LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Inclusión obligatoria del instalador ensurepip en el core de Python para empaquetar pip por defecto vía PEP 453.

0x00

SEGURIDAD Y EXPLOITS
Malware y Exploits

Descubrimiento del ciberataque masivo a SolarWinds, infectando agencias estatales mediante actualizaciones de software legítimas.

0x01

SEGURIDAD Y EXPLOITS
Malware y Exploits

Alerta global por la vulnerabilidad Shellshock en el intérprete de comandos GNU Bash, latente de forma invisible durante más de 25 años.

0x02

LENGUAJES Y RUNTIMES
Frameworks Web

Nacimiento de FastAPI, articulando Starlette, Pydantic y tipado estático con generación automática de esquemas OpenAPI.

0x03

SISTEMAS Y REDES
Sistemas UNIX / Linux

Debut del intérprete de comandos GNU Bash (Bourne-Again SHell), diseñado como el reemplazo libre del clásico shell de Stephen Bourne.

0x04

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Oficialización del módulo enum en la biblioteca estándar para enumeraciones robustas bajo el PEP 435.

0x05

SISTEMAS Y REDES
Sistemas UNIX / Linux

Invención de las tuberías de procesamiento (pipes) en Unix, permitiendo encadenar la salida de un programa con la entrada de otro.

0x06

SEGURIDAD Y EXPLOITS
Desastres de Software

Crisis mortal de los aviones Boeing 737 MAX, cobrándose la vida de 346 personas en dos accidentes aéreos catastróficos.

0x07

CLOUD E INFRAESTRUCTURA
Automatización IaC

Publicación del paper sobre Chubby, el servicio de bloqueo distribuido con Paxos que sostiene los clústeres de Google.

0x08

SISTEMAS Y REDES
Sistemas UNIX / Linux

Anuncio oficial del Proyecto GNU en el grupo de noticias net.unix-wizards para construir un sistema Unix completamente libre.

0x09

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Paper de Dynamo de Amazon, demostrando las ventajas de la consistencia eventual y el diseño NoSQL distribuido.

0x0A

HARDWARE
Retrochips y GPUs

Lanzamiento del procesador Apple M1, demostrando la superioridad del silicio ARM frente a x86 en laptops de consumo.

0x0B

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Debut del emparejamiento de patrones estructurales (structural pattern matching) con sintaxis match/case en el PEP 634.

0x0C

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Publicación del paper fundacional sobre el modelo relacional de datos por parte de un investigador de IBM.

0x0D

SEGURIDAD Y EXPLOITS
Malware y Exploits

Detección milagrosa de una puerta trasera estatal (backdoor) infiltrada pacientemente durante años en el código fuente de XZ Utils.

0x0E

SEGURIDAD Y EXPLOITS
Desastres de Software

Histeria y gasto financiero global de más de 300,000 millones de dólares ante la llegada del temido Bug del Milenio (Y2K).

0x0F

CÓMPUTO E IA
Redes Neuronales y Papers

Formulación de la arquitectura Transformer en el célebre paper Attention Is All You Need, basada enteramente en autoatención.

0x10

SISTEMAS Y REDES
Herramientas de Desarrollo

Lanzamiento del editor Atom por parte de GitHub bajo el lema "A hackable text editor for the 21st Century".

0x11

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Vol 0: Kernel Foundations

Stéphane Chazelas

2014

Permitía inyectar comandos de sistema arbitrarios dentro de variables de entorno transmitidas a scripts CGI en Apache.

0x02

Vol 0: Kernel Foundations

Inteligencia Exterior Rusa (SVR)

2020

Inyectaron la puerta trasera Sunburst en el pipeline de compilación de SolarWinds, espiando al gobierno de EE.UU. por 9 meses.

0x01

Vol 0: Kernel Foundations

Donald Stufft y Nick Coghlan

2014

Erradicó el tedioso ritual de descargar manualmente scripts flotantes como get-pip.py para configurar entornos de desarrollo nuevos.

0x00

Vol 0: Kernel Foundations

Barry Warsaw y Eli Bendersky

2013

Terminó con años de soluciones ad-hoc caseras basadas en clases con constantes enteras o tuplas dispersas.

0x05

Vol 0: Kernel Foundations

Brian Fox (Free Software Foundation)

1989

Su nombre jugaba con Born Again; se convirtió en el shell interactivo por defecto indiscutible de casi todo Linux.

0x04

Vol 0: Kernel Foundations

Sebastián Ramírez (tiangolo)

2018

Revolucionó las APIs; su documentación interactiva autogenerada en Swagger UI lo catapultó al estrellato global.

0x03

Vol 0: Kernel Foundations

Mike Burrows (Google)

2006

Burrows diseñó Chubby enfatizando la facilidad de uso y la sincronización gruesa para miles de clientes.

0x08

Vol 0: Kernel Foundations

Boeing

2018

El software MCAS forzó el morro hacia abajo por un sensor defectuoso sin informar a los pilotos, causando 346 muertes.

0x07

Vol 0: Kernel Foundations

Doug McIlroy (Bell Labs)

1972

Thompson implementó las pipes en una noche; la filosofía Unix de "hacer una sola cosa y hacerla bien" quedó sellada.

0x06

Vol 0: Kernel Foundations

Apple (Johny Srouji)

2020

Unió CPU, GPU y memoria unificada en un silicio ultraeficiente, permitiendo a Apple romper su alianza histórica con Intel.

0x0B

Vol 0: Kernel Foundations

Amazon (Werner Vogels et al.)

2007

Nació tras caer el carrito de Amazon en el Black Friday de 2004, superando la rigidez de bases relacionales clásicas.

0x0A

Vol 0: Kernel Foundations

Richard Stallman (FSF)

1983

Su nombre es un acrónimo recursivo que significa "GNU's Not Unix"; sentó las bases legales, morales y filosóficas del movimiento del software libre.

0x09

Vol 0: Kernel Foundations

Andrés Freund (Microsoft)

2024

Andres Freund cazó el backdoor en xz por casualidad al notar que SSH consumía 500ms más de CPU en su computadora personal.

0x0E

Vol 0: Kernel Foundations

Edgar F. Codd (IBM Research)

1970

Directivos de IBM quisieron frenar el paper porque amenazaba el lucrativo negocio de sus bases de datos jerárquicas IMS.

0x0D

Vol 0: Kernel Foundations

Brandt Bucher y Guido van Rossum

2020

Aportó capacidades de descomposición de objetos complejas inspiradas en lenguajes como Scala y Rust, superando con creces a un simple switch.

0x0C

Vol 0: Kernel Foundations

GitHub (Chris Wanstrath)

2014

Para crearlo inventaron Electron (Chromium + Node.js), base que luego impulsaría apps como Slack, Discord y VS Code.

0x11

Vol 0: Kernel Foundations

Ashish Vaswani et al. (Google Brain)

2017

Diseñado para traducción, eliminó dependencias secuenciales y permitió entrenar modelos en paralelo a escala masiva.

0x10

Vol 0: Kernel Foundations

Industria Tecnológica Global

2000

Guardar años con 2 dígitos (99 en vez de 1999) amenazaba colapsar sistemas; una colosal inversión global evitó el caos.

0x0F

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Debut de las comprensiones de listas (list comprehensions) y del recolector de basura cíclico en Python 2.0.

0x12

CÓMPUTO E IA
Ecosistema Científico

Lanzamiento de Scikit-learn, toolkit canónico de machine learning para regresión, clasificación y clustering.

0x13

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Nacimiento de MongoDB, apostando por el almacenamiento de documentos semiestructurados en formato binario BSON y esquemas dinámicos.

0x14

CULTURA HACKER Y WEB
Plataformas de Colaboración

Consolidación del sitio de noticias técnicas Slashdot bajo el icónico lema "News for Nerds. Stuff that Matters".

0x15

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Inicio del movimiento por el software libre tras el incidente de una impresora láser Xerox atascada en el laboratorio del MIT.

0x16

CÓMPUTO E IA
Modelos de Lenguaje (LLMs)

Lanzamiento de Ollama, facilitando la descarga y ejecución de LLMs en la terminal local con un sencillo comando de una sola línea.

0x17

SEGURIDAD Y EXPLOITS
Desastres de Software

Falla del sensor de CrowdStrike, congelando infraestructuras globales por un puntero nulo en Windows.

0x18

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Adición de argumentos nombrados (keyword arguments) en las llamadas a funciones con la llegada de Python 1.1.

0x19

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Creación de Fortran (Formula Translating System), el primer lenguaje de programación de alto nivel comercialmente exitoso.

0x1A

SISTEMAS Y REDES
Herramientas de Desarrollo

Creación urgente de Git como sistema de control de versiones distribuido tras la crisis con el software propietario BitKeeper.

0x1B

CÓMPUTO E IA
Ecosistema Científico

Nacimiento de IPython, una consola interactiva avanzada para Python con comandos mágicos, resaltado y depuración en vivo.

0x1C

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Transformación de print de sentencia mágica de lenguaje a función estándar print() dictaminada en el PEP 3105.

0x1D

HARDWARE
Retrochips y GPUs

Lanzamiento del microprocesador Intel 8080, el cerebro que detonó la revolución de la informática personal en el Altair 8800.

0x1E

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Oficialización del tipo booleano primitivo bool con las constantes fijas True y False en el PEP 285.

0x1F

CULTURA HACKER Y WEB
Plataformas de Colaboración

Lanzamiento de Discord, plataforma de chat y voz que se convirtió en el punto de encuentro clave de comunidades dev.

0x20

SISTEMAS Y REDES
Herramientas de Desarrollo

Presentación de Visual Studio Code (VS Code), convirtiéndose en el editor de código más adoptado del planeta.

0x21

SEGURIDAD Y EXPLOITS
Malware y Exploits

Apagón informático mundial de CrowdStrike, provocando una pantalla azul de la muerte (BSOD) en bucle en 8.5 millones de máquinas Windows.

0x22

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Creación de Redis (REmote DIctionary Server), revolucionando el almacenamiento en memoria con estructuras de datos complejas nativas.

0x23

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Vol 0: Kernel Foundations

Eliot Horowitz et al. (10gen / MongoDB)

2009

Viene de "humongous"; lideró el auge NoSQL al permitir almacenar documentos JSON dinámicos sin migraciones de esquemas.

0x14

Vol 0: Kernel Foundations

Fabian Pedregosa et al. (scikit-learn)

2010

Iniciado en un Summer of Code de Google; su API .fit(), .predict() y .transform() fijó el estándar de oro.

0x13

Vol 0: Kernel Foundations

BeOpen PythonLabs (Guido van Rossum)

2000

La sintaxis [x for x in s] fue importada directamente del lenguaje funcional Haskell, transformando la manera idiomática de procesar colecciones.

0x12

Vol 0: Kernel Foundations

Jeffrey Morgan (Ollama)

2023

Empaquetó inferencia nativa en un CLI simple, volviéndose el estándar de facto para correr modelos y agentes locales.

0x17

Vol 0: Kernel Foundations

Richard Stallman (MIT AI Lab)

1983

Al negarle el código de una impresora atascada por acuerdos de confidencialidad, Stallman inició su rebelión por el software libre.

0x16

Vol 0: Kernel Foundations

Rob Malda (CmdrTaco)

1999

Originó el "Efecto Slashdot", la caída fulminante de webs pequeñas que colapsaban al ser enlazadas en su portada.

0x15

Vol 0: Kernel Foundations

John Backus (IBM)

1957

Había escepticismo de que un compilador igualara al ensamblador manual; el equipo de Backus demostró lo contrario.

0x1A

Vol 0: Kernel Foundations

Guido van Rossum

1994

Esta característica eliminó de raíz los errores por confusión posicional en funciones con listas largas de parámetros opcionales.

0x19

Vol 0: Kernel Foundations

CrowdStrike (Equipo de Calidad)

2024

Un puntero nulo en el archivo Channel File 291 desató un pantallazo azul masivo en Windows en millones de servidores del mundo.

0x18

Vol 0: Kernel Foundations

Georg Brandl

2006

Desató gran controversia entre veteranos, pero habilitó pasar argumentos nombrados como sep y end, además de permitir sobrescribir la función.

0x1D

Vol 0: Kernel Foundations

Fernando Pérez

2001

Fernando Pérez lo creó como estudiante de física cuántica en Colorado en una tarde para agilizar su tesis doctoral.

0x1C

Vol 0: Kernel Foundations

Linus Torvalds

2005

Linus programó el grafo DAG de Git en dos semanas, asombrando por su velocidad para ramificar y fusionar repositorios.

0x1B

Vol 0: Kernel Foundations

Jason Citron y Stanislav Vishnevskiy

2015

Con arquitectura distribuida en Elixir y Rust, absorbió el tráfico de viejos canales de IRC y servidores gamer.

0x20

Vol 0: Kernel Foundations

Guido van Rossum

2002

Por razones de compatibilidad hacia atrás estricta con código añejo, bool se definió como una subclase de int, haciendo que True + True == 2.

0x1F

Vol 0: Kernel Foundations

Federico Faggin y Masatoshi Shima

1974

Requería tres voltajes distintos (+5V, -5V y +12V) y operaba a una vertiginosa frecuencia de 2 MHz.

0x1E

Vol 0: Kernel Foundations

Salvatore Sanfilippo (antirez)

2009

Creado en Sicilia para analítica en tiempo real, deslumbró por su código C poético, eficiente y ultracompacto.

0x23

Vol 0: Kernel Foundations

CrowdStrike Inc. (George Kurtz)

2024

Un archivo corrupto de pocos KB en Windows colapsó aerolíneas, hospitales y cajeros en el mayor apagón informático mundial.

0x22

Vol 0: Kernel Foundations

Microsoft (Erich Gamma)

2015

Creado por Erich Gamma, sorprendió por ser libre, extensible, ágil y multiplataforma desde su primer día de lanzamiento.

0x21

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

SISTEMAS Y REDES
Estándares Web

Liberación histórica en código abierto del navegador Netscape Communicator, dando origen al proyecto Mozilla.

0x24

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Formalización del sistema de pistas de tipado estático (type hints) mediante la aprobación del PEP 484.

0x25

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Creación de la Free Software Foundation (FSF) y publicación del Manifiesto GNU en la revista Dr. Dobb's Journal.

0x26

CÓMPUTO E IA
Redes Neuronales y Papers

Presentación del dataset ImageNet, reuniendo más de 14 millones de fotografías para clasificar visión artificial.

0x27

CULTURA HACKER Y WEB
Cultura P2P

Lanzamiento de Napster, desatando la primera revolución planetaria de intercambio de música en archivos MP3 punto a punto (P2P).

0x28

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Incorporación de la función incorporada enumerate() para iterar índices y valores simultáneamente gracias al PEP 279.

0x29

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Aprobación del operador condicional ternario x if cond else y para expresiones en una sola línea en el PEP 308.

0x2A

HARDWARE
Retrochips y GPUs

Nacimiento de la computadora monoplaca Raspberry Pi, ofreciendo una placa Linux completa para enseñanza de software por solo $35 dólares.

0x2B

SISTEMAS Y REDES
Estándares Web

Redacción de la propuesta fundacional de la World Wide Web en un documento titulado "Information Management: A Proposal".

0x2C

LENGUAJES Y RUNTIMES
Ecosistema Rust

Debut del framework de testing pytest, revolucionando la verificación de código mediante reescritura de la sentencia estándar assert.

0x2D

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Fundación de la lista de correo de los Cypherpunks en la bahía de San Francisco bajo el lema "La privacidad a través de la criptografía".

0x2E

HARDWARE
Retrochips y GPUs

Presentación de CUDA, transformando tarjetas gráficas en supercomputadoras de cálculo paralelo masivo.

0x2F

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Lanzamiento del software de anonimato y enrutamiento en cebolla Tor (The Onion Routing) para proteger el tráfico web contra vigilancia.

0x30

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Creación de SQLite, un motor SQL embebido y contenido en una única librería en C sin necesidad de configuración ni procesos cliente-servidor.

0x31

SISTEMAS Y REDES
Sistemas UNIX / Linux

Presentación de Android, el sistema operativo móvil de Google basado en una versión personalizada del kernel de Linux.

0x32

CÓMPUTO E IA
IA Competitiva

Victoria mundial de AlphaGo sobre el legendario campeón surcoreano Lee Sedol por 4 victorias a 1 en el milenario juego del Go.

0x33

CÓMPUTO E IA
IA Competitiva

Demostración de refuerzo profundo (DQN) dominando juegos de Atari 2600 aprendiendo directo de los píxeles en pantalla.

0x34

SISTEMAS Y REDES
Sistemas UNIX / Linux

Fundación del proyecto Debian GNU/Linux bajo el manifiesto de crear una distribución universal guiada por principios éticos comunitarios.

0x35

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Vol 0: Kernel Foundations

Richard Stallman

1985

Fijó las cuatro libertades del software: usarlo, estudiarlo, modificarlo y compartir copias sin restricciones.

0x26

Vol 0: Kernel Foundations

Guido van Rossum, J. Lehtosalo y Langa

2015

Inspirado en mypy, revolucionó el análisis estático en grandes empresas sin alterar la ejecución dinámica de Python.

0x25

Vol 0: Kernel Foundations

Netscape Communications

1998

Hito histórico: una corporación entregó su código más valioso a la comunidad, germen que dio vida a Firefox.

0x24

Vol 0: Kernel Foundations

Raymond Hettinger

2003

Desterró el antipatrón omnipresente de usar variables contadoras manuales i = 0 seguidas de i += 1 dentro de bucles for.

0x29

Vol 0: Kernel Foundations

Shawn Fanning y Sean Parker

1999

Reunió a 80M de personas descargando MP3s en dormitorios universitarios hasta que demandas de Metallica y la RIAA lo cerraron.

0x28

Vol 0: Kernel Foundations

Fei-Fei Li (Princeton University)

2009

Se tildó de pérdida de tiempo; el dataset de Fei-Fei Li se convirtió en el combustible del boom del Deep Learning.

0x27

Vol 0: Kernel Foundations

Tim Berners-Lee (CERN)

1989

Su supervisor en el CERN anotó en la portada: "Vague but exciting..." antes de darle una computadora NeXT para programar.

0x2C

Vol 0: Kernel Foundations

Eben Upton (Raspberry Pi Foundation)

2012

Esperaban vender mil placas a alumnos de Cambridge; la demanda superó los 60 millones de unidades en todo el planeta.

0x2B

Vol 0: Kernel Foundations

Guido van Rossum y Raymond Hettinger

2005

Guido lo rechazó por años temiendo código feo, hasta ver que se abusaba del truco inseguro (cond and [x] or [y])[0].

0x2A

Vol 0: Kernel Foundations

NVIDIA (Ian Buck)

2006

Permitió programar GPUs en C/C++; fue el catalizador que permitió entrenar todas las grandes redes neuronales modernas.

0x2F

Vol 0: Kernel Foundations

E. Hughes, T. C. May y J. Gilmore

1992

Postularon que la criptografía fuerte es la única garantía de privacidad y libertad frente a corporaciones y estados.

0x2E

Vol 0: Kernel Foundations

Holger Krekel (PyPy Team)

2009

Eliminó herencias pesadas de unittest.TestCase, haciendo las pruebas legibles y concisas con simples assert.

0x2D

Vol 0: Kernel Foundations

Google (Andy Rubin)

2008

Llevó a Linux a miles de millones de teléfonos móviles, convirtiéndose en el SO con mayor base instalada del planeta.

0x32

Vol 0: Kernel Foundations

D. Richard Hipp

2000

Hipp lo creó a bordo de un buque de guerra tras ver caer servidores de bases de datos durante maniobras navales.

0x31

Vol 0: Kernel Foundations

R. Dingledine, N. Mathewson y Syverson

2004

Paradójicamente, Tor fue inventado y financiado por la Marina de EE.UU. para proteger comunicaciones de agentes encubiertos.

0x30

Vol 0: Kernel Foundations

Ian Murdock

1993

Murdock unió el nombre de su prometida Debra con el suyo (Ian), creando el estándar de empaquetado .deb y apt.

0x35

Vol 0: Kernel Foundations

DeepMind (Volodymyr Mnih et al.)

2013

En Breakout, aprendió por sí solo a perforar un túnel lateral para atrapar la bola arriba y sumar puntos sin parar.

0x34

Vol 0: Kernel Foundations

DeepMind (David Silver y D. Hassabis)

2016

En la partida 2 ejecutó el Movimiento 37, jugada que comentaristas creyeron un error y terminó redefiniendo el Go.

0x33

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CÓMPUTO E IA
Redes Neuronales y Papers

Publicación del devastador libro Perceptrons, demostrando matemáticamente que una sola capa lineal no podía resolver la función lógica XOR.

0x36

SEGURIDAD Y EXPLOITS
Malware y Exploits

Pandemia informática provocada por el ransomware WannaCry, bloqueando hospitales, bancos y redes ferroviarias en más de 150 países.

0x37

CÓMPUTO E IA
IA Competitiva

Victoria de Deep Blue derrotando en un match de ajedrez al campeón mundial Garry Kasparov.

0x38

CÓMPUTO E IA
IA Competitiva

Triunfo arrollador de la supercomputadora Watson derrotando a los dos campeones legendarios del concurso de televisión Jeopardy!.

0x39

SISTEMAS Y REDES
Sistemas UNIX / Linux

Introducción de systemd, rediseñando la inicialización, servicios y logs en el espacio de usuario de Linux.

0x3A

CÓMPUTO E IA
Modelos de Lenguaje (LLMs)

Presentación del motor de inferencia de alto rendimiento vLLM y su revolucionario algoritmo de memoria PagedAttention.

0x3B

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Presentación de PostgreSQL 6.0, marcando la refundación del proyecto con soporte SQL nativo completo impulsado por la comunidad libre.

0x3C

SEGURIDAD Y EXPLOITS
Malware y Exploits

Infección masiva del Gusano Morris, paralizando el 10% de todas las computadoras conectadas a la incipiente red ARPANET.

0x3D

SISTEMAS Y REDES
Herramientas de Desarrollo

Nacimiento del editor de texto extensible Emacs en el laboratorio de inteligencia artificial del MIT.

0x3E

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Creación de Clojure, resucitando la filosofía homoicónica de Lisp adaptada a estructuras de datos inmutables y concurrencia sobre la JVM.

0x3F

LENGUAJES Y RUNTIMES
Ecosistema Rust

Nacimiento del repositorio central PyPI (Python Package Index) bajo el alias humorístico "The Cheese Shop".

0x40

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Publicación del primer compilador funcional de Rust respaldado por Mozilla Research para sustituir a C++ con seguridad de memoria estricta.

0x41

SEGURIDAD Y EXPLOITS
Desastres de Software

Robo a The DAO en Ethereum, obligando a ejecutar un hard fork de la cadena de bloques para rescatar los fondos.

0x42

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Creación de C# (C-Sharp) como el pilar del nuevo entorno de desarrollo .NET de Microsoft para competir contra la plataforma Java.

0x43

SISTEMAS Y REDES
Herramientas de Desarrollo

Aparición de Sublime Text, conquistando a los desarrolladores por su velocidad instantánea y su minimapa lateral de código.

0x44

LENGUAJES Y RUNTIMES
CPython Runtime

Publicación del instalador oficial con soporte experimental para modo libre de hilos (free-threaded build) en Python 3.13.

0x45

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Anuncio de Go por Google, diseñado para compilar en segundos y simplificar concurrencia con goroutines.

0x46

SISTEMAS Y REDES
Estándares Web

Lanzamiento de Mozilla Firefox 1.0, recuperando la competencia real en el mercado con pestañas de navegación y bloqueador de pop-ups.

0x47

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Vol 0: Kernel Foundations

IBM (Feng-hsiung Hsu et al.)

1997

Kasparov acusó a IBM de trampa humana tras ser desconcertado por una jugada de Deep Blue provocada por un bug de software.

0x38

Vol 0: Kernel Foundations

The Shadow Brokers y Lazarus Group

2017

Usó el exploit militar EternalBlue de la NSA; Marcus Hutchins, de 22 años, lo frenó registrando un dominio en su código.

0x37

Vol 0: Kernel Foundations

Marvin Minsky y Seymour Papert (MIT)

1969

Su rigurosa crítica secó la financiación para redes neuronales por una década, desatando el Primer Invierno de la IA.

0x36

Vol 0: Kernel Foundations

UC Berkeley (Woosuk Kwon et al.)

2023

Inspirado en la paginación de los SO, redujo el desperdicio en caché KV, multiplicando por 20 los usuarios concurrentes.

0x3B

Vol 0: Kernel Foundations

Lennart Poettering y Kay Sievers

2010

Desató feroces flamewars en el software libre entre partidarios de la modernización y puristas de la filosofía Unix.

0x3A

Vol 0: Kernel Foundations

IBM Research (David Ferrucci)

2011

Watson leyó petabytes de libros offline, descifrando juegos de palabras, ironías y rimas en milisegundos en Jeopardy!.

0x39

Vol 0: Kernel Foundations

Richard Stallman y Guy Steele (MIT)

1976

Concebido por Stallman sobre macros TECO; extensible en Lisp, se convirtió en un entorno operativo en sí mismo.

0x3E

Vol 0: Kernel Foundations

Robert Tappan Morris (Cornell University)

1988

Morris solo quería medir Internet; un bug causó que su gusano se reinstalara en bucle infinito colapsando los servidores Unix.

0x3D

Vol 0: Kernel Foundations

PostgreSQL Global Development Group

1997

Sumó claves foráneas, índices GiST y disparadores, consagrándose como la base de datos relacional abierta más potente.

0x3C

Vol 0: Kernel Foundations

Graydon Hoare (Mozilla)

2010

Su modelo de pertenencia (borrow checker) eliminó fugas de memoria y condiciones de carrera sin usar garbage collector.

0x41

Vol 0: Kernel Foundations

Richard Jones

2003

Su apodo rendía homenaje a un célebre sketch de los Monty Python sobre una tienda de quesos que, cómicamente, no tenía queso alguno a la venta.

0x40

Vol 0: Kernel Foundations

Rich Hickey

2007

Hickey lo creó tomándose un año sabático con ahorros; su defensa de la inmutabilidad impactó a la ingeniería de datos moderna.

0x3F

Vol 0: Kernel Foundations

Jon Skinner

2008

Programado en C++ y Python sin lag; popularizó la selección con múltiples cursores simultáneos (Ctrl+D).

0x44

Vol 0: Kernel Foundations

Anders Hejlsberg (Microsoft)

2000

Creado por el autor de Turbo Pascal, dotó a C# de una rápida evolución sumando LINQ y tipado dinámico elegante.

0x43

Vol 0: Kernel Foundations

Christoph Jentzsch y Comunidad Ethereum

2016

Un hacker robó 3.6M de ether con un bug de reentrada, forzando un hard fork que dividió la red entre Ethereum y Ethereum Classic.

0x42

Vol 0: Kernel Foundations

Mozilla (Blake Ross y Joe Hewitt)

2004

Financiado por una campaña comunitaria que pagó un anuncio a doble página en The New York Times, rompió el estancamiento de Internet Explorer.

0x47

Vol 0: Kernel Foundations

Robert Griesemer, Rob Pike y Thompson

2009

Hartos de compilar horas en C++, ingenieros de Google crearon un lenguaje veloz con concurrencia basada en canales CSP.

0x46

Vol 0: Kernel Foundations

CPython Core Team

2024

Ejecuta CPython sin GIL, impulsando a NumPy y PyTorch a adaptar sus extensiones C a concurrencia real.

0x45

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Introducción de soporte nativo para números complejos y palabras clave con valores por defecto en Python 1.4.

0x48

SISTEMAS Y REDES
Sistemas UNIX / Linux

Cancelación unilateral del modelo tradicional de CentOS Linux por parte de Red Hat a favor de la variante experimental CentOS Stream.

0x49

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento de Podman en Red Hat, ejecutando contenedores sin demonio central y en modo rootless sin root.

0x4A

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Publicación oficial de Python 3.0 (Python 3000), rompiendo deliberadamente la compatibilidad hacia atrás para limpiar el lenguaje.

0x4B

CÓMPUTO E IA
Redes Neuronales y Papers

Divulgación del algoritmo de retropropagación del error (Backpropagation) en un célebre artículo en la revista Nature.

0x4C

CULTURA HACKER Y WEB
Plataformas de Colaboración

Lanzamiento oficial de GitHub bajo la bandera del "Social Coding", transformando los repositorios de Git en una red social interactiva.

0x4D

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Publicación del paper de CouchDB, pionera en almacenamiento de documentos JSON sin esquema con replicación multi-maestro.

0x4E

CÓMPUTO E IA
Ecosistema Científico

Presentación de Apache Arrow, definiendo un formato columnar en memoria para procesamiento analítico sin serialización.

0x4F

CÓMPUTO E IA
Modelos de Lenguaje (LLMs)

Presentación de GPT-2, reteniendo el modelo de 1.5B bajo el polémico argumento de ser "demasiado peligroso para liberarse".

0x50

SISTEMAS Y REDES
Estándares Web

Culminación y recomendación formal del estándar HTML5 por el W3C, integrando video nativo <video>, <audio> y gráficos <canvas>.

0x51

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Reemplazo de optparse por el nuevo módulo estándar argparse para parsing de CLI mediante el PEP 389.

0x52

SEGURIDAD Y EXPLOITS
Desastres de Software

Destrucción de la sonda Mariner 1 rumbo a Venus por un guion omitido en una fórmula manuscrita de Fortran.

0x53

SISTEMAS Y REDES
Herramientas de Desarrollo

Lanzamiento y explosión de adopción del editor Cursor, integrando capacidades agénticas de IA con indexación semántica del repositorio.

0x54

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Adopción del algoritmo C3 para el orden de resolución de métodos (Method Resolution Order - MRO) en Python 2.3.

0x55

SISTEMAS Y REDES
Herramientas de Desarrollo

Aparición de RCS (Revision Control System), popularizando el bloqueo estricto de archivos por usuario en entornos Unix.

0x56

SISTEMAS Y REDES
Sistemas UNIX / Linux

Lanzamiento de BSD (Berkeley Software Distribution), transformando el Unix académico con soporte para memoria virtual y TCP/IP.

0x57

SISTEMAS Y REDES
Estándares Web

Retiro oficial y corte de soporte definitivo de Internet Explorer 11 por parte de Microsoft tras 27 años ininterrumpidos de servicio.

0x58

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Nacimiento de MySQL, concebido como un motor relacional extremadamente rápido y ligero para bases de datos de páginas web.

0x59

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Vol 0: Kernel Foundations

Dan Walsh (Red Hat)

2018

Logró compatibilidad total con Docker eliminando el demonio central con privilegios de root (alias docker=podman).

0x4A

Vol 0: Kernel Foundations

Red Hat (IBM)

2020

Enfureció a miles de empresas de hosting, desatando el nacimiento inmediato de clones binarios como Rocky y AlmaLinux.

0x49

Vol 0: Kernel Foundations

Guido van Rossum

1996

Facilitó la adopción inmediata de Python en laboratorios científicos, sentando las bases del futuro ecosistema numérico del lenguaje.

0x48

Vol 0: Kernel Foundations

Chris Wanstrath et al. (GitHub)

2008

Proyecto de fin de semana en Rails; crearon el Pull Request visual, transformando la colaboración de software para siempre.

0x4D

Vol 0: Kernel Foundations

David Rumelhart, Hinton y Williams

1986

Hinton y su equipo demostraron que la retropropagación permitía a redes multicapa aprender representaciones profundas.

0x4C

Vol 0: Kernel Foundations

Guido van Rossum

2008

La separación tajante entre texto str (Unicode) y datos binarios bytes provocó una traumática transición de casi una década en la industria.

0x4B

Vol 0: Kernel Foundations

OpenAI (Alec Radford y Ilya Sutskever)

2019

Retenerlo por 'muy peligroso' causó acusaciones de marketing; meses después liberaron los pesos sin que ocurriera el apocalipsis.

0x50

Vol 0: Kernel Foundations

Wes McKinney y Apache Software Foundation

2016

Permite compartir datos en memoria sin coste de copia (zero-copy) entre Python, R, C++, Rust y motores SQL.

0x4F

Vol 0: Kernel Foundations

Damien Katz (Apache)

2005

Katz financió el desarrollo inicial vendiendo la mayor parte de sus pertenencias personales.

0x4E

Vol 0: Kernel Foundations

NASA

1962

Un guion faltante en una fórmula matemática desvió al cohete espacial hacia Venus, costando 18.5 millones de dólares.

0x53

Vol 0: Kernel Foundations

Steven Bethard

2009

Unificó la gestión de subcomandos, argumentos requeridos y mensajes de ayuda en la terminal, convirtiéndose en el estándar de utilitarios CLI.

0x52

Vol 0: Kernel Foundations

W3C y WHATWG (Ian Hickson)

2014

Materializó la visión de una plataforma de aplicaciones rica e independiente de plugins propietarios como Adobe Flash o Microsoft Silverlight.

0x51

Vol 0: Kernel Foundations

Walter F. Tichy (Purdue University)

1982

Guardaba deltas inversos para que ver la última versión fuera instantáneo, motor estándar de servidores compartidos.

0x56

Vol 0: Kernel Foundations

Michele Simionato

2003

Eliminó ambigüedades insolubles en herencia múltiple compleja, adaptando una técnica demostrada matemáticamente en el lenguaje Dylan.

0x55

Vol 0: Kernel Foundations

Anysphere

2024

Bifurcación de VS Code con IA nativa para editar múltiples archivos en conjunto y predecir cambios en tiempo real.

0x54

Vol 0: Kernel Foundations

Michael Widenius y David Axmark

1995

Nombrado por la hija de Monty, My; años después, al crear el fork libre tras vender MySQL a Sun, lo llamó MariaDB.

0x59

Vol 0: Kernel Foundations

Microsoft

2022

Celebrado por millones de devs frontend, permitió retirar incontables polyfills y parches viejos de la web.

0x58

Vol 0: Kernel Foundations

Bill Joy (UC Berkeley CSRG)

1977

Financiado por DARPA, su stack TCP/IP se convirtió en la base de las redes de telecomunicaciones de todo el planeta.

0x57

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CULTURA HACKER Y WEB
Flame Wars y Debates

Inicio de la campaña de demandas judiciales del consorcio SCO Group contra IBM y empresas de Linux alegando código robado de Unix.

0x5A

Vol 0: Kernel Foundations

CÓMPUTO E IA
Ecosistema Científico

Anuncio del proyecto Jupyter, desacoplando el notebook web de Python para soportar múltiples lenguajes.

0x5B

Vol 0: Kernel Foundations

CÓMPUTO E IA
Ecosistema Científico

Desarrollo de Pandas, introduciendo las estructuras Series y DataFrame para el análisis y manipulación de datos tabulares en Python.

0x5C

Vol 0: Kernel Foundations

SISTEMAS Y REDES
Sistemas UNIX / Linux

Lanzamiento de la primera versión de Ubuntu (Ubuntu 4.10 Warty Warthog), financiada por el millonario sudafricano Mark Shuttleworth.

0x5D

Vol 0: Kernel Foundations

CÓMPUTO E IA
Modelos de Lenguaje (LLMs)

Apertura pública de ChatGPT, convirtiéndose en la app de consumo de más rápido crecimiento en la historia.

0x5E

Vol 0: Kernel Foundations

SEGURIDAD Y EXPLOITS
Malware y Exploits

Vulnerabilidad crítica en el protocolo DNS, permitiendo suplantar dominios mediante envenenamiento de caché.

0x5F

Vol 0: Kernel Foundations

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Creación del módulo decimal en la biblioteca estándar para aritmética de coma flotante de precisión fija vía PEP 327.

0x60

Vol 0: Kernel Foundations

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Unificación definitiva de tipos primitivos y clases (new-style classes) con herencia obligatoria de object en el PEP 252.

0x61

Vol 0: Kernel Foundations

CULTURA HACKER Y WEB
Cultura P2P

Ofensiva judicial de las editoriales comerciales y ciberataque masivo contra el Internet Archive y la Wayback Machine.

0x62

Vol 0: Kernel Foundations

CULTURA HACKER Y WEB
Plataformas de Colaboración

Creación de la plataforma de preguntas y respuestas Stack Overflow, premiando el conocimiento técnico con puntos de reputación y medallas.

0x63

Vol 0: Kernel Foundations

CÓMPUTO E IA
IA Competitiva

Hazaña de la IA Libratus derrotando a profesionales del mundo en partidas de póker Texas Hold'em.

0x64

Vol 0: Kernel Foundations

HARDWARE
Retrochips y GPUs

Lanzamiento de Intel Pentium, abandonando la numeración para poder registrar una marca comercial propia.

0x65

Vol 0: Kernel Foundations

SEGURIDAD Y EXPLOITS
Malware y Exploits

Publicación de las vulnerabilidades de hardware Spectre y Meltdown, rompiendo el aislamiento de memoria en procesadores modernos.

0x66

Vol 0: Kernel Foundations

CULTURA HACKER Y WEB
Flame Wars y Debates

Inicio de la mítica Guerra de los Editores entre los partidarios devotos de Emacs y los usuarios de Vi / Vim.

0x67

Vol 0: Kernel Foundations

CÓMPUTO E IA
Ecosistema Científico

Liberación de Pandas 2.0, incorporando el backend de memoria de Apache Arrow como motor nativo opcional de tipos de datos.

0x68

Vol 0: Kernel Foundations

SISTEMAS Y REDES
Estándares Web

Creación del primer navegador y editor web del mundo, bautizado como WorldWideWeb, corriendo en el sistema operativo NeXTSTEP.

0x69

Vol 0: Kernel Foundations

SISTEMAS Y REDES
Herramientas de Desarrollo

Debut de TextMate para Mac OS X, redefiniendo la estética y ergonomía de los editores con snippets dinámicos y navegación difusa.

0x6A

Vol 0: Kernel Foundations

SISTEMAS Y REDES
Sistemas UNIX / Linux

Creación del sistema operativo Unix en los laboratorios Bell tras el fracaso y cancelación del mastodóntico proyecto Multics.

0x6B

Vol 0: Kernel Foundations

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Wes McKinney (AQR Capital Management)

2008

Nació en un hedge fund de Wall Street para manipular series temporales financieras complejas y valores NaN.

0x5C

Vol 0: Kernel Foundations

Fernando Pérez y Brian Granger

2014

Homenaje triple a Julia, Python y R, y a las lunas de Júpiter descubiertas por Galileo Galilei.

0x5B

Vol 0: Kernel Foundations

Darl McBride (The SCO Group)

2003

SCO demandó a usuarios de Linux alegando ser dueña de Unix; tras 10 años de litigios, los jueces fallaron a favor de Novell y SCO quebró.

0x5A

Vol 0: Kernel Foundations

Dan Kaminsky

2008

Kaminsky coordinó en secreto un parche simultáneo con gigantes de la industria antes de revelar el fallo de DNS, salvando la web.

0x5F

Vol 0: Kernel Foundations

OpenAI (Sam Altman y equipo)

2022

Llegó a 100M de usuarios en 60 días, forzando a Google a emitir un "Código Rojo" de emergencia ante la amenaza a su buscador.

0x5E

Vol 0: Kernel Foundations

Canonical (Mark Shuttleworth)

2004

Shuttleworth invirtió millones en enviar CDs de instalación gratis por correo a cualquier rincón del mundo.

0x5D

Vol 0: Kernel Foundations

Tribunales de EE.UU. y Hacktivistas

2024

Un fallo judicial contra el préstamo digital controlado de libros puso en jaque a la mayor biblioteca de preservación de la web.

0x62

Vol 0: Kernel Foundations

Guido van Rossum

2001

Resolvió la anomalía donde tipos nativos y clases de usuario diferían, unificando tipos y clases bajo el mismo modelo.

0x61

Vol 0: Kernel Foundations

Facundo Batista

2004

Resolvió los clásicos y embarazosos errores de redondeo de punto flotante en cálculos contables bancarios donde 0.1 + 0.2 != 0.3.

0x60

Vol 0: Kernel Foundations

Intel (Vinod Dham)

1993

Como los números no podían patentarse como marcas registradas, Intel bautizó a la 5ª generación "Pentium" (del griego pente).

0x65

Vol 0: Kernel Foundations

CMU (Tuomas Sandholm y Noam Brown)

2017

Calculó equilibrios de Nash en juegos de información oculta, aprendiendo a farolear y detectar engaños en póker.

0x64

Vol 0: Kernel Foundations

Jeff Atwood y Joel Spolsky

2008

Construido con .NET, desbancó a portales de pago y se convirtió en la salvación diaria de millones de desarrolladores.

0x63

Vol 0: Kernel Foundations

Pandas Development Team

2023

Mejoró drásticamente el uso de memoria en columnas de texto y nulos integrando el backend de Apache Arrow.

0x68

Vol 0: Kernel Foundations

Richard Stallman y Bill Joy

1985

Stallman creó la Iglesia de Emacs como San IGNUcius; fans de Vi bromeaban diciendo que Emacs usaba 8 MB y consumía toda la RAM.

0x67

Vol 0: Kernel Foundations

Jann Horn (Project Zero) y TU Graz

2018

Demostraron que la ejecución especulativa de CPUs Intel/AMD permitía a apps comunes espiar secretos del kernel en memoria.

0x66

Vol 0: Kernel Foundations

Ken Thompson y Dennis Ritchie (Bell)

1969

Thompson programó el primer Unix en un fin de semana en una PDP-7 de 8 KB para poder jugar a su videojuego Space Travel.

0x6B

Vol 0: Kernel Foundations

Allan Odgaard (MacroMates)

2004

Su coloreado de sintaxis y el atajo Cmd+T para abrir archivos al instante inspiraron a todos los editores modernos.

0x6A

Vol 0: Kernel Foundations

Tim Berners-Lee

1990

No solo mostraba hipervínculos, sino que permitía editar páginas al vuelo, visión interactiva perdida en navegadores comerciales.

0x69

Vol 0: Kernel Foundations

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

SISTEMAS Y REDES
Estándares Web

Publicación del estándar HTTP/2 en el RFC 7540, introduciendo multiplexación binaria sobre una única conexión TCP.

0x6C

Vol 0: Kernel Foundations

CÓMPUTO E IA
Ecosistema Científico

Lanzamiento de Polars, motor de DataFrames ultrarrápido escrito en Rust sobre memoria Apache Arrow.

0x6D

Vol 0: Kernel Foundations

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento de Memcached, un sistema distribuido de caché en memoria RAM para acelerar la carga de sitios web dinámicos.

0x6E

Vol 0: Kernel Foundations

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Estándar formal de Haskell, consolidando un lenguaje funcional puro, perezoso (lazy) y con sistema de mónadas.

0x6F

Vol 0: Kernel Foundations

SISTEMAS Y REDES
Sistemas UNIX / Linux

Relanzamiento del kernel Linux bajo licencia GNU GPL v2, facilitando su desarrollo comunitario masivo.

0x70

Vol 0: Kernel Foundations

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Desarrollo del lenguaje C en los laboratorios Bell para reescribir el sistema operativo Unix sobre máquinas PDP-11.

0x71

Vol 0: Kernel Foundations

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Debut de C++ ("C with Classes"), sumando orientación a objetos y sobrecarga sin perder rendimiento de C.

0x72

Vol 0: Kernel Foundations

SEGURIDAD Y EXPLOITS
Malware y Exploits

Propagación del virus ILOVEYOU mediante un correo electrónico que fingía ser una carta de amor en un script adjunto .vbs.

0x73

Vol 0: Kernel Foundations

CULTURA HACKER Y WEB
Cultura P2P

Apagón masivo de Internet en protesta contra las leyes de censura y propiedad intelectual SOPA y PIPA en el Congreso de EE.UU.

0x74

Vol 0: Kernel Foundations

SISTEMAS Y REDES
Estándares Web

Presentación de la biblioteca jQuery bajo el célebre lema publicitario "Write Less, Do More".

0x75

Vol 0: Kernel Foundations

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Introducción del directorio __pycache__ para almacenar archivos compilados .pyc mediante el PEP 3147.

0x76

Vol 0: Kernel Foundations

SISTEMAS Y REDES
Estándares Web

Lanzamiento comercial de Netscape Navigator 1.0, desatando el frenesí de la burbuja Dot-Com en Silicon Valley.

0x77

Vol 0: Kernel Foundations

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Filtración de los Documentos de Halloween, donde ingenieros de Microsoft admitían la enorme amenaza competitiva de Linux.

0x78

Vol 0: Kernel Foundations

SISTEMAS Y REDES
Sistemas UNIX / Linux

Elección oficial del pingüino Tux como la mascota y símbolo universal representativo del kernel Linux.

0x79

Vol 0: Kernel Foundations

SISTEMAS Y REDES
Sistemas UNIX / Linux

Publicación del histórico mensaje en comp.os.minix anunciando un sistema operativo gratuito desarrollado como simple pasatiempo.

0x7A

Vol 0: Kernel Foundations

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Demostración sorpresa de Docker en una charla relámpago de la conferencia PyCon, desatando la locura por los contenedores.

0x7B

Vol 0: Kernel Foundations

CÓMPUTO E IA
Ecosistema Científico

Creación de Numeric (precursor de NumPy), introduciendo el primer objeto de array multidimensional homogéneo en C para Python.

0x7C

Vol 0: Kernel Foundations

SEGURIDAD Y EXPLOITS
Desastres de Software

Explosión del cohete Ariane 5 (vuelo 501) a los 37 segundos, destruyendo una carga de 370 millones de dólares.

0x7D

Vol 0: Kernel Foundations

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Vol 0: Kernel Foundations

Brad Fitzpatrick (Danga Interactive)

2003

Creado para acelerar LiveJournal en Perl, fue reescrito en C para despachar cientos de miles de requests en RAM.

0x6E

Vol 0: Kernel Foundations

Ritchie Vink

2020

Con evaluación perezosa y paralelismo nativo en Rust, pulverizó los tiempos de procesamiento de Pandas.

0x6D

Vol 0: Kernel Foundations

IETF (Mark Nottingham et al.)

2015

Basado en SPDY de Google, multiplicó la velocidad web permitiendo multiplexación y compresión de cabeceras HPACK.

0x6C

Vol 0: Kernel Foundations

Dennis Ritchie (Bell Labs)

1972

Derivado del lenguaje B, su eficiencia implacable y punteros directos lo volvieron la lingua franca de la informática.

0x71

Vol 0: Kernel Foundations

Linus Torvalds

1992

Linus confesó que adoptar la licencia GPL v2 fue la mejor decisión estratégica de su vida para el éxito de Linux.

0x70

Vol 0: Kernel Foundations

Comité Haskell (Simon Peyton Jones)

1990

Nombrado por Haskell Curry; su pureza funcional y rigor de tipos inspiró a casi todos los lenguajes modernos.

0x6F

Vol 0: Kernel Foundations

Wikipedia, Reddit, Google y activistas

2012

Wikipedia y miles de sitios apagaron sus webs por 24 horas en señal de luto digital, enterrando la ley antipiratería SOPA.

0x74

Vol 0: Kernel Foundations

Onel de Guzman (Filipinas)

2000

Infectó 50M de PCs en 10 horas forzando al Pentágono a desconectarse; su autor fue absuelto por falta de leyes en Filipinas.

0x73

Vol 0: Kernel Foundations

Bjarne Stroustrup (Bell Labs)

1983

Stroustrup usó ++ para sugerir evolución de C; hoy es el estándar indiscutible de videojuegos y software de sistemas.

0x72

Vol 0: Kernel Foundations

Netscape (Jim Clark y M. Andreessen)

1994

Alcanzó el 80% de cuota de mercado; su salida a bolsa en 1995 fue el pistoletazo financiero de la era puntocom.

0x77

Vol 0: Kernel Foundations

Barry Warsaw

2011

Terminó con la molesta proliferación de archivos .pyc huérfanos mezclados entre los archivos de código fuente en los repositorios de trabajo.

0x76

Vol 0: Kernel Foundations

John Resig

2006

Ocultó las inconsistencias de navegadores con $("selector"), dominando más del 70% de la web durante una década.

0x75

Vol 0: Kernel Foundations

Linus Torvalds

1991

Linus anunció: "just a hobby, won't be big like gnu", sin prever que dominaría la infraestructura de servidores mundial.

0x7A

Vol 0: Kernel Foundations

Linus Torvalds y Larry Ewing

1996

Linus eligió el pingüino Tux recordando con humor cómo uno lo mordió en el dedo en el zoológico de Canberra en Australia.

0x79

Vol 0: Kernel Foundations

Eric S. Raymond y V. Valloppillil

1998

Memos internos filtrados alertaban a Bill Gates de que Linux y el código abierto superaban en calidad a Microsoft.

0x78

Vol 0: Kernel Foundations

Agencia Espacial Europea (ESA)

1996

Desbordamiento al convertir flotante de 64 bits a entero de 16 bits en código Ada reciclado, destruyendo el cohete en 37 segundos.

0x7D

Vol 0: Kernel Foundations

Jim Hugunin

1995

Creado con el laboratorio LLNL; demostró que Python podía rivalizar con MATLAB y Fortran en cálculo matricial.

0x7C

Vol 0: Kernel Foundations

Solomon Hykes (dotCloud)

2013

La demo en vivo de 5 minutos causó tal impacto que dotCloud abandonó su negocio PaaS para renacer como Docker Inc.

0x7B

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Despliegue industrial de algoritmos de criptografía poscuántica (PQC) certificados por el NIST en navegadores y servidores web.

Vol 0: Kernel Foundations
0x7E

CÓMPUTO E IA
IA Competitiva

Creación de AlphaZero, aprendiendo ajedrez, shogi y Go desde cero absoluto en menos de 24 horas jugando exclusivamente contra sí mismo.

Vol 0: Kernel Foundations
0x7F

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Publicación del manifiesto The Crypto Anarchist Manifesto, prediciendo la economía digital cifrada y el declive del control estatal.

Vol 1: Hacker Lore
1x00

CULTURA HACKER Y WEB
Flame Wars y Debates

Oleada de deserciones corporativas del código abierto tradicional hacia licencias de código disponible (SSPL, BSL y RSALv2).

Vol 1: Hacker Lore
1x01

CULTURA HACKER Y WEB
Flame Wars y Debates

Estallido del célebre debate Tanenbaum vs. Torvalds en el grupo de noticias comp.os.minix sobre la arquitectura de kernels.

Vol 1: Hacker Lore
1x02

CULTURA HACKER Y WEB
Flame Wars y Debates

Rebelión y choque ideológico por la invasión de systemd en las principales distribuciones de Linux.

Vol 1: Hacker Lore
1x03

CULTURA HACKER Y WEB
Cultura P2P

Creación de Gnutella, primera red P2P de intercambio de archivos totalmente descentralizada sin servidores centrales.

Vol 1: Hacker Lore
1x04

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Fundación de la Electronic Frontier Foundation (EFF) para defender los derechos civiles y la libertad de expresión en el ciberespacio.

Vol 1: Hacker Lore
1x05

SEGURIDAD Y EXPLOITS
Malware y Exploits

Descubrimiento del bug Heartbleed en OpenSSL, permitiendo extraer claves privadas de servidores HTTPS.

Vol 1: Hacker Lore
1x06

SEGURIDAD Y EXPLOITS
Malware y Exploits

Aparición del troyano bancario y malware modular Zeus (Zbot), robando credenciales bancarias mediante inyección web en navegadores.

Vol 1: Hacker Lore
1x07

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Fundación de la Open Source Security Foundation (OpenSSF) por la Linux Foundation para blindar los repositorios abiertos globales.

Vol 1: Hacker Lore
1x08

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Presentación de la propuesta del Clipper Chip por el gobierno de EE.UU., intentando imponer un chip de cifrado con puerta trasera estatal.

Vol 1: Hacker Lore
1x09

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Publicación de PGP (Pretty Good Privacy) en Usenet para ofrecer cifrado militar de comunicaciones a cualquier ciudadano común.

Vol 1: Hacker Lore
1x0A

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Impresión del código de PGP en formato de libro de tapa dura para exportarlo legalmente bajo el amparo de la Primera Enmienda.

Vol 1: Hacker Lore
1x0B

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Envío a la lista de correo de criptografía del whitepaper fundacional de Bitcoin bajo el seudónimo de Satoshi Nakamoto.

Vol 1: Hacker Lore
1x0C

SEGURIDAD Y EXPLOITS
Malware y Exploits

Ataque del virus Chernobyl (CIH), capaz de destruir físicamente placas madre sobrescribiendo la memoria Flash BIOS.

Vol 1: Hacker Lore
1x0D

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Primera edición de la conferencia hacker DEF CON en Las Vegas, originalmente concebida como fiesta de despedida de redes BBS.

Vol 1: Hacker Lore
1x0E

CULTURA HACKER Y WEB
Flame Wars y Debates

Popularización y posterior resaca arquitectónica de la fiebre de los Microservicios frente al Monolito.

Vol 1: Hacker Lore
1x0F

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Timothy C. May

1988

Fue leído por primera vez en la reunión fundacional de los cypherpunks en Aptos, California.

Vol 1: Hacker Lore · 1x00

Google DeepMind (David Silver et al.)

2017

Sin aperturas humanas, demolió al motor campeón Stockfish desplegando sacrificios y juego posicional asombroso.

Vol 0: Kernel Foundations · 0x7F

NIST, Cloudflare y Google

2024

Estandarizó algoritmos resistentes a computación cuántica (ML-KEM / Kyber), blindando el futuro del cifrado web.

Vol 0: Kernel Foundations · 0x7E

Lennart Poettering y comunidad Linux

2014

Detractores acusaron a systemd de romper la regla Unix de hacer una sola cosa bien, desatando feroces disputas en listas de correo.

Vol 1: Hacker Lore · 1x03

Andrew S. Tanenbaum y Linus Torvalds

1992

Tanenbaum afirmó en Usenet que "LINUX is obsolete"; Linus respondió con fiereza defendiendo la velocidad monolítica pragmática.

Vol 1: Hacker Lore · 1x02

MongoDB, Elastic, HashiCorp y Redis

2023

Firmas de código abierto acusaron a AWS de lucrar con sus proyectos gratis sin aportar código, empujando licencias como SSPL y BSL.

Vol 1: Hacker Lore · 1x01

Neel Mehta (Google) y Codenomicon

2014

Fallo de lectura de buffer en C; su logo del corazón sangrante inició la era de las vulnerabilidades con marca comercial.

Vol 1: Hacker Lore · 1x06

John Perry Barlow, Mitch Kapor

1990

Se creó tras las redadas abusivas del Servicio Secreto contra Steve Jackson Games por un juego de rol.

Vol 1: Hacker Lore · 1x05

Justin Frankel y Tom Pepper (Nullsoft)

2000

Lanzado a espaldas de AOL en Nullsoft; la empresa borró la descarga horas después, pero la red P2P ya era imparable e inmortal.

Vol 1: Hacker Lore · 1x04

NSA / Gobierno de EE.UU.

1993

Matt Blaze descubrió una debilidad crítica en el protocolo que permitía eludir la custodia de claves del gobierno.

Vol 1: Hacker Lore · 1x09

Linux Foundation y gigantes tecnológicos

2020

Surgió tras ataques a npm y PyPI, reuniendo fondos de gigantes tecnológicos para auditar proyectos críticos de código abierto.

Vol 1: Hacker Lore · 1x08

Evgeniy Bogachev (Slavik)

2007

Afectó a millones de computadoras corporativas y el FBI ofreció la recompensa cibernética más alta hasta entonces.

Vol 1: Hacker Lore · 1x07

Satoshi Nakamoto

2008

Resolvió el problema del doble gasto sin bancos centrales combinando cadenas de bloques (Blockchain) y prueba de trabajo.

Vol 1: Hacker Lore · 1x0C

Phil Zimmermann / MIT Press

1995

Al estar protegido como libertad de expresión en papel, cualquiera fuera de EE.UU. podía escanearlo mediante OCR.

Vol 1: Hacker Lore · 1x0B

Phil Zimmermann

1991

EE.UU. investigó penalmente a Zimmermann por exportar armas de guerra, pues la criptografía fuerte estaba clasificada como munición.

Vol 1: Hacker Lore · 1x0A

Martin Fowler y James Lewis

2014

Tras fragmentar apps simples en cientos de microservicios caóticos, ingenieros reivindicaron el valor del Majestic Monolith.

Vol 1: Hacker Lore · 1x0F

Jeff Moss (The Dark Tangent)

1993

Asistieron unas 100 personas y hoy reúne a más de 30,000 hackers e investigadores de todo el mundo.

Vol 1: Hacker Lore · 1x0E

Chen Ing-hau (Taiwán)

1998

Se activaba cada 26 de abril (aniversario de Chernóbil), borrando la BIOS de miles de PCs con Windows 95 y 98 en el mundo.

Vol 1: Hacker Lore · 1x0D

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

SEGURIDAD Y EXPLOITS
Malware y Exploits

Detección del ciberarma Stuxnet, saboteando físicamente centrifugadoras nucleares de enriquecimiento de uranio en Irán.

Vol 1: Hacker Lore 1x10

SEGURIDAD Y EXPLOITS
Malware y Exploits

Publicación de la herramienta de administración remota y troyano Back Orifice durante la conferencia DEF CON 6.

Vol 1: Hacker Lore 1x11

CULTURA HACKER Y WEB
Flame Wars y Debates

Debate filosófico entre defensores del Copyleft estricto (GPL) y las Licencias Permisivas (MIT/BSD).

Vol 1: Hacker Lore 1x12

CULTURA HACKER Y WEB
Cultura P2P

Inicio del Septiembre Eterno (Eternal September), cuando el proveedor AOL abrió las compuertas de Usenet a millones de usuarios novatos.

Vol 1: Hacker Lore 1x13

SEGURIDAD Y EXPLOITS
Desastres de Software

Fallo del sistema antimisiles Patriot en Dhahran, costando la vida a 28 soldados al no interceptar un misil Scud.

Vol 1: Hacker Lore 1x14

CULTURA HACKER Y WEB
Cultura P2P

Presentación del protocolo BitTorrent, resolviendo la distribución eficiente de archivos gigantescos mediante la compartición en enjambres.

Vol 1: Hacker Lore 1x15

SEGURIDAD Y EXPLOITS
Malware y Exploits

Arresto de Kevin Mitnick en Raleigh por el FBI tras una persecución técnica rastreada mediante análisis de registros celulares.

Vol 1: Hacker Lore 1x16

SEGURIDAD Y EXPLOITS
Malware y Exploits

Alerta por Log4Shell, vulnerabilidad crítica de ejecución remota en la biblioteca Java Apache Log4j.

Vol 1: Hacker Lore 1x17

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Publicación del Hacker Manifesto (The Conscience of a Hacker) en el fanzine Phrack tras ser arrestado.

Vol 1: Hacker Lore 1x18

CULTURA HACKER Y WEB
Cultura P2P

Lanzamiento del protocolo de compartición de archivos descentralizado Gnutella, desafiando la clausura judicial de Napster.

Vol 1: Hacker Lore 1x19

SEGURIDAD Y EXPLOITS
Malware y Exploits

Detección de la intrusión informática Operation Aurora, vulnerando a Google y decenas de corporaciones desde China.

Vol 1: Hacker Lore 1x1A

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Redacción de la primera versión de la GNU General Public License (GPL v1), inventando el concepto legal de Copyleft.

Vol 1: Hacker Lore 1x1B

CULTURA HACKER Y WEB
Cultura P2P

Levantamiento de r/wallstreetbets, coordinando la compra masiva de GameStop contra fondos de Wall Street.

Vol 1: Hacker Lore 1x1C

SEGURIDAD Y EXPLOITS
Malware y Exploits

Infiltración en el sistema central COSMOS de Pacific Bell mediante ingeniería social pura por un joven Kevin Mitnick.

Vol 1: Hacker Lore 1x1D

SEGURIDAD Y EXPLOITS
Malware y Exploits

Ataque del gusano SQL Slammer, colapsando la infraestructura de telecomunicaciones global en apenas diez minutos.

Vol 1: Hacker Lore 1x1E

SEGURIDAD Y EXPLOITS
Desastres de Software

Escándalo del software Horizon de la Oficina Postal británica, condenando injustamente a cientos de directores.

Vol 1: Hacker Lore 1x1F

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Fundación del colectivo ciberactivista Telecomix, combatiendo la censura en Oriente Medio durante la Primavera Árabe.

Vol 1: Hacker Lore 1x20

CULTURA HACKER Y WEB
Cultura P2P

Redada militar del FBI en Nueva Zelanda para clausurar el portal de descargas directas Megaupload y arrestar a su fundador Kim Dotcom.

Vol 1: Hacker Lore 1x21

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Richard Stallman vs. Apache y BSD

2007

Stallman defendió que el software derivado debe ser siempre libre; las corporaciones prefirieron MIT y Apache para cerrar código.

Vol 1: Hacker Lore · 1x12

Cult of the Dead Cow (cDc)

1998

Su nombre parodiaba al software de servidor corporativo Microsoft BackOffice.

Vol 1: Hacker Lore · 1x11

EE.UU. e Israel (Olympic Games)

2010

Usó cuatro 0-days y memorias USB para alterar centrifugadoras nucleares en Irán hasta hacerlas estallar físicamente.

Vol 1: Hacker Lore · 1x10

Bram Cohen

2001

Con la regla tit-for-tat que premia a quien más comparte, BitTorrent llegó a consumir un tercio del ancho de banda de Internet.

Vol 1: Hacker Lore · 1x15

Ejército de EE.UU. y Raytheon

1991

Un redondeo acumulado en el reloj tras 100 horas encendido desfasó el radar por un tercio de segundo, fallando ante un misil Scud.

Vol 1: Hacker Lore · 1x14

America Online (AOL)

1993

Cada septiembre llegaban universitarios novatos que aprendían netiqueta en 30 días; con AOL la oleada comercial masiva nunca cesó.

Vol 1: Hacker Lore · 1x13

The Mentor (Loyd Blankenship)

1986

Fue redactado después de que el autor fuera arrestado por el Servicio Secreto en Austin, Texas.

Vol 1: Hacker Lore · 1x18

Chen Zhaojun (Alibaba Cloud Security)

2021

Gravedad máxima 10/10 en CVSS; bastaba enviar texto malicioso en un chat de Minecraft para tomar control total del servidor.

Vol 1: Hacker Lore · 1x17

FBI / Tsutomu Shimomura

1995

Mitnick pasó casi cinco años en prisión preventiva antes de su juicio, convirtiéndose en icono de 'Free Kevin'.

Vol 1: Hacker Lore · 1x16

Richard Stallman

1989

Usó el copyright al revés (copyleft) para garantizar que cualquier trabajo derivado mantenga el código libre para siempre.

Vol 1: Hacker Lore · 1x1B

Google Information Security

2009

Llevó a Google a rediseñar toda su seguridad interna y dio origen al paradigma 'Zero Trust' de BeyondCorp.

Vol 1: Hacker Lore · 1x1A

Justin Frankel y Tom Pepper

2000

Nullsoft lo publicó en su web y AOL lo retiró horas después, pero el protocolo ya había sido descargado.

Vol 1: Hacker Lore · 1x19

Autor desconocido

2003

Con apenas 376 bytes en un paquete UDP sobre SQL Server, saturó routers y paralizó cajeros y aeropuertos globales.

Vol 1: Hacker Lore · 1x1E

Kevin Mitnick (The Condor)

1981

Mitnick y sus amigos obtuvieron listas de claves y manuales del sistema de telefonía hurgando en contenedores de basura.

Vol 1: Hacker Lore · 1x1D

Comunidad r/wallstreetbets

2021

Usuarios de Reddit causaron pérdidas multimillonarias a fondos de Wall Street coordinando compras de GameStop (short squeeze).

Vol 1: Hacker Lore · 1x1C

Departamento de Justicia de EE.UU. y FBI

2012

El FBI allanó la mansión de Kim Dotcom y clausuró el sitio (4% del tráfico web), borrando archivos legítimos sin juicio previo.

Vol 1: Hacker Lore · 1x21

Ciberactivistas Telecomix

2009

Proveyeron conexiones de acceso telefónico por módem a ciudadanos sirios y egipcios cuando apagaron Internet.

Vol 1: Hacker Lore · 1x20

Fujitsu y Post Office Ltd.

2020

Bugs en el software Horizon inventaron desfalcos falsos; Correos encubrió el fallo y mandó a prisión a cientos de directores inocentes.

Vol 1: Hacker Lore · 1x1F

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CULTURA HACKER Y WEB
Cultura P2P

Publicación de DeCSS por el adolescente "DVD Jon", reventando el sistema de cifrado comercial de películas en DVD.

Vol 1: Hacker Lore 1x22

CULTURA HACKER Y WEB
Flame Wars y Debates

Debate mundial sobre los derechos de autor y el uso legítimo (Fair Use) en el entrenamiento de modelos de lenguaje con código abierto.

Vol 1: Hacker Lore 1x23

SEGURIDAD Y EXPLOITS
Desastres de Software

Quiebra fulminante de la firma de bolsa Knight Capital Group tras perder $440 millones de dólares en apenas 45 minutos.

Vol 1: Hacker Lore 1x24

SEGURIDAD Y EXPLOITS
Malware y Exploits

Brecha de seguridad en Equifax, exponiendo datos financieros sensibles de más de 147 millones de personas.

Vol 1: Hacker Lore 1x25

SEGURIDAD Y EXPLOITS
Malware y Exploits

Filtración pública del exploit EternalBlue por el misterioso grupo The Shadow Brokers tras ser sustraído a la NSA.

Vol 1: Hacker Lore 1x26

SEGURIDAD Y EXPLOITS
Malware y Exploits

Arresto del célebre hacker Kevin Poulsen tras aparecer en el programa de TV Unsolved Mysteries y cortarse la emisión.

Vol 1: Hacker Lore 1x27

HARDWARE
Retrochips y GPUs

Lanzamiento del procesador AMD Opteron, introduciendo la extensión de 64 bits AMD64 y obligando a Intel a licenciarla.

Vol 1: Hacker Lore 1x28

SEGURIDAD Y EXPLOITS
Desastres de Software

Pérdida de la nave Mars Climate Orbiter, desintegrada en Marte por una catastrófica confusión de unidades de medida.

Vol 1: Hacker Lore 1x29

SEGURIDAD Y EXPLOITS
Desastres de Software

Colapso del exchange Mt. Gox tras el robo de 850,000 bitcoins por fallas en maleabilidad de transacciones.

Vol 1: Hacker Lore 1x2A

SEGURIDAD Y EXPLOITS
Malware y Exploits

Propagación del ciberataque destructivo NotPetya, paralizando puertos, hospitales y gigantes navieros mundiales.

Vol 1: Hacker Lore 1x2B

SEGURIDAD Y EXPLOITS
Malware y Exploits

Los siete miembros del grupo L0pht Heavy Industries testifican ante el Senado de EE.UU. afirmando poder derribar Internet en 30 minutos.

Vol 1: Hacker Lore 1x2C

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Publicación del primer número de la revista de culto hacker y phreaking 2600: The Hacker Quarterly.

Vol 1: Hacker Lore 1x2D

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Publicación del whitepaper de b-money, proponiendo dinero electrónico anónimo y descentralizado basado en prueba de trabajo.

Vol 1: Hacker Lore 1x2E

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Publicación de la licencia GNU GPL v3 tras meses de feroz debate comunitario para combatir la práctica de la "tivoización".

Vol 1: Hacker Lore 1x2F

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Filtración de documentos de la NSA revelando el programa de vigilancia PRISM y espionaje masivo de fibra óptica.

Vol 1: Hacker Lore 1x30

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Difusión del influyente ensayo The Cathedral and the Bazaar, analizando las virtudes del modelo de desarrollo abierto y caótico de Linux.

Vol 1: Hacker Lore 1x31

CULTURA HACKER Y WEB
Flame Wars y Debates

Choque cultural en la empresa Basecamp / 37signals tras prohibir debates políticos y sociales en los canales internos de trabajo.

Vol 1: Hacker Lore 1x32

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Creación del concepto de Bit Gold, una arquitectura teórica de dinero digital basada en acertijos computacionales.

Vol 1: Hacker Lore 1x33

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Knight Capital

2012

Olvidaron actualizar 1 de sus 8 servidores de bolsa; el código viejo revivió y compró a pérdida, perdiendo 440M en 45 minutos.

Vol 1: Hacker Lore 1x24

Devs independientes vs. Gigantes de IA

2024

Demandas contra Copilot cuestionaron si entrenar IA con repositorios públicos de código abierto viola licencias libres o es uso legítimo.

Vol 1: Hacker Lore 1x23

Jon Lech Johansen (DVD Jon)

1999

Jon quería ver DVDs legales en Linux; la industria del cine intentó censurar judicialmente la clave DeCSS en toda la web.

Vol 1: Hacker Lore 1x22

FBI

1991

Poulsen trucó las líneas de la radio KIIS-FM de Los Ángeles para ser la llamada número 102 y ganar un Porsche 944.

Vol 1: Hacker Lore 1x27

The Shadow Brokers

2017

Explotaba una falla en SMBv1 y semanas después alimentó los ataques globales de WannaCry y NotPetya.

Vol 1: Hacker Lore 1x26

Hackers estatales (acusación del DoJ)

2017

Entraron por un bug público en Apache Struts que administradores de Equifax olvidaron parchar por meses, robando 147M de datos.

Vol 1: Hacker Lore 1x25

Mark Karpelès (Mt. Gox)

2014

Procesaba el 70% de compras de Bitcoin del mundo; la pérdida de 850,000 bitcoins dejó a miles atrapados en litigios por una década.

Vol 1: Hacker Lore 1x2A

NASA y Lockheed Martin

1999

Lockheed usó libras-fuerza imperiales y la NASA esperaba newtons métricos, desintegrando la sonda en la atmósfera de Marte.

Vol 1: Hacker Lore 1x29

AMD (Fred Weber)

2003

Intel intentó imponer Itanium sin compatibilidad; el triunfo de AMD obligó a Intel a licenciar el estándar x86-64 de su rival.

Vol 1: Hacker Lore 1x28

Emmanuel Goldstein (Eric Corley)

1984

El número '2600' rinde homenaje a la frecuencia en hercios del famoso silbato de cereal Cap'n Crunch.

Vol 1: Hacker Lore 1x2D

L0pht (Mudge, Dildog, Weld Pond)

1998

Fue la primera vez que un comité senatorial reconoció a hackers usando exclusivamente sus 'handles' hacker.

Vol 1: Hacker Lore 1x2C

Sandworm (GRU)

2017

Disfrazado de ransomware, no tenía mecanismo real de descifrado y fue catalogado como el ciberataque más costoso.

Vol 1: Hacker Lore 1x2B

Edward Snowden (filtración masiva)

2013

Snowden arriesgó su vida para revelar que la NSA interceptaba llamadas, correos y chats de ciudadanos en Silicon Valley.

Vol 1: Hacker Lore 1x30

Richard Stallman (FSF)

2007

Frenó la "tivoización": prohibió empaquetar GPL en hardware con candados digitales que bloqueen versiones modificadas.

Vol 1: Hacker Lore 1x2F

Wei Dai

1998

Satoshi Nakamoto citó a b-money en la primera página del whitepaper de Bitcoin en 2008.

Vol 1: Hacker Lore 1x2E

Nick Szabo

1998

Diseñó también el concepto pionero de 'contratos inteligentes' (smart contracts) en 1994.

Vol 1: Hacker Lore 1x33

Jason Fried y David H. Hansson (DHH)

2021

Basecamp prohibió debates políticos internos; un tercio del equipo renunció en 48 horas desatando debate sobre activismo laboral.

Vol 1: Hacker Lore 1x32

Eric S. Raymond

1997

Acuñó la Ley de Linus: "Con suficientes ojos, todos los errores son superficiales"; inspiró a Netscape a abrir su código.

Vol 1: Hacker Lore 1x31

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CULTURA HACKER Y WEB
Cultura P2P

Fundación del portal de enlaces magnéticos The Pirate Bay por el colectivo sueco de cultura libre Piratbyrån.

Vol 1: Hacker Lore
1x34

SEGURIDAD Y EXPLOITS
Desastres de Software

Tragedia del acelerador Therac-25, administrando radiación letal por una condición de carrera en su software de control.

Vol 1: Hacker Lore
1x35

CULTURA HACKER Y WEB
Flame Wars y Debates

Apogeo del polarizado debate entre Espacios vs. Tabulaciones (Spaces vs. Tabs) para la indentación del código fuente.

Vol 1: Hacker Lore
1x36

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Nacimiento formal del término Open Source en una reunión estratégica en Palo Alto y fundación de la Open Source Initiative (OSI).

Vol 1: Hacker Lore
1x37

SEGURIDAD Y EXPLOITS
Malware y Exploits

Despliegue del ataque contra la red de Sony PlayStation Network, comprometiendo los datos de 77 millones de cuentas.

Vol 1: Hacker Lore
1x38

SEGURIDAD Y EXPLOITS
Malware y Exploits

Descubrimiento de la vulnerabilidad de suplantación de identidad en caché DNS Cache Poisoning a escala global.

Vol 1: Hacker Lore
1x39

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Lanzamiento de Tor (The Onion Router) como software libre bajo licencia BSD para anonimato en red.

Vol 1: Hacker Lore
1x3A

SEGURIDAD Y EXPLOITS
Malware y Exploits

Aparición de la vulnerabilidad por desbordamiento de enteros Stagefright en Android, explotable con un simple mensaje multimedia SMS.

Vol 1: Hacker Lore
1x3B

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Publicación del software esteganográfico Steghide, permitiendo ocultar mensajes cifrados dentro de archivos de imagen y audio.

Vol 1: Hacker Lore
1x3C

CULTURA HACKER Y WEB
Cultura P2P

Captura de Ross Ulbricht y clausura del mercado clandestino en la dark web Silk Road.

Vol 1: Hacker Lore
1x3D

HARDWARE
Retrochips y GPUs

Invención del concepto GPU (Graphics Processing Unit) por parte de NVIDIA con el lanzamiento de la tarjeta GeForce 256.

Vol 1: Hacker Lore
1x3E

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Publicación de A Cypherpunk's Manifesto, proclamando que la privacidad es vital para la sociedad digital.

Vol 1: Hacker Lore
1x3F

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Creación del lenguaje Lua en la universidad PUC-Río de Brasil como un motor de scripting embebido ultra liviano en C ANSI.

Vol 2: Hardware
2x00

HARDWARE
Retrochips y GPUs

Lanzamiento del microprocesador de 8 bits Zilog Z80, dominando la era del cómputo doméstico y salas arcade.

Vol 2: Hardware
2x01

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Debut de KVM (Kernel-based Virtual Machine), convirtiendo al propio kernel de Linux en un hipervisor de tipo 1 de alto rendimiento.

Vol 2: Hardware
2x02

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Introducción de la llamada al sistema chroot en Unix Versión 7, permitiendo enjaular la raíz del sistema de archivos para un proceso.

Vol 2: Hardware
2x03

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Creación de Perl como una herramienta de extracción y generación de informes (Practical Extraction and Report Language).

Vol 2: Hardware
2x04

HARDWARE
Retrochips y GPUs

Lanzamiento del procesador MOS Technology 6502 a una fracción del costo de sus rivales, encendiendo la revolución de la PC.

Vol 2: Hardware
2x05

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Comunidad Mundial de Programadores

2004

PEP 8 zanjó el debate con 4 espacios; un estudio en broma de Stack Overflow halló que devs que usan espacios ganaban más salario.

Vol 1: Hacker Lore · 1x36

Atomic Energy of Canada Limited (AECL)

1985

Retiraron los seguros mecánicos confiando ciegamente en software programado por una sola persona en ensamblador; murieron 6 pacientes.

Vol 1: Hacker Lore · 1x35

The Pirate Bay (Svartholm et al.)

2003

Famoso por publicar respuestas sarcásticas y desternillantes a las cartas de abogados y estudios de Hollywood.

Vol 1: Hacker Lore · 1x34

Dan Kaminsky

2008

Kaminsky coordinó en secreto el mayor parche conjunto de la historia de Internet con decenas de empresas.

Vol 1: Hacker Lore · 1x39

Atacantes no identificados

2011

Obligó a Sony a apagar sus servidores durante 23 días consecutivos en todo el mundo.

Vol 1: Hacker Lore · 1x38

C. Peterson, B. Perens y E. Raymond

1998

Acuñaron el término "Open Source" para alejar el sesgo político de "Free" y atraer a las corporaciones al código abierto.

Vol 1: Hacker Lore · 1x37

Stefan Hetzl

2002

Utiliza un algoritmo gráfico basado en teoría de emparejamiento para evitar alterar las frecuencias del archivo.

Vol 1: Hacker Lore · 1x3C

Joshua Drake (Zimperium)

2015

Afectó a casi mil millones de teléfonos Android sin requerir ninguna interacción por parte de la víctima.

Vol 1: Hacker Lore · 1x3B

Roger Dingledine, Nick Mathewson

2004

El concepto de enrutamiento cebolla original fue desarrollado por el Laboratorio de Investigación Naval de EE.UU.

Vol 1: Hacker Lore · 1x3A

Eric Hughes

1993

Hughes sentenció: "Los Cypherpunks escribimos código. Alguien debe defender la privacidad y vamos a escribirlo nosotros".

Vol 1: Hacker Lore · 1x3F

NVIDIA (Jensen Huang)

1999

Primer procesador gráfico dedicado en un chip con transformación geométrica e iluminación 3D integradas por hardware.

Vol 1: Hacker Lore · 1x3E

FBI (Agente Christopher Tarbell)

2013

Agentes del FBI fingieron una pelea de novios en una biblioteca para arrebatarle la laptop abierta a Ulbricht sin que cifrara el disco.

Vol 1: Hacker Lore · 1x3D

Avi Kivity (Qumranet)

2007

Integrado de inmediato en el kernel Linux; Qumranet fue comprada por Red Hat por más de 100 millones de dólares.

Vol 2: Hardware · 2x02

Federico Faggin y Masatoshi Shima

1976

Faggin dejó Intel para fundar Zilog tras haber codiseñado el 4004 y 8080.

Vol 2: Hardware · 2x01

Roberto Ierusalimschy et al. (PUC-Rio)

1993

Creado en Brasil, Lua se convirtió en el motor de scripting preferido de gigantes como World of Warcraft y Roblox.

Vol 2: Hardware · 2x00

Chuck Peddle y Bill Mensch

1975

Costaba solo 25 dólares cuando los chips de Motorola e Intel se vendían por cientos.

Vol 2: Hardware · 2x05

Larry Wall

1987

Acuñó el lema TMTOWTDI; dominó los scripts web CGI en los 90 antes del auge de Python y PHP.

Vol 2: Hardware · 2x04

Bill Joy (Bell Labs)

1979

Concebida originalmente para probar instaladores de software sin comprometer los archivos maestros del sistema operativo anfitrión.

Vol 2: Hardware · 2x03

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Fundación de la CNCF y la OCI para salvaguardar la interoperabilidad y estándares abiertos de contenedores.

Vol 2: Hardware 2x06

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Especificación de COBOL, diseñado con sintaxis cercana al inglés para aplicaciones y finanzas empresariales.

Vol 2: Hardware 2x07

SISTEMAS Y REDES
Estándares Web

Estándar HTTP/1.1 (RFC 2068), incorporando conexiones persistentes (keep-alive) y cabecera Host.

Vol 2: Hardware 2x08

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Presentación de Julia, diseñado para resolver el problema de los dos lenguajes en cómputo científico.

Vol 2: Hardware 2x09

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Presentación de Ada, un lenguaje militar de tipado estricto diseñado para sistemas embebidos de misión crítica y armamento militar.

Vol 2: Hardware 2x0A

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Lanzamiento de Scala, integrando la orientación a objetos pura con la programación funcional sobre la máquina virtual de Java.

Vol 2: Hardware 2x0B

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Presentación de Kotlin, un lenguaje pragmático para la JVM totalmente interoperable con Java y con comprobación estricta de nulos.

Vol 2: Hardware 2x0C

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Publicación de Pascal, concebido como un lenguaje estructurado estricto orientado a la enseñanza de buenas prácticas algorítmicas.

Vol 2: Hardware 2x0D

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Integración nativa de microcontenedores basados en WebAssembly en clusters de producción mediante el proyecto SpinKube.

Vol 2: Hardware 2x0E

HARDWARE
Retrochips y GPUs

Nacimiento de la venerable arquitectura x86 con la introducción al mercado del microprocesador de 16 bits Intel 8086.

Vol 2: Hardware 2x0F

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Aparición de PHP (Personal Home Page Tools), facilitando la incrustación de lógica dinámica directamente dentro de documentos HTML.

Vol 2: Hardware 2x10

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Presentación de BASIC en la universidad de Dartmouth, democratizando el acceso a la computación para estudiantes no científicos.

Vol 2: Hardware 2x11

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Nacimiento de Ruby, fusionando la elegancia de Smalltalk, la potencia funcional de Lisp y la ergonomía textual de Perl.

Vol 2: Hardware 2x12

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Presentación pública de Kubernetes (K8s) como proyecto de código abierto inspirado en los sistemas internos Borg y Omega.

Vol 2: Hardware 2x13

HARDWARE
Retrochips y GPUs

Lanzamiento del microprocesador de 16/32 bits Motorola 68000, motor del Apple Macintosh, Amiga y Atari ST.

Vol 2: Hardware 2x14

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Aparición de Zig, lenguaje de sistemas para reemplazar a C sin costos ocultos y con evaluación en compilación (comptime).

Vol 2: Hardware 2x15

HARDWARE
Retrochips y GPUs

Desarrollo del procesador ARM1, el primer diseño comercial de arquitectura RISC de Acorn Computers.

Vol 2: Hardware 2x16

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Creación de FreeBSD Jails, ampliando el concepto de jaula para aislar procesos, interfaces de red y cuentas de usuario completas.

Vol 2: Hardware 2x17

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

IETF (Roy Fielding et al.)

1997

La cabecera Host habilitó el hosting virtual, permitiendo que millones de webs distintas compartieran una misma IP.

Vol 2: Hardware 2x08

Grace Hopper y Comité CODASYL

1959

Grace Hopper defendió que el código debía escribirse en palabras legibles en inglés, venciendo burlas de teóricos militares.

Vol 2: Hardware 2x07

Linux Foundation y gigantes tecnológicos

2015

Docker donó la especificación de imágenes y su motor central de ejecución de bajo nivel runc para evitar la fragmentación propietaria del sector.

Vol 2: Hardware 2x06

Martin Odersky (EPFL)

2003

Unió tipado funcional avanzado con la JVM, siendo la base sobre la que se levantaron gigantes como Apache Spark y Twitter.

Vol 2: Hardware 2x0B

Jean Ichbiah (DoD)

1980

Nombrado por Ada Lovelace; se volvió estándar para controlar aviónica, reactores nucleares y trenes de alta velocidad.

Vol 2: Hardware 2x0A

Jeff Bezanson, S. Karpinski et al.

2012

Une la expresividad de Python con la velocidad de C y Fortran mediante despacho múltiple y compilación JIT.

Vol 2: Hardware 2x09

Deislabs (Fermin Galan et al.)

2024

Arranca cargas en Kubernetes en microsegundos consumiendo una fracción de los recursos de un contenedor Linux.

Vol 2: Hardware 2x0E

Niklaus Wirth

1970

Nombrado por Blaise Pascal, generaba p-code, precursor conceptual del bytecode moderno de Java y C#.

Vol 2: Hardware 2x0D

JetBrains (Andrey Breslav)

2011

Creado por JetBrains, vivió un éxito arrollador tras ser elegido por Google como lenguaje oficial para Android.

Vol 2: Hardware 2x0C

John G. Kemeny y Thomas E. Kurtz

1964

Encendió la era de las microcomputadoras; la versión de Bill Gates y Paul Allen fue el primer producto de Microsoft.

Vol 2: Hardware 2x11

Rasmus Lerdorf

1995

Lerdorf solo quería una colección de binarios C para rastrear visitas a su currículum online, sin buscar crear un lenguaje.

Vol 2: Hardware 2x10

Stephen Morse (Intel)

1978

IBM eligió su variante económica (8088) para la primera PC, otorgándole a la arquitectura x86 un reinado que perdura.

Vol 2: Hardware 2x0F

Motorola Semiconductor

1979

Tenía registros de 32 bits y 68,000 transistores, de donde tomó su emblemático nombre.

Vol 2: Hardware 2x14

Google (Joe Beda, B. Burns et al.)

2014

Del griego "timonel"; su logo de 7 radios homenajea a Seven of Nine de Star Trek, en alusión al proyecto Borg de Google.

Vol 2: Hardware 2x13

Yukihiro Matsumoto (Matz)

1993

Diseñado priorizando la felicidad humana sobre la máquina, detonó una revolución mundial con Ruby on Rails.

Vol 2: Hardware 2x12

Poul-Henning Kamp

2000

Diseñado en FreeBSD para evitar que usuarios de hosting compartido espiaran procesos de otras cuentas vecinas.

Vol 2: Hardware 2x17

Sophie Wilson y Steve Furber

1985

Funcionaba con tan poca energía que seguía operando solo con la corriente residual de prueba.

Vol 2: Hardware 2x16

Andrew Kelley

2016

Su comando zig cc incluye un compilador cruzado de C/C++, permitiendo compilar en cualquier SO sin dolores de cabeza.

Vol 2: Hardware 2x15

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Nacimiento de Lisp, introduciendo la notación de listas con paréntesis, recursión y recolección de basura.

Vol 2: Hardware
2x18

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento de VMware Workstation 1.0, logrando virtualizar sistemas operativos completos sobre procesadores Intel x86.

Vol 2: Hardware
2x19

HARDWARE
Retrochips y GPUs

Lanzamiento del procesador gráfico Sony PlayStation GPU, popularizando el renderizado de polígonos 3D en consolas domésticas.

Vol 2: Hardware
2x1A

HARDWARE
Retrochips y GPUs

Introducción de la arquitectura de microprocesador MIPS R2000, pionera en computación RISC comercial para estaciones de trabajo.

Vol 2: Hardware
2x1B

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Presentación de Solaris Zones (Contenedores de Solaris) en el sistema operativo Solaris 10 de Sun Microsystems.

Vol 2: Hardware
2x1C

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Lanzamiento de Visual Basic 1.0, permitiendo arrastrar controles para programar ventanas en Windows.

Vol 2: Hardware
2x1D

SISTEMAS Y REDES
Estándares Web

Estandarización y despliegue masivo de la API de WebGPU para aceleración de gráficos y cómputo de modelos de IA en el navegador.

Vol 2: Hardware
2x1E

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Implementación de JavaScript en apenas diez días para dotar de dinamismo y scripts al navegador Netscape Navigator.

Vol 2: Hardware
2x1F

SISTEMAS Y REDES
Sistemas UNIX / Linux

Publicación del sistema operativo educativo MINIX y su correspondiente libro de arquitectura de sistemas por Prentice Hall.

Vol 2: Hardware
2x20

HARDWARE
Retrochips y GPUs

Lanzamiento del procesador MOS 6510, corazón del Commodore 64, la computadora personal más vendida de la historia.

Vol 2: Hardware
2x21

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Lanzamiento de Erlang en Ericsson, implementando el modelo de actores para telecomunicaciones tolerantes a fallos.

Vol 2: Hardware
2x22

HARDWARE
Retrochips y GPUs

Lanzamiento del supercomputador de procesamiento vectorial Cray-1, con su característica forma de 'C' acolchada.

Vol 2: Hardware
2x23

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Donación oficial y graduación de Kubernetes 1.0 bajo la tutela de la nueva fundación CNCF.

Vol 2: Hardware
2x24

CÓMPUTO E IA
Redes Neuronales y Papers

Aplicación pionera de una red neuronal convolucional (CNN) entrenada por retropropagación para reconocer códigos postales manuscritos.

Vol 2: Hardware
2x25

HARDWARE
Retrochips y GPUs

Presentación de la arquitectura de aceleradores de inteligencia artificial NVIDIA Blackwell B200 con 208,000 millones de transistores.

Vol 2: Hardware
2x26

HARDWARE
Retrochips y GPUs

Creación del Intel 4004, el primer microprocesador comercial del mundo integrado en un único chip de silicio.

Vol 2: Hardware
2x27

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Nacimiento de Scheme, elegante variante de Lisp con alcance léxico estático y llamadas de cola optimizadas.

Vol 2: Hardware
2x28

CÓMPUTO E IA
Ecosistema Científico

Estandarización de tensores en formatos de coma flotante de precisión ultra reducida (FP8 y FP4) en todo el stack analítico de Python.

Vol 2: Hardware
2x29

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Sony y Toshiba

1994

Diseñado por Ken Kutaragi, era capaz de procesar hasta 360,000 polígonos planos por segundo.

Vol 2: Hardware 2x1A

D. Greene y M. Rosenblum (VMware)

1999

Inventaron traducción binaria al vuelo para sortear limitaciones x86 que no permitían virtualizar por hardware.

Vol 2: Hardware 2x19

John McCarthy (MIT)

1958

McCarthy concibió Lisp como un modelo teórico; su alumno Steve Russell sorprendió a todos al implementarlo en una IBM 704.

Vol 2: Hardware 2x18

Microsoft (Alan Cooper)

1991

Ideado por Alan Cooper bajo el proyecto Ruby, Bill Gates compró el prototipo de inmediato para Windows 3.0.

Vol 2: Hardware 2x1D

Sun Microsystems

2004

Permitía crear miles de entornos virtuales aislados en un solo kernel sin el pesado costo de emular hardware completo.

Vol 2: Hardware 2x1C

John L. Hennessy (Stanford / MIPS)

1986

La arquitectura MIPS potenció la consola Nintendo 64 y la estación espacial de Silicon Graphics.

Vol 2: Hardware 2x1B

Andrew S. Tanenbaum

1987

Diseñado con un microkernel limpio para enseñar a alumnos cómo funciona un SO sin violar el copyright de AT&T.

Vol 2: Hardware 2x20

Brendan Eich (Netscape)

1995

Pese a sus prisas de diseño, se convirtió en el único lenguaje ejecutado nativamente por todos los navegadores de la Tierra.

Vol 2: Hardware 2x1F

W3C GPU for the Web Community Group

2024

Da acceso directo a GPUs con Vulkan/Metal, permitiendo ejecutar inferencia de LLMs locales directo en la pestaña web.

Vol 2: Hardware 2x1E

Seymour Cray

1976

La forma en herradura fue diseñada para mantener los cables extremadamente cortos y reducir latencias.

Vol 2: Hardware 2x23

Joe Armstrong, R. Virding y Williams

1986

Bajo el lema "Let it crash", sus centrales telefónicas lograron disponibilidad de nueve nueves (99.9999999%).

Vol 2: Hardware 2x22

MOS Technology / Commodore

1982

Incluía un puerto de E/S de 8 bits integrado que permitía intercambiar el mapa de memoria mediante 'bank switching'.

Vol 2: Hardware 2x21

NVIDIA (Jensen Huang)

2024

Unió dos dados de silicio al límite físico de la litografía ultravioleta, catapultando a NVIDIA a la cima del valor de mercado global.

Vol 2: Hardware 2x26

Yann LeCun (Bell Labs)

1989

Adoptado por el correo de EE.UU., procesó hasta el 20% de cheques bancarios manuscritos de Norteamérica en los años 90.

Vol 2: Hardware 2x25

Google y Linux Foundation

2015

Consagró a Kubernetes frente a Swarm y Mesos, convirtiéndose en el sistema operativo indiscutible de la nube.

Vol 2: Hardware 2x24

Consorcio Científico Abierto

2025

Permite manipular matrices y pesos gigantes en laptops comunes, reduciendo a la mitad el uso de memoria RAM.

Vol 2: Hardware 2x29

Guy L. Steele y G. J. Sussman (MIT)

1975

Fue el vehículo didáctico del célebre libro SICP, formando a generaciones de científicos computacionales en el MIT.

Vol 2: Hardware 2x28

Federico Faggin et al. (Intel)

1971

Creado para una calculadora japonesa; Federico Faggin grabó sus iniciales ("F.F.") en una esquina del silicio de cada oblea producida.

Vol 2: Hardware 2x27

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Incorporación de los grupos de control (cgroups) en el kernel de Linux tras la contribución de parches de ingenieros de Google.

Vol 2: Hardware
2x2A

HARDWARE
Retrochips y GPUs

Estallido del escándalo del Pentium FDIV bug, provocando pérdidas de más de $475 millones de dólares a Intel en reposiciones de chips.

Vol 2: Hardware
2x2B

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Creación de Smalltalk, sentando las bases puras de la programación orientada a objetos basada en el paso dinámico de mensajes.

Vol 2: Hardware
2x2C

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento de Rocket (rkt) por parte de CoreOS, desafiando el crecimiento descontrolado del daemon monolítico de Docker.

Vol 2: Hardware
2x2D

LENGUAJES Y RUNTIMES
CPython Runtime

Creación de CircuitPython, una derivación educativa y amigable de MicroPython impulsada para placas de desarrollo Adafruit.

Vol 2: Hardware
2x2E

HARDWARE
Retrochips y GPUs

Lanzamiento del procesador digital de señales Texas Instruments TMS32010, pionero en procesamiento de señales en tiempo real.

Vol 2: Hardware
2x2F

HARDWARE
Retrochips y GPUs

Construcción del computador Apollo Guidance Computer (AGC) para guiar las misiones lunares Apolo de la NASA.

Vol 2: Hardware
2x30

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Creación de Elixir, llevando la robustez tolerante a fallos de la máquina virtual BEAM de Erlang a una sintaxis moderna inspirada en Ruby.

Vol 2: Hardware
2x31

HARDWARE
Retrochips y GPUs

Lanzamiento del microcontrolador de bajo costo con Wi-Fi integrado ESP8266, revolucionando los proyectos de IoT en todo el mundo.

Vol 2: Hardware
2x32

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Presentación pública de Java bajo el lema promocional "Write Once, Run Anywhere" gracias a la máquina virtual JVM.

Vol 2: Hardware
2x33

HARDWARE
Retrochips y GPUs

Lanzamiento del microcontrolador de 8 bits con memoria Flash AVR (ATmega), base del futuro ecosistema Arduino.

Vol 2: Hardware
2x34

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Debut de Swift, el moderno lenguaje compilado creado por Apple para reemplazar progresivamente a Objective-C en iOS y macOS.

Vol 2: Hardware
2x35

HARDWARE
Retrochips y GPUs

Lanzamiento del chip microcontrolador dual-core RP2040 y la placa Raspberry Pi Pico, el primer silicio propio de la fundación.

Vol 2: Hardware
2x36

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Aparición de Prolog (Programmation en Logique), fundando el paradigma de la programación lógica declarativa basada en cláusulas de Horn.

Vol 2: Hardware
2x37

HARDWARE
Retrochips y GPUs

Publicación de la especificación abierta y libre de regalías para la arquitectura de procesadores RISC-V.

Vol 2: Hardware
2x38

SISTEMAS Y REDES
Herramientas de Desarrollo

Publicación de Vim (Vi IMproved) para computadoras Commodore Amiga bajo un modelo de software benéfico (charityware).

Vol 2: Hardware
2x39

HARDWARE
Retrochips y GPUs

Presentación de la arquitectura SPARC (Scalable Processor Architecture), llevando cómputo RISC a estaciones Sun Microsystems.

Vol 2: Hardware
2x3A

HARDWARE
Retrochips y GPUs

Lanzamiento de la plataforma de desarrollo de código abierto Arduino (placa USB) en el Interaction Design Institute Ivrea.

Vol 2: Hardware
2x3B

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Alan Kay, Dan Ingalls et al. (Xerox)

1972

Pionero de ventanas e iconos; Steve Jobs visitó Xerox PARC en 1979 y quedó maravillado por su interfaz y orientación a objetos.

Vol 2: Hardware · 2x2C

Thomas Nicely (descubridor) e Intel

1994

Un profesor descubrió una tabla de división defectuosa en el coprocesador; la soberbia de Intel costó 475 millones en reemplazos.

Vol 2: Hardware · 2x2B

Paul Menage y Rohit Seth (Google)

2007

Creados en Google como Process Containers, fijaron límites estrictos de RAM, CPU e I/O por grupo de procesos.

Vol 2: Hardware · 2x2A

Texas Instruments

1983

Tenía arquitectura Harvard modificada que permitía multiplicar enteros de 16 bits en un solo ciclo de reloj.

Vol 2: Hardware · 2x2F

Adafruit Industries (Scott Shawcroft)

2017

El microcontrolador se monta como memoria USB, permitiendo programar editando directo el archivo code.py.

Vol 2: Hardware · 2x2E

CoreOS (Alex Polvi)

2014

Acusó a Docker de abandonar su estándar modular; el conflicto forzó a la industria a crear la especificación de la OCI.

Vol 2: Hardware · 2x2D

Espressif Systems

2014

Comenzó con documentación filtrada en foros chinos y pronto acumuló millones de firmwares comunitarios.

Vol 2: Hardware · 2x32

José Valim

2012

Valim, del núcleo de Rails, descubrió la potencia de Erlang y creó Elixir para dominar la concurrencia web moderna.

Vol 2: Hardware · 2x31

MIT Instrumentation Lab

1966

Usaba memoria de núcleos trenzados a mano por trabajadoras textiles, apodada 'memoria LOL' (Little Old Ladies).

Vol 2: Hardware · 2x30

Chris Lattner (Apple)

2014

Diseñado en secreto por Chris Lattner, sorprendió en WWDC al erradicar la arcaica sintaxis de corchetes de Objective-C.

Vol 2: Hardware · 2x35

Alf-Egil Bogen y Vegard Wollan (Atmel)

1997

Los dos estudiantes noruegos de NTH concibieron una arquitectura RISC optimizada para código en lenguaje C.

Vol 2: Hardware · 2x34

James Gosling (Sun Microsystems)

1995

Nació como Oak para TV interactiva; el auge de Internet llevó a Sun a convertirlo en el motor del software empresarial.

Vol 2: Hardware · 2x33

Krste Asanović y David Patterson

2010

Nació en UC Berkeley como un proyecto académico de verano para evitar pagar costosas licencias x86/ARM.

Vol 2: Hardware · 2x38

Alain Colmerauer y Robert Kowalski

1972

En vez de pasos imperativos, declara hechos lógicos para que el motor de inferencia deduzca respuestas automáticamente.

Vol 2: Hardware · 2x37

Raspberry Pi Foundation

2021

Introdujo bloques PIO (Programmable I/O) capaces de sintetizar protocolos de hardware a medida.

Vol 2: Hardware · 2x36

Massimo Banzi, David Cuartielles

2005

Su nombre proviene de un bar en Ivrea, Italia, donde solían reunirse los fundadores del proyecto.

Vol 2: Hardware · 2x3B

Sun Microsystems

1987

Implementó una innovadora ventana de registros circulares para agilizar llamadas a funciones.

Vol 2: Hardware · 2x3A

Bram Moolenaar

1991

Bram pedía a los usuarios donar a orfanatos en Uganda, manteniendo esta causa humanitaria activa por más de 30 años.

Vol 2: Hardware · 2x39

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

HARDWARE
Retrochips y GPUs

Lanzamiento del microcontrolador PIC16C84, el primer chip embebido popular con memoria EEPROM reprogramable.

Vol 2: Hardware
2x3C

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Lanzamiento público de TypeScript, introduciendo tipado estático opcional y capacidades de compilación a JavaScript estándar.

Vol 2: Hardware
2x3D

LENGUAJES Y RUNTIMES
CPython Runtime

Financiación y estreno de MicroPython, reimplementación de Python 3 optimizada para microcontroladores y chips embebidos.

Vol 2: Hardware
2x3E

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Estandarización de WebAssembly (Wasm) como el cuarto lenguaje oficial de la web junto a HTML, CSS y JavaScript.

Vol 2: Hardware
2x3F

SISTEMAS Y REDES
Estándares Web

Publicación del RFC 882, introduciendo el Domain Name System (DNS) para reemplazar el archivo estático HOSTS.TXT de ARPANET.

Vol 3: Unix & Networks
3x00

SISTEMAS Y REDES
Sistemas UNIX / Linux

Nacimiento de Alpine Linux, una distribución orientada a routers embebidos construida sobre la librería de C musl y BusyBox.

Vol 3: Unix & Networks
3x01

SISTEMAS Y REDES
Herramientas de Desarrollo

Lanzamiento de Mercurial (hg), concebido la misma semana que Git como alternativa distribuida escrita en Python y C.

Vol 3: Unix & Networks
3x02

SISTEMAS Y REDES
Estándares Web

Publicación del RFC 1105 definiendo el Border Gateway Protocol (BGP), el pegamento de enrutamiento que une Internet.

Vol 3: Unix & Networks
3x03

SISTEMAS Y REDES
Herramientas de Desarrollo

Lanzamiento de Open vSwitch (OVS), el conmutador virtual multicapa de producción para redes definidas por software (SDN).

Vol 3: Unix & Networks
3x04

CLOUD E INFRAESTRUCTURA
Automatización IaC

Presentación de la legendaria conferencia "10+ Deploys Per Day: Dev and Ops Cooperation at Flickr", alumbrando el movimiento DevOps.

Vol 3: Unix & Networks
3x05

SISTEMAS Y REDES
Estándares Web

Publicación del RFC 5321 actualizando la especificación formal del protocolo de transporte de correo SMTP.

Vol 3: Unix & Networks
3x06

SISTEMAS Y REDES
Estándares Web

Lanzamiento de Network Time Protocol (NTP) (RFC 958), sincronizando relojes informáticos a través de redes de paquetes con latencia.

Vol 3: Unix & Networks
3x07

SISTEMAS Y REDES
Sistemas UNIX / Linux

Creación de Slackware Linux, la distribución comercial más veterana que aún se mantiene en desarrollo activo continuo.

Vol 3: Unix & Networks
3x08

CULTURA HACKER Y WEB
Plataformas de Colaboración

Lanzamiento de SourceForge, el primer portal centralizado gratuito para hospedar proyectos de código abierto con CVS, foros y descargas.

Vol 3: Unix & Networks
3x09

SISTEMAS Y REDES
Estándares Web

Publicación del RFC 2616 definiendo la especificación estándar de HTTP/1.1, introduciendo conexiones persistentes Keep-Alive.

Vol 3: Unix & Networks
3x0A

SISTEMAS Y REDES
Estándares Web

Creación del grupo rebelde de estandarización WHATWG por ingenieros de Apple, Mozilla y Opera para rescatar el HTML.

Vol 3: Unix & Networks
3x0B

SISTEMAS Y REDES
Herramientas de Desarrollo

Creación de la utilidad de diagnóstico Traceroute, rastreando los saltos de enrutamiento en redes IP usando el campo TTL.

Vol 3: Unix & Networks
3x0C

SISTEMAS Y REDES
Herramientas de Desarrollo

Desarrollo de Vi, el editor modal de pantalla completa incluido en la primera distribución BSD de Unix.

Vol 3: Unix & Networks
3x0D

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Damien George

2013

Empaquetó compilador, runtime y REPL de Python en chips con apenas 16 KB de RAM, abriendo la electrónica embebida.

Vol 2: Hardware · 2x3E

Anders Hejlsberg (Microsoft)

2012

Pese al escepticismo inicial, se consagró como el estándar global de facto para desarrollo a gran escala en TypeScript.

Vol 2: Hardware · 2x3D

Microchip Technology

1993

Permitió a los ingenieros borrar y reescribir código en segundos sin usar lámparas de luz ultravioleta.

Vol 2: Hardware · 2x3C

Natanael Copa

2005

Se convirtió en la imagen base favorita de millones de contenedores Docker por su diminuto tamaño de apenas 5 MB.

Vol 3: Unix & Networks · 3x01

Paul Mockapetris (USC-ISI)

1983

Antes del DNS, Elizabeth Feinler en el NIC actualizaba manualmente el archivo HOSTS.TXT por teléfono.

Vol 3: Unix & Networks · 3x00

W3C (World Wide Web Consortium)

2019

Permitió ejecutar binarios de C, C++, Rust y Go en el navegador a velocidades casi nativas dentro de una máquina virtual segura y aislada en sandbox.

Vol 2: Hardware · 2x3F

Ben Pfaff, Justin Pettit (Nicira)

2009

Permitió a los hipervisores conmutar tráfico de máquinas virtuales a velocidad de línea mediante OpenFlow.

Vol 3: Unix & Networks · 3x04

Kirk Lougheed y Yakov Rekhter

1989

El borrador inicial fue bosquejado en tres servilletas manchadas de kétchup durante una reunión de IETF.

Vol 3: Unix & Networks · 3x03

Matt Mackall

2005

Compitió con Git por Linux; aunque Linus usó Git, firmas como Facebook y Google operaron monorepos colosales con Mercurial.

Vol 3: Unix & Networks · 3x02

David L. Mills (Univ. de Delaware)

1985

Mills operaba satélites y antenas de radio en su propio patio trasero para calibrar señales horarias.

Vol 3: Unix & Networks · 3x07

John Klensin

2008

Consolidó tres décadas de extensiones de correo sobre el diseño original de Jon Postel de 1982.

Vol 3: Unix & Networks · 3x06

John Allspaw y Paul Hammond (Flickr)

2009

Mostraron que desarrollo y operaciones podían colaborar en armonía, desterrando los despliegues traumáticos de antaño.

Vol 3: Unix & Networks · 3x05

IETF (Roy Fielding, Jim Gettys et al.)

1999

Solucionó la ineficiencia de abrir una conexión TCP distinta para cada imagen o recurso de una página web.

Vol 3: Unix & Networks · 3x0A

VA Linux Systems (Tim Perdue)

1999

Alojaba casi todo el software libre del mundo, hasta que inyectar adware en instaladores arruinó su reputación.

Vol 3: Unix & Networks · 3x09

Patrick Volkerding

1993

Mantuvo una adhesión estricta a la filosofía Unix tipo BSD, con scripts de inicio limpios y sin herramientas opacas.

Vol 3: Unix & Networks · 3x08

Bill Joy (UC Berkeley)

1976

Joy usó h,j,k,l porque el terminal Lear Siegler ADM-3A tenía flechas impresas en esas teclas físicas.

Vol 3: Unix & Networks · 3x0D

Van Jacobson (LBL)

1988

Se le ocurrió la idea tras el gran colapso por congestión de Internet ocurrido a finales de los 80.

Vol 3: Unix & Networks · 3x0C

Ian Hickson, David Hyatt et al.

2004

Se rebelaron contra los planes del W3C de abandonar HTML para imponer el complejo e intolerante estándar XHTML 2.0, fundando las bases de HTML5.

Vol 3: Unix & Networks · 3x0B

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

SISTEMAS Y REDES
Estándares Web

Publicación del RFC 3031 formalizando la arquitectura MPLS (Multiprotocol Label Switching) para redes troncales.

3x0E

SISTEMAS Y REDES
Sistemas UNIX / Linux

Nacimiento de Red Hat Linux, apostando por el empaquetado binario corporativo mediante el formato .rpm.

3x0F

SISTEMAS Y REDES
Estándares Web

Firma del pacto histórico entre W3C y WHATWG, acordando que HTML se regirá únicamente como un estándar vivo (Living Standard).

3x10

SISTEMAS Y REDES
Herramientas de Desarrollo

Lanzamiento de IntelliJ IDEA, revolucionando los entornos de desarrollo integrados con autocompletado y refactorización semántica avanzada.

3x11

SISTEMAS Y REDES
Herramientas de Desarrollo

Creación de SCCS (Source Code Control System), el primer sistema automatizado de control de versiones de código de la historia.

3x12

SISTEMAS Y REDES
Sistemas UNIX / Linux

Integración de un kernel de Linux real completo dentro del sistema operativo Windows con la llegada de WSL2.

3x13

SISTEMAS Y REDES
Sistemas UNIX / Linux

Creación del programa Sudo (Superuser Do), permitiendo a administradores delegar comandos privilegiados con auditoría.

3x14

SISTEMAS Y REDES
Sistemas UNIX / Linux

Conquista del umbral histórico del 4% de cuota de mercado en computadoras de escritorio por parte de Linux a nivel mundial.

3x15

SISTEMAS Y REDES
Estándares Web

Lanzamiento del protocolo de enrutamiento criptográfico seguro y anónimo de baja latencia I2P (Invisible Internet Project).

3x16

SISTEMAS Y REDES
Herramientas de Desarrollo

Primer lanzamiento de la shell Bash (Bourne Again SHell) como parte del Proyecto GNU, reemplazando la Bourne Shell original.

3x17

SISTEMAS Y REDES
Herramientas de Desarrollo

Lanzamiento de la herramienta de sincronización incremental de archivos remotos Rsync, minimizando el ancho de banda transferido.

3x18

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Lanzamiento de la primera versión del protocolo y suite de cifrado remoto SSH-1 (Secure Shell) para sustituir a telnet y rsh.

3x19

SISTEMAS Y REDES
Estándares Web

Debut del navegador gráfico NCSA Mosaic, introduciendo por primera vez la etiqueta <img> para visualizar imágenes intercaladas con texto.

3x1A

SISTEMAS Y REDES
Estándares Web

Presentación de Safari 1.0 para Mac OS X por parte de Steve Jobs, impulsado por el motor de código abierto WebKit.

3x1B

CULTURA HACKER Y WEB
Cypherpunks y FOSS

Creación del proyecto OpenSSH como bifurcación abierta y libre de la suite SSH original dentro de OpenBSD.

3x1C

SISTEMAS Y REDES
Estándares Web

Presentación de Internet Explorer 1.0 por parte de Microsoft, incluido como un paquete complementario en Microsoft Plus! for Windows 95.

3x1D

SISTEMAS Y REDES
Herramientas de Desarrollo

Nacimiento de Subversion (SVN), diseñado para corregir las graves deficiencias estructurales de CVS.

3x1E

SISTEMAS Y REDES
Herramientas de Desarrollo

Lanzamiento del multiplexor de terminales en modo texto GNU Screen, permitiendo sesiones persistentes en consolas Unix.

3x1F

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

W3C y WHATWG

2019

Puso fin a quince años de disputas burocráticas entre organismos, unificando la documentación canónica de la web bajo un único consorcio.

Vol 3: Unix & Networks

3x10

Marc Ewing y Bob Young

1994

Nombrado por la gorra roja de Cornell que Ewing usaba en el laboratorio para que la gente le pidiera ayuda técnica.

Vol 3: Unix & Networks

3x0F

Eric Rosen, Arun Viswanathan

2001

Aceleró el enrutamiento de paquetes en proveedores de Internet sustituyendo búsquedas IP por etiquetas fijas.

Vol 3: Unix & Networks

3x0E

Microsoft

2019

Simbolizó la reconciliación de Microsoft con Linux, permitiendo correr binarios y herramientas ELF nativas en Windows.

Vol 3: Unix & Networks

3x13

Marc Rochkind (Bell Labs)

1972

Creado para IBM sobre tarjetas perforadas; guardaba deltas entrelazados para ahorrar espacio en discos magnéticos chicos.

Vol 3: Unix & Networks

3x12

JetBrains (S. Dmitriev y V. Kipiatkov)

2001

Primer IDE en procesar el árbol sintáctico en RAM para refactorizar Java de forma segura, triunfando con cobro de licencias.

Vol 3: Unix & Networks

3x11

jrandom

2003

Implementa enrutamiento de ajo (garlic routing) agrupando múltiples mensajes para dificultar el análisis de tráfico.

Vol 3: Unix & Networks

3x16

Comunidad Linux y Valve Software

2024

Impulsado por la consola Steam Deck de Valve y Proton, demostró la madurez de Linux en el gaming de alta gama.

Vol 3: Unix & Networks

3x15

Bob Coggeshall y Cliff Spencer

1980

Nació en SUNY Buffalo porque los operadores necesitaban reiniciar colas de impresión sin ser root.

Vol 3: Unix & Networks

3x14

Tatu Ylönen (HUT)

1995

Lo programó en pocas semanas después de que un ataque de sniffing comprometiera su red universitaria.

Vol 3: Unix & Networks

3x19

Andrew Tridgell y Paul Mackerras

1996

Su algoritmo de suma de verificación rotatoria revolucionó los respaldos en redes lentas.

Vol 3: Unix & Networks

3x18

Brian Fox (Free Software Foundation)

1989

Su nombre es un juego de palabras homenajeando a Steve Bourne y al concepto de 'renacimiento'.

Vol 3: Unix & Networks

3x17

Theo de Raadt y Markus Friedl

1999

Hoy protege los accesos administrativos de prácticamente todos los servidores Linux del planeta.

Vol 3: Unix & Networks

3x1C

Apple (Don Melton)

2003

Apple bifurcó KHTML de KDE para crear WebKit, desterrando para siempre a Internet Explorer de las computadoras Mac.

Vol 3: Unix & Networks

3x1B

Marc Andreessen y Eric Bina

1993

Desató la fiebre web; antes de Mosaic, las fotos debían descargarse como archivos y verse en ventanas externas separadas.

Vol 3: Unix & Networks

3x1A

Oliver Laumann y Carsten Bormann

1987

Permitió que una sesión no se cerrara si la conexión por módem telefónico se cortaba repentinamente.

Vol 3: Unix & Networks

3x1F

CollabNet (Jim Blandy y Karl Fogel)

2000

Sumó commits atómicos y renombrado limpio de carpetas y archivos binarios sin romper el historial del repositorio.

Vol 3: Unix & Networks

3x1E

Microsoft (Thomas Reardon)

1995

Microsoft licenció el código de Spyglass Mosaic, desatando la Primera Guerra de los Navegadores contra Netscape.

Vol 3: Unix & Networks

3x1D

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

SISTEMAS Y REDES
Herramientas de Desarrollo

Presentación pública de Zed, un editor de código colaborativo de latencia ultra baja programado íntegramente en lenguaje Rust.

Vol 3: Unix & Networks
3x20

SISTEMAS Y REDES
Estándares Web

Publicación del RFC 2131 estandarizando DHCP (Dynamic Host Configuration Protocol) para asignación automática de IPs.

Vol 3: Unix & Networks
3x21

SISTEMAS Y REDES
Estándares Web

Apagado definitivo y bloqueo de ejecución de Adobe Flash Player en todo el ecosistema global de navegadores web.

Vol 3: Unix & Networks
3x22

CULTURA HACKER Y WEB
Plataformas de Colaboración

Apagón masivo de comunidades (Reddit Blackout) en protesta por las abusivas tarifas impuestas por la directiva al acceso a su API.

Vol 3: Unix & Networks
3x23

SISTEMAS Y REDES
Estándares Web

Publicación de la especificación de HTTP/3 en el RFC 9114, reemplazando el protocolo TCP por conexiones sobre QUIC (UDP).

Vol 3: Unix & Networks
3x24

SISTEMAS Y REDES
Sistemas UNIX / Linux

Lanzamiento del sistema de inicio orientado a eventos Upstart en Ubuntu, diseñado para reemplazar SysV init asíncronamente.

Vol 3: Unix & Networks
3x25

SISTEMAS Y REDES
Estándares Web

Presentación del sistema de archivos distribuido de red NFS (Network File System) permitiendo montar discos transparentemente.

Vol 3: Unix & Networks
3x26

SISTEMAS Y REDES
Herramientas de Desarrollo

Lanzamiento del multiplexor de terminales moderno Tmux, optimizado con soporte cliente-servidor nativo y BSD-friendly.

Vol 3: Unix & Networks
3x27

SISTEMAS Y REDES
Estándares Web

Estandarización de JavaScript como ECMAScript 1st Edition (ES1) formalizada en la norma ECMA-262.

Vol 3: Unix & Networks
3x28

SISTEMAS Y REDES
Estándares Web

Creación del protocolo de bitácora remota Syslog para el proyecto Sendmail en entornos BSD Unix.

Vol 3: Unix & Networks
3x29

SISTEMAS Y REDES
Sistemas UNIX / Linux

Lanzamiento del programador cron moderno Vixie Cron, introduciendo sintaxis avanzada de temporizado para SysAdmins.

Vol 3: Unix & Networks
3x2A

SISTEMAS Y REDES
Herramientas de Desarrollo

Publicación del analizador de tráfico de paquetes Tcpdump y la biblioteca portátil libpcap.

Vol 3: Unix & Networks
3x2B

SISTEMAS Y REDES
Estándares Web

Lanzamiento de Internet Explorer 6.0, alcanzando un monopolio del 95% de la navegación web mundial.

Vol 3: Unix & Networks
3x2C

SISTEMAS Y REDES
Herramientas de Desarrollo

Lanzamiento del gestor de paquetes alternativo de Linux Nix, implementando despliegues puramente funcionales y atómicos.

Vol 3: Unix & Networks
3x2D

CLOUD E INFRAESTRUCTURA
Automatización IaC

Creación de CFEngine en la Universidad de Oslo, formulando la teoría del mantenimiento del estado deseado y la idempotencia.

Vol 3: Unix & Networks
3x2E

SISTEMAS Y REDES
Estándares Web

Fundación del consorcio W3C (World Wide Web Consortium) en el MIT para coordinar la evolución de los estándares abiertos de la red.

Vol 3: Unix & Networks
3x2F

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Divulgación pública del concepto Cloud Computing por parte de Eric Schmidt en una conferencia de ejecutivos en San José.

Vol 3: Unix & Networks
3x30

SISTEMAS Y REDES
Herramientas de Desarrollo

Creación de CVS (Concurrent Versions System), permitiendo a múltiples programadores modificar un mismo archivo en paralelo.

Vol 3: Unix & Networks
3x31

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Adobe Systems y gigantes web

2020

Cumplió la profecía de la carta de Steve Jobs de 2010, enterrando 25 años de Flash por seguridad y rendimiento.

Vol 3: Unix & Networks 3x22

Ralph Droms (Bucknell University)

1997

Evolucionó a partir del protocolo BOOTP para satisfacer la demanda de clientes móviles en redes LAN.

Vol 3: Unix & Networks 3x21

Nathan Sobo (Zed Industries)

2023

Creado por los autores de Atom, abandonó Electron para renderizar con la GPU a 120 FPS fluidos en lenguaje Rust.

Vol 3: Unix & Networks 3x20

Scott James Remnant (Canonical)

2006

Gestionaba de manera nativa la conexión y desconexión de dispositivos USB en caliente antes de la llegada de systemd.

Vol 3: Unix & Networks 3x25

IETF (Robin Marx et al.)

2022

Reduce la latencia móvil y erradica el bloqueo de cabeza de línea de TCP usando canales UDP cifrados.

Vol 3: Unix & Networks 3x24

Comunidades y Moderadores de Reddit

2023

Miles de subreddits cerraron en protesta por cobros de APIs, quebrando apps populares como Apollo y avivando redes libres.

Vol 3: Unix & Networks 3x23

Ecma International

1997

Netscape cedió el lenguaje a Ecma en Suiza para protegerlo ante los intentos de Microsoft de fragmentarlo con JScript.

Vol 3: Unix & Networks 3x28

Nicholas Marriott

2007

Diseñado como alternativa ligera a Screen con una arquitectura limpia de cliente-servidor.

Vol 3: Unix & Networks 3x27

Sun Microsystems (Russ Sandberg)

1984

Operaba sobre RPC y XDR, convirtiendo a las estaciones de trabajo en clientes sin disco local.

Vol 3: Unix & Networks 3x26

Van Jacobson, Craig Leres (LBL)

1988

Introdujo el filtro de paquetes BPF (Berkeley Packet Filter), ancestro directo de eBPF.

Vol 3: Unix & Networks 3x2B

Paul Vixie

1987

Vixie lo escribió cansado de las limitaciones del comando cron tradicional de AT&T Unix.

Vol 3: Unix & Networks 3x2A

Eric Allman (UC Berkeley)

1980

Operó como estándar 'de facto' durante más de dos décadas antes de ser formalizado en el RFC 3164.

Vol 3: Unix & Networks 3x29

Mark Burgess

1993

Físico teórico, aplicó teoría de control para que un script pudiera ejecutarse mil veces dando siempre el mismo estado.

Vol 3: Unix & Networks 3x2E

Eelco Dolstra (Univ. de Utrecht)

2004

Almacena paquetes bajo hashes criptográficos únicos en /nix/store, erradicando el infierno de dependencias.

Vol 3: Unix & Networks 3x2D

Microsoft

2001

Al derrotar a Netscape, Microsoft desarmó el equipo de IE; sus bugs de renderizado congelaron la web por un lustro.

Vol 3: Unix & Networks 3x2C

Dick Grune

1986

Scripts sobre RCS con fusiones optimistas, eliminando el bloqueo manual de archivos que paralizaba a los equipos.

Vol 3: Unix & Networks 3x31

Eric Schmidt (Google)

2006

Eric Schmidt definió la nube como un nuevo paradigma donde datos y cómputo residen en servidores accesibles vía web.

Vol 3: Unix & Networks 3x30

Tim Berners-Lee

1994

Berners-Lee rechazó patentar la web o cobrar regalías, garantizando que fuera un bien común libre y universal.

Vol 3: Unix & Networks 3x2F

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

SISTEMAS Y REDES
Estándares Web

Lanzamiento de Wireshark (originalmente bautizado como Ethereal), el analizador de protocolos de red con interfaz gráfica.

Vol 3: Unix & Networks · 3x32

CLOUD E INFRAESTRUCTURA
Automatización IaC

Presentación de Argo CD, un controlador declarativo de despliegue continuo para Kubernetes guiado por principios GitOps.

Vol 3: Unix & Networks · 3x33

SISTEMAS Y REDES
Estándares Web

Aprobación de la recomendación de CSS1 (Cascading Style Sheets) por parte del W3C para separar la estructura visual del diseño.

Vol 3: Unix & Networks · 3x34

SISTEMAS Y REDES
Estándares Web

Adopción generalizada de la extensión de seguridad Encrypted Client Hello (ECH) en conexiones TLS para blindar la privacidad web.

Vol 3: Unix & Networks · 3x35

CULTURA HACKER Y WEB
Plataformas de Colaboración

Lanzamiento del programa GitHub Sponsors, permitiendo a individuos y empresas financiar económicamente a mantenedores de código abierto.

Vol 3: Unix & Networks · 3x36

SISTEMAS Y REDES
Estándares Web

Lanzamiento sorpresa del navegador Google Chrome y su motor de ejecución de JavaScript V8 con compilación a código máquina en memoria.

Vol 3: Unix & Networks · 3x37

SISTEMAS Y REDES
Estándares Web

Estándar ECMAScript 5 (ES5), sumando el modo estricto "use strict", getters/setters y JSON.parse().

Vol 3: Unix & Networks · 3x38

SISTEMAS Y REDES
Herramientas de Desarrollo

Creación de Neovim, bifurcación de Vim enfocada en refactorizar el código en C y añadir scripting nativo en Lua.

Vol 3: Unix & Networks · 3x39

SISTEMAS Y REDES
Herramientas de Desarrollo

Fallecimiento de Bram Moolenaar, conmocionando a la comunidad global de desarrollo tras 32 años de custodia ininterrumpida de Vim.

Vol 3: Unix & Networks · 3x3A

SISTEMAS Y REDES
Estándares Web

Acuñación del concepto AJAX (Asynchronous JavaScript and XML) en un influyente ensayo de diseño web interactivo.

Vol 3: Unix & Networks · 3x3B

SISTEMAS Y REDES
Estándares Web

Publicación de la especificación oficial de HTTP/1.0 bajo el documento de estándares RFC 1945.

Vol 3: Unix & Networks · 3x3C

SISTEMAS Y REDES
Herramientas de Desarrollo

Lanzamiento del emulador de terminal moderno y acelerado por GPU Alacritty, priorizando simplicidad y velocidad extrema.

Vol 3: Unix & Networks · 3x3D

CULTURA HACKER Y WEB
Plataformas de Colaboración

Creación de GitLab en Ucrania como una alternativa de código abierto autohospedable a GitHub.

Vol 3: Unix & Networks · 3x3E

SISTEMAS Y REDES
Sistemas UNIX / Linux

Salida triunfal de Red Hat a la bolsa de Wall Street en pleno apogeo de la euforia financiera de la era puntocom.

Vol 3: Unix & Networks · 3x3F

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento comercial de Oracle Version 2, convirtiéndose en el primer sistema de gestión de bases de datos relacional con SQL del mercado.

Vol 4: Backend · 4x00

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento de la base de datos clave-valor distribuida en memoria Aerospike, optimizada para flash y latencia submilisegundo.

Vol 4: Backend · 4x01

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de Terraform 0.1, consolidando el paradigma de Infraestructura como Código (IaC) agnóstico a cualquier proveedor de nube.

Vol 4: Backend · 4x02

CLOUD E INFRAESTRUCTURA
Automatización IaC

Adquisición millonaria de Ansible por parte de Red Hat para integrarlo como el cerebro de automatización de sus soluciones empresariales.

Vol 4: Backend · 4x03

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Håkon Wium Lie y Bert Bos

1996

Eliminó las monstruosas tablas anidadas y gifs transparentes que los diseñadores usaban para alinear cajas.

Vol 3: Unix & Networks

3x34

Applatix / Intuit

2018

Su consola visual compara en tiempo real el árbol deseado de Git con el estado vivo del cluster de Kubernetes.

Vol 3: Unix & Networks

3x33

Gerald Combs

1998

Combs lo programó para resolver problemas de red en su empleo en un proveedor de Internet.

Vol 3: Unix & Networks

3x32

Google (Lars Bak)

2008

Anunciado con un cómic dibujado por Scott McCloud; su motor V8 obligó a la competencia a reinventar sus navegadores.

Vol 3: Unix & Networks

3x37

Nat Friedman (GitHub)

2019

GitHub igualó donaciones hasta $5,000 por usuario sin cobrar comisiones, impulsando la sostenibilidad del software libre.

Vol 3: Unix & Networks

3x36

IETF y proveedores CDN

2025

Cifra el campo SNI, impidiendo que proveedores de internet y censores estatales vean qué sitios específicos visitas.

Vol 3: Unix & Networks

3x35

Comunidad Mundial de Software Libre

2023

Comunidades del mundo le rindieron tributo; en su testamento legó fondos para continuar su causa caritativa en Uganda.

Vol 3: Unix & Networks

3x3A

Thiago de Arruda

2014

Nació tras el rechazo de parches de refactorización en Vim; Neovim revivió el fervor por editores de terminal modernos.

Vol 3: Unix & Networks

3x39

TC39 (Ecma International)

2009

Desbloqueó una década de estancamiento político en el comité TC39 tras el abandono definitivo del megalómano y divisivo proyecto ECMAScript 4.

Vol 3: Unix & Networks

3x38

Joe Wilm

2017

Escrito en Rust, aprovecha OpenGL para descargar el renderizado tipográfico directamente en la tarjeta gráfica.

Vol 3: Unix & Networks

3x3D

Tim Berners-Lee, R. Fielding y Frystyk

1996

Introdujo cabeceras HTTP, códigos de estado como 200 OK y 404 Not Found, y soporte de tipos MIME multimedia.

Vol 3: Unix & Networks

3x3C

Jesse James Garrett (Adaptive Path)

2005

Articuló cómo XHR, DOM y JS lograban experiencias fluidas de escritorio en la web, impulsado por Google Maps.

Vol 3: Unix & Networks

3x3B

Larry Ellison, B. Miner y E. Oates

1979

La llamaron "Versión 2" porque Ellison sabía que ninguna gran empresa compraría la v1.0 de un gestor de base de datos.

Vol 4: Backend

4x00

Red Hat Inc.

1999

Primera firma de software libre en NASDAQ, logrando la octava mayor ganancia de debut en la historia de Wall Street.

Vol 3: Unix & Networks

3x3F

Dmitriy Zaporozhets y Valery Sizov

2011

Creado en Ucrania porque su autor no tenía presupuesto para costear licencias de GitHub en su empleo de entonces.

Vol 3: Unix & Networks

3x3E

Red Hat

2015

Red Hat pagó más de $100 millones por una startup chica, consagrando la madurez comercial de Ansible y Python.

Vol 4: Backend

4x03

Mitchell Hashimoto y A. Dadgar

2014

Con HCL y el archivo terraform.tfstate, permitió versionar redes, discos y clusters en la nube como código puro.

Vol 4: Backend

4x02

Srini Srinivasan y Brian Bulkowski

2012

Originalmente llamada Citrusleaf, fue programada para saltarse el sistema de archivos tradicional y hablar directo al SSD.

Vol 4: Backend

4x01

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Publicación de la base de datos relacional columnar en memoria DuckDB, orientada al procesamiento analítico embebido (OLAP).

Vol 4: Backend
4x04

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Apertura pública de ClickHouse, base de datos relacional columnar para analítica masiva en tiempo real (OLAP).

Vol 4: Backend
4x05

CLOUD E INFRAESTRUCTURA
Automatización IaC

Publicación del paper del algoritmo de consenso distribuido Raft, diseñado explícitamente para ser comprensible frente a Paxos.

Vol 4: Backend
4x06

CLOUD E INFRAESTRUCTURA
Automatización IaC

Publicación de Paxos Made Simple, desglosando el clásico algoritmo de consenso distribuido en prosa matemática directa.

Vol 4: Backend
4x07

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento de Apache Cassandra en código abierto por parte de ingenieros de Facebook para su función de búsqueda de mensajes.

Vol 4: Backend
4x08

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Liberación de Apache Kafka, plataforma distribuida de streaming de eventos basada en un registro continuo (commit log).

Vol 4: Backend
4x09

CLOUD E INFRAESTRUCTURA
Automatización IaC

Formulación formal de la conjetura del Teorema CAP sobre consistencia, disponibilidad y tolerancia a particiones de red.

Vol 4: Backend
4x0A

CLOUD E INFRAESTRUCTURA
Automatización IaC

Publicación del paper sobre Hadoop Distributed File System (HDFS), democratizando el almacenamiento de big data en clústeres.

Vol 4: Backend
4x0B

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Publicación del paper de Spanner, la primera base de datos distribuida a escala global con consistencia externa y TrueTime.

Vol 4: Backend
4x0C

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Auge fulminante de pgvector, transformando a PostgreSQL en base de datos vectorial para IA generativa y pipelines RAG.

Vol 4: Backend
4x0D

CÓMPUTO E IA
Ecosistema Científico

Lanzamiento del motor de procesamiento masivo en memoria Apache Spark, superando los cuellos de botella de MapReduce.

Vol 4: Backend
4x0E

SISTEMAS Y REDES
Herramientas de Desarrollo

Publicación del lenguaje de consulta para APIs GraphQL, ofreciendo una alternativa declarativa y tipada a las APIs REST.

Vol 4: Backend
4x0F

SISTEMAS Y REDES
Estándares Web

Lanzamiento del servidor web asíncrono y reverse proxy Nginx, diseñado para resolver el problema de las 10,000 conexiones (C10k).

Vol 4: Backend
4x10

CLOUD E INFRAESTRUCTURA
Automatización IaC

Controversia por el cambio de licencia de Terraform en HashiCorp, sustituyendo el código abierto por la licencia BSL.

Vol 4: Backend
4x11

CLOUD E INFRAESTRUCTURA
Automatización IaC

Creación de Helm, el gestor de paquetes de aplicaciones para orquestar despliegues repetibles de Kubernetes mediante plantillas (charts).

Vol 4: Backend
4x12

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Presentación de Node.js en la conferencia JSConf EU, introduciendo un runtime de JavaScript asíncrono basado en V8.

Vol 4: Backend
4x13

LENGUAJES Y RUNTIMES
Frameworks Web

Lanzamiento del framework web Ruby on Rails, popularizando la filosofía de Convention over Configuration y la arquitectura MVC.

Vol 4: Backend
4x14

LENGUAJES Y RUNTIMES
Frameworks Web

Liberación del framework web Django, diseñado para construir aplicaciones de noticias rápidas bajo plazos de redacción exigentes.

Vol 4: Backend
4x15

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Diego Ongaro y John Ousterhout

2014

Fue presentado en USENIX ATC bajo el título 'In Search of an Understandable Consensus Algorithm'.

Vol 4: Backend

4x06

Yandex (Alexey Milovidov)

2016

Diseñado para analizar billones de clics web por segundo, pulverizó récords en analítica columnar masiva (OLAP).

Vol 4: Backend

4x05

Hannes Mühleisen y Mark Raasveldt

2019

Bautizada 'Duck' porque los patos son animales resistentes y sencillos, y un profesor del autor tenía un pato como mascota.

Vol 4: Backend

4x04

Jay Kreps, N. Narkhede y J. Rao

2011

Kreps eligió el nombre del autor literario Franz Kafka porque el software estaba obsesivamente optimizado para escribir datos a máxima velocidad.

Vol 4: Backend

4x09

Avinash Lakshman y P. Malik (Facebook)

2008

Unió el modelo tabular de Bigtable con nodos sin maestro central de Dynamo, logrando tolerancia a fallos total.

Vol 4: Backend

4x08

Leslie Lamport

2001

Lamport escribió este artículo tras quejarse de que casi nadie había entendido su publicación original de 1998.

Vol 4: Backend

4x07

James C. Corbett et al. (Google)

2012

Utilizaba receptores GPS y relojes atómicos dentro de los datacenters para acotar la incertidumbre temporal.

Vol 4: Backend

4x0C

Doug Cutting y Mike Cafarella

2010

Inspirado en el Google File System (GFS), fue bautizado con el nombre del elefante de juguete del hijo de Cutting.

Vol 4: Backend

4x0B

Eric Brewer (UC Berkeley)

2000

Fue probado formalmente como teorema matemático en 2002 por Seth Gilbert y Nancy Lynch del MIT.

Vol 4: Backend

4x0A

Lee Byron, Nick Schrock (Facebook)

2015

Había sido desarrollado internamente desde 2012 para alimentar la aplicación móvil de noticias de Facebook.

Vol 4: Backend

4x0F

Matei Zaharia (UC Berkeley AMPLab)

2014

Introdujo los RDD (Resilient Distributed Datasets) para ejecutar algoritmos iterativos hasta 100x más rápido.

Vol 4: Backend

4x0E

Andrew Kane

2023

Permitió almacenar y consultar embeddings de IA en Postgres con similitud coseno sin requerir bases vectoriales externas.

Vol 4: Backend

4x0D

Deis (Matt Butcher et al.)

2015

Nacido en un hackatón de Deis, se volvió el gestor oficial de paquetes de la CNCF para aplicaciones en Kubernetes.

Vol 4: Backend

4x12

HashiCorp

2023

Desató una rebelión en la industria y el nacimiento del fork libre OpenTofu bajo la Linux Foundation.

Vol 4: Backend

4x11

Igor Sysoev

2004

Sysoev lo programó para atender el gigantesco tráfico del portal ruso Rambler.ru.

Vol 4: Backend

4x10

Adrian Holovaty y Simon Willison

2005

Fue bautizado en homenaje al legendario virtuoso de la guitarra de jazz gitano Django Reinhardt.

Vol 4: Backend

4x15

David Heinemeier Hansson (DHH)

2004

Fue extraído de la base de código que DHH programó para la herramienta de gestión de proyectos Basecamp.

Vol 4: Backend

4x14

Ryan Dahl

2009

Dahl se inspiró para crearlo tras frustrarse viendo una barra de carga lenta de subida de fotos en Flickr.

Vol 4: Backend

4x13

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Publicación del conjunto de herramientas de resiliencia Simian Army y Chaos Monkey por parte de ingenieros de Netflix.

Vol 4: Backend
4x16

LENGUAJES Y RUNTIMES
Frameworks Web

Lanzamiento del microframework web en Python Flask, surgido originalmente como una broma del día de los inocentes (April Fools).

Vol 4: Backend
4x17

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Inicio del proyecto Postgres en la Universidad de California en Berkeley como el sucesor de nueva generación de Ingres.

Vol 4: Backend
4x18

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de Alibaba Cloud (Aliyun), iniciando la infraestructura de nube a hiperescala del sudeste asiático.

Vol 4: Backend
4x19

LENGUAJES Y RUNTIMES
Frameworks Web

Lanzamiento del framework web Phoenix sobre Elixir, aprovechando la máquina virtual BEAM para soportar millones de WebSockets.

Vol 4: Backend
4x1A

CLOUD E INFRAESTRUCTURA
Automatización IaC

Liberación del proxy de red de alto rendimiento Envoy Proxy, optimizado para arquitecturas de microservicios y service mesh.

Vol 4: Backend
4x1B

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento del motor de base de datos distribuida NewSQL TiDB, compatible con el protocolo MySQL y arquitectura Raft.

Vol 4: Backend
4x1C

SISTEMAS Y REDES
Estándares Web

Lanzamiento del balanceador de carga y acelerador de tráfico TCP/HTTP HAProxy de ultra-alta velocidad y estabilidad.

Vol 4: Backend
4x1D

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento de Meilisearch, motor de búsqueda ultrarrápido y de código abierto diseñado para búsqueda incremental al escribir.

Vol 4: Backend
4x1E

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento del motor de búsqueda y analítica distribuido Elasticsearch, desarrollado sobre la biblioteca Apache Lucene.

Vol 4: Backend
4x1F

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Presentación de CockroachDB, base de datos SQL distribuida transaccional resistente a caídas de servidores y regiones enteras.

Vol 4: Backend
4x20

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Presentación de Neon, introduciendo una arquitectura Serverless que desacopla el cómputo de la capa de almacenamiento en PostgreSQL.

Vol 4: Backend
4x21

SISTEMAS Y REDES
Herramientas de Desarrollo

Publicación del framework de llamadas a procedimiento remoto de alto rendimiento gRPC sobre HTTP/2 y Protocol Buffers.

Vol 4: Backend
4x22

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Fractura en el ecosistema de Redis tras el cambio de licencia de código abierto a licencias restrictivas de código disponible (RSALv2).

Vol 4: Backend
4x23

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Liberación de la biblioteca de almacenamiento embebido clave-valor de alto rendimiento RocksDB, derivada de LevelDB.

Vol 4: Backend
4x24

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento del intermediario de mensajes y colas distribuidas RabbitMQ, implementando el estándar abierto AMQP.

Vol 4: Backend
4x25

SISTEMAS Y REDES
Estándares Web

Presentación de la biblioteca de mensajería asíncrona sin broker ZeroMQ (ØMQ) para computación distribuida ultrarrápida.

Vol 4: Backend
4x26

CLOUD E INFRAESTRUCTURA
Automatización IaC

Publicación del algoritmo de consenso distribuido Epaxos (Egalitarian Paxos), eliminando el cuello de botella del líder.

Vol 4: Backend
4x27

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Michael Stonebraker (UC Berkeley)

1986

Significaba "Post-Ingres"; Stonebraker introdujo tipos complejos de usuario, disparadores y herencia de tablas.

Vol 4: Backend
4x18

Armin Ronacher (Pocoo)

2010

Ronacher empaquetó Werkzeug y Jinja en un solo archivo .zip para demostrar que se podía crear un framework en una tarde.

Vol 4: Backend
4x17

Netflix (Greg Orzell)

2011

Mataba servidores en producción aleatoriamente para forzar a ingenieros a diseñar arquitecturas inmunes a fallos.

Vol 4: Backend
4x16

Matt Klein (Lyft)

2016

Fue creado en Lyft para erradicar las fallas transitorias de red entre sus cientos de microservicios.

Vol 4: Backend
4x1B

Chris McCord

2014

Demostró en un famoso benchmark soportar 2 millones de conexiones de sockets simultáneas en un solo servidor.

Vol 4: Backend
4x1A

Alibaba Group (Jack Ma)

2009

Su plataforma 'Apsara' fue diseñada desde cero para soportar el masivo tráfico del 'Día de los Solteros' (11.11).

Vol 4: Backend
4x19

Quentin de Quelen y Clément Renault

2018

Escrito en Rust, ofrece búsqueda predictiva instantánea de errores tipográficos en menos de 50 milisegundos.

Vol 4: Backend
4x1E

Willy Tarreau

2001

Es célebre por su confiabilidad legendaria, corriendo meses o años sin consumir apenas CPU ni memoria.

Vol 4: Backend
4x1D

PingCAP (Max Liu, Ed Huang)

2015

Inspirado en el paper Spanner/F1 de Google, separa la capa de cómputo stateless de la capa de almacenamiento distribuido.

Vol 4: Backend
4x1C

Heikki Linnakangas y equipo Neon

2021

Permitió crear copias y ramas de bases de datos en microsegundos como en Git, revolucionando entornos de staging.

Vol 4: Backend
4x21

Spencer Kimball, Peter Mattis

2014

Sus creadores, ex ingenieros de Google, eligieron el nombre 'cucaracha' por su extrema capacidad de supervivencia.

Vol 4: Backend
4x20

Shay Banon

2010

El creador comenzó el proyecto previo (Compass) para ayudar a su esposa a buscar recetas de cocina.

Vol 4: Backend
4x1F

Facebook

2013

Optimizada para unidades SSD y memoria rápida, sirve como motor de almacenamiento en Kafka, MySQL y CockroachDB.

Vol 4: Backend
4x24

Redis Ltd. y Linux Foundation

2024

La comunidad y grandes empresas de nube respondieron de inmediato fundando la bifurcación Valkey bajo la tutela neutral de la Linux Foundation.

Vol 4: Backend
4x23

Google

2015

Evolucionó a partir de 'Stubby', el sistema interno de RPC con el que Google conectaba sus microservicios.

Vol 4: Backend
4x22

Iulian Moraru, David Andersen (CMU)

2013

Permite que cualquier réplica proponga comandos concurrentemente logrando consenso en 1 RTT sin pasar por un nodo maestro.

Vol 4: Backend
4x27

Pieter Hintjens y Martin Sustrik

2007

El 'cero' en el nombre representa cero latencia, cero costo de administración y cero necesidad de intermediarios.

Vol 4: Backend
4x26

Rabbit Technologies (Alexis Richardson)

2007

Escrito en Erlang para garantizar alta disponibilidad y tolerancia a fallos en el ruteo de eventos empresariales.

Vol 4: Backend
4x25

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento de etcd, el almacén distribuido de clave-valor fuertemente consistente que actúa como cerebro de Kubernetes.

Vol 4: Backend
4x28

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento de RethinkDB, la base de datos distribuida JSON diseñada para empujar cambios en tiempo real a las aplicaciones.

Vol 4: Backend
4x29

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Presentación de la base de datos orientada a columnas distribuida para analítica en tiempo real Apache Druid.

Vol 4: Backend
4x2A

SISTEMAS Y REDES
Estándares Web

Publicación de la versión 1.0 del servidor web Apache HTTP Server, dominando la infraestructura de la emergente World Wide Web.

Vol 4: Backend
4x2B

SISTEMAS Y REDES
Herramientas de Desarrollo

Publicación de la biblioteca de serialización eficiente de datos estructurados binarios Protocol Buffers (protobuf).

Vol 4: Backend
4x2C

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de la plataforma de streaming y procesamiento de eventos continuos en tiempo real Apache Flink.

Vol 4: Backend
4x2D

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento del servidor de almacenamiento de objetos distribuido compatible con S3 MinIO, diseñado para nubes privadas.

Vol 4: Backend
4x2E

LENGUAJES Y RUNTIMES
Lenguajes de Programación

Lanzamiento de la biblioteca de comunicación entre procesos y llamadas asíncronas ZeroMQ implementando sockets inteligentes.

Vol 4: Backend
4x2F

CLOUD E INFRAESTRUCTURA
Automatización IaC

Publicación de la biblioteca de tolerancia a fallos y cortocircuito en llamadas remotas de microservicios Hystrix.

Vol 4: Backend
4x30

CLOUD E INFRAESTRUCTURA
Automatización IaC

Creación de ZooKeeper, el servicio centralizado para coordinar información de estado y bloqueos en clústeres distribuidos.

Vol 4: Backend
4x31

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento de la base de datos distribuida en grafos Neo4j, optimizando el almacenamiento y consultas de relaciones complejas.

Vol 4: Backend
4x32

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Despliegue de los procesadores personalizados con arquitectura ARM AWS Graviton2 en centros de datos a gran escala.

Vol 4: Backend
4x33

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Paper de Bigtable, describiendo la base de datos distribuida de Google para almacenar petabytes no estructurados.

Vol 4: Backend
4x34

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento de ScyllaDB, una reimplementación en C++ de ultra-baja latencia compatible con Apache Cassandra.

Vol 4: Backend
4x35

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Creación del lenguaje SEQUEL (posteriormente acortado a SQL) en el laboratorio de investigación de IBM en San José.

Vol 4: Backend
4x36

CLOUD E INFRAESTRUCTURA
Automatización IaC

Publicación del artículo que definió el modelo de ordenación mediante Relojes Vectoriales en sistemas distribuidos.

Vol 4: Backend
4x37

SISTEMAS Y REDES
Herramientas de Desarrollo

Creación del runtime seguro de JavaScript y TypeScript Deno, remediando las decisiones de diseño del creador en Node.js.

Vol 4: Backend
4x38

LENGUAJES Y RUNTIMES
Frameworks Web

Publicación de la especificación de diseño arquitectónico REST (Representational State Transfer) en una tesis doctoral.

Vol 4: Backend
4x39

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Eric Tschetter, Fangjin Yang (Metamarkets)

2012

Nació para solucionar la lentitud al agregar billones de impresiones de subastas de publicidad online en tiempo real.

Vol 4: Backend · 4x2A

Slava Akhmechet y Michael Glukhovsky

2012

Introdujo la innovadora arquitectura Changefeeds para evitar que los clientes hicieran sondeo de datos continuamente.

Vol 4: Backend · 4x29

CoreOS (Brandon Philips)

2013

Su nombre proviene de configurar la carpeta Unix '/etc' como un servicio 'distribuido' (d).

Vol 4: Backend · 4x28

Apache Software Foundation (TU Berlin)

2014

El proyecto original se llamaba Stratosphere y adoptó una ardilla marrón rojiza como su simpático logotipo.

Vol 4: Backend · 4x2D

Google (Kenton Varda)

2008

Reemplazó el tráfico XML redundante en las comunicaciones entre servidores de toda la infraestructura de Google.

Vol 4: Backend · 4x2C

Robert McCool y Apache Group

1995

Nació como un conjunto de parches ('a patchy server') sobre el demonio de servidor web NCSA HTTPd.

Vol 4: Backend · 4x2B

Netflix Engineering

2012

Introdujo el patrón 'Circuit Breaker' en la industria para prevenir fallos en cascada en horas pico de streaming.

Vol 4: Backend · 4x30

Pieter Hintjens y Martin Sustrik

2007

Permitió construir topologías distribuidas complejas (pub-sub, push-pull) sin necesidad de configurar un broker central.

Vol 4: Backend · 4x2F

Anand Babu Periasamy (AB)

2014

El creador ya había cofundado previamente el popular sistema de archivos distribuido GlusterFS.

Vol 4: Backend · 4x2E

Amazon Web Services (Annapurna Labs)

2020

Logró hasta 40% más rendimiento por vatio que chips Intel/AMD tradicionales, acelerando la nube hacia chips ARM.

Vol 4: Backend · 4x33

Emil Eifrem y Johan Svensson

2007

Popularizó las consultas de patrones con el lenguaje Cypher emulando cláusulas visuales de grafos.

Vol 4: Backend · 4x32

Yahoo! Research (Benjamin Reed)

2008

Se llamó 'cuidador del zoológico' porque coordinaba proyectos con nombres de animales como Hadoop, Pig y Chukwa.

Vol 4: Backend · 4x31

Donald Chamberlin y Raymond Boyce

1974

Cambiaron SEQUEL a SQL porque la sigla original era una marca registrada de una fábrica de aviones británica.

Vol 4: Backend · 4x36

Dor Laor y Avi Kivity

2015

Aprovechó el framework de hilos Seastar y la experiencia de sus creadores diseñando el hipervisor KVM.

Vol 4: Backend · 4x35

Google (Jeff Dean, S. Ghemawat et al.)

2006

El paper de Google inspiró a Silicon Valley, dando origen a proyectos libres como Cassandra y HBase en Big Data.

Vol 4: Backend · 4x34

Roy Fielding (UC Irvine)

2000

Fielding era también uno de los principales autores de las especificaciones de HTTP/1.0 y HTTP/1.1.

Vol 4: Backend · 4x39

Ryan Dahl

2018

Dahl presentó el proyecto durante su famosa conferencia '10 Things I Regret About Node.js'.

Vol 4: Backend · 4x38

Colin Fidge y Friedemann Mattern

1988

Permitió rastrear la causalidad exacta de eventos concurrentes sin recurrir a un reloj físico sincronizado.

Vol 4: Backend · 4x37

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Inicio del movimiento de salida de la nube (Cloud Exit) liderado por la empresa 37signals tras repatriar sus servidores al metal propio.

Vol 4: Backend
4x3A

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Nacimiento de DuckDB, concebido como el equivalente a SQLite para analítica en memoria y archivos Parquet.

Vol 4: Backend
4x3B

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento del acelerador y caché de contenido web HTTP Varnish Cache, optimizado para la memoria virtual del kernel.

Vol 4: Backend
4x3C

SISTEMAS Y REDES
Estándares Web

Publicación del RFC 6455 estandarizando el protocolo bidireccional en tiempo real WebSocket para navegadores y servidores.

Vol 4: Backend
4x3D

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento del servidor de aplicaciones de mensajería asíncrona de alto rendimiento NATS, diseñado en Go para microservicios.

Vol 4: Backend
4x3E

ALMACENAMIENTO Y DATOS
Motores de Bases de Datos

Lanzamiento de BadgerDB, motor de base de datos embebida clave-valor en Go optimizado para unidades de estado sólido SSD.

Vol 4: Backend
4x3F

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Lanzamiento comercial de Amazon S3 (Simple Storage Service), inaugurando la era de la computación en la nube moderna de AWS.

Vol 5: Cloud & SRE
5x00

CLOUD E INFRAESTRUCTURA
Automatización IaC

Anuncio formal de la adquisición de HashiCorp por parte de IBM en una operación valorada en $6,400 millones de dólares.

Vol 5: Cloud & SRE
5x01

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Creación de OpenStack como iniciativa de código abierto en Python impulsada conjuntamente por la NASA y Rackspace Hosting.

Vol 5: Cloud & SRE
5x02

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Lanzamiento comercial de Google Compute Engine (GCE), permitiendo ejecutar máquinas virtuales a escala global en la nube de Google.

Vol 5: Cloud & SRE
5x03

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Lanzamiento de Azure Functions, la plataforma de computación sin servidor (Serverless) orientada a eventos de Microsoft.

Vol 5: Cloud & SRE
5x04

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Extracción y publicación de containerd como un daemon de ejecución de contenedores desacoplado e independiente por parte de Docker.

Vol 5: Cloud & SRE
5x05

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Lanzamiento de Google App Engine, la primera oferta comercial de Google Cloud que permitía desplegar apps Python en su nube.

Vol 5: Cloud & SRE
5x06

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Anuncio formal de Windows Azure (luego Microsoft Azure) durante la conferencia PDC en Los Ángeles.

Vol 5: Cloud & SRE
5x07

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Lanzamiento de la plataforma en la nube orientada a frontend y funciones serverless edge Vercel (originalmente Zeit Now).

Vol 5: Cloud & SRE
5x08

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Lanzamiento de la plataforma en la nube Fly.io, permitiendo correr contenedores Docker completos pegados al usuario en el Edge.

Vol 5: Cloud & SRE
5x09

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento de LXC (Linux Containers) 1.0, la primera versión estable de contenedores a nivel de sistema operativo para Linux.

Vol 5: Cloud & SRE
5x0A

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de GitHub Actions, integrando CI/CD y flujos de trabajo automatizados directo en los repositorios.

Vol 5: Cloud & SRE
5x0B

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Poul-Henning Kamp

2006

Diseñado en Noruega, delegó la paginación al propio sistema operativo descartando gestionar cachés en espacio de usuario.

Vol 4: Backend · 4x3C

Hannes Mühleisen y Mark Raasveldt (CWI)

2019

Permite consultar gigabytes de archivos de datos locales directamente desde Python o R con consultas vectorizadas sin configurar clusters pesados.

Vol 4: Backend · 4x3B

David Heinemeier Hansson (DHH)

2023

DHH mostró cómo salir de AWS y comprar servidores propios ahorró a Basecamp $3.2 millones de dólares en 5 años.

Vol 4: Backend · 4x3A

Manish Jain (Dgraph)

2017

Implementó la arquitectura WiscKey separando las claves del árbol LSM de los valores guardados en el disco.

Vol 4: Backend · 4x3F

Derek Collison (Apcera)

2011

Collison, ex arquitecto de messaging en TIBCO, lo diseñó bajo la premisa de simplicidad extrema y latencias de microsegundos.

Vol 4: Backend · 4x3E

Ian Fette y Alexey Melnikov

2011

Permitió reemplazar las ineficientes técnicas de 'long-polling' por una única conexión TCP persistente.

Vol 4: Backend · 4x3D

NASA y Rackspace

2010

NASA y Rackspace unieron sus tecnologías soñando con forjar el estándar abierto de la nube pública con OpenStack.

Vol 5: Cloud & SRE · 5x02

IBM (Arvind Krishna)

2024

Reflejó la necesidad de IBM de controlar la plataforma líder de infraestructura como código en centros de datos.

Vol 5: Cloud & SRE · 5x01

Amazon Web Services (Andy Jassy)

2006

Ofrecía almacenamiento duradero por 15 centavos de dólar por gigabyte al mes mediante una simple API REST.

Vol 5: Cloud & SRE · 5x00

Docker Inc.

2017

Gestiona imágenes y supervisión de procesos en bajo nivel sin la sobrecarga comercial ni la interfaz CLI de Docker.

Vol 5: Cloud & SRE · 5x05

Microsoft Azure

2016

Permitió integrar funciones serverless directamente con el ecosistema empresarial de Office 365 y Active Directory.

Vol 5: Cloud & SRE · 5x04

Google Cloud

2013

Presentado por Urs Hölzle, destacó por la rapidez de arranque de sus VMs y facturación por minutos.

Vol 5: Cloud & SRE · 5x03

Guillermo Rauch

2016

Permitió desplegar aplicaciones web globales con certificado SSL en segundos usando un único comando now en terminal.

Vol 5: Cloud & SRE · 5x08

Microsoft (Ray Ozzie)

2008

El proyecto fue desarrollado bajo el nombre en clave secreto 'Project Red Dog'.

Vol 5: Cloud & SRE · 5x07

Google

2008

Fue el pionero de los modelos PaaS modernos, mucho antes del surgimiento de Docker y Kubernetes.

Vol 5: Cloud & SRE · 5x06

GitHub (Nat Friedman)

2019

Reemplazó viejos servidores Jenkins, permitiendo probar y desplegar software con workflows en YAML en el repo.

Vol 5: Cloud & SRE · 5x0B

Stéphane Graber y Serge Hallyn

2014

Sirvió como el motor de ejecución inicial sobre el cual Solomon Hykes construyó las primeras versiones de Docker.

Vol 5: Cloud & SRE · 5x0A

Kurt Mackey y Jerome Leclanche

2020

Convierte automáticamente contenedores Docker estándar en microVMs de Firecracker distribuidas en todo el mundo.

Vol 5: Cloud & SRE · 5x09

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Publicación del artículo académico sobre Borg, el sistema secreto de gestión de clústeres a gran escala de Google.

Vol 5: Cloud & SRE
5x0C

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento de CoreOS, un sistema operativo Linux mínimo diseñado específicamente para hospedar contenedores en clúster.

Vol 5: Cloud & SRE
5x0D

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Creación de la Open Container Initiative (OCI) por Docker, CoreOS y gigantes de la industria para estandarizar formatos de imagen.

Vol 5: Cloud & SRE
5x0E

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento de Kata Containers, integrando la velocidad de los contenedores con el aislamiento de seguridad de máquinas virtuales.

Vol 5: Cloud & SRE
5x0F

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento de Podman 1.0, permitiendo gestionar pods y contenedores OCI compatibles con Docker sin proceso daemon ni privilegios root.

Vol 5: Cloud & SRE
5x10

SISTEMAS Y REDES
Sistemas UNIX / Linux

Incorporación de Control Groups (cgroups) en el kernel de Linux 2.6.24, la base para aislar recursos en contenedores.

Vol 5: Cloud & SRE
5x11

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento de rkt (Rocket) por CoreOS como alternativa orientada a seguridad frente al modelo monolítico de Docker.

Vol 5: Cloud & SRE
5x12

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Publicación del hipervisor de código abierto Xen en el simposio ACM SOSP, popularizando la paravirtualización.

Vol 5: Cloud & SRE
5x13

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de Vagrant, simplificando la creación y gestión de entornos de desarrollo virtualizados reproducibles.

Vol 5: Cloud & SRE
5x14

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Popularización de los despliegues web instantáneos con enlaces de vista previa por rama mediante la plataforma Vercel.

Vol 5: Cloud & SRE
5x15

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de Packer, una herramienta de código abierto para crear imágenes idénticas de máquinas virtuales multi-plataforma.

Vol 5: Cloud & SRE
5x16

CLOUD E INFRAESTRUCTURA
Automatización IaC

Debut de Ansible, un sistema de automatización en Python sin agentes (agentless) que orquesta servidores mediante conexiones seguras SSH.

Vol 5: Cloud & SRE
5x17

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de Consul, plataforma de descubrimiento de servicios, mallas de servicios y configuración distribuida.

Vol 5: Cloud & SRE
5x18

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento del orquestador de cargas de trabajo flexible Nomad, capaz de programar contenedores y binarios tradicionales.

Vol 5: Cloud & SRE
5x19

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de Bicep, el lenguaje declarativo específico de dominio (DSL) para desplegar infraestructura en Azure de forma limpia.

Vol 5: Cloud & SRE
5x1A

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de Puppet, herramienta pionera de gestión de configuración declarativa para administradores de sistemas.

Vol 5: Cloud & SRE
5x1B

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento del gestor de configuración en Ruby Chef, tratando a la infraestructura enteramente como código en recetas.

Vol 5: Cloud & SRE
5x1C

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de SaltStack (Salt), sistema de ejecución remota de alta velocidad basado en colas ZeroMQ y Python.

Vol 5: Cloud & SRE
5x1D

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

The Linux Foundation

2015

Evitó una fragmentación irreversible en la industria tras la guerra comercial entre Docker y el motor rkt de CoreOS.

Vol 5: Cloud & SRE
5x0E

Alex Polvi y Brandon Philips

2013

Implementó actualizaciones automáticas particionadas activas/inactivas inspiradas en el esquema de ChromeOS.

Vol 5: Cloud & SRE
5x0D

Abhishek Verma, John Wilkes et al.

2015

Durante más de una década gestionó millones de tareas de Gmail, Search y YouTube antes de inspirar Kubernetes.

Vol 5: Cloud & SRE
5x0C

Paul Menage y Rohit Seth (Google)

2008

Originalmente se llamaba 'Process Containers', pero cambiaron el nombre para evitar confusiones semánticas.

Vol 5: Cloud & SRE
5x11

Red Hat (Dan Walsh)

2019

Popularizó el alias alias docker=podman gracias a su compatibilidad directa comando a comando.

Vol 5: Cloud & SRE
5x10

OpenStack Foundation / Intel

2017

Fusionó las tecnologías Clear Containers de Intel y RunV de Hyper.sh para ejecutar cada pod en un hipervisor ultraligero.

Vol 5: Cloud & SRE
5x0F

Mitchell Hashimoto

2010

Comenzó como un proyecto de fines de semana mientras Mitchell trabajaba como consultor de software.

Vol 5: Cloud & SRE
5x14

Universidad de Cambridge (Ian Pratt)

2003

Fue el motor principal que impulsó la nube pública de Amazon EC2 en sus primeros diez años de vida.

Vol 5: Cloud & SRE
5x13

CoreOS (Alex Polvi)

2014

Prescindía de un proceso demonio central con privilegios de root, impulsando la creación del estándar OCI.

Vol 5: Cloud & SRE
5x12

Michael DeHaan

2012

Usaba playbooks en YAML y conexión por SSH sin agentes cliente, arrasando en adopción frente a Puppet y Chef.

Vol 5: Cloud & SRE
5x17

HashiCorp

2013

Permitió generar AMIs de AWS, imágenes de Docker y discos de VirtualBox a partir de una única plantilla JSON.

Vol 5: Cloud & SRE
5x16

Guillermo Rauch (Vercel)

2018

Creador de Socket.io y Next.js, Rauch unió cada commit en Git con despliegues edge globales instantáneos.

Vol 5: Cloud & SRE
5x15

Microsoft

2020

Se diseñó para librar a los ingenieros de Azure de la tediosa y verbosa sintaxis de las plantillas ARM en formato JSON.

Vol 5: Cloud & SRE
5x1A

HashiCorp

2015

Diseñado como un único ejecutable sin dependencias externas, capaz de desplegar 1 millón de contenedores en 2 minutos.

Vol 5: Cloud & SRE
5x19

HashiCorp (Armon Dadgar)

2014

Combina el protocolo de chismes Serf con el consenso Raft para ofrecer alta disponibilidad en datacenters.

Vol 5: Cloud & SRE
5x18

Thomas Hatch

2011

Se diseñó para comunicarse con decenas de miles de servidores en milisegundos usando un bus de eventos persistente.

Vol 5: Cloud & SRE
5x1D

Adam Jacob y Jesse Robbins

2009

Adoptó metáforas culinarias en todo su diseño: recetas, libros de cocina (cookbooks) y cuchillos (knife).

Vol 5: Cloud & SRE
5x1C

Luke Kanies

2005

Kanies lo escribió motivado por la rigidez y falta de mantenimiento que sufría al administrar entornos con CFEngine.

Vol 5: Cloud & SRE
5x1B

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Publicación del proyecto Apache Mesos para abstracción y reparto fino de recursos en datacenters a gran escala.

Vol 5: Cloud & SRE
5x1E

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de Helm v3, eliminando por completo el controversial componente servidor Tiller por seguridad en Kubernetes.

Vol 5: Cloud & SRE
5x1F

CLOUD E INFRAESTRUCTURA
Automatización IaC

Anuncio de la malla de servicios Istio, facilitando balanceo inteligente, telemetría y cifrado mTLS entre microservicios.

Vol 5: Cloud & SRE
5x20

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de Linkerd, la primera malla de servicios (service mesh) de la industria, acuñando formalmente el término.

Vol 5: Cloud & SRE
5x21

CLOUD E INFRAESTRUCTURA
Automatización IaC

Creación del proyecto de observabilidad OpenTelemetry (OTel), fusionando los proyectos OpenTracing y OpenCensus.

Vol 5: Cloud & SRE
5x22

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Nacimiento de Heroku, inventando el paradigma de plataforma como servicio (PaaS) con el comando mágico git push heroku master.

Vol 5: Cloud & SRE
5x23

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Presentación de AWS Lambda en la conferencia re:Invent, inaugurando la era de la computación Serverless (FaaS).

Vol 5: Cloud & SRE
5x24

CLOUD E INFRAESTRUCTURA
Automatización IaC

Liberación de Jaeger, plataforma de rastreo distribuido de transacciones entre microservicios inspirada en Dapper de Google.

Vol 5: Cloud & SRE
5x25

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Despliegue de superclusters de entrenamiento de inteligencia artificial con cientos de miles de GPUs interconectadas por redes InfiniBand.

Vol 5: Cloud & SRE
5x26

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de la plataforma de tableros y analítica visual de métricas en series temporales Grafana.

Vol 5: Cloud & SRE
5x27

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de LitmusChaos, marco nativo de ingeniería de caos para inyectar fallos de red y pod en clústeres de Kubernetes.

Vol 5: Cloud & SRE
5x28

CLOUD E INFRAESTRUCTURA
Automatización IaC

Publicación del libro canónico Site Reliability Engineering (SRE), revelando cómo gestiona Google sus sistemas de producción.

Vol 5: Cloud & SRE
5x29

CULTURA HACKER Y WEB
Plataformas de Colaboración

Publicación de la influyente novela de gestión técnica The Phoenix Project, popularizando la cultura y metodologías DevOps.

Vol 5: Cloud & SRE
5x2A

CLOUD E INFRAESTRUCTURA
Automatización IaC

Acuñación del término y paradigma GitOps, usando Git como única fuente de verdad para la infraestructura declarativa.

Vol 5: Cloud & SRE
5x2B

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de la plataforma de infraestructura como código Pulumi, permitiendo usar TypeScript, Python o Go en vez de YAML.

Vol 5: Cloud & SRE
5x2C

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Lanzamiento de Cloudflare Workers, ejecutando funciones serverless en el borde (Edge Computing) mediante V8 Isolates.

Vol 5: Cloud & SRE
5x2D

SISTEMAS Y REDES
Estándares Web

Lanzamiento de Tailscale, red privada virtual en malla basada en WireGuard que no requiere abrir puertos ni configurar firewalls.

Vol 5: Cloud & SRE
5x2E

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento del framework de computación sin servidor en Kubernetes Knative, unificando el ecosistema cloud-native.

Vol 5: Cloud & SRE
5x2F

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Google, IBM y Lyft

2017

Aprovechó el proxy Envoy como plano de datos colocándolo como contenedor 'sidecar' junto a cada aplicación.

Vol 5: Cloud & SRE

5x20

Helm Community / CNCF

2019

La eliminación de Tiller simplificó el modelo de seguridad apoyándose directamente en el RBAC nativo del clúster.

Vol 5: Cloud & SRE

5x1F

Benjamin Hindman et al. (UC Berkeley)

2011

Twitter adoptó Mesos para abandonar su infame pantalla de ballena caída ('Fail Whale') y estabilizar su plataforma.

Vol 5: Cloud & SRE

5x1E

James Lindenbaum et al. (Heroku)

2007

Permitía a los programadores de Ruby on Rails poner una aplicación en producción en diez segundos sin configurar servidores Linux ni servidores web.

Vol 5: Cloud & SRE

5x23

CNCF

2019

Se convirtió rápidamente en el segundo proyecto más activo de toda la fundación CNCF, sólo detrás de Kubernetes.

Vol 5: Cloud & SRE

5x22

William Morgan y Oliver Gould (Buoyant)

2016

Inspirado en la biblioteca interna Finagle que ambos ingenieros ayudaron a construir en Twitter.

Vol 5: Cloud & SRE

5x21

NVIDIA, Microsoft Azure y AWS

2024

Las nubes pasaron de alquilar CPUs a despachar clusters masivos de miles de GPUs para entrenar modelos de IA.

Vol 5: Cloud & SRE

5x26

Yuri Shkuro (Uber Technologies)

2015

Uber lo desarrolló internamente cuando pasaron de un monolito en Python a miles de microservicios distribuidos.

Vol 5: Cloud & SRE

5x25

AWS (Andy Jassy y Tim Wagner)

2014

Permitió ejecutar funciones de código disparadas por eventos cobrando únicamente por los milisegundos exactos de procesamiento consumidos.

Vol 5: Cloud & SRE

5x24

Betsy Beyer, Chris Jones et al. (Google)

2016

Popularizó principios revolucionarios como Error Budgets, eliminación de 'toil' y la doctrina 'Hope is not a strategy'.

Vol 5: Cloud & SRE

5x29

MayaData / CNCF

2018

Permite definir experimentos de caos directamente como recursos personalizados CRD de Kubernetes.

Vol 5: Cloud & SRE

5x28

Torkel Ödegaard

2014

Nació originalmente como una bifurcación de la interfaz Kibana enfocada específicamente en métricas numéricas.

Vol 5: Cloud & SRE

5x27

Joe Duffy y Eric Rudder

2018

Permite usar bucles, clases y pruebas unitarias de software reales para instanciar recursos de nube.

Vol 5: Cloud & SRE

5x2C

Alexis Richardson (Weaveworks)

2017

El concepto demostró que la convergencia continua de software es más segura que lanzar scripts desde pipelines CI.

Vol 5: Cloud & SRE

5x2B

Gene Kim, Kevin Behr y George Spafford

2013

Adaptó la teoría de restricciones de la manufactura industrial ('The Goal') al caos de los departamentos de TI.

Vol 5: Cloud & SRE

5x2A

Google, Pivotal, Red Hat

2018

Se dividió en dos componentes centrales: Knative Serving (escalado a cero) y Knative Eventing.

Vol 5: Cloud & SRE

5x2F

Avery Pennarun, Brad Fitzpatrick et al.

2020

Aprovechó técnicas avanzadas de NAT traversal y el protocolo DERP para conectar máquinas directamente punto a punto.

Vol 5: Cloud & SRE

5x2E

Kenton Varda (Cloudflare)

2017

Eliminó el arranque en frío (cold start) al prescindir de contenedores y usar aislamiento nativo de memoria en V8.

Vol 5: Cloud & SRE

5x2D

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Caída catastrófica de la región US-East-1 (Norte de Virginia) de AWS, tumbando servicios de bancos, aerolíneas y streaming mundiales.

Vol 5: Cloud & SRE
5x30

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento de LXC (Linux Containers), combinando los namespaces de Linux y cgroups en una interfaz coherente en espacio de usuario.

Vol 5: Cloud & SRE
5x31

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento de K3s, una distribución ligera de Kubernetes en un solo binario optimizada para IoT y entornos Edge.

Vol 5: Cloud & SRE
5x32

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento del runtime de microVM ultrarrápidas y seguras Firecracker, optimizado para cargas Serverless y contenedores.

Vol 5: Cloud & SRE
5x33

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de Crossplane, permitiendo orquestar infraestructura de nube multi-proveedor utilizando directamente APIs de Kubernetes.

Vol 5: Cloud & SRE
5x34

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Apertura de la beta limitada de Amazon EC2 (Elastic Compute Cloud), permitiendo alquilar máquinas virtuales por horas.

Vol 5: Cloud & SRE
5x35

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento de Kind (Kubernetes in Docker), herramienta para ejecutar clústeres locales de Kubernetes usando nodos en contenedores.

Vol 5: Cloud & SRE
5x36

CLOUD E INFRAESTRUCTURA
Automatización IaC

Creación del proyecto de agregación de registros ligero Grafana Loki, optimizado para indexar solo metadatos y etiquetas.

Vol 5: Cloud & SRE
5x37

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de Cilium, proporcionando redes, seguridad y observabilidad de alto rendimiento en Kubernetes utilizando eBPF.

Vol 5: Cloud & SRE
5x38

CLOUD E INFRAESTRUCTURA
Contenedores y Virtualización

Lanzamiento de gVisor, un sandbox de seguridad a nivel de usuario que intercepta llamadas al sistema para aislar contenedores.

Vol 5: Cloud & SRE
5x39

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de KEDA (Kubernetes Event-driven Autoscaling), permitiendo escalar pods a cero y hasta miles según colas de eventos.

Vol 5: Cloud & SRE
5x3A

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de Thanos, añadiendo almacenamiento de largo plazo de alta disponibilidad y consultas globales a Prometheus.

Vol 5: Cloud & SRE
5x3B

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento de la plataforma de observabilidad de infraestructura y aplicaciones SaaS basada en la nube Datadog.

Vol 5: Cloud & SRE
5x3C

CLOUD E INFRAESTRUCTURA
Automatización IaC

Lanzamiento del motor de políticas declarativas para la nube nativa Open Policy Agent (OPA).

Vol 5: Cloud & SRE
5x3D

CLOUD E INFRAESTRUCTURA
Nube y Serverless

Apertura de las primeras interfaces web gratuitas de Amazon.com para programadores externos mediante llamadas XML y SOAP.

Vol 5: Cloud & SRE
5x3E

CLOUD E INFRAESTRUCTURA
Automatización IaC

Publicación del primer informe State of DevOps Report de DORA, estableciendo métricas científicas de rendimiento en ingeniería.

Vol 5: Cloud & SRE
5x3F

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Delimitación de parámetros posicionales estrictos mediante el símbolo / con la especificación del PEP 570.

Vol 6: Python Track
6x00

CULTURA HACKER Y WEB
Comunidad Python

Ratificación del PEP 13, reemplazando la monarquía benevolente de Guido por un Steering Council de 5 personas elegidas democráticamente.

Vol 6: Python Track
6x01

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Rancher Labs (Darren Shepherd)

2019

Reemplazó el pesado almacén etcd por SQLite embebido y redujo el consumo de RAM a menos de 512 MB.

Vol 5: Cloud & SRE · 5x32

Daniel Lezcano y Serge Hallyn

2008

Fue la primera implementación de contenedores integral y madura en Linux sin requerir parches experimentales fuera del kernel estándar.

Vol 5: Cloud & SRE · 5x31

Amazon Web Services

2021

Evidenció la enorme centralización de la web en el norte de Virginia (us-east-1), reabriendo el debate antimonopolio.

Vol 5: Cloud & SRE · 5x30

Amazon Web Services (Chris Pinkham)

2006

Sus primeras máquinas virtuales (m1.small) fueron desarrolladas en secreto en Sudáfrica sobre hipervisores Xen.

Vol 5: Cloud & SRE · 5x35

Upbound (Bassam Tabbara)

2018

Convierte al plano de control de Kubernetes en un orquestador universal de bases de datos y redes de AWS y GCP.

Vol 5: Cloud & SRE · 5x34

Amazon Web Services

2018

Escrito en Rust sobre KVM, es la tecnología subyacente que arranca funciones en AWS Lambda en milisegundos.

Vol 5: Cloud & SRE · 5x33

Thomas Graf y Dan Wendlandt (Isovalent)

2016

Reemplazó las complejas tablas de iptables por programas BPF inyectados dinámicamente en el kernel de Linux.

Vol 5: Cloud & SRE · 5x38

Grafana Labs

2018

Diseñado bajo la filosofía 'como Prometheus pero para logs', prescindiendo de los pesados índices de texto completo.

Vol 5: Cloud & SRE · 5x37

Ben Moss y James Munnelly (Kubernetes SIGs)

2018

Se convirtió en la herramienta fundamental para pruebas automatizadas e integración continua del propio Kubernetes.

Vol 5: Cloud & SRE · 5x36

Bartek Plotka y Fabian Reinartz (Improbable)

2018

Permitió retener métricas de monitoreo durante años almacenando bloques comprimidos en cubos de Amazon S3.

Vol 5: Cloud & SRE · 5x3B

Microsoft y Red Hat

2019

Añadió escalado reactivo basado en métricas externas como colas de RabbitMQ, Kafka o Azure Service Bus.

Vol 5: Cloud & SRE · 5x3A

Google

2018

Implementa un kernel sustituto escrito en Go ('Sentry') para evitar que un contenedor comprometido toque el host.

Vol 5: Cloud & SRE · 5x39

Amazon (Colin Bryar)

2002

Surgió tras el mandato de Bezos exigiendo a cada equipo exponer sus datos vía APIs desacopladas bajo pena de despido.

Vol 5: Cloud & SRE · 5x3E

Styra (Tim Hinrichs y Torin Sandall)

2016

Utiliza el lenguaje declarativo Rego para unificar la aplicación de políticas de seguridad en toda la pila tecnológica.

Vol 5: Cloud & SRE · 5x3D

Olivier Pomel y Alexis Lê-Quôc

2010

Nació de la frustración entre los equipos de desarrollo y operaciones para acordar métricas comunes de salud.

Vol 5: Cloud & SRE · 5x3C

Brett Cannon, Guido van Rossum et al.

2018

Fijó un mecanismo constitucional para evitar vacíos de poder, garantizando que Python evolucione sin personalismos.

Vol 6: Python Track · 6x01

Larry Hastings y Pablo Galindo Salgado

2018

Permitió diseñar APIs seguras donde los nombres de los argumentos internos pueden cambiarse en el futuro sin romper el código de los consumidores.

Vol 6: Python Track · 6x00

Nicole Forsgren, Gene Kim, Jez Humble

2014

Definió las cuatro métricas clave (lead time, frecuencia de despliegue, MTTR y tasa de fallo de cambios).

Vol 5: Cloud & SRE · 5x3F

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Estandarización de genéricos de tipado en colecciones nativas (list[str], dict[str, int]) sin importar typing vía PEP 585.

Vol 6: Python Track
6x02

CULTURA HACKER Y WEB
Comunidad Python

Lanzamiento del primer taller de Django Girls en la conferencia EuroPython en Berlín por dos programadoras polacas.

Vol 6: Python Track
6x03

LENGUAJES Y RUNTIMES
Ecosistema Rust

Creación de Maturin, la herramienta de construcción que permite compilar y empaquetar módulos de Python implementados en Rust (PyO3).

Vol 6: Python Track
6x04

LENGUAJES Y RUNTIMES
CPython Runtime

Despliegue de optimizaciones de desvirtualización de llamadas y análisis de escape en el compilador JIT experimental de Python 3.14.

Vol 6: Python Track
6x05

CULTURA HACKER Y WEB
Comunidad Python

Institucionalización de los Development Sprints presenciales de varios días tras finalizar las charlas principales de PyCon.

Vol 6: Python Track
6x06

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Implementación de ámbitos léxicos anidados (nested lexical scopes) resolviendo la regla clásica LEGB mediante el PEP 227.

Vol 6: Python Track
6x07

LENGUAJES Y RUNTIMES
CPython Runtime

Aprobación histórica de la propuesta Free-threaded CPython para eliminar el GIL del núcleo del lenguaje mediante el PEP 703.

Vol 6: Python Track
6x08

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Establecimiento de la jerarquía numérica abstracta (Number, Real, Integral) en el módulo numbers vía PEP 3141.

Vol 6: Python Track
6x09

LENGUAJES Y RUNTIMES
CPython Runtime

Estandarización del protocolo de llamadas ultrarrápido Vectorcall en CPython a través del PEP 590.

Vol 6: Python Track
6x0A

CULTURA HACKER Y WEB
Comunidad Python

Celebración del histórico primer taller de usuarios (First Python Workshop) en las instalaciones del instituto NIST en Maryland.

Vol 6: Python Track
6x0B

LENGUAJES Y RUNTIMES
CPython Runtime

Sustitución del histórico parser LL(1) por un nuevo motor PEG (Parsing Expression Grammar) en el PEP 617 para Python 3.9.

Vol 6: Python Track
6x0C

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Introducción de la función reversed() para iteración inversa sin mutar la secuencia original bajo el PEP 322.

Vol 6: Python Track
6x0D

LENGUAJES Y RUNTIMES
CPython Runtime

Aparición de Shed Skin, un compilador experimental que traducía programas en Python tipado estricto a código C++ nativo de alto rendimiento.

Vol 6: Python Track
6x0E

CULTURA HACKER Y WEB
Comunidad Python

Votación histórica de la PSF para prorrogar el fin de ciclo de vida (EOL) de Python 2.7 por cinco años adicionales.

Vol 6: Python Track
6x0F

LENGUAJES Y RUNTIMES
CPython Runtime

Anuncio oficial de Mojo, lenguaje que combina la sintaxis de Python con rendimiento directo en hardware y aceleradores de ML.

Vol 6: Python Track
6x10

LENGUAJES Y RUNTIMES
Ecosistema Rust

Creación de virtualenv, permitiendo aislar entornos de librerías independientes por proyecto sin contaminar el sistema operativo.

Vol 6: Python Track
6x11

LENGUAJES Y RUNTIMES
Frameworks Web

Debut de Django REST Framework (DRF), brindando serializadores avanzados y vistas navegables para construir APIs RESTful profesionales.

Vol 6: Python Track
6x12

LENGUAJES Y RUNTIMES
CPython Runtime

Introducción del recolector de basura generacional para ciclos de referencias huérfanas en Python 2.0.

Vol 6: Python Track
6x13

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Konstantin Sitnikov (PyO3 Team)

2020

Facilitó que cualquier programador de Rust publicara paquetes nativos en PyPI con la misma simplicidad que un paquete de Python puro.

Vol 6: Python Track
6x04

Ola Sitarska y Ola Sendecka

2014

Con un tutorial claro y sin jerga, impulsó a más de 20,000 mujeres a desplegar su primera web en un fin de semana.

Vol 6: Python Track
6x03

Łukasz Langa

2019

Eliminó la molesta duplicación entre tipos nativos y alias en mayúsculas como typing.List o typing.Dict, simplificando las anotaciones.

Vol 6: Python Track
6x02

Jeremy Hylton

2001

Requirió inicialmente from __future__ import nested_scopes, desatando debates entre partidarios del esquema plano anterior.

Vol 6: Python Track
6x07

PyCon Committee

2005

Sentar a novatos y veteranos codo a codo en sprints impulsó el cierre de bugs y forjó futuros core developers.

Vol 6: Python Track
6x06

Faster CPython Team

2025

Traduce llamadas monomórficas en saltos directos de memoria sin consultar el diccionario de clase en tiempo real.

Vol 6: Python Track
6x05

Mark Shannon y Jeroen Demeyer

2019

Evitó la costosa creación de tuplas y diccionarios intermedios temporales cada vez que se invocaba una función con argumentos posicionales o de clave.

Vol 6: Python Track
6x0A

Jeffrey Yasskin

2006

Empleó clases base abstractas (ABCs) para permitir que librerías externas como NumPy interactuaran armoniosamente con los tipos nativos.

Vol 6: Python Track
6x09

Sam Gross (Meta)

2023

Tras décadas de resistencia, el Steering Council aceptó hacer del modo sin GIL el futuro oficial de CPython.

Vol 6: Python Track
6x08

Raymond Hettinger

2004

Sustituyó el costoso truco de clonar e invertir listas en memoria mediante la notación de rebanado inverso seq[::-1].

Vol 6: Python Track
6x0D

Guido van Rossum, P. Galindo y Nikolaou

2020

Permitió sintaxis complejas con múltiples miradas adelante (lookahead), base del match/case de Python 3.10.

Vol 6: Python Track
6x0C

NIST (Guido van Rossum)

1994

Apenas asistieron 20 pioneros para conocer a Guido; casi la mitad trabajaba en agencias y laboratorios de EE.UU.

Vol 6: Python Track
6x0B

Modular (Chris Lattner)

2023

Creado por el autor de LLVM y Swift, promete acelerar código Python hasta 35,000x combinando tipado y GPUs.

Vol 6: Python Track
6x10

Python Software Foundation

2014

La directiva reconoció que retirar Python 2 en 2015 fracturaría la industria dada la lenta migración a Python 3.

Vol 6: Python Track
6x0F

Mark Dufour

2005

Demostró aceleraciones de hasta 200x en matemáticas puras, anticipando el interés de la industria por dialectos compilados.

Vol 6: Python Track
6x0E

Neil Schemenauer

2000

Añadió tres generaciones de recolección cíclica, evitando fugas de memoria sin degradar el conteo de referencias.

Vol 6: Python Track
6x13

Tom Christie

2011

Su interfaz web interactiva para explorar endpoints y enviar payloads al vuelo transformó el desarrollo de APIs backend.

Vol 6: Python Track
6x12

Ian Bicking

2007

Modificaba rutas copiando symlinks del binario de Python a carpetas locales, erradicando los temidos conflictos entre versiones de paquetes globales.

Vol 6: Python Track
6x11

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

LENGUAJES Y RUNTIMES
Ecosistema Rust

Lanzamiento de Ruff Format, ofreciendo un reemplazo directo y drop-in para Black programado íntegramente en Rust.

Vol 6: Python Track
6x14

LENGUAJES Y RUNTIMES
Frameworks Web

Nacimiento de Twisted, framework asíncrono por eventos antes de existir corrutinas nativas en Python.

Vol 6: Python Track
6x15

LENGUAJES Y RUNTIMES
CPython Runtime

Rediseño radical del mecanismo de paso de hilos en el núcleo con la llegada del New GIL a través del PEP 399 en Python 3.2.

Vol 6: Python Track
6x16

LENGUAJES Y RUNTIMES
Ecosistema Rust

Consagración de flake8, uniendo PyFlakes, el verificador de estilo pep8 y el medidor de complejidad ciclomática McCabe en una sola CLI.

Vol 6: Python Track
6x17

LENGUAJES Y RUNTIMES
Frameworks Web

Lanzamiento de uvloop, una reimplementación del bucle de eventos de asyncio programada en Cython sobre la biblioteca en C libuv.

Vol 6: Python Track
6x18

LENGUAJES Y RUNTIMES
Ecosistema Rust

Lanzamiento del nuevo algoritmo de resolución de dependencias con retroceso (backtracking resolver) en pip 20.3.

Vol 6: Python Track
6x19

LENGUAJES Y RUNTIMES
Ecosistema Rust

Lanzamiento de Pipenv, combinando Pipfile.lock y entornos virtuales automáticos bajo el lema "Dev Workflow for Humans".

Vol 6: Python Track
6x1A

LENGUAJES Y RUNTIMES
CPython Runtime

Lanzamiento público de GraalPy, la implementación de Python para la máquina virtual políglota de alto rendimiento GraalVM.

Vol 6: Python Track
6x1B

CULTURA HACKER Y WEB
Comunidad Python

Cancelación del formato presencial de PyCon US en Pittsburgh debido al estallido de la pandemia de COVID-19, pivotando a modalidad remota.

Vol 6: Python Track
6x1C

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Creación del módulo orientado a objetos pathlib para gestión semántica de rutas de archivos en el PEP 428.

Vol 6: Python Track
6x1D

LENGUAJES Y RUNTIMES
Frameworks Web

Nacimiento de Celery, sistema distribuido de colas asíncronas para segundo plano con RabbitMQ y Redis.

Vol 6: Python Track
6x1E

LENGUAJES Y RUNTIMES
Ecosistema Rust

Eliminación definitiva del módulo histórico distutils de la biblioteca estándar de CPython en Python 3.12.

Vol 6: Python Track
6x1F

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Incorporación de las @dataclass para autogeneración de métodos especiales (__init__, __repr__) bajo el PEP 557.

Vol 6: Python Track
6x20

CULTURA HACKER Y WEB
Comunidad Python

Lanzamiento del programa de subvenciones comunitarias (PSF Grants Program) para financiar eventos y conferencias locales en todo el mundo.

Vol 6: Python Track
6x21

LENGUAJES Y RUNTIMES
CPython Runtime

Incorporación del intérprete adaptativo con especialización dinámica de instrucciones en el PEP 659 para Python 3.11.

Vol 6: Python Track
6x22

CULTURA HACKER Y WEB
Comunidad Python

Celebración del funeral técnico oficial y apagado definitivo (Sunset) de Python 2.7 el 1 de enero.

Vol 6: Python Track
6x23

LENGUAJES Y RUNTIMES
CPython Runtime

Intento pionero de remover el GIL mediante el parche experimental Free Threading en Python 1.4.

Vol 6: Python Track
6x24

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Lanzamiento de Python 2.7, concebida como la versión de transición final y de soporte extendido de la rama 2.x.

Vol 6: Python Track
6x25

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Antoine Pitrou

2010

Reemplazó los 100 bytecodes por un temporizador (switch_interval), erradicando batallas destructivas entre hilos.

Vol 6: Python Track

6x16

Glyph Lefkowitz

2001

Creado para un videojuego online, sus objetos Deferred inspiraron las promesas asíncronas modernas en JavaScript.

Vol 6: Python Track

6x15

Astral

2023

Replica el formato de Black con 99% de fidelidad, formateando repositorios enteros en microsegundos.

Vol 6: Python Track

6x14

Pradyun Gedam y Paul Moore

2020

Eliminó el bug de pip que instalaba dependencias incompatibles si un paquete posterior pisaba al anterior.

Vol 6: Python Track

6x19

Yury Selivanov (MagicStack)

2016

Demostró que un bucle en Python podía ser de 2 a 4 veces más rápido que Node.js, superando 100k requests/seg.

Vol 6: Python Track

6x18

Tarek Ziadé

2010

Fue el juez inflexible de estilo que cuidó millones de repositorios antes de los formateadores automáticos.

Vol 6: Python Track

6x17

PSF Staff y Comité Organizador

2020

Pese al impacto financiero de cancelar el evento físico, la versión online gratuita unió a la comunidad global.

Vol 6: Python Track

6x1C

Oracle Labs

2021

Logró interoperabilidad nativa sin costo (zero-overhead) entre Python, Java, JavaScript, Ruby y lenguajes compilados por LLVM.

Vol 6: Python Track

6x1B

Kenneth Reitz

2017

Respaldado por la PSF para emular la experiencia de npm, causó furor aunque luego sufrió lentitud en su resolver.

Vol 6: Python Track

6x1A

Victor Stinner y CPython Core Team

2024

Tras una década de avisos, forzó a la comunidad a migrar de setup.py hacia estándares en pyproject.toml.

Vol 6: Python Track

6x1F

Ask Solem

2009

Es el estándar para descargar tareas pesadas de emails, reportes y modelos fuera del ciclo de respuesta HTTP.

Vol 6: Python Track

6x1E

Antoine Pitrou

2014

Permitió manipular rutas mediante sobrecarga del operador de barra / (ruta / "archivo.txt"), jubilando las llamadas anidadas a os.path.

Vol 6: Python Track

6x1D

Mark Shannon

2022

Reemplaza bytecodes en caliente por variantes monomórficas según tipos observados, acelerando CPython hasta un 60%.

Vol 6: Python Track

6x22

Python Software Foundation

2007

Ha repartido millones de dólares en becas para talleres, sprints de código y congresos en África, América Latina y Asia.

Vol 6: Python Track

6x21

Eric V. Smith

2017

Inspirado en la popular librería comunitaria attrs, redujo toneladas de código repetitivo (boilerplate) en clases contenedoras de datos.

Vol 6: Python Track

6x20

Benjamin Peterson

2010

Su fin de vida previsto para 2015 tuvo que postergarse al 1 de enero de 2020 por el inmenso código legacy sin migrar.

Vol 6: Python Track

6x25

Greg Stein

1999

Logró hilos paralelos, pero bloquear cada operación duplicó el tiempo de ejecución monohilo, forzando su descarte.

Vol 6: Python Track

6x24

Python Software Foundation

2020

La PSF celebró un funeral simbólico con pastel y confeti, conminando al mundo a apagar Python 2 para siempre.

Vol 6: Python Track

6x23

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

LENGUAJES Y RUNTIMES
Frameworks Web

Creación de Django Channels, extendiendo el corazón síncrono de Django para admitir WebSockets, chats y tareas en tiempo real.

Vol 6: Python Track
6x26

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Estandarización de cabeceras de codificación de archivos fuente con la directiva # -- coding: utf-8 -- vía PEP 263.

Vol 6: Python Track
6x27

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Introducción de la sintaxis formal de anotaciones de tipos para variables de clase y módulo vía PEP 526.

Vol 6: Python Track
6x28

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Transformación de generadores en corrutinas completas mediante el método .send() y excepciones con el PEP 342.

Vol 6: Python Track
6x29

LENGUAJES Y RUNTIMES
Ecosistema Rust

Expansión de uv como gestor integral de proyectos, scripts monosemilla y versiones de Python.

Vol 6: Python Track
6x2A

LENGUAJES Y RUNTIMES
CPython Runtime

Creación del objeto inmortal (PEP 683), permitiendo compartir tipos entre núcleos sin mutar conteos de referencias.

Vol 6: Python Track
6x2B

LENGUAJES Y RUNTIMES
Frameworks Web

Creación de aiohttp, proveyendo cliente y servidor HTTP asíncrono construido sobre el nuevo bucle de eventos asyncio de Python 3.4.

Vol 6: Python Track
6x2C

LENGUAJES Y RUNTIMES
Frameworks Web

Lanzamiento del toolkit y microframework Starlette, concebido como el pilar fundamental para la nueva era de aplicaciones ASGI.

Vol 6: Python Track
6x2D

LENGUAJES Y RUNTIMES
Ecosistema Rust

Nacimiento de Poetry, gestionando dependencias deterministas, resolución de grafos y empaquetado en pyproject.toml.

Vol 6: Python Track
6x2E

LENGUAJES Y RUNTIMES
CPython Runtime

Debut de Pyrex, compilador precursor de extensiones en C mezclando sintaxis de Python con tipos estáticos.

Vol 6: Python Track
6x2F

LENGUAJES Y RUNTIMES
Ecosistema Rust

Especificación de la interfaz estándar entre herramientas de empaquetado e instaladores mediante el PEP 517.

Vol 6: Python Track
6x30

LENGUAJES Y RUNTIMES
CPython Runtime

Creación de Stackless Python, desacoplando la pila de C para habilitar micro-hilos livianos (tasklets).

Vol 6: Python Track
6x31

LENGUAJES Y RUNTIMES
Frameworks Web

Liberación de Zope 2, servidor de aplicaciones pionero que introdujo la base de datos orientada a objetos ZODB.

Vol 6: Python Track
6x32

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Creación del tipo mutable de secuencias de bytes bytearray con la especificación del PEP 3137.

Vol 6: Python Track
6x33

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Introducción del método avanzado de formateo de cadenas str.format() a través de la especificación del PEP 3101.

Vol 6: Python Track
6x34

LENGUAJES Y RUNTIMES
CPython Runtime

Presentación pública de IronPython, llevando la sintaxis de Python al entorno de ejecución de Microsoft .NET y Mono.

Vol 6: Python Track
6x35

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Simplificación de la unión de tipos estáticos utilizando el operador de tubería int | str mediante el PEP 604.

Vol 6: Python Track
6x36

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Adición de guiones bajos como separadores visuales de dígitos en literales numéricos (1_000_000) en el PEP 515.

Vol 6: Python Track
6x37

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Ryan Gonzalez y Guido van Rossum

2016

Habilitó expresiones directas como primer_nombre: str = "Ada", allanando el camino para modelos declarativos como Pydantic.

Vol 6: Python Track · 6x28

Marc-André Lemburg y Martin von Löwis

2003

Evitó que el intérprete fallara con excepciones de sintaxis al encontrar acentos, tildes o caracteres asiáticos en comentarios y strings.

Vol 6: Python Track · 6x27

Andrew Godwin

2016

Introdujo el concepto de capas de canales (channel layers) sobre Redis, permitiendo a Django cruzar la frontera de las arquitecturas HTTP clásicas.

Vol 6: Python Track · 6x26

Eric Snow y Eddie Elizondo (Meta)

2023

Permitió a Instagram/Meta compartir gigabytes de memoria entre procesos sin activar Copy-on-Write accidentalmente.

Vol 6: Python Track · 6x2B

Astral

2024

Permitió ejecutar scripts con dependencias embebidas directamente mediante uv run y descargar versiones de CPython a la carta en milisegundos.

Vol 6: Python Track · 6x2A

Guido van Rossum y Phillip Eby

2005

Permitió inyectar valores directamente en generadores en ejecución, sentando los cimientos teóricos para los futuros frameworks asíncronos.

Vol 6: Python Track · 6x29

Sébastien Eustace

2018

Su estricto resolver y su elegante CLI en Python lo convirtieron en el estándar predilecto de la comunidad.

Vol 6: Python Track · 6x2E

Tom Christie

2018

Aportó manejo robusto de WebSockets, middleware asíncrono y streaming de datos con cero dependencias externas pesadas.

Vol 6: Python Track · 6x2D

Nikolay Kim y Andrew Svetlov

2013

Fue la primera gran librería en demostrar el potencial de las corrutinas para consumir y exponer APIs asíncronas con miles de conexiones abiertas.

Vol 6: Python Track · 6x2C

Christian Tismer

1998

Fue el motor de EVE Online, permitiendo a sus servidores coordinar miles de naves espaciales en un único proceso.

Vol 6: Python Track · 6x31

Thomas Kluyver, Brett Cannon et al.

2019

Desacopló definitivamente a instaladores como pip de backends de compilación como flit, hatch o poetry-core.

Vol 6: Python Track · 6x30

Greg Ewing

2002

Permitió escribir extensiones rápidas sin manipular manualmente las complejas macros de la Python C API.

Vol 6: Python Track · 6x2F

Talin

2006

Diseñado para modernizar y eventualmente jubilar al venerable operador de interpolación heredado de C % (printf-style).

Vol 6: Python Track · 6x34

Guido van Rossum

2008

Brindó manipulación de buffers binarios en memoria sin incurrir en las continuas copias e inmutabilidad del tipo bytes.

Vol 6: Python Track · 6x33

Jim Fulton (Digital Creations)

2000

Guardaba objetos Python en disco sin usar SQL, impulsando las primeras aplicaciones web corporativas del lenguaje.

Vol 6: Python Track · 6x32

Georg Brandl

2016

Mejoró enormemente la legibilidad de máscaras binarias, direcciones hexadecimales y grandes números enteros sin alterar su valor en runtime.

Vol 6: Python Track · 6x37

Philippe Prados y Maggie Moss

2020

Reemplazó la pesada y verbose sintaxis Union[int, str] de typing por una notación mucho más limpia y cercana a otros lenguajes modernos.

Vol 6: Python Track · 6x36

Jim Hugunin (Microsoft)

2004

La demo en el CLR impresionó tanto a Microsoft que contrataron de inmediato a Hugunin para liderar el proyecto.

Vol 6: Python Track · 6x35

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

LENGUAJES Y RUNTIMES
Frameworks Web

Lanzamiento del motor de plantillas Jinja2, optimizado para compilar código HTML directamente a código Python en memoria.

Vol 6: Python Track
6x38

LENGUAJES Y RUNTIMES
Sintaxis y PEPs

Publicación de la propuesta de cadenas con plantilla etiquetada (template strings) t"..." para seguridad en el PEP 750.

Vol 6: Python Track
6x39

LENGUAJES Y RUNTIMES
CPython Runtime

Reescritura del compilador de CPython introduciendo la estructura de Árbol de Sintaxis Abstracta (AST) formal mediante el PEP 339.

Vol 6: Python Track
6x3A

LENGUAJES Y RUNTIMES
Ecosistema Rust

Nacimiento de Conda, el gestor de paquetes y entornos binarios multiplataforma creado para el ecosistema analítico y científico.

Vol 6: Python Track
6x3B

LENGUAJES Y RUNTIMES
CPython Runtime

Implementación original del cerrojo global del intérprete (Global Interpreter Lock o GIL) en el bucle central de CPython.

Vol 6: Python Track
6x3C

CULTURA HACKER Y WEB
Comunidad Python

Redacción del PEP 1, institucionalizando las propuestas de mejora como pilar de la evolución de Python.

Vol 6: Python Track
6x3D

LENGUAJES Y RUNTIMES
Ecosistema Rust

Creación de tox, la herramienta de orquestación de pruebas automatizadas sobre múltiples versiones de Python en entornos aislados.

Vol 6: Python Track
6x3E

LENGUAJES Y RUNTIMES
CPython Runtime

Evaluación del modo libre de hilos para declarar el soporte no-GIL por defecto a partir de Python 3.15.

Vol 6: Python Track
6x3F

CULTURA HACKER Y WEB
Cine Hacker

Estreno de WarGames, popularizando el wardialing, los módems acústicos de 300 baudios y la supercomputadora militar WOPR.

Vol 7: Sci-Fi & Pop Culture
7x00

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Blade Runner 2049, explorando el amor y la conciencia sintética mediante la IA holográfica de compañía Joi.

Vol 7: Sci-Fi & Pop Culture
7x01

CULTURA HACKER Y WEB
Cine Hacker

Estreno de Pirates of Silicon Valley, dramatizando la rivalidad fundacional entre Steve Jobs (Apple) y Bill Gates (Microsoft).

Vol 7: Sci-Fi & Pop Culture
7x02

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Westworld, empleando por primera vez gráficos digitales pixelados 2D generados por ordenador para la visión del pistolero.

Vol 7: Sci-Fi & Pop Culture
7x03

CÓMPUTO E IA
Literatura Especulativa

Publicación de la Guía del autoestopista galáctico, introduciendo la supercomputadora Pensamiento Profundo y el sentido de la vida (42).

Vol 7: Sci-Fi & Pop Culture
7x04

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Las dos torres, consagrando la captura de movimiento corporal y animación facial fotorrealista con Gollum.

Vol 7: Sci-Fi & Pop Culture
7x05

CULTURA HACKER Y WEB
Cine Hacker

Estreno de The Social Network, retratando con precisión técnica el desarrollo de software LAMP y scripts de web scraping en Python y Perl.

Vol 7: Sci-Fi & Pop Culture
7x06

CÓMPUTO E IA
Literatura Especulativa

Publicación de De la Tierra a la Luna, anticipando con rigor la balística espacial, el lanzamiento en Florida y la gravedad cero.

Vol 7: Sci-Fi & Pop Culture
7x07

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Her, explorando la intimidad emocional con un sistema operativo conversacional consciente basado en procesamiento de lenguaje.

Vol 7: Sci-Fi & Pop Culture
7x08

CÓMPUTO E IA
Literatura Especulativa

Publicación de Un mundo feliz, explorando el condicionamiento psicológico, la clonación humana y la felicidad tecnocrática forzada.

Vol 7: Sci-Fi & Pop Culture
7x09

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Brett Cannon y Neil Schemenauer

2005

Reemplazó el arcaico compilador de un solo paso por un diseño modular de dos fases que facilitó el análisis estático y la optimización de bytecode.

Vol 6: Python Track · 6x3A

Jim Baker y Guido van Rossum

2025

Previene inyecciones SQL y XSS al diferir la evaluación de strings formateados hasta que un parser seguro los valide.

Vol 6: Python Track · 6x39

Armin Ronacher

2008

Tomó la sintaxis {{ variable }} de Django, perfeccionando su velocidad y seguridad con auto-escape automático nativo.

Vol 6: Python Track · 6x38

Barry Warsaw y Jeremy Hylton

2000

Inspirado en los RFCs de Internet, fijó que ningún cambio mayor ocurriría sin una propuesta pública documentada.

Vol 6: Python Track · 6x3D

Guido van Rossum

1992

Evitaba carreras de memoria en C, sin prever el dolor de cabeza que causaría para el cómputo multinúcleo décadas después.

Vol 6: Python Track · 6x3C

Travis Oliphant y Peter Wang

2012

Resolvió instalar librerías científicas con dependencias en Fortran y C sin compilar en Windows, macOS o Linux.

Vol 6: Python Track · 6x3B

John Badham

1983

Inspiró al presidente Ronald Reagan a emitir la primera directiva de seguridad informática gubernamental en EE.UU.

Vol 7: Sci-Fi & Pop Culture · 7x00

Python Steering Council

2026

Evaluó la estabilidad del ecosistema C para apagar el GIL por defecto y unificar la distribución del lenguaje.

Vol 6: Python Track · 6x3F

Holger Krekel

2010

Se convirtió en el guardián indiscutible de la integración continua (CI) para asegurar que una librería funcionara idénticamente en Python 2 y 3.

Vol 6: Python Track · 6x3E

Michael Crichton

1973

John Whitney Jr. tardó meses en procesar apenas dos minutos de metraje pixelado en computadoras mainframe de la época.

Vol 7: Sci-Fi & Pop Culture · 7x03

Martyn Burke

1999

Steve Jobs invitó al actor Noah Wyle a la keynote de Macworld tras quedar impresionado por su personificación.

Vol 7: Sci-Fi & Pop Culture · 7x02

Denis Villeneuve

2017

Roger Deakins ganó su primer premio Óscar a mejor fotografía tras catorce nominaciones por su iluminación geométrica.

Vol 7: Sci-Fi & Pop Culture · 7x01

David Fincher

2010

Aaron Sorkin y Fincher exigieron que los fragmentos de código en pantalla y las llamadas a wget fueran técnicamente auténticos.

Vol 7: Sci-Fi & Pop Culture · 7x06

Peter Jackson (Weta Digital)

2002

Andy Serkis grabó todas las escenas en el set vistiendo un traje ajustado con marcadores ópticos de seguimiento.

Vol 7: Sci-Fi & Pop Culture · 7x05

Douglas Adams

1979

Popularizó el pez de Babel, concepto adoptado por Yahoo! para bautizar a su histórico servicio de traducción web.

Vol 7: Sci-Fi & Pop Culture · 7x04

Aldous Huxley

1932

Inspirada en el fordismo industrial y las fábricas de ensamblaje en serie de Henry Ford.

Vol 7: Sci-Fi & Pop Culture · 7x09

Spike Jonze

2013

Scarlett Johansson grabó la voz del sistema operativo Samantha aislada en una cabina insonorizada tras el rodaje.

Vol 7: Sci-Fi & Pop Culture · 7x08

Julio Verne

1865

Calculó una velocidad de escape de 11 km/s y situó el cañón en Cabo Cañaveral, a pocos kilómetros de la NASA.

Vol 7: Sci-Fi & Pop Culture · 7x07

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Tron, pionera en integrar animación computacional 3D como entorno narrativo virtual y trajes con circuitos brillantes.

7x0A

CÓMPUTO E IA
Literatura Especulativa

Publicación de Dune, estableciendo la Yihad Butleriana y el mandato 'No construirás una máquina a semejanza de la mente humana'.

7x0B

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de RoboCop, explorando la privatización corporativa de la ley y la reconstrucción cibernética con interfaz de puntería HUD.

7x0C

CULTURA HACKER Y WEB
Cine Hacker

Estreno de Hackers, reflejando la subcultura informática de los 90 con visualizaciones 3D de servidores Gibson y fono-hackeo.

7x0D

CULTURA HACKER Y WEB
Cine Hacker

Estreno de la miniserie The Playlist, retratando la creación de Spotify, el streaming P2P legal y la disrupción discográfica.

7x0E

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Ex Machina, aplicando un test de Turing interactivo con el robot humanoide Ava dentro de una mansión domótica aislada.

7x0F

CÓMPUTO E IA
Literatura Especulativa

Publicación de la recopilación Yo, Robot, formalizando las célebres Tres Leyes de la Robótica en la literatura.

7x10

CULTURA HACKER Y WEB
Cine Hacker

Emisión de la serie Mr. Robot, estableciendo el estándar de oro de autenticidad hacker con Kali Linux y comandos de terminal verídicos.

7x11

CÓMPUTO E IA
Literatura Especulativa

Publicación de El jardín de senderos que se bifurcan, precursor conceptual del hipertexto, universos paralelos y redes ramificadas.

7x12

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Le Voyage dans la Lune, inaugurando los efectos visuales, animación fotograma a fotograma y maquetas cinematográficas.

7x13

CULTURA HACKER Y WEB
Cine Hacker

Estreno de Santo contra las mujeres vampiro, mostrando al héroe usando su reloj-transmisor y pantallas de circuito cerrado.

7x14

CÓMPUTO E IA
Literatura Especulativa

Publicación de Fundación, presentando la Psicohistoria como el análisis estadístico y algorítmico masivo de civilizaciones.

7x15

CÓMPUTO E IA
Literatura Especulativa

Publicación de Veinte mil leguas de viaje submarino, concibiendo el submarino eléctrico Nautilus con baterías de sodio.

7x16

CÓMPUTO E IA
Literatura Especulativa

Publicación de Frankenstein, inaugurando el dilema ético de la creación de vida artificial y la soberbia científica.

7x17

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Star Wars, proyectando la simulación vectorial tridimensional de la Estrella de la Muerte en la reunión táctica rebelde.

7x18

CÓMPUTO E IA
Literatura Especulativa

Publicación de las Notas sobre la Máquina Analítica, concibiendo el primer algoritmo ejecutable por un computador universal.

7x19

CÓMPUTO E IA
Literatura Especulativa

Publicación de El cuento de la criada, clásica distopía sobre el control totalitario del cuerpo, la reproducción y la teocracia.

7x1A

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de The Matrix, revolucionando la cinematografía con el efecto Bullet Time y la icónica lluvia digital de código verde.

7x1B

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Vol 7: Sci-Fi & Pop Culture

Paul Verhoeven

1987

El actor Peter Weller tardaba más de tres horas diarias en colocarse el pesado traje de fibra de vidrio y titanio.

7x0C

Vol 7: Sci-Fi & Pop Culture

Frank Herbert

1965

La prohibición de ordenadores dio origen a los Mentat, humanos entrenados como calculadoras lógicas vivientes.

7x0B

Vol 7: Sci-Fi & Pop Culture

Steven Lisberger

1982

La Academia de Hollywood no la nominó a mejores efectos visuales porque consideró que 'usar computadoras era hacer trampa'.

7x0A

Vol 7: Sci-Fi & Pop Culture

Alex Garland

2014

La actriz Alicia Vikander, ex bailarina clásica, utilizó movimientos de precisión micrométrica para acentuar la robótica de Ava.

7x0F

Vol 7: Sci-Fi & Pop Culture

Per-Olav Sørensen (Netflix)

2022

Daniel Ek y Ludvig Strigeus contrataron al creador de µTorrent para diseñar la arquitectura de streaming con latencia casi cero.

7x0E

Vol 7: Sci-Fi & Pop Culture

Iain Softley

1995

La página web oficial de la película fue hackeada semanas después de su estreno por internautas reales.

7x0D

Vol 7: Sci-Fi & Pop Culture

Jorge Luis Borges

1941

Ted Nelson y pioneros del hipertexto citaron el laberinto temporal de Borges como metáfora de la navegación en la World Wide Web.

7x12

Vol 7: Sci-Fi & Pop Culture

Sam Esmail

2015

Cada pantalla mostraba código real de exploits, scripts de Python e ingeniería social testeada por expertos de seguridad.

7x11

Vol 7: Sci-Fi & Pop Culture

Isaac Asimov

1950

Las leyes fueron refinadas en conversación con John W. Campbell, legendario editor de la revista Astounding Science Fiction.

7x10

Vol 7: Sci-Fi & Pop Culture

Isaac Asimov

1951

Asimov concibió la trama inspirándose en la lectura de 'Historia de la decadencia y caída del Imperio romano' de Gibbon.

7x15

Vol 7: Sci-Fi & Pop Culture

Alfonso Corona Blake

1962

El Enmascarado de Plata popularizó en el cine latinoamericano la tecnología de comunicación portátil décadas antes de los smartwatches.

7x14

Vol 7: Sci-Fi & Pop Culture

Georges Méliès

1902

Méliès descubrió accidentalmente el truco de sustitución (stop-trick) cuando su cámara se atascó rodando en París.

7x13

Vol 7: Sci-Fi & Pop Culture

George Lucas

1977

La animación vectorial por alambre (wireframe) fue programada en un minicomputador PDP-11 por el artista Larry Cuba.

7x18

Vol 7: Sci-Fi & Pop Culture

Mary Shelley

1818

Escrita durante el 'año sin verano' en Villa Diodati tras un desafío literario propuesto por Lord Byron.

7x17

Vol 7: Sci-Fi & Pop Culture

Julio Verne

1870

Verne predijo la propulsión marina eléctrica y el buceo autónomo décadas antes de su adopción militar formal.

7x16

Vol 7: Sci-Fi & Pop Culture

Hermanas Wachowski

1999

El famoso código verde de la pantalla estaba compuesto por recetas de sushi en caracteres japoneses katakana invertidos.

7x1B

Vol 7: Sci-Fi & Pop Culture

Margaret Atwood

1985

Atwood se impuso la regla estricta de no incluir ninguna atrocidad que no hubiera ocurrido ya en la historia real.

7x1A

Vol 7: Sci-Fi & Pop Culture

Ada Lovelace

1843

Profetizó que la máquina podría componer música elaborada y manipular cualquier símbolo abstracto.

7x19

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CULTURA HACKER Y WEB
Clásicos Ciberpunk

Publicación de la novela Neuromante, obra cumbre del ciberpunk acuñando el ciberespacio, matriz, implantes y ciberdecks.

Vol 7: Sci-Fi & Pop Culture
7x1C

CÓMPUTO E IA
Literatura Especulativa

Publicación de la novela Solaris, abordando los límites del entendimiento humano ante un océano vivo e inteligencia no antropomórfica.

Vol 7: Sci-Fi & Pop Culture
7x1D

CÓMPUTO E IA
Literatura Especulativa

Publicación de El jinete de la onda de choque, anticipando redes digitales masivas y acuñando el término gusano (worm) informático.

Vol 7: Sci-Fi & Pop Culture
7x1E

CÓMPUTO E IA
Literatura Especulativa

Publicación de la obra cumbre de George Orwell, introduciendo el Gran Hermano, la Neolengua y las telepantallas de vigilancia.

Vol 7: Sci-Fi & Pop Culture
7x1F

CULTURA HACKER Y WEB
Clásicos Ciberpunk

Publicación de Snow Crash, acuñando los términos Metaverso, avatares interactivos tridimensionales y criptoanarquía urbana.

Vol 7: Sci-Fi & Pop Culture
7x20

CÓMPUTO E IA
Literatura Especulativa

Publicación de La guerra de los mundos, inmortalizando la invasión marciana con trípodes mecánicos y rayos de energía calórica.

Vol 7: Sci-Fi & Pop Culture
7x21

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Minority Report, concibiendo la interfaz espacial gestual de pantallas transparentes y predicción analítica del delito.

Vol 7: Sci-Fi & Pop Culture
7x22

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno del filme anime Akira, deslumbrando con la distopía ciberpunk de Neo-Tokio, animación tradicional récord y luz estroboscópica.

Vol 7: Sci-Fi & Pop Culture
7x23

CÓMPUTO E IA
Literatura Especulativa

Publicación de Fahrenheit 451, anticipando auriculares interactivos minúsculos (conchas marinas) y pantallas de pared parlantes.

Vol 7: Sci-Fi & Pop Culture
7x24

CULTURA HACKER Y WEB
Clásicos Ciberpunk

Publicación de Criptonomicón, epopeya tecnológica uniendo el descifrado de Enigma en la guerra con la creación de criptomonedas privadas.

Vol 7: Sci-Fi & Pop Culture
7x25

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Terminator 2: Judgment Day, revolucionando el cine con el androide de metal líquido T-1000 modelado en Silicon Graphics.

Vol 7: Sci-Fi & Pop Culture
7x26

CÓMPUTO E IA
Literatura Especulativa

Publicación de ¿Sueñan los androides con ovejas eléctricas?, cuestionando la humanidad biológica frente a réplicas sintéticas.

Vol 7: Sci-Fi & Pop Culture
7x27

CÓMPUTO E IA
Literatura Especulativa

Publicación de La invención de Morel, pionera de la realidad virtual y proyecciones holográficas reproduciendo la realidad en bucle.

Vol 7: Sci-Fi & Pop Culture
7x28

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Forbidden Planet, presentando a Robby the Robot y la primera banda sonora cinematográfica 100% electrónica.

Vol 7: Sci-Fi & Pop Culture
7x29

CÓMPUTO E IA
Literatura Especulativa

Publicación del influyente ensayo As We May Think, concibiendo el Memex como precursor conceptual del hipertexto y la Web.

Vol 7: Sci-Fi & Pop Culture
7x2A

CÓMPUTO E IA
Literatura Especulativa

Publicación de El problema de los tres cuerpos, concibiendo una computadora masiva hecha con millones de soldados formando compuertas.

Vol 7: Sci-Fi & Pop Culture
7x2B

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Back to the Future, inmortalizando los viajes en el tiempo con el auto DeLorean, el condensador de fluzo y 1.21 gigawatts.

Vol 7: Sci-Fi & Pop Culture
7x2C

CÓMPUTO E IA
Literatura Especulativa

Publicación de La máquina del tiempo, popularizando la traslación temporal mediante un dispositivo mecánico de relojería.

Vol 7: Sci-Fi & Pop Culture
7x2D

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

John Brunner

1975

Los investigadores de Xerox PARC bautizaron oficialmente sus programas de red como 'worms' en honor a este libro.

Vol 7: Sci-Fi & Pop Culture

7x1E

Stanisław Lem

1961

Lem criticó duramente las adaptaciones cinematográficas afirmando que convirtieron su obra en un drama psicológico terrenal.

Vol 7: Sci-Fi & Pop Culture

7x1D

William Gibson

1984

Gibson casi abandonó la novela tras ver Blade Runner en cines, temiendo que el público creyera que le había copiado la estética.

Vol 7: Sci-Fi & Pop Culture

7x1C

H.G. Wells

1898

Orson Welles causó pánico radiofónico masivo cuatro décadas después al simular la invasión en directo en CBS.

Vol 7: Sci-Fi & Pop Culture

7x21

Neal Stephenson

1992

Inspiró directamente a los fundadores de Google Earth y a los pioneros de la realidad virtual y mundos 3D.

Vol 7: Sci-Fi & Pop Culture

7x20

George Orwell

1949

Orwell mecanografió el manuscrito final recluido en una remota y fría granja de la isla escocesa de Jura.

Vol 7: Sci-Fi & Pop Culture

7x1F

Ray Bradbury

1953

Bradbury redactó la novela en el sótano de la biblioteca de UCLA alquilando una máquina de escribir por diez centavos la media hora.

Vol 7: Sci-Fi & Pop Culture

7x24

Katsuhiro Otomo

1988

Se utilizaron más de 160,000 celdas de animación y 327 colores creados a medida, popularizando el anime masivo en Occidente.

Vol 7: Sci-Fi & Pop Culture

7x23

Steven Spielberg

2002

Spielberg reunió a científicos y futuristas del MIT Media Lab para proyectar cómo sería la interacción humana en el futuro.

Vol 7: Sci-Fi & Pop Culture

7x22

Philip K. Dick

1968

Sentó las bases para el test Voight-Kampff y la icónica adaptación cinematográfica Blade Runner de Ridley Scott.

Vol 7: Sci-Fi & Pop Culture

7x27

James Cameron

1991

ILM pasó de 6 empleados de gráficos por computadora a más de 40 para poder completar las tomas del T-1000.

Vol 7: Sci-Fi & Pop Culture

7x26

Neal Stephenson

1999

Incluye el algoritmo de cifrado 'Pontifex' diseñado especialmente para el libro por el criptógrafo Bruce Schneier.

Vol 7: Sci-Fi & Pop Culture

7x25

Vannevar Bush

1945

Inspiró directamente las investigaciones de Ted Nelson (Xanadu) y Douglas Engelbart (el ratón y GUI).

Vol 7: Sci-Fi & Pop Culture

7x2A

Fred M. Wilcox

1956

Louis y Bebe Barron diseñaron circuitos cibernéticos caseros que se autodestruían para generar los tonos electrónicos del filme.

Vol 7: Sci-Fi & Pop Culture

7x29

Adolfo Bioy Casares

1940

Con prólogo de Jorge Luis Borges, influyó directamente en la serie televisiva Lost y acuñó en filosofía el 'efecto Morel'.

Vol 7: Sci-Fi & Pop Culture

7x28

H.G. Wells

1895

Wells introdujo la noción del tiempo como una cuarta dimensión matemática física.

Vol 7: Sci-Fi & Pop Culture

7x2D

Robert Zemeckis

1985

En los primeros borradores del guion, la máquina del tiempo era una heladera doméstica impulsada por una explosión nuclear.

Vol 7: Sci-Fi & Pop Culture

7x2C

Liu Cixin

2008

Fue el primer autor asiático en ganar el prestigioso premio Hugo a la mejor novela tras ser traducido por Ken Liu.

Vol 7: Sci-Fi & Pop Culture

7x2B

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Metrópolis, inmortalizando la transformación del androide María (Maschinenmensch) en la iconografía de la ciencia ficción.

Vol 7: Sci-Fi & Pop Culture
7x2E

CULTURA HACKER Y WEB
Cine Hacker

Estreno de Snowden, retratando la filtración de programas de cibervigilancia masiva global de la NSA mediante tarjetas microSD.

Vol 7: Sci-Fi & Pop Culture
7x2F

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Soylent Green (Cuando el destino nos alcance), distopía sobre sobrepoblación, colapso climático y la industria alimentaria.

Vol 7: Sci-Fi & Pop Culture
7x30

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Star Trek II, deslumbrando con el efecto Génesis, primer uso cinematográfico de sistemas de partículas fractales 3D.

Vol 7: Sci-Fi & Pop Culture
7x31

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno del filme Una odisea del espacio, revolucionando el cine con interfaces vectoriales realistas y la desconexión de HAL 9000.

Vol 7: Sci-Fi & Pop Culture
7x32

CULTURA HACKER Y WEB
Cine Hacker

Estreno de The Net, alertando sobre el robo de identidad digital, vulnerabilidades backdoor corporativas y pedidos web por IP.

Vol 7: Sci-Fi & Pop Culture
7x33

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Aliens, popularizando el exoesqueleto robótico industrial de carga Caterpillar P-5000 Power Loader en combate.

Vol 7: Sci-Fi & Pop Culture
7x34

CULTURA HACKER Y WEB
Clásicos Ciberpunk

Publicación del manga Ghost in the Shell, explorando cerebros cibernéticos interconectados, hackeo cerebral y la Sección 9.

Vol 7: Sci-Fi & Pop Culture
7x35

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Blade Runner, inmortalizando el test Voight-Kampff, lentes corneales de replicantes y la consola de aumento fotográfico Esper.

Vol 7: Sci-Fi & Pop Culture
7x36

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Alien, presentando a la computadora de a bordo MU-TH-UR 6000 y al sintético encubierto Ash con directivas secretas.

Vol 7: Sci-Fi & Pop Culture
7x37

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Iron Man, fascinando al público con la interfaz espacial holográfica 3D interactiva y la inteligencia artificial JARVIS.

Vol 7: Sci-Fi & Pop Culture
7x38

CULTURA HACKER Y WEB
Clásicos Ciberpunk

Publicación de la novela corta True Names, describiendo con rigor matemático el ciberespacio, encriptación y anonimato digital.

Vol 7: Sci-Fi & Pop Culture
7x39

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Jurassic Park, sustituyendo el stop-motion por dinosaurios CGI fotorrealistas y mostrando la interfaz 3D fsn de IRIX UNIX.

Vol 7: Sci-Fi & Pop Culture
7x3A

CÓMPUTO E IA
Literatura Especulativa

Estreno de la obra teatral R.U.R., introduciendo por primera vez la palabra robot en el vocabulario mundial.

Vol 7: Sci-Fi & Pop Culture
7x3B

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Terminator, presentando a la red militar Skynet y la interfaz HUD del T-800 con código ensamblador 6502 en pantalla.

Vol 7: Sci-Fi & Pop Culture
7x3C

CÓMPUTO E IA
Literatura Especulativa

Publicación de Ready Player One, fenómeno sobre mundos virtuales inmersivos masivos (OASIS) y competencia corporativa por su control.

Vol 7: Sci-Fi & Pop Culture
7x3D

CÓMPUTO E IA
CGI y Efectos Visuales

Estreno de Toy Story, el primer largometraje en la historia del cine animado completamente por computadora.

Vol 7: Sci-Fi & Pop Culture
7x3E

CULTURA HACKER Y WEB
Cine Hacker

Emisión de la serie Silicon Valley, satirizando la cultura startup con el algoritmo de compresión de datos Pied Piper.

Vol 7: Sci-Fi & Pop Culture
7x3F

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org

Richard Fleischer

1973

Charlton Heston protagonizó el mítico desenlace sobre la verdad de las galletas verdes alimentarias de la corporación.

Vol 7: Sci-Fi & Pop Culture
7x30

Oliver Stone

2016

Oliver Stone se reunió clandestinamente con Edward Snowden en Moscú en múltiples ocasiones para afinar el guion.

Vol 7: Sci-Fi & Pop Culture
7x2F

Fritz Lang

1927

Eugen Schüfftan inventó el célebre 'Proceso Schüfftan' usando espejos para integrar a los actores en maquetas gigantes.

Vol 7: Sci-Fi & Pop Culture
7x2E

Irwin Winkler

1995

Mostró una de las primeras compras de comida a domicilio por Internet ordenando pizza en línea en pantalla.

Vol 7: Sci-Fi & Pop Culture
7x33

Stanley Kubrick

1968

Douglas Trumbull perfeccionó la técnica de fotografía 'slit-scan' para producir la alucinante secuencia del viaje estelar.

Vol 7: Sci-Fi & Pop Culture
7x32

Nicholas Meyer (Lucasfilm)

1982

Fue creado por la división informática de Lucasfilm (futura Pixar) bajo la dirección de Alvy Ray Smith y Loren Carpenter.

Vol 7: Sci-Fi & Pop Culture
7x31

Ridley Scott

1982

La llovizna ácida constante y la densa niebla urbana se añadieron para ocultar las uniones de las maquetas de los edificios.

Vol 7: Sci-Fi & Pop Culture
7x36

Masamune Shirow

1989

Shirow llenó las páginas de notas técnicas sobre ciberseguridad militar, prótesis mecánicas y filosofía transhumanista.

Vol 7: Sci-Fi & Pop Culture
7x35

James Cameron

1986

El montacargas robótico medía más de 3 metros y requería un titiritero oculto detrás de Sigourney Weaver para moverlo.

Vol 7: Sci-Fi & Pop Culture
7x34

Vernor Vinge

1981

Considerada por la comunidad hacker y criptográfica como el acta de nacimiento de la narrativa ciberpunk.

Vol 7: Sci-Fi & Pop Culture
7x39

Jon Favreau (Marvel Studios)

2008

Favreau y Robert Downey Jr. consultaron con Elon Musk en las instalaciones de SpaceX para moldear el perfil de Tony Stark.

Vol 7: Sci-Fi & Pop Culture
7x38

Ridley Scott

1979

La impactante escena de la revelación de Ash utilizó leche condensada, fideos y caviar vegetal como fluidos biomecánicos.

Vol 7: Sci-Fi & Pop Culture
7x37

James Cameron

1984

El código ensamblador visible en los ojos de Terminator fue tomado de programas publicados en revistas de Apple II.

Vol 7: Sci-Fi & Pop Culture
7x3C

Karel Čapek

1920

La palabra deriva del término checo 'robota', que designaba el trabajo forzado y servidumbre feudal.

Vol 7: Sci-Fi & Pop Culture
7x3B

Steven Spielberg

1993

La frase 'It\'s a UNIX system, I know this!' mostraba el explorador de archivos tridimensional real fsn de Silicon Graphics.

Vol 7: Sci-Fi & Pop Culture
7x3A

Mike Judge (HBO)

2014

Científicos de Stanford crearon una fórmula matemática real ('Weissman Score') para medir la compresión ficticia de la serie.

Vol 7: Sci-Fi & Pop Culture
7x3F

John Lasseter (Pixar)

1995

Se requirieron 117 computadoras Sun Microsystems trabajando 24 horas al día para renderizar sus 77 minutos.

Vol 7: Sci-Fi & Pop Culture
7x3E

Ernest Cline

2011

Cline incluyó un 'huevo de pascua' secreto en el propio libro impreso que premiaba al primer lector con un automóvil DeLorean.

Vol 7: Sci-Fi & Pop Culture
7x3D

© 2026 Shellaquiles Org · Hit-Tazos Tech v1.0.0-rc2 · Licencia MIT · Descarga gratuita y actualizaciones en https://shellaquiles.org