



**COMILLAS**  
UNIVERSIDAD PONTIFICIA  
**ICAI**

# Proyecto Final Piedra, Papel o Tijera

Computer Vision

José María Bengochea

Miguel Martín Vieira - 202203898  
Pablo Güell Borrajo - 202207546

11 de Enero de 2026

<b>1. INTRODUCCIÓN.....</b>	<b>3</b>
1.1. Alcance Del Proyecto.....	3
1.2. Objetivos Principales.....	3
1.3. Alcance Técnico.....	3
<b>2. METODOLOGÍA.....</b>	<b>4</b>
2.1. Calibración de la Cámara.....	4
2.2. Secuencia de Transformación de la Imagen.....	5
2.3. Sistema de Seguridad: Detección de Patrones y Extracción de Información.....	6
2.4. Sistema Propuesto: Tracker, Ampliaciones y Salida de Vídeo.....	6
<b>3. CONCLUSIONES.....</b>	<b>7</b>

# 1. INTRODUCCIÓN

## 1.1. Alcance Del Proyecto

El presente proyecto desarrolla un sistema completo de visión artificial para reconocimiento de gestos manuales en tiempo real, aplicado al juego clásico de Piedra, Papel o Tijera. El sistema implementa técnicas avanzadas de procesamiento de imágenes y visión por ordenador para crear una experiencia de juego interactiva y robusta.

## 1.2. Objetivos Principales

El proyecto abarca desde la captura y preprocesamiento de video, pasando por la calibración geométrica de la cámara, hasta la implementación de algoritmos de reconocimiento de patrones visuales y la lógica del juego. Se utiliza Python 3.9+ con OpenCV como librería principal para procesamiento de imágenes, junto con NumPy para operaciones matemáticas.

## 1.3. Alcance Técnico

- Implementar un sistema de calibración de cámara para corregir distorsiones ópticas
- Desarrollar algoritmos de detección y clasificación de gestos manuales mediante análisis de contornos y defectos convexos
- Crear un selector de modo de juego basado en reconocimiento de patrones de colores
- Proporcionar dos modos de juego: Jugador vs Jugador (PvP) y Jugador vs CPU (PvE)
- Diseñar una interfaz de usuario visual intuitiva con retroalimentación en tiempo real

## 2. METODOLOGÍA

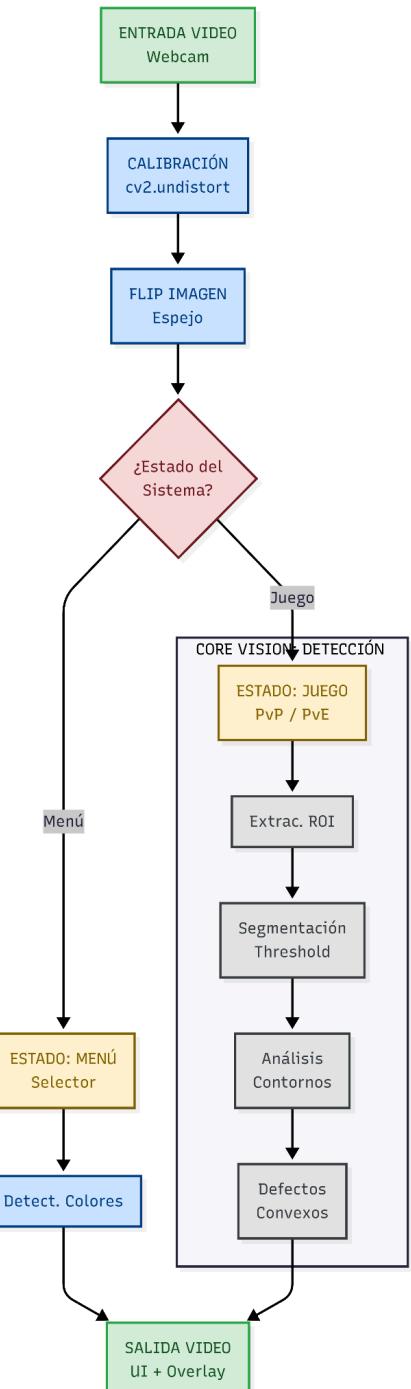
### 2.1. Calibración de la Cámara

La calibración de la cámara es fundamental para corregir las distorsiones introducidas por las lentes, especialmente la distorsión radial y tangencial. El proceso implementado consta de dos fases:

#### Fase 1: Captura de Imágenes de Calibración

Se utiliza un patrón de tablero de ajedrez (checkerboard) de dimensiones conocidas (9x6 esquinas internas). El script `capture\_calibration\_images.py` permite:

- Captura interactiva de 15-20 imágenes del patrón desde diferentes ángulos y distancias
- Validación en tiempo real de la detección correcta de esquinas
- Almacenamiento de imágenes válidas en el directorio "captured\_images/"



## Fase 2: Cálculo de Parámetros Intrínsecos

El script calibrate.py procesa las imágenes capturadas mediante el algoritmo de calibración de Zhang:

1. Detección de esquinas del tablero usando cv2.findChessboardCorners()
2. Refinamiento subpíxel con cv2.cornerSubPix() para mayor precisión
3. Cálculo de la matriz de cámara (K) y coeficientes de distorsión usando cv2.calibrateCamera()
4. Guardado de parámetros en formato NumPy comprimido (calibration\_data.npz)

### Resultado:

$$E_{RMS} = 0.300673$$

$$K = \begin{bmatrix} 518.536 & 0 & 332.085 \\ 0 & 517.849 & 241.800 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D = [-0.1050 \quad 0.8491 \quad -0.0001 \quad -0.0015 \quad -1.4271]$$

## 2.2. Secuencia de Transformación de la Imagen

El procesamiento de cada frame sigue una pipeline secuencial optimizada:

### 1. Captura y Corrección Geométrica

Frame RGB → Undistort (K, dist) → Crop ROI → Flip Horizontal

### 2. Conversión de Espacios de Color

- Para detección de colores: BGR → HSV (mejor separación cromática)
- Para detección de gestos: BGR → Grayscale

### 3. Filtrado y Reducción de Ruido

- Desenfoque Gaussiano: kernel 11×11 (menú), 35×35 (gestos)
- Propósito: Eliminar ruido de alta frecuencia

### 4. Segmentación - Método Otsu: Umbralización adaptativa binaria inversa

- Resultado: Máscara binaria donde la mano aparece en blanco

## 5. Operaciones Morfológicas

- **Opening:** Eliminar ruido blanco (píxeles aislados)
- **Closing:** Rellenar agujeros dentro de la región de interés
- Kernel:  $5 \times 5$  píxeles

## 6. Extracción de Características

- Detección de contornos con cv2.RETR\_TREE
- Cálculo de envolvente convexa (cv2.convexHull)
- Identificación de defectos convexos (cv2.convexityDefects)

## 2.3. Sistema de Seguridad: Detección de Patrones y Extracción de Información

Segmentación por Color en HSV:

- Rojo:  $[0-10, 120-255, 70-255] \cup [170-180, 120-255, 70-255]$
- Azul:  $[94-126, 80-255, 2-255]$
- Amarillo:  $[20-35, 100-255, 100-255]$

Validación de Circularidad:  $4\pi \times \text{Área} / \text{Perímetro}^2$  con un 60% de circularidad perfecta

Filtro de Área Mínima: 2000 píxeles (evita falsos positivos)

Requiere 15 frames consecutivos con el mismo color, previene detecciones erráticas por movimiento

**PvP:** Rojo → Amarillo → Azul

**PvE:** Azul → Amarillo → Rojo

## 2.4. Sistema Propuesto: Tracker, Ampliaciones y Salida de Vídeo

### Tracking en Tiempo Real:

- Detección continua en ambas ROIs (regions of interest) durante fase de preparación
- Captura instantánea al final de la cuenta regresiva (3-2-1-¡YA!)
- Frame congelado para evaluación final

### Ampliaciones

**1. Interfaz de Usuario Profesional:** diseño con paleta de colores moderna, rectángulos con esquinas redondeadas (radio 20-30px), fondos semitransparentes con alpha blending (0.7-0.9), texto con contornos para máxima legibilidad

**2. Retroalimentación Visual Avanzada:** círculo de progreso animado durante cuenta regresiva, efecto de pulso en números (escala dinámica 1.0-1.3), confeti aleatorio para celebración de victoria, contornos de mano en tiempo real con puntos de defecto

**3. Retroalimentación Auditiva:** beeps distintivos en cuenta regresiva (1000 Hz, 200ms), sonido de inicio "¡YA!" (2000 Hz, 400ms), tonos diferenciados para ganador 1 (500 Hz) vs ganador 2 (1500 Hz), sistema de threading para evitar congelamiento

### **3. CONCLUSIONES**

#### **3.1. Limitaciones Observadas:**

- Dependencia de condiciones de iluminación consistentes
- Requiere cámara de al menos 720p para detección confiable
- Puede haber latencia mínima (~100-200ms) en sistemas de bajo rendimiento

#### **3.2. Factores Críticos:**

- Iluminación: Mejor rendimiento con luz uniforme y sin sombras fuertes
- Fondo: Contraste adecuado entre mano y fondo mejora segmentación
- Distancia: Óptima a 40-70 cm de la cámara

#### **3.3. Tasa de Acierto**

- Piedra: ~95% (muy distintiva por ausencia de defectos)
- Papel: ~90% (clara separación de dedos)
- Tijera: ~85% (puede confundirse con Papel si dedos no están bien separados)

#### **3.4. Mejoras A Futuro**

- Machine Learning para clasificar objetos
- Tracking de la mano al completo
- Segmentación Semántica