

## **Trabajo Práctico en Etapas – Parte 2**

### **Juego del Ahorcado**

#### **Objetivo**

Extender la funcionalidad del Juego logrado en la Parte 1, según los requerimientos que se detallan en cada una de las etapas.

#### **Entrega**

El vencimiento de esta segunda parte será el **6 de Julio 2021**.

Deberá ser entregada a través del campus; cumpliendo con las condiciones de entrega establecidas.

#### **Etapas 6 – Refactorización del Código de la Parte 1**

En esta etapa deberán mejorar el código logrado en la Parte 1, introduciendo las mejoras y modificaciones indicadas por el Ayudante supervisor como devolución de la defensa realizada.

Tengan en cuenta en la refactorización, entre otras cosas, mejorar:

- Las reglas utilizadas para depurar las palabras, tales como, recuperar palabras que en los extremos tengan caracteres que no sean letras, como el caso de los signos de puntuación; y cambiar las vocales acentuadas por no acentuadas.
- Mejorar la estructuración del código y la división funcional realizada.
- Optimizar las soluciones en las que se recorran estructuras, por ejemplo, evitar recorrer en el proceso de depuración de las palabras, más de una vez, el texto completo ó la lista de palabras.

Deberán versionar el anterior y el nuevo código refactorizado.

A continuación, podrán avanzar con las siguientes etapas, extendiendo la funcionalidad sobre el código refactorizado.

#### **Etapas 7 - Multijugadores**

En esta etapa daremos la posibilidad que jueguen hasta 5 jugadores, unos contra otros. Para ello sumaremos los siguientes requerimientos:

Al inicio se deberá solicitar el nombre de los jugadores, controlando que no se ingresen más de 5, y que los nombre no se repitan. Podrá considerar que el ingreso de nombres finaliza, cuando se da enter y no se ingresa nada.

Ingresados los nombres de los participantes, se deberán ordenar al azar para asignarle el turno de juego a cada uno de ellos; y se les debe informar antes del comienzo de la partida.

Luego se preguntará cuál será la longitud de la palabra a adivinar, y se aplicará la misma longitud para todos los jugadores, pero se les deberá otorgar distintas palabras. Si no se ingresa nada como valor de longitud, se entenderá que participan todas las palabras, pero en este caso también, al elegir aleatoriamente las palabras a adivinar para cada jugador, se les debe otorgar a todos palabras de igual longitud. Se podría optar por elegir la primer palabra y a partir de esta, el resto de igual longitud, pero siempre la elección de la palabra deberá ser aleatoria entre las no elegidas.

Comienza el primer jugador según su turno. A partir de aquí la dinámica será, si acierta la letra, suma 2 puntos, y elige una nueva letra; así hasta que comete un desacierto, en ese caso se le resta 1 punto, y el turno pasa al siguiente jugador.

Ganará, quien acierte primero la palabra; y suma 10 puntos a su puntaje. Si ninguno acierta, gana el programa, y en este caso se le restarán 5 puntos a cada jugador.

Cuando uno de los jugadores llega al máximo de desaciertos, la partida continúa con el resto de los jugadores.

En cada turno, se debe mostrar el nombre de quien está participando, además de toda la información que se mostraba en la Parte 1 (palabra enmascarada, aciertos, desaciertos, lista de letras utilizadas, puntaje parcial). El programa debe mostrar los mismos mensajes que mostraba en la Parte 1, tanto para los aciertos, desaciertos, letras ya elegidas, e ingresos inválidos.

Al finalizar cada partida, se debe mostrar el resultado de la partida, indicando para cada jugador la palabra que debía adivinar, cantidad de aciertos y desaciertos, el puntaje obtenido en la partida, y si hubo ganador, debe estar indicado.

Si se ha jugado más de una partida, también se debe mostrar los Resultados Generales, indicando la cantidad de partidas jugadas; mostrando por jugador, los datos ordenados por Puntaje Total, junto con la cantidad de aciertos y desaciertos, y la cantidad de palabras adivinadas ó las veces que ha ganado.

A continuación se debe preguntar si se desea jugar una nueva partida.

En cada nueva partida, volver a solicitar la longitud de la palabra a adivinar.

A partir de la segunda partida, el orden de los jugadores, deberá generarse nuevamente, pero el ganador de la última partida, siempre tendrá el primer turno de la siguiente; el resto de los turnos debe otorgarse de forma aleatoria.

## **Etapas 8 - Archivos de Texto**

Introduciremos un cambio a nuestra forma de generar el diccionario de palabras. Ahora que sabemos procesar archivos de texto, en lugar de utilizar la función que nos devolvía un texto, a partir del cual generamos nuestras palabras; debemos reemplazar esa parte del proceso, por otra que tome el texto, desde 3 ó más archivos.

Los archivos serán provistos por la cátedra, los encontrarás en el campus, en la solapa TPs.

Ten en cuenta que si los textos son muy grandes, se hará complicado que leas todo el texto con una única instrucción de lectura; por ello te pediremos que los leas línea por línea y deberás adaptar las funciones que generaban el diccionario de palabras, a esta nueva modalidad.

Evita leer los archivos más de una vez, al igual que debes optimizar el acceso a cualquier estructura auxiliar que utilices.

Como salida de la lectura de los archivos, deberán generar el archivo palabras.csv, que contendrá, todas las palabras obtenidas, ordenadas alfabéticamente, y en cada línea, debe estar la palabra, y la cantidad de veces que aparece en cada archivo. Por ejemplo:

```
nubes,18,4,0
pajaros,3,0,12
puerta,5,8,2
```

También deberán agregar un archivo de configuración, que se llamará configuracion.csv, del cual leerán los valores iniciales que tomarán en la partida, los siguientes elementos:

```
MAX_USUARIOS,10
LONG_PALABRA_MIN,5
MAX_DESACIERTOS,7
PUNTOS_ACIERTOS,10
PUNTOS_DESACIERTOS,5
PUNTOS_ADIVINA_PALABRA,100
PUNTOS_RESTA_GANA_PROGRAMA,20
```

Si por algún motivo, uno o más valores no pueden ser recuperados del archivo de configuración, deben ser establecidos con valores por defecto. Una vez establecidos los valores, al inicio de la aplicación, mostrar por pantalla, el valor efectivamente asignado a cada elemento y si fue dado por omisión ó por configuración.

A partir de ahora, estos serán los valores que rijan nuestra partida, y podrán ser variados antes de iniciar una nueva partida de cero.

**Condiciones de Entrega**

Las siguientes condiciones deben ser respetadas para que la entrega sea considerada válida:

1. Cada función que forma parte del código debe tener debajo de su firma, una descripción corta de cuál es su objetivo y quien es el autor ó responsable de dicha función.
2. El código correspondiente, debe ser subido al campus. El nombre a dar al archivo será `TP2_NombreGrupo.py`. Deberán reemplazar *NombreGrupo*, por el nombre dado a su grupo. Si la entrega está compuesta por más de un archivo .py, generar un .zip con todos los archivos .py, y nombrarlo de igual modo, pero con extensión zip.
3. Deberán grabar 2 videos y subirlos a un canal de Youtube, ó a Google Drive. El primer video, cada integrante del equipo, deberá contar mostrando el código, qué parte estuvo bajo su responsabilidad y los puntos de solución dados, que considere más relevantes. El video total no debe superar los 10 minutos. Comenzar cada uno de los relatos, diciendo el nombre y apellido.
4. Deberán grabar un segundo video, en el que se muestre al menos una jugada completa, y que contemple distintos casos que muestran que la aplicación responde según lo esperado. Deberán ir relatando los eventos de la jugada. En este caso el video puede estar realizado por 1 único integrante.