



AWS Fundamentos multirregionales

# AWS Guía prescriptiva



# AWS Guía prescriptiva: AWS Fundamentos multirregionales

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

---

# Table of Contents

|   |    |
|---|----|
| Introducción .....  | 1  |
| ¿Dispone de Well-Architected? .....   | 1  |
| Introducción .....  | 1  |
| Diseñar y operar para la resiliencia en una sola región .....                         | 3  |
| Fundamento multirregional 1: comprender los requisitos .....                          | 4  |
| Guía clave .....  | 6  |
| Fundamento multirregional 2: entender los datos .....                                 | 7  |
| 2.a: Comprender los requisitos de coherencia de los datos .....                       | 7  |
| 2.b: Comprender los patrones de acceso a los datos .....                              | 9  |
| Guía clave .....  | 10 |
| Fundamento multirregional 3: comprender las dependencias de su carga de trabajo ..... | 11 |
| 3.a: Servicios de AWS .....   | 11 |
| 3.b: Dependencias internas y de terceros .....  | 11 |
| 3.c: Mecanismo de conmutación por error .....   | 12 |
| 3.d: Dependencias de configuración .....  | 13 |
| Guía clave .....  | 13 |
| Fundamento multirregional 4: Preparación operativa .....                              | 14 |
| 4.a: gestión Cuenta de AWS .....  | 14 |
| 4.b: Prácticas de despliegue .....  | 14 |
| 4.c: Observabilidad .....   | 15 |
| 4.d: Procesos y procedimientos .....  | 16 |
| 4.e: Pruebas .....  | 16 |
| 4.f: Coste y complejidad .....  | 17 |
| 4.g: Estrategia organizativa de conmutación por error multirregional .....            | 18 |
| Guía clave .....  | 19 |
| Conclusión y recursos .....   | 20 |
| Historial de documentos .....   | 21 |
| Glosario .....  | 22 |
| # .....   | 22 |
| A .....   | 23 |
| B .....   | 26 |
| C .....   | 28 |
| D .....   | 31 |
| E .....   | 36 |

---

|         |      |
|---------|------|
| F ..... | 38   |
| G ..... | 40   |
| H ..... | 41   |
| I ..... | 42   |
| L ..... | 45   |
| M ..... | 46   |
| O ..... | 50   |
| P ..... | 53   |
| Q ..... | 56   |
| R ..... | 56   |
| S ..... | 59   |
| T ..... | 63   |
| U ..... | 65   |
| V ..... | 66   |
| W ..... | 66   |
| Z ..... | 67   |
| .....   | Ixix |

# AWS Fundamentos multirregionales

John Formento, Amazon Web Services (AWS)

Septiembre de 2025 ([historial del documento](#))

Esta guía avanzada de 300 niveles está destinada a arquitectos de la nube y líderes sénior que crean cargas de trabajo a partir de una arquitectura multirregional AWS y estén interesados en utilizarla para mejorar la resiliencia de sus cargas de trabajo. En esta guía se parte de un conocimiento básico sobre la infraestructura y los servicios. AWS Describe los casos de uso comunes en varias regiones, comparte los conceptos y las implicaciones fundamentales de varias regiones en torno al diseño, el desarrollo y la implementación, y proporciona orientación prescriptiva para ayudarlo a determinar mejor si una arquitectura multirregional es adecuada para sus cargas de trabajo.

En esta guía:

- [Diseñamos y operamos para lograr la resiliencia en una sola región](#)
- [Fundamento multirregional 1: comprender los requisitos](#)
- [Fundamento multirregional 2: Entender los datos](#)
- [Fundamento multirregional 3: comprender las dependencias de su carga de trabajo](#)
- [Fundamento 4 de la multirregión: Preparación operativa](#)
- [Conclusión y recursos](#)
- [Historial de revisión](#)

## ¿Dispone de Well-Architected?

El [AWS Well-Architected](#) Framework le ayuda a comprender las ventajas y desventajas de las decisiones que toma al crear sistemas en la nube. Los seis pilares del Marco proporcionan las mejores prácticas arquitectónicas para diseñar y operar sistemas confiables, seguros, eficientes, rentables y sostenibles. Puede utilizar el [AWS Well-Architected Tool](#), que está disponible de forma gratuita en el [Consola de administración de AWS](#), para comparar sus cargas de trabajo con estas mejores prácticas respondiendo a una serie de preguntas para cada pilar.

Para obtener orientación adicional de expertos y mejores prácticas para su arquitectura de nube, incluidas las implementaciones de arquitecturas de referencia, los diagramas y las guías técnicas, consulte el Centro de [AWS Arquitectura](#).

## Introducción

Cada una [Región de AWS](#) consta de varias zonas de disponibilidad independientes y físicamente separadas dentro de un área geográfica. Se mantiene una separación lógica estricta entre los servicios de software de cada región. Este diseño específico garantiza que una falla en la infraestructura o el servicio en una región no provoque una falla correlacionada en otra región.

La mayoría de AWS los usuarios pueden alcanzar sus objetivos de resiliencia para una carga de trabajo en una sola región mediante el uso de varias zonas de disponibilidad o regionales. Servicios de AWS Sin embargo, un subconjunto de usuarios opta por arquitecturas multirregionales por tres motivos:

- Tienen requisitos de alta disponibilidad y continuidad de las operaciones para sus cargas de trabajo de nivel más alto y desean establecer un tiempo de recuperación limitado en caso de deficiencias que afecten a los recursos de una sola región.
- Deben cumplir con los requisitos de [soberanía de los datos](#) (como el cumplimiento de las leyes y reglamentos locales) que exigen que las cargas de trabajo operen dentro de una jurisdicción determinada.
- Necesitan mejorar el rendimiento y la experiencia del cliente en relación con la carga de trabajo mediante la ejecución de las cargas de trabajo en las ubicaciones más cercanas a sus usuarios finales.

Esta guía se centra en los requisitos de alta disponibilidad y continuidad de las operaciones, y le ayuda a analizar las consideraciones a la hora de adoptar una arquitectura multirregional para una carga de trabajo. Describe los conceptos fundamentales que se aplican al diseño, el desarrollo y el despliegue de una carga de trabajo multirregional y proporciona un marco prescriptivo que le ayudará a determinar si una arquitectura multirregional es la opción correcta para una carga de trabajo concreta. Debe asegurarse de que una arquitectura multirregional sea la opción correcta para su carga de trabajo, ya que estas arquitecturas suponen un desafío y, si la arquitectura multirregional no se construye correctamente, es posible que la disponibilidad general de la carga de trabajo disminuya.

# Diseñar y operar para la resiliencia en una sola región

Antes de profundizar en los conceptos multirregionales, comience por confirmar que su carga de trabajo ya es lo más resiliente posible en una sola región. Para lograrlo, evalúe su carga de trabajo en relación con el [pilar de confiabilidad](#) y el [pilar de excelencia operativa](#) del AWS Well-Architected Framework y realice los cambios necesarios en función de las compensaciones y la evaluación de riesgos. Los siguientes conceptos se tratan en el AWS Well-Architected Framework:

- [Segmentación de la carga de trabajo en función de los límites del dominio](#)
- [Contratos de servicio bien definidos](#)
- [Gestión de la dependencia y acoplamiento](#)
- [Gestionar los errores, los reintentos y las estrategias de espera](#)
- [Operaciones idempotentes y transacciones con estado frente a transacciones sin estado](#)
- [Preparación operativa y gestión de cambios](#)
- [Comprensión del estado de las cargas](#)
- [Responder a los eventos](#)

Para llevar la resiliencia de una sola región más allá, revise y aplique los conceptos que se analizan en el documento [Advanced Multi-AZ Resilience Patterns: Detecting and Mitigating Gray Failures](#). Este paper proporciona las mejores prácticas para el uso de réplicas en cada zona de disponibilidad para contener los errores y amplía los conceptos de zonas de disponibilidad múltiples que se introducen en AWS Well Architected Framework. Si bien una arquitectura multirregional puede mitigar los modos de error que están vinculados a las zonas de disponibilidad, un enfoque multirregional conlleva algunas desventajas que debe tener en cuenta. Por eso, le recomendamos que comience con un enfoque multizona de disponibilidad y, a continuación, evalúe una carga de trabajo específica comparándola con los aspectos fundamentales de las arquitecturas multirregionales para determinar si un enfoque multirregional puede aumentar la resiliencia de la carga de trabajo.

# Fundamento multirregional 1: comprender los requisitos

Como se mencionó anteriormente, la alta disponibilidad y la continuidad de las operaciones son motivos habituales para optar por arquitecturas multirregionales. Las métricas de disponibilidad miden el porcentaje de tiempo que una carga de trabajo está disponible para su uso durante un período definido, mientras que las métricas de continuidad de las operaciones miden el tiempo de recuperación para eventos a gran escala y, por lo general, de mayor duración.

[Medir la disponibilidad](#) es un proceso casi continuo. Las mediciones específicas pueden variar, pero por lo general se agrupan en torno a una métrica de disponibilidad objetivo, que en la mayoría de los casos se denomina nueves (por ejemplo, una disponibilidad del 99,99 por ciento). Con los objetivos de disponibilidad, no hay una solución única para todos. Debe establecer objetivos de disponibilidad a nivel de carga de trabajo y separar los componentes no críticos de los componentes críticos, en lugar de aplicar un objetivo único a todas las cargas de trabajo.

Para la continuidad de las operaciones, se suelen utilizar las siguientes point-in-time medidas:

- **Objetivo de tiempo de recuperación (RTO):** el RTO es el retraso máximo aceptable entre la interrupción del servicio y el restablecimiento del servicio. Este valor determina un período aceptable durante el cual el servicio estará interrumpido.
- **Objetivo de punto de recuperación (RPO):** el RPO es el tiempo máximo aceptable desde el último punto de recuperación de datos. Esto determina qué se considera una pérdida de datos aceptable entre el último punto de recuperación y una interrupción del servicio.

Al igual que cuando se establecen los objetivos de disponibilidad, el RTO y el RPO también deben definirse a nivel de carga de trabajo. Una continuidad de las operaciones más agresiva o una alta disponibilidad requieren una mayor inversión. Dicho esto, no todas las aplicaciones pueden exigir o requieren el mismo nivel de resiliencia. Alinear a los propietarios empresariales y de TI para evaluar la criticidad de las aplicaciones en función del impacto empresarial y, luego, organizarlas en niveles en consecuencia puede ayudar a proporcionar un punto de partida. En las siguientes tablas se proporcionan ejemplos de organización en niveles.

En esta tabla se muestra un ejemplo de organización por niveles de resiliencia para los acuerdos de nivel de servicio (SLAs).

| Nivel de resiliencia | SLA de disponibilidad | Tiempo de inactividad aceptable por año |
|----------------------|-----------------------|---|
| Platino              | 99,99%                | 52,60 minutos                           |
| Oro                  | 99,90%                | 8,77 horas                              |
| Plata                | 99,5%                 | 1,83 días                               |

La siguiente tabla muestra un ejemplo de estratificación de resiliencia para RTO y RPO.

| Nivel de resiliencia | RTO máximo           | RPO máximo | Criterios  | Costo  |
|----------------------|----------------------|------------|--|--------|
| Platino              | 15 minutos           | 5 minutos  | Cargas de trabajo de misión crítica              | \$\$\$ |
| Oro                  | 15 minutos — 6 horas | 2 horas    | Cargas de trabajo importantes pero no esenciales | \$\$   |
| Plata                | 6 horas, unos días   | 24 horas   | Cargas de trabajo no críticas                    | \$     |

Cuando diseñe cargas de trabajo que tengan resiliencia, tenga en cuenta la relación entre la alta disponibilidad y la continuidad de las operaciones. Por ejemplo, si una carga de trabajo requiere una disponibilidad del 99,99 por ciento, no se toleran más de 53 minutos de inactividad al año. La detección de una falla puede tardar al menos 5 minutos y un operador puede tardar otros 10 minutos en intervenir, tomar decisiones sobre los pasos de recuperación y llevarlos a cabo. No es inusual tardar entre 30 y 45 minutos en recuperarse de un solo problema. En este caso, es beneficioso contar con una estrategia multirregional para proporcionar una instancia aislada que elimine el impacto correlacionado. Esto permite continuar con las operaciones al realizar la conmutación por error dentro de un tiempo limitado y, al mismo tiempo, clasificar la avería inicial de forma

independiente. Aquí es donde es necesario definir el tiempo de recuperación limitado adecuado y garantizar que haya una alineación.

Un enfoque multirregional podría ser adecuado para cargas de trabajo de misión crítica que tienen necesidades de disponibilidad extremas (por ejemplo, una disponibilidad del 99,99 por ciento o más) o para requisitos estrictos de continuidad de las operaciones que solo se pueden cumplir si se transfieren por error a otra región. Sin embargo, estos requisitos suelen aplicarse solo a un pequeño subconjunto de la cartera de cargas de trabajo de una empresa que tiene un tiempo de recuperación limitado medido en minutos u horas. A menos que una aplicación necesite un tiempo de recuperación de unos minutos o unas horas, lo mejor sería esperar a que se corrija una interrupción regional de la aplicación en la región afectada. Este enfoque suele estar alineado con las cargas de trabajo de nivel inferior.

Antes de implementar una arquitectura multirregional, los responsables de la toma de decisiones empresariales y los equipos técnicos deben tener en cuenta las implicaciones de los costos, incluidos los factores de costos operativos y de infraestructura. Una arquitectura multirregional típica puede incurrir en un costo dos veces mayor que un enfoque de una sola región. Si bien existen varios patrones multirregionales para la continuidad del negocio, como funcionar con un modo de [espera en caliente, en modo](#) de [espera en caliente](#) o con una [luz piloto](#), el patrón con el menor riesgo de cumplir los objetivos de recuperación implicará utilizar un modo de espera activo y duplicará el coste de la carga de trabajo.

## Guía clave

- Los objetivos de disponibilidad y continuidad de las operaciones, como el RTO y el RPO, deben establecerse por carga de trabajo y alinearse con las partes interesadas de la empresa y de TI.
- La mayoría de los objetivos de disponibilidad y continuidad de las operaciones se pueden cumplir en una sola región. En el caso de los objetivos que no se puedan cumplir dentro de una sola región, opte por la opción de multirregión con una visión clara de las ventajas y desventajas entre costo, complejidad y beneficios.

# Fundamento multirregional 2: entender los datos

La administración de los datos no es un problema trivial cuando se adoptan arquitecturas multirregionales. La distancia geográfica entre las regiones impone una latencia inevitable que se manifiesta como el tiempo que se tarda en replicar los datos en todas las regiones. Será necesario hacer concesiones entre la disponibilidad, la coherencia de los datos y la introducción de una mayor latencia en una carga de trabajo que utilice una arquitectura multirregional. Ya sea que utilice la replicación asíncrona o sincrónica, tendrá que modificar la aplicación para que pueda gestionar los cambios de comportamiento que impone la tecnología de replicación. Los desafíos relacionados con la consistencia y la latencia de los datos hacen que sea muy difícil convertir una aplicación existente diseñada para una sola región y convertirla en multirregión. Comprender los requisitos de coherencia de los datos y los patrones de acceso a los datos para determinadas cargas de trabajo es fundamental para sopesar las desventajas.

## 2.a: Comprender los requisitos de coherencia de los datos

El [teorema CAP](#) proporciona una referencia para razonar sobre las compensaciones entre la consistencia de los datos, la disponibilidad y las particiones de la red. Solo se pueden cumplir dos de estos requisitos al mismo tiempo para una carga de trabajo. Por definición, una arquitectura multirregional incluye particiones de red entre regiones, por lo que debe elegir entre disponibilidad y coherencia.

Si selecciona la disponibilidad de los datos en todas las regiones, no incurrirá en una latencia significativa durante las operaciones de escritura transaccional, ya que al depender de la replicación asíncrona de los datos comprometidos entre las regiones se reduce la coherencia entre las regiones hasta que se complete la replicación. Con la replicación asíncrona, cuando se produce un error en la región principal, existe una alta probabilidad de que las operaciones de escritura estén pendientes de replicación desde la región principal. Esto lleva a una situación en la que los datos más recientes no están disponibles hasta que se reanude la replicación y se necesita un proceso de reconciliación para gestionar las transacciones en curso que no se replicaron desde la región que sufrió la interrupción. En este escenario, es necesario comprender la lógica empresarial y crear un proceso específico para reproducir la transacción o comparar los almacenes de datos entre regiones.

Para las cargas de trabajo en las que se prefiere la replicación asíncrona, puede utilizar servicios como Amazon Aurora y [Amazon](#) DynamoDB para la replicación [asíncrona](#) entre regiones. Tanto las [bases de datos globales de Amazon Aurora](#) como las tablas [globales de Amazon DynamoDB](#)

tienen métricas de [CloudWatchAmazon](#) predeterminadas que ayudan a monitorizar el retraso en la replicación. Una base de datos global de Aurora consta de una región principal en la que se escriben los datos y hasta cinco regiones secundarias de solo lectura. Las tablas globales de DynamoDB constan de tablas de réplicas multiactivas en cualquier número de regiones en las que se escriben y desde las que se leen los datos.

Diseñar la carga de trabajo para aprovechar las arquitecturas basadas en eventos es una ventaja para una estrategia multirregional, ya que significa que la carga de trabajo puede incluir la replicación asíncrona de los datos y permite reconstruir el estado mediante la reproducción de los eventos. Dado que los servicios de transmisión y mensajería almacenan los datos de carga útil de los mensajes en una sola región, un proceso de conmutación por error o recuperación regional debe incluir un mecanismo para redirigir los flujos de datos de entrada del cliente. El proceso también debe conciliar las cargas útiles en vuelo o no entregadas almacenadas en la región que sufrió la interrupción.

Si elige el requisito de coherencia de la CAP y utiliza una base de datos replicada de forma sincrónica en todas las regiones para respaldar las aplicaciones que se ejecutan simultáneamente desde varias regiones, elimina el riesgo de pérdida de datos y mantiene los datos sincronizados entre las regiones. Sin embargo, esto introduce características de latencia más altas, ya que las escrituras deben asignarse a más de una región y las regiones pueden estar a cientos o miles de millas una de la otra. Debe tener en cuenta esta característica de latencia en el diseño de la aplicación. Además, la replicación sincrónica puede suponer la posibilidad de que se produzcan errores correlacionados, ya que las escrituras deberán realizarse en más de una región para que se realicen correctamente. Si hay algún impedimento en una región, tendrás que formar quórum para que las escrituras se realicen correctamente. Por lo general, esto implica configurar la base de datos en tres regiones y establecer un quórum de dos de las tres regiones. Tecnologías como [Paxos](#) pueden ayudar a replicar y almacenar datos de forma sincrónica, pero requieren una importante inversión de los desarrolladores.

Cuando las escrituras implican una replicación sincrónica en varias regiones para cumplir con estrictos requisitos de coherencia, la latencia de escritura aumenta en un orden de magnitud. Una latencia de escritura más alta no es algo que normalmente se pueda adaptar a una aplicación sin cambios significativos, como revisar el tiempo de espera y la estrategia de reintentos de la aplicación. Lo ideal es que se tenga en cuenta al diseñar la aplicación por primera vez. Para las cargas de trabajo multirregionales en las que la replicación sincrónica es una prioridad, [AWS Partner las soluciones](#) pueden ayudar.

## 2.b: Comprender los patrones de acceso a los datos

Los patrones de acceso a los datos de la carga de trabajo son intensivos en lectura o escritura. Comprender esta característica de una carga de trabajo concreta le ayudará a seleccionar una arquitectura multirregional adecuada.

Para cargas de trabajo de lectura intensiva, como el contenido estático que es completamente de solo lectura, puede lograr una arquitectura multirregional [activa-activa](#) que tenga menos complejidad de ingeniería en comparación con una carga de trabajo de escritura intensiva. Ofrecer contenido estático en la periferia mediante una red de entrega de contenido (CDN) garantiza la disponibilidad al almacenar en caché el contenido más cercano al usuario final; el uso de conjuntos de funciones como la [conmutación por error de origen en Amazon CloudFront](#) puede ayudar a lograrlo. Otra opción es implementar la informática sin estado en varias regiones y usar el DNS para dirigir a los usuarios a la región más cercana para leer el contenido. Para ello, puede utilizar [Amazon Route 53 con una política de enrutamiento de geolocalización](#).

Para las cargas de trabajo de lectura intensiva que tienen un porcentaje de tráfico de lectura mayor que de tráfico de escritura, puede utilizar una estrategia de [lectura local](#) y de escritura global. Esto significa que todas las solicitudes de escritura van a una base de datos de una región específica, los datos se replican de forma asíncrona en todas las demás regiones y las lecturas se pueden realizar en cualquier región. Este enfoque requiere una carga de trabajo para lograr una coherencia final, ya que las lecturas locales pueden quedar obsoletas como resultado del aumento de la latencia para la replicación de escrituras entre regiones.

[Las bases de datos globales de Aurora](#) pueden ayudar a aprovisionar [réplicas de lectura](#) en una región en espera que solo puede gestionar todo el tráfico de lectura de forma local y a aprovisionar un único almacén de datos principal en una región específica para gestionar el tráfico de escritura. Los datos se replican de forma asíncrona desde la base de datos principal a las bases de datos en espera (réplicas de lectura), y las bases de datos en espera se pueden convertir en principales si es necesario realizar una conmutación por error de las operaciones a la región en espera. También puede utilizar DynamoDB en este enfoque. Las tablas [globales de DynamoDB](#) pueden [aprovisionar tablas de réplica](#) en todas las regiones, cada una de las cuales puede escalarse para admitir cualquier volumen de tráfico de lectura o escritura local. Cuando una aplicación escribe datos en una réplica de tabla de una región, DynamoDB propaga automáticamente la operación de escritura en el resto de las réplicas de tablas de las otras regiones de . Con esta configuración, los datos se replican de forma asíncrona desde una región principal definida a las tablas de réplica de las regiones en espera. Las réplicas de tablas de cualquier región siempre aceptan escrituras, por lo que la promoción de una región en espera a la categoría principal se gestiona a nivel de aplicación. Una

vez más, la carga de trabajo tiene que ser coherente con el tiempo, por lo que podría ser necesario volver a escribirla si no se diseñó para ello desde el principio.

Para las cargas de trabajo con un uso intensivo de escritura, se debe seleccionar una región principal y se debe incorporar a la carga de trabajo la capacidad de realizar la conmutación por error a una región en espera. En comparación con un enfoque activo-activo, un enfoque de reserva [primaria](#) tiene ventajas adicionales. Esto se debe a que, en el caso de una arquitectura activa-activa, es necesario reescribir la carga de trabajo para gestionar el enrutamiento inteligente a Regions, establecer la afinidad de las sesiones, garantizar la idempotencia de las transacciones y gestionar los posibles conflictos.

La mayoría de las cargas de trabajo que utilizan un enfoque multirregional para mejorar la resiliencia no requieren un enfoque activo-activo. Puede utilizar una estrategia de [fragmentación](#) para aumentar la resiliencia al limitar el alcance del impacto de una discapacidad en la base de clientes. Si puede fragmentar una base de clientes de forma eficaz, puede seleccionar diferentes regiones principales para cada fragmento. Por ejemplo, puede dividir los clientes para que la mitad de los clientes estén alineados con la región uno y la otra mitad estén alineados con la región dos. Al tratar las regiones como células, puede crear un enfoque de células multirregionales, lo que reduce el alcance del impacto en su carga de trabajo. Para obtener más información, consulte la [presentación de AWS re:Invent sobre este enfoque](#).

Puede combinar el enfoque de fragmentación con un enfoque principal en espera para proporcionar capacidades de conmutación por error a los fragmentos. Deberá incorporar a la carga de trabajo un proceso de conmutación por error probado, así como un proceso de reconciliación de datos, a fin de garantizar la coherencia transaccional de los almacenes de datos tras la conmutación por error. Estos aspectos se describen con mayor detalle más adelante en esta guía.

## Guía clave

- Existe una alta probabilidad de que las escrituras pendientes de replicación no se envíen a la región en espera si se produce un error. Los datos no estarán disponibles hasta que se reanude la replicación (suponiendo que la replicación sea asíncrona).
- Como parte de la conmutación por error, se necesitará un proceso de reconciliación de datos para garantizar que los almacenes de datos que utilizan la replicación asíncrona mantengan un estado coherente desde el punto de vista de las transacciones. Esto requiere una lógica empresarial específica y no es algo que deba gestionar el propio almacén de datos.
- Cuando se requiere una coherencia sólida, las cargas de trabajo deberán modificarse para tolerar la latencia requerida de un almacén de datos que se replica de forma sincrónica.

# Fundamento multirregional 3: comprender las dependencias de su carga de trabajo

Una carga de trabajo específica puede tener varias dependencias en una región, como la Servicios de AWS utilizada, las dependencias internas, las dependencias de terceros, las dependencias de red, los certificados, las claves, los secretos y los parámetros. Para garantizar el funcionamiento de la carga de trabajo en un escenario de fallo, no debe haber dependencias entre la región principal y la región en espera; cada una debe poder funcionar de forma independiente. Para lograrlo, analice todas las dependencias de la carga de trabajo para asegurarse de que estén disponibles en cada región. Esto es necesario porque un error en la región principal no debería afectar a la región en espera. Además, debe comprender cómo funciona la carga de trabajo cuando una dependencia se encuentra en un estado degradado o no está disponible por completo, de modo que pueda diseñar soluciones que lo gestionen de forma adecuada.

## 3.a: Servicios de AWS

Al diseñar una arquitectura multirregional, es importante entender qué Servicios de AWS se utilizará, las [características multirregionales](#) de esos servicios y qué soluciones necesitará diseñar para lograr los objetivos multirregionales. Por ejemplo, Amazon Aurora y Amazon DynamoDB pueden replicar datos de forma asíncrona en una región en espera. Todos Servicio de AWS las dependencias deberán estar disponibles en todas las regiones desde las que se vaya a ejecutar la carga de trabajo. Para confirmar que los servicios que utilizas están disponibles en las regiones deseadas, consulta la [lista Servicios de AWS por región](#).

## 3.b: Dependencias internas y de terceros

Asegúrese de que las dependencias internas de cada carga de trabajo estén disponibles en las regiones desde las que operan. Por ejemplo, si la carga de trabajo está compuesta por muchos microservicios, identifique todos los microservicios que componen una capacidad empresarial y compruebe que todos esos microservicios estén implementados en cada región desde la que opera la carga de trabajo. Como alternativa, defina una estrategia para gestionar adecuadamente los microservicios que dejen de estar disponibles.

No se recomiendan las llamadas interregionales entre microservicios dentro de una carga de trabajo, y se debe mantener el aislamiento regional. Esto se debe a que la creación de dependencias entre

regiones aumenta el riesgo de que se produzcan errores correlacionados, lo que compensa los beneficios de las implementaciones regionales aisladas de la carga de trabajo. Las dependencias locales también pueden ser parte de la carga de trabajo, por lo que es importante entender cómo podrían cambiar las características de estas integraciones si cambiara la región principal. Por ejemplo, si la región en espera se encuentra más alejada del entorno local, el aumento de la latencia podría tener un impacto negativo.

Comprender las soluciones de software como servicio (SaaS), los kits de desarrollo de software (SDKs) y otras dependencias de productos de terceros, y poder analizar situaciones en las que estas dependencias se degradan o no están disponibles, proporcionará más información sobre cómo funciona y se comporta la cadena de sistemas en diferentes modos de falla. Estas dependencias pueden estar dentro del código de la aplicación, como la administración externa de secretos mediante el uso [AWS Secrets Manager](#), o pueden implicar una solución de almacén de terceros (por ejemplo HashiCorp) o sistemas de autenticación que dependan de los inicios de sesión federados. [AWS IAM Identity Center](#)

Tener redundancia en lo que respecta a las dependencias puede aumentar la resiliencia. Si una solución SaaS o una dependencia de terceros utiliza la misma carga principal Región de AWS que la carga de trabajo, trabaje con el proveedor para determinar si su postura de resiliencia se ajusta a sus requisitos para la carga de trabajo.

Además, tenga en cuenta el destino compartido entre la carga de trabajo y sus dependencias, como las aplicaciones de terceros. Si las dependencias no están disponibles en (o desde) una región secundaria tras una conmutación por error, es posible que la carga de trabajo no se recupere por completo.

### 3.c: Mecanismo de conmutación por error

El DNS se suele utilizar como mecanismo de conmutación por error para desplazar el tráfico de la región principal a una región de reserva. Revise y analice críticamente todas las dependencias que implica el mecanismo de conmutación por error. Por ejemplo, si su carga de trabajo utiliza [Amazon Route 53](#), entender que el plano de control está hospedado us-east-1 significa que está pasando a depender del plano de control de esa región específica. Esto no se recomienda como parte de un mecanismo de conmutación por error si la región principal también se us-east-1 debe a que crea un único punto de error. Si utiliza otro mecanismo de conmutación por error, debe tener un conocimiento profundo de los escenarios en los que la conmutación por error no funcionaría según lo esperado y, a continuación, planificar la contingencia o desarrollar un nuevo mecanismo si fuera necesario. El [conmutador de región Amazon Application Recovery Controller \(ARC\)](#) es un

servicio de recuperación multirregional totalmente gestionado que puede utilizar como mecanismo de conmutación por error.

Como se ha explicado en la sección anterior, todos los microservicios que forman parte de una capacidad empresarial deben estar disponibles en cada región en la que se despliegue la carga de trabajo. Como parte de la estrategia de conmutación por error, todos los microservicios que forman parte de la capacidad empresarial deben realizar la conmutación por error al mismo tiempo para evitar que se produzcan llamadas entre regiones. Como alternativa, si los microservicios se conmutan por error de forma independiente, existe la posibilidad de que se produzcan comportamientos no deseados, como la posibilidad de que los microservicios realicen llamadas entre regiones. Esto introduce latencia y puede provocar que la carga de trabajo deje de estar disponible durante los tiempos de espera de los clientes.

### 3.d: Dependencias de configuración

Los certificados, las claves, los secretos, las imágenes de Amazon Machine Images (AMIs), las imágenes de contenedores y los parámetros forman parte del análisis de dependencias necesario al diseñar una arquitectura multirregional. Siempre que sea posible, es mejor localizar estos componentes dentro de cada región para que estas dependencias no tengan un destino compartido entre las regiones. Por ejemplo, debe variar las fechas de caducidad de los certificados para evitar que un certificado que caduque (con las alarmas configuradas para «notificar con antelación») afecte a varias regiones.

Las claves y los secretos de cifrado también deben ser específicos de cada región. De esta forma, si se produce un error en la rotación de una clave o secreto, el impacto se limita a una región específica.

Por último, todos los parámetros de la carga de trabajo deben almacenarse localmente para que la carga de trabajo se recupere en la región específica.

## Guía clave

- Una arquitectura multirregional se beneficia de la separación física y lógica entre las regiones. La introducción de dependencias entre regiones en la capa de aplicación reduce este beneficio. Evite este tipo de dependencias.
- Los controles de conmutación por error deberían funcionar sin depender de la región principal.
- La conmutación por error debe coordinarse a lo largo del recorrido del usuario para eliminar la posibilidad de que aumenten la latencia y la dependencia de las llamadas entre regiones.

# Fundamento multirregional 4: Preparación operativa

Gestionar una carga de trabajo multirregional es una tarea compleja que conlleva desafíos operativos específicos de una arquitectura multirregional. Estos incluyen la Cuenta de AWS administración, la reorganización de los procesos de implementación, la creación de una estrategia de observabilidad multirregional, la creación y prueba de los procesos de recuperación y, posteriormente, la administración del costo. Una [revisión de la preparación operativa \(ORR\)](#) puede ayudar a los equipos a preparar una carga de trabajo para la producción, ya sea que se ejecute en una sola región o en varias regiones.

## 4.a: gestión Cuenta de AWS

Para implementar una carga de trabajo en todas las regiones Regiones de AWS, asegúrate de que haya paridad en todas las [Servicio de AWS cuotas](#) de una cuenta en todas las regiones. En primer lugar, identifique todos los elementos Servicios de AWS que forman parte de la arquitectura, analice el uso planificado en las regiones en espera y, a continuación, compare el uso planificado con el uso actual. En algunos casos, si la región en espera no se ha utilizado antes, puede consultar las [cuotas de servicio predeterminadas](#) para comprender el punto de partida. Luego, en todos los servicios que se utilizarán, solicite un aumento de cuota mediante la [consola Service Quotas](#) (es necesario iniciar sesión) o [APIs](#).

Configure las funciones [AWS Identity and Access Management \(IAM\)](#) en cada región para proporcionar a los operadores, las herramientas de automatización y Servicios de AWS los permisos adecuados para acceder a los recursos de la región en espera. Para lograr el aislamiento regional en las arquitecturas multirregionales, aisle las funciones por región. Asegúrese de que los permisos estén en vigor antes de poner en marcha una región en espera.

## 4.b: Prácticas de despliegue

Las capacidades multirregionales pueden complicar el despliegue de una carga de trabajo en varias regiones. Debe asegurarse de realizar la implementación en una región a la vez. Por ejemplo, si utiliza un enfoque activo-pasivo, primero debe realizar el despliegue en la región principal y, después, en la región en espera. [AWS CloudFormation](#) le ayuda a implementar la infraestructura en una o varias regiones y se puede personalizar en función de sus necesidades. [AWS CodePipeline](#) le ayuda a crear una cartera de integración/continuous entrega continua (CI/CD), que incluye [acciones interregionales que permiten la](#) implementación en regiones distintas de la región en la que se

encuentra la cartera. Esto, combinado con [estrategias de despliegue](#) sólidas, como la [azul/verde](#), permite un despliegue con un tiempo de inactividad mínimo o nulo.

Sin embargo, el despliegue de capacidades con estado puede resultar más complejo cuando el estado de la aplicación o los datos no se externalizan a un almacén persistente. En estas situaciones, adapte cuidadosamente el proceso de implementación para que se adapte a sus necesidades. Diseñe el proceso y el proceso de despliegue para que se desplieguen en una región a la vez, en lugar de desplegarlos en varias regiones simultáneamente. Esto reduce la posibilidad de que se produzcan errores correlacionados entre las regiones. Para obtener más información sobre las técnicas que Amazon utiliza para automatizar las implementaciones de software, consulta el artículo de AWS Builders' Library sobre cómo [automatizar las implementaciones seguras y sin intervención](#).

## 4.c: Observabilidad

Cuando diseñe para varias regiones, considere cómo va a monitorear el estado de todos los componentes de cada región para obtener una visión holística de la salud regional. Esto podría incluir el monitoreo de las métricas para detectar el retraso en la replicación, lo que no es una consideración para una carga de trabajo de una sola región.

Al crear una arquitectura multirregional, considere la posibilidad de observar también el rendimiento de la carga de trabajo desde las regiones en espera. Esto incluye realizar controles de salud y realizar canarios (pruebas sintéticas) desde la región de reserva para obtener una visión externa del estado de salud de la región principal. Además, puede usar [Amazon CloudWatch Internet Monitor](#) para comprender el estado de la red externa y el rendimiento de sus cargas de trabajo desde la perspectiva del usuario final. La región principal debe tener la misma observabilidad para monitorear la región en espera.

Los canarios de la región de reserva deberían supervisar las métricas de experiencia de los clientes para determinar el estado general de la carga de trabajo. Esto es necesario porque si hay un problema en la región principal, la observabilidad en la principal podría verse afectada y afectar a la capacidad de evaluar el estado de la carga de trabajo.

En ese caso, observar fuera de esa región puede proporcionar información. Estas métricas deben agruparse en paneles de control disponibles en cada región y en alarmas que se creen en cada región. Como [CloudWatch](#) se trata de un servicio regional, es obligatorio disponer de alarmas en ambas regiones. Estos datos de supervisión se utilizarán para realizar la llamada de conmutación por error de una región principal a una en espera.

## 4.d: Procesos y procedimientos

El mejor momento para responder a la pregunta «¿Cuándo debo realizar la conmutación por error?» pasará mucho tiempo antes de que lo necesites. Defina planes de recuperación que incluyan a las personas, los procesos y la tecnología mucho antes de que se produzca un problema y pruébelos periódicamente. Decida un marco de decisión de recuperación. Si hay un proceso de recuperación bien practicado y se conoce bien el tiempo de recuperación, puede optar por iniciar el proceso de recuperación mediante una conmutación por error que cumpla con el objetivo de RTO. Este momento puede ocurrir inmediatamente después de que se detecte un problema con la aplicación en la región principal, o puede ocurrir más adelante, cuando se hayan agotado las opciones de recuperación de la aplicación de la región.

La acción de conmutación por error en sí misma debería estar automatizada al cien por cien, pero la decisión de activar la conmutación por error la deben tomar personas, normalmente un número reducido de personas predeterminadas de la organización. Estas personas deberían tener en cuenta la pérdida de datos y la información sobre el evento. Además, los criterios para una conmutación por error deben estar claramente definidos y ser entendidos globalmente en la organización. Para definir y completar estos procesos, puede utilizar el [conmutador de región Amazon Application Recovery Controller \(ARC\)](#), que permite una end-to-end automatización completa y garantiza la coherencia de los procesos que se ejecutan durante las pruebas y la conmutación por error.

Cuando crea un plan de cambio de región, lo replica automáticamente en las regiones principal y en espera para garantizar que no se dependa de una sola región. Cuando esta automatización esté implementada, defina y siga un ritmo de pruebas regular. Esto garantiza que, cuando se produzca un evento real, la respuesta siga un proceso bien definido y practicado en el que la organización confíe. También es importante tener en cuenta las tolerancias establecidas para los procesos de reconciliación de datos. Confirme que el proceso propuesto cumpla con RPO/RTO los requisitos establecidos.

## 4.e: Pruebas

Tener un enfoque de recuperación no probado equivale a no tener un enfoque de recuperación. Un nivel básico de prueba sería ejecutar un procedimiento de recuperación para cambiar la región operativa de la aplicación. A veces, esto se denomina enfoque de rotación de aplicaciones. Le recomendamos que desarrolle la capacidad de cambiar de región a su postura operativa normal; sin embargo, esta prueba por sí sola no es suficiente.

Las pruebas de resiliencia también son fundamentales para validar el enfoque de recuperación de una aplicación. Esto implica detectar escenarios de error específicos, comprender cómo reaccionan la aplicación y el proceso de recuperación y, a continuación, implementar las medidas de mitigación necesarias si la prueba no se llevó a cabo según lo planeado. Probar el procedimiento de recuperación sin errores no le indicará cómo se comporta la aplicación en su conjunto cuando se producen errores. Debe desarrollar un plan para evaluar su recuperación en función de los escenarios de error esperados. [AWS Fault Injection Service](#) proporciona una lista cada vez mayor de [escenarios](#) para que pueda empezar.

Esto es especialmente importante en el caso de las aplicaciones de alta disponibilidad, en las que se requieren pruebas rigurosas para garantizar el cumplimiento de los objetivos de continuidad empresarial. Las pruebas proactivas de las capacidades de recuperación reducen el riesgo de fallas en la producción, lo que genera confianza en que la aplicación puede alcanzar el tiempo de recuperación limitado deseado. Las pruebas periódicas también aumentan la experiencia operativa, lo que permite al equipo recuperarse de forma rápida y fiable de las interrupciones cuando se producen. Poner en práctica el elemento humano, o proceso, de su enfoque de recuperación es tan importante como los aspectos técnicos.

## 4.f: Coste y complejidad

Las implicaciones económicas de una arquitectura multirregional se deben al aumento del uso de la infraestructura, la sobrecarga operativa y el tiempo dedicado a los recursos. Como se mencionó anteriormente, el costo de infraestructura en una región en espera es similar al costo de infraestructura en una región principal cuando se realiza el aprovisionamiento previo, por lo que duplica el costo total. Aprovechne la capacidad de forma que sea suficiente para las operaciones diarias y, al mismo tiempo, reserve suficiente capacidad de búfer para tolerar los picos de demanda. A continuación, configure los mismos límites en cada región.

Además, si va a adoptar una arquitectura activa-activa, es posible que tenga que realizar cambios en el nivel de la aplicación para ejecutarla correctamente en una arquitectura multirregional. El diseño y la operación de estos cambios pueden requerir mucho tiempo y recursos. Como mínimo, las organizaciones deben dedicar tiempo a comprender las dependencias técnicas y empresariales de cada región y a diseñar los procesos de conmutación por error y recuperación.

Los equipos también deberían realizar los ejercicios habituales de conmutación por error y conmutación por recuperación para sentirse cómodos con los manuales de instrucciones que se utilizarían durante un evento. Si bien estos ejercicios son cruciales para obtener los resultados

esperados de una inversión en varias regiones, representan un coste de oportunidad y consumen tiempo y recursos que se destinan a otras actividades.

## 4.g: Estrategia organizativa de conmutación por error multirregional

Regiones de AWS proporcionan límites de aislamiento de fallas que eviten las fallas correlacionadas y contengan el impacto de las Servicio de AWS deficiencias, cuando se producen, en una sola región. Puede usar estos límites de falla para crear aplicaciones multirregionales que consistan en réplicas independientes y aisladas de fallas en cada región para limitar los escenarios de destino compartido. Esto le permite crear aplicaciones multirregionales y utilizar una variedad de enfoques (desde el backup y la restauración hasta la versión piloto o activo-activa) para implementar su arquitectura multirregional. Sin embargo, las aplicaciones no suelen funcionar de forma aislada, así que considere tanto los componentes que utilizará como sus dependencias como parte de su estrategia de conmutación por error. Por lo general, varias aplicaciones funcionan juntas para respaldar una historia de usuario, que es una capacidad específica que se ofrece a un usuario final, como publicar una imagen y un subtítulo en una aplicación de redes sociales o realizar una compra en un sitio de comercio electrónico. Por ello, debe desarrollar una estrategia organizativa de conmutación por error multirregional que proporcione la coordinación y la coherencia necesarias para que su enfoque sea exitoso.

Hay cuatro estrategias de alto nivel entre las que las organizaciones pueden elegir para guiar un enfoque multirregional. Se enumeran desde el enfoque más detallado hasta el más amplio:

- Conmutación por error a nivel de componentes
- Conmutación por error de aplicaciones individuales
- Conmutación por error en el gráfico de dependencias
- Conmutación por error de toda la cartera de aplicaciones

Cada estrategia tiene ventajas y desventajas y aborda diferentes desafíos, como la flexibilidad de la toma de decisiones sobre la conmutación por error, la capacidad de probar las combinaciones de la conmutación por error, la presencia de un comportamiento modal y la inversión organizacional en la planificación y la implementación. Para analizar cada estrategia con más detalle, consulte la entrada del AWS blog [Cómo crear una](#) estrategia organizacional de conmutación por error multirregional.

## Guía clave

- Revise todas Servicio de AWS las cuotas para asegurarse de que son iguales en todas las regiones en las que operará la carga de trabajo.
- El proceso de despliegue debe centrarse en una región a la vez, en lugar de incluir a varias regiones simultáneamente.
- Las métricas adicionales, como el retraso en la replicación, son específicas de los escenarios multirregionales y deben supervisarse.
- Amplíe la supervisión de la carga de trabajo más allá de la región principal. Supervise las métricas de experiencia del cliente de cada región y mida estos datos desde fuera de cada región en la que se ejecute una carga de trabajo.
- Pruebe la conmutación por error y la conmutación por recuperación con regularidad. Implemente un único manual para los procesos de conmutación por error y conmutación por recuperación y utilícelo tanto para las pruebas como para los eventos en directo. Los manuales de ejecución para las pruebas y los eventos en directo no deberían ser diferentes.
- Comprenda las ventajas y desventajas de las estrategias de conmutación por error. Implemente un gráfico de dependencias o una estrategia de cartera de aplicaciones completa.

## Conclusión y recursos

Esta guía abordó los casos de uso más comunes de las arquitecturas multirregionales, los fundamentos de la implementación de estas arquitecturas y las implicaciones de este enfoque. Puede aplicar estos fundamentos a cualquier carga de trabajo y utilizar la información como marco para decidir si una arquitectura multirregional es el enfoque adecuado para su empresa.

Para obtener más información, consulte los siguientes recursos:

- [AWS Centro de arquitectura](#)
- [AWS Marco de buena arquitectura](#)
- [AWS Well-Architected Tool](#)
- [Creación de una estrategia organizacional de conmutación por error multirregional](#) (AWS entrada de blog)
- [AWS Capacidades multirregionales \(artículo de Re:post\)](#)AWS

# Historial de documentos

En la siguiente tabla, se describen cambios significativos de esta guía. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las [notificaciones RSS](#).

| Cambio   | Descripción   | Fecha                    |
|--|---|--------------------------|
| <a href="#">Actualizaciones para el conmutador de región ARC</a> | En las secciones <a href="#">3.c</a> y <a href="#">4.d</a> , se agregó información sobre el cambio de región de Amazon Application Recovery Controller (ARC) para gestionar las tareas de recuperación. | 29 de septiembre de 2025 |
| <a href="#">Actualizaciones</a>                                  | Actualizaciones a lo largo de la guía.  | 27 de diciembre de 2024  |
| <a href="#">Publicación inicial</a>                              | —   | 20 de diciembre de 2022  |

# AWS Glosario de orientación prescriptiva

Los siguientes son términos de uso común en las estrategias, guías y patrones proporcionados por la Guía AWS prescriptiva. Para sugerir entradas, utilice el enlace [Enviar comentarios](#) al final del glosario.

## Números

### Las 7 R

Siete estrategias de migración comunes para trasladar aplicaciones a la nube. Estas estrategias se basan en las 5 R que Gartner identificó en 2011 y consisten en lo siguiente:

- **Refactorizar/rediseñar:** traslade una aplicación y modifique su arquitectura mediante el máximo aprovechamiento de las características nativas en la nube para mejorar la agilidad, el rendimiento y la escalabilidad. Por lo general, esto implica trasladar el sistema operativo y la base de datos. Ejemplo: Migrar la base de datos de Oracle en las instalaciones a Amazon Aurora PostgreSQL-Compatible Edition.
- **Redefinir la plataforma (transportar y redefinir):** traslade una aplicación a la nube e introduzca algún nivel de optimización para aprovechar las capacidades de la nube. Ejemplo: Migrar la base de datos Oracle en las instalaciones a Amazon Relational Database Service (Amazon RDS) para Oracle en la nube de Nube de AWS.
- **Recomprar (readquirir):** cambie a un producto diferente, lo cual se suele llevar a cabo al pasar de una licencia tradicional a un modelo SaaS. Ejemplo: Migrar el sistema de administración de las relaciones con los clientes (CRM) a Salesforce.com.
- **Volver a alojar (migrar mediante lift-and-shift):** traslade una aplicación a la nube sin realizar cambios para aprovechar las capacidades de la nube. Ejemplo: Migrar la base de datos de Oracle en las instalaciones a Oracle en una instancia de EC2 en la Nube de AWS.
- **Reubicar:** (migrar el hipervisor mediante lift and shift): traslade la infraestructura a la nube sin comprar equipo nuevo, reescribir aplicaciones o modificar las operaciones actuales. Los servidores se migran de una plataforma en las instalaciones a un servicio en la nube para la misma plataforma. Ejemplo: migrar una Microsoft Hyper-V aplicación a AWS.
- **Retener (revisitar):** conserve las aplicaciones en el entorno de origen. Estas pueden incluir las aplicaciones que requieren una refactorización importante, que desee posponer para más adelante, y las aplicaciones heredadas que desee retener, ya que no hay ninguna justificación empresarial para migrarlas.

- Retirar: retire o elimine las aplicaciones que ya no sean necesarias en un entorno de origen.

## A

### ABAC

Consulte [control de acceso basado en atributos](#).

### servicios abstractos

Consulte [servicios administrados](#).

### ACID

Consulte [atomicidad, consistencia, aislamiento, durabilidad](#).

### migración activa-activa

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas (mediante una herramienta de replicación bidireccional o mediante operaciones de escritura doble) y ambas bases de datos gestionan las transacciones de las aplicaciones conectadas durante la migración. Este método permite la migración en lotes pequeños y controlados, en lugar de requerir una transición única. Es más flexible, pero requiere más trabajo que una [migración activa-pasiva](#).

### migración activa-pasiva

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas, pero solo la de origen gestiona las transacciones de las aplicaciones conectadas, mientras los datos se replican en la de destino. La base de datos de destino no acepta ninguna transacción durante la migración.

### función de agregación

Función SQL que actúa en un grupo de filas y calcula un único valor de devolución para el grupo. Entre los ejemplos de funciones de agregación se incluyen SUM y MAX.

## IA

Consulte [inteligencia artificial](#).

### AIOps

Consulte [operaciones de inteligencia artificial](#)

## anonimización

El proceso de eliminar permanentemente la información personal de un conjunto de datos. La anonimización puede ayudar a proteger la privacidad personal. Los datos anonimizados ya no se consideran datos personales.

## antipatrones

Una solución que se utiliza con frecuencia para un problema recurrente en el que la solución es contraproducente, ineficaz o menos eficaz que una alternativa.

## control de aplicaciones

Enfoque de seguridad que permite usar de manera exclusiva aplicaciones aprobadas para ayudar a proteger un sistema contra el malware.

## cartera de aplicaciones

Recopilación de información detallada sobre cada aplicación que utiliza una organización, incluido el costo de creación y mantenimiento de la aplicación y su valor empresarial. Esta información es clave para [el proceso de detección y análisis de la cartera](#) y ayuda a identificar y priorizar las aplicaciones que se van a migrar, modernizar y optimizar.

## inteligencia artificial (IA)

El campo de la informática que se dedica al uso de tecnologías informáticas para realizar funciones cognitivas que suelen estar asociadas a los seres humanos, como el aprendizaje, la resolución de problemas y el reconocimiento de patrones. Para más información, consulte [¿Qué es la inteligencia artificial?](#)

## operaciones de inteligencia artificial (AIOps)

El proceso de utilizar técnicas de machine learning para resolver problemas operativos, reducir los incidentes operativos y la intervención humana, y mejorar la calidad del servicio. Para obtener más información sobre cómo AIOps se utiliza en la estrategia de AWS migración, consulte la [guía de integración de operaciones](#).

## cifrado asimétrico

Algoritmo de cifrado que utiliza un par de claves, una clave pública para el cifrado y una clave privada para el descifrado. Puede compartir la clave pública porque no se utiliza para el descifrado, pero el acceso a la clave privada debe estar sumamente restringido.

## atomicidad, consistencia, aislamiento, durabilidad (ACID)

Conjunto de propiedades de software que garantizan la validez de los datos y la fiabilidad operativa de una base de datos, incluso en caso de errores, cortes de energía u otros problemas.

## control de acceso basado en atributos (ABAC)

La práctica de crear permisos detallados basados en los atributos del usuario, como el departamento, el puesto de trabajo y el nombre del equipo. Para obtener más información, consulte [ABAC AWS en la](#) documentación AWS Identity and Access Management (IAM).

## origen de datos fidedigno

Ubicación en la que se almacena la versión principal de los datos, que se considera la fuente de información más fiable. Puede copiar los datos del origen de datos autorizado a otras ubicaciones con el fin de procesarlos o modificarlos, por ejemplo, anonimizarlos, redactarlos o seudonimizarlos.

## Zona de disponibilidad

Una ubicación distinta dentro de una Región de AWS que está aislada de los fallos en otras zonas de disponibilidad y que proporciona una conectividad de red económica y de baja latencia a otras zonas de disponibilidad de la misma región.

## AWS Marco de adopción de la nube (AWS CAF)

Un marco de directrices y mejores prácticas AWS para ayudar a las organizaciones a desarrollar un plan eficiente y eficaz para migrar con éxito a la nube. AWS CAF organiza la orientación en seis áreas de enfoque denominadas perspectivas: negocios, personas, gobierno, plataforma, seguridad y operaciones. Las perspectivas empresariales, humanas y de gobernanza se centran en las habilidades y los procesos empresariales; las perspectivas de plataforma, seguridad y operaciones se centran en las habilidades y los procesos técnicos. Por ejemplo, la perspectiva humana se dirige a las partes interesadas que se ocupan de los Recursos Humanos (RR. HH.), las funciones del personal y la administración de las personas. Desde esta perspectiva, AWS CAF proporciona orientación para el desarrollo, la formación y la comunicación de las personas a fin de preparar a la organización para una adopción exitosa de la nube. Para obtener más información, consulte la [Página web de AWS CAF](#) y el [Documento técnico de AWS CAF](#).

## AWS Marco de calificación de la carga de trabajo (AWS WQF)

Herramienta que evalúa las cargas de trabajo de migración de bases de datos, recomienda estrategias de migración y proporciona estimaciones de trabajo. AWS WQF se incluye con AWS

Schema Conversion Tool ().AWS SCT Analiza los esquemas de bases de datos y los objetos de código, el código de las aplicaciones, las dependencias y las características de rendimiento y proporciona informes de evaluación.

## B

bot malicioso

[Bot](#) destinado a causar interrupciones o daños a personas u organizaciones.

BCP

Consulte [planificación de la continuidad del negocio](#).

gráfico de comportamiento

Una vista unificada e interactiva del comportamiento de los recursos y de las interacciones a lo largo del tiempo. Puede utilizar un gráfico de comportamiento con Amazon Detective para examinar los intentos de inicio de sesión fallidos, las llamadas sospechosas a la API y acciones similares. Para obtener más información, consulte [Datos en un gráfico de comportamiento](#) en la documentación de Detective.

sistema big-endian

Un sistema que almacena primero el byte más significativo. Consulte también [endianidad](#).

clasificación binaria

Un proceso que predice un resultado binario (una de las dos clases posibles). Por ejemplo, es posible que su modelo de ML necesite predecir problemas como “¿Este correo electrónico es spam o no es spam?” o “¿Este producto es un libro o un automóvil?”.

filtro de floración

Estructura de datos probabilística y eficiente en términos de memoria que se utiliza para comprobar si un elemento es miembro de un conjunto.

implementación azul/verde

Estrategia de implementación en la que se crean dos entornos separados, pero idénticos. La versión actual de la aplicación se ejecuta en un entorno (azul) y la nueva versión de la aplicación se ejecuta en el otro entorno (verde). Esta estrategia lo ayuda a hacer reversiones rápidas con un impacto mínimo.

## bot

Aplicación de software que ejecuta tareas automatizadas a través de Internet y simula la actividad o interacción humana. Algunos bots son útiles o beneficiosos, como los rastreadores web que indexan la información de Internet. Otros bots, conocidos como bots maliciosos, tienen como objetivo causar interrupciones o daños a personas u organizaciones.

## botnet

Redes de [bots](#) infectadas por [malware](#) y que están bajo el control de una sola parte, conocida como pastor de bots u operador de bots. Las botnets son el mecanismo más conocido para escalar los bots y su impacto.

## branch

Área contenida de un repositorio de código. La primera rama que se crea en un repositorio es la rama principal. Puede crear una rama nueva a partir de una rama existente y, a continuación, desarrollar características o corregir errores en la rama nueva. Una rama que se genera para crear una característica se denomina comúnmente rama de característica. Cuando la característica se encuentra lista para su lanzamiento, se vuelve a combinar la rama de característica con la rama principal. Para obtener más información, consulte [Acerca de las sucursales](#) (GitHub documentación).

## acceso de emergencia

En circunstancias excepcionales y mediante un proceso aprobado, es una forma rápida de que un usuario pueda acceder a un Cuenta de AWS sitio al que normalmente no tiene permisos de acceso. Para más información, consulte el indicador [Implement break-glass procedures](#) en la guía de AWS Well-Architected.

## estrategia de implementación sobre infraestructura existente

La infraestructura existente en su entorno. Al adoptar una estrategia de implementación sobre infraestructura existente para una arquitectura de sistemas, se diseña la arquitectura en función de las limitaciones de los sistemas y la infraestructura actuales. Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de [implementación desde cero](#).

## caché de búfer

El área de memoria donde se almacenan los datos a los que se accede con más frecuencia.

## capacidad empresarial

Lo que hace una empresa para generar valor (por ejemplo, ventas, servicio al cliente o marketing). Las arquitecturas de microservicios y las decisiones de desarrollo pueden estar impulsadas por las capacidades empresariales. Para obtener más información, consulte la sección [Organizado en torno a las capacidades empresariales](#) del documento técnico [Ejecutar microservicios en contenedores en AWS](#).

## planificación de la continuidad del negocio (BCP)

Plan que aborda el posible impacto de un evento disruptivo, como una migración a gran escala en las operaciones y permite a la empresa reanudar las operaciones rápidamente.

# C

## CAF

Consulte [AWS Cloud Adoption Framework](#).

## implementación canario

Lanzamiento lento e incremental de una versión para los usuarios finales. Cuando tenga mayor confianza en la nueva versión, la implementa y reemplaza la versión actual en su totalidad.

## CCoE

Consulte [Centro de excelencia en la nube](#).

## CDC

Consulte [captura de datos de cambios](#).

## captura de datos de cambio (CDC)

Proceso de seguimiento de los cambios en un origen de datos, como una tabla de base de datos, y registro de los metadatos relacionados con el cambio. Puede utilizar los CDC para diversos fines, como auditar o replicar los cambios en un sistema de destino para mantener la sincronización.

## ingeniería del caos

Introducción intencionada de fallos o eventos disruptivos para poner a prueba la resiliencia de un sistema. Puedes usar [AWS Fault Injection Service \(AWS FIS\)](#) para realizar experimentos que estresen tus AWS cargas de trabajo y evalúen su respuesta.

## CI/CD

Consulte [integración continua y entrega continua](#).

## clasificación

Un proceso de categorización que permite generar predicciones. Los modelos de ML para problemas de clasificación predicen un valor discreto. Los valores discretos siempre son distintos entre sí. Por ejemplo, es posible que un modelo necesite evaluar si hay o no un automóvil en una imagen.

## cifrado del cliente

Cifrado de datos localmente, antes de que el objetivo los Servicio de AWS reciba.

## Centro de excelencia en la nube (CCoE)

Equipo multidisciplinario que impulsa los esfuerzos de adopción de la nube en toda la organización, incluido el desarrollo de las prácticas recomendadas en la nube, la movilización de recursos, el establecimiento de plazos de migración y la dirección de la organización durante las transformaciones a gran escala. Para obtener más información, consulte las [publicaciones de CCoE](#) en el blog de estrategia Nube de AWS empresarial.

## computación en la nube

La tecnología en la nube que se utiliza normalmente para la administración de dispositivos de IoT y el almacenamiento de datos de forma remota. La computación en la nube suele estar relacionada con la tecnología de [computación de periferia](#).

## modelo operativo en la nube

En una organización de TI, el modelo operativo que se utiliza para crear, madurar y optimizar uno o más entornos de nube. Para obtener más información, consulte [Creación de su modelo operativo de nube](#).

## etapas de adopción de la nube

Las siguientes son las cuatro fases por las que suelen pasar las empresas cuando migran a la Nube de AWS:

- Proyecto: ejecución de algunos proyectos relacionados con la nube con fines de prueba de concepto y aprendizaje
- Fundamento: realizar inversiones fundamentales para escalar su adopción de la nube (p. ej., crear una landing zone, definir una CCoE, establecer un modelo de operaciones)

- Migración: migración de aplicaciones individuales
- Reinención: optimización de productos y servicios e innovación en la nube

Stephen Orban definió estas etapas en la entrada del blog [The Journey Toward Cloud-First & the Stages of Adoption en el blog Nube de AWS Enterprise Strategy](#). Para obtener información sobre su relación con la estrategia de AWS migración, consulte la guía de [preparación para la migración](#).

## CMDB

Consulte [base de datos de administración de configuración](#).

## repositorio de código

Una ubicación donde el código fuente y otros activos, como documentación, muestras y scripts, se almacenan y actualizan mediante procesos de control de versiones. Algunos repositorios en la nube comunes son GitHub o Bitbucket Cloud. Cada versión del código se denomina rama. En una estructura de microservicios, cada repositorio se encuentra dedicado a una única funcionalidad. Una sola canalización de CI/CD puede utilizar varios repositorios.

## caché en frío

Una caché de búfer que está vacía no está bien poblada o contiene datos obsoletos o irrelevantes. Esto afecta al rendimiento, ya que la instancia de la base de datos debe leer desde la memoria principal o el disco, lo que es más lento que leer desde la memoria caché del búfer.

## datos fríos

Datos a los que se accede con poca frecuencia y que suelen ser históricos. Al consultar este tipo de datos, normalmente se aceptan consultas lentas. Trasladar estos datos a niveles o clases de almacenamiento de menor rendimiento y menos costosos puede reducir los costos.

## visión artificial (CV)

Campo de la [IA](#) que utiliza el machine learning para analizar y extraer información de formatos visuales, como imágenes y videos digitales. Por ejemplo, Amazon SageMaker AI proporciona algoritmos de procesamiento de imágenes para CV.

## deriva de configuración

En el caso de una carga de trabajo, un cambio en la configuración con respecto al estado esperado. Podría provocar que la carga de trabajo deje de cumplir las normas y, por lo general, es gradual e involuntaria.

## base de datos de administración de configuración (CMDB)

Repositorio que almacena y administra información sobre una base de datos y su entorno de TI, incluidos los componentes de hardware y software y sus configuraciones. Por lo general, los datos de una CMDB se utilizan en la etapa de detección y análisis de la cartera de productos durante la migración.

## paquete de conformidad

Un conjunto de AWS Config reglas y medidas correctivas que puede reunir para personalizar sus controles de conformidad y seguridad. Puede implementar un paquete de conformidad como una entidad única en una región Cuenta de AWS y, o en una organización, mediante una plantilla YAML. Para obtener más información, consulta los [paquetes de conformidad](#) en la documentación. AWS Config

## integración y entrega continuas (CI/CD)

El proceso de automatización de las etapas de origen, compilación, prueba, puesta en escena y producción del proceso de publicación del software. CI/CD se describe comúnmente como una canalización. CI/CD puede ayudarlo a automatizar los procesos, mejorar la productividad, mejorar la calidad del código y entregar más rápido. Para obtener más información, consulte [Beneficios de la entrega continua](#). CD también puede significar implementación continua. Para obtener más información, consulte [Entrega continua frente a implementación continua](#).

## CV

Consulte [visión artificial](#).

## D

### datos en reposo

Datos que están estacionarios en la red, como los datos que se encuentran almacenados.

### clasificación de datos

Un proceso para identificar y clasificar los datos de su red en función de su importancia y sensibilidad. Es un componente fundamental de cualquier estrategia de administración de riesgos de ciberseguridad porque lo ayuda a determinar los controles de protección y retención adecuados para los datos. La clasificación de datos es un componente del pilar de seguridad del AWS Well-Architected Framework. Para obtener más información, consulte [Clasificación de datos](#).

## deriva de datos

Una variación significativa entre los datos de producción y los datos que se utilizaron para entrenar un modelo de machine learning, o un cambio significativo en los datos de entrada a lo largo del tiempo. La deriva de datos puede reducir la calidad, la precisión y la imparcialidad generales de las predicciones de los modelos de machine learning.

## datos en tránsito

Datos que se mueven de forma activa por la red, por ejemplo, entre los recursos de la red.

## mallado de datos

Marco de arquitectura que proporciona una propiedad de datos distribuida y descentralizada con una administración y una gobernanza centralizadas.

## minimización de datos

El principio de recopilar y procesar solo los datos estrictamente necesarios. Practicar la minimización de los datos Nube de AWS puede reducir los riesgos de privacidad, los costos y la huella de carbono de la analítica.

## perímetro de datos

Un conjunto de barreras preventivas en su AWS entorno que ayudan a garantizar que solo las identidades confiables accedan a los recursos confiables desde las redes esperadas. Para obtener más información, consulte [Crear un perímetro de datos sobre](#) AWS

## preprocesamiento de datos

Transformar los datos sin procesar en un formato que su modelo de ML pueda analizar fácilmente. El preprocesamiento de datos puede implicar eliminar determinadas columnas o filas y corregir los valores faltantes, incoherentes o duplicados.

## procedencia de los datos

El proceso de rastrear el origen y el historial de los datos a lo largo de su ciclo de vida, por ejemplo, la forma en que se generaron, transmitieron y almacenaron los datos.

## titular de los datos

Persona cuyos datos se recopilan y procesan.

## almacenamiento de datos

Sistema de administración de datos que respalda la inteligencia empresarial, como los análisis. Los almacenes de datos suelen contener grandes cantidades de datos históricos y, por lo general, se utilizan para las consultas y los análisis.

## lenguaje de definición de datos (DDL)

Instrucciones o comandos para crear o modificar la estructura de tablas y objetos de una base de datos.

## lenguaje de manipulación de datos (DML)

Instrucciones o comandos para modificar (insertar, actualizar y eliminar) la información de una base de datos.

## DDL

Consulte [lenguaje de definición de bases de datos](#).

## conjunto profundo

Combinar varios modelos de aprendizaje profundo para la predicción. Puede utilizar conjuntos profundos para obtener una predicción más precisa o para estimar la incertidumbre de las predicciones.

## aprendizaje profundo

Un subcampo del ML que utiliza múltiples capas de redes neuronales artificiales para identificar el mapeo entre los datos de entrada y las variables objetivo de interés.

## defense-in-depth

Un enfoque de seguridad de la información en el que se distribuyen cuidadosamente una serie de mecanismos y controles de seguridad en una red informática para proteger la confidencialidad, la integridad y la disponibilidad de la red y de los datos que contiene. Al adoptar esta estrategia AWS, se añaden varios controles en diferentes capas de la AWS Organizations estructura para ayudar a proteger los recursos. Por ejemplo, un defense-in-depth enfoque podría combinar la autenticación multifactorial, la segmentación de la red y el cifrado.

## administrador delegado

En AWS Organizations, un servicio compatible puede registrar una cuenta de AWS miembro para administrar las cuentas de la organización y gestionar los permisos de ese servicio. Esta

cuenta se denomina administrador delegado para ese servicio. Para obtener más información y una lista de servicios compatibles, consulte [Servicios que funcionan con AWS Organizations](#) en la documentación de AWS Organizations .

## Implementación

El proceso de hacer que una aplicación, características nuevas o correcciones de código se encuentren disponibles en el entorno de destino. La implementación abarca implementar cambios en una base de código y, a continuación, crear y ejecutar esa base en los entornos de la aplicación.

### entorno de desarrollo

Consulte [entorno](#).

### control de detección

Un control de seguridad que se ha diseñado para detectar, registrar y alertar después de que se produzca un evento. Estos controles son una segunda línea de defensa, ya que lo advierten sobre los eventos de seguridad que han eludido los controles preventivos establecidos. Para obtener más información, consulte [Controles de detección](#) en Implementación de controles de seguridad en AWS.

### asignación de flujos de valor para el desarrollo (DVSM)

Proceso que se utiliza para identificar y priorizar las restricciones que afectan negativamente a la velocidad y la calidad en el ciclo de vida del desarrollo de software. DVSM amplía el proceso de asignación del flujo de valor diseñado originalmente para las prácticas de fabricación ajustada. Se centra en los pasos y los equipos necesarios para crear y transferir valor a través del proceso de desarrollo de software.

### gemelo digital

Representación virtual de un sistema del mundo real, como un edificio, una fábrica, un equipo industrial o una línea de producción. Los gemelos digitales son compatibles con el mantenimiento predictivo, la supervisión remota y la optimización de la producción.

### tabla de dimensiones

En un [esquema en estrella](#), tabla más pequeña que contiene los atributos de datos sobre los datos cuantitativos en una tabla de hechos. Los atributos de la tabla de dimensiones suelen ser campos de texto o números discretos que se comportan como texto. Estos atributos se suelen utilizar para restringir consultas, filtrarlas y etiquetar los conjuntos de resultados.

## desastre

Un evento que impide que una carga de trabajo o un sistema cumplan sus objetivos empresariales en su ubicación principal de implementación. Estos eventos pueden ser desastres naturales, fallos técnicos o el resultado de acciones humanas, como una configuración incorrecta involuntaria o un ataque de malware.

## recuperación de desastres (DR)

Estrategia y proceso que utiliza para minimizar el tiempo de inactividad y la pérdida de datos a causa de un [desastre](#). Para obtener más información, consulte [Recuperación ante desastres de cargas de trabajo en AWS: Recovery in the Cloud in the AWS Well-Architected Framework](#).

## DML

Consulte [lenguaje de manipulación de bases de datos](#).

## diseño basado en el dominio

Un enfoque para desarrollar un sistema de software complejo mediante la conexión de sus componentes a dominios en evolución, o a los objetivos empresariales principales, a los que sirve cada componente. Este concepto lo introdujo Eric Evans en su libro, *Diseño impulsado por el dominio: abordando la complejidad en el corazón del software* (Boston: Addison-Wesley Professional, 2003). Para obtener información sobre cómo utilizar el diseño basado en dominios con el patrón de higos estranguladores, consulte [Modernización gradual de los servicios web antiguos de Microsoft ASP.NET \(ASMX\) mediante contenedores y Amazon API Gateway](#).

## DR

Consulte [recuperación ante desastres](#).

## Detección de desviaciones

Seguimiento de las desviaciones con respecto a una configuración con línea de base. Por ejemplo, puedes usarlo AWS CloudFormation para [detectar desviaciones en los recursos del sistema](#) o puedes usarlo AWS Control Tower para [detectar cambios en tu landing zone](#) que puedan afectar al cumplimiento de los requisitos de gobierno.

## DVSM

Consulte [asignación de flujos de valor para el desarrollo](#).

# E

## EDA

Consulte [análisis de datos de tipo exploratorio](#).

## EDI

Consulte [intercambio electrónico de datos](#).

## computación en la periferia

La tecnología que aumenta la potencia de cálculo de los dispositivos inteligentes en la periferia de una red de IoT. En comparación con la [computación en la nube](#), la computación de periferia puede reducir la latencia de la comunicación y mejorar el tiempo de respuesta.

## intercambio electrónico de datos (EDI)

Intercambio automatizado de documentos comerciales entre organizaciones. Para más información, consulte [¿Qué es el intercambio electrónico de datos?](#)

## cifrado

Proceso de computación que transforma datos de texto plano, que son legibles por humanos, en texto cifrado.

## clave de cifrado

Cadena criptográfica de bits aleatorios que se genera mediante un algoritmo de cifrado. Las claves pueden variar en longitud y cada una se ha diseñado para ser impredecible y única.

## endianidad

El orden en el que se almacenan los bytes en la memoria del ordenador. Los sistemas big-endianos almacenan primero el byte más significativo. Los sistemas Little-Endian almacenan primero el byte menos significativo.

## punto de conexión

Consulte [punto de conexión de servicio](#).

## servicio de punto de conexión

Servicio que puede alojar en una nube privada virtual (VPC) para compartir con otros usuarios. Puede crear un servicio de punto final AWS PrivateLink y conceder permisos a otras Cuentas de AWS o a responsables AWS Identity and Access Management (de IAM). Estas cuentas o

entidades principales pueden conectarse a su servicio de punto de conexión de forma privada mediante la creación de puntos de conexión de VPC de interfaz. Para obtener más información, consulte [Creación de un servicio de punto de conexión](#) en la documentación de Amazon Virtual Private Cloud (Amazon VPC).

planificación de recursos empresariales (ERP)

Sistema que automatiza y administra los procesos empresariales clave (como la contabilidad, [MES](#) y la administración de proyectos) de una empresa.

cifrado de sobre

El proceso de cifrar una clave de cifrado con otra clave de cifrado. Para obtener más información, consulte el [cifrado de sobres](#) en la documentación de AWS Key Management Service (AWS KMS).

entorno

Una instancia de una aplicación en ejecución. Los siguientes son los tipos de entornos más comunes en la computación en la nube:

- entorno de desarrollo: instancia de una aplicación en ejecución que solo se encuentra disponible para el equipo principal responsable del mantenimiento de la aplicación. Los entornos de desarrollo se utilizan para probar los cambios antes de promocionarlos a los entornos superiores. Este tipo de entorno a veces se denomina entorno de prueba.
- entornos inferiores: todos los entornos de desarrollo de una aplicación, como los que se utilizan para las compilaciones y pruebas iniciales.
- entorno de producción: instancia de una aplicación en ejecución a la que pueden acceder los usuarios finales. En un CI/CD proceso, el entorno de producción es el último entorno de implementación.
- entornos superiores: todos los entornos a los que pueden acceder usuarios que no sean del equipo de desarrollo principal. Esto puede incluir un entorno de producción, entornos de preproducción y entornos para las pruebas de aceptación por parte de los usuarios.

epopeya

En las metodologías ágiles, son categorías funcionales que ayudan a organizar y priorizar el trabajo. Las epopeyas brindan una descripción detallada de los requisitos y las tareas de implementación. Por ejemplo, las epopeyas AWS de seguridad de CAF incluyen la gestión de identidades y accesos, los controles de detección, la seguridad de la infraestructura, la protección de datos y la respuesta a incidentes. Para obtener más información sobre las epopeyas en la estrategia de migración de AWS, consulte la [Guía de implementación del programa](#).

## ERP

Consulte [planificación de recursos empresariales](#).

### análisis de datos de tipo exploratorio (EDA)

El proceso de analizar un conjunto de datos para comprender sus características principales. Se recopilan o agregan datos y, a continuación, se realizan las investigaciones iniciales para encontrar patrones, detectar anomalías y comprobar las suposiciones. El EDA se realiza mediante el cálculo de estadísticas resumidas y la creación de visualizaciones de datos.

## F

### tabla de hechos

Tabla central de un [esquema en estrella](#). Almacena datos cuantitativos sobre operaciones empresariales. Por lo general, una tabla de hechos contiene dos tipos de columnas: las que contienen medidas y las que contienen una clave externa para una tabla de dimensiones.

### Fail Fast

Filosofía que utiliza pruebas frecuentes e incrementales para reducir el ciclo de vida del desarrollo. Es una parte fundamental de los enfoques ágiles.

### límite de aislamiento de errores

En el Nube de AWS, un límite, como una zona de disponibilidad Región de AWS, un plano de control o un plano de datos, que limita el efecto de una falla y ayuda a mejorar la resiliencia de las cargas de trabajo. Para más información, consulte [AWS Fault Isolation Boundaries](#).

### rama de característica

Consulte [rama](#).

### características

Los datos de entrada que se utilizan para hacer una predicción. Por ejemplo, en un contexto de fabricación, las características pueden ser imágenes que se capturan periódicamente desde la línea de fabricación.

### importancia de las características

La importancia que tiene una característica para las predicciones de un modelo. Por lo general, esto se expresa como una puntuación numérica que se puede calcular mediante diversas

técnicas, como las explicaciones aditivas de Shapley (SHAP) y los gradientes integrados. Para obtener más información, consulte [Interpretabilidad del modelo de aprendizaje automático](#) con AWS

## transformación de funciones

Optimizar los datos para el proceso de ML, lo que incluye enriquecer los datos con fuentes adicionales, escalar los valores o extraer varios conjuntos de información de un solo campo de datos. Esto permite que el modelo de ML se beneficie de los datos. Por ejemplo, si divide la fecha del “27 de mayo de 2021 00:15:37” en “jueves”, “mayo”, “2021” y “15”, puede ayudar al algoritmo de aprendizaje a aprender patrones matizados asociados a los diferentes componentes de los datos.

## peticiones con pocos pasos

Proporcionar a un [LLM](#) una pequeña cantidad de ejemplos que demuestren la tarea y el resultado deseado antes de pedirle que lleve a cabo una tarea similar. Esta técnica es una aplicación del aprendizaje contextual, mediante el que los modelos aprenden a partir de ejemplos (pasos) incrustados en las peticiones. La técnica de peticiones con pocos pasos puede ser eficaz para las tareas que requieren un formato, un razonamiento o un conocimiento del dominio específicos. Consulte también [peticiones desde cero](#).

## FGAC

Consulte [control de acceso detallado](#).

## control de acceso preciso (FGAC)

El uso de varias condiciones que tienen por objetivo permitir o denegar una solicitud de acceso.  
migración relámpago

Método de migración de bases de datos que utiliza la replicación continua de datos mediante la [captura de datos de cambio](#) para migrar los datos en el menor tiempo posible, en lugar de utilizar un enfoque gradual. El objetivo es reducir al mínimo el tiempo de inactividad.

## FM

Consulte [modelo fundacional](#).

## Modelo fundacional (FM)

Una gran red neuronal de aprendizaje profundo que se ha estado entrenando con conjuntos de datos masivos de datos generalizados y sin etiquetar. FMs son capaces de realizar una amplia variedad de tareas generales, como comprender el lenguaje, generar texto e imágenes

y conversar en lenguaje natural. Para más información, consulte [¿Qué son los modelos fundacionales?](#)

## G

### IA generativa

Subconjunto de modelos de [IA](#) que se entrenaron con grandes cantidades de datos y que pueden utilizar una simple petición de texto para crear contenido y artefactos nuevos, como imágenes, videos, texto y audio. Para más información, consulte [¿Qué es la IA generativa?](#)

### bloqueo geográfico

Consulte [restricciones geográficas](#).

### restricciones geográficas (bloqueo geográfico)

En Amazon CloudFront, una opción para impedir que los usuarios de países específicos accedan a las distribuciones de contenido. Puede utilizar una lista de permitidos o bloqueados para especificar los países aprobados y prohibidos. Para obtener más información, consulta [la sección Restringir la distribución geográfica del contenido](#) en la CloudFront documentación.

### Flujo de trabajo de Gitflow

Un enfoque en el que los entornos inferiores y superiores utilizan diferentes ramas en un repositorio de código fuente. El flujo de trabajo de Gitflow se considera heredado, mientras que el [flujo de trabajo basado en enlaces troncales](#) es el enfoque moderno preferido.

### imagen dorada

Instantánea de un sistema o software que se usa como plantilla para implementar nuevas instancias de ese sistema o software. Por ejemplo, en la fabricación, una imagen dorada se puede utilizar para aprovisionar software en varios dispositivos y ayuda a mejorar la velocidad, la escalabilidad y la productividad de las operaciones de fabricación de dispositivos.

### estrategia de implementación desde cero

La ausencia de infraestructura existente en un entorno nuevo. Al adoptar una estrategia de implementación desde cero para una arquitectura de sistemas, puede seleccionar todas las tecnologías nuevas sin que estas deban ser compatibles con una infraestructura existente, lo que también se conoce como [implementación sobre infraestructura existente](#). Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de implementación desde cero.

## barrera de protección

Una regla de alto nivel que ayuda a regular los recursos, las políticas y el cumplimiento en todas las unidades organizativas (OUs). Las barreras de protección preventivas aplican políticas para garantizar la alineación con los estándares de conformidad. Se implementan mediante políticas de control de servicios y límites de permisos de IAM. Las barreras de protección de detección detectan las vulneraciones de las políticas y los problemas de conformidad, y generan alertas para su corrección. Se implementan mediante Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, Amazon Inspector y AWS Lambda cheques personalizados.

# H

## HA

Consulte [alta disponibilidad](#).

### migración heterogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que utilice un motor de base de datos diferente (por ejemplo, de Oracle a Amazon Aurora). La migración heterogénea suele ser parte de un esfuerzo de rediseño de la arquitectura y convertir el esquema puede ser una tarea compleja. [AWS ofrece AWS SCT](#), lo cual ayuda con las conversiones de esquemas.

### alta disponibilidad (HA)

La capacidad de una carga de trabajo para funcionar de forma continua, sin intervención, en caso de desafíos o desastres. Los sistemas de alta disponibilidad están diseñados para realizar una conmutación por error automática, ofrecer un rendimiento de alta calidad de forma constante y gestionar diferentes cargas y fallos con un impacto mínimo en el rendimiento.

### modernización histórica

Un enfoque utilizado para modernizar y actualizar los sistemas de tecnología operativa (TO) a fin de satisfacer mejor las necesidades de la industria manufacturera. Un histórico es un tipo de base de datos que se utiliza para recopilar y almacenar datos de diversas fuentes en una fábrica.

### datos de reserva

Parte de los datos históricos etiquetados que se ocultan de un conjunto de datos que se utiliza para entrenar un modelo de [machine learning](#). Puede utilizar los datos de reserva para evaluar el rendimiento del modelo mediante la comparación de las predicciones del modelo con los datos de reserva.

## migración homogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que comparte el mismo motor de base de datos (por ejemplo, Microsoft SQL Server a Amazon RDS para SQL Server). La migración homogénea suele formar parte de un esfuerzo para volver a alojar o redefinir la plataforma. Puede utilizar las utilidades de bases de datos nativas para migrar el esquema.

## datos recientes

Datos a los que se accede con frecuencia, como datos en tiempo real o datos traslacionales recientes. Por lo general, estos datos requieren un nivel o una clase de almacenamiento de alto rendimiento para proporcionar respuestas rápidas a las consultas.

## hotfix

Una solución urgente para un problema crítico en un entorno de producción. Debido a su urgencia, una revisión suele realizarse fuera del flujo de trabajo de DevOps publicación típico.

## periodo de hiperatención

Periodo, inmediatamente después de la transición, durante el cual un equipo de migración administra y monitorea las aplicaciones migradas en la nube para solucionar cualquier problema. Por lo general, este periodo dura de 1 a 4 días. Al final del periodo de hiperatención, el equipo de migración suele transferir la responsabilidad de las aplicaciones al equipo de operaciones en la nube.

## I

## IaC

Consulte [infraestructura como código](#).

## políticas basadas en identidades

Política asociada a uno o más directores de IAM que define sus permisos en el entorno. Nube de AWS

## aplicación inactiva

Aplicación que utiliza un promedio de CPU y memoria de entre 5 y 20 por ciento durante un periodo de 90 días. En un proyecto de migración, es habitual retirar estas aplicaciones o mantenerlas en las instalaciones.

## IloT

Consulte [Internet de las cosas industrial](#).

### infraestructura inmutable

Modelo que implementa una nueva infraestructura para las cargas de trabajo de producción en lugar de actualizar o modificar la infraestructura existente o aplicarle revisiones. Las infraestructuras inmutables son de manera intrínseca más coherentes, fiables y predecibles que las [infraestructuras mutables](#). Para más información, consulte la práctica recomendada [Implementación mediante una infraestructura inmutable](#) en el Marco de AWS Well-Architected.

### VPC entrante (de entrada)

En una arquitectura de AWS cuentas múltiples, una VPC que acepta, inspecciona y enruta las conexiones de red desde fuera de una aplicación. La [arquitectura AWS de referencia de seguridad](#) recomienda configurar la cuenta de red con entradas, salidas e inspección VPCs para proteger la interfaz bidireccional entre la aplicación y el resto de Internet.

### migración gradual

Estrategia de transición en la que se migra la aplicación en partes pequeñas en lugar de realizar una transición única y completa. Por ejemplo, puede trasladar inicialmente solo unos pocos microservicios o usuarios al nuevo sistema. Tras comprobar que todo funciona correctamente, puede trasladar microservicios o usuarios adicionales de forma gradual hasta que pueda retirar su sistema heredado. Esta estrategia reduce los riesgos asociados a las grandes migraciones.

### Industria 4.0

Término que introdujo [Klaus Schwab](#) en 2016 para referirse a la modernización de los procesos de fabricación mediante los avances en la conectividad, los datos en tiempo real, la automatización, el análisis, la IA y el ML.

### infraestructura

Todos los recursos y activos que se encuentran en el entorno de una aplicación.

### infraestructura como código (IaC)

Proceso de aprovisionamiento y administración de la infraestructura de una aplicación mediante un conjunto de archivos de configuración. La IaC se ha diseñado para ayudarlo a centralizar la administración de la infraestructura, estandarizar los recursos y escalar con rapidez a fin de que los entornos nuevos sean repetibles, fiables y consistentes.

## Internet de las cosas industrial (T) Ilo

El uso de sensores y dispositivos conectados a Internet en los sectores industriales, como el productivo, el eléctrico, el automotriz, el sanitario, el de las ciencias de la vida y el de la agricultura. Para obtener más información, consulte [Creación de una estrategia de transformación digital de la Internet de las cosas \(IIoT\) industrial](#).

## VPC de inspección

En una arquitectura de AWS cuentas múltiples, una VPC centralizada que gestiona las inspecciones del tráfico de red VPCs entre Internet y las redes locales (en una misma o Regiones de AWS diferente). La [arquitectura AWS de referencia de seguridad](#) recomienda configurar su cuenta de red con entrada, salida e inspección VPCs para proteger la interfaz bidireccional entre la aplicación e Internet en general.

## Internet de las cosas (IoT)

Red de objetos físicos conectados con sensores o procesadores integrados que se comunican con otros dispositivos y sistemas a través de Internet o de una red de comunicación local. Para obtener más información, consulte [¿Qué es IoT?](#).

## interpretabilidad

Característica de un modelo de machine learning que describe el grado en que un ser humano puede entender cómo las predicciones del modelo dependen de sus entradas. Para obtener más información, consulte Interpretabilidad del [modelo de aprendizaje automático](#) con AWS

## IoT

Consulte [Internet de las cosas](#).

## biblioteca de información de TI (ITIL)

Conjunto de prácticas recomendadas para ofrecer servicios de TI y alinearlos con los requisitos empresariales. La ITIL proporciona la base para la ITSM.

## administración de servicios de TI (ITSM)

Actividades asociadas con el diseño, la implementación, la administración y el soporte de los servicios de TI para una organización. Para obtener información sobre la integración de las operaciones en la nube con las herramientas de ITSM, consulte la [Guía de integración de operaciones](#).

## ITIL

Consulte [biblioteca de información de TI](#).

## ITSM

Consulte [administración de servicios de TI](#).

## L

### control de acceso basado en etiquetas (LBAC)

Una implementación del control de acceso obligatorio (MAC) en la que a los usuarios y a los propios datos se les asigna explícitamente un valor de etiqueta de seguridad. La intersección entre la etiqueta de seguridad del usuario y la etiqueta de seguridad de los datos determina qué filas y columnas puede ver el usuario.

### zona de aterrizaje

Una landing zone es un AWS entorno multicuenta bien diseñado, escalable y seguro. Este es un punto de partida desde el cual las empresas pueden lanzar e implementar rápidamente cargas de trabajo y aplicaciones con confianza en su entorno de seguridad e infraestructura. Para obtener más información sobre las zonas de aterrizaje, consulte [Configuración de un entorno de AWS seguro y escalable con varias cuentas](#).

### modelo de lenguaje de gran tamaño (LLM)

Modelo de [IA](#) de aprendizaje profundo que se entrenó previamente con una gran cantidad de datos. Un LLM puede llevar a cabo varias tareas, como responder preguntas, resumir documentos, traducir textos a otros idiomas y completar oraciones. [Para obtener más información, consulte Qué son. LLMs](#)

### migración grande

Migración de 300 servidores o más.

### LBAC

Consulte [control de acceso basado en etiquetas](#).

### privilegio mínimo

La práctica recomendada de seguridad que consiste en conceder los permisos mínimos necesarios para realizar una tarea. Para obtener más información, consulte [Aplicar permisos de privilegio mínimo](#) en la documentación de IAM.

### migrar mediante lift-and-shift

Consulte [Las 7 R](#).

## sistema little-endian

Un sistema que almacena primero el byte menos significativo. Consulte también [endianidad](#).

## LLM

Consulte [modelo de lenguaje de gran tamaño](#).

## entornos inferiores

Consulte [entorno](#).

# M

## machine learning (ML)

Un tipo de inteligencia artificial que utiliza algoritmos y técnicas para el reconocimiento y el aprendizaje de patrones. El ML analiza y aprende de los datos registrados, como los datos del Internet de las cosas (IoT), para generar un modelo estadístico basado en patrones. Para más información, consulte [Machine learning](#).

## rama principal

Consulte [rama](#).

## malware

Software diseñado para comprometer la seguridad o la privacidad de la computadora. El malware podría interrumpir los sistemas informáticos, filtrar información confidencial u obtener acceso no autorizado. Algunos ejemplos de malware son los virus, los gusanos, el ransomware, los troyanos, el spyware y los registradores de pulsaciones de teclas.

## Servicios administrados

Servicios de AWS para lo cual AWS opera la capa de infraestructura, el sistema operativo y las plataformas, y se accede a los puntos finales para almacenar y recuperar datos. Amazon Simple Storage Service (Amazon S3) y Amazon DynamoDB son ejemplos de servicios administrados. También se conocen como servicios abstractos.

## sistema de ejecución de fabricación (MES)

Sistema de software para seguir, supervisar, documentar y controlar los procesos de producción que convierten las materias primas en productos acabados en la zona de producción.

## MAP

Consulte [Programa de aceleración de la migración](#).

### mecanismo

Proceso completo mediante el que se crea una herramienta, se impulsa su adopción y, a continuación, se inspeccionan los resultados para hacer ajustes. Un mecanismo es un ciclo que se refuerza y mejora por sí mismo a medida que funciona. Para obtener más información, consulte [Creación de mecanismos](#) en el AWS Well-Architected Framework.

### cuenta de miembro

Todas las Cuentas de AWS demás cuentas, excepto la de administración, que forman parte de una organización. AWS Organizations Una cuenta no puede pertenecer a más de una organización a la vez.

## MES

Consulte [sistema de ejecución de fabricación](#).

### Message Queuing Telemetry Transport (MQTT)

[Un protocolo de comunicación ligero machine-to-machine \(M2M\), basado en el patrón de publicación/suscripción, para dispositivos de IoT con recursos limitados.](#)

### microservicio

Un servicio pequeño e independiente que se comunica a través de una red bien definida APIs y que, por lo general, es propiedad de equipos pequeños e independientes. Por ejemplo, un sistema de seguros puede incluir microservicios que se adapten a las capacidades empresariales, como las de ventas o marketing, o a subdominios, como las de compras, reclamaciones o análisis. Los beneficios de los microservicios incluyen la agilidad, la escalabilidad flexible, la facilidad de implementación, el código reutilizable y la resiliencia. Para obtener más información, consulte [Integrar microservicios mediante AWS servicios sin servidor](#).

### arquitectura de microservicios

Un enfoque para crear una aplicación con componentes independientes que ejecutan cada proceso de la aplicación como un microservicio. Estos microservicios se comunican a través de una interfaz bien definida mediante un uso ligero. APIs Cada microservicio de esta arquitectura se puede actualizar, implementar y escalar para satisfacer la demanda de funciones específicas de una aplicación. Para obtener más información, consulte [Implementación de microservicios](#) en AWS

## Programa de aceleración de la migración (MAP)

Un AWS programa que proporciona soporte de consultoría, formación y servicios para ayudar a las organizaciones a crear una base operativa sólida para migrar a la nube y para ayudar a compensar el costo inicial de las migraciones. El MAP incluye una metodología de migración para ejecutar las migraciones antiguas de forma metódica y un conjunto de herramientas para automatizar y acelerar los escenarios de migración más comunes.

### migración a escala

Proceso de transferencia de la mayoría de la cartera de aplicaciones a la nube en oleadas, con más aplicaciones desplazadas a un ritmo más rápido en cada oleada. En esta fase, se utilizan las prácticas recomendadas y las lecciones aprendidas en las fases anteriores para implementar una fábrica de migración de equipos, herramientas y procesos con el fin de agilizar la migración de las cargas de trabajo mediante la automatización y la entrega ágil. Esta es la tercera fase de la [estrategia de migración de AWS](#).

### fábrica de migración

Equipos multifuncionales que agilizan la migración de las cargas de trabajo mediante enfoques automatizados y ágiles. Los equipos de las fábricas de migración suelen incluir a analistas y propietarios de operaciones, empresas, ingenieros de migración, desarrolladores y DevOps profesionales que trabajan a pasos agigantados. Entre el 20 y el 50 por ciento de la cartera de aplicaciones empresariales se compone de patrones repetidos que pueden optimizarse mediante un enfoque de fábrica. Para obtener más información, consulte la [discusión sobre las fábricas de migración](#) y la [Guía de fábricas de migración a la nube](#) en este contenido.

### metadatos de migración

Información sobre la aplicación y el servidor que se necesita para completar la migración. Cada patrón de migración requiere un conjunto diferente de metadatos de migración. Algunos ejemplos de metadatos de migración son la subred de destino, el grupo de seguridad y AWS la cuenta.

### patrón de migración

Tarea de migración repetible que detalla la estrategia de migración, el destino de la migración y la aplicación o el servicio de migración utilizados. Ejemplo: rehospede la migración a Amazon EC2 AWS con Application Migration Service.

## Migration Portfolio Assessment (MPA)

Herramienta en línea que proporciona información a fin de validar los argumentos comerciales necesarios para migrar a la Nube de AWS. La MPA ofrece una evaluación detallada de la cartera

(adecuación del tamaño de los servidores, precios, comparaciones del costo total de propiedad, análisis de los costos de migración), así como una planificación de la migración (análisis y recopilación de datos de aplicaciones, agrupación de aplicaciones, priorización de la migración y planificación de oleadas). La [herramienta MPA](#) (requiere iniciar sesión) está disponible de forma gratuita para todos los AWS consultores y consultores de los socios de APN.

#### Evaluación de la preparación para la migración (MRA)

Proceso que consiste en obtener información sobre el estado de preparación de una organización para la nube, identificar sus puntos fuertes y débiles y elaborar un plan de acción para cerrar las brechas identificadas mediante el AWS CAF. Para obtener más información, consulte la [Guía de preparación para la migración](#). La MRA es la primera fase de la [estrategia de migración de AWS](#).

#### estrategia de migración

Enfoque utilizado para migrar una carga de trabajo a la Nube de AWS. Para más información, consulte la entrada [Las 7 R](#) de este glosario y también [Mobilize your organization to accelerate large-scale migrations](#).

#### ML

Consulte [machine learning](#).

#### modernización

Transformar una aplicación obsoleta (antigua o monolítica) y su infraestructura en un sistema ágil, elástico y de alta disponibilidad en la nube para reducir los gastos, aumentar la eficiencia y aprovechar las innovaciones. Para más información, consulte [Strategy for modernizing applications in the Nube de AWS](#).

#### evaluación de la preparación para la modernización

Evaluación que ayuda a determinar la preparación para la modernización de las aplicaciones de una organización; identifica los beneficios, los riesgos y las dependencias; y determina qué tan bien la organización puede soportar el estado futuro de esas aplicaciones. El resultado de la evaluación es un esquema de la arquitectura objetivo, una hoja de ruta que detalla las fases de desarrollo y los hitos del proceso de modernización y un plan de acción para abordar las brechas identificadas. Para más información, consulte [Evaluating modernization readiness for applications in the Nube de AWS](#).

#### aplicaciones monolíticas (monolitos)

Aplicaciones que se ejecutan como un único servicio con procesos estrechamente acoplados. Las aplicaciones monolíticas presentan varios inconvenientes. Si una característica de la

aplicación experimenta un aumento en la demanda, se debe escalar toda la arquitectura. Agregar o mejorar las características de una aplicación monolítica también se vuelve más complejo a medida que crece la base de código. Para solucionar problemas con la aplicación, puede utilizar una arquitectura de microservicios. Para obtener más información, consulte [Descomposición de monolitos en microservicios](#).

## MPA

Consulte [Migration Portfolio Assessment](#).

## MQTT

Consulte [Message Queuing Telemetry Transport](#).

## clasificación multiclase

Un proceso que ayuda a generar predicciones para varias clases (predice uno de más de dos resultados). Por ejemplo, un modelo de ML podría preguntar “¿Este producto es un libro, un automóvil o un teléfono?” o “¿Qué categoría de productos es más interesante para este cliente?”.

## infraestructura mutable

Modelo que actualiza y modifica la infraestructura actual para las cargas de trabajo de producción. Para mejorar la coherencia, la fiabilidad y la previsibilidad, el AWS Well-Architected Framework recomienda el uso [de una infraestructura inmutable](#) como práctica recomendada.

## O

### OAC

Consulte [control de acceso de origen](#).

### OAI

Consulte [identidad de acceso de origen](#).

### OCM

Consulte [administración del cambio organizacional](#).

## migración fuera de línea

Método de migración en el que la carga de trabajo de origen se elimina durante el proceso de migración. Este método implica un tiempo de inactividad prolongado y, por lo general, se utiliza para cargas de trabajo pequeñas y no críticas.

## OI

Consulte [integración de operaciones](#).

## OLA

Consulte [acuerdo de nivel operativo](#).

## migración en línea

Método de migración en el que la carga de trabajo de origen se copia al sistema de destino sin que se desconecte. Las aplicaciones que están conectadas a la carga de trabajo pueden seguir funcionando durante la migración. Este método implica un tiempo de inactividad nulo o mínimo y, por lo general, se utiliza para cargas de trabajo de producción críticas.

## OPC-UA

Consulte [Open Process Communications: arquitectura unificada](#).

## Open Process Communications: arquitectura unificada (OPC-UA)

Un protocolo de machine-to-machine comunicación (M2M) para la automatización industrial. OPC-UA establece un estándar de interoperabilidad con esquemas de autenticación, autorización y cifrado de datos.

## acuerdo de nivel operativo (OLA)

Acuerdo que aclara lo que los grupos de TI operativos se comprometen a ofrecerse entre sí, para respaldar un acuerdo de nivel de servicio (SLA).

## revisión de la preparación operativa (ORR)

Lista de comprobación de preguntas y prácticas recomendadas asociadas que son útiles para comprender, evaluar, prevenir o reducir el alcance de los incidentes y posibles errores. Para más información, consulte [Operational Readiness Reviews \(ORR\)](#) en el Marco de AWS Well-Architected.

## tecnología operativa (TO)

Sistemas de hardware y software que funcionan con el entorno físico para controlar las operaciones, los equipos y la infraestructura industriales. En el sector de la fabricación, la integración de los sistemas de TO y tecnología de la información (TI) es un enfoque clave para las transformaciones de la [industria 4.0](#).

## integración de operaciones (OI)

Proceso de modernización de las operaciones en la nube, que implica la planificación de la preparación, la automatización y la integración. Para obtener más información, consulte la [Guía de integración de las operaciones](#).

## registro de seguimiento organizativo

Un registro creado por y AWS CloudTrail que registra todos los eventos para todos los miembros Cuentas de AWS de una organización. AWS Organizations Este registro de seguimiento se crea en cada Cuenta de AWS que forma parte de la organización y realiza un seguimiento de la actividad en cada cuenta. Para obtener más información, consulte [Crear un registro para una organización](#) en la CloudTrail documentación.

## administración del cambio organizacional (OCM)

Marco para administrar las transformaciones empresariales importantes y disruptivas desde la perspectiva de las personas, la cultura y el liderazgo. La OCM ayuda a las empresas a prepararse para nuevos sistemas y estrategias y a realizar la transición a ellos, al acelerar la adopción de cambios, abordar los problemas de transición e impulsar cambios culturales y organizacionales. En la estrategia de AWS migración, este marco se denomina aceleración de personal, debido a la velocidad de cambio que requieren los proyectos de adopción de la nube. Para obtener más información, consulte la [Guía de OCM](#).

## control de acceso de origen (OAC)

En CloudFront, una opción mejorada para restringir el acceso y proteger el contenido del Amazon Simple Storage Service (Amazon S3). El OAC admite todos los buckets de S3 Regiones de AWS, el cifrado del lado del servidor AWS KMS (SSE-KMS) y las solicitudes dinámicas PUT y DELETE dirigidas al bucket de S3.

## identidad de acceso de origen (OAI)

En CloudFront, una opción para restringir el acceso y proteger el contenido de Amazon S3. Cuando utiliza OAI, CloudFront crea un principal con el que Amazon S3 puede autenticarse. Los directores autenticados solo pueden acceder al contenido de un bucket de S3 a través de una distribución específica. CloudFront Consulte también el [OAC](#), que proporciona un control de acceso más detallado y mejorado.

## ORR

Consulte [revisión de la preparación operativa](#).

## OT

Consulte [tecnología operativa](#).

## VPC saliente (de salida)

En una arquitectura de AWS cuentas múltiples, una VPC que gestiona las conexiones de red que se inician desde una aplicación. La [arquitectura AWS de referencia de seguridad](#) recomienda configurar la cuenta de red con entradas, salidas e inspección VPCs para proteger la interfaz bidireccional entre la aplicación e Internet en general.

## P

### límite de permisos

Una política de administración de IAM que se adjunta a las entidades principales de IAM para establecer los permisos máximos que puede tener el usuario o el rol. Para obtener más información, consulte [Límites de permisos](#) en la documentación de IAM.

### información de identificación personal (PII)

Información que, vista directamente o combinada con otros datos relacionados, puede utilizarse para deducir de manera razonable la identidad de una persona. Algunos ejemplos de información de identificación personal son los nombres, las direcciones y la información de contacto.

## PII

Consulte [información de identificación personal](#).

### manual de estrategias

Conjunto de pasos predefinidos que capturan el trabajo asociado a las migraciones, como la entrega de las funciones de operaciones principales en la nube. Un manual puede adoptar la forma de scripts, manuales de procedimientos automatizados o resúmenes de los procesos o pasos necesarios para operar un entorno modernizado.

## PLC

Consulte [controlador lógico programable](#).

## PLM

Consulte [administración del ciclo de vida del producto](#).

## policy

Objeto que puede definir permisos (consulte [política basada en identidad](#)), especificar las condiciones de acceso (consulte [política basada en recursos](#)) o definir los permisos máximos para todas las cuentas de una organización de AWS Organizations (consulte [política de control de servicio](#)).

## persistencia políglota

Elegir de forma independiente la tecnología de almacenamiento de datos de un microservicio en función de los patrones de acceso a los datos y otros requisitos. Si sus microservicios tienen la misma tecnología de almacenamiento de datos, pueden enfrentarse a desafíos de implementación o experimentar un rendimiento deficiente. Los microservicios se implementan más fácilmente y logran un mejor rendimiento y escalabilidad si utilizan el almacén de datos que mejor se adapte a sus necesidades.

## evaluación de cartera

Proceso de detección, análisis y priorización de la cartera de aplicaciones para planificar la migración. Para obtener más información, consulte la [Evaluación de la preparación para la migración](#).

## predicate

Condición de consulta que devuelve true o false. En general, se encuentra en una cláusula WHERE.

## inserción de predicados

Técnica de optimización de consultas en bases de datos que filtra los datos de la consulta antes de transferirlos. Esta técnica reduce la cantidad de datos de la base de datos relacional que se tienen que recuperar y procesar. Además, mejora el rendimiento de las consultas.

## control preventivo

Un control de seguridad diseñado para evitar que ocurra un evento. Estos controles son la primera línea de defensa para evitar el acceso no autorizado o los cambios no deseados en la red. Para obtener más información, consulte [Controles preventivos](#) en Implementación de controles de seguridad en AWS.

## entidad principal

Una entidad AWS que puede realizar acciones y acceder a los recursos. Esta entidad suele ser un usuario raíz para un Cuenta de AWS rol de IAM o un usuario. Para obtener más información, consulte Entidad principal en [Términos y conceptos de roles](#) en la documentación de IAM.

## Privacidad desde el diseño

Enfoque de ingeniería de sistemas que tiene en cuenta la privacidad durante todo el proceso de desarrollo.

### zonas alojadas privadas

Un contenedor que contiene información sobre cómo desea que Amazon Route 53 responda a las consultas de DNS de un dominio y sus subdominios dentro de uno o más VPCs. Para obtener más información, consulte [Uso de zonas alojadas privadas](#) en la documentación de Route 53.

### control proactivo

[Control de seguridad](#) que se diseñó para evitar la implementación de recursos que no cumplan con la normativa. Estos controles analizan los recursos antes de aprovisionarlos. Si el recurso no cumple con los requisitos del control, no se aprovisiona. Para obtener más información, consulte la [guía de referencia de controles](#) en la AWS Control Tower documentación y consulte [Controles proactivos](#) en la sección Implementación de controles de seguridad en AWS.

### administración del ciclo de vida del producto (PLM)

Administración de los datos y los procesos de un producto a lo largo de todo su ciclo de vida, desde el diseño, el desarrollo y el lanzamiento, pasando por el crecimiento y la madurez, hasta la reducción de su uso y su retirada.

### entorno de producción

Consulte [entorno](#).

### controlador lógico programable (PLC)

En el sector de la fabricación, computadora adaptable y altamente fiable que supervisa las máquinas y automatiza los procesos de fabricación.

### encadenamiento de peticiones

Uso de la salida de una petición de [LLM](#) como entrada para la siguiente petición a fin de generar mejores respuestas. Esta técnica se utiliza para dividir una tarea compleja en tareas secundarias o para refinar o ampliar de forma iterativa una respuesta preliminar. Ayuda a mejorar la precisión y la relevancia de las respuestas de un modelo y permite obtener resultados más detallados y personalizados.

## seudonimización

El proceso de reemplazar los identificadores personales de un conjunto de datos por valores de marcadores de posición. La seudonimización puede ayudar a proteger la privacidad personal. Los datos seudonimizados siguen considerándose datos personales.

## publish/subscribe (pub/sub)

Patrón que permite establecer comunicaciones asíncronas entre microservicios para mejorar la escalabilidad y la capacidad de respuesta. Por ejemplo, en un [MES](#) basado en microservicios, un microservicio puede publicar mensajes de eventos en un canal al que se pueden suscribir otros microservicios. El sistema puede agregar nuevos microservicios sin cambiar el servicio de publicación.

## Q

### plan de consulta

Serie de pasos, como instrucciones, que se utilizan para acceder a los datos de un sistema de base de datos relacional SQL.

### regresión del plan de consulta

El optimizador de servicios de la base de datos elige un plan menos óptimo que antes de un cambio determinado en el entorno de la base de datos. Los cambios en estadísticas, restricciones, configuración del entorno, enlaces de parámetros de consultas y actualizaciones del motor de base de datos PostgreSQL pueden provocar una regresión del plan.

## R

### Matriz RACI

Consulte [responsable, fiable, consultada e informada \(RACI\)](#).

### RAG

Consulte [generación aumentada por recuperación](#).

### ransomware

Software malicioso que se ha diseñado para bloquear el acceso a un sistema informático o a los datos hasta que se efectúe un pago.

## Matriz RASCI

Consulte [responsable, fiable, consultada e informada \(RACI\)](#).

## RCAC

Consulte [control de acceso por filas y columnas](#).

## réplica de lectura

Una copia de una base de datos que se utiliza con fines de solo lectura. Puede enrutar las consultas a la réplica de lectura para reducir la carga en la base de datos principal.

## rediseñar

Consulte [Las 7 R](#).

## objetivo de punto de recuperación (RPO)

La cantidad de tiempo máximo aceptable desde el último punto de recuperación de datos. Esto determina qué se considera una pérdida de datos aceptable entre el último punto de recuperación y la interrupción del servicio.

## objetivo de tiempo de recuperación (RTO)

La demora máxima aceptable entre la interrupción del servicio y el restablecimiento del servicio.

## refactorizar

Consulte [Las 7 R](#).

## Region

Conjunto de AWS recursos en un área geográfica. Cada uno Región de AWS está aislado e independiente de los demás para proporcionar tolerancia a las fallas, estabilidad y resiliencia. Para más información, consulte [Specify which Regions de AWS your account can use](#).

## regresión

Una técnica de ML que predice un valor numérico. Por ejemplo, para resolver el problema de “¿A qué precio se venderá esta casa?”, un modelo de ML podría utilizar un modelo de regresión lineal para predecir el precio de venta de una vivienda en función de datos conocidos sobre ella (por ejemplo, los metros cuadrados).

## volver a alojar

Consulte [Las 7 R](#).

## versión

En un proceso de implementación, el acto de promover cambios en un entorno de producción.

## reubicar

Consulte [Las 7 R](#).

## redefinir la plataforma

Consulte [Las 7 R](#).

## recomprar

Consulte [Las 7 R](#).

## resiliencia

Capacidad de una aplicación para resistir interrupciones o recuperarse de ellas. Al planificar la resiliencia en la Nube de AWS, la [alta disponibilidad](#) y la [recuperación ante desastres](#) son consideraciones comunes. Para más información, consulte [Resiliencia en la Nube de AWS](#).

## política basada en recursos

Una política asociada a un recurso, como un bucket de Amazon S3, un punto de conexión o una clave de cifrado. Este tipo de política especifica a qué entidades principales se les permite el acceso, las acciones compatibles y cualquier otra condición que deba cumplirse.

## matriz responsable, confiable, consultada e informada (RACI)

Una matriz que define las funciones y responsabilidades de todas las partes involucradas en las actividades de migración y las operaciones de la nube. El nombre de la matriz se deriva de los tipos de responsabilidad definidos en la matriz: responsable (R), contable (A), consultado (C) e informado (I). El tipo de soporte (S) es opcional. Si incluye el soporte, la matriz se denomina matriz RASCI y, si la excluye, se denomina matriz RACI.

## control receptivo

Un control de seguridad que se ha diseñado para corregir los eventos adversos o las desviaciones con respecto a su base de seguridad. Para obtener más información, consulte [Controles receptivos](#) en Implementación de controles de seguridad en AWS.

## retain

Consulte [Las 7 R](#).

## retirar

Consulte [Las 7 R](#).

## Generación aumentada de recuperación (RAG)

Tecnología de [IA generativa](#) mediante la que un [LLM](#) hace referencia a un origen de datos autorizado que se encuentra fuera de sus orígenes de datos de entrenamiento antes de generar una respuesta. Por ejemplo, un modelo de RAG podría hacer una búsqueda semántica en la base de conocimientos o en los datos personalizados de una organización. Para más información, consulte [¿Qué es RAG \(generación aumentada por recuperación\)?](#)

## rotación

Proceso mediante el que periódicamente se actualiza un [secreto](#) para que resulte más difícil que un atacante pueda acceder a las credenciales.

## control de acceso por filas y columnas (RCAC)

El uso de expresiones SQL básicas y flexibles que tienen reglas de acceso definidas. El RCAC consta de permisos de fila y máscaras de columnas.

## RPO

Consulte [objetivo de punto de recuperación](#).

## RTO

Consulte [objetivo de tiempo de recuperación](#).

## manual de procedimientos

Conjunto de procedimientos manuales o automatizados necesarios para realizar una tarea específica. Por lo general, se diseñan para agilizar las operaciones o los procedimientos repetitivos con altas tasas de error.

## S

### SAML 2.0

Un estándar abierto que utilizan muchos proveedores de identidad (IdPs). Esta función permite el inicio de sesión único (SSO) federado, de modo que los usuarios pueden iniciar sesión Consola de administración de AWS o llamar a las operaciones de la AWS API sin tener que crear un

usuario en IAM para todos los miembros de la organización. Para obtener más información sobre la federación basada en SAML 2.0, consulte [Acerca de la federación basada en SAML 2.0](#) en la documentación de IAM.

## SCADA

Consulte [control de supervisión y adquisición de datos](#).

## SCP

Consulte [política de control de servicio](#).

## secreta

En AWS Secrets Manager, información confidencial o restringida, como una contraseña o credenciales de usuario, que se almacena de forma cifrada. Se compone del valor del secreto y de sus metadatos. El valor del secreto puede ser binario, una sola cadena o varias cadenas. Para más información, consulte [What's in a Secrets Manager secret?](#) en la documentación de Secrets Manager.

## seguridad desde el diseño

Enfoque de ingeniería de sistemas que tiene en cuenta la seguridad durante todo el proceso de desarrollo.

## control de seguridad

Barrera de protección técnica o administrativa que impide, detecta o reduce la capacidad de un agente de amenazas para aprovechar una vulnerabilidad de seguridad. Existen cuatro tipos de controles de seguridad principales: [preventivos](#), [de detección](#), [de respuesta](#) y [proactivos](#).

## refuerzo de la seguridad

Proceso de reducir la superficie expuesta a ataques para hacerla más resistente a los ataques. Esto puede incluir acciones, como la eliminación de los recursos que ya no se necesitan, la implementación de prácticas recomendadas de seguridad consistente en conceder privilegios mínimos o la desactivación de características innecesarias en los archivos de configuración.

## sistema de información sobre seguridad y administración de eventos (SIEM)

Herramientas y servicios que combinan sistemas de administración de información sobre seguridad (SIM) y de administración de eventos de seguridad (SEM). Un sistema de SIEM recopila, monitorea y analiza los datos de servidores, redes, dispositivos y otras fuentes para detectar amenazas y brechas de seguridad y generar alertas.

## automatización de la respuesta de seguridad

Acción predefinida y programada que está diseñada para responder automáticamente a un evento de seguridad o corregirlo. Estas automatizaciones sirven como controles de seguridad [preventivos o adaptables](#) que le ayudan a implementar las mejores prácticas AWS de seguridad. La modificación de un grupo de seguridad de VPC, la aplicación de revisiones a una instancia de Amazon EC2 o la rotación de credenciales son algunos ejemplos de acciones de respuesta automatizadas.

## cifrado del servidor

Cifrado de los datos en su destino, por parte de Servicio de AWS quien los recibe.

## política de control de servicio (SCP)

Política que proporciona un control centralizado de los permisos de todas las cuentas de una organización en AWS Organizations. SCPs defina barreras o establezca límites a las acciones que un administrador puede delegar en usuarios o roles. Puede utilizarlas SCPs como listas de permitidos o rechazados para especificar qué servicios o acciones están permitidos o prohibidos. Para obtener más información, consulte [las políticas de control de servicios](#) en la AWS Organizations documentación.

## punto de enlace de servicio

La URL del punto de entrada de un Servicio de AWS. Para conectarse mediante programación a un servicio de destino, puede utilizar un punto de conexión. Para obtener más información, consulte [Puntos de conexión de Servicio de AWS](#) en Referencia general de AWS.

## acuerdo de nivel de servicio (SLA)

Acuerdo que aclara lo que un equipo de TI se compromete a ofrecer a los clientes, como el tiempo de actividad y el rendimiento del servicio.

## indicador de nivel de servicio (SLI)

Medición de un aspecto del rendimiento de un servicio, como la tasa de errores, la disponibilidad o el rendimiento.

## objetivo de nivel de servicio (SLO)

Métrica objetivo que representa el estado de un servicio medido mediante un [indicador de nivel de servicio](#).

## modelo de responsabilidad compartida

Un modelo que describe la responsabilidad con AWS la que compartes la seguridad y el cumplimiento de la nube. AWS es responsable de la seguridad de la nube, mientras que usted es responsable de la seguridad en la nube. Para obtener más información, consulte el [Modelo de responsabilidad compartida](#).

## SIEM

Consulte [sistema de administración de eventos e información de seguridad](#).

## único punto de error (SPOF)

Error en un único componente crítico de una aplicación que puede interrumpir el sistema.

## SLA

Consulte [acuerdo de nivel de servicio](#).

## SLI

Consulte [indicador de nivel de servicio](#).

## SLO

Consulte [objetivo de nivel de servicio](#).

## split-and-seed modelo

Un patrón para escalar y acelerar los proyectos de modernización. A medida que se definen las nuevas funciones y los lanzamientos de los productos, el equipo principal se divide para crear nuevos equipos de productos. Esto ayuda a ampliar las capacidades y los servicios de su organización, mejora la productividad de los desarrolladores y apoya la innovación rápida. Para más información, consulte [Phased approach to modernizing applications in the Nube de AWS](#).

## SPOF

Consulte [único punto de error](#).

## esquema en estrella

Estructura organizativa de una base de datos que utiliza una tabla de hechos de gran tamaño para almacenar datos transaccionales o medidos y una o varias tablas dimensionales más pequeñas para almacenar los atributos de los datos. Esta estructura está diseñada para utilizarse en un [almacén de datos](#) o con fines de inteligencia empresarial.

## patrón de higo estrangulador

Un enfoque para modernizar los sistemas monolíticos mediante la reescritura y el reemplazo gradual de las funciones del sistema hasta que se pueda desmantelar el sistema heredado. Este patrón utiliza la analogía de una higuera que crece hasta convertirse en un árbol estable y, finalmente, se apodera y reemplaza a su host. El patrón fue [presentado por Martin Fowler](#) como una forma de gestionar el riesgo al reescribir sistemas monolíticos. Para ver un ejemplo con la aplicación de este patrón, consulte [Modernización gradual de los servicios web antiguos de Microsoft ASP.NET \(ASMX\) mediante contenedores y Amazon API Gateway](#).

## subred

Un intervalo de direcciones IP en la VPC. Una subred debe residir en una sola zona de disponibilidad.

## control de supervisión y adquisición de datos (SCADA)

En el sector de la fabricación, sistema que utiliza hardware y software para supervisar los activos físicos y las operaciones de producción.

## cifrado simétrico

Un algoritmo de cifrado que utiliza la misma clave para cifrar y descifrar los datos.

## pruebas sintéticas

Prueba de un sistema de manera que simule las interacciones de los usuarios para detectar posibles problemas o supervisar el rendimiento. Puede usar [Amazon CloudWatch Synthetics](#) para crear estas pruebas.

## petición del sistema

Técnica para proporcionar contexto, instrucciones o pautas a un [LLM](#) para dirigir su comportamiento. Las peticiones del sistema ayudan a establecer el contexto y las reglas para las interacciones con los usuarios.

# T

## etiquetas

Pares clave-valor que actúan como metadatos para organizar los recursos. AWS Las etiquetas pueden ayudar a administrar, identificar, organizar, buscar y filtrar recursos de . Para obtener más información, consulte [Etiquetado de los recursos de AWS](#).

## variable de destino

El valor que intenta predecir en el ML supervisado. Esto también se conoce como variable de resultado. Por ejemplo, en un entorno de fabricación, la variable objetivo podría ser un defecto del producto.

## lista de tareas

Herramienta que se utiliza para hacer un seguimiento del progreso mediante un manual de procedimientos. La lista de tareas contiene una descripción general del manual de procedimientos y una lista de las tareas generales que deben completarse. Para cada tarea general, se incluye la cantidad estimada de tiempo necesario, el propietario y el progreso.

## entorno de prueba

Consulte [entorno](#).

## entrenamiento

Proporcionar datos de los que pueda aprender su modelo de ML. Los datos de entrenamiento deben contener la respuesta correcta. El algoritmo de aprendizaje encuentra patrones en los datos de entrenamiento que asignan los atributos de los datos de entrada al destino (la respuesta que desea predecir). Genera un modelo de ML que captura estos patrones. Luego, el modelo de ML se puede utilizar para obtener predicciones sobre datos nuevos para los que no se conoce el destino.

## puerta de enlace de tránsito

Un centro de tránsito de red que puede usar para interconectar sus redes con VPCs las locales. Para obtener más información, consulte [Qué es una pasarela de tránsito](#) en la AWS Transit Gateway documentación.

## flujo de trabajo basado en enlaces troncales

Un enfoque en el que los desarrolladores crean y prueban características de forma local en una rama de característica y, a continuación, combinan esos cambios en la rama principal. Luego, la rama principal se adapta a los entornos de desarrollo, preproducción y producción, de forma secuencial.

## acceso de confianza

Otorgar permisos a un servicio que especifique para realizar tareas en su organización AWS Organizations y en sus cuentas en su nombre. El servicio de confianza crea un rol vinculado al servicio en cada cuenta, cuando ese rol es necesario, para realizar las tareas de administración

por usted. Para obtener más información, consulte [AWS Organizations Utilización con otros AWS servicios](#) en la AWS Organizations documentación.

## ajuste

Cambiar aspectos de su proceso de formación a fin de mejorar la precisión del modelo de ML. Por ejemplo, puede entrenar el modelo de ML al generar un conjunto de etiquetas, incorporar etiquetas y, luego, repetir estos pasos varias veces con diferentes ajustes para optimizar el modelo.

## equipo de dos pizzas

Un DevOps equipo pequeño al que puedes alimentar con dos pizzas. Un equipo formado por dos integrantes garantiza la mejor oportunidad posible de colaboración en el desarrollo de software.

# U

## incertidumbre

Un concepto que hace referencia a información imprecisa, incompleta o desconocida que puede socavar la fiabilidad de los modelos predictivos de ML. Hay dos tipos de incertidumbre: la incertidumbre epistémica se debe a datos limitados e incompletos, mientras que la incertidumbre aleatoria se debe al ruido y la aleatoriedad inherentes a los datos. Para más información, consulte la guía [Cuantificación de la incertidumbre en los sistemas de aprendizaje profundo](#).

## tareas indiferenciadas

También conocido como tareas arduas, es el trabajo que es necesario para crear y operar una aplicación, pero que no proporciona un valor directo al usuario final ni proporciona una ventaja competitiva. Algunos ejemplos de tareas indiferenciadas son la adquisición, el mantenimiento y la planificación de la capacidad.

## entornos superiores

Consulte [entorno](#).

## V

### succión

Una operación de mantenimiento de bases de datos que implica limpiar después de las actualizaciones incrementales para recuperar espacio de almacenamiento y mejorar el rendimiento.

### control de versión

Procesos y herramientas que realizan un seguimiento de los cambios, como los cambios en el código fuente de un repositorio.

### Emparejamiento de VPC

Una conexión entre dos VPCs que le permite enrutar el tráfico mediante direcciones IP privadas. Para obtener más información, consulte [¿Qué es una interconexión de VPC?](#) en la documentación de Amazon VPC.

### vulnerabilidad

Defecto de software o hardware que pone en peligro la seguridad del sistema.

## W

### caché caliente

Un búfer caché que contiene datos actuales y relevantes a los que se accede con frecuencia. La instancia de base de datos puede leer desde la caché del búfer, lo que es más rápido que leer desde la memoria principal o el disco.

### datos templados

Datos a los que el acceso es infrecuente. Al consultar este tipo de datos, normalmente se aceptan consultas moderadamente lentas.

### función de ventana

Función SQL que hace un cálculo en un grupo de filas que se relacionan de alguna manera con el registro actual. Las funciones de ventana son útiles para las tareas de procesamiento, como calcular una media móvil o acceder al valor de las filas en función de la posición relativa de la fila actual.

## carga de trabajo

Conjunto de recursos y código que ofrece valor comercial, como una aplicación orientada al cliente o un proceso de backend.

## flujo de trabajo

Grupos funcionales de un proyecto de migración que son responsables de un conjunto específico de tareas. Cada flujo de trabajo es independiente, pero respalda a los demás flujos de trabajo del proyecto. Por ejemplo, el flujo de trabajo de la cartera es responsable de priorizar las aplicaciones, planificar las oleadas y recopilar los metadatos de migración. El flujo de trabajo de la cartera entrega estos recursos al flujo de trabajo de migración, que luego migra los servidores y las aplicaciones.

## WORM

Consulte [escritura única y lectura múltiple](#).

## WQF

Consulte [AWS Workload Qualification Framework](#).

## escritura única y lectura múltiple (WORM)

Modelo de almacenamiento que escribe los datos una sola vez y evita que se eliminen o modifiquen. Los usuarios autorizados pueden leer los datos tantas veces como sea necesario, pero no los pueden cambiar. Esta infraestructura de almacenamiento de datos se considera [inmutable](#).

## Z

### ataque de día cero

Ataque, normalmente de malware, que se aprovecha de una [vulnerabilidad de día cero](#).

### vulnerabilidad de día cero

Un defecto o una vulnerabilidad sin mitigación en un sistema de producción. Los agentes de amenazas pueden usar este tipo de vulnerabilidad para atacar el sistema. Los desarrolladores suelen darse cuenta de la vulnerabilidad a raíz del ataque.

### peticiones desde cero

Proporcionar a un [LLM](#) instrucciones para llevar a cabo una tarea, pero sin ejemplos (pasos) que puedan ayudar a guiarlo. El LLM debe usar los conocimientos del entrenamiento previo para

llevar a cabo la tarea. La eficacia de la petición desde cero depende de la complejidad de la tarea y de la calidad de la petición. Consulte también [peticiones con pocos pasos](#).

#### aplicación zombi

Aplicación que utiliza un promedio de CPU y memoria menor al 5 por ciento. En un proyecto de migración, es habitual retirar estas aplicaciones.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.