



ISIS-1221

INTRODUCCIÓN A LA PROGRAMACIÓN

Nivel 2 – Laboratorio condicionales

Objetivos

1. Entender la utilidad de los condicionales y cómo aplicarlos a diferentes problemas.
2. Practicar el uso de condicionales para la solución de problemas por casos.

Preparación del ambiente de trabajo

1. Cree una carpeta para trabajar, poniéndole su nombre o login.
2. Descargue de BrightSpace el archivo `consola_condicionales.py` y guárdelo en su carpeta de trabajo. Este archivo es un esqueleto de la consola de su programa, el cual usted debe completar en el punto 4 de este laboratorio.
3. Abra Spyder y cambie la carpeta de trabajo para que sea la carpeta que acaba de crear.
4. Cree un nuevo módulo de funciones llamado `condicionales.py` en el que escribirá las funciones correspondientes a los puntos 1 a 3 de este laboratorio.

Punto 1

Es posible determinar si un año es bisiesto teniendo únicamente el número del año. En este punto usted debe escribir una función llamada `bisiesto` que reciba un año (por ejemplo 1969) e informe (verdadero o falso) si el año es bisiesto o no.

```
def bisiesto(anio: int) -> bool:
    """Retorna True si anio es un año bisiesto y False de lo contrario"""
```

Ayuda: Sabiendo que los años bisiestos ocurren cada cuatro años, el siguiente algoritmo sirve para determinar si un año es bisiesto o no:

- (1) Si un año **NO** es divisible por 4 entonces **NO** es bisiesto.
- (2) De lo contrario, si un año **NO** es divisible por 100 **SÍ** es bisiesto.
- (3) De lo contrario, si un año **NO** es divisible por 400 **NO** es bisiesto.
- (4) Si ninguna de las condiciones anteriores se cumple, **SÍ** es bisiesto.

Punto 2

Escriba una función `clasificar` que dados los tres ángulos de un triángulo (en grados), determine si es el triángulo es equilátero, isósceles, o escaleno.

Recuerde que un triángulo es equilátero si la longitud de sus lados es igual, isósceles si dos de sus lados son iguales, y escaleno si todos sus lados son de longitudes diferentes.

```
def clasificar(a1:float, a2:float, a3:float) -> str:
    """Retorna "Equilatero si el triángulo es equilatero, "Isósceles" si
    es isósceles y "Escaleno si es escaleno"""
```

Ayuda: Piense de qué manera se relacionan los lados de los triángulos con sus ángulos.

Punto 3

Escriba una función `solucionar` que resuelva ecuaciones cuadráticas usando condicionales. En caso de que no haya una solución retorne la cadena “no tiene solución”.

Recuerde que la fórmula cuadrática es $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

```
def solucionar(a:float, b:float, c:float) -> str:
    """Retorna una cadena con la(s) soluciones de la ecuación o una cadena
    diciendo que no tiene solución."""
```

Punto 4

Abra el archivo `consola_funciones.py` y estudie cuidadosamente su contenido. Ejecútelo y asegúrese de haber comprendido el rol de las siguientes funciones: `iniciar_aplicacion`, `mostrar_menu`, `ejecutar_bisiesto`, `ejecutar_clasificar` y `ejecutar_solucionar`. Ahora proceda a completar este archivo de interfaz por consola, siguiendo las marcas `#TODO` presentes en el código.

Entrega

Cree un archivo `.zip` con los 2 archivos en los que trabajó: `condicionales.py` y `consola_condicionales.py`. Cambie el nombre del archivo `zip` para que tenga la forma `n2-l1-login.zip` (donde `login` es su usuario Uniandes) y finalmente entréguelo en BrightSpace en el laboratorio del Nivel 2 designado como “**L1: Condicionales**”.