Engineer

macOS

Sistema operativo de escritorio de Apple

Qué es

macOS es el sistema operativo que Apple entrega con cada Mac: el escritorio y el Finder, las aplicaciones incluidas y, debajo, un sistema Unix con terminal, intérprete de órdenes y las herramientas habituales de línea de comandos. Apple lo desarrolla para su propio hardware y vende ambos juntos, por lo que elegir macOS significa, en la práctica, elegir un Mac.

Su base es Darwin, cuyo código publica Apple. Su núcleo, XNU, es híbrido: combina el núcleo Mach de la Universidad Carnegie Mellon con código de FreeBSD y con IOKit, un marco de C++ para controladores; su nombre significa «X is Not Unix», es decir, «X no es Unix». Gran parte del entorno de usuario también procede de FreeBSD, y los manuales de `sed` y `date` en un Mac describen las versiones de FreeBSD. Desde macOS 10.15 Catalina, el intérprete predeterminado es zsh.

El hardware ha cambiado recientemente bajo ese sistema. A finales de 2020, Apple comenzó a sustituir los procesadores Intel del Mac por sus propios chips, Apple silicon; todas las líneas, desde el MacBook Air hasta el Mac Pro, tienen ya modelos con ellos. Son chips arm64, algo relevante en cuanto el software construido en un Mac debe ejecutarse en un servidor x86-64.

En portátiles y equipos de escritorio, macOS se compara con Windows y Linux.

Utilice el concepto en una decisión

Aplique «¿Qué es macOS?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Resolver con rapidez y lenguaje claro los problemas del portátil, la impresora, el acceso o el correo, sin hacer sentir mal a nadie.

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. Creación de un departamento de soporte técnico — 10,000 clientes atendidos por un departamento de soporte creado desde cero

Creación de un departamento de soporte técnico Póngase en contacto →

Por qué importa

Para una empresa que desarrolla software, macOS cumple dos funciones fáciles de confundir: es la máquina donde trabajan muchos desarrolladores y una plataforma donde los clientes ejecutan software.

Como estación de trabajo, un Mac difiere de los servidores Linux donde se despliega el software en tres aspectos. Sus herramientas de línea de comandos proceden de FreeBSD, en lugar de GNU, por lo que una misma opción puede significar cosas distintas en un script. Su procesador es arm64, mientras que el servidor puede ser x86-64. Además, no tiene núcleo Linux: como los contenedores Linux comparten el núcleo del anfitrión, las herramientas que los ejecutan en un Mac arrancan primero una máquina virtual Linux; la herramienta `container` de Apple, que requiere Apple silicon y macOS 26, proporciona a cada contenedor una máquina virtual ligera propia. Nada de ello descarta usar un Mac: exige probar en el sistema de destino en lugar de confiar en una ejecución correcta en el portátil.

Como plataforma, macOS plantea cómo desarrollar para ella. Una envoltura multiplataforma, con una sola base de código y un motor de navegador integrado, distribuida para Windows, Linux y Mac, permite mantener varios sistemas al día a bajo coste; pero en el Mac incluye un entorno de ejecución de navegador que una aplicación nativa no necesita y dibuja sus propios controles en lugar de usar los del sistema. Una aplicación nativa en Swift y SwiftUI utiliza los controles y materiales del sistema y respeta los ajustes de apariencia, color de acento, contraste, tamaño del texto y movimiento reducido del usuario sin código adicional. El precio es un cliente separado y personal que conozca la plataforma. La opción nativa compensa cuando los clientes son usuarios de Mac.

La distribución para Mac también pasa por un control. El software distribuido fuera de Mac App Store se firma con un certificado Developer ID y se envía al servicio de notarización de Apple: un análisis automatizado de contenido malicioso y problemas de firma, separado de App Review. Desde macOS 10.15, todo software construido después del 1 de junio de 2019 y distribuido con Developer ID debe estar notarizado. Apple devuelve un ticket que se adjunta al software y publica el mismo ticket en línea, donde Gatekeeper lo busca cuando el

usuario abre la aplicación por primera vez. El certificado forma parte de Apple Developer Program, que cuesta $99 al año. Incluya la afiliación y la firma en el plan de la primera versión.

Cómo se aplica en la práctica

La experiencia de escritorio de un producto de mensajería en macOS era una envoltura multiplataforma que no parecía ni se comportaba como una aplicación Mac. Su núcleo Rust, bien probado, expone una interfaz C y Swift puede llamar a C, de modo que se construyó un cliente nativo en Swift y SwiftUI directamente sobre esa interfaz. El núcleo se enlaza como archivo estático desde una revisión fijada del proyecto de origen, manteniendo reproducible la compilación; una capa fina gestiona la propiedad de cada puntero que cruza entre lenguajes. El cliente arranca como aplicación Mac, utiliza los materiales y controles de la plataforma y no incorpora navegador. La página reconoce la carencia: no hay distribución notarizada ni canal de actualizaciones; es una aplicación funcional, no un producto que otras personas ejecuten. La notarización anterior es el paso que separa ambas situaciones.

La apariencia de la plataforma puede imponerse mediante reglas, en lugar de imitarse. Un sistema de diseño con 48 tokens con nombre, repartidos en seis grupos, construye el color y el material sobre los colores semánticos y el material de cristal de la plataforma, en lugar de valores fijos. Así, la interfaz sigue la apariencia del sistema, el color de acento y el contraste sin código que vigile sus cambios, y sus estilos de texto se escalan con la preferencia de tamaño del usuario. Un analizador rechaza un número literal donde corresponde un token, un tamaño de fuente fijado en el código y una animación que ignore la preferencia de reducir movimiento. La página también precisa su límite: los tokens hacen coherente una interfaz, algo distinto de hacerla buena.

La memoria comprimida y el intercambio permiten que una aplicación macOS crezca muy por encima de la memoria física. El registro de un incidente documenta informes de diagnóstico con 111 GB de páginas comprimidas en una máquina de 36 GB el 2026-09-01, y una muestra separada del consumo de memoria que mostraba un crecimiento de unos 41 MB por segundo durante un bucle de reintentos de conexión. La ruta de eventos retenía los eventos del log del núcleo de la aplicación en un flujo principal sin límite y los copiaba a suscriptores que no los usaban. Se limitaron los búferes, se filtraron las suscripciones por tipo de evento y se asignó un límite de frecuencia al registro. Una comparación de ejecuciones completas de pruebas registró una caída de 610.996 líneas del log del núcleo de la aplicación a 1.411: son mediciones de

pruebas, no una cifra de memoria medida después de la corrección ni una prueba comparativa de producción. La contrapartida es explícita: pueden descartarse eventos de diagnóstico repetitivos y actualizarse el estado afectado en lugar de retener eventos sin límite. Otros tiempos de espera durante el cierre necesitaron una corrección separada.

Dónde surgen los problemas

Activity Monitor muestra que apenas queda memoria en un Mac que funciona perfectamente. macOS guarda en caché los archivos usados recientemente en la memoria que quedaría libre; cuando se llena, comprime las aplicaciones inactivas para dejar espacio a las activas. Por eso es normal que Memory Used (memoria usada) se acerque a Physical Memory (memoria física) y que la cifra de Compressed (memoria comprimida) sea alta. La propia guía de Apple indica que la memoria libre no hace necesariamente más rápido un Mac. Consulte el gráfico Memory Pressure (presión de memoria): el verde significa que la RAM se utiliza eficientemente; el amarillo, que el Mac podría acabar necesitando más; el rojo, que necesita más. El intercambio es la señal más clara:

```
sysctl vm.swapusage
vm_stat | grep -E '^Swap(ins|outs)'
```

Los contadores de `vm_stat` acumulan desde el arranque: compare dos lecturas separadas por un minuto; si las salidas de páginas al área de intercambio siguen aumentando, el trabajo ha superado la capacidad de la RAM.

Un script funciona en el Mac y se detiene con un error extraño en el servidor Linux, o al revés. El `sed` del Mac, derivado de FreeBSD, interpreta la palabra posterior a `-i` como sufijo de copia de seguridad; por eso, la costumbre GNU `sed -i 's/a/b/' file` toma la expresión como sufijo y el nombre del archivo como script. La costumbre Mac `sed -i '' …` hace que GNU `sed` interprete la expresión como nombre de archivo. Un sufijo unido a la opción funciona en ambos:

```
sed -i.bak 's/^port=80$/port=8080/' app.conf && rm app.conf.bak
```

La solución duradera es ejecutar los scripts en CI sobre el sistema que encontrarán en producción.

Gatekeeper bloquea una compilación compartida fuera del equipo cuando el destinatario la abre. La aplicación se firmó para desarrollo, o se firmó sin notarizarse. Fírmela con un certificado Developer ID, con el entorno de ejecución reforzado habilitado y una marca de tiempo segura; envíela al servicio de notarización y adjunte el ticket para que Gatekeeper lo encuentre incluso sin conexión. Un archivo ZIP puede notarizarse, pero no admite un ticket adjunto: adjúntelo a la aplicación que contiene:

```
ditto -c -k --keepParent App.app App.zip
xcrun notarytool submit App.zip --keychain-profile notary --wait
xcrun stapler staple App.app
```

Una imagen de contenedor funciona en el Mac del desarrollador y falla en el servidor con `exec format error`. Docker crea imágenes de forma predeterminada para la plataforma del demonio BuildKit; en un Mac con Apple silicon, ese demonio se ejecuta en una máquina virtual arm64. Un contenedor comparte el núcleo del anfitrión, por lo que, sin emulación, un anfitrión x86-64 solo ejecuta código x86-64. Compruebe el servidor con `uname -m` y construya para él con `docker build --platform linux/amd64`, o cree la imagen en CI sobre una máquina con el mismo tipo de CPU que el servidor.

Lecturas adicionales

- Guía del usuario de Mac — la guía de macOS de Apple, mantenida para cada versión vigente.
- Comprobar si el Mac necesita más RAM — el gráfico Memory Pressure y el significado de sus tres colores.
- Notarizar software macOS antes de distribuirlo — Developer ID, el entorno de ejecución reforzado y cómo utiliza Gatekeeper el ticket.
- Creación de imágenes multiplataforma — por qué los anfitriones arm64 y amd64 necesitan imágenes distintas y cómo un solo proceso de creación genera ambas.
- container — la herramienta de Apple para ejecutar contenedores Linux en un Mac, cada uno en una máquina virtual ligera.
- XNU — el código del núcleo de macOS, con una breve explicación de la integración de Mach, FreeBSD e IOKit.

Glosario

Elija según sus restricciones reales

Describa el dispositivo o servicio, los síntomas observados y los datos que deben conservarse. Podemos establecer el paso de diagnóstico útil más seguro.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es macOS?» 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/macos/