



# Tecnológico de Monterrey

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 302)

**Profesor:**

Edgar Covantes Osuna  
Jose Daniel Azofeifa Ugalde

**Equipo:**

**Integrantes:**

Maria Fernanda De León Gomez	A01197340
Diana Laura Hernández Villarreal	A01570679
Roberto Abraham Pérez Iga	A01384237
Jorge Claudio González Becerril	A01412375
Andrea Corona Arroyo	A01366768
Gabriel Crisóstomo Navidad	A01367480

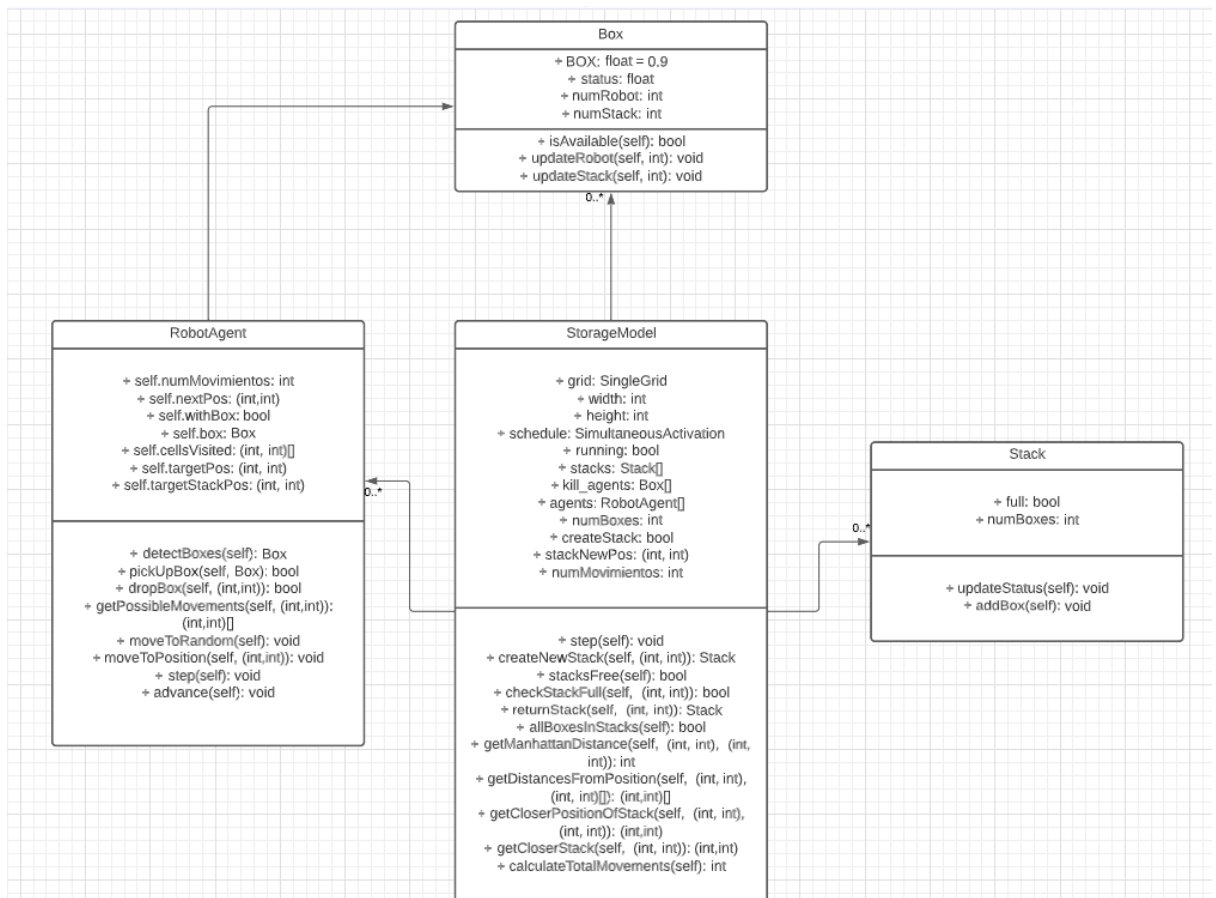
Fecha: Viernes 25 de noviembre del 2022

**Evidencia 1. Actividad Integradora**

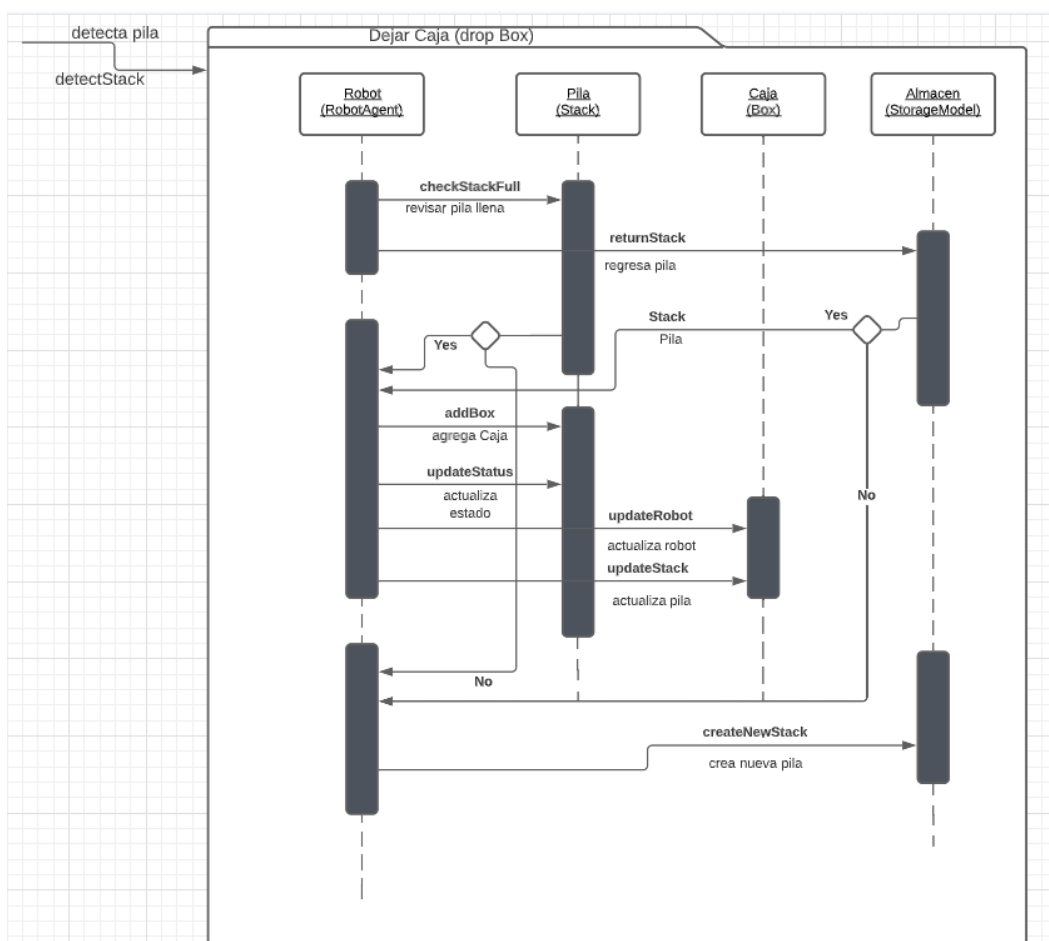
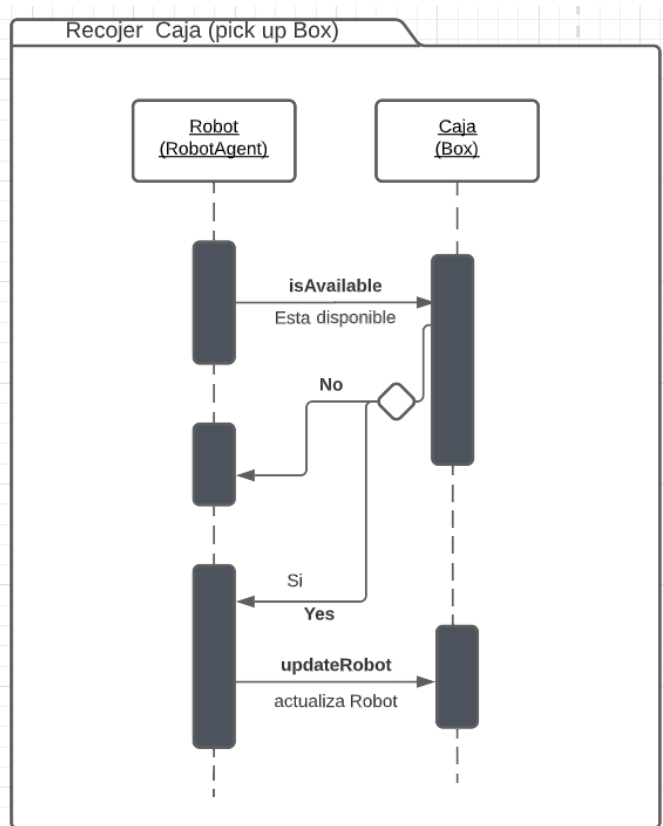
## Índice

<b>I. Diagrama de clases</b>	<b>3</b>
<b>II. Protocolo de agentes/interacción</b>	<b>4</b>
<b>III. Estrategia cooperativa/Plan de trabajo</b>	<b>5</b>
<b>IV. Aprendizaje adquirido</b>	<b>5</b>

## I. Diagrama de clases



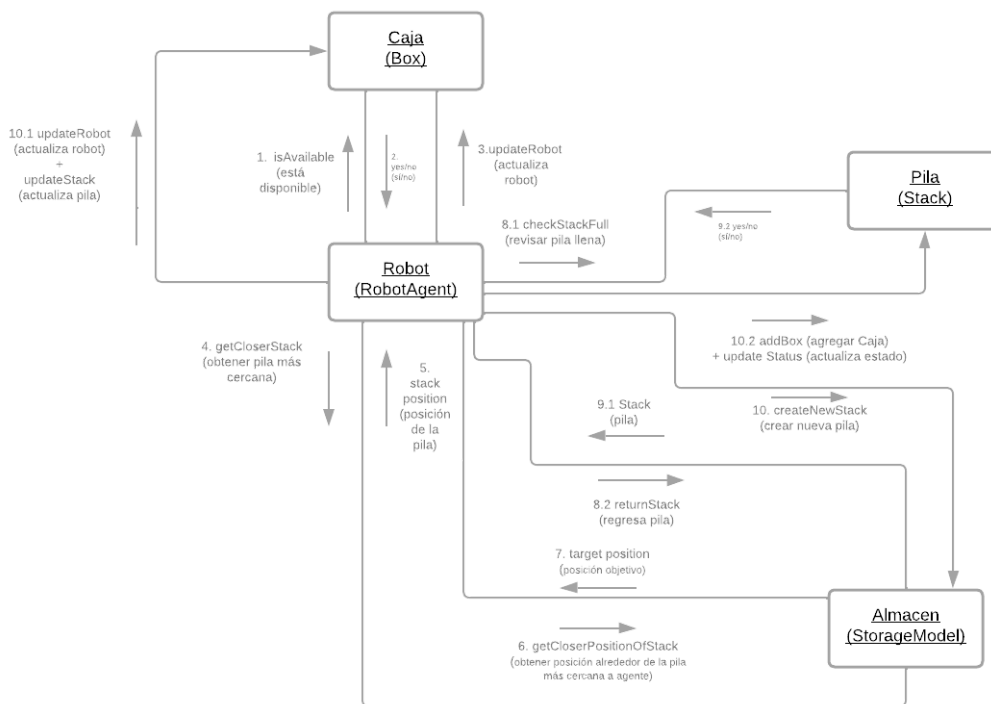
## II. Protocolo de agentes/interacción



### III. Estrategia cooperativa/Plan de trabajo

Nombre	Actividad	Tiempo Estimado	Tiempo Final
Diana Laura Hernández Villarreal	Comunicación entre agentes en Python Diseño de animación en Unity Modelado de prototipos en Unity Script C# de animación en Unity Corrección de drop zone en Python Protocolo de agentes	8 horas Martes 22	16 horas
Maria Fernanda De León Gomez	Elaboración de Github grupal Comunicación entre agentes en Python Diseño de animación en Unity Modelado de prototipos en Unity Script C# de animación en Unity Corrección de drop zone en Python Protocolo de agentes	8 horas Martes 22	16 horas
Roberto Abraham Pérez Iga	Aprendizaje multiagente en Python Implementación de colliders en Unity Protocolo de agentes Solución de errores en colores en script Python Elaboración del documento final Revisión final	10 horas Domingo 20	8 horas
Jorge Claudio González Becerril	Aprendizaje multiagente en Python Implementación de colliders en Unity Diagrama de clases Elaboración del documento final Solución de errores en animación de Unity Revisión final	14 horas Domingo 20	18 horas
Andrea Corona Arroyo	Aprendizaje multiagente en Python Integración del proyecto Diagrama de clases Elaboración del documento final Solución de errores en animación de Unity Revisión final	14 horas Domingo 20	18 horas
Gabriel Crisóstomo Navidad	Aprendizaje multiagente en Python Integración del proyecto Diagrama de clases Solución de errores en colores en script Python Solución de errores en animación de Unity Elaboración del documento final	12 horas Martes 22	14 horas

Diagrama colaborativo que muestra la interacción entre agentes



#### IV. Aprendizaje adquirido

##### **Diana Laura Hernández Villarreal:**

Para la elaboración de esta evidencia desarrollamos una simulación de una serie de robots que organizan cajas en un estante. Esto haciendo uso de python y mesa para la sección de agentes. Y unity para el desarrollo de la parte gráfica de la solución, es decir, una simulación. Para poder llegar a la solución del problema debimos plantear cuál sería el modelo y cuáles serían nuestros agentes ya que es necesario definir qué objetos trabajan simultáneamente y cuales serían todas las posibles situaciones en las que se podrían enfrentar al momento de realizar la simulación, por ejemplo, en caso de que se encuentre con una caja, que el almacén se encuentre lleno, etc. En fin, es necesario entender cómo funciona la situación para plantear todos los posibles escenarios ya que si nos olvidamos de alguno podría generar que la simulación dejará de funcionar y se presenten errores debido a que los agentes no sabrían cómo actuar en dicha situación, es así como definimos todas las funciones de nuestra solución en python. Mientras que en la parte de gráficas tuvimos que plantear cuáles serían los objetos que necesitábamos modelar, en este caso era un robot, almacén, estantes y cajas. Estas se modelaron utilizando el paquete de probuilder. Utilizando esta herramienta y utilizando texturas diseñamos los modelos de estos objetos para poder implementar una solución visual de este problema.

**Andrea Corona Arroyo:**

En esta evidencia hicimos uso de la librería Mesa en Python, para realizar la simulación de la parte Multiagente. Para ello, tuvimos que entender a profundidad el problema, para poder plantear de mejor manera qué elementos serían considerados como agentes y cuáles como modelo, así como definir sus acciones y la información que se transmitirán entre ellos así como sus interacciones entre sí. Esto nos permitió practicar más el modelo con agentes para poder establecer comportamientos de acuerdo a diferentes escenarios. Por otro lado, en la parte de Unity, nos sirvió para practicar el modelado en 3D con la librería probuilder, con lo cual se practicó el uso de materiales y texturas para lograr simular más realísticamente el entorno en el cual se ubican los robots. De igual manera, se practicó el uso de funciones utilizadas para manejar colisiones simples, de forma que los robots pudieran realizar acciones que permitieran que no se trabaran en su totalidad. Si bien en esta evidencia todavía no se realizó la unión de ambas partes, es importante conocer ambos programas de manera independiente para poder resolver cualquier problema que pudiera surgir al realizar la implementación del reto final de este bloque.

**Jorge Claudio González Becerril:**

Esta materia fue mi introducción a un sistema de multiagentes, como lo simulamos en Python con Mesa. Aprendí conceptos nuevos como lo es el agente, y cómo este interactúa de manera interesante y espontánea con otros. Con Mesa aprendí a definir agentes y el modelo en el que 'vivirán', dándoles metas, estados, protocolos, y, básicamente, indicando cómo se deben comportar con respecto a la información que obtienen del modelo.

En la parte de gráficas computacionales, me abrió a un mundo totalmente nuevo. Nos adentramos al funcionamiento de lo visual en las computadoras, y cómo estas forman las imágenes que vemos, en tiempo real, en nuestras pantallas. Aprendí los fundamentos de cómo se traduce de geometría e información a lo visual, y algoritmos que se utilizan para optimizar estos procesos de renderizado.

**Gabriel Crisostomo Navidad:**

Para esta primera evidencia del bloque, se hizo uso de distintas herramientas, que ejemplifican las 2 partes fundamentales de este bloque de formación. Siendo la parte del modelaje del almacén y los robots, la parte de modelos multiagentes, y la generación de una simulación muy básica en Unity, la parte de gráficas computacionales. Pasando primero con la la parte de multiagente, esta primera parte de la evidencia fue muy útil para poder planificar nuevas estrategias y formas de modelar distintas problemáticas, desde un acercamiento de programación de agentes. Para la segunda parte de la evidencia, fue muy práctico e interactivo hacer el modelaje de los distintos elementos de la simulación del almacén, también sirvió para poner en práctica distintos conceptos de matemáticas de gráficas computacionales, como la ubicación espacial en 3D. Siendo el siguiente paso en el

bloque, juntar ambas partes, pero esta evidencia fue de mucha utilidad para que podamos empezar unir la parte gráfica de la parte de lógica y control de agentes.

### **Fernanda De Leon:**

A lo largo del bloque de formación de multiagentes fui capaz de identificar los tipos de agentes inteligentes, en esta evidencia se trabajó con multi agentes reactivos simples ya que deciden que realizar debido a la percepción actual. Por ejemplo, como un robot avanza hasta encontrar una caja y llevarlo al drop zone. Posteriormente aprendí sobre la comunicación entre ellos mediante estados con informar o solicitar a los agentes vecinos. Mediante la distribución de información es que se resuelve de forma cooperativa el problema, en este caso la organización de cajas por los robots. Posteriormente aprendí sobre el aprendizaje en modelos primitivos y la toma de decisiones por ejemplo tomar una caja y calcular la distancia más corta hasta el área destinada o recalcular en caso de que el agente vecino fuese otra caja que impide su camino. Finalmente en el bloque de gráficas computacionales aprendí a modelar objetos e iluminación con ProBuilder en los robos, cajas y almacén y cómo establecer métodos de rotación, transformación y traslación con matrices y cómo implementarlo en C# para la animación de la problemática. Finalmente aprendí la implementación de object occlusion que permite hacer más eficiente la simulación con la aparición mínima de objetos visibles.

### **Roberto Abraham Pérez Iga:**

Durante el periodo de multiagentes se pudieron desarrollar habilidades y conocimientos en el uso de la librería Mesa de Python para su aplicación en los sistemas multiagentes, poniendo en práctica la lógica detrás de los agentes y cómo estos van interactuando uno con el otro con la información brindada en el código, actuando de tal forma que se puede interpretar como un tipo de inteligencia artificial. Durante esta evidencia se puso a prueba el conocimiento de Mesa para aplicarlo en una simulación renderizada en Unity en 3D, dando como resultado un entorno (almacén) donde robots van apilando cajas una encima de otra hasta que todas estén ordenadas - por lo que se deben de tomar muchos aspectos a consideración, como todos los posibles escenarios en los que se puedan encontrar nuestros agentes. El reto era que se aplicará un código que pudiera adaptarse de manera efectiva para evitar así cualquier tipo de colisiones con los mismos robots o con los objetos a su alrededor, solo falta que se conecte todo para dar un resultado completo y funcional. Pero se obtuvo un excelente resultado inicial para la evidencia 1 del reto.