



Escuela
Politécnica
Superior

Entorno virtual para tareas de manipulación y agarre con un brazo robótico asistencial



Máster Universitario en Automática y
Robótica

Trabajo Fin de Máster

Autora:

Míriam Máximo Gutiérrez

Tutor:

Andrés Úbeda Castellanos

Julio 2022



Universitat d'Alacant
Universidad de Alicante

Entorno virtual para tareas de manipulación y agarre con un brazo robótico asistencial

Control mioeléctrico de un entorno virtual de robótica asistencial

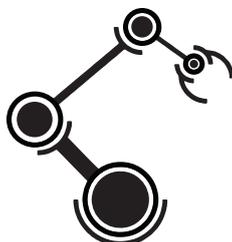
Autora

Miriam Máximo Gutiérrez

Tutor

Andrés Úbeda Castellanos

Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal



Máster Universitario en Automática y Robótica



Escuela
Politécnica
Superior



Universitat d'Alacant
Universidad de Alicante

ALICANTE, Julio 2022

Resumen

En la actualidad se han desarrollado multitud de tecnologías asistenciales que permiten a las personas que poseen algún tipo de discapacidad realizar actividades que no pueden realizar de manera autónoma. Este trabajo se centrará en las tecnologías relacionadas con la robótica asistencial.

Con este proyecto se pretende dar asistencia en actividades de la vida diaria, tales como el agarre y manipulación de objetos, para su uso por parte de personas con amputación de miembro superior, ya que es una tarea que se realiza de manera continuada para ejecutar múltiples cometidos.

El control utilizado para dirigir los movimientos del robot será un control mediante señales mioeléctricas. Este método está basado en acciones de control gobernadas por contracciones musculares. Este tipo de control requiere de un entrenamiento para dominar el sistema de manera óptima. Para este entrenamiento se ha diseñado un entorno virtual.

Este entorno virtual será ambientado como un entorno doméstico con el objetivo de aumentar la sensación de inmersión por parte del usuario. En dicho entorno se dispone de un brazo robótico asistencial el cuál posee cuatro grados de libertad, debido al movimiento de tres articulaciones prismáticas y al agarre de la pinza que se encuentra en su efector final. En este entorno encontraremos distintos objetos los cuales el usuario tendrá la posibilidad de manipular. Además, se han implementado distintos modos con el objetivo de obtener diferentes maneras de control del robot, cada una de las cuales poseen un grado de autonomía diferente por parte del robot.

Finalmente, con el objetivo de evaluar el funcionamiento del sistema de control mioeléctrico en el entorno virtual, se han realizado ensayos con distintos usuarios en los que se han analizado distintos parámetros, así como las experiencias de los sujetos. A partir de estos datos se han obtenido varias conclusiones que nos permiten valorar el sistema.

Abstract

Nowadays, many assistive technologies have been developed to enable people with disabilities to independently perform activities that they cannot do for themselves. This project focuses on the field of assistive robotics.

The implementation of this project intends to help people with an upper limb amputation in activities of daily living, such as grasping and handling objects, as this is an activity that is constantly did to perform multiple tasks.

The robot's movements are controlled by myoelectric signals based on the action of muscle contractions. This type of control requires training to be able to control the system optimally. A virtual environment was created for this training.

This virtual environment has been configured as a domestic environment to increase the user's sense of immersion. In this environment there is an assistive robotic arm with four degrees of freedom, due to the movement of three prismatic joints and the grasp of the gripper in its effector joint. In this environment we find different objects that the user can handle. In addition, different modes have been implemented to provide different ways to control the robot, each providing a different degree of autonomy for the robot.

Finally, to evaluate the functioning of the myoelectric control system in the virtual environment, tests were conducted with different users, analyzing different parameters and user experiences. From these data, several conclusions were drawn that allow an evaluation of the system.

Preámbulo

“La razón principal que ha llevado a la realización de este proyecto es el interés de encontrar métodos que puedan asistir a personas que por diversas causas como accidentes o enfermedades disponen de alguna discapacidad debida a la cuál carecen de las habilidades necesarias para realizar actividades de la vida diaria.

Muchas de las actividades de la vida diaria conllevan la manipulación y agarre de objetos, por ello con este proyecto se pretende conseguir un sistema de entrenamiento para personas con amputación de miembro superior, con el fin de que aprendan a realizar correctamente el control de un brazo robótico mediante el cual llevar a cabo estas actividades.”

Agradecimientos

En primer lugar, gracias a mi tutor de TFM, Andrés Úbeda, por brindarme los conocimientos y los medios necesarios para llevar a cabo este trabajo, y por desarrollar este tipo de investigaciones que buscan la inclusión de personas con distintas capacidades. Así como al resto de profesores con los que he aprendido tanto durante este año.

En segundo lugar, agradecer a Cristina, mi compañera en este trabajo, por su duro esfuerzo y perseverancia durante estos meses. Agradecer también a mis compañeros de clase, los cuales me han acompañado este año y me han ofrecido siempre la ayuda que he necesitado.

También, me gustaría agradecer a todas las personas que han dedicado su tiempo para realizar las pruebas como usuarios en este proyecto, gracias por su tiempo y ayuda.

Por último, agradecer a mi familia, mi pareja y mis amigos, por todo el apoyo y comprensión que he recibido siempre de ellos.

*No vayas por donde el camino te lleve.
Ve en cambio por donde no hay camino y deja rastro.*

Ralph Waldo Emerson

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Estructura de la memoria	2
2. Objetivos	3
3. Marco Teórico	5
3.1. Tecnologías de asistencia	5
3.1.1. Definición y áreas de aplicación	5
3.1.2. Robótica asistencial	6
3.2. Realidad virtual	9
3.2.1. Definición y componentes	9
3.2.2. Historia de la Realidad Virtual	11
3.2.3. Clasificación de los Sistemas de Realidad Virtual	12
3.3. Realidad Virtual como tecnología de asistencia	13
4. Planificación del proyecto	17
5. Metodología y materiales empleados	19
5.1. Metodología	19
5.2. Materiales empleados	19
5.2.1. Ordenador	20
5.2.2. Noraxon	20
5.2.3. Matlab	21
5.2.4. Unity	22
5.2.4.1. Características generales	22
5.2.4.2. Interfaz de Unity	22
5.2.4.3. Elementos de Unity	23
5.2.4.4. Físicas en Unity	24
6. Desarrollo del software	29
6.1. Lectura de la señal muscular	29
6.2. Procesamiento de la señal	30
6.2.1. Comunicación entre Matlab y Unity	30
6.3. Entorno virtual del robot asistencial	31
6.3.1. Elementos de la escena	31
6.3.2. Robot	34
6.3.3. Modos de uso	37
6.3.3.1. Modo 1: Movimiento libre en tres ejes y grasping manual	38

6.3.3.2. Modo 2: Movimiento libre en dos ejes y grasping automático	41
6.3.3.3. Modo 3: Movimiento y grasping automático	44
7. Pruebas de usabilidad	49
7.1. Protocolo de pruebas	49
7.2. Métricas obtenidas	53
7.3. Resultados obtenidos	54
7.3.1. Análisis a nivel del usuario	54
7.3.1.1. Usuario 1	55
7.3.1.2. Usuario 2	56
7.3.1.3. Usuario 3	58
7.3.1.4. Usuario 4	59
7.3.1.5. Usuario 5	61
7.3.1.6. Usuario 6	62
7.3.1.7. Usuario 7	63
7.3.1.8. Usuario 8	65
7.3.2. Análisis a nivel de modo	67
7.3.3. Análisis global	68
7.3.4. Formularios	68
7.4. Conclusiones de los resultados	70
8. Conclusiones y posibles mejoras	73
8.1. Conclusiones	73
8.2. Posibles mejoras	74
Bibliografía	75
Lista de Acrónimos y Abreviaturas	79
A. Anexo I: Diagrama de Gantt	81
B. Anexo II: Resultados de las pruebas	83
C. Anexo III: Resultados de las encuestas de los usuarios	89

Índice de figuras

3.1. Robot Care-o-Bot. <i>Fuente:[1]</i>	6
3.2. Silla de ruedas Panasonic. <i>Fuente: [2]</i>	7
3.3. Prótesis robótica de mano Michelangelo. <i>Fuente: [3]</i>	7
3.4. Exoesqueleto de miembro superior MyoPro. <i>Fuente: [4]</i>	8
3.5. Robot Kinova Jaco. <i>Fuente: [5]</i>	8
3.6. Robot UR3. <i>Fuente: [6]</i>	9
3.7. Elementos de un sistema de realidad virtual	9
3.8. Dispositivos de entrada de sistemas de realidad virtual <i>Fuentes: [7] [8]</i>	10
3.9. Dispositivos de salida de sistemas de realidad virtual. <i>Fuentes: [9] [10]</i>	10
3.10. Gafas estereoscópicas. <i>Fuente: [11]</i>	11
3.11. Shuterland con el dispositivo "Sword of Damocles". <i>Fuente: [12]</i>	12
3.12. Plataforma de movimiento, sistema háptico y silla de una interfaz. <i>Fuente: [13]</i>	14
3.13. Uso de dispositivo de realidad virtual en conjunto con el control mioeléctrico. <i>Fuente: [14]</i>	15
3.14. Elementos del sistema de control robótico controlado por la lengua <i>Fuente: [15]</i>	15
4.1. Análisis DAFO	17
4.2. Análisis CAME	18
5.1. Recursos utilizados en la realización del trabajo	20
5.2. Interfaz de Noraxon MR3	21
5.3. Interfaz de Matlab	21
5.4. Interfaz de Unity	23
5.5. Componente Character Controller en Unity. <i>Fuente: [16]</i>	24
5.6. Componente Rigidbody en Unity. <i>Fuente:[16]</i>	25
5.7. Componente Box Collider en Unity. <i>Fuente:[16]</i>	25
5.8. Componente Character Joint en Unity. <i>Fuente:[16]</i>	25
5.9. Jerarquía de un Articulation Body en Unity. <i>Fuente:[16]</i>	26
5.10. Componente Articulation Body en Unity. <i>Fuente:[16]</i>	26
6.1. Diagrama general de la funcionalidad de cada uno de los elementos del proyecto	29
6.2. Interfaz de Matlab en el modo 1	30
6.3. Entorno virtual creado en Unity	32
6.4. Objetos del entorno virtual	33
6.5. Caja del entorno virtual	33
6.6. Diseño virtual del robot UR3	34
6.7. Elementos que componen el brazo robótico	35
6.8. Componentes Articulation Body de cada uno de los tres ejes de movimiento del brazo robótico	36

6.9. Posición de los objetos MovX, MovY y MovZ en el espacio	36
6.10. Flujograma de funcionamiento de Unity en modo 1	39
6.11. Feedback visual del estado de la pinza del brazo robótico	40
6.12. Flujograma de funcionamiento de Unity en modo 2	43
6.13. Interfaz para la selección de distintas actividades a realizar por el brazo robótico	44
6.14. Flujograma de funcionamiento de Unity en modo 3	45
7.1. Lectura de una flexión	51
7.2. Lectura de una extensión	51
7.3. Lectura de una cocontracción	52
7.4. Resultados de los tiempos de las pruebas del usuario 1 con respecto al promedio.	56
7.5. Resultados de los tiempos de las pruebas del usuario 2 con respecto al promedio.	57
7.6. Resultados de los tiempos de las pruebas del usuario 3 con respecto al promedio.	59
7.7. Resultados de los tiempos de las pruebas del usuario 4 con respecto al promedio.	60
7.8. Resultados de los tiempos de las pruebas del usuario 5 con respecto al promedio.	62
7.9. Resultados de los tiempos de las pruebas del usuario 6 con respecto al promedio.	63
7.10. Resultados de los tiempos de las pruebas del usuario 7 con respecto al promedio.	65
7.11. Resultados de los tiempos de las pruebas del usuario 8 con respecto al promedio.	66
7.12. Resultados de los tiempos promedio en cada objeto para cada modo	67
7.13. Tasa de acierto respecto a todas las pruebas realizadas	68

Índice de tablas

7.1. Datos obtenidos para el usuario 1	55
7.2. Datos obtenidos para el usuario 2	57
7.3. Datos obtenidos para el usuario 3	58
7.4. Datos obtenidos para el usuario 4	60
7.5. Datos obtenidos para el usuario 5	61
7.6. Datos obtenidos para el usuario 6	62
7.7. Datos obtenidos para el usuario 7	64
7.8. Datos obtenidos para el usuario 8	65
7.9. Datos promedios obtenidos por modos	67
7.10. Tiempo total de duración de las pruebas	69
B.1. Datos obtenidos para el objeto taza en el modo 1	83
B.2. Datos obtenidos para el objeto manzana en el modo 1	83
B.3. Datos obtenidos para el objeto botella en el modo 1	84
B.4. Datos obtenidos para el objeto refresco en el modo 1	84
B.5. Datos obtenidos para el objeto croissant en el modo 1	84
B.6. Datos obtenidos para el objeto taza en el modo 2	85
B.7. Datos obtenidos para el objeto manzana en el modo 2	85
B.8. Datos obtenidos para el objeto botella en el modo 2	86
B.9. Datos obtenidos para el objeto refresco en el modo 2	86
B.10. Datos obtenidos para el objeto croissant en el modo 2	86
B.11. Datos obtenidos para el objeto taza en el modo 3	87
B.12. Datos obtenidos para el objeto manzana en el modo 3	87
B.13. Datos obtenidos para el objeto botella en el modo 3	88
B.14. Datos obtenidos para el objeto refresco en el modo 3	88
B.15. Datos obtenidos para el objeto croissant en el modo 3	88

1. Introducción

1.1. Motivación

Las tecnologías asistenciales permiten a muchas personas realizar actividades de la vida diaria, las cuáles no pueden realizar por sí mismos debido a que poseen alguna discapacidad. En la actualidad estas tecnologías están ampliamente utilizadas y ayudan en tareas tales cómo comer, desplazarse o incluso, comunicarse.

En este trabajo se va a centrar la atención en actividades de manipulación y agarre de objetos para su uso por parte de personas con amputación de miembro superior, ya que es una tarea que en la vida diaria se realiza de manera continuada para ejecutar múltiples cometidos. Para ello se va a hacer uso de un robot asistencial, estos dispositivos son utilizados para realizar tareas que el usuario no puede realizar sin su ayuda, entre ellos nos encontramos robots de guiado, sillas de ruedas autónomas, prótesis, exoesqueletos y brazos robóticos de asistencia. Será este último el que será utilizado, por sus características, siendo similar a un brazo humano, para agarrar objetos.

Para el control de este brazo robótico se ha seleccionado un control mediante señales mioeléctricas, ya que este método se adapta a las posibilidades de personas con amputación de miembro superior. Este método basa sus acciones de control en las contracciones de un determinado músculo en el que se han colocado sensores que realizarán estas mediciones.

El uso de tecnologías de asistencia con el tipo de control mencionado posee una alta tasa de abandono por parte de los usuarios, ya que el control no es el óptimo para cada usuario, además familiarizarse con este tipo de control requiere de entrenamiento. Para ello se ha diseñado un entorno virtual con un brazo robótico asistencial y elementos cotidianos, en el cuál el usuario puede, mediante el desplazamiento y acción del brazo robótico y las pinzas que posee, agarrar y manipular los diferentes objetos del entorno.

El trabajo que se expone a continuación es una parte del proyecto “Control mioeléctrico de un entorno virtual de robótica asistencial” elaborado en colaboración con Cristina Polo Hortigüela, la cual en su Trabajo Final de Máster (TFM) titulado “Evaluación y validación de control mioeléctrico en un entorno virtual de robótica asistencial” expone de qué manera se ha realizado el control mioeléctrico [17]. En este TFM se mostrará el software desarrollado para crear este entorno virtual. Las pruebas realizadas así como su evaluación se mostrarán en ambos trabajos.

1.2. Estructura de la memoria

Esta memoria está estructurada en diferentes apartados en los que se pondrá en antecedentes al lector para posteriormente explicar los contenidos que comprenden el proyecto realizado. Algunos de estos apartados aparecen explicados de manera más ampliada y con más detalle en el TFM en colaboración con este [17].

En primer lugar, en el apartado 2 se mostrarán los objetivos que se pretenden alcanzar con la realización del proyecto. Posteriormente, en el apartado 3 se explicarán los conceptos teóricos que intervienen en este proyecto, haciendo un recorrido por estudios similares a este que se han llevado a cabo.

Más tarde, en el apartado 4 se mostrará la planificación planteada que se ha seguido para ejecutar cada una de las partes que componen el trabajo. Seguidamente en el apartado 5 se mostrará tanto la metodología empleada como cada uno de los materiales de los que se ha hecho uso para elaborar el trabajo.

En el apartado 6 se expondrá el software que finalmente se ha desarrollado. Las pruebas realizadas con este software serán detalladas en el apartado 7, tanto los procedimientos llevados a cabo como los resultados obtenidos. Finalmente esta memoria terminará en el apartado 8 con la exposición de las conclusiones que se han obtenido.

2. Objetivos

Los objetivos generales del proyecto en conjunto, es decir, tanto este proyecto como el relativo al estudio del control mioeléctrico, se detallan a continuación:

- Implementación de un control mioeléctrico, evaluando diferentes algoritmos.
- Diseño de un entorno virtual con un brazo robótico asistencial, capaz de realizar actividades de manipulación y agarre.
- Creación de una interfaz para la retroalimentación del control mioeléctrico.
- Evaluación del sistema mediante la realización de pruebas con diferentes usuarios.

Por otra parte, los objetivos específicos de este proyecto son los siguientes:

- Diseño de un escenario con elementos de la vida diaria en Unity, con las físicas necesarias para el desarrollo de las actividades previstas.
- Implementación de un robot articulado asistencial en la interfaz virtual en Unity.
- Programación de distintos modos de funcionamiento del robot asistencial para la realización de tareas de agarre y manipulación de los elementos de la interfaz.
- Implementación de biofeedback de las señales electromiográficas.
- Estudio de los mensajes recibidos procedentes de Matlab para la realización de las diferentes actividades.
- Estudio de los resultados cuantitativos y cualitativos obtenidos para los usuarios para su análisis desde un punto de vista de utilidad para el usuario final.
- Analizar las diferencias entre los modos implementados con los datos obtenidos en las pruebas.

3. Marco Teórico

En este apartado se van a definir algunos de los conceptos que han de conocerse para la realización de este trabajo. Además, se va a hacer un recorrido histórico y revisión de algunos estudios similares.

En primer lugar, se va a hacer un análisis de las tecnologías de asistencia, tanto su definición y áreas de aplicación, cómo algunos ejemplos de robótica asistencial. Luego, se va definir el concepto de realidad virtual y los componentes que lo conforman, historia de la misma y su uso como tecnología de asistencia.

3.1. Tecnologías de asistencia

3.1.1. Definición y áreas de aplicación

Las tecnologías de asistencia o tecnologías de apoyo, se definen según la Asociación de la Industria de las Tecnologías de Asistencia (ATIA), como “cualquier artículo, pieza de equipo, programa de software o sistema de producto que se utiliza para aumentar, mantener o mejorar las capacidades funcionales de las personas con discapacidades” [18].

Las personas a las que están destinadas este tipo de tecnologías son pacientes con alguna discapacidad, personas mayores o personas con trastornos de salud mental, como demencia o autismo. El uso de tecnologías de asistencia permite a sus usuarios realizar determinadas actividades de forma autónoma lo que les proporciona una mayor independencia[19].

Las tecnologías de asistencia se pueden clasificar según su nivel tecnológico en los siguientes grupos [20]:

- De no tecnología: en este grupo se engloban los dispositivos de uso común a los que se les ha dado una funcionalidad determinada, como pueden ser bastones o gafas.
- De baja tecnología: se corresponden a utensilios que han sufrido adaptaciones, aunque no demasiado complejas, es el caso de cubiertos o materiales escolares adaptados.
- De media tecnología: se incluyen los productos con una complejidad tecnológica mayor a los anteriores, como por ejemplo una silla de ruedas con funcionamiento manual.
- De alta tecnología: la complejidad tecnológica de estos productos es elevada, y además, están fundamentados en técnicas como la robótica, la ingeniería biomédica o las tecnologías de la información. Algunos ejemplos de estos productos son prótesis o productos de teleasistencia.

Este tipo de técnicas están presentes en la actualidad en multitud de áreas, se presentan a continuación algunas de ellas [21]:

- **Comunicación:** utilizadas por personas con trastornos de comunicación, tales como personas con dificultades en el habla, personas con parálisis cerebral o autismo entre otros. Un ejemplo es el dispositivo generador de voz, que puede poseer distintos métodos de entrada según las capacidades de los usuarios, y producen una salida de voz. Otro ejemplo, son los dispositivos electrónicos de fluidez, que ayudan a mejorar la fluidez a personas tartamudas.
- **Movilidad:** destinados a personas con limitaciones en su movilidad, a causa de una avanzada edad, enfermedades o accidentes. Estos dispositivos permiten a estas personas realizar desplazamientos. Entre ellos podemos encontrar sillas de ruedas.
- **Actividades de la vida diaria:** permiten que sus usuarios puedan realizar tareas cotidianas de una manera independiente. Algunas de estas tecnologías ayudan en tareas como comer, cocinar, limpiar, vestirse o realizar la higiene diaria.

3.1.2. Robótica asistencial

En muchas de las áreas anteriormente explicadas se utilizan dispositivos de alta tecnología como son los robots para realizar esas tareas de asistencia. Los robots de asistencia realizan tareas las cuales el usuario no puede realizar por él mismo. Entre ellos nos encontramos robots de guiado, sillas de ruedas autónomas, prótesis, exoesqueletos y brazos robóticos de asistencia [22].

Los robots de guiado tienen la función de guiar al usuario por rutas que han sido predefinidas y que estos robots realizan de manera autónoma. Son robots con apariencia semihumanoide, un ejemplo de este tipo de robot es el Care-o-Bot, mostrado en la Figura 3.1.



Figura 3.1: Robot Care-o-Bot. *Fuente:[1]*

Las sillas de ruedas autónomas son otros dispositivos robóticos de asistencia, entre las características que poseen se encuentra la detección de obstáculos o el paro inteligente. Un

ejemplo es la silla de ruedas diseñada por Panasonic, puede observarse su apariencia en la Figura 3.2, esta es capaz de moverse de forma automática a un destino fijado escogiendo la mejor ruta, posee sensores capaces de detectar obstáculos y función de parada automática por si ello fuera necesario.



Figura 3.2: Silla de ruedas Panasonic. *Fuente: [2]*

Las prótesis son dispositivos capaces de realizar una funcionalidad determinada sustituyendo la de una parte del cuerpo, debido a que por la existencia de alguna discapacidad el usuario no es capaz de realizarla por sí mismo. La prótesis de miembro superior Michelangelo de Ottobock es una de las más extendidas, esta mano robótica mostrada en la Figura 3.3 es controlada mediante control mioeléctrico, es decir, mediante señales generadas debidas a la actividad de los músculos y que son medidas en la superficie de la piel.



Figura 3.3: Prótesis robótica de mano Michelangelo. *Fuente: [3]*

Los exoesqueletos son estructuras que se colocan sobre el cuerpo humano, de forma similar a una prenda de ropa, y cuya función es la de asistir movimientos o aumentar las capacidades del cuerpo humano [23]. En la Figura 3.4 podemos ver el modelo MyoPro, un exoesqueleto de miembro superior, este asiste en los movimientos del codo y de los dedos, su monitorización

se realiza del mismo modo que la prótesis Michelangelo, con control mioeléctrico.



Figura 3.4: Exoesqueleto de miembro superior MyoPro. *Fuente:* [4]

Los brazos robóticos de asistencia son robots que poseen, como mínimo, tres articulaciones. Debido a que trabajan próximos a las personas estos deben tener un control de colisiones, poca fuerza en el efector final y deben ser colaborativos. Un modelo que se encuentra presente en el mercado en la actualidad es el brazo asistencial Kinova Jaco. Este robot permite a sus usuarios realizar actividades de la vida diaria como es comer, realizar su higiene personal e incluso realizar actividades de ocio, como por ejemplo pintar. El control se realiza gracias a un joystick, cuyos movimientos se transforman en los del brazo robótico, mostrado en la Figura 3.5.



Figura 3.5: Robot Kinova Jaco. *Fuente:* [5]

Además, existen otros modelos de robots colaborativos que pueden usarse para este fin, es decir, para asistir a personas con discapacidades, es el caso del robot UR3, al ser un robot colaborativo está diseñado para trabajar cerca de personas, por lo que se ajusta a las condiciones de seguridad y a los requisitos para funcionar como un brazo robótico asistencial. Se muestra este robot en la Figura 3.6.

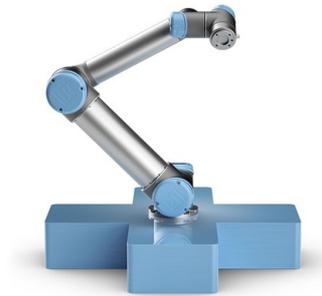


Figura 3.6: Robot UR3. Fuente: [6]

3.2. Realidad virtual

3.2.1. Definición y componentes

La Realidad Virtual (RV) se define como la simulación generada mediante computador de una escena 3D, y que permite la interacción con esta en tiempo real, de manera que se permita al usuario sentirse inmerso en el espacio. Los tres rasgos fundamentales que caracterizan un sistema de RV son los siguientes [24]:

- Inmersión: se refiere a la desconexión del usuario con el mundo real, de manera que la información que capte a través de sus sentidos provenga únicamente del mundo virtual.
- Percepción de presencia: del usuario en el entorno virtual, de este modo se siente presente en la simulación.
- Interacción: referida a la capacidad de comunicación entre el usuario y el entorno virtual.

Los elementos que forman un sistema de RV son: los periféricos de entrada y salida y el computador, que hace uso de modelos geométricos 3D, del software de tratamiento de datos, del software de simulación física y del software de simulación sensorial [25].



Figura 3.7: Elementos de un sistema de realidad virtual

- Periféricos de entrada: su función es obtener las acciones del usuario y enviarlas a la computadora. Entre los periféricos de entrada más utilizados se encuentran posicionadores (Figura 3.8b), guantes (Figura 3.8a), joysticks y micrófonos.



(a) Guantes de realidad virtual.

(b) Cámara Kinect.

Figura 3.8: Dispositivos de entrada de sistemas de realidad virtual *Fuentes: [7] [8]*

- Periféricos de salida: son, junto con los periféricos de entrada, el enlace entre el usuario y el computador, se encargan de enviar los estímulos auditivos y visuales, entre otros, al usuario. Ejemplos de periféricos de salida son cascos estereoscópicos (Figura 3.9a), pantallas de proyección, altavoces o dispositivos hápticos (Figura 3.9b).



(a) Gafas de realidad virtual.

(b) Guantes con dispositivo háptico.

Figura 3.9: Dispositivos de salida de sistemas de realidad virtual. *Fuentes: [9] [10]*

- Computador: su objetivo es realizar la simulación de forma interactiva, gracias al modelo geométrico 3D, el software de tratamiento de datos de entrada, de simulación física y de simulación sensorial.
- Modelo geométrico 3D: gracias al cuál pueden generarse distintas imágenes dependiendo del punto de vista del usuario.
- Software de tratamiento de datos de entrada: su función es realizar la lectura y el procesamiento de las señales de entrada,. Algunos ejemplos son los sistemas de reconocimiento de voz, o sistemas de transformación de movimientos de la cabeza a un sistema de coordenadas.
- Software de simulación física: se corresponde con todas las actuaciones relacionadas con modificaciones de la representación digital de la escena, ya sea por acciones del usuario o por la evolución interna del sistema, como pueden ser colisiones o deformaciones.

- Software de simulación sensorial: su función es la de obtener la representación digital de lo que serán los estímulos auditivos, táctiles y visuales, siendo estos últimos los que cobran mayor importancia.

3.2.2. Historia de la Realidad Virtual

El primer intento de inmersión en un entorno no real se remonta al siglo XIX, con las pinturas panorámicas, realizadas con el objetivo de dotar al observador de la sensación de estar presente en escenas históricas, como por ejemplo batallas, esto se conseguía llenando el campo de visión del observador. Posteriormente, gracias al descubrimiento de Charles Wheatstone en 1838, de que el cerebro obtiene las imágenes en dos dimensiones de cada ojo y las transforma en una escena tridimensional, se desarrolló el estereoscópico, que dotaba al usuario de una sensación de profundidad e inmersión mediante la visualización de dos imágenes [26].



Figura 3.10: Gafas estereoscópicas. *Fuente:* [11]

En el siglo XX, se sucedieron varios dispositivos de inmersión, como es el primer simulador de vuelo, creado en 1929 por Edward Link, este fue utilizado durante la Segunda Guerra Mundial para el entrenamiento inicial y la mejora de las habilidades de los pilotos del ejército de los EE.UU. Más tarde, en la década de 1950 se desarrolló el Sensorama gracias a Morton Heiling, este permitía que el usuario se sintiera inmerso en un cortometraje mediante estímulos auditivos, olfativos y hápticos.

Otro de los dispositivos desarrollados por Morton Heiling fue el Telesphere Mask en 1960, este fue el primer caso de pantalla montada en la cabeza o Head Mounted Display (HMD), aunque está no disponía de seguimiento de movimientos de la cabeza. Esto no fue hasta 1961, cuando Comeau y Brian, desarrollaron el Headsight, una pantalla montada en la cabeza con seguimiento de movimiento, fue creada para el uso de militares, su utilización consistía en el movimiento de una cámara remota mediante los movimientos de la cabeza, esto permitía observar escenarios peligrosos a distancia y de manera inmersiva.

Otro proyecto en el que se incluyó gráficos interactivos, además de estímulos de sonido,

olor y tacto, fue el de Ivan Sutherland en 1965 y fue llamado "The Ultimate Display", y transformaba los movimientos de la cabeza en cambios del punto de vista en las imágenes. Este proyecto fue financiado por Harvard más tarde y recibió el nombre de "Sword of Damocles", estaba formado por dos tubos de rayos catódicos que mostraban imágenes gráficas que se reflejaban en una serie de espejos. Este dispositivo colgaba del techo, debido a su gran peso, y permitía saber en todo momento la posición y orientación de la cabeza del usuario, este es considerado como el primer caso de visor de realidad virtual [27].



Figura 3.11: Shuterland con el dispositivo "Sword of Damocles". *Fuente:* [12]

Durante finales del siglo XX también se desarrollaron otros sistemas de realidad virtual como son el proyecto VIEW de la NASA, Sega VR-1 o Nintendo Virtual Boy. A principio del siglo XXI se desarrolló Street View, imágenes de 360° incluidas en el servicio Maps de Google, otro gran lanzamiento fueron las gafas de RV Oculus Rift.

En la actualidad, existen numerosas opciones disponibles de elementos de realidad virtual, por lo que su uso se está expandiendo cada vez a más sectores, de modo que en el presente esta tecnología es un recurso frecuente en diversos campos como son la medicina, el entretenimiento, la conducción, la educación o el entrenamiento.

3.2.3. Clasificación de los Sistemas de Realidad Virtual

La inmersión en RV, se refiere a la sensación del usuario de sentirse presente en la escena no real, esto puede conseguirse mediante imágenes, sonidos o texturas que imiten la realidad. Existe una clasificación de sistemas de RV dependiendo del grado de inmersión, son los siguientes [28]:

- Sistemas no inmersivos: Se refiere a tecnologías en las que se usa un monitor con escenas

del mundo virtual como dispositivo de salida, por lo que durante su uso el usuario es pleno conocedor de su entorno real. Los elementos por los que está compuesto son la pantalla y dispositivos de entrada tales como un teclado, un ratón, un micrófono o un joystick.

Esta tecnología es ampliamente utilizada en visualizaciones científicas o como herramienta para entrenamiento. Algunas de las ventajas que proporciona son el coste, ya que es más barato, y además provee de menores tiempo de adaptación a los usuarios.

- **Sistemas semi-inmersivos:** En este tipo de RV, el dispositivo de salida son cuatro pantallas dispuestas en forma de cubo, de manera que tres de ellas formarían las paredes y la restante el suelo. Los dispositivos de entrada en este caso son unas gafas y un dispositivo de seguimiento de movimientos de la cabeza.

Este tipo de sistemas es utilizado en determinados casos en los que se desea que el usuario sea capaz de obtener estímulos del mundo real.

- **Sistemas inmersivos:** Este tipo de sistemas proporcionan una experiencia en la que el usuario se siente inmerso en el mundo virtual. Algunos de los elementos de entrada que suelen ser utilizados son guantes o joysticks. Además, para la visualización suele hacerse uso de gafas RV o pantallas HMD, montadas en la cabeza, estos elementos crean un efecto 3D estereoscópico, que unido al seguimiento de los dispositivos de entrada ofrece una experiencia inmersiva.

Los sistemas inmersivos son utilizados comúnmente para aplicaciones de entretenimiento, tales como videojuegos. Además, es utilizado para actividades de entrenamiento, y en otros sectores como la educación.

3.3. Realidad Virtual como tecnología de asistencia

Los entornos de RV son utilizados en muchos casos como medio de adaptación a sistemas de tecnología asistenciales, además gracias al uso de estas simulaciones es posible acondicionar el sistema al usuario, y saber cuáles son sus capacidades.

Estos entornos proporcionan al usuario una retroalimentación visual mediante pantallas o gafas virtuales. También pueden utilizar otro tipo de retroalimentación tales como elementos hápticos, sonidos u olores. De esta manera el usuario interactúa con el entorno a través de mecanismos, anteriormente nombrados, como son joysticks y teclados entre otros [29].

Algunas de las ventajas que proporciona el uso de realidad virtual son la realización de tareas con objetivos concretos y la posibilidad de realizar múltiples repeticiones, ambos son factores importantes en la rehabilitación neurológica.

Otra de las ventajas es que el uso de esta permite probar tareas que no son seguras en el mundo real. Además, muchos de estos programas están diseñados para su uso sin la super-

visión de un especialista. Por otra parte, el uso de realidad virtual supone un método más interesante y agradable para sus usuarios.

Se van a detallar algunas de las investigaciones que se han realizado en torno a estudios que utilizan la realidad virtual como herramienta para un fin de asistencia haciendo uso de diferentes sistemas de interacción hombre-máquina.

En primer lugar, se va a presentar un estudio en el que se creó un entorno virtual como tecnología asistencial a deportistas con movilidad reducida [13]. El simulador creado estaba compuesto por una computadora, una interfaz con la silla de ruedas, sistemas de captura corporal óptico y guantes de datos y dispositivos de visualización. Por ello el sistema de interacción hombre-máquina es la plataforma mostrada en la Figura 3.12. En este caso, la aplicación está destinada a jugadores de baloncesto en silla de ruedas con el objetivo de mejorar su rendimiento deportivo.



Figura 3.12: Plataforma de movimiento, sistema háptico y silla de una interfaz. *Fuente:* [13]

Otro estudio a mencionar en el que se ha utilizado de manera conjunta el control mioeléctrico y realidad virtual ha sido el realizado para entrenamiento protésico mioeléctrico [14]. Una demostración del uso de este sistema puede observarse en la Figura 3.13. Este estudio ha demostrado que este entrenamiento de forma inmersiva produce mejoras en el desempeño de tareas y habilidades funcionales del usuario. Además, concluyó con que un entrenamiento de este tipo reduce las tasas de rechazo, y proporciona al paciente un método de adaptación para la prótesis.

Otro estudio realizado, aunque en este no se ha hecho uso de la realidad virtual, es el llevado a cabo para personas con tetraplejia, en el que se consiguió controlar un brazo robótico asistencial mediante el uso de un sistema de control intraoral inalámbrico el cuál se domina mediante la lengua [15]. En este estudio se consiguió obtener un control sobre 14 grados de libertad de movimiento con el robot Kinova Jaco el cual dispone de una pinza mediante la cual pueden agarrarse los objetos. Para ello se creó una interfaz a través de la cual el usuario pudiese obtener información para facilitarle el control.

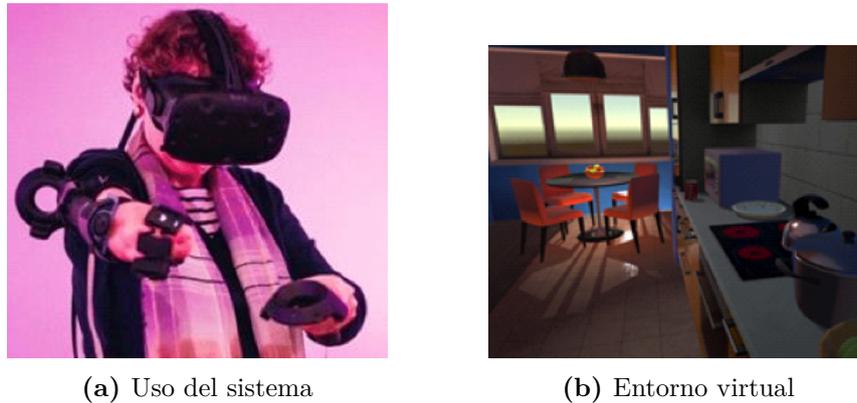


Figura 3.13: Uso de dispositivo de realidad virtual en conjunto con el control mioeléctrico. *Fuente:* [14]

En la Figura 3.14 pueden observarse los elementos que intervienen en el citado estudio, estos son los siguientes:

- a) El sistema de interacción con el sistema, este incorpora 18 sensores que se activan mediante el dispositivo de metal que se encontrará en la lengua del paciente.
- b) Unidad inalámbrica que recibe la señal del sistema anterior y lo transforma en caracteres.
- c) Este ordenador recibe los caracteres de la unidad anterior y los convierte en comandos, los cuales se transmiten al brazo robótico.
- d) La pantalla del ordenador que muestra la retroalimentación al paciente en cuanto a que sensor está activando, en que se traduce en cuanto a los movimientos del robot y el software del robot.

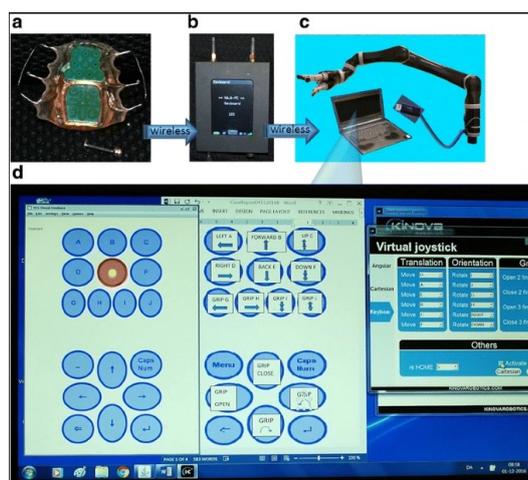


Figura 3.14: Elementos del sistema de control robótico controlado por la lengua *Fuente:* [15]

4. Planificación del proyecto

Previamente al comienzo de este proyecto conjunto, formado por los trabajos mencionados en el apartado 1, se sentaron las bases del proyecto. Esto es, un primer boceto de cómo se iba a llevar a cabo el trabajo para cumplir los objetivos que se habían propuesto. Una vez concretado este desarrollo se observó que, la realización del trabajo requería del uso de diferentes software, además de que estas actividades iban a ser realizadas por dos personas de manera conjunta. Por todo ello, se decidió realizar un análisis de los factores que intervenían, además de una planificación en detalle de todas las actividades a realizar.

Para el análisis de los factores a intervenir se estudiaron las posibles debilidades, amenazas, fortalezas y oportunidades con el fin de obtener una visión general de los agentes que intervienen en el trabajo y que dependen, o no de las dos componentes que han realizado este proyecto. Este análisis puede observarse en el DAFO desarrollado y mostrado en la Figura 4.1.



Figura 4.1: Análisis DAFO

Con el fin de establecer soluciones para afrontar cada uno de los aspectos presentados en el análisis DAFO, se establecieron las acciones para corregir, afrontar, mantener y explotar los aspectos anteriormente mencionados. Este análisis está recogido en el CAME mostrado en la Figura 4.2.

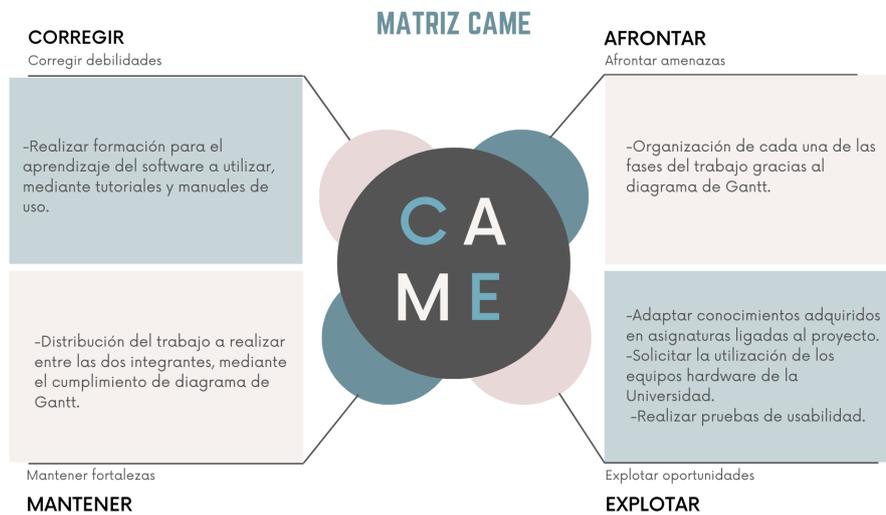


Figura 4.2: Análisis CAME

Además, destacar que el proyecto se ha desglosado en varias partes, las cuáles se planificaron antes del comienzo del trabajo con el fin de distribuir en el tiempo disponible cada una de las actividades a cumplir para conseguir los objetivos propuestos y mostrados en el apartado 2. Esta distribución de actividades puede observarse en el Diagrama de Gantt mostrado en el Anexo A.

5. Metodología y materiales empleados

El propósito de este apartado es mostrar la metodología que se ha llevado a cabo en la elaboración del proyecto, y definir el funcionamiento de cada uno de los materiales que se han empleado para ello.

5.1. Metodología

La realización de este trabajo ha constado de varias fases que se han seguido durante su elaboración, estas se han basado en el Gantt realizado de manera previa al inicio del trabajo y mostrado en el apartado 4 de este TFM. Destacar, que estas fases se han hecho de manera conjunta entre las dos integrantes del grupo. A continuación, se explican brevemente cada una de estas etapas.

- En primer lugar, se llevó a cabo un estudio previo del software para llegar a obtener los conocimientos necesarios para la realización del trabajo.
- La siguiente fase consistió en la selección del robot a emplear para las actividades de asistencia en el entorno virtual, y adaptación del mismo a las necesidades requeridas, esto es únicamente tres grados de libertad.
- Después se creó el entorno virtual, este ha sido ambientado en una cocina. En esta fase se desarrollaron los elementos de la estancia, los objetos a manipular, la caja para la realización de las pruebas y todos los elementos que conforman el entorno virtual.
- A continuación, se programaron cada uno de los scripts necesarios para realizar los movimientos lineales del robot en los ejes x , y y z , y el movimiento de la pinza.
- Una vez obtenido el entorno virtual y los movimientos del robot se realizó la personalización de cada uno de los modos, ya que cada uno de ellos requería de un funcionamiento diferente. Además, se realizaron pruebas junto a Matlab, para cerciorar que el socket funcionaba correctamente.
- Por último, se realizaron pruebas con voluntarios para evaluar el funcionamiento del sistema en conjunto con el control mioeléctrico. Posteriormente, se realizó el análisis de estos datos.

5.2. Materiales empleados

Para el desarrollo del entorno virtual se han utilizado varios recursos como son un ordenador y el software de Unity. Además, se ha hecho uso de otros medios como son, el software

Noraxon MR3 junto a su equipo hardware, y el software Matlab, estos serán utilizados en conjunto para definir la información que llegue a Unity y que será la que controle los movimientos del robot. Estos dos últimos recursos se explicarán más en profundidad en el trabajo en colaboración a este [17]. En este apartado se describirá brevemente el funcionamiento de cada uno de ellos, para entender su papel en la realización del trabajo. Todos estos se muestran en la Figura 5.1.

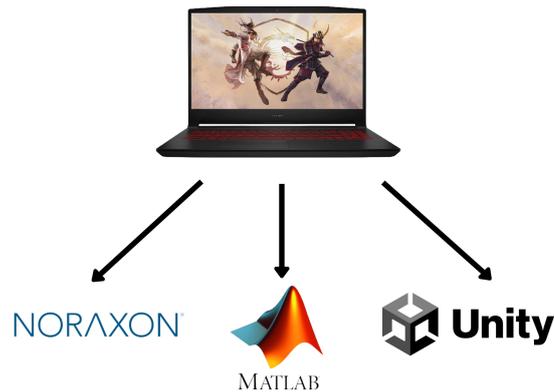


Figura 5.1: Recursos utilizados en la realización del trabajo

5.2.1. Ordenador

La ejecución de este proyecto requiere de un ordenador con unas características determinadas que soporte el software que se va a utilizar. Por ello el ordenador escogido para las pruebas ha sido el MSI Katana GF66 11UD-229ES. Entre las características que posee están:

- Procesador Intel Core i7-11800H
- Memoria RAM: 16GB
- Tarjeta gráfica: GeForce RTX 3050 Ti

5.2.2. Noraxon

El software Noraxon Myo Research 3 (MR3) es utilizado para la codificación, sincronización y presentación de datos, que han sido obtenidos desde dispositivos hardware compatibles, como es el caso del Noraxon Mini Direct Transmission System (MiniDTS), utilizado en este proyecto. Estos dispositivos hardware miden datos correspondientes a las señales electromiográficas superficiales [30].

De este modo, desde Noraxon MR3 se pueden visualizar, en tiempo real, las señales de Electromiografía (EMG) de un músculo determinado del individuo. Estas señales se verán modificadas según la contracción de los músculos del usuario. En este proyecto se hará uso de las señales EMG para controlar los movimientos del robot del entorno virtual.

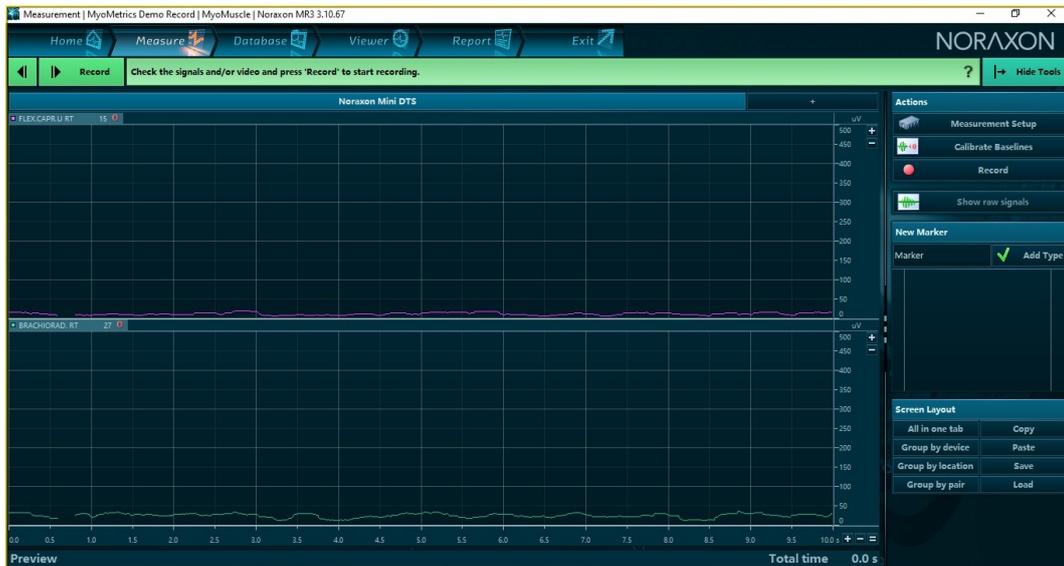


Figura 5.2: Interfaz de Noraxon MR3

5.2.3. Matlab

El software Matlab es un entorno de programación que fue desarrollado por MathWorks. Entre las múltiples prestaciones de Matlab podemos encontrar el análisis de datos, la realización de gráficas, la creación de aplicaciones, el uso con otros lenguajes como Python o Java, la conexión con determinados hardware, el cálculo mediante equipos multinúcleo o el cálculo en la nube [31]. La interfaz de Matlab se muestra en la Figura 5.3.

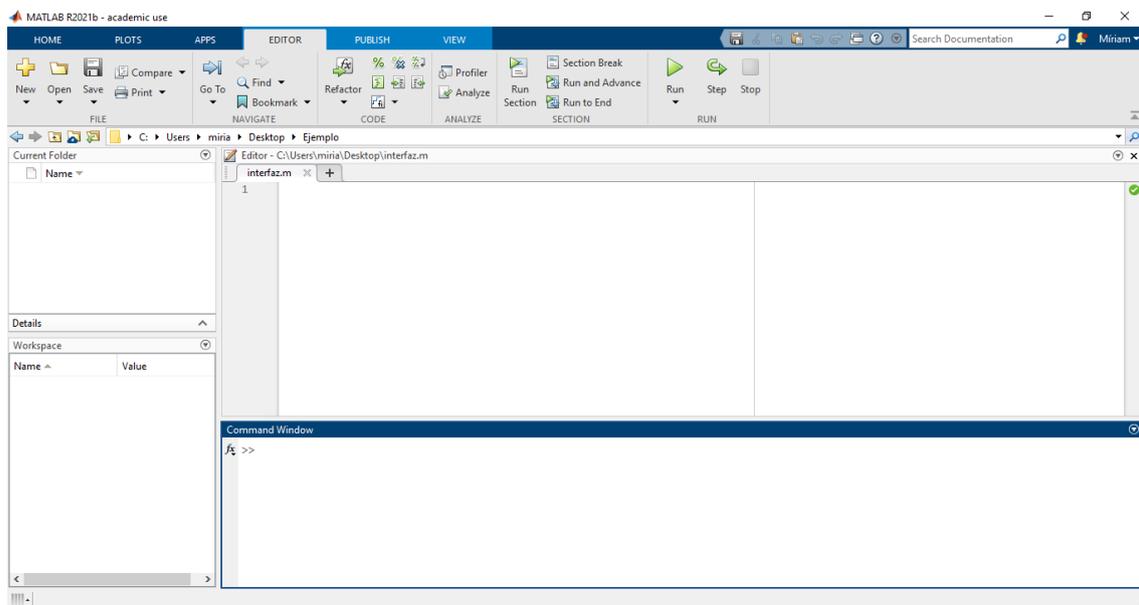


Figura 5.3: Interfaz de Matlab

La utilización de esta plataforma en este trabajo, permite, realizando una lectura de los datos suministrados por Noraxon MR3, tomar unas acciones u otras, que enviará a Unity para realizar los movimientos del robot. Además, ofrece una interfaz visual, para conocer información relevante en el proceso.

5.2.4. Unity

5.2.4.1. Características generales

El software Unity es un motor de videojuegos, es decir, está conformado por una serie de rutinas de programación que posibilitan el diseño, la creación y el funcionamiento de un videojuego. Fue lanzado por Unity Technologies en 2005, y está disponible como plataforma de desarrollo para Windows, Mac OS y Linux. Además, tiene soporte de compilación en otras plataformas como son web (WebGL), dispositivos móviles (iOS, Android, Windows Phone, Tizen), televisiones inteligentes (tvOS, Samsung Smart Tv, Android TV), consolas (PlayStation 4, Xbox 360, Wii U, Nintendo Switch) y dispositivos de realidad virtual (Oculus Rift, PlayStation VR, Samsung Gear VR).

En esta plataforma se pueden importar objetos creados en Blender, Autodesk 3ds Max, Autodesk Maya, Softimage y Adobe Photoshop entre otros. El motor gráfico utiliza OpenGL, Direct3D, OpenGL ES o interfaces propietarias, dependiendo de la plataforma en la que se compile. El motor gráfico integra el sistema de cálculo de físicas desarrollado por Nvidia PhysX, mediante el cual el juego cobra de un sentido más realista gracias a las físicas que incorpora.

El lenguaje de programación con el que se escriben los scripts es C#. Estos scripts manejan el comportamiento de los GameObjects, de esta manera los scripts en conjunto con los componentes pertenecientes a los GameObjects y su manera de interactuar es lo que conforman el videojuego. Un GameObject es en Unity la clase base para todas las entidades en las escenas de Unity.

5.2.4.2. Interfaz de Unity

La interfaz de Unity está compuesta por varias ventanas, estas pueden ser modificadas por el usuario, pero las más comunes se muestran en la Figura 5.4, y se detallan a continuación:

- Project Window o ventana de proyecto: es el lugar dónde aparecen los assets, importados o creados, presentes en el proyecto, tanto modelos 3D, como scripts, texturas e incluso archivos de audio.
 - Scene View: permite navegar, de manera visual, por la escena y editar el tamaño, la posición y la orientación de los objetos presentes en ella.
 - Hierarchy Window: se incluyen los objetos de la escena con el nombre que se les ha asignado, estos aparecen representados de manera jerárquica, indicando la estructura
-

de los objetos que se encuentran agrupados dentro de otros. Unity utiliza jerarquías padre-hijo, de modo que el hijo hereda las propiedades del padre.

- Inspector Window: mediante la cual se pueden visualizar y editar las propiedades de los objetos.
- Toolbar: en ella se presentan las herramientas básicas para trabajar en Unity.

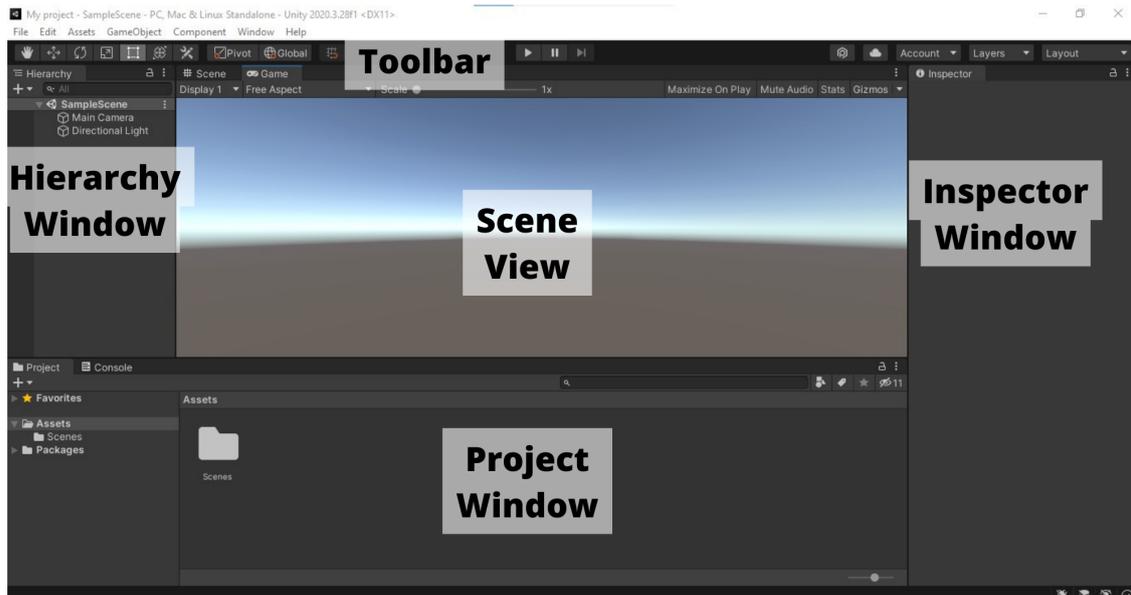


Figura 5.4: Interfaz de Unity

5.2.4.3. Elementos de Unity

Los juegos se crean en cada una de las escenas, estas contienen los personajes, las decoraciones y todo lo relacionado con el entorno. Una escena por defecto incluye una luz y una cámara, dependiendo de la posición en la que se coloque la cámara el juego se verá desde un lugar u otro.

Dentro de una escena tendremos GameObjects, estos son el elemento principal del editor de Unity. Todos los objetos del juego son GameObjects, ya sean personajes, elementos, luces o cámaras. Pero este necesita de propiedades para dejar de ser un objeto inanimado, estas propiedades se consiguen agregando componentes al GameObject. El componente principal de un GameObject es el Transform, que incluye las propiedades de posición, rotación y escala de un objeto.

Los GameObjects y algunos de sus componentes pueden ser creados directamente desde la escena o arrastrando a la escena un asset desde la ventana de proyecto. Estos assets a su vez pueden ser creados por el usuario desde otras aplicaciones o ser obtenidos desde el Asset

Store, este es un recurso de Unity mediante el cual pueden obtenerse modelos 3D, texturas, tutoriales y extensiones entre otros. Algunos tipos comunes de assets son los siguientes:

- Archivos de imagen: pueden ser importados como BMP, TIF, TGA, JPG y PSD.
- Archivos de modelos 3D: conformarán los personajes y los entornos de los juegos.
- Mallas y animaciones: correspondientes a los archivos de modelos 3D, las mallas y animaciones se importarán junto a ellos.
- Archivos de audio: algunos de los formatos que soporta son .mp3, .ogg o .wav.

5.2.4.4. Físicas en Unity

En Unity es posible representar físicas con el objetivo de simular las colisiones, la gravedad y otras fuerzas entre los objetos presentes en la escena. Estas físicas se consiguen gracias a determinados componentes que se asignan a los GameObjects. A continuación, se van a detallar los componentes principales de los que se hacen uso para la creación de estas físicas:

- Character Controller: este componente dota al personaje de la capacidad de mantenerse erguido, sin caerse al suelo, además de no atravesar paredes u otros objetos. Este no necesita de un Rigidbody para realizar las colisiones. Sin embargo, los efectos de impulso no son realistas, y este no se verá acelerado por colisiones. Pueden modificarse algunos parámetros de este componente como son el límite de pendiente, la compensación de peso y el ancho de piel entre otros, según se muestra en la Figura 5.5.

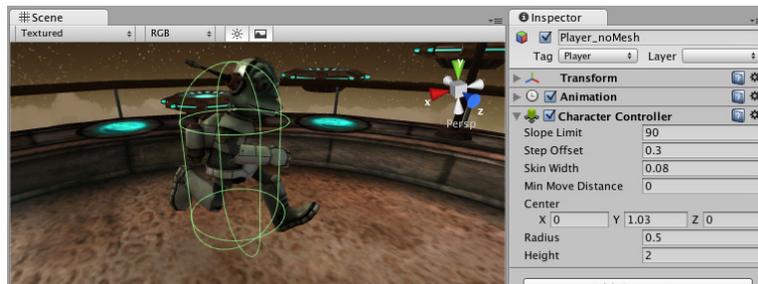


Figura 5.5: Componente Character Controller en Unity. *Fuente:* [16]

- Rigidbody: este es el componente principal que permite a los GameObjects el comportamiento basado en la física. Basta con añadir este componente a un objeto para que responda a la gravedad. Una vez hemos añadido este componente no se debe intentar realizar movimientos del objeto cambiando su componente Transform mediante un script, en su lugar deben de aplicarse fuerzas al objeto y dejar que el motor de físicas calcule los resultados. Por otra parte, este componente puede modificarse según su masa, su arrastre o rozamiento con el aire y deshabilitar la gravedad entre otros, según se muestra en la Figura 5.6.

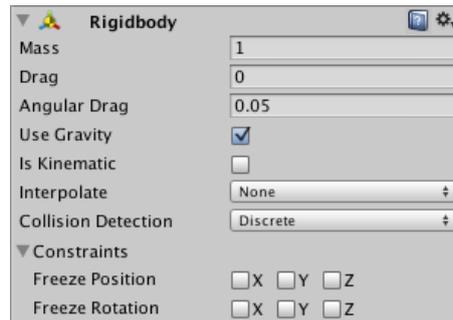


Figura 5.6: Componente Rigidbody en Unity. Fuente:[16]

- Colliders: se necesitan para configurar de qué manera colisionan los objetos, mediante la definición de la forma física del objeto. Existen colliders más simples con forma de cubo (Box Collider) mostrado en la Figura 5.7, esfera (Sphere Collider) o cápsula (Capsule Collider), o más complejos como las mallas (Mesh Colliders) que se ajustan de manera exacta a la forma del objeto. Según si se ha añadido o no un Rigidbody, hablaremos de colisionadores dinámicos o estáticos, estos últimos referidos a objetos inmóviles como paredes o suelos, al contrario que los dinámicos que se mueven en respuesta a otras colisiones.

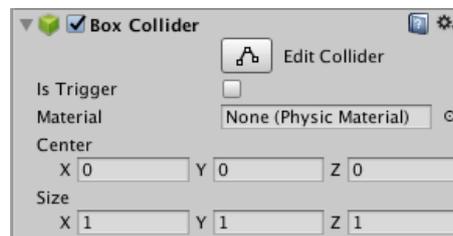


Figura 5.7: Componente Box Collider en Unity. Fuente:[16]

- Joints: mediante los cuales se conectan un Rigidbody a otro en un punto fijo cualquiera. Gracias a estas articulaciones se aplican fuerzas que mueven los Rigidbody, y definiendo límites en estas articulaciones puede restringirse el movimiento. Existen varios tipos de componentes relativos a las articulaciones como el Character Joint mostrado en la Figura 5.8, o el Articulation Body detallado a continuación.

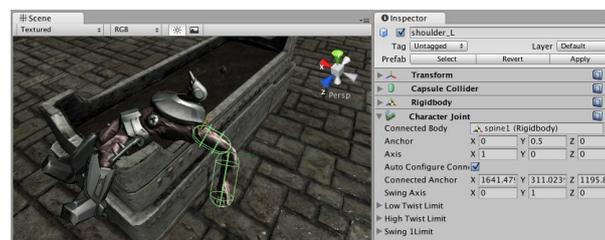


Figura 5.8: Componente Character Joint en Unity. Fuente:[16]

- **Articulation Body:** es un componente que permite agrupar las articulaciones según un árbol lógico, de manera que un movimiento de un objeto superior en la jerarquía refleje un movimiento de los objetos inferiores según las restricciones impuestas. Permiten la simulación de brazos robóticos y cadenas cinemáticas. Por ello, será el componente principal en la creación de los movimientos del robot en nuestro entorno virtual. En las Figuras 5.9 y 5.10 se muestran la jerarquía de un Articulation Body y los parámetros que intervienen en este componente

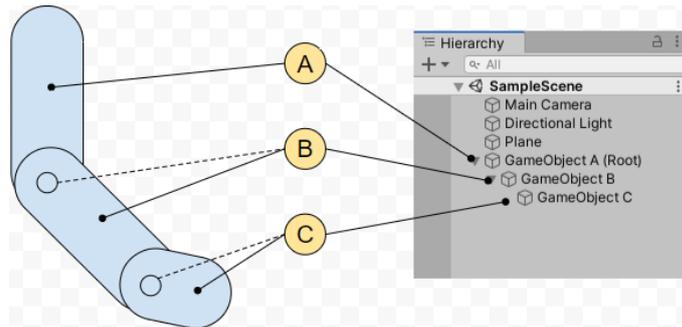


Figura 5.9: Jerarquía de un Articulation Body en Unity. *Fuente:[16]*

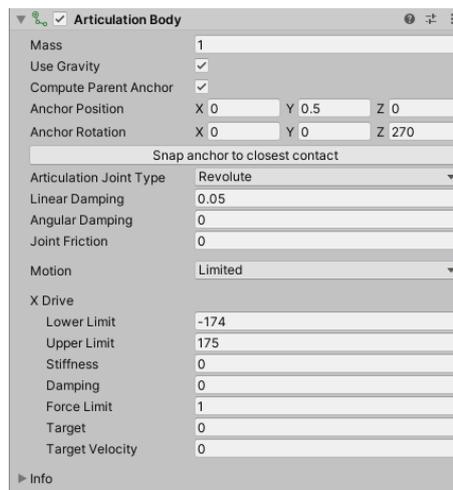


Figura 5.10: Componente Articulation Body en Unity. *Fuente:[16]*

Los tipos de articulación que pueden seleccionarse son los siguientes:

- **Fixed:** fija los dos objetos de manera que no puedan realizarse movimientos entre ellos.
- **Prismatic:** esto significa que posee un único grado de movilidad, por lo que el movimiento sólo puede realizarse de manera rectilínea a lo largo de un eje.
- **Revolute:** con este tipo de articulación se establece un movimiento de revolución en torno a un eje.

– Spherical: este tipo de articulación permite la revolución en tornos a los tres ejes. Los tipos de fuerzas que intervienen y cuyos parámetros hay que fijar en la articulación son:

- Stiffness: la rigidez con la que la articulación alcanza el punto al que se dirige.
 - Damping: la amortiguación con la que la articulación alcanza el punto al que se dirige.
 - Force Limit: la cantidad máxima de fuerza o torsión que puede producir.
-

6. Desarrollo del software

Tal y como se ha explicado en el apartado 5 se van a utilizar tres software diferentes para la realización del proyecto. Cada uno de ellos tiene una función diferente en el proceso, cada una de estas funciones se observan en el esquema de la Figura 6.1. En este apartado se explicarán las distintas partes desarrolladas en la realización del proyecto. Respecto a la lectura y procesamiento de señales se hará una breve explicación, ya que en el proyecto colaborativo a este se exponen más en detalle.

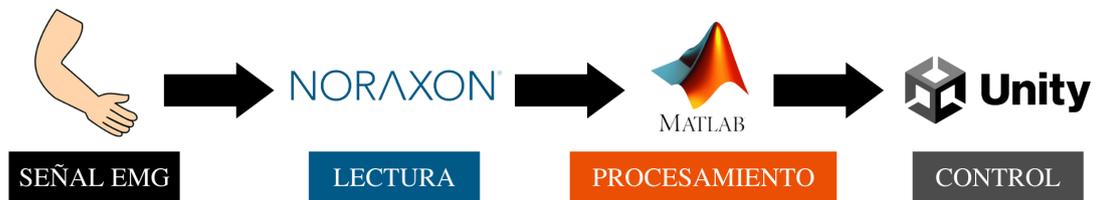


Figura 6.1: Diagrama general de la funcionalidad de cada uno de los elementos del proyecto

6.1. Lectura de la señal muscular

El software Noraxon MR3 proporciona las señales EMG del usuario, estos datos se corresponden con el grado de contracción muscular y son medidas en μV . Para medir las señales se han conectado dos electrodos en cada uno de los músculos a evaluar, esta señal muscular es amplificada y enviada al software de Noraxon MR3 gracias al dispositivo Noraxon MiniDTS. Los datos musculares son leídos de manera continua y enviadas a Matlab en tiempo real mediante un socket Hypertext Transfer Protocol (HTTP).

Las señales que se han medido han sido las relativas a dos músculos, el músculo flexor y extensor largo del pulgar, estos dos se encuentran en la parte interior y exterior del antebrazo respectivamente. Estas señales representarán el grado de contracción del usuario, por lo tanto, el sujeto podrá realizar tres posibles acciones:

- Flexión: mediante la contracción únicamente del músculo flexor largo del pulgar.
- Extensión: mediante la contracción únicamente del músculo extensor largo del pulgar.
- Cocontracción: mediante la contracción de los dos músculos anteriores de manera simultánea.

6.2. Procesamiento de la señal

Matlab obtiene los valores relativos a las señales musculares y dependiendo de estos datos enviará un valor u otro a Unity. Estas decisiones serán tomadas haciendo uso de una máquina de estados, la cual es necesaria para que con las tres posibles acciones del usuario se pueda tener control de todos y cada uno de los movimientos del robot. Dependiendo de cada uno de los modos esta programación será diferente.

Además, este programa cuenta con una interfaz que ofrece al usuario información tal como el estado actual en el que se encuentra respecto a la máquina de estados, los valores de tensión que está generando con cada músculo y el recorrido que ha seguido con el robot en el plano xz . La interfaz ofrecida para uno de los modos, en concreto el modo 1, se muestra en la Figura 6.2. Esta interfaz, al igual que el procesamiento de la señal, será diferente dependiendo del modo en el que nos encontremos.

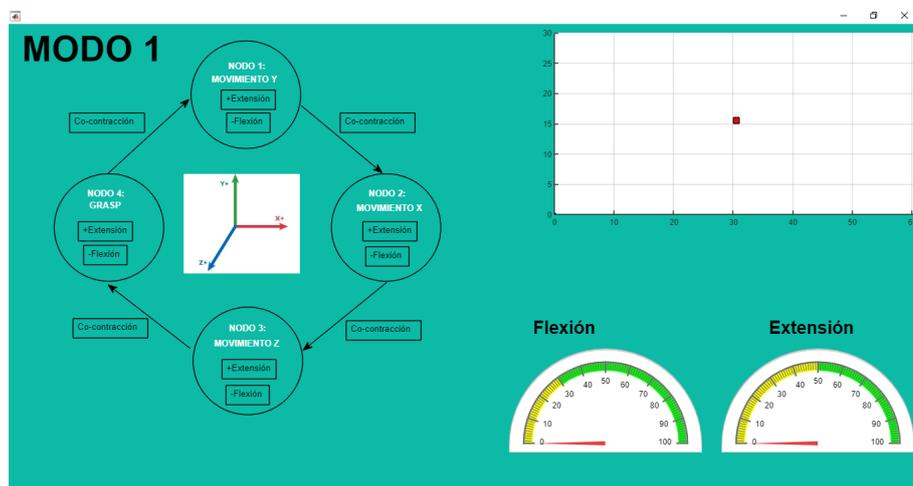


Figura 6.2: Interfaz de Matlab en el modo 1

6.2.1. Comunicación entre Matlab y Unity

Una vez procesada la señal y transformada en un mensaje determinado esta se envía a Unity, para ello se ha realizado una comunicación entre Matlab y Unity. Esta comunicación está basada en el Protocolo de Control de Transmisión (TCP), el cual es un protocolo de transporte superpuesto al Protocolo de Internet (IP), además es uno de los protocolos de redes más extendidos.

Este protocolo está basado en el modelo cliente-servidor. En dónde el cliente dirige las peticiones de servicio hacia el servidor seleccionado, mientras que el servidor debe mantenerse en todo momento en funcionamiento a la espera de las peticiones de servicio de los clientes en un puerto conocido. En este caso Matlab será el cliente y Unity el servidor. Destacar que el cliente debe ejecutarse antes que el servidor, por ello en este caso debemos ejecutar Unity primero.

Por otra parte, este protocolo hace uso de sockets para realizar la comunicación, estos son puntos de acceso para acceder a servicios de comunicación. La dirección de un socket permite definir una dirección única en internet. Debido a que la comunicación Matlab-Unity se realizará para la conexión entre estas dos plataformas en un mismo ordenador, la IP de destino y origen será la misma, correspondiente a la dirección de localhost 127.0.0.1. Mientras que el puerto será el 55002.

Mediante esta comunicación se permite la escritura desde Matlab de datos ASCII y binarios, que serán leídos en Unity. En este caso se enviarán mensajes con un único número, que Unity interpretará como un mensaje determinado para realizar cierta acción.

6.3. Entorno virtual del robot asistencial

El brazo robótico asistencial se ha desarrollado en el entorno de Unity, este se controlará según se ha explicado anteriormente. Se ha creado una escena en la que se encuentra inmerso el brazo robótico, esta escena intenta simular un entorno doméstico, en concreto el de una cocina, con el objetivo de hacer sentir al usuario en un ambiente habitual para él, y de esta manera aumentar su sensación de inmersión en el entorno virtual. La escena se muestra en la Figura 6.3.

En los siguientes apartados se explicarán los elementos que componen esta escena, así como los movimientos que es capaz de realizar el robot. Además, se explicarán en detalle cada uno de los modos que se han elaborado para la realización de las distintas pruebas.

6.3.1. Elementos de la escena

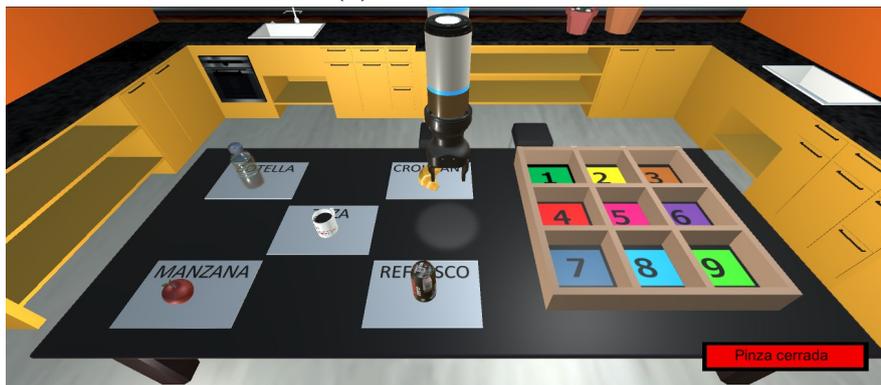
La escena está compuesta por diversos elementos, entre ellos podemos encontrar: los objetos a desplazar, la caja, algunos elementos decorativos y el robot, el cuál se definirá en el siguiente apartado. Todos los elementos, a excepción del robot y de la caja, han sido obtenidos del Asset Store.

Los objetos a desplazar se encuentran encima de una mesa, estos objetos son una botella, un croissant, una taza, una manzana y una lata de refresco, se han escogido estos objetos por ser elementos usuales en una cocina y familiares para el usuario. Los componentes asociados a estos GameObjects son un Rigidbody y un Collider, gracias a estos conseguimos que sean elementos a los que les afecte la gravedad, y que colisionen con otros objetos cuando estén en contacto. Gracias a estas físicas, es posible que el robot pueda agarrar estos objetos, ya que sin estas colisiones no sería posible.

En la Figura 6.4 pueden observarse los objetos que han de desplazarse, el prisma rectangular verde que rodea cada uno de los objetos se corresponde con un Box Collider, se ha seleccionado esta forma por ser más sencillo de realizar el agarre con el robot. El tamaño se



(a) Vista de escena



(b) Vista de juego

Figura 6.3: Entorno virtual creado en Unity

ha ajustado a cada uno de los objetos y se ha localizado en el centro del mismo.

Para que los objetos reaccionen a la fuerza de la gravedad y puedan ser desplazados se ha añadido el componente RigidBody con las características detalladas a continuación:

- Masa: 1 kilogramo
- Rozamiento con el aire: 0
- Rozamiento con el aire en giro: 0.05
- Uso de gravedad
- Detección de colisión: Discreta

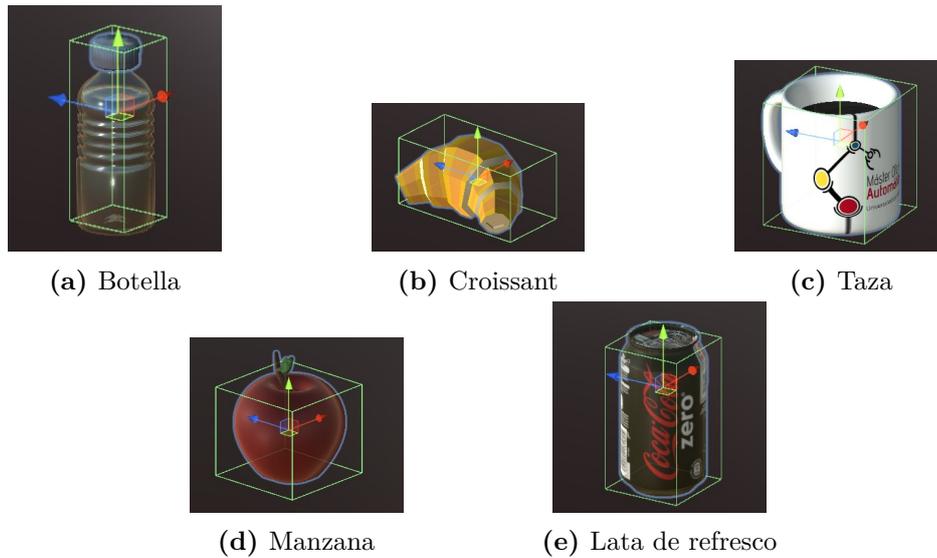


Figura 6.4: Objetos del entorno virtual

Por otra parte, la caja ha sido creada gracias a una aplicación de modelado 3D, llamada 3D Builder. Esta caja posee nueve posiciones diferentes, que han sido numeradas e identificadas con diferentes colores. Estas posiciones serán dónde han de colocarse los objetos durante las pruebas. Esta caja lleva asociada un elemento Collider para que se produzcan las colisiones necesarias con el robot. En la Figura 6.5 se observa la apariencia de esta caja.



Figura 6.5: Caja del entorno virtual

El resto de elementos decorativos, como las paredes, el suelo, la mesa y los muebles de la cocina tienen una única función y es la de hacer un entorno más agradable para el usuario. Algunos de estos elementos, como la mesa y el suelo lleva un Box Collider añadido para que se produzcan las colisiones, y los objetos y el robot no los atraviesen.

Los puntos de luz son otro de los elementos que se encuentran en el entorno, estos se han colocado en determinados lugares para aumentar la visibilidad en los mismos. Otro elemento, que se encuentra en el entorno es la cámara, la localización de esta ha sido tal que proporcione al usuario una buena visualización de todos los elementos que se van a manipular. Por último, destacar que también se ha incluido un objeto vacío en el que se incluirá el script correspondiente para realizar la comunicación con Matlab y guardar los mensajes que recibe de este.

6.3.2. Robot

El robot utilizado es un diseño del robot UR3, este ha sido obtenido de recursos de GitHub [32], el robot completo se muestra en la Figura 6.6, de este se han obtenido los últimos tres eslabones y el efector final junto a una pinza, ya que no es necesario el uso del robot completo, pues no se hará uso de las articulaciones de las que dispone que proporcionan que el robot se mueva en torno a seis grados de libertad en el espacio.

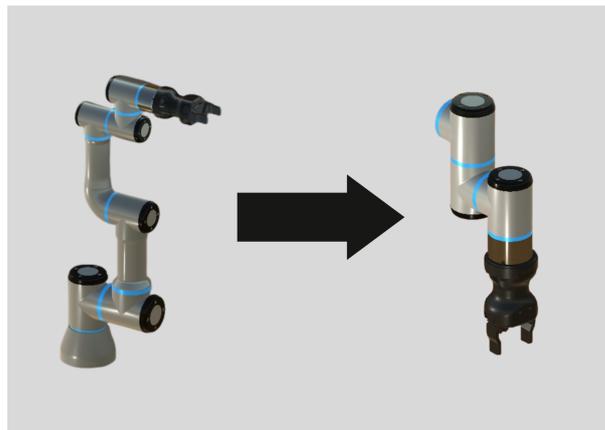
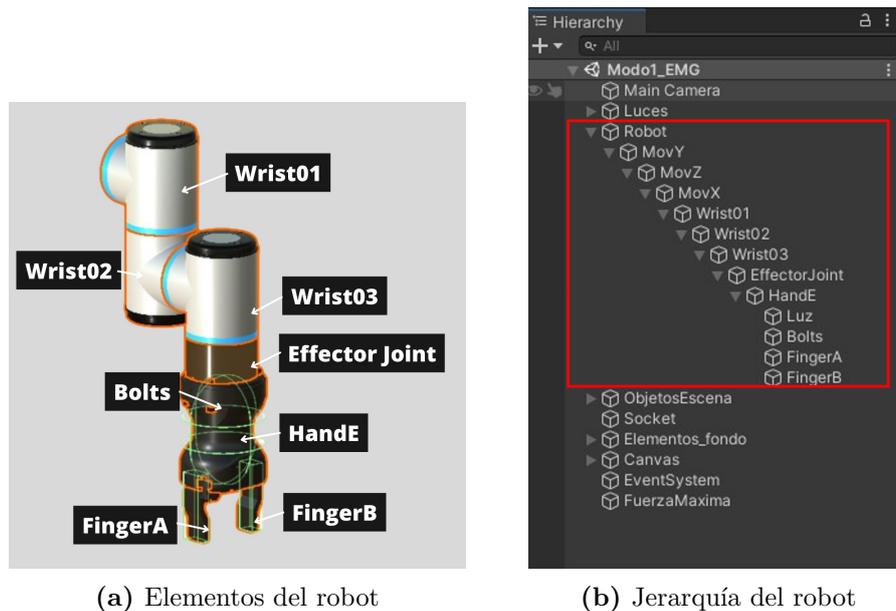


Figura 6.6: Diseño virtual del robot UR3

El robot resultante posee tres grados de libertad, además del movimiento de la pinza, pues únicamente puede moverse en los ejes x , y y z . Para realizar estos movimientos no se hace uso de las articulaciones del robot, ya que no conseguiríamos este movimiento de robot cartesiano con ellas. En su lugar se han creado tres Empty Objects u objetos vacíos, a los que les añadiremos el componente de Articulation Body.

En la Figura 6.7 se muestra la jerarquía de los elementos del robot, y el lugar dónde se encuentran cada uno de los elementos, excepto los objetos vacíos, ya que estos no son visibles. A continuación, se detallan cada uno de estos elementos:

- Robot: es un objeto vacío, su única función es la de ser la base del Articulation Body.
- MovY, MovX y MovZ: son también objetos vacíos, y gracias a ellos se produce el movimiento lineal en los ejes x , y y z . Entre sus componentes encontramos el script para



(a) Elementos del robot

(b) Jerarquía del robot

Figura 6.7: Elementos que componen el brazo robótico

el movimiento, y el Articulation Body. Este último está configurado según vemos en la Figura 6.8, como una articulación prismática por lo que el movimiento sólo puede realizarse de manera rectilínea a lo largo de un eje. Además, este movimiento está limitado, esto es así para que el robot no pueda desplazarse fuera de los límites de la mesa, y por ende fuera de los límites de lo que el usuario visualiza durante la prueba. Cada uno de estos límites es diferente para cada eje.

Por otra parte, se han fijado los siguientes parámetros relativos a las físicas del brazo robótico:

- Stifness: 100000
- Damping: 9000
- Force Limit: 3.402823e+38

Estos parámetros se han establecido de manera que permitan al robot moverse a una velocidad adecuada para facilitar al usuario el control.

La ubicación de estos tres objetos es la misma, es decir, su componente Transform está situado en el mismo lugar, este lugar se muestra en la Figura 6.9, y cómo se observa se encuentra entre las dos pinzas, esto ha sido así para que en los modos que requieren de la automatización para el agarre de los objetos, se asigne la posición relativa de este a la articulación y el objeto quede en un lugar óptimo para su agarre.

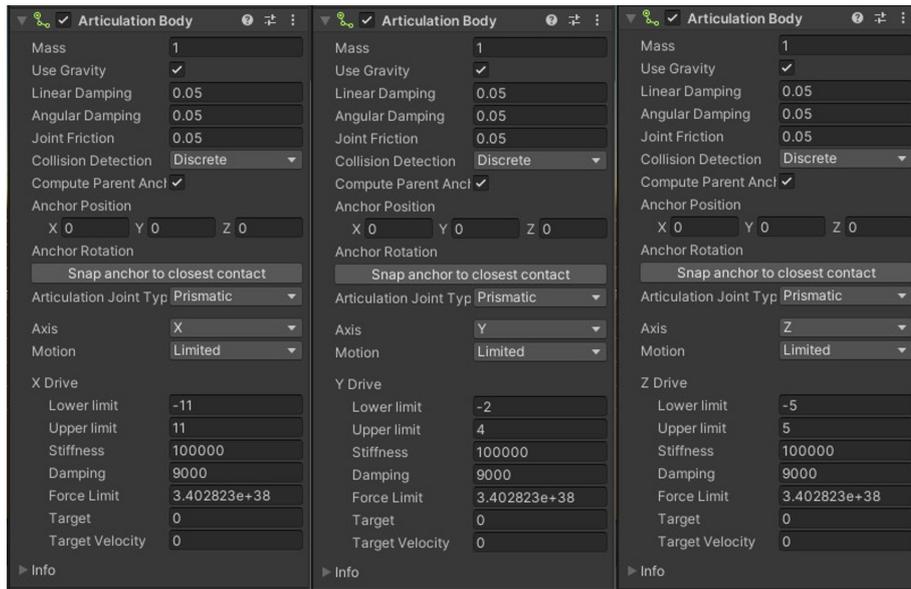


Figura 6.8: Componentes Articulation Body de cada uno de los tres ejes de movimiento del brazo robótico

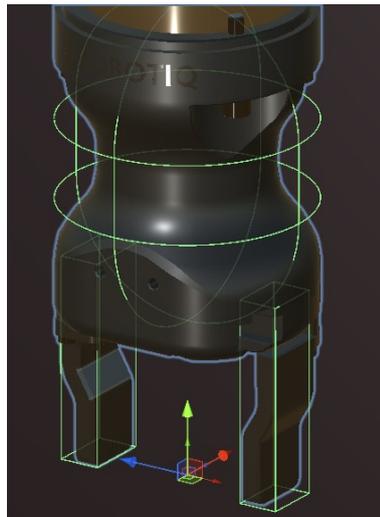


Figura 6.9: Posición de los objetos MovX, MovY y MovZ en el espacio

- Wrist01, Wrist02, Wrist03 y Effector Joint: son los tres eslabones finales del robot y su articulación final. Dotan al brazo robótico de una apariencia física visible. Estos elementos se han designado como articulaciones fijas dentro del Articulation Body, esto es necesario para mantener la cadena cinemática de eslabones, su única función es la de visualizar el robot, pues no tienen ningún script adjunto que defina su movimiento, pero al añadir los scripts en un objeto superior en la jerarquía (MovX, MovY y MovZ) el resto de elementos inferiores se mueven con él.

- **HandE:** se corresponde con el cuerpo que sujeta las dos pinzas, el componente *Articulation Body* añadido en este caso también está configurado como fija. Este objeto tiene además adjunto un componente *Capsule Collider*, por lo que generamos un colisionador estático, y no dinámico por no añadir un *RigidBody*, tal y como se ha explicado en el apartado 5. Para los elementos del robot que requieren colisiones, como los elementos de la pinza para poder agarrar los objetos, se han utilizado colisionadores estáticos, ya que los dinámicos producen que el robot se mueva de manera incontrolable por la acción de otros cuerpos.
- **Luz:** se ha decidido colocar un punto de luz en medio de las pinzas para que el usuario sea capaz de observar gracias a la luz reflejada en la mesa dónde se encuentra, para facilitar la acción de coger objetos.
- **Bolts:** estos elementos tienen un fin puramente decorativo, simulan los tornillos de la pinza. Se les ha adjuntado una articulación fija, ya que no sufrirán movimientos respecto al robot.
- **FingerA y FingerB:** se corresponden con cada una de las pinzas que se abrirán y cerrarán para agarrar los objetos, para ello se ha añadido una articulación prismática en cada una de ellas. El movimiento de estas articulaciones estará limitado mediante un script, para que las pinzas no se abran más allá de la anchura del robot.

Por lo tanto, gracias a esta estructura es posible que el brazo robótico se desplace por el espacio según lo haría un robot cartesiano, es decir, con tres grados de libertad según movimientos rectilíneos en los tres ejes del espacio. Además, la pinza proporciona la capacidad de agarrar objetos.

Por otra parte, destacar que el robot se mueve por el espacio a una velocidad aproximada de 0.017 m/s, teniendo en cuenta que se ha considerado que los ejes de coordenadas de Unity están en decímetros. Esta velocidad es aproximada ya que depende de los valores de rigidez impuestos en la articulación, y no es un valor fijado.

Además, considerar que respecto a los movimientos de articulación prismática que se producen por acción del usuario de manera manual, estos se realizan en base al mensaje recibido de manera que un mensaje provoca un desplazamiento aproximado de 1.1 cm en el robot.

6.3.3. Modos de uso

Se han creado tres escenas diferentes en Unity para realizar tres modos distintos, en estas tres escenas se representa el mismo entorno, que ha sido detallado en los apartados anteriores. Estos modos son los siguientes:

- **Modo 1:** Movimiento libre en tres ejes y grasping manual.
 - **Modo 2:** Movimiento libre en dos ejes y grasping automático.
-

- Modo 3: Movimiento y grasping automático.

La diferencia existente entre cada uno de estos modos reside en la funcionalidad de los mismos, ya que en cada uno de ellos se controlará al robot de manera diferente. Para esto se implementarán algoritmos diferentes en cada modo. En este apartado se explicarán las funcionalidades de control de cada modo, así como los scripts implementados en cada uno de ellos para conseguir este funcionamiento.

6.3.3.1. Modo 1: Movimiento libre en tres ejes y grasping manual

En este primer modo el usuario tiene plena libertad de movimiento con el brazo robótico por lo que el conjunto de acciones que puede realizar en el entorno son nueve, estas son el movimiento tanto en sentido positivo como negativo de los tres ejes, la apertura y cierre de la pinza y el reposo del robot. Estas acciones las realizará según el mensaje que obtiene del socket, este mensaje ha sido enviado desde Matlab, y contiene un string con un número del 0 al 9. A continuación, se detallan cada uno de los mensajes que puede recibir Unity y las acciones que desencadena ese mensaje:

- “0”: este caso implica que el robot no realice ninguna acción y se mantenga estático.
- “1”: este mensaje desencadena un movimiento positivo en el eje y , esto significa un movimiento hacia arriba del robot.
- “2”: este mensaje produce un movimiento negativo en el eje y , es decir, el robot baja.
- “3”: este mensaje produce un movimiento positivo en el eje x , o lo que es lo mismo un desplazamiento del robot a la derecha.
- “4”: este mensaje produce un movimiento negativo en el eje x , por lo que se observará un movimiento del robot a la izquierda.
- “5”: este mensaje produce un movimiento positivo en el eje z , el robot se moverá hacia atrás.
- “6”: este mensaje produce un movimiento negativo en el eje z , o lo que es lo mismo, el desplazamiento hacia delante del robot.
- “7”: tras recibir este mensaje se producirá una apertura completa de las pinzas hasta el límite fijado.
- “8”: cuando se recibe este mensaje el robot cerrará por completo las pinzas hasta el límite fijado.

En el flujograma de la Figura 6.10, se puede observar cada una de las acciones del robot que se han explicado, desencadenadas a causa del mensaje recibido desde Matlab y que recibe el nombre de “msg”. Estas acciones son realizadas gracias a los scripts implementados en los distintos objetos de la escena, estos scripts se exponen a continuación. Todos estos scripts se encuentran alojados en un repositorio Github.

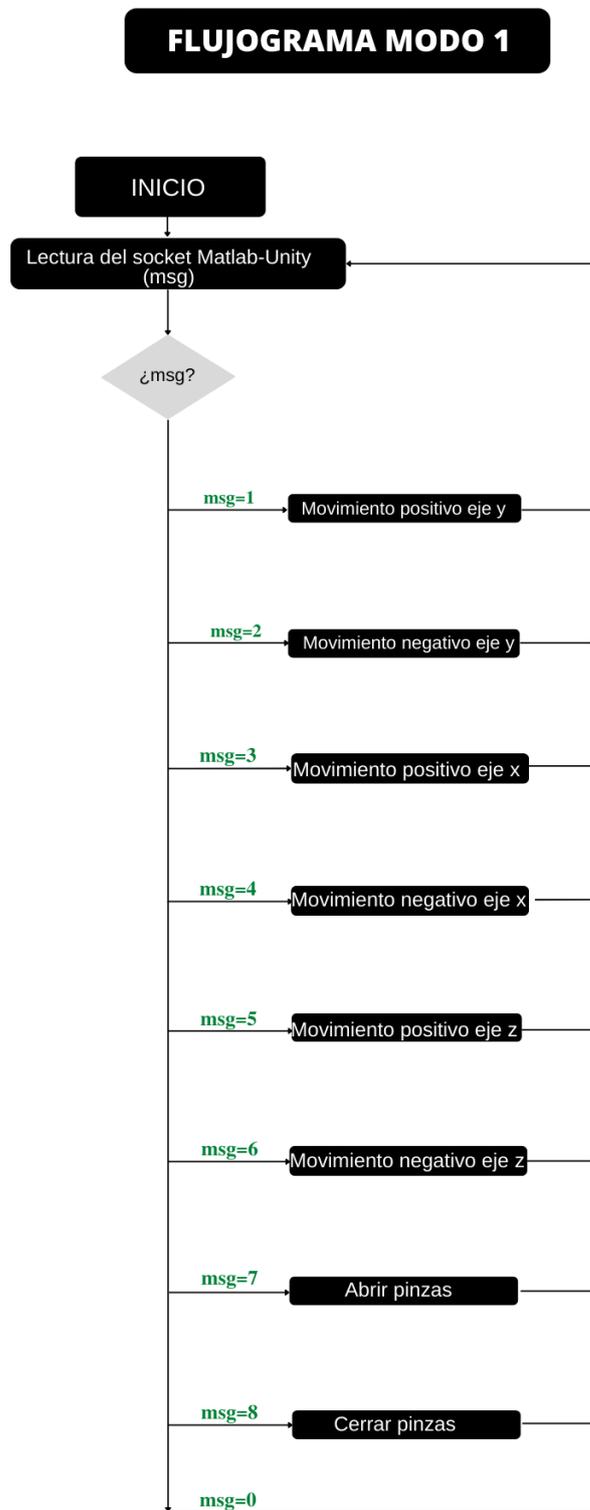


Figura 6.10: Flujograma de funcionamiento de Unity en modo 1

- **Movimientox.cs**: este script está asociado al objeto MovX, por lo tanto controlará la articulación prismática del eje x . Se activa únicamente cuando el mensaje “msg” es “3” o “4”, en cuyo caso enviará a la articulación el movimiento que debe realizar.
- **Movimientoy.cs**: funciona de la misma manera que el script anterior, pero en este caso está asociado al objeto MovY, por lo que controla la articulación del eje y , y sólo hace una acción cuando “msg” es igual a “1” o “2”.
- **Movimientoz.cs**: al igual que los dos scripts anteriores tiene la función de mover al robot de manera lineal, en este caso en el eje z , por lo que se asocia al objeto MovZ, y realiza una acción cuando “msg” es igual a “5” o “6”.
- **PincherController.cs**: este script se encuentra adjunto como componente en el objeto HandE, aunque no produce ningún control en el mismo. La función de este script será la de enviar a las pinzas la función de abrir o cerrar, cuando el mensaje “msg” sea igual a “7” u “8”, respectivamente.
- **PincherFingerController.cs**: este script estará adjunto tanto en el FingerA como en el FingerB, es el encargado de fijar los límites de las articulaciones de estos para que no sobresalgan del robot al abrirse, y de ejecutar los movimientos de estas pinzas, una vez el script anterior ha recibido el mensaje.
- **barraFuerza.cs**: este script, será el encargado de informar al usuario del estado de la pinza, ya que en algunos casos cuando se encuentra agarrando un objeto, puede que el estado de la pinza no pueda observarse correctamente. Esto lo realiza gracias a la información que recibe del script PincherController.cs de la fuerza que está realizando la pinza en un momento determinado. Este tipo de feedback se denomina biofeedback y es utilizada en muchos casos en entrenamiento en sistemas controlados por señales EMG.

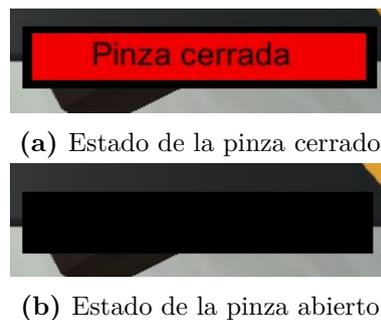


Figura 6.11: Feedback visual del estado de la pinza del brazo robótico

- **readSocket**: este script es el encargado de realizar la comunicación con Matlab, está asociado al objeto vacío Socket. Funciona de manera que guarda en una variable pública, para poder ser leído en el resto de scripts, el mensaje recibido de Matlab.

Con este modo se busca dar plena libertad al usuario del brazo robótico. Es por ello que es el que más acciones de control necesita para una determinada tarea. A la vez que el agarre,

al no realizarse de manera automática como en el resto de modos, supone una dificultad en cuanto a que debe obtenerse un control profundo del sistema.

6.3.3.2. Modo 2: Movimiento libre en dos ejes y grasping automático

En este segundo modo se reduce el número de acciones que el usuario puede realizar a seis. En este caso el usuario puede moverse libremente en el sentido positivo y negativo de los ejes x y z . Además, existe una tercera posibilidad de agarrar el objeto más cercano. A continuación se detallan en profundidad cada uno de los movimientos los cuales el usuario puede controlar, según los siguiente mensajes recibidos desde Matlab:

- “0”: cuando se recibe este mensaje el robot se mantiene en reposo por lo que no se altera su posición.
- “1”: cuando Unity recibe este mensaje se produce un movimiento negativo en el eje x , es decir, el robot se desplaza a la izquierda.
- “2”: este mensaje produce un movimiento positivo en el eje x , por lo que el robot se mueve hacia la derecha.
- “3”: este mensaje provoca que el robot realice un movimiento positivo en el eje z , es decir, hacia atrás.
- “4”: este mensaje provoca que el robot realice un movimiento negativo en el eje z , es decir, hacia delante.
- “5”: cuando se recibe este mensaje, se comprobará si el robot tiene agarrado un objeto o no, mediante la comprobación de una variable llamada “pieza”, que será igual a 1 cuando tenga un objeto agarrado y 0 en caso contrario. Por lo tanto, si este es 0 y “msg” igual a “5”, se ordenará al robot a que vaya a coger el objeto más cercano, realizando de manera autónoma los movimientos necesarios. En caso de que ya tenga un objeto agarrado, es decir “pieza” sea igual a 1, el robot soltará el objeto en la posición del plano xz en la que se encuentre.

El funcionamiento de este modo viene detallado en el flujograma de la Figura 6.12. En este se ha realizado un esquema de las decisiones que realiza el programa dependiendo del mensaje recibido desde Matlab, denominado “msg”. Al igual que en el apartado anterior se han programado distintos scripts para realizar estas acciones, se detallan a continuación cada uno de ellos.

- Movimientoxmdos.cs: este script está asociado al objeto MovX, realizará los movimientos rectilíneos pertinentes en el eje x cuando el mensaje “msg” sea igual a “1” o “2”. Además, realizará los movimientos necesarios de manera automática para acercarse al objeto más cercano cuando se requiera.
 - Movimientozmdos.cs: este script está asociado al objeto MovZ, por lo que controla los movimientos de dicha articulación. Envía una acción a la articulación cuando el mensaje
-

“msg” es igual a “3” o “4”. También, realizará ciertos movimientos automáticos para acercarse al objeto más cercano.

- `MovimientoY.cs`: este script se ha adjuntado al objeto `MovY`. En este caso se realizarán movimientos en el eje y pero estos no estarán controlados por el mensaje “msg”, si no por las acciones de control del script `modo2cogerpieza.cs`.
- `PincherControllermodo2.cs`: el funcionamiento de este script es el mismo que en el modo 1, aunque al igual que ocurre con el movimiento en el eje y estos movimientos estarán controlados por el script `modo2cogerpieza.cs`.
- `PincherFingerControllermodo2.cs`: su funcionamiento se lleva a cabo al igual que en el modo 1.
- `modo2cogerpieza.cs`: este script es el encargado de dirigir los movimientos que se efectúan para coger o dejar un objeto. Para conocer cuál es el objeto que se encuentra más cerca al robot se realiza el cálculo de la distancia euclídea en el plano xz entre el robot y cada uno de los cinco objetos a manipular de la escena, una vez hemos obtenido las cinco distancias euclídeas se calcula cuál de estas es la menor, y esta será la respectiva al objeto más cercano.

Tal y como se ha explicado anteriormente dependiendo de si el robot tiene agarrado un objeto o no, la variable “pieza” será 1 o 0, y este script enviará las acciones pertinentes para agarrar o soltar un objeto. Estas acciones consisten en modificar varias veces una variable para que de manera secuencial se produzcan los movimientos en x , y , z y de la pinza.

- `readSocket`: este script coincide con el del modo anterior, ya que su función, al igual que en la del modo 1, es la recibir el mensaje de Matlab y guardarlo en un mensaje denominado “msg”.

Por lo tanto, gracias a este modo se automatiza parte del proceso con el fin de realizar las actividades en un menor tiempo, y sin ser necesarias tantas acciones de control como en el modo 1. Por otra parte, el usuario pierde el control de ciertas acciones, por lo que hay movimientos que no tendrá posibilidad de realizar.

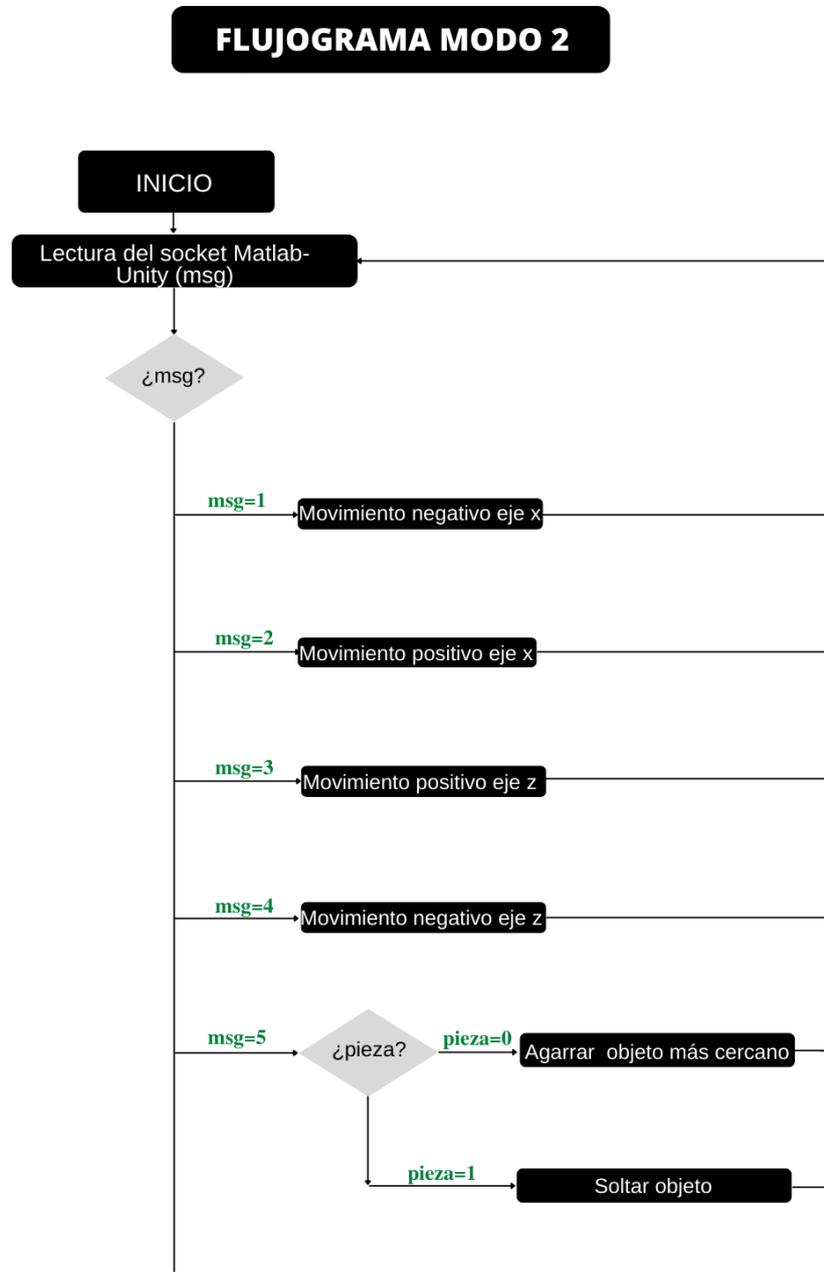
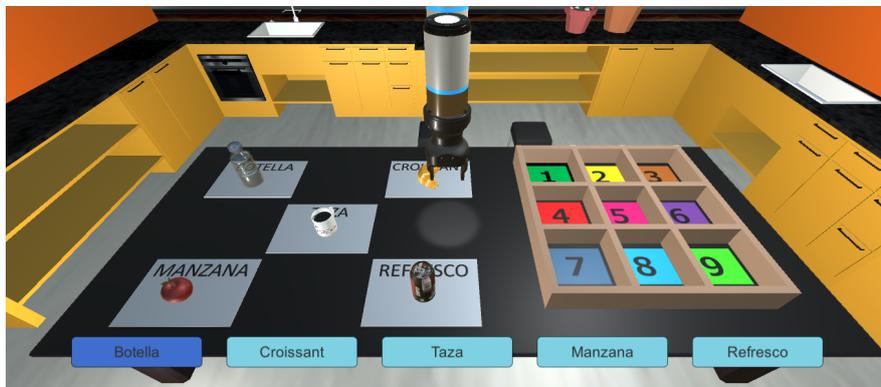


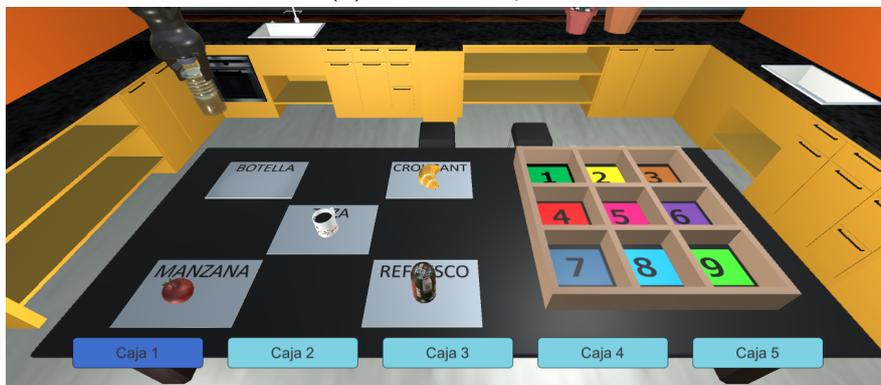
Figura 6.12: Flujo de funcionamiento de Unity en modo 2

6.3.3.3. Modo 3: Movimiento y grasping automático

Este es el último de los modos implementados, en el cual todos los movimientos del robot se realizan de forma automática y nuestras acciones de control se basarán en seleccionar el lugar al cuál queremos que se dirija. Esto se consigue gracias a la selección de una de las opciones que aparecerán en el menú que se ha implementado, el cual consta de varios botones. Se han creado dos menús diferentes, uno para la selección del objeto que se desea que el brazo robótico agarre y otro para la posición de destino en la que se desea que el robot suelte el objeto agarrado previamente. Este segundo menú aparecerá cuando hayamos seleccionado el objeto, momento en el que además desaparecerá ese primer menú. Ambos menús pueden observarse en la Figura 6.13.



(a) Menú de objetos



(b) Menú de destinos

Figura 6.13: Interfaz para la selección de distintas actividades a realizar por el brazo robótico

De esta manera las posibles acciones de control en este modo sólo se efectuarán para manipular cada uno de los menús existentes. Los posibles mensajes que se pueden recibir desde Matlab son:

- “0”: este mensaje implicará que el robot no realice ningún movimiento y permanezca de manera estática.

- “1”: gracias a este mensaje se podrá desplazar una posición a la derecha en el menú en el que el usuario se encuentre en ese momento.
- ”2”: con este mensaje se seleccionará la opción sobre la que el usuario se encuentre, es decir, la marcada con color azul oscuro.

El funcionamiento de este modo queda recogido en el flujograma mostrado en la Figura 6.14. En este se observa de forma esquematizada cada uno de los procesos que intervienen en la marcha de este modo.

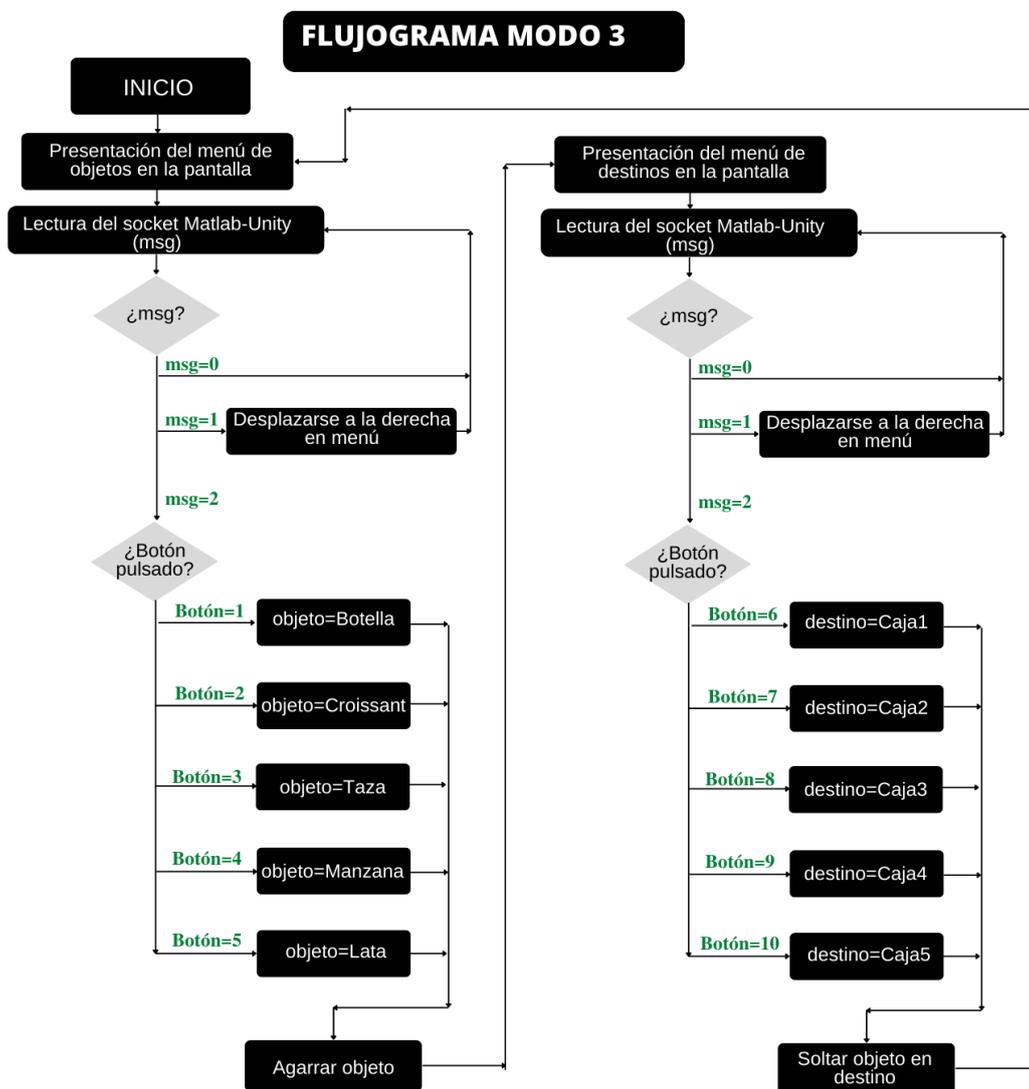


Figura 6.14: Flujograma de funcionamiento de Unity en modo 3

Al igual que con los dos modos anteriores, se ha necesitado de la programación de varios algoritmos para que este modo funcione tal y como se ha explicado. Para ello han sido necesarios los siguientes scripts:

- `Movimientoxtres.cs`: este script controla los movimientos respectivos a la articulación prismática que permite los movimientos en el eje x , estos movimientos son realizados a partir de las instrucciones guiadas dadas por el script `seguirtarget.cs`. Este script ha sido adjuntado al objeto `MovX`.
- `Movimientoymtres.cs`: este script funciona de igual manera que el anterior, pero de manera respectiva al objeto `MovY`, por lo que el movimiento se realizará en el eje y .
- `Movimientozmtres.cs`: de igual manera que los dos anteriores maneja el movimiento de una de las articulaciones a partir de las órdenes del script `seguirtarget.cs`. El movimiento que articula es el correspondiente al del objeto `MovZ`.
- `PincherControllermodo3.cs`: este script posee las mismas funcionalidades que en los anteriores modos. Recibe la señal, esta vez desde el script `seguirtarget.cs`, de abrir o cerrar las pinzas y la envía a los scripts de cada pinza.
- `PincherFingerControllermodo3.cs`: este script se encuentra adjunto en cada una de las pinzas, es decir, en `FingerA` y `FingerB`, y sus funcionalidades son las ya mencionadas para otros modos.
- `seguirtarget.cs`: este script está programado de manera que contiene dos máquinas de estados, una relativa a los cinco primeros botones, respectivos a los objetos a agarrar, y otra para los cinco últimos botones, correspondientes a los destinos dónde el objeto será soltado.

Para los cinco primeros botones, este script funcionará tal que, una vez seleccionado un determinado botón se guardará en la variable "objeto" el `GameObject` correspondiente al objeto seleccionado. Después se accederá a la primera máquina de estados y está realizará en cada uno de ellos el cambio de una variable pública, que actuará de aviso para activar cada uno de los movimientos necesarios hasta agarrar el objeto seleccionado.

Por otra parte, para los botones de destino, el script funcionará de manera que cuando se seleccione uno de ellos, se guardará en las variables "posx_destino" y "posz_destino" las posiciones en x y z de la caja que ha sido seleccionada como destino dónde se soltará el objeto. Tras esto se accederá a la segunda máquina de estados, que funcionará de forma similar a la anterior, pero esta vez la finalidad será dejar el objeto en lugar de cogerlo.

- `UIcontrol`: gracias a este script se realizan todos los procesos relacionados con el menú, como son el cambio entre los dos menús, el movimiento dentro del menú y la selección de las diferentes opciones.
 - `readSocketmtres`: este script se ejecutará de la misma manera que los anteriores, leyendo el mensaje de Matlab y guardándolo en una variable pública.
-

Por lo tanto, este modo presenta una interfaz con la que interactuaremos y que nos permite realizar una cantidad de movimientos del brazo robótico limitada. Además, se pierde el control directo sobre cada uno de los movimientos del robot. Aunque, las actividades que queramos llevar a cabo se realizarán con mayor rapidez.

7. Pruebas de usabilidad

Una vez desarrollado cada uno de los algoritmos que intervienen en el proceso, y que se han detallado en el apartado 6, se procede a realizar las pruebas con diferentes usuarios para evaluar el sistema mediante los resultados obtenidos en estas pruebas.

Cada una de estas pruebas se basará en la realización de una actividad, la cual consiste en agarrar un determinado objeto y desplazarlo hasta la caja que se precise. Esto será realizado para cada uno de los cinco objetos detallados anteriormente y en cada uno de los tres modos. Con el fin de evaluar las diferencias existentes entre los modos y la dificultad que supone para el usuario cada uno de ellos. Además, posteriormente a este ensayo se realizará una encuesta con el objetivo de conocer la experiencia del usuario.

En la realización de las pruebas intervienen diferentes software y hardware, además para obtener resultados concluyentes las pruebas han de realizarse de la misma manera, por ello se ha establecido un protocolo de pruebas a cumplir. Este protocolo será explicado en este apartado.

En este apartado también se explicarán las métricas que se han medido en cada una de las pruebas para obtener datos tanto cuantitativos como cualitativos con los que podamos realizar una evaluación. Además, se detallarán cada uno de los resultados que se han obtenido durante el desarrollo de las pruebas.

7.1. Protocolo de pruebas

En cada uno de los estudios realizados para los usuarios se han seguido una serie de pasos, estos conforman el protocolo de pruebas, el cual se detalla a continuación:

- Firma consentimiento: en primer lugar, el usuario firma un consentimiento en el que da permiso para el uso de sus señales biomédicas.
- Preparación de los medios necesarios: se realiza la preparación de los medios que intervienen en el proceso, detallados en el apartado 5. Para ello se iniciará en el ordenador utilizado cada uno de los software necesarios, además de la conexión y colocación del hardware necesario, estos son:
 - Noraxon: En primer lugar se realizarán las conexiones hardware requeridas, tanto la conexión del receptor MiniDTS al puerto USB del ordenador como la colocación de los sensores al usuario, limpiando previamente la zona dónde estos se van a colocar con alcohol y gasas.

Después, se abrirá el software Noraxon MR3, y se ajustará la configuración a las características requeridas, estas son:

- * Selección de dos canales EMG, ya que sólo usaremos dos sensores.
 - * Selección de los músculos con los que se va a trabajar.
 - * Selección del procesamiento de la señal que se desea, en este caso se ha seleccionado un filtrado de la señal mediante un filtro paso-banda, seguido de un rectificado y finalmente un suavizado.
 - * Activación del puerto para la comunicación HTTP que envía los datos a Matlab.
 - * Por último acceder a la pestaña “Measure”, ya que es en esta en la cual el programa medirá las señales EMG que recibe.
- Matlab: se abrirá cada uno de los tres archivos relativos a cada uno de los modos programados, estos archivos se encuentran alojados en Github.
 - Unity: se abrirá el proyecto de Unity el cuál se ha detallado en el apartado 6, y que contiene las tres escenas con los distintos modos. Este archivo se encuentra colgado en Github dentro de un repositorio privado.

Además, se conectará al ordenador una segunda pantalla, ya que en la realización de la prueba el usuario necesitará visualizar dos interfaces diferentes para el control del brazo robótico, estas son las respectivas a Unity y Matlab. Una vez, realizados todos estos pasos se procederá a realizar una prueba de simulacro para comprobar que el sistema funciona de una manera correcta.

- Medida de los umbrales del usuario: con el fin de adaptar el sistema a las capacidades del usuario, se fijarán dos valores a los que se denominarán umbrales, para los que se establezca que el usuario está realizando una contracción de cada uno de los músculos.

Cada uno de estos dos umbrales se especifican gracias a la visualización de las señales musculares en el software Noraxon, en el momento en el que el usuario está realizando las acciones de flexión, extensión y cocontracción.

Es necesario fijar estos umbrales lo suficientemente bajos para que el usuario consiga alcanzar el valor para un músculo tanto en la contracción del mismo como cuando lo hace simultáneamente con otro músculo, en este caso se suele alcanzar un valor menor. Por otra parte, el valor del umbral debe ser lo suficientemente alto como para que la coactivación no suponga un problema en cuánto al control, esto se produce porque la contracción de un determinado músculo supone la contracción involuntaria de otros músculos que también están interviniendo en el proceso.

En las Figuras 7.1, 7.2 y 7.3 se mostrarán ejemplos de flexión, extensión y cocontracción realizados por el usuario.



Figura 7.1: Lectura de una flexión



Figura 7.2: Lectura de una extensión



Figura 7.3: Lectura de una cocontracción

- Entrenamiento: se realizará una breve explicación del funcionamiento de cada uno de los modos, seguido de unos minutos en los que se ofrecerá plena libertad al usuario de realizar las actividades que desee en los diferentes modos con el fin de aprender a realizar el control y observar la comodidad del usuario. Este entrenamiento tendrá una extensión aproximada de 15 minutos.

En esta fase además se modificarán los umbrales si es necesario ya que puede ser que el usuario presente dificultades para alcanzar los valores de contracciones de los músculos establecidos previamente. También se comprueba que los sensores hayan sido colocados correctamente en el músculo, pudiendo modificarse esta colocación si se considera necesario.

- Realización de las pruebas: las pruebas consistirán en la realización de 15 actividades diferentes. Estas actividades se aleatorizarán previamente, de manera que cada usuario realice la prueba en un orden diferente. Esta aleatorización se realiza con el objetivo de que los resultados no se vean afectados por el cansancio del usuario en un modo determinado, ya que el modo 1, al requerir de más acciones de control, puede suponer la fatiga del usuario, por lo que si se realizase el primero el usuario realizaría las pruebas de los dos últimos modos fatigado y los resultados no estarían acorde con las capacidades reales del usuario.

Estas 15 actividades se corresponden con el agarre de un objeto y su desplazamiento a la caja requerida para cada uno de los cinco objetos y en los tres modos. Las 5 actividades a realizar en cada uno de los modos son las siguientes:

- Desplazar la botella a la caja número 1.
- Desplazar el croissant a la caja número 2.
- Desplazar la taza a la caja número 3.
- Desplazar la manzana a la caja número 4.
- Desplazar la lata a la caja número 5.

Durante la realización de estas pruebas se tomarán distintos datos que serán utilizados para realizar un análisis cuantitativo de las pruebas. Estas métricas se detallan en el siguiente apartado.

- Realización de un cuestionarios: al finalizar la prueba los usuarios realizarán un cuestionario con el fin de evaluar de una manera cualitativa las pruebas y comparar los resultados de las métricas, que se explicarán en el siguiente apartado, con la experiencia del usuario. Este cuestionario está dividido en tres bloques:
 - Preguntas generales: en esta parte del cuestionario se engloban preguntas tales como la edad, el sexo y otros antecedentes del usuario.
 - Control mioeléctrico: las preguntas realizadas en esta parte son las relacionadas con la lectura de señales EMG y la experiencia del usuario en cuanto a los sensores utilizados, y la interfaz relativa a la retroalimentación del estado de la máquina de estado controlada por el usuario.
 - Entorno virtual: estas preguntas se corresponden con la experiencia ligada al entorno virtual de Unity.
 - Valoración final: en estas preguntas el usuario hace una valoración general del sistema.

7.2. Métricas obtenidas

En las pruebas realizadas se han analizado varios parámetros los cuáles servirán como métricas para evaluar el funcionamiento del entorno virtual controlado mediante las señales EMG del usuario. Las métricas utilizadas han sido las siguientes:

- Orden de las pruebas: las 15 pruebas realizadas tienen un orden diferente para cada usuario, ya que este orden ha sido aleatorizado. Este orden, aunque no se tendrá en cuenta, es apuntado, para justificar en algún caso si el usuario se siente fatigado por una prueba que se ha realizado en último lugar.
 - Umbrales: los umbrales son los respectivos a los valores en μV de las contracciones de cada uno de los dos músculos que se están evaluando. Este valor nos proporciona información sobre la capacidad del usuario de contraer cierto músculo. Además, se compararán entre los diferentes usuarios.
 - Tiempo de realización de cada prueba: la duración de cada prueba es un parámetro importante a medir, ya que nos ofrece información de lo que el usuario tarda en conseguir el objetivo fijado para cada prueba, esto nos permite contrastar la rapidez en los distintos
-

modos y objetos. Además, con los parámetros de todos los usuarios de tiempo para un determinado objeto en un determinado modo se calculará el promedio, con el fin de comparar los valores individuales de los usuarios con esta media.

- Número de flexiones, extensiones y cocontracciones: se define la cantidad de movimientos, ya sea en flexión, en extensión o en cocontracción, cómo el número de veces que se envía un determinado mensaje a Unity para realizar esa acción determinada, es decir, una flexión se correspondería en el caso del modo 1 y estando en el nodo de moverse en el eje *y* como un "1" mandado a Unity. Estos datos son contabilizados en Matlab.

Este parámetro es importante ya que nos ofrece información sobre la cantidad de movimientos necesarios de cada usuario para alcanzar el objetivo propuesto, así como comparar entre el número de movimientos en flexión, extensión y cocontracción.

- Finalización de la prueba: este indicador será evaluado con un OK o NO OK, dependiendo de si el usuario ha completado la prueba con éxito o no. Se considera una prueba fallida cuando el usuario no consigue desplazar el objeto hasta la caja de destino, y por el contrario se considera como prueba exitosa cuando el usuario sí es capaz de hacerlo. Este parámetro nos dará indicios sobre el éxito o no de las pruebas realizadas, es realmente lo que se pretende conseguir, ya que el objetivo de la prueba es la manipulación y posicionamiento de un objeto según se indique.
- Tiempo total del ensayo: este valor mide el tiempo total de ensayo. Este tiempo ha de considerarse para que no sea demasiado extenso y no se produzca la fatiga mental o muscular del usuario.
- Comentarios: se detallarán cada una de las observaciones que se han visto durante las pruebas realizadas para justificar algunos hechos, como el fallo de alguna prueba o realizarla en un tiempo demasiado largo.

7.3. Resultados obtenidos

En este apartado se va a hacer un análisis de los datos obtenidos en los ensayos. En el anexo C se muestran las tablas de todos los datos que se han obtenido durante las pruebas. En el TFM realizado en colaboración con este se muestra el análisis de los parámetros respectivos a las señales EMG [17]. El análisis se realizará a nivel de usuario y a nivel de modo. Después se realizará un análisis de estos datos a nivel global. Y por último, se analizarán los resultados obtenidos en las encuestas.

7.3.1. Análisis a nivel del usuario

A continuación, se analizarán los datos obtenidos para cada uno de los ocho usuarios que han realizado las pruebas.

7.3.1.1. Usuario 1

Los resultados de estas pruebas en cuánto a tiempo y finalización o no de la prueba, así como el orden en el que se han realizado dichas pruebas se muestra en la Tabla 7.1.

DATOS PRUEBA USUARIO 1			
Nº PRUEBA	PRUEBA	TIEMPO (s)	FINALIZADO
1	Modo 1: Desplazar manzana a caja 4	183.70	OK
2	Modo 1: Desplazar croissant a caja 2	155.89	OK
3	Modo 2: Desplazar croissant a caja 2	116.47	OK
4	Modo 3: Desplazar taza a caja 3	43.20	OK
5	Modo 3: Desplazar refresco a caja 5	86.30	OK
6	Modo 3: Desplazar manzana a caja 4	42.89	OK
7	Modo 2: Desplazar manzana a caja 4	111.2	OK
8	Modo 2: Desplazar refresco a caja 5	70.23	OK
9	Modo 2: Desplazar taza a caja 3	55.30	OK
10	Modo 1: Desplazar botella a caja 1	357.17	OK
11	Modo 1: Desplazar refresco a caja 5	349.37	OK
12	Modo 2: Desplazar botella a caja 1	62.58	OK
13	Modo 3: Desplazar botella a caja 1	41.06	OK
14	Modo 3: Desplazar croissant a caja 2	98.16	OK
15	Modo 3: Desplazar taza a caja 3	127.84	OK

Tabla 7.1: Datos obtenidos para el usuario 1

Se han representado estos datos mediante la gráfica mostrada en la Figura 7.4, contrastando los datos del usuario 1 y los datos promedio globales de todos los usuarios. Los valores de los cuáles se han obtenido estos promedios vienen reflejados en el Anexo B. El número de prueba representada en el eje x pertenece a la descripción presentada en la Tabla 7.1, pero estas han sido ordenadas por modo.

Para este usuario, destacar las siguientes observaciones respecto a los datos obtenidos y anotaciones tomadas durante las pruebas:

- En primer lugar, destacar que este usuario ha realizado todas las pruebas de manera correcta.
- Respecto a los objetos del modo 1, puede observarse que la prueba 10, correspondiente a la botella es la prueba que más tiempo ha durado, esto es porque el usuario presentó dificultades para agarrarla teniendo que, finalmente volcarla para ello. Esto también se observa en la prueba 11.
- Respecto al modo 2, subrayar que el usuario se encuentra por debajo a los datos promedios en cuánto a tiempo en todas las pruebas a excepción de la 3.

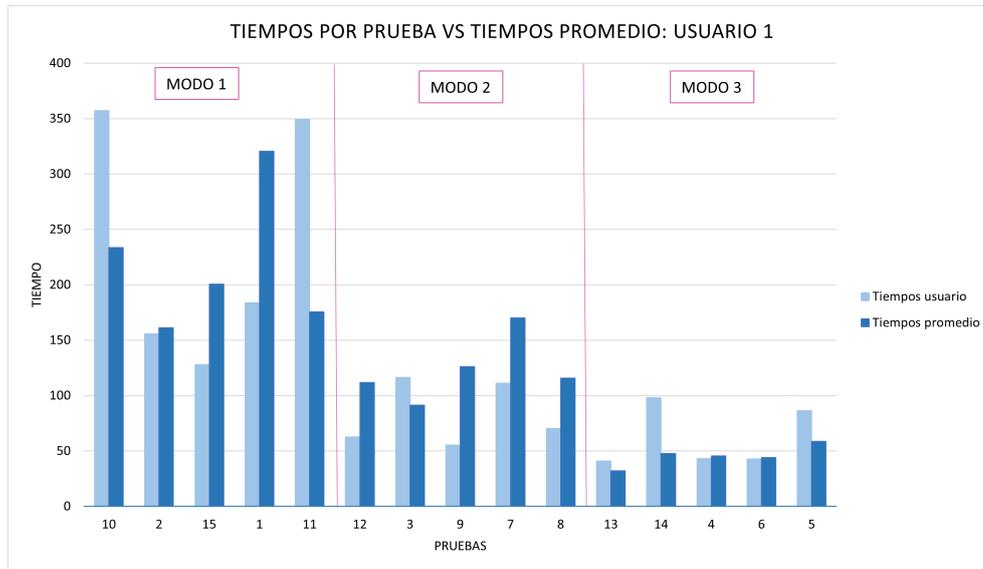


Figura 7.4: Resultados de los tiempos de las pruebas del usuario 1 con respecto al promedio.

- Respecto al modo 3, se observa que en dos de las pruebas el usuario ha tardado un tiempo superior al promedio, esto ha sido debido a que no ha elegido a la primera las opciones del menú correctas, por lo que el tiempo ha sido mayor.

7.3.1.2. Usuario 2

Los resultados de estas pruebas en cuánto a tiempo y finalización o no de la prueba, así como el orden en el que se han realizado dichas pruebas se muestra en la Tabla 7.2.

Se han representado estos datos mediante la gráfica mostrada en la Figura 7.5, contrastando los datos del usuario 2 y los datos promedio globales de todos los usuarios. Los valores de los cuáles se han obtenido estos promedios vienen reflejados en el Anexo B. El número de prueba representada en el eje x pertenece a la descripción presentada en la Tabla 7.2, pero estas han sido ordenadas por modo.

Para este usuario, destacar las siguientes observaciones respecto a los datos obtenidos y anotaciones tomadas durante las pruebas:

- En primer lugar, destacar que este usuario ha fallado en la prueba 11 ya que no consigue realizar los objetivos de dicha prueba, por lo que estos datos no han sido tenidos en cuenta para el cálculo de promedios.
- Respecto al modo 1, puede observarse que las pruebas que sí se han realizado con éxito poseen un valor de tiempo menores a los promedios.

DATOS PRUEBA USUARIO 2			
Nº PRUEBA	PRUEBA	TIEMPO(s)	FINALIZADO
1	Modo 1: Desplazar croissant a caja 2	158.83	OK
2	Modo 2: Desplazar manzana a caja 4	60.32	OK
3	Modo 3: Desplazar manzana a caja 4	52.08	OK
4	Modo 3: Desplazar croissant a caja 2	38.15	OK
5	Modo 3: Desplazar refresco a caja 5	39.44	OK
6	Modo 1: Desplazar taza a caja 3	187.16	OK
7	Modo 2: Desplazar refresco a caja 5	85.18	OK
8	Modo 1: Desplazar refresco a caja 5	100.87	OK
9	Modo 3: Desplazar botella a caja 1	33.85	OK
10	Modo 2: Desplazar taza a caja 3	64.73	OK
11	Modo 1: Desplazar botella a caja 1	-	NO OK
12	Modo 3: Desplazar taza a caja 3	43.20	OK
13	Modo 1: Desplazar manzana a caja 4	111.03	OK
14	Modo 2: Desplazar croissant a caja 2	42.10	OK
15	Modo 2: Desplazar botella a caja 1	54.52	OK

Tabla 7.2: Datos obtenidos para el usuario 2

- Respecto al modo 2, ocurre de igual manera que para el modo 1 ya que el usuario se encuentra por debajo a los datos promedios de tiempo.
- Respecto al modo 3 puede observarse un comportamiento similar al de los dos modos anteriores. Destacar que en la prueba 3 se observa como el usuario ha tardado un tiempo superior al promedio.

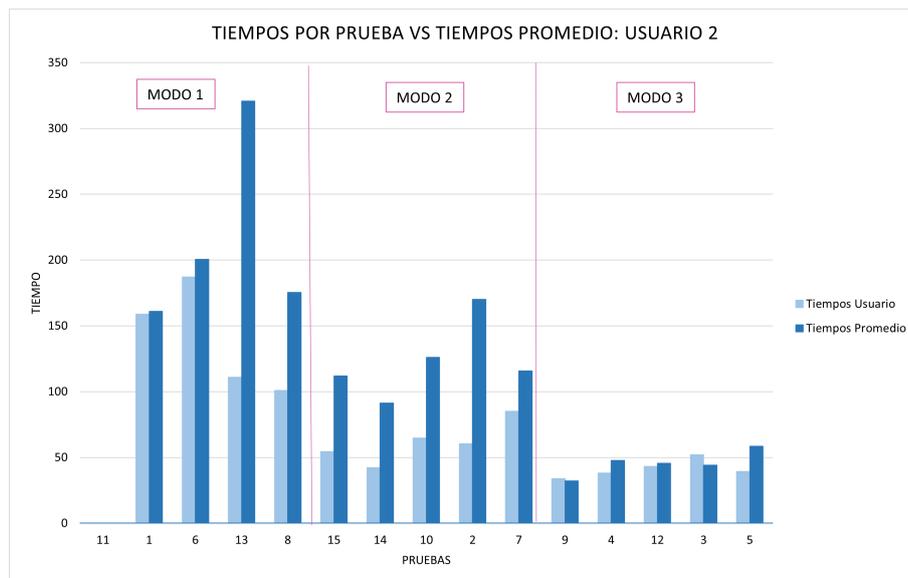


Figura 7.5: Resultados de los tiempos de las pruebas del usuario 2 con respecto al promedio.

7.3.1.3. Usuario 3

Los resultados de estas pruebas en cuánto a tiempo y finalización o no de la prueba, así como el orden en el que se han realizado dichas pruebas se muestra en la Tabla 7.3.

DATOS PRUEBA USUARIO 3			
Nº PRUEBA	PRUEBA	TIEM- PO(s)	FINALIZADO
1	Modo 2: Desplazar manzana a caja 4	189.88	OK
2	Modo 3: Desplazar croissant a caja 2	43.66	OK
3	Modo 3: Desplazar taza a caja 3	-	OK
4	Modo 1: Desplazar croissant a caja 2	162.00	OK
5	Modo 1: Desplazar manzana a caja 4	726.92	OK
6	Modo 3: Desplazar refresco a caja 5	58.27	OK
7	Modo 1: Desplazar refresco a caja 5	148.43	OK
8	Modo 2: Desplazar croissant a caja 2	83.14	OK
9	Modo 3: Desplazar manzana a caja 4	43.54	OK
10	Modo 2: Desplazar taza a caja 3	64.73	OK
11	Modo 2: Desplazar botella a caja 1	133.72	OK
12	Modo 1: Desplazar botella a caja 1	-	NO OK
13	Modo 1: Desplazar taza a caja 3	289	OK
14	Modo 3: Desplazar botella a caja 1	34,12	OK
15	Modo 3: Desplazar taza a caja 3	69,53	OK

Tabla 7.3: Datos obtenidos para el usuario 3

Se han representado estos datos mediante la gráfica mostrada en la Figura 7.6, contrastando los datos del usuario 3 y los datos promedio globales de todos los usuarios. Los valores de los cuáles se han obtenido estos promedios vienen reflejados en el Anexo B. El número de prueba representada en el eje x pertenece a la descripción presentada en la Tabla 7.3, pero estas han sido ordenadas por modo.

Para este usuario, destacar las siguientes observaciones respecto a los datos obtenidos y anotaciones tomadas durante las pruebas:

- En primer lugar, respecto al modo 1, puede observarse que la prueba 12 no se ha finalizado correctamente, esto ha ocurrido porque el usuario no ha realizado el control de manera correcta, pues en varias ocasiones hace movimientos no deseados provocando que se le caiga el objeto sin desearlo, finalmente se le cae el objeto en una zona imposible de agarrar por lo que se da la prueba como finalizada. El resto de pruebas en el modo 1 transcurren de modo correcto aunque los tiempos son superiores a la media.
- Respecto al modo 2, la prueba 3 no se finalizó satisfactoriamente. En este caso el objeto fue depositado en un límite de la caja, y a causa de las físicas implementadas en Unity, el objeto rebotó y cayó al suelo. El resto de modos se realizó correctamente.

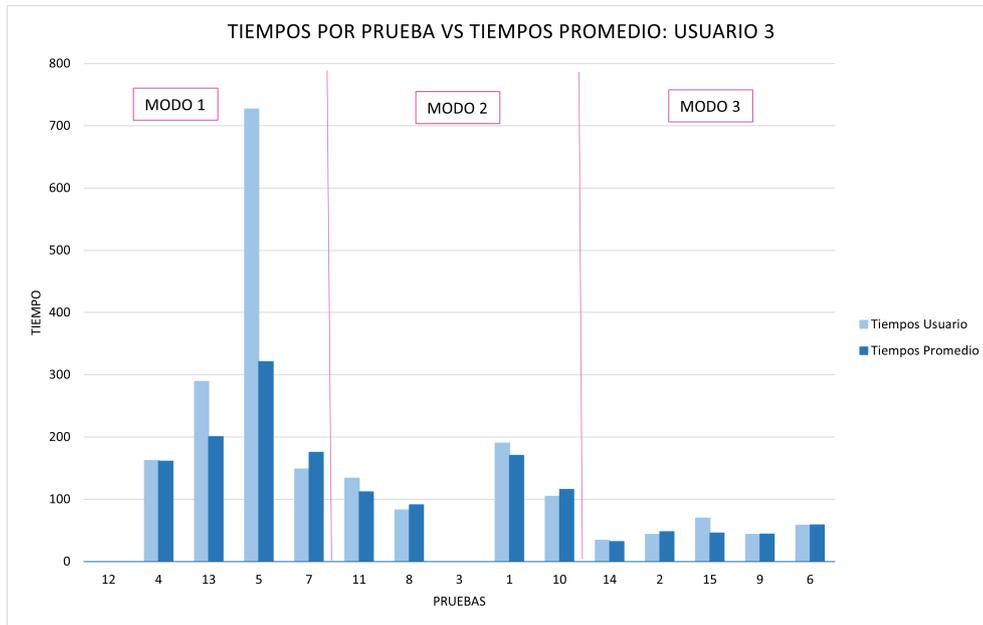


Figura 7.6: Resultados de los tiempos de las pruebas del usuario 3 con respecto al promedio.

- Respecto al modo 3, puede observarse que todos los objetos son llevados a su lugar final correctamente.

7.3.1.4. Usuario 4

Los resultados de estas pruebas en cuánto a tiempo y finalización o no de la prueba, así como el orden en el que se han realizado dichas pruebas se muestra en la Tabla 7.4.

Se han representado estos datos mediante la gráfica mostrada en la Figura 7.7, contrastando los datos del usuario 4 y los datos promedio globales de todos los usuarios. Los valores de los cuáles se han obtenido estos promedios vienen reflejados en el Anexo B. El número de prueba representada en el eje x pertenece a la descripción presentada en la Tabla 7.4, pero estas han sido ordenadas por modo.

Para este usuario, destacar las siguientes observaciones respecto a los datos obtenidos y anotaciones tomadas durante las pruebas:

- En primer lugar, respecto al modo 1, las pruebas han sido realizadas en tiempos similares a los promedios. No es así el caso de la prueba 1, en la que el usuario tarda más tiempo del promedio.
- Respecto al modo 2, la prueba 2 no se ha cumplimentado de manera satisfactoria, esto ocurre porque el usuario deposita el objeto en una caja errónea, y el objeto se gira, por lo que se imposibilita su agarre, debido a que la pinza no permite el giro de la misma. En cuanto al resto de pruebas los tiempos exceden a los promedios del resto de usuarios.

DATOS PRUEBA USUARIO 4			
Nº PRUEBA	PRUEBA	TIEM-PO(s)	FINALIZADO
1	Modo 1: Desplazar croissant a caja 2	251.73	OK
2	Modo 2: Desplazar taza a caja 3	155.52	NO OK
3	Modo 2: Desplazar croissant a caja 2	124.31	OK
4	Modo 3: Desplazar refresco a caja 5	41.64	OK
5	Modo 3: Desplazar botella a caja 1	21.32	OK
6	Modo 2: Desplazar manzana a caja 4	235.52	OK
7	Modo 1: Desplazar botella a caja 1	228.18	OK
8	Modo 1: Desplazar taza a caja 3	202.98	OK
9	Modo 1: Desplazar refresco a caja 5	141.58	OK
10	Modo 3: Desplazar croissant a caja 2	35.02	OK
11	Modo 3: Desplazar taza a caja 3	34.15	OK
12	Modo 1: Desplazar manzana a caja 4	325	OK
13	Modo 2: Desplazar botella a caja 1	131.44	OK
14	Modo 3: Desplazar manzana a caja 4	40.57	OK
15	Modo 2: Desplazar refresco a caja 5	120.3	OK

Tabla 7.4: Datos obtenidos para el usuario 4

- El modo 3 se ha cumplimentado por completo de manera exitosa cumpliendo con unos tiempos menores al promedio.

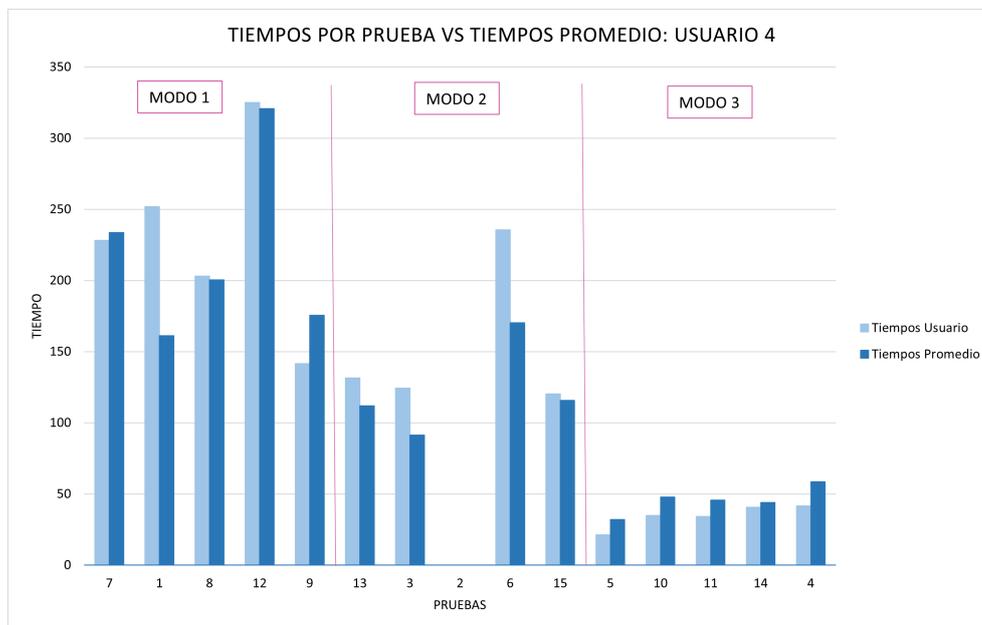


Figura 7.7: Resultados de los tiempos de las pruebas del usuario 4 con respecto al promedio.

7.3.1.5. Usuario 5

Los resultados de estas pruebas en cuánto a tiempo y finalización o no de la prueba, así como el orden en el que se han realizado dichas pruebas se muestra en la Tabla 7.5.

DATOS PRUEBA USUARIO 5			
Nº PRUEBA	PRUEBA	TIEM- PO(s)	FINALIZADO
1	Modo 2: Desplazar refresco a caja 5	45.84	OK
2	Modo 3: Desplazar botella a caja 1	21.32	OK
3	Modo 3: Desplazar taza a caja 3	38.65	OK
4	Modo 2: Desplazar botella a caja 1	74.27	OK
5	Modo 3: Desplazar manzana a caja 4	45.64	OK
6	Modo 1: Desplazar taza a caja 3	141.10	OK
7	Modo 1: Desplazar botella a caja 1	156.165	OK
8	Modo 2: Desplazar croissant a caja 2	62.77	OK
9	Modo 3: Desplazar refresco a caja 5	98.11	OK
10	Modo 3: Desplazar croissant a caja 2	40.99	OK
11	Modo 1: Desplazar croissant a caja 2	123.10	OK
12	Modo 1: Desplazar manzana a caja 4	310.00	OK
13	Modo 2: Desplazar taza a caja 3	141.63	OK
14	Modo 1: Desplazar refresco a caja 5	176.40	OK
15	Modo 2: Desplazar manzana a caja 4	179.72	OK

Tabla 7.5: Datos obtenidos para el usuario 5

Se han representado estos datos mediante la gráfica mostrada en la Figura 7.8, contrastando los datos del usuario 5 y los datos promedio globales de todos los usuarios. Los valores de los cuáles se han obtenido estos promedios vienen reflejados en el Anexo B. El número de prueba representada en el eje x pertenece a la descripción presentada en la Tabla 7.5, pero estas han sido ordenadas por modo.

Para este usuario, destacar las siguientes observaciones respecto a los datos obtenidos y anotaciones tomadas durante las pruebas:

- En primer lugar, en el modo 1 podemos observar que todas las actividades han sido realizadas correctamente, con tiempos menores o iguales al promedio.
- En el modo 2, no se observan hechos significativos distintos a los promedios, salvo en la prueba 13 en la que el usuario consigue llevar el objeto al destino con bastante menos movimientos al promedio.
- El modo 3 se ha cumplimentado de manera satisfactoria, con valores similares al promedio. Destacar, que en la prueba 4 el usuario selecciona primero un destino incorrecto, lo que provoca tener que volver a coger el objeto y colocarlo en el destino correcto, a causa de esto se observa un mayor tiempo.

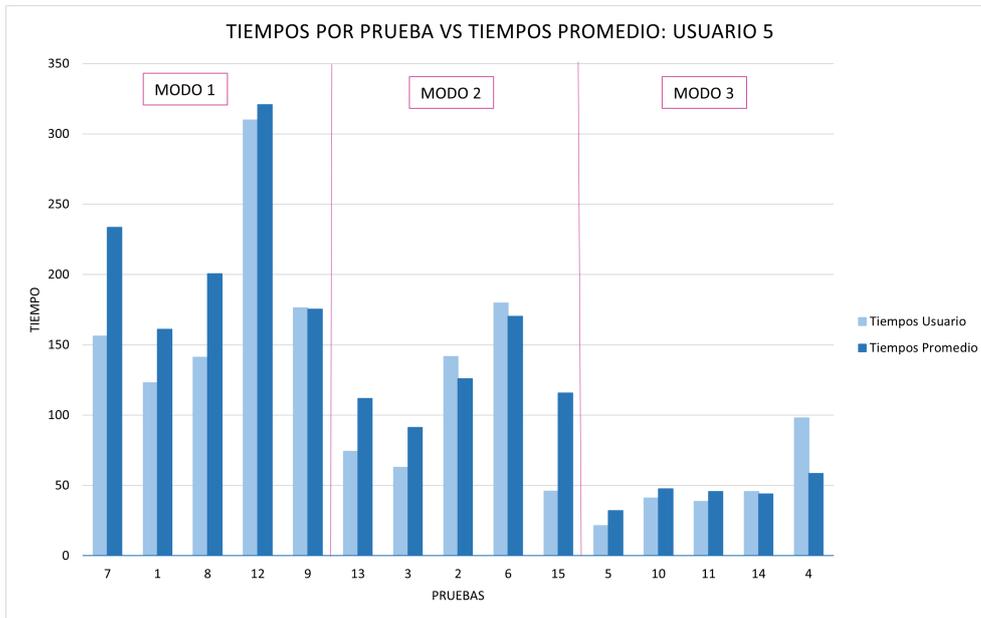


Figura 7.8: Resultados de los tiempos de las pruebas del usuario 5 con respecto al promedio.

7.3.1.6. Usuario 6

Los resultados de estas pruebas en cuánto a tiempo y finalización o no de la prueba, así como el orden en el que se han realizado dichas pruebas se muestra en la Tabla 7.6.

DATOS PRUEBA USUARIO 6			
Nº PRUEBA	PRUEBA	TIEMPO(s)	FINALIZADO
1	Modo 3: Desplazar botella a caja 1	32.41	OK
2	Modo 1: Desplazar manzana a caja 4	257.15	OK
3	Modo 2: Desplazar manzana a caja 4	190.94	OK
4	Modo 2: Desplazar botella a caja 1	139.44	OK
5	Modo 3: Desplazar croissant a caja 2	40.80	OK
6	Modo 3: Desplazar taza a caja 3	37.94	OK
7	Modo 3: Desplazar manzana a caja 4	38.47	OK
8	Modo 1: Desplazar refresco a caja 5	147.00	OK
9	Modo 1: Desplazar croissant a caja 2	130.00	OK
10	Modo 1: Desplazar botella a caja 1	199.07	OK
11	Modo 2: Desplazar taza a caja 3	169.25	OK
12	Modo 2: Desplazar croissant a caja 2	99.64	OK
13	Modo 3: Desplazar refresco a caja 5	40.68	OK
14	Modo 2: Desplazar refresco a caja 5	180.00	OK
15	Modo 1: Desplazar taza a caja 3	185.08	OK

Tabla 7.6: Datos obtenidos para el usuario 6

Se han representado estos datos mediante la gráfica mostrada en la Figura 7.9, contrastando los datos del usuario 6 y los datos promedio globales de todos los usuarios. Los valores de los cuáles se han obtenido estos promedios vienen reflejados en el Anexo B. El número de prueba representada en el eje x pertenece a la descripción presentada en la Tabla 7.6, pero estas han sido ordenadas por modo.

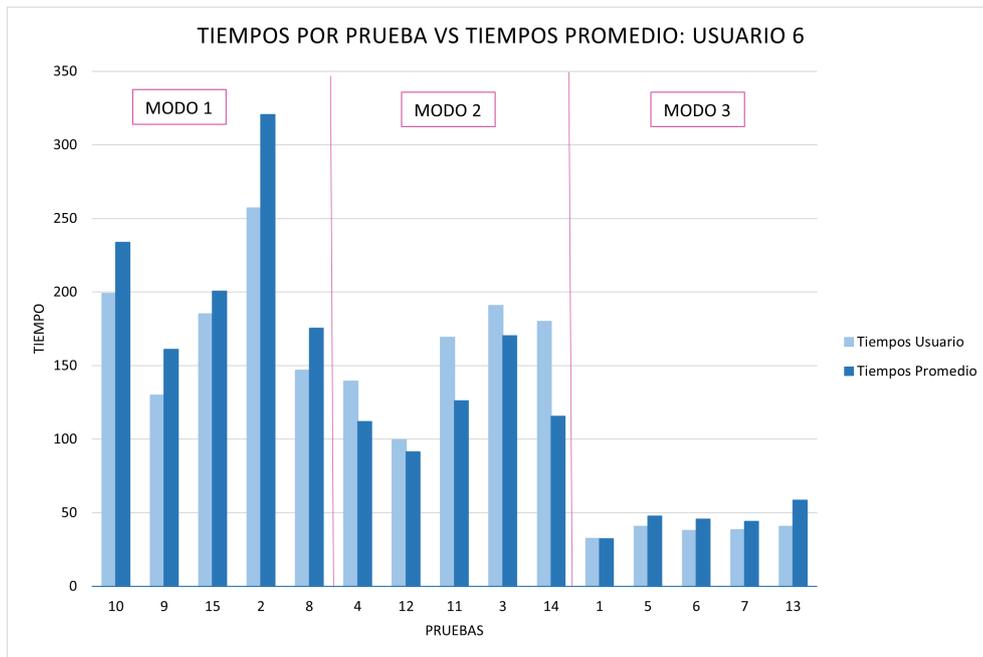


Figura 7.9: Resultados de los tiempos de las pruebas del usuario 6 con respecto al promedio.

Para este usuario, destacar las siguientes observaciones respecto a los datos obtenidos y anotaciones tomadas durante las pruebas:

- En primer lugar, en el modo 1 se ha completado de manera satisfactoria, además, consiguiendo unos tiempos menores a las medias.
- En el modo 2, se observa cómo los tiempos son en todos los casos superiores a los promedios. Sin embargo, todas las pruebas se realizan de manera correcta.
- En el modo 3, todas las pruebas se han realizado correctamente con tiempos menores a los promedios.

7.3.1.7. Usuario 7

Los resultados de estas pruebas en cuánto a tiempo y finalización o no de la prueba, así como el orden en el que se han realizado dichas pruebas se muestra en la Tabla 7.7.

DATOS PRUEBA USUARIO 7			
Nº PRUEBA	PRUEBA	TIEM- PO(s)	FINALIZADO
1	Modo 3: Desplazar botella a caja 1	32.41	OK
2	Modo 1: Desplazar taza a caja 3	245.14	OK
3	Modo 1: Desplazar manzana a caja 4	223.60	OK
4	Modo 2: Desplazar botella a caja 1	123.79	OK
5	Modo 2: Desplazar refresco a caja 5	150.60	OK
6	Modo 2: Desplazar croissant a caja 2	119.94	OK
7	Modo 2: Desplazar manzana a caja 4	226.85	OK
8	Modo 3: Desplazar croissant a caja 2	37.58	OK
9	Modo 3: Desplazar taza a caja 3	58.55	OK
10	Modo 3: Desplazar refresco a caja 5	52.02	OK
11	Modo 1: Desplazar refresco a caja 5	164.36	OK
12	Modo 1: Desplazar botella a caja 1	-	NO OK
13	Modo 1: Desplazar croissant a caja 2	-	NO OK
14	Modo 3: Desplazar manzana a caja 4	37.15	OK
15	Modo 2: Desplazar taza a caja 3	-	NO OK

Tabla 7.7: Datos obtenidos para el usuario 7

Se han representado estos datos mediante la gráfica mostrada en la Figura 7.10, contrastando los datos del usuario 7 y los datos promedio globales de todos los usuarios. Los valores de los cuáles se han obtenido estos promedios vienen reflejados en el Anexo B. El número de prueba representada en el eje x pertenece a la descripción presentada en la Tabla 7.7, pero estas han sido ordenadas por modo.

Para este usuario, destacar las siguientes observaciones respecto a los datos obtenidos y anotaciones tomadas durante las pruebas:

- Respecto al modo 1 puede observarse que dos de las pruebas, la 12 y 13 no se han terminado de manera correcta, esto ha sido debido a que estas pruebas se han realizado cuándo el usuario estaba fatigado y excedió los tiempos límite, por lo que estos datos no se han utilizado para los promedios. El resto de pruebas de este modo se realizaron de manera satisfactoria.
- Respecto al modo 2, se observa cómo la prueba 15 no se ha cumplimentado, esto ha sido también debida a la fatiga del usuario. El resto de pruebas se han cumplimentado exitosamente.
- Respecto al modo 3, todas las pruebas se han realizado correctamente. Los tiempos son menores o similares a los promedios.

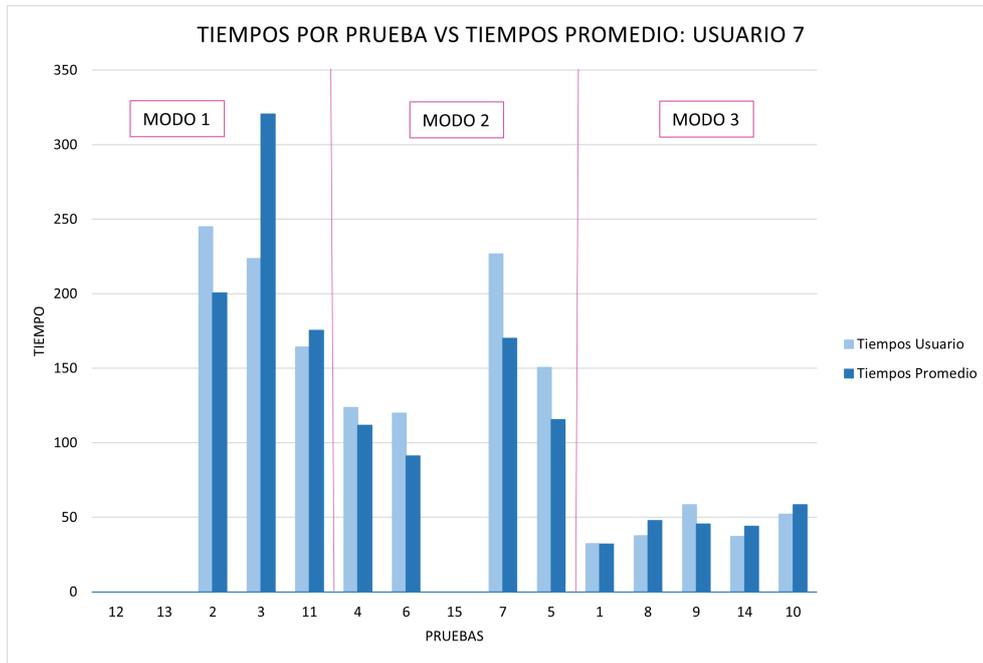


Figura 7.10: Resultados de los tiempos de las pruebas del usuario 7 con respecto al promedio.

7.3.1.8. Usuario 8

Los resultados de estas pruebas en cuánto a tiempo y finalización o no de la prueba, así como el orden en el que se han realizado dichas pruebas se muestra en la Tabla 7.8.

DATOS PRUEBA USUARIO 8			
Nº PRUEBA	PRUEBA	TIEM-PO(s)	FINALIZADO
1	Modo 2: Desplazar croissant a caja 2	81.65	OK
2	Modo 1: Desplazar taza a caja 3	225.62	OK
3	Modo 1: Desplazar manzana a caja 4	428.13	OK
4	Modo 3: Desplazar taza a caja 3	39.61	OK
5	Modo 3: Desplazar refresco a caja 5	52.35	OK
6	Modo 3: Desplazar botella a caja 1	40.70	OK
7	Modo 3: Desplazar croissant a caja 2	47.63	OK
8	Modo 1: Desplazar croissant a caja 2	146.06	OK
9	Modo 2: Desplazar botella a caja 1	174.76	OK
10	Modo 2: Desplazar manzana a caja 4	167.02	OK
11	Modo 1: Desplazar refresco a caja 5	-	NO OK
12	Modo 1: Desplazar botella a caja 1	227.67	OK
13	Modo 3: Desplazar manzana a caja 4	52.02	OK
14	Modo 2: Desplazar taza a caja 3	198.96	OK
15	Modo 2: Desplazar refresco a caja 5	168.15	OK

Tabla 7.8: Datos obtenidos para el usuario 8

Se han representado estos datos mediante la gráfica mostrada en la Figura 7.11, contrastando los datos del usuario 8 y los datos promedio globales de todos los usuarios. Los valores de los cuáles se han obtenido estos promedios vienen reflejados en el Anexo B. El número de prueba representada en el eje x pertenece a la descripción presentada en la Tabla 7.8, pero estas han sido ordenadas por modo.

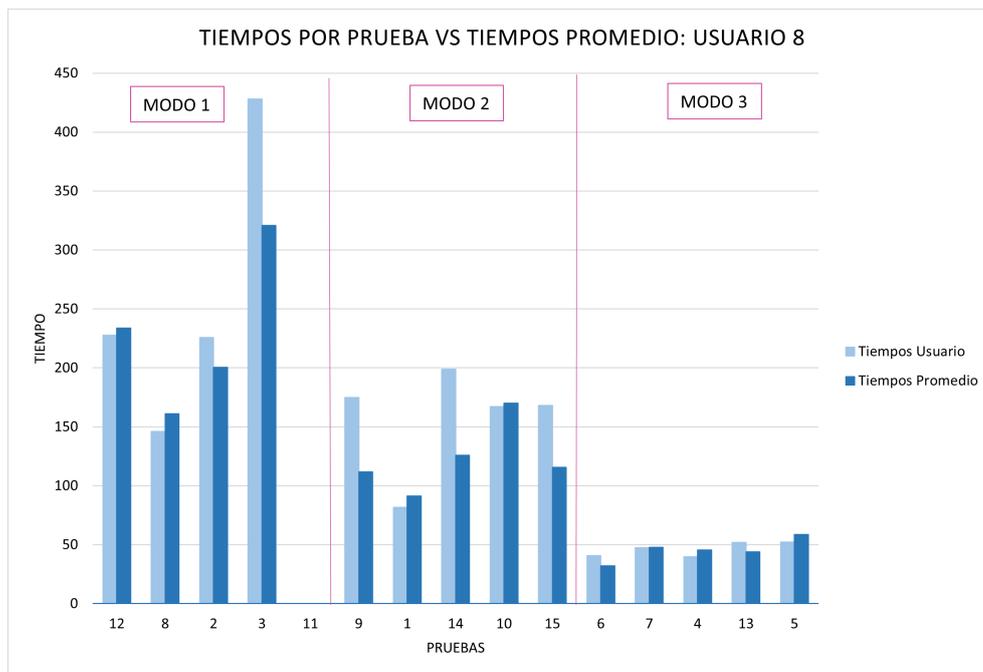


Figura 7.11: Resultados de los tiempos de las pruebas del usuario 8 con respecto al promedio.

Para este usuario, destacar las siguientes observaciones respecto a los datos obtenidos y anotaciones tomadas durante las pruebas:

- Respecto al modo 1 puede comprobarse que la prueba 11 no se ha realizado correctamente, pues en este caso el objeto se giró por la acción del robot y se colocó en una posición imposible de agarrar por la pinza. El resto de pruebas se han realizando correctamente, destacar la prueba 3 en la que el usuario ha tardado un tiempo en realizarla bastante superior al promedio, esto es debido a que el usuario presenta dificultades para agarrar este objeto, por no posicionarse en el lugar correcto para ello.
- En el modo 2 todas las pruebas se han realizado correctamente con tiempos similares a los promedios.
- En el modo 3 ocurre de manera similar al modo 2, ya que todas las pruebas se han realizado correctamente con parámetros similares a los promedios.

7.3.2. Análisis a nivel de modo

En este apartado se va a realizar una comparación entre los resultados obtenidos para observar y comparar los parámetros de cada uno de los modos. Se muestra en la Tabla 7.9 los datos promedio obtenidos de los distintos usuarios en los diferentes modos. Por otra parte, se muestra en la Figura 7.12 los tiempos promedio de los usuarios distinguidos por modo y objeto.

OBJETO	TIEMPO (s) EN MODO 1	TIEMPO (s) EN MODO 2	TIEMPO (s) EN MODO 3
Taza	200.49	125.97	45.70
Manzana	320.69	170.18	44.05
Botella	233.65	111.82	32.15
Refresco	175.43	115.66	58.60
Croissant	161.09	91.52	47.71

Tabla 7.9: Datos promedios obtenidos por modos

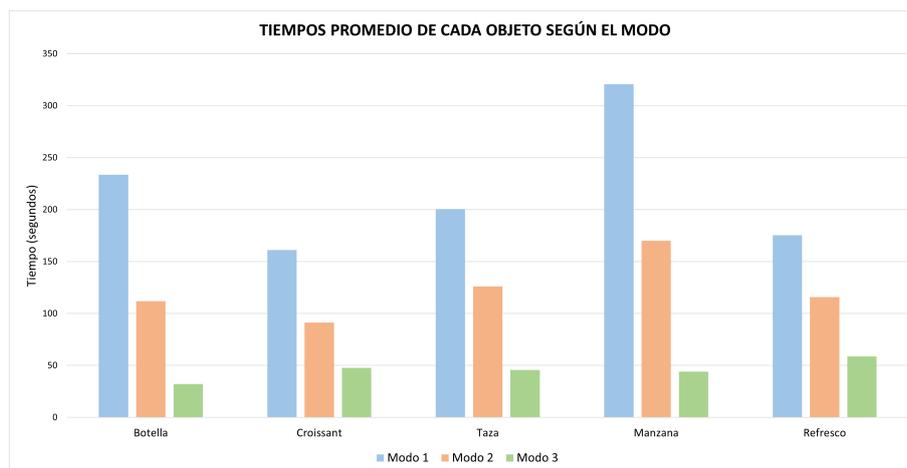


Figura 7.12: Resultados de los tiempos promedio en cada objeto para cada modo

Una vez obtenidos los datos de manera gráfica tal y cómo se ha visto en la Figura 7.12 podemos observar los siguientes aspectos:

- En todos los casos el modo 1 se ha desarrollado en un tiempo mayor al resto de casos, seguido del modo 2 y por último el más rápido el modo 3.
- El objeto que requiere de un mayor tiempo de prueba es la manzana para los modos 1 y 2, este objeto es a su vez el que se encuentra más alejado de la caja. Para el modo 3 el objeto que requiere de más tiempo es el refresco, el cual se encuentra en última posición en el menú de la interfaz.

- El objeto que requiere un menor tiempo de prueba es el croissant para los modos 1 y 2, y la botella para el tercer modo.

7.3.3. Análisis global

En este apartado se va a realizar un análisis global de las pruebas realizadas analizando algunos de los parámetros que se han tomado en las pruebas, distintos a los vistos anteriormente.

En primer lugar, se ha representado en la Figura 7.13 la tasa de acierto de las pruebas siendo el total las 120 pruebas realizadas, 15 de ellas por cada uno de los 8 usuarios. En esta se observa que la tasa de acierto es del 92%, lo que indica un éxito general en las pruebas, ya que la mayoría han sido completadas de manera satisfactoria frente a solo un 8% en las que se han cometido errores.

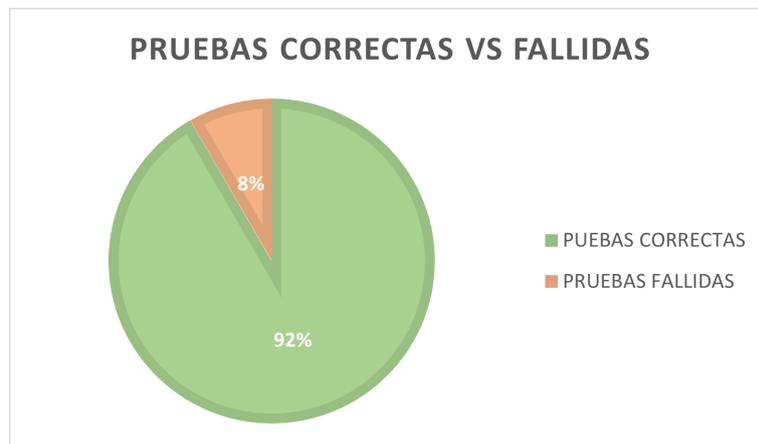


Figura 7.13: Tasa de acierto respecto a todas las pruebas realizadas

Por otra parte, en la Tabla 7.10 se muestran los valores de tiempo total de ensayo, este tiempo engloba desde el inicio del ensayo hasta que se han cumplimentado las pruebas, incluyendo todas ellas, incluso las fallidas, y los tiempos de preparación entre pruebas. El tiempo promedio de estas pruebas es 51'42", este factor nos indica la duración estimada de las pruebas.

7.3.4. Formularios

Finalmente, se va a hacer un análisis del formulario contestado por los usuarios tras realizar todas las pruebas que conforman el ensayo. Este formulario se ha realizado a través de la plataforma Google Forms, la cual permite completar el cuestionario de manera online y graficar estos resultados de manera automática. Los resultados obtenidos así como las preguntas realizadas vienen contemplados en el Anexo C.

Usuario	Tiempo total ensayo
Usuario 1	58'01"
Usuario 2	45'50"
Usuario 3	58'56"
Usuario 4	50'04"
Usuario 5	50'45"
Usuario 6	42'07"
Usuario 7	51'00"
Usuario 8	57'00"

Tabla 7.10: Tiempo total de duración de las pruebas

En primer lugar, en cuanto a las preguntas generales, destacar que la edad media de los participantes es de 25 años, todos ellos realizan ejercicio físico de manera usual u ocasional. En cuanto a la mano dominante destacar que el 75 % de los usuarios son diestros, mientras que el 25% son ambidiestros. Por otra parte, el 25% de los usuarios ha realizado previamente estudios que requieren de control mioeléctrico o entornos virtuales, y todos ellos poseen estudios universitarios, estos dos hechos facilitan el aprendizaje del control en el entorno creado.

En segundo lugar, se han realizado las preguntas relacionadas con el control mioeléctrico, la discusión de las mismas se detalla en el TFM sobre control mioeléctrico en colaboración con este [17].

En cuanto a las preguntas relacionadas con el entorno virtual, el 62.5% de los usuarios dice de haber realizado estudios previos en cuánto a entornos virtuales se refiere. Por otro lado, el 87.5% de los usuarios considera que habría sido interesante el uso de gafas de realidad virtual, y el 50% cree que hubiese aportado algo al proyecto el uso de estas gafas. En cuanto al seguimiento de los movimientos en la interfaz el 87.5% ha votado como cómodo, este mismo porcentaje ha dicho sentirse inmerso en la interfaz. Los cambios de estados por cocontracción han presentado un grado de dificultad promedio de 4.5 sobre 10. El 87.5% de los usuarios considera que el robot debería tener una velocidad mayor.

En cuanto a la evaluación de modos, para todos los usuarios el modo más sencillo ha sido el modo 3, al contrario del modo 1 que ha sido evaluado por el 87.5% como el más complejo. Valorando el modo 1 con una dificultad promedio de 6.75, el modo 2 con 3.75 y el modo 3 con 2.38. En el modo 1 el objeto más difícil de manipular ha sido la botella con el 62.5%, y como más sencillo el croissant y la manzana con el 37.5% cada uno. Por otra parte, todos los usuarios consideran la navegación del modo 2 como ventajosa respecto al modo 1, además de una cómoda navegación por la interfaz del modo 3. El 75% de los usuarios considera que el modo 2 es el mejor para aplicaciones de la vida diaria.

Los usuarios evaluaron con un promedio de 9.25 sobre 10 su experiencia global en este estudio. Además, han descrito el grado de dificultad del entrenamiento con un 4.38 sobre 10 de promedio. Todos los usuarios consideran este método como un buen método de control de

tecnologías asistenciales, y el 75% lo consideran un sistema fiable y seguro.

Finalmente, se ha preguntado a los usuarios posibles mejoras del sistema. Algunas de ellas contemplan el cambio de la disposición de la máquina de estados de Matlab que controlar el eje en el que nos desplazamos. Además, de mantener el movimiento del robot unos segundos con una misma acción de control. También, se comenta la posibilidad de añadir un control proporcional para que el usuario tenga posibilidad de regular la velocidad del brazo robótico. Otra propuesta interesante es la de considerar umbrales de desactivación que sean menores a los de activación, de manera que el usuario pueda mantener un movimiento sin necesitar de estar manteniendo todo el tiempo el umbral alto. Otra posible mejora es la de añadir nuevos nodos para añadir más grados de libertad del robot como el giro de la pinza.

7.4. Conclusiones de los resultados

Una vez analizadas todas las pruebas y cada uno de los datos que se han evaluado durante las mismas, se va a proceder a enumerar las conclusiones que se han extraído a partir de los resultados obtenidos.

- Se han comprobado las diferencias entre los distintos modos, cada uno de ellos se ha programado para controlar el brazo robótico de una manera diferente pero para conseguir realizar las mismas funciones, la de agarrar un objeto y llevarlo a un destino determinado. Destacar, que se ha percibido como el uso de un modo u otro ha provocado que los usuarios tengan una experiencia distinta para un mismo objeto, esto ha sido reflejado en los tiempos que se han utilizado, así como en los resultados de los formularios. Concluyendo que la mayoría de los usuarios consideran que el modo 1 es el más complicado de controlar y el modo 3 el más sencillo. Coincidiendo estos datos con los parámetros de tiempos, en los cuales el modo con un tiempo mayor es el modo 1, y con un tiempo menor el modo 3.

Sin embargo, para la mayoría de usuarios el control realizado del modo 2 es el que aplicarían en actividades de la vida diaria, a pesar de no ser el modo más rápido, siendo este el modo 3. El modo 2 ofrece varios grados de libertad, aunque sin presentar la libertad del modo 1, y las restricciones del modo 3.

- Las diferencias entre los objetos sólo se han mostrado en el modo 1, ya que en el resto de modos el agarre se realiza de manera automática. Entre estas diferencias se observó en las pruebas una dificultad superior del agarre de la botella, a pesar de no ser esta la que mayor tiempo promedio posee, siendo este la manzana debido a que los tiempos del usuario 3 son mucho mayores al promedio, de manera que este se ve modificado. Respecto al objeto más sencillo de coger en las pruebas se ha observado que es el croissant, este hecho coincide con los tiempos y con los formularios.
 - Por otra parte, a pesar de que el ensayo esté formado por 15 pruebas, sólo uno de los usuarios ha presentado una fatiga muscular tal que le impida realizar algunas de las
-

pruebas. Pues el resto de usuarios, aunque con algunos fallos, han logrado completar todas ellas.

- Por último, concluir que las pruebas han sido realizadas con éxito en la mayoría de los casos tal y cómo se ha mostrado en la Figura 7.13. Pues aunque algunos de los usuarios han tenido unos tiempos elevados comparados con los promedios, se ha cumplido la finalidad de la prueba, que era el de finalizar correctamente estas actividades.

8. Conclusiones y posibles mejoras

8.1. Conclusiones

Tras elaborar este proyecto podemos concluir con que los objetivos propuestos han sido cumplidos. Además, durante la elaboración del proyecto se han observado posibles mejoras del sistema que se enumerarán en el siguiente apartado.

En primer lugar, se ha diseñado en Unity un entorno virtual, este pretendía que fuese un entorno familiar para los usuarios con el fin de dotar al sujeto de la sensación de inmersión, por ello se han utilizado elementos de la vida cotidiana, tales como los de una cocina. Este objetivo ha sido alcanzado ya que según las encuestas proporcionadas el 87.5% de los usuarios se han sentido inmersos en la interfaz.

Esta interfaz se ha completado con un brazo robótico asistencial en este caso ha sido el robot UR3, el cual es un robot colaborativo y cumple con los requisitos de un robot asistencial. Este nos ha permitido realizar las tareas de agarre y manipulación, gracias a la implementación del cuerpo articulado que incluye cada una de las articulaciones necesarias y a las físicas implementadas. Esto nos permite el movimiento cartesiano en los tres ejes del espacio, y la posibilidad de agarrar objetos.

Una vez obtenida esta interfaz se ha conseguido otro de los objetivos propuestos gracias a la programación de tres modos que ofrecen diferentes grados de libertad sobre el robot para la realización de las mismas tareas. Estos modos se han programado de manera que cada uno posea un grado de autonomía diferente, consiguiendo con todos ellos realizar las actividades de manipulación y agarre de los elementos de la interfaz, tal y cómo se propuso en los objetivos.

Por otra parte, se ha conseguido que cada uno de estos modos funcione a partir de los mensajes recibidos de Matlab, de esta manera dependiendo del mensaje obtenido el brazo robótico ejecutará unas acciones u otras.

Además, se ha implementado una retroalimentación, aparte de la visual ofrecida por el seguimiento de movimientos del brazo robótico en la interfaz, la cual proporciona información sobre el estado de las pinzas en cada momento.

Finalmente, y tras cumplir los objetivos que implican el desarrollo del software se han llevado a cabo pruebas de usabilidad en las que han intervenido ocho usuarios diferentes, gracias a estas pruebas ha podido realizarse una evaluación del funcionamiento del proyecto en conjunto con el estudio realizado de las señales de control mioeléctrico [17]. Esta evaluación ha sido desarrollada a partir del análisis de los datos tanto cuantitativos como cualitativos

tomados en las pruebas.

Gracias a estos datos, hemos podido analizar las diferencias existentes entre los distintos modos, concluyendo con que el modo 2 es el que la mayoría de usuarios (75%) preferirían para su uso en la vida cotidiana. No obstante, se ha observado como el control para cada usuario es diferente, pues el 25% preferirían usar el modo 3. Por lo tanto, este sistema podría proporcionar que un usuario en concreto pueda realizar la selección del modo óptimo que prefiere para implementar en su vida diaria.

Se puede concluir este trabajo con que el proyecto finalizado ha sido satisfactorio, ya que la tasa de aciertos ha mostrado cómo las pruebas se han completado en su mayoría, lo que indica que es un sistema que cumple su función principal. Además, se cumple el objetivo final, el cuál es enseñar a los usuarios a realizar el control del brazo robótico mediante las señales electromiográficas. Ya que se han observado diferencias entre el control del brazo robótico que realizaban en el entrenamiento inicial respecto a las pruebas finalmente realizadas.

8.2. Posibles mejoras

A pesar de alcanzar todos y cada uno de los objetivos planteados en el inicio de este proyecto existen posibles mejoras que han sido planteadas durante la realización de las pruebas al observar distintos aspectos durante la funcionamiento. Estas mejoras ofrecen nuevas funcionalidades al sistema o aportan una mayor comodidad al usuario. Se exponen a continuación cada una de las posibles mejoras a desarrollar en un futuro:

- Durante las pruebas se comprobó que la velocidad del robot era demasiado lenta respecto a los movimientos que no se realizaban de manera automática, este hecho ha podido corroborarse con la realización de las encuestas en las que el 87.5% de los usuarios han considerado que la velocidad del robot debería ser más rápida. Por ello se propone como mejora aumentar esta velocidad, aunque no de manera excesiva porque esto provocaría un difícil control para colocarse en la posición correcta para agarrar el objeto,
 - Otra de las mejoras que se propone es el uso de gafas de realidad virtual, gracias a las cuales se aumentaría la sensación de inmersión por parte del usuario en el entorno, además, posibilitaría al usuario el cambio de vista de la escena lo que facilitaría en la función de coger un objeto debido a una mejor visión del mismo.
 - Por último, destacar la posibilidad en líneas futuras de enviar los movimientos del brazo robótico del entorno virtual a un brazo robótico real, el cuál permita realizar las actividades de asistencia destacadas en este trabajo, como son el agarre y manipulación de objetos en un caso real.
-

Bibliografía

- [1] RobotXperience. <https://robots.nu/es/robot/care-o-bot-4>, 2022. Accedido en junio de 2022.
- [2] “ANA and Panasonic Partner to Test Self-Driving electric Wheelchairs at Narita Airport.” <https://news.panasonic.com/global/topics/2019/68529.html>, Mayo 2019. Accedido en junio de 2022.
- [3] Ottobock. <https://www.ottobock.com/es-es/product/8E500>. Accedido en junio de 2022.
- [4] Myomo. <https://myomo.com/>, 2022. Accedido en junio de 2022.
- [5] “Kinova robotics.” <https://assistive.kinovarobotics.com/product/jaco-robotic-arm>, 2022. Accedido en junio de 2022.
- [6] “Universal robots.” <https://www.universal-robots.com/es/productos/robot-ur3/>, 2022. Accedido en junio de 2022.
- [7] MANUS. <https://www.manus-meta.com/products/prime-x>. Accedido en junio de 2022.
- [8] Z. Zhang, “Microsoft kinect sensor and its effect,” *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012.
- [9] Meta. https://store.facebook.com/es/es/quest/products/quest-2/?utm_source=www.oculus.com&utm_medium=oculusredirect, 2022. Accedido en junio de 2022.
- [10] Haptx. <https://haptx.com/technology/>. Accedido en junio de 2022.
- [11] A. Gifreu, “La realidad virtual. como afectará a los webdocs. parte 2.” <https://blog.rtve.es/webdocs/2014/12/la-realidad-virtual-como-afectar%C3%A1-a-los-webdocs-parte-5.html>, Diciembre 2014. Accedido en junio de 2022.
- [12] UOC, “Introducción a la RV.” http://cv.uoc.edu/annotation/8ebfc11d61d9fb2feed41b629265e634/463715/PID_00150738/modul_1.html. Accedido en junio de 2022.
- [13] I. Cáceres Rodríguez, E. Pérez Hernández, I. Tejado Balsera, B. M. Vinagre Jara, M. P. Merchán García, and S. Salamanca Miño, “Realidad virtual como tecnología asistencial a deportistas con movilidad reducida en silla de ruedas,” *Actas de las XXXIX Jornadas de Automática, Badajoz, 5-7 de Septiembre de 2018*, 2018.

-
- [14] I. Phelan, M. Arden, M. Matsangidou, A. Carrion-Plaza, and S. Lindley, “Designing a virtual reality myoelectric prosthesis training system for amputees,” *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–7, 2021.
- [15] L. N. S. Andreasen, L. Lindhardt, R. Lontis, M. Gaihede, and B. Bentsen, “Wireless intraoral tongue control of an assistive robotic arm for individuals with tetraplegia,” *Journal of neuroengineering and rehabilitation*, vol. 14, no. 1, pp. 1–8, 2017.
- [16] Unity, “Manual de Unity.” <https://docs.unity3d.com/es/530/Manual/UnityManual.html>. Accedido en junio de 2022.
- [17] C. Polo, “Evaluación y validación de control mioeléctrico en un entorno virtual de robótica asistencial,” *Trabajo Fin de Máster*, julio 2022.
- [18] ATIA, “What is AT?.” <https://www.atia.org/home/at-resources/what-is-at/#:~:text=online%20ATIA%20webinar%3F-,What%20is%20assistive%20technology%3F,of%20cardboard%20or%20fuzzy%20felt.>, 2022. Accedido en junio de 2022.
- [19] OMS, “Tecnología de asistencia.” <https://www.who.int/es/news-room/fact-sheets/detail/assistive-technology>, Mayo 2018. Accedido en junio de 2022.
- [20] C. N. Perdomo, “Tecnología de asistencia en terapia ocupacional: Uso del modelo HAAT,” vol. 16, Noviembre 2019.
- [21] C. Gatchalian, “Assistive technologies in the 21st century,” *Technology and the Curriculum: Summer 2019*, 2019.
- [22] A. Mancisidor, “Control avanzado para robótica asistencial y sanitaria,” *Universidad del País Vasco*, Diciembre 2017.
- [23] M. A. Chávez, F. Rodríguez, and A. Baradica, “Exoesqueletos para potenciar las capacidades humanas y apoyar la rehabilitación,” *Revista Ingeniería Biomédica*, vol. 4, pp. 63–73, Mayo 2010.
- [24] P. Cipresso, I. A. Chicchi, M. Alcañiz, and G. Riva, “The past, present, and future of virtual and augmented reality research: A network and cluster analysis of the literature,” *Frontiers in psychology*, p. 2086, Noviembre 2018.
- [25] “Introducción a la realidad virtual,” *Universidad Politécnica de Cataluña*.
- [26] V. R. Society, “History of Virtual Reality.” <https://www.vrs.org.uk/virtual-reality/history.html>, 2017. Accedido en junio de 2022.
- [27] P. Croft, “Realidad virtual: origen, actualidad y futuro,” *MeriStation*, enero 2018.
- [28] F. Marotta, J. A. Montes, and G. A. Addati, “Simulaciones con realidad inmersiva, semi inmersiva y no inmersiva,” *Serie Documentos de Trabajo*, Julio 2020.
- [29] K. E. Laver, B. Lange, S. George, J. E. Deutsch, G. Saposnik, and M. Crotty, “Virtual reality for stroke rehabilitation,” *Cochrane database of systematic reviews*, no. 11, 2017.
-

- [30] NORAXON, “myoRESEARCH 3.16 User Manual,” 2018.
 - [31] Matlab. <https://es.mathworks.com/products/matlab.html>. Accedido en junio de 2022.
 - [32] GitHub, “articulations-robot-demo.” <https://github.com/Unity-Technologies/articulations-robot-demo>. Accedido en enero de 2022.
-

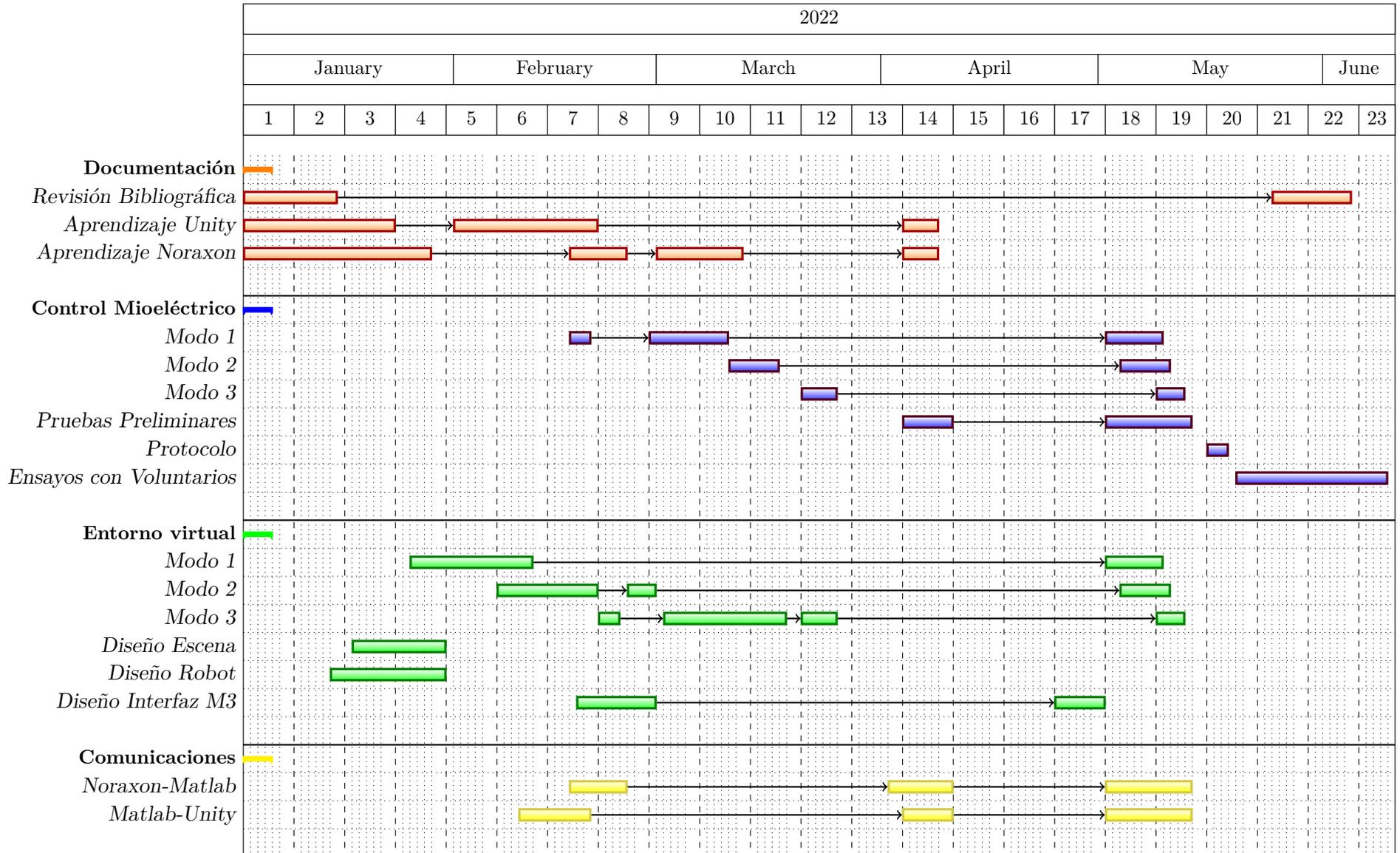
Lista de Acrónimos y Abreviaturas

ATIA	Asociación de la Industria de las Tecnologías de Asistencia.
EMG	Electromiografía.
HMD	Head Mounted Display.
HTTP	Hypertext Transfer Protocol.
IP	Protocolo de Internet.
MiniDTS	Mini Direct Transmission System.
MR3	Myo Research 3.
RV	Realidad Virtual.
TCP	Protocolo de Control de Transmisión.
TFM	Trabajo Final de Máster.

A. Anexo I: Diagrama de Gantt

En este Anexo se muestra el Diagrama de Gantt desarrollado al inicio del proyecto, el cuál se ha ido actualizando en el transcurso del trabajo. Este ha sido dividido en cuatro secciones diferentes:

- Documentación: En esta fase del proyecto se ha realizado la revisión bibliográfica necesaria para alcanzar los conceptos necesarios, así como el aprendizaje de los software Unity y Noraxon.
- Control mioeléctrico: En este se engloban los modos de desarrollados en Matlab, así como las pruebas preliminares realizadas para ajustar algunos aspectos de cara a las pruebas finales. Además, del establecimiento de un protocolo para la realización de las pruebas y los ensayos con los voluntarios.
- Entorno virtual: En este se incluyen el diseño tanto de la escena como del robot, la programación de los modos 1, 2 y 3 y el diseño de la interfaz de este último modo.
- Comunicaciones: engloba las comunicaciones a realizar entre Noraxon- Matlab y Matlab-Unity.



B. Anexo II: Resultados de las pruebas

1. Datos modo 1: Movimiento libre en tres ejes y grasping manual

TAZA					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	127.84	29	33	19	OK
Usuario 2	187.16	80	130	23	OK
Usuario 3	289.00	175	285	27	OK
Usuario 4	202.98	162	266	15	OK
Usuario 5	141.10	40	42	23	OK
Usuario 6	185.08	148	248	15	OK
Usuario 7	245.14	112	285	35	OK
Usuario 8	225.62	189	293	31	OK
TOTAL	1603.92	935	1582	188	-
PROMEDIO	200.49	116.88	195.75	23.5	-

Tabla B.1: Datos obtenidos para el objeto taza en el modo 1

MANZANA					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	183.70	50	52	19	OK
Usuario 2	111.03	98	125	11	OK
Usuario 3	726.92	450	490	87	OK
Usuario 4	325.00	261	314	23	OK
Usuario 5	310.00	234	296	31	OK
Usuario 6	257.15	276	313	31	OK
Usuario 7	223.60	228	312	27	OK
Usuario 8	428.13	379	274	59	OK
TOTAL	2565.53	1976	2176	288	-
PROMEDIO	320.69	247.00	272.00	36.00	-

Tabla B.2: Datos obtenidos para el objeto manzana en el modo 1

BOTELLA DE AGUA					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	357.17	73	88	43	OK
Usuario 2	NO APLICA	NO APLICA	NO APLICA	NO APLICA	NO OK
Usuario 3	NO APLICA	NO APLICA	NO APLICA	NO APLICA	NO OK
Usuario 4	228.18	210	282	15	OK
Usuario 5	156.17	43	59	19	OK
Usuario 6	199.07	252	246	19	OK
Usuario 7	NO APLICA	NO APLICA	NO APLICA	NO APLICA	NO OK
Usuario 8	227.67	231	260	15	OK
TOTAL	1168.26	809	935	111	-
PROMEDIO	233.65	161.8	187.00	22.20	-

Tabla B.3: Datos obtenidos para el objeto botella en el modo 1

REFRESCO					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	349.37	68	85	51	OK
Usuario 2	100.87	57	100	15	OK
Usuario 3	148.43	90	203	11	OK
Usuario 4	141.58	104	207	15	OK
Usuario 5	176.4	118	191	19	OK
Usuario 6	147	98	190	11	OK
Usuario 7	164.36	90	214	20	OK
Usuario 8	NO APLICA	NO APLICA	NO APLICA	NO APLICA	NO OK
TOTAL	1228.01	625	1190	142	-
PROMEDIO	175.43	89.29	170.00	20.29	-

Tabla B.4: Datos obtenidos para el objeto refresco en el modo 1

CROISSANT					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	155.89	34	40	23	OK
Usuario 2	158.83	139	132	27	OK
Usuario 3	162	86	461	23	OK
Usuario 4	251.73	154	207	23	OK
Usuario 5	123.1	83	131	15	OK
Usuario 6	130	97	168	11	OK
Usuario 7	NO APLICA	NO APLICA	NO APLICA	NO APLICA	NO OK
Usuario 8	146.06	114	179	19	OK
TOTAL	1227.61	707	1318	141	-
PROMEDIO	161.09	101.00	188.29	20.14	-

Tabla B.5: Datos obtenidos para el objeto croissant en el modo 1

2. Datos modo 2: Movimiento libre en dos ejes y grasping automático

TAZA					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	55.3	15	17	7	OK
Usuario 2	64.73	37	50	4	OK
Usuario 3	NO APLICA	NO APLICA	NO APLICA	NO APLICA	NO OK
Usuario 4	NO APLICA	NO APLICA	NO APLICA	NO APLICA	NO OK
Usuario 5	141.63	95	147	10	OK
Usuario 6	169.25	96	155	11	OK
Usuario 7	NO APLICA	NO APLICA	NO APLICA	NO APLICA	NO OK
Usuario 8	198.96	108	155	2	OK
TOTAL	629.87	351	524	53	-
PROMEDIO	125.974	70.2	104.8	10.6	-

Tabla B.6: Datos obtenidos para el objeto taza en el modo 2

MANZANA					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	111.22	34	48	7	OK
Usuario 2	60.32	42	51	7	OK
Usuario 3	189.88	140	171	11	OK
Usuario 4	235.52	187	237	17	OK
Usuario 5	179.72	155	179	7	OK
Usuario 6	190.94	139	158	14	OK
Usuario 7	226.85	126	246	31	OK
Usuario 8	167.02	146	211	4	OK
TOTAL	1361.47	969	1301	98	-
PROMEDIO	170.18	121.13	162.63	12.25	-

Tabla B.7: Datos obtenidos para el objeto manzana en el modo 2

BOTELLA					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	62.58	21	20	4	OK
Usuario 2	54.52	42	50	4	OK
Usuario 3	133.72	106	129	7	OK
Usuario 4	131.44	129	141	4	OK
Usuario 5	74.27	31	27	7	OK
Usuario 6	139.44	111	135	4	OK
Usuario 7	123.79	92	137	7	OK
Usuario 8	174.76	131	153	19	OK
TOTAL	894.52	663	792	56	-
PROMEDIO	111.815	82.875	99	7	-

Tabla B.8: Datos obtenidos para el objeto botella en el modo 2

REFRESCO					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	70.23	22	28	7	OK
Usuario 2	85.18	34	71	8	OK
Usuario 3	104.94	64	87	7	OK
Usuario 4	120.3	86	117	10	OK
Usuario 5	45.84	10	14	4	OK
Usuario 6	180	77	146	10	OK
Usuario 7	150.6	78	122	23	OK
Usuario 8	168.15	91	120	14	OK
TOTAL	925.24	462	705	83	-
PROMEDIO	115.655	57.75	88.13	10.38	-

Tabla B.9: Datos obtenidos para el objeto refresco en el modo 2

CROISSANT					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	116.47	18	18	14	OK
Usuario 2	42.10	19	32	4	OK
Usuario 3	83.14	30	77	4	OK
Usuario 4	124.31	29	98	13	OK
Usuario 5	62.77	54	6	7	OK
Usuario 6	99.64	25	89	7	OK
Usuario 7	119.94	15	122	32	OK
Usuario 8	81.65	26	77	10	OK
TOTAL	730.02	216	519	91	-
PROMEDIO	91.25	27	64.87	11.8	-

Tabla B.10: Datos obtenidos para el objeto croissant en el modo 2

3. Datos modo 3: Movimiento y grasping automático

TAZA					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	44.00	4	4	0	OK
Usuario 2	43.20	3	4	0	OK
Usuario 3	69.53	6	5	0	OK
Usuario 4	34.15	3	8	0	OK
Usuario 5	38.65	5	4	0	OK
Usuario 6	37.94	2	4	0	OK
Usuario 7	58.55	3	29	0	OK
Usuario 8	39.61	2	4	0	OK
TOTAL	364.84	27	62	0	-
PROMEDIO	45.60	3.38	7.75	0	-

Tabla B.11: Datos obtenidos para el objeto taza en el modo 3

MANZANA					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	42.89	2	6	0	OK
Usuario 2	52.08	3	30	0	OK
Usuario 3	43.54	2	6	0	OK
Usuario 4	40.57	2	19	0	OK
Usuario 5	45.64	5	11	0	OK
Usuario 6	38.47	3	7	0	OK
Usuario 7	37.15	3	7	0	OK
Usuario 8	52.02	3	11	0	OK
TOTAL	352.36	23	97	0	-
PROMEDIO	44.05	2.88	12.13	0	-

Tabla B.12: Datos obtenidos para el objeto manzana en el modo 3

BOTELLA DE AGUA					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	41.06	2	0	0	OK
Usuario 2	33.85	5	2	0	OK
Usuario 3	34.12	2	0	0	OK
Usuario 4	21.32	11	0	0	OK
Usuario 5	21.32	11	0	0	OK
Usuario 6	32.41	3	0	0	OK
Usuario 7	32.41	3	0	0	OK
Usuario 8	40.7	4	11	0	OK
TOTAL	257.19	41	13	0	-
PROMEDIO	32.15	5.13	1.63	0	-

Tabla B.13: Datos obtenidos para el objeto botella en el modo 3

REFRESCO					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	86.30	6	13	0	OK
Usuario 2	39.44	2	10	0	OK
Usuario 3	58.27	2	13	0	OK
Usuario 4	41.64	4	12	0	OK
Usuario 5	98.11	9	34	0	OK
Usuario 6	40.68	3	8	0	OK
Usuario 7	52.02	3	40	1	OK
Usuario 8	52.35	2	18	0	OK
TOTAL	468.81	31	148	1	-
PROMEDIO	58.60	3.88	18.50	18.50	-

Tabla B.14: Datos obtenidos para el objeto refresco en el modo 3

CROISSANT					
USUARIO	TIEMPO (s)	Nº FLEXIONES	Nº EXTENSIONES	Nº COCONTRACCIONES	FINALIZADO
Usuario 1	98.16	8	13	2	OK
Usuario 2	38.15	3	7	0	OK
Usuario 3	43.66	2	2	0	OK
Usuario 4	35.02	2	4	0	OK
Usuario 5	40.99	4	8	0	OK
Usuario 6	40.80	3	2	0	OK
Usuario 7	37.58	2	17	0	OK
Usuario 8	47.36	3	17	0	OK
TOTAL	381.72	27	70	2	-
PROMEDIO	47.71	3.38	8.75	0.25	-

Tabla B.15: Datos obtenidos para el objeto croissant en el modo 3

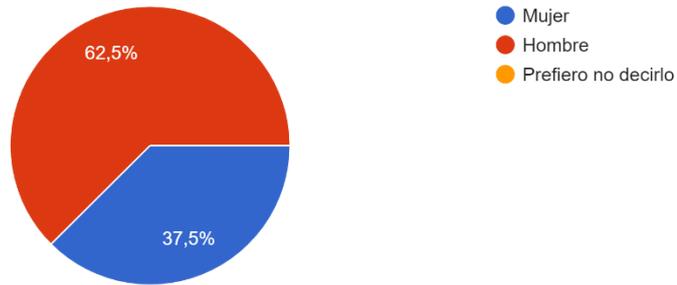
C. Anexo III: Resultados de las encuestas de los usuarios

En este anexo se muestran los resultados de las encuestas elaboradas. Esta encuesta está dividida en los siguientes apartados:

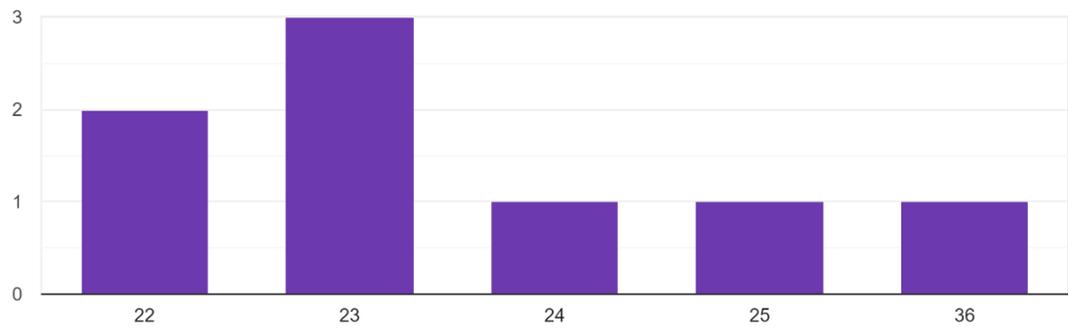
- Preguntas generales.
- Control mioeléctrico.
- Entorno virtual.
- Valoración final.

• **PRIMERA PARTE: PREGUNTAS GENERALES:**

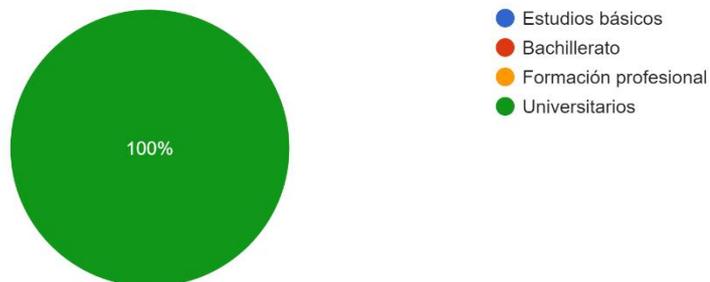
1-GÉNERO
8 respuestas



2-EDAD
8 respuestas

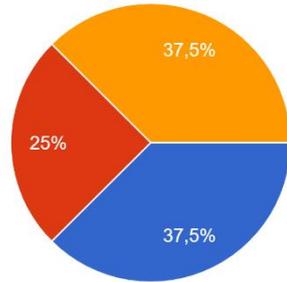


3-¿Cuál es tu nivel máximo de estudios alcanzado?
8 respuestas



4-¿Con cuánta frecuencia realizas ejercicio físico?

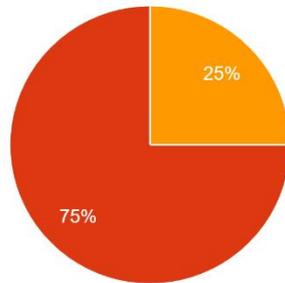
8 respuestas



- Diariamente
- Semanalmente
- Cuando me apetece
- Nunca

5-¿Qué lado de tu cuerpo utilizas preferentemente?

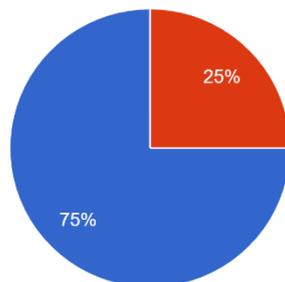
8 respuestas



- Soy zurdo/a
- Soy diestro/diestra
- Soy ambidiestro/ambidiestra

6-¿Has participado anteriormente en algún estudio relacionado con el control mioeléctrico o entorno virtual?

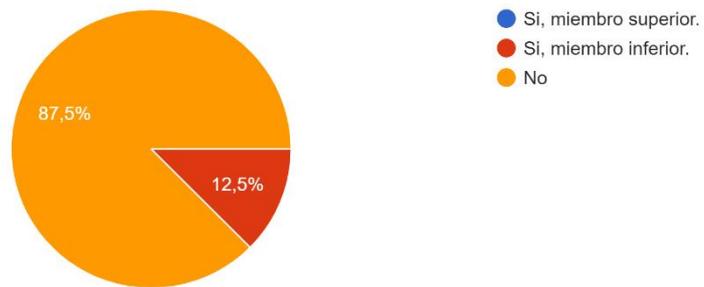
8 respuestas



- Sí
 - No
-

7-¿Conoces alguna persona que presente amputación del algún miembro del cuerpo?

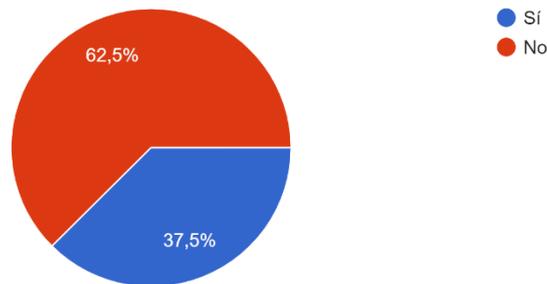
8 respuestas



• **SEGUNDA PARTE:** CONTROL MIOELÉCTRICO:

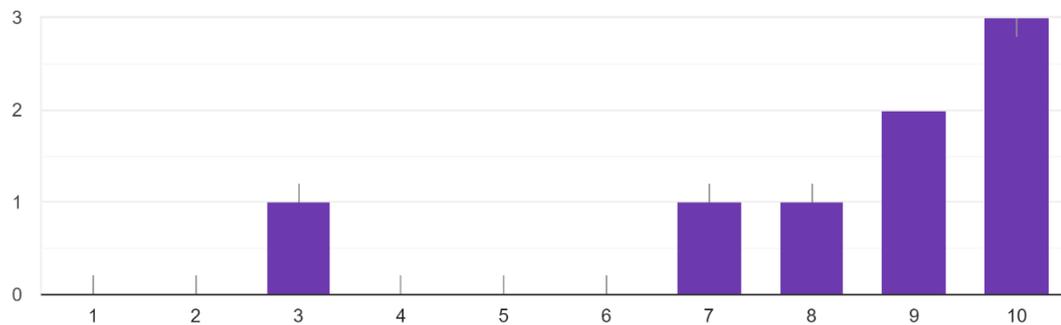
8-¿Esta prueba ha sido tu primera vez en el uso de sensores mioeléctricos?

8 respuestas



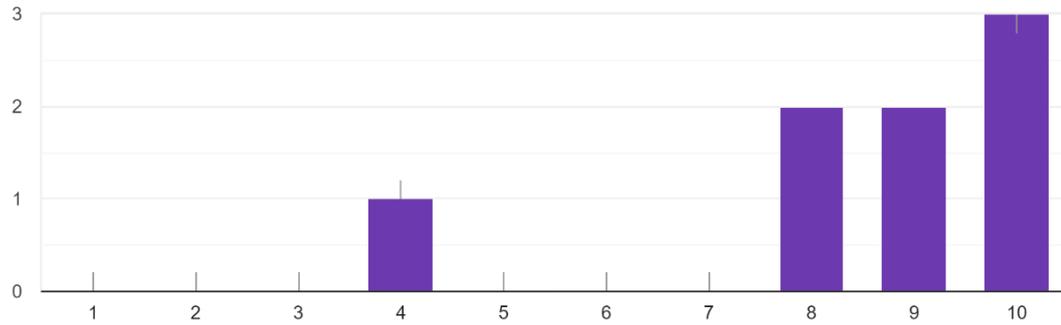
9-¿Del 1 al 10 cómo de cómodo te han resultado los sensores mioeléctricos utilizados?

8 respuestas



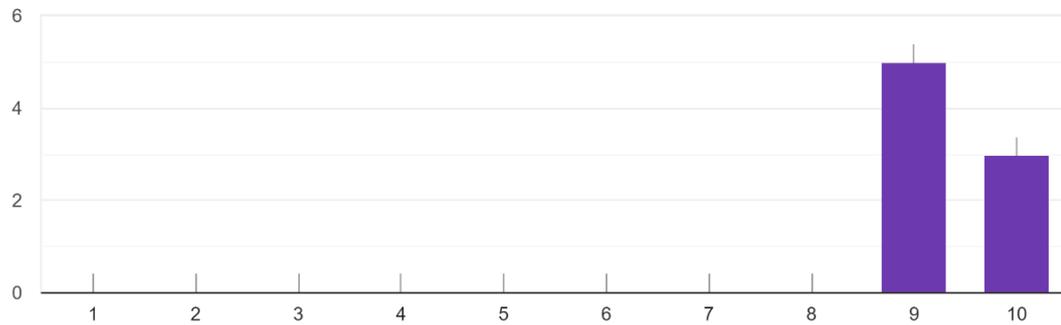
10- ¿Te ha resultado fácil el aprendizaje del control de los sensores?

8 respuestas



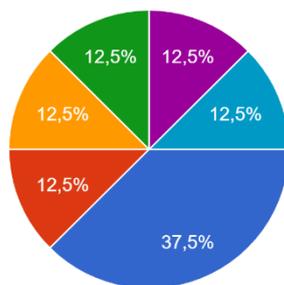
11- ¿Te ha resultado intuitiva la interfaz de control de Matlab?

8 respuestas



12- ¿Añadirías algún elemento adicional del feedback a los mostrados en la interfaz? En caso afirmativo indica tu opinión.

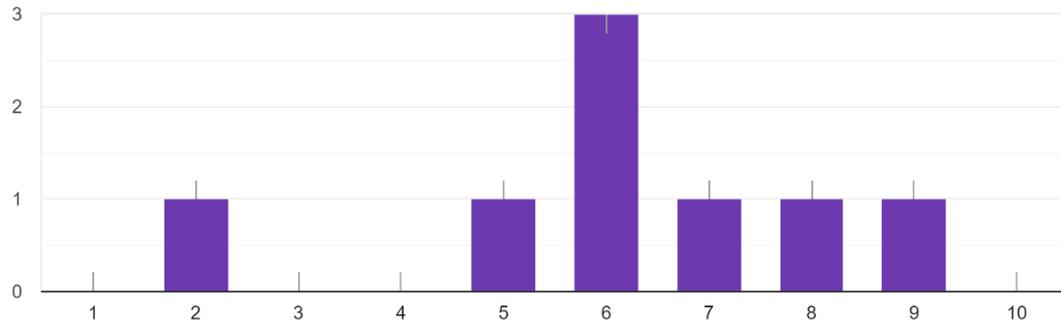
8 respuestas



- No
- Feedback auditivo o vibrotáctil cuando la pinza coge el objeto
- Quizás se podría añadir dónde está el umbral seleccionado para cada músculo...
- Algo que deje más claro si la extensión o la flexión cual cierra o abre pinzas
- Una imagen con la dirección de movimiento de la muñeca relacionada...
- en cuanto al feedback visual, aparte d...

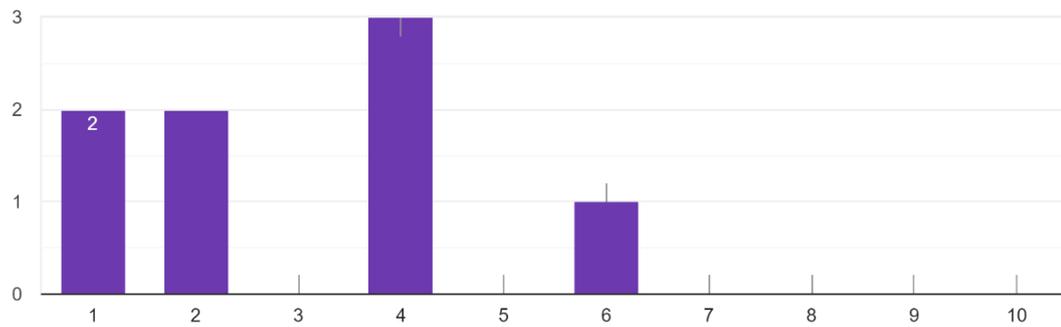
13-Indicar el grado de fatiga muscular que has sentido tras finalizar la prueba.

8 respuestas



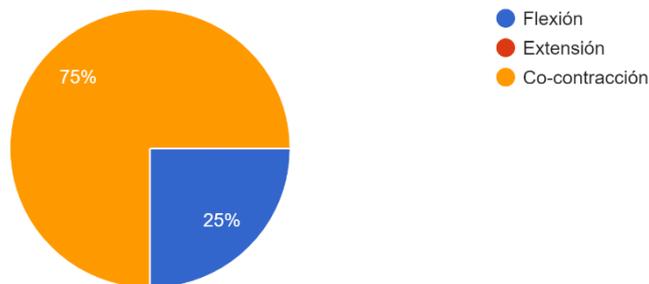
14-Indicar el grado de fatiga mental que has sentido tras finalizar la prueba.

8 respuestas



15¿Cuál es el movimiento que más difícil te ha resultado de realizar respecto a los umbrales?

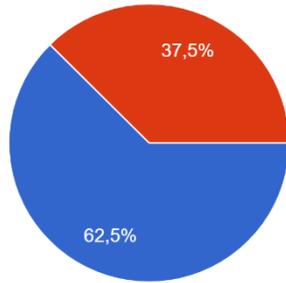
8 respuestas



• **TERCERA PARTE: ENTORNO VIRTUAL UNITY**

16- ¿Conoces el concepto de entorno virtual? ¿Has hecho alguna vez uso de sus herramientas (videojuegos, gafas virtuales...)?

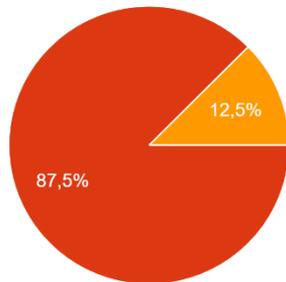
8 respuestas



- Si lo conozco y he utilizado sus herramientas.
- Si lo conozco pero no he hecho uso de sus herramientas.
- No lo conozco ni he utilizado ninguna de sus herramientas.

17- ¿Te hubiera parecido interesante el uso de gafas de realidad virtual en el proyecto?

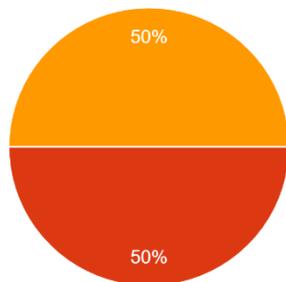
8 respuestas



- No
- Sí
- Tal vez

18- ¿Crees que hubiese aportado algo adicional el uso de las gafas virtuales?

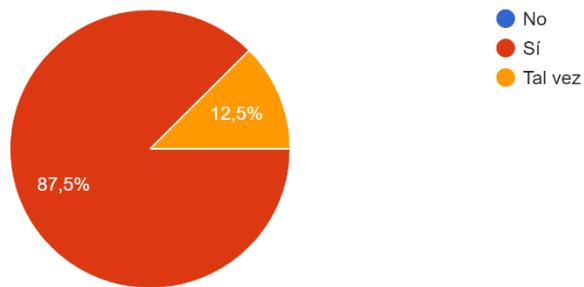
8 respuestas



- No
- Sí
- Tal vez

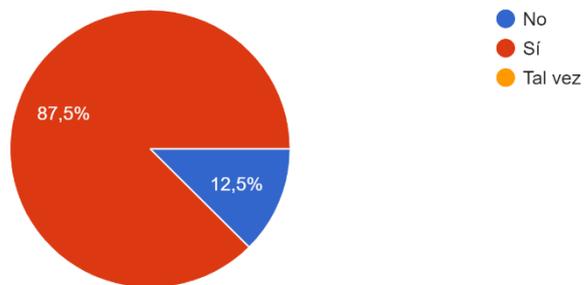
19-¿Te ha resultado cómodo el seguimiento de los movimientos del robot a través de la interfaz?

8 respuestas



20-¿Te has sentido inmerso en la interfaz ?

8 respuestas



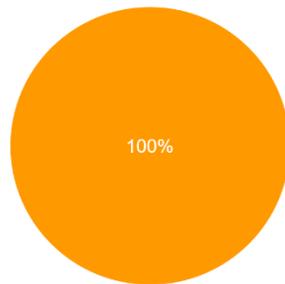
21-¿Cuál es el modo que más complejo te ha resultado de controlar?

8 respuestas



22-¿Cuál es el modo que más sencillo te ha resultado de controlar?

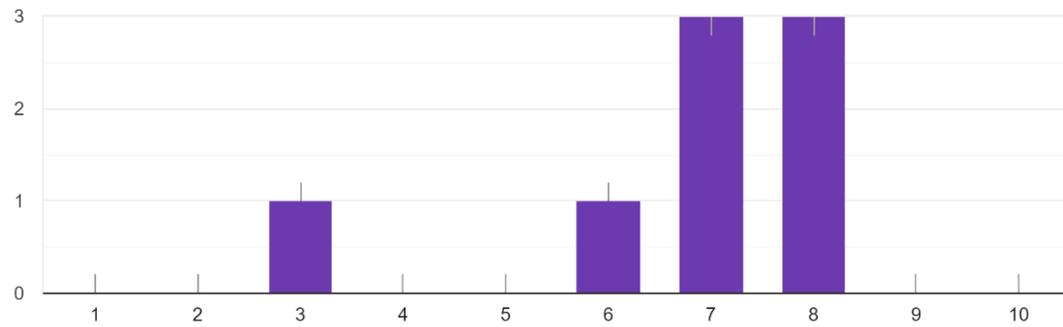
8 respuestas



- Modo 1: Libertad de movimiento.
- Modo 2: Objeto más cercano.
- Modo 3: Control por interfaz.

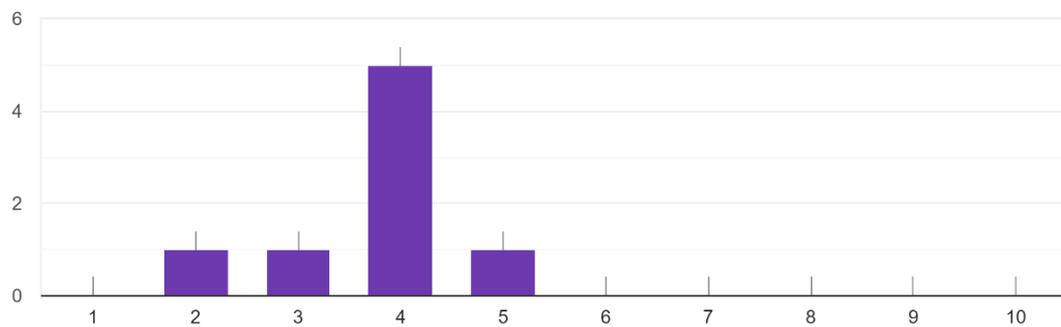
23-¿Con qué dificultad valorarías del 1 al 10 el Modo 1?

8 respuestas



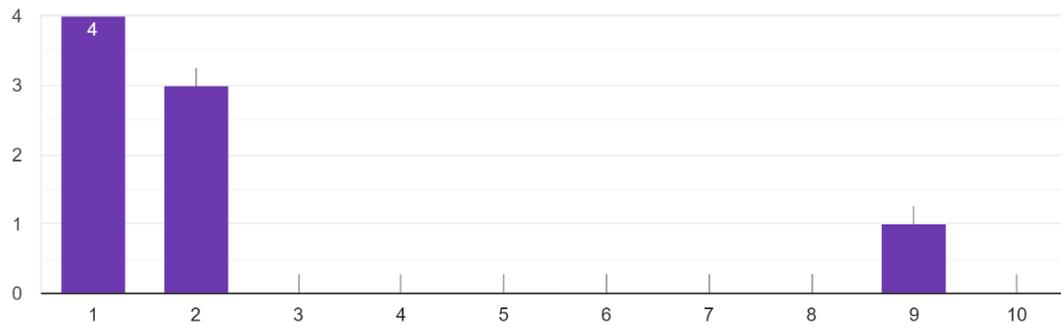
24-¿Con qué dificultad valorarías del 1 al 10 el Modo 2?

8 respuestas



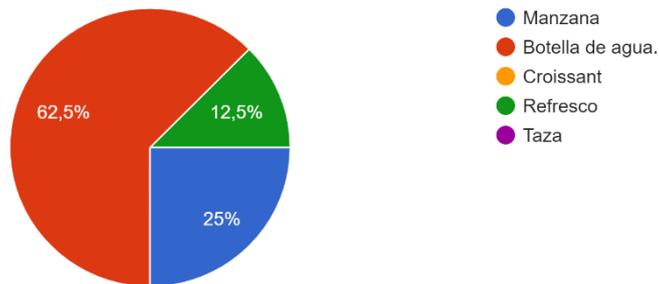
25-¿Con qué dificultad valorarías del 1 al 10 el Modo 3?

8 respuestas



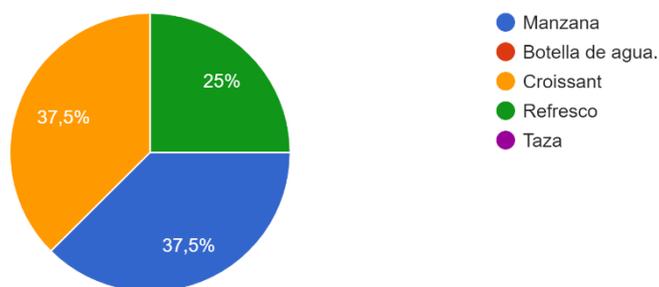
26-Respecto al modo 1 ¿Cuál es el elemento que te ha resultado más difícil manipular?

8 respuestas



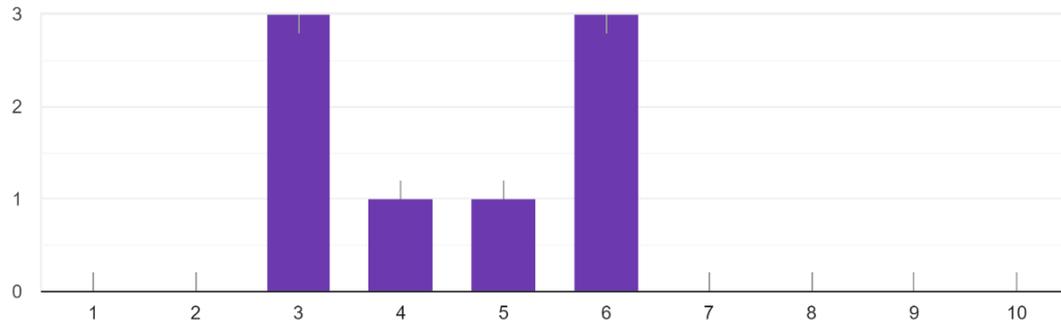
27-Respecto al modo 1 ¿Cuál es el elemento que te ha resultado más fácil manipular?

8 respuestas



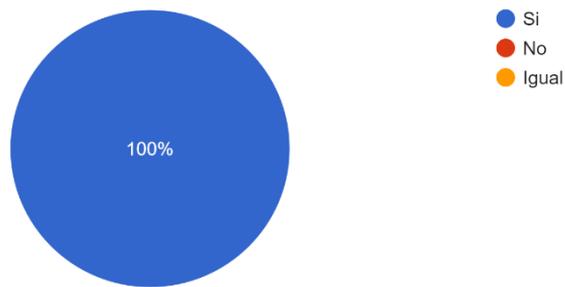
28-¿Qué grado de dificultad te ha supuesto los cambios de estado por co-contracción?

8 respuestas



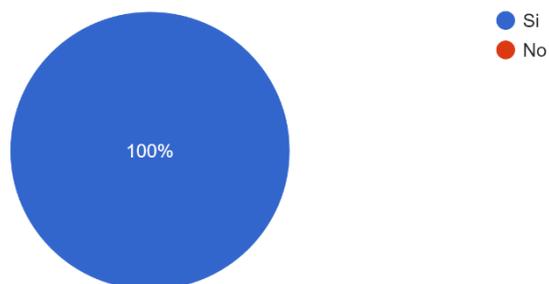
29-Respecto al modo 2 ¿Crees que es más ventajoso en cuanto a la navegación del modo 1?

8 respuestas



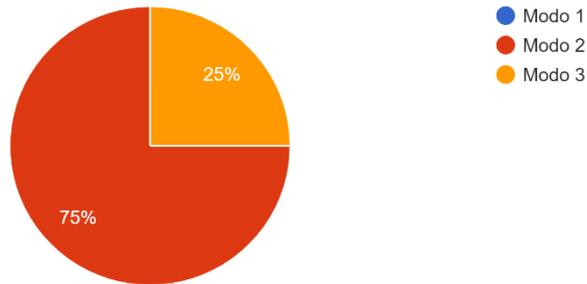
30-Respecto al modo 3 ¿Te ha resultado cómoda la navegación por la interfaz?

8 respuestas



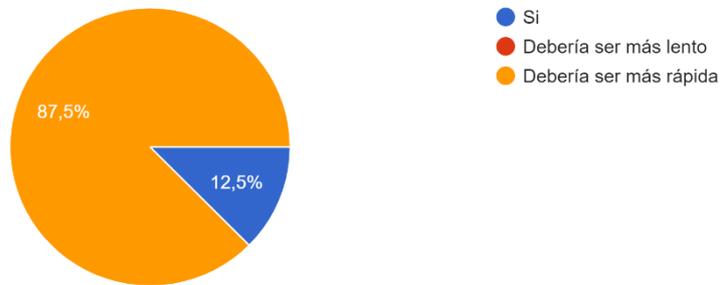
31-¿Qué método te parece mejor para utilizarlo en aplicaciones de la vida diaria?

8 respuestas



32-¿Crees que la velocidad del robot era adecuada?

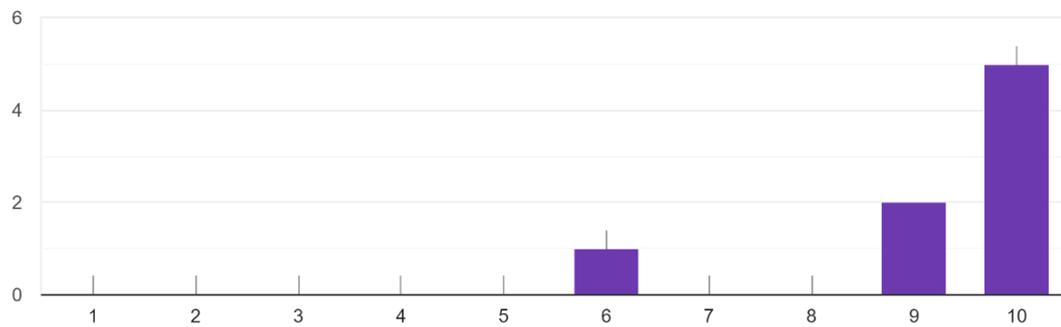
8 respuestas



• **CUARTA PARTE: VALORACIÓN FINAL**

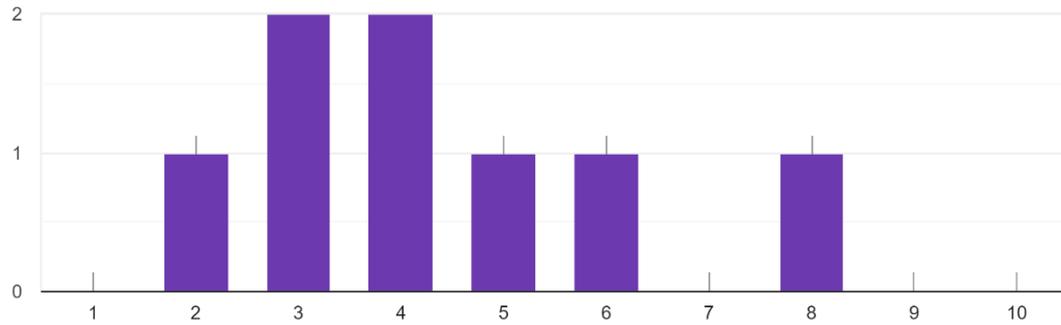
33-¿Cómo puntuarías del 1 al 10 tu experiencia en este estudio?

8 respuestas



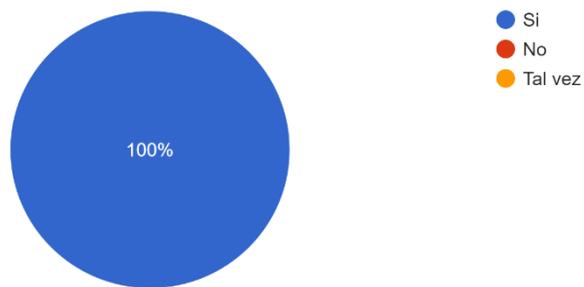
34-¿Con qué grado de dificultad describirías el entrenamiento del control de los movimientos?

8 respuestas



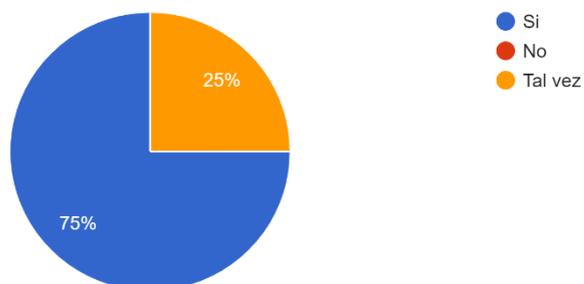
35-¿Crees que es un buen método de control de tecnologías asistenciales para personas con amputación de miembro superior?

8 respuestas



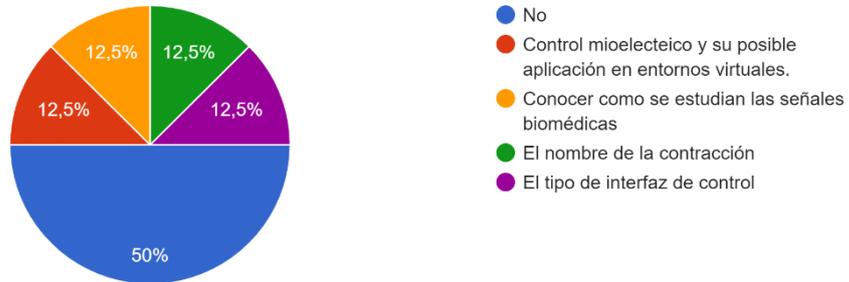
36-¿Crees que es sistema fiable y seguro?

8 respuestas



37-¿Crees que esta prueba te ha aportado algún conocimiento nuevo que no sabías? En caso afirmativo indicar cual.

8 respuestas



38-Finalmente ¿en qué crees que podría mejorar el sistema de control mioeléctrico?

6 respuestas

Nada

Nada

Creo que depende mucho también del propio entrenamiento del usuario, con unas cuantas repeticiones del experimento y unos umbrales bien ajustados los resultados mejorarían mucho. Quizás sería interesante que en desplazamientos largos no hubiera que mantener la contracción para evitar cansancio y dar un tiempo (1 segundo o así) sin actuar al entrar en el nodo de pinza para evitar errores. Otra cosa que añadiría sería poner de nodo inicial el movimiento en horizontal y poner de último nodo de movimiento el movimiento de acercamiento y alejamiento de la mesa, porque creo que sería más rápido el realizar el trial.

Se le podría hacer un control proporcional para que el usuario pueda regular la velocidad

Se podría separar el umbral de activación y el de desactivación, de forma que se active el movimiento de flexión o extensión en un umbral alto, y se desactive con un umbral más bajo.
Utilizar umbrales más altos para el modo 3, puesto que los movimientos son más "bruscos" (solamente cambiar de botón, no requiere de precisión)