

junio, 2024

**SIGMA: Sistema de Interacción Gestual para la Mejora de la
Accesibilidad
Interacción Gestual**

Alberto García Martín



**VNiVERSiDAD
D SALAMANCA**

**Máster en Sistemas Inteligentes 2023–2024
Universidad de Salamanca**

Índice

1. Prototipo	1
1.1. Descripción del prototipo	1
1.2. Evaluación con usuarios	5
2. Memoria	7
2.1. Resumen	7
2.2. Antecedentes y estado actual del tema	8
2.3. Objetivos generales y específicos	10
2.4. Metodología	12
2.5. Plan de trabajo y cronograma	14
2.6. Recursos humanos, materiales y de equipamiento	19
2.7. Resultados e impacto esperados	21
2.8. Uso de IA	22
Referencias	22

Índice de figuras

1.	Ejemplos de los gestos “Point”, “Drag” y “Loop” de Alashhab et al. [3].	2
2.	Ejemplos de los gestos “Like”, “Dislike” y otros del conjunto HaGRID [4].	2
3.	Captura de vídeo de la extensión ejecutando el modelo.	3
4.	Ejemplo de cómo se muestra el cursor virtual en una página web. . .	4
5.	Tutorial de los gestos que se pueden realizar con la extensión.	5
6.	Diagrama de Gantt de actividades y tareas del proyecto.	18

Indice de tablas

1.	Hiperparámetros utilizados para entrenar el modelo.	2
2.	Actividades y tareas del proyecto.	14
3.	Actividad 1: Preparación e inicio del proyecto.	15
4.	Actividad 2: Investigación y recolección de datos.	15
5.	Actividad 3: Desarrollo del modelo de aprendizaje profundo.	16
6.	Actividad 4: Integración del modelo con el sistema.	16
7.	Actividad 5: Pruebas y validación.	17
8.	Actividad 6: Optimización y rendimiento.	17
9.	Actividad 7: Documentación y Publicación.	18

1. Prototipo

1.1. Descripción del prototipo

Para este proyecto, se ha desarrollado un prototipo de un sistema de navegación web mediante control gestual. Este prototipo en cuestión es una extensión de navegador web que permite al usuario controlar un cursor virtual mediante gestos realizados con la mano, permitiendo así interactuar con las páginas web sin necesidad de utilizar un ratón.

Desarrollo del modelo de identificación de gestos

La detección de la mano y los gestos se realiza mediante un modelo de aprendizaje profundo, que es capaz de identificar la posición de la mano en la imagen capturada por la cámara web y clasificar los gestos realizados por el usuario. Este modelo se ha desarrollado utilizando la biblioteca MediaPipe de Google [1], que proporciona una implementación de un modelo de detección de manos y gestos pre-entrenado y permite extenderlo para adaptarlo a las necesidades del proyecto. Este modelo posteriormente puede ser ejecutado en diferentes plataformas, entre ellas el navegador web, mediante la biblioteca TensorFlow.js [2].

Para hacer que la red neuronal aprenda a reconocer los gestos que se quieren utilizar en el prototipo, se han utilizado dos conjuntos de datos de gestos filtrados para que solo contengan los gestos de interés. En total serán cinco gestos diferentes, con una categoría adicional para los gestos no etiquetados.

El primer conjunto de datos fue recopilado por Alashhab et al. [3] y contiene cuatro gestos diferentes, además de una categoría con otros gestos no etiquetados y otra sin manos en la imagen. También contiene un conjunto de imágenes sintéticas generadas por ordenador y un conjunto real. De este conjunto de imágenes se han seleccionado tres gestos del conjunto de imágenes reales que en el trabajo original denominan “Point”, “Drag” y “Loop”, en la figura 1 se pueden ejemplos de estos gestos. El gesto de “Pinch” es el único gesto etiquetado de este conjunto que no se ha utilizado en el prototipo, debido a que la tasa de acierto del modelo en la detección de este gesto era muy baja.

El segundo conjunto de datos empleado es HaGRID, recopilado por Kapitanov et al. [4] y contiene 18 clases de gestos diferentes. Sin embargo, para este proyecto solo se han utilizado dos gestos para complementar a los anteriores, el de “Like” y el de “Dislike”. En la figura 2 se pueden ver ejemplos de estos gestos, junto a los otros gestos que no se han utilizado en el prototipo. Para reducir el tamaño en disco del conjunto de datos, se ha utilizado la versión ligera de HaGRID, que contiene imágenes de una menor resolución, de 512×512 píxeles. Además, para mantener una distribución balanceada de los gestos, se ha realizado un submuestreo aleatorio de las imágenes de todas las clases, para que cada una tenga un número igual de imágenes, que es de 2000 imágenes por clase.

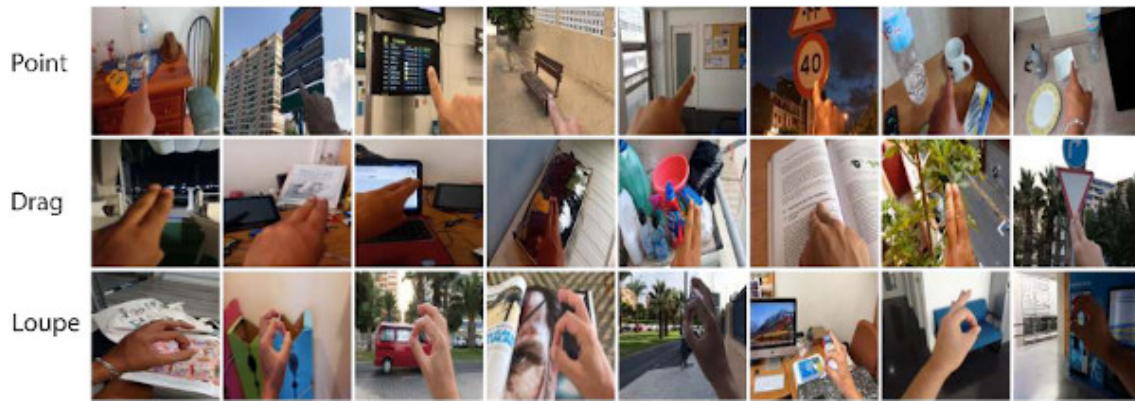


Figura 1: Ejemplos de los gestos “Point”, “Drag” y “Loop” de Alashhab et al. [3].

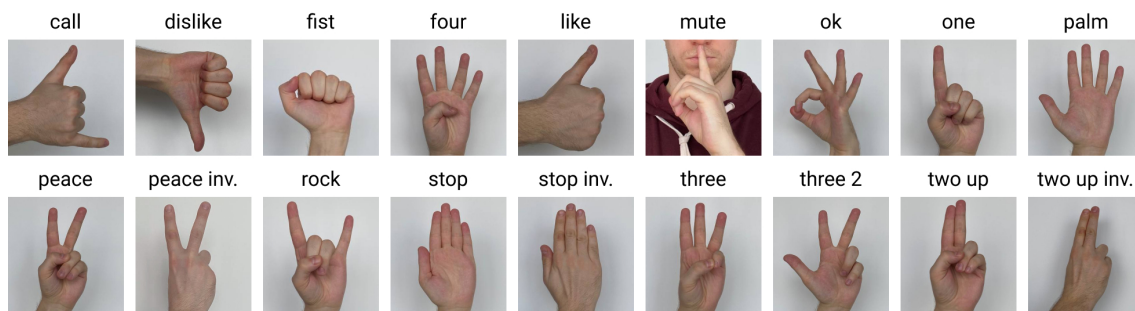


Figura 2: Ejemplos de los gestos “Like”, “Dislike” y otros del conjunto HaGRID [4].

Una vez se han seleccionado las imágenes que conforman el conjunto de datos con el que se entrenará el modelo, se cargan en memoria y se barajan, para posteriormente dividir las en tres conjuntos, uno de entrenamiento, uno de validación y otro de test, con una proporción de 80 %, 10 % y 10 % respectivamente. Posteriormente, se ha procedido a entrenar el modelo con los hiperparámetros que aparecen en la tabla 1, que han sido seleccionados mediante validación cruzada sobre el conjunto de validación. El modelo resultante ha sido evaluado con el conjunto de test, obteniendo una exactitud media del 94.62 %, y finalmente se ha exportado a un modelo de TensorFlow Lite para ser ejecutado en el prototipo.

Hiperparámetro	Valor
Tasa de aprendizaje	0.001
Tamaño del lote	16
Número de épocas	20
<i>Learning rate decay</i>	1
Gamma	2
Tasa de <i>dropout</i>	0.1

Tabla 1: Hiperparámetros utilizados para entrenar el modelo.

Desarrollo de la extensión de navegador

La extensión de navegador para el prototipo se ha desarrollado utilizando el marco de trabajo WXT ¹, que permite crear extensiones para navegadores web de forma sencilla y rápida, simplificando el proceso de desarrollo. Esta extensión se encarga de capturar el flujo de vídeo de la webcam del dispositivo, procesarlo con el modelo de detección de manos y gestos y enviar los resultados al navegador web para que se puedan utilizar en la página web que se esté visualizando. Además, la extensión permite al usuario activar y desactivar el control gestual en cualquier momento.

Para que la extensión pueda capturar el vídeo de la webcam y procesarlo en tiempo real, se ha utilizado la API de MediaStream de JavaScript, que permite acceder a los dispositivos multimedia del dispositivo, como la cámara. Para poder hacer uso de esta API, es necesario solicitar permiso al usuario para acceder a los dispositivos multimedia, y posteriormente se tiene que mantener abierta una ventana en el navegador para poder capturar el vídeo y mostrarlo al usuario.

El modelo de detección de manos y gestos se carga en la extensión cuando se activa el control gestual, y se ejecuta en cada fotograma del vídeo capturado por la cámara. El modelo devuelve la posición de la mano en la imagen y la clase del gesto realizado por el usuario, esto se muestra en la captura de vídeo que se visualiza en la ventana de la extensión, como se puede ver en la figura 3.

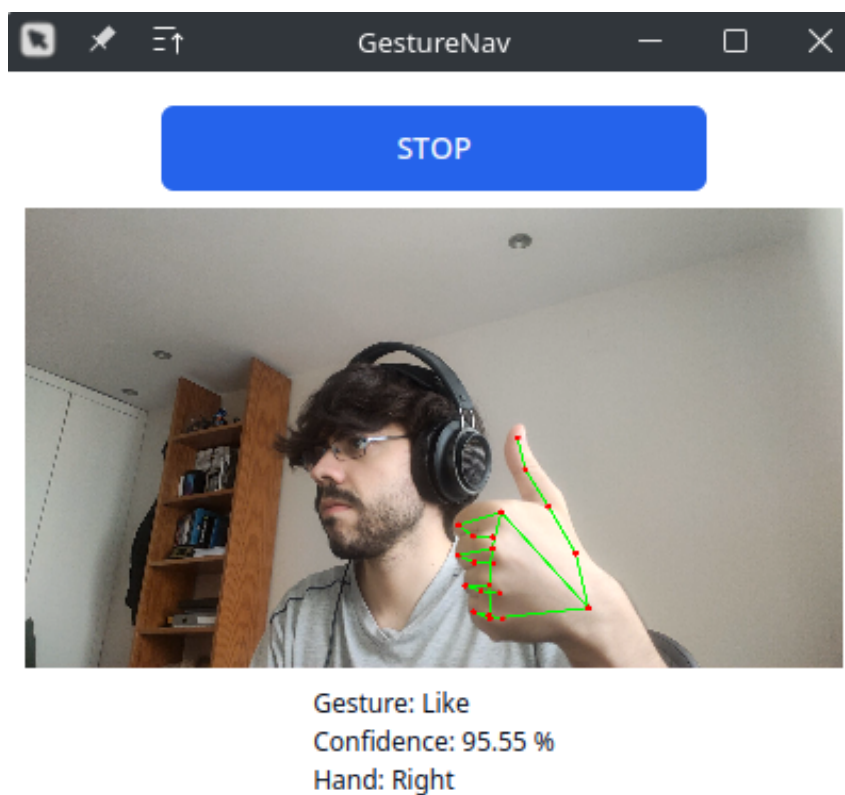


Figura 3: Captura de vídeo de la extensión ejecutando el modelo.

¹<https://wxt.dev/>

Los resultados del modelo se envían a la página web con la que está interactuando el usuario mediante mensajes. Estos mensajes son capturados por un script que se ha insertado en la página web, y que se encarga de interpretar los resultados y dibujar el cursor virtual en la posición que corresponda. Además, este script también se encarga de enviar eventos de ratón a la página web, como clics o desplazamientos, para que el usuario pueda interactuar con ella. En la figura 4 se puede ver un ejemplo de cómo se muestra el cursor virtual en una página web.

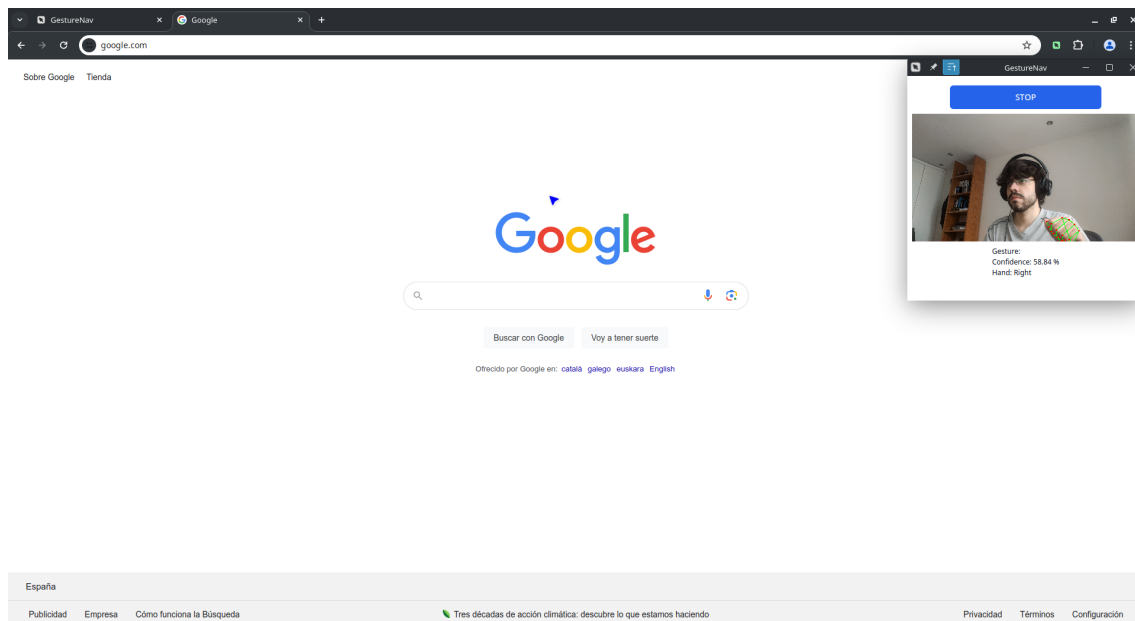


Figura 4: Ejemplo de cómo se muestra el cursor virtual en una página web.

El cursor virtual se puede mover por la página web mediante dos formas distintas, por defecto se utiliza un modo de movimiento relativo, donde el cursor se mueve en la dirección en la que esté la mano del usuario con respecto al centro de la imagen capturada por la cámara. Este punto de referencia se puede cambiar al punto donde se encuentre la mano realizando un gesto de OK. También se puede cambiar al modo de movimiento absoluto, donde el cursor se mueve a la posición de la mano en la imagen capturada por la cámara, este modo está activo cuando el usuario está apuntando con el dedo índice y corazón. La extensión hace *scroll* de forma automática cuando el cursor se acerca a los bordes de la ventana del navegador.

Para realizar clics en la página web, el usuario tiene que levantar el dedo índice, para navegar a la página anterior, el usuario tiene que realizar un gesto de *dislike*, mientras que para navegar a la página siguiente, el usuario tiene que realizar un gesto de *like*. Al instalar la extensión, se muestra al usuario la información de los gestos que puede realizar y ejemplos de cómo hacerlos. Esto se puede ver en la figura 5.

El cursor cambia de color cuando se detecta un gesto, para indicar al usuario que se ha reconocido correctamente, además, cuando se utiliza la mano izquierda, el cursor se invierte para mostrar que se está reconociendo esa mano. Cuando el usuario cambia de pestaña o ventana, la extensión se desactiva automáticamente para la página anterior y se activa para la nueva página.

Welcome to GestureNav

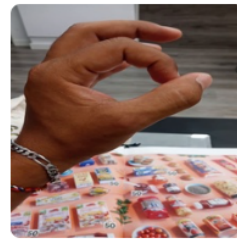
Control your cursor and navigate websites with hand gestures! Click the toolbar icon or the button below to enable the extension and see the cursor appear on the website. In "pull" mode, the cursor moves based on your hand's position relative to the screen's center. You can reset the center point with an "ok" gesture. Switch to "pointer" mode with a "two" gesture to control the cursor directly with your hand. Use a "one" gesture to click, a "like" gesture to go forward in history, and a "dislike" gesture to go back.



One Gesture: Click



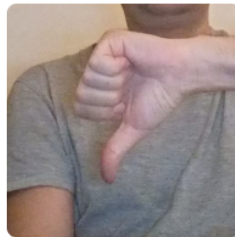
Two Gesture: Switch to Pointer Mode



OK Gesture: Recalibrate Center



Like Gesture: Go Forward in History



Dislike Gesture: Go Back in History

ENABLE WEBCAM

Figura 5: Tutorial de los gestos que se pueden realizar con la extensión.

1.2. Evaluación con usuarios

El proceso de evaluación del prototipo se ha llevado a cabo con tres usuarios distintos, que han probado la extensión en diferentes páginas web y han realizado distintas tareas con ella. Los usuarios han tenido que completar una serie de tareas, con el objetivo de obtener un feedback detallado y realista sobre la usabilidad y eficacia del sistema de control gestual implementado.

Este proceso comenzó con una breve introducción al prototipo y una explicación verbal de los gestos que se pueden realizar. A cada usuario se le dio una descripción clara de cómo realizar los gestos para mover el cursor virtual en las direcciones deseadas, así como para realizar acciones de navegación. Esta explicación inicial tenía como objetivo proporcionar a los usuarios un conocimiento básico de cómo interactuar con el sistema.

Tras esta introducción, se permitió a los usuarios utilizar la extensión por sí mismos. Esta prueba fue útil para observar el proceso de adaptación de los usuarios, aunque evidenció ciertas dificultades iniciales en todos los casos. Los usuarios mostraron dificultades para entender cómo utilizar la extensión de manera efectiva al principio,

necesitando un período de tiempo para adaptarse a los controles gestuales. Esto puede indicar que una demostración detallada en vivo, donde se mostrara visualmente cómo realizar los gestos y cómo interactuar con la interfaz, podría haber facilitado una adaptación más rápida y una comprensión más clara de los controles por parte de los usuarios.

Durante las pruebas de los usuarios se observó que el reconocimiento de gestos no era siempre consistente. Hubo momentos en que los gestos no fueron reconocidos correctamente, lo que generó frustración entre los usuarios. Esta inconsistencia en el reconocimiento de gestos indica que el modelo utilizado podría necesitar ajustes adicionales para mejorar su precisión y mejorar la experiencia del usuario.

Un aspecto destacado durante la evaluación fue la experiencia del usuario de mayor edad, quien indicó sentirse cansado después de realizar los gestos necesarios para mover el cursor. Los gestos requieren hacer movimientos del brazo en direcciones horizontales y verticales para mover el cursor, lo que explica la fatiga experimentada por el usuario. Este hallazgo sugiere que el sistema de control gestual podría no ser adecuado para todos los usuarios, especialmente aquellos con limitaciones físicas o problemas de movilidad. Por lo tanto, es importante considerar la diversidad de los usuarios al diseñar un sistema de interacción gestual y proporcionar alternativas para aquellos que puedan tener dificultades con los gestos físicos.

En general, los usuarios expresaron una actitud positiva hacia la extensión y consideraron que era una forma innovadora de interactuar con las páginas web. Sin embargo, también señalaron que la extensión podría beneficiarse de mejoras en la precisión del reconocimiento de gestos y en la facilidad de uso. Además, los usuarios sugirieron la inclusión de ajustes personalizables para adaptar tanto la sensibilidad de los gestos como el tipo de gesto asociado a cada acción. Esta evaluación ha proporcionado información valiosa sobre la usabilidad y eficacia del prototipo, así como ideas para futuras mejoras que se podrían tener en cuenta en versiones posteriores.

2. Memoria

En esta memoria se presenta la propuesta para el proyecto SIGMA, un sistema de interacción gestual para la mejora de la accesibilidad.

2.1. Resumen

En el contexto actual de la interacción persona-ordenador, existe un creciente interés en desarrollar métodos de control más intuitivos y naturales. El problema a resolver es la dependencia de dispositivos físicos como teclados y ratones para interactuar con ordenadores, lo que puede resultar incómodo y limitado para ciertos usuarios. El objetivo principal de este proyecto de investigación es desarrollar un sistema de control gestual que permita a los usuarios controlar un ordenador mediante gestos sin necesidad de utilizar otros dispositivos adicionales, más allá de una cámara.

La metodología propuesta se basa en un modelo de aprendizaje profundo capaz de detectar y reconocer gestos realizados con la mano en tiempo real. Este sistema debería ser capaz de identificar una amplia variedad de gestos con una precisión y velocidad aceptables, para proporcionar una experiencia de usuario fluida y sin interrupciones. El sistema se implementará como un programa de escritorio que se ejecutará en segundo plano y enviará eventos de ratón al sistema operativo, permitiendo al usuario controlar el ordenador mediante gestos. También deberá ser altamente configurable y personalizable, para adaptarse a las preferencias y necesidades de cada usuario.

Los resultados esperados incluyen un sistema de control gestual funcional y preciso, que permita a los usuarios interactuar con un ordenador sin contacto. Este sistema debería ser fácil de usar y accesible para una amplia variedad de usuarios, incluidos aquellos con discapacidades físicas. Además, se espera que este sistema mejore la experiencia del usuario en la navegación web y otras aplicaciones informáticas.

En conclusión, este proyecto pretende demostrar la viabilidad y utilidad de los sistemas de control gestual en aplicaciones del mundo real, y explorar su potencial para mejorar la interacción persona-ordenador en entornos cotidianos.

2.2. Antecedentes y estado actual del tema

El campo de la interacción persona-ordenador ha sufrido una evolución significativa desde sus inicios, pasando de los sistemas de línea de comandos a las interfaces gráficas de usuario y, más recientemente, a las interfaces táctiles y gestuales. Estas últimas han ganado popularidad en los últimos años debido a su capacidad para proporcionar una experiencia de usuario más intuitiva.

La interacción gestual, en particular, ha despertado un gran interés en la comunidad de investigación y desarrollo, ya que ofrece la posibilidad de controlar dispositivos y sistemas mediante gestos realizados con las manos u otras partes del cuerpo, sin necesidad de utilizar dispositivos de entrada tradicionales como ratones o teclados.

Los sistemas de control por gestos han sido explorados en diversas formas durante las últimas décadas. Uno de los primeros ejemplos de interacción gestual fue el guante de Thomas G. Zimmerman y Jaron Lanier [5] en la década de 1980, que permitía a los usuarios controlar un entorno virtual mediante gestos realizados con guantes equipados con sensores de movimiento. Desde entonces, se han desarrollado numerosos sistemas de control gestual, como la Kinect de Microsoft, que permiten a los usuarios interactuar con dispositivos y sistemas de forma natural y sin contacto.

Los avances en la última década en el campo del aprendizaje profundo y en el de la visión por ordenador han permitido el desarrollo de sistemas de detección de gestos muy precisos y eficientes que no requieren hardware especializado, como cámaras 3D o sensores de movimiento. Incluso en entornos desafiantes, los algoritmos modernos, como las redes neuronales convolucionales, son capaces de detectar y reconocer gestos con una precisión y velocidad más que aceptable. Esto permite la implementación de sistemas de control gestual en tiempo real en una amplia variedad de dispositivos, mientras dispongan de una cámara.

Sin embargo, a pesar de los avances en la detección de gestos, las implementaciones de sistemas de control gestual en aplicaciones del mundo real siguen siendo limitadas. La mayoría de los sistemas de control gestual existentes se centran en aplicaciones de entretenimiento, como videojuegos o realidad virtual, o en aplicaciones altamente especializadas. La integración de sistemas de control gestual en aplicaciones de uso diario, como puede ser la navegación web, sigue siendo un área de investigación activa y un desafío técnico.

La precisión y la fiabilidad del reconocimiento de gestos son fundamentales para la adopción generalizada de los sistemas de control gestual en aplicaciones del mundo real. Los sistemas de control gestual deben ser capaces de reconocer una amplia variedad de gestos de forma extremadamente precisa y en tiempo real, para proporcionar una experiencia de usuario fluida y sin problemas. Además, los sistemas de control gestual deben ser fáciles de usar y accesibles para una amplia variedad de usuarios, incluidos aquellos con discapacidades o limitaciones físicas.

Algunos de los trabajos más recientes han intentado explorar soluciones de control gestual que permitan reemplazar los dispositivos de entrada tradicionales, como ratones y teclados, como por ejemplo el trabajo de Song et al. [6] que propone una

cámara por USB que detecte los gestos de la mano y los convierta en eventos de ratón. Otros trabajos, como el de Reddy et al. [7], intentan mejorar la precisión de este tipo de sistemas mediante el uso de cintas de colores en los dedos para mejorar la detección de las distintas partes de la mano.

Sin embargo, estos sistemas no tienen por qué limitarse a la detección de gestos de la mano, sino que pueden extenderse a otras partes del cuerpo, como en el trabajo de Kumar et al. [8] que extiende un sistema con detección de gestos de manos para también detectar gestos faciales en el control de un ratón virtual. El trabajo de Bellani et al. [9], también aplica un sistema similar, pero en este caso para el control de un sistema de info-entretenimiento en un coche, donde es necesario minimizar el tiempo de distracción del conductor.

También recientemente, las grandes empresas tecnológicas han invertido en investigación en este campo para conseguir mejorar estos sistemas de control gestual, tanto en términos de precisión, como de usabilidad. Apple, en 2024, lanzó al mercado las Vision Pro, unas gafas de realidad virtual cuyo control se realiza exclusivamente mediante gestos que son detectados por las cámaras, sin necesidad de ningún otro dispositivo como un mando que contenga sensores de movimiento. En este marco, Shahi et al. [10] han desarrollado un sistema de personalización de gestos que permite a los usuarios definir sus propios gestos y detectarlos posteriormente con un 97 % de precisión.

Por otro lado, investigadores de Microsoft han desarrollado un sistema que permite a los usuarios de dispositivos de realidad virtual o aumentada, controlar el sistema mediante gestos de la boca, sin tener que emplear hardware adicional, como se muestra en el trabajo de Gemicioglu et al.[11]. Este sistema es especialmente útil para personas con discapacidades motoras que no pueden utilizar las manos para interactuar con este tipo dispositivos.

Mientras tanto Google, en 2023, anunció el desarrollo de “Project Gameface”² una herramienta para Android y Windows que tiene como objetivo permitir a los usuarios controlar un cursor mediante gestos de la cara, como guiños o movimientos de la boca. Esta solución es la que más se asemeja a los objetivos de este proyecto, ya que pretende desarrollar un sistema de control gestual que permita a los usuarios interactuar con todo el sistema operativo, sin requerir hardware adicional. Sin embargo, este proyecto aún se encuentra en una fase temprana de desarrollo y no se ha publicado ningún trabajo científico que detalle su funcionamiento.

Tras realizar esta revisión, se puede observar que existe un interés creciente en el desarrollo de sistemas de control gestual que permitan a los usuarios interactuar con dispositivos y sistemas de forma natural y sin contacto. Sin embargo, todavía no existe una solución universal que cumpla con los requisitos de precisión, fiabilidad y facilidad de uso necesarios para la adopción generalizada de los sistemas de control gestual en aplicaciones del mundo real.

²<https://github.com/google/project-gameface>

2.3. Objetivos generales y específicos

El objetivo general de este proyecto es desarrollar un sistema de control gestual que permita a los usuarios controlar un ordenador mediante gestos realizados con la mano, sin necesidad de utilizar dispositivos de entrada adicionales, como ratones o teclados. Este sistema deberá ser capaz de detectar y reconocer una amplia variedad de gestos con una precisión y velocidad aceptables, para proporcionar una experiencia de usuario fluida y sin interrupciones.

Los objetivos específicos que son necesarios para alcanzar el objetivo general se listan a continuación.

Desarrollar un modelo de aprendizaje profundo

Este modelo deberá ser capaz de detectar la posición de la mano en la imagen capturada por la cámara y reconocer al menos 10 gestos diferentes realizados con la mano. También deberá ser capaz de reconocer al menos 10 gestos faciales distintos. Tendrá que lograr una precisión mínima del 90 % en la detección de gestos en condiciones controladas de prueba. Para ello se utilizarán conjuntos de datos existentes de gestos, pudiendo extenderlos si fuera necesario, y se entrenará un modelo pre-entrenado para adaptarlo a las necesidades del proyecto. Se emplearán técnicas probadas de visión por computadora y redes neuronales para asegurar la viabilidad técnica. El desarrollo y la validación del modelo debería haberse completado en un periodo de 6 meses.

Implementar el sistema como un programa de escritorio

Se desarrollará un programa para Windows que ejecute el modelo de aprendizaje profundo para reconocer los gestos y permita controlar el cursor del sistema mediante gestos, realizados tanto con la mano como con la cara. Para lograr una velocidad aceptable, el sistema deberá responder a los gestos en menos de 100 milisegundos en el 95 % de las interacciones. El sistema deberá utilizar tecnologías estandarizadas y las API de Windows disponibles para interactuar con el sistema operativo y enviar eventos de ratón. Se realizarán pruebas en múltiples sistemas con distintas especificaciones para garantizar la compatibilidad y el rendimiento del sistema. Se debería tener lista una versión beta funcional del programa en un plazo de 4 meses.

Evaluar la usabilidad y accesibilidad del sistema

Habría que realizar pruebas de usabilidad con al menos 30 usuarios de diferentes perfiles, incluyendo personas con discapacidades físicas si es posible. Se recopilará feedback cualitativo y cuantitativo sobre la facilidad de uso, la eficacia y la satisfacción del usuario con el sistema, empleando un sistema de puntuación sobre 5. Tras iterar sobre el feedback recibido, se deberá haber mejorado el sistema para obtener una puntuación de al menos 4 sobre 5 en todas las áreas. Los participantes se podrán

seleccionar a través de una convocatoria abierta y se les compensará por su tiempo. Esta fase de pruebas y análisis de resultados debería haberse completado en un periodo de 3 meses.

Optimizar el rendimiento del sistema en tiempo real

Se mejorarán los algoritmos de procesamiento para garantizar que el sistema funcione sin retrasos perceptibles, arreglando posibles problemas técnicos o bugs que se hayan detectado durante la evaluación. El uso de CPU y memoria RAM en ordenadores con especificaciones medias deberá ser inferior al 20 % en todo momento, también se seguirán estándares de rendimiento aceptados en la industria, garantizando así que el sistema no afecte al rendimiento general del ordenador. Se podrán implementar técnicas de optimización de código, o reducir los parámetros del modelo si fuera necesario. También se usará hardware de aceleración como GPU cuando sea posible. Las optimizaciones necesarias deberán haberse completado en un periodo de 2 meses.

Documentar y publicar el sistema

Se creará una documentación completa para desarrolladores y usuarios finales, incluyendo tutoriales y guías de uso, compuesta al menos de 3 documentos distintos, uno para desarrolladores, otro para usuarios finales y otro de estilo científico. También se grabarán al menos 2 vídeo-tutoriales dirigidos a usuarios finales y desarrolladores. Para ello se utilizarán herramientas y metodologías estándar de documentación, y se colaborará con creadores de contenido si fuera necesario. La documentación tendrá que cubrir todas las funcionalidades del sistema y posibles problemas de usuario. Se completará y publicará toda la documentación junto con el sistema en un periodo de 1 mes.

2.4. Metodología

Para el desarrollo de este proyecto, se ha decidido utilizar la metodología Scrum debido a su enfoque iterativo e incremental, lo cual es adecuado para un proyecto como este que puede requerir de adaptabilidad y mejoras continuas. Esta metodología permitirá gestionar eficientemente el desarrollo del sistema, asegurando la entrega de resultados funcionales en cada iteración.

El proyecto se dividirá en los siguientes sprints, cada uno con una duración específica que se alinea con los objetivos del proyecto:

■ **Sprint 0: Preparación e inicio del proyecto (1 semana)**

- Establecer el backlog del producto con todos los requisitos y funcionalidades deseadas.
- Definir el equipo de desarrollo y los roles: scrum master, product owner y equipo de desarrollo.
- Configuración del entorno de desarrollo y herramientas de gestión de proyectos.

■ **Sprint 1: Investigación y recolección de datos (2 semanas)**

- Realizar una investigación exhaustiva sobre modelos de aprendizaje profundo y técnicas de visión por computadora para el reconocimiento de gestos.
- Recolectar y preprocesar conjuntos de datos existentes de gestos de manos y faciales.
- Establecer métricas de evaluación para la precisión y rendimiento del modelo.

■ **Sprint 2: Desarrollo del modelo de aprendizaje profundo (24 semanas)**

- Implementar un modelo inicial de aprendizaje profundo para el reconocimiento de gestos de manos y faciales, utilizando las técnicas y herramientas investigadas en el Sprint 1.
- Entrenar y validar el modelo utilizando los conjuntos de datos recolectados.
- Ajustar hiperparámetros para mejorar la precisión del modelo.

■ **Sprint 3: Integración del modelo con el sistema (16 semanas)**

- Desarrollar un programa de escritorio para Windows que integre el modelo de reconocimiento de gestos.
- Implementar la funcionalidad de control del cursor del sistema mediante gestos.
- Asegurar que el sistema responda a los gestos en menos de 100 milisegundos en el 95 % de las interacciones.

■ **Sprint 4: Pruebas y validación (12 semanas)**

- Realizar pruebas de usabilidad con al menos 30 usuarios de diferentes perfiles.
- Recopilar y analizar feedback cualitativo y cuantitativo.
- Iterar sobre el feedback recibido para mejorar la usabilidad y accesibilidad del sistema.

■ **Sprint 5: Optimización y rendimiento (8 semanas)**

- Optimizar los algoritmos de procesamiento para minimizar el uso de CPU y memoria RAM.
- Implementar técnicas de optimización de código y utilizar la GPU cuando sea posible.
- Asegurar que el uso de recursos sea inferior al 20 % en ordenadores con especificaciones medias.

■ **Sprint 6: Documentación y publicación (4 semanas)**

- Crear documentación completa para desarrolladores y usuarios finales.
- Grabar y editar al menos 2 vídeo-tutoriales.
- Preparar el lanzamiento oficial del sistema, incluyendo toda la documentación y materiales de apoyo.

Al final de cada sprint, se realizará una reunión de revisión para evaluar el progreso y realizar demostraciones de las funcionalidades completadas. También se llevarán a cabo reuniones para identificar áreas de mejora en el proceso de desarrollo y ajustar la planificación de los siguientes sprints. De esta forma se asegura la capacidad de adaptarse a cambios en los requisitos, garantizando que el proyecto se mantenga alineado con los objetivos.

2.5. Plan de trabajo y cronograma

Para el desarrollo del proyecto se han planteado una serie de actividades que se indican en el siguiente cuadro:

ACTIVIDADES	
Actividad 1: Preparación e inicio del Proyecto	
Tarea 1.1	Subtarea 1.1.1: Establecer el backlog del producto
	Subtarea 1.1.2: Definir el equipo de desarrollo y roles
	Subtarea 1.1.3: Configuración del entorno de desarrollo
Actividad 2: Investigación y recolección de Datos	
Tarea 2.1	Subtarea 2.1.1: Investigación sobre modelos de aprendizaje profundo
	Subtarea 2.1.2: Recolección y preprocesamiento de datos
	Subtarea 2.1.3: Establecimiento de métricas de evaluación
Actividad 3: Desarrollo del modelo de aprendizaje profundo	
Tarea 3.1	Subtarea 3.1.1: Implementación del modelo inicial
	Subtarea 3.1.2: Entrenamiento y validación del modelo
	Subtarea 3.1.3: Ajuste de hiperparámetros
Actividad 4: Integración del modelo con el sistema	
Tarea 4.1	Subtarea 4.1.1: Desarrollo del programa de escritorio para Windows
	Subtarea 4.1.2: Implementación de la funcionalidad de control del cursor
	Subtarea 4.1.3: Pruebas de respuesta del sistema
Actividad 5: Pruebas y Validación	
Tarea 5.1	Subtarea 5.1.1: Pruebas de usabilidad con usuarios
	Subtarea 5.1.2: Recopilación y análisis de feedback
	Subtarea 5.1.3: Iteración y mejora del sistema
Actividad 6: Optimización y rendimiento	
Tarea 6.1	Subtarea 6.1.1: Optimización de algoritmos
	Subtarea 6.1.2: Implementación de técnicas de optimización de código
	Subtarea 6.1.3: Uso de hardware de aceleración (GPU)
Actividad 7: Documentación y publicación	
Tarea 7.1	Subtarea 7.1.1: Creación de documentación para desarrolladores
	Subtarea 7.1.2: Creación de documentación para usuarios finales
	Subtarea 7.1.3: Grabación y edición de vídeo-tutoriales

Tabla 2: Actividades y tareas del proyecto.

El proyecto se ha organizado en actividades, tareas y subtareas, que abordan y organizan el conjunto total del proyecto. Las tareas y subtareas se presentan a continuación:

ACTIVIDAD 1: Preparación e inicio del proyecto	
Tarea 1.1	TÍTULO: Planificación y configuración
Responsable	Equipo de gestión
Duración	1 semana
Participan	Gestor del proyecto, desarrolladores
Subcontratación de la tarea	No
Objetivo de la actividad	Establecer la base para el desarrollo del proyecto mediante la planificación y configuración del entorno de trabajo.
Subtarea 1.1.1: Establecer el backlog del producto	
Subtarea 1.1.2: Definir el equipo de desarrollo y roles	
Subtarea 1.1.3: Configuración del entorno de desarrollo	
Resultados	Backlog establecido, equipo definido, entorno configurado

Tabla 3: Actividad 1: Preparación e inicio del proyecto.

ACTIVIDAD 2: Investigación y recolección de datos	
Tarea 2.1	TÍTULO: Investigación inicial
Responsable	Investigador principal
Duración	2 semanas
Participan	Investigadores, desarrolladores
Subcontratación de la tarea	No
Objetivo de la actividad	Recolectar y preprocesar datos relevantes para el desarrollo del modelo.
Subtarea 2.1.1: Investigación sobre modelos de aprendizaje profundo	
Subtarea 2.1.2: Recolección y preprocesamiento de datos	
Subtarea 2.1.3: Establecimiento de métricas de evaluación	
Resultados	Datos recolectados y preprocesados, métricas definidas

Tabla 4: Actividad 2: Investigación y recolección de datos.

ACTIVIDAD 3: Desarrollo del modelo de aprendizaje profundo	
Tarea 3.1	TÍTULO: Implementación del modelo
Responsable	Equipo de desarrollo
Duración	24 semanas
Participan	Desarrolladores, científicos de datos
Subcontratación de la tarea	No
Objetivo de la actividad	Crear un modelo de aprendizaje profundo para el reconocimiento de gestos.
Subtarea 3.1.1: Implementación del modelo inicial	
Subtarea 3.1.2: Entrenamiento y validación del modelo	
Subtarea 3.1.3: Ajuste de hiperparámetros	
Resultados	Modelo implementado y validado, hiperparámetros ajustados

Tabla 5: Actividad 3: Desarrollo del modelo de aprendizaje profundo.

ACTIVIDAD 4: Integración del modelo con el sistema	
Tarea 4.1	TÍTULO: Integración y pruebas
Responsable	Equipo de desarrollo
Duración	16 semanas
Participan	Desarrolladores
Subcontratación de la tarea	No
Objetivo de la actividad	Integrar el modelo de reconocimiento de gestos en un programa de escritorio y realizar pruebas.
Subtarea 4.1.1: Desarrollo del programa de escritorio para Windows	
Subtarea 4.1.2: Implementación de la funcionalidad de control del cursor	
Subtarea 4.1.3: Pruebas de respuesta del sistema	
Resultados	Programa desarrollado, funcionalidad implementada, pruebas realizadas

Tabla 6: Actividad 4: Integración del modelo con el sistema.

ACTIVIDAD 5: Pruebas y validación	
Tarea 5.1	TÍTULO: Evaluación del sistema
Responsable	Equipo de QA
Duración	12 semanas
Participan	Testers, usuarios finales
Subcontratación de la tarea	No
Objetivo de la actividad	Realizar pruebas de usabilidad y recoger feedback para mejorar el sistema.
Subtarea 5.1.1: Pruebas de usabilidad con usuarios	
Subtarea 5.1.2: Recopilación y análisis de feedback	
Subtarea 5.1.3: Iteración y mejora del sistema	
Resultados	Pruebas realizadas, feedback recopilado, sistema mejorado

Tabla 7: Actividad 5: Pruebas y validación.

ACTIVIDAD 6: Optimización y rendimiento	
Tarea 6.1	TÍTULO: Mejoras de eficiencia
Responsable	Equipo de desarrollo
Duración	8 semanas
Participan	Desarrolladores
Subcontratación de la tarea	No
Objetivo de la actividad	Optimizar los algoritmos de procesamiento para minimizar el uso de CPU y memoria RAM.
Subtarea 6.1.1: Optimización de algoritmos	
Subtarea 6.1.2: Implementación de técnicas de optimización de código	
Subtarea 6.1.3: Uso de hardware de aceleración (GPU)	
Resultados	Algoritmos optimizados, uso de CPU y memoria RAM reducido

Tabla 8: Actividad 6: Optimización y rendimiento.

ACTIVIDAD 7: Documentación y Publicación	
Tarea 7.1	TÍTULO: Preparación de la Documentación
Responsable	Equipo de documentación
Duración	4 semanas
Participan	Desarrolladores, creadores de contenido
Subcontratación de la tarea	No
Objetivo de la actividad	Crear documentación completa para desarrolladores y usuarios finales, y preparar materiales de lanzamiento.
Subtarea 7.1.1: Creación de documentación para desarrolladores	
Subtarea 7.1.2: Creación de documentación para usuarios finales	
Subtarea 7.1.3: Grabación y edición de vídeo-tutoriales	
Resultados	Documentación creada y materiales de lanzamiento preparados

Tabla 9: Actividad 7: Documentación y Publicación.

A su vez, en la figura 6 se propone la planificación temporal de las tareas y actividades descritas.

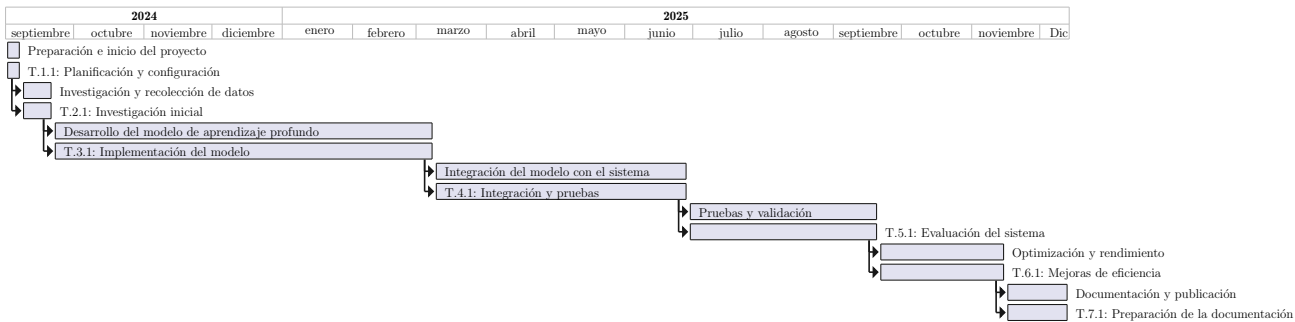


Figura 6: Diagrama de Gantt de actividades y tareas del proyecto.

2.6. Recursos humanos, materiales y de equipamiento

Para llevar a cabo este proyecto, es imprescindible contar con un equipo multidisciplinario, así como con los recursos materiales y de equipamiento adecuados. A continuación, se detallan los recursos necesarios, junto con una justificación completa de su necesidad y uso.

Recursos humanos

Equipo de gestión:

- **Gestor del proyecto:** Responsable de la planificación, ejecución y cierre del proyecto. Asegura que el proyecto se complete a tiempo y dentro del presupuesto. Coordina con todos los equipos y mantiene la comunicación con las partes interesadas.

Equipo de desarrollo:

- **Desarrolladores de software:** Encargados de implementar y probar el software necesario para el sistema de control gestual. Deben tener experiencia en desarrollo de software, especialmente en tecnologías de aprendizaje profundo y visión por computadora. También se encargan de la integración del modelo en el sistema de escritorio, garantizando el rendimiento del software.
- **Científicos de datos:** Especialistas en modelado y análisis de datos, responsables de desarrollar y entrenar los modelos de aprendizaje profundo. Además, aseguran la precisión y eficiencia del sistema de reconocimiento de gestos.

Equipo de investigación:

- **Investigadores:** Realizan la investigación inicial sobre modelos de aprendizaje profundo y técnicas de visión por computadora. Su labor incluye la recolección y preprocesamiento de datos.

Equipo de QA (aseguramiento de calidad):

- **Testers:** Realizan pruebas de usabilidad y funcionalidad para asegurar que el sistema cumple con los requisitos definidos. Recogen feedback y colaboran en la mejora continua del sistema.

Equipo de documentación:

- **Especialistas en documentación:** Encargados de crear la documentación para desarrolladores y usuarios finales. Su trabajo asegura que toda la información necesaria esté claramente detallada y accesible.
- **Creadores de contenido:** Responsables de grabar y editar los vídeo-tutoriales que acompañarán al lanzamiento del sistema.

Recursos materiales

Equipos informáticos:

- **Ordenadores de alto rendimiento:** Necesarios para el desarrollo, entrenamiento y pruebas de los modelos de aprendizaje profundo. Equipados con GPUs para realizar los procesos de entrenamiento.
- **Cámaras de alta resolución:** Utilizadas para capturar los gestos del usuario.

Software y herramientas:

- **Herramientas de desarrollo:** Incluyen entornos de desarrollo integrado, bibliotecas de aprendizaje profundo, y herramientas de visión por computadora.
- **Herramientas de gestión de proyectos:** Utilizadas para planificar y monitorear el progreso del proyecto.
- **Herramientas de control de versiones:** Git y plataformas como GitHub para la gestión del código fuente y la colaboración entre desarrolladores.

Justificación de recursos

La selección de un equipo multidisciplinario con experiencia en las áreas clave del proyecto asegura que cada fase del desarrollo cuente con personal con la experiencia y formación necesarias para llevar a cabo las tareas asignadas. Los científicos de datos con experiencia en aprendizaje profundo y visión por computadora son esenciales para crear un modelo preciso y eficiente. Mientras que los desarrolladores son necesarios para integrar este modelo en un sistema funcional y robusto.

El equipo de QA garantiza que el sistema se pruebe exhaustivamente, identificando y solucionando cualquier problema antes del lanzamiento. La documentación clara y accesible, preparada por especialistas, asegura que los usuarios y desarrolladores puedan utilizar y mantener el sistema con facilidad.

Los recursos materiales, como las computadoras de alto rendimiento y las cámaras de alta resolución, son imprescindibles para el desarrollo y la implementación del sistema de reconocimiento de gestos. Las herramientas de desarrollo y gestión de proyectos facilitan un flujo de trabajo organizado y eficiente, asegurando que el proyecto se mantenga en el camino correcto y se complete según lo planificado.

Esta combinación adecuada de recursos humanos, materiales y de equipamiento, asegura que el equipo esté bien preparado para llevar a cabo el proyecto con éxito. Por lo tanto, es fundamental contar con estos recursos para garantizar un desarrollo eficiente y de alta calidad.

2.7. Resultados e impacto esperados

El principal resultado esperado de este proyecto es el desarrollo de un sistema de control gestual altamente funcional y preciso que permita a los usuarios interactuar con un ordenador mediante gestos realizados con la mano, sin la necesidad de dispositivos de entrada adicionales como ratones o teclados. Este sistema, implementado como un programa de escritorio, ofrecerá una interfaz intuitiva para controlar el cursor y navegar por diferentes aplicaciones, incluyendo navegadores web.

Se espera que el sistema sea capaz de reconocer una amplia variedad de gestos con una precisión mínima y en tiempo real, proporcionando una experiencia de usuario fluida, necesaria para la satisfacción del usuario final. Además, se anticipa que el sistema sea configurable, permitiendo a los usuarios ajustar los gestos a sus preferencias y necesidades específicas.

El impacto esperado del proyecto es significativo en varios aspectos. En primer lugar, mejorará la accesibilidad tecnológica para personas con discapacidades físicas, proporcionando una alternativa viable a los métodos tradicionales de interacción con computadoras. Esto no solo amplía el acceso a la tecnología, sino que también promueve la inclusión digital.

En el ámbito de la productividad, el sistema puede ofrecer una forma más eficiente de interactuar con los ordenadores en casos determinados, como presentaciones, demostraciones y entornos de trabajo colaborativos. Esta capacidad de controlar el cursor y realizar acciones con gestos puede acelerar ciertas tareas y mejorar la eficiencia en entornos laborales.

Para asegurar la calidad del proyecto, se implementarán varios mecanismos de control. Durante el desarrollo, se realizarán revisiones periódicas de código y pruebas automatizadas para detectar y corregir errores tempranamente. El equipo de QA llevará a cabo pruebas exhaustivas de usabilidad y rendimiento, recopilando feedback de usuarios reales para identificar áreas de mejora. Además, se establecerán métricas claras para evaluar la precisión y la eficiencia del sistema, asegurando que cumpla con los estándares esperados.

En términos de análisis de riesgos, se identificarán y gestionarán posibles desafíos técnicos y operativos. Uno de los principales riesgos es la variabilidad en las condiciones de iluminación y entorno, que podría afectar la precisión del reconocimiento de gestos. Para mitigar este riesgo, se utilizarán técnicas avanzadas de procesamiento de imágenes y se entrenarán los modelos con un conjunto de datos diversificado que represente diferentes condiciones. Otro riesgo es la posible resistencia de los usuarios a adoptar nuevas tecnologías; para abordar esto, se desarrollarán interfaces de usuario intuitivas y se proporcionarán tutoriales y soporte adecuados.

En definitiva, los resultados esperados de este proyecto implican un impacto positivo en la accesibilidad, productividad y desarrollo tecnológico, ofreciendo una solución innovadora para la interacción con ordenadores.

2.8. Uso de IA

Durante el desarrollo del trabajo se ha empleado la extensión GitHub Copilot para VS Code, mientras que en la redacción de la memoria se ha utilizado ChatGPT para obtener un esbozo inicial de las secciones del documento.

Referencias

- [1] C. Lugaresi, J. Tang, H. Nash et al., «MediaPipe: A Framework for Perceiving and Processing Reality,» en *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*, 2019. dirección: https://mixedreality.cs.cornell.edu/s/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf.
- [2] D. Smilkov, N. Thorat, Y. Assogba et al., *TensorFlow.js: Machine Learning for the Web and Beyond*, 2019. arXiv: 1901.05350 [cs.LG].
- [3] S. Alashhab, A. J. Gallego y M. Á. Lozano, «Efficient gesture recognition for the assistance of visually impaired people using multi-head neural networks,» vol. 114, 2022, pág. 105188. DOI: <https://doi.org/10.1016/j.engappai.2022.105188>. dirección: <https://www.sciencedirect.com/science/article/pii/S0952197622002871>.
- [4] A. Kapitanov, K. Kvanchiani, A. Nagaev, R. Kraynov y A. Makhliarchuk, «HaGRID – HAnd Gesture Recognition Image Dataset,» en *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, ene. de 2024, págs. 4572-4581.
- [5] Bell Telephone Laboratories, Incorporated, «Digital data entry glove interface device,» US4414537A, Priority date: 1981-09-15, nov. de 1983. dirección: <https://patents.google.com/patent/US4414537A/en>.
- [6] S. Song, D. Yan e Y. Xie, «Design of control system based on hand gesture recognition,» en *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, 2018, págs. 1-4. DOI: 10.1109/ICNSC.2018.8361351.
- [7] V. V. Reddy, T. Dhyanachand, G. V. Krishna y S. Maheshwaram, «Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition,» en *2020 IEEE-HYDCON*, sep. de 2020, págs. 1-5. DOI: 10.1109/HYDCON48903.2020.9242677.
- [8] A. Kumar, N. Pathak, M. Kirola, N. Sharma, B. Rajakumar y K. Joshi, «AI based mouse using Face Recognition and Hand Gesture Recognition,» en *2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1)*, 2023, págs. 1-6. DOI: 10.1109/ICAIA57370.2023.10169243.

- [9] P. Bellani, A. Picardi, F. Caruso et al., «Enhancing User Engagement in Shared Autonomous Vehicles: An Innovative Gesture-Based Windshield Interaction System,» *Applied Sciences*, vol. 13, n.º 17, 2023, ISSN: 2076-3417. DOI: 10.3390/app13179901. dirección: <https://www.mdpi.com/2076-3417/13/17/9901>.
- [10] S. Shahi, C. T. Park, R. Kang et al., *Vision-Based Hand Gesture Customization from a Single Demonstration*, 2024. dirección: <https://arxiv.org/abs/2402.08420>.
- [11] T. Gemicioğlu, R. M. Winters, Y.-T. Wang, T. M. Gable e I. J. Tashev, «TongueTap: Multimodal Tongue Gesture Recognition with Head-Worn Devices,» en *Proceedings of the 25th International Conference on Multimodal Interaction*, ép. ICMI '23, Paris, France: Association for Computing Machinery, 2023, págs. 564-573, ISBN: 9798400700552. DOI: 10.1145/3577190.3614120. dirección: <https://doi.org/10.1145/3577190.3614120>.