

Trabajo Práctico en Etapas **Juego FIUBLE – Parte 2**

Objetivo

Adaptar y extender la funcionalidad del Juego logrado en la Parte 1, a los requerimientos que se detallan en cada una de las siguientes etapas.

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 lo realizado en la parte 1, y/o aquellas que ustedes consideren que quedaron pendientes.

Tengan en cuenta en la refactorización, entre otras cosas:

- Mejorar la estructuración del código y la división funcional realizada.
- Optimizar las soluciones en las que se recorran estructuras
- Revisar el PEP de la cátedra y verificar que se respeten las reglas allí indicadas.

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 – Registración de los Jugadores

De ahora en más, para que un jugador pueda participar de la jugada, debe estar registrado. Para ello, la aplicación deberá manejar un archivo usuarios.csv, que contenga dos campos, el nombre de usuario ó jugador, y su clave.

Para lograr esto, se debe desarrollar una interfaz gráfica de ingreso de jugadores, que solicite el ingreso del nombre y la respectiva clave de cada uno de los jugadores, y que además, tenga un botón para registrarse, para cuando el jugador no se encuentre registrado.

Para que el jugador se registre se debe diseñar una nueva ventana gráfica, que permita:

1. El ingreso de un nombre de usuario, el que se debe validar que no exista, y que tenga, como mínimo, un largo de 4 caracteres y un máximo de 15, y estar formado sólo por letras, números y el bajo guion.
2. Una clave, que deben solicitar que se ingrese dos veces, y compararla asegurando que los dos ingresos sean iguales.

La clave debe:

- Tener una longitud entre 8 y 12 caracteres.
- Estar formada sólo por caracteres alfanuméricos, a excepción de letras acentuadas, y por los caracteres "_ " "-"
- Debe contener al menos una letra mayúscula, una letra minúscula, un número, y alguno de los siguientes caracteres: "_ " "-"

Etapas 8 – Obtención de las palabras candidatas

Introduciremos un cambio a nuestra forma de obtener las palabras secretas candidatas a adivinar.

Ahora que sabemos procesar archivos de texto, en lugar de utilizar la función `obtener_palabras_validas()`, que nos devolvía la lista de posibles palabras a adivinar; deberemos escribir las funciones que consideremos necesarias, para procesar 3 archivos de texto, a partir de los cuales generaremos un diccionario de palabras secretas candidatas.

Los archivos serán provistos por la cátedra, los encontrarán en el campus, en la solapa donde se encuentra el enunciado de este TP.

Deberán tener en cuenta que, si los textos son muy grandes, se hará complicado que lean todo el texto con una única instrucción de lectura; por ello la lectura debe ser línea por línea. A medida que vayan leyendo y como parte del procesamiento, deberán extraer de lo leído, las palabras; considerando como palabra sólo aquellas que estén formadas por caracteres alfabéticos. A medida que se lean las líneas y se obtienen las palabras, se debe generar un diccionario con las palabras secretas candidatas, de la longitud que se haya indicado; ya que en esta oportunidad la cantidad de letras de la palabra secreta, dependerá de lo indicado en el archivo de configuración que deberán procesar (ver Etapa 10).

Los archivos sólo pueden ser leídos una vez.

El acceso a cualquier estructura auxiliar que utilices, deberá ser eficiente y óptimo.

Como salida de la lectura de los archivos, deberán generar también el archivo `palabras.csv`, que contendrá, todas las palabras secretas obtenidas, de la longitud indicada, ordenadas alfabéticamente; y en cada línea, debe estar la palabra, y la cantidad de veces que aparece en cada archivo.

Por ejemplo:

ballena,18,4,0

.....

pajaros,3,0,12

puertas,5,8,2

Etapas 9 – Limitación a la cantidad de partidas

Seguiremos dando la posibilidad de jugar múltiples partidas. Después de la finalización de cada partida, y a continuación de la información de los resultados de la partida, preguntaremos si se desea jugar una nueva partida. Pero esta vez, la cantidad de partidas, estará limitada por lo indicado en el archivo de configuración (Ver Etapa 10). Los jugadores podrán dejar de jugar antes de alcanzar el máximo, pero no se podrá continuar con otra partida, si se ha alcanzado el máximo.

Si se finaliza con el juego, ya sea porque se alcanzó el máximo de partidas o se decidió no jugar más, se debe mostrar un resumen total, en base a todas las partidas jugadas.

Se debe generar el archivo `partidas.csv`, en el cual se debe grabar al finalizar cada partida, por cada uno de los jugadores, la siguiente información:

fecha_partida,hora_finalización,nombre_jugador,aciertos,intentos

ordenado en forma descendente por la cantidad de aciertos.

Este archivo es acumulativo, no debe ser reiniciado en cada juego, a menos que esté indicado en el archivo de configuración (ver Etapa 10).

Etapa 10 - Archivo de Configuración

Deberán agregar un archivo de configuración, que se llamará `configuracion.csv`, del cual leerán los valores iniciales, que deben tomar para parametrizar el juego.

Dicho archivo debe contener los siguientes elementos:

```
LONGITUD_PALABRA_SECRETA,7  
MAXIMO_PARTIDAS,5  
REINICIAR_ARCHIVO_PARTIDAS,False
```

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 del juego, 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 todas las partidas de nuestro juego. Podrán ser modificados antes de iniciar un nuevo juego de cero, y se deberá ver reflejado dicho cambio en el juego, por ejemplo, que las palabras secretas, en lugar de ser de longitud 7, sean de longitud 6.

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 pero debe ser distinto, al que lo realizó en la Parte 1.