



# **Fruit Ninja con Computer Vision**

Proyecto de imagen digital, Universidad de Extremadura



# contexto del proyecto

## Objetivo:

El objetivo de este proyecto es crear una versión interactiva del juego Fruit Ninja utilizando técnicas de computer vision.

El usuario puede interactuar con el juego en tiempo real, cortando fruta y esquivando bombas mediante movimientos de la mano detectados por la cámara.

Este proyecto demuestra el uso de bibliotecas de visión por ordenador y técnicas de detección en Python, aplicadas en un contexto de juego interactivo.

## Introducción:

- Python
- OpenCV
- MediaPipe
- Haar Cascade model



# Tecnologías utilizadas - OpenCv

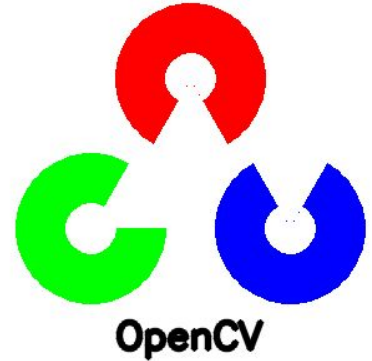
*Se utilizó en este proyecto:*

Captura de vídeo: **Capturar el flujo de vídeo de la webcam en tiempo real.**

Procesamiento de imagen: **Aplicar filtros y operaciones de transformación.**

Detección de caras: **Uso de plantillas Haar Cascade preentrenadas para colocar elementos gráficos sobre el usuario (como el mensaje «Game Over»).**

Superposición de imágenes y efectos gráficos: **Superponer imágenes como fondos semitransparentes e imágenes de frutas/bombas, y aplicar efectos de transparencia.**





## Tecnologías utilizadas - MediaPipe



MediaPipe es la biblioteca de código abierto de Google para computer vision, diseñada para aplicaciones en tiempo real como la detección de manos, caras, poses y objetos.

Tracking de manos: para permitir al usuario interactuar con el juego sin un mando.

Este seguimiento se utiliza para detectar los movimientos de la mano y permitir al usuario cortar fruta pasando la mano por

La gestión del seguimiento de las manos se separó de la lógica del juego y se desarrolló en la clase `hand_tracking.py`



# Tecnologías utilizadas - Haar cascade model

El modelo en cascada Haar es un método de detección de objetos que utiliza características Haar para identificar rápidamente caras, ojos y otros objetos en una imagen. Este modelo, entrenado en grandes conjuntos de datos, es conocido por su rapidez y eficacia.

Estos modelos se utilizan para la detección de caras y ojos, necesaria en determinadas pantallas, como la de derrota («Game Over»).

Los modelos Haar Cascade preentrenados permiten identificar rápidamente la cara y los ojos en el fotograma, lo que facilita la adición de elementos gráficos colocados sobre el usuario (como texto o imágenes).



# Conclusiones

## Posibles ampliaciones del proyecto

1. **Inteligencia Artificial para el Reconocimiento de Gestos:** Implementar un modelo de aprendizaje automático para detectar gestos específicos de la mano y ampliar el conjunto de acciones que el usuario puede hacer en el juego.
2. **Adición de dificultad creciente:** A medida que aumenta la puntuación, aumentar la velocidad de la fruta y las bombas para hacer el juego más desafiante.