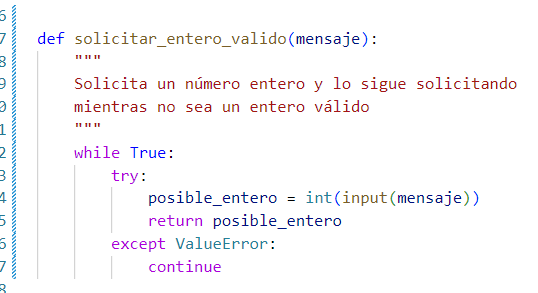
1. Métodos de entrada de datos



Confirma que se ingrese un entero valido.

Texto

Descripción generada automáticamente

Confirma que se ingrese un valor en el rango permitido por consola.

Texto

Descripción generada automáticamente

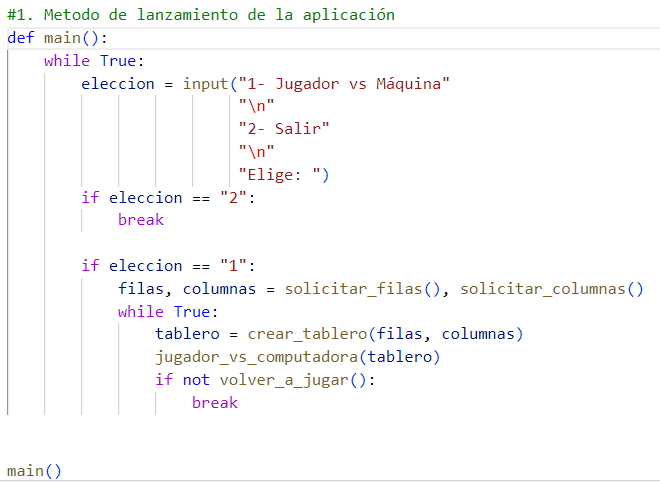
Confirma que se ingrese un valor en el rango permitido por consola.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

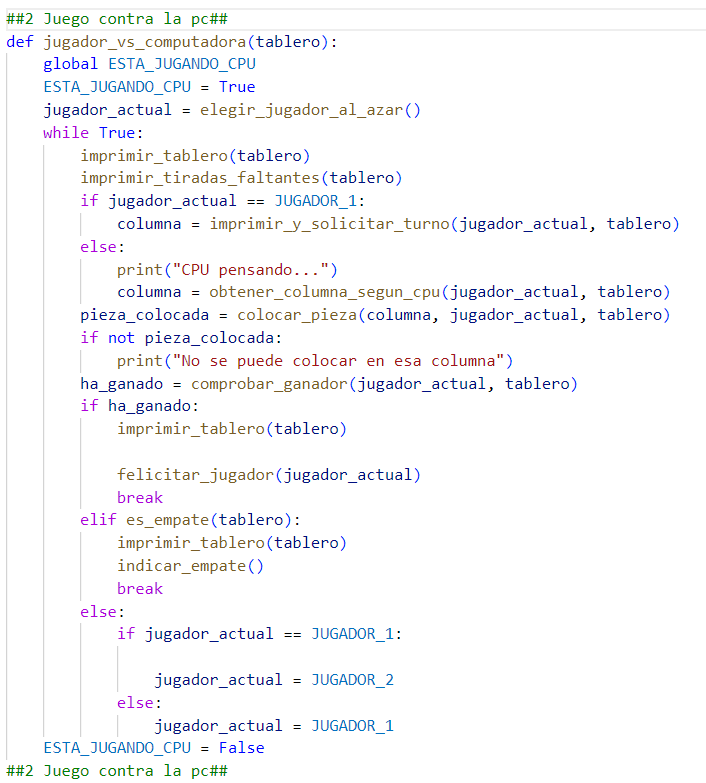
Genera la listas de sublistas.

1. Método de lanzamiento de la aplicación



El método inicia la aplicación de manera que por medio de un ciclo mantenga la ejecución de las iteraciones del turno de cada jugador. El juego maneja un menú en consola para moverse entro los métodos por medio de condiciones.

1. Juego contra la PC, iniciando la partida



Este método controla el inicio de los turnos con el método **elegir\_jugador\_al\_azar()** el cual utiliza la función Ramdom.

Elegir jugador al azar con la función ramdom

Texto

Descripción generada automáticamente

El método por medio del ciclo genera las iteraciones de los turnos y comprueba:

* Imprime el tablero con el método **imprimir\_tablero(tablero).**

Texto

Descripción generada automáticamente

Método para imprimir las listas de forma acorde a una matriz, además de agregar símbolos y definir el color de cada posición.

Texto

Descripción generada automáticamente

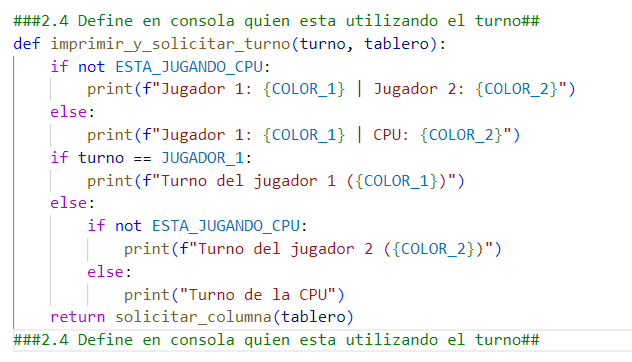
* Imprime los tiros aun posibles **imprimir\_tiradas\_faltantes(tablero).**

Texto

Descripción generada automáticamente

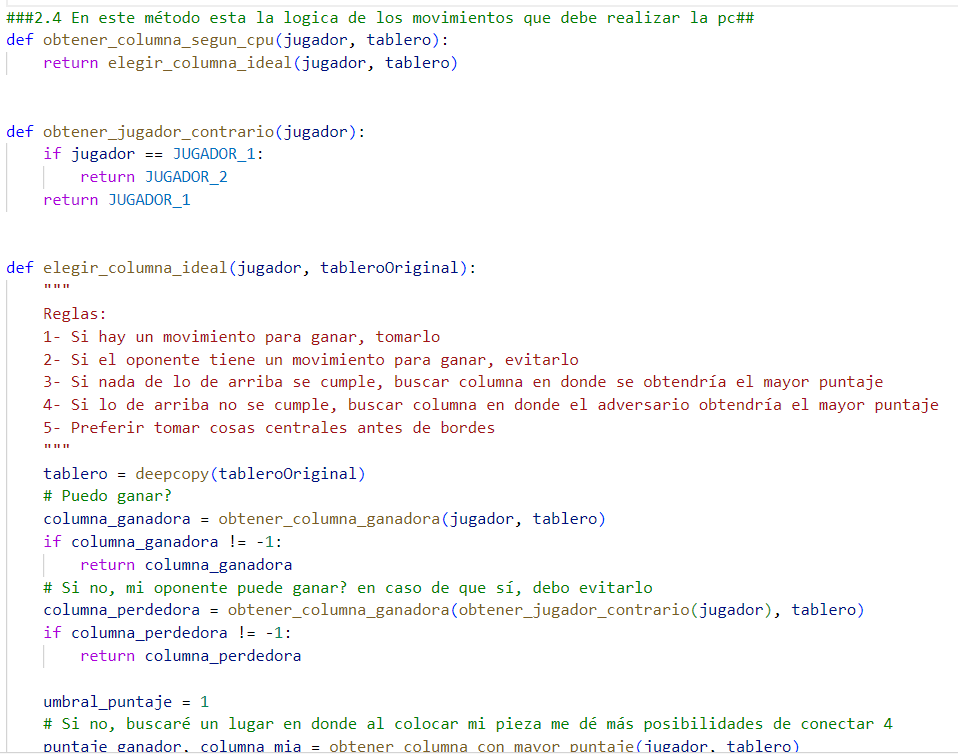
Básicamente vamos contando las posiciones con espacios en blanco.

* Define la columna seleccionada por el jugador **imprimir\_y\_solicitar\_turno(jugador\_actual, tablero).**



Adicionalmente el método de retorno obtiene la columna que seleccione el jugador.

* Analiza el espacio que debe llenar la CPu **obtener\_columna\_segun\_cpu().**



Reglas: la lógica del algoritmo se basa en las siguientes reglas.

1- Si hay un movimiento para ganar, tomarlo

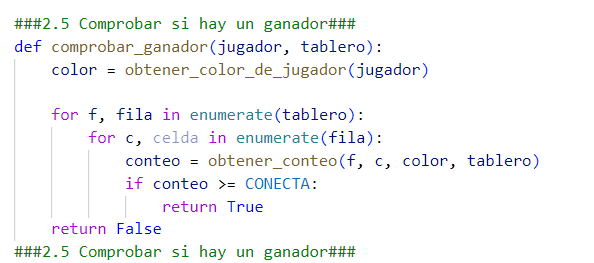
2- Si el oponente tiene un movimiento para ganar, evitarlo

3- Si nada de lo de arriba se cumple, buscar columna en donde se obtendría el mayor puntaje

4- Si lo de arriba no se cumple, buscar columna en donde el adversario obtendría el mayor puntaje

5- Preferir tomar cosas centrales antes de bordes

* Comprueba si en los movimientos realizados hay un ganador **comprobar\_ganador(jugador\_actual, tablero**).

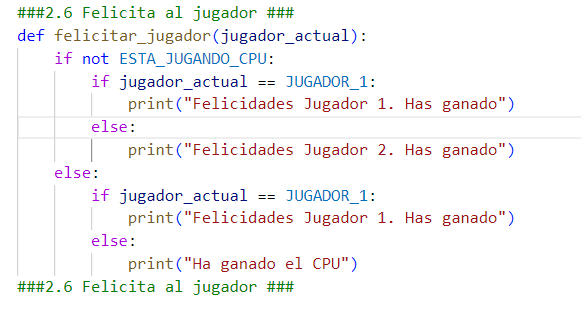


Por medio de un FOR revisamos fila a fila las fichas y comprobamos las opciones de ganar. Al definir los movimientos a revisar



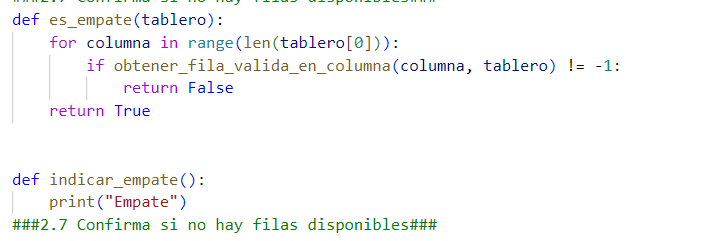
En el método **obtener\_conteo() a**se define una función globals para llamar cada uno de los movimientos a revisar por medio de un ciclo en espera de un TRUE.

* Felicita un jugador si ha ganado **felicitar\_jugador().**



Imprime un mensaje de felicitaciones al jugador

* Encuentra si hay un empate **indicar\_empate().**



Confirma si ya no hay posiciones jugables para definir empate