

# PRACTICA 1

DARÍO MENESES HERNANDEZ

## PAC-MAN SIMPLIFICADO

[https://github.com/dmh1006/Practica1\\_DarioMeneses2C](https://github.com/dmh1006/Practica1_DarioMeneses2C)

## CONTENIDO

Pac-Man Simplificado.....	1
Introducción .....	2
Estructura del proyecto .....	3
Lógica del juego .....	3
Módulo utils.py .....	3
Clase JuegoPacman (clases.py).....	4
Modo visual (Notebook) .....	4
Modo consola (pacman.py) .....	4
Diagrama de clases .....	5
Conclusión.....	6

## INTRODUCCIÓN

En esta práctica se ha desarrollado una versión simplificada del juego Pac-Man con el objetivo de aplicar los conceptos vistos en la asignatura de Sistemas Inteligentes. El sistema implementado permite gestionar el estado del juego, el movimiento del personaje, la interacción con el tablero y la contabilización de la puntuación.

La implementación se ha realizado de forma modular, separando la lógica del juego de su representación visual. Para ello, se ha definido una clase principal que encapsula el estado del juego y un conjunto de funciones auxiliares que gestionan los movimientos, el tablero y la entrada del usuario.

El juego puede ejecutarse tanto en un entorno visual interactivo, mediante un notebook con representación HTML, como desde la consola, permitiendo dos formas distintas de interacción a partir de la misma base lógica.

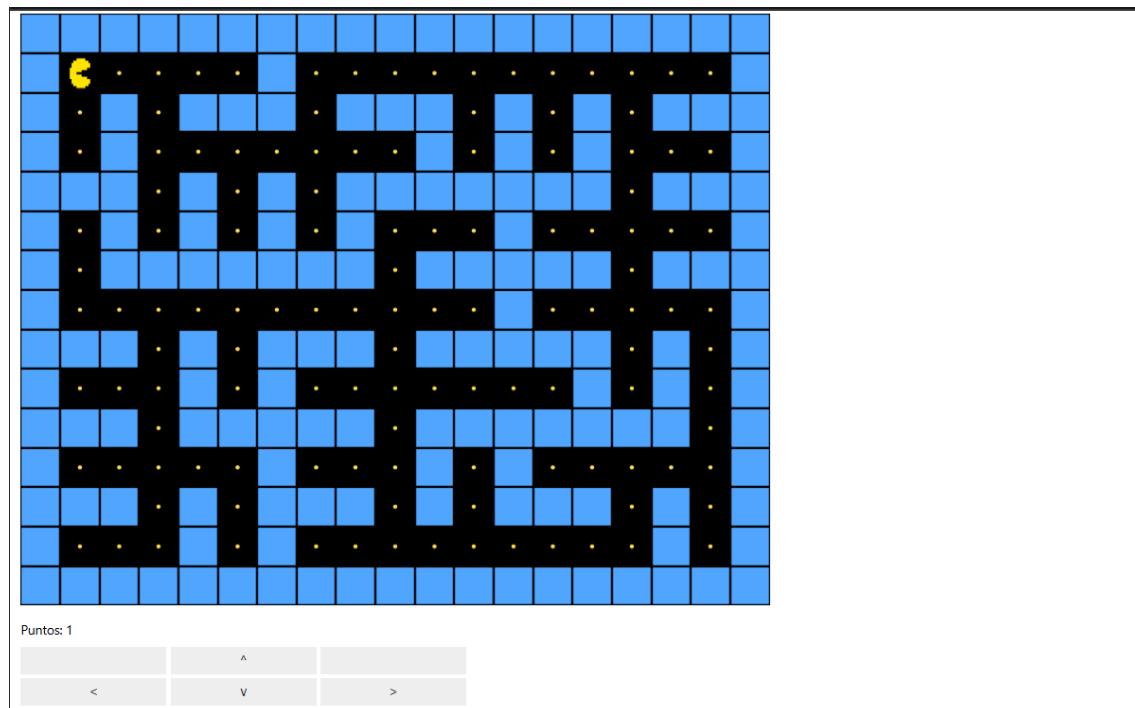


Figura 1: Ejecución del juego en modo visual

## ESTRUCTURA DEL PROYECTO

El proyecto se ha organizado siguiendo una estructura modular, separando claramente los distintos componentes del sistema. Esta organización facilita la comprensión del código, su mantenimiento y su reutilización, y permite identificar fácilmente la responsabilidad de cada archivo.

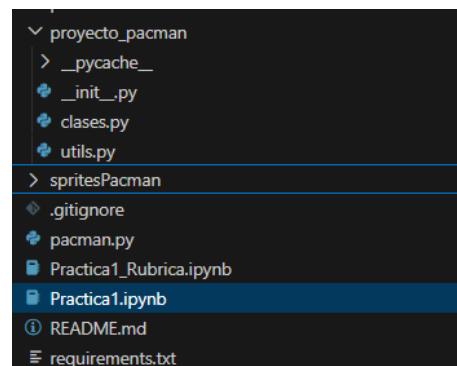


Figura 2: Estructura en VsCode

## LÓGICA DEL JUEGO

El tablero del juego se representa mediante una matriz de valores binarios, donde el valor 1 indica la presencia de una pared y el valor 0 indica una casilla transitable. A partir de esta representación se inicializa la posición del Pac-Man y la distribución de la comida en todas las casillas libres.

El movimiento del personaje está restringido a las posiciones transitables del tablero. Cada vez que el Pac-Man se desplaza a una nueva posición, se comprueba si existe comida en dicha casilla, incrementando la puntuación y eliminando la comida si procede. El juego finaliza cuando no queda comida disponible en el tablero.

## MÓDULO UTILS.PY

El archivo utils.py contiene el conjunto de funciones auxiliares necesarias para el funcionamiento del juego. En este módulo se definen los movimientos posibles del Pac-Man, los distintos tableros disponibles según la dificultad, así como funciones que permiten comprobar si una posición es válida, mover el personaje y gestionar la entrada del usuario.

Entre las funciones implementadas destacan:

- la selección del tablero mediante un menú por consola,
- la lectura de las pulsaciones del teclado,
- y las funciones de apoyo para comprobar colisiones y comida.

```

def mostrar_menu():
    """
    Muestra un menú sencillo por consola para seleccionar la dificultad.
    Devuelve la opción elegida como string: 'easy', 'medium' o 'hard'.
    """
    while True:
        print("== PAC-MAN ==")
        print("Selecciona dificultad:")
        print("1 - Easy")
        print("2 - Medium")
        print("3 - Hard")

        opcion = input("Opción: ")

        if opcion == "1":
            return "easy"
        elif opcion == "2":
            return "medium"
        elif opcion == "3":
            return "hard"
        else:
            print("Opción no válida. Intenta de nuevo.\n")

```

*Figura 3: Código implementado para mostrar el menú de selección de mapa*

## CLASE JUEGOPACMAN (CLASES.PY)

La clase JuegoPacman encapsula el estado completo del juego. En ella se almacenan el tablero, la posición actual del Pac-Man, la distribución de la comida y la puntuación obtenida.

Esta clase proporciona los métodos necesarios para actualizar el estado del juego tras cada movimiento, comprobar si queda comida disponible y obtener la puntuación actual. Además, implementa el método `__str__`, que permite representar el estado del juego en formato texto para su visualización en consola.

## MODO VISUAL (NOTEBOOK)

El modo visual del juego se implementa en el archivo Practica1.ipynb. En este modo, el estado del juego se representa mediante una tabla HTML que muestra una imagen diferente para cada tipo de elemento del tablero (pared, comida, espacio vacío y Pac-Man).

La interacción con el usuario se realiza a través de botones que simulan un teclado direccional. Cada pulsación actualiza el estado del juego y refresca la representación visual, permitiendo una interacción sencilla e intuitiva.

## MODO CONSOLA (PACMAN.PY)

El archivo pacman.py permite ejecutar el juego desde la consola. Al iniciar el programa, se muestra un menú que permite seleccionar la dificultad del tablero. El movimiento del Pac-Man se controla mediante las flechas del teclado, mientras que la tecla ESC permite finalizar la ejecución.

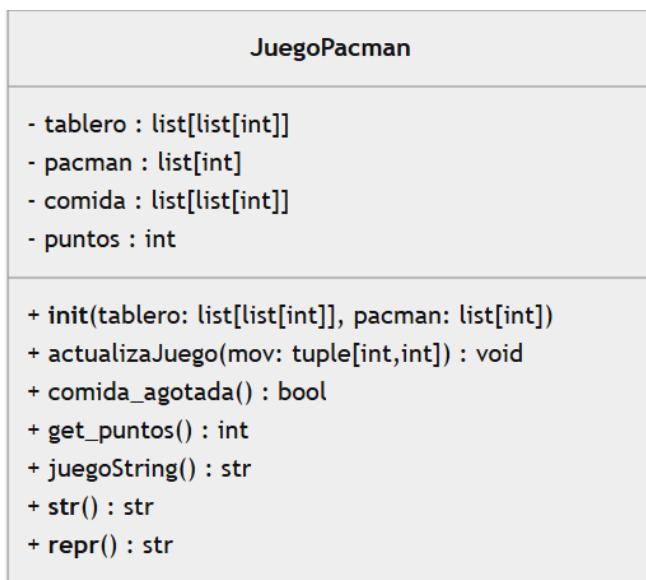
Tras cada movimiento, el estado del tablero y la puntuación se muestran por pantalla en formato texto, permitiendo seguir el desarrollo de la partida hasta que se consume toda la comida o se interrumpe el juego.

```
#####
#@...#
.#.#
.#.#
#####
#.#
#....#
#.###.#
#####
#####
```

Puntos: 1

Figura 4: Ejecución del juego en la consola

## DIAGRAMA DE CLASES



## CONCLUSIÓN

Mediante el desarrollo de esta práctica se han aplicado de forma práctica los conceptos fundamentales de la asignatura, reforzando la importancia de una correcta organización del código y de la separación entre lógica y presentación en el desarrollo de sistemas software.