



Facultad de  
**INFORMÁTICA**



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Informe Trabajo Final

## Python



**Materia:** Seminario de Lenguajes (opción Python)

**Profesoras:** Banchoff, Claudia; Harari, Viviana

**Alumnos:** Garea, Antonella (17614/5); Garofalo, Pedro (17136/5); Giorgetti, Valentín (17133/2)

# ÍNDICE DE CONTENIDOS

<b>Introducción</b>	<b>2</b>
<b>Reglas del juego</b>	<b>3</b>
Niveles del juego	4
Casillas especiales	4
Fin de la partida	5
<b>Temas estudiados (marco teórico)</b>	<b>6</b>
Git y GitHub	6
Entornos virtuales	6
Librerías utilizadas	7
PySimpleGUI	7
Pattern	7
Playsound	8
<b>Problemas y soluciones surgidas durante el desarrollo</b>	<b>9</b>
Diseño de la interfaz	9
Lógica de la computadora	9
Guardado de partida	9
<b>Consideraciones éticas sobre el desarrollo</b>	<b>11</b>
<b>Conclusiones y trabajos futuros</b>	<b>12</b>
<b>Referencias</b>	<b>13</b>
<b>Guía para el usuario</b>	<b>14</b>
Ventana principal	14
Ventana de configuración	15
Ventana de reglas	16
Ventana del top de puntajes	17
Ventana de juego	18
Imágenes de los niveles	21
<b>Guía para el desarrollador</b>	<b>23</b>
Estructura del programa	23
Inteligencia artificial	25

# Introducción

En este informe se detalla el desarrollo del trabajo final de la materia “Seminario de Lenguajes” opción Python. El mismo consiste en implementar el famoso juego “*Scrabble*”<sup>1</sup> usando el lenguaje de programación *Python*<sup>2</sup>, aplicando los conocimientos adquiridos durante la cursada de la materia.

La versión implementada del juego tiene algunas diferencias respecto a las descritas por Jorge Hirthe en el reglamento del juego original (Hirthe J., 2012). El nombre elegido para la versión desarrollada es “*ScrabbleAR*”. Se juega contra la computadora y el jugador puede modificar varios parámetros de la partida.

Las palabras con las que se juega son sustantivos, adjetivos y verbos. Para el análisis sintáctico de las palabras usadas en el juego se utilizó la librería *Pattern*<sup>3</sup>, y para el diseño de la interfaz gráfica se usó *PySimpleGUI*<sup>4</sup>. También se añadieron sonidos haciendo uso de la librería *PlaySound*<sup>5</sup>.

El objetivo del juego es intentar ganar más puntos que la computadora antes de que se acabe el tiempo, formando palabras sobre un tablero, tratando de aprovechar las casillas especiales ubicadas en el mismo.

---

<sup>1</sup> Sitio web de la Asociación Argentina de Scrabble (AAS): [scrabble.org.ar](http://scrabble.org.ar) (último acceso: 06/10/2020).

Sitio web de la Federación Internacional de Scrabble en Español (FISE): [fisescrabble.org](http://fisescrabble.org) (último acceso: 06/10/2020).

<sup>2</sup> Más información sobre el lenguaje de programación utilizado: [docs.python.org/3.6](https://docs.python.org/3.6) (último acceso: 06/10/2020).

<sup>3</sup> Repositorio oficial de Pattern: [github.com/clips/pattern](https://github.com/clips/pattern) (último acceso: 06/10/2020).

<sup>4</sup> Guía oficial de PySimpleGUI: [pysimplegui.readthedocs.io/en/latest](https://pysimplegui.readthedocs.io/en/latest) (último acceso: 06/10/2020).

<sup>5</sup> Repositorio oficial de PlaySound: [github.com/TaylorSMarks/playsound](https://github.com/TaylorSMarks/playsound) (último acceso: 06/10/2020).

## Reglas del juego

Para comenzar la partida se deberá presionar el botón “Iniciar”. Esta acción hará que se elija en forma aleatoria, entre el jugador y la computadora, quien tendrá el primer turno.

Cada partida cuenta con un tiempo límite, y una bolsa de fichas en la cual se almacena una cierta cantidad de fichas de cada letra. Cada ficha tiene su propio puntaje.

Al comenzar la partida se reparten en forma aleatoria 7 (siete) fichas a cada participante. En todo momento, tanto el jugador como la computadora, deben disponer de esta cantidad de fichas en su atril. Las fichas de la computadora se muestran como símbolos de interrogación para no otorgar ventaja al jugador.

Las palabras colocadas en el tablero no deben cruzarse con otras, sólo pueden situarse de izquierda a derecha (orientación horizontal) o de arriba hacia abajo (orientación vertical). No se pueden ingresar palabras colocadas anteriormente. Los tipos de palabras admitidas dependen del nivel seleccionado.

Se debe formar una palabra usando 2 (dos) o más fichas. En la primera jugada, una de ellas deberá estar situada en la casilla rotulada “Inicio”, la cual se encuentra en el centro del tablero.

Una vez ingresada la palabra en el tablero, el jugador debe presionar el botón “Confirmar palabra” para verificar si la palabra es válida. En caso de que la palabra formada no sea válida, se le informará al jugador y el mismo podrá intentarlo nuevamente la cantidad de veces que desee. Si la palabra ingresada es válida, se informará en pantalla la palabra formada, la clasificación y los puntos obtenidos. Luego se repartirán aleatoriamente nuevas fichas al jugador, para reponer las utilizadas en la jugada, y finalmente, se pasará el turno a la computadora.

Si el turno es de la computadora, tratará de formar una palabra válida con las fichas que dispone. En caso de no poder formar una palabra válida, se informará en pantalla e intercambiará todas sus fichas. Luego se pasará el turno al jugador. Si la palabra ingresada es válida, se informará en pantalla la palabra formada, la clasificación y los puntos obtenidos. Luego, se repartirán aleatoriamente nuevas fichas a la computadora, para reponer las utilizadas en la jugada, y finalmente se pasará el turno al jugador.

En cualquier momento de la partida, el jugador puede presionar el botón “Cambiar fichas” para cambiar algunas o todas sus fichas, devolviéndolas a la bolsa y

reemplazándolas por la misma cantidad. Si el jugador realizó algún cambio, se pasará el turno a la computadora.

Tanto el jugador como la computadora pueden realizar un máximo de 3 (tres) cambios durante la partida. En el caso de la computadora, ésta únicamente tendrá la opción de intercambiar todas sus fichas.

El juego cuenta con una configuración predeterminada para cada nivel, de esta forma el jugador puede iniciar una partida rápidamente. Los parámetros que se pueden modificar para una partida son: el nivel, el tiempo límite, la bolsa de fichas (cantidad y puntaje de las fichas) y el nick del jugador. Las configuraciones elegidas por el usuario son guardadas y puede restablecerse a valores por defecto en cualquier momento.

## Niveles del juego

Nivel fácil: el jugador puede colocar cualquier tipo de palabra: sustantivos, adjetivos o verbos. El tamaño del tablero es 19 x 19, tiene muchas casillas con premios (73 en total) y muy pocas con descuentos (16 en total).

Nivel medio: las palabras permitidas en este nivel son adjetivos y verbos. El tamaño del tablero es 17 x 17, tiene menos casillas con premios (65) y más con descuentos (20).

Nivel difícil: en este nivel, el tipo de palabra permitida se selecciona aleatoriamente entre adjetivos y verbos. Es decir, durante la partida sólo se podrán formar adjetivos o verbos, dependiendo del resultado de la elección aleatoria. El tamaño del tablero es 15 x 15, hay algunas casillas con premios (41) y muchas con descuentos (28).

Las figuras 1, 2 y 3 del apartado *imágenes de los niveles* de la *guía para el usuario* muestran el diseño los tableros de los niveles mencionados anteriormente.

## Casillas especiales

- La casilla de inicio suma un punto a la ficha colocada en dicha posición.



*Casilla de inicio*

- Las casillas azules rotuladas "F x2" y "F x3" multiplican por 2 o por 3 respectivamente, el puntaje de la ficha colocada en esa posición.



*Casillas que multiplican el valor de la ficha*

- Las casillas violetas rotuladas "P x2" y "P x3" multiplican por 2 o por 3 respectivamente, el puntaje de la palabra formada. Los multiplicadores pueden acumularse, es decir, si la palabra ocupa una casilla "P x3" y una "P x2", se multiplicará por 5 el puntaje de la palabra formada.



*Casillas que multiplican el valor de la palabra*

- Las casillas negras rotuladas "F -1", "F -2" y "F -3" restan 1, 2 o 3 puntos respectivamente, a la ficha colocada en esa posición.



*Casillas que restan el valor de la ficha*

En todo momento estará visible el puntaje del jugador y de la computadora. Para calcular el puntaje de una jugada se le suma el de cada ficha. Además, si la ficha está ubicada en una casilla especial, se aplicará el modificador correspondiente. El puntaje mínimo de una jugada es 0 (cero).

## Fin de la partida

La partida finaliza cuando ocurre alguna de las siguientes situaciones:

- El jugador presiona el botón "Terminar".
- La computadora no puede formar ninguna palabra y no puede intercambiar sus fichas debido a que alcanzó el máximo de cambios.
- No hay suficientes fichas en la bolsa para repartir a los jugadores.
- Se acaba el tiempo límite de la partida.

Una vez finalizada la partida, se revelan las fichas del atril de la computadora y al puntaje final de cada jugador se le resta el puntaje de las fichas que quedaron en sus atriles. Finalmente, se informa si ocurrió un empate o, en caso contrario, el ganador de la partida con su respectivo puntaje.

## Temas estudiados (marco teórico)

### Git y GitHub

Git es un sistema de control de versiones distribuido comúnmente utilizado para el desarrollo de software comercial y de código abierto. Permite el acceso completo a cada archivo, rama e iteración de un proyecto, y permiten que cada usuario acceda a un historial completo y autónomo de todos los cambios.

“En el 2005, la relación entre la comunidad que desarrollaba el kernel de Linux y la compañía que desarrollaba BitKeeper se vino abajo y la herramienta dejó de ser ofrecida de manera gratuita. Esto impulsó a la comunidad de desarrollo de Linux (y en particular a Linus Torvalds, el creador de Linux) a desarrollar su propia herramienta basada en algunas de las lecciones que aprendieron mientras usaban BitKeeper.” (Chacon & Straub, 2014, p.13).

GitHub es un servicio para alojamiento de repositorios de software que incorpora las características de control de versiones de Git para que puedan ser utilizadas de forma colaborativa. Los desarrolladores pueden trabajar en cualquier lugar y colaborar de forma asincrónica desde cualquier zona horaria.

Estas herramientas fueron utilizadas para el desarrollo del proyecto y fueron fundamentales para la administración del mismo, así como también para el trabajo en equipo. De esta manera fue posible mantener un control sobre el código desarrollado, conservando sus distintas versiones.

Fuente consultada: [guides.github.com/introduction/git-handbook](https://guides.github.com/introduction/git-handbook) (último acceso: 06/10/2020).

### Entornos virtuales

Un entorno virtual es una herramienta que ayuda a mantener separadas las dependencias requeridas por diferentes proyectos mediante la creación de entornos virtuales de Python aislados para cada uno de ellos. "Es útil hacer esta separación porque en el desarrollo de software es muy común que queramos usar en distintos proyectos distintas versiones de la misma librería." (Otarán F., s.f).

Para crear un entorno virtual se utilizó la herramienta "*virtualenv*". Al activar dicho entorno, todas las dependencias del proyecto son instaladas allí.

Durante el desarrollo del proyecto se utilizaron entornos virtuales para actualizar las versiones de las librerías usadas y corroborar que dichas actualizaciones no provocaran errores durante la ejecución del programa. También se usaron entornos virtuales para comprobar que el programa funcionara correctamente en Windows y Linux.

Fuente consultada: [geeksforgeeks.org/python-virtual-environment](https://www.geeksforgeeks.org/python-virtual-environment/) (último acceso: 06/10/2020).

## Librerías utilizadas

Para la instalación de las siguientes librerías se utilizó *PIP* (Python Package Installer).

### PySimpleGUI

PySimpleGUI fue lanzado en 2018 y está basado en Tkinter. Esta librería permitió el diseño y creación de interfaces gráficas del proyecto.

En PySimpleGUI los objetos gráficos (widgets) de las ventanas se colocan en filas y columnas, de acuerdo a la posición y orden en el que se desea que se visualicen. La librería ofrece una amplia variedad de widgets, cada uno con una gran cantidad de parámetros que permiten personalizarlos a gusto del usuario.

Fuente consultada: [pysimplegui.readthedocs.io](https://pysimplegui.readthedocs.io) (último acceso: 06/10/2020).

### Pattern

Pattern es una librería lanzada en 2011 con herramientas para scraping, minería web, procesamiento de lenguaje natural, aprendizaje automático, análisis y visualización de redes.

En el proyecto se utilizó módulo "pattern.es" para el análisis de las palabras ingresadas por los jugadores. Este provee herramientas para la conjugación de verbos, singularización y pluralización de sustantivos, así como también la clasificación de palabras (identifica si una palabra es un sustantivo, adjetivo, verbo, etc.).

Fuente consultada: [github.com/clips/pattern/wiki](https://github.com/clips/pattern/wiki) (último acceso: 06/10/2020).



## Playsound

Playsound fue desarrollado en 2016 y se utiliza para reproducir sonidos desde un archivo local o desde una URL.

Esta librería se usó en el proyecto para emitir un sonido al presionar un botón en las ventanas y al verificar que una palabra ingresada al tablero sea correcta (se reproduce un sonido diferente en caso de que la palabra sea válida o no).

Fuente consultada: [github.com/TaylorSMarks/playsound](https://github.com/TaylorSMarks/playsound) (último acceso: 06/10/2020).

# Problemas y soluciones surgidas durante el desarrollo

## Diseño de la interfaz

La primera dificultad al comenzar el desarrollo del proyecto fue el diseño de la interfaz de usuario. En primer lugar, se realizó un boceto del diseño pensado y luego la idea fue plasmada en código para comprobar el resultado final.

Con la corrección de la primera entrega del proyecto fueron recibidas sugerencias respecto a la navegación en la interfaz. A partir de éstas se realizaron las modificaciones correspondientes: se redujeron el número de ventanas para simplificar la interfaz, logrando que la misma sea más intuitiva y una mejor navegación.

## Lógica de la computadora

El diseño de la lógica para que la computadora pueda jugar fue una de las partes más desafiantes durante el desarrollo del proyecto. Debía lograrse que la computadora forme una palabra y la colocara en el tablero y, de esta manera, sumar puntos y competir contra el jugador.

El algoritmo desarrollado para lograr esto consiste en buscar una ubicación en el tablero donde sea posible colocar una palabra y luego encontrar, a partir de las fichas de la computadora, una palabra válida que quepa en dicha ubicación. Este algoritmo se encuentra detallado en la sección "*Inteligencia artificial*" de la *guía para el desarrollador*.

## Guardado de partida

Otra dificultad encontrada durante el desarrollo fue brindar la posibilidad de que el jugador pueda posponer la partida para continuarla en cualquier otro momento. Para conseguir esto se optó por guardar, en un archivo binario, un diccionario que almacene: las posiciones ocupadas del tablero (un diccionario donde para cada posición almacena la letra ubicada y el color), el turno actual, el tiempo restante, la bolsa de fichas, un booleano que indica si se realizó alguna jugada durante la partida, un string que indica el nivel, tamaño y centro del tablero, un string que indica las palabras válidas, una lista con las palabras ingresadas, la instancia del jugador y de la computadora (las cuales almacenan el puntaje, cambios restantes, etc.).

Al reanudar una partida se utilizará el diccionario mencionado anteriormente para restaurar el juego y el estado del tablero.

Para el manejo de archivos binarios se utilizó el módulo "pickle".

## Consideraciones éticas sobre el desarrollo

La licencia elegida para el software desarrollado es la "Licencia Pública General de GNU" (GNU General Public License), la cual es una licencia de derecho de autor ampliamente usada en el mundo del software libre y código abierto, y garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software.

El propósito de esta licencia es doble: declarar que el software cubierto por la misma es libre, y protegerlo (mediante una práctica conocida como copyleft) de intentos de apropiación que restrinjan esas libertades a nuevos usuarios cada vez que la obra es distribuida, modificada o ampliada.

Esta licencia fue creada originalmente por Richard Stallman, fundador de la Free Software Foundation (FSF) para el proyecto GNU. "Diseñé la GPL de GNU para apoyar y defender las libertades que definen el software libre. [...] Esta licencia garantiza la libertad para estudiar, modificar y redistribuir el programa, y explicita que nadie está autorizado a quitarle estas libertades redistribuyendo el programa bajo una licencia restrictiva." (Stallman R., s.f.)

La imagen utilizada en el menú principal del programa es propia. Los sonidos utilizados fueron buscados en la web "freesound.org". Dicha web es un repositorio colaborativo de muestras de audio con licencia Creative Commons y una organización sin fines de lucro.

Con respecto a las decisiones de diseño, se hizo lo posible para que las ventanas sean lo suficientemente pequeñas para que quepan en la mayoría de las pantallas, requiriendo una dimensión mínima de 1000 x 750 píxeles, para que las mismas se visualicen correctamente.

## Conclusiones y trabajos futuros

Se lograron cumplir todos los objetivos propuestos en la consigna y en las pautas de corrección del trabajo. Esto permitió y requirió asentar los contenidos principales de la materia: el uso del software de control de versiones “git”, la plataforma de desarrollo “github” para trabajar en equipo sobre un proyecto de software, y el lenguaje de programación multiparadigma “Python”.

Se utilizaron varios paradigmas que ofrece el lenguaje: programación imperativa, funcional y orientada a objetos. También se aprendió el uso de las librerías “PySimpleGUI” para la realización de la interfaz de usuario y “Pattern” para verificar las palabras colocadas en el tablero de juego. Además de las dos librerías anteriormente mencionadas, las cuales formaron parte del contenido de la materia, se utilizó “PlaySound” para reproducir varios sonidos durante el juego.

Como trabajo a futuro sería interesante la implementación de un modo de juego multijugador, tanto local como en línea, para poder jugar contra otra persona (amigos, familiares, etc). Otra característica a implementar sería la utilización de ciertas librerías que permitan animar los distintos “widgets” del juego: se podría utilizar, por ejemplo, para arrastrar y soltar las fichas en el tablero. Además sería interesante desarrollar un sistema capaz de traducir el juego a varios idiomas, para de esta forma aumentar el alcance del software desarrollado.

## Referencias

- (Hirthe J., 2012) Reglamento del juego. Disponible en el sitio web de FISE - Federación Internacional de Scrabble en Español: [fisescrabble.org/reglamentos/modalidad-clasica/reglamento-juego-2009](https://fisescrabble.org/reglamentos/modalidad-clasica/reglamento-juego-2009) (último acceso: 06/10/2020).
- (Chacon S. & Straub B., 2014) Pro Git, 2da edición. Disponible en el sitio: [git-scm.com/book/es/v2](https://git-scm.com/book/es/v2) (ultimo acceso: 06/10/2020).
- (Otarán F., sin fecha) Entornos virtuales - Python UNLP Documentación. Disponible en el sitio: [fedeotaran.github.io/py\\_site/entornos\\_virtuales](https://fedeotaran.github.io/py_site/entornos_virtuales) (último acceso: 06/10/2020).
- (Stallman R., sin fecha) La GPL de GNU y el «estilo americano». Disponible en el sitio: [gnu.org/philosophy/gpl-american-way.es.html](https://gnu.org/philosophy/gpl-american-way.es.html) (último acceso: 06/10/2020).

## Guía para el usuario

### Ventana principal

Para iniciar el programa el usuario debe ejecutar el archivo "ScrabbleAR.py". Al hacer esto se mostrará la ventana principal:



*Ventana principal*

A partir de esta ventana, el usuario puede acceder a diferentes apartados del juego: configuración, reglas, top de puntajes, reanudar o iniciar una partida.

La opción "Reanudar partida" sólo estará habilitada en caso de que exista una partida guardada. Si una partida hubiera sido pospuesta y el jugador presiona la opción "Nueva partida" se le pedirá una confirmación, ya que si inicia una nueva se borrará la anterior (sólo puede haber una partida guardada).

## Ventana de configuración

Al seleccionar la opción “Configuración” se mostrará la siguiente ventana:

Letra	Puntaje	Cantidad fichas
A	1	15
B	1	10
C	1	10
D	1	10
E	1	15
F	1	10
G	1	10
H	1	10
I	1	15
J	1	10

*Ventana de configuración*

Desde aquí, el usuario puede modificar diferentes parámetros de la partida. La configuración que se muestra al abrir la ventana es la última seleccionada por el usuario, que en un principio coincide con la configuración por defecto.

La columna de la izquierda es donde el usuario ingresa las configuraciones. Cada nivel tiene su propio perfil de configuración, es decir, la configuración elegida por el usuario afectará únicamente al nivel seleccionado.

La columna de la derecha es donde se muestra la configuración actual. De esta forma, el usuario puede corroborar la configuración seleccionada.

- Para seleccionar el nivel de la partida simplemente se debe presionar el botón correspondiente al nivel.
- Para configurar la cantidad máxima de minutos de la partida se debe ingresar dicha cantidad, la cual debe ser mayor a 0 (cero), y presionar el botón "Confirmar".
- Para modificar la configuración de las fichas se debe ingresar la letra a modificar, el puntaje y/o cantidad de fichas y presionar el botón "Confirmar". Tanto el puntaje como la cantidad de fichas deben ser mayores a 0 (cero).

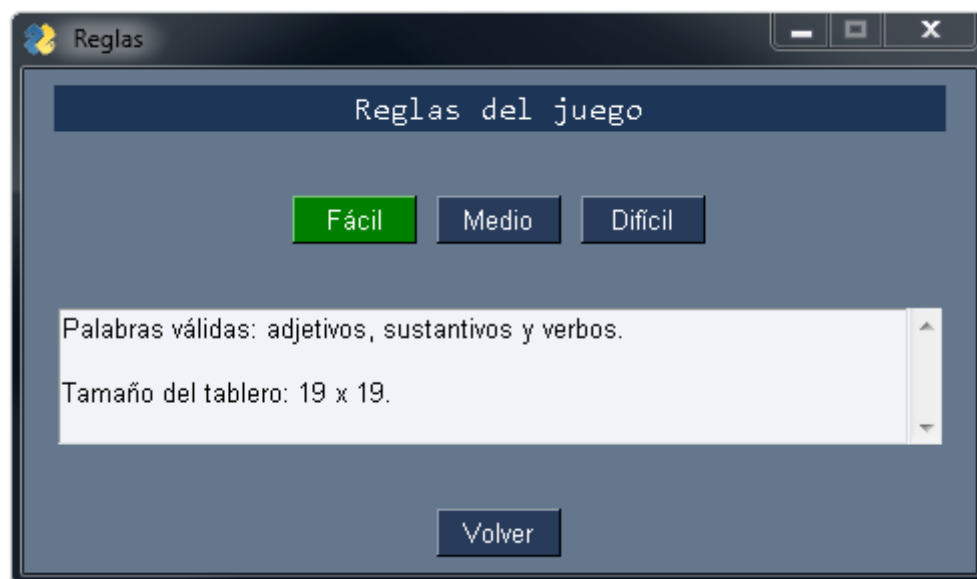


- El usuario puede elegir un nick, que se mostrará en el top de puntajes, para esto debe ingresarlo y presionar el botón "Confirmar".
- En cualquier momento, el jugador puede restablecer la configuración del nivel a valores por defecto, presionando el botón "Restablecer configuración".

Para cerrar la ventana de configuración y volver al menú principal se puede presionar el botón "Aceptar".

## Ventana de reglas

Al seleccionar la opción "Reglas del juego" se mostrará la siguiente ventana:

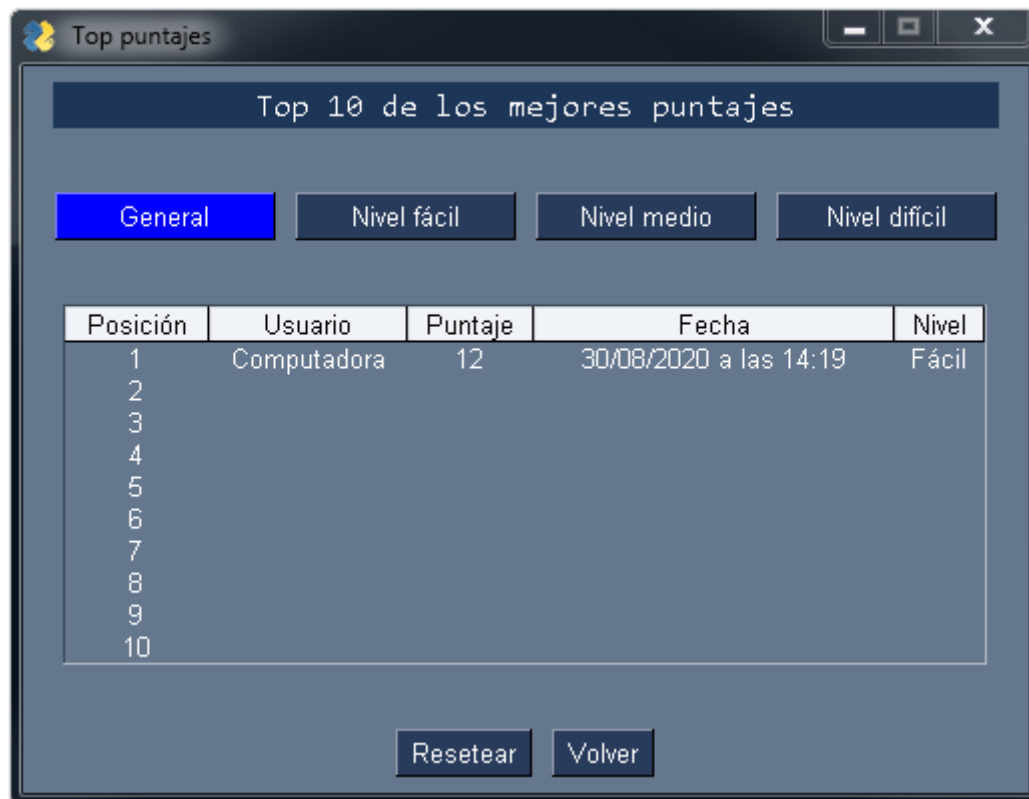


*Ventana de reglas*

Desde aquí, se informa al usuario el tipo de palabras válidas y tamaño del tablero del nivel seleccionado. Para cerrar la ventana de reglas y volver al menú principal se puede presionar el botón "Volver".

## Ventana del top de puntajes

Al seleccionar la opción “Top de puntajes” se mostrará la siguiente ventana:



*Ventana del top de puntajes*

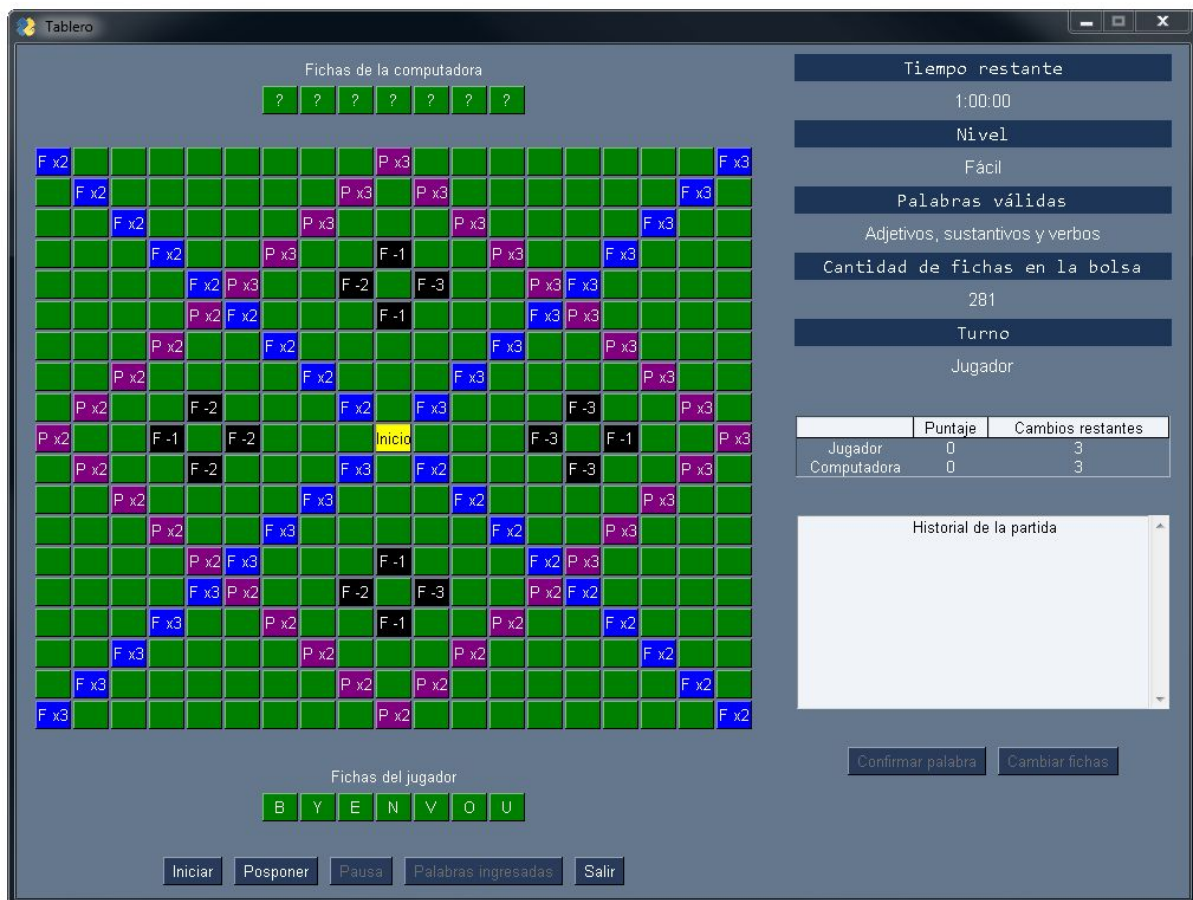
Desde aquí, el usuario puede ver el top diez de los mejores puntajes de cada nivel y, un top general, el cual incluye todos los niveles.

Se muestra una tabla ordenada de mayor a menor puntaje, con información acerca de las partidas: el nick del usuario, el puntaje obtenido, la fecha y el nivel jugado.

El usuario tiene la opción de resetear el top mostrado, presionando el botón “Resetear”.

## Ventana de juego

Al seleccionar la opción “Reanudar partida” o “Nueva partida” se mostrará la ventana de juego:



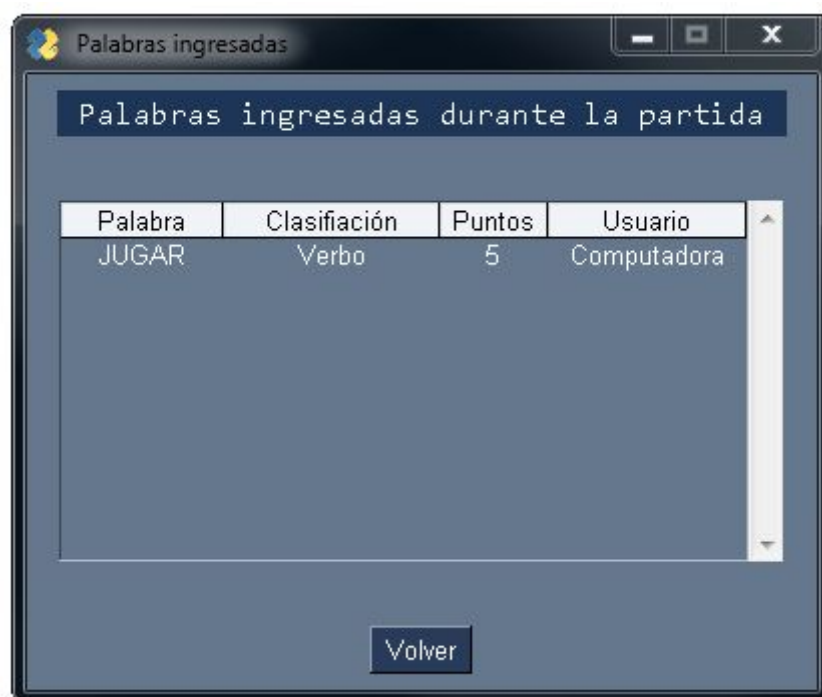
*Ventana de juego*

En la columna de la izquierda se encuentran los atriles de los jugadores (en la parte superior, el de la computadora; en la parte inferior, el del usuario) y el tablero correspondiente al nivel seleccionado. Las fichas usadas por la computadora serán de color naranja y, de color rojo, aquellas usadas por el jugador.

En la columna de la derecha se muestra información sobre la partida: el tiempo restante, el nivel, los tipos de palabras válidas, la cantidad total de fichas en la bolsa, el turno actual, una tabla (ordenada de mayor a menor puntaje) que muestra información sobre el puntaje total y los cambios restantes de cada jugador, y un cuadro de texto donde se irá informando lo ocurrido en la partida (las palabras formadas, los puntos obtenidos, etc.).

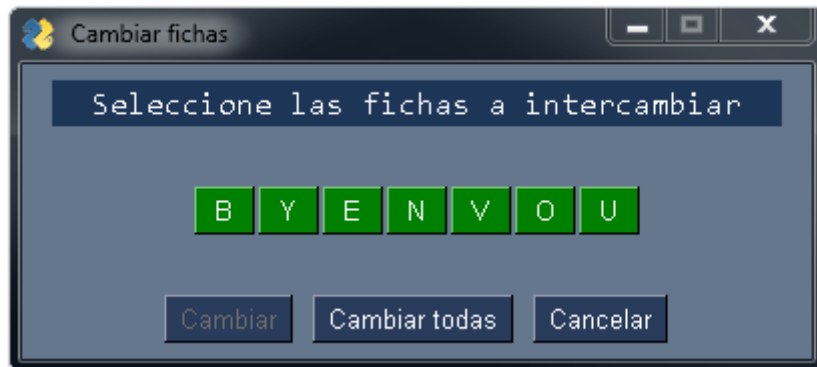
- Si el usuario quiere salir de la partida antes de comenzar a jugar, deberá presionar el botón "Salir".

- Para iniciar la partida, el jugador deberá presionar el botón "Iniciar". Esto también provocará que el mismo se oculte; el botón "Salir" será reemplazado por "Terminar" y se habilitarán los demás botones de la interfaz.
- Para colocar una ficha en el tablero, el jugador debe seleccionar -en su atril- la ficha a colocar y luego, seleccionar una posición del tablero. Para intercambiar una ficha colocada, el usuario debe seleccionar en su atril la nueva ficha y luego, seleccionar la posición del tablero donde se encuentra la ficha a reemplazar. Para quitar una ficha colocada debe estar situada en uno de los extremos de la palabra formada: por ejemplo, si el usuario ingresa la palabra "VER" podrá quitar la ficha "V" o la ficha "R".
- Para guardar la partida, se debe presionar el botón "Posponer". De esta forma, se guardarán los datos de la partida y el usuario podrá reanudarla en otra ocasión.
- Para pausar la partida, el jugador deberá presionar el botón "Pausar".
- Al presionar el botón "Palabras ingresadas" se abrirá una ventana que mostrará una tabla con: las palabras usadas durante la partida, la clasificación, los puntos obtenidos por la palabra y el usuario. Dicha tabla se encuentra ordenada de mayor a menor puntaje.



*Ventana que muestra las palabras ingresadas durante la partida*

- Para que el usuario confirme una palabra ingresada en el tablero deberá presionar el botón "Confirmar palabra".
- Al presionar el botón "Cambiar fichas" se mostrará una pequeña ventana para que el usuario cambie todas o algunas de sus fichas. Si el usuario quiere cambiar algunas fichas, primero debe seleccionarlás y luego presionar el botón "Cambiar". Para cambiar rápidamente todas las fichas se puede presionar el botón "Cambiar todas". Al presionar el botón "Cancelar" se cancelará la operación.



*Ventana de cambio de fichas*

- Para terminar la partida, el jugador debe presionar el botón "Terminar". Esto hará que el mismo se reemplace por el botón "Salir", de esta forma podrá volver al menú principal.

## Imágenes de los niveles

Las figuras 1, 2 y 3 muestran el diseño de los tableros propuestos para los diferentes niveles del juego.

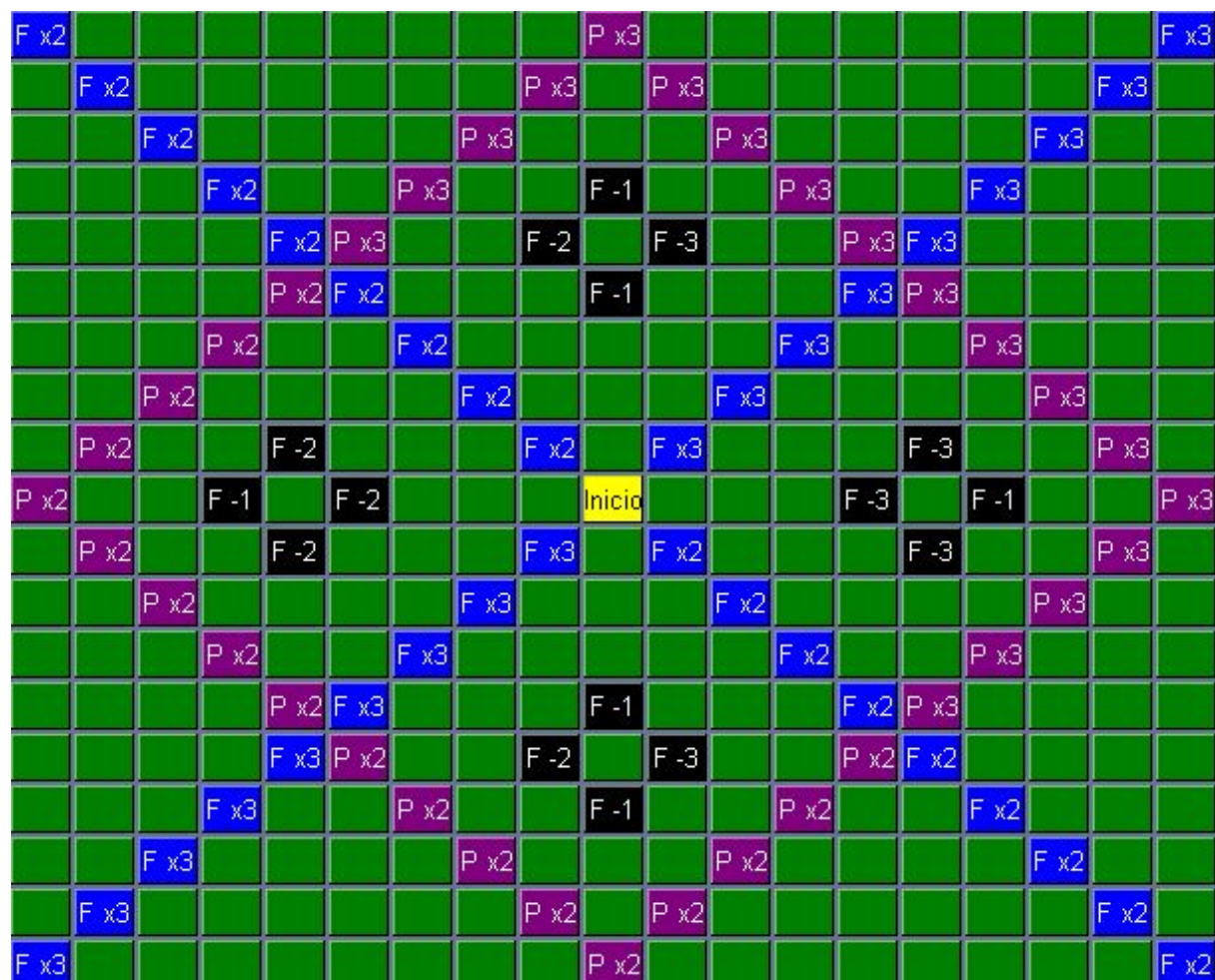


Figura 1: disposición del tablero en el nivel fácil



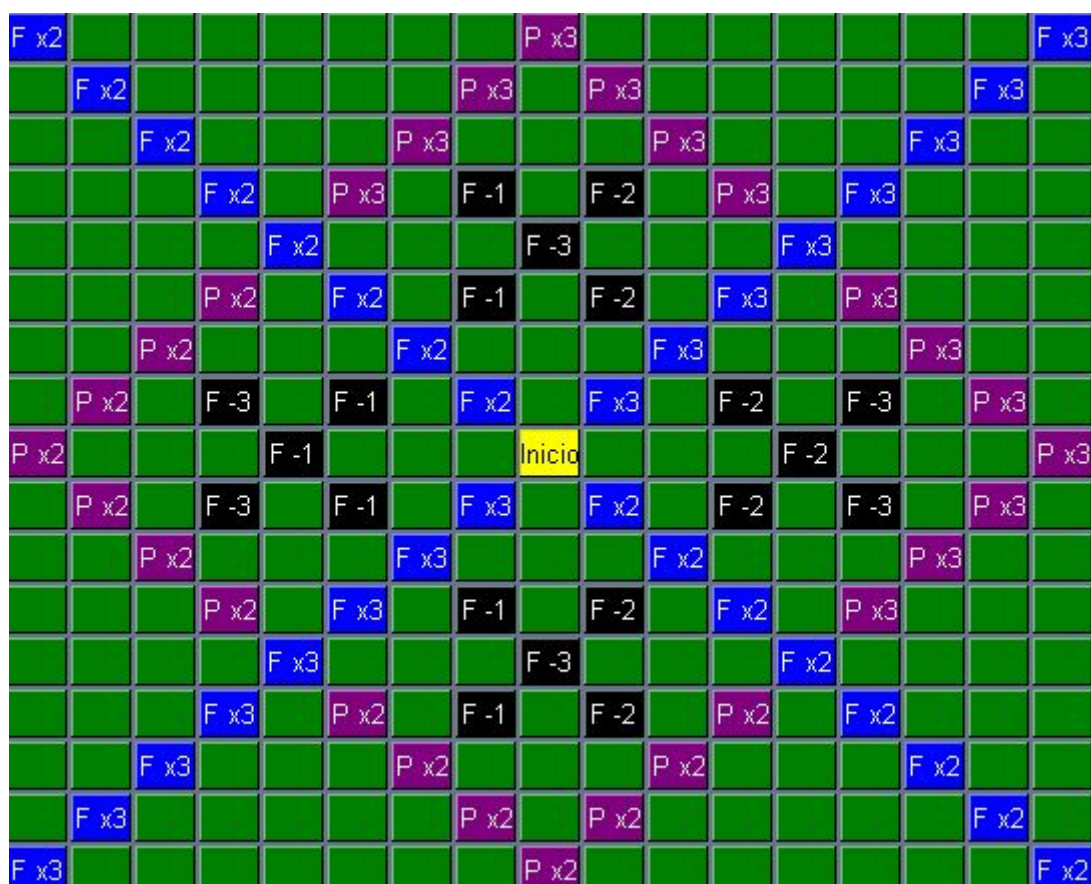


Figura 2: disposición del tablero en el nivel medio

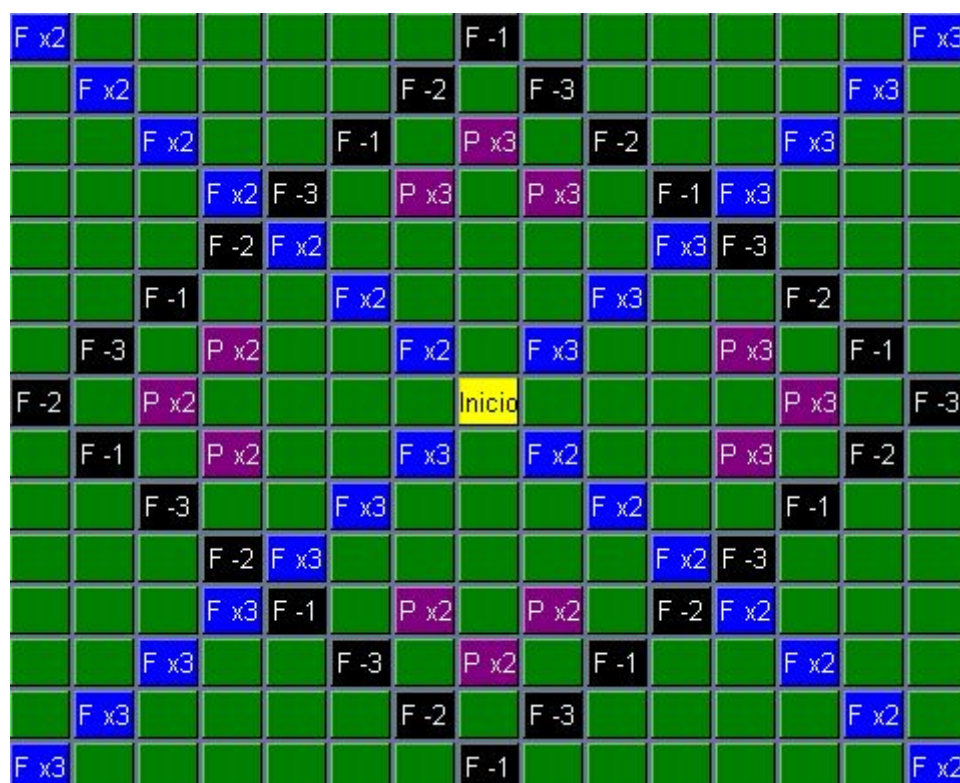


Figura 3: disposición del tablero en el nivel difícil

# Guía para el desarrollador

## Estructura del programa

A continuación, se detalla la estructura del programa, explicando el contenido de los diferentes directorios y módulos del mismo (omitiendo los archivos “\_\_init\_\_.py”).

Lo que se encuentra dentro del repositorio es:

- El archivo ".gitignore", donde se indica que se ignoren los directorios "\_\_pycache\_\_" al subir cambios al repositorio, ya que no son necesarios para ejecutar el programa.
  - El archivo "LICENSE.md", donde se detalla la licencia. La elegida es la Licencia Pública General de GNU (GNU General Public License), una de las más utilizadas en el mundo del software libre y código abierto. La misma garantiza a los usuarios finales la libertad de usar, compartir y modificar el software.
  - El archivo "README.md", el cual tiene información sobre los integrantes del grupo, detalles a tener en cuenta (como por ejemplo las versiones de las librerías usadas), una introducción al proyecto, las reglas y más información relacionada al juego (una pequeña guía para el usuario).
  - El archivo "requirements.txt", que incluye las librerías usadas en el proyecto con sus respectivas versiones.
  - El módulo "ScrabbleAR.py", el cual es principal del programa. Tiene una función "main" que se ejecuta sólo si se abre este mismo archivo en forma directa. Esta función muestra la ventana principal del programa.
- Por último, el directorio "componentes", el cual almacena:
- Una carpeta "imagenes", que contiene una imagen propia (el logo del menú principal).
  - Una carpeta "sonidos", que almacena los tres sonidos usados en el proyecto: "boton.mp3" se reproduce al presionar cualquier botón de las ventanas, "palabra\_correcta.mp3" se reproduce al ingresar una palabra válida, si es inválida se reproduce "palabra\_incorrecta.mp3".
  - Una carpeta "informacion\_guardada", que almacena dos archivos JSON y dos archivos binarios. Los archivos JSON almacenan la configuración predeterminada y



la última configuración seleccionada por el usuario. Los archivos binarios almacenan la partida guardada y el top de puntajes.

Con respecto a los archivos de configuración, ambos coinciden en un primer momento pero, el de última configuración, se actualiza a medida que el usuario modifica parámetros de la partida. Consisten en un diccionario que tiene un campo "nivel\_seleccionado" que almacena un string el cual indica el nivel de la partida, y un diccionario por cada nivel. Estos diccionarios tienen un campo que almacena el nick del jugador, otro para las palabras válidas, el tiempo límite en minutos y un diccionario que almacena información sobre las fichas (un diccionario por cada letra que donde se almacena el puntaje y cantidad de fichas de la misma).

Con respecto a los archivos binarios, el archivo "partida\_guardada" almacena None en caso de que no exista una. En caso contrario, si hay una partida guardada, almacena un diccionario con la información necesaria para restaurar el estado de la anterior (tiempo restante, las palabras ingresadas, etc.). El archivo "top\_puntajes" almacena un diccionario donde las claves son los niveles (strings) y el valor es una lista con el top correspondiente.

- El módulo "jugador.py" donde se define la clase Jugador. Tiene un método "\_\_init\_\_" donde se inicializan las variables de instancia de un Jugador (puntaje, cambios restantes, fichas, nick, color de ficha). Se definen propiedades para dichas variables y un método de instancia "informacion" que devuelve una lista con información del jugador (nombre, puntaje y cantidad de cambios restantes) utilizada para actualizar un widget del tablero de juego.

- Por último, el directorio "ventanas", donde se encuentra la lógica de las ventanas del programa. Dicho directorio almacena:

- El módulo "general", en el cual se definen distintos parámetros usados para la creación de los widgets de las ventanas y una función para leer un evento (retorna el evento, los valores y el tiempo transcurrido hasta que se leyó un evento).

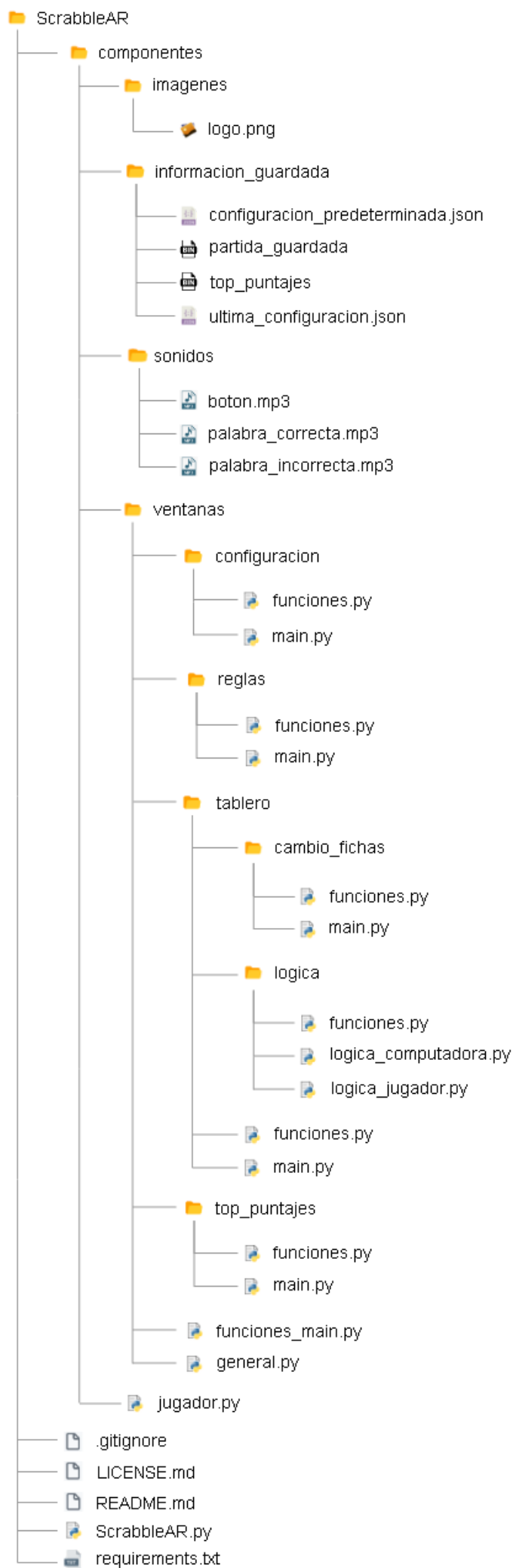
- El módulo "funciones\_main", donde se definen las funciones utilizadas por la ventana principal.

- Una carpeta por cada ventana del programa: almacenan un módulo principal llamado "main", el cual muestra la ventana (lee y responde a los eventos) y un módulo "funciones", donde se definen las funciones usadas para responder a los diferentes eventos de la ventana.

Dentro de la carpeta "tablero" se encuentra otra llamada "lógica". Ésta almacena:

- Un módulo “funciones”, donde se definen diferentes funciones utilizadas tanto por el jugador como por la inteligencia de la computadora (verificar si una palabra es válida, contar los puntos de una jugada, etc.).
- Un módulo “logica\_jugador”, donde se definen las funciones que controlan la lógica del jugador: seleccionar y colocar una ficha en el tablero (verificando que la posición seleccionada para colocarla sea válida), confirmar la palabra ingresada, etc.
- Un módulo “logica\_computadora”, donde se definen las funciones usadas por la inteligencia de la computadora: buscar una ubicación libre, buscar una palabra válida, colocar una palabra en el tablero, etc.

En la siguiente página se muestra la estructura del programa desarrollado, omitiendo los módulos “\_\_init\_\_”.



## Inteligencia artificial

La lógica para que la computadora pueda jugar es la siguiente:

- Primero se busca una ubicación (de un máximo de siete casillas adyacentes libres) donde se pueda colocar una palabra. Los pasos para conseguir esto son:

- Se selecciona una orientación al azar.
- Si es la primera jugada de la partida se retornan siete posiciones adyacentes donde una de ellas corresponde al centro del tablero, ya que si es la primera jugada una de las letras de la palabra debe estar colocada en la casilla inicial.
- Si no es la primera jugada se busca una ubicación de un máximo de siete casillas adyacentes libres.

Para buscar dicha ubicación se selecciona una orientación al azar y una posición aleatoria del tablero. A partir de allí se comprueba si las casillas adyacentes están ocupadas. El proceso se repite hasta encontrar una ubicación de longitud siete o al alcanzar cien iteraciones, retornando la ubicación de mayor longitud encontrada. Si dicha longitud es menor a dos, la computadora intercambia todas sus fichas y se pasa el turno al jugador.

- Luego, si se encontró una ubicación disponible, se busca una palabra válida a partir de las fichas de la computadora que quepa en dicha ubicación. Por ejemplo, si se encontró una ubicación de siete casillas libres:

- Primero se calculan las permutaciones con siete fichas de la computadora. Para esto se utilizó la función “permutations” del módulo “itertools”.
- Luego se verifica si alguna permutación es una palabra válida.
- Si no se encontró ninguna palabra se continúa con las permutaciones, ahora tomando 6 de las 7 fichas. El proceso se repite hasta llegar a permutaciones tomando 2 de las 7 fichas. Si se llegó a este punto y no se encontró una palabra válida, la computadora intercambia todas sus fichas y se pasa el turno al jugador.

- Si se encontró una palabra válida, ésta es ubicada en el tablero, se informa en pantalla la palabra formada, la clasificación y los puntos obtenidos. Luego, se reparten aleatoriamente nuevas fichas a la computadora, para reponer las utilizadas en la jugada y, finalmente, se pasa el turno al jugador.