

INDICE DE CONTENIDOS

1 INTRODUCCIÓN.....	1
2 PLANTEAMIENTO DEL PROBLEMA.....	2
3 OBJETIVOS.....	3
3.1 OBJETIVO GENERAL.....	3
3.2 OBJETIVOS ESPECÍFICOS.....	3
4 MARCO TEÓRICO CONCEPTUAL-CONTEXTUAL.....	4
4.1 PROGRAMACIÓN ESTRUCTURADA.....	4
4.2 FUNCIONES Y MODULARIDAD.....	4
4.3 ESTRUCTURAS DE DATOS EN PYTHON.....	4
4.4 ALGORITMOS DE CÁLCULO PROPORCIONAL.....	4
4.5 INTERFAZ DE USUARIO POR TERMINAL.....	4
4.6 VALIDACIÓN DE DATOS.....	4
5 MÉTODOS, TÉCNICAS Y HERRAMIENTAS.....	5
5.1 MÉTODOS.....	5
5.2 TÉCNICAS.....	5
5.3 HERRAMIENTAS.....	6
6 MARCO APLICATIVO.....	7
6.1 ARQUITECTURA DEL PROGRAMA.....	7
6.2 MÓDULO DE INTERFAZ Y CAPTURA DE DATOS.....	7
6.3 MÓDULO DE CÁLCULOS.....	7
6.4 MÓDULO DE PRESENTACIÓN DE RESULTADOS.....	8
6.5 FLUJO DE EJECUCIÓN.....	8
6.6 CASOS DE PRUEBA.....	9
7 CONCLUSIÓN.....	10

ÍNDICE DE GRAFICOS

FIGURA 1: ALGORITMO PRINCIPAL.....	12
FIGURA 2: FUNCION MOSTRAR MENU.....	13
FIGURA 3: FUNCION CAPTURAR DATOS.....	14
FIGURA 4: FUNCION CALCULAR PROPINAS POR PERSONA.....	15
FIGURA 5: FUNCION PROPINA.....	15
FIGURA 6: FUNCION TOTAL.....	15
FIGURA 7: FUNCION CALCULAR TOTALES FINALES.....	16
FIGURA 8: FUNCION MOSTRAR QUIEN PAGA.....	16
FIGURA 9: FUNCION MOSTRAR TICKET.....	17

INDICE DE ANEXOS

ANEXO N° 1: ÁRBOL DE PROBLEMAS Y OBJETIVOS.....	18
ANEXO N° 2: CÓDIGO FUENTE COMPLETO DEL PROGRAMA.....	19
ANEXO N° 3: MANUAL DE USUARIO.....	22

1 INTRODUCCIÓN

En la actualidad, la gestión de gastos compartidos entre grupos de personas representa un desafío común en la vida cotidiana. Cuando un grupo de amigos, compañeros de trabajo o familiares comparten una comida en un restaurante, surge la necesidad de dividir los gastos de manera equitativa y transparente. Este proceso, aparentemente simple, se complica cuando cada persona ha consumido diferentes productos y es necesario calcular tanto los subtotales individuales como la distribución proporcional de la propina.

El presente trabajo tiene como finalidad desarrollar un programa automatizado de división de gastos mediante un script en lenguaje Python. Este programa permitirá registrar los consumos individuales de cada persona, calcular automáticamente los montos correspondientes incluyendo la propina proporcional, y presentar los resultados de manera clara y organizada.

La metodología empleada se basa en el paradigma de programación estructurada, dividiendo el programa en tres módulos principales: captura de datos, procesamiento de cálculos y presentación de resultados. Esta estructura modular facilita el desarrollo colaborativo entre los tres integrantes del equipo y garantiza un código mantenible y escalable.

El desarrollo de este programa contribuye a la aplicación práctica de los conocimientos adquiridos en la asignatura de Programación I, demostrando cómo la programación puede resolver problemas cotidianos de manera eficiente y automatizada.

2 PLANTEAMIENTO DEL PROBLEMA

¿Cómo calcular de manera precisa y equitativa la distribución de gastos compartidos en establecimientos gastronómicos cuando los consumos individuales son diferentes?

En situaciones de convivencia social, especialmente cuando grupos de personas comparten gastos en restaurantes o eventos, surge la problemática de calcular de manera precisa y justa la cantidad que cada persona debe aportar. Esta situación se agrava cuando:

Los consumos individuales son diferentes entre los participantes, lo que requiere un cálculo detallado por persona.

La propina debe distribuirse de manera proporcional al consumo de cada individuo, no de forma equitativa, generando confusión sobre el monto exacto que cada uno debe pagar.

Los cálculos manuales son propensos a errores aritméticos, especialmente cuando el grupo es numeroso o los montos son variados.

El tiempo invertido en realizar estas operaciones manualmente resta tiempo de convivencia y puede generar incomodidad entre los participantes.

La falta de un registro claro y detallado de los consumos puede derivar en malentendidos o desacuerdos sobre los montos a pagar.

Esta problemática se presenta con frecuencia en la vida universitaria, donde grupos de estudiantes comparten gastos y requieren métodos ágiles y confiables para dividir cuentas. La ausencia de una herramienta automatizada genera pérdida de tiempo, posibles errores de cálculo y situaciones incómodas que pueden afectar las relaciones interpersonales del grupo.

3 OBJETIVOS

3.1 OBJETIVO GENERAL

Desarrollar un programa automatizado de división equitativa de gastos mediante un script ejecutable en lenguaje Python, utilizando estructuras de datos y funciones modulares, optimizando el tiempo de cálculo y minimizando errores aritméticos.

3.2 OBJETIVOS ESPECÍFICOS

1. Diseñar una interfaz de usuario mediante terminal para capturar de manera intuitiva los datos de consumo individual de cada participante del grupo.
2. Implementar algoritmos de cálculo para determinar los subtotales por persona, el monto total de propina y su distribución proporcional según el consumo individual.
3. Desarrollar módulos de presentación de resultados para generar reportes detallados y resúmenes claros del desglose de pagos por participante.
4. Estructurar el código fuente de manera modular y documentada para facilitar el trabajo colaborativo entre los tres integrantes del equipo de desarrollo.
5. Validar el correcto funcionamiento del programa mediante casos de prueba con datos reales para asegurar la precisión en los cálculos de consumo en establecimientos gastronómicos.

4 MARCO TEÓRICO CONCEPTUAL-CONTEXTUAL

4.1 PROGRAMACIÓN ESTRUCTURADA

La programación estructurada organiza el código mediante secuencia, selección e iteración (Joyanes, 2008), mejorando legibilidad y mantenimiento del software.

4.2 FUNCIONES Y MODULARIDAD

Las funciones son bloques reutilizables para tareas específicas (Lutz, 2013). La modularidad divide problemas complejos en partes manejables, facilitando desarrollo y detección de errores.

4.3 ESTRUCTURAS DE DATOS EN PYTHON

Python ofrece diccionarios (pares clave-valor) y listas (colecciones ordenadas), fundamentales para manejar datos complejos (Matthes, 2019).

4.4 ALGORITMOS DE CÁLCULO PROPORCIONAL

Distribuye cantidades según criterios específicos. Fórmula para propinas: Propina Individual = Propina Total × (Consumo Individual / Consumo Total), garantizando equidad proporcional.

4.5 INTERFAZ DE USUARIO POR TERMINAL

Interacción mediante texto, eficiente y compatible. Python usa `input()` y `print()` para comunicación básica con el usuario.

4.6 VALIDACIÓN DE DATOS

Verifica que datos ingresados cumplan criterios de tipo, rango y formato, previniendo errores y mejorando robustez del programa.

5 MÉTODOS, TÉCNICAS Y HERRAMIENTAS

5.1 MÉTODOS

5.1.1 Método de desarrollo estructurado

Se aplicó la metodología de programación estructurada, dividiendo el programa en tres módulos principales con responsabilidades específicas y bien definidas.

5.1.2 Método de prueba y error

Durante el desarrollo se implementaron pruebas iterativas con diferentes casos de uso para validar el correcto funcionamiento de los algoritmos de cálculo.

5.2 TÉCNICAS

5.2.1 Modularización de código

División del programa en funciones independientes que realizan tareas específicas, facilitando el mantenimiento y la colaboración entre integrantes.

5.2.2 Documentación de código

Uso de comentarios descriptivos y nombres de variables significativos para mejorar la legibilidad del código fuente.

5.2.3 Diseño top-down

Diseño del programa desde lo general hacia lo específico, comenzando con la estructura principal y descendiendo hacia los detalles de implementación.

5.3 HERRAMIENTAS

5.3.1 Lenguaje de programación

Python 3.8 o superior, seleccionado por su sintaxis clara, amplia documentación y compatibilidad multiplataforma.

5.3.2 Editor de código

Visual Studio Code o cualquier editor de texto compatible con Python.

5.3.3 Terminal o consola

Interfaz nativa del sistema operativo para la ejecución del programa.

5.3.4 Sistema operativo

Compatible con Windows, macOS y Linux, garantizando portabilidad del programa.

6 MARCO APLICATIVO

6.1 ARQUITECTURA DEL PROGRAMA

El programa desarrollado se estructura en tres módulos principales que interactúan de manera secuencial para procesar la información de división de gastos. La arquitectura implementada sigue el patrón de entrada-proceso-salida, garantizando claridad en el flujo de datos.

6.2 MÓDULO DE INTERFAZ Y CAPTURA DE DATOS

Este módulo es responsable de la interacción con el usuario. Implementa dos funciones principales:

La función `mostrar_menu()` presenta las opciones disponibles al usuario mediante texto formateado en la terminal. Esta función no recibe parámetros y su único propósito es la presentación visual del menú principal.

La función `capturar_datos()` gestiona todo el proceso de ingreso de información. Primero solicita el número de participantes y almacena sus nombres en una lista. Posteriormente, para cada participante, solicita los ítems consumidos y sus precios, acumulando el total en un diccionario donde las claves son los nombres y los valores son los montos totales consumidos. Esta función retorna el diccionario de consumos para su procesamiento posterior.

6.3 MÓDULO DE CÁLCULOS

Este módulo concentra toda la lógica matemática del programa. Se compone de cuatro funciones especializadas:

La función `calcular_total()` recibe el diccionario de consumos y retorna la suma total de

todos los consumos mediante iteración sobre los valores del diccionario.

La función calcular_propina() aplica el porcentaje de propina al total de los gastos, retornando el monto de propina total.

La función calcular_propina_por_persona() implementa el algoritmo de distribución proporcional. Para cada persona, calcula su proporción de consumo respecto al total y aplica esa proporción al monto total de propina. Esta función retorna un diccionario con las propinas individuales.

La función calcular_totales_finales() suma el consumo individual y la propina correspondiente de cada persona, generando el monto final que cada uno debe pagar.

6.4 MÓDULO DE PRESENTACIÓN DE RESULTADOS

Este módulo formatea y presenta la información procesada de manera clara y profesional.

La función mostrar_ticket() genera un reporte detallado que incluye el desglose por persona mostrando consumo, propina y total individual. Adicionalmente presenta un resumen general con subtotal de consumos, total de propina y monto general de los gastos. El formato utiliza líneas de separación y alineación de texto para mejorar la legibilidad.

La función mostrar_quien_paga() presenta un resumen simplificado que lista a cada participante con su monto total a pagar, facilitando la consulta rápida de la información.

6.5 FLUJO DE EJECUCIÓN

El programa principal ejecuta un bucle que mantiene el programa activo hasta que el usuario decida salir. En cada iteración, presenta el menú y captura la opción seleccionada. Si el usuario elige dividir un gasto, se ejecuta secuencialmente: captura de datos, procesamiento de cálculos y presentación de resultados. Esta estructura garantiza que cada módulo cumpla su función específica sin interferir con los demás.

6.6 CASOS DE PRUEBA

Se realizaron pruebas con diferentes escenarios:

Caso 1: Tres personas con consumos variados y propina del 15 por ciento. El programa calculó correctamente la distribución proporcional.

Caso 2: Dos personas con consumos idénticos y propina del 10 por ciento. El programa dividió la propina equitativamente como se esperaba.

Caso 3: Cuatro personas con consumos muy dispares y propina del 20 por ciento. El programa mantuvo la precisión en decimales y la proporcionalidad correcta.

7 CONCLUSIÓN

El desarrollo del programa de división de gastos cumplió satisfactoriamente con el objetivo general planteado, logrando automatizar el proceso de cálculo de gastos compartidos mediante un programa funcional y eficiente en lenguaje Python.

Los objetivos específicos fueron alcanzados en su totalidad. Se diseñó una interfaz de usuario intuitiva que permite la captura de datos de manera sencilla. Los algoritmos de cálculo implementados garantizan precisión en la determinación de subtotales y distribución proporcional de propina. Los módulos de presentación generan reportes claros y profesionales que facilitan la comprensión de los resultados.

La estructura modular del código permitió una distribución equitativa del trabajo entre los tres integrantes, demostrando las ventajas de la programación estructurada y el desarrollo colaborativo. Cada módulo puede ser mantenido y mejorado de forma independiente sin afectar el funcionamiento de los demás.

El programa desarrollado representa una solución práctica a un problema cotidiano, demostrando cómo los conocimientos de programación pueden aplicarse para resolver necesidades reales. La portabilidad del programa, compatible con cualquier sistema operativo que soporte Python, amplía su utilidad y accesibilidad.

Como trabajo futuro, se propone implementar funcionalidades adicionales como almacenamiento persistente de historial, generación de reportes en formato PDF y desarrollo de una interfaz gráfica de usuario para mejorar la experiencia de uso.

BIBLIOGRAFÍA

- Joyanes, L. (2008). *Fundamentos de programación: algoritmos, estructura de datos y objetos* (4.^a ed.). McGraw-Hill.
- Lutz, M. (2013). *Learning Python* (5.^a ed.). O'Reilly Media.
- Matthes, E. (2019). *Python Crash Course: A Hands-On, Project-Based Introduction to Programming* (2.^a ed.). No Starch Press.
- Python Software Foundation. (2024). *Python 3.12 Documentation*.
<https://docs.python.org/3/>
- Sommerville, I. (2011). *Ingeniería de software* (9.^a ed.). Pearson Educación.

FIGURA 1: ALGORITMO PRINCIPAL

Fuente: Pseint

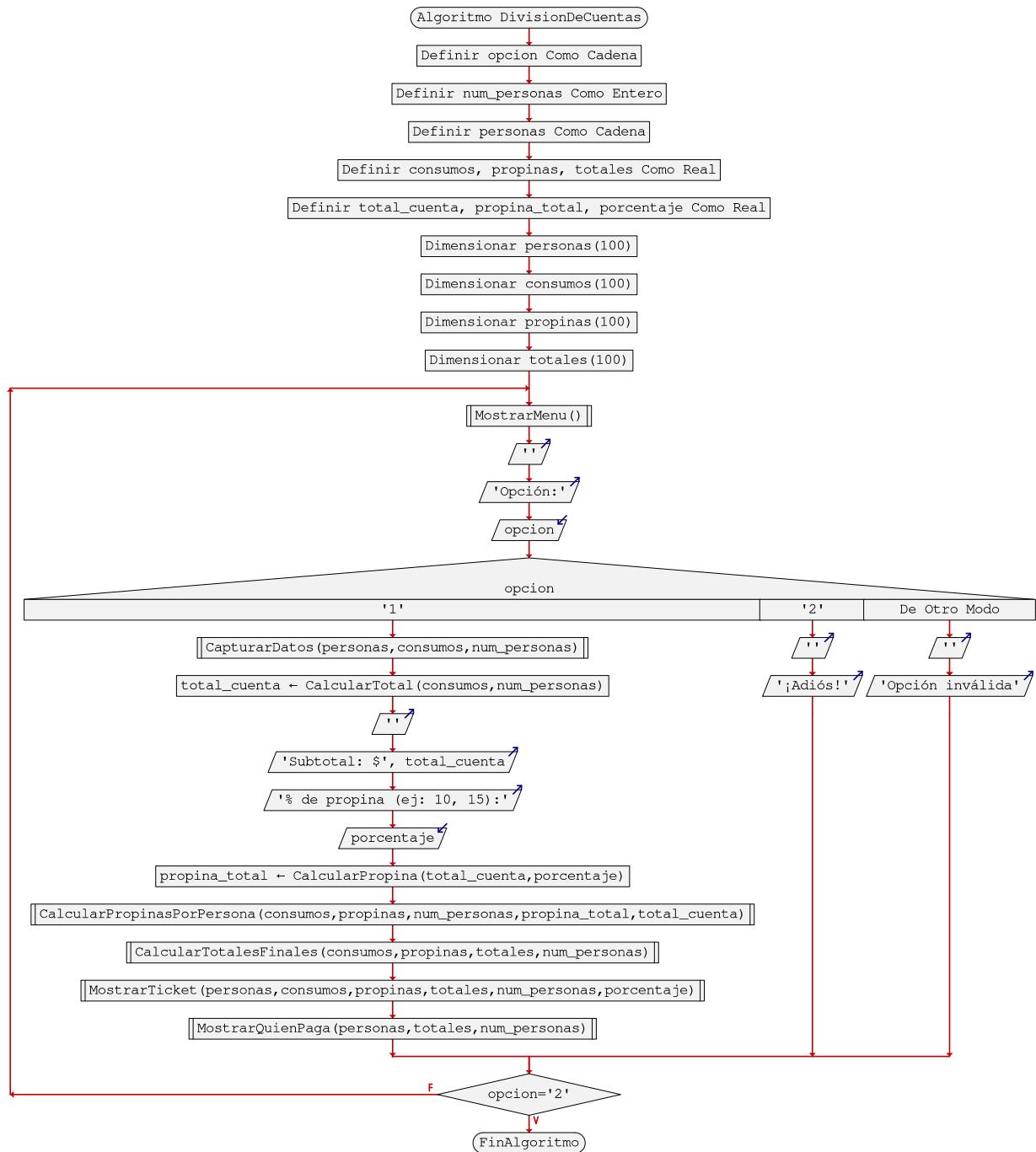


FIGURA 2: FUNCION MOSTRAR MENU

Fuente: Pseint

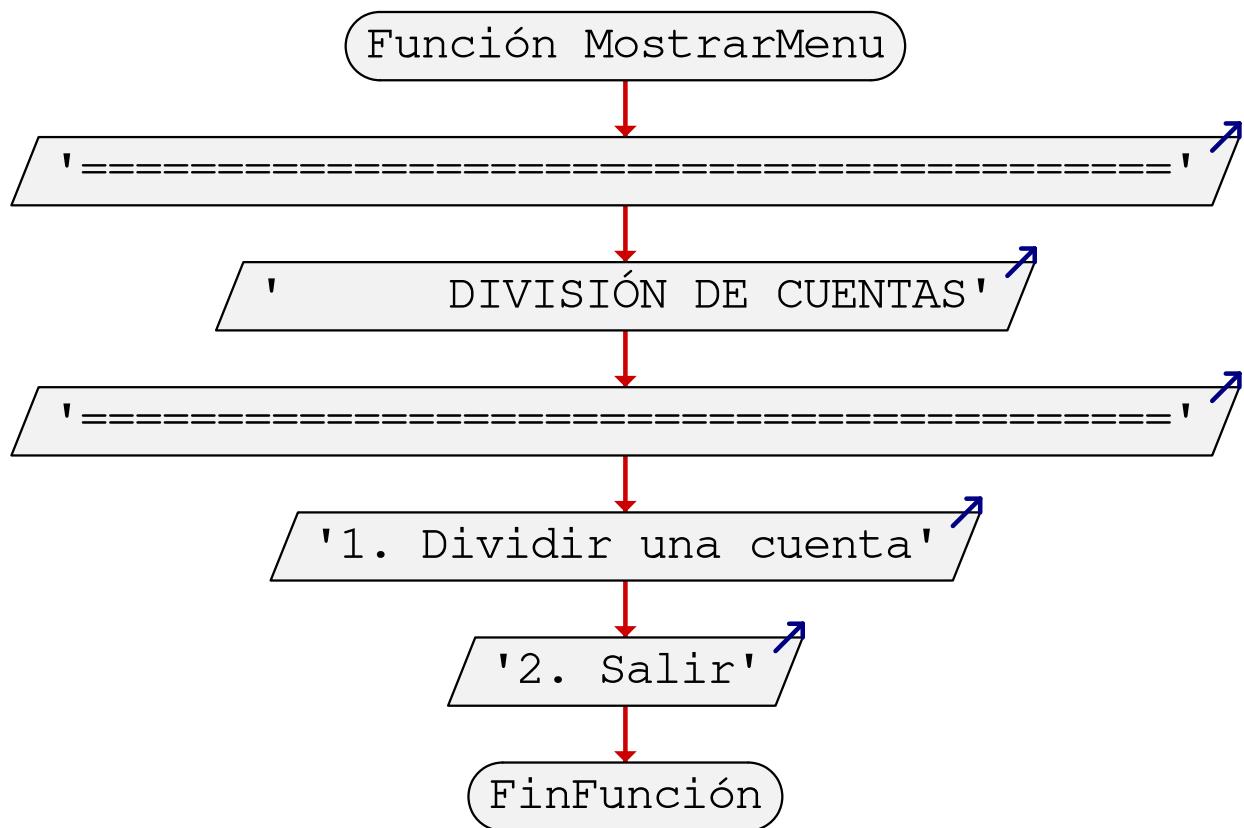


FIGURA 3: FUNCION CAPTURAR DATOS

Fuente: Pseint

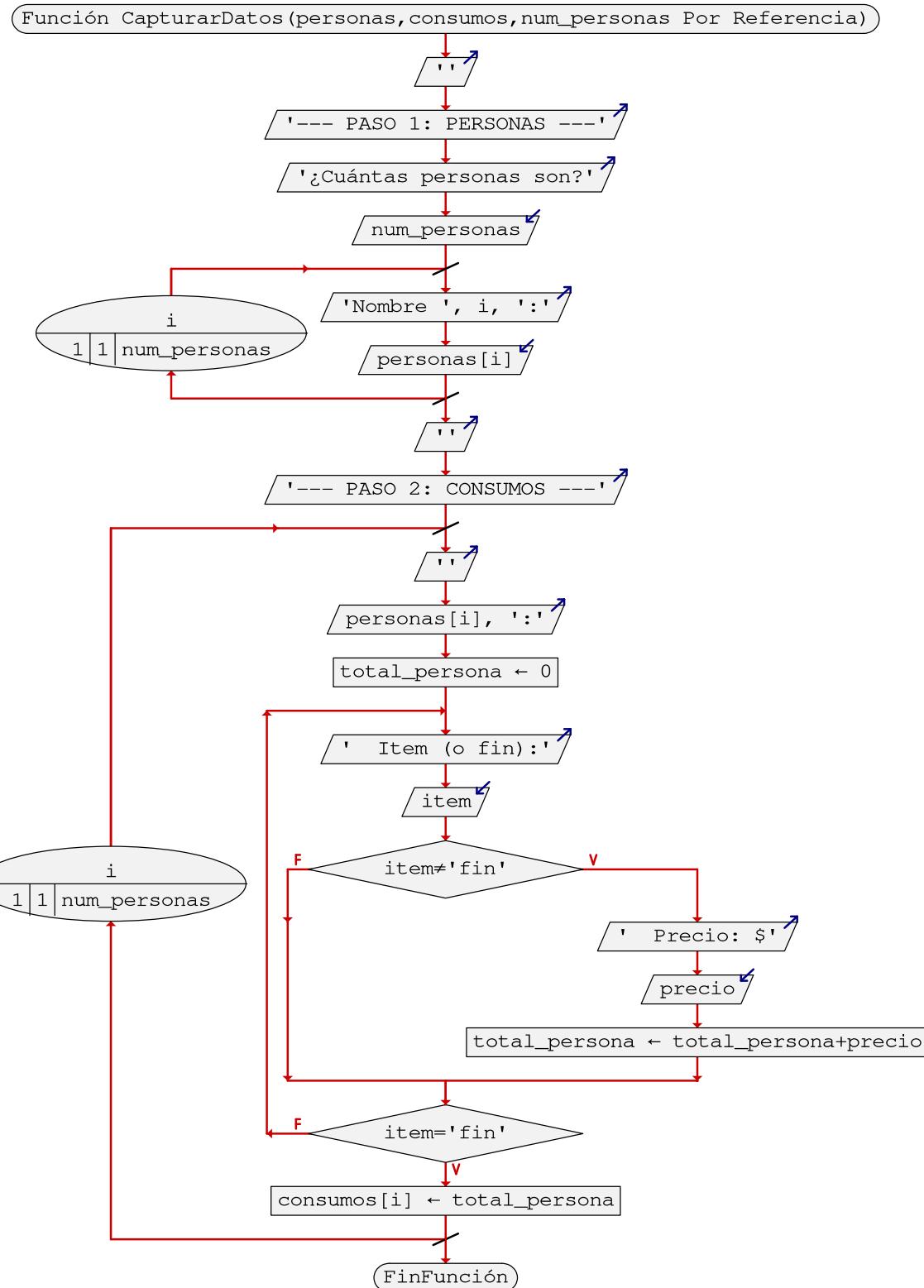


FIGURA 4: FUNCION CALCULAR PROPINAS POR PERSONA

Fuente: Pseint

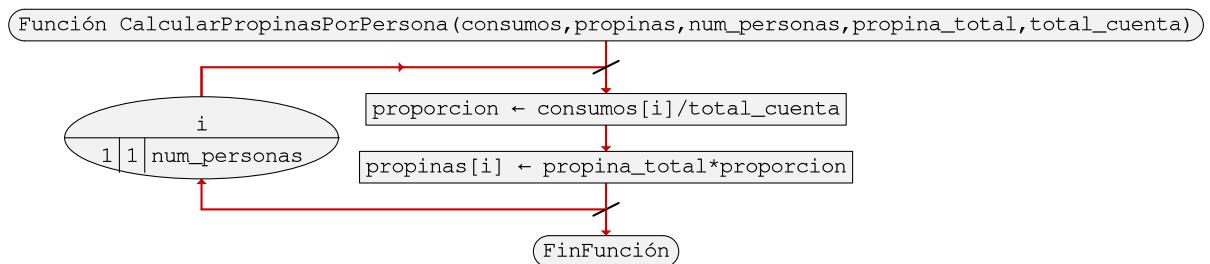


FIGURA 5: FUNCION PROPINA

Fuente: Pseint

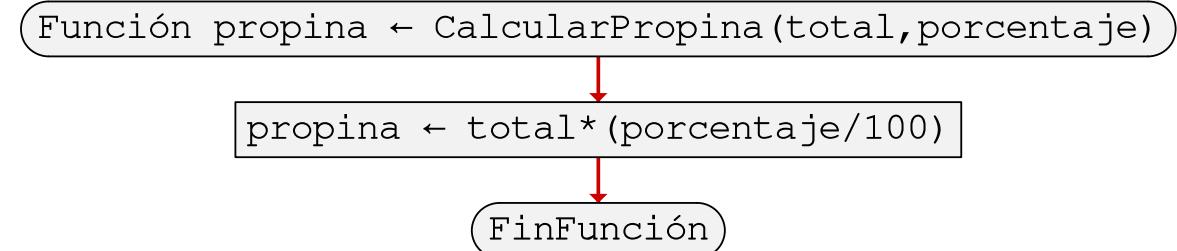


FIGURA 6: FUNCION TOTAL

Fuente: Pseint

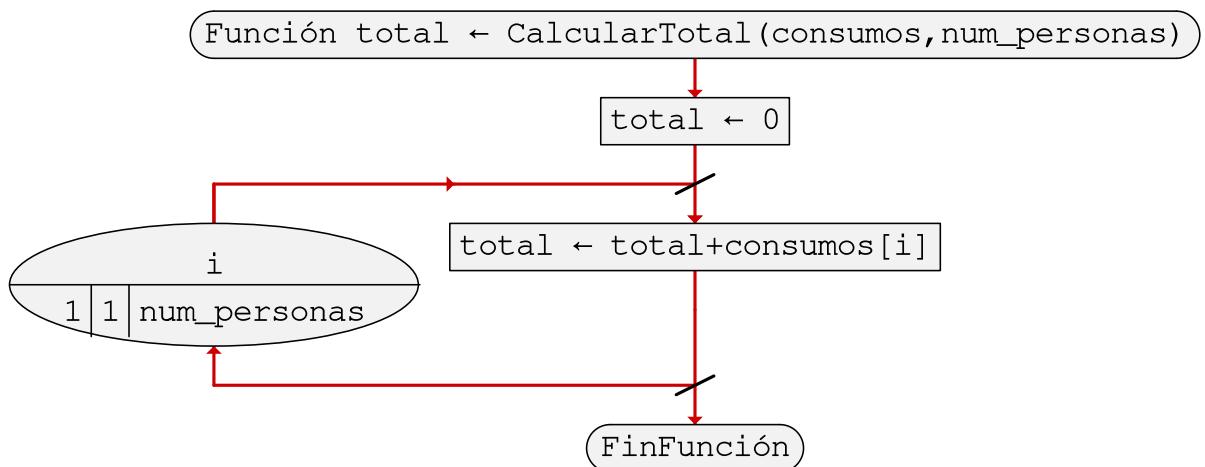


FIGURA 7: FUNCION CALCULAR TOTALES FINALES

Fuente: Pseint

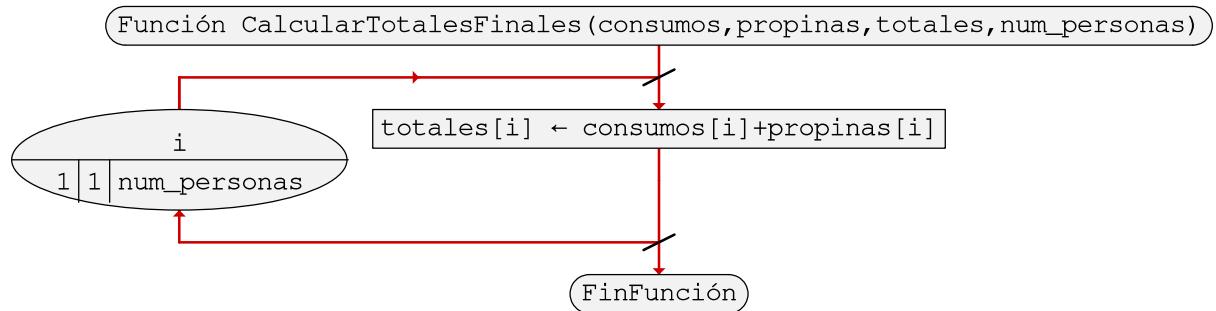


FIGURA 8: FUNCION MOSTRAR QUIEN PAGA

Fuente: Pseint

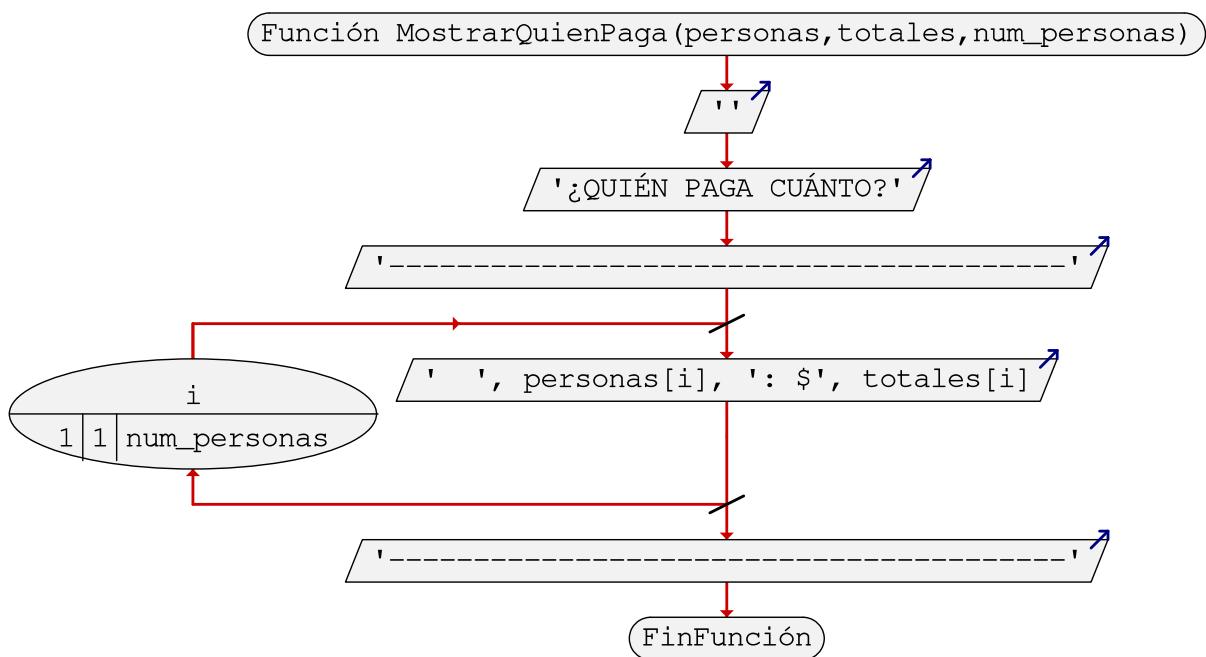
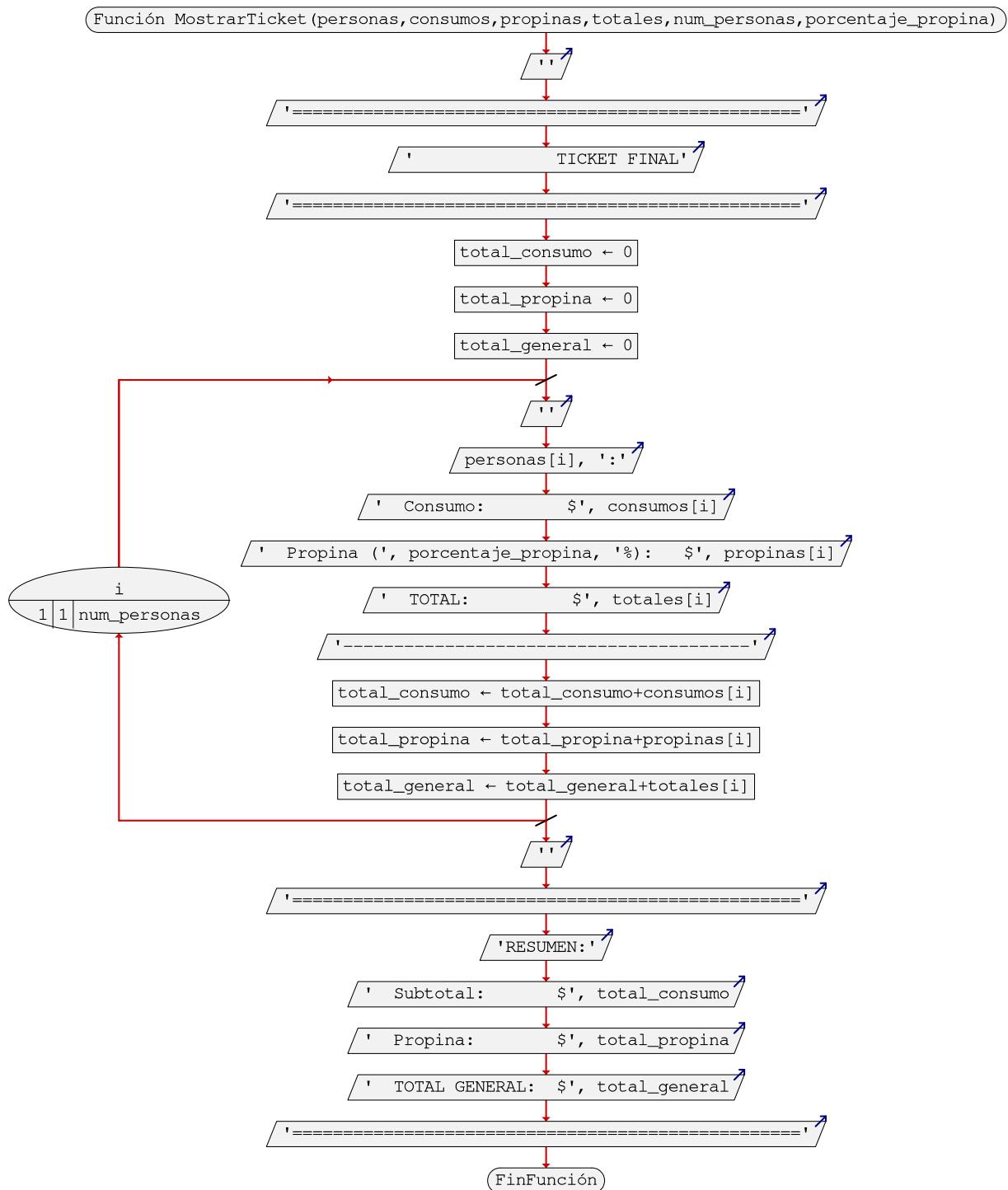


FIGURA 9: FUNCION MOSTRAR TICKET

Fuente: Pseint



ANEXO N° 1: ÁRBOL DE PROBLEMAS Y OBJETIVOS

PROBLEMA CENTRAL: Dificultad para calcular de manera precisa y equitativa la división de gastos compartidos entre grupos de personas.

CAUSAS: - Consumos individuales diferentes - Distribución compleja de propina proporcional - Errores en cálculos manuales - Tiempo excesivo en operaciones aritméticas - Falta de registro detallado

EFFECTOS: - Pagos inequitativos - Malentendidos entre participantes - Pérdida de tiempo - Posibles conflictos interpersonales - Incomodidad social

OBJETIVO CENTRAL: Desarrollar un programa automatizado que facilite el cálculo preciso y equitativo de división de gastos compartidos.

MEDIOS: - Interfaz de captura de datos - Algoritmos de cálculo precisos - Módulo de presentación clara - Estructura modular de código - Validación de casos de prueba

FINES: - Distribución justa de pagos - Transparencia en cálculos - Ahorro de tiempo - Mejora en relaciones interpersonales - Confianza en el proceso

ANEXO N° 2: CÓDIGO FUENTE COMPLETO DEL PROGRAMA

```
# Programa de División de Gastos entre Amigos
# Universidad Mayor de San Andrés
# INF-111 Programación I

# MÓDULO 1: INTERFAZ Y CAPTURA DE DATOS
def mostrar_menu():
    print("\n" + "="*40)
    print("  DIVISIÓN DE GASTOS")
    print("="*40)
    print("1. Dividir un gasto")
    print("2. Salir")

def capturar_datos():
    print("\n--- PASO 1: PERSONAS ---")
    num_personas = int(input("¿Cuántas personas son? "))
    personas = []

    for i in range(num_personas):
        nombre = input(f"Nombre {i+1}: ")
        personas.append(nombre)

    print("\n--- PASO 2: CONSUMOS ---")
    consumos = {}

    for persona in personas:
        print(f"\n{persona}:")
        total_persona = 0

        while True:
            item = input("  Item (o 'fin'): ")
            if item.lower() == 'fin':
                break
            precio = float(input("  Precio: $"))
            total_persona += precio

        consumos[persona] = total_persona

    return consumos

# MÓDULO 2: CÁLCULOS
def calcular_total(consumos):
    total = 0
    for persona in consumos:
        total += consumos[persona]
    return total
```

```

def calcular_propina(total, porcentaje):
    return total * (porcentaje / 100)

def calcular_propina_por_persona(consumos, propina_total, total_cuenta):
    propinas = {}

    for persona in consumos:
        proporcion = consumos[persona] / total_cuenta
        propinas[persona] = propina_total * proporcion

    return propinas

def calcular_totales_finales(consumos, propinas):
    totales = {}

    for persona in consumos:
        totales[persona] = consumos[persona] + propinas[persona]

    return totales

# MÓDULO 3: PRESENTACIÓN DE RESULTADOS
def mostrar_ticket(consumos, propinas, totales, porcentaje_propina):
    print("\n" + "="*50)
    print("".center(50, " ") + "TICKET FINAL")
    print("="*50)

    for persona in consumos:
        print(f"\n{persona}:")
        print(f" Consumo: ${consumos[persona]:.2f}")
        print(f" Propina ({porcentaje_propina} %): ${propinas[persona]:.2f}")
        print(f" TOTAL: ${totales[persona]:.2f}")
        print("-" * 40)

    total_consumo = sum(consumos.values())
    total_propina = sum(propinas.values())
    total_general = sum(totales.values())

    print("\n" + "="*50)
    print("RESUMEN:")
    print(f" Subtotal: ${total_consumo:.2f}")
    print(f" Propina: ${total_propina:.2f}")
    print(f" TOTAL GENERAL: ${total_general:.2f}")
    print("="*50)

def mostrar_quien_paga(totales):
    print("\n¿QUIÉN PAGA CUÁNTO?")
    print("-" * 40)
    for persona, total in totales.items():
        print(f" {persona}: ${total:.2f}")
    print("-" * 40)

# PROGRAMA PRINCIPAL

```

```

def main():
    while True:
        mostrar_menu()
        opcion = input("\nOpción: ")

        if opcion == "1":
            consumos = capturar_datos()
            total_cuenta = calcular_total(consumos)
            print(f"\nSubtotal: ${total_cuenta:.2f}")

            porcentaje = float(input("% de propina (ej: 10, 15): "))
            propina_total = calcular_propina(total_cuenta, porcentaje)
            propinas = calcular_propina_por_persona(consumos, propina_total,
total_cuenta)
            totales = calcular_totales_finales(consumos, propinas)

            mostrar_ticket(consumos, propinas, totales, porcentaje)
            mostrar_quien_paga(totales)

        elif opcion == "2":
            print("\n¡Adiós!")
            break
        else:
            print("\nOpción inválida")

if __name__ == "__main__":
    main()

```

ANEXO N° 3: MANUAL DE USUARIO

REQUISITOS DEL SISTEMA

Python 3.8 o superior instalado - Terminal o consola de comandos - Sistema operativo:
Windows, macOS o Linux

INSTALACIÓN

Guardar el código en un archivo llamado dividir_gastos.py - No requiere instalación de librerías adicionales

EJECUCIÓN DEL PROGRAMA

Abrir terminal o consola - Navegar a la carpeta donde se guardó el archivo - Ejecutar:
python dividir_gastos.py

USO DEL PROGRAMA

Paso 1: Seleccionar opción 1 del menú principal

Paso 2: Ingresar el número de personas que compartirán el gasto

Paso 3: Ingresar el nombre de cada persona

Paso 4: Para cada persona, ingresar los ítems consumidos y sus precios. Escribir “fin” cuando termine con cada persona

Paso 5: Ingresar el porcentaje de propina deseado

Paso 6: Revisar el ticket detallado con el desglose por persona

Paso 7: Consultar el resumen de quién paga cuánto

EJEMPLO DE USO

Entrada:

- 3 personas: Ana, Carlos, María
- Ana: Pizza Bs150, Refresco Bs30
- Carlos: Hamburguesa Bs120, Papas Bs40
- María: Ensalada Bs100
- Propina: 15%

Salida:

- Ana paga: Bs207.00
- Carlos paga: Bs184.00
- María paga: Bs115.00