Laboratorio1(Septiembre 2025)

Juan Diego Hernández Castellanos juanhernandez125@unisangil.edu.co
Sindy Carolina Pinilla Murcia Sindypinilla224@unisangil.edu.co

Abstract

This laboratory aims to apply the problem-solving methodology in the development of basic algorithms using programming tools. Through practical exercises, the objective is to strengthen skills in designing solutions, implementing them in Python, and representing them with flowcharts in PSeInt.

The work addresses real-world situations such as calculating delays in software projects, determining the volume of a sphere for manufacturing purposes, simulating an access control system in high-security buildings, and applying discount policies in online shopping. For each case, algorithms were designed to clearly and systematically represent the proposed solution.

Additionally, tools such as Git and GitHub were used for version control and collaborative work, as well as Visual Studio Code as the development environment. The laboratory strengthened the students' programming knowledge, promoted teamwork, and encouraged the application of good practices in software development.

Keywords
Problem-solving, algorithms, Python, PSeInt, Git, GitHub, software development.

Resumen El presente laboratorio tiene como objetivo aplicar la metodología de resolución de problemas en el desarrollo de algoritmos básicos utilizando herramientas de programación. A través de diferentes ejercicios prácticos, se busca fortalecer las competencias en el diseño *de* soluciones, la implementación en Python y la representación mediante diagramas de flujo en PSeInt.

El trabajo aborda situaciones reales como el cálculo de retrasos en proyectos de software, la determinación del volumen de una esfera para fines de manufactura, la simulación de un sistema de control de acceso en edificios de alta seguridad y la aplicación de políticas de descuentos en compras en línea. Para cada uno de los casos, se plantearon algoritmos que permiten representar de manera clara y estructurada la solución al problema.

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. El laboratorio permitió afianzar los conocimientos adquiridos en programación, fomentar el trabajo en equipo y aplicar buenas prácticas en el desarrollo de software.
I. Palabras claves: Resolución de problemas, algoritmos, Python, Pseint , Git , GitHub, Desarrollo de Software.

I. Introducción

El presente laboratorio tiene como finalidad poner en práctica la metodología de resolución de problemas mediante el diseño de algoritmos, su representación en diagramas de flujo y su implementación en Python. Asimismo, se hace uso de herramientas complementarias como PSeInt, para la construcción gráfica de algoritmos; Git y GitHub, para el control de versiones y el trabajo colaborativo; y Visual Studio Code como entorno de desarrollo.

A través de la resolución de cuatro problemas de aplicación, se busca que los estudiantes adquieran competencias en la estructuración lógica, el uso de operadores de comparación y lógicos, y la aplicación de estructuras condicionales. De esta manera, el laboratorio contribuye al fortalecimiento de habilidades de programación orientadas a la solución de problemas cotidianos y al desarrollo de destrezas para el trabajo en equipo con buenas prácticas de software.

II. Análisis del problema

Para la resolución de los ejercicios planteados en el laboratorio, se aplicó la técnica de descomposición en entradas, procesos y salidas, con el fin de identificar de manera clara los componentes de cada algoritmo.

Ejercicio A. Cálculo de retraso en proyectos de software

1. Entradas: Fecha límite de la tarea, fecha de finalización real.
2. Proceso: Calcular los días de retraso (diferencia entre fechas) y el porcentaje de retraso respecto al tiempo total asignado.

Documento enviado para revisión el 22 de septiembre de 2025.Fue desarrollado con los conocimientos y recursos brindados por la Unisangil Chiquinquirá

Salida: Número de días de retraso y porcentaje de incumplimiento.

Ejercicio B. Cálculo del volumen de una esfera

1. Entradas: Radio de la esfera.
2. Proceso: Aplicar la fórmula matemática

$$V = \frac{4}{3}\pi r^3$$

Salida: Volumen de la esfera en unidades cúbicas.

Ejercicio C. Sistema de control de acceso

1. Entradas: Nivel de acceso del usuario (0-5), estado de la tarjeta (activa/inactiva), validez del cambio de contraseña (días).
2. Proceso: Verificar si el nivel de acceso es suficiente, si la tarjeta está activa y si la contraseña se cambió en los últimos 30 días.

Salida: Mensaje de "Acceso permitido" o "Acceso denegado".

Ejercicio D. Sistema de descuentos en tienda en línea

1. Entradas: Monto total de la compra, condición de cliente VIP, existencia de código de descuento.
2. Proceso: Calcular descuentos acumulativos (20% si la compra supera $100, 10% si es VIP y 5% adicional si hay código).

Salida: Total a pagar después de aplicar los descuentos y porcentaje total de descuento aplicado.

A. *Análisis de los problemas*

Ejercicio	Entradas	Proceso	Salidas
A. Retraso en proyectos de software	Fecha límite, fecha real de entrega	Calcular diferencia de días y porcentaje de retraso respecto al tiempo asignado	Días de retraso, porcentaje de incumplimiento
B. Volumen de una esfera	Radio de la esfera	Aplicar fórmula $V = \frac{4}{3}\pi r^3$	Volumen de la esfera en unidades cúbicas
C. Sistema de control de acceso	Nivel de acceso (0-5), tarjeta activa (True/False), días desde último cambio de contraseña	Verificar si nivel ≥ requerido, tarjeta activa y contraseña actualizada	Mensaje de "Acceso permitido" o "Acceso denegado"
D. Sistema de descuentos en tienda en línea	Monto de compra, condición de cliente VIP, código de descuento (True/False)	Calcular descuentos acumulativos: 20% > $100, +10% VIP, +5% código especial	Monto total a pagar y porcentaje de descuento aplicado

Fig1:La tabla presenta la descomposición de cada uno de los ejercicios del laboratorio en términos de entradas, procesos y salidas

III. METODOLOGÍA

Para el desarrollo del laboratorio se aplicó la metodología de resolución de problemas, la cual consiste en identificar la situación planteada, analizarla, diseñar una solución en forma de algoritmo y finalmente implementarla en un lenguaje de programación.

En primer lugar, cada problema fue analizado para comprender sus requerimientos y restricciones. Posteriormente, se elaboraron diagramas de flujo utilizando la herramienta PSeInt[4], lo que permitió visualizar de forma estructurada los pasos a seguir en la solución. Una vez definidos los algoritmos, se procedió a su implementación en el lenguaje Python[1], empleando el entorno de desarrollo Visual Studio Code.

El trabajo colaborativo se llevó a cabo mediante el uso de Git y GitHub, donde se creó un repositorio central en el que cada integrante del equipo realizó aportes y actualizaciones, favoreciendo la integración de las soluciones y la trazabilidad de los cambios.

Finalmente, los resultados obtenidos se documentaron en un informe en formato IEEE, siguiendo los lineamientos establecidos por la materia de Programación 1.

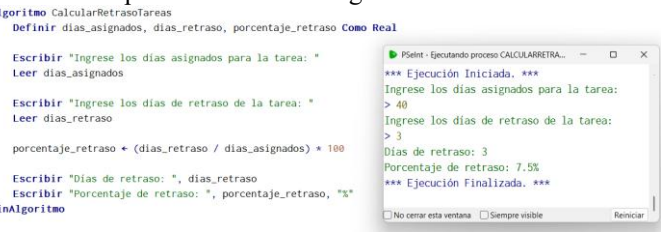

Imagen de uso de Pseint para ejercicio b

A. *Cálculo de retraso en proyectos de software*

1. Identificación del problema: determinar los días de retraso en tareas entregadas fuera de plazo.
2. Definición de variables: fecha límite, fecha de finalización real, tiempo total asignado.
3. Proceso: calcular la diferencia entre fechas y determinar el porcentaje de retraso con la fórmula establecida.
4. Implementación: se diseñó un diagrama de flujo en PSeInt y posteriormente se codificó en Python.
5. Validación: se probaron diferentes escenarios con entregas puntuales y retrasadas.

```python
dias_asignados = float(input("Ingrese los días asignados para la tarea: "))
dias_retraso = float(input("Ingrese los días de retraso de la tarea: "))

porcentaje_retraso = (dias_retraso / dias_asignados) * 100

print("Días de retraso:", dias_retraso)
print("Porcentaje de retraso:", porcentaje_retraso, "%")
```

Fig2:Código del programa realizado en Python.Ejercicio1.

B. *Cálculo del volumen de una esfera*

1. Identificación del problema: calcular el volumen de una esfera a partir de su radio.
2. Definición de variables: radio (dato de entrada) y volumen (dato de salida).
3. Proceso: aplicar la fórmula matemática

$$V = \frac{4}{3}\pi r^3$$

4. Implementación: se realizó un diagrama de flujo en PSeInt y el código fue escrito en Python.
5. Validación: se comprobaron valores de radios distintos y se verificaron los resultados con la fórmula matemática.

```python
import math

radio = float(input("Ingrese el radio de la pelota (esfera): "))
pi = 3.14159

volumen = (4/3) * pi * (radio**3)

print("El volumen de la pelota es:", volumen)
```

Fig3:Código del programa realizado en Python. Ejercicio2.

C. Sistema de control de acceso

1. Identificación del problema: verificar si un usuario puede acceder a un área restringida según su nivel de acceso y condiciones de seguridad.
2. Definición de variables: nivel de acceso, estado de la tarjeta, días desde el último cambio de contraseña.
3. Proceso: aplicar condiciones lógicas con estructuras condicionales ***if-else.***
4. Implementación: se elaboró el diagrama de flujo en PSeInt y se desarrolló el programa en Python.
5. Validación: se ingresaron diferentes combinaciones de datos para comprobar accesos permitidos y denegados.

```python
nivel_acceso_requerido = 3

nivel_acceso_usuario = int(input("Ingrese su nivel de acceso (0-5): "))
tarjeta_activa = input("¿Su tarjeta está activa? (True/False): ") == 'True'
contrasena_reciente = input("¿Cambió su contraseña en los últimos 30 días? (True/False): ") == 'True'

if nivel_acceso_usuario >= nivel_acceso_requerido and tarjeta_activa and contrasena_reciente:
    print("Acceso permitido.")
else:
    print("Acceso denegado.")
```

Fig4:Código del programa realizado en Python.Ejercicio3.

D. Sistema de descuentos en tienda en línea

1. Identificación del problema: calcular el monto a pagar aplicando descuentos acumulativos según condiciones de compra.
2. Definición de variables: monto total de la compra, condición VIP, existencia de código de descuento.
3. Proceso: verificar reglas de negocio y aplicar descuentos progresivos (20%, 10% y 5%).
4. Implementación: se diseñó un diagrama de flujo en PSeInt y se programó en Python.
5. Validación: se realizaron pruebas con diferentes montos y condiciones (cliente normal, VIP, con o sin código).

```python
monto_compra = float(input("Ingrese el monto total de la compra: "))
es_vip = input("¿Es usted miembro VIP? (True/False): ") == 'True'
tiene_codigo = input("¿Tiene un código de descuento especial? (True/False): ") == 'True'

descuento_total = 0

if monto_compra > 100:
    descuento_total = descuento_total + 0.20

if es_vip:
    descuento_total = descuento_total + 0.10

if tiene_codigo:
    descuento_total = descuento_total + 0.05

monto_con_descuento = monto_compra * (1 - descuento_total)

print("Monto original:", monto_compra)
print("Descuento total aplicado:", descuento_total * 100, "%")
print("Total a pagar después de los descuentos:", monto_con_descuento)
```

Fig5:Código del programa realizado en Python.Ejercicio4.

IV. Resultados

A. Cálculo de retraso en proyectos de software

El programa permitió calcular correctamente los días de retraso y el porcentaje de incumplimiento.

- **Entrada:** Fecha límite = 10 de marzo, Fecha real = 15 de marzo, Tiempo total asignado = 20 días.
- **Proceso:** Retraso = 5 días, Porcentaje de retraso = (5/20) × 100 = 25%.
- **Salida:** "La tarea presenta 5 días de retraso, equivalente al 25% del tiempo asignado". Esto confirma que el algoritmo es capaz de medir objetivamente el incumplimiento en los proyectos.

```
Ingrese los días asignados para la tarea: 20
Ingrese los días de retraso de la tarea: 5
Días de retraso: 5.0
Porcentaje de retraso: 25.0 %
PS C:\Users\juand\Documents\Repo_lab 10sep>
```

Resultados con 20 días asignados y 5 de retraso

B. Cálculo del volumen de una esfera

Entrada: Radio = 5.

- **Proceso:** Volumen = (4/3) × π × (5^3) = 523.598.
- **Salida:** "El volumen de la pelota es: 523.5983333333332".

El resultado coincide con el valor esperado, validando la implementación.

```
Ingrese el radio de la pelota (esfera): 5
El volumen de la pelota es: 523.5983333333332
PS C:\Users\juand\Documents\Repo_lab 10sep>
```

Resultados con radio 5

C. Sistema de control de acceso

El programa permitió verificar accesos de acuerdo con las condiciones definidas.

- **Caso 1 (válido):** Nivel de acceso = 4, Tarjeta activa = True, Contraseña reciente = True.
  - **Salida:** "Acceso permitido".
- **Caso 2 (inválido):** Nivel de acceso = 2, Tarjeta activa = True, Contraseña reciente = True.
  - **Salida:** "Acceso denegado". Esto demuestra que el sistema aplica correctamente las reglas de seguridad y restringe accesos indebidos.

```
Ingrese su nivel de acceso (0-5): 4
¿Su tarjeta está activa? (True/False): True
¿Cambió su contraseña en los últimos 30 días? (True/False): True
Acceso permitido.
PS C:\Users\juand\Documents\Repo_lab 10sep>
```

Acceso permitido

```
Ingrese su nivel de acceso (0-5): 2
¿Su tarjeta está activa? (True/False): False
¿Cambió su contraseña en los últimos 30 días? (True/False): False
Acceso denegado.
PS C:\Users\juand\Documents\Repo_lab 10sep>
```

Acceso denegado

D. Sistema de descuentos en tienda en línea

El algoritmo calculó descuentos acumulativos de manera precisa.

- **Caso 1:** Compra = $120, VIP = True, Código = True.
  - Descuentos = 20% + 10% + 5% = 35%.
  - **Salida:** "Total a pagar: $78.0".
- **Caso 2:** Compra = $80, VIP = False, Código = True.
  - Descuento = 5%.
  - **Salida:** "Total a pagar: $76.0". Los resultados muestran que el sistema aplica correctamente las políticas de descuento según las condiciones ingresadas.

```
Ingrese el monto total de la compra: 120
¿Es usted miembro VIP? (True/False): True
¿Tiene un código de descuento especial? (True/False): True
Monto original: 120.0
Descuento total aplicado: 35.0 %
Total a pagar después de los descuentos: 77.99999999999999
PS C:\Users\juand\Documents\Repo_lab 10sep>
```

Caso1

```
Ingrese el monto total de la compra: 80
¿Es usted miembro VIP? (True/False): True
¿Tiene un código de descuento especial? (True/False): False
Monto original: 80.0
Descuento total aplicado: 10.0 %
Total a pagar después de los descuentos: 72.0
PS C:\Users\juand\Documents\Repo_lab 10sep>
```

Caso2

.

V. EVALUACIÓN PRUEBAS Y DESPLIEGUE

E Para garantizar la validez de los algoritmos desarrollados, se realizaron pruebas unitarias y de funcionamiento en cada uno de los ejercicios planteados. La evaluación se centró en tres aspectos: exactitud de los cálculos, cumplimiento de las condiciones lógicas y presentación de resultados correctos al usuario.

Pruebas realizadas

Ejercicio A: Se validaron casos con tareas entregadas en fecha, con retrasos cortos y con retrasos mayores al tiempo asignado, verificando que el cálculo de días y porcentajes fuera consistente.

Ejercicio B: Se probaron radios enteros y decimales, confirmando que los resultados coincidieran con la fórmula matemática del volumen de la esfera.

Ejercicio C: Se evaluaron combinaciones de niveles de acceso, tarjetas activas/inactivas y cambios de contraseña recientes/no recientes, comprobando que solo los usuarios que cumplían con todas las condiciones obtenían acceso.

Ejercicio D: Se analizaron diferentes escenarios de compra (mayores y menores a $100), con clientes VIP y con códigos de descuento, comprobando que los descuentos acumulativos se aplicaran correctamente.

Evaluación de resultados

Los programas cumplieron con las especificaciones planteadas y los resultados coincidieron con los valores esperados en todos los casos de prueba. El uso de estructuras condicionales, operadores lógicos y fórmulas matemáticas se implementó de manera adecuada.

Despliegue y control de versiones

La implementación se realizó en el entorno de desarrollo Visual Studio Code, y la gestión colaborativa del proyecto se efectuó a través de Git y GitHub, lo que permitió mantener un repositorio actualizado, registrar los cambios realizados por cada integrante del equipo y asegurar la trazabilidad del desarrollo.

.

VI. CONCLUSIONES

1. El laboratorio permitió aplicar de manera práctica la metodología de resolución de problemas, estructurando algoritmos desde su análisis hasta su implementación en Python.

2. El uso de herramientas como PSeInt para diagramas de flujo y Python para la programación facilitó la comprensión de los procesos lógicos, mientras que Git y GitHub fortalecieron las competencias en trabajo colaborativo y control de versiones.

3. Cada ejercicio representó un escenario realista, desde el cálculo de retrasos en proyectos hasta la implementación de sistemas de seguridad y descuentos en línea, lo cual demostró la aplicabilidad de la programación a situaciones del mundo real.

4. Se fortalecieron habilidades fundamentales para la ingeniería de sistemas, como el razonamiento lógico,

el trabajo en equipo y las buenas prácticas en el desarrollo de software, sentando bases sólidas para proyectos más complejos en el futuro.

VII. Referencias

[1] Python Software Foundation, Python 3.12 Documentation. [En línea]. Disponible en: https://docs.python.org/3/

[2] Math Module — Python 3.12 Documentation, Mathematical functions. [En línea]. Disponible en: https://docs.python.org/3/library/math.html

[3] J. Guttag, Introduction to Computation and Programming Using Python, 3rd ed. MIT Press, 2021.

[4] L. Ramírez, Algoritmos y diagramas de flujo con PSeInt. Bogotá: Ecoe Ediciones, 2019.

[5] Git Documentation, Pro Git Book. [En línea]. Disponible en: https://git-scm.com/doc

[6] GitHub Docs, Getting started with GitHub. [En línea]. Disponible en: https://docs.github.com/

[7] Microsoft, Visual Studio Code Documentation. [En línea]. Disponible en: https://code.visualstudio.com/docs