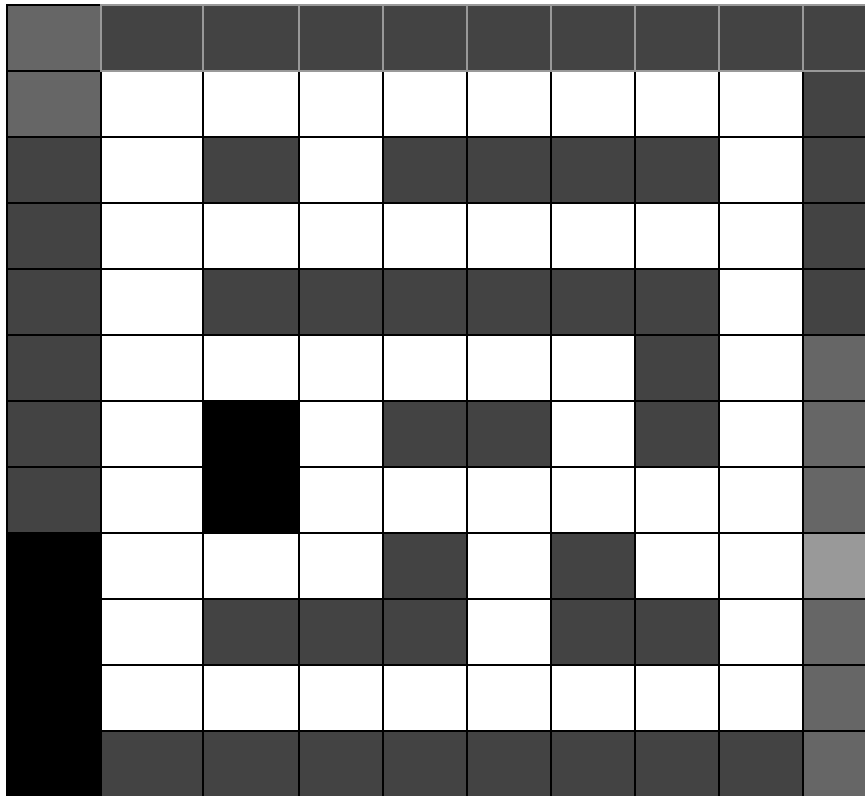


# Trabajo Práctico N° 0: Validador de Laberintos

## Enunciado

Se deberá implementar una herramienta que permita validar laberintos con un conjunto de reglas definidas. En esta primera instancia, se validarán las reglas de un laberinto del tipo "Pacman". Esta herramienta será utilizada para otros proyectos más adelante.

Ejemplo de un Laberinto:



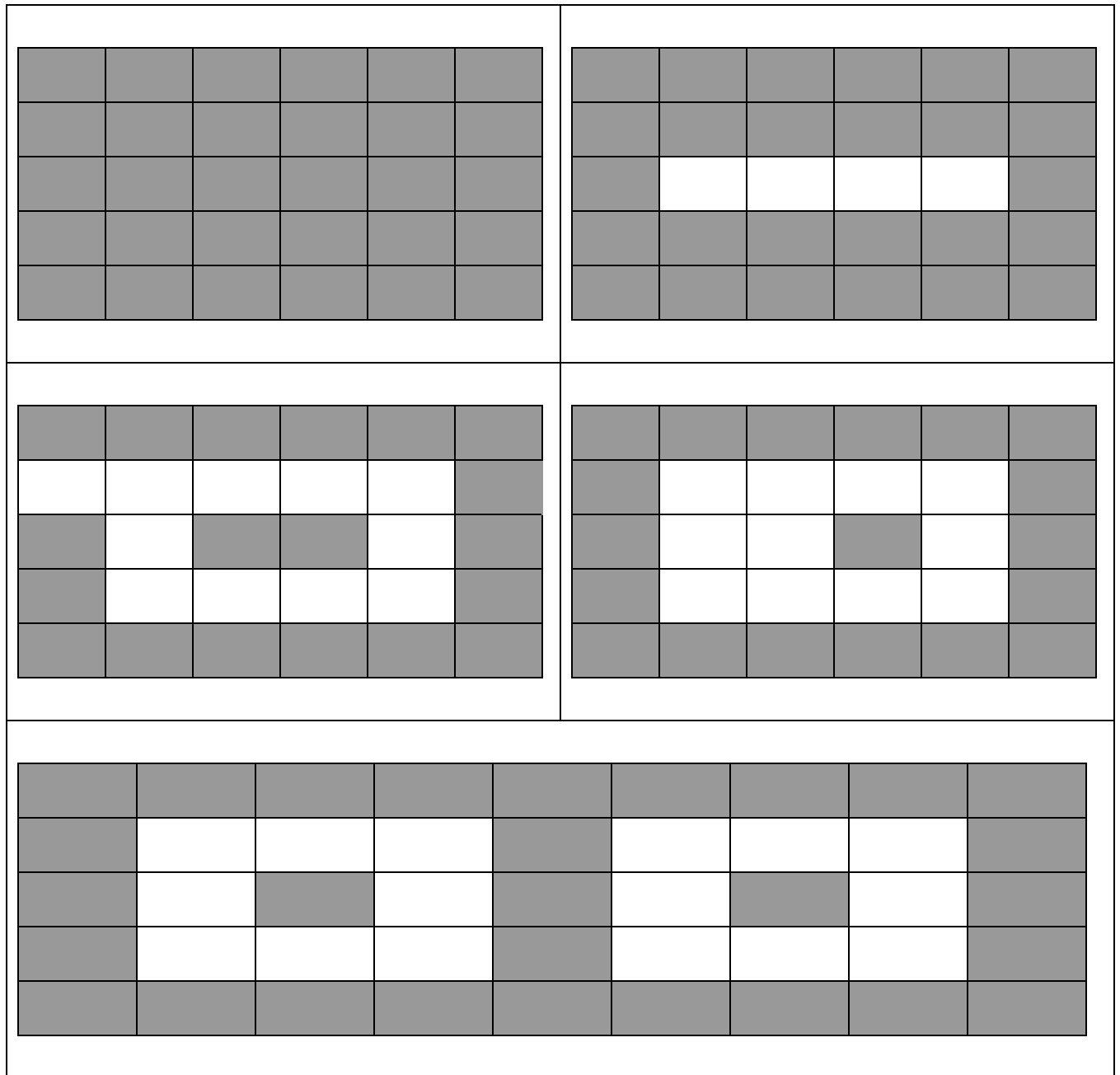
## Reglas de laberintos tipo Pacman:

- Dos celdas transitables adyacentes horizontal o verticalmente están conectadas.
- No hay forma de transitar por las paredes, o por el vacío que rodea el mapa.
- El laberinto debe tener al menos dos celdas transitables.
- Toda celda transitable tiene al menos dos conexiones a otras celdas.
- Existe al menos un camino entre cada par de celdas transitables.
- Partiendo de una celda dada, no puedo regresar a la misma celda en menos de 4 movimientos sin pasar dos veces por la misma celda.

	<pre>{   "cells": [     [0 0 0 0 0 0 0],     [0 1 1 1 1 1 0],     [0 1 2 1 2 1 0],     [0 1 1 1 1 1 0],     [0 0 0 0 0 0 0]   ],   "impassableTypes": [     0,     2   ] }</pre>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Ejemplos de Laberintos no válidos:

--	--



## Funcionalidad a Implementar

Implementar la API provista de forma que se pueda realizar este procesamiento.

## Objetivos

- Implementar una versión inicial de la herramienta que cumpla los requerimientos.
- Aplicar las técnicas vistas en la teoría y en la práctica.
- Ganar conocimiento sobre las herramientas vistas en la materia.

## Herramientas a utilizar

- Javascript
- Node.js
- Docker
- Git

## Restricciones

- **Queda prohibido ejecutar directamente valores provistos por el usuario.** Esto incluye el uso de eval y otros mecanismos similares.
- Se trabajará sobre el repositorio git de **GitLab**, el cual deberán crear e invitar al usuario **tecnicas7510** como miembro **Maintainer**
- Trabajo realizado individualmente.

## Criterios de Corrección

- Cumplimiento de las restricciones
- Diseño del modelo
- Diseño del código
- Documentación entregada (informe, diagramas y tests)
- Correctitud
- Completitud
- Buen uso de las herramientas indicadas
- Uso de buenas prácticas en general

## Entrega

- Para el Hito **Entrega TP, via GIT**: Crear un tag "entrega-0" (Campo "Tag Version" en GitLab, comando ``git tag entrega-0 ; git push origin entrega-0`` ) sobre el commit a entregar
- La entrega deberá incluir un conjunto de test unitarios y de aceptación que validen escenarios válidos e inválidos.

## Calendario

FECHA	HITO
Jueves 04/4	Presentación del TP
Jueves 11/4	Revisión estado del TP
Miercoles 17/4	Entrega TP, via GIT
Jueves 25/4	Devolución y comentarios de las entregas

## API del sistema

La API del sistema se basa en un único endpoint:

**POST** /v1.0/maze

```
{
  "cells": [
    [0 0 0 0 0 0],
    [0 1 1 1 1 0],
    [0 1 2 1 2 1 0],
    [0 1 1 1 1 0],
    [0 0 0 0 0 0]
  ],
  "impassableTypes": [
    0,
    2
  ]
}
```

retorna success o fail con un mensaje de que regla falló.

Posibles retornos:

- HTTP Code 200 Success
 

```
{
}
```
- HTTP Code 422 Unprocessable Entity
 

```
{
  "message" : "Invalid Rule: xxxxxx";
}
```