



## **Anexo B: Guía para el desarrollador**

---

# **ScrabbleAR**

***1.0***

**Julia Saenz**

**04 de octubre de 2020**



# CAPÍTULO 1

---

## Inicio del Programa

---

Trabajo para Seminario de Python 2020 - Alumna Saenz Julia

`Inicio.correr_inicio()`

Abre la ventana de inicio y, de ser necesario, inicializa las variables de la partida

`Inicio.correr_tutorial()`

Abre la ventana de Tutorial



### 2.1 Casillero

Trabajo para Seminario de Python 2020 - Alumna Saenz Julia

```
class Clases.Casillero.Casillero (tipo_='normal')  
    _tipo El tipo de casillero que es  
    _letra La letra que ocupa ese casillero  
    _bloqueado Los casilleros usados al fin del turno se bloquean y no pueden modificarse  
  
    bloquear ()  
        Pone self._bloqueado en True  
  
    devolver_puntos (puntos, palabra_)  
        Depende del tipo de casillero devuelve los puntos correspondientes  
  
    dibujar (clave)  
        Recibe la clave del botón y devuelve el botón según el tipo de casillero  
  
    doble_letra (puntos)  
        Devuelve el doble puntaje de la letra  
  
    doble_palabra (puntos, pal)  
        Devuelve el doble puntaje de la palabra  
  
    esta_bloqueado ()  
        Devuelve si el Casillero está bloqueado  
  
    get_letra ()  
        Devuelve la letra en el Casillero  
  
    get_tipo ()  
        Devuelve el tipo de Casillero
```

**menos\_dos** (*puntos*)  
Resta 2 a la letra

**menos\_tres** (*puntos*)  
Resta 3 a la letra

**menos\_uno** (*puntos*)  
Resta 1 a la letra

**set\_letra** (*letra*)  
Actualiza la letra en el Casillero

**set\_tipo** (*tip*)  
Actualiza el tipo de Casillero

**triple\_letra** (*puntos*)  
Devuelve el triple puntaje de la letra

**triple\_palabra** (*puntos, pal*)  
Devuelve el triple puntaje de la palabra

## 2.2 Tablero

Trabajo para Seminario de Python 2020 - Alumna Saenz Julia

**class** Clases.Tablero.**Tablero** (*tipos*)

Todos los cambios que se quieran a hacer a un Casillero deben hacerse a través de los métodos en Tablero

**\_dimension** Cantidad de casilleros de alto y ancho que tiene el tablero

**\_matriz** Arreglo de Casilleros de tamaño `_dimension` x `_dimension`

**\_posiciones** Arreglo de tuplas de posición de tamaño `_dimension` x `_dimension`

**Tipos** Arreglo de strings que especifican el tipo de casillero de tamaño `_dimension` x `_dimension`

**actualizar\_casillero** (*letra, pos*)  
Recibe una letra y una posición y actualiza ese Casillero

**actualizar\_tipo** (*pos, tip*)  
Recibe una letra y una posición y actualiza el tipo de ese Casillero

**armar\_posiciones** ()  
Crea una matriz de tuplas con las posiciones para los botones

**bloquear\_casilleros** (*casilleros*)  
Recibe una letra y una posición y bloquea ese Casillero

**continuar\_partida** (*arreglo*)  
Recibe una matriz con los casilleros ocupados y actualiza el Tablero

**dibujar** ()  
Devuelve un arreglo de `_dimension` x `_dimension` de botones

**esta\_bloqueado** (*pos*)  
Recibe una letra y una posición y devuelve si ese si ese Casillero está bloauqueado

**get\_casillero** (*pos*)  
 Recibe una letra y una posición y devuelve ese Casillero

**get\_matriz** ()  
 Devuelve el arreglo de la matriz

**get\_posiciones** ()  
 Devuelve la matriz de tuplas de posiciones

**limpiar\_matriz** ()  
 Elimina de la matriz cualquier ficha que no esté bloqueada

**pausar\_partida** ()  
 Guarda los valores de la matriz y los devuelve

## 2.3 Jugador

Trabajo para Seminario de Python 2020 - Alumna Saenz Julia

**class** Clases.Jugador.**Jugador** (*nom*)

**Bolsa** Variable de Clase. Arreglo de letras en el juego

**\_nombre** El nombre del jugador

**\_puntaje** Puntos acumulados en la partida

**\_atril** Arreglo de letras en el atril del jugador

**Cambios** Cantidad de cambios posibles del usuario

**\_casilleros\_usados** Posiciones de los casilleros con letras puestas por el usuario

**actualizar\_puntaje** (*pun*)  
 Suma al puntaje total los puntos nuevos acumulados

**add\_casilleros\_usados** (*cas*)  
 Agrega a un arreglo la posición de un casillero

**armar\_atril** ()  
 Crea el arreglo de 7 fichas aleatorias sacadas de la bolsa

**continuar\_turno** (*datos*)  
 Recibe y actualiza el puntaje, cambios y atril del jugador

**dibujar** ()  
 Devuelve un arreglo de botones con las fichas del atril

**fin\_de\_turno** (*puntos, usadas, cas*)  
 Actualiza el puntaje, agrega los casilleros usados y repone el atril

**get\_atril** ()  
 Devuelve las fichas del atril

**get\_cambios** ()  
 Devuelve la cantidad de cambios hechos en la partida

**get\_cant\_bolsa** ()  
 Devuelve la cantidad de fichas en la bolsa



**get\_ficha** (*pos*)  
Devuelve la ficha correspondiente a una posición del atril

**get\_nombre** ()  
Devuelve el nombre del jugador

**get\_posicion\_letra** (*letra*)  
Devuelve la posición en el atril de una letra

**get\_puntaje** ()  
Devuelve el puntaje del jugador

**guardar\_partida** (*nivel*)  
Guarda y devuelve el nombre y puntaje del jugador, la fecha actual y el nivel al que jugó

**pausar\_turno** ()  
Guarda y devuelve el nombre, puntaje, atril, cambios y casilleros usados del jugador

**reponer\_atril** (*usadas*)  
Repone el atril según la cantidad de fichas que se hayan usado en el turno

**sacar\_fichas** (*fichas*)  
Saca una ficha del atril

**set\_puntaje** (*pun*)  
Setea el puntaje del jugador

**shuffle** (*fichas*)  
Elimina las letras seleccionadas del atril y repone con fichas aleatorias de la bolsa

**terminar\_partida** (*puntos*)  
Resta del puntaje total el valor de las fichas en el atril

## 2.4 Computadora

Trabajo para Seminario de Python 2020 - Alumna Saenz Julia

**class** Clases.Computadora.Computadora

Extiende Jugador, por lo que tiene todos sus variables y métodos

**\_palabra** Guarda la palabra elegida por la computadora cada turno

**\_casilleros** Arreglo de posiciones de los casilleros usados en el turno

**\_atril\_usadas** Arreglo de las letras usadas del atril

**\_puntos** Puntos acumulados en un turno

**\_max** Longitud de la palabra mas larga posible en un turno

**definir\_puntos** (*matriz, puntos, casilleros*)  
Cuenta la cantidad de puntos que suma una palabra

**dibujar** ()  
Dibuja el atril de la compu como botones desactivados

**get\_casilleros** ()  
Devuelve los casilleros en que utiliza la palabra

**get\_palabra()**  
Devuelve la palabra creada por la Computadora

**get\_puntaje\_palabra()**  
Devuelve el puntaje de la palabra

**jugada** (*matriz, diccionario, nivel, primer\_turno*)  
Arma la palabra y la ubica en el Tablero

**sacar\_y\_reponer\_atril()**  
Saca las fichas del atril, repone fichas y borra el valor de la palabra y puntos

## 2.5 Turno

Trabajo para Seminario de Python 2020 - Alumna Saenz Julia

```
class Clases.Turno.Turno
```

**\_letras** Letras usadas en el turno (sin orden)

**\_palabra** La palabra que se forma

**\_casilleros\_usados** Guarda las posiciones de los casilleros usados

**\_letra\_actual** Guarda el valor de la letra actual

**\_pos\_actual** El último valor del atril presionado

**\_orientacion** La orientación de la palabra

**\_turno\_usuario\_** True si es el turno del usuario

**\_lista\_palabras** Lista de palabras jugadas en el la partida

**\_puntaje** Los puntos de la palabra jugada en el punto

**Primer\_turno** True hasta que se juegue el primer turno

**add\_atril\_usada** (*num*)  
Agrega a un arreglo las letras del atril usadas

**add\_lista\_palabras** (*palabra, puntos, nom*)  
Agrega una palabra, su puntaje y quién la jugó a la lista de palabras jugadas

**agregar\_casillero** (*pos*)  
Agrega los casilleros usados en el turno, si hay más de dos guardados, calcula la orientación de la palabra

**definir\_palabra** (*matriz*)  
Según la orientación, ordena los casilleros usados, recupera las letras correspondientes y evalua si la palabra es válida

**definir\_puntos** (*matriz, puntos*)  
Toma los casilleros usados para sumar el puntaje total de la palabra

**es\_turno\_usuario** ()  
Devuelve si es el turno del usuario

**evaluar\_palabra** (*matriz, diccionario, nivel*)  
Se fija si la palabra es válida y en ese caso devuelve sus puntos

**get\_atril\_usadas ()**  
Devuelve las letras del atril usadas

**get\_casilleros\_usados ()**  
Devuelve los casilleros usados

**get\_letra\_actual ()**  
Devuelve la última letra del atril seleccionada

**get\_letras ()**  
Devuelve las letras usadas

**get\_lista\_palabras ()**  
Devuelve la lista de palabras usadas como texto

**get\_orientacion ()**  
Devuelve la orientación de la palabra

**get\_palabra ()**  
Devuelve la palabra

**get\_pos\_actual ()**  
Devuelve el último casillero del atril seleccionado

**get\_primer\_turno ()**  
Devuelve si es el primer turno de la partida

**guardar\_lista\_palabras ()**  
Devuelve la lista de palabras

**juegue\_primer\_turno ()**  
Marca el primer turno como jugado

**limpiar ()**  
Reinicia todas las variables del turno

**reinicio (nom)**  
Agrega la palabra jugada a la lista, reinicia los valores de turno y actualiza de quién es el turno

**sacar\_casillero (pos)**  
Saca el casillero indicado de los casilleros usados

**set\_casilleros\_usados (c)**  
Actualiza los casilleros usados a una lista pasada por parámetro

**set\_letra\_actual (letra)**  
Actualiza la última letra del atril seleccionada

**set\_letras (letra)**  
Actualiza las letras usadas

**set\_lista\_palabras (lista)**  
Actualiza la lista de palabras con una lista pasada por parámetro

**set\_palabra (p)**  
Actualiza la palabra a una pasada por parámetro

**set\_pos\_actual (pos)**  
Actualiza el último casillero del atril seleccionado

**set\_puntaje**(*p*)  
 Actualiza el puntaje con un valor pasado por parámetro

**set\_turno\_usuario**(*va*)  
 Actualiza si es el turno del usuario

**validar\_turno**()  
 Se fija, si es el primer turno, que haya una ficha en el casillero del medio

## 2.6 Tableros Generativos

Trabajo para Seminario de Python 2020 - Alumna Saenz Julia

```
class Funciones.tableros_extra.Celda(estado=1)
```

**Estado** Puede ser 1 o 0

**\_estado\_futuro** Guarda cual será el próximo estado de la Celda

```
actualizar_celda(celdas, i, j)
```

Actualiza el estado\_futuro de la celda según el estado de sus vecinas

```
recargar()
```

Pasa su estado\_futuro a su estado actual

```
class Funciones.tableros_extra.Grilla(cantidad_juegos_: object, cantidad_epocas_: object)
```

**Juegos** Lista de Juegos de La Vida

**Cuantos\_necesito** Cual es el máximo número al que necesito llegar en por lo menos una celda

**Cantidad\_juegos** Cantidad de Juegos de La Vida que quiero

**Cantidad\_epocas** Cantidad de épocas para cada Juego de La Vida

**\_continuar** Si es True, sigue creando Juegos

```
grilla_final()
```

Traduce toda la grilla final de números a strings de tipo de Casillero y devuelve esa matriz

```
hay_suficientes()
```

Se fija si hay por lo menos una celda que, sumando todos los juegos, valga 8

```
imprimir()
```

Imprime la grilla final

```
class Funciones.tableros_extra.JuegoDeLaVida
```

Objeto basado en el Juego de la Vida de John Conway para armar tableros generativos

```
actualizar()
```

Actualiza todas las celdas de la matriz y las recarga

```
armar_celdas()
```

Inicia en cada espacio de la matriz una Celda que vale 0 o 1 aleatoriamente de forma que el patrón sea doblemente simétrico

```
hay_vivas()
```

Devuelve si hay en la matriz más de 20 celdas vivas (que valgan 1)

**imprimir()**

Imprime el estado de las celdas

`Funciones.tableros_extra.tablero_aleatorio()`

Genera un tablero aleatorio

#### 3.1 Validación de Palabras

Trabajo para Seminario de Python 2020 - Alumna Saenz Julia

`Funciones.funciones_palabras.existe_palabra (palabra, diccionario)`

Devuelve True si la palabra sin tilde existe entre las claves del diccionario

`Funciones.funciones_palabras.palabra_es_valida (palabra, diccionario, lista)`

Se ingresa la palabra y la cantidad de categorias para chequear

nivel fácil = ingresar 1, 2

nivel medio = ingresar 2

nivel difícil = ingresar 2

`Funciones.funciones_palabras.palabras_sin_tilde ()`

Crea un diccionario que tiene de clave las palabras sin tilde y de valor la palabra con tilde(en caso de tenerla). Solo agrega palabras entre 2 y 7 letras que no tengan números o signos de puntuación

`Funciones.funciones_palabras.tiene_estas (palabra)`

Devuelve si la palabra pasada por parámetro incluye un número o signo de puntuación

#### 3.2 Funciones de Partida

Trabajo para Seminario de Python 2020 - Alumna Saenz Julia

`Funciones.funciones_partida.letra_actual (event, turno, jugador, window)`

La letra del atril seleccionada por el usuario

`Funciones.funciones_partida.limpiar (turno, tabla, window)`

Saca las fichas no bloqueadas del tablero y las vuelve a activar en el atril

`Funciones.funciones_partida.pausar` (*turno, jugador, compu, tabla, window, config, bolsa, act\_config*)

Guarda los datos de los jugadores, el tablero, la bolsa y el nivel en un archivo

`Funciones.funciones_partida.poner_ficha` (*event, turno, tabla, window*)

Ubica una ficha en el tablero

`Funciones.funciones_partida.que_color` (*tipo*)

Recibe un String con el tipo de casillero y devuelve el color correspondiente

`Funciones.funciones_partida.reinicio_partida` (*window, config, tiempo, Jugador, turno, jugador, compu, diccionario, act\_config, continuar, tabla, niveles*)

Elimina las variables usadas para reiniciar la partida

`Funciones.funciones_partida.shuffle` (*turno, tabla, jugador, window*)

Cambia las fichas del atril seleccionadas por el usuario y saltea el turno, se puede hacer 3 veces por partida

`Funciones.funciones_partida.terminar_partida` (*jugador, compu, window, config, nivel*)

Se cuentan los puntos finales y se muestra el ganador, si el usuario gana, le da la opción de guardar el puntaje

`Funciones.funciones_partida.terminar_turno` (*turno, tabla, jugador, window, diccionario, config*)

Si la palabra es correcta pasa al turno de la compu, sino limpia el tablero

`Funciones.funciones_partida.top_10` ()

Abre un archivo JSON con los 10 mejores puntajes guardados, ordenados de mayor a menor y lo muestra en forma de PopUp

`Funciones.funciones_partida.turno_compu` (*turno, tabla, compu, window, config, diccionario*)

La computadora elige palabra y la posiciona según el nivel

### 3.3 Ventanas Secundarias

Trabajo para Seminario de Python 2020 - Alumna Saenz Julia

`Funciones.Ventanas_secundarias.actualizar_todo_dicc` (*config, niveles, dificultad, tiempo, bolsa, act\_config*)

Actualiza los datos de la configuración actual según la nueva dificultad seleccionada

`Funciones.Ventanas_secundarias.configurar_dificultad` (*config, niveles, bolsa, tiempo, act\_config*)

La configuración de la dificultad: por niveles o personalizada

`Funciones.Ventanas_secundarias.configurar_letras` (*dicc*)

Permite cambiar por letra su cantidad en la bolsa o su puntaje

`Funciones.Ventanas_secundarias.inicio` ()

Ventana de inicio

`Funciones.Ventanas_secundarias.tutorial()`

Ventana de tutorial

`Funciones.Ventanas_secundarias.ventana_shuffle(atril)`

Permite elegir qué fichas del atril intercambiar

## 3.4 Tablero Top10

`Funciones.leaderboard.entra_en_top(puntaje)`

Calcula si el puntaje de la partida es mayor al último de los mejores 10 puntajes

`Funciones.leaderboard.guardar_partida(jugador)`

Guarda el nombre y puntaje del usuario junto con la fecha y el nivel en el Top10





### 4.1 Creación de Niveles

Trabajo para Seminario de Python 2020 - Alumna Saenz Julia

```
Funciones.Niveles.categorias = {'difícil': [2], 'fácil': [1, 2], 'medio': [2]}
```

Diccionario de elementos que configuran un nivel

### 4.2 Estilo de Interfaz

Variables de estilos de los botones, textos y títulos de la interfaz