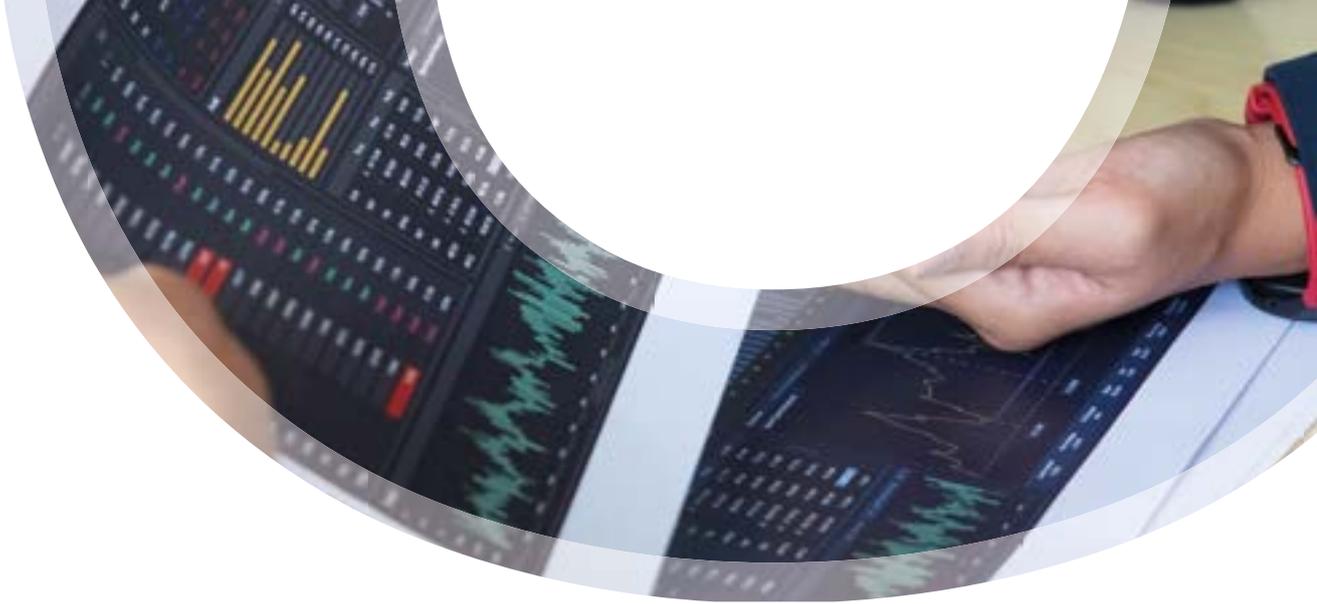




La estabilidad de tu eCommerce ante escenarios de pedidos punta o compras masivas

¿Estás seguro de que tu ecommerce está preparado a nivel de infraestructura para superar días como el Blackfriday o tras campañas de Marketing exitosas?



1/ Identificar los factores de capacidad limitantes de la plataforma y reacondicionar la capacidad provisionada si no se cumplen las expectativas

Es crítico tener identificado el **rendimiento de la plataforma** en términos tanto de usuarios concurrentes navegando por la tienda online como en volumen de pedidos, y saber cuál es el **límite de la plataforma** para poder diseñar **planes de actuación adecuados en caso de que se alcancen dichos límites** durante el periodo punta de pedidos. Para ello, es muy recomendable la realización de **pruebas de carga en entornos preproductivos** cada vez que se realizan modificaciones de gran calado en la aplicación.

Los principales casos de uso de cualquier tienda online suelen estar bastante bien definidos, y se recomienda **realizar simulaciones con herramientas de prueba con proporciones similares al tráfico real** hasta poner al límite la infraestructura. Un ejemplo de plan de simulación podría ser el siguiente:

Tipo de usuario	Usuarios concurrentes				
	T1	T2	T3	T4	T5
Navegación anónima	12	24	48	96	192
Compradores anónimos	1	2	4	8	16
Navegación de usuarios registrados	1	2	4	8	16
Compradores registrados	1	2	4	8	16
Concurrencia TOTAL	15	30	60	120	240



La idea es subir progresivamente la carga simulada hasta detectar el momento en que el servicio comienza a degradarse o incluso hasta que consigamos “tumbar” el sitio, y revisar la monitorización de cada uno de los sistemas en busca del **cuello de botella** que se haya alcanzado primero. Los principales cuellos de botella de sistemas de comercio electrónico suelen ser, en orden:

CPU del sistema de base de datos relacional.

Capacidad de IO del sistema de base de datos relacional

CPU de los servidores web y/o aplicaciones donde ejecuta el aplicativo

En el caso de los **sistemas de base de datos** la solución habitual suele ser planificar un aumento de capacidad, redimensionando verticalmente el nodo primario (y habitualmente el único) que da este servicio. Sin embargo, **la provisión de réplicas de lectura para repartir la carga** puede ayudar a paliar sensiblemente los problemas de rendimiento a la vez que habilita la posibilidad de tener un mecanismo de respaldo en caso de fallo catastrófico en el primario.

En el caso de los **servidores web**, lo recomendable es siempre realizar un escalado horizontal mediante la adición de más frontales siempre y cuando la arquitectura lo permita. La alternativa obvia es el **escalado vertical hasta poder garantizar en torno al 50% de capacidad extra** sobre las previsiones de visitas y pedidos.

En algunos sitios que implementan **Redis como motor de cache compartida en memoria** y tienen volúmenes grandes de tráfico suelen tener, adicionalmente, algunos de los siguientes problemas:

CPU del sistema de caché. Al ser Redis una **implementación monolítica**, cada nodo de Redis tiene como principal limitación la potencia de un Core singular del procesador donde ejecutan. Aunque Redis es un producto bien optimizado, a cierta escala es normal saturarlo en CPU.

Saturación del enlace de red de los servidores de Redis. Muchos aplicativos de comercio electrónico leen grandes cantidades de información de estos sistemas para evitar utilizar otros servicios más lentos como la base de datos. Llevado a escala, no es raro encontrarse con que los servidores de cache compartida están entregando datos a los servidores web al máximo de la capacidad de sus interfaces de red.

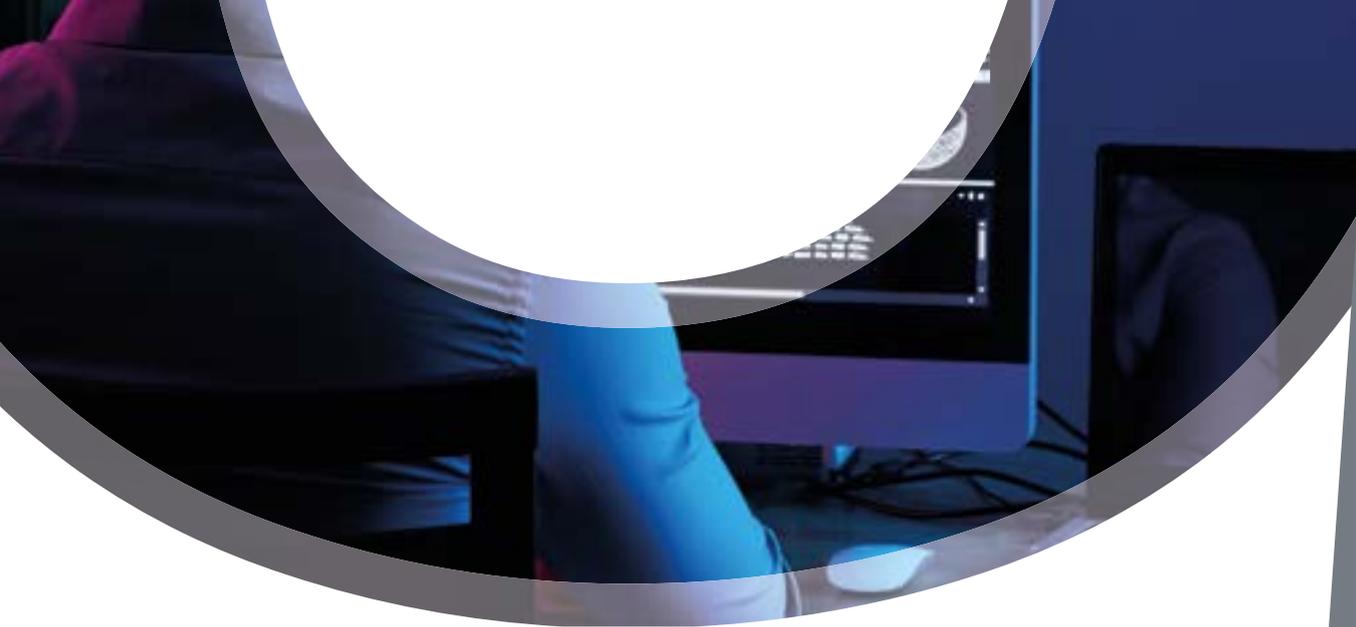
En estos casos lo recomendable siempre es realizar **sharding de datos y operaciones**, bien sea mediante despliegues de Redis maestro-esclavo repartiendo la carga de lectura sobre las distintas réplicas o bien **desplegando Redis en modo clúster**.

2/ Coordinación de modificaciones administrativas en el backend

Durante estos **periodos punta** en los que el sistema está sujeto a una carga muy por encima de la habitual es cuando suelen producirse sorpresas desagradables cada vez que un operador de catálogo hace una gestión pesada como por ejemplo una **actualización masiva de precios para una oferta flash**, o una **modificación cosmética de última hora para incluir un mensaje personalizado** en todas las páginas del catálogo.

Aunque este tipo de acciones pueden ser habituales, suele ocurrir que, en la eventualidad de ejecutarse cuando el sistema se encuentra sobrecargado por encima de los niveles habituales, se provocan **efectos “bola de nieve”** por la magnitud del tráfico recibido. Es normal que este tipo de modificaciones tengan como efecto secundario el vencimiento de datos en los sistemas de caché, provocando que el tráfico de usuarios sea, al menos durante las primeras visitas a cada página del catálogo, mucho más pesado de computar, provocando con ello **un pico de sobrecarga en los frontales y la base de datos**.

Es recomendable planificar y coordinar este tipo de modificaciones entre los **departamentos de e-commerce y TI** para tratar de **realizarlos en un periodo valle dentro de la campaña**, o incluso, si la arquitectura y el proveedor de infraestructura lo permiten, **provisionar temporalmente mayor capacidad en los puntos clave para soportar el pico**.



3/ Emisión escalonada de notificaciones push

Las **notificaciones push** son una de las herramientas favoritas del equipo de marketing por múltiples. Son prácticamente **instantáneas en la entrega, ofrecen buena visibilidad de volumetría de entrega y feedback, son sencillas de planificar**, etc. Sin embargo, su naturaleza puede provocar que muchos usuarios se sincronicen para acceder simultáneamente al sitio web sobrecargándolo y, eventualmente, degradando o incluso tumbando temporalmente el servicio.

Es recomendable que, cuando **la masa de usuarios a notificar sea grande**, estas notificaciones **se envíen de forma escalonada para prevenir la sincronización de usuarios** accediendo a la vez a la plataforma.

4/ Planes de recuperación ante desastres y alta disponibilidad

El **sistema de comercio electrónico supone el principal punto de entrada de clientes** para un gran número de empresas. Es por ello que se recomienda encarecidamente dotar de la mayor disponibilidad posible a la arquitectura que da soporte a la plataforma, así como **tener bien planificados los pasos a realizar para tratar de recuperar el servicio** lo antes posible en caso de fallo.

Para ello, existen numerosos **patrones de diseño de arquitectura** que pueden ayudar en este aspecto:

- Servicios de balanceo de carga y disposición de múltiples frontales web.
- Réplicas de lectura en los servicios de datos
- Servicio de DNS con soporte de conmutación por error para poder redirigir a los usuarios a páginas de mantenimiento estáticas automáticamente en caso de fallo en el sitio principal.

5/ Maximizar el uso de servicios gestionados de proveedores de cloud computing

La gran mayoría de **proveedores de infraestructura en la nube** proporcionan servicios gestionados para resolver rápidamente problemas de capacidad y disponibilidad. Es muy recomendable prescindir en la medida de lo posible de servidores administrados en favor de estas soluciones ya que están diseñadas para resolver casos de uso muy habituales y su funcionamiento está bien probado. Algunos de los casos de uso habituales son:

Auto-escalado de frontales web

Balanceo de carga integrado con la mecánica de auto-escalado

Bases de datos relacionales con soporte para conmutación por error automática o incluso auto-escalado en ciertos casos.

Servicios de cache en memoria con soporte para conmutación por error automática

Motores de búsqueda

Servidores de ficheros

Todos estos servicios suelen proporcionar **su propia monitorización y mejoras constantes**, lo que no hace sino dotarles de **mayor valor respecto de una infraestructura administrada**.

6/ Desacoplar el panel de administración y tareas programadas en un servidor independiente

Es altamente recomendable que los procesos por lotes programados y los usuarios administrativos que manipulan el **backend del sitio web** lo hagan sobre una infraestructura dedicada y aislada de la que da servicio a los usuarios (aunque inevitablemente compartiendo los servicios de datos). De esta forma se previenen posibles **problemas de congestión en el servidor web** derivados de acciones pesadas por parte de un administrador.

Este **patrón de arquitectura** habilita adicionalmente la posibilidad de **restringir el acceso a nivel de red desde ciertas IPs** o incluso restringirlo para requerir **servicios de conectividad VPN empresarial**, dotando de un mayor grado de seguridad a la plataforma.

En el caso de que **la integración con el ERP empresarial** se realice mediante **el consumo de APIs REST**, se recomienda también delegarlas sobre esta infraestructura dedicada.

¿Te interesa?

Contacta con nosotros