

# Python 2025 - Actividad 2

## Objetivos

- Resolver problemas más complejos utilizando estructuras de datos adecuadas.
- Aplicar correctamente el uso de Jupyter Notebook.
- Desarrollar soluciones considerando dependencias y buenas prácticas.

## Recomendación

- Realizar la práctica en un entorno de **Jupyter Notebook**.

## Ejercicios

### 1. Análisis del Zen de Python

Copie el texto del Zen de Python en una variable e imprima todas las líneas cuya segunda palabra comience con una vocal (A, E, I, O, U, a, e, i, o, u).

*El Zen de Python es una colección de principios que guían la filosofía de diseño de Python, escrita en forma de aforismos. Resalta la simplicidad, legibilidad y claridad en el código.*

Para verlo en Python, ejecute en la consola `import this`

Para este punto debe colocar el contenido del Zen de python en una variable string:

```
zen_text = """Colocar aquí el Zen de Python"""
```

**Salida esperada:**

```
Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
Complex is better than complicated.  
... (otras líneas que cumplan la condición)
```

### 2. Título más largo

Dado un listado de títulos de streams en Twitch:

```
titles = [  
    "Speedrun de Super Mario en tiempo récord",
```

```
"Charla sobre desarrollo de videojuegos",  
"Jugando al nuevo FPS del momento con amigos",  
"Música en vivo: improvisaciones al piano"  
]
```

Encuentre el título con más palabras y muéstrelo en pantalla.

**Salida esperada:**

```
El título más largo es: "Jugando al nuevo FPS del momento con amigos"
```

### 3. Código de conducta

Dado un código de conducta para un servidor de Discord:

```
rules = """Respetar a los demás. No se permiten insultos ni lenguaje  
ofensivo.  
Evitar el spam. No publiques enlaces sospechosos o repetitivos.  
No compartas información personal.  
Usa los canales adecuados para cada tema.  
Sigue las instrucciones de los moderadores."""
```

Solicite una palabra clave al usuario e imprima todas las reglas que la contengan.

**Salida esperada:**

```
Ingrese una palabra clave: moderadores  
Sigue las instrucciones de los moderadores.
```

### 4. Validación de nombre de usuario

Valide un nombre de usuario con los siguientes criterios:

- Al menos 5 caracteres.
- Contiene al menos un número.
- Contiene al menos una letra mayúscula.
- Solo puede contener letras y números.

**Ejemplo de entrada y salida esperada:**

```
Ingrese un nombre de usuario: Gamer123  
El nombre de usuario es válido.
```

```
Ingrese un nombre de usuario: gamer
```

```
El nombre de usuario no cumple con los requisitos.
```

## 5. Clasificación de velocidad de reacción en un juego

Dado el tiempo de reacción de un jugador en milisegundos, clasifíquelo en las siguientes categorías:

- Menos de 200 ms: Rápido
- Entre 200 y 500 ms: Normal
- Más de 500 ms: Lento

**Ejemplo de salida esperada:**

```
Ingrese su tiempo de reacción en ms: 320
```

```
Categoría: Normal
```

## 6. Análisis de descripciones de una plataforma de stream

Dadas varias descripciones de streams en Twitch, cuente cuántas menciones hay de "entretenimiento", "música" y "charla".

```
descriptions = [  
    "Streaming de música en vivo con covers y composiciones",  
    "Charla interactiva con la audiencia sobre series y películas",  
    "Jugamos a juegos retro y charlamos sobre su historia",  
    "Exploramos la mejor música de los 80s y 90s",  
    "Programa de entretenimiento con noticias y curiosidades del mundo  
gamer",  
    "Sesión de charla con invitados especiales del mundo del streaming",  
    "Música en directo con improvisaciones y peticiones del chat",  
    "Un espacio para charlar relajada sobre tecnología y cultura digital",  
    "Exploramos el impacto de la música en los videojuegos clásicos"  
]
```

**Salida esperada:**

```
Menciones de 'música': 4
```

```
Menciones de 'charla': 2
```

```
Menciones de 'entretenimiento': 1
```

## 7. Generador de códigos de descuento

Genere un código de descuento aleatorio para un usuario en base a su nombre, la fecha actual y el resto deben ser números o letras aleatorias. El código debe tener una longitud de 30 caracteres, todas las letras deben ser mayúsculas.

El usuario debe ingresarse por teclado y debe validar que no exeda los 15 caracteres.

#### Entrada:

Usuario: StreamerPro

Fecha: 2025-04-10

#### Salida esperada:

Código de descuento: STREAMERPRO-20250410-AB12CDECA

## 8. Identificación de anagramas

Determine si dos palabras ingresadas son anagramas (contienen las mismas letras en diferente orden).

#### Ejemplo de entrada y salida esperada:

Ingrese la primera palabra: amor

Ingrese la segunda palabra: roma

Son anagramas.

Ingrese la primera palabra: python

Ingrese la segunda palabra: java

No son anagramas.

## 9. Limpieza de Datos para un Sistema de Clientes

Eres un analista de datos en una empresa que ha recibido una lista de nombres de clientes de distintas fuentes.

Sin embargo, estos datos contienen errores, espacios en blanco, nombres repetidos y valores nulos.

Tu tarea es desarrollar un sistema que limpie y estandarice estos datos para poder usarlos en el sistema de facturación.

#### Datos de entrada

La empresa ha recopilado nombres de clientes desde diferentes formularios, bases de datos y registros manuales.

La lista de datos inicial es la siguiente:

```

clients = [
    " Ana López ", "Pedro Gómez", "maria martínez", "Pedro Gómez ", "",
    " Luis Rodríguez ", None, "ana lópez", "JUAN PÉREZ", "MARTA SUÁREZ",
    "luis rodríguez", "maría martínez ", " claudia torres", "CLAUDIA
TORRES",
    " ", "pedro gómez", "Juan Pérez", None, "Ricardo Fernández", "LAURA
RAMOS",
    "carlos mendes", "RICARDO FERNÁNDEZ ", " Laura ramos", "CARLOS MENDES",
    "alejandro gonzález", " ALEJANDRO GONZÁLEZ ", "Patricia Vega",
    "patricia VEGA", "Andrés Ocampo", " andrés ocampo", "Monica Herrera",
    "MONICA HERRERA ", "gabriela ruíz", "Gabriela Ruíz", "sandra morales",
    "SANDRA MORALES", "miguel ángel", "Miguel Ángel ", " Damián Castillo",
    "Damián Castillo ", None, "", " "
]

```

es desarrollar un sistema que permita realizar las siguientes operaciones en los datos:

- **Eliminar espacios extra** en los nombres.
- **Convertir todos los nombres a formato de título** (primera letra en mayúscula y el resto en minúscula).
- **Eliminar registros duplicados** para evitar clientes repetidos.
- **Eliminar valores vacíos o nulos**, ya que no aportan información válida.
- **Mostrar la lista limpia de clientes listos para usar en el sistema de facturación.**

### Salida esperada

Después de ejecutar las funciones de limpieza, la lista de clientes debería verse así:

Lista limpia de clientes al realizar todas las operaciones:

```

['Alejandro González', 'Ana López', 'Andrés Ocampo', 'Carlos Mendes',
'Claudia Torres', 'Damián Castillo', 'Gabriela Ruíz', 'Juan Pérez',
'Laura Ramos', 'Luis Rodríguez', 'Maria Martínez', 'Miguel Ángel',
'Monica Herrera', 'Patricia Vega', 'Pedro Gómez', 'Ricardo Fernández',
'Sandra Morales']

```

## 10. Simulación de partidas y ranking en shooter online

Simule varias partidas de un juego de disparos y genere un ranking basado en el puntaje total de cada jugador. Se utilizará el siguiente sistema de puntuación:

Acción	Puntos
Kill	3
Asistencia	1

Acción	Puntos
Muerte	-1

Dado el siguiente conjunto de rondas:

```

rounds = [
    {
        'Shadow': {'kills': 2, 'assists': 1, 'deaths': True},
        'Blaze': {'kills': 1, 'assists': 0, 'deaths': False},
        'Viper': {'kills': 1, 'assists': 2, 'deaths': True},
        'Frost': {'kills': 0, 'assists': 1, 'deaths': False},
        'Reaper': {'kills': 1, 'assists': 1, 'deaths': False}
    },
    {
        'Shadow': {'kills': 0, 'assists': 2, 'deaths': False},
        'Blaze': {'kills': 2, 'assists': 0, 'deaths': True},
        'Viper': {'kills': 1, 'assists': 1, 'deaths': False},
        'Frost': {'kills': 2, 'assists': 1, 'deaths': True},
        'Reaper': {'kills': 0, 'assists': 1, 'deaths': False}
    },
    {
        'Shadow': {'kills': 1, 'assists': 0, 'deaths': False},
        'Blaze': {'kills': 2, 'assists': 2, 'deaths': True},
        'Viper': {'kills': 1, 'assists': 1, 'deaths': True},
        'Frost': {'kills': 0, 'assists': 1, 'deaths': False},
        'Reaper': {'kills': 1, 'assists': 1, 'deaths': False}
    },
    {
        'Shadow': {'kills': 2, 'assists': 1, 'deaths': False},
        'Blaze': {'kills': 1, 'assists': 0, 'deaths': True},
        'Viper': {'kills': 0, 'assists': 2, 'deaths': False},
        'Frost': {'kills': 1, 'assists': 1, 'deaths': True},
        'Reaper': {'kills': 1, 'assists': 1, 'deaths': False}
    },
    {
        'Shadow': {'kills': 1, 'assists': 2, 'deaths': True},
        'Blaze': {'kills': 0, 'assists': 1, 'deaths': False},
        'Viper': {'kills': 2, 'assists': 0, 'deaths': True},
        'Frost': {'kills': 1, 'assists': 1, 'deaths': False},
        'Reaper': {'kills': 1, 'assists': 1, 'deaths': True}
    }
]

```

Imprima las tablas de resultados luego de la inserción de cada una de las rondas pudiendo ver el progreso en el equipo.

Además cada ronda debe tener un MVP (Mejor Jugador del Partido/ronda) basado en su puntuación. La cantidad de veces que el jugador ha sido MVP también se debe contabilizar

Se debe imprimir el total de kills, asistencias, muertes, cantidad de MVP y puntos totales. La tabla tienen que estar en orden decreciente por los puntos totales.

### Salida esperada:

Ranking ronda 1:

...

Ranking ronda 2:

...

Ranking final:

Jugador	Kills	Asistencias	Muertes	MVPs	Puntos
Shadow	6	6	2	2	22
Blaze	6	3	3	1	18
Viper	5	6	3	1	18
Reaper	4	5	1	0	16
Frost	4	5	2	1	15

*Nota: No hace falta que la tabla se vea exactamente igual. Lo importante es que tengas los valores correctos y se entienda.*

## Entrega

Se deberá entregar como mínimo el **ejercicio 10** completo con las tablas de resultados y los MVPs.

Los ejercicios a entregar (y el resto que realice), deberá poder ejecutarse desde un notebook dentro del proyecto de la entrega en una carpeta de nombre `notebooks`.

El código que tenga las funciones necesarias para la resolución deberán estar dentro del directorio `src`.

Se debe también crear un `README.md` que explique como instalar las dependencias y ejecutar el programa.

**Se explicará con más detalle esta estructura de directorio en la explicación práctica.**

Además, deberá entregar la grabación de un video mostrando la ejecución del ejercicio y luego haciendo un repaso rápido del código y que hace cada parte. El video debe tener una duración aproximada de 5 minutos.

# Pautas

- **Puntos:** 15.
- **Fecha límite de entrega:** Viernes, 4 de abril de 2025, 23:59 hs
- **Modalidad de entrega:** Copie el enlace de su repositorio y del video remoto con la resolución en la tarea de Cátedras. Controle el acceso a los enlaces para que el ayudante pueda acceder a ellos.