Engineer

Migración

traslado planificado a una versión o plataforma nueva

Qué es

Una migración es el traslado planificado de algo que ya se utiliza a una versión o plataforma nueva, realizado por pasos y con la retirada de la anterior al final. Puede trasladarse una carga de trabajo completa, una aplicación, un conjunto de datos o la estructura que los almacena. Lo que distingue una migración de una reconstrucción es que el sistema sigue utilizándose durante el traslado: sus usuarios, registros e integraciones lo acompañan, y la versión anterior continúa funcionando hasta que la nueva toma el relevo.

En ingeniería se usa la palabra a dos escalas, a menudo sin aclarar cuál.

A gran escala, se trata de pasar a otra plataforma o a una nueva versión principal. Cloud Adoption Framework de Microsoft divide el traslado de cargas de trabajo a Azure, desde servidores locales o desde otra nube, en cinco pasos: planificar, preparar las cargas, ejecutar, optimizar y retirar el origen. Los clientes de Esri trasladan sus mapas y datos de ArcMap a ArcGIS Pro. Una aplicación Next.js pasa de Pages Router a App Router, y `getServerSideProps` deja paso a la obtención de datos dentro de Server Components. Un clúster PostgreSQL pasa a una nueva versión principal, que a menudo cambia la disposición de las tablas del sistema; los datos se trasladan con `pg_upgrade` o mediante un volcado y una restauración.

A pequeña escala, una migración es un cambio versionado de la estructura de una base de datos, conservado como archivo en el repositorio y aplicado por una herramienta en un orden fijo. Django describe sus migraciones como un sistema de control de versiones para el esquema de la base de datos: cada archivo registra un cambio en los modelos, se incorpora al historial junto con el código y se ejecuta en la misma secuencia en el portátil de desarrollo, en el entorno de pruebas previo a producción y en producción. GitLab distribuye migraciones en segundo plano por lotes que reescriben tablas grandes, un lote cada vez,

después de una actualización; por ejemplo, ampliar columnas de enteros a `bigint` antes de que se desborden. También entra en esta categoría una migración de datos: cambia filas en lugar de columnas, como rellenar un campo nuevo a partir de dos antiguos.

Son una misma idea a dos escalas. Ambas tienen un estado inicial conocido, una secuencia ordenada de pasos, un destino y una forma de volver atrás si falla algún paso. El traslado de plataforma suele incluir migraciones de esquema; una larga sucesión de estas constituye, con los años, una plataforma que se transforma pieza a pieza mientras sus usuarios siguen utilizándola.

Utilice el concepto en una decisión

Aplique «¿Qué es una migración?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Trasladar los datos de un sistema envejecido a otro con soporte, manteniendo el negocio en funcionamiento durante el proceso.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Mapas GIS 10 veces más rápidos con PostgreSQL — 10x mapas más rápidos tras abandonar MSSQL

Mapas GIS 10 veces más rápidos con PostgreSQL Póngase en contacto →

Por qué importa

El software sobrevive a aquello sobre lo que se construyó. Se renegocian contratos de nube, los marcos de trabajo cambian su modelo y los proveedores retiran productos. Un sistema que se conserva una década se migrará varias veces durante ella: la capacidad del equipo para migrar acaba importando más que su acierto al elegir la primera plataforma.

- **El coste se concentra en el estado dividido.** Mientras una parte del sistema se ejecuta en la plataforma antigua y otra en la nueva, hay que operar, monitorizar y mantener coherentes ambas. Cloud Adoption Framework indica que se acorte ese periodo y que los componentes directamente dependientes entre sí se trasladen en la misma oleada.
- **Una base de datos no puede volver a desplegarse como el código.** El código puede reconstruirse y sustituirse; las filas de producción, no. Las migraciones de esquema convierten un cambio de la estructura en uso en un archivo revisable, comprobable contra una copia, aplicable en orden y, para la mayoría de los cambios, reversible. La alternativa es un esquema de producción que sea lo que dejó la última persona con acceso a la consola.

- **Un traslado planificado puede deshacerse.** Cloud Adoption Framework pide definir criterios y procedimientos de reversión antes de iniciar cualquier migración, probar la reversión y establecer puntos de decisión para continuar o detenerse durante el recorrido.

  Las contrapartidas se concentran en el cambio definitivo. Cloud Adoption Framework distingue entre migraciones con interrupción y con interrupción casi nula: en estas últimas, los datos se replican continuamente y el cambio se realiza al final. Para una versión principal de PostgreSQL, `pg_upgrade` acorta ese cambio: reutiliza los archivos de datos existentes y evita el volcado y la restauración que suele exigir una actualización de ese tipo. En modo `--link`, que no copia archivos, resulta aún mucho más rápido.

Cómo se aplica en la práctica

Tres proyectos de este sitio muestran el término a tres escalas, desde una plataforma completa hasta el orden de los archivos de un repositorio.

Una migración consistió en abandonar un proveedor de nube. La configuración AWS de un cliente tenía recursos sobredimensionados y el trabajo comenzó por revisar qué se utilizaba realmente: se eliminaron servicios innecesarios y se ajustó la capacidad restante. Después, las cargas de trabajo salieron de AWS. El registro del encargo respalda esos cambios de infraestructura. Eliminar servicios sin uso antes del traslado redujo lo que había que transportar; no demuestra una proporción de costes entre configuraciones equivalentes, un resultado de productividad ni una migración sin interrupciones.

La segunda fue un cambio de motor con un diseño nuevo como destino. La base de datos de una aplicación cartográfica pasó de MSSQL a PostgreSQL con la extensión PostGIS; el destino no era una copia de la estructura anterior. El nuevo esquema se diseñó para datos espaciales, con índices GiST y SP-GiST y restricciones estrictas de claves foráneas y de comprobación que aplican reglas de validación a las coordenadas espaciales. Más de 50 millones de registros espaciales llegaron mediante flujos de procesamiento ETL que los transformaron a EPSG:4326. Adoptar un diseño nuevo permite expresar las reglas de los datos en el propio esquema. La página informa de una mejora de 10 veces en el rendimiento de las consultas espaciales y de una reducción del 65 % en la carga de CPU del backend.

La tercera corresponde a la pequeña escala, mantenida durante años. El esquema de una plataforma es la suma de 1.022 migraciones, numeradas de 036 a 1102; la página califica de poco vistosa e innegociable la disciplina que mantiene su orden. El historial del repositorio describe la producción y solo sigue siendo cierto mientras cualquier base de datos, desde una copia nueva de desarrollo hasta una restauración, pueda reconstruirse reproduciéndolo desde el primer archivo hasta el último.

Dónde surgen los problemas

La planificación se centra en el cambio definitivo, pero los problemas suelen acumularse en las semanas que lo rodean, con cuatro formas reconocibles.

Dos sistemas conviven más tiempo del previsto. El traslado se atasca a medio camino, quedan las piezas difíciles y el equipo mantiene ambas plataformas durante meses. Los usuarios notan la separación: la guía de Next.js indica que pasar entre una página de Pages Router y otra de App Router implica una navegación completa y que `next/link` no precarga entre ambos. Mantenga pequeñas las oleadas y reduzca el coste de cada paso siguiente. Una interfaz de este portafolio va trasladando sus páginas a una factoría compartida que carga cada página en un solo viaje de ida y vuelta; así, cada página convertida hereda ese comportamiento.

Un cambio definitivo sin vuelta atrás. Algunos pasos alteran tanto el origen como el destino. Con `pg_upgrade --link`, una vez iniciado el clúster nuevo, este habrá escrito en archivos compartidos con el anterior, que entonces deberá restaurarse desde una copia de seguridad. Esri advierte de que un conjunto de datos actualizado para ArcGIS Pro puede dejar de ser utilizable en ArcMap e indica que se copie primero. En Django, una operación `RunSQL` sin SQL de reversión provoca `IrreversibleError` en cuanto alguien migra hacia atrás. Defina los criterios de reversión antes del primer paso, ensaye la vuelta y conviértala en una comprobación: un repositorio de este portafolio ejecuta un motor de controles cuyos validadores incluyen una comprobación de que una migración puede revertirse.

Se modifica una migración ya ejecutada. Las herramientas registran qué archivos se han aplicado y después los omiten. Si cambia un archivo que ya llegó a producción, esta conserva la versión antigua y cada base nueva recibe la nueva: dos esquemas bajo un solo historial. No modifique las migraciones aplicadas;

corríjalas con otra nueva. Un proceso de construcción que vuelve a aplicar el historial a una base de datos vacía y compara el resultado con una copia migrada detecta la divergencia.

La base de datos actualizada va lenta el primer día. `pg_upgrade` traslada la mayoría de las estadísticas del planificador, pero no todas; este hace estimaciones sobre el resto hasta que se reconstruyen. Su modo `--check` no cambia nada y puede ejecutarse mientras el servidor anterior sigue atendiendo solicitudes: úselo primero y reconstruya las estadísticas ausentes inmediatamente después de la actualización, antes de recopilar las restantes:

```
pg_upgrade --check -b "$OLD/bin" -B "$NEW/bin" -d "$OLD/data" -D "$NEW/data"
pg_upgrade -b "$OLD/bin" -B "$NEW/bin" -d "$OLD/data" -D "$NEW/data"
vacuumdb --all --analyze-in-stages --missing-stats-only
vacuumdb --all --analyze-only
```

Lecturas adicionales

- Planifique su migración — el método de Microsoft para trasladar cargas a Azure: oleadas, opciones de cambio definitivo y planificación de la reversión.
- Migraciones — la guía de Django sobre cambios de esquema versionados, su reversión y las migraciones de datos.
- pg_upgrade — la herramienta de actualización de PostgreSQL, con el modo de comprobación, el de enlaces, la vuelta atrás y los pasos posteriores para las estadísticas.
- Cómo migrar de Pages a App Router — trasladar una aplicación Next.js al nuevo enrutador, página por página.
- Migrar trabajo existente — qué se conserva de ArcMap a ArcGIS Pro, qué debe convertirse y de qué conviene hacer copia antes.
- Comprobar las migraciones antes de actualizar — cómo las migraciones en segundo plano de GitLab reescriben tablas grandes por lotes tras una actualización y cómo comprobar que han terminado.



Glosario

Elija según sus restricciones reales

Traiga un conjunto de datos representativo, las preguntas que debe responder y el cuello de botella actual. Podemos acordar cómo medir la calidad, la capacidad de procesamiento y la recuperación.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es una migración?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/migration/