Universidad Politécnica Internacional

TECNICAS DE PROGRAMACION

PROYECTO 1

ENTREGABLE #1

SISTEMA DE GESTIÓN DE GASTOS COMPARTIDOS

ESTUDIANTE

JOSE LUIS ALVARES MORALES

Indice

1. Introducción

El objetivo de este proyecto es el desarrollo de un sistema de escritorio en C#, utilizando la tecnología de Windows Forms, para gestionar gastos compartidos entre usuarios. Este sistema permitira crear grupos, registrar gastos de forma detallada, generar reporte, visualizar balances individuales y grupales, así como mostrar estadísticas claras y comprensibles.

El desarrollo se basará en los principios fundamentales de la programación orientada a objetos, incorporando buenas prácticas como Clean Code, los principios SOLID y la arquitectura MVC (Modelo-Vista-Controlador) para asegurar una estructura modular, mantenible y escalable.

Este sistema esta orientado a usuarios que necesiten llevar un control organizado de sus finanzas, como compañeros de apartamento, equipos de trabajo o familias. La aplicación ofrecerá una interfaz intuitiva y funcionalidades que ayuden a eliminar confusiones financieras y saber cuanto debe cada persona y cómo distribuir equitativamente los gastos.

Esta primera entrega corresponde a la fase de planificación inicial del proyecto. En ella se detallan las funcionalidades principales agrupadas por epic, features y PBIs, que servirán como hoja de ruta para la implementación técnica del sistema. Además, se establece la base para gestionar adecuadamente el tiempo, los recursos y las prioridades del desarrollo.

2. División del Proyecto

Epic1: Sistema de Gestión de Gastos Compartidos

Feature 1.1: Gestión de Usuarios y Grupos

Item 1.1.1: Registro de Usuarios

Descripción: El sistema deberá permitir registrar nuevos usuarios con su información básica.

Criterio de Aceptación: Se podrá crear un usuario único con nombre de usuario y contraseña.

Tareas:
1. Crear clase Usuario.
2. Crear formulario de registro.
3. Validar que no existan usuarios repetidos.

Item 1.1.2: Creación de Grupos

Descripción: Se podrán crear grupos con usuarios existentes, indicando un nombre de grupo e imagen opcional.

Criterio de Aceptación: El grupo debe poder asociar múltiples usuarios registrados.

Tareas:
1. Crear clase Grupo.
2. Relacionar grupo con usuarios existentes.
3. Crear interfaz para mostrar usuarios y crear grupos.

Feature 1.2: Registro de Gastos

Item 1.2.1: Ingreso de Gastos

Descripción: Los usuarios podrán registrar gastos indicando nombre, descripción, monto, quién pagó, a quiénes aplica, y fecha.

Criterio de Aceptación: El sistema debe permitir agregar un gasto al grupo activo y mostrarlo en una lista.

Tareas:
1. Crear clase Gasto.
2. Crear interfaz para ingresar y mostrar gastos.
3. Validar datos de entrada.

Feature 1.3: Reportes y Estadísticas

Item 1.3.1: Visualización de Balance General

Descripción: Se generará un resumen con los saldos y deudas entre los miembros del grupo.

Criterio de Aceptación: El sistema muestra en pantalla el balance actual por usuario.

Tareas:
1. Calcular saldo neto por usuario.
2. Generar reporte visual con tablas.

Item 1.3.2: Reportes por Fecha

Descripción: Se podrán generar reportes de gastos por intervalos de fecha definidos por el usuario.

Criterio de Aceptación: Se muestra un listado de gastos filtrado por fecha.

Tareas:
1. Agregar opción de filtrar gastos por fecha.
2. Mostrar resultados en pantalla o archivo.

3. Conclusiones

Durante la estructuración de este entregable se aplicaron conocimientos clave en el análisis de requerimientos, el diseño orientado a objetos y el uso de metodologías ágiles para planificar el desarrollo del sistema. La división del proyecto en epics, features y PBIs permitió descomponer el Proyecto en partes manejables, enfocadas en aportar valor de forma progresiva.

Esta planificación no solo facilita la organización y un seguimiento ordenado, sino que también permite prever los módulos que requerirán mayor complejidad técnica, definiendo con anticipación los retos y los recursos necesarios. Además, el uso de buenas prácticas como Clean Code, los principios SOLID y la arquitectura MVC desde la etapa de diseño garantiza que la futura implementación será más limpia, modular y fácil de mantener.