FUNDAMENTOS DE PROGRAMACIÓN

Proyecto Final:

Juego del ahorcado en Python

Integrantes:

Danna Loaiza

Fernando Ortiz

Jose Gabela

Introducción e instrucciones del juego:
En este proyecto, el usuario debe adivinar una palabra seleccionada de manera aleatoria, debe escribir letra por letra hasta completar la palabra. Tiene 6 intentos para completar la palabra sino perderá el usuario. Escribimos el código dividido en varias funciones para que de esa manera se puedan ejecutar diferentes acciones.

Explicación del código:

En el juego de ahorcado, el código que realizamos demuestra que primeramente se importa una librería llamada 'import random' la cual se usa para la selección de palabras aleatorias, lo que es muy importante y necesario para que el juego del ahorcado funcione de manera correcta.

```
import random #Se usa para hacer la seleccion de palabras aleatoria

def bancopalarbras(): #Escoge una palabra para el juego de manera aleatoria
    banco = {
        1: 'banco',
        2: 'pasto',
        3: 'clero',
        4: 'clase',
        5: 'letra'
    }
    palabra = banco.get(random.randint(1, 5)) #
    return palabra
```

Posteriormente, se define la función con la variable "bancopalabras(): " e implementamos en esta función un banco de 5 palabras que permitirá al usuario tener opciones para poder jugar. Además, esta función escoge una palabra de entre esas 5 opciones de manera aleatoria.

Luego, hemos implantado otra función llamada 'intentos(fallos,respuesta)' . Básicamente, después de haberse seleccionado la palabra aleatoria (la cual el usuario no puede ver) esta función tiene el atributo de preguntarle al usuario las letras que cree que están en la palabra, si el usuario falla se suma un punto a la variable fallos.

```
def intentos(resp_usuario,fallos, respuesta):
    if resp_usuario in respuesta:
        return fallos
    else:
        fallos += 1
        return fallos
```

La función marcador es aquella que actualiza el juego con las letras que escriba el usuario, esto de tal manera que el usuario pueda ver en qué letra acertó, usamos listas para verificar si la letra que puso se encuentra y si es correcta que se sustituya en la posición. Por último convierte la lista actualizada en una cadena para que se guíe con las letras que ya colocó.

```
def marcador(resp_usuario, respuesta, palabra):
    respuesta = list(respuesta)
    if resp_usuario in respuesta:
        indice_letra = respuesta.index(resp_usuario)
        palabra[indice_letra] = resp_usuario
        marc = ' '.join(palabra)
        return marc
    else:
        marc = ' '.join(palabra)
        return marc
```

Para hacer la parte visual de los errores, añadimos una función que pone cada parte del cuerpo con cada error que cometa el usuario. Utilizamos una serie de if para imprimir cada parte del cuerpo.

```
def visuales(fallos):
    if fallos == 0:
        print(' +_____+')
        print(' |     |')
        print(' |     ')
        print(' |     ')
        print(' |     ')
        print(' |     ')
        print(' |      ')
        print(' ---------')
    elif fallos == 1:
        print(' +_____+')
        print(' |     |')
        print(' |     0')
        print(' |     ')
        print(' |      ')
        print(' |     ')
        print(' |      ')
        print(' ---------')
```

Finalmente, incluimos la función ‘denuevo(jugar)’. Esta función le pregunta al usuario si está interesado en seguir jugando, si su respuesta es si, el código vuelve a escoger otra palabra aleatoria e inicia nuevamente el proceso. Ahora bien, si su respuesta es no, el código no vuelve a inicializarse y el juego acaba ahí.

```
def denuevo(jugar): #Pregunta si el usuario quiere seguir jugando o no
    a = input('Quiere seguir jugando?: ')
    if a == 'si':
        jugar = True
    elif a == 'Si':
        jugar = True
    else:
        jugar = False
    return jugar
```