

PRACTICA 3 IPC1 D

MANUAL TÉCNICO

DATOS:

01. Victor Eduardo José Rodríguez Alonzo

02. 201900018

```
if ( $this->rule_exists( $resource_details['id'], $role_details['id'] ) {  
    // Remove the rule as there is currently no need for it  
    $details['access'] = ! $access;  
    $this->_sql->delete( 'acl_rules', $details );  
} else {  
    // Update the rule with the new access value  
    $this->_sql->update( 'acl_rules', array( 'access' => $access )  
    foreach( $this->rules as $key => $rule ) {  
        if ( $details['role_id'] == $rule['role_id'] && $details['role_id'] == $rule['role_id'] ) {  
            if ( $access == false ) {  
                $this->_sql->delete( 'acl_rules', $details );  
            }  
        }  
    }  
}
```

PACMAN

INTRODUCCIÓN

Este manual será utilizado para ayudar a los futuros programadores que quieran modificar o utilizar este código a poder entenderlo y mejorarlo de una forma fácil y sencilla de explicar la funcionalidad de cada línea de código, métodos, funciones etc, mediante una aplicación de consola, se utilizará el lenguaje de programación Python para una funcionalidad más fácil.

OBJETIVOS

Mostrar una forma agradable de especificar el código al programador, para que pueda comprender y entender el juego pacman que se realizo en lenguaje de programación Python.

OBJETIVOS ESPECIFICOS

- Que el programador sea capaz de modificar y entender el código de nuestro pacman.
- Que el programador visualice internamente los métodos, funciones que se utilizaron al crear el programa
- Que el programador se familiarice con el lenguaje de programación Python.

ESPECIFICACIÓN TÉCNICA

➤ Requisitos de Hardware.

- Computadora de escritorio o portátil.
- Mínimo 4GB de Memoria RAM.
- 250GB de Disco Duro o Superior.
- Procesador Inter Core I3 o Superior
- Resolución gráfica mínimo 1024 x 728 pixeles.

➤ Requisitos de Software.

- Sistema Operativo Windows 10 o Superior.
- Tener instalado Python última versión.
- Tener instalado Visual Studio Code.
- Lenguaje de Programación PYTHON.
- Extensiones Visual Code:
 - Pylance.
 - Python.

PACMAN

CLASES, CONSTRUCTORES, ENCAPSULAMIENTO.

Para poder empezar el código del juego nos tenemos que percatar que es lo que debemos de realizar al momento necesitamos una clase comida, una clase jugador para poder usar sus atributos y parámetros.

```
1 class jugador():
2     def __init__(self):
3         self.nombre = ""
4         self.posx = -1
5         self.posy = -1
6         self.puntos = 0
7         self.movimientos = 0
8
9
10
11     def getPosX(self):
12         return self.posx
13
14     def getPosY(self):
15         return self.posy
16
17     def getMovimientos(self):
18         return self.movimientos
19
20     def getPuntos(self):
21         return self.puntos
22
23     def setPosY(self, _posy):
24         self.posy = _posy
25
26     def setPosX(self, _posx):
27         self.posx = _posx
28
29     def addMovimiento(self):
30         self.movimientos = self.movimientos + 1
31
32     def addPuntos(self):
33         self.puntos = self.puntos + 5
34
35     def __gt__(self, j):
36         return self.movimientos > j.movimientos
```

```
1 class Comida():
2
3     def __init__(self, posx, posy): #init sirve para el constructor
4         self.posx = posx #EL SELF ES IGUAL A THIS
5         self.posy = posy
6         self.isEat = False
7
8     def getPosX(self):
9         return self.posx
10
11     def getPosY(self):
12         return self.posy
13
14     def isComido(self):
15         return self.isEat
16
17     def setPosY(self, _posy):
18         self.posy = _posy
19
20     def setPosX(self, _posx):
21         self.posx = _posx
22
23     def setEat(self):
24         self.isEat = True
```

PACMAN

MENU.

Para poder crear un menú es muy sencillo se ingresa lo solicitado en un ciclo while para que pueda repetirse cuando sea necesario y así mismo dentro del mismo para seleccionar cada opción creamos un if para que podamos trabajar luego en cada if llamamos a los métodos necesarios para poder realizar cada acción.



```
1  #MÉTODO PARA EL MENU DE JUEGO.
2  def menu(jugadores):
3      while True:
4          print(" 『PACMAN ★ IPC1 ★ 2022』 ")
5          print("*****")
6          print("1.      Iniciar Juego")
7          print("2.      Tabla de Posiciones")
8          print("3.      Salir")
9          print("*****")
10         num = int(input("Escoge tu opción: "))
11         if (num == 1):
12             Inicio(jugadores)
13
14         elif(num == 2):
15             Posiciones()
16
17         elif(num == 3):
18             break
```

También tenemos que tomar en cuenta que vamos a crear comida y con estos varios métodos para poder saber que comida se va agregar, que pasa si ya no hay comida o si en el siguiente punto hay comida, acá abajo mostraremos los métodos utilizados para la clase comida.

PACMAN

```
1  #MÉTODO PARA CREAR LA COMIDA EN POSICIONES ALEATORIAS
2  def crearComida(comidita, comidas: int):
3      index = 0
4      while index < comidas: #iterar mientras index sea menor a comida
5          posxcgenerada = random.randint(0,9)
6          posycgenerada = random.randint(0,9)
7          Ocupado = False
8          for cbusqueda in comidita:
9              if cbusqueda.getPosX() == posxcgenerada and cbusqueda.getPosY() == posycgenerada:
10                 Ocupado = True
11
12             if not Ocupado:
13                 ccreada = Comida(posxcgenerada, posycgenerada)
14                 comidita.append(ccreada)
15                 index = index + 1
16
17  #MÉTODO PARA VALIDAD SI HAY COMIDA EN LA SIGUIENTE CASILLA
18  def sigComida(jugadores, comidita):
19      for comida in comidita:
20          if jugadores.getPosX() == comida.getPosX() and jugadores.getPosY() == comida.getPosY():
21              comida.setEat()
22              jugadores.addPuntos()
23              return True
24      return False
25
26  #MÉTODO PARA VALIDAD SI AUN HAY COMIDA
27  def HabraComidas(comidita):
28      for comida in comidita:
29          if not comida.isComido():
30              return False
31      return True
```

Especificando cada método lo que se hace es lo siguiente el crear comida se sirve para poder crear toda la comida en general lo que hace esto es que genera comida en x y y en aleatorio.

El método sigComida es para especificar que al momento de que haya una comida en la siguiente posición donde vaya a comer el jugador, se le sumen los puntos

El método habraComidas es solamente para especificar cual es el resto de comida y al momento de que solo haya pocas comidas pueda acabar el juego.

PACMAN

También para indicar necesitamos la lógica de movimientos que lo vamos a trabajar con los siguientes métodos.

```
1 #MÉTODO PARA LA LÓGICA DE LOS MOVIMIENTOS DEL JUEGO
2 def movimientos(jugadores, comidita):
3     while True:
4         movimiento = input("Movimiento: ")
5         if str(movimiento) == "a" or str(movimiento) == "d":
6             Izquierda(jugadores, comidita)
7         elif str(movimiento) == "d" or str(movimiento) == "g":
8             Derecha(jugadores, comidita)
9         elif str(movimiento) == "u" or str(movimiento) == "b":
10            Arriba(jugadores, comidita)
11        elif str(movimiento) == "s" or str(movimiento) == "g":
12            Abajo(jugadores, comidita)
13        elif str(movimiento) == "e":
14            return menu(jugadores)
15    print(" - JUGADORES: {} - PUNTOS: {} - MOVIMIENTOS: {}".format(jugadores.nombre, jugadores.getPuntos(), jugadores.getMovimientos()))
16    matriz = Tablero(comidita, jugadores)
17    GenerarTablero(matriz)
18    if jugadores.getPuntos() == 40 or HabraComidas(comidita):
19        aux = jugador()
20        aux.nombre = jugadores.nombre
21        aux.puntos = jugadores.puntos
22        aux.movimientos = jugadores.movimientos
23        listaJugadores.append(aux)
24        jugadores.nombre = ""
25        jugadores.puntos = 0
26        jugadores.movimientos = 0
27        break
28
29
30 #MÉTODO PARA LA POSICIÓN ALEATORIA DEL JUGADOR
31 def JugadorAl(jugadores, comidita):
32     while True:
33         x = random.randint(0,9)
34         y = random.randint(0,9)
35
36         Ocupado = False
37         for c in comidita:
38
39             try:
40                 if c.getPosX() == x and c.PosY() == y:
41                     Ocupado = True
42             except Exception as e:
43                 pass
44
45             if not Ocupado:
46                 jugadores.setPosX(x)
47                 jugadores.setPosY(y)
48                 break
49
50 #MÉTODO PARA MOVERSE A LA IZQUIERDA
51 def Izquierda(jugadores, comidita):
52     x = jugadores.getPosX()
53     y = jugadores.getPosY() - 1
54
55     if y >= 0:
56         jugadores.setPosX(int(x))
57         jugadores.setPosY(int(y))
58         m = sigComida(jugadores, comidita)
59         jugadores.addMovimiento()
60
61 #MÉTODO PARA MOVERSE A LA DERECHA
62 def Derecha(jugadores, comidita):
63     x = jugadores.getPosX()
64     y = jugadores.getPosY() + 1
65
66     if y <= 9:
67         jugadores.setPosX(int(x))
68         jugadores.setPosY(int(y))
69         m = sigComida(jugadores, comidita)
70         jugadores.addMovimiento()
71
72 #MÉTODO PARA MOVERSE PARA ARRIBA
73 def Arriba(jugadores, comidita):
74     x = jugadores.getPosX() - 1
75     y = jugadores.getPosY()
76
77     if x >= 0:
78         jugadores.setPosX(int(x))
79         jugadores.setPosY(int(y))
80         m = sigComida(jugadores, comidita)
81         jugadores.addMovimiento()
82
83 #MÉTODO PARA MOVERSE PARA ABAJO
84 def Abajo(jugadores, comidita):
85     x = jugadores.getPosX() + 1
86     y = jugadores.getPosY()
87
88     if x <= 9:
89         jugadores.setPosX(int(x))
90         jugadores.setPosY(int(y))
91         m = sigComida(jugadores, comidita)
92         jugadores.addMovimiento()
```


PACMAN

El método principal que se llama movimientos este método nos sirve para poder meter toda la lógica de este, por lo que podemos ver que tenemos métodos de mover arriba, derecha, abajo, izquierda, estos métodos solamente es para indicar el movimiento y al llegar a las paredes no pueda traspasar como que en cada movimiento pueda indicar la cantidad de movimiento el puntaje y así mismo que imprima el tablero.

También tenemos el método de jugador aleatorio este nos sirve para indicar que al momento de iniciar una partida, el jugador pueda salir en cualquier posición.

Ahora para finalizar es necesario hacer un método que se llama en el menú y donde se junten todos estos métodos en este caso será llamado como lo anterior.

```
1  #MÉTODO PARA INICIAR JUEGO
2  def Inicio(jugadores):
3      name = input("Ingrese su Nombre: ")
4      jugadores.nombre = name
5      comidassol = random.randint(1,int(10*10*0.4))
6      comidita = []
7      crearComida(comidita, comidassol)
8
9      JugadorAl(jugadores, comidita)
10
11     matriz = Tablero(comidita, jugadores)
12     GenerarTablero(matriz)
13
14     movimientos(jugadores, comidita)
```

PACMAN

TABLA DE POSICIONES.

Para la tabla de posiciones necesitamos hacer 2 cosas una ya la realizamos y se encuentra en la clase jugador que es un método de ordenamiento ya introducido por Python, luego se crea un método para realizar la acción de ordenamiento por movimientos.

```
1 #MÉTODO PARA LA TABLA DE POSICIONES
2 def Posiciones():
3     listaJugadores.sort()
4
5     print("TABLA DE POSICIONES: ")
6     if len(listaJugadores) == 1:
7         print("1. {0} - {1} MOVIMIENTOS {2} - PUNTOS".format(listaJugadores[0].nombre, str(listaJugadores[0].movimientos), str(listaJugadores[0].puntos)))
8     elif len(listaJugadores) == 2:
9         print("1. {0} - {1} MOVIMIENTOS {2} - PUNTOS".format(listaJugadores[0].nombre, str(listaJugadores[0].movimientos), str(listaJugadores[0].puntos)))
10        print("2. {0} - {1} MOVIMIENTOS {2} - PUNTOS".format(listaJugadores[1].nombre, str(listaJugadores[1].movimientos), str(listaJugadores[1].puntos)))
11    elif len(listaJugadores) == 3:
12        print("1. {0} - {1} MOVIMIENTOS {2} PUNTOS".format(listaJugadores[0].nombre, str(listaJugadores[0].movimientos), str(listaJugadores[0].puntos)))
13        print("2. {0} - {1} MOVIMIENTOS {2} PUNTOS".format(listaJugadores[1].nombre, str(listaJugadores[1].movimientos), str(listaJugadores[1].puntos)))
14        print("3. {0} - {1} MOVIMIENTOS {2} PUNTOS".format(listaJugadores[2].nombre, str(listaJugadores[2].movimientos), str(listaJugadores[2].puntos)))
15    print("*****")
```