

## Práctica 3

Trabaje los problemas en archivos por separado de acuerdo con el problema planteado. **Para todos los casos, emplee manejo de errores y funciones según corresponda.**

### Manejo de Errores:

#### Problema 1:



Implemente un programa que solicite al usuario una fracción, con formato X/Y, donde cada uno de X e Y es un número entero, y luego muestra, como un porcentaje redondeado al número entero más cercano, donde se indicará la cantidad de combustible en el tanque. Se debe tener en cuenta los siguientes casos:

- Colocar E en caso X/Y sea menor a 1% del total
- Colocar F en caso X/Y sea mayor a 99%.
- En otro caso, devolver el valor en porcentaje %

También debe tomar en cuenta los siguientes casos:

- X y Y deben ser números enteros
- X debe ser menor o igual a Y, y Y != 0

De no cumplirse estos casos, se debe volver a preguntar al usuario. Asegúrese de detectar cualquier excepción como ValueError o ZeroDivisionError.

#### Ejemplos:

- Input: 4/0 Acción: Volver a preguntar al usuario dada la excepción ZeroDivisionError
- Input 1.5/3 Acción: Error dado que solo se permiten números enteros ValueError
- Input 5 / 4 Acción: Volver a preguntar al usuario
- Input 3 / 4 Output: 75%
- Input 4/4: Output F

**Nota:** Le será de utilidad aplicar

```
try:
    ...
except ValueError:
    ...
except ZeroDivisionError:
    ...
```

#### Problema 2:

Cree un programa que solicite al usuario una lista de calificaciones separadas por comas. Divida la cadena en calificaciones individuales y almacénelas en una lista para luego convertir cada calificación en un entero. Deberá utilizar una sentencia try/except para informar al usuario

cuando los valores introducidos no puedan ser convertidos debido a un error de tipeo o formato. (Los métodos de cadena le serán de utilidad)

## **POO:**

Material Complementario POO [enlace](#)

### **Problema 3:**

Definir una clase llamada "CIRCULO" la cual contenga un atributo inicializador **radio**. La clase "CIRCULO" debe tener un método que puede calcular el área en utilizando el atributo radio. Cree 2 objetos de tipo circulo y calcule su área.

### **Problema 4:**

Definir una clase llamada "RECTANGULO" que puede ser construida por los atributos largo y ancho. La clase "RECTANGULO" debe tener un método que puede calcular el área utilizando los atributos de la clase. Además cree una clase CUADRADO que heredé de rectangulo. Cree un objeto de tipo rectangulo y 1 de tipo cuadrado.

### **Problema 5:**

Cree una clase Alumno e inicialícela con el nombre y el número de registro. Haga los métodos para:

1. **Display** - Debe mostrar toda la información del estudiante (nombre y número de registro).
2. **setAge** - Debe asignar la edad al estudiante
3. **setNota** - Debe asignar las notas al estudiante.

## **Librerías:**

### **Problema 6:**

Empleando la API de SUNAT vista en clase, debemos obtener los diferentes valores para el tipo de cambio durante el año 2025 hasta donde se tenga información. Una vez realizado ello calcular:

- Obtener las fechas donde el valor de compra del dólar sea el mínimo.
- Obtener las fechas donde el valor de venta del dólar sea máximo.
- Obtener aquellas fechas donde el valor de la diferencia de compraventa sea máxima.

EJEMPLO API:

<https://api.apis.net.pe/v1/tipo-cambio-sunat?month=5&year=2025>

DOCUMENTACIÓN API:

<https://apis.net.pe/api-tipo-cambio.html>

Para el manejo de excepciones se puede realizar lo siguiente:

```
import requests

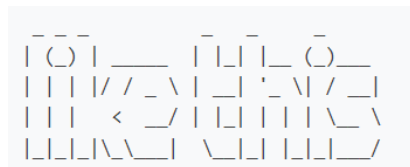
try:
    ...
except requests.RequestException:
    ...
```

Recuerde instalar la librería Requests mediante el comando: `pip install requests`

### Problema 7:

FIGlet, llamado así por las cartas de Frank, Ian y Glen, es un programa de principios de la década de 1990 para hacer letras grandes a partir de texto ordinario, una forma de arte ASCII:

- En la siguiente web puede ver una lista de fuentes admitidas por FIGlet [figlet.org/examples.html](http://figlet.org/examples.html)
- Desde entonces, FIGlet ha sido portado a Python como un módulo llamado pyfiglet.



Cree un programa el cual cumpla con las siguientes especificaciones:

- Solicite al usuario el nombre de una fuente a utilizar. En caso no se ingrese ninguna fuente, su programa deberá seleccionar de forma aleatoria la fuente a utilizar.
- Solicite al usuario un texto.
- Finalmente, su programa deberá imprimir el texto solicitado usando la fuente apropiada.

### Notas:

- Instalar la librería usando: `pip install pyfiglet`
- Para usar la librería, debe hacer:  

```
from pyfiglet import Figlet
figlet = Figlet()
```
- Puede obtener la lista de fuentes disponibles usando: `figlet.getFonts()`
- Para seleccionar el fondo a utilizar emplee: `figlet.setFont(font=fuente_seleccionada)`
- Finalmente podrá imprimir el texto usando : `print(figlet.renderText(texto_imprimir))`
- Recuerde que random tiene un método random choice

### Ejemplo:

- Con una fuente aleatoria y un texto "hello, world" podríamos obtener



- Ingresando la fuente "rectangles" y texto "hello, world" obtenemos



### Problema 8:

Del siguiente URL  
<https://images.unsplash.com/photo-1546527868-ccb7ee7dfa6a?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D>

Descargue la imagen que más le agrade, según lo revisado en la clase. Posteriormente crear un programa que permita el almacenamiento de la imagen como un archivo zip. Finalmente cree un código que permita hacer un unzip al archivo zipeado.