Vol.11 Num,2

Celebrando 10 AÑOS

ISSN 2007-5448

RECIBE

Revista electrónica

DE COMPUTACIÓN, INFORMÁTICA, BIOMÉDICA Y ELECTRÓNICA

ReCIBE, Vol. 11 No. 2, nov. 2022

Índice

Computación e Informática

- **Los Asistentes virtuales basados en Inteligencia Artificial** — **C1**
  Pablo Ramires Hernandez, David Valle Cruz
- **Vida artificial y criaturas virtuales** — **C2**
  Rafael Mercado Herrera, Vianney Muñoz-Jiménez, Marco Antonio Ramos
- **Introducción a enzo para simulaciones hidrodinámicas en astrofísica** — **C3**
  Eduardo Teófilo-Salvador, Patricia Ambrocio-Cruz, Margarita Rosado-Solís
- **Inteligencia Artificial y Juegos de Tablero: Desde el Turco hasta AlphaZero** — **C4**
  Ivan Francisco Valencia, Rosa María Valdovinos Rosas, J. Raymundo Marcial Romero , Roberto Alejo Eleuterio

Recibido 29 Oct. 2022 *ReCIBE, Año 11 No. 2, nov. 2022*

Aceptado 7 Nov. 2022

Asistentes virtuales basados en Inteligencia Artificial

Virtual assistants based on Artificial Intelligence

Pablo Ramires Hernandez[1]
pablo_rh98@outlook.es

David Valle Cruz[1]
davacr@uaemex.mx

[1]Universidad Autónoma del Estado de México

RESUMEN

La inteligencia artificial (IA) ha comenzado a adentrarse en diferentes ámbitos y sectores. Actualmente, existen aplicaciones de IA en nuestros teléfonos, las plataformas de streaming y para el servicio al cliente, lo que representa una actividad estratégica fundamental en las organizaciones. En este documento se presenta el estado del arte de los asistentes virtuales basados en IA. Los hallazgos de esta investigación exploratoria identificaron que, aunque, la IA permite desarrollar diferentes herramientas, como el que un asistente virtual simule conversaciones humanas, las arquitecturas existentes no son del todo eficientes. Esta situación se debe a que los chatbots actuales no explotan al máximo el potencial de las técnicas de IA, pues básicamente solo funcionan con base en menús.

Palabras clave: Chatbot, Inteligencia Artificial, PRISMA, Análisis Sistemático, Procesamiento del Lenguaje Natural, Comprensión del Lenguaje Natural, Asistente virtual.

ABSTRACT

Artificial intelligence (AI) has started to penetrate into different areas and sectors. Nowadays, there are AI applications in our smartphones, streaming platforms and for customer service, which represents a fundamental strategic activity in organizations. This paper aims to analyze the state of the art of AI-based virtual assistants. The findings of this exploratory research identified that, although AI allows the development of different tools, such as a virtual assistant to simulate human conversations, existing architectures are not fully efficient. This situation is due to the fact that current chatbots do not exploit the full potential of AI techniques, as they basically only work on a menu-driven basis.

Key words: Chatbot, Artificial Intelligence, PRISMA, Systematic Analysis, Natural Language Processing, Natural Language Understanding, Virtual Assistant.

1. INTRODUCCIÓN

Hoy en día la inteligencia artificial (IA) está presente en la optimización de los procesos y tareas tanto de las grandes empresas y organizaciones como de la vida diaria de los ciudadanos. Mediante el uso de diversas técnicas, esta tecnología ha llegado para facilitarnos la vida y ha permitido realizar procesos que anteriormente requerían la intervención humana (Boden, 2016). El uso de la IA, en la actualidad, se presenta en la asistencia de toma de decisiones y la prestación de servicios. Dicha tecnología se ha implementado en varios sectores: educación, salud, economía política, entre otros. La IA ha transformado las formas tradicionales, con aportes que han permitido mejorar los bienes y servicios que se ofrecen, así como la ventaja competitiva (Aoki, 2020).

Particularmente, la automatización del servicio al cliente se ha convertido en un puente para potenciar la interacción con organizaciones. Ya que, la creciente población que utiliza estos servicios genera la necesidad de sistemas cada vez más rápidos y eficientes (Aoki, 2020).

Día a día la población presenta grandes incrementos. Por ejemplo, según datos del INEGI del 2010 al 2020 hubo un incremento de aproximadamente 14 millones de habitantes. Debido a la importancia de la atención al cliente y el incremento drástico en la población, el sector público y privado se han visto en la necesidad de implementar herramientas tecnológicas que mejoren el servicio de atención a los ciudadanos y/o clientes. Es por ello por lo que se ha optado por desarrollar e implementar asistentes virtuales, principalmente de tipo chatbot, que permitan el acceso en tiempo real a información y apoyo referente a los bienes y/o servicios brindados por las organizaciones públicas y privadas (Makasi, Nili, Desouza y Tate, 2021).

Un asistente virtual puede ser un agente conversacional desarrollado con inteligencia artificial, particularmente con procesamiento de lenguaje natural que interactúa con los usuarios respondiendo sus consultas/preguntas. Hoy en día algunas organizaciones, asociaciones gubernamentales y asociaciones sin fines de lucro han desarrollado chatbots que son de utilidad en diversas áreas, especialmente en la atención al cliente (Abdallah, Wahab, Fahad y AlHamed, 2020).

La idea de los sistemas de asistentes virtuales de tipo chatbot se originó en el Instituto Tecnológico de Massachusetts. Inicialmente, Weizbeum implementó ELIZA (chatbot para emular a un psicoterapeuta) y luego se desarrolló PARRY para simular un paciente paranoico como "una herramienta para estudiar la naturaleza de la paranoia". ELIZA fue considerado como un agente potencial que podría atender de forma autónoma a varios pacientes. Hoy en día, existen múltiples áreas y usos donde se puede implementar un chatbot para mejorar un servicio de diferentes formas (AbuShawar y Atwell, 2015). De esta manera, este documento revisa el estado del arte de los chatbots y propone una arquitectura para su implementación.

2. REVISIÓN SISTEMÁTICA SOBRE LOS ASISTENTES VIRTUALES BASADOS EN IA

Se realizó una revisión bibliográfica sistemática para analizar y discutir artículos e informes sobre el uso de chatbots. La revisión bibliográfica se basó en la metodología PRISMA (Preferred Reporting Items for Systematic reviews and Meta- Analyses) publicada en 2009. Esta metodología se diseñó para ayudar a los autores de revisiones sistemáticas a documentar de manera transparente por qué realizaron la revisión, qué hicieron los autores y qué encontraron (McKenzie, 2021). El estado del arte, de este documento, muestra los resultados obtenidos mediante el uso de chatbots. Para lograr esto, los autores realizaron una revisión sistemática de la literatura en Google Scholar, Redalyc, IEEE Xplore, ACM Digital Library e IGI Global utilizando la metodología del PRISMA.

Para la revisión bibliográfica se realizó la siguiente búsqueda lógica "virtual assistant" y ("development platforms", OR "AI techniques", OR "use cases"). Inicialmente, se analizaron títulos, resúmenes y palabras clave.

El estudio de la presente revisión de literatura permitió identificar algunos tipos de chatbot implementados en la actualidad, así como diversos usos que se les han dado, las arquitecturas empleadas como referencia en su desarrollo, las plataformas y/o programas utilizados para su creación y las técnicas de inteligencia artificial usadas y por usar.

3. CLASIFICACIÓN DE LOS ASISTENTES VIRTUALES EXISTENTES

Actualmente existen diferentes características que poseen los asistentes virtuales que se pueden implementar para facilitar las funciones y procesos. La Tabla 1 muestra la clasificación de los diversos tipos de asistentes virtuales encontrados en la revisión de la literatura.

ACORDE AL GRADO DE INTERACCIÓN CON LOS USUARIOS	
DIRIGIDOS	Hacen preguntaspredeterminadas a los usuarios a través de algunos elementos fijos, por ejemplo, botones. De esta forma, la interacción con el usuario estará muycontrolada. También hay algunos que funcionanmediante el reconocimiento de palabras clave, apartir de las cuales dan una respuesta preconfigurada (Makasi, Nili, Desouza y Tate, 2021).
CONVERSACIONALES	Dan libertad a las preguntas que el usuario quiera hacer. Este tipo de chatbots mantiene una interacción mucho más natural con el usuario (Makasi, Nili, Desouza y Tate, 2021).
EN CUANTO A LAS FUNCIONES Y FINALIDAD	
SERVICIO DE COMUNICACIÓN Y MARKETING	Brinda un servicio de consulta dentro de una aplicación móvilo web. Por ejemplo, los bancos las integran para ayudar a los usuarios a navegar por sus sitios web oaplicaciones móviles (AbuShawar y Atwell, 2015).
ATENCIÓN AL CLIENTE	Su finalidad es asistir a losusuarios resolviendo sus dudas y consultas a través de prácticas conversaciones en una jornada ininterrumpida y sin esperas. Actualmente, algunos centros médicos concretan la cita para una consultamédica con estos servicios, también se atiende a laspreguntas frecuentes de los clientes (AbuShawar y Atwell, 2015).
SERVICIO DE MEJORA DE PROCESOS	Este servicio se implementa con el objetivo de reducir eltiempo dedicado a una tarea (Valverde y Ferreira, 2019).
MEDIO DE INTERACCIÓN	
TEXTO	Utiliza solo texto como plataforma de chat básica (Makasi, Nili, Desouza y Tate, 2021).
MULTIMEDIA	Integra texto, imágenes, botones y otros contenidos. Suelen ser chatbots basados en comandos (Makasi, Nili, Desouza y Tate, 2021).
VOZ	Utilizando lavoz como interfaz, el chatbot puede interpretar la voz del usuario y proporcionar respuestas contextuales (Makasi, Nili, Desouza y Tate, 2021).
GRADO DE AFECTIVIDAD	
NO EMOCIONALES	Son los chatbots tradicionales que se limitan a dar la respuesta oportuna a lo que solicita el usuario, permitiéndole cumplir con su función de atención al usuario (Shumanov y Johnson, 2020).
EMOCIONALES	Son chatbots diseñados pararelacionarse y comprender a las personas que interactúan con ellos a través de conversaciones informales. Este tipo de chatbot permite al usuario recibir una atención personalizada que le permite sentirse cómodo hablando con el chatbot, expresando sus sentimientos y emociones y, a su vez, le permite al chatbot aprender sobre las emociones humanas y determinar la estructura de sus respuestas (Shumanov y Johnson, 2020).

Tabla 1. Clasificación de los tipos de asistentes virtuales

En relación con el desarrollo e implementación de asistentes virtuales en las organizaciones públicas y privadas, se pudieron identificar las siguientes características:

1. En cuanto al grado de interacción, actualmente utilizan asistentes dirigidos.
2. En cuanto a las funciones y finalidad, el que más predomina es el de tipo de atención al cliente.
3. En cuanto al medio de interacción, únicamente se ha empleado el de tipo texto (chatbot).
4. En cuanto al grado de afectividad, todos son no emocionales.

Por consiguiente, los asistentes implementados actualmente son de tipo chatbot, los cuales únicamente ofrecen un menú sobre determinados temas en el cual el usuario ingresa el numero acorde a la función o duda que tiene, este tipo de chatbots acorde a la opinión de diversos investigadores, carecen del uso de técnicas de inteligencia artificial y limitan la interacción entre el usuario y el chatbot (Ferrera, 2020).

Además de la improvisación, la mala selección de las características del asistente, limitan la implementación de los asistentes virtuales. Ya que es más complicado que cumplan en su totalidad los objetivos y/o propósitos por los cuales fueron desarrollados.

4. PROPUESTA DE ARQUITECTURA PARA ASISTENTES VIRTUALES BASADOS EN IA

El asistente virtual analiza la entrada de texto ingresada por el usuario para identificar la pregunta y luego responder al usuario con la respuesta más adecuada, la arquitectura del asistente virtual incluye 3 partes principales:

1. Comprensión del lenguaje natural (NLU)
2. Administrador de diálogo (DM)
3. Generador de lenguaje natural (NLG)

La Figura 1 muestra los pasos previos al desarrollo del asistente virtual para definir características y funciones relevantes del mismo.

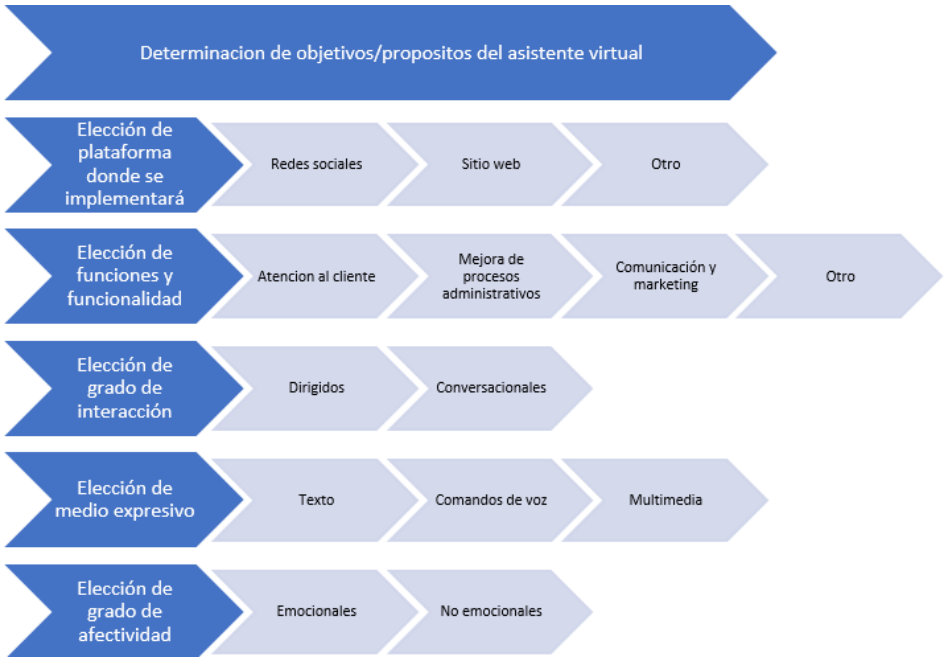

Figura 1: Selección de las características del asistente virtual
Fuente: Elaboración Propia

Para la selección del asistente virtual se deben tomar en cuenta los siguientes puntos:

1. Determinar los propósitos y objetivos del asistente.
2. Determinar la plataforma donde se implementará el asistente virtual.
3. A partir de la identificación de los propósitos y objetivos determinar las funciones y finalidad del asistente que permitirán llevar a cabo dichos propósitos.
4. Una vez identificadas las funciones se establecerá el grado de interacción más adecuado en función de los propósitos y objetivos del asistente.
5. Al identificar el grado de interacción podremos reconocer fácilmente el medio de interacción más adecuado.
6. En función a los propósitos y objetivos del asistente virtual establecer si será o no emocional el asistente.

A continuación, la Figura 2 presenta una arquitectura para la selección, desarrollo e implementación de un asistente virtual.

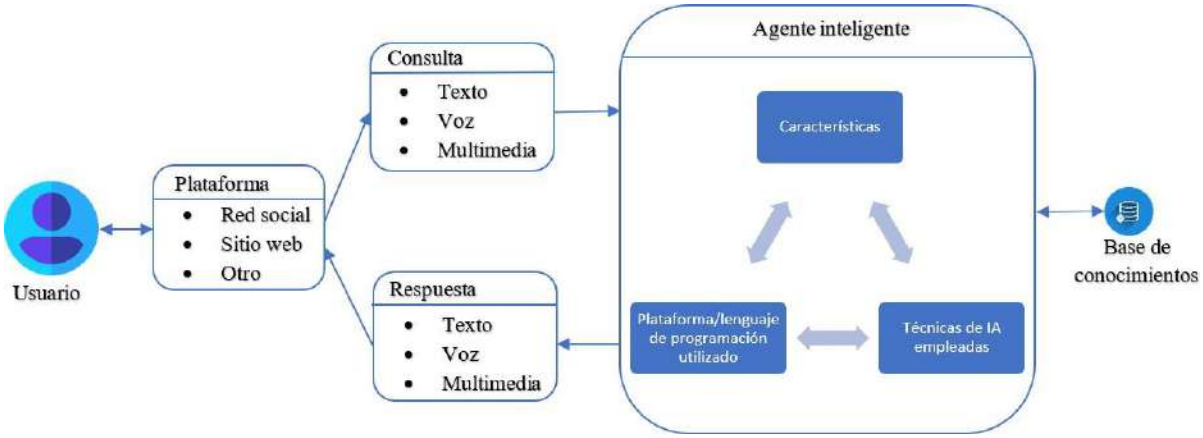

Figura 2: Arquitectura propuesta para el desarrollo de un asistente virtual

Fuente: Elaboración Propia

Una vez establecidas las características del asistente virtual, se deberán identificar las técnicas de inteligencia artificial que permitan cubrir con las características previamente establecidas. Finalmente, se evaluarán las plataformas y programas que permiten el desarrollo de asistentes virtuales, para identificar cual contiene las técnicas de inteligencia artificial requeridas, permitiendo hasta este punto iniciar el desarrollo del asistente virtual. A continuación, se presenta el proceso de funcionamiento básico de un asistente virtual:

1. El usuario accede a la página web o red social donde se encuentra el asistente virtual.
2. Se realiza una consulta o pregunta por medio de un mensaje de texto/voz/multimedia entrante.
3. El agente inteligente debe realizar un análisis sistemático que realice la interpretación de palabras, símbolos y expresiones.
4. A través de la comprensión del lenguaje natural, se debe realizar el procesamiento de la información expresada en lenguaje humano, donde se clasifican los datos que ingresan al usuario y se extrae el significado de estos.
5. El procesamiento del lenguaje natural debe encargarse de analizar el texto y extraer metadatos como palabras clave, conceptos, relaciones, etc. Como la entrada del usuario es una pregunta sobre un trámite, se genera la necesidad de acceso a NLG.
6. A través de la comprensión del lenguaje natural, se debe realizar el procesamiento de la información expresada en lenguaje humano, donde se clasifican los datos que ingresan al usuario y se extrae el significado de estos.
7. El procesamiento del lenguaje natural debe encargarse de analizar el texto/voz y extraer metadatos como palabras clave, conceptos, relaciones, etc. Como la entrada del usuario es una pregunta sobre un trámite, se genera la necesidad de acceso a NLG.
8. El administrador de diálogos recibirá los datos de NLG. El rol del administrador de diálogos es preparar las respuestas del sistema que se enviarán al generador de lenguaje natural. El administrador de diálogo adopta una estrategia basada en reglas, en conocimiento o en recuperación y accede a plantillas de intención y entidad. Dependiendo de la estrategia, elige la mejor plantilla para formar una respuesta significativa.
9. En caso de que la entrada del usuario no coincida con ninguna de las plantillas, la red neuronal y los componentes de recuperación de información en línea analizan la entrada. Los componentes de la red neuronal utilizan métodos de aprendizaje automático/aprendizaje profundo y los componentes de recuperación en línea buscan plantillas coincidentes en línea (incluso en las redes sociales).

La arquitectura anteriormente mostrada y descrita pretende guiar al lector en la selección de las características del asistente, las técnicas de inteligencia artificial a implementar y las plataformas o programas para su desarrollo, permitiendo así generar asistentes virtuales que brinden una mejor experiencia de usuario permitiendo cumplir con los propósitos y objetivos establecidos. La atención y el servicio al cliente son uno de los pilares de toda organización lo que permite proyectar una mejor imagen como empresa y/u organización.

El desarrollo de los asistentes virtuales (chatbots) enfocados a diferentes sectores de la sociedad tales como: educación, finanzas, medicina, política, atención al cliente, etc. Actualmente, según estadísticas, en el 2020 el uso mundial de estos asistentes fue de 4.200 millones de usuarios, y se prevé que para 2024 habrá alrededor de 8.400 millones de usuarios. Según el estudio de Juniper Research realizado en España (Aoki, 2020).

Los asistentes virtuales pueden reducir significativamente la carga administrativa de las organizaciones públicas y/o privadas, al mejorar la comunicación entre organizaciones y clientes en la prestación de servicios., Por esta razón, es importante contar con una arquitectura eficiente que sirva de base para el desarrollo de asistentes virtuales eficientes, que generen respuestas rápidas y adeudadas. Otras características que deberían tener este tipo de artefactos tecnológicos consisten en la interpretación correcta de las preguntas del usuario utilizando procesamiento del lenguaje natural y permitiendo una interacción natural con el usuario. De igual forma, el mejorar la experiencia de usuario y reducir los tiempos de respuesta tiene el potencial de brindar beneficios a las organizaciones relacionados con el servicio al cliente (Valverde y Ferreira, 2019).

Con el paso de los años se han logrado desarrollar diversas técnicas de inteligencia artificial que han potenciado y mejorado en gran medida los servicios de organizaciones públicas y privadas. Hoy en día la interacción empresa-cliente, mediante nuevas tecnologías, ha evolucionado tanto, que puede ser innecesaria la contratación de múltiples empleados para la atención al cliente. Los asistentes virtuales tienen el potencial de encargarse de esta labor (Valverde y Ferreira, 2019).

De acuerdo con los hallazgos del estado del arte, las técnicas de inteligencia artificial más utilizadas para el desarrollo de chatbots son: análisis de sentimientos, procesamiento del lenguaje natural, análisis de sentimientos, aprendizaje automático y computación afectiva (Miklosik, Evans, Ahmed, 2021).

Además, existen varias herramientas y plataformas que permiten el desarrollo e implementación de chatbots. Cada una de ellas tiene características distintas y brindan diferentes funcionalidades. La revisión sistemática de la literatura permitió identificar los aspectos relevantes de cada plataforma y saber cuáles brindan las mejores alternativas para el desarrollo de chatbots. Según Ravelo (2019) y Sandoval (2020) las plataformas para el desarrollo de chatbots son las siguientes: DigitalGenius, Aivo, Semantic Machines, Msg.ai, Twyla.ai, Pandorabots, It's Alive, Rebot.me, Chatterbot, Reply.ai, Gubshup, Microsoft LUIS, KITT.AI, FlowXO, Chatfuel, wit.ai, Live Agent, Pypestream y DialogFlow.

De acuerdo con la revisión sistemática de la literatura, las tecnologías utilizadas para el desarrollo de asistentes virtuales se basan en su mayoría en el uso de Dialogflow, que es una plataforma de comprensión del lenguaje natural utilizada para diseñar, desarrollar e integrar asistentes virtuales de tipo chatbot. Las aplicaciones de este tipo de chatbots se pueden utilizar en diversas plataformas como sitios web, redes sociales (Facebook, Telegram, WhatsApp, etc.) (Neumann, Guirguis y Steiner, 2022).

5. CONCLUSIONES

Si bien es cierto que la implementación de los asistentes virtuales en las oficinas de gobierno y empresas han mejorado la atención al cliente, también es cierto que aún queda mucho camino por recorrer. Como se mencionó anteriormente, en la actualidad los asistentes virtuales que predominan poseen la característica de ser asistentes virtuales guiados o dirigidos los cuales controlan en gran parte la interacción usuario-asistente virtual. En ciertos casos, esta situación, dificulta una mejora significativa en el servicio al usuario, dado que limitar la interacción con el usuario (mediante el uso de preguntas predefinidas y respuestas preconfiguradas con el uso de menús y/o mensajes) deja un vacío en la atención al cliente. Todo esto implica que los asistentes virtuales no sean completamente funcionales, ya que las respuestas predefinidas no ayudan al usuario en todas las dudas o consultas que requiere (Villagrasa, 2020).

Por lo anterior, es necesario la implementación de una arquitectura que sirva de guía en la selección de las características, técnicas de inteligencia artificial y plataformas o programas de desarrollo adecuadas para el desarrollo de asistentes virtuales acorde a las necesidades, objetivos y propósitos establecidos.

Esto lleva a la necesidad de migrar al uso de otras tecnologías que permitan crear asistentes virtuales conversacionales desde su diseño. Dichos asistentes potencialmente podrían interactuar de forma completamente natural con el usuario aprendiendo de cada una de las consultas que se realizan.

Hasta ahora, los asistentes virtuales desarrollados se enfocan en apoyar el servicio a los usuarios, dando respuestas a consultas simples, brindando información relevante. Estudios de Harvard afirman que el uso de asistentes virtuales puede ir más allá de la automatización de la atención al cliente, estableciendo otras funciones como la búsqueda de documentos y la búsqueda de pautas para el llenado de determinados formularios, la obtención de la opinión ciudadana sobre determinados temas, la traducción de la información gubernamental para hacerla más comprensible para los ciudadanos y redacción de documentos con respuestas a las preguntas de los ciudadanos (Shumanov y Johnson, 2020).

Las arquitecturas utilizadas para el desarrollo de asistentes virtuales empleadas, hasta el momento, tienen áreas de oportunidad y pueden mejorarse con el uso de diversas técnicas de inteligencia artificial que potencien la atención de los usuarios al realizar trámites, procesos y consultas de manera inteligente.

6. REFERENCIAS BIBLIOGRÁFICAS

Abdallah, A., Wahab, A., Fahad, A., & AlHamed, S. (2020). Developing an Intelligent Framework for Improving the Quality of Service in the Government Organizations in the Kingdom of Saudi Arabia. International Journal of Advanced Computer Science and Applications, 9.

AbuShawar, B., & Atwell, E. (2015). ALICE Chatbot: Trials and Outputs. Computación y sistemas, 9.

Aoki, N. (2020). An experimental study of public trust in AI chatbots in the public sector. Government Information Quarterly, 10.

Boden, M. (2016). Inteligencia Artificial. Madrid: Turner.

Makasi, T., Nili, A., Desouza, K., & Tate, M. (2021). A Typology of Chatbots in Public Service Delivery. Government Information Quarterly, 8.

McKenzie, J. (2021). PRISMA 2020 statement: an updated guide for the publication of systematic reviews. Rev Esp Cardiol, 9.

Miklosik, A., Evans, N., & Ahmed Qureshi, A. M. (2021). The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review. IEEE ACCESS, 10.

Neumann, O., Guirguis, K., & Steiner, R. (2022). Exploring artificial intelligence adoption in public organizations: a comparative case study. Public Management Review, 28.

Ferrera Uña, A. (2020). The smart chatbot revolution. Ciencia y Tecnologia, 29.

Shumanov, M., & Johnson, L. (2020). Making conversations with chatbots more personalized. Computers in Human Behavior, 7.

Ravelo Moreno, A. (2019). Desing and Development of a Chatbot. InfoTech, 16.

Rosário Valverde, M. S., & Ferreira Ferrão, A. (2019). Chatbot in the Online Provision of Government Services. AIS Electronic Library, 20.

Sandoval, N. (2020). Review on Implementation Techniques of Chatbot. Advancing Technology for Humanity, 5.

Villagrasa, O. C. (2020). La inteligencia artificial del sector público: desarrollo y regulación de la actuación administrativa inteligente en la cuarta revolución industrial. Revista de Internet, Derecho y Política, 14.

Esta obra está bajo una licencia de Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.

Recibido 10 Nov. 2022 *ReCIBE, Año 11 No. 2, nov. 2022*
Aceptado 14 Nov. 2022

Vida artificial y criaturas virtuales

Artificial life and virtual creatures

Rafael Mercado Herrera[1]
rmercadoh098@alumno.uaemex.mx

Vianney Muñoz-Jiménez[1]
vmunozj@uaemex.mx

Marco Antonio Ramos[1]
maramosc@uaemex.mx

[1]Universidad Autónoma del Estado de México

***Resumen*– Los organismos vivos son estructuras complejas que cumplen funciones como alimenta- ción, desarrollo y reproducción, todo esto inmerso en un ecosistema. La vida artificial busca generar enti- dades llamadas criaturas virtuales que imiten las propiedades de los organismos vivos y que realicen fun- ciones similares a las encontradas en la naturaleza, pero con ciertas restricciones. El área de las ciencias computacionales provee la capacidad de cálculo para reproducir ciertas propiedades de los organismos vivos y simularlas en ambientes controlados, llamados ambientes virtuales. Las simulaciones permiten acelerar la adaptabilidad de estos organismos sintéticos, a fin de comprender características más finas de estos fenómenos e incluso emplearlos para otros fines, tales como: investigación, desarrollo, diseño, simulación, esparcimiento, etc. Las funciones que realizan las criaturas virtuales son más simples que las de los organismos vivos, pero son similares, por ejemplo búsqueda de alimento (energía), reproduc- ción y evolución; esta similitud nos permite observar los posibles comportamientos de los seres vivos en ambientes controlados. Una criatura virtual no necesariamente debe tener una estructura morfológica conocida, pero si una que le permita realizar sus funciones o acciones definidas, por lo que, la criatura virtual debe contar con una cabeza, un cuerpo y extremidades que le permitan evolucionar de manera con- tinua para adaptarse al medio ambiente en el que se encuentra inmersa. Este artículo muestra una vista de los avances que se han hecho en el área de morfología de criaturas virtuales, desde su invención hasta la actualidad y los acercamientos empleados para dar una idea de cómo son modeladas.**

Palabras clave: *vida artificial; criaturas virtuales; morfología*

***Abstract*– Living organisms as we know them are complex structures that fulfill functions such as forag- ing, development, and reproduction, all immersed in an ecosystem. Artificial life seeks to generate entities called virtual creatures that mimic the properties of living organisms and perform functions similar to those found in nature but with certain restrictions. The area of computer science provides us with the ability to calculate and reproduce specific properties of living organisms, and be able to simulate them in controlled environments, called virtual environments. In virtual environments, it is possible to accelerate the adapt- ability of these synthetic organisms to understand the characteristics of these phenomena and even use them for other purposes, such as research, development, design, simulation, recreation, etc. Whereas the functions performed by virtual creatures are not as complex as those of living organisms, but they have to do actions such as foraging (energy), reproduction, and evolution. These actions allow us to generate an entity to observe the possible behaviors of living beings in controlled environments. A virtual crea- ture's morphological structure may not be known, but it enables the creature's functions or actions. So the virtual creature must have a head, a body, and limbs that allow it to evolve continuously to adapt to the environment in which it is immersed. This article shows an overview of the advances in the area of virtual creature morphology, from inception to actuality as well as the different approaches used to give the reader an idea of the modeling process for virtual creatures.**

Keywords: *artificial life; virtual creatures; morphology*

1 Introducción

El estudio de la vida como se conoce, es de suma importancia para los seres humanos, dado que con ello se comprende como estamos conformados y el entorno que nos rodea. Es gracias a estos estudios que emergen áreas como la medicina que busca el bienestar de los humanos, tratando patologías y enfermedades que afectan a los individuos. Pero es la biología que se encarga de estudiar los fenómenos de los seres vivos [Hills et al., 2020], gracias a ello se comprende con mayor detalle el ciclo de vida que enfrentan los seres vivos, como lo es el nacimiento, el desarrollo, la reproducción y la muerte. En donde para persistir en el entorno se debe obtener energía por medio de la nutrición y se requiere de constantes cambios para adaptarse a las condiciones del ecosistema.

Actualmente las ciencias de la vida se apoyan de las matemáticas y de las ciencias computacionales para simular funciones vitales de los seres vivos como la metabolización necesaria para la transformación y liberación de energía empleada en sus actividades esenciales. Es por este tipo de estudios y el trabajo colaborativo entre diversas disciplinas que surge el área de la vida artificial. La idea de retomar conceptos de la naturaleza y aplicarlos en las ciencias computacionales data de 1950 por Alan Turing [Turing and Haugeland, 1950] quien origina la propuesta de emplear el concepto de la evolución para dotar a una computadora a pensar por si misma. La evolución desde el enfoque computacional fue abordada desde múltiples puntos de vista a lo largo de los años, no fue hasta la década de los 70's cuando John Holland [Holland et al., 1992] propone un modelo de algoritmo genético; los algoritmos genéticos toman poblaciones de individuos a los que se les aplican operaciones similares a las que ocurren durante la evolución de los seres vivos, como son la selección de individuos, cruza (reproducción) y mutación, con la finalidad de proveerse de características necesarias para sobrevivir en el ambiente o en el entorno.

La posibilidad de poder reproducir de manera sintética los fenómenos y características de los seres vivos nos permiten describir el mundo que nos rodea y la posibilidad de generar artefactos que den soluciones a diversos problemas de nuestra vida cotidiana o incluso tareas más complejas donde la integridad de los humanos este comprometida, esto se realiza mediante mecanismos complejos que son tratados por la ingeniería. La posibilidad de crear vida sintética [Langton, 1986] permitiría comprender los diversos niveles de organización celular en los organismos vivos y en la construcción de artefactos que mejoren la calidad de vida de los seres humanos.

Christopher Langton [Langton, 1986] propuso que la vida artificial debe ocurrir primeramente en ambientes simulados y controlados, de tal forma que permitan identificar las características verdaderamente fundamentales para la vida, estos van desde su estructura que puede ir desde biomoléculas hasta la formación de tejidos, huesos y órganos, así como la capacidad de autorregularse para compensar internamente los cambios que ocurren en el ambiente. Esta capacidad es fundamental para la supervivencia y conservación de la especie.

En este trabajo hablaremos de criaturas virtuales simuladas, que cuentan con una estructura inicial básica (morfología), que les permiten interactuar en el medio ambiente en el que se encuentran inmersas. Dicha estructura inicial va evolucionando conforme a las necesidades de supervivencia de la criatura y la especie. Se presentan los acercamientos actuales en la construcción de criaturas virtuales con el objeto de observar las propiedades de los seres vivos y sus alcances.

2 Criaturas virtuales versus Morfología

La estructura de los seres vivos se forma con células, estas células forman tejidos; los tejidos forman los huesos, los cuales originan una morfología propia a cada organismo. Esta morfología es puesta a prueba en el medio ambiente la cual evoluciona a través de las generaciones.

El trabajo seminal de Karl Sims [Sims, 1991, Sims, 1994], parte de la construcción de morfologías básicas por medio de cubos que hace evolucionar en un ambiente virtual logrando que realicen funciones de movimiento, sus estudios dan origen a lo que ahora conocemos como criaturas virtuales. En la figura 1 se observan algunos ejemplos de las criaturas virtuales que logró generar. Conforme las criaturas van evolucionando, estas son capaces de realizar movimientos como caminar, nadar o saltar de acuerdo a las condiciones del medio ambiente.

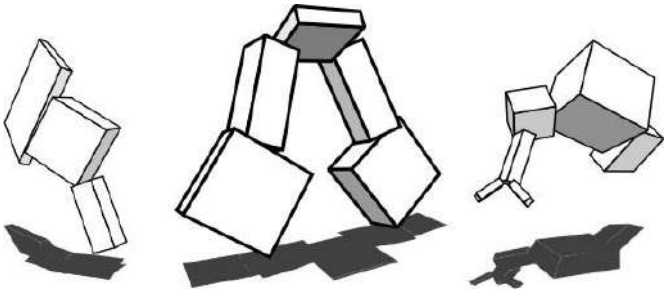

Figura 1. Criaturas virtuales generadas por Karl Sims [Sims, 1994]

Sims trabaja con la evolución de la morfología de sus criaturas para lograr criaturas adaptables al entorno, haciendo uso de la teoría de las especies, es decir, solo los mejores individuos o con mejores características son seleccionados como padres para generar una nueva criatura, en la figura 2 se muestra una criatura evolucionada. Las aportaciones realizadas por Sims permiten observar el potencial que tienen las criaturas virtuales en diferentes campos de estudio, tanto de la ciencia como de las actividades propias del ser humano.

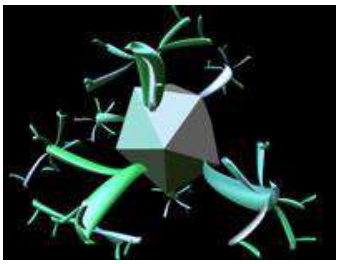

Figura 2. Ejemplo de una criatura evolucionada durante una instalación artística con participación del público llamada Galápagos [Sims, 1997].

Del mismo modo, Maciej Komosinski [Komosiński and Ulatowski, 1999, Komosinski, 2003] desarrolló un entorno de experimentación con criaturas virtuales llamado Framsticks, donde la estructura morfológica de las criaturas es un conjunto de cápsulas; en la figura 3 se muestra un ejemplo de las criaturas que se generan.

Figura 3. Criaturas virtuales generadas en Framsticks [Komosinski, 2003].

Si bien el entorno de pruebas presentado por Komosinsky fue presentado hace ya más de 20 años, hoy en día se sigue empleando para identificar una forma de cuantificar la similaridad morfológica entre criaturas virtuales.

Desde el punto de vista filosófico, recordemos que la aseveración que dió inicio a los estudios de vida artificial fue la capacidad de producir “vida” en un medio completamente sintético, tal que mimetice ciertas características de los fenómenos del mundo natural.

Entre los primeros trabajos con este punto de vista se encuentra el entorno virtual *Terra* presentado por Thomas Ray [Ray, 1993]. En *Terra*, las criaturas tienen como cuerpo bloques de código que se ejecutan por turnos y son capaces de hacer referencia a bloques de código de otras criaturas. En la figura 4 se muestran dos criaturas, sus segmentos de código y las referencias entre ellos.

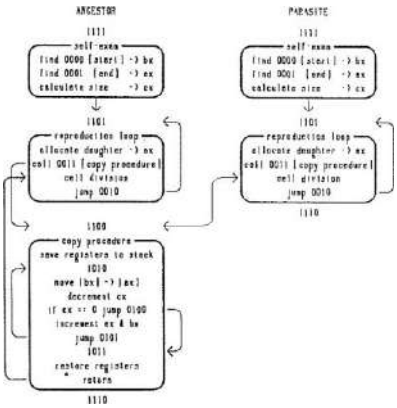

Figura 4. Criaturas virtuales del entorno *Terra* desarrollado por Thomas Ray [Ray, 1993], las criaturas son bloques de código que se ejecutan durante la simulación

El objetivo de Ray, se centró en demostrar la viabilidad de sintetizar vida en un entorno completamente artificial, tal que eventualmente sea posible proponer y comprobar nuevos fundamentos para la vida. En este entorno, Ray identificó interacciones complejas entre sus criaturas como es el parasitismo, donde una criatura roba el código de reproducción de otras criaturas para generar más copias de sí misma, tal como sucede hoy en día con los famosos virus informáticos.

Por otro lado, y en concordancia con la idea general de Langton, las criaturas virtuales también han sido utilizadas para identificar comportamientos y procesos en simulaciones similares a lo que se busca estudiar en la naturaleza, ya sea para verificar las capacidades de los modelos de criaturas virtuales que se proponen, o para obtener información de estos modelos que permitan explicar los fenómenos naturales.

Un ejemplo de este uso se observa en los trabajos de Takaya Arita et al. [Arita et al., 2016]. Su trabajo se centró en la evolución de morfología de las criaturas virtuales con un objetivo específico de forrajeo, buscar comida en el entorno. Durante los años que duró este proyecto pasaron del estudio de forrajeo de criaturas individuales a la competencia por alimento y las interacciones presa-depredador.

En dichos estudios se identificaron comportamientos de población en concordancia con modelos de biología y una evolución de la morfología de las presas para evitar ser comidas por los depredadores; en la figura 5 se muestran ejemplos de las presas con distintas morfologías defensivas. En esta propuesta, una criatura es comida (devorada) cuando su núcleo es tocado por un depredador, por lo que la evolución empujó a las criaturas a generar extremidades grandes para cubrirse, en ocasiones incluso favoreciendo la supervivencia sobre su capacidad de movimiento.

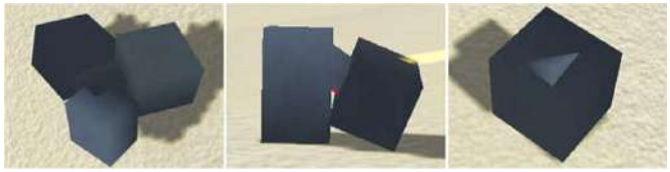

Figura 5. Estudio de coevolución de presas y depredadores con criaturas virtuales [Arita et al., 2016]. Se muestran tres tipos de presas que evolucionaron para evadir depredadores; generan extremidades grandes que bloquean a los depredadores de sus núcleos

Las criaturas virtuales también son empleadas para resolver problemas de ingeniería, como son el diseño de mecanismos y el control de robots. En la figura 6 se muestra uno de los usos actuales de criaturas virtuales en robótica; en esta, Kriegman et al. emplean la simulación de criaturas virtuales para evolucionar el diseño de*soft-robots*, donde su movimiento es dictado por cambios en el volumen de cada sección del robot y es resistente a efectos externos.

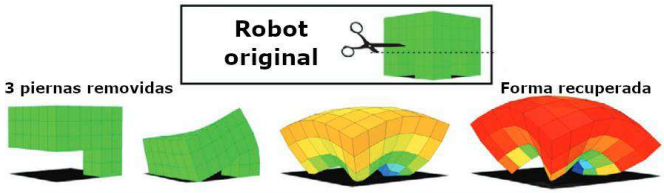

Figura 6. Una criatura virtual modelada para representar el diseño de un *soft-robot* [Shah et al., 2021]. Cada cubo de su cuerpo cambia de volumen (representado por los colores) y la conjunción de estos cambios genera movimiento en la criatura.

Como podemos observar la mayoría de los trabajos existentes basan la construcción de la criatura virtual en la morfología inicial y mediante mecanismos evolutivos esta se adecua a las condiciones del entorno.

3 Morfología y generación de criaturas virtuales

Una de las dificultades que se tienen en la generación de criaturas virtuales es definir cual será su morfología inicial, que sea el punto de partida que les permita evolucionar de manera similar a los seres vivos.

Karl Sims [Sims, 1994], propone un grafo dirigido donde cada nodo representa un elemento del cuerpo de la criatura y los arcos determinan la jerarquía de estos elementos; en la figura 7 se muestra un ejemplo de esta representación. La representación de Sims resulta intuitiva, sin embargo, tiene algunas limitantes en cuanto a la reproducción, ya que no es posible hacer evolucionar de forma sencilla la morfología de las futuras generaciones de criaturas virtuales.

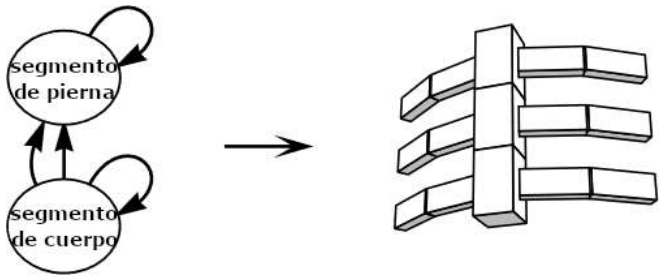

Figura 7. Modelo de representación de criaturas virtuales en [Sims, 1994]. Una gráfica dirigida donde la dirección de los arcos indica la jerarquía de los elementos del cuerpo de la criatura

El uso de grafos dirigidos permite la generación de criaturas virtuales de forma rápida permitiendo estudiar y simular propiedades especificas de los seres vivos, pero no una evolución completa. En la figura 8 se muestra una implementación de criaturas virtuales mediante grafos dirigidos utilizando el motor gráfico Unity3D. Es por esta razón que investigadores de todo el mundo realizan aportaciones sobre como representar la morfología inicial de las criaturas virtuales y haciendo uso de técnicas evolutivas puedan modificar la morfología inicial adaptándose al medio ambiente.

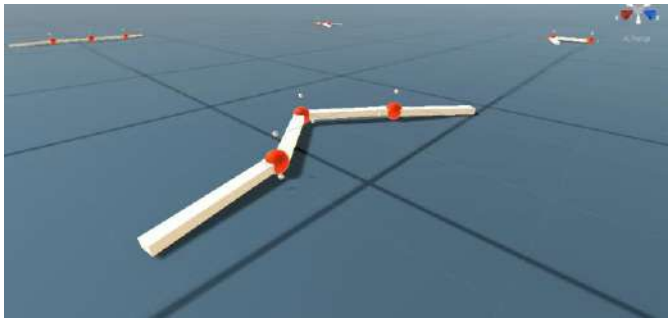

Figura 8. Generación de criaturas virtuales mediante grafos dirigidos en Unity3D, ViancoLab2021 [VianCoLab, 2021]

A fin de obtener una morfología inicial adecuada al momento de crear una criatura virtual, Kenneth Stanley propone los *Compositional Pattern Producing Networks (CPPN)* [Stanley, 2007] para modelar el desarrollo de la morfología de acuerdo a la posición celular. Esta propuesta se apega más a los conceptos de la vida y consiste en un conjunto de funciones interconectadas que reciben como argumento la posición actual y determinan la existencia o no, así como el tipo de elemento que existe en esa posición. Cheney et al. [Cheney et al., 2014] aplicaron este modelo a la generación de criaturas virtuales de la manera que se observa en la figura 9.

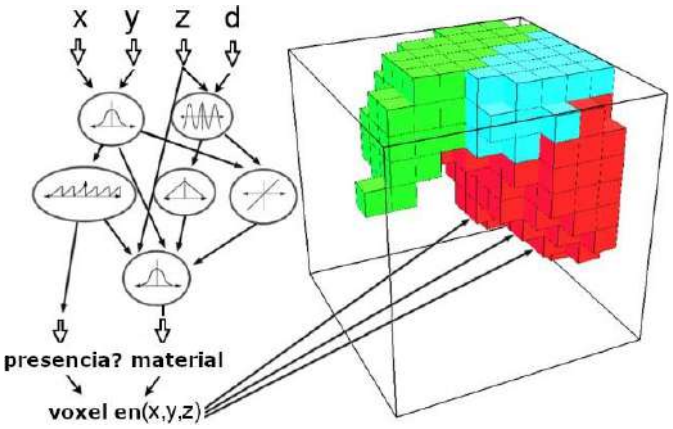

Figura 9. Aplicación de CPPN a la generación de criaturas virtuales propuesta por Cheney et al. [Cheney et al., 2014] para la experimentación en el diseño de *soft-robots*

Existen métodos modulares para la generación de criaturas virtuales, estos se basan en sistemas u órganos predefinidos por ejemplo; los ojos y mediante los mecanismos de evolución y presión del medio ambiente se define la cantidad de ojos y extremidades con las que contara la criatura virtual. Esto lo podemos apreciar en la figura 10, en ella observamos un grupo de criaturas virtuales con morfologías diferentes.

Figura 10. Aplicación de conceptos evolutivos para la generación de criaturas virtuales en SPORE™

Como podemos observar en los trabajos revisados, la construcción de criaturas virtuales es un problema abierto y propone retos interesantes que pueden ser tratados de forma multidisciplinar y las ciencias computacionales. En el trabajo de investigación que se está desarrollando actualmente se utilizan gramáticas para la generación de las morfologías iniciales de las criaturas virtuales, así mismo estas representan el código genético de la criatura, que será utilizado en futuras generaciones gracias al intercambio genético con otras criaturas virtuales.

El uso de gramáticas requiere de validaciones que permitan garantizar una carga genética suficiente para contar con una morfología inicial que pueda ser puesta a prueba y que pueda ser utilizada para las siguientes generaciones mediante mecanismos evolutivos. En la figura 11 se muestra un ejemplo de la carga genética y la morfología inicial de criatura virtual obtenida.

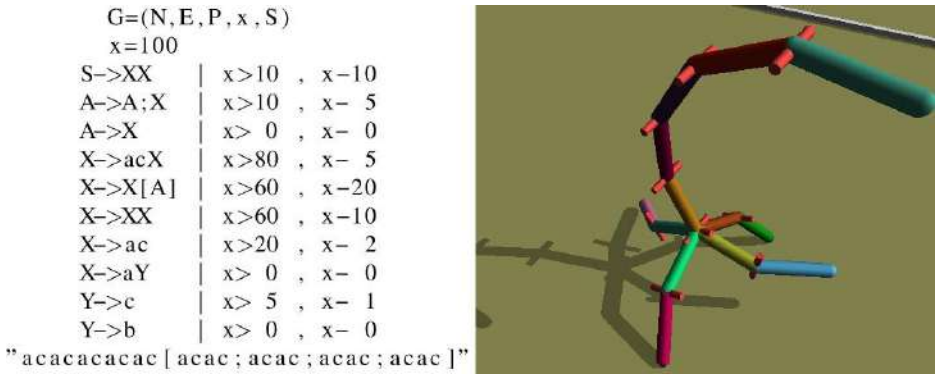

Figura 11. Una gramática y la morfología resultante de su aplicación. La palabra inicial *S* se sustituye de acuerdo a las reglas de producción y el resultado sigue el mismo proceso hasta que no haya reglas aplicables. Elaboración propia.

La originalidad de esta propuesta es el uso de premisas biológicas, una de estas es la "energía." Cada regla de producción cuenta con un requerimiento de energía para ser aplicable y un costo que reduce el valor de energía disponible para las siguientes reglas de producción.

Cada uno de estos modelos cuenta con sus peculiaridades y aplicaciones, además de su nivel de desarrollo y aceptación, tomando en cuenta que el área de criaturas virtuales es dinámica y se encuentra en constante crecimiento.

4 Conclusiones

El uso de las criaturas virtuales permite la construcción de artefactos con estructuras morfológicas que difícilmente pueden ser concebidas por los seres humanos. Un ejemplo de esto son los robot utilizados para la exploración del planeta Marte.

La posibilidad de poder generar diferentes morfologías a partir de las criaturas virtuales, permite poder explorar algunas soluciones en los diferentes sectores de la sociedad. Como por ejemplo en la medicina la creación de exoesqueletos para personas con un problema de movilidad.

El estudio de criaturas virtuales se encuentra en constante cambio y crecimiento, con propuestas que retoman conocimientos de múltiples áreas, haciendo este estudio multidisciplinario. La generación de criaturas virtuales nos permite observar la síntesis de la vida muy parecida a la que observamos en el mundo natural y con ellos simular el comportamiento de los seres vivos.

5 Agradecimientos

Los autores agradecemos al Consejo Nacional de Ciencia y Tecnología por su apoyo financiero para desarrollar el presente trabajo a través de su programa de becas nacionales.

Referencias

[Arita et al., 2016] Arita, T., Joachimczak, M., Ito, T., Asakura, A., and Suzuki, R. (2016). Alife approach to eco-evo-devo using evolution of virtual creatures. *Artificial Life and Robotics*, 21(2):141–148.

[Cheney et al., 2014] Cheney, N., MacCurdy, R., Clune, J., and Lipson, H. (2014). Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. *ACM SIGEVOlution*, 7(1):11–23.

[Hills et al., 2020] Hills, D. M., Heller, H. C., Berenbaum, M. R., and Hall, D. W. (2020). *Life: The Science of Biology*. W. H. Freeman, 12 edition.

[Holland et al., 1992] Holland, J. H. et al. (1992). *Adaptation in natural and artificial systems: an introductory analy- sis with applications to biology, control, and artificial intelligence*. MIT press.

[Komosinski, 2003] Komosinski, M. (2003). The Framsticks system: versatile simulator of 3D agents and theirevolution. *Kybernetes*, 32(1/2):156–173.

[Komosiński and Ulatowski, 1999] Komosiński, M. and Ulatowski, S. (1999). Framsticks: Towards a simulation of a nature-like world, creatures and evolution. In *European Conference on Artificial Life*, pages 261–265. Springer.

[Langton, 1986] Langton, C. G. (1986). Studying artificial life with cellular automata. *Physica D: Nonlinear Pheno-mena*, 22(1-3):120–149.

[Ray, 1993] Ray, T. S. (1993). An evolutionary approach to synthetic biology: Zen and the art of creating life. *Artificial Life*, 1(1_2):179–209.

[Shah et al., 2021] Shah, D., Yang, B., Kriegman, S., Levin, M., Bongard, J., and Kramer-Bottiglio, R. (2021). Shapechanging robots: bioinspiration, simulation, and physical realization. *Advanced Materials*, 33(19):2002882.

[Sims, 1991] Sims, K. (1991). Artificial evolution for computer graphics. In *Proceedings of the 18th annual confe-rence on Computer graphics and interactive techniques*, pages 319–328.

[Sims, 1994] Sims, K. (1994). Evolving virtual creatures. In *Proceedings of the 21st annual conference on Com-puter graphics and interactive techniques*, pages 15–22. ACM.

[Sims, 1997] Sims, K. (1997). Galapagos interactive exhibit by karl sims, 1997.

[Stanley, 2007] Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of develop-ment. *Genetic programming and evolvable machines*, 8(2):131–162.

[Turing and Haugeland, 1950] Turing, A. M. and Haugeland, J. (1950). *Computing machinery and intelligence*. MIT Press Cambridge, MA.

[VianCoLab, 2021] VianCoLab (Consultado el 05-Julio-2021). Donde la biología encuentra a la computación https://vianco.uaemex.mx.

Esta obra está bajo una licencia de Creative Commons
Atribución-NoComercial-CompartirIgual 4.0 Internacional.

Recibido 19 Jul. 2022 *ReCIBE, Año 11 No. 2, nov. 2022*

Aceptado 12 Sep. 2022

INTRODUCCIÓN A ENZO PARA SIMULACIONES HIDRODINÁMICAS EN ASTROFÍSICA

INTRODUCTION TO ENZO FOR HYDRODYNAMIC SIMULATIONS IN ASTROPHYSICS

Teófilo-Salvador Eduardo[1]

mca.ts.eduardo2015@gmail.com

Ambrocio-Cruz Patricia[1]

Rosado-Solís Margarita[2]

[1] Universidad Autónoma del Estado de Hidalgo, Unidad Tlahuelilpan, México.

[2] Universidad Nacional Autónoma de México, Instituto de Astronomía, México.

Resumen

Los códigos y algoritmos en el estudio de fenómenos físicos han sido de gran utilidad, sin embargo, poco ha sido la aplicabilidad en los últimos 20 años desde su aparición al desconocer las bondades y ventajas, además de la robustez y complejidad que implica la manipulación. El objetivo fue revisar, identificar y reconocer la estructura de Enzo, las formulaciones matemáticas base, condiciones de frontera, refinamiento, validación y verificación con énfasis hidrodinámico. De esto se revisaron las aplicaciones que se le han asignado al código en el estudio de procesos cosmológicos y astrofísicos. La evolución ha sido notable, considerando que prioritariamente el equipo original de trabajo que lo desarrollo ha continuado investigando, generando lazos académicos para la conservación y mejora de Enzo. Conocer y divulgar este tipo de códigos puede permitir en un futuro no solo aplicarlo a procesos astrofísicos, sino a climatología e ingeniería.

Palabras clave: código, solución numérica, malla adaptativa refinada

Summary

Codes and algorithms in the study of physical phenomena have been very useful, but their applicability has been limited over the last 20 years due to a lack of understanding of their benefits, advantages, as well as the robustness and complexity involved in their manipulation. The objective was to review, identify and recognize the structure of Enzo, its mathematical formulations, boundary conditions, refinement, validation, and verification with hydrodynamic emphasis. Applications assigned to the code in the study of cosmological and astrophysical processes were also reviewed. The evolution has been notable, considering that the original development team has continued researching and creating academic partnerships for the preservation and improvement of Enzo. Understanding and disseminating these types of codes could allow for their future application not only to astrophysical processes, but also to climatology and engineering.

Keywords: code, numerical solution, refined adaptive mesh.

1. Introducción

Actualmente se han desarrollado diversos códigos numéricos para el estudio de procesos cosmológicos y astrofísicos, algunos han evolucionado sustancialmente desde su aparición, otros se han mantenido y otros han servido de base para la creación y formulación de nuevos códigos. A su vez, estos códigos requieren de algoritmos escritos en lenguajes de programación, uso de librerías y dependencias, para simular numéricamente de forma ordenada un fenómeno para ciertos lapsos de tiempo, en casos astrofísicos Giga-años, que con el uso de supercomputadores el tiempo se reduce a miles de horas.

Enzo es un código cuya primera versión fue en 1996, desde entonces se han sumado unos 100 investigadores que estudian una gran variedad de fenómenos cosmológicos y astrofísicos (Dubey et al, 2014). El código contiene varios módulos para procesos específicos, como una estructura de malla adaptativa refinada, en coordenadas cartesianas para 1D, 2D y 3D, con alta resolución espacial y temporal en la modelación hidrodinámica de fluidos, dinámica de N-cuerpos, semigravedad de fluidos y partículas, gas, plasma, radiación, expansión y formación estelar (Bryan et al., 2014; Goldbaum, 2017). Las ecuaciones Eulerianas que resuelve Enzo son la de conservación de masa, de momento, de la energía total del fluido y la de inducción magnética respectivamente:

$$\frac{\partial \rho}{\partial t} + \frac{1}{a}\nabla \cdot \left(\rho \mathrm{v}\right) = 0 \qquad (1)$$

$$\frac{\partial \rho \mathrm{v}}{\partial t} + \frac{1}{a}\nabla \cdot \left(\rho \mathrm{vv} + \mathrm{I}p^* - \frac{\mathrm{BB}}{a}\right) = -\frac{\dot{a}}{a}\rho \mathrm{v} - \frac{1}{a}\rho \nabla \phi \qquad (2)$$

$$\frac{\partial E}{\partial t} + \frac{1}{a}\nabla \cdot \left[\left(E + p^*\right)\mathrm{v} - \frac{1}{a}\mathrm{B}\left(\mathrm{B}\cdot \mathrm{v}\right)\right] = -\frac{\dot{a}}{a}\left(2E - \frac{B^2}{2a}\right) - \frac{\rho}{a}\mathrm{v}\cdot \nabla \phi - \Lambda + \Gamma + \frac{1}{a^2}\nabla \cdot \mathrm{F}_{\mathrm{cond}} \qquad (3)$$

$$\frac{\partial \mathrm{B}}{\partial t} - \frac{1}{a}\nabla \mathrm{x}\left(\mathrm{vxB}\right) = 0 \qquad (4)$$

Donde E es la energía total del fluido, ρ la densidad del gas, v la velocidad peculiar y B la intensidad del campo magnético, I es la matriz idéntica, a el parámetro de expansión cosmológica, Λ es el enfriamiento radiativo, Γ calentamiento radiativo, y F_{cond} el flujo debido a la conducción de calor térmico.

Además:

$$E = e + \frac{\rho v^2}{2} + \frac{B^2}{2a} \qquad (5)$$

e es la densidad de la energía térmica, y la presión isotrópica se estima con:

$$p^* = p + \frac{B^2}{2a} \qquad (6)$$

p como la presión térmica. Además, la ecuación de estado y ecuación de Poisson para potencial gravitacional φ:

$$e = \frac{p}{(\gamma - 1)} \tag{7}$$

$$\nabla^2 \phi = \frac{4\pi G}{a}(\rho_{tot} - \rho_0) \tag{8}$$

$$\rho_{tot} = \rho_{gas} + \rho_{dm} + \rho_{stars} \tag{9}$$

Donde ρ_0 es la densidad promedio, además de considerar la densidad del gas, de materia oscura y de las estrellas. En el caso puramente hidrodinámico HD (B = 0) y sin expansión cosmológica (a = 1, a = 0). Adicionalmente, Enzo implementa la ecuación de flujo de calor isotrópico;

$$F_{cond} = -k_{cond} \nabla T \tag{10}$$

$$k_{cond} = f_{sp} k_{sp} \tag{11}$$

Así el coeficiente de conducción esta dado como una fracción de la conductividad de Spitzer y la temperatura del gas.

2. Metodología aplicativa

2.1 Refinamiento de malla adaptativa estructurada (SAMR)

Cada pieza o elemento observado es tratado como un objeto individual o como partícula. Las cuadrículas se organizan en jerarquía dinámica y se distribuyen por pieza en la malla. Cada cuadrícula se resuelve como un problema independiente con las condiciones de frontera de Dirichlet, almacenadas en zonas ghost, así al inicio de cada lapso de tiempo las celdas en un nivel determinado llenan las zonas ghost mediante interpolación, después las cuadrículas corrigen el flujo en el límite de la interfaz entre las zonas y proyectan los datos de la zona activa a la cuadrícula principal, mostrado en la Figura 1 y 2.

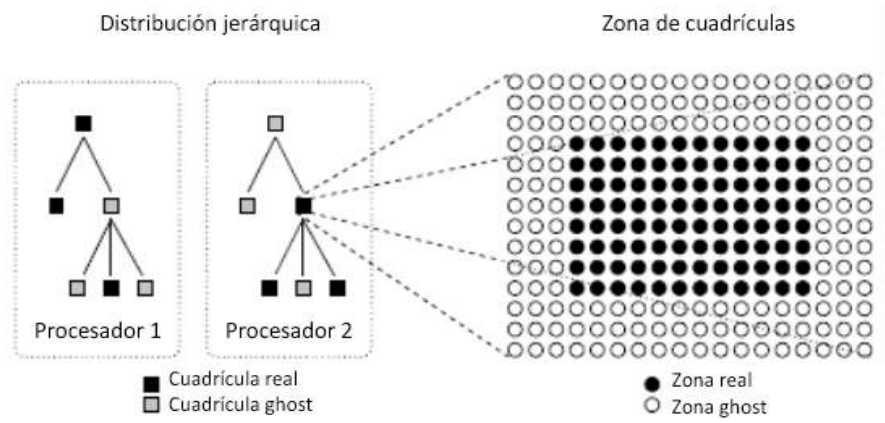

Figura 1. *Lado izquierdo distribución jerárquica AMR, a la derecha con malla real y ghost necesarias para sentencias hydro (modificado de Bryan et al., 2014).*

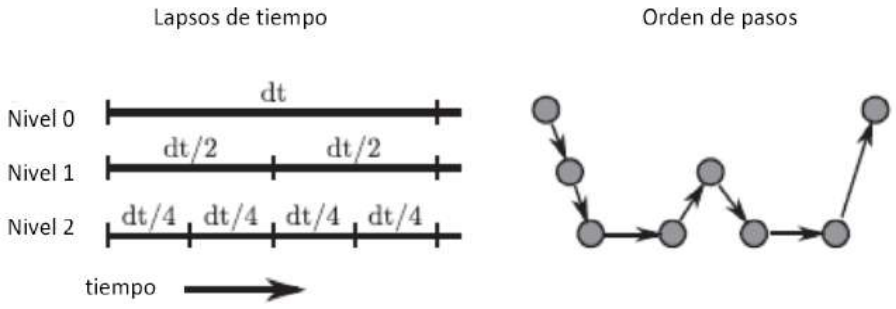

Figura 2. *Lado izquierda lapsos de tiempo de nivel 2 de jerarquía AMR, a la derecha orden en malla AMR para cada nivel (modificado de Bryan et al., 2014).*

2.2 Configuración de valores de frontera

Las condiciones de frontera para cada pieza o elemento son:

a) Reflecting: la frontera se comporta como un espejo q(-x) = q(+x) pero con velocidad normal a la frontera invertida $v_x(-x) = -v_x(+x)$.

b) Outflow: aproximación de las condiciones de frontera del flujo de salida duplicando la solución de dominio computacional q(-x) = q(0).

c) Inflow: los valores limite son fijados para una función predeterminada q(-x) = q0(-x,t). Enzo proporciona una forma de establecer valores de flujo de entrada constantes en el tiempo y espacio.

d) Periodic: la solución de frontera se obtiene de otro lado de la cuadrícula $q(-x) = q(x_{max} - x)$.

2.3 Solucionador de ecuaciones hidrodinámicas sobre la malla

- Para desarrollar la interpolación, las ecuaciones 1-3 se resuelven como flujo conservativo en las interfases y referencias de posición de las celdas.
- Es necesario un promedio pesado, tal que la conservación se puede restaurar al corregir las celdas gruesas localizadas fuera de la región fina.
- Se puede implementar un refinamiento adicional para determinar los límites que abarcan las celdas de interés.
- Para reconstrucción jerárquica, se localiza la región más pequeña, se calcula la eficiencia (> 30 %) como una relación entre estas celdas y el total.
- Si la región no es aceptable, se divide en dos, como una dimensión más larga y se estima el sumando de las celdas refinadas con el mismo índice, es posible utilizar diferencias finitas de tres puntos.
- Si no hay puntos de inflexión en el eje más largo, se prosigue con el siguiente, si no se corta a la mitad generando dos nuevas regiones o proto-subcuadrículas, y se aplica nuevamente a cada una de las nuevas regiones.
- El algoritmo anterior no es óptimo, pero ha demostrado un razonable y eficiente refinamiento de celdas, esto permite acomodar plantillas más grandes, además del acceso a máquinas de memoria más predecibles y lineales.

2.4 Criterios de refinamiento

a) Pendiente. Normalizada como $(q_{i+1} – q_{i-1})/(2q_i)$, con valor típico de 0.3.

b) Masa bariónica. Intenta mantener un resolución de masa fija, ($M_g = \rho(\Delta x)^d$) más grande que $\rho_{flag}(\Delta x_{root})^d r^{\epsilon l l}$, donde el primer término es la densidad equivalente sobre la malla raíz, r el factor de refinamiento, l el nivel y ϵl el parámetro de refinamiento (< 0 agresivo, > 0 menos agresivo que un refinamiento lagrangiano).

c) Choques. $(p_{i+1} – p_{i-1})/\min(p_{i+1}, p_{i-1}) > 0.33$, $u_{i-1} – u_{i+1} > 0$. $e_i/E_i > 0.1$, 0.33 y 0.1 determinados empíricamente.

d) Masa de partículas. Similar a la masa bariónica, pero utiliza densidad de partículas de la nube de celdas (materia oscura y densidad estelar).

e) Longitud de Jeans. Es necesario forzar para un número fijo de celdas.

f) Tiempo de enfriamiento. En simulaciones es un proceso rápido, y está dado por $t_{cool} = 1.5k_BT/(n\Lambda(T))$ y el tiempo de cruce del sonido, $t_{cross} = \Delta x/c_s$, con la velocidad del sonido.

g) Partículas imprescindibles. Partículas con fuerzas gravitatorias, con trayectorias sin colisión, esto permite un refinamiento para un nivel específico.

h) Cortante. Se pueden modelar inestabilidades de Kelvin-Helmholtz al utilizar el cortante como parámetro de refinamiento, y ocurre cuando las diferencias finitas cumplen la desigualdad:

$$\left(\frac{\partial u_x}{\partial y}+\frac{\partial u_y}{\partial x}\right)^2+\left(\frac{\partial u_z}{\partial y}+\frac{\partial u_y}{\partial z}\right)^2+\left(\frac{\partial u_x}{\partial z}+\frac{\partial u_z}{\partial x}\right)^2>\left(\epsilon_s\frac{c_s}{\Delta x}\right)^2 \qquad (12)$$

ϵ_s es adimensional y definido por el usuario.

i) Profundidad óptica. De una celda es: $\tau = \sigma_{HI} n_{HI} \Delta x$, con el primer término como sección eficaz de absorción de hidrógeno atómico neutro, la densidad propia de HI.

j) Longitud resistiva. Puede ser forzado para resolver escalabilidad para un número fijo de celdas.

k) Regiones refinadas. Para garantizar un refinamiento dada una simulación en un nivel.

l) Metalicidad. Definida como $Z \equiv \rho_z/\rho_b/0.022$, con densidad de metales, densidad bariónica (incluyendo metales).

3. Hidrodinámica: método parabólico por partes (PPM)

Incluyendo variantes: Reasignación (Lagrangiano) y Directo (Euleriano), donde este último es mejor para simulaciones AMR (Adaptative Mesh Refinement), además con las siguientes expresiones en su forma conservativa:

$$\frac{\partial\rho}{\partial t}+\frac{1}{a}\frac{\partial\rho v}{\partial x}=0 \qquad (13)$$

$$\frac{\partial\rho v}{\partial t}+\frac{1}{a}\frac{\partial\rho v^2}{\partial x}+\frac{1}{a}\frac{\partial p}{\partial x}=\rho\frac{g}{a} \qquad (14)$$

$$\frac{\partial \rho E}{\partial t} + \frac{1}{a}\frac{\partial \rho v E}{\partial x} = \rho v \frac{g}{a} \qquad (15)$$

Para el comportamiento de la velocidad del gas bariónico, estas ecuaciones ya se pueden resolver con PPM, primero se calcula el lado izquierdo y derecho que sean efectivos en el límite de la cuadrícula de las variables primarias u originales, después se promedian las regiones que corresponden a las distancias que puede moverse (v, v-c_s, v+c_s). Posteriormente se utiliza Riemann con los estados anteriores para poder estimar los flujos en función de la solución del mismo problema para poder actualizar las cantidades. Así las ecuaciones pueden ser rescritas de la forma:

$$\rho_j^{n+1} = \rho_j^n + \Delta t \left(\frac{\overline{\rho}_{j+1/2}\overline{v}_{j+1/2} - \overline{\rho}_{j-1/2}\overline{v}_{j-1/2}}{\Delta x_j} \right) \qquad (16)$$

$$\rho_j^{n+1} v_j^{n+1} = \rho_j^{n+1} v_j^n + \Delta t \left(\frac{\overline{\rho}_{j+1/2}\overline{v}_{j+1/2}^2 - \overline{\rho}_{j-1/2}\overline{v}_{j-1/2}^2 + \overline{p}_{j+1/2} - \overline{p}_{j-1/2}}{\Delta x_j} \right) + \frac{\Delta t}{2} g_j^{n+1/2} \left(\rho_j^n + \rho_j^{n+1} \right) \qquad (17)$$

$$\rho_j^{n+1} E_j^{n+1} = \rho_j^n E_j^n + \Delta t \left(\frac{\overline{\rho}_{j+1/2}\overline{v}_{j+1/2}\overline{E}_{j+1/2} - \overline{\rho}_{j-1/2}\overline{v}_{j-1/2}\overline{E}_{j-1/2} + \overline{v}_{j+1/2}\overline{p}_{j+1/2} - \overline{v}_{j-1/2}\overline{p}_{j-1/2}}{\Delta x_j} \right) + \frac{\Delta t}{2} g_j^{n+1/2} \left(\rho_j^n v_j^n + \rho_j^{n+1} v_j^{n+1} \right) \qquad (18)$$

Lo anterior muestra la discretización en el espacio, con zona centrada (j) y cara centrada (j+1/2) y los superíndices indican la posición en el tiempo, además el ancho de la celda Δx_j.

Adicionalmente, es necesario construir interpolaciones parabólicas por partes de tercer orden en 1D para la presión, densidad y velocidad.

La presión es determinada de la ecuación de estado (7), así la fórmula de interpolación para alguna cantidad de flujo q está dada por:

$$q_j(x) = q_{L,j} + \left(x \equiv \frac{x - x_{j-1/2}}{\Delta x_j} \right) \left[\Delta q_j + q_{6,j} \left(1 - x \right) \right] \qquad (19)$$

Para $x_{j-1/2} \leq x \leq x_{j+1/2}$, y $q_{L,j}$, Δq_j, $q_{6,j}$ pueden ser vistas como constantes de interpolación. Una vez reconstruidas las cantidades originales (ρ, p, v y E), se promedian sobre los dominios de v-c, v o v+c. Las ecuaciones lineales de la dinámica del gas son utilizadas para calcular el estado izquierdo y derecho de segundo orden.

Las variables con j ±1/2 permiten una precisión de tercer orden en el espacio y de segundo orden en el tiempo, y son utilizadas para resolver la diferencia de las ecuaciones para ρ^{n+1}, v^{n+1} y E^{n+1}.

El esquema tridimensional se logra dividendo por operadores y repitiendo el procedimiento anterior en las dos direcciones ortogonales. Las velocidades transversales y cualquier cantidad adicional se agregan al sistema.

3.1 Formulación de energía dual para flujos con altos números de Mach

- Lo desarrollado anteriormente funciona para sistemas gravitatorios con números de Mach < 100, siempre que las estructuras estén bien resueltas.
- Para flujos hipersónicos existe alta energía cinética E_k y energía interna del gas e con valores como de 10^8.
- La energía interna es muy pequeña comparada con la energía total. La presión es proporcional a E-Ek, esta diferencia no genera grandes errores en la distribución de presión en estas regiones, además la dinámica y el balance de energía del flujo no se ven afectados.
- Con la ecuación de la energía interna:

$$\frac{\partial e}{\partial t}+\frac{1}{a}\mathrm{v}\cdot\nabla e=-3\frac{\dot{a}}{a}\frac{p}{\rho}-\frac{p}{a\rho}\nabla\cdot\mathrm{v} \qquad (20)$$

Representando el transporte y la expansión del sistema, por lo que:

$$\rho_j^{n+1}e_j^{n+1}=\rho_j^n e_j^n+\Delta t\left(\frac{\overline{\rho}_{j+1/2}\overline{v}_{j+1/2}\overline{e}_{j+1/2}-\overline{\rho}_{j-1/2}\overline{v}_{j-1/2}\overline{e}_{j-1/2}}{\Delta x_j}\right)-\Delta t p_j^n\left(\frac{\overline{v}_{j+1/2}-\overline{v}_{j-1/2}}{\Delta x_j}\right) \qquad (21)$$

- No hay flujo conservativo, la presión se evalúa en el centro de la celda, pero es difícil de estimar sin generar grandes errores, lo que genera perturbaciones, por lo que la ecuación solo se usa en flujos hipersónicos.
- Es necesario combinar las dos formulaciones cuidando la energía interna ya que esta interviene en la dinámica del gas.
- En estas simulaciones se utiliza la energía total E para rutinas hidrodinámicas y la energía interna e para perfiles de temperatura.
- Cuando la presión se requiere para fines dinámicos se utiliza el criterio por celda:

$$p=\begin{cases}\rho(\gamma-1)\left(E-\mathrm{v}^2/2\right), & \left(E-\mathrm{v}^2/2\right)/E>\eta_1\\ \rho(\gamma-1)e, & \left(E-\mathrm{v}^2/2\right)/E<\eta_1\end{cases} \qquad (22)$$

- Si η_1 es muy pequeño el método de energía dual no tiene efectos dinámicos, y típicamente $\eta_1 = 10^{-3}$ para simulaciones en Enzo.
- Para los errores de advección, es necesario la vecindad de cada celda, de la forma:

$$e=\begin{cases}\left(E-\mathrm{v}^2/2\right), & \rho\left(E-\mathrm{v}^2/2\right)/\max\left(\rho_{j-1}E_{j-1},\rho_j E_j,\rho_{j+1}E_{j+1}\right)>\eta_2\\ e, & \rho\left(E-\mathrm{v}^2/2\right)/\max\left(\rho_{j-1}E_{j-1},\rho_j E_j,\rho_{j+1}E_{j+1}\right)<\eta_2\end{cases} \qquad (23)$$

- Así η_2 determina la sincronización de e con E, valores altos generan choques débiles, con posible calentamiento falso si es demasiado bajo. Un valor conservador opcional es $\eta_2 = 0.1$, ya que en simulaciones a gran escala el gas se enfría debido a la expansión, pero hay flujos a granel debido a la formación de estructuras.

3.2 Hidrodinámica: método de ZEUS

En este se incluye implementación del algoritmo hidrodinámico de diferencias finitas (ZEUS), el transporte de fluidos se resuelve en una cuadrícula cartesiana en esquema de advección monótona contra vientos, lo que genera una precisión de segundo orden en espacio y de primer orden en el tiempo. Como en PPM, se divide la solución de las ecuaciones en partes, y cada parte se evalúa sucesivamente utilizando los resultados precedidos. Además, la velocidad se registra en una cuadrícula escalonada en comparación con la densidad, la presión y la energía, que se encuentran en el centro de la celda. Una limitación al usar viscosidad artificial, es que mientras se logran las condiciones de salto de Rankine-Hugoniot, los choques se amplían a tres o cuatro celdas de la malla. Aunque es más robusto que PPM (Bryan et al., 2014).

3.3 Complementarios en hidrodinámica

Gravedad

1) Cálculo de la masa gravitatoria total: las partículas masivas (materia oscura y estrellas) se distribuyen en las rejillas con una interpolación de segundo orden. Se forma un campo de densidad discretizado, y las partículas avanzan temporalmente $0.5v^n\Delta t$. Se agrega la malla bariónica para obtener la masa gravitatoria total.
2) Resolución del campo de potencial gravitacional con las condiciones de frontera adecuadas: para el campo potencial de masa gravitatorio en la malla raíz, se puede usar la función de Green $-k^{-2}$ o el equivalente en diferencias finitas, mediante:

$$G(k) = -\frac{\Delta x}{2\left[\sin\left(\frac{k_x \Delta x}{2}\right)^2 + \sin\left(\frac{k_y \Delta y}{2}\right)^2 + \sin\left(\frac{k_z \Delta z}{2}\right)^2\right]} \qquad (24)$$

Donde $k^2 = k_x^2 + k_y^2 + k_z^2$ es el número de onda en el espacio, y el potencial se calcula con el ritmo de k como $\phi(k) = G(k)\rho(k)$. Para condiciones de frontera aisladas se usa el método de James.

Para potenciales más precisos en las subcuadrículas, Enzo usa una técnica similar pero ligeramente diferente a la malla raíz.

Las condiciones de frontera se interpolan a partir de los valores de potencia en la cuadrícula principal, la ecuación de potencial en cada subred se resuelve con las condiciones de frontera de Dirichlet. No se garantiza que las subredes vecinas generen los mismos valores de potencial debido a la falta de una solución que incluya toda la jerarquía de la red.

3) Diferenciación del potencial para obtener la aceleración, se puede interpolar, se permiten 4 iteraciones ya que no se resuelve el problema total (Bryan et al., 2014).

N-cuerpos

Se usa el método para malla de N-cuerpos o partículas para calcular la dinámica de los sistemas sin colisión, dado que en niveles astrofísicos la precisión requerida en las simulaciones genera formación de estructuras. Es necesario posiciones y velocidades de las partículas a partir de x^n y v^n utilizando una técnica estándar deriva-retroceso-deriva de la forma:

$$\mathrm{x}^{n+1/2} = \mathrm{x}^{n} + \frac{\Delta t}{2a^{n}}\mathrm{v}^{n}$$

$$\mathrm{v}^{n+1} = \mathrm{v}^{n}\left(1 - \frac{\dot{a}^{n+1/2}}{a^{n+1/2}}\right) + \frac{\Delta t}{a^{n+1/2}} g^{n+1/2}$$

$$\mathrm{x}^{n+1} = \mathrm{x}^{n+1/2} + \frac{\Delta t}{2a^{n+1}}\mathrm{v}^{n+1} \qquad (25)$$

Las partículas se almacenan en una cuadrícula más refinada, y las partículas que se mueven fuera cubren el volumen adyacente con una resolución más fina. Enzo proporciona un parámetro que gobierna el nivel máximo en el que se consideran las partículas como masas puntuales. En niveles más altos las contribuciones de las partículas al campo de masas gravitatorias se suavizan en una región esférica centrada de cada partícula.

Enfriamiento y calentamiento radiativo

Basado en cambios de energía, asumiendo que el gas se puede modelar ópticamente delgado o como una aproximación local al espesor óptico. Las reacciones que caracterizan a un gas libre de metales, se pueden estimar sin equilibrio aun en simulaciones grandes. El enfriamiento se resuelve calculando las velocidades de enfriamiento y calentamiento de los procesos individuales para el H y He atómicos: excitación e ionización por colisión, recombinación, emisión libre, dispersión. El enfriamiento radiativo este acoplado a la red química dentro del mismo bucle de subciclo. En el caso metagaláctico, este afecta al gas mediante fotocalentamiento y fotoionización, dependiente del corrimiento al rojo de las correspondientes ecuaciones químicas (Bryan et al., 2014).

4. Pruebas del Código Enzo

De acuerdo con Bryan et al. (2014*)* Enzo usa dos métodos: i) problemas de prueba que se pueden comparar con versiones anteriores, cambio de equipo, nuevo compilador o cambios en el código base y, ii) comparaciones directas con otros códigos de dinámica de flujos astrofísicos.

4.1 Verificación y validación

- En casos solo se requiere un archivo de entrada.
- En simulaciones cosmológicas toma condiciones iniciales creadas por otros códigos.
- Lo anterior permite verificar el comportamiento del código (estable en el tiempo, nuevas plataformas y modificaciones al código base).
- Permite ejecutar problemas de prueba y comparar resultados con otros producidos por otra versión del código.
- Hay una estructura organizada de directorios de acuerdo a la prueba (hidrodinámica, gravedad, enfriamiento, etc).
- Los problemas de prueba se agrupan en rápido (15 min), conjunto rápido (1 hr) y conjunto completo (60 hr), en equipos nuevos.
- Después de las simulaciones de prueba, el ejecutor de pruebas realiza una serie de tares y herramientas de análisis básicas.
- Los análisis incluyen cálculos estadísticos como intervalos, media y varianza, así los resultados se comparan con un conjunto de resultados estándar, además de poder almacenar localmente y comparar con otras versiones.
- Existen scripts para generar las figuras, también disponibles en el repositorio de Bitbucket de Enzo.
- Enzo se comporta de manera similar con otros códigos cosmológicos como GADGET SPH.
- Enzo tiene dificultades para la solución de problemas a pequeña escala de estructura gravitatorias.
- Funciona muy bien como un código basado en árboles para grandes estructuras por ser de orden superior (hidrodinámica PPM), por lo que es superior en la descripción de comportamientos a los otros códigos.
- Muestra buena resolución en fluidos y de choques para regiones de baja densidad en comparación con un código basado en partículas.
- Enzo y Gadget brindan resultados en métricas por ejemplo en el espectro de energía de la materia comparable en un margen del 5%.
- Presenta menores diferencias entre varios códigos, por ejemplo, en el gas y las masas de las partículas donde el nivel de Enzo es sustancialmente más alto.

5.0 Requerimientos

El link para acceder al código es https://enzo.readthedocs.io/en/latest/ en el cual se describe que es necesario contar con Fortran 90, C++, MPI parallel, CUDA support for Hydro, Librerias: HDF5, git e yt (visualización y análisis) y un Sistema Operativo como Linux, Mac OS X, AIX.

5.1 Actuales aplicaciones de Enzo

El código Enzo ha adoptado otras aplicaciones que también sirven de base para la solución de problemas físicos, astrofísicos y cosmológicos, como son: conducción térmica, formación de estrellas y retroalimentación, trayectoria de partículas, transporte radiativo y química, así como una combinación de la transformada rápida de Fourier para la solución de la ecuación de Poisson, en una cuadrícula de raíz periódica y en técnicas de redes múltiples como en subredes no periódicas (O´Shea et al., 2005).

Actualmente se han agregado nuevas características al código (Brummel-Smith et al., 2019), como son: compatibilidad con librerías de enfriamiento y química, nuevos algoritmos de refinamiento de malla adaptativa, presión de rayos cósmicos, difusión e inyección, modelación y cálculos de turbulencia, supernova sintética, supernova magnética, para partículas complejas un marco de partículas activas y evolución de materia oscura difusa. Enzo requiere de yt como un conjunto de herramientas de visualización y análisis para crear un entorno de consulta en diversos formatos (Turk, 2013; Turk et al., 2011).

Stewart et al. (2017) compararon Enzo, Art, Ramses, Arepo y Gizmo-PSPH para investigar el momento angular en galaxias del tamaño de la Vía Láctea de alta resolución, en cada uno implementando condiciones cosmológicas iniciales idénticas, las partículas de materia oscura se establecieron idénticas y se agregaron bariones en función de la distribución. Enzo ha evolucionado para el estudio en la exploración de formación de galaxias en el universo temprano, reionización del universo, examinación con alta resolución del entorno galáctico, impacto de hoyos negros supermasivos, turbulencia astrofísica, formación estelar, comportamiento de galaxias en el medio interestelar y deflagración de supernovas (Brummel-Smith et al., 2019). Salem et al. (2018) describieron semigravedad del gas, resaltando que la solución a un en regiones más complejas se obtiene refinada, además de la precisión del solucionador, ya que el momento angular se conserva mejor cuando el movimiento de rotación se alinea con algún eje de la cuadrícula. Boss y Keiser (2014) utilizaron el código para estudiar el colapso de nubes moleculares esferoidales con núcleos más condensados, donde las condiciones de frontera reflectantes se ajustaron mejor a la conservación de la masa y momento angular. Collins y Norman (2004) implementaron campos magnéticos al código debido a la necesidad del origen de la formación estelar y campos magnéticos en astrofísica moderna.

Burns et al. (2010) comparo observaciones con perfiles de temperatura utilizando Enzo, como un simple gas adiabático en regiones exteriores de cúmulos donde los tiempos de enfriamiento superan el tiempo de Hubble. Norman et al. (2009) describen una extensión del código en la que se incluye transporte de radiación ionizante con difusión aproximada, lo que genera una solución implícita acoplada de transporte de radiación, cinética de ionización y fotocalentamiento de gas, lo que hace que el paso de tiempo para este cálculo sea independiente. Forbes et al. (2016) utilizaron las condiciones iniciales que corresponden a galaxias enanas aisladas con un disco de gas, inicialmente laminar con perfil de densidad superficial cayendo exponencialmente, un halo caliente estacionario y partículas sin colisión a fin de representar estrellas y materia escura en la simulación de formación de estrellas. Latif et al. (2013) simuló remolinos de alta resolución siguiendo la evolución más allá de la formación de los picos, para reconocer la acumulación masiva central y fragmentación potencial. Padoan et al. (2006) utilizaron Enzo para probar un

método en la estimación del espectro de potencia de la turbulencia en nubes moleculares comparando los espectros en mapas de intensidad y mapas de velocidad.

Enzo refina en cualquier punto donde el gradiente de cantidades derivadas excede valores críticos, por ejemplo, al comparar con FLASH (Tasker et al.,2008), pero aplicado en casos más complejos como un choque esférico Enzo (con ajuste ZEUS) conduce a falta de conservación de energía y a una recuperación deficiente de la posición, perdida impulso con el tiempo. Schive et al. (2018) ha comparado la escalabilidad y rendimiento con GAMER-2 con simulaciones de galaxias de discos aislados y con FLASH en simulaciones de fusión de cúmulos de galaxias, donde GAMER-2 supera a Enzo aunque este tiene redes más grandes debido a la convergencia. En términos de escalabilidad es muy superior a otros códigos, el problema crítico es el solucionador de gravedad AMR que es un conjunto precondicionado de múltiples redes (Norman et al., 2018). Bordner y Norman (2018), indican que el código tiene lagunas limitaciones con respecto con su escalabilidad paralela, es decir no escala mucho más allá de unos pocos miles de núcleos al usar MAR, lo que limita su aplicabilidad, esto llevó a desarrollar Pentascale Enzo.

El código GRACKLE fue inicialmente incorporado a Enzo, los módulos del solucionador de enfriamiento fueron sencillos, lo que generó la extracción de la química y acondicionamiento de enfriamiento, aunque algunos procesos no están incluidos en las últimas versiones (Regan et al., 2017; Smith et al., 2017). Además, el solucionador MHD e hidráulico se han ajustado al marco de CUDA de NVIDIA, por lo que para grandes simulaciones supera las restricciones de memoria de las CPU actuales, gracias a la GPU (Gheller et al., 2015). Vazza et al. (2009) Implementaron un criterio de refinamiento de la malla adaptativa, diseñada explícitamente para aumentar la resolución espacial alrededor de discontinuidades en un campo de velocidades en simulaciones cosmológicas con Enzo, donde este criterio muestra una densidad de gas interna más baja, perfiles de entropía más planos, un balance de turbulencia mayor en todos los radios y un balance de energía térmica más grande en choques de acreción.

Schornbaum (2018) indica que en SAMR es importante analizar el aumento en el consumo de memoria debido a la replicación de datos, lo que genera cuellos de botella de escalas muy importante al utilizar Enzo. Dubey et al. (2014) indica que FLASH y Enzo requieren anualmente cientos de millones de horas en CPU en supercomputadoras administradas por organismos y agencias internacionales, aunque Enzo es un poco menos usado. Dubey et al (2014) indica que Enzo es similar a FLASH, pero con dificultad para optimizar solucionadores y equilibrio de cargas analizadas. Kitsionas et al. (2009) indican que en los códigos de cuadrícula al usar un parámetro de difusión más pequeño conduce a una menor disipación, dando como resultado un efecto de cuello de botella más grande, por ejemplo, al comparar Enzo con FLASH.

Finalmente, esta revisión del código Enzo ha permitido conceptualizar la estructura, bondades, ventajas y desventajas obtenidas por diversos investigadores que lo han utilizado. Sin embargo, esta introducción permite ampliar el conocimiento a un campo más grande como lo es en estudios de pregrado y posgrado en el área de ciencias astrofísicas, y posiblemente en un futuro a climatología e ingeniería aplicada, debido a que procesos no terrestres y terrestres presentan similitud, además de que las ecuaciones gobernantes presentan semejanza.

Agradecimientos

Se agradece al Consejo Nacional de Ciencia y Tecnología de México por los recursos financieros otorgados para la estancia posdoctoral con número 930457, en la Escuela Superior de Tlahuelilpan de la Universidad Autónoma del Estado de Hidalgo, así como al proyecto de Ciencias de Frontera CF-86367 y CY-253085 del CONACyT.

Referencias

Bordner, J. & Norman, M. L. (2018). Computational cosmology and astrophysics on adaptive meshes using charm++. arXiv:1810.01319v1 [astro-ph.IM]. https://doi.org/10.48550/arXiv.1810.01319

Boss, A. P. & Keiser, S. A. (2014). Collapse and fragmentation of magnetic molecular cloud cores with the Enzo AMR MHD code. ii. prolate and oblate cores. *The Astrophysical Journal, 794:*44, 1-14. Doi:10.1088/0004-637X/794/1/44

Brummel-Smith, C., Bryan, G., Butsky, I., Corlies, L., Emerick, A., Forbes, J., et al. (2019). ENZO: An adaptive mesh refinement code for astrophysics (version 2.6). *The Journal of Open Source Software, 14*(42). https://doi.org/10.21105/joss.01636

Bryan, G. L., Norman, M. L., O´Shea, B. W., Abel, T., Wise, J. H., Turk, M. J., et al. (2014). Enzo: an adaptative mesh refinement code for astrophysics. *The Astrophysical Journal Supplement Series, 2011:*19, 1-52.

Burns, J. O., Skillman, S. W. & O´Shea, B. W. (2010). Galaxy clusters at the edge: temperature, entropy, and gas dynamics near the virial radius. *The Astrophysical Journal, 721*:1105–1112. Doi:10.1088/0004-637X/721/2/1105

Collins, D. C. & Norman, M. L. (2004). Development of an AMR MHR module for the code Enzo. *American Astronomical Society Meeting 205, id.153.14; Bulletin of the American Astronomical Society, 36*, p.1605

Dubey, A., Almgren, A., Bell, J., Berzins, M., Brandt, S., Bryan, G., Colella, P., Graves, D., Lijewski, M., Loffler, F., o´Shea, B., Schnetter, E., van Straalen, B. & Weide, K. (2014). A survey of high level frameworks in block-structured adaptive mesh refinement packages. *J. Parallel Distrib. Comput.* http://dx.doi.org/10.1016/j.jpdc.2014.07.001

Dubey, A., Turk, M. J. & O´Shea, B. W. (2014). The impact of community software in astrophysics. *11th World Congress on Computational Mechanics (WCCM XI), 5th European Conference on Computational Mechanics (ECCM V), 6th European Conference on Computational Fluid Dynamics (ECFD VI)*, 1-8.

Forbes, J. C., Krumholz, M. R., Goldbaum, N. J. & Dekel, A. (2016). Suppression of star formation in dwarf galaxies by photoelectric grain heating feedback. *Letters Research, 0*, 1-10. Doi:10.1038/nature18292

Gheller, C., Wang, P., Vazza, F. & Teyssier, R. (2015). Numerical cosmology on the GPU with Enzo and Ramses. *XXVI IUPAP Conference on Computational Physics (CCP2014), Journal of Physics: Conference Series 640*, 1-6. Doi:10.1088/1742-6596/640/1/012058

Goldbaum, N. J. (2017). Extracting insights from astrophysics simulations. arXiv:1711.10373 [astro-ph.IM]. https://doi.org/10.48550/arXiv.1711.10373

Kitsionas, S., Federrath, C., Klessen, R. S., Schmidt, W., Price, D. J., Dursi, L. J., Gritschneder, M., Walch, S., Piontek, R., Kim, J., Jappsen, A. -K., Ciecielag, P. & MacLow, M. -M. (2009). Algorithmic comparisons of decaying, isothermal, supersonic turbulence. *A & A, 508,* 541-560. Doi: 10.1051/0004-6361/200811170

Latif, M. A., Schleicher, D. R. G., Schmidt, W. & Niemeyer, J. (2013). The formation of massive population iii stars in the presence of turbulence. *The Astrophysical Journal Letters, 772:*L3, 1-7. Doi:10.1088/2041-8205/772/1/L3

Norman, M. L., Reynolda, D. R. & So, G. C. (2009). Cosmological radiation hydrodynamics with Enzo. En I. Hubeny, J. M. Stone, K. MacGregor and K. Werner (Eds.), *Recent Directions in Astrophysical Quantitative Spectroscopy and Radiation Hydrodynamics* (260-272). American Institute of Physics 978-0-7354-0710-7/09

Norman, M. L., Smith, B. D. & Bordner, J. (2018). Simulating the cosmic dawn with Enzo. *Frontiers in Astronomy and Space Sciences, 5,* 1-21. Doi: 10.3389/fspas.2018.00034

O´Shea, B. W., Bryan, G., Bordner, J., Norman, M. L., Abel, T., Karkness, R. & Kritsuk, A. (2005). Introducing Enzo, an AMR cosmology application. En T. J. Barth, M. Briebel, D. E. Keyes, R. M. Nieminen, D. Roose and T. Schlick (eds), T. Plewa, T. Linde and V. G. Weirs (Eds.), *Adaptive Mesh Refinement – Theory and Applications* (341-350). Springer.

Padoan, P., Juvela, M., Kritsuk, A. & Norman, M. L. (2006). The power spectrum of supersonic turbulence in perseus. *The Astrophysical Journal, 653:* L125–L128.

Regan, J. A., Visbal, E., Wise, J. H., Haiman, Z., Johansson, P. H. & Bryan, G. L. (2017). Rapid formation of massive black holes in close proximity to embryonic protogalaxies. *Nature Astronomy, 1*, 1-6. Doi: 10.1038/s41550-017-0075

Salem, M., Besla, G., Bryan, G., Putman, M., van der Marel, R. P. & Tonnesen, S. (2018). Ram pressure stripping of the large magellanic cloud's disk as a probe of the milky way's circumgalactic medium. Draft Version.

Schive, H., ZuHone, J. A., Goldbaum, N. J., Turk, M. J., Gaspari, M., & Cheng, C. (2018). GAMER-2: a GPU-accelerated adaptive mesh refinement code – accuracy, performance, and scalability. *Monthly Notices of the Royal Astonomical Society, 481,* 4815-4840. Doi:10.1093/mnras/sty2586

Schornbaum, F. (2018). *Block-structured Adaptive Mesh Refinement for simulations on extreme-scale supercomputers.* Tesis Doctoral. Technischen Fakultät der Friedrich-Alexander-Universität Erlangen-Nürnberg. 152 p.

Smith, B. D., Bryan, G. L., Glover, S. C. O., Goldbaum, N. J., Turk, M. J., Regan, J., Wise, J. H., Schive, H., Abel, T., Emerick, A., O´Shea, B. W., Anninos, P., Hunmels, C. B & Khochfar, S. (2017). GRACKLE:

a chemistry and cooling library for astrophysics. *Monthly Notice of the Royal Astronomical Society, 466,* 2217-2234. Doi:10.1093/mnras/stw3291

Stewart, K. R., Maller, A. H., Oñorbe, J., Bullock, J. S., Joung, M. R., Devriendt, J., Ceverino, D., Kares, D., Hopkins, P. F. & Faucher-Giguere, C. (2017). High angular momentum halo gas: a feedback and code-independent prediction of LCDM. *The Astrophysical Journal, 843*:47, 1-15. https://doi.org/10.3847/1538-4357/aa6dff

Tasker, E. J., Brunino, R., Mitchell, N. L., Michielsen, D., Hopton, S.,Pearce, F. R., et al. (2008). A test suite for quantitative comparison of hydrodynamic codes in astrophysics. *Mon. Not. R. Astron. Soc., 390,* 1267-1281. Doi:10.1111/j.1365-2966.2008.13836.x

Turk, M. J. (2013). Scaling a code in the human dimension. *XSEDE '13: Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery.* https://doi.org/10.1145/2484762.2484782

Turk, M. J., Smith, B. D., Oishi, J. S., Skory, S., Skillman, S. W., Abel, T. & Norman, M. L. (2011). yt: a multi-code analysis toolkit for astrophysical simulation data. *The Astrophysical Journal Supplement Series, 192*:9, 1-16. Doi:10.1088/0067-0049/192/1/9

Vazza, F., Brunetti, G., Kritsuk, A., Wagner, R., Gheller, C. & Norman, M. (2009). Turbulent motions and shocks waves in galaxy clusters simulated with adaptive mesh refinement. *A & A, 504,* 33-43. Doi: 10.1051/0004-6361/200912535

Esta obra está bajo una licencia de Creative Commons
Atribución-NoComercial-CompartirIgual 4.0 Internacional.

COMPUTATION E INFORMATICA

Recibido 23 Oct.. 2022 *ReCIBE, Año 11 No. 2, nov. 2022*

Aceptado 03 Nov.. 2022

Inteligencia Artificial y Juegos de Tablero: Desde el Turco hasta AlphaZero

Artificial Intelligence and Board Games: From Turkish to AlphaZero

Ivan Francisco Valencia[1]
if.valencia19@gmail.com

Rosa Maria Valdovinos Rosas[1]

J. Raymundo Marcial Romero[1]

Roberto Alejo Eleuterio[2]

[1]Universidad Autónoma del Estado de México
[2]Tecnológico Nacional de México/IT Toluca, México, México

Resumen: El presente artículo tiene como objetivo mostrar la evolución del desarrollo de agentes inteligentes capaces de jugar juegos de tablero. Se muestra una breve revisión histórica de los agentes que se han desarrollado para diversos juegos y se describe el agente AlphaZero, creado por DeepMind, el cual hoy en día es el agente más avanzado en esta área es capaz de vencer a campeones humanos en el juego de Go, el cual se considera el juego de tablero más complejo que existe, incluso más que el ajedrez. De igual manera, se da una discusión sobre el paso natural que seguirán los agentes jugadores debido al éxito de AlphaZero y al surgimiento del paradigma de General Game Playing, el cual busca la creación de agentes capaces de jugar cualquier juego de tablero sin ninguna intervención humana.

Palabras clave: AlphaZero, Inteligencia Artificial, Juegos de Tablero.

Abstract: This paper has a goal to show the evolution of intelligent agents capable of play board games. Also, this paper gives a short historical review above agents that have been developed for various board games, and describes AlphaZero, developed by DeepMind, which today is the most advanced agent in this area, and it is capable of defeat human champions in the game Go which is considered the most complex game that exists even more than Chess. In the same way, a discussion is given about the next natural step that player agents will follow due to the success of AlphaZero and the rise of General Game Playing, which seeks create agents capable of play any board game without any human intervention.

Keywords: AlphaZero, Artificial Intelligence, Board Games.

1. Introducción

Desde sus inicios, la Inteligencia Artificial (IA) tiene como uno de sus retos desarrollar agentes inteligentes que sean capaces de jugar juegos de tablero al mismo nivel (o superior) que el ser humano (Björnsson & Schiffel, 2013).

El ánimo de desarrollar agentes jugadores se debe a que requieren características propias de la inteligencia humana como: deducción, razonamiento, resolución de problemas, búsqueda inteligente, representación del conocimiento, planificación, aprendizaje, creatividad, percepción y procesamiento de lenguaje natural, entre otras (Świechowski & Mańdziuk, 2015).

Hoy en día se han desarrollado agentes capaces de jugar a nivel de campeones humanos en ciertos juegos, por ejemplo: Chinook (Schaeffer, Lake, Lu, & Bryant, 1996) para Damas, Deep Blue (Campbell, Hoane, & Hsu, 2002) para el ajedrez, Dark Knight (Hsueh, Wu, Tseng, Yen, & Chen, 2015) para Banqi o ajedrez medio chino, y recientemente AlphaZero (Silver, y otros, 2016; Silver, y otros, 2017) para el juego de Go. De estos resalta AlphaZero, el cual al estar basado en técnicas de aprendizaje automático ha roto el paradigma del desarrollo de agentes y es capaz de desarrollar por sí mismo estrategias de juego con las cuales puede ganar a campeones humanos (Silver, y otros, 2016; Silver, y otros, 2017).

En este artículo se presenta una breve revisión histórica sobre el desarrollo de agentes capases jugar juegos de tablero antes y después de la IA. Igualmente se da una discusión del futuro de los agentes partiendo del éxito de AlphaZero, además de valorar el cómo pueden ser de utilidad para el desarrollo de agentes multidominio.

2. Revisión histórica sobre los agentes jugadores

Una inquietud del ser humano ha sido, el crear una máquina capaz de replicar su conducta, personalidad o acciones. Uno de los esfuerzos por lograr este objetivo se observa en las máquinas capaces de jugar juegos de tablero.

Uno de los primeros intentos de máquinas capaces de jugar juegos de tablero fue el autómata conocido como El Turco del año 1770 (Russell, 2010), el cual era un mecanismo que podía *jugar* Ajedrez de manera autónoma que, de acuerdo con los relatos, venció a Napoleón I en 1809 en la batalla de Wagram.

Sin embargo, años después de su creación se descubrió que el Turco, era un fraude, ya que realmente quien jugaba era un ser humano escondido dentro del autómata (ver la Figura 1).

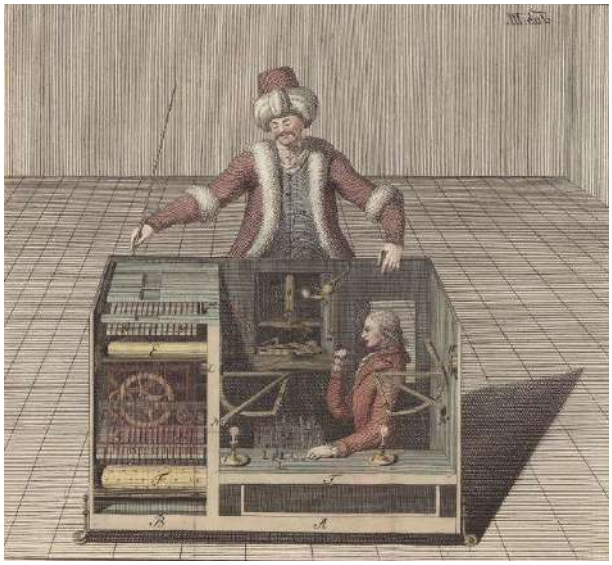

Figura 1. El Turco, autómata que jugaba ajedrez (El jugador de ajedrez de Maelzel, 2013).

En 1911, Leonardo Torres y Quevedo creó El Ajedrecista (Randell, 1982), considerada como la primera máquina electromecánica capaz de jugar ajedrez de manera autónoma. Aunque no era capaz de jugar por completo una partida, si podía jugar un final entre una Torre y un Rey blancas contra el Rey negro. El Ajedrecista jugaba blancas y siempre ganaba, aunque no necesariamente cumpliendo con la regla de los 50 movimientos.

En 1948, Alan Turing y David Champernowne desarrollaron el que se considera el primer programa capaz de jugar ajedrez: TuroChamp, el cual era capaz de vencer a aficionados del ajedrez (Dawson Jr, 2007). Sin embargo, derivado a la baja potencia de las computadoras de aquella época y la alta complejidad de los cálculos de TuroChamp, este no pudo ser programado en alguna computadora a excepción de un intento por parte Turing en la computadora Ferranti Mark I de la Universidad de Manchester. TuroChamp estableció métodos que aun hoy en día se usan para desarrollar agentes jugadores: el método MiniMax y la Búsqueda en Profundidad (Brange, 2021). Como celebración por los 100 años del natalicio de Turing, se realizó una partida de exhibición entre TuroChamp y el campeón Garry Kasparov la cual puede verse en (Turochamp (Computer) vs Garry Kasparov (2012), s.f.).

Para el año de 1949, Claude Shannon publica su artículo Programming a Computer for Playing Chess (Claude, 1950), donde muestra cómo se puede utilizar el método MiniMax en combinación con una función que evalúa las posiciones del tablero para poder construir un agente capaz de jugar ajedrez. Otra de las aportaciones de Shannon en el mismo artículo es la estimación de la Complejidad del Árbol (CA) de juego del ajedrez, dando una cota inferior de 10^{120} posiciones posibles desde la posición inicial que, a palabras propias de Shannon, una máquina que pruebe una posición por *micromicrosegundo* tardaría 10^{90} años en determinar su primer movimiento.

En 1997, IBM enfrentó su máquina de ajedrez Deep Blue (Campbell, Hoane, & Hsu, 2002) contra el campeón mundial Garry Kasparov. De seis partidas, Deep Blue ganó tres, empató dos y perdió solo una (puede ver el registro de las partidas en (Deep Blue (Computer) vs Garry Kasparov, s.f.)). Hasta ese momento ninguna computadora se consideraba capaz de vencer a campeones mundiales en juegos tan complejos como el ajedrez. Sin embargo, el impacto de la victoria de Deep Blue fue tal que jugadores profesionales creen que muchos de los movimientos realizados por Deep Blue fueron *muy humanos*.

Inspirado por el enfrentamiento Deep Blue y Kasparov, Omar Syed desarrolló el juego Arimaa (Syed & Syed, 2003) el cual se puede jugar haciendo uso del tablero y piezas del ajedrez. Syed se dio cuenta que muchas de las computadoras de aquella época hacían uso de algoritmos de fuerza bruta y de búsqueda en árbol, por ello y con la finalidad de dificultar a las computadoras el poder ganar a un ser humano, Syed creó a Arimaa con un Factor de Ramificación (FR) del árbol de juego de 17,281, siendo bastante alto en comparación al del ajedrez de solo 35. Este FR provoca que las computadoras no puedan analizar el árbol de juego más allá de unos pocos niveles y por ende sus movimientos no sean buenos. Syed en el 2009 ofreció un premio de $10,000 dólares a quien pudiera desarrollar un agente que, ejecutado en una computadora comercial, sea capaz de vencer a tres jugadores humanos en al menos tres enfrentamientos. Este reto permaneció vigente hasta abril del 2015, cuando el agente Sharp, que hace uso del método de poda Alpha-Beta, ganó siete partidas y perdió únicamente dos (Wu, 2015).

Si bien, el ajedrez es el juego que ha tomado los reflectores en cuanto al desarrollo de agentes, los esfuerzos no se han limitado a este. En Damas se tiene a Chinook (Schaeffer, Lake, Lu, & Bryant, 1996) agente que en los noventa venció al campeón mundial Marion Tinsley y al Gran Maestro Don Lafferty. En 2007, Schaeffer et al. (Schaeffer, y otros, 2007) mostraron que, en Damas, sí ambos jugadores juegan de manera perfecta, siempre se llegará a empate.

Othello es un juego de tablero en que se supera al ser humano, debido a su baja complejidad (ver Tabla 1), ya que en una partida se pueden realizar a lo máximo 60 movimientos. El primer programa en vencer a un campeón mundial fue The Moor que en 1980 venció al campeón Hiroshi Inouie (Buro, The evolution of strong othello programs, 2003). En 1997 Logistello venció al campeón mundial Takeshi Murakami reafirmando la superioridad de los agentes en Othello (Buro, The Othello match of the year: Takeshi Murakami vs. Logistello, 1997).

	Posiciones Legales	Factor de Ramificación	Complejidad de Árbol
Gato	10^{3}	4	10^{5}
Othello	10^{30}	10	10^{28}
Damas	10^{54}	4	10^{54}
Ajedrez	10^{43}	35	10^{123}
Arimaa	10^{43}	17281	10^{402}
Go	10^{172}	250	10^{360}

Tabla 1. Juegos y Complejidad

En el ámbito de los juegos que involucran azar podemos encontrar al agente TD-Gammon (Tesauro, 1994) para Backgammon. TD-Gammon está basado en un enfoque de aprendizaje por refuerzo. En un inicio únicamente realiza movimientos aleatorios, pero conforme juega contra sí mismo es capaz de *aprender* mejores jugadas, hasta el punto de alcanzar un nivel de juego intermedio e incluso, de acuerdo con sus creadores puede llegar a nivel maestro si se agrega conocimiento especifico (Tesauro, 1994).

Con el ánimo de incentivar el desarrollo de agentes en juegos diferentes al ajedrez, en 1989 se creó la Computer Olympiad (Schaeffer, The Computer Olympiads 1989--2021, 2021). Al día de hoy se han realizado 25 torneos, el último fue efectuado en julio del 2022. En este último torneo las competencias se realizaron en los siguientes juegos: Ajedrez Chino, Ajedrez Medio Chino, Ataxx, Breakthrough, Bridge, Conecta 6, Damas, Damas Brasileñas, Damas Canadienses, EinStein Würfelt Nicht!, Gomoku, Lines of Action, Mahjong, Nonogram, Othello 16x16, Othello 8x8, Santorini y Surakarta. Pueden verse los agentes participantes y los resultados de las competencias del 2022 en (Schaeffer, Computer Olympiad results, 2022). Los resultados de torneos de años anteriores pueden consultarse en (Computer Olympiad, 2022; Computer Olympiad (ICGA Tournaments), s.f.).

3. AlphaZero

Uno de los juegos que presenta más dificultades para los agentes es el juego de Go (ver la Figura 2). Este juego se considera sumamente complejo (Allis, 1994), desde el estado inicial Go tiene un numero de posiciones legales de $3^{361} \approx 10^{172}$, un FR de 250 y una CA de juego de 10^{360} (ver Tabla 1). Por lo anterior, hasta hace algunos años se creía que las computadoras no podían superar a los seres humanos en este juego.

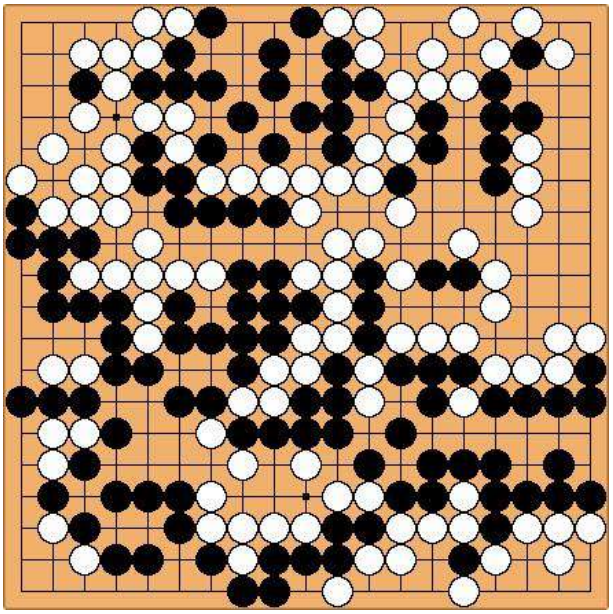

Figura 2. El juego de Go (Karigane Junichi at Sensei's Library, 2018)

En 2016, DeepMind propiedad de Google enfrentó su programa AlphaGo (Silver, y otros, 2016) contra el jugador profesional y campeón mundial Lee Sedol, quien confiado pronosticaba una victoria aplastante. Sin embargo, y para sorpresa de muchos, AlphaGo resultó ganador en cuatro de cinco partidas.

AlphaGo combina Redes Neuronales Profundas y Monte Carlo Tree Search (MCTS) de la siguiente forma (Silver, y otros, 2016):

- **Red Neuronal de Valor:** Tiene como objetivo reducir la profundidad de la búsqueda de árbol. Evalúa las posiciones del tablero y sustituye el paso de simulación en MCTS.
- **Red Neuronal de Política:** Tiene como objetivo reducir la amplitud de la búsqueda de árbol. Elige los movimientos a explorar en el paso de expansión de MCTS.

Las redes neuronales se entrenaron con millones de partidas de tomadas de Kiseido Go Server (KGS Go Server, s.f.), tanto de jugadores profesionales como de amateurs. Posteriormente AlphaGo se enfrentó a sí mismo para mejorar continuamente hasta obtener el nivel suficiente para vencer a campeones mundiales.

Una versión mejorada conocida como AlphaGo Zero (Silver, y otros, 2017) superó completamente a AlphaGo al derrotarlo en 100 partidas. El secreto de AlphaGo Zero radica en que, no hace uso de conocimiento humano, sino de aprendizaje automático además de combinar las dos redes neuronales en una sola. En un inicio AlphaGo Zero no tiene información alguna sobre cómo jugar, únicamente realiza movimientos al azar y conforme juega contra sí mismo, AlphaGo Zero se va mejorando. Con solo tres días de autoaprendizaje AlphaGo Zero fue capaz de vencer a AlphaGo. El hecho de no usar conocimiento humano permitió a AlphaGo Zero descubrir nuevas estrategias que no se habían contemplado en toda la historia de Go, por lo que jugadores profesionales se han inspirado en las jugadas de AlphaGo Zero para diseñar sus propias estrategias, lo que muestra que la IA lejos de ser una amenaza puede verse como una herramienta para potenciar la inteligencia humana.

Para 2017, DeepMind libera AlphaZero (Silver, y otros, 2017) una generalización de AlphaGo Zero, que elimina por completo el conocimiento especifico de los juegos que generalmente se usa para el desarrollo de los agentes y lo remplaza por una Red Neuronal Profunda en combinación con MCTS, con lo cual solo requirió de 24 horas de autoaprendizaje para vencer a campeones no humanos en tres juegos: ajedrez, ajedrez japones y Go. Por lo anterior AlphaZero ha definido un antes y un después en el desarrollo de agentes jugadores, pues puede considerarse el logro más importante en esta área.

4. Conclusiones: El futuro de los agentes jugadores

En este artículo se mostró cómo han evolucionado los agentes jugadores, desde ser simples mecanismos capaces de realizar jugadas simples hasta llegar a ser sistemas inteligentes capaces de sobrepasar por completo a los seres humanos.

Sin embargo, una crítica bastante recurrente sobre los agentes jugadores es determinar si estos son inteligentes o no, ya que las estrategias que estos usan son diseñadas por los investigadores o desarrolladores que los programan, dejando a los agentes como simples autómatas carentes de inteligencia cuya única tarea es ejecutar las ordenes que se les han dado, por ello surge la pregunta: ¿los agentes jugadores pueden ser llamados inteligentes si es una persona quien les da toda la lógica para jugar?.

Otra critica radica en la sobre especialización que estos agentes llegan a tener tomando, por ejemplo, a Deep Blue el cual es bastante competente a jugar ajedrez, este agente requeriría de grandes cambios o modificaciones para que sea capaz de jugar otro juego, aunque este sea más simple. Por ello surgen las siguientes preguntas: ¿los agentes podrán jugar igual de bien cualquier otro juego para el que no fueron diseñados? y ¿reflejan inteligencia si solo pueden jugar un único juego?

Hace algunos años se planteó el paradigma de General Game Playing (GGP) (Genesereth & Björnsson, The international general game playing competition, 2013; Genesereth, Love, & Pell, General game playing: Overview of the AAAI competition, 2005) en este se busca abordar las problemáticas antes mencionadas al desarrollar agentes capaces de jugar cualquier juego de tablero sin conocimiento previo y sin intervención humana.

En un inicio los agentes GGP (Genesereth, Love, & Pell, General game playing: Overview of the AAAI competition, 2005) usaban algoritmos como MiniMax, Alfa-Beta y Búsqueda en Profundidad. Sin embargo, los agentes basados en estos algoritmos resultaban bastante ineficientes. No fue hasta la llegada del método MCTS (Browne, y otros, 2012), y su característica de ser independiente de dominio, que permitió a los agentes aumentar en su rendimiento, definiendo el estado del arte de GGP.

Como ejemplo de los agentes desarrollados bajo este paradigma tenemos a CadiaPlayer (Bjornsson & Finnsson, 2009) el cual ganó The International General Game Playing Competition, en los años 2005, 2007, 2008 y 2012. En este último año CadiaPlayer, además, venció a un jugador humano (Genesereth, Love, & Pell, General game playing: Overview of the AAAI competition, 2005). Por lo anterior se puede decir que CadiaPlayer es el primer agente jugador multi-juego que ha tenido un relativo éxito.

Sin embargo, con la llegada de AlphaZero y su enfoque basado en autoaprendizaje se ha logrado un avance enorme en el desarrollo de agentes jugadores multi-juego. Se puede afirmar que AlphaZero ha dado una solución a las críticas realizadas a los agentes jugadores ya que:

- AlphaZero no requiere del conocimiento previo del juego ni intervención humana y es capaz de desarrollar sus propias estrategias al jugar contra sí mismo.
- AlphaZero ha aprendido a jugar en tres juegos: Ajedrez, Ajedrez Japones y Go, por lo que se considera el primer agente jugador multi-juego que ha superado al ser humano.

Aunque AlphaZero es un gran logro, el área del desarrollo de agentes jugadores aun es basto, hasta el momento los esfuerzos se han enfocado en juegos de tablero competitivos para dos jugadores y con movimientos por turnos, como lo es el ajedrez. Existen otros juegos de tablero que pueden reflejar escenarios de la vida real como lo son los juegos cooperativos, los de más de dos jugadores, los de movimientos simultáneos o los de información parcial. Incluso un área de interés puede ser el desarrollo de agentes capaces de jugar videojuegos (Levine, y otros, 2013; Schrittwieser, y otros, 2020)ya que presentan escenarios retadores al tener un mayor número de posibles acciones a realizar y la toma de decisiones debe ser prácticamente en tiempo real.

No cabe duda de que abordar estos escenarios permitirá desarrollar nuevos algoritmos y/o métodos aplicables en otras áreas como: la planificación, la gestión de procesos, el comercio electrónico, las operaciones militares, la conducción de vehículos autónomos, en general en escenarios dinámicos, no predecibles y con alta incertidumbre. Con esta revisión histórica se muestra que el área del desarrollo de agentes jugadores aun es un campo frutífero cuyos avances pueden llevar al desarrollo de agentes multidominio y multipropósito acercándonos cada vez más a la IA General la cual emule a la inteligencia humana.

Referencias

Allis, L. (1994). *Searching for solutions in games and artificial intelligence.*

Björnsson, Y., & Schiffel, S. (2013). Comparison of GDL reasoners. *Proceedings of the IJCAI-13 workshop on general game playing (GIGA'13)*, 55-62.

Brange, H. (2021). Evaluating Heuristic and Algorithmic Improvements for Alpha-Beta Search in a Chess Engine.

Buro, M. (1997). The Othello match of the year: Takeshi Murakami vs. Logistello. *ICGA Journal, 20*(3), 189-193.

Buro, M. (2003). The evolution of strong othello programs. *Entertainment Computing*, 81-88.

Campbell, M., Hoane, A., & Hsu, F.-h. (2002). Deep blue. (Elsevier, Ed.) *Artificial intelligence, 134*(1-2), 57-83.

Claude, E. (1950). Programming a computer for playing chess. *Philosophical Magazine, Ser*, 637656.

Computer Olympiad (ICGA Tournaments). (s.f.). Recuperado el 16 de octubre de 2022, de ICGA Tournaments.

Computer Olympiad. (14 de marzo de 2022). Recuperado el 16 de octubre de 2022, de Chessprogramming Wiki: https://www.chessprogramming.org/Computer_Olympiad

Dawson Jr, J. (2007). The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Ingelligence, and Artificial Life plus The Secrets of Enigma. *Review of Modern Logic, 10*(3-4), 179-181.

Deep Blue (Computer) vs Garry Kasparov. (s.f.). Recuperado el 16 de octubre de 2022, de Chessgames.com: https://www.chessgames.com/perl/chess.pl?pid=29912&pid2=15940

El jugador de ajedrez de Maelzel. (30 de mayo de 2013). Recuperado el 16 de octubre de 2022, de https://ztfnews.wordpress.com/2013/05/30/el-jugador-de-ajedrez-de-maelzel/

Genesereth, M., & Björnsson, Y. (2013). The international general game playing competition. *AI Magazine, 34*(2), 107-107.

Genesereth, M., Love, N., & Pell, B. (2005). General game playing: Overview of the AAAI competition. *AI magazine, 26*(2), 62-62.

Hsueh, C.-H., Wu, I.-C., Tseng, W.-J., Yen, S.-J., & Chen, J.-C. (2015). Strength improvement and analysis for an MCTS-based Chinese dark chess program. *Advances in Computer Games*, 29-40.

Karigane Junichi at Sensei's Library. (08 de septiembre de 2018). Recuperado el 16 de octubre de 2022, de Sensei's Library: https://senseis.xmp.net/?Karigane

KGS Go Server. (s.f.). Recuperado el 16 de octubre de 2022, de The KGS Go Server: https://www.gokgs.com

Randell, B. (1982). From analytical engine to electronic digital computer: The contributions of Ludgate, Torres, and Bush. *Annals of the History of Computing, 4*(4), 327-341.

Russell, S. (2010). *Artificial intelligence a modern approach.* Pearson Education, Inc.

Schaeffer, J. (2021). The Computer Olympiads 1989--2021. *ICGA Journal*, 1-4.

Schaeffer, J. (23 de agosto de 2022). *Computer Olympiad results*. Recuperado el 16 de octubre de 2022, de International Computer Games Association: https://icga.org/?p=3721

Schaeffer, J., Burch, N., Bjornsson, Y., Kishimoto, A., Muller, M., Lake, R., . . . Sutphen, S. (2007). Checkers is solved. *science, 317*(5844), 1518-1522.

Schaeffer, J., Lake, R., Lu, P., & Bryant, M. (1996). Chinook the world man-machine checkers champion. *AI Magazine, 17*(1), 21-29.

Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Van Den Driessche, G., . . . Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *nature, 529*(7587), 484-489.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., . . . Hassabis, D. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., . . . Hassabis, D. (2017). Mastering the game of go without human knowledge. *nature, 550*(7676), 354-359.

Świechowski, M., & Mańdziuk, J. (2015). Specialized vs. multi-game approaches to AI in games. *Intelligent Systems'2014*, 243-254.

Syed, O., & Syed, A. (2003). Arimaa-a new game designed to be difficult for computers. *ICGA Journal, 26*(2), 138-139.

Tesauro, G. (1994). TD-Gammon, a Self-Teaching Backgammon Program, Achieves Master-Level Play. *Neural Computation, 6*(2), 215-219.

Turochamp (Computer) vs Garry Kasparov (2012). (s.f.). Recuperado el 16 de octubre de 2022, de www.chessgames.com: https://www.chessgames.com/perl/chessgame?gid=1670503 Wu, D. (2015). Designing a winning Arimaa program. *ICGA Journal, 38*(1), 19-40.

Esta obra está bajo una licencia de Creative Commons
Atribución-NoComercial-CompartirIgual 4.0 Internacional.