



**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**  
**FACULTAD DE CIENCIAS FÍSICO**  
**MATEMÁTICAS**

Producto Integrador: Sistema de Punto de Venta  
de Suscripciones de Gimnasio  
**MANUAL TÉCNICO**

Laboratorio de programación orientada a objetos

Profesor:

José Anastasio Hernández Saldaña

Alumna:

Kelly Yiseth Cano Montiel

Matricula: 1979822

## 1. Introducción y Objetivo

El sistema **GymPOS** es una aplicación de escritorio desarrollada en **Java** utilizando la biblioteca gráfica **JavaFX**. Su objetivo es administrar los procesos operativos de un gimnasio, incluyendo la gestión de suscripciones, procesamiento de pagos y control de acceso físico.

El software fue diseñado siguiendo el patrón de arquitectura **Modelo-Vista-Controlador (MVC)** para garantizar la separación de responsabilidades, la escalabilidad y la facilidad de mantenimiento.

### 1.1 Tecnologías Utilizadas

- **Lenguaje:** java version "21.0.9" 2025-10-21 LTS.
- **Interfaz Gráfica:** JavaFX SDK 17.
- **IDE:** Apache NetBeans IDE 25.

## 2. Arquitectura del Sistema

El proyecto implementa estrictamente el patrón **MVC**, organizando el código en paquetes funcionales que separan la lógica de negocio de la interfaz de usuario.

### 2.1 Estructura de Paquetes

La aplicación se divide en los siguientes paquetes dentro de src/gymposkc9822:

- **view:** Contiene los archivos .fxml (diseño visual) y recursos gráficos. Es la capa de presentación.
- **controller:** Contiene las clases que gestionan la interacción entre la vista y el modelo. Manejan eventos (clics, entradas de texto).
- **model:** Contiene las clases POJO (Plain Old Java Objects) que representan las entidades (Cliente, Pago, etc.).
- **service:** Contiene los gestores de datos y lógica. Aquí se encuentra la persistencia (lectura/escritura de archivos .dat).

### 3. Conclusión

El desarrollo de **GymPOSKC9822** permitió la aplicación práctica de los conceptos fundamentales de la Programación Orientada a Objetos:

1. **Encapsulamiento:** Protegiendo los datos del cliente mediante modificadores de acceso privados.
2. **Herencia/Polimorfismo:** Utilizados en la estructura de los controladores y componentes de JavaFX.
3. **Abstracción:** Separando la lógica de "cómo se guarda un cliente" de "cómo se muestra en pantalla".

El resultado es un sistema funcional, resistente a fallos y que cumple con los requerimientos de gestión administrativa de un gimnasio moderno.