

# Introducción a la programación

## La palabra escondida

11 de octubre de 2022



## Introducción

El trabajo consiste en hacer un juego cuyo objetivo es adivinar una palabra. No se arriesgan letras sino palabras completas. Se sabe de antemano la longitud de la palabra y el juego debe avisar cuales letras de las palabras son incorrectas (no estan en la palabra), cuales son parcialmente correctas (estan en la palabra pero en otra ubicación) y cuales correctas (estan en la palabra exactamente en esa ubicación). Por ejemplo: La palabra oculta es "TRAPO" y el usuario escribe "CUERO" el juego debe avisar que son incorrectas las letras "C U E", es parcialmente correcta "R" y es correcta la "O". Entonces ahora con esa información el usuario escribe: "TRAZO" y el juego le avisa: incorrectas "Z", parcialmente correctas " " y correctas "T R A O" y así hasta acertar o acabar 5 intentos o haber transcurrido un tiempo predefinido.

Gran parte del juego ya está resuelto, faltan implementar las funcionalidades más importantes.

## 1. El Juego

### Reglas del Juego

Se juega de a un jugador, que cuenta con 60 segundos o 5 intentos para adivinar la palabra.

### Lo que ya esta implementado

El juego actualmente consta de un archivo con el programa principal, otro de configuración y uno de extras. Estos se encargan de capturar la entrada del teclado, llevar la cuenta del tiempo, así como también de dibujar en la pantalla. También cuenta con un archivo, funciones VACIAS que debe completar. Se brinda como ejemplo un archivo de texto con un lecionario, palabras del diccionario sin las definiciones que puede cambiar. Recomendamos utilizar otro archivo con pocas palabras para realizar los test.

Para la mayoría de estas tareas, el programa hace uso de una biblioteca de código llamada *PyGame*. Una biblioteca de código es un conjunto de subprogramas utilizados para desarrollar software. En particular PyGame es una biblioteca especialmente diseñada para el desarrollo de juegos interactivos en Python.

La posición (0,0) de la pantalla es el vértice superior izquierdo, las  $x$  crecen hacia la derecha y las  $y$  crecen hacia abajo.

## Lo que falta implementar

Lo primero que se debe hacer es leer el archivo. Luego debe hacer funcionar el juego, para ello debe completar las funciones vacías obligatorias.

Aun faltan implementar las funciones del archivo `funcionesVACIAS.py`. Estas funciones son utilizadas desde el programa principal.

La función `nuevaPalabra(lista)`: Que recibe una lista de palabras y devuelve una de ellas elegida al azar

La función `lectura(archivo, salida, largo)`: lee el archivo que ya fue abierto en principal y carga en salida solo las palabras cuya longitud es el indicado en el tercer parámetro (`largo`).

La función `revision(palabraCorrecta, palabra, correctas, incorrectas, casi)` chequea la palabra ingresada por el usuario, carga las letras en la lista que corresponda y devuelve `True` en caso de que la palabra sea la correcta y `False` en caso contrario.

No permitir que el usuario ingrese palabras de longitud diferente a la palabra que se desea adivinar.

No permitir que el usuario ingrese palabras que no esten en el lecionario.

Mostrar las letras incorrectas, correctas parcialemnte y correctas de diferentes colores.

## 2. Cómo empiezo

### 2.1. Instalar PyGame

Deben dirigirse a Herramientas, luego Herramientas, Python Packages with pip y ahí escribir `pygame`. Si tienen problemas les dejo links para descargarse las últimas versiones de todo:

<https://sourceforge.net/projects/pyscripter/>

<https://www.python.org/downloads/>

<https://www.pygame.org/download.shtml>

### 2.2. Descargar archivos

Descargar del moodle de la materia el archivo comprimido con todos los archivos necesarios para el TP. Descomprimir todo el contenido del archivo en una carpeta y abrir los archivos `.py` con el PyScripter. El único archivo que se espera que modifiquen al principio es `funcionesVACIAS.py` pero a la hora de ejecutar el proyecto, hay que ejecutar el archivo principal.py.

## 3. Consigna

Implementar las funciones requeridas para el correcto funcionamiento del juego.

Pensar e implementar funciones auxiliares que resuelvan tareas intermedias, de forma tal que el código sea más claro, sencillo, ordenado, legible y fácil de corregir.

Las funciones que reciben listas como parámetros deberán también chequear que dichas listas permanezcan en el estado correcto luego de utilizada la función.

Sugerimos fuertemente probar y corregir las funciones más sencillas antes de encarar las funciones más complicadas.

## 4. Requisitos de aprobación y criterio de corrección

El presente trabajo debe realizarse en grupo, consultar la cantidad máxima a sus docentes. Para aprobar el trabajo, se deberá cumplir los siguientes items:

- El juego debe funcionar correctamente.
- El código debe ser claro. Es decir, las variables y funciones deben tener nombres que hagan fácil de entender el código a quien lo lea, y deben haber comentarios que ayuden al fácil entendimiento de cada porción de código. Además debe hacer funciones siempre que se considere necesario, y se evaluará el buen uso de las mismas.
- El código debe ser coherente. Es decir, no deben haber variables que no se usan, funciones que tomen parámetros que no necesitan, ciclos innecesarios, etc.
- Los casos de prueba deben ser completos y debe ser posible correrlos de nuevo por cualquiera.

Nota: El correcto funcionamiento del juego no es suficiente para la aprobación del trabajo, son necesarios todos los items mencionados arriba.

## 5. Opcionales

Las siguientes funcionalidades del juego no son necesarias para la aprobación (con nota mínima), pero sirven para mejorar la nota del trabajo. De optar por hacerlas, se aplican las mismas reglas y criterios de corrección que para las funcionalidades básicas. Cualquier otra funcionalidad extra que se desee implementar debe ser antes consultada con los docentes.

### Totalmente por tiempo

Si el usuario acierta la palabra aparece una nueva. El reto es acertar muchas palabras en un lapso de tiempo.

### Sin repetidos

Evita que el usuario ingrese mas de una vez una palabra ya utilizada.

### Cierre

Al final el juego hay un cartel de Ganó o perdió indicando cuál era la palabra correcta

### Efectos de sonido

Hacer que el juego reproduzca efectos de sonido cuando sucedan los eventos más importantes: acierta al menos una letra, hay varios aciertos o todos errores, etc.

### Mejores records

Hacer que el usuario luego de jugar tenga la opción de ingresar su nombre y se muestren los 10 mejores puntajes históricos con sus nombres.

### Diferentes niveles

Que el usuario tenga opciones de niveles donde se modifique el juego, por ejemplo:

- el usuario indica la longitud que deben tener las palabras,
- Temático.
- Se juega por tiempo, por cantidad de aciertos, etc

## Juego predefinido

Leer archivos con pocas palabras para poder testear el juego.

## 6. Fecha de entrega

El trabajo debe ser entregado en la fecha estipulada en el cronograma, recordar que es requisito hacer pre-entregas.

## 7. Forma de entrega

El TP se considera correctamente entregado y apto para ser corregido si se cumple con una entrega impresa y una digital.

- La defensa del TP es donde se muestra el correcto funcionamiento del programa. Deben asistir a la misma todos los integrantes del grupo y es necesario que todos los integrantes entiendan todo el trabajo realizado por todo el grupo.
- La entrega digital consiste en enviar un e-mail de la siguiente manera:
  1. A la cuenta de correo electrónico de los docentes de la comisión.
  2. Con asunto: entrega-tp 2022 *nombre – estudiantes*.< *nro – grupo* >.
  3. Adjuntando por separado un archivo comprimido con los archivos que conforman el código fuente de todo el programa y un archivo con el informe (preferentemente en formato pdf). El código debe cargar sin errores. Si el trabajo es muy pesado recomendamos subirlo a google drive, dropbox, etc y pasarnos el link.

Nota: Se acepta una sólo entrega por grupo, de modo que si un grupo entrega su trabajo más de una vez, los docentes elegirán cuál de las versiones entregadas corregir.