**Engineer**

# Puerto

Número que selecciona un servicio en un host

## Qué es

Un puerto es un número de 16 bits en la cabecera de cada segmento TCP y UDP que indica a la máquina receptora a cuál de sus programas pertenecen los datos. La dirección lleva un paquete al host correcto; el puerto lo lleva al proceso correcto una vez allí. Los ingenieros dicen número de puerto cuando quieren ser precisos, y puertos en plural suele referirse al conjunto en el que escucha un servidor. El término no guarda relación con un puerto físico de un conmutador ni con la adaptación de software a otra plataforma.

Como el campo tiene 16 bits, hay 65.536 valores, del 0 al 65535, y TCP y UDP tienen cada uno su propio conjunto: el puerto TCP 53 y el puerto UDP 53 son dos elementos distintos, y DNS utiliza ambos. La especificación TCP, RFC 9293, define un socket como una dirección de Internet unida a un puerto, y una conexión como un par de sockets. Eso da cuatro valores: dirección del cliente, puerto del cliente, dirección del servidor y puerto del servidor. También explica por qué un puerto de servidor puede mantener miles de conversaciones simultáneas. Cada conversación difiere en al menos uno de los otros tres valores, normalmente el puerto del cliente.

RFC 6335 divide el intervalo en tres. Los puertos 0–1023 son puertos de sistema, los 1024–49151 son puertos de usuario y los 49152–65535 son puertos dinámicos, que IANA nunca asigna. IANA mantiene el registro que vincula un nombre de servicio con un número: 22 para SSH, 53 para DNS, 443 para HTTPS y 5432 para PostgreSQL. El registro es una convención en la que confían los clientes, y cualquier programa puede escuchar en cualquier número libre; así acaba SSH en un puerto que nadie adivinaría.

El lado cliente de una conexión suele recibir un puerto que nunca eligió. Cuando un programa abre una conexión saliente, el sistema operativo escoge uno de su intervalo efímero, y Linux usa de forma predeterminada 32768–60999 en lugar del intervalo dinámico de IANA. Solicitar la vinculación al puerto 0 significa lo mismo: asígneme cualquiera que esté libre.

Utilice el concepto en una decisión

Aplique «¿Qué es un puerto?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Wi-Fi, cortafuegos y acceso remoto configurados para trabajar desde cualquier lugar y permitir el acceso solo a las personas autorizadas.

**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. Administración de 30 VPS Ubuntu Linux — 30 servidores Linux administrados con recuperación planificada

**Administración de 30 VPS Ubuntu Linux** Póngase en contacto →

## Por qué importa

La decisión detrás de cada puerto es la exposición: qué programas responden desde fuera y en qué dirección. Un puerto sin proceso a la escucha provoca un rechazo. Un puerto con un proceso a la escucha es un programa con el que puede comunicarse cualquiera que alcance esa dirección, errores incluidos. El servicio más barato de proteger es uno que escucha en 127.0.0.1, porque los paquetes de otras máquinas no pueden alcanzar la dirección de bucle local, diga lo que diga el cortafuegos.

Por tanto, la primera pregunta para cada servicio es si realmente necesita ser accesible desde fuera. Una base de datos con la que solo habla la aplicación de la misma máquina debe quedar en el bucle local. Una herramienta usada únicamente por administradores puede permanecer en él y ser accesible a través de un host de salto o una VPN. Lo que queda, normalmente 80 y 443 para un sitio web y un puerto para SSH, es la superficie pública. Mantenerla pequeña es una decisión de coste tanto como de seguridad: cada servicio que escucha públicamente necesita parches, registros y una regla de cortafuegos que le corresponda.

La segunda pregunta es el número. Los puertos estándar no exigen configuración adicional: navegadores, clientes y redes restrictivas esperan el 443 y lo dejan pasar. Mover SSH del 22 reduce el ruido de los escáneres que solo prueban el predeterminado y facilita leer los registros, pero un análisis de todo el

intervalo lo encuentra de todas formas. Además, cada herramienta que hace referencia al puerto debe conocer el número nuevo. Esto último es lo que suelen olvidar los equipos.

La tercera cuestión son los privilegios y el uso compartido. En Linux, un proceso necesita root o la capacidad CAP_NET_BIND_SERVICE para vincularse a un puerto inferior a 1024, salvo que ip_unprivileged_port_start se haya reducido desde su valor predeterminado de 1024. Normalmente, un programa ocupa una dirección y un puerto concretos, de modo que dos servicios que quieren el 443 necesitan algo delante que reparta el tráfico, por lo general un proxy inverso que lea el nombre de host solicitado por el cliente.

## Cómo se aplica en la práctica

Cuando la empresa instaló su propia forja de Git en julio de 2026, el servicio SSH de la forja quedó vinculado al bucle local en el puerto 23231, y cualquiera que accede desde fuera lo hace a través de un host de salto, por lo que el número de puertos públicos no cambió. Está escrito, pero desactivado, un multiplexor que serviría HTTPS y Git por SSH desde un mismo puerto público, examinando los primeros bytes de cada conexión. Un servicio a la escucha sin usuarios también tendría que defenderse.

En agosto de 2026, una revisión de endurecimiento descubrió que SSH había pasado al puerto 1986 durante el primer endurecimiento del host, mientras que el mecanismo de bloqueo de intrusiones seguía vigilando el 22. Cada bloqueo del mecanismo afectaba a un puerto sin ningún proceso a la escucha, de modo que la protección nunca se había aplicado. La corrección fue pequeña. Lo que permaneció fue una regla: cada control de seguridad va acompañado de una aserción que lo pone a prueba.

El host tiene 464 MB de memoria y ningún panel, de modo que las preguntas sobre él las responden siete roles de informes de solo lectura que describen en Markdown la máquina en funcionamiento. El informe de datos del sistema enumera los sockets a la escucha: el inventario de puertos que mantiene realmente el kernel, en lugar del descrito por la configuración. Entre los hallazgos reales de los informes había un puerto a la escucha que ninguna declaración mencionaba.

# Dónde surgen los problemas

**Un servicio destinado a la propia máquina responde a todo Internet.** Muchos programas escuchan de forma predeterminada en todas las direcciones, que aparecen como `0.0.0.0`, `[::]` o `*` en la lista de sockets. Una base de datos o un panel de administración destinado al uso local queda entonces a un solo error de cortafuegos de estar expuesto. Establezca explícitamente la dirección de escucha en `127.0.0.1` o `::1` y confírmela en la columna de dirección local:

```
ss -tlnp
ss -ulnp
```

**Una protección sigue vigilando un puerto que el servicio abandonó.** El servicio pasó a un número nuevo y la regla de cortafuegos, el mecanismo de bloqueo, la sonda de monitorización o la configuración de un cliente sigue indicando el antiguo. Todas esas herramientas siguen funcionando correctamente sobre un puerto donde no escucha ningún demonio. Mantenga el número en una sola variable que lean todos sus consumidores y añada una comprobación que compare el socket a la escucha con lo que vigila cada control.

**Las conexiones salientes empiezan a fallar bajo carga con «Cannot assign requested address».** Cada conexión saliente toma un puerto local del intervalo efímero, y una conexión cerrada mantiene reservados sus cuatro valores en TIME_WAIT durante un tiempo posterior. Con el intervalo predeterminado de Linux de 32768–60999, una dirección de cliente puede mantener 28.232 conexiones con la misma dirección y puerto de servidor, abiertas o cerradas recientemente. Un proxy o proceso de trabajo que abre una conexión nueva por petición a un mismo backend las agota. Reutilice primero las conexiones mediante keep-alive o un pool; ampliar el intervalo o añadir direcciones de origen solo da más margen.

**Un servicio iniciado como usuario normal se detiene con «Permission denied» en el puerto 80.** Vincularse por debajo de 1024 requiere privilegios. Conceda al servicio la capacidad CAP_NET_BIND_SERVICE mediante su archivo de unidad, colóquelo en un puerto alto detrás de un proxy inverso o, en un host donde todos los usuarios locales sean de confianza, reduzca ip_unprivileged_port_start. Ejecutar todo el servicio como root para conseguir un puerto es la peor alternativa.

**El arranque se detiene con «Address already in use».** Otro proceso ocupa el puerto, a menudo una copia antigua del mismo servicio que aún se está cerrando o un segundo servicio con el mismo valor predeterminado —8080 es habitual—. `ss -tlnp` identifica al propietario. Decida qué servicio ocupa el número y asigne a cada uno un puerto explícito en su configuración en lugar de depender de los valores predeterminados.

## Lecturas adicionales

- RFC 9293, sección 4: glosario — Definiciones de puerto, socket y conexión de la propia especificación TCP.
- RFC 6335: procedimientos de IANA para gestionar el registro de nombres de servicio y números de puerto de protocolos de transporte — Los tres intervalos de puertos y cómo se asignan sus números.
- Registro de nombres de servicio y números de puerto de protocolos de transporte — Lista de IANA que indica qué servicio está registrado en cada número.
- RFC 768: protocolo de datagramas de usuario — La especificación UDP de tres páginas, incluidos los puertos de origen y destino.
- RFC 6056: recomendaciones para aleatorizar puertos de protocolos de transporte — Cómo debería elegir un cliente sus puertos efímeros para que resulten difíciles de adivinar.
- IP Sysctl — Ajustes del kernel de Linux, entre ellos ip_local_port_range e ip_unprivileged_port_start.

Glosario

Elija según sus restricciones reales

Describa la ruta de despliegue, los riesgos operativos y quiénes se encargan de las guardias. Podemos definir el alcance de un cambio con comprobaciones, una vía de recuperación y responsabilidades claras.

**Vea cuánto cuesta** Póngase en contacto →

# Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es un puerto?» 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/port/