

# Tarea 2

April 25, 2022

## 1 Tarea #2

### 1.0.1 Universidad de Costa Rica

Escuela de Ciencias de la Computación e Informática

CI0202 - Principios de Informática. Grupos G02,G13

Prof. Jose Pablo Ramírez Méndez

**I-2022 Objetivo:** Repasar, mediante ejercicios prácticos, los conceptos estudiados en clase de: Entrada y Salida de Datos con manejo de excepciones y Operadores. **Formato:** Individual

**Entregar:** Debe entregar su código fuente ya sea en 1 archivo por cada ejercicio con extensión ( .py ) o 1 cuaderno de Jupyter para todos lo ejercicios de la tarea ( Puede utilizar este como base para su entregable )

- En esta asignación, se debe resolver los ejercicios mencionados posteriormente. Para cada enunciado distinto, se deberá entregar un archivo de código fuente en Python 3 que lleve a cabo la tarea especificada (con extensión .py). Puede también y se recomienda entregar el laboratorio en un cuaderno de Jupyter.
- No olvide incluir su **nombre completo y carnet** en su entrega, ya sea como comentarios de códigos o bien, como celdas de textos si decide utilizar cuadernos de Jupyter.
- La tarea es de elaboración y entrega individual. No se aceptarán entregas después de la fecha y hora límite. Tareas elaboradas en grupos serán invalidadas; y serán considerados como **plagio** aquellas que sean idénticas o casi idénticas.
- No olvide aplicar cada una de las etapas de resolución de problemas vista en clase. Se recomienda que en la fase de Diseño e implementación, escriba su algoritmo en pseudocódigo como comentarios de línea en su código.

### 1.0.2 Evaluación

Para cada ejercicio, se evaluará lo siguiente: - **Resultado esperado:** 25% - Lo que el programa realiza coincide con el enunciado. Para una entrada tal, el programa produce el resultado esperado. El programa solicita los datos adecuados y muestra los datos adecuados. - **Resolución del problema:** 65% - Internamente, el programa resuelve o intenta resolver el problema estatado en el enunciado. Esto significa que, aunque su programa tenga errores de cálculo o formato, si se entiende bien lo que están tratando de hacer, los puntos de esta sección serán otorgados. - **Buen**

**formato:** 10% - Utilice nombres de variables significativos (por ejemplo “coordX” lugar de “x” o “nombre” en lugar de “n”). Haga manejo de entrada y salida con texto descriptivo. Muestre la cantidad de decimales correcta.

Cada ejercicio tiene un puntaje asignado. La nota final es la sumatoria de multiplicar el puntaje de cada ejercicio por la nota obtenida en ese ejercicio, por un máximo de 100 puntos.

### Puntos extra

Para cada problema planteado en la tarea adjunte el resultado de la etapa de *Diseño* del flujo del proceso resolución de problemas visto en clase como algoritmo en pseudocódigo. Puede agregarlo como comentarios de código al principio de su implementación o bien, como un documento a parte para todos los ejercicios de la tarea. Puede obtener un máximo de 10 puntos. Por ejemplo, si obtiene un 95 en la nota de la tarea, pero realizó los diseños correspondientes y corresponden **directamente** a la solución implementada, obtiene 10 puntos y su nota sube a 105. Cada diseño tiene valor proporcional a los puntos del ejercicio respecto al máximo de 10 puntos mencionado anteriormente.

### 1.0.3 Enunciado

Para cada uno de los siguientes ejercicios, elabore un programa que lo resuelva.

### 1.0.4 Ejercicio 1. Valor 50 pts

**Aproximación de la función seno** Dado un ángulo en radianes  $x$ , el valor del seno puede aproximarse mediante la siguiente sucesión de Taylor en los primeros 4 términos:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \quad (1)$$

Escriba un programa al que se le ingrese un ángulo **en grados** y calcule la aproximación de su seno mediante el polinomio de Taylor.

Dado que los resultados de este polinomio de Taylor solo ofrecen una aproximación al valor real de la función seno, su programa debe calcular el error en la aproximación. Dicho error está definido como: `valor_real - valor_aprox`, donde el `valor_real` es la imagen de la función seno para el grado  $x$  y el `valor_aprox` el resultado de la evaluación de  $x$  en el polinomio. Utilice exactamente 6 decimales para mostrar el resultado de la aproximación, el valor real y el error.

Esta aproximación es válida para valores de  $-180$  a  $180$  (o  $-\pi$  a  $\pi$ ). Para valores fuera de este rango, va a topar con errores muy grandes. Muestre además un mensaje que indique si el error a mostrar es muy grande dado el valor del ángulo  $x$ .

**Nota:** En Python, el seno se utiliza con `sin(ángulo)`. Para utilizarlo, debe agregar al inicio de su programa la línea `from math import sin`. Lo mismo para `pi` (lo ocupará para convertir de grados a radianes) y `factorial (!)`:

```
[1]: from math import pi, sin

sin_real = sin(pi/2)
print(sin_real)
```

1.0

```
[2]: from math import factorial
      resultado = factorial(12)
      print(resultado)
```

479001600

**Ejemplo 1:** Para un ángulo de 90 grados, su aproximación del seno es 1.000004, y el error de aproximación es -0.000004 (ya que el seno de 90 es 1).

Ingrese un ángulo en grados: 90

Seno aproximado: +1.000004

Seno real: +1.000000

Error de aproximación: -0.000004

El error para el ángulo 1.570796 puede ser demasiado grande?: False

**Ejemplo 2:** Para la lectura de un valor incorrecto, la excepción es atrapada de la siguiente manera:

Ingrese un ángulo en grados: un ángulo

Error, ángulo inválido.

### 1.0.5 Ejercicio 2. Valor 50 pts

**Medición de PH** Con la Revolución Industrial, empezó el uso del carbón como fuente de energía. Incluso en la actualidad, el carbón es fuente de electricidad en varios países avanzados. Desde mediados del siglo XIX, se sabe que el oxígeno usado en el proceso de quema, se combina con el carbono y el azufre del carbón, produciendo dióxido de carbono y dióxido de azufre. Cuando estos gases ascienden a la atmósfera, el dióxido de azufre se combina con el vapor de agua y el oxígeno en el aire, formando ácido sulfúrico. De este se separan iones de hidrógeno, que son los que cambian el nivel de acidez de la tierra y el agua cuando caen por medio de la lluvia.

La acidez de los bosques y lagos se mide en pH utilizando la fórmula:

$$pH = -\log_{10}(\text{concentración de iones de hidrógeno})$$

La concentración de iones de hidrógeno típicamente se mide en *mol/L*. Un nivel de pH de 7 es neutral, inferior a 7 indica presencia de un ácido, y superior a 7 indica presencia de un alcalino. Note que en el caso de una concentración de 0 (o negativa), el logaritmo se indefine. En ese caso, **asuma un pH de 14**.

Escriba un programa que solicite una concentración de iones de hidrógeno en *mol/L* y calcule el pH correspondiente. Además, muestre si es Ácido, Alcalino, o Neutral.

**Nota:** En Python, el logaritmo base 10 se utiliza con `log10(numero)`. Para utilizarlo, debe agregar al inicio de su programa la línea `from math import log10`. Además, muestre el resultado del nivel de pH exactamente con un decimal.

**Ejemplo 1:**

Ingrese la concentración de iones de hidrógeno (mol/L): 0.000001

Nivel de pH: 6.0

Es Alcalino: False

Es Neutral: False  
Es Ácido: True

### Ejemplo 2:

Ingrese la concentración de iones de hidrógeno (mol/L): 0

Nivel de pH: 14.0  
Es Alcalino: True  
Es Neutral: False  
Es Ácido: False

### Ejemplo 3:

Ingrese la concentración de iones de hidrógeno (mol/L): entrada incorrecta

Nivel de pH: 14.0  
Es Alcalino: True  
Es Neutral: False  
Es Ácido: False

```
[15]: from math import log10  
      print(log10(90))
```

1.954242509439325

**Nota final:** No olvide hacer manejo de excepciones para la entrada. Si se ingresa algo que no es un número al programa, asuma un valor por defecto de 0 u otro apropiado.

Ejercicio original del profesor Leonardo V.