

# Prototipo de mouse virtual con visión artificial para interacción humano-computadora

Kevin Pascacio Rodríguez<sup>1,2</sup>, Francisco López Orozco<sup>1,2</sup>,  
Josué Domínguez Guerrero<sup>1,2</sup>

<sup>1</sup> Instituto de Ingeniería y Tecnología (IIT),  
Universidad Autónoma de Ciudad Juárez,  
México

<sup>2</sup> Laboratorio de Inteligencia Artificial y Analítica de Datos (LIAAD),  
Universidad Autónoma de Ciudad Juárez,  
México

{a1176714,francisco.orozco,josue.dominguez}@uacj.mx,  
www.uacj.mx/CiudadUniversitaria/LaboratorioCUIIT.html

**Resumen.** El uso prolongado de dispositivos de entrada tradicionales como el mouse físico o el *touchpad* puede generar incomodidad, lesiones musculoesqueléticas y costos asociados al mantenimiento y reemplazo. Diversas investigaciones han propuesto interfaces alternativas basadas en gestos y visión artificial para reducir estas problemáticas [1,5]. En este trabajo se presenta el diseño, implementación y evaluación de un prototipo de mouse virtual controlado por gestos de la mano, empleando técnicas de visión artificial en tiempo real. El sistema utiliza una cámara integrada o webcam convencional junto con bibliotecas de código abierto para detectar puntos de referencia de la mano y mapear sus movimientos al desplazamiento del cursor, así como ejecutar acciones de clic mediante gestos específicos. Se realizaron evaluaciones con usuarios potenciales y se siguió un proceso iterativo de refinamiento del prototipo basado en retroalimentación. Los resultados muestran que el prototipo es funcional, intuitivo y una vez que se haya desarrollado completamente, podría representar una alternativa viable al mouse tradicional, con beneficios en ergonomía, ahorro económico y experiencia de usuario.

**Palabras clave:** Visión artificial, interacción humano-computadora, mouse virtual, detección de gestos, seguimiento de mano.

## Prototype of a Virtual Mouse with Artificial Vision for Human-computer Interaction

**Abstract.** Prolonged use of traditional input devices such as physical mice or touchpads can lead to discomfort, musculoskeletal injuries, and associated maintenance and replacement costs. Several studies have proposed alternative interfaces based on gestures and computer vision

to mitigate these problems [1,5]. This paper presents the design, implementation, and evaluation of a virtual mouse prototype controlled by hand gestures, employing real-time computer vision techniques. The system uses an integrated camera or conventional webcam along with open-source libraries to detect hand landmarks and map their movements to cursor displacement, as well as execute click actions through specific gestures. Evaluations were conducted with potential users, and an iterative process of prototype refinement based on feedback was followed. The results show that the prototype is functional and intuitive, and once fully developed, it could represent a viable alternative to the traditional mouse, offering benefits in ergonomics, cost savings, and user experience.

**Keywords:** Computer vision, human-computer interaction, virtual mouse, gesture detection, hand tracking.

## 1. Introducción

La interacción humano-computadora (HCI) es un área de investigación que busca desarrollar métodos más naturales y eficientes para el control de sistemas computacionales. A pesar de la evolución tecnológica, el mouse físico continúa siendo el principal dispositivo de entrada en computadoras de escritorio y portátiles. Sin embargo, su uso prolongado se ha asociado con problemas de salud como fatiga muscular y síndrome del túnel carpiano [4,12].

En este contexto, se han propuesto diversas interfaces alternativas denominadas *free-hand*, las cuales eliminan el contacto físico directo con un dispositivo periférico [8]. Estas interfaces aprovechan tecnologías emergentes como el seguimiento ocular, señales biomédicas, reconocimiento facial y, de manera destacada, la visión artificial aplicada al reconocimiento de gestos de la mano [1,5,13].

La visión artificial permite a las computadoras interpretar imágenes y videos para ejecutar acciones específicas, emulando en cierta medida la percepción visual humana [17,18]. Aplicada a la HCI, esta tecnología posibilita el control del cursor y comandos básicos mediante gestos manuales capturados por una cámara convencional [11].

El objetivo de este trabajo es desarrollar y evaluar un prototipo de mouse virtual controlado por la mano del usuario, utilizando técnicas de visión artificial en tiempo real, con el fin de proporcionar una alternativa ergonómica, económica y accesible al mouse tradicional o *touchpad*.

## 2. Trabajos relacionados

Existen numerosos trabajos que abordan el control del cursor mediante métodos alternativos al mouse convencional. En [1] se presenta un mouse virtual basado en gestos de la mano capturados mediante una cámara, demostrando la viabilidad de este enfoque como alternativa de interacción. Si bien está muy relacionado al que se presenta en este artículo, los autores reconocen que se trata

de un desarrollo en un entorno totalmente controlado. Asimismo, Marín Cano [2] propone un sistema de detección de movimientos orientado a personas con movilidad reducida, empleando visión artificial en tres dimensiones. Este trabajo plantea e integra el diseño de un dispositivo 3D especializado con la interfaz de usuario. Otros estudios se han enfocado en la accesibilidad, desarrollando dispositivos controlados por movimientos de la cabeza o señales biomédicas [4,12,14]. Estas propuestas buscan la inclusión digital de personas con discapacidades motoras, aunque suelen requerir hardware especializado. Recientemente, se han incorporado técnicas de aprendizaje profundo para el reconocimiento de gestos en tiempo real, logrando mayor precisión sin necesidad de contacto físico, como se muestra en [5] y [13]. No obstante, algunos de estos sistemas presentan alta complejidad computacional o dependencia de condiciones controladas.

A diferencia de estos trabajos, la propuesta presentada en este artículo se enfoca en una solución ligera, basada únicamente en el uso de una cámara convencional y bibliotecas de software de libre acceso, orientada tanto a usuarios generales como a escenarios educativos o domésticos.

### **3. Metodología**

Para el desarrollo de este trabajo se empleó la metodología de prototipado, adecuada para sistemas interactivos donde la retroalimentación del usuario resulta fundamental [3]. Esta metodología permitió realizar mejoras progresivas al sistema conforme se detectaban limitaciones en su uso real. Las etapas consideradas fueron: (1) recolección y análisis de requisitos, (2) diseño rápido del sistema, (3) construcción del prototipo, (4) evaluación con usuarios, (5) refinamiento del prototipo y (6) desarrollo del producto de ingeniería final. Durante la fase inicial se analizó el funcionamiento de los dispositivos de entrada convencionales, así como sus desventajas ergonómicas y de mantenimiento [16]. Posteriormente, se definieron los gestos necesarios para el control del cursor y la ejecución de acciones básicas, inspirados en propuestas similares reportadas en la literatura [1,11].

### **4. Implementación del sistema**

El prototipo fue desarrollado en el lenguaje Python debido a su flexibilidad y amplio soporte para bibliotecas de visión artificial [15]. Para la captura y procesamiento de imágenes se utilizó la biblioteca OpenCV, ampliamente empleada en aplicaciones de visión por computadora [16].

#### **4.1. Detección de la mano**

Las coordenadas o puntos de referencia de la mano son fundamentales para el funcionamiento del prototipo, debido a que mediante estas se hace el control del mouse virtual, estas coordenadas se muestran al momento que la cámara

captura la mano del usuario y estos puntos de referencia son parte de la biblioteca *mediapipe*, *zhang2020mediapipehands*. Se toman algunos puntos específicos como el punto 8 (punta del dedo índice) y el punto 4 (punta del dedo pulgar). Un ejemplo de los puntos marcados se puede visualizar en la Fig. 1.

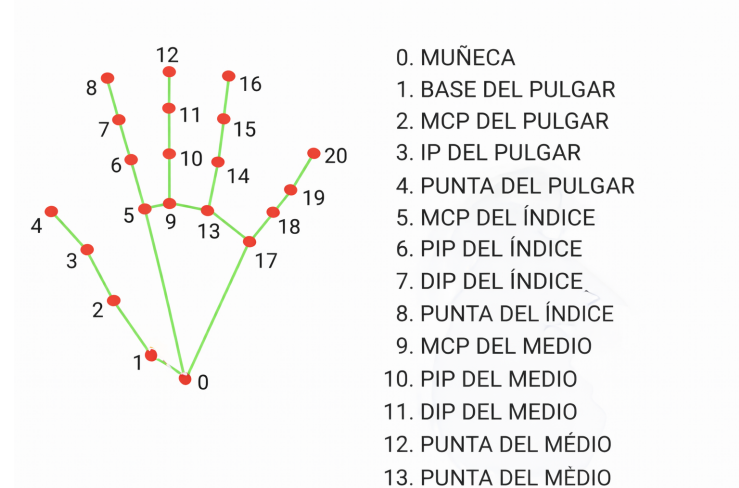


Fig. 1. Puntos de referencia en la detección de la mano en *mediapipe* [19].

La detección y seguimiento de la mano se realizó mediante la biblioteca *mediapipe*, la cual proporciona puntos de referencia (*landmarks*) de alta precisión para las articulaciones de la mano [13]. En particular, se utilizaron los puntos correspondientes a la punta del dedo índice y del pulgar, siguiendo enfoques similares a los reportados en [11]. En la Fig. 2 se presenta la detección de la mano de un usuario para realizar el movimiento del mouse virtual.

#### 4.2. Detección de dedos específicos de la mano

El control del cursor y la ejecución de clics se implementaron con la biblioteca *PyAutoGUI*, permitiendo mapear las coordenadas detectadas en la cámara a la resolución completa de la pantalla. El movimiento del cursor se activa cuando los dedos índice y pulgar se mantienen paralelos, mientras que el clic se ejecuta al unir ambos dedos como se muestra en la Fig. 3.

#### 4.3. Clics en la pantalla mediante el mouse virtual

Para que el mouse virtual realice un clic para abrir una carpeta, archivo, programa o similar, es necesario que la mano del usuario esté situada en la pantalla que se genera al iniciar la aplicación, además, el cursor debe estar en el



Fig. 2. Detección y puntos de referencia de la mano de un usuario.



Fig. 3. Detección de dedos específicos para realizar el movimiento del cursor.

lugar donde se hará el clic dependiendo lo que se desee abrir, seguido, se deben juntar las puntas del dedo índice y pulgar para que se realice el clic. Se muestra un ejemplo de selección (clic) un artículo en Amazon.com. Primero se mueve el cursor del mouse con la mano hacia la barra de búsqueda. Después se coloca el cursor sobre la barra de búsqueda, como se muestra en la Fig. 4; Al final, se juntan las puntas del dedo índice y del dedo pulgar para hacer el clic y la pagina nos direcciona automáticamente al producto elegido, como se muestra en la Fig. 5.

#### 4.4. Gestos para que la computadora no realice alguna acción

Para que el sistema no haga ningún movimiento del cursor en la pantalla de la computadora y tampoco realice algún clic, es necesario que la mano no se muestre en la pantalla inicial del sistema o simplemente se debe quedar abierta sin juntar los dedos índice y pulgar.

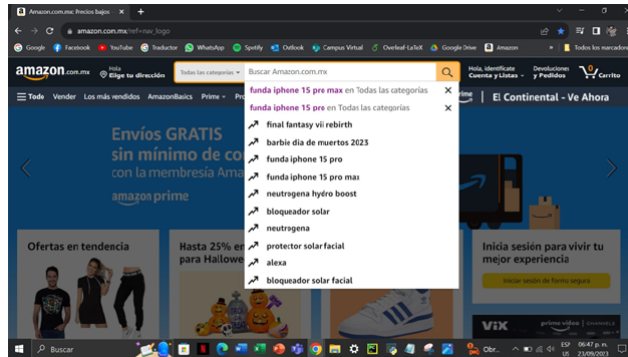


Fig. 4. Clic en la barra de búsqueda de Amazon para seleccionar artículo.

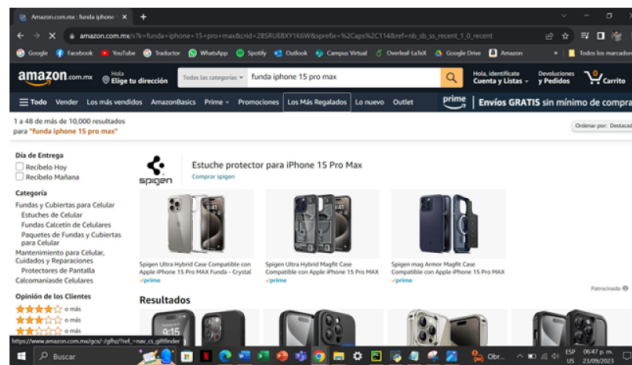


Fig. 5. Clic del usuario en un artículo seleccionado.

#### 4.5. Cierre de la aplicación

Para cerrar la ventana en la que la cámara web realiza la detección de la mano, solo es necesario pulsar la tecla “Esc” del teclado para cerrar dicha ventana y terminar la aplicación. En la Tabla 1 se describen los principales parámetros de operación de la aplicación propuesta.

### 5. Funcionamiento del mouse virtual

La Fig. 6 muestra un diagrama de flujo que describe cómo debe llevarse a cabo la secuencia de procesos para que el movimiento del cursor o clics se ejecuten, una vez detectada la mano del usuario. El Algoritmo 1, describe como el control del cursor fue implementado usando Python.

**Tabla 1.** Parámetros de configuración del sistema para el prototipo de mouse virtual.

Categoría	Parámetro	Valor / Descripción
Cámara	Modelo	Webcam integrada o  USB estándar
	Resolución	1280×720 píxeles (HD)
	FPS	30 cuadros por segundo
	Distancia usuario	40–60 cm entre mano y cámara
	Iluminación	Luz uniforme, sin sombras directas
Hardware	CPU	Intel Core i5 o superior
	RAM	8 GB mínimo
	GPU	Compatible con OpenCV
Sistema operativo	Plataforma	Windows 10 o superior
	Arquitectura	64 bits
Bibliotecas	OpenCV ( <code>cv2</code> )	Captura y procesamiento de imagen
	MediaPipe ( <code>mp.solutions.hands</code> )	Detección y rastreo de manos
	PyAutoGUI ( <code>pyautogui</code> )	Control del cursor y clics
Umbrales	Índice–pulgar clic	< 20 píxeles
	Índice–pulgar movimiento	< 100 píxeles
	Tolerancia de error	±5 píxeles
Configuración	Mapeo coordenadas	Escalado cámara → pantalla
	Modo espejo	Activado ( <code>cv2.flip(frame,1)</code> )
	Color marcadores	Amarillo (RGB: 0,255,255)

## 6. Evaluación experimental

La evaluación del prototipo se realizó en dos fases. En la primera participaron 58 usuarios, de los cuales 40 eran estudiantes de licenciatura y 18 eran empleados (administrativos, técnicos), 45 % eran mujeres, las edades comprendidas entre 21 y 38 años. Todos ellos reportaron tener un uso regular de computadoras (3 horas o más por día), además de tener conocimientos básicos de computadoras como el uso del mouse, teclado y aplicaciones de software de uso común.

Ninguno de ellos tenía alteraciones motoras que afectaran el movimiento de la mano, además su participación fue voluntaria y de por medio con un consentimiento informado. Se les pidió que realizaran tareas básicas como mover el cursor, abrir aplicaciones y navegar por páginas web. Estudios previos señalan que este tipo de pruebas de usabilidad son fundamentales para sistemas interactivos basados en gestos [3,7].

Tras analizar los resultados iniciales, se realizó un refinamiento del sistema orientado a mejorar la sensibilidad del cursor y la detección del clic. Como resumen de esta fase, se cuantificó el dominio que mostraron los usuarios al utilizar el mouse virtual. El 43.1 % de los participantes se adaptaron muy bien al prototipo, el 41.1 % de los usuarios, logró dominar bien el mouse virtual y el 15.5 % lo dominó medianamente bien. Todos estos resultados fueron obtenidos probando

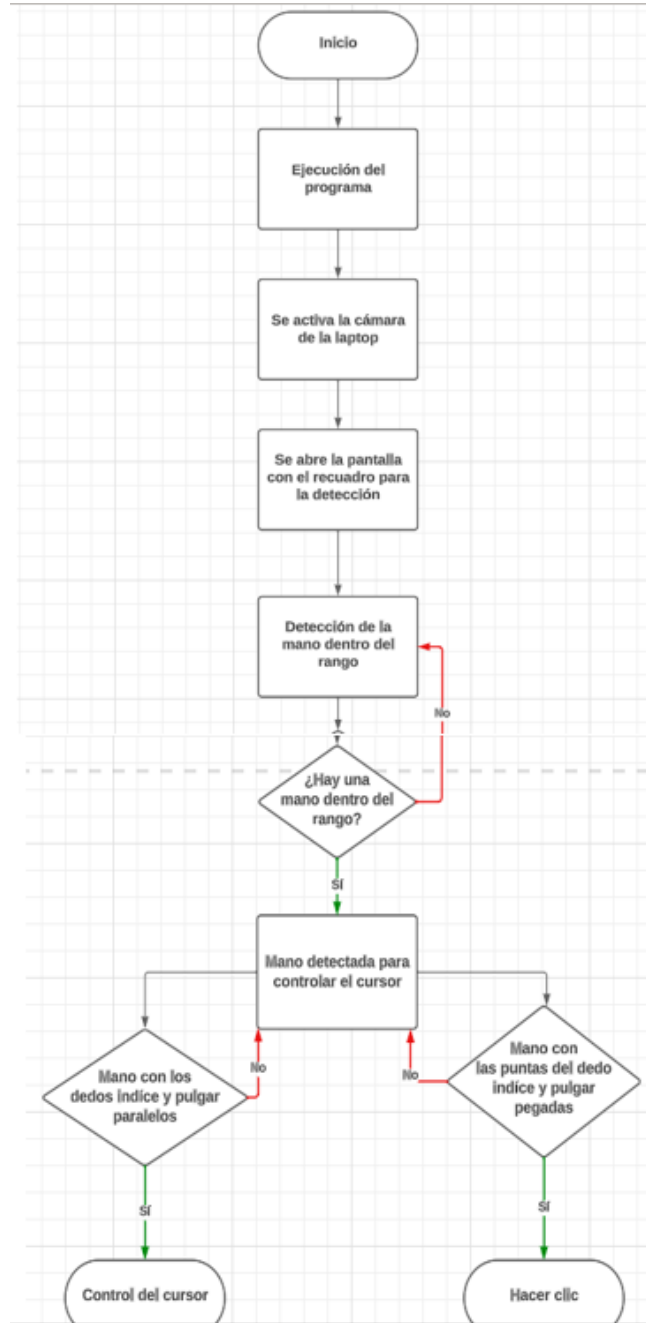


Fig. 6. Diagrama de procesos del funcionamiento del mouse virtual.

---

**Algorithm 1** Mouse virtual mediante visión artificial.

---

```
1: Inicializar cámara, detector de manos y tamaño de pantalla
2:  $index\_y \leftarrow 0$ 
3: while verdadero do
4:   Capturar y voltear frame
5:   Convertir frame a RGB y detectar manos
6:   if hay manos then
7:     for cada landmark do
8:       Calcular coordenadas  $(x, y)$ 
9:       if  $id = 8$  then
10:        Actualizar posición del cursor
11:       end if
12:       if  $id = 4$  then
13:        Calcular distancia índice-pulgar
14:        if distancia  $< 20$  then
15:          Clic
16:        else if distancia  $< 100$  then
17:          Mover cursor
18:        end if
19:       end if
20:     end for
21:   end if
22:   Mostrar frame
23:   if tecla = ESC then
24:     romper
25:   end if
26: end while
27: Liberar cámara y cerrar ventanas
```

---

en contextos no controlados (a veces con poca iluminación o la distancia a la que se posicionaba el usuario no era la adecuada).

Sin embargo se optó por hacer dichas pruebas en estos entornos no controlados a fin de probar en cierta forma la robustez de la implementación. Posteriormente, una segunda evaluación fue aplicada a 38 usuarios distintos, con el objetivo de validar las mejoras implementadas. En esta segunda evaluación, las edades comprendidas de los participantes osciló entre 23 y 40 años, además reportaron uso diario de computadoras y de dispositivos apuntadores. De igual forma se hizo uso de un consentimiento informado. En esta parte se aseguró que los participantes no tuvieran experiencia previa con prototipos de mouse virtuales para disminuir el posible sesgo en la retroalimentación.

También para el refinamiento de requisitos del prototipo, se les preguntó a los usuarios sobre lo que le gustaría que se implementara en un futuro. Las respuestas más comunes fueron: desplazamiento hacia abajo y hacia arriba, zoom en la pantalla, desplazamiento de diapositivas, selección de archivos múltiples, precisión del mouse en el momento de dar clic, y una movilidad más estable. Cabe destacar que la primera evaluación permitió destacar principalmente problemas

de precisión, latencia y clics fallidos. En el caso de la segunda, se presentaron mejoras significativas en tiempos de respuesta, precisión del cursor y satisfacción del usuario.

Estos resultados se presentan en las Tablas 2 y 3. La Tabla 2 muestra una comparativa de las métricas de desempeño, entre la primera (1a. Eval.) y la segunda evaluación (2a. Eval.). En primera instancia se incluye el tiempo promedio requerido por los usuarios para completar la tarea, el porcentaje o tasa de error, el número de clics fallidos, así como la precisión del cursor. Finalmente en el último renglón se presenta la latencia. De manera gráfica se incluyen estos resultados en el Fig. 7.

En la Tabla 3 se resumen las métricas tomadas en cuenta respecto a la usabilidad y experiencia de usuario, entre las dos evaluaciones (1a. Eval. , 2a. Eval). En este caso se reportan el nivel de respuesta, la comodidad declarada por el usuario, el nivel de satisfacción, la estimación de usabilidad y el porcentaje de falsos positivos así como el dominio del mouse y el tiempo de finalización de las tareas. De manera anticipada y a reserva de que en etapas posteriores se pretende llevar a cabo una nueva etapa, hasta el momento fue posible observar que las mejoras implementadas facilitaron el aprendizaje y uso del sistema, aunque factores externos como iluminación, calidad de cámara o distancia de la mano siguieron influyendo en los resultados. La Fig. 8 muestra estos resultados.

**Tabla 2.** Métricas de desempeño antes y después del refinamiento.

Métrica	1 <sup>a</sup> Eval. (58)	2 <sup>a</sup> Eval. (38)
Tiempo prom. (s)	32.5 ± 5.4	25.8 ± 4.7
Tasa de error (%)	12.4 ± 3.1	7.1 ± 2.6
Clics fallidos	4.2 ± 1.3	2.1 ± 0.9
Precisión cursor (%)	87.6 ± 4.2	93.8 ± 3.5
Latencia (ms)	95 ± 12	68 ± 9

**Tabla 3.** Usabilidad y experiencia de usuario.

Dimensión	1 <sup>a</sup> Eval.	2 <sup>a</sup> Eval.
Nivel de respuesta	Moderado	Alto
Comodidad (Likert 1–5)	3.2 ± 0.8	4.1 ± 0.7
SUS (0–100)	68 ± 6.5	78 ± 5.9
Satisfacción (%)	72 ± 8.1	85 ± 7.4
Falsos positivos (%)	9.5 ± 2.2	4.2 ± 1.6
Dominio mouse (%)	–	43.1 muy bien, 41.1 bien, 15.5 med.
Tiempo tareas	–	70.7% entre 11–40 s

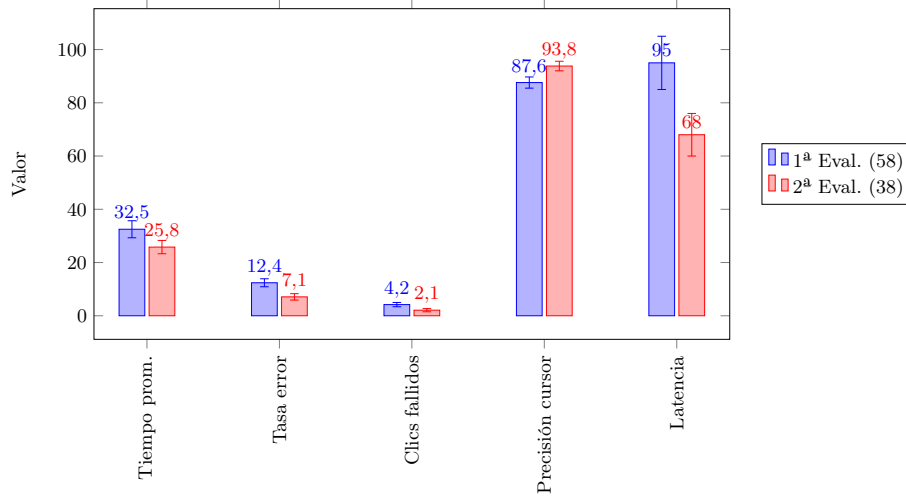


Fig. 7. Gráfico comparativo de las métricas de desempeño.

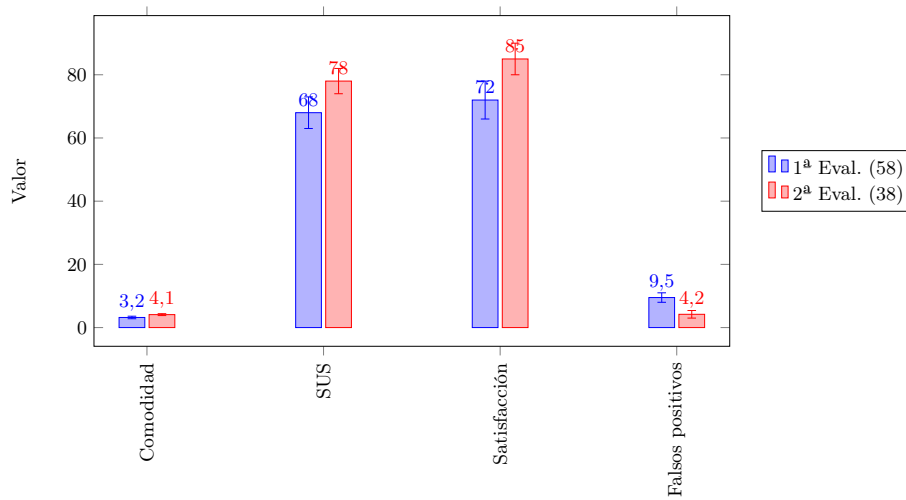


Fig. 8. Comparación de usabilidad y experiencia de usuario.

## 7. Resultados y discusión

Los resultados obtenidos indican que la mayoría de los usuarios logró adaptarse al sistema después de un breve periodo de uso. Más del 65% de los participantes completaron tareas comunes en menos de 26 segundos durante

la segunda evaluación, mostrando una mejora significativa respecto a la primera fase.

Estos resultados son consistentes con los reportados en trabajos previos sobre mouse virtual y control por gestos [1,5,13]. No obstante, se identificaron limitaciones relacionadas con la iluminación del entorno y la calidad de la cámara, factores que también han sido señalados en investigaciones similares [11].

## 8. Conclusiones y trabajo futuro

En este trabajo se presentó un prototipo de mouse virtual con visión artificial que permite controlar el cursor de la computadora mediante gestos de la mano. Los resultados experimentales muestran que el sistema es funcional y podría representar una alternativa al mouse tradicional, ofreciendo beneficios en ergonomía, ahorro económico y comodidad del usuario.

Esto siempre y cuando se mejoren las funciones actuales e incorporen algunas otras como el clic derecho, desplazamiento vertical, zoom y selección múltiple, además de mejorar la robustez del sistema ante variaciones de iluminación, en concordancia con las líneas de investigación señaladas en [5,13].

**Acknowledgments.** Se agradece el apoyo recibido por el Laboratorio de Inteligencia Artificial y Analítica de Datos (LIAAD) del Campus Ciudad Universitaria de la Universidad Autónoma de Ciudad Juárez (UACJ).

## Referencias

1. Basurto Segovia, G., Vences Gutiérrez, J. E., Catalán Salgado, E. A., Ortega Pacheco, J. D.: Mouse virtual usando gestos de la mano. Tesis de licenciatura, Instituto Politécnico Nacional (2017)
2. Marín Cano, A.: Diseño de un sistema detector de movimientos usando técnicas de visión artificial 3D para la interacción entre el computador y personas con movilidad reducida. Tesis doctoral (2017)
3. Carrillo Tomala, X.: Análisis y desarrollo de la aplicación móvil mouse y teclado remoto para compartir con estudiantes la pc del aula, para la carrera de ingeniería de sistemas de la universidad politécnica salesiana. B.S. thesis (2018)
4. Jaramillo Ruiz, J. E.: Mouse para personas con discapacidad en sus extremidades superiores controlado por medio de movimientos de cabeza y gestos. B.S. thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas (2016)
5. Shriram, S., Nagaraj, B., Jaya, J., Shankar, S., Ajay, P., et al.: Deep learning-based realtime ai virtual mouse system using computer vision to avoid covid-19 spread. *Journal of healthcare engineering*, vol. 2021 (2021)
6. Fu, Y., Huang, T. S.: hmouse: Head tracking driven virtual computer mouse. *IEEE Workshop on Applications of Computer Vision (WACV'07)*, pp. 30–30, IEEE (2007)
7. Zhou, Q., Fitzmaurice, G., Anderson, F.: In-depth mouse: Integrating desktop mouse into virtual reality. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–17 (2022)

8. Shibly, K. H., Kumar Dey, S., Islam, M. A., Iftekhar Showrav, S.: Design and Development of Hand Gesture Based Virtual Mouse. 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, pp. 1–5 (2019). doi: 10.1109/ICASERT.2019.8934612
9. Melozzi, F., Woodman, M. M., Jirsa, V. K., Bernard, C.: The virtual mouse brain: a computational neuroinformatics platform to study whole mouse brain dynamics. *Eneuro*, vol. 4, no. 3 (2017)
10. Vergara, S.: Prototipo lector de movimientos de la mano para la vinculación con la realidad virtual. Recuperado de <http://hdl.handle.net/11349/30222> (2021)
11. Praditasari, W. A., Aprilliyani, R., Kholis, I.: Design and implementation of interactive virtual museum based on hand tracking opencv in Indonesia. 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), pp. 253–256, IEEE (2021)
12. Díaz Suarez, J. E., Morales Vargas, C. R.: Desarrollo de una interfaz mediante señales EOG para el manejo de la computadora por parte de una persona con discapacidad en los miembros superiores (2019)
13. Martínez Hoces, E.: Hundsor: sistema para el control remoto de un ratón basado en el reconocimiento visual de los gestos de la mano mediante Deep Learning. E.T.S.I. de Sistemas Informáticos (UPM), Madrid (2023)
14. Arboleda Clavijo, C., García Cossio, E., Palacio Ochoa, M.: Control de mouse para computador mediante potenciales eléctricos oculares. *Revista Ingeniería Biomédica*, vol. 1, no. 1, pp. 47–51 (2007)
15. van Rossum, G., Drake, F.L.: Python Reference Manual. Centrum voor Wiskunde en Informatica, Amsterdam (1995)
16. Bradski, G.: The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000)
17. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. 4th edn. Pearson, New York (2018)
18. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer, London (2010)
19. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., Grundmann, M.: MediaPipe Hands: On-device Real-time Hand Tracking. CVPR Workshop on Computer Vision for Augmented and Virtual Reality (2020)