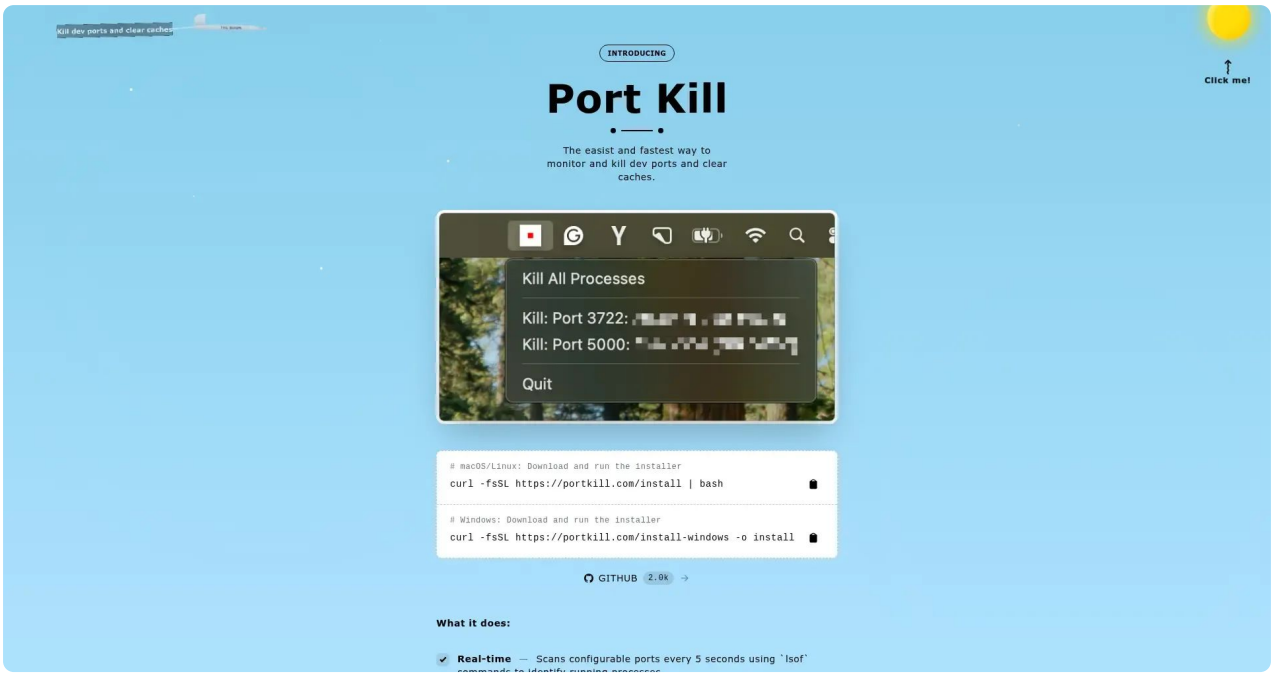

Port Kill

Port Kill es una herramienta CLI y de barra de estado diseñada para desarrolladores que necesitan gestionar procesos que bloquean puertos de red y limpiar cachés de desarrollo. Ideal para profesionales backend, frontend y DevOps que enfrentan errores EADDRINUSE, permite identificar, terminar y reiniciar procesos automáticamente mediante Smart Restart. Además, orquesta servicios mediante archivos YAML, limpia cachés de lenguajes como Rust o JS y ofrece integración con servidores MCP para asistentes de IA.

Visitar Sitio Oficial Preguntar a ChatGPT Preguntar a Claude Preguntar a Grok

Contenido del Dossier

INFORMACIÓN DE LA HERRAMIENTA

Qué y para quién es
Port Kill es una herramienta de línea de comandos (CLI) y de barra de estado diseñada para desarrolladores que necesitan gestionar procesos que bloquean puertos de red y limpiar cachés de desarrollo de forma rápida. Está pensada para profesionales que trabajan en entornos locales o remotos (SSH) y que se enfrentan frecuentemente a errores de tipo "EADDRINUSE" o conflictos de puertos al lanzar microservicios o servidores web.

Principal ventaja profesional
A diferencia de comandos manuales como lsof o netstat, Port Kill no solo identifica y termina procesos, sino que implementa **Smart Restart**: recuerda el comando original que inició el proceso y permite reiniciarlo automáticamente con una sola instrucción, manteniendo el flujo de trabajo sin interrupciones.

Para quién no es
No está dirigida a administradores de sistemas que buscan una herramienta de monitorización de red empresarial a gran escala o seguridad perimetral. Tampoco es ideal para usuarios que no se sientan cómodos con la terminal (CLI), a menos que utilicen exclusivamente la versión con interfaz básica para macOS.

Funcionalidades clave
- **Terminación Inteligente:** Estrategia de cierre progresivo (SIGTERM seguido de SIGKILL) para liberar puertos de forma segura.
- **Smart Restart:** Guarda y ejecuta el comando de inicio original de un proceso tras haberlo matado.
- **Gestión de Cachés:** Detecta y limpia cachés de construcción de lenguajes (Rust, JS, Python, Java) y plataformas (Vercel, Cloudflare, Hugging Face).
- **Modo Guardián:** Monitoriza puertos específicos y puede autorreiniciar procesos si estos fallan o se detienen.
- **Orquestación de Servicios:** Permite definir un archivo .port-kill.yaml para levantar múltiples servicios con dependencias y retardos de inicio.
- **Detección Automática:** Escanea el proyecto para identificar scripts de npm, servicios de Docker Compose o aplicaciones Python.

Precios
- **Versión Gratuita:** La herramienta es Open Source y totalmente gratuita para su uso profesional y personal. No existen planes de suscripción ni limitaciones de funcionalidades por pago.

Perfil del usuario
- **Desarrolladores Full-stack y Backend:** Gestión de múltiples microservicios simultáneos.
- **Ingenieros de DevOps:** Limpieza de entornos de CI/CD y gestión de servidores remotos vía SSH.
- **Desarrolladores Frontend:** Resolución de conflictos en puertos habituales (3000, 8080) y limpieza de carpetas node_modules o cachés de Next.js/Vite.

Nivel técnico requerido
- **Uso:** Bajo-Medio (uso de comandos básicos en terminal).
- **Instalación:** Bajo (script de autoinstalación de una línea).
- **Configuración:** Medio (solo si se desea usar la orquestación mediante archivos YAML).

Ejemplos de uso profesional
- **Conflictos de Puerto:** Liberar instantáneamente el puerto 3000 ocupado por un proceso zombie de una sesión anterior.
- **Reinicio tras Crash:** Usar el modo --guard para que un servidor de desarrollo se reinicie automáticamente si encuentra un error crítico.
- **Limpieza de Disco:** Ejecutar port-kill cache --clean para recuperar gigabytes de espacio ocupados por cachés antiguas de librerías y dependencias.
- **Despliegue Rápido:** Levantar todo un stack tecnológico (Base de datos + API + Web) respetando el orden de encendido con un solo comando port-kill --up.

Uso y distribución
- **Versión Escritorio:** Aplicación para la barra de estado (exclusivo para macOS).

- **CLI:** Binario ejecutable para Windows, Linux y macOS.
- **SSH:** Soporta ejecución remota para gestionar puertos en servidores externos.

Open source
El proyecto es de código abierto, desarrollado principalmente en Rust, lo que garantiza un consumo mínimo de recursos y alta velocidad de ejecución.

Integraciones
- **MCP (Model Context Protocol):** Dispone de un servidor MCP para que herramientas de IA como Cursor o Claude puedan gestionar los puertos y cachés del sistema directamente.
- **Ecosistema JS:** Integración nativa para detectar y limpiar cachés de npm, yarn, pnpm y npx.
- **Cloud:** Soporta limpieza de cachés para Vercel y Cloudflare (requiere tokens de API).

Notas finales
Información legal, licencias y contratos
- Port Kill se distribuye bajo una licencia de código abierto (repositorio gestionado por Treadie HQ). El uso es libre y los binarios se instalan localmente sin recolección de datos sensibles, siendo una herramienta de utilidad técnica sin contratos de permanencia ni términos complejos de propiedad intelectual para el usuario final.

Para más información:
- Sitio web oficial
- Github

CONSEJOS DE IMPLANTACIÓN

Aplicación profesional
Port Kill se dirige a empresas de desarrollo de software, agencias digitales y departamentos de IT que operan con arquitecturas de microservicios o entornos de desarrollo intensivos.
- **Tipos de empresa:** Startups tecnológicas, factorías de software y equipos DevOps que gestionan múltiples stacks simultáneamente (Node.js, Python, Docker).
- **Presupuesto:** Herramienta Open Source sin coste de licencia. El ahorro se cuantifica en la reducción de tiempos muertos por resolución de conflictos de puertos (errores EADDRINUSE).
- **Puntos clave:** Centralización de la gestión de puertos, orquestación de servicios mediante YAML y limpieza automatizada de cachés de lenguajes (Rust, JS, Java, Python).

Madurez digital requerida
- **Usuarios:** Desarrolladores y perfiles técnicos habituados al uso de la terminal (CLI) y edición de archivos de configuración (.yaml).
- **Empresa:** Equipos que ya utilicen herramientas de contenedores (Docker) o múltiples servicios locales y busquen optimizar su flujo de trabajo (workflow) diario.

Plan orientativo de implantación
Pasos necesarios y estimaciones
- **Evaluación inicial (1 día):** Identificar los puertos y servicios críticos que suelen generar bloqueos en el equipo de desarrollo.
- **Configuración y Piloto (1-3 días):** Crear un archivo .port-kill.yaml centralizado para los proyectos principales, definiendo dependencias y órdenes de arranque.
- **Implantación (Inmediata):** Instalación mediante el script oficial de una sola línea en las máquinas de los desarrolladores.
- **Capacitación (1 sesión):** Formación breve sobre el uso de Smart Restart y el Modo Guardián para mantener servicios críticos siempre activos.

Necesidades de formación del equipo
El equipo debe conocer los comandos básicos de la CLI y cómo interpretar los resultados del comando --list. Es fundamental entender el concepto de Smart Restart para evitar lanzamientos duplicados de servicios.

Perfiles necesarios
- **Perfil técnico interno:** Un desarrollador senior o Lead Developer para estandarizar el archivo de configuración .port-kill.yaml del proyecto.
- **Personal externo:** No se requiere.
- **Otros:** Integración con herramientas de IA (Cursor o Claude) mediante su servidor **MCP (Model Context Protocol)** para que la IA gestione puertos automáticamente.

Retorno de la inversión (ROI)
- **Tiempos:** Recuperación inmediata de productividad tras fallos de procesos "zombie" que antes requerían reinicios manuales o búsqueda de PIDs mediante comandos complejos (lsof -i).
- **Cómo medirlo:** KPI de "Tiempo medio de recuperación ante conflictos de puerto" y reducción de tickets de soporte interno relacionados con entornos locales bloqueados.

Otros aspectos de interés
- **Seguridad:** Existe una vulnerabilidad reportada (GHSA-4fmr-m2w5-f73j) en el servidor MCP para versiones antiguas; se recomienda encarecidamente usar la última versión estable (v0.5.41 o superior) y evitar el uso de entradas no saneadas en el parámetro --remote.
- **Compatibilidad SSH:** Permite gestionar puertos en servidores remotos con la misma sintaxis que en local, ideal para depuración en entornos de staging.
- **Gestión de Caches:** No solo mata procesos; su capacidad para limpiar inteligentemente cachés de node_modules, target/ de Rust o .venv de Python lo convierte en una herramienta de mantenimiento de sistema.

PREGUNTAS FRECUENTES

¿Qué es Port Kill y cuál es su función principal?

Port Kill es una herramienta de interfaz de línea de comandos (CLI) y de sistema desarrollada en Rust, diseñada específicamente para identificar y finalizar procesos que bloquean puertos de red de forma indebida. Su función principal es resolver errores de tipo 'EADDRINUSE' y gestionar el ciclo de vida de los procesos de desarrollo, permitiendo liberar recursos del sistema de manera eficiente.

¿Cómo funciona la característica de Smart Restart?

Smart Restart es una funcionalidad avanzada que permite a la herramienta recordar el comando original utilizado para iniciar un proceso específico. Al ejecutar la terminación del proceso, Port Kill ofrece la capacidad de reiniciarlo automáticamente con una sola instrucción, eliminando la necesidad de volver a escribir manualmente comandos complejos en la terminal.

¿Es Port Kill una herramienta gratuita o de pago?

La herramienta es de código abierto (Open Source) y totalmente gratuita para su uso profesional y personal. No cuenta con planes de suscripción, limitaciones de funcionalidades tras muros de pago ni costes ocultos, distribuyéndose bajo un modelo de software libre.

¿Qué tecnologías y plataformas son compatibles con Port Kill?

Es compatible con los sistemas operativos Windows, Linux y macOS. Además de su versión CLI, ofrece una aplicación de barra de estado específica para macOS. En términos de lenguajes y entornos, cuenta con integraciones nativas para identificar y gestionar servicios de Node.js (npm, yarn, pnpm), Python, Rust, Java, y plataformas como Docker Compose, Vercel y Cloudflare.

¿Cómo aborda Port Kill la seguridad y la privacidad de los datos?

La herramienta se ejecuta de forma local como un binario ejecutable y no realiza recolección de datos sensibles del usuario. Al ser un proyecto de código abierto alojado en GitHub, su funcionamiento es transparente y verificable por la comunidad profesional. No requiere contratos de permanencia ni términos de propiedad intelectual que comprometan al usuario final.

¿Qué diferencia a Port Kill de comandos tradicionales como lsof o netstat?

A diferencia de las herramientas estándar del sistema que solo listan procesos o conexiones, Port Kill automatiza la terminación progresiva (usando señales SIGTERM y SIGKILL) y añade capas de valor como la gestión de cachés de construcción, orquestación de servicios mediante archivos YAML y un modo de monitorización activa (Modo Guardián).

¿Es posible utilizar Port Kill en entornos de servidor remoto?

Sí, Port Kill está diseñado para ser compatible con flujos de trabajo a través de SSH. Esto permite a los ingenieros de DevOps y administradores gestionar conflictos de puertos y limpiar entornos de integración continua (CI/CD) en servidores remotos con la misma facilidad que en local.

¿Qué es la integración con MCP y para qué sirve?

Port Kill dispone de un servidor compatible con el Model Context Protocol (MCP). Esto permite que herramientas de desarrollo basadas en Inteligencia Artificial, como Cursor o Claude, puedan interactuar directamente con el sistema para gestionar puertos y cachés, facilitando una depuración asistida por IA más profunda.

¿Qué capacidad tiene para la gestión de archivos y cachés?

La herramienta incluye una función específica para la limpieza de cachés de desarrollo (`port-kill cache --clean`), capaz de detectar y eliminar archivos temporales y carpetas de dependencias (como node_modules o cachés de Rust) para optimizar el espacio en disco y resolver problemas de construcción corrupta.

¿Cómo se realiza la orquestación de múltiples servicios?

Mediante el uso de un archivo de configuración opcional denominado `.port-kill.yaml`, el usuario puede definir una lista de servicios, sus dependencias y retardos específicos de inicio. Esto permite levantar stacks tecnológicos completos (bases de datos, APIs y front-end) de forma ordenada con un único comando.

CONTRATOS Y CONDICIONES

Principales recomendaciones
- Realizar auditorías periódicas de los permisos de ejecución: Al ser una herramienta que requiere privilegios para finalizar procesos (matar puertos), se recomienda limitar su uso a usuarios con perfil técnico y en entornos de desarrollo controlados.
- Extremar precaución con el servidor MCP (Model Context Protocol): Se ha detectado una vulnerabilidad de severidad alta (inyección de comandos) en versiones específicas cuando se usa con herramientas de IA (como Cursor o Claude). No procesar entradas de usuarios externos a través de estos modelos.
- Verificar el origen de los binarios: Al instalar mediante scripts de ejecución directa (curl | bash), asegúrese de que el dominio y el certificado TLS pertenecen a la fuente oficial para evitar ataques de cadena de suministro.
- Configurar exclusiones en sistemas de producción: No utilizar en servidores de producción sin una configuración estricta de "listas blancas" para evitar el cierre accidental de servicios críticos del sistema o de la empresa.

Privacidad y protección de datos
- Responsabilidades: La empresa es responsable de la seguridad de la terminal donde se ejecute. La herramienta actúa de forma local, por lo que no existe una relación de "encargado de tratamiento" con el desarrollador (Treadie HQ).
- Ubicación de los datos: El procesamiento es 100% local. No se han identificado transferencias de datos personales a servidores externos durante la ejecución de los comandos de limpieza de puertos.
- Transferencia internacional: Únicamente aplicable si se integran funciones de limpieza de caché de terceros (Vercel, Cloudflare) mediante el uso de sus respectivas APIs, en cuyo caso la responsabilidad recae sobre dichos proveedores de nube.
- Derechos ARCO: No aplica de forma directa ya que la herramienta no crea perfiles de usuario ni almacena datos personales identificativos en sus bases de datos.

Propiedad intelectual
- Propiedad de datos: Los archivos de configuración (.port-kill.yaml) y la lógica de orquestación creada por el desarrollador pertenecen íntegramente a la empresa/usuario.
- Propiedad del resultado: El código fuente de Port Kill es open source (Código Abierto), lo que permite su auditoría y modificación interna, pero el software se entrega "tal cual" sin garantías explícitas por parte de los autores originales.

Usos y prohibiciones
- Usos prohibidos: Utilizar la herramienta para finalizar procesos de seguridad del sistema o de monitorización corporativa sin autorización. Uso del servidor MCP con prompts no saneados que puedan derivar en ejecución remota de código (RCE).
- Usos admitidos: Gestión de puertos en desarrollo local (localhost), limpieza de archivos temporales de compilación (caché de Rust, JS, Python) y orquestación de microservicios en entornos de pre-producción.

Seguridad y certificaciones
- Seguridad: La herramienta utiliza Rust como lenguaje principal, lo que reduce riesgos de seguridad de memoria (memory safety). Sin embargo, carece de firma de código corporativo estándar, lo que puede disparar alertas en algunos sistemas antivirus empresariales.
- Certificaciones: No cuenta con certificaciones tipo ISO 27001 o SOC2, habitual en herramientas de software local de código abierto gestionadas por comunidades o startups pequeñas.

Otros
- Vulnerabilidad conocida: Existe un reporte de seguridad (GHSA-4fmr-m2w5-f73j) sobre inyección de comandos en el componente port-kill-mcp. Se recomienda actualizar a la última versión disponible y no utilizar la función --remote con argumentos que provengan de fuentes no confiables.

Fuentes consultada:
- Repositorio oficial en GitHub
- Aviso de seguridad GHSA-4fmr-m2w5-f73j
- Web oficial y documentación
- Configuración de servidor MCP

Para más información y herramientas:

Explora look4.tools para descubrir las mejores soluciones tecnológicas del mercado.

Inicio Todas las herramientas Categorías