

GRADO EN INGENIERÍA DE TELECOMUNICACIONES

TRABAJO FIN DE GRADO

SOFTWARE DE CONTROL DE UN DISPOSITIVO MEDIANTE GESTOS



Estudiante: Nasello Tomaszuk, Germán

Director/Directora: Ferro Vázquez, Armando

Curso: 2023-2024

Fecha: Bilbao, 26/07/2024

Resumen

En un entorno cada vez más integrado con la tecnología, nos encontramos muy limitado a la hora de interactuar con la misma ya que la interacción humano-maquina no ha tenido una evolución tan significativa en los últimos años.

A lo largo de esta memoria, se presenta una propuesta de proyecto para el desarrollo de software para una aplicación, que permita el control de dispositivos mediante gestos. Se propone el uso de librerías para detección de gestos (hand-tracking), así como algoritmos que permitan la interpretación y reconocimiento preciso de los gestos realizados por el usuario. El objetivo principal del proyecto es crear una interfaz intuitiva y eficiente que mejore la experiencia del usuario al interactuar con los dispositivos, ofreciendo una alternativa innovadora y sin contacto físico en comparación con los métodos tradicionales de control.

Laburpena

Teknologiarekin gero eta integratuago dagoen ingurune batean, teknologiarekin elkarreragiteko orduan oso mugatuta gaude, giza eta makinaren arteko elkarreraginak ez baitu bilakaera hain esanguratsurik izan azken urteotan.

Memoria honetan aplikazio baterako softwarea garatzeko proiektu-proposamen bat aurkezten da, keinuen bidez gailuak kontrolatu ahal izateko. Keinuen detektatzeko (hand-tracking) liburutegiak erabiltzea proposatzen da, baita erabiltzaileak egindako keinuen behar bezala interpretatzea eta ezagutzea ahalbidetzen duten algoritmoak ere. Proiektuaren helburu nagusia interfaze intuitibo eta eraginkor bat sortzea da, erabiltzailearen esperientzia hobetzeko gailuekin elkarreraginean, alternatiba berritzaile bat eskainiz eta kontaktu fisikorik gabe, kontrol-metodo tradizionalekin alderatuta.

Abstract

In an increasingly technology-integrated environment, we find ourselves very limited in how we interact with it, as human-computer interaction has not seen significant evolution in recent years.

During this paper, a project proposal about software development of an application that allows controlling devices based on gestures. It is proposed the use of hand-tracking libraries and algorithms that allow precise interpretation and recognition of the gestures made by the user. The main goal is to create an intuitive and efficient interface that improves the user's experience when interacting with surrounding technology, offering an innovative and contactless alternative to traditional control methods.

Palabras clave: Control, gestos, algoritmo, python, hand-tracking

Índice de ilustraciones

Ilustración 1: Esquema de módulos del sistema	23
Ilustración 2: Pantalla principal.....	24
Ilustración 3: Ventana de configuración de gestos	25
Ilustración 4: Diagrama de funcionamiento del sistema	29
Ilustración 5: Información obtenida de librería de hand-tracking.....	30
Ilustración 6: Diagrama de realización de acción.....	31
Ilustración 7: Diagrama de añadir acción.....	32
Ilustración 8: Diagrama de Casos de Uso.....	33
Ilustración 9: Diagrama de actividad.....	34
Ilustración 10: Diagrama de Gantt	37

Índice de tablas

Tabla 1: Criterios de Selección de librería.....	16
Tabla 2: Criterios de Selección de distribución.....	19
Tabla 3: Criterios de selección de implementación.....	21
Tabla 4: Equipos de trabajo.....	36
Tabla 5: Entregables de paquetes de trabajo.....	39
Tabla 6: Hitos del proyecto.....	39
Tabla 7: Costes de recursos humanos	40
Tabla 8: Costes de gastos.....	40
Tabla 9: Costes de amortizaciones.....	41
Tabla 10: Costes del proyecto	41

Índice

1.	Introducción.....	6
1.1	Contexto.....	7
1.2	Objetivos y Alcance.....	8
2.	Conexión con los Objetivos de Desarrollo Sostenible y los principios y valores democráticos	9
3.	Beneficios.....	10
3.1	Beneficios técnicos.....	10
3.2	Beneficios económicos	10
3.3	Beneficios sociales.....	11
4.	Especificaciones	12
4.1	Especificaciones de diseño	12
5.	Análisis de problemática.....	14
5.1	Selección de librería de HandTracking.....	14
5.1.1	MediaPipe	14
5.1.2	HandTrack.js	15
5.1.3	HandPose	15
5.1.4	Criterios de selección.....	15
5.1.5	Toma de decisión	16
5.1.6	Modelo de distribución.....	17
5.1.6	Modelo Open Source	17
5.1.7	Modelo Privado	17
5.1.8	Distribución de licencias	18
5.1.9	Criterios de selección.....	18
5.1.10	Toma de decisión.....	19
5.2	Decisiones de implementación.....	20
5.2.1	Aplicación Local	20
5.2.2	Aplicación integrada con servidor	20
5.2.3	Criterios de selección.....	21
5.2.4	Toma de decisión	21
6.	Diseño de alto nivel	22
6.1	Introducción.....	22
6.2	Arquitectura del sistema.....	22

6.3	Módulo de Interfaz Gráfica de Control.....	24
6.4	Módulo de cálculos de hand tracking.....	27
6.5	Módulo de configuración de acciones.....	28
6.6	Funcionamiento del Técnico.....	29
6.7	Diagrama de Secuencia.....	31
6.7.1	Diagrama de Realización de acción	31
6.7.2	Diagrama de añadir acción	32
6.8	Diagrama de Casos de Uso.....	33
6.9	Diagrama de Actividad del sistema.....	34
7.	Plan de trabajo.....	36
7.1	Introducción.....	36
7.2	Equipo de trabajo.....	36
7.3	Paquetes de trabajo.....	36
7.4	Materiales.....	38
7.5	Entregables.....	39
7.6	Hitos	39
8.	Presupuesto	40
8.1	Recursos humanos.....	40
8.2	Gastos.....	40
8.3	Amortizaciones.....	41
8.4	Costes del proyecto.....	41
9.	Conclusiones.....	42

1. Introducción

La tecnología avanza a pasos agigantados, y también lo hace el uso que hacemos de la misma. Actualmente, podemos decir que está presente de forma constante en nuestras vidas. En todo momento estamos comunicados gracias a los teléfonos móviles, a diario accedemos a internet para buscar información de temas de interés, trabajo y ocio, entre otros.

Pero, a pesar de ser la tecnología una herramienta vital, es notoria la dificultad de los usuarios a la hora de comunicarse con la misma de forma natural. Generalmente se usan métodos de contacto directo con el dispositivo o de forma remota a través de mandos de control remoto. Esta limitación empeora significativamente la experiencia de los usuarios con la tecnología, ya que el control del dispositivo no se hace de una forma natural.

Últimamente, gracias a la aparición de asistentes personales como Alexa, Siri y Google Assistant, se ha vuelto más común el uso de la voz a la hora de controlar dispositivos lo que ha demostrado que es posible un control más intuitivo de nuestros dispositivos. Sin embargo, aún existe una amplia selección de opciones nuevas a explorar como otras formas de comunicación, que puedan ser más accesibles en nuestra vida.

Surge así la necesidad de un software que permita comunicarnos con la tecnología como una extensión de nuestro cuerpo. A lo largo de este documento se detalla una propuesta para permitir aumentar aún más dicha comunicación mediante el uso de librerías de control gestual (hand-tracking), es decir convertir los gestos que hace un usuario a acciones de control automatizadas.

1.1 Contexto

Para entender mejor este proyecto, es importante definir el contexto sobre el que ha sido desarrollado, y tener así una mejor comprensión de sus objetivos y motivaciones durante el tiempo de vida del proyecto.

Nace inicialmente como un reto personal y motivado por la falta de disponibilidad de soluciones para controlar dispositivos de mi hogar con mayor facilidad y sin la necesidad de tener tantos mandos y conectores perdidos en la casa. Pero al ver el potencial técnico y social del mismo, se convirtió en un proyecto más grande y ambicioso.

Para conseguir este objetivo, se ha desarrollado código que haciendo uso de técnicas de Computer Vision es capaz de entender los movimientos de un usuario para poder así reaccionar en consecuencia. El Computer Vision es una rama de la inteligencia artificial cuyo objetivo es desarrollar modelos que sean capaces de entender imágenes del mismo modo que lo haría un humano.

La solución que desarrollaremos en este documento se centra en la creación de un software que permita llegar a ese objetivo. Todo el código estará basado en el lenguaje de programación Python, conocido por su simpleza y facilidad en el manejo de datos. También se utiliza la librería de hand-tracking *MediaPipe*, desarrollada por Google, para poder detectar y procesar la posición y movimiento de las manos.

1.2 Objetivos y Alcance

El objetivo de este proyecto es el desarrollo de un programa que permita al usuario controlar el dispositivo mediante gestos de la mano. Esta propuesta utiliza tecnologías de visión por ordenador para interpretar los gestos del usuario y traducirlo en comandos a ejecutar. Para ello se definen los siguientes objetivos específicos:

- **Hacer un estudio de las librerías** más adecuadas para reconocer los gestos (hand-tracking). Esto incluye un estudio de características, precisión e información. La selección de una librería adecuada nos permitirá cumplir el resto de los objetivos con mayor facilidad.
- **Desarrollar un software básico** que entienda los gestos que se realizan, y poder llevar a cabo acciones predefinidas en consecuencia. Con el cumplimiento de este objetivo tendremos un código funcional
- **Diseñar una interfaz gráfica** que permita hacer el sistema más configurable, es decir, permitir que el usuario decida qué gestos están relacionados con que acción, y configurar apartados del funcionamiento como la sensibilidad y la necesidad de detección mínima

Definir correctamente los objetivos permite limitar con mayor claridad el alcance del proyecto. El alcance incluye el desarrollo de una aplicación configurable que sea capaz de reconocer y entender gestos de las manos. Esto implica realizar un análisis exhaustivo de las diferentes librerías disponibles, evaluando su precisión y características para seleccionar la más adecuada.

Es importante aclarar que no se va a llegar a un producto final, sino que es una prueba de concepto. En cuanto al ámbito de la aplicación se limita a una solución básica, aplicable en un marco más general. No obstante, partiendo de este proyecto es posible explotar esta solución para obtener una aplicación que cumpla las especificaciones requeridas.

2. Conexión con los Objetivos de Desarrollo Sostenible y los principios y valores democráticos

Los Objetivos de Desarrollo Sostenible (ODS) son un total de 17 objetivos globales que se adoptaron por la Asamblea General de Naciones Unidas con la intención de afrontar problemas mundiales como la pobreza, el cambio climático y la desigualdad. Los ODS que tienen relación con este proyecto son:

Objetivo 9. Industria, Innovación e Infraestructura: El Objetivo 9 pretende construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación. El crecimiento económico, el desarrollo social y la acción por el clima dependen en gran medida de las inversiones en infraestructuras, el desarrollo industrial sostenible y el progreso tecnológico. Ante la rápida evolución del panorama económico mundial y el aumento de las desigualdades, el crecimiento sostenido debe implicar una industrialización que, en primer lugar, haga accesibles las oportunidades a todas las personas y, en segundo lugar, se apoye en la innovación y en infraestructuras resistentes. [5]

Objetivo 10. Reducción de las Desigualdades: La desigualdad amenaza el desarrollo social y económico a largo plazo, frena la reducción de la pobreza y destruye el sentido de realización y autoestima de las personas. Para reducir la desigualdad tanto dentro de los países como entre ellos es necesario distribuir equitativamente los recursos, invertir en la enseñanza y el desarrollo de capacidades, implementar medidas de protección social, luchar contra la discriminación, apoyar a los grupos marginados y fomentar la cooperación internacional para un comercio y sistemas financieros justos. [5]

Este proyecto, por una parte, cumple el objetivo 9 porque permite adaptar las fábricas y sus infraestructuras para que sean más seguras y avanzadas, pudiendo aplicar estas soluciones en automatización, control de maquinaria y seguridad del trabajador.

Por otra parte, el objetivo 10 pretende acabar con la desigualdad. Este proyecto se puede aplicar con este objetivo, concretamente, en contra de las desigualdades por causas de discapacidad, permitiendo a estas controlar dispositivos con mayor facilidad que con los métodos tradicionales.

3. Beneficios

3.1 Beneficios técnicos

Se entiende como beneficios técnicos aquellos beneficios que aporta este proyecto dentro de los apartados de la programación o los apartados más relacionados con la ingeniería del sistema. En el caso de este proyecto los beneficios técnicos más relevantes son los siguientes:

- **Innovación en Interacción Humano-Computadora:** La aplicación utiliza tecnología avanzada de hand-tracking para permitir el control de dispositivos mediante gestos, representando un avance significativo en la interacción humano-computadora.
- **Solución de Código Abierto:** La utilización de librerías de código abierto como Mediapipe y herramientas gratuitas como Python fomenta la transparencia y la colaboración en la comunidad tecnológica.

3.2 Beneficios económicos

Inicialmente este proyecto no ha sido pensado para obtener beneficios económicos sino como un proyecto lúdico para el análisis y la potencialidad de este tipo de lenguaje de comunicación. No obstante, y luego de entender las posibles aplicaciones comerciales, he detectado un rango nicho de mercado donde puede ser aplicado y comercializado:

- **Fábricas con altos niveles de ruido:** Donde no sean funcionales otros modos de comunicación, un sistema que sea capaz de entender el lenguaje natural puede ser de gran ayuda.
- **Asociaciones de personas con discapacidad:** Estas asociaciones pueden estar realmente interesadas en soluciones que ayuden a mejorar la calidad de vida de las personas con discapacidad al hacer la tecnología más accesible para ellos.

- **Industria Química:** En estas industrias, al estar regularmente manejando productos potencialmente peligrosos, hay situaciones en las que el control mediante tacto no es una opción. Por eso, esta solución tiene hueco en ese mercado ya que permite controlar los dispositivos sin necesidad de tocarlos

3.3 Beneficios sociales

Al distribuirse este software de forma gratuita y permitir a todo usuario disfrutar del mismo, los beneficios sociales que este proyecto aporta son la posibilidad de tener a su disposición un nuevo método para controlar los dispositivos que le rodean. Esto es de gran relevancia porque permite a personas con problemas psicomotrices interactuar con su ordenador sin hacer uso del teclado o ratón.

- **Inclusión y Accesibilidad:** Proporciona una solución accesible para personas con discapacidades físicas, permitiéndoles controlar dispositivos sin necesidad de contacto físico.
- **Reducción de Barreras Digitales:** Al ofrecer una forma alternativa y natural de interactuar con la tecnología, se reducen las barreras digitales y se fomenta una mayor inclusión digital.

4. Especificaciones

El sistema utilizará algoritmos para identificación y seguimiento de manos para diferenciar los gestos hechos por el usuario y reaccionar en consecuencia. La arquitectura del sistema tendrá que ser simple para que el software sea flexible, adaptable y tenga la suficiente facilidad de integración con otros dispositivos como para poder ser exportada a una gran cantidad de dispositivos.

También necesitará acceso a una cámara web integrada o externa para poder capturar las imágenes que posteriormente serán procesadas en busca de manos gracias a MediaPipe. De este modo se conseguirá la información necesaria para interpretar las acciones del usuario. Estas imágenes son totalmente necesarias para el funcionamiento del programa, gracias a estas se guardará la información del gesto realizado.

Además, se ha definido una interfaz intuitiva gráfica que permitirá al usuario añadir y eliminar relaciones entre gestos con sus correspondientes acciones. Esta interfaz facilita el uso de la aplicación, permitiendo al usuario configurar el funcionamiento del programa de modo que se adecue a sus necesidades. Un gesto está compuesto por la posición de la mano y su movimiento. Se entiende por posición de una mano como el estado de cada uno de los dedos (abierto o cerrado) y por movimiento se entiende como la suma de desplazamientos que ha tenido la mano durante la realización del gesto.

En última instancia, el programa ha de ser capaz de permitirle al usuario tener un control fluido de sus dispositivos, y hacer que la comunicación con los mismos se haga de forma clara. De esta manera se espera que la propuesta no solo contribuya a un desarrollo de la relación que tenemos con la tecnología, sino que también sirva como guía para desarrolladores que quieran seguir en por estas mismas vías de trabajo.

4.1 Especificaciones de diseño

El diseño de la arquitectura también tiene que seguir ciertas especificaciones, dentro de ellas encontramos que se tiene que separar las funciones en distintos módulos. Cada uno de estos módulos debe tener su propio propósito, como el manejo de la interfaz gráfica o procesar los datos. Esto no solo facilita el mantenimiento del software, sino que también permite que la solución sea más escalable, ya que si se quiere en un futuro añadir nuevas funcionalidades solo habría que implementarlas en cada uno de los módulos.

Además, el sistema ha de ser compatible con los principales sistemas operativos. Esto aumenta el mercado potencial y no limita la disponibilidad del servicio por motivos de software en uso, asegurando una mayor accesibilidad.

Por último, es necesario optimizar el procesamiento de datos para asegurar un uso eficiente de la CPU y la memoria, que pueden ser críticos en la calidad del funcionamiento que percibe el usuario. Esto incluye técnicas como el procesado eficiente de los datos y uso correcto de los recursos del sistema.

5. Análisis de problemática

Una vez explicadas las especificaciones y el funcionamiento del programa se tiene un entendimiento más en profundidad de la aplicación. Es por eso por lo que por lo que ahora se van a analizar diferentes problemas de diseño que ha tenido que afrontar este proyecto y las decisiones tomadas en cada caso:

- Selección de librería de HandTracking
- Modelo de distribución
- Decisiones de implementación

Para ello, teniendo en cuenta los objetivos del proyecto, se puede determinar cuáles de las opciones revisadas y los motivos de selección.

5.1 Selección de librería de HandTracking

Este proyecto se basa en el uso de librerías de programación que son capaces de gracias al Computer Vision son capaces de analizar imágenes para ofrecer información de las manos que se encuentran en la imagen. A estas librerías se les llama librerías de hand-tracking. Esta información varía en función de la librería, pero generalmente incluye información sobre las coordenadas de la mano.

Por lo tanto, a la hora de afrontar este proyecto, la primera y más obvia decisión a tomar fue qué librería usar para la detección y trackeo de las manos. Las librerías disponibles para hand-tracking no presentan una gran diferencia entre ellas, pero sí son suficientes para plantearlo como un problema de análisis de alternativas.

5.1.1 MediaPipe

Por una parte, MediaPipe ofrece una amplia variedad de plataformas sobre las que implementar su solución. Está desarrollada por Google y a fecha de hoy sigue teniendo soporte por lo que es constantemente actualizado y mantenido por los desarrolladores. Tiene una gran cantidad de opciones, tales como detección y seguimiento del rostro y cuerpo, seguimiento del iris, etc. De todas las opciones que permite, la que se va a usar en este proyecto es la detección de manos de alta precisión y rendimiento optimizado. Su principal punto fuerte es que además de informarnos de donde está situada una mano, nos ofrece información de las coordenadas X, Y y Z de diferentes puntos de la mano. Como gran desventaja tenemos la reducida cantidad de documentación en esta librería

5.1.2 HandTrack.js

Por otra parte, HandTrack.js es una librería que funciona únicamente sobre JavaScript. Esta también tiene soporte, pero al estar desarrollada por un desarrollador independiente y no estar respaldado por una gran organización, las actualizaciones no son tan frecuentes. Esta librería nos permite detectar manos, pero a diferencia de MediaPipe no otorga información detallada sobre los dedos de la mano.

5.1.3 HandPose

HandPose, librería desarrollada por TensorFlow, funciona sobre JavaScript y a día de hoy sigue recibiendo soporte activo por parte de sus desarrolladores. A diferencia de MediaPipe, en su versión actual solo es capaz de detectar una mano al mismo tiempo, con la promesa de aumentar este límite en futuras versiones. Además, otorga una aproximación del esqueleto de la mano mediante coordenadas tridimensionales al igual que MediaPipe.

5.1.4 Criterios de selección

Al querer que el programa analice en tiempo real y de forma efectiva los gestos realizados por el usuario, los criterios principales a la hora de elegir librería serán los siguientes:

- **Rendimiento (40%):** El rendimiento es el criterio principal para esta selección, ya que permite minimizar la latencia y los errores del sistema.
- **Nivel de detalle (40%):** La capacidad de proporcionar información detallada sobre la posición de los dedos de la mano es otro criterio importante, ya que permite que el programa sea más escalable.

- Número máximo de manos a detectar (20%):** El último criterio será cuál es el número máximo de manos que el programa es capaz de detectar, ya que a un mayor número de manos sea capaz de detectar, menor es la posibilidad de que el programa no funcione.

5.1.5 Toma de decisión

Teniendo en cuenta los criterios mencionados en el punto anterior, podemos tomar una decisión según la siguiente tabla:

Criterio	MediaPipe	HandTrack.js	HandPose
Rendimiento	7	5	7
Nivel de detalle	10	2	7
N.º máximo de manos	6	6	1
Total	8	4	5.8

Tabla 1: Criterios de Selección de librería

Observando la tabla, podemos observar como la librería MediaPipe ha obtenido la puntuación más alta en todos los criterios, ya que como bien se ha mencionado antes, es la librería que aporta más información y tiene un mayor rendimiento.

1.1 Modelo de distribución

El modelo de distribución determina cómo se entrega y se pone a disposición del usuario final el código desarrollado: de manera gratuita, de pago o mediante licencias. Analizar estas alternativas es importante para determinar la rentabilidad y los beneficios anteriormente mencionados.

5.1.6 Modelo Open Source

Según el modelo Open Source, la totalidad del código está disponible para que cualquiera pueda acceder al código fuente, mientras que el propietario es el único que puede permitir cambios en el repositorio. Esto no impide que el resto de los usuarios y desarrolladores puedan ejecutar y modificar el proyecto, pero esto tendrá que hacerse desde una copia local. Un modelo open source también puede ser desarrollado en diferentes líneas en paralelo, lo que significa que al mismo tiempo se puede trabajar en funcionalidades o versiones distintas del código por diferentes motivos, tales como desarrollar el código para diferentes sistemas operativos o preparar el código para diferentes tipos de usuarios.

Este modelo destaca por compartir el conocimiento y las soluciones por encima del aspecto económico, por lo que no es el ideal si se pretende que el proyecto sea lucrativo, pero al permitir que haya más contribuyentes capaces de trabajar en el código, es posible desarrollar mejores funcionalidades y minimizar errores.

5.1.7 Modelo Privado

Otro modelo por destacar es el modelo privado, según el cual el software se distribuye sólo entre los usuarios que hayan pagado. Este enfoque se caracteriza por buscar que el proyecto sea lo más lucrativo posible. Si no se incluyen técnicas para verificar que las credenciales del usuario coincidan con un usuario pagado se corre el riesgo de que se distribuyan copias ilícitas del software.

Solo los usuarios que han pagado pueden acceder al software, por lo que el acceso al mismo es más controlado. También permite una mayor protección sobre la propiedad intelectual, ya que solo se distribuye el código compilado, sin acceso al código base. Los beneficios económicos que ofrece este modelo de distribución es que la monetización se puede ampliar mediante servicios adicionales como ofreciendo soporte continuo y actualizaciones regulares.

5.1.8 Distribución de licencias

De forma similar al anterior, en este enfoque solo se permite el uso del software a usuarios de pago, ya que para usar el código es necesario proporcionar una licencia para la activación. Esto requiere del despliegue de un servicio que compruebe la validez de las licencias extendidas. La ventaja del cobro mediante licencias es que permite una mayor variación en la tarificación y en la limitación de los productos en función del plan seleccionado.

Un software que es distribuido mediante licencias puede permitir a los usuarios modificar el software siguiendo la licencia del software, donde el creador detalla las condiciones impuestas para modificar el software.

Un apartado en el que esté destaca es el apartado económico, ya que permite:

- **Ofrecer distintas soluciones** a distintos usuarios, personalizando la solución a las necesidades.
- **Ingresos constantes** durante la duración de la licencia.
- **Planes de pago personalizados**, ya que se pueden ofrecer distintos precios con distintas ventajas, de modo que un usuario que necesite una experiencia básica no está obligado a pagar un precio fijo muy alto, y viceversa, ofrecerle a un usuario que necesite una gran cantidad de opciones a un precio mayor.

5.1.9 Criterios de selección

En cuanto a los criterios de selección, los criterios a valorar serán los siguientes:

- **Menor requerimientos computacionales (80%):** Al ser un proyecto de bajos recursos que no está orientado a obtener tantos beneficios económicos, la elección que reduzca en mayor el impacto sobre los gastos será de gran importancia.
- **Escalabilidad de código (20%):** Para que el proyecto sea escalable y permita incluir nuevas funcionalidades en un futuro, es importante escoger un modelo de distribución que permita que el código se pueda desarrollar de forma estandarizada

Al ser un proyecto nuevo en desarrollo y estar más enfocado hacia el desarrollo de soluciones más innovadoras, se ha realizado este análisis de alternativas sin centrarse en su apartado económico.

Si el proyecto avanza a una fase comercial, este punto tiene que ser reanalizado según los nuevos parámetros.

5.1.10 Toma de decisión

	Open Source	Privado	Licencias
Menor requerimiento computacional	10	7	5
Escalabilidad de código	6	8	4
Total	9.2	7.2	4.8

Tabla 2: Criterios de Selección de distribución

Observando los resultados de cada una de las alternativas, se comprueba que el que mejor métrica ponderada tiene es el modelo de código abierto. Si bien el modelo privado rinde significativamente mejor en el apartado de escalabilidad de código, ya que permite que al desarrollar nuevas funcionalidades se hace de forma estandarizada. Esto también es posible con el resto de los modelos, pero en el modelo de open source requiere un gran trabajo por parte del encargado del código.

5.2 Decisiones de implementación

Las decisiones de implementación tienen en cuenta de qué manera se va a desplegar y donde se va a ejecutar cada parte de la lógica de la aplicación. Estas decisiones afectan al rendimiento general de la aplicación, por lo que es de vital importancia un correcto análisis de alternativas y consecuencias.

5.2.1 Aplicación Local

Una aplicación local se ejecuta en su totalidad en el dispositivo en el que está instalado. En este caso el rendimiento depende únicamente del dispositivo del usuario, por lo que se evita una posible situación de deterioro de la experiencia de usuario por motivos de saturación en el sistema, a pesar de que pida ordenadores con capacidades gráficas ligeramente superiores para procesar las imágenes en tiempo real. Estas carencias pueden ser suplidas si se permite al usuario configurar parámetros de rendimiento para adecuar el grado de procesamiento a sus necesidades personales.

Por otra parte, al no necesitar una conexión con servicios externos, el programa está disponible para el usuario en todo momento. Además, le añade seguridad al sistema e integridad a los datos del usuario. Esto es de gran importancia ya que para que un usuario esté cómodo con usar de forma regular un programa que captura las imágenes de forma constante.

5.2.2 Aplicación integrada con servidor

Si la lógica de la aplicación se integra en un servidor, en el dispositivo solo se ejecutará la parte gráfica de la aplicación y los comandos resultantes del uso de la aplicación, es decir, los comandos que se tienen que realizar una vez se haga el gesto. Los beneficios de esta arquitectura son que minimizan totalmente las necesidades computacionales por parte del equipo del usuario, permitiendo que se ejecute en una amplia variedad de dispositivos, especialmente si en futuras evoluciones se requiere de mayor potencia de procesamiento. Como contraparte, al ser mayor el tiempo de comunicación con un servidor que el tiempo de procesamiento en local, el tiempo de reacción del sistema será mucho mayor. Destacamos que el estado actual de la tecnología permite minimizar estos tiempos (redes de fibra óptica con tiempos de latencia bajos), pero aún podríamos encontrarnos con problemas como pueblos perdidos en el medio de la nada o necesidades de tecnología que requieran configuración.

También se requiere una arquitectura externa para dar servicio a los usuarios, lo que aumenta el coste del proyecto, es susceptible a fallos de red y alto volumen de usuarios, aumenta el riesgo ante ataques y tal y como comentamos en el apartado anterior, puede suponer desconfianza entre los usuarios.

5.2.3 Criterios de selección

Una vez explicadas las alternativas relativas a esta selección, se procede a describir los criterios ponderados para tomar una mejor decisión.

- **Maximizar experiencia de usuario (40%):** Es importante que el usuario tenga una buena sensación a la hora de usar este sistema. Esto incluye aspectos como reducir el tiempo de respuesta al mínimo y disponibilidad constante.
- **Seguridad de datos (40%):** Al hacer uso de la cámara de forma constante, la seguridad que se le ofrece al usuario tiene que ser alta para evitar que las imágenes sean capturadas por entidades externas
- **Económico (20%):** El aspecto económico es también importante ya que al no estar este proyecto planteado como algo económicamente rentable sino ser una prueba de concepto, la opción más viable será más positiva

5.2.4 Toma de decisión

Si aplicamos los criterios de selección a las alternativas planteadas, obtenemos los siguientes datos:

	Aplicación local	Aplicación integrada con servidor
Experiencia de usuario	8	5
Seguridad de datos	10	4
Aspectos Económicos	7	4
Total	8.6	4.4

Tabla 3: Criterios de selección de implementación

Como se puede observar, la alternativa con mejor puntuación en el análisis de alternativas es la aplicación que funcione en modo local. Esto tiene sentido ya que destaca en los tres criterios. Es por ello por lo que se ha optado por desarrollar el software sin dependencia de software externo.

6. Diseño de alto nivel

6.1 Introducción

En este apartado se detalla la solución técnica propuesta para el desarrollo del sistema. Inicialmente, se describe el funcionamiento general del sistema, destacando sus componentes principales y cómo interactúan entre sí. Después, se presentará un diagrama UML para ilustrar de manera visual la arquitectura del sistema y las relaciones entre sus distintos elementos. Empezamos explicando los distintos módulos.

6.2 Arquitectura del sistema

Este proyecto ha sido diseñado en distintos módulos, teniendo cada uno de ellos su propio objetivo y funcionalidades. Todos ellos están relacionados entre sí a través del programa principal, donde se guarda la lógica principal del código.

Los distintos módulos son:

- **Módulo de Interfaz Gráfica de Control:** Permite la comunicación entre el usuario y el programa
- **Módulo de cálculos de hand tracking:** Se encarga de realizar los cálculos y procesar los datos
- **Módulo de configuración de acciones:** Se encarga de relacionar los distintos gestos con las acciones.

Esquema del Sistema

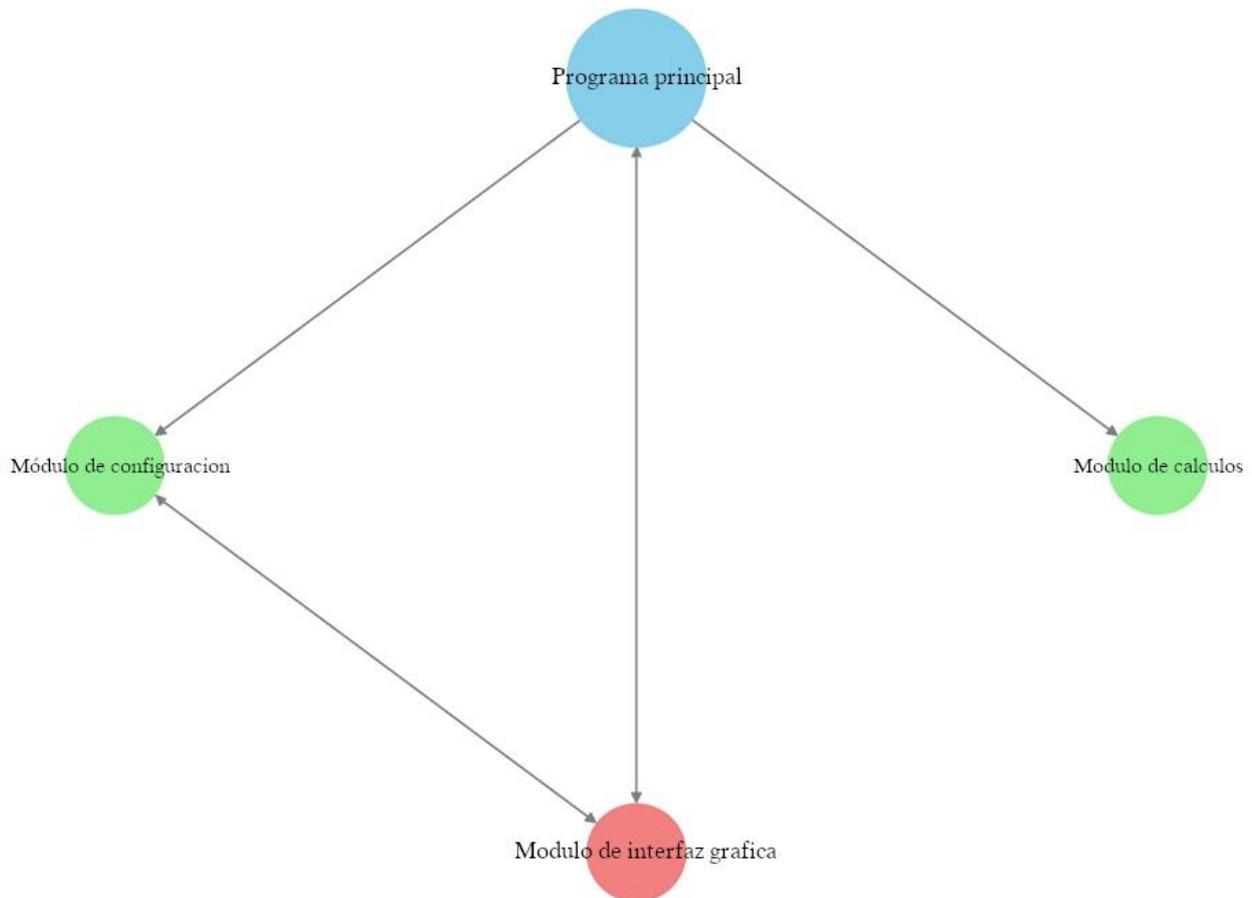


Ilustración 1: Esquema de módulos del sistema

Como se ve en la imagen, el módulo de configuración de acciones está integrado con el programa principal. Esto significa que toda su lógica se implementa directamente en el código del programa.

El módulo de cálculos de hand tracking, que se encarga de procesar toda la información obtenida de la librería de hand tracking, también está integrado en el programa principal. La única diferencia con el anterior es que su lógica se ha diseñado en una clase nueva, y se ha hecho un objeto de esta en el código del programa.

El módulo de interfaz gráfica de control funciona de forma paralela al programa principal, cada uno en un hilo distinto. La interfaz gráfica sólo se comunicará con el programa principal cuando pase uno de los siguientes eventos:

- El usuario quiere pausar/reanudar la ejecución, en cuyo caso cambiará el valor de una variable booleana que controla el funcionamiento del sistema
- El usuario ha cambiado la configuración o los gestos conocidos, en cuyo caso le avisará al programa principal que ha de volver a cargar esta información

También hay una relación entre este módulo y el de configuración de acciones, ya que desde la interfaz gráfica se accede y se modifica el archivo que guarda la lista de acciones directamente desde la interfaz gráfica

6.3 Módulo de Interfaz Gráfica de Control

La interfaz gráfica es lo que verá el usuario y lo único que tendrá disponible para comunicarse con el sistema. Esta está compuesta por 4 botones y una consola inferior a través de la cual se van a comunicar los sucesos

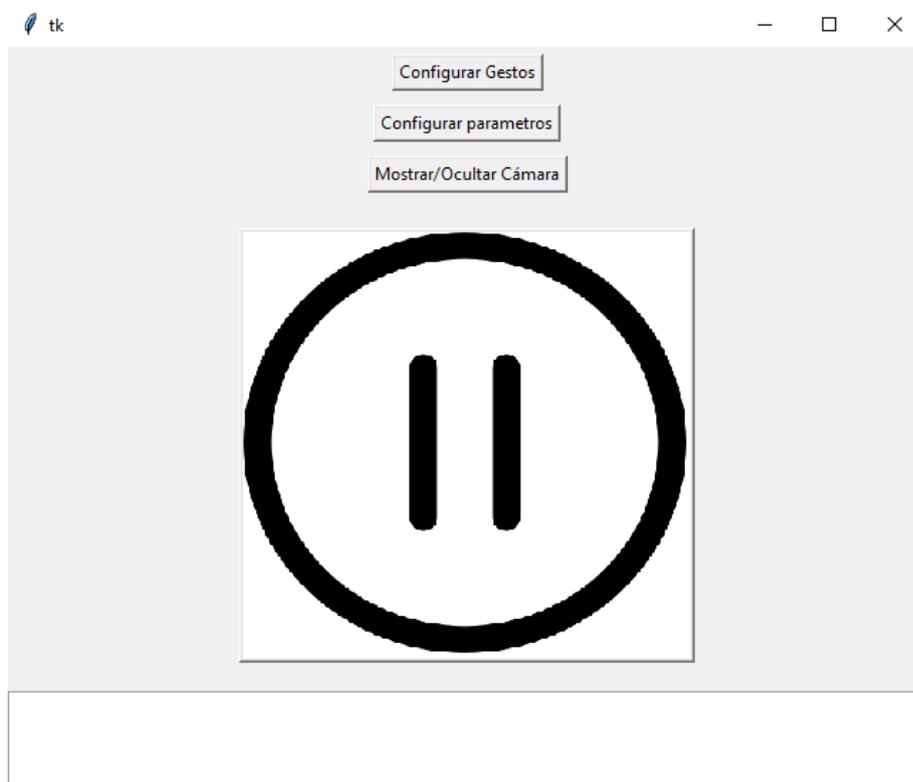


Ilustración 2: Pantalla principal

Como se puede ver en la foto, los dos primeros botones se utilizan para la configuración, siendo el primero de ellos para configurar los gestos y las acciones asociadas al mismo. Es importante recordar que, en este contexto, un gesto está compuesto por la posición de los dedos y un movimiento en una de las direcciones principales. Al hacer clic se abrirá una nueva ventana que permitirá consultar, añadir y eliminar relaciones entre gestos y acciones.

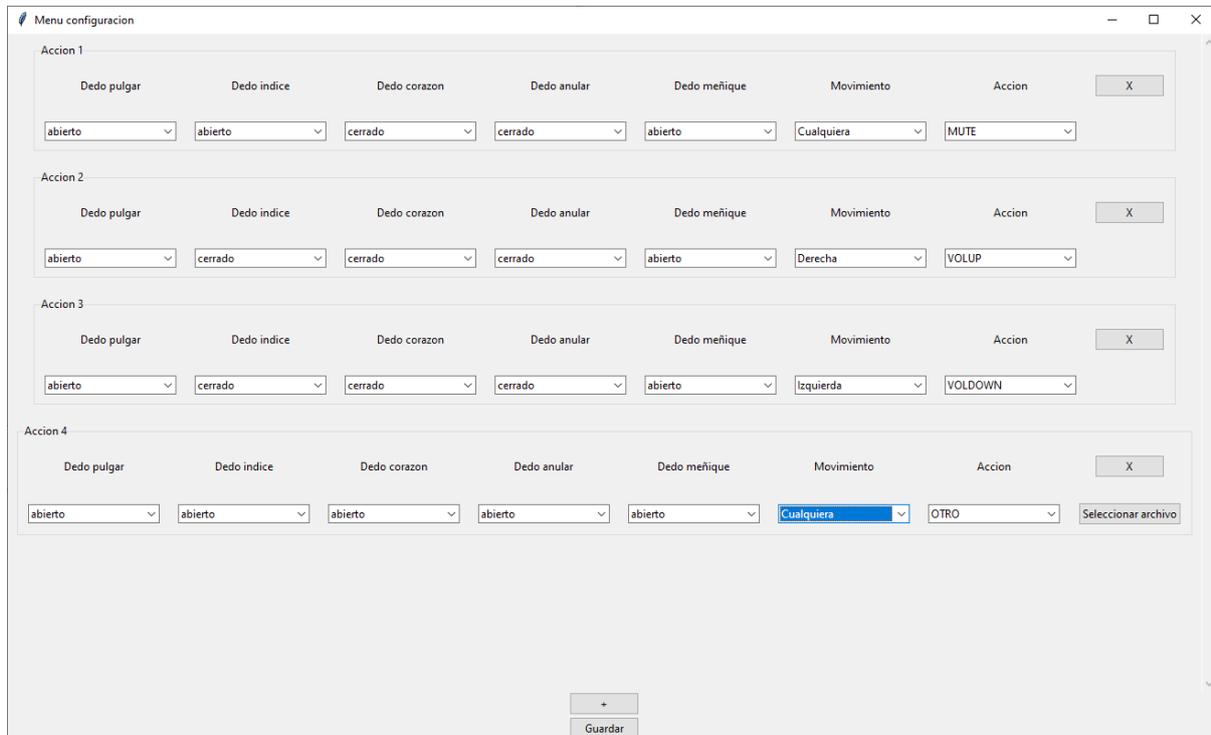


Ilustración 3: Ventana de configuración de gestos

En la primera ejecución, vienen cargados las 3 primeras acciones, de ahí en adelante si se quiere añadir un nuevo gesto hay que darle al botón inferior “+” y rellenar la información de la nueva línea que nos aparece. Por cada dedo, hay que decir en qué posición queremos que esté el (abierto/cerrado), también hay que seleccionar el movimiento (arriba, abajo, izquierda, derecha o cualquiera) y por último hay que seleccionar la acción asociada al gesto. Hay 3 acciones predefinidas, **Mute, Subir Volumen o Bajar Volumen**, pero también se puede abrir un archivo del sistema a seleccionar, por eso, en caso de que la acción seleccionada sea **Otro**, aparecerá un botón que nos permitirá seleccionar el archivo. El botón Guardar, cargará la configuración al archivo acciones.json. Es ahí donde se guardan las relaciones entre gestos y acciones.

Por otra parte, el botón de **Configurar Parámetros** abre una ventana que permite personalizar el funcionamiento del sistema. Las opciones por configurar son las siguientes:

1. **Umbral de inicio:** Este parámetro indica la sensibilidad para realizar una acción, es decir, durante cuantos fotogramas consecutivos tiene que ver la misma posición de la mano como para entender qué está haciendo un gesto.
2. **Umbral de fin:** De forma opuesta al apartado anterior, este parámetro determina el número de fotogramas necesarios sin detectar el gesto para considerar que se había terminado de hacer el gesto
3. **Confianza mínima de detección:** Este valor es el umbral de probabilidad requerido para que el sistema considere que ha detectado una mano, debe tener valores entre 0 y 1, siendo 1 el valor más restrictivo
4. **Confianza mínima de trackeo:** De forma similar al anterior, este valor indica el umbral de probabilidad para que el sistema continúe siga la mano detectada en fotogramas consecutivos

Estos valores se guardan en el archivo *config.json* el cual es abierto por el programa principal para cargar sus datos y aplicarlos en el funcionamiento.

El botón **Mostrar/Ocultar Cámara**, como su nombre indica, abrirá o cerrará una ventana que muestra lo que el programa ve. Este es especialmente útil para saber si el gesto se está haciendo bien centrado en la cámara y si el software está detectando la mano correctamente. También sirve para saber si los parámetros configurados adecuan el funcionamiento a sus necesidades por el entorno.

El último botón sirve para poner en pausa o en marcha el funcionamiento. Mientras esté en pausa no se capturaron imágenes por la cámara ni se procesa ninguna información.

Por último, en el panel inferior se mostrará información de la aplicación como los gestos detectados y la acción realizada.

6.4 Módulo de cálculos de hand tracking

Para el correcto funcionamiento del sistema, se necesita garantizar que la información que se recibe se procesa de forma adecuada. Para ello se ha desarrollado un módulo que se hará cargo de procesar y almacenar la información de las manos. Sus funciones serán las siguientes:

- **Calcular distancias entre coordenadas:** Gracias a este módulo podemos obtener la distancia entre dos puntos distintos. Las coordenadas de cada uno de los puntos se obtienen en un espacio tridimensional, y si bien la dimensión de profundidad no tiene relevancia en los cálculos actualmente, es interesante hacer los dejar la lógica en un módulo aparte en aras del futuro, mejorando su escalabilidad.
- **Comprobación de posición de la mano:** Esta es la función principal del módulo, recibiendo la información de los puntos de la mano aplicando la lógica anteriormente mencionada:

'0': dedo abierto

'1': dedo cerrado

Con esta información, compone y devuelve un código que representa el estado de la mano.

- **Guardar y procesar el movimiento de la mano:** Durante la realización de un gesto, para saber en qué dirección se ha desplazado la mano guarda las coordenadas X e Y de la muñeca. Una vez terminado el gesto, la suma aritmética de estas coordenadas define en cuál de los ejes y sentidos se ha movido la mano.

6.5 Módulo de configuración de acciones

Este módulo tiene como objetivo relacionar los gestos con las acciones. Está integrado dentro del programa principal. Con el objetivo de entender mejor la lógica que sigue este módulo, hay que tener en cuenta que un gesto está formado por una posición de las manos y su movimiento.

Para la que el programa sea capaz de entender los distintos gestos de forma programática y darles un valor asignable se ha desarrollado la siguiente lógica: Partiendo de que la librería de hand tracking proporciona las coordenadas X, Y y Z de diferentes puntos de la mano, con comparar las distancias entre: muñeca-nudillo y muñeca-punta del dedo se puede saber si el dedo está abierto o cerrado.

El estado de una mano se asocia a una cadena de 5 caracteres, cada carácter indica si un dedo está abierto o cerrado, de modo que una mano abierta tendrá código "00000" y una mano cerrada tendrá código "11111". Siguiendo la misma lógica, un gesto se compone por el estado de la mano haciendo el gesto y una dirección, por lo que a este código se le puede añadir un sexto carácter que indica hacia donde se ha movido. Por ejemplo, "00000U" indica que una mano con todos los dedos abiertos se ha movido hacia arriba (Up). Teniendo esto en cuenta, los gestos configurados por el usuario se guardan en el archivo acciones.json, donde su clave serán los gestos guardados y su valor la acción asociada. Por ejemplo:

```
{  
  "00000U": "Acción / ruta_del_archivo"  
}
```

Al cargar la aplicación o al guardar un nuevo gesto, el programa principal lee y guarda códigos de los gestos guardados en este archivo. Esto se hace para que a lo largo del funcionamiento del programa comparar si la posición de la mano coincide con algún elemento de esta lista.

Una vez que el usuario termine de realizar el gesto, el programa principal obtiene del módulo de hand tracking la posición de las manos al realizar el gesto y la dirección hacia donde se mueve. De este modo, obtiene un código de 6 caracteres formados por estos dos datos. Este código lo compara con los códigos guardados en el en la lista de acciones, y realiza su acción asociada.

6.6 Funcionamiento del Técnico

Para entender en profundidad la arquitectura del sistema, es importante entender también su funcionamiento, tanto técnico como práctico, así como sus diferentes módulos.

Sistema de Control por Gestos

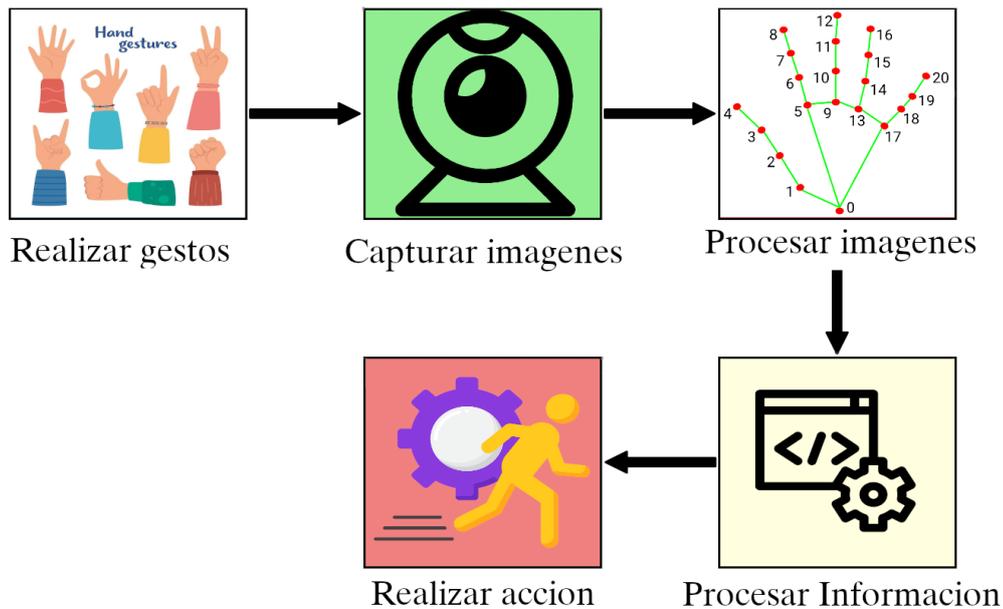


Ilustración 4: Diagrama de funcionamiento del sistema

Al iniciar la aplicación, se cargan dos módulos distintos: la lógica de la aplicación y la interfaz gráfica, y será ahí cuando se carguen los parámetros de configuración. El programa estará constantemente tomando instantáneas a través de la cámara web. Estas serán procesadas por el software de hand tracking, del cual obtendremos información de la posición de los 21 puntos distintos de la mano como se ve en la siguiente foto:

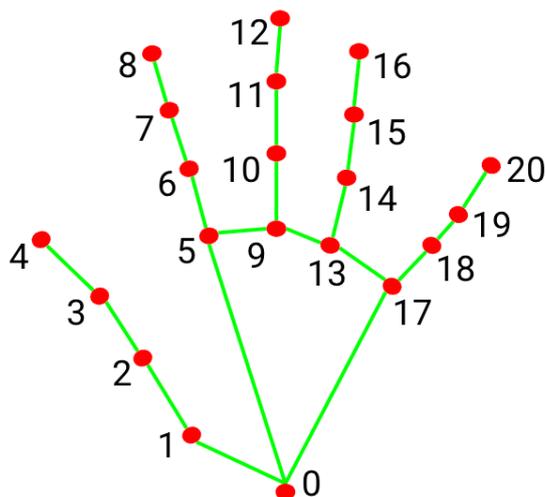


Ilustración 5: Información obtenida de librería de hand-tracking

Cada uno de estos puntos tiene una coordenada X, Y y Z, lo que se usará para hacer todos los cálculos. Cada una de las manos detectadas se compara con las posiciones de los gestos que tiene configurados el usuario. En caso de que haya detectado una coincidencia tantas veces como la configuración lo dictamine, se entenderá que el usuario está realizando un gesto. En los siguientes fotogramas y hasta considerar que se ha terminado de realizar el gesto, se guardarán las coordenadas del punto 0 (la muñeca). Una vez terminado el gesto, se calculará en qué dirección se ha movido la mano. Juntando la posición de los dedos y el movimiento de la mano se consigue un gesto completo, el cual tiene asociado una acción configurada por el usuario.

Es en este momento cuando el programa ha terminado de analizar la información que ha recibido de la imagen procesada, y sabrá si al gesto realizado le corresponde hacer realizar una acción, y cual es esta.

6.7 Diagrama de Secuencia

Los diagramas de secuencia se utilizan para detallar cómo se desarrollan los eventos para los diferentes actores en cada situación. Se diferencian 2 diagramas de secuencia. Por un lado, tenemos el de realizar la acción y por otro, añadir un gesto.

6.7.1 Diagrama de Realización de acción

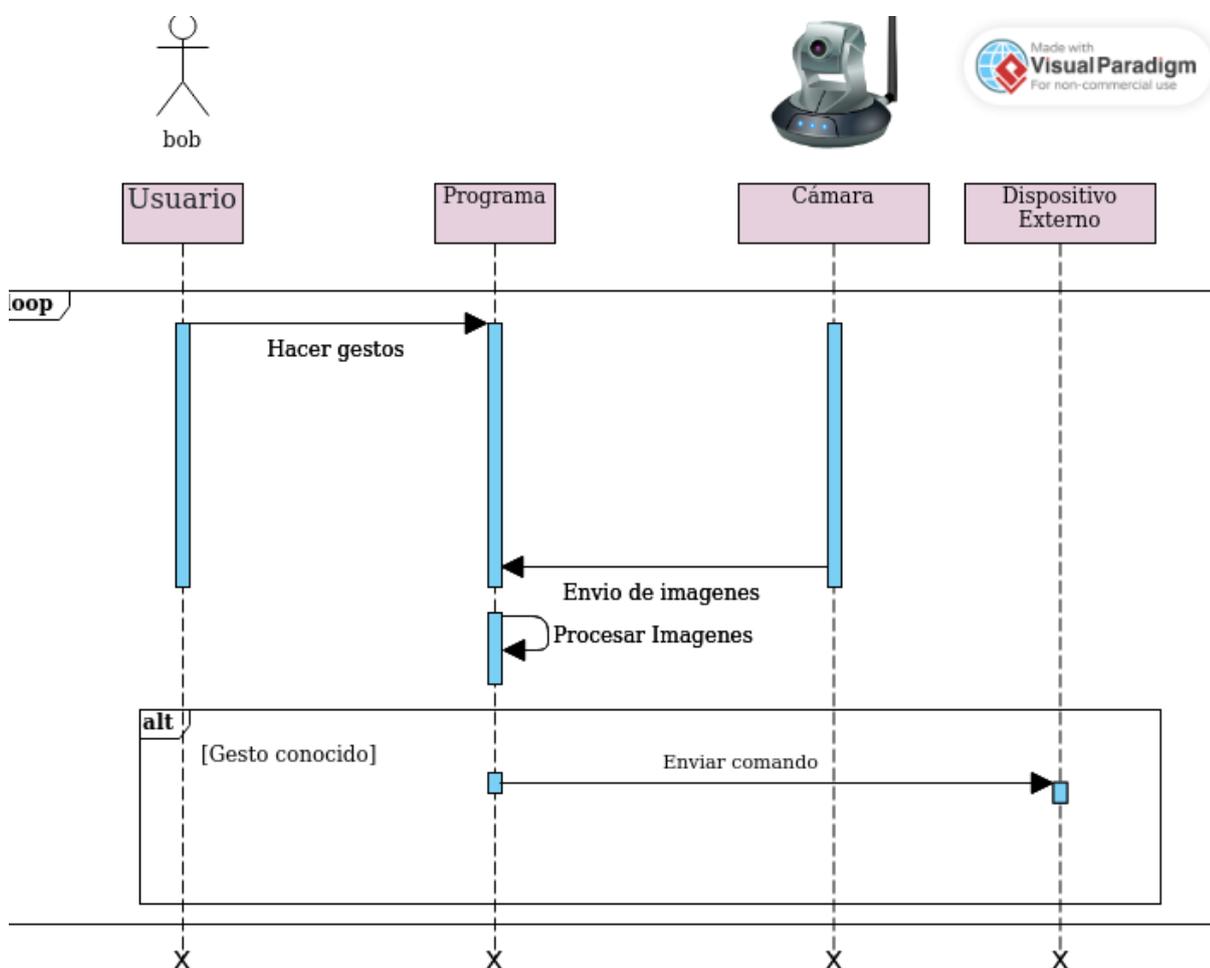


Ilustración 6: Diagrama de realización de acción

Encontramos 4 lifelines encabezados por: usuario, programa, cámara y dispositivo externo. La secuencia empieza con el gesto del usuario donde el programa es el que lo recibe a través de la cámara. Una vez recibido enviará al programa frame por frame todas las imágenes. El programa al recibir las imágenes procederá a procesarlas para captar el posible gesto realizado. En caso de reconocer el gesto conocido el programa enviará un comando al dispositivo con la acción que tiene

que realizar. En caso de no reconocer nada seguirá captando y analizando todas las imágenes que le envíe la cámara en bucle.

6.7.2 Diagrama de añadir acción

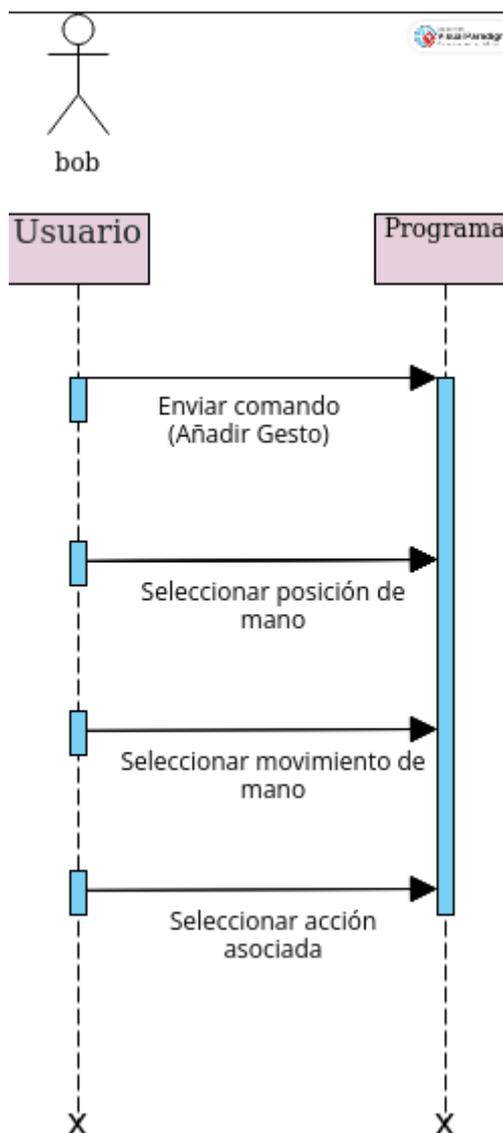


Ilustración 7: Diagrama de añadir acción

En esta secuencia solo tenemos 2 actores: el usuario y el programa. Esta secuencia está centrada en asociar una acción a un gesto así que el primer paso será enviar el comando pertinente al programa para avisar que se añadirá un gesto nuevo. Una vez enviado el comando se seleccionarán la posición de la mano y el movimiento que se realizará. Por último, una vez conocido el gesto se asocia con una acción nueva. De esta manera, el gesto quedará asociado a la acción seleccionada y volverá al principio donde se puede asociar una nueva acción.

6.8 Diagrama de Casos de Uso

Los diagramas de casos de uso son útiles para entender el comportamiento general del sistema. Para ello, partiendo del siguiente diagrama:

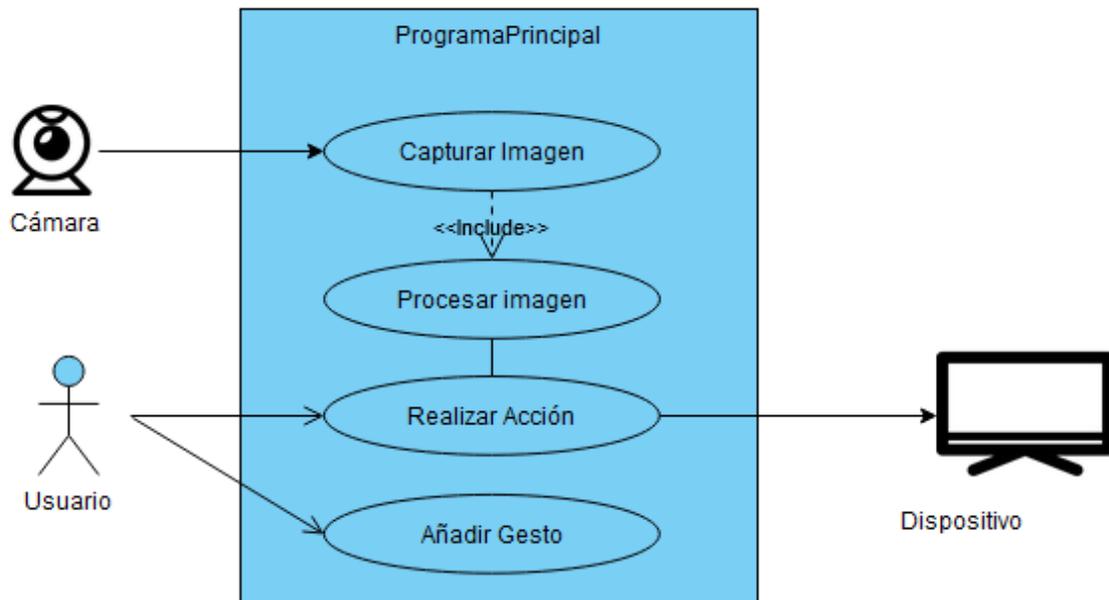


Ilustración 8: Diagrama de Casos de Uso

Se pueden diferenciar los siguientes actores: la cámara, el usuario y el dispositivo. La secuencia empieza con la captura de imágenes realizadas por la cámara. El programa se encargará de recoger todas estas imágenes y hacer un reconocimiento en todo momento para captar cualquier acción realizada con las manos. El usuario en cualquier momento realizará un gesto el cual será capturado y procesado por la cámara. El programa analizará el gesto realizado y en caso de que este coincida con alguno de los gestos registrados, realizará la acción asociada.

6.9 Diagrama de Actividad del sistema

El diagrama de actividad del sistema detalla paso a paso el flujo del funcionamiento del sistema. Para este proyecto se ha diseñado el siguiente diagrama de actividad:

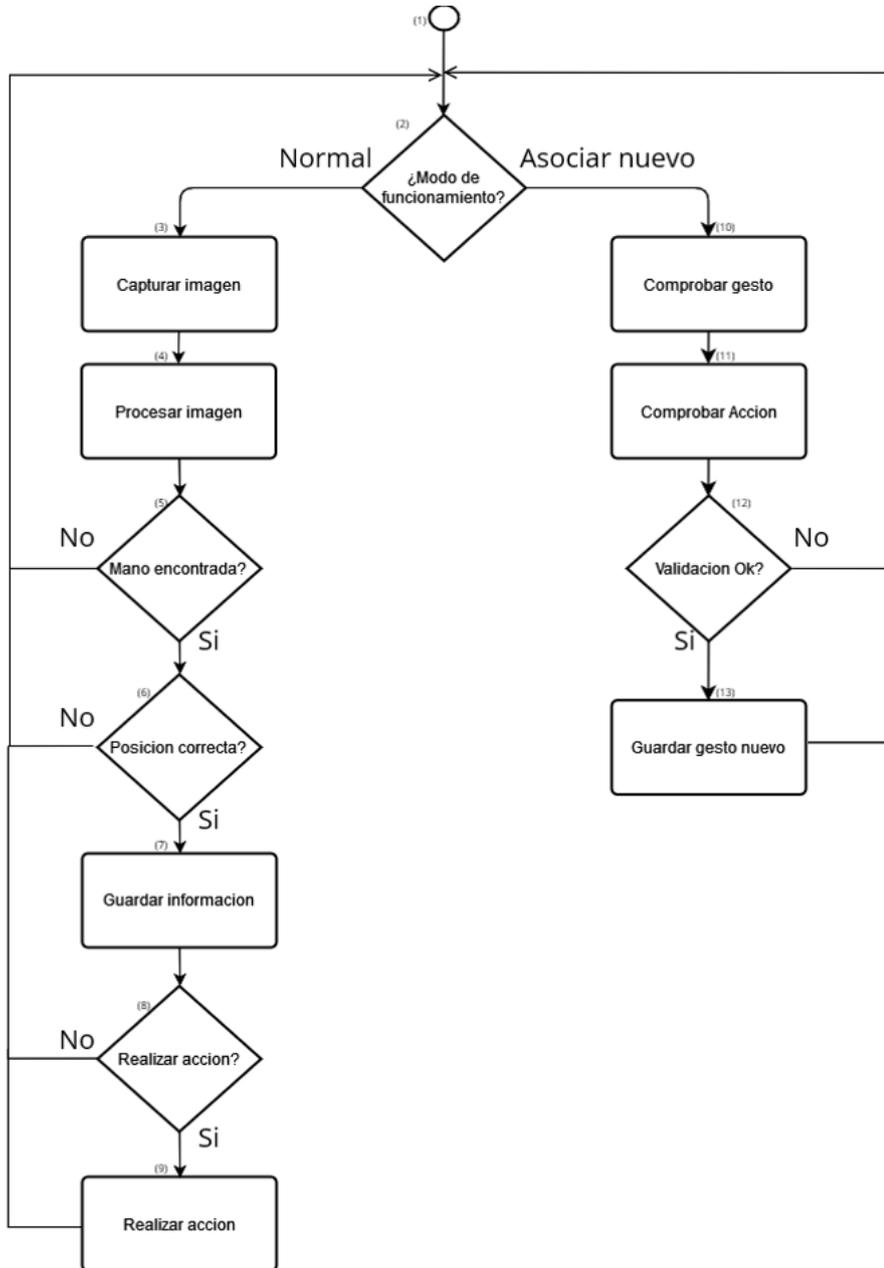


Ilustración 9: Diagrama de actividad

El diagrama de actividad muestra de qué manera reacciona nuestro proyecto ante las diferentes situaciones. Se pueden diferenciar dos modos de funcionamiento:

- **Modo configuración:** En caso de que el usuario quiera añadir una nueva relación gesto-acción, el programa recibirá mediante la interfaz gráfica la configuración de los dedos, el movimiento hacia donde se tiene que desplazar la mano y la acción asociada a estos. Una vez se desee guardar esta información, se validará que los datos sean correctos y en ese caso se actualizará la lista de gestos conocidos
- **Funcionamiento normal:** Mientras esté en este modo, estará en todo momento capturando y procesando imágenes mediante MediaPipe en busca de manos. En caso de que se encuentre en la imagen la presencia de alguna mano, se comprueba en qué posición están los dedos de esta. Se entiende que la mano está en una posición correcta cuando la colocación de los dedos coincide con las colocaciones configuradas por el usuario. Este proceso se realiza por cada mano detectada en cada fotograma. Es importante hacer esta comprobación ya que si no hay ninguna mano cuya posición de los dedos coincida con la configuración, no hay información que sea relevante para el funcionamiento del programa.

Para tener que realizar alguna acción tiene que suceder una de las siguientes situaciones:

1. No se han detectado manos durante un transcurso de tiempo definido por el usuario
2. No se ha detectado ninguna mano en la posición correcta durante un transcurso de tiempo definido por el usuario

De este modo, hacemos que el programa sea menos sensible a errores de la cámara.

7. Plan de trabajo

7.1 Introducción

Para realizar este proyecto hemos hecho uso de los siguientes paquetes de trabajos, cada uno con una serie de objetivos y tareas a cumplir. Este plan de trabajo está pensado para ser llevado a cabo por dos perfiles de ingenieros: uno senior encargado de tomar las decisiones de implementación y arquitectura y otro junior encargado del desarrollo de menor nivel. En este apartado se...

7.2 Equipo de trabajo

El equipo está formado por dos personas donde el estudiante asume dos roles diferentes que corresponden a actividades de distintas capacitaciones. En esta tabla se puede ver el rol de cada uno y los costes asociados.

Acrónimo	Nombre	Rol	Coste
AFV	Armando Ferro	Supervisor	45.3
GNT-ING	Germán Nasello	Ingeniero	
GNT-PRG	Germán Nasello	Programador	

Tabla 4: Equipos de trabajo

7.3 Paquetes de trabajo

Todo el proyecto ha sido planificado por paquetes de trabajo. Cada uno de estos tiene sus tareas específicas y sus entregables. Como se puede ver en la planificación del proyecto, al acabar un plan de trabajo empieza el siguiente. A excepción del paquete de trabajo de XXXX, al acabar un paquete de trabajo empieza el siguiente sin solaparse entre ellos. Esto sucede porque se ha plantado una metodología incremental donde cada tarea sigue desarrollando las tareas del plan de proyecto anterior. El paquete XXXX tiene una duración de todo el proyecto ya que se hace un trabajo constante de control de documentación y código.

de desarrollo, definición de lógica de programación y desarrollo del código. Para validar el prototipo se tienen que llevar a cabo pruebas unitarias que aseguren un correcto funcionamiento

Desarrollo de modelo avanzado

Es esta fase y partiendo de un prototipo funcional, se desarrolla el código en profundidad, añadiendo nuevos módulos y aplicando la lógica que relacione los gestos y las acciones. Es necesario empezar definiendo la lógica detallada de gestos y desarrollar el módulo de configuración. También se tiene que unificar módulos desarrollados con el prototipo y validar el código mediante pruebas predefinidas.

- Desarrollo de interfaz gráfica
- Como se ha explicado anteriormente, uno de los objetivos del proyecto es el desarrollo de una interfaz gráfica que permita al usuario interactuar con el sistema. Para ello se empieza por diseñar una interfaz que luego será desarrollada. Una vez la interfaz esté integrada con el programa, se llevarán a cabo pruebas para comprobar el correcto funcionamiento.

Análisis y mejoras

Por último, una vez se ha desarrollado el código y el producto cumple con los objetivos, la última fase del proyecto se basa en la recepción de feedback del uso de la aplicación por parte de gente externa al proyecto y mejora del código. Se reciben las opiniones de usuarios que usen la aplicación. Estos datos se analizarán para aplicar las correcciones y mejoras correspondientes.

Gestión de proyecto

Este plan de trabajo durará lo que dure el proyecto ya que en el se llevaran a cabo tareas que aseguren un correcto desarrollo del proyecto. Estas tareas son el control del proyecto y la documentación de este.

7.4 Materiales

En el desarrollo del sistema de reconocimiento de gestos, se utilizan diversos materiales y herramientas esenciales para asegurar el éxito del proyecto. A continuación, se detallan los materiales principales:

- Cámara web
- Ordenador de desarrollo
- Material de oficina

7.5 Entregables

Por cada paquete de trabajo, se definen también entregables que han de ser verificados al finalizar todas las tareas. Estas son de gran importancia para el correcto control del desarrollo del proyecto:

Entregable	Paquete de trabajo	Descripción
Documento de especificaciones del proyecto	PT1	Documento que detalla los objetivos y alcance del proyecto, además de un diseño preliminar de la arquitectura
Prototipo funcional	PT2	Primera versión del código a modo de prototipo
Prototipo avanzado	PT3	Código con funcionalidades avanzadas
Interfaz gráfica	PT4	Integración de la interfaz gráfica en el código previo
Código revisado	PT5	Código final con mejoras aplicadas

Tabla 5: Entregables de paquetes de trabajo

7.6 Hitos

Los hitos del proyecto son:

Hito	Fecha	Título	Descripción
1	01/06/2023	Inicio proyecto	Comienzo de planificación y recursos.
2	01/11/2023	Revisión de diseño	Evaluación del prototipo
3	15/02/2024	Pruebas finales	Verificación de funcionalidad y calidad.
4	30/03/2024	Fin de proyecto	Finalización del trabajo.

Tabla 6: Hitos del proyecto

8. Presupuesto

En este apartado, se presentará un desglose detallado de los aspectos económicos relacionados con el proyecto. Es importante señalar que los números mencionados reflejan únicamente los costos asociados con el diseño y desarrollo del proyecto. En estos gastos no se contempla el IVA.

Para hacer este análisis económico, los gastos se han dividido en los siguientes conceptos:

- **Recursos humanos:** costos del tiempo de trabajo del equipo interno.
- **Gastos:** costes de material usado en el proyecto que no se ha podido reusar.
- **Amortizaciones:** costos de activos, como equipos o software, usados en el proyecto.

8.1 Recursos humanos

En este se definen los dos miembros principales del equipo de trabajo: ingeniero y programador. El primero se encargará de las decisiones vitales para el correcto desarrollo del proyecto y el ingeniero junior llevará a cabo las tareas de más bajo nivel. Además, abra un encargado del proyecto que supervisara su gestión.

Concepto	N.º horas	Precio hora	Ratio	Total (€)
AFV	28	47,2 €/hora	46,20%	1.321,60 €
GNT - ING	268	17,60 €/hora	48,97%	8.710,00 €
GNT -PRG	284	32,50 €/hora	4,82%	4.998,40 €

Tabla 7: Costes de recursos humanos

8.2 Gastos

Dentro de los gastos asociados al proyecto, se encuentran los materiales de oficina y suministros de papelería.

Concepto	N.º unidades	Precio Unitario (€)	Total (€)
Fungibles	–	47.9	47,90 €

Tabla 8: Costes de gastos

8.3 Amortizaciones

Para este proyecto la única amortización a tener en cuenta será el uso los ordenadores que se usarán para a lo largo del desarrollo:

Concepto	N.º unidades	Precio Unitario (€)	Vida útil (años)	Uso (años)	Total (€)
Ordenador	2	699,00 €	5	1	280,00 €

Tabla 9: Costes de amortizaciones

8.4 Costes del proyecto

Teniendo en cuenta las partidas individuales, los costes asociados al desarrollo del proyecto y aplicando que los costes indirectos han supuesto un 5 % del coste total del proyecto:

Partida	Coste (€)
Horas Internas	15.030,00 €
Gastos	47,90 €
Amortizaciones	280,00 €
Costes Indirectos (5%)	767,89 €
Total	16.125,79 €

Tabla 10: Costes del proyecto

Analizando la tabla anterior, podemos ver que la partida con mayor coste es la de Horas Internas, alcanzando un total de quince mil treinta euros. Esto va acorde con la planificación del proyecto, ya que ahí se explica que participan un total de dos ingenieros distintos.

Por otra parte, la partida de amortizaciones y gastos no presentan un gasto significativo, ya que al usar alternativas de código abierto se han evitado el pago de licencias. Además, ha sido planteado como un proyecto de bajo coste.

El presupuesto de costes indirectos se ha destinado a cubrir aquellos gastos que no se pueden atribuir directamente a una partida concreta. Teniendo en cuenta que los costes indirectos han supuesto setecientos sesenta y siete coma ochenta y nueve euros, el coste total de desarrollo de este proyecto asciende a dieciséis mil ciento veinticinco coma setenta y nueve euros

9. Conclusiones

Este proyecto supone un avance significativo en la interacción humano-maquina, haciendo que la comunicación sea más intuitiva ya que permite entender los gestos realizados por el usuario convertirlos en acciones. Este era el objetivo principal del proyecto, y ha sido cumplido, ya que el sistema es capaz de reconocer la posición de los dedos de la mano, y en qué dirección se desliza.

Otro de los objetivos es el diseño e implementación de una interfaz gráfica que permita la personalización de los gestos y configuración del funcionamiento del sistema. Este también ha sido cumplido, pero no con tanto éxito como el anterior ya que, si bien la interfaz gráfica es funcional, no es tan intuitiva para usuarios no familiarizados con el uso de tecnología.

Tal y como se ha mencionado en el alcance del proyecto, este no ha sido diseñado para ser un producto final, sino que una prueba de concepto, demostrando que es posible cambiar el modo en el que interactuamos con la tecnología y aún hay espacio para la innovación en la comunicación humano-máquina.

En conclusión, podemos decir que el proyecto ha sido un éxito puesto que se han alcanzado los objetivos y se ha desarrollado una solución general que abre paso a nuevos estudios en el campo del lenguaje natural, pero hay que reconocer que es necesario un gran trabajo de desarrollo y adaptación para convertir este prototipo en un software con usos prácticos y económicamente rentable

ANEXO I

Pliego de condiciones

El presente documento detalla el pliego de condiciones para el desarrollo y explotación de este proyecto. Esta solución utiliza tecnologías de visión por computadora e inteligencia artificial y permite el control de dispositivos mediante gestos, entendiendo los gestos que hace el usuario y realizando su acción asociada. A continuación, se presentan los requisitos y condiciones esenciales para la explotación y uso del sistema.

- **Explotación de la solución**

Como se ha comentado, este proyecto no ha sido desarrollado como un producto final, sino como un prototipo que abra nuevas vías de desarrollo en el área de la interacción humano-maquina. Cualquier entidad o persona interesada en explotar esta tecnología con la intención de convertirlo en un producto comercial, debe ponerse en contacto con el responsable del proyecto para discutir los términos y condiciones de licenciamiento y colaboración.

Plan de pruebas

Para probar el correcto funcionamiento del sistema, se ha llevado a cabo el siguiente plan de pruebas:

Prueba	Superada
Es capaz de detectar manos	Si / No
Es capaz de diferenciar la posición de la mano	Si / No
Es capaz de saber en que dirección se mueve la mano	Si / No
Realiza la acción asociada al gesto	Si / No
Se pueden modificar las relaciones gesto-acción	Si / No
Se pueden modificar los parámetros de funcionamiento del sistema	Si / No

ANEXO II

Se deja un enlace al repositorio del código: <https://github.com/GermanNasello/GestureWizard>

ANEXO III

A lo largo del desarrollo del proyecto, se han identificado ciertos riesgos. Si bien no son importantes en este momento ya que está el trabajo finalizado, se explican a continuación para dar un punto de vista más amplio Repasar en busca de fallos o incongruencias

Privacidad

Este es el riesgo principal a la hora de usar esta solución. Esto se debe a que para el reconocimiento de gestos es necesario poner una cámara delante que capte imágenes en todo momento. Esto puede permitir a un atacante acceder a la cámara comprometiendo la privacidad de los usuarios.

Acceso no autorizado

El acceso y el control de los dispositivos es un aspecto a tener en cuenta en aplicaciones de este tipo. Siempre existe el riesgo de que personas no autorizadas puedan acceder a la aplicación y hagan un uso indebido de la aplicación. Por esta razón las medidas de autenticación y autorización deben ser adecuadas.

Dispositivos externos

El entorno puede afectar directamente con el correcto funcionamiento de la aplicación así que es un factor a tener en cuenta. La aplicación funciona gracias a una cámara que hace el reconocimiento del gesto. La cámara es un dispositivo externo que el usuario que use la aplicación deberá tener. Es ese el motivo por el que el riesgo variará en función al usuario. Si la persona que usa esta aplicación está haciendo uso de una cámara con mejores prestaciones que otra persona que está usando otra de menor calidad es muy probable que la primera persona tenga una mejor experiencia de usuario.

Dependencia del entorno

El entorno en el que se use la aplicación es otro limitante del mismo. La importancia de la cámara antes mencionada hace que el entorno en el que se use también lo sea. Es decir, si el usuario se encuentra en un sitio muy mal iluminado no será posible captar con precisión todos los gestos que se desean ya sea por un fallo de reconocimiento o por un error del sistema.

Para la evaluación de los riesgos se realizará una matriz de impacto probabilidad donde se colocará cada riesgo en la tabla con su respectiva probabilidad de suceso. Una vez clasificada se procederá a tomar una medida en base a su clasificación.

En la siguiente tabla encontramos 3 categorías divididas por colores indicándonos la gravedad del riesgo:

- Verde poco riesgo o poco probable
- Naranja riesgo o probabilidad moderados
- Rojo riesgo alto o de alta probabilidad

	Despreciable	Marginal	Moderado	Crítico	Catastrófico
Cierto					
Probable					
Posible					
Improbable					
Excepcional					

Privacidad

Como se ha comentado es el principal riesgo de la aplicación y a pesar de que no tiene una gran de probabilidad de suceso se ha decidido clasificarla de la siguiente manera:

	Despreciable	Marginal	Moderado	Crítico	Catastrófico
Cierto					
Probable					
Posible					
Improbable					
Excepcional					

Debido a que no podemos eliminar por completo este riesgo se ha decidido tomar una serie de medidas para intentar paliar este riesgo lo máximo posible sin que afecte al correcto funcionamiento de la aplicación. Las medidas que se ha creído oportunas son:

- Apagado opcional de la cámara
- No peticiones web local
- Va frame por frame

Acceso no autorizado

En este caso, el riesgo no es tan significativo como el anterior y su probabilidad de suceso es similar así que se ha decidido clasificarlo de la siguiente manera:

	Despreciable	Marginal	Moderado	Crítico	Catastrófico
Cierto					
Probable					
Posible					
Improbable		X			
Excepcional					

Cómo se observa se encuentra en la región verde así que el impacto que tiene dentro de la aplicación es mínimo. Es por esa razón que no se han tomado grandes medidas, ni se han destinado muchos recursos a reducir su impacto o probabilidad.

Dispositivos externos

En el caso de los dispositivos externos la probabilidad sigue rondando los mismos valores mientras que el impacto disminuye quedando la tabla de la siguiente manera:

	Despreciable	Marginal	Moderado	Crítico	Catastrófico
Cierto					
Probable					
Posible					
Improbable	X				
Excepcional					

Esto se debe a que hoy en día gracias a los avances de la tecnología se puede tener una calidad decente sin un desembolso muy grande de dinero. Esto consigue que un gran porcentaje de los usuarios tengan una cámara lo suficientemente buena como para no notar diferencias a la hora de usar la aplicación.

De nuevo, como el anterior riesgo se encuentra dentro de la zona verde por tanto no es necesario tomar medidas muy drásticas.

Dependencia del entorno

Este último riesgo sigue teniendo poco impacto en este proyecto, pero la probabilidad de impacto aumenta un poco.

	Despreciable	Marginal	Moderado	Crítico	Catastrófico
Cierto					
Probable					
Posible	X				
Improbable					
Excepcional					

A pesar de que tenga una probabilidad un poco más elevada en comparación con el resto no deja de estar en la zona verde. Es por eso que no es necesario el uso de soluciones para este riesgo.

Referencias

- [1]. Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang eta Matthias Grundmann. 2020. «MediaPipe Hands: On-device Real-time Hand Tracking». *Google Research*,1-5.
- [2] *Hand landmarks detection guide for Python*. (s.f.). Google For Developers. https://developers.google.com/mediapipe/solutions/vision/hand_landmarker/python
- [3] Fredrik Lundh. 2003. «An Introduction to Tkinter».
- [4] Alexander Mordvintsev & Abid K. 2017. «OpenCV-Python Tutorials Documentation».
- [5] Gamez, M. J. (2022, 24 mayo). *Objetivos y metas de desarrollo sostenible - Desarrollo Sostenible*. Desarrollo Sostenible. <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>