

Reto 4

¿Cómo aplicar el paradigma funcional a la solución en el reto 3?

Juan José Duque CC 1001724543

Alejandro Vargas Ocampo 1088298091

En el código realizado para el reto 3, no se aplicó el paradigma funcional en su totalidad. El código sigue un enfoque imperativo y secuencial, donde las funciones están definidas principalmente para dividir el programa en secciones lógicas y realizar tareas específicas, pero no se sigue el enfoque funcional de la misma manera que se haría en un programa funcional puro. De hecho, de manera inicial en el reto 3 no se planeó resolverlo bajo el paradigma funcional, sino que se buscó una solución secuencial al reto.

A continuación, se presentan algunas observaciones sobre cómo se podría aplicar el paradigma funcional y mejorar el código:

- **Mayor Uso de Funciones:** En un enfoque funcional, se fomenta la creación de funciones puras que no tienen efectos secundarios y que toman argumentos y devuelven resultados. El código podría beneficiarse de más funciones que realicen tareas específicas, como cálculos y operaciones en lugar de realizarlos directamente en el cuerpo del programa.
- **Inmutabilidad:** En programación funcional, se evita la mutabilidad de los datos siempre que sea posible. Esto significa que los valores de las variables no se cambian una vez que se establecen. En el código actual, hay muchas asignaciones directas de variables que podrían reemplazarse por cálculos funcionales.
- **Composición de Funciones:** El código podría utilizar la composición de funciones para realizar cálculos más complejos. Por ejemplo, en lugar de realizar cálculos de costo en una sola función, podrían crearse funciones separadas para calcular diferentes componentes de costos, y luego componerlas para obtener el costo total.
- **Evitar Variables Globales:** En programación funcional, se evita el uso de variables globales. En el código del reto 3, la variable cultivos es global y se modifica directamente. En su lugar, podría considerarse pasar esta variable como argumento a las funciones que la necesitan.

- Manejo de Errores Funcional: En lugar de devolver mensajes de error en forma de cadenas, sería más apropiado manejar los errores utilizando valores especiales o excepciones, lo que es más coherente con un enfoque funcional.
- División en Funciones más Pequeñas: Las funciones podrían dividirse aún más en funciones más pequeñas y especializadas para facilitar la lectura y el mantenimiento del código.
- Uso de Funciones de Orden Superior: En Python, se pueden utilizar funciones de orden superior para simplificar ciertos cálculos o procesamiento. Por ejemplo, las funciones map, reduce, y filter pueden utilizarse en lugar de bucles explícitos para operaciones en listas.
- Evitar Mutación de Variables en Lugar de Reasignación: En lugar de cambiar el valor de una variable existente, se pueden crear nuevas variables con los resultados de las operaciones. Esto es más coherente con un enfoque funcional.

Aplicar completamente el paradigma funcional en este código requeriría una revisión y refactorización significativa. Se podrían crear funciones puras que tomen argumentos, realicen cálculos y devuelvan resultados, evitando efectos secundarios.