



Universidad
Politécnica
Internacional

Entregable 1, Sistema de Gestión para Gimnasio

Proyecto - III Cuatrimestre 2024

Castro Mora Fabiola María,
Santamaría Allen Akane,
Ingeniería Informática

Universidad Politécnica Internacional

Técnicas de Programación

Mora Umaña Luis Felipe

Noviembre, 2024

Contenido

Introducción	3
Estructura del Proyecto - Planificación de Epics, Features, PBIs y Tareas	4
Epic 1: Gestion de Usuarios	4
Feature 1.1: Gestión de Clientes y Entrenadores	4
Product Backlog Ítem 1.1.1: Creación de Usuarios	4
Epic 2: Gestión de Membresías	4
Feature 2.1: Notificaciones de Vencimiento de Membresía	4
Product Backlog Ítem 2.1.1: Implementación de Notificaciones	4
Epic 3: Gestión de Clases y Reservas	4
Feature 3.1: Creación y Gestión de Clases	4
Feature 3.2: Sistema de Reservas	5
Product Backlog Ítem 3.2.1: Reservas de Clases	5
Epic 4: Gestión de inventario	5
Feature 4.1: Control de Equipamiento	5
Product Backlog Ítem 4.1.1: Registro de Equipos	5
Feature 4.2: Seguimiento de Uso de Equipamiento	6
Product Backlog Ítem 4.2.1: Registro de Uso	6
Epic 5: Reportes	6
Feature 5.1: Generación de Reportes de Clases y Membresías	6
Product Backlog Item 5.1.1: Reporte de Clases Populares	6
Product Backlog Item 5.1.2: Reporte de Membresías Activas	6
Epic 6: Facturación	7
Feature 6.1: Gestión de Facturas	7
Product Backlog Item 6.1.1: Generación Automática de Facturas	7
Product Backlog Item 6.1.2: Almacenamiento y Consulta de Facturas	7
Diagrama de Gantt	8
Conclusiones y Análisis de Aprendizaje	9
Bibliografía	10

Introducción

El presente documento tiene como objetivo presentar una planificación del proyecto para el desarrollo de un Sistema de Gestión para un Gimnasio, diseñado para poder optimizar la administración de recursos, usuarios y servicios dentro de un entorno fitness. A medida que la industria del bienestar y el fitness continúa creciendo, se hace evidente la necesidad de herramientas eficaces que faciliten la gestión diaria de un gimnasio, asegurando una experiencia fluida tanto para los entrenadores como para los clientes del lugar.

Este sistema busca integrar diversas funcionalidades claves que abarcan desde la gestión de usuarios y membresías hasta la programación de clases y reservas, pasando por el control de inventario y la generación de reportes. La implementación de estas características no solo mejorará la eficiencia operativa del gimnasio, sino que también proporcionará un mejor servicio a los clientes, al permitirles acceder a información relevante de manera rápida y sencilla para facilitar el uso.

El proyecto se fundamenta en principios de programación orientada a objetos (POO) y los principios SOLID, asegurando un diseño modular y extensible que facilitará futuras iteraciones y mejoras. Se adoptarán buenas prácticas de desarrollo, como el uso de Clean Code y el patrón Modelo-Vista-Controlador (MVC) visto en clase, que no solo mejoran la calidad del código, sino que también garantizan la mantenibilidad y escalabilidad del sistema a largo plazo. La primera versión del presente proyecto será presentada en C#.

La planificación de este proyecto se ha dividido en diferentes niveles, comenzando con epics que representan grandes áreas funcionales, seguidas de features que describen características específicas y, finalmente, los Product Backlog Items (PBIs) que son tareas concretas para realizar. Esta estructura permitirá un seguimiento claro del progreso y facilitará la asignación de tareas entre los miembros del equipo.

Además, se contempla la implementación de un diagrama de Gantt que visualizará la cronología del proyecto, facilitando la identificación clave y asegurando que se cumplan los plazos establecidos. La organización del trabajo se realizará de manera colaborativa utilizando herramientas de gestión como Jira, permitiendo un seguimiento detallado del avance y la resolución de posibles obstáculos que surjan durante el desarrollo de este proyecto.

Estructura del Proyecto - Planificación de Epics, Features, PBIs y Tareas

Para asegurar una planificación detallada y eficaz, el proyecto se divide en las siguientes secciones principales (Epics): Gestión de Usuarios, Gestión de Membresías, Gestión de Clases y Reservas, Gestión de Inventario, Facturación y Reportes. Cada Epic se desglosa en funcionalidades específicas (Features), que a su vez se dividen en Product Backlog Items (PBIs) y tareas específicas de implementación.

Epic 1: Gestion de Usuarios

Feature 1.1: Gestión de Clientes y Entrenadores

Permitir la creación y gestión de usuarios con roles específicos de cliente y entrenador. Los entrenadores tendrán horarios y especializaciones predeterminados, mientras que los clientes podrán ver y reservar clases.

Product Backlog Ítem 1.1.1: Creación de Usuarios

Descripción

Criterio de Aceptación: El sistema permite registrar usuarios con roles específicos y datos relevantes. Los entrenadores deben visualizar sus horarios y especialidades.

Tareas

- Crear interfaz de registro de usuarios.
- Configurar roles de usuario en la base de datos.
- Implementar lógica para asignar horarios y especialidades.

Epic 2: Gestión de Membresías

Feature 2.1: Notificaciones de Vencimiento de Membresía

Notificar a los clientes cuando su membresía esté a punto de expirar.

Product Backlog Ítem 2.1.1: Implementación de Notificaciones

Descripción

Criterio de Aceptación: El sistema envía una alerta al cliente al iniciar sesión si su membresía está a 5 días o menos de expirar.

Tareas

- Crear lógica de notificación de vencimiento.
- Configurar base de datos para alertas.
- Probar funcionalidad de alertas en el inicio de sesión.

Epic 3: Gestión de Clases y Reservas

Feature 3.1: Creación y Gestión de Clases

Permitir a los entrenadores crear y gestionar clases, incluyendo horarios, descripciones y niveles de dificultad.

Product Backlog Ítem 3.1.1: Creación de Clases

Descripción: Permitir a los entrenadores crear clases con detalles específicos.

Criterio de Aceptación: El sistema permite a los entrenadores ingresar el nombre, la descripción, la duración y el nivel de dificultad de la clase.

Tareas

- Crear interfaz para la creación de clases.
- Implementar validaciones para los campos requeridos.
- Almacenar la información de clases en la base de datos.

Feature 3.2: Sistema de Reservas

Permitir a los clientes reservar clases disponibles.

Product Backlog Ítem 3.2.1: Reservas de Clases

Descripción: Facilitar a los clientes la opción de reservar clases.

Criterio de Aceptación: Los clientes pueden ver un calendario de clases y reservar en función de su disponibilidad.

Tareas

- Desarrollar el calendario de clases.
- Implementar la lógica de reservas y disponibilidad.
- Notificar a los entrenadores sobre las reservas realizadas.

Epic 4: Gestión de inventario

Feature 4.1: Control de Equipamiento

Gestionar el inventario de equipamiento disponible para clases y entrenadores.

Product Backlog Ítem 4.1.1: Registro de Equipos

Descripción

Permitir la entrada y actualización del equipamiento en el inventario.

Criterio de Aceptación: El sistema debe permitir agregar, editar y eliminar artículos del inventario.

Tareas

- Crear interfaz para la gestión de equipamiento.
- Configurar la base de datos para almacenar información de inventario.
- Implementar lógica de notificación para bajo stock.

Feature 4.2: Seguimiento de Uso de Equipamiento

Registrar el uso del equipamiento por parte de los entrenadores y clientes.

Product Backlog Ítem 4.2.1: Registro de Uso

Descripción

Facilitar el seguimiento del uso de equipamiento durante las clases.

Criterio de Aceptación: El sistema debe permitir registrar y consultar el uso de cada equipo por clase.

Tareas

- Implementar funcionalidad para registrar el uso de equipamiento.
- Crear reportes de uso de equipamiento.

Epic 5: Reportes

Feature 5.1: Generación de Reportes de Clases y Membresías

Permitir al entrenador generar reportes sobre el estado de las clases y el crecimiento de las membresías.

Product Backlog Item 5.1.1: Reporte de Clases Populares

Descripción: Generar un informe sobre las clases y horarios más populares.

Criterio de Aceptación: El sistema genera un reporte visual sobre las clases con mayor asistencia.

Tareas:

- Crear lógica de generación de reportes.
- Implementar visualización en pantalla para los reportes.
- Probar funcionalidad de generación de reportes.

Product Backlog Item 5.1.2: Reporte de Membresías Activas

Descripción: Proveer información sobre el crecimiento de membresías activas en el gimnasio.

Criterio de Aceptación: El sistema genera un reporte sobre el crecimiento o disminución de las membresías en un rango de tiempo.

Tareas:

- Implementar lógica para calcular el crecimiento de membresías.
- Crear visualización de reportes para este propósito.
- Probar la funcionalidad de reporte de membresías.

Epic 6: Facturación

Feature 6.1: Gestión de Facturas

El sistema debe gestionar todas las facturas generadas mensualmente, almacenarlas y permitir su consulta en cualquier momento.

Product Backlog Item 6.1.1: Generación Automática de Facturas

Descripción: Generar una factura para cada cliente cada mes, en función de su membresía activa.

Criterio de Aceptación: El sistema crea una factura única para cada membresía activa al inicio de cada mes.

Tareas:

- Desarrollar lógica de generación de facturas automáticas.
- Configurar formato de factura con la información requerida (nombre del cliente, tipo de membresía, monto).
- Probar la generación automática para varios clientes con diferentes membresías.

Product Backlog Item 6.1.2: Almacenamiento y Consulta de Facturas

Descripción: Almacenar todas las facturas generadas y permitir que los clientes y el personal del gimnasio puedan consultarlas.

Criterio de Aceptación: Las facturas pueden ser consultadas por mes, cliente o número de factura en una pantalla de búsqueda.

Tareas:

- Configurar base de datos para almacenar las facturas.
- Crear interfaz de consulta para facturas en función de criterios como mes o cliente.
- Probar la funcionalidad de consulta de facturas con distintos criterios.

Diagrama de Gantt

Diagrama de Gantt (herramientas como Jira o MS Project, pegar link)

<https://fabima2213.atlassian.net/jira/software/projects/P1/boards/3/timeline?timeline=WEEKS&shard=&atlOrigin=eyJpIjoiNzA1OTdkZmNiMzEyNGIyOTg1NTQ5YWZhY2NINTE5ZTciLCJwIjoiIj9>

Conclusiones y Análisis de Aprendizaje

En conclusión el desarrollo del proyecto, estructurado en Epics, Features, PBIs y tareas, ha permitido crear una planificación detallada y organizada. Esta metodología ha facilitado la identificación de prioridades, la gestión de dependencias y la asignación clara de responsabilidades para cada persona, lo que es esencial para el éxito del proyecto y llevarlo a cabo con una mejor estructura utilizando la herramienta de Jira y obteniendo buen aprendizaje y manejo de la herramienta.

Además, la implementación de GitHub como sistema de control de versiones permitirá al equipo trabajar de manera colaborativa y eficiente, gestionando cambios y actualizaciones de forma segura. GitHub será clave para llevar un registro de las versiones del proyecto, facilitando la identificación y solución de problemas a lo largo del desarrollo. Al trabajar en ramas y fusionar cambios mediante pull requests, cada miembro del equipo podrá contribuir sin riesgo de sobrescribir el trabajo de otros, manteniendo una versión estable del código en todo momento. Esta experiencia contribuirá significativamente al desarrollo de habilidades en control de versiones, esenciales para proyectos de software a gran escala.

El uso de herramientas de gestión como Jira, en combinación con el diagrama de Gantt, facilitará el seguimiento de las tareas y el cumplimiento de los plazos, proporcionando una visión clara del avance del proyecto en cada etapa. Este proyecto representa una oportunidad valiosa de aprendizaje en planificación y ejecución de proyectos complejos, así como en el manejo de buenas prácticas de desarrollo, como SOLID y Clean Code, que serán aplicadas una vez que el desarrollo inicie.

Con esta base sólida, el equipo se encuentra preparado para iniciar el proyecto con una estructura clara y objetivos bien definidos, optimizando el uso de recursos y garantizando la calidad del sistema desde las primeras etapas de su desarrollo.

Bibliografía

<https://docs.github.com/es/get-started/start-your-journey/hello-world>

<https://www.atlassian.com/es/software/jira/guides/getting-started/basics>