[[1]](#footnote-1)

Cálculo del Consumo y almacenamiento de agua en un edificio

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

Sindy Carolina Pinilla Murcia Sindypinilla224@unisangil.edu.co

**Resumen — Este informe de laboratorio presenta el desarrollo de cuatro ejercicios de programación enfocados en la aplicación de metodologías de resolución de problemas y el pensamiento computacional mediante el lenguaje Python. Los problemas propuestos simulan escenarios reales en áreas como la gestión de proyectos, el diseño de productos, los sistemas de control de acceso seguro y las políticas de descuentos en comercio electrónico. Cada ejercicio requirió el diseño de un algoritmo, la creación de diagramas de flujo en PSeInt y la implementación de programas ejecutables en Python. Adicionalmente, se utilizaron Git y GitHub como herramientas colaborativas para la gestión del código fuente y el control de versiones, fomentando el trabajo en equipo y las buenas prácticas de la ingeniería de software. Los resultados evidencian la correcta aplicación de estructuras condicionales, operadores lógicos y de comparación, fórmulas matemáticas y principios de diseño algorítmico. Este trabajo constituye una introducción a la resolución de problemas mediante la programación, reforzando las competencias fundamentales necesarias para cursos posteriores en desarrollo de software.**

**Palabras clave — Python, algoritmos, programación, formato IEEE, resolución de problemas, GitHub, diagramas de flujo.**

# introducción

La programación constituye una de las competencias fundamentales en la formación de un ingeniero en sistemas, ya que permite desarrollar soluciones eficientes a problemas reales mediante el uso de algoritmos y estructuras computacionales. El presente laboratorio se diseñó con el propósito de aplicar la metodología de resolución de problemas en escenarios prácticos, empleando herramientas como PSeInt para el diseño de diagramas de flujo, Python para la implementación de algoritmos y GitHub como plataforma de control de versiones y trabajo colaborativo.

El laboratorio está compuesto por cuatro ejercicios que integran distintos conceptos básicos de programación. El primero se centra en el cálculo de retrasos en proyectos de software, aplicando fórmulas matemáticas y operadores aritméticos. El segundo aborda la determinación del volumen de una esfera, con el fin de simular un caso en el área de manufactura y diseño de productos. El tercero desarrolla un sistema de control de acceso seguro, donde se ponen en práctica operadores lógicos y estructuras condicionales. Finalmente, el cuarto ejercicio presenta un caso de comercio electrónico en el que se calculan descuentos acumulativos según diversas condiciones, aplicando operadores de comparación y condicionales múltiples.

Este informe, elaborado bajo el formato IEEE, expone la descripción de los problemas planteados, el análisis de cada situación, los diagramas de flujo correspondientes, la implementación en Python y la verificación de resultados. De esta forma, se busca reforzar el pensamiento lógico, la capacidad de abstracción y el uso de herramientas básicas de programación como fundamentos para el desarrollo de software en contextos más complejos.

# análisis del problema(ejercicio a)

En la gestión de proyectos de desarrollo de software, cada miembro del equipo recibe tareas con un tiempo límite previamente asignado. Cuando una tarea se completa fuera de plazo, es necesario calcular el retraso en días y el porcentaje de retraso con respecto al tiempo total asignado. Este cálculo permite a los gerentes de proyecto evaluar el rendimiento del equipo, identificar riesgos y tomar decisiones para mejorar la planificación.

Para resolver este problema, se requiere un algoritmo que reciba como entrada el número de días asignados y el número de días reales empleados en completar la tarea. Con base en estos datos, se determina el retraso en días a partir de la diferencia entre el tiempo real y el tiempo asignado. Posteriormente, se calcula el porcentaje de retraso dividiendo el retraso en días entre el tiempo asignado, y multiplicando el resultado por 100.

El resultado del algoritmo debe indicar claramente si hubo o no retraso. En caso afirmativo, se mostrarán los días de retraso y el porcentaje correspondiente. Si el tiempo real empleado no supera el asignado, se debe indicar que la tarea fue completada en el plazo previsto.

Este análisis constituye la base para la construcción del diagrama de flujo en PSeInt y la posterior implementación en Python.

## Archivos electrónicos de Imagen

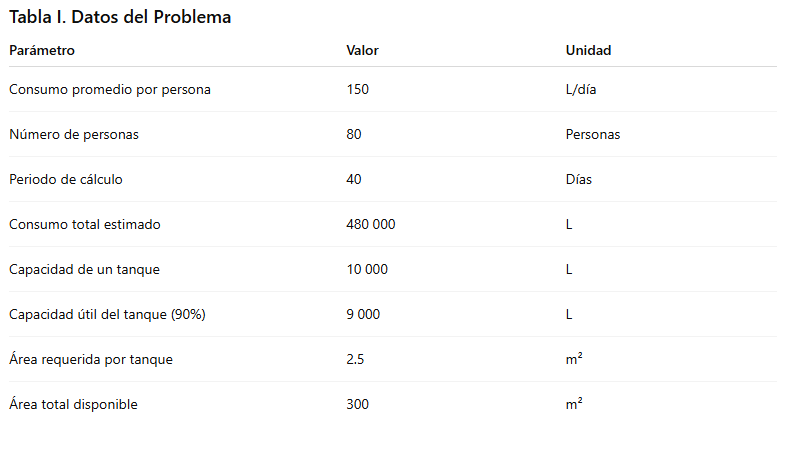


Fig1:se demuestran los datos obtenidos y necesarios para ejecutar la solucion del problema

# metodología

La metodología aplicada en este trabajo se divide en dos etapas principales:

1. **Cálculo manual**: se establecieron los requerimientos de consumo de agua y se compararon con la capacidad de los tanques, considerando el área física disponible. Este cálculo proporciona una primera aproximación a la cantidad mínima de tanques necesarios.
2. **Cálculo computacional**: se diseñó un programa en Python empleando operadores aritméticos, de comparación y lógicos. El propósito de este programa es verificar automáticamente los resultados manuales y permitir su aplicación a escenarios distintos (variando el número de personas, capacidad de tanques, eficiencia, entre otros parámetros).

#### A. Procedimiento de cálculo manual

El consumo total se obtiene a partir de la expresión:

CT​=Cp​×N×D

donde:

* CT​ = consumo total (L),
* Cp = consumo por persona (L/día),
* N = número de personas,
* D = número de días.

Luego, el número de tanques necesarios se determina como:

T=

donde:

* Cut = capacidad útil del tanque (90% de 10 000 L).

Finalmente, se valida la restricción espacial mediante:

AT​=T×At​≤Ad​

donde:

ATA\_TAT​ = área total requerida,

AtA\_tAt​ = área por tanque (m²),

AdA\_dAd​ = área disponible (m²).

B. Desarrollo del Programa en Python

Para confirmar los cálculos, se desarrolló un programa en Python con estructuras simples. El código aplica operadores aritméticos (para el cálculo de consumos), operadores de comparación (para determinar la cantidad mínima de tanques) y operadores lógicos (para verificar la disponibilidad de espacio).

A continuación, se muestra un fragmento del código implementado: 

Este programa garantiza resultados consistentes con los cálculos manuales y permite realizar ajustes de forma inmediata al modificar los valores de entrada.

# Resultados

El consumo diario se calcula multiplicando el consumo promedio por persona y el número de habitantes:

Qd= 150 X 80 Personas = 12000 L/día

El consumo para 40 días es:

Q40= Qd × 40 = 12 000 L/día × 40 días = 480 000 L

La capacidad útil de cada tanque, considerando una eficiencia del 90%, es:

Ctanque = 10 000 L × 0.90 = 9 000 L

El número de tanques necesarios se obtiene dividiendo el consumo total entre la capacidad útil:

Ntanques = = 53.3 ≈ 54 tanques

Finalmente, el área total requerida para la instalación es:

Atotal = 54 × 2.5 m² = 135 m² (5)

Dado que el edificio dispone de 300 m², el espacio es suficiente para la instalación.

# Discusión

El estudio demuestra que el consumo de agua para 80 personas durante 40 días requiere de 480 000 litros en total. Con tanques de 10 000 litros de capacidad nominal, pero considerando una eficiencia del 90 %, la instalación de 54 tanques asegura la cobertura del consumo total.

El área de instalación es un factor crítico en proyectos de infraestructura. En este caso, los 135 m² necesarios para los tanques no superan el límite de los 300 m² disponibles, lo que garantiza la viabilidad del sistema.

El programa en Python se consolidó como una herramienta confiable, ya que automatizó los cálculos y permitió validar rápidamente la factibilidad del proyecto. Además, el modelo es flexible: modificando los valores iniciales, puede aplicarse a otros escenarios de consumo o infraestructura.

# Conclusiones

* Se confirmó que el número de tanques necesarios para cubrir el consumo de 80 personas durante 40 días es de **54 tanques** de 10 000 litros, con un 90 % de capacidad útil.
* La restricción de espacio no representa un problema, dado que el área requerida (135 m²) es menor al área disponible (300 m²).
* El desarrollo del programa en Python permitió verificar los cálculos de forma precisa y rápida, mostrando la utilidad de la programación en la resolución de problemas cotidianos.
* La metodología empleada combina adecuadamente las matemáticas aplicadas y la computación, lo cual constituye un enfoque interdisciplinario en la formación de ingenieros en sistemas.

Reconocimiento

Instituto Tecnológico de Costa Rica

Referencias

[1] Organización de las Naciones Unidas, “Informe mundial sobre el desarrollo de los recursos hídricos 2023,” UNESCO, París, 2023.

[2] World Health Organization, “Domestic water quantity, service level and health,” Geneva, 2003.

[3] J. Smith and R. Brown, “Water storage systems in residential buildings,” IEEE Transactions on Sustainable Energy, vol. 12, no. 4, pp. 2301–2308, 2021.

1. Documento enviado para revisión el 010 de septiembre de 2025.Fue desarrollado con los conocimientos y recursos brindados por la Unisangil Chiquinquirá [↑](#footnote-ref-1)