

# Intelligent Agents Applied to E-Commerce: A Comparative Study of AutoGen and Semantic Kernel



## Colaboración

Alba Beth-birai López Aguilar; Jesús Eduardo Ceballos Contreras; Walter Alexander Mata López; Jesús Alberto Verduzco Ramírez; Víctor Hugo Topete Castillo, Universidad de Colima

Fecha de recepción: 25 de agosto 2025

Fecha de aceptación: 13 de septiembre de 2025

**RESUMEN:** La inteligencia artificial (IA) ha transformado significativamente el comercio electrónico, permitiendo la automatización de procesos y la optimización de la toma de decisiones. Dentro de este contexto, los Modelos de Lenguaje de Gran Escala (LLM) han facilitado la interacción con los usuarios mediante Procesamiento de Lenguaje Natural (PLN), mejorando la comunicación en plataformas digitales.

Este artículo explora el uso de agentes inteligentes en la negociación electrónica, destacando su capacidad para automatizar interacciones y mejorar la eficiencia en la toma de decisiones comerciales. Se analizan dos herramientas desarrolladas por Microsoft: Semantic Kernel y AutoGen, ambas diseñadas para facilitar la creación y gestión de agentes inteligentes en entornos de comercio digital.

La investigación compara el desempeño de ambas herramientas en aspectos como personalización, facilidad de implementación y optimización. AutoGen se destaca en la gestión de sistemas multi-agentes y toma de decisiones automatizada, mientras que Semantic Kernel sobresale en procesamiento del lenguaje natural y personalización de interacciones.

**PALABRAS CLAVE:** AutoGen, Semantic kernel, Agentes inteligentes, E-commerce, prompt Tuning, Procesamiento del Lenguaje Natural.

**ABSTRACT:** Artificial Intelligence (AI) has significantly transformed e-commerce by enabling process automation and optimizing decision-making. Within this context, Large Language Models (LLMs) have facilitated user interaction through Natural Language Processing (NLP), improving communication on digital platforms.

This article explores the use of intelligent agents in electronic negotiation, highlighting their ability to automate interactions and improve the efficiency of business decision-making. It analyzes two tools developed by Microsoft: Semantic Kernel and AutoGen, both designed to support the creation and management of intelligent agents in digital commerce environments.

The research compares the performance of both tools in aspects such as

personalization, ease of implementation, and optimization. AutoGen excels in managing multi-agent systems and automated decision-making, while Semantic Kernel stands out in natural language processing and personalization of interactions.

**KEYWORDS:** AutoGen, Semantic kernel, Intelligent Agents, E-commerce, prompt Tuning, Natural Language Processing.

## INTRODUCCIÓN

La Inteligencia artificial (IA) es considerada como una llave hacia un nuevo mundo tecnológico y avanza a grandes pasos generando impacto significativo en diversas áreas. Uno de los muchos sectores donde la IA ha tenido mayor influencia es el comercio electrónico, optimizando la automatización de pro-

cesos, la toma de decisiones y la personalización de preferencias.

En este contexto, la aparición de la IA generativa crea una nueva visión de la manera en la que se crea contenido, al combinarse con los Modelos de Lenguaje de Gran Escala (LLM, por sus siglas en inglés). Estos modelos permiten la creación automatizada de contenido textual, gráfico, sonoro y audiovisual con alta calidad [7] (Franganillo, 2023). Un ejemplo de IA Generativas incluye CHATGPT, Gemini, Copilot y Deepseek, estas IA han transformado la forma en la que los usuarios se comunican y buscan información de manera rápida y sencilla.

Este avance no sería posible sin el Procesamiento de Lenguaje Natural (PLN) que facilita la interacción entre los modelos de lenguaje y los usuarios a través de los prompts, gracias a esta tecnología, las IA Generativas pueden comprender, procesar y responder solicitudes de múltiples usuarios, mejorando eficiencia y accesibilidad de búsqueda de información.

Por su parte, el desarrollo de agentes inteligentes no se limita únicamente a los modelos generativos. Existen diversas tecnologías como API (Un API funciona como un puente que permite que diferentes programas o servicios se comuniquen entre sí, facilitando el acceso a funcionalidades externas sin necesidad de desarrollar todo desde cero), kernel y librerías que permiten crear agentes de inteligencia artificial sin sobrecargar nuestro ordenador computacional. Entre las herramientas más destacadas en este campo se encuentran Semantic kernel de Microsoft y AutoGen, ambas desarrollada por Microsoft. Semantic Kernel facilita la integración de modelos de lenguaje con flujos de trabajo personalizados mediante la orquestación de prompts y la combinación de distintas habilidades de IA. Por su parte, AUTOGEN permite la creación de agentes autónomos capaces de colaborar con modelos de lenguaje más estructurada y eficiente (Microsoft, 2023).

En proyectos anteriores, se ha desarrollado un modelo de agentes inteligentes enfocados al comercio electrónico enfocada en la compra y renta de inmuebles dentro de la organización de eventos [11] (López Aguilar, 2024). Para la realización de dicho modelo se utilizó la API de Microsoft AUTOGEN (Microsoft, 2023), una API diseñada para crear sistemas de múltiples agentes permitiendo la generación automática de contenido y administración. Sin embargo, aunque esta herramienta ha generado logros excepcionales para el propósito ideal, nace la interesante interrogativa dónde probemos distintos métodos para poder optimizar los procesos en las negociaciones y las tomas de decisiones dentro de estas mismas.

## Antecedentes de los agentes inteligentes en el e-commerce

En años recientes, los agentes inteligentes han evolu-

cionado magnánimamente gracias a los modelos de lenguaje a gran escala (LLM), una de las contribuciones más relevantes en esta línea de investigaciones es el trabajo de [4] Cheng et al. (2024) quienes ofrecen una visión integral sobre los agentes basados en modelos, este estudio destaca agentes que no solo operan de forma autónoma, sino que también permiten la construcción de sistemas multi-agentes con capacidades avanzadas de planificación, razonamiento cognitivo y el uso de herramientas externas. Los autores señalan que este tipo de arquitectura es fundamental para aplicaciones con entornos dinámicos, que es el caso del comercio electrónico, donde existe la personalización, el análisis contextual y toma de decisiones esenciales. El enfoque descrito en el manuscrito refuerza la importancia de utilizar herramientas tecnológicas como AutoGen y Semantic Kernel, las cuales aprovechan la capacidad de los LLM para interactuar y adaptarse al comportamiento del usuario.

Según [5] Elizondo, E. & Salazar (2019) los agentes inteligentes son sistemas autónomos que perciben su entorno mediante sensores, que procesan la información utilizando técnicas de inteligencia artificial actuando en un entorno diseñado para alcanzar objetivos predefinidos. Estos agentes permiten cumplir objetivos con interacciones automáticas sin intervención humana.

Existen diversas clasificaciones de agentes inteligentes según su funcionalidad y uso dentro de la industria de la tecnología, de las cuales se encuentran:

- Agentes colaborativos
- Agentes de Interfaz
- Agentes de la información
- Agentes Reactivos
- Agentes Sociales
- Agentes Móviles
- Agentes Híbridos

Para la implementación de este proyecto, se utilizan principalmente los agentes colaborativos y de la información, lo cual se alinea con el enfoque en el comercio digital.

Por ejemplo, [12]Matta Catora (2012) desarrolló un sistema web para gestionar ventas y facturación electrónica, en la cual los agentes inteligentes optimizan el manejo de información, la aplicación demostró el valor de los agentes en procesos empresariales que requieren manejo de datos precisos y toman decisiones de manera autónoma. De una manera muy similar a nuestro modelo, el agente vendedor gestiona transacciones e interacción con el cliente.

## Procesamiento de lenguaje natural en agentes inteligentes

El PLN es una rama de la inteligencia artificial que permite a las máquinas comprender, procesar y generar lenguaje humano. Los modelos de lenguaje como los

utilizados en Semantic Kernel emplean técnicas avanzadas de PLN para interpretar las intenciones del usuario y generar respuestas coherentes [10] (Jurafsky & Martin, 2022). Aplicación de PLN en comercio electrónico:

- Chatbots inteligentes que mejoran la atención al cliente.
- Análisis de sentimiento para evaluar la satisfacción del usuario.
- Automatización de respuestas en negociaciones electrónicas.

En este artículo se abordará el tema con más detalle mediante la comparativa de AutoGen y Semantic Kernel, evaluando las funcionalidades, ventajas y limitaciones en el desarrollo de agentes inteligentes dentro de nuestro campo objetivo; el comercio electrónico.

### Negociación electrónica y agentes inteligentes

La negociación electrónica es el proceso automatizado en el que agentes inteligentes representan a compradores y vendedores para alcanzar acuerdos óptimos en transacciones digitales [1] (Baarslag et al., 2016). Dichos agentes utilizan algoritmos de toma de decisiones combinado con técnicas de PLN para poder analizar propuestas, generar respuestas y mejorar la eficiencia en la negociación.

Según el contexto dado, los agentes pueden:

- Automatizar la comunicación entre clientes y proveedores.
- Personalizar ofertas y recomendaciones en tiempo real.
- Ajustar estrategias de precios dinámicamente según la demanda del comercio.

Un ejemplo de uso de este tipo de agentes es la plataforma comercial de Amazon, la cual usa IA para ajustar dinámicamente los precios según la competencia.

### Negociación electrónica con autogen y semantic kernel

La implementación de agentes inteligentes en la negociación electrónica es clave para optimizar la automatización en interacciones y la toma de decisiones, es así como Microsoft AutoGen y Semantic kernel pueden desempeñar roles específicos que potencian estos sistemas.

#### AutoGen en la negociación

- Creación de agentes multi-agentes que simulan el proceso de negociación.
- Automatización de respuestas y ajustes en ofertas en tiempo real
- Integración con modelos de lenguaje para evaluar intenciones del mismo usuario.

Tal es el caso del modelo de agentes inteligentes que simulan la interacción y negociación en un sector de organización de eventos, donde uno de los agentes toma el papel del usuario y el agente compañero es el agente

vendedor [11] (López Aguilar, 2024), así se genera una negociación y una contraoferta automática según las preferencias que maximice la ganancia.

#### Semantic kernel

Con el SDK, podemos encontrar las funcionalidades para las negociaciones electrónicas como las siguientes:

- Procesamiento del lenguaje Natural para poder analizar el tono y contexto de las ofertas.
- Personalización de respuestas según perfil del cliente y datos previos.
- Integración con herramientas externas para optimizar estrategias de negociación.

Tal es que un sistema basado en Semantic Kernel puede analizar los patrones de compra de un usuario y generar recomendaciones personalizadas antes de iniciar la negociación.

### Comparativa entre semantic kernel y autogen

Con el objetivo de optimizar las negociaciones electrónicas, este estudio se enfoca en la comparativa de dos herramientas claves y especializadas en la creación de sistemas de agentes inteligentes; Semantic kernel y AutoGen, ambas desarrolladas por Microsoft.

#### Semantic Kernel

Semantic Kernel es un SDK de código abierto que permite integrar grandes modelos lingüísticos (LLM) en su propio código, con el kernel se puede crear aplicaciones inteligentes que comprendan y respondan mensajes de lenguaje natural (Microsoft, n.d.).

Sus principales funcionalidades son:

- Comprender el lenguaje natural.
  - Orquestación de prompts para gestionar tareas de IA.
  - Proporciona recomendaciones de alta personalización.
- Este SDK es ideal para aplicaciones de IA conversacional y análisis de texto, brindando flexibilidad en la creación de agentes que su principal funcionalidad es la optimización de comunicación con el usuario.

#### AutoGen

Por otra parte, el API de AUTOGEN, ofrece un marco de programación de código abierto, hecho para construir agentes de IA facilitando la comparación de múltiples de agentes entre sí, algunas de sus características clave son:

- Permitir crear aplicaciones LLM con mínimo esfuerzo
- Tiene alta compatibilidad para conversaciones de diversos patrones para flujos de trabajo complejos.
- Proporciona una colección de sistemas de trabajo con diferentes complejidades.

Siendo así que AutoGen es el más adecuado para sistemas multi-agentes que requieren colaboración estructurada y toma de decisiones automatizada.



Tabla 1. Comparativa AutoGen y Semantic Kernel.

CRITERIO	MICROSOFT AUTOGEN	MICROSOFT SEMANTIC KERNEL
<b>Enfoque principal</b>	Multi-agentes y automatización de interacciones.	Integración de LLMs
<b>Capacidad de personalización</b>	Configuración limitada de agentes	Personalización alta de respuestas y funciones.
<b>Uso de comercio electrónico</b>	Recomendado para automatizar negociaciones e interacciones.	Útil para análisis de texto y personalización
<b>Facilidad de implementación</b>	Requiere configuración de agentes	Sencillo para integraciones de IA conversacional

Fuente: Elaboración propia 2025.

Con la comparación de la Tabla 1, se busca determinar cuál de estas herramientas optimiza mejor los procesos de negociación en el comercio electrónico, mejorando la eficiencia y personalización de las interacciones automatizadas.

## MATERIALES Y MÉTODOS

### Pruebas experimentales

En el trabajo "Design and implementation of a Two-Agent Prototype Applied to E-Commerce and Goods and/or Services" [11] (López Aguilar, 2024) se establece la problemática de automatización en negociaciones entre dos agentes inteligentes, se declara en esa investigación que la problemática general que se deseaba abordar era la ineficiencia en las negociaciones entorno a organizaciones de eventos, es así que plantean un modelo de dos agentes inteligentes el cual uno de ellos aborda el papel del usuario y el otro acoge el papel del comerciante, automatizando las negociaciones y reduciendo errores humanos.

La implementación de dicho modelo se realizó en ayuda del API de AutoGen, creando un entorno específico para el modelado del agente vendedor (quién toma el papel del comerciante) y comprador (el usuario) según lo mencionan, aunque los resultados de dicho artículo dicen "muestran que el agente vendedor es capaz de gestionar múltiples interacciones de forma eficiente" si bien los resultados de dicha investigación fueron bastante conclusivos, se intenta en esta investigación el poder que otro tipo de tecnologías expertas en agentes podría hacer en el mismo entorno de comercio electrónico, específicamente en organización de eventos.

Para poder llevar a cabo una comparativa completa respecto a las dos herramientas de creación de agentes inteligentes, se implementó un prototipo donde se pone en uso el SDK de Semantic en un entorno de comercio electrónico.

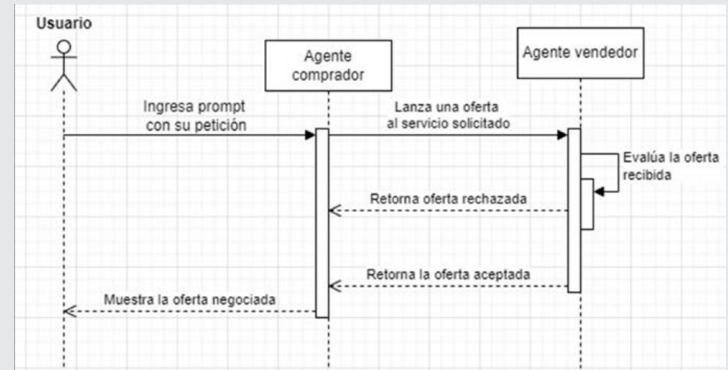


Figura 1. Diagrama de secuencia del sistema con AutoGen (Fuente: Alba & Ceballos, 2024).

### Arquitectura de los agentes

El diseño imperativo de ambos agentes fue el mismo para ambas puertas de comparación, el enfoque permitió tener mayor flexibilidad y escalabilidad, la arquitectura de ambos agentes (Vendedor y Comprador), mantienen una estructura casi similar, pero con mecanismos internos de tomas de decisiones diferentes.

#### Agente Comprador

Este agente representa al usuario, recopilando preferencias, restricciones y objetivos de la negociación a optimizar.

- Módulo de entrada de criterios.
- Módulo de decisión.
- Módulo de comunicación.

#### Agente vendedor

Simula el comportamiento de un proveedor y comerciante, se encarga de ofertar propuestas en base a disponibilidad, precios y condiciones, según lo solicitado por su compañero.

- Base de datos.
- Módulo de negociación.
- Adaptador de contexto (uso de PLN).

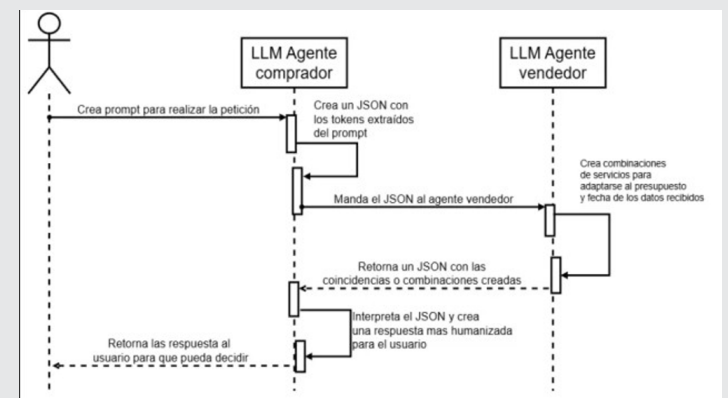


Figura 2. Diagrama de comunicación de los agentes con Semantic Kernel. (Fuente: propia, 2025)

### Resultados de las pruebas experimentales

Se aplicaron un conjunto de métricas para poder determinar la factibilidad de cada uno de estos creadores de agentes inteligentes, con el fin de poder determinar las características, fortalezas, limitaciones y optimizaciones de cada uno de estos modelos de agentes inteligentes, las métricas aplicadas fueron:

#### Rapidez de respuesta

Aunque para ambos modelos los requerimientos fueron los mismos, para la aplicación de Semantic Kernel hubo una respuesta negativa para la métrica de rapidez, contando con un tiempo de respuesta de 5 a 10 minutos con el modelo ya afinado, sin embargo, el problema no recae en el modelo, la ineficiencia en tiempo de respuesta se debe a las limitaciones computacionales de los equipos utilizados, ya que los modelos que se utilizan para las pruebas son altamente pesados y requieren recursos computacionales con capacidad para cómputo intensivo.

Sin embargo, cabe mencionar que, si se contara con un cómputo idóneo, las respuestas serían más rápidas y precisas.

Sorprendentemente con AutoGen, los resultados con este servicio fueron satisfactorias, consta con un tiempo de respuesta de 3 segundos generando un conjunto de respuestas precisas bastante acercadas al propósito inicial, esto debido a que AutoGen no utiliza modelos de lenguajes que es el caso contrario con Semantic, con ayuda de procesamiento de lenguaje natural se extraen los tokens de los prompts, para posterior AutoGen crear los agentes, logrando encontrar comparaciones exitosas respecto a lo solicitado.

#### Consumo de recursos

Como se mencionó anteriormente, la ejecución de estos tipos de modelos de lenguajes especializados en poder automatizar negociaciones o procesos son enormemente pesados, complicando la compilación en dispositivos limitados en recursos de GPU, CPU y memoria RAM. Las pruebas realizadas se implementaron en equipos con unas especificaciones básicas, contando con:

- Procesador: AMD RYZEN 5 5500U con Radeon Graphic
- RAM: 16.0GB MHz 3200
- SO: Windows 11.

Con las especificaciones anteriores, y el uso de Semantic Kernel, los resultados llegan a ser difíciles de obtener, más no imposibles, la gran falta de cómputo de alto nivel nos otorga una vista algo limitada en base a lo esperado, por ejemplo, ejecutando el modelo Mistral, se usan un total de 4GB de RAM generando que el tiempo de respuesta sea lento y poco

preciso, esto en base a los prompts ingresados para su análisis posterior, calificando como una respuesta lenta y poco efectiva. Se concluye con la premisa de "A mayor capacidad de equipo, mayor precisión de respuesta y optimización".

Para AutoGen, partiendo con las especificaciones mencionadas anteriormente, AutoGen nos proporciona un panorama más sencillo y menos complejo en base al requerimiento de consumo de recursos, AutoGen no impone demasiada demanda siendo casi imperceptible, podría considerarse el uso de AutoGen como una librería más en nuestras líneas de código, siendo sencillo y amigable a los desarrolladores.

#### Escalabilidad

Durante la incesante pregunta respecto a la decisión sobre que herramienta sería de uso excelente para el trabajo, la curiosidad por encontrar herramientas que nos permitieran tener la oportunidad de escalar las ideas a algo más grande persistían. La escalabilidad se define como la habilidad de mantener similar o mayor el desempeño en un modelo específico [3] (Cavazos-Arroyo & Giuliani, 2017), aunque, en la informática, la escalabilidad se basa en la capacidad de un sistema de admitir cualquier incremento en demandas de recursos y capacidades de procesamiento [6] (Escalabilidad, 2025). Es así, que las comparativas nos arrojan de manera panorámicas respuestas para poder encontrar la herramienta necesaria para el cumplimiento del objetivo.

Semantic Kernel se considera de escalabilidad media-alta. Esto debido a que no está enfocado al manejo de múltiples agentes (por ello considerada media), sin embargo, es altamente escalable en integración de flujos de trabajo y orquestación de habilidades.

En aspecto de capacidad funcional, es decir, nuevas habilidades; es altamente considerable, esto referente a que permite que un mismo agente maneje grandes cantidades de habilidades o plugin. En el sentido de orquestación de tareas complejas; su gran capacidad de combinar diferentes prompts y funciones de IA se logra generar flujos de conversaciones extensos pudiendo aumentar la complejidad técnica del desarrollador.

Por otro lado, AutoGen promete una escalabilidad algo más prometedora, tomando como referencia a los mismos aspectos dichos anteriormente, referente a la escalabilidad enfocada al uso de multi-agentes es bastante alta, esto ya que pueden gestionar múltiples agentes que colaboran y compiten entre sí, más, sin embargo, en base a la funcionalidad de nuevas habilidades, es considerada media, ya que el enfoque es más hacia el lado de escalar operaciones y automatizar interacciones.

Complejidad de configuración y compatibilidad con llms

La complejidad se mide en respecto a que tanto conoces el ámbito de programación de un sistema, sin embargo, es necesario tener nociones algo amplias respecto a Python, dado que es el lenguaje en el que los agentes están siendo programados y configurados. Para Semantic Kernel, cabe mencionar que su documentación original enfocado con lenguaje C# (por lo que el conocimiento en .Net es fundamental). Dado que esta tecnología es basada en modelos de lenguaje, tiene la compatibilidad de manejar modelos como OpenAI, Ollama, onnx y anthropic, etc.

Es necesario comentar que, en este caso, para la implementación dentro del ambiente de un comercio electrónico fue necesario configurar o pulir con fine-tuning. El fine-tuning permite adaptar conocimientos que los modelos de lenguajes grandes ya poseen (responder preguntas, generar texto o clasificar texto), con el fin de mejorar su capacidad para abordar tareas más concretas y especializadas [15] (Travieso García, Á. J. 2024). Si bien, con este proceso el modelo arrojó respuestas aceptables, se debe recordar que el ambiente es un comercio de organización de eventos, por lo que la interacción con el modelo es mediante prompts, por lo que se utilizó una técnica derivada del fine-tuning llamada "Prompt Tuning", esta técnica se utiliza para ajustar y optimizar los prompts, personalizándolo y ajustándolo a la tarea específica en cuestión [14] (Saavedra, 2023) implica un ajuste más fino y ligero con respecto con otras técnicas de finura, realizando solo una modificación en la manera en la que se presentan las solicitudes, permitiendo que el modelo se adapte a diferentes contextos o tareas sin la necesidad de cambiar estructuras internas.

AutoGen, en esencia, es utilizada mediante agregarla a la cabecera de tu código como una librería, específicamente una superclase, siendo sencillo poder configurarlo como un agente que es controlado o controla gestiones y adaptaciones de procesos.

Dado que AutoGen es utilizada mediante superclases, no actúa precisamente con modelos de lenguaje, sin embargo, AutoGen cuenta con una librería que combina Semantic Kernel con AutoGen, pudiendo extraer los modelos de lenguaje y gestionarlos a gusto propios.

Documentación disponible

Semantic fue lanzado públicamente el 17 de marzo del 2023, se considera que es una tecnología de nuevo comienzo, por lo que la mayor parte del tiempo se encuentra en actualización, actualmente se hayan pequeños ejemplos de uso, pero nada extravagante, dando carta abierta a poder explotar el potencial se Semantic, para el uso en esta ocasión tomamos la versión más estable para trabajar. Por otra parte, Au-

toGen lleva en el mercado más tiempo que Semantic, por lo que la información para su implementación es basta y amplia, además que existen dentro de la documentación de AutoGen, distintos cursos que refuerzan el uso de los multi-agentes.

Optimización

A este punto, según la RAE (2024), optimizar es el acto de mejorar algo al máximo posible. En contexto con el proyecto, se refiere a la capacidad de las herramientas tecnológicas para maximizar la eficiencia en la comunicación, minimizar los recursos computacionales utilizados y agilizar el proceso de negociación electrónica entre los agentes inteligentes. En este sentido, Semantic Kernel muestra una alta capacidad de optimización en tareas relacionadas con el procesamiento del lenguaje natural, es así que la generación de respuestas humanizadas, y la ejecución de flujos conversacionales complejos. Sin embargo, como se había explicado antes, esto requiere de mayores recursos computacionales.

Por otro lado, AutoGen presenta una optimización sobresaliente en cuanto a eficiencia y facilidad de implementación, su arquitectura ligera ha permitido integrar en sistemas con bajos recursos sin comprometer la funcionalidad, logrando negociaciones efectivas con configuraciones mínimas.

Tabla 2: comparativa de los resultados.

Aspecto que comparar	SEMANTIC KERNEL	AUTOGEN
Rapidez de respuesta	Lenta por limitaciones computacionales	Rápida.
Consumo de recursos	Excesivo.	Bajo.
Escalabilidad	Media-alta.	Media.
Complejidad de configuración y compatibilidad	Baja con conocimientos básicos de programación	
Documentación disponible	Escasa.	Bastante.
Optimización	Muy bueno, pero con limitaciones.	Muy bueno.

Fuente: Elaboración propia 2025.

Pruebas funcionales

Aunque los resultados obtenidos nos permitan observar diferencias en cuanto a eficiencia y rapidez de optimización entre AutoGen y Semantic Kernel, no se ha realizado pruebas en un experimento sistemático de estrés que permita evaluar el comportamiento de ambas tecnologías bajo condiciones de alta carga.

El objetivo principal es evaluar el comportamiento de AutoGen y Semantic Kernel bajo condiciones de carga elevada, simulando un entorno de uso realista e intenso en el comercio electrónico. Para evaluar este



desempeño, la prueba se realizó utilizando un equipo con las siguientes especificaciones técnicas:

- Procesador: AMD Ryzen 7 7840HS (8 núcleos, 16 hilos)
- Memoria RAM: 32 GB DDR5 a 5600MH
- Almacenamiento: SSD NVMe de 1TB con velocidad de lectura de 7200 MB/s
- Tarjeta gráfica: Radeon 780M (no utilizada directamente)

Los modelos de lenguaje fueron ejecutados directamente en CPU, sin aceleración por GPU. El objetivo se evaluó mediante la rapidez de respuesta, uso de CPU/ RAM y tasa de éxito bajo diferentes niveles de carga concurrente (10,50,100,200 y 500 usuarios simulados.)

Métricas evaluadas:

- Tiempo de respuesta
- Uso de CPU (%)
- Uso de RAM
- Tasa de éxito de las respuestas correctas (%)

Tabla 3. Tabla de resultados.

Carga recurrente	Tecnología	Tiempo de respuesta (ms).	Uso de CPU %	Uso de RAM %	Tasa %
10 usuarios	AutoGen	65	21%	380	100%
	Semantic Kernel	55	26%	420	100%
50 usuarios	AutoGen	120	38%	470	99%
	Semantic Kernel	110	45%	560	98%
100 usuarios	AutoGen	210	56%	570	97%
	Semantic Kernel	200	64%	670	95%
200 usuarios.	AutoGen	350	72%	700	93%
	Semantic Kernel	380	78%	850	90%
500 usuarios	AutoGen	510	86%	890	88%
	Semantic Kernel	550	93%	1100	83%

Fuente: Elaboración propia 2025.

Semantic demostró mejor repuesta a bajas cargas, con mayor capacidad de análisis semántico, aunque su consumo de RAM fue elevado a medida que aumentaban los usuarios.

AutoGen mantuvo una respuesta estable y eficiente en cargas medianas y altas, mostrando mayor tolerancia en entornos con menor recursos. Ambos sistemas funcionaron bien hasta con 100 usuarios concurrentes, donde mantuvieron tasas de éxito superiores al 95%.

Con la finalidad de reforzar la metodología, se incorporó un proceso de validación que garantiza la confiabilidad de los resultados obtenidos en las pruebas comparativas entre AutoGen y Semantic kernel, para ello, cada escenario de carga (10,50,100,200 y 500 usuarios) fue ejecutado en cinco corridas independientes, lo que permitió evaluar la repetibilidad de los experimentos. En cada métrica, desde tiempo de respuesta, uso de CPU y uso de memoria RAM, así como la tasa de éxito en las negociaciones, se calcularon valores de media, desviación, y error relativo, asegurando estadísticas de los datos. Así mismo, se realizó una prueba de estrés progresiva con incremento gradual de usuarios recurrentes hasta identificar el punto de degradación de cada una de las herramientas. Por conclusión, se establece que ambas tecnologías fueron sometidas a las mismas condiciones experimentales, lo que permitió establecer una comparación justa y confiable de los desempeños individuales.

RESULTADOS

Tras los resultados de las métricas, la percepción respecto a cuál de estas tecnologías de creación y configuración enfocadas en agentes inteligentes fuera idónea para un ambiente de comercio electrónico, las disyuntivas fueron excepcionales. Por un lado, utilizar AutoGen que es una manera fiable y amigable para un entorno con recursos limitados hablando computacionalmente, debido que su implementación es más sencilla pudiendo solo añadiéndolo como una superclase que ayuda a la realización de actividades, esta accesibilidad hace que AutoGen una herramienta útil para equipos de desarrollo pequeño, pruebas de conceptos o prototipos funcionales que buscan automatizar tareas específicas dentro de un flujo de negociación o recomendación.

Por el contrario, Semantic se vuelve más demandante por razones específicas, el uso de modelos de lenguajes son bastantes pesados, sin embargo, demostró un mayor potencial en términos de personalización, comprensión contextual y flexibilidad, aunque el uso de grandes modelos de lenguajes ayuda a comprender, analizar e incluso anticipar intenciones del usuario este nivel de sofisticación conlleva un nivel de aprendizaje profundo y con gran necesidad de optimización de rendimiento. Semantic sobresale en escenarios donde la calidad de lenguaje y respuesta personalizada es primordial.

Entonces, se concluye qué; AutoGen funciona de manera más eficaz en esquemas donde se requiere una toma de decisiones estructurada, fluida y colaborativa entre agentes autónomos. Mientras que Semantic so-

bresale en escenarios donde la calidad de lenguaje y respuesta personalizada es primordial.

La elección de una y otra herramienta depende en gran magnitud en el contexto del proyecto, los recursos y objetivos específicos que se le quieran otorgar a los agentes.

## CONCLUSIONES

Esta comparativa no se trata sobre cual herramienta es superior a la otra, sino de que soluciones responden a necesidades diferentes, el análisis refuerza la idea de la integración correcta y estratégica de herramientas adecuadas en el desarrollo de agentes inteligentes que marcan la diferencia significativa en conceptos de escalabilidad, eficiencia, experiencia de usuario y facilidad de implementación [13] (Russel & Norving, 2021; Hernández y Martínez, 2024) y uso dentro del marco del comercio electrónico.

Los hallazgos obtenidos en este estudio tienen una aplicación directa en plataformas de e-commerce que buscan optimizar procesos de negociación, personalización y toma de decisiones. En particular, el uso de Semantic Kernel resulta adecuado para escenarios donde la prioridad de la interacción conversacional avanzada con los usuarios, como en chatbots de atención al cliente o sistemas de recomendación en tiempo real. Por otro lado, AutoGen se muestra especialmente útil en entornos donde es necesario coordinar múltiples agentes colaborativos, como es el caso de la negociación automática entre proveedores, clientes y plataformas intermediarias, reduciendo tiempos y costos operativos.

En contexto más prácticos, estos resultados sugieren que las empresas de comercio electrónico pueden integrar ambas tecnologías de manera complementaria: Semantic kernel para mejorar la experiencia del usuario final y AutoGen para la integración de gestión automatizada de negociaciones complejas y procesos internos, abriendo la posibilidad de desarrollar soluciones híbridas que no solo incrementen la eficiencia técnica, si no también generar ventajas competitivas en entornos de producto digitales.

En las menciones de trabajos recientes, aborda el éxito de los agentes inteligentes no solo depende de la arquitectura misma de ellos, sino de su adecuada alineación con las metas específicas del sistema [10] (Jurafsky & Martin, 2022). En futuras investigaciones, se propone aplicar estas herramientas en entorno reales y medir su impacto en términos de rendimiento de satisfacción del usuario y retorno de inversión.

## BIBLIOGRAFÍA

[1] Aceituno-Madera, P. et al (2011). *Melanoma, altitud y radiación uvb*. *Actas dermo-sifiliograficas*, 102(3): 199-205.

[2] Acosta, E., Fierro, et al. (2009). *Melanoma: patogénesis, clínica e histopatología*. Consultada el 13 del 09 del 2023.

[3] Ali, M., Tang, et al. (2020). *Algoritmo semantic-k-nn: una versión mejorada del algoritmo k-nn tradicional*. ISSN: 0957-4174, 151.

[4] Arias, V. et al. (2019). *Una introducción a las aplicaciones de la inteligencia artificial en medicina: Aspectos históricos*. *Revista Latinoamericana de Hipertensión*, 14(5):590-600.

[5] Artola Moreno Á. (2019). *Clasificación de imágenes usando redes neuronales convolucionales en Python*. pages 28-29.

[6] Barrios Tao, H. et al. (2020). *Subjetividades e inteligencia artificial: desafíos para 'lo humano'*. *Veritas*, (47):81-107.

[7] Bernal-de Lazaro, J. et al. (2011). *Estudio comparativo de clasificadores empleados en el diagnóstico de fallos de sistemas industriales*. *Ingeniería Mecánica*, 14(2):87-98.

[8] Bonilla Carrión, C. (2020). *Redes convolucionales*. page 16.

[9] Duran Suarez, J. (2017). *Redes neuronales convolucionales en r: Reconocimiento de caracteres escritos a mano*. page 25.

[10] Hardy, T. (2001). *la (inteligencia artificial)*. *Polis: Revista Latinoamericana*, (2):18.

[11] Herrera, J. C. a. et al. (2016). *Diseño de un sistema automático de selección de frutos de café mediante técnicas de visión artificial*. *Revista UIS ingenierías*, 15(1):7-14.

[12] Infante Carbonell, M. G. et al. (2019). *Melanoma cutáneo: algunas consideraciones actuales*. *Medisan*, 23(1):146-164.

[13] Leitner, R. (2006). *Epidemiología del melanoma cutáneo*. *Revista argentina de dermatología*, 87(2):86-97.

[14] Lucero Carrillo, E. D. et al. (2016). *Utilización de técnicas de visión artificial para la detección automática de defectos externos del mango*.

[15] Lugo-Reyes, Saúl Oswaldo, a. o. (2014). *Inteligencia artificial para asistir el diagnóstico clínico en medicina*. *Revista Alergia México*, 61(2):110-120.

[16] Maria Zuluaga, A. I. et al. (2016). *Actualización en el tratamiento quirúrgico del melanoma cutá-*



neo primario y metastásico. *Cirugía y Cirujanos*, 84(1):77–84.

[17] Micó, M. (1996). Algoritmos de búsqueda de vecinos más próximos en espacios métricos. PhD thesis, Tesis doctoral, Universidad Politécnica de Valencia, Valencia, España.

[18] M.L. Villalobos León, R. M. V. et al. (2013). Melanoma. *Medicine-Programa de Formación Médica Continuada Acreditado*, 11(26):1597–1607.

[19] Mordoh, A. (2009). clínica del melanoma. *Acta bio-química clínica latinoamericana*, 43(3):327–331.

[20] Narváez Vilema, M. E. et al. (2021). Predicción de clientes potenciales utilizando k vecino más cercano en el área de negocios de la cooperativa Riobamba.

[21] Nguyen Cong, R. P. et al. (2015). Aprendizaje super-visado de funciones de distancia: estado del arte. *Revista Cubana de Ciencias Informáticas*, 9(2):14–28.

[22] Porcelli, A. M. (2020). La inteligencia artificial y la robótica: sus dilemas sociales, éticos y jurídicos. *Derecho global. Estudios sobre derecho y justicia*, 6(16):49–105.

[23] Sordo, C. and Gutiérrez, C. (2013). Cáncer de piel y radiación solar: experiencia peruana en la prevención y detección temprana del cáncer de piel y melanoma. *Revista Peruana de Medicina Experimental y Salud Publica*, 30:113–117.

[24] Vega, M. Á. et al. (2020). Inteligencia artificial y aprendizaje automático en medicina. *Revista médica sinergia*, 5(8):e557–e557 Zaballos, P. et al. (2004). Criterios dermatoscópicos para el diagnóstico del melanoma. *Med Cutan Iber Lat Am*, 32(1):3–17.

[25] Zaballos, P. et al. (2004). Criterios dermatoscópicos para el diagnóstico del melanoma. *Med Cutan Iber Lat Am*, 32(1):3–17.

