Python para principiantes

ACTIVIDAD PRÁCTICA

Reto 4

**RETO**

**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.

**Solución:**

Creo que el código que proporcioné no se adhiere al paradigma de programación funcional. Debido a que, según lo visto, la programación funcional se centra en el uso de funciones puras, la inmutabilidad y la evitación del estado compartido, sin embargo, el código que entregué sigue principalmente un enfoque imperativo, ya que se basa en gran medida en estructuras de datos mutables y operaciones de entrada/salida.

Por ello, algunas mejoras a este código para que se ajuste más al paradigma de programación funcional podrían ser:

1. **Inmutabilidad**: En programación funcional, se promueve la inmutabilidad. En este código, los datos dentro de **cultivos** son mutables. Para seguir el paradigma funcional, podríamos considerar crear nuevas estructuras de datos en lugar de modificar las existentes.
2. **Funciones puras**: La programación funcional se basa en funciones puras que no tienen efectos secundarios. Algunas funciones en este código, como **mostrar\_horarios** y **mostrar\_etapas**, tienen efectos secundarios (como imprimir en la consola) y no devuelven valores. Podríamos modificar estas funciones para que devuelvan datos en lugar de imprimirlos.
3. **Evitar bucles**: En lugar de utilizar bucles explícitos, la programación funcional favorece la recursividad y las funciones de orden superior. El código contiene un bucle **while** en la función **mostrar\_menu**. Podríamos considerar el uso de recursividad o funciones de orden superior para procesar la entrada del usuario.
4. **Transformación de datos**: La programación funcional implica la transformación de datos a través de funciones como map, filtery reduce. En este código, se calculan varias métricas financieras manualmente, por ello podrían reanudarse en funciones puras que operan en colecciones de datos.
5. **Separación de preocupaciones**: Para seguir el paradigma funcional, sería bueno modularizar las preocupaciones en funciones o módulos separados en lugar de combinar lógicas de interfaz de usuario, procesamiento de datos y cálculos en un solo script como está en mi código de la tarea 3.

Aplicando los cambios anteriores, se podría lograr una base de código más funcional y modular que se alinee mejor con los principios de la programación funcional.