Proyecto: Sistema de Gestión de Gimnasio

Nombres: Jennifer Rebeca García Moncada

Yostin Vinicio Luna Sánchez

Universidad Politécnica Internacional

90501 Técnicas de Programación

Profesor: Luis Felipe Mora Umaña

Miércoles 6 de noviembre del 2024

Universidad Politécnica Internacional

Tabla de contenido

Descripción del Proyecto 3

Epic 1: Sistema de Gestión de Gimnasio 4

- Feature 1.1: Gestión de Usuarios 4
- Product Backlog Item 1.1.1: Creación de usuarios 4
- Product Backlog Item 1.1.2: Gestión de Horarios y Características de los Entrenadores 4
- Feature 1.2: Gestión de Membresías 4
- Product Backlog Item 1.2.1: Notificación de Vencimiento de Membresía 5
- Feature 1.3: Gestión de Clases y Reservas 5
- Product Backlog Item 1.3.1: Reserva de Clases 5
- Product Backlog Item 1.3.2: Visualización de Reservas para Entrenadores 5
- Feature 1.4: Gestión de Inventario 6
- Product Backlog Item 1.4.1: Registro y Control de Activos 6
- Feature 1.5: Reportes 6
- Product Backlog Item 1.5.1: Reporte de Crecimiento de Membresías 6
- Product Backlog Item 1.5.2: Reporte Contable 6
- Feature 1.6: Facturación 7
- Product Backlog Item 1.6.1: Generación y Consulta de Facturas 7

Conclusiones y Análisis de Aprendizaje 8

Descripción del Proyecto

El objetivo de este proyecto es desarrollar un sistema de gestión para un gimnasio, utilizando el lenguaje de programación C#, APLICANDO LOS PRINCIPIOS DE Programación Orientada a Objetos (POO), Clean Code y el patrón de diseño Modelo-Vista-Controlador (MVC). El propósito de este sistema es optimizar y organizar diversas operaciones administrativas y de servicio en el gimnasio, tales como la administración de membresías, reserva de clases, control de inventarios, facturación y generación de reportes.

El sistema está diseñado para ser utilizado por dos tipos principales de usuarios: Clientes y entrenadores. Los clientes podrán interactuar con el sistema para gestionar su afiliación, reservar clases y acceder a su historial de las actividades. Por parte de los entrenadores tendrán acceso a la gestión de las clases que imparten, visualización de los participantes inscritos y acceso a informes sobre la asistencia y popularidad de las clases.

El sistema empleará archivos de configuración en formato CSV o JSON para cargar los datos de usuarios, clases y dispositivos. Esto facilitará un desarrollo modular y de fácil adaptación para iteraciones futuras. La estructura de la aplicación adoptará el modelo MVC, que simplifica la distinción entre la lógica, la interfaz del usuario y la gestión de datos, lo que permite que el sistema sea escalable y pueda adaptarse a futuras modificaciones o mejoras.

En esta etapa, también se enfocará el proyecto en implementar prácticas de Clean Code, asegurando que el código sea comprensible, sostenible y ampliable. Algunos de los principios fundamentales que se implementarán incluyen nombres significativos para clases y métodos, métodos breves y enfocados en una sola responsabilidad, y una correcta administración de excepciones y comentarios cuando se requieran. Adicionalmente, la implementación de los fundamentos SOLID y de al menos un patrón de diseño asegurará que la aplicación posea una estructura sólida y adaptable.

El sistema se construirá empleando C# y WinForms para diseñar la interfaz de usuario gráfica, ofreciendo una experiencia de usuario sencilla e intuitiva. Además, se emplearán herramientas de gestión de versiones como GitHub para administrar el registro de modificaciones en el código y Jira para estructurar y llevar a cabo un monitoreo del progreso de las distintas características, tareas y actividades del proyecto.

Epic 1: Sistema de Gestión de Gimnasio

Feature 1.1: Gestión de Usuarios

Objetivo: Permitir la creación y gestión de usuarios en el sistema, distinguiendo entre clientes y entrenadores, administrando sus horarios y puntos fuertes.

Product Backlog Item 1.1.1: Creación de usuarios

Descripción: Como administradores, Se necesita generar usuarios (clientes y entrenadores) que tengan la posibilidad de ingresar al sistema y llevar a cabo sus tareas.

Criterio de aceptación: El sistema facilita la creación de usuarios con información básica y su asignación como cliente o entrenador.

Tareas:

- Crear la clase usuario con sus atributos.
- Crear la interfaz en WinForms para el registro de usuarios.
- Configurar la carga de usuarios desde archivos CSV o JSON.

Product Backlog Item 1.1.2: Gestión de Horarios y Características de los Entrenadores

Descripción: Como administradores, se asignarán horarios y definirán puntos fuertes para cada entrenador.

Criterio de aceptación: El sistema posibilita asignarle a los entrenadores horarios y puntos fuertes a través de una interfaz de usuario.

Tareas:

- Agregar atributos de horarios y puntos fuertes en la clase Entrenador.
- Crear vista para asignar horarios y puntos fuertes.
- Configurar la carga de horarios desde archivos de configuración.

Feature 1.2: Gestión de Membresías

Objetivo: Administrar las suscripciones de los clientes, notificarles sobre vencimientos y registrar pagos.

Product Backlog Item 1.2.1: Notificación de Vencimiento de Membresía

Descripción: Como cliente, recibirá una notificación cuando falten 5 días o menos para el vencimiento la membresía.

Criterio de Aceptación: El sistema mostrará una notificación al cliente cuando inicie sesión y falten 5 días o menos para el vencimiento.

Tareas:

- Crear clase Membresia con atributos como fecha_inicio y fecha_vencimiento.
- Implementar lógica para calcular los días restantes hasta el vencimiento.
- Configurar la lógica de notificación en la interfaz de usuario de WinForms.

Feature 1.3: Gestión de Clases y Reservas

Objetivo: Permitir la reserva de clases, gestionar cupos y mostrar información relevante a los entrenadores.

Product Backlog Item 1.3.1: Reserva de Clases

Descripción: Como cliente, reservar el lugar en clases disponibles para asegurar la asistencia.

Criterio de Aceptación: El sistema permitirá reservar un lugar en una clase y actualizará el cupo disponible.

Tareas:

- Crear clases Clase y Reserva con sus atributos.
- Crear vista para realizar reservas.
- Implementar lógica para actualizar cupos.

Product Backlog Item 1.3.2: Visualización de Reservas para Entrenadores

Descripción: Como entrenador, visualizará una lista de los asistentes a sus clases.

Criterio de Aceptación: El sistema muestra al entrenador la lista de clientes registrados en sus clases.

Tareas:

- Crear vista de reservas para entrenadores.
- Sincronizar datos de reservas entre clases y entrenadores.

Feature 1.4: Gestión de Inventario

Objetivo: Controlar los activos del gimnasio e informar sobre aquellos que están por finalizar su vida útil.

Product Backlog Item 1.4.1: Registro y Control de Activos

Descripción: Como administradores, se gestionará el inventario de equipos y recibirá notificaciones sobre equipos cerca de cumplir su vida útil.

Criterio de Aceptación: El sistema notifica sobre equipos con vida útil de 3 meses o menos.

Tareas:

- Crear clase Inventario con sus atributos.
- Crear la lógica para alertar fin de vida útil.
- Crear una interfaz en WinForms para visualizar el inventario.

Feature 1.5: Reportes

Objetivo: Elaborar informes que proporcionen perspectivas acerca del aumento de las membresías, la contabilidad y la popularidad de las clases.

Product Backlog Item 1.5.1: Reporte de Crecimiento de Membresías

Descripción: Como administradores, se visualizará el crecimiento mensual de membresías.

Criterio de Aceptación: El sistema generará un gráfico con el crecimiento de membresías por mes.

Tareas:

- Crear clase Reporte.
- Implementar filtro de fechas en la interfaz.
- Crear gráfico para visualizar datos de crecimiento.

Product Backlog Item 1.5.2: Reporte Contable

Descripción: Como administradores, se visualizará un reporte financiero de ingresos y gastos.

Criterio de Aceptación: El sistema muestra un balance contable con filtros de fechas.

Tareas:

- Implementar los cálculos de ingresos vs gastos.

- Crear vista en WinForms para la visualización contable.
- Agregar opciones de exportación de los datos.

Feature 1.6: Facturación

Objetivo: Administrar el cobro de membresías y el almacenamiento de estas.

Product Backlog Item 1.6.1: Generación y Consulta de Facturas

Descripción: Como administradores, se emitirán y almacenarán facturas para cada membresía.

Criterio de Aceptación: El sistema permite consultar facturas previas y emite una factura mensual por membresía.

Tareas:

- Crear clase Factura.
- Implementar el almacenamiento y la consulta de facturas.
- Configurar la emisión mensual.

Conclusiones y Análisis de Aprendizaje

La organización y ejecución de este proyecto de la gestión de un gimnasio ha sido un ejercicio enriquecedor que nos va a brindar la oportunidad de implementar principios fundamentales de la programación orientada a objetos (POO), tales como encapsulación, herencia y polimorfismo, además de fundamentos de diseño de software como SOLID y Clean Code. Durante esta etapa inicial, hemos estructurado la labor en Epics, Features y Product Backlog Items (PBIs), lo que nos ha facilitado la organización del proyecto de manera estructurada y modular.

1. Aplicación de los principios de la POO y el Código Limpio

Durante esta fase de planificación, hemos identificado la importancia de los principios de POO para crear una base sólida y extensible para nuestro proyecto. La implementación de clases como Usuario, Membresía, Clasee Inventario, y su organización en el patrón Modelo-Viaster-Control (MVC), nos permite separar la lógica de negocio de la interfaz gráfica, lo que facilita el mantenimiento y futuras ampliaciones del sistema.

2. Aprendizaje clave

Este proyecto nos ha enseñado la importancia de una planificación detallada en el desarrollo de software. Al segmentar el sistema en elementos más reducidos y asignar responsabilidades concretas para cada funcionalidad, hemos logrado entender de manera más efectiva las demandas del sistema y prever posibles problemas de implementación.

En conclusión, esta etapa inicial de planificación ha contribuido a establecer las bases de un sistema de calidad, empleando buenas prácticas y herramientas eficientes que no solo simplifican el progreso del proyecto, sino que aseguran la viabilidad y la escalabilidad del proyecto a largo plazo.