Al pulsar sobre el botón “Granular” se aplica un modelo donde los puntos de la serie se espacian y muestra la nueva serie de datos en pantalla, tal y como se muestra en la siguiente figura.



Resultado de haber aplicado el proceso de granulado  
Fuente: elaboración propia

### Simulador

Se pulsa en el botón “Siguiete” y se accede a la pantalla de simulación. En pantalla aparece un video donde se ve como el robot realiza una trayectoria pasando por los diferentes puntos de la trayectoria final generada.



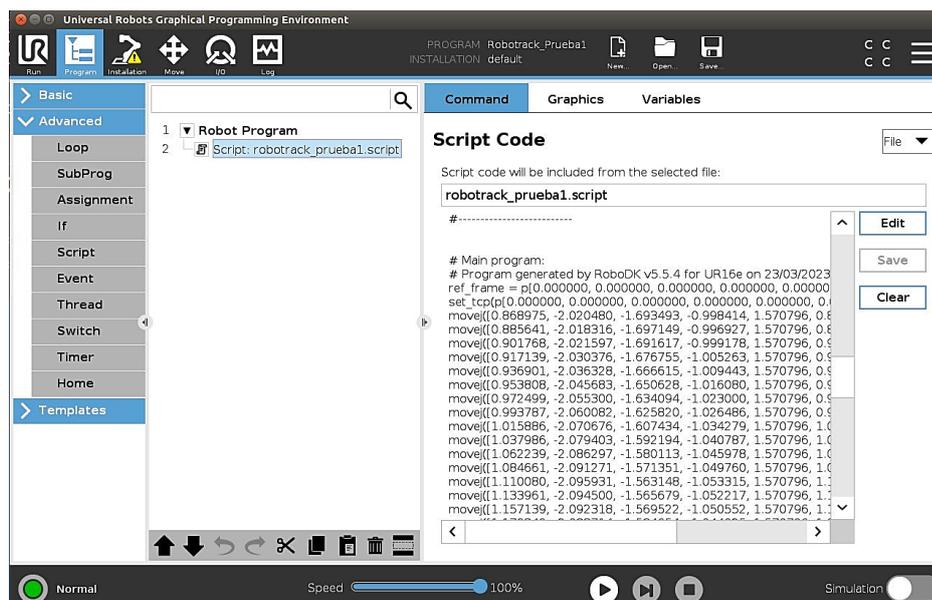
Resultado de haber aplicado el proceso de simulación  
Fuente: elaboración propia

### Sistema de generación de programa de robot

Se pulsa sobre el botón “Descargar programa” y se muestra el avance la descarga del script para el robot.

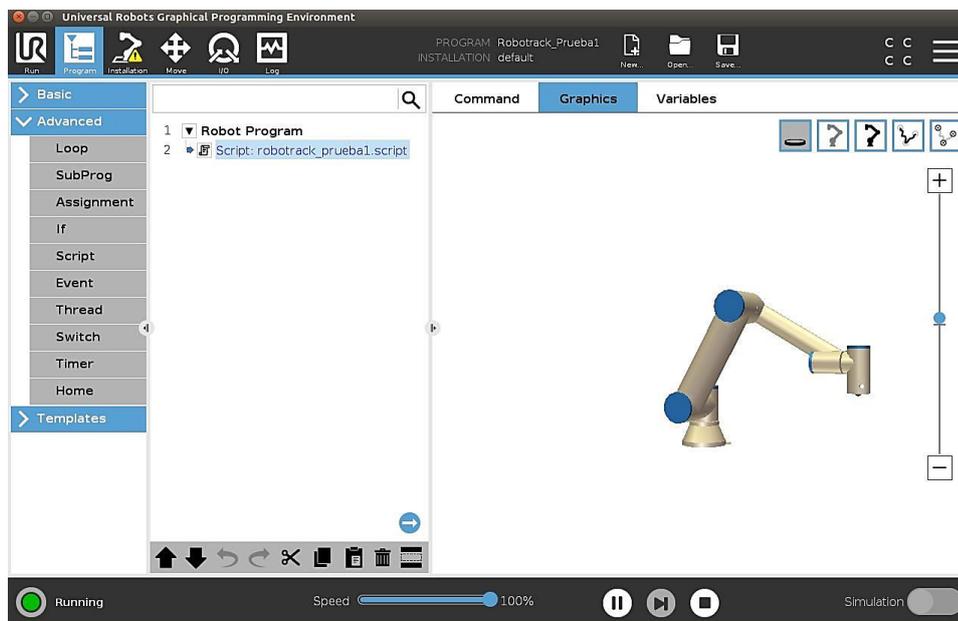
### Sistema de comunicación y transferencia de programa

En la siguiente imagen se puede ver como se ha cargado un ejemplo de programa en URscript (“robotrack\_prueba1.script”) generado por medio de la interfaz HMI desarrollada en Robotrack. La prueba se ha realizado en el programa Polyscope, entorno de programación de los robots Universal Robots.



Script generado por la HMI de Robotrack y cargado en Polyscope  
Fuente: elaboración propia

En la siguiente imagen se puede ver como se simula desde el propio entorno Polyscope, la ejecución del programa generado desde la HMI de Robotrack.



Ejecución en Polyscope del script generado en la HMI de Robotrack  
Fuente: elaboración propia

## 2.3- DESARROLLO DE UN DEMOSTRADOR

El objetivo de este paquete de trabajo era desarrollar el sistema completo que permita, a partir de los datos capturados, analizar los mismos, convertirlos en un programa de robot que imite las acciones capturadas, probar el programa en un simulador virtual y finalmente transferirlo al controlador del robot.

Todos estos sistemas se debían integrar en el interfaz HMI, de forma que se garantizase su usabilidad por parte de usuarios industriales.

Se desarrollaron las siguientes actividades

### 2.3.1- Diseño del demostrador

#### 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 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

#### 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 pizzas 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)

El proceso de llenado y precintado de los cubos de 15 kilos puede ser automatizado mediante la integración de un robot colaborativo, pero las características de dicho proceso no hacen posible que esta automatización pueda realizarse mediante el aprendizaje por demostración.

#### Caso envasado alimentación (propuesto por empresa colaboradora Alnut)

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

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.

#### Caso proceso de lijado de piezas

Se trata de un proceso donde en función del tipo de pieza a procesar (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.

#### 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.

#### KPIs y método de la validación

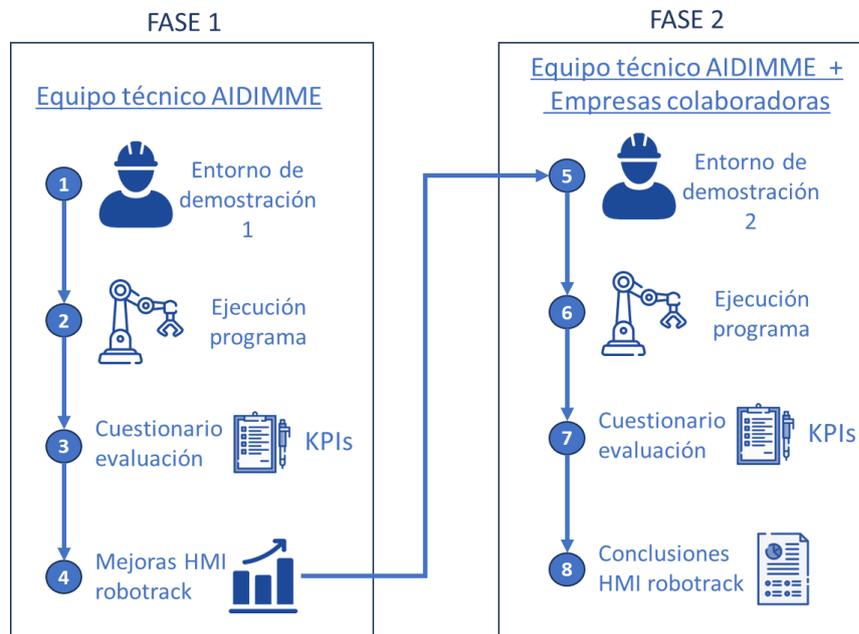
En la fase de demostración se ha buscado la participación tanto de personal técnico de AIDIMME como de las empresas colaboradoras en el proyecto. Para ello, se han planificado dos fases de demostración.

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.



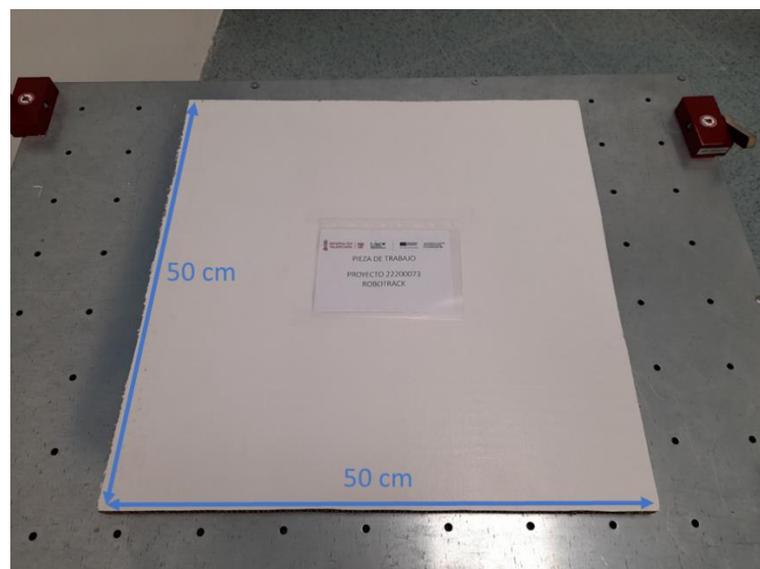
Esquema del proceso de validación  
Fuente: elaboración propia

### 2.3.2- Montaje del demostrador y pruebas

#### 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.

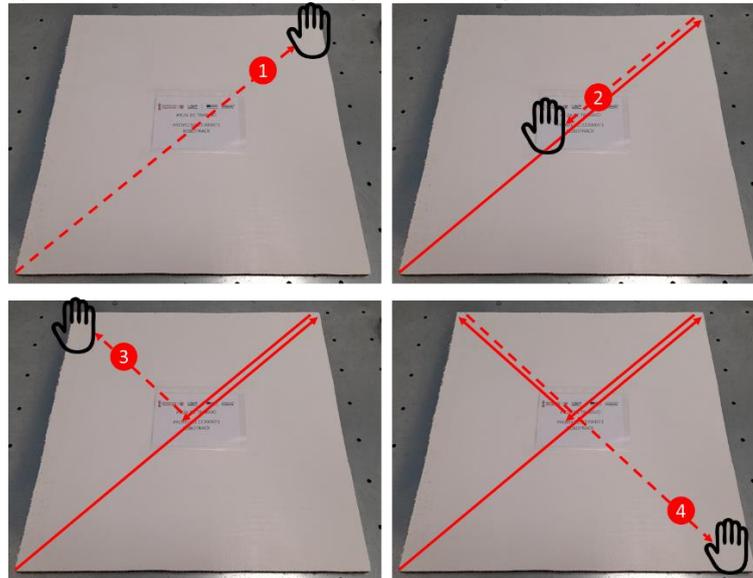


Dimensiones pieza empleada en las pruebas de demostración

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.



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 (entorno Windows) donde está instalada la aplicación Robotrack.

En otra zona, se dispone de un robot UR5e. Para evitar riesgos 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.



Robot UR5 empleado en las pruebas  
Fuente: elaboración propia

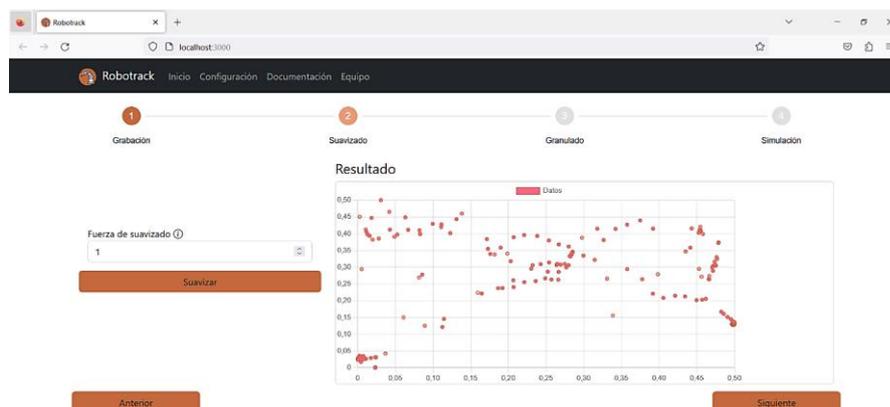
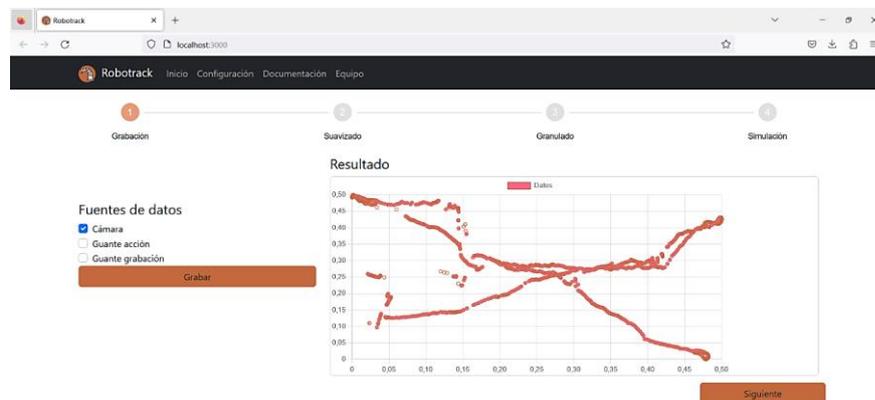
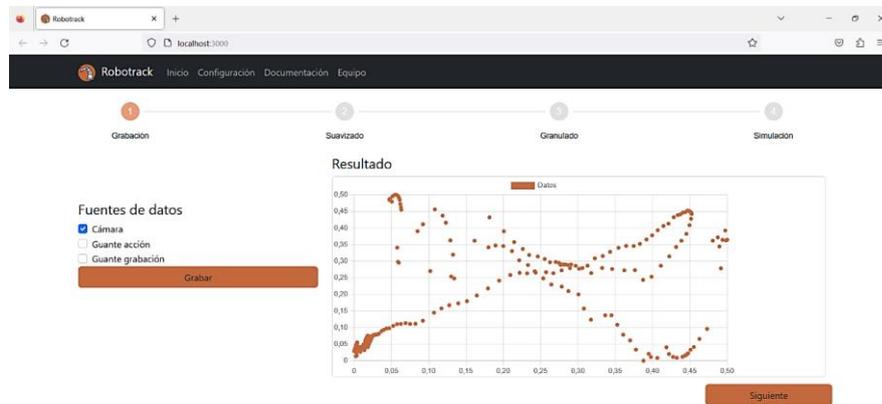
### Ejecución de las pruebas

#### *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 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.



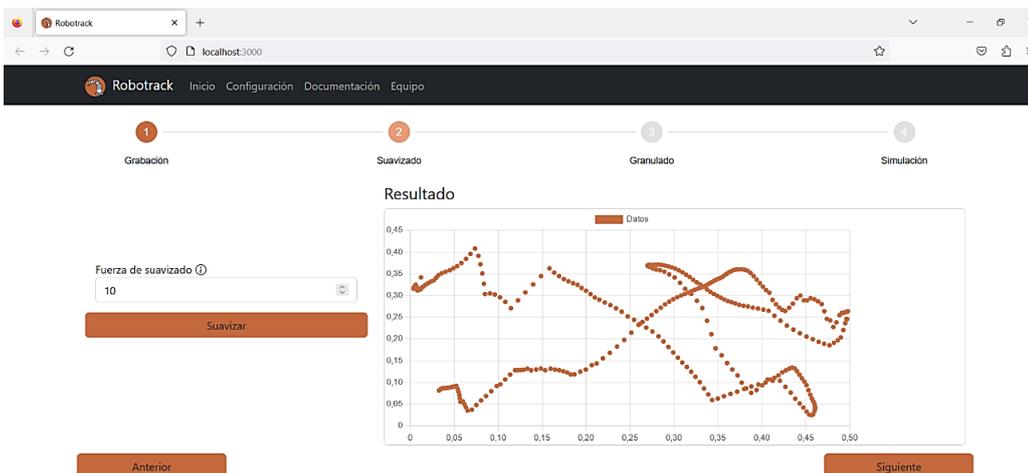
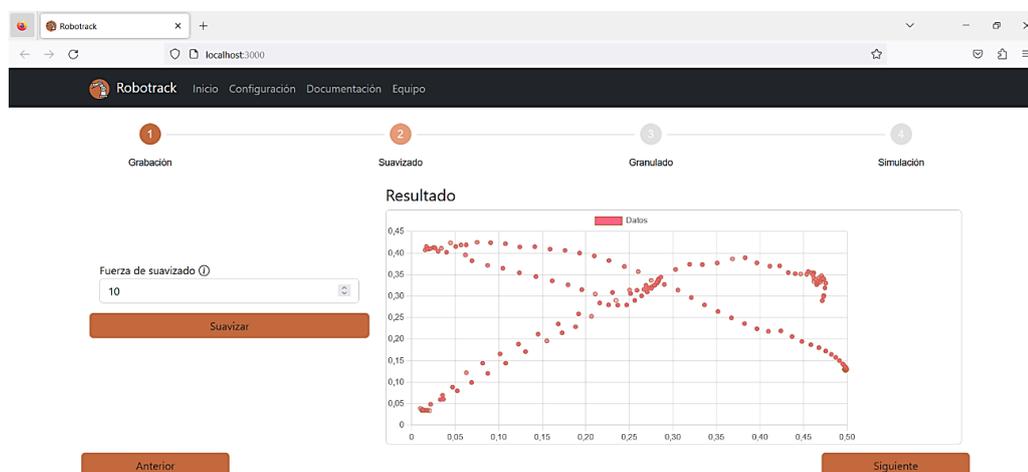
Datos en bruto capturados durante la demostración  
Fuente: elaboración propia

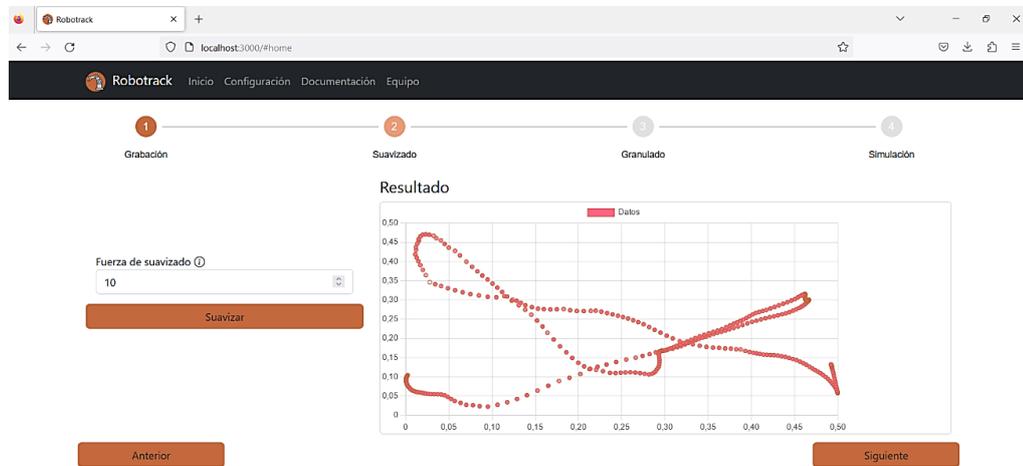
Cuando el usuario ha realizado la trayectoria de forma muy lenta se han acumulado una gran cantidad de puntos. En el caso de los usuarios que han ejecutado la trayectoria de forma muy rápida, la cantidad de puntos recogido es menor.

### 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.



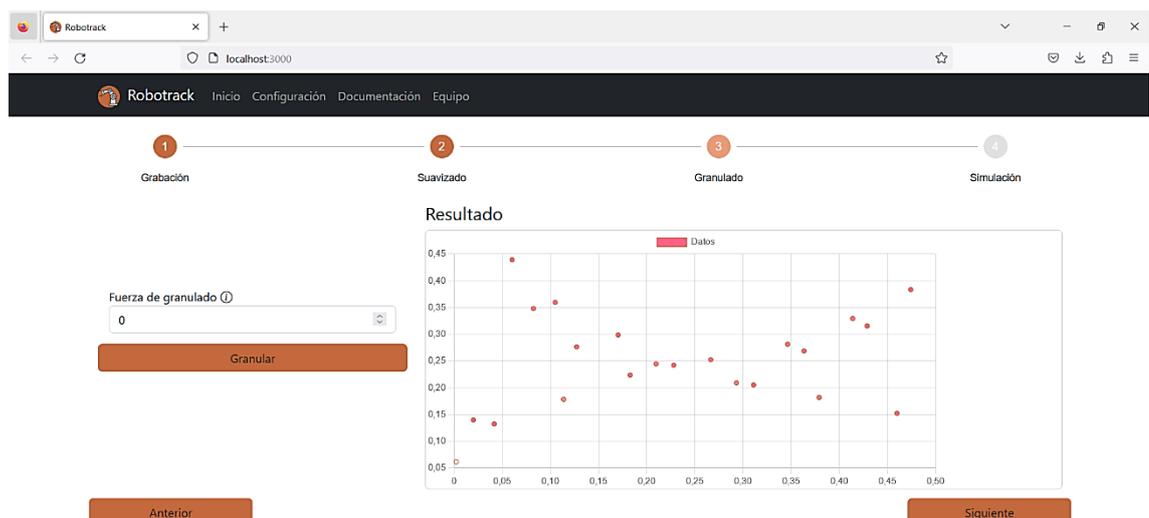


Ejemplos de trayectoria tras el proceso de suavizado  
Fuente: elaboración propia

### 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 cuántos centímetros deja un punto de la trayectoria, pudiendo elegir entre 1 y 10 cm.

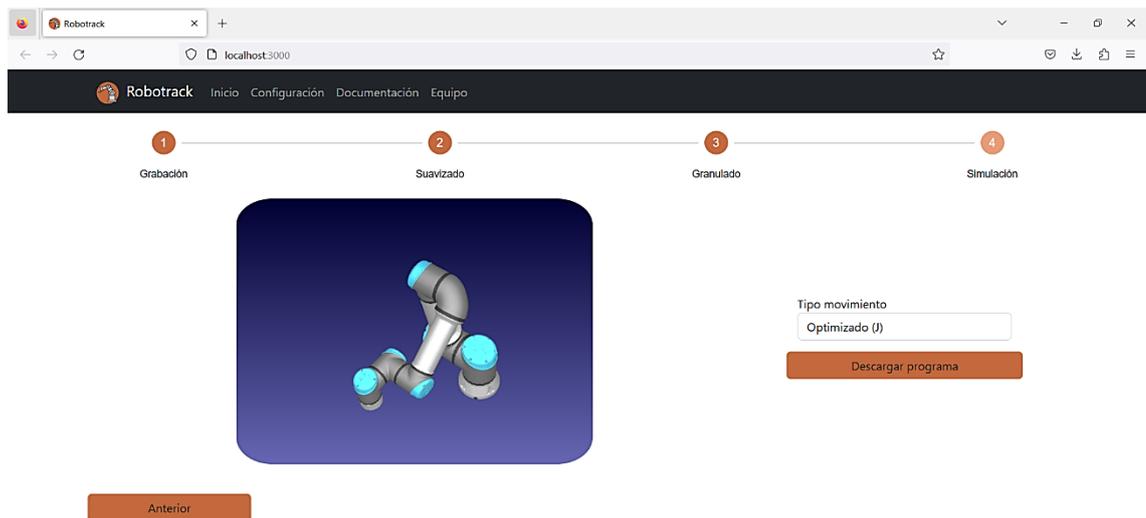
A continuación, en las siguientes imágenes se muestra el resultado de un ejemplo tras llevar a cabo el proceso de granulado.



Ejemplo de trayectoria tras el proceso de granulado  
Fuente: elaboración propia

### Etapa de simulación

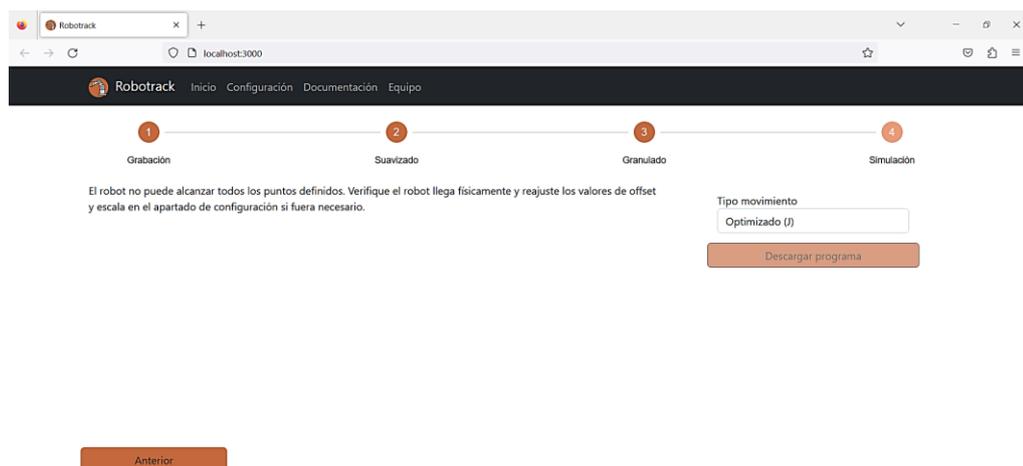
A continuación, se genera video de simulación de la trayectoria del robot siguiendo la trayectoria generada tras el suavizado y granulado de los datos capturados inicialmente.



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.



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.

```

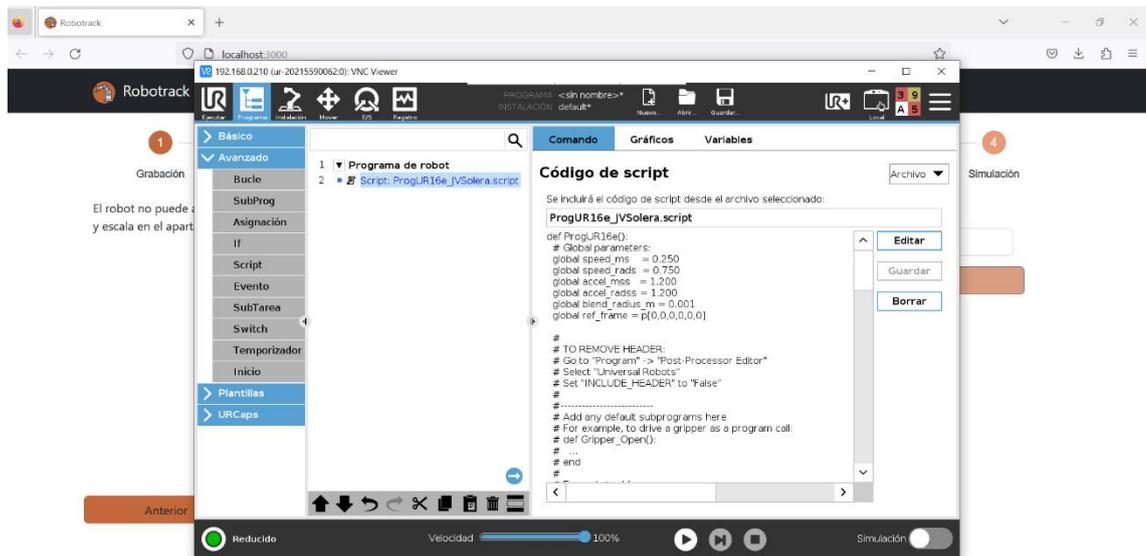
1 def ProgR16e():
2   # Global parameters:
3   global speed_m = 0.250
4   global speed_rads = 0.750
5   global accel_mss = 1.200
6   global accel_rados = 1.000
7   global blend_radius_m = 0.001
8   global ref_frame = p[0,0,0,0,0]
9
10
11 # TO REMOVE HEADER:
12 # Go to "Program" -> "Post-Processor Editor"
13 # Select "Universal Robots"
14 # Set "INCLUDE_HEADER" to "false"
15 #
16 # .....
17 # Add any default subprograms here
18 # For example, to drive a gripper as a program call:
19 # def Gripper_Open():
20 #   ...
21 # end
22 #
23 # Example to drive a spray gun:
24 def SprayOn(value):
25   # use the value as an output:
26   DO_SPRAY = 5
27   if value == 0:
28     set_standard_digital_out(DO_SPRAY, False)
29   else:
30     set_standard_digital_out(DO_SPRAY, True)
31   end
32 end
33
34 # Example to drive an extruder:
35 def Extruder(value):
36   # use the value as an output:
37   if value < 0:
38     # stop extruder
39   else:
40     # start extruder
41   end
42 end
43
44 # Example to move an external axis:
45 def MoveAxis(value):
46   # use the value as an output:

```

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.



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.



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 de 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 de 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.

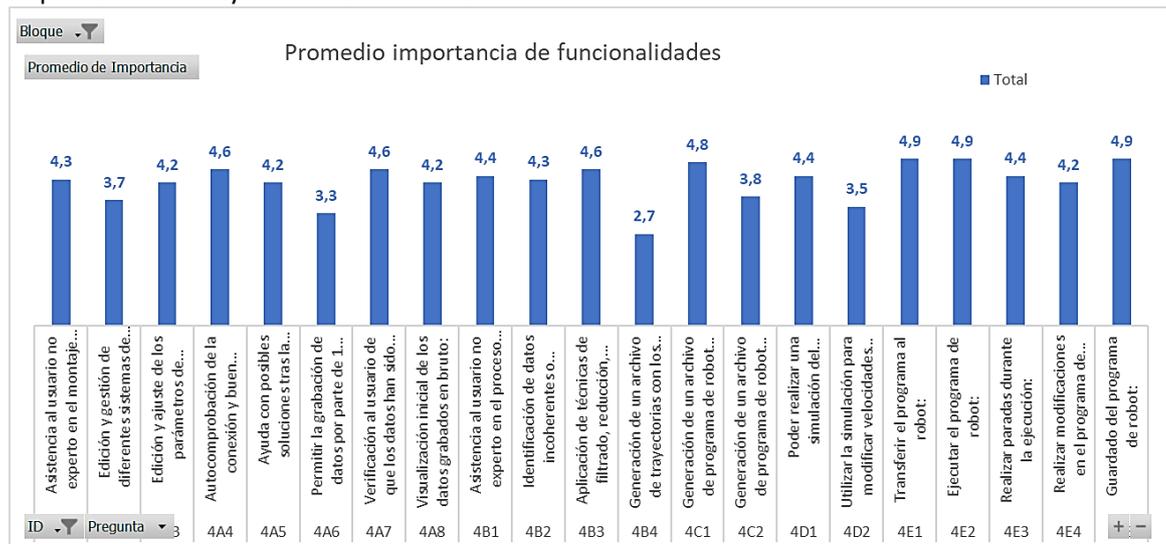


Figura 1 – Resultado promedio de la importancia de las funcionalidades

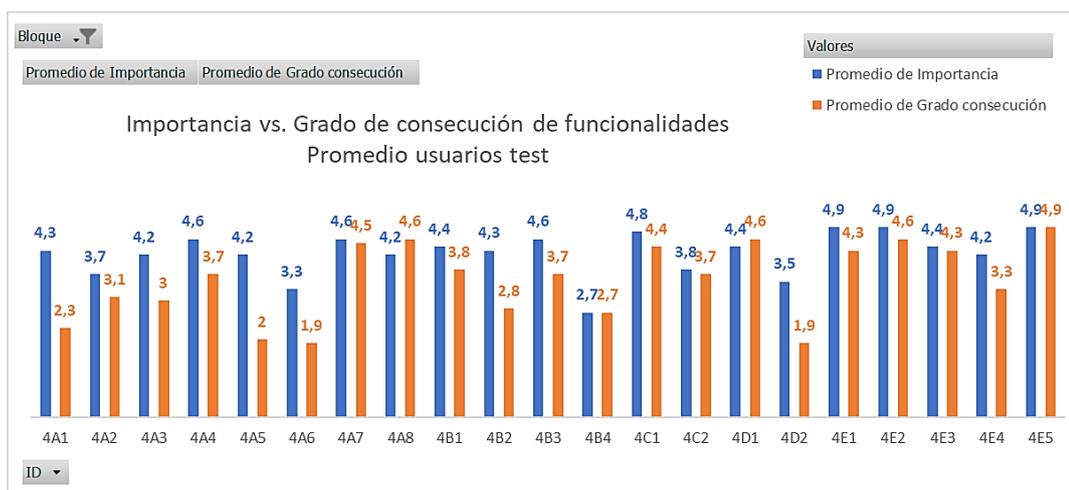
Fuente: elaboración propia

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 de 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:

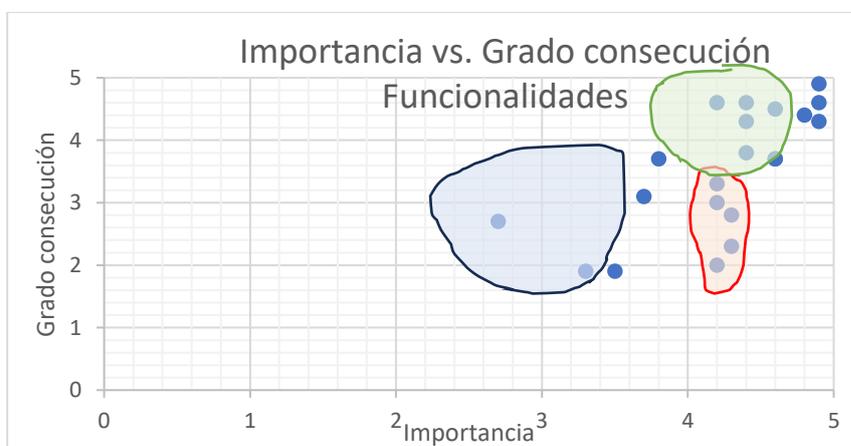
En la siguiente gráfica se muestra una comparativa entre la importancia otorgada a cada funcionalidad y el grado de alcance percibido por los usuarios tras la etapa de pruebas.



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.



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 de 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:

## 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 trabajará 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.

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.

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.

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 más suave. Se ha definido el parámetro para que sea el usuario el que lo defina según sus necesidades.

Modificaciones en la configuración del demostrador

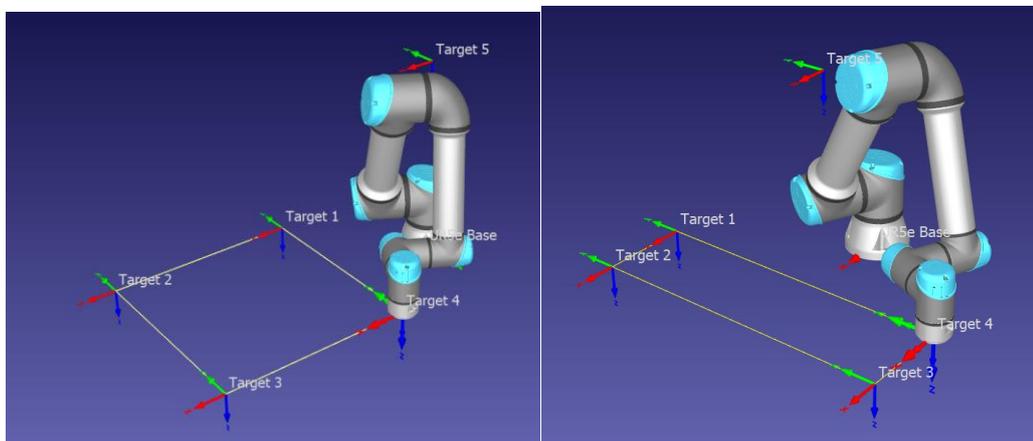
Para la segunda fase de las demostraciones se utiliza la misma mesa y pieza de trabajo que en la fase 1. Para la ubicación de la cámara de visión se utiliza un trípode ajustable en altura para verificar la incidencia de la altura de la cámara sobre la calidad de los datos de trayectoria capturados.



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.

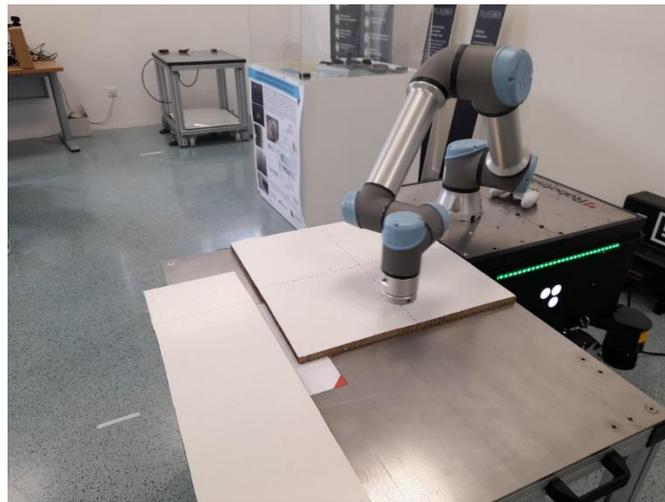
Previamente se ha comprobado mediante el software RoboDK, que el robot UR podía ejecutar trayectorias sobre ambas superficies, ya que su alcance es reducido.



Verificación de alcances del UR5e sobre ambas piezas

Fuente: elaboración propia

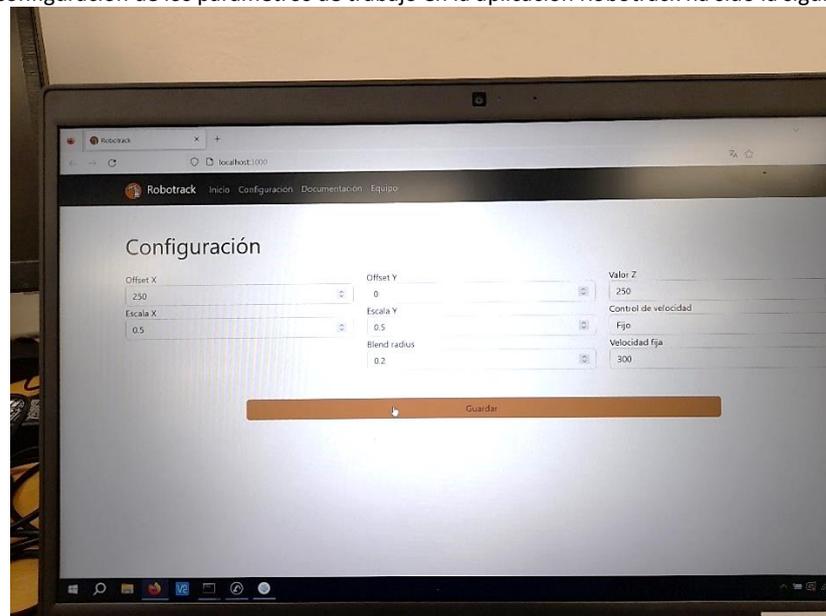
Se ha establecido un offset tanto en el eje x como en el eje y, para definir el punto de origen sobre el que el robot toma su origen de zona de trabajo. A partir de este origen (situado a 250 mm en el eje x, 0 mm en el eje y) se monta la mesa de trabajo donde ubicar las piezas sobre las que tenía que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



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.



Configuración de parámetros iniciales en Robotrack

Fuente: elaboración propia

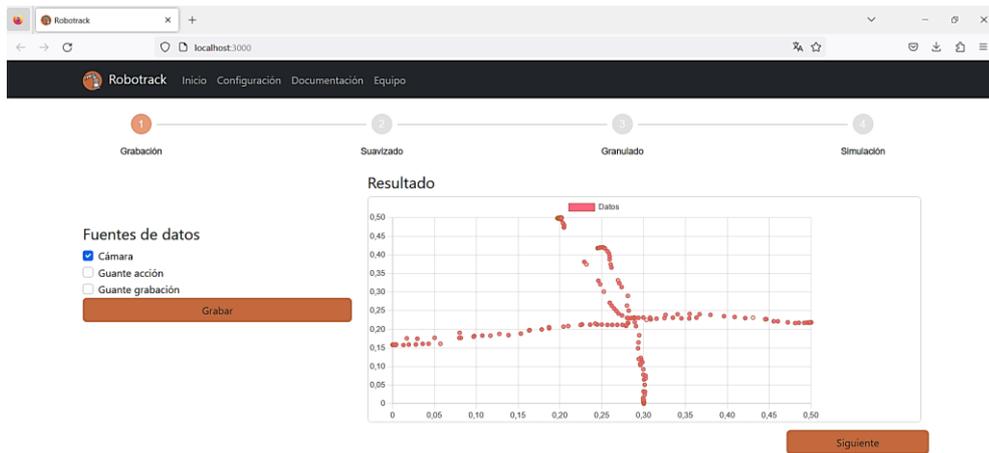
## Ejecución de pruebas en la Fase 2

### Etapas 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).

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).

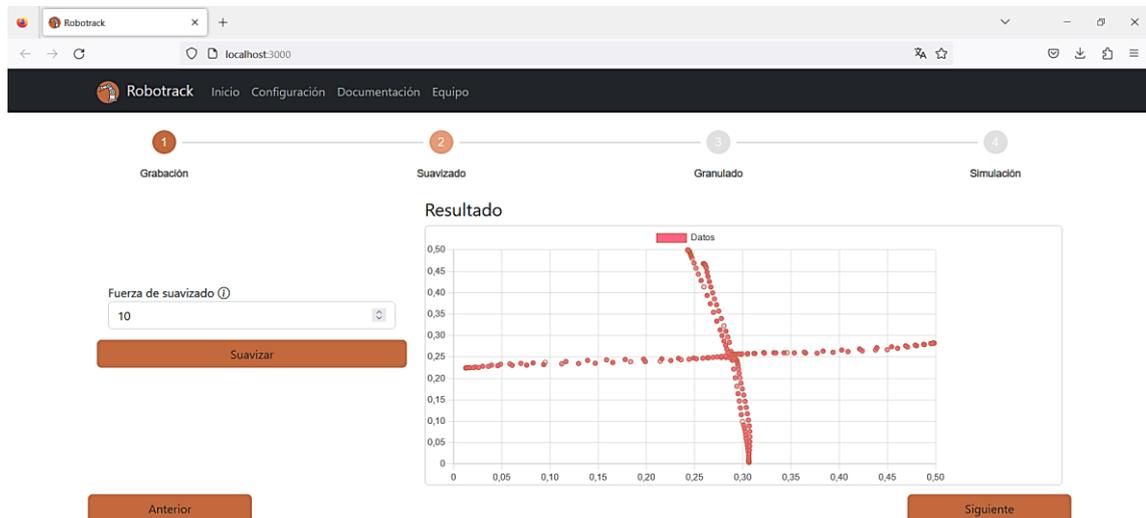
En una primera instancia lo que se pidió es que los participantes ejecutaran una trayectoria simple sobre la pieza, para comprobar visualmente la precisión de la ejecución del programa de robot generado. A continuación, se muestran datos en bruto registrados para diferentes usuarios.



Datos en bruto capturados  
Fuente: elaboración propia

### Etapa de suavizado y suavizado de datos en bruto

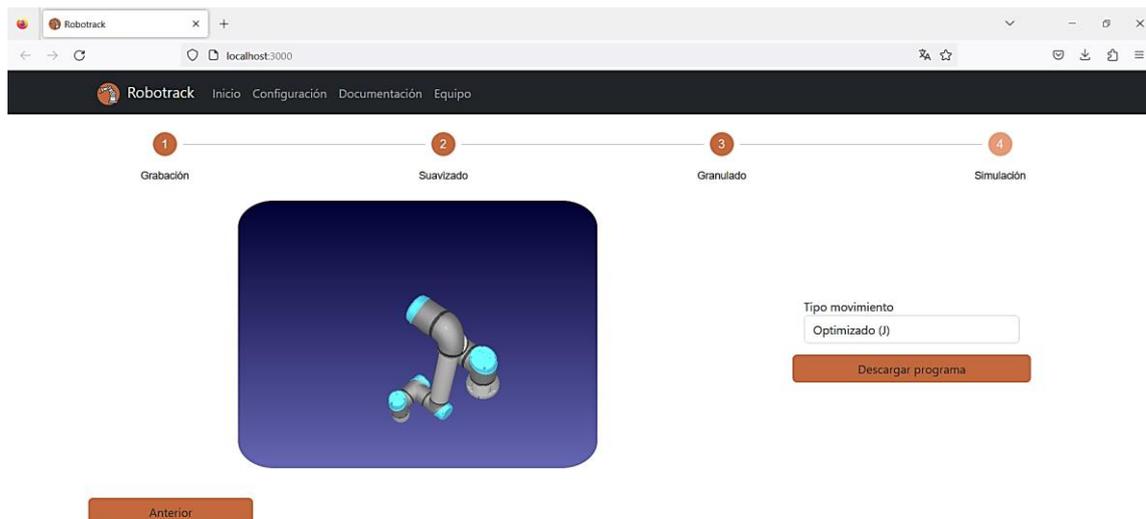
Al igual que en Fase 1 de la validación, se lleva a cabo el proceso de suavizado y granulado de los datos capturados en bruto. A continuación, se muestra un ejemplo de captura de pantalla tras haber llevado a cabo un proceso de suavizado.



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 video de simulación del robot ejecutando la trayectoria finalmente definida. Se muestra en la siguiente imagen un ejemplo de simulación generada.



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_202309181525114185():
# 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]

#
# 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
end

```

Ejemplo video de script para UR5e

Fuente: elaboración propia

### *Etapas de ejecución del programa de robot*

Los programas generados se guardan en un pendrive, y se cargan en el robot UR5e. 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.



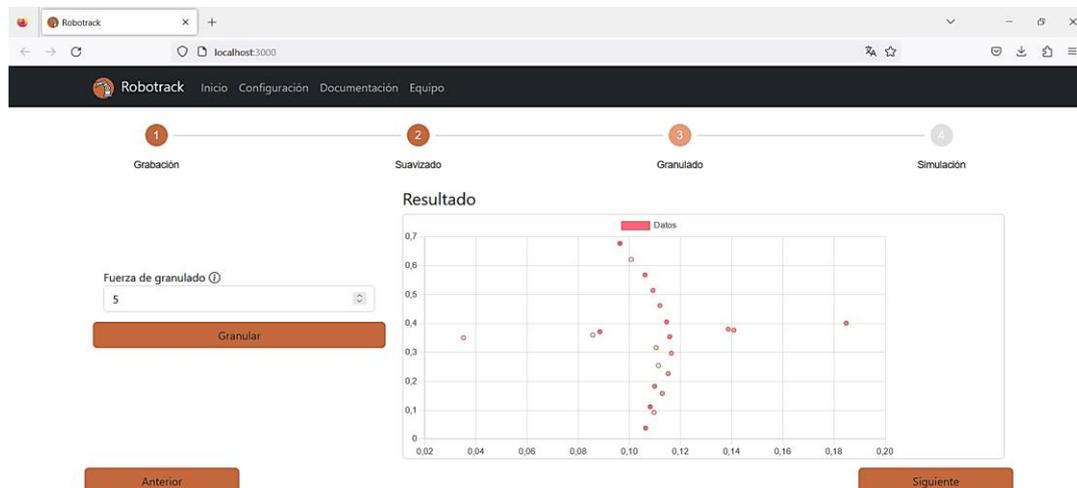
Ejemplo de robot ejecutando la trayectoria

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.

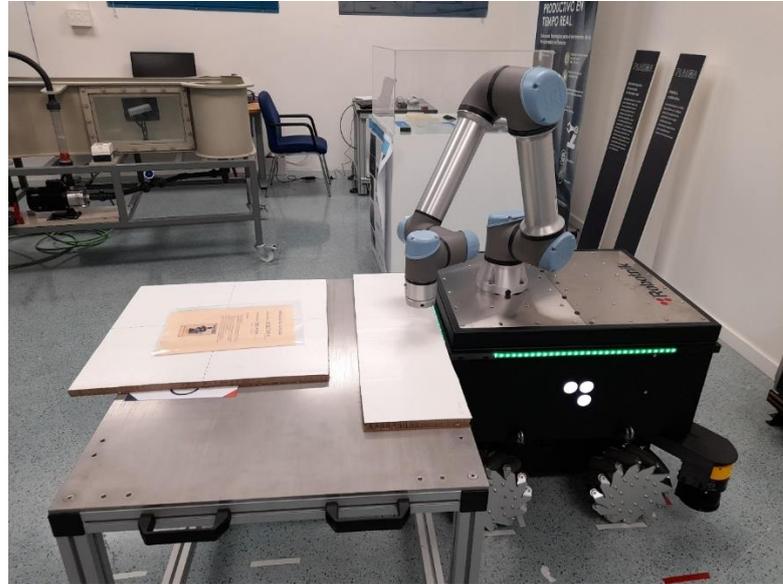
Mediante la pestaña de configuración se modificaron las dimensiones de la pieza objetivo a las de una pieza de 200 x 700 mm. Con los mismos datos en bruto grabados para la pieza de 500 x 500 mm, se realizó el suavizado y granulado de los datos, tal y como se muestra en la siguiente imagen.



Trayectoria suavizada y granulada para la nueva pieza

Fuente: elaboración propia

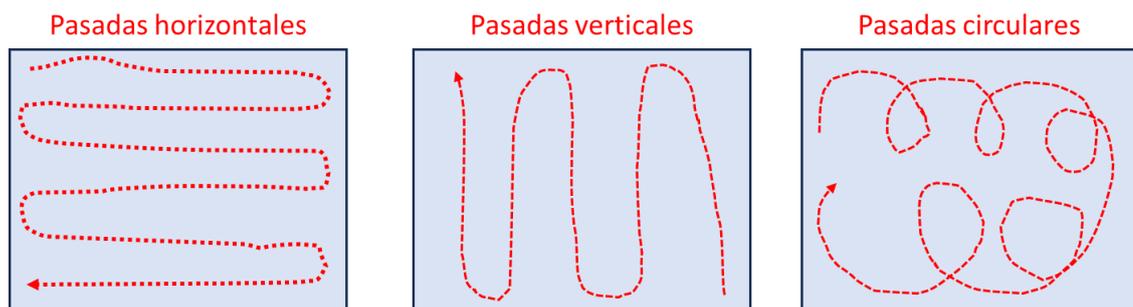
A continuación, se generaron las simulaciones y los programas asociados. Estos nuevos scripts se cargaron en el robot y se verificó que se ejecutaba correctamente la trayectoria original escalada sobre esta nueva pieza.



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.



Tipos de trayectorias complejas  
Fuente: elaboración propia

### Trayectorias mediante pasadas horizontales

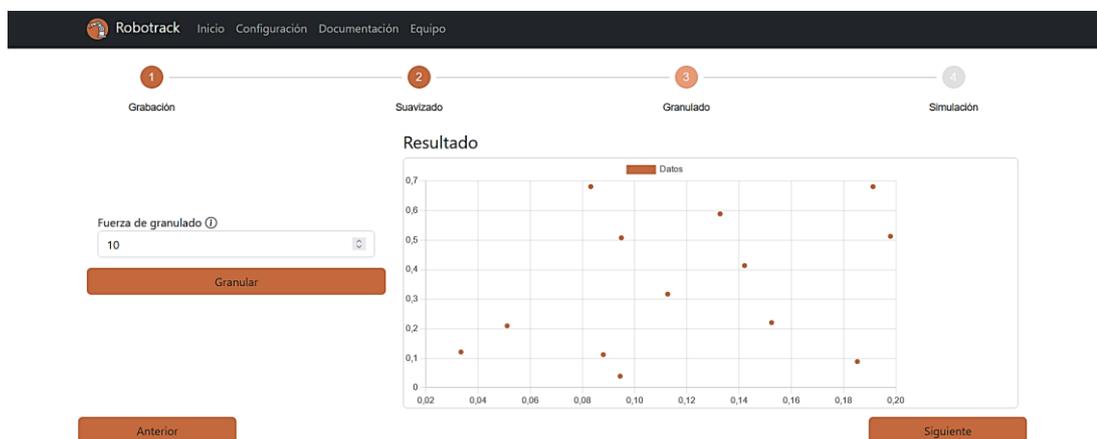
A continuación, se muestran un ejemplo de trayectoria generada imitando un proceso de lijado mediante pasadas horizontales.



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.

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.

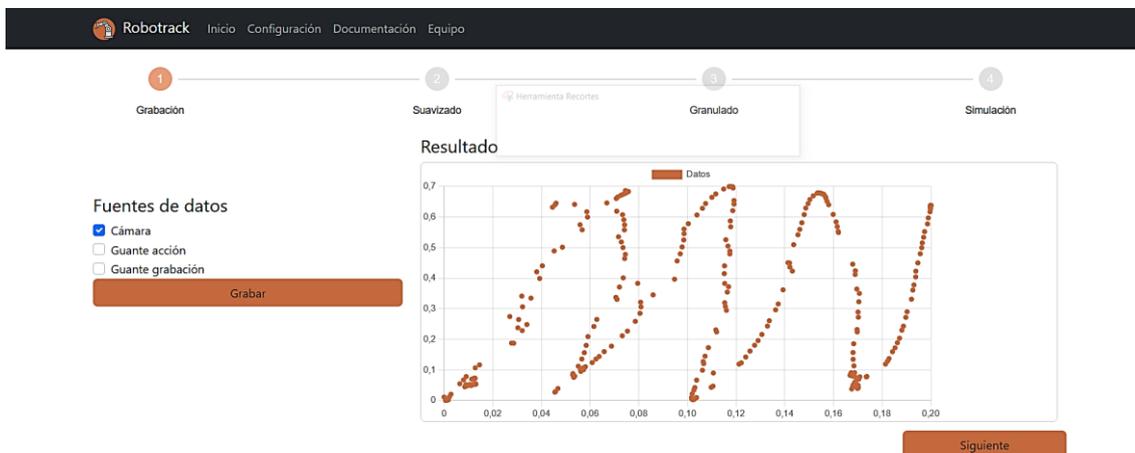


Ejemplo de granulado de datos excesivo  
Fuente: elaboración propia

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.

### 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.



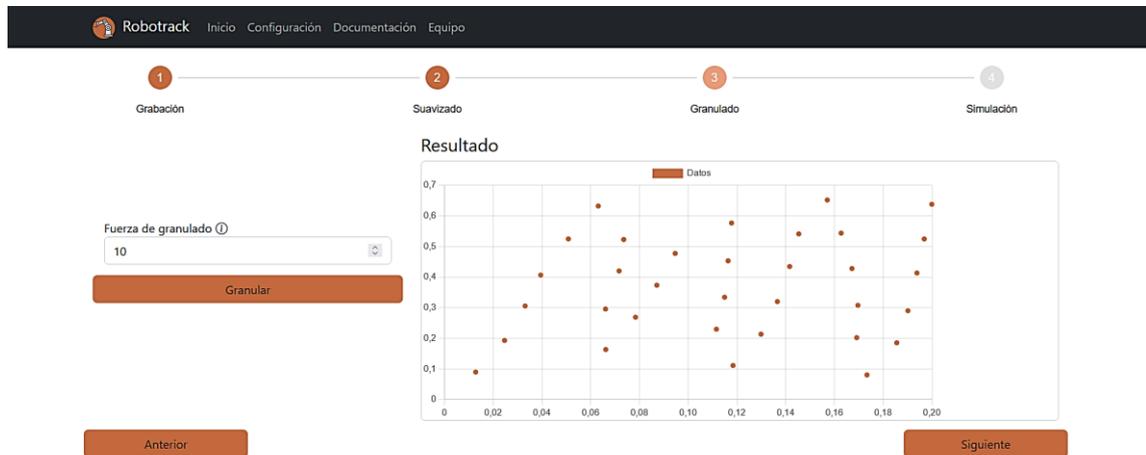
Ejemplo de datos en bruto para trayectorias con líneas verticales  
Fuente: elaboración propia

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.



Ejemplo de suavizado con fuerza 10  
Fuente: elaboración propia



Ejemplo de granulado con fuerza 10  
Fuente: elaboración propia

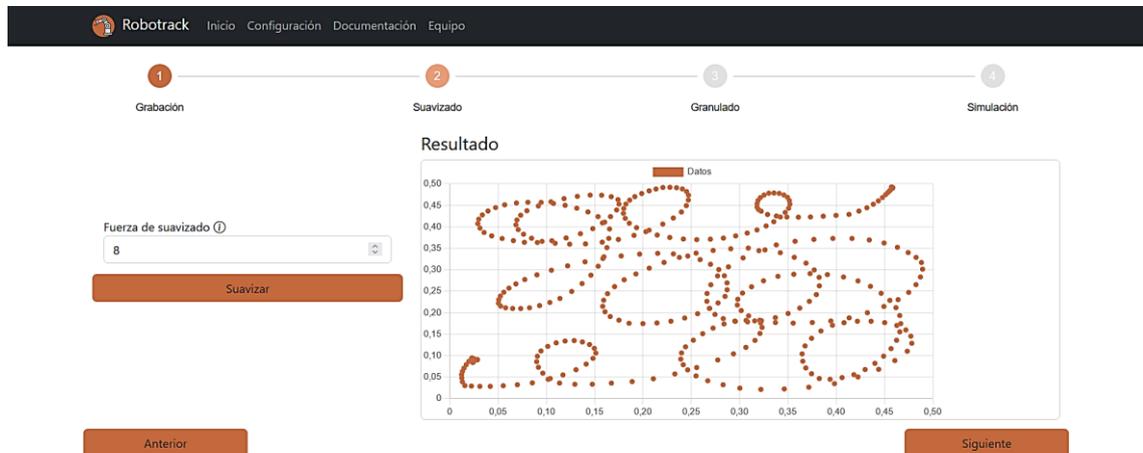
Se genera la simulación y la descarga del programa de robot, para ser testado visualmente sobre la pieza.

### 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.



Ejemplo de datos en bruto de trayectorias circulares  
Fuente: elaboración propia



Ejemplo de suavizado de datos con fuerza 10  
Fuente: elaboración propia



Figura 2 – 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.

### 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.

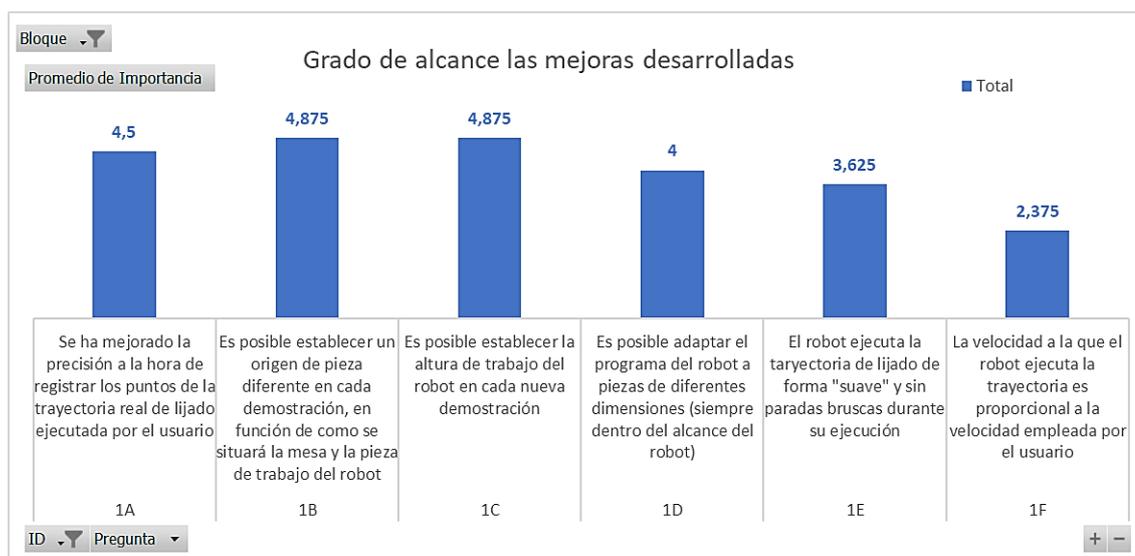
ID	Pregunta
----	----------

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

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.



Grado de alcance de mejoras desarrolladas

Fuente: elaboración propia

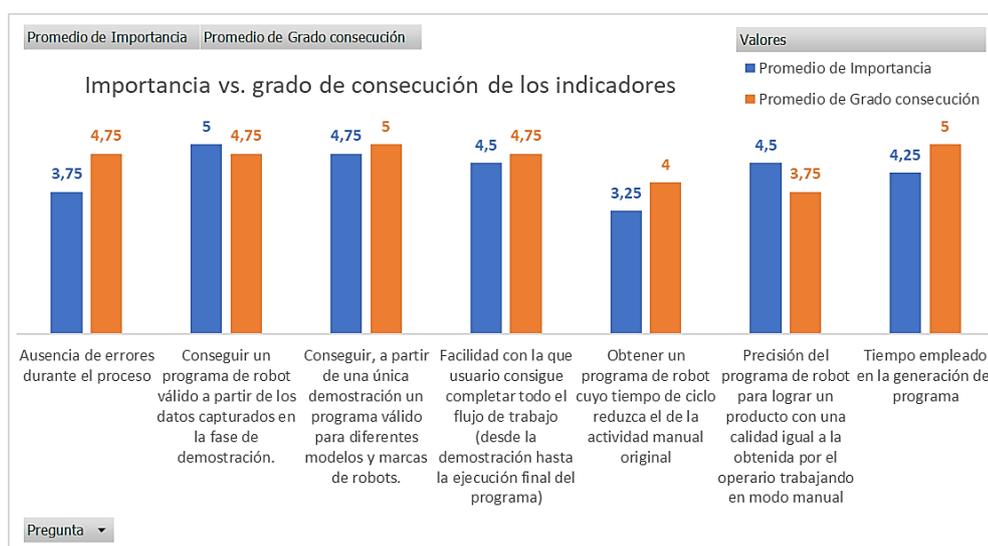
Las mejoras relacionadas con la "suavidad" de la trayectoria de lijado y la proporcionalidad de la velocidad proporcional a la ejecutada por el usuario presentan una puntuación más baja. Esta puntuación más baja se debe a los siguientes motivos:

- 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 más lenta que el usuario.

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

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.



Grado de alcance de KPIs – empresas colaboradoras

Fuente: elaboración propia

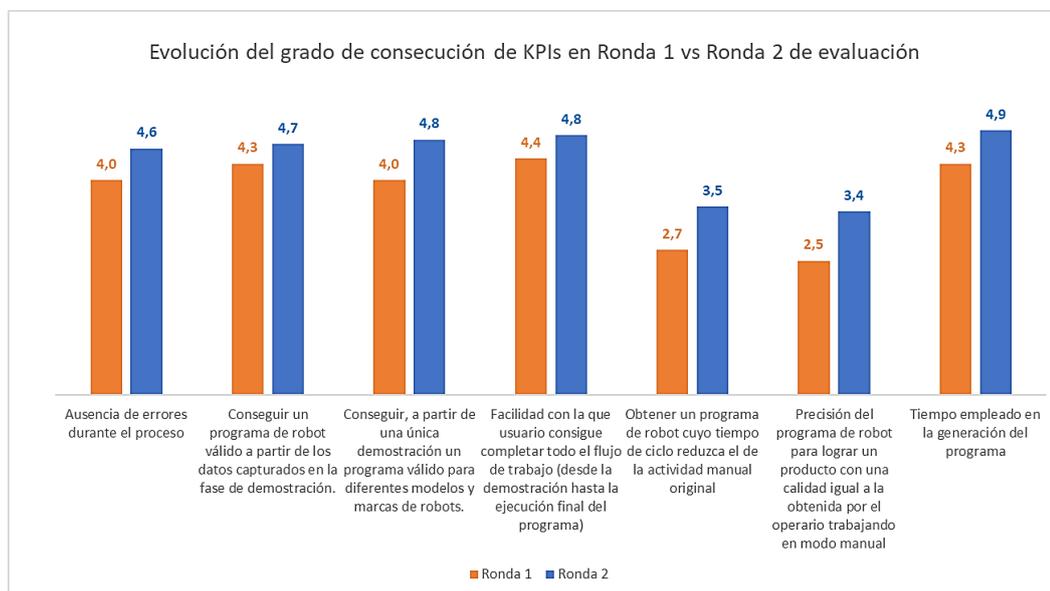
De forma general, dan por cumplidos todos los objetivos planteados. El KPI de menor promedio es la precisión del programa, que viene ligada por la precisión de la cámara de visión.

### 2.3.3- Validación del sistema

#### 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.

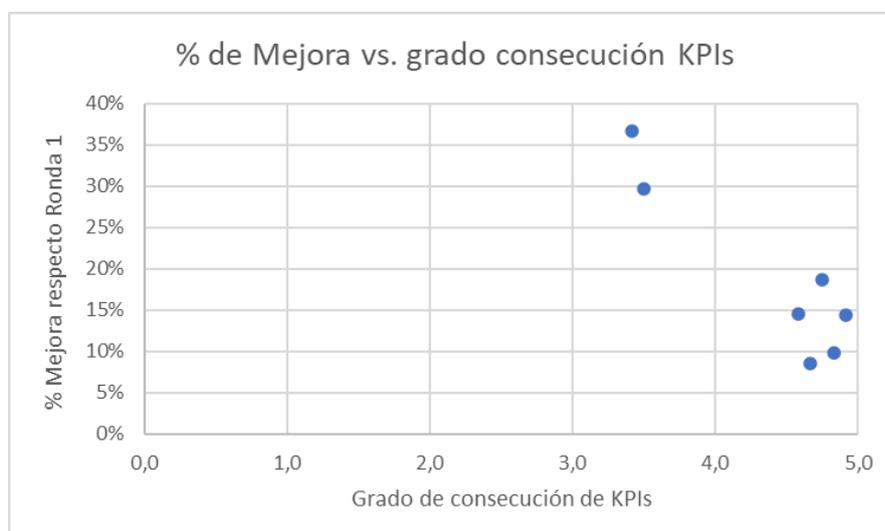
Tras las modificaciones realizadas, todos los indicadores han mejorado en la segunda ronda, respecto de la puntuación promedio obtenida en la primera ronda.



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.



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.

#### 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.



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 PT4, 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.

### 3. Resultados y conclusiones

**Como conclusión**, 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.

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 participación de: ALNUT, PINTURAS BLATEM, IT8 SOFTWARE, HURTADO RIVAS y CFZ COBOTS.

# **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](mailto:aidimme@aidimme.es)

[www.aidimme.es](http://www.aidimme.es)