

Universidad Politécnica Internacional

Técnicas de Programación

Profesor: Luis Felipe Mora Umaña

Estudiante: José Armando Orozco Sibaja

III Cuatrimestre 2024

Entregable #1

## Índice

1. Introducción.....	3
Epic1: Sistema de Gestión de Gimnasio El Olimpo .....	4
Feature1.1: Gestión de Usuarios .....	4
Feature1.2: Gestión de Membresías .....	4
Feature1.3: Gestión de Clases y Reservas .....	5
Feature1.4: Gestión de Inventario .....	5
Feature1.5: Reportes .....	6
Feature1.6: Facturación .....	6
Conclusión .....	7
Análisis de Aprendizaje.....	8

## 1. Introducción

En este proyecto se realizará la planificación inicial para el desarrollo de un sistema de gestión para un gimnasio, implementado en C# utilizando la interfaz gráfica de WinForms. El desarrollo se guiará bajo principios sólidos de Programación Orientada a Objetos (POO), junto con prácticas de Clean Code, el patrón arquitectónico Modelo-Vista-Controlador (MVC), y varios patrones de diseño que optimizarán la organización y mantenimiento del código.

El sistema abarcará múltiples módulos para la administración de las operaciones del gimnasio, tales como:

- Gestión de usuarios, donde se controlarán los perfiles de clientes y entrenadores.
- Gestión de membresías, con un enfoque en la renovación y el control de pagos.
- Administración de clases y reservas, facilitando la inscripción a actividades y la asignación de horarios.
- Control de inventario, para gestionar el estado y la disponibilidad de los equipos del gimnasio.
- Generación de reportes y análisis de datos para el control financiero y de uso de las instalaciones.
- Facturación, con la creación y almacenamiento de facturas de membresías.

Para la organización y seguimiento del proyecto, se empleará Jira como herramienta de gestión de tareas. En Jira, el trabajo se estructurará en Epics, Features, y Product Backlog Items (PBIs), permitiendo una administración eficiente de cada aspecto del desarrollo.

# Epic1: Sistema de Gestión de Gimnasio El Olimpo

## Feature1.1: Gestión de Usuarios

Este módulo permite administrar los perfiles de los usuarios del gimnasio, dividiéndolos en dos tipos: clientes y entrenadores. Cada entrenador cuenta con un horario y una especialización, mientras que los clientes pueden ver las clases y hacer reservas. Los datos de usuarios y clases estarán preconfigurados mediante archivos de configuración.

### *Product Backlog Item 1.1.1: Creación y Gestión de Usuarios (Clientes y Entrenadores)*

Descripción: Como desarrollador, se necesita implementar la funcionalidad para gestionar los perfiles de clientes y entrenadores, asegurando que los datos se carguen desde archivos de configuración y se almacenen en el sistema. Los entrenadores deben tener horarios asignados y una especialización visible.

#### **Criterio de Aceptación:**

Los usuarios (clientes y entrenadores) se pueden ver en el sistema una vez cargados los datos.

Los horarios y especializaciones de los entrenadores están disponibles para consulta en el perfil de cada entrenador.

#### **Tareas:**

Cargar y validar datos de usuarios desde archivos de Excel.

Crear clase de usuario con atributos específicos para clientes y entrenadores.

Configurar interfaz de visualización de perfiles de entrenadores y clientes.

## Feature1.2: Gestión de Membresías

Este módulo se encarga de administrar las membresías y pagos de los clientes. El sistema enviará notificaciones a los clientes cuando queden 5 días o menos para el vencimiento de la membresía.

### *Product Backlog Item 1.2.1: Notificación de Membresía Próxima a Vencimiento*

Descripción: Como cliente, quiero recibir una notificación cuando falten 5 días o menos para el vencimiento de mi membresía, para asegurarme de renovarla a tiempo.

#### **Criterio de Aceptación:**

Al iniciar sesión, el cliente es notificado si su membresía vence en 5 días o menos.

La notificación aparece en la pantalla de inicio del cliente.

**Tareas:**

Crear lógica para calcular los días restantes de la membresía.

Implementar notificación visual en la interfaz del cliente.

Probar la funcionalidad de notificación en el flujo de inicio de sesión.

## Feature 1.3: Gestión de Clases y Reservas

Permite a los clientes reservar cupos para clases de Zumba, CardioDance y funcionales, con sus respectivos horarios y entrenadores asignados. Los entrenadores pueden ver las reservas y los clientes inscritos en sus clases.

### *Product Backlog Item 1.3.1: Reserva de Clases para Clientes*

Descripción: Como cliente, quiero poder reservar cupos en clases de diferentes actividades y horarios para asegurar mi participación.

**Criterio de Aceptación:**

Los clientes pueden ver la disponibilidad de clases y reservar un cupo.

La reserva se registra y se muestra en el perfil del cliente.

**Tareas:**

Configurar carga de clases con cupos y horarios.

Implementar interfaz de reserva y disponibilidad de clases para el cliente.

Crear vista de reservas confirmadas en el perfil del cliente.

## Feature 1.4: Gestión de Inventario

Administra los equipos disponibles en el gimnasio y notifica a los entrenadores cuando un equipo se acerca al final de su vida útil (3 meses o menos).

### *Product Backlog Item 1.4.1: Notificación de Equipos Próximos a Fin de Vida Útil*

Descripción: Como entrenador, quiero recibir una notificación cuando un equipo esté cerca de cumplir su vida útil para poder tomar las medidas necesarias.

**Criterio de Aceptación:**

Los entrenadores reciben una notificación cuando inician sesión, informándoles de los equipos que tienen 3 meses o menos de vida útil.

**Tareas:**

Configurar inventario de equipos con fecha de adquisición y vida útil.

Implementar lógica de cálculo de vida útil restante.

Mostrar notificación en la interfaz de entrenadores.

## Feature 1.5: Reportes

Permite generar reportes sobre la matrícula del gimnasio, el ingreso y egreso de dinero, y las clases más populares.

### *Product Backlog Item 1.5.1: Reporte de Crecimiento de Matrícula*

Descripción: Como administrador, quiero visualizar el crecimiento o disminución en la matrícula del gimnasio a lo largo del tiempo.

#### **Criterio de Aceptación:**

El sistema mostrará una tabla con los datos de inscripción de clientes en un rango de fechas.

#### **Tareas:**

Crear clase para analizar datos de matrículas.

Configurar visualización de datos en gráficos o tablas.

Implementar filtros para seleccionar el rango de fechas.

## Feature 1.6: Facturación

Gestiona el almacenamiento y consulta de facturas mensuales de membresías.

### *Product Backlog Item 1.6.1: Generación y Almacenamiento de Facturas*

Descripción: Como administrador, quiero que el sistema genere y almacene automáticamente las facturas mensuales para cada cliente y permita consultarlas en cualquier momento.

#### **Criterio de Aceptación:**

Las facturas se generan al inicio de cada mes y se almacenan en un Excel.

Los administradores pueden buscar facturas por cliente o fecha.

#### **Tareas:**

Crear lógica para generación mensual de facturas.

Configurar almacenamiento y consulta de facturas.

Implementar interfaz de búsqueda y visualización de facturas.

## Conclusión

La realización de este proyecto de gestión para un gimnasio me permitirá comprender y aplicar mejor los principios de SOLID en la Programación Orientada a Objetos (POO), patrones de diseño, y prácticas de Clean Code, lo cual ayuda a la mantenibilidad del código. Además, el uso de herramientas como Jira y GitHub ayudará a llevar una organización eficiente y controlada del proyecto, permitiendo gestionar tareas, realizar un seguimiento de avances y documentar adecuadamente el desarrollo.

## Análisis de Aprendizaje

Este proyecto me brindará la oportunidad de aprender y mejorar en varias áreas clave relacionadas con el desarrollo de software. A continuación, destaco las herramientas y conceptos que voy a dominar:

**Gestión de Proyectos con Jira:** Aprenderé a gestionar proyectos de manera más estructurada utilizando Jira. Esto me permitirá desglosar el trabajo en tareas más pequeñas, como epics, features y product backlog items (PBIs), lo que mejorará mi capacidad para planificar, organizar y seguir el progreso de un proyecto.

**Control de Versiones con GitHub:** Usar GitHub me ayudará a colaborar de forma más eficiente con otros miembros del equipo. Aprenderé a manejar el control de versiones, lo que incluye hacer revisiones de código, gestionar ramas y mantener un historial claro de todos los cambios realizados, asegurando que el trabajo en equipo sea fluido y bien organizado.

**Prácticas de Clean Code y Principios SOLID:** Aplicar las buenas prácticas de Clean Code y los principios SOLID me permitirá escribir un código más limpio, legible y fácil de mantener. Esto no solo mejorará la calidad del código, sino que también facilitará futuras modificaciones o mejoras.

**Desarrollo con MVC y POO:** Al implementar el patrón Modelo-Vista-Controlador (MVC) y aplicar principios de Programación Orientada a Objetos (POO), aprenderé a estructurar mejor las aplicaciones. El uso de estos patrones me permitirá crear un código más modular, donde los componentes sean independientes y reutilizables, lo que facilita tanto el mantenimiento como la escalabilidad del software.