**Presentado por: Laiz Murillo & Paola Gómez.**

**RETO 4**

**Preguntas de interés:**

* ¿Pueden existir funciones dentro de funciones?

Si se pueden definir funciones dentro de otras funciones. En este caso, cada función tiene sus variables locales y puede acceder tanto a las globales como a las de sus funciones superiores.

* ¿Se puede enviar como parámetro de una función otra función con retorno?

Si se puede enviar como parámetro de una función otra función con retorno. Las funciones son objetos de primera clase y se les puede asignar a otra variable.

* ¿Qué sucede si dentro de una función se hace un llamado a sí misma?

Cuando una función se llama a sí mismo, se asigna espacio en la pila para las nuevas variables locales y parámetros. A esto se le denomina recursividad. Se dice que una función es recursiva cuando el cuerpo de la función utiliza a la propia función, es una técnica de programación que nos permite que un bloque de instrucciones se ejecute n veces.

* ¿Una variable creada como global dentro de una función, ¿sigue existiendo fuera de la función?

### Si, sigue existiendo siempre y cuando esta variable no se modifique. En caso tal, tomaría este valor dentro y fuera. Se coloca global dentro de la función **para modificar la variable global**.

* Después de indicar una variable en una función como global, ¿hay alguna manera de volver a usar la variable local?

Si se utiliza la variable en una función y esta se modifica sin definirla dentro de ella como global, la reasignación simplemente no tiene lugar, se crea una nueva variable local, con el mismo nombre que la global y que la enmascara. Es a esa nueva variable local a la que se le asigna el nuevo objeto. La variable global no se modifica y la local deja de existir nada más retornar la función.

**Punto 1.**

Para la solución del reto 3, no aplicamos el paradigma funcional. Teniendo en cuento el objetivo de las funciones (evitar la repetición de código y promover su reutilización de código). En nuestra solución del reto 3, tenemos varias porciones de código que son muy habituales y conviene abstraerlas en una función. Por lo cual sugerimos realizar las siguientes modificaciones:

1. Para la aplicación del paradigma funcional en el reto 3, se crearán 6 funciones:

def mostrar\_informacion\_contable()# Función para ingresar todos los valores necesarios para realizar los cálculos del informe contable: Gastos, Costos, Producción, Informe económico (llama la función informe\_contable () para realizar los cálculos y mostrarlos como un informe contable, y Menú Principal (llama la función volver() para preguntar si desea o no, volver al menú principal.

def mostrar\_etapa\_cultivo(): # En esta función se encuentran 4 opciones, según la información que desee saber el usuario sobre su cultivo: (Siembra, Crecimiento, Cosecha) y Menú Principal (Pregunta al usuario si desea o no, volver al menú principal.

def mostrar\_gestion\_cultivo(): # En esta función se encuentran 2 opciones. La opción N°1 permite obtener la información sobre los días y horarios de mantenimiento / abono / riego dependiendo el tipo de cultivo que haya seleccionado en un comienzo y Menú Principal (Pregunta al usuario si desea o no, volver al menú principal.

def informe\_contable (): #Función para mostrar el informe contable.

def volver(): #Función por la cual se pregunta al usuario si desea o no, volver al menú principal.

def mostrar\_menu\_principal(): #Esta función permite mostrar la información de cada opción del menú principal según la selección. Llama las otras funciones para mostrar el resultado dependiendo la opción ingresada por teclado y asignada en la variable menu\_principal.

menu\_principal= ' 1. Horario de Gestión de Cultivo 2. Etapas del cultivo 3. Información contable 4. Seleccionar otro tipo de cultivo 5. Salir '

opcion\_menu\_principal=input(menu\_principal.

La estructura de la función principal es la siguiente:

def mostrar\_menu\_principal():

  global tipo\_cultivo,opcion\_menu\_principal,gestion\_cultivo, info\_contable, go\_back

  global info\_contable, mes1\_medicamentos, mes2\_medicamentos, mes3\_medicamentos, mes4\_medicamentos, mes5\_medicamentos, mes1\_imprevistos, mes2\_imprevistos, mes3\_imprevistos, mes4\_imprevistos, mes5\_imprevistos, mes1\_mano\_obra, mes2\_mano\_obra , mes3\_mano\_obra, mes4\_mano\_obra, mes5\_mano\_obra,  mes1\_agua,  mes2\_agua,  mes3\_agua,  mes4\_agua,  mes5\_agua, mes1\_mtto, mes2\_mtto, mes3\_mtto, mes4\_mtto, mes5\_mtto,  mes1\_abono,  mes2\_abono,  mes3\_abono,  mes4\_abono,  mes5\_abono, mes1\_valor\_cultivo, mes2\_valor\_cultivo, mes3\_valor\_cultivo, mes4\_valor\_cultivo, mes5\_valor\_cultivo, mes1\_cantidad, mes2\_cantidad, mes3\_cantidad, mes4\_cantidad, mes5\_cantidad

  opcion\_menu\_principal=input(menu\_principal)

  if opcion\_menu\_principal == '1':

    gestion\_cultivo = mostrar\_gestion\_cultivo()

    print(gestion\_cultivo)

    if go\_back:

      mostrar\_menu\_principal()

  elif opcion\_menu\_principal == '2':

    etapa\_cultivo = mostrar\_etapa\_cultivo()

    print(etapa\_cultivo)

    if go\_back:

      mostrar\_menu\_principal()

  elif opcion\_menu\_principal == '3':

    mes1\_medicamentos, mes2\_medicamentos, mes3\_medicamentos, mes4\_medicamentos, mes5\_medicamentos, mes1\_imprevistos, mes2\_imprevistos, mes3\_imprevistos, mes4\_imprevistos, mes5\_imprevistos, mes1\_mano\_obra, mes2\_mano\_obra , mes3\_mano\_obra, mes4\_mano\_obra, mes5\_mano\_obra,  mes1\_agua,  mes2\_agua,  mes3\_agua,  mes4\_agua,  mes5\_agua, mes1\_mtto, mes2\_mtto, mes3\_mtto, mes4\_mtto, mes5\_mtto,  mes1\_abono,  mes2\_abono,  mes3\_abono,  mes4\_abono,  mes5\_abono, mes1\_valor\_cultivo, mes2\_valor\_cultivo, mes3\_valor\_cultivo, mes4\_valor\_cultivo, mes5\_valor\_cultivo, mes1\_cantidad, mes2\_cantidad, mes3\_cantidad, mes4\_cantidad, mes5\_cantidad, info\_contable = mostrar\_informacion\_contable()

    print(info\_contable)

    print(seleccion\_opc)

    mostrar\_menu\_principal()

  elif opcion\_menu\_principal == '4':

    print(mensaje\_bienvenida)

    print(seleccion\_opc)

    tipo\_cultivo=input(menu\_tipo\_cultivo)

    mostrar\_menu\_principal()

  elif opcion\_menu\_principal == '5':

    close = True #Esta variable se define en un inicio como False. Por fuera de todas las funciones

  return opcion\_menu\_principal

Nota: Realizamos el reto 3 aplicando paradigma funcional, adjunto en el archivo LaizMurillo,PaolaGomez\_reto3funciones.