

Estructura de Datos

Amy Cárdenas Silva

Grupo: 1360

Carrera: ingeniería en
computación

laberinto, muestra de
funcionamiento

```

public class Main {
    public static void main(String[] args) {
        Laberinto lab = new Laberinto(7,7);
        lab.setInicio(5,0);
        lab.setFinal(5,6);
        lab.setCamino(1,1);
        lab.setCamino(1,4);
        lab.setCamino(2,1);
        lab.setCamino(2,2);
        lab.setCamino(2,3);
        lab.setCamino(2,4);
        lab.setCamino(3,3);
        lab.setCamino(4,1);
        lab.setCamino(4,3);
        lab.setCamino(4,5);
        lab.setCamino(5,1);
        lab.setCamino(5,2);
        lab.setCamino(5,3);
        lab.setCamino(5,4);
        lab.setCamino(5,5);
        lab.resolver();
    }
}

public class Laberinto {
    private int ren;
    private int col;
    private Array2d mapa;
    private Pila exploradorRen = new Pila<>();
    private Pila exploradorCol = new Pila<>();
    private int finalRen;
    private int finalCol;

    public Laberinto (int ren, int col){
        this.ren = ren;
        this.col = col;
        this.mapa = new Array2d(ren,col);
        this.mapa.clear('■');
    }

    public char getEspacio(int ren, int col){
        return this.mapa.getItem(ren,col);
    }
}

```

```

public void setPared(int ren, int col){
    this.mapa.setItem(ren,col,'■');
}

public void setCamino(int ren, int col){
    this.mapa.setItem(ren,col,'□');
}

public void setInicio(int ren, int col){
    this.mapa.setItem(ren,col,'I');
}

public void setFinal(int ren, int col){
    this.mapa.setItem(ren,col,'F');
    this.finalRen = ren;
    this.finalCol = col;
}

public void imprimirTablero(){
    System.out.println("┌─┐ × ──┐ ♡÷ . ∴ . ");
    for (int i = 0; i < ren; i++) {
        for (int j = 0; j < col; j++) {
            System.out.print(" " + this.mapa.getItem(i,j) + " ");
        }
        System.out.println("");
    }
}

public void setExplorador(){
    for (int i = 0; i < ren; i++) {
        for (int j = 0; j < col; j++) {
            if (this.mapa.getItem(i,j)=='I'){
                this.mapa.setItem(i,j,'o');
                this.exploradorRen.push(i);
                this.exploradorCol.push(j);
            }
        }
    }
}

public void paso(){
    int expRen = (int)exploradorRen.peek();
    int expCol = (int)exploradorCol.peek();

```

```

        if( expCol > 0 && (this.mapa.getItem(expRen,expCol - 1) == '□' ||
this.mapa.getItem(expRen,expCol - 1) == 'F')){
            this.mapa.setItem(expRen,expCol - 1, 'o');
            this.exploradorRen.push(expRen);
            this.exploradorCol.push(expCol - 1);
        }else{
            if(expRen > 0 && (this.mapa.getItem(expRen - 1,expCol) == '□' ||
this.mapa.getItem(expRen - 1,expCol) == 'F')){
                this.mapa.setItem(expRen - 1 ,expCol, 'o');
                this.exploradorRen.push(expRen - 1);
                this.exploradorCol.push(expCol);
            }else{
                if(expCol < this.col && (this.mapa.getItem(expRen,expCol + 1) == '□' ||
this.mapa.getItem(expRen,expCol + 1) == 'F')){
                    this.mapa.setItem(expRen,expCol + 1, 'o');
                    this.exploradorRen.push(expRen);
                    this.exploradorCol.push(expCol + 1);
                }else{
                    if(expRen < this.ren && (this.mapa.getItem(expRen + 1,expCol ) == '□'
|| this.mapa.getItem(expRen + 1,expCol ) == 'F')) {
                        this.mapa.setItem(expRen + 1, expCol, 'o');
                        this.exploradorRen.push(expRen + 1);
                        this.exploradorCol.push(expCol);
                    }else{
                        this.mapa.setItem(expRen,expCol,'x');
                        this.exploradorRen.pop();
                        this.exploradorCol.pop();
                    }
                }
            }
        }
    }
}

```

```

}

```

```

public int getFinalRen(){
    return this.finalRen;
}

```

```

public int getFinalCol(){
    return this.finalCol;
}

```

```

public void resolver(){
    imprimirTablero();
    setExplorador();
    imprimirTablero();
    while (this.mapa.getItem(this.finalRen,this.finalCol) == 'F'){
        paso();
        imprimirTablero();
    }
}

```

```

@Override
public String toString() {
    return "Laberinto{" +
        "ren=" + ren +
        ", col=" + col +
        ", mapa=" + mapa +
        '}';
}

```

```

import java.util.Arrays;

```

```

public class Array2d {
    private Character [][]data;
    private int rowSize;
    private int colSize;

    public Array2d(int ren, int col) {
        this.rowSize = ren;
        this.colSize = col;
        this.data = new Character[ren][col];
    }

    public void clear(Character dato){
        for (int i = 0; i < this.rowSize; i++) {
            for (int j = 0; j < this.colSize; j++) {
                this.data[i][j] = dato;
            }
        }
    }

    public int getRowSize() {
        return rowSize;
    }
}

```

```

public int getColSize() {
    return colSize;
}

```

```

@Override
public String toString() {
    String str = "";

    for (int i = 0; i < this.rowSize; i++) {
        for (int j = 0; j < this.colSize; j++) {
            str = str + this.data[i][j] + ", ";
        }
        str = str + "\n";
    }
    return str;
}

```

```

public void setItem(int ren, int col, Character dato){
    if (ren>=0 && ren < this.rowSize && col >= 0 && col < this.colSize){
        this.data[ren][col]=dato;
    }else{
        System.out.println("Indices fuera de rango");
    }
}

```

```

public Character getItem(int ren, int col){
    if (ren>=0 && ren < this.rowSize && col >= 0 && col < this.colSize){
        return this.data[ren][col];
    }else{
        System.out.println("Indices fuera de rango");
    }
    return '\0';
}

```

```

}

```

```

import java.util.Stack;

```

```

public class Pila <E>{
    public Stack<E> data;
}

```

```
public Pila(){
    this.data = new Stack<>();
}

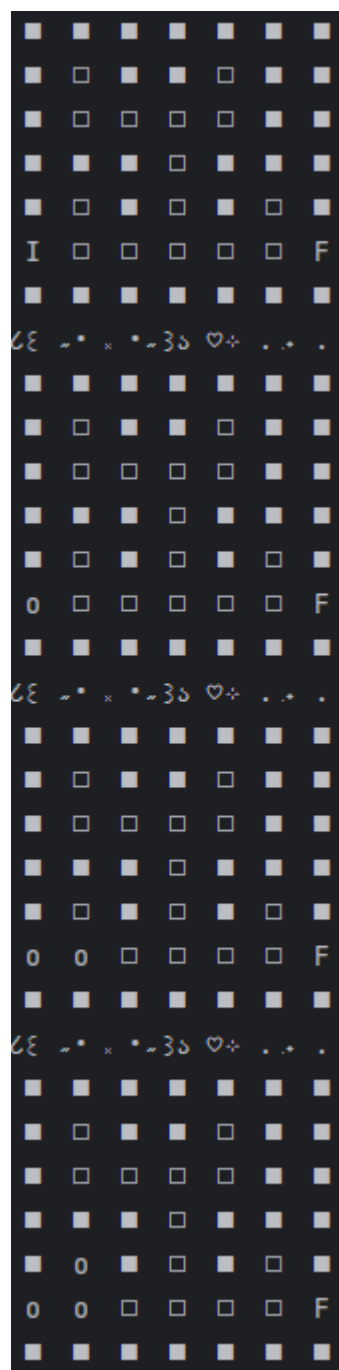
public boolean estaVacia(){
    return this.data.empty();
}

public E peek(){
    return this.data.peek();
}

public E pop(){
    return this.data.pop();
}

public void push(E objeto){
    this.data.push(objeto);
}

public int buscar(E object){
    return this.data.search(object);
}
}
```



6E ~• x • ~3b ♡÷ . . .

■	■	■	■	■	■	■
■	□	■	■	□	■	■
■	□	□	□	□	■	■
■	■	■	□	■	■	■
■	X	■	□	■	□	■
0	0	□	□	□	□	F
■	■	■	■	■	■	■

6E ~• x • ~3b ♡÷ . . .

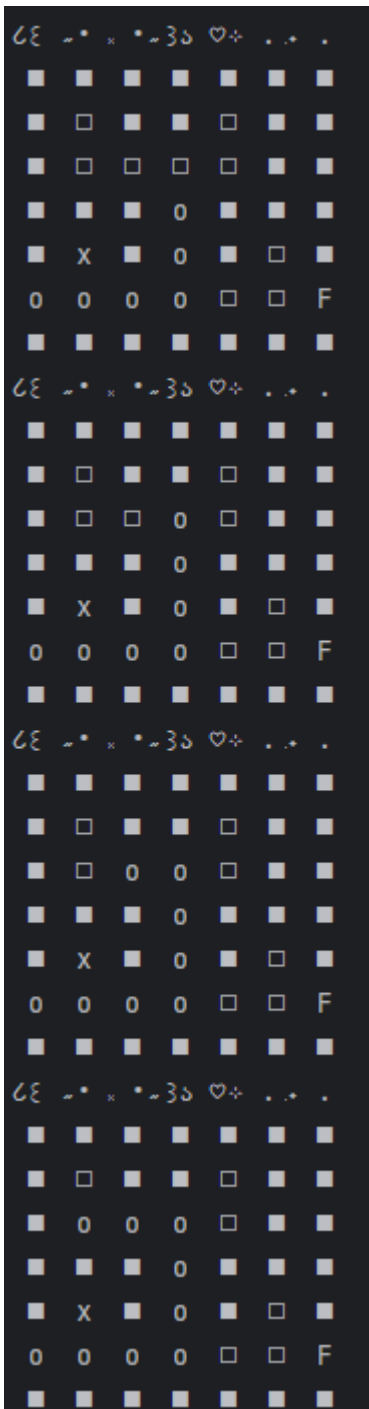
■	■	■	■	■	■	■
■	□	■	■	□	■	■
■	□	□	□	□	■	■
■	■	■	□	■	■	■
■	X	■	□	■	□	■
0	0	0	□	□	□	F
■	■	■	■	■	■	■

6E ~• x • ~3b ♡÷ . . .

■	■	■	■	■	■	■
■	□	■	■	□	■	■
■	□	□	□	□	■	■
■	■	■	□	■	■	■
■	X	■	□	■	□	■
0	0	0	0	□	□	F
■	■	■	■	■	■	■

6E ~• x • ~3b ♡÷ . . .

■	■	■	■	■	■	■
■	□	■	■	□	■	■
■	□	□	□	□	■	■
■	■	■	□	■	■	■
■	X	■	0	■	□	■
0	0	0	0	□	□	F
■	■	■	■	■	■	■



6E ~• x •~3b ♡÷ . . .

■	■	■	■	■	■	■
■	0	■	■	□	■	■
■	0	0	0	□	■	■
■	■	■	0	■	■	■
■	X	■	0	■	□	■
0	0	0	0	□	□	F
■	■	■	■	■	■	■

6E ~• x •~3b ♡÷ . . .

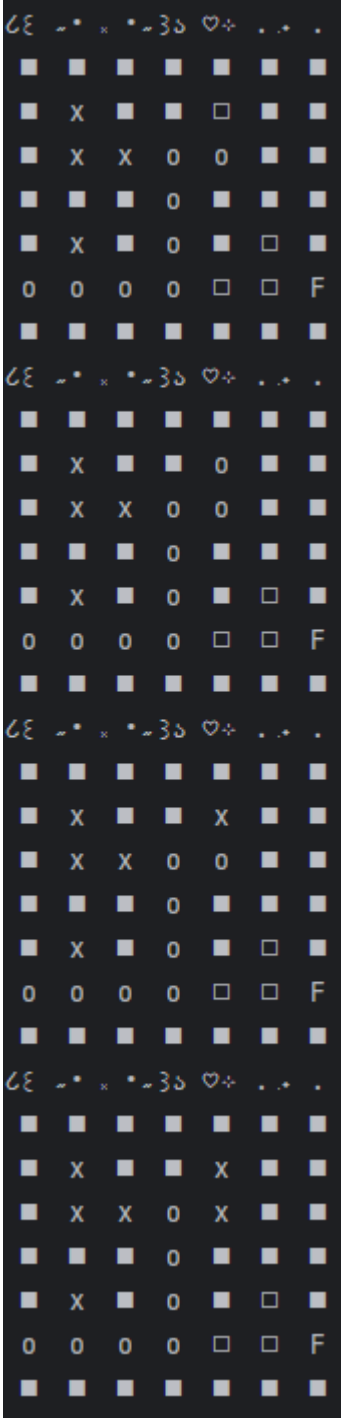
■	■	■	■	■	■	■
■	X	■	■	□	■	■
■	0	0	0	□	■	■
■	■	■	0	■	■	■
■	X	■	0	■	□	■
0	0	0	0	□	□	F
■	■	■	■	■	■	■

6E ~• x •~3b ♡÷ . . .

■	■	■	■	■	■	■
■	X	■	■	□	■	■
■	X	0	0	□	■	■
■	■	■	0	■	■	■
■	X	■	0	■	□	■
0	0	0	0	□	□	F
■	■	■	■	■	■	■

6E ~• x •~3b ♡÷ . . .

■	■	■	■	■	■	■
■	X	■	■	□	■	■
■	X	X	0	□	■	■
■	■	■	0	■	■	■
■	X	■	0	■	□	■
0	0	0	0	□	□	F
■	■	■	■	■	■	■



6E ~* x *-36 ♡÷ .+ .

■	■	■	■	■	■	■
■	X	■	■	X	■	■
■	X	X	X	X	■	■
■	■	■	0	■	■	■
■	X	■	0	■	□	■
0	0	0	0	□	□	F
■	■	■	■	■	■	■

6E ~* x *-36 ♡÷ .+ .

■	■	■	■	■	■	■
■	X	■	■	X	■	■
■	X	X	X	X	■	■
■	■	■	X	■	■	■
■	X	■	0	■	□	■
0	0	0	0	□	□	F
■	■	■	■	■	■	■

6E ~* x *-36 ♡÷ .+ .

■	■	■	■	■	■	■
■	X	■	■	X	■	■
■	X	X	X	X	■	■
■	■	■	X	■	■	■
■	X	■	X	■	□	■
0	0	0	0	□	□	F
■	■	■	■	■	■	■

6E ~* x *-36 ♡÷ .+ .

■	■	■	■	■	■	■
■	X	■	■	X	■	■
■	X	X	X	X	■	■
■	■	■	X	■	■	■
■	X	■	X	■	□	■
0	0	0	0	0	□	F
■	■	■	■	■	■	■

ℳ ⋈ ⋈ ⋈ ⋈ ⋈ ⋈ ⋈

■ ■ ■ ■ ■ ■ ■ ■

■ X ■ ■ X ■ ■ ■

■ X X X X ■ ■ ■

■ ■ ■ X ■ ■ ■ ■

■ X ■ X ■ □ ■ ■

0 0 0 0 0 0 F

■ ■ ■ ■ ■ ■ ■ ■

ℳ ⋈ ⋈ ⋈ ⋈ ⋈ ⋈ ⋈

■ ■ ■ ■ ■ ■ ■ ■

■ X ■ ■ X ■ ■ ■

■ X X X X ■ ■ ■

■ ■ ■ X ■ ■ ■ ■

■ X ■ X ■ 0 ■ ■

0 0 0 0 0 0 F

■ ■ ■ ■ ■ ■ ■ ■

ℳ ⋈ ⋈ ⋈ ⋈ ⋈ ⋈ ⋈

■ ■ ■ ■ ■ ■ ■ ■

■ X ■ ■ X ■ ■ ■

■ X X X X ■ ■ ■

■ ■ ■ X ■ ■ ■ ■

■ X ■ X ■ X ■ ■

0 0 0 0 0 0 F

■ ■ ■ ■ ■ ■ ■ ■

ℳ ⋈ ⋈ ⋈ ⋈ ⋈ ⋈ ⋈

■ ■ ■ ■ ■ ■ ■ ■

■ X ■ ■ X ■ ■ ■

■ X X X X ■ ■ ■

■ ■ ■ X ■ ■ ■ ■

■ X ■ X ■ X ■ ■

0 0 0 0 0 0 0

■ ■ ■ ■ ■ ■ ■ ■