[[1]](#footnote-1)

Quiz-Reto 3(Octubre2025)

Juan Diego Hernández Castellanos [juanhernandez125@unisangil.edu.co](mailto:juanhernandez125@unisangil.edu.co) x

Sindy Carolina Pinilla Murcia Sindypinilla224@unisangil.edu.co

Abstract This report presents the development of an algorithm for analyzing a person's monthly expenses as part of the structured programming learning process. The system asks the user to enter the values ​​for each month of the year, with the option to adjust the number of months to analyze. The algorithm then performs calculations that include the total expenditure for the period, the monthly average, identifying the month with the highest and lowest expenditure, and the difference between the two. This provides a tool that allows the user to obtain a clear and summarized view of their personal finances. The project was developed following a problem-solving methodology, applying fundamental programming concepts such as control structures, accumulators, extreme value searches, and cycle management. The results obtained demonstrate the importance of programming as a support for everyday decision-making, in this case focused on the planning and control of financial resources.

Keywords: Algorithm, programing, Expenses, summary, Average, Analysis

*Resumen El presente informe expone el desarrollo de un algoritmo orientado al análisis de gastos mensuales de una persona, como parte del proceso de aprendizaje en programación estructurada. El sistema solicita al usuario el ingreso de los valores correspondientes a cada mes del año, con la posibilidad de ajustar el número de meses a analizar. Posteriormente, el algoritmo realiza cálculos que incluyen el gasto total del periodo, el promedio mensual, la identificación del mes con mayor y menor gasto, así como la diferencia entre ambos. De esta manera, se proporciona una herramienta que permite al usuario obtener una visión clara y resumida de sus finanzas personales. El proyecto se elaboró siguiendo la metodología de resolución de problemas, aplicando conceptos fundamentales de programación como estructuras de control, acumuladores, búsqueda de valores extremos y manejo de ciclos. Los resultados obtenidos evidencian la importancia de la programación como apoyo en la toma de decisiones cotidianas, en este caso enfocadas a la planeación y control de los recursos económicos*.

***Palabras clave:*** *Algoritmos, programación, gastos, promedio, análisis.*

Además, se emplearon herramientas como Git y GitHub para la gestión de versiones y el trabajo colaborativo, así como el entorno de desarrollo VS Code.

# I. Introducción

El control de los gastos personales es una práctica esencial para lograr una adecuada administración de los recursos financieros. Muchas veces, las personas no llevan un registro detallado de sus consumos mensuales, lo que dificulta la planeación y puede generar desequilibrios económicos. En este contexto, las herramientas computacionales y los algoritmos básicos de programación ofrecen una alternativa práctica para facilitar la organización y el análisis de información financiera.

El presente reto busca diseñar un algoritmo que permita al usuario ingresar sus gastos mensuales durante un periodo de tiempo determinado, calcular el gasto total, identificar los meses de mayor y menor consumo, obtener el promedio de gastos y determinar la diferencia entre valores extremos. De esta manera, el programa no solo cumple una función académica al aplicar la metodología de resolución de problemas en programación, sino que también demuestra cómo los conceptos aprendidos pueden aportar soluciones a necesidades reales de la vida cotidiana.

# II. Análisis del problema

El problema planteado consiste en diseñar un algoritmo que permita a una persona analizar sus gastos mensuales a lo largo de un periodo de tiempo determinado, generalmente un año. La situación surge de la necesidad de organizar y comprender de manera clara cómo se distribuyen los recursos económicos, ya que la falta de control sobre los gastos puede dificultar la planeación financiera.

Para resolver el problema, es necesario identificar los siguientes elementos:

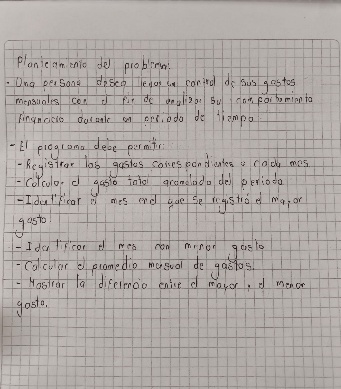
* Entradas: valores de los gastos correspondientes a cada mes. El número de meses puede ser fijo (12) o variable según la necesidad del usuario.
* Procesos:
  1. Acumular los gastos totales.
  2. Calcular el promedio mensual.
  3. Identificar el mes con mayor gasto.
  4. Identificar el mes con menor gasto.
  5. Calcular la diferencia entre el gasto máximo y el mínimo.
* Salidas:
  1. Total de gastos del periodo.
  2. Promedio de gastos mensuales.
  3. Mes con mayor gasto.
  4. Mes con menor gasto.
  5. Diferencia entre el gasto máximo y mínimo.

De esta manera, el problema se centra en aplicar operaciones básicas de suma, comparación y promedio dentro de un ciclo que permita recorrer los valores ingresados por el usuario. Además, requiere estructuras de decisión para identificar los gastos extremos y presentar los resultados de manera clara.

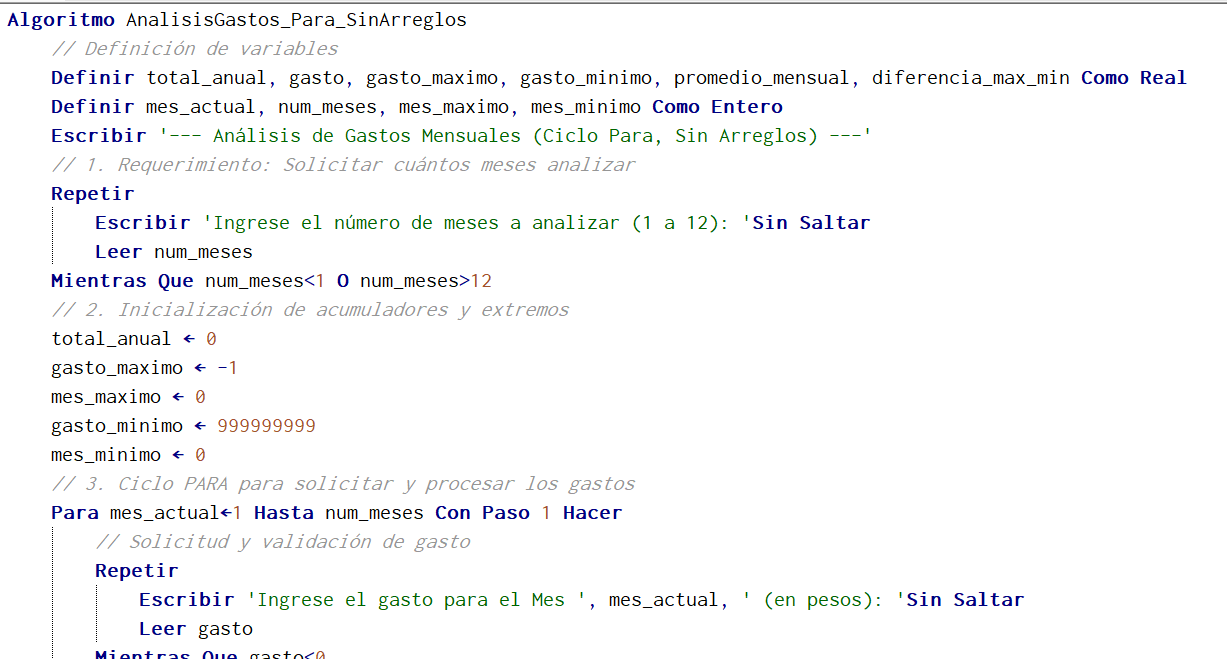
# III. metodología

El desarrollo del reto se realizó siguiendo la metodología de **resolución de problemas en programación**, apoyada en el uso de herramientas de desarrollo y control de versiones. El proceso se llevó a cabo en las siguientes etapas:

1. **Planteamiento del problema:** Se definieron los objetivos del programa, consistentes en registrar los gastos mensuales, calcular el total anual, identificar los meses de mayor y menor gasto, obtener el promedio mensual y determinar la diferencia entre los gastos extremos.

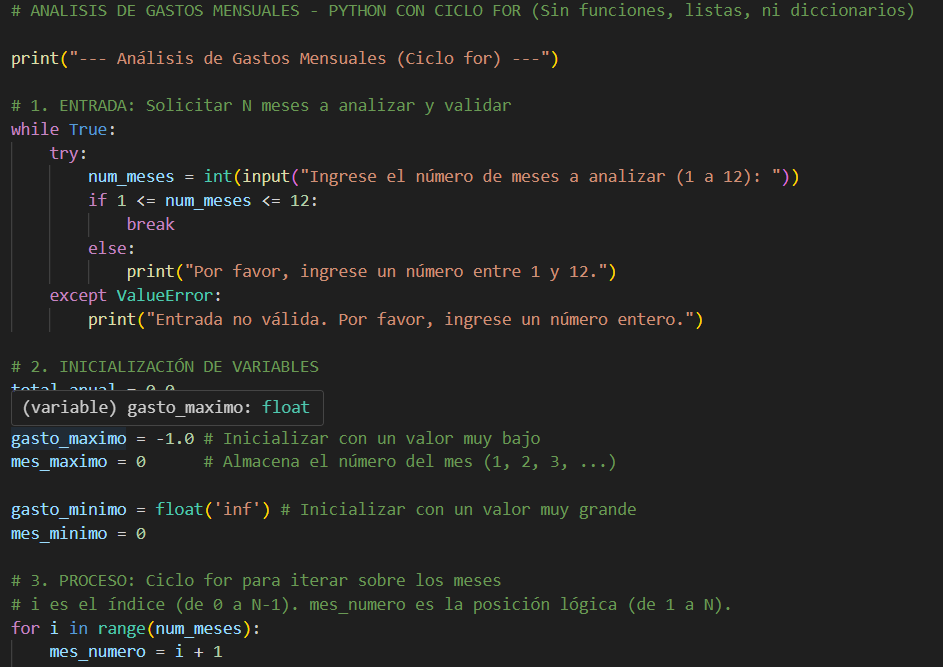
****

1. **Diseño del algoritmo en PSeInt:** Como primera fase de la solución, se elaboró un pseudocódigo en la herramienta **PSeInt**, con el fin de representar la lógica del programa de manera estructurada antes de pasar a la codificación.

*Evidencia 1: Captura del pseudocódigo en PSeInt.*

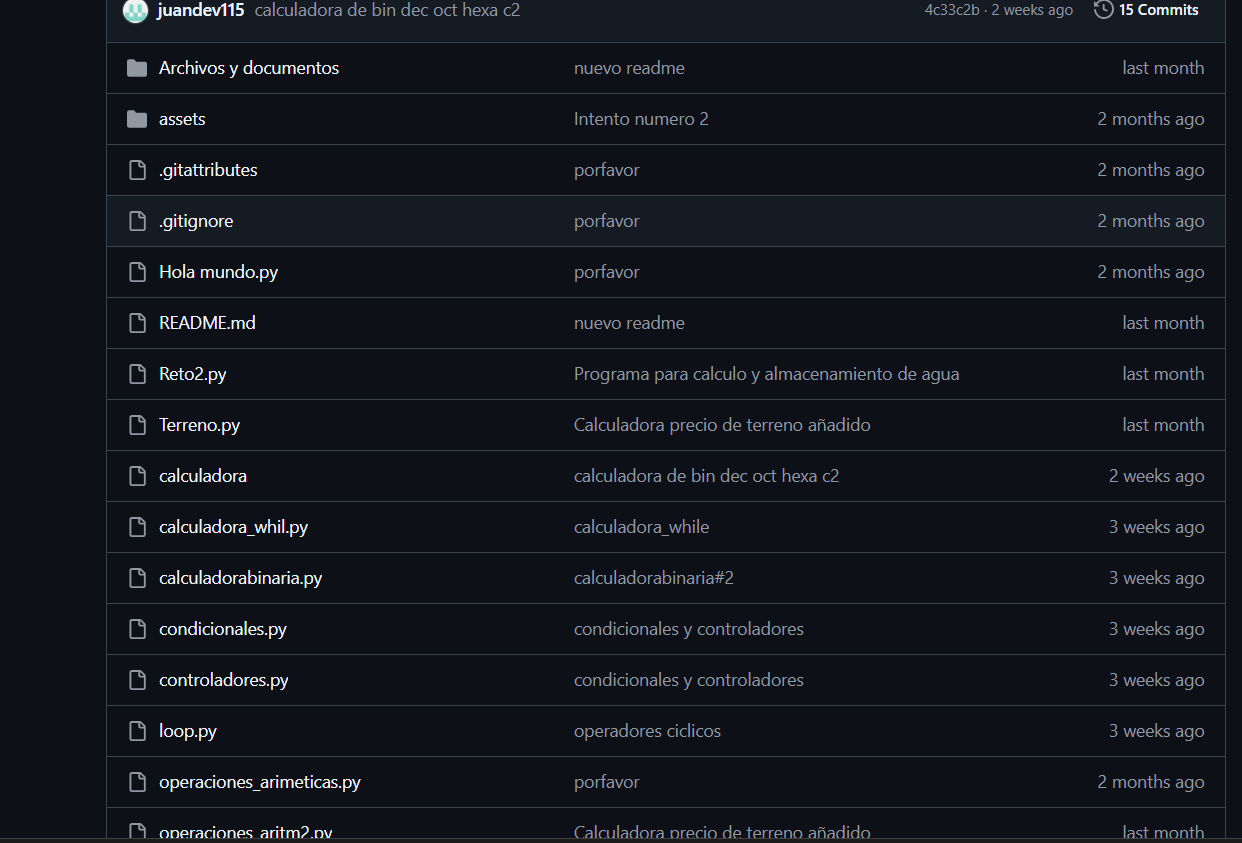
1. **Implementación en Python:** Posteriormente, se codificó el algoritmo en **Python**, utilizando el entorno de desarrollo **Visual Studio Code (VS Code)**. El programa hace uso de ciclos para la captura de los datos, estructuras condicionales para la identificación de los valores extremos y acumuladores para calcular el gasto total y el promedio.

*Evidencia 2: Fragmento del código en Python (VS Code).*



1. **Uso de Git y GitHub:** Para el control de versiones y almacenamiento del proyecto, se utilizó **Git** como herramienta de control local y **GitHub** como repositorio remoto. Esto permitió llevar un registro organizado de los avances, respaldar la información y compartir los resultados con el docente a través del repositorio creado.

*Evidencia 3: Captura del repositorio en GitHub mostrando la carpeta del reto.*



1. **Pruebas y validación:** Finalmente, se realizaron pruebas de ejecución con diferentes valores de entrada, con el fin de comprobar la precisión de los cálculos y la correcta identificación de los gastos máximos y mínimos.

# IV. Resultados

Durante la ejecución del programa se ingresaron 4 meses de gastos con los siguientes valores:

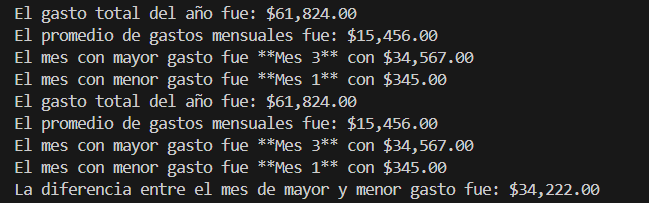
* Mes 1: $345
* Mes 2: $23,456
* Mes 3: $34,567
* Mes 4: $3,456

A partir de estos datos, el algoritmo generó los siguientes resultados:

* 1. Número de meses analizados: 4
  2. Gasto total del año: $61,824.00
  3. Promedio de gastos mensuales: $15,456.00
  4. Mes con mayor gasto: Mes 3 con $34,567.00
  5. Mes con menor gasto: Mes 1 con $345.00

Diferencia entre el gasto mayor y menor: $34,222.00

Estos resultados evidencian el correcto funcionamiento del algoritmo, mostrando cálculos precisos y una adecuada identificación de los valores extremos.

*Evidencia 4: Captura de la consola en Visual Studio Code *

# V. evaluación pruebas y despliegue

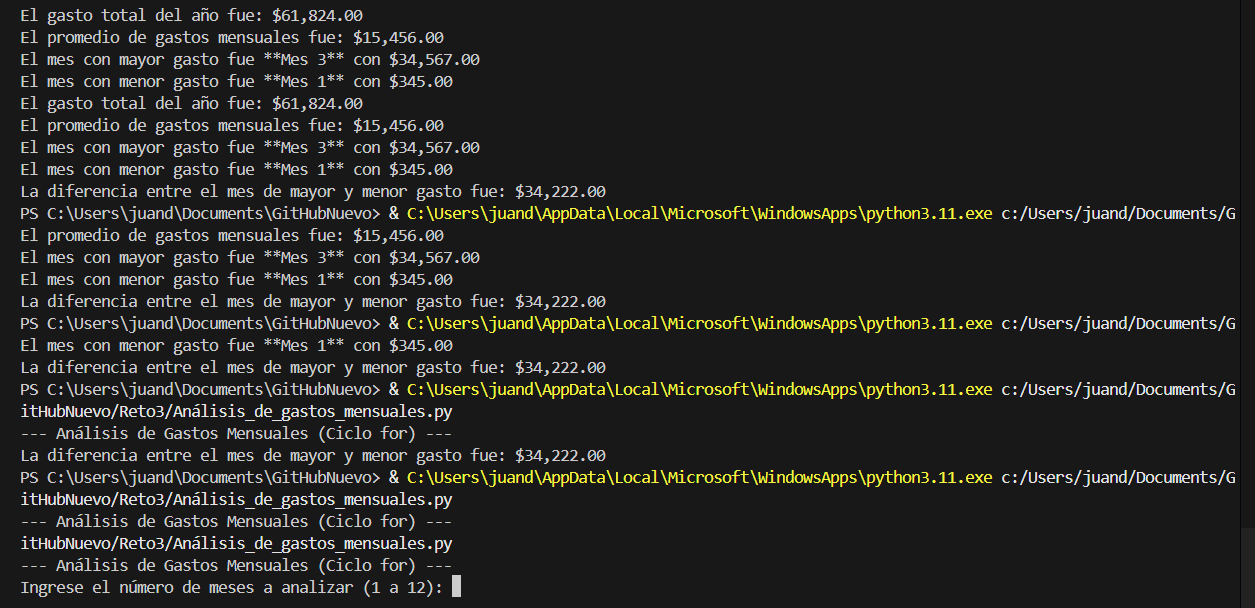
Con el fin de validar la solución propuesta, se realizaron pruebas de funcionamiento del algoritmo utilizando diferentes cantidades de meses y valores de gasto. Estas pruebas permitieron comprobar la correcta ejecución de los cálculos, así como la precisión en la identificación de los meses con mayor y menor gasto.

1. **Evaluación:**  
   Se evaluó el programa verificando que cumpliera con los requerimientos planteados:
   * Registro dinámico del número de meses a analizar.
   * Acumulación correcta del gasto total.
   * Identificación adecuada de los gastos máximos y mínimos.
   * Cálculo del promedio mensual y la diferencia entre valores extremos.
2. **Pruebas:**  
   Se realizaron pruebas con datos pequeños y grandes para garantizar la estabilidad del algoritmo. En todos los casos, los resultados obtenidos coincidieron con los cálculos esperados, lo que evidencia la confiabilidad de la solución.

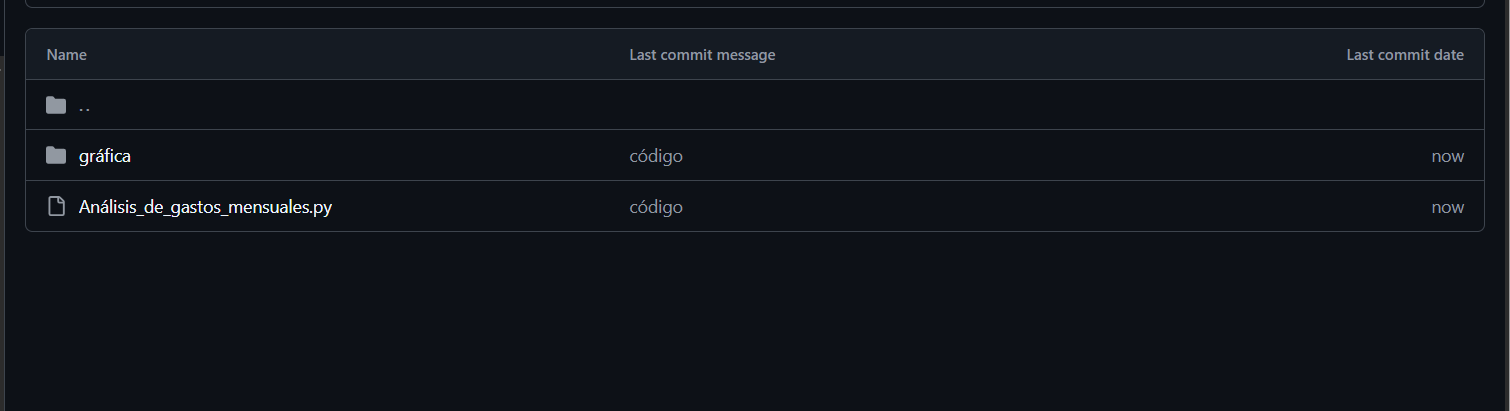
*Ejemplo de prueba:*

* + Datos de entrada: 4 meses con valores [345, 23,456, 34,567, 3,456].
  + Resultados esperados: Total = $61,824, Promedio = $15,456, Máximo = $34,567, Mínimo = $345, Diferencia = $34,222.
  + Resultados obtenidos: coinciden con lo esperado.

1. **Despliegue:**  
   El programa fue implementado en **Python** utilizando **Visual Studio Code** como entorno de desarrollo. Para el control de versiones y respaldo, se empleó **Git** de manera local y **GitHub** como repositorio remoto, lo que permitió almacenar el proyecto y dejar evidencia del proceso de desarrollo.
   * *Evidencia 1:* Captura de la ejecución del programa en consola.



* + *Evidencia 2:* Captura del repositorio en GitHub mostrando la carpeta del reto.



En conclusión, las pruebas y el despliegue demostraron que el programa cumple satisfactoriamente con los objetivos planteados, mostrando resultados confiables y listos para ser utilizados como apoyo en el análisis de gastos personales.

.

# VI. Conclusiones

El desarrollo de este reto permitió aplicar de manera práctica los conceptos básicos de programación orientados a la solución de problemas cotidianos. A través del diseño y la implementación del algoritmo en Python, fue posible comprobar la utilidad de las estructuras de control, ciclos, acumuladores y condicionales en la construcción de una solución eficiente y clara.

El programa cumplió satisfactoriamente con los objetivos planteados: calcular el gasto total, identificar los meses de mayor y menor gasto, determinar el promedio mensual y establecer la diferencia entre los valores extremos. Los resultados obtenidos validan la correcta lógica del algoritmo y evidencian la importancia de planear de manera ordenada el desarrollo de un programa antes de su codificación.

Asimismo, el uso de herramientas como PSeInt para la creación del pseudocódigo, Visual Studio Code para la codificación en Python, y GitHub para el control de versiones y despliegue del proyecto, reforzó habilidades técnicas fundamentales para la formación en ingeniería de sistemas.

Finalmente, se concluye que el proyecto no solo aportó al aprendizaje de programación estructurada, sino que también demostró cómo la informática puede ofrecer soluciones simples y efectivas para la administración financiera personal, sirviendo como base para el desarrollo de aplicaciones más completas en el futuro.

1. Documento enviado para revisión el de octubre de 2025.Fue desarrollado con los conocimientos y recursos brindados por la Unisangil Chiquinquirá [↑](#footnote-ref-1)