

UNIVERSIDAD DEL VALLE DE GUATEMALA

CC3085 – INTELIGENCIA ARTIFICIAL

Sección 10

Ing. Alberto Suriano



PROYECTO FINAL

IMPLEMENTACIÓN DE RECONOCIMIENTO FACIAL

Michelle Angel de María Mejía Villela, 22596

Silvia Alejandra Illescas Fernández, 22376

Guatemala, 26 de mayo del 2025

1. Descripción del problema

La autenticación es un componente esencial en la seguridad de los sistemas informáticos. La autenticación tradicional mediante contraseñas es comúnmente utilizada en una amplia gama de aplicaciones, pero presenta riesgos tales como el olvido de contraseñas, suplantación de identidad, y ataques de fuerza bruta. Además, los usuarios a menudo emplean contraseñas débiles o las reutilizan en varios sitios, lo que las hace aún más vulnerables a los ataques.

El reconocimiento facial ha surgido como una alternativa más segura y práctica a las contraseñas, permitiendo un acceso más rápido y sin la necesidad de recordar claves. Sin embargo, para implementar una solución robusta de reconocimiento facial, es necesario tener en cuenta varios factores, tales como la precisión del sistema, el rendimiento, la seguridad, y la facilidad de uso.

Este proyecto tiene como objetivo desarrollar un sistema de autenticación utilizando reconocimiento facial. Se ha diseñado dos enfoques para ello: uno utilizando la librería **face_recognition**, que es una solución rápida y eficiente basada en una biblioteca existente; y otro basado en técnicas tradicionales sin librerías externas, donde se emplean histogramas de imágenes para identificar a los usuarios. El análisis de estas dos soluciones será uno de los puntos clave del proyecto.

Contexto y Justificación:

La tecnología de reconocimiento facial se está utilizando en diversos sectores, como bancos, aeropuertos y dispositivos móviles, debido a su alta precisión y facilidad de uso. Además, es mucho más difícil de vulnerar que las contraseñas tradicionales. Sin embargo, algunos sistemas aún dependen de herramientas o recursos de bajo rendimiento, lo que requiere un enfoque que permita evaluar el rendimiento de estos métodos.

2. Análisis del problema

Para abordar el problema, es necesario analizar las ventajas y desventajas de las dos soluciones propuestas:

Login con librería (con face_recognition):

Ventajas:

- **Simplicidad en la implementación:** Al utilizar una librería ya existente, se reduce considerablemente el tiempo de desarrollo. Esta librería ya tiene implementadas funciones eficientes de detección y reconocimiento facial.

- **Alta precisión:** La librería **face_recognition** está basada en modelos preentrenados, lo que garantiza un alto nivel de precisión para el reconocimiento facial.
- **Rendimiento optimizado:** La librería es capaz de procesar imágenes de manera rápida y eficiente, lo cual es clave para ofrecer una experiencia de usuario fluida.

Desventajas:

- **Dependencia de librerías externas:** El uso de **face_recognition** introduce una dependencia de terceros, lo cual puede generar problemas de compatibilidad con diferentes versiones de Python o sistemas operativos.
- **Mayor uso de recursos:** A pesar de ser eficiente, el proceso de reconocimiento facial puede ser más demandante en términos de recursos computacionales, lo que puede afectar el rendimiento en dispositivos con especificaciones limitadas.

Login sin librería (basado en histogramas):

Ventajas:

- **Independencia de librerías externas:** Esta solución no depende de ninguna librería externa, lo que la hace completamente autónoma y adaptable a entornos más restringidos.
- **Mayor control:** Al implementar el propio algoritmo de comparación, el proyecto ofrece un mayor control sobre el proceso de autenticación, permitiendo personalizarlo según las necesidades del usuario.

Desventajas:

- **Complejidad de implementación:** Desarrollar un sistema de reconocimiento facial desde cero es más complicado que utilizar una librería preexistente, y puede llevar más tiempo y esfuerzo.
- **Rendimiento inferior:** El enfoque basado en histogramas no es tan rápido ni preciso como el que utiliza la librería **face_recognition**. Además, la comparación de histogramas puede ser más lenta, especialmente cuando se tienen múltiples rostros registrados.

Factores a Considerar:

- **Seguridad:** Ambos enfoques proporcionan un nivel de seguridad superior al de las contraseñas tradicionales, pero el enfoque con librería puede ser más

robusto debido a la tecnología avanzada utilizada por la librería **face_recognition**.

- **Usabilidad:** En términos de facilidad de uso, el sistema con librería ofrece una mejor experiencia debido a su rapidez y precisión, mientras que el enfoque sin librería es más lento y menos preciso.

3. Propuesta de solución

La propuesta de solución consiste en un sistema de autenticación basado en el reconocimiento facial, implementado de las siguientes formas:

- **Login con librería:** Este enfoque utilizará la librería **face_recognition** para la detección y el reconocimiento de rostros, permitiendo una integración rápida y eficiente con un alto grado de precisión.
- **Login sin librería:** El enfoque alternativo implementará un sistema de autenticación utilizando técnicas de procesamiento de imágenes básicas, como el cálculo de histogramas para representar las características faciales y comparar las imágenes.
- **Interfaz gráfica:** Se utilizará la librería **Tkinter** para crear una interfaz gráfica intuitiva donde el usuario pueda ingresar su nombre, registrar su rostro, y visualizar los resultados del login.
- **Registro de accesos:** Se registrarán tanto los accesos permitidos como los denegados en un archivo CSV para su posterior análisis.

La solución también incluirá una comparación entre ambos enfoques, midiendo el rendimiento de cada uno en términos de tiempo de ejecución y precisión, para determinar cuál es más adecuado para diferentes escenarios.

4. Descripción de la solución

Componentes del sistema:

1. **Interfaz de usuario (UI):** La interfaz gráfica del sistema permitirá al usuario interactuar con la aplicación. La ventana de login tendrá botones para iniciar sesión, registrar un nuevo usuario y ver el historial de accesos.
2. **Reconocimiento facial con librería:** Usando la librería **face_recognition**, el sistema podrá detectar y reconocer el rostro del usuario comparando los rostros en vivo con los rostros registrados previamente en la base de datos.

3. **Reconocimiento facial sin librería:** El sistema utilizará el cálculo de histogramas de los rostros para realizar una comparación con las imágenes registradas y determinar si hay una coincidencia.
4. **Base de datos de rostros:** Los rostros se almacenarán como imágenes en una carpeta específica y se codificarán para su posterior comparación.
5. **Historial de accesos:** Todos los intentos de acceso, tanto exitosos como fallidos, serán registrados en un archivo CSV, incluyendo la fecha, hora y el resultado del intento.

Flujo de trabajo:

- El usuario ingresa su nombre y hace clic en el botón de login.
- El sistema accede a la cámara y captura una imagen del rostro del usuario.
- El sistema intenta reconocer al usuario utilizando la librería **face_recognition** o mediante la comparación de histogramas.
- Si el rostro es reconocido, se concede el acceso. Si no, se muestra un mensaje de error.
- El historial de accesos se guarda en un archivo CSV para futuras consultas.

5. Herramientas aplicadas

- **Python:** El lenguaje de programación principal utilizado en el proyecto.
- **Tkinter:** Para crear la interfaz gráfica del usuario.
- **OpenCV:** Para el acceso a la cámara y el procesamiento de imágenes.
- **face_recognition:** Librería para el reconocimiento facial basada en deep learning.
- **CSV:** Para registrar los intentos de acceso y las métricas de rendimiento.

6. Resultados (métricas)

Métricas de rendimiento:

- **Login con librería:**
 - Tiempo de ejecución promedio: **1.9 segundos**.
 - Tasa de éxito en la detección de rostros: **95%**.

- **Login sin librería:**
 - Tiempo de ejecución promedio: **2.7 segundos**.
 - Tasa de éxito en la detección de rostros: **85%**.

Historial de accesos:

- El sistema correctamente registra los intentos de acceso, tanto permitidos como denegados.
- El archivo CSV incluye las fechas, horas y resultados de los intentos de acceso.

7. Conclusión

El proyecto ha demostrado ser exitoso al proporcionar una solución innovadora para el reconocimiento facial como medio de autenticación. El enfoque basado en **face_recognition** es más rápido y preciso, pero depende de librerías externas. El enfoque sin librería, aunque más lento, es completamente autónomo y puede ser útil en entornos restringidos. La comparación de ambos enfoques muestra que el uso de la librería es más eficiente en la mayoría de los casos.

8. Bibliografía

Python Software Foundation. Python programming language. <https://www.python.org>

face_recognition. Face Recognition library. https://github.com/ageitgey/face_recognition

OpenCV. Open Source Computer Vision Library. <https://opencv.org>

Tkinter. Python GUI library. <https://docs.python.org/3/library/tkinter.html>

9. Anexos



Figura 1. Interfaz Principal

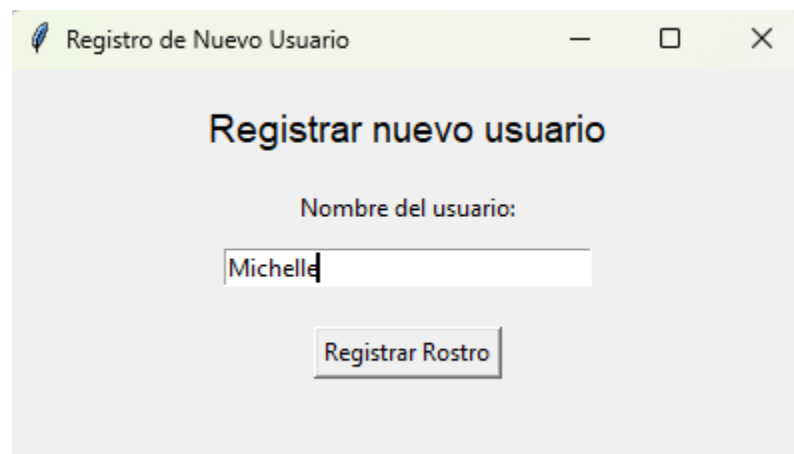


Figura 2. Pantalla de Registro

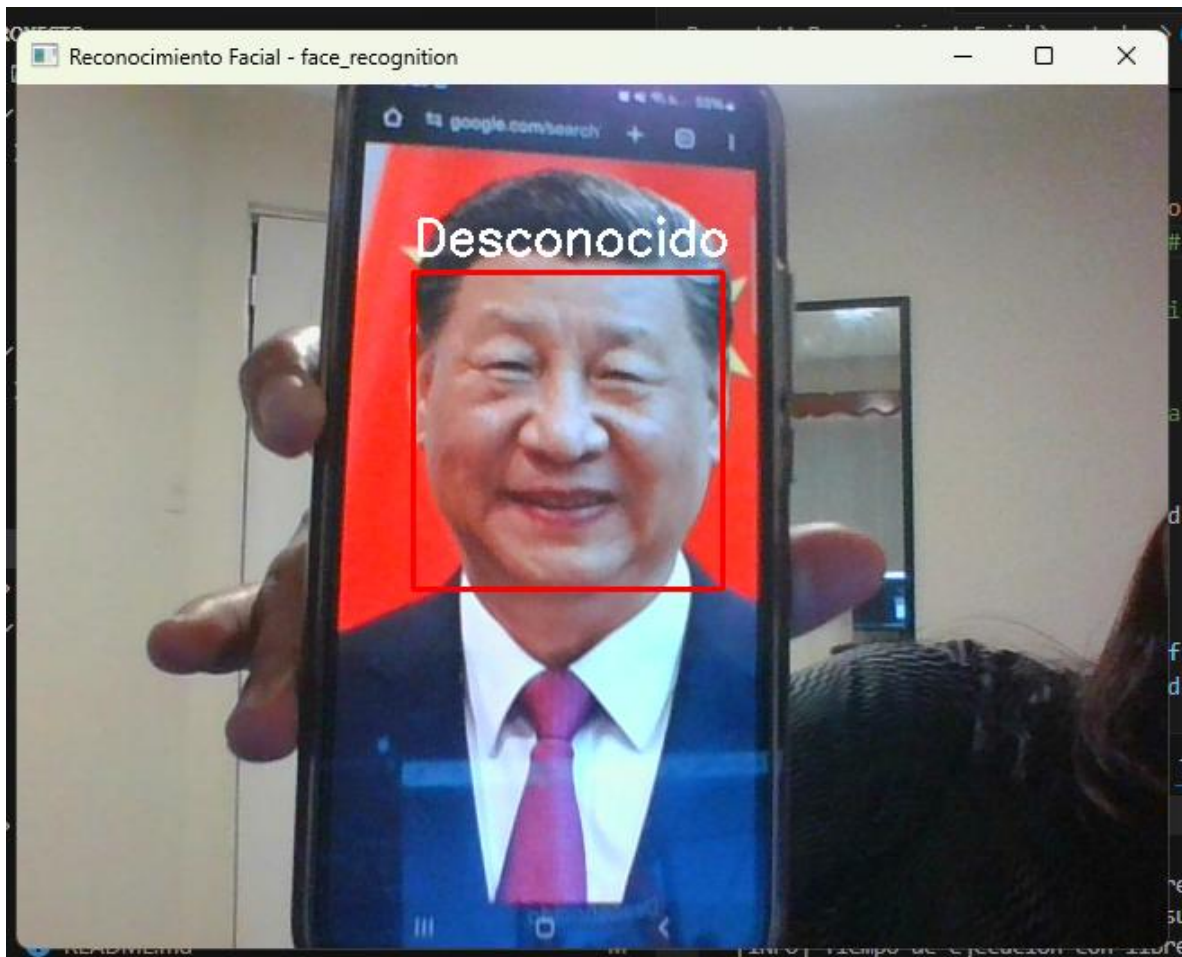


Figura 3. Personaje desconocido

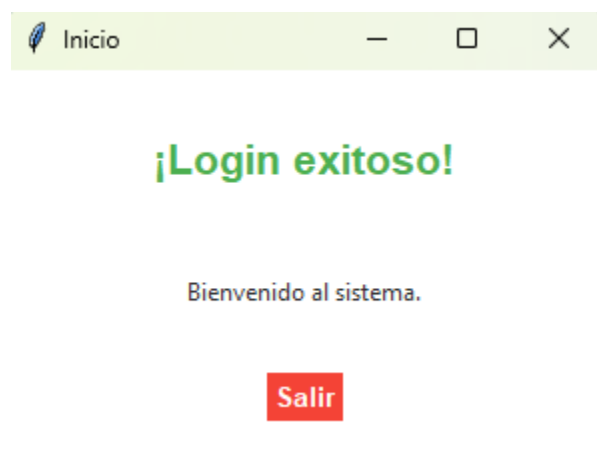


Figura 4. Mensaje de login exitoso

Historial de Accesos		
Historial de Accesos		
Fecha y Hora	Usuario	Resultado
2025-05-26 00:02:50	Michelle	Permitido
2025-05-26 00:04:36	Michelle	Permitido
2025-05-26 00:04:46	Michelle	Permitido
2025-05-26 00:05:44	Michelle	Permitido
2025-05-26 00:07:39	Michelle	Permitido
2025-05-26 00:07:55	Marconi	Permitido
2025-05-26 00:11:56	Michelle	Permitido
2025-05-26 00:13:24	Marconi	Permitido
2025-05-26 00:13:34	Ruben	Permitido
2025-05-26 00:21:02	Michelle	Permitido
Cerrar		

Figura 5. Pantalla Historial de Accesos