

SEMINARIO DE LENGUAJES

Opción Python

Práctica 2 - 2019

1. Dada una lista con nombres de colores

```
colores = ['azul', 'amarillo', 'rojo', 'blanco', 'negro']
```

y una lista con coordenadas

```
coordenadas = [(2, 3), (5, 6), (8, 8), (10, 2), (-5, -8)]
```

- Generar una estructura que contenga coordenadas y un color asociado. La forma de asociar las coordenadas con el color debe ser aleatoria sin importar que se repitan los colores elegidos.
 - Generar una estructura que contenga coordenadas y un color asociado. La forma de asociar las coordenadas con el color debe ser aleatoria sin que se repitan los colores.
2. Usando el diccionario del Ejercicio 1, acceder a las coordenadas y, según el color asociado, ejecute una función, donde las funciones planteen la resolución de ejercicios simples como ser:
 - (a) Suma de dos números que se generen en forma aleatoria cada vez que se llama a la función, reciba el resultado por teclado y verifique el resultado.
 - (b) Dada la estructura que contiene palabras clasificadas según su acentuación:

```
palabras = [('grave', ['molesto']), ('aguda', ['ratón']),  
            ('esdrujula', ['murciélagos'])]
```

cada vez que se ejecuta la función elija una palabra en forma aleatoria, consulte al usuario sobre el tipo que es por su acentuación y verifique la respuesta.

3. Utilizar como estructura de datos de referencia la generada en el Ejercicio 7 de la Práctica 1 y generar funciones que ejecuten lo siguiente:
 - (a) Imprimir los 10 primeros puntajes de la estructura.
 - (b) Imprimir los datos de los usuarios ordenados alfabéticamente por apellido.
 - (c) Imprimir los datos de los usuarios ordenados por nivel alcanzado.
4. Imprima un listado de los archivos que contiene el directorio actual. La información a mostrar deberá ser nombre de archivo, tamaño y fecha de su último acceso. Tener en cuenta que funcione en cualquier plataforma.
5. Realice las siguientes operaciones utilizando funciones **lambda** en el listado de archivos del ejercicio anterior:
 - (a) Obtener una lista de los archivos del directorio con la primer letra del nombre de cada uno en mayúscula
 - (b) Generar una lista con aquellos archivos que han sido modificados en los últimos 3 días
 - (c) Imprimir el tamaño total de los archivos
 - (d) Ordenar los archivos del directorio por tamaño

6. Dada la frase del ejercicio 1 de la Práctica 1, imprima sus 3 palabras más utilizadas mediante la clase Counter del módulo collections.

Pysimplegui

7. Realice una interfaz gráfica con PySimpleGUI para el ejercicio 7 de la práctica 1. La interfaz debe contener 3 campos inputs para los siguientes datos:
- nombre
 - apellido
 - puntaje

La interfaz debe mostrar además la cantidad de jugadores agregados. Ver figura 1.

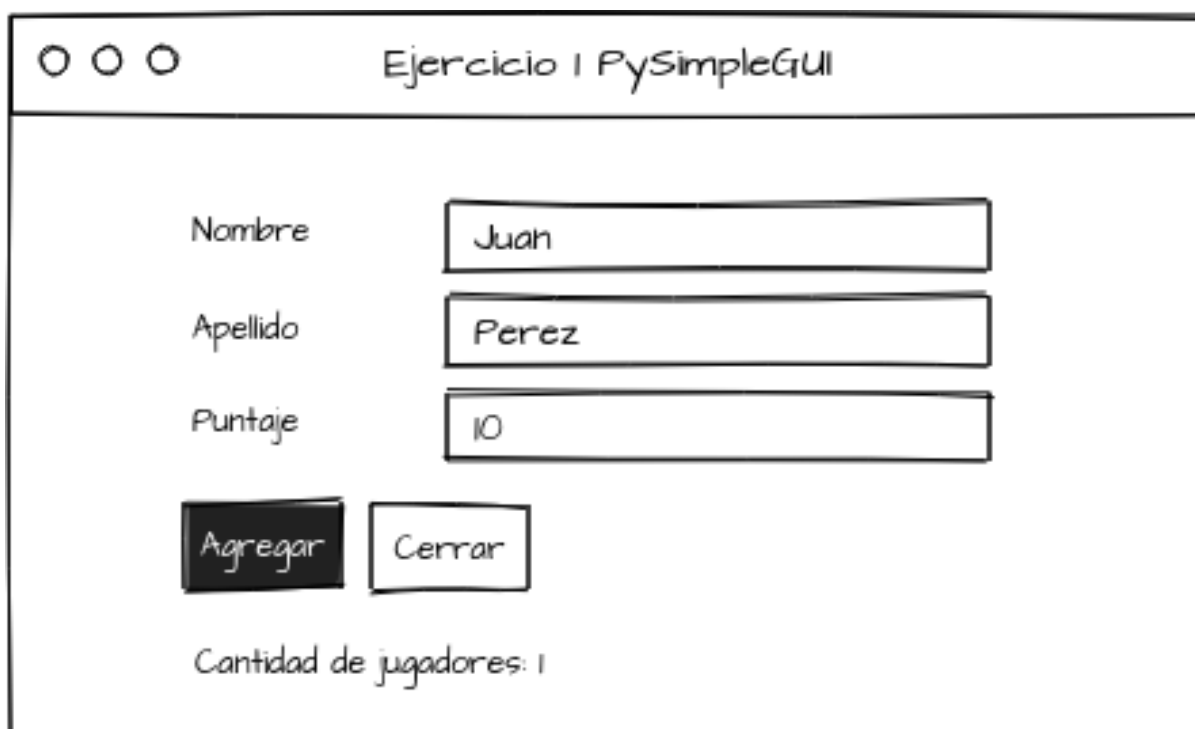


Figura 1: Ejemplo de la interfaz solicitada 7

8. Agregar a la interfaz realizada en el ejercicio 7 un componente que permita mostrar los nombres y puntajes de los jugadores que han sido agregados. Los datos en el componente deben actualizarse a medida que se van agregando nuevos jugadores. Ver figura 2.
9. Modificar la interfaz realizada en el ejercicio 8 de modo que cuando se agregue un jugador nuevo, el componente los muestre en forma ordenada según el puntaje, de menor a mayor.

Pattern

En esta práctica vamos a trabajar con el módulo pattern.es que permite trabajar con el análisis sintáctico del idioma.

Para más información sobre este módulo en [español](#) y en [inglés](#).

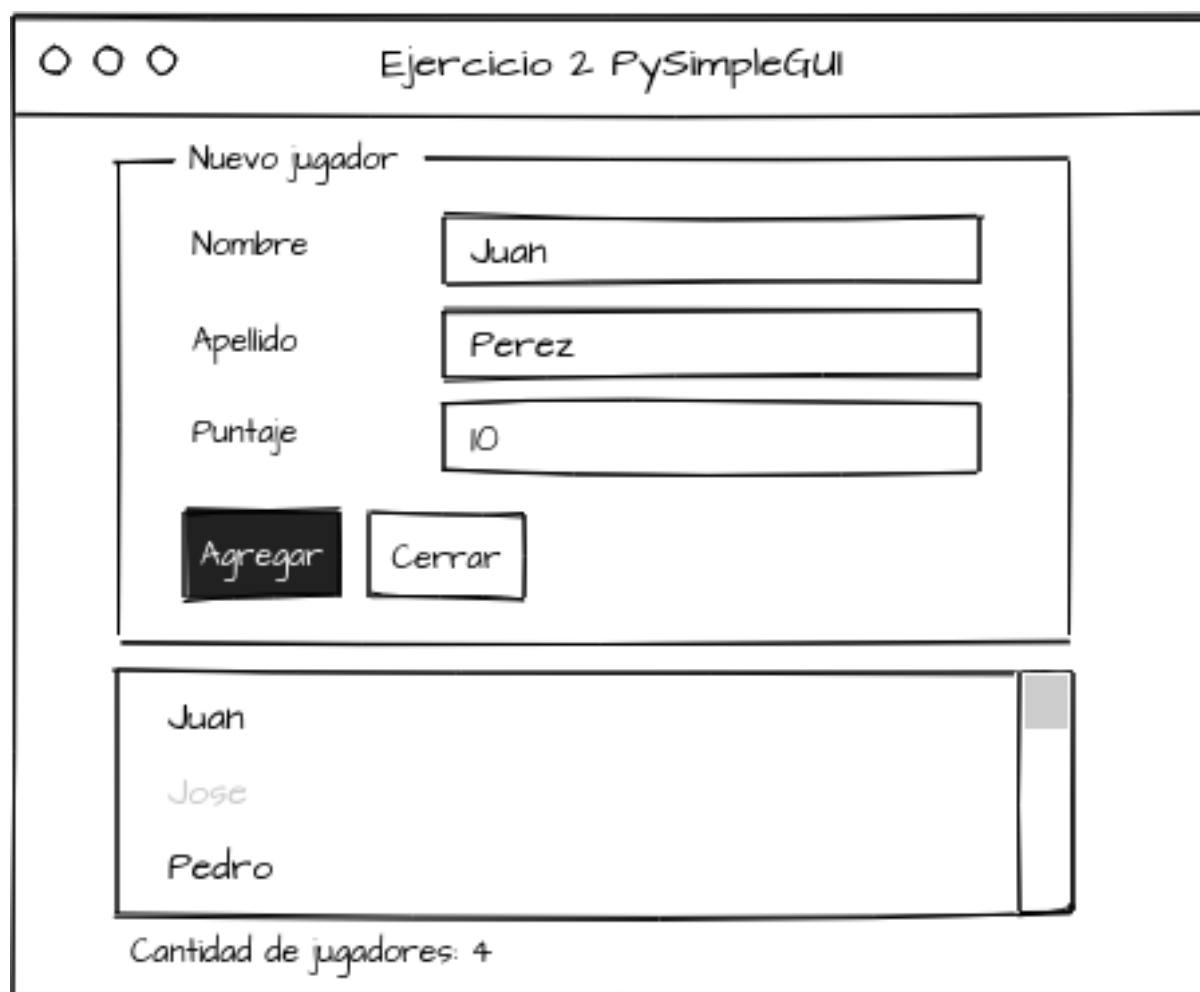


Figura 2: Ejemplo de la interfaz solicitada en 8

10. Defina una función denominada **convertir** que reciba un diccionario con dos keys: 's' y 'p'. Donde 's' indica que la lista asociada contiene palabras en singular y 'p' indica que la lista asociada contiene palabras en plural. La función debe retornar un diccionario con los sustantivos invertidos, es decir, los de singular en plural y los de plural en singular. Nota: utilice el módulo **pattern.es**.

Ejemplo:

```

cambiar = {'s': ['gato', 'caballo', 'silla'], 'p': ['informaticas', 'psicologas', 'ingenieras']}

devuelve:
{'p': ['informatica', 'psicologa', 'ingeniera'], 's': ['gatos', 'caballos', 'sillas']}

```

11. Utilizando el módulo **pattern.es** defina una función denominada **verbosInfinitivos** que reciba un string, el cual puede contener varias oraciones y devuelva una lista de los verbos en infinitivo.
12. Dado un párrafo ingresado por teclado, imprima los 3 verbos más usados en él (en infinitivo, del más común al menos común), junto con la cantidad de apariciones de cada uno utilizando

Counter.

Aclaración: No importa el orden de los verbos cuando tienen igual cantidad de repeticiones. Utilice el módulo **pattern.es**.

Ejemplo:

Este es un párrafo de prueba. El verbo ser, será el mas utilizado. El otro será crear, por eso se creó la oración de esta manera. Por último, se creará esta oración que posee el tercer verbo: poseer. Nada más que decir.

ser 4
crear 3
poseer 2