



# Tecnológico de Monterrey

## **Modelación de sistemas multi agentes y gráficas computacionales**

**Actividad integradora: Parte 1. Sistemas  
multiagentes**

David Christopher Valderas Silva

Sergio Ruiz Loza

ITC

A01658889

## Estrategia:

Para la solución de este problema se necesita que 5 robots encuentren dentro de un espacio las cajas en el piso y los caminos para colocarlas en una localización predeterminada en pilas de máximo 5 cajas. Los robots deben de trabajar en conjunto para terminar de acomodar todas las cajas en el tiempo límite. En el espacio habrá algunos obstáculos como estantes y una puerta, por lo que los robots tendrán que encontrar en el espacio los caminos por los que puedan pasar.

Para esta solución se tiene principalmente al ambiente y como parte de él a los agentes (robots) y al suelo (ground).

Los robots contarán un id, una velocidad al azar, un tiempo límite para encontrar un nuevo camino y una variable booleana para saber si están sosteniendo o no una caja. Esta clase tiene las funciones para tomar caja y dejar caja, y dependen del atributo hasBox ya que si esta variable es True entra la función leaveBox para que el robot coloque en su lugar la caja que lleva, de manera similar si este atributo es False entra la función takeBox para tomar una de las cajas sueltas. La función findNewPath se encarga de encontrar el camino que el robot seguirá también dependiendo de hasBox, pues si este es True el destino del camino a encontrar es en donde se debe de colocar la caja y en base a esto esta función encuentra el camino más corto hacia ese destino, de igual manera si este atributo es False el destino cambia a ser donde se encuentran las cajas para que el agente siga el camino que lo lleve a tomar una nueva caja.

El suelo es el espacio sobre el que los robots caminarán y donde se encuentran las cajas, es decir los posibles caminos (paths) y arrays de cajas, de máximo 5 datos cada uno.

Boxes solo cuenta con la posición de la caja, la cual Ground recibe y comunica al agente por medio del ambiente para que los robots conozcan la ubicación de estas y definan su siguiente path con destino a ellas.

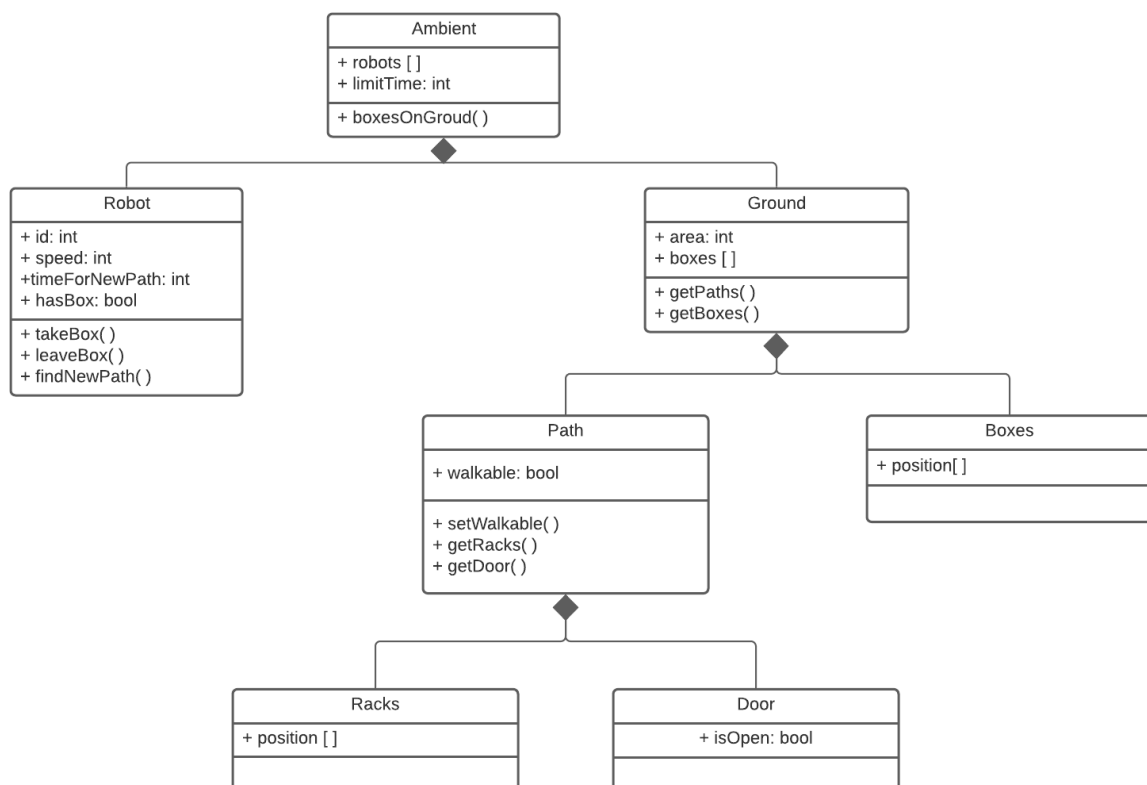
Paths son los diferentes caminos posibles en el ground, pero tomando en cuenta que cada path tiene obstáculos como racks y una puerta, se tiene que definir por qué caminos se puede caminar, para lo que existe una variable

booleana en Paths, así como también cuenta con funciones para obtener los racks, la puerta y definir la variable walkable con estos datos.

Racks cuentan con su posición, la cual Paths obtendrá para definir qué caminos no son caminables.

Door solo cuenta con una variable booleana que define si está abierta o cerrada, la cual Paths obtiene para definir el camino que pase por ella como caminable si esta está abierta, o no caminable si está cerrada.

### Diagrama de clases:



## Protocolo de agentes:

