ACTIVIDAD PRÁCTICA

Reto 5

# Título: Estructura de datos

# Modalidad: equipos de cinco estudiantes

# Resultado de aprendizaje

* El estudiante entiende el concepto de paradigma de programación y reconoce los tipos de paradigmas.
* El estudiante entiende el paradigma funcional y su utilidad en la solución *software* de problemas.
* El estudiante es capaz de diseñar la solución a un problema orientado al paradigma funcional.
* El estudiante es capaz de implementar el paradigma funcional en una solución *software* usando el lenguaje de programación Python.
* El estudiante es capaz de identificar cuándo es necesario o útil el uso del paradigma funcional.

# Recursos

* “Material fundamental”: Paradigma funcional
* Colab del paradigma funcional
* Ejercicios prácticos del “material fundamental”: Paradigma funcional
* Tutorías sincrónicas
* Taller sincrónico

# Indicaciones

* Revise y estudie el “material fundamental”.
* Invierta un poco de tiempo haciendo pruebas en los **ejercicios prácticos.**
* Revise los ejemplos dados por el tutor en el taller.
* Trabaje en equipos de cuatro personas**.**
* Intente responder las siguientes **preguntas de interés** por medio de pruebas que realice usted mismo, debatiéndolas con sus compañeros o investigándolas:
* ¿Pueden existir funciones dentro de funciones?
* ¿Se puede enviar como parámetro de una función otra función con retorno?
* ¿Qué sucede si dentro de una función se hace un llamado a sí misma?
* ¿Una variable creada como global dentro de una función, ¿sigue existiendo fuera de la función?
* Después de indicar una variable en una función como global, ¿hay alguna manera de volver a usar la variable local?
* Si se le dificulta la búsqueda de información para solucionar problemas, puede emplear el foro **Preguntas al tutor**, en el que el tutor o uno de sus compañeros puede ayudarle, o puede haber una pregunta similar a la suya con solución previa que haya logrado resolver la misma duda que usted tiene. Finalmente, el tutor lo guiará y ayudará a resolver su dificultad.
* Lea atentamente el siguiente enunciado del problema e intente reconocer los elementos presentados en el tema “Control de flujo”, que le pueden servir para crear la aplicación (recuerde que una aplicación no es más que el nombre que se le da a un archivo de código que cumple con una tarea).

**RETO**

Desarrolle una aplicación que permita administrar las notas de un grupo de estudiantes.

Primero se debe tener una estructura de datos en la cual se puedan almacenar los siguientes datos de un estudiante:

* Identificación
* Nombre
* Correo
* Teléfono
* Fecha de nacimiento
* Nota 1
* Nota 2
* Nota 3
* Nota 4

En caso de que decida usar como estructura de dato un “diccionario”, defina y escoja el dato más indicado para ser su llave, o defina otro dato como llave nueva.

Cree un menú dentro de una función, el cual imprima las siguientes opciones numeradas para luego recibir por consola la elección del usuario. Cada una de las opciones debe ejecutar el llamado de una función en la que se ejecute la tarea indicada:

* **Agregar:** se debe pedir al usuario que ingrese todos los datos del estudiante (se pueden crear todos los estudiantes que se quiera agregar).
* **Buscar:** se debe pedir al usuario que ingrese un dato con el cual se pueda identificar un estudiante que desea consultar (el dato para buscar lo elige el desarrollador) y verificar si el estudiante existe para luego imprimir toda la información de este.
* **Modificar**: se debe realizar el mismo proceso de búsqueda de la opción “Buscar” y luego se debe pedir al usuario que ingrese todos los nuevos datos para el estudiante; **solo se pueden modificar notas**.
* **Cancelación de materia**: se debe realizar el mismo proceso de búsqueda en la opción “Buscar” y luego se debe confirmar si el usuario quiere eliminar el contacto.
* **Resultados por estudiante**: se debe inicialmente buscar el estudiante del cual se quiere conocer los resultados (se sugiere usar el código usado en la opción “Buscar”), y luego se deben realizar los siguientes cálculos y mostrar los resultados en pantalla.
  + - * + Nota final del estudiante
        + Nota promedio del grupo
        + Informar si el estudiante estuvo por encima o por debajo de la media
        + Informar si el estudiante ganó o perdió el curso
        + Informar en qué percentil se encuentra el estudiante
* **Informe de grupo**: haga los cálculos necesarios y muestre en pantalla los siguientes resultados:
  + - * Nota final por estudiante
      * Nota promedio
      * Número de estudiantes por encima, por debajo e iguales al promedio
      * Número de estudiantes ganadores y perdedores (establezca el valor a partir del cual se gana)
      * Porcentaje de ganadores y porcentaje de perdedores
      * Distribución de estudiantes por percentil
      * Moda
      * Mediana
      * Desviación estándar
* **Salir:** se debe finalizar la ejecución de la aplicación.

Tras finalizar las tareas que correspondan a la opción seleccionada, se debe volver a imprimir el menú principal.

Finalmente, desarrolle una solución al problema creando la aplicación en el lenguaje de programación Python de manera que cumpla con los requerimientos que se piden y siguiendo las indicaciones que se le dan en el enunciado.

# Criterios de valoración de la evidencia

La aplicación funciona (40%).

La aplicación cumple con todos los requerimientos pedidos (40%).

Usa correctamente del paradigma funcional (20%).

# Protocolo de entrega

* **Formato**: Comprima el archivo en .zip con el siguiente nombre:

[NOMBREINTEGRANTE1,NOMBREINTEGRANTE2]\_Reto5.zip

donde [NOMBREINTEGRANTE] es su nombre y apellidos, empezando por mayúscula cada uno. Ej:

AndresRestrepoG,DianaLopez\_Reto5.zip

* **Medio**: suba el archivo .zip en la plataforma de Moodle en la sección correspondiente para la “entrega de trabajos del reto 5”.
* **Plazo máximo de entrega**: de acuerdo con el cronograma de actividades.

PROGRAMA

nota\_para\_aprobar = 7

cantidad\_notas = 7

respuesta = "s"

total\_alumnos = 0

total\_suspendidas = 0

total\_aprobadas = 0

total\_alumnos\_todo\_aprobado = 0

total\_alumnos\_alguna\_suspendida = 0

suma\_notas\_medias\_repetidores = 0

suma\_notas\_medias\_no\_repetidores = 0

alumno\_nota\_media\_mas\_alta = ""

nota\_media\_mas\_alta = 0

total\_repetidores = 0

total\_no\_repetidores = 0

while respuesta.lower() == "s":

identificacion = input("Identificacion: ")

    nombre = input("Nombre: ")

    edad = input("Edad: ")

correo = input("Correo: ")

telefono= input("Telefono: ")

    repetidor = input("Repetidor: [s/n] ")

    nota\_maxima = 0

    nota\_minima = 11

    suma\_notas = 0

    suspendidas = 0

    aprobadas = 0

    nota = -1

    for i in range(cantidad\_notas):

        while nota < 0 or nota > 10:

            nota = float(input("Ingrese la nota #" + str(i+1) + ": "))

        suma\_notas += nota

        if nota > nota\_maxima:

            nota\_maxima = nota

        if nota < nota\_minima:

            nota\_minima = nota

        if nota < nota\_para\_aprobar:

            suspendidas += 1

            total\_suspendidas += 1

        else:

            aprobadas += 1

            total\_aprobadas += 1

        nota = -1

    total\_alumnos += 1

    nota\_media = suma\_notas / cantidad\_notas

    if nota\_media > nota\_media\_mas\_alta:

        alumno\_nota\_media\_mas\_alta = nombre

        nota\_media\_mas\_alta = nota\_media

    if suspendidas <= 0:

        total\_alumnos\_todo\_aprobado += 1

    else:

        total\_alumnos\_alguna\_suspendida += 1

    if repetidor.lower() == "s":

        suma\_notas\_medias\_repetidores += nota\_media

        total\_repetidores += 1

    else:

        suma\_notas\_medias\_no\_repetidores += nota\_media

        total\_no\_repetidores += 1

    print("Estadística de " + nombre)

    if repetidor.lower() == "s":

        print("-Repetidor: sí")

    else:

        print("-Repetidor: no")

    print("-Nota máxima: " + str(nota\_maxima))

    print("-Nota mínima: " + str(nota\_minima))

    valor\_equivalente = ""

    if nota\_media < 5:

        valor\_equivalente = "Insuficiente"

    elif nota\_media < 6:

        valor\_equivalente = "Suficiente"

    elif nota\_media < 7:

        valor\_equivalente = "Bien"

    elif nota\_media < 8:

        valor\_equivalente = "Notable"

    else:

        valor\_equivalente = "Sobresaliente"

    print("-Nota media: " + str(nota\_media) + " ("+valor\_equivalente+")")

    print("-Cuántas suspendidas: " + str(suspendidas))

    print("-Cuántas aprobadas: " + str(aprobadas))

    respuesta = input("Se van a introducir más alumnos? [s/n] ")

nota\_media\_repetidores = 0.0

if total\_repetidores > 0:

    nota\_media\_repetidores = suma\_notas\_medias\_repetidores / total\_repetidores

nota\_media\_no\_repetidores = 0.0

if total\_no\_repetidores > 0:

    nota\_media\_no\_repetidores = suma\_notas\_medias\_no\_repetidores / total\_no\_repetidores

print("Estadística de la clase")

print("-Cuántos alumnos: " + str(total\_alumnos))

print("-Cuántas notas suspendidas: " + str(total\_suspendidas))

print("-Cuántas notas aprobadas: " + str(total\_aprobadas))

print("-Cuántos alumnos con todo aprobado: " + str(total\_alumnos\_todo\_aprobado))

print("-Cuántos alumnos con alguna suspendida: " +

      str(total\_alumnos\_alguna\_suspendida))

print("-Nota media de los repetidores: " + str(nota\_media\_repetidores))

print("-Nota media de los no repetidores: " + str(nota\_media\_no\_repetidores))

print("-Alumno con la nota media más alta: " +

      str(alumno\_nota\_media\_mas\_alta) + " (" + str(nota\_media\_mas\_alta) + ")")