

```

from six import print_ as xprint
from six.moves import input
import random

```

```

QUEMANDOSE = ['']

```

```

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

```

```

/|\  |
/\   |
====='''

```

#Conjunto de palabras para el juego

```

words = ('pato oso camello gato cobra jaguar lobo aguila perro coyote
'
         'mono rana canario perico leon llama tiburon raton conejo '
         'salmon tigre python ganso tortuga pavo ballena zebra buho '
         'hormiga cabra ').split()

```

```

class Ahorcado:
    def __init__(self, words):
        """
        Inicio de partida de juego
        Selecciona un palabra del listado de forma aleatoria
        """

        self._letra_error = ''
        self._letra_correcta = ''
        self._palabra_secreta = random.choice(words)
        self._juego_listo = False

    def _display_board(self):
        """Displiega el tablero de juego."""

        xprint(QUEMANDOSE[len(self._letra_error)])
        xprint()

        xprint('Letras fallidas:', end=' ')
        for letra in self._letra_error:
            xprint(letra, end=' ')
        xprint()

        blanks = '_' * len(self._palabra_secreta)

        # reemplaza los espacios con la letra adivinada
        for i in range(len(self._palabra_secreta)):
            if self._palabra_secreta[i] in self._letra_correcta:
                blanks = blanks[:i] + self._palabra_secreta[i] +
blanks[i+1:]

        # muestra la palabra secreta con espacios entre cada letra
        for letra in blanks:
            xprint(letra, end=' ')
        xprint()

```

```

def _que_adivino(self, ya_seleccionada):
    """
    Toma la seleccion del usuario.
    Valida que la entrada sea solamente una letra y no un caracter
especial
    Ademas valida que no se repida la seleccion
    """

    while True:
        xprint('Selecciona una letra.')
        guess = input().lower()
        if len(guess) != 1:
            xprint('Teclea solo una letra.')
        elif guess in ya_seleccionada:
            xprint('Esta letra ya la incluiste, selecciona otra
distinta.')
        elif guess not in 'abcdefghijklmnopqrstuvwxyz':
            xprint('Solo escribe una LETRA.')
        else:
            return guess

def _check_win(self):
    """
    Returns True si gano el jugador, de lo contrario False.
    Valida que el jugador adivino correctamente la palabra.
    """

    for i in range(len(self._palabra_secreta)):
        if self._palabra_secreta[i] not in self._letra_correcta:
            return False

    xprint('De lujo! La palabra secreta es "{0}"! ' #{} se
substituye por valor de 'self._palabra_secreta'
          'Tu ganaste!'.format(self._palabra_secreta))

    return True

def _check_lost(self):
    """Returns True si el jugador perdio, False de lo contrario
    Da aviso al jugador que sus oportunidades terminaron y no
    logro adivinar la palabra.
    """

    if len(self._letra_error) == len(QUEMANDOSE) - 1:
        self._display_board()

        missed = len(self._letra_error) #contador de errores
        correct = len(self._letra_correcta) #contador de aciertos
        word = self._palabra_secreta #palabra secreta

```

```

        xprint('Lo siento, no mas oportunidades!')
        xprint('Despues de {0} fallos y {1} aciertos, '
              'la palabra era: "{2}"'.format(missed, correct,
word))
        #{} son substituidos por los valores de variables 'missed,
correct, word'

        return True

    return False

def run(self):
    """Inicio de juego y reset de valores iniciales."""

    xprint('A H O R C A D O')

    while not self._juego_listo:
        self._display_board()

        letra_elegida = self._letra_error + self._letra_correcta
        guess = self._que_adevino(letra_elegida)

        if guess in self._palabra_secreta:
            self._letra_correcta = self._letra_correcta + guess
            self._juego_listo = self._check_win()
        else:
            self._letra_error = self._letra_error + guess
            self._juego_listo = self._check_lost()

def play_again():
    """
    Returns True si el jugador desea repetir juego, Falso de lo
    contrario
    """

    xprint('Quiere jugar otra vez? (si or no)')
    return input().lower() == 'si'

def main():
    """Secuencia principal."""

    current_game = Ahorcado(words)

    while True:
        current_game.run()
        if play_again():
            current_game = Ahorcado(words)
        else:

```

```
break
```

```
if __name__ == "__main__":  
    main()
```