ENTREGABLE
PROYECTOS—
2023

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

Entregable: E4.1- Desarrollo de la infraestructura del interfaz HMI

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

AIDIMME
INSTITUTO TECNOLÓGICO

ÍNDICE

A. Objetivo del entregable

Este entregable recoge las actividades realizadas durante la ejecución del **PT4 - Desarrollo de interfaz HMI para captura de secuencias de trabajo y transferencia de información al control del robot**

El objetivo del paquete de trabajo era diseñar y desarrollar un interfaz HMI que permita la integración de todos los sistemas implicados en el proceso de aprendizaje de robots mediante demostración humana, desde la captura de información relevante en las secuencias de trabajo manuales, hasta la transferencia del programa desarrollado para que el robot imite las secuencias capturadas. Algunos de los componentes del sistema se desarrollarán en el PT5, aunque esta circunstancia se considera en el diseño de la interfaz.

El objetivo concreto de este entregable es presentar los análisis, diseños y desarrollos llevados a cabo para la implementación final de la parte de la interfaz HMI encargada de gobernar los sistemas de captura de datos de la demostración humana.

B. Actividades realizadas

1. Necesidades de captura de información

1.1. Procesos por automatizar mediante aprendizaje por demostración

Se ha realizado 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 ha recogido la perspectiva de las empresas colaboradoras del proyecto (Alimentación y Nutrición Familiar S.L., Pinturas Blatem S.L., CFZ cobots S.L., Hurtado Rivas S.L. e IT8 Software Engineering S.L.) 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:

Figura 1 – 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.

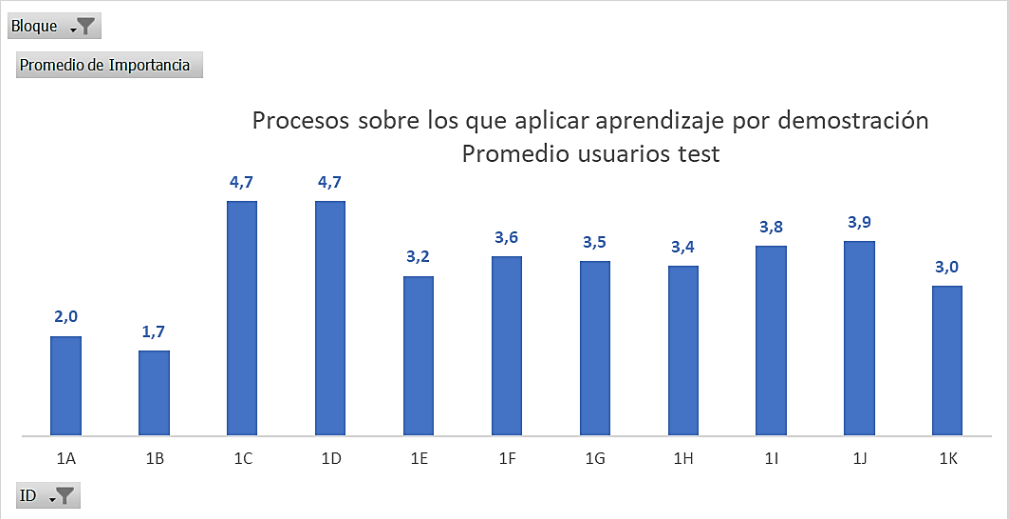

Figura 2 – Resultado promedio de procesos sobre los que aplicar aprendizaje por demostración
Fuente: elaboración propia

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. SOLUCIONES PARA LA CAPTURA DE INFORMACIÓN

2.1. 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 han identificado 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.):

Figura 3 – 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.

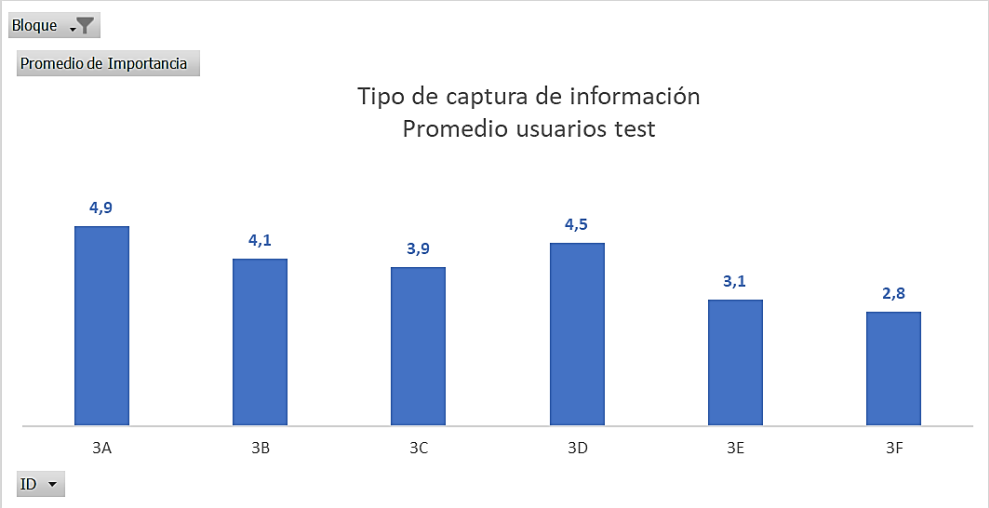

Figura 4 – Resultado promedio de parámetros que se considera importante capturar
Fuente: elaboración propia

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.

2.2. Identificación de sistemas de captura de datos

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.

2.2.1. 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.

Figura 5 – Cámara Kinect de Xbox (i)
Fuente: https://es.wikipedia.org/wiki/Kinect

El sistema Kinect presenta las siguientes características técnicas, que son de interés para las necesidades del proyecto:

- Seguimiento de Movimiento en 3D: Kinect utiliza una combinación de cámaras RGB y sensores de profundidad para capturar el movimiento del cuerpo en tres dimensiones. Esto permite a los jugadores controlar juegos y aplicaciones con gestos y movimientos naturales.

- Reconocimiento de Gestos: Kinect puede reconocer y seguir una variedad de gestos y movimientos, como agacharse, saltar, mover los brazos y mucho más. Esto permite a los jugadores interactuar con los juegos y la interfaz de la consola sin necesidad de un controlador físico.

- Reconocimiento de Voz: Kinect tiene la capacidad de reconocer comandos de voz, lo que permite a los usuarios controlar la consola y las aplicaciones utilizando instrucciones habladas. Puedes encender y apagar la consola, controlar la reproducción de medios y realizar otras funciones usando solo tu voz.

- Cancelación de Ruido: Para mejorar la precisión del reconocimiento de voz, Kinect incorpora tecnología de cancelación de ruido, lo que ayuda a filtrar el ruido de fondo y permitía una interacción más efectiva.

- Cámara RGB Integrada: Además de la cámara de profundidad, Kinect también tiene una cámara RGB estándar. Esto permite capturar imágenes en color y video, lo que se utiliza para diversas aplicaciones, como videoconferencias y juegos que incorporan elementos del entorno del jugador en el juego.

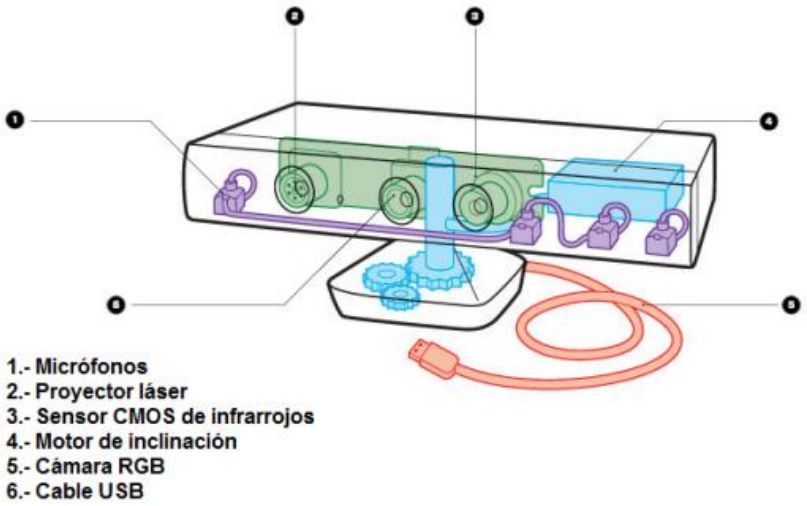

Figura 6 – Cámara Kinect de Xbox (ii)
Fuente: https://zaguan.unizar.es/record/12845/files/TAZ-PFC-2013-649_ANE.pdf

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.

Figura 7 – 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. Algunas de sus características y capacidades incluyen:

- Captura de Profundidad: Los sensores de profundidad permiten a las cámaras RealSense medir la distancia entre los objetos y la cámara, creando así un mapa tridimensional del entorno. Esto es útil para detectar objetos, personas y su ubicación en relación con la cámara.

- Reconocimiento de Gestos: Al igual que la tecnología Kinect, las cámaras RealSense son capaces de reconocer y seguir los gestos y movimientos de las personas. Esto permite a los usuarios interactuar con dispositivos y aplicaciones utilizando movimientos naturales.

- Reconocimiento Facial: Las cámaras RealSense pueden identificar y rastrear rostros en tiempo real. Esto es útil para aplicaciones de seguridad, autenticación y seguimiento de usuarios.

- Control por Voz: Además del reconocimiento visual, algunas versiones de las cámaras RealSense también incluyen capacidades de reconocimiento de voz, lo que permite a los usuarios controlar dispositivos y aplicaciones utilizando comandos de voz.

- Escaneo 3D: Algunas variantes de las cámaras RealSense tienen la capacidad de escanear objetos tridimensionales y convertirlos en modelos digitales que luego pueden ser utilizados en aplicaciones de diseño, impresión 3D y más.

- Aplicaciones Industriales y Robótica: Las cámaras RealSense también se utilizan en aplicaciones industriales y robóticas para permitir que los robots y las máquinas interactúen de manera más inteligente y precisa con su entorno.

- Realidad Virtual y Aumentada: Las cámaras RealSense son utilizadas por algunos dispositivos de realidad virtual y aumentada para rastrear el movimiento del usuario y permitir experiencias más inmersivas.

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.

Figura 8 – Guantes Senso Glove
Fuente: https://senso.me/dev

2.2.2. Identificación sistemas para la captura de posición de un objeto o pieza

Una segunda opción es 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: Dispositivos como las cámaras Intel RealSense, o Microsoft Kinect (identificadas anteriormente para capturar movimiento del operario), cámaras LiDAR y otros dispositivos de este tipo pueden capturar información tridimensional, incluyendo las coordenadas x, y, z de un objeto. Estos sensores utilizan principios como la triangulación láser o la medición del tiempo de vuelo para determinar la distancia entre el sensor y el objeto, lo que se utiliza para calcular las coordenadas.
  Estas cámaras pueden ofrecer una precisión en el rango de centímetros.

- Sistemas de Captura de Movimiento: Estos sistemas utilizan marcadores en el objeto para rastrear su movimiento en el espacio. Utilizan cámaras infrarrojas u ópticas para seguir la posición y la orientación de los marcadores, lo que permite calcular las coordenadas tridimensionales.

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: Estos sensores miden la orientación y los cambios en la velocidad y la aceleración de un objeto. Si se conocen las condiciones iniciales, se pueden integrar estas mediciones para obtener información sobre las coordenadas x, y, z del objeto.
  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 de realidad virtual y aumentada pueden utilizar sensores y cámaras para rastrear la posición y el movimiento de objetos virtuales y reales en el espacio tridimensional.
  La precisión en sistemas de realidad virtual y aumentada puede variar. Los sistemas más avanzados pueden ofrecer precisiones en el rango de milímetros a centímetros.

- Sistemas de Posicionamiento Global (GPS): Si el objeto se encuentra al aire libre, el GPS puede proporcionar coordenadas geográficas en un sistema de coordenadas tridimensional global. Sin embargo, la precisión del GPS puede variar y puede no ser adecuada para aplicaciones que requieran alta precisión en espacios reducidos.
  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.

2.3. Pruebas y desarrollo de sistemas

2.3.1. Sistemas para la captura de la posición de las manos

2.3.1.1. 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.

La aplicación se desarrolla bajo el lenguaje de programación Python 3.6. El programa procesa la secuencia de imágenes proveniente de la cámara de video con el módulo "Skeleton Tracker". Este módulo realiza una identificación de las personas que aparecen en una imagen (skeletons) así como identifica la posición de las distintas articulaciones y otros puntos de referencia (joints) de cada uno de los cuerpos identificados. La identificación se realiza en tiempo real pudiéndose observar en el monitor del sistema donde se ejecuta la misma.

Por último, 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.

Figura 9 – Montaje de la cámara Intel Realsense
Fuente: elaboración propia

2.3.1.2. *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.

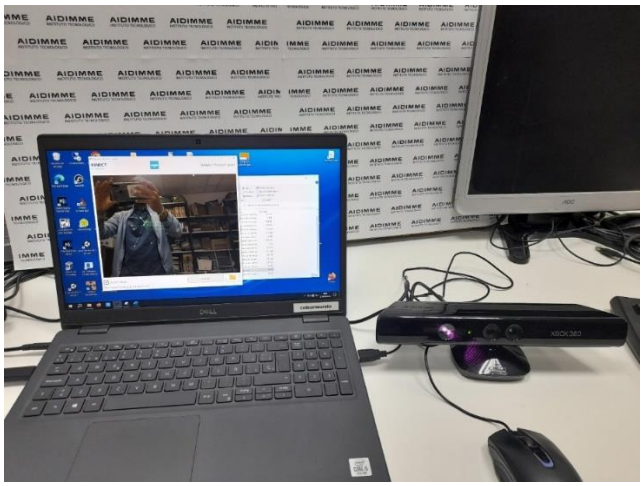

Figura 10 – Montaje de la cámara Kinect
Fuente: elaboración propia

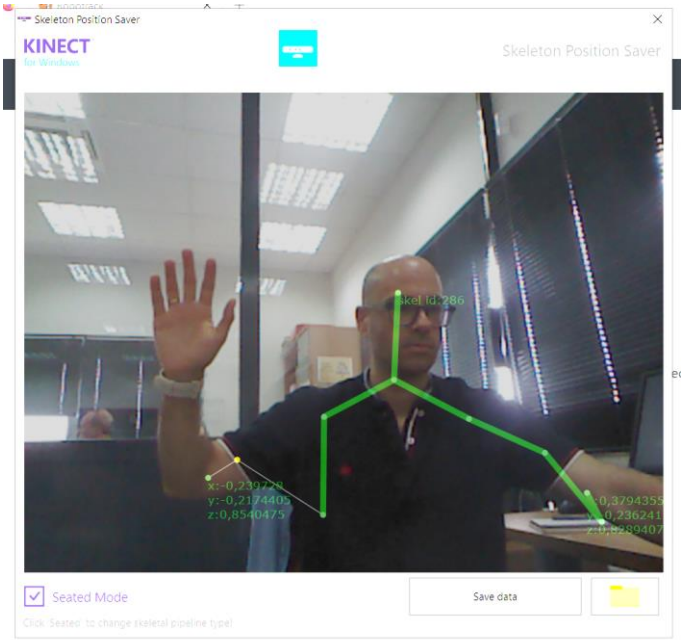

Figura 11 – Aplicación de prueba de cámara Kinect
Fuente: Skeleton Position Saver

El formato de la trama de datos guardada en el fichero .csv es el siguiente:

timestamp; skeletonid; [nº articulacion, x, y, z]

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

Las articulaciones detectadas (20) siguen este orden por su denominación en inglés en el SDK: HipCenter, Spine, ShoulderCenter, Head, ShoulderLeft, ElbowLeft, WristLeft, HandLeft, ShoulderRight, ElbowRight, WristRight, HandRight, HipLeft, KneeLeft, AnkleLeft, FootLeft, HipRight, KneeRight, AnkleRight, FootRight.

2.3.2. Desarrollo 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.

A continuación, se muestra un esbozo del esquema eléctrico del dispositivo.

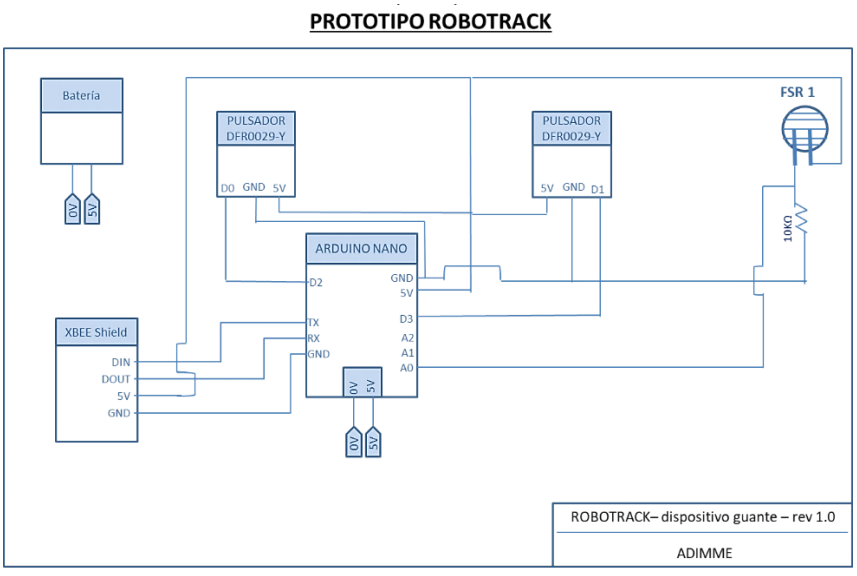

Figura 12 – Esquema eléctrico del guante de captura de acciones de usuario
Fuente: elaboración propia

Se ha montado 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.

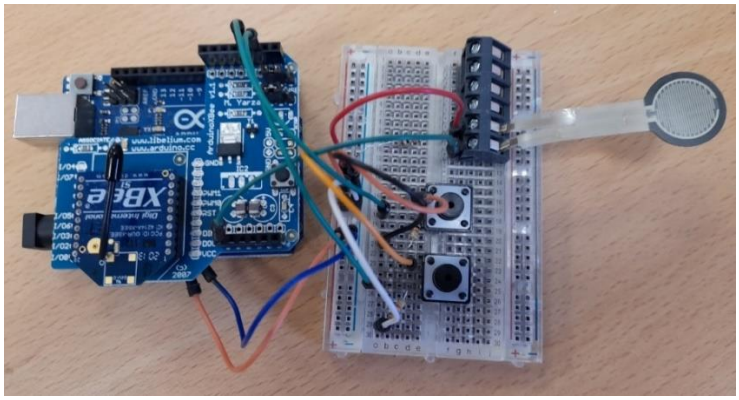

Figura 13 – Prototipo inicial del guante de captura de acciones de usuario
Fuente: elaboración propia

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, tal y como se muestra en la imagen:

	A	B	C	D	E
16	2023:5:25:10:17:38	D1	57	0	0
17	2023:5:25:10:17:38	D1	90	0	0
18	2023:5:25:10:17:38	D1	65	0	0
19	2023:5:25:10:17:39	D1	66	0	0
20	2023:5:25:10:17:39	D1	93	0	0
21	2023:5:25:10:17:39	D1	0	0	0
22	2023:5:25:10:17:39	D1	0	0	0
23	2023:5:25:10:17:39	D1	0	0	0
24	2023:5:25:10:17:39	D1	0	0	0
25	2023:5:25:10:17:39	D1	0	0	0
26	2023:5:25:10:17:39	D1	0	0	0
27	2023:5:25:10:17:39	D1	0	0	0
28	2023:5:25:10:17:39	D1	0	0	0
29	2023:5:25:10:17:40	D1	0	0	0
30	2023:5:25:10:17:40	D1	0	1	0
31	2023:5:25:10:17:40	D1	0	1	0
32	2023:5:25:10:17:40	D1	0	0	0
33	2023:5:25:10:17:40	D1	0	0	0
34	2023:5:25:10:17:40	D1	0	1	1
35	2023:5:25:10:17:40	D1	0	1	1
36	2023:5:25:10:17:40	D1	0	1	1
37	2023:5:25:10:17:40	D1	0	1	1
38	2023:5:25:10:17:40	D1	0	1	1
39	2023:5:25:10:17:41	D1	0	1	0
40	2023:5:25:10:17:41	D1	0	1	0
41	2023:5:25:10:17:41	D1	0	1	1
42	2023:5:25:10:17:41	D1	0	1	1
43	2023:5:25:10:17:41	D1	0	1	1
44	2023:5:25:10:17:41	D1	0	0	0

Robotrack_Guante_data

Figura 14 – Fichero de datos registrado por el dispositivo
Fuente: elaboración propia

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

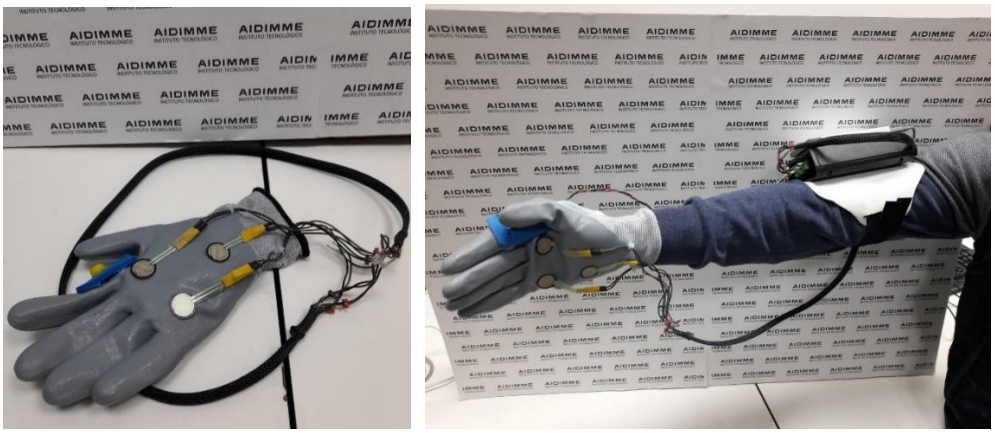

Figura 15 – Prototipo inicial instalado en el guante de captura de acciones de usuario
Fuente: elaboración propia

3. INTERFAZ HMI

3.1. 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.

Figura 16 – Tareas a ejecutar dentro de HMI Robotrack
Fuente: elaboración propia

Se detalla a continuación, todas las funcionalidades que inicialmente se identificaron necesarias para la interfaz Robotrack.

3.1.1. Capturar datos del proceso

US1.1.- Iniciar un nuevo proceso de aprendizaje por demostración

- El usuario debe poder iniciar una tarea de captura de datos de un nuevo proceso de aprendizaje por demostración.

AIDIMME INSTITUTO TECNOLÓGICO
GENERALITAT VALENCIANA
IVACE INSTITUTO VALENCIANO DE COMPETITIVIDAD EMPRESARIAL
Financiado por la Unión Europea

US1.2.- Seleccionar sistemas de captura de datos a utilizar

- El usuario debe poder indicar que sistemas de captura de datos va a utilizar. Se propone hacer clic sobre un listado inicial de dispositivos (cámara Intel, cámara Kinect, pulsadores, sensores presión, etc.).

US1.3.- Iniciar la grabación de datos

- Cuando todos los dispositivos están correctamente conectados, y han sido seleccionados en la HMI, el usuario podrá hacer clic sobre un botón que de inicio a la grabación y almacenaje de los datos que se capturen del operario realizando el proceso.

- Es posible que sea necesario habilitar esta acción mediante un pulsador en la zona de trabajo, para evitar que se capturen datos durante el tiempo que pasa desde que se hace clic en el PC hasta que el usuario inicia la tarea.

- A partir de este momento, el usuario podrá ejecutar la tarea a automatizar, y los datos asociados a la misma estarán siendo capturados.

- Sería útil ver en pantalla algún tipo de icono o mensaje que indique que se está en proceso de grabación de datos.

US1.4.- Finalizar la grabación de datos

- Cuando el usuario haya terminado la tarea, podrá detener la grabación haciendo clic en un botón habilitado en la HMI.

- Es posible que sea necesario habilitar esta acción mediante un pulsador en la zona de trabajo, para evitar que se capturen datos durante el tiempo que pasa desde que finaliza la tarea hasta que se desplaza al PC y hace clic en el botón.

- Sería útil algún tipo de icono o mensaje que permita confirmar visualmente que la grabación de datos se ha detenido.

US1.5.- Guardar el archivo/s de datos

- Sería útil algún tipo de icono o mensaje que permita confirmar visualmente que los datos se han guardado correctamente.

3.1.2. Revisar datos grabados en bruto

US2.1.- Revisar información básica de la grabación

- El usuario podrá ver una representación de la trayectoria en bruto de los puntos grabados.

US2.3.- Validación de los archivos grabados

- En función de la información de los archivos el usuario puede detectar que hubo un error durante la grabación de los datos, y descartar esos archivos. O bien, los da por buenos y los válida para pasar a la siguiente fase del aprendizaje por demostración.

3.1.3. Crear trayectoria

US3.1.- Escribir los parámetros del algoritmo que genera la nueva trayectoria

- El usuario debe poder escribir Los parámetros que requiere el algoritmo para generar la nueva trayectoria:
    - FUERZA DE SUAVIZADO => hace referencia a cuantos puntos coge el algoritmo para realizar una media móvil. Será un número entero.
    - GRANULARIDAD => hace referencia a la distancia mínima entre puntos en metros. Por ejemplo: si el usuario escribe 0’05, significa que el algoritmo descartará los puntos que estén a menos de 5 cm. Será un número decimal (porque va en metros).
    - LARGO Y ANCHO => hace referencia a las dimensiones en metros del área de trabajo del robot. El algoritmo realiza un escalado con estos datos. Será un número decimal.

US3.2.- Visualizar el resultado de la trayectoria generada

- El usuario debe poder visualizar el aspecto de la trayectoria generada por IA para valorar si se parece a la trayectoria ejecutada manualmente por el operario

US3.3.- Validación de la trayectoria generada

- En función de la información visualizada en pantalla el usuario decide continuar con el proceso de aprendizaje por demostración, o abandonar el proceso iniciado con esos datos e iniciar uno nuevo. Si la trayectoria generada no fuera satisfactoria el usuario debería poder repetir la generación de la trayectoria cambiando la GRANULARIDAD o la FUERZA DE SUAVIZADO.

3.1.4. Generar programa robot

US4.1.- Visualizar que se ha generado el archivo Python genérico

- El usuario debe poder visualizar que se está generando el archivo Python genérico, a partir de los puntos de trayectoria del apartado anterior.

- El usuario debe poder visualizar en un mensaje, que el archivo se ha generado finalmente.

US4.2.- Visualizar que se ha generado el script de programa de robot específico

- El usuario debe poder visualizar que se está generando el script final de programa de robot (en el lenguaje nativo de un robot, por ejemplo, UR script), a partir del fichero Python anterior.

- El usuario debe poder visualizar en un mensaje, que el archivo se ha generado finalmente.

US4.3.- Validación del programa de programa de robot

- En función de la información visualizada en pantalla el usuario decide continuar con el proceso de aprendizaje por demostración, o abandonar el proceso iniciado con esos datos e iniciar uno nuevo.

3.1.5. Generar simulación

US5.1.- Visualizar que se ha generado el archivo de la simulación

- El usuario debe poder visualizar que se está generando el archivo de la simulación, a partir del fichero Python de la etapa anterior.

- El usuario debe poder visualizar en un mensaje, que el archivo se ha generado finalmente.

US5.2.- Visualizar la simulación

- El usuario debe poder visualizar la simulación generada. Ver el video de la simulación del programa de robot

US5.3.- Validación del programa de programa de robot

- En función de la información visualizada en pantalla el usuario decide continuar con el proceso de aprendizaje por demostración, o abandonar el proceso iniciado con esos datos e iniciar uno nuevo.

3.1.6. Transferir programa a robot

US6.1.- Transferir programa al robot

- El usuario debe poder transferir el script generado con anterioridad al robot.

- Sería interesante recibir un mensaje de confirmación de que el archivo ha sido transferido correctamente.

3.2. Diseño interfaz HMI

A continuación, se exponen algunas mockups del diseño inicial de la HMI para cumplir con las funcionalidades expuestas en el apartado anterior.

3.2.1. 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.

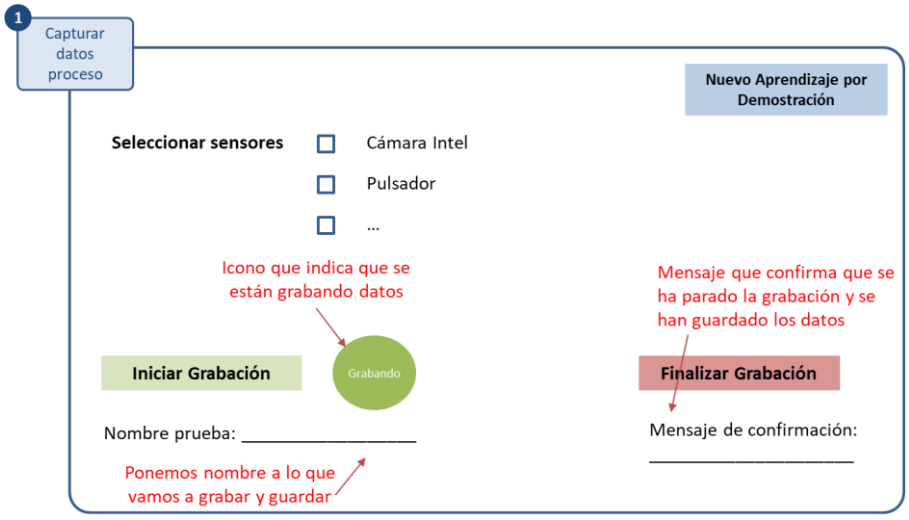

Figura 17 – Mockup del proceso de captura de datos
Fuente: elaboración propia

3.2.2. 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.

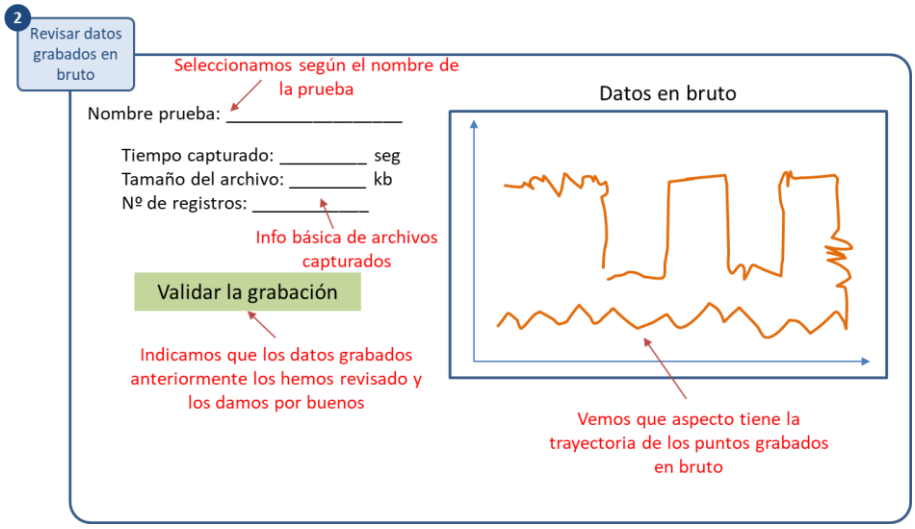

Figura 18 – Mockup del proceso de revisión de datos grabados
Fuente: elaboración propia

3.2.3. Crear trayectoria

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

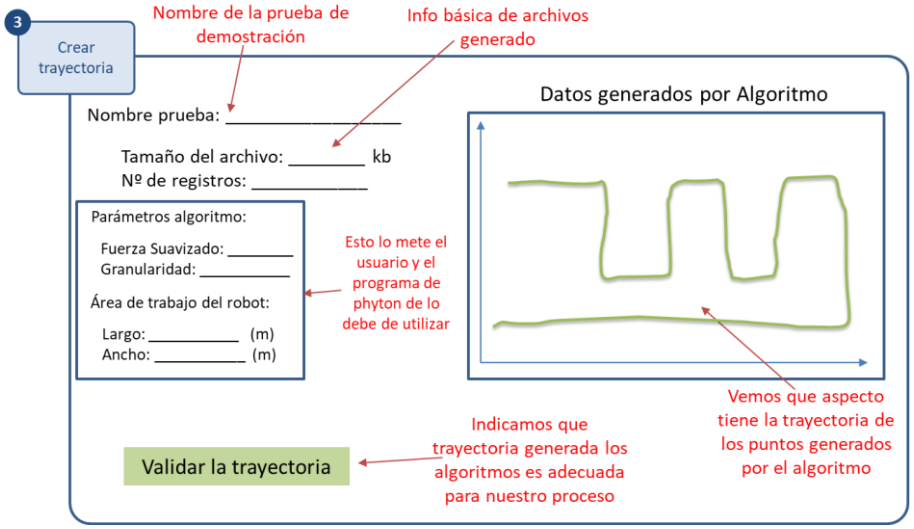

Figura 19 – Mockup del proceso de generación de trayectoria de robot
Fuente: elaboración propia

3.2.4. 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.

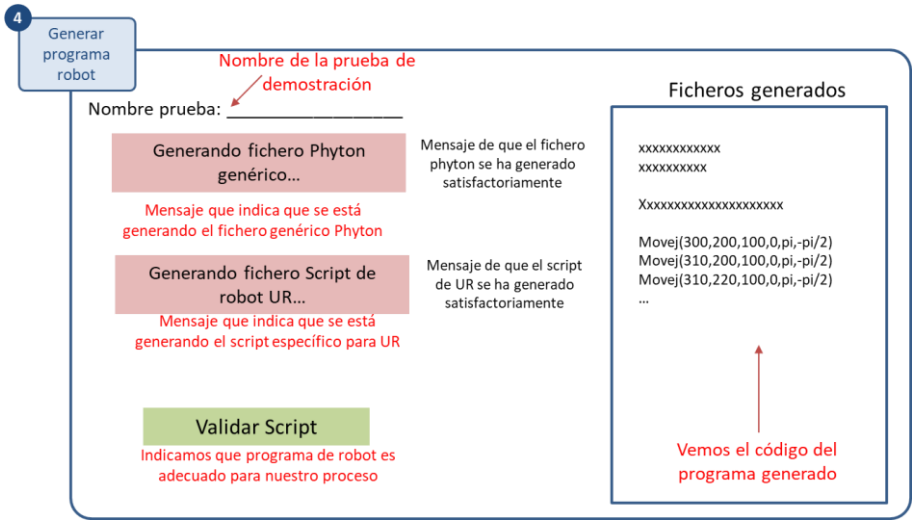

Figura 20 – Mockup del proceso de generación del programa de robot
Fuente: elaboración propia

3.2.5. 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.

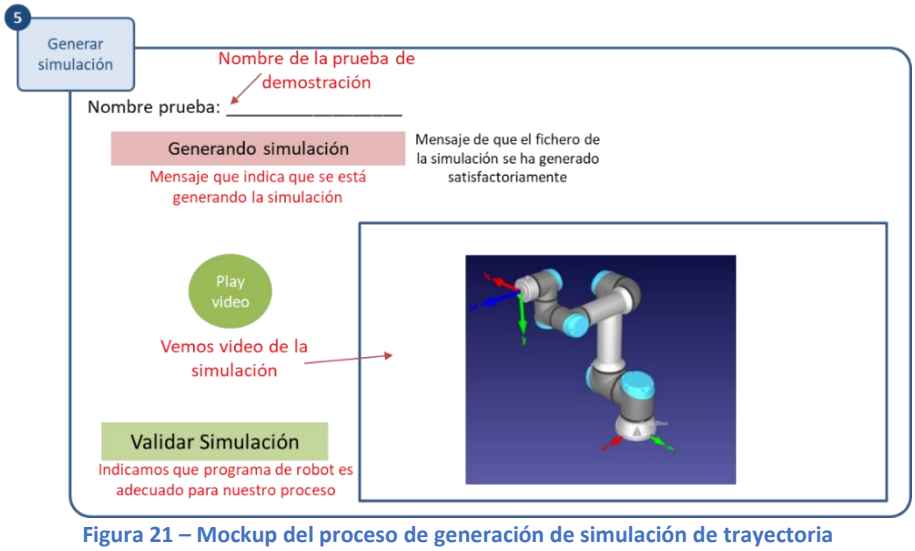

Figura 21 – Mockup del proceso de generación de simulación de trayectoria
Fuente: elaboración propia

3.2.6. TRANSFERIR PROGRAMA A ROBOT

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

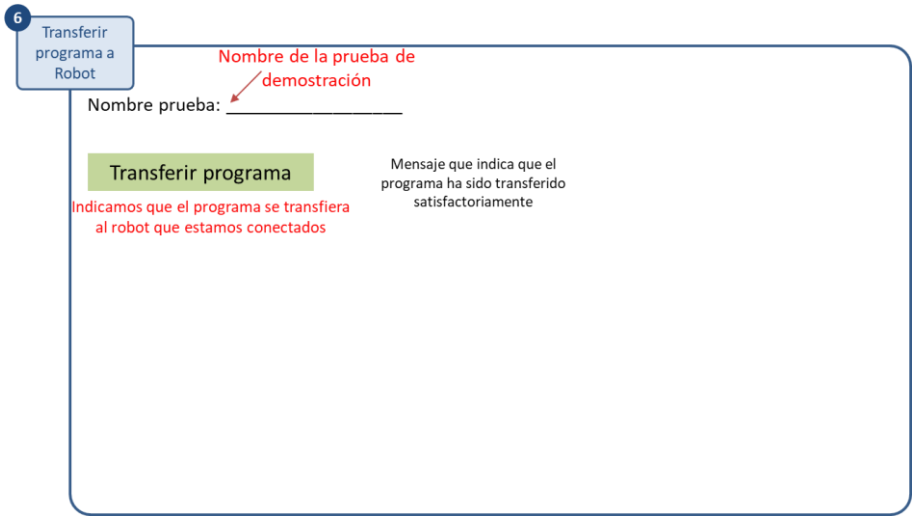

Figura 22 – Mockup del proceso de transferencia del programa de robot
Fuente: elaboración propia

3.3. 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. A lo largo de esta sección se explicará el desarrollo de cada uno de ellos y así como su funcionamiento.

Las aplicaciones web dividen su ejecución en 2 computadores, un cliente, el ordenador que utiliza el usuario final, y un servidor, un ordenador donde se almacena la aplicación, la base de datos y toda la información relativa a esta. Frontend hace referencia a la parte de la aplicación que se ejecuta en el cliente, mientras que el backend es la parte de la aplicación que funciona en el servidor.

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. Sin embargo, sobre estos 3 elementos básicos, existen multitud de herramientas como librerías o frameworks que facilitan y agilizan el desarrollo de este tipo de aplicaciones. En particular, para el desarrollo de esta aplicación se ha utilizado la librería React.js. React permite estructurar la aplicación web en elementos reutilizables conocidos como componentes, a la vez que implementa un sistema de variables y propiedades para controlar el estado de la aplicación en tiempo real. Se ha elegido React, en contraposición a otras tecnologías que ofrecen características similares como Vue o Angular, debido a la experiencia de los desarrolladores de AIDIMME con esta tecnología.

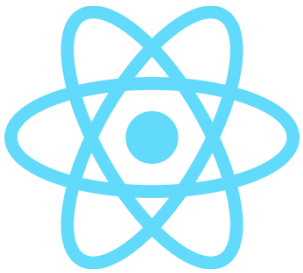

Figura 23 - Logo de React
Fuente: https://es.react.dev/

Los componentes de React se estructuran mediante una jerarquía padre – hijo, creando lo que se conoce como árbol de componentes. De esta forma, la aplicación se estructura sobre un componente llamado procesador. Éste mostrará a sus distintos hijos dependiendo del punto del proceso en el que se encuentre el usuario que podrá seleccionar o modificar las opciones de manera sencilla.

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.

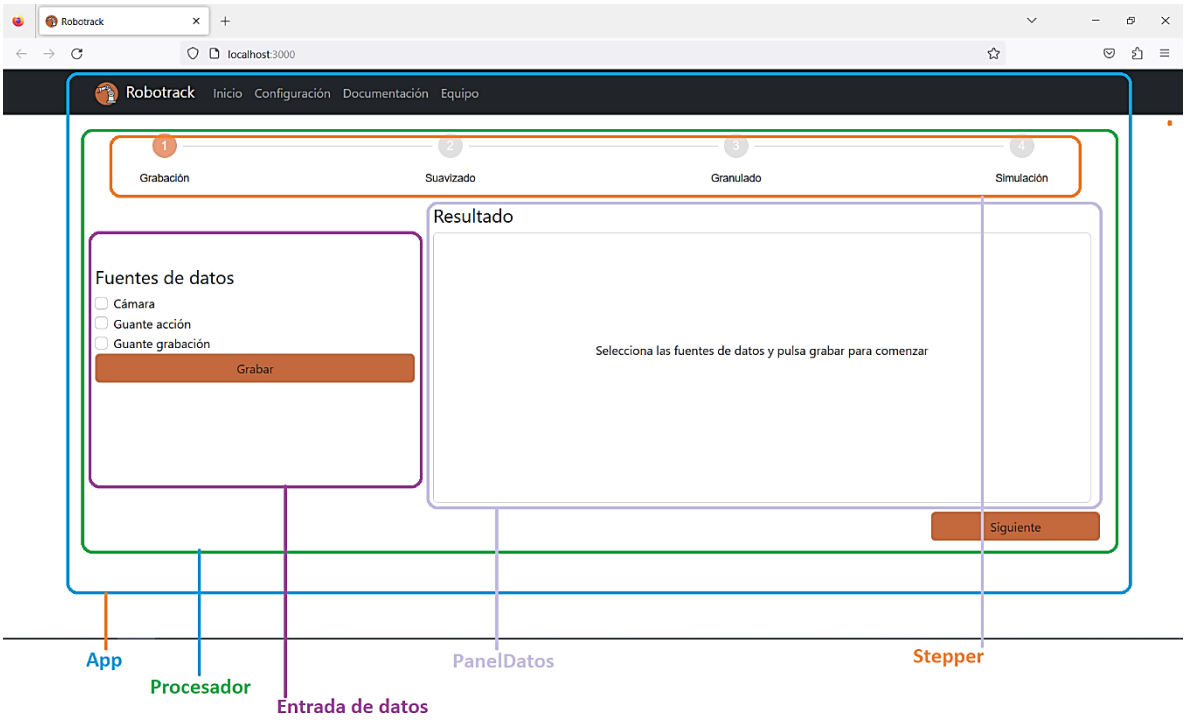

Figura 24 - Jerarquía de componentes de la aplicación
Fuente: elaboración propia

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.

Figura 25 – 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.

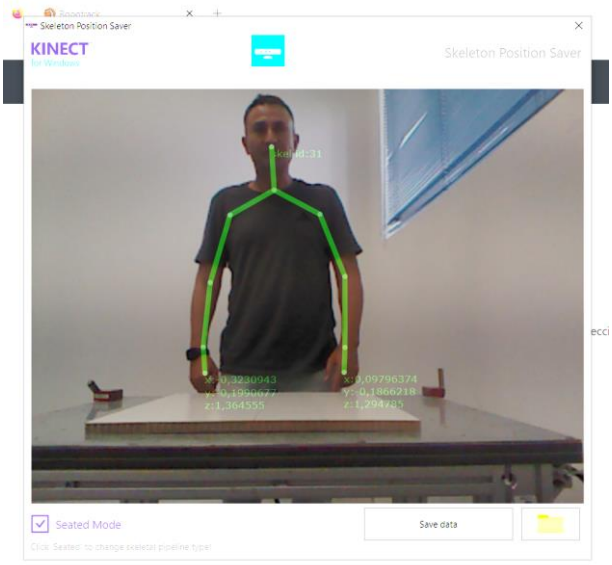

Figura 26 – Funcionamiento del sistema de captura de posición de las manos
Fuente: elaboración propia

AIDIMME INSTITUTO TECNOLÓGICO
GENERALITAT VALENCIANA
iVACE INSTITUTO VALENCIANO DE COMPETITIVIDAD EMPRESARIAL
Financiado por la Unión Europea

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.

Figura 27 – 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.

Figura 28 - Logo de Flask
Fuente: https://flask.palletsprojects.com/en/2.3.x/

Python incluye dos frameworks populares para la creación de backends: Flask y Django. Ambos frameworks siguen filosofías distintas, Django incluye muchas funcionalidades "*out of the box*", es decir, que el desarrollador no tiene que emplear tiempo en desarrollar algunas de las funcionalidades más comunes dentro del desarrollo web como un sistema de conexión con la base de datos, la metodología de autenticación, el ruteado o el modelado de datos. Por otra parte, Flask sigue la idea contraria, es un framework que basa su funcionamiento en la sencillez y facilidad de desarrollo, suponiendo esto que es un sistema mucho menos ligero al no incluir tantas herramientas desde la instalación. Dado que muchas de las funcionalidades que aporta Django son innecesarias para el proyecto, se ha decidido utilizar Flask, primando así la ligereza del sistema y la sencillez de desarrollo.

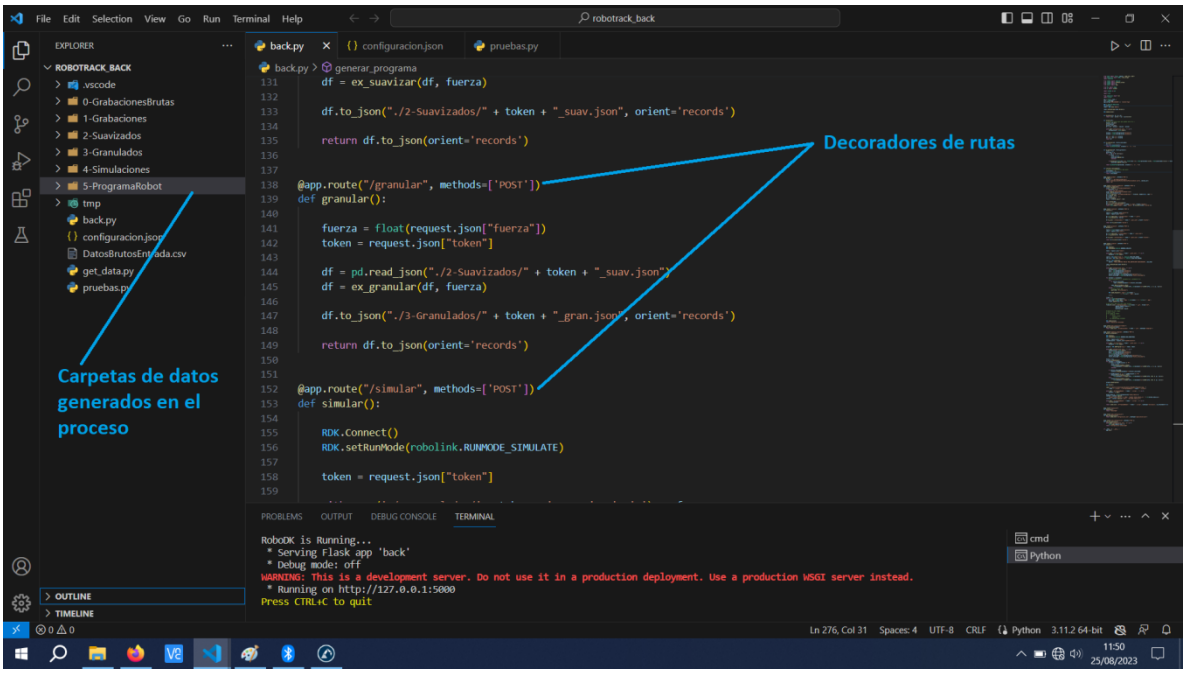

Figura 29 - Código Flask del backend y estructura de carpetas
Fuente: elaboración propia

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.

C. Resumen y conclusiones

Se ha llevado a cabo un análisis de diferentes procesos llevados a cabo de manera manual, que pueden llegar a ser automatizados mediante la implantación de robots colaborativos y que tendría sentido llegar a realizar el programa del robot mediante un proceso de demostración humana.

Para llevar a cabo la fase de validación de los desarrollos del proyecto se selecciona como piloto demostrativo los procesos de lijado de piezas planas. Sobre este proceso se han revisado los sistemas de captura de datos que se requerirían.

Se ha desarrollado un sistema de captura de la trayectoria de las manos del usuario (registra las coordenadas x, y, z) mediante un sistema de visión. Para gobernar este sistema se ha desarrollado una aplicación que simplifica el funcionamiento del sistema al usuario.

El resultado obtenido es un sistema de captura de datos de la trayectoria seguida por las manos del usuario que realice la demostración basado en un sistema de visión, así como una interfaz de usuario propia que asiste al usuario y gobierna este sistema.

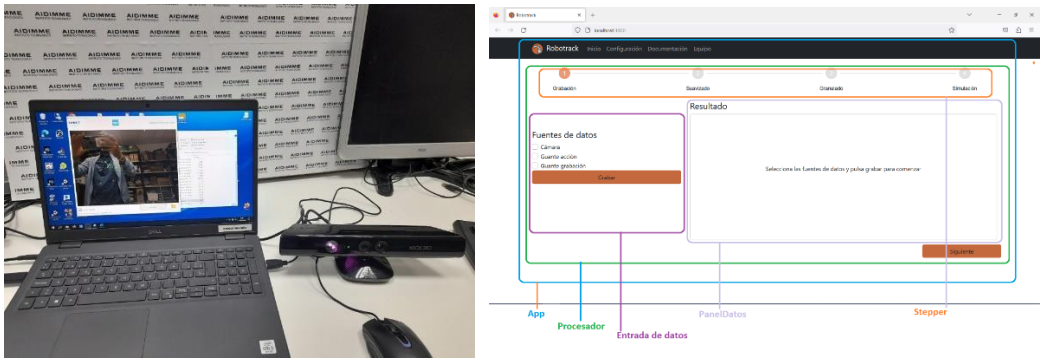

Figura 30 – Sistema de captura de datos y pantalla inicial de la interfaz HMI
Fuente: elaboración propia

Con la colaboración de:

AIDIMME
INSTITUTO TECNOLÓGICO

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

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

aidimme@aidimme.es
www.aidimme.es