TPI Programación

Profesores: Ivana garabello y Francisco Becerra

Integrantes: Fermin Lopez, Gonzalez Javier, Emanuel Dequino, Tomas Oscar Andre

Juego: Adivinar el número

*Problema: Realizar un juego a través de python.
*Objetivo del juego: El objetivo es adivinar un número secreto que ha sido elegido aleatoriamente por el programa. Este número está entre 1 y 20
*Reglas: Ingresar un número del 1 al 20, ingresar solo valores numéricos.
*como jugar: Elegir el juego en el menú principal, ingresar un numero del 1 al 20,si el número que ingresaste es menor que el número secreto, recibirás un mensaje indicandolo, si el número que ingresaste es mayor que el número secreto, también recibirás una pista, tratar de adivinar el número con las pistas de mayor y menor, una vez adivinado el número se te mostrará cuantos intentos necesitaste para adivinarlo y por último se te preguntará si quieres jugar de vuelta
*Pseudocódigo:

```
:Proceso Adivina_Numero
    Definir numero_para_adivinar, numero, intentos Como Entero
    Definir jugar Como Caracter

    // Función que inicia el juego
    numero_para_adivinar <- Aleatorio(20) + 1  // Genera un número aleatorio entre 1 y 20
    intentos <- 0
    Escribir "¡¡¡Descubre el número del 1 al 20!!!"

    Mientras Verdadero Hacer
        Intentar
            Escribir "Ingrese un número:"
            Leer numero
            intentos <- intentos + 1

            Si numero < numero_para_adivinar Entonces
                Escribir "-" * 45
                Escribir "-> EL número ", numero, " es menor al número secreto."
                Escribir "-" * 45
            Sino Si numero > numero_para_adivinar Entonces
                Escribir "-" * 45
                Escribir "-> EL número ", numero, " es mayor al número secreto."
                Escribir "-" * 45
            Sino
                Escribir "◙" * 30
                Escribir "¡¡Felicidades, adivinaste el número!!"
                Escribir "Número de intentos: ", intentos
```

```
            Escribir "◙" * 30
            Romper
        Fin Si

    Capturar Error Como ValueError
        Escribir "Ingrese un valor numérico válido:"
    Fin Intentar
Fin Mientras

// Función para preguntar si el usuario quiere jugar de nuevo
Hacer
    Escribir "Ingrese:"
    Escribir "1- Si quiere jugar de nuevo"
    Escribir "2- Si no quiere jugar más"
    Leer jugar

    Si jugar == '1' Entonces
        Iniciar de nuevo el proceso Adivina_Numero
    Sino
            Si jugar == '2' Entonces
        Escribir "Gracias por jugar!"
        Romper
    Sino
        Escribir "☼" * 20
        Escribir "Ingrese solamente 1 o 2"
        Escribir "☼" * 20
    Fin Si
Mientras jugar != '2'
Fin Proceso
```

JUEGO: PIEDRA PAPEL O TIJERA

Vidas: Ambos comienzan con 3 vidas.

Juego: En cada turno, el jugador elige entre Piedra, Papel o Tijera, mientras que la PC elige de forma aleatoria.

Reglas:
Piedra gana a Tijera.
Papel gana a Piedra.
Tijera gana a Papel.

El juego termina cuando uno de los dos queda sin vidas.

Se ofrece la opción de volver a jugar al final vidas.

PSEUDOCÓDIGO:

```
Proceso Piedra_Papel_Tijera
    Definir vida_usuario, vida_pc, opcion, aleatorio Como Entero
    Definir usuario, pc Como Cadena
    Definir jugar_nuevamente Como Caracter

    Mientras Verdadero Hacer
        vida_usuario <- 3
        vida_pc <- 3

        Mientras vida_usuario > 0 Y vida_pc > 0 Hacer
            aleatorio <- Azar(3) - 1 // Generar número aleatorio entre 0 y 2
            pc <- ""

            Escribir "----------------------------------------"
            Escribir "Vidas restantes de 'Jugador': ", vida_usuario, "♥"
            Escribir "Vidas restantes de 'PC': ", vida_pc, "♥"
            Escribir "----------------------------------------"

            // Comprobar la entrada del usuario
            Mientras Verdadero Hacer
                Escribir "¿Qué eliges?"
                Escribir "1) Piedra"
                Escribir "2) Papel"
                Escribir "3) Tijera"
                Leer opcion

                Si opcion >= 1 Y opcion <= 3 Entonces
                    Romper
                SiNo
                    Escribir "Por favor, ingresa un número entre 1 y 3."
                FinSi
            FinMientras

            // Asignar la elección del usuario
            Segun opcion Hacer
                1: usuario <- "piedra"
                2: usuario <- "papel"
                3: usuario <- "tijera"
            FinSegun
            Escribir "Jugador eligió: -> ", usuario

            // Asignar la elección de la PC
            Segun aleatorio Hacer
                0: pc <- "piedra"
                1: pc <- "papel"
                2: pc <- "tijera"
```

```
    FinSegun
    Escribir "PC eligió: -> ", pc

    Escribir "**************************************"

    // Comparar elecciones
    Si (pc = "piedra" Y usuario = "papel") O
       (pc = "papel" Y usuario = "tijera") O
       (pc = "tijera" Y usuario = "piedra") Entonces
        Escribir "¡¡GANASTE!!"
        vida_pc <- vida_pc - 1
    FinSi

    Si (pc = "papel" Y usuario = "piedra") O
       (pc = "tijera" Y usuario = "papel") O
       (pc = "piedra" Y usuario = "tijera") Entonces
        Escribir "¡¡PERDISTE!!"
        vida_usuario <- vida_usuario - 1
    FinSi

    Si pc = usuario Entonces
        Escribir "EMPATE"
    FinSi

    Escribir "**************************************"
FinMientras

// Resultado del juego
Si vida_usuario = 0 Entonces
    Escribir "oooooooooooooooooooooooooooooo"
    Escribir "JUEGO TERMINADO!! GANA PC"
    Escribir "oooooooooooooooooooooooooooooo"
FinSi

Si vida_pc = 0 Entonces
    Escribir "oooooooooooooooooooooooooooooo"
    Escribir "JUEGO TERMINADO!! ¡GANASTE!"
    Escribir "oooooooooooooooooooooooooooooo"
FinSi

// Rejugar
Repetir
    Escribir "¿Quieres jugar nuevamente? (S/N)"
    Leer jugar_nuevamente
    jugar_nuevamente <- Mayusculas(jugar_nuevamente)
Hasta Que jugar_nuevamente = "S" O jugar_nuevamente = "N"
```

```
        Si jugar_nuevamente = "N" Entonces
            Escribir "¡Gracias por jugar! :)"
            Romper
        FinSi
    FinMientras
FinProceso
```

JUEGO: TIRA LA MONEDA (cara o cruz)

problema: realizar un juego en python
objetivo del juego: el objetivo del juego es intentar adivinar en qué posición caerá la moneda (cara o cruz)
como jugar: ingresar uno de los dos valores pedidos…cara o cruz, sin errores al escribirlo o con otros caracteres erroneos…espera a que la moneda gire y cuando caiga el juego te dirá si ganaste o no.

pseudocodigo:

```
Algoritmo TiraLaMoneda
    Escribir "¡Bienvenido al juego de cara o cruz!"

    // Ciclo principal del juego
    Repetir
        // Solicita al usuario que elija "cara" o "cruz"
        Repetir
            Escribir "Elige: cara o cruz: "
            Leer eleccion
            eleccion = Minusculas(eleccion)

            // Verifica si la entrada es válida
            Si eleccion <> "cara" y eleccion <> "cruz" Entonces
                Escribir "Opción inválida. Por favor ingresa 'cara' o 'cruz'."
            FinSi
        Hasta Que eleccion = "cara" o eleccion = "cruz"

        // Simula el lanzamiento de la moneda
        si Aleatorio(0, 1) = 0 Entonces
            resultado = "cara"
        SiNo
            resultado = "cruz"
        FinSi
        Escribir "La moneda cayó en: ", resultado

        // Verifica si el jugador ganó o perdió
        Si eleccion = resultado Entonces
            Escribir "¡Ganaste!"
        SiNo
            Escribir "Perdiste."
```

```
        FinSi

        // Pregunta si el jugador quiere otra partida
        Repetir
            Escribir "¿Quieres jugar otra partida? (s/n): "
            Leer jugar_otra
            jugar_otra = Minusculas(jugar_otra)

            // Verifica si la entrada es válida
            Si jugar_otra <> "s" y jugar_otra <> "n" Entonces
                Escribir "Opción inválida. Por favor ingresa 's' o 'n'."
            FinSi
        Hasta Que jugar_otra = "s" o jugar_otra = "n"

    Hasta Que jugar_otra = "n"

    // Mensaje de despedida
    Escribir "Gracias por jugar. ¡Hasta la próxima!"
FinAlgoritmo
```

Juego: Adivina la palabra

Problema:

El código de este juego esta echo para un juego de adivinar palabras, en el que un jugador tendrá que ingresar una palabra y otro jugador intentar adivinarla ingresando letras para descubrir la palabra oculta. Esta tiene un número limitado de intentos, en el que por cada letra incorrecta los intentos disminuyen en 1 y cundo se adivine una letra esta se mostrara en la posición que corresponda de la palabra. Al ganar o perder, el resultado de guarda en archivo de texto.

Guía de uso:

1. **Inicio del juego**: Se pide al usuario que ingrese una palabra secreta para comenzar el juego, la cual será la palabra a adivinar.

2. **Intentar adivinar la palabra**: El jugador ira ingresando letras para comprobar si estas están en la palabra oculta.

    - Si la letra ingresada está en la palabra oculta, se revela la letra y su posición en la palabra.

- Si la letra ingresada ya fue ingresada anterior mente, se indica con un mensaje y no se restan intentos.

- Si es incorrecta, el número de intentos se reduce en uno por cada letra incorrecta ingresada.

3. **Fin del juego:**

   - Si se adivinan todas las letras, el jugador gana.

   - Si el jugador se queda sin intentos pierde, y la palabra oculta se muestra.

4. **Resultados:** El resultado del juego ya sea el de ganar o perder, se guarda en un archivo llamado **resultado_juego.txt.**

Razonamiento de resolución

Se utilizaron 3 funciones:

- **Función progreso palabra (palabra, letras_adivinadas):**

  - Esta función recibe la palabra oculta y las letras que el jugador adivina.

  - Se encarga de ocultar la palabra con guiones bajos (‘_’) en las letras de la palabra aun no son adivinadas, e ir mostrando las letras adivinadas en la posición que corresponda.

- **Función guardar_resultado (juego_info):**

  - Esta función recibe un texto **(juego_info)** que contiene el resultado del juego (Gano, Perdió), y la palabra que se intentó adivinar.

  - La información se guarda en un archivo llamado **resultado_juego.txt,** en una línea distinta por cada juego o palabra nueva.

- **Función principal juego_adivina_palabra ():**

Esta es la función principal y del juego y contiene lo siguiente:

- **Configuración inicial:** Donde se pide al jugador que ingrese la palabra secreta y la convierte en mayúscula para estandarizar el juego. Se asegura

que la palabra ingresada tenga más de dos letras y solo contenga caracteres alfabéticos.

- **Gestión de intentos:** Se inicializa el contador de los intentos permitidos en (5) y se crean dos conjuntos vacíos, uno para almacenar las letras adivinadas correctamente y otro para las incorrectas.

- **Bucles del juego:** Mientras el jugador tenga intentos disponibles:

    - Se pide ingresar una letra y se le da pasaje a mayúscula.

    - Se verifica si letra ya fue adivinada o es nueva. Si es nueva la clasifica como (Correcta o Incorrecta) mostrando el progreso actualizado.

    - Si la letra está en la palabra, se agrega al conjunto (**letras_adivinadas)**, y si no, se restan intentos y se agrega al conjunto (**letras_erradas).**

    - Si el jugador adivina todas las letras de la palabra, el juego finaliza con un mensaje de victoria.

- **Finalización:** Si el jugador adivina la palabra o se queda sin intentos, se guarda el resultado en el archivo y se pregunta si desea jugar nuevamente.

Repositorio: https://github.com/TomasAndre/TpiLaboratorio1_2024.git