**GUION**

**2.1 Identificación de Patrones**

*Cada miembro del equipo explica con claridad al menos un patrón y justifica su aplicación en la solución. Se demuestra un alto nivel de comprensión y dominio del concepto.*

Hola, en este proyecto aplicamos varios patrones del pensamiento computacional que fueron fundamentales para resolver el problema propuesto; El gestionar los diferentes presupuestos de marketing en algunas regiones del Mundo

Para esto primero, utilizamos estructuras condicionales como if, elif y else.

Estas nos permitieron darle “cierta” inteligencia al sistema, esto debido a la facilidad que estas estructuras nos brindan, ya que este tipo de estructuras permiten que el código/sistema sea más flexible y adaptable a diversas situaciones

Por ejemplo, cuando el usuario elige una opción del menú, usamos condicionales para que el programa ejecute solo lo que le corresponde a esa opción; y dado el caso que la opción no se encuentre disponible dentro del menú, el sistema arrojara: “Opción incorrecta, intente de nuevo.”

Dándole a entender al usuario que ha digitado el numero de un menú no existente, llevándolo de nuevo al menú de opciones para que pueda digitar el número del menú que desea usar.

También las usamos para clasificar el presupuesto total como “bajo”, “equilibrado” o “alto”, dependiendo de los valores ingresados

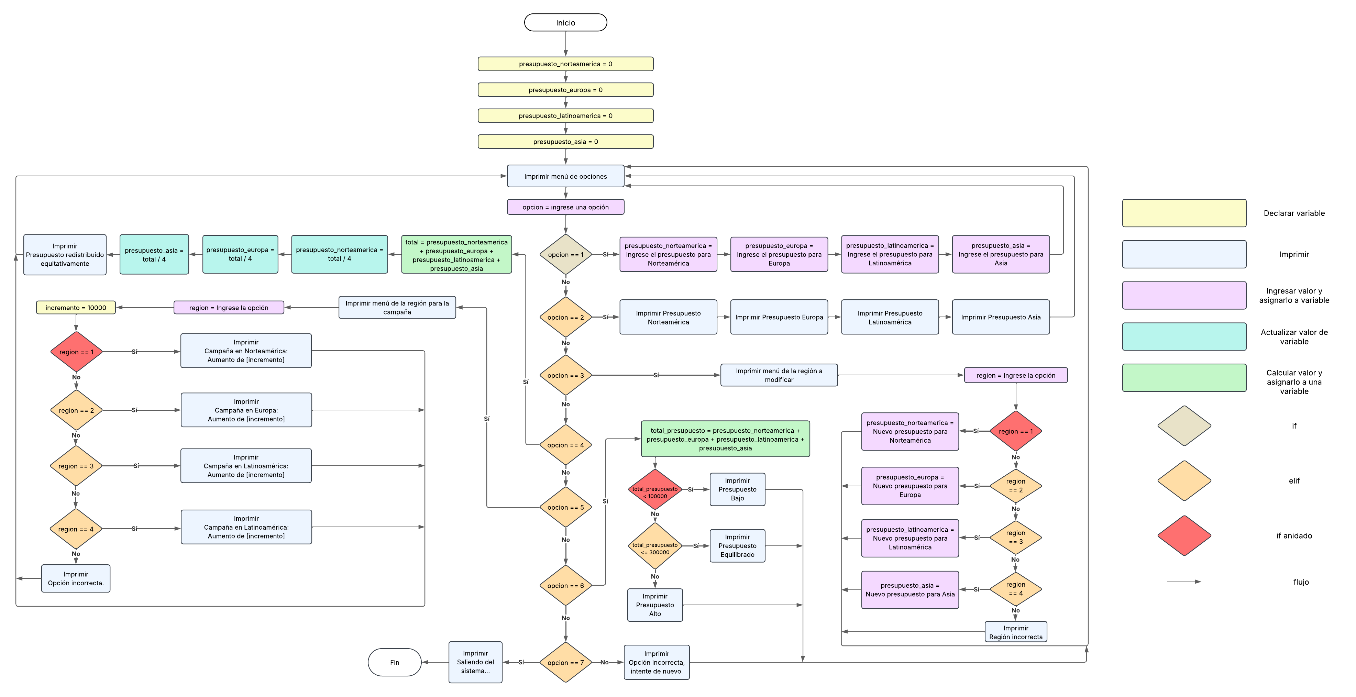
Además, usamos un ciclo while que permite que el menú del sistema se repita indefinidamente hasta que el usuario decida salir del sistema. De esta manera el usuario puede hacer varias acciones sin tener que reiniciar el programa cada vez que desear cambiar de menú, ya sea para ver los presupuestos, modificarlos, etc.

El uso de este ciclo nos permite brindarle al usuario una experiencia más agradable; gracias a los beneficios mencionados

También implementamos variables individuales para manejar el presupuesto de cada región: Norteamérica, Europa, Latinoamérica y Asia. Esto nos dio control directo sobre cada dato, y nos permitió hacer operaciones como modificar presupuestos por separado, redistribuirlos de forma equitativa, o clasificar su presupuesto

**2.2 Lógica y Algoritmia-Diagrama de Flujo**

*Cada miembro del equipo describe con precisión la lógica y la secuencia de dos opciones del sistema (excluyendo la opción de salir). La explicación es clara, estructurada y demuestra entendimiento del flujo del programa.*



Bueno, primeramente, al ejecutar el programa se despliega el menú de opciones donde el usuario podrá observar que se le permitirá hacer en cada uno de ellos.

Dado el caso que el usuario elija por ejemplo la opción de menú #1, este podrá Ingresar los diferentes presupuestos para cada una de las regiones, visto de la siguiente forma:

Menú de Opciones:

(1) Ingresar presupuesto inicial

(2) Mostrar presupuesto actual

…

(7) Salir

Ingrese una opción: 1 (Opción Ingresada por el Usuario)

Ingrese el presupuesto para Norteamérica: 100

Ingrese el presupuesto para Europa: 200

Ingrese el presupuesto para Latinoamérica: 300

Ingrese el presupuesto para Asia: 1000

Cuando el Usuario termine de Ingresar los presupuestos para cada una de las regiones este volverá al menú principal, esto debido al uso del Ciclo While, y podrá hacer uso del sistema hasta que lo desee.

Es de aclarar que después de ingresar un presupuesto podrá ingresar el presupuesto de otra región, evidenciado en el diagrama, dándole una estructura organizada al sistema.

**(VIDEO\_1)**

De la misma forma, al usuario elegir la opción #4 permitirá repartir el presupuesto total de manera equitativa entre las cuatro regiones: Norteamérica, Europa, Latinoamérica y Asia.

Cuando el usuario elige esta opción, el programa primero suma todo el dinero disponible en las cuatro regiones.

Luego, divide ese total entre cuatro, para que cada región reciba exactamente la misma cantidad.

Después de hacer ese cálculo, actualiza los valores de cada región con ese nuevo número.

Y por último, muestra un mensaje que dice que el presupuesto fue redistribuido exitosamente.

Y de la misma forma que la opción #1, será llevado otra vez al menú principal, así con las demás opciones, excepto con la #7 que es salir del programa.

De forma rápida, la opción #6 sirve para saber si el presupuesto total de las regiones es bajo, equilibrado o alto.

Cuando el usuario la selecciona, el programa primero suma todos los presupuestos de las regiones.

Con ese valor total, se evalúa en qué rango se encuentra:

Si es menor a 100.000, se considera que el presupuesto es bajo.

Si está entre 100.000 y 300.000, se muestra como equilibrado.

Y si supera los 300.000, el programa lo clasifica como un presupuesto alto.

Después de hacer la evaluación, el sistema muestra el resultado por pantalla y vuelve al menú principal.

Todo esto se repite de forma continua gracias al ciclo while, que permite que el usuario siga interactuando con el sistema sin tener que reiniciarlo.

**2.3 Diseño de la Interacción con el Sistema**

*Cada miembro del equipo presenta con claridad al menos dos ejemplos de interacción del usuario con el sistema, explicando cómo se navega y se reciben respuestas. La exposición facilita la comprensión del uso del sistema.*

En esta parte voy a mostrar cómo es la interacción entre el usuario y nuestro sistema. Vamos a ver dos ejemplos concretos que muestran cómo el usuario puede navegar fácilmente por el menú y cómo el sistema responde a esas acciones.

**Ejemplo 1: Ingresar presupuestos iniciales y consultarlos**

Cuando el usuario inicia el programa, se le presenta un menú con varias opciones. Supongamos que elige la opción (1) Ingresar presupuesto inicial.

El sistema le pide que ingrese los valores para cada región: Norteamérica, Europa, Latinoamérica y Asia. Por ejemplo:

Ingrese el presupuesto para Norteamérica: 50.000

Ingrese el presupuesto para Europa: 40.000

Ingrese el presupuesto para Latinoamérica: 30.000

Ingrese el presupuesto para Asia: 60.000

Una vez ingresados, el usuario puede volver al menú principal y elegir la opción (2) Mostrar presupuesto actual, y el sistema le muestra de forma clara el estado de cada región:

Presupuesto Norteamérica: 50.000

Presupuesto Europa: 40.000

Presupuesto Latinoamérica: 30.000

Presupuesto Asia: 60.000

Esto demuestra que el sistema es amigable y responde inmediatamente a las acciones del usuario, permitiendo visualizar el estado general con facilidad.

**(VIDEO\_2)**

**Ejemplo 2: Modificar una región y redistribuir**

Ahora veamos otro caso: el usuario elige la opción (3) Modificar presupuesto de una región.

El sistema le muestra una lista con las regiones disponibles. Supongamos que al principio se equivoca y pone un número fuera del rango, por ejemplo:

Seleccione la región: 5

Región incorrecta

Pero luego lo intenta de nuevo, selecciona la opción 1 (Norteamérica) y actualiza su presupuesto a 55000. El sistema lo registra y vuelve al menú.

Después de esto, el usuario puede elegir la opción (4) Redistribuir presupuesto equitativamente, y el sistema se encarga de hacer todos los cálculos automáticamente para repartir el dinero en partes iguales entre las cuatro regiones.

Por último, puede volver a consultar el estado del presupuesto con la opción (2), y verá que todas las regiones tienen el mismo valor, por ejemplo:

(partiendo del ejemplo anterior)

Presupuesto Norteamérica: 46.250

Presupuesto Europa: 46.250

Presupuesto Latinoamérica: 46.250

Presupuesto Asia: 46250

Estos ejemplos muestran que la interacción es simple, clara y efectiva. El usuario tiene siempre el control, y el sistema da respuestas inmediatas para facilitar la toma de decisiones.

Además, gracias al menú que se repite constantemente, no es necesario reiniciar el programa cada vez que se quiere hacer algo nuevo.

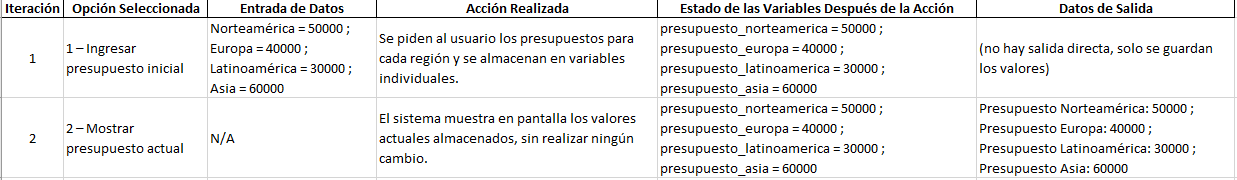
**(VIDEO\_3)**

**3.1 Prueba de Escritorio**

*Cada miembro del equipo presenta al menos dos pruebas de escritorio con valores de entrada y salida bien documentados. Explica con claridad cómo los resultados se relacionan con la lógica del sistema.*

En este punto voy a presentar dos pruebas de escritorio que nos permiten verificar que el sistema funciona correctamente. Las pruebas muestran paso a paso qué datos se ingresan, qué opción se elige, cómo cambian las variables internas del programa y cuáles son los datos de salida.

**Prueba de escritorio 1: Ingreso y Muestra de Presupuestos**



**Iteración #1**

Esta interacción valida que el sistema permite la captura inicial de datos correctamente.

Se relaciona directamente con el objetivo de inicializar las variables de forma manual y precisa a través de input(), tal como está definido en el bloque de código bajo if opcion == 1:.

Se cumple la lógica de entrada y almacenamiento sin procesar ni mostrar resultados aún.

**Iteración #2**

En esta etapa se comprueba que el sistema puede recuperar y mostrar los valores almacenados previamente.

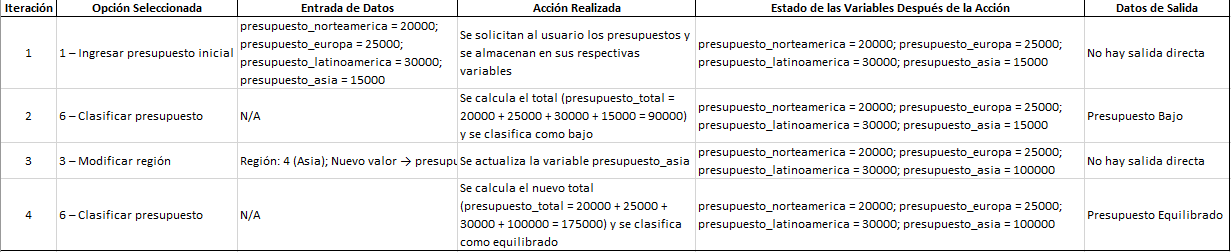
Corresponde a la condición if opcion == 2: del código, donde se accede a las variables ya definidas y se imprime su contenido.

No hay procesamiento adicional, solo lectura y salida de datos, lo cual valida que los valores ingresados fueron guardados correctamente.

Esto demuestra que la lógica de visualización del estado actual funciona como una herramienta de verificación para el usuario.

**(Prueba\_Escritorio\_1)**

**Prueba de escritorio 2: Clasificación del Presupuesto Total**



**Iteración #1**

Esta interacción valida que el sistema permite la captura inicial de datos correctamente.

Se relaciona directamente con el objetivo de inicializar las variables de forma manual y precisa a través de input(), tal como está definido en el bloque de código bajo if opcion == 1:.

Se cumple la lógica de entrada y almacenamiento sin procesar ni mostrar resultados aún.

**Iteración #2**

Aquí se aplica la lógica condicional definida en if opcion == 6: para evaluar el total del presupuesto ingresado.

El sistema calcula la suma de los valores y compara el resultado con los rangos establecidos para decidir si es un presupuesto Bajo, Equilibrado o Alto.

Dado que el total es 90000, se cumple la condición if total < 100000, y se muestra correctamente “Presupuesto Bajo”.

**Iteración #3**

Se comprueba la lógica de modificación de una sola región, que se encuentra bajo if opcion == 3:.

El sistema despliega un submenú, el usuario selecciona una región (Asia) y se le solicita un nuevo valor.

El nuevo valor reemplaza al anterior directamente, modificando solo la variable correspondiente sin afectar las demás.

Esta lógica permite ajustar presupuestos de forma puntual según las necesidades de la simulación.

**Iteración #4**

Una vez actualizado el valor de Asia, se vuelve a ejecutar la clasificación del presupuesto.

El nuevo total ahora es mayor, y se activa la segunda condición elif 100000 <= total\_presupuesto <= 300000, devolviendo “Presupuesto Equilibrado”.

Esto confirma que el sistema responde dinámicamente a los cambios y actualiza sus resultados con base en las nuevas condiciones.

**(Prueba\_Escritorio\_2)**

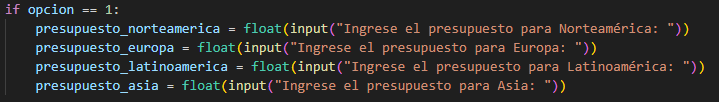
**3.2 Implementación Inicial en Python**

*Cada miembro del equipo explica con precisión el código correspondiente a una o dos opciones del sistema, detallando su funcionalidad y estructura. Se demuestra comprensión del lenguaje y buenas*

A continuación, voy a explicar con detalle dos de las opciones más representativas del sistema, mostrando cómo están programadas y cómo funcionan.

**Opción 1 – Ingresar presupuesto inicial**

Esta opción permite al usuario establecer los presupuestos de marketing para las cuatro regiones: Norteamérica, Europa, Latinoamérica y Asia. Es una parte fundamental porque da inicio al uso del sistema.

¿Qué está pasando aquí?

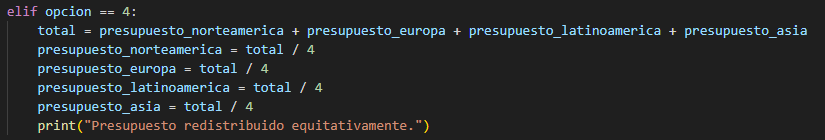
1. Se utiliza input() para pedir al usuario los datos.
2. Se convierten a números decimales con float() para permitir presupuestos con centavos.
3. Los valores ingresados se guardan en variables específicas para cada región.

Esto demuestra que entendemos cómo almacenar y manipular datos de entrada del usuario usando variables y tipos numéricos.

**(Opcion\_1)**

**Opción 4 – Redistribuir presupuesto equitativamente**

Esta opción toma el presupuesto total y lo reparte de forma igual entre todas las regiones, útil si se quiere estandarizar la inversión.



¿Qué hace esta parte del código?

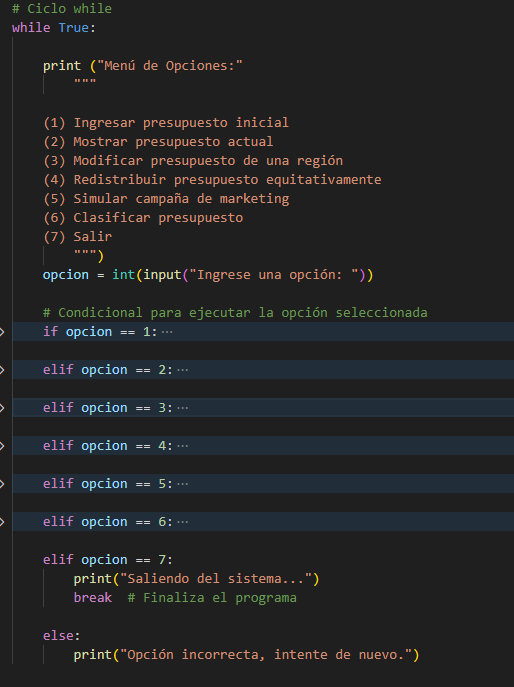
1. suma los presupuestos actuales de todas las regiones.
2. Luego divide el total en 4 partes iguales.
3. Finalmente, asigna ese valor a cada región para que todas tengan el mismo presupuesto.

Aquí demostramos el uso correcto de operaciones aritméticas, reutilización de variables y salida por pantalla con print().

**(Opcion\_4)**

**Estructura general**

Todo el código está contenido dentro de un while True, lo que permite que el sistema funcione de forma continua hasta que el usuario elija salir con la opción 7. Además, usamos condicionales if, elif y else para controlar el flujo del programa según la opción seleccionada.



**(Ciclo\_While)**