

Informe de Depuración de

Código: Solución de Errores y

Adopción de Buenas Prácticas

En este ejercicio de ingeniería inversa desafía a los candidatos a comprender, identificar errores, señalar prácticas y corregir un código problemático como parte del proceso de selección de una empresa.

El código proporcionado para el análisis es el siguiente:

```
p = [5, 3, 8, 9, 7]
def calc(a, b, c):
    if (c == 0):
        return -1
    elif (a[c] == b):
        return c
    else:
        calc(a, b, c - 1)

x = input()
calc(p, x, length(p))
```

a) ¿Qué hace o qué pretende hacer el código?

El código parece intentar buscar un valor *b* en la lista *a* y devolver el índice de la primera aparición de ese valor. Hay algunos errores en el código que impiden que funcione correctamente.

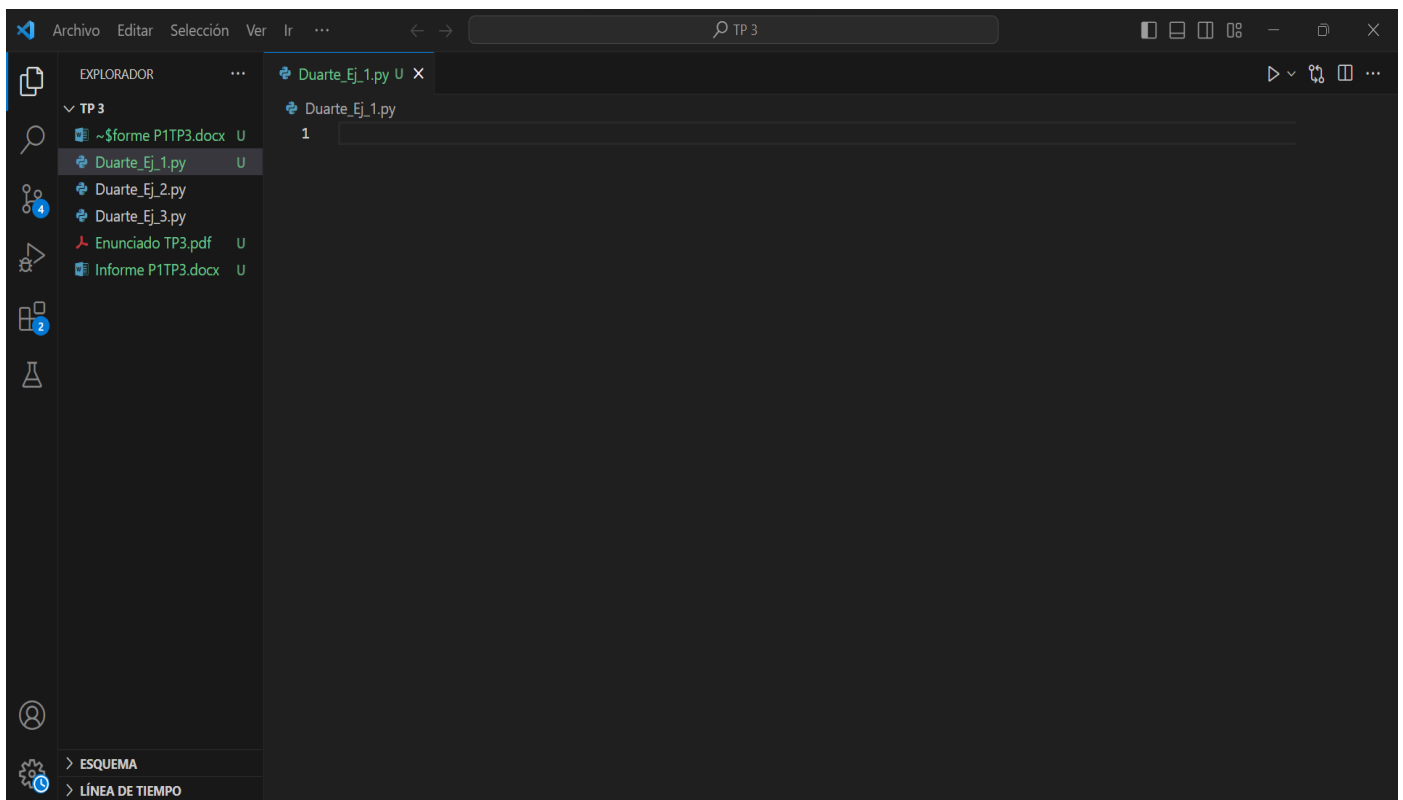
b) ¿Qué errores tiene? Enumerarlos.

1° La función `length` no está definida. Debería ser `len`.

2° La llamada a la función `calc` en la última línea del código tiene un error tipográfico en el nombre de la función. Debería ser `len(p)` en lugar de `length(p)`.

Utilizaré Visual Studio Code como entorno de desarrollo para mejorar el código proporcionado y verificar si hay algún error mediante el debugging. Abordare los cambios necesarios para optimizar el código y de su funcionamiento correcto. El proceso incluirá corrección de posibles errores, así como la verificación de la lógica del programa a través del debugging.

Creo un archivo .py para verificar y mejorar el código. Posteriormente, compartiré una captura de pantalla para documentar el proceso.



Código original a verificar y solucionar

```
Duarte_Ej_1.py 7, U •
Duarte_Ej_1.py > ...
1  p = [5, 3, 8, 9, 7]
2  def calc(a, b, c):
3      if (c == 0):
4          return -1
5          elif (a[c] == b):
6              return c
7      else:
8          calc(a, b, c - 1)
9      x = input()
10     calc(p, x, length(p))
11
```

Código verificado y corregido listo para debugging

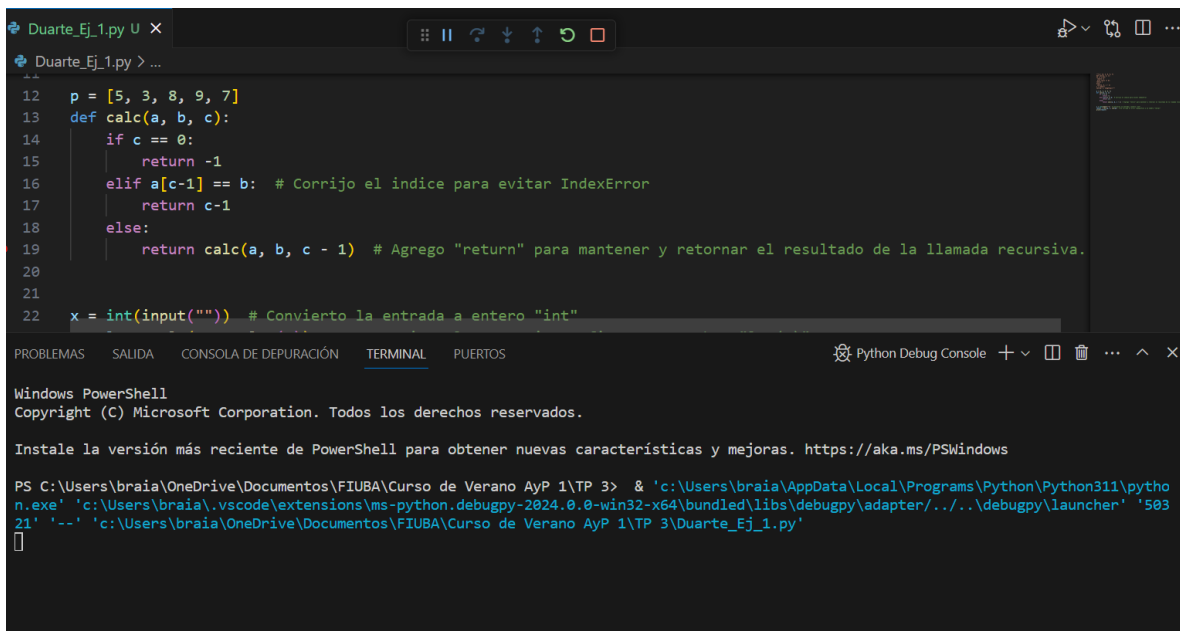
```
p = [5, 3, 8, 9, 7]
def calc(a, b, c):
    if c == 0:
        return -1
    elif a[c-1] == b: # Corrijo el indice para evitar IndexError
        return c-1
    else:
        return calc(a, b, c - 1) # Agrego "return" para mantener y retornar el resultado de la llamada recursiva.

x = int(input()) # Convierto la entrada a entero "int"
result = calc(p, x, len(p)) # Se corrige el error tipografico y se nombra "len(p)"
print(result)
```

Se ha colocado un breakpoint en la línea `return calc(a, b, c - 1)` dentro de la función `calc`.

```
18         else:
19             return calc(a, b, c - 1)
20
```

Al iniciar la depuración, apareció el mensaje en la terminal. Este mensaje indica el inicio del proceso de depuración en Visual Studio Code.



The screenshot shows the Visual Studio Code interface. The editor displays a Python file named `Duarte_Ej_1.py` with the following code:

```
12 p = [5, 3, 8, 9, 7]
13 def calc(a, b, c):
14     if c == 0:
15         return -1
16     elif a[c-1] == b: # Corrijo el indice para evitar IndexError
17         return c-1
18     else:
19         return calc(a, b, c - 1) # Agrego "return" para mantener y retornar el resultado de la llamada recursiva.
20
21
22 x = int(input("")) # Convierto la entrada a entero "int"
```

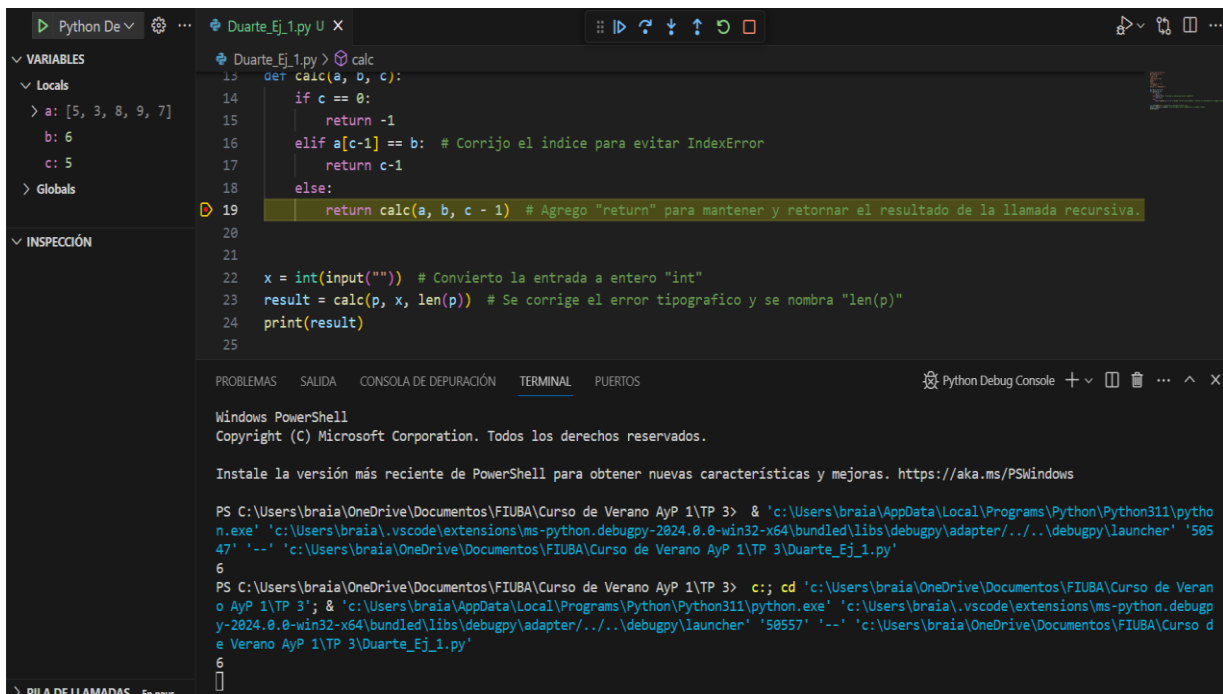
A red dot breakpoint is set on line 19. The bottom panel shows the 'TERMINAL' tab with the following output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\braia\OneDrive\Documentos\FIUBA\Curso de Verano AyP 1\TP 3> & 'c:\Users\braia\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\braia\.vscode\extensions\ms-python.debugpy-2024.0.0-win32-x64\bundle\libs\debugpy\adapter\..\..\debugpy\launcher' '50321' '--' 'c:\Users\braia\OneDrive\Documentos\FIUBA\Curso de Verano AyP 1\TP 3\Duarte_Ej_1.py'
```

Después de ingresar el número 6 y presionar Enter, el programa continúa su ejecución desde el punto del breakpoint. La flecha a la izquierda del return indica la próxima línea de código a ejecutar después de salir de la función calc.



```
Python De v ... Duarte_Ej_1.py U X
Duarte_Ej_1.py > calc
13 def calc(a, b, c):
14     if c == 0:
15         return -1
16     elif a[c-1] == b: # Corrijo el indice para evitar IndexError
17         return c-1
18     else:
19         return calc(a, b, c - 1) # Agrego "return" para mantener y retornar el resultado de la llamada recursiva.
20
21
22 x = int(input("")) # Convierto la entrada a entero "int"
23 result = calc(p, x, len(p)) # Se corrige el error tipografico y se nombra "len(p)"
24 print(result)
25

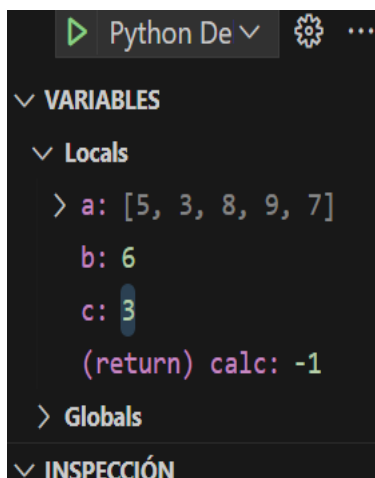
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
Python Debug Console + - [ ] ... ^ x

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\braia\OneDrive\Documentos\FIUBA\Curso de Verano AyP 1\TP 3> & 'c:\Users\braia\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\braia\.vscode\extensions\ms-python.debugpy-2024.0.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '50547' '--' 'c:\Users\braia\OneDrive\Documentos\FIUBA\Curso de Verano AyP 1\TP 3\Duarte_Ej_1.py'
6
PS C:\Users\braia\OneDrive\Documentos\FIUBA\Curso de Verano AyP 1\TP 3> c::cd 'c:\Users\braia\OneDrive\Documentos\FIUBA\Curso de Verano AyP 1\TP 3'; & 'c:\Users\braia\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\braia\.vscode\extensions\ms-python.debugpy-2024.0.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '50557' '--' 'c:\Users\braia\OneDrive\Documentos\FIUBA\Curso de Verano AyP 1\TP 3\Duarte_Ej_1.py'
6
[]
```

Durante la depuración, en el lado izquierdo se presenta el estado de las variables. En el Centro Superior se encuentra los botones del paso a paso con las distintas flechas de depuración.

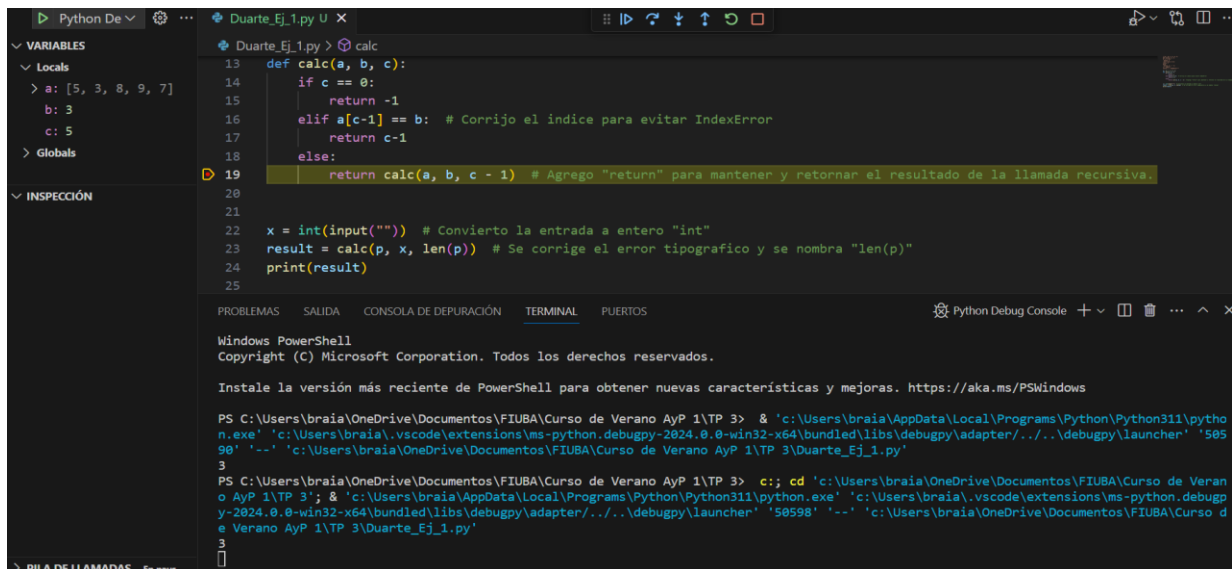


```
Python De v ...
VARIABLES
Locals
> a: [5, 3, 8, 9, 7]
b: 6
c: 3
(return) calc: -1
> Globals
INSPECCIÓN
```

Durante la depuración, al ingresar el número 6, las variables locales disminuyen gradualmente en la función hasta llegar al punto de impresión final.

El valor -1 se devolvió porque C llegó a ser 0 durante la recursión. Cuando C es 0, la función devuelve -1, lo que indica que el valor buscado no está en la lista.

En Caso contrario que el número se encuentre en la lista

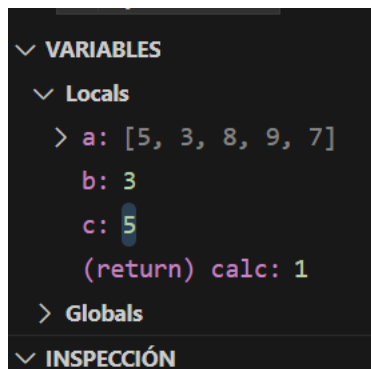


The screenshot shows a VS Code editor with a Python file named `Duarte_Ej_1.py`. The code defines a recursive function `calc(a, b, c)` that checks if `b` is in the list `a`. If `b` is found, it returns `-1`; otherwise, it returns `calc(a, b, c - 1)`. The main part of the script takes user input `x`, converts it to an integer, and calls `calc(p, x, len(p))` where `p` is the list `[5, 3, 8, 9, 7]`. The terminal shows the execution of the script, where the user enters `3` and the output is `1`.

```
def calc(a, b, c):
    if c == 0:
        return -1
    elif a[c-1] == b: # Corrijo el indice para evitar IndexError
        return c-1
    else:
        return calc(a, b, c - 1) # Agrego "return" para mantener y retornar el resultado de la llamada recursiva.

x = int(input("")) # Convierto la entrada a entero "int"
result = calc(p, x, len(p)) # Se corrige el error tipografico y se nombra "len(p)"
print(result)
```

```
PS C:\Users\braia\OneDrive\Documentos\FIUBA\Curso de Verano AyP 1\TP 3> & 'c:\Users\braia\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\braia\.vscode\extensions\ms-python.debugpy-2024.0.0-win32-x64\bundle\libs\debugpy\adapter\..\..\debugpy\launcher' '50598' '--' 'c:\Users\braia\OneDrive\Documentos\FIUBA\Curso de Verano AyP 1\TP 3\Duarte_Ej_1.py'
3
PS C:\Users\braia\OneDrive\Documentos\FIUBA\Curso de Verano AyP 1\TP 3> c;; cd 'c:\Users\braia\OneDrive\Documentos\FIUBA\Curso de Verano AyP 1\TP 3'; & 'c:\Users\braia\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\braia\.vscode\extensions\ms-python.debugpy-2024.0.0-win32-x64\bundle\libs\debugpy\adapter\..\..\debugpy\launcher' '50598' '--' 'c:\Users\braia\OneDrive\Documentos\FIUBA\Curso de Verano AyP 1\TP 3\Duarte_Ej_1.py'
3
1
```



The Variables panel shows the following state:

- Locals:**
 - `a`: `[5, 3, 8, 9, 7]`
 - `b`: `3`
 - `c`: `5`
 - `(return) calc`: `1`
- Globals:** (empty)

Si se ingresa el número 3 y está en la lista, devolvería en este caso, el valor 1.

C) ¿Qué cosas señalarías que no son errores, pero no se corresponden con las buenas prácticas explicadas en clase? Enumerarlas.

1° La falta de comentarios que expliquen el propósito y el funcionamiento del código.

2° La variable `p` podría tener un nombre más descriptivo.

3° La función `calc` no sigue las convenciones de nombres de funciones, que deberían ser verbos

4° La recursión en la función `calc` no parece tener un propósito claro y puede causar un desbordamiento para listas grandes.

d) Corregir el programa para que funcione correctamente y cumpla con las buenas prácticas enseñadas.

```
#Codigo mas legible y mas autentico

def buscar_indice_elemento(lista, elemento, indice):

    """
    PRE: La lista debe tener al menos un elemento, y el índice y el elemento buscado deben ser validos.

    POST: La función devuelve un índice válido (o -1 si el elemento no se encuentra), buscando el elemento de manera descendente en la lista.

    """
    if indice == 0:
        return -1
    elif lista[indice - 1] == elemento:
        return indice - 1
    else:
        return buscar_indice_elemento(lista, elemento, indice - 1)

def main():
    lista_numeros = [5, 3, 8, 9, 7]
    numero_buscado = int(input("Ingrese un número: "))
    resultado = buscar_indice_elemento(lista_numeros, numero_buscado, len(lista_numeros))
    if resultado != -1:
        print(f"El elemento {numero_buscado} se encuentra en el índice {resultado}.")
    else:
        print(f"El elemento {numero_buscado} no se encuentra en la lista.")

if __name__ == "__main__":
    main()
```

Se ha creado una función llamada `buscar_indice_elemento` para buscar un elemento de manera descendente en una lista. La función recibe tres parámetros la lista en la que se busca, el elemento que se quiere encontrar y el índice inicial de búsqueda. En la función `main`, se solicita al usuario ingresar un número para buscar en la lista `lista_numeros`, luego se llama a `buscar_indice_elemento` y se muestra un mensaje indicando si el elemento se encontró y en qué índice, o si no se encontró en la lista.