

Software Para La Estimación De Posición y Velocidad De La Prueba Tapping Test  
Mediante Visión Por Computador

Álvaro Sebastián Montenegro Chamorro  
Fabian Alexander Zambrano Urbina

PROGRAMA DE INGENIERÍA  
FACULTAD DE INGENIERÍA ELECTRONICA  
Universidad CESMAG

2024

Software Para La Estimación De Posición y Velocidad De La Prueba Tapping Test  
Mediante Visión Por Computador

Álvaro Sebastián Montenegro Chamorro

Fabian Alexander Zambrano Urbina

Proyecto de Trabajo de Grado en la Modalidad Estancia en Línea Presentada al Comité  
Curricular del Programa de Ingeniería Electrónica

Asesor

Mario Fernando Henao Rosero

PROGRAMA DE INGENIERÍA  
FACULTAD DE INGENIERÍA ELECTRÓNICA  
Universidad CESMAG

2024

## Agradecimientos

Queremos dedicar unas palabras llenas de gratitud a todas las personas que han estado a nuestro lado durante este proceso, sin las cuales este proyecto no habría sido posible. En primer lugar, un agradecimiento muy especial a nuestras familias. No solo por su apoyo, sino por todo el amor y paciencia que nos brindaron en cada momento. Ellos fueron nuestra mayor fuente de motivación y fortaleza. Este proyecto no solo refleja el esfuerzo que pusimos en él, sino también el sacrificio y la dedicación que ellos han mostrado hacia nosotros a lo largo de todo este camino. Nos ayudaron a mantenernos enfocados, nos levantaron en los momentos difíciles y siempre creyeron en nosotros, incluso cuando nosotros dudábamos. A ellos, les debemos todo.

También queremos agradecer de corazón a nuestros profesores, en especial al Mag. Mario Fernando Henao. Su guía y apoyo fueron fundamentales para este proyecto. Agradecemos profundamente su paciencia, su tiempo y, sobre todo, su capacidad para transmitirnos conocimientos y hacer que cada dificultad se convirtiera en una oportunidad de aprendizaje. Gracias a su experiencia, pudimos superar los desafíos que surgieron en el camino. Su dedicación ha sido clave para llegar a esta meta, y no podríamos estar más agradecidos por su acompañamiento.

A nuestros amigos y compañeros de estudio, no tenemos palabras suficientes para agradecerles. Gracias por las horas de trabajo en equipo, por compartir ideas y por siempre estar ahí para apoyarnos. Este proceso habría sido mucho más difícil sin su compañía. Cada momento de estudio y cada intercambio de ideas enriquecieron este proyecto de una manera única. Gracias por ser parte de este camino y por demostrar que la amistad y el trabajo en equipo pueden hacer que todo sea más llevadero.

Y, por supuesto, queremos agradecer a todas esas personas que, aunque no están directamente involucradas en este proyecto, nos han brindado su apoyo en distintas formas. Cada palabra de aliento, cada consejo y cada gesto de confianza nos dieron la energía necesaria para continuar. A veces, un pequeño detalle puede marcar una gran diferencia, y eso es algo que siempre vamos a recordar.

Este proyecto es el resultado de un esfuerzo compartido, y por eso queremos dedicarlo a todas las personas que han sido parte de él, de una u otra manera. Cada uno de ustedes ha dejado una huella en este camino, y este logro es tan suyo como nuestro.

## Dedicatoria

Dedico este proyecto, con todo mi corazón, a las personas más importantes en mi vida, quienes han sido el pilar fundamental de mi camino y sin cuyo apoyo no habría podido llegar hasta aquí.

En primer lugar, a mi madre, quien ha sido mi mayor fuente de inspiración y fortaleza. No solo por el esfuerzo y sacrificio que ha hecho a lo largo de mi vida para que pudiera llegar a este momento, sino por los valores que me ha inculcado desde siempre. Su amor incondicional, su dedicación y su compromiso han sido mi guía en cada paso que he dado. Sin ella, no estaría aquí, hoy celebrando este logro. Gracias, mamá, por siempre estar a mi lado, por darme la confianza y la seguridad de que, pase lo que pase, siempre podría contar con tu apoyo. Este proyecto es tanto tuyo como mío, porque sé que cada uno de mis éxitos es también un reflejo de tu esfuerzo.

A mis abuelos, que son una parte fundamental de mi vida, quiero dedicarles este logro con todo mi corazón. A mi abuela, por su paciencia, perseverancia y esas inquebrantables ganas de vivir que me inspiran cada día. A mi abuelo, por ser un pilar firme y fuerte, siempre guiándome con su sabiduría y ejemplo. Gracias a su amor, valores y apoyo incondicional, hoy soy quien soy. Este proyecto es también para ustedes, que siguen siendo mi mayor fuente de inspiración.

A mi familia, que ha puesto sus ilusiones en mí y ha creído en mis capacidades desde el primer día. Gracias por su apoyo constante, por ser mi motor cuando las fuerzas flaqueaban. Este proyecto no solo es fruto de mi esfuerzo personal, sino también del amor y las expectativas que ustedes han puesto en mí. No tengo palabras para expresar lo agradecido que estoy por contar con su respaldo incondicional. Gracias por acompañarme en cada paso de este viaje, por las conversaciones, los consejos y por ser mis principales motivadores.

A los seres queridos que ya no están físicamente a mi lado, pero que siempre han sido parte de este proceso. A mi primo Andres, mi hermana Fernanda y mi abuela Fabiola. Que me dieron fuerzas cuando más lo necesitaba, aunque su presencia ya no sea tangible, su espíritu sigue siendo un faro que ilumina mi camino. Cada vez que pensé que no podía seguir adelante, sentí su apoyo y su energía, y su presencia en mi vida me dio la fuerza para no rendirme. Este proyecto también es para ellos, para honrar su memoria y todo lo que me enseñaron en vida.

A mi hijo Josué, quien ha sido y sigue siendo mi mayor motivación. Desde el momento en que llegaste a mi vida, todo lo que hago tiene un propósito más grande, un propósito que va más allá de mí mismo. He realizado cada una de mis metas pensando en ti, con la esperanza de que algún día sigas mis pasos y seas un ejemplo de perseverancia, esfuerzo y dedicación. Eres el motor que me impulsa a no detenerme, a luchar por un futuro mejor para ti y para todos los que amo. Gracias por ser mi inspiración diaria.

A mi pareja Pammela, quien ha estado a mi lado en cada momento de este proceso, brindándome su apoyo, su amor y su comprensión. Ella ha sido mi compañera de vida, no solo en lo personal, sino también en lo académico, ayudándome a mantener el equilibrio y el enfoque. A su familia, que me ha recibido con los brazos abiertos y me ha brindado su apoyo de muchas maneras, también les agradezco profundamente. Ustedes han sido una parte esencial en este camino, y sin su ayuda, este proyecto no habría sido posible.

Finalmente, este es mi pequeño homenaje y agradecimiento hacia todos ustedes. Este proyecto es la forma de devolverles, aunque sea un poco, todo el amor, el tiempo, las palabras y el esfuerzo que me han dedicado. Un día, cuando comencé este camino, no sabía si estaba tomando la decisión correcta, si este era realmente el camino que quería seguir, pero hoy, gracias a ustedes, sé con certeza cuál es mi propósito y hacia dónde quiero ir. La frase que me acompañó durante toda mi carrera, y que me sigue guiando hoy, es: "El tiempo de Dios es perfecto." Siempre confié en que todo llegaría en el momento justo, y hoy puedo ver que todo el esfuerzo ha valido la pena. Gracias, de corazón, a todos los que han sido parte de esta travesía. Este logro es tanto mío como de ustedes.

Alvaro Sebastian Montenegro Chamorro

Con gran orgullo y gratitud, dedico este proyecto de grado a las personas más significativas de mi vida, quienes han sido pilares fundamentales en mi formación, no solo académica sino también personal.

En primer lugar, a mis padres, Paulo Zambrano y Adriana Urbina, a quienes debo todo lo que soy. Ustedes han sido mi primera y más grande fuente de inspiración, demostrándome con su ejemplo que el esfuerzo, la perseverancia y la dedicación son las claves para alcanzar cualquier meta. Papá, gracias por enseñarme la importancia de la disciplina la responsabilidad y la constancia, por tus palabras de ánimo en los momentos más difíciles y por ser mi guía en este camino. Mamá, gracias por tu amor incondicional, tu fortaleza y tus sabios consejos, que siempre fueron el motor que me impulsó a superar los retos más desafiantes. Este logro es tan mío como de ustedes, porque sin su apoyo y sacrificios nada de esto habría sido posible.

A mi hermana menor, Nayelet Zambrano, quien con su dulzura y alegría ha sido un recordatorio constante de la importancia de mis sueños y metas. Nayelet, aunque seas la menor, tu luz y tus ganas de aprender me han inspirado profundamente. Este proyecto es también un legado para ti, como muestra de que con esfuerzo y dedicación todo es posible. Espero que siempre recuerdes que no hay límites para lo que puedes alcanzar, y que este logro sea una motivación más para que sigas persiguiendo tus propios sueños.

A mi pareja, quien ha sido mi compañera incondicional durante este recorrido. Ángela, gracias por tu paciencia, comprensión y por estar a mi lado incluso en los momentos más complejos. Tu fe en mí, incluso cuando yo dudaba de mis propias capacidades, ha sido una fuente inagotable de fuerza. Tu apoyo ha sido fundamental para mantenerme enfocado y motivado, y por eso siempre estaré profundamente agradecido. Este logro también es un reflejo del amor, la confianza y el compromiso que hemos construido juntos.

Quiero también destacar que este camino no fue fácil, pues inicié mi carrera un poco tarde y tuve que compaginar mis estudios con el trabajo. Cada día fue un desafío, pero también una oportunidad para demostrarme a mí mismo que la constancia y la pasión son más fuertes que cualquier obstáculo. Las largas jornadas de estudio después del trabajo y los sacrificios realizados han valido la pena, y este proyecto es el resultado de ese esfuerzo constante.

Este proyecto no solo representa el final de una etapa académica, sino también el inicio de un sueño largamente anhelado que marca un nuevo capítulo de mi vida lleno de retos

y oportunidades. En él se encierra no solo el conocimiento adquirido, sino también las experiencias vividas, los momentos de incertidumbre y las victorias alcanzadas. Es el testimonio de que, aunque el camino pueda ser complicado y lleno de obstáculos, la perseverancia y la fe pueden superar cualquier adversidad. Cada sacrificio realizado, cada noche sin descanso y cada desafío enfrentado tuvieron un propósito: convertirme en una mejor versión de mí mismo y demostrar que los sueños, por grandes que sean, se pueden alcanzar. Este logro es un recordatorio de que, cuando se pone el corazón y el alma en algo, el esfuerzo siempre será recompensado. A todas las personas que me acompañaron, me apoyaron y creyeron en mí, les dedico este triunfo con la certeza de que su amor y confianza fueron el motor que me impulsó a cruzar la meta. Gracias por ser parte de este camino y por darle sentido a cada paso que di para llegar hasta aquí.

Paulo Fabian Alexander Zambrano Urbina

## Contenido

Contenido .....	8
Introducción .....	12
1. El problema de investigación.....	15
1.1 Objeto o tema de investigación.....	15
1.2 Línea de investigación .....	15
1.3 Sub-línea de investigación .....	15
1.4 Planteamiento o descripción del problema .....	15
1.5 Formulación del problema .....	17
1.6 Objetivos .....	17
1.6.1 Objetivo General.....	17
1.6.2 Objetivos Específicos.....	17
1.7 Justificación .....	18
1.8 Viabilidad.....	19
1.9 Delimitación.....	19
2. Tópicos del marco teórico.....	20
2.1 Antecedentes .....	20
2.1.1 Hand gesture recognition on python and Open CV .....	20
2.1.2 Finger Tapping Test. Precisión del diseño de medida entre muestras de deportistas de élite y no deportistas.....	21
2.1.3 aplicación de la prueba de velocidad 10x5 metros, sprint de 20 metros y tapping-test con los brazos: resultados y análisis estadístico en educación secundaria. ....	22
2.2 Enunciados de los supuestos teóricos .....	22



2.2.1	Visión por computador .....	22
2.2.2	Inteligencia Artificial (IA) .....	23
2.2.3	Capacidades de control de movimiento .....	23
2.2.4	Plate Tapping Test .....	23
2.2.5	Kinovea .....	24
2.3	Definición de conceptos.....	24
2.3.1	Definición nominal de conceptos.....	24
2.3.2	Definición operativa de conceptos.....	24
2.4	Hipótesis .....	24
3.	Metodología .....	26
3.1	Enfoque.....	26
3.2	Paradigma .....	26
3.3	Método .....	27
3.4	Tipo de investigación.....	27
3.5	Diseño de investigación .....	27
3.6	Universo.....	28
3.7	Muestra .....	29
3.8	Técnicas de recolección de información.....	29
3.8.1	Validez de la Técnica.....	29
3.8.1.1	Instrumentos de Recolección de la Información.....	30
3.8.2	Confiabilidad de la Técnica .....	31
4.	Resultados de la investigación .....	33
4.1	Desarrollo de la interfaz grafica.....	33
4.1.1	Carga de archivos de video .....	35
4.1.2	Procesamiento de video .....	35

4.1.3	Visualización de resultados.....	36
4.1.4	Principales clases en el desarrollo de la interfaz.....	36
4.1.4.1	MainApp .....	36
4.1.4.2	AgregarResultado .....	37
4.1.4.3	BuscarUsuario.....	38
4.2	Procesamiento de video .....	39
4.2.1	Algoritmos de Procesamiento .....	40
4.2.1.2	Detección de manos .....	40
4.2.1.2	Filtrado de movimientos no relevantes .....	41
4.2.1.3	Detección de taps .....	43
4.2.1.3	Cálculo de velocidad y aceleración.....	45
4.3	Implementación en Estudiantes y Pruebas Realizadas .....	46
4.3.2	Proceso de Captura de Video.....	46
4.3.2	Proceso de Obtención de datos utilizando Kinovea.....	46
4.3.3	Comparación de Resultados.....	48
4.3.3.1	Aplicación de la formula MAE (Error Absoluto Medio).....	49
4.3.3.2	Coefficiente de correlación de Pearson .....	49
4.3.3.3	Error Promedio Relativo Absoluto .....	50
4.3.3.4	Confiabilidad del sistema.....	50
5.	Análisis de Resultados .....	52
5.1	Repeticiones y consistencias.....	52
5.2	Validación y Confiabilidad del Sistema .....	52
5.3	Análisis del Error y Desviaciones.....	53
5.3.1	Tipos de Errores Observados .....	53
5.3.1.1	Análisis estadístico de errores .....	55

5.3.1.2	Coeficiente de Correlación.....	55
5.3.1.3	Análisis de la Hipótesis.....	55
5.3.2	Fuentes de Desviaciones.....	55
6.	Conclusiones.....	57
7.	Recomendaciones.....	59
8.	Entregables.....	61
9.	Referencias bibliográficas.....	62
	Anexos.....	64

## Introducción

La parte central de la investigación planteada en el presente trabajo tiene como objetivo investigar y evaluar la confiabilidad de un software desarrollado para automatizar la prueba Tapping Test, que busca un cambio en la comprensión de la evaluación del movimiento humano. En el Tapping Test, se realiza una serie de movimientos alternativos, de las extremidades superiores del cuerpo, en el cual el brazo dominante se coloca entre dos círculos sobre una superficie cercana a la cadera y luego alternativamente entre los círculos. Esta prueba se utiliza para valorar las funciones del movimiento, la coordinación ojo-mano y la velocidad de los movimientos de la mano. Los usos específicos de esta prueba incluyen la evaluación de trastornos motores, la medición de la función cognitiva, la evaluación de habilidades deportivas y la investigación científica sobre el movimiento y la coordinación. La prueba genera una manera efectiva de medir y comprender el desempeño y las habilidades en una variedad de entornos. Se ha demostrado que es valioso para evaluar las habilidades humanas.

De acuerdo con diversos estudios, este tipo de pruebas se han consolidado como una herramienta de gran utilidad., por lo que se ha aplicado en variedad de campos, desde la medicina hasta la psicología (e incluso el deporte). Por ello, se ha comprobado que además de facilitar la medición de la velocidad, coordinación, precisión y rapidez de los movimientos de la mano, en el campo de la medicina refleja el estado funcional del sistema nervioso y en la psicología al igual que en el deporte, determinan la función cognitiva y la capacidad motora del paciente o deportista. Esto permite a estos profesionales realizar ajustes o adaptar algunos programas de entrenamiento; todo con el fin de optimizar el rendimiento físico y fortalecer habilidades motrices.

A medida que estas disciplinas avanzan hacia la búsqueda de métodos más eficientes y precisos para la evaluación de las habilidades motoras, la implementación de tecnologías como la visión por computador permite alcanzar una mayor precisión y confiabilidad, superando las limitaciones inherentes a la observación manual. Además, esta tecnología abre nuevas posibilidades para la monitorización y análisis de las funciones motrices, proporcionando herramientas más robustas para la evaluación detallada del desempeño motor.

Este proyecto se centra en la siguiente idea: la aplicación de la visión artificial y el análisis de datos basados en software para evaluar la ejecución de la prueba Tapping Test. En

lugar de depender de la observación humana, este enfoque tecnológico aprovecha la potencia de la grabación de videos a 240 frames por segundo y el análisis de datos informáticos para medir la velocidad y aceleración de los movimientos de las manos de manera precisa y eficiente. Esto es necesario, ya que de estas variables depende la precisión y eficiencia del sistema desarrollado. Por ende, esta exploración se basa en la observación directa de participantes realizando la prueba Tapping Test en las condiciones que define la Batería de Pruebas Eurofit [1], seguida del procesamiento y análisis de los datos recopilados a través del software desarrollado. La confiabilidad de este enfoque tecnológico tiene un impacto potencial en campos que van desde la medicina y la rehabilitación hasta la psicología y la investigación de habilidades motoras.

El presente estudio, logró cumplir con el objetivo general de analizar la confiabilidad de un software basado en visión por computador para la estimación de la posición y velocidad en la prueba Tapping Test. A lo largo de las pruebas realizadas, se demostró que el sistema desarrollado alcanzó un 83.91% de exactitud en comparación con soluciones semi-automáticas como Kinovea, lo que valida su capacidad para evaluar de manera precisa los movimientos de la mano. Estos resultados destacan la eficiencia del software propuesto en la investigación macro, permitiendo analizar la automatización en un proceso que en soluciones tradicionales como Kinovea requiere intervención manual. Además, el desarrollo de este sistema representa un aporte significativo desde la ingeniería electrónica, ya que combina tecnologías avanzadas de visión por computador con algoritmos de procesamiento de señales para ofrecer una herramienta completa. Su impacto se extiende más allá de los campos de la rehabilitación y el deporte, con aplicaciones potenciales en áreas como la robótica, la industria automotriz y sistemas de control biométrico, donde la precisión y la automatización de la captura de movimientos humanos son esenciales. Por último, el proyecto no solo contribuye a mejorar los métodos actuales de evaluación de habilidades motoras, sino que también abre nuevas posibilidades en el diseño de soluciones tecnológicas aplicadas en diferentes sectores.

Ahora bien, este documento se estructurará en diversas secciones. En la Sección 1 se aborda el problema de investigación, detallando el objeto o tema de estudio, su línea y Sub-línea de investigación, el planteamiento del problema, los objetivos (generales y específicos), la justificación, la viabilidad y la delimitación. La Sección 2 desarrolla los tópicos del marco

teórico, incluyendo antecedentes, estudios previos sobre el Finger Tapping Test y la visión por computador, los supuestos teóricos, las definiciones conceptuales y la hipótesis. En la Sección 3 se detalla la metodología, describiendo el enfoque, paradigma, método, tipo y diseño de la investigación, así como el universo y muestra de estudio, y las técnicas de recolección de información, evaluando la validez y confiabilidad de los instrumentos. En la Sección 4 se presentan los resultados, como el desarrollo de la interfaz gráfica, el procesamiento de video, la detección de movimientos y taps, así como las pruebas realizadas y la comparación de resultados mediante MAE, RMSE y el coeficiente de correlación de Pearson. La Sección 5 contiene el análisis de los resultados, abordando la repetición, consistencia, validación, confiabilidad del sistema y el análisis de errores y desviaciones. En la Sección 6 se describen los recursos utilizados, mientras que las Secciones 7 y 8 exponen las conclusiones y recomendaciones. Finalmente, en las Secciones 9 y 10 se presentan los entregables y las referencias bibliográficas.

## **1. El problema de investigación.**

### **1.1 Objeto o tema de investigación**

Software de extracción automática de posición y velocidad de la mano dominante en la ejecución de la prueba Tapping Test.

### **1.2 Línea de investigación**

La línea de investigación que corresponde al proyecto es sistemas de automatización y control de la Universidad CESMAG, en donde se desarrollan procesos investigativos orientados al modelamiento, simulación, diseño, desarrollo y evaluación de algoritmos de control, sistemas de control, sistemas inteligentes, control de procesos industriales, sistemas embebidos, acondicionamiento y procesamiento de señales, robótica, domótica e inteligencia artificial [2].

### **1.3 Sub-línea de investigación**

La Sub-línea que corresponde a la investigación es inteligencia artificial, lo que surge como una respuesta al deseo de aproximar el comportamiento humano y el pensamiento racional a diversos sistemas para la solución de determinadas problemáticas a través de diferentes técnicas para la solución de problemas, entre las que se destacan la lógica difusa, las redes neurales, los sistemas neuro-difusos y los algoritmos genéticos. De esta forma los resultados que se desprenden de los procesos investigativos desarrollados en esta línea se orientan a la generación de algoritmos y metodologías que presentan un comportamiento autónomo, dinámico que se adecua a la evolución del entorno [2].

### **1.4 Planteamiento o descripción del problema**

Actualmente, la tecnología se ha convertido en una herramienta de gran ayuda para el ser humano, pues, gracias a su eficiencia, confiabilidad y progreso, han facilitado realizar tareas diarias y actividades que requieren de tiempo, son altamente repetitivas o implican arduo trabajo, por lo que para la sociedad es frecuente plantear interrogantes, tales como ¿es posible que una herramienta o un mecanismo pueda realizar una tarea la cantidad de veces

necesarias con menor error y que tan confiable puede llegar a ser? Esto, ha permitido hacer la comparación entre el dominio natural y el artificial, de esta manera, surge el problema en cuanto a la subjetividad del trabajo del ser humano, que, en este caso es el evaluador en la evaluación de la prueba física Tapping Test, dado que, al momento de examinar el desempeño motor de una persona, es más probable que un evaluador humano presente errores, debidos a factores físicos como el cansancio, estrés o factores psicológicos, debidos a sesgos que voluntaria o involuntariamente puedes surgir. presente error en la evaluación, ya sea por factores físicos como el cansancio, estrés. En términos generales, su exactitud y confiabilidad se ven afectadas.

El Tapping test, es una prueba que evalúa la velocidad y precisión de una persona al realizar una tarea de toque repetitivo en una pantalla o superficie. En la configuración del ejercicio, un evaluador usa su experticia para juzgar los movimientos realizados por el ejecutor, al tiempo cuánto tarda en completar los 25 o 50 toques con el brazo dominante definidos para la prueba. Posteriormente valora, en comparación con un grupo de referencia (la prueba original define una escala europea), el desarrollo motriz del evaluado, lo que le permite definir un plan de entrenamiento que se ajuste a sus necesidades. Además, el evaluador debe interpretar los resultados del test, ya que estos pueden identificar estados de fatiga, modificaciones en el sistema nervioso, o medir la velocidad gestual tanto de miembros superiores como inferiores.

Como resultado, es posible que varios evaluadores tengan criterios subjetivos al evaluar la prueba. Esto puede tener un impacto en la precisión de los resultados desde la subjetividad del evaluador, además esto puede persuadir en la forma cómo se interpretan la velocidad y la precisión de las acciones del individuo.

Asimismo, la subjetividad del evaluador también puede verse afectada por factores como la experiencia previa, la capacidad de ver errores o malinterpretar las instrucciones de la prueba y el tiempo. Estos pueden ser algunos de los factores que pueden diferir entre los evaluadores e incidir en la confiabilidad y consistencia de los resultados obtenidos.

Para finalizar, el problema del subjetivismo del evaluador en comparación a la exactitud del Tapping test, pone en duda la validez y confiabilidad de los resultados de la prueba, ya que la interpretación personal del evaluador puede afectar la precisión de la evaluación. Esto es especialmente relevante en situaciones en las que se requiere tomar



decisiones o comparar el rendimiento de diferentes individuos, ya que dicha prueba, puede introducir sesgos y distorsiones en los resultados.

Agregando a lo anterior, los grupos de investigación Rampa y Cooper de la Universidad CESMAG se encuentran desarrollando un software basado en visión por computador, que estima la posición y velocidad de la mano dominante en la ejecución de la prueba Tapping Test a partir de un video, y, que consta de varios scripts para ser utilizados mediante línea de comandos. Sin embargo, aún no se ha verificado la confiabilidad del mismo en términos del porcentaje de determinación de la posición de las manos por fotograma, ni tampoco el error en la estimación de sus posiciones. Este es precisamente el objetivo de este trabajo, además de brindar una interfaz de usuario que facilite el manejo de la aplicación.

## **1.5 Formulación del problema**

¿Cuál es la confiabilidad de un software basado en visión por computador, para evaluar la prueba Tapping Test?

## **1.6 Objetivos**

### **1.6.1 Objetivo General**

Analizar la confiabilidad de un Software para la estimación de posición y velocidad de la prueba Tapping Test mediante visión por computador.

### **1.6.2 Objetivos Específicos**

- Ajustar el diseño del software de análisis de datos de posición y velocidad de la prueba Tapping Test propuesto por Rampa.
- Implementar el software de análisis de datos de posición y velocidad de la prueba Tapping Test.
- Validar el software de análisis de datos de posición y velocidad de la prueba Tapping Test.

## 1.7 Justificación

Hoy en día, la industria de la tecnología se encuentra inmersa en todos los ámbitos de la vida personal y social, bien sea en el trabajo, la educación, la medicina, la comunicación, etc. Por lo cual, se ha implementado y modificado, alcanzando grandes resultados. Esta nueva era, presenta nuevas herramientas y mecanismos sólidos que buscan sustituir las actividades manuales en los procesos rutinarios y repetitivos de los seres humanos, para que de esta manera

las empresas reemplacen la mano de obra de las personas, por un software que realice procesos repetitivos para desempeñar labores de mayor importancia y que esto permita ser más eficientes a nivel productivo y social.

En este mismo sentido, se han producido efectos positivos en la realización de pruebas valoradas mediante softwares,[3] desarrolló un sistema para la adquisición y análisis de la prueba de golpeteo de dedos (FTT) utilizando una aplicación móvil de Android llamada Finger-TApp.

El sistema fue validado con 64 participantes sanos y encontró que la distribución de golpes realizados en el FTT corresponde a una distribución normal. El estudio concluyó que el sistema propuesto permite una medición objetiva del FTT y tiene potencial para futuras investigaciones en la evaluación de condiciones neurológicas.

De esta forma, se plantea este software con el fin de reducir la subjetividad de un evaluador, de forma que se logre alcanzar resultados más confiables que le permitan valorar mejor las habilidades de un individuo. Esto es ventajoso cuando se requiere realizar pruebas a un grupo numeroso de personas, que ocasionan un desgaste físico o cansancio en los evaluadores. Claro está, que las capacidades de evaluación humana, pueden utilizarse en el análisis cualitativo de deficiencias, normalidades y potencialidades en el desarrollo motriz para el deportista.

El resultado de estas pruebas, abre un sinnúmero de posibilidades de diferenciaciones de movimiento que permitirá matizar cuantitativamente a cada individuo desde una perspectiva de optimización, enriqueciéndose y potenciado aún más sus destrezas, creando metodologías cada vez más adecuadas para dar respuesta a las diferentes necesidades motoras de cada individuo.

Finalmente, y reconociendo la importancia que tiene el desarrollo motor y sus beneficios ya mencionados, el proyecto se encuentra inmerso en un proyecto de mayor alcance, por lo que, se quiere lograr evaluar el desarrollo motor de un individuo y en base a ello, lograr determinar la confiabilidad de la prueba Tapping Test, por lo que es necesario señalar que, los scripts originales no facilitan que un experto evalúe el desarrollo físico. Por ello, es necesario incorporar una interfaz gráfica que permita la interacción mediante elementos visuales, mejorando así la eficacia en su uso. Además, los scripts iniciales no habían sido verificados en cuanto a su confiabilidad, motivo por el cual este proyecto se enfocó en realizar dicha verificación.

## **1.8 Viabilidad**

Este proyecto es viable, ya que existen librerías y lenguaje de uso libre para reconocimiento de puntos clave del cuerpo humano a partir del análisis de video. El software actualmente ha implementado una librería basada en redes neuronales convolucionales llamada HandTrack, que hace posible el análisis de video offline (no en tiempo real), que usa OpenCV y Mediapipe. Debido a que el sistema no requiere dar un resultado al sujeto evaluado en tiempo real, esta aplicación puede ser ejecutada en máquinas sin prestaciones especiales. De igual manera, la Universidad cuenta con las plataformas hardware y software para llevar a cabo el proyecto y además cuenta con personal capacitado tanto en Educación Física como en Ingeniería Electrónica de los grupos de investigación Cooper y Rampa, que se encuentra involucrado en el proyecto y sirve de apoyo a su desarrollo.

Para la obtención del video, se necesita realizar capturas a 240 fps, (fotogramas por segundo), lo cual es fácilmente proporcionado por las cámaras de video de dispositivos de telefonía móvil. Por lo tanto, la inversión económica no es elevada.

## **1.9 Delimitación**

El alcance del proyecto concluye con la validación del software y la determinación de su confiabilidad, por lo que se encuentra inmerso en un trabajo de investigación docente, que tendrá en cuenta la funcionalidad y precisión de este software, para validar o analizar las capacidades de desarrollo motriz de un individuo.

## 2. Tópicos del marco teórico

### 2.1 Antecedentes

#### 2.1.1 Hand gesture recognition on python and Open CV

En este artículo [5] se discute el desarrollo de un sistema de reconocimiento de gestos manuales utilizando Python y Open CV. El sistema utiliza segmentación manual y el clasificador en cascada Haar para detectar e interpretar gestos con las manos. La cascada Haar es un algoritmo de detección de objetos que emplea un conjunto de características Haar, las cuales son patrones de contraste de brillo en las imágenes. Este clasificador utiliza múltiples etapas en cascada, donde en cada una se descartan rápidamente las áreas no relevantes de la imagen, permitiendo así detectar objetos como rostros o manos de manera eficiente. El sistema ha logrado sus objetivos de detectar, reconocer e interpretar gestos con las manos a través de la visión por computadora. Dentro de este artículo se encuentra un diagrama de flujo sobre el procesamiento con Open CV y Python, el cual se muestra en la siguiente figura. [1]

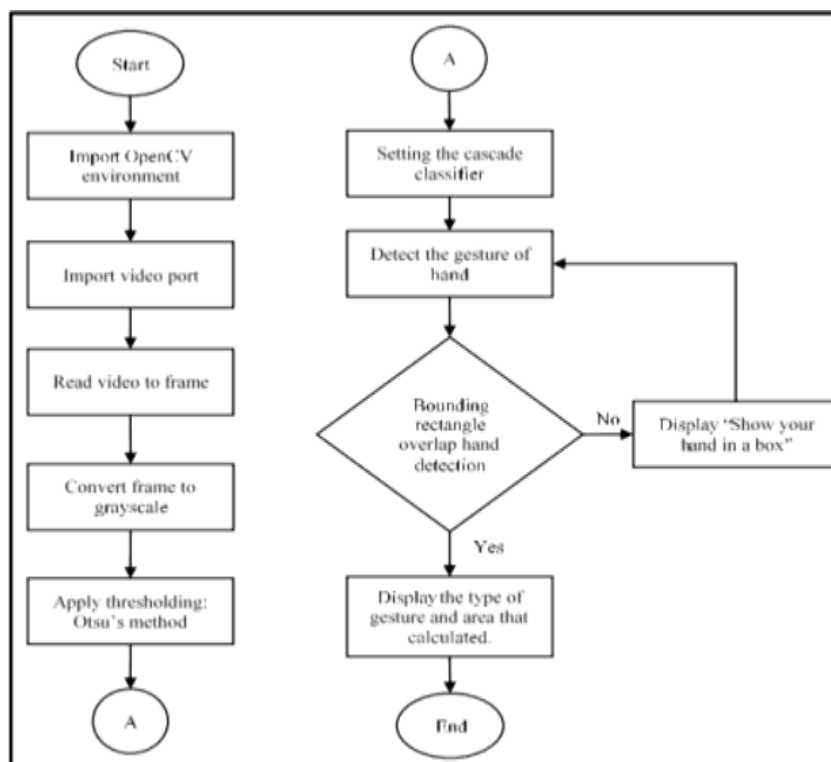


Figura 1. Diagrama de flujo del archivo Python para el reconocimiento de gestos con las manos [5].

El artículo menciona que una de las diferencias del reconocimiento de gestos de la mano es la variabilidad en la iluminación. Según [5] la intensidad de la luz en una habitación es importante para obtener un video óptimo que cuente con la intensidad y dirección de la luz correcta”.

El reconocimiento de gestos de mano a través de la visión por computadora plantea desafíos en términos de variabilidad en la velocidad, posición y orientación de la mano. La detección de gestos de mano en tiempo real a través de una cámara web puede ser afectada por la velocidad del movimiento de la mano, así como por la variación en la posición y orientación de la mano en relación con la cámara. Estos factores pueden influir en la precisión del reconocimiento de gestos de mano y en la detección de la mano en un video en tiempo real.

En resumen, el uso de Python y para reconocer gestos de la mano resulta ser una manera efectiva de captar y entender sus movimientos en tiempo real. Esto se logra al separar la mano del fondo y emplear un tipo de software llamados clasificadores en cascada de Haar, lo que permite identificar con precisión los gestos realizados. No obstante, hay desafíos a superar, como la variabilidad en la postura y la iluminación, además de requerir que la mano y la cámara web se mantengan en posiciones específicas para obtener una identificación precisa de los gestos. A pesar de estas limitaciones, el sistema ha cumplido con sus objetivos y se pueden hacer mejoras, como incorporar el reconocimiento de gestos con ambas manos y desarrollar una interfaz gráfica de usuario para facilitar su uso.

### **2.1.2 Finger Tapping Test. Precisión del diseño de medida entre muestras de deportistas de élite y no deportistas.**

El artículo [6], hace una revisión del Finger Tapping Test (FTT). Se presenta un algoritmo de libre distribución para realizarlo. El FTT se administra a una muestra de 204 participantes, de los cuales 103 (50, 5%) eran deportistas de élite y 101 (49, 5%) eran no deportistas. Se realiza un análisis de fiabilidad a través un método denominado Alpha de Cronbach, el cual es usado para medir la consistencia interna de una serie, es decir, para evaluar la magnitud en que los ítems de un instrumento están correlacionados. También se realiza un análisis general obteniendo resultados satisfactorios en validez, precisión y fiabilidad tanto para la estructura de datos como para el algoritmo. Se han obtenido resultados

que avalan la prueba como una medida simple de velocidad y de control motor. Se han encontrado resultados significativos sobre la relación entre la edad y la ejecución del test, y en relación a la distinta ejecución en función del género.

De manera que, esta investigación fue viable gracias al uso de la batería de evaluación Finger Tapping Test (FTT), el cual demostró resultados satisfactorios en términos de precisión y confiabilidad al momento de realizar el golpeteo con los dedos, por lo que, los resultados obtenidos generan experiencias significativas para el desarrollo de la presente investigación.

### **2.1.3 aplicación de la prueba de velocidad 10x5 metros, sprint de 20 metros y tapping-test con los brazos: resultados y análisis estadístico en educación secundaria.**

El trabajo expuesto por [7], el cual propone la ejecución de algunas pruebas existentes en el ámbito educativo, para evaluar la velocidad de deportistas o cualquier individuo. Las pruebas como: Velocidad 10 x 5 metros, Sprint de 20 metros y Tapping Test con los brazos se caracterizan por ser herramientas utilizadas por docentes en el campo de entrenamiento físico, por lo que, se ha realizado un estudio sobre los criterios de calidad de los mismos, así como una aplicación a 505 alumnos de educación secundaria. El análisis de los resultados permite obtener amplios valores estadísticos y ha facilitado la elaboración y clasificación por sexo en cada uno de los diferentes niveles educativos de la educación secundaria.

Dicho esto, se determina importante este aporte para la investigación, porque la prueba de Tapping Test resulta ser una herramienta de gran ayuda para el análisis de resultados precisos en cuanto a los movimientos cíclicos producidos por los brazos, por lo tanto, facilitará la adquisición de datos y ya no se deberá realizar o tenerse en cuenta manualmente o a través de prácticas observatorias, es ahí, donde se ve reflejado este trabajo y se tendrá en cuenta para la aplicación de futuros proyectos profesionales y para el buen desempeño de un deportista.

## **2.2 Enunciados de los supuestos teóricos**

### **2.2.1 Visión por computador**

La visión artificial o visión por computador es la tecnología que permite a las máquinas "ver", es decir, extraer información de imágenes digitales, para resolver alguna tarea o entender la escena que están analizando. Hoy en día se utilizan cada vez más las técnicas de visión artificial en el campo del diseño interactivo mediante la interacción con

superficies multitáctiles (multitouch), la interacción con tangibles (objetos) y el reconocimiento de gestos corporales [8].

### **2.2.2 Inteligencia Artificial (IA)**

La IA es la capacidad de las máquinas para, aprender a partir de datos y utilizar lo aprendido en la toma de decisiones en forma semejante a un ser humano. Sin embargo, a diferencia de las personas, los dispositivos basados en IA no necesitan descansar y pueden analizar grandes volúmenes de información a la vez. Asimismo, la proporción de errores es significativamente menor en las máquinas que realizan las mismas tareas que sus contrapartes humanas. La idea de que los ordenadores o los programas informáticos puedan tanto aprender como tomar decisiones, es particularmente importante y algo sobre lo que se debería ser consciente, ya que sus procesos están creciendo exponencialmente con el tiempo. Debido a estas dos capacidades, los sistemas de inteligencia artificial pueden realizar ahora muchas de las tareas que antes estaban reservadas sólo a humanos.[9]

### **2.2.3 Capacidades de control de movimiento**

Las capacidades de control de movimiento tienen como objetivo principal el dominio de una acción motriz con una intencionalidad clara, estas capacidades se originan alrededor de acciones eficaces para la obtención de logros. El control de movimiento requiere de la percepción y de la ejecución del movimiento en el espacio [10]. Según estos investigadores, la construcción de la capacidad de control de movimiento se origina a través de la interacción de diferentes procesos perceptivos, cognitivos y motores; la percepción proporciona información necesaria del ambiente al cuerpo; el aspecto cognitivo permite asimilar esta información, con el objetivo de producir una respuesta motriz [10].

### **2.2.4 Plate Tapping Test**

Tal como lo han señalado algunos estudios, el Plate Tapping Test es un método que se utiliza habitualmente en la literatura para evaluar la destreza, la velocidad y la coordinación motora mediante el movimiento repetitivo, realizando una acción de golpeteo sobre una

superficie plana. Durante el test, la activación de regiones cerebrales específicas mejora las habilidades motoras finas, mejorando el control motor [11].

### 2.2.5 Kinovea

Kinovea es un software de análisis del movimiento utiliza secuencias de vídeo para seguir y medir el movimiento de partes específicas del cuerpo, proporcionando datos sobre ángulos articulares y patrones de movimiento. El software puede utilizarse para el análisis de la marcha y otras evaluaciones del movimiento [12].

## 2.3 Definición de conceptos

### 2.3.1 Definición nominal de conceptos

Lo que se está tratando de determinar aquí es cuán confiable es el sistema de visión artificial en comparación con los datos recopilados a través del software Kinovea (semiautomático) en el desarrollo de pruebas "Tapping Test" para obtener un margen de error absoluto medio (MAE), esto es, calculando la diferencia media entre cada punto de datos del software ( $\hat{y}_i$ ) y los datos recopilados a través de Kinovea ( $y_i$ ).

### 2.3.2 Definición operativa de conceptos

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

### Ecuación 1. Confiabilidad del Software

Donde

- $y_i$ : valor observado (del sistema Kinovea)
- $\hat{y}_i$ : valor calculado (del sistema desarrollado)
- $n$ : número de observaciones.

## 2.4 Hipótesis



**Pregunta de investigación**

¿Cuál es la confiabilidad de un software basado en visión por computador, para evaluar la prueba Tapping Test?

**Hi:** La confiabilidad de un software basado en visión por computador, para evaluar la prueba Tapping Test es mayor del 95%, con respecto a la evaluación de referencia.

**Ho:** La confiabilidad de un software basado en visión por computador, para evaluar la prueba Tapping Test no es mayor del 95%, con respecto a la evaluación de referencia.

**Ha:** La confiabilidad de un software basado en visión por computador, para evaluar la prueba Tapping Test esta entre el 85% y 95%, con respecto a la evaluación de referencia.

### 3. Metodología

#### 3.1 Enfoque

Este es un estudio cuantitativo que se centra en la recopilación, procesamiento y análisis de datos numéricos mediante visión por computadora. El objetivo principal es desarrollar y evaluar un software con el que se puedan realizar estimaciones precisas del posicionamiento de la mano y de la velocidad en el proceso del Tapping Test de la mano dominante. Por lo tanto, esto exige una grabación de vídeo de alta velocidad, procesada posteriormente mediante un algoritmo especializado, para buscar los "toques" y medir los tiempos entre estos eventos clave.

Se ha implementado un sistema que combina bibliotecas de procesamiento de imágenes como OpenCV con herramientas de última generación como Mediapipe, que se utilizan para capturar las posiciones de las manos en el fotograma del vídeo, después de lo cual se han aplicado filtros para eliminar el ruido de modo que sólo se analizan los movimientos de la mano dominante. Dichos datos generados por el software incluirían las posiciones de los ejes X e Y con el tiempo necesario para cada "toque" y permitirían un análisis más profundo de la velocidad y la precisión de los movimientos.

#### 3.2 Paradigma

Desde una perspectiva epistemológica y metodológica, la investigación cuantitativa se guía por el paradigma del neopositivismo, un término acuñado por el filósofo francés Auguste Comte (1798-1857) según lo señalado por Quijano Vodniza [14].

El proyecto se fundamenta en un enfoque neopositivista en su metodología. Su objetivo primordial consiste en la validación de hipótesis a través de experimentos concretos. Dentro de este contexto, se encuentra en desarrollo un software destinado a estimar la velocidad y aceleración en la prueba Tapping Test utilizando técnicas de visión artificial. Se llevó a cabo la revisión de experimentos con el propósito de verificar las hipótesis relacionadas con la eficacia del software en la medición de los parámetros establecidos. El enfoque seguido fue similar al utilizado en pruebas de laboratorio, orientado a confirmar o refutar las teorías planteadas.

### 3.3 Método

Dado que este proyecto tiene un enfoque cuantitativo, se ha aplicado un método científico, también conocido como método empírico-analítico, tal como lo indica [14]. La variable principal bajo estudio en este proyecto es la confiabilidad del sistema de estimación de velocidad y aceleración desarrollado mediante visión artificial. El propósito central es poner a prueba y validar las hipótesis planteadas en relación con la confiabilidad de dicho sistema.

### 3.4 Tipo de investigación

Esta investigación adopta un enfoque descriptivo, ya que se busca analizar en detalle el comportamiento del software creado para estimar la velocidad y aceleración en la prueba Tapping Test mediante visión artificial. Del mismo modo, se examina cómo el software se desempeña en diversas situaciones de prueba con el objetivo de evaluar su confiabilidad en la tarea de estimación.

### 3.5 Diseño de investigación

Esta investigación, se define como un experimento puro, ya que evalúa la respuesta de una variable dependiente tras realizar un cambio en la variable independiente. En este proyecto, la variable independiente se considera el método de análisis de video. La variable (XX), representa el uso del software de visión artificial para analizar los videos de las personas que ejecutan la prueba Tapping Test, mientras que (00), representa la confiabilidad del sistema.

Las dos variables primarias que constituyen la base de este estudio para establecer la confiabilidad del sistema desarrollado incluyen la precisión en la detección de la posición de la mano con respecto a Kinovea y el porcentaje de detección de ambas manos en todos los fotogramas del vídeo. Los aspectos más importantes que deben considerarse en relación con la confiabilidad se miden sobre la base de:

- Exactitud en las posiciones de las manos: Se valoró mediante el Error Absoluto Medio (MAE) al comparar el tiempo de contacto entre Kinovea y el software

creado, midiendo de esta manera la diferencia entre ambos sistemas y corroborando la exactitud del software.

- Porcentaje de ambas manos identificadas exitosamente: Esto se estima verificando la cantidad total de imágenes del vídeo dónde el sistema consiguió identificar ambas manos. Esto permite evidenciar la uniformidad del sistema en la detección a lo largo de todo el ensayo.

Ambas medidas se consideraron necesarias para garantizar que el sistema desarrollado proporcione resultados confiables y coherentes.

Para llevar a cabo este estudio, se grabaron videos de personas realizando la prueba Tapping Test, y el software de visión artificial analiza los datos en cada uno de los fotogramas (frames) para detectar la posición de las manos y contar los toques. La confiabilidad del sistema se evaluará al comparar los resultados obtenidos mediante el software con los resultados del método semi-automático(Kinovea).

En resumen, este diseño de investigación busca determinar la confiabilidad del software de análisis de posición y velocidad en la prueba Tapping Test en comparación con el software Kinovea, considerando la detección de la posición de las manos y el conteo de toques como variables dependientes. El objetivo es evaluar si existe una diferencia significativa en la confiabilidad entre ambos métodos de análisis.

<b>RG1</b>	<b>X</b>	<b>O1</b>
<b>RG2</b>	<b>-</b>	<b>O2</b>

**RG1=RG2:** Videos de la ejecución de la prueba Tapping Test

**X:** Software de visión artificial.

**-:** Método semiautomático (usando kinovea)

**O1=O2:** Confiabilidad del sistema

### 3.6 Universo

Para la investigación en proceso cuando se habla del universo hace referencia al

Software para la estimación de velocidad y aceleración de la prueba Tapping Test mediante visión por computador.

### **3.7 Muestra**

La muestra es la misma que el universo para esta investigación.

### **3.8 Técnicas de recolección de información.**

Observación directa: En este proyecto, la técnica de observación directa juega un papel crucial en la recolección de datos, utilizando grabaciones de video a 240 fotogramas por segundo (fps). Esta técnica se apoya en una cámara de alta velocidad que captura con detalle los movimientos rápidos y precisos que caracterizan la prueba Tapping Test. Cada vídeo registrado constituye una fuente primaria de datos para el análisis del comportamiento motor, en especial la velocidad y la precisión de los movimientos de las manos.

Durante la observación directa, los participantes realizan la prueba Tapping Test, que consiste en alternar movimientos rápidos entre dos puntos de contacto. Estos movimientos son grabados desde una perspectiva frontal que captura tanto la posición como el tiempo en el que se realizan los taps. La alta frecuencia de captura (240 fps) asegura que incluso los movimientos más rápidos sean registrados con precisión, permitiendo un análisis detallado.

Los videos son procesados posteriormente mediante algoritmos de visión artificial que han sido desarrollados e implementados en Python utilizando librerías como OpenCV y Mediapipe. Este procesamiento tiene como objetivo identificar la posición de la mano en cada fotograma, eliminar el ruido generado por movimientos irrelevantes o no deseados y extraer la información precisa de cada toque (punto de cambio de dirección de la mano).

#### **3.8.1 Validez de la Técnica**

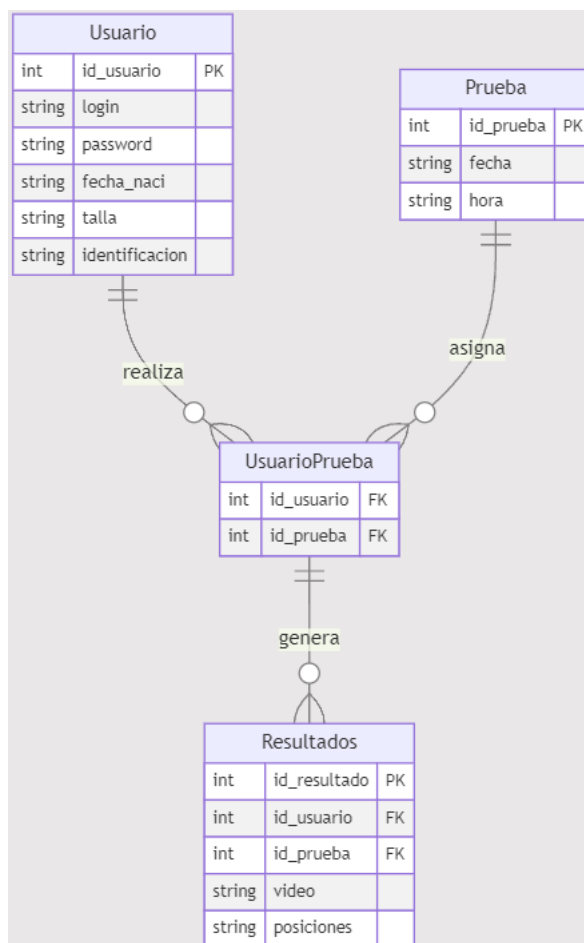
La técnica de recolección de información basada en la grabación de videos a 240 frames por segundo se considera válida en el contexto de este proyecto, ya que proporciona datos precisos y relevantes que se alinean directamente con los objetivos de investigación y permiten una evaluación efectiva de la confiabilidad del software de análisis en la prueba Tapping Test.

### 3.8.1.1 Instrumentos de Recolección de la Información

Las formas de recopilación de datos serán absolutamente decisivas en relación con el registro y el análisis del movimiento realizado en la prueba de tapping bajo un enfoque basado en la visión dentro de un proyecto de estas características relacionado con la ingeniería electrónica. En este caso, la principal instalación central será una cámara de alta velocidad, que tomará imágenes con una frecuencia de 240 fotogramas por segundo, lo que proporcionará a los examinadores una impresión extremadamente detallada y precisa de los cambios en la dirección de la mano durante el procedimiento de examen. Los datos recogidos se procesaron posteriormente mediante un software escrito en Python que utilizaba las bibliotecas OpenCV y Mediapipe para la detección de la posición de la mano y el análisis de vídeo fotograma a fotograma. El esquema de la base de datos representa uno de los componentes de la estructuración y el almacenamiento de la información experimental. En este proyecto, se estableció un esquema de base de datos relacional constituido por cuatro tablas principales:

- **Usuarios:** La tabla contiene toda la información aplicable sobre los usuarios que ejecutan las pruebas, incluyendo su nombre de usuario, contraseña, fecha de nacimiento, altura y un número de identificación único.
- **Pruebas:** Registra todas las pruebas realizadas, junto con detalles como la fecha y hora de cada prueba.
- **UsuarioPrueba:** Vincula los usuarios a las pruebas que han realizado, utilizando identificadores de seguimiento entre las dos tablas.
- **Resultados:** Almacena los datos de cada prueba, como el archivo de vídeo y las posiciones de las manos, que posteriormente se procesan para generar métricas de movimiento.

El siguiente diagrama entidad-relación explica cómo se relacionan las tablas entre sí dentro de la base de datos, garantizando que la estructura sea eficaz y ordenada para el almacenamiento y el análisis de datos.



*Figura 2. Diagrama ERD base de datos*

Cada vez que se completa una prueba, toda la información sobre sus movimientos se registra en tiempo real y se guarda automáticamente en la base de datos, lo que permite gestionar fácilmente una enorme cantidad de información. A continuación, los gráficos facilitan la observación de las relaciones entre los datos del usuario, los datos de la prueba y los resultados, lo que simplifica los análisis comparativos posteriores.

### 3.8.2 Confiabilidad de la Técnica

La confiabilidad de la técnica de recolección de información en este proyecto se sustenta en un proceso meticuloso. En primer lugar, la obtención de datos implica la captura de videos de alta calidad grabados a una velocidad de 240 frames por segundo, lo que requiere una cámara de video capaz de grabar a esta velocidad. Estos videos sirvieron como la fuente primaria de datos para el análisis de la prueba Tapping Test.

La técnica de recolección de datos se realiza mediante la grabación de personas ejecutando la prueba Tapping Test. Los videos se procesaron utilizando Python y OpenCV para un análisis detallado de cada fotograma. El software desarrollado, integrado en el entorno de Python, empleando algoritmos de visión artificial para detectar y analizar de manera precisa la posición de las manos en cada uno de los frames de video.

Los resultados obtenidos de esta observación directa se integraron en una interfaz gráfica de usuario (GUI) especialmente diseñada para presentar y analizar los datos recopilados. Esta interfaz facilito la evaluación en tiempo real de la confiabilidad del software de análisis en la prueba Tapping Test, proporcionando una herramienta efectiva y accesible para evaluar el desempeño del software en la obtención de datos de alta calidad.



## 4. Resultados de la investigación

### 4.1 Desarrollo de la interfaz grafica

El desarrollo de la interfaz gráfica del usuario fue uno de los aspectos fundamentales para crear programas que permitieran a los usuarios interactuar fácilmente y convenientemente con las funcionalidades del sistema a efectos de accesibilidad y continuidad. La interfaz se desarrolló utilizando el lenguaje de programación Python y bibliotecas, como PyQt5. Estas bibliotecas se utilizaron para gestionar los elementos visuales y las interacciones de los usuarios. La interfaz ha sido diseñada para facilitar a los usuarios el acceso a las diversas opciones que ofrece el sistema, como el registro de usuarios, una suite de pruebas y vistas de resultados de pruebas.

Se incluyó un diagrama de clases, en el que se detallan las relaciones entre las diferentes clases que se utilizarán para el desarrollo, con el fin de obtener una visión completa de la estructura del software.

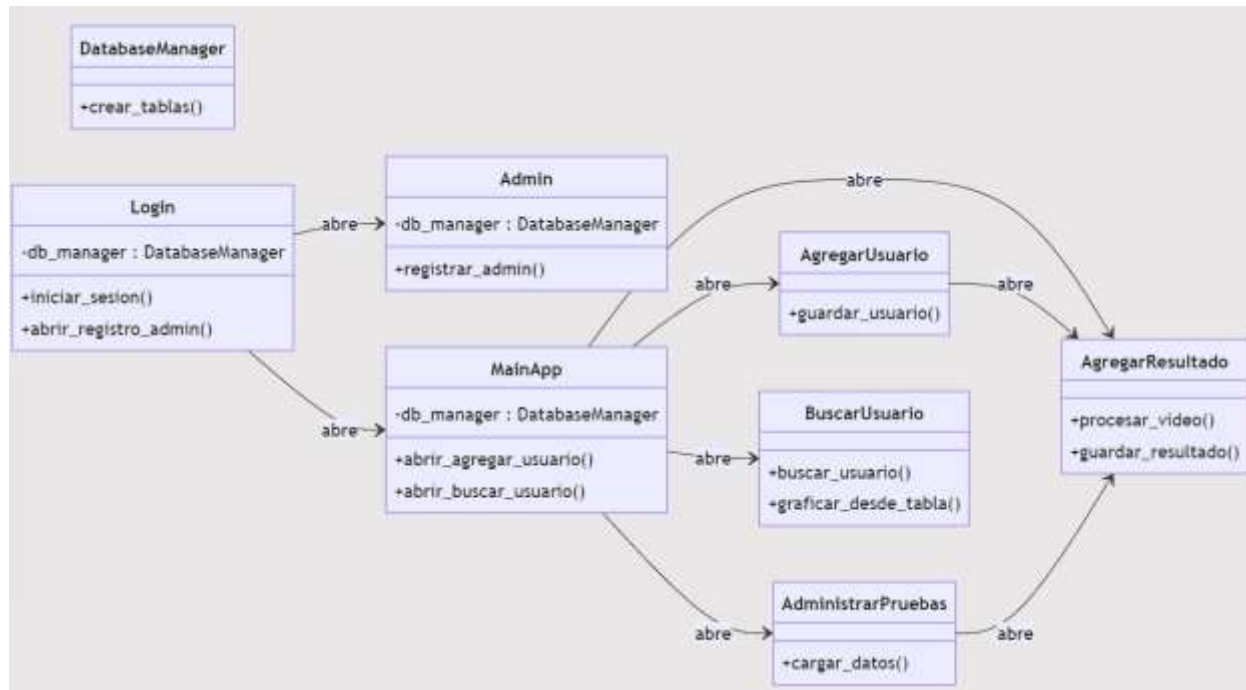


Figura 4. Diagrama de Clases

Este diagrama presenta las interacciones entre diferentes partes del sistema, tales como

Login, Admin, MainApp, AgregarUsuario y AgregarResultado, sus principales atributos y métodos. El diagrama de clases da una representación pictórica de la forma en que cada componente diferente del software fue diseñado para trabajar de manera modular, por lo tanto, es fácil de mantener y fácil para las expansiones del sistema en el futuro. Además de esto, un diagrama de estructura de base de datos entidad-relación (ERD) demostró visualmente las relaciones entre el usuario, prueba, y los resultados de la prueba. Esto permitió tener una visión clara de cómo se organizaban y almacenaban los datos recogidos durante las pruebas realizadas por los usuarios.

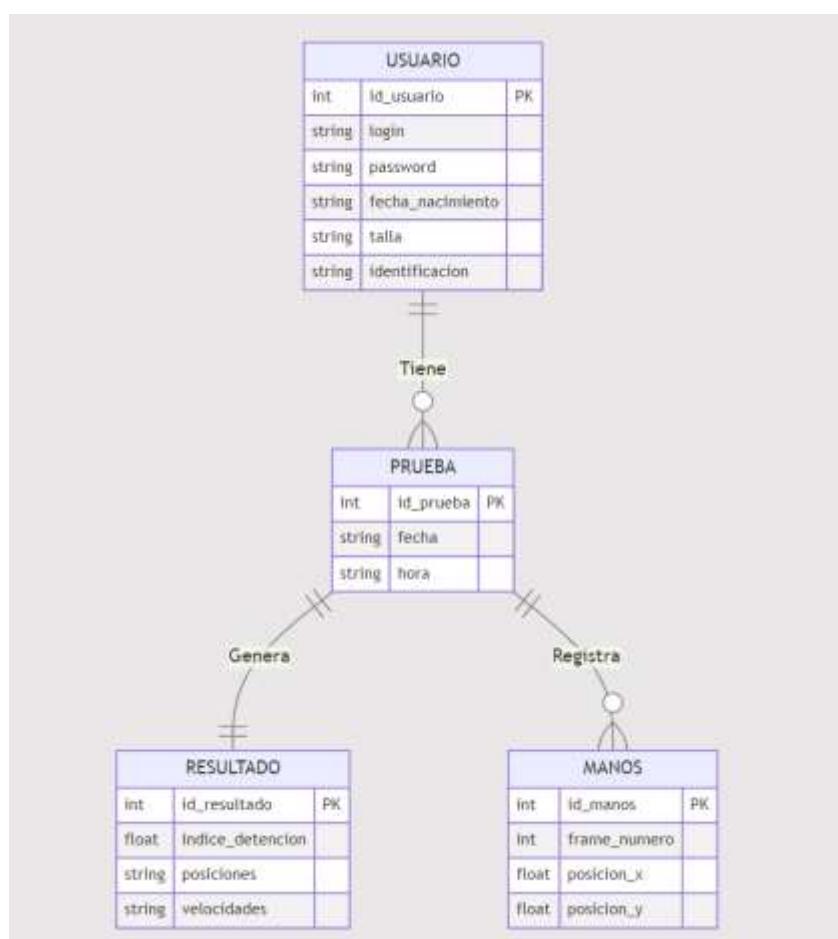


Figura 5. Diagrama ERD

Después de cada prueba realizada por el usuario, los resultados se guardaban en una base de datos donde se almacenaba información muy detallada sobre los movimientos detectados y las métricas calculadas. Estas representaciones gráficas, tanto en forma de diagrama de clases como de diagrama ERD, han contribuido a que la arquitectura del software

sea mucho más clara e informativa; Establecer una base sólida para la interpretación de los resultados obtenidos y proporcionar medios para un análisis comparativo sencillo de los datos recogidos.

#### 4.1.1 Carga de archivos de video

Los usuarios tienen la opción de seleccionar y cargar videos desde su equipo. El sistema fue diseñado para aceptar videos grabados a 240 fps, un parámetro clave para la correcta detección y análisis de los movimientos durante la prueba Tapping Test.

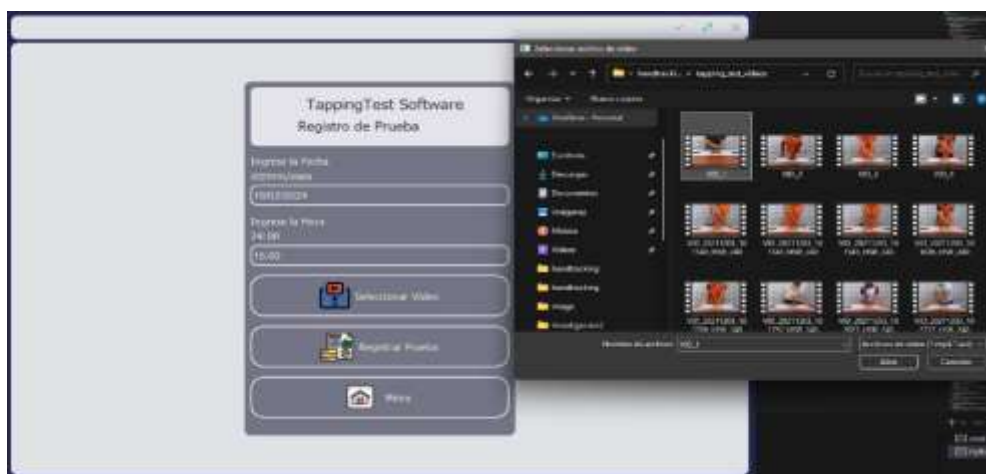


Figura 6. Carga de archivos de video

#### 4.1.2 Procesamiento de video

Una vez cargado el video, los usuarios pueden iniciar el procesamiento. La interfaz está diseñada para mostrar mensajes de progreso para mantener informado al usuario durante el análisis, que puede tomar tiempo dependiendo de la duración del vídeo.



Figura 7. Procesamiento de video



Figura 8. Video procesado

### 4.1.3 Visualización de resultados

La interfaz permite visualizar los resultados en gráficos interactivos. Los usuarios pueden explorar los datos, identificar los taps detectados y observar las velocidades calculadas. Esto facilita la comparación visual entre el sistema desarrollado y otras herramientas como Kinovea.



Figura 9. Visualización de resultados

### 4.1.4 Principales clases en el desarrollo de la interfaz

#### 4.1.4.1 MainApp

Es el corazón de la interfaz gráfica. Inicializa la ventana principal, donde los usuarios pueden interactuar con los elementos visuales y comenzar a cargar videos. También gestiona las conexiones entre los módulos de procesamiento de video y los gráficos de visualización de resultados.

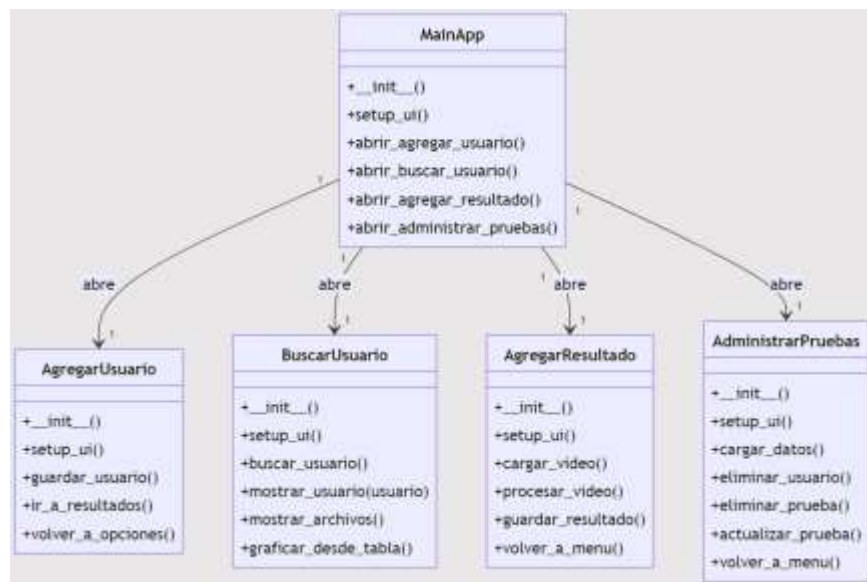


Figura 10. MainApp

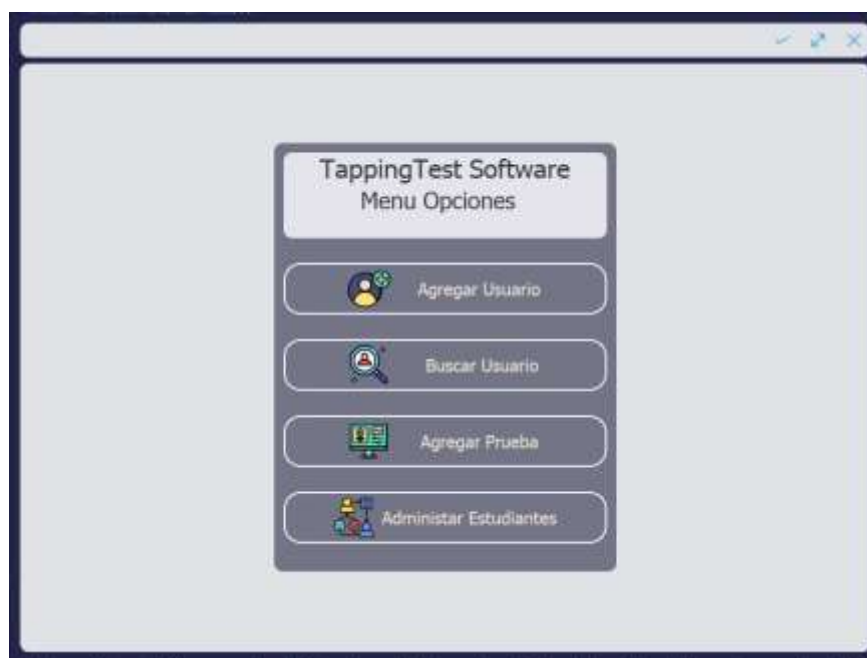


Figura 11. Menú de opciones

#### 4.1.4.2 AgregarResultado

Gestiona el procesamiento y la visualización de los resultados obtenidos. Muestra las gráficas de los movimientos de la mano dominante, los tiempos entre taps y la velocidad entre movimientos, todo de forma clara y organizada.

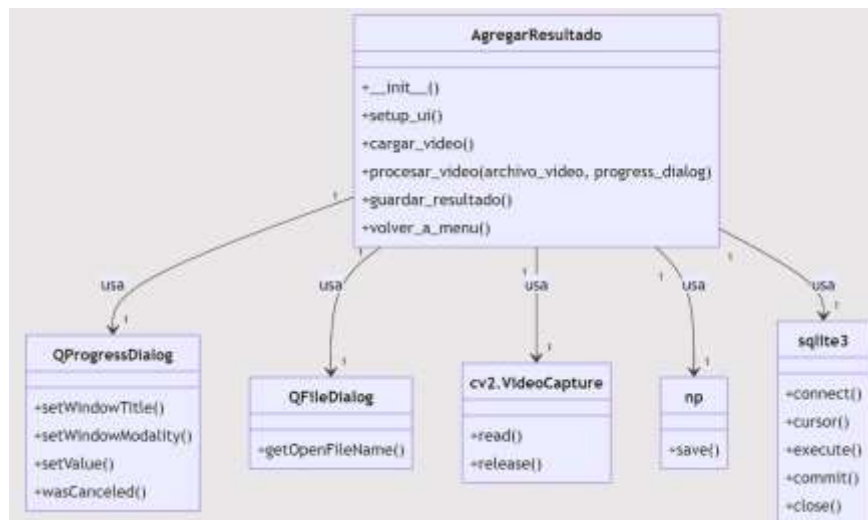


Figura 12. Interfaz agregar resultado

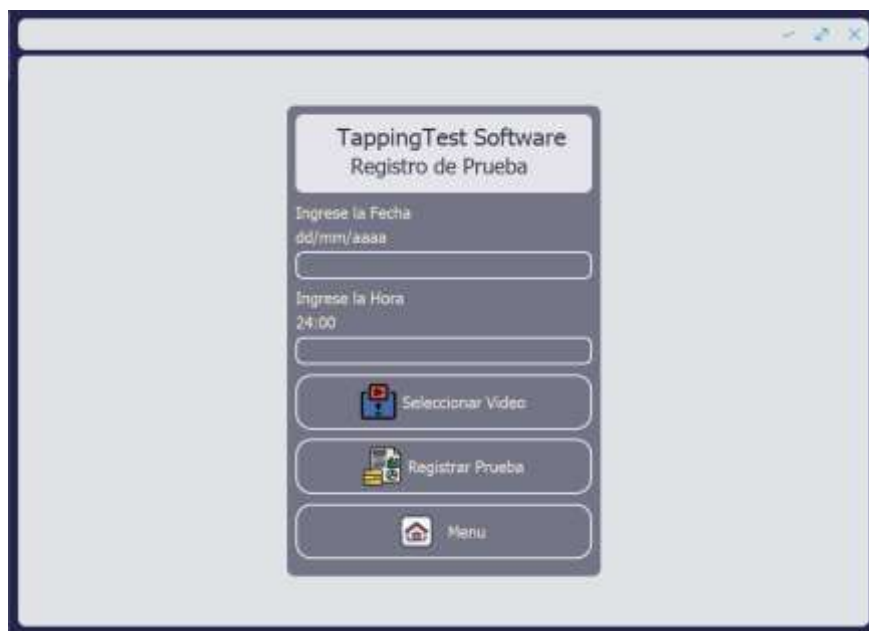


Figura 13. Registro de prueba

#### 4.1.4.3 BuscarUsuario

Esta clase facilita la búsqueda de resultados previos. Permite a los usuarios comparar los datos de pruebas anteriores realizadas por el mismo participante, brindando una visión comparativa de su evolución en la prueba Tapping Test.

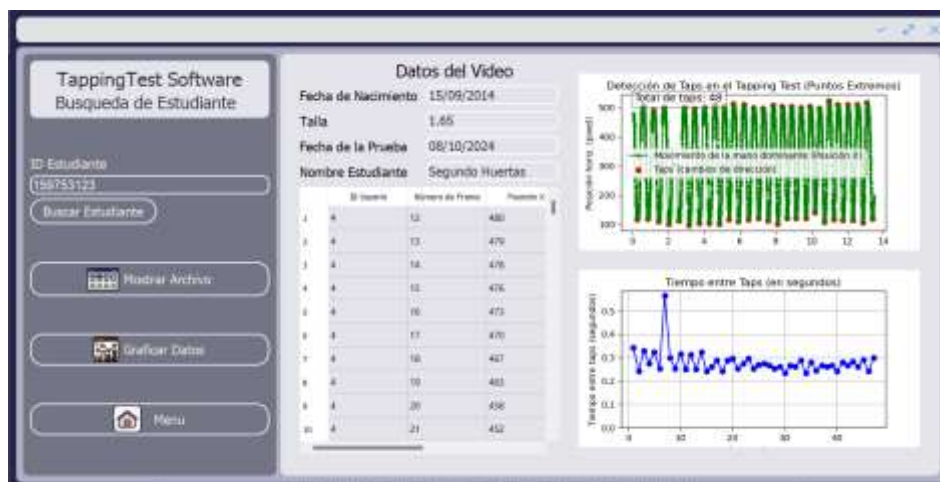


Figura 14. Interfaz búsqueda estudiante o usuario

## 4.2 Procesamiento de video

El procesamiento de video es uno de los componentes más importantes del sistema. Los videos capturados a 240 fps son analizados utilizando bibliotecas de visión por computador como OpenCV y Mediapipe. Estos videos contienen información detallada de los movimientos de las manos durante la prueba, y el sistema desarrollado tiene la capacidad de procesar cada fotograma para detectar las posiciones exactas de las manos.

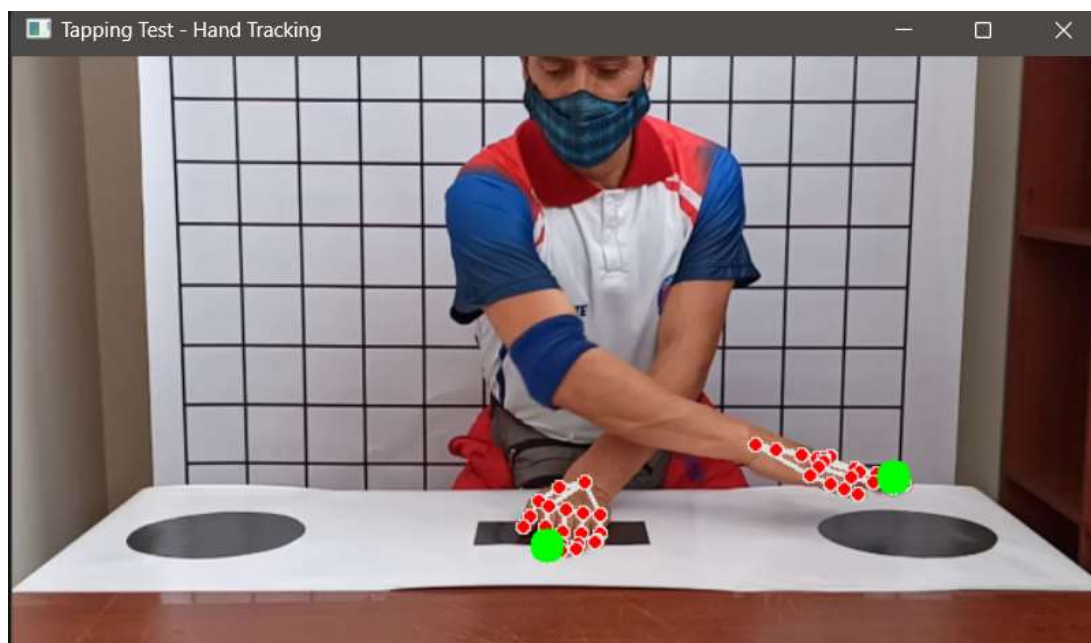


Figura 15. Procesamiento de video hand tracking

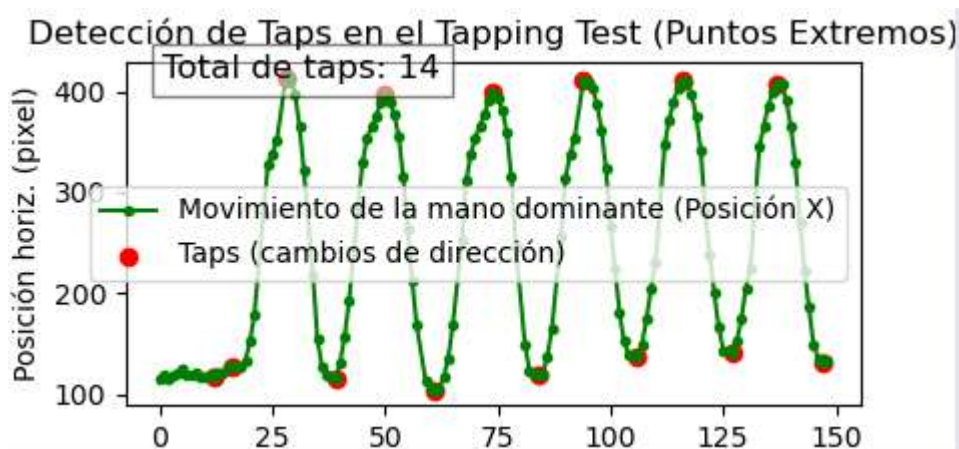


Figura 16. Grafica detección de taps puntos extremos

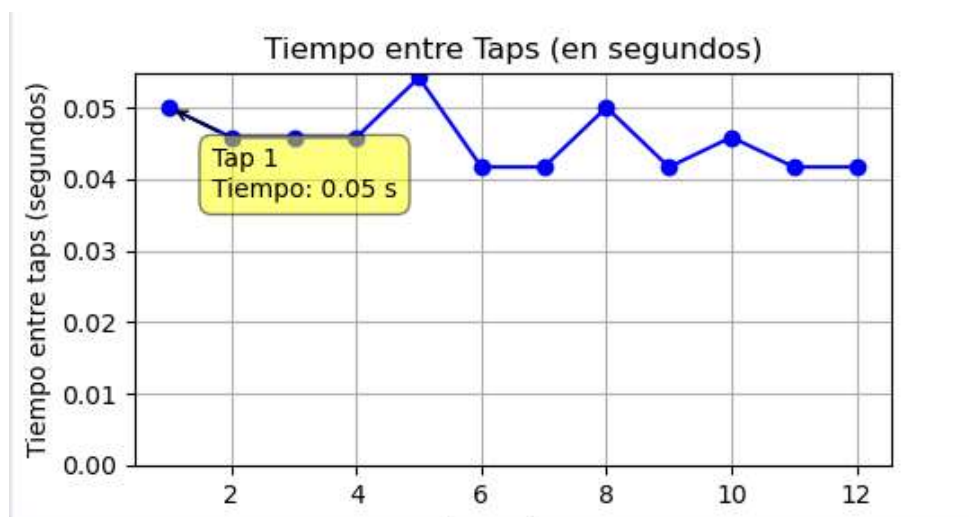


Figura 17. Grafica tiempo entre taps

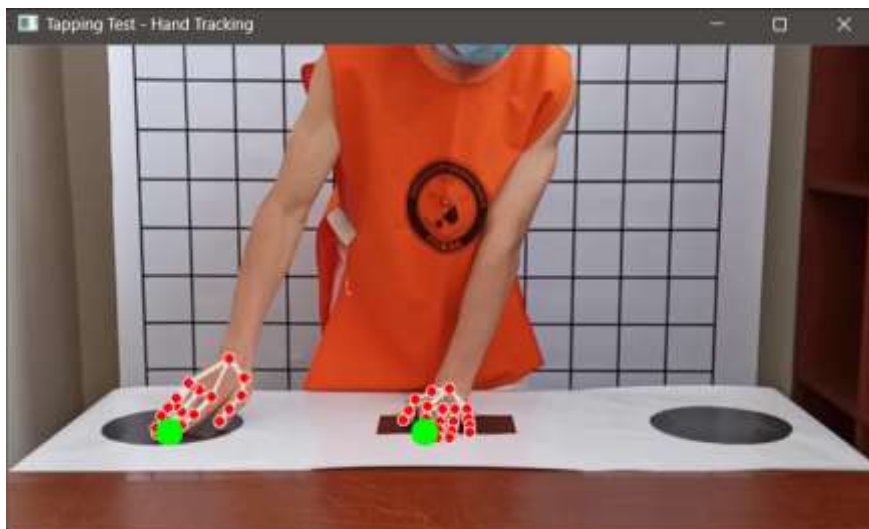
## 4.2.1 Algoritmos de Procesamiento

### 4.2.1.2 Detección de manos

Se utiliza la biblioteca Mediapipe para detectar las manos en cada fotograma del video. Esta biblioteca es capaz de localizar múltiples puntos clave en la mano, lo



que facilita identificar la posición precisa del dedo índice, que es crucial para detectar los taps en la prueba.



*Figura 18. Detección de manos*

#### **4.2.1.2 Filtrado de movimientos no relevantes**

Al observar la prueba de Tapping, se observó que no todos los movimientos eran muy importantes para su consideración: los movimientos insignificantes, sin la adición de cambios radicales en la dirección, podían introducir ruido y potencialmente sesgar los resultados. Como tal, se colocó un filtro de movimiento irrelevante para evitar movimientos tan poco importantes y mantener el enfoque en los puntos clave del examen.

De hecho, el filtro funciona detectando los puntos en los que se producen cambios de dirección de la mano; estos se marcan como los que toques que se realizan. Se definieron las entradas, los parámetros principales y la salida.

Entradas:

- Posiciones de la mano en los fotogramas: Un array con las posiciones (X, Y) de la mano en cada frame del video.
- Velocidad entre frames: Calculada a partir del desplazamiento entre las posiciones sucesivas.

Parámetros:

- Distancia mínima entre taps: Se define una distancia mínima en número de

frames entre cada cambio de dirección significativo. Este parámetro es crucial para evitar que se detecten múltiples taps en movimientos pequeños o temblores involuntarios.

- Umbral de prominencia: Define cuán pronunciado debe ser el cambio de dirección para considerarlo un tap relevante.

Salida:

- Puntos filtrados (taps relevantes): Una lista de los puntos donde ocurrieron cambios de dirección importantes, que corresponden a los taps que serán utilizados en el análisis.

Ejemplo de Funcionamiento: Se tomó como muestra una secuencia de movimiento donde la mano se mueve de forma repetitiva entre dos posiciones (por ejemplo, los puntos 0 y 400 píxeles en el eje X). Los movimientos pequeños, como temblores o ajustes menores, no son significativos para el análisis. El filtro identifica los momentos en los que la mano cambia de dirección de manera significativa, descartando los ajustes menores.

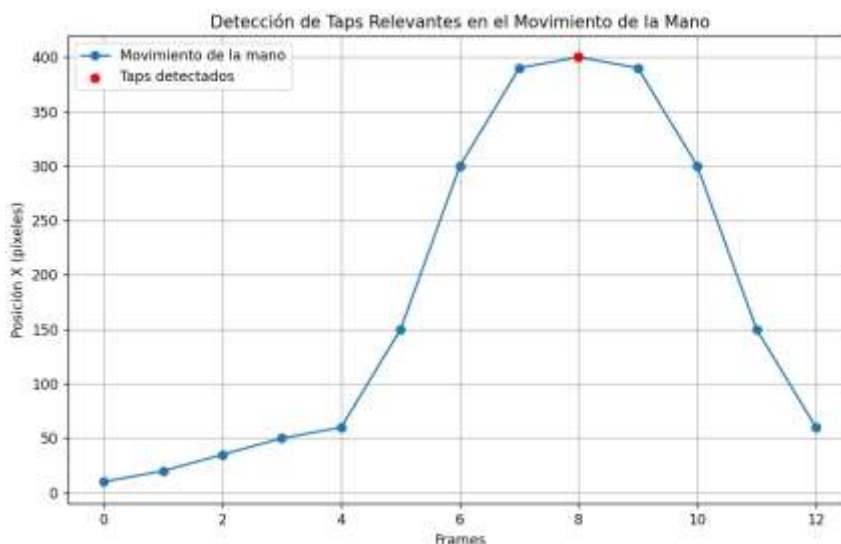
Donde se tiene una serie de movimientos como esta:

*Tabla 1. Datos Frame VS Posicion Mano*

Frame	Posicion X de la mano
1	10
2	20
3	35
4	50
5	60
6	150
7	300
8	390
9	400
10	390
11	300

12	150
13	60

Sin aplicar el filtro, cada uno de estos valores podría ser considerado en el análisis, lo que generaría datos inconsistentes. Sin embargo, el filtro identifica únicamente los cambios de dirección importantes y los reporta como taps relevantes



*Figura 19. Ejemplo de Implementación del Filtro*

El uso de un filtrado, fundamentado en la identificación de variación en la dirección de la pendiente, fue efectivo para detectar los puntos de contacto en la prueba de Tapping Test. El algoritmo se enfoca en los puntos de cambio en la curva del movimiento, de manera que solo se considera las modificaciones significantes de dirección, mientras que se descartan los movimientos aleatorios, que podrían generar ruido en los resultados. Este método incremento la exactitud y confianza del sistema, asegurando que la información registrada representa de manera precisa la realización de la prueba lo que facilita una valoración más exacta del desempeño del participante.

#### **4.2.1.3 Detección de taps**

La detección de los 'taps' constituye uno de los procesos más críticos en la evaluación de los resultados de las pruebas de Tapping, ya que éstos representan los puntos exactos en los que el movimiento de la mano cambia de dirección. Porque identificar los puntos significativos es mucho más importante que cubrir todo el

movimiento del participante, ya que en estos puntos dados el cambio de dirección es más relevante.

El algoritmo para la detección de 'taps' se basó en el proceso de analizar los máximos y mínimos locales del movimiento de las manos a lo largo del eje X. El procesamiento de picos se lleva a cabo para detectar los puntos de inflexión que forman los máximos y mínimos a lo largo del camino de la mano. La descripción de los principales componentes del proceso es la siguiente:

- Posiciones X de la mano: una matriz que proporciona las coordenadas X de la mano en cada fotograma del video.
- Distancia mínima entre picos: Define cuántos fotogramas deben haber al menos entre dos picos sucesivos.
- Prominencia: Una medida de cuan pronunciado un cambio de dirección debe ser para ser contado como un 'toque'.

Proceso de Detección:

- Detección de pico y valle: Los puntos de pico y valle se detectan en el camino de la mano. Los picos son los puntos en que la mano alcanza una distancia máxima de su movimiento, mientras que los valles son los puntos de la distancia mínima de movimiento de la mano.
- Filtrado de la prominencia: Solo se consideran los picos y valles que cruzan un umbral de prominencia, asegurando que no se detecten pequeños movimientos o temblores demasiado débiles como 'taps'.
- Cálculo de promedios: Se utiliza el promedio de los valores pico y valle para establecer el límite de tolerancia.
- Filtrado adicional: se descartan los pares de máximos y mínimos locales que se alejan demasiado del promedio, garantizando que solo se consideran los cambios más significativos en la dirección.

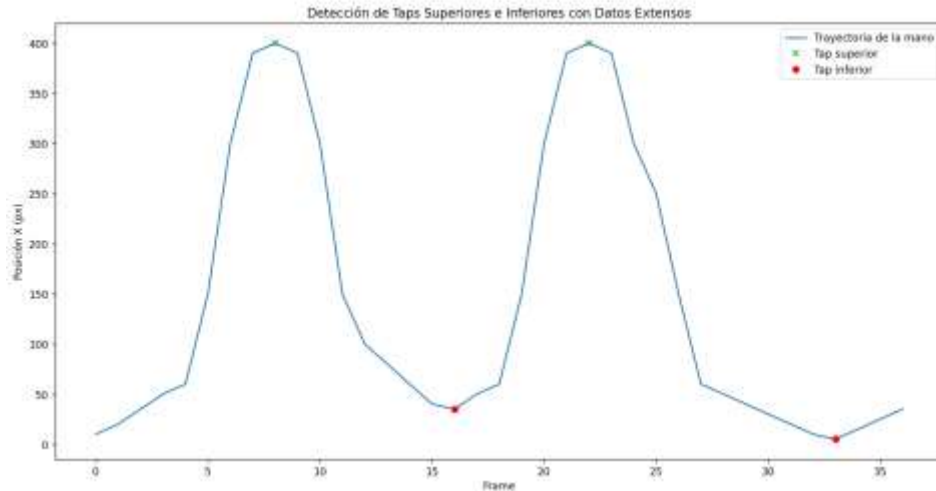


Figura 20. Ejemplo de Picos y Valles

#### 4.2.1.3 Cálculo de velocidad y aceleración

Una vez detectados los taps, el sistema calcula la velocidad de la mano entre estos puntos clave. Esto permite una evaluación más precisa de la ejecución de la prueba, La velocidad se calcula usando la fórmula básica de desplazamiento dividido entre el tiempo:

$$v = \frac{d}{t}$$

Donde:

- $v$  es la velocidad (en píxeles por segundo).
- $d$  es la diferencia en la posición de la mano (en píxeles) entre dos taps.
- $t$  es el tiempo transcurrido entre esos taps (en segundos).

La aceleración se calcula a partir del cambio en la velocidad dividido por el tiempo transcurrido entre dos velocidades:

$$a = \frac{\Delta v}{t} = \frac{v_2 - v_1}{t}$$

Donde:

- $a$  es la aceleración (en píxeles por segundo cuadrado).
- $v_2$  es la velocidad final (en píxeles por segundo).
- $v_1$  es la velocidad inicial (en píxeles por segundo).
- $t$  es el tiempo transcurrido entre dos taps consecutivos.

### 4.3 Implementación en Estudiantes y Pruebas Realizadas

Para evaluar la efectividad y confiabilidad del sistema desarrollado, se realizaron pruebas con un grupo de estudiantes bajo las condiciones establecidas por la Batería de Pruebas Eurofit, lo que permitió obtener un conjunto de datos estandarizados y comparables. Cada estudiante realizó la prueba Tapping Test, que consiste en tocar alternadamente dos puntos con la mano dominante, y se utilizó la tecnología desarrollada para analizar estos movimientos.

#### 4.3.2 Proceso de Captura de Video

Los videos se grabaron utilizando cámaras de alta calidad que permitieron capturar movimientos a 240 fotogramas por segundo (fps). Esta alta tasa de captura es crucial para lograr una precisión suficiente en el análisis de los movimientos rápidos, como los que se realizan durante la prueba Tapping Test.

*Tabla 2* resume las características de los equipos de grabación utilizados:

*Tabla 2. Características de los equipos*

Características	Detalles
Resolución	1920x1080 (Full HD)
Velocidad de Fotogramas	240 fps
Duración del Video	Variable según el participante
Formato de Video	MP4

Estos videos fueron almacenados y organizados en una base de datos, manteniendo un registro de cada sesión de prueba para su análisis posterior.

#### 4.3.2 Proceso de Obtención de datos utilizando Kinovea

Kinovea es un software ampliamente conocido y utilizado para el análisis de movimiento en videos. En el ámbito deportivo y de rehabilitación, es una herramienta popular para detectar problemas de movimiento y de producción de fuerza. Dado que el objetivo de este proyecto era el desarrollo de un software para obtener los datos de posición y cronometraje durante el Tapping Test, Kinovea se usó como un sistema de referencia de

medición, con el fin de comparar los resultados. A continuación, se describe el procesamiento de video en Kinovea para los datos de la *Tabla 3*

#### ***4.3.2.1 Captura del Video***

Se grabaron videos con una cámara a 240 fps del participante realizando la prueba Tapping Test y se cargaron en Kinovea para ser analizados.

#### ***4.3.2.2 Selección de puntos de referencia***

Aunque este programa tiene una herramienta de seguimiento automático, en esta prueba era ineficaz en casos en los que, por ejemplo, la mano dominante pasaba por arriba de la no dominante, y se perdía el sistema de rastreo. Por esto, se decidió marcar los toques con la variable “taps” de forma manual. De tal manera, con la función frame a frame se siguió cada movimiento de las manos para saber con exactitud los momentos del toque. Así, se pudo hacer una referencia exacta para establecer los “taps”.



*Figura 21. Perdida del seguimiento automatico(Kinovea)*

#### ***4.3.2.3 Exportación y análisis de datos***

Finalmente, después de hacer una identificación manual de los puntos de taps, los datos resultantes fueron exportados a Excel (*Tabla 3*). Esta se utilizó para calcular y analizar las métricas de comparación.

*Tabla 3. Tabla de obtención de datos en Kinovea*

Time (s)	X (px)	Y (px)
0,000	322,00	-185,00
0,254	-416,74	-216,00
0,566	375,81	-195,91
...	...	...
...	...	...
13,398	-400,00	-188,09

### 4.3.3 Comparación de Resultados

Dado que Kinovea era la referencia, la confiabilidad del sistema desarrollado se determina mediante la consistencia en el tiempo o los números de taps detectados por ambas herramientas. La confiabilidad se calculó mediante dos parámetros prominentes:

- Tiempos entre taps: Comparación de los tiempos entre taps calculados por ambos sistemas.
- Velocidad de la mano: Comparación de la velocidad calculada en cada movimiento.

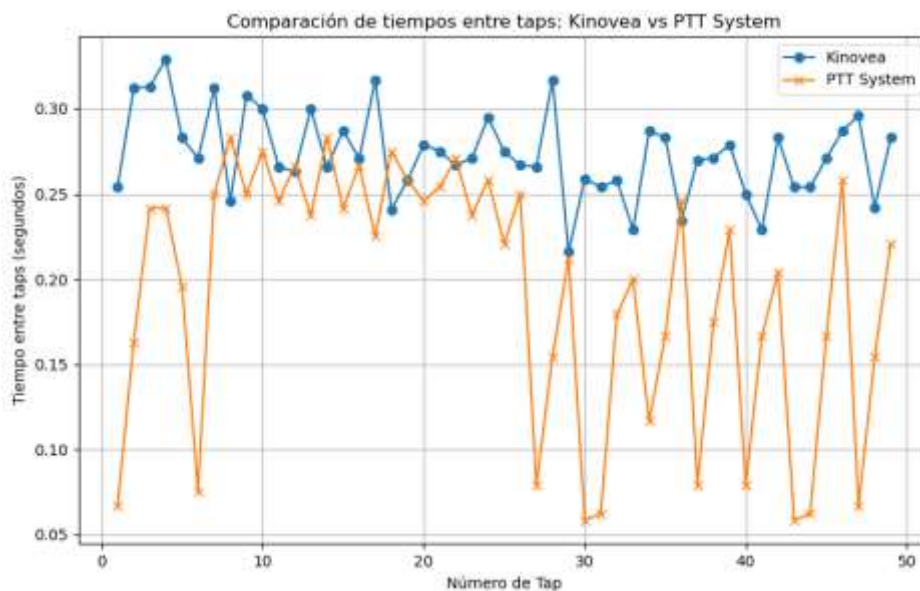


Figura 22. Comparación de los dos sistemas

A continuación, se presenta la *Tabla 4*, que resume las métricas de error calculadas para esta comparación, incluyendo el Error Absoluto Medio (MAE).



Tabla 4. Métricas Calculadas

Video	MAE (segundos)	Error promedio relativo absoluto (%)	Confiability (%)
Video 1	0.04 segundos	15.95	84.05
Video 2	0.05 segundos	16.80	83.20
Video 3	0.03 segundos	14.50	85.50
Video 4	0.06 segundos	17.10	82.90
Promedio	0.045	16.08	83.91

#### 4.3.3.1 Aplicación de la formula MAE (Error Absoluto Medio)

El MAE mide la diferencia media entre los valores del sistema realizado y los valores tomados por Kinovea. La fórmula es:

$$MAE = \frac{1}{n} \sum_{i=1}^n |T_{Kinovea,i} - T_{TTSystem,i}|$$

- $T_{Kinovea,i}$  son los tiempos entre taps detectados por Kinovea en la  $i$  –ésima medición
- $T_{TTSystem,i}$  son los tiempos entre taps detectados por el sistema desarrollado en la  $i$  –ésima medición.
- $n$  es el número total de mediciones comparadas.

#### 4.3.3.2 Coeficiente de correlación de Pearson

El coeficiente de correlación mide la fuerza y dirección de la relación lineal entre dos variables

$$r = \frac{\sum(T_{Kinovea,i} - \bar{T}_{Kinovea})(T_{TTSystem,i} - \bar{T}_{TTSystem})}{\sqrt{(\sum(T_{Kinovea,i} - \bar{T}_{Kinovea})^2) \sum(T_{TTSystem,i} - \bar{T}_{TTSystem})^2}}$$

Donde:

- $\bar{T}_{Kinovea}$  y  $\bar{T}_{Systema}$ : son las medias de los tiempos entre taps detectados por Kinovea y el sistema desarrollado, respectivamente.
- $T_{Kinovea,i}$  y  $T_{Systema,i}$ : son los tiempos individuales entre taps en ambas herramientas.

El coeficiente de correlación  $r$  varía entre -1 y 1, donde valores cercanos a 1 indican una alta correlación positiva entre los resultados de ambos sistemas. En este caso, el valor obtenido de 0.91 sugiere una fuerte correlación entre ambos sistemas.

#### 4.3.3.3 Error Promedio Relativo Absoluto

El error promedio relativo absoluto compara los valores absolutos relativos de los errores:

$$\text{Error Promedio Relativo Absoluto (\%)} = \frac{\sum |T_{Kinovea,i} - T_{TSystem,i}|}{\sum T_{Kinovea,i}} \times 100$$

Este valor, que en este caso fue del 15.95%, refleja el porcentaje promedio de error en comparación con los valores de referencia obtenidos mediante Kinovea.

#### 4.3.3.4 Confiabilidad del sistema

La confiabilidad se puede calcular como el complemento del error promedio relativo absoluto:

$$\text{Confiabilidad(\%)} = 100\% - \text{Error Promedio Relativo Absoluto}$$

Promedio Confiabilidad del Sistema

$$\frac{84.05 + 83.20 + 85.50 + 82.90}{4} = 83.91\%$$

En este caso, la confiabilidad del sistema se estimó en un 83.91%, lo que sugiere que el sistema es razonablemente preciso para detectar los taps y medir el tiempo entre ellos en la prueba Tapping Test.

La *figura 21* y *22* muestra la comparación gráfica de los tiempos entre taps obtenidos por ambos sistemas. Los gráficos fueron generados para cada participante, y se observa que las diferencias entre el sistema desarrollado y **Kinovea** son mínimas.

En la *Figura 23*, se muestra una comparación entre los tiempos entre taps detectados por nuestro sistema de visión por computador en comparación con Kinovea. En general, se puede apreciar que, para múltiples zonas del gráfico, las barras de ambos sistemas tienen aproximadamente la misma altura, lo que sugiere una alta concordancia entre la detección de taps entre el sistema desarrollado y Kinovea. Esto sugiere que, para los segmentos de la

prueba en los que esto se cumple, nuestro sistema de visión por computador puede detectar los taps con una precisión comparable a la referencia semiautomática de Kinovea.

No obstante, en algunos taps, se nota una variación considerable en la altura de las barras, lo que señala una diferencia equivalente en el tiempo registrado entre ambos sistemas. Estas diferencias pueden ser resultado de restricciones en el algoritmo para identificar cambios de dirección, cambios en la percepción de la imagen, o movimientos mínimos del individuo que el sistema no está identificando con la exactitud que Kinovea logra.

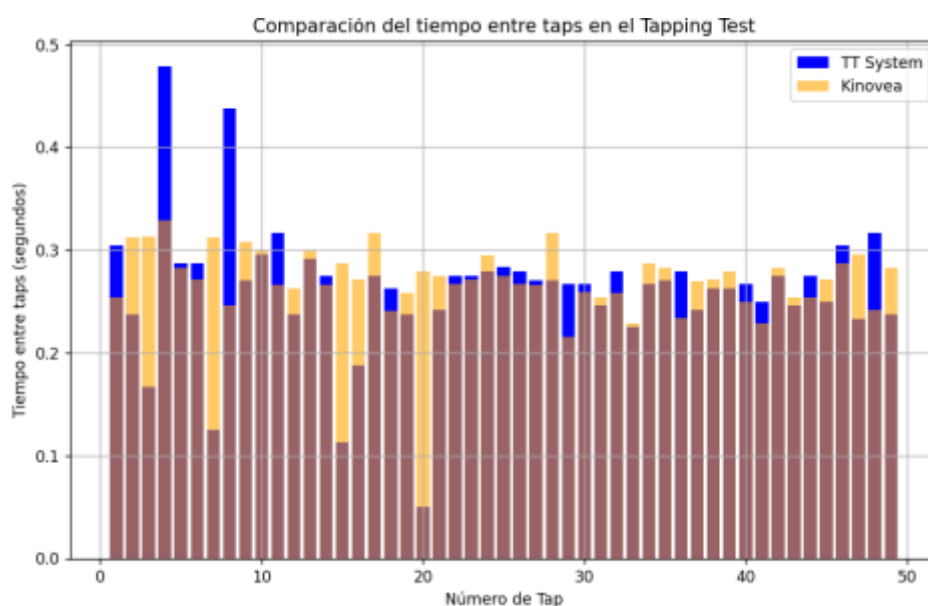


Figura 23. Comparación de tiempos entre taps

## 5. Análisis de Resultados

### 5.1 Repeticiones y consistencias

Cada uno de los sujetos tuvo una sola ejecución del Tapping Test. Este enfoque permitió obtener la precisión del sistema para cada sujeto individual. Así, los resultados obtenidos mostraron un alto nivel de consistencia en el proceso de detección de “Taps”, fortaleciendo la medición del ensayo bajo condiciones reales desarrolladas.

Las pruebas de los ensayos permitieron obtener informaciones precisas sobre los toques realizados por los alumnos, así como informaciones valiosas sobre la velocidad y la coordinación motora. La información debe, sin embargo, ser interpretada en presencia de un profesional de la educación física o de la salud ya que no evalúa directamente el rendimiento motor, pero da datos valiosos para apoyar un proceso de decisión.

### 5.2 Validación y Confiabilidad del Sistema

Para validar la confiabilidad del sistema en condiciones reales, se utilizó una muestra representativa de estudiantes. Los resultados obtenidos indicaron una fuerte correlación entre los datos de Kinovea y los datos generados por el sistema desarrollado. Esto sugiere que la solución propuesta es adecuada para evaluar con precisión la prueba Tapping Test en diversos contextos, incluyendo estudios de rehabilitación, psicomotricidad, y evaluaciones deportivas.

Los datos obtenidos se presentan de manera gráfica (*Figura 23*), donde se observan los puntos de taps detectados y las velocidades entre ellos. Estas gráficas permiten una visualización clara y detallada del rendimiento de los estudiantes en cada repetición de la prueba.

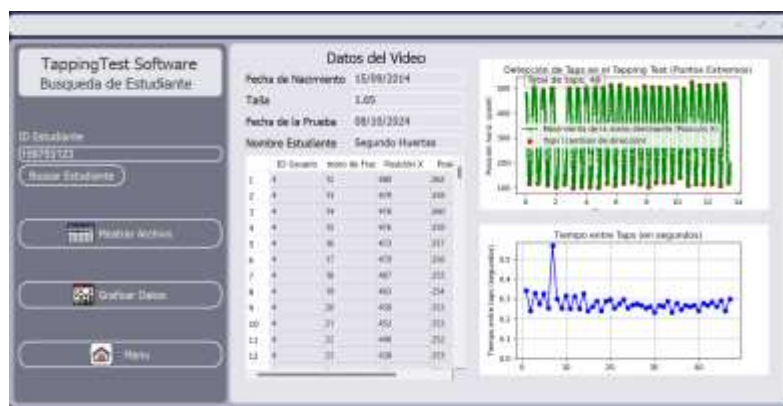


Figura 24. Grafica del rendimiento de los estudiantes en TT System

### 5.3 Análisis del Error y Desviaciones

Adicionalmente, se llevaron a cabo exámenes con un grupo de cinco alumnos -todos voluntarios del programa de grado en Educación Física, con el objetivo de verificar la fiabilidad del sistema bajo circunstancias reales. La información obtenida de estas pruebas se cotejó con la información adquirida a través del software Kinovea, frecuentemente empleado para el estudio de los movimientos humanos.

Los resultados demuestran claramente la fuerte correlación entre los datos generados por los dos sistemas. Sugiere que la solución propuesta es adecuada para la evaluación de pruebas de Tapping en diferentes contextos, como estudios de rehabilitación, psicomotricidad y evaluaciones deportivas.

Se determinó la intensidad de la correlación entre las medidas temporales registradas por el sistema desarrollado y las de Kinovea, empleando el coeficiente de correlación de Pearson. Como se ilustra en la *Tabla 4*, estos coeficientes destacan la solidez y confiabilidad del sistema para identificar grifos y cuantificar los periodos de tiempo que los separan. Las cifras recolectadas se ilustran visualmente en las *figuras 21 y 22*, las cuales presentan de manera clara los puntos de conexión identificados y las velocidades de estos. Estos diagramas proporcionan una perspectiva precisa y profunda del desempeño de los estudiantes en cada repetición de prueba, lo que simplifica el estudio del rendimiento.

*Tabla 4. El coeficiente de correlación de Pearson en videos procesados*

<b>Estudiante</b>	<b>Coficiente de Correlación (Pearson)</b>
Estudiante 1	0.93
Estudiante 2	0.91
Estudiante 3	0.89
Estudiante 4	0.92
Estudiante 5	0.94

#### 5.3.1 Tipos de Errores Observados

Las pruebas realizadas a los estudiantes mostraron fallos fundamentales en el sistema: estos fallos se denominaron aleatorios. Las evaluaciones estadísticas de las descripciones y

explicaciones de este error facilitan la valoración de la confiabilidad del sistema creado en relación con la herramienta de referencia, Kinovea.

- **Errores aleatorios:** Los errores aleatorios son menos previsibles y se originan por elementos independientes, tales como la calidad del vídeo, las fluctuaciones en la iluminación como en la *Figura 25* o las acciones inesperadas previstas en la *Figura 24* donde hace un falso movimiento. Estas variables generan una variabilidad en las mediciones, que incide principalmente en la determinación del tiempo entre las tomas. El uso de filtros para los movimientos insignificantes contribuyó a disminuir su efecto, aunque resultan más complicados de rectificar.



*Figura 25. Falso movimiento del estudiante*



*Figura 26. Confusión del sistema por colores del entorno*

### **5.3.1.1 *Análisis estadístico de errores***

Para obtener más información acerca de la confiabilidad del sistema, se llevó a cabo un estudio estadístico de los errores. Se empleó la métrica de error absoluto medio (MAE) para evaluar la exactitud del sistema en la identificación del intervalo de tiempo entre tomas. Los hallazgos revelaron un MAE promedio de 0,04 segundos, lo que señala una variabilidad reducida del error.

### **5.3.1.2 *Coefficiente de Correlación***

El coeficiente de correlación de Pearson fue calculado para analizar la correlación entre los tiempos de tap identificados por el sistema desarrollado y los datos recabados de Kinovea. Los hallazgos revelaron un coeficiente de correlación medio de 0,91, lo que señala una relación intensa entre ambos sistemas.

### **5.3.1.3 *Análisis de la Hipótesis***

Se contrastaron los resultados estadísticos con las hipótesis siguientes:

- **Ho (Hipótesis nula):** La hipótesis nula sugería que la fiabilidad del sistema desarrollado no excedería el 95%. Según los resultados logrados, la confiabilidad del sistema alcanzó el 83.91%, corroborando así esta hipótesis. Ya que la confiabilidad no llegó ni excedió el 95%, los datos corroboraron la hipótesis nula.
- **Ha (hipótesis alternativa):** La hipótesis alternativa (Ha) proponía que el sistema tendría una confiabilidad de entre el 85% y el 95%. No obstante, el estudio de los resultados reveló que la confiabilidad global del sistema llegó al 83.91%, una cifra que no está dentro del margen estimado por esta hipótesis inicial. A pesar de que la confiabilidad se aproximaba a los límites fijados, no fue suficiente para corroborar la hipótesis opcional.

## **5.3.2 Fuentes de Desviaciones**

A lo largo del proceso, se observaron varias fuentes de desviaciones entre el sistema de visión artificial y las mediciones manuales o las obtenidas con Kinovea. Entre las principales se encuentran:

- **Resolución del video:** La calidad y resolución de los videos procesados afecta directamente la precisión con la que se detectan los movimientos. Videos de baja calidad o con un número reducido de píxeles pueden dificultar la detección precisa de los taps, lo que genera desviaciones mayores entre la medición real y la estimada.
- **Iluminación y contraste:** La cantidad de luz en el entorno influye en la calidad de la detección de los bordes de las manos. Condiciones de iluminación deficientes pueden generar sombras o brillos que interfieren con el análisis de los fotogramas.
- **Ruido visual:** Elementos no controlados en el entorno, como objetos adicionales en el campo de visión o el movimiento de otras personas, también pueden afectar la precisión del sistema



## 6. Conclusiones

Este proyecto ha logrado implementar con éxito un sistema basado en visión por computador para la evaluación de la prueba Tapping Test, abordando uno de los principales desafíos en el ámbito de la evaluación de habilidades motoras: la confiabilidad. A través de la integración de tecnologías como Python, OpenCV y técnicas de procesamiento de imágenes, se ha desarrollado un software capaz de analizar los movimientos de la mano en tiempo real, ofreciendo una alternativa eficiente y precisa frente a métodos tradicionales basados en la observación manual.

Uno de los aspectos más relevantes es la capacidad del sistema para procesar videos a alta velocidad (240 fps), lo que permite una evaluación más detallada y exacta de los tiempos entre taps y de las posiciones de las manos. Este nivel de precisión es fundamental para realizar un análisis exhaustivo del rendimiento motor en diversas áreas, como la medicina, el deporte y la rehabilitación. La automatización del proceso de análisis no solo reduce la intervención humana y el margen de error, sino que también acelera significativamente el tiempo de procesamiento.

El sistema creado fue evaluado con un conjunto de estudiantes, quienes llevaron a cabo la prueba Tapping Test bajo las condiciones de la Batería de Pruebas Eurofit. En contraste con Kinovea, que demanda intervención manual en la mayoría de sus procesos, el software sugerido es totalmente automático, lo que acelera y mejora el análisis de los movimientos sin la necesidad de modificaciones manuales en cada cuadro. Se evaluó y contrastó la información obtenida con los resultados de Kinovea, evidenciando un elevado nivel de consistencia en los tiempos entre los taps y la rapidez de los movimientos. Los fallos identificados, se examinaron minuciosamente, lo que facilitó la identificación de zonas de optimización del sistema. El beneficio clave de este enfoque automático es la eficiencia en el procesamiento de datos y la reducción de la posibilidad de errores humanos, lo que lo convierte en una solución más rápida y precisa para este tipo de pruebas.

En cuanto a la confiabilidad, los resultados muestran que el sistema es altamente fiable, alcanzando un porcentaje del 83.91% en las mediciones. Este dato, en conjunto con el Error Absoluto Medio (MAE) de 0.04 segundos en los tiempos entre taps, indica que el sistema es una alternativa viable para aplicaciones clínicas y deportivas, donde la precisión es clave. No obstante, se identificaron desviaciones mínimas que pueden ser atribuidas a factores

externos, como la calidad de los videos capturados o la calibración de la cámara.

Además de la precisión técnica, otro aspecto clave del proyecto fue el desarrollo de una interfaz gráfica intuitiva que facilita el uso del sistema. El diseño amigable y funcional permite que personas sin conocimientos avanzados en tecnología o programación puedan utilizar el software de manera eficiente. Esto aumenta el potencial del sistema para ser adoptado en clínicas, laboratorios de investigación o centros deportivos, donde la facilidad de uso es un factor determinante.

El análisis del desempeño comparativo entre el sistema y Kinovea también reveló que, si bien ambas herramientas tienen un alto grado de precisión, el sistema desarrollado ofrece ventajas significativas en términos de automatización y velocidad de procesamiento. Esto abre nuevas oportunidades para futuras investigaciones y mejoras del software, con la posibilidad de integrar algoritmos de inteligencia artificial o redes neuronales para mejorar aún más la precisión y reducir el margen de error.

Finalmente, el proyecto ha demostrado el gran potencial que tienen las tecnologías de visión por computador para transformar la evaluación de habilidades motoras. La capacidad del software para detectar, analizar y reportar de manera eficiente y precisa los movimientos durante el Tapping Test representa un avance importante no solo en el ámbito de la ingeniería, sino también en áreas como la medicina deportiva y la rehabilitación. Las herramientas desarrolladas en este proyecto tienen un impacto significativo en la forma en que se realizan estas evaluaciones en entornos profesionales.

En conclusión, este proyecto no solo ha cumplido con los objetivos propuestos, sino que también ha sentado las bases para futuras investigaciones. El camino hacia la optimización de este sistema está abierto, y con las mejoras necesarias en la calidad de los videos y el perfeccionamiento de los algoritmos de detección, el sistema puede consolidarse como una herramienta de referencia en la evaluación de habilidades motoras.

## 7. Recomendaciones

Para mejorar el sistema de estimación de posición y velocidad desarrollado en este proyecto, es fundamental incorporar una serie de mejoras tanto a nivel técnico como de usabilidad. Una de las primeras recomendaciones clave es optimizar la detección de la mano dominante. Esto puede lograrse mediante el uso de un guante con sensores o marcadores visibles, como puntos de color contrastante en la mano. Este tipo de marcadores permitiría al sistema identificar la mano de manera más precisa y evitar confusiones con otras partes del cuerpo o el entorno, lo que resultaría en una detección más confiable y eficaz durante la prueba.

Otra mejora técnica que se sugiere es la optimización de los algoritmos de detección actuales. Si bien el sistema ya ofrece resultados bastante confiables, el uso de técnicas más avanzadas como el aprendizaje automático, en particular el uso de redes neuronales, podría ayudar a refinar aún más la precisión. Estas técnicas permitirían reconocer patrones complejos en los movimientos de las manos y ajustar los cálculos, reduciendo así los errores en situaciones de movimientos rápidos o difíciles de detectar.

La calidad de los videos es también un factor crítico para garantizar la precisión del sistema. Por lo tanto, se recomienda realizar una calibración cuidadosa de la cámara utilizada, asegurando que capture imágenes nítidas a 240 fps sin distorsión. Además, el entorno de grabación debe estar libre de interferencias como luces excesivamente brillantes o sombras que puedan afectar la visibilidad de los movimientos, garantizando así que las mediciones sean consistentes y fiables.

Dado que el objetivo es desarrollar un sistema aplicable a una amplia gama de usuarios, sería valioso realizar pruebas adicionales con distintos perfiles de participantes. Incluir personas de diferentes edades, habilidades motoras y condiciones físicas ayudaría a validar la eficacia del software en contextos variados y a identificar posibles áreas de mejora, especialmente para usuarios con dificultades motoras o necesidades específicas.

Asimismo, sería recomendable continuar validando el sistema comparándolo con otras herramientas de análisis de movimiento, como Kinovea, pero también explorando nuevas plataformas disponibles en el mercado. Esto aseguraría que el sistema desarrollado cumpla con los estándares más altos en cuanto a confiabilidad y precisión, y permitiría detectar y corregir cualquier desviación que pudiera presentarse con el tiempo.

Para mejorar la experiencia del usuario, se sugiere desarrollar una interfaz gráfica más personalizable, permitiendo a los usuarios ajustar parámetros del análisis en tiempo real, visualizar los resultados de manera más flexible y generar reportes automáticos. Esta funcionalidad haría que el software sea más accesible tanto en clínicas como en centros deportivos, facilitando su integración en diferentes contextos.

Además, sería beneficioso incorporar métricas adicionales al análisis, tales como la aceleración, la fluidez del movimiento y la variabilidad en los taps. Estas métricas complementarían la evaluación de la velocidad y los tiempos entre taps, ofreciendo un análisis más profundo del rendimiento motor del usuario y proporcionando una visión más integral de sus habilidades.

Finalmente, se recomienda explorar la posibilidad de incorporar hardware adicional, como un guante con sensores de presión o acelerómetros, que permitirían capturar datos sobre la fuerza aplicada durante los movimientos o la aceleración de la mano. Estos dispositivos enriquecerían el análisis y aumentarían la precisión del sistema, contribuyendo a una evaluación más completa y detallada de las habilidades motoras del usuario.

## 8. Entregables

El proyecto cuenta con dos principales entregables que garantizan la operatividad y correcta implementación del sistema de análisis para la prueba Tapping Test. El primero de estos es el ejecutable del software, que permite realizar un análisis detallado de los movimientos en la prueba, empleando visión por computador para detectar la mano dominante, calcular la velocidad de los taps y el número de repeticiones, y generar reportes precisos de cada sesión. Este ejecutable ha sido diseñado para funcionar en entornos Windows, con una interfaz gráfica (GUI) amigable e intuitiva, que facilita su uso en diversos entornos, ya sea clínico, deportivo o de investigación. El diseño de la interfaz no requiere conocimientos técnicos avanzados, lo que permite que cualquier usuario pueda interactuar con el software de forma sencilla.

Además del ejecutable, se entrega un manual de usuario que acompaña al software. Este manual explica, paso a paso, el proceso de instalación del sistema, la configuración inicial (como la calibración de la cámara) y el uso de cada una de las funcionalidades disponibles. A lo largo del manual, se incluyen capturas de pantalla y ejemplos prácticos para asegurar que el usuario pueda seguir las instrucciones sin dificultades. También cuenta con una sección de solución de problemas, que abarca los inconvenientes más comunes que podrían surgir durante el uso del software, proporcionando soluciones claras y sencillas.

En conjunto, estos entregables aseguran que el sistema desarrollado no solo funcione de manera eficaz, sino que también sea accesible y fácil de usar para cualquier usuario final, sin importar su nivel de experiencia técnica.

## 9. Referencias bibliográficas

- [1] A. J. Gálvez Garrido, “Medición y evaluación de la condición física: batería de test Eurofit,” *Educ. Física y Deport.* , pp. 1–7, 2010, [Online]. Available: <https://www.efdeportes.com/efd141/bateria-de-test-eurofit.htm>
- [2] Programa de Ingeniería Electrónica, “Proyecto Educativo del programa.” Universidad Cesmag, Pasto, p. 68, 2015.
- [3] A. Bermeo, M. Bravo, M. Huerta, J. Bermeo, B. Punin, and B. Barzallo, “Statistical characterization of the finger tapping test using an android mobile app,” *2017 IEEE 2nd Ecuador Tech. Chapters Meet. ETCM 2017*, vol. 2017-Janua, pp. 1–6, 2018, doi: 10.1109/ETCM.2017.8247459.
- [4] V. Dibia, “Detección de manos en tiempo real mediante redes neuronales (SSD) tensor Flow.,” *Medium*, 2017. <https://medium.com/@victor.dibia/how-to-build-a-real-time-hand-detector-using-neural-networks-ssd-on-tensorflow-d6bac0e4b2ce>. (accessed Nov. 21, 2023).
- [5] A. P. Ismail, F. A. A. Aziz, N. M. Kasim, and K. Daud, “Hand gesture recognition on python and opencv,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1045, no. 1, p. 012043, Feb. 2021, doi: 10.1088/1757-899x/1045/1/012043.
- [6] A. H. Mendo, V. M. Sánchez, and V. G. Morales, “Finger tapping test. Precisión del diseño de medidas entre muestras de deportistas de elite y no deportistas,” *Cuad. Psicol. del Deport.*, vol. 11, no. 1, pp. 29–43, 2011.
- [7] E. J. Martínez López, “Aplicación De La Prueba De Velocidad 10 X 5 Metros , Sprint De 20 Metros Y Tapping-Test Con Los Brazos . Resultados Y Análisis Estadístico En Educación Secundaria Application of the Speed Test 10 X 5 Meters , Sprint of 20 Meters and Tapping-Test With the,” vol. 4, pp. 1–17, 2004.
- [8] Eloi Maduell i Garcia, “Visión artificial Eloi Maduell i Garcia PID\_00184756,” *Visión Artif.*, p. 31, 2012.
- [9] L. Rouhiainen, “Inteligencia artificial 101 cosas que debes saber hoy sobre nuestro futuro,” *Alenta Editor.*, p. 22, 2018, [Online]. Available: [https://planetadelibrosar0.cdnstatics.com/libros\\_contenido\\_extra/40/39307\\_Inteligencia\\_artificial.pdf](https://planetadelibrosar0.cdnstatics.com/libros_contenido_extra/40/39307_Inteligencia_artificial.pdf)

- [10] M. Shumway-Cook, A., & Woollacott, “Motor Control,” *Encyclopedia of Human Behavior: Second Edition*. pp. 657–667, 1995. doi: 10.1016/B978-0-12-375000-6.00240- 8.
- [11] T. Di Libero, C. Carissimo, G. Cerro, AM Abbatecola, A. Marino, G. Miele, ... y A. Rodio, "Una arquitectura automatizada global basada en el protocolo de medición de la prueba de golpeteo: evaluación de la destreza manual a través de un método objetivo innovador", *Sensores*, vol. 24, núm. 13, pág. 4133, 2024.
- [12] P. Fernández-González, A. Koutsou, A. Cuesta-Gómez, M. Carratalá-Tejada, J. C. Miangolarra-Page, and F. Molina-Rueda, “Reliability of kinovea® software and agreement with a three-dimensional motion system for gait analysis in healthy subjects,” *Sensors (Switzerland)*, vol. 20, no. 11, 2020, doi: 10.3390/s20113154.
- [13] A. J. Quijano Vodniza, “Guía de investigación cuantitativa,” p. 19, 2009.

## Anexos

### Código de la Interfaz Grafica

```

from PyQt5.QtWidgets import QMainWindow, QApplication, QLineEdit, QMessageBox,
QPushButton, QVBoxLayout, QTableWidgetItem, QFileDialog, QWidget, QFrame,
QProgressDialog
from PyQt5.QtGui import QIcon
from botones import icon, icon1, icon2, icon3
from PyQt5.QtCore import QSize
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
from scipy.signal import find_peaks
from PyQt5.uic import loadUi
from PyQt5.QtCore import Qt
import sqlite3
import numpy as np
import matplotlib.pyplot as plt
import mplcursors
import sys
import os
import cv2
import mediapipe as mp

# Función para obtener la ruta absoluta del archivo .ui
def resource_path(relative_path):
    """Obtiene la ruta absoluta al recurso, funciona para desarrollo y con PyInstaller."""
    base_path = getattr(sys, '_MEIPASS', os.path.dirname(os.path.abspath(__file__)))
    return os.path.join(base_path, relative_path)

class DatabaseManager:
    def __init__(self, db_name='BaseDatos1.db'):
        self.db_name = db_name
        self.crear_tablas()

    def crear_tablas(self):
        conexion = sqlite3.connect(self.db_name)
        cursor = conexion.cursor()

        cursor.execute("""
        CREATE TABLE IF NOT EXISTS Usuario (
            id_usuario INTEGER PRIMARY KEY AUTOINCREMENT,
            login TEXT NOT NULL,

```



```

password TEXT NOT NULL,
fecha_naci TEXT,
talla TEXT,
identificacion TEXT
)
")

```

```

cursor.execute("""
CREATE TABLE IF NOT EXISTS Prueba (
    id_prueba INTEGER PRIMARY KEY AUTOINCREMENT,
    fecha TEXT,
    hora TEXT
)
""")

```

```

cursor.execute("""
CREATE TABLE IF NOT EXISTS UsuarioPrueba (
    id_usuario INTEGER,
    id_prueba INTEGER,
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),
    FOREIGN KEY (id_prueba) REFERENCES Prueba(id_prueba)
)
""")

```

```

cursor.execute("""
CREATE TABLE IF NOT EXISTS Resultados (
    id_resultado INTEGER PRIMARY KEY AUTOINCREMENT,
    id_usuario INTEGER,
    id_prueba INTEGER,
    video BLOB,
    posiciones TEXT,
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),
    FOREIGN KEY (id_prueba) REFERENCES Prueba(id_prueba)
)
""")

```

```

conexion.commit()
conexion.close()

```

```

class Login(QMainWindow):
    def __init__(self):
        super(Login, self).__init__()
        ui_path = resource_path('login.ui')
        loadUi(ui_path, self)
        self.bt_login.clicked.connect(self.iniciar_sesion)
        self.bt_crear_cuenta = self.findChild(QPushButton, 'crear_cuenta')

```

```

self.bt_crear_cuenta.clicked.connect(self.abrir_registro_admin)
self.setup_ui()

def setup_ui(self):
    self.bt_normal.hide()
    self.click_posicion = None
    self.bt_minimize.clicked.connect(lambda: self.showMinimized())
    self.bt_normal.clicked.connect(self.control_bt_normal)
    self.bt_maximize.clicked.connect(self.control_bt_maximize)
    self.bt_close.clicked.connect(lambda: self.close())

    self.setWindowFlag(Qt.FramelessWindowHint)
    self.setWindowOpacity(1)
    self.setWindowFlags(Qt.FramelessWindowHint)
    self.setAttribute(Qt.WA_TranslucentBackground)

    self.gripSize = 10
    self.grip = QPushButton("", self)
    self.grip.resize(self.gripSize, self.gripSize)
    self.frame_superior.mouseMoveEvent = self.mover_ventana

def control_bt_normal(self):
    self.showNormal()
    self.bt_normal.hide()
    self.bt_maximize.show()

def control_bt_maximize(self):
    self.showMaximized()
    self.bt_maximize.hide()
    self.bt_normal.show()

def resizeEvent(self, event):
    rect = self.rect()
    self.grip.move(rect.right() - self.gripSize, rect.bottom() - self.gripSize)

def mousePressEvent(self, event):
    self.click_posicion = event.globalPos()

def mover_ventana(self, event):
    if not self.isMaximized():
        if event.buttons() == Qt.LeftButton:
            self.move(self.pos() + event.globalPos() - self.click_posicion)
            self.click_posicion = event.globalPos()
            event.accept()

def iniciar_sesion(self):

```

```

correo = self.lineEdit1.text()
contraseña = self.lineEdit2.text()

try:
    conexion = sqlite3.connect('BaseDatos1.db')
    cursor = conexion.cursor()

    cursor.execute("""
        SELECT * FROM Administrador WHERE username = ? AND password = ?
        """, (correo, contraseña))
    admin = cursor.fetchone()

    if admin:
        QMessageBox.information(self, 'Login exitoso', '¡Has iniciado sesión correctamente!')
        self.abrir_pagina_principal()
    else:
        QMessageBox.warning(self, 'Error', 'Correo o contraseña incorrectos.')

    conexion.close()
except sqlite3.Error as e:
    print(f"Error interacting with database: {e}")
    QMessageBox.critical(self, 'Error', f"Error interacting with database: {e}")

def abrir_registro_admin(self):
    self.admin_window = Admin()
    self.admin_window.show()
    self.close()

def abrir_pagina_principal(self):
    try:
        self.main_window = MainApp()
        self.main_window.show()
        self.close()
    except Exception as e:
        print(f"Error opening main window: {e}")
        QMessageBox.critical(self, 'Error', 'Error opening main window.')

class Admin(QMainWindow):
    def __init__(self):
        super(Admin, self).__init__()
        ui_path = resource_path('admin.ui')
        loadUi(ui_path, self)
        self.bt_registrar = self.findChild(QPushButton, 'bt_registrar')
        self.username = self.findChild(QLineEdit, 'username')
        self.password = self.findChild(QLineEdit, 'password')
        self.bt_close = self.findChild(QPushButton, 'bt_close')

```

```

self.bt_minimize = self.findChild(QPushButton, 'bt_minimize')
self.bt_maximize = self.findChild(QPushButton, 'bt_maximize')
self.bt_normal = self.findChild(QPushButton, 'bt_normal')
self.bt_volver = self.findChild(QPushButton, 'bt_volver')

self.setup_ui()

self.bt_registrar.clicked.connect(self.registrar_admin)
self.bt_volver.clicked.connect(self.volver_a_login)
self.bt_close.clicked.connect(lambda: self.close())

def setup_ui(self):
    self.bt_normal.hide()
    self.click_posicion = None
    self.bt_minimize.clicked.connect(lambda: self.showMinimized())
    self.bt_normal.clicked.connect(self.control_bt_normal)
    self.bt_maximize.clicked.connect(self.control_bt_maximize)
    self.bt_close.clicked.connect(lambda: self.close())

    self.setWindowFlag(Qt.FramelessWindowHint)
    self.setWindowOpacity(1)
    self.setWindowFlags(Qt.FramelessWindowHint)
    self.setAttribute(Qt.WA_TranslucentBackground)

    self.gripSize = 10
    self.grip = QPushButton("", self)
    self.grip.resize(self.gripSize, self.gripSize)
    self.frame_superior.mousePressEvent = self.mover_ventana

def control_bt_normal(self):
    self.showNormal()
    self.bt_normal.hide()
    self.bt_maximize.show()

def control_bt_maximize(self):
    self.showMaximized()
    self.bt_maximize.hide()
    self.bt_normal.show()

def resizeEvent(self, event):
    rect = self.rect()
    self.grip.move(rect.right() - self.gripSize, rect.bottom() - self.gripSize)

def mousePressEvent(self, event):
    self.click_posicion = event.globalPos()

```

```

def mover_ventana(self, event):
    if not self.isMaximized():
        if event.buttons() == Qt.LeftButton:
            self.move(self.pos() + event.globalPos() - self.click_posicion)
            self.click_posicion = event.globalPos()
            event.accept()

def registrar_admin(self):
    username = self.username.text()
    password = self.password.text()

    if not username or not password:
        QMessageBox.warning(self, 'Error', 'Todos los campos son obligatorios.')
        return

    try:
        conexion = sqlite3.connect('BaseDatos1.db')
        cursor = conexion.cursor()

        cursor.execute("""
        INSERT INTO Administrador (username, password)
        VALUES (?, ?)
        """, (username, password))

        conexion.commit()
        conexion.close()

        QMessageBox.information(self, 'Éxito', '¡Administrador registrado correctamente!')

    except sqlite3.Error as e:
        print(f"Error interacting with database: {e}")
        QMessageBox.critical(self, 'Error', f"Error interacting with database: {e}")

def volver_a_login(self):
    try:
        self.login_window = Login()
        self.login_window.show()
        self.close()
    except Exception as e:
        print(f"Error opening login window: {e}")
        QMessageBox.critical(self, 'Error', 'Error opening login window.')

class MainApp(QMainWindow):
    def __init__(self):
        super(MainApp, self).__init__()
        ui_path = resource_path('Opciones1_0.ui')

```

```

loadUi(ui_path, self)
self.setup_ui()

def setup_ui(self):
    self.bt_normal.hide()
    self.click_posicion = None
    self.bt_minimize.clicked.connect(lambda: self.showMinimized())
    self.bt_normal.clicked.connect(self.control_bt_normal)
    self.bt_maximize.clicked.connect(self.control_bt_maximize)
    self.bt_close.clicked.connect(lambda: self.close())

    self.ag_usuario.clicked.connect(self.abrir_agregar_usuario)
    self.buscar_u.clicked.connect(self.abrir_buscar_usuario)
    self.ag_prueba.clicked.connect(self.abrir_agregar_resultado) # Cambiado de
    abrir_agregar_prueba a abrir_agregar_resultado
    self.buscar_p.clicked.connect(self.abrir_administrar_pruebas)

    self.setWindowFlag(Qt.FramelessWindowHint)
    self.setWindowOpacity(1)
    self.setWindowFlags(Qt.FramelessWindowHint)
    self.setAttribute(Qt.WA_TranslucentBackground)

    self.gripSize = 10
    self.grip = QPushButton("", self)
    self.grip.resize(self.gripSize, self.gripSize)
    self.frame_superior.mousePressEvent = self.mover_ventana

def control_bt_normal(self):
    self.showNormal()
    self.bt_normal.hide()
    self.bt_maximize.show()

def control_bt_maximize(self):
    self.showMaximized()
    self.bt_maximize.hide()
    self.bt_normal.show()

def resizeEvent(self, event):
    rect = self.rect()
    self.grip.move(rect.right() - self.gripSize, rect.bottom() - self.gripSize)

def mousePressEvent(self, event):
    self.click_posicion = event.globalPos()

def mover_ventana(self, event):
    if not self.isMaximized():

```

```

    if event.buttons() == Qt.LeftButton:
        self.move(self.pos() + event.globalPos() - self.click_posicion)
        self.click_posicion = event.globalPos()
        event.accept()

def abrir_agregar_usuario(self):
    try:
        self.agregar_usuario_window = AgregarUsuario()
        self.agregar_usuario_window.show()
        self.close()
    except Exception as e:
        print(f"Error opening agregar_user window: {e}")
        QMessageBox.critical(self, 'Error', 'Error opening agregar_user window.')

def abrir_buscar_usuario(self):
    try:
        self.buscar_usuario_window = BuscarUsuario()
        self.buscar_usuario_window.show()
        self.close()
    except Exception as e:
        print(f"Error opening buscar_user window: {e}")
        QMessageBox.critical(self, 'Error', 'Error opening buscar_user window.')

def abrir_agregar_resultado(self):
    try:
        self.agregar_resultado_window = AgregarResultado()
        self.agregar_resultado_window.show()
        self.close()
    except Exception as e:
        print(f"Error opening agregar_resultado window: {e}")
        QMessageBox.critical(self, 'Error', 'Error opening agregar_resultado window.')

def abrir_administrar_pruebas(self):
    try:
        self.administrar_pruebas_window = AdministrarPruebas()
        self.administrar_pruebas_window.show()
        self.close()
    except Exception as e:
        print(f"Error opening administrar_pruebas window: {e}")
        QMessageBox.critical(self, 'Error', 'Error opening administrar_pruebas window.')

class AgregarUsuario(QMainWindow):
    def __init__(self):
        super(AgregarUsuario, self).__init__()
        ui_path = resource_path('agregar_user1_0.ui')

```

```

loadUi(ui_path, self)

# Widgets
self.bt_guardar = self.findChild(QPushButton, 'bt_guardar')
self.bt_menu = self.findChild(QPushButton, 'bt_menu')
self.bt_ir_a_resultados = self.findChild(QPushButton, 'bt_ir_a_resultados') # Nuevo botón
para agregar prueba
self.login = self.findChild(QLineEdit, 'login')
self.password = self.findChild(QLineEdit, 'password')
self.fecha_naci = self.findChild(QLineEdit, 'fecha_naci')
self.talla = self.findChild(QLineEdit, 'talla')
self.identificacion = self.findChild(QLineEdit, 'identificacion')

# Conexión de botones
self.bt_guardar.clicked.connect(self.guardar_usuario)
self.bt_menu.clicked.connect(self.volver_a_opciones)
self.bt_ir_a_resultados.clicked.connect(self.ir_a_resultados) # Conexión del botón para
agregar prueba

self.id_usuario = None # Inicializar id_usuario
self.setup_ui()

def setup_ui(self):
    self.bt_normal.hide()
    self.click_posicion = None
    self.bt_minimize.clicked.connect(lambda: self.showMinimized())
    self.bt_normal.clicked.connect(self.control_bt_normal)
    self.bt_maximize.clicked.connect(self.control_bt_maximize)
    self.bt_close.clicked.connect(lambda: self.close())

    self.setWindowFlag(Qt.FramelessWindowHint)
    self.setWindowOpacity(1)
    self.setWindowFlags(Qt.FramelessWindowHint)
    self.setAttribute(Qt.WA_TranslucentBackground)

    self.gripSize = 10
    self.grip = QPushButton("", self)
    self.grip.resize(self.gripSize, self.gripSize)
    self.frame_superior.mousePressEvent = self.mover_ventana

def control_bt_normal(self):
    self.showNormal()
    self.bt_normal.hide()
    self.bt_maximize.show()

def control_bt_maximize(self):

```



```

self.showMaximized()
self.bt_maximize.hide()
self.bt_normal.show()

def resizeEvent(self, event):
    rect = self.rect()
    self.grip.move(rect.right() - self.gripSize, rect.bottom() - self.gripSize)

def mousePressEvent(self, event):
    self.click_posicion = event.globalPos()

def mover_ventana(self, event):
    if not self.isMaximized():
        if event.buttons() == Qt.LeftButton:
            self.move(self.pos() + event.globalPos() - self.click_posicion)
            self.click_posicion = event.globalPos()
            event.accept()

def guardar_usuario(self):
    login = self.login.text()
    password = self.password.text()
    fecha_naci = self.fecha_naci.text()
    talla = self.talla.text()
    identificacion = self.identificacion.text()

    if not login or not password or not fecha_naci or not talla or not identificacion:
        QMessageBox.warning(self, 'Error', 'Todos los campos son obligatorios.')
        return

    try:
        conexion = sqlite3.connect('BaseDatos1.db')
        cursor = conexion.cursor()

        cursor.execute("""
            INSERT INTO Usuario (login, password, fecha_naci, talla, identificacion)
            VALUES (?, ?, ?, ?, ?)
            """, (login, password, fecha_naci, talla, identificacion))

        conexion.commit()
        self.id_usuario = cursor.lastrowid # Guardar el ID del usuario recién creado para usarlo
luego
        conexion.close()

        QMessageBox.information(self, 'Éxito', '¡Usuario guardado correctamente!')

    except sqlite3.Error as e:

```

```

print(f'Error interacting with database: {e}')
QMessageBox.critical(self, 'Error', f'Error interacting with database: {e}')

def ir_a_resultados(self):
    # Verificar que el usuario haya sido guardado antes de agregar una prueba
    if not self.id_usuario:
        QMessageBox.warning(self, 'Error', 'Primero debe guardar el usuario.')
        return

    try:
        # Abrir la ventana de agregar resultado y pasar el id_usuario
        self.agregar_resultado_window = AgregarResultado()
        self.agregar_resultado_window.id_usuario = self.id_usuario # Pasar el ID del usuario al
nuevo objeto
        self.agregar_resultado_window.show()
        self.close()
    except Exception as e:
        print(f'Error opening agregar_resultado window: {e}')
        QMessageBox.critical(self, 'Error', 'Error opening agregar_resultado window.')

def volver_a_opciones(self):
    try:
        self.opciones_window = MainApp()
        self.opciones_window.show()
        self.close()
    except Exception as e:
        print(f'Error opening opciones window: {e}')
        QMessageBox.critical(self, 'Error', 'Error opening opciones window.')

class AgregarResultado(QMainWindow):
    def __init__(self):
        super(AgregarResultado, self).__init__()
        ui_path = resource_path('agregar_resultado.ui')
        loadUi(ui_path, self)

        # Widgets
        self.bt_guardar = self.findChild(QPushButton, 'bt_guardar')
        self.bt_cargar = self.findChild(QPushButton, 'bt_cargar')
        self.bt_menu = self.findChild(QPushButton, 'bt_menu')
        self.fecha = self.findChild(QLineEdit, 'fecha')
        self.hora = self.findChild(QLineEdit, 'hora')

        # Conexión de botones
        self.bt_guardar.clicked.connect(self.guardar_resultado)
        self.bt_cargar.clicked.connect(self.cargar_video)

```

```

self.bt_menu.clicked.connect(self.volver_a_menu)

self.id_usuario = None # Asegurarse de inicializar el id_usuario
self.setup_ui()

def setup_ui(self):
    self.bt_normal.hide()
    self.click_posicion = None
    self.bt_minimize.clicked.connect(lambda: self.showMinimized())
    self.bt_normal.clicked.connect(self.control_bt_normal)
    self.bt_maximize.clicked.connect(self.control_bt_maximize)
    self.bt_close.clicked.connect(lambda: self.close())

    self.setWindowFlag(Qt.FramelessWindowHint)
    self.setWindowOpacity(1)
    self.setWindowFlags(Qt.FramelessWindowHint)
    self.setAttribute(Qt.WA_TranslucentBackground)

    self.gripSize = 10
    self.grip = QPushButton("", self)
    self.grip.resize(self.gripSize, self.gripSize)
    self.frame_superior.mousePressEvent = self.mover_ventana

def control_bt_normal(self):
    self.showNormal()
    self.bt_normal.hide()
    self.bt_maximize.show()

def control_bt_maximize(self):
    self.showMaximized()
    self.bt_maximize.hide()
    self.bt_normal.show()

def resizeEvent(self, event):
    rect = self.rect()
    self.grip.move(rect.right() - self.gripSize, rect.bottom() - self.gripSize)

def mousePressEvent(self, event):
    self.click_posicion = event.globalPos()

def mover_ventana(self, event):
    if not self.isMaximized():
        if event.buttons() == Qt.LeftButton:
            self.move(self.pos() + event.globalPos() - self.click_posicion)
            self.click_posicion = event.globalPos()
            event.accept()

```

```

def cargar_video(self):
    opciones = QFileDialog.Options()
    archivo, _ = QFileDialog.getOpenFileName(self, "Seleccionar archivo de video", "",
"Archivos de video (*.mp4 *.avi);;Todos los archivos (*)", options=opciones)
    if archivo:
        try:
            # Mostrar el diálogo de progreso antes de procesar el video
            progress_dialog = QProgressDialog("Procesando video, por favor espera...",
"Cancelar", 0, 100, self)
            progress_dialog.setWindowTitle("Procesando Video")
            progress_dialog.setWindowModality(Qt.WindowModal)
            progress_dialog.setValue(0)

            # Procesar el video
            self.procesar_video(archivo, progress_dialog)

            # Cerrar el diálogo de progreso una vez que termine el procesamiento
            progress_dialog.setValue(100)
            QMessageBox.information(self, 'Éxito', '¡Video procesado y datos guardados
correctamente!')

        except Exception as e:
            print(f"Error processing video: {e}")
            QMessageBox.critical(self, 'Error', f"Error processing video: {e}")

def procesar_video(self, archivo_video, progress_dialog):
    # Inicializa Mediapipe y OpenCV
    mp_drawing = mp.solutions.drawing_utils
    mp_hands = mp.solutions.hands
    hands = mp_hands.Hands(max_num_hands=2)

    vclip = cv2.VideoCapture(archivo_video)
    hand_data = []

    # Configuración para procesamiento de video
    width = vclip.get(cv2.CAP_PROP_FRAME_WIDTH)
    height = vclip.get(cv2.CAP_PROP_FRAME_HEIGHT)
    fps = vclip.get(cv2.CAP_PROP_FPS)
    scale_percent = 50
    new_width = int(width * scale_percent / 100)
    new_height = int(height * scale_percent / 100)
    dim = (new_width, new_height)

    frame_index = 0
    total_frames = int(vclip.get(cv2.CAP_PROP_FRAME_COUNT))

```

```

while vclip.isOpened():
    ret, frame = vclip.read()
    if not ret:
        break

    # Procesar cada frame
    frame_resized = cv2.resize(frame, dim, interpolation=cv2.INTER_AREA)
    frame_rgb = cv2.cvtColor(frame_resized, cv2.COLOR_BGR2RGB)
    results = hands.process(frame_rgb)

    if results.multi_hand_landmarks:
        hand_positions = []
        for hand_landmarks in results.multi_hand_landmarks:
            mp_drawing.draw_landmarks(frame_resized, hand_landmarks,
mp_hands.HAND_CONNECTIONS)

            index_finger_tip =
hand_landmarks.landmark[mp_hands.HandLandmark.INDEX_FINGER_TIP]
            finger_x, finger_y = int(index_finger_tip.x * new_width), int(index_finger_tip.y *
new_height)
            hand_positions.append((finger_x, finger_y))

            if len(hand_positions) == 2:
                hand_data.append([frame_index, hand_positions[0][0], hand_positions[0][1],
hand_positions[1][0], hand_positions[1][1]])

        # Actualizar el progreso
        frame_index += 1
        progress = int((frame_index / total_frames) * 100)
        progress_dialog.setValue(progress)

        # Cancelar si el usuario pulsa el botón de cancelar
        if progress_dialog.wasCanceled():
            vclip.release()
            cv2.destroyAllWindows()
            QMessageBox.warning(self, 'Cancelado', 'El procesamiento del video fue cancelado.')
            return

vclip.release()
cv2.destroyAllWindows()

# Guardar los datos en un archivo .npz
archivo_npy = f"hand_tracking_data_{self.id_usuario}.npz"
np.save(archivo_npy, hand_data)

```

```

# Guardar la ruta del archivo .npv en la base de datos asociado al usuario
conexion = sqlite3.connect('BaseDatos1.db')
cursor = conexion.cursor()

cursor.execute("""
    INSERT INTO Resultados (id_usuario, id_prueba, video, posiciones)
    VALUES (?, ?, ?, ?)
""", (self.id_usuario, None, archivo_video, archivo_npy))

conexion.commit()
conexion.close()

def guardar_resultado(self):
    # Implementación de la función para guardar los resultados
    fecha = self.fecha.text()
    hora = self.hora.text()

    try:
        conexion = sqlite3.connect('BaseDatos1.db')
        cursor = conexion.cursor()

        cursor.execute("""
            INSERT INTO Prueba (fecha, hora)
            VALUES (?, ?)
            """, (fecha, hora))

        id_prueba = cursor.lastrowid # Obtener el ID de la prueba recién creada

        cursor.execute("""
            INSERT INTO UsuarioPrueba (id_usuario, id_prueba)
            VALUES (?, ?)
            """, (self.id_usuario, id_prueba))

        conexion.commit()
        conexion.close()

        QMessageBox.information(self, 'Éxito', '¡Resultado guardado correctamente!')
    except sqlite3.Error as e:
        print(f"Error interacting with database: {e}")
        QMessageBox.critical(self, 'Error', f"Error interacting with database: {e}")

def volver_a_menu(self):
    try:
        self.opciones_window = MainApp()
        self.opciones_window.show()
        self.close()

```

```

except Exception as e:
    print(f"Error opening menu window: {e}")
    QMessageBox.critical(self, 'Error', 'Error opening menu window.')

```

```

class AdministrarPruebas(QMainWindow):
    def __init__(self):
        super(AdministrarPruebas, self).__init__()
        ui_path = resource_path('administrar_pruebas.ui')
        loadUi(ui_path, self)

        # Conectar botones
        self.bt_eliminar_usuario = self.findChild(QPushButton, 'bt_eliminar_usuario')
        self.bt_eliminar_prueba = self.findChild(QPushButton, 'bt_eliminar_prueba')
        self.bt_actualizar_prueba = self.findChild(QPushButton, 'bt_actualizar_prueba')
        self.bt_menu = self.findChild(QPushButton, 'bt_menu')

        # Conectar botones a funciones
        self.bt_eliminar_usuario.clicked.connect(self.eliminar_usuario)
        self.bt_eliminar_prueba.clicked.connect(self.eliminar_prueba)
        self.bt_actualizar_prueba.clicked.connect(self.actualizar_prueba)
        self.bt_menu.clicked.connect(self.volver_a_menu)

        # Cargar datos en la tabla
        self.cargar_datos()
        self.setup_ui()

    def setup_ui(self):
        self.bt_normal.hide()
        self.click_posicion = None
        self.bt_minimize.clicked.connect(lambda: self.showMinimized())
        self.bt_normal.clicked.connect(self.control_bt_normal)
        self.bt_maximize.clicked.connect(self.control_bt_maximize)
        self.bt_close.clicked.connect(lambda: self.close())

        self.setWindowFlag(Qt.FramelessWindowHint)
        self.setWindowOpacity(1)
        self.setWindowFlags(Qt.FramelessWindowHint)
        self.setAttribute(Qt.WA_TranslucentBackground)

        self.gripSize = 10
        self.grip = QPushButton("", self)
        self.grip.resize(self.gripSize, self.gripSize)
        self.frame_superior.mousePressEvent = self.mover_ventana

    def control_bt_normal(self):
        self.showNormal()

```

```

self.bt_normal.hide()
self.bt_maximize.show()

def control_bt_maximize(self):
    self.showMaximized()
    self.bt_maximize.hide()
    self.bt_normal.show()

def resizeEvent(self, event):
    rect = self.rect()
    self.grip.move(rect.right() - self.gripSize, rect.bottom() - self.gripSize)

def mousePressEvent(self, event):
    self.click_posicion = event.globalPos()

def mover_ventana(self, event):
    if not self.isMaximized():
        if event.buttons() == Qt.LeftButton:
            self.move(self.pos() + event.globalPos() - self.click_posicion)
            self.click_posicion = event.globalPos()
            event.accept()

def cargar_datos(self):
    try:
        conexion = sqlite3.connect('BaseDatos1.db')
        cursor = conexion.cursor()

        cursor.execute("""
            SELECT Usuario.id_usuario, Usuario.login, Prueba.id_prueba, Prueba.fecha,
Prueba.hora
            FROM Usuario
            LEFT JOIN UsuarioPrueba ON Usuario.id_usuario = UsuarioPrueba.id_usuario
            LEFT JOIN Prueba ON UsuarioPrueba.id_prueba = Prueba.id_prueba
            """)
        resultados = cursor.fetchall()

        self.tableWidget.setRowCount(len(resultados))
        self.tableWidget.setColumnCount(5)
        headers = ["ID Usuario", "Nombre Estudiante", "ID Prueba", "Fecha Prueba", "Hora
Prueba"]
        self.tableWidget.setHorizontalHeaderLabels(headers)

        for i, fila in enumerate(resultados):
            for j, dato in enumerate(fila):
                self.tableWidget.setItem(i, j, QTableWidgetItem(str(dato)))

```



```

        conexion.close()
    except sqlite3.Error as e:
        print(f"Error al cargar datos: {e}")
        QMessageBox.critical(self, 'Error', f"Error al cargar datos: {e}")

def eliminar_usuario(self):
    selected_row = self.tableWidget.currentRow()
    if selected_row == -1:
        QMessageBox.warning(self, 'Error', 'Selecciona un usuario para eliminar.')
        return

    id_usuario = self.tableWidget.item(selected_row, 0).text()

    confirm = QMessageBox.question(self, 'Confirmar Eliminación',
                                   f"¿Estás seguro de que deseas eliminar al usuario con ID {id_usuario}?
    Todas sus pruebas también serán eliminadas.",
                                   QMessageBox.Yes | QMessageBox.No)
    if confirm == QMessageBox.Yes:
        try:
            conexion = sqlite3.connect('BaseDatos1.db')
            cursor = conexion.cursor()

            cursor.execute("DELETE FROM UsuarioPrueba WHERE id_usuario = ?",
                           (id_usuario,))
            cursor.execute("DELETE FROM Prueba WHERE id_prueba IN (SELECT id_prueba
            FROM UsuarioPrueba WHERE id_usuario = ?)", (id_usuario,))
            cursor.execute("DELETE FROM Usuario WHERE id_usuario = ?", (id_usuario,))

            conexion.commit()
            conexion.close()

            self.cargar_datos()
            QMessageBox.information(self, 'Éxito', 'Usuario y pruebas eliminadas correctamente.')
        except sqlite3.Error as e:
            print(f"Error al eliminar usuario: {e}")
            QMessageBox.critical(self, 'Error', f"Error al eliminar usuario: {e}")

def eliminar_prueba(self):
    selected_row = self.tableWidget.currentRow()
    if selected_row == -1:
        QMessageBox.warning(self, 'Error', 'Selecciona una prueba para eliminar.')
        return

    id_prueba = self.tableWidget.item(selected_row, 2).text()

    confirm = QMessageBox.question(self, 'Confirmar Eliminación',

```

```

        f"¿Estás seguro de que deseas eliminar la prueba con ID {id_prueba}?",
        QMessageBox.Yes | QMessageBox.No)
if confirm == QMessageBox.Yes:
    try:
        conexion = sqlite3.connect('BaseDatos1.db')
        cursor = conexion.cursor()

        cursor.execute("DELETE FROM Prueba WHERE id_prueba = ?", (id_prueba,))
        cursor.execute("DELETE FROM UsuarioPrueba WHERE id_prueba = ?",
(id_prueba,))

        conexion.commit()
        conexion.close()

        self.cargar_datos()
        QMessageBox.information(self, 'Éxito', 'Prueba eliminada correctamente.')
    except sqlite3.Error as e:
        print(f"Error al eliminar prueba: {e}")
        QMessageBox.critical(self, 'Error', f"Error al eliminar prueba: {e}")

def actualizar_prueba(self):
    selected_row = self.tableWidget.currentRow()
    if selected_row == -1:
        QMessageBox.warning(self, 'Error', 'Selecciona un usuario para actualizar la prueba.')
        return

    id_usuario = self.tableWidget.item(selected_row, 0).text()

    # Abrir la nueva ventana para actualizar la prueba
    try:
        self.agregar_resultado_window = AgregarResultado()
        self.agregar_resultado_window.id_usuario = id_usuario # Pasar el ID del usuario a la
nueva ventana
        self.agregar_resultado_window.show()
        self.close()
    except Exception as e:
        print(f"Error opening agregar_resultado window: {e}")
        QMessageBox.critical(self, 'Error', 'Error opening agregar_resultado window.')

def volver_a_menu(self):
    try:
        self.opciones_window = MainApp()
        self.opciones_window.show()
        self.close()
    except Exception as e:
        print(f"Error opening menu window: {e}")

```

```

        QMessageBox.critical(self, 'Error', 'Error opening menu window.')

class BuscarUsuario(QMainWindow):
    def __init__(self):
        super(BuscarUsuario, self).__init__()
        ui_path = resource_path('buscar_us1_0.ui')
        loadUi(ui_path, self)

        # Widgets
        self.bt_menu = self.findChild(QPushButton, 'bt_menu')
        self.bt_mostrar_archivos = self.findChild(QPushButton, 'bt_mostrar_archivos')
        self.bt_graficar = self.findChild(QPushButton, 'bt_graficar')
        self.bt_buscar = self.findChild(QPushButton, 'bt_buscar') # Asegúrate de tener el botón
        buscar
        self.identificacion = self.findChild(QLineEdit, 'identificacion')
        self.tableWidget = self.findChild(QTableWidget, 'tableWidget')
        self.frame_grafica = self.findChild(QFrame, 'frame_grafica')
        self.grafica_velocidad = self.findChild(QFrame, 'grafica_velocidad')

        # Conectar botones
        self.bt_menu.clicked.connect(self.volver_a_opciones)
        self.bt_mostrar_archivos.clicked.connect(self.mostrar_archivos)
        self.bt_graficar.clicked.connect(self.graficar_desde_tabla)
        self.bt_buscar.clicked.connect(self.buscar_usuario) # Conectar el botón buscar al método
        buscar_usuario

        self.setup_ui()

    def setup_ui(self):
        self.bt_normal.hide()
        self.click_posicion = None
        self.bt_minimize.clicked.connect(lambda: self.showMinimized())
        self.bt_normal.clicked.connect(self.control_bt_normal)
        self.bt_maximize.clicked.connect(self.control_bt_maximize)
        self.bt_close.clicked.connect(lambda: self.close())

        self.setWindowFlag(Qt.FramelessWindowHint)
        self.setWindowOpacity(1)
        self.setWindowFlags(Qt.FramelessWindowHint)
        self.setAttribute(Qt.WA_TranslucentBackground)

        self.gripSize = 10
        self.grip = QPushButton("", self)
        self.grip.resize(self.gripSize, self.gripSize)
        self.frame_superior.mousePressEvent = self.mover_ventana

```

```

def control_bt_normal(self):
    self.showNormal()
    self.bt_normal.hide()
    self.bt_maximize.show()

def control_bt_maximize(self):
    self.showMaximized()
    self.bt_maximize.hide()
    self.bt_normal.show()

def resizeEvent(self, event):
    rect = self.rect()
    self.grip.move(rect.right() - self.gripSize, rect.bottom() - self.gripSize)

def mousePressEvent(self, event):
    self.click_posicion = event.globalPos()

def mover_ventana(self, event):
    if not self.isMaximized():
        if event.buttons() == Qt.LeftButton:
            self.move(self.pos() + event.globalPos() - self.click_posicion)
            self.click_posicion = event.globalPos()
            event.accept()

def volver_a_opciones(self):
    try:
        self.opciones_window = MainApp()
        self.opciones_window.show()
        self.close()
    except Exception as e:
        print(f"Error opening opciones window: {e}")
        QMessageBox.critical(self, 'Error', 'Error opening opciones window.')

def buscar_usuario(self):
    identificacion = self.identificacion.text().strip()

    # Verificar si la identificación está vacía
    if not identificacion:
        QMessageBox.warning(self, 'Error', 'Por favor, introduce una identificación.')
        return

    try:
        # Conectar a la base de datos
        conexion = sqlite3.connect('BaseDatos1.db')
        cursor = conexion.cursor()

```

```

# Mensaje de depuración para ver qué identificación se busca
print(f"Buscando usuario con identificación: {identificacion}")

# Realizar la consulta para buscar al usuario por la identificación
cursor.execute("""
SELECT * FROM Usuario WHERE identificacion = ?
""", (identificacion,))
usuario = cursor.fetchone()

# Mensaje de depuración para ver el resultado de la consulta
print(f"Resultado de la consulta: {usuario}")

# Verificar si se encontró el usuario
if usuario:
    # Mostrar los detalles del usuario en los campos correspondientes
    self.mostrar_usuario(usuario)

    # Buscar la fecha de la última prueba asociada al usuario
    cursor.execute("""
SELECT Prueba.fecha
FROM Prueba
JOIN UsuarioPrueba ON Prueba.id_prueba = UsuarioPrueba.id_prueba
WHERE UsuarioPrueba.id_usuario = ?
ORDER BY Prueba.fecha DESC LIMIT 1
""", (usuario[0],))
    prueba = cursor.fetchone()

    # Mostrar la fecha de la última prueba si existe
    if prueba:
        print(f"Fecha de la última prueba: {prueba[0]}")
        self.fecha_prueba.setText(prueba[0]) # Mostrar la fecha en el campo fecha_prueba
    else:
        print("No se encontraron pruebas registradas.")
        self.fecha_prueba.setText('Sin pruebas registradas') # Mostrar mensaje si no hay
pruebas

else:
    print("Usuario no encontrado.")
    QMessageBox.warning(self, 'Error', 'Usuario no encontrado.')

# Cerrar la conexión a la base de datos
conexion.close()

except sqlite3.Error as e:
    # En caso de error con la base de datos, mostrar un mensaje

```

```

print(f'Error interacting with database: {e}')
QMessageBox.critical(self, 'Error', f'Error interacting with database: {e}')

def mostrar_usuario(self, usuario):
    # Verifica que los widgets estén definidos correctamente
    try:
        self.login.setText(usuario[1]) # Nombre del usuario
        self.fecha_naci.setText(usuario[3]) # Fecha de nacimiento
        self.talla.setText(usuario[4]) # Talla del usuario
        self.identificacion.setText(usuario[5]) # Identificación del usuario

        # Opcionalmente, podrías agregar una verificación para asegurarte de que no sean valores
nulos
        print(f'Datos mostrados en la interfaz: Login - {usuario[1]}, Fecha Nac. - {usuario[3]},
Talla - {usuario[4]}, Identificación - {usuario[5]}")
    except AttributeError as e:
        print(f'Error mostrando datos del usuario: {e}')
        QMessageBox.critical(self, 'Error', f'Error mostrando datos del usuario: {e}')

def mostrar_archivos(self):
    identificacion = self.identificacion.text().strip()
    if not identificacion:
        QMessageBox.warning(self, 'Error', 'Por favor, introduce una identificación.')
        return

    try:
        conexion = sqlite3.connect('BaseDatos1.db')
        cursor = conexion.cursor()

        # Buscar el ID del usuario basado en la identificación
        cursor.execute("""
SELECT id_usuario FROM Usuario WHERE identificacion = ?
""", (identificacion,))
        result = cursor.fetchone()

        if result:
            id_usuario = result[0]
            # Buscar archivos asociados al usuario en la tabla Resultados
            cursor.execute("""SELECT * FROM Resultados WHERE id_usuario = ?""",
(id_usuario,))
            archivos = cursor.fetchall()

            if archivos:
                self.mostrar_archivos_en_tabla(archivos)
            else:

```

```

        QMessageBox.information(self, 'Información', 'El usuario no tiene archivos
guardados.')
    else:
        QMessageBox.warning(self, 'Error', 'Usuario no encontrado.')

    conexion.close()
except sqlite3.Error as e:
    print(f"Error fetching files: {e}")
    QMessageBox.critical(self, 'Error', f"Error fetching files: {e}")

def mostrar_archivos_en_tabla(self, archivos):
    # Limpiar la tabla antes de llenarla
    self.tableWidget.setRowCount(0)
    self.tableWidget.setColumnCount(4) # Establecemos la cantidad de columnas necesaria
    headers = ["ID Usuario", "Número de Frame", "Posición X", "Posición Y"]
    self.tableWidget.setHorizontalHeaderLabels(headers)

    for archivo in archivos:
        # El archivo debe contener el id_usuario, el número de frame, posición x e y
        id_usuario = archivo[0] # Asegúrate de que el primer valor sea id_usuario
        posiciones_data = np.load(archivo[4], allow_pickle=True) # Cargar archivo .npy
        (asegúrate de que archivo[4] sea la ruta correcta)

        # Mostrar los datos de cada frame en la tabla
        for fila in posiciones_data:
            row_position = self.tableWidget.rowCount()
            self.tableWidget.insertRow(row_position)

            # Rellenar las celdas con los valores correspondientes
            self.tableWidget.setItem(row_position, 0, QTableWidgetItem(str(id_usuario))) # ID
del usuario
            self.tableWidget.setItem(row_position, 1, QTableWidgetItem(str(fila[0]))) #
Número de frame
            self.tableWidget.setItem(row_position, 2, QTableWidgetItem(str(fila[1]))) #
Posición X
            self.tableWidget.setItem(row_position, 3, QTableWidgetItem(str(fila[2]))) #
Posición Y

def graficar_desde_tabla(self):
    try:
        # Establecer el valor de FPS (por ejemplo, 240)
        self.fps = 240 # Asegúrate de usar el valor correcto según tu video

        # Verificar si hay datos en la tabla
        rowCount = self.tableWidget.rowCount()
        if rowCount == 0:

```

```

    QMessageBox.warning(self, 'Error', 'No hay datos para graficar.')
    return

# Extraer los datos de la tabla
frames = []
posiciones_x = []
posiciones_y = []

for row in range(rowCount):
    frame_item = self.tableWidget.item(row, 1)
    pos_x_item = self.tableWidget.item(row, 2)
    pos_y_item = self.tableWidget.item(row, 3)

    if frame_item is None or pos_x_item is None or pos_y_item is None:
        QMessageBox.warning(self, 'Error', f'Datos no válidos en la fila {row}. Una o
más celdas son None.')
        return

    frames.append(float(frame_item.text()))
    posiciones_x.append(float(pos_x_item.text()))
    posiciones_y.append(float(pos_y_item.text()))

# Conversión a NumPy arrays para graficar
frames = np.array(frames)
posiciones_x = np.array(posiciones_x)

# Detectar los puntos donde la mano cambia de dirección (máximos y mínimos
locales)
peaks, _ = find_peaks(posiciones_x, distance=20, prominence=40) # Detectar
máximos locales
valleys, _ = find_peaks(-posiciones_x, distance=20, prominence=40) # Detectar
mínimos locales

# Obtener los valores de los picos y valles para promediar
peak_values = posiciones_x[peaks]
valley_values = posiciones_x[valleys]

# Calcular los promedios
avg_peak = np.mean(peak_values)
avg_valley = np.mean(valley_values)

# Definir un umbral de tolerancia (puedes ajustar este valor según sea necesario)
tolerance = 30 # Por ejemplo, 30 píxeles de tolerancia

# Filtrar los picos y valles fuera del rango de tolerancia
filtered_peaks = peaks[np.abs(peak_values - avg_peak) < tolerance]

```



```

filtered_valleys = valleys[np.abs(valley_values - avg_valley) < tolerance]

# Unir los picos y valles filtrados y ordenarlos
switchpoints = np.sort(np.concatenate((filtered_peaks, filtered_valleys)))

# Crear la gráfica principal (Movimiento de la Mano Dominante y Taps)
fig, ax1 = plt.subplots(figsize=(10, 6))
ax1.plot(frames / self.fps, posiciones_x, marker='.', c='g', label='Movimiento de la
mano dominante (Posición X)')
ax1.set_xlabel('Tiempo (segundos)')
ax1.set_ylabel('Posición horiz. (pixel)')
ax1.set_title('Detección de Taps en el Tapping Test (Puntos Extremos)')

# Agregar los taps detectados en la gráfica
if len(switchpoints) > 0:
    ax1.scatter(frames[switchpoints] / self.fps, posiciones_x[switchpoints], marker='o',
c='r', label='Taps (cambios de dirección)')
    ax1.legend()

# Mostrar el número total de taps detectados
num_taps = len(switchpoints)
ax1.text(0.05, 0.95, f'Total de taps: {num_taps}', transform=ax1.transAxes,
fontsize=12, color='black',
bbox=dict(facecolor='white', alpha=0.5))

# Integrar la primera figura en la interfaz
canvas1 = FigureCanvas(fig)
layout1 = QVBoxLayout()
layout1.addWidget(canvas1)

# Limpiar el layout existente en frame_grafica
if self.frame_grafica.layout() is not None:
    old_layout = self.frame_grafica.layout()
    while old_layout.count():
        child = old_layout.takeAt(0)
        if child.widget():
            child.widget().deleteLater()
    old_layout.deleteLater()

self.frame_grafica.setLayout(layout1)

# Crear el gráfico de tiempo entre *taps* en segundos con una gráfica de líneas
if len(switchpoints) > 1:
    times_between_taps = np.diff(frames[switchpoints]) / self.fps # Diferencias entre
los frames sucesivos en segundos

```

```

fig2, ax2 = plt.subplots(figsize=(10, 6))
ax2.plot(range(1, len(times_between_taps) + 1), times_between_taps, marker='o',
color='blue', linestyle='-', label='Tiempo entre Taps')
ax2.set_xlabel('Número de Tap')
ax2.set_ylabel('Tiempo entre taps (segundos)')
ax2.set_title('Tiempo entre Taps (en segundos)')
ax2.grid()

# Ajuste para que el primer *tap* esté en el tiempo cero
ax2.set_ylim(bottom=0)

# Añadir interactividad para mostrar la información exacta con 3 decimales
cursor = mplcursors.cursor(ax2, hover=True)
cursor.connect("add", lambda sel: sel.annotation.set_text(f'Tap
{int(sel.target[0])}\nTiempo: {sel.target[1]:.3f} s'))

# Integrar la segunda figura en la interfaz
canvas2 = FigureCanvas(fig2)
layout2 = QVBoxLayout()
layout2.addWidget(canvas2)

# Limpiar el layout existente en grafica_velocidad
if self.grafica_velocidad.layout() is not None:
    old_layout = self.grafica_velocidad.layout()
    while old_layout.count():
        child = old_layout.takeAt(0)
        if child.widget():
            child.widget().deleteLater()
    old_layout.deleteLater()

self.grafica_velocidad.setLayout(layout2)

except Exception as e:
    print(f'Error al graficar: {e}')
    QMessageBox.critical(self, 'Error', f'Error al graficar: {e}')

def integrar_grafica_en_interfaz(self, figura, layout_widget):
    try:
        # Crear el canvas con la figura proporcionada
        canvas = FigureCanvas(figura)

        # Intentar obtener el layout existente del widget donde se agregará la gráfica
        layout = layout_widget.layout()

        # Si no existe un layout, crear uno nuevo y asignarlo al widget
        if layout is None:

```

```

        print("El layout no existe. Se va a crear un nuevo layout.")
        layout = QVBoxLayout(layout_widget)
        layout_widget.setLayout(layout)
    else:
        print("Layout existente encontrado.")

    # Eliminar gráficos anteriores del layout
    for i in reversed(range(layout.count())):
        child = layout.takeAt(i)
        if child.widget():
            child.widget().setParent(None)
            print(f"Gráfico eliminado del layout en la posición {i}.")

    # Añadir el nuevo gráfico al layout
    layout.addWidget(canvas)
    print("Nuevo gráfico agregado al layout.")
except Exception as e:
    print(f"Error en integrar_grafica_en_interfaz: {e}")
    QMessageBox.critical(self, 'Error', f"Error en integrar_grafica_en_interfaz: {e}")

if __name__ == '__main__':

    try:
        app = QApplication(sys.argv)
        db_manager = DatabaseManager()

        login_window = Login()
        login_window.show()
        sys.exit(app.exec_())
    except Exception as e:
        print(f"Unhandled error: {e}")
        sys.exit(1)

```

### **Codigo para importar la personalización de los botones**

```

import sys
import os
from PyQt5.QtGui import QIcon
from PyQt5.QtCore import QSize

def resource_path(relative_path):
    """ Obtiene la ruta absoluta al recurso, tanto en desarrollo como en el ejecutable. """
    base_path = getattr(sys, '_MEIPASS', os.path.dirname(os.path.abspath(__file__)))
    return os.path.join(base_path, relative_path)

```

```

icon = QIcon()
icon.addFile(u"Mini.svg", QSize(), QIcon.Normal, QIcon.Off)

icon1 = QIcon()
icon1.addFile(u"Normal.svg", QSize(), QIcon.Normal, QIcon.Off)

icon2 = QIcon()
icon2.addFile(u"Casa.svg", QSize(), QIcon.Normal, QIcon.Off)

icon3 = QIcon()
icon3.addFile(u"CerrarAzul.svg", QSize(), QIcon.Normal, QIcon.Off)

```

### **Código para estimar la confiabilidad del sistema en comparación con Kinovea**

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_absolute_error, mean_squared_error
import mplcursors

# Cargar los datos generados de la mano dominante (de tu sistema)
data_file = "hand_tracking_data_12.npy"
foundarr = np.load(data_file).astype(np.float64)

# Cargar los datos de Kinovea desde el archivo Excel
kinovea_data = pd.read_excel('VID_1_DATOS.xlsx')

# Parámetros
fps = 240.0 # Asegúrate de usar el mismo valor de FPS para ambos sistemas
min_tap_distance = 10 # Distancia mínima en frames para evitar taps duplicados muy cercanos

# Eliminar puntos medios entre ciertos límites
mindel = foundarr[:, 1].max() * 0.35
maxdel = foundarr[:, 1].max() * 0.95
darm = np.delete(foundarr, np.where((foundarr[:, 1] > mindel) & (foundarr[:, 1] < maxdel)),
axis=0)

# Detectar los puntos de cambio de dirección (taps)
switchpoints = []
n_around = 10
for i in range(n_around, darm.shape[0] - n_around):
    curr_slope = (darm[i, 1] - darm[i - 1, 1]) / (darm[i, 0] - darm[i - 1, 0])
    prev_slope = (darm[i - 1, 1] - darm[i - 2, 1]) / (darm[i - 1, 0] - darm[i - 2, 0])
    if np.sign(curr_slope) != np.sign(prev_slope):
        switchpoints.append((darm[i - 1][0], darm[i - 1][1]))

```

```

# Convertir a un array
switchpoints = np.array(switchpoints)

# Filtrar taps duplicados al inicio
filtered_switchpoints = [switchpoints[0]]
for i in range(1, len(switchpoints)):
    if switchpoints[i][0] > 20: # Evitar taps en los primeros 20 frames
        filtered_switchpoints.append(switchpoints[i])
    elif switchpoints[i][0] - switchpoints[i - 1][0] > 20:
        filtered_switchpoints.append(switchpoints[i])

filtered_switchpoints = np.array(filtered_switchpoints)

# Filtrar taps cercanos
final_switchpoints = [filtered_switchpoints[0]]
for i in range(1, len(filtered_switchpoints)):
    if filtered_switchpoints[i][0] - filtered_switchpoints[i - 1][0] > min_tap_distance:
        final_switchpoints.append(filtered_switchpoints[i])

final_switchpoints = np.array(final_switchpoints)

# Asegurarse de que los taps importantes estén presentes
for frame in [115, 126, 137, 145]:
    tap_frame = darm[np.isclose(darm[:, 0], frame, atol=5)]
    if len(tap_frame) > 0:
        final_switchpoints = np.vstack([final_switchpoints, tap_frame[0][:2]])

# Obtener los tiempos entre taps (de tu sistema)
times_between_taps = np.diff(final_switchpoints[:, 0]) / fps

# Obtener los tiempos entre taps de Kinovea a partir de la columna 'Time (s)'
kinovea_times = kinovea_data['Time (s)'].values
kinovea_times_between_taps = np.diff(kinovea_times) # Diferencias entre los tiempos
sucesivos de Kinovea

# Sincronizar ambos sistemas para que tengan el mismo número de taps
min_len = min(len(times_between_taps), len(kinovea_times_between_taps))
times_between_taps = times_between_taps[:min_len]
kinovea_times_between_taps = kinovea_times_between_taps[:min_len]

# Comparar los tiempos entre taps
mae_taps = mean_absolute_error(kinovea_times_between_taps, times_between_taps)
rmse_taps = np.sqrt(mean_squared_error(kinovea_times_between_taps, times_between_taps))
correlation_taps = np.corrcoef(kinovea_times_between_taps, times_between_taps)[0, 1]

# Calcular el error relativo absoluto por tap

```

```

relative_errors = np.abs(times_between_taps - kinovea_times_between_taps) /
kinovea_times_between_taps * 100

# Calcular el promedio del error relativo absoluto
mean_relative_error = np.mean(relative_errors)

# Confiabilidad en porcentaje
reliability_percentage = 100 - mean_relative_error

print(f"MAE en tiempos entre taps: {mae_taps:.2f} segundos")
print(f"RMSE en tiempos entre taps: {rmse_taps:.2f} segundos")
print(f"Coeficiente de correlación en tiempos entre taps: {correlation_taps:.2f}")
print(f"Error promedio relativo absoluto: {mean_relative_error:.2f}%")
print(f"Confiabilidad del sistema: {reliability_percentage:.2f}%")

# === GRAFICAS ===

# 1. Gráfico comparativo de los tiempos entre taps
plt.figure(num="Figura 1", figsize=(10, 6))
plt.plot(range(1, len(kinovea_times_between_taps) + 1), kinovea_times_between_taps,
label='Kinovea', marker='o')
plt.plot(range(1, len(times_between_taps) + 1), times_between_taps, label='Tu sistema',
marker='x')
plt.xlabel('Número de Tap')
plt.ylabel('Tiempo entre taps (segundos)')
plt.title('Comparación de tiempos entre taps: Kinovea vs Tu sistema')
plt.legend()
plt.grid()

# 2. Gráfico de las posiciones X
plt.figure(num="Figura 2", figsize=(10, 6))
plt.plot(kinovea_data["Time (s)"], kinovea_data["X (px)"], label='Kinovea X', marker='o')
plt.plot(final_switchpoints[:, 0] / fps, final_switchpoints[:, 1], label='Tu sistema X', marker='x')
plt.xlabel('Tiempo (segundos)')
plt.ylabel('Posición X (px)')
plt.title('Comparación de posiciones X entre Kinovea y tu sistema')
plt.legend()
plt.grid()

# 3. Gráfico de las posiciones Y
plt.figure(num="Figura 3", figsize=(10, 6))
plt.plot(kinovea_data["Time (s)"], kinovea_data["Y (px)"], label='Kinovea Y', marker='o')
plt.plot(final_switchpoints[:, 0] / fps, final_switchpoints[:, 1], label='Tu sistema Y', marker='x')
plt.xlabel('Tiempo (segundos)')
plt.ylabel('Posición Y (px)')
plt.title('Comparación de posiciones Y entre Kinovea y tu sistema')

```

```
plt.legend()
plt.grid()
```

```
# 4. Gráfico de la velocidad entre taps en tu sistema
```

```
speed = np.diff(final_switchpoints[:, 1]) / np.diff(final_switchpoints[:, 0] / fps)
plt.figure(num="Figura 4", figsize=(10, 6))
plt.plot(final_switchpoints[1:, 0] / fps, speed, marker='o', label='Velocidad entre Taps (px/seg)')
plt.xlabel('Tiempo (segundos)')
plt.ylabel('Velocidad (px/seg)')
plt.title('Velocidad de la mano dominante entre taps (Tu sistema)')
plt.legend()
plt.grid()
```

```
# 5. Gráfico de barras del tiempo entre taps con puntero
```

```
plt.figure(num="Figura 5", figsize=(10, 6))
bars1 = plt.bar(range(1, len(times_between_taps) + 1), times_between_taps, color='blue',
label='Tu sistema')
bars2 = plt.bar(range(1, len(kinovea_times_between_taps) + 1), kinovea_times_between_taps,
color='orange', alpha=0.6, label='Kinovea')
plt.xlabel('Número de Tap')
plt.ylabel('Tiempo entre taps (segundos)')
plt.title('Comparación del tiempo entre taps en el Tapping Test')
plt.legend()
plt.grid()
```

```
# Añadir interactividad con mplcursors para mostrar los valores exactos en las barras
```

```
cursor = mplcursors.cursor(bars1 + bars2, hover=True)
cursor.connect("add", lambda sel: sel.annotation.set_text(f'Tap {int(sel.target[0])+1}\nTiempo:
{sel.target[1]:.3f} s'))
```

```
# Mostrar todas las gráficas al mismo tiempo
```

```
plt.show()
```

 <p>UNIVERSIDAD <b>CESMAG</b> NIT: 800.109.387-7 VIGILADA MINEDUCACIÓN</p>	<b>CARTA DE ENTREGA TRABAJO DE GRADO O TRABAJO DE APLICACIÓN – ASESOR(A)</b>	<b>CÓDIGO:</b> AAC-BL-FR-032
		<b>VERSIÓN:</b> 1
		<b>FECHA:</b> 09/JUN/2022

San Juan de Pasto, 26 de noviembre de 2024

Biblioteca  
**REMIGIO FIORE FORTEZZA OFM. CAP.**  
Universidad CESMAG  
Pasto

Saludo de paz y bien.


Por medio de la presente se hace entrega del Trabajo de Grado / Trabajo de Aplicación denominado Software Para La Estimación De Posición y Velocidad De La Prueba Tapping Test Mediante Visión Por Computador, presentado por el (los) autor(es) Alvaro Sebastian Montenegro Chamorro, y Paulo Fabian Alexander Zambrano Urbina del Programa Académico Ingeniería electrónica al correo electrónico biblioteca.trabajosdegrado@unicesmag.edu.co. Manifiesto como asesor(a), que su contenido, resumen, anexos y formato PDF cumple con las especificaciones de calidad, guía de presentación de Trabajos de Grado o de Aplicación, establecidos por la Universidad CESMAG, por lo tanto, se solicita el paz y salvo respectivo.

Atentamente,

-----  


**Mario Fernando Henao Rosero**  
c.c. 4612719  
Programa de Ing. Electrónica  
3163455039  
mfhenao@unicesmag.edu.co




 <b>UNIVERSIDAD CESMAG</b> <small>NIT: 800.109.387-7 VIGILADA MINEDUCACIÓN</small>	<b>AUTORIZACIÓN PARA PUBLICACIÓN DE TRABAJOS DE GRADO O TRABAJOS DE APLICACIÓN EN REPOSITORIO INSTITUCIONAL</b>	<b>CÓDIGO:</b> AAC-BL-FR-031
		<b>VERSIÓN:</b> 1
		<b>FECHA:</b> 09/JUN/2022

<b>INFORMACIÓN DEL (LOS) AUTOR(ES)</b>	
<b>Nombres y apellidos del autor:</b> Alvaro Sebastian Montenegro Chamorro	<b>Documento de identidad:</b> 1010138332
<b>Correo electrónico:</b> Sebasmontenegro44@gmail.com	<b>Número de contacto:</b> 3143931802
<b>Nombres y apellidos del autor:</b> Paulo Fabian Alexander Zambrano Urbina	<b>Documento de identidad:</b> 1085324394
<b>Correo electrónico:</b> fabiko.fz@gmail.com	<b>Número de contacto:</b> 3165074150
<b>Nombres y apellidos del autor:</b>	<b>Documento de identidad:</b>
<b>Correo electrónico:</b>	<b>Número de contacto:</b>
<b>Nombres y apellidos del autor:</b>	<b>Documento de identidad:</b>
<b>Correo electrónico:</b>	<b>Número de contacto:</b>
<b>Nombres y apellidos del asesor:</b> Mario Fernando Henao Rosero	<b>Documento de identidad:</b> 4612719
<b>Correo electrónico:</b> mfhenao@unicesmag.edu.co	<b>Número de contacto:</b> 3163455039
<b>Título del trabajo de grado:</b> Software Para La Estimación De Posición y Velocidad De La Prueba Tapping Test Mediante Visión Por Computador	
<b>Facultad y Programa Académico:</b> Facultad de Ingeniería – Programa Ingeniería electrónica	

En mi (nuestra) calidad de autor(es) y/o titular (es) del derecho de autor del Trabajo de Grado o de Aplicación señalado en el encabezado, confiero (conferimos) a la Universidad CESMAG una licencia no exclusiva, limitada y gratuita, para la inclusión del trabajo de grado en el repositorio institucional. Por consiguiente, el alcance de la licencia que se otorga a través del presente documento, abarca las siguientes características:

- a) La autorización se otorga desde la fecha de suscripción del presente documento y durante todo el término en el que el (los) firmante(s) del presente documento conserve (mos) la titularidad de los derechos patrimoniales de autor. En el evento en el que deje (mos) de tener la titularidad de los derechos patrimoniales sobre el Trabajo de Grado o de Aplicación, me (nos) comprometo (comprometemos) a informar de manera inmediata sobre dicha situación a la Universidad CESMAG. Por consiguiente, hasta que no exista comunicación escrita de mi(nuestra) parte informando sobre dicha situación, la Universidad CESMAG se encontrará debidamente habilitada para continuar con la publicación del Trabajo de Grado o de Aplicación dentro del repositorio institucional. Conozco(conocemos) que esta autorización podrá revocarse en cualquier momento, siempre y cuando se eleve la solicitud por escrito para dicho fin ante la Universidad CESMAG. En estos eventos, la Universidad CESMAG cuenta con el plazo de un mes después de recibida la petición, para desmarcar la visualización del Trabajo de Grado o de Aplicación del repositorio institucional.

 <b>UNIVERSIDAD CESMAG</b> <small>NIT: 800.109.387-7 VIGILADA MINEDUCACIÓN</small>	<b>AUTORIZACIÓN PARA PUBLICACIÓN DE TRABAJOS DE GRADO O TRABAJOS DE APLICACIÓN EN REPOSITORIO INSTITUCIONAL</b>	<b>CÓDIGO:</b> AAC-BL-FR-031
		<b>VERSIÓN:</b> 1
		<b>FECHA:</b> 09/JUN/2022

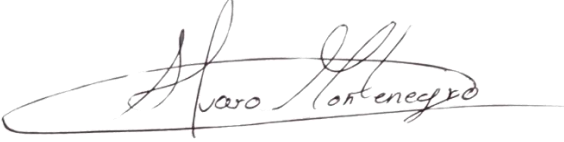

- b) Se autoriza a la Universidad CESMAG para publicar el Trabajo de Grado o de Aplicación en formato digital y teniendo en cuenta que uno de los medios de publicación del repositorio institucional es el internet, acepto(amos) que el Trabajo de Grado o de Aplicación circulará con un alcance mundial.
- c) Acepto (aceptamos) que la autorización que se otorga a través del presente documento se realiza a título gratuito, por lo tanto, renuncio(amos) a recibir emolumento alguno por la publicación, distribución, comunicación pública y/o cualquier otro uso que se haga en los términos de la presente autorización y de la licencia o programa a través del cual sea publicado el Trabajo de grado o de Aplicación.
- d) Manifiesto (manifestamos) que el Trabajo de Grado o de Aplicación es original realizado sin violar o usurpar derechos de autor de terceros y que ostento(amos) los derechos patrimoniales de autor sobre la misma. Por consiguiente, asumo(asumimos) toda la responsabilidad sobre su contenido ante la Universidad CESMAG y frente a terceros, manteniéndose indemne de cualquier reclamación que surja en virtud de la misma. En todo caso, la Universidad CESMAG se compromete a indicar siempre la autoría del escrito incluyendo nombre de(los) autor(es) y la fecha de publicación.
- e) Autorizo(autorizamos) a la Universidad CESMAG para incluir el Trabajo de Grado o de Aplicación en los índices y buscadores que se estimen necesarios para promover su difusión. Así mismo autorizo (autorizamos) a la Universidad CESMAG para que pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.

**NOTA:** En los eventos en los que el trabajo de grado o de aplicación haya sido trabajado con el apoyo o patrocinio de una agencia, organización o cualquier otra entidad diferente a la Universidad CESMAG. Como autor(es) garantizo(amos) que he(hemos) cumplido con los derechos y obligaciones asumidos con dicha entidad y como consecuencia de ello dejo(dejamos) constancia que la autorización que se concede a través del presente escrito no interfiere ni transgrede derechos de terceros.

Como consecuencia de lo anterior, autorizo(autorizamos) la publicación, difusión, consulta y uso del Trabajo de Grado o de Aplicación por parte de la Universidad CESMAG y sus usuarios así:

- Permiso(permitimos) que mi(nuestro) Trabajo de Grado o de Aplicación haga parte del catálogo de colección del repositorio digital de la Universidad CESMAG por lo tanto, su contenido será de acceso abierto donde podrá ser consultado, descargado y compartido con otras personas, siempre que se reconozca su autoría o reconocimiento con fines no comerciales.

En señal de conformidad, se suscribe este documento en San Juan de Pasto a los 26 días del mes de noviembre del año 2024

	
Firma del autor	Firma del autor
Nombre del autor: Alvaro Sebastian Montenegro Chamorro	Nombre del autor: Paulo Fabian Alexander Zambrano Urbina
Firma del autor	Firma del autor
Nombre del autor:	Nombre del autor:



UNIVERSIDAD  
**CESMAG**  
NIT: 800.109.387-7  
VIGILADA MINEDUCACIÓN

**AUTORIZACIÓN PARA PUBLICACIÓN DE TRABAJOS DE  
GRADO O TRABAJOS DE APLICACIÓN EN  
REPOSITORIO INSTITUCIONAL**

**CÓDIGO:** AAC-BL-FR-031

**VERSIÓN:** 1

**FECHA:** 09/JUN/2022

Nombre del asesor: Mario Fernando Henao Rosero