



Estudiante: Ana María Patiño Cano

Documento de identificación: 1214724501

Curso: Fundamentos de programación con Python

Actividad: Reto 4

Punto 1. Tome el enunciado y la solución elaborada del reto 3, identifique si aplicó el paradigma funcional, si lo aplicó correctamente, si faltaron funciones o pudieron realizarse otras. Construya un documento que recopile las falencias, aciertos, dudas y dificultades que se le presentaron al aplicar los conceptos del paradigma funcional a la solución del reto 3.

En caso de no haber aplicado el paradigma funcional en el reto 3, analice cómo pudiera aplicarlo, qué funciones haría y qué pasos debería seguir para crear la solución bajo este paradigma, y construya un documento tipo texto con los resultados de este análisis y sugerencias, de tal forma que un programador pudiera realizar dicha implementación bajo el paradigma funcional.

Desarrollo:

El código realizado fue el siguiente:

<https://drive.google.com/file/d/1CflsNp4P8ZgDMtSZeLGcM4o1NxdtQpel/view?usp=sharing>. Sí se desarrolló teniendo en cuenta elementos del paradigma funcional.

1. Aciertos:

- 1.1 La definición de funciones como “entrada”, “ingreso_costos”, “costos_totales” y “calcular_promedio”.

```
def entrada(valor_entrada):
    valor_entrada = valor_entrada
    if (valor_entrada==""):
        valor_entrada = float(0)
    else:
        valor_entrada = float(valor_entrada)
    return valor_entrada

def ingreso_costos(mes):
    obra = input(f"Ingrese los costos de mano de obra para el mes {mes}: ")
    obra = entrada(obra)
    abono = input(f"Ingrese los costos de abono para el mes {mes}: ")
    abono = entrada(abono)
    agua = input(f"Ingrese los costos de agua para el mes {mes}: ")
    agua = entrada(agua)
    mantenimiento = input(f"Ingrese los costos de mantenimiento para el mes {mes}: ")
    mantenimiento = entrada(mantenimiento)
    return obra, abono, agua, mantenimiento

def costos_totales(costos_totales):
    return sum(costos_totales)
def calcular_promedio(costos_totales):
    return sum(costos_totales) / len(costos_totales)
```



- 1.2 La función de “info_contable” que solicita al usuario información específica para poder realizar el cálculo de los costos totales y promedio del cultivo del arroz.

```
def info_contable():  
    meses = int(input("Indique la duración en meses del cultivo de arroz:"))  
    meses = (meses + 1)
```

- 1.3 El uso de la lista “costos_totales” fue útil para almacenar los costos mensuales, ya que permite operaciones como la suma y el cálculo del promedio en todo el conjunto de datos.

```
costos_totales = []  
  
for mes in range(1, meses):  
    obra, abono, agua, mantenimiento = ingreso_costos(mes)  
  
    costos_mes_total = obra + abono + agua + mantenimiento  
    costos_totales.append(costos_mes_total)  
  
print("""  
Costos totales discriminados por meses de cultivo: """)  
for mes in range(1, meses):  
    print("Mes", mes, ":")  
    print("Costos Totales: $",(costos_totales[mes - 1]))  
    print("""Promedio de Costos mensuales: """, round(calcular_promedio(costos_totales), 2))
```

2. Mejoras del código:

- 2.1 **Menú principal:** Fue desarrollado con un bucle “while”, sin embargo cada opción del menú podría ser una función separada para que sea más legible dividiendo en funciones pequeñas.

<pre>#El menú principal se plantea a modo de bucle: while True: print(""" Bienvenido a la información para la huerta escolar, por favor indique una de las siguientes opciones: 1. Horario de gestión del cultivo 2. Etapas del cultivo 3. Información contable 4. Salir""") #Se indica a continuación submenú 1 para el tipo de cultivo de interés, por lo #sería un condicional dentro de otro condicional opcion = input() if opcion == "1": print(""" Para saber los horarios de gestión, indique el cultivo en el que tiene interés 1. Arroz 2. Garbanzo 3. Lenteja""") #Esta opción para la gestión del cultivo de arroz (condicional) opcion2 = input() if opcion2 == "1": print(""" El cultivo de arroz requiere: Mantenimiento: Eliminación de malezas y monitoreo de plagas. Días Lunes a viernes, 9:00am o 5:00pm Abono: Al inicio de la temporada de crecimiento. Días martes y jueves, 9:00am Regado: Días Miércoles y viernes, 9:00am o 5:00pm""")</pre>	Código anterior
---	------------------------



Por ejemplo, incluyendo nuevas funciones para “mostrar_menu_principal”, “obtener_opcion”, “gestionar_cultivo”, “mostrar_etapas_cultivo”, entre otras:

```
def mostrar_menu_principal():  
    print("""  
    Bienvenido a la información para la huerta escolar, por favor indique una  
    de las siguientes opciones:  
  
    1. Horario de gestión del cultivo  
    2. Etapas del cultivo  
    3. Información contable  
    4. Salir""")  
  
def obtener_opcion():  
    opcion = input("Ingrese su opción: ")  
    return opcion  
  
def gestionar_cultivo():  
    print("""  
    Para saber los horarios de gestión, indique el cultivo en el que tiene interés  
    1. Arroz  
    2. Garbanzo  
    3. Lenteja""")  
    opcion2 = input("Ingrese su opción: ")  
  
def mostrar_etapas_cultivo():  
    print("""  
    Para saber las etapas, indique el cultivo en el que tiene interés  
    1. Arroz  
    2. Garbanzo  
    3. Lenteja""")  
    opcion3 = input("Ingrese su opción: ")
```

Código mejorado

Definidas dichas funciones, luego podría implementarse de mejor manera en el bucle “while” principal:

```
# El bucle principal  
while True:  
    mostrar_menu_principal()  
    opcion = obtener_opcion()  
  
    if opcion == "1":  
        gestionar_cultivo()  
  
    elif opcion == "2":  
        mostrar_etapas_cultivo()  
  
    elif opcion == "3":  
        info_contable()  
  
    elif opcion == "4":  
        break  
  
    else:  
        print("Opción incorrecta")
```

Código mejorado



- 2.2 **Comentarios:** Pudo estar más documentado el proceso de la definición del código para mayor legibilidad, esto es, realizar más comentarios con “#” para que la documentación explique el funcionamiento y el propósito de las líneas de código.

3. Dificultades:

3.1 **Informe económico:** Fue difícil para mí realizar código para los cálculos de ganancia en caso de que el precio del arroz se incrementara en un 37% de su valor actual, y el cálculo de ganancia si los costos y los gastos se disminuyen en un 5% y la cantidad de arrobas producidas en un 63%. Esta dificultad referida más a la interpretación del ejercicio que no la hice adecuadamente, lo que me complicó su desarrollo en código.

3.2 **Trabajo en equipo:** La programación en equipo requiere de buena comunicación, en este reto, fue muy difícil lograrla por lo que el ejercicio finalmente lo desarrollé de manera individual (reto 3 y punto 1 de reto 4)