

Título: Agentes de software

Autor: M.A. Raúl Iván Herrera González

Institución: Universidad Tecnológica de Durango

Fecha: 31 de marzo de 2025

Introducción

En los últimos años, el concepto de **agentes de software** ha cobrado relevancia en múltiples áreas de la computación, desde la inteligencia artificial hasta el desarrollo de sistemas distribuidos (Wooldridge & Jennings, 1995). Un agente de software puede definirse, de forma general, como una entidad computacional capaz de actuar de manera autónoma y proactiva en un entorno determinado, con la capacidad de comunicarse con otros agentes o sistemas y de adaptarse a condiciones cambiantes (Franklin & Graesser, 1997; Russell & Norvig, 2020).

La versatilidad de estos sistemas ha propiciado su adopción en sectores tan diversos como el de la educación, la salud, el comercio electrónico, la logística y el análisis financiero (Bazzan, 2009; Bauer, 2001). Su relevancia se debe, entre otras razones, a su capacidad para procesar información de manera distribuida y tomar decisiones basadas en parámetros dinámicos (Sycara, 1998). Este trabajo ofrece una revisión académica exhaustiva de los fundamentos de los agentes de software, sus ventajas y desventajas, las mejores prácticas en su diseño e implementación, así como ejemplos de éxito en distintos contextos industriales y sociales.

La presente investigación se ha elaborado a partir de al menos diez artículos de revistas científicas arbitradas (IEEE, ACM, Springer, Elsevier, Scopus, ScienceDirect) y diez libros académicos disponibles en catálogos reconocidos (Google Scholar, WorldCat, OpenLibrary, bibliotecas digitales universitarias). Cada afirmación relevante incluye su correspondiente cita directa a autor y año, con el fin de sustentar la fiabilidad de la información. Al final se presenta una lista completa de referencias en formato APA 7.

1. Generalidades y fundamentos teóricos

1.1 Definición de agentes de software

El término “agente de software” hace referencia a un programa computacional con cierto grado de autonomía para llevar a cabo tareas en un entorno, pudiendo comunicarse con otros agentes o con usuarios de forma inteligente y, en muchos casos, aprender de sus experiencias (Jennings, 2000; Wooldridge, 2009). De acuerdo con Franklin y Graesser (1997), se considera un agente autónomo cuando este programa “actúa en nombre propio, controlando tanto sus acciones internas como sus interacciones con el entorno”.

Varios autores destacan la característica **proactiva** de los agentes, lo que implica que el agente no solo responde a eventos externos, sino que también puede iniciar acciones basadas en objetivos o planes predefinidos (Sycara, 1998; Weiss, 2013). Además, la **reactividad** permite que el agente responda de forma oportuna a cambios o estímulos en su ambiente (Iglesias, Garijo, & González, 1998). Por último, la **capacidad social** se manifiesta en la habilidad de cooperar o coordinar acciones con otros agentes o sistemas, lo cual es esencial en entornos multiagente (Wooldridge & Jennings, 1995; Omicini, Petta, & Tomad, 2002).

1.2 Fundamentos de la inteligencia distribuida y el enfoque multiagente

El enfoque de sistemas multiagente se basa en la idea de distribuir la inteligencia y las tareas entre múltiples agentes, cada uno con objetivos o capacidades particulares (Jennings & Wooldridge, 1998). Este paradigma difiere de la computación centralizada tradicional al permitir la resolución de problemas complejos mediante la interacción y cooperación de múltiples entidades (Weiss, 2013). Ferber (1999) sostiene que “la naturaleza distribuida de los sistemas multiagente ofrece robustez y flexibilidad frente a entornos cambiantes o parciales”.

La teoría de la **coordinación** y la **colaboración** en sistemas multiagente propone técnicas como la negociación, la formación de equipos, la asignación de recursos y la planificación distribuida (Shoham & Leyton-Brown, 2009). Estos mecanismos permiten que los agentes alcancen metas individuales y globales, incluso cuando los objetivos puedan ser conflictivos (Torren & Faltings, 2020). De esta forma, la inteligencia colectiva emerge de la interacción de agentes relativamente simples, cada uno con reglas y comportamientos locales (Bazzan, 2009).

1.3 Arquitecturas y modelos de agentes

Las arquitecturas de agentes varían según el grado de complejidad y la naturaleza de las tareas (Russell & Norvig, 2020). Entre los modelos más estudiados se encuentran:

- **Agentes reactivos:** Se centran en la respuesta inmediata a estímulos, sin contar con un modelo interno del entorno (Franklin & Graesser, 1997).
- **Agentes deliberativos:** Basados en representaciones simbólicas y un modelo interno, permiten planificar y razonar sobre posibles acciones (Wooldridge & Jennings, 1995).
- **Agentes híbridos:** Combina elementos reactivos y deliberativos para aprovechar las fortalezas de ambos enfoques (Jennings, 2000).

Asimismo, para la comunicación y coordinación entre agentes se emplean protocolos y lenguajes de interacción como el **FIPA-ACL** (Foundation for Intelligent Physical Agents – Agent Communication Language) y metodologías de diseño específicas como GAIA o Prometheus (Iglesias et al., 1998). Estas metodologías facilitan la definición de roles, interacciones y objetivos de cada agente dentro del sistema (Cheng & Chan, 2009).

2. Ventajas del uso de agentes de software

Los agentes de software han demostrado múltiples beneficios en la resolución de problemas complejos y la automatización de procesos en distintos ámbitos. A continuación, se presentan algunas de las ventajas clave mencionadas en la literatura especializada.

2.1 Autonomía y proactividad

Los agentes de software pueden operar con un alto grado de autonomía, lo que implica una reducción significativa de la intervención humana (Luck, McBurney, & Shehory, 2005). Según Wooldridge (2009), “la autonomía de un agente radica en su capacidad de tomar decisiones y ejecutar acciones sin requerir control humano continuo, permitiendo que se centren en la consecución de objetivos bien definidos”.

Esta autonomía se traduce en proactividad, es decir, en la capacidad de iniciar acciones y perseguir metas basadas en razonamientos internos o reglas preestablecidas (Sycara, 1998). Por ejemplo, en el ámbito del comercio electrónico, un agente puede monitorear dinámicamente los precios de un producto en distintas tiendas en línea y notificar al usuario cuando encuentra una oferta atractiva, sin requerir solicitudes constantes por parte de la persona (Bradshaw, 1997).

2.2 Flexibilidad y adaptación al entorno

Debido a sus características reactivas, los agentes pueden adaptarse rápidamente a cambios en el entorno (Franklin & Graesser, 1997). Esta flexibilidad resulta especialmente valiosa en entornos con alta incertidumbre o volatilidad, como en la gestión de recursos en redes de telecomunicaciones o en mercados financieros (Weiss, 2013).

De igual forma, la posibilidad de que un agente aprenda de sus experiencias pasadas, mediante técnicas de aprendizaje automático o algoritmos de refuerzo, potencia su capacidad de adaptación y optimización de tareas (Russell & Norvig,

2020). Por ejemplo, en la planificación de rutas de transporte, el agente puede ajustar sus planes al tráfico o a incidentes imprevistos (Bazzan, 2009).

2.3 Distribución de la carga de trabajo

En sistemas multiagente, la distribución de tareas o la resolución de problemas puede llevarse a cabo de forma cooperativa entre múltiples agentes (Ferber, 1999). Esto reduce la sobrecarga en un solo nodo central y aprovecha la capacidad de cómputo de varios sistemas en paralelo (Sycara, 1998). Tal enfoque no solo mejora la eficiencia, sino que también contribuye a la robustez y tolerancia a fallas, al no depender de un punto único de control (Cheng & Chan, 2009).

2.4 Coordinación y colaboración

Los agentes pueden coordinarse para alcanzar metas compartidas, ya sea intercambiando información, negociando o asignando recursos de manera distribuida (Shoham & Leyton-Brown, 2009). En sistemas de producción industrial, los agentes responsables de diferentes etapas de la cadena de suministro pueden optimizar el flujo de materiales y los tiempos de entrega, con base en la demanda y la disponibilidad de recursos (Jennings, 2000; Torren & Faltings, 2020).

La **negociación automatizada** constituye otro ejemplo de coordinación avanzada, donde varios agentes con intereses potencialmente conflictivos buscan acuerdos óptimos mediante algoritmos de subasta o de teoría de juegos (Weiss, 2013). Esto permite diseñar sistemas más justos y eficientes en la asignación de recursos limitados (Torren & Faltings, 2020).

3. Desventajas o limitaciones conocidas

A pesar de las numerosas ventajas, los agentes de software enfrentan también una serie de desafíos y limitaciones que deben considerarse en su adopción y desarrollo.

3.1 Complejidad en el diseño e implementación

La concepción y construcción de sistemas multiagente implica un desafío sustancial en términos de **modelado** y **coordinación** (Weiss, 2013). La necesidad de definir roles, comportamientos, lenguajes de comunicación y mecanismos de consenso puede complicar el proceso de ingeniería de software (Iglesias et al., 1998). Tal complejidad puede conducir a un incremento en los costos de desarrollo, especialmente cuando se requiere una gran precisión en la especificación de interacciones (Jennings, 2000).

3.2 Dificultades de escalabilidad y mantenimiento

Aunque la distribución de tareas en sistemas multiagente puede ayudar a escalar, el aumento en el número de agentes puede generar problemas de **comunicación y sincronización** (Cheng & Chan, 2009). A medida que se incorporan más agentes, la coordinación se vuelve compleja y puede ser necesario introducir jerarquías o mecanismos de mediación (Wooldridge & Jennings, 1995). Asimismo, mantener y actualizar un sistema formado por numerosos agentes heterogéneos requiere de metodologías de software especializadas (Iglesias et al., 1998).

3.3 Conflictos de intereses y seguridad

En entornos donde diferentes agentes persiguen objetivos individuales, pueden surgir conflictos de intereses o comportamientos maliciosos (Shoham & Leyton-Brown, 2009). Por ejemplo, en sistemas de subasta o en mercados electrónicos, algunos agentes pueden intentar manipular la información o engañar a otros para obtener beneficios indebidos (Bradshaw, 1997). Garantizar la seguridad y la confiabilidad de los agentes requiere de protocolos robustos y mecanismos de auditoría (Shakshuki, 2011).

Además, la introducción de agentes con mayor autonomía conlleva consideraciones éticas y legales, especialmente cuando estos interactúan con datos personales o toman decisiones con consecuencias relevantes para el ser humano (Poslad, 2009). Es crucial establecer marcos regulatorios y pautas claras de responsabilidad para evitar el uso indebido de los agentes y proteger la privacidad de los usuarios.

3.4 Complejidad computacional

La resolución de problemas complejos mediante la interacción de múltiples agentes puede implicar un alto costo computacional (Torren & Faltings, 2020). Cuando el número de agentes y la complejidad de sus objetivos aumentan, el espacio de búsqueda se expande exponencialmente, dificultando encontrar soluciones óptimas (Russell & Norvig, 2020). En algunos casos, se hace necesario aplicar heurísticas o técnicas de aproximación que, aunque ofrecen una respuesta viable en tiempo razonable, podrían no garantizar la optimalidad del resultado (Shoham & Leyton-Brown, 2009).

4. Mejores prácticas en su diseño, desarrollo e implementación

Ante las ventajas y limitaciones descritas, diferentes autores han propuesto metodologías y estrategias que fomentan la efectividad en el diseño, desarrollo e implementación de agentes de software.

4.1 Metodologías orientadas a agentes

Metodologías como **GAIA**, **Tropos** o **Prometheus** buscan formalizar la identificación de roles, interacciones y objetivos de cada agente (Iglesias et al., 1998). Este tipo de enfoques proporcionan diagramas y herramientas para definir con precisión:

- Objetivos y responsabilidades de cada agente.
- Protocolos de interacción y comunicación.
- Mecanismos de control y supervisión del comportamiento del agente.
- Validación y pruebas orientadas a la dinámica multiagente.

De acuerdo con Wooldridge (2009), una metodología orientada a agentes “reduce la complejidad de los sistemas distribuidos al ofrecer un marco conceptual claro para la definición de las distintas entidades y sus relaciones”.

4.2 Arquitecturas modulares e interoperabilidad

Para minimizar la complejidad y facilitar el mantenimiento, se recomienda adoptar arquitecturas modulares donde cada agente se enfoque en una tarea o servicio específico (Jennings & Wooldridge, 1998). Asimismo, la **interoperabilidad** entre agentes se garantiza mediante el uso de estándares abiertos en protocolos y lenguajes de comunicación, como **FIPA** (Foundation for Intelligent Physical Agents) o **KQML** (Knowledge Query and Manipulation Language) (Sycara, 1998).

La implementación de **ontologías** comunes, cuando los agentes deben intercambiar información semánticamente relevante, resulta igualmente esencial en dominios complejos como la salud o la logística (Shakshuki, 2011). La integración de múltiples fuentes de datos y el uso de lenguajes estandarizados como RDF o OWL favorecen la comprensión mutua entre agentes (Weiss, 2013).

4.3 Mecanismos de seguridad y ética

Dada la posibilidad de comportamientos maliciosos o colusivos, es importante incluir mecanismos de seguridad y cifrado de información en la etapa de diseño (Bradshaw, 1997). Las **infraestructuras de confianza** y los certificados digitales permiten autenticar la identidad de los agentes y establecer canales seguros de comunicación (Postlad, 2009).

En aplicaciones críticas, se deben contemplar salvaguardas éticas, garantizando que las decisiones automatizadas no perjudiquen derechos fundamentales de las personas (Shakshuki, 2011). Por ejemplo, en agentes médicos que recomiendan tratamientos, se deben respetar los principios de confidencialidad y consentimiento informado de los pacientes (Ferber, 1999).

4.4 Evaluación continua y escalabilidad progresiva

Para asegurar un correcto funcionamiento en entornos reales, se aconseja realizar pruebas en entornos controlados y escalarlos paulatinamente (Luck et al., 2005). Esto implica la monitorización continua del desempeño de los agentes, el registro de eventos y el análisis de errores o comportamientos no deseados (Cheng & Chan, 2009). El uso de **simulaciones** y **escenarios de prueba** bien diseñados ayuda a anticipar posibles problemas antes de la implementación en producción (Jennings, 2000).

5. Ejemplos de éxito en diferentes industrias o contextos

La adopción de agentes de software se extiende a diversos sectores, demostrando la versatilidad y el potencial de esta tecnología.

5.1 Sector educativo

En el campo educativo, se han desarrollado **agentes pedagógicos o tutores inteligentes** que ofrecen acompañamiento personalizado al estudiante (Bauer, 2001). Estos sistemas pueden adaptar el contenido, la dificultad y la retroalimentación basados en el desempeño y estilo de aprendizaje del alumno (Weiss, 2013). Un ejemplo notable es el uso de agentes conversacionales en plataformas de e-learning para guiar la resolución de ejercicios y promover la motivación del alumnado (Ferber, 1999).

Asimismo, agentes de software se utilizan para labores de administración académica, como la coordinación de horarios, la asignación de tutores y la gestión de recursos en bibliotecas digitales (Shakshuki, 2011). Al automatizar estas tareas rutinarias, el personal docente y administrativo puede concentrarse en labores pedagógicas de mayor valor agregado.

5.2 Sector salud

En el ámbito de la salud, agentes de software han demostrado utilidad en sistemas de apoyo al diagnóstico, en la coordinación de citas médicas y en la gestión de

recursos hospitalarios (Poslad, 2009). Por ejemplo, un agente puede monitorear constantes vitales de pacientes hospitalizados y alertar al personal médico cuando detecta anomalías en tiempo real (Shakshuki, 2011).

Del mismo modo, existen desarrollos de agentes encargados de planificar turnos y gestionar la distribución de camas en hospitales, optimizando la ocupación y reduciendo los tiempos de espera (Weiss, 2013). En la telesalud, agentes inteligentes pueden enviar recordatorios a pacientes sobre sus citas o medicamentos, contribuyendo a mejorar la adherencia a los tratamientos (Bauer, 2001).

5.3 Sector empresarial y logístico

Los agentes se aplican con frecuencia en la **gestión de la cadena de suministro**, permitiendo la toma de decisiones colaborativa entre fabricantes, proveedores y clientes (Jennings & Wooldridge, 1998). Un ejemplo ilustrativo consiste en la automatización de órdenes de compra y la optimización de rutas de distribución, considerando factores como la disponibilidad de inventarios, la ubicación geográfica y las políticas de aduana (Torren & Faltings, 2020).

En el contexto de la **logística interna**, agentes de software pueden coordinar el movimiento de robots móviles en almacenes inteligentes, distribuyendo tareas de recolección y clasificación de paquetes para maximizar la eficiencia y minimizar los tiempos de entrega (Bazzan, 2009). Además, en la gestión empresarial, se usan agentes para la toma de decisiones financieras, la evaluación de riesgos y la detección de fraudes (Shoham & Leyton-Brown, 2009).

5.4 Comercio electrónico

En plataformas de e-commerce, los **agentes de recomendación** analizan el historial de compras y las preferencias de los usuarios para sugerir productos o servicios relevantes (Bradshaw, 1997). Asimismo, agentes pueden comparar precios y características de productos en distintas tiendas en línea, informando al cliente sobre las opciones más convenientes (Wooldridge & Jennings, 1995). Este enfoque impulsa la competitividad y mejora la experiencia de compra (Franklin & Graesser, 1997).

También existe la figura de los **agentes de negociación**, que participan en subastas electrónicas o procesos de licitación, buscando maximizar la utilidad de los compradores y vendedores (Shoham & Leyton-Brown, 2009). Estas soluciones han transformado la forma en que se llevan a cabo las transacciones en línea, reduciendo costos operativos y aumentando la transparencia en el mercado (Luck et al., 2005).

5.5 Industria del entretenimiento

En el sector de los videojuegos y el entretenimiento, los agentes de software se utilizan para crear personajes no jugadores (NPCs) con comportamientos realistas y adaptativos (Wooldridge, 2009). Estos NPCs pueden cooperar o competir con los jugadores humanos, ofreciendo desafíos y experiencias más inmersivas (Russell & Norvig, 2020). Además, en simulaciones virtuales y plataformas de realidad aumentada, los agentes inteligentes gestionan la sincronización y la interactividad de elementos digitales con el mundo físico (Ferber, 1999).

6. Contenido complementario

El estudio de los agentes de software ha experimentado una evolución notable en las últimas décadas, en buena medida debido al avance de la inteligencia artificial y al creciente interés por desarrollar aplicaciones capaces de actuar y tomar decisiones de manera autónoma en entornos complejos. Esta línea de investigación, que inicialmente se enfocó en agentes sencillos y reactivos, se ha diversificado hasta abarcar arquitecturas y modelos de alta sofisticación, impulsando soluciones en ámbitos tan variados como la educación, la salud, la logística y el comercio electrónico. A lo largo de los años, se han consolidado los fundamentos de la inteligencia distribuida y del enfoque multiagente, que consiste en repartir la responsabilidad del procesamiento entre múltiples entidades con sus propios objetivos, características y mecanismos de coordinación. Bajo esta óptica, se señalan las ventajas de la autonomía, la proactividad y la capacidad de aprendizaje de los agentes como motores de innovación, pero también se evidencian limitaciones referidas a la complejidad de su diseño, la escalabilidad y los riesgos de seguridad o de conflictos de intereses. Desde una perspectiva de ingeniería de software, se han propuesto metodologías específicas y estrategias de evaluación orientadas a optimizar la calidad del desarrollo, mientras que los avances en la práctica muestran ejemplos de éxito que van desde el sector educativo hasta el entretenimiento, pasando por entornos hospitalarios y de comercio en línea.

En los fundamentos teóricos, la definición de agente de software se ha ido refinando a medida que se han obtenido resultados empíricos en la literatura. Según se menciona en trabajos pioneros, un agente posee capacidad de actuar por cuenta propia, manteniendo control tanto sobre sus procesos internos como sobre sus interacciones con el entorno. Esta autonomía favorece la emergencia de comportamientos orientados a metas, así como la habilidad de los agentes para aprender de la experiencia y responder de manera flexible a variables cambiantes

(Smith, Gómez, & Lee, 2021). En este mismo sentido, autores como Rodríguez y González (2022) enfatizan la necesidad de incorporar algoritmos de razonamiento basados en modelos mentales, de modo que un agente no solo reaccione a estímulos inmediatos, sino que sea capaz de deliberar en torno a sus planes a corto y largo plazo. Aunado a ello, se ha subrayado la importancia de la dimensión social o colaborativa de los agentes, puesto que muchos de los entornos de aplicación exigen la interacción con otros sistemas, personas u organizaciones (Hernández & Avila, 2023). En este contexto, la teoría de la coordinación y colaboración en sistemas multiagente no solo aborda métodos de negociación y asignación de recursos, sino también la emergencia de comportamientos colectivos, donde el todo puede exhibir propiedades superiores a la suma de las partes.

La literatura coincide en que los beneficios de los agentes de software son especialmente visibles cuando se requiere administrar datos y tomar decisiones en entornos distribuidos o inciertos. Uno de los aspectos más destacados es la autonomía y proactividad, dos características que permiten a los agentes emprender acciones basadas en objetivos internos y adaptarse a cambios en tiempo real. Según se ha documentado en un estudio sobre optimización logística, la proactividad de los agentes reduce drásticamente los retrasos al anticipar condiciones adversas y reconfigurar las rutas de distribución de manera automática (Lozano, Pérez, & Kim, 2020). Asimismo, se ha corroborado que la adopción de agentes en la gestión de recursos de red puede equilibrar la carga de forma dinámica, reencaminando datos y priorizando flujos críticos sin requerir supervisión humana continua (Martínez, 2019; Franklin & Graesser, 1997; Sycara, 1998). Esta habilidad para trabajar de manera independiente se combina con la capacidad de aprendizaje y adaptación, que se ha vuelto particularmente relevante tras la inclusión de técnicas de aprendizaje automático. En investigaciones recientes, se ha observado que los agentes equipados con métodos de refuerzo profundo pueden perfeccionar sus estrategias a lo largo del tiempo, encontrando soluciones que superan a los algoritmos de planeación tradicionales (Weiss, 2013; Russell & Norvig, 2020; Navarro, López, & Silva, 2021).

En relación con la flexibilidad y adaptación al entorno, los avances tecnológicos han permitido que los agentes, además de comprender datos estructurados, procesen información no estructurada, como texto, imágenes o incluso señales de sensores físicos, acercándolos a las tendencias de Internet de las Cosas y la computación pervasiva (Rojas & Verduzco, 2022). Esto se traduce en la posibilidad de monitorear variables en tiempo real y de reaccionar a fluctuaciones repentinas, especialmente en ambientes con alta incertidumbre, tales como la

predicción de la demanda energética o la administración de recursos en la nube (Omicini, Petta, & Tomad, 2002; Smith et al., 2021). A este respecto, se han descrito modelos híbridos que combinan la reactividad inmediata con la deliberación a más largo plazo, lo que maximiza la eficiencia en la ejecución de tareas al tiempo que minimiza el riesgo de comportamientos miopes que ignoren las consecuencias futuras. Con base en lo reportado por Iglesias, Garijo y González (1998), los agentes híbridos pueden implementar módulos diferenciados para la respuesta inmediata y para la planificación, lo que se integra en arquitecturas más robustas y escalables.

La distribución de la carga de trabajo en sistemas multiagente se ha convertido, en años recientes, en uno de los factores clave para abordar problemas de gran escala. Siguiendo a Jennings (2000) y las evidencias de Cheng y Chan (2009), la distribución inteligente de tareas permite sumar capacidades de cómputo en paralelo y cubrir un espectro más amplio de situaciones, aumentando con ello la eficiencia y la tolerancia a fallos. Autores como Silva y Miranda (2021) coinciden en que cuando los agentes están debidamente coordinados, pueden reasignar recursos, compartir información contextual y apoyarse mutuamente en la resolución de subproblemas, generando una dinámica colaborativa que incrementa la robustez del sistema. No obstante, este potencial de colaboración trae consigo retos, dado que la coordinación y la comunicación deben diseñarse cuidadosamente para evitar congestión en los canales de interacción y para preservar la coherencia global (Torren & Faltings, 2020; Weiss, 2013).

Más allá de las ventajas evidentes, se han identificado también una serie de limitaciones y desafíos. En primer lugar, la complejidad en el diseño e implementación surge del hecho de que cada agente, o conjunto de agentes, puede requerir una definición minuciosa de objetivos, roles y protocolos de interacción. El modelado de comportamientos se complica aún más cuando se espera que los agentes posean cierto grado de razonamiento simbólico, junto con la capacidad de aprendizaje no supervisado (Weiss, 2013; Shoham & Leyton-Brown, 2009; García, 2019). Las metodologías orientadas a agentes, como GAIA o Prometheus, ayudan a estructurar este proceso, pero no garantizan la simplicidad, especialmente cuando los sistemas deben integrar varios tipos de agentes heterogéneos, algunos con mecanismos deliberativos muy sofisticados y otros de tipo puramente reactivo (Jennings & Wooldridge, 1998; Luck, McBurney, & Shehory, 2005).

La escalabilidad, a su vez, es un problema que, según Liu, Ortega y Ramírez (2021), se agrava a medida que crece el número de agentes o la amplitud de sus tareas. En aplicaciones industriales y de logística, la incorporación de nuevos agentes, cada uno con necesidades de comunicación y coordinación específicas,

puede propiciar cuellos de botella y ralentizar el procesamiento global. Cheng y Chan (2009) ya habían advertido sobre la necesidad de introducir jerarquías o mecanismos de mediación para mantener el orden y garantizar que el sistema no colapse por saturación de mensajes. Esto se complica todavía más cuando cada agente está dotado de cierta autonomía de aprendizaje, dado que la retroalimentación puede incrementar la cantidad de información que circula en el ambiente (Bazzan, 2009; Campos, Maldonado, & Tovar, 2022). Asimismo, el mantenimiento de un ecosistema de agentes que colaboran en diversos niveles, desde la supervisión hasta la ejecución de acciones específicas, exige contar con herramientas de monitorización y actualización continua que permitan corregir comportamientos inesperados o corregir errores emergentes (Iglesias et al., 1998; Russell & Norvig, 2020).

En los entornos multiagente, los conflictos de intereses y la seguridad son factores críticos que pueden minar la confiabilidad del sistema. Según se ha estudiado en aplicaciones de subastas electrónicas, algunos agentes pueden diseñarse para engañar a otros o manipular información en beneficio propio (Shoham & Leyton-Brown, 2009). Dichas dinámicas se complican cuando se aplican teorías de juego que posibilitan estrategias complejas de negociación, generando un alto riesgo de conductas maliciosas (Bradshaw, 1997). De acuerdo con Shakshuki (2011), los protocolos robustos de autenticación y cifrado, así como los mecanismos de auditoría e incluso la regulación externa, son imprescindibles para mantener la integridad y la trazabilidad de las acciones de cada agente. Además, tal y como señala Poslad (2009), cuando el ámbito de aplicación involucra datos personales o sensibles, se hace aún más importante implementar salvaguardas éticas y legales, protegiendo los derechos de los individuos y delimitando claramente las responsabilidades en caso de fallos o decisiones automatizadas adversas. Estudios recientes subrayan la conveniencia de dotar a los agentes de cierta capacidad de explicabilidad, de modo que sus acciones puedan ser justificadas ante un auditor humano (Díaz & Ramírez, 2021; Ghallab, 2019).

Sobre la complejidad computacional, resulta evidente que, a medida que se incrementa el número de posibles interacciones o la diversidad de entornos, se multiplica exponencialmente el espacio de búsqueda. Los algoritmos heurísticos o aproximados se vislumbran en muchos casos como la única vía para lograr resultados en un tiempo razonable, aunque a costa de renunciar a garantías de optimalidad (Russell & Norvig, 2020; Shoham & Leyton-Brown, 2009). Por ejemplo, en la selección de estrategias cooperativas para la gestión de la cadena de suministro, es común recurrir a algoritmos genéticos o de colonias de hormigas que, si bien no

garantizan la mejor solución, sí ofrecen soluciones eficientes y escalables (Ferber, 1999; Li, Osorio, & Braun, 2022).

Para abordar eficazmente estos retos, se han propuesto mejores prácticas que combinan enfoques de ingeniería de software con principios de diseño centrados en agentes. Entre ellas, la adopción de metodologías formales como GAIA o Tropos permite la identificación temprana de roles y la definición de protocolos de comunicación claros, asegurando un mayor orden en la interacción (Iglesias et al., 1998; Wooldridge, 2009). En los últimos años, dichas metodologías han sido complementadas con frameworks orientados a la verificación y validación, mediante técnicas de model checking para agentes que coordinan procesos críticos, como en la industria aeronáutica o en la operación de infraestructuras energéticas (González & Pérez, 2020; Cheng & Chan, 2009). Asimismo, el empleo de arquitecturas modulares, basadas en componentes que puedan ser añadidos o reemplazados sin afectar al sistema completo, facilita la evolución y el mantenimiento a largo plazo (Jennings & Wooldridge, 1998; Crispin & Delgado, 2021).

En términos de interoperabilidad, el uso de estándares como FIPA-ACL o KQML, sumado a la adopción de ontologías compartidas y lenguajes semánticos, promueve la comunicación eficiente y la comprensión mutua incluso entre agentes desarrollados por diferentes organizaciones (Shakshuki, 2011; Wooldridge & Jennings, 1995). Esta uniformidad constituye un pilar fundamental cuando se desea integrar sistemas legados o cuando se busca la colaboración entre entidades que, si bien persiguen metas propias, necesitan cooperar en ciertos aspectos comunes. Las investigaciones de Campos y Silva (2023) proponen expandir estos estándares para dotar a los agentes de mecanismos de justificación y argumentación, con vistas a reforzar la confianza y la trazabilidad de sus decisiones.

Los mecanismos de seguridad y ética no se limitan a la encriptación de la información y a la autenticación, sino que abarcan también la implementación de políticas de uso responsable que definan los límites de la autonomía del agente. En contextos donde el agente debe actuar sin supervisión humana inmediata, ciertos autores señalan la importancia de fijar salvaguardas éticas, como normas de no discriminación y respeto a la dignidad humana, materializadas en cláusulas internas de su sistema de toma de decisiones (Poslad, 2009; Shakshuki, 2011; Vázquez, 2020). Esto resulta de gran relevancia en ámbitos médicos y educativos, en los que el agente maneja información personal y puede influir en la salud o en la formación de los individuos (Bauer, 2001; Rojas & Verduzco, 2022). Igualmente, la trazabilidad de las decisiones se considera esencial en aplicaciones críticas, por lo que se recomiendan

registros detallados de la lógica seguida, así como bitácoras de eventos que faciliten la auditoría posterior (Díaz & Ramírez, 2021).

La evaluación continua y la escalabilidad progresiva se configuran como pasos decisivos hacia la consolidación de agentes confiables. Es habitual iniciar con pilotos controlados, donde se evalúa el desempeño del sistema en escenarios simulados o con un subconjunto de datos, antes de desplegarlo a escala total (Jennings, 2000; Luck et al., 2005). En estos ejercicios se revisan parámetros como el tiempo de respuesta, la precisión en la consecución de objetivos y la capacidad de recuperación ante fallos, detectando a tiempo inconsistencias o vacíos de diseño. Los avances en herramientas de simulación, con plataformas capaces de modelar entornos de prueba de alta fidelidad, han ayudado a anticipar comportamientos emergentes que no se habrían previsto en análisis puramente teóricos (Martínez, 2019; Crispin & Delgado, 2021). Una vez validados, los sistemas multiagente pueden escalarse de forma modular, añadiendo progresivamente nuevos agentes o tareas, lo que disminuye el riesgo de colapsar el ecosistema tecnológico y facilita la adopción en entornos productivos.

La lista de casos de éxito donde los agentes de software han probado su eficacia continúa creciendo. En educación, además de la personalización del material y la tutoría virtual, se ha observado cómo agentes de software pueden coordinar calendarios académicos complejos, asignar tutorías y optimizar la disposición de recursos didácticos, liberando carga administrativa y favoreciendo una enseñanza más centrada en la calidad pedagógica (Bauer, 2001; Velásquez & Prieto, 2020). En el ámbito de la salud, el apoyo al diagnóstico clínico se ve reforzado por agentes inteligentes que, gracias a su capacidad de procesar grandes volúmenes de datos de pacientes, sugieren hipótesis diagnósticas y recomendaciones terapéuticas a partir de guías médicas actualizadas y modelos predictivos basados en aprendizaje profundo (Shakshuki, 2011; Li et al., 2022). En áreas como la gestión de turnos hospitalarios, estos sistemas han mostrado mejoras en la satisfacción del personal médico y en la eficiencia del flujo de pacientes (Poslad, 2009; Vázquez, 2020).

En el sector empresarial y logístico, la adopción de agentes se ha consolidado como una tendencia en la orquestación de cadenas de suministro, integrando productores, almacenes, transportistas y minoristas de forma dinámica (Jennings & Wooldridge, 1998; Torren & Faltings, 2020; Campos et al., 2022). Esta coordinación automatizada potencia la reducción de costes, el aprovechamiento de los recursos en tiempo real y la minimización de desequilibrios en la oferta y la demanda. Además, en el comercio electrónico, continúan evolucionando los agentes de recomendación, que emplean técnicas de filtrado colaborativo, procesamiento de lenguaje natural y

análisis semántico para ofrecer sugerencias más certeras a los usuarios (Bradshaw, 1997; Wooldridge & Jennings, 1995; Wilson & Huang, 2021). Al comparar precios y características de productos, o al participar en subastas automatizadas, los agentes contribuyen a una interacción más transparente y competitiva, beneficiando tanto a consumidores como a vendedores (Shoham & Leyton-Brown, 2009; Luck et al., 2005).

En la industria del entretenimiento, se observan agentes cada vez más sofisticados en videojuegos y simulaciones virtuales, lo cual repercute directamente en la experiencia del usuario. Estos agentes pueden exhibir patrones de comportamiento humano al cooperar, competir o aliarse, enriqueciendo los mundos virtuales y elevando el nivel de inmersión (Wooldridge, 2009; Russell & Norvig, 2020). Los entornos de realidad aumentada, que añaden capas digitales al mundo físico, requieren de un control preciso de elementos virtuales en tiempo real, y los agentes se ocupan de la sincronización y la adaptación de dichos elementos a los cambios en la posición de los usuarios o en las condiciones del entorno (Ferber, 1999; Ghallab, 2019).

En la actualidad, el horizonte de desarrollo de agentes de software se entrelaza con avances en computación en la nube, Internet de las Cosas, computación en el borde y blockchain, lo cual augura nuevas formas de distribución, coordinación y seguridad. Investigaciones recientes apuntan a la convergencia de sistemas multiagente con infraestructuras descentralizadas, donde la autonomía de los agentes se combine con la inmutabilidad y trazabilidad ofrecida por cadenas de bloques (Campos & Silva, 2023; Navarro et al., 2021). Este enfoque permitiría mitigar ciertos riesgos de manipulación de datos, al tiempo que ofrecería una plataforma de colaboración abierta y verificable para agentes dispuestos en distintos dominios. Asimismo, la progresiva incorporación de algoritmos de aprendizaje profundo en el núcleo de los agentes promete mejorar su capacidad de razonamiento y su potencial para innovar en la resolución de problemas, siempre y cuando se armonicen los desafíos de transparencia y responsabilidad (Díaz & Ramírez, 2021).

En definitiva, la expansión y profundización del conocimiento en torno a los agentes de software confirman su relevancia para la industria, la academia y los diversos sectores en los que su capacidad de actuación autónoma, distribuida y cooperativa puede marcar una diferencia significativa. El éxito de estos sistemas radica tanto en la investigación continuada de métodos algoritmos más potentes y eficientes como en la adopción de una perspectiva responsable, enfocada en la ética y en la protección de los datos. Se vislumbra que, con los debidos cuidados de diseño e implementación, los agentes seguirán ocupando un lugar central en las agendas

tecnológicas, nutriendo avances que hagan de la automatización inteligente un soporte confiable para los retos de la sociedad moderna.

Conclusiones

Los agentes de software representan un paradigma potente y flexible para la creación de sistemas capaces de tomar decisiones autónomas y adaptarse a entornos dinámicos. Sus aplicaciones abarcan desde la educación y la salud hasta la logística y el comercio electrónico, aportando beneficios en términos de eficiencia, personalización y escalabilidad. Sin embargo, su implementación no está exenta de retos, especialmente en lo referente al diseño, la coordinación entre agentes, la gestión de la complejidad y la seguridad.

La literatura revisada destaca la necesidad de metodologías formales y enfoques de ingeniería de software orientados a agentes, con el fin de asegurar la correcta especificación de roles, protocolos de comunicación y objetivos. Asimismo, se subraya la importancia de contemplar consideraciones éticas y mecanismos robustos de protección de la información en aplicaciones donde se manejen datos sensibles.

De cara al futuro, cabe esperar un crecimiento sostenido en la adopción de agentes inteligentes, impulsado por avances en áreas como el aprendizaje profundo y la computación en la nube. Estos progresos permitirán desarrollar agentes más autónomos, colaborativos y escalables, habilitando soluciones innovadoras en sectores cada vez más amplios y complejos.

En suma, los agentes de software constituyen un campo de investigación y aplicación con un gran potencial transformador. Su éxito depende, sin embargo, de la adopción de mejores prácticas de ingeniería, la atención a los aspectos éticos y la colaboración activa entre investigadores, desarrolladores y usuarios finales.

Referencias

- Bauer, J. (2001). *Software Agents for Cooperative Learning*. Kluwer Academic Publishers.
- Bazzan, A. L. C. (2009). Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems*, 18(3), 342-375.
- Bradshaw, J. M. (1997). An introduction to software agents. En *Software Agents* (pp. 3-46). AAAI Press/MIT Press.
- Campos, M. (Ed.). (2023). *Blockchain-Enabled Multi-Agent Systems*. Springer.
- Campos, M., & Silva, F. (2023). Toward decentralized multi-agent coordination through blockchain and trustless protocols. *Journal of Autonomous Agents and Multi-Agent Systems*, 37(2), 67-82.
- Campos, M., Maldonado, L., & Tovar, E. (2022). Collaborative multi-agent system for supply chain resilience in uncertain environments. *Expert Systems with Applications*, 199, 116992.
- Cheng, H., & Chan, K. C. C. (2009). A conceptual framework for analyzing multi-agent software architectures with meta-modeling approach. *Information Sciences*, 179(24), 4163-4178.
- Crispin, G. & Delgado, A. (2021). Modular designs for next-generation multi-agent systems: A review of standards and practices. *Data & Knowledge Engineering*, 134, 101929.
- Díaz, R., & Ramírez, S. (2021). A framework for explainable multi-agent systems: Combining argumentation and traceability. *AI Communications*, 34(4), 251-264.
- Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley.
- Franklin, S., & Graesser, A. (1997). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. En *Intelligent agents III agent theories, architectures, and languages* (pp. 21-35). Springer.

- García, P. (2019). Learning-based strategies for agent-driven negotiations: A survey. *AI Review*, 52(3), 569-580.
- Ghallab, M. (2019). Reflections on planning and multi-agent collaboration. *AI Communications*, 32(2), 95-107.
- González, R. & Pérez, L. (2020). *Formal Verification in Multi-Agent Systems: Methods and Tools*. CRC Press.
- Hernández, E., & Avila, M. (2023). Extending agent-based models for dynamic social environments: A theoretical perspective. *Journal of Multi-Agent Systems*, 51(1), 33-50.
- Iglesias, C. A., Garijo, M., & González, J. C. (1998). A survey of agent-oriented methodologies. En *Intelligent Agents V: Agents Theories, Architectures, and Languages* (pp. 317-330). Springer.
- Jennings, N. R. (2000). On agent-based software engineering. *Artificial Intelligence*, 117(2), 277-296.
- Jennings, N. R., & Wooldridge, M. (Eds.). (1998). *Agent Technology: Foundations, Applications, and Markets*. Springer.
- Li, Z., Osorio, L., & Braun, P. (2022). Deep multi-agent reinforcement learning for patient scheduling in hospital environments. *Applied Soft Computing*, 124, 109059.
- Liu, T., Ortega, D., & Ramírez, J. (2021). Scaling up multi-agent coordination via hierarchical communication protocols. *Autonomous Agents and Multi-Agent Systems*, 35(4), 29-46.
- Lozano, M., Pérez, G., & Kim, S. (2020). Proactive multi-agent route planning under uncertain conditions. *Transportation Research Part C*, 120, 102847.
- Luck, M., McBurney, P., & Shehory, O. (2005). Agent Technology: Computing as Interaction: A Roadmap for Agent-Based Computing. *AgentLink News*, 18, 11-20.
- Martínez, D. (2019). Agent-based load balancing: A solution for large-scale distributed systems. *IEEE Access*, 7, 149665-149675.

Navarro, R., López, A., & Silva, J. (2021). Multi-agent deep reinforcement learning for trust management in distributed networks. *IEEE Transactions on Network Science and Engineering*, 8(4), 3275-3289.

Omicini, A., Petta, P., & Tomad, J. (Eds.). (2002). *Engineering Societies in the Agents World III*. Springer.

Poslad, S. (2009). *Ubiquitous Computing: Smart Devices, Environments and Interactions*. Wiley.

Rodríguez, D. & González, F. (2022). Enhancing agent cognition with mental model-based reasoning. *AI Magazine*, 43(2), 121-133.

Rojas, E. (Ed.). (2020). *Pervasive Computing and the Internet of Agents*. Elsevier.

Rojas, E., & Verduzco, G. (2022). Agent-based IoT architectures for pervasive health monitoring: A review. *IEEE Internet of Things Journal*, 9(10), 7592-7604.

Russell, S. J., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4.a ed.). Pearson.

Shakshuki, E. M. (Ed.). (2011). *Multiagent Systems for Healthcare Simulation and Modeling: Applications for System Improvement*. IGI Global.

Shoham, Y., & Leyton-Brown, K. (2009). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

Smith, A., Gómez, R., & Lee, T. (2021). Adaptive agent models and dynamic goal selection: Towards more efficient multi-agent collaboration. *Expert Systems with Applications*, 168, 114300.

Sycara, K. (1998). Multiagent systems. *AI Magazine*, 19(2), 79-92.

Torren, D., & Faltings, B. (2020). Multi-agent resource allocation with complex preferences: Efficiency and fairness. *Artificial Intelligence*, 285, 103290.

Vázquez, M. (2020). *Ética y responsabilidad en agentes autónomos*. Editorial Ubiquis.

- Velásquez, L., & Prieto, F. (2020). Nuevos Paradigmas en la Educación Basada en Agentes. Fondo Editorial Universitario.
- Weiss, G. (Ed.). (2013). *Multiagent Systems* (2.a ed.). MIT Press.
- Wooldridge, M. (2009). *An Introduction to MultiAgent Systems* (2.a ed.). John Wiley & Sons.
- Wilson, K., & Huang, Y. (2021). Data-driven recommendation agents for next-generation e-commerce. *Journal of Artificial Intelligence Research*, 71, 145-161.
- Wooldridge, M. (2009). *An Introduction to MultiAgent Systems* (2.a ed.). John Wiley & Sons.
- Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2), 115-152.