e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

Alta disponibilidad en las PYMES del Ecuador a través del uso de Docker como tecnología Open Source

High availability in Ecuadorian smes through the use of docker as open-source technology

https://doi.org/10.5281/zenodo.20432224

AUTORES:

Omar Rodrigo Montece Moreno[1]
Universidad Técnica de Babahoyo
https://orcid.org/0000-0001-5421-1581
omontece@utb.edu.ec

Omar Oswaldo Veintimilla Benitez[2]
Universidad Técnica de Babahoyo
https://orcid.org/0009-0002-3546-1731
veintimillabenitez@fafi.utb.edu.ec

Vidal Meyith Dicado Sanchez[3]
Universidad Técnica de Babahoyo
https://orcid.org/0009-0008-5302-1995
vdicado619@fafi.utb.edu.ec

Arlintong Behael Pazmiño Bolaños[4]
Universidad Técnica de Babahoyo
https://orcid.org/0009-0000-3124-9122
abpazminob@fafi.utb.edu.ec

DIRECCIÓN PARA CORRESPONDENCIA: omontece@utb.edu.ec

Fecha de recepción: 20 / 11 / 2025

Fecha de aceptación: 27 / 11 / 2025

RESUMEN

Las pequeñas y medianas empresas (PYMES) en el Ecuador enfrentan importantes limitaciones para mantener la continuidad de sus servicios digitales, debido a restricciones económicas, escasa infraestructura tecnológica y dependencia de personal externo. La interrupción de servicios por fallos técnicos representa una amenaza directa a la

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

productividad y sostenibilidad de estos negocios. En este escenario, el concepto de alta disponibilidad (HA) adquiere especial relevancia como solución para minimizar el tiempo de inactividad, garantizar el acceso continuo a los sistemas y preservar la operatividad ante eventos adversos.

Este estudio propone una arquitectura distribuida basada en tecnologías open source, diseñada específicamente para el contexto de las PYMES locales. La solución se estructura con Docker Swarm como motor de orquestación de contenedores, HAProxy como balanceador de carga externo y Keepalived como herramienta de conmutación por fallo mediante el uso de una IP flotante. La propuesta fue implementada en un entorno virtualizado controlado, replicando las condiciones reales de una empresa con recursos limitados y sin acceso a soluciones comerciales de infraestructura redundante.

Se ejecutaron pruebas técnicas enfocadas en la medición del tiempo de recuperación ante fallos, consumo de recursos del sistema y disponibilidad efectiva de red, utilizando herramientas como ping, docker stats, htop y scripts de monitoreo. Los resultados evidencian que la arquitectura combinada permite mantener operatividad sin interrupciones significativas, con bajo consumo de CPU y memoria. Esta investigación demuestra que las herramientas open source ofrecen una vía accesible, escalable y técnicamente viable para implementar alta disponibilidad en las PYMES ecuatorianas.

Palabras clave: *Alta disponibilidad, Docker, PYMES, Open Source, HAproxy, Virtualización.*

ABSTRACT

Small and medium-sized enterprises (SMEs) in Ecuador face significant challenges in maintaining the continuity of their digital services due to limited financial resources, technological infrastructure, and dependence on external technical personnel. Service interruptions caused by system failures pose a direct threat to productivity and business sustainability. In this context, the concept of high availability (HA) becomes essential for minimizing downtime, ensuring continuous access to systems, and maintaining operational capacity during adverse events.

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

This study proposes a distributed architecture based on open-source technologies, specifically designed for the realities of Ecuadorian SMEs. The solution integrates Docker Swarm as the container orchestration engine, HAProxy as an external load balancer, and Keepalived to implement failover through a floating IP. The architecture was deployed in a controlled virtualized environment simulating limited-resource conditions, replicating real-world scenarios where access to commercial high-availability solutions is not feasible.

Technical tests focused on measuring recovery time after node failure, system resource consumption, and network availability using tools such as ping, docker stats, htop, and custom monitoring scripts. The results showed that the combined architecture can maintain uninterrupted service with minimal latency and low CPU and memory usage. This research demonstrates that open-source tools offer an accessible, scalable, and technically viable path to implement high availability in the SME sector, helping close the technological gap and fostering operational resilience.

Keywords: *High Availability, Docker, SEMEs, Open Sourse, HAproxy, Virtualization.*

INTRODUCCIÓN

Las pequeñas y medianas empresas (PYMES) constituyen el núcleo del tejido productivo en el Ecuador, representando un alto porcentaje del empleo formal, la generación de valor agregado y la dinamización de las economías locales. A pesar de su relevancia económica, muchas de estas empresas enfrentan limitaciones significativas en cuanto al acceso, adopción y sostenibilidad de tecnologías de información.

Una de las principales debilidades estructurales que enfrentan las PYMES ecuatorianas es la dependencia de servicios informáticos frágiles, poco escalables y, en muchos casos, centralizados en infraestructuras monolíticas. Esta realidad se traduce en una alta vulnerabilidad ante fallos técnicos, caídas de servicios o interrupciones inesperadas, las cuales pueden generar pérdidas económicas, comprometer la atención al cliente y afectar la reputación empresarial.

En entornos donde la digitalización es una condición necesaria para competir, la alta disponibilidad (HA) se convierte en una estrategia fundamental. Este concepto se refiere a la capacidad de un sistema informático de mantener su operatividad continua, incluso ante

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

fallos parciales, garantizando la accesibilidad y reduciendo al mínimo el tiempo de inactividad. Sin embargo, las soluciones tradicionales de HA suelen implicar costos elevados en hardware redundante, licencias propietarias y personal altamente especializado.

En respuesta a estas restricciones, el ecosistema del software libre ofrece un conjunto de herramientas cada vez más robustas, escalables y accesibles. Las tecnologías open source no solo reducen costos de implementación, sino que también permiten a las empresas adaptar sus soluciones a necesidades específicas, manteniendo el control sobre su infraestructura tecnológica.

Este estudio propone una arquitectura distribuida basada en Docker Swarm, una herramienta de orquestación de contenedores que permite desplegar servicios de forma replicada entre múltiples nodos. Para garantizar el acceso externo constante, se integra HAProxy, un balanceador de carga de alto rendimiento que distribuye el tráfico entrante. Finalmente, se emplea Keepalived para habilitar una IP flotante, permitiendo la conmutación automática entre nodos en caso de falla.

Docker es un proyecto de código abierto basado en contenedores de Linux. Se trata de un motor de contenedores que usan las características del Kernel de Linux, como espacios de nombres y controles de grupos, para crear contenedores encima del Sistema operativo y automatizar el despliegue de aplicaciones en estos contenedores.

La implementación de la arquitectura de Docker es una de las más exitosas de las arquitecturas de contenerización. Ofrece un conjunto completo de servicios como el almacenamiento, realizando las conexiones con las aplicaciones con complementos que ofrecen una gran escalabilidad. (García Ramírez, 2017)

Docker Swarm es una plataforma de orquestación de contenedores que permite la gestión y escalabilidad de aplicaciones distribuidas. Su adopción en el ámbito educativo ofrece ventajas como la automatización de tareas, la escalabilidad horizontal y la eficiencia en el uso de recursos. Sin embargo, la complejidad de la arquitectura Docker Swarm introduce nuevas superficies de ataque que deben ser consideradas y mitigadas. (Paúl, 2024)

Docker Swarm nos permite gestionar un grupo de hosts de Docker como un solo host de Docker virtual. Swarm utiliza la API estándar de Docker, por lo que cualquier herramienta

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

que se comunique con el Docker Daemon puede utilizar Docker Swarm, que permite la escalabilidad a varios hosts. (Ponsico Martin, 2017)

HAProxy es un motor de un solo subproceso, controlado por eventos y sin bloqueos, que combina una capa de E/S muy rápida con un programador basado en prioridades. Al estar diseñado con el objetivo de reenvío de datos, su arquitectura está optimizada para mover los datos lo más rápido posible con el mínimo de operaciones. Por ello, implementa un modelo en capas que ofrece mecanismos de derivación en cada nivel, lo que garantiza que los datos no alcancen niveles superiores cuando no se necesitan. La mayor parte del procesamiento se realiza en el núcleo, y HAProxy hace todo lo posible para que el núcleo realice el trabajo lo más rápido posible, dando algunas pistas o evitando ciertas operaciones cuando cree que podrían agruparse posteriormente. Como resultado, las cifras típicas muestran que el 15 % del tiempo de procesamiento se invierte en HAProxy frente al 85 % en el núcleo en modo de cierre TCP o HTTP, y aproximadamente el 30 % para HAProxy frente al 70 % para el núcleo en modo de mantenimiento de conexión HTTP. (HAProxy Version, 2022)

HAProxy es un software de código abierto que proporciona alta disponibilidad, balanceo de carga y un servidor proxy para aplicaciones TCP y HTTP y distribuye las peticiones de los clientes entre múltiples servidores.

A continuación, explicaremos los tres conceptos básicos para poder comprender el tipo de balanceos de carga que se pueden implementar con HAProxy:

Listas de control de acceso (ACL)

Las ACL son usadas como sentencias condicionales, es decir, se comprueba una condición y en función del resultado realizar una acción. El uso de estas listas nos permite dirigir el tráfico de red basado en distintos factores como podrían ser la identificación de patrones.

Un backend, en esta tecnología, es un conjunto de servidores que recibe peticiones reenviadas desde HAProxy. Se definen en el fichero de configuración y pueden ser definidas de dos formas:

Según el algoritmo de balanceo de carga que deben usar.

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

Según una lista de servidores y puertos. Un backend puede contener uno o más servidores. Añadir un mayor número de servidores a un backend concreto incrementa el potencial de balanceo de carga repartiendo la carga entre los múltiples servidores.

El frontend define como las peticiones tienen que ser redirigidas a los backends. También se definen en el fichero de configuración de HAProxy y están compuestas de los siguientes elementos:

Un conjunto de direcciones IP y un puerto.

Listas de control de acceso (ACL). 3. Reglas que definen que backend se debe usar Keepalived es un software escrito en C cuyo principal objetivo es proveer medios para mantener alta disponibilidad en sistemas e infraestructuras basadas en Linux de una manera sencilla y robusta, basado en el protocolo VRRP (Virtual Router Redundancy Protocol, Protocolo de Redundancia de Enrutador Virtual). (Tellez Milián & Aguila Jerez, 2016)

Es un sofware de enrutamiento cuyo objetivo es proporcionar de forma simple y robusta alta disponibilidad y balanceo de carga para sistemas basados en Linux. Implementa un sistema de comprobaciones que de forma dinámica y adaptable maneja un grupo de servidores en función de su salud (disponibilidad), con el objetivo de balancear su carga. Utiliza el protocolo Virtual Router Redundancy Protocol (VRRP) para crear routers redundantes, lo que garantiza que, si uno falla, otro puede asumir rápidamente el control, manteniendo la alta disponibilidad del servicio. Para la detección rápida de fallos en la red implementa el protocolo Bidirectional Forwarding Detection (BFD), el cual permite que los dispositivos de red detecten rápidamente cuando hay un fallo en la comunicación entre ellos. (Domínguez Cameán, 2024)

La *alta disponibilidad* es la capacidad que tiene un sistema de TI para ser accesible y confiable casi todo el tiempo, lo cual elimina o disminuye el tiempo de inactividad. Combina dos conceptos para determinar si un sistema cumple con su nivel de rendimiento operativo: el primero tiene que ver con la accesibilidad o la disponibilidad prácticamente permanentes de un servicio o servidor sin tiempo de inactividad; y el segundo se refiere a su funcionamiento según las expectativas razonables y durante un período establecido. No se trata solo de cumplir con el tiempo de actividad indicado en el acuerdo de nivel del servicio (SLA) o con

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

las expectativas establecidas entre el proveedor y el cliente, sino que implica que el sistema sea resistente, confiable y eficaz. (Redhat, 2025)

La integración de estas tres tecnologías open source busca construir un entorno de alta disponibilidad completo, ligero y funcional, adaptado a las condiciones reales de las PYMES ecuatorianas. La propuesta se diseñó bajo criterios de bajo consumo de recursos, facilidad de replicación y tolerancia a fallos, sin comprometer el rendimiento ni requerir grandes inversiones.

La arquitectura fue implementada en un entorno virtualizado controlado, simulando la infraestructura básica de una PYME con recursos limitados. Se ejecutaron pruebas orientadas a medir el tiempo de recuperación ante fallos, la disponibilidad efectiva del sistema, el consumo de CPU y memoria, así como la eficiencia del balanceo de carga entre nodos activos.

Esta investigación se enmarca en la necesidad urgente de ofrecer soluciones viables que permitan a las empresas pequeñas sostener sus operaciones digitales con fiabilidad. La propuesta técnica aquí presentada demuestra que es posible alcanzar niveles aceptables de alta disponibilidad mediante el uso estratégico de herramientas open source, contribuyendo a reducir la brecha tecnológica que separa a las PYMES de soluciones empresariales avanzadas.

METODOLOGÍA

La presente investigación adopta un enfoque cuantitativo, ya que se fundamenta en la recolección y análisis de datos numéricos obtenidos a través de pruebas controladas en un entorno virtualizado. Este enfoque permite medir objetivamente el comportamiento técnico de una arquitectura de alta disponibilidad construida exclusivamente con herramientas open source, orientada a satisfacer las necesidades tecnológicas de pequeñas y medianas empresas (PYMES).

Desde el punto de vista metodológico, se trata de una investigación aplicada, ya que busca resolver un problema técnico específico mediante la validación práctica de una solución replicable. El estudio es de tipo pre-experimental, dado que se trabaja con un solo grupo de

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

prueba sin grupo de control ni aleatorización, pero con condiciones técnicas homogéneas y controladas. Adicionalmente, se considera de diseño transversal, puesto que la recolección de datos se realiza en un único periodo definido, sin aplicar seguimiento longitudinal.

Este enfoque permite evaluar con objetividad el rendimiento de la arquitectura propuesta frente a variables críticas como el tiempo de recuperación ante fallos, la disponibilidad efectiva del sistema y el consumo de recursos operativos (CPU y memoria). La aplicación de métricas cuantificables y replicables asegura la validez técnica de los resultados y su posible extrapolación a contextos empresariales similares.

La presente investigación se clasifica como documental y técnica experimental. Es documental porque se fundamenta en el análisis de fuentes bibliográficas, artículos científicos, documentación oficial y manuales técnicos sobre las tecnologías evaluadas, como Docker Swarm, HAProxy y Keepalived. Esta revisión permitió establecer el marco teórico y operativo necesario para comprender el concepto de alta disponibilidad, así como los fundamentos funcionales de cada herramienta seleccionada (Tamayo y Tamayo, 2011).

Asimismo, es una investigación de tipo experimental técnico-aplicado, ya que se realizaron pruebas prácticas en un entorno virtualizado controlado, diseñado específicamente para simular condiciones reales de operación en servidores utilizados por pequeñas y medianas empresas. A través de este enfoque, se recolectaron datos empíricos derivados de la implementación real de la arquitectura, lo cual permitió analizar de forma directa su comportamiento ante eventos de fallo, recuperación y balanceo de carga.

Esta combinación metodológica permite no solo comprender los conceptos subyacentes desde una perspectiva teórica, sino también contrastarlos con evidencia generada mediante simulación estructurada en condiciones homogéneas, facilitando la evaluación objetiva de una solución de alta disponibilidad orientada a PYMES con infraestructura limitada.

El alcance de la presente investigación es de tipo descriptivo-explicativo con enfoque técnico-aplicado, ya que busca evaluar detalladamente el comportamiento de una arquitectura de alta disponibilidad implementada con tecnologías open source. Esta arquitectura combina Docker Swarm como sistema de orquestación de contenedores,

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

HAProxy como balanceador de carga externo, y Keepalived como mecanismo de conmutación mediante IP flotante.

No se trata de una comparación entre soluciones distintas, sino de la validación empírica de una única propuesta integral, diseñada para funcionar de manera conjunta. El objetivo es identificar su capacidad para mantener la continuidad operativa del servicio, minimizar el tiempo de inactividad y distribuir eficientemente el tráfico de red, todo ello bajo condiciones controladas y representativas de una pequeña empresa con recursos tecnológicos limitados.

Este tipo de alcance es pertinente cuando se busca generar evidencia técnica confiable sobre la efectividad de una solución específica, observando variables como el tiempo de recuperación ante fallos, la disponibilidad efectiva, y el consumo de recursos operativos. La arquitectura no fue modificada internamente ni se alteraron sus algoritmos, permitiendo una observación objetiva basada en pruebas de carga, simulación de fallos y monitoreo del rendimiento en tiempo real.

El estudio se enfoca en analizar detalladamente cómo se comporta esta solución frente a condiciones adversas típicas de entornos empresariales con infraestructura limitada. A través de pruebas sistemáticas y métricas cuantificables, se busca determinar si esta arquitectura puede ser replicada con éxito por pequeñas y medianas empresas ecuatorianas que requieren alta disponibilidad sin incurrir en altos costos ni depender de tecnologías propietarias.

El diseño adoptado en esta investigación es de tipo pre-experimental, ya que se realizó una intervención técnica controlada sobre un único grupo de estudio, sin utilizar aleatorización ni grupo de control. Esta elección metodológica es adecuada cuando el propósito es evaluar el desempeño de una solución tecnológica en un entorno simulado, sin manipular sus componentes internos ni requerir comparación entre grupos.

La arquitectura de alta disponibilidad compuesta por Docker Swarm, HAProxy y Keepalived fue desplegada en un entorno virtualizado homogéneo, asegurando condiciones constantes de hardware, red y configuración. Las pruebas fueron diseñadas para simular escenarios de fallo, recuperación y balanceo de carga, con el objetivo de observar directamente el comportamiento del sistema y medir su rendimiento ante eventos críticos.

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

Asimismo, la investigación se enmarca en un enfoque transversal, ya que la recolección de datos se realizó en un único momento o etapa experimental bien definida. Las observaciones, métricas y evidencias fueron registradas durante sesiones específicas de prueba, sin aplicar seguimiento en el tiempo, lo cual permitió capturar una fotografía técnica precisa del comportamiento de la arquitectura frente a fallos operativos y demandas de carga. La investigación se desarrolló en un entorno virtualizado controlado, utilizando VirtualBox como hipervisor principal. Se configuraron dos máquinas virtuales independientes basadas en Ubuntu Server 24.04 LTS, cada una con una asignación de 2 núcleos de CPU virtual, 4 GB de memoria RAM y 20 GB de almacenamiento, simulando condiciones realistas propias de una pequeña o mediana empresa con recursos limitados.

En este entorno se implementó una arquitectura única y combinada de alta disponibilidad, utilizando exclusivamente tecnologías open source. La solución técnica estuvo compuesta por tres elementos principales:

Docker Swarm: como sistema de orquestación de contenedores, encargado de gestionar la replicación de servicios, distribuir automáticamente las cargas internas entre los nodos y mantener los servicios activos incluso ante la caída de uno de ellos.

HAProxy: como balanceador de carga externo, configurado para recibir las peticiones de los usuarios finales y redirigirlas dinámicamente a los contenedores activos desplegados en el clúster de Docker Swarm, utilizando un algoritmo de tipo round-robin.

Keepalived: como mecanismo de failover, encargado de gestionar una IP flotante entre los dos nodos, permitiendo que el acceso al sistema se mantenga constante y transparente para el usuario final, incluso ante la caída del nodo principal.

La arquitectura fue diseñada para garantizar alta disponibilidad tanto a nivel interno (servicios Docker) como a nivel externo (acceso de red). Para ello, se simularon caídas de nodos, fallos en la red y reinicios del sistema, con el fin de evaluar el comportamiento real del conjunto bajo condiciones adversas.Durante las pruebas, se utilizaron herramientas como ping, htop, docker stats, journalctl, y cronómetros digitales para medir variables como tiempo de recuperación, consumo de recursos y disponibilidad efectiva. Además, se recopilaron capturas de pantalla, logs del sistema y salidas de consola que permitieron registrar

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

objetivamente el comportamiento del entorno.Este entorno fue seleccionado específicamente por su flexibilidad, control técnico y bajo costo, lo cual permite replicar las condiciones en empresas que no disponen de infraestructura física dedicada ni personal técnico altamente especializado.

En esta investigación se establecieron tres variables técnicas clave para evaluar el desempeño de la arquitectura de alta disponibilidad construida con Docker Swarm, HAProxy y Keepalived. Estas variables fueron definidas con base en métricas comúnmente empleadas en pruebas de infraestructura técnica y entornos simulados controlados.

A continuación, se describen las variables observadas:

Tiempo de recuperación ante fallos: Se refiere al tiempo exacto, en segundos, que tarda el sistema en restaurar el servicio tras una interrupción o caída intencionada. Esta variable es fundamental para determinar la capacidad de respuesta de la arquitectura ante eventos imprevistos.

Porcentaje de disponibilidad efectiva: Representa el tiempo total durante el cual el servicio estuvo accesible y operativo, expresado como un porcentaje sobre el tiempo total de prueba. Este indicador refleja la estabilidad y confiabilidad general del sistema desde la perspectiva del usuario final.

Consumo promedio de recursos: Incluye el uso de memoria RAM (en megabytes) y uso de CPU (en porcentaje promedio) durante el funcionamiento normal y en escenarios de falla. Esta variable permite medir la eficiencia operativa de la solución en entornos con recursos limitados.

La selección de estas variables responde a la necesidad de contar con medidas objetivas, cuantificables y replicables, que permitan validar técnicamente si la arquitectura propuesta es viable para su implementación en PYMES reales. Además, estas métricas coinciden con los criterios técnicos relevantes que una pequeña o mediana empresa podría considerar al momento de adoptar una solución de alta disponibilidad sin invertir en hardware especializado.

Las mediciones fueron registradas con herramientas del sistema como ping, docker stats, htop y cronómetros digitales. La información fue organizada para permitir un análisis

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

descriptivo, con evidencia visual y métricas verificables. A continuación, se presenta la tabla de operacionalización de las variables observadas.

Tabla 1

Operacionalización de las variables técnicas evaluadas en la arquitectura Docker Swarm + HAProxy + Keepalived

Variable	Definición operacional	Indicador	Unidad de medida	Instrumento de recolección
Tiempo de recuperación	Intervalo desde la caída de un nodo hasta la restauración automática del servicio	Tiempo de failover	Segundos (s)	Cronómetro, ping, journalctl
Disponibilidad efectiva	Porcentaje del tiempo total en que el servicio estuvo activo durante el periodo de prueba	Porcentaje de uptime	Porcentaje (%)	Script de monitoreo, registros del sistema
Consumo de memoria RAM	Promedio de uso de memoria durante operación normal y durante eventos de falla	Memoria promedio utilizada	Megabytes (MB)	htop, free -m, docker stats
Uso de CPU	Porcentaje promedio de carga de CPU durante ejecución estable y fallos simulados	Carga promedio de CPU	Porcentaje (%)	htop, top, docker stats

Fuente: *Elaborado por los Autores.*

Nota. Las unidades de medida corresponden a los valores reportados directamente por las herramientas del sistema bajo condiciones técnicas estandarizadas.
Para garantizar la objetividad y replicabilidad de los resultados, se emplearon instrumentos técnicos de recolección de datos específicamente orientados al monitoreo de variables de rendimiento en sistemas operativos basados en Linux. La elección de cada herramienta respondió a su disponibilidad, precisión y adecuación al entorno de pruebas controlado.
Entre los instrumentos utilizados se destacan los siguientes: *htop:* Monitor en tiempo real del sistema que permite observar el uso de CPU y memoria RAM de manera detallada. Se empleó

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

para registrar el comportamiento de los recursos durante la operación normal y tras la ocurrencia de fallos simulados. *docker stats*: Herramienta propia del ecosistema Docker que permite monitorear el consumo de recursos por contenedor. Fue utilizada en el entorno con Docker Swarm para obtener métricas precisas a nivel de servicio. *ping:* Comando de diagnóstico utilizado para verificar la disponibilidad del servicio durante las pruebas. Permite identificar el momento exacto de pérdida y recuperación de conectividad. *journalctl:* Visualizador de logs del sistema en distribuciones basadas en systemd. Se utilizó para registrar eventos relevantes del sistema, como caída y reinicio de servicios, así como para validar el comportamiento de HAProxy y Keepalived. *Cronómetro digital*: Herramienta manual utilizada para validar el tiempo de failover, contrastando la observación directa con los datos registrados por el sistema. *Scripts de monitoreo personalizados:* Se desarrollaron pequeños scripts en Bash para automatizar la verificación de disponibilidad en intervalos regulares y registrar los resultados en archivos de texto plano para su posterior análisis. Todos los datos recolectados fueron organizados en hojas de cálculo y graficados para su análisis descriptivo, facilitando la interpretación objetiva de los resultados obtenidos. No se emplearon instrumentos cualitativos ni formularios tipo encuesta, dado que el estudio se centró exclusivamente en la observación técnica de una única arquitectura integrada desplegada en un entorno simulado.La presente investigación se desarrolló bajo un enfoque transversal, ya que la recolección de datos se realizó en un único momento definido dentro del entorno experimental (Hernández Sampieri, 2014). Las pruebas técnicas fueron ejecutadas en una ventana temporal delimitada, durante la cual se sometió el sistema a condiciones puntuales de fallo, recuperación y carga controlada, con el objetivo de capturar métricas precisas del comportamiento técnico de la solución. Este tipo de enfoque temporal resulta apropiado para estudios experimentales donde se busca obtener una fotografía clara y puntual del rendimiento de una solución tecnológica ante situaciones específicas, sin requerir seguimiento longitudinal. Las condiciones de prueba, los parámetros de monitoreo y los instrumentos utilizados se mantuvieron constantes durante todo el periodo de evaluación, asegurando la consistencia de los datos recolectados.

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

RESULTADOS

Los resultados obtenidos permiten validar la efectividad técnica de la arquitectura propuesta, conformada por Docker Swarm como motor de orquestación de contenedores y HAProxy con Keepalived para el manejo del acceso externo. El sistema demostró capacidad de recuperación automática ante fallos, con un tiempo estimado de recuperación de aproximadamente 15 segundos y pérdida mínima de conectividad. Esta tolerancia a fallos es crítica para entornos PYMES, donde la continuidad del servicio es vital pese a recursos limitados.

Tabla 2
Desempeño tecnico de la arquitectura DOcker Swarm más Haproxy y Keepalived en entorno simulado

Variable técnica	Resultado	Unidad de medida	Método de prueba
Tiempo de recuperación	~15	Segundos	Apagado de nodo (poweroff)
Paquetes perdidos (ping)	1	Paquetes	ping constante
Disponibilidad efectiva	>99.8 %	Porcentaje	Cálculo estimado con ping
Uso promedio de CPU	0.04–2.2	Porcentaje (%) por contenedor	docker stats / htop
Uso promedio de RAM	~13.1	Megabytes por contenedor nginx	docker stats
Balanceo de carga	Alternancia validada	Round-robin funcional	Navegador y PowerShell

Fuente: *Elaborado por los Autores.*

Además, las métricas de consumo evidencian una solución **ligera y eficiente**, con un uso promedio de CPU inferior al 3 % y requerimientos mínimos de memoria por contenedor, lo que la hace adecuada para escenarios de infraestructura modesta. El correcto funcionamiento del balanceo de carga round-robin garantiza la distribución equitativa del tráfico, maximizando el aprovechamiento de los recursos disponibles.

En conjunto, estos resultados confirman que la combinación de tecnologías open source empleadas permite construir un entorno de alta disponibilidad funcional, replicable y viable en condiciones propias de pequeñas y medianas empresas ecuatorianas.

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

Tabla 3

Resumen de resultados técnicos obtenidos durante la validación de la arquitectura Docker Swarm + HAProxy + Keepalived

Escenario de prueba	Tiempo de recuperación (s)	Disponibilidad efectiva (%)	Uso promedio de CPU (%)	Uso promedio de RAM (MB)
Operación normal (sin fallos)	–	100 %	1.2 %	12.8 MB
Caída del nodo A (failover)	15 s	99.7 %	1.6 %	13.1 MB
Simulación de tráfico continuo	–	100 %	2.2 %	14.2 MB

Fuente: *Elaborado por los Autores.*

Nota. Los valores fueron registrados con docker stats, htop, ping y cronómetros digitales en tiempo real durante eventos controlados.

El consumo de recursos en cada nodo durante la Prueba 3, combinando métricas a nivel de contenedor (obtenidas con docker stats) y a nivel de sistema (medidas con htop). En ella se observa que cada instancia de webstack_web utiliza aproximadamente 13 MiB de RAM y menos de 0,1 % de CPU bajo carga mínima, mientras que el servidor completo mantiene un uso global de memoria inferior al 7 % de su capacidad y un consumo de CPU cercano al 2 %. Estos resultados evidencian que la arquitectura es extremadamente ligera y eficiente, validando su viabilidad para entornos de pequeñas y medianas empresas donde la optimización de recursos es prioritaria.

Tabla 4

Consumo de recursos en nodos durante Prueba de Alta Disponibilidad

Nodo	Contenedor	RAM usada (Docker stats)	CPU (%) (Docker stats)	RAM total usada (htop)	CPU global usada (htop)
nodo-a	webstack_web.1	13.15 MiB	0.04 %	266 MiB / 3.82 GiB	~ 2.0 %
nodo-b	webstack_web.2	13.16 MiB	0.06 %	266 MiB / 3.82 GiB	~ 2.1 %

Fuente: *Elaborado por los Autores.*

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

DISCUSIÓN

Los resultados obtenidos en la implementación de la arquitectura basada en Docker Swarm, HAProxy y Keepalived evidencian que es posible construir un entorno de alta disponibilidad funcional y eficiente, utilizando exclusivamente tecnologías open source. Esta solución, al ser ligera en el uso de recursos y técnicamente replicable, resulta especialmente adecuada para pequeñas y medianas empresas ecuatorianas, que enfrentan limitaciones económicas y de infraestructura tecnológica.

La capacidad del sistema para recuperarse automáticamente tras la caída de un nodo, con una interrupción mínima del servicio, confirma la efectividad del diseño propuesto en términos de tolerancia a fallos. Asimismo, el uso de una IP flotante gestionada por Keepalived permitió mantener la continuidad de red prácticamente sin pérdida de paquetes, lo cual es un indicador crucial para entornos que requieren estabilidad en sus servicios digitales.
La eficiencia operativa también quedó demostrada con el bajo consumo de CPU y memoria, lo que valida el potencial de esta solución en entornos donde no es factible adquirir hardware especializado ni contratar servicios comerciales de alta disponibilidad. Esto se alinea con diversos estudios que resaltan el valor del software libre como herramienta estratégica para reducir la brecha tecnológica en sectores empresariales pequeños o emergentes.
El balanceo de carga mediante HAProxy no solo mejora la distribución del tráfico, sino que asegura que los recursos disponibles se utilicen de forma óptima, evitando sobrecargas en nodos individuales. Este aspecto es fundamental para mantener el rendimiento sostenido del sistema y prevenir caídas por saturación, algo común en infraestructuras no redundantes.
Si bien la arquitectura cumplió con los objetivos técnicos, es importante destacar que su implementación requiere ciertos conocimientos técnicos para su configuración inicial. Por tanto, se recomienda que las PYMES interesadas en adoptarla cuenten con soporte profesional mínimo, o bien se capaciten mediante recursos comunitarios y documentación oficial disponible.
En conjunto, esta investigación demuestra que es posible aplicar soluciones tecnológicas de alta disponibilidad en contextos reales con recursos limitados, sin comprometer la

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

funcionalidad ni la estabilidad del sistema, gracias al uso adecuado y combinado de herramientas open source como Docker y HAProxy.

CONCLUSIONES

La presente investigación permitió validar que es posible implementar una arquitectura de alta disponibilidad eficiente, funcional y accesible para las PYMES del Ecuador mediante el uso combinado de tecnologías open source como Docker Swarm, HAProxy y Keepalived. La solución propuesta demostró capacidad de recuperación automática ante fallos, continuidad operativa de red y balanceo de carga dinámico, todo ello bajo condiciones controladas que simulan la realidad de muchas empresas con infraestructura limitada.

El uso de contenedores orquestados con Docker Swarm, complementado por HAProxy como punto de acceso externo, permitió superar una de las limitaciones más comunes de las PYMES: la dependencia de servidores únicos sin tolerancia a fallos. La arquitectura validada logró mantener la disponibilidad del servicio incluso ante la caída de un nodo, con tiempos de recuperación aceptables y mínima interrupción del tráfico.

Asimismo, el bajo consumo de recursos registrado durante las pruebas confirma que esta solución puede ser implementada incluso en equipos de gama media, sin necesidad de adquirir hardware especializado o licencias propietarias, lo que representa un ahorro considerable y una oportunidad estratégica para la transformación digital del sector.

Como proyección, se recomienda a las PYMES interesadas considerar la adopción progresiva de tecnologías open source de infraestructura, con acompañamiento técnico inicial y capacitación interna. La arquitectura validada puede ser escalada fácilmente y adaptarse a diversos servicios, consolidándose como una alternativa sostenible y replicable para entornos productivos de misión crítica.

REFERENCIAS BIBLIOGRÁFICAS

Anicas, M. (2022, 28 marzo). *An Introduction to HAProxy and Load Balancing Concepts*. DigitalOcean. https://www.digitalocean.com/community/tutorials/an-introduction-to-haproxy-and-load-balancing-concepts

CAF. (2020). *Las PYMES en Ecuador: Estructura, desafíos y oportunidades*. Obtenido de

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

Banco de Desarrollo de América Latina: https://scioteca.caf.com/bitstream/handle/123456789/2132/CAF_PYMES_ECUADOR.pdf

Docker Inc. (2025). *Swarm mode overview*. Obtenido de Docker Docs: https://docs.docker.com/engine/swarm/

Domínguez Cameán, I. D. C. (2024, junio). *Despliegue y confguración automatizada de un clúster kubernetes con alta disponibilidad*. https://ruc.udc.es/rest/api/core/bitstreams/128fc21e-45b4-4638-84ee-2e006e42d08e/content

García Ramírez, M. G. R. (2017). *Automatización del despliegue de una aplicación web en un clúster Swarm utilizando los servicios en la nube de Amazon Web Services y tecnología Docker*. accedaCRIS. https://accedacris.ulpgc.es/handle/10553/23930

HAProxy Technologies. (26 de 02 de 2025). *HAProxy Documentation*. Obtenido de HAProxy.org: https://www.haproxy.org

HAProxy Version. (2022). *HAProxy Version 1.7-Dev3 - Starter Guide*. https://cbonte.github.io/haproxy-dconv/intro-1.7.html#3

Hernández Sampieri, R. F. (2014). *Metodología de la investigación*. México: McGraw-Hill Education.

Keepalived Project. (13 de 06 de 2020). *Keepalived*. Obtenido de Keepalived.org: https://keepalived.org

Paúl, A. M. A. (2024). *Análisis de seguridad para despliegues de plataformas educativas tecnológicas moodle en docker swarm desarrolladas por la empresa TRASCEND-IT*. https://repositorio.puce.edu.ec/items/7e8cbc04-63e4-4fb1-8535-78d3a0b628c8

Ponsico Martin, P. P. M. (2017, 18 octubre). *Tecnología de contenedores docker*. https://upcommons.upc.edu/handle/2117/113040

Redhat. (2025). *Alta disponibilidad (HA): ¿Qué es y cómo funciona?* https://www.redhat.com/es/topics/linux/what-is-high-availability

Revista Espacios. (2017). *Análisis estructural de las MIPYMES en América Latina*. Revista Espacios, 38(53), 15. Obtenido de https://www.revistaespacios.com/a17v38n53/a17v38n53p15.pdf

Sheldon, R. &. (29 de 07 de 2024). *TechTarget*. Obtenido de TechTarget:

e-ISSN: 2528-8091
Vol. 10, N° V CICACI - 2025

https://www.techtarget.com/searchdatacenter/definition/high-availability

Tamayo y Tamayo, M. (2011). *El proceso de la investigación científica*. México: Limusa.

Tellez Milián, H. T. M., & Aguila Jerez, A. D. A. J. (2016). *Sistema para la configuración y monitoreo de alta disponibilidad, balanceo de carga en aplicaciones con HAProxy y Keepalived*. https://repositorio.uci.cu/handle/123456789/7627