**Tecnológico Nacional de México**

**Instituto Tecnológico de la Laguna**



**Inteligencia Artificial**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**Portafolio de evidencia de la unidad 2**

Presenta:

Aracely Cisneros Rdz 20130838

David Alejandro Luna González C20130522

Ricardo Rodríguez Pacheco 18131275

Shalom Isai Salazar Arguijo 20130805

Torreón, Coahuila

Noviembre del 2023

# Índice

[**Introducción**](#_heading=h.nkl3kgw1d9he) **3**

[**Lógica difusa con librería**](#_heading=h.tyjcwt) **4**

[**Conclusiones**](#_heading=h.uge3svq2n72c) **16**

[**Referencias**](#_heading=h.wld2sfy78s04) **17**

# Introducción

Cuando nosotros pensamos que es lo que queremos hacer generalmente pensamos entre más de una opción, como lo vamos a hacer, también pensamos varias formas de cómo llegar a nuestro objetivo final, o intermedio, o cualquiera que sea el fin. Todo esto lo hacemos porque para nosotros como personas es nuestra forma de razonar, de pensar para solucionar nuestros problemas.

En el mundo de las computadoras, está la lógica difusa, que es diferente a la lógica clásica que conocemos, aquella con la que se trabaja con verdadero y falso, unos y ceros, en cambio , la lógica difusa, que viene del inglés “Fuzzy Logic”, difuso que puede significar que es vago o no muy claro. Esta lógica se basa en que los sistemas de cómputo en nuestro caso puedan trabajar con valores no solamente exclusivos de unos y ceros.

Por ejemplo, con la forma de trabajar de un ventilador común, una persona normalmente escoge la velocidad que quiera, baja, normal o alta, y no solamente encenderlo o apagarlo.

La lógica difusa es valiosa en situaciones donde las reglas no son claras, y la incertidumbre y la imprecisión están presentes. Permite un enfoque más flexible y adaptable, lo que la convierte en una herramienta importante para abordar problemas en los que la imprecisión y la incertidumbre son inherentes, contribuyendo a la mejora de la toma de decisiones y la adaptabilidad en entornos complejos.

# 

**Desarrollo**

Este código crea un sistema de control difuso para calcular la propina en un restaurante. El sistema tiene tres variables difusas:

* Servicio: La calidad del servicio recibido.
* Comida: La calidad de la comida recibida.
* Propina: El monto de la propina a dejar.

Cada variable difusa tiene términos o calificaciones

**Servicio del mesero**:

* Pésimo: El servicio fue muy malo.
* Promedio: El servicio fue aceptable.
* Excelente: El servicio fue muy bueno.

**Estado de la comida**:

* Desagradable: La comida fue muy deficiente.
* Aceptable: La comida fue aceptable.
* Deliciosa: La comida fue excelente.

El sistema tiene nueve reglas:

* Regla 1: Si el servicio fue pésimo y la comida fue desagradable, la propina será nada.
* Regla 2: Si el servicio fue pésimo y la comida fue aceptable, la propina será nada.
* Regla 3: Si el servicio fue pésimo y la comida fue excelente, la propina será poca.
* Regla 4: Si el servicio fue promedio y la comida fue desagradable, la propina será nada.
* Regla 5: Si el servicio fue promedio y la comida fue aceptable, la propina será normal.
* Regla 6: Si el servicio fue promedio y la comida fue excelente, la propina será normal.
* Regla 7: Si el servicio fue excelente y la comida fue desagradable, la propina será nada.
* Regla 8: Si el servicio fue excelente y la comida fue aceptable, la propina será normal.
* Regla 9: Si el servicio fue excelente y la comida fue excelente, la propina será generosa.

Para usar el sistema, primero se deben establecer los valores de entrada para las variables difusas de servicio y comida. En este caso, el servicio se establece en 5.5 y la comida se establece en 8.4.

Una vez que se establecen los valores de entrada, el sistema calcula la salida, que es el monto de la propina. En este caso, el monto de la propina es de 10.507670162168358.

El lenguaje de programación será Python. Este código en Python utiliza la biblioteca scikit-fuzzy (skfuzzy) para implementar un sistema de lógica difusa para calcular la cantidad de propina en un restaurante basándose en la calidad del servicio y de la comida.

* Se importa numpy para cálculos numéricos.
* Se importa skfuzzy para trabajar con lógica difusa.
* Se importa el módulo de control de skfuzzy para definir y simular el sistema de control difuso.

**a. Análisis del problema**

El sistema de control difuso actual tiene conjuntos difusos y reglas predefinidas para evaluar la calidad del servicio y la comida en un restaurante.

**b. Objetivo y justificación**

Desarrollar una versión mejorada del sistema de control difuso que permita una configuración más flexible de conjuntos difusos y reglas.

La mejora del sistema permitirá una adaptabilidad más amplia a diferentes entornos de restaurantes, proporcionando una solución más robusta y personalizable.

**c. Especificación de requerimientos**

**Hardware:**

* Computadora con SO Windows, MacOs o Linux reciente
* Procesador de doble núcleo a 2.0 GHz o superior.
* Memoria RAM de al menos 4 GB.
* Espacio en disco duro: al menos 100 MB de espacio libre.

**Sistema Operativo:**

El programa es compatible con Windows, macOS y versiones posteriores, y distribuciones de Linux.

**Requerimientos de Software:**

Versión de Python: El programa requiere Python 3.0 o superior.

Bibliotecas de Python: Se deben instalar las bibliotecas necesarias mediante el siguiente comando:

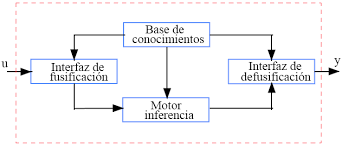
pip install numpy scikit-fuzzy matplotlib

**d. Diagramas de procesos y secuencias, prototipo, modelo de datos(diseño)**

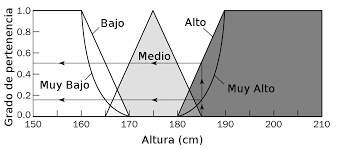
**Diagrama de procesos**



**Prototipo**



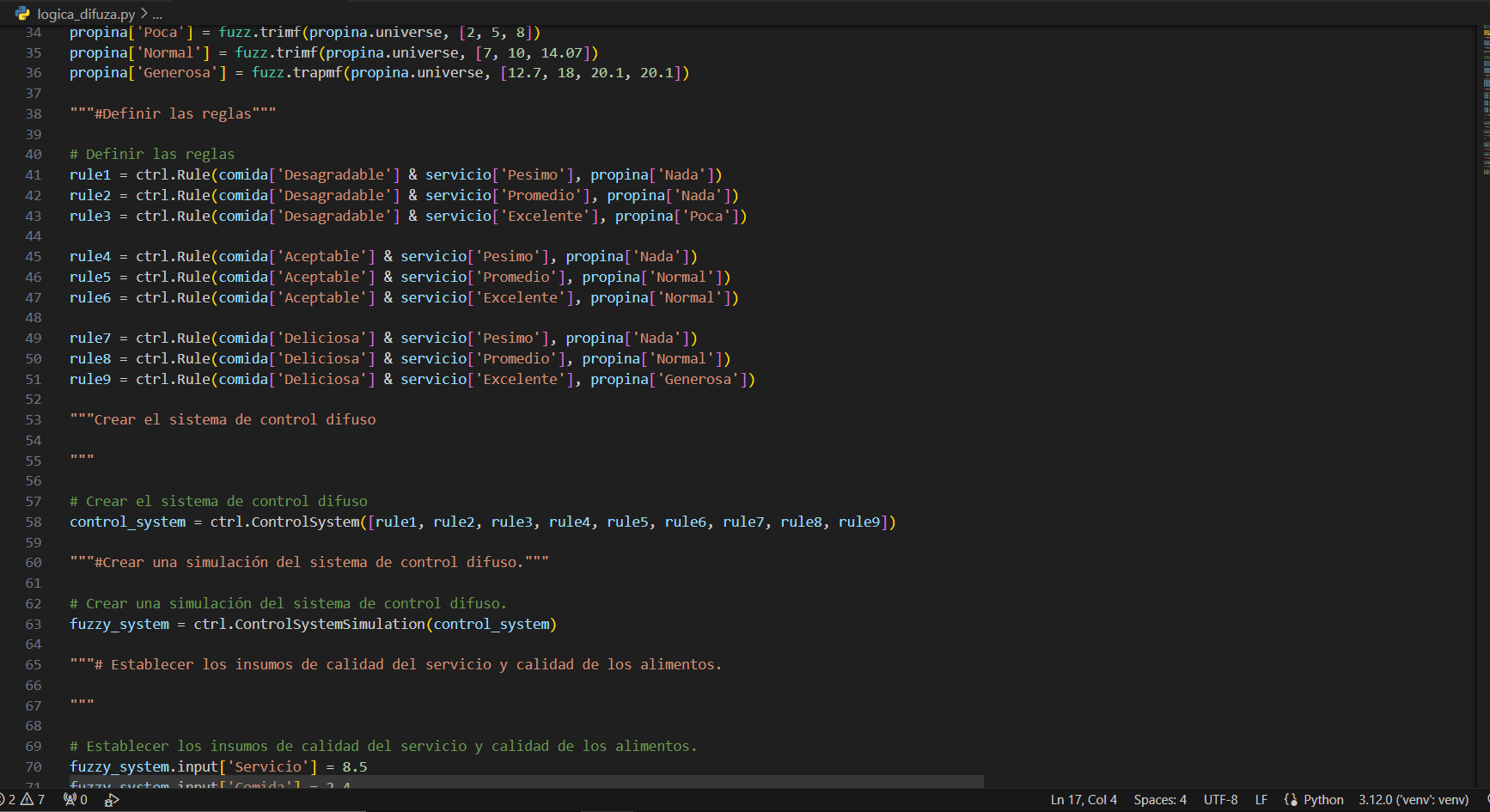
**Diseños**

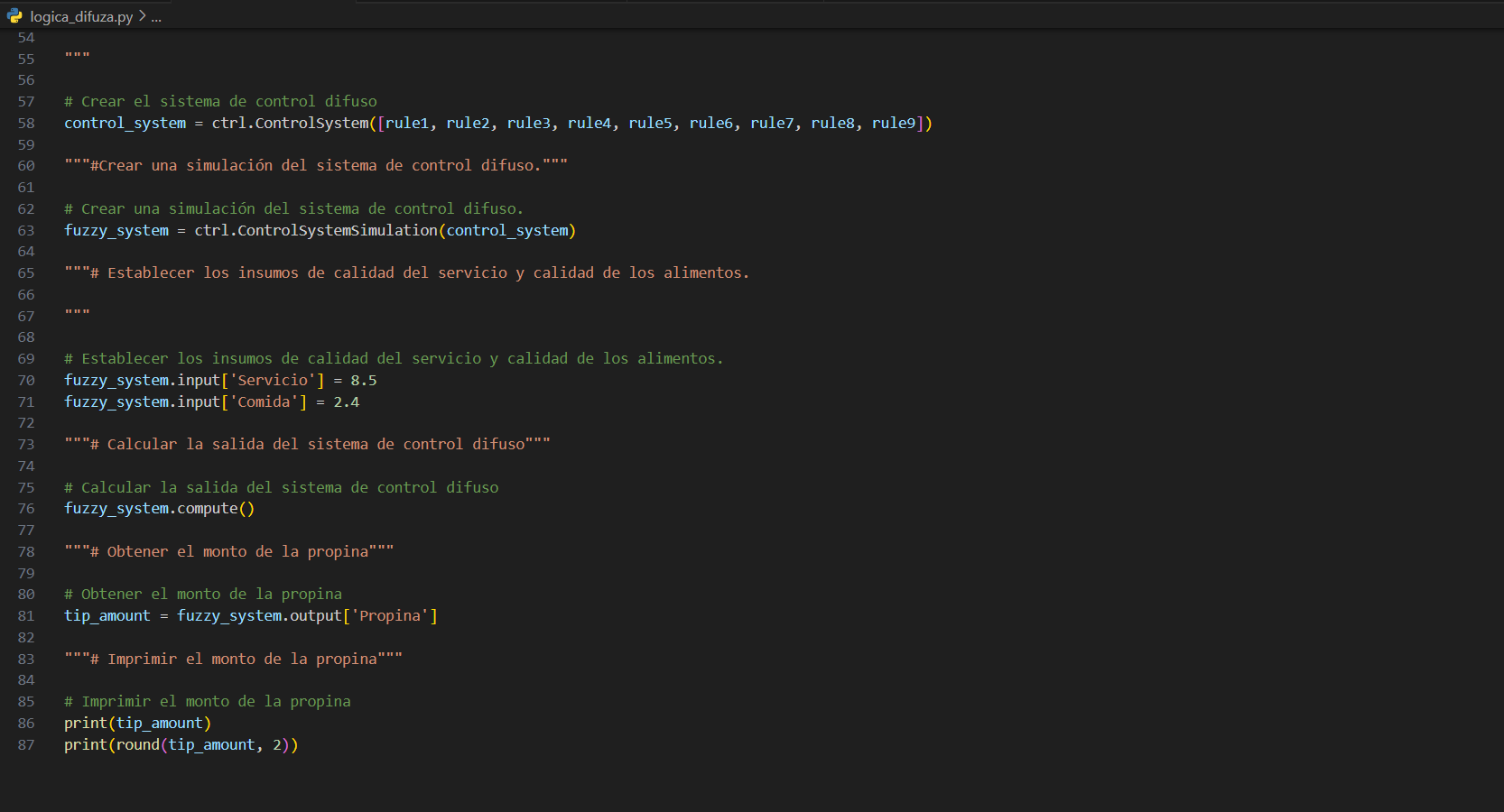
****

**Lógica difusa con librería**

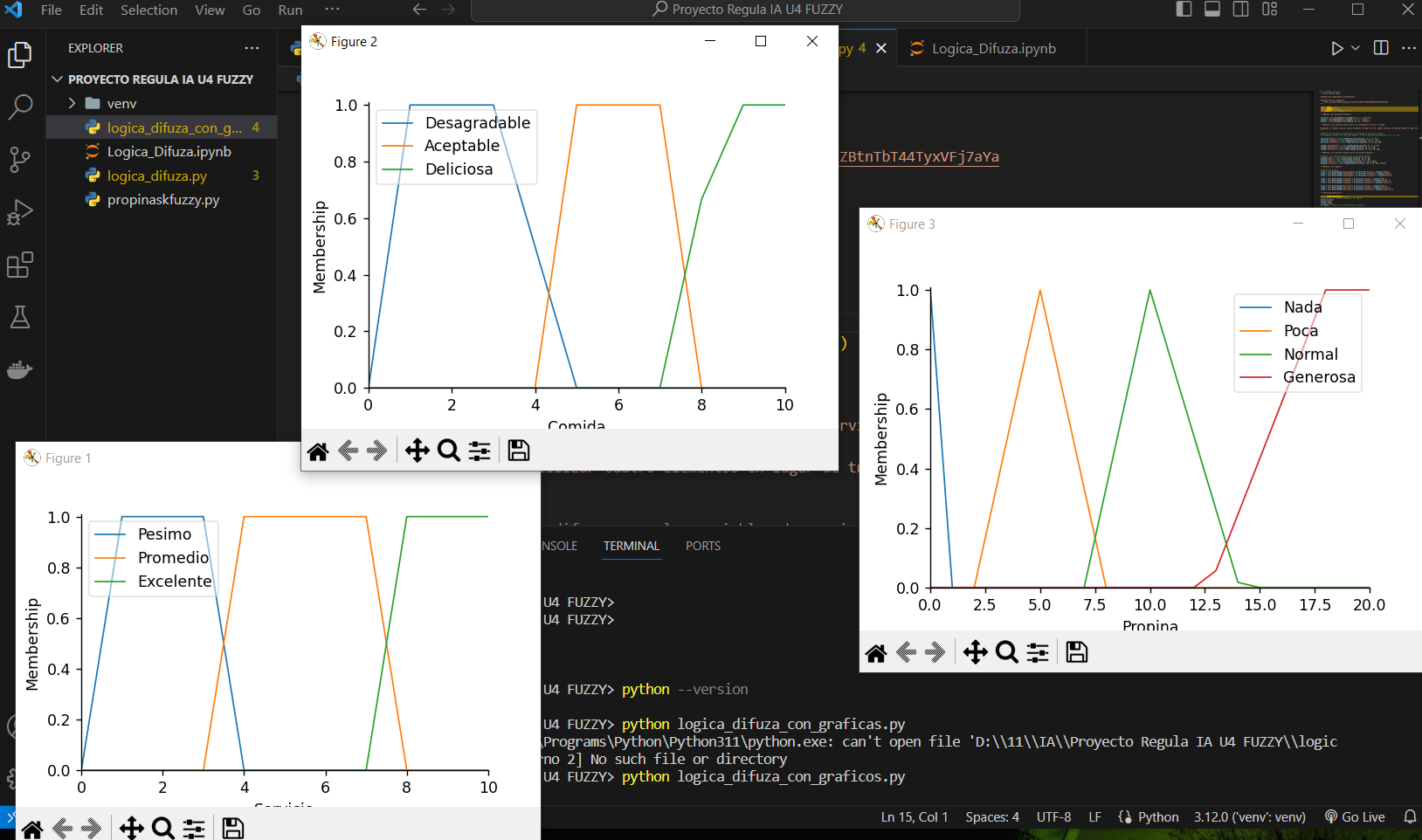
**e. Implementación(código, documentado)**

****

****

****

**Pruebas**



**Manuales: usuario, tecnico e instalacion**

**Manual Tecnico**

Este manual proporciona detalles sobre la implementación de un programa de lógica difusa, incluyendo la estructura de la red neuronal, la compilación con el optimizador y la función de pérdida, así como la visualización de los resultados.

Este manual técnico proporciona información detallada para los desarrolladores y personal técnico que participarán en el mantenimiento, modificación o extensión del sistema de control difuso. Incluye detalles sobre la arquitectura, el diseño y la implementación del sistema.

Este manual abarca aspectos técnicos del sistema, incluyendo la estructura del código, las dependencias, la lógica de negocio y otros aspectos relevantes para aquellos que trabajarán en el desarrollo y mantenimiento del sistema.

Reglas generales para dar propina, lógica del programa

If (comida is Desagradable) and (servicio is Pesimo) then (propina is nada)

If (comida is Desagradable) and (servicio is Promedio) then (propina is nada)

If (comida is Desagradable) and (servicio is Excelente) then (propina is poca)

If (comida is Aceptable) and (servicio is Pesimo) then (propina is nada)

If (comida is Aceptable) and (servicio is Promedio) then (propina is normal)

If (comida is Aceptable) and (servicio is Excelente) then (propina is normal)

If (comida is Deliciosa) and (servicio is Pesimo) then (propina is nada)

If (comida is Deliciosa) and (servicio is Promedio) then (propina is normal)

If (comida is Deliciosa) and (servicio is Excelente) then (propina is generosa)

**Requisitos del sistema:**

* Python 3.x
* Skfuzzy
* Numpy
* Matplotlib
* Una computadora con conexión a internet
* Visual Studio Code

**Instalación**

Si desea ejecutar el programa desde su computadora y alguno de estos programas no está incluido en su Sistema Operativo tendrá que realizar los siguientes pasos en su computadora.

1. Instalar Python desde python.org.
2. Instalar Skfuzzy: [Comando] pip install scikit-fuzzy.
3. Instalar Numpy: [Comando] pip install numpy.
4. Instalar Matplotlib: [Comando] pip install matplotlib.

**Ejecución:**

Ejecutar el script de Python.

Verificar que estén instaladas todas las bibliotecas requeridas.

Observar las gráficas de los valores y el valor resultante.

**Entorno virtual**

Si se desea, puede generar un entorno virtual para instalar ahí todas las bibliotecas.

**Crear comando virtual en Windows**

python -m venv venv

**Activación**

venv\Scripts\activate.bat

**Instalar librerias dentro del entorno virtual**

pip install numpy scikit-fuzzy matplotlib

**Desactivación**

deactivate

**Manual de usuario**

Este manual te guiará a través de la instalación, configuración y uso de nuestro sistema para ayudarte a determinar la propina en función de la calidad del servicio y la comida.

**Instalación**

**Proceso de Instalación**

Por favor seguir el proceso de instalación explicado en el manual técnico

**Requisitos**

**Hardware:**

Una computadora reciente con sistema operativo Windows 7 o superior, con un procesador dual-core a 2.0 GHz o superior, a partir de 2 o 4 GB de RAM y al menos 200 MB de espacio libre en disco.

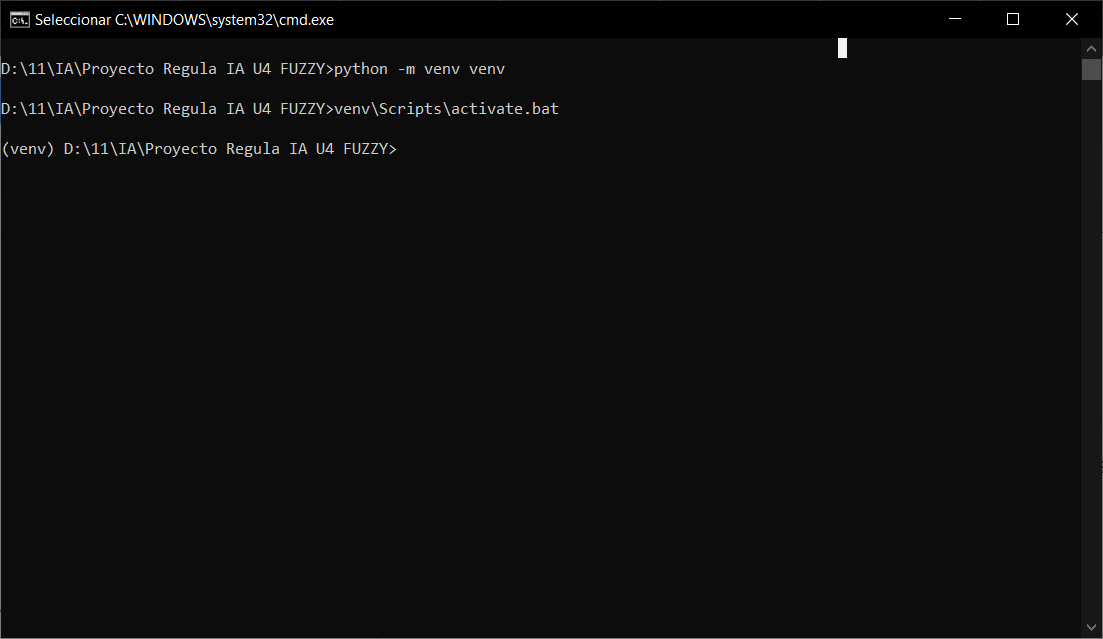
**Software:**

Python 3.0 o superior instalado.

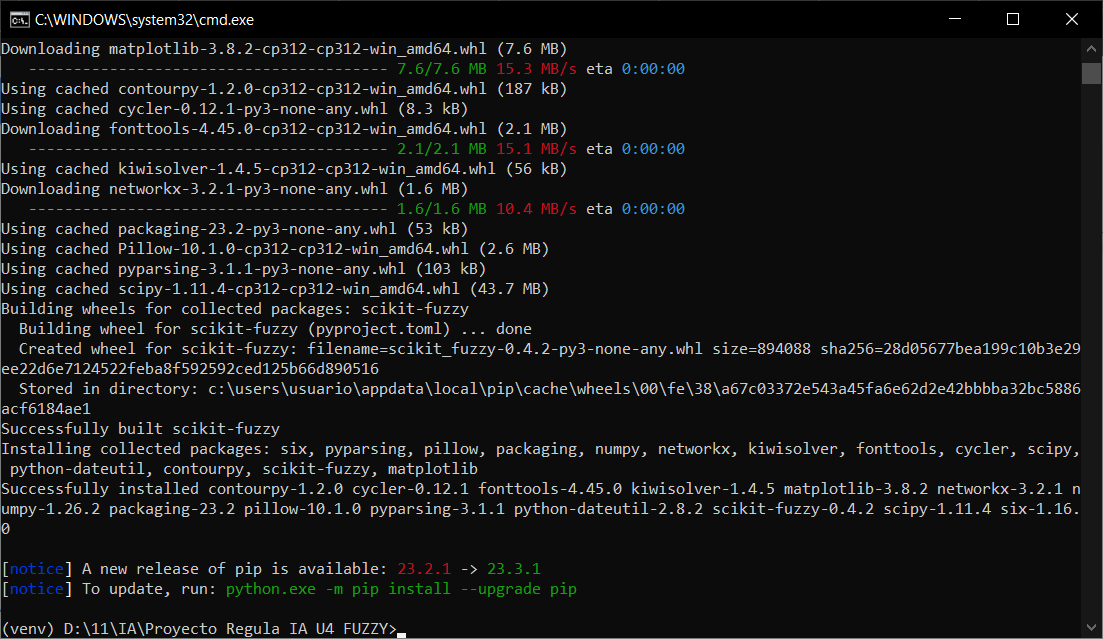
**Versión de python usado**: Python 3.11.5

**Como usar**

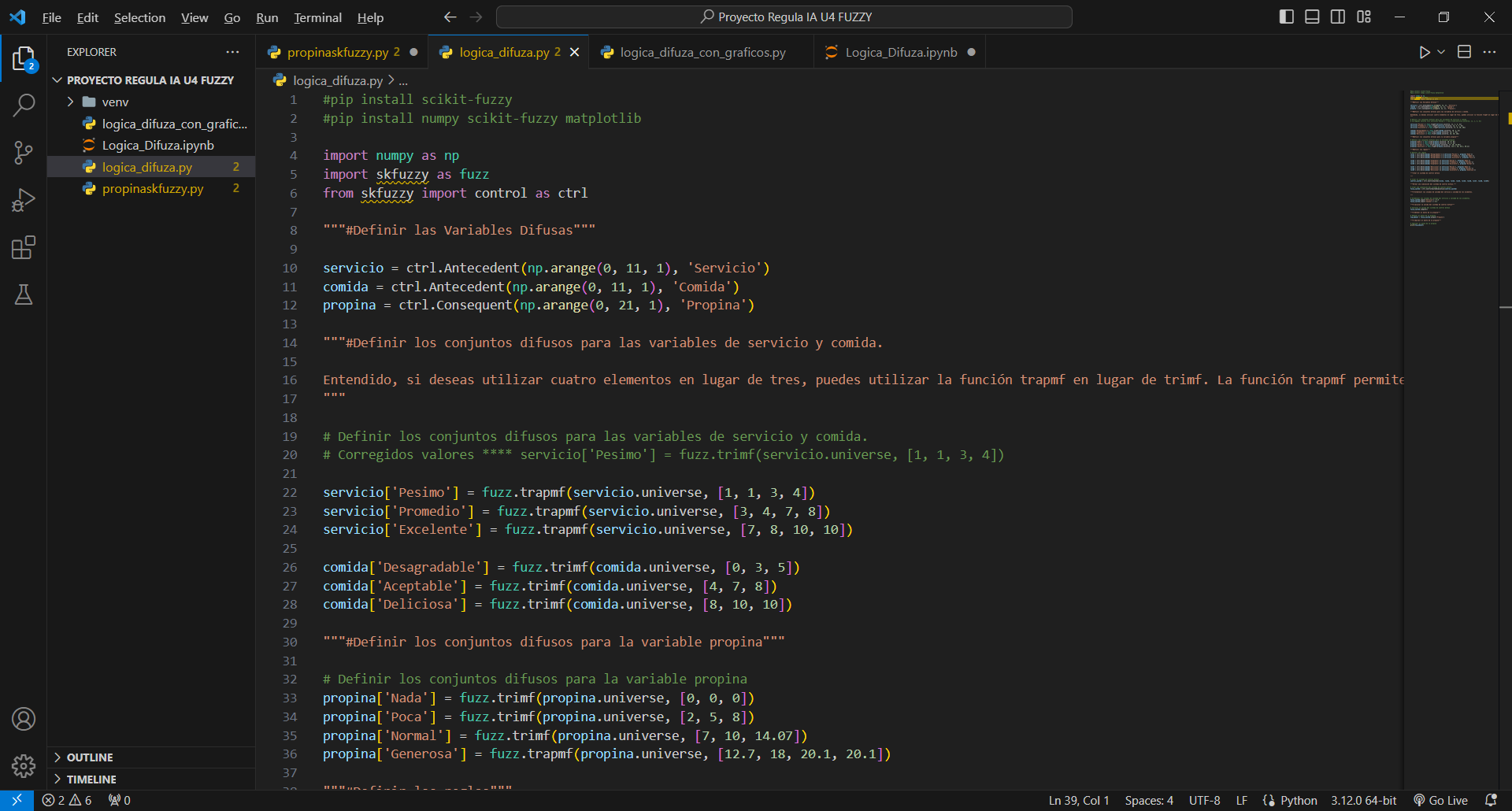
1. Tener el proyecto en un carpeta
2. Abrir una terminal como CMD
3. Crear un entorno virtual y activarlo



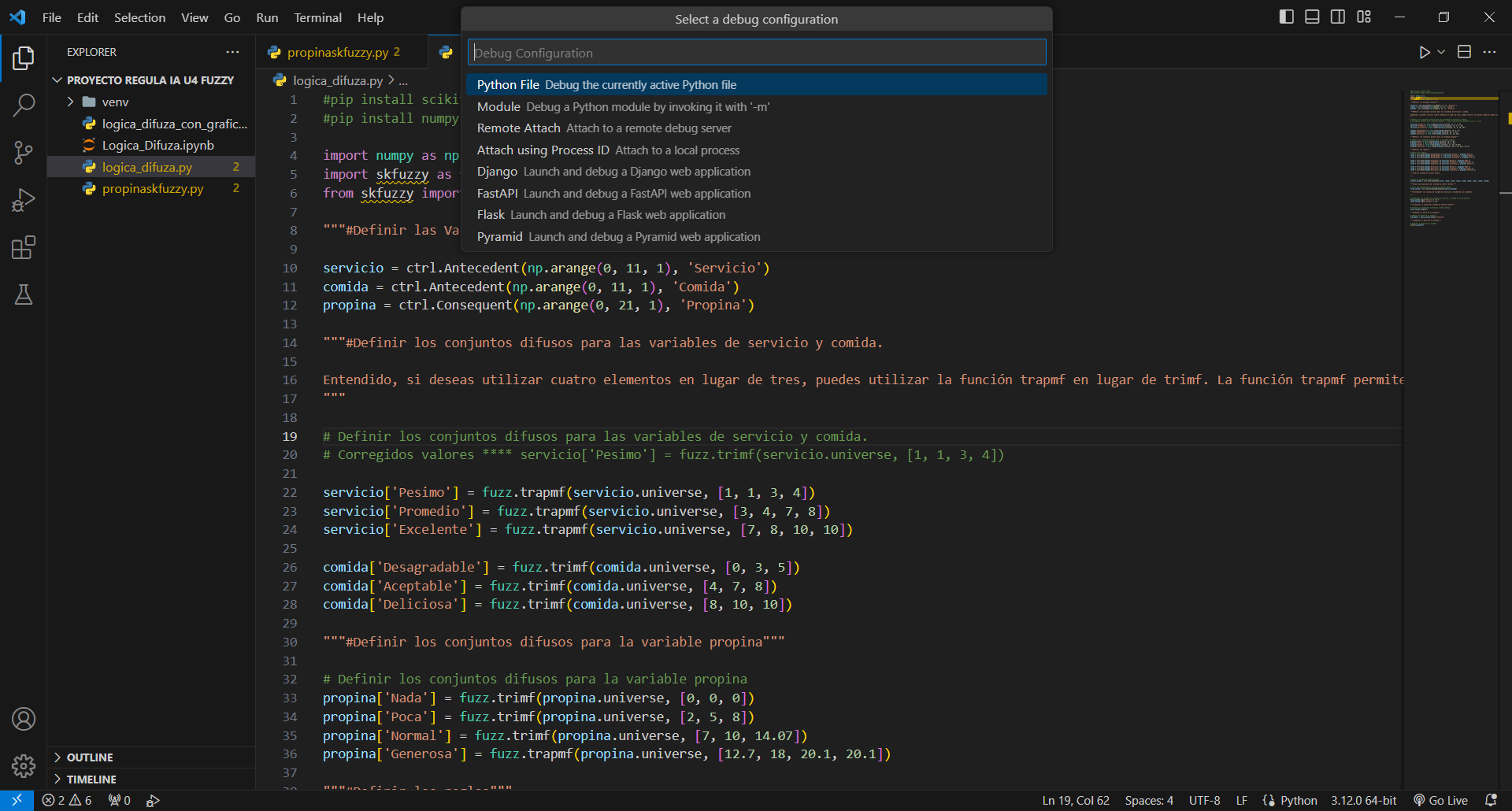
1. Instalar librerias



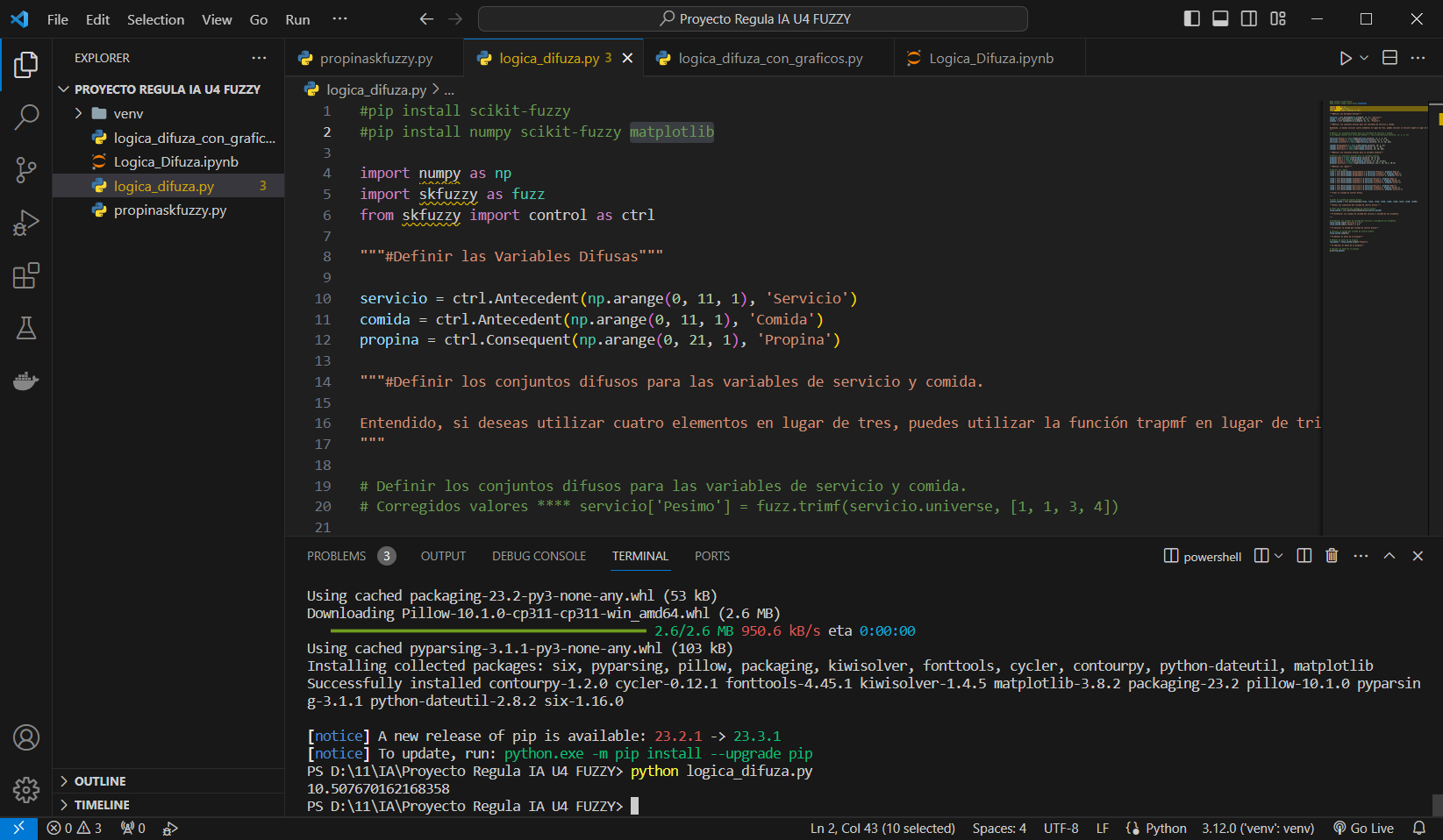
1. Abrir el proyecto en un editor como Visual Studio Code



1. Seleccionar entorno virtual.
2. Ejecutar el archivo con python logica\_difuza.py en una terminal asegurándonos que la ruta este en donde se encuentra el archivo



1. Observar resultado en consola



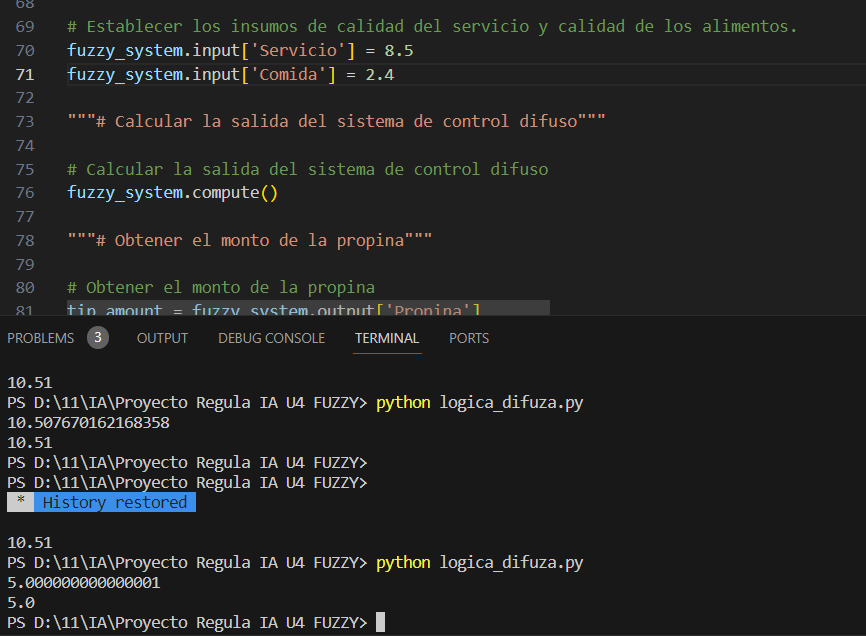
En este caso el resultado fue 10.507670162168358, redondeado a 10.51

con una variables de entrada calificadas de Servicio=5.5 y Comida=8.4

Cambiar los resultados con datos de su agrado.

Por ejemplo Servicio=8.5 y Comida=2.4

Guardamos los cambios y.



Nos dice que da una propina del 5%.

Para ver los datos de las gráficas usar la versión de: logica\_difuza\_con\_graficos.py o agregando las siguientes lineas de código al programa logica\_difuza.py antes de crear el control de lógica difuso.

import matplotlib.pyplot as plt

# Visualizar los conjuntos difusos y las reglas

servicio.view()

comida.view()

propina.view()

plt.show() # Muestra la visualización en pantalla

# También puedes visualizar las reglas

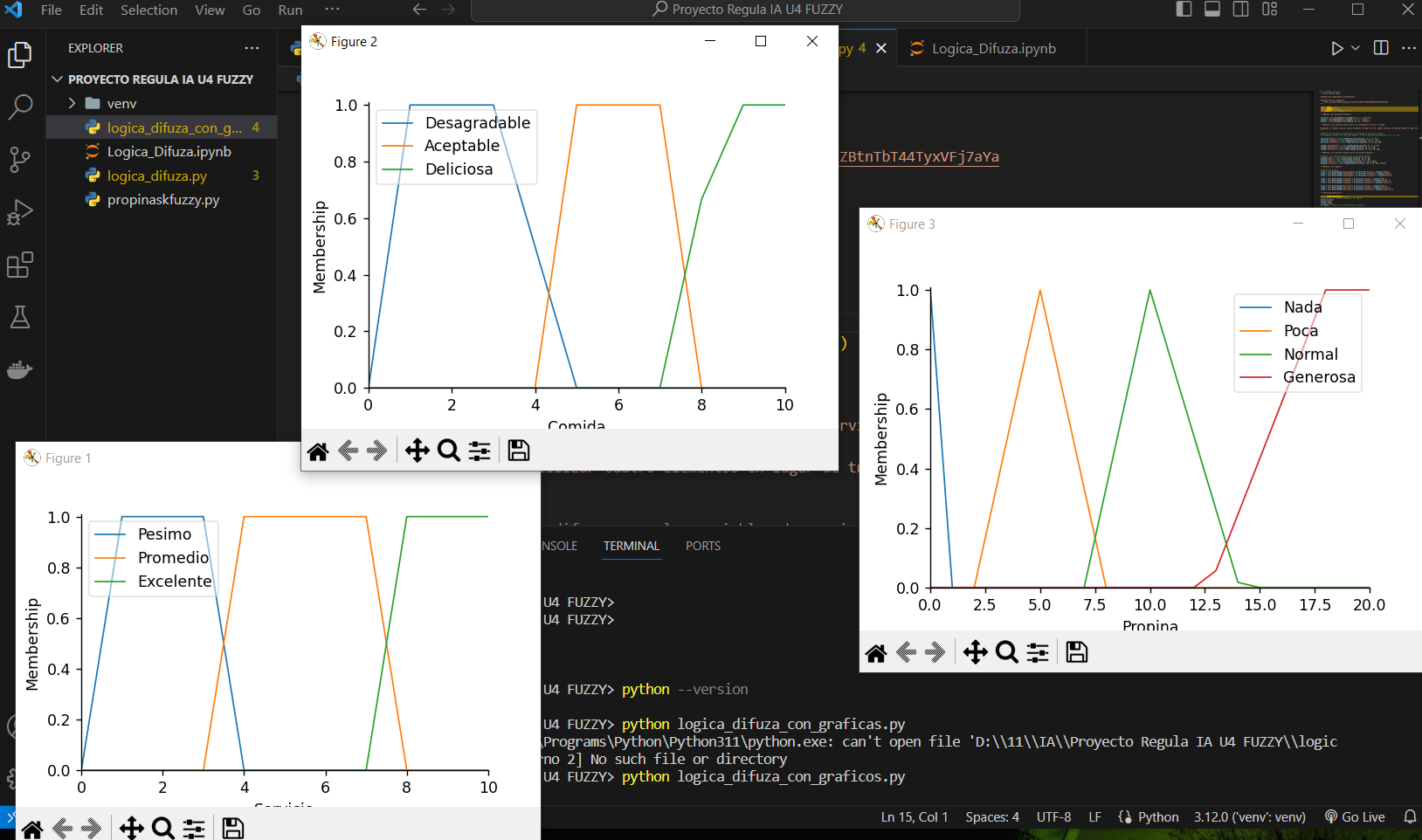
rule1.view()

rule2.view()

rule3.view()

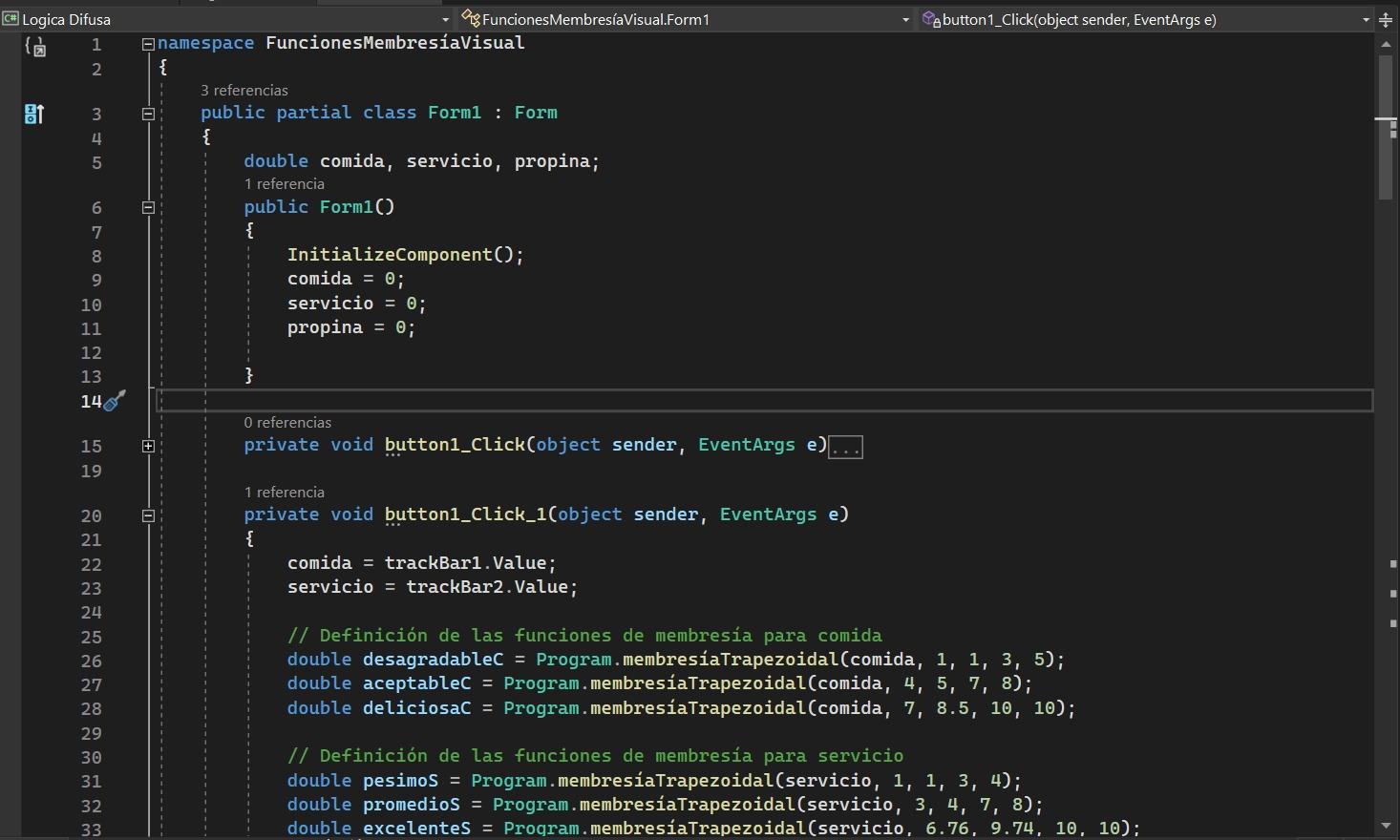
# ... visualizar las demás reglas

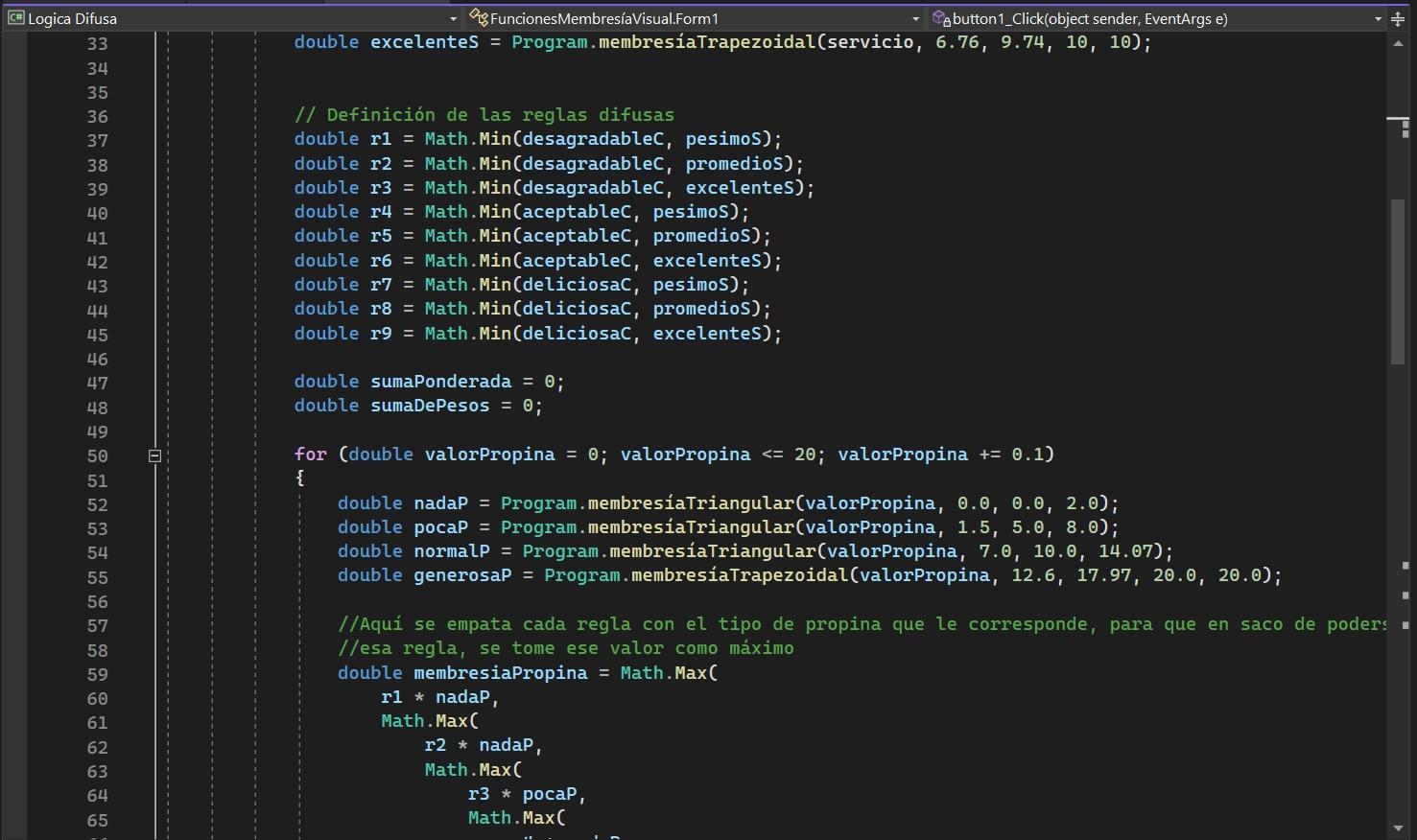
plt.show() # Muestra la visualización en pantalla

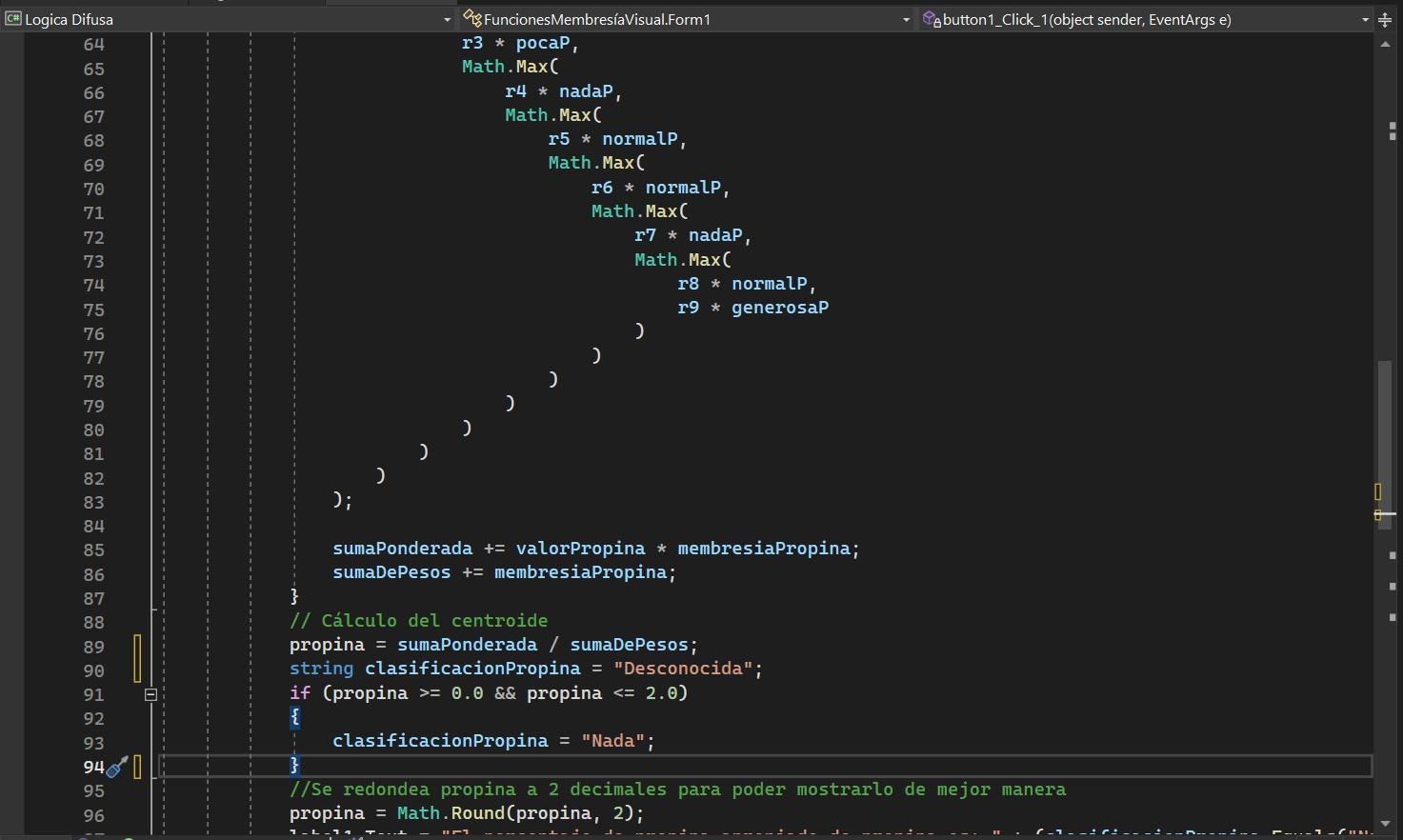


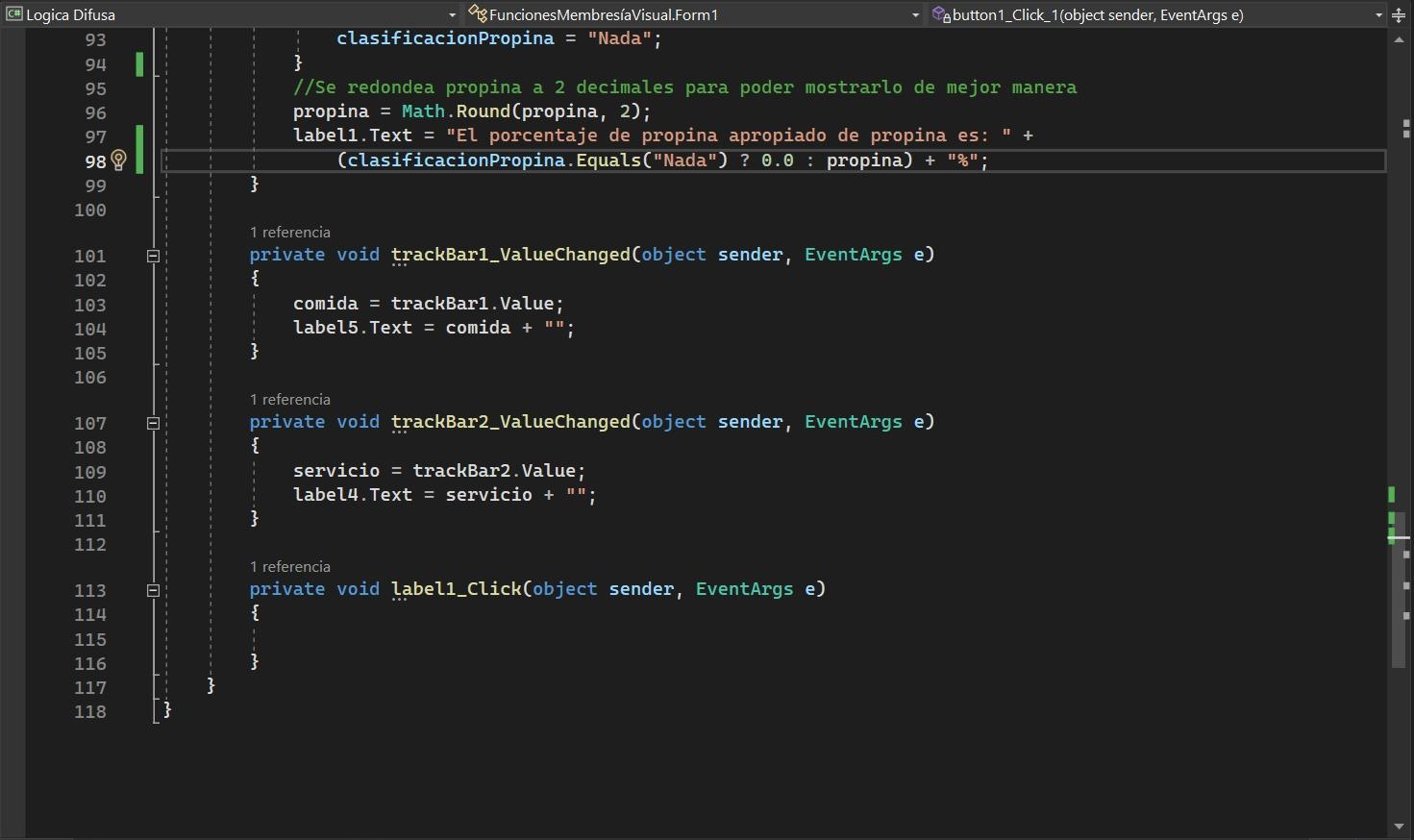
**Logica difusa sin librería**

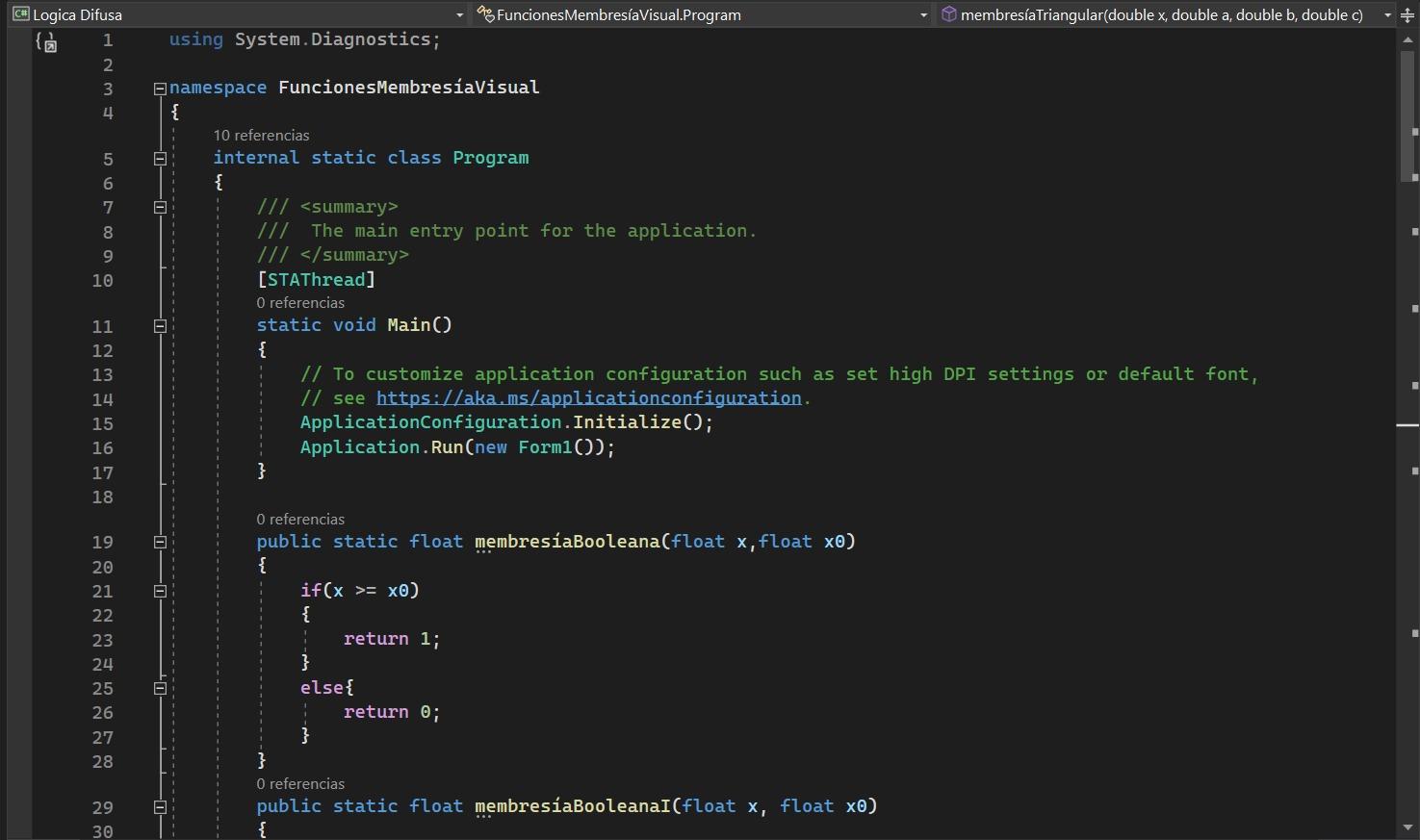
**e. Implementación(código, documentado)**

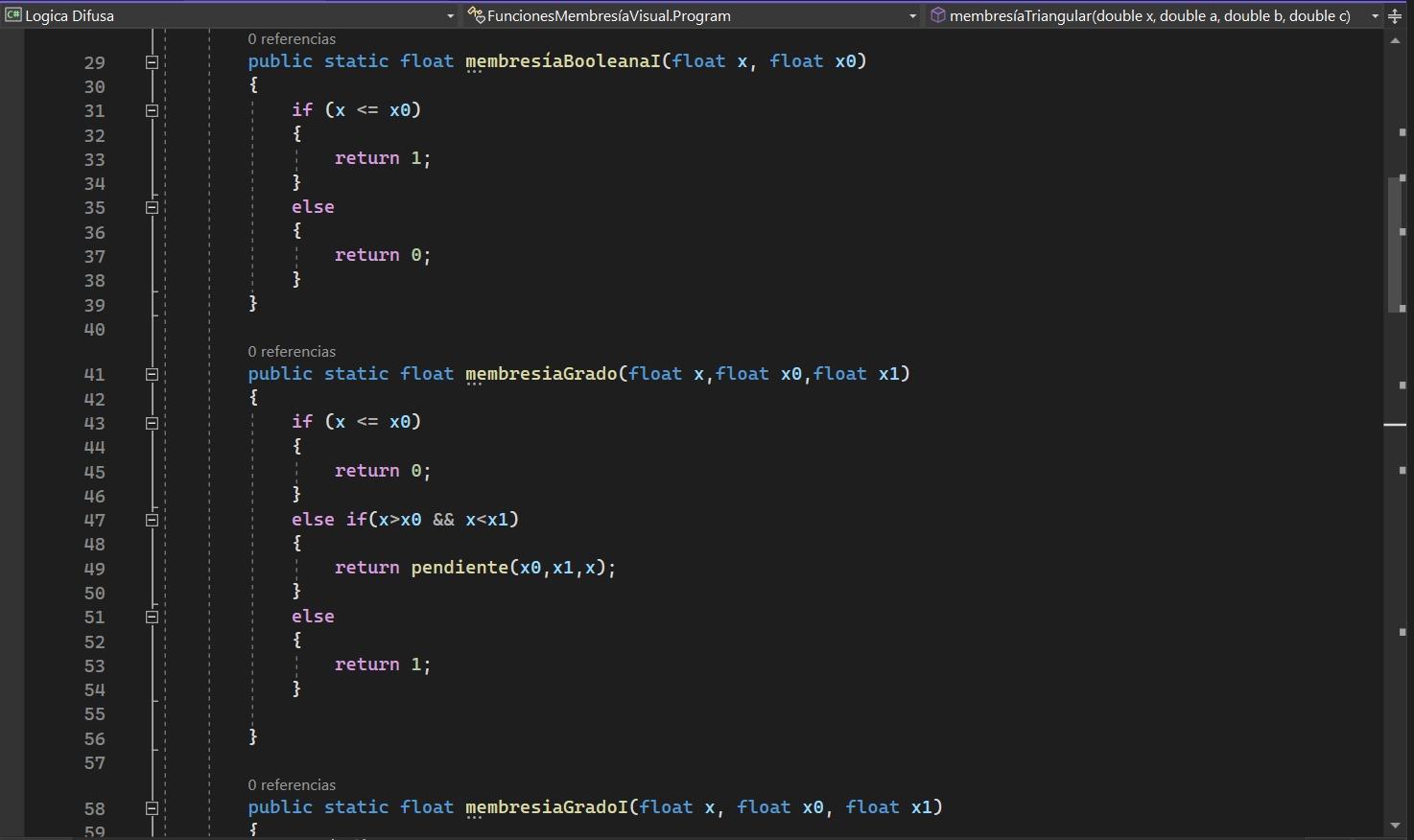


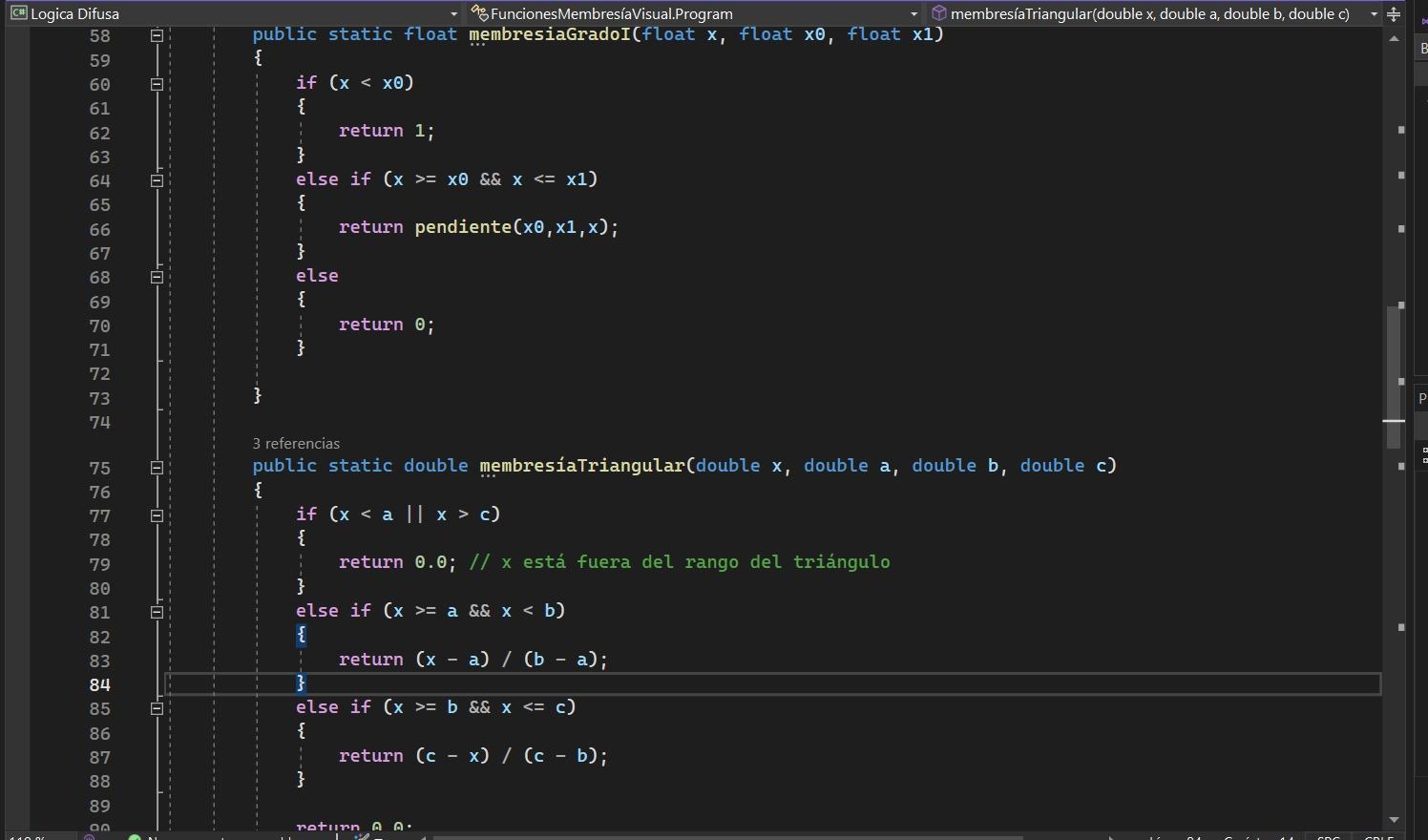


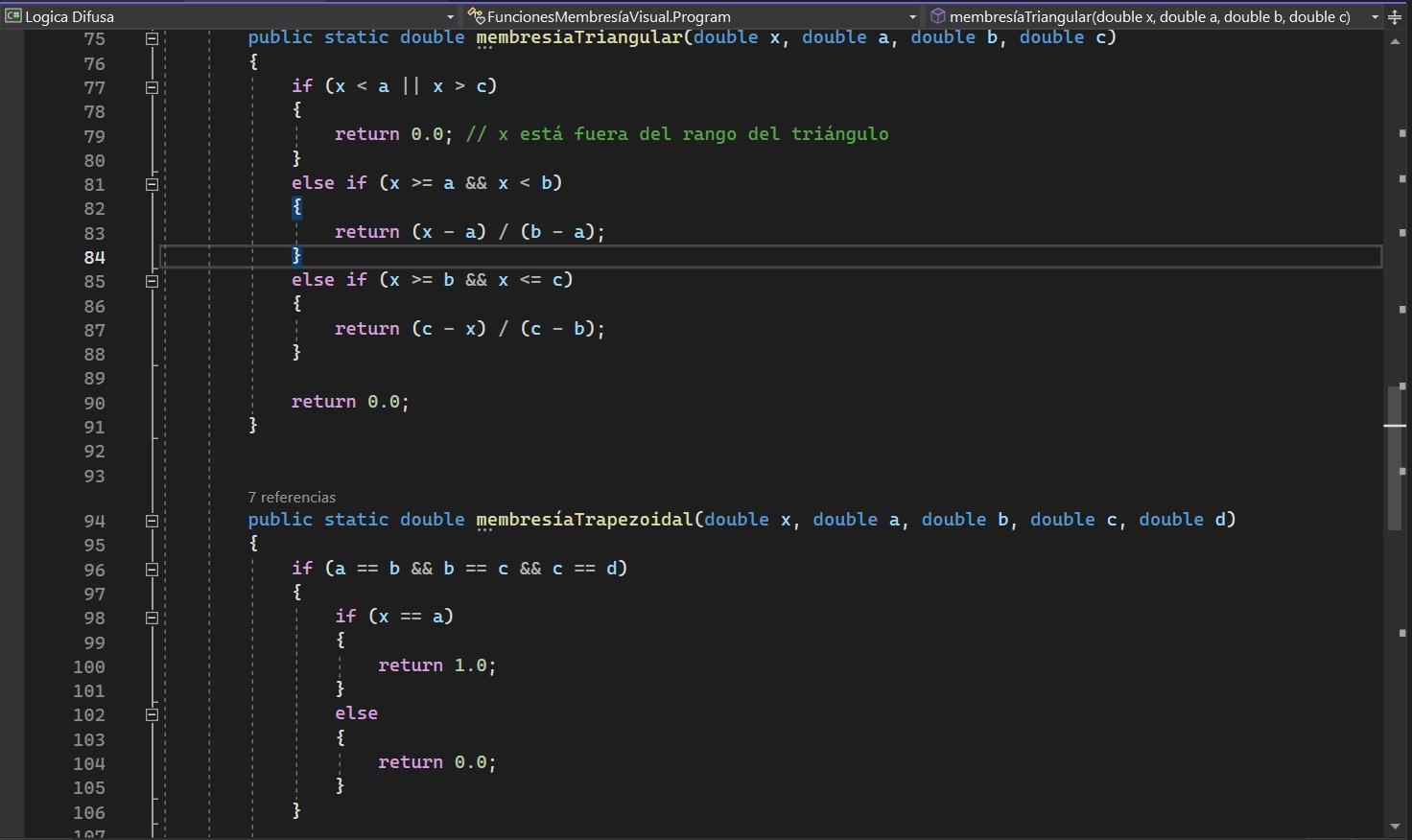


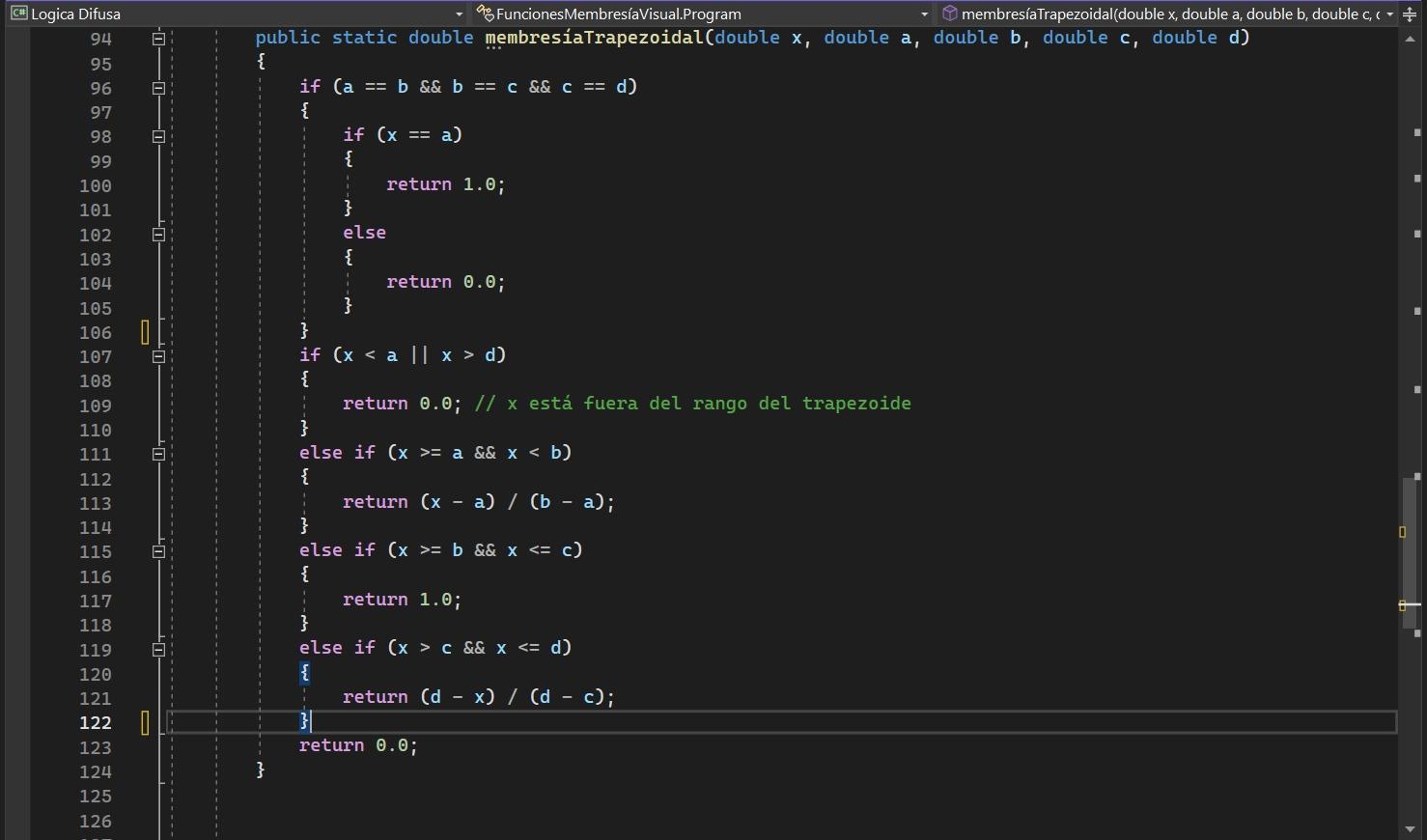










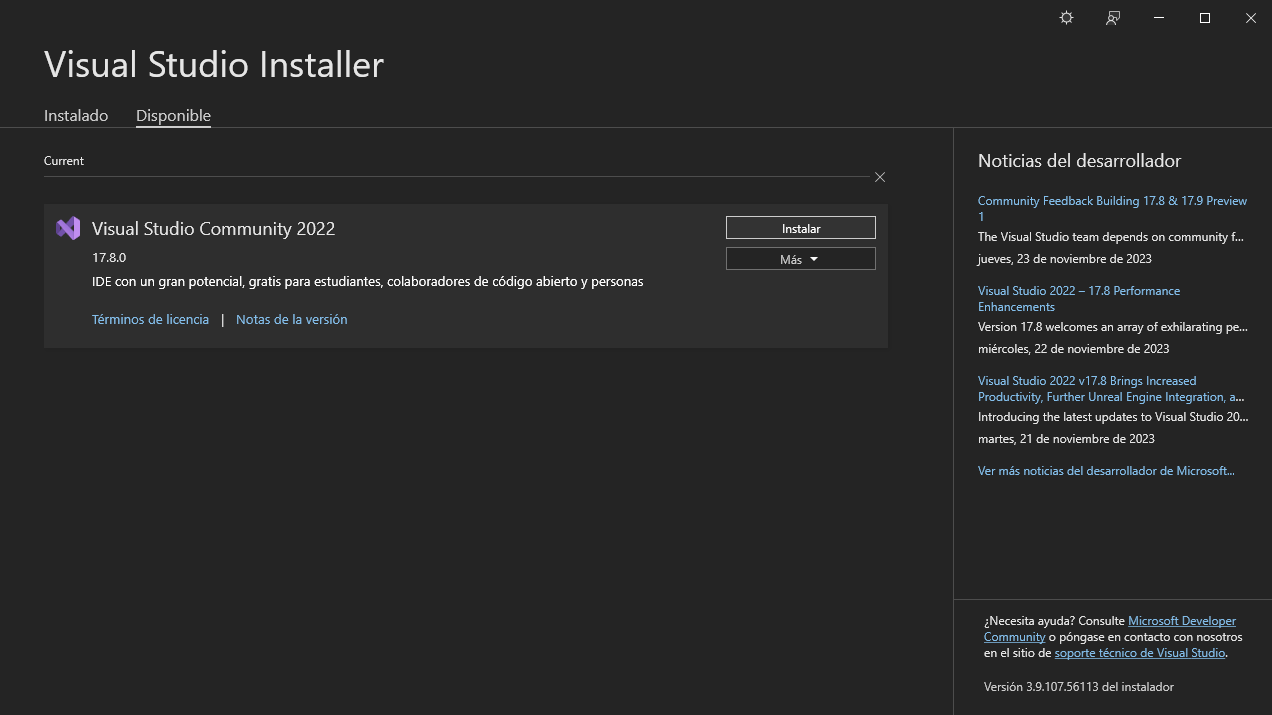


**Manual técnico**

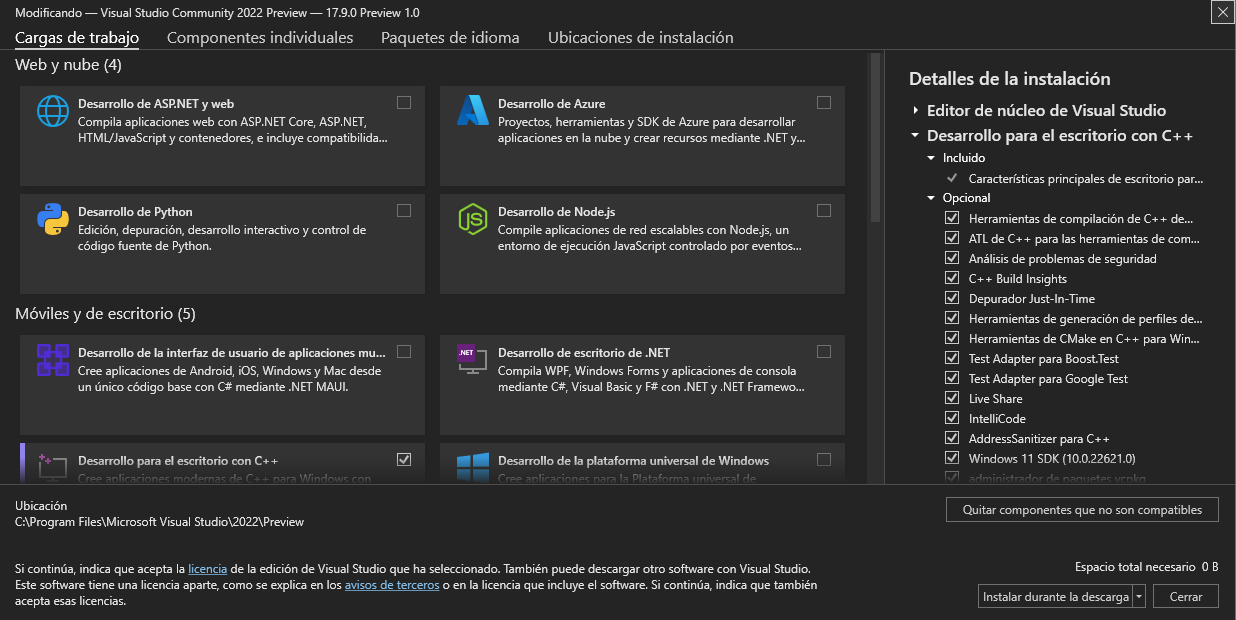
Para el uso de nuestro proyecto, es necesario tener descargada previamente el editor de codigo usado, Visual Studio 2022.

1. Para ello, nos dirigimos a la pagina de descarga de Visual Studio, podemos realizarlo dando clic en el siguiente link: <https://visualstudio.microsoft.com/es/downloads/>
2. Podemos elegir distintas ediciones de VS, pero se recomienda la descarga de la versión de Community





1. Después de ello, proseguiremos a descargar los paquetes necesarios para su uso

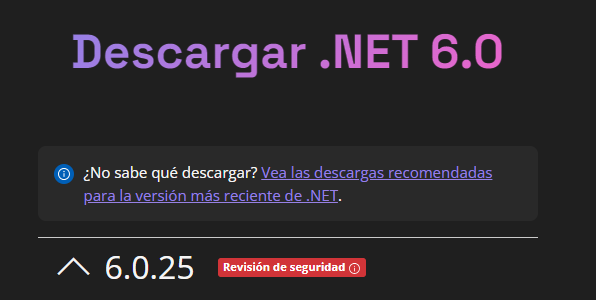


1. Normalmente su descarga puede durar unos minutos, por lo que es recomendable ser pacientes durante este proceso

**Manual de Usuario**

A continuación, pasaremos a resaltar los aspectos necesarios para finalmente realizar la ejecución del proyecto

1. Necesitaremos ahora la descarga e instalacion de .net 6.0, para ello podemos hacer clic en el siguiente enlace: <https://dotnet.microsoft.com/es-es/download/dotnet/6.0>
2. Dentro de dicho enlace, encontraremos la version que mejor se nos acople, tomando en cuenta nuestro sistema operativo y la arquitectura con la que cuente nuestra computadora, esto como requisitos necesarios para su uso.





1. Finalmente, debemos esperar para su descarga y terminamos los requerimientos para su uso
2. Apartir de aqui, bastara con usar nuestro editor de codigo Visual Studio y ejecutar dicho proyecto.

# Conclusión

La lógica difusa permite una representación más completa y precisa de la incertidumbre en la toma de decisiones. En lugar de imponer categorías rígidas, permite expresar grados de pertenencia a esas categorías. Este enfoque es particularmente útil en situaciones donde los límites entre categorías son difusos y la precisión es limitada.

Se utiliza en una variedad de campos, desde sistemas de control hasta inteligencia artificial y toma de decisiones en el mundo real, y permite tomar decisiones más flexibles e inteligentes al incorporar inexactitud e incertidumbre

.

Su creación nos permite destacar sus contribuciones mas importantes como el manejo de la incertidumbre y ambigüedad, flexibilidad en la representación del conocimiento, la toma de decisiones contextualizada, el control de sistemas dinámicos, la creación de sistemas expertos y con soporte a decisiones, la adaptabilidad a diferentes dominios, lidiar con datos incompletos y obtener a cambio sistemas de recomendación personalizados

La lógica difusa desempeña un papel crucial en numerosos campos de la ciencia, la ingeniería y la toma de decisiones en situaciones donde la imprecisión y la incertidumbre son intrínsecas. Su capacidad para modelar la complejidad y la imprecisión en el razonamiento humano, lo que lleva a soluciones más adaptables y contextualmente conscientes en una variedad de aplicaciones y escenarios del mundo real.

# 

# **Referencias**

* Ross T. “Fuzzy Logic with Engineering Applications". <https://home.iitk.ac.in/~avrs/ManyValuedLogic/FuzzyLogicforEngineers.pdf>
* *Enhanced methods of fuzzy logic control*. (1995). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/409700/>
* Hasegawa M. "Fuzzy Logic: A Practical Approach". <http://dl.icdst.org/pdfs/files3/0a49f251ee759262992a990bc66b32ed.pdf>
* A. Zadeh L . (s/n). “Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I & II" . <https://www.dca.fee.unicamp.br/~gomide/courses/EA072/artigos/Fuzzy_Logic_Control_Systems_II_Lee_1990.pdf>
* Singh, H. (2013). Real-Life applications of fuzzy logic. www.academia.edu. <https://www.academia.edu/81914776/Real_Life_Applications_of_Fuzzy_Logic?f_ri=305>
* Fuzzy Control: Principles, practice and perspectives. (1992). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/258603/>
* Bello, E. (2022, 21 junio). Lógica difusa o fuzzy logic: Qué es y cómo funciona + ejemplos. Thinking for Innovation. <https://www.iebschool.com/blog/fuzzy-logic-que-es-big-data/>
* Lógica difusa – acervo para el mejoramiento del aprendizaje de alumnos de ingeniería, en inteligencia artificial. (s. f.). <http://virtual.cuautitlan.unam.mx/intar/?page_id=997>
* Hackeando Tec. (2015, 4 agosto). Lógica difusa - Introducción al curso y aplicaciones - Hackeando Tec [Vídeo]. YouTube. <https://www.youtube.com/watch?v=xLFNUo0mTZE>
* Estudiando Ingeniería. (2020, 27 agosto). Lógica difusa (Fuzzy logic) Parte 1 [Vídeo]. YouTube. <https://www.youtube.com/watch?v=4uyTQtU_JEY>
* Mitchell Paulo Blancas Nuñez. (2020, 21 agosto). Lógica difusa pt1 [Vídeo]. YouTube. <https://www.youtube.com/watch?v=OXZ3XE6-UsE>