

Análisis de Paradigma Funcional - Reto 3

Estudiante: Elkin Mauricio Arcila Flórez

Al revisar la solución que desarrollé para el Reto 3 de la huerta escolar, me di cuenta de que no apliqué completamente el paradigma funcional. Aquí está mi análisis:

- En el código utilicé algunas técnicas funcionales como las list comprehensions para calcular los meses sin gastos y los meses con gastos menores a 100,000.

Algunos puntos de mejora:

1. Estado Mutable:

En la clase `InformacionContable` usé listas que se modifican (`self.gastos_por_mes` y `self.costos_por_mes`). En programación funcional, deberíamos evitar cambiar el estado y en su lugar crear nuevas estructuras de datos inmutables.

2. Efectos Secundarios:

Las funciones tienen muchos efectos secundarios, especialmente en la entrada/salida de datos. Por ejemplo, `ingresar_datos_mensuales()` modifica el estado de la clase y además maneja la entrada de datos.

Las dificultades que enfrenté al pensar en esta solución fueron:

1. Me costó trabajo pensar en cómo manejar la entrada de datos de forma funcional, ya que estamos acostumbrados a modificar variables.
2. No estaba seguro de cómo estructurar el código para mantener la inmutabilidad sin hacer el programa demasiado complejo.
3. Me di cuenta de que necesitaba aprender más sobre conceptos como funtores y mónadas para manejar mejor los efectos secundarios.

Para implementar completamente el paradigma funcional, sugiero:

1. Crear estructuras de datos inmutables usando `dataclasses`
2. Separar la lógica pura de los efectos secundarios (entrada/salida)
3. Usar funciones de orden superior como `map`, `filter` y `reduce`
4. Implementar manejo de errores funcional usando `Either` o `Maybe`