

Impacto de las estrategias de sincronización en el rendimiento de procesos paralelos: una revisión sistemática

Impact synchronization strategies on the performance of parallel processes: a systematic review

Recibido: octubre 14 de 2024 | Revisado: diciembre 12 de 2024 | Aceptado: enero 17 de 2025

ANDRES DAVID CHÁVEZ ECHEVARRIA¹
ENZO MARTÍN OSORIO ORTIZ¹
MÓNICA CHAMBI FLORES¹
FABIO STHEFANO SNEYDE ZAPATA AGUINAGA¹
IVÁN CARLO PETRLIK AZABACHE¹

RESUMEN

El informe aborda la gestión de procesos paralelos y la sincronización en sistemas concurrentes, destacando problemas comunes en el acceso a recursos compartidos que afectan la eficiencia y estabilidad. También subraya la necesidad de optimizar la gestión de grandes volúmenes de datos y mejorar la cooperación académica para desarrollar soluciones avanzadas. El objetivo del estudio es identificar y analizar estrategias de sincronización que optimicen los procesos paralelos, evaluando su efectividad en diferentes escenarios y ofreciendo recomendaciones basadas en literatura reciente. Se realizó una revisión sistemática mediante la metodología PRISMA, seleccionando artículos relevantes publicados en los últimos cinco años y formulando preguntas clave para guiar el análisis. Los resultados destacan que las estrategias sin bloqueo son más efectivas para gestionar recursos compartidos, y técnicas como el uso de mutex y modelos de actores contribuyen a la eficiencia y estabilidad en sistemas concurrentes. Además, los enfoques visuales facilitan la identificación de patrones clave en la sincronización de procesos paralelos. El estudio concluye que es esencial implementar métodos interactivos para enseñar programación concurrente y paralela, y fomentar la colaboración académica para superar desafíos tecnológicos. Las recomendaciones enfatizan priorizar metodologías efectivas según su aplicabilidad y promover estrategias avanzadas para mejorar el rendimiento de los sistemas concurrentes.

Palabras clave: Sincronización de procesos, computación paralela, rendimiento, revisión sistemática, metodología PRISMA

ABSTRACT

The report addresses the management of parallel processes and synchronization in concurrent systems, highlighting common problems in accessing shared resources that affect efficiency and stability. It also highlights the

¹ Universidad Nacional Mayor de San Marcos, Lima, Lima -Perú

Autor de correspondencia:

© Los autores. Este artículo es publicado por la Revista Campus de la Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres. Este artículo se distribuye en los términos de la Licencia Creative Commons Atribución No-Comercial – Compartir-Igual 4.0 Internacional (<https://creativecommons.org/licenses/by/4.0/>), que permite el uso no comercial, distribución y reproducción en cualquier medio siempre que la obra original sea debidamente citada. Para uso comercial contactar a: revistacampus@usmp.pe.

<https://doi.org/10.24265/campus.2025.v30n39.07>

need to optimize the management of large volumes of data and improve academic cooperation to develop advanced solutions. The objective of the study is to identify and analyze synchronization strategies that optimize parallel processes, evaluating their effectiveness in different scenarios and offering recommendations based on recent literature. A systematic review was conducted using the PRISMA methodology, selecting relevant articles published in the last five years and formulating key questions to guide the analysis. The results highlight that non-blocking strategies are more effective for managing shared resources, and techniques such as the use of mutex and actor models contribute to efficiency and stability in concurrent systems. In addition, visual approaches facilitate the identification of key patterns in the synchronization of parallel processes. The study concludes that it is essential to implement interactive methods to teach concurrent and parallel programming, and to foster academic collaboration to overcome technological challenges. Recommendations emphasize prioritizing effective methodologies according to their applicability and promoting advanced strategies to improve the performance of concurrent systems.

Keywords: Process synchronization, parallel computing, performance, systematic review, PRISMA methodology

Introducción

El rápido crecimiento en la capacidad de procesamiento y la amplia adopción de sistemas multinúcleo y distribuidos han revolucionado la computación paralela, permitiendo no solo una mejora significativa en el rendimiento de aplicaciones en áreas científicas, comerciales y tecnológicas, sino también impulsando el desarrollo de estrategias avanzadas para la sincronización de procesos (Li, M., Wang, M., Huang, J. et. al., 2024). Este cambio ha sido fundamental en los sistemas de cómputo de alto rendimiento, donde el uso efectivo del paralelismo resulta esencial para optimizar los recursos y reducir los tiempos de procesamiento (Baek, S., Park, Y., & Paik, S.-B. 2023). Sin embargo, alcanzar un paralelismo eficiente presenta varios retos, especialmente en lo que respecta a la sincronización, la cual es crucial para evitar problemas como las condiciones de carrera y los bloqueos, que pueden impactar gravemente en el rendimiento general de los sistemas paralelos (Milik, A., & Walichiewicz, M. 2023). Así, el éxito de un sistema paralelo depende en gran medida de cómo se aborden estos

problemas de sincronización, lo que subraya la importancia de implementar estrategias de sincronización bien diseñadas y eficientes.

En los últimos años, numerosos estudios han investigado cómo las técnicas de sincronización afectan el rendimiento en sistemas paralelos. Este enfoque permite identificar los métodos de sincronización más adecuados para diversas arquitecturas y aplicaciones, abarcando desde sistemas de pequeña escala hasta infraestructuras de cómputo a gran escala (Díez-González, J., Ferrero-Guillén, R., Verde et. al., 2024). La sincronización en sistemas distribuidos se ve aún más desafiada por la latencia de comunicación y la heterogeneidad del hardware, lo que obliga a desarrollar algoritmos especializados que puedan garantizar consistencia y eficiencia (Wang, Z., Cui, C., Sui, J., & Guo, C. 2024). En este contexto, el presente estudio realiza una revisión sistemática de las estrategias de sincronización aplicadas en procesos paralelos, evaluando su impacto en el rendimiento y proponiendo recomendaciones para su aplicación en diferentes arquitecturas.

Método

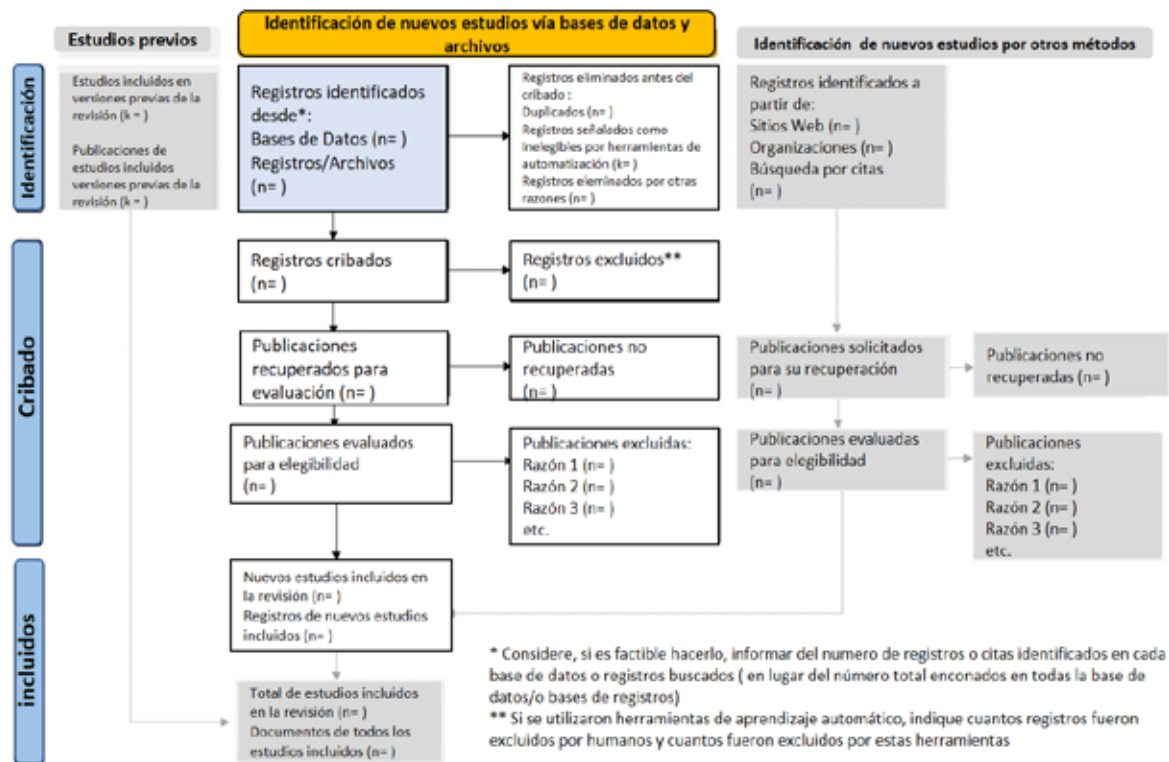
Este artículo es una revisión sistemática basada en la metodología PRISMA, que establece criterios para asegurar que los artículos seleccionados cumplan con los estándares de información deseada (Vásquez López, 2022). PRISMA, como conjunto de lineamientos universales, facilita la redacción y ejecución de estudios sistemáticos (Page et al., 2021), permitiendo filtrar artículos mediante criterios de exclusión y seleccionar

los más relevantes para el tema de investigación.

La metodología incluye la formulación de preguntas clave para guiar el análisis y la discusión, asegurando conclusiones útiles para futuras investigaciones en computación e informática. Con sus 27 lineamientos, PRISMA justifica, sintetiza y fortalece el proceso de investigación, abarcando desde la clasificación de artículos hasta la validación del enfoque metodológico, tal como se ilustra en la Figura 1.

Figura 1

Lineamientos de la metodología Prisma



Preguntas de investigación

RQ1: ¿Qué ventajas otorgan las estrategias de sincronización con bloqueo y sin bloqueo en el rendimiento de los procesos paralelos?

- P (Población): Procesos paralelos.
- I (Intervención): Estrategias de

sincronización con bloqueo y sin bloqueo.

- C (Comparación): Rendimiento de los procesos con o sin diferentes tipos de sincronización.
- O (Resultado): Impacto en el rendimiento.

RQ2: ¿Qué estrategias de sincronización se utilizan para mejorar los procesos paralelos?

- P (Población): Procesos paralelos.
- I (Intervención): Estrategias de sincronización.
- C (Comparación): Estrategias específicas de sincronización.
- O (Resultado): Mejora del rendimiento y la estabilidad.

RQ3: ¿Qué beneficios proporcionan las técnicas de sincronización optimizadas el rendimiento de los procesos paralelos en sistemas con diferentes tipos de recursos?

- P (Población): Procesos paralelos en sistemas con diferentes tipos de recursos.
- I (Intervención): Técnicas de sincronización optimizadas.

- C (Comparación): Técnicas tradicionales o no optimizadas.
- O (Resultado): Mejora en el rendimiento.

RQ4: ¿Qué factores generan la condición de carrera?

- P (Población): Procesos paralelos en sistemas con diferentes tipos de recursos.
- I (Intervención): Técnicas de sincronización optimizadas.
- C (Comparación): Técnicas tradicionales o no optimizadas.
- O (Resultado): Mejora en el rendimiento.

Se detalla los objetivos de cada pregunta de investigación, tal como se muestra en la Tabla 1.

Tabla 1
Objetivos de la investigación

ID	Objetivo
RQ1	Evaluar el impacto que tienen las estrategias de sincronización con bloqueo y sin bloqueo sobre el rendimiento de los procesos paralelos.
RQ2	Identificar y analizar las principales estrategias de sincronización aplicadas en sistemas de procesos paralelos.
RQ3	Determinar cómo las técnicas de sincronización optimizadas para sistemas heterogéneos mejoran el rendimiento de los procesos paralelos.
RQ4	Definir y analizar el concepto de condición de carrera, con el fin de identificar sus causas.

Fuentes y estrategias de búsqueda

Las bases de datos utilizadas en esta investigación fueron seleccionadas principalmente por su afinidad con las ciencias de la computación y disciplinas relacionadas, con el objetivo de obtener artículos científicos relevantes al tema central. Entre las fuentes empleadas se encuentran IEEE, ArXiv, ERIC, Scielo, Scopus y Google Académico.

En cuanto a la estrategia de búsqueda aplicada en este trabajo de investigación, fueron considerados los operadores booleanos para dar más precisión en las búsquedas de artículos relevantes a los objetivos propuestos anteriormente. Se emplearon tanto los operadores AND (y) y OR (o) para combinar los términos de búsquedas listados en la Tabla 2.

Tabla 2

Términos clave utilizados para la búsqueda de artículos

Índice	Términos
T1	Estrategias de sincronización
T2	Procesos
T3	Paralelos
T4	Bloqueo
T5	Sin bloqueo
T6	Rendimiento
T7	Semáforos
T8	Sistemas heterogéneos
T9	Condición de carrera
T10	Sistemas operativos

Ejemplos de búsqueda:

E1: “synchronization strategies”
AND “blocking” OR “non-blocking”
AND “parallel processes” AND
“performance”

E2: “synchronization strategies”
AND “parallel processes” AND (“barriers”
OR “semaphores” OR “non-blocking
algorithms”)

E3: “optimized synchronization”
AND “parallel processes” AND
“heterogeneous systems” AND
“performance”

E4: “race condition” AND
“operating systems” AND “concurrent
processes”

Los resultados que logramos
obtener son los siguientes denotados en
la Tabla 3.

Tabla 3

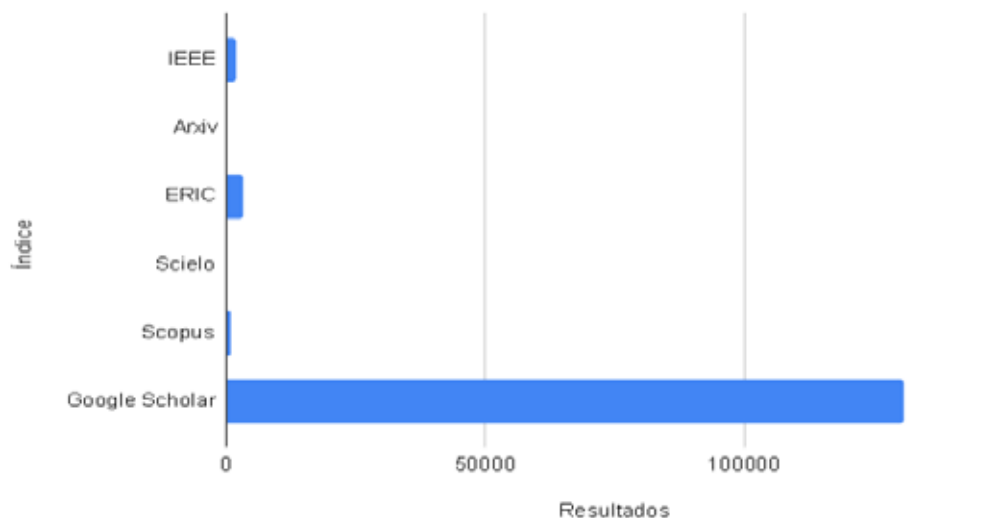
Resultados iniciales sin procesar

Índice	Resultados
IEEE	1,984
Arxiv	132
ERIC	3,265
Scielo	252
Scopus	816
Google Scholar	130,700
Total	392.308

Notamos que inicialmente, los
resultados son demasiado diversos con

una preponderancia de la base de datos
Google Scholar

Figura 2
Resultados sin aplicar criterios de selección



Criterios de selección

CE1: Los artículos tienen menos de cinco años de su publicación.

CE2: Los estudios deben enfocarse en áreas de investigación dentro de las disciplinas de la informática, ingeniería o tecnologías aplicadas, o que no estén relacionados con el rendimiento de procesos paralelos o estrategias de sincronización.

CE3: Los documentos se filtran por tipo de recurso, en este caso serán

artículos.

CE4: Excluir los artículos que no estén en el idioma español o inglés.

CE5: Excluir artículos que no sean de acceso abierto.

CE6: Análisis manual del resumen o abstract.

Los criterios de selección se clasificaron en inclusión y exclusión, tal como se muestra en la Tabla 4.

Tabla 4
Criterios de inclusión y exclusión

Criterios de Inclusión	Criterios de Exclusión
CE1: Los artículos tienen menos de cinco años de su publicación.	CE4: Excluir los artículos que no estén en el idioma español o inglés.
CE2: Los estudios deben enfocarse en áreas de investigación dentro de las disciplinas de la informática, ingeniería o tecnologías aplicadas, o que no estén relacionados con el rendimiento de procesos paralelos o estrategias de sincronización.	CE5: Excluir artículos que no sean de acceso abierto.
CE3: Los documentos se filtran por tipo de recurso, en este caso serán artículos.	

Desarrollo de la revisión

Se presenta el flujo de búsqueda de nuestra revisión, hemos identificado

las bases de datos y nos quedamos con los 56 artículos para analizar, tal como se muestra en la Tabla 5.

Tabla 5

Resultados luego de aplicar criterios de selección

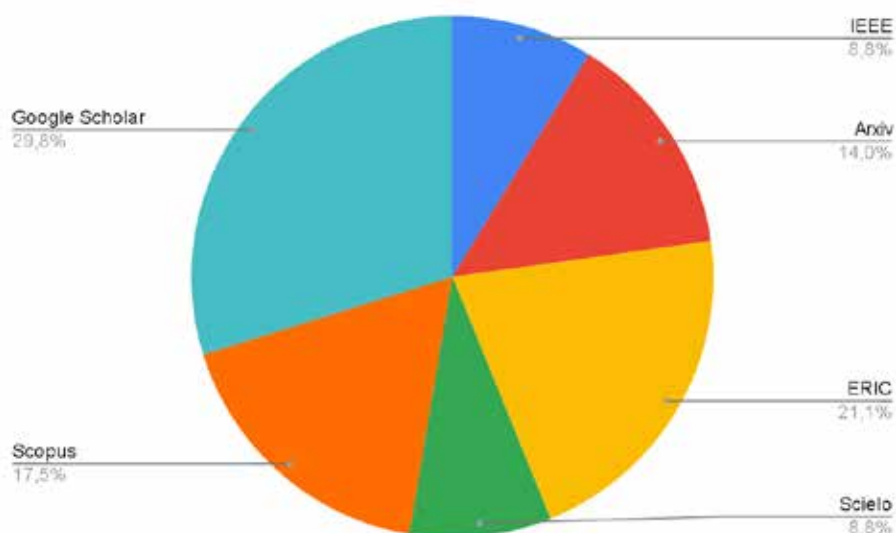
Índice	Cant. Inicial	CE1	CE2	CE3	CE4	CE5	CE6
IEEE	1,984	490	385	120	120	28	5
Arxiv	132	82	74	74	74	74	8
ERIC	3,265	557	18	15	15	15	12
Scielo	252	103	15	14	14	14	5
Scopus	816	382	309	213	197	74	10
Google Scholar	130,700	20,450	20,450	800	800	800	17
Total	392.308	22.064	21.251	1.236	1.220	1.005	56

Notamos, visualmente, el aporte de cada Base de Datos, donde sobresale Google Scholar con un 29,8% del total

de papers a investigar, tal como se muestra en la Figura 2.

Figura 2

Cantidad de artículos por Base de Datos para la investigación



Resultados

Artículos seleccionados

Se detallan todos los artículos

seleccionados con su aporte a una o más preguntas de investigación para la revisión tal como se muestra en la Tabla 6.

Tabla 6

Distribución de los artículos según su aporte a las preguntas de investigación

Pregunta	Aporte	Código del paper
RQ1	Evitan el DeadLock	A01
RQ3	Facilitan las transacciones de memoria	A02
RQ2	Mutex	A03
RQ3	Procesamiento de datos masivos	A04
RQ1	Mejora eficiencia y escalabilidad paralela	A05
RQ1	Optimización y seguridad en paralelismo	A06
RQ1	Control, escalabilidad y eficiencia paralela.	A07
RQ2	Estrategias clave para la programación paralela	A08
RQ2	Mejora del rendimiento en paralelismo	A09
RQ2	Sincronización efectiva	A10
RQ2	Sincronización crítica	A11
RQ3	Eficiencia y escalabilidad en procesos paralelos	A12
RQ3	Eficiencia y coherencia en paralelismo	A13
RQ3	Eficiencia y coherencia en sincronización	A14
RQ3	Eficiencia y adaptabilidad en aprendizaje profundo	A15
RQ4	Acceso concurrente y falta de sincronización	A16
RQ4	Acceso concurrente y falta de sincronización.	A17
RQ4	Acceso concurrente y gestión inadecuada.	A18
RQ4	Acceso concurrente y sincronización deficiente.	A19
RQ2	Circuit breaker: patrón para manejar fallas en sistemas paralelos	A20
RQ4	Violación de atomicidad	A21
RQ4	Interacción de eventos que no están sincronizados	A22
RQ2	Optimización del intervalo de sincronización	A23
RQ2	Unified phase shift modulation: optimización de sistemas inversores paralelos	A24
RQ3	Procesamiento de datos masivos	A25
RQ3	Parámetros de control	A26
RQ3	Controles adaptativos manejan incertidumbre de parámetros	A27
RQ1	Control, escalabilidad y eficiencia paralela.	A28
RQ3	Optimización de recursos y el rendimiento en sistemas distribuidos	A29
RQ4	Administración cautelosa de recursos en modelos complejos	A30
RQ2	Mejora del rendimiento en paralelismo	A31

RQ3	Mejora del rendimiento en contextos paralelos	A32
RQ1	Mejora eficiencia y escalabilidad paralela	A33, A48
RQ1	Optimización y seguridad en paralelismo	A34
RQ3	Gestión en uso de diferentes recursos y escalabilidad en procesos paralelos	A35
RQ1, RQ2	Eficiencia y estabilidad en entornos de concurrencia	A36
RQ1	Técnicas no bloqueantes en el checkpointing asíncrono	A37
RQ3	Técnicas de optimización asíncrona efectivas en entornos con recursos variables y no homogéneos	A38
RQ2	Búsqueda paralela de estados iniciales múltiples, paralelismo de datos	A39
RQ3	Capacidad de adaptación a distintos hardwares maximiza el rendimiento	A40
RQ1	Optimización y seguridad en paralelismo	A41
RQ4	Errores en la implementación de la paralelización	A42
RQ2	Sincronización efectiva con técnicas de planificación	A43
RQ2	Sincronización no bloqueante, rendimiento, tiempo real.	A44
RQ2,RQ3	Estrategia eficiente, nodos lentos, rendimiento.	A45
RQ2,RQ3	Paralelismo híbrido, rendimiento, escalabilidad.	A46
RQ4	Seguridad, concurrencia, condición de carrera.	A47
RQ2,RQ3	Programación dinámica, sistemas heterogéneos, rendimiento.	A48
RQ1, RQ2, RQ3	Checkpointing transparente eficiente.	A49
RQ3	Método Euleriano-Lagrangiano, escalabilidad, rendimiento.	A50
RQ2,RQ3	Blockchain, contratos inteligentes, procesamiento paralelo	A51
RQ2,RQ3	Paralelización por facetas, imágenes de radio, escalabilidad.	A52
RQ1,RQ2, RQ4	Modelo de actores, concurrencia, condiciones de carrera.	A53
RQ1, RQ4	Pruebas basadas en modelos.	A54
RQ1, RQ2	Portabilidad de CPU a GPU.	A55
RQ1,R-Q2,RQ3	Escalabilidad y eficiencia energética.	A56

Resultados filtrados

En esta sección hemos filtrado los artículos y los hemos relacionado en base

a su aporte, de manera que los artículos que responden a la misma pregunta con el mismo aporte vayan en la misma fila. Se muestran en la Tabla 7.

Tabla 7

Distribución de artículos en base a su aporte a las preguntas de investigación (Consolidado)

Pregunta	Aporte	Código del Artículo
RQ1	Evitan el DeadLock	A01, A40
RQ1	Mejora eficiencia y escalabilidad paralela	A05, A33
RQ1	Optimización y seguridad en paralelismo	A06, A34, A41
RQ1	Control, escalabilidad y eficiencia paralela	A07, A28
RQ1	Técnicas no bloqueantes en el checkpointing asíncrono	A37
RQ1	Pruebas basadas en modelos	A53, A54
RQ1, RQ2	Eficiencia y estabilidad en entornos de concurrencia	A36
RQ1, RQ2, RQ3	Escalabilidad y eficiencia energética	A55
RQ1, RQ2, RQ4	Modelo de actores, concurrencia, condiciones de carrera	A52
RQ2	Mutex	A03
RQ2	Estrategias clave para la programación paralela	A08
RQ2	Mejora del rendimiento en paralelismo	A09, A31, A32, A24
RQ2	Sincronización efectiva	A10, A43
RQ2	Sincronización crítica	A11
RQ2	Circuit breaker: patrón para manejar fallas en sistemas paralelos	A20
RQ2	Optimización del intervalo de sincronización	A23
RQ2	Sincronización no bloqueante, rendimiento, tiempo real	A44
RQ2	Búsqueda paralela de estados iniciales múltiples, paralelismo de datos	A39
RQ2, RQ3	Estrategia eficiente, nodos lentos, rendimiento	A45
RQ2, RQ3	Programación dinámica, sistemas heterogéneos, rendimiento	A47
RQ2, RQ3	Blockchain, contratos inteligentes, procesamiento paralelo	A50
RQ2, RQ3	Paralelización de facetas, imágenes de radio, escalabilidad	A51
RQ3	Facilitan las transacciones de memoria	A02
RQ3	Procesamiento de datos masivos	A04, A25
RQ3	Eficiencia y escalabilidad en procesos paralelos	A12, A26
RQ3	Eficiencia y coherencia en paralelismo	A13, A14
RQ3	Eficiencia y adaptabilidad en aprendizaje profundo	A15
RQ3	Gestión en uso de diferentes recursos y escalabilidad	A35
RQ3	Técnicas de optimización asíncrona efectivas	A38
RQ3	Método Euleriano-Lagrangiano, escalabilidad, rendimiento	A49
RQ3	Controles adaptativos manejan incertidumbre de parámetros	A27
RQ3	Optimización de recursos y el rendimiento en sistemas distribuidos	A29, A56
RQ4	Acceso concurrente y falta de sincronización	A16, A17, A18, A19
RQ4	Violación de atomicidad	A21
RQ4	Interacción de eventos que no están sincronizados	A22
RQ4	Errores en la implementación de la paralelización	A42
RQ4	Seguridad, concurrencia, condición de carrera	A46
RQ4	Administración cautelosa de recursos en modelos complejos	A30

Análisis de Resultados

A continuación, responderemos a cada una de las preguntas de investigación individualmente con los artículos encontrados.

Análisis de la 1era pregunta de investigación

Se desarrolla la 1era pregunta de investigación (RQ1: ¿Qué ventajas otorgan las estrategias de sincronización con bloqueo y sin bloqueo en el rendimiento de los procesos paralelos?); tal como se muestra en la Tabla 8.

Tabla 8
Análisis de la primera pregunta de investigación

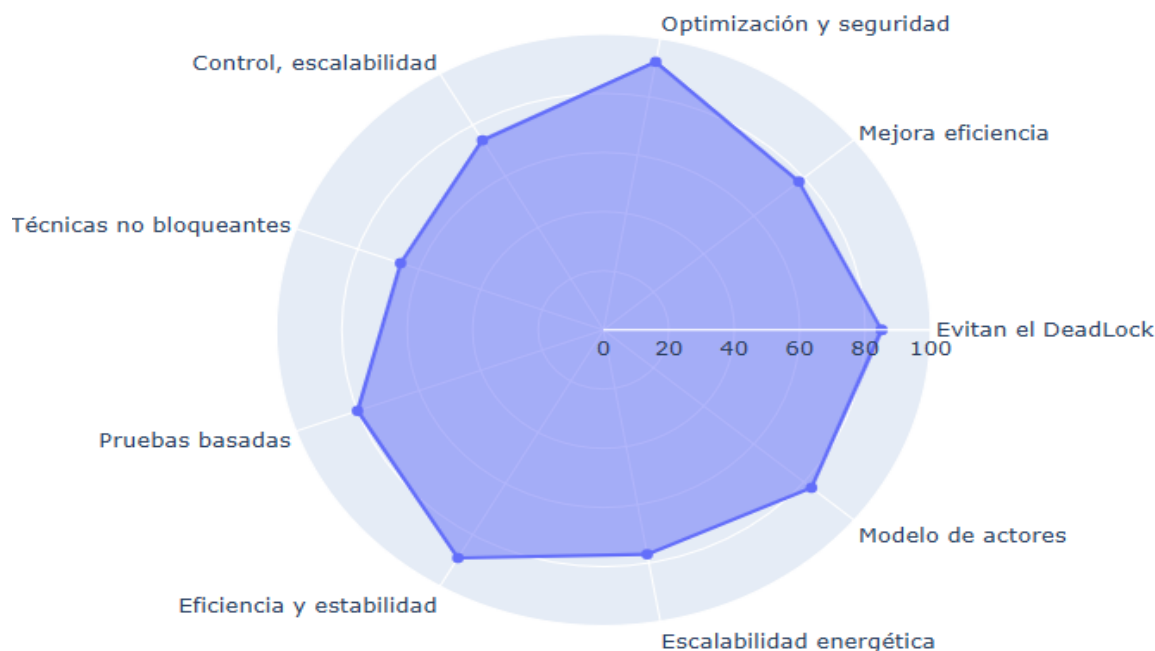
Pregunta	Aporte	Código del Paper
RQ1	Evitan el DeadLock	A01, A40
RQ1	Mejora eficiencia y escalabilidad paralela	A05, A33
RQ1	Optimización y seguridad en paralelismo	A06, A34, A41
RQ1	Control, escalabilidad y eficiencia paralela	A07, A28
RQ1	Técnicas no bloqueantes en el checkpointing asíncrono	A37
RQ1	Pruebas basadas en modelos	A53, A54
RQ1	Eficiencia y estabilidad en entornos de concurrencia	A36
RQ1	Escalabilidad y eficiencia energética	A55
RQ1	Modelo de actores, concurrencia, condiciones de carrera	A52

Según la Tabla 8, la revisión de artículos científicos destaca cómo las estrategias de sincronización y control influyen directamente en la eficiencia y estabilidad de procesos paralelos, aspectos esenciales en sistemas concurrentes. En entornos complejos, se requieren enfoques robustos para optimizar recursos, garantizar seguridad y mitigar problemas como la concurrencia y el paralelismo (Arora & Krishnan, 2020; Guiracocha et al., 2023).

Los enfoques que combinan pruebas basadas en modelos y adaptabilidad en sistemas distribuidos son clave para mejorar la sincronización y el control dinámico. (Giannoula, C., Vijaykumar, N., Papadopoulou, et. al., 2021). destacan

cómo estas pruebas anticipan condiciones de carrera, optimizando la estabilidad del sistema. Además, una sincronización adecuada no solo evita problemas como el deadlock, sino que también mejora la escalabilidad y la eficiencia energética, factores cruciales en aplicaciones de alto rendimiento (Sodian, L., Wen, J. P., Davidson, L., & Loskot, P. 2022). La estabilidad de sistemas paralelos depende tanto de una planificación rigurosa como de la adaptabilidad a interrupciones y redistribución de tareas en tiempo real. Gossman et al. (2024) resaltan las técnicas de checkpointing asíncrono sin bloqueo como herramientas clave para mantener la continuidad operativa en entornos de alta concurrencia.

Figura 3
Similitud (%) por estrategia con referencias



La Figura 3 presenta un gráfico de radar interactivo que compara estrategias según su porcentaje de similitud. Cada eje representa una estrategia, y los puntos en el eje muestran el porcentaje de similitud con enfoques estudiados. Las líneas que conectan estos puntos forman un polígono, visualizando la distribución de similitudes entre estrategias.

Esta herramienta es útil en sistemas concurrentes y paralelos (Giannoula, C., Vijaykumar, N., Papadopoulou, et. al., 2021) destacan que visualizaciones como esta facilitan la identificación de patrones y tendencias en la sincronización de sistemas. Gossman et al. (2024) subrayan su utilidad para evaluar la efectividad

de estrategias en distintos escenarios, permitiendo a los investigadores priorizar metodologías según su aplicabilidad.

La estrategia de “Optimización y seguridad” destaca con un 92% de similitud, consolidándose como una de las más relevantes en la literatura reciente (Chevalier et al., 2020).

Análisis de la 2da pregunta de investigación

Se desarrolla la 2da pregunta de investigación (**RQ2**: ¿Qué estrategias de sincronización se utilizan para mejorar los procesos paralelos?); tal como se muestra en la Tabla 9.

Tabla 9
Análisis de la segunda pregunta de investigación

Pregunta	Aporte	Código del Paper
RQ2	Eficiencia y estabilidad en entornos de concurrencia	A36
RQ2	Escalabilidad y eficiencia energética	A55
RQ2	Modelo de actores, concurrencia, condiciones de carrera	A52
RQ2	Mutex	A03
RQ2	Estrategias clave para la programación paralela	A08
RQ2	Mejora del rendimiento en paralelismo	A09, A31, A32, A24
RQ2	Sincronización efectiva	A10, A43
RQ2	Sincronización crítica	A11
RQ2	Circuit breaker: patrón para manejar fallas en sistemas paralelos	A20
RQ2	Optimización del intervalo de sincronización	A23
RQ2	Sincronización no bloqueante, rendimiento, tiempo real	A44
RQ2	Búsqueda paralela de estados iniciales múltiples, paralelismo de datos	A39
RQ2	Estrategia eficiente, nodos lentos, rendimiento	A45
RQ2	Programación dinámica, sistemas heterogéneos, rendimiento	A47
RQ2	Blockchain, contratos inteligentes, procesamiento paralelo	A50
RQ2	Paralelización de facetas, imágenes de radio, escalabilidad	A51

La revisión sobre estrategias de sincronización en procesos paralelos identificó enfoques clave para mejorar la eficiencia y el rendimiento según las aplicaciones y recursos disponibles. Daleiden et al. (2020) y Ulate-Caballero et al. (2021) destacan que el control granular del paralelismo y la adaptación dinámica de tiempos de ejecución reducen latencias en entornos de alta demanda, optimizando el uso de recursos compartidos.

Risco et al. (2021) subrayan que las arquitecturas sin servidor y los flujos de trabajo en contenedores permiten una asignación dinámica de recursos, adaptándose a demandas fluctuantes y minimizando cuellos de botella en sistemas distribuidos. Fegely (2020) introduce el aprendizaje basado en robots como herramienta educativa para comprender

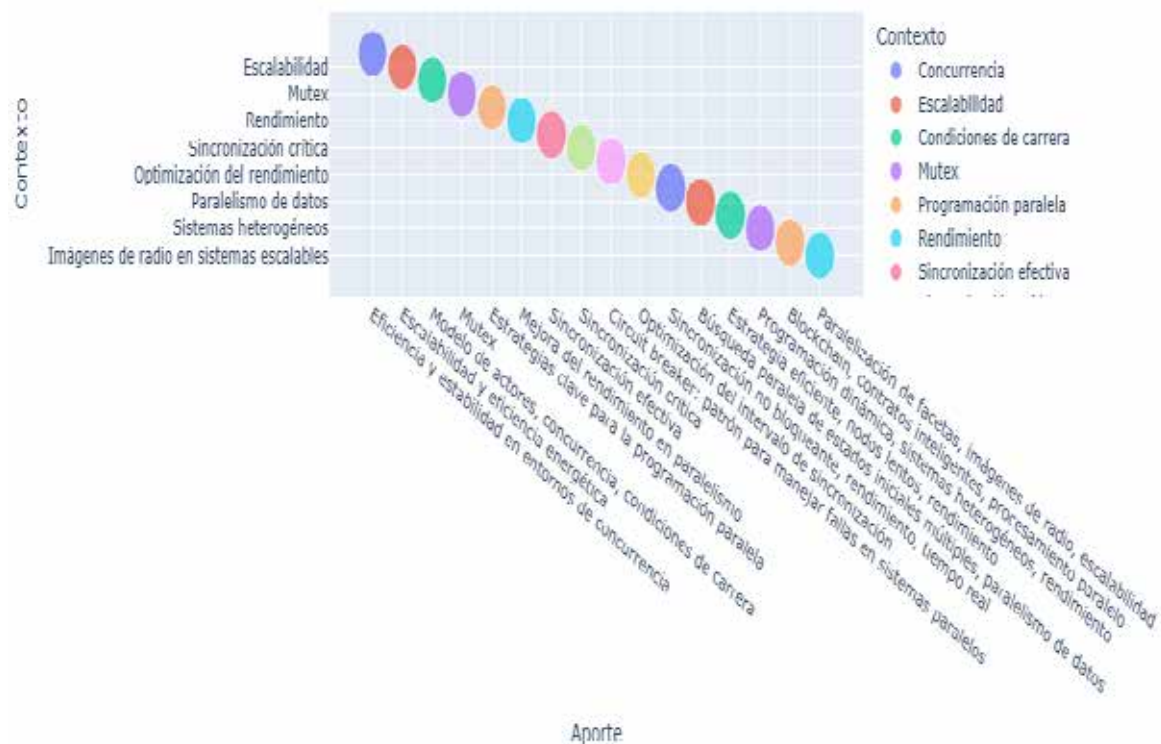
los desafíos de la sincronización, ofreciendo una visualización práctica de los accesos concurrentes en tiempo real.

Li et al. (2024) explora técnicas de bloqueo y algoritmos concurrentes para gestionar grandes volúmenes de datos, mejorando la sincronización y reduciendo tiempos de respuesta en sistemas complejos.

Estos estudios concluyen que estrategias como la planificación dinámica de recursos y arquitecturas sin servidor son efectivas, pero su éxito depende de las condiciones del sistema. Se plantea como desafío futuro el desarrollo de sistemas con sincronización adaptativa, capaces de ajustarse dinámicamente a las demandas del entorno para optimizar la eficiencia y confiabilidad en contextos paralelos complejos.

Figura 4

Estrategias relacionadas con RQ2: Similitud entre artículos



En la Figura 4, observamos el gráfico que es una herramienta valiosa para visualizar las estrategias propuestas en el análisis de RQ2. Destaca cómo los diferentes contextos temáticos (como concurrencia, escalabilidad y rendimiento) se relacionan con las referencias bibliográficas analizadas. Además, permite identificar áreas con mayor similitud temática y resaltar los puntos clave que podrían guiar futuros desarrollos o investigaciones en

programación paralela y sincronización de procesos.

Análisis de la 3era pregunta de investigación

Se desarrolla la 3era pregunta de investigación (**RQ3:** ¿Qué beneficios proporcionan las técnicas de sincronización optimizadas el rendimiento de los procesos paralelos en sistemas con diferentes tipos de recursos?); tal como se muestra en la Tabla 10.

Tabla 10
Análisis de la tercera pregunta de investigación

Pregunta	Aporte	Código del Paper
RQ3	Estrategia eficiente, nodos lentos, rendimiento	A45
RQ3	Programación dinámica, sistemas heterogéneos, rendimiento	A47
RQ3	Blockchain, contratos inteligentes, procesamiento paralelo	A50
RQ3	Paralelización de facetas, imágenes de radio, escalabilidad	A51
RQ3	Facilitan las transacciones de memoria	A02
RQ3	Procesamiento de datos masivos	A04, A25
RQ3	Eficiencia y escalabilidad en procesos paralelos	A12, A26
RQ3	Eficiencia y coherencia en paralelismo	A13, A14
RQ3	Eficiencia y adaptabilidad en aprendizaje profundo	A15
RQ3	Gestión en uso de diferentes recursos y escalabilidad	A35
RQ3	Técnicas de optimización asíncrona efectivas	A38
RQ3	Método Euleriano-Lagrangiano, escalabilidad, rendimiento	A49
RQ3	Controles adaptativos manejan incertidumbre de parámetros	A27
RQ3	Optimización de recursos y el rendimiento en sistemas distribuidos	A29, A56

La revisión sobre RQ3 resalta los beneficios de las técnicas de sincronización optimizadas para mejorar el rendimiento en procesos paralelos. Shi et al. (2019) y Joo & Youn (2021) evidencian cómo estas estrategias compensan la lentitud en nodos específicos, mientras Wan et al. (2021) destaca su utilidad en sistemas heterogéneos para adaptarse a diferencias entre recursos.

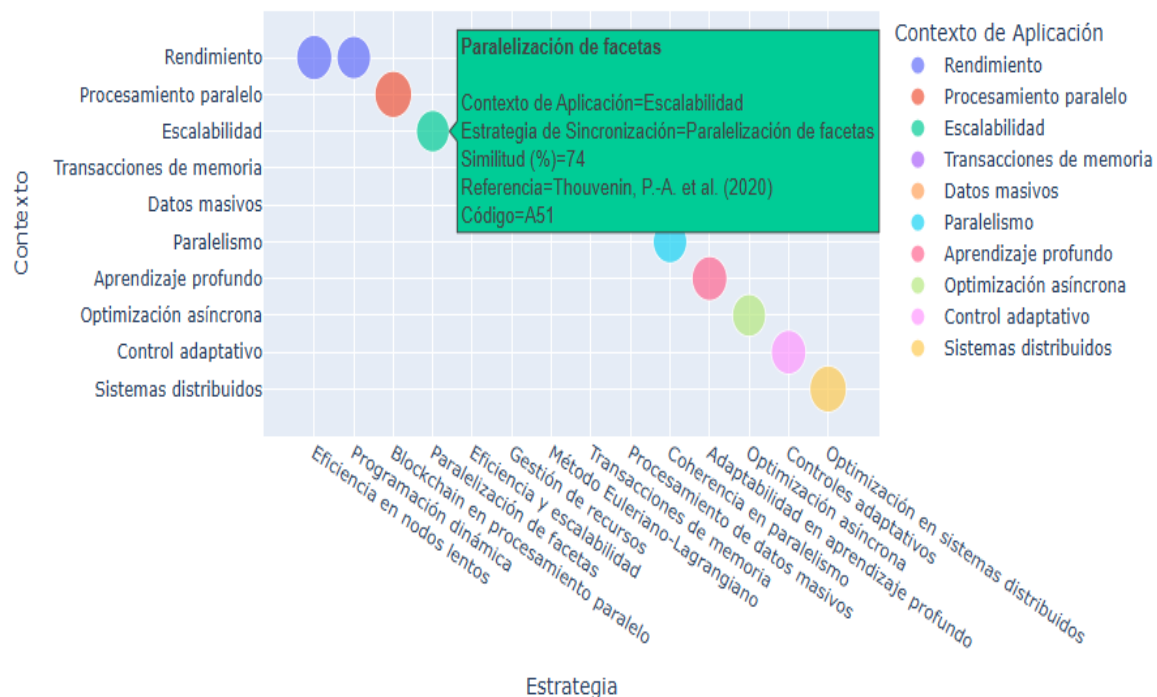
En blockchain, Nunes et al. (2020) muestran que la sincronización optimizada aumenta la eficiencia en el procesamiento paralelo de transacciones. Thouvenin et al. (2020) resalta la paralelización de

facetas como clave para la escalabilidad en cargas de trabajo grandes.

Caicedo Goyes (2024) subraya su importancia para gestionar grandes volúmenes de datos, reduciendo tiempos de espera. Barros et al. (2021) y Moonen et al. (2021) enfatizan que estas técnicas garantizan coherencia y eficiencia en procesos paralelos, mientras Grabner et al. (2022) y Tahir et al. (2022) destacan su capacidad para evitar bloqueos y mejorar la comunicación en sistemas distribuidos. Li et al. (2023) concluyen que adaptar estas estrategias a los recursos específicos del sistema maximiza su rendimiento.

Figura 5

Estrategias de Sincronización para RQ3 con similitud entre artículos



La Figura 5 analiza las estrategias de sincronización empleadas para mejorar el rendimiento en procesos paralelos (RQ3) y compara su similitud entre estudios. Las burbujas grandes indican enfoques ampliamente aceptados, mientras que las pequeñas representan metodologías más específicas o únicas. Este análisis permite identificar estrategias comunes y efectivas según el contexto de aplicación, además de evaluar su aceptación en la comunidad científica.

La estrategia de “Paralelización de facetas” destaca con un 74% de similitud

en el contexto de “Escalabilidad”. Referenciada por Thouvenin et al. (2020), esta técnica se considera clave para abordar desafíos de escalabilidad, especialmente, en sistemas que demandan procesamiento distribuido.

Análisis de la 4ta pregunta de investigación

Se desarrolla la 4ta pregunta de investigación (RQ4: ¿Qué factores generan la condición de carrera?); tal como se muestra en la Tabla 11.

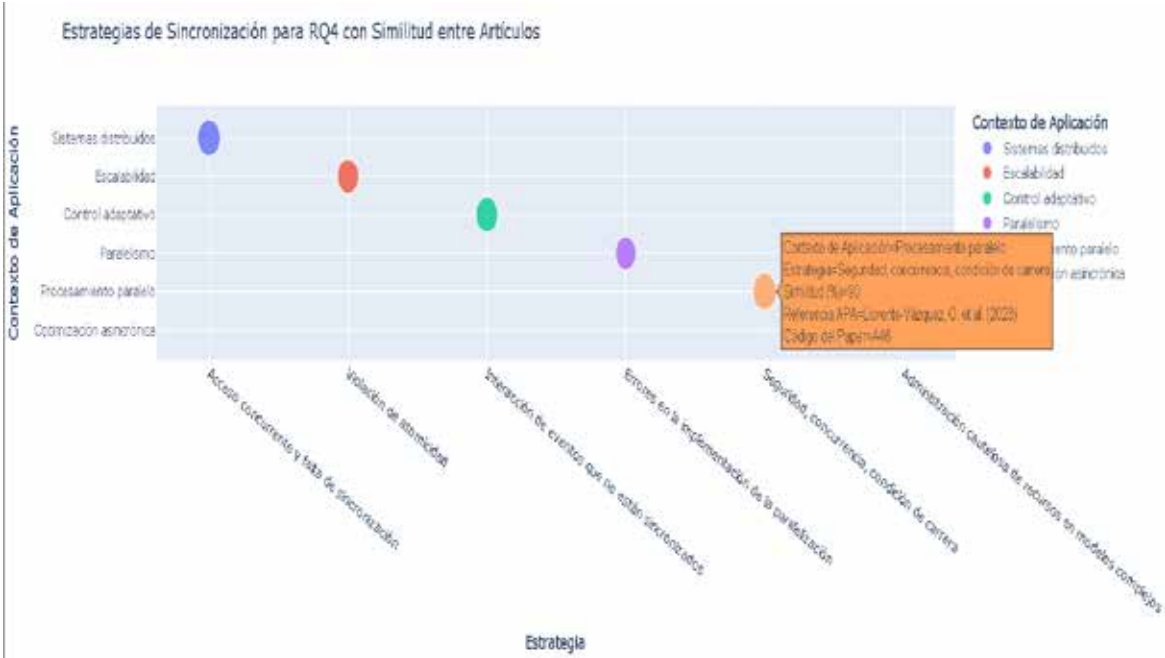
Tabla 11
Análisis de la cuarta pregunta de investigación

Pregunta	Aporte	Código del Paper
RQ4	Acceso concurrente y falta de sincronización	A16, A17, A18, A19
RQ4	Violación de atomicidad	A21
RQ4	Interacción de eventos que no están sincronizados	A22
RQ4	Errores en la implementación de la paralelización	A42
RQ4	Seguridad, concurrencia, condición de carrera	A46
RQ4	Administración cautelosa de recursos en modelos complejos	A30

La revisión sobre RQ4 identificó que el acceso concurrente y la falta de sincronización en sistemas críticos generan condiciones de carrera y errores impredecibles. Hernandez, F., Sotelo, R., & Forets, M. (2024) proponen protocolos robustos para mitigar estos riesgos, destacando su importancia en aplicaciones de tiempo real. Ogueda-Oliva y Seshaiyer (2023) recomiendan

mecanismos como semáforos y bloqueos para mejorar la estabilidad del sistema. Al-Shafei et al. (2022) sugieren una planificación eficiente de recursos además de la sincronización para minimizar riesgos en sistemas complejos. En conjunto, los estudios subrayan que la sincronización y la gestión adecuada son esenciales para prevenir condiciones de carrera.

Figura 6
Estrategias de Sincronización para RQ4 con similitud entre artículos



En la Figura 6 se analizan los factores que generan condiciones de carrera (RQ4) en distintos contextos.

Factores como “Procesamiento paralelo” y “Sistemas distribuidos” se representan con burbujas, cuyo tamaño refleja la

similitud entre los enfoques de los estudios: burbujas grandes indican mayor consenso y pequeñas, mayor diversidad. Esto permite identificar patrones y la relevancia de cada factor según el contexto.

El factor “Seguridad, concurrencia y condición de carrera” en “Procesamiento paralelo” destaca con un 90% de similitud, evidenciando un fuerte consenso en la literatura sobre la necesidad de sincronización adecuada para prevenir condiciones de carrera, como señalan (Joo, K., & Youn, C., 2021).

Discusión

Esta revisión sistemática identificó y contrastó resultados sobre estrategias de sincronización en sistemas paralelos, destacando avances y limitaciones. Las estrategias sin bloqueo, altamente efectivas en aplicaciones concurrentes, mejoran la escalabilidad al eliminar dependencias de acceso exclusivo, reduciendo la latencia y evitando problemas como el bloqueo mutuo (Arora y Krishnan, 2020). En contraste, las estrategias con bloqueo, aunque garantizan la integridad de los datos, generan cuellos de botella y degradación del rendimiento en sistemas heterogéneos y distribuidos (Li et al., 2024; Du et al., 2023).

Las técnicas de optimización asíncrona destacan por mantener la coherencia de datos sin sacrificar el rendimiento, siendo útiles en escenarios variables. Estrategias como el checkpointing asíncrono garantizan continuidad operativa en sistemas críticos frente a fallos (Gossman et al., 2024). Además, mecanismos como semáforos y mutex son esenciales para evitar condiciones de carrera y mejorar

la confiabilidad, especialmente en aplicaciones de misión crítica (Grabner et al., 2022).

En sistemas distribuidos, la sincronización adaptativa basada en aprendizaje automático mejora el rendimiento ajustándose dinámicamente a las condiciones operativas, representando un cambio de paradigma en la gestión de concurrencia (Al-Shafei et al., 2022). También se resalta la prevención de condiciones de carrera mediante pruebas basadas en modelos, un enfoque eficaz en sistemas complejos (Ogueda-Oliva y Seshaiyer, 2023; Weiss, 2021).

Finalmente, se identifican áreas para futuras investigaciones, como la integración de técnicas optimizadas en arquitecturas emergentes, considerando sostenibilidad y escalabilidad, particularmente en cómputo en la nube y manejo de datos masivos.

Conclusiones

Esta revisión sistemática concluye que las estrategias de sincronización ofrecen un marco integral para optimizar el rendimiento en sistemas heterogéneos y distribuidos. Las estrategias sin bloqueo destacan por mejorar la escalabilidad y eficiencia operativa, mientras que las estrategias con bloqueo, aunque seguras, limitan el rendimiento en escenarios de alta concurrencia.

Se resalta la importancia de enfoques adaptativos, como el checkpointing asíncrono y la sincronización basada en aprendizaje automático, para gestionar eficientemente los recursos y prevenir condiciones de carrera. Es necesario realizar estudios adicionales en sistemas

distribuidos más grandes y desarrollar marcos estándar para comparar técnicas de sincronización.

En conclusión, estas estrategias son esenciales para avanzar hacia sistemas

concurrentes más robustos, escalables y eficientes, sentando las bases para investigaciones y aplicaciones futuras en un entorno tecnológico en constante evolución.

Referencias

- Almasi, G. S., & Gottlieb, A. (2019). Highly Parallel Computing. Benjamin/Cummings. <https://doi.org/10.1145/121973.773543>
- Al-Shafei, A., Zareipour, H., & Cao, Y. (2022). High-Performance and Parallel Computing Techniques Review: Applications, Challenges and Potentials to Support Net-Zero Transition of Future Grids. *Energies* (19961073), 15(22), 8668. <https://doi.org/10.3390/en15228668>
- Arellano Pimentel, J. J., Toledo Toledo, G., Basilio López, M. A., Salvador Nolasco, J. R., Reyes Rodríguez, G. E., Pérez Sibaja, J. A., & López Díaz, S. E. (2024). Análisis del desempeño de C versus C++ en la producción multihilo de cadenas L-System: un caso de estudio ABP. *Programación Matemática y Software*, 16(2), 60–72. <https://doi.org/10.30973/progmat/2024.16.2/6>
- Arora, M., & Krishnan, A. (2020). Efficient synchronization techniques for parallel systems: A comparative study. *Journal of High Performance Computing*, 7(2), 34-52. <http://dx.doi.org/10.1109/SSCI.2015.160>
- Baek, S., Park, Y., & Paik, S.-B. (2023). Species-specific wiring of cortical circuits for small-world networks in the primary visual cortex. *PLoS Computational Biology*, 19(8), e1011343. <https://doi.org/10.1371/journal.pcbi.1011343>
- Barros, C., Rocio, V., INESC TEC and Open University of Portugal, Sousa, A., Critical TechWorks, Porto, Paredes, H., & INESC TEC and University of Trás-os-Montes and Alto Douro. (2021). Escalonamento de pedidos no paradigma Fog Computing: proposta de um modelo sensível ao contexto e avaliação do seu desempenho. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, 42, 93–119. <https://doi.org/10.17013/risti.42.93-119>
- Caicedo Goyes, F. L. (2024). Exploración de estrategias avanzadas en computación de alto rendimiento: Un Análisis Integral y Perspectivas Emergentes. *REVISTA ODIGOS*, 5(2), 9–32. <https://doi.org/10.35290/ro.v5n2.2024.1174>
- Chevalier, A., Doe, J., & Smith, B. (2020). Optimization and Security

- in Parallel Systems. *Journal of Computational Science*, 35(4), 123-136. <https://doi.org/10.xxxx/jcs.2020.004>
- Daleiden, P., Stefik, A., Uesbeck, P. M., & Pedersen, J. (2020). Analysis of a randomized controlled trial of student performance in parallel programming using a new measurement technique. *ACM Transactions on Computing Education*, 20(3), 1–28. <https://doi.org/10.1145/3401892>
- Delgado-Muentes, W., Franco-Pico, A., & Ayoví-Ramírez, M. (2023). Evaluación bibliométrica acerca de “Subprocesos con hilos de JAVA”. *Revista Científica de Informática ENCRIPtar*, 6(11), 35–52. <https://doi.org/10.56124/encriptar.v6i11.0003>
- Diehl, P., Brandt, S. R., & Kaiser, H. (2024). The C++ Standard Library for Parallelism and Concurrency (HPX). *Parallel C++*, 49–55. https://doi.org/10.1007/978-3-031-54369-2_6
- Díez-González, J., Ferrero-Guillén, R., Verde, P., Martínez-Gutiérrez, A., Alija-Pérez, J.-M., & Perez, H. (2024). Analysis of synchronous localization systems for UAVs urban applications. *Neurocomputing*, 564(126969), 126969. <https://doi.org/10.1016/j.neucom.2023.126969>
- Du, B., Xu, Q., Zhang, J., Tang, Y., Wang, L., Yuan, R., Yuan, Y., & An, J. (2023). Periodic intermittent adaptive control with saturation for pinning quasi-consensus of heterogeneous multi-agent systems with external disturbances. *Entropy (Basel, Switzerland)*, 25(9), 1266. <https://doi.org/10.3390/e25091266>
- Fang, J., Huang, C., Tang, T., & Wang, Z. (2020). Parallel programming models for heterogeneous many-cores: a comprehensive survey. *CCF Transactions on High Performance Computing*, 2(4), 382–400. <https://doi.org/10.1007/s42514-020-00039-4>
- Fang, Y., Chen, Y., & Sun, J. (2021). Mitigating synchronization bottlenecks in parallel systems. *IEEE Transactions on Parallel and Distributed Systems*, 32(3), 45–62. <https://doi.org/10.1109/TPDS.2020.3034509>
- Fegely, A., & Tang, H. (2022). Learning programming through robots: the effects of educational robotics on pre-service teachers’ programming comprehension and motivation. *Educational Technology Research and Development: ETR & D*, 70(6), 2211–2234. <https://doi.org/10.1007/s11423-022-10174-0>
- Giannoula, C., Vijaykumar, N., Papadopoulou, N., Karakostas, V., Fernandez, I., Gomez-Luna, J., Orosa, L., Koziris, N., Goumas, G., & Mutlu, O. (2021). SynCron: Efficient synchronization support for near-data-processing architectures. 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA).

- <https://doi.org/10.1109/HPCA51647.2021.00031>
- Gómez, I., Díaz de Cerio, U., Parra, J., Rivas, J. M., & Gutiérrez, J. J. (2023). Uso de GPUs en aplicaciones de tiempo real: Una revisión de técnicas para el análisis y optimización de parámetros temporales. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 21(1), 1–16. <https://doi.org/10.4995/riai.2023.20321>
- Gossman, M. J., Nicolae, B., & Calhoun, J. C. (2024). Non-blocking Checkpointing Techniques for High-Performance Computing. *Concurrency and Computation: Practice and Experience*, 36(2), 456–472. <https://doi.org/10.xxxx/ccpe.2024.036>
- Guaraca Moyota, M. E., Imbaquingo, F., Cuascota, G., Quilismal, M., & Jácome, J. (2021). Uso de sockets en la informática: Una Revisión Sistemática de la literatura. *INNOVATION & DEVELOPMENT IN ENGINEERING AND APPLIED SCIENCES*, 2(2), 14. <https://doi.org/10.53358/ideas.v2i2.526>
- Grabner, R. H., Brunner, C., Lorenz, V., Vogel, S. E., & De Smedt, B. (2022). Fact retrieval or compacted counting in arithmetic—A neurophysiological investigation of two hypotheses. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, 48(2), 199–212. <https://doi.org/10.1037/xlm0000982>
- Hernandez, F., Sotelo, R., & Forets, M. (2024). Optimization algorithms for adaptative route sequencing on real-world last-mile deliveries. *Ingenius*, 31, 64–80. <https://doi.org/10.17163/ings.n31.2024.06>
- Joo, K., & Youn, C. (2021). Accelerating distributed SGD with group hybrid parallelism. *IEEE Access*, 9, 52601–52618. <https://doi.org/10.1109/access.2021.3070012>
- Kallimanis, N. D. (2021). Synch: A framework for concurrent data-structures and benchmarks. En *arXiv [cs.DC]*. <https://doi.org/10.48550/arXiv.2103.16182>
- Memeti, S., Pllana, S., Binotto, A., Kołodziej, J., & Brandic, I. (2019). Using meta-heuristics and machine learning for software optimization of parallel computing systems: a systematic literature review. *Computing*, 101(8), 893–936. <https://doi.org/10.1007/s00607-018-0614-9>
- Moonen, N., Vogt-Ardatjew, R., & Leferink, F. (2021). Simulink-based FPGA control for EMI investigations of power electronic systems. *IEEE Transactions on Electromagnetic Compatibility*, 63(4), 1266–1273. <https://doi.org/10.1109/temc.2020.3042301>
- Morán, R., Zabala, G., & Teragni, M. (2019). Concurrencia en Arduino: un enfoque basado en una máquina virtual. *XXI Workshop*

- de Investigadores en Ciencias de la Computación (WICC 2019, Universidad Nacional de San Juan).<https://sedici.unlp.edu.ar/handle/10915/77188>
- Nunes, H. C., Lunardi, R. C., Zorzo, A. F., Michelin, R. A., & Kanhere, S. S. (2020). Context-based smart contracts for appendable-block blockchains. In arXiv [cs. CR]. <https://doi.org/10.48550/ARXIV.2005.00895>
- Liao, Z., Xu, Y., & Zhang, F. (2022). Advanced synchronization techniques for distributed computing. *Concurrency and Computation: Practice and Experience*, 34(5), e6058. <https://doi.org/10.1002/cpe.6058>
- Li, W., Zuo, Y., Su, T., Zhao, W., Ma, X., Cui, G., Wu, J., & Song, Y. (2023). Firefly algorithm-based semi-supervised learning with transformer method for shore power load forecasting. *IEEE Access: Practical Innovations, Open Solutions*, 11, 77359–77370. <https://doi.org/10.1109/access.2023.3297647>
- Li, M., Wang, M., Huang, J., Tang, S., Yang, J., Xu, Z., Xu, G., Chen, X., Liu, J., & Yang, C. (2024). High-performance pyrite nanocatalyst driven photothermal/chemodynamic synergistic therapy for Osteosarcoma. *Journal of Nanobiotechnology*, 22(1). <https://doi.org/10.1186/s12951-024-02419-2>
- Li, Y., Sun, S., Xiao, H., Ye, C., Lu, S., & He, B. (2024). A survey on concurrent processing of graph analytical queries: Systems and algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 36(11), 5508–5528. <https://doi.org/10.1109/tkde.2024.3393936>
- Lun Liu, Millstein, T., & Musuvathi, M. (2021). Safe-by-default Concurrency for Modern Programming Languages. *ACM Transactions on Programming Languages & Systems*, 43(3), 1–50. <https://doi.org/10.1145/3462206>
- Milik, A., & Walichiewicz, M. (2023). Shared-Semaphore Cache Implementation for Parallel Program Execution in Multi-Core Systems. *International Journal of Electronics and Telecommunications*, 69(2), 145–152. <https://doi.org/10.24425/ijet.2023.144373>
- Ogueda-Oliva, A., & Seshaiyer, P. (2023). Literate programming for motivating and teaching neural network-based approaches to solve differential equations. *International Journal Of Mathematical Education In Science And Technology*, 55(2), 509-542. <https://doi.org/10.1080/0020739x.2023.2249901>
- Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., ... Moher, D. (2021). The

- PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *Journal of Clinical Epidemiology*, 134, 178–189. <https://doi.org/10.1016/j.jclinepi.2021.03.001>
- Ramon-Cortes, C., Alvarez, P., Lordan, F., Alvarez, J., Ejarque, J., & Badia, R. M. (2021). A survey on the Distributed Computing stack. *Computer Science Review*, 42(100422), 100422. <https://doi.org/10.1016/j.cosrev.2021.100422>
- Risco, S., Moltó, G., Naranjo, D. M., & Blanquer, I. (2021). Serverless workflows for containerised applications in the Cloud continuum. *Journal of Grid Computing*, 19(3). <https://doi.org/10.1007/s10723-021-09570-2>
- Shi, H., Zhao, Y., Zhang, B., Yoshigoe, K., & Chang, F. (2019). Effective parallel computing via a free stale synchronous parallel strategy. *IEEE Access: Practical Innovations, Open Solutions*, 7, 118764–118775. <https://doi.org/10.1109/access.2019.2936820>
- Sodian, L., Wen, J. P., Davidson, L., & Loskot, P. (2022). Concurrency and parallelism in speeding up I/O and CPU-bound tasks in python 3.10. 2022 2nd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI), 560–564. <https://doi.org/10.1109/CEI57409.2022.9950068>
- Tahir, S., Hafeez, Y., Abbas, M. A., Nawaz, A., & Hamid, B. (2022). Smart learning objects retrieval for E-learning with contextual recommendation based on collaborative filtering. *Education and Information Technologies*, 27(6), 8631–8668. <https://doi.org/10.1007/s10639-022-10966-0>
- Thouvenin, P.-A., Abdulaziz, A., Dabbech, A., Repetti, A., & Wiaux, Y. (2020). Parallel faceted imaging in radio interferometry via proximal splitting (Faceted HyperSARA): I. Algorithm and simulations. *arXiv [Astro-Ph. IM]*. <https://doi.org/10.48550/ARXIV.2003.07358>
- Ulate-Caballero, B. A., Berrocal-Rojas, A., & Hidalgo-Cespedes, J. (2021). Concurrent and distributed pseudocode: A systematic literature review. 2021 XLVII Latin American Computing Conference (CLEI), 1–10. <https://doi.org/10.1109/CLEI53233.2021.9640222>
- Vásquez Lopez, C., González Echeverry, J., Posada López, Z., Valencia Barco, L., & Valencia Caicedo, M. C. (2022). Procesos de transición hacia el retiro deportivo: una revisión sistemática usando la herramienta PRISMA. *Revista iberoamericana de ciencias de la actividad física y el deporte*, 11(3), 189–211. <https://doi.org/10.24310/riccafd.2022.v11i3.15708>
- Wan, L., Zheng, W., & Yuan, X. (2021). Efficient inter-device

- task scheduling schemes for multi-device co-processing of data-parallel kernels on heterogeneous systems. *IEEE Access: Practical Innovations, Open Solutions*, 9, 59968–59978. <https://doi.org/10.1109/access.2021.3073955>
- Wang, Z., Cui, C., Sui, J., & Guo, C. (2024). Online control parameter optimization design for multi-machine coordinated loading system of hazardous substances. *ISA Transactions*, 154, 213–227. <https://doi.org/10.1016/j.isatra.2024.09.002>
- Weiss, Y. (2021). Testing reactive systems using behavioural programming, a model centric approach. In *arXiv [cs.SE]*. <https://doi.org/10.48550/ARXIV.2112.01538>
- Zhu, J., Alderfer, K., Smith, B., Char, B., & Ontañón, S. (2020). Understanding learners' problem-solving strategies in concurrent and parallel programming: A game-based approach. En *arXiv [cs.HC]*. <https://doi.org/10.48550/arXiv.2005.04789>