```
#include <iostream>
#include <vector>
using namespace std;
class Laberinto {
private:
  vector<vector<int>> laberinto;
  vector<vector<bool>> visitado;
  int filas, columnas;

  const int dx[4] = {-1, 0, 1, 0};
  const int dy[4] = {0, 1, 0, -1};
public:
  Laberinto(vector<vector<int>> lab) : laberinto(lab) {
    filas = laberinto.size();
    columnas = laberinto[0].size();
    visitado = vector<vector<bool>>(filas, vector<bool>(columnas, false));
  }
  bool esValido(int x, int y) {
    return (x >= 0 && x < filas && y >= 0 && y < columnas &&
        laberinto[x][y] != 0 && !visitado[x][y]);
  }
  bool resolver(int x, int y) {
    if (laberinto[x][y] == 2) {
      cout << "La salida esta en: (" << x << ", " << y << ")" << endl;
      return true;
    }
    visitado[x][y] = true;
    cout << "Estamos en: (" << x << ", " << y << ")" << endl;
    for (int i = 0; i < 4; i++) {
      int nuevoX = x + dx[i];
      int nuevoY = y + dy[i];
```

```
            if (esValido(nuevoX, nuevoY)) {
                if (resolver(nuevoX, nuevoY)) {
                    cout << "Camino: (" << x << ", " << y << ") -> (" << nuevoX << ", " << nuevoY << ")" << endl;
                    return true;
                }
            }
        }
        visitado[x][y] = false;
        return false;
    }
};
int main() {
    vector<vector<int>> lab = {
        {1, 0, 1, 1, 1},
        {1, 1, 1, 0, 1},
        {0, 0, 1, 0, 1},
        {0, 0, 1, 1, 1},
        {0, 0, 0, 0, 2}
    };
    Laberinto laberinto(lab);
    cout << "Posición inicial: (0, 0)" << endl;
    if (laberinto.resolver(0, 0)) {
        cout << "Salida encontrada" << endl;
    } else {
        cout << "No hay salida." << endl;
    }
    return 0;
}
```