

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Laboratorio IPC1 sección A



## **Creación de programa en consola Pacman en lenguaje de programación Python**

Nombre: Carlos Josué Calderón Cabrera

Carnet: 201314753

Fecha: 31/03/2022

## Índice

Introducción.....	3
Objetivo.....	4
Procedimientos.....	5-10

## **Introducción**

El Presente manual dará una breve explicación del uso de la aplicación desarrollada por el estudiante con la finalidad de poder interpretar el código utilizado, dicha aplicación conocida como fue creada en el lenguaje de programación Python.

## **Objetivo**

Implementar el uso de algoritmos y métodos para el desarrollo de aplicación de consola en el lenguaje de programación Python. La aplicación a desarrollar es el conocido juego de pacman y se realizó con la mayor cantidad de similitudes que pueden permitirse por la consola.

## Procedimiento

1. Creación de un menú el cual estará dentro de un ciclo while con la finalidad de que sea repetitivo siempre que se cumplan dichas condiciones se validara que la opción opc no sea igual a 3 y se encuentre en un rango mayor a 0 y menor a 4.

```
while opc != 3 and opc > 0 and opc<4:
    print("*****Bienvenido***** \n***1. Iniciar Juego*** \n***2. Historial***** \n***3. salir*****")
    opc = int(input("ingrese una opcion: "))
    if opc < 0 and opc > 4:
        opc = int(input("ingrese una opcion valida: "))
    if opc == 1: ...
    if opc == 2: ...
```

2. Si el usuario ingresa la opción 1 se instancia el método realizado para dicha opción el cual consta de la creación de múltiples variables globales las cuales tienen finalidad de poder ser de utilidad a lo largo de programa.

```
if opc == 1:
    n = n + 1
    opcion1()
```

```
def opcion1():##metodo para la opcion 1
    global nom, c, fil, col, tablero, pos, pts, contd, contq, contres, hist
```

3. Luego de creadas las variables globales se crearan las variables objeto con la finalidad de optimizar en el código el ingreso de las mismas siendo estas las siguientes: p10 = [ @], p15 = [ \$], q10 = [ #] y pacman = [:v].

```
pacman="[:v]"
p10 = "[ @]"
p15 = "[ $]"
q10 = "[ #]"
```

4. Se creó una variable de tipo listado múltiple llamado "hist" la cual servirá para guardar los datos de cada partida iniciada.

```
hist=["[ ]"for i in range(3*n)]
```

5. Luego se procede a darle un valor a las variables de datos del usuario con un Input y respectivo casteo a las variables que no serán de tipo cadena como el carnet y las filas y columnas del tablero.
6. Durante el ingreso de valores de filas y columnas se consideró el mínimo establecido de 7x7 a lo cual se implementó un ciclo while para evitar un ingreso menor al establecido.

<pre> nom = input("ingrese su nombre: ") c = int(input("ingrese su carnet: ")) fil = int(input("ingrese no filas: ")) while(fil &lt; 7):     fil = int(input("ingrese no filas mayor a 7: ")) col = int(input("ingrese no columnas: ")) while(col &lt; 7):     col = int(input("ingrese no columnas mayor a 7: ")) </pre>	5
	6

7. Se creó una variable “tablero” la cual tiene como finalidad ser nuestro tablero de juego en el cual se colocaran los comodines de ponderación y el personaje pacman dicho tablero estará lleno de espacios vacíos denotados de la siguiente forma “[ ]”.
8. Con la finalidad de crear una posición para ubicar y desplazar a nuestro icono “pacman” se le brinda a la variable “pos” un valor random el cual estará en un rango desde 0 hasta el resultado de la multiplicación de filas y columnas restándole una unidad a dicha multiplicación.
9. Luego de obtener el resultado de nuestra variable “pos” se procede a recorrer el tablero colocar a nuestro personaje en la celda que tenga el mismo valor numérico que “pos”.

<pre> tablero = ["[ ]" for i in range(fil*col)] pos = random.randint(0, len(tablero)-1) if tablero[pos]=="[ ]":     tablero[pos]=pacman </pre>	7	
	8	
	9	

10. Procedemos a darle un valor random a las variables “contd”, “contq” y “contres” las cuales funcionaran a la vez como un contador de la cantidad de apariciones de los iconos de puntaje con la cantidad establecida mínima y el porcentaje de aparición en el tablero.

11. Ya con los valores establecidos se procede a ingresar la cantidad obtenida de cada icono en el tablero siempre que el espacio en el tablero este vacío.
12. Luego de haber finalizado la colocación del tablero y su contenido procedemos a realizar la modalidad de juego con un while verificamos que se realizaran movimientos siempre que la variable booleana "continuar" sea verdadera.

```
contd = random.randint(8, (len(tablero)*30)//100)
contq = random.randint(6, (len(tablero)*20)//100)
contres = random.randint(8, (len(tablero)*25)//100)
continuar = True
pts = 10
for d in range(0, contd):
    posd = random.randint(0, len(tablero)-1)
    while tablero[posd] == "[ ]":
        tablero[posd]=p10
for d in range(0, contq):
    posq = random.randint(0, len(tablero)-1)
    while tablero[posq] == "[ ]":
        tablero[posq]=p15
for d in range(0, contrs):
    posr = random.randint(0, len(tablero)-1)
    while tablero[posr] == "[ ]":
        tablero[posr]=q10
```

13. Durante la ejecución de los movimientos se procede a imprimir el nombre del jugador, su carnet y la cantidad de puntos obtenidos durante la partida y se verificara si los puntos son mayores o iguales a 30 o si los puntos son menores o igual a 0 cumpliéndose cualquiera de las dos opciones anteriores para finalizar la partida.

```
while continuar == True:
    print(f"nombre: {nom} carnet: {c} puntos: {pts}")
    if pts >= 30:
        print("ganaste")
        pts=30
        continuar = False
    elif pts <= 0:
        print("perdiste")
        pts=0
        continuar = False
```

14. Se procede a imprimir en consola el tablero y pedirle al usuario que ingrese un movimiento dicho movimiento quedara registrado en la variable mov y se

validara que el movimiento sea “A”, “S”, “D”, “W” o “M”, siendo “M” el movimiento que finalice la partida volviendo “continuar” en un parámetro falso sin necesidad de verificar los puntos.

```
for i in range(fil):
    for j in range(col):
        print(tablero[i*col+j], end = " ")
    print()
mov = input("Ingrese un movimiento: ")
if mov == "m":
    continuar = False
if mov != "a" and mov != "s" and mov != "w" and mov != "d" and mov != "m":
    print("solo puede ingresar las tecas A, S, D, W o M")
```

15. Luego se verifica si la variable “mov” es igual a “W” quiere decir que la variable “pos” disminuirá una fila y se verificara si en dado caso la posición actual del icono “pacman” se encuentra en los límites superiores dicho icono aparecerá en los límites inferiores de la misma columna, luego se verifica si la celda donde se encuentra la nueva posición no está vacía se aplicara la siguiente comparación de cual icono es el que se encuentra y se suman o restan los puntos respectivamente.

```
if mov == "w":
    if pos - fil < 0:
        tablero[pos]="[  ]"
        pos = (fil * col-1)-((col-1)-pos)
    else:
        pos = pos - col
        tablero[pos+col]="[  ]"
    if tablero[pos]==p10:
        pts = pts+10
    elif tablero[pos]==p15:
        pts = pts+15
    elif tablero[pos]==q10:
        pts = pts-10
    tablero[pos]="[  ]"
    tablero[pos]=pacman
```

16. Luego se verifica si la variable “mov” es igual a “A” quiere decir que la variable “pos” se le restara una unidad y se verificara si en dado caso la posición actual del icono “pacman” se encuentra en los límites izquierdos dicho icono aparecerá en los límites derechos de la misma fila, luego se verifica si la celda donde se encuentra la nueva posición no está vacía se



aplicara la siguiente comparación de cual icono es el que se encuentra y se suman o restan los puntos respectivamente.

```
if mov == "a":
    if pos % col != 0:
        pos = pos - 1
        tablero[pos+1]="[  ]"
    elif pos % col == 0:
        pos = pos + (col-1)
        tablero[pos-(col-1)] = "[  ]"
    if tablero[pos]==p10:
        pts = pts+10
    elif tablero[pos]==p15:
        pts = pts+15
    elif tablero[pos]==q10:
        pts = pts-10
    tablero[pos]=pacman
```

17. Luego se verifica si la variable “mov” es igual a “S” quiere decir que la variable “pos” aumentara una fila y se verificara si en dado caso la posición actual del icono “pacman” se encuentra en los límites inferiores dicho icono aparecerá en los límites superiores de la misma columna, luego se verifica si la celda donde se encuentra la nueva posición no está vacía se aplicara la siguiente comparación de cual icono es el que se encuentra y se suman o restan los puntos respectivamente.

```
if mov == "s":
    if pos + fil > (fil * col -1):
        tablero[pos]="[  ]"
        pos = (col-1)-((fil * col-1)-pos)
    else:
        pos = pos + col
        tablero[pos-col]="[  ]"
    if tablero[pos]==p10:
        pts = pts+10
    elif tablero[pos]==p15:
        pts = pts+15
    elif tablero[pos]==q10:
        pts = pts-10
    tablero[pos]=pacman
```

18. Luego se verifica si la variable “mov” es igual a “D” quiere decir que la variable “pos” se le aumenta una unidad y se verificara si en dado caso la

posición actual del icono “pacman” se encuentra en los límites derechos dicho icono aparecerá en los límites izquierdos de la misma fila, luego se verifica si la celda donde se encuentra la nueva posición no está vacía se aplicara la siguiente comparación de cual icono es el que se encuentra y se suman o restan los puntos respectivamente.

```
if mov == "d":
    if (pos+1) % col != 0:
        pos = pos + 1
        tablero[pos-1]="[ ]"
    elif (pos+1) % col == 0:
        pos = pos - (col-1)
        tablero[pos+(col-1)] = "[ ]"
    if tablero[pos]==p10:
        pts = pts+10
    elif tablero[pos]==p15:
        pts = pts+15
    elif tablero[pos]==q10:
        pts = pts-10
    tablero[pos]=pacman
```

19. Luego de validarse todos los movimientos y finalizado la partida se procede a ingresar los datos obtenidos de esta partida en la variable “hist” la cual cada vez se ingresa a una nueva partida aumentara una fila.

```
for i in range(n):
    for j in range(3):
        if(hist[i*3+j]=="[ ]"):
            hist[i*3+0]=nom
            hist[i*3+1]=c
            hist[i*3+2]=pts
```

20. Si el jugador decide ingresar a la opción 2 se procede a imprimir el historial de partidas anteriores siendo esta la impresión de la variable “hist”.

```
if opc == 2:
    print("Historial de juego")
    for i in range(n):
        for j in range(3):
            print(hist[i*3+j], end = " ")
        print()
```

21. Si el jugador decide ingresar la opción 3 la consola finalizara su función.