# SISTEMA WEB PARA GESTIÓN Y AUTOMATIZACIÓN DE GIMNASIO CON TECNOLOGÍA BIOMÉTRICA

Gael Alejandro Lozano Rodríguez, gaelalejandro912@gmail.com ✉(1), Pablo Jair Tenorio Escalón, pjtesc31@gmail.com (2)

**INSTITUCIÓN**
Facultad de Ingeniería Mecánica y Eléctrica, Estudiante.

**RESUMEN**
Este proyecto presenta el desarrollo de un sistema integral para gimnasios enfocado en la automatización de procesos esenciales como el control de accesos, la venta de productos y la administración de membresías. La plataforma combina tecnologías web y biométricas, utilizando Laravel, Python, React, Inertia y MySQL para ofrecer una solución moderna y accesible. Uno de los componentes clave del sistema es el reconocimiento facial, que permite validar la entrada y salida de los clientes sin intervención manual, asegurando un control de acceso eficiente, seguro y sin contacto. Además, se implementó una funcionalidad de compras automatizadas, donde los usuarios pueden escanear el código QR de un producto, añadirlo a un carrito y finalizar la transacción mediante reconocimiento facial, lo cual registra automáticamente la compra en su membresía. El sistema está diseñado para operar con recursos mínimos, ya que funciona correctamente con cualquier cámara estándar y conexión a internet estable, elementos que ya se encuentran comúnmente en la mayoría de los gimnasios. Esta característica lo convierte en una solución viable incluso para establecimientos que no cuentan con infraestructura avanzada. En conjunto, el proyecto busca modernizar y simplificar la operación diaria de un gimnasio, mejorando la experiencia tanto para los usuarios como para los administradores.

**PALABRAS CLAVE:** RECONOCIMIENTO FACIAL, AUTOMATIZACIÓN DE PROCESOS, GIMNASIOS INTELIGENTES, BIOMETRÍA, LARAVEL, PYTHON, REACT, CONTROL DE ACCESOS.

**ABSTRACT**
This project presents the development of an integrated management system for gyms, aimed at automating key processes such as access control, product sales, and membership administration. The platform leverages web and biometric technologies, using Laravel, Python, React, Inertia, and MySQL to deliver a modern, accessible solution. A core feature of the system is facial recognition, allowing users to check in and out without manual verification, ensuring secure, efficient, and contactless access. Additionally, the system enables automated product purchases by scanning a product's QR code, adding it to a virtual cart, and finalizing the purchase through facial recognition, automatically associating the transaction with the user's membership.

Designed to operate with minimal resources, the system works with standard webcams and internet connections, which are widely available in most gym environments. This approach makes the solution viable for gyms that may not have access to advanced infrastructure. Beyond access and purchases, the system manages client records, active memberships, and related data in a unified digital environment, reducing administrative workload and human error. The goal is to improve operational efficiency, enhance the user experience, and provide real-time visibility for administrators over gym operations. By integrating low-cost technology and automation, this project demonstrates how digital tools can modernize traditional fitness centers. Ultimately, it offers an adaptable and scalable solution that can be implemented in a variety of gym types, supporting innovation in the health and fitness sector while minimizing implementation barriers.

**KEYWORDS:** FACIAL RECOGNITION, PROCESS AUTOMATION, SMART GYMS, BIOMETRICS, LARAVEL, PYTHON, REACT, ACCESS CONTROL.

## 1. INTRODUCCIÓN

En la actualidad, los gimnasios enfrentan desafíos relacionados con la gestión eficiente de sus operaciones, la atención al cliente y la seguridad en los accesos. La mayoría de estos establecimientos aún realiza sus procesos de manera manual o semiautomática, lo que puede generar errores, pérdidas de tiempo y una mala experiencia para los usuarios. Ante este panorama, surge la necesidad de implementar soluciones tecnológicas que automatizan y modernicen los procesos más importantes, sin requerir una infraestructura costosa. Este proyecto propone el desarrollo de un sistema web integral que automatiza el control de accesos mediante reconocimiento facial, la venta de productos a través de escaneo de códigos QR, y la gestión de membresías y clientes en un entorno centralizado. Utilizando herramientas como Laravel, Python, React, Inertia y MySQL, el sistema está diseñado para ser accesible, eficiente y funcional con equipos comunes como cámaras web y conexión a internet, los cuales ya están presentes en la mayoría de los gimnasios.

El objetivo principal del proyecto es reducir la intervención humana en procesos repetitivos, mejorar la seguridad, agilizar la atención al cliente y brindar a los administradores una plataforma robusta para el control y monitoreo de las actividades del gimnasio. Esta solución busca ser escalable, adaptable y de bajo costo, permitiendo su implementación en gimnasios pequeños, medianos o grandes que deseen evolucionar hacia una operación más inteligente y automatizada.

## 2. DESARROLLO DEL SISTEMA DE AUTOMATIZACIÓN PARA GIMNASIOS

### 2.1 **Diseño general del sistema**

El sistema desarrollado está pensado para cubrir las operaciones más comunes dentro de un gimnasio, pero de forma automatizada y accesible, sin necesidad de tener personal vigilando cada parte. La plataforma se divide en varios módulos que trabajan en conjunto:

el módulo de clientes y membresías, el módulo de control de accesos, el módulo de ventas de productos y el panel administrativo general. Todos estos se comunican a través de un backend central hecho con Laravel, un framework basado en MVC ampliamente utilizado para desarrollo web (Laravel, 2024), que sirve tanto para administrar la base de datos como para exponer la información mediante una API (Laravel, 2024).
El frontend fue construido usando React junto con Inertia.js, que permite conectar Laravel directamente con el frontend sin necesidad de APIs REST (Inertia.js, 2023). La parte del reconocimiento facial se maneja con un microservicio en Python que se comunica con Laravel. Este microservicio se encarga de identificar al cliente usando una cámara común, y regresa los datos necesarios para que el sistema decida si puede entrar o si tiene alguna restricción. Como se muestra en la figura 1, cada módulo está interconectado mediante rutas API y puede operar en conjunto o por separado. Esta estructura modular permite escalar el sistema fácilmente o adaptarlo a otros negocios con requerimientos similares.

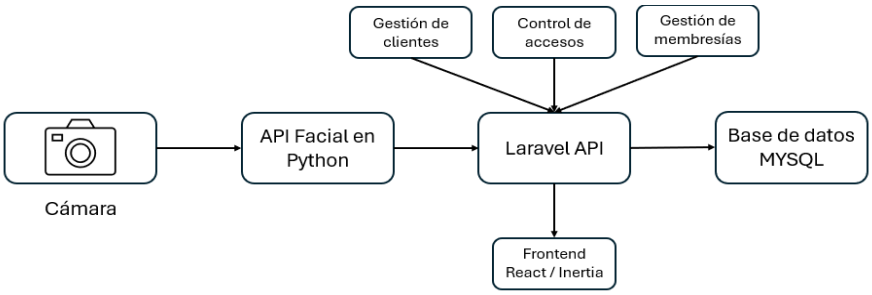


**Figura 1. Diagrama general del sistema propuesto**

## 2.2 Tecnologías utilizadas

Para construir un sistema modular y automatizado, se integraron tecnologías modernas que cubren el frontend, backend y la capa de inteligencia artificial. En el backend se utilizó Laravel, framework PHP basado en MVC. Para mostrar alertas personalizadas se utilizó la biblioteca SweetAlert2, que mejora la experiencia visual del usuario con ventanas accesibles (Limonte, 2022).
Para la base de datos, se usa MYSQL, uno de los motores de base de datos relacional más utilizados a nivel mundial (Oracle, 2023) junto con PHPMyAdmin, una herramienta web que facilita le gestión de bases de datos desde el navegador (phpMyAdmin, 2023). El modelo entidad-relación fue cuidadosamente diseñado para reflejar las relaciones entre clientes, membresías, accesos, productos y compras. Este esquema permite mantener la integridad de los datos y facilita la escalabilidad del sistema. La estructura completa del modelo puede apreciarse en la figura 2, donde se muestran las principales tablas y sus conexiones, permitiendo visualizar cómo se relacionan los módulos clave del sistema.

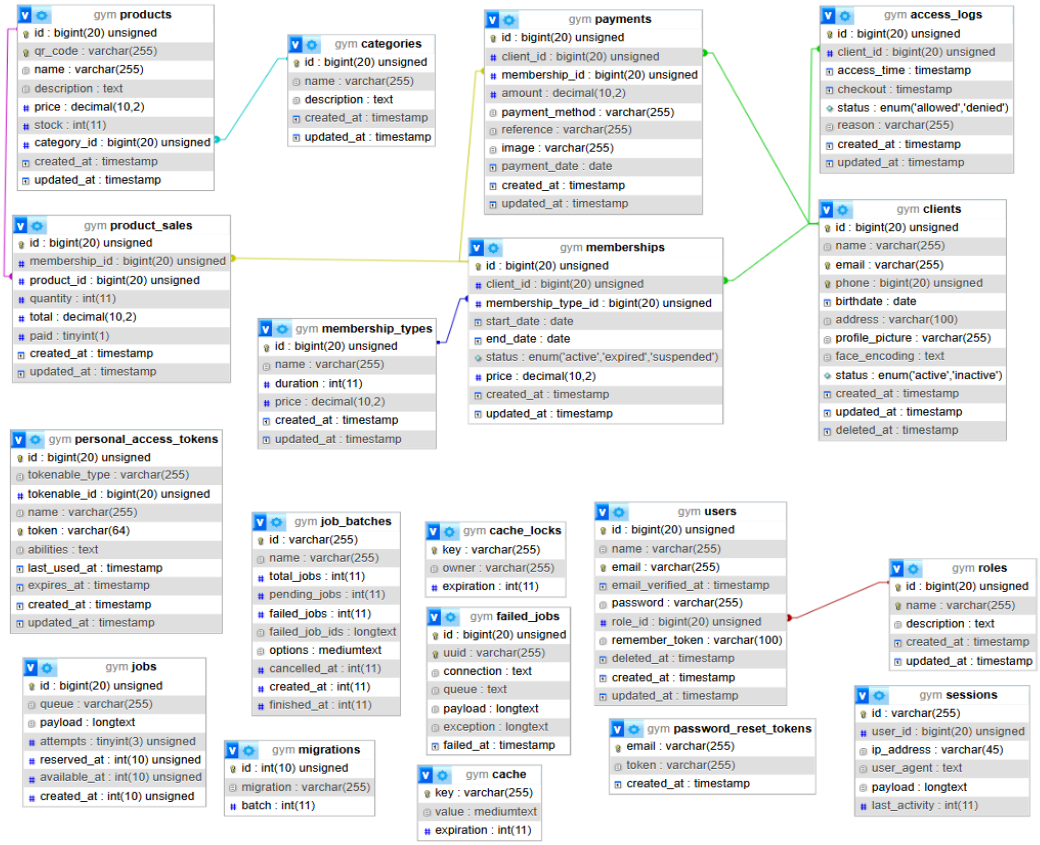


**Figura 2. Diagrama entidad-relación del sistema generado automáticamente por phpMyAdmin. Se observan las relaciones entre clientes, membresías, accesos, pagos y productos.**

El frontend se desarrolló con React, una biblioteca de JavaScript desarrollada por Meta que facilita la creación de interfaces dinámicas, permitiendo una experiencia fluida sin recargar páginas (Meta, 2023). se aplicaron estilos con Tailwind CSS, una herramienta basada en clases utilitarias que agiliza el diseño responsivo (Tailwind Labs, 2023). El reconocimiento facial se implementó como un microservicio en Python utilizando Flask, un microframework ligero para construir APIs REST (Grinberg, 2018).

Para procesar las imágenes capturadas se utilizó la biblioteca OpenCV, ampliamente empleada en visión por computadora (OpenCV, 2023). El microservicio se encarga de recibir las imágenes capturadas por la cámara, procesarlas y devolver el resultado al sistema principal desarrollado en Laravel a través de una API.

Esta separación permite mantener la lógica de inteligencia artificial desacoplada, facilitando su mantenimiento, pruebas e incluso su reutilización en otros proyectos.

Como se muestra en la figura 3, el método recognize() es el encargado de recibir la imagen en base64, decodificarla, procesarla y compararla contra los registros biométricos almacenados. La comparación facial se realiza usando face_recognition, una librería que permite codificar rostros en vectores numéricos para su comparación (Ageitgey, 2022). Si se encuentra una coincidencia, el microservicio responde con el ID correspondiente del cliente identificado, permitiendo así al sistema tomar decisiones como autorizar el acceso o registrar una compra.

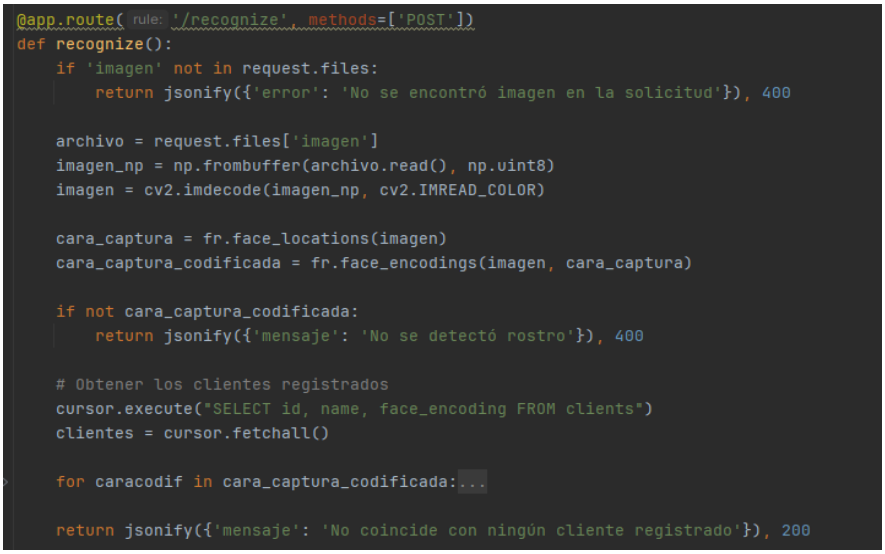


**Figura 3. Fragmento visual del método recognize() del microservicio en Flask. Se muestra el procesamiento de la imagen y la comparación biométrica con la base de datos.**

### 2.3 Módulo de gestión de clientes y membresías

El módulo de clientes y membresías permite registrar, editar y consultar información detallada de los usuarios, incluyendo su historial de pagos y el estado de su membresía. Cada cliente tiene una ficha única, y se mantiene una relación uno a uno con su membresía activa, lo cual facilita las validaciones al momento de acceder o realizar compras.

El sistema permite crear distintos tipos de membresías (mensuales, trimestrales, etc.), con opciones como precio, duración y renovación automática. Estas pueden suspenderse o cancelarse, y su estatus se actualiza automáticamente según la fecha de vencimiento. Además, si un cliente ya cuenta con una membresía activa, el sistema lo detecta y muestra un resumen en lugar del formulario de creación, evitando duplicidades y mejorando la experiencia del administrador.

### 2.4 Módulo de control de accesos con reconocimiento facial

El módulo de control de accesos con reconocimiento facial automatiza la entrada y salida de clientes, validando su identidad y el estado de su membresía en tiempo real. También permite autorizar compras, cargándolas directamente a la membresía activa del cliente.

El proceso inicia con una cámara web común que captura el rostro del cliente al acercarse. La imagen es enviada al microservicio en Python con Flask, que la compara con los registros biométricos existentes. Si hay coincidencia, se regresa el ID del cliente al backend. Laravel consulta ese ID en la base de datos y verifica si la membresía está activa y sin adeudos. Si es válido, se registra la entrada; si ya había una entrada sin salida, se registra la salida. Todo el flujo ocurre sin intervención manual.
En la figura 4 se muestra el mensaje que aparece al detectar una entrada válida por reconocimiento facial.

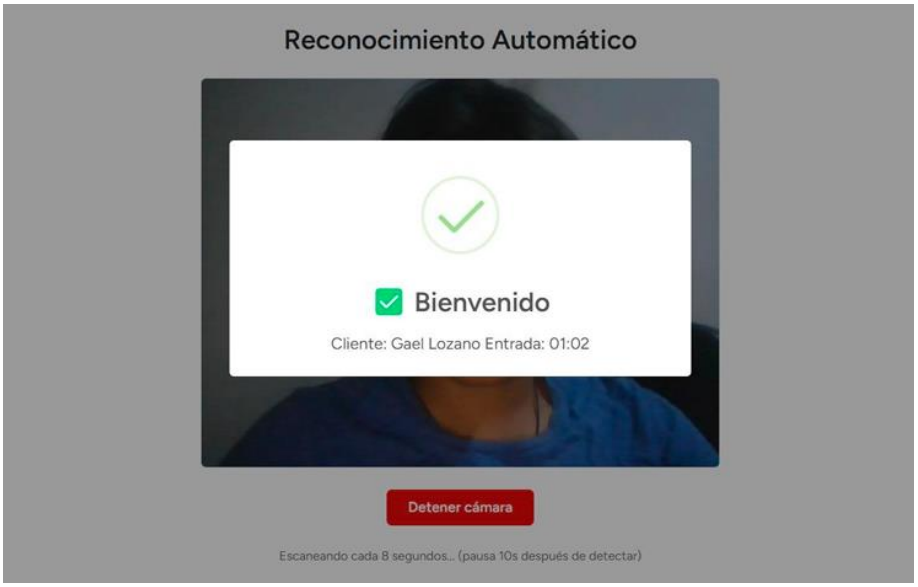


**Figura 4. Mensaje en pantalla al registrar una entrada por reconocimiento facial.**

En caso de que el cliente ya se encuentre dentro del gimnasio y se acerque nuevamente a la cámara, el sistema reconoce automáticamente que existe una entrada previa sin salida registrada. En este escenario, se interpreta como una **salida**, y el sistema actualiza el registro en la base de datos sin necesidad de intervención humana. Este proceso también genera un mensaje visual que confirma la acción, como se muestra en la figura 5. Esta lógica permite mantener un control preciso del tiempo de estancia de cada cliente, evitando entradas duplicadas o errores humanos. Además, el hecho de que el flujo se base en reconocimiento facial garantiza un registro confiable y seguro de cada acceso.

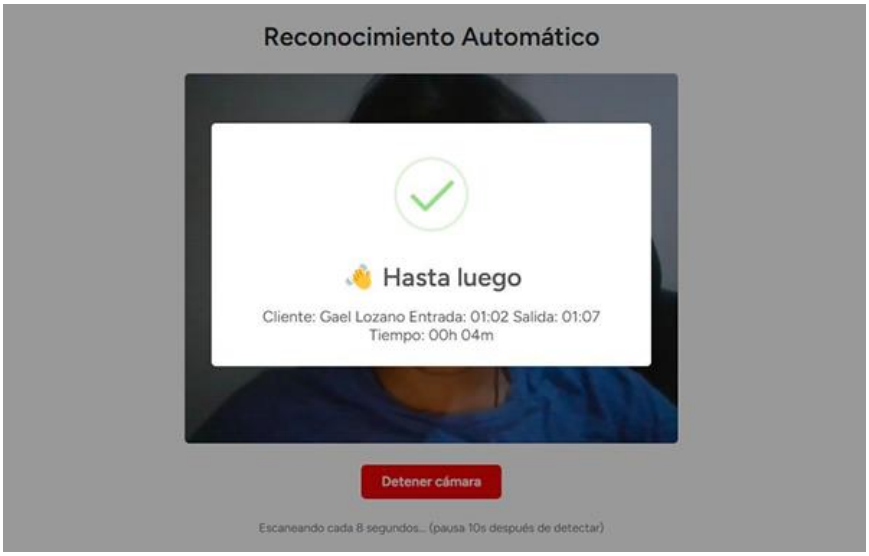


**Figura 5. Detección de salida automática tras validar rostro previamente registrado.**

Este mismo mecanismo se utiliza durante la compra de productos. El cliente escanea el código QR del producto, lo agrega a un carrito virtual y, al finalizar, se activa la cámara para validar su identidad. Si el cliente es reconocido y su membresía está activa, la compra se registra y se carga automáticamente a su membresía, quedando almacenada en el historial de ventas. Este flujo puede observarse en la figura 6, donde se muestra la interfaz del carrito y el mensaje de validación facial para confirmar la transacción.
Gracias a este módulo, se eliminan filas, se reduce la necesidad de personal en la recepción o caja, y se mejora la seguridad del gimnasio, ya que solo los clientes válidos pueden ingresar o realizar compras. Además, se conserva un registro confiable de cada entrada, salida y venta realizada.


**Figura 6. Escaneo de producto mediante QR y confirmación de agregado al carrito.**

## 2.5 Módulo de venta de productos con QR

Este módulo permite a los clientes adquirir productos dentro del gimnasio de forma ágil, sin pasar por caja ni depender de personal. Todo el proceso está diseñado para ser intuitivo, rápido y completamente automatizado. La idea principal es que cada producto tenga un código QR visible físicamente, ya sea impreso en una etiqueta o mostrado en una pantalla cercana.
El flujo comienza cuando el cliente escanea el código QR de un producto utilizando la interfaz desarrollada. El sistema detecta automáticamente el producto y lo agrega al carrito virtual, mostrando un mensaje que confirma su incorporación. En el carrito pueden verse los productos añadidos, ajustar cantidades o eliminarlos.

Una vez finalizada la selección, el sistema solicita la validación de identidad mediante reconocimiento facial. Al colocarse frente a la cámara, el microservicio en Python con Flask identifica al cliente. Si cuenta con una membresía activa, se despliega un mensaje personalizado con su nombre, tipo de membresía y los productos seleccionados, solicitando confirmación para finalizar la compra, como se observa en la figura 7.

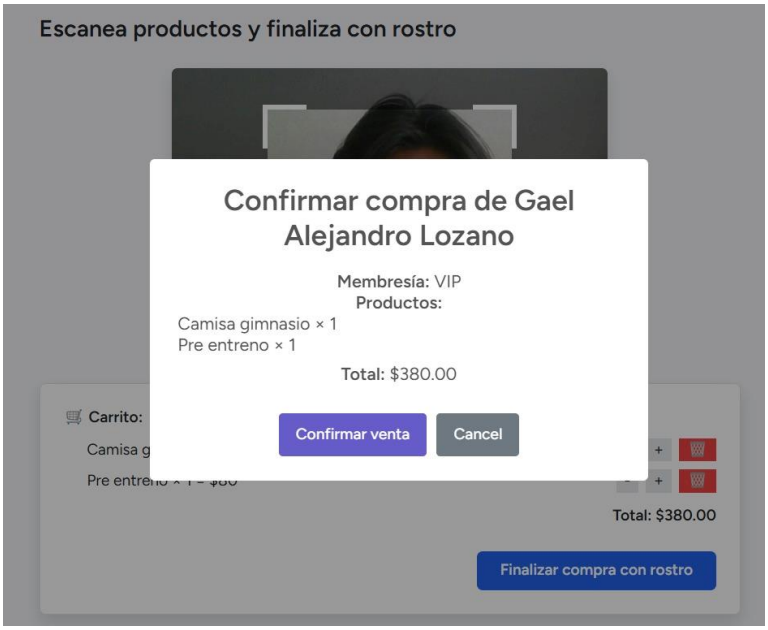


**Figura 7. Validación facial del cliente y confirmación de compra personalizada.**

Cuando el cliente confirma, la compra se registra automáticamente y se carga a su membresía activa. Toda esta operación queda almacenada en la base de datos como parte del historial del cliente.

Además, el sistema cuenta con una vista dedicada donde pueden consultarse todos los productos registrados y sus respectivos códigos QR, lo que facilita el escaneo en caso de que no estén visibles físicamente.

Este módulo aporta una experiencia moderna, fluida y segura. Elimina la necesidad de cobros presenciales, reduce filas y garantiza que cada compra quede correctamente vinculada a un cliente válido.

Cuando el cliente confirma, la compra se registra en el sistema y se carga automáticamente a su membresía.

Además, el sistema cuenta con una vista dedicada donde pueden consultarse todos los productos registrados y sus respectivos códigos QR, para facilitar el escaneo en caso de que no estén visibles físicamente. Esta vista puede apreciarse en la figura 8.

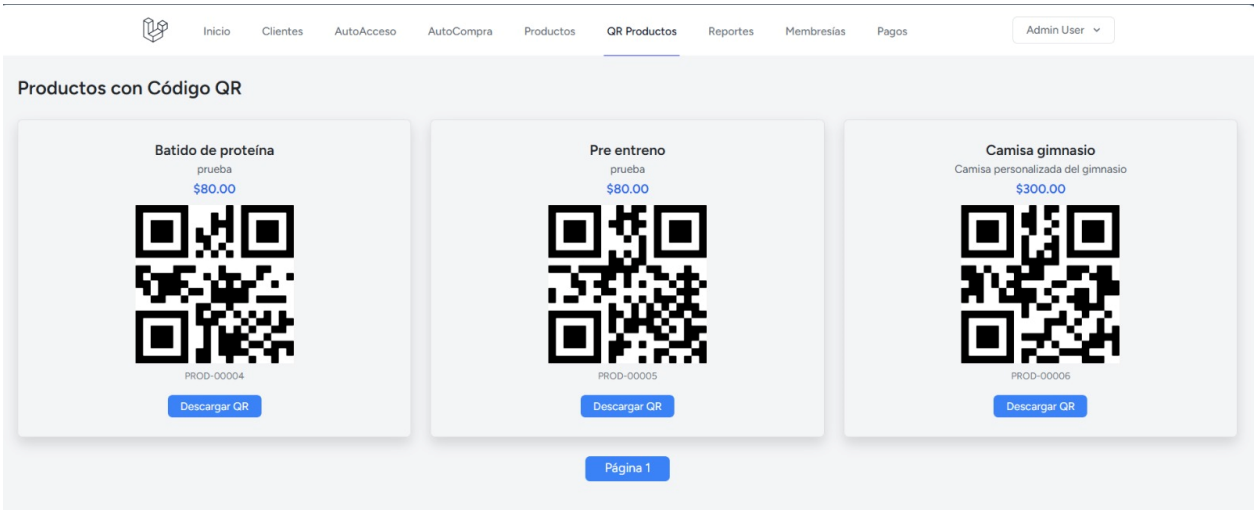


**Figura 8. Vista de productos disponibles con sus respectivos códigos QR.**

Este módulo aporta una experiencia moderna, sin fricciones y segura. Elimina la necesidad de cobros presenciales, evita filas y garantiza que todas las compras queden vinculadas a un cliente válido, con trazabilidad total.

### 2.6 Automatización e infraestructura mínima requerida

Uno de los objetivos principales del sistema es lograr una automatización completa en los procesos esenciales del gimnasio sin requerir infraestructura costosa ni especializada. Todo fue diseñado pensando en gimnasios que desean modernizar su operación, pero que no cuentan con grandes recursos tecnológicos o personal capacitado en sistemas complejos.

Desde el inicio, se buscó eliminar la necesidad de interacción humana en tareas rutinarias como el control de acceso, la venta de productos y la validación de membresías. Gracias al reconocimiento facial, las cámaras detectan al cliente de forma automática y, según el contexto, se registra su entrada, salida o se valida una compra sin tocar un solo botón. Todo esto se lleva a cabo sin intervención manual ni uso de tarjetas físicas.

En cuanto a requerimientos técnicos, el sistema puede ejecutarse correctamente en cualquier computadora de gama media con conexión a internet y una cámara web funcional. No es necesario contar con servidores dedicados ni hardware específico. La base de datos puede correr en un entorno local (por ejemplo, usando XAMPP) o en la nube, y tanto el backend como el microservicio de reconocimiento facial pueden ejecutarse en máquinas separadas o en la misma, según la capacidad del equipo disponible.

Gracias a su arquitectura modular y a la integración entre tecnologías como Laravel, Python, React e Inertia, el sistema se adapta fácilmente a distintos escenarios. Es posible activar o desactivar módulos según las necesidades del gimnasio, ya sea que solo requieran control de accesos, venta de productos o administración de membresías. Esto lo convierte en una solución viable incluso para gimnasios pequeños o recién establecidos.

## 3. RESULTADOS

Se realizaron pruebas funcionales para verificar el comportamiento del sistema en condiciones reales. En el módulo de accesos, se comprobó que el sistema reconoce correctamente a los clientes registrados, valida el estado de su membresía y registra automáticamente la entrada o salida sin intervención manual. También se negó el acceso a usuarios con membresías vencidas, mostrando mensajes adecuados.

En el módulo de ventas, el escaneo de productos mediante QR funcionó de forma precisa, y la compra se registró exitosamente tras la validación facial. Todas las operaciones quedaron almacenadas en la base de datos y vinculadas a la membresía del cliente.

Como apoyo visual y administrativo, se desarrolló una vista de monitoreo en tiempo real que muestra en una misma pantalla el historial de accesos y las compras realizadas. Esto permite supervisar el flujo del sistema de forma práctica, como se muestra en la figura 9.


**Figura 9. Vista del historial en tiempo real de accesos y compras generadas durante las pruebas.**

## 4. CONCLUSIONES

El desarrollo de este sistema web demostró que es posible automatizar los procesos clave de un gimnasio utilizando tecnologías accesibles y bien integradas. Gracias al uso de Laravel, Python y React, se logró crear una plataforma modular capaz de gestionar clientes, controlar accesos y registrar ventas, todo en tiempo real y sin intervención manual. Uno de los principales logros fue la implementación del reconocimiento facial tanto para entradas como para compras, eliminando la necesidad de tarjetas, contraseñas o personal en puntos de validación. Además, el sistema permite monitorear en vivo las operaciones mediante vistas administrativas de accesos y ventas, mejorando el control interno.

El diseño pensado para equipos comunes con conexión a internet facilita su adopción en gimnasios pequeños o medianos, incluso sin experiencia técnica avanzada. La arquitectura modular también abre la posibilidad de extender o adaptar el sistema a otros modelos de negocio con necesidades similares.

Podría decirse que el sistema cumple su objetivo principal: ofrecer una solución moderna, funcional y automatizada, con bajo requerimiento de infraestructura y alto impacto en la eficiencia operativa.

## 5. REFERENCIAS


Ageitgey, A. (2022). *face_recognition: Face recognition using dlib in Python*. https://github.com/ageitgey/face_recognition

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.

Inertia.js. (2023). *Inertia.js: The modern monolith*. https://inertiajs.com

Laravel. (2024). *Laravel: The PHP Framework for Web Artisans*. https://laravel.com

Limonte, A. (2022). *SweetAlert2*. https://sweetalert2.github.io/

Meta. (2023). *React: A JavaScript library for building user interfaces*. https://reactjs.org

OpenCV. (2023). *Open Source Computer Vision Library*. https://opencv.org

Oracle. (2023). *MySQL: The world's most popular open source database*. https://www.mysql.com

phpMyAdmin. (2023). *phpMyAdmin*. https://www.phpmyadmin.net

Tailwind Labs. (2023). *Tailwind CSS: Rapidly build modern websites*. https://tailwindcss.com


## 6. VIDEO DE FUNCIONAMIENTO

https://youtu.be/5UUWUHHmViU?si=Ld5d5QiSmMxWbfjj