

SISTEMAS



Paradigma Serverless: Función como servicio





ACISTIC 2023

SOSTENIBILIDAD + INNOVACIÓN + TECNOLOGÍA

Un mundo más sostenible desde la
Innovación y la Ingeniería

24, 25 y 26 de octubre

12: 00 - 1: 30 pm
&
5:30 - 7:00 pm

Más información
3015530540 - 3043463413
www.acis.org.co/ACISTIC2023

Con la participación de :



NOS RENOVAMOS

LA ASOCIACIÓN COLOMBIANA DE INFORMÁTICA, SISTEMAS Y TECNOLOGÍAS AFINES



ACIS

ASOCIACIÓN COLOMBIANA DE INFORMÁTICA, SISTEMAS Y TECNOLOGÍAS AFINES

Más información en:
www.ACIS.org.co
o escríbenos a:
301 5530540
Suscripciones@acis.org.co
Cursos@acis.org.co

Queremos expandirnos. Es por esto,
que hemos decidido ampliar
nuestro nombre, para poder tener
mayor alcance a todas las personas
e instituciones que van de la mano
con la tecnología.

En esta edición

Editorial

Paradigma serverless: función como servicio
DOI: 10.29236/sistemas.n168a1

4

Columnista Invitado

Serverless: un camino para impulsar la innovación
DOI: 10.29236/sistemas.n168a2

8

Entrevista

Jeison Vergara Vargas
DOI: 10.29236/sistemas.n168a3

Profesional con amplia experiencia para abordar diferentes asuntos relacionados con el "Paradigma Serverless".

14

Investigación

Paradigma serverless: función como servicio
DOI: 10.29236/sistemas.n168a4

El presente estudio es el resultado de una encuesta a través de Internet, dirigida a profesionales del sector TI a nivel nacional. Se basó en un formulario virtual de diez preguntas y la respuesta fue de 71 encuestados a nivel nacional.

18

Cara y Sello

Paradigma Serverless: ¿evolución de la computación en la nube?
DOI: 10.29236/sistemas.n168a5

Para analizar los pormenores de este nuevo paradigma, la revista Sistemas reunió a algunos expertos.

28

Uno

Serverless Computing ("Computación sin servidor")
DOI: 10.29236/sistemas.n168a6

Reflexiones básicas sobre seguridad y control.

35

Dos

Paradigma Serverless: función como servicio
DOI: 10.29236/sistemas.n168a7

44

Publicación de la Asociación Colombiana de
Ingenieros de Sistemas (ACIS)
Resolución No. 003983 del
Ministerio de Gobierno
Tarifa Postal Reducida Servicios Postales
Nacional S.A. No. 2015-186 4-72
ISSN 0120-5919
Apartado Aéreo No. 94334
Bogotá D.C., Colombia

Dirección General

Jeimy J. Cano M.

Consejo de Redacción

Francisco Rueda F.
Gabriela Sánchez A.
Manuel Dávila S.
Andrés Ricardo Almanza J.
Emir Hernando Pernet C.
Fabio Augusto González O.
Jorge Eliécer Camargo M.
María Mercedes Corral S.

Editores Técnicos

Jorge Eliécer Camargo M.
Fabio Augusto González O.

Editora

Sara Gallardo M.

Junta Directiva ACIS

2022-2024

Presidente

Luis Javier Parra B.

Vicepresidente

Jorge Fernando Bejarano L.

Secretario

Rodrigo Rebolledo M.

Tesorero

Jaime García C.

Vocales

Hilda Cristina Chaparro L.
Soledad Mercedes Gutiérrez R.

Directora Ejecutiva

Beatriz E. Caicedo R.

Diseño y diagramación

Bruce Garavito

Los artículos que aparecen en esta edición no
reflejan necesariamente el pensamiento de la
Asociación. Se publican bajo la responsabilidad
de los autores.

Julio - Septiembre 2023

Calle 93 No.13 - 32 Of. 102
Teléfonos 616 1407 - 616 1409
A.A. 94334
Bogotá D.C.
www.acis.org.co

NASCO

NACIONAL DE COMPUTADORES S.A.

APOYA ESTA PUBLICACIÓN

TEL: 6 06 06 06- CR 15 No 72-73



Confía en 4-72,
el servicio de envíos
de Colombia

Línea de atención al cliente:
(57 - 1) 472 2000 en Bogotá
01 8000 111 210 a nivel Nacional

.....
www.4-72.com.co

MARATÓN REGIONAL 2023

¡Anunciamos la realización de la Maratón Regional de Programación 2023! para la comunidad de ACIS. La Maratón Regional se llevará a cabo en el campus principal del Politécnico Grancolombiano, donde 40 equipos clasificados de la Maratón Nacional competirán. El evento tendrá lugar el próximo 21 de octubre, donde los estudiantes demostrarán sus habilidades y creatividad para resolver problemas complejos en un corto período de tiempo.

Apoya:



icpc.foundation



Paradigma serverless: función como servicio

DOI: 10.29236/sistemas.n168a1

Jorge Eliécer Camargo Mendoza, PhD

Fabio Augusto González Osorio, PhD

El paradigma de la computación serverless se impone hoy en día como una prometedora alternativa al tradicional enfoque de construcción de software en el que los desarrolladores de software se tienen que preocupar por la infraestructura de hardware en la que correrán sus aplicaciones [1]. En este nuevo enfoque, se elimina la necesidad de administrar infraestructura tradicional y permite a los desarrolladores de software centrarse en la creación de aplicaciones sin preocuparse por la gestión de servido-

res. Como otras tecnologías que emergen día a día, la adopción de la computación sin servidor (*serverless*) resulta “natural” para quienes la empiezan a utilizar, lo cual genera ventajas y desventajas que deben ser analizadas cuidadosamente para determinar si es lo más adecuado para las nuevas soluciones de software que se construyan o como una alternativa a las existentes [2].

En este número, exploraremos qué es exactamente el paradigma de

computación serverless, su impacto en la eficiencia y agilidad del desarrollo de software, así como sus implicaciones en otros aspectos clave a la hora de seleccionar los componentes de la arquitectura para una aplicación. Descubriremos cómo esta innovación está dando forma al futuro de la computación y abriendo nuevas posibilidades para las empresas y para los desarrolladores de software en todo el mundo.

En la entrevista realizada a Jeisson Vergara Vargas, profesor auxiliar del Departamento de Ingeniería de Sistemas e Industrial de la Universidad Nacional de Colombia sede Bogotá, se aborda el paradigma serverless con respecto a la seguridad, arquitectura de software, proveedores de servicios, retos, y ventajas/desventajas.

Como columnista invitado tenemos a Nicolás Rozo Espinoza, Ingeniero de Sistemas de la Universidad ECCI, experto en Arquitectura Empresarial y en Arquitectura de Soluciones en la Nube, nos plantea los retos y desafíos de la computación serverless desde una perspectiva organizacional, con un análisis de las dependencias que se generan con los proveedores de servicios de computación serverless, así como los costos asociados al uso de este tipo de tecnologías.

En la investigación, se analizaron los resultados de una encuesta que realizamos para medir el impacto

de las Apps en las personas, en la cual participaron 394 profesionales de TI, a nivel nacional. Los resultados de la encuesta arrojaron resultados muy interesantes: la mayoría de los profesionales encuestados están utilizando servicios serverless, con un enfoque principal en AWS Lambda, Google Cloud Functions y Azure Functions. Las empresas encuentran beneficios significativos al utilizar servicios serverless, con ahorro de costos, escalabilidad y flexibilidad como los principales impulsores. Aunque hay beneficios notables, las empresas enfrentan desafíos, siendo la falta de habilidades y experiencia, así como la necesidad de un cambio cultural, los obstáculos más prominentes. La complejidad de configuración y gestión de la seguridad también son áreas de preocupación. Las empresas están tomando medidas para adaptar sus procesos de desarrollo y operaciones para incorporar servicios serverless, a través de la introducción de nuevos procesos, formación de equipos y adopción de metodologías ágiles. Las empresas están utilizando una variedad de métodos para medir el rendimiento y el costo de sus aplicaciones serverless, incluyendo definir métricas clave, utilizar herramientas de monitorización y análisis de datos, y adoptar un enfoque proactivo basado en datos históricos. Por último, la gran mayoría de los encuestados recomiendan servicios serverless, señalando su capacidad para ahorrar costos, escalar rápidamente y me-

jorar la agilidad. Sin embargo, un pequeño porcentaje advierte que no son adecuados para todas las empresas y sugiere una evaluación detallada antes de su adopción.

En la sección Cara y Sello tuvimos como invitados a: Nicolás Rozo, gerente de Arquitectura & i+I+D en Nequi; Jonatan Ponzó director Arquitectura, DevEx y cosas en Rappi; y a Fabio A. González, miembro Comité Editorial de la Revista Sistemas. Se discutieron aspectos de la computación serverless tales como los riesgos de seguridad, aspectos a tener en cuenta en la arquitectura de las aplicaciones bajo este paradigma, qué tan adecuado es utilizar este paradigma, dependencia de los proveedores de nube, y el futuro de esta tecnología.

En el artículo UNO titulado “*Serverless Computing (Computación sin servidor). Reflexiones básicas sobre seguridad y control*” realizado por Jeimy J. Cano, se presenta el concepto de “computación sin servidor” y la evolución de la computación en la nube. También el autor plantea las amenazas y controles de seguridad, así como los diferentes tipos de adversarios y tipos


de vulnerabilidades que surgen en la computación sin servidor.

En el artículo DOS titulado “*Paradigma Serverless: función como servicio*” elaborado por José Bocanegra, se presenta cómo nace el modelo de “función como servicio” (FaaS, por sus siglas en Inglés), explicando sus ventajas/desventajas y se esbozan los desafíos a futuro.

En este contexto, esta edición muestra en sus diferentes secciones temas relacionados con la adopción de los servicios de computación serverless y otros temas relacionados, los cuales esperamos sean de interés para el lector.

Referencias

[1] Samuel Kounev, Nikolas Herbst, Cristina L. Abad, Alexandru Iosup, Ian Foster, Prashant Shenoy, Omer Rana, and Andrew A. Chien. 2023. Serverless Computing: What It Is, and What It Is Not? *Commun. ACM* 66, 9 (September 2023), 80–92. <https://doi.org/10.1145/3587249>

[2] Y. Li, Y. Lin, Y. Wang, K. Ye and C. Xu, "Serverless Computing: State-of-the-Art, Challenges and Opportunities," in *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 1522-1539, 1 March-April 2023, doi: 10.1109/TSC.2022.3166553. 

Jorge Eliécer Camargo Mendoza, PhD. Decano de la Facultad de Ingeniería de Sistemas de la Universidad Antonio Nariño. Ingeniero de Sistemas de la Universidad Nacional de Colombia, Magíster en Ingeniería de Sistemas y Computación de la Universidad de los Andes. Doctor en Ingeniería de Sistemas y Computación de la Universidad Nacional de Colombia (Tesis Laureada). Cuenta con más de 100 publicaciones científicas en revistas, libros y conferencias internacionales relacionadas con Big Data, Machine Learning, Inteligencia Artificial y Ciberseguridad.

Fabio González Osorio, PhD. Fabio Augusto González O. Profesor titular del Departamento de Ingeniería de Sistemas e Industrial de la Universidad Nacional de Colombia, sede Bogotá. Es ingeniero de Sistemas y Magíster en Matemáticas de la Universidad Nacional de Colombia, MSc y PhD en Ciencias de la Computación de la Universidad de Memphis, Estados Unidos. Lidera el grupo de investigación MindLab. Su trabajo de investigación se enfoca en los fundamentos de aprendizaje de máquina y recuperación de información y su aplicación en el análisis de imágenes y texto, visión por computador y minería de datos.

Serverless: un camino para impulsar la innovación

DOI: 10.29236/sistemas.n168a2



Nicolás Rozo Espinosa

Este texto escrito para la sección de la revista denominada *columnista invitado* explica cómo la arquitectura *serverless* libera a las empresas de la gestión y operación de la infraestructura, impulsando la innovación y la agilidad.

Con una gestión adecuada, las empresas pueden maximizar *serverless* y liderar en la era digital.

Serverless: retos y desafíos en el camino

En la actualidad, las empresas se encuentran en una carrera por **desarrollar iniciativas innovadoras, diferenciales y de gran impacto de una manera flexible, ágil y eficiente**; estas compañías se apalancan en la evolución acelerada de la tecnología y, en especial de la computación en la nube,

donde se han generado mecanismos que permiten una mayor agilidad y versatilidad en la materialización de sus proyectos.

En este contexto, *serverless* llega como una respuesta a esta necesidad: permite a las organizaciones **construir y ejecutar aplicaciones sin tener que preocuparse por la infraestructura**, lo que les permite enfocar a los equipos de desarrollo en crear soluciones innovadoras para sus clientes a nivel de software, sin tener que pensar sobre qué infraestructura se va a soportar.

Bondades de *serverless* para las organizaciones

Es increíble cómo la tecnología evoluciona, se vuelve más potente, adaptable y asequible. Muestra de ello es *serverless* que ha ido mejorando su flexibilidad en la ejecución de aplicaciones y su facilidad para integrarse con otros servicios en la nube, lo que facilita la **construcción de aplicaciones de alta disponibilidad y escalables**.

Además, ofrece mayor velocidad de desarrollo y mejor rendimiento, en comparación con otros modelos de computación en la nube. Sin embargo, también tiene limitaciones que deben ser consideradas al implementar esta tecnología: menor control sobre la infraestructura, dificultada para hacer pruebas locales, limitaciones en el almacenamiento temporal y su tiempo de ejecución máximo, si esto llega a su-

ceder podría ser necesario dividir las cargas, implicando una mayor complejidad.

De otro lado, en lugar de administrar servidores y recursos, los desarrolladores solo necesitan escribir código en forma de funciones que se ejecutan en la nube. Estas funciones son invocadas por eventos, como solicitudes HTTP o mensajes de cola, y su tiempo de ejecución es facturado por uso. Este modelo de computación ha revolucionado la forma en que se construyen y ejecutan aplicaciones, gracias a la habilidad de escalar automáticamente las capacidades en función de la demanda, **lo que significa que solo pagan por los recursos que utilizan**.

Otro caso de implementación de *serverless* es en la **capacidad de procesar grandes cantidades de datos de manera más eficiente y escalable**. Al utilizar servicios de almacenamiento de datos en la nube, como Amazon S3 o Google Cloud Storage y otros servicios, las organizaciones pueden almacenar grandes cantidades de datos y acceder a ellos de manera rápida y eficiente.

Además, la integración con servicios de procesamiento, gobierno, análisis y visualización de datos, como AWS Athena o Google Big-Query, permite a las empresas realizar consultas y análisis complejos de datos sin tener que preocuparse por la infraestructura.

En este sentido, la computación *serverless* también se puede utilizar para **almacenar y servir contenido estático**, como imágenes, videos o archivos de texto.

Al usar servicios de almacenamiento en la nube, como AWS S3 o Google Cloud Storage, las empresas pueden guardar grandes cantidades de contenido y acceder a ellos de manera rápida y eficiente mediante funciones; y no solo eso, estos servicios permiten desplegar contenido estático web, cachear y distribuir contenido alrededor del mundo.

En Nequi utilizamos varias prácticas de la arquitectura de *serverless*. Por ejemplo, nuestra estrategia de datos es 100% *serverless* y nos apalancamos en los servicios de AWS para el almacenamiento (S3), procesamiento (Glue), construcción modelos (SageMaker) y visualización de datos (QuickSight).

Al estar sobre un esquema de *serverless*, en la medida en que vamos construyendo y consolidando datos, optimizamos costos, debido a que se factura por uso. En las primeras etapas de implementación de dicha estrategia de datos, nuestros costos eran bajos, mientras íbamos avanzando y pasando a producción los costos se iban adaptando al uso; en un proyecto tradicional, no tendríamos esta flexibilidad ni podríamos crecer de manera elástica como nos ofrece *serverless*.

Factor clave: control de gastos

Aunque *serverless* ofrece las ventajas que hemos mencionado, la eficiencia es un gran desafío para muchas organizaciones. Las empresas solo pagan por el tiempo de ejecución, lo que significa que, teóricamente, los costos deberían ser menores. A pesar de ello, en la práctica, **es fácil perder el control de los costos si no se implementan mecanismos adecuados de control**, monitoreo y optimización constante, idealmente que estos controles se puedan automatizar.

Cuando en las compañías nos enfrentamos a algunas funciones que no fueron diseñadas de forma eficiente, tardan más de lo necesario en ejecutarse o significan un cambio drástico de la demanda, vemos que esto puede aumentar significativamente los costos; por eso, hacemos optimizaciones para mejorar los rendimientos. En nuestra organización tomamos decisiones con datos y hechos, hacemos las pruebas respectivas que nos permitan tener la información adecuada para implementar las acciones necesarias, en caso tal de que las funciones se ejecuten con demasiada frecuencia o durante periodos de baja demanda.

Serverless ofrece muchas ventajas, también presenta desafíos en el gobierno, implementación y en el control de costos. Sin embargo, al tener buenas prácticas de FinOps las empresas pueden aprovecharlo al máximo. Algunas prácticas que

tenemos en Nequi que impactan positivamente la adopción de esta arquitectura son: diseños eficientes, monitoreo constante, políticas de escalado, capacidades reservadas, uso de alarmas, políticas de retención de logs, gestión del ciclo de vida de los datos.

Desmintiendo el *vendor locking* en *serverless*

Una de las principales preocupaciones de *serverless* es el problema del *vendor locking*. El experto Gregor Hohpe lo describe como “la dificultad de cambiar de un proveedor a otro competidor. Por ejemplo, si migrar de Siebel CRM a Salesforce CRM o de una base de datos IBM DB2 a Oracle le costará un brazo y una pierna, está “atrapado””. En el contexto de *serverless*, el *vendor locking* puede ser un gran desafío, ya que las empresas dependen en gran medida de los proveedores de servicios en la nube para proporcionar la infraestructura y plataforma para ejecutar sus aplicaciones.

Una forma de abordar este desafío es a través del uso de arquitecturas hexagonales o arquitecturas limpias. Estas se centran en separar la lógica de negocio de la infraestructura, lo que permite tener componentes agnósticos sin tener que reescribir su código, cambiar su arquitectura o sin hacer un cambio significativo.

Además, estas arquitecturas promueven la modularidad y la inde-

pendencia de la plataforma, lo que facilita la integración con otros servicios y herramientas de la nube. En resumen, el uso de arquitecturas hexagonales o arquitecturas limpias puede ayudar a mitigar este riesgo, en especial en *serverless*, y aumentar su flexibilidad para aprovechar lo mejor de cada proveedor de nube, o utilizar múltiples proveedores.

Soft limit: ¿una advertencia amigable?

En el mundo *serverless*, el *Soft limit* hace referencia a las restricciones impuestas por el proveedor de nube en cuanto a los recursos que un servicio *serverless* puede consumir. Estos límites generalmente fueron construidos para evitar que un componente mal diseñado o un error en el código consuma recursos en exceso y afecte el rendimiento general del sistema. Por lo tanto, las restricciones son como advertencias amigables que buscan proteger la estabilidad del entorno.

Cuando un servicio alcanza su *Soft limit*, el proveedor de nube podría tomar medidas para mitigar el impacto negativo, como ralentizar las solicitudes entrantes, limitar la capacidad de procesamiento o limitar su escalabilidad. Esto puede afectar negativamente la experiencia del usuario, provocando retrasos en las respuestas y una disminución en la eficiencia de la aplicación, es importante tener una buena estrategia de monitoreo de re-

cursos; y con ello sabe cómo jugar con los Soft Limit.

Hard limit: donde la advertencia se convierte en bloqueo

El "Hard Limit", por otro lado, es la barrera final que evita que un servicio *serverless* exceda ciertos límites de recursos. Cuando se alcanza el *Hard limit*, la plataforma puede detener completamente la ejecución de la función o incluso suspender todo el servicio *serverless*. Esta situación puede ocurrir debido a un tráfico inesperadamente alto, un bucle infinito en el código o cualquier otro escenario que sobrepase los límites preestablecidos.

El *Hard limit*, aunque puede parecer drástico, es esencial para garantizar la estabilidad y seguridad del entorno *serverless*. Evita que un solo servicio o función monopolice los recursos y deje a otros servicios sin la capacidad de procesamiento necesaria. Es crucial cuando se está evaluando una solución, tener en cuenta estos límites para tomar la decisión antes de implementarla.

Conclusión

En última instancia, el éxito en la adopción de *serverless* en Nequi radica en la alineación con la estrategia organizacional y las necesidades del negocio, una gestión consciente de los recursos y una constante búsqueda de mejores prácticas y soluciones adaptadas a las necesidades específicas de cada proyecto. Al mantenerse infor-

mas y actualizadas sobre las últimas tendencias y herramientas, las organizaciones pueden liberar todo el potencial de *serverless* para seguir innovando y prosperando en el competitivo mundo tecnológico de hoy.

Serverless no es solo una tendencia, sino una verdadera revolución tecnológica que está cambiando la forma en que construimos y ejecutamos aplicaciones en la nube. Adoptando una mentalidad ágil, centrada en la optimización y en la resolución creativa de desafíos, las empresas pueden aprovechar plenamente su poder y liderar el camino hacia un futuro donde la innovación y la eficiencia se fusionen para crear experiencias excepcionales para los clientes y usuarios finales. Con el paso del tiempo la tecnología evoluciona, se adapta, se optimiza, se vuelve asequible y escalable.

Referencias

Amazon Web Services. (15 de Septiembre de 2021). Optimizing Enterprise Economics with Serverless Architectures. <https://docs.aws.amazon.com/whitepapers/latest/optimizing-enterprise-economics-with-serverless/optimizing-enterprise-economics-with-serverless.html>.

Microsoft Azure. (s.f.). Multicloud solutions with the Serverless Framework. [Sitio web]. Recuperado el 15 de septiembre de 2021, de <https://azure.microsoft.com/en-us/overview/serverless-computing/>

MartinFlowler.com. Don't get locked up into avoiding lock-in. [Sitio web].

Nicolás Edilberto Rozo Espinosa. Ingeniero de Sistemas de la Universidad ECCI, Especializado en Arquitectura Empresarial de Software de la Pontificia Universidad Javeriana; Certificado como AWS Solution Architect Associate; se desempeña actualmente como director de Arquitectura, I+D+I y DevSecOps en Nequi. Cuenta con más de 10 años de experiencia en la industria de tecnología en los sectores Consultoría, Telecom, Pensiones, Asegurador y Financiero. Tiene estudios en liderazgo, desarrollo de capacidades ejecutivas y servicio, en instituciones como la Universidad de los Andes y Eclasss.

Jeison Vergara Vargas

DOI: 10.29236/sistemas.n168a3

Profesional con amplia experiencia para abordar diferentes asuntos relacionados con el “Paradigma Serverless”.

Sara Gallardo M.

En su amplio recorrido como arquitecto de software en los sectores público y privado del país, Jeison Vergara Vargas ha dejado huella. Así mismo en el exterior como Doctorat en Informatique et Architectures Numériques, Université Bretagne Sud, Francia; profesor visitante en Université de Poitiers, Francia (2017) y profesor visitante, EPI-TECH París, Francia (2019).

Considera una fortuna ser egresado de la Universidad Nacional de Colombia, en donde se desempeña como profesor auxiliar del Departamento de Ingeniería de Sistemas e Industrial, sede Bogotá. Además, es investigador doctoral del laboratorio IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires) en Francia.



Sus áreas de interés contemplan arquitectura de software, lenguajes de dominio específico, métodos formales e ingeniería de software dirigida por modelos. Así mismo es líder de la línea de investigación en Arquitectura de Software del Grupo de Investigación en Ingeniería de Software (CoISWE).

A las actividades profesionales les suma un permanente deseo por conocer diferentes culturas del mundo y su gastronomía, combinado con otras lecturas sobre arte, astronomía y un espacio para la natación y los juegos de mesa.

Revista Sistemas: *¿Cómo se podría materializar el concepto de ciberseguridad en el paradigma "serverless" cuando sólo existen "funciones" que se usan? ¿Sería sólo lo correspondiente al desarrollo de aplicaciones o funciones seguras?*

Jeison Vergara Vargas

La computación tipo «serverless» es un modelo basado en la nube, en donde los recursos de infraestructura (cómputo y red, principalmente) son gestionados por el proveedor del servicio, ocultando su identidad a los clientes. En este modelo, los clientes no tienen la responsabilidad de la creación, configuración y mantenimiento de estos recursos de infraestructura.

La ciberseguridad es un área que busca garantizar con énfasis la confidencialidad, la integridad y la disponibilidad de los recursos manejados desde una perspectiva computacional: datos, software y hardware. Independientemente del uso del modelo *serverless*, la seguridad mantiene retos que abarcan, entre otras cosas:

- Seguridad a nivel de la arquitectura del software: diseño estructural del sistema, seguridad a nivel de componentes y conectores.
- Seguridad a nivel de diseño detallado del software: aspectos de seguridad durante la codificación de los componentes del sistema, incluyendo FaaS (funciones como servicio).

- Seguridad a nivel de flujos de datos: aspectos de seguridad cuando los datos viajan de un componente a otro, o de un sistema a otro.

RS: *Se ha venido trabajando en los últimos años en arquitecturas basadas en microservicios como estrategia arquitectural para que las soluciones sean más escalables y flexibles. ¿Cómo encaja una arquitectura serverless en esta estrategia?*

JVV: Las arquitecturas de software basadas en microservicios se caracterizan por su estructuración a partir de un conjunto de componentes lógicos; es decir, componentes con responsabilidad de lógica de negocio dentro de un sistema de software, los cuales se implementan por medio de algún lenguaje de programación de propósito general. Estos componentes son pequeños y poseen una responsabilidad funcional única. En este sentido, dado que el modelo *serverless* promueve el uso de funciones como servicio, estas funciones pueden ser usadas para la implementación de componentes de tipo microservicio, por lo cual se convierten en una alternativa bastante atractiva para el diseño de arquitecturas de software distribuidas y basadas en la nube.

RS: *¿Qué opina acerca de la dependencia que se genera entre la arquitectura de un sistema bajo el paradigma *serverless* y los proveedores de soluciones de nube?*

JVV: La dependencia que se pueda generar es relativa. En el caso de un buen diseño arquitectónico, dado que la arquitectura de un sistema debe ser agnóstica a la tecnología, la dependencia se puede reducir. Una de las formas más comunes es generando redundancia entre los componentes. De esta forma, un componente puede estar ejecutándose como una función en el proveedor A y, a su vez, como otra función en el proveedor B. Las dos funciones son equivalentes, pero se ejecutan de manera independiente.

RS: *¿Es posible no tener que "caerse" con un proveedor de nube específico para una solución bajo el paradigma *serverless*?*

JVV: Sí, existen varias alternativas a partir de variaciones en el diseño de la arquitectura del sistema. Un ejemplo es el descrito en la pregunta anterior.

RS: *¿Cuáles son las mayores ventajas del paradigma *serverless* con respecto a los paradigmas tradicionales?*

JVV: Definitivamente la agilidad en la construcción del componente.

RS: *Los desarrolladores de software ya no tendrían el control del uso de los recursos de hardware en el paradigma *serverless*, por lo cual, los atributos de calidad arquitecturales relacionados con desempeño ya no son gobernados por los desarrolladores y arquitectos. ¿Cuáles son los retos a los que*

deben enfrentarse quienes construyen soluciones de software bajo este nuevo concepto?

JVV: Hay varios retos a tener en cuenta a la hora de diseñar y construir sistemas de software bajo el modelo *serverless*, entre ellos:

- Seguridad durante la ejecución de componentes basados en funciones.
- Gestión eficiente en el uso de recursos proveídos por el proveedor de nube.
- Proceso de pruebas adecuado: no se tiene control sobre pruebas de nivel 3 (sistema – no funcionales).
- Latencia: tiempos de respuesta variables.
- Manejo de estado.

RS: *¿Este paradigma *serverless* es una evolución normal de la arquitectura de software? ¿O es una "moda"? ¿Cómo se ve este nuevo paradigma ante los ojos de la comunidad académica de arquitectura de software e ingeniería de software?*

JVV: No, el modelo *serverless* no corresponde a una evolución normal de la arquitectura de software.

La evolución de la arquitectura se fundamenta sobre cambios a nivel estructural; es decir, a nivel de los componentes y conectores que conforman el sistema, y cómo estos se comunican entre sí; no en términos de cambios a nivel de despliegue. El uso del modelo *serverless* no tiene trascendencia a nivel estructural, pero sí a nivel de despliegue.

RS: *¿Qué tipo de soluciones de software se ven beneficiadas de este nuevo paradigma, ¿cuáles no?*

JVV: Sistemas de software de procesamiento de datos en tiempo real, tareas en segundo plano, algoritmos de inteligencia artificial que automaticen procesos. Sistemas que requieran distribución en su arquitectura (microservicios y brokers de mensajes). 🌐

Sara Gallardo M. Periodista comunicadora, universidad Jorge Tadeo Lozano. Ha sido directora de las revistas Uno y Cero, Gestión empresarial y Acuc Noticias. Editora de Aló Computadores del diario El Tiempo. Redactora en las revistas Cambio 16, Cambio y Clase Empresarial. Coautora del libro "Lo que cuesta el abuso del poder". Ha sido corresponsal de la revista Infochannel de México; de los diarios La Prensa de Panamá y La Prensa Gráfica de El Salvador y corresponsal de la revista IN de Lanchile e investigadora en publicaciones culturales. Se ha desempeñado también como gerente de Comunicaciones y Servicio al Comensal en Inmaculada Guadalupe y amigos en Cía. S.A. (Andrés Carne de Res) y editora de Alfaomega Colombiana S.A.; en la actualidad es asesora y editora en escritura y producción de libros. Es editora de esta revista.

Encuesta Nacional

Paradigma

serverless:

función como

servicio

El presente estudio es el resultado de una encuesta a través de Internet, dirigida a profesionales del sector TI a nivel nacional. Se basó en un formulario virtual de diez preguntas y la respuesta fue de 71 encuestados a nivel nacional.

DOI: 10.29236/sistemas.n168a4

Esta investigación busca evaluar de manera cuantitativa el impacto que tienen las aplicaciones móviles en el día a día de las personas, con el objetivo de revelar aspectos relacionados con privacidad de la información, frecuencia de uso e impacto, entre otros.

Metodología

La encuesta se realizó a través de Internet, dirigida a profesionales del sector TI a nivel nacional. Se basó en un formulario virtual de diez preguntas.

Entre los encuestados hay una mezcla representativa de organizaciones grandes, pequeñas y medianas de los sectores de industria, servicios financieros, telecomunicaciones, tecnología, desarrollo de software, Gobierno y otros.

Resultados

Se realizaron nueve preguntas dirigidas a profesionales de TI y con-

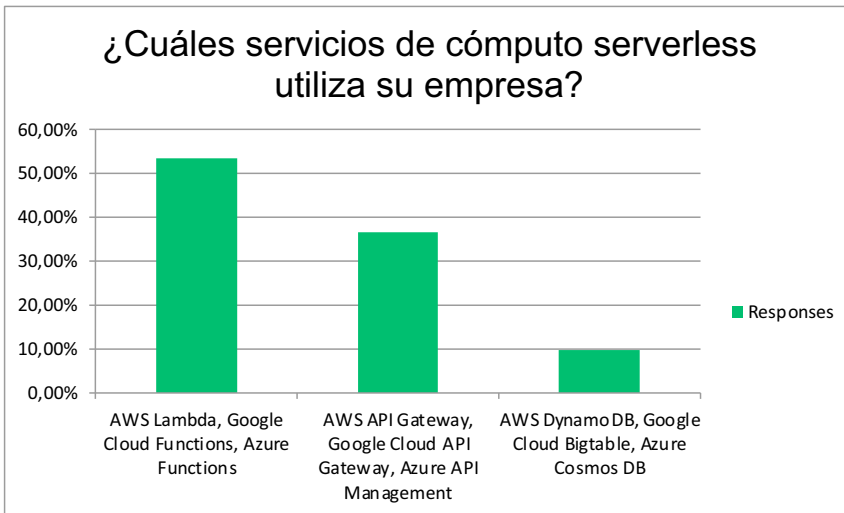
forme a las respuestas se registraron las siguientes distribuciones:

1. ¿Cuáles servicios de cómputo serverless utiliza su empresa? (Gráfica 1).

El **53.5%** de los encuestados manifestó que utiliza con mayor frecuencia los servicios **AWS Lambda, Google Cloud Functions y Azure Functions**. Los servicios **AWS API Gateway, Google Cloud API Gateway, y Azure API Management** le siguen en frecuencia con **36.6%**. Otros servicios como **AWS DynamoDB, Google Cloud Bigtable y Azure Cosmos**

Tabla 1. Ficha técnica

Nombre de la investigación	Paradigma serverless: función como servicio
Objetivo general	Identificar el nivel de apropiación de los servicios de computación serverless en las empresas
Realización	Asociación Colombiana de Ingenieros de Sistemas – ACIS
Tipo de investigación	Muestreo probabilístico
Población objetivo	Profesionales del sector de TI de Colombia.
Número de encuestados	71
Técnica de recolección	Encuesta online
Fecha de recolección	7 de septiembre al 22 de septiembre de 2023



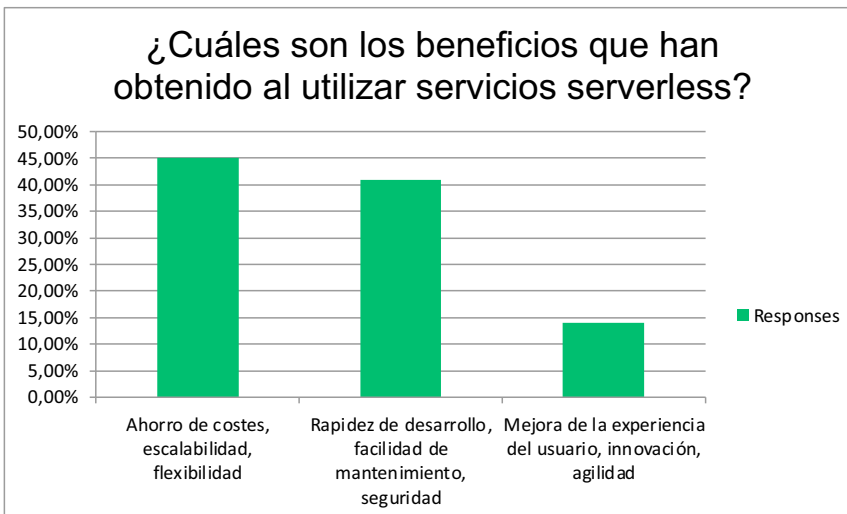
Gráfica 1.

Servicios de cómputo serverless utilizados en las empresas.

DB son menos utilizados, alcanzando tan solo un **9.8%**.

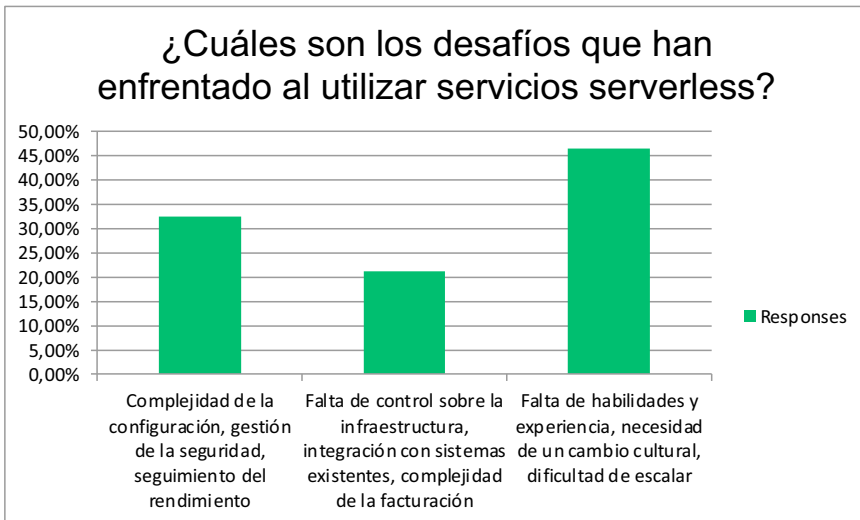
2. ¿Cuáles son los beneficios que han obtenido al utilizar servicios serverless? (Gráfica 2).

El **45%** del total de encuestados respondió que obtienen **ahorro de costes, escalabilidad y flexibilidad** cuando utilizan servicios serverless. El **40.8%** considera que los beneficios son **rapidez de desa-**



Gráfica 2.

Beneficios que se obtienen al utilizar servicios serverless



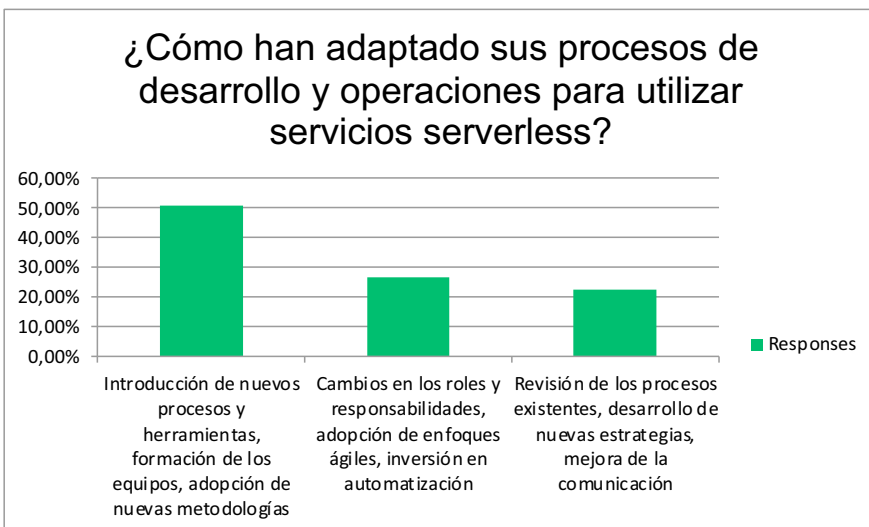
Gráfica 3.

Desafíos que han enfrentado las empresas al utilizar servicios serverless.

rollo, facilidad de mantenimiento y seguridad. El 14% de los encuestados respondió que los beneficios se centran en la **mejora de la experiencia del usuario, innovación y agilidad.**

3. ¿Cuáles son los desafíos que han enfrentado al utilizar servicios serverless? (Gráfica 3).

El 46.4% de los encuestados indicó que los desafíos que han enfrenta-



Gráfica 4.

Acciones adoptadas en los procesos de las empresas al implementar servicios serverless.

do son la **falta de habilidades y experiencia, necesidad de un cambio cultural y la dificultad de escalar**. El **32.3%** de los encuestados indicó que los desafíos que han enfrentado son la **complejidad de la configuración, gestión de la seguridad, seguimiento del rendimiento**. El **21.1%** manifestó que los desafíos son la **falta de control sobre la infraestructura, integración con sistemas existentes y la complejidad de la facturación**.

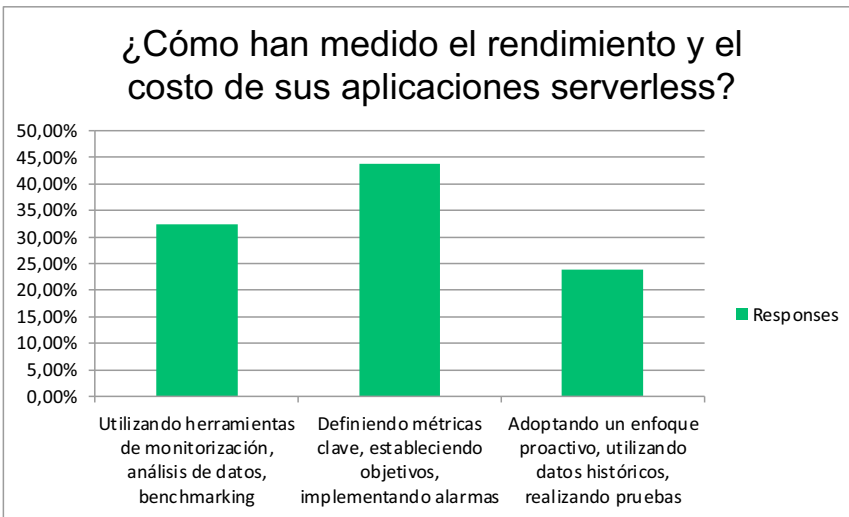
4. ¿Cómo han adaptado sus procesos de desarrollo y operaciones para utilizar servicios serverless? (Gráfica 4).

El **50.7%** de los encuestados manifiesta que han adaptado sus procesos con la **introducción de nuevos procesos y herramientas,**

formación de los equipos, y adopción de nuevas metodologías. El **26.7%** manifiesta que han realizado **cambios en los roles y responsabilidades, adopción de enfoques ágiles, e inversión en automatización**. El **22.4%** de los encuestados indica que han **realizado revisión de los procesos existentes, desarrollo de nuevas estrategias, y mejora de la comunicación**.

5. ¿Cómo han medido el rendimiento y el costo de sus aplicaciones serverless? (Gráfica 5).

El **43.6%** de los encuestados lo ha hecho definiendo métricas clave, estableciendo objetivos, implementando alarmas. El **32.3%** lo ha hecho utilizando herramientas de monitorización, análisis de datos, benchmarking. El **23.9%** lo ha he-



Gráfica 5.

Estrategias de medición del rendimiento y costo de aplicaciones serverless.

cho adoptando un enfoque proactivo, utilizando datos históricos, y realizando pruebas.

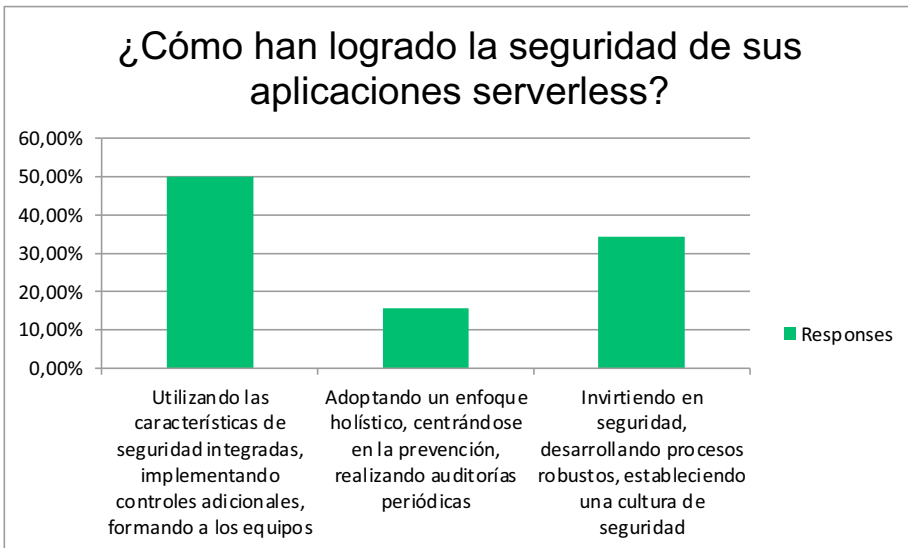
6. ¿Cómo han logrado la seguridad de sus aplicaciones serverless? (Gráfica 6).

El 50% de los encuestados lo han logrado utilizando las características de seguridad integradas, implementando controles adicionales, formando a los equipos. El 34,2% lo han logrado utilizando las características de seguridad integradas, implementando controles adicionales, formando a los equipos. El 15,7% lo han logrado utilizando las características de seguridad integradas, implementando controles adicionales, y formando a los equipos.

7. ¿Cómo han escalado sus aplicaciones serverless para satisfacer las demandas de sus usuarios? (Gráfica 7).

El 43.6% de los encuestados lo logra escalar las aplicaciones utilizando características de escalado automático, adoptando un enfoque proactivo, y realizando pruebas de carga. El 42.2% lo hace definiendo métricas clave, estableciendo objetivos, e implementando alarmas. El 14% lo hace invirtiendo en infraestructura, desarrollando procesos robustos, y estableciendo una cultura de escalabilidad.

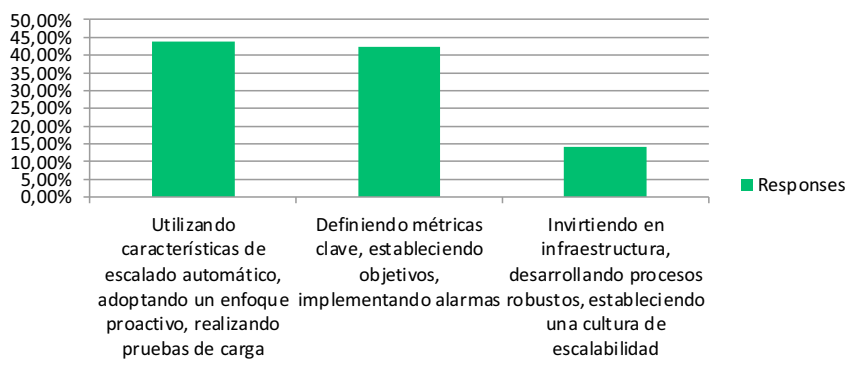
8. ¿Cómo han integrado sus aplicaciones serverless con otras aplicaciones y sistemas existentes?



Gráfica 6.

Estrategias para lograr la seguridad en las aplicaciones serverless.

¿Cómo han escalado sus aplicaciones serverless para satisfacer las demandas de sus usuarios?



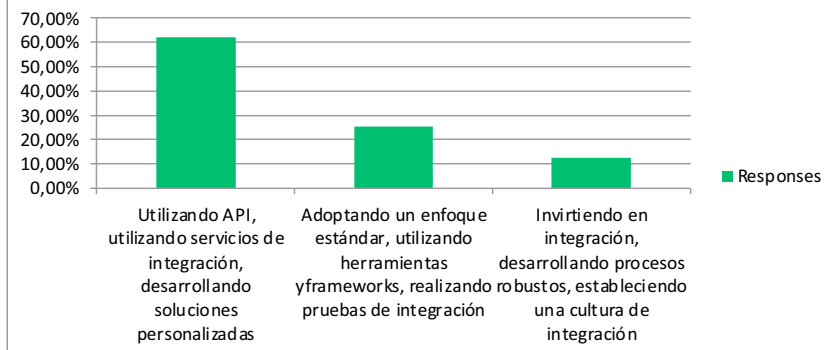
Gráfica 7.

Acciones realizadas para escalar las aplicaciones basada en el paradigma serverless.

El **61.9%** integra las aplicaciones serverless con otras aplicaciones existentes **utilizando API, utilizando servicios de integración, y desarrollando soluciones personalizadas**. El **25.31%** lo hace **utilizando API, utilizando servicios**

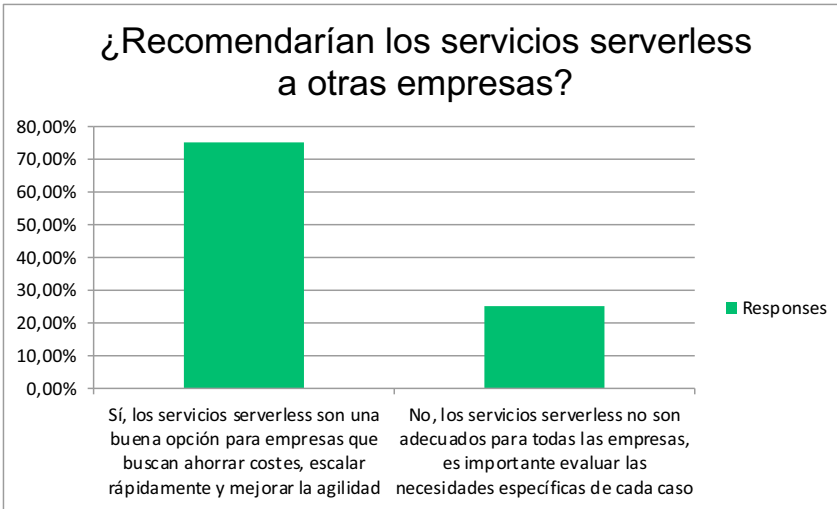
de integración, y desarrollando soluciones personalizadas. El **12.6%** lo hace **invirtiendo en integración, desarrollando procesos robustos, y estableciendo una cultura de integración**.

¿Cómo han integrado sus aplicaciones serverless con otras aplicaciones y sistemas existentes?



Gráfica 8.

Estrategias de integración de aplicaciones serverless con otras aplicaciones.



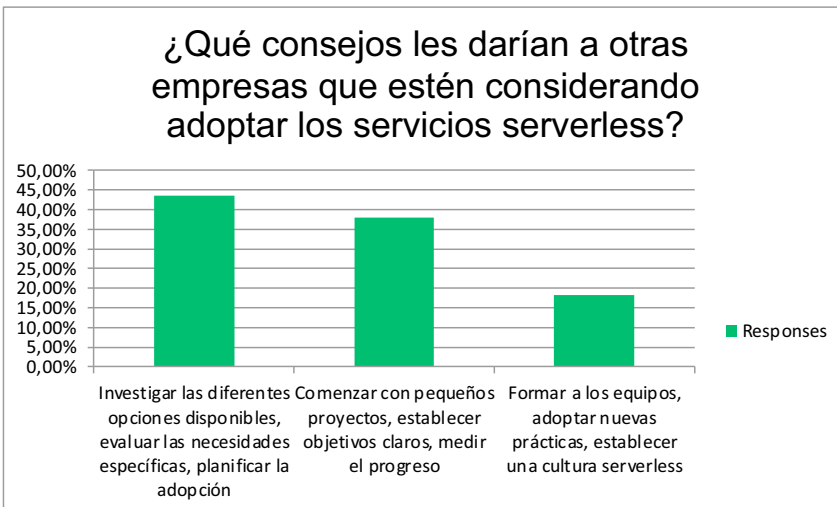
Gráfica 9.

Lo que opinan los encuestados frente a si recomendarían los servicios serverless.

9. ¿Recomendarían los servicios serverless a otras empresas? (Gráfica 9).

ya que son una buena opción para empresas que buscan ahorrar costes, escalar rápidamente y mejorar la agilidad. El **25%** de los encuestados **no** recomienda ya que no son adecuados para todas las empre-

El **75%** de los encuestados **sí** recomienda los servicios serverless



Gráfica 10.

Consejos que dan los encuestados a quienes estén considerando adoptar servicios serverless.

sas, y consideran que es importante evaluar las necesidades específicas de cada caso.

10. ¿Qué consejos les darían a otras empresas que estén considerando adoptar los servicios serverless? (Gráfica 10).

El **43.6%** de los encuestados recomiendan **investigar las diferentes opciones disponibles, evaluar las necesidades específicas, y planificar la adopción**. El **38%** recomienda **comenzar con pequeños proyectos, establecer objetivos claros, medir el progreso**. El **18.3%** recomienda **formar a los equipos, adoptar nuevas prácticas, y establecer una cultura serverless**.

Conclusiones

La encuesta realizada revela interesantes aspectos de cómo los servicios de computación serverless están impactando a las organizaciones. Los resultados de la encuesta propician varias conclusiones y asuntos para tener en cuenta:

- La mayoría de las empresas encuestadas están utilizando servicios serverless, con un enfoque principal en AWS Lambda, Google Cloud Functions y Azure Functions. También se observa una adopción significativa de servicios de gestión de API.
- Las empresas encuentran beneficios significativos al utilizar servicios serverless, con ahorro

de costos, escalabilidad y flexibilidad como los principales impulsores. Además, la rapidez de desarrollo, facilidad de mantenimiento y seguridad también son valorados.

- Aunque hay beneficios notables, las empresas enfrentan desafíos, siendo la falta de habilidades y experiencia, así como la necesidad de un cambio cultural, los obstáculos más prominentes. La complejidad de configuración y gestión de la seguridad también son áreas de preocupación.
- Las empresas están tomando medidas para adaptar sus procesos de desarrollo y operaciones para incorporar servicios serverless, a través de la introducción de nuevos procesos, formación de equipos y adopción de metodologías ágiles.
- Las empresas están utilizando una variedad de métodos para medir el rendimiento y el costo de sus aplicaciones serverless, incluyendo definir métricas clave, utilizar herramientas de monitorización y análisis de datos, y adoptar un enfoque proactivo basado en datos históricos.
- Las empresas están logrando seguridad y escalabilidad a través de una combinación de características de seguridad integradas, controles adicionales y formación de equipos. Además,

se observa un enfoque proactivo en la escalabilidad.

- La mayoría de las empresas están integrando sus aplicaciones serverless con otras aplicaciones y sistemas existentes utilizando API y servicios de integración. Esto sugiere una preocupación por la interoperabilidad y la coexistencia con sistemas existentes.
- La gran mayoría de los encuestados recomiendan servicios serverless, señalando su capacidad para ahorrar costos, escalar rápidamente y mejorar la agilidad. Sin embargo, un pequeño porcentaje advierte que no son adecuados para todas las empresas y sugiere una eva-

luación detallada antes de la adopción.

- El resultado de la encuesta sugiere una aproximación cuidadosa y planificada, que puede incluir investigar opciones disponibles, evaluar necesidades específicas y comenzar con proyectos pequeños antes de expandir la adopción definitiva.

En resumen, la adopción de servicios serverless está ganando tracción en las organizaciones, y aunque hay beneficios notables, existen desafíos que deben ser abordados de manera proactiva. La planificación cuidadosa, la formación y la adaptación de procesos son cruciales para aprovechar al máximo los servicios serverless. 🌐

Jorge Eliécer Camargo Mendoza, PhD. Decano de la Facultad de Ingeniería de Sistemas de la Universidad Antonio Nariño. Ingeniero de Sistemas de la Universidad Nacional de Colombia, Magíster en Ingeniería de Sistemas y Computación de la Universidad de los Andes. Doctor en Ingeniería de Sistemas y Computación de la Universidad Nacional de Colombia (Tesis Laureada). Cuenta con más de 100 publicaciones científicas en revistas, libros y conferencias internacionales relacionadas con Big Data, Machine Learning, Inteligencia Artificial y Ciberseguridad.

Fabio González Osorio, PhD. Fabio Augusto González O. Profesor titular del Departamento de Ingeniería de Sistemas e Industrial de la Universidad Nacional de Colombia, sede Bogotá. Es ingeniero de Sistemas y Magíster en Matemáticas de la Universidad Nacional de Colombia, MSc y PhD en Ciencias de la Computación de la Universidad de Memphis, Estados Unidos. Lidera el grupo de investigación MindLab. Su trabajo de investigación se enfoca en los fundamentos de aprendizaje de máquina y recuperación de información y su aplicación en el análisis de imágenes y texto, visión por computador y minería de datos.

Paradigma Serverless: ¿evolución de la computación en la nube?

DOI: 10.29236/sistemas.n168a5

Para analizar los pormenores de este nuevo paradigma, la revista Sistemas reunió a algunos expertos.

Sara Gallardo M.

Con la moderación de Jorge Eliécer Camargo, editor técnico de este número de la publicación y la asistencia del director de la revista, Jeimy J. Cano M., se abrió la puerta al enriquecedor diálogo.

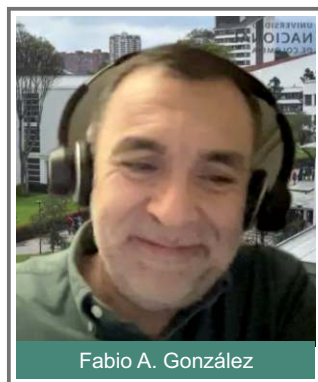
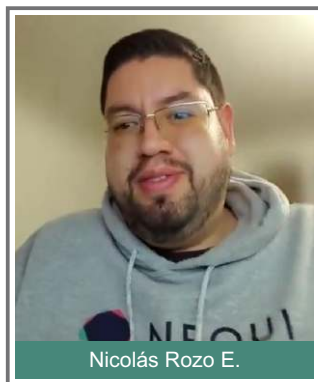
Después de la bienvenida a los invitados Nicolás Rozo Espinosa, director de Arquitectura de Nequi y

Jonatan Ponzó de Arquitectura, DevEx y cosas de Rappi, el editor técnico señaló: “a manera de introducción en la computación serverless, los involucrados en el tema no se preocupan por temas de infraestructura relacionados como la memoria, procesamiento o almacenamiento; sino que en la medida en que se van utilizando tales servi-

cios están ubicados físicamente en las nubes, a diferencia de la computación basada en servidor en la que tenemos que atender el aprovisionamiento entre las máquinas, si tenemos una gran cantidad de usuarios conectándose a esa aplicación llegará un momento en que empezará a rechazar conexiones y, en tal sentido, tendremos que pensar en escalar esas máquinas.

En esto se centra el tema que estamos abordando en este número de la revista. Dentro de esto aparecen subcategorías y en los proveedores de nube uno encuentra las funciones como servicios, lugares

en donde ponemos código en algún lenguaje de programación, se exponen como tal a través de un servicio web consumido por los clientes. Pero, también existe la computación serverless a nivel de almacenamiento, el hecho de guardar objetos como tal en estas nubes también se pueden manejar en ese paradigma serverless. Y, la otra subcategoría en las nubes es a nivel de servicios de integración de aplicaciones, es decir, todo el manejo de mensajes que envían las aplicaciones a través de servicios web, servicios de *backend* y una gran cantidad de servicios disponibles en la nube”.



Jorge Eliécer Camargo

Editor Técnico

Miembro Comité Editorial

¿Cuáles son los riesgos de seguridad de la computación serverless?



Nicolás Rozo E.

Gerente de Arquitectura & i+I+D

Nequi (By Bancolombia)

Vemos que la tecnología avanza de una forma acelerada, exponencial. En este ambiente es posible manejar los retos que teníamos en el pasado, pero resultan unos nuevos que son inevitables. De manera que es importante desde el lado de serverless entender ese modelo de responsabilidad compartida, ahí arranca. Desde este modelo se define la responsabilidad del proveedor, versus la de los clientes. Son muchos los riesgos relacionados con la codificación y es necesario dar un correcto manejo para atender la información sensible y hacer el análisis de código estático.

Estos son los retos que existen en la actualidad en la computación y no se refieren a serverless puntualmente, entorno que debe verse como un patrón arquitectónico que entra a acelerar o agilizar la adopción de una solución de los objetivos de negocio.

En el entorno serverless uno se refiere más a la infraestructura, a prácticas de desarrollos seguros; es necesario contar con un buen esquema de seguridad y al mismo tiempo, poder aplicar esos controles previamente. Una mala configuración está entre los riesgos.

Jonatan Ponzo

Director Arquitectura, DevEx y cosas

Rappi

Estoy de acuerdo con lo ya planteado. Lo que puedo agregar es que, en ciertos casos de uso, no tanto orientados al computing, sino más bien a datos, nos resuelve o simplifica la tarea, porque los mismos proveedores de estas arquitecturas cumplen con tales desarrollos que los profesionales de la arquitectura tenían que desarrollar y que hoy en día al utilizar esos servicios ya se está cumpliendo. Se trata también de abstraer que el desarrollador no tenga qué preocuparse por todo ese tipo de cosas, en la medida en que hay desarrolladores encargados de las bases de datos o un equipo para consumir de una manera segura esos servicios. De tal manera ellos colaboran de una forma importante en este

entorno. Serverless entonces se orienta a simplificar y abstraer complejidad de configuración, sobre todo a esos desarrolladores resolviendo el cloud u otras plataformas.

Fabio A. González
*Miembro Comité Editorial
Revista Sistemas*



Estos asuntos han cambiado la forma como típicamente se construían aplicaciones y los riesgos de seguridad están presentes. Así mismo, se trata de comparar este tipo de arquitectura con otros más tradicionales y si los riesgos se incrementan o se pueden disminuir para facilitar la gestión por parte del usuario.

Jorge E. Camargo
Es muy importante algo que ustedes mencionan y es la granularidad y desde ese punto de vista manejar los riesgos es muy importante para no dejar espacios vulnerables.

La siguiente pregunta apunta a:
¿Cómo afecta la computación serverless la arquitectura de las aplicaciones, pensando en aquellas móviles o también en los videojuegos y en la lógica de los mismos? En el caso las aplicaciones como Rappi y Nequi están consumiendo este tipo de servicios.

Jonatan Ponzo
La principal ventaja es la velocidad para crecer y para escalar sin convertir a la infraestructura en un cuello de botella. En cuanto a la arquitectura de ese ecosistema de servicio, no sé si puede llegar a afectarle tanto. Facilita la velocidad para crecer. Así mismo existen otros retos, en la medida de la concurrencia en servicio necesitamos mucha más experiencia particularmente en serverless, como y en servicio encontramos algunas limitaciones. No me parece un término disruptivo frente a la arquitectura final de un ecosistema de servicios para una aplicación.

Fabio González
Una pregunta para Jonatan, en el caso de que serverless lleve a pensar en mayor modularidad o en componentes que no tienen estado, ¿eso no modifica de alguna manera la forma como funciona?

Jonatan Ponzo
Eso no tiene que ver con el paradigma serverless, sino más bien con el paradigma de servicio. En una arquitectura que por debajo tiene máquinas virtuales es necesario

estar preparado para que cualquier container muera en cualquier momento. Es más bien por una arquitectura basada en microservicios.

Jorge E. Camargo

¿Qué tipo de aplicaciones son adecuadas para una plataforma serverless?

Nicolás Rozo



Esa es una pregunta difícil, de gustos y disgustos. Desde la experiencia me parece que serverless se adapta a una plataforma, es posible encontrarle dificultades, pero en un año es posible solucionarlo. Uno de los problemas se presentaba cuando era necesario implementar soluciones que tenían que ver con el costo de la aplicación. Con las opciones que han salido se soluciona un problema que antes era un reto. Primero hay que partir de que la tecnología avanza muy rápido.

Cuando Serverless llega a un nivel de concurrencia tal, los costos pueden ser contraproducentes. El reto es poder tener las mismas herramientas. Es un caso de uso de arquitecturas y existen retos. No es un tema de arquitectura, sino de la particularidad del contexto, del negocio, de la plataforma, del caso de uso que se esté implementando.

Serverless permite implementar de una manera transparente. Existen retos dentro de los negocios particulares del caso de uso que se esté implementando.

Fabio A. González

¿Usar una plataforma serverless no se basa más en un proveedor particular?

Nicolás Rozo

Tenemos que ser agnósticos en las nubes y es posible tener una estrategia especial. La codificación debe tener unas capas que no dependan de un proveedor. Si se tiene un código agnóstico. De hecho, uno de los artículos muy interesantes sobre el tema es el Microsoft tiene una arquitectura de referencias en serverless.

Aquí es clave ver a Serverless como ese vehículo, abstraernos de la tecnología y verlo como un patrón. Si se tienen muy buenos patrones de diseño se revisan para funcionar. La estrategia multicloud está ahí y uno debe ser consciente de lo que significa para sacar un buen provecho.

No es una bala de plata que exista una única alternativa, sino todo depende de las necesidades y es una situación manejable.

Jonatan Ponzo



Coincido con lo expuesto, tiene que ver con el uso y lo veo como una herramienta más, entendiendo los beneficios de uso y las limitaciones que pueda tener. Yo tengo dos tareas programadas que corren en un período de tiempo asincrónico, puedo aprovechar muchísimo lo que encuentro en serverless.

Siendo más concreto en nuestro caso serverless está enfocado en procesos que corren en períodos determinados de tiempo o que tienen un nivel de tráfico y criticidad muy bajo, pero no en situaciones de alta concurrencia.

Jorge Camargo

¿Cómo se podrían comparar las plataformas serverless entre sí pa-

ra seleccionar la más apropiada ante los cuatro grandes proveedores de nube?

Jonatan Ponzo

Los pilares son funcionalidades que ofrecen escalabilidad y el más importante se relaciona con los costos. Entre esos cuatro pilares se basa el análisis, sin dejar de lado otra serie de requerimientos que cada compañía tiene relacionadas con integraciones o requerimientos muy particulares que hacen que el usuario se incline más por un proveedor u otro.

Nicolás Rozo

De acuerdo con lo planteado y es necesario observar un caso de negocio de cara a los beneficios que uno está buscando. Adicionalmente, es necesario elegir la mejor plataforma. Los cuatro grandes tienen exactamente el mismo portafolio, los precios son muy similares, de manera que resultan otros criterios que son los que se relacionan con la operación, de cara a los expertos en el tema. De ahí la consulta entre las diferentes opciones y las preguntas que resultan con base en diferentes cuestionamientos.

La decisión trasciende los aspectos técnicos, burocráticos y de operaciones, entre otros. Existen variables que no son técnicas y se debe buscar un aliado tecnológico en la búsqueda de apoyo y beneficios. La decisión trasciende los aspectos técnicos, pero es importante tenerlo muy presente.

Jorge E. Camargo

¿Cuál es el futuro de la computación serverless? ¿Se trata de un tema de moda?

Jonatan Ponzo

No creo que sea una moda; la evolución lógica con todo lo difícil que es predecir en tecnología, cada uno tenía sus propios datacenter.

El paso siguiente fueron las máquinas virtuales, luego los containers, provisionarlos sobre máquinas virtuales. Este es el paso más lógico, observar lo que hay debajo. Finalmente creo que no se está en el nivel de madurez suficiente como para reemplazar por completo el paradigma anterior, sobre todo para aplicaciones de alta concurrencia. En serverless todavía hay incertidumbre y hasta que los proveedores nos den la tranquilidad y nos quiten la preocupación por performance. La tranquilidad que proporciona serverless es muy poderosa, es oro hoy en día con la complejidad en la contratación de personal.

Todavía no se alcanza la madurez suficiente para su puesta en marcha frente a la oferta existente.

Nicolás Rozo E.

Sobre este aspecto nos referíamos al comienzo de este encuentro, tiene que ver con un factor exponencial. Conozco bancos altamente transaccionales que le apostaron a Serverless. La pregunta es cómo elijo la tecnología que mejor me sirva. No es una moda, es parte de afinamiento y de evolución lógica.

El futuro es masificar esa adopción que se convertirá en esa realidad, de cara a tecnologías emergentes de este portafolio que crece en alternativas y se van mejorando.

En poco tiempo el panorama se verá más claro y tendremos más seguridad y opciones para tomar decisiones.

No se trata de satanizar una tecnología, se trata de afinarla. 🌐

Sara Gallardo M. Periodista comunicadora, universidad Jorge Tadeo Lozano. Ha sido directora de las revistas *Uno y Cero*, *Gestión empresarial* y *Acuc Noticias*. Editora de *Aló Computadores* del diario *El Tiempo*. Redactora en las revistas *Cambio 16*, *Cambio y Clase Empresarial*. Coautora del libro "Lo que cuesta el abuso del poder". Ha sido corresponsal de la revista *Infochannel* de México; de los diarios *La Prensa* de Panamá y *La Prensa Gráfica* de El Salvador y corresponsal de la revista *IN* de Lanchile e investigadora en publicaciones culturales. Se ha desempeñado también como gerente de *Comunicaciones y Servicio al Comensal* en *Inmaculada Guadalupe* y amigos en *Cía. S.A. (Andrés Carne de Res)* y editora de *Alfaomega Colombiana S.A.*; en la actualidad es asesora y editora en escritura y producción de libros. Es editora de esta revista.

Serverless Computing ("Computación sin servidor")

DOI: 10.29236/sistemas.n168a6

Reflexiones básicas sobre seguridad y control.

Resumen

La computación sin servidor (CSS) es parte de la evolución de la computación en la nube. Este nuevo paradigma de computación establece oportunidades relevantes para las organizaciones de cara a su estrategia digital para concretar iniciativas ágiles y eficientes que den cuenta de las experiencias que exigen sus clientes en la actualidad. Sin perjuicio de lo anterior, la CSS implica un alto acoplamiento e interacción de sus diferentes componentes lo que aumenta la complejidad de su ejecución y, por tanto, los riesgos de seguridad que pueden terminar comprometiendo tanto la infraestructura, como los datos de las personas. En este sentido, este artículo hace una revisión básica de las oportunidades y retos de seguridad y control de la CSS como una primera aproximación conceptual para mantener en foco no sólo sus ventajas, sino disuadir al adversario desde la seguridad de la nube y en la nube para hacer más resistentes las propuestas digitales en este nuevo ecosistema tecnológico.

Palabras claves

Computación en la nube, Computación sin servidor, Función como servicio, Multitenencia, Ecosistema tecnológico

Introducción

La acelerada transformación digital de las organizaciones, las exigencias de mayores y mejores experiencias que demandan los clientes y un aumento exponencial de ataques cibernéticos, establece el contexto actual donde las organizaciones deben adelantar sus iniciativas digitales, comoquiera que, es allí donde se concretan las nuevas apuestas y promesas de valor que permiten crear escenarios inéditos para desarrollar ideas de negocio novedosas y disruptivas que cambian la vista tradicional de lo que hacen las empresas en la actualidad (Woerner et al., 2022).

Para hacerlo, se requieren condiciones de agilidad y despliegue eficiente de soluciones, lo que implica acelerar los ciclos de desarrollo y utilizar marcos de trabajo que integren diferentes componentes de forma modular y funcional para contar con las aplicaciones esperadas, con las exigencias particulares de los clientes y sobremanera atendiendo los retos del negocio y los desafíos de cumplimiento propio de los diferentes sectores de las empresas.

En línea con lo anterior, el paradigma de “computación sin servidor” (CSS) (en inglés *serverless computing*) aparece como la apuesta clave de las organizaciones para concretar un uso más eficiente de las soluciones en la nube, con el fin

de acelerar las iniciativas digitales y crear ventajas competitivas que transformen rápidamente la manera de hacer las cosas. La CSS ofrece muchas ventajas sobre la arquitectura tradicional basada en la nube: menores costos operativos, procesos simplificados, mayor flexibilidad y escalabilidad, y despliegue más rápido (Chichioco, 2019).

Si bien estas ventajas se concretan en las iniciativas digitales que transforman las organizaciones, igualmente generan riesgos cibernéticos relevantes que deben ser analizados y atendidos, para efectos de decidir el nivel de apetito de este riesgo y las capacidades que las organizaciones deben desarrollar para ajustarse a esta decisión y así, apalancar no sólo los despliegues de las aplicaciones de forma ágil y eficiente, sino igualmente confiables y menos inseguras con el fin de generar confianza digital en sus clientes.

En este sentido, este artículo hace una breve revisión de los fundamentos de la CSS, así como los retos de seguridad y control propio de este nuevo paradigma, que no sólo actualiza la lectura de las prácticas de seguridad en la computación en la nube, sino que explora los temas particulares de aseguramiento y endurecimiento de aplicaciones en la modalidad de “*Function as Service*” (Función como servicio).

“Computación sin servidor”. Evolución de la computación en la nube

La CSS es parte de la evolución de la computación en la nube. Es un ejercicio por hacer aún más transparente tanto para los clientes como para los desarrolladores aquello que se usa y cómo se usa sin importar dónde se encuentre. En este contexto, la computación en la nube marca una transformación de la computación corporativa creando un mundo basado en pago por consumo, ubicación compartida o exclusiva y creación de múltiples máquinas virtuales según necesidad, que cambia la lectura de los grandes centros de datos empresariales, los retos de las actualizaciones de software base y los desafíos de la obsolescencia tecnológica ahora todo en manos de un tercero de confianza.

El primer cambio hacia la computación en la nube se da en función de la infraestructura tecnológica, lo que se ha denominado “*Infrastructure as a Service*” (IaaS) (Infraestructura como servicio). Una estrategia que implica la simplificación de la administración haciendo más fácil la configuración y gestión de la infraestructura (*hardware*), particularmente a través de servidores virtuales y redes de centros de datos masivos de múltiple tenencia. En este escenario, el proveedor se hace responsable y custodio de la infraestructura de almacenamiento, procesamiento y redes con el fin de asegurar un entorno de computa-

ción confiable para las empresas y sus necesidades (Schleier-Smith et al., 2021).

La segunda transición se refleja en el ocultamiento de los servidores y creación de entornos de programación para desarrolladores que la literatura denomina “*Platform as a Service*” (PaaS) y “*Software as a Service*” (SaaS). Si en la primera etapa fue el hardware, en esta segunda son las máquinas y el software como marcos de trabajo simplificados que permite a los desarrolladores avanzar rápidamente en los despliegues de aplicaciones y productos especializados, para dar cumplimiento con los retos que tienen las organizaciones mejorando del “time to market” (tiempo de puesta en operación). Esto es, los desarrolladores no tienen por qué preocuparse por actualizar el sistema operativo, las herramientas de desarrollo y ni mantener el hardware (Schleier-Smith et al., 20-21).

En un tercer momento de la evolución ahora se ocultan los servidores y la programación de las aplicaciones y su operación lo que se ha denominado “*Serverless computing*” (Computación sin servidor). Actualmente, la CSS se presenta en dos versiones diferentes, conocidas como backend como servicio (BaaS) y función como servicio (FaaS). La idea central de BaaS es proporcionar a los desarrolladores de software un conjunto de servicios y herramientas (por ejemplo,

bases de datos, API (*Application Program Interface*), almacenamiento de archivos o notificaciones push) para facilitar y acelerar el desarrollo de aplicaciones móviles y web. En cuanto a FaaS, se centra en permitir a los desarrolladores de software desplegar y ejecutar sus propias funciones en la nube (sin perjuicio de que las funciones también pueden utilizar servicios adicionales como los ofrecidos en BaaS) (Marin et al., 2022).

Una plataforma “sin servidores” se compone al menos de cinco componentes: (Marin et al., 2022)

- Funciones - Las funciones suelen ejecutarse dentro de un entorno de ejecución aislado y recién generado (por ejemplo, un contenedor) dentro de un nodo operacional.
- Pasarelas API – Puerta de enlace para que las aplicaciones accedan a los datos, lógica empresarial o funcionalidad requerida por los clientes.
- Servicios en la nube (compartidos) – Integración de diferentes gamas de servicios como recolectar varios tipos de datos para reaccionar rápidamente a eventos en las plataformas, gestión del ciclo de vida de aplicaciones, habilitar capacidades DevOps o almacenamiento de corto o largo plazo.
- Herramientas de seguridad - Conjunto de herramientas y servicios para facilitar la gestión de la seguridad de los flujos de tra-

bajo. Algunas de ellas IAM (Identity and Access Management), que permite configurar controles de acceso detallados para autenticar y restringir los recursos a los que tienen acceso las funciones y VPC (Virtual Private Cloud), que permite crear redes privadas y aisladas para comunicaciones seguras entre aplicaciones que pertenecen a la misma organización.

- Plano de control - Componente de monitorización que se utiliza para comprobar periódicamente el estado de los nodos del sistema de trabajo, el software que ejecutan y los entornos de ejecución que se implementan en ellos.

“Computación sin servidor”. Adversarios y amenazas de seguridad

Entender los retos de seguridad y control de la CSS implica reconocer dos tipos de adversario: los externos y los internos, los cuales tienen perspectivas distintas que es importante reconocer y analizar para concretar un mejor aseguramiento de este tipo de computación.

Los *adversarios externos* suelen llevar a cabo sus ataques desde fuera de la nube aprovechando los campos de entrada controlados por el usuario en cualquiera de las APIs existentes que se ofrecen para gestionar eventos. Estos ataques pueden permitir a los atacantes ejecutar comandos arbitrarios dentro de la función con el fin de recuperar

datos confidenciales (por ejemplo, tokens de sesión almacenados en tablas de entorno) o manipular la ejecución de cualquier función (o servicio en la nube que reciba datos de entrada maliciosamente elaborados y no aplique técnicas adecuadas de aseguramiento de datos de entrada) (Marin et al., 2022).

De otra parte, los *adversarios internos* pueden tomar el control total de una (o más) funciones y realizar ataques desde el interior de la nube. En el caso de las nubes públicas, es relativamente sencillo para estos adversarios desplegar funciones maliciosas para intentar realizar ataques desde dentro. Estos adversarios pueden intentar: (Marin et al., 2022).

- crear canales encubiertos,
- realizar ataques de escalada de privilegios (por ejemplo, para comprometer otras funciones co-residentes o nodos trabajadores),
- recuperar o manipular datos sensibles (por ejemplo, datos en servicios de almacenamiento),
- recopilar conocimiento sobre entornos de ejecución e infraestructura,
- llevar a cabo varios tipos de ataques de denegación de servicio.

Como modelo de servicio en la nube multiusuario, la CSS es susceptible de amenazas a la seguridad, que considerando los dos tipos de adversarios previamente comentados, pueden dividirse en cinco (5)

categorías en función del lugar desde el que se lanzan: (Li et al., 2023)

1. Ataques externos a las aplicaciones por parte de usuarios malintencionados, como ataques de cross-site scripting y ataques de inyección.
2. Ataques internos a las aplicaciones por parte de intrusos malintencionados, como el acceso interno ilegal y los ataques de sniffing. Los adversarios de la red interna pueden incluso percibir información sensible a partir del patrón de comunicación y el nivel de actividad de las funciones.
3. Las tres categorías restantes son ataques horizontales *entre inquilinos*:
 - i. ataques de canal lateral,
 - ii. ataques verticales a infraestructuras sin servidor desde inquilinos maliciosos, y
 - iii. ataques verticales a aplicaciones desde plataformas maliciosas.

“Computación sin servidor”. **Retos de seguridad y control**

De acuerdo con un reciente estudio realizado por Radware (2022), se tiene que el 95% de las organizaciones utilizan al menos dos tipos de infraestructura, y casi la mitad de las organizaciones despliegan aplicaciones en cuatro o más plataformas diferentes. Como resultado, el despliegue de aplicaciones en entornos multi-nube y de nube híbrida se ha convertido en la nueva normalidad.

En este contexto, los retos de seguridad y control se vuelven relevantes no sólo para las aplicaciones que actualmente se ejecutan en estos entornos, sino para la dinámica de la organización y sus desafíos de negocio, que demandan mayor agilidad y mejores experiencias para los clientes (Sharaf, 2020), y así concretar la confianza digital. En este sentido, se plantean al menos cinco desafíos para la seguridad en la CSS como son: (Li et al., 2023)

- aislamiento de recursos – La CSS se basa en la liberación y reutilización dinámicas de software y hardware para mejorar la utilización de recursos y reducir costos. Por lo tanto, existe un fuerte requisito de aislamiento de recursos, especialmente entre múltiples inquilinos.
- monitoreo de la seguridad - La naturaleza efímera de las funciones reduce la superficie de ataque, también estrecha significativamente la ventana para que los desarrolladores descubran y diagnostiquen los problemas.
- controles de seguridad - Las fronteras fragmentadas de las aplicaciones sin servidor aumentan la dificultad del control de la seguridad. La naturaleza distribuida de las funciones agrava aún más este reto.
- protección de datos - Ya sea durante la ejecución de código en un entorno no controlado o durante el procesamiento de datos

sensibles a través de los servicios de la plataforma, pueden surgir problemas de privacidad y cumplimiento.

- investigaciones forenses - Uno de los principales retos de la investigación forense en la nube es el hecho de que los datos se almacenan a menudo en servidores situados en múltiples ubicaciones geográficas, con múltiples proveedores y modelos compartidos, lo que puede dificultar la identificación y recopilación de los datos pertinentes.

En consecuencia, cuatro elementos claves son relevantes para efectos de avanzar frente a los riesgos enumerados en secciones anteriores: (Sharaf, 2020)

- Permisos de gestión - reducir la probabilidad de que un funcionamiento independiente contenga una amenaza de seguridad oculta, por ejemplo la ejecución de un malware.
- Perímetro de seguridad - reducir el riesgo de múltiples puntos de vulnerabilidad introducidos por la aplicación sin servidor, asociados con cada función.
- Dependencia de terceros – reducir la dependencia de paquetes de terceros, reduce la complejidad de implementar la seguridad perimetral a cada función.
- Cifrado doble y fragmentación de datos - En esta técnica, reduce el acceso no autorizado y la fuga de información combinando dos técnicas de cifrado diferentes para cifrar los datos y la

fragmentación de datos para dividirlos en varias partes y distribuirlos por redes/servidores (Kumari et al, 2022).

La seguridad en la CSS exige al igual que en la computación en la nube que el aseguramiento propio de los terceros sea parte fundamental del servicio que se presta, lo que implica mantener en foco en vulnerabilidades como: (Parast et al., 2022)

- Ataques DDoS – Impedir acceso legítimo a los recursos deseados.
- Ataques a máquinas virtuales – Control de máquinas virtuales aprovechando vulnerabilidades del hipervisor.
- Secuestro de sesión – Acceso a credenciales de los usuarios y robo de identidad.
- Multitenencia – Coexistencia de múltiples usuarios en una instancia de recursos físicos de máquinas virtuales.
- Amenaza interna del proveedor – Empleado del proveedor accede a datos sensibles y utiliza la información de forma no autorizada.
- Fuga de información – Configuraciones y accesos inadecuados aumentan la filtración de datos en entorno de nube.

Reflexiones finales

Si bien la computación en la nube se ha venido convirtiendo en una estrategia natural de las organizaciones para hacer frente a las demanda de flexibilidad, elasticidad,

agilidad y despliegue efectivo y eficiente de nuevas aplicaciones y estrategias digitales de las empresas, la computación sin servidor, como parte de la evolución de la nube, establece un nuevo paradigma que requiere ser revisado y analizado en profundidad para capitalizar las oportunidades que ofrece y atender los retos que esta forma de computación revela tanto a programadores como a los proveedores (O'Meara & Lennon, 2020).

La computación sin servidor abre un espacio de construcción abierto y funcional que requiere un conocimiento en detalle de las funciones y su contexto en el cual se despliegan, por tanto la configuración de los ecosistemas tecnológicos donde residen debe responder no sólo a condiciones óptimas y específicas para el procesamiento, almacenamiento y enrutamiento de la ejecución de las aplicaciones, sino igualmente a consideraciones de seguridad y control que generen confianza digital en cada una de las relaciones e interacciones de sus componentes (Kumari et al., 2022).

Así las cosas, la CSS deberá estar alineada en dos dimensiones: seguridad *de la nube* y seguridad *en la nube*. La *seguridad de la nube* relacionada con la responsabilidad de los proveedores de la nube y las medidas establecidas para mantener la infraestructura subyacente y los servicios en la nube resistentes a las acciones no autorizadas de los adversarios. Y por otro lado, la


seguridad en la nube asociada con la responsabilidad de los desarrolladores de software. Esto es, los mecanismos de seguridad empleados para: prevenir vulnerabilidades en las funciones, proteger los datos de la aplicación (almacenados en servicios en la nube) y, asegurar la totalidad de los flujos de trabajo (por ejemplo, garantizando que todas las funciones se ejecuten con los privilegios mínimos necesarios) (Marin et al., 2022).

Finalmente, la CSS busca ampliar la capacidad de la computación en la nube para permitir a las empresas centrarse únicamente en el código y pagar únicamente por el uso de la potencia de computación, lo que necesariamente aumenta los riesgos inherentes a su configuración debido a la complejidad que surge en el despliegue de servicios en la nube en función de la demanda (Sharaf, 2020). La CSS requiere una asignación dinámica de recursos tecnológicos lo que implica un marco de seguridad y control igualmente dinámico y flexible, un reto que aún está por resolver y que demanda un enfoque diferencial que repiense la postura de seguridad de las arquitecturas sin servidores y en ecosistemas multinube.

Referencias

- Chichioco, A. (2019). Going Serverless? Common Serverless Security Issues and Best Practices. Hakin9. <https://hakin9.org/going-serverless-common-serverless-security-issues-and-best-practices/>
- Kumari, S., Solanki, K., Dalal, S. & Dhan-khar, A. (2022). Analysis Of Cloud Computing Security Threats and Countermeasures. 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India. pp. 1-6, doi: 10.1109/ICRITO56286.2022.9964632.
- Li, X., Leng, X. & Chen, Y. (2023). Securing Serverless Computing: Challenges, Solutions, and Opportunities. IEEE Network, 37(2). 166-173. doi: 10.1109/MNET.005.2100335.
- Marin, E., Perino, D. & Di Pietro, R. (2022). Serverless computing: a security perspective. Journal of Cloud Computing. 11(69). <https://doi.org/10.1186/s13677-022-00347-w>
- O'Meara, W. & Lennon, R. G. (2020). Serverless Computing Security: Protecting Application Logic. 2020 31st Irish Signals and Systems Conference (ISSC), Letterkenny, Ireland. 1-5. doi: 10.1109/ISSC49989.2020.9180214.
- Parast, F. K., Sindhav, C., Nikam, S., Yek-ta, H. I., Kent, K. B. & Hakak, S. (2022). Cloud computing security: A survey of service-based models. Computers & Security. 114. <https://doi.org/10.1016/j.cose.2021.102580>
- Radware (2022). Application Security In A Multi-Cloud World. <https://www.radware.com/multi-cloud-report-2022/>
- Schleier-Smith et al. (2021). What serverless computing is and should become: the next phase of cloud computing. Communications of the ACM. 64(5). Doi: 10.1145/3406011
- Sharaf, S. (2020). Security Issues in Serverless Computing Architecture. Inter-

national Journal of Emerging Trends in Engineering Research. 8(2).
<https://doi.org/10.30534/ijeter/2020/43822020>

Woerner, S., Weill, P. & Sebastian, I. (20-22). Future ready. The four pathways to capturing digital value. Boston, MA. USA: Harvard Business Review Press 

Jeimy J. Cano M., Ph.D, CFE, CICA. Ingeniero y Magíster en Ingeniería de Sistemas y Computación, Universidad de los Andes. Especialista en Derecho Disciplinario, Universidad Externado de Colombia; Ph.D en Business Administration, Newport University, CA. USA. y Ph.D en Educación, Universidad Santo Tomás. Profesional certificado como Certified Fraud Examiner (CFE), por la Association of Certified Fraud Examiners y Certified Internal Control Auditor (CICA) por The Institute of Internal Controls. Profesor Distinguido de la Facultad de Derecho, Universidad de los Andes. Es director de la Revista SISTEMAS de la Asociación Colombiana de Ingenieros de Sistemas –ACIS–.

Paradigma Serverless: función como servicio

DOI: 10.29236/sistemas.n168a7

Resumen

Las funciones como servicio (FaaS) nacen como un modelo de computación en la nube en el cual los desarrolladores de software solo se preocupan por el diseño y ejecución de pequeños fragmentos de código (es decir, las funciones) delegando a un tercero los aspectos concernientes al aprovisionamiento, operación y mantenimiento de la infraestructura de hardware y software necesarias para la ejecución de la aplicación. En este artículo se presenta una introducción al modelo FaaS; se explican sus ventajas y desventajas y se esbozan los desafíos a futuro.

Palabras claves

Cloud Computing, FaaS, Serverless

Introducción

Para que una aplicación (por ejemplo, una aplicación web) pueda funcionar se requiere el aprovisionamiento, configuración y administración de varios componentes (ver Figura 1), entre los que podríamos destacar el hardware, la capa de virtualización, el sistema operativo, el sistema de *runtime* (e.g., Java, Node o .NET Core) y la aplicación propiamente dicha. Hasta hace algunos años la gestión de toda esa infraestructura era responsabilidad de la empresa encargada de la aplicación. Este modelo, conocido co-

mo *on-premise*, es por tanto, costoso y complejo de administrar.

En la década del 2000, Amazon presentó Amazon Web Service (AWS), un nuevo modelo de servicios de computación denominado infraestructura como servicio (IaaS) (Mathew y Varia, 2014). En este modelo, AWS se encarga de proveer al usuario las capas de hardware y virtualización. Esto hace que los costos de operación se reduzcan ya que la empresa dueña de la aplicación no se tendrá que ocupar por el mantenimiento de

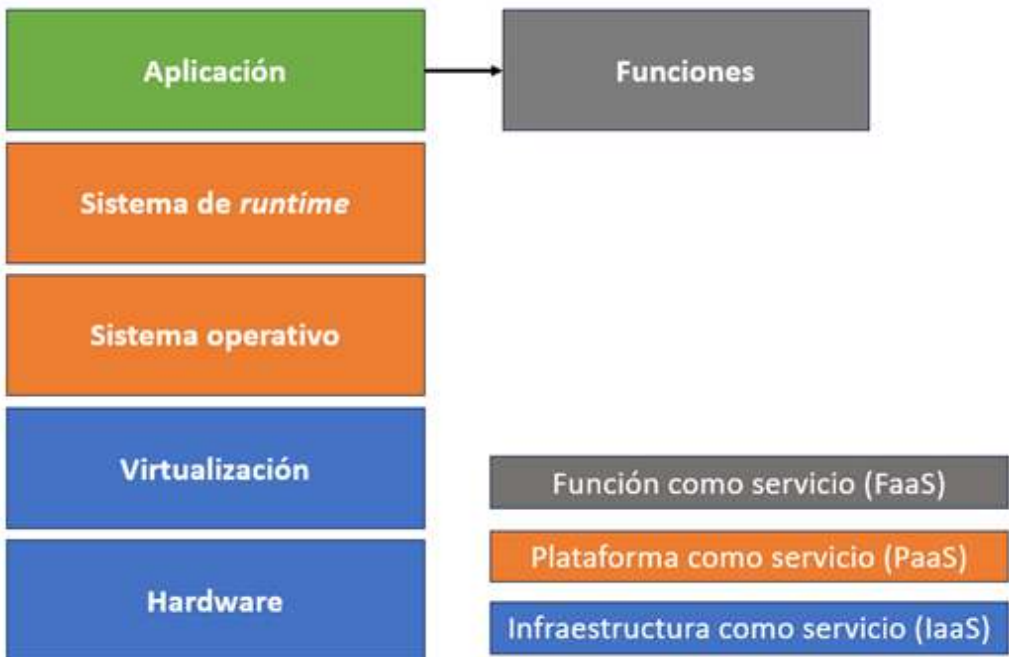


Figura 1. Componentes requeridos para la ejecución de una aplicación web

esas capas; además el aprovisionamiento del hardware que antes podría tomar meses mientras se hacían los procesos de compra, instalación y configuración, ahora solo toma unas cuantas horas o incluso, minutos.

El siguiente modelo que emerge es el denominado plataforma como servicio (PaaS) (Keller y Rexford, 2010). En este modelo se incluyen, además del hardware y la virtualización, las capas correspondientes al sistema operativo y de *runtime*. De este modo la preocupación de la empresa radica ahora solo en la administración de la aplicación.

En el 2014, hook.io¹ presentó un nuevo modelo de computación conocido como función como servicio (FaaS). Este modelo pretende abstraer la complejidad de una aplicación para que la empresa pueda

enfocarse únicamente en sus componentes principales que en este caso son las funciones.

Cómo funciona

Para entender cómo funciona este modelo pensemos, por ejemplo, en una aplicación que se encarga de registrar la actividad física de un deportista. Esta aplicación dependerá de varias funciones. Una de ellas podría ser la que consulta el plan del usuario (i.e., `getUserPlan`) para saber el tipo de plan que tiene contratado y si está vigente o no. Esta función se conectará con la base de datos de planes. Otra función, por su parte, permitirá consultar el detalle de la actividad de un usuario para un día en específico (i.e., `getActivity`). Esta función se conecta con la base de datos de registro de actividad.

En una vista simplificada de una posible arquitectura (ver Figura 2), la aplicación se conecta por ejemplo a un API Gateway que identifica

¹ <https://hook.io-mps.mto.zing.vn/>

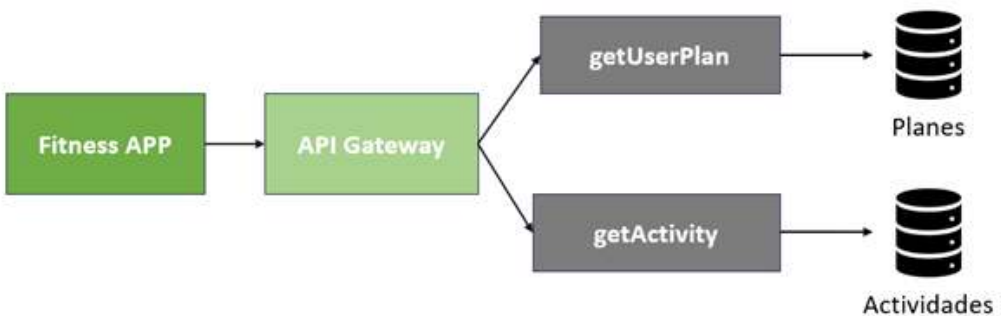


Figura 2. Arquitectura de una aplicación construida bajo el modelo FaaS

el tipo de petición y la redirige a la función como servicio requerida. De este modo, una petición del tipo GET /api/userplans será redirigida a la función getUserPlan, mientras que la petición GET /api/activities será redirigida a la función getActivity.

Lo anterior denota que una función como servicio es una pequeña porción de código que se ejecuta en la nube en contenedores sin estado (*stateless containers*). En otras palabras, estas funciones ejecutan la tarea para la cual están destinadas, no almacenan ningún estado y se apagan luego de su ejecución.

Las funciones como servicio se invocan y comienzan a ejecutarse mediante activadores, también conocidos como *triggers*. Algunos activadores pueden ser intervalos de tiempo predefinidos o una petición HTTP. De igual forma, las funciones tienen la capacidad de interactuar con una amplia gama de servicios, por ejemplo, de almacenamiento, autenticación, seguridad, entre otros. Algunas plataformas permiten configurar esos activadores de forma declarativa sin escribir ningún código.

En resumen, una función se ejecuta cada vez que es invocada por un activador. La función se ejecuta durante un intervalo de tiempo particular y entra en un estado inactivo. Luego, se activa y se ejecuta cada vez que un disparador la invoca nuevamente.

Ventajas y desventajas de las FaaS

En una función como servicio podemos resaltar cuatro ventajas. La primera es que solo se paga por lo que usa; de este modo se evita el pago por los denominados tiempos muertos o *idle time*, que es el tiempo en el que la función no se está usando. Por su naturaleza el tiempo de ejecución de una función se puede medir en milisegundos, y ese será el periodo por el cual el proveedor del servicio cobra. La segunda es que los desarrolladores solo deben enfocarse en el fragmento de código de interés, dejando de lado temas relacionados con la infraestructura subyacente. La tercera ventaja es que el escalado se hace de forma automática: si la demanda de la función es mayor (o menor), la infraestructura que da soporte se ajusta por sí misma dependiendo de las reglas definidas. La cuarta radica en que son servicios con alta disponibilidad. Esto implica que una función como servicio se puede desplegar en diferentes zonas geográficas.

A pesar de las ventajas mencionadas, el modelo de función como servicio presenta una desventaja fundamental. Recordemos que las funciones se ejecutan cuando se activan y pasan a un estado inactivo cuando no realizan ningún trabajo. Cuando una función está inactiva, pasará algún tiempo antes de que entre en acción cada vez que se active. Esto se debe a que la infraestructura subyacente tardará

algún tiempo en alistarse (también denominado como el proceso de calentamiento o *warm up*) y comenzar a ejecutar el código. Este fenómeno se conoce como el problema de inicio en frío (o *cold start*) que debe tenerse en cuenta al diseñar soluciones basadas en FaaS (Gias y Casale, 2020).

El futuro

Una vez se ha diseñado una solución basada en funciones como servicios viene el proceso de despliegue en la plataforma seleccionada, entre las que están AWS Lambda, Google Cloud Functions, Microsoft Azure Functions, IBM/ Apache's OpenWhisk u Oracle Cloud Fn. No obstante, sin embargo, el despliegue de la solución para cada plataforma puede distar bastante de ser trivial.

En este escenario lo ideal es aplicar un principio ampliamente usado en la ingeniería de software que consiste en la incorporación de una nueva capa de abstracción que opere sobre las plataformas existentes.

Conclusiones / Reflexiones finales

Un modelo de computación basado en FaaS supone unas claras ventajas por encima de los modelos *on-premise*. De este modo, la empresa se encargará de solo de administrar los pequeños fragmentos de código que constituyen una aplicación. Esto permite ahorros significativos y reduce la complejidad de

la administración de la infraestructura tecnológica.

A pesar de las ventajas, también existen condicionantes que se deben tener en cuenta. Por eso antes de implementar una solución basada en una función como servicio, conviene una reflexión sobre los siguientes aspectos:

- Se debe decidir que las funciones como servicio es la mejor aproximación para el escenario en el cual las quiere implementar.
- Hay un mercado amplio de proveedores. Por eso se hace clave elegir proveedores existentes el mejor plan de hosting y revisar que la facturación se ajuste al presupuesto disponible.
- Una solución FaaS implica que existan retrasos en el inicio de la ejecución de una función (*cold start*); por esto se debe mitigar su impacto en la solución.
- Revise aquellas funciones que impliquen largos periodos de duración en su ejecución; esto tendrá una correlación directa con el valor de la facturación.
- Facilite la integración y comunicación entre otros servicios (particularmente con los que ofrece el mismo proveedor de las soluciones de FaaS).
- Identifique y gestione los posibles cuellos de botella.
- Asegúrese de que su solución sea tolerante a fallos pero también facilite el monitoreo eficiente y la depuración de las fallas.

- Incorpore prácticas de DevOps y aporte un enfoque de infraestructura como código (IaC).

Referencias

- Gias, A. U., & Casale, G. (2020). Cocoa: Cold start aware capacity planning for function-as-a-service platforms. En *2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)* 1-8. IEEE. Doi: 10.1109/MASCOTS50786.2020.9285966
- Keller, E., & Rexford, J. (2010). The “Platform as a Service” Model for Networking. *INM/WREN*. 10, 95-108. https://www.researchgate.net/publication/228824473_The_Platform_as_a_Service_Model_for_Networking
- Mathew, S., & Varia, J. (2014). Overview of amazon web services. *Amazon Whitepapers*, 105, 1-22. https://media.amazonwebservices.com/AWS_Overview.pdf

José Bocanegra. Ingeniero de sistemas de la Universidad Distrital Francisco José de Caldas. DEA en Tecnología e Ingeniería de Software de la Universidad de Sevilla; doctor en Tecnología de la Universitat de Girona y doctor en Ingeniería de la Pontificia Universidad Javeriana. Actualmente se desempeña como profesor asistente del Departamento de Ingeniería de Sistemas y Computación de la Universidad de los Andes. Sus áreas de interés se centran en la Ingeniería de Software Dirigida por Modelos y los Lenguajes Específicos de Dominio.

AFILIATE BENEFICIOS

- ★ Inclusión en el gremio de Ingenieros de Sistemas más importante del país.
- ★ Asesoría en trámite para la tarjeta profesional.
- ★ Certificado de Miembro Profesional de la Asociación.
- ★ Podrá escoger la participación en una de las Jornadas académicas organizada por ACIS sin costo adicional.
- ★ Actualización en formación profesional y académica de manera constante.
- ★ Descuentos especiales en cursos y eventos exclusivos en el área de las TIC.
- ★ Pertener a los grupos de interés especializados en el sector (GI).
- ★ Candidato a Director o CoDirector de Grupo de Interés (GI).
- ★ Revista Sistemas (Publicación virtual Trimestral).
- ★ Candidato a Miembro de Consejo Editorial de la Revista Sistemas.
- ★ Asesoría en la edición de hojas de vida.
- ★ Referencia profesional para empleos.
- ★ Candidato a participación Profesional en proyectos de ACIS.
- ★ Candidato a participar en eventos nacionales e internacionales como delegado de ACIS.
- ★ Candidato a participar en Consultorías solicitadas a ACIS por entidades públicas o privadas.
- ★ Referencia Profesional a Cargos Público Ejecutivo.
- ★ Referencia Profesional para Miembro de la Junta Directiva ACIS o Externa.
- ★ Referencia Profesional para Perito de Procesos de Arbitraje.
- ★ Acceso diferido a la base de Webinars de ACIS. Consulte la Programación de Conferencias o YouTube
- ★ Podrá tener su correo con @acis.org.co
- ★ Acceso a Oportunidad laboral en nuestros portales Oferta de Empleo y Perfil de Ingenieros.
- ★ Suscripción Anual.
- ★ Descuento por Antigüedad.
- ★ Descuento en Convenios.



Más Información en:
www.acis.org.co
3015530540 - 3043463413

Para solicitar sus beneficios o afiliarse debe
enviar un correo electrónico a
suscripciones@acis.org.co

**CONECTA CON
NOSOTROS**

@Comunidadacis



ASOCIACIÓN COLOMBIANA DE INFORMÁTICA, SISTEMAS Y TECNOLOGÍAS AFINES

WWW.ACIS.ORG.CO