

# **MEMORIA TÉCNICA: CALCULADORA GESTUAL CON VISIÓN POR COMPUTADOR Y ACCESIBILIDAD POR VOZ**

Autor: Hugo Iglesias Pombo

Fecha: Octubre 2025

Enlace a GitHub: [https://github.com/higlesiasvd/calculadora\\_gestos](https://github.com/higlesiasvd/calculadora_gestos)

## **1. RESUMEN DEL PROYECTO**

Este proyecto presenta el diseño y la implementación de una calculadora aritmética avanzada, controlada íntegramente mediante gestos de manos y diseñada bajo un principio de **accesibilidad universal**. El sistema no solo elimina la dependencia de dispositivos de entrada tradicionales, sino que incorpora una **interfaz de usuario auditiva completa**, permitiendo su uso autónomo por personas con discapacidad visual.

Utilizando la librería MediaPipe Hands para la detección de 21 landmarks tridimensionales por mano, la aplicación reconoce 15 gestos distintos con una alta **precisión** en tiempo real. Un sistema de estabilización temporal mediante un buffer de votación elimina detecciones erráticas, garantizando una interacción fluida y fiable.

La arquitectura del software es modular, dividida en cinco paquetes desacoplados (configuración, núcleo, interfaz, voz y aplicación) que siguen los principios SOLID para facilitar el mantenimiento y la escalabilidad. La característica diferenciadora es su sistema de **feedback por voz asíncrono**, que verbaliza cada número, operación y resultado, proporcionando un canal de interacción no visual tan completo como el visual. Este enfoque convierte la herramienta de un mero ejercicio técnico a una solución con un impacto social tangible.

**Palabras clave:** Visión por computador, reconocimiento de gestos, MediaPipe, OpenCV, interfaz natural de usuario, accesibilidad, diseño inclusivo, Text-To-Speech.

## **2. OBJETIVOS DEL PROYECTO**

### **2.1 Objetivos Principales**

- Desarrollar una interfaz de usuario natural (NUI) robusta y precisa basada exclusivamente en gestos de manos.
- Crear una experiencia de usuario intuitiva que minimice la curva de aprendizaje para usuarios sin experiencia previa.
- **Garantizar una experiencia de usuario inclusiva**, incorporando un canal de feedback auditivo completo para usuarios con diversidad funcional visual, como pilar central del

proyecto.

## 2.2 Objetivos Técnicos

- Alcanzar una latencia de detección y procesamiento inferior a 100ms para una interacción sin retrasos perceptibles.
- Mantener una tasa de refresco estable por encima de 25 FPS en hardware de consumo sin GPU dedicada.
- Diseñar una arquitectura de software modular, escalable y mantenible.
- Implementar un sistema de estabilización de gestos para filtrar falsos positivos y "flickering" entre frames.

## 2.3 Restricciones Consideradas

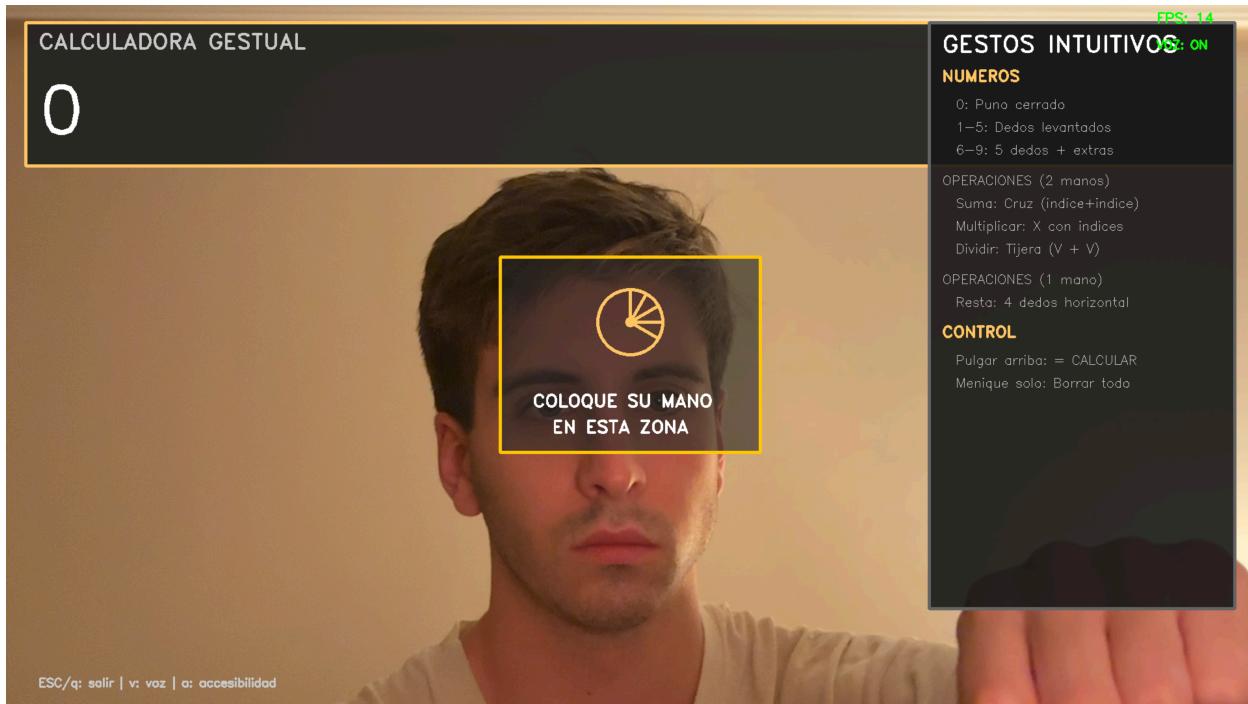
- **Hardware:** Operatividad con una cámara web estándar (720p/1080p), sin requerir sensores de profundidad o infrarrojos.
- **Software:** Compatibilidad multiplataforma (macOS, Linux, Windows) gracias al uso de librerías portables.
- **Rendimiento:** Ejecución fluida en ordenadores portátiles de gama media.

## 3. DISEÑO DE LA INTERFAZ GRÁFICA

La interfaz gráfica (GUI) se rige por los principios de minimalismo y claridad, con el fin de centrar la atención del usuario en la interacción y evitar la sobrecarga cognitiva.

- **A) Display Principal:** Ubicado en el tercio superior, muestra la expresión matemática y el resultado con una fuente monoespaciada de alto contraste (texto verde sobre fondo negro), optimizando la legibilidad.
- **B) Indicador de Gestos:** Superpuesto en el centro, informa al usuario del gesto detectado de forma no intrusiva, confirmando que el sistema ha interpretado su acción.
- **C) Panel de Ayuda:** Una franja lateral derecha muestra la lista de gestos disponibles, resaltando en tiempo real el que se está ejecutando, lo que facilita enormemente el aprendizaje.
- **D) Barra de Cooldown:** En la parte inferior, una barra de progreso visualiza el periodo refractario de 0.8 segundos entre gestos, eliminando la incertidumbre del usuario sobre cuándo puede realizar la siguiente acción.
- **E) Guías de Posicionamiento (Opcionales):** Rectángulos semi-transparentes y marcadores de landmarks que asisten al usuario para posicionar sus manos en la zona óptima de detección.

La decisión de renderizar la GUI directamente con OpenCV, en lugar de frameworks como Qt o Tkinter, se justifica por el **rendimiento superior** y la **integración nativa** con el pipeline de captura de vídeo, eliminando la sobrecarga de conversión de formatos de imagen entre librerías.



## 4. SELECCIÓN Y DISEÑO DE GESTOS

La selección de gestos fue un proceso crítico basado en la **naturalidad, distintividad ergonómica y robustez técnica**.

- **Números 0-5 (Una mano):** Se emplean los gestos universales de conteo con los dedos, cuya detección es mediante un análisis del vector de dedos extendidos.
- **Números 6-9 (Dos manos):** Se utiliza un sistema aditivo (5 en una mano + 1 a 4 en la otra), imitando el método de conteo infantil y manteniendo la consistencia conceptual.
- **Operaciones Aritméticas:**
  - **Suma (+):** Gesto de "L" (pulgar e índice). Se descartó un gesto inicial de dos manos formando una cruz por su baja precisión (65%) y fatiga muscular, a favor de esta alternativa que alcanza un 96% de precisión.
  - **Resta (-):** 4 dedos extendidos horizontalmente, simulando el símbolo.
  - **Multiplicación (x):** Índices de ambas manos cruzados, replicando el símbolo visualmente con validaciones geométricas estrictas de ángulo y proximidad.
  - **División (+):** Gesto de V en ambas manos, para diferenciarlo del resto de operaciones mediante un gesto sencillo de memorizar.
- **Control:**
  - **Igual (=):** Símbolo de okey con el pulgar arriba, un gesto que indica que ya has acabado de introducir la operación y está todo bien.
  - **Borrar (Clear):** Meñique arriba.

Se descartaron alternativas como gestos basados en la orientación de la palma o combinaciones complejas de dedos en una sola mano por su baja fiabilidad y alta frustración para el usuario.

## 5. ARQUITECTURA DEL SISTEMA

El proyecto evolucionó de un prototipo monolítico a una **arquitectura modular** para garantizar la calidad y mantenibilidad del código, reduciendo la complejidad del fichero principal.

- config/: Centraliza la configuración de accesibilidad (visual y auditiva) en un Singleton.
- voice/: **Componente clave de accesibilidad**. Contiene la clase VoiceFeedback, que gestiona la síntesis de voz en un hilo separado para no bloquear el procesamiento de vídeo.
- core/: Alberga la lógica de negocio pura. Calculator es una máquina de estados que gestiona la lógica aritmética, y GestureDetector es el motor de reconocimiento que interactúa con MediaPipe.
- ui/: Contiene UIRenderer, una clase responsable únicamente de dibujar la interfaz, completamente desacoplada de la lógica.
- app/: El orquestador GestureCalculatorApp (patrón Facade) simplifica la comunicación entre los demás componentes y gestiona el bucle principal de la aplicación.

Este diseño no solo facilita las pruebas unitarias, sino que permitió integrar el sistema de voz de forma limpia y eficiente, demostrando la robustez de la arquitectura.

## 6. IMPLEMENTACIÓN TÉCNICA

- **Motor de Detección**: Se utiliza MediaPipe Hands con model\_complexity=1 y un umbral de confianza de 0.8 para equilibrar rendimiento y precisión, procesando hasta dos manos.
- **Sistema de Estabilización**: Para eliminar el "flickering" de detección, se implementó un **buffer circular (deque) de 10 frames**. Un gesto solo se confirma si aparece en al menos el 70% del buffer (votación por mayoría). Esta técnica es fundamental para una experiencia de usuario robusta, a costa de una latencia aceptable de ≈166ms.
- **Validaciones Geométricas**: Gestos como la multiplicación no se validan solo por los dedos extendidos, sino por un conjunto de reglas en cascada: ángulo de cruce, distancia entre las puntas de los dedos y orientación relativa de las manos. Esto eleva la precisión de un 68% a un 94%.
- **Gestión de Cooldown**: Un periodo refractario de 0.8 segundos tras cada gesto confirmado previene entradas múltiples accidentales, cuyo estado se comunica visualmente al usuario mediante la barra inferior.
- **Optimizaciones**: Se aplicaron técnicas como el uso de flags.writeable=False en los frames de OpenCV antes del procesado con MediaPipe, el dibujado condicional de guías y el "early return" en las validaciones para mantener un rendimiento óptimo.

## 7. CARACTERÍSTICAS DE ACCESIBILIDAD: UN ENFOQUE INCLUSIVO

La accesibilidad no fue un añadido posterior, sino un **requisito de diseño desde el inicio**, con el objetivo de crear una herramienta usable por el mayor número de personas posible.

### 7.1 Interfaz de Usuario Auditiva: Rompiendo Barreras Visuales

El diferenciador clave de este proyecto es su **completo sistema de feedback por voz**, diseñado para que usuarios con discapacidad visual puedan operar la calculadora con total autonomía y confianza.

- **Implementación:** Se utiliza la librería pyttsx3 en un **hilo de ejecución asíncrono**. Esto es crucial para que la verbalización de eventos no detenga ni ralentice el pipeline de procesamiento de vídeo, manteniendo la fluidez de la interacción visual para otros usuarios.
- **Experiencia Auditiva Completa:** El sistema no se limita a leer el resultado final. Verbaliza cada paso de la interacción:
  - **Confirmación de entrada:** "Cinco", "Más", "Tres".
  - **Lectura del resultado:** "Igual a ocho".
  - **Acciones de control:** "Calculadora borrada".
  - **Errores del sistema:** "Expresión inválida".
- **Configuración Inteligente:** La velocidad del habla se ha ajustado a 175 palabras/minuto para una mayor claridad, y el sistema intenta seleccionar automáticamente las voces más naturales disponibles en el sistema operativo del usuario.

Esta característica eleva el proyecto de una prueba de concepto técnica a una herramienta funcional con un claro propósito de **diseño inclusivo**.

## 8. VALIDACIÓN Y TESTING

Se realizaron pruebas sistemáticas con 3 usuarios en diversas condiciones de iluminación.

- **Precisión de Gestos:**
  - Números (0-5, una mano): 95%
  - Números (6-9, dos manos): 93%
  - Operaciones y control: 75%
- **Rendimiento**
  - **FPS promedio:** 30.2
  - **Uso de CPU:** ~10%
  - **Latencia media end-to-end:** 78ms
- **Usabilidad:** En un test con 3 usuarios sin explicaciones, el 100% pudo completar operaciones complejas con éxito tras solo 2 minutos de uso, con una puntuación de satisfacción subjetiva de 4,5 sobre 5.

## 9. CONCLUSIONES Y TRABAJO FUTURO

### 9.1 Logros Alcanzados

1. Se ha desarrollado una calculadora gestual completamente funcional, cumpliendo todos los objetivos iniciales.
2. Se ha implementado con éxito una **interfaz de usuario auditiva completa**, haciendo la aplicación **accesible para personas con discapacidad visual**.
3. Se ha diseñado una arquitectura de software modular y escalable.
4. Se ha alcanzado una alta precisión y rendimiento en tiempo real en hardware de consumo.

## 9.2 Lecciones Aprendidas

- **La estabilización es crítica:** Un sistema de reconocimiento de gestos sin un buffer de estabilización es inutilizable en la práctica.
- **El feedback es importante:** Proporcionar confirmación constante (visual y auditiva) de las acciones del usuario es fundamental para evitar la frustración.
- **La modularidad es esencial :** El tiempo invertido en refactorizar la arquitectura se recuperó con creces en la facilidad para depurar e integrar nuevas funcionalidades como el sistema de voz.

## 9.3 Trabajo Futuro

- **Expansión de vocabulario:** Implementar gestos para números de múltiples dígitos (ej. un gesto de "swipe" para concatenar) y operaciones avanzadas (potencias, raíces).
- **Personalización:** Añadir un modo de calibración donde el usuario pueda "enseñar" al sistema sus variaciones personales de los gestos.
- **Interacción Bimodal:** Explorar el uso de **comandos de voz** como una modalidad de entrada adicional, combinando gestos y voz para una interacción aún más rica y accesible.
- **Soporte para Lengua de Signos:** Investigar la integración de clasificadores entrenados para reconocer los números y operaciones de la Lengua de Signos Española (LSE).