



Tecnicatura Universitaria en Programación

Laboratorio de Computación I

Trabajo Práctico Integrador

INTEGRANTES:

- Brusadin Leonardo Andrés
- Charras Luis
- Bolatti Nicolás

PROFESOR: LEANDRO SALDIAS.

FECHA DE ENTREGA: 27/11/2022

1. **Tarea 1:** Elección de los juegos.

- a. Brusadin Leonardo Andrés: eligió el Black Jack.
- b. Charras Luis: eligió la Ruleta
- c. Bolatti Nicolás: eligió el Tragamonedas

2. **Tarea 2:** Análisis del enunciado.

El programa comienza con un menú donde se le pregunta al usuario que juego quiere elegir.

En esta sección utilizamos:

ENTRADA	Nombre variable	Descripción	Tipo
	Decisión	Se le pregunta al usuario que juego quiere utilizar	C
Salidas			
Datos			
Procedimiento/Estrategia de resolución	En la explicación sobre el juego de la ruleta hacemos referencia a los ciclos utilizados en este apartado		

A) **BLACKJACK**

⇒ Estructuras utilizadas en la creación del juego:

- ✓ Estructuras selectivas
- ✓ Datos, Estructuras Secuenciales y Estructuras Selectivas

- ✓ Estructuras Repetitivas
- ✓ Arrays
- ✓ Librería Ramdon
- ✓ Librería Time
- ✓ Python

El **Blackjack**, también llamado **veintiuno**, es un juego de cartas, propio de los casinos con una o más barajas inglesas de 52 cartas sin los comodines, que consiste en sumar un valor lo más próximo a 21 pero sin pasarse. En un casino cada jugador de la mesa juega únicamente contra el crupier, intentando conseguir una mejor jugada que este. El crupier está sujeto a reglas fijas que le impiden tomar decisiones sobre el juego. Por ejemplo, está obligado a pedir carta siempre que su puntuación sume 16 o menos, y obligado a plantarse si suma 17 o más. Las cartas numéricas suman su valor, las figuras suman 10 y el As vale 11 o 1, a elección del jugador. En el caso del crupier, los Ases valen 11 mientras no se pase de 21, y 1 en caso contrario. La mejor jugada es conseguir 21 con solo dos cartas, esto es con un As más carta de valor 10. Esta jugada se conoce como Blackjack o 21 natural. Un Blackjack gana sobre un 21 conseguido con más de dos cartas.

Entradas	Nombre variable	Descripción	Tipo
	<pre>valoresJugador[i] = int(input(f"Eliza qué valor le da al As. Las opciones son 1 y 11: "))</pre>	Asigna a un As dentro de las cartas del jugador el valor entero de 11 o 1	E
	<pre>decision = int(input(f"Ingrese 1 para pedir otra carta, 2 para detenerse: "))</pre>	Permite mantener el ciclo en funcionamiento si recibe el valor 1 y lo interrumpe si recibe 2.	E
Salidas	puntosCrupier	Almacena los puntos del juego del crupier	E
	manoCrupier	Muestra una cadena formada por los elementos de manoCrupier en una sola línea	Lista

	x	Representa un elemento de la lista manoCrupier que alimenta la función print	cadena
	manoJugador	Muestra una cadena formada por elementos de manoCrupier en una sola línea	Lista
Datos			
Procedimiento/Estrategia de resolución	<ul style="list-style-type: none"> 		

Funcionamiento del algoritmo:

```
def barajaNueva(numero):
```

Esta función toma como argumento un entero. Si el entero es igual a 1, crea un nuevo mazo de cartas, donde cada carta se crea como una combinación de los elementos de

palos = ["♠", "♥", "♣", "♦"] y valores = ['A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K']

Cada palo se almacena en una sub lista con su nombre dentro de la matriz baraja, que es la salida de la función

```
def repartir(baraja, mano, valor):
```

La función toma como entradas 3 listas. Luego selecciona una carta al azar y la elimina de la baraja. La salida son las 3 variables de entrada actualizadas.

```
def valorCarta(carta):
```

Toma una carta como cadena de caracteres y devuelve el valor numérico entero asociado a la misma.

```
def esBlackJack(puntos, mano):
```

Evalúa los puntos de la mano y las cartas que la componen para determinar si es Blackjack. Si la mano suma 21 y está formada por 2 cartas, la mano es Blackjack y su salida es 1. De lo contrario devuelve 0.

```
def nuevaPartida(k):
```

La función toma como entrada una variable que usamos como condición para crear un ciclo repetitivo y devuelve el valor que mantiene el ciclo funcionando.

```
partida = 1
while partida == 1:
    otraPartida = 0
    partida = 0
    baraja = barajaNueva(1)
```

Ambas variables (partida y otraPartida) son usadas por la función nuevaPartida() para crear y manipular el ciclo repetitivo del juego. Además, se crea una nueva baraja al comienzo de cada partida.

```
manoCrupier = []
valoresCrupier = []
```

```
puntosCrupier = 0
```

Inicializo las variables que almacenarán las cartas del crupier, su valor y el puntaje de la mano.

```
repartir(baraja, manoCrupier, valoresCrupier)
```

```
repartir(baraja, manoCrupier, valoresCrupier)
```

Le reparto 2 cartas al crupier.

```
while juegoCrupier == 0:
```

Ciclo repetitivo donde el crupier juega su mano hasta que juegoCrupier cambie de valor. El As vale 11 si tiene puntos < 21 y 1 si pasa de 21. Si tiene puntos <= 16 pide carta, pero se detiene cuando los puntos son >= 17.

```
blackJackCrupier = esBlackJack(puntosCrupier, manoCrupier)
```

Chequeamos si la mano del crupier es Blackjack.

```
print(f"Los puntos del crupier son: {(puntosCrupier)}\n")
    if puntosCrupier > 21:
        print(f"\n GANASTE!\n El crupier se pasó de 21\n")
        partida = nuevaPartida(otraPartida)
        if partida == 1:
            continue
        elif partida == 2:
            break
```

Controlamos que la mano del crupier no pase los 21 puntos. Si lo hace, el jugador gana la mano y se llama a la función nuevaPartida(otraPartida) para decidir si el ciclo vuelve a comenzar o salimos del juego.

```
partida = nuevaPartida(otraPartida)
if partida == 1:
    continue
elif partida == 2:
    break
```

Este bloque, que permite al jugador iniciar una nueva partida o regresar al menú principal, se repetirá luego de cada resultado posible.

```
while juegoJugador == 0:
```

Creamos el ciclo para el juego del jugador. Éste se repite hasta que juegoJugador cambie su valor. Dentro del mismo le permitimos al usuario decidir si valúa al As en 11 o en 1 en caso de contar con uno en su mano. Luego, le damos la opción de pedir otra carta o de detenerse. El usuario deberá elegir qué opción le conviene de acuerdo a su puntaje y al del crupier.

```
blackJackJugador = esBlackJack(puntosJugador, manoJugador)
```

```
if puntosJugador > 21:
    print(f"\n PERDISTE!\n Te pasaste de 21\n")
```

Chequeamos si el puntaje del jugador se corresponde con un Blackjack. Luego verificamos que su puntaje no exceda los 21 puntos. Si lo hace, el jugador pierde.

```
If puntosJugador==puntosCrupier and blackJackCrupier==blackJackJugador==1:
    print(f"\n El jugador y el crupier EMPATAN!\n Ambos tienen BLACK JACK!")
```

Si tienen igualdad de puntos, pero además ambos tienen Blackjack, empatan.

```
elif blackJackCrupier == 1 and blackJackJugador != 1:
    print(f"\n El crupier tiene \n BLACKJACK!\n PERDISTE!")
```

```
elif blackJackJugador == 1 and blackJackCrupier != 1:  
    print(f"\n El jugador tiene \n BLACKJACK!\n GANASTE!")
```

Si uno de los dos tiene Blackjack y el otro no, gana el Blackjack.

```
elif blackJackCrupier == blackJackJugador != 1:  
    if puntosCrupier == puntosJugador:  
        print(f"\n EMPATAN!\n Ambos tienen {puntosCrupier} puntos.")  
elif puntosJugador > puntosCrupier:  
    print(f"\n GANASTE!\n")  
    elif puntosCrupier > puntosJugador:  
        print(f"\n El crupier gana con {puntosCrupier} puntos.\n  
PERDISTE!")
```

Como podemos ver, cuando los resultados son diferentes al Blackjack, el resultado del juego es bastante intuitivo. A igualdad de puntos, empatan. En caso de tener diferente puntaje, el que consiga el mayor puntaje sin exceder los 21 puntos, gana.

B) RULETA EUROPEA

La **ruleta europea**, también llamada ruleta francesa, es una de las dos variantes tradicionales del juego de la ruleta. La otra variante es la ruleta americana, la cual consta de reglas similares, pero con ligeros cambios. En su momento, la rueda tenía 31 números, aunque lo habitual hoy en día es que tenga 37 en Europa y 38 en los Estados Unidos

El origen de este popular juego de casino se remonta al siglo XVII. Su creador original, un matemático de nombre Blaise Pascal, la concibió en su forma más primitiva.

Su nombre en francés, Roulette, significa rueda pequeña. A partir de este hecho, el juego de la ruleta europea empezó a hacerse cada vez más famoso. En un principio, estaba relacionada con experimentos acerca de la probabilidad, pero poco a poco fue tomando la forma del popular juego de azar que conocemos hoy en día.

⊃ Estructuras utilizadas en la creación del juego:

- ✓ Estructuras selectivas
- ✓ Datos, Estructuras Secuenciales y Estructuras Selectivas
- ✓ Estructuras Repetitivas
- ✓ Arrays

- ✓ Librería Ramdon
- ✓ Librería Time
- ✓ Python

⇒ **Funcionamiento de algoritmo:**

En el algoritmo de la ruleta utilizamos un ciclo WHILE y en el comienzo se le pide al usuario que elija alguna de las distintas opciones de juego; Si bien la ruleta cuenta con distintas variantes de juego las elegidas para este caso fueron:

- ✓ Apostar a números elegidos individualmente.
- ✓ Números pares e impares.
- ✓ Colores : negro o rojo

I. **Apostar a números elegidos individualmente:**

Si el usuario escribe el número uno (1) el programa saltara a una secuencia selectiva donde podrá elegir qué cantidad de números quiere jugar. Para eso utilizamos un ciclo FOR para saber la longitud del array con el propósito de llenar un vector con los números que elija el usuario.

Luego, con la librería ramdon, que actuara como si fuera la bola de la ruleta, proporcionara un número como resultado que se comparara con los números elegidos por el usuario. En caso de que el número ganador este dentro del array ingresado por el usuario el programa devolverá el texto **“FELICIDADES USTED GANO”**, en caso contrario se mostrará **“USTED PERDIO”**.

Finalizado este ciclo, se ingresara en una nueva estructura selectiva donde preguntara si quiere regresar al menú de la ruleta o directamente salir y retornar al menú principal.

II. **Números pares e impares**

En caso de que el usuario decida mantenerse dentro de la ruleta pero cambiar de modalidad, tiene la opción de escribir el numero dos (2) y en este caso, el programa entrara en una interfaz nueva en la que la modalidad de juego ser para apostar a números pares o impares de la ruleta.

Para esto, se debe ingresar el número 1 o el 2 dependiendo si es par o no; Una vez seleccionado el algoritmo hará uso nuevamente de la función random para dar un número de la ruleta. En este caso, se utilizan los ciclos condicionales **“IF” “ELIF” & “ELSE”**.

Si el usuario apostó a números pares y el número que salió de la función random está dentro del array de los números pares el programa escribirá **"FELICIDADES USTED GANO"** en caso contrario el programa escribirá **"USTED PERDIO"**.

Finalizado el minijuego, entrará nuevamente nueva estructura selectiva donde preguntará si se quiere jugar otro juego de la ruleta o directamente salir y retornar al menú principal.

III. Colores : negro o rojo

Si el usuario escribe el número tres, habrá entrado en la última modalidad y el programa ingresará en una interfaz nueva donde pedirá si se quiere apostar a números negros o rojos de la ruleta. Para ello, se deberá ingresar el número 1 o el 2 dependiendo a que queramos apostar.

Una vez seleccionado, el algoritmo hará uso de la función random para dar un número de la ruleta. Luego se utilizan los ciclos condicionales "IF" "ELIF" & "ELSE". Si el usuario apostó a números negros y el número que salió de la función random está dentro del array de los números negros el programa escribirá **"FELICIDADES USTED GANO"** en caso contrario el programa escribirá **"USTED PERDIO"**.

A los fines de poder aclarar los arrays utilizados y su organización adjuntamos la siguiente imagen:



Entradas	Nombre variable	Descripción	Tipo
	DECISION2	Dentro del menú de la Ruleta, es la que le lleva a las distintas modalidades de ese juego	C

	CANTIDADNUMA	Cuando se elige números a elección, representa la longitud de la cantidad de números que el usuario quiere jugar	N
	Desicioncolores	Esta variable se usar cuando el usuario elige el color en caso que haya elegido esa modalidad	N
	NUMEROPAS	Se usa para elegir los números que se quieren jugar y llenar el array	N
	DECISION4	Cuando el usuario eligio par o impar, es la variable que selecciona uno o dos	N
Salidas	RANDOMRULETA	Representa la bola de la ruleta, o sea el número que sale	N
	NUMEROSNUMA	Es el array de los números elegidos por el usuario que se compara con la variable RANDOMRULETA (es decir la de los números que salió)	N
	RandomPARES	Es la variable que se utiliza como la bola de la ruleta	N
	Pares	Arrays de los números pares	N
	Impares	Arrays de los números impares	N
	Todos	Array de todos los números	N
	Rojos	Arrays de los números rojos	N
	Negros	Arrays de los números negros	N
	RandomROJO	Representa la bola de la ruleta	N
	RandomNEGRO	Representa la bola de la ruleta	N
Datos			

Procedimiento/Estrategia de resolución	El procedimiento utilizado se detallo arriba
---	--

C) TRAGAMONEDAS

⊇ Estructuras utilizadas en la creación del juego:

- ✓ Estructuras selectivas
- ✓ Datos, Estructuras Secuenciales y Estructuras Selectivas
- ✓ Estructuras Repetitivas
- ✓ Arrays
- ✓ Librería Ramdon
- ✓ Librería Time
- ✓ Python

⊇ Funcionamiento de algoritmo:

Este juego tiene una sola modalidad de jugado, por lo que si el usuario en el menú principal elige la opción correspondiente al tragamonedas, el cuerpo a ejecutarse del código será uno solo.

En este algoritmo utilizamos una array que lo completamos con símbolos similares a los del tragamonedas y tres arrays que se van completando con símbolos; para esto utilizamos la librería Ramdon.

Para llenar cada array con símbolos aleatorios utilizamos un ciclo "FOR" con el rango de la longitud de los 3 arrays y por cada paso del ciclo van llenando los tres arrays al mismo tiempo con un símbolo aleatorio en el índice del array del ciclo.

Una vez que los tres arrays estén completos pasaran a una estructura selectiva que compara los valores de los índices del segundo array, en caso de ser todos iguales los símbolos el programa escribirá **“FELICIDADES USTED GANO”** en caso contrario el programa escribirá **“USTED PERDIO”**. Luego se ejecuta la estructura selectiva donde preguntará si se quiere jugar otro minijuego.

Entradas	Nombre variable	Descripción	Tipo
	Desiciontirada	Solo se utiliza cuando se presiona el 1 para realizar una tirada en el tragamoneda	N
Salidas	Array1	Es el primer array del juego	N
	Array2	Es el segundo array del juego	N
	Array3	Es el tercer array del juego	N
Datos			
Procedimiento/Estrategia de resolución	El procedimiento fue explicado arriba		

3. **Tarea 3:** Elección de programa

Para la creación y ejecución de los códigos todos los integrantes decidimos utilizar el programa Python y el IDE VISUAL CODE.

4. **Tarea 4:** Gestor de tareas.

En cuanto a la utilización del gestor se decidió:

- Usar MeisterTask.
- Creamos el Proyecto y creamos tareas para darle seguimiento a nuestro proyecto.

5. **Tarea 5:** Repositorio.

En cuanto al repositorio se decidió:

- a. Utilizar GitHub.
- b. Creamos el repositorio y cargamos los proyectos a repositorio.
- c. Actualizamos versiones hasta llegar a la versión final del proyecto.

6. **Tarea 6:** Prueba de escritorio.

- a. Tragamonedas
- b. Blackjack
 - i. Ingresamos a Casino.py
 - ii. Ejecutamos el archivo
 - iii. Seguimos las instrucciones del juego
 - iv. Probamos todas las combinaciones

c. Ruleta

7. **Tarea 7:** Documentación de referencia

Para el trabajo se utilizó:

- ✓ Bibliografía de la materia (Fundamentos de Programación).
- ✓ Filminas correspondientes a las estructuras utilizadas
- ✓ La mayoría de los integrantes ya contaban con conocimiento previo del lenguaje de programación Python.