Proyecto #1

Técnicas de programación

Integrantes:

LUIS DIEGO VILLALOBOS MOLINA

JOSEPH GONZÁLEZ CHAVES

Profesor:

LUIS FELIPE MORA UMAÑA

Fecha de entrega: 27 de noviembre

Índice

1. Introducción

En este proyecto tendremos como objetivo desarrollar un sistema de gestión para un gimnasio en C# utilizando principios de Programación Orientada a Objetos (POO) y el patrón MVC.

Este sistema debera organizar información sobre usuarios, membresías, clases, inventario, y reportes financieros.

Este documento detalla la planificación del proyecto, desglosando cada funcionalidad en los epics, features y pbis, para facilitar su desarrollo.

2. División del Proyecto

En esta sección desglosaremos el proyecto en epics, features y PBIs para estructurar y organizar el desarrollo del sistema.

Epic 1: Gestión de Usuarios

Objetivo: Crear y gestionar perfiles de usuarios (clientes y entrenadores) y permitir la autenticación en el sistema.

Feature 1.1: Creación de Usuarios

PBI 1.1.1: Crear la clase Usuario

Descripción: Como desarrollador, necesito crear una clase Usuario que almacene información básica de los usuarios.

Criterio de Aceptación: La clase debe contener atributos como Id, Nombre, y Correo.

Tareas:

Definir atributos y métodos para la clase Usuario.

Crear métodos de acceso para cada atributo.

PBI 1.1.2: Crear subclases Cliente y Entrenador

Descripción: Extender la clase Usuario con subclases Cliente y Entrenador.

Criterio de Aceptación: Cliente debe incluir **FechaMembresia** y Entrenador debe incluir **Especialidad.**

Tareas:

Crear subclase Cliente con atributo **FechaMembresia.**

Crear subclase Entrenador con atributo **Especialidad.**

Feature 1.2: Autenticación de Usuarios

PBI 1.2.1: Implementar autenticación de usuarios

Descripción: Crear una función para que los usuarios puedan iniciar sesión.

Criterio de Aceptación: El sistema debe autenticar solo si las credenciales son válidas.

Tareas:

Crear método de autenticación en Usuario.

Implementar validación de credenciales usando datos precargados.

Epic 2: Gestión de Membresías

Objetivo: Gestionar las membresías de los clientes, incluyendo notificaciones de vencimiento.

Feature 2.1: Notificación de Vencimiento de Membresía

PBI 2.1.1: Crear la clase Membresia

Descripción: Crear una clase que almacene la fecha de vencimiento de cada cliente.

Criterio de Aceptación: La clase debe incluir el atributo FechaVencimiento.

Tareas:

Definir atributos y métodos para la clase Membresia.

PBI 2.1.2: Verificar vencimiento de membresía

Descripción: Implementar una función que verifique si una membresía está próxima a vencer.

Criterio de Aceptación: Devuelve true si faltan 5 días o menos.

Tareas:

Crear método **EstaPorVencer** en **Membresia.**

Configurar notificación al iniciar sesión.

Epic 3: Gestión de Clases y Reservas

Objetivo: Permitir a los clientes reservar clases y gestionar la disponibilidad de cupos.

Feature 3.1: Reserva de Clases

PBI 3.1.1: Crear la clase **Clase**

Descripción: La clase Clase representa las clases del gimnasio, con atributos como Nombre, Horario, Cupo, y Entrenador.

Criterio de Aceptación: La clase debe tener los atributos mencionados.

Tareas:

Definir atributos de la clase **Clase.**

PBI 3.1.2: Implementar sistema de reservas

Descripción: Permitir a los clientes reservar clases verificando disponibilidad.

Criterio de Aceptación: Solo confirma la reserva si hay cupo.

Tareas:

Crear clase Reserva para gestionar reservas.

Verificar disponibilidad antes de confirmar.

Feature 3.2: Ver Detalles de Clases

PBI 3.2.1: Mostrar detalles de la clase

Descripción: Los clientes deben ver detalles de cada clase (nombre, horario, entrenador, cupo).

Criterio de Aceptación: La vista debe mostrar la información completa de cada clase.

Tareas de implementación:

Implementar método MostrarDetalles en Clase.

Crear una vista en WinForms para presentar detalles.

Epic 4: Gestión de Inventario

Objetivo: Llevar un control de los equipos del gimnasio, monitoreando su vida útil.

Feature 4.1: Monitoreo de Equipos

PBI 4.1.1: Crear clase Equipo

Descripción: La clase Equipo debe contener información sobre cada equipo (como Nombre, FechaDeAdquisicion, y VidaUtil).

Criterio de Aceptación: Incluir un método para verificar si el equipo está cerca de su vida útil.

Tareas de implementación:

Crear clase Equipo con los atributos mencionados.

Implementar método que indique si el equipo está próximo a cumplir su vida útil.

Epic 5: Reportes

Objetivo: Generar reportes de las métricas importantes del gimnasio, como asistencia y finanzas.

Feature 5.1: Generación de Reportes

PBI 5.1.1: Crear reporte de asistencia

Descripción: Generar un reporte que muestre la asistencia de los clientes a las clases por mes, rango de fechas o año en específico.

Criterio de Aceptación: Mostrar la asistencia en función del rango de fechas solicitado.

Tareas:

Implementar clase Reporte para calcular asistencia.

Crear vista que muestre el reporte de asistencia.

PBI 5.1.2: Crear reporte financiero

Descripción: Generar un reporte que muestre ingresos y gastos.

Criterio de Aceptación: Mostrar un resumen financiero basado en el rango de fechas.

Tareas:

Implementar función que calcule ingresos y gastos.

Crear vista para mostrar el reporte financiero.

Epic 6: Facturación

Gestor de Facturación

Feature 6.1: Generación y Almacenamiento de Facturas Mensuales

Descripción: El sistema debe generar una factura mensual para cada miembro con una membresía activa. Cada factura debe estar asociada a la mensualidad y se generará automáticamente al inicio de cada mes. Todas

las facturas deben ser almacenadas en el sistema y accesibles para consultas futuras.

PBI6.1.1: Creación de Facturas Mensuales

Descripción: Como administrador, necesito que el sistema genere una factura por cada miembro activo al inicio de cada mes, que refleje el pago de la membresía y servicios adicionales.

Criterio de Aceptación:

El sistema crea una factura nueva para cada usuario con membresía activa al inicio de cada mes.

La factura debe incluir el nombre del cliente, monto de la mensualidad, fecha de emisión y detalles de la membresía.

Tareas:

Crear clase para la generación de facturas.

Implementar función de cálculo de mensualidad.

Crear método de almacenamiento de facturas en la base de datos.

PBI 6.1.2: Consulta de Facturas

Descripción: Como administrador, quiero poder consultar las facturas generadas para verificar el historial de pagos de los clientes.

Criterio de Aceptación:

El sistema permite consultar todas las facturas de un cliente dado.

Las facturas pueden buscarse por fecha, cliente o estado de pago.

Tareas:

Crear pantalla de consulta de facturas.

Implementar búsqueda de facturas por diferentes criterios (cliente, fecha, estado).

3. Conclusiones

La planificación que elaboramos divide el desarrollo del sistema en epics, features y PBIs, cubriendo cada funcionalidad esencial para el gimnasio a nuestro punto de vista.

Esta pequeña estructura ayuda a abordar el proyecto paso a paso, asegurando que cada parte cumpla con los objetivos establecidos en el pdf de proyecto #1

Aprendimos a cómo utilizar correctamente el método Scrum para diseñar la división del proyecto

4- bibliografía:

https://donetonic.com/es/que-es-el-product-backlog/#:~:text=PBI%3A%20Un%20Product%20Backlog%20Item,necesitan%20para%20desarrollar%20el%20producto. – PBIs,Tareas.

https://www.youtube.com/watch?v=2X50sKeBAcQ - principios SOLID.

https://www.youtube.com/watch?v=DYDmBoV99Xg - Jira.

https://www.youtube.com/watch?v=PqV8EalcxZ0 - Clean Code.

https://www.youtube.com/watch?v=WZ8U_NHVdhI Scrum(pbis,epic,Features).

material visto en clases de teams.