



# Decodificando la Sonrisa con IA y Geometría Facial

**PhD. Pablo Eduardo Caicedo-Rodríguez**

Documento para  
Facultad de Ingeniería Biomédica

**19 Enero 2026**

**Universidad Escuela  
Colombiana de Ingeniería  
“Julio Garavito”**

📞 (03) 9903 4416  
✉️ decbiomedica@escuelaing.edu.co

## 1 Descripción del Taller

Este taller tutorial sumerge a los estudiantes en el campo de la Inteligencia Artificial (IA) perceptual. A diferencia de los enfoques clásicos de “caja negra”, los estudiantes utilizarán la moderna biblioteca MediaPipe de Google para generar una malla 3D de puntos clave (landmarks) faciales en tiempo real.

En lugar de depender de un clasificador pre-entrenado para “sonrisas”, los estudiantes diseñarán su *propio* detector. Aplicarán conceptos de geometría y biomecánica para calcular un “índice de sonrisa” basado en la deformación de los puntos clave de la boca, construyendo un sistema de IA más transparente, interpretable y robusto.

## 2 Objetivos de Aprendizaje

Al finalizar este taller, el estudiante será capaz de:

- Comprender la diferencia entre un enfoque de IA de “caja negra” (clasificación) y uno basado en características (análisis de puntos clave).
- Configurar un entorno de desarrollo en Python para pipelines de Machine Learning perceptual.
- Utilizar MediaPipe para la detección y seguimiento de puntos clave faciales (Face Mesh).
- Aplicar conceptos de geometría euclidiana (cálculo de distancias) para extraer características relevantes (señales) del rostro.
- Diseñar y calibrar un algoritmo simple (basado en umbrales) para detectar un evento biomecánico (la sonrisa).
- Reflexionar críticamente sobre la definición de “sonrisa” y cómo un algoritmo puede (o no) capturarla, abordando conceptos de IA confiable.

## 3 Audiencia Objetivo

- Estudiantes de educación secundaria (bachillerato).
- Edades: 15 a 17 años.
- **Prerrequisitos:** Interés en la tecnología y la programación. Se recomienda una comprensión básica de coordenadas cartesianas (plano x, y) y lógica de programación.

## 4 Duración Estimada

- **Duración Total:** 180 minutos (3 horas).

- Introducción (IA Perceptual, Geometría Facial): 25 min.
- Configuración del entorno (Python, OpenCV, MediaPipe): 25 min.
- Actividad de codificación (Paso 1: Malla Facial): 40 min.
- Actividad de codificación (Paso 2: Métrica de Sonrisa): 50 min.
- Pruebas, calibración de umbrales y experimentación: 20 min.
- Discusión y reflexión (IA Confiable, Sesgo, Aplicaciones): 20 min.

*Nota: La duración se extiende a 3 horas para acomodar la configuración de software y la fase de calibración de umbrales, que es pedagógicamente crucial.*

## 5 Materiales y Requerimientos

### 5.1 Materiales por estudiante (o por pareja):

- Un computador (portátil o de escritorio) con sistema operativo Windows, macOS o Linux.
- Cámara web (integrada o externa).
- Acceso a Internet (para descargar bibliotecas).

### 5.2 Software (a instalar):

- Python 3.8+: Entorno de programación.
- Bibliotecas de Python (a instalar vía pip):
  - opencv-python
  - mediapipe
  - numpy (generalmente instalado como dependencia)

## 6 Fases de la Actividad

### 6.1 Fase 1: ¿Cómo “Ve” la IA Moderna? (25 min)

- **Discusión guiada:** ¿Cómo funcionan los filtros de TikTok que ponen máscaras en tu cara? No solo detectan un rostro, detectan su *forma* 3D.
- **Conceptos Clave:**
  - **IA Perceptual:** IA que entiende el mundo a través de sensores (como la visión).
  - **Puntos Clave (Landmarks):** En lugar de una caja, la IA moderna encuentra puntos clave (ojos, nariz, boca) con coordenadas (x, y, z).
  - **Malla Facial (Face Mesh):** La red de puntos clave que define la geometría del rostro.
  - **Nuestra Tarea:** No le preguntaremos a la IA “¿está sonriendo?”, le preguntaremos “¿dónde están las comisuras de los labios?” y decidiremos *nosotros* si eso es una sonrisa.

## 6.2 Fase 2: Configuración del Entorno (25 min)

- Instalación de bibliotecas. Se recomienda verificar la instalación ejecutando un script simple que confirme el acceso a la cámara. bash      `pip install opencv-python mediapipe`

## 6.3 Fase 3: ¡A Programar! (90 min)

### Parte A: La Malla Facial (El “Wow”)

1. Importar bibliotecas (`cv2`, `mediapipe`).
2. Inicializar MediaPipe FaceMesh.
3. Iniciar la captura de video.
4. Bucle principal:
  - Leer `frame` y convertirlo (BGR a RGB) para MediaPipe.
  - Procesar el frame: `face_mesh.process(image)`.
  - Obtener resultados: Si se detectan rostros, iterar sobre `results.multi_face_landmarks`.
  - Dibujar la malla: Usar `mp_drawing.draw_landmarks` para dibujar la teselación facial completa.
  - ¡Momento de Pausa y Prueba! Los estudiantes deben ver una malla 3D en sus rostros. Este es el principal “gancho” visual.

### Parte B: La Métrica de la Sonrisa (La “Ciencia”)

1. Identificar Puntos Clave: Se proporcionará a los estudiantes un mapa de los índices de los puntos clave (Ej. 61: Comisura izquierda, 291: Comisura derecha, 13: Labio superior, 14: Labio inferior).
2. Calcular Distancias:
  - Usar numpy para calcular la distancia euclídea entre las comisuras (`dist_comisuras`).
  - Usar numpy para calcular la distancia entre los labios (`dist_labios`).
3. Definir la Métrica: La “sonrisa” ocurre cuando las comisuras se separan o cuando los labios se separan (mostrando dientes).
  - `indice_sonrisa_horizontal = dist_comisuras`
  - `indice_sonrisa_vertical = dist_labios`
4. Normalización (Opcional/Avanzado): Para que la detección sea independiente del tamaño del rostro en pantalla, normalizar las distancias (ej. `dist_comisuras / dist_ojos`).
5. Calibrar el Umbral:
  - Imprimir los valores del `indice_sonrisa` en la pantalla.
  - Los estudiantes deben sonreír y ver qué valor alcanzan.
  - Definir un umbral: `if indice_sonrisa_horizontal > UMBRAL_H:`
  - Mostrar “SONRISA” en la pantalla (`cv2.putText()`).

#### 6.4 Fase 4: Calibración y Reflexión (40 min)

- **Calibración (Experimentación):** Cada estudiante ajustará sus propios umbrales. ¿Por qué mi umbral es diferente al de mi compañero? (Diferente iluminación, diferente forma de rostro, diferente “nivel” de sonrisa).
- **Discusión (IA Confiable):**
  - ¿Es nuestro detector “justo”? ¿Qué pasa si alguien tiene una sonrisa más ancha o más estrecha naturalmente?
  - ¿Es robusto? ¿Qué pasa si te tapas la boca?
  - ¿Qué es mejor? ¿Nuestro detector “transparente” (que podemos depurar) o la “caja negra” de Haar (que no podemos)?
  - **Conexión Ética:** Discutir el *paper* “Gender Shades” y cómo los sistemas de IA (incluido el reconocimiento facial) pueden fallar en diferentes grupos demográficos.

### 7 Referencias de Soporte

1. Lugares, F., et al. (2019). MediaPipe: A Framework for Building Perceptible ML Pipelines. *arXiv preprint*. arXiv:1906.08172. (Referencia técnica del framework utilizado).
2. Howse, J., & D’Souza, P. (2020). *Learning OpenCV 4 Computer Vision with Python 3*. Packt Publishing. (ISBN: 978-1789531619). (Referencia práctica para la captura de video con OpenCV).
3. Buolamwini, J., & Gebru, T. (2018). Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. *Proceedings of Machine Learning Research (PMLR)*, 81: 77-91. (Referencia fundamental para la discusión ética sobre sesgos).