

**UNIVERSIDAD DEL VALLE DE GUATEMALA**

**CC3085 – INTELIGENCIA ARTIFICIAL**

**Sección 10**

**Ing. Alberto Suriano**



# **PROPUESTA PROYECTO FINAL**

**IMPLEMENTACIÓN DE RECONOCIMIENTO FACIAL**

**Michelle Angel de María Mejía Villela, 22596**

**Silvia Alejandra Illescas Fernández, 22376**

**Guatemala, 25 de abril del 2025**

# Sistema de Login con Reconocimiento Facial en Python

## 1. Descripción del problema

En un mundo donde la seguridad informática es cada vez más importante, el uso de contraseñas tradicionales representa un riesgo debido a vulnerabilidades como el phishing, la reutilización de contraseñas y el acceso no autorizado. Por esta razón, la autenticación biométrica, específicamente el reconocimiento facial, se plantea como una solución moderna, segura y práctica.

Este proyecto busca implementar un sistema de login mediante reconocimiento facial en Python, que pueda identificar si un usuario es válido o no con solo analizar su rostro en tiempo real usando una webcam. A su vez, el proyecto comparará dos enfoques distintos: uno basado en una librería de IA preentrenada (`face_recognition`), y otro mediante una implementación propia utilizando únicamente OpenCV y algoritmos clásicos de visión por computadora.

## 2. Análisis

El reconocimiento facial es una técnica de inteligencia artificial que permite identificar o verificar a una persona a partir de su rostro. El sistema será capaz de:

- Registrar el rostro de usuarios autorizados.
- Capturar la imagen de un rostro en tiempo real y compararla con las caras autorizadas.
- Permitir o denegar el acceso en función de la similitud facial detectada.

Para lograr esto, se utilizarán dos enfoques complementarios:

Con librería `face_recognition`:

- Utiliza redes neuronales convolucionales (CNN) previamente entrenadas para extraer codificaciones (embeddings) del rostro.
- Muy preciso, pero depende de una librería avanzada ya entrenada.

Implementación propia (sin librerías de IA):

- Utiliza Haar Cascades para detección facial.
- Emplea histogramas o gradientes (HOG) para extraer características manuales de la imagen del rostro.
- Compara rostros mediante distancia euclidiana entre vectores de características.

Este enfoque nos permitirá evaluar las ventajas y limitaciones de utilizar una librería de IA frente a una implementación tradicional.

### 3. Propuesta de solución

La solución consiste en una aplicación simple escrita en Python que permita:

1. Registrar usuarios (almacenando sus características faciales).
2. Realizar autenticación facial en tiempo real mediante una cámara web.
3. Decidir si otorgar o denegar acceso en función de la similitud con los usuarios registrados.

El sistema se dividirá en dos versiones:

- ✓ Versión 1: con face\_recognition, para aprovechar una solución basada en deep learning.
- ✓ Versión 2: implementación propia con OpenCV puro, sin modelos entrenados.

Adicionalmente, se considera incluir:

- Soporte para múltiples usuarios.
- Almacenamiento local de imágenes y codificaciones.
- Interfaz simple en consola o con PySimpleGUI.

### 4. Descripción de la solución

- Versión con face\_recognition:

**Registro:** Se captura una imagen del rostro y se extrae su codificación (128 valores flotantes).

**Login:** Se obtiene una nueva codificación del rostro detectado y se compara con las almacenadas usando compare\_faces().

**Acceso:** Se muestra "Acceso permitido" si la distancia es suficientemente baja.

- Versión propia sin librerías externas:

**Registro:** Se captura una imagen del rostro usando Haar Cascades.

**Extracción de características:** Se genera un vector de características (histograma o HOG básico).

**Comparación:** Se calcula la distancia euclidiana entre el rostro actual y los vectores registrados.

**Acceso:** Si la similitud es alta, se permite el acceso.

Ambas versiones compartirán la lógica general, permitiendo una comparación directa.

## 5. Herramientas aplicadas

Herramienta	Uso Principal
Python 3.11	Lenguaje principal del proyecto.
OpenCV	Captura de video, dibujo de rectángulos y texto, manipulación de imagen.
face_recognition	Librería basada en Dlib para detectar y codificar rostros.
NumPy	Cálculo de distancias entre codificaciones.
Pickle / JSON	Guardado de usuarios y sus codificaciones
Haar Cascades	Detección tradicional de rostros

Nota: face\_recognition está basada en el modelo FaceNet y usa internamente dlib con redes neuronales convolucionales para generar vectores de 128 dimensiones por rostro, mientras que la versión propia se basa en detección de patrones visuales sin redes neuronales.

## 6. Resultados esperados

Métrica	Valor Estimado fase_recognition	Valor Estimado Versión propia
Tiempo de detección	~0.5 segundos	~0.3–0.6 s
Precisión de reconocimiento	90–98% con buena luz	70–85%
Falsos positivos	Menos del 5%	Hasta 10%
Falsos negativos	Muy bajos con buena calidad de imagen	Mayor en condiciones adversas
Usuarios soportados	Escalable	Moderada

## 7. Conclusión

Este proyecto no solo explora el potencial del reconocimiento facial como sistema de autenticación, sino que también compara dos enfoques distintos: uno basado en inteligencia artificial preentrenada y otro construido desde cero con herramientas clásicas. Esto permite evidenciar las ventajas de la IA moderna, pero también la viabilidad de soluciones propias cuando se cuenta con recursos limitados o se busca mayor comprensión del funcionamiento interno.

A futuro, ambas versiones podrían integrarse en aplicaciones más robustas, móviles, web o sistemas físicos de control de acceso.

## 8. Anexos

