

Trabajo Práctico en Etapas – Parte 1

Juego Pasapalabra

Objetivo

Construir un programa en etapas, que permita jugar al pasapalabra, teniendo en cuenta las reglas que más adelante se detallan.

Entregas

El Trabajo Práctico está dividido en dos partes. Ambas tendrán una fecha de entrega establecida y es requisito necesario aprobarlas en tiempo y forma; para así poder continuar con la materia y aprobarla.

La primera parte deberá ser entregada a través del campus, y cumpliendo las condiciones establecidas, y su fecha de vencimiento.

La segunda parte agrega nuevas etapas de construcción a la primera entrega, y su fecha de entrega será informada oportunamente, así como la descripción narrativa de las funcionalidades que se sumarán.

Descripción de las Etapas de la Parte 1

La parte 1 está formada por 5 etapas de desarrollo. Esta división en etapas, es para ayudarlos en la organización del diseño y desarrollo total del trabajo. Se les sugiere ir cumpliendo con cada una de las etapas, sin saltar ninguna; ó intentando resolver la totalidad del TP de una. También podrían dividirse las etapas de forma tal de trabajar al menos 2 alumnos en cada una de las etapas, programando colaborativamente y luego en la etapa de integración trabajar todos los miembros del equipo. Es importante destacar, que si siguen este camino, todos deben interiorizarse sobre el código total producido.

Si nunca has jugado al pasapalabra, es recomendable que busques una versión para jugar en línea, y dediques un instante a jugar (por ejemplo: <https://www.cristic.com/pasapalabra/>).

Ten en cuenta que las reglas que estableceremos a continuación, podrían diferir de las cuales vos hayas jugado.

Etapas 1 - Interacción con el Jugador

En esta etapa deberás escribir las funciones que consideres necesarias, y que permitan una interacción con el jugador, y que sigan los lineamientos que se dan a continuación.

Inicialmente, comenzaremos por mostrar el tablero con las letras participantes, debajo de cada letra se mostrará el resultado de haber adivinado la palabra de dicha letra, siendo "a" de acierto, o bien "e" de error.

Al usuario se le debe indicar el turno actual, la cantidad de letras de la palabra a adivinar y la definición de la misma.

Una vez que el usuario ingrese la palabra se le indicará si fue correcta o no, y en el caso de ser incorrecta se le muestra la palabra correcta. y luego se pasa al siguiente turno de letra.

Cuando la palabra es ingresada por el usuario debe validarse que esté compuesta sólo por letras, no están permitidos los números, espacios ni ningún carácter especial, y que sea de la longitud correcta para el turno.

```
[A][C][D][G][I][L][M][P][S][V]
[ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

Aciertos: 0
Errores: 0
Turno letra A - Palabra de 7 letras
Definición: 1. m. Establecimiento militar o particular en que se construyen reparan y conservan las
embarcaciones y se guardan los pertrechos y géneros necesarios para equiparlas
Ingrese palabra:
```

Luego de jugar algunos turnos se vería el tablero de la siguiente forma:

```
[A][C][D][G][I][L][M][P][S][V]
[a][e][ ][ ][ ][ ][ ][ ][ ][ ]

Aciertos: 1
Errores: 1
Turno letra D - Palabra de 5 letras
Definición: 1. m. Utensilio pequeño, ligeramente cónico y hueco, con la superficie llena de hoyuelos
y cerrado a veces por un casquete esférico para proteger el dedo al coser.
Ingrese palabra:
```

Una vez que finalizaron los turnos se debe indicar todo el resumen de la partida con el puntaje final:

```
Turno letra A - Palabra de 7 letras - arsenal - acierto
Turno letra C - Palabra de 6 letras - cabeza - error - Palabra Correcta: cabeza
...
Turno letra Z - Palabra de 5 letras - .....
Puntaje final: 45
```

Etapla 2 - Construcción de un Diccionario de Palabras candidatas

Ahora el objetivo será generar un diccionario de palabras candidatas a adivinar.

Para ello se proveerá una función que retorna un texto en formato:

```
[  
    ["palabra1", "definición1"],  
    ["palabra2", "definición2"],  
    ...  
]
```

Se les proveerá de una función que devolverá un texto del cual extraerán las palabras con sus definiciones para formar el diccionario. Las palabras seleccionadas deberán tener un mínimo de 5 letras.

Una vez generado el diccionario de palabras, se debe mostrar por pantalla el total de palabras que hay por cada letra, y el total que hay en el diccionario.

Etapla 3 - Elección de las palabras candidatas para formar el "rosco"

En esta variante del juego, se seleccionarán al azar 10 letras de todo el conjunto de letras para formar el rosco.

Ahora que tenemos nuestro diccionario, podremos utilizarlo para obtener una palabra candidata de cada letra a adivinar.

Escribí una función, que reciba como primer parámetro el diccionario y como segundo la lista de letras participantes. La función deberá devolver aleatoriamente una palabra que empiece con cada letra participante de entre todas las posibles, esto será retornado como una lista de palabras ordenadas alfabéticamente.

Para probar tu función, utiliza un ciclo que la invoque al menos 100 veces, y analiza lo que obtienes como palabras a adivinar. Repite el proceso varias veces.

Además de la función principal de esta etapa, puedes escribir todas las que consideres necesarias, teniendo en cuenta los conceptos aprendidos en clase sobre programación estructurada y programación modular.

Lista de letras que deben procesar:

```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'ñ', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w',  
'x', 'y', 'z']
```

Etapas 4 - Integración

En esta etapa debemos integrar las funcionalidades resueltas en cada una de las etapas anteriores, haciendo un uso adecuado de las funciones escritas.

La secuencia del juego debe ser la siguiente:

1. Se deberá comenzar con la generación del diccionario de palabras.
2. Luego se deben seleccionar las 10 letras participantes.
3. El programa elegirá al azar la lista de palabras a adivinar por el jugador.
4. Luego se armará el tablero que visualizará el usuario, y dará comienzo la partida, implementando así, lo realizado en la etapa 1.

Etapas 5 - Puntaje

Hasta ahora nuestro jugador, simplemente gana o pierde.

En esta etapa vamos a permitir que obtenga puntaje y que el mismo se acumule de partida tras partida.

Por cada palabra correcta el usuario suma 10 puntos, por cada palabra incorrecta el jugador pierde 3 puntos. El puntaje final obtenido podrá ser negativo.

Al finalizar una partida, se le ofrecerá si desea jugar otra; así hasta que responda que no. El puntaje obtenido en la última partida, se tomará como inicio de la siguiente. Al inicio de la ejecución del 1er. juego, el puntaje se encuentra en cero.

Condiciones de Entrega

Las siguientes condiciones deben ser respetadas en su totalidad 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 o responsable de dicha función.
2. Se deben incluir pruebas unitarias en al menos 5 funciones.
3. El código correspondiente a la Parte 1, debe ser subido al campus. El nombre a dar al archivo será TP1_*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.

4. Deben llevar adelante el desarrollo del código fuente mediante el uso de git mediante la plataforma de github, el mismo será revisado durante la corrección del trabajo práctico.
5. Se deberá incluir un archivo README.md en donde se incluya el nombre del grupo, sus integrantes y un link al repositorio de github (debe ser público).
6. 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.
Las exposiciones se deben entender y ver claramente; y deben intentar que sean homogéneas.
7. 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 y no debe superar los 10 minutos.