

CONCLUSIONES SOBRE EL PARADIGMA FUNCIONAL EN EL RETO 3

Para resolver el reto 3 tuve que usar el paradigma funcional con el objetivo de definir las diferentes partes de la calculadora de gestión de cultivos, ya que el paradigma funcional me da la posibilidad de definir variables que no son locales, así mismo como de integrar aquellas que sí son locales.

Por ejemplo, para acoplar los cultivos hice uso de un diccionario con una longitud de 3 (los tres cultivos que usé por *default*) y gracias al paradigma funcional y al *print* con formato me permitió llamar los valores correspondientes a cada llave en el diccionario de cultivos, así hacer el código más dinámico. También, para poder crear un ciclo infinito en el cuál el menú siempre se estuviera mostrando, usé una función sin argumentos para definir el menú principal y luego con el controlador de flujo *while* pude llamar el menú principal de manera infinita y alternar las demás opciones usando condicionales correspondientes a cada uno de los números que define para las diferentes funciones (Horarios de gestión, etapas, información contable, etc.).

Intenté empezar la calculadora usando condicionales e inputs, pero se me hizo demasiado tedioso, ya que sentía que el menú era demasiado estático y debía definir un sinnúmero de variables locales, lo cuál hace el código muy largo, en ese sentido me frustré mucho. Buscando en la documentación y con ayuda del material de Tema 4, investigué más sobre el uso de funciones para definir pesados de código y luego llamarlos para que retornen algo, luego supe que definiendo funciones vacías que sólo impriman cosas, podía crear un menú principal que en apariencia fuera estático, pero a partir de otras funciones y usando un controlador de flujo podía hacer de este menú principal más dinámico. Para poder usar las definiciones de cada cultivo encontré información sobre el formato en el *print*, que permite llamar variables directamente al *output* sin concatenar ni preocuparse por el tipo de dato (Bueno, al menos en los casos en que lo usé).

Considero que el paradigma funcional es una parte muy importante de la programación y considero que su principal utilidad radica en la posibilidad de elaborar grandes pedazos de código que pueden usarse múltiples veces sólo llamando a la función declarada, sin tener que escribir el código una y otra vez para cada parámetro que se trabaje.

Tuve varias dificultades a la hora de trabajar con el paradigma funcional, principalmente con el *return*, pensaba que siempre debía retornar algo cuando se

usaba una función, pero con la ayuda de la documentación de Python y el material de Ingenia, todo se fue aclarando de a poco.