

Proyecto Final

Enunciado:








Usando una fila de guiones, se representa la palabra a adivinar, dando el número de letras, números y categoría. Si el jugador adivinador sugiere una letra o número que aparece en la palabra, el otro jugador la escribe en todas sus posiciones correctas. Si la letra o el número sugerido no ocurre en la palabra, el otro jugador saca un elemento de la figura de hombre palo ahorcado como una marca de conteo.

El juego termina cuando:

- El jugador adivinador completa la palabra, o adivina la palabra completa correctamente
- El otro jugador completa el diagrama

Ejemplo de juego

El siguiente ejemplo de juego ilustra a un jugador tratando de adivinar la palabra Ahorcado utilizando una estrategia basada únicamente en la frecuencia de letra.

0		Palabra: ----- Adivinar: E Letras erradas:	6		Palabra: A_OR_A_O Adivinar: H Letras erradas: e,s,n
1		Palabra: ----- Adivinar: A Letras erradas: e	7		Palabra: AHOR_A_O Adivinar: D Letras erradas: e,s,n
2		Palabra: A____A__ Adivinar: O Letras erradas: e	8		Palabra: AHOR_ADO Adivinar: L Letras erradas: e,s,n
3		Palabra: A_O__A_O Adivinar: S Letras erradas: e	9		Palabra: AHOR_ADO Adivinar: C Letras erradas: e,s,n,i
4		Palabra: A_O__A_O Adivinar: R Letras erradas: e,s	10		Palabra: AHORCADO Adivinar: Letras erradas: e,s,n,i
5		Palabra: A_OR_A_O Adivinar: N Letras erradas: e,s			
6		Palabra: A_OR_A_O Adivinar: H Letras erradas: e,s,n			

El adivinador gana, la respuesta, como descubrió el adivinador, era ahorcado.

Actividad

Diseña un programa que nos permita realizar la simulación del juego del ahorcado en el lenguaje de programación Python.

Puntos para evaluar

- Uso de listas.
- Programación orientada a objetos.
- Declaración de variables y funciones.
- Uso de estructuras de control y ciclos de repetición.
- Comentarios en el código.
- Usos de archivos externos.

En esta práctica no se hace uso de ninguna librería para dibujar de manera gráfica el muñeco.

Propuesta de solución:

Para la resolución del problema se propone un programa principal en el que se reparten dentro de funciones para la reutilización de código los cuales constan de:

- Función para hacer recursivo el juego, de manera que depende de un valor booleano se puedan reiniciar variables o terminar el juego.
- Función para retornar una palabra aleatoria desde un archivo de texto que contendrá una lista de palabras definidas. La lista será una lista dentro de una cadena, de manera que con un método de listas se pueda dividir dentro de una lista.
- Función para actualizar de manera grafica 'el ahorcado', así mismo hará la separación por letras de la palabra a adivinar y guardará las letras correctas, como las incorrectas.

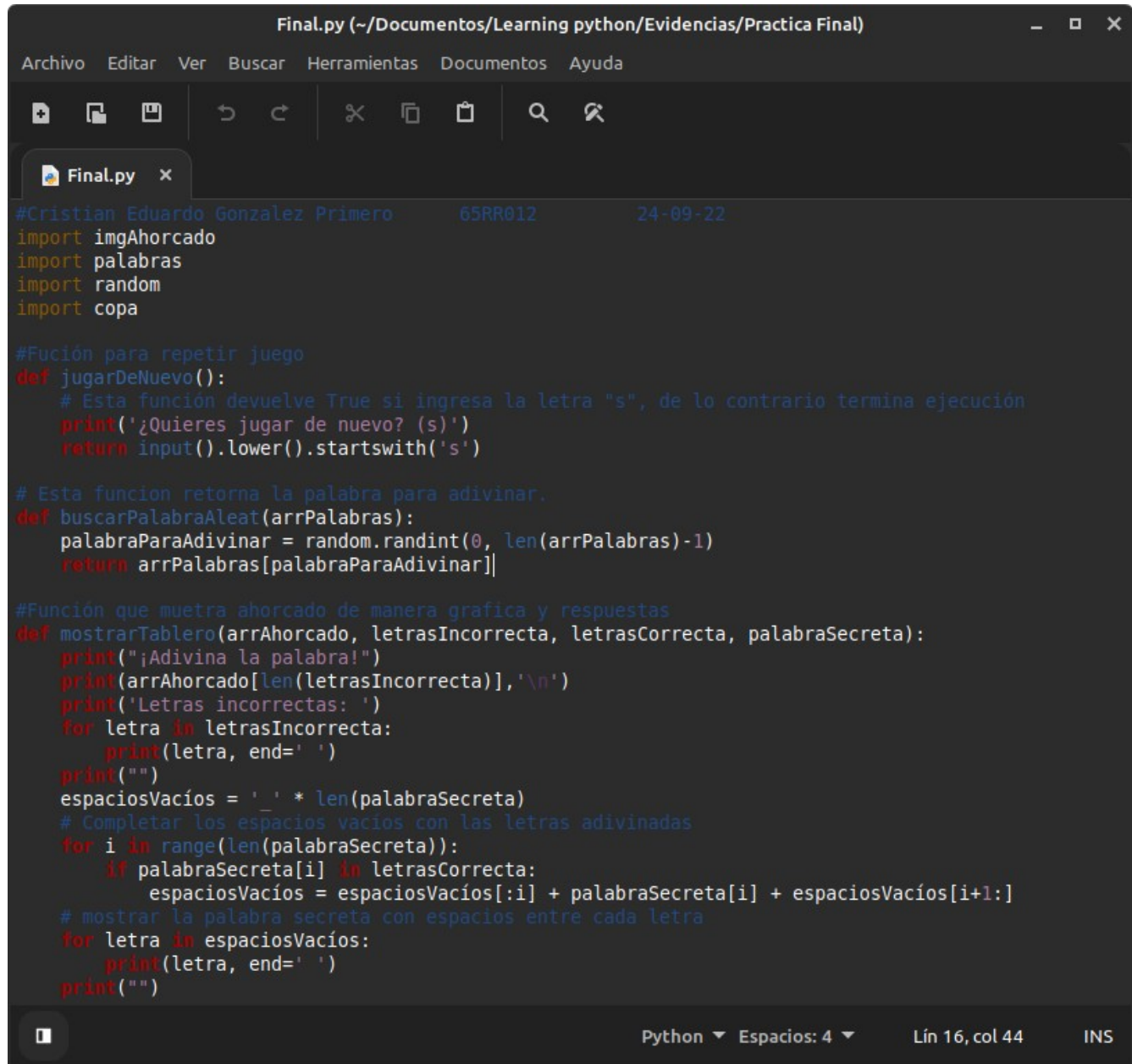
Para graficar 'el ahorcado' se tomará un archivo de texto con imágenes en código ascii dentro de una lista. Dependiendo de los errores se sumará 1 para cambiar de index.

- Función para verificar entradas del usuario, evitando errores por entrada de caracteres especiales, o letras repetidas.

Dentro del programa principal se hace la verificación y en caso de fallar mandará un valor False y break para terminar las comparaciones.

Toda entrada de cadenas será pasada a minúsculas para evitar errores.

Programa principal



```
Final.py (~/Documentos/Learning python/Evidencias/Practica Final)
Archivo  Editar  Ver  Buscar  Herramientas  Documentos  Ayuda

Final.py x

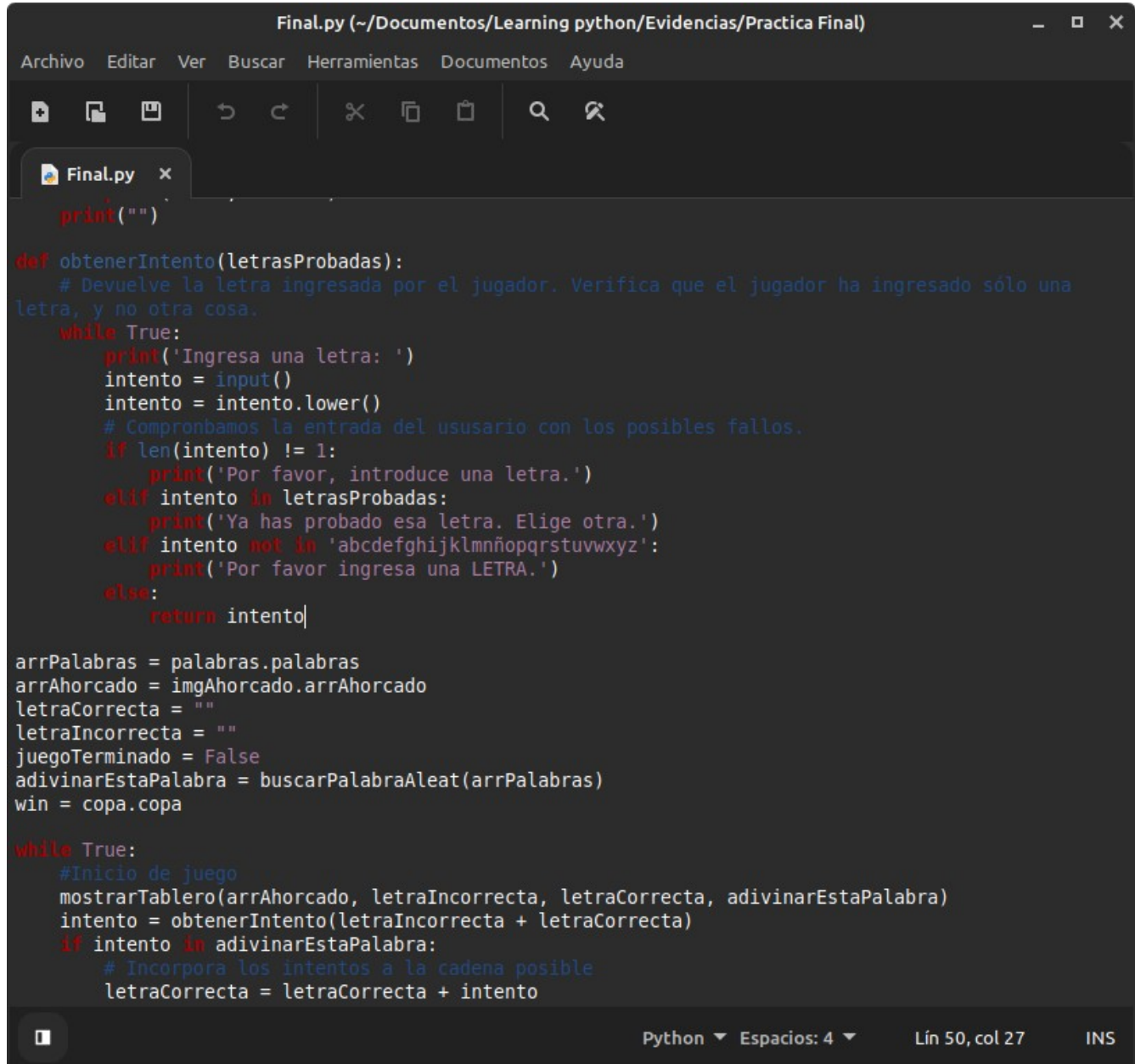
#Cristian Eduardo Gonzalez Primero      65RR012      24-09-22
import imgAhorcado
import palabras
import random
import copa

#Función para repetir juego
def jugarDeNuevo():
    # Esta función devuelve True si ingresa la letra "s", de lo contrario termina ejecución
    print('¿Quieres jugar de nuevo? (s)')
    return input().lower().startswith('s')

# Esta función retorna la palabra para adivinar.
def buscarPalabraAleat(arrPalabras):
    palabraParaAdivinar = random.randint(0, len(arrPalabras)-1)
    return arrPalabras[palabraParaAdivinar]

#Función que muestra ahorcado de manera grafica y respuestas
def mostrarTablero(arrAhorcado, letrasIncorrecta, letrasCorrecta, palabraSecreta):
    print("¡Adivina la palabra!")
    print(arrAhorcado[len(letrasIncorrecta)], '\n')
    print('Letras incorrectas: ')
    for letra in letrasIncorrecta:
        print(letra, end=' ')
    print("")
    espaciosVacíos = '_' * len(palabraSecreta)
    # Completar los espacios vacíos con las letras adivinadas
    for i in range(len(palabraSecreta)):
        if palabraSecreta[i] in letrasCorrecta:
            espaciosVacíos = espaciosVacíos[:i] + palabraSecreta[i] + espaciosVacíos[i+1:]
    # mostrar la palabra secreta con espacios entre cada letra
    for letra in espaciosVacíos:
        print(letra, end=' ')
    print("")
```

Python ▾ Espacios: 4 ▾ Lín 16, col 44 INS



```
Final.py (~./Documentos/Learning python/Evidencias/Practica Final)
Archivo  Editar  Ver  Buscar  Herramientas  Documentos  Ayuda

+  [icon]  [icon]  [icon]  [icon]  [icon]  [icon]  [icon]  [icon]
Final.py x

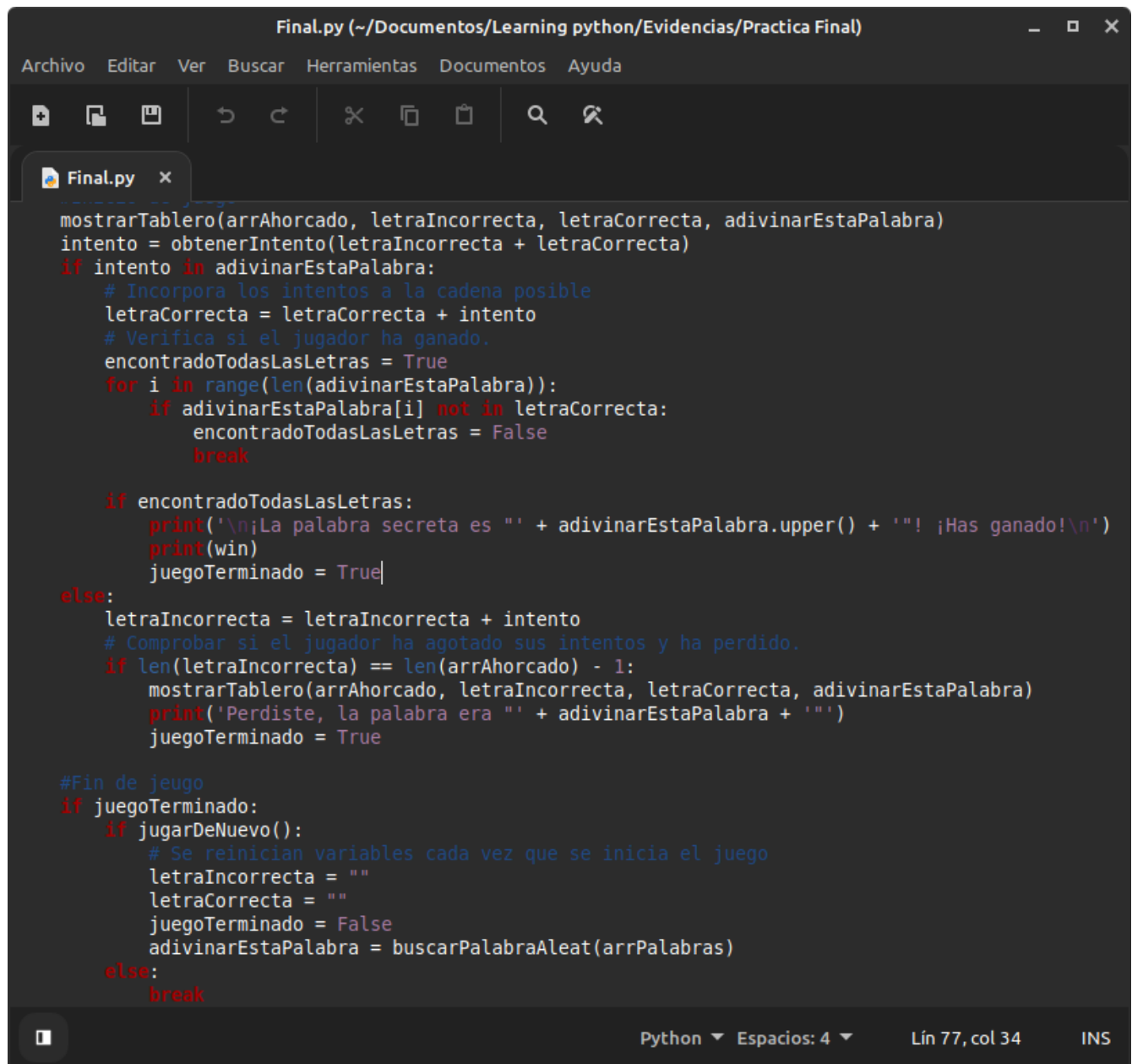
print("")

def obtenerIntento(letrasProbadas):
    # Devuelve la letra ingresada por el jugador. Verifica que el jugador ha ingresado sólo una
    # letra, y no otra cosa.
    while True:
        print('Ingresa una letra: ')
        intento = input()
        intento = intento.lower()
        # Compramos la entrada del usuario con los posibles fallos.
        if len(intento) != 1:
            print('Por favor, introduce una letra.')
        elif intento in letrasProbadas:
            print('Ya has probado esa letra. Elige otra.')
        elif intento not in 'abcdefghijklmnopqrstuvwxyz':
            print('Por favor ingresa una LETRA.')
        else:
            return intento

arrPalabras = palabras.palabras
arrAhorcado = imgAhorcado.arrAhorcado
letraCorrecta = ""
letraIncorrecta = ""
juegoTerminado = False
adivinarEstaPalabra = buscarPalabraAleat(arrPalabras)
win = copa.copa

while True:
    #Inicio de juego
    mostrarTablero(arrAhorcado, letraIncorrecta, letraCorrecta, adivinarEstaPalabra)
    intento = obtenerIntento(letraIncorrecta + letraCorrecta)
    if intento in adivinarEstaPalabra:
        # Incorpora los intentos a la cadena posible
        letraCorrecta = letraCorrecta + intento
```

Python ▾ Espacios: 4 ▾ Lín 50, col 27 INS



The image shows a screenshot of a Python IDE window titled "Final.py (~/Documentos/Learning python/Evidencias/Practica Final)". The window has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Herramientas", "Documentos", and "Ayuda". Below the menu bar is a toolbar with icons for file operations and editing. The main editor area displays a Python script for a Hangman game. The script includes functions for displaying the game board, getting a guess, and checking if the guess is correct. It also includes a main loop that continues to play the game until the user chooses to stop.

```
mostrarTablero(arrAhorcado, letraIncorrecta, letraCorrecta, adivinarEstaPalabra)
intento = obtenerIntento(letraIncorrecta + letraCorrecta)
if intento in adivinarEstaPalabra:
    # Incorpora los intentos a la cadena posible
    letraCorrecta = letraCorrecta + intento
    # Verifica si el jugador ha ganado.
    encontradoTodasLasLetras = True
    for i in range(len(adivinarEstaPalabra)):
        if adivinarEstaPalabra[i] not in letraCorrecta:
            encontradoTodasLasLetras = False
            break

    if encontradoTodasLasLetras:
        print('\n¡La palabra secreta es "' + adivinarEstaPalabra.upper() + '"! ¡Has ganado!\n')
        print(win)
        juegoTerminado = True
else:
    letraIncorrecta = letraIncorrecta + intento
    # Comprobar si el jugador ha agotado sus intentos y ha perdido.
    if len(letraIncorrecta) == len(arrAhorcado) - 1:
        mostrarTablero(arrAhorcado, letraIncorrecta, letraCorrecta, adivinarEstaPalabra)
        print('Perdiste, la palabra era "' + adivinarEstaPalabra + '"')
        juegoTerminado = True

#Fin de juego
if juegoTerminado:
    if jugarDeNuevo():
        # Se reinician variables cada vez que se inicia el juego
        letraIncorrecta = ""
        letraCorrecta = ""
        juegoTerminado = False
        adivinarEstaPalabra = buscarPalabraAleat(arrPalabras)
    else:
        break
```

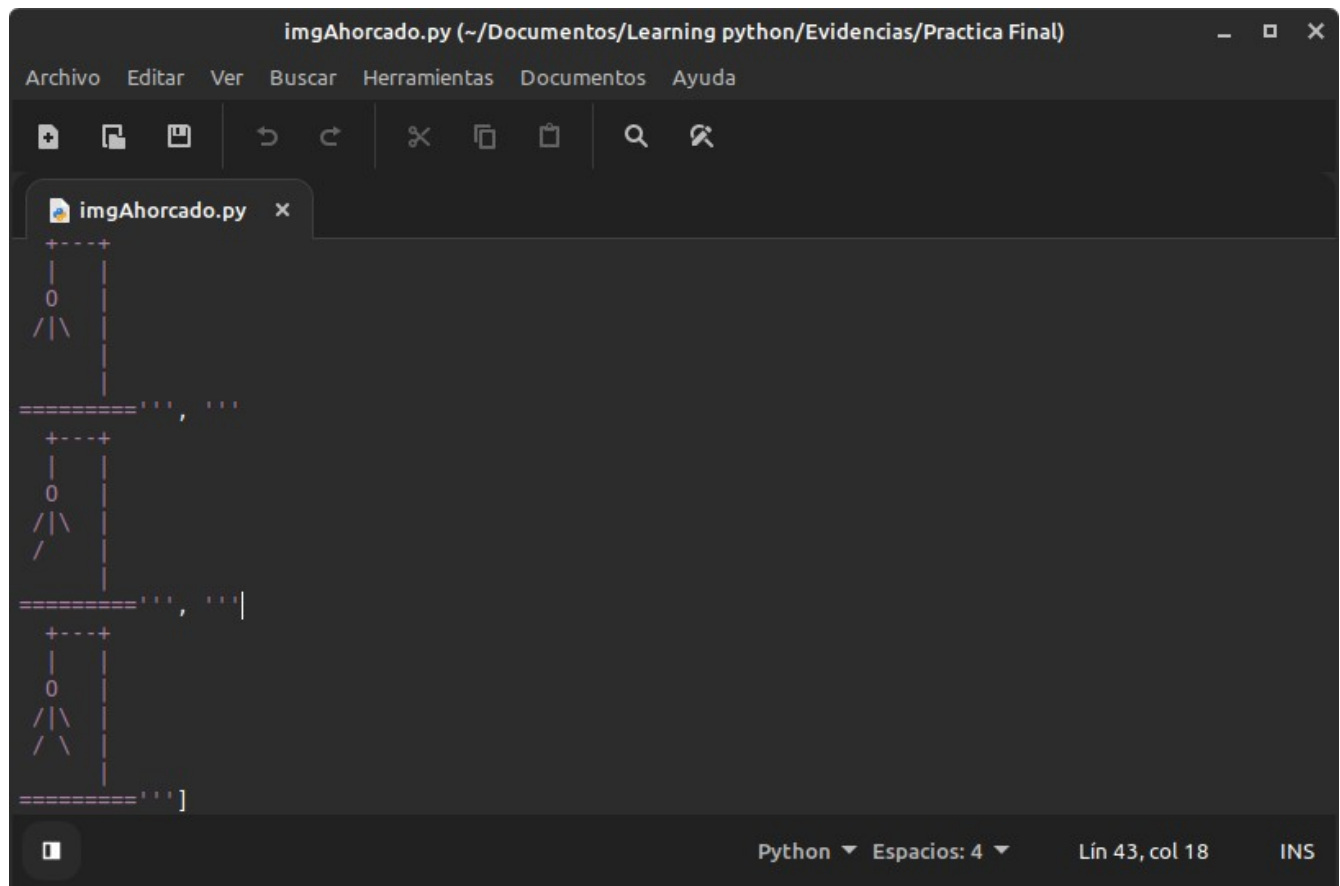
Python ▾ Espacios: 4 ▾ Lín 77, col 34 INS

Ahorcado en ascii

```

imgAhorcado.py (~/Documentos/Learning python/Evidencias/Practica Final)
Archivo Editar Ver Buscar Herramientas Documentos Ayuda
+ - + 
imgAhorcado.py x
arrAhorcado = ['''
+---+
|
===== ''', '''
+---+
|
===== ''', '''
+---+
|
===== ''', '''
+---+
|
===== ''']

```



```
imgAhorcado.py (~/Documentos/Learning python/Evidencias/Practica Final)
Archivo  Editar  Ver  Buscar  Herramientas  Documentos  Ayuda

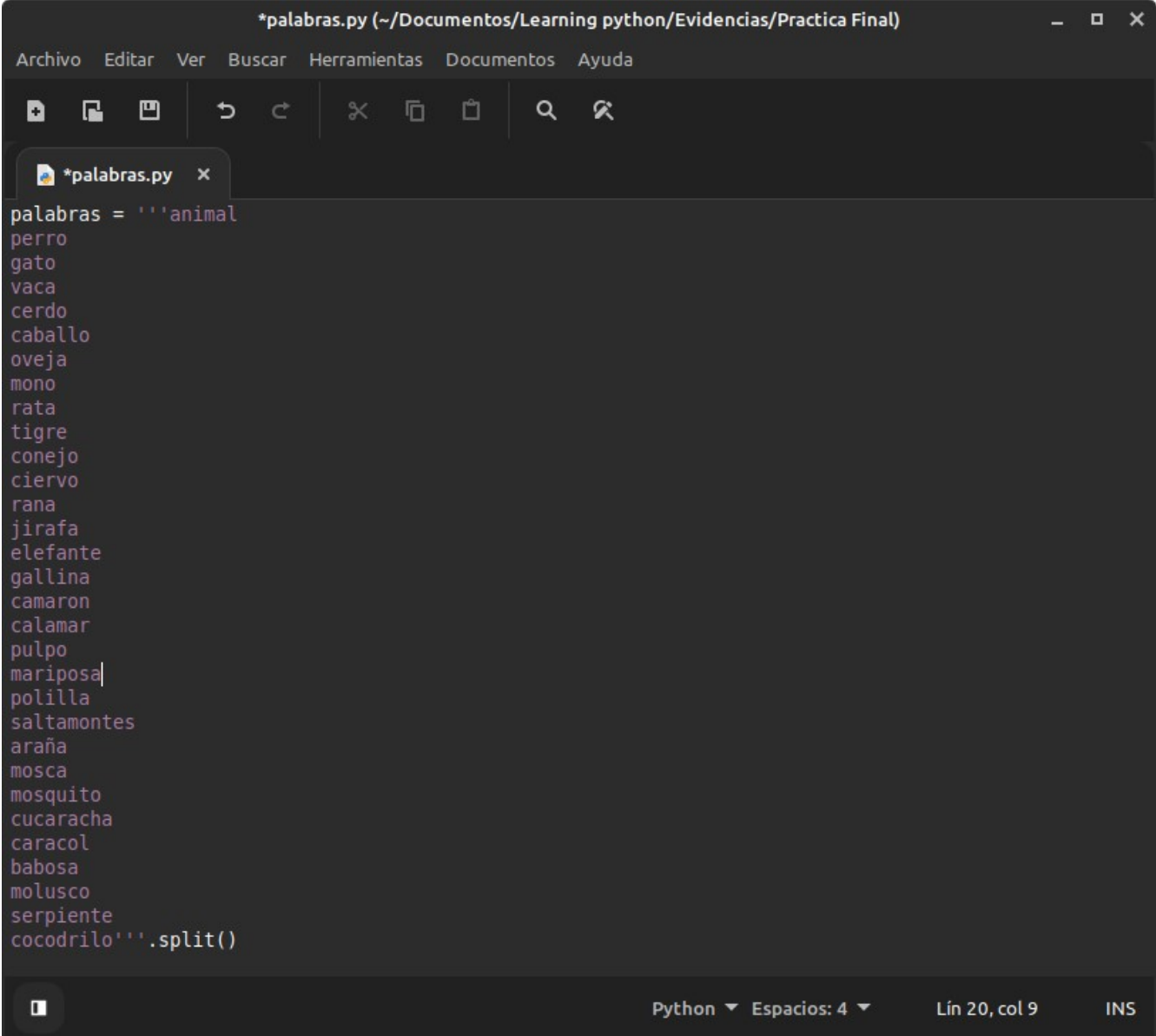
+---+
|   |
| 0  |
|/|\ |
|   |
+---+
===== ' ', ' '

+---+
|   |
| 0  |
|/|\ |
|/  |
+---+
===== ' ', ' '

+---+
|   |
| 0  |
|/|\ |
|/ \ |
+---+
===== ' ' ]

Python  Espacios: 4  Lín 43, col 18  INS
```

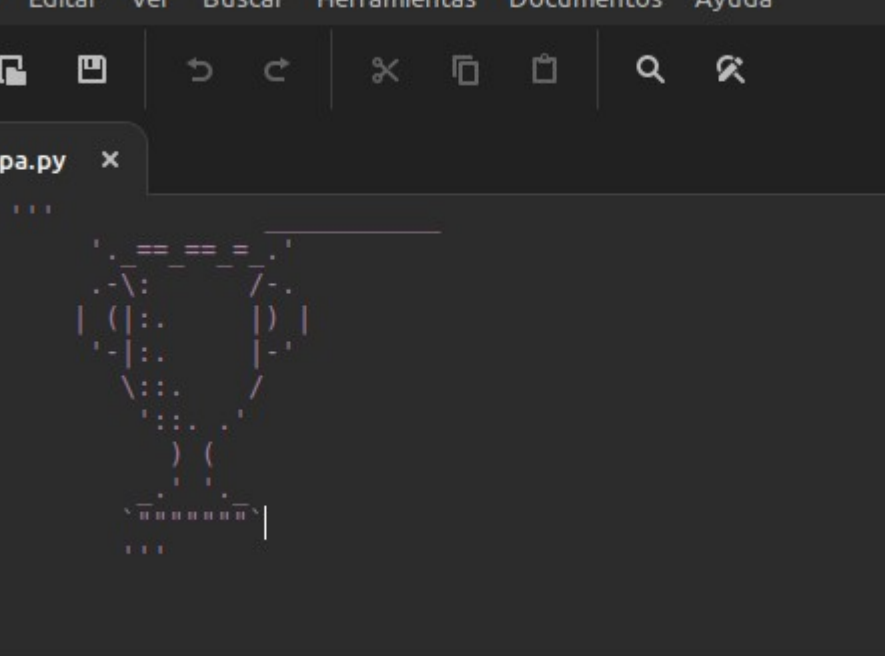
Listado de palabras



The image shows a screenshot of a code editor window titled `*palabras.py (~/Documentos/Learning python/Evidencias/Practica Final)`. The editor has a menu bar with `Archivo`, `Editar`, `Ver`, `Buscar`, `Herramientas`, `Documentos`, and `Ayuda`. Below the menu is a toolbar with icons for file operations and editing. A tab labeled `*palabras.py` is active. The code in the editor is a Python script that defines a list of animal names. The list starts with `palabras = '''animal` and ends with `cocodrilo'''.split()`. The animal names listed are: `perro`, `gato`, `vaca`, `cerdo`, `caballo`, `oveja`, `mono`, `rata`, `tigre`, `conejo`, `ciervo`, `rana`, `jirafa`, `elefante`, `gallina`, `camaron`, `calamar`, `pulpo`, `mariposa`, `polilla`, `saltamontes`, `araña`, `mosca`, `mosquito`, `cucaracha`, `caracol`, `babosa`, `molusco`, `serpiente`, and `cocodrilo`. The status bar at the bottom shows `Python`, `Espacios: 4`, `Lín 20, col 9`, and `INS`.

```
palabras = '''animal
perro
gato
vaca
cerdo
caballo
oveja
mono
rata
tigre
conejo
ciervo
rana
jirafa
elefante
gallina
camaron
calamar
pulpo
mariposa
polilla
saltamontes
araña
mosca
mosquito
cucaracha
caracol
babosa
molusco
serpiente
cocodrilo'''.split()
```


Copa de ganador



```
copa = '''
    _ _ _ _ _
    _\_: /-
    | (|: |) |
    _|: | -
    \:: /
      . .
      ) (
      _ _
    _ _ _ _ _
    '''
```

Capturas de juego

```
soyskipper@cristianhpmint: ~/Documentos/Learning python/Evidencias/Practica Final
Archivo Editar Ver Buscar Terminal Ayuda
soyskipper@cristianhpmint:~/Documentos/Learning python/Evidencias/Practica Final$ python3 Final.py
¡Adivina la palabra!

+---+
|
|
|
|
|
|
=====

Letras incorrectas:

Ingresa una letra:
#
Por favor ingresa una LETRA.
Ingresa una letra:
8
Por favor ingresa una LETRA.
Ingresa una letra:
@
Por favor ingresa una LETRA.
Ingresa una letra:
Probando
Por favor, introduce una letra.
Ingresa una letra:
a
¡Adivina la palabra!

+---+
|
0
|
|
|
|
=====

Letras incorrectas:
a
_ _ _ _ _
```

```
soyskipper@cristianhpmint: ~/Documentos/Learning python/Evidencias/Practica Final
Archivo Editar Ver Buscar Terminal Ayuda
Letras incorrectas:
a
_ _ _ _ _
Ingresa una letra:
e
¡Adivina la palabra!

+---+
|   |
| 0  |
|   |
|   |
+---+

=====

Letras incorrectas:
a
      e
_ _ _ _ _
Ingresa una letra:
i
¡Adivina la palabra!

+---+
|   |
| 0  |
|   |
|   |
+---+

=====

Letras incorrectas:
a i
      e
_ _ _ _ _
Ingresa una letra:
o
¡Adivina la palabra!

+---+
|   |
| 0  |
|   |
|   |
+---+
```

```
soyskipper@cristianhpmint: ~/Documentos/Learning python/Evidencias/Practica Final
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

Letras incorrectas:
a i
_ _ e _ _ o
Ingresa una letra:
u
¡Adivina la palabra!

+---+
|   |
| 0  |
|   |
|   |
+---+

=====

Letras incorrectas:
a i
_ u e _ _ o
Ingresa una letra:
cuervo
Por favor, introduce una letra.
Ingresa una letra:
c
¡Adivina la palabra!

+---+
|   |
| 0  |
|   |
|   |
+---+

=====

Letras incorrectas:
a i
c u e _ _ o
Ingresa una letra:
r
¡Adivina la palabra!

+---+
```

```
soyskipper@cristianhpmin: ~/Documentos/Learning python/Evidencias/Practica Final
```

```
Archivo Editar Ver Buscar Terminal Ayuda
```

```
| |  
| |  
=====
```

```
Letras incorrectas:  
a i  
c u e _ _ o  
Ingresa una letra:  
r  
¡Adivina la palabra!
```

```
+---+  
| |  
0 |  
| |  
| |  
=====
```

```
Letras incorrectas:  
a i  
c u e r _ o  
Ingresa una letra:  
v  
  
¡La palabra secreta es "CUERVO"! ¡Has ganado!
```

```
' . == == = '  
.-\:- /-.  
| (: : ) |  
' - | : | - '  
 \:: /  
' : : '  
 ) (  
' : : '  
 ~~~~~~ '
```

```
¿Quieres jugar de nuevo? (s)
```

```
soyskipper@cristianhpmin: ~/Documentos/Learning python/Evidencias/Practica Final
Archivo Editar Ver Buscar Terminal Ayuda
      .   .
     _ _ _ _ _
    /         \

¿Quieres jugar de nuevo? (s)
s
¡Adivina la palabra!

+---+
|   |
|   |
|   |
|   |
|   |
=====

Letras incorrectas:

Ingresa una letra:
jfajfgkajfgja
Por favor, introduce una letra.
Ingresa una letra:
/n
Por favor, introduce una letra.
Ingresa una letra:
a
¡Adivina la palabra!

+---+
|   |
|   |
|   |
|   |
|   |
=====

Letras incorrectas:

Ingresa una letra:

```

```
soyskipper@cristianhpmint: ~/Documentos/Learning python/Evidencias/Practica Final
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

=====
Letras incorrectas:
e l e f a _ e
Ingresa una letra:
n
¡Adivina la palabra!

+---+
|   |
|   |
|   |
|   |
|   |
=====

Letras incorrectas:
e l e f a n _ e
Ingresa una letra:
t

¡La palabra secreta es "ELEFANTE"! ¡Has ganado!

      _ _ _ _
     /      \
    /  .  .  \
   /  (  |  )  \
  /  |  |  |  |  \
 /  |  |  |  |  \
/   \  .  .  /
   \  .  .  /
    \  .  .  /
     \  .  .  /
      _ _ _ _
       (  )
        |  |
       _ _ _ _

¿Quieres jugar de nuevo? (s)
n]
soyskipper@cristianhpmint:~/Documentos/Learning python/Evidencias/Practica Final$
```

El juego termina al dar entrada otra letra distinta a 's'

Conclusiones:

Esta práctica nos ayudó a poner en práctica varios elementos vistos a lo largo de curso dentro de un programa de consola.

Aprendí a leer otros archivos externos importándolos como módulos y dar una menor carga y mejor lectura al programa principal. También aprendí sobre el manejo de listas y sus posiciones desde escoger una palabra, así como a separar caracteres, también en el uso de estos para las comparaciones dentro de los operadores lógicos.

Puse en práctica la programación funcional para el llamado de funciones y funciones superiores.

Por último me gustaría agradecerle profesor por la paciencia y siempre la resolución de mis dudas por lógicas que fueran o el aporte extra de distintas situaciones