Calculadora con Agentes – MESA Python

Arquitectura del Sistema

El sistema fue desarrollado en **Python** usando el framework **MESA**, diseñado para modelar sistemas basados en agentes (ABM).
En este caso, cada **agente** representa una operación matemática o componente funcional de una calculadora.

Componentes principales:

- **Modelo principal:** ModeloCalculadora
- **Agentes de operación:**
    - AgenteSuma
    - AgenteResta
    - AgenteMultiplicacion
    - AgenteDivision
    - AgentePotencia
- **Agentes trigonométricos:**
    - AgenteSeno
    - AgenteCoseno
- **Agente de control:**
    - AgenteEntradaSalida → encargado de recibir la expresión del usuario, interpretarla y coordinar el uso de los demás agentes.

El modelo usa un scheduler aleatorio (mesa.time.RandomActivation), aunque en este caso las operaciones no dependen del orden de activación, sino de las llamadas directas desde el agente de entrada.

Interacción entre los Agentes

El agente **AgenteEntradaSalida** actúa como un **agente coordinador o mediador** dentro del sistema.
Cuando el usuario ingresa una expresión (por ejemplo, 6 * 4), este agente:

1. **Recibe la cadena** y elimina espacios.

2. **Identifica el tipo de operación** (por ejemplo, * para multiplicación).
3. **Convierte los operandos** en números flotantes.
4. **Envía la solicitud** al agente correspondiente:
    - self.model.agente_multiplicacion.operar(a, b)

El agente especializado (por ejemplo, AgenteMultiplicacion) **procesa la operación** en su propio método operar() y **retorna el resultado** al agente de entrada, que finalmente lo muestra al usuario.

Ejemplo:

Interacción:

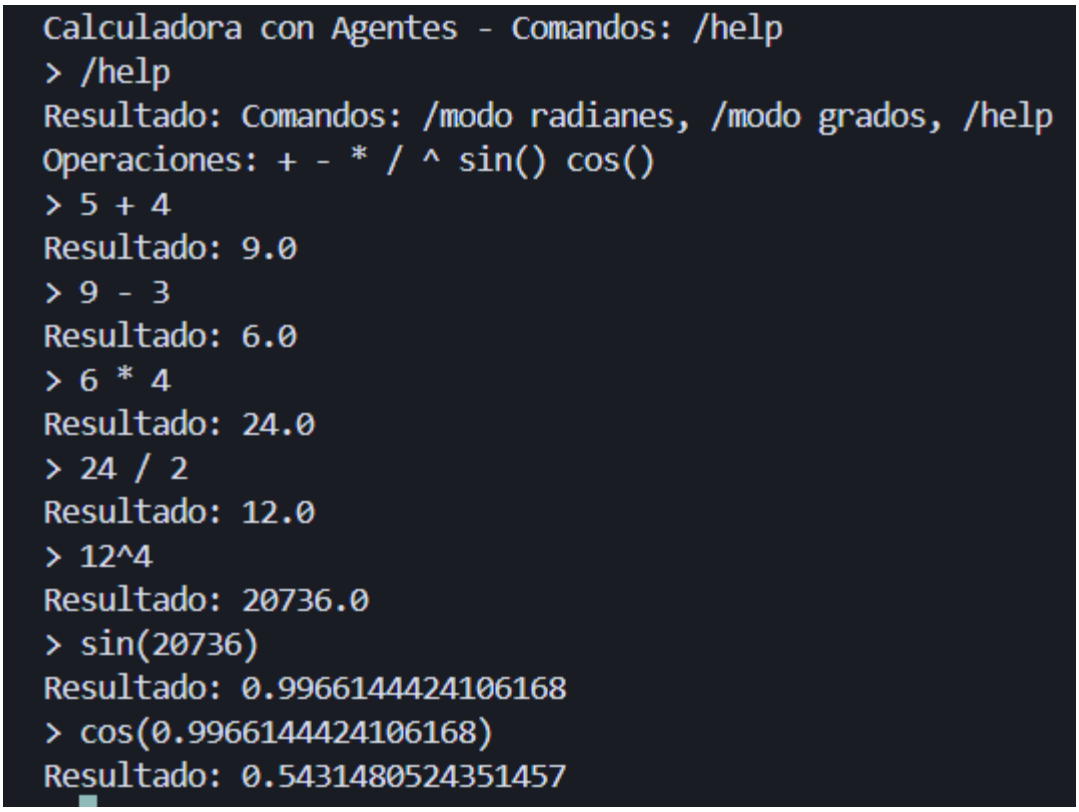
1. El usuario ingresa 5 + 4.
2. AgenteEntradaSalida detecta el operador +.
3. Llama al método operar(5, 4) del AgenteSuma.
4. AgenteSuma devuelve 9.0.
5. El resultado se imprime como Resultado: 9.0.

Mecanismos de Comunicación

El sistema utiliza **comunicación directa mediante llamadas de métodos** entre los agentes, no intercambio de mensajes asíncronos.

- La comunicación se da en **una sola dirección**:
  AgenteEntradaSalida → AgenteOperación → AgenteEntradaSalida
- No existe un entorno de comunicación global ni buffers; cada llamada es sincrónica y se resuelve inmediatamente.
- Los agentes **no almacenan estados complejos**, salvo los agentes trigonométricos, que mantienen la variable modo (grados o radianes), la cual puede modificarse mediante comandos (/modo grados, /modo radianes).

Capturas de Ejemplo



Cada interacción corresponde a un mensaje de cálculo que el agente de entrada dirige al agente apropiado, mostrando cómo el sistema distribuye las responsabilidades.

Conclusiones

Este sistema implementa una **arquitectura modular basada en agentes cooperativos**, donde cada agente tiene una responsabilidad única:

- **Descomposición funcional clara**: cada operación matemática se maneja de forma independiente.
- **Escalabilidad**: se pueden agregar fácilmente nuevos agentes (por ejemplo, AgenteTangente, AgenteLogaritmo).
- **Interacción fluida**: la comunicación entre agentes es directa, eficiente y de bajo acoplamiento.

El resultado es una calculadora distribuida que demuestra los principios de los modelos multiagente, aplicados a un problema clásico de forma didáctica