

Implementación de un sistema de control de acceso para gimnasios basado en reconocimiento facial orientado a objetos.

(Diciembre 2024)

Juan Pablo Avila Jaimes , Cesar David Porras Quintero , Yennier Danielo Peraza

INTRODUCCIÓN.

El reconocimiento facial se refiere al proceso de identificar o verificar la identidad de un individuo basándose en las características faciales. Aunque esta tarea es realizada de manera natural por los seres humanos, las máquinas también son capaces de llevarla a cabo, especialmente mediante el uso de la tecnología de reconocimiento facial [1]. A lo largo de las últimas tres décadas, el reconocimiento facial ha atraído un interés considerable, convirtiéndose en un área clave de investigación y desarrollo debido a su vasto potencial en diversas aplicaciones. Se considera una de las formas más accesibles y efectivas de análisis de imágenes y reconocimiento de patrones, lo que lo ha posicionado como un campo esencial en la inteligencia artificial y la visión por computadora [2].

Sin embargo, a pesar de los avances significativos en este campo, el reconocimiento facial continúa enfrentando una serie de desafíos técnicos y éticos que fomentan su constante evolución [3, 6]. Entre los retos técnicos más destacados se encuentran las variaciones en las condiciones de iluminación, las diferentes poses de los usuarios y los cambios en las características faciales debido a la edad, todos los cuales pueden afectar negativamente [10, 11, 12]. Los sistemas de control de acceso basados en biometría facial, en particular, han experimentado un crecimiento considerable en los últimos años. De hecho, más del 50% de las publicaciones en este campo han sido

realizadas en el último lustro, y se han empleado tecnologías innovadoras como redes neuronales [5], análisis de componentes principales y patrones binarios locales para mejorar su precisión y aplicabilidad.

Teniendo en cuenta que el reconocimiento facial es una de las muchas tecnologías con un gran potencial de aplicación, uno de sus usos más destacados es en el control de acceso y seguridad [4]. Este sistema permite identificar a las personas de manera rápida y eficiente, lo cual es esencial en entornos donde la seguridad es una prioridad, como en edificios gubernamentales, instituciones financieras y aeropuertos [9]. La automatización de este proceso no solo mejora la precisión, sino que también optimiza la eficiencia al reducir la necesidad de intervención manual [7, 8].

Para el desarrollo de sistemas robustos de reconocimiento facial, herramientas avanzadas como EfficientNet u otros [13] [14], una arquitectura de redes neuronales convolucionales optimizada [15] para lograr alta precisión con menor complejidad computacional, son esenciales. Estas herramientas permiten abordar desafíos técnicos como las variaciones en las condiciones de iluminación, las diferentes poses de los usuarios y los cambios en las características faciales debido a la edad.

Integrar estas tecnologías en plataformas basadas en programación orientada a objetos (POO) resulta especialmente eficiente.

Gracias a los principios fundamentales de la POO, como el encapsulamiento, la herencia y el polimorfismo, es posible estructurar el código en módulos reutilizables y escalables. Esto no solo facilita la expansión del sistema, sino que también mejora su mantenimiento y permite modelar componentes clave, como el registro y verificación de usuarios, como objetos interrelacionados.

En particular, la POO ofrece una base sólida para diseñar sistemas de control de acceso mediante reconocimiento facial, permitiendo una integración sencilla con otros sistemas y personalización según las necesidades específicas del usuario. Este enfoque modular y estructurado asegura que las plataformas puedan evolucionar de manera eficiente para enfrentar las demandas crecientes de aplicaciones como la seguridad y la gestión de accesos en tiempo real.

En este contexto, el proyecto se enfoca en desarrollar una plataforma orientada a objetos con Python, diseñada específicamente para facilitar la gestión de acceso en gimnasios de barrio. La idea principal es brindar a estos espacios una herramienta tecnológica que les permita optimizar su funcionamiento diario, mejorando la experiencia tanto para los administradores como para los usuarios.

El sistema estará centrado en dos funciones clave: el registro de usuarios y la confirmación de identidad mediante reconocimiento facial. Los usuarios podrán registrarse de forma rápida y segura, capturando sus características faciales, las cuales se almacenarán para futuras verificaciones. Una vez registrados, el sistema permitirá identificar a las personas al ingresar al gimnasio, eliminando la

necesidad de tarjetas físicas o listas manuales de asistencia. Esto agilizará el proceso de acceso, haciendo que sea más eficiente para todos los involucrados.

Además de estas funciones principales, se espera que el sistema incluya características que impulsen la productividad de los gimnasios. Por ejemplo, permitirá registrar los horarios de entrada y salida de los usuarios, ofreciendo datos valiosos para el análisis de la actividad diaria del gimnasio. Esto ayudará a los administradores a identificar patrones de uso, gestionar mejor los recursos disponibles y tomar decisiones informadas para mejorar el servicio.

El objetivo final es ofrecer un sistema que no solo sea preciso y seguro, sino que también sea fácil de usar e integrar en la rutina de los gimnasios de barrio. La plataforma busca ser una herramienta que les permita a estos espacios crecer y profesionalizarse, sin la necesidad de grandes inversiones en tecnología compleja. Con este sistema, se espera que los gimnasios puedan optimizar su operatividad, ofrecer una experiencia más cómoda a sus usuarios y mantenerse competitivos en un mercado en constante evolución.

METODOLOGÍA

La metodología seguida en este trabajo abarcó desde la definición de los requerimientos y la arquitectura del sistema, hasta la implementación, pruebas y ajustes finales. Cada fase se apoyó en el uso de Python como lenguaje de programación, el paradigma de Programación Orientada a Objetos (POO) como base conceptual, y el entorno de desarrollo Visual Studio Code para la edición y depuración del código. A continuación, se describen las etapas de forma detallada.

1. Análisis y Definición de Requerimientos:

Inicialmente, se llevó a cabo un análisis de las necesidades del gimnasio. A partir de entrevistas con encargados y personal administrativo, se establecieron los requerimientos fundamentales:

- Registro de usuarios: Incluir nombre, correo, contraseña encriptada y características faciales.
- Autenticación múltiple: Permitir el acceso mediante contraseña o reconocimiento facial.
- Control de accesos: Registrar las entradas (y potencialmente salidas) de cada usuario, almacenando la fecha y hora.
- Rol de administrador: Disponer de un usuario con credenciales predefinidas para realizar tareas como ver y eliminar usuarios, así como registrar nuevos perfiles.

2. Se definió también la necesidad de una interfaz gráfica sencilla y un esquema de base de datos relacional para mantener la coherencia de la información. Estos requerimientos guiaron el diseño de la arquitectura y los módulos del sistema.

3. Diseño de la Arquitectura del Sistema (POO):

Con los requerimientos claros, se diseñó una arquitectura basada en POO, definiendo clases y sus responsabilidades. Se crearon clases como:

- Clase Usuario: Gestiona la información de cada usuario, incluye métodos para autenticar vía contraseña y sirve de base para la verificación mediante reconocimiento facial.
- Clase Administrador: Especialización responsable de registrar, ver y eliminar usuarios. Esto encapsula la lógica de gestión administrativa, separándose de la lógica de usuario común.
- Clase Reconocimiento Facial: Encargada de la captura de imágenes desde cámara, extracción de vectores faciales y comparación de datos. Esta clase utiliza librerías externas como face_recognition y OpenCV, y se integra con el sistema a través de métodos claros y documentados.
- Clase BaseDeDatos: Abstracción de la conexión con MySQL, manejo de consultas SQL parametrizadas y comunicación con la capa de persistencia, asegurando que la lógica de negocio no se mezcle con la lógica de acceso a datos.
- Clase GUI (Interfaz Gráfica): Implementada con Tkinter, su responsabilidad es la interacción con el usuario final. Se diseñó utilizando principios de POO para separar la lógica de la presentación.

4. Este diseño modular, mediante clases y métodos bien definidos, facilita la expansión del sistema, el reemplazo de componentes (por ejemplo, cambiar el motor de base de datos o la librería de reconocimiento facial) y la mantenibilidad a largo plazo.

5. Implementación en Python y Visual Studio Code:

La implementación se realizó íntegramente en Python, aprovechando su sintaxis clara y su amplia gama de librerías para reconocimiento facial y manejo de interfaz gráfica. El entorno de desarrollo elegido fue Visual Studio Code, ya que proporciona:

- Destacado de sintaxis y sugerencias inteligentes: Facilitan la escritura de código más rápido y con menos errores.
 - Integración con Python y entornos virtuales: Permite la instalación de paquetes y la gestión del entorno virtual (ej. venv) donde se aislaron las dependencias, asegurando la reproducibilidad del proyecto.
 - Depuración paso a paso: Herramientas integradas para colocar breakpoints, inspeccionar variables y flujo de ejecución, siendo crucial para resolver problemas lógicos en las primeras versiones.
6. Durante la codificación se siguieron buenas prácticas de POO, aplicando principios sólidos en la medida de lo posible. Por ejemplo, se buscó el principio de responsabilidad única,

de modo que cada clase tuviera una función clara y específica, evitando acoplamientos excesivos entre componentes.

7. Gestión de la Base de Datos MySQL:

Se utilizó MySQL como sistema gestor de base de datos, creándose un esquema con las tablas de usuarios.. Se definieron claves foráneas para mantener la integridad referencial, y se utilizaron consultas parametrizadas para prevenir inyecciones SQL. La clase BaseDeDatos encapsuló estas operaciones, ofreciendo métodos generales (ejecutar consulta, obtener_resultados) que el resto del sistema invocó sin necesidad de conocer detalles internos.

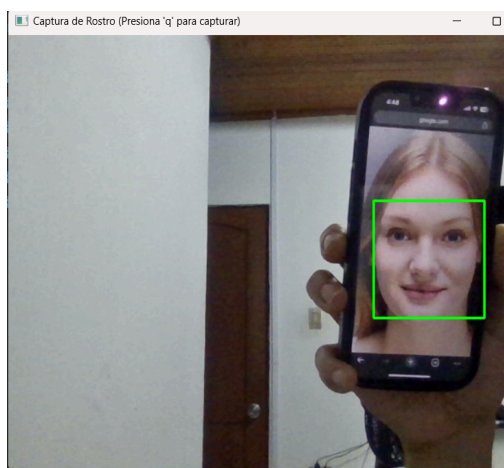
8. Pruebas Unitarias e Integración Continua:

Antes de proceder a las pruebas funcionales, se llevaron a cabo pruebas unitarias sobre las clases Usuario, Administrador y BaseDeDatos empleando datos simulados. Estas pruebas se configuraron desde Visual Studio Code, ejecutando scripts de test en el terminal integrado.

Posteriormente, se realizaron pruebas de integración, verificando el flujo completo: registro de un usuario, almacenamiento en la base de datos, captura de rostro, reconocimiento al momento del acceso y registro en registros_ingresos. Estas pruebas también incluyeron la interacción con la interfaz gráfica, asegurando que los botones, diálogos y ventanas funcionasen según lo esperado.

RESULTADOS

- El sistema detectó eficazmente los rostros presentes en la imagen de la cámara, marcándose con un recuadro verde. Esta funcionalidad, implementada gracias a la integración de librerías de visión por computador, ofreció al usuario una referencia visual inmediata sobre el momento adecuado para capturar el rostro.



2. Reconocimiento facial preciso y eficiente:

```

5 class ReconocimientoFacial:
6     def __init__(self):
7         pass
8
9     def convertir_a_bytes(datos):
10         return datos.tobytes() if datos is not None else None
11
12     @staticmethod
13     def convertir_a_ndarray(datos_bytes):
14         return np.frombuffer(datos_bytes, dtype=np.float64) if datos_bytes else None
15
16     @staticmethod
17     def comparar_rostros(datos1, datos2):
18         if datos1 is None or datos2 is None:
19             return False
20         return face_recognition.compare_faces([datos1], datos2)[0]
21
22 if __name__ == '__main__':
23     pass

```

3. Integración con la Base de Datos y Control de Accesos:

La consulta y registro de información en la base de datos MySQL se gestionaron de forma encapsulada a través de clases específicas. Este diseño orientado a objetos permite mantener la separación de responsabilidades y facilitó la ejecución de consultas parametrizadas, garantizando integridad y seguridad.

```

baseDeDatos.py X
baseDeDatos.py > BaseDeDatos > __init__
1 import mysql.connector
2
3 class BaseDeDatos:
4     def __init__(self, host, user, password, database):
5
6         self.connection = mysql.connector.connect(
7             host='localhost',
8             user='root',
9             password='2024',
10            database='gymapp'
11        )
12        self.cursor = self.connection.cursor()
13
14
15
16     def ejecutar_consulta(self, consulta, datos=None):
17         self.cursor.execute(consulta, datos or ())
18         self.connection.commit()
19         return self.cursor
20
21     def obtener_resultados(self, consulta, datos=None):
22         self.cursor.execute(consulta, datos or ())
23         return self.cursor.fetchall()
24
25     def cerrar_conexion(self):
26         self.cursor.close()
27         self.connection.close()
28

```

Figura 3: Código de conexión con base de datos MySQL.

4. Modularidad y Mantenibilidad del Código:

Gracias a la Programación Orientada a Objetos, se logró una separación lógica entre componentes: la interfaz gráfica, el reconocimiento facial y la interacción con la base de datos se representan como objetos independientes. Esto no solo facilitó las pruebas unitarias, sino que también permite futuras mejoras o sustituciones de componentes (por ejemplo, cambiar la biblioteca de reconocimiento facial sin alterar el resto del sistema).

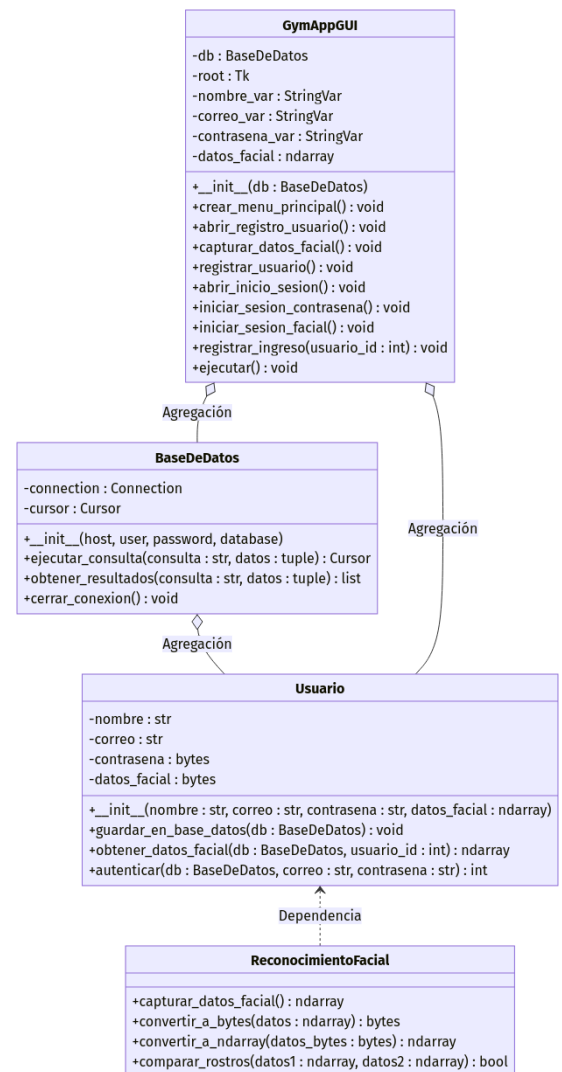


Figura 4: Diagrama de clases.

CONCLUSIONES

El proyecto desarrollado demuestra que es posible implementar un sistema de control de acceso basado en reconocimiento facial de manera eficiente y funcional, adaptado a las necesidades específicas de gimnasios de barrio. A través del uso de la Programación Orientada a Objetos (POO) y herramientas tecnológicas avanzadas, se logró construir una solución modular, escalable y fácilmente mantenible.

Durante el desarrollo, la integración de librerías como OpenCV y face_recognition

permitió implementar funcionalidades de detección y reconocimiento facial con alta precisión y rapidez, cumpliendo con los requerimientos planteados. El uso de una base de datos relacional, gestionada a través de MySQL, garantizó la consistencia y seguridad de la información almacenada, mientras que el diseño modular de clases facilitó la separación de responsabilidades y la claridad en la implementación del código.

Las pruebas realizadas confirmaron que el sistema es capaz de operar en condiciones reales, aunque se identificaron ciertos factores, como la iluminación, que pueden influir en su rendimiento. Sin embargo, se realizaron los ajustes necesarios para optimizar el desempeño en estos escenarios.

En conclusión, este proyecto constituye una solución tecnológica accesible y funcional para gimnasios de barrio, con potencial para ser extendido a otros contextos o mejorar su alcance mediante la integración de nuevas funcionalidades. La implementación reafirma la viabilidad de combinar tecnologías modernas y paradigmas de programación robustos para resolver problemas específicos de manera eficiente y escalable.

REFERENCIAS

[1] Simmler, M., & Canova, G. (2025). Facial recognition technology in law enforcement: Regulating data analysis of another kind. *Computer Law & Security Review*, 56, 106092. <https://doi.org/10.1016/j.clsr.2024.106092>

[2] Adjabi, I., Ouahabi, A., Benzaoui, A., & Taleb Ahmed, A. (2020). Pasado, presente y futuro del reconocimiento facial: una revisión. *Electrónica*, 9(8), 1188. <https://doi.org/10.3390/electronics9081188>

[3] Smith, M., Miller, S. The ethical application of biometric facial recognition technology. *AI & Soc* 37, 167–175 (2022). <https://doi.org/10.1007/s00146-021-01199-9>

[4] Hyesun Choung, David, P., & Ling, T.-W. (2024). Acceptance of AI-powered facial recognition technology in surveillance scenarios: Role of trust, security, and privacy perceptions. *Technology in Society*, 79, 102721. <https://doi.org/10.1016/j.techsoc.2024.102721>

[5] Barman, A., & Dutta, P. (2024). Facial expression recognition using Reversible Neural Network. *Applied Soft Computing*, 162, 111815. <https://doi.org/10.1016/j.asoc.2024.111815>

[6] Simmler, M., y Canova, G. (2025). Tecnología de reconocimiento facial en la aplicación de la ley: regulación de otro tipo de análisis de datos. *Computer Law and Security Report*, 56 (106092), 106092. <https://doi.org/10.1016/j.clsr.2024.106092>

[7] Aleluya, ERM y Vicente, CT (2018). Faceture ID: autenticación multifactorial mediante gestos faciales y manuales mediante aprendizaje profundo. *Procedia Computer Science*, 135, 147–154. <https://doi.org/10.1016/j.procs.2018.08.160>

[8] Wu, X., Xu, J., Wang, J., Li, Y., Li, W. y Guo, Y. (2019). Autenticación de identidad en dispositivos móviles mediante verificación facial y reconocimiento de imágenes de identificación. *Procedia Computer Science*, 162, 932–939. <https://doi.org/10.1016/j.procs.2019.12.070>

- [9] Johnson, T. L., Johnson, N. N., Topalli, V., McCurdy, D., & Wallace, A. (2024). Police facial recognition applications and violent crime control in U.S. cities. *Cities*, 155, 105472. <https://doi.org/10.1016/j.cities.2024.105472>
- [10] Cao, Q., Shen, L., Xie, W., Parkhi, OM, & Zisserman, A. (2018). VGGFace2: un conjunto de datos para reconocer rostros según pose y edad. *Decimotercera Conferencia Internacional IEEE de 2018 sobre reconocimiento automático de rostros y gestos (FG 2018)*, 67–74. <https://doi.org/10.1109/FG.2018.00020>
- [11] Fang, X., Fu, X., & Xu, X. (2017). ID card identification system based on image recognition. *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)* <https://doi.org/10.1109/ICIEA.2017.828307>
- [12] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10), 1499–1503. <https://doi.org/10.1109/LSP.2016.2603342>
- [13] Ipinze Tutuianu, G., Liu, Y., Alamäki, A., & Kauttonen, J. (2024). Benchmarking deep Facial Expression Recognition: An extensive protocol with balanced dataset in the wild. *Engineering Applications of Artificial Intelligence*, 136(Part B), 108983. <https://doi.org/10.1016/j.engappai.2024.108983>
- [14] Li, J., Ding, Y., Shao, Z., & Jiang, W. (2024). Face recognition method based on fusion of improved MobileFaceNet and adaptive Gamma algorithm. *Journal of the Franklin Institute*, 361(17), 107306. <https://doi.org/10.1016/j.jfranklin.2024.107306>
- [15] Elhani, D., Megherbi, A. C., Zitouni, A., Dornaika, F., Sbaa, S., & Taleb-Ahmed, A. (2023). Optimizing convolutional neural networks architecture using a modified particle swarm optimization for image classification. *Expert Systems with Applications*, 229(Part A), 120411. <https://doi.org/10.1016/j.eswa.2023.120411>