

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.