

13 de Junio de 2025

Inteligencia Artificial

# PROYECTO FINAL

Seguimiento Ocular

Angel Gabriel Torres Hernández

# PLANTEAMIENTO

Hoy en día las personas que padecen una discapacidad motriz parcial o completa enfrentan múltiples retos para interactuar con dispositivos electrónicos, lo que limita significativamente su autonomía y su capacidad de comunicación, así como el acceso a las plataformas digitales. Las interfaces tradicionales como lo son un teclado y ratón requieren un control motor que estas personas no pueden ejercer, y las excluye de herramientas importantes para su educación, trabajo o vida social.

# ARQUITECTURA

El sistema implementa un teclado virtual controlado por el movimiento de la mano, utilizando visión computacional y seguimiento de puntos clave (landmarks). Su objetivo es facilitar la escritura sin contacto físico, integrando autocompletado de palabras para mejorar la eficiencia del usuario.

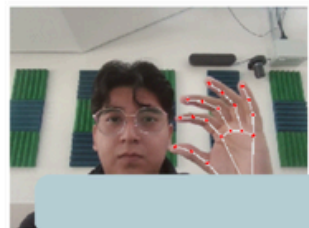
- ✓ La clase HandTracker usa OpenCV y MediaPipe para el seguimiento de la mano y visión computacional.
- ✓ Detecta la mano en video y extrae la posición de sus puntos clave (landmarks).
- ✓ Identifica un "clic" midiendo la distancia entre la punta del dedo índice y el pulgar.

# SOLUCIÓN PROPUESTA

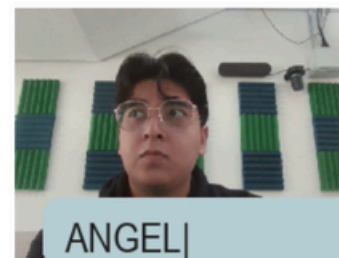
El prototipo de teclado por IA utiliza una interfaz visual mediante el seguimiento motriz usando técnicas de IA.

- OpenCV: Usado por HandTracker para captura y procesamiento de video (ej. `cv2.VideoCapture`).
- MediaPipe: Implementado en HandTracker (`mp.solutions.hands`) para el seguimiento de los 21 puntos clave de la mano.
- Predicción de Palabras (Mejora): Tu código usa una Red Neuronal LSTM (`NeuralWordPredictor`) en lugar de `difflib` para un autocompletado avanzado.
- Sonidos: MainApp utiliza `pygame.mixer.Sound` para los sonidos de clic.

# DEMOSTRACIÓN DE PROTOTIPO



¡Calibración completa! Puede empezar a escribir.



¡Calibración completa! Puede empezar a escribir.

ANGEL|



# VENTAJAS Y DESVENTAJAS

## Ventajas

- Interacción sin contacto físico
- Tecnología accesible
- Feedback auditivo y visual
- Personalizable
- Bajo costo de implementación

## Desventajas

- Precisión limitada
- Experiencia de usuario (UX) mejorable
- Funcionalidades básicas
- Accesibilidad
- Hardware dependiente

# CONCLUSIÓN

Se ofrece una interfaz vital para que personas con discapacidad motriz interactúen con dispositivos. El código demuestra la viabilidad de un teclado sin contacto, y la implementación del NeuralWordPredictor con una red LSTM es una mejora clave para una escritura más fluida y precisa.

# APRENDIZAJES ADQUIRIDOS

- **Integración de Visión Computacional e IA:** Aprendí a combinar seguimiento de mano y procesamiento de lenguaje natural para una interfaz interactiva.
  - Mi Código: La base está en HandTracker (visión) y NeuralWordPredictor (IA/PNL).
- **Interfaces Sin Contacto Físico:** Logré desarrollar un entorno funcional donde las teclas se seleccionan solo con el seguimiento del dedo índice.
  - **Mi Código:** Esto se ve en cómo HandTracker mapea la posición a la pantalla y la clase Button detecta el hover y la lógica de "pellizco" en MainApp genera el clic.



# APRENDIZAJES ADQUIRIDOS

- **Funcionalidades Inteligentes para Teclados:** Implementé autocompletado, borrado y sugerencias de texto.
  - **Mi Código:** La `NeuralWordPredictor` es clave para el autocompletado, y la lógica de `handle_typing` en `MainApp` maneja el borrado y el espacio.
- **Mejora de la Experiencia del Usuario (UX):** Comprendí la importancia de la retroalimentación visual, la tolerancia al error y la adaptabilidad.
  - **Mi Código:** Implementé retroalimentación visual (cambio de color en hover, cursor de mano), suavizado del cursor (`SMOOTHING_FACTOR`) para tolerancia al error, y una fase de calibración para la adaptabilidad.

**¡MUCHAS  
GRACIAS!**