



Tecnológico de Monterrey

Campus Estado de México

Proceso de instalación

Materia:

**Modelación de sistemas multiagentes con gráficas computacionales
(Grupo 301)**

Alumnos:

Giovanna Lorena Delgado Mendoza A01656039

Lauren Lissette Llauradó Reyes A01754196

José Alonso Segura De Lucio A01747872

Bernardo Alejandro Limón Montes de Oca A01736575

Carrera:

ITC

Profesores:

**Jorge Adolfo Rmírez Uresti
Sergio Ruiz Loza**

01 de diciembre de 2023

Proceso de instalación

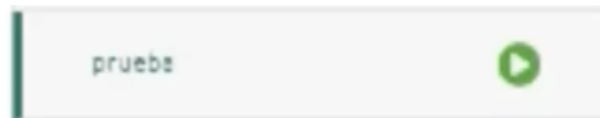
Anaconda Navigator con Python y Mesa para la modelación del SMA

1. Instalación y uso de Anaconda Navigator

- Requisitos técnicos para Anaconda Navigator
Sistema operativo: Windows, macOS o Linux
- Proceso de instalación
 - a. Vaya al sitio web de Anaconda y descargue el instalador para su sistema operativo: <https://anaconda.org/anaconda/anaconda-navigator>
 - b. Ejecute el instalador y siga las instrucciones en pantalla
- Creación de ambiente
 - a. Vaya a la pestaña “Environments”
 - b. Presione el botón “Create” y configure el nombre e instale el paquete de Python, con una versión igual o mayor a 3.10
 - c. Espere a que se cree el ambiente

2. Instalación de MESA y flask

- a. Al tener el ambiente creado se debe instalar el paquete de flask, esperamos a que se descargue para instalarlo en el ambiente
- b. Para descargar MESA debemos dar click en el botón verde junto al nombre del ambiente



Esto abrirá nuestra terminal y debemos colocar el siguiente comando: “pip install mesa” y darle enter, esperar a que se descargue.


- c. Para asegurarnos de que mesa y flask están instalados correctamente podemos en la misma consola poner el siguiente comando: “pip show mesa” y “pip show flask”, aparecerá la información de mesa y de flask como se muestra en la siguiente imagen

```
C:\Users\lorei>pip show mesa
Name: Mesa
Version: 2.1.3
Summary: Agent-based modeling (ABM) in Python 3+
Home-page: https://github.com/projectmesa/mesa
Author: Project Mesa Team
Author-email: projectmesa@googlegroups.com
License: Apache 2.0
Location: C:\Users\lorei\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages
Requires: click, cookiecutter, matplotlib, mesa-viz-tornado, networkx, numpy, pandas, solara, tqdm
Required-by:

C:\Users\lorei>pip show flask
Name: Flask
Version: 3.0.0
Summary: A simple framework for building complex web applications.
Home-page:
Author:
Author-email:
License:
Location: C:\Users\lorei\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages
Requires: blinker, click, itsdangerous, Jinja2, Werkzeug
Required-by:
```

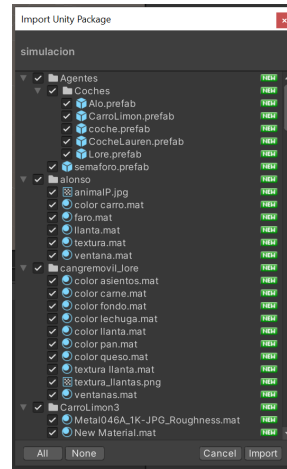
3. Descarga del archivo de python
 - a. Descarga los archivos
 - i. En la siguiente liga se puede descargar los archivos:
<https://github.com/Bernardo0173/TC2008B.301.2/tree/main>
 - ii. Se descargan todos los archivos
 - b. Se extraen los documentos de la carpeta

Configuración para Unity

1. Instalación de Unity
 - a. Requisitos técnicos para Unity
 - i. Sistema operativo: Windows, macOS o Linux
 - ii. Procesador: Intel Core i3 o superior
 - iii. Memoria: 4 GB o más
 - iv. Espacio en disco: 20 GB o más
 - b. Procedimiento de instalación
 - i. Vaya al sitio web de Unity y descargue Unity Hub.
<https://unity.com/download>
 - ii. Ejecuta el instalador y sigue las instrucciones en pantalla.
 - iii. Cuando se complete la instalación, Unity Hub se instalará en su sistema.
 - iv. Abra Unity Hub.
 - v. Haga clic en el botón "Crea un nuevo proyecto".
 - vi. Seleccione el tipo de proyecto que desea crear: 3D.
 - vii. Seleccione una ubicación para el proyecto.
 - viii. Agregue un nombre al proyecto
 - ix. Haga clic en el botón "Crear".
2. Preparación del proyecto
 - a. Descarga el archivo del proyecto
 - i. El archivo se descarga desde github, la liga es la siguiente:
<https://github.com/Bernardo0173/TC2008B.301.2>
 - ii. El archivo es el siguiente:
 1. <https://github.com/Bernardo0173/TC2008B.301.2/blob/main/Gr%C3%A1ficos/avanceBuildingsCompleto.unitypackage.zip>
 2. Para descargarlo se debe dar click en el siguiente botón:

 - b. Coloca el archivo del proyecto en la carpeta de archivos de Unity
 - i. Después de descargar el archivo y descomprimirlo debemos introducirlo en la carpeta de archivos de Unity, esto lo podemos hacer al abrir Unity Hub y darle click en el apartado de agregar que está arriba a la derecha,

ahí podemos seleccionar el archivo que queremos agregar. O bien, una vez con nuestro proyecto nuevo en Unity abierto, en Documentos: abrir el archivo (ya descomprimido) y se intentará importar automáticamente en Unity:

3.

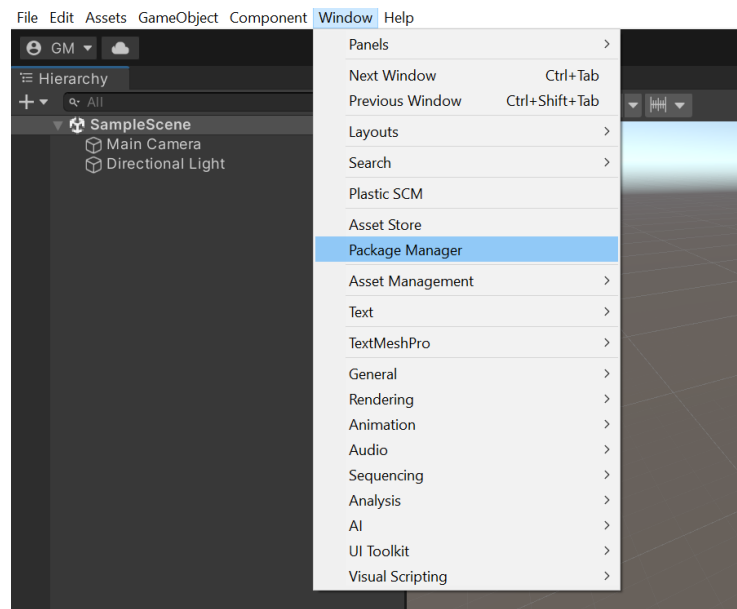


i.

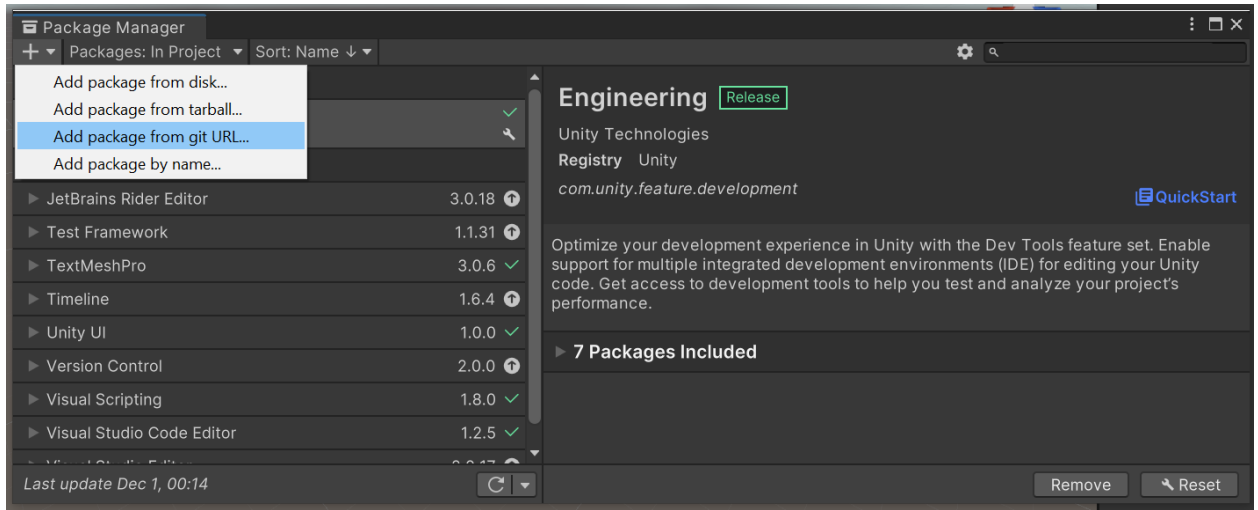
Presionar “Import”

- ii. Agregar la dependencia Json.Net
Para agregar el package:

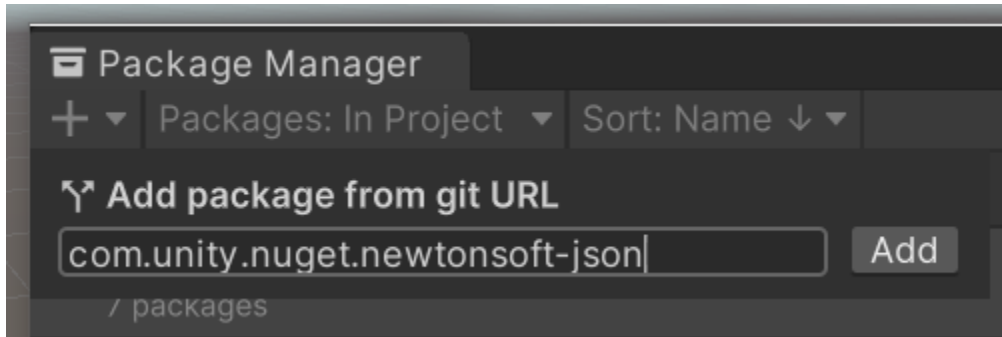
1. Dirigirse a “Window” -> “Package Manager



2. Se abrirá la siguiente ventana



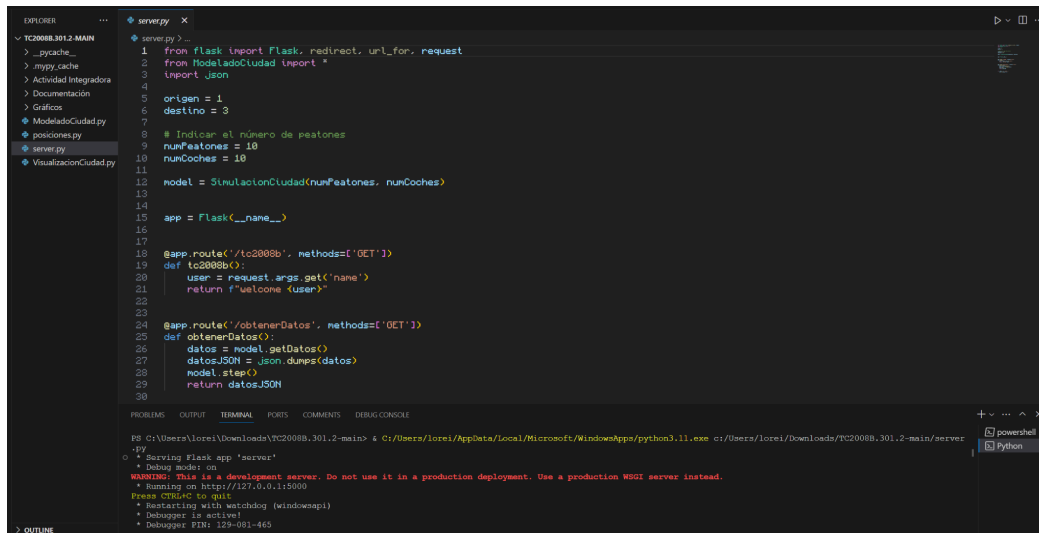
3. Presionar “+” -> “Add package from git URL”



Pegar en el cuadro de texto el siguiente link: `com.unity.nuget.newtonsoft-json` y presionar “Add”

4. Ejecutar el proyecto

- a. Para ejecutar el proyecto en Unity, previamente se debe haber iniciado el servidor en Anaconda Navigator

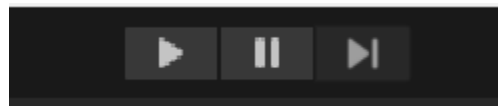


```
server.py
1 from flask import Flask, redirect, url_for, request
2 from ModeladoCiudad import *
3 import json
4
5 origen = 1
6 destino = 3
7
8 # Indicar el número de peatones
9 numPeatones = 10
10 numCoches = 10
11
12 model = SimulacionCiudad(numPeatones, numCoches)
13
14
15 app = Flask(__name__)
16
17
18 @app.route('/tc2008b', methods=['GET'])
19 def tc2008b():
20     user = request.args.get('name')
21     return f'welcome {user}'
22
23
24 @app.route('/obtenerDatos', methods=['GET'])
25 def obtenerDatos():
26     datos = model.getDatos()
27     datosJSON = json.dumps(datos)
28     model.step()
29     return datosJSON
30
```

PROBLEMS OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE

PS C:\Users\lorei\Downloads\TC2008B.301.2-main> & C:\Users\lorei\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Users\lorei\Downloads\TC2008B.301.2-main/server.py

- Serving Flask app "server"
- Debug mode: on
- WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
- Running on http://127.0.0.1:5000
- Press CTRL+C to quit
- Restarting with watchdog (windowsapi)
- Debugger is active!
- Debugger PIN: 129-081-465



b.

Presionar el botón de “Play” en Unity y enseguida cambiar la vista de “Game” a “Scene”:

