



ESTRUCTURA DE DATOS

90501 G2

Entrega Proyecto 1

Tema: SplitBuddies

Estudiante:

Pablo Andrés Torres Villalobos

Prof. Luis Felipe Mora Umaña

II Cuatrimestre 2025

Indice

Introducción 3

Objetivos General 3

Objetivos Específicos 3

Planificación del Proyecto..... 4

 Menu 1: Gestión de usuarios 4

 Menu 2: Grupos de gasto 4

 Menu 3: Registro de gastos 5

 Menu 4: Estadísticas y reportes 5

 Menu 5: Importación de datos..... 6

 Diagrama de Gantt 6

Conclusiones 7

Referencias 7

Introducción

SplitBuddies es una herramienta que busca facilitar la vida cuando se trata de dividir gastos entre varias personas, ya sea entre amigos, compañeros de cuarto o familiares. La finalidad es permitir registrar, organizar y consultar fácilmente quién pagó qué, a quién le toca poner cuánto y cómo están los saldos entre todos.

Este proyecto solicitado aplicará principios como MVC, SOLID y Clean Code para asegurar que sea escalable y fácil de mantener.

Objetivos General

Desarrollar un programa que cumpla con registrar, organizar, distribuir y consultar gastos.

Objetivos Específicos

- Desarrollar una aplicación sencilla e intuitiva para dividir gastos en grupo.
- Aplicar principios de programación orientada a objetos de manera práctica.
- Usar buenas prácticas de desarrollo como SOLID, Clean Code y separación en capas (MVC).
- Preparar una base sólida para el resto del desarrollo del proyecto.

Planificación del Proyecto

Planeación por menús, según requerimiento:

Menu 1: Gestión de usuarios

Esta Menu cubre todo lo relacionado con la administración de los usuarios que usarán la aplicación.

- Registro de usuario nuevo
 - Crear la clase con los atributos esenciales (nombre, correo, etc.)
 - Diseñar un formulario de registro en Windows Forms con validaciones básicas para asegurar la integridad de los datos.
 - Implementar la lógica para mostrar mensajes claros de éxito o error al usuario.
- Inicio de sesión
 - Desarrollar un formulario de login sencillo.
 - Implementar la validación de credenciales para permitir el acceso solo a usuarios registrados.

Menu 2: Grupos de gasto

Enfocado en la funcionalidad para crear y gestionar los grupos donde se compartirán los gastos.

- Crear grupos
 - Definir la clase Grupo.cs con propiedades como nombre, una posible imagen representativa y una lista de sus miembros.
 - Crear un formulario para que los usuarios puedan generar nuevos grupos, seleccionando a otros usuarios ya registrados.
 - Asegurar que se valide un mínimo de dos miembros por grupo.
- Ver balances del grupo

- Implementar la lógica para calcular el saldo de cada miembro dentro de un grupo (quién debe a quién y cuánto).
- Diseñar una interfaz para mostrar este resumen de saldos de manera clara en pantalla.

Menu 3: Registro de gastos

Este Menu, permitiendo a los usuarios registrar y consultar sus gastos.

- Registrar nuevo gasto
 - Crear la clase Gasto.cs que incluya descripción, monto, fecha, quién realizó el pago y a quiénes afecta el gasto.
 - Desarrollar un formulario intuitivo para el registro de gastos.
 - Implementar validaciones para asegurar que el gasto esté correctamente distribuido entre los miembros seleccionados.
- Historial de gastos
 - Diseñar una tabla o lista para visualizar todos los gastos de un grupo específico.
 - Añadir la posibilidad de filtrar el historial por fecha o por usuario involucrado.

Menu 4: Estadísticas y reportes

Este Menu se centra en ofrecer a los usuarios una visión global de sus finanzas grupales.

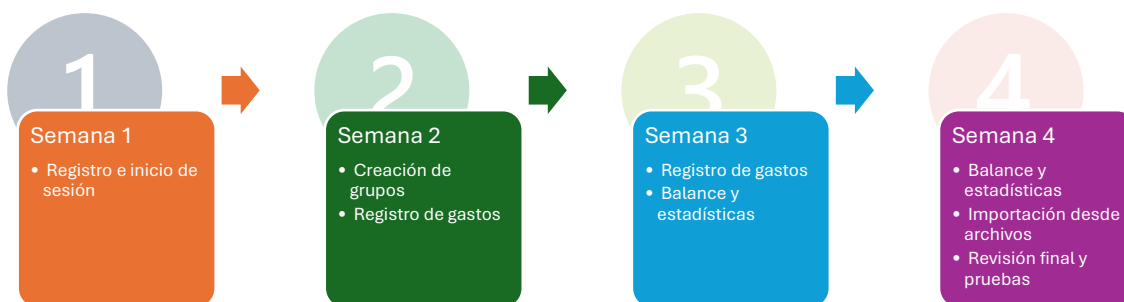
- Balance general de usuario
 - Calcular y mostrar cuánto ha pagado un usuario en total versus cuánto debe.
 - Presentar promedios y totales generales para cada usuario.
- Reportes por fecha
 - Desarrollar un filtro que permita ver los gastos mensuales, anuales o diarios.
 - Preparar la lógica para una futura funcionalidad de exportación de estos reportes.

Menu 5: Importación de datos

Menu clave para facilitar la carga masiva de información.

- Leer archivo CSV de gastos
 - Implementar la funcionalidad para leer archivos CSV con al menos 100 registros de gastos.
 - Asegurar la validación de la estructura y el contenido del CSV para evitar errores.
- Cargar usuarios y grupos desde JSON
 - Desarrollar la capacidad de leer archivos JSON que contengan al menos 50 usuarios y 30 grupos.
 - Implementar el manejo de errores para estructuras JSON inválidas y asegurar que los datos se muestren correctamente en el sistema.

Diagrama de Gantt



Conclusiones

La planificación de este proyecto permite ver con claridad el alcance y las tareas necesarias para lograr una aplicación funcional y bien estructurada. También comprender la magnitud de la complejidad de lo solicitado para un proyecto educativo. Más allá del código, este trabajo es una excelente oportunidad para aplicar prácticas reales de desarrollo profesional: aprender a trabajar por tareas, usar control de versiones, diseñar una arquitectura limpia y enfocarnos en un desarrollo modular. También permite experimentar con el uso de herramientas como Jira y Git, que son esenciales en entornos reales de trabajo en equipo.

Referencias

- Atlassian. (n.d.). *Introducción a Jira: guía completa para principiantes*. Atlassian. Retrieved July 7, 2025, from <https://www.atlassian.com/es/software/jira/guides/getting-started/introduction>
- Clean code: principios, ventajas y ejemplos*. (2022, December 2). IONOS Digital Guide. <https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/clean-code-que-es-el-codigo-limpio/>
- Fundamentos de MVC - El pequeño libro de Asp.net Core*. (n.d.). Aspnetcoremaster.com. Retrieved July 7, 2025, from <https://aspnetcoremaster.com/little-aspnetcore-book/chapters/mvc-basics/>
- Martins, J. (2025, February 2). *Diagrama de Gantt: qué es y cómo crear uno con ejemplos*. Asana. <https://asana.com/es/resources/gantt-chart-basics>
- Rodriguez, D. S. (2022, November 28). *Los principios SOLID de programación orientada a objetos explicados en Español sencillo*. freecodecamp.org. <https://www.freecodecamp.org/espanol/news/los-principios-solid-explicados-en-espanol/>