

El cielo o el infierno

En un castillo encantado hay una habitación que al ingresar se transforma en un laberinto que lleva al Cielo o al Infierno.

Lamentablemente no hay lógica de salida y quien lo recorre tiene que ir eligiendo si va para un lado o para el otro (solo en dos direcciones puede ir, digamos 0 o 1, derecha o izquierda).

El desafío es armar un programa que permita jugar al Cielo o el Infierno. Eso tiene dos partes:

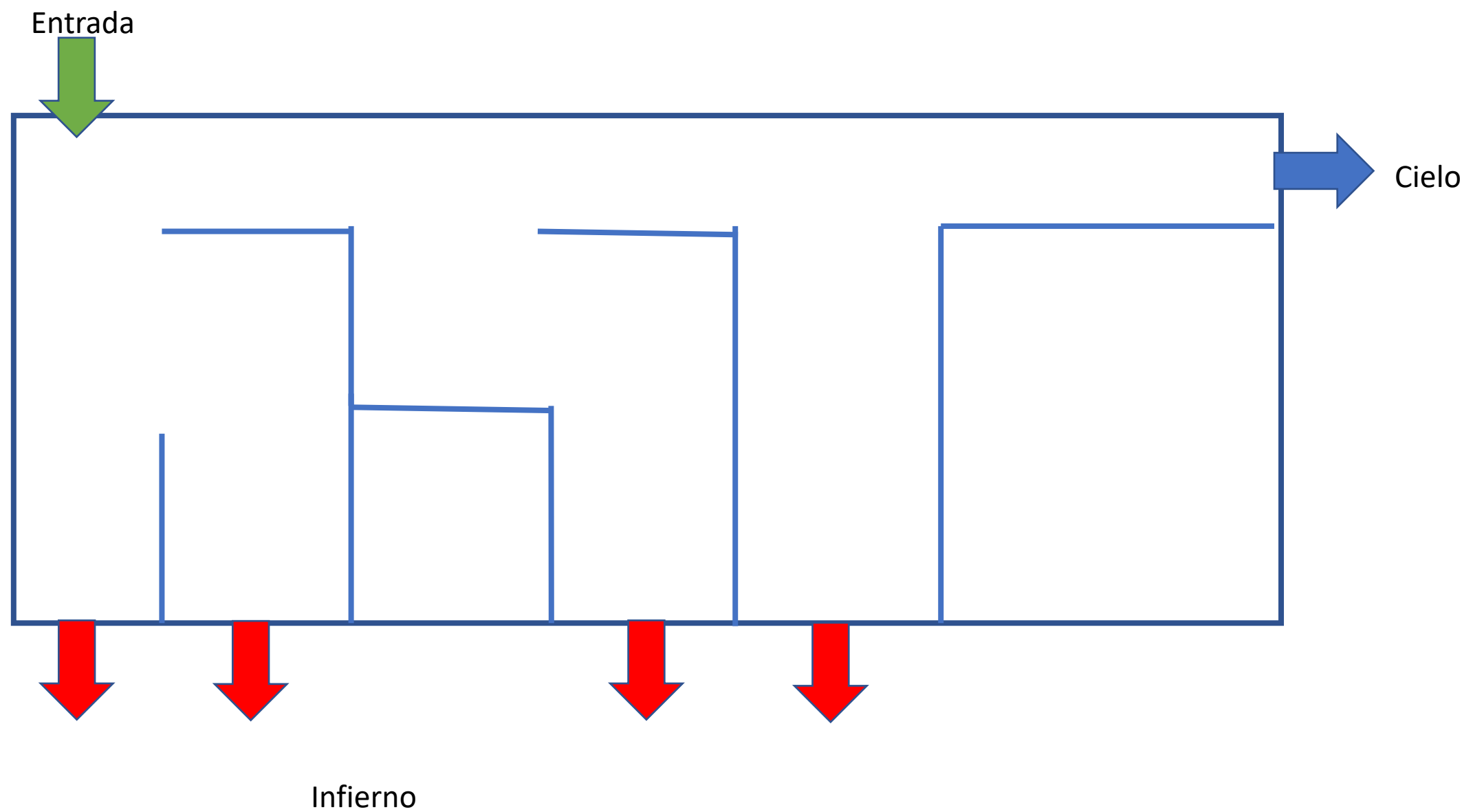
A) Desarrollar la necesario para armar los laberintos

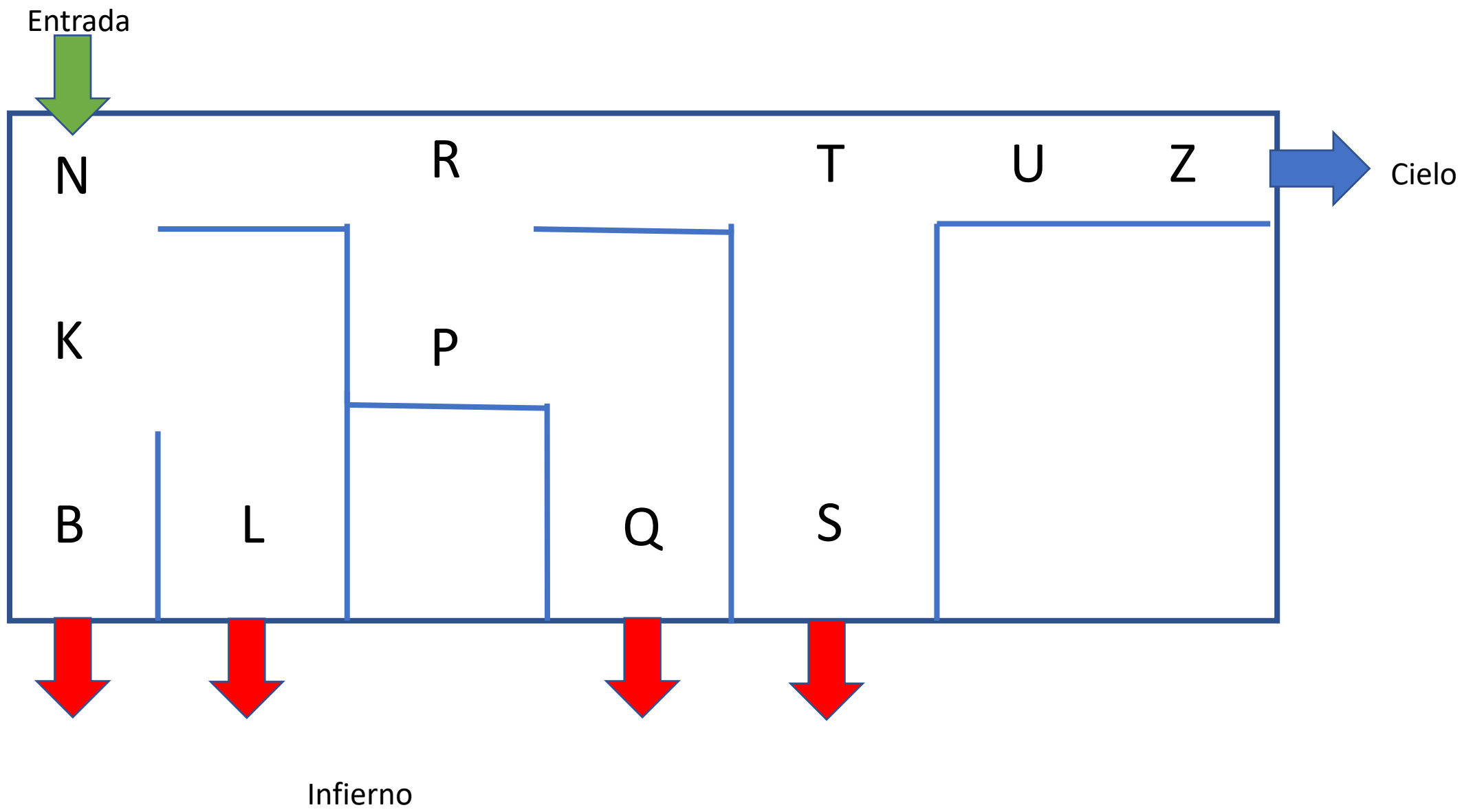
- RESTRICCIONES:
 - Cada laberinto tiene una única ENTRADA
 - Solamente se puede mover en dos direcciones (digamos, derecha o izquierda)
 - Puede haber varias salidas pero una sola es el CIELO

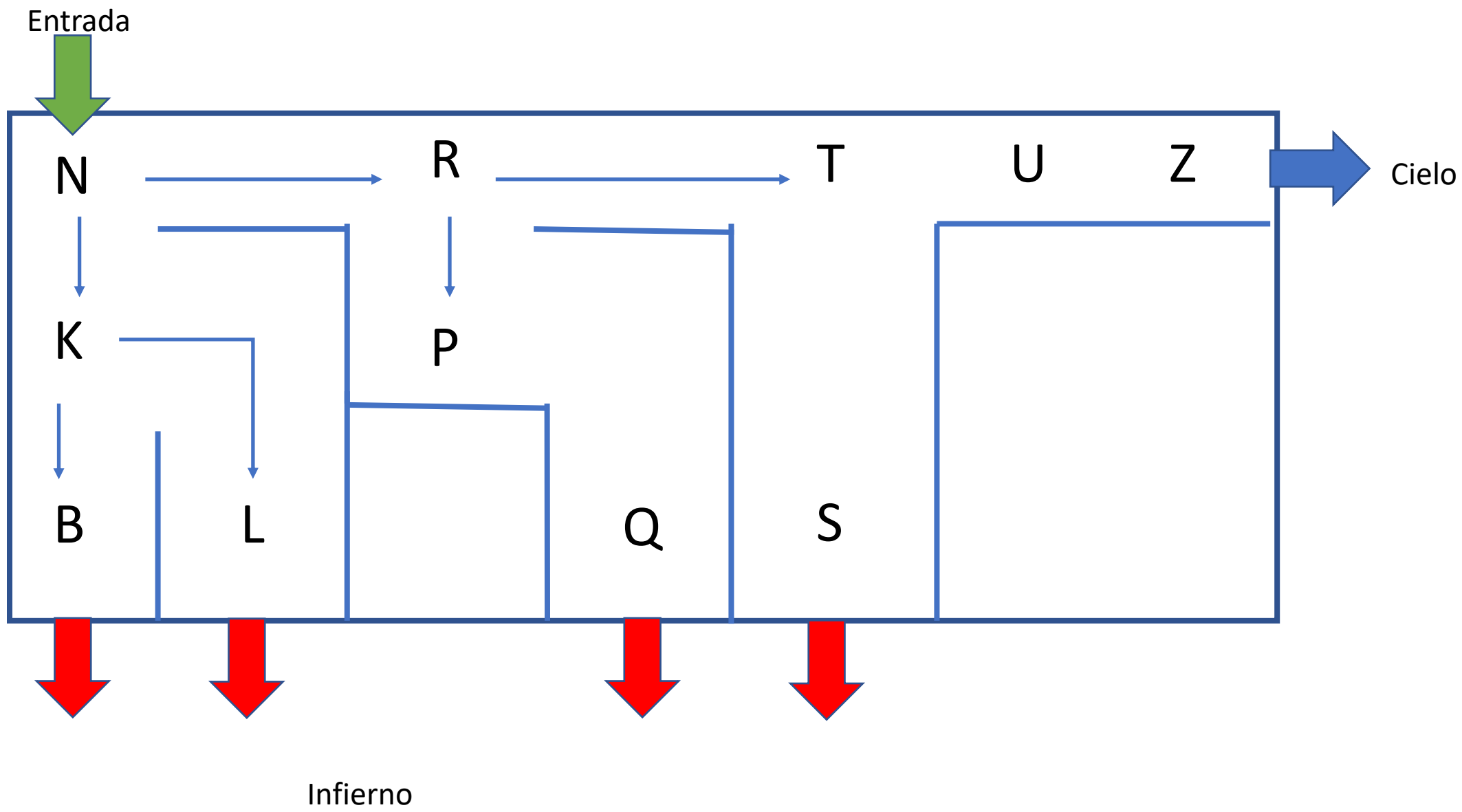
B) Desarrollo del

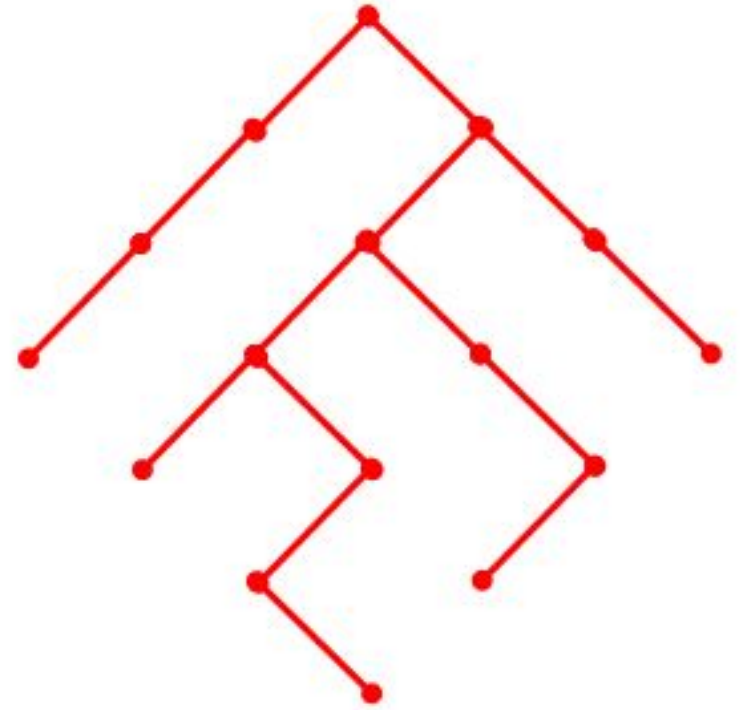
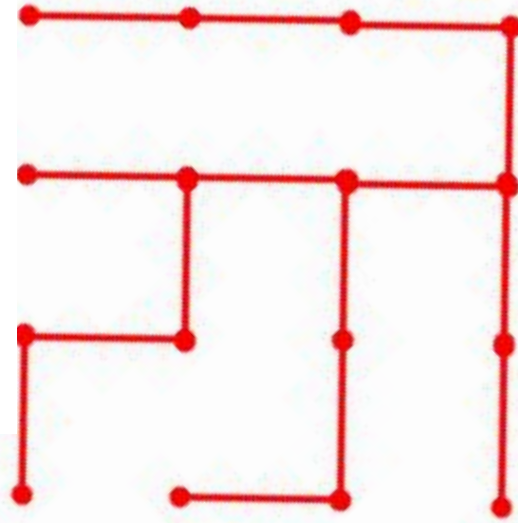
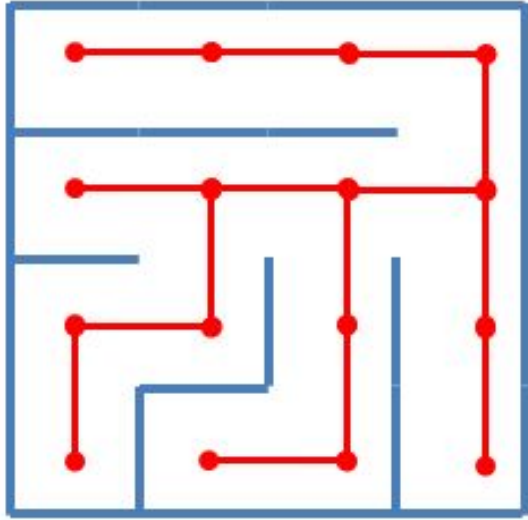
- REGLAS:
 - El jugador elige en qué dirección moverse (derecha o izquierda) y no puede volver atrás
 - A medida que avanza y decide puede caer en el Cielo o en el Infierno

A) El laberinto . Este es un ejemplo ¿Cómo podríamos armar diferentes tipos de laberintos?









PARTE A:

a.1) Crear una función `laberinto(listadepasos)` que recibe una lista de pasos del laberinto y arma un laberinto. Retorna el laberinto. Esta lista tiene el inicio y final indicados

- Esta función necesita generar caracteres al azar

```
import random
```

```
def randomCar():
```

```
    """ el seed (semilla) permite cambiar la generación de  
    números al azar """
```

```
    ##random.seed()
```

```
    return chr(random.randrange(97, 97 + 26))
```

PARTE A- ESTRATEGIA PARA EL ARMADO EL ÁRBOL DEL LABERINTO:

1) Armar una lista con pasos de una cantidad x , un INICIO y una SALIDA

- NO SE PUEDEN REPETIR !!!
- Insertar en la lista el INICIO
- Mientras no supere la cantidad de pasos deseados
 - A) generar un CAR al azar
 - B) controlar si NO está en la lista
 - Agregar a la lista
 - Incrementar la cantidad de pasos agregados
- Insertar en la lista la SALIDA

2) Armar el árbol recorriendo la lista por posición

- Crear el árbol
- Mientras la pos de la lista es menor que su tamaño inserto en el árbol la posición

PARTE B – Recorrer el árbol para preguntar cómo moverse en el laberinto

1) ESTRATEGIA: cosas a tener en cuenta

- Recorrer el árbol mientras tenga hijos
- Preguntar para dónde moverse
- Moverse al nodo siguiente (si se puede)
- IMPORTANTE
 - Si quiere moverse a una dirección que no hay hijos del nodo mostrar mensaje de alerta "solo puede moverse en la dirección contraria" -> sería como chocar contra una pared