

Algoritmos y Estructuras de Datos

Práctica 6. El problema del laberinto

Modificación a realizar

- ☐ **Definir e implementar un método de la clase `maze_t` que encuentre una nueva entrada al laberinto.**
- ☐ Es necesario que se pruebe con un ejemplo su funcionamiento y se incluya también el código de prueba en la entrega de la modificación.

Invocación en main

❑ Invocar en **main_maze_t.ccp**

```
...  
if (M.gate())  
{  
    cout << " Encontre otra entrada... " << endl;  
    cout << M << endl;  
}  
else  
{  
    cout << "No la encuentre... " << endl;  
    cout << M << endl;  
}
```

Invocación y cabeceras en C++

❑ Cabecera en `maze_t.hpp`

```
...  
bool gate(void);  
...
```

❑ Definición en `maze_t.cpp`

```
public:  
bool gate(void);
```

```
...  
bool  
    maze_t::gate()  
    {  
        ...  
    }
```

Chuleta...

```
bool maze_t::gate()
```

```
{
```

- Desplazarse por los bordes de la matrix
- Determinar si la nueva entrada es una puerta válida.
 - solver_(i,j)

```
}
```