

## Clase 23/03/23

En la clase anterior a esta, hemos trabajado para descubrir si la letra que el jugador escribe está en la palabra del juego o no, y en caso de que esta letra se encontrara en la palabra, hemos detectado también en cuál posición estaba.

Dando continuidad a esa parte, el siguiente paso sería mostrar las letras que existan en la palabra en su debida posición, por ejemplo, para la palabra “PASTEL”, debería pasar lo siguiente al hacer el input de la letra “a” o “A”:

```
PASTEL
¿Qué letra quieres intentar? a
_A____
>>> |
```

Lo mismo con “e” o “E”:

```
PASTEL
¿Qué letra quieres intentar? e
____E_
>>>
```

Para hacerlo, lo primero que se debe hacer es crear una variable que tenga tantos guiones bajos (“\_”) como el número de letras de la palabra, esta variable debe de ser de tipo string. Por ejemplo, para la palabra “PASTEL”, la variable tendría 6 guiones bajos, ya si la palabra es “AGUACATE”, tendría 8 guiones bajos.

Teniendo esto en cuenta, concluyese que la variable debe de ser *dinámica*, es decir, su contenido va a depender de la palabra que estemos usando en el juego, esto significa que definir la variable tal que así: **guiones** = “\_\_\_\_\_” sería inválido, ya que de ese modo esta variable solo sería valida para representar palabras de 6 letras.

Una vez creada esta variable, debemos aprovechar el código de la clase anterior que identifica si la letra introducida por el jugador existe en la palabra o no, y en caso de que exista, detecta en qué posición de la palabra está la letra. Usaremos esto para sustituir los guiones correspondientes en la variable de guiones por las letras correspondientes en cada posición.

Para hacer esta parte, necesitamos tener en cuenta que la variable con los guiones es de tipo string, como hemos visto, hacer algo como: **guiones[4] = "E"** sería invalido porque las variables de tipo string no admiten reasignación de caracteres individuales de sus cadenas. Entonces, la manera que tenemos de cambiar un guion por una letra sería siguiendo los pasos abajo:

1º - Verificar si la letra que ingresó el usuario existe en la palabra o no;

2º - En caso de que la letra elegida por el jugador exista en la palabra:

1º - Convertir la cadena de guiones a una lista, esta conversión debe hacerse de tal modo que cada guion de la variable **guiones** ocupe un elemento de la lista, ejemplo:

La variable **guiones** con el valor `"_____"` generaría la variable **lista\_guiones** y su valor sería `["_","_","_","_","_","_","_"]`

**¡OJO!** **guiones** y **lista\_guiones** son variables distintas, en ningún momento hemos cambiado el contenido de la variable **guiones** hasta esta parte, pero la hemos usado como base para crear la variable **lista\_guiones**.

2º - Identificar en qué posición se encuentra la letra ingresada por el usuario en la palabra;

3º - Cambiar la posición correspondiente a esta en la variable **lista\_guiones** por la letra ubicada en esa posición. Por ejemplo, si la palabra es "PASTEL" y el jugador introduce la letra "e", la variable **lista\_guiones** debería de tener el valor en su posición **4** cambiado a una "E", luego, la lista finalizaría con el siguiente valor: `["_","_","_","_","E","_","_"]`

4º - Después de hacer **TODOS** los cambios necesarios en **lista\_guiones**, es necesario deshacer la conversión del primer paso y pasar la variable de tipo lista para tipo string con los valores todos juntos. En esta parte, sustituimos el valor de **guiones** por el valor de **lista\_guiones** después que ésta ya no esté en forma de lista y todos sus valores estén agrupados en una sola string. Ejemplo:

**guiones** vale `"_____"`

**lista\_guiones** vale `["_","_","_","_","E","_","_"]`

convertimos **lista\_guiones** a string y su valor pasa a ser éste: `"____E_"`

Una vez hecha la conversión, guardamos el valor de **lista\_guiones** en **guiones**, con eso, **guiones** pasaría a valer `"____E_"` y su valor estaría actualizado con la letra que el jugador haya introducido.

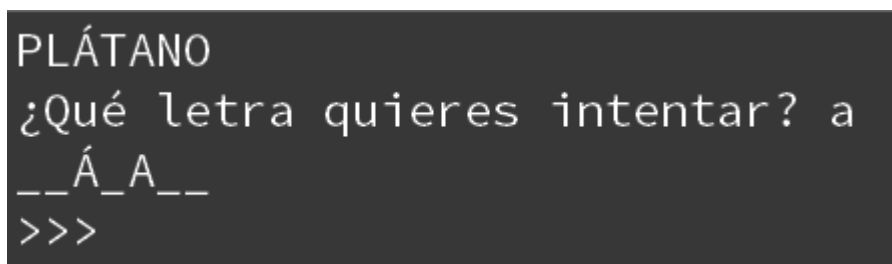
3º- Si la letra no está en la palabra y tampoco es una clave del diccionario de excepciones, hacemos un print con un mensaje diciendo que la letra no está en la palabra y no cambiamos nada en la variable con los guiones.

Tras hacer esto, ya habremos cambiado el contenido de guiones y tenemos un resultado como el enseñado en las capturas al principio.

## Clase 30/03/23

Dando continuidad a lo que fue hecho la clase anterior dónde creábamos la variable con los guiones y cambiábamos el valor de sus posiciones cuando el jugador elegía una letra, ahora tenemos que lidiar con posibles excepciones en el código.

Como Python siempre busca por la correspondencia exacta en los valores de una variable, existe la distinción de, por ejemplo, la letra “A” y letra “Á” en el código y una no es considerada igual a la otra. Para que el juego quede más fácil de jugar y más intuitivo, tenemos que hacer con que el código considere como correcta la letra “Á” si el jugador introduce una “A”, así como considerar válido si en la palabra está la letra “É” y el jugador introduce un “E”, y así en adelante. Podemos ver como debería funcionar esto para la palabra “PLÁTANO”:



```
PLÁTANO
¿Qué letra quieres intentar? a
__Á_A__
>>>
```

Como es posible observar en la captura de pantalla, al introducir la letra “a”, el programa busca en la palabra las “A” y también las “Á”. Esta es la tarea que tenemos que realizar en esta etapa, crear un código capaz de lidiar con estas excepciones.

Para hacerlo, será necesario el uso de una variable de tipo *diccionario*, esta variable asociará las letras con sus posibles excepciones. En la lengua española hay 6 letras que podrían generar estos problemas, que son: A (Á), E (É), I (Í), O (Ó), U (Ú/Ü), N (Ñ), sin embargo, no vamos a considerar la Ü un posible caso, ya que en el conjunto de palabras he eliminado todas las palabras con Ü para que el código pudiera ser más sencillo. Si llevamos en cuenta las dos variaciones de la U, el código sería más complejo y es algo que no nos vale a pena de momento. (Aquí tenéis suerte de no hacer el juego en portugués ya que hay mucho más casos de excepciones, por ejemplo, para la A hay Á, Â y Ã, eso pasa con casi todas las vocales y aumenta considerablemente la complejidad del código)

Lo primero, entonces, es crear un diccionario asociando cada letra con su posible excepción (la A con la Á, la E con la É, etc) luego, implementando el código de la clase anterior, debemos seguir los siguientes pasos para utilizar ese diccionario y tratar las excepciones:

1º - Verificar si la letra que ingresó el usuario existe en la palabra o no;

2º - En caso de que la letra elegida por el jugador exista en la palabra:

➔ Se hace lo de la clase anterior

3º - Caso la letra no esté en la palabra, hay que mirar si la letra tiene un caso especial y genera una excepción, para ello basta mirar si la letra introducida es A, E, I, O, U o N:

Para hacerlo, simplemente tenemos que mirar si la letra indicada por el jugador es una de las claves del diccionario (**¡!!!ESA ES LA CLAVE PARA LOGRAR ESTE PASO!!!**)

1º - Si la letra es una clave del diccionario, miramos si el valor asociado a esta clave (que sería una letra de entre Á, É, Í, Ó, Ú o Ñ) está en la palabra, tal cual lo hemos hecho en la clase anterior. Si la letra asociada a la clave del diccionario está en la palabra, cambiamos el guion en la posición correspondiente de la variable de guiones por la letra que debería estar en esa posición (también, igual lo que fue hecho la clase anterior). Si no está en la palabra, no se hace nada.

4º - Si la letra no está en la palabra y tampoco es una clave del diccionario de excepciones, hacemos un print con un mensaje diciendo que la letra no está en la palabra y no cambiamos nada en la variable con los guiones.

Una vez seguido estos pasos, ya tendríamos el código funcionando como se enseña en la captura al principio.