

Manual Técnico: Aplicación en Python con Streamlit

Universidad de Colima (UdeC)

Facultad de Ingeniería en Computación inteligente (ICI)

Materia: Programación Funcional

Nombre del Estudiante: Luis Carlos Cortez Guzmán

1. Introducción

Este manual detalla el funcionamiento de una aplicación interactiva desarrollada en Python utilizando la biblioteca Streamlit. La aplicación incluye un menú lateral con varias opciones, cada una representando diferentes ejercicios y funciones.

2. Descripción del Código

2.1. Librerías Utilizadas

El código utiliza la librería streamlit para crear una interfaz gráfica interactiva. Esta herramienta permite que los usuarios seleccionen diversas opciones y realicen cálculos o interactúen con datos de forma visual.

```
import streamlit as st
```

2.2. Menú de Opciones

El menú lateral contiene múltiples opciones, que permiten acceder a diferentes funcionalidades del programa. Las opciones son presentadas mediante un selectbox.

```
# Opciones del menú
menu_opciones = ["Inicio", "Saludo", "Suma de dos números", "Área de un triángulo",
                 "Calculadora de descuento", "Suma de lista", "Funciones con valores predeterminados",
                 "Pares e impares", "Multiplicación con *args", "Información personal con **kwargs",
                 "Calculadora flexible", "Acerca de"]
selected_option = st.sidebar.selectbox("Selecciona una opción", menu_opciones)
```

2.3. Funciones Definidas

A continuación, se describen las principales funciones implementadas:

2.3.1. Función saludar

Recibe un nombre como parámetro y retorna un saludo personalizado.

```
def saludar(nombre: str) -> str:  
    return f"Hola, {nombre}"
```

2.3.2. Función sumar

Realiza la suma de dos números y devuelve el resultado.

```
def sumar(n1: float, n2: float) -> str:  
    return f"El resultado es: {n1 + n2}"
```

2.3.3. Función calcular_area_triangulo

Calcula el área de un triángulo con base en la fórmula: $\text{Área} = (\text{base} * \text{altura}) / 2$

```
def calcular_area_triangulo(base: float, altura: float) -> str:  
    return f"El área del triángulo es: {(base * altura) / 2}"
```

2.3.4. Función calcular_precio_final

Calcula el precio final de un producto aplicando un descuento y un impuesto sobre el valor.

```
def calcular_precio_final(precio_original: float, descuento: float = 10, impuesto: float = 16) -> str:  
    precio_con_descuento = precio_original * (1 - descuento / 100)  
    precio_final = precio_con_descuento * (1 + impuesto / 100)  
    return f"El precio final con un {descuento}% de descuento y {impuesto}% de impuesto es: {round(precio_final, 2)}"
```

2.3.5. Función sumar_lista

Suma todos los elementos de una lista de números.

```
def sumar_lista(numeros: list) -> float:  
    return sum(numeros)
```

2.3.6. Función producto

Calcula el costo total de una cantidad específica de productos a un precio dado.

```
def producto(nombre: str, cantidad: int = 1, precio: float = 10.0) -> str:
    return f"El total a pagar por {cantidad} {nombre} es: {cantidad * precio}"
```

2.3.7. Función numeros_pares_e_impares

Separa una lista de números en dos listas: una de números pares y otra de impares.

```
def numeros_pares_e_impares(numeros: List) -> tuple:
    pares = [n for n in numeros if n % 2 == 0]
    impares = [n for n in numeros if n % 2 != 0]
    return pares, impares
```

2.3.8. Función multiplicar_todos

Multiplica todos los números proporcionados como argumentos usando *args.

```
def multiplicar_todos(*args) -> int:
    resultado = 1
    for numero in args:
        resultado *= numero
    return resultado
```

2.3.9. Función informacion_personal

Recibe múltiples valores de entrada usando **kwargs y los muestra en pantalla.

```
def informacion_personal(**kwargs):
    for clave, valor in kwargs.items():
        st.write(f"{clave.capitalize()}: {valor}")
```

2.3.10. Función calculadora_flexible

Realiza operaciones básicas (suma, resta, multiplicación y división) entre dos números.

```
def calculadora_flexible(num1, num2, operacion="suma"):
    operaciones = {
        "suma": num1 + num2,
        "resta": num1 - num2,
        "multiplicación": num1 * num2,
        "división": num1 / num2 if num2 != 0 else "Error: División por cero"
    }
    return operaciones.get(operacion, "Operación no válida")
```

3. Flujo del Programa

Dependiendo de la opción seleccionada en el menú, el programa ejecuta diferentes secciones de código. Para cada opción, se muestra una interfaz donde el usuario puede ingresar los datos requeridos y obtener el resultado de la función correspondiente.

4. Conclusión

Este programa es una excelente demostración de cómo combinar funciones matemáticas y estructuras de control en un entorno interactivo utilizando Streamlit. Proporciona múltiples ejercicios que refuerzan conceptos fundamentales de programación funcional y la gestión de datos en Python.