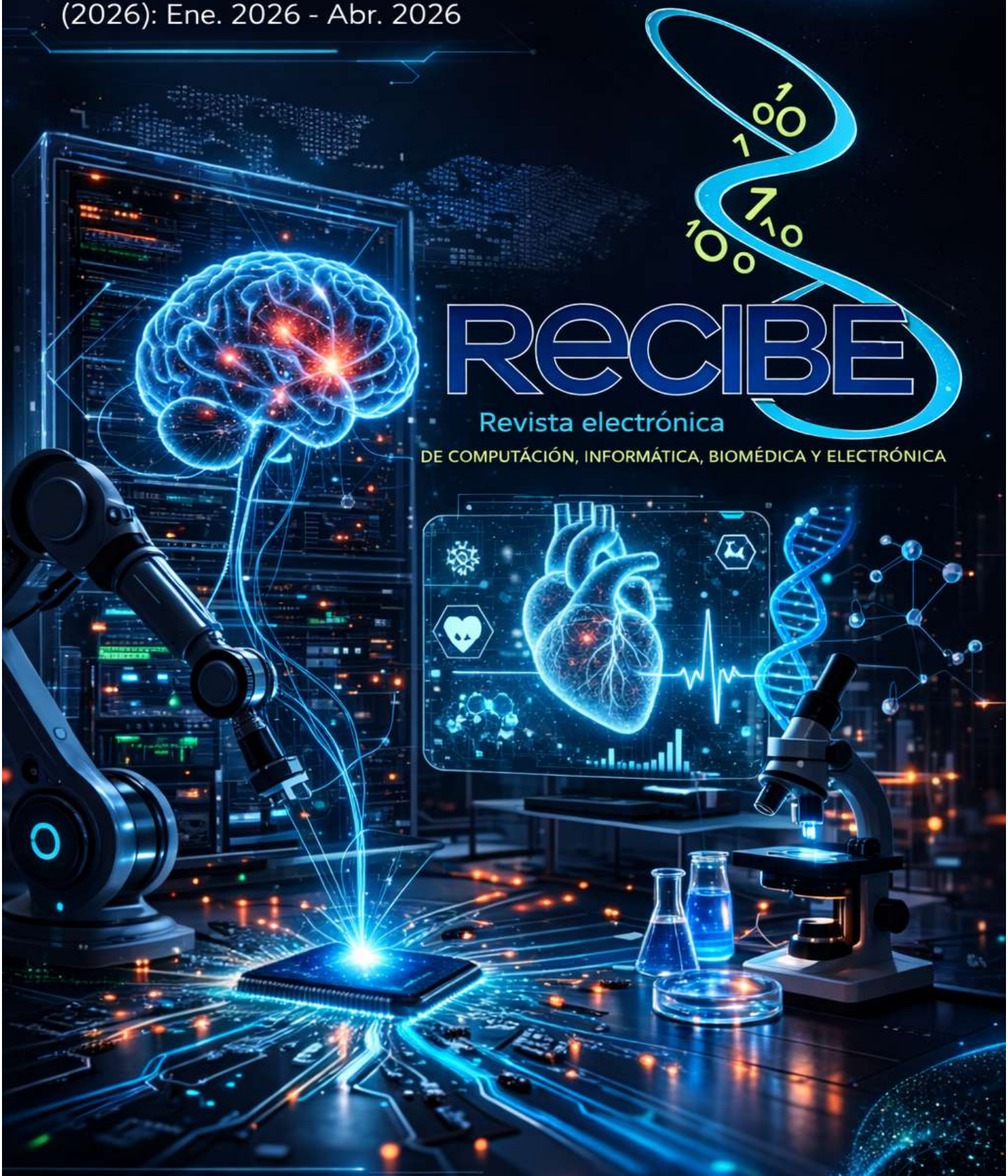
VOL. 15 | NÚM. 1
(2026): Ene. 2026 - Abr. 2026
ISSN: 2007-5448



RECIBE

Revista electrónica
DE COMPUTACIÓN, INFORMÁTICA, BIOMÉDICA Y ELECTRÓNICA

REVISTA ELECTRÓNICA
DE COMPUTACIÓN, INFORMÁTICA, BIOMÉDICA Y ELECTRÓNICA

Índice

Computación e Informática

Modelado Basado en Agentes Aplicado a Estrategias de Evacuación **C1-1**

Jorge De Jesús Galvez Rodríguez, Miguel Ángel Alejandro Islas Toski, Karla Avila Cardenas, Héctor Joaquin Escobar Cuevas

Método Eye tracking en la detección de somnolencia en conductores: Una revisión de literatura **C2-1**

Martin Laguna Estrada, José Manuel Delgado Pérez, Norma Verónica Ramírez Pérez, Luis Alberto López Gonzalez, Juan Ignacio Cerca Vázquez, José Manuel Malagon Soldara

Modelo de sostenibilidad humana para productos de software con infraestructura limitada: Caso Providencia - Colombia **C3-1**

Paola Noreña, Elizabeth Suescún, Santiago Manco, Santiago Idárraga

Prácticas de Ciberseguridad en la Ingeniería de Software: Una Revisión de la Literatura **C4-1**

Eduardo Antonio Castillo Garrido, Juan Carlos Pérez-Arriaga, Héctor Xavier Limón-Riaño, Saúl Domínguez-Isidro

Técnicas de Análisis de Supervivencia para la Identificación de Deserción en Estudiantes: Una Revisión Sistemática de la Literatura **C5-1**

Franz Jesús Rivera-Alcántara, Ángel J. Sánchez-García, José Juan Muñoz-León, Jorge Octavio Ocharán-Hernández, José Luis Sánchez-Leyva

Evaluación de flexibilidad para microservicios basado en atributos de calidad **C6-1**

Said Misael Venoso Lara, Juan Carlos Rojas Pérez, Olivia Graciela Fragoso Díaz

Roboto: Camino a la Justicia. Un videojuego educativo para la autoprotección infantil en contextos judiciales **C7-1**

Daniel Rodríguez Orozco, Luis Manuel Palma Pinto, Santiago Efrain Itzincab Poot, Víctor Hugo Menéndez Domínguez, Reyna Faride Peña Castillo

Recibido 7 Jul 2025 *ReCIBE, Año 15 No. 1, mayo 2026*

Aceptado 25 Oct 2025

Modelado Basado en Agentes Aplicado a Estrategias de Evacuación

Agent-Based Modeling Applied to Evacuation Strategies

Jorge De Jesús Galvez Rodríguez
jorge.galvez@academicos.udg.mx
Miguel Ángel Alejandro Islas Toski
alejandro.islastoski@academicos.udg.mx
Karla Avila Cardenas
karla.acardenas@academicos.udg.mx
Héctor Joaquin Escobar Cuevas
hector.escobar@academicos.udg.mx

1 Universidad de Guadalajara

Resumen
Los edificios y sus infraestructuras pueden volverse frágiles frente a desastres, ya sean naturales o provocados por el ser humano. En estos escenarios, garantizar la seguridad de las personas es una prioridad, y para ello, los modelos de evacuación se han convertido en herramientas clave. Su propósito principal es simular con realismo cómo un grupo grande de personas puede dirigirse hacia las salidas disponibles de manera eficiente durante una emergencia. Este artículo presenta un modelo de evacuación basado en agentes, donde interactúan dinámicamente las personas, los obstáculos y las salidas. El modelo se basa en solo cinco reglas para reproducir el proceso de evacuación, pero es capaz de incorporar fenómenos complejos como los atascos y los movimientos irregulares que suelen aparecer bajo presión. A diferencia de otros enfoques, este modelo introduce comportamientos habituales en situaciones de estrés extremo, como caídas, desorientación o episodios de pánico, lo que permite una representación más cercana a lo que realmente ocurre en situaciones críticas. Para evaluar el desempeño de la propuesta, se llevaron a cabo diversos experimentos y estudios de caso en entornos urbanos reales. Los resultados muestran que el modelo no solo reproduce con mayor fidelidad el comportamiento humano durante una evacuación, sino que también ofrece información valiosa para mejorar la planificación y respuesta ante emergencias.

Palabras clave: Modelos Basados en Agentes; Modelos de Evacuación; Escenarios de Emergencia; Seguridad de Edificios

Abstract
Buildings and their infrastructures can become fragile in the face of disasters, whether natural or man-made. In these scenarios, ensuring people's safety is a priority, and evacuation models have become key tools for this purpose. Their main purpose is to realistically simulate how a large group of people can move towards the available exits efficiently during an emergency. This paper presents an agent-based evacuation model where people, obstacles and exits interact dynamically. The model relies on only five rules to reproduce the evacuation process, but is able to incorporate complex phenomena such as traffic jams and irregular movements that often appear under pressure. Unlike other approaches, this model introduces common behaviors in situations of extreme stress, such as falls, disorientation or panic episodes, allowing a closer representation of what actually happens in critical situations. To evaluate the performance of the proposal, several experiments and case studies were carried out in real urban environments. The results show that the model not only more closely reproduces human behavior during an evacuation, but also provides valuable information to improve emergency planning and response.

Keywords: Agent-Based Models; Evacuation Models; Emergency Scenarios; Building Safety

1. Introducción

Las ciudades modernas enfrentan el riesgo constante de sufrir catástrofes, producto tanto de causas naturales como de acciones humanas. La rápida expansión urbana ha dado lugar a zonas densamente pobladas que, en muchos casos, carecen de una planificación adecuada y de medidas efectivas de resiliencia. Esto las vuelve especialmente vulnerables frente a eventos como sismos, inundaciones, huracanes o incendios. A este escenario se suma el impacto del cambio climático, que ha intensificado la frecuencia y gravedad de muchos desastres naturales. Por otro lado, también existen amenazas derivadas de la actividad humana, como los accidentes industriales, los actos de terrorismo o las fallas en las infraestructuras esenciales (Xie et al., 2024). La alta concentración de personas y recursos en las áreas urbanas multiplica las consecuencias de estos eventos, haciendo indispensable una planificación estratégica y una preparación efectiva para reducir los riesgos y proteger tanto a la población como al entorno urbano.

Los modelos de evacuación juegan un papel fundamental en la gestión del riesgo y la protección de las personas en entornos urbanos (Sun et al., 2024). Gracias a estos modelos, los responsables de la planificación pueden simular distintos tipos de emergencias tales como incendios, terremotos o inundaciones y anticipar cómo reaccionarían y se desplazarían las personas ante esos escenarios. Al incorporar enfoques avanzados, estos modelos ofrecen información clave que permite mejorar la asignación de recursos, afinar los planes de respuesta y optimizar el diseño de edificios y espacios públicos (Islas-Toski et al., 2024). Más allá de reducir los riesgos, su aplicación contribuye a que las ciudades evolucionen hacia entornos más seguros, preparados y resilientes, capaces de responder con eficacia cuando se presentan situaciones críticas.

A lo largo del tiempo, han surgido distintos enfoques para modelar el proceso de evacuación, cada uno con sus propias estrategias y fundamentos. Entre ellos destacan los modelos de flujo continuo, como los modelos de Greenshields (Islas-Toski et al., 2024) y otros desarrollos centrados en el movimiento peatonal (Ensing & Amrit, 2023; Fernández-Garza et al., 2019), que se basan en principios de la mecánica de fluidos para representar cómo se desplazan las personas en distintos entornos. Estos modelos han sido ampliamente utilizados para analizar el flujo de personas en edificios, estaciones o espacios públicos. Por otro lado, los Modelos Basado en Agentes (MBAs) han cobrado gran relevancia por su capacidad de representar de forma más realista las decisiones individuales y las interacciones entre personas. Dentro de estos modelos, se encuentran; el Modelo de Fuerza Social (MFS) (Seyfried et al., 2006), los Modelos de Autómatas Celulares (MAC) (Li et al., 2020), y los modelos de redes dinámicas (Bayram, 2016).

En la Tabla 1 se presenta un resumen de los modelos de evacuación más relevantes descritos en la literatura, junto con sus características principales. Si bien estos enfoques han aportado resultados valiosos, estos enfoques presentan limitaciones importantes (An et al., 2015). Uno de los principales problemas es que muchos de ellos dependen de formulaciones matemáticas complejas, lo que complica su adaptación a nuevos contextos o su modificación para representar diferentes escenarios (Ronchi, 2016). Además, suelen ser modelos estáticos (Hu et al., 2024), es decir, no contemplan los cambios que pueden producirse en tiempo real durante una evacuación. Esto representa una debilidad considerable, ya que en situaciones reales, como desastres naturales, incendios o atentados, es común que se alteren las condiciones del entorno.

Modelo de evacuación	Principales elementos	Referencia
Modelos Greenshields	Utilizan los principios de la mecánica de fluidos para representar el movimiento de las personas, aplicado en entornos específicos.	(Islas-Toski et al., 2024)
Modelos de flujo continuo de peatones	Aplican los principios de la mecánica de fluidos para simular el movimiento peatonal, ampliamente utilizado en el análisis del flujo en edificios y espacios públicos.	(Ensing & Amrit, 2023; Fernández-Garza et al., 2019)
Modelo de Fuerza Social (MFS)	Se centran en las interacciones individuales con fuerzas sociales como la atracción y la repulsión, y modela la dinámica colectiva.	(Seyfried et al., 2006)

Modelos de Autómatas Celulares (MAC)	Discretizan el espacio en celdas, simula el movimiento de una celda a otra según reglas, eficaz para la dinámica de multitudes.	(Li et al., 2020)
Modelos de Redes Dinámicas	Utiliza la teoría de redes para trazar conexiones y rutas en entornos complejos.	(Bayram, 2016)

Tabla 1. Modelos de evacuación propuestos en la literatura y sus principales características.

Otra de las limitaciones de dichos enfoques es que se centran casi exclusivamente en el comportamiento racional y predecible de los individuos. En la mayoría de estos enfoques, las personas son representadas como agentes que siguen reglas matemáticas de forma sistemática, sin contemplar los aspectos emocionales que surgen en situaciones de alta tensión. Sin embargo, en contextos reales, las emociones como el miedo, la histeria o la desorientación juegan un papel crucial en las decisiones y reacciones de las personas, afectando directamente los tiempos, las trayectorias y, en general, el desarrollo de la evacuación. La falta de consideración de estos factores emocionales y dinámicos limita el realismo de las simulaciones y, por ende, reduce la efectividad de estos modelos al momento de aplicarlos en escenarios reales.

Los MBAs (Helbing, 2012) son herramientas computacionales diseñadas para simular cómo actúan e interactúan agentes autónomos dentro de un entorno específico. En los MBAs, cada agente sigue un conjunto de reglas y comportamientos (Liu et al., 2024), lo que le permite tomar decisiones, adaptarse a cambios y relacionarse tanto con otros agentes como con su entorno. Este enfoque es especialmente valioso para estudiar sistemas complejos donde el comportamiento colectivo surge de la interacción de muchas partes y no puede predecirse simplemente analizando a cada agente por separado.

Los MBAs se utilizan en diversas áreas, como las ciencias sociales, la economía, la epidemiología y la planificación urbana (Macal & North, 2005), ya que ofrecen una perspectiva sobre cómo las interacciones descentralizadas pueden dar lugar a patrones y dinámicas complejas. Al capturar la diversidad y capacidad de adaptación de los agentes, estos modelos se han convertido en una herramienta clave para analizar situaciones que van desde la evolución de mercados hasta la gestión de evacuaciones, ayudando a entender y controlar sistemas complejos y cambiantes.

La diferencia principal entre los MBAs y los modelos convencionales, que se apoyan en ecuaciones matemáticas en diferencias, para representar y simular sistemas basándose no solo en las acciones autónomas de cada agente, sino de las interacciones entre agentes. Mientras que los modelos convencionales describen el comportamiento general de todo un sistema usando ecuaciones diferenciales, para explicar cómo se relacionan sus variables (Hirtz et al., 2002). Esto hace que los métodos convencionales solo consideren aspectos generales de la simulación, sin considerar interacción entre las variables involucradas. Debido a lo anterior, el MBA permite capturar la diversidad y la capacidad de adaptación de las personas, reflejando mejor cómo varían los comportamientos y las decisiones dentro de un grupo.

Los MBAs destacan especialmente por su capacidad para reflejar la diversidad entre individuos, ya que pueden simular interacciones complejas y fenómenos emergentes que surgen justamente de esas diferencias. Esto permite ofrecer una visión más detallada y realista de sistemas donde las variaciones individuales influyen de manera importante en los resultados. Por eso, los MBAs son especialmente adecuados para construir modelos de evacuación, ya que pueden reproducir los distintos comportamientos e interacciones de las personas en entornos que cambian

constantemente. Esto ayuda a capturar la complejidad del comportamiento humano en una evacuación, incluyendo fenómenos como el pánico, la cooperación o la influencia de los vínculos sociales. Además, estos modelos pueden ajustarse en tiempo real a cambios en el entorno, como salidas bloqueadas o peligros que van apareciendo, lo que resulta en simulaciones mucho más realistas. Al integrar tanto la variabilidad individual como las condiciones cambiantes, los MBAs proporcionan un marco sólido para evaluar distintas estrategias de evacuación, optimizar la ubicación de recursos y mejorar la seguridad y eficiencia en la planificación de emergencias. Gracias a esta capacidad de adaptación y realismo, los MBAs se posicionan como una herramienta flexible para diseñar modelos de evacuación completos y efectivos.

Este artículo propone un modelo de evacuación basado en agentes con un diseño sencillo, que utiliza solo cinco reglas para replicar cómo se desarrolla una evacuación. Estas reglas capturan detalles importantes, como el movimiento de las personas y los momentos en que se forman atascos, además de reflejar comportamientos comunes en situaciones de emergencia, como accidentes, pánico y desorientación. A diferencia de muchos modelos convencionales, este enfoque incorpora estas reacciones típicas que suelen presentarse en escenarios de alta tensión. El modelo propuesto, representa de manera más realista cómo actúan las personas durante una evacuación, lo que se traduce en resultados más prácticos y confiables. Para probar la efectividad del modelo propuesto, se tomó en cuenta en un entorno real de evacuación, el Hospital Civil Viejo Fray Antonio Alcalá en Guadalajara, México. Los resultados muestran que el modelo proporciona información útil sobre el proceso de evacuación y permite evaluar con mayor precisión cómo se comportan las personas en situaciones de emergencia.

2. Modelado Basado en Agentes.

Un Modelo Basado en Agentes (MBA) es una herramienta de simulación computacional que representa y analiza cómo actúan e interactúan agentes autónomos dentro de un entorno específico (Siegenfeld & Bar-Yam, 2020). Cada agente sigue un conjunto de reglas y comportamientos que le permiten tomar decisiones, adaptarse a cambios y relacionarse tanto con otros agentes como con su entorno. Este enfoque centrado en el agente ayuda a capturar la diversidad y la complejidad del comportamiento individual, así como su impacto en el sistema en conjunto.

Las reglas en los MBAs son fundamentales porque definen cómo se comportan e interactúan los agentes dentro de la simulación (Page et al., 2013). Estas reglas establecen de qué manera los agentes toman decisiones, reaccionan a su entorno y se relacionan con otros agentes, y en conjunto determinan cómo evoluciona todo el sistema. Entre las reglas más comunes se encuentran las de movimiento, que indican cómo los agentes se desplazan por el espacio; las de interacción, que regulan cómo los agentes influyen y son influenciados por sus pares; y las de toma de decisiones, que describen los criterios que usan los agentes para elegir acciones según su estado interno o estímulos externos.

El proceso de un modelo basado en agentes implica la interacción de una población $\boldsymbol{P}$ de un total de p agentes diferentes que se definen de la siguiente manera:

$$\boldsymbol{P} = \{\boldsymbol{A_1}, \boldsymbol{A_2}, \dots, \boldsymbol{A_p}\} \tag{1}$$

Un MBA considera varios pasos importantes. Inicialmente, se inicializan los p agentes en $\boldsymbol{P}$. En este proceso, a cada agente se le asignan atributos aleatorios que determinan su posición, estado o

condición. Con estos atributos aleatorios, la población de agentes será heterogénea. A continuación, durante el proceso de simulación, los agentes se seleccionan aleatoriamente o siguen un orden predefinido. El agente seleccionado, $\boldsymbol{A_k}, k \in 1, \ldots, p$ sigue un conjunto de reglas establecidas, modificando su estado, posición o relaciones con otros agentes. Estas reglas suelen tener en cuenta al agente actual, a sus homólogos vecinos y al entorno. Por último, se define un criterio de parada para concluir el proceso una vez alcanzado un estado o condición determinados. Este planteamiento estructurado permite simular sistemas complejos y dinámicos, lo que permite comprender mejor los comportamientos emergentes y las interacciones dentro de poblaciones heterogéneas.

Las reglas en los MBAs pueden capturar la diversidad y complejidad de comportamientos dentro de una población, mostrando cómo los agentes reaccionan a distintos estímulos y se relacionan entre sí y con el entorno. Estas reglas permiten crear simulaciones realistas y flexibles, útiles para estudiar una gran variedad de situaciones y fenómenos. Un tipo común de regla en los MBAs son las reglas IF-THEN, que ofrecen una manera sencilla de programar el comportamiento de los agentes. Estas reglas establecen respuestas condicionales basadas en el estado actual del agente o su entorno, lo que facilita la representación de procesos complejos de toma de decisiones y comportamientos que se adaptan a las circunstancias. Así, usando reglas IF-THEN, los MBAs pueden simular cómo los agentes reaccionan a diferentes situaciones, dando lugar a patrones emergentes y ayudando a comprender mejor el sistema que se está estudiando.

En el contexto de los MBAs, el concepto de emergencia se refiere a cómo surgen patrones, comportamientos y propiedades complejas a partir de las interacciones simples entre agentes individuales dentro del sistema (Heard et al., 2015). Estas características emergentes no se programan de forma directa en el modelo, sino que aparecen como resultado de la dinámica colectiva y las interacciones locales de los agentes que siguen reglas básicas. La emergencia muestra cómo las acciones descentralizadas y autónomas de cada agente pueden generar fenómenos organizados y, a menudo, inesperados a nivel global, ayudándonos a entender mejor los mecanismos que operan en sistemas complejos.

3. El modelo basado en agentes propuesto para evacuación

En esta sección se presenta el MBA diseñado para simular los procesos de evacuación en escenarios complejos. Este modelo logra capturar la diversidad de comportamientos individuales y las interacciones complejas, especialmente en situaciones emocionales difíciles como la histeria o la desorientación, que suelen darse durante una evacuación. Para facilitar la explicación del modelo propuesto, el análisis se divide en tres partes principales: (i) Creación del entorno, (ii) Agentes y reglas, y (iii) Procedimiento computacional. La parte (i) se enfoca en la creación del entorno, es decir, una forma computacional de representar el entorno, las personas, las salidas y los obstáculos. En la parte (ii), se definen las reglas de comportamiento de los agentes. Finalmente, la parte (iii) describe el procedimiento computacional completo.

3.1 Creación del entorno

En los MBAs, contar con un entorno definido es clave para ofrecer el contexto espacial necesario en las simulaciones. En este trabajo, se discretiza el mapa del entorno a simular, en términos de una cuadricula C discreta bidimensional de MxN celdas, para representar el espacio donde se mueven y actúan los agentes. Esta estructura tiene varias ventajas. Primero, ofrece un marco de referencia claro e intuitivo para modelar las relaciones espaciales y los movimientos, lo que facilita

visualizar y analizar cómo los agentes interactúan entre sí y con el entorno. Cada celda de la cuadrícula puede representar diferentes elementos, como áreas abiertas, personas, obstáculos o salidas, permitiendo así una representación detallada y flexible del entorno. Además, esta configuración simplifica la aplicación de las reglas de comportamiento de los agentes, porque tanto el movimiento como las interacciones se pueden gestionar fácilmente dentro de la cuadrícula.

En la inicialización del entorno, todos los elementos $x_{ij}, i \in 1, \dots, M, j \in 1, \dots, N$ de la cuadricula C se inicializan en cero. A continuación, los elementos correspondientes a las salidas del modelo de evacuación se ponen a uno. Esta información se utiliza para crear un mapa de distancias (MD). Este mapa visualiza en cada celda de la cuadrícula la distancia más corta al elemento con valor 1 (la salida en el modelo de evacuación). Para elaborar el mapa de distancias MD, se utiliza el algoritmo de Chamfer. El algoritmo Chamfer es una técnica utilizada para calcular distancias aproximadas dentro de una cuadrícula discreta, empleada normalmente en el contexto de la búsqueda de trayectorias y el análisis de formas. Funciona propagando iterativamente valores de distancia a través de la cuadrícula, creando un mapa de distancias que representa el camino más corto desde cada celda a un conjunto de celdas objetivo, como obstáculos o puntos de salida. El algoritmo suele utilizar un paso hacia delante y otro hacia atrás para actualizar los valores de distancia. En el paso hacia delante, la distancia de cada celda se actualiza en función de sus celdas vecinas situadas arriba y a la izquierda (utilizando una máscara M_M^I), mientras que, en el paso hacia atrás, las actualizaciones se basan en las celdas vecinas situadas abajo y a la derecha (utilizando una máscara M_M^D). Este enfoque de doble pasada garantiza que los valores de distancia se propaguen con precisión por toda la cuadrícula. Las distancias se calculan utilizando un conjunto de pesos predefinidos que se aproximan a la distancia euclidiana, lo que permite al algoritmo manejar con eficacia diversas configuraciones de la cuadricula.

Como resultado del proceso, se produce el mapa de distancias MD. La capacidad del algoritmo Chamfer para generar rápidamente mapas de distancias lo convierte en una herramienta ideal para aplicaciones en tiempo real en los MBAs, permitiendo actualizaciones dinámicas y simulaciones realistas de movimientos e interacciones de agentes dentro de entornos complejos.

La Figura 1 ilustra la evolución del proceso realizado por el algoritmo Chamfer para generar el mapa de distancias. La Figura 1(a) representa la cuadrícula original C, en la que todos sus valores son cero excepto dos elementos que representan las salidas ($x_{1,1} = x_{4,4} = 1$). En estas condiciones, el algoritmo Chamfer, calcula la distancia de cada celda (con valor cero) al elemento o elementos de la cuadrícula con valor uno. Considerando la Figura 1(a) como entrada, la transformación Chamfer se aplica mediante un proceso dual con las máscaras mostradas en la Figura 1(b). El resultado de este procesamiento es el mapa de distancias MD que se muestra en la Figura 1(c). En este mapa MD, a todas las celdas con un valor inicial de cero en la cuadrícula C se le ha asignado un valor correspondiente a la distancia de esa celda a la salida más cercana. Por ejemplo, en la Figura 1(c), el elemento $x_{3,6}$ tiene un valor de 3, lo que indica su distancia a una de las salidas definidas.

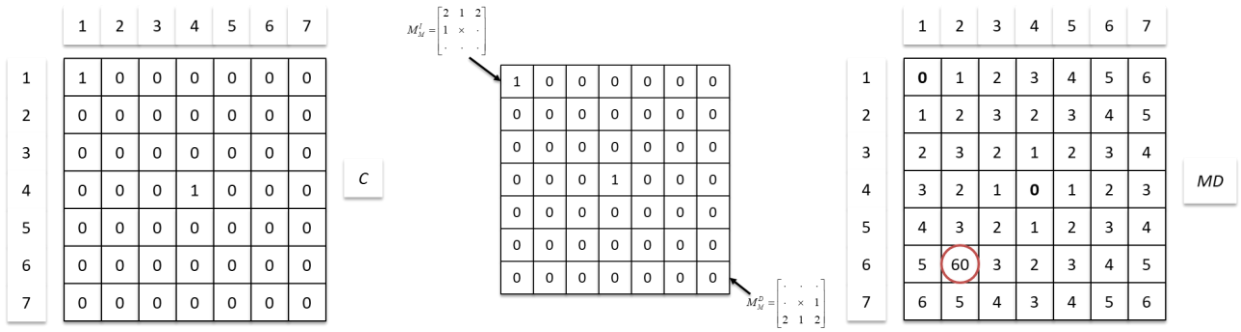


Figura 1. Evolución del proceso realizado por el algoritmo Chamfer para generar el mapa de distancias MD. (a) Cuadrícula original C, (b) Máscaras y (c) Mapa MD producido.

Una vez que se ha creado el mapa de distancias MD, se pueden definir los obstáculos. Estos obstáculos son celdas, o grupos de celdas, que no pueden ser ocupadas, por lo que se les asigna un valor de distancia (O) hacia las salidas muy alto. Al poner un número tan grande, estas celdas quedan bloqueadas y no se consideran rutas válidas para la evacuación. El valor O debe ser mayor que el producto de las dimensiones de la cuadrícula, es decir, $M \times N$.

La Figura 1(c) muestra este proceso, donde la celda en la posición $x_{6,2}$ se marca como obstáculo al asignarle un valor de 60. Este valor elevado asegura que esa celda no sea utilizada como parte de las rutas de evacuación. Así, la celda queda efectivamente inaccesible para los agentes, reflejando con precisión la existencia de barreras físicas en el entorno simulado.

3.2 Agentes y reglas

En el modelo propuesto, los agentes $\{\boldsymbol{A_1}, \boldsymbol{A_2}, \ldots, \boldsymbol{A_p}\}$, se inicializan o localizan aleatoriamente dentro de las celdas válidas de la cuadrícula C. Esta configuración garantiza que cada agente comience desde una ubicación factible, facilitando patrones de movimiento realistas dentro del entorno. Al situar a los agentes en celdas accesibles y desocupadas, el modelo refleja con exactitud las condiciones iniciales de un escenario de evacuación, permitiendo la simulación de interacciones y movimientos realistas a medida que los agentes navegan hacia las salidas mientras evitan obstáculos y a los demás agentes.

Los agentes en el modelo de evacuación siguen un conjunto de reglas que determinan cómo se mueven, interactúan y toman decisiones. Estas reglas están pensadas para imitar el comportamiento humano en situaciones de evacuación, incluyendo casos donde pueden ocurrir accidentes, pánico o desorientación. El modelo cuenta con cinco reglas: una de movimiento, una de estancamiento y tres de comportamiento. La regla de movimiento indica cómo los agentes se desplazan por el entorno, eligiendo rutas basadas en el mapa de distancias y evitando obstáculos. La regla de estancamiento se activa cuando los agentes quedan bloqueados o atrapados, y les guía sobre cómo actuar en estas situaciones. Por último, las tres reglas de comportamiento consideran los aspectos emocionales y psicológicos, como los movimientos irracionales causados por el pánico o la indecisión producto de la desorientación. A continuación, se explican en detalle cada una de estas reglas.

3.2.1 Regla de Movimiento

La regla de movimiento modela cómo los agentes navegan por el entorno, eligiendo caminos basados en el mapa de distancias y evitando obstáculos. En nuestro modelo de evacuación, cada agente intenta desplazarse hacia la salida más cercana basándose en el camino más corto proporcionado por el mapa de distancias MD, ajustando su ruta en respuesta a los cambios del entorno. En esta regla, cada agente $\boldsymbol{A_k}$ explora su vecindario más cercano (VEC), definido por el vecindario de Moore (véase la Figura 2(a)), para decidir su siguiente movimiento. El vecindario de Moore incluye las ocho celdas adyacentes que rodean al agente. Por lo tanto, los agentes pueden moverse vertical, horizontal o diagonalmente a una celda adyacente que no esté ocupada por otro agente. La regla especifica que el agente se moverá a cualquier celda dentro del vecindario VEC que tenga un valor de distancia menor o igual que la posición actual del agente. Si varias celdas circundantes dentro de la vecindad tienen el mismo valor de distancia mínima, el agente elegirá aleatoriamente una de estas celdas. Así, si el agente $\boldsymbol{A_k}$ se encuentra en la posición $x_{i,j}$ de la cuadrícula, deberá comprobar cuáles de las ocho celdas dentro de VEC tienen un valor de distancia menor que la distancia en la posición $x_{i,j}$. La vecindad VEC se define matemáticamente como sigue:

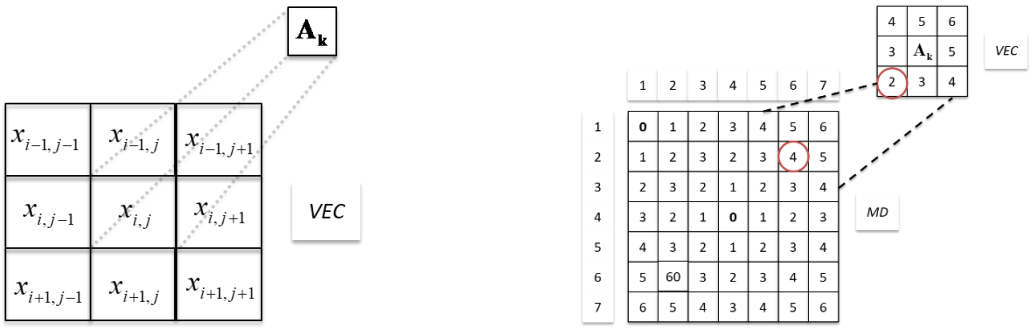
$$VEC(x_{i,j}) = \{x_{i-1,j-1}, x_{i-1,j}, x_{i-1,j+1}, x_{i,j-1}, x_{i,j+1}, x_{i+1,j-1}, x_{i+1,j}, x_{i+1,j+1}\} \tag{2}$$

Una vez encontrada la celda que satisface esta condición, el agente puede moverse a esa nueva posición de celda. Así, la regla de movimiento (R_M) puede definirse como el cambio de posición del agente $\boldsymbol{A_k}$ de $x_{i,j}$ a $x_{a,b}$ tal que la distancia de la posición $x_{a,b}$ sea menor que la de $x_{i,j}$ dentro de la vecindad VEC de $x_{i,j}$. Esta regla puede describirse matemáticamente como sigue:

$$R_M{:}x_{a,b} \leftarrow \boldsymbol{A_k} \quad \left\{\left(x_{a,b} \in VEC(x_{i,j})\right) \wedge \left(d(x_{a,b}) \leq d(x_{i,j})\right)\right\} \tag{3}$$

Aquí, $d(\cdot)$ representa la distancia desde una posición específica dentro de la cuadrícula. La Figura 2(b) ilustra cómo se aplica esta regla. En el ejemplo, se muestra un agente $\boldsymbol{A_k}$ ubicado en la celda $x_{2,6}$. Su vecindario inmediato está compuesto por las celdas del conjunto $\{x_{1,5}, x_{1,6}, x_{1,7}, x_{2,5}, x_{2,7}, x_{3,5}, x_{3,6}, x_{3,7}\}$.

La distancia actual del agente a la salida es 4. Sin embargo, dentro de su vecindario existe una celda, $x_{3,5}$, con una distancia menor, en este caso 2. Por lo tanto, el agente se moverá a esa celda, ya que representa un avance hacia la salida.



(a) (b)

Figura 2. Proceso de la regla de movimiento. (a) Barrio de Moore y (b) proceso de aplicación de la regla.

3.2.2 Regla de Estancamiento

En muchas situaciones, los agentes tienden a aglomerarse cerca de las salidas, avanzando lentamente, o se topan con obstáculos que bloquean su paso en determinada dirección. Cuando eso sucede, actúan de forma similar a los seres humanos: intentan seguir moviéndose, buscando caminos alternativos que les permitan llegar más rápido a la salida. Este tipo de comportamiento adaptativo no puede lograrse únicamente con la regla de movimiento. Según esa regla, un agente $\boldsymbol{A_k}$ solo puede trasladarse desde su posición actual $x_{i,j}$ a una celda vecina $x_{a,b}$ (siempre que $x_{a,b} \in VEC(x_{i,j})$)si la nueva celda tiene un valor de distancia $d(x_{a,b})$ menor que el de su posición actual $d(x_{i,j})$. Si no encuentra ninguna celda que cumpla con esta condición, el agente no puede avanzar y permanece donde está. Este estado se conoce como estancamiento. Esta limitación evidencia la necesidad de incorporar reglas adicionales que permitan a los agentes reaccionar ante bloqueos y buscar rutas alternativas de forma dinámica, lo que aporta un comportamiento más realista y humano al modelo de evacuación.

Para evitar que los agentes se acumulen en un mismo punto sin poder avanzar, se introduce la regla de estancamiento R_E. Esta regla actúa cada vez que un agente entra en estado de estancamiento, es decir, cuando no puede moverse porque ninguna de las celdas vecinas tiene una distancia menor que la actual. En ese caso, el valor de distancia de su posición se incrementa en una unidad: $d(x_{i,j}) = d(x_{i,j}) + 1$ Con este pequeño ajuste, llegará un momento en que la celda en la que se encuentra el agente tendrá una distancia mayor que alguna de sus vecinas, lo que le permitirá moverse siguiendo la regla de movimiento normal $\left(d(x_{a,b}) < d(x_{i,j})\right)$. Este mecanismo evita que los agentes queden atrapados permanentemente, favorece el flujo continuo y les da la posibilidad de descubrir rutas alternativas, reduciendo el riesgo de congestión en las salidas. En conjunto, permite una simulación más fluida y cercana a lo que podría ocurrir en una evacuación real. La regla R_E se resume como el aumento, en cada iteración, del valor de distancia $d(x_{i,j})$ de un agente $\boldsymbol{A_k}$ que ha quedado bloqueado.

$$R_E: IF\ \boldsymbol{A_k}\ \text{esta}\ Estancado\ THEN\ d(x_{i,j}) = d(x_{i,j}) + 1 \tag{4}$$

La Figura 3 ilustra cómo actúa la regla de estancamiento. En la Figura 3(a) se observa el mapa de distancias que representa el entorno. El agente $\boldsymbol{A_k}$ se encuentra en la celda $x_{3,1}$, bloqueado por tres obstáculos ubicados en $x_{2,1}, x_{2,2}$ y $x_{3,2}$. En esta situación, el agente no tendría posibilidad de avanzar, ya que todas las celdas vecinas están ocupadas o no ofrecen una distancia menor hacia la salida, lo que lo dejaría completamente estancado. Para resolver esto, se aplica la regla de estancamiento, que incrementa artificialmente el valor de distancia de la celda donde se encuentra el agente. En este caso, después de dos iteraciones, el valor de $d(x_{3,1})$ aumenta de 2 a 5. Con este nuevo valor, la celda vecina $x_{4,2}$, que tiene una distancia de 3, pasa a ser una mejor opción, permitiendo que el agente se desplace hacia ella según la regla de movimiento. Este ejemplo muestra cómo la regla de estancamiento ayuda a los agentes a liberarse de bloqueos, sortear obstáculos y continuar su camino hacia la salida, evitando que queden atrapados de forma indefinida.

Figura 3. Ejemplo del efecto de la regla de estancamiento. (a) Entorno original y (b) Efecto de la regla.

3.2.3 Reglas de Comportamiento

Durante una evacuación, ciertos factores de comportamiento pueden tener un impacto decisivo en la forma en que las personas reaccionan y se desplazan por el espacio. Entre estos factores destacan los accidentes, la histeria y la desorientación. Los accidentes pueden generar lesiones o bloqueos en zonas clave, lo que retrasa la evacuación y contribuye a la formación de cuellos de botella. La histeria, impulsada por el miedo o el pánico, puede dar lugar a comportamientos desordenados e impredecibles, como empujones, estampidas o la desobediencia de instrucciones de seguridad, lo que agrava aún más la situación. La desorientación, por su parte, suele deberse a la confusión o al desconocimiento del entorno, y puede llevar a las personas a tomar direcciones equivocadas o a no localizar las salidas a tiempo, complicando seriamente su escape. Aunque estos comportamientos pueden tener un alto impacto, estudios previos indican que en escenarios de evacuación típicos no más del 10 % de la población tiende a presentar este tipo de reacciones (Percy et al., 2011).

No obstante, este porcentaje puede modificarse según cambien las condiciones de la población, como un aumento en los niveles de estrés o la influencia de factores externos. También puede ajustarse para reflejar mejor las características de comportamiento de distintos grupos sociales o demográficos. Esta posibilidad de adaptación hace que el modelo sea mucho más flexible y adecuado para una variedad de escenarios contemporáneos, permitiendo simular con mayor precisión distintas dinámicas de evacuación. Esta capacidad de ajuste es clave para que el modelo se adapte a contextos específicos y ofrezca una representación más fiel y útil de los procesos de evacuación en situaciones reales.

Incluir estos perfiles de comportamiento en un modelo de evacuación resulta fundamental para representar con mayor fidelidad lo que ocurre en situaciones reales. Esta incorporación permite diseñar estrategias de evacuación más efectivas, ya que ayuda a detectar posibles puntos críticos y a fortalecer los planes de acción ante emergencias. Además, brinda a los responsables de la gestión de crisis y al personal de primera respuesta una herramienta útil para anticiparse a las reacciones humanas durante eventos caóticos, facilitando una mejor toma de decisiones. En conjunto, esto se traduce en una evacuación más segura, organizada y, en muchos casos, en la posibilidad de salvar vidas.

En el modelo se establecen tres reglas para simular comportamientos que un agente $\boldsymbol{A_k}$ puede presentar durante una evacuación: Regla de accidente (R_A), Regla de histeria (R_H) y Regla de desorientación (R_D). El primer paso consiste en determinar si un agente desarrollará alguno de

estos comportamientos. Para ello, se realiza una prueba probabilística: se genera un número aleatorio r, distribuido de forma uniforme entre 0 y 1. Si el valor de r es menor o igual a 0.1 (es decir, un 10 %), se considera que el agente presentará uno de estos tres estados. Si el valor supera 0.1, se asume que el agente no sufrirá ninguna alteración. Este enfoque basado en la probabilidad permite introducir cierta variabilidad y realismo en la simulación, ya que refleja cómo, en la vida real, solo una parte de la población reacciona de manera extrema o inesperada durante una emergencia.

Una vez que se ha decidido que el agente $\boldsymbol{A_k}$ experimentará uno de los tres comportamientos, entonces se elige aleatoriamente uno de ellos con la misma probabilidad para elegir uno de los tres. Si se selecciona la condición de accidente como comportamiento del agente $\boldsymbol{A_k}$, se activa la regla de accidente R_A. Bajo R_A, se considera que el agente $\boldsymbol{A_k}$ ha sufrido un accidente o lesión, dejando al agente inmóvil. Como resultado, el agente $\boldsymbol{A_k}$ permanece como obstáculo 0 en la celda $x_{i,j}$ que ocupa en la cuadrícula. Esto significa que el agente bloqueará la celda durante el resto de la simulación (durante el resto de iteraciones), impidiendo que otros agentes se muevan u ocupen esa celda. R_A se describe como:

$$R_A: IF\ \boldsymbol{A_k}\ \text{esta en situación de accidente}\ THEN\ \boldsymbol{A_k}\ es\ un\ obstaculo \tag{5}$$

Esta regla simula el impacto de los accidentes durante la evacuación, destacando cómo las lesiones pueden crear barreras adicionales y complicar el proceso de evacuación, proporcionando así un escenario más realista y desafiante para la planificación de emergencias y las estrategias de respuesta.

Por otro lado, si el comportamiento del agente $\boldsymbol{A_k}$ es histeria, se activa la regla de histeria R_H. Según esta regla, el agente $\boldsymbol{A_k}$ permanece como obstáculo en la misma posición durante un número fijo de iteraciones n. Durante este periodo, el agente queda inmovilizado, bloqueando efectivamente la celda $x_{i,j}$ que ocupa e impactando en el movimiento de otros agentes a su alrededor. Una vez transcurrido el número especificado de iteraciones n, el agente $\boldsymbol{A_k}$ asume un comportamiento normal y puede moverse según la regla de movimiento estándar R_M, como cualquier otro agente. La regla R_H puede describirse como sigue:

$$R_H: IF\ \boldsymbol{A_k}\ \text{esta en situación de histeria}\ THEN\ \boldsymbol{A_k}\ es\ un\ obstaculo\ cierta\ cantidad\ de\ iteraciones\ (n) \tag{6}$$

Esta regla simula la parálisis temporal o el comportamiento irracional que puede ocurrir debido a la histeria, proporcionando una representación más realista de cómo el pánico puede afectar a la dinámica de evacuación y potencialmente causar cuellos de botella o retrasos temporales.

Finalmente, si se activa el comportamiento de desorientación, la regla de desorientación R_D rige el comportamiento. Esta regla afecta al agente durante m iteraciones. Durante este periodo, el agente se confunde y se mueve aleatoriamente desde su posición actual $x_{i,j}$ a otra celda aleatoria $x_{c,d}$ dentro de su vecindad $VEC(x_{i,j})$. En esta regla, el agente se moverá aleatoriamente a cualquier celda vecina, independientemente de los valores de distancia, con la única restricción de que la celda no esté ocupada. La regla R_D puede describirse como sigue:

$$R_D: IF\ \boldsymbol{A_k}\ \text{esta en situación de desorientación}\ THEN\ x_{c,d} \leftarrow \boldsymbol{A_k}\ \ (n) \tag{7}$$

Esta regla emula la desorientación que puede sufrir un individuo durante el proceso de evacuación, haciendo que el agente realice movimientos aparentemente irracionales que no conducen necesariamente hacia la salida. Esta regla ayuda a modelar el impacto de la confusión y

la falta de conciencia espacial en el proceso de evacuación, añadiendo otra capa de realismo a la simulación al tener en cuenta la naturaleza impredecible del comportamiento humano en situaciones de emergencia.

3.3 Proceso Computacional

En esta sección describe el procedimiento computacional completo del modelo basado en agentes para emular el proceso de evacuación. Todas las operaciones se detallan en pseudocódigo en el Algoritmo 1 y el diagrama presentado en la Figura 4. Como se puede observar en el Algoritmo 1, el método comienza inicializando sus parámetros generales de funcionamiento (línea 1), como el tamaño de la cuadricula ($M \times N$), el número de personas que hay en la sala a evacuar (p), el número de iteraciones que los agentes podrían desorientarse según la regla de desorientación (n) y el número máximo de iteraciones ($iteramax$) para las que se mostrará el proceso de evacuación. Los parámetros de inicialización son fundamentales, ya que definen el entorno de simulación y las condiciones en las que se producirá la evacuación. La línea 2 construye una cuadrícula que representa la imagen binaria del entorno que se desea modelar. La definición del entorno de esta manera permite simular fácilmente movimientos e interacciones, ya que al discretizar un mapa como una imagen binaria, facilita el conocer las posiciones tanto en el eje horizontal como vertical, de cada agente y obstáculo. En la línea 3, se aplica el algoritmo de Chamfer sobre la cuadricula para obtener el mapa de distancias MD.

1:	Entrada: $\boldsymbol{p}, iteramax\ , M, N, n, m$
2:	$C \leftarrow$ ConfigurarEntorno$(M, N, \boldsymbol{p})$
3:	$MP \leftarrow$ AlgoritmoChamfer(C)
4:	**for** iteración=1: $iteramax$
5:	**for** $\boldsymbol{k}$ =1: $\boldsymbol{p}$
6:	$\{C\} \leftarrow$ Regla $R_M(\boldsymbol{A}_k)$
7:	$\{C, MD\} \leftarrow$ Regla $R_E(\boldsymbol{A}_k)$
8:	$\{C\} \leftarrow$ Regla $R_A(\boldsymbol{A}_k)$
9:	$\{C\} \leftarrow$ Regla $R_H(\boldsymbol{A}_k, n)$
10:	$\{C\} \leftarrow$ Regla $R_D(\boldsymbol{A}_k, m)$
11:	**end for**
12:	**end for**
13:	Salida: C

Algoritmo 1. Pseudocódigo del modelo de evacuación propuesto.

Una vez, que el mapa de distancia fue creado, el ciclo principal de la simulación es ejecutado un número máximo de iteraciones. Durante cada iteración, los agentes son sometidos a la evaluación de las reglas descritas con anterioridad. Durante el proceso iterativo, los agentes

interactúan entre sí generando comportamientos emergentes que describen la dinámica de la situación de evacuación.

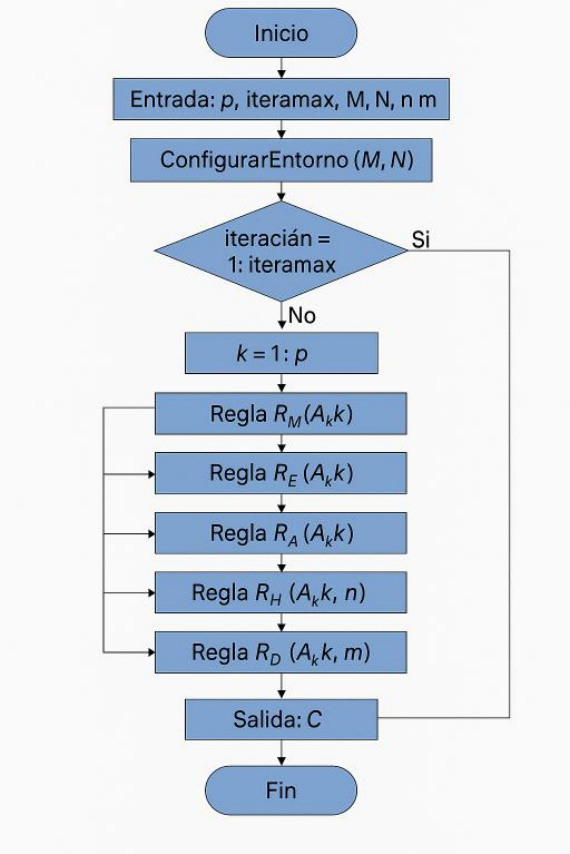


Figura 4. Diagrama de flujo del modelo de evacuación propuesto.

4. Resultados experimentales

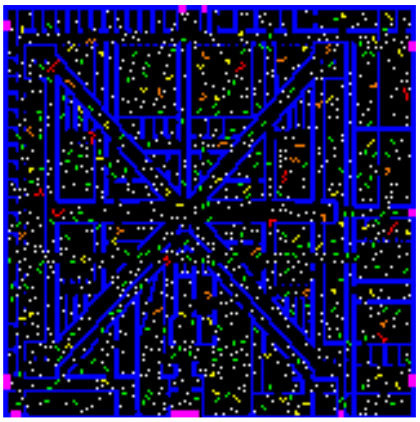
Una de las pruebas clave para evaluar la eficacia del modelo de evacuación basado en agentes se llevó a cabo utilizando una recreación del Hospital Civil Viejo Fray Antonio Alcalde, en Guadalajara, México. Este entorno fue elegido por su diseño complejo y su valor como caso real, ideal para comprobar si el modelo puede adaptarse a espacios amplios y con múltiples rutas, simulando además comportamientos humanos ante situaciones de emergencia. La simulación fue programada en MATLAB y ejecutada en un equipo portátil de gama media-alta (Huawei Matebook E, procesador Intel Core i7 de 11ª generación, 16 GB de RAM y gráficos Intel Iris Xe). En promedio, cada simulación tomó poco más de dos minutos en completarse (141 segundos).

Para trasladar los resultados a un contexto más realista, se tomó en cuenta la velocidad promedio de una persona al caminar, que ronda los 1.4 metros por segundo (aproximadamente 5 km/h). En el modelo, el hospital fue representado como una cuadrícula de 160 por 160 píxeles, correspondiendo a los 160 metros de lado del edificio real. Cada agente avanzaba una celda por iteración, simulando así un metro recorrido por segundo.

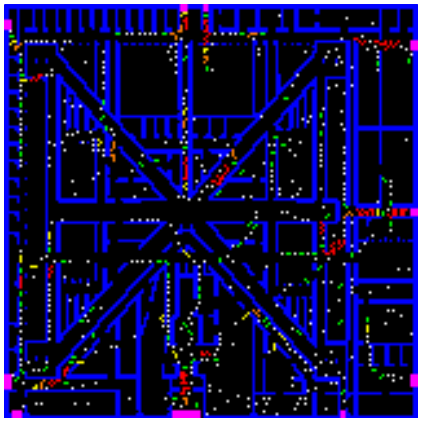
Durante la simulación (véase Figura 5), se logró evacuar exitosamente a 1500 agentes (personas) a través de 10 salidas distribuidas en un espacio de 160 × 160 píxeles. La evacuación se completó en 278 iteraciones, dentro de un total de 500 programadas. Si trasladamos esto al mundo real, considerando que cada celda equivale a un metro y que la velocidad promedio de desplazamiento

humano es de 1.4 metros por segundo, los 278 píxeles recorridos representan 278 metros, lo que equivale a un tiempo estimado de evacuación de 198.6 segundos, o aproximadamente 3.31 minutos. Lo que indica un proceso ágil y bien distribuido entre las salidas.

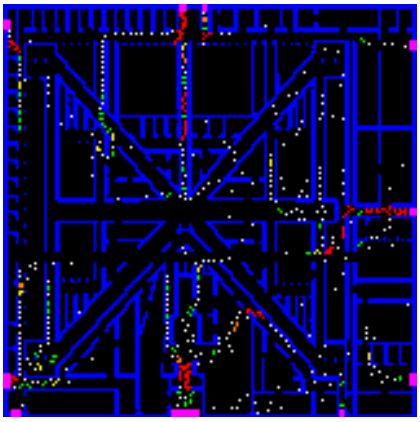
Para poner los resultados en perspectiva, se puede comparar con un simulacro de evacuación real realizado en ese mismo hospital (*Realizan Simulacros En Los Hospitales Civiles de Guadalajara | Coordinación General de Comunicación Social*), en el que participaron 1276 personas (320 empleados y 956 usuarios). A pesar de que el simulacro involucró pacientes que requerían asistencia especializada, la evacuación se completó en diez minutos, con el apoyo de 80 brigadistas, personal médico y de seguridad. Este contraste muestra que el modelo no solo logra tiempos de evacuación cercanos a los de un evento real, sino que también tiene la capacidad de simular condiciones complejas, como la necesidad de ayuda adicional o la presencia de obstáculos. Así, el modelo demuestra ser una herramienta útil para planear evacuaciones más seguras y eficientes, anticipándose a escenarios que podrían presentarse en la vida real.



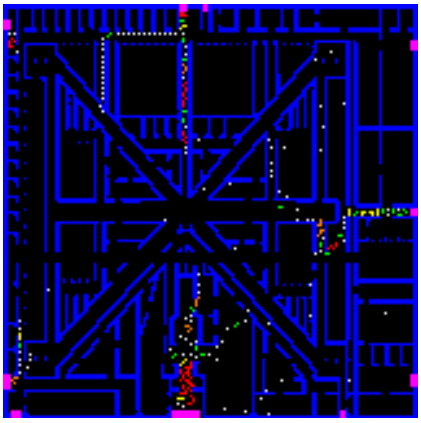
(a) 1 iteración



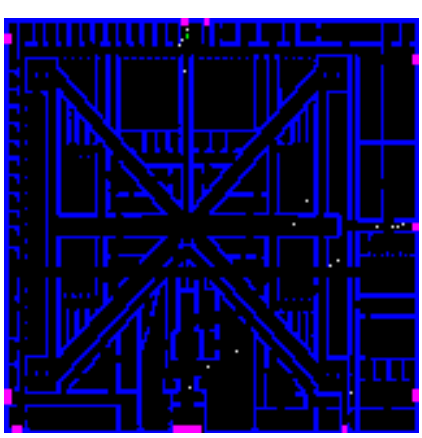
(b) 25 iteraciones



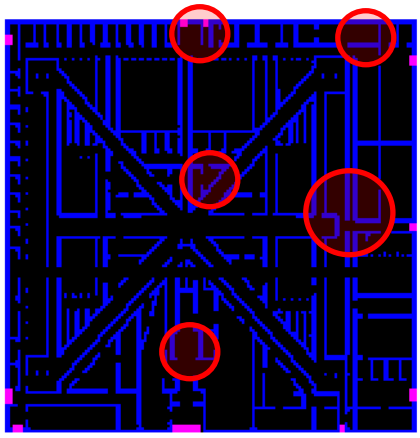
(c) 50 iteraciones



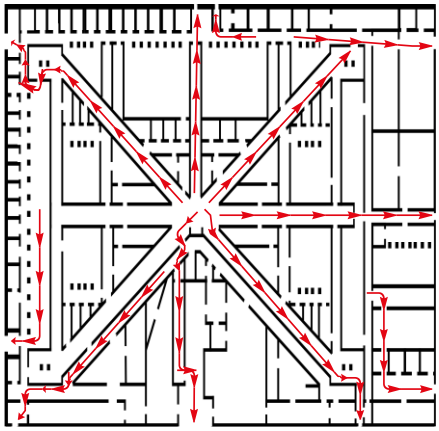
(d) 100 iteraciones



(e) 200 iteraciones



(f) 350 iteraciones



(g) Mejores rutas encontradas

Figura 5. Fray Antonio Alcalde Hospital simulando 1500 agentes.

La Figura 5 ilustra la evolución completa de la simulación de evacuación en el entorno hospitalario. En la Figura 5(a), la simulación comienza con agentes distribuidos aleatoriamente por el espacio. En la iteración 25 (Figura 5(b)), empiezan a surgir los primeros patrones de movimiento y comportamientos de seguimiento de los agentes. Un sistema codificado por colores resalta los grupos: blanco para agentes aislados, verde para dos cercanos, amarillo para tres, naranja para cuatro y rojo para zonas con cinco o más agentes; esta señal visual ayuda a identificar posibles cuellos de botella en los que mejoras en la infraestructura podrían mejorar el flujo de evacuación. A medida que avanza la simulación, las Figuras 5(c) y 5(d) capturan el entorno en las iteraciones 50 y 100, respectivamente, mostrando una dinámica de movimientos cada vez más definida. En la iteración 200 (Figura 5(e)), sólo quedan 17 agentes en el entorno, y en la iteración 350 (Figura 5(f)), el espacio está totalmente despejado, con círculos que marcan las zonas que anteriormente tenían la mayor concentración de agentes. Por último, la Figura 5(g) muestra las rutas de evacuación óptimas derivadas del comportamiento colectivo de los agentes, que convergen de forma natural en rutas que equilibran la distancia más corta con una congestión mínima, lo que demuestra cómo los agentes se adaptan tanto a la disposición espacial como a la densidad para formar las rutas de escape más eficientes.

Los resultados del experimento, realizado utilizando el entorno simulado del Hospital Fray Antonio Alcalde, muestran una relación notable entre el tiempo de evacuación del modelo y la duración observada en un simulacro real. Mientras que la simulación completó la evacuación en

aproximadamente 3,31 minutos en comparación con los 10 minutos registrados durante el evento real, esta diferencia puede explicarse por varias limitaciones del mundo real. En escenarios reales, factores como la asistencia a pacientes encamados, los retrasos en la transmisión de instrucciones y los distintos niveles de movilidad ralentizan considerablemente el proceso. En cambio, las ejecuciones iniciales del modelo basado en agentes produjeron tiempos de evacuación cercanos a la cifra real, pero tras detectar patrones emergentes y refinar la simulación, el sistema fue capaz de optimizar las rutas y reducir el tiempo necesario para despejar el entorno. Uno de los elementos más reveladores de la simulación fue la aparición de puntos de congestión (como se ve en la Figura 5(f)), que ponen al descubierto zonas físicas del hospital que podrían beneficiarse de mejoras estructurales, como ensanchar las puertas, rediseñar los pasillos o mejorar la señalización para gestionar mejor el flujo de personas. Las rutas óptimas descubiertas (Figura 5(g)) ofrecen una base sólida para desarrollar estrategias de evacuación adaptadas a la disposición del hospital. Además, el modelo reproduce con éxito comportamientos humanos clave en situaciones de estrés, como la agrupación y la navegación adaptativa, lo que refuerza su fiabilidad a la hora de reproducir dinámicas de evacuación realistas. Estos comportamientos emergentes no sólo mejoran el realismo del modelo, sino que también sirven de apoyo a aplicaciones prácticas de planificación arquitectónica y preparación ante emergencias. Al señalar las zonas críticas donde es más probable que se produzcan atascos, el modelo se convierte en una valiosa herramienta para perfeccionar los protocolos de seguridad en entornos con alta densidad de ocupantes.

5. Discusión y Limitaciones

Los resultados obtenidos con el modelo de evacuación propuesto demuestran su eficacia y su potencial para generalizarse a diversos entornos urbanos y arquitectónicos. Basado en un conjunto reducido pero sólido de reglas que describen las interacciones entre agentes, obstáculos y salidas, el modelo presenta una estructura adaptable que permite ajustarse fácilmente a diferentes contextos. Con modificaciones mínimas en parámetros como la densidad de ocupación, la configuración espacial o las características de las rutas de evacuación, puede aplicarse a escenarios como hospitales, centros comerciales, estaciones de transporte o complejos residenciales. Esta versatilidad lo convierte en una herramienta útil para la planificación y evaluación de estrategias de seguridad en infraestructuras de distinta escala y complejidad.

Una de sus principales fortalezas radica en la capacidad del enfoque basado en agentes para representar la naturaleza emergente y no lineal del comportamiento humano durante una evacuación. Al incorporar tanto patrones de movimiento planificados como reacciones emocionales, por ejemplo, pánico, desorientación o caídas, el modelo reproduce fenómenos como los cuellos de botella, las zonas de congestión o la formación espontánea de grupos. Además, su estructura modular facilita la integración con sistemas de información geográfica y plataformas de simulación en tiempo real, ampliando su aplicabilidad hacia escenarios de ciudad inteligente y gestión dinámica de emergencias.

No obstante, al ser un modelo computacional, el modelo presenta ciertas limitaciones que deben considerarse para su uso y futura mejora. En primer lugar, la simulación depende de la calidad y disponibilidad de los datos de entrada, como los planos arquitectónicos o los patrones de movilidad, los cuales pueden no reflejar con precisión la complejidad del entorno físico o social real. Asimismo, aunque se incluyen variables emocionales básicas, la representación del comportamiento humano sigue siendo una simplificación, ya que no abarca toda la diversidad de respuestas psicológicas que pueden manifestarse bajo estrés extremo. Por otra parte, el tiempo de

simulación observado fue menor que en situaciones reales, lo que evidencia la necesidad de calibrar mejor la interacción entre factores humanos y condiciones operativas, como la asistencia a personas vulnerables o la comunicación de instrucciones durante la emergencia.

En conjunto, estas limitaciones no restan validez al modelo, pero sí subrayan la importancia de futuras investigaciones orientadas a su refinamiento. Incluir datos empíricos más detallados, mejorar la representación de los estados emocionales y validar los resultados mediante simulacros controlados permitiría fortalecer su precisión y aplicabilidad. En definitiva, el modelo se consolida como una herramienta prometedora para la planificación de evacuaciones en entornos urbanos complejos, con un amplio potencial de evolución y mejora a medida que se integren nuevas fuentes de información y capacidades de simulación.

6. Trabajo Futuro

Una de las líneas más prometedoras para la evolución del modelo de evacuación propuesto es el desarrollo de enfoques híbridos que combinen la simulación basada en agentes con técnicas de modelado continuo o basadas en ecuaciones de flujo. Este tipo de integración permitiría capturar con mayor precisión tanto la dinámica individual de cada persona como los patrones colectivos de movimiento, optimizando la representación de fenómenos complejos como congestiones masivas, retrocesos o formaciones espontáneas de grupos.

Asimismo, la incorporación de sensores IoT (Internet of Things) abre la posibilidad de realizar simulaciones en tiempo real, basadas en datos actualizados sobre la ocupación, el flujo de personas y las condiciones ambientales dentro de los espacios urbanos o edificios complejos. Al integrar estos datos directamente en la simulación, sería posible ajustar dinámicamente las rutas de evacuación, identificar cuellos de botella en tiempo real y evaluar la efectividad de las estrategias de emergencia bajo condiciones cambiantes.

El uso combinado de modelos híbridos y sensores IoT también facilitaría la creación de sistemas inteligentes de gestión de emergencias, donde la planificación preventiva y la respuesta operativa puedan coordinarse de manera más eficiente. Esta línea de investigación tiene el potencial de transformar las simulaciones de evacuación, pasando de escenarios teóricos a aplicaciones prácticas con impacto directo en la seguridad de personas y la resiliencia de infraestructuras urbanas.

7. Conclusiones

Este artículo presenta un modelo de evacuación basado en las interacciones de los agentes, construido en torno a sólo cinco reglas básicas que rigen la forma en que las personas se mueven por espacios que contienen obstáculos y salidas. A diferencia de muchos modelos tradicionales, éste tiene en cuenta las reacciones humanas impredecibles en situaciones de estrés, como accidentes, histeria y la desorientación. El modelo ofrece una visión más realista del comportamiento de las personas en situaciones de emergencia. Estos criterios de comportamiento suelen pasarse por alto en las simulaciones convencionales, sin embargo, en esta propuesta desempeñan un papel crucial, mejorando la capacidad del modelo para reflejar la naturaleza caótica de las evacuaciones reales. Al captar tanto los aspectos estructurados del movimiento como los elementos más desordenados del comportamiento humano, el modelo ofrece perspectivas que no sólo son técnicamente sólidas, sino también aplicables a la planificación en el mundo real. En esta propuesta, se desarrolló la simulación de evacuación entorno al Hospital Fray Antonio Alcalde ubicado en Guadalajara, Jalisco, México. El modelo permitió no solo reproducir con precisión el

comportamiento humano durante una evacuación, sino también identificar oportunidades concretas de mejora en la infraestructura y los protocolos del recinto. A pesar de que el tiempo simulado fue considerablemente menor al observado en un simulacro real, esta diferencia es comprensible dadas las limitaciones propias del entorno físico, como la asistencia a pacientes o la comunicación de instrucciones. Más allá del tiempo, lo más valioso del modelo es su capacidad para detectar patrones de comportamiento emergentes, como zonas de congestión o agrupamientos espontáneos, que brindan información clave para el rediseño de espacios y la optimización de rutas de evacuación. Al incorporar variables emocionales y dinámicas adaptativas, el modelo se posiciona como una herramienta útil para la planificación de emergencias, contribuyendo a la toma de decisiones fundamentadas y al diseño de entornos más seguros y eficientes.

Referencias

[1] S. An, Z. Wang, and J. Cui, "Integrating Regret Psychology to Travel Mode Choice for a Transit-Oriented Evacuation Strategy," *Sustainability*, vol. 7, no. 7, pp. 8116–8131, 2015. DOI: 10.3390/su7078116

[2] V. Bayram, "Optimization models for large scale network evacuation planning and management: A literature review," *Surveys in Operations Research and Management Science*, vol. 21, no. 2, pp. 63–84, 2016. DOI: 10.1016/j.sorms.2016.11.001

[3] S. Ensing and C. Amrit, "Agent-Based Modelling and Simulation of Public Transport to Identify Effects of Network Changes on Passenger Flows," *Lecture Notes in Computer Science*, vol. 13741 LNCS, pp. 373–385, 2023. DOI: 10.1007/978-3-031-27199-1_37

[4] A. Fernández-Garza and H. Hernández-Vega, "Estudio de la movilidad peatonal en un centro urbano: un caso en Costa Rica," *Revista Geográfica de América Central*, vol. 1, no. 62, pp. 244–277, 2019. DOI: 10.15359/rgac.62-1.10

[5] D. Heard, G. Dent, T. Schifeling, and D. Banks, "Agent-based models and microsimulation," *Annual Review of Statistics and Its Application*, vol. 2, pp. 259–272, 2015. DOI: 10.1146/annurev-statistics-010814-020218

[6] D. Helbing, "Modeling of Socio-Economic Systems," in *Social Self-Organization: Understanding Complex Systems*, Berlin, Heidelberg: Springer, 2012. DOI: 10.1007/978-3-642-24004-1_1

[7] J. Hirtz, R. B. Stone, D. A. McAdams, S. Szykman, and K. L. Wood, "A functional basis for engineering design: Reconciling and evolving previous efforts," *Research in Engineering Design*, vol. 13, no. 2, pp. 65–82, 2002. DOI: 10.1007/s00163-001-0008-3

[8] Y. Hu, L. Zhang, Z. Zhang, Z. Li, and Q. Tang, "Metaheuristic and learning-oriented multi-objective artificial bee colony algorithm for energy-aware flexible assembly job shop scheduling problem," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108634, 2024. DOI: 10.1016/j.engappai.2024.108634

[9] M. Islas-Toski, E. Cuevas, M. Pérez-Cisneros, and H. Escobar, "Agent-Based Evacuation Modeling: Enhancing Building Safety in Emergency Scenarios," *Smart Cities*, vol. 7, no. 6, pp. 3165–3187, 2024. DOI: 10.3390/smartcities7060123

[10] Y. Li, M. Chen, X. Zheng, Z. Dou, and Y. Cheng, "Relationship between behavior aggressiveness and pedestrian dynamics using behavior-based cellular automata model," *Applied Mathematics and Computation*, vol. 371, p. 124941, 2020. DOI: 10.1016/j.amc.2019.124941

[11] Z. Liu, Z. Wang, L. Chen, T. Tang, and Z. Shen, "An agent-based approach to simulating evacuation in elderly care facilities with disabled older adults," *International Journal of Disaster Risk Reduction*, vol. 113, p. 104815, 2024. DOI: 10.1016/j.ijdrr.2024.104815

[12] C. M. Macal and M. J. North, "Tutorial on agent-based modeling and simulation," *Proceedings of the Winter Simulation Conference*, pp. 2–15, 2005. DOI: 10.1109/WSC.2005.1574234

[13] C. Le Page et al., "Agent-Based Modelling and Simulation Applied to Environmental Management," pp. 499–540, 2013. DOI: 10.1007/978-3-319-66948-9_22

[14] C. Percy et al., "The Contribution Of Human Psychology To Disaster Management: Mitigation, Advance Preparedness, Response And Recovery," *WIT Transactions on The Built Environment*, vol. 119, pp. 195–208, 2011. DOI: 10.2495/DMAN110181

[15] "Realizan simulacros en los Hospitales Civiles de Guadalajara," *Coordinación General de Comunicación Social*, Jul. 6, 2025. [En línea]. Disponible en: https://comsoc.udg.mx/noticia/realizan-simulacros-en-los-hospitales-civiles-de-guadalajara

[16] E. Ronchi, “A fire safety engineering perspective on crowd evacuation dynamics,” 2016. DOI: 10.1016/j.plrev.2016.05.014
[17] A. Seyfried, B. Steffen, and T. Lippert, “Basics of modelling the pedestrian flow,” *Physica A: Statistical Mechanics and Its Applications*, vol. 368, no. 1, pp. 232–238, 2006. DOI: 10.1016/j.physa.2005.11.052
[18] A. F. Siegenfeld and Y. Bar-Yam, “An Introduction to Complex Systems Science and Its Applications,” *Complexity*, vol. 2020, no. 1, p. 6105872, 2020. DOI: 10.1155/2020/6105872
[19] Y. Sun et al., “Social vulnerabilities and wildfire evacuations: A case study of the 2019 Kincade fire,” *Safety Science*, vol. 176, p. 106557, 2024. DOI: 10.1016/j.ssci.2024.106557
[20] J. Xie, N. Wei, and Q. Gao, “Assessing spatiotemporal population density dynamics from 2000 to 2020 in megacities using urban and rural morphologies,” *Scientific Reports*, vol. 14, no. 1, pp. 1–13, 2024. DOI: 10.1038/s41598-024-63311-5

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

Recibido 11 Agos 2025 *ReCIBE, Año 15 No. 1, mayo 2026*

Aceptado 30 Nov 2025

Método Eye tracking en la detección de somnolencia en conductores: Una revisión de literatura

Eye tracking method in detecting drowsiness in drivers: A literature review

Martin Laguna Estrada [1]
martin.laguna@itcelaya.edu.mx
José Manuel Delgado Pérez[1]
M2403045@itcelaya.edu.mx
Norma Verónica Ramírez Pérez[1]
norma.ramirez@itcelaya.edu.mx
Luis Alberto López Gonzalez[1]
luislao@celaya.tecnm.mx
Juan Ignacio Cerca Vázquez[1]
nacho@itcelaya.edu.mx
José Manuel Malagon Soldara[1]
salvador.malagon@itcelaya.edu.mx

1 Instituto Tecnológico de Celaya

Resumen
En la presente investigación se realiza una revisión de literatura relacionada con el uso del método de Eye Tracking (ET) o seguimiento ocular, para la detección, obtención y análisis de perfiles de somnolencia, el cual es uno de los factores de riesgo más presente en todo tipo de conductores, aumentando la probabilidad de causar accidentes. Para llevar a cabo esta revisión se utilizó la metodología PRISMA versión 2020, que permite realizar una investigación exhaustiva a través de la identificación, filtrado, elegibilidad e inclusión de los artículos que abordan la detección de somnolencia en los conductores. Como resultado del trabajo realizado, se elaboró una tabla de doble entrada de los estudios consultados en artículos científicos publicados en revistas de alto impacto, limitado a aquellos de reciente data período 2019 al 2025 con pertinencia y actualización de datos además de un aporte de evidencias empíricas en torno al objeto de estudio.

Palabras clave: Eye tracking; Somnolencia; Red neuronal; Deep learning; Procesamiento de imágenes.

Abstract
In the present study, a literature review was conducted on the use of the Eye-tracking method (ET) for detecting, obtaining, and analyzing driver fatigue profiles, one of the most prevalent risk factors among all kinds of drivers, which increases the likelihood of causing accidents. To carry out this review, the PRISMA 2020 methodology was employed, enabling a thorough investigation through the identification, screening, eligibility assessment, and inclusion of articles addressing driver drowsiness and fatigue detection. As a result, a two-way table was developed summarizing the consulted studies, drawn from scientific articles published in high impact journals, limited to those with recent publication dates from 2019 to 2025, relevant and up to date data, and empirical evidence contributions related to the study's focus.

Keywords: Eye tracking; Drowsiness; Neural network; Deep learning; Image processing.

1. Introducción

Las sociedades modernas según, Zhang, et al. (2017), citado por Jasim et al. (2022), afirman la importancia de garantizar a los pasajeros un viaje seguro, por lo que proteger la vida del conductor y el pasajero requiere una mejora en la seguridad al conducir. Como lo plantea You et al. (2020), entre los factores de riesgo que inciden en la seguridad se encuentra la somnolencia de los conductores. El propósito de mejorar la seguridad lleva a investigar y utilizar aplicaciones o sistemas de detección de somnolencia eficientes e innovadores como medida crucial para la prevención de accidentes.

En este mismo orden de ideas, Kumar et al. (2023) sostiene que la somnolencia del conductor ha sido una de las mayores causas de accidentes de tráfico ocasionando muertes y lesiones graves. Afirma, que se contabiliza un aproximado de más de un millón de personas que murieron en todo el mundo por este tipo de accidentes. Cuando un conductor se siente fuertemente cansado y con una necesidad alta de dormir, el conducir bajo este síntoma se le llama somnolencia del conductor y entre los factores que la producen se encuentran: la falta de sueño, consumo de alcohol o de medicamentos.

Chirra et al. (2019) explica que el trabajo realizado por agentes de policía y patrullaje ha revelado que mayoritariamente los accidentes registrados se producen entre las 2 y 5 am, siendo uno de los factores predisponentes el no dormir bien. Esto verifica la necesidad de mejorar las tecnologías que detectan la somnolencia para reducir el número de accidentes viales. Debido a las razones expuestas, se hace necesario identificar sistemas que alerten a los conductores permitiendo minimizar los accidentes de tránsito a la vez que se preserva la vida de las personas y las propiedades públicas. Uno de los métodos de detección de somnolencia, es el estudio del seguimiento ocular, el cual se ha utilizado

en múltiples campos de la investigación como una herramienta que permite obtener respuestas a diversas interrogantes.

La presente investigación se justifica por el creciente número de siniestros viales en los que la somnolencia del conductor provoca consecuencias adversas para pasajeros además de peatones y mobiliario presente en la zona. La OMS (2023) estima una cifra de entre 20 y 50 millones de personas que resultan heridas debido a accidentes vehiculares, de las cuales, al menos 1.19 millones de personas pierden la vida. A nivel nacional durante 2024 la organización INEGI (2024) reportó un total de 374,949 accidentes viales. La relación de este tipo de accidentes con la somnolencia ha sido vinculada en investigaciones como la desarrollada por la fundación de la seguridad vial AAA (2024) localizada en Estados Unidos, en la cual se llega a la conclusión de que, al menos en un 17.8 % de los accidentes donde alguno de los pasajeros pierde la vida, hubo un conductor con síntomas de somnolencia involucrado.

La relevancia social de la presente investigación se demuestra porque desde un nivel práctico, sirve para dar a conocer los métodos más eficaces que se han identificado en la actualidad, y a la vez, que permite concientizar a la sociedad sobre la importancia de prevenir accidentes, identificando la somnolencia del conductor por medio de la detección de movimiento ocular con métodos eficientes.

En el ámbito profesional, se justifica como una oportunidad y un desafío para los ingenieros en Sistemas, Mecatrónicos e ingenieros en Electromovilidad entre otros, para conocer y mejorar la metodología utilizada actualmente en la detección de la somnolencia en los conductores, siendo el presente estudio una base para mejorar las investigaciones realizadas. El objetivo de esta revisión fue analizar la efectividad del método ET para la obtención y análisis de perfiles de somnolencia en conductores, a través de Identificar las evidencias científicas actuales relacionadas a la utilización del este método.

2. Metodología

El presente trabajo de investigación se realizó bajo la modalidad de revisión sistemática, en la búsqueda de referentes teóricos que aportaron evidencias empíricas mediante la utilización del método ET para identificar la somnolencia en los conductores.

Los criterios de inclusión fueron establecidos con el propósito integrar investigaciones útiles para evaluar el panorama actual mediante el uso de la metodología ET para la detección de somnolencia. Se utilizó como criterio el tiempo de publicación entre 2019 y 2025 para contar con investigaciones de carácter reciente. Los trabajos incluidos debían contar con evidencia de los resultados obtenidos en la investigación que fueran congruentes con el resto de la metodología aplicada y debían provenir de publicaciones de alto impacto. Se excluyeron aquellos trabajos que no contaban explícitamente con un autor. En la tabla 1 se muestran todos los criterios tomados en cuenta para la selección de investigaciones.

Categoría	Criterios de Inclusión	Criterios de Exclusión
Publicación	Artículos científicos publicados en revistas de alto impacto.	Artículos sin identificación de autoría.
Temporalidad	Artículos publicados en el período 2019 - 2025.	Artículos publicados con anterioridad al año 2018.
Relevancia y Contenido	Pertinencia y actualización de los datos del estudio.	Investigaciones que no muestren evidencia explícita de los hallazgos encontrados.

	Artículos que aporten evidencias empíricas directas en torno al objeto de estudio.	

Tabla 1. Criterios de inclusión para las investigaciones seleccionadas

2.1 Pregunta P.I.C.O

Aunque la metodología PRISMA 2020 es una buena herramienta para nuestra investigación, consideramos importante la elaboración de la pregunta P.I.C.O aplicando el nemotécnico PICoR, el cual nos ayuda a dar claridad en la investigación, es centrada en los elementos más relevantes. Lo anterior se representa en la tabla 2.

P: Pacientes	Conductores
I: Intervención	Método Eye tracking
C: Comparación	Somnolencia y estado de alerta
O: Resultados	Efectividad del método

Tabla 2. Nemotécnico PICoR aplicado a la elaboración de la pregunta P.I.C.O

Se considera la siguiente pregunta PICO (patient, intervention, comparison, outcome) para la realización de la presente revisión sistemática: En conductores de vehículos ¿Cuál es la efectividad del método ET para detectar la somnolencia?

La figura 1 muestra en forma gráfica de barras el total de estudios por etapa.

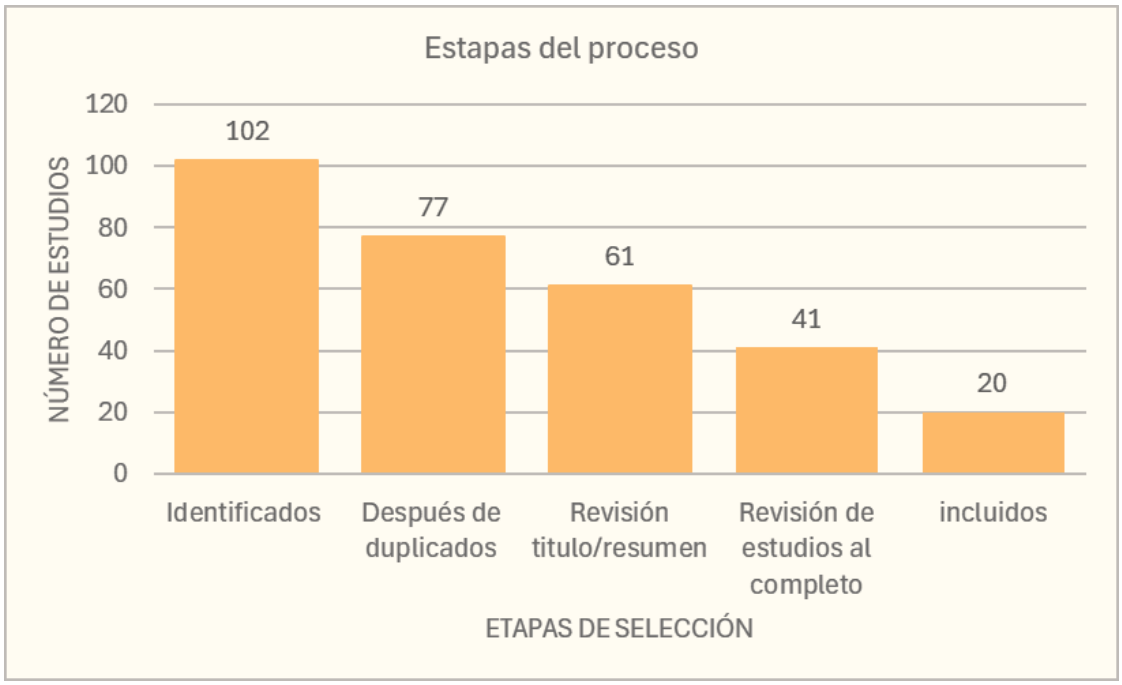


Figura 1. Gráfica derivada de la selección de literatura

2.2 Evaluación de la calidad

Se realizó una búsqueda en bases de datos consideradas de gran confiabilidad como: Pubmed, Cochrane, Scholar, Scielo y ScienceDirect entre otras. Las ecuaciones de búsqueda utilizadas fueron las siguientes: «Eye tracking * method to determine fatigue in drivers». En cuanto al uso de los booleanos (AND, OR Y NOT) los términos utilizados para realizar la búsqueda fueron: " Eye tracking " OR " determine fatigue in drivers "AND " method * lográndose encontrar 102 artículos vinculados con una combinación de operadores boléanos (AND), seleccionándose finalmente 20 literaturas como se muestra en el flujograma de artículos científicos.

2.3 Extracción de datos

Los datos considerados de relevancia para el estudio fueron extraídos con la colaboración de dos revisores distintos y categorizados tomando en cuenta el tema y objetivo de la investigación.

2.4 Análisis y síntesis de los datos

El estudio contó al final del proceso con 20 artículos científicos para la elaboración de la síntesis de datos, realizando un análisis a profundidad y la extracción de los temas analíticos. En base a los objetivos de investigación y las limitaciones impuestas para la inclusión de estudios se desarrolló la tabla 3 que reúne las principales características de los artículos seleccionados.

Título de articulo	Autor	Revista, Factor de Impacto País	Tamaño muestra Tipo de estudio	Duración de experimento
R2DS: A novel hierarchical framework for driver fatigue detection in mountain freeway	You F, Gong YB, Li XL, Wang HW. (2020).	Revista: Math Biosci Eng Factor de impacto: 0.48 País: China	No indica Tipo de estudio: Experimental	No indica
Driver Drowsiness Detection Using Gray Wolf Optimizer Based on Face and Eye Tracking	Jasim, S. S., Hassan, A. K. A. y Turner, S. (2022)	Revista: Aro-The Scientific Journal of Koya University País: Irak	17 personas de diversas nacionalidades Tipo de estudio: Experimental	No indica
Deep CNN: A Machine Learning Approach for Driver Drowsiness Detection Based on Eye State	Chirra, V. R. R., Uyyala, S. R. y Kolli, V. K. K. (2019)	Revista: Revue d'Intelligence Artificielle Factor de impacto: 0.28 País: India	20 conductores Tipo de estudio: Experimental	No indica
Driver Drowsiness Detection	Ablahd, A. Z., Aloraibi, A. Q., Dawwod, S. A.	Revista: Scalable Computing: Practice and Experience	3000 imágenes (1500 “drowsy” y 1500 “alert”)	No indica

	(2024)	Factor de impacto: 1.1 País: Rumania	Tipo de estudio: Experimental	
Automated Drowsiness Detection for Driver Safety: A Deep Learning-based Approach	Satapathy, S. K., Mori, H., Naik, K., Rajput, N. S. (2023)	Revista/conferencia: 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions País: India	No indica Tipo de estudio: Experimental	No indica
Una propuesta de Adaptación del Cuestionario de Síntomas de Fatiga Laboral	Vega-Valero, C. Z. (2019)	Revista: Revista de Psicología y Ciencias del Comportamiento de la Unidad Académica de Ciencias Jurídicas y Sociales País: México	250 trabajadores de distintas áreas (muestra final tras pilotaje) Tipo de estudio: Experimental	No indica
A CNN Based Wearable System for Driver Drowsiness Detection	Li, Y., Zhang, S., Zhu, G., Huang, Z., Wang, R., Duan, X., Wang, Z. (2023)	Revista: Sensors Factor de impacto: 3.4 País: China	3 personas Tipo de estudio: Experimental	Las pruebas se llevaron a cabo en periodos de 15 minutos, no se menciona la fecha ni hora de la realización del experimento.
Evaluation of a Fatigue Detector Using Eye Closure-Associated Indicators Acquired from Truck Drivers in a Simulator Study	Dziuda, Ł., Baran, P., Zieliński, P., Murawski, K., Dziwosz, M., Krej, M., Piotrowski, M., Stablewski, R., Wojdas, A., Strus, W., Gasiul, H., Kosobudzki, M. y Bortkiewicz, A. (2021)	Revista: Sensors Factor de impacto: 3.4 País: Polonia	8 personas Tipo de estudio: Experimental	Las pruebas se llevaron a cabo en un periodo de 48 minutos por conductor, no se mencionan fechas ni horas especificas
Driver drowsiness multi-method detection for vehicles with autonomous driving functions.	Beles, H., Vesselenyi, T., Rus, A., Mitran, T., Scurt, F. B, Tolea, B. A. (2024)	Revista: Sensors Factor de impacto: 3.4 País: Rumania	Una persona Tipo de estudio: Experimental	se realizaron cuatro jornadas de manejo con tiempos finales variables, todas realizadas en la mañana y concluidas antes de las 15:00pm

Driver Fatigue Detection Method Based on Eye States With Pupil and Iris Segmentation	Zhuang, Q., Zhang, K., Wang, J. y Chen, Q. (2020)	Revista: IEEE Access Factor de impacto: 3.367 País: China	Dataset de video con 4 videos de validación y 4 de prueba Tipo de estudio: Experimental	No indica
Eye Aspect Ratio for Real-Time Drowsiness Detection to Improve Driver Safety	Dewi, C., Chen, R.-C., Chang, C.-W., Wu, S.-H., Jiang, X., Yu, H. (2022)	Revista: Electronics Factor de impacto: 2.6 Países: Indonesia, Taiwán, Alemania, Reino Unido	15 663 fotogramas totales (TalkingFace: 5 000; Eyeblink8: 10 663) Tipo de estudio: Experimental	No indica
A CNN-Based Approach for Driver Drowsiness Detection by Real-Time Eye State Identification	Florez, R., Palomino-Quispe, F., Coaquira-Castillo, R. J., Herrera-Levano, J. C., Paixão, T., Alvarez, A. B. (2023)	Revista: Applied Sciences Factor de impacto: 2.5 Países: Perú, Brasil	6 800 imágenes extraídas de 6 videos del dataset NITYMED (4 760 entrenamiento; 1 020 validación; 1 020 prueba) Tipo de estudio: Experimental	No indica
A Deep-Learning Approach to Driver Drowsiness Detection	Ahmed, M. I. B., Alabdulkarem, H., Alomair, F., Aldossary, D., Alahmari, M., Alhumaidan, M., Alrassan, S., Rahman, A., Youldash, M., Zaman, G. (2023)	Revista: safety Factor de impacto: 2.1 País: Arabia Saudita	2 900 imágenes etiquetadas en 4 clases (open, closed, yawning, no yawning) Tipo de estudio: Experimental	No indica
Driving fatigue onset and visual attention: An electroencephalography-driven analysis of ocular behavior in a driving simulation task	Giorgi, A., Borghini, G., Colaiuda, F., Menicocci, S., Ronca, V., Vozzi, A., Rossi, D., Aricò, P., Capotorto, R., Sportiello, S., Petrelli, M., Polidori, C., Varga, R., Van Gasteren, M., Babiloni, F., Di Flumeri, G. (2024)	Revista: Behavioral Sciences Factor de impacto: 2.6 Países: Italia, España	19 conductores profesionales (9 furgonetas, 10 camiones; 18 hombres, 1 mujer; edad Tipo de estudio: Experimental	Conducción simulada de 1 hora por participante: 15 min de adaptación + 45 min de conducción monótona a 25 fps (1080p/720p) en ambiente de simulador, realizada en horario vespertino

Real-time driver monitoring system with facial landmark-based eye closure detection and head pose recognition	Kim, D., Park, H., Kim, T., Kim, W., Paik, J. (2023)	Revista: Scientific Reports Factor de impacto: 3.9 País: Corea del Sur	160 secuencias de vídeo infrarrojo Tipo de estudio: Experimental	No indica
Detection of human drowsiness by the vestibulo-ocular reflex compensating for heartbeat-induced head perturbations	Yamanaka, T., Hirata, Y. (2024)	Revista: Scientific Reports Factor de impacto: 3.9 País: Japón	15 participantes (12 hombres, 3 mujeres; 22 ± 1.8 años) Tipo de estudio: Experimental	Sesiones de reposo de 12 min con autoevaluaciones de somnolencia cada 2 min
Real-Time Deep Learning-Based Drowsiness Detection: Leveraging Computer-Vision and Eye-Blink Analyses for Enhanced Road Safety	Safarov, F.; Akhmedov, F.; Abdusalomov, A. B.; Nasimov, R.; Cho, Y. I. (2023)	Revista: Sensors Factor de impacto: 3.4 Países: República de Corea, Uzbekistán	No indica Tipo de estudio: Experimental	No indica
Exploration of Eye Fatigue Detection Features and Algorithm Based on Eye-Tracking Signal	Sun, W., Wang, Y., Hu, B., Wang, Q. (2024)	Revista: Electronics Factor de impacto: 2.6 País: China	19 participantes (y 20 000 muestras de datos) Tipo de estudio: Experimental	Sesiones de cuarenta minutos repartidas en 5 minutos por tarea realizada
Real-time driver drowsiness detection using transformer architectures: a novel deep learning approach	Hassan, O. F., Ibrahim, A. F., Gomaa, A., Makhlouf, M. A., Hafiz, B. (2025)	Revista: Scientific Reports Factor de impacto: 3.9 País: Egipto	178 619 imágenes totales (84 898 del MRL Eye, 66 521 del NTHU-DDD, 27 200 del CEW) Tipo de estudio: Experimental	No indica
A novel approach of drowsiness levels detection using Vis-Net combined with facial emotion	Phan, T.-C., Phan, A.-C. y Nguyen, N.-H.-Q. (2025)	Revista: Systems and Soft Computing Factor de impacto: 3.6 País: Vietnam	12 conductores (6 hombres y 6 mujeres), edades 22–50 años. Tipo de estudio: Experimental	cada sujeto realizó 4 sesiones de conducción de 20 min en distintas condiciones (iluminación diurna/nocturna, con/sin gafas o mascarilla), totalizando 48 sesiones

Tabla 3. Sistematización de las principales características de los estudios seleccionados para la revisión.

En total se seleccionaron 20 artículos, 19 pertenecientes a diversas publicaciones y uno perteneciente a un congreso internacional. Del total de artículos, 20 % pertenecen a la publicación Sensors, 15% a Scientific reports y 10% a Electronics, el resto de las publicaciones aportan un 5% de representación cada una.

Al realizar un análisis de los estudios presentados, se puede determinar la presencia de un sesgo significativo en la representación de la población y las condiciones experimentales que limita a los hallazgos encontrados. Al no contar con metodologías estandarizadas, es difícil analizar eficientemente la información, debido a que las metodologías tienen muestras, duración y condiciones generales de cada uno de los experimentos). Los estudios hacen uso de datasets de imágenes y técnicas de Deep learning que son considerados como técnicas innovadoras. Sin embargo, se considera que se deben utilizar muestras más grandes, estandarizar y detallar mejor los experimentos de somnolencia, realizar validaciones con simuladores o bien con datasets ya existentes, así como clarificar el uso de éstos.

Abordar estos sesgos permitirá construir una base de evidencia más robusta y aplicable para el desarrollo de sistemas efectivos de detección de somnolencia en conductores.

3. Resultados

Basado en el objetivo principal de esta revisión sistemática y en evidencias aportadas por la literatura, se generaron temas analíticos como: Somnolencia del conductor y métodos para detectar la somnolencia en el conductor.

3.1 Somnolencia del conductor

La somnolencia en conductores constituye una de las causas más comunes de accidentes de tráficos a nivel mundial You et al. (2020). Se describe como un sentimiento donde el individuo se percibe agotado, cansado y con poca energía motivado al desgaste físico por la ejecución de una actividad. Ricci, Chee, Lorandeau y Berger (2007, citado en Vega-Valero, 2019) describen la somnolencia como la consecuencia de múltiples factores que agotan la energía de un individuo. Las personas que la padecen reportan síntomas de agotamiento físico, dolores en la cabeza, músculos y con disminución de actividad física. Por las razones expuestas, se requiere que se realicen investigaciones que conduzcan a reconocer de forma inteligente y eficiente, la somnolencia en los conductores que son los responsables de su propia vida y la de sus pasajeros incluyendo personas del entorno donde éstos desempeñan su oficio.

3.2 Métodos para detección de somnolencia

Con la finalidad de mejorar la metodología e incrementar la precisión de detección de la somnolencia del conductor, se han desarrollado varios enfoques, presentando a continuación algunos de ellos.

En el marco de integrar y mejorar la detección de la somnolencia en conductores, You et al. (2020) propone una metodología que pretende medir características de precisión, velocidad y robustez que son consideradas elementos claves para la detección de síntomas de estas condiciones. En tal sentido, su propuesta incluye una totalidad de tres módulos para la extracción de:

- Rasgos faciales.
- Regiones oculares.
- Detección de somnolencia.

En el primero de los módulos se realiza un diseño de red neuronal convolucional en cascada profunda que conduce a la detección del rostro y a la vez localiza los puntos clave de los ojos. Seguidamente, se pasa a la construcción del submódulo que realiza el seguimiento facial aumenta la velocidad de algoritmo aplicándose la validación facial lo que mejora la estabilidad de la detección. Con el fin de garantizar la función ordenada de los submódulos, se diseñó un bucle de reconocimiento basado en la máquina de estados finitos, la cual extrae los rasgos faciales del conductor.

En el segundo módulo, se hizo una captura de las regiones de los ojos del conductor utilizando como característica geométrica la cara y los ojos. En el tercer módulo que permitió detectar la somnolencia, se aplicó el método de ajuste de elipse para obtener la forma de las pupilas del conductor. De acuerdo con la relación entre los ejes largo y corto de la elipse, decide el estado de los ojos ya sea abiertos o cerrados. Finalmente, se usó el PERCLOS, que se define calculando el número de ojos cerrados en un período, para determinar si se conduce en estado de somnolencia o no. Los resultados experimentales evidenciaron que la precisión integral de la detección de somnolencia es del 95.87%. La tasa promedio del algoritmo es de 32.29 ms/f en una imagen de 640×480 píxeles. Esta investigación contribuye a mejorar el diseño de otros estudios que tengan como propósito la generación de nuevos sistemas para detectar la somnolencia en los conductores.

Por su parte, Jasim et al. (2022) utilizó un método económico y no intrusivo con la finalidad de detectar la somnolencia del conductor mediante el monitoreo del rostro y los ojos, colocando una cámara de buena resolución y bajo costo que escribe (CAM 313). En su estudio concluyó que el reconocimiento del estado de los ojos es una tecnología muy precisa. Para prevenir accidentes, pueden detectar la somnolencia del conductor de forma temprana. A modo de comparación, descubrieron que el uso de un enfoque híbrido brinda una precisión del 91.18 % y el 97.06 %, mientras que el uso solo de una ANN brinda una precisión del 82.35 % y el 86.76 %, lo que hace que el enfoque híbrido sea superior al algoritmo de ANN. Aunque el método propuesto produjo resultados de gran precisión, aún hay margen para el cambio. Los enfoques conductuales combinados con estrategias fisiológicas pueden proporcionar una ventaja significativa en el futuro.

Siguiendo los métodos que tienen como objetivo detectar la somnolencia del conductor, Chirra et al. (2019) analizaron la función de los ojos como medio de detección. Con este análisis se determinaba si el individuo estaba somnoliento o alerta activando una alarma cuando el estado del ojo indicaba el primer estado. Se utilizó para este objetivo el algoritmo de detección de Viola Jones. En el proceso de desarrollo una red neuronal convolucional profunda aplicada para extraer características en la fase de aprendizaje. Se hizo uso de una capa SoftMax en el clasificador CNN para clasificar al conductor como dormido o no dormido. El sistema propuesto alcanzó una precisión del 96. 42 % evidenciándose que identificaba de forma eficaz el estado del conductor emitiendo una alarma cuando el modelo predice un estado de somnolencia. De los resultados anteriores, se afirma que es recomendable en el futuro utilizar el aprendizaje por transferencia para el mejoramiento del rendimiento del sistema.

Para verificar la efectividad de la metodología ET en la detección de somnolencia, Sun et al. (2024) utilizaron múltiples estructuras de red neuronal. El estudio fue realizado utilizando un dispositivo eye tracker diseñado previamente por los autores, con el objetivo de obtener parámetros como el tamaño de la pupila, la duración de parpadeo y la fijación de la vista. Se realizó un análisis ANOVA con el objetivo de interpretar los datos para relacionarlos con los síntomas de somnolencia. Además de los modelos anteriores se desarrollaron otros para verificar qué enfoque funcionaba mejor. Los dos enfoques de entrenamiento fueron: procesado de imágenes y con datos numéricos directamente extraídos del dispositivo eye tracker. Los resultados favorecieron a los modelos de

procesamiento de imágenes siendo el de mejor rendimiento el tipo de red auto encoder con un resultado de precisión de 82.4%. Además de lo anterior se desarrolló una red que combinaba ambos enfoques llamada Auto-Encoder Feature Union que alcanzó una precisión de 87.9%.

Otro método que vale la pena destacar fue el utilizado por Li et al. (2023), el cual se basa en un par de lentes que monitorean de cerca el estado de los ojos. Para el desarrollo de este sistema, se utilizó un armazón al cual se le colocó un tipo de lente especial denominado espejo caliente, con la característica de reflejar luz infrarroja mientras le permite el paso al resto de la luz. En el armazón se encuentran instalados una unidad procesadora de datos, una cámara que adquiere imágenes del ojo y una fuente de luz infrarroja.

Para realizar el reconocimiento del estado del ojo (si este se encuentra abierto o cerrado) primero se entrenó una red neuronal ligera con una estructura de bloques residuales tipo hourglass, a la cual se le introdujeron imágenes provenientes de la cámara ubicada en el armazón de los lentes, estas imágenes pasaron por un proceso de modificación en el cual se agregaron puntos al rededor del ojo conocidos como landmarks. Estos landmarks eran necesarios para permitir el rastreo de los movimientos del párpado mediante cálculos EAR (Eye Aspect Ratio). El método EAR calcula el área que delimitan los puntos agregados a las imágenes de entrenamiento permitiendo determinar datos como el porcentaje de cierre del ojo a partir de umbrales de valores de área. El uso de la red tipo hourglass fue importante debido a que, al capturar imágenes del ojo en ángulo mediante un reflejo, la forma de éste se distorsionaba dando como resultado una imagen difícil de analizar.

Se realizó una comparativa de funcionamiento con un dispositivo previamente existente denominado eyelink, la comparativa se llevó a cabo con dos métodos de detección de somnolencia, el porcentaje de cierre de ojo (PERCLOS) y la frecuencia de parpadeo, el segundo caso en particular muestra resultados similares para ambos dispositivos. Los datos finales indican que el dispositivo es capaz de detectar parpadeos un 93.9% de las ocasiones. El hecho de que la frecuencia de parpadeo haya sido similar en ambos dispositivos, demuestra que la capacidad de captura y procesado no necesariamente debe ser extremadamente alta para permitir la captura de movimientos oculares.

Una investigación que se enfoca en la utilización del eye aspecto ratio para monitoreo ocular, es la presentada por Dewi et al. (2022) la cual se vale de la librería Dlib para ubicar coordenadas específicas de áreas de interés en los ojos en forma de landmarks. Una vez ubicadas las coordenadas, se aplica la fórmula del EAR para lograr medir el área de apertura del ojo. Este autor experimentó con cuatro umbrales de EAR para definir el punto en el que un movimiento del ojo es un parpadeo (0.18, 0.20, 0.225 y 0.25). Los cuatro umbrales son analizados en tres bases de datos y se analizan los resultados a partir del área bajo la curva. Los resultados presentados por los escritores citados previamente, muestran que el sistema final cuenta con un 97% de precisión en la detección de parpadeos utilizando un umbral de 0.18. El uso del EAR puede presentar ciertas limitaciones, ya que se trata de una métrica matemática derivada de la geometría del ojo. Esto la hace sensible a factores como el ángulo de captura, la resolución de la imagen o pequeñas imprecisiones en la detección de los landmarks. Por este motivo, establecer umbrales adecuados para distinguir entre ojos abiertos y cerrados es fundamental, aunque la captura de imágenes puede requerir ajuste dependiendo del entorno y del sujeto.

Una investigación realizada por Dziuda et al. (2021), dio como resultado un prototipo de sensor detector de somnolencia, además de una evaluación de métricas de somnolencia que compara el porcentaje de cierre, la frecuencia de parpadeo y la velocidad de cierre de

ojo con el objetivo de verificar qué tan efectivas eran estas métricas, se desarrolló una cabina de pruebas en la cual ocho sujetos debían realizar un recorrido. El sensor de somnolencia utilizado consta de una cámara Ximea MQ013RG-E2 a la que se le agrega un objetivo Tamron de distancia focal fija que permite un buen grado de captura de luz infrarroja. Lo anterior es utilizado en conjunto con un anillo de leds infrarrojos para no perder imagen del sujeto en entornos de oscuridad.

Toda la cámara y el anillo de iluminación están integrados en una carcasa de aluminio anodizado, diseñada a medida para fijarse al tablero del simulador con un soporte articulado de tres grados de libertad. Para la detección de rostro se utilizó la librería FaceSDK y Template Matching con los rostros de cada uno con los sujetos de prueba.

Para la obtención de datos los ocho sujetos de prueba, se realizaron recorridos en el simulador con duración entre 40 y 48 minutos y se analizaban los datos obtenidos por métrica. La conclusión a la que llegó el estudio, fue que la velocidad de cierre de ojo es el parámetro más fiable, siendo capaz de detectar estados generales de somnolencia además de la diferencia en la severidad del estado conforme pasa el tiempo. La segunda mejor métrica fue el porcentaje de cierre, que se considera un indicador fiable del aumento de la somnolencia con el paso del tiempo. La frecuencia de parpadeo como indicador no mostró resultados particularmente efectivos como métrica de detección de somnolencia. La anterior información es vital para el desarrollo de cualquier sistema de detección de somnolencia debido a que indica cuáles son las métricas a las que se les debe dar más importancia en el análisis ocular.

Un sistema que vale la pena destacar es el desarrollado por Beles et al. (2024), el cual combina el monitoreo de señales bioeléctricas con el seguimiento ocular. En este caso se emplea electrooculografía junto con una red entrenada bajo una arquitectura de perceptrón multicapa para clasificar el estado del párpado. Para la parte de señales bioeléctricas se utilizan electrodos que se colocan alrededor del ojo. Las señales provenientes de estos electrodos se filtran y acondicionan para permitir una correcta interpretación de la señal, luego se buscan picos, los cuales representan el movimiento de los ojos y se utiliza lógica difusa para permitir clasificaciones de comportamiento.

Para el reconocimiento del estado de los ojos, se utiliza una red neuronal alimentada con imágenes de ojos tanto abiertos como cerrados. Se utilizó una arquitectura de perceptrón multicapa y un conjunto de entrenamiento de 200 imágenes.

Los resultados de ambos monitoreos alimentan una lógica difusa tipo Mamdani que extrae el nivel de alerta según la frecuencia/distribución de parpadeo mediante electrooculografía y la clasificación binaria de los ojos mediante cálculo de EAR. Para la parte bioeléctrica, el sistema mostró un grado alto de repetibilidad y consistencia para detectar parpadeos y movimientos oculares característicos del estado de somnolencia, mientras que para la red neuronal que detecta el estado del ojo se tiene una precisión del 90%. La lógica difusa es otra forma válida para el desarrollo de este tipo de sistemas, si bien es algo más restrictiva con sus criterios de evaluación, esto mismo la vuelve más fácil de controlar para desempeñar sus tareas de forma efectiva.

La investigación de Zhuang et al. (2020) presenta otro dispositivo de detección de somnolencia que funciona a partir de una red neuronal que detecta el porcentaje de apertura del ojo. Esto es posible utilizando la librería Dlib que ubica landmarks particulares en el rostro. El dispositivo emplea una red tipo U-Net ligera o de segmentación a nivel de píxel, para extraer mapas de características de pupila e iris con

el objetivo de que la red se enfoque en las regiones importantes del ojo. Se tiene un segundo módulo de decisión el cual determina el porcentaje de cierre. Las pruebas de precisión se llevaron a cabo utilizando un conjunto de datos obtenido del repositorio NTHU-DDD (National Tsing Hua University Drowsy Driver Detection Video Dataset). Como resultado, el método exhibió una precisión del 96.72%.

La investigación de Ahmed et al. (2023) muestra el entrenamiento de dos redes neuronales que detectan el estado de tanto los ojos como la boca a partir de un dataset preexistente. Una de las redes cuenta con una estructura ligera personalizada, compuesta por capas Conv2D, MaxPooling2D, Flatten y Dropout sin precarga de pesos y con una función de activación softmax para clasificación. La otra red está basada en la arquitectura VGG16 que consta de trece capas convolucionales y tres capas densas o completamente conectadas. A la red de arquitectura VGG16 se le precargaron pesos provenientes del dataset ImageNet a partir del método de transferlearning.

Las métricas principales para la validación de los resultados fueron precisión, recall y f1-score. En el caso del modelo que utiliza una estructura ligera personalizada se obtuvieron métricas casi perfectas (de 99%) para todas las salidas. El modelo de VGG16 sin embargo tuvo resultados precarios de al menos una de las métricas en cada salida. Cabe destacar que el modelo de estructura VGG16 se entrenó utilizando los pesos precargados de ImageNet por lo cual es posible que estos pesos afectaran el rendimiento de la red, sería conveniente revisar el rendimiento de VGG16 cuando no se utiliza transferlearning.

Otra investigación que presenta el uso de metodología de seguimiento ocular en conjunto con señales bioeléctricas es la realizada por Giorgi et al. (2024) que se basa en caracterizar los comportamientos de los ojos cuando un conductor entra en estado de somnolencia. Para lograr lo anterior se realizaron pruebas en un total de 20 conductores expertos, quienes manejaron en un simulador dos tipos de secciones, una más dinámica y otra más monótona. En esta investigación el papel del seguimiento ocular fue monitorear el desplazamiento de atención visual en conjunto con resultados de monitoreo electroencefalográfico para determinar cómo varían los comportamientos en un estado de somnolencia. La investigación demostró que existe una tendencia en los conductores en estado de somnolencia de enfocar su vista en entornos con menor nivel de información en lugar de zonas más importantes, lo cual favorece las probabilidades de accidentes viales. El descubrimiento de estas tendencias presentes en el comportamiento de los conductores da una mayor gama de posibilidades para el desarrollo de nuevos y más eficientes sistemas de seguimiento ocular que permitan la detección de síntomas de somnolencia.

Yamanaka y Hirata (2024) presentan una investigación con una metodología de detección de somnolencia algo menos conocida en comparación con PERCLOS o la duración de parpadeo. El enfoque principal es el monitoreo del reflejo vestíbulo-ocular, que permite que la imagen captada por el ojo se mantenga visualmente estable al realizar movimientos con la cabeza. Las investigaciones previas mencionan que la efectividad de este reflejo se reduce al encontrarse en estados de somnolencia por lo que su monitoreo podría ser empleado para detectar esta condición. Este monitoreo es posible debido a que se descubrió que los latidos del corazón generan pequeños movimientos que son capaces de activar este reflejo, haciendo que la metodología se vuelva más factible. Para comprobar lo anterior se utilizaron los parámetros de ganancia y variabilidad.

La ganancia determina si la velocidad de movimiento del ojo concuerda con la velocidad del movimiento de la cabeza. Una velocidad 100% igual implica una buena

capacidad de reflejo vestíbulo-ocular, y ésta se va reduciendo conforme aumenta el grado de somnolencia. Los escritores entrenaron una SVM (Support Vector Machine) para reconocer estados de somnolencia a partir de los parámetros de reflejo vestíbulo-ocular. Los resultados muestran una precisión en la detección de somnolencia de 80% para movimientos horizontales y 71.2% para movimientos verticales.

Otro sistema destacable es el desarrollado por Kim et al. (2023) el cual utiliza un modelo de red neuronal creado a partir de la estructura YOLOv7 que está basado en dos módulos que funcionan a partir de la detección facial. El primero de estos módulos es un detector de somnolencia que realiza la inferencia en función de la posición de las landmarks presentes en los ojos. El segundo módulo es un detector de atención del conductor que monitorea la posición de la cabeza para verificar que la visión siempre esté enfocada en el camino. Para la creación de este sistema, el laboratorio CANLAB desarrolló una cámara infrarroja personalizada que funciona como dispositivo de adquisición y procesamiento de imágenes. Esta cámara se encuentra ubicada en la parte trasera del volante del vehículo con el objetivo de reducir la distancia con el rostro del conductor, de modo que las imágenes no pierdan un grado alto de calidad. Esta estrategia es muy efectiva, debido a que los autores reportan un grado perfecto de precisión(100%) en el modelo de detección de rostro independientemente del grado de iluminación.

Un elemento para tomar en cuenta en el desarrollo del dispositivo es el uso de YOLOv7 debido a que esta arquitectura se relaciona con modelos pesados que no son tan sencillos de ejecutar en sistemas embebidos, prueba de esto es que utilizando una jetson Xavier, que es una tarjeta de desarrollo particularmente potente y diseñada para trabajar con modelos de IA, lograron un rendimiento de 10fps que en algunos casos se considera apenas aceptable para aplicaciones de video en tiempo real. Sin embargo, el sistema final es capaz de funcionar a una taza de 30 fps, esto se debe a la cámara especialmente diseñada para el sistema por el laboratorio CANLAB.

Siguiendo con el uso de estructuras con un alto consumo de recursos para reconocimiento facial, Hassan et al. (2025) utiliza la arquitectura transformer con el objetivo de detectar el estado en el que se encuentran los ojos (abierto o cerrado). Se utilizaron dos arquitecturas tipo transformer, Vit y swim. Con el objetivo de validar el uso de las arquitecturas en el proceso de detección de somnolencia, se realizó una comparación con redes convolucionales que utilizaban transferlearning. Para la evaluación se utilizó el dataset MLR en todos los entrenamientos. Ambas redes transformer presentaron una precisión del 99%, dando el mejor resultado de las estructuras de red utilizadas en la comparación.

Otro uso de la arquitectura Vision transfomer es el presentado por Phan et al. (2025) en donde utilizan un dataset original para entrenar múltiples modelos de detección de somnolencia. Al igual que en el artículo presentado por Hassan et al. (2025), se utiliza la arquitectura vision transformer además de múltiples redes CNN como punto de comparación. En lugar de realizar la detección y segmentación de las partes importantes del rostro, ya sean los ojos o boca, el modelo es entrenado utilizando el rostro al completo, de esta forma, en lugar de utilizar porcentajes o velocidad de cierre se utilizan cinco niveles que describen el estado del conductor: Alerta total, Leve somnolencia, somnolencia moderada, somnolencia severa y estado de sueño o pérdida completa de conciencia. El modelo vision transformer alcanzó una precisión al menos de 98% para todos los niveles, logrando un algoritmo bastante efectivo para realizar la detección de estados de somnolencia.

El avance de los modelos de reconocimiento y detección facial ha generado múltiples herramientas que pueden ser muy útiles en la detección de síntomas de somnolencia, un buen ejemplo es el framework mediapipe que permite realizar una gran variedad de tareas

referentes al reconocimiento facial que Safarov et al. (2023) aprovechan para el desarrollo de un algoritmo con la capacidad de detectar síntomas de somnolencia utilizando como indicadores la velocidad de cierre de ojos, la frecuencia de cierre, los bostezos y el cambio de posición de la cabeza. La detección de cada uno de estos síntomas trabaja en función a los umbrales que establece el autor, para determinar el porcentaje de cierre de los ojos utiliza EAR, para la detección de bostezos MAR (Mouth Aspect Radio) el cual tiene un funcionamiento similar al EAR pero aplicado en la boca, en el caso de la inclinación de la cabeza se mide el desplazamiento que han tenido la frente y la barbilla con respecto al frame anterior. Como métricas de precisión se obtienen: 95.8 % de en detección de ojos cerrados, 97 % en detección de ojos abiertos, 84 % en detección de bostezos y 98 % y 100 % en inclinación de cabeza a derecha e izquierda, respectivamente. El desarrollo de este de este algoritmo permite confirmar la viabilidad del uso de frameworks como mediapipe con el objetivo de obtener herramientas que permitan detectar los síntomas de somnolencia.

Un sistema de detección de somnolencia particularmente completo es el presentado por Florez et al. (2024) el cual plantea un sistema embebido que detecta síntomas de somnolencia a partir de métricas estrictas de porcentaje de cierre de ojo y conteo de bostezos, utilizan la librería mediapipe con el objetivo de segmentar las regiones de interés que en este caso son los ojos (ambos en una sola salida) y la boca. La arquitectura utilizada es una personalizada que los autores denominan DD-AI, que cuenta con un total de ocho capas. Además de lo anterior, se utilizaron arquitecturas VGG, ResNet e Inception. El sistema embebido utilizado es una jetson Nano debido a sus capacidades para trabajar con modelos de inteligencia artificial. Los resultados de entrenamiento de todos los modelos empleados muestran métricas de precisión para la detección de casos de somnolencia del 98% para todos los modelos empleados, incluyendo la estructura DD-AI. La investigación anterior demuestra que las redes ligeras pueden realizar un buen trabajo en la detección de estados de somnolencia.

Finalmente, Ablahd et al. (2024) diseñó un sistema especial inteligente, fiable y preciso, utilizando el lenguaje Python 3.6 para Windows, para crear un sistema de alerta con la finalidad de detectar la somnolencia de los conductores. Este sistema se consideró primordial para la reducción de accidentes de tráfico. El sistema implementaba un detector en tiempo real que controlaba los estados de los conductores durante la conducción. La metodología usada utilizó cámaras inteligentes de 16 megapíxeles para garantizar que las fotografías capturadas tuvieran una alta calidad. Se procedió a recopilar el conjunto de datos del conductor en diferentes estados incluyendo los de alerta y somnolencia.

El conjunto de datos recopilados se procesó extrayendo todas las características relevantes, como el movimiento de la cabeza, los bostezos y el cierre de los ojos, que se utilizaron para identificar la somnolencia del conductor. Las bibliotecas de Python, como TensorFlow, OpenCV, Keras y Pygame, se utilizan para extraer todas las características anteriores. El algoritmo de Viola-Jones se utiliza para detectar y extraer la región ocular de la imagen de la cara en el sistema propuesto. Se usó un algoritmo de máquina de vectores de soporte (SVM) para clasificar entre conductores somnolientos y no somnolientos. El sistema de prueba se evaluó en el mundo real para garantizar que sea confiable y robusto, mostrando un alto rendimiento y una precisión de aproximadamente el 99.1%.

4. Discusión

La presente revisión tuvo como propósito analizar la efectividad del método ET para la obtención y análisis de perfiles de somnolencia en conductores. Desde la perspectiva

de los autores se evidenció, que munal You et al. (2020); Ricci, Chee, Lorandeau y Berger (2007) como citado en Vega-Valero, (2019), consideran que la somnolencia se manifiesta en los conductores como el deterioro de sus habilidades debido a la ejecución de su tarea. En tal sentido, valoran que se realicen investigaciones que lleven a identificar metodologías inteligentes para reconocer la somnolencia en los conductores y de esta manera salvaguardar sus vidas y la de sus pasajeros.

Mientras, que You et al. (2020) propuso una metodología que se enfatizaba en la precisión, la velocidad y la robustez dividida en módulos donde se incluía el método ET, misma demostró gran efectividad. En este mismo objetivo, Jasim et al. (2022) basó su estudio en un método económico que evitaba ser intrusivo y pretendió detectar la somnolencia del conductor mediante el monitoreo del rostro y los ojos, colocando una cámara de buena resolución (CAM 313). El estudio demostró presión respecto al objetivo propuesto y recomendó a futuro utilizar enfoques conductuales combinados.

En este mismo orden de ideas, Siguiendo los métodos que tienen como objetivo detectar la somnolencia en conductores, Chirra et al. (2019) utilizó el algoritmo de detección de Viola Jones, desarrolló una red neuronal convolucional profunda apilada para extraer características en la fase de aprendizaje. Su estudio alcanzó una precisión de 96.42 % demostrando eficacia para detectar signos de somnolencia en conductores. Por su parte, Ablahd et al. (2024) diseñó un sistema de alerta que se destacó por el elevado rendimiento y precisión ubicada en 99.1% en la detección de somnolencia en conductores.

Los trabajos presentados por Beles et al. (2024), Kim et al. (2023), Zhuang et al. (2020) y Florez et al. (2023) demuestran que es posible desarrollar sistemas que funcionen en tiempo real con una taza de frames lo suficientemente estable como para permitir el monitoreo del proceso de parpadeo completo, además de brindar estrategias para que estos sistemas funcionen aun con una cantidad baja de iluminación.

Tanto Safarov et al. (2023) como Florez et al. (2023) demuestran las ventajas de utilizar modelos o frameworks preexistentes con el fin de mejorar y facilitar los procesos de detección y segmentación de zonas faciales de interés para el reconocimiento de síntomas de somnolencia.

Así mismo, vale la pena destacar los esfuerzos de múltiples estudios que se dedicaron a la validación de metodologías de detección de somnolencia. Sun et al. (2024) validaron múltiples parámetros biométricos para la detección de cambios en zonas de interés, mientras que Dewi et al. (2022) se dedicaron a buscar umbrales de “eye aspecto radio” que se pudieran aplicar de forma general. Dziuda et al. (2021) puso a prueba los métodos PERCLOS, velocidad y frecuencia de parpadeo para determinar si eran marcadores de somnolencia efectivos. Por último, Yamanaka e Hirata (2024) se centraron en el monitoreo del reflejo vestíbulo-ocular con el fundamento de que el propio latido del corazón genera pequeños movimientos que lo activan.

Las investigaciones de Hassan et al. (2025), Phan et al. (2025) y Kim et al. (2023) se enfocaron en el desarrollo de redes a partir de arquitecturas que dan como resultado modelos más precisos, pero también algo más demandantes a nivel de hardware, siendo en el caso de los primeros dos artículos la arquitectura transformer y en el tercero la arquitectura yolov7. En los primeros dos artículos las métricas de precisión fueron muy favorables, aunque no se llegó a comprobar el rendimiento en sistemas embebidos, para el tercer caso se diseñó un equipo especializado que recolectaba imágenes y las procesaba a la vez dando resultados favorables de precisión y una buena taza de frames.

Ahmed et al. (2023) desarrollaron una estructura de red neuronal propia con un menor gasto computacional, la cual fue capaz de igualar en precisión a estructuras de red preestablecidas a pesar de que éstas utilizaban transfer learning.

Li et al. (2023) presentaron un enfoque en el cual la recolección de imágenes ocurría extremadamente cerca de las zonas de interés. Utilizando una estructura de gafas, fueron capaces de obtener un sistema con un grado aceptable de precisión que podía sacar imágenes detalladas de los ojos sin tener que lidiar con los problemas que conlleva la distancia entre la cámara y el conductor evitando problemas de ruido.

5. Conclusiones

Los hallazgos demostraron que el método ET para la obtención y análisis de perfiles de somnolencia en conductores, muestra una gran efectividad visible en las evidencias científicas analizadas, lo que lleva a los encargados de diseñar mejoras en la metodología de estos sistemas a seguir profundizando en las investigaciones para la detección de la somnolencia en los conductores, ya que se ha convertido una problemática que afecta las vidas de miles de personas a lo largo del mundo. El conjunto de trabajos analizados brinda un panorama claro del uso y avance de esta tecnología en la actualidad, permitiendo identificar cuáles son las áreas de mejora en un área de estudio que podría salvar innumerables vidas. La presente investigación ofrece un panorama con respecto al estado más actual de la tecnología ET en el ámbito de la detección de somnolencia, demostrado en el 65% de los trabajos incluidos que datan de 2023 a 2025. Esto en conjunto con el criterio de selección basado en fuentes de acceso abierto pone de manifiesto que la investigación no solo está al día con los avances metodológicos más recientes, sino que además ofrece una síntesis reproducible y utilizable inmediatamente por cualquier lector.

Uno de los motivos para la realización de la investigación fue el de revisar antecedentes recientes del uso de la tecnología ET en la detección de somnolencia, con el fin de conocer su estado actual y las áreas de mejora existentes en las implementaciones de estas alternativas. Lo anterior se aplicará en el diseño de un sistema de detección de somnolencia que aplicará el método ET tomando en cuenta las variables utilizadas en el monitoreo de esta condición.

6. Referencias

[1] AAA Foundation for Traffic Safety, Drowsy driving in fatal crashes, United States, 2017–2021 (Research Brief, marzo 2024). Disponible en: https://aaafoundation.org/wp-content/uploads/2024/03/202304-AAAFTS-Drowsy-Driving-Countermeasures.pdf

[2] A. Z. Ablahd, A. Q. Aloraibi y S. A. Dawwod, "Driver drowsiness detection," Scalable Computing: Practice and Experience, vol. 25, no. 5, 2024. https://doi.org/10.12694/scpe.v25i5.3046

[3] M. I. B. Ahmed, H. Alabdulkarem, F. Alomair, D. Aldossary, M. Alahmari, M. Alhumaidan, S. Alrassan, A. Rahman, M. Youldash y G. Zaman, "A deep-learning approach to driver drowsiness detection," Safety, vol. 9, no. 3, p. 65, 2023. https://doi.org/10.3390/safety9030065

[4] H. Beles, T. Vesselenyi, A. Rus, T. Mitran, F. B. Scurt y B. A. Tolea, "Driver drowsiness multi-method detection for vehicles with autonomous driving functions," Sensors, vol. 24, no. 5, p. 1541, 2024. https://doi.org/10.3390/s24051541

[5] V. Chirra, U. S. Reddy y V. KishoreKolli, "Deep CNN: A machine learning approach for driver drowsiness detection based on eye state," Revue d'Intelligence Artificielle, vol. 33, pp. 461–466, 2019. https://doi.org/10.18280/ria.330609

[6] C. Dewi, R.-C. Chen, C.-W. Chang, S.-H. Wu, X. Jiang y H. Yu, "Eye aspect ratio for real-time drowsiness detection to improve driver safety," Electronics, vol. 11, no. 19, p. 3183, 2022. https://doi.org/10.3390/electronics11193183

[7] Ł. Dziuda et al., "Evaluation of a fatigue detector using eye closure-associated indicators acquired from truck drivers in a simulator study," Sensors, vol. 21, no. 19, p. 6449, 2021. https://doi.org/10.3390/s21196449

[8] R. Florez, F. Palomino-Quispe, R. J. Coaquira-Castillo, J. C. Herrera-Levano, T. Paixão y A. B. Alvarez, "A CNN-based approach for driver drowsiness detection by real-time eye state identification," Applied Sciences, vol. 13, no. 13, p. 7849, 2023. https://doi.org/10.3390/app13137849

[9] A. Giorgi et al., "Driving fatigue onset and visual attention: An electroencephalography-driven analysis of ocular behavior in a driving simulation task," Behavioral Sciences, vol. 14, no. 11, p. 1090, 2024. https://doi.org/10.3390/bs14111090

[10] O. F. Hassan, A. F. Ibrahim, A. Gomaa, M. A. Makhlouf y B. Hafiz, "Real-time driver drowsiness detection using transformer architectures: A novel deep learning approach," Scientific Reports, vol. 15, p. 17493, 2025. https://doi.org/10.1038/s41598-025-02111-x

[11] Instituto Nacional de Estadística y Geografía (INEGI), "Accidentes de tránsito," Recuperado el 26 de octubre de 2025. Disponible en: https://www.inegi.org.mx/temas/accidentes/

[12] S. S. Jasim, A. K. Abdul Hassan y S. Turner, "Driver drowsiness detection using Gray Wolf optimizer based on face and eye tracking," ARO – The Scientific Journal of Koya University, vol. 10, no. 1, pp. 49–56, 2022. https://doi.org/10.14500/aro.10928

[13] D. Kim, H. Park, T. Kim, W. Kim y J. Paik, "Real time driver monitoring system with facial landmark based eye closure detection and head pose recognition," Scientific Reports, vol. 13, p. 18264, 2023. https://doi.org/10.1038/s41598-023-44955-1

[14] S. Kumar Satapathy, H. Mori, K. Naik y N. S. Rajput, "Automated drowsiness detection for driver safety: A deep learning-based approach," en 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), pp. 774–779, 2023. https://doi.org/10.1109/CISES58720.2023.10183506

[15] Y. Li et al., "A CNN based wearable system for driver drowsiness detection," Sensors, vol. 23, no. 7, p. 3475, 2023. https://doi.org/10.3390/s23073475

[16] Organización Mundial de la Salud, Traumatismos causados por el tránsito, 2023. https://www.who.int/es/news-room/fact-sheets/detail/road-traffic-injuries

[17] T.-C. Phan, A.-C. Phan y N.-H.-Q. Nguyen, "A novel approach of drowsiness levels detection using Vis Net combined with facial emotion," Systems and Soft Computing, vol. 7, p. 200288, 2025. https://doi.org/10.1016/j.sasc.2025.200288

[18] F. Safarov, F. Akhmedov, A. B. Abdusalomov, R. Nasimov y Y. I. Cho, "Real-time deep learning-based drowsiness detection: Leveraging computer vision and eye blink analyses for enhanced road safety," Sensors, vol. 23, no. 14, p. 6459, 2023. https://doi.org/10.3390/s23146459

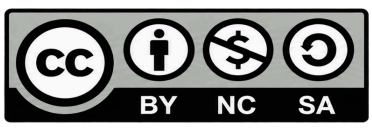
[19] W. Sun, Y. Wang, B. Hu y Q. Wang, "Exploration of eye fatigue detection features and algorithm based on eye-tracking signal," Electronics, vol. 13, no. 10, p. 1798, 2024. https://doi.org/10.3390/electronics13101798

[20] C. Z. Vega-Valero, "Una propuesta de adaptación del cuestionario de síntomas de fatiga laboral," Revista de Psicología y Ciencias del Comportamiento de la Unidad Académica de Ciencias Jurídicas y Sociales, vol. 10, no. 1, pp. 74–86, 2019. https://doi.org/10.29059/rpcc.20190602-82

[21] T. Yamanaka y Y. Hirata, "Detection of human drowsiness by the vestibulo-ocular reflex compensating for heartbeat-induced head perturbations," Scientific Reports, vol. 14, p. 23737, 2024. https://doi.org/10.1038/s41598-024-74202-0

[22] F. You, Y. B. Gong, X. L. Li y H. W. Wang, "R2DS: A novel hierarchical framework for driver fatigue detection in mountain freeway," Mathematical Biosciences and Engineering, vol. 17, no. 4, pp. 3356–3381, 2020. https://doi.org/10.3934/mbe.2020190

[23] Q. Zhuang, K. Zhang, J. Wang y Q. Chen, "Driver fatigue detection method based on eye states with pupil and iris segmentation," IEEE Access, vol. 8, pp. 173440–173449, 2020. https://doi.org/10.1109/ACCESS.2020.3025818



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

COMPUTACIÓN E INFORMÁTICA

Recibido 02 Nov 2025 *ReCIBE, Año 15 No. 1, mayo 2026*

Aceptado 22 Abr 2026

Modelo de sostenibilidad humana para productos de software con infraestructura limitada: Caso Providencia – Colombia

Human Sustainability Model for Software Products with Limited Infrastructure: The Providencia – Colombia Case Human Sustainability Model for Software Products with Limited Infrastructure: The Providencia – Colombia Case

Paola Noreña[1]
panorenaaac@eafit.edu.co
Elizabeth Suescún[1]
esuescu1 @eafit.edu.co
Santiago Manco[1]
smancom@eafit.edu.co
Santiago Idárraga[1]
sidarragac@eafit.edu.co

[1] *Universidad EAFIT, Medellín, Colombia.*

Resumen
La sostenibilidad del software se basa en integrar prácticas sostenibles en su desarrollo. Una de sus dimensiones es la sostenibilidad humana, que considera el impacto en los equipos de desarrollo y usuarios. Cuando esta infraestructura es insuficiente o presenta restricciones, se denomina infraestructura tecnológica limitada, lo que dificulta el desarrollo de software sostenible y su impacto positivo en la sociedad. Los estudios en sostenibilidad del software priorizan la infraestructura de las aplicaciones, dejando de lado la sostenibilidad humana y su impacto en comunidades vulnerables con infraestructura limitada. Estas comunidades enfrentan desafíos de conectividad, acceso a recursos tecnológicos, falta de habilidades en tecnologías de la información (TI) y habilidades blandas para el trabajo, lo que limita sus oportunidades en su contexto. En este artículo se propone un modelo de sostenibilidad humana en productos de software con infraestructura limitada. Este modelo busca incorporar prácticas sostenibles en el ciclo de vida del desarrollo de software, priorizando el aspecto humano y el acceso equitativo. El modelo se aplicó en la isla de Providencia-Colombia, una comunidad con características vulnerables que enfrenta desafíos en su infraestructura como recursos energéticos limitados, conectividad a internet inestable, acceso restringido a equipos y plataformas tecnológicas y falta de habilidades en TI. Este modelo fomenta el desarrollo de software sostenible desde la ingeniería de software, promueve el acceso a plataformas digitales, la equidad social y laboral mediante la formación en habilidades digitales, y el uso responsable de productos de software sostenible, impulsando así el desarrollo económico en las comunidades impactadas.

Palabras Clave: Infraestructura tecnológica limitada, sostenibilidad humana, sostenibilidad del software.

Abstract
Software sustainability is based on integrating sustainable practices into its development. One of its dimensions is human sustainability, which considers the impact on development teams and users. When this infrastructure is insufficient or restricted, it is called limited technological infrastructure, which hinders the development of sustainable software and its positive impact on society. Studies in software sustainability prioritize application infrastructure, neglecting human sustainability and its impact on vulnerable communities with limited infrastructure. These communities face challenges related to connectivity, access to technological resources, a lack of information technology (IT) skills, and a lack of soft skills for work, which limit their opportunities in their context. This article proposes a human sustainability model for software products with limited infrastructure. This model seeks to incorporate sustainable practices into the software development lifecycle, prioritizing the human aspect and equitable access. The model was applied on the island of Providencia, Colombia, a vulnerable community facing infrastructure challenges, including limited energy resources, unstable internet connectivity, restricted access to equipment and technological platforms, and a lack of IT skills. This model fosters the development of sustainable software from software engineering, promotes access to digital platforms, social and labor equity through training in digital skills, and the responsible use of sustainable software products, thus boosting economic development in the impacted communities.

Keywords: Human sustainability, Limited technological infrastructure, Software sustainability

1. Introducción

La sostenibilidad y el enfoque humano son dos temas relevantes en la industria 5.0, la cual es una evolución de la industria 4.0, donde enfatiza la colaboración entre humanos y máquinas inteligentes, priorizando la sostenibilidad (Breque et al., 2021). El software juega un papel importante en la sostenibilidad para generar eficiencia energética, ya que su comportamiento tiene una influencia significativa en las características de ahorro de energía integradas a las plataformas y tecnologías. La sostenibilidad del software se basa en prácticas que permiten incorporar la sostenibilidad desde tres dimensiones (Calero et al., 2021): software sostenible (productos de software que involucran procesos sostenibles en el ciclo de vida del software o en software que apoya la sostenibilidad en otras áreas; Noreña et al., 2024a), sostenibilidad humana (aspectos sociológicos y psicológicos de la comunidad del software, es decir, usuarios, equipos y sociedad; Noreña et al., 2024b) y sostenibilidad económica (aspectos financieros del proyecto). La infraestructura tecnológica se refiere al acceso y uso de tecnologías de información, hardware, conectividad de internet y recursos digitales que son imprescindibles para el desarrollo de software. Esta infraestructura suele ser suficiente cuando tiene los recursos requeridos o limitada cuando tiene restricción a estos recursos (Torres et al., 2010). Según Calero et al. (2021), para las tres dimensiones de sostenibilidad del software, algunos de los trabajos desde la literatura han sido en software verde que estudia el uso y mantenimiento del software con el objetivo de reducir su impacto ambiental, especialmente en términos de consumo de energía y emisión de gases de efecto invernadero, entre otros aspectos.

Algunos autores trabajaron en sostenibilidad humana: Semerikov et al. (2020) y Heldal et al. (2024) presentaron competencias sostenibles para ingeniería de software. Grundy et al. (2020) desarrollaron un laboratorio ágil y de co-creación entre los ingenieros de software y usuarios con extracción de aspectos humanos desde los requisitos. Amrutha y Geetha (2020) y Ramautar et al. (2021) definieron desafíos en esta dimensión: (i) capacitación de habilidades de TI, especialmente en desarrollo ágil, (ii) comercio transfronterizo en temas de empleabilidad y comunicación y (iii) barreras culturales. Un caso de estudio aplicado a Kenya en competencias de desarrollo permitiendo activar su economía (Were, et al., 2020) y un diseño de una arquitectura para el ambiente laboral de equipos de desarrollo fue realizado por Alavi, et al. (2018). También otros autores han trabajado en los siguientes enfoques desde software verde: Restrepo (2021) propuso un mapeo sistemático de la literatura guiado por el ciclo de vida del desarrollo de software con enfoque en la sostenibilidad, Restrepo et al. (2023) propuso un *framework* para el diseño arquitectural de sistemas ciber-físicos que contempla de manera transversal la sostenibilidad para el agro. Restrepo (2025) presenta un diseño arquitectural para sistemas ciber-físicos sostenibles. Restrepo et al. (2024) propone una ontología en la sostenibilidad del software a partir de atributos de calidad. Calero et al. (2021) y otros autores del libro presentaron procesos de medida para el software verde, un *framework* para pruebas de eficiencia energética del software, impacto de la sostenibilidad del software en la organización, entre otros. Los siguientes son enfoques desde la sostenibilidad económica: Peláez y Corrales (2022) proponen un *framework* de prácticas para el desarrollo sostenible en la dimensión económica y Ojameruaye y Bahsoon (2021) presentaron un enfoque basado en esta dimensión para medir requisitos sostenibles. Si bien se han logrado avances en las tres dimensiones de la sostenibilidad, persiste la necesidad de enfoques que incorporen el componente humano en las prácticas sostenibles del desarrollo de software.

También, se encuentran diversos trabajos que se enfocan en mejorar la conectividad de diferentes comunidades: la transformación de la infraestructura digital en África abrió las oportunidades de acceso digital mediante conectividad con financiamiento de diferentes empresas

de TI como Meta, Google, Orange, entre otras (White & Case, 2022). Países latinoamericanos también vienen haciendo esfuerzos en adoptar TI para comunidades rurales donde no tienen acceso a internet, como el proyecto Internet Society-Capítulo Panamá, el cual fomenta el desarrollo de la infraestructura con acceso a internet trabajando en cerrar las brechas de conectividad y permitiendo el involucramiento de las comunidades mediante capacitaciones con expertos técnicos para planificar, construir y mantener la infraestructura de interconexión (Internet Society Capítulo Panama, s.f.). Sin embargo, no atienden temas en desarrollo de software. Los siguientes proyectos se enfocan en trabajar por los jóvenes de comunidades vulnerables y sus habilidades para el trabajo: (i) proyecto parceros, con el propósito de dar oportunidades a jóvenes para que tengan proyectos de vida, educación, formación laboral, proyectos de emprendimiento. "Durante el acompañamiento se ofrece intervención psicosocial, formación del ser, orientación vocacional, bolsa de empleo, formación para el trabajo, materiales de apoyo, transporte y refrigerios" (EL TIEMPO, 2024) y (ii) la Escuela de Código comuna 13 en Medellín-Colombia donde diferentes empresas se unieron para capacitar a jóvenes de estratos bajos en el desarrollo de habilidades en desarrollo de código e inteligencia emocional, con el objetivo de disminuir las brechas de desigualdad social y combatir la pobreza (Colombia Crea, s.f.).

Aunque la mayoría de estos trabajos en sostenibilidad del software se centran en la infraestructura tecnológica de las aplicaciones, desde el ciclo de vida del desarrollo de software (sus herramientas, procesos, metodologías y roles) rara vez consideran la sostenibilidad y la eficiencia energética como un objetivo (Calero et al., 2021). Sumado a esto, faltan esfuerzos desde la sostenibilidad humana para impactar comunidades (equipos de desarrollo y usuarios) con infraestructura tecnológica limitada; algunos desafíos visibles son: (i) habilidades de TI, especialmente en desarrollo ágil, (ii) comercio transfronterizo (donde un desarrollador pueda ofrecer sus servicios desde cualquier lugar y comunicarse desde su zona horaria) y (iii) barreras culturales (Amrutha & Geetha, 2019; Ramautar et al., 2021). Desde la infraestructura tecnológica limitada son (iv) energía eléctrica débil, (v) acceso a internet y (vi) acceso a recursos tecnológicos (Ortiz et al., 2019).

En este artículo se propone un modelo de sostenibilidad humana en productos de software con infraestructura tecnológica limitada, que permita aplicar prácticas sostenibles en el ciclo de vida del desarrollo de software integrando el aspecto humano (equipos de desarrollo y usuarios). Se aplica un caso de estudio a la infraestructura tecnológica y a la comunidad de Providencia-Colombia, que presenta características vulnerables como vulnerabilidad económica, ambiental y de infraestructura. Las características mencionadas sirvieron para seleccionar la comunidad de Providencia como un espacio de validación del modelo propuesto que busca la inclusión de estas comunidades con nuevas formas de crear y compartir conocimiento, de forma co-activa, auto-empoderada y co-creativa. Adicional a eso, las personas de estas comunidades de bajos ingresos pueden tener habilidades para el empleo y/o comenzar a interesarse por la tecnología digital. Por lo que se espera que el modelo propuesto permita generar procesos de desarrollo de software sostenible, promueva el acceso a plataformas en su contexto, equidad social y laboral al incentivar habilidades de desarrollo de software y el uso responsable de productos de software sostenibles, impactando también la dimensión económica del software en Colombia.

La estructura de este artículo es la siguiente: similares; la segunda sección contiene el método de investigación que guía este trabajo; la tercera sección presenta el modelo propuesto; la cuarta sección aborda el caso de aplicación en la comunidad de Providencia-Colombia; en la quinta sección se generan los resultados. Finalmente, se incluye la discusión y las conclusiones.

2. Método

Para la investigación se utilizó el método de *Design science*, el cual estructura el proceso para solucionar un problema en un contexto social donde se generan artefactos que responden a las preguntas y a la solución de sistemas de información e ingeniería de software (Weiringa, 2014). Las fases de la metodología se presentan en la Tabla 1. Este método se trabajó con estrategias de investigación aplicada desde el Semillero de investigación en Ingeniería de Software de la Universidad EAFIT, combinando co-creación y planeación en un entorno colaborativo y un enfoque basado en proyectos, donde los participantes identifican problemáticas, plantean soluciones, desarrollan propuestas y validan resultados. El trabajo se complementa con charlas y salidas de campo.

A continuación, se presentan los instrumentos utilizados en el método de investigación:

User persona: Técnica de creación de personas o *user* persona, que se utiliza para comprender cuáles son sus motivaciones, necesidades, gustos, rutinas, situaciones socioeconómicas, conocimientos base en tech, ocupaciones, expectativas y su contexto familiar mediante una ficha, que se utiliza en el diseño de experiencia usuario para el desarrollo de software (Goodwin, 2011). En el presente proyecto una encuesta socioeconómica para conocer al grupo y diseñar el material no era posible dado que los usuarios eran menores de edad e imposibilitaba el tratamiento de los datos; el *User* Persona resultó útil porque permitió representar quiénes son nuestros usuarios y guiar las decisiones de diseño e implementación del modelo (Véase la Figura 1).

Fases	Actividades
Diseño	Diseñar un modelo de sostenibilidad humana en productos de software con infraestructura tecnológica limitada (participación de expertos, investigadores en formación y profesores)
Demostración y evaluación	Validar el modelo en un caso de estudio con la infraestructura tecnológica y la comunidad de Providencia donde se pueda capacitar en habilidades de desarrollo y aplicar el modelo propuesto.
Comunicación	Generar una estrategia de divulgación en la comunidad de Providencia (mediante medios de comunicación oficiales de la Universidad y de la Isla y mediante la interacción presencial).

Tabla 1. Fases del método

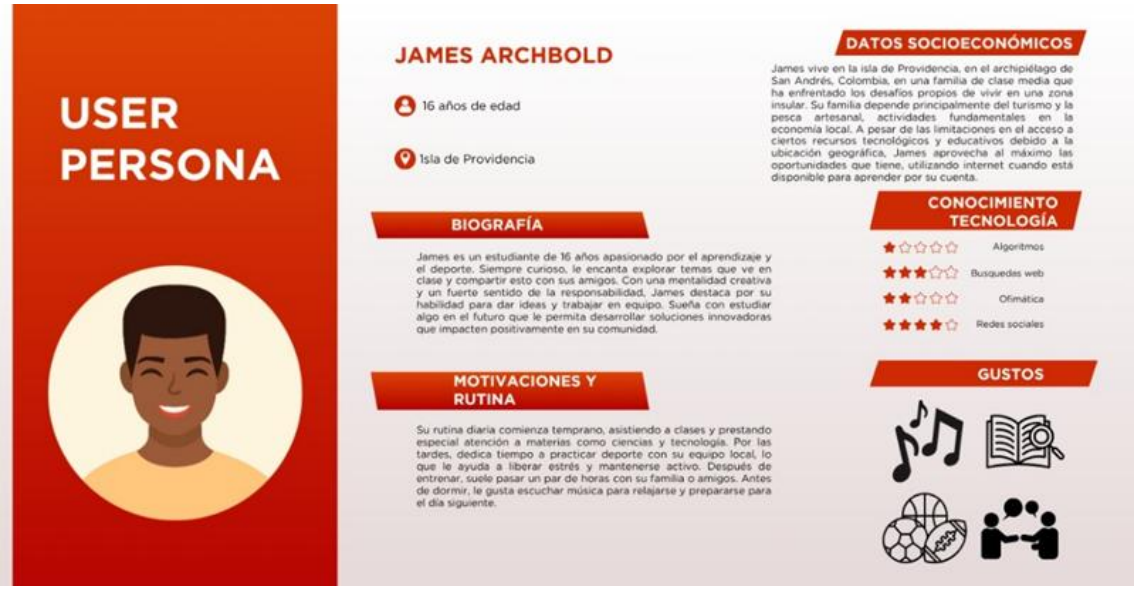


Figura 1. *User* persona de los jóvenes I.E. Junín

Diagnóstico de entrada: Instrumento estructurado para evaluar el nivel de conocimiento que tienen los participantes en relación con las tecnologías y sistemas. Se aplicó antes de la intervención para identificar brechas de aprendizaje y establecer una línea base que permitió diseñar estrategias didácticas adecuadas. En suma, las preguntas seleccionadas en el diagnóstico se relacionaron con los objetivos de la intervención, las competencias que se buscaban desarrollar y los conocimientos previos necesarios para el aprendizaje efectivo.

Diagnóstico de salida: Instrumento diseñado para evaluar el impacto de la intervención y recoger información sobre la percepción y el aprendizaje de los participantes. El diagnóstico de salida tuvo tres momentos: (i) un cuestionario teórico de conocimientos posteriores al taller, (ii) verificación de las actividades de desarrollo de software y habilidades blandas y (iii) una encuesta de satisfacción sobre la experiencia durante el taller. Este es un instrumento de valor ya que permite evaluar el aprendizaje alcanzado, valorar la efectividad del proceso de enseñanza y reflexionar sobre el desarrollo de competencias durante la intervención, permitiendo contrastar incluso con el diagnóstico de entrada.

3. Modelo de sostenibilidad humana para productos de software con infraestructura limitada con infraestructura tecnológica limitada

El modelo propuesto consta de cuatro fases que permiten introducir prácticas de sostenibilidad humana en productos de software con infraestructura tecnológica limitada, las cuales se representan en la Figura 2.

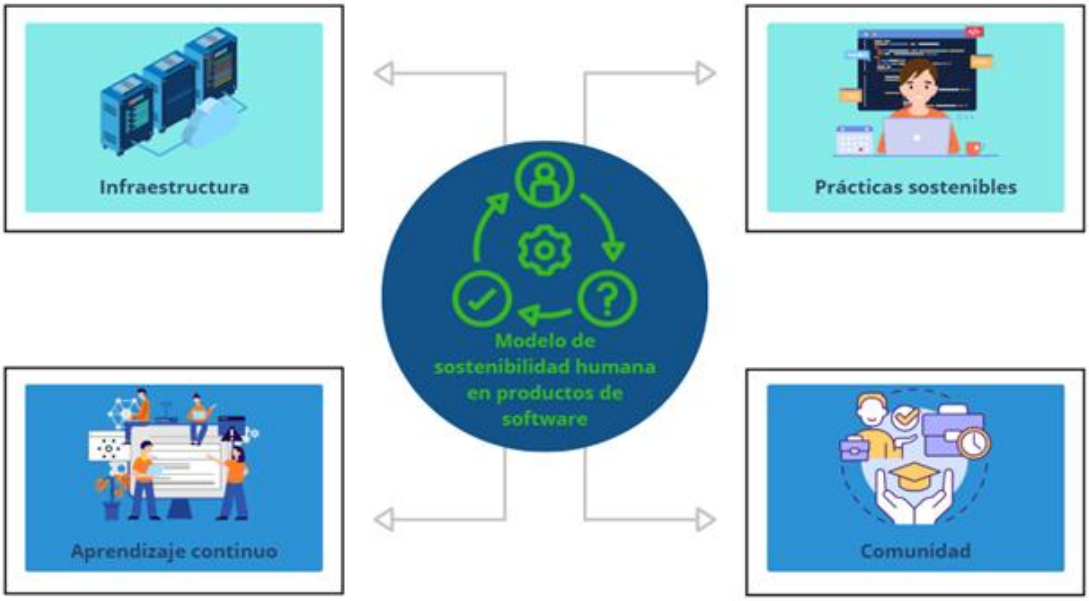


Figura 2. Fases del modelo propuesto

3.1 Infraestructura

La infraestructura tecnológica comprende el conjunto de recursos físicos, digitales y organizativos que permiten el acceso, uso y desarrollo de TI. En comunidades con infraestructura tecnológica limitada, estas condiciones se traducen en la falta de recursos en conectividad a internet, acceso a dispositivos tecnológicos, disponibilidad de software actualizado y capacidades técnicas para su mantenimiento y uso. Estas limitaciones pueden dificultar el desarrollo educativo, económico y social de las comunidades afectadas (Noreña et al., 2024c).

3.2 Prácticas sostenibles

Estrategias y acciones que buscan minimizar el impacto ambiental, económico y social en el uso de recursos. En el contexto tecnológico, las prácticas sostenibles promueven el reciclaje de dispositivos electrónicos, la reducción del consumo energético en infraestructuras digitales, la promoción de la economía circular en el desarrollo de hardware y software, y la adopción de tecnologías con menor huella ecológica, bienestar, equidad y equidad de las personas involucradas en el desarrollo de software. Para mayor detalle, se propusieron en un trabajo anterior las prácticas en sostenibilidad humana y sostenibilidad del software (Noreña et al., 2024c), las cuales se pueden observar en las Figuras 3 y 4.

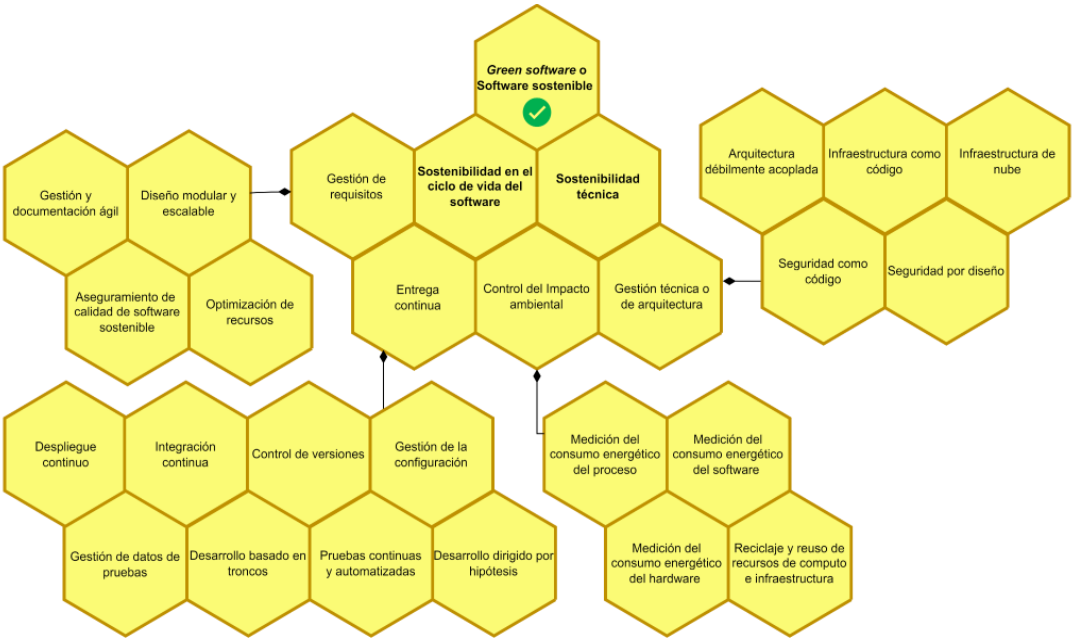


Figura 3. Prácticas de sostenibilidad humana en software (Noreña et al., 2024c)

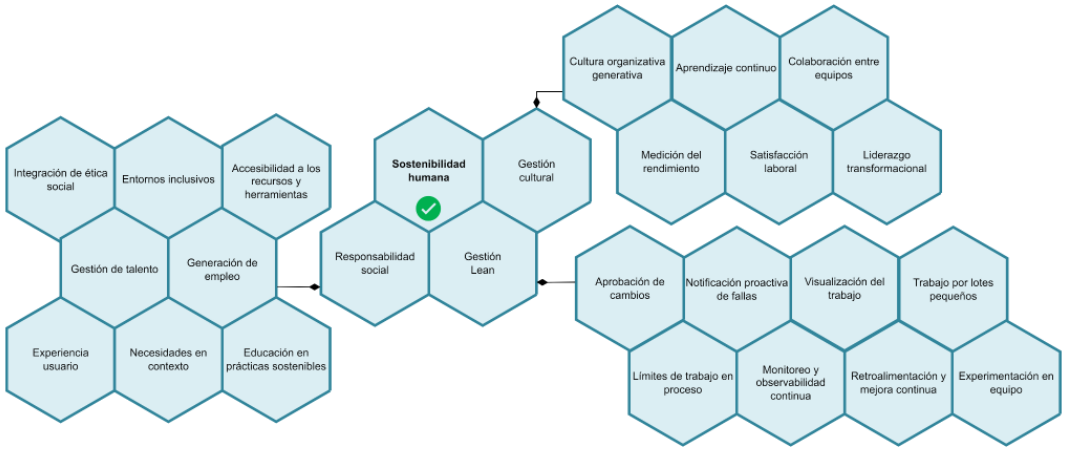


Figura 4. Prácticas de software sostenible (Noreña et al., 2024c)

3.3 Aprendizaje continuo

Actualización y adquisición de conocimientos sobre nuevas tecnologías y su relación con la sostenibilidad (Noreña et al., 2025a). En comunidades vulnerables, esto implica capacitaciones accesibles y permanentes que permitan el desarrollo de habilidades digitales, el aprovechamiento de herramientas tecnológicas y la implementación de soluciones innovadoras que promuevan la sostenibilidad del software y la sostenibilidad humana (Noreña et al., 2024d; Noreña et al., 2025b)

3.4 Comunidad

La integración de tecnologías y prácticas sostenibles en comunidades vulnerables tiene como objetivo mejorar su calidad de vida, promoviendo el acceso equitativo a los recursos tecnológicos, el acceso a internet, la educación, la salud, el empleo y otros servicios esenciales. Esto se logra mediante estrategias como la instalación de centros de conectividad comunitaria con energía renovable, el uso de tecnologías para facilitar el acceso a la educación en línea, el impulso de

emprendimientos digitales inclusivos, y la integración de tecnologías y prácticas sostenibles en comunidades vulnerables, constituye una estrategia clave para mejorar su calidad de vida y reducir las desigualdades sociales. Esta integración busca garantizar el acceso equitativo a recursos tecnológicos, servicios de conectividad, educación, salud, empleo y otros servicios esenciales. Entre las principales aplicaciones se destacan la instalación de centros comunitarios dotados de energía renovable, la implementación de plataformas digitales para educación a distancia, el fomento de emprendimientos digitales inclusivos y la promoción de programas de alfabetización digital con enfoque sostenible. Estas iniciativas no solo contribuyen a disminuir la brecha digital, sino que también favorecen el empoderamiento social y económico de las comunidades, fortaleciendo su resiliencia y capacidad de autogestión (UNESCO, 2023).

4. Caso de aplicación: Providencia – Colombia

4.1 Infraestructura

Los habitantes de la isla de Providencia reciben energía en sus hogares mediante tres posibles alternativas: servicio del sistema de energía, paneles solares y motores de combustión Diesel (Howard, 2024). La Isla cuenta con internet satelital, a pesar de que después del paso del huracán Iota en 2020 se mejoraron los servicios de infraestructura, no toda la Isla cuenta con acceso a internet y sólo se encuentran dos puntos gratuitos de conexión (UNGRD, 2020). En particular, la Institución Educativa (IE) Junín que presta los servicios educativos de enseñanza primaria y secundaria para aproximadamente 200 niños y jóvenes de la Isla, cuenta con algunos dispositivos electrónicos que conforman su infraestructura tecnológica, la cual se especifica en la Tabla 2.

Computadores (28)	
Disco duro	500Gb
Ram	4Gb
Procesador Intel celeron	1.80 Ghz
Sistema operativo	Windows 64-bits
Marca	Compumax
1 tablero interactivo (televisor)	65 pulgadas
Punto de acceso	Sin conexión a internet

Tabla 2. Especificación de tecnología institución

4.2 Prácticas sostenibles

Para una comunidad como la Isla de Providencia, se implementaron prácticas de sostenibilidad humana desde un enfoque de responsabilidad social y gestión en la sostenibilidad del software. Algunas de las prácticas aplicadas fueron (Noreña et al., 2024c):

- Entornos inclusivos: Se aseguraron condiciones que permitieran la participación de los jóvenes de la comunidad desde la IE Junín y miembros de la comunidad. Algunas de las actividades colaborativas buscaban transmitir saberes, tradiciones gastronómicas respetando su identidad cultural y promoviendo la equidad. Educación en prácticas sostenibles: Se impartieron conocimientos y herramientas sobre el uso eficiente de tecnologías de desarrollo de software en un modo desconectado, el aprovechamiento de sus recursos tecnológicos y fomentando una cultura de sostenibilidad dentro de la comunidad que permita aplicar y seguir enseñando estas prácticas. También, a través de los conceptos de software se enseñaron temáticas en contexto como fueron el ahorro y la gestión financiera personal y familiar.

- Accesibilidad a los recursos y herramientas: Se brindó acceso a tecnologías y metodologías clave para el desarrollo local. Un ejemplo fue la capacitación en herramientas de programación como Python, utilizando la *Suite Anaconda*, el cuaderno interactivo *Jupyter Notebook* y *Visual Studio Code*, lo que permitió a los participantes adquirir habilidades en desarrollo de software.

- Necesidades en contexto y generación de empleo: Se diseñaron estrategias alineadas con las condiciones específicas de la isla, promoviendo la adopción de tecnologías y conocimientos que permitieron nuevas habilidades para el trabajo, que incentivaron la autonomía económica de la comunidad.

- Colaboración entre equipos: Se fomenta el trabajo en equipo mediante el reto de desarrollar un módulo en la aplicación. Esta sinergia permite una mejor toma de decisiones y la implementación de soluciones adaptadas a la realidad local y a los participantes.

4.3 Aprendizaje continuo

El objetivo de aplicar el proyecto se centró en enseñar a los estudiantes diversos conceptos sobre los fundamentos de programación, los cuales son clave para el desarrollo de software. Posteriormente, con el fin de aplicar lo aprendido en el curso, se creó una plantilla base de una página web orientada al ahorro individual y familiar. Los estudiantes debían completar ciertas instrucciones en el código para que la plantilla fuera completamente funcional. Sin embargo, esta actividad, además de buscar consolidar conocimientos, esperaba despertar el interés por seguir aprendiendo. Por esta razón, los participantes del proyecto obtuvieron acceso a material adicional sobre emprendimiento, otras áreas de la programación, como ciencia y arquitectura de datos, y el desarrollo de habilidades blandas mediante talleres de escritura y preparación para el mundo laboral construido por diferentes áreas de la Universidad EAFIT: Escuela de Ciencias Aplicadas e Ingenierías, Escuela de Artes y Humanidades, On.going (Centro de emprendimiento) y Nodo EAFIT (Cursos virtuales) y las empresas Magneto Global y Smart Green IT.

Además, se designaron mentores en la isla con el propósito de dar continuidad al desafío de transmitir los conocimientos adquiridos durante el taller, inspirar a las próximas generaciones y asegurar que lo impartido genere un impacto a largo plazo. Por tanto, se espera que esta iniciativa

se comparta con la comunidad y sirva de motor para que más personas exploren el mundo de la tecnología y la utilicen como herramienta para construir sus proyectos de vida. Para explorar el contenido abordado en esta fase de aprendizaje continuo, puede ingresar al siguiente enlace: Workshop

4.4Comunidad

La Isla de Providencia está ubicada en el mar Caribe, en el departamento colombiano del Archipiélago de San Andrés, Providencia y Santa Catalina (véase la Figura 5). La isla cuenta con una población aproximada de 5000 habitantes en total (Asociación Colombiana de Ciudades Capitales, s.f.). Es una comunidad vulnerable que en su mayoría vive de actividades turísticas y pesca. La comunidad raizal (personas nativas de la Isla) cuenta con gran ventaja en temas de barreras culturales e idiomáticas, además de hablar su idioma creole, hablan español e inglés. En términos culturales, esta comunidad se caracteriza por sus costumbres y/o tradiciones que engloban su gastronomía isleña, su arquitectura raizal con la que se construyen los hogares, su música, entre otros.

A partir de la investigación y análisis de usuarios reales se muestra en la Figura 1 el instrumento *user* persona para caracterizar a los jóvenes impactados en el proyecto y su contexto.

Este enfoque permitió analizar sus necesidades, motivaciones y desafíos en el aprendizaje de programación, para diseñar estrategias con un aprendizaje cercano a la realidad de su comunidad. En total, participaron 17 estudiantes de undécimo grado de la IE Junín, ubicada en la Isla de Providencia-Colombia (este representa el último grado de la secundaria). Sus edades oscilaron entre 16 y 18 años, reflejando un grupo homogéneo en términos de nivel académico, pero diverso en intereses y habilidades previas en tecnología. Este instrumento también permitió identificar aspectos clave como el nivel de familiaridad de los estudiantes con herramientas digitales, su acceso a dispositivos y conectividad, así como sus expectativas sobre el aprendizaje de fundamentos de programación. Además, proporcionó información valiosa sobre sus motivaciones personales, aspiraciones profesionales y los desafíos que enfrentan en su contexto educativo y social.

Figura 5. Isla de Providencia-Colombia (Google Maps)

5. Resultados

De manera previa se compartió material de estudio para familiarizar a los estudiantes con las temáticas objetivo, luego se realizó una salida de campo a Providencia-Colombia donde se llevó a cabo el primer Workshop de Desarrollo de software impartido por el Semillero de Ingeniería de software EAFIT, en temas de proyectos de software, prácticas sostenibles, producto mínimo viable, escritura, desarrollo de software en Python (Variables, operadores, condicionales, ciclos y listas) y habilidades blandas como trabajo en equipo, comunicación y liderazgo. Se delegan mentores con el objetivo de compartir con otros grupos y personas el conocimiento recibido.

Los Semilleristas impartieron sus conocimientos durante cinco días a 17 estudiantes de undécimo grado de la Institución Educativa Junín, como se observa en la Figura 6.

Figura 6. Semilleristas y estudiantes en el Workshop

5.1Diagnóstico de entrada

El diagnóstico de entrada permitió establecer un punto de referencia sobre los conocimientos previos de los participantes en el ámbito tecnológico. Se aplicó una encuesta inicial para identificar su familiaridad con herramientas digitales, su nivel de experiencia en programación y sus necesidades específicas de aprendizaje. Los resultados mostraron que la mayoría de los participantes tenían conocimientos en herramientas ofimáticas, redes sociales e internet, lo que indica una familiaridad básica con el uso de tecnologías digitales. Sin embargo, se evidenció una ausencia significativa de experiencia en desarrollo de software. Para complementar este análisis, se empleó un instrumento del *user* persona (véase la Figura 1).

A partir de la información recopilada, se construyeron perfiles representativos de los participantes, permitiendo clasificar sus niveles de conocimiento y sus expectativas con el workshop. Esta estrategia ayudó a planificar y personalizar la fase de aprendizaje continuo, asegurando que el contenido y las tecnologías fueran accesibles y relevantes para su contexto con un acompañamiento didáctico y actividades prácticas que facilitarán la comprensión de conceptos clave.

5.2 Diagnóstico de salida

Con el fin de contrastar los resultados del workshop con el diagnóstico de entrada, se realizó un diagnóstico de salida basado en tres pilares:

- Cuestionario teórico: Como parte del proceso metodológico, se aplicó un cuestionario teórico para evaluar la adquisición de conocimientos fundamentales en programación y pensamiento computacional. Esta evaluación incluyó preguntas sobre conceptos clave, tales como: fundamentos de programación y su aplicación en el desarrollo de software, pilares del pensamiento computacional, incluyendo descomposición de problemas, reconocimiento de patrones, abstracción y diseño de algoritmos; conceptos básicos de programación, como el uso de variables y operadores lógicos y aritméticos y estructuras de control de flujo, abarcando ciclos (*for* y *while*) y condicionales (*if-else*). permitieran nuevas habilidades para el trabajo, que incentivaran la autonomía económica de la comunidad.

  Los resultados obtenidos en esta evaluación reflejaron que, en el proceso de aprendizaje en el workshop, los participantes lograron responder adecuadamente las preguntas planteadas, evidenciando una apropiación efectiva del conocimiento. Este desempeño sugiere que los objetivos de enseñanza se cumplieron satisfactoriamente, permitiendo a los asistentes comprender los principios fundamentales de la programación y aplicarlos en la resolución de problemas computacionales.

- Verificación de actividades en el desarrollo de software y actividades de habilidades blandas: A lo largo del taller, se realizó un seguimiento constante del progreso de los participantes tanto en el desarrollo de software como en el fortalecimiento de sus habilidades blandas. Cada día, lograron superar los retos planteados y avanzar en la creación de la aplicación propuesta, lo que les permitió culminar exitosamente el taller. Como parte de la metodología, se incluyó una actividad de co-creación y colaboración en la que los participantes trabajaron en equipo para diseñar soluciones alineadas con las necesidades del contexto. Esta dinámica fue guiada desde el curso de escritura impartido por profesores de la Escuela de Artes y Humanidades de la Universidad EAFIT, promoviendo no solo el pensamiento crítico y la creatividad, sino también la integración de la tecnología con la narrativa y la identidad cultural de la comunidad. Uno de los productos resultantes de esta actividad fue un texto elaborado colectivamente por los equipos, en el que expresaron su visión sobre Providencia y su proceso de resiliencia tras el paso del huracán Iota: "*Bienvenidos a Providencia, una isla pequeña con mucha vegetación y naturaleza hermosa, cultura y tradiciones que tiene un tesoro: el mar de siete colores. La comunidad es muy unida, eso se pudo evidenciar cuando llegó el huracán Iota. La hermosa isla no se veía igual, pero algunos meses después, los isleños* unieron esfuerzos para plantar vegetación y manglares, y en algunos años, la isla volvió a resplandecer."

  Este ejercicio no sólo permitió integrar el aprendizaje técnico con la reflexión sobre el entorno y la identidad de la comunidad, sino que también fortaleció la capacidad de los participantes para trabajar en equipo, comunicar ideas y desarrollar soluciones de manera colaborativa.

- Cuestionario de satisfacción: Con el fin de evaluar el workshop desde la perspectiva de los estudiantes, se realizó un cuestionario que consta de cinco preguntas: en la primera pregunta, los participantes calificaron la experiencia de aprendizaje en una escala del 1 al 5, donde 1 representa la menor satisfacción y 5 la mayor. De los 17 encuestados, la

mayoría (9 personas, 52.9%) otorgó la calificación más alta (5), mientras que 6 personas (35.3%) calificaron con un 4 como se observa en la Figura 7. Estos resultados reflejan una percepción mayoritariamente positiva del workshop, con un 88.2% de las respuestas ubicándose en los niveles más altos de satisfacción (4 y 5). La ausencia de calificaciones en el nivel más bajo (1) sugiere que ningún participante percibió la experiencia como totalmente insatisfactoria. Importante resaltar que, aunque el cuestionario fue breve, su diseño se centró en captar los aspectos esenciales del objetivo planteado: conocer el grado de satisfacción de los participantes respecto a la metodología, los contenidos y la experiencia global de la intervención. Además, la simplicidad del formato favoreció respuestas honestas y rápidas, reduciendo la fatiga del encuestado y aumentando la confiabilidad de los datos obtenidos.

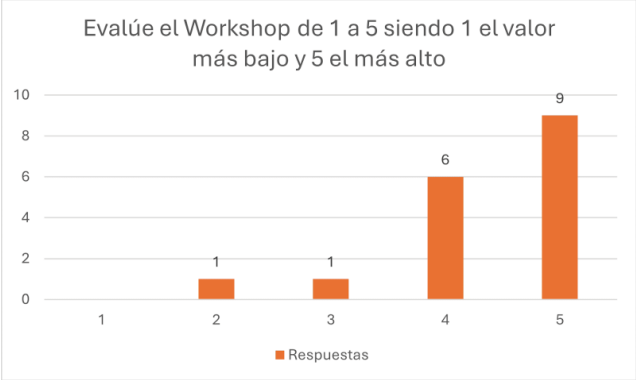


Figura 7. Resultados de la pregunta 1

En la segunda pregunta, se indaga a los participantes si consideran haber adquirido más conocimientos tras la actividad. Los resultados de la Figura 8 reflejan una percepción mayoritariamente positiva: 9 personas (52.9%) manifestaron estar muy de acuerdo, 6 personas (35.3%) estuvieron de acuerdo y 1 persona (5.9%) se mantuvo neutral. En total, el 88.2% reconoció haber mejorado sus habilidades en programación tras el workshop, lo que sugiere que las metodologías y herramientas empleadas fueron efectivas para el aprendizaje.

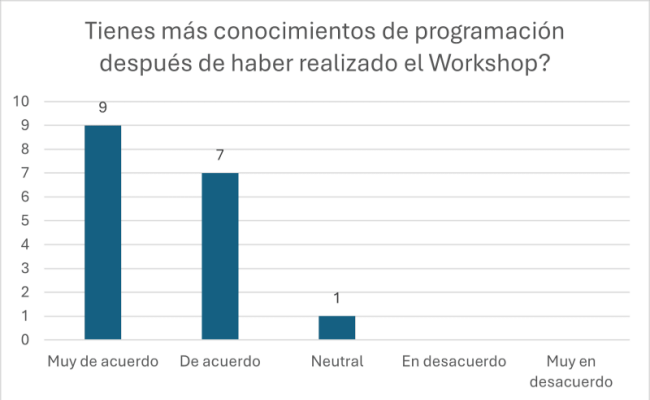


Figura 8. Resultados de la pregunta 2

En la pregunta tres, se consultó a los participantes si consideraban haber adquirido conocimientos en el lenguaje de programación Python tras el workshop. Los resultados fueron los siguientes de acuerdo con la Figura 9: 5 personas (29.4%) indicaron estar muy de acuerdo en haber aprendido sobre Python, 9 personas (52.9%) expresaron estar de acuerdo con esta afirmación y 3 personas (17.6%) se posicionaron de manera neutral. En total, el 82.3% de los encuestados reconoció haber adquirido conocimientos en Python, evidenciando un impacto positivo del workshop en la formación de los participantes. La presencia de respuestas neutrales sugiere la posibilidad de distintos niveles de familiarización previa con el lenguaje o diferencias en la apropiación de los contenidos abordados.

En la pregunta cuatro, se consultó si los participantes consideraban haber desarrollado habilidades blandas como trabajo en equipo, comunicación y liderazgo durante el workshop. Los resultados mostraron una percepción positiva basada en la Figura 10: 6 personas (35.3%) indicaron estar muy de acuerdo en haber fortalecido estas habilidades, 7 personas (41.2%) señalaron estar de acuerdo y 4 personas (23.5%) se posicionaron de manera neutral. En total, el 76.5% de los participantes reconoció haber mejorado en estas habilidades, lo que indica que la dinámica del workshop fomentó la interacción, la colaboración y la toma de decisiones en equipo. La presencia de respuestas neutrales puede indicar que algunos participantes percibieron un impacto menor en su desarrollo.

En la última pregunta cualitativa de observaciones se obtuvieron comentarios positivos como: el workshop es recomendado porque aprendes a crear un MVP (producto mínimo viable) o versión más simple que permite validar una idea, programación, Python y habilidades blandas y algunos indicaron que les gustaron las clases y la experiencia de aprendizaje en el taller. Los documentos de la salida de campo, instrumentos y plantillas usadas se pueden encontrar en el siguiente enlace: Salida de campo

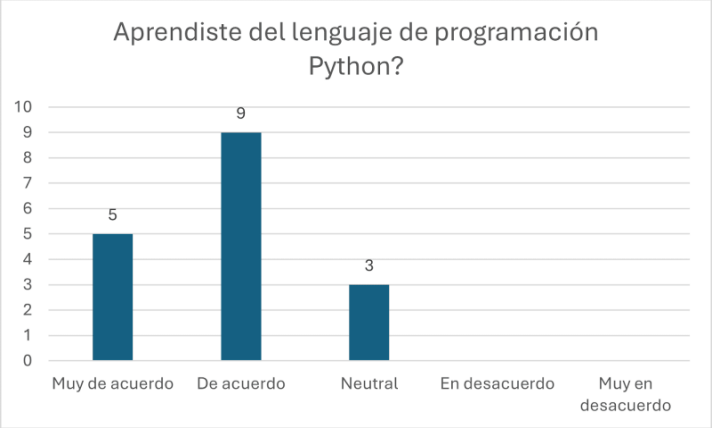


Figura 9. Resultados de la pregunta 3

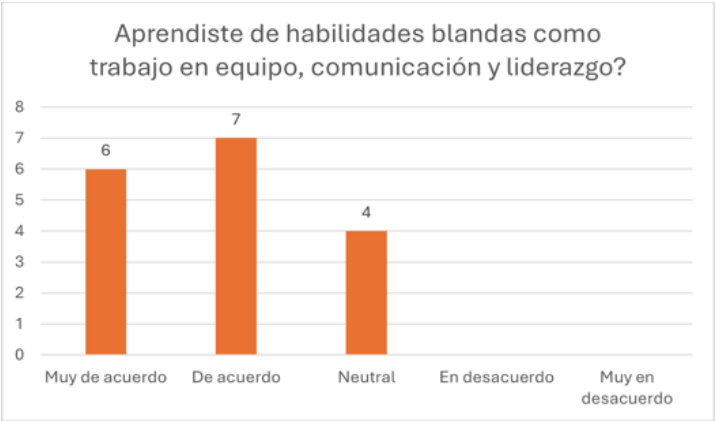


Figura 10. Resultados de la pregunta 4

5.3 Sesgos y limitaciones de la investigación

La encuesta fue aplicada a 17 participantes, lo que representa un grupo reducido. Si bien los resultados reflejan una tendencia positiva, una muestra más amplia permitiría obtener conclusiones más generalizables. También, aunque la primera parte se inició tiempo antes de forma virtual, la duración del workshop pudo haber limitado la profundidad con la que se abordan ciertos temas, especialmente aquellos relacionados con estructuras más avanzadas de programación. Un periodo de formación más extenso podría haber permitido un aprendizaje más consolidado. Por otro lado, algunos resultados, como los obtenidos en las encuestas de satisfacción, dependen de la percepción subjetiva de los participantes y no necesariamente reflejan una medición objetiva del aprendizaje. También, aunque se proporcionaron herramientas digitales para desarrollar software, no todos los participantes contaban con acceso continuo a dispositivos adecuados o conexión estable a internet fuera del workshop, lo que podría afectar la continuidad del aprendizaje después de la actividad. Estas limitaciones ayudan a contextualizar los resultados y pueden servir como base para mejorar futuras implementaciones del modelo propuesto.

De manera general, al aplicar la metodología *Design science,* es fundamental reconocer y declarar posibles sesgos que puedan afectar la validez del estudio. Entre los más relevantes identificados por los participantes son: (i) el sesgo de confirmación, al tener trabajos que respalden el presente trabajo y al posiblemente haber obviado trabajos que pudieron haber aportado al presente estudio, el cual fue mitigado socializando el modelo con actores externos para recibir retroalimentación; (ii) el sesgo en el diseño el cual puede presentar preferencias personales, este fue mitigado al evaluar distintas alternativas antes de decidir la definición y aplicación del modelo; (iii) el sesgo de usuario representativo, para ello se garantiza una muestra diversa en la evaluación, y (iv) el sesgo de generalización, para ello se hizo necesario documentar el proceso evidenciando su replicabilidad y su relevancia en el contexto real.

6. Discusión y conclusiones

En conclusión, al abordar la sostenibilidad del software parece obligatoria la perspectiva técnica y que esta sea el punto de partida, sin embargo, se hace necesario evidenciar la integración de la dimensión humana, especialmente en contextos con infraestructura tecnológica limitada, este concepto puede referirse a las limitaciones de una plataforma o a las condiciones de infraestructura

tecnológica que tiene una comunidad. El modelo propuesto demuestra que es posible desarrollar productos de software sostenibles que no solo minimicen el impacto ambiental, sino que también promuevan la equidad digital y la inclusión social.

En este contexto, la aplicación del modelo propuesto en la isla de Providencia-Colombia resaltó la importancia de identificar las necesidades específicas de comunidades vulnerables, mostró que diseñar en contexto y garantizar el acceso a la tecnología puede hacer de nuestra sociedad más inclusiva, también evidenció que la formación en habilidades digitales permite democratizar las tecnologías y ofrecer caminos hacia otras oportunidades laborales para este tipo de población. Así, este tipo de enfoque contribuye al tejido social, al desarrollo sostenible, impulsando el crecimiento económico y social a través de la tecnología.

El presente trabajo representa una oportunidad y un punto de partida para que sea implementado en otros contextos, otras comunidades con características socioeconómicas y tecnológicas diversas para evaluar su adaptabilidad y efectividad. Se evidenció que una de las principales barreras para la digitalización en muchas comunidades es la infraestructura con la que cuentan. Ya que muchas de estas comunidades suelen estar situadas en zonas alejadas de las principales ciudades, donde sí se cuenta con acceso, uso de hardware, conectividad a internet y recursos digitales de manera adecuada. Es por esto que la experiencia aquí presentada abre la posibilidad de mejorar el modelo para promover la inclusión digital en otros ámbitos, por ejemplo, tenemos planes de usar el modelo en contextos de economía plateada, dado que la autonomía y la participación significativa de las personas mayores en entornos donde la tecnología cumple un papel cada vez más central. La experiencia previa en contextos educativos con infraestructura limitada demuestra que el modelo puede implementarse con éxito en escenarios de diversa complejidad tecnológica, lo que también favorece su aplicación en comunidades con distintos niveles de alfabetización digital.

Finalmente, las características de la infraestructura en ciertas comunidades como la presentada en este artículo tienen limitaciones y restricciones para el crecimiento social y económico. Superar ese desafío es posible conociendo y entendiendo cómo es su infraestructura e identificando mecanismos para optimizar dicho recurso en función de un mejor aprovechamiento, el impacto positivo y la generación de valor, en pocas palabras, aplicando un modelo que contemple prácticas sostenibles en el ciclo de vida del desarrollo de software y que integre el aspecto humano. Es así como esta propuesta tiene el potencial de mejorar significativamente la calidad de vida y resulta fundamental para reducir la brecha digital.

Agradecimientos

Este artículo surge como resultado del proyecto “Modelo de sostenibilidad humana en productos de desarrollo de software con infraestructura tecnológica limitada” en el programa de Mujeres en la ciencia 2023 patrocinado por Minciencias, Icetex y L’oreal-Unesco y ejecución en Universidad EAFIT. Agradecimientos especiales a los estudiantes del Semillero de Ingeniería de software de la Universidad EAFIT: Isis Amaya, Diego Galvis, Jonathan Betancur, Alejandro Restrepo, Laura Restrepo, Miguel Sosa, Laura Castrillón, Miguel Mercado, Sofía Flores y Paula Llanos. También sinceros agradecimientos a Nodo y On.going EAFIT, a los profesores Liliana González, Carlos Álvarez, Carolina Ospina, Paula Colorado y Giovanni Gómez; a Felipe Soto (independiente), a Gerardo Archbold y Julieta Hooker (IE Junín), a los estudiantes del grado 11 en el año 2024 (IE Junín), a la comunidad de Providencia, Canal TeleIslas y a las empresas Source Meridian, Smart Green IT y Magneto por sus contribuciones al proyecto.

Referencias

[1] Alavi, H. S., Verma, H., Mlynar, J., & Lalanne, D. (2018, April). The hide and seek of workspace: Towards human-centric sustainable architecture. In *Proceedings of the 2018 CHI conference on human factors in computing systems* (pp. 1-12).

[2] Amrutha, V. N., & Geetha, S. N. (2020). A systematic review on green human resource management: Implications for social sustainability. *Journal of Cleaner production*, *247*, 119131.

[3] Asociación Colombiana de Ciudades Capitales. (s.f.). "Providencia y Santa Catalina," Available: https://www.asocapitales.co/providencia-y-santa-catalina/

[4] Breque, M., De Nul, L., & Petridis, A. (2021). *Industry 5.0: towards a sustainable, human-centric and resilient European industry* (No. KI-BD-20-021-EN-N). Directorate General for Research and Innovation (DG RTD) of the European Commission.

[5] Calero, C., Moraga, M. Á., & Piattini, M. (2021). Introduction to software sustainability. In *Software Sustainability* (pp. 1-15). Cham: Springer International Publishing.

[6] Colombia Crea (s.f.). "Código comuna 13" Available: https://colombiacrea.org/proyecto/codigo-comuna-13-escuela-de-formacion-de-programacion-de-software/

[7] EL TIEMPO (2024). "Alcaldía de Medellín lanzó 'Parceros', el programa con el que busca evitar el reclutamiento y la explotación sexual," Available: https://www.eltiempo.com/colombia/medellin/alcaldia-de-medellin-lanzo-parceros-el-programa-con-el-que-busca-evitar-el-reclutamiento-y-la-explotacion-sexual-3345463

[8] Goodwin, K. (2011). *Designing for the digital age: How to create human-centered products and services*. John Wiley & Sons.

[9] Grundy, J., Khalajzadeh, H., McIntosh, J., Kanij, T., & Mueller, I. (2020). Humanise: Approaches to achieve more human-centric software engineering. In *International Conference on Evaluation of Novel Approaches to Software Engineering* (pp. 444-468). Cham: Springer International Publishing.

[10] Heldal, R., Nguyen, N. T., Moreira, A., Lago, P., Duboc, L., Betz, S., ... & Venters, C. C. (2024). Sustainability competencies and skills in software engineering: An industry perspective. *Journal of Systems and Software*, *211*, 111978.

[11] Howard, E. (2024). "Santa Catalina Verde tendrá el 100% de energía renovable," El Isleño, 29-feb-2024. Available: https://www.elisleño.com/index.php?option=com_content&view=article&id=28737%3A2024-02-29-08-58-35&catid=41%3Aambiental&Itemid=83

[12] Internet Society Capítulo Panamá (s.f.). Available: https://www.isoc.org.pa/

[13] Noreña, P. A., Suescún, E., & Mejia, J. (2024a). Desarrollo dirigido por hipótesis en productos de software sostenible: Un mapeo sistemático. JAIIO, Jornadas Argentinas de Informática, 10(2), 46-59.

[14] Noreña, P., Suescún, E., Caro, M., Toro, M., & Fernández, P. (2024b). Integration of human sustainability practices to a LMS using cognitive functions for early dropout risk alerts at Universidad EAFIT: a representation in PCS. In International Conference in Information Technology and Education (pp. 248-257). Cham: Springer Nature Switzerland.

[15] Noreña, P. A., Suescún, E., & González, L. (2024c). Caracterización de prácticas sostenibles en el desarrollo de software y la sostenibilidad humana usando Essence. *CISTI*, 1-6.

[16] Noreña, P., Suescún, E., Calle, J., Orozco, C., Álvarez, C. (2024d). "Representación de dominios de software científico: un aprendizaje continuo a partir de estructuras matemáticas de los esquemas preconceptuales," Congr. Col. Computación.

[17] Noreña, P., Durango, C., Suescún, E., & Pardo, C. (2025a). Improving understanding of the DevOps framework using Essence a visual representation. Periodicals of Engineering and Natural Sciences, 13(2), 305-326.

[18] Noreña, P., Suescún, E., Zapata, C. M., Scaunasu, B., Elliot, G., Jacobsen, H. A., & Mosquera, D. (2025b). Continuous learning of event-based systems by using pre-conceptual schemas. Journal of Systems and Software, 230, 112529.

[19] Ojameruaye B. y Bahsoon R. (2021). Sustainability ArchDebts: AnEconomics-Driven Approach for Evaluating Sustainable Requirements", 369-398. Software Sustainability. Springer.

[20] Ortiz, M. L., Ortega, I. D. M., Martinell, A. R., & Alvarado, M. A. C. (2019). La integración de Colombia a la sociedad de la información. *RevIISE-Revista de Ciencias Sociales y Humanas*, *14*(14), 73-86.

[21] Peláez, D. A., y Corrales, C. A. (2018). Identificación y caracterización de buenas prácticas orientadas al desarrollo de software sostenible desde el punto de vista económico, Trabajo de grado, Tecnológico de Antioquia.

[22] Ramautar, V., Overbeek, S., & España, S. (2021). Human sustainability in software development. In *Software sustainability* (pp. 329-348). Cham: Springer International Publishing.

[23] Restrepo, L. (2025). *A conceptual framework for the architectural design of sustainable cyber-physical systems (CPS)*. PhD Thesis. Universidad EAFIT.

[24] Restrepo, L., Aguilar, J., Toro, M., & Suescun, E. (2021). A sustainable-development approach for self-adaptive cyber–physical system's life cycle: A systematic mapping study. *Journal of Systems and Software*, *180*, 111010.

[25] Restrepo, L., Bernal, P., Suescun, E., Toro, M., Aguilar, J. & Pardo, C. (2023). Toward a conceptual framework for designing sustainable cyber-physical system architectures: A systematic mapping study. *Heritage and Sustainable Development*, *5*(2), 253.

[26] Restrepo, L., Pardo, C., Aguilar, J., Toro, M., & Suescún, E. (2024). SinSO: An ontology of sustainability in software. *Applied Ontology*, *19*(3), 231-263.

[27] Semerikov, S., Striuk, A., Striuk, L., Striuk, M., & Shalatska, H. (2020). Sustainability in Software Engineering Education: a case of general professional competencies. In *E3S Web of Conferences* (Vol. 166, p. 10036). EDP Sciences.

[28] Torres, S. Á., Barona, C., & García, O. (2010). Infraestructura tecnológica y apropiación de las TIC en la Universidad Autónoma del Estado de Morelos: Estudio de caso. *Perfiles educativos*, *32*(127), 105-127.

[29] UNESCO. (2023). "Technology and sustainable development: Bridging digital divides in vulnerable communities. United Nations Educational, Scientific and Cultural Organization".

[30] UNGRD. (2020). "Avanzan las acciones en materia de telecomunicaciones y conectividad en San Andrés, Providencia y Santa Catalina," 2020. Available: http://portal.gestiondelriesgo.gov.co/Paginas/Noticias/2020/Avanzan-las-acciones-en-materia-de-telecomunicaciones-y-conectividad-en-San-Andres-Providencia-y-Santa-Catalina.aspx

[31] Were, P., Madeley, J., & Munsell, M. (2020). Fair Trade Software: empowering people, enabling economies. *Journal of Fair Trade*, *2*(1), 4-12.

[32] White & Case (2022). "Africa's digital infrastructure transformation," Available: https://www.whitecase.com/insight-our-thinking/africas-digital-infrastructure-transformation

[33] Wieringa, R. (2014). *Design science methodology for information systems and software engineering*. Springer-Verlag Berlin Heidelberg.

NOTAS BIOGRÁFICAS

Paola Andrea Noreña Cardona es profesional, M.Sc. y Ph.D. en Ingeniería de Sistemas de la Universidad Nacional de Colombia. Actualmente es profesora e investigadora de la Universidad EAFIT. Sus áreas de interés son Ingeniería de Software, Diseño y Sostenibilidad.

Elizabeth Suescún Monsalve es Ingeniera Informática por el Politécnico Colombiano JIC, Magíster en Ciencias de la Computación y Doctor en Ciencias de la Computación por la Universidade Católica do Rio de Janeiro Brasil. Sus áreas de interés son la Ingeniería de Software, IHC, UX-UI, Transparencia de Software.

Santiago Manco Maya es estudiante del Pregrado en Ingeniería de Sistemas de la Universidad EAFIT. Actualmente hace parte del Semillero de Ingeniería de Software y es el Presidente del Capítulo SPIE de la Universidad EAFIT. Sus áreas de interés son la Ingeniería de Software, la Inteligencia Artificial y la Arquitectura de Software.

Santiago Idárraga Ceballos es estudiante del pregrado en Ingeniería de Sistemas de la Universidad EAFIT. Actualmente es miembro activo del Semillero de Ingeniería de Software, Representante Estudiantil ante el comité de pregrado en Ingeniería de Sistemas y es el Director Logístico del comité de Innovación y Emprendimiento de la Organización Estudiantil (VISIÓN OE). Sus áreas de interés son la Ingeniería de Software y los Sistemas de Gestión de Datos.

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

Recibido 02 Nov 2025 *ReCIBE, Año 15 No. 1, mayo 2026*

Aceptado 22 Abr 2026

Prácticas de Ciberseguridad en la Ingeniería de Software: Una Revisión de la Literatura

Cybersecurity Practices in Software Engineering: A Literature Review

Eduardo Antonio Castillo Garrido 1
e.antonio.cg@outlook.com
Juan Carlos Pérez-Arriaga 1
juaperez@uv.mx
Héctor Xavier Limón-Riaño 1
hlimon@uv.mx
Saúl Domínguez-Isidro 1
sauldominguez@uv.mx

[1] *Facultad de Estadística e Informática,*
Universidad Veracruzana, Xalapa, Veracruz, México.

Resumen

El desarrollo de productos de software que consideren aspectos de ciberseguridad en el diseño sigue representando un reto considerablemente alto para los equipos encargados de su construcción. En la Ingeniería de Software se han implementado diversas prácticas en distintas fases del desarrollo de software que tienen como objetivo construir productos cada vez más seguros, con la finalidad de satisfacer las demandas de la industria. A través de una revisión sistemática de 77 artículos, hemos identificado 30 prácticas relevantes de ciberseguridad, incluyendo políticas de seguridad, modelado de amenazas, análisis estático y pruebas de penetración. También se destacan artefactos clave como diagramas de clases, casos de mal uso y patrones de seguridad, junto con herramientas, modelos, estándares y marcos asociados. Estos hallazgos muestran cómo la ciberseguridad se integra en todas las etapas del ciclo de vida del desarrollo de software, con la finalidad de producir productos de software que satisfagan uno de los atributos de calidad más relevantes en la actualidad.

Palabras Clave: Software Seguro; SDLC; Desarrollo de Software Seguro; Ingeniería de Software; Revisión Sistemática de la Literatura; Síntesis Narrativa.

Abstract

Developing software products that incorporate Cybersecurity into their design remains a relatively hard challenge for development teams. Software Engineering has implemented several practices at different stages of the software development lifecycle, aimed at building increasingly secure products to meet industry demands. Through a systematic review of 77 articles, we identified 30 relevant cybersecurity practices, including security policies, threat modeling, static analysis, and penetration testing. Key artifacts such as class diagrams, misuse cases, and security patterns are also highlighted, along with associated tools, models, standards, and frameworks. These findings demonstrate how cybersecurity is integrated into all stages of the software development lifecycle to produce software products that meet one of today's most important quality attributes.

Keywords: Secure Software; SDLC; Software Secure Development; Software Engineering; Literature Review Synthesis, Narrative Synthesis.

Introducción

Hoy en día el software representa una parte esencial para la industria, la economía, el entretenimiento, las comodidades cotidianas y muchas otras áreas que antiguamente estaban desvinculadas de la tecnología. Sin embargo, el crecimiento constante de estas tecnologías trae consigo problemas de ciberseguridad. Casos como el de ION Group en 2023 (Lemos, 2023) son un ejemplo del impacto que las amenazas de ciberseguridad tienen en los mercados globales y que consecuentemente pueden afectar a gran parte de la población en el mundo. Este dato cobra más relevancia al saber que al menos para el año 2022, el 60% de las corporaciones del mundo almacenaban sus datos en la nube (Vailshery, 2022). Estos hechos refuerzan la necesidad de proteger el desarrollo de sistemas seguros ante las actuales y futuras amenazas cibernéticas (Vijayan, 2023).

Dado que el software constituye un área dominante dentro de la dependencia tecnológica actual, la ciberseguridad en el desarrollo de software puede contribuir a mitigar futuros ataques a los sistemas y prevenir la pérdida de activos importantes.

Para los ingenieros de software, la seguridad es una tarea crucial, ya que implica satisfacer una necesidad que no siempre se reconoce de manera consciente. Generalmente un ingeniero de software emplea metodologías que ayudan a crear productos de software de manera sistemática, de modo que el resultado cumpla con las expectativas de funcionalidad y calidad establecidas por los

interesados (stakeholders) (Sommerville, 2011). No obstante, es bien sabido que las metodologías de desarrollo clásicas tienden a dejar el aspecto de la seguridad hasta el final de las actividades críticas (Emami et al., 2010), lo que representa un riesgo para quienes desconocen su importancia real.

Straub (2020) destaca que la ingeniería de software constituye la primera línea de defensa contra los ciberataques y encontró que diversos autores proponen técnicas enfocadas en gestionar la seguridad durante el proceso de desarrollo de software. Estos autores también abogan por la enseñanza de la ciberseguridad como parte de los programas de ingeniería de software. El autor enfatiza que es necesario identificar los requisitos de seguridad y gestionarlos a lo largo del desarrollo, el despliegue y el mantenimiento del software. Por esta razón, han surgido diversas metodologías que buscan implementar la seguridad en el desarrollo de software, y cuyos enfoques se han ido incorporando gradualmente en programas educativos relacionados con las ciencias de la computación (González et al., 2019).

Shift Left Security, por ejemplo, es un enfoque con el cual se pretende incorporar la seguridad a lo largo de todo el ciclo de vida del desarrollo de software. El objetivo es trasladar la seguridad al inicio del proceso de desarrollo para priorizar las prácticas de seguridad antes de que tengan lugar las actividades típicas de desarrollo. Mientras tanto, un enfoque de DevSecOps promueve la comunicación y colaboración entre equipos de desarrollo, operaciones y equipos de seguridad. Al mismo tiempo que involucra un cambio de cultura de trabajo, al priorizar la seguridad a lo largo del proceso de desarrollo de software (Abiona et al., 2024). Ejemplos del desarrollo de software seguro incluyen modelos como Secure Software Development Life Cycle (SSDLC), el cual es un enfoque que integra la seguridad en los procesos del desarrollo de software y que incluye prácticas de seguridad como el desarrollo de requisitos de seguridad, el análisis de riesgos y las pruebas de seguridad transversalmente a las etapas correspondientes en el SDLC (Check Point, 2024).

Por lo tanto, la ingeniería de software representa uno de los medios más adecuados para implementar prácticas de ciberseguridad, de modo que los ingenieros de software puedan crear aplicaciones y otros productos que satisfagan las necesidades de seguridad actuales (González et al., 2019).

El resto del documento se organiza de la siguiente manera: la Sección 2 aborda los trabajos relacionados como antecedentes. La Sección 3 explica el método de investigación utilizado para llevar a cabo el estudio. La Sección 4 describe el proceso de la investigación. La Sección 5 presenta los resultados obtenidos. La Sección 6 discute los hallazgos y sus implicaciones. La Sección 7 expone las amenazas a la validez de nuestro estudio. Finalmente, la Sección 8 concluye el artículo con un resumen de las contribuciones y sugerencias para investigaciones futuras.

Trabajos relacionados

En los últimos años, ha crecido el interés por incluir prácticas de seguridad en el desarrollo de software. Algunos estudios previos abordan enfoques de desarrollo similares al nuestro y, a su vez, demuestran que aún existen áreas de oportunidad para la exploración de temas relacionados con la ciberseguridad y el desarrollo de software.

En el primero de los estudios relacionados, Khan et al. (2020) investigaron el estado del arte de las prácticas de seguridad en software, identificando hasta 37 procesos, modelos y frameworks relacionados con la implementación de seguridad en la ingeniería de software. Aunque este artículo no especifica prácticas de ciberseguridad en particular, resulta un punto de partida útil para considerar estándares y modelos a seguir con el fin de incluir prácticas ampliamente aceptadas en

la industria. Nuestro trabajo, por otra parte, identifica prácticas de ciberseguridad específicas del desarrollo de software y reporta frameworks y modelos relacionados con estas prácticas.

En el trabajo de González et al. (2021) se aborda la necesidad de concientizar a los estudiantes de ingeniería de software sobre la inclusión de la seguridad en las etapas de la ingeniería de requisitos. Propusieron una guía con 14 requisitos y sugerencias para incluir en cursos de desarrollo de software, enfocándose en buenas prácticas de Ciberseguridad y consideraciones para mantener un entorno de trabajo seguro. También sugieren considerar modelos como SSDLC y OWASP Top 10; sin embargo, los autores aclaran que no proponen prácticas específicas alineadas con el ciclo de vida del desarrollo de software. Aunque en este trabajo se trata la concienciación de la Ciberseguridad en los estudiantes de nivel universitario, la mayoría de las consideraciones propuestas por los autores se relacionan con buenos hábitos en el entorno de desarrollo con prácticas de sanitización e higiene. Nuestro trabajo se centra en explorar aquellas prácticas relacionadas con los procesos de la ingeniería del software mediante la clasificación de las prácticas entre las etapas del desarrollo de software.

Posteriormente Khan et al. (2022a) identifican prácticas de seguridad asociadas a un conjunto de riesgos de seguridad. Los autores han reportado 423 prácticas para el desarrollo de software seguro incluyendo las etapas de despliegue y mantenimiento de software. Adicionalmente Khan et al. (2022b) reportan un conjunto de métricas de seguridad relacionadas con la Ingeniería de Software segura. También han reportado herramientas, estándares, y algunos temas de investigación asociados con las prácticas para la seguridad del software, sin embargo, no reportan prácticas específicas. Nuestro trabajo, además de reportar prácticas, también abarca los hallazgos sobre artefactos, herramientas, estándares, modelos de referencia, frameworks e incluso recursos de información de ciberseguridad asociados a las mismas. Además, hemos realizado la asociación entre tales prácticas y herramientas, por lo que es más sencillo diferenciar en qué etapas es adecuado considerar cada una de estas.

Finalmente, Selva-Mora & Quesada-López (2024) realizaron un estudio sobre prácticas de seguridad en el desarrollo ágil. En su investigación, reportan al menos 12 prácticas clave, categorizadas en las etapas de desarrollo del modelo BSIMM. Al mismo tiempo, reportan diversos beneficios de estas prácticas y los diferentes desafíos relacionados con la implementación de estas. Nuestro trabajo, por su parte, reporta un mayor número de prácticas y elementos relacionados con las mismas, los cuales mencionamos antes. Además, en el apartado de discusión de este trabajo hemos analizado los desafíos relacionados con la integración de prácticas de ciberseguridad en el desarrollo de software actual.

A partir de la identificación y clasificación de los hallazgos de este trabajo, se consolida una base documental de referencia que permita a interesados en el tema, profundizar en el diseño de estrategias enfocadas en incluir una perspectiva de ciberseguridad en procesos de desarrollo de software y enseñanza de la ingeniería de software.

Método de investigación

Esta investigación se ha desarrollado mediante una Revisión Sistemática de la Literatura (RSL) siguiendo el método propuesto por Kitchenham et al. (2015).

La metodología de Kitchenham es ampliamente reconocida como el estándar de facto para conducir revisiones sistemáticas en el ámbito de la ingeniería de software. Su adopción se justifica por ofrecer un proceso estructurado y reproducible, compuesto por tres fases claramente definidas:

planificación, ejecución y documentación de la revisión. Este enfoque garantiza la trazabilidad, transparencia de los resultados y aspectos esenciales para mantener la validez científica del estudio.

Si bien existen otras propuestas metodológicas para revisiones sistemáticas, el método de Kitchenham se distingue por su adaptación específica a la investigación empírica en ingeniería de software, a diferencia de enfoques más generales provenientes de otras disciplinas. Además, su amplio uso en la comunidad científica permite comparar y validar resultados con estudios previos, fortaleciendo la consistencia metodológica de esta investigación.

Preguntas de investigación (PI)

Se definieron tres preguntas de investigación que proporcionan una comprensión del estado del arte sobre las principales prácticas de Ciberseguridad utilizadas en el desarrollo de software, incluyendo la identificación de artefactos y diversos aspectos orientados a implementar dichas prácticas.

Pregunta de Investigación	Objetivo
PI-1: ¿Qué prácticas de Ciberseguridad se utilizan en la ingeniería de software?	Identificar las prácticas, actividades o técnicas de Ciberseguridad aplicadas en procesos de ingeniería de software.
PI-2: ¿Cuáles son las fases en las que se aplican las prácticas de Ciberseguridad?	Determinar las fases del SDLC en las que se integran dichas prácticas.
PI-3: ¿Qué tipo de artefactos se producen al realizar prácticas de Ciberseguridad?	Reconocer los artefactos de software comúnmente generados por estas prácticas.

Tabla 1. Preguntas de Investigación

Proceso de Búsqueda

El proceso de búsqueda para identificar estudios relevantes en este campo estuvo guiado por el método de Zhang et al. (2011). Realizamos una búsqueda manual limitada a un pequeño conjunto de estudios relevantes de diversas conferencias relacionadas con Ciberseguridad y desarrollo de software. Con base en estos estudios, se seleccionó un conjunto de palabras clave con las cuales se desarrolló la cadena de búsqueda, misma que a su vez fue evaluada mediante las métricas de **Recall** y **Precision** del método *Quasi-Gold Standard* (Zhang et al., 2011) utilizando los estudios de la búsqueda manual. La cadena de búsqueda fue modificada hasta alcanzar una versión final con un valor recall de 0.8 y una precisión de 0.0073 en la búsqueda de dichos artículos.

	AND		
OR	Software Engineering	Cyber Security	Practice
	Development Process	Computer Security	Approach
	SDLC	Secure Software	Technique
	Software Development	Software Security	Method
	Software Development Life Cycle	DevSecOps	Paradigm
			Framework

Tabla 2. Términos de Búsqueda

Cadena Final:

"Software Engineering" **OR** "Development Process" **OR** "SDLC" **OR**
"Software Development" **OR** "Software Development Life Cycle"
AND
"Cyber Security" **OR** "Computer Security" **OR** "Secure Software" **OR**
"Software Security" **OR** "DevSecOps"
AND
"Practice" **OR** "Approach" **OR** "Framework" **OR** "Method" **OR** "Paradigm" **OR**
"Technique"

Proceso de Selección

Se definieron los criterios de inclusión y exclusión (ver Tabla 3) con el fin de realizar una selección efectiva de artículos. Dichos criterios se dividieron en cuatro fases de la revisión sistemática de la literatura, presentadas en la Tabla 6 de la siguiente sección.

Criterios de Inclusión	Criterios de Exclusión
CI-1: El artículo está en inglés	CE-1: No hay acceso al documento
CI-2: La fecha de publicación está entre 2018 y 2024	CE-2: El documento no es un artículo de investigación
CI-3: El título o resumen responde a una pregunta de investigación	CE-3: El artículo es un estudio secundario
CI-4: El artículo responde al menos a una pregunta de investigación	CE-4: El artículo está repetido

Tabla 3. Criterios de Elegibilidad

Evaluación de la Calidad

Para llevar a cabo la investigación y la selección de artículos de calidad, se adoptaron algunas de las preguntas propuestas por el enfoque de Kitchenham (2015, pp. 83--85). La puntuación de cada estudio se determinó en función del cumplimiento de cada pregunta: 1 punto si se cumplía, 0.5 puntos si se cumplía parcialmente, y 0 puntos si no se cumplía. Las preguntas para evaluación de la calidad se enlistan a continuación:

1	¿Está el artículo basado en una investigación (o es meramente un reporte basado en 'lecciones aprendidas' bajo la opinión de un experto)?
2	¿Se enuncian claramente los objetivos de la investigación?
3	¿Se describe adecuadamente el contexto en el que se llevó a cabo la investigación?
4	¿El diseño de la investigación fue adecuado para abordar los objetivos de la investigación?
5	¿El análisis de los datos fue lo suficientemente riguroso para los objetivos de la investigación?
6	¿Se reportan claramente los hallazgos del estudio?
7	¿Se comunica adecuadamente a la audiencia las conclusiones, las implicaciones para la práctica y la investigación futura?

Tabla 4. Preguntas de Calidad

Proceso de Extracción y Síntesis de Datos

El formato de extracción de datos, disponible en el Anexo C, permitió separar los hallazgos de cada artículo en relación con las preguntas de investigación. Para analizar los hallazgos, sintetizamos la información mediante un enfoque de Síntesis Narrativa (*narrative synthesis*), de acuerdo con Popay et al. (2006). Este enfoque incluye el desarrollo de una teoría de cambio, la tabulación de los datos, el análisis de relaciones entre ellos y la evaluación de la solidez de la síntesis.

DATOS GENERALES	
Título	
Autor(es)	
Año	
Fuente	
Tipo de Publicación	

Referencia	
Palabras clave	
Abstract o resumen	
Pregunta de investigación que responde	PI-1 Práctica:
	PI-2 Etapa del desarrollo:
	PI-3 Artefactos:
Notas	
ANÁLISIS GENERAL DEL ESTUDIO	

Tabla 5. Formato de extracción de información de los estudios primarios

Conducción de la revisión

El proceso de búsqueda consistió en aplicar los criterios de elegibilidad a todos los registros recuperados de las bibliotecas digitales seleccionadas: IEEE Xplore, ACM Digital Library, SpringerLink y ScienceDirect, a través de las siguientes etapas:

- Etapa 0: Se ejecutó la cadena de búsqueda sin criterios de elegibilidad.
- Etapa 1: Se excluyeron aquellos registros que no cumplían con los criterios básicos de inclusión: publicaciones en inglés, con acceso al texto completo y publicadas entre 2018 y 2024.
- Etapa 2: Se excluyeron los documentos que no eran considerados artículos de investigación primaria.
- Etapa 3: Se excluyeron los estudios secundarios como revisiones sistemáticas y las entradas duplicadas. Se incluyeron los estudios cuyos títulos y resúmenes mostraban indicios de responder al menos a una de las preguntas de investigación.
- Etapa 4: Se incluyeron aquellos artículos cuyo contenido respondía de manera efectiva a una o más de las preguntas de investigación definidas. La cantidad de artículos durante el proceso de filtrado se resume en la Tabla 6.

Etapas	IEEE Xplore	ACM DL	Springer	Science Direct	Total
0	1954	11949	7789	3638	25330
1	1160	6405	3367	2189	13121
2	1107	400	1924	1762	5193
3	96	28	33	44	201
4	40	14	10	13	77

Tabla 6. Resultado por cada etapa de búsqueda

Para la evaluación de calidad disponible en el Anexo B, la puntuación promedio de los estudios alcanzó los 6.53 de un total de 7 puntos. Además, 18 estudios obtuvieron la puntuación máxima. La deficiencia más común en los estudios se debió a la falta del contexto especifico en el que se llevaron a cabo los estudios. Este resultado indica un alto nivel de calidad metodológica en los estudios seleccionados.

Resultados

Se revisaron 77 estudios publicados entre 2018 y 2024, disponibles en el Anexo A,. Los años con mayor número de estudios fueron 2019, 2023 y 2024 (Ver Figura 1). El 52% de estos artículos provienen de congresos, mientras que el 48% restante corresponde a artículos de journals. IEEE Xplore es la fuente con la mayor distribución de artículos, representando 40 de los 77 artículos en total. De las demás fuentes, se incluyen 14 estudios de la ACM Digital Library, 13 de Science Direct y 10 de Springer Link (Ver Figura 2). Si bien IEEE Xplore presenta la mayor concentración de estudios, comparte publicaciones de conferencias con la ACM Digital Library, por lo que algunos estudios duplicados fueron excluidos de ACM.

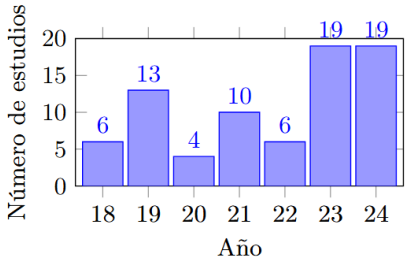


Figura 1. Distribución de los estudios hallados por año de publicación

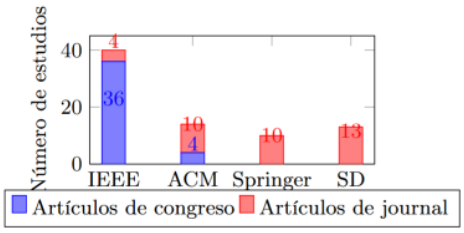


Figura 2. Tipos de publicaciones por fuente

Respuesta a las Preguntas de Investigación (PI) 1 y 2

Para responder a las Preguntas de Investigación 1 y 2 (PI-1: ¿Qué prácticas de Ciberseguridad se utilizan en la ingeniería de software?; PI-2: ¿En qué fases del ciclo de vida del desarrollo de software se aplican?), los hallazgos reportados en la Tabla 7 muestran que las prácticas identificadas se aplican a lo largo de todo el ciclo de vida, con mayor concentración en las etapas iniciales. La aplicación de políticas de seguridad asegura el cumplimiento de estándares y establece controles organizacionales, mientras que la gestión de riesgos, el modelado de casos mal uso / uso

indebido (Misuse Cases) y las historias de usuario permiten recopilar y definir requisitos de seguridad. En la fase de diseño, la práctica Threat Modeling evalúa y aborda riesgos. Durante el desarrollo, el análisis estático de código, la revisión de código y las pruebas unitarias e integrales controlan la calidad y detectan vulnerabilidades tempranas. Finalmente, las pruebas de software, incluyendo carga, estrés, rendimiento y estructurales, buscan vulnerabilidades que solo se manifiestan en ejecución, asegurando la resiliencia del software en entornos exigentes.

Práctica	ID del Artículo	Frec.	Fase SDLC
Threat Modeling	1, 4, 6, 7, 8, 10, 11, 15, 19, 20, 25, 33, 36, 37, 38, 42, 43, 47, 48, 59, 60, 66, 67, 68, 69, 70	26	Software Design
Risk Management	1, 2, 3, 4, 5, 7, 17, 20, 22, 31, 33, 35, 43, 47, 49, 50, 55, 65, 67, 71	19	Software Requirements
Static Code Analysis	7, 11, 13, 18, 23, 26, 27, 30, 34, 36, 37, 39, 40, 41, 65, 67, 69, 70	18	Software Construction
Apply Security Policies and Regulations	2, 4, 5, 7, 13, 17, 20, 21, 28, 32, 33, 43, 55, 56, 65	15	All Phases
Model Misuse Cases	6, 9, 12, 16, 19, 20, 23, 31, 34, 35, 39, 67	14	Software Design
Code Review	4, 11, 19, 20, 21, 28, 30, 31, 41, 49, 65, 67	12	Software Construction
Design Security Architecture	4, 5, 9, 1, 25, 27, 56, 66, 67, 70	10	Software Design
Penetration Testing	4, 6, 13, 23, 27, 28, 33, 36, 45	9	Software Testing
Continuous Monitoring	3, 14, 20, 21, 25, 26, 33, 47, 56	9	Software Construction & Testing
Dynamic Code Analysis	13, 26, 34, 36, 38, 39, 40, 69	8	Software Construction
Model Abuse Cases	2, 6, 12, 20, 23, 67, 70	7	Software Requirements
Automating Testing	11, 15, 16, 21, 32, 67, 36	7	Software Construction & Testing
Design Review	4, 5, 11, 15, 20, 21	6	Software Design
Security Audits	4, 6, 11, 20, 41, 66	6	All Phases

Include Coding Standard / Coding Rules	3, 6, 15, 17, 67	5	Software Construction
Fuzz Testing	20, 21, 28, 65	4	Software Testing
Apply metrics to software evaluation	3, 21, 27	3	All Phases
Separation of Roles	4, 20, 24	3	Software Construction & Testing
Unit Testing	4, 21	2	Software Construction
Use of Version Control System	16, 49	2	Software Construction
Include Security Patterns	44, 56	2	Software Design
Integration Testing	16, 21	2	Software Testing
Privacy Testing	4, 21	2	Software Testing
Simulate Attack	4, 39	2	Software Testing
Validate Input and Outputs	14	1	Software Construction & Testing
Regression Testing	21	1	Software Testing
Load Tests	21	1	Software Testing
Stress Test	21	1	Software Testing
Performance Test	21	1	Software Testing
Structural Testing	21	1	Software Testing

Tabla 7. Prácticas de Ciberseguridad

Respuesta a la Pregunta de Investigación (PI) 3

Para responder a la PI-3 (¿Qué tipo de artefactos se producen al aplicar prácticas de Ciberseguridad?), los resultados mostrados en la Tabla 8 indican que el artefacto más recurrente es el Misuse Case Diagram, vinculado con los diagramas de casos de uso y presente en el 22.5% de los estudios. Otros artefactos destacados incluyen, en la etapa de requisitos, el Abuse Case Diagram, el Context Diagram y el Domain Model; y en la etapa de diseño de software, diagramas de clases, diagramas de flujo de datos y de secuencia, árboles de ataque y varios diagramas de patrones de diseño. En las etapas posteriores del desarrollo no se identificaron artefactos

específicos, salvo el uso implícito de diagramas de flujo de datos en pruebas durante la construcción.

Artefacto	ID del Artículo	Frec.	Fase SDLC
Misuse Cases / Abuse Case Diagram	2, 6, 9, 12, 16, 19, 20, 23, 31, 34, 35, 39, 41, 67, 74	15	Software Requirements & Software Design
Use Case Diagram	2, 9, 12, 16, 31, 34, 39, 44, 56	9	Software Design
Class Diagram	2, 9, 20, 44, 56, 69, 70	7	Software Design
Data Flow Diagram	1, 8, 10, 42, 70	5	Software Design
Sequence Diagram	2, 56, 44, 69	4	Software Design
Checklists	4, 13, 69	3	Verification & Validation (V&V)
Context Diagram	1, 2	2	Software Requirements
Activity Diagram	44, 56	2	Software Design
Abstract Security Pattern	44, 56	2	Software Design
Architectural Security Pattern Diagram	44, 56	2	Software Design
Security Solution Frames	44, 56	2	Software Design
Threats and Type of Threats List	1	1	Software Requirements
Abuse Histories	19	1	Software Requirements
Malicious User Stories	19	1	Software Requirements
Misuse Patterns	44	1	Software Design
Security Cluster	44	1	Software Design
Cloud Security and Privacy Metamodel	44	1	Software Design
Attack Tree Diagram	55	1	Software Design
State Diagram	56	1	Software Design

Tabla 8. Artefactos producto de la aplicación de Prácticas de Ciberseguridad en los procesos de Ingeniería de Software

Otros Hallazgos

Además de las prácticas y artefactos, este estudio identificó herramientas, estándares, modelos, marcos de trabajo y algunos recursos de información relevantes para la ciberseguridad en el desarrollo de software. La Tabla 9 resume estos hallazgos, destacando normas como ISO 31000, ISO/IEC 27005 y la metodología OCTAVE, que proporcionan lineamientos para la gestión de riesgos. Muchos de estos elementos se aplican a múltiples prácticas y consideran aspectos organizacionales relacionados con la implementación de la Ciberseguridad en la creación de software.

Estándar / Modelo / Framework	ID del Artículo	Frec.
DevSecOps	8, 11, 13, 24, 25, 26, 27, 32, 33, 34, 35, 37, 39, 41, 57, 58, 59, 61, 62	19
Microsoft Security Development Lifecycle (Microsoft SDL)	3, 6, 11, 20, 35, 36, 38, 40, 43, 50	10
Microsoft STRIDE	1, 8, 20, 43, 44, 66, 68, 70	8
BSIMM	5, 6, 7, 11, 20, 35, 50, 66	8
OWASP SAMM	5, 7, 11, 20, 21, 50, 66	7
SAFECode	6, 7, 15, 20, 21, 67	6
OWASP Top Ten	8, 11, 18, 21, 27, 55	6
OWASP CLASP	3, 6, 11, 16, 20, 50	6
Common Weakness Enumeration (CWE)	18, 65, 69, 70	4
SQuaRE (ISO 25000:2005)	5, 21, 35, 50	4
CMMI	3, 4, 5, 35	4
NIST SP 800	3, 17, 22	3
Common Criteria	7, 20, 67	3
ISO/IEC 27001	8, 21, 43	3
ISO/IEC 25010	16, 21, 43	3
SSDLC / Secure SDLC	4, 36	2
Secure TROPOS	5, 35	2

SDL Agile	6, 11	2
Common Vulnerabilities and Exposures (CVE)	18, 69	2
Software Security Touchpoints	20, 67	2
OCTAVE	29, 43	2
NICE Framework	55, 65	2
Microsoft DREAD Model	29, 43	2
ISO 3100	1	1
Security Technical Implementation Guide (STIG)	4	1
CIS Benchmark	4	1
IEC 63443	4	1
Security Requirements Engineering Process (SREP)	5	1
Software Security Approach (SWSec)	6	1
COCOMO II	6	1
IFPUG	6	1
COSMIC Function Points	6	1
SEER-SEM	6	1
Quality Assurance (QA)	8	1
ISMS (Information Security Management System)	8	1
Oracle Software Security Assurance (OSSA)	11	1
TSP-Secure	11	1
OWASP Proactive Controls	14	1
Fisa-XP	16	1
SecSDM	17	1
IEEE 1220-2005	17	1

IATF (Information Assurance Technical Framework)	17	1
ISO/IEC TR 13335-3 (Withdrawn)	17	1
SEED Project (Hands-on Labs)	18	1
OWASP ASVS	19	1
Securosis	20	1
ISO/IEC 15939:2002	21	1
MITRE ATT&CK®	21	1
ISO/IEC/IEEE 15288	22	1
ISO/IEC/IEEE 42010	22	1
ISO/IEC/IEEE 29119	23	1
NICE NCWF	24	1
CAE CD core Kus	26	1
SSE-CMM	28	1
VAHTI	28	1
KATAKRI	28	1
CAPEC	31	1

Tabla 9. Otros Hallazgos

La Tabla 10 presenta la lista de diversas herramientas mencionadas en los estudios, las cuales están asociadas a varias de las prácticas identificadas. Aunque la mayoría de las herramientas se relacionan con prácticas como el análisis estático de código y distintas pruebas de software, algunas resultan útiles para la identificación de riesgos y amenazas de seguridad, como *Protection Poker* y *Elevation of Privilege Card Game*. En cuanto al software seguro, encontramos herramientas como *UMLsec* y *UMLpac*, que permiten a los ingenieros de software crear diseños UML con características de seguridad.

Herramienta	ID del Artículo	Práctica Asociada
UMLsec	2, 6, 9, 23, 35, 71	Security Requirements and Design
Protection Poker	19, 23, 66	Risk Assessment
SonarQube	23, 27, 36	Static Code Analysis
Microsoft Threat Modelling Tool	8, 43	Threat Modeling
FlawFinder	18, 65	Static Code Analysis
CheckMarx	23, 27	Static Code Analysis
OWASP Zed Attack Proxy (ZAP)	23, 36	Penetration Testing
Burp Suite	23, 36	Penetration Testing
UMLpac	2	Security Requirements and Design
Elevation of Privilege (EoP) Card Game	8	Threat Modeling
Vulnerable Software Packages	10	Software Security Training
Loggly by Solar Winds	18	Log Analysis
LowViewPlus	18	Log Analysis
Clang-Tidy	18	Static Code Analysis
HP Fortify SCA	23	Static Code Analysis
Jenkins	23	Static Code Analysis
Threadfix	23	Static Code Analysis
JMeter	23	Load and Performance Testing
Postman	23	Load and Performance Testing
BlazeMeter	23	Load and Performance Testing
Goldeneye	23	Penetration Testing
HTTPerf	23	Performance Testing
SoapUI	23	Web Testing
CA APM	23	Performance Testing
Arachni	23	Penetration Testing

XSS ME	23	Penetration Testing
SQL Injection ME	23	Penetration Testing
Meta Exploit	23	Penetration Testing
NMAP	23	Penetration Testing
Whatweb	23	Penetration Testing
Snyk	24	Static Code Analysis
Retire JS	24	Static Code Analysis
Hakiri	24	Static Code Analysis
PVS-Studio	24	Static Code Analysis
Veracode	24	Static Code Analysis
Coverity	24	Static Code Analysis
Rational Team Concert	33	Security Audit
Rational Quality Manager	33	Software Testing
Bugzilla	69	Error Tracing

Tabla 10. Herramientas para el desarrollo de software seguro

Discusión

Se identificaron un total de 30 prácticas de Ciberseguridad asociadas a las diferentes etapas del Ciclo de Vida del Desarrollo de Software (SDLC, por sus siglas en inglés). Aunque la Tabla 7 muestra una concentración de prácticas vinculadas con las fases de prueba y construcción, las prácticas consideradas más críticas —la gestión de riesgos y el modelado de amenazas— se ubican en las etapas iniciales de requisitos y diseño. Esto coincide con la literatura previa, que subraya la necesidad de incorporar la Ciberseguridad desde las fases tempranas del desarrollo, donde las decisiones arquitectónicas tienen un impacto duradero.

Los *Misuse Cases* (y su variante llamada *Abuse Cases*) constituyen otra práctica altamente citada en los estudios, especialmente en la fase de diseño, donde se utilizan para refinar los requisitos de seguridad. Su estrecha relación con los diagramas de casos de uso tradicionales facilita la identificación de amenazas y comportamientos no deseados, lo que explica su amplia adopción en la literatura revisada. Dado que estas prácticas generan artefactos tangibles, los *Misuse Cases* también fueron catalogados en la tabla de artefactos cada vez que los autores los mencionaron como tales.

Con respecto a los estándares y metodologías, OWASP fue el marco más citado, especialmente mediante sus proyectos específicos (*Top 10, CLASP y SAMM*), lo que evidencia una fuerte orientación práctica en los estudios revisados. También se identificaron referencias a *BSIMM,*

CMMI, NIST e *ISO/IEC 27001*, lo que muestra un esfuerzo por adaptar buenas prácticas corporativas y marcos de gestión de seguridad al ámbito del desarrollo de software. Este hallazgo sugiere una convergencia entre enfoques organizacionales y de ingeniería.

El análisis de herramientas refleja una tendencia creciente hacia la automatización de la seguridad. Herramientas como *UMLsec*, *Protection Poker* y las de análisis estático de código (*FlawFinder, SonarQube, Checkmarx*) muestran que la comunidad busca integrar la seguridad dentro de flujos ágiles y DevSecOps. No obstante, pocos estudios discuten la efectividad de estas herramientas en escenarios reales o su adopción en pequeñas y medianas empresas, lo que representa un vacío en la literatura.

Asimismo, varios trabajos destacaron el uso de métricas para monitorear la calidad y seguridad del software, aunque su aplicación sistemática aún es limitada. Esto apunta a la necesidad de desarrollar indicadores más específicos que permitan evaluar la incorporación de prácticas de Ciberseguridad a lo largo del SDLC.

Finalmente, un hallazgo transversal es la dificultad de integrar efectivamente la Ciberseguridad en los procesos de desarrollo de software actuales. La falta de concienciación y formación en Ciberseguridad, tanto en el área académica como en la industrial, sigue siendo una barrera persistente en los diversos contextos en los que opera el desarrollo de software actual, siendo el contexto organizacional algo que pocos estudios consideran al momento de considerar mejores prácticas.

A pesar de los avances conceptuales y metodológicos, aún existe una brecha entre la teoría y la práctica en la implementación de software seguro. Si bien muchos de los procesos metodológicos recomendados en la literatura contienen pautas generales para la implementación de mejores prácticas, no siempre han sido claras o especificas las condiciones en las que estas fueron efectivas.

Por otro lado, aunque en la literatura existe diversa información sobre mejores prácticas independientemente del rubro dentro del desarrollo de software, es poco frecuente que se mencionen específicamente aquellas malas o peores prácticas que tanto los ingenieros de software como las organizaciones, podrían estar permitiendo sin ser conscientes de ello. Aunque muchas de las mejores prácticas, manejan de manera implícita aquellos malos hábitos en el desarrollo de software, algunos otros podrían no ser tan evidentes si se omite su análisis.

Por lo tanto, es posible ver que existen áreas de oportunidad si se analizan las partes contrarias a la información que habitualmente tratamos de hallar. De este modo es posible extender este trabajo hacia mejores y peores prácticas, beneficios y perjuicios de aplicarlas, y los riesgos y oportunidades según el contexto en el que estas se apliquen.

Amenazas a la Validez

Entre las principales amenazas a la validez se encuentran la limitada selección de estudios debido a restricciones de acceso en las fuentes de información, así como posibles sesgos de interpretación. Para mitigar dichas amenazas, se utilizaron los enfoques sistemáticos propuestos por Kitchenham et al. (2015) y el enfoque Quasi-Gold Standard de Zhang et al. (2011). Junto con la supervisión de asesores y la retroalimentación de expertos, se contribuyó a mitigar dichos sesgos y mejorar la validez de los resultados.

Conclusión

El objetivo de esta investigación fue identificar y analizar prácticas de ciberseguridad aplicadas en el desarrollo de software y compatibles con los fundamentos de la ingeniería de software, integrando evidencia proveniente de la literatura científica mediante una revisión sistemática de la

literatura. Los resultados muestran que las prácticas más relevantes —gestión de riesgos, modelado de amenazas y diseño de *misuse cases*— se concentran en las etapas tempranas del SDLC, confirmando la importancia de incorporar la seguridad desde el inicio del proceso de desarrollo. Asimismo, se identificó una fuerte presencia de marcos y estándares reconocidos (OWASP, NIST, ISO), así como un creciente uso de herramientas de análisis estático y metodologías ágiles con enfoque DevSecOps.

Este trabajo aporta una sistematización actualizada de prácticas, artefactos, herramientas y frameworks de ciberseguridad en la ingeniería de software, ofreciendo una visión integral que conecta la teoría con la práctica. Sus hallazgos pueden servir como referencia para investigadores, docentes y profesionales interesados en incorporar la ciberseguridad en procesos de desarrollo reales y en programas educativos. Como implicación práctica, los resultados apoyan el diseño de estrategias formativas y metodológicas orientadas a fortalecer la conciencia y competencia en ciberseguridad dentro de equipos de desarrollo. En el ámbito académico, contribuye a consolidar la base de conocimiento sobre ciberseguridad en ingeniería de software, identificando tendencias y vacíos para futuras investigaciones.

Entre las líneas futuras destacan la integración de la ciberseguridad en entornos ágiles, el desarrollo de métricas que midan la madurez de la seguridad durante el desarrollo de proyectos de software, la evaluación empírica del impacto de estas prácticas en proyectos de software reales y la exploración profunda sobre las implicaciones individuales, organizacionales y contextuales al integrar prácticas de ciberseguridad.

Referencias

[1]. Lemos, R. (2023, February). *Cyberattack on fintech firm disrupts derivatives trading globally*. Dark Reading. https://www.darkreading.com/cyberattacks-data-breaches/cyberattack-fintech-firm-disrupts-derivatives-trading

[2]. Vijayan, J. (2023, January). *How noob website hackers can become persistent threats*. Dark Reading. https://www.darkreading.com/cyberattacks-data-breaches/noob-hackers-become-persistent-threats

[3]. Vailshery, L. S. (2022, March). *Share of corporate data stored in the cloud in organizations worldwide from 2015 to 2022*. Statista. https://www.statista.com/statistics/1062879/worldwide-cloud-storage-of-corporate-data/

[4]. Sommerville, I. (2011). *Software engineering* (9th ed.). Addison-Wesley.

[5]. Emami, M. S., Ithnin, N. B., & Ibrahim, O. (2010). Software process engineering: Strengths, weaknesses, opportunities and threats. In Proceedings of the 6th International Conference on Networked Computing (INC2010).

[6]. Straub, J. (2020). Software engineering: The first line of defense for cybersecurity. In *2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS)*.

[7]. González, H., Llamas Contreras, R., & Montaño Rivas, O. (2019). When software engineering meets cybersecurity at the classroom. In *2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT)*.

[8]. Khan, R. A., Khan, S. U., Ilyas, M., & Idris, M. Y. (2020). The state of the art on secure software engineering: A systematic mapping study. In *Proceedings of the 24th International Conference on Evaluation and Assessment in Software Engineering (EASE '20)*.

[9]. González, H., Llamas Contreras, R., & Guerra García, C. (2021). Cybersecurity practices at the initial stages of the software engineering process. In *2021 9th International Conference in Software Engineering Research and Innovation (CONISOFT)*.

[10]. Khan, R. A., Khan, S. U., Khan, H. U., & Ilyas, M. (2022a). Systematic literature review on security risks and its practices in secure software development. *IEEE Access, 10*, 5456–5481.

[11]. Khan, R. A., Khan, S. U., & Ilyas, M. (2022b). Exploring security procedures in secure software engineering: A systematic mapping study. In *Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering (EASE '22)*.

[12]. Selva-Mora, A., & Quesada-López, C. (2024). Security practices in agile software development: A mapping study. In *Proceedings of the 7th ACM/IEEE International Workshop on Software-intensive Business (IWSiB '24)*.

[13]. Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). *Evidence-based software engineering and systematic reviews*. Chapman & Hall/CRC.

[14]. Zhang, H., Babar, M. A., & Tell, P. (2011). Identifying relevant studies in software engineering. *Information and Software Technology, 53*(6), 625–637.

[15]. Popay, J., Roberts, H., Sowden, A., Petticrew, M., Arai, L., Rodgers, M., Britten, N., Roen, K., & Duffy, S. (2006). *Guidance on the conduct of narrative synthesis in systematic reviews*. ESRC Methods Programme. https://www.lancaster.ac.uk/media/lancaster-university/content-assets/documents/fhm/dhr/chir/NSsynthesisguidanceVersion1-April2006.pdf

[16]. Abiona, O. O., Oladapo, O. J., Modupe, O. T., Oyeniran, O. C., Adewusi, A. O., & Komolafe, A. M. (2024). The emergence and importance of DevSecOps: Integrating and reviewing security practices within the DevOps pipeline. *World Journal of Advanced Engineering Technology and Sciences, 11*(2), 127–133.

[17]. Check Point. (2024). *What is secure SDLC?* https://www.checkpoint.com/cyber-hub/cloud-security/what-is-secure-sdlc/

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

NOTAS BIOGRÁFICAS

Eduardo Antonio Castillo Garrido es Licenciado en Ingeniería de Software por la Facultad de Estadística e Informática de la Universidad Veracruzana en Xalapa, Veracruz. Sus áreas de interés académico y profesional incluyen la ingeniería de requisitos, la arquitectura de software y el diseño de sistemas resilientes. El presente trabajo se deriva de su proyecto de titulación en dicha institución.

Juan Carlos Pérez Arriaga es Licenciado en Informática por la Facultad de Estadística e Informática de la Universidad Veracruzana. Maestro en Ciencias de la Computación por la Fundación Arturo Rosenblueth. Actualmente es profesor de tiempo completo de la Facultad de Estadística e Informática de la Universidad Veracruzana. Cuenta con el reconocimiento del Programa de Desarrollo Profesional Docente (PRODEP). Su labor investigadora comprende las áreas de construcción de software, seguridad en el desarrollo de software y accesibilidad en el desarrollo de software.

Héctor Xavier Limón Riaño trabaja como docente de tiempo completo en la facultad de Estadística e Informática de la Universidad Veracruzana. Actualmente es miembro del Sistema Nacional de Investigadores. Es Licenciado en Informática, con maestría y Doctorado en Inteligencia Artificial. Sus áreas de investigación de interés son Sistemas Multi-Agente, Minería de datos, Ciberseguridad y Sistemas Distribuidos; contando con diversas publicaciones en estas áreas.

Saúl Domínguez-Isidro es profesor e investigador en la Facultad de Estadística e Informática de la Universidad Veracruzana, México. Sus intereses de investigación se centran en ingeniería de software, particularmente en search-based software testing, algoritmos bioinspirados, pruebas automatizadas de software, DevOps y aplicaciones de inteligencia artificial en el desarrollo de software. Ha participado en diversos proyectos de investigación y colabora activamente en la formación de estudiantes de licenciatura y posgrado en ciencias de la computación e ingeniería de software.

Recibido 2 Nov 2025 *ReCIBE, Año 15 No. 1, mayo 2026*

Aceptado 22 Abri 2026

Técnicas de Análisis de Supervivencia para la Identificación de Deserción en Estudiantes: Una Revisión Sistemática de la Literatura

Survival Analysis Techniques for Identifying Student Dropout: A Systematic Literature Review

Franz Jesús Rivera-Alcántara [1]
franz.j.rivera@outlook.com
Ángel J. Sánchez-García [1]
angesanchez@uv.mx
José Juan Muñoz-León [1]
juanmunoz@uv.mx
Jorge Octavio Ocharán-Hernández [1]
jocharan@uv.mx
José Luis Sánchez-Leyva [2]
luissanchez01@uv.mx

[1] *Facultad de Estadística e Informática,*
Universidad Veracruzana,Xalapa-Enríquez, Veracruz, México.

[2] *Facultad de Contaduría y Administración,*
Universidad Veracruzana,Coatzacoalcos, Veracruz, México.

Resumen
Este trabajo presenta una Revisión Sistemática de la Literatura (RSL) sobre el uso de técnicas de Análisis de Supervivencia (AS) para predecir la deserción estudiantil para el problema de Ingeniería de Software. La educación en Ingeniería de Software enfrenta altas tasas de abandono, lo que resalta la necesidad de enfoques predictivos que apoyen la retención escolar. El AS, aplicado con éxito en áreas como la salud, permite modelar tanto la ocurrencia como el momento de un evento, ofreciendo ventajas frente a otros métodos. Siguiendo las directrices de Kitchenham y el enfoque Quasi-Gold Standard, se identificaron 36 estudios primarios publicados entre 2013 y 2025. Los resultados muestran que el modelo de riesgos proporcionales de Cox es la técnica más utilizada, junto con Kaplan-Meier y otros enfoques de regresión. Asimismo, se identificaron 29 variables relevantes y 16 métricas de validación. La revisión evidencia que el AS facilita la detección temprana de estudiantes en riesgo, aunque enfrenta retos como la necesidad de datos extensos y su limitada adopción en el ámbito educativo.

Palabras Clave: Educación en Ingeniería de Software; Análisis de Supervivencia; Deserción estudiantil; Revisión Sistemática de la Literatura.

Abstract
This work presents a Systematic Literature Review (SLR) on the use of Survival Analysis (SA) techniques to predict student dropout in the context of Software Engineering. Software Engineering education faces high attrition rates, highlighting the need for predictive approaches that support student retention. SA, successfully applied in fields such as healthcare, enables modeling both the occurrence and timing of an event, offering advantages over other methods. Following Kitchenham's guidelines and the Quasi-Gold Standard approach, 36 primary studies published between 2013 and 2025 were identified. The results show that Cox proportional hazards model is the most widely used technique, along with Kaplan-Meier and other regression approaches. In addition, 29 relevant variables and 16 validation metrics were identified. The review demonstrates that SA facilitates the early detection of at-risk students, although it faces challenges such as the need for extensive datasets and its limited adoption in the educational domain.

Keywords: Software Engineering Education; Survival Analysis; Student dropout; Systematic Literature Review.

1. Introducción

La Ingeniería de Software (IS) es una disciplina que adopta enfoques de ingeniería para desarrollar software de calidad [1]. La IS brinda técnicas de apoyo a la especificación, diseño y evolución del software, garantizando su calidad [2]. Para tal efecto, el ingeniero de software adopta un enfoque sistemático y organizado de su trabajo para producir software de alta calidad [2].

La educación en la IS desempeña un papel esencial en la formación de los estudiantes sobre tecnologías, procesos y prácticas que aseguren la calidad en el desarrollo de software [3]. Sin embargo, el número de desarrolladores de software titulados en IS es pequeño en comparación con el número de personas que desarrollan o modifican software [4]. El estudio de la deserción escolar (DE) en la IS y campos relacionados con la informática ha recibido una mayor atención debido a la necesidad de profesionales en estas carreras [5].

El Análisis de Supervivencia (AS) surge como propuesta por parte de investigadores de las comunidades de aprendizaje automático y minería de datos para enfrentar esta problemática [6]. El AS permite identificar los factores que afectan un evento de interés

[7]. En el ámbito de la DE, el AS facilita la identificación de estudiantes en riesgo y de los factores que influyen en esta condición, mediante el análisis de los datos presentados en su perfil de ingreso (PI) y TE.

A pesar del éxito de los métodos de AS en ámbitos como la sanidad, la salud y en recientes estudios enfocados en la DE, la aplicación de estos métodos ha sido escasamente explorada en comparación con otros enfoques en este problema. Esta carencia en la aplicación del AS en la DE resalta la necesidad de realizar una Revisión Sistemática de la Literatura (RSL) centrada en las técnicas de AS utilizadas como colaboración para resolver el problema de la DE.

Este trabajo aborda dicha brecha mediante una Revisión Sistemática de la Literatura (RSL) que consolida y analiza 36 estudios primarios publicados entre 2013 y 2025, siguiendo las directrices de Kitchenham y el enfoque Quasi-Gold Standard (QGS). La revisión ofrece un mapeo estructurado de las técnicas de AS más utilizadas (destacando la prevalencia del modelo de Riesgos Proporcionales de Cox y el estimador Kaplan-Meier) junto con 29 variables relevantes y 16 métricas de validación. Asimismo, identifica los principales retos y oportunidades de investigación para mejorar la detección temprana de estudiantes en riesgo. Al sintetizar el estado del arte, este trabajo aporta una referencia fundamental para futuras investigaciones y el desarrollo de modelos predictivos orientados a fortalecer la retención estudiantil en la educación en Ingeniería de Software.

El resto de este artículo se organiza de la siguiente manera: la Sección 2 presenta los trabajos relacionados. La Sección 3 describe la metodología aplicada en este estudio. La Sección 4 reporta los resultados experimentales, mientras que la Sección 5 ofrece una discusión de los hallazgos. La Sección 6 expone las amenazas a la validez, y la Sección 7 presenta las conclusiones.

2. Trabajo relacionado

Uno de los primeros antecedentes del uso de Análisis de Supervivencia (AS) aplicado a la deserción escolar (DE) fue el trabajo de Plank y colaboradores, quienes emplearon el modelo de regresión de Cox para estudiar la relación entre la DE en educación secundaria y el plan de estudios. En particular, analizaron cómo la combinación de educación técnica (CTE) y cursos académicos influía en la probabilidad de abandono, concluyendo que la exposición a ambos tipos de formación podía fortalecer la motivación y el apego del estudiante a la escuela [8]. Posteriormente, Ameri et al. [6] propusieron un marco de predicción temprana de la DE en la Wayne State University, también basado en el modelo de Cox, incorporando variables demográficas, socioeconómicas y del trayecto escolar. Sus resultados evidenciaron la utilidad de información previa a la matrícula y de factores académicos como créditos aprobados/reprobados y promedio semestral, para identificar estudiantes en riesgo y orientar intervenciones oportunas [6].

En un estudio más reciente, Lisboa y colaboradores [9] analizaron la DE en estudiantes de Ciencias de la Computación de la Universidad de Brasilia (UnB) mediante un modelo de regresión Log-Normal. Este permitió estimar el número de semestres hasta la deserción o el último seguimiento, destacando la influencia de factores como la modalidad de ingreso y el desempeño académico en la probabilidad de abandono [9]. Estos estudios reflejan la utilidad del AS como técnica robusta para modelar y comprender los factores asociados a la deserción en distintos niveles educativos y disciplinas.

A pesar de la creciente preocupación por las altas tasas de deserción en los programas de Ingeniería de Software, la literatura carece de una síntesis sistemática sobre la aplicación de técnicas de AS para predecir la deserción estudiantil en este dominio. Aunque el AS ha sido empleado con éxito en áreas como la salud para modelar tanto la

ocurrencia como el momento de un evento, su adopción en contextos educativos (y particularmente en la educación en Ingeniería de Software) sigue siendo limitada y dispersa. Los estudios existentes suelen centrarse en métodos de clasificación o regresión tradicionales, sin considerar la dimensión temporal que el AS permite analizar. Además, no existe una revisión integral que identifique qué técnicas, variables y métricas de validación han resultado más efectivas en la predicción de la deserción estudiantil.

En conjunto, la evidencia muestra la preocupación de las instituciones por fortalecer la retención escolar (RE) y alcanzar una mayor eficiencia terminal (ET). La deserción, particularmente en áreas de informática, sigue mostrando tasas elevadas pese a la creciente demanda de profesionales en Ingeniería de Software. Ante la insuficiencia de estrategias tradicionales, la aplicación del AS se presenta como una alternativa prometedora para anticipar riesgos y diseñar intervenciones focalizadas. Sin embargo, la escasez de estudios que empleen esta técnica en el análisis de la DE en programas de IS, junto con los crecientes índices de abandono, justifican la pertinencia de continuar investigando en esta línea.

3. Método de investigación

La metodología seguida para esta RSL se basó en las directrices propuestas en [10], que define tres etapas principales: planificación, ejecución y presentación de los hallazgos. Durante la etapa de ejecución, se aplicó una estrategia de búsqueda basada en el enfoque Quasi-Gold Standard propuesta en [11].

3.1 Planeación

Preguntas de investigación Para realizar este estudio, se establecieron cinco preguntas de investigación (PI) mostradas en la Tabla 1 con su respectiva motivación. Estas preguntas sirvieron como guía para extraer aspectos clave del AS en la deserción escolar.

Pregunta de Investigación	Motivación
PI1. ¿Cuáles son las técnicas de AS utilizadas en la predicción de deserción de estudiantes?	Identificar las técnicas de AS más utilizadas en el ámbito de la DE en estudiantes.
PI2. ¿Cuáles son las variables utilizadas para modelar el AS en la predicción de deserción de estudiantes?	Identificar los factores más utilizados para modelar el AS en el ámbito de la DE en estudiantes.
PI3. ¿Cuáles son las métricas para validar los modelos de AS en la predicción de deserción de estudiantes?	Identificar las métricas utilizadas para validar los modelos de AS en el ámbito de la DE en estudiantes.
PI4. ¿Cuáles son las ventajas de las técnicas de AS reportadas? dad	Identificar los beneficios de las técnicas de AS reportadas en la literatura en el ámbito de la DE en estudiantes.
PI5. ¿Cuáles son los retos de las técnicas de AS reportadas?	Identificar los desafíos de las técnicas de AS reportadas en la literatura en el ámbito de la DE en estudiantes.

Tabla 1. Preguntas de investigación y sus motivaciones.

Proceso de búsqueda La estrategia de búsqueda siguió el proceso manual propuesto por Zhang [11] que determinó las bibliotecas digitales más utilizadas en las RSL para la búsqueda de estudios relevantes en IS. Los recursos editoriales clave presentados en el estudio son IEEEXplore, ACM Digital Library, ScienceDirect (Elsevier) y SpringerLink.

Además, se consideraron las plataformas Wiley Online Library y Taylor and Francis Online, debido a la naturaleza multidisciplinaria de esta RSL. Estas fuentes proveen un lugar de búsqueda confiable para la búsqueda de estudios relevantes en el ámbito del AS para la predicción de la DE.

Concepto en idioma español	Concepto en idioma inglés	Términos y sinónimos alternativos
Análisis de Supervivencia	Survival Analysis	Survival Function, Survival Model, Survival Probability, Survival Time
Deserción Estudiantil	Student Dropout	School Dropout
Retención de Estudiantes	Student Retention	—

Tabla 2. Términos de búsqueda (palabras clave).

La cadena de búsqueda se construyó basándose en la propuesta realizada por [11]. Esta propuesta comprende la combinación de un método subjetivo y un método objetivo. Esta combinación permitió la identificación de los términos de búsqueda presentados en la Tabla 2, con el propósito de construir las cadenas de búsqueda presentadas en la Tabla 3, para la búsqueda automatizada de estudios primarios relacionados con el AS aplicado en la predicción de la DE.

ID	CB	ERS	ERO	EOCB	Recall	Effort
CB1	"survival*" AND ("school dropout" OR "student dropout" OR "student retention")	6	5	1,366	83.33%	0.36%
CB2	"survival analysis" AND ("school dropout" OR "student dropout" OR "student retention")	6	5	102	83.33%	4.90%
CB3	("survival analysis" OR "survival function" OR "survival model" OR "survival probability" OR "survival time") AND ("school dropout" OR "student dropout" OR "student retention")	6	5	132	83.33%	3.78%

Tabla 3. Cadenas de búsqueda (ID: Identificador de la Cadena de Búsqueda; CB: Cadena de Búsqueda; ERS: Número de estudios relevantes seleccionados; ERO: Total de estudios relevantes obtenidos; EOCB: Total de estudios obtenidos por la cadena de búsqueda).

Para la evaluación de la cadena de búsqueda, se utilizó la técnica propuesta por [11]. Esta técnica utiliza un enfoque objetivo, que emplea un conjunto de criterios cuantitativos para valorar el rendimiento de la estrategia de búsqueda: sensibilidad (recall) y precisión (effort). La sensibilidad se define como la proporción de estudios relevantes recuperados para un tema determinado; y la precisión es la proporción de estudios recuperados que son estudios relevantes. La sensibilidad y la precisión correspondientes a la cadena de búsqueda está dada por (1) y (2):

$$(1)\ Recall = \frac{\text{Retrieved relevant studies}}{Relevant\ studies}$$

$$(2)\ Precision = \frac{\text{Retrieved relevant studies}}{Retrieved\ studies}$$

La evaluación de la cadena de búsqueda se realizó en el motor de búsqueda multidisciplinario SpringerLink, la selección del motor de búsqueda se realizó. Esta evaluación revela la competencia de las cadenas de búsqueda sugeridas para la recuperación de estudios pertinentes, basándose en los valores adquiridos para los criterios de sensibilidad (recall) y precisión (effort). Esta apreciación evidencia que las cadenas de búsqueda propuestas alcanzaron un porcentaje de 83.33% con respecto al total de ERS.

En el proceso de evaluación propuesto por [11], se sugiere un umbral en el criterio recall para la aprobación de la cadena de búsqueda, que son los valores que superan el 80%. En esta propuesta, el valor de recall adquiere mayor importancia al representar el valor de estudios relevantes obtenidos por la cadena de búsqueda. Por este motivo, la cadena de búsqueda seleccionada para esta investigación es la cadena de búsqueda CB3. Esta cadena de búsqueda demostró una efectiva recuperación del total de ERS, agregando los términos relacionados pertinentes para la investigación. Aunque la cadena de búsqueda CB2 presentó un incremento en el valor correspondiente al criterio effort en relación con la cadena de búsqueda CB3, se consideró la cadena de búsqueda CB3 debido a su adaptabilidad en los demás motores de búsqueda seleccionados.

("survival analysis" OR "survival function" OR "survival model" OR "survival probability" OR "survival time") AND ("school dropout" OR "student dropout" OR "student retention")

Extracción de datos El objetivo de esta subsección del proceso de revisión es obtener los datos necesarios de los estudios primarios para responder a las preguntas de la investigación [10]. La plantilla representada en la Tabla 4 expone los datos generales que se extraerán de los estudios primarios seleccionados.

Criterios de selección Los criterios de selección de estudios primarios se formulan con el fin de identificar aquellos estudios capaces de aportar pruebas que sean de relevancia para las preguntas de investigación [10]. Estos criterios se expresan en dos conjuntos: criterios de inclusión y criterios de exclusión mostrados en las Tablas 5 y 6 respectivamente.

Dato	Descripción
ID	Identificador único del estudio generado con la nomenclatura EP-#, que corresponde a Estudio Primario (EP) - #número.
Título	Título del estudio.
Autores	Nombres de los autores responsables de la publicación del estudio.
Año de publicación	Año de publicación del estudio primario, comprendido entre los años 2013–2025, alineándose a los criterios de inclusión.
Fuente	Dirección URL donde se encuentra el estudio en línea.

Tipo de publicación	Tipo de publicación (conferencia o journal).
Referencia o detalles de la publicación	Referencia de la publicación.
Palabras clave	Palabras clave establecidas en el estudio.
Venue	Nombre de la conferencia o journal donde fue presentado el estudio.
DOI	Identificador único del estudio publicado.
Abstract o resumen	Resumen (abstract) establecido en el estudio.
Pregunta(s) de investigación relacionada(s)	Preguntas de investigación que el estudio responde.
Técnica de AS utilizada en la predicción de DE	Técnica de Aprendizaje Supervisado (AS) utilizada en la predicción de deserción de estudiantes. (PI1).
Variable(s) utilizadas para modelar el AS en la predicción de DE	Variables utilizadas para modelar el AS en la predicción de deserción de estudiantes. (PI2).
Métrica utilizada para validar el modelo de AS en la predicción de DE	Métrica utilizada para validar el modelo de AS en la predicción de deserción de estudiantes. (PI3).
Ventaja(s) de la técnica de AS en la predicción de DE	Ventaja(s) reportadas por la técnica de AS utilizada en la predicción de deserción de estudiantes. (PI4).
Reto(s) de la técnica de AS en la predicción de DE	Reto(s) reportados por la técnica de AS utilizada en la predicción de deserción de estudiantes. (PI5).

Tabla 4. Formato de extracción de los datos.

ID	**Criterio**
CI1	El estudio está publicado entre 2013 y 2025.
CI2	El estudio está escrito en el idioma inglés.
CI3	Se tiene acceso completo al estudio.
CI4	Al leer el título y el resumen (abstract) del estudio se tienen indicios de responder al menos una pregunta de investigación.
CI5	Al leer completamente el estudio se responde al menos una pregunta de investigación.

Tabla 5. Criterios de inclusión.

ID	Criterio
CE1	El estudio está publicado como diapositiva, resumen, reporte técnico o capítulo de libro.
CE2	Al leer el título y el resumen (abstract) del estudio se identifica que el enfoque principal se encuentra fuera del ámbito de la educación.
CE3	El estudio está duplicado.

Tabla 6. Criterios de exclusión.

3.2 Ejecución

Cuatro fases fueron definidas, cada una con criterios de inclusión y exclusión específicos mostrados a continuación:

Fase 1: Se incluyeron estudios publicados entre 2013 y julio de 2025 y además escritos en el idioma inglés.

Fase 2: Se incluyeron estudios con acceso completo y se excluyeron aquellos que estuvieran publicados como presentación, resumen, reporte técnico o capítulo de libro.

Fase 3: Se incluyeron aquellos estudios que al leer el título y el resumen (abstract) del estudio se tienen indicios de responder al menos una pregunta de investigación.

Fase 4: Finalmente se incluyeron aquellos estudios que contestaran al menos una pregunta de investigación y se excluyeron los estudios duplicados.

3.3 Reporte

Se seleccionó utilizar una síntesis narrativa siguiendo los pasos presentados por [12]. En esta síntesis, se analizó los hallazgos, identificando patrones en las gráficas y tabulaciones.

4. Resultados

Como resultado de la estrategia de búsqueda y el proceso de selección, se identificaron 36 estudios primarios que respondieron a las preguntas de investigación. La síntesis narrativa se desarrolló siguiendo la metodología propuesta por Popay [12].

La fuente Taylor & Francis fue la fuente que más aportó a los estudios primarios con el 27.7% (10 estudios) seguido de Springer Link con 25% (9 estudios) y Wiley 19.4% (7 estudios). ACM DL y IEEE Xplore aportaron la misma cantidad de estudios con un 11.1% (4 estudios), seguida Science Direct con 5.55% (2 estudios). Los 36 estudios primarios se pueden encontrar en [13].

4.1 PI1. ¿Cuáles son las técnicas de Análisis de Supervivencia utilizadas en la predicción de deserción de estudiantes?

A partir de la Fig. 1, se puede observar que el modelo Cox proportional hazards model fue la técnica de AS más utilizada, con un 48.57% del total de estudios primarios que respondieron la pregunta de investigación. La segunda técnica más utilizada es el modelo de Kaplan-Meier, con un 31.42% del total de estudios primarios que respondieron la pregunta de investigación. El modelo Logistic regression model y el enfoque Discrete-time hazard model ocuparon el tercer lugar, con un 11.42% del total de estudios primarios que respondieron la pregunta de investigación, respectivamente. El modelo Multinomial logistic regression model fue la última técnica en reportar una frecuencia mayor en comparación con el resto de las técnicas que reportaron una única intervención en los

estudios primarios, con un 8.57% del total de estudios primarios que respondieron la pregunta de investigación.

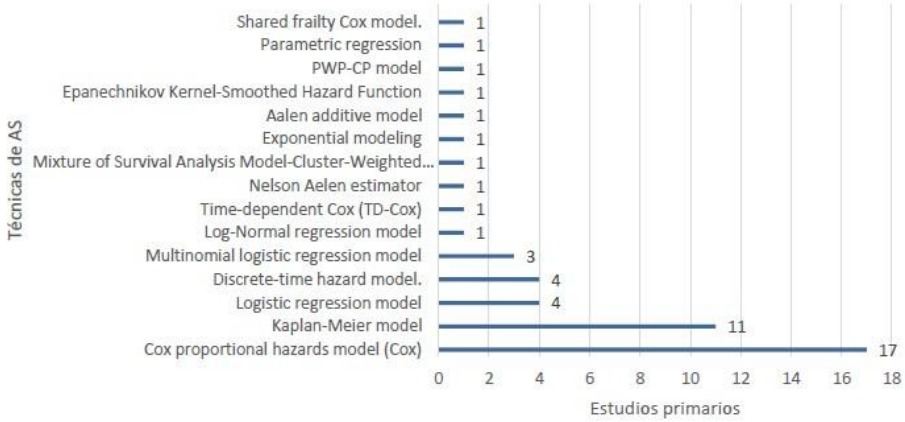


Figura 1. Distribución de estudios primarios por técnica de AS.

La razón de la preferencia en la adopción del modelo Cox proportional hazards model en el ámbito educativo es su amplio historial de uso en otros ámbitos, como lo determina [6] en su intervención en la Wayne State University, en donde afirma que, a pesar del éxito de los métodos de AS en áreas como la salud, la ingeniería, etc., solo se ha intentado utilizar estos métodos de forma limitada en el problema de la retención de estudiantes.

4.2 PI2. ¿Cuáles son las variables utilizadas para modelar el Análisis de Supervivencia en la predicción de deserción de estudiantes?

Se extrajeron un total de 29 variables clave pertenecientes a entornos universitarios, cursos de plataformas MOOC y entorno social particular. Estos atributos caracterizaron a la población estudiantil en ámbitos como los datos demográficos, atributos financieros, antecedentes académicos e información semestral.

La Figura 2 muestra las variables clave en el contexto universitario, en donde se aprecia que el género, el promedio, el puntaje de examen de admisión, la educación de los padres, la edad, el promedio de escuela secundaria o bachillerato, las bajas por semestre, el abandono de un curso y la ubicación de residencia son los factores que más influyeron en el abandono de los estudiantes de institutos de educación superior.

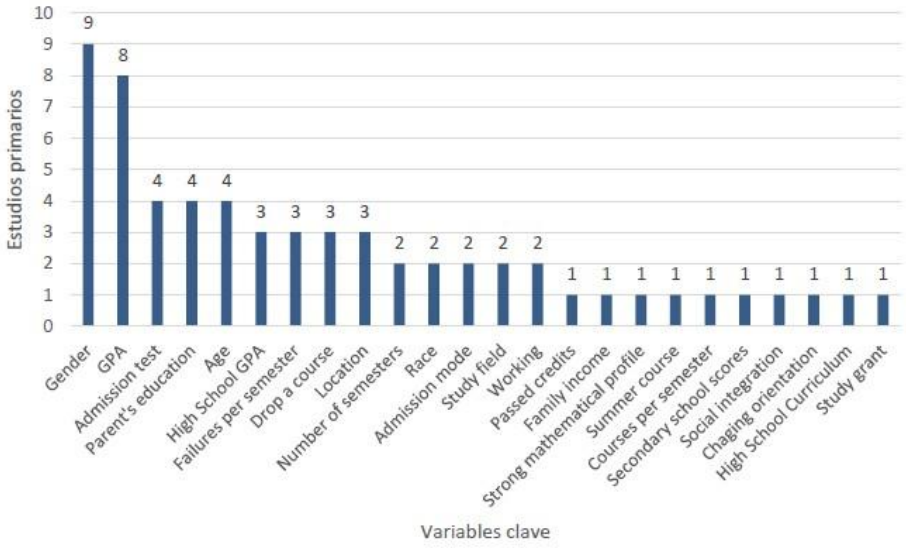


Figura 2. Variables clave en el contexto universitario.

4.3 PI3. ¿Cuáles son las métricas para validar los modelos de Análisis de Supervivencia en la predicción de deserción de estudiantes?

Del total de estudios primarios que respondieron esta pregunta de investigación, se extrajeron 16 métricas de validación de modelos de AS en la predicción de la DE. La Figura 3 muestra la distribución de estudios primarios segmentados por métrica reportada.

Las métricas más frecuentes proporcionan una visión detallada de cómo estos modelos funcionan específicamente en el contexto educativo. Entre las métricas reportadas, el Log-rank test es el más común con un 30.76% de los estudios, seguido de cerca por Recall, AUC, Akaike Information Criterion (AIC) y Concordance Index (C-index), cada uno con un 23.07%. Además, métricas como Accuracy, Precision, F-measure, y Bayesian Information Criterion (BIC) muestran una frecuencia del 15.38%.

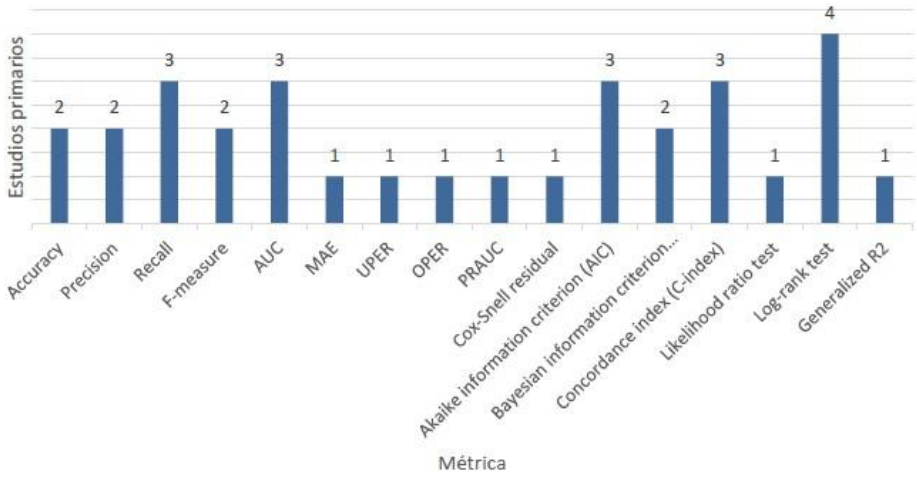


Figura 3. Distribución de estudios primarios por métrica reportada.

4.4 PI4. ¿Cuáles son las ventajas de las técnicas de Análisis de Supervivencia reportadas?

En la literatura educativa, los investigadores resaltan los beneficios de aplicar técnicas basadas en AS para obtener información valiosa sobre los estudiantes en riesgo. A continuación, la Tabla 7 presenta las ventajas generales del AS identificadas durante la revisión.

4.5 PI5. ¿Cuáles son los retos de las técnicas de Análisis de Supervivencia reportadas?

De los 36 estudios primarios,3 reportaron retos de las técnicas de AS utilizadas para la predicción de deserción de estudiantes. Asimismo, se identificaron 2 retos generales del AS aplicado en el ámbito de la DE, proporcionados por los estudios EP-1 y EP-23. Tras el análisis de los estudios primarios, se encontraron un total de 6 retos específicos de las técnicas de AS, los cuales se relacionan únicamente con 3 modelos de AS del total de 15 modelos identificados.

El primer reto general adquirido de los estudios primarios es el uso limitado del AS en el ámbito de la retención de estudiantes y de la DE. Existe una tendencia por parte de los autores en la modificación y extensión de los modelos estadísticos de AS para mejorar el ajuste a los datos. A pesar de que el modelo de AS tiene más flexibilidad para tratar el problema de la retención de estudiantes, hay pocos esfuerzos en la literatura en este ámbito de la educación.

Otro reto más específico del AS y en comparación con otros métodos de estimación utilizados en la educación, es que tiende a requerir más información, esto puede suponer un reto importante para aquellos estudios con limitada cantidad de datos disponibles. Este reto se presenta en el estudio realizado por [14], en donde se demuestra que el AS necesitó una mayor cantidad de datos en comparación con los modelos logit y probit.

Ventaja	Estudio
El AS puede incorporar dos componentes: si se produce o no un acontecimiento (en el ámbito de la DE sería el abandono) y cuándo se producirá el acontecimiento. Por lo tanto, la ventaja de utilizar el AS en lugar de regresión logística u otros métodos de minería de datos es la capacidad de añadir el componente temporal al modelo.	EP-1
En el AS, cualquier cambio en el riesgo de abandono cuando cambia una de las características del estudiante es fácil de observar.	EP-11
El AS no requiere grandes muestras de entrenamiento para lograr un aprendizaje automático que clasifique a un estudiante en un riesgo específico de abandono.	EP-11
En el AS, los datos censurados hacen referencia a datos de estudiantes que no presentaron el evento de interés (abandono) durante el período de observación. Barragán y colaboradores resaltan el beneficio del AS al incorporar información censurada.	EP-11
El AS permite evaluar cómo diferentes grupos dentro de una población difieren respecto al tiempo que toma hasta que ocurre el evento estudiado (en el ámbito educativo, el abandono). Estas diferencias pueden ser por género, área de conocimiento, y demás atributos que caractericen la población.	EP-11
El AS puede identificar cuales variables o factores tienen un impacto significativo en la probabilidad de abandonar.	EP-11

Tabla 7. Ventajas del uso de AS en la deserción escolar.

5. Discusión

Los hallazgos de esta revisión y la aplicación de estos modelos encontrados en la literatura permitirían a las autoridades educativas:

– Identificar a los estudiantes que requieren intervención temprana, brindando así apoyo oportuno antes de que se materialice el riesgo de deserción escolar.

– Implementar programas de tutoría y mentoría en colaboración con el profesorado (docentes y estudiantes).

– Reconocer a los estudiantes con buenos candidatos para becas u oportunidades de ayuda financiera.

– Revisar y adaptar el currículo académico para alinearlo mejor con las necesidades y trayectorias de aprendizaje de los estudiantes.

– Monitorear los factores no académicos que influyen en el rendimiento académico, incluyendo el nivel socioeconómico, las características sociodemográficas, las condiciones laborales, los aspectos psicológicos y otras variables contextuales.

6. Amenazas a la validez

En este estudio se consideraron únicamente aquellos motores de búsqueda de acceso disponible por medio de la institución. Sin embargo, durante el proceso de la investigación no se contó con el acceso completo a algunos motores de búsqueda, esta

limitación en la disponibilidad de algunos estudios revela la posible omisión de algún estudio relevante en la revisión.

Durante el proceso de creación de las cadenas de búsqueda consideradas para conducir la revisión, se evaluaron utilizando un Quasi-Gold Standard conformado únicamente por estudios relevantes pertenecientes al motor de búsqueda SpringerLink debido al enfoque multidisciplinario del tema de investigación. A pesar de estas limitaciones, se presentan las siguientes medidas para mitigar las posibles consecuencias de los elementos descritos anteriormente:

– Pese a la falta de acceso completo en algunas bases de datos, la selección de los motores de búsqueda se basó en el estudio realizado [11], que considera las fuentes de información más utilizadas en las revisiones sistemáticas e investigaciones relacionadas con la IS.

– Se siguió de forma sistematizada el proceso propuesto por [10] para planificar y conducir la revisión, el método para identificar estudios relevantes en IS publicado por Zhang y colaboradores [11] y se sintetizaron los resultados reportados por la literatura utilizando elementos de la síntesis narrativa diseñada por [12].

– La cadena de búsqueda CB3 se seleccionó debido a los valores reportados para los criterios Recall y Effort durante su evaluación en el motor de búsqueda SpringerLink, que cumplen con los valores aceptados en el método de evaluación propuesto por Zhang y colaboradores. Asimismo, la cadena de búsqueda se probó en cada fuente de información seleccionada para asegurar su adaptación.

– Durante las fases de planificación, conducción y documentación de la revisión propuestas por [10], se consultó la opinión de profesionales en el ámbito estadístico y de inteligencia artificial para asegurar la calidad de la revisión.

– Otra limitación importante es la ausencia de una evaluación formal de la calidad de los estudios primarios. Si bien se consideró que el proceso de revisión por pares de revistas y conferencias proporciona un nivel básico de rigor, dicho proceso no es homogéneo entre distintas sedes, y sus estándares pueden variar significativamente. Sin una evaluación sistemática de aspectos como la solidez metodológica, la validez de los conjuntos de datos, la reproducibilidad de los experimentos y la claridad en la presentación de los resultados, la robustez de las conclusiones extraídas de los estudios incluidos puede verse debilitada.

7. Conclusión

Esta investigación reporta el proceso y resultados de una Revisión sistemática de la Literatura, con el objetivo de conocer el estado actual de las técnicas de AS utilizadas en el ámbito de la predicción de estudiantes en riesgo.

Las respuestas a las preguntas de investigación revelaron la utilidad del AS en el contexto educativo. Durante la revisión se identificaron 36 estudios primarios, que reportaron un total de 15 técnicas de AS, 29 variables clave, 16 métricas y detallaron las ventajas de la aplicación del AS en la predicción de estudiantes en riesgo. Además, se identificó un área de oportunidad en la especificación de los retos que rodean al AS. La literatura ha demostrado interés en la comprobación del éxito del AS como herramienta de prevención para el problema de la DE. Sin embargo, existe una amplia área de oportunidad en la identificación de las limitaciones relacionadas a su aplicación.

Este estudio permitirá conocer el estado del arte de las técnicas de AS utilizadas en el ámbito de la DE. Brindando información sobre los propósitos de las técnicas identificadas, las ventajas y retos inherentes a estos métodos y al AS en general. Los estudios reportan la efectividad de su uso en un contexto real, beneficiando al personal de las instituciones a implantar estrategias de retención efectivas.

El trabajo futuro se centra en la recopilación de más datos que caractericen a los estudiantes, con el objetivo de generar modelos más robustos y de mayor capacidad predictiva. Asimismo, existe un área de oportunidad en la identificación de los retos inherentes en la aplicación de técnicas de AS en la DE.

Referencias

Y. Wang, Software engineering foundations: A software science perspective, Auerbach Publications, 2007.

Sommerville, Software engineering tenth edition (2016).

Cico, Lean software startup practices and software engineering education, in: Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings, 2022, pp. 281–285.

N. R. Mead, D. Garlan, M. Shaw, Half a century of software engineering education: The cmu exemplar, IEEE Software 35 (5) (2018) 25–31.

S. Bayona-Oré, Dropout in computer science, systems engineering and software engineering programs, in: World Conference on Information Systems and Technologies, Springer, 2023, pp. 592–599.

S. Ameri, M. J. Fard, R. B. Chinnam, C. K. Reddy, Survival analysis based framework for early prediction of student dropouts, in: Proceedings of the 25th ACM international on conference on information and knowledge management, 2016, pp. 903–912.

H. Shi, Y. Zhou, Stay or leave? exploring student factors associated with dropout patterns in massive open online courses, in: 2023 IEEE International Conference on Advanced Learning Technologies (ICALT), IEEE, 2023, pp. 26–30.

S. B. Plank, S. DeLuca, A. Estacion, High school dropout and the role of career and technical education: A survival analysis of surviving high school, Sociology of Education 81 (4) (2008) 345–370.

J. B. F. Gomes, M. Holanda, C. C. Koike, M. T. L. Costa, et al., Study on computer science undergraduate students dropout at the university of brasilia, in: 2023 IEEE Frontiers in Education Conference (FIE), IEEE, 2023, pp. 1–7.

B. A. Kitchenham, D. Budgen, P. Brereton, Evidence-Based Software Engineering and Systematic Reviews, Chapman and Hall/CRC, 2015. doi:10.1201/b19467.

H. Zhang, M. A. Babar, P. Tell, Identifying relevant studies in software engineering, Information and Software Technology 53 (2011) 625–637. doi:10.1016/j.infsof.2010.12.010.

J. Popay, L. Arai, M. Rodgers, N. Britten, Guidance on the conduct of narrative synthesis in systematic reviews: A product from the esrc methods programme (2006). doi:10.13140/2.1.1018.4643.

J. Sanchez-Garcia, Apéndice a: Reposirotio estudios primarios (2025). doi: 10.5281/zenodo.17518840. URL https://doi.org/10.5281/zenodo.17518840

K. Boualaphet, H. Goto, Determinants of school dropout in lao people's democratic republic: A survival analysis, Journal of International Development 32 (6) (2020) 961–975.

NOTAS BIOGRÁFICAS

Franz Jesús Rivera-Alcántara, Licenciado en Ingeniería de Software por la Universidad Veracruzana, en la Facultad de Estadística e Informática. En su trayectoria académica, ha mostrado especial interés en aplicaciones de inteligencia artificial y ciencia de datos. Colaboró en el desarrollo de una solución basada en Retrieval-Augmented Generation (RAG) en Sophinauta LTD.

Ángel Juan Sánchez-García es Licenciado en Informática, Maestro en Inteligencia Artificial, Especialista en Métodos Estadísticos y Doctor en Inteligencia Artificial por la Universidad Veracruzana, México. Actualmente es profesor en la Facultad de Estadística e Informática de la Universidad Veracruzana. Es miembro del Cuerpo académico Ingeniería y Tecnología de Software. Es miembro desde 2018 del Sistema Nacional de Investigadoras e Investigadores del CONAHCYT (actualmente Nivel 1, área 8) y cuenta con el reconocimiento del Programa de Desarrollo Profesional Docente (PRODEP) desde 2018. Su trabajo de investigación incluye las áreas de Aprendizaje automático e Inteligencia Artificial aplicada a la Ingeniería de software. Página web: www.uv.mx/personal/angesanchez.

José Juan Muñoz-León es licenciado en Matemáticas y Maestro en Gestión de Calidad por la Universidad Veracruzana. Además, es Maestro y Doctor en Didáctica de las Matemáticas y las Ciencias Experimentales por la Universidad Autónoma de Barcelona. Actualmente es profesor en la Facultad de Estadística e Informática de la Universidad Veracruzana. Responsable del grupo de investigación Data Sciencie Applications Reserch Group y con reconocimiento del Programa de Desarrollo Profesional Docente (PRODEP) hasta 2028. Su trabajo de investigación incluye áreas de analítica de datos y educación en el nivel superior.

Jorge Octavio Ocharán-Hernández obtuvo su Maestría en Ingeniería de Software y su Doctorado en Ciencias de la Computación en la Universidad Veracruzana, Veracruz, México. En Ingeniero en Tecnologías Estratégicas de la Información por la Universidad Anáhuac Veracruz. Actualmente es profesor de tiempo completo en la Facultad de Estadística e Informática de la Universidad Veracruzana. Cuenta con más de 20 años de experiencia en la industria y la formación en Ingeniería de Software. Sus intereses de investigación incluyen la ingeniería de requisitos, la arquitectura de software, el diseño de software y API, y la

aplicación de la inteligencia artificial en la ingeniería de software. Página web: https://uv.mx/personal/jocharan/.

El Dr. José Luis Sánchez-Leyva es Director General de Desarrollo Académico e Innovación Educativa de la Universidad Veracruzana, Académico de Carrera de Tiempo Completo Titular C con adscripción en la Facultad de Contaduría y Administración de la Universidad Veracruzana en la Región Coatzacoalcos-Minatitlán; es miembro del Sistema Nacional de Investigadores Nivel 1y Miembro del Padrón Veracruzano de Investigadores del Consejo Veracruzano de Investigación Científica y Desarrollo Tecnológico; cuenta con reconocimiento de Perfil Deseable del Programa para el Desarrollo Profesional Docente (PRODEP) y con la Certificación Profesional como Licenciado en Administración. En su formación profesional, es Doctor en Gobierno y Administración Pública; Maestro en Gobierno y Asuntos Públicos y Licenciado en Administración de Empresas. Además, es Responsable del Cuerpo Académico en Consolidación (UV-CA-506) “Economía del Conocimiento e Innovación” y fungió como director general del Área Académica Económico-Administrativa de 2022 a 2025 y como director de la Facultad de Contaduría y Administración en la Región Coatzacoalcos-Minatitlán de 2013 a 2018. El Dr. Sánchez Leyva es autor y coautor de artículos publicados en revistas y capítulos de libros, participa como conferencista y ponente en congresos nacionales e internacionales y es miembro de comité editoriales en diversas revistas y congresos.

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

Recibido 2 Nov 2025 *ReCIBE, Año 15 No. 1, mayo 2026*

Aceptado 22 Abri 2026

Evaluación de flexibilidad para microservicios basado en atributos de calidad

Flexibility assessment for microservices based on quality attributes

Said Misael Venoso Lara[1]
m24ce008@cenidet.tecnm.mx
Juan Carlos Rojas Pérez[1]
juan.rp@cenidet.tecnm.mx
Olivia Graciela Fragoso Díaz[1]
olivia.fd@cenidet.tecnm.mx

[1]*Centro Nacional de Investigación y Desarrollo Tecnológico Cuernavaca Morelos, México*

Resumen.
En los últimos años, la Arquitectura de Microservicios (MSA) se ha consolidado como un enfoque clave para el desarrollo de sistemas distribuidos, ofreciendo ventajas como flexibilidad, escalabilidad y agilidad. No obstante, la literatura evidencia una carencia de un esquema de calidad específico que permita evaluar atributos críticos en este contexto, particularmente la flexibilidad, ya que las métricas heredadas de otras arquitecturas no siempre resultan aplicables o carecen de validación empírica. Ante esta limitación, se presenta un esquema de evaluación de la flexibilidad, organizada en tres características principales: adaptabilidad, escalabilidad y portabilidad, cada una con subatributos específicos derivados de la ISO/IEC 25010, la ISO/IEC 9126, la ISO/IEC 2382, entre otras fuentes. Los resultados preliminares, obtenidos a partir del análisis de catorce microservicios en Java asociados a tres repositorios: Zull (47.37%), ACME Air Microservice (43.47%) y E-Commerce (33.7%) sugieren que la adopción de microservicios no garantiza por sí sola una alta flexibilidad, sino que depende del cumplimiento de prácticas de diseño adecuadas. El esquema permitió identificar debilidades recurrentes en documentación, instalabilidad y mecanismos de seguridad, lo que confirma su utilidad como herramienta para visibilizar áreas de mejora y orientar el diseño de sistemas más robustos y adaptables.

Palabras Clave: Microservicios - Atributos de calidad - Métricas de calidad - Desarrollo - Flexibilidad - Medición - Modelo de calidad - Métricas.

Abstract.
In recent years, Microservices Architecture (MSA) has established itself as a key approach to distributed systems development, offering advantages such as flexibility, scalability, and agility. However, the literature shows a lack of a specific quality framework that allows for the evaluation of critical attributes in this context, particularly flexibility, since metrics inherited from other architectures are not always applicable or lack empirical validation. Given this limitation, we present a flexibility assessment framework organized into three main characteristics: adaptability, scalability, and portability, each with specific sub-attributes derived from ISO/IEC 25010, ISO/IEC 9126, ISO/IEC 2382, among other sources. Preliminary results, obtained from the analysis of fourteen microservices in Java associated with three repositories: Zull (47.37%), ACME Air Microservice (43.47%), and E-Commerce (33.7%), suggest that the adoption of microservices alone does not guarantee high flexibility, but rather depends on compliance with appropriate design practices. The scheme allowed us to identify recurring weaknesses in documentation, installability, and security mechanisms, confirming its usefulness as a tool for highlighting areas for improvement and guiding the design of more robust and adaptable systems.

Keywords: Microservices - Quality attributes – Quality metrics – Development – Flexibility – Measurement – Quality model - Metrics.

1. Introducción

En los últimos años, la Arquitectura de Microservicios (MSA) se ha consolidado como una de las principales estrategias para el desarrollo de aplicaciones distribuidas, ofreciendo ventajas como flexibilidad, escalabilidad y agilidad [8,10,11]. Este enfoque permite descomponer sistemas monolíticos en componentes independientes, facilitando su mantenimiento y evolución. Sin embargo, a pesar de sus ventajas evidentes en escalabilidad y despliegue, la evaluación de la calidad en microservicios sigue representando un desafío técnico y metodológico relevante pues las métricas empleadas en Arquitecturas Orientadas a Objetos (OOA), Arquitecturas Orientadas a Servicios (SOA) o Computación Orientada a Servicios (SOC) no siempre pueden aplicarse directamente sobre las MSA [17, 38]. En muchos casos, las métricas propuestas para estas arquitecturas se han quedado en etapas conceptuales, carecen de validación empírica o presentan dificultades de implementación en entornos de microservicios [8, 41].

La calidad en el contexto de los microservicios no puede medirse únicamente con los criterios tradicionales aplicados a arquitecturas monolíticas u orientadas a objetos, pues la MSA introduce dimensiones adicionales relacionadas con la distribución, la comunicación entre servicios, la autonomía y la independencia del despliegue [12,13]. Autores como Dragoni et al. [12] y Soldani et al. [27] destacan que, si bien la MSA facilita la evolución del software, también incrementa la complejidad arquitectónica y operacional, lo que impacta directamente en atributos de calidad como la mantenibilidad, la seguridad y, especialmente, la flexibilidad.

De acuerdo con la ISO/IEC 25010 [2], la flexibilidad forma parte de los atributos vinculados a la capacidad de un sistema para responder a cambios en su entorno o en los requisitos. No obstante, esta definición resulta general y, por tanto, insuficiente para capturar las particularidades de los microservicios, donde la flexibilidad no solo depende del código fuente, sino también de la configuración del entorno, la documentación técnica, el desacoplamiento de servicios y la interoperabilidad. En este sentido, estudios como los de Al-Debagy y Martinek [8] o Vera-Rivera [15,16] coinciden en que el desafío no radica únicamente en definir métricas, sino en adaptarlas a las condiciones dinámicas y distribuidas de la MSA.

Autores como Berander [9] señalan que medir la calidad del software es un proceso complejo que implica seleccionar un esquema de calidad adecuado, identificar los atributos relevantes y definir métricas claras y medibles. Actualmente, dentro de la literatura se menciona que, en microservicios, existe una falta de consenso sobre qué métricas utilizar para ciertos atributos de calidad, cómo medirlas y en qué situaciones aplicarlas [15-19]. Esta ausencia de estandarización no solo genera ambigüedad en la evaluación, sino que también aumenta el riesgo de problemas como acoplamiento excesivo, baja cohesión, vulnerabilidades de seguridad y dificultades de mantenimiento. Por ello, surge la necesidad de contar con esquema de calidad específicos que permitan medir de manera objetiva dichos atributos, ofreciendo a desarrolladores y arquitectos una guía clara para garantizar que el diseño e implementación de microservicios cumpla con los estándares de calidad requeridos.

El presente trabajo propone un esquema para evaluar la flexibilidad en microservicios, abordando así la brecha de evaluación existente. El esquema se fundamenta en las normas internacionales (ISO/IEC 25010 [2], ISO/IEC 9126 [3], ISO/IEC 2382[4]) y literatura reciente. Para validar dicho esquema, se aplica un análisis preliminar sobre repositorios de prueba en Java, midiendo la flexibilidad total a través de la evaluación de sus tres subatributos propuestos por el esquema (adaptabilidad, escalabilidad, portabilidad).

Durante el análisis realizado en esta investigación de la arquitectura MSA se determinó que los desarrolladores a menudo suelen pasar a segundo plano el atributo de flexibilidad,

ya que la MSA ofrece beneficios como escalabilidad, facilidad de mantenimiento e independencia [8, 12, 37], que son características contempladas por la ISO/IEC 25010 como subatributos, y que a menudo suelen atribuirse a una arquitectura flexible. En el presente trabajo se llevó a cabo un análisis preliminar de catorce microservicios en tres repositorios para conocer el grado de flexibilidad presente en estos de acuerdo con las normas y estándares citados.

El objetivo central es proporcionar un esquema medible y adaptable que permita valorar la flexibilidad desde un enfoque empírico, brindando una herramienta útil para desarrolladores y arquitectos en la toma de decisiones relacionadas con el diseño, la refactorización y la evaluación de arquitecturas basadas en microservicios.

2. Trabajos relacionados

La evaluación de la calidad del software en arquitecturas de microservicios (MSA) continúa siendo un desafío tanto conceptual como práctico. A pesar de su adopción masiva en la industria, particularmente en sistemas de alta disponibilidad y entornos de despliegue continuo, los marcos teóricos y las métricas específicas para MSA aún están en consolidación. Las ventajas tradicionalmente atribuidas a las MSA, como la flexibilidad, la independencia de despliegue y la capacidad de escalado selectivo, han sido ampliamente documentadas [2, 23, 24]; sin embargo, los métodos cuantitativos para medir estos beneficios no han seguido el mismo ritmo de evolución.

En esta revisión de la literatura se clasifican los trabajos existentes en tres categorías principales, aplicación de métricas de arquitecturas monolíticas y orientadas a servicios (OOA y SOA), enfoques centrados en métricas de granularidad y acoplamiento, y modelos de calidad conceptuales que carecen de validación empírica.

2.1 Métricas Tradicionales (OOA y SOA)

Las métricas empleadas en OAA y SOA fueron pioneras en la cuantificación de la calidad estructural y funcional del software. Sin embargo, no son directamente aplicables a MSA. Los trabajos de la era de SOA, por ejemplo, se centraron extensamente en la cohesión de servicios y la complejidad de la orquestación [24, 37, 38, 39]. Si bien son relevantes, esas métricas fallan en capturar la naturaleza de MSA, que favorece la coreográfica sobre la orquestación y define la independencia de despliegue como un pilar central, un concepto ajeno a la mayoría de las métricas de OOA.

En el contexto de microservicios, la granularidad del despliegue y la autonomía del ciclo de vida introducen nuevas variables que no se contemplaban en las métricas tradicionales, por lo que éstas no capturan adecuadamente dependencias asíncronas o la interacción mediante colas de mensajería [15, 38, 40]. Además, la noción de interoperabilidad de seguridad o de configuración dinámica están ausente en estos modelos [39, 42].

2.2 Granularidad y Acoplamiento

La investigación más reciente sobre calidad en MSA se ha centrado casi exclusivamente en granularidad y la medición de acoplamiento, considerados como indicadores indirectos de flexibilidad y mantenibilidad. En cuanto a la granularidad autores como Vera-Rivera han propuesto dentro de sus trabajos [15, 16] modelos para definir la granularidad, pero el enfoque suele ser como definir el tamaño del microservicio, no cómo el tamaño resultante impacta en un atributo de calidad superior como flexibilidad. Para el acoplamiento [8, 14, 30, 38, 40] proponen métricas para medir el acoplamiento estructural y el acoplamiento sincrónico (llamadas HTTP directas) entre servicios. Estos trabajos son fundamentales y se alinean con nuestra investigación. Sin

embargo, estos enfoques son insuficientes. Miden indicadores de flexibilidad (como el bajo acoplamiento) pero no miden la flexibilidad en sí misma. Un sistema puede tener bajo acoplamiento y aun así ser inflexible si, por ejemplo, su instalabilidad es pobre o si carece de interoperabilidad semántica para adaptarse a nuevos requisitos comerciales.

2.3 Modelos de Calidad Integrales

La principal motivación de este trabajo es abordar la falta de un esquema integral y validado para la flexibilidad. Mientras que la norma ISO/IEC 25010 provee una base taxonómica, su aplicación directa es demasiado genérica. Otros autores han propuesto modelos de calidad para MSA, pero tienden a permanecer a nivel conceptual o se enfocan en otros atributos [35, 36]. Sin embargo, estos modelos suelen quedarse en el plano conceptual y carecen de validación empírica en entornos de producción. Adicionalmente, pocas propuestas han considerado la flexibilidad como un atributo autónomo de calidad, tratándola más bien como una consecuencia indirecta de otros factores.

A la fecha de realización de este trabajo, no se han encontrado trabajos que propongan un método de evaluación para la flexibilidad utilizando una descomposición jerárquica como la que se propone en este trabajo y que esté soportada en múltiples estándares ISO. El esquema propuesto permite evaluar la flexibilidad en la arquitectura MSA a través de descomponer la flexibilidad en subatributos medibles (adaptabilidad, escalabilidad y portabilidad), y establece un método de ponderación para su evaluación. El esquema aquí presentado fue validado mediante la aplicación a proyectos de código abierto, permitiendo una interpretación cuantitativa y replicable del atributo de flexibilidad en MSA.

3. Descripción del esquema

Para el desarrollo del esquema de calidad es importante definir el objeto a medir, de modo que la definición permita identificar las propiedades a evaluar. La ISO/IEC 25010 define la flexibilidad como “La capacidad del producto para adaptarse a cambios en sus requisitos, contextos de uso o entorno del sistema” [2], este término resulta ser bastante general, por lo que podría aplicar a ciertos modelos de arquitecturas de software, pero no precisamente en todas.

De acuerdo con Martin Fowler [1] el término Microservicio: “Es un enfoque para desarrollar una sola aplicación como un conjunto de pequeños servicios, cada uno ejecutándose en su propio proceso y comunicándose con mecanismos ligeros, a menudo una API de recurso HTTP. Estos servicios están construidos alrededor de requisitos comerciales y son desplegables de manera independiente por maquinaria de despliegue completamente automatizada”, donde se enfatiza la independencia, el desacoplamiento y la automatización del despliegue. Este contraste entre la definición de microservicio propuesta por Martín Fowler y flexibilidad según la ISO genera una problemática ya que no se refleja lo que la norma describe como flexibilidad y por ende no refleja adecuadamente los desafíos específicos de MSA, como la interoperabilidad entre servicios, la gestión de dependencias o la capacidad de escalar componentes de forma autónoma. A partir de dichos problemas, se seleccionó como propiedad de interés la flexibilidad, la cual en la ISO/IEC 25010 se compone de los subatributos de adaptabilidad, escalabilidad, instalabilidad y reemplazabilidad. No obstante, para su aplicación en entornos de microservicios fue necesario sintetizar ambas visiones, ya que, para este trabajo, la flexibilidad para MSA no es un atributo monolítico, sino una cualidad emergente que resulta del equilibrio de tres pilares arquitectónicos, como lo son: la capacidad de adaptarse a nuevas reglas de negocio (adaptabilidad), la capacidad de crecer bajo demanda (escalabilidad) y la capacidad de moverse a nuevos entornos (portabilidad).

Para el desarrollo del esquema se analizaron diversas normas que pudieran ofrecer un mayor entendimiento del concepto de flexibilidad, donde la fuente principal es la ISO/IEC 25010; sin embargo, se realizó una adaptación de los subatributos de instalabilidad y reemplazabilidad provenientes de dicha norma. En este proceso se incorporó el concepto de portabilidad, retomado de la ISO/IEC 9126, ubicándolo en un nivel jerárquico superior a los mencionados subatributos. Esta inclusión responde a la necesidad de reflejar de manera más amplia la capacidad del sistema para operar en distintos entornos tecnológicos, característica esencial en los sistemas basados en microservicios, donde los componentes deben poder desplegarse, migrarse o reemplazarse de forma ágil entre plataformas, contenedores o infraestructuras heterogéneas [10, 16, 23]. De este modo, la portabilidad actúa como un elemento integrador que abarca tanto la instalabilidad como la reemplazabilidad, ofreciendo una visión más coherente y representativa del comportamiento flexible que se espera en arquitecturas distribuidas y dinámicas.

El resultado del esquema de calidad propuesto para la flexibilidad se muestra en la *Figura 1*, donde la flexibilidad se organiza en tres características principales con sus respectivos subatributos y la ponderación para estos. Estas características constituyen la base del esquema de evaluación propuesto, el cual utiliza métricas seleccionadas asociadas a los microservicios, la claridad de sus definiciones y la viabilidad de su cálculo.

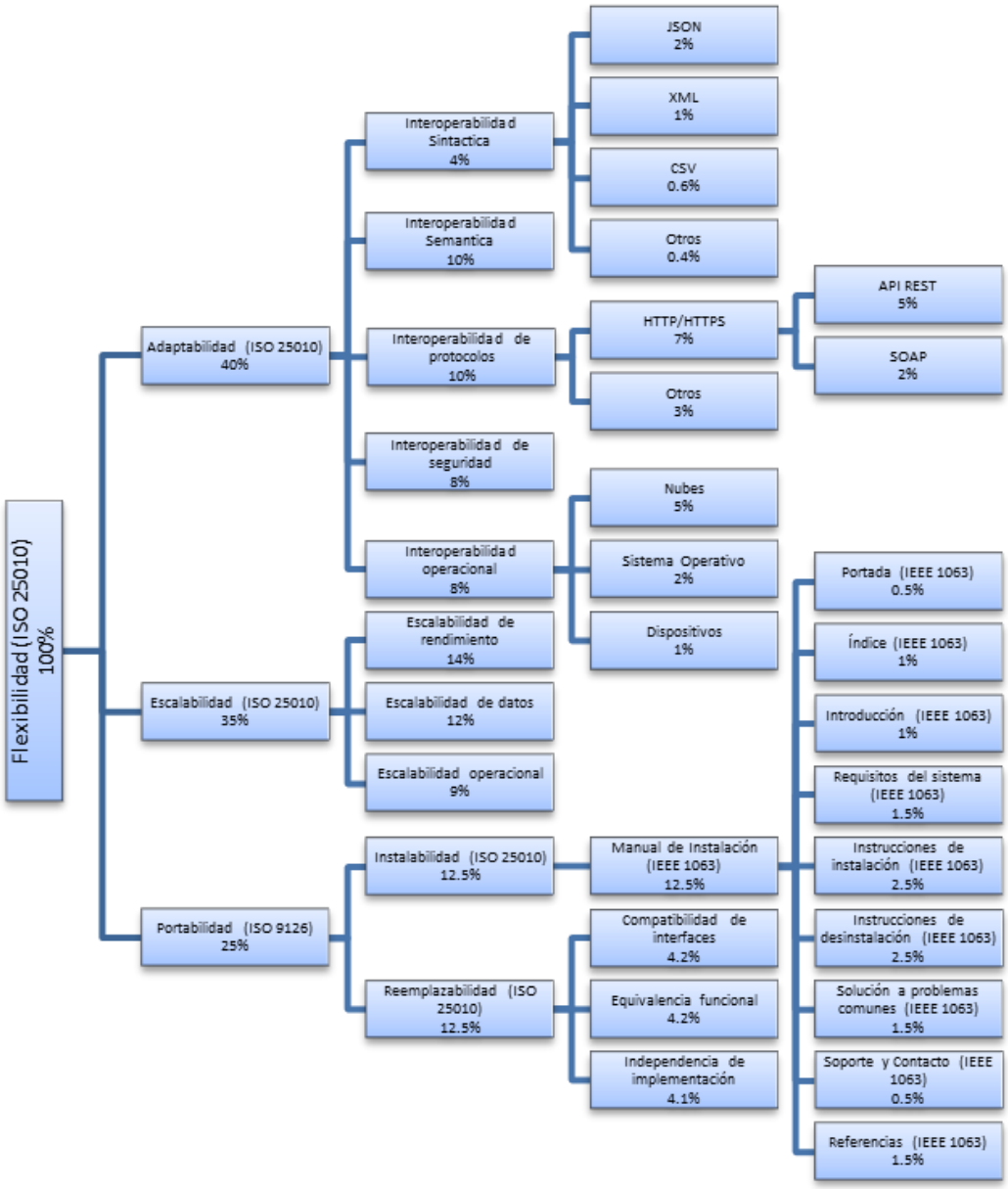


Figura 1 Esquema de calidad para flexibilidad

Es importante mencionar que las ponderaciones asignadas a cada subatributo se determinaron mediante un análisis empírico sustentado en la revisión conceptual de la literatura. Si bien no se aplicó un método formal de consenso de expertos, las proporciones se definieron con base en la relevancia observada en estudios previos y en la influencia práctica que cada subatributo ejerce sobre la flexibilidad [2, 3, 16, 24], así como el comportamiento del sistema [14, 24, 37, 40]. Esta asignación de ponderaciones permitió establecer una valoración cuantitativa/cualitativa orientada a reflejar la importancia relativa de cada elemento dentro del esquema propuesto.

3.1 Adaptabilidad [2].

La adaptabilidad recibe el mayor peso (40%) debido a que representa la agilidad del negocio. En MSA una de las principales características es que los microservicios se diseñan para operar en entornos cambiantes, con distintas tecnologías y requisitos. Su capacidad de interoperar con otros sistemas es crucial.

En MSA la interoperabilidad se convierte en un pilar fundamental de la adaptabilidad, ya que, si bien la norma ISO/IEC 25010 la sitúa como un subatributo de la Compatibilidad, la propia naturaleza de los microservicios obliga a reevaluar esta relación. En MSA, un sistema es funcionalmente inadaptable si sus componentes no pueden conectarse, entenderse y cooperar eficazmente [24, 40].

Por lo tanto, la "Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada" [2] no es exclusivamente un asunto de compatibilidad, sino un requisito para que la adaptación sea posible. Es por ello que se subdivide la adaptabilidad de la siguiente manera:

Interoperabilidad Sintáctica: Peso de 4% porque se limita a la compatibilidad de formatos de datos.

Interoperabilidad Semántica: Peso de 10%, ya que la correcta interpretación del significado de los datos es crítica para la integración efectiva.

Interoperabilidad de protocolos: Se le otorga el mismo peso que la semántica porque elegir y soportar protocolos adecuados define la capacidad de comunicación.

Interoperabilidad de Seguridad: Peso de 8%, ya que sin un mecanismo de autenticación y autorización la interoperabilidad sería riesgosa.

Interoperabilidad Operacional: Mismo peso que seguridad, ya que asegura la integración en procesos de despliegue y ejecución, garantizando continuidad de negocio.

3.2 Escalabilidad [2].

La escalabilidad recibe un peso del 35% porque es la respuesta de la arquitectura a la flexibilidad operativa. Mientras la adaptabilidad se centra en el cambio de la lógica de negocio, la escalabilidad se centra en la capacidad del sistema para reaccionar a cambios en la demanda (carga de trabajo) sin degradar el servicio.

En MSA la escalabilidad granular es la capacidad de escalar únicamente el servicio que experimenta la alta demanda de forma independiente [16]. Si un servicio no puede escalar de forma aislada, provoca un fallo en cascada o requiere el escalado de componentes no relacionados, tornando la arquitectura rígida e inflexible [43], sin embargo, no siempre es el primer obstáculo en la integración de sistemas y está estrechamente ligada a una capacidad empresarial definida que será distinta para cada usuario.

Esta promesa, sin embargo, es compleja de alcanzar. Un sistema no es escalable simplemente porque está dividido en microservicios. La verdadera escalabilidad granular es una cualidad emergente que depende de tres subatributos: escalabilidad de Rendimiento para el servicio, escalabilidad de sus datos, y la escalabilidad Operacional para la automatización de la infraestructura [8, 15, 17].

Por esta razón, nuestro esquema subdivide la escalabilidad en los siguientes subatributos:

Rendimiento: Recibe peso de 14% pues la capacidad de responder a más usuarios o procesos es el núcleo de la escalabilidad.

Datos: Se asigna un peso de 12% porque el manejo eficiente de grandes volúmenes de datos es crítico en sistemas distribuidos.

Operacional: Peso de 9% porque está más enfocado en aspectos de gestión y operación, pero dependen en gran medida de la infraestructura.

3.3 Portabilidad [3].

La portabilidad se pondera con 25% ya que una de las características clave de MSA es la capacidad de construir una vez y ejecutar en cualquier lugar. Un sistema es inflexible si está sujeto a un proveedor de nube especifico, a un sistema operativo propietario o a

una configuración de hardware particular. La portabilidad mide la libertad de la arquitectura para moverse, evolucionar y ser desplegada en diversos entornos, lo cual es vital, ya que resulta fundamental para trasladar el sistema a diferentes entornos sin grandes esfuerzos para la estrategia de negocio a largo plazo, aunque su impacto es menor que adaptabilidad y escalabilidad.

Según la ISO/IEC 9126, la portabilidad se define como "Conjunto de atributos relacionados con la capacidad de un sistema de software para ser transferido y adaptado desde una plataforma a otra" [3], razón por la cual, se considera que los atributos de instalabilidad y reemplazabilidad quedan en un nivel de jerarquía inferior dentro del esquema, como se presenta a continuación:

Instalabilidad [2]: Se asigna el peso de 12.5% ya que la facilidad de instalación incide directamente en la aceptación de un sistema**.**

Manual de instalación [7]: Se hereda el mismo peso ya que se considera que para que un sistema tenga una buena instalabilidad, está sujeto al manual de usuario.

Portada: Con un peso de 0.5% ya que, aunque el estándar de la IEEE 1063 lo contempla, no se considera de vital importancia

Índice: Un peso de 1% ya que el tener un índice aporta a que el usuario pueda encontrar más fácilmente lo que busca

Introducción: Igual peso que el índice ya que con él, el usuario podrá entender el contenido del documento.

Requisitos del sistema: Con un peso mayor de 1.5% ya que resulta de vital importancia conocer los requisitos mínimos para que el sistema opere de la mejor manera posible dentro de nuestro entorno

Instrucciones de instalación: Con un peso mayor de 2.5% ya que es parte fundamental saber el paso a paso de la instalación del sistema.

Instrucciones de desinstalación: Igual peso que el anterior, donde resulta importante conocer la mejor manera para desinstalar el sistema sin tener repercusiones posteriores derivadas de una mala desinstalación.

Solución a problemas comunes: Con un peso de 1.5% ya que optimiza la búsqueda de soluciones a los principales problemas que puedan surgir antes, durante y después del uso del sistema.

Soporte y contacto: 0.5% de peso ya que el principal motivo del documento es que el usuario pueda encontrar la solución e información necesarias sin la ayuda del fabricante.

Referencias: 1.5% de peso ya que resulta de vital importancia que el usuario pueda encontrar las referencias a los componentes del sistema de los que se refiere en el documento.

Reemplazabilidad [2]: Se asigna el peso de 12.5% porque sustituir un componente sin afectar el resto es clave en los microservicios.

Compatibilidad de interfaces: Peso de 4.2% porque una interfaz estable es la base de la sustitución.

Equivalencia funcional: Igual peso que el anterior porque la funcionalidad consistente garantiza que la sustitución no afecte el negocio.

Independencia de implementación: Un peso menor como lo es 4.1% ya que depende más de las buenas prácticas de desacoplamiento.

3.4 Ventajas del esquema propuesto.

A continuación, se listan ventajas observadas en el esquema de calidad propuesto.

El esquema propuesto cuenta con un sólido respaldo normativo, al basarse en los estándares internacionales, ampliamente utilizados como marco de referencia para la evaluación de la calidad del software. [2-7, 25]

Diversos estudios señalan que la calidad en sistemas basados en microservicios depende no solo de factores tecnológicos, sino también de elementos organizativos y de documentación [12, 26], por lo que el esquema propuesto integra dichas dimensiones para ofrecer una evaluación más completa.

La naturaleza modular de los microservicios permite adaptar las métricas o atributos de calidad a diferentes niveles de análisis [27, 28], por lo que el esquema propuesto puede aplicarse de manera parcial o integral según las necesidades de quien lo evalúe.

La flexibilidad constituye un atributo esencial en los entornos de microservicios, dado que estos requieren adaptarse a entornos cambiantes y despliegues distribuidos [30, 31]. Por ello, el esquema centra su análisis en este atributo como componente clave de la calidad del sistema.

La posibilidad de ajustar los pesos de los subatributos según el contexto del sistema es consistente con los enfoques modernos de evaluación de calidad que promueven la adaptabilidad del esquema a las prioridades del usuario [29].

4. Experimentación y resultados

El proceso experimental tuvo como propósito validar la aplicabilidad del esquema propuesto en repositorios de acceso libre de microservicios desarrollados en Java [20-22], ya que esta tecnología es ampliamente utilizada tanto en entornos empresariales como académicos, y cuenta con una estrecha compatibilidad con las MSA. De acuerdo con el State of Spring Survey 2024 [32], los microservicios constituyen uno de los principales focos dentro del ecosistema Spring, reflejando su relevancia en el desarrollo moderno de software distribuido. Asimismo, el estudio de Sõnajalg [33] reporta que el 86 % de los desarrolladores Java encuestados utilizan Spring Boot para la creación de microservicios, consolidándolo como el framework predominante en este ámbito. No obstante, aunque Spring Boot será considerado como el principal entorno de referencia dentro del análisis, no se establecerá como un criterio de exclusión, permitiendo así la inclusión de microservicios desarrollados con otros frameworks o librerías compatibles con Java. Finalmente, restringir el análisis a una misma plataforma tecnológica permitió garantizar la homogeneidad del entorno de evaluación, reduciendo posibles variaciones derivadas de diferencias en lenguajes o frameworks y facilitando la comparación de los resultados obtenidos.

Para el proceso de experimentación se seleccionaron tres repositorios públicos (Zull [20], ACME Air Microservice [21] y E-Commerce [22]), que en conjunto incluyen catorce microservicios con distintos niveles de complejidad, documentación y modularidad, los cuales se presentan a detalle dentro de la *Tabla 1*.

El análisis se centró en observar la presencia de prácticas y artefactos que evidencien la existencia de los subatributos definidos (adaptabilidad, escalabilidad y portabilidad). La revisión se realizó de manera iterativa y manual, complementada con herramientas de apoyo como DeepWiki [44], con el fin de localizar evidencias de interoperabilidad, rendimiento, portabilidad o documentación técnica.

Nombre	Autor	No. De Microservicios	Función
ZULL [20]	Lindsey Reynolds	4	Proporciona capacidades de enrutamiento dinámico, monitorización, resistencia, seguridad y mucho más.
Acme AIR-Microservice [21]	Joe McClure	4	Un controlador de carga de trabajo para la aplicación de muestras de Air Acme.
E-Commerce [22]	Rainbow Forest	6	Implementación de un Microservicio REST en un ECommerce con Spring boot, Cloud y múltiples módulos.

Tabla 1. Repositorios seleccionados

Con el motivo de garantizar validez de los resultados entre los proyectos analizados, se establecieron criterios de exclusión específicos. En primer lugar, se descartaron los repositorios cuyo código fuente no estuviera implementado íntegramente en Java, dado que esta tecnología representa uno de los entornos más maduros y ampliamente adoptados para la implementación de arquitecturas de microservicios [32, 33].

Aunque Spring Boot fue el framework predominante en los proyectos seleccionados, no se limitó la inclusión exclusivamente a este; se admitieron otras herramientas o librerías siempre que mantuvieran compatibilidad con el ecosistema de Java y con principios arquitectónicos equivalentes. Esta decisión se alinea con lo señalado en [12, 27] donde se subraya la necesidad de conservar consistencia tecnológica al evaluar sistemas distribuidos, sin restringir la diversidad de implementaciones.

Así mismo, se excluyeron los repositorios que no evidenciaban una estructura claramente orientada a microservicios, por ejemplo, aplicaciones monolíticas o hibridas, siguiendo con las recomendaciones propuestas en [34] ya que la mezcla de estilos arquitectónicos impacta directamente en la solidez de cualquier evaluación de calidad.

Por último, se excluyeron los repositorios con un número reducido de componentes o microservicios (menor a dos), ya que no permitían una evaluación representativa de los atributos de calidad definidos en el esquema propuesto.

Para realizar las pruebas se realizó una revisión directa al código fuente, con el fin de identificar evidencias que permitieran aplicar el esquema de calidad propuesto. Este procedimiento no se basó en un protocolo formal documentado, la revisión se realizó de forma iterativa sobre los repositorios seleccionados, verificando la presencia de prácticas o patrones asociados a los indicadores definidos en el esquema. Esta aproximación permitió mantener coherencia metodológica, priorizando la interpretación contextual del código y la arquitectura frente a la simple medición automática.

El servició de DeepWiki [44] se utilizó como recurso complementario con el propósito de agilizar la localización de segmentos de código, documentación técnica y descripciones arquitectónicas relevantes. Dicha herramienta, basada en inteligencia artificial y entrenada sobre proyectos disponibles en GitHub, facilitó la obtención de información preliminar sobre la estructura y funcionamiento de los microservicios. No obstante, toda la información recuperada a través de DeepWiki [44] fue contrastada manualmente con el contenido real de los repositorios, asegurando la verificación y validez de los hallazgos.

Con el fin de ofrecer una visión más profunda del comportamiento observado en cada proyecto y analizar de forma diferenciada las evidencias que contribuyen a la flexibilidad, a continuación, se presenta un análisis individual para cada repositorio.

4.1 Repositorio Zull

El repositorio Zull desarrollado por Netflix [20], se caracteriza por ofrecer capacidades avanzadas de enrutamiento dinámico, supervisión, resiliencia, seguridad y mucho más para aplicaciones basadas en la nube. Actúa como un servicio periférico que enruta y filtra el tráfico hacia los servicios backend, funcionando eficazmente como puerta de entrada a todas las solicitudes de dispositivos y sitios web al backend del ecosistema de aplicaciones de streamming de Netflix.

En la *Tabla 2* se presentan los resultados obtenidos por cada uno de los microservicios que componen el repositorio Zull [20], así mismo se presentan las calificaciones obtenidas por cada una de las tres características principales que según el esquema propuesto, caracterizan a la flexibilidad.

REPOSITORIO	**MICROSERVICIO**	**ADAPTABILIDAD**	**ESCALABILIDAD**	**PORTABILIDAD**	**FLEXIBILIDAD**
ZULL [20]	Zull-Core	30%	24%	8.6%	62.6%
	Zull-Discovery	20%	18%	8.6%	46.6%
	Zull-Processor	10%	9%	13.4%	32.4%
	Zull-Integration-Test	21%	17%	9.9%	47.9%

Tabla 2. Resultados del repositorio Zull [20] por microservicio

Los resultados obtenidos para el repositorio Zull evidencian fortalezas en adaptabilidad y escalabilidad (valores promedio ≈ 20.25% y 17% respectivamente), pero estas cifras están significativamente por debajo de los objetivos fijados por el esquema (40 % y 35 %). Además, la portabilidad, que tiene un peso en la evaluación de 25%, se sitúa en torno al 10 %; esto significa que los microservicios de Zull alcanzan aproximadamente el 51 % del objetivo en adaptabilidad (20.25/40 ≈ 50.6%), el 48.57% en escalabilidad (17/35 = 48.57 %) y solo el 40 % en portabilidad (10/25 ≈ 40%).

La consecuencia práctica es que, aunque Zull cumple bien con funciones propias de un *API Gateway* (enrutamiento, filtros, resiliencia), la flexibilidad global queda severamente limitada por la suma de déficits en los subatributos del esquema. Esto evidencia la necesidad de no conformarse con "buenas prácticas aisladas" sino de implementar mejoras transversales (automatización del despliegue, abstracción de dependencias del ecosistema Netflix OSS, documentación operativa y estrategias de migración) para aproximarse a los valores objetivo y, por ende, elevar la flexibilidad total del sistema.

Estos resultados también sugieren que, a pesar de la fortaleza del proyecto en los aspectos de comunicación y gestión de tráfico, su despliegue depende en gran medida de configuraciones propias del ecosistema Spring y de componentes específicos de Netflix OSS, lo cual reduce su independencia tecnológica y limita la posibilidad de trasladar el sistema a otros entornos. Esta limitación en portabilidad se convierte en el factor crítico que modera el nivel total de flexibilidad, confirmando lo señalado por Dragoni et al. [12]

y Li et al. [10] respecto a la necesidad de equilibrar la adaptabilidad con la capacidad de despliegue en entornos heterogéneos.

En cuanto a la adaptabilidad, se observó un uso extensivo de configuraciones en formato YAML y anotaciones Java que permiten parametrizar el comportamiento de los servicios, lo cual incrementa la interoperabilidad sintáctica y semántica. No obstante, la ausencia de mecanismos formales de autenticación y autorización limita la interoperabilidad de seguridad y, con ello, la flexibilidad integral del sistema. Este hallazgo es consistente con lo planteado por Taibi y Lenarduzzi [26], quienes destacan que la falta de políticas unificadas de seguridad y documentación puede afectar negativamente la mantenibilidad y la calidad global en sistemas distribuidos.

Desde una perspectiva comparativa, el microservicio Zull-Core obtuvo la puntuación más alta de flexibilidad (62.6 %), mientras que Zull-Processor presentó la más baja (32.4 %). Esta diferencia evidencia que los microservicios con mayor exposición al flujo de comunicación y gestión de peticiones suelen mostrar mayor madurez arquitectónica, mientras que los componentes de soporte tienden a reflejar un menor grado de desacoplamiento y portabilidad. En síntesis, el análisis del repositorio Zull confirma que la carencia de estrategias de portabilidad y documentación operativa limita la flexibilidad total, aun cuando la arquitectura exhibe una excelente adaptabilidad y una escalabilidad aceptable.

4.2 Repositorio ACME Air Microservice

El proyecto [21] muestra un patrón de arquitectura de microservicios con cuatro servicios independientes que gestionan la autenticación, la gestión de clientes, la información de vuelos y las operaciones de reserva lo cual permite estudiar la escalabilidad y adaptabilidad bajo escenarios de alta concurrencia.

Los resultados presentados en la *Tabla 3* evidencian que, aunque existen elementos de diseño modular (adaptabilidad ≈ 22%) y una escalabilidad moderada (≈ 9.75%), estos valores están consistentemente por debajo de los objetivos del esquema (adaptabilidad = 40 %, escalabilidad = 35 %, portabilidad = 25 %). Tomando como ejemplo el BookingService (adaptabilidad 26 %, escalabilidad 15 %, portabilidad 12.6 %), se observa que alcanza aproximadamente el 65 % del objetivo en adaptabilidad (26/40), el 42.85 % en escalabilidad (15/35) y solo el 50.4% en portabilidad (12.6/25).

REPOSITORIO	**MICROSERVICIO**	**ADAPTABILIDAD**	**ESCALABILIDAD**	**PORTABILIDAD**	**FLEXIBILIDAD**
ACME AIR MICROSERVICE [21]	AuthService	18%	8%	8.7%	34.7%
	CustomerService	22%	8%	12.8%	42.8%
	FlightService	22%	8%	12.8%	42.8%
	BookingService	26%	15%	12.6%	53.6%

Tabla 3. Resultados del repositorio ACME Air Microservice [21] por microservicio

Este patrón revela que, aunque algunos servicios (p.ej. BookingService) presentan mayor madurez funcional, ninguno alcanza el conjunto de metas mínimas esperadas; por tanto, la flexibilidad global queda restringida por déficits acumulados en más de un subatributo. La baja portabilidad, además de limitaciones en pruebas de rendimiento y automatización de despliegues, sugiere que la arquitectura necesita intervenciones integradas: estandarización de APIs (OpenAPI), generación de artefactos de despliegue

reproducibles (Dockerfiles, Helm charts), pipelines CI/CD y documentación de dependencias para cerrar las brechas hacia los objetivos planteados.

En conjunto, el análisis del repositorio Acme Air Microservice demuestra que, aunque la arquitectura distribuida logra mantener una adaptabilidad aceptable, la baja portabilidad se erige como el principal obstáculo para alcanzar un nivel de flexibilidad óptimo. Esto refuerza la necesidad de considerar la portabilidad como un factor determinante dentro del esquema de evaluación y de implementar estrategias orientadas a la automatización, documentación y despliegue multiplataforma.

4.3 Repositorio E-Commerce

Este repositorio [22] agrupa un conjunto de microservicios desarrollados en Spring Boot y Spring Cloud, los cuales se enfocan en la gestión de usuarios, pedidos y catálogo de productos. Dentro de la *Tabla 4* se presentan los resultados obtenidos tras aplicar el esquema de evaluación a cada microservicio.

REPOSITORIO	**MICROSERVICIO**	**ADAPTABILIDAD**	**ESCALABILIDAD**	**PORTABILIDAD**	**FLEXIBILIDAD**
E-COMMERCE [22]	Api-Gateway	14%	9%	9%	32%
	Eureka-Server	17%	9%	11.1%	37.1%
	User-Service	13%	9%	9.9%	31.9%
	Product-Catalog-Service	15%	9%	8.2%	32.2%
	Order-Service	16%	13%	9.7%	38.7%
	Product-Recommendation-Service	12%	9%	9.5%	30.5%

Tabla 4. Resultados del repositorio E-Commerce [22] por microservicio

Los resultados pertenecientes al repositorio E-Commerce [22] reflejan un comportamiento generalizado de baja flexibilidad, con valores que oscilan entre 30.5 % y 38.7 %. Este conjunto de microservicios evidencia deficiencias simultáneas en los tres subatributos principales del esquema: adaptabilidad, escalabilidad y portabilidad. En promedio, los valores registrados (≈ 14.5 %, 9.7 % y 9.6 % respectivamente) se sitúan muy por debajo de los objetivos establecidos (40 %, 35 % y 25 %). Esto significa que, de forma aproximada, los servicios del repositorio alcanzan apenas el 36.25% del ideal en adaptabilidad, el 27.71% en escalabilidad y el 38.4% en portabilidad, lo que explica las bajas puntuaciones globales de flexibilidad.

Como es posible denotar, existe una diferencia considerable entre el resultado obtenido tras la evaluación y la ponderación propuesta en el esquema, lo cual sugiere que, aunque el sistema resulta funcional, éste presenta un diseño orientado a la operatividad básica y modular, dejando de lado la calidad arquitectónica y la sostenibilidad. Las causas principales de este comportamiento se asocian a tres factores: (1) acoplamiento elevado entre servicios por la dependencia del servidor de descubrimiento Eureka; (2) ausencia de documentación técnica y manuales de despliegue, lo cual limita la portabilidad e instalabilidad; y (3) falta de mecanismos automatizados de escalamiento y monitoreo de

rendimiento, que restringen la capacidad del sistema para ajustarse a variaciones en la carga de trabajo.

En particular, el microservicio Order-Service obtuvo la puntuación más alta (38.7 %), mostrando una ligera ventaja en escalabilidad (13 %) y adaptabilidad (16 %), lo cual refleja su rol como componente intermedio que coordina transacciones entre clientes y catálogo de productos. Sin embargo, incluso en este caso, los valores alcanzan apenas un valor cercano al 40% del ideal, lo que denota un déficit estructural generalizado. En contraste, Product-Recommendation-Service, con 30.5 %, presentó la menor flexibilidad, evidenciando limitaciones severas en adaptabilidad (12 %) y portabilidad (9.5 %), relacionadas con la escasa documentación de dependencias y la falta de mecanismos de integración semántica con otros módulos del sistema.

El bajo desempeño en portabilidad, particularmente relevante por su ponderación de 35 %, constituye el principal factor que reduce la flexibilidad total del repositorio. Tras el análisis del código estático se denota que la falta de scripts de despliegue, la configuración estática y la inexistencia de contenedores reproducibles hacen que la migración o ejecución del sistema en diferentes entornos requiera intervención manual, afectando directamente su capacidad de adaptación. Así mismo, esta carencia se ve agravada por el hecho de que los valores de adaptabilidad y escalabilidad también están lejos del ideal, de modo que la pérdida de flexibilidad no proviene de un solo atributo, sino de una combinación de brechas acumuladas en los tres.

En la *Tabla 5* se presentan los resultados de manera específica en cuanto al microservicio *Product-Recommendation-Service* tras aplicar el esquema de calidad propuesto para evaluar la flexibilidad.

Atributos por medir				Elemento		Valor total por subatributo %	Valor total por atributo %	Valor total por atributo global
Flexibilidad	Adaptabilidad	Interoperabilidad Sintáctica		JSON (Mayor prioridad)		2	2	12
				XML (Prioridad media)		0		
				CSV (Prioridad baja)		0		
				Otros		0		
		Interoperabilidad Semántica				3	3	
		Interoperabilidad de Protocolos	HTTP/HTTPS	API REST (Mayor prioridad)		3	4	
				SOAP (Prioridad baja)		0		
			Otros			1		
		Interoperabilidad de Seguridad				0	0	
		Interoperabilidad Operacional		Nubes (Mayor prioridad)		2	3	
				Sistema Operativo (Prioridad media)		1		
				Dispositivos (Prioridad baja)		0		
	Escalabilidad	Capacidad Empresarial		Escalabilidad de Rendimiento		4	9	9
				Escalabilidad de Datos		3		
				Escalabilidad Operacional		2		
	Portabilidad	Instalabilidad		Manual de instalación	Portada	0.3	3.3	9.5
					Índice	0.4		
					Introducción	0.5		
					Requisitos del sistema	1.2		

				Instrucciones de instalación	0.8		
				Instrucciones de desinstalación	0		
				Solución a problemas comunes	0		
				Soporte y contacto	0		
				Referencias	0.1		
		Reemplazabilidad	Compatibilidad de interfaces		1.8	6.2	
			Equivalencia funcional		2.1		
			Independencia de implementación		2.3		

Tabla 5. Resultados de flexibilidad del microservicio product-recommendation-service del repositorio E-Commerce [22]

En síntesis, los resultados del repositorio E-Commerce confirman la tendencia observada en los demás proyectos: las tres dimensiones de flexibilidad presentan déficits simultáneos respecto a los valores objetivo (40% para adaptabilidad, 35% para escalabilidad y 25% para portabilidad), con la portabilidad como el factor más crítico. La suma de estas brechas explica la baja flexibilidad global e ilustra la necesidad de adoptar un enfoque integral que considere la interacción entre adaptabilidad, escalabilidad y portabilidad como elementos interdependientes dentro del modelo de evaluación propuesto.

En conjunto, los resultados obtenidos de los tres repositorios evaluados (Zull, Acme Air Microservice y E-Commerce) evidencian una tendencia sistemática de baja flexibilidad en los proyectos analizados. Si bien cada repositorio presenta características particulares, se observa un patrón común: los valores alcanzados para los subatributos de adaptabilidad, escalabilidad y portabilidad se mantienen por debajo de los niveles objetivo, definidos en el esquema (40%, 35% y 25%, respectivamente). Esta situación prueba que las implementaciones de microservicios en entornos Java, incluso aquellas con arquitecturas maduras y un uso extensivo de frameworks consolidados como Spring Boot o Netflix OSS, no logran cubrir integralmente los aspectos de calidad asociados a la flexibilidad.

La *Figura 2* muestra que, en términos relativos, la adaptabilidad presenta en promedio los valores más estables ya que es el subatributo que mejor valuado está, lo cual indica que los proyectos tienden a mantener cierta coherencia modular y semántica entre

servicios. Sin embargo, los resultados revelan debilidades más marcadas en escalabilidad y portabilidad, subatributos que dependen en mayor medida de la existencia de mecanismos de automatización, documentación formal y estrategias de despliegue reproducibles. La baja presencia de los subatributos de escalabilidad y portabilidad demuestra un desequilibrio importante en el diseño arquitectónico de los microservicios, donde el desarrollo se centra principalmente a la funcionalidad individual de los servicios, lo cual reduce la capacidad del sistema para adaptarse y afectando de manera directa la flexibilidad de la arquitectura.

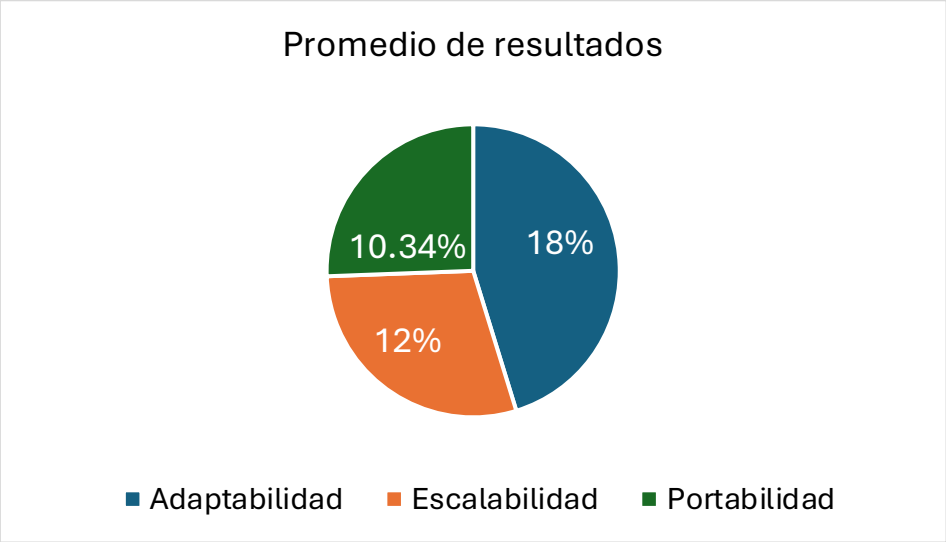


Figura 2 Promedio de resultados por subatributo

Por tanto, los resultados del análisis de los tres repositorios refuerzan la validez del esquema propuesto, al permitir identificar y cuantificar las brechas hacia los valores ideales de flexibilidad, lo cual propicia una visión más precisa de las áreas críticas que deben atenderse. Asimismo, se concluye que la flexibilidad no depende de un único subatributo, sino de la interacción equilibrada entre adaptabilidad, escalabilidad y portabilidad, lo que confirma la necesidad de adoptar mejores prácticas para integrar documentación, automatización y diseño desde las etapas iniciales del desarrollo.

5. Discusión

Los resultados obtenidos reflejan un patrón consistente con lo reportado en estudios previos, tal como lo señalaron [10, 12], la adopción de microservicios no garantiza por sí misma la consecución de altos niveles de calidad, ya que los problemas de acoplamiento, seguridad y documentación persisten incluso en sistemas bien diseñados. En el contexto del presente estudio, estas limitaciones se manifestaron especialmente en los subatributos de instalabilidad y reemplazabilidad, confirmado que la falta de manuales técnicos y mecanismos de despliegue automatizados reduce la flexibilidad operativa.

Asimismo, los hallazgos coinciden con lo observado en [26], quienes destacan que los denominados *microservice bad smells* como la duplicación de código, el acoplamiento implícito y la carencia de documentación afectan de forma directa la mantenibilidad y, en consecuencia, la flexibilidad del sistema. De modo similar, los trabajos de Vera-Rivera [15, 16] sobre granularidad evidencian que una incorrecta segmentación de servicios puede obstaculizar la adaptabilidad, al aumentar la complejidad de dependencias.

Estos resultados evidencian que la flexibilidad no debería considerarse un atributo garantizado por la MSA, sino como un resultado derivado del equilibrio entre sus subatributos. La interacción entre adaptabilidad, escalabilidad y portabilidad se configura

como un triángulo de calidad donde el debilitamiento de uno de ellos compromete a los demás.

Finalmente, el esquema propuesto demuestra su utilidad para identificar áreas de mejora y orientar estrategias de refactorización o rediseño ya que el análisis de los repositorios evidenció carencias recurrentes en aspectos clave de la flexibilidad, particularmente en lo referente a la documentación, manuales de instalación, gestión de dependencias y mecanismos de seguridad. La ausencia de dichos aspectos afecta directamente a atributos esenciales de la flexibilidad como la portabilidad, la cual a su vez se subdivide en instalabilidad y reemplazabilidad, comprometiendo su adaptabilidad en entornos reales.

De este modo, se demuestra que el simple hecho de adoptar una arquitectura basada en microservicios no garantiza, por sí sola, la flexibilidad. Para alcanzar este atributo se requiere un análisis más profundo que considere no solo la existencia de los subatributos, sino también la manera en que estos se relacionan entre sí. Esta relación resulta crítica, pues la flexibilidad emerge precisamente de la interacción y equilibrio entre sus componentes, y no de manera aislada. Sin embargo, lograr esta articulación no es tarea sencilla, ya que demanda un proceso de evaluación exhaustivo que considere los requerimientos específicos del sistema y la complejidad propia de cada entorno.

6. Conclusiones

La literatura evidencia avances significativos en el entendimiento estructural de las MSA, pero una limitada madurez en la medición empírica de sus atributos de calidad. Las métricas tradicionales resultan insuficientes, los enfoques centrados en granularidad y acoplamiento se mantienen parciales, lo cual justifica la necesidad de un modelo integrador, como el propuesto en este trabajo, que combine formalismo normativo, operacionalización métrica y aplicación empírica en contextos reales de desarrollo de microservicios.

El esquema de evaluación propuesto constituye una contribución relevante para el estudio de la flexibilidad en arquitecturas de microservicios, al integrar criterios normativos (ISO/IEC 25010 [2], ISO/IEC 9126 [3], ISO/IEC 2382 [4]) con un enfoque empírico. Su aplicación permitió comprobar que la flexibilidad depende de un conjunto de factores interrelacionados como la documentación, el desacoplamiento, la interoperabilidad y la automatización del despliegue, que deben ser gestionados de forma conjunta para garantizar la calidad del sistema.

Los resultados obtenidos confirman que las implementaciones actuales presentan brechas importantes en portabilidad y más puntualmente en instalabilidad, reflejando la necesidad de incorporar prácticas de ingeniería más estandarizadas y mecanismos de documentación más completos. Además, se demuestra que la flexibilidad no es un atributo estático, sino un indicador dinámico de madurez arquitectónica, que puede evolucionar conforme se adapten mejores prácticas de diseño y despliegue. El esquema propuesto aporta valor en este sentido, pues permite visibilizar de manera estructurada las debilidades de los microservicios. La asignación de pesos para cada atributo facilita identificar con mayor precisión qué componentes tienen mayor impacto sobre la flexibilidad de la arquitectura. No obstante, la determinación de dichos pesos exige un análisis riguroso, ya que implica valorar tanto la importancia relativa de cada subatributo como el grado de relación que guardan entre sí.

Es importante señalar que los resultados obtenidos son preliminares, al estar basados en un conjunto limitado de repositorios seleccionados bajo criterios específicos: proyectos diseñados en lenguaje Java, con el framework Spring Boot y de libre acceso. Asimismo, cabe mencionar que, si bien existen herramientas como MAAT (Microservice

Architecture Analysis Tool) [17], que ayudan a analizar aspectos de semántica, refactorización, calidad y arquitectura de microservicios, estas no ofrecen un sustento basado en normas y estándares de calidad como el que se presenta en este esquema, ni se enfocan de manera específica en el atributo de flexibilidad, el cual con frecuencia se asume erróneamente como garantizado por el simple uso de la arquitectura de microservicios.

Como trabajo futuro, se propone extender la validación del esquema mediante la aplicación de métodos formales de ponderación, incorporar métricas cuantitativas automáticas y evaluar su efectividad en entornos productivos o industriales. Con ello, se espera avanzar hacia la construcción de un marco de referencia integral para la medición de la calidad de microservicios, capaz de contribuir a la mejora continua del software y a la consolidación de estándares específicos para este tipo de arquitectura.

Así mismo se propone como trabajo futuro realizar una herramienta que pueda automatizar la evaluación de sistemas de microservicios, partiendo del esquema propuesto como base de evaluación, ya que el análisis realizado en este trabajo ha sido llevado a cabo de manera manual lo cual implica emplear una cantidad sustancial de tiempo con el que muchas veces, las empresas principalmente no cuentan.

7. Referencias

[1] Fowler, J. L. (2014). Microservices. Obtenido de MartinFawler.com: https://martinfowler.com/articles/microservices.html

[2] ISO/IEC 25000. (n.d.). ISO/IEC 25010: Características de calidad del producto. Retrieved October 18, 2024, from https://iso25000.com.

[3] International Organization for Standardization (ISO). (2001). *ISO/IEC 9126: Software engineering — Product quality* (Partes 1-4). ISO.

[4] International Organization for Standardization (ISO). (2015). *ISO/IEC 2382: Information technology — Vocabulary* (Partes 1-36). ISO. (Greg Boss et al. 2007)

[5] ISO/IEC 11179-1:2015 - Information technology — Metadata registries (MDR) — Part 1: Framework. n.d. https://www.iso.org/standard/61932.html.

[6] ISO/IEC 27001:2022 - Information security management systems. n.d. https://www.iso.org/standard/27001. (Tanenbaum and Bos 2014)

[7] IEEE SA - IEEE 1063-2001. n.d. Retrieved August 21, 2025. https://standards.ieee.org/ieee/1063/1554/.

[8] Al-Debagy, O. and Martinek, P. (2020a). A metrics framework for evaluating microservices architecture designs. Journal of Web Engineering, 19, 341–370. https://doi.org/10.13,052/jwe1540-9589.19341.

[9] Berander, P. et al. (2005). Software quality attributes and trade-offs.

[10] Li, S. et al. (2021). Understanding and addressing quality attributes of microservices architecture: A systematic literature review. Information and Software Technology, 131, 106449. https://doi.org/10.1016/j.infsof.2020.106449.

[11] Lima, S. et al. (2021). Improving observability in event sourcing systems. Journal of Systems and Software, 181, 111015. https://doi.org/10.1016/j.jss.2021.111015.

[12] Dragoni, N. et al. (2017). Microservices: Yesterday, today, and tomorrow

[13] Newman, Sam. 2015. "Building Microservices - Chapter 1, 4 and 11." Building Microservices 102.

[14] Raj, Vinay & Sadam, Ravichandra. (2021). Evaluation of SOA-Based Web Services and Microservices Architecture Using Complexity Metrics. SN Computer Science. 2. 10.1007/s42979-021-00767-6.

[15] Vera-Rivera, F. H. et al. (2021). Defining and measuring microservice granularity. PeerJ Computer Science, 7, e695. https://doi.org/10.7717/peerj-cs.695.

[16] Vera-Rivera, F. H. (2021). Modelo inteligente de especificación de la granularidad de microservicios (unpublished doctoral thesis, Univ. del Valle).

[17] Engel, T. et al. (2018). Evaluation microservice architectures: A metric-based approach. CAiSE 2018. https://doi.org/10.1007/978-3-319-92901-9_8.

[18] Li, Y. et al. (2020). Granularity decision in microservice splitting. Multi-Goal Decision Making, 1057902. https://doi.org/10.1155/2020/1057902.

[19] Perepletchikov, M., Ryan, C. and Frampton, K. (2007). Cohesion metrics for predicting maintainability of SOA. QSIC 2007, 328–335. https://doi.org/10.1109/QSIC.2007.4385516.

[20] Netflix. (2025). *Zuul* [Repositorio de código]. GitHub. https://github.com/Netflix/zuul

[21] anitagoel. (2022). AcmeAir_Microservice [Repositorio de código]. GitHub. https://github.com/anitagoel/AcmeAir_Microservice
[22] RainbowForest. (2025). E-commerce Microservices [Repositorio de código]. GitHub. https://github.com/RainbowForest/e-commerce-microservices/
[23] Booklet, D., Al-Debagy, O., & Martinek, P. (2021). Microservices Identification Methods and Quality Metrics.
[24] Ulander, D. (2017). Software Architectural Metrics for the Scania Internet of Things Platform From a Microservice Perspective. http://www.teknat.uu.se/student
[25] Kitchenham, B., & Lawrence Pfleeger, S. (1996). Software Quality: The Elusive Target. https://doi.org/10.1109/52.476281
[26] Taibi, D., & Lenarduzzi, V. (2018). On the Definition of Microservice Bad Smells. https://doi.org/10.1109/MS.2018.2141031
[27] Soldani, J., Tamburri, D. A., & van den Heuvel, W. J. (2018). The pains and gains of microservices: A Systematic grey literature review. *Journal of Systems and Software*, *146*, 215–232. https://doi.org/10.1016/j.jss.2018.09.082
[28] Newman, S. (2015). Building Microservices. http://safaribooksonline.com
[29] Al-Qutaish, R. E., & Ain, A. (2010). uality Models in Software Engineering Literature: An Analytical and Comparative Study. In *Journal of American Science* (Vol. 6, Issue 3). http://www.americanscience.orgeditor@americanscience.org166
[30] Bucchiarone, A., Kessler, B., Dragoni, N., Dustdar, S., Wien, T. U., Larsen, S. T., & Bank, D. (2018). From Monolithic to Microservices An Experience Report from the Banking Domain. https://doi.org/10.1109/MS.2018.2141026
[31] Nadareishvili, I., Miitra, R., McLarty, M., & Amundsen, M. (2016). Microservice Architecture: Aligning Principles, Practices and Culture. 146.
[32] Minella, M. (2024, June 3). State of Spring Survey 2024 Results. https://spring.io/blog/2024/06/03/state-of-spring-survey-2024-results?utm_source=chatgpt.com
[33] Sõnajalg, S. (2019, August 8). Microservices in Java: Survey Highlights | *JRebel*. https://www.jrebel.com/blog/microservices-in-java-survey?utm_source=chatgpt.com
[34] Taibi, D., Lenarduzzi, V., & Pahl, C. (2017). Processes, Motivations, and Issues for Migrating to Microservices Architectures: An Empirical Investigation. https://doi.org/10.1109/MCC.2017.4250931
[35] Pulnil, S., & Senivongse, T. (2022). A Microservices Quality Model Based on Microservices Anti-patterns. 2022 19th International Joint Conference on Computer Science and Software Engineering, JCSSE 2022. https://doi.org/10.1109/JCSSE54890.2022.9836297
[36] Yu, J., Ge, J., & Sun, J. (2021). Research on Quality Model and Measurement for Microservices.
[37] Apolinário, D. R. F., & de França, B. B. N. (2021). A method for monitoring the coupling evolution of microservice-based architectures. Journal of the Brazilian Computer Society, 27(1). https://doi.org/10.1186/s13173-021-00120-y
[38] Bogner, J., Wagner, S., & Zimmermann, A. (2017). Automatically measuring the maintainability of service- and microservice-based systems - a literature review. ACM International Conference Proceeding Series, Part F131936, 107–115. https://doi.org/10.1145/3143434.3143443
[39] Vinay Babu, D., & Darsi, M. P. (2013). A Survey on Service Oriented Architecture and Metrics to Measure Coupling.3
[40] Driessen Supervisors, F., Ferreira Pires, L., Luiz Rebelo Moreira, J., Sperotto, A., van den Bosch, S., & Verhoeven June, P. (2023). A quantitative assessment method for microservices granularity to improve maintainability.
[41] Panichella, S., Rahman, M. I., & Taibi, D. (2021). Structural Coupling for Microservices. https://doi.org/10.5220/0010481902800287
[42] Chidamber, S. R., & Kemerer, C. F. (1994). A Metrics Suite for Object Oriented Design. In *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING* (Vol. 20, Issue 6).
[43] Chen, Y. H., Chen, N. J., Xu, W. X., Lian, L. M., & Tu, H. (2021). MFRL-CA: Microservice Fault Root Cause Location based on Correlation Analysis. Proceedings - 2021 8th International Conference on Dependable Systems and Their Applications, DSA 2021, 90–101. https://doi.org/10.1109/DSA52907.2021.00018
[44] DeepWiki | AI documentation you can talk to, for every repo. (n.d.). Retrieved November 3, 2025, from https://deepwiki.com/

NOTAS BIOGRAFICAS

Said Misael Venoso Lara, es Ingeniero en Sistemas Computacionales, egresado del Tecnológico Nacional de México campus Zacatepec, actualmente es estudiante de maestría en el Tecnológico Nacional de México campus CENIDET, donde desarrolla su tesis relacionada a las métricas aplicables para microservicios. Tiene como principales gustos temas como calidad de software, métricas, principios y patrones de diseño de software, arquitecturas basadas en microservicios, arquitecturas basadas en servicios web y modelos de calidad.

Juan Carlos Rojas Pérez, doctor en Ciencias en Ciencias de la Computación egresado del TecNM/CENIDET. Actualmente profesor en el área de ingeniería de software del Departamento de Ciencias Computacionales. Áreas de interés: arquitecturas orientadas a servicios, servicios web, microservicios, lenguajes web, bases de datos, Big data, temas generales de ingeniería de software. Miembro del SNII nivel "c", perfil deseable. Evaluador de artículos para las revistas IEEE Latin America Transactions, Dyna, JCyTA, CYTCA, CIM. Fue miembro del Consejo Técnico del Examen General para el Egreso de la Licenciatura (EGEL Plus) en Ingeniería Computacional EGEL+D-ICOMPU (CENEVAL). Trabajo alrededor de 15 años en empresas de desarrollo de software:

Softtek monterrey: desarrollo de sistemas de software para General Electric Power Systems, General Electric Nuclear.

E-siglo: desarrollo de sistemas de software para Inbursa, Gobierno de México, IMSS.

Instituto de Electricidad y Energías Limpias: desarrollo de sistemas de software para el Centro Nacional de Control de energía (CENACE), Subdirección de Energéticos (SDE).

Olivia Graciela Fragoso Diaz. es actualmente profesora de doctorado de tiempo completo en ciencias de la computación en el Centro Nacional de Investigación y Desarrollo Tecnológico (TecNM/Cenidet), México. También lidera proyectos de ingeniería de software que buscan elementos de calidad utilizando tecnologías abiertas para generar mejoras en los objetivos de aprendizaje electrónico en universidades, instituciones educativas y entornos laborales. Sus intereses de investigación incluyen arquitecturas de microservicios, arquitecturas orientadas a servicios, métricas y modelos de calidad.

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

Recibido 2 Nov 2025 *ReCIBE, Año 15 No. 1, mayo 2026*

Aceptado 30 Abri 2026

Roboto: Camino a la Justicia. Un videojuego educativo para la autoprotección infantil en contextos judiciales

Roboto: Path to Justice. An Educational Video Game for Children's Self-Protection in Judicial Contexts.

Daniel Rodríguez Orozco [1]
danielr.o@outlook.es
Luis Manuel Palma Pinto [1]
palmapintoluis08@gmail.com
Santiago Efrain Itzincab Poot [1]
santiagochif@gmail.com
Víctor Hugo Menéndez Domínguez [1]
mdoming@correo.uady.mx
Reyna Faride Peña Castillo[1]
faride.pecastillo@correo.uady.mx

[1] *Facultad de Matemáticas, Universidad Autónoma de Yucatán,Mérida, Yucatán, México.*

Resumen

En México la violencia infantil es una problemática persistente, particularmente en el estado de Yucatán. Como resultado de esta realidad, se han realizado iniciativas como la plataforma Tribunales Amigables, orientada al acceso de justicia para niñas, niños y adolescentes mediante recursos didácticos y videojuegos. Bajo esta premisa, para la plataforma se propone Roboto: Camino a la Justicia, un videojuego 2D educativo con alta calidad gráfica, de género serio que tiene como objetivo enseñar principios de autoprotección, reconocimiento de situaciones de riesgo, construcción de redes de apoyo y conocimiento del proceso judicial, confianza en figuras de autoridad y el proceso básico para interponer una denuncia.

Palabras Clave: Juego Serio; Videojuego Educativo; Gamificación; Protección Infantil; Acceso a la Justicia; Educación Preventiva.

Abstract

Child violence in Mexico remains a persistent social issue, particularly in the state of Yucatán. In response to this situation, initiatives such as the Tribunales Amigables platform have been developed to promote access to justice for children and adolescents through educational resources and video games. Within this context, Roboto: Path to Justice is proposed as part of the platform: a high-quality 2D educational serious game designed to teach principles of self-protection, recognition of risky situations, the development of support networks, and understanding of the judicial process. Additionally, the game seeks to foster trust in authority figures and provide basic knowledge of the procedures required to file a report.

Keywords: Serious Game; Educational Video Game; Gamification; Child Protection; Access to Justice; Preventive Education.

Introducción

México es un país que tiene muy presente la violencia infantil en todas sus facetas: física, psicológica y sexual. Según (REDIM, 2024) durante el año 2022, tan solo en el estado de Yucatán, un total de 328 personas de entre 1 a 17 años fueron atendidas en hospitales por violencia familiar, 275 por violencia sexual y 79 por violencia física.

Como consecuencia de estos actos, las niñas y niños presentan problemas de ansiedad, comunicación, depresión, incursión de adicciones y conductas antisociales que dificultan establecer relaciones con su entorno.

En Yucatán se han promovido campañas de concientización y educación legal infantil como, por ejemplo, “De boca en boca, a mi cuerpo nadie lo toca”, que promueve el autocuidado y autoprotección en este sector de la población. De igual forma, en el contexto escolar, en años recientes, se han generado protocolos especializados para la detección y atención de casos de violencia en comunidades educativas, con el fin de proteger a niñas y niños y fortalecer la prevención y la detección oportuna.

Una iniciativa destacable en Yucatán es la plataforma digital Tribunales Amigables, que es una propuesta desarrollada por académicos y estudiantes en formación de Ingeniería de Software y Ciencias de la Computación de la Universidad Autónoma de Yucatán (UADY).

Este software interactivo tiene como propósito principal explicar de manera amigable y adaptada a las infancias y adolescencias cómo funciona el proceso de denuncia, así como su acceso y paso por el Sistema de Justicia. Se incluyen materiales de apoyo para personas cuidadoras, así como herramientas y videojuegos que permiten identificar posibles situaciones de riesgo.

En este contexto se propone Roboto: Camino a la Justicia, un videojuego 2D educativo o género serio, diseñado como parte de Tribunales Amigables, que tiene como propósito enseñar a los menores principios de autoprotección, identificar propuestas malintencionadas de personas desconocidas, fomentar la confianza en figuras de autoridad, así como el promover la expresión segura de sus vivencias dentro de una red de apoyo.

Marco Teórico

Violencia Infantil: Definición y Tipos

La Organización Mundial de la Salud (OMS) describe la violencia infantil como todo tipo de maltrato físico o afectivo, abuso sexual, desatención negligencia y explotación comercial que dañe la salud o desarrollo de un menor de 18 años (OMS, 2025).

La violencia física según (FAPMI, 2025) es cualquier acción que provoque daño físico a un menor de edad, por ejemplo, golpes, patadas, pellizcos, entre otros. Por su parte, (AEPED, 2025) la violencia psicológica o emocional se describe como interacciones que causen miedo, amenazas, explotación, rechazo o discriminación hacia una niña, niño o adolescente (NNA). También se considera la existencia de maltrato de forma expresiva a través de insultos, intimidación o ciberacoso. Finalmente, en (Pérez, 2013) la violencia sexual, también llamado abuso sexual, ocurre cuando se inducen prácticas sexuales no deseadas que causan daño, dado que estos no tienen la madurez suficiente para dar un consentimiento válido.

Iniciativas con la Violencia Infantil en Yucatán desde la UADY

Como parte de las intervenciones a nivel preventivo desarrolladas en el Estado (FGE, 2025), desde la responsabilidad social coordinado por la Universidad Autónoma de Yucatán, se cuenta con el Programa “De Boca en Boca, a mi Cuerpo Nadie lo Toca” cuyo objetivo es desarrollar habilidades y fortalecer el autocuidado y autoprotección de NNA para la detección de situaciones de riesgo, proporcionando estrategias a través del uso de material psicoeducativo y lúdico, utilizando sus recursos personales con la colaboración importante de progenitores, personas cuidadoras y docentes. Este programa se ha aplicado en coordinación de la Secretaría de Educación Pública del Estado de Yucatán a nivel preescolar, en municipios donde se ha ubicado la alerta de género. Los cuatro grandes ejes temáticos del programa son: partes públicas y privadas, secretos buenos y malos, situaciones de riesgo y personas de confianza. Se cuentan con cuentos, audiocuentos, un teatro guiñol, guías para personas cuidadoras y un kit psicoeducativo interactivo.

Es por ello, en respuesta a esta necesidad nace el proyecto Tribunales Amigables como una plataforma en línea que busca facilitar el acceso a recursos de justicia para niñas, niños y adolescentes a través de una manera interactiva. Esta plataforma representa una guía relevante para padres y menores durante los procesos jurídicos en los que participan, siendo así un avance significativo para esta población vulnerable (Poder Judicial del Estado de Yucatán, 2025).

De igual forma, Tribunales Amigables realiza acciones de prevención de la victimización secundaria dirigida a niñas, niños, niñes y adolescentes en contacto con el sistema de justicia (Tribunales Amigables, 2025). Desde la preparación previa a su participación en audiencias, se brindan herramientas psicoemocionales y acompañamiento sensible para reducir riesgos de revictimización, ansiedad y trauma, convirtiéndose en una iniciativa innovadora y única en su estilo en el país para amortiguar procesos sensibles en el acceso a justicia con recursos que acompañan y promueven una justicia más sanadora.

Gamificación y Juegos Serios como Herramientas Educativas

Según (Moore-Russo, 2018) la gamificación es un concepto popular en la educación, se ha comprobado que mejora los resultados de aprendizaje en los estudiantes y surge a partir de juegos interactivos. Su principal función es la de marcar un compromiso con los estudiantes, utilizando elementos competitivos, sistemas de recompensas, progresión y narrativa fantástica. Por otra parte, (Peña, 2014) define que un juego serio es una aplicación interactiva cuya finalidad no es el ocio, sino la formación de estudiantes en diversos ámbitos como salud, marketing y educación. Estos tipos de juegos tienen como propósito utilizarse como herramienta formativa.

Videojuegos de género serio relacionados al abuso infantil

En esta sección se hace una recopilación de videojuegos de género serio a nivel nacional en México que tienen como tópico central el abuso o violencia infantil.

- ***Chuka: Rompe el Silencio.*** Este es un videojuego desarrollado por la Oficina de las Naciones Unidas contra la Droga y el Delito (UNODC) y Secretaría de Gobernación de México. Su principal objetivo es enseñar a niñas y niños (de entre 7 y 12 años) a reconocer y enfrentar distintas formas de violencia (física, psicológica, sexual) mediante minijuegos y una narrativa simbólica. El videojuego no aborda directamente el proceso judicial ni mecanismos de denuncia; su enfoque es general en violencia, no exclusivamente en abuso infantil (UNODC México, 2025).
- ***Patito y las Burbujas.*** El desarrollador es Pink Bear Games (estudio independiente mexicano). Su objetivo principal es el de enseñar a las niñas y niños de entre 4 a 8 años sobre autocuidado, límites personales y expresión emocional de forma lúdica. En este videojuego no se aborda de forma directa el abuso infantil, se enfoca en la prevención general y el desarrollo emocional (Pink Bear Games, 2025).
- ***Kiddy Kingdom***. Este es un proyecto educativo mexicano elaborado por la empresa Ludolab como desarrolladores independientes. Está diseñado para niñas y niños de entre 6 a 10 años y se centra en prevenir el abuso infantil a través de dinámicas que enseñan a reconocer situaciones peligrosas y pedir ayuda. Proyecto experimental, no disponible públicamente en plataformas oficiales; en fase de desarrollo o prototipo (Ludolab, 2025).
- ***Ley de vida.*** Este videojuego creado por alumnos de la Universidad del Tecnológico de Monterrey, en su sede ubicada en Cuernavaca, trabajando en conjunto con el Tribunal Unitario de Justicia Penal para Adolescentes (TUJPA) de la entidad federativa de Morelos. Su propósito fundamental es crear conciencia en los jóvenes sobre las consecuencias de tomar malas decisiones, así como sobre el consumo de sustancias y los peligros de la violencia. El juego busca promover la reflexión acerca de la responsabilidad individual y el efecto social que tienen los actos violentos mediante una narración divertida e interactiva. Aunque no se

centra directamente en el maltrato a los niños, comparte la intención educativa y preventiva típica de los juegos serios (Tecnológico de Monterrey, 2025).

- ***The Book of Glory.*** Diseñado por alumnos del Tecnológico de Monterrey, campus Chihuahua, este juego de rol tiene como objetivo fomentar la salud mental y el bienestar emocional en jóvenes. El jugador considera valores como la resiliencia, la autoestima y la empatía a través de decisiones dentro de una historia fantástica. Este proyecto se distingue por su componente psicológico y su propósito de fomentar el aprendizaje emocional en los jóvenes, a pesar de que no trata directamente el tema del abuso infantil (Tecnológico de Monterrey, 2025).
- ***Yes or No.*** Protocolo para prevenir y detectar el maltrato infantil. Este juego educativo interactivo, desarrollado en México y accesible mediante la plataforma Educaplay, tiene como objetivo concientizar a niñas y niños con respecto a la violencia de género, el acoso escolar y el abuso sexual. Se fomenta que los adultos de confianza sean consultados y que las situaciones de riesgo sean reconocidas mediante escenarios de decisión y preguntas tipo trivia. Aunque su estructura es más sencilla que la de un videojuego convencional, su finalidad educativa lo sitúa dentro del género de los juegos serios (Educaplay México, 2025).

A diferencia de los videojuegos listados, Roboto: Camino a la Justicia no solo se centra en enseñar a los menores a reconocer situaciones de riesgo, sino que también los guía paso a paso sobre cómo buscar ayuda, cómo confiar en las figuras que son de autoridad y cómo aprender y entender los pasos básicos para interponer una denuncia. Además, su narrativa está contextualizada en la realidad yucateca y su diseño visual atractivo permite un aprendizaje lúdico y significativo. Por estas razones, representa una de las propuesta más completa e innovadora dentro del ámbito de los videojuegos serios enfocados principalmente en la protección infantil en México.

Roboto: Camino a la Justicia

Relevancia y Justificación

Roboto: Camino a la Justicia nace como una propuesta creativa ante la necesidad de ayudar a niñas y niños con procesos de justicia, cuidado propio y confianza en figuras de autoridad responsables. En un entorno social donde muchos menores enfrentan riesgos como abusos, manipulaciones, o intimidación, este videojuego resulta en una idea creativa para ofrecer conocimiento de forma lúdica.

El videojuego emplea personajes de la plataforma Tribunales Amigables, como Tito, un jaguar valiente que representa una figura de autoridad positiva; Roboto, un robot curioso que está aprendiendo sobre las emociones humanas; y los dos personajes principales que el usuario podrá seleccionar al iniciar el juego: Itzel o Juanito (Figura 1). Estos dos personajes viven una aventura en búsqueda de justicia.

Figura 1. Menú para seleccionar los personajes del juego

La historia comienza cuando Tito enseña a Roboto sobre cómo presentar una denuncia ante los tribunales. De pronto, un grupo de personas malintencionadas los sorprenden y roban las partes del robot, dejando únicamente su cabeza (Figura 2). A partir de ahí, el jugador debe ayudar a reconstruir a Roboto, recorriendo distintos escenarios desde una comunidad maya hasta llegar al tribunal. Durante el camino, se enfrentan retos que reflejan situaciones reales que un menor podría atravesar al intentar buscar ayuda.

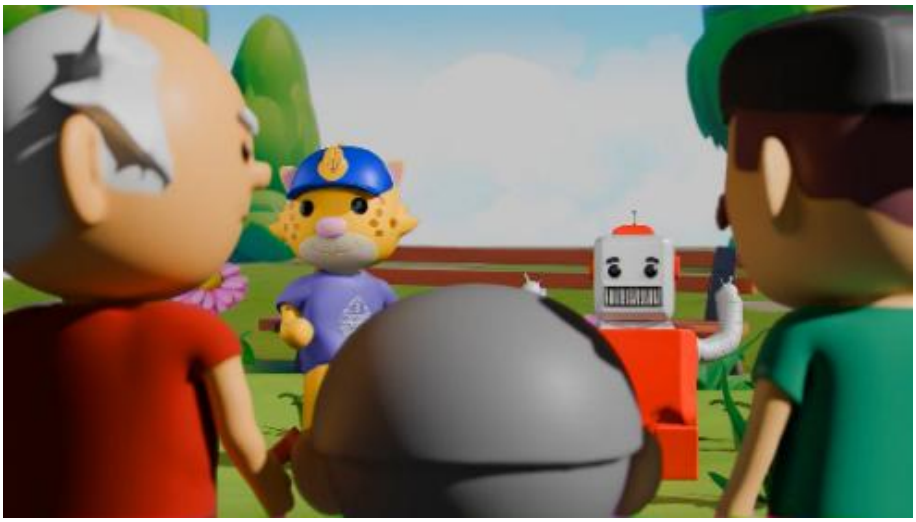


Figura 2. Segmento de cinemática inicial

Funcionalidades y Características Generales

El videojuego consta de tres niveles, cada uno con objetivos específicos y desafíos distintos. Cada nivel muestra problemas que un niño podría enfrentar al buscar ayuda: desde no confiar en personas desconocidas, hasta lidiar con amenazas. Así, el propósito del juego no es solo entretener; también reproduce momentos difíciles que permiten al jugador mejorar habilidades sociales, pensar críticamente y aprender maneras de defenderse.

En el primer nivel se presentan personas de confianza, como médicos, policías y jueces que ayudan al jugador a avanzar. También aparecen obstáculos y personajes hostiles que deben ser evitados. El segundo nivel presenta al villano principal, quien

lanza preguntas con intenciones engañosas para determinar la capacidad de intuición de los niños al estar enfrente de personas desconocidas. Finalmente, el tercer nivel presenta todos los procesos básicos para realizar una denuncia.

La propuesta combina tres estilos de jugabilidad: novela gráfica interactiva, juego de plataformas, como el clásico Super Mario Bros y un estilo RPG con vista cenital, como Pokémon Esmeralda 2005. Esta fusión busca reforzar tanto el contenido formativo como el atractivo visual del juego.

El juego fue creado y diseñado para funcionar en diversas plataformas como celulares y computadoras, pero su enfoque fue la implementación en web para facilitar su uso en ambientes educativos e incorporación a la plataforma de Tribunales Amigables. En la Figura 3 se muestra, itch.io, una página web dedicada para el alojamiento de videojuegos y en donde se realizaron las primeras pruebas correspondientes.

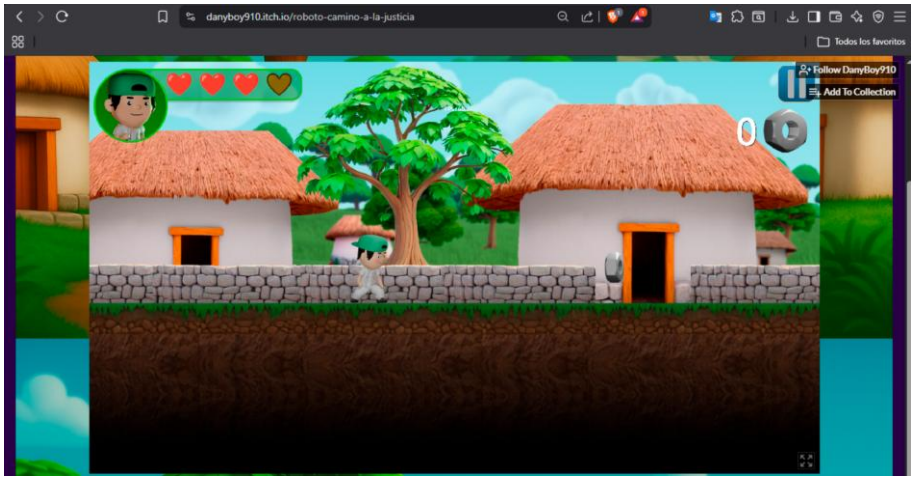


Figura 3. Ejecución del juego en el navegador Brave

Metodología para el Desarrollo del Videojuego

En el diagrama de la Figura 4, se desglosan las etapas seguidas para el desarrollo de Roboto: Camino a la Justicia. Para gestionar el proyecto se optó por la metodología ADDIE (Analysis, Design, Development, Implementation & Evaluation), descritas en (Branson RK, 1988). Sin embargo, el modelo original estándar no contempla elementos propiamente para el desarrollo de videojuegos de género serio, como el diseño de mecánicas de juego o inclusive la gestión o creación de diseños visuales como recursos 3D. Es por ello, que fue necesario adaptar el modelo. A continuación, se detallarán las fases y ajustes que se realizaron para desarrollar el videojuego.

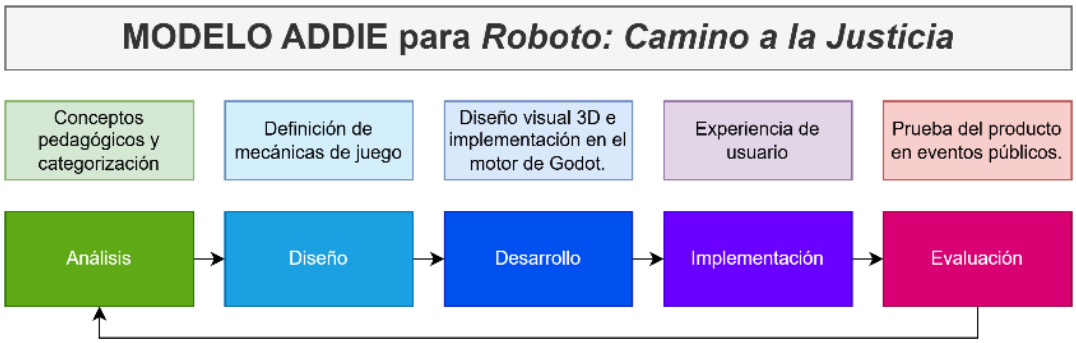


Figura 4. Diagrama ADDIE para el desarrollo del videojuego Roboto: Camino a la justicia

Con ayuda de especialistas en el área de psicología de la Universidad Autónoma de Yucatán en la fase de Análisis se planteó el problema educativo que se buscaba resolver, así como los objetivos de aprendizaje. Dentro de este rubro se planteó que el juego debía enseñar principios de autoprotección, el reconocimiento de situaciones de riesgo, la identificación de personas de confianza y el conocimiento básico del proceso de denuncia.

Posteriormente en la etapa de Diseño, se retomaron los objetivos de aprendizaje para poder crear las mecánicas de juego donde se aplicarían los conceptos de gamificación para un aprendizaje lúdico. Para obtener variabilidad en los formatos de juego se crearon tres modos de juego. Un formato de novela gráfica interactiva el cual con ayuda de un apoyo narrativo se presentaban situaciones de riesgo y trivias para que el jugador tome decisiones. También se presentó un estilo de plataformas 2D tipo Super Mario Bros para la exploración de niveles, reconocimiento de personas de confianza y personajes enemigos. El tercer modo de juego se basó en un RPG de vista cenital de exploración al estilo de Pokémon, en el cual el jugador debía explorar las instalaciones reales del Centro de Instituciones Operadoras del Sistema Penal Oral y Acusatorio (CIOSPOA) y la Sala de Audiencias para Personas en Condición de Vulnerabilidad (SAPCOV), así mismo el objetivo principal era el proceso básico para interponer una denuncia, Figura 5.

Figura 5. Tercer modo de juego RPG con vista cenital en el CIOSPOA

En la fase de Desarrollo, se llevó a cabo un extenso trabajo de diseño visual y programación. El motivo por el cual se seleccionó el motor de videojuegos Godot

Engine es porque es de código abierto, cumple con el requisito de exportación en HTML5 y la curva de aprendizaje es baja, dado que el lenguaje de programación que utiliza (GDScript) es similar a Python. La implementación tuvo una extensa carga de trabajo en la parte visual. Se llevó a cabo la elaboración de 14 personajes principales más 4 personajes de tipo extra en 3D, Figura 6. Algunos de los personajes ya contaban con sus referencias en 2D, pero otros se debían realizar desde cero. La elaboración de los modelos 3D se explicarán con más detalle en las siguientes secciones.

Figura 6. Modelos 3D de los personajes principales del videojuego

Por otra parte, también se realizó la creación de los niveles y mapas 3D para cada respectiva misión. El primer nivel se ambientó en una mezcla de una comunidad rural maya con casas tradicionales, pasando por la selva maya, así como también por cenotes. El segundo nivel se basó en las calles del centro de la Ciudad de Mérida Yucatán, tomando ideas representativas como el transporte público "Va y Ven" y las casas coloniales. El tercer nivel implicó una carga de trabajo más grande, puesto que se debía modelar en 3D las instalaciones del CIOSPOA y el SAPCOV. Se realizó una visita guiada con el fin de tomar imágenes de referencia de la distribución espacial para modelar a detalle las locaciones del edificio, Figura 7.

Figura 7. Modelo 3D del CIOSPOA implementado en Blender

En la fase de Implementación se consideraron los aspectos relacionados a interacción humano computadora, como son los elementos gráficos de menús interactivos, pruebas técnicas, detección de errores, despliegue en un entorno web real de una primera versión de prueba, optimizar elementos de carga, pruebas de rendimiento en dispositivos de gama baja, y mejorar la experiencia de usuario.

Finalmente, en la fase de Evaluación, consistió en la realización de actividades con niñas y niños durante diferentes eventos abiertos al público en general, Figura 8. El objetivo de esta etapa fue evaluar elementos de interacción, así como la recepción de los conceptos pedagógicos. Gracias a esto, se obtuvieron las primeras retroalimentaciones que de forma natural sirvió como una base para el trabajo futuro en las actualizaciones, identificando aquellas mejoras que involucraban la optimización de recursos.

Figura 8. Fase de Evaluación en niñas y niños en la Facultad de Matemáticas UADY

En el diagrama presentado de la Figura 4, se debe notar que al final de la fase de Evaluación se muestra una flecha retornando, como símbolo de que este proceso se

realizó de forma iterativa. Con esto se lograba refinar el producto, facilitar el desarrollo y posteriormente desplegar de nuevo.

Aspectos Técnicos

Modelos 3D de Personajes

El proyecto inició con una ventaja: el equipo de Tribunales Amigables tenia los diseños de los personajes en formato 2D, esto proporcionó una base sólida para la generar sus versiones en 3D. Se utilizaron tres herramientas para ello:

- ***Rodin (Hyper3D.ia):*** Utilizada para transformar imágenes 2D en modelos 3D básicos mediante inteligencia artificial.
- ***Blender:*** software libre para modelado y animación 3D (Blender Foundation, 2025).

Mixamo (Adobe): plataforma que facilitó la animación de los personales.

El flujo de trabajo consistió en capturar una imagen del personaje en 2D, procesarla en Rodin para generar una modelo 3D inicial y ajustarla con Blender, mejorando proporciones, texturas y detalles visuales. Finalmente se cargó en Mixamo para generar las animaciones y armaduras, (Figura 9).

Figura 9. De izquierda a derecha se muestra el proceso de creación del personaje Juanito en formato 3D. Se parte de un dibujo hecho con un diseño 2D, se procesa con Rodin y finalmente se ajusta el modelo 3D en Blender

Diseño de Niveles

Dado que la historia cuenta la aventura de recuperar las piezas de “Roboto” lo más acertado era utilizar diversos escenarios y ambientaciones únicos para cada nivel. De esta forma, el primer nivel transcurre en una zona rural donde la cultura maya tiene un papel protagónico. Cuando se planeó el bocetaje de este nivel se pensó en utilizar aquellos elementos representativos de la cultura que permitirían darle una sensación de que realmente la historia ocurriera en Yucatán. Para ello, se recrearon elementos como la Pirámide de Chichen Itzá que aparece brevemente en el fondo del juego al iniciar la primera misión, En la Figura 10 se observa el contraste entre un boceto inicial del primer nivel (arriba) y su versión final (abajo).

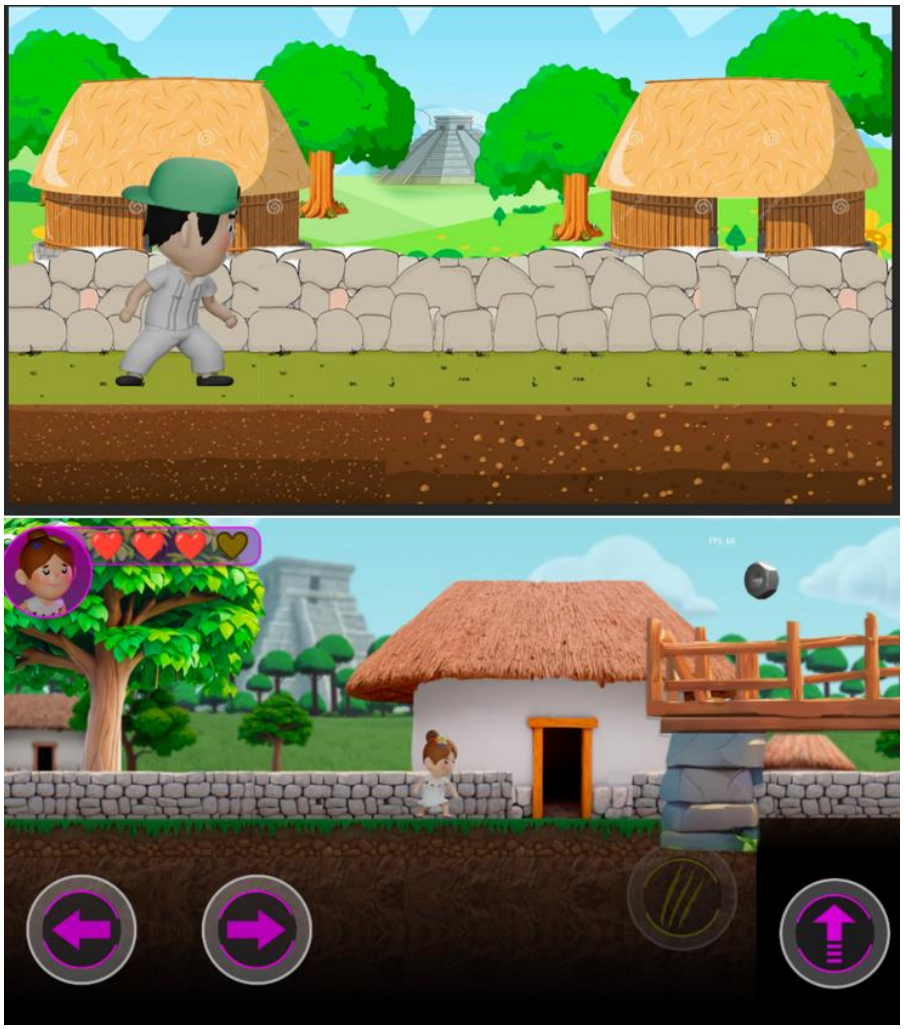


Figura 10. Boceto y versión final del nivel uno, La Casa Maya

También se diseñaron las casas tradicionales mayas con sus arquitecturas características de vernáculos. Conforme se avanza por este nivel, se atraviesa la selva maya característica de la región, así mismo, se diseñó una sección para representar a los cenotes, Figura 11.

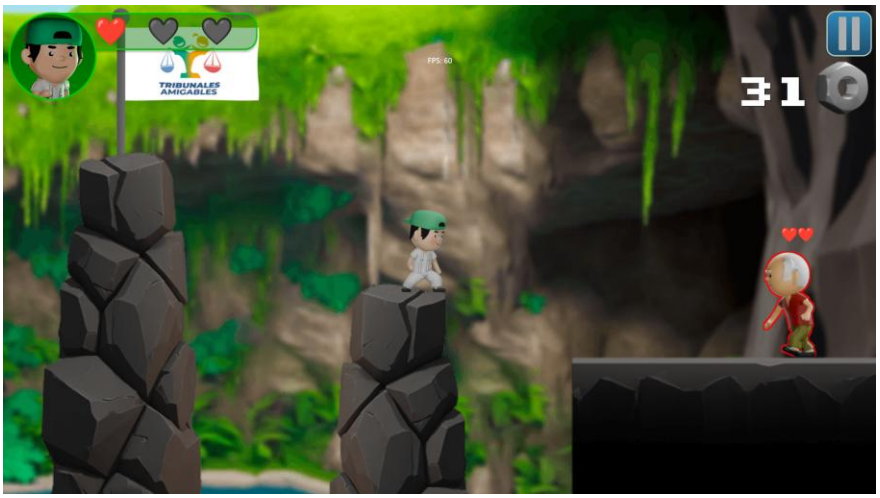


Figura 11. Segmento del primer nivel en un cenote

El segundo nivel se ambienta en el centro de Mérida, Yucatán; Este nivel a diferencia del anterior, refleja lo que se vive diariamente en la ciudad. Se contrastan ambos panoramas con el fin de que el juego se sienta más inmersivo. Las características principales que denotan a este nivel son las distintas casas coloridas con arquitectura colonial que existen en la ciudad. Dentro de la jugabilidad de este nivel se implementó como enemigos al tránsito vehicular de la ciudad, es decir coches particulares y servicios de transporte público, Figura 12.

Figura 12. Segundo nivel ambientado en el centro de la ciudad de Mérida, Yucatán

Finalmente, el último nivel es el edificio CIOSPOA. Aquí se le da especial énfasis en la Sala de Audiencia para Personas en Condición de Vulnerabilidad (SAPCOV), Figura 13. Se realizó una visita particular al edificio y se tomaron fotos para conocer los espacios. El resultado fue un modelo 3D muy similar al original considerando todas las características posibles para que los niños puedan ubicar exactamente el lugar sin la necesidad de inclusive haber asistido.

Figura 13. Tercer nivel dentro del SAPCOV

Para crear estos niveles se utilizaron herramientas de inteligencia artificial generativas como ChatGPT (Open AI, 2025) y Fooocus. Cabe aclarar que esta etapa fue de ensayo y error, hasta que se logró dar con un estilo que fuera visualmente

amigable, colorido y con un tono cálido, la cual se definió como “arte 3D con estética de juguete”.

Al final, los escenarios de cada nivel eran una combinación de varios elementos que fueron editados en Photoshop, donde se ajustaron los componentes visuales. Figma (Figma, 2025) permitió realizar los prototipos de los niveles antes de importarlos en el motor.

Novela Gráfica

La novela gráfica forma una parte esencial del juego, ya que plantea situaciones reales de riesgo que los jugadores deben identificar y resolver. Cada escena presenta un entorno visual coherente con los niveles, acompañado de personajes, diálogos y un sistema de preguntas y respuestas tipo trivia, lo que mantiene la atención continua del jugador (Figura 14). Para el diseño narrativo (diálogos, vocabulario y trivias) se contó con el apoyo de personal de la Unidad Universitaria de Clínica e Investigación Victimológica (UNIVICT) de la UADY.

Figura 14. Novela gráfica del nivel dos. Un sistema de preguntas y respuestas

La arquitectura utilizada para la novela gráfica incluye:

- Imagen de fondo persistente por escena.
- Personajes con aparición y participación en diálogos con el jugador.
- Cuadro de texto que presenta los diálogos del personaje principal.
- Botón de ayuda o pistas.
- Voces, música y efectos para una experiencia inmersiva.

Animaciones y Cinemática

Las cinemáticas no solamente fungen como un elemento de entretenimiento, bajo este contexto refuerza el vínculo emocional dado que por medio de un video se argumenta el trasfondo de la historia. Al brindar vida a los personajes por medio de acciones se fomenta la empatía con el jugador. Cuando se logra este tipo de conexión entre videojuego y usuario es más fácil que se logre transmitir el mensaje que se desea. Por otra parte, debido a que el público objetivo son especialmente niñas y niños las cinemáticas funcionan como una recompensa cognitiva, es decir, se ofrece una recompensa visual que ayuda a mantener el interés durante el juego.

Además, las cinemáticas especialmente se utilizaron para traducir aquellos conceptos crudos como lo son el robo, la agresión, la resolución de conflicto y otros

elementos de una manera más narrativamente visual y digerible. Algo que particularmente que con diálogo o lectura sería más complicado de lograr. Con el propósito de brindar una experiencia más entretenida se crearon tres cinemáticas animadas con Blender: una escena de introducción donde se presenta el robo de las piezas de Roboto, una de cierre con la captura del villano, y una escena oculta como premio para quienes completan el juego con misiones secundarias. Estas cinemáticas se diseñaron pensando especialmente en captar la atención del público infantil y mantener el vínculo emocional con la historia, Figura 15.

Figura 15. Fotogramas de distintas cinemáticas

Voces y Sonidos

Era indispensable agregar música, sonido y voces con la finalidad de darle vida al videojuego. Se generaron las voces de los personajes con la herramienta Eleven Labs. (Eleven Labs, 2025) De toda la colección disponible se fueron haciendo pruebas para decidir la más adecuada a cada personaje según su personalidad. Los efectos de sonido (pisadas, golpes, sonidos de UI) fueron obtenidos de bibliotecas libres como Pixabay. Posteriormente fueron procesados en Audacity optimizarlo formato, tamaño y calidad. Para la música de fondo, se empleó la herramienta Suno AI [Suno AI, 2025], que permitió generar pistas personalizadas a partir de descripciones narrativas del videojuego.

Programación del Videojuego

La condición para la creación del videojuego fue que pudiera exportarse en HTML5 para ser ejecutado en navegadores web. Así que se revisaron varios motores de videojuegos como Unreal Engine o Unity, pero se descartaron por requerir mayor carga técnica o hardware más robusto. Finalmente, Godot (Godot Engine, 2025) resultó ser la mejor opción ya que cumplía con la principal exigencia. Su lenguaje de programación GDScript es muy similar a Python, un lenguaje con el que todos los miembros del equipo de desarrollo ya tenían experiencia, lo que disminuyó la curva de aprendizaje y permitió centrarse en el desarrollo. Fue tan ágil la velocidad de desarrollo que se logró tener una primera versión de prueba del juego (ver https://danyboy910.itch.io/roboto-camino-a-la-justicia). Esto representó una gran ventaja para el equipo de desarrollo, dado que al estar desplegado en un ambiente similar al final permitió realizar pruebas para detectar errores y mejorar aspectos como el rendimiento de carga, especialmente en dispositivos de gama media o baja.

Por otro lado, el Diagrama de Capas, Figura 16, resume la arquitectura del juego que se organiza en una estructura de tres niveles que trabajan de forma integrada para entregar una experiencia completa al usuario. Primero, en la Capa de Usuario o de presentación, representa un contacto directo con el jugador, ya que integra todos los componentes de interfaz visual y auditiva del juego. Aquí se encuentran elementos como Sprites, animaciones, personajes, escenarios, sonido y música que dan vida a al videojuego. Igual, cuenta con una interfaz de usuario (HUD/UI) que muestra información esencial para el jugador, esto le permite monitorear su progreso en

tiempo real como puntuación, vidas restantes y monedas recolectadas. Cabe señalar que estos elementos son recuperados gracias a la capa de información.

Segundo, la Capa de Negocios, principalmente permite orquestar la experiencia completa para el jugador, cuando se trata de jugabilidad. Esta capa se divide en dos secciones de lógica principal: la primera se enfoca en la novela gráfica que presenta diálogos, animaciones, trivias y una narrativa educativa que tienen como propósito transmitir un mensaje sobre derechos y justicia. Y la segunda sección contiene la lógica para el nivel de plataforma, que maneja varios subsistemas que trabajan simultáneamente como: la gestión de acciones del jugador, que considera el movimiento del personaje (caminar, saltar, golpear, etc.) e interacción con el ambiente y otros personajes como Npc's; el manejo de datos importantes como metas, caídas, y puntos de control; un sistema de físicas, que detecta colisiones con objetos del ambiente, enemigos, golpes, interacción con objetos interactivos como potenciadores y monedas, entre otros; y un sistema de gravedad, que permite identificar el movimiento e interacción de un personaje al momento de descender si es que se ha elevado realizando un salto.

Finalmente, la Capa de Información almacena y organiza todos los recursos necesarios para que el juego funcione correctamente, actuando como la base de datos (sistema de archivos organizados que utiliza el propio motor del videojuego para recuperar información, no es una base de datos como tal) fundamental del proyecto. Esta capa se divide en dos componentes principales que trabajan en conjunto:

El primero es el Sistema de niveles, que administra la estructura completa de cada nivel mediante "TileMaps" que definen el diseño y distribución del terreno, las plataformas que marcan el recorrido del jugador, los objetos interactivos y decorativos dispersos en cada escenario, y los puntos de aparición del jugador principal. Este sistema mantiene una organización completa relacionada al progreso y diseño de cada fase (etapa) del juego, de tal forma, se asegura que cada nivel cargue correctamente con todos sus elementos en las posiciones correspondientes.

El segundo componente es el Núcleo del Sistema, responsable de gestionar los recursos básicos como Sprites (personajes y objetos), sonidos, y scripts que contienen el código de programación necesario para el funcionamiento de cada mecánica. Además, este núcleo se encarga de controlar las entradas del jugador mediante un sistema de mapeo de controles, es decir, interpreta los clics del ratón, las teclas presionadas por el teclado, o las entradas de los mandos (controles de videojuegos) con la finalidad de traducirlas en acciones dentro del juego. De igual forma, se encarga de administrar elementos fundamentales como el "Game Manager", in componente que se encarga del control de puntuaciones, vidas, y la gestión de variables globales que deben persistir en diferentes escenas. Por último, este núcleo maneja la carga de transiciones entre escenas para garantizar una experiencia fluida y sin interrupciones.

Figura 16. Diagrama de Capas de la Arquitectura del juego

Conclusión

A través de este desarrollo, se logró crear una herramienta que permite educar a los menores sobre métodos de autoprotección, toma de decisiones y expresión emocional en contextos seguros. Se observa que Roboto: Camino a la Justicia puede ser utilizado como una herramienta pedagógica que además de entretener pueda demostrar conceptos importantes para los niños. Aunado a esto, se establece que la metodología ADDIE con un ajuste puede funcionar como base para la elaboración de videojuegos serios orientados a la educación. El uso de la metodología facilitó la comunicación entre profesionistas de otras áreas, así como el diseño de una experiencia atractiva y comprensible.

Además de su función educativa, también es un caso de cómo las tecnologías interactivas pueden contribuir al fortalecimiento de la cultura de denuncia y protección de los niños en México. Su evolución evidencia que los videojuegos tienen el potencial de ser una herramienta seria para la transformación social si se combinan con un enfoque interdisciplinario, narrativa contextual y contenido pedagógico. Este tipo de ideas fomentan la formación de nuevas generaciones que estén mejor informadas, con mayor empatía y más seguras de sus derechos, contribuyendo a forjar así una sociedad más equitativa e inclusiva.

De igual forma este proyecto permite que en el futuro se desarrollen nuevas líneas de investigación y desarrollo relacionadas con la utilización de videojuegos serios como métodos de aprendizaje en ámbitos educativos y sociales. Lo que se ha diseñado y se ha evaluado podría servir como base para crear nuevas herramientas digitales que se ocupen de otros problemas que afectan a los niños, por ejemplo, el

acoso escolar, la violencia doméstica o la educación emocional. De esta manera, no solo aporta una respuesta concreta, sino que también sienta las bases para seguir evaluando el potencial de los videojuegos como herramientas para la transformación social y educativa.

Como trabajo futuro se pretende mejorar el proceso de carga optimizando los recursos de multimedia, la expansión de la narrativa con nuevas misiones, agregar más personajes o un sistema donde el jugador pueda crear su propio avatar; la incorporación de un sistema de retroalimentación directa por los jugadores. Estas mejoras podrían potenciar mucho más el alcance y efectividad del videojuego como una herramienta de prevención y apoyo para la infancia.

Es necesario aclarar que para la primera versión de prueba el videojuego se alojó en una página gratuita de videojuegos llamada Itch.io donde cualquier usuario puede tener acceso actualmente. Por otra parte, el producto original terminado para su lanzamiento estará disponible en la página oficial de Tribunales Amigables, el cual se ejecutará directamente en web o se podrán descargar las versiones multiplataformas para Windows y Android.

Referencias

[1]. Red por los Derechos de la Infancia en México (REDIM). (2024). Balance anual REDIM 2024. Retrieved July 16, 2025, from https://issuu.com/infanciacuenta/docs/balance_anual_redim_2024

[2]. Organización Mundial de la Salud. (2025). Maltrato infantil. Retrieved July 2, 2025, from https://www.who.int/es/news-room/fact-sheets/detail/child-maltreatment

[3]. FAPMI – Federación de Asociaciones para la Prevención del Maltrato Infantil. (2025). Tipos de violencia contra niños y niñas. Retrieved July 2, 2025, from https://fapmi.es/Tipos-de-Violencia-Contra-Ninos-y-Ninas/

[4]. Asociación Española de Pediatría. (2025). Definiciones sobre tipos de violencia. Retrieved July 2, 2025, from https://www.aeped.es/Una-Vision-Global-Violencia-Contra-Ninos/Definiciones

[5]. Pérez Contreras, M. M. (2013). Violencia y maltrato infantil en la familia: una reflexión actual y prospectiva. Instituto de Investigaciones Jurídicas, UNAM.

[6]. Fiscalía General del Estado de Yucatán. (2025). Sociedad y gobierno suman esfuerzos contra el abuso sexual infantil. Retrieved July 2, 2025, from https://fge.yucatan.gob.mx/noticias/5042

[7]. Poder Judicial del Estado de Yucatán. (2025). Poder Judicial de Yucatán da paso firme para fortalecer el acceso a una justicia adaptada y especializada. Retrieved July 2, 2025, from https://www.pjyucatan.gob.mx/secciones/prensa_comunicado/173_poder_judicial_de_yucatan_da_paso_firme_para_fortalecer_el_acceso_a_una_justicia_adaptada_y_especializada

[8]. Tribunales Amigables. (2025). Prevención Amigable. Retrieved July 3, 2025, from https://prevencionamigable.com.mx/

[9]. Moore-Russo, D., Wiss, A., & Grabowski, J. (2018). Integration of gamification into course design: A noble endeavor with potential pitfalls. College Teaching, 66, 3–5. https://doi.org/10.1080/87567555.2017.1295016

[10]. Peña Miguel, N., & Sedano-Hoyuelos, M. (2014). Serious games: herramientas para el aprendizaje. Edutech Review: International Education Technology Review, 1. https://doi.org/10.37467/GKA-REVEDUTECH.V1.1094

[11]. United Nations Office on Drugs and Crime. (2025). Chuka: Rompe el silencio. Retrieved October 30, 2025, from https://www.unodc.org/e4j/es/primary/e4j-tools-and-materials/chuka.html

[12]. Pink Bear Games. (2025). Patito y las burbujas: Juego educativo sobre autocuidado infantil. Retrieved October 30, 2025, from https://play.google.com/store/apps/details?id=com.PinkBearGames.Patito

[13]. Ludolab. (2025). Kiddy Kingdom: Un juego serio que busca prevenir el abuso infantil. Retrieved October 30, 2025, from https://ludolab.org/kiddy-kingdom-un-juego-serio-que-busca-prevenir-el-abuso-infantil/

[14]. Tecnológico de Monterrey. (2025). Ley de vida: Videojuego desarrollado para el Tribunal Unitario de Justicia Penal para Adolescentes (TUJPA) de Morelos. Retrieved October 30, 2025, from https://conecta.tec.mx/es/noticias/cuernavaca/educacion/game-over-alumno-tec-disena-videojuego-que-busca-prevenir-violencia

[15]. Tecnológico de Monterrey. (2025). The Book of Glory: Videojuego educativo creado por estudiantes del campus Chihuahua para promover la salud mental en jóvenes. Retrieved October 30, 2025, from https://conecta.tec.mx/es/noticias/chihuahua/emprendedores/con-juego-de-rol-alumnos-del-tec-promueven-la-salud-mental
[16]. Educaplay México. (2025). Yes or No: Protocolo para la prevención y detección del abuso infantil. Retrieved October 30, 2025, from https://www.educaplay.com/learning-resources/20737728-protocolo_para_la_prevencion_deteccion_y_actuacion_en_caso_de_abuso_sexual_infantil_acoso_escolar_y_maltrato_en_las_escuelas_de_educacion_basica_del_seiem.html
[17]. Branson, R. K., Rayner, G. T., Cox, J. L., Furman, J. P., King, F. J., & Hannum, W. H. (1975). Interservice procedures for instructional systems development. NTIS.
[18]. Norman, D. A. (1988). The design of everyday things. Basic Books.
[19]. Blender Foundation. (2025). Blender. Retrieved July 2, 2025, from https://www.blender.org
[20]. ChatGPT. (2025). Retrieved July 2, 2025, from https://chat.openai.com
[21]. Figma. (2025). Retrieved July 2, 2025, from https://www.figma.com
[22]. ElevenLabs. (2025). ElevenLabs voice AI. Retrieved July 2, 2025, from https://www.elevenlabs.io
[23]. Suno AI. (2025). Retrieved July 2, 2025, from https://www.suno.ai
[24]. Godot Engine. (2025). Retrieved July 2, 2025, from https://godotengine.org

NOTAS BIOGRÁFICAS

Daniel Rodríguez Orozco es Licenciado en Ciencias de la Computación por la Facultad de Matemáticas de la Universidad Autónoma de Yucatán (UADY). Su interés se centra en la animación 3D, el desarrollo multimedia, la creación de software y los videojuegos. Ha participado en proyectos donde integra diseño y programación para el desarrollo de experiencias interactivas, así como en procesos de validación mediante pruebas de usabilidad, investigación de usuarios e iteración continua, orientados a la generación de soluciones funcionales y centradas en el usuario.

Luis Manuel Palma Pinto es Licenciado en Ciencias de la Computación por la Facultad de Matemáticas de la Universidad Autónoma de Yucatán (UADY). Durante su formación ha explorado áreas como la inteligencia artificial, la realidad aumentada y el desarrollo web full-stack. Actualmente enfoca su interés en el fortalecimiento de sus conocimientos en tecnologías backend y servicios en la nube, participando en el desarrollo de soluciones tecnológicas orientadas a entornos modernos y escalables..

Santiago Efrain Itzincab Poot es Licenciado en Ciencias de la Computación por la Facultad de Matemáticas de la Universidad Autónoma de Yucatán (UADY). Su interés se centra en el desarrollo de software de aplicación, el análisis de datos y la ciberseguridad. Ha participado en proyectos de desarrollo de software y aplicaciones educativas, así como en el procesamiento y análisis de información orientados a la generación de soluciones tecnológicas eficientes.

Victor Hugo Menéndez Domínguez es Doctor en Tecnologías Informáticas Avanzadas por la Universidad de Castilla-La Mancha. Es Profesor Titular en la Facultad de Matemáticas de la Universidad Autónoma de Yucatán (UADY), donde desarrolla investigación en las áreas de educación a distancia, ingeniería web y representación del conocimiento, participando en proyectos orientados al desarrollo de soluciones tecnológicas para entornos educativos.

Reyna Faride Peña Castillo Reyna Faride Peña Castillo es Doctora en Ciencias Sociales por la Universidad Autónoma de Yucatán (UADY). Su trabajo se centra en las áreas de Criminología, Victimología, Psicología Jurídica y Forense. Ha participado en el desarrollo de servicios especializados de atención a víctimas en colaboración con instancias gubernamentales, así como en la implementación de protocolos institucionales en materia de prevención y atención de la violencia. Actualmente, colabora en proyectos de política pública orientados a la niñez y adolescencia en el estado de Yucatán.

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