



# INSTITUTO POLITÉCNICO NACIONAL

## ESCUELA SUPERIOR DE CÓMPUTO

CARRERA:

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Materia:

Paradigmas de programación

Profesor: García Floriano Andrés

Alumno:

Escobar Rodriguez Alfonso

3CV1

Fecha de entrega: 22 de mayo del 2024.

## Código

Encontrar el camino:

```
def find_path(self):
    path = []
    if self._find_path(0, 0, path):
        return path
    return None
```

- `find\_path` intenta encontrar un camino desde la posición (0, 0) hasta la esquina inferior derecha.

- Inicializa una lista `path` para almacenar el camino.
- Llama al método privado `\_find\_path` para buscar el camino.
- Si se encuentra un camino, lo retorna; si no, retorna `None`.

Buscar el camino recursivamente:

```
def _find_path(self, row, col, path):
    if row == self.rows - 1 and col == self.cols - 1:
        path.append((row, col))
        return True
    if row >= self.rows or col >= self.cols or self.grid[row][col] == 1:
        return False
    if self._find_path(row + 1, col, path) or self._find_path(row, col + 1, path):
        path.append((row, col))
        return True
    return False
```

- `\_find\_path` es un método recursivo que trata de encontrar un camino desde `(row, col)` hasta la esquina inferior derecha.

- Si la posición actual `(row, col)` es la meta, añade la posición al camino y retorna `True`.
- Si la posición actual está fuera de los límites de la cuadrícula o es un obstáculo (`self.grid[row][col] == 1`), retorna `False`.
- Intenta moverse hacia abajo `(row + 1, col)` o hacia la derecha `(row, col + 1)`.
- Si alguno de estos movimientos encuentra un camino válido, añade la posición actual al camino y retorna `True`.
- Si ambos movimientos fallan, retorna `False`.

## Prueba de Validez

```
Ruta encontrada:
0, 0
0, 1
1, 1
2, 1
3, 1
3, 2
3, 3
PS C:\Users\spide\Documents\IPN\web>
```