**Engineer**

# Presentación completa

Engineer ApS al completo: servicios con ejemplos de trabajos realizados, IA, plataformas, principios de trabajo y datos de la empresa en una página.

Lea la presentación completa a continuación: nuestros servicios y los trabajos que los respaldan, IA, plataformas, principios de trabajo y datos de la empresa. El índice también lleva a cada tema y a nuestro espacio de trabajo en Copenhague.

## Explorar todos los temas


- Quiénes somos — Engineer ApS es una consultora de software y TI con sede en Copenhague, fundada en 2025.
- Portafolio — Explore el portafolio completo de logros de ingeniería de Engineer ApS, cada uno con una reseña detallada.
- Servicios — Qué hace Engineer ApS: un catálogo de servicios de software, datos, nube e informática, cada uno respaldado por trabajos de proyectos reales.
- Tecnologías que utilizamos — Las tecnologías con las que construye Engineer ApS: lenguajes, plataformas de datos, nube e infraestructura, y cómo decidimos qué utilizar.
- Preparado para agentes de IA — Revisión de sitios y aplicaciones: rastreo, contenido legible, datos estructurados e interfaces para agentes, con implementación de alcance definido.
- Datos de empresa listos para agentes de IA — Conecte los registros de su empresa a un flujo de IA acordado, con fuentes claras, controles de calidad, permisos y un primer resultado trazable.
- Agentes de IA a medida para su sector — Agentes de IA a medida para un flujo de trabajo acordado, conectados a sus sistemas y evaluados con sus casos, con revisión humana y acceso delimitado.
- Seguridad para sus agentes de IA — Revisiones y controles de acceso para agentes de IA: inventario acordado, hallazgos priorizados, pruebas de permisos y registro de verificación.

- Su software como herramientas para agentes de IA — Operaciones de su software como herramientas descritas para agentes de IA, con permisos, comprobaciones de contrato y un despliegue planificado.
- Nuestro espacio de trabajo en Copenhague

# Construimos los sistemas **que hacen funcionar su empresa** — y respondemos por ellos.

Software, datos, nube e informática para mantenerlo todo en marcha — con sede en Copenhague desde 2025.

---

company
**Engineer**
**Fundada en 2025**

**Servicios de desarrollo de software y consultoría informática**

+45 50 10 61 84 · Copenhague, Dinamarca

Creado con esmero y cariño por **D.**

## Trabajamos con **las plataformas** que su empresa ya utiliza.

Apple Microsoft DigitalOcean AWS Google

**Vea toda la pila tecnológica**

# Preparados para los agentes de IA que envían sus clientes

## Localizable

Un archivo robots que indica qué pueden hacer los agentes, un mapa del sitio con fechas reales y cada cambio anunciado mediante IndexNow el día de su publicación.

## Legible

Una copia limpia en Markdown de cada página, un mapa llms.txt para modelos de lenguaje y datos estructurados que describen lo que hace su empresa.

## Utilizable

Formularios y controles accesibles, además de una API descrita para las operaciones de la aplicación seleccionadas para el acceso de agentes.

**Vea cómo funciona** Pregunte por su sitio →

# Datos de empresa con los que sus agentes de IA pueden responder

## Reunidos

Su CRM, ERP, unidades compartidas y archivos de correo reunidos en una base de datos a la que un agente accede por una única puerta descrita.

## Depurados

Identificadores acordados y controles de calidad conectan las fuentes de datos seleccionadas, con la gestión de duplicados y las reglas de validación documentadas.

## Con permisos

Los controles de acceso y el registro de auditoría siguen los permisos y los límites de datos acordados para la integración.

**Vea el trabajo con datos** Pregunte por sus datos →

# Agentes de IA creados para el trabajo de su sector

## Conectados

Conectados a su ERP, CRM, sistema de reservas y buzón mediante herramientas descritas, cada una con una tarea concreta y limitada.

## Medidos

Evaluados con casos reales escritos por sus especialistas, de modo que cada cambio en una instrucción, herramienta o modelo se prueba antes de publicarse.

## Supervisados

Lo que hace por sí solo y lo que aprueba una persona, acordado por escrito con usted, aplicado en las herramientas y trazado paso a paso.

**Vea cómo se construyen los agentes** Pregunte por su sector →

# Revisión de seguridad de los agentes de IA incluidos en el alcance

## Inventariados

Un inventario acordado de los agentes, credenciales y accesos a datos incluidos en el alcance, conservado junto a su configuración.

## Protegidos

Cada agente con identidad propia y privilegios mínimos; el texto que lee se trata como datos, nunca como órdenes.

## Registrados

Cada acción registrada junto a la persona para la que se realizó, donde el agente no puede modificarla, y alertas ante lo inusual.

**Vea cómo se protegen los agentes** Pregunte por sus agentes →

# El software que utiliza, convertido en herramientas que los agentes pueden invocar

## Desentrañado

Las reglas ocultas en pantallas, formularios y procedimientos almacenados, identificadas, documentadas y trasladadas a una sola capa.

## Expuesto

Cada operación se ofrece como una herramienta tipada y descrita mediante una API o un servidor MCP, que actúa solo dentro de los permisos de su usuario.

## En funcionamiento

Una integración de alcance definido puede exponer una operación cada vez junto a las pantallas existentes; los cambios de la aplicación se acuerdan tras revisar sus interfaces.

**Vea cómo se abre el software** Pregunte por su software →

Conozca el trabajo **en su contexto** — las decisiones, la entrega y los resultados comunicados.

**Vea el catálogo completo**

Un solo colaborador para **38 servicios** en seis disciplinas.

Plataforma y backend, datos y bases de datos, nube y DevOps, consultoría y liderazgo, soporte informático, marca y marketing.

**Explore el catálogo**

{ Ingeniería de software }

# Decidir cómo construir el sistema completo

Arquitectura de plataformas y soluciones

La estructura del sistema se define primero sobre el papel para que las piezas encajen y los costes se conozcan antes de escribir código.

Concurrencia estricta Swift 6 sobre C

Adopté la concurrencia estricta completa de Swift 6 sin actores, conectando un bucle de eventos C bloqueante con el actor principal mediante un productor, un consumidor y un único orden, tras constatar que una tarea por evento pierde el orden del que depende la interfaz.

### Un flujo de eventos limitado para una aplicación macOS

Limité los búferes de eventos y la generación de registros tras un incidente de macOS documentado con un crecimiento de 41 MB por segundo y 111 GB de páginas comprimidas en una máquina de 36 GB; una ejecución completa de pruebas documentada redujo las líneas de registro del núcleo de 610.996 a 1.411.

### Cliente macOS nativo en Swift 6

Desarrollé un cliente de mensajería nativo para macOS en Swift 6 y SwiftUI — 11.141 líneas en 53 archivos — sobre la interfaz C de un núcleo Rust enlazado como biblioteca estática desde una revisión fijada.

Ver los 20 logros

## Cómo trabajamos

### Primero, comprender

Conocemos el problema antes de proponer una solución. La descripción con la que llega inicia esa conversación; no la cierra.

### Responder por el trabajo

Acordamos la operación continuada o el soporte como parte del encargo. El traspaso deja claras las responsabilidades y la siguiente vía de contacto.

### Construir para durar

Durabilidad antes que rapidez, claridad antes que ingenio. El sistema debe seguir siendo manejable para las personas que lo hereden.

{ Ingeniería de software }

# El motor de su aplicación o sitio web

Desarrollo de backend y API

La parte que el usuario no ve: dónde se guardan los datos, qué reglas se aplican y cómo pueden comunicarse otros sistemas con ella.

### La mitad sin utilizar del núcleo, implementada

Implementé la mitad del núcleo de mensajería que la aplicación no utilizaba — transferencia de copias de seguridad, mensajes temporales, edición y reenvío de mensajes, invitaciones verificadas, proxies y política de cifrado —, ejecutando cada prueba contra la biblioteca real, sin simulaciones.

### Analizador que verifica cada llamada FFI

Escribí un analizador que lee la cabecera C real de 7.308 líneas y verifica cada punto de llamada, cada constante de enumeración y que toda clase propietaria de un puntero sea final, después de que una cabecera provisional escrita a mano permitiera compilar y enlazar llamadas a tres funciones eliminadas que causaban fallos al ejecutarse.

### Concurrencia estricta Swift 6 sobre C

Adopté la concurrencia estricta completa de Swift 6 sin actores, conectando un bucle de eventos C bloqueante con el actor principal mediante un productor, un consumidor y un único orden, tras constatar que una tarea por evento pierde el orden del que depende la interfaz.

**Ver los 32 logros**

{ Ingeniería de software }

# Las pantallas que utiliza la gente

Desarrollo de frontend

Interfaces que cargan rápido, funcionan en el teléfono y se comportan igual en los navegadores que utilizan sus clientes.

### Sistema de diseño macOS de 48 tokens

Desarrollé un sistema visual de 48 tokens sobre el material de vidrio propio de la plataforma y escribí el analizador que rechaza números sin significado declarado, tamaños de fuente fijados directamente y animaciones que ignoran la preferencia de movimiento reducido.

### 17 iconos con nombre para lectores de pantalla

Asigné nombres accesibles a todos los controles de la interfaz compuestos solo por un icono tras encontrar el botón de envío anunciado como «arrow up circle, button», y escribí el analizador que exige una etiqueta a no más de ocho líneas del icono.

### Un flujo de eventos limitado para una aplicación macOS

Limité los búferes de eventos y la generación de registros tras un incidente de macOS documentado con un crecimiento de 41 MB por segundo y 111 GB de páginas comprimidas en una máquina de 36 GB; una ejecución completa de pruebas documentada redujo las líneas de registro del núcleo de 610.996 a 1.411.

**Ver los 30 logros**

{ Ingeniería de software }

## Un producto que funciona, de principio a fin

Desarrollo integral de producto

Un equipo que lleva una idea del boceto a un producto al que sus clientes pueden acceder, sin pasarla entre tres proveedores.

### Cliente macOS nativo en Swift 6

Desarrollé un cliente de mensajería nativo para macOS en Swift 6 y SwiftUI — 11.141 líneas en 53 archivos — sobre la interfaz C de un núcleo Rust enlazado como biblioteca estática desde una revisión fijada.

### Generador de CV basado en base de datos

Desarrollé en Python un generador de CV, referencias, portafolios y cartas de presentación basado en base de datos — 41 módulos y 10.580 líneas — que genera seis formatos de salida desde una fuente SQLite de 23 tablas ensamblada mediante un flujo idempotente de 19 pasos.

### Sitio Hugo multilingüe en 3 idiomas

Desarrollé un sitio estático multilingüe en Hugo con 108 archivos de plantillas, incluidos 53 parciales, que publica cada página en cuatro representaciones desde un árbol de contenido en tres idiomas, y también en siete subdominios especializados construidos desde el mismo árbol.

**Ver los 16 logros**

{ Ingeniería de software }

# Un sitio web que puede actualizar su equipo

Desarrollo de sitios web y CMS

Un sitio rápido, fácil de encontrar y editable por su propio personal, sin una factura por cada cambio de texto.

## Espejo onion de Tor con una dirección legible

Publiqué el sitio como espejo onion de Tor en una dirección legible que comienza por engineer, con la clave generada fuera del servidor para que nunca llegara a este repositorio ni a un portátil, y lo mantuve como la única de las cuatro puertas de entrada que nunca se contabiliza.

## Identidad visual creada a partir de 2 dibujos

Desarrollé los conjuntos de iconos y favicons de la empresa a partir de dos dibujos originales, con todos los recursos derivados regenerados mediante scripts y una comprobación que falla si se confirma un archivo derivado anterior a su fuente.

## 106 de 106 comprobaciones del perfil PDF/UA-1

Registré la validación PDF/UA-1 en nueve documentos, con 106 de 106 comprobaciones superadas en el perfil de accesibilidad configurado, mientras la variante de archivo incumplía una de 146 reglas.

**Ver los 22 logros**

{ Ingeniería de software }

# Aparecer cuando sus clientes preguntan a una IA

Preparación para agentes de IA

Un sitio o una aplicación que los asistentes de IA de sus clientes puedan encontrar, leer y utilizar, con comprobaciones en cada versión en lugar de una promesa inicial.

## Desarrollo organizado en 51 iniciativas escritas

Organicé el desarrollo del sitio en 51 iniciativas escritas repartidas en 54 documentos de planificación, cada uno con tareas pendientes, mediciones y decisiones descartadas, incluidas seis de ocho conclusiones de preparación para agentes rechazadas con su motivo registrado.

## Sitemap corregido: fechas que eran iguales

Corregí un sitemap en el que 172 de 176 URL compartían una misma fecha de modificación, tomando la fecha del historial git tras constatar que la exportación reescribe todos los archivos en cada ejecución.

## 963 bloques de metadatos por un único grafo

Sustituí 963 bloques anónimos de datos estructurados que repetían la empresa 1.671 veces por un único grafo enlazado de 16 tipos con identificadores de origen estables.

**Ver los 5 logros**

{ Ingeniería de software }

## Permitir que la IA utilice el software que ya tiene

Herramientas para agentes de IA y servidores MCP

Su software actual disponible como herramientas que un agente de IA puede invocar, sin reescribir lo que ya funciona.

### Panel interno de operaciones de la plataforma

Desarrollé el panel interno de operaciones — unas 40 rutas de administración y 128 componentes para reclamaciones de titularidad, moderación, indicadores de funcionalidades, caché y diagnóstico — para operar la plataforma sin acceder directamente a la base de datos.

### Trabajo lento trasladado a una cola River

Saqué las tareas lentas del procesamiento de solicitudes y las trasladé a una cola River — 15 módulos de trabajadores, 8 tareas programadas y 20 tareas pg_cron — para responder a las solicitudes mientras el trabajo continúa en segundo plano.

### Contrato de API generado desde la base de datos

Generé el contrato de la API desde la base de datos hacia las capas externas — OpenAPI, un cliente TypeScript tipado de 44.076 líneas, 61 manejadores simulados y los límites que aplica la interfaz — con una comprobación en cada paso que falla ante cualquier divergencia.

Ver los 5 logros

{ Ingeniería de software }

## Mapas que responden preguntas

Soluciones GIS y geoespaciales

Todo lo que ocurre en un lugar — barcos, vehículos, parcelas, sensores — representado en un mapa que se puede consultar, medir y compartir.

### Un producto cartográfico de principio a fin

Diseñé la arquitectura, desarrollé y mantuve infraestructura, procesamiento de datos y una aplicación de mapas durante 2 años, cubriendo tanto entregas de software como mantenimiento operativo.

### Más de 30 proyectos GIS: desarrollo y soporte

Dirigí el desarrollo, despliegue y soporte de más de 30 proyectos GIS con PostgreSQL, Bash, Python, JavaScript, GDAL, ArcGIS, PostGIS y Mapbox.

### Entrega de 30 proyectos GIS satisfactorios

Gestioné la ejecución de 30 proyectos GIS satisfactorios, dirigiendo la entrega de soluciones innovadoras en este ámbito.

**Ver los 13 logros**

{ Ingeniería de software }

## Que resulte evidente cómo se utiliza

Diseño UI/UX y sistemas de diseño

Distribución, textos y recorridos organizados para que las personas terminen lo que venían a hacer en vez de llamar a soporte.

### Identidad visual creada a partir de 2 dibujos

Desarrollé los conjuntos de iconos y favicons de la empresa a partir de dos dibujos originales, con todos los recursos derivados regenerados mediante scripts y una comprobación que falla si se confirma un archivo derivado anterior a su fuente.

### Errores que dejan de culpar al usuario

Reescribí los mensajes de error de la aplicación conforme a un criterio de tono documentado después de que un inicio de sesión rechazado culpara al usuario de escribir mal la contraseña cuando el proveedor exigía una específica para aplicaciones, y lo cubrí con una prueba que identifica al proveedor.

### Sistema de diseño macOS de 48 tokens

Desarrollé un sistema visual de 48 tokens sobre el material de vidrio propio de la plataforma y escribí el analizador que rechaza números sin significado declarado, tamaños de fuente fijados directamente y animaciones que ignoran la preferencia de movimiento reducido.

**Ver los 20 logros**

{ Ingeniería de software }

# Funcionar en más de un idioma

Internacionalización y localización

Software que se lee con naturalidad en cada mercado donde vende, incluidas las fechas, las monedas y la forma de ordenar los nombres.

## Comprobación de cifras entre idiomas

Desarrollé una comprobación de contenido entre idiomas que falla si una traducción omite una cifra del inglés o usa una notación ajena al idioma; detectó dos descripciones danesas sin una métrica y dieciséis pasajes ucranianos con comillas del estilo inglés.

## Todas las cadenas en un árbol de contenido

Trasladé todas las cadenas visibles para el usuario de Python a un árbol de 874 archivos de contenido en tres idiomas tras encontrar traducciones muertas cuya inutilidad no se veía y una comprobación que evaluaba silenciosamente solo un tercio de los logros.

## Generador de CV basado en base de datos

Desarrollé en Python un generador de CV, referencias, portafolios y cartas de presentación basado en base de datos — 41 módulos y 10.580 líneas — que genera seis formatos de salida desde una fuente SQLite de 23 tablas ensamblada mediante un flujo idempotente de 19 pasos.

**Ver los 8 logros**

{ Ingeniería de software }

# Un asistente de IA que conoce su actividad

Desarrollo de agentes de IA a medida

Un agente conectado a los sistemas que utiliza, probado con casos definidos por sus especialistas, supervisado por su personal y sin depender de un único modelo.

## Seis recolectores declaran lo que omitieron

Hice que seis recolectores de informes declararan la parte de sus entradas que nunca habían leído — archivos no abiertos, sondeos no instalados y líneas del diario no reconocidas — y ajusté los umbrales a 116 muestras medidas tras constatar que una recolección ciega y un servidor sano generan la misma página.

## Pruebas para los propios verificadores

Escribí pruebas para los propios verificadores tras constatar que un verificador que solo recibe entradas correctas puede acabar informando de éxito por no haber leído nada; introduje una falta de ortografía para confirmar su detección y obtuve un intervalo de identificadores de la base de datos en vez de fijarlo en la prueba.

## Control de commits con 22 analizadores

Desarrollé un control obligatorio de commits con 22 analizadores de una línea y cinco que requieren una explicación, sin nivel de advertencias ni supresiones en línea permitidas, que cubre HTML, CSS, JavaScript, Python, YAML, Markdown, Shell, enlaces, ortografía, secretos y tipografía.

**Ver los 5 logros**

{ Datos y bases de datos }

# Un lugar adecuado para guardar sus datos

Diseño y modelado de bases de datos

Una estructura que refleja cómo funciona el negocio para poder responder también a las preguntas que surjan más adelante.

### Generador de CV basado en base de datos

Desarrollé en Python un generador de CV, referencias, portafolios y cartas de presentación basado en base de datos — 41 módulos y 10.580 líneas — que genera seis formatos de salida desde una fuente SQLite de 23 tablas ensamblada mediante un flujo idempotente de 19 pasos.

### Espacios de contratación y carrera marítima

Construí espacios de contratación y de carrera profesional para marinos; la instantánea de implementación registrada contiene 151 funciones almacenadas en 48 tablas para vacantes, candidaturas, certificados, progresión de rango y tiempo de embarque.

### Monedero de fidelidad, ligas e insignias

Desarrollé el sistema de fidelidad y reputación — 67 funciones sobre un registro de 31 tablas, con ligas, insignias y una tienda de canje —, bloqueando la fila del saldo para cerrar la ventana de doble gasto.

**Ver los 14 logros**

{ Datos y bases de datos }

# Cuidar de la base de datos

Administración de bases de datos (DBA)

Una persona responsable del sistema en el que se apoya toda la empresa: que funcione, tenga copias de seguridad y no se deteriore sin que nadie lo advierta.

### 1.022 migraciones con control de coherencia

Mantuve la coherencia del esquema durante 1.022 migraciones con un control CI que crea la base de datos por dos vías — una instalación nueva y una instalación seguida de todas las migraciones — y falla si los resultados difieren.

### Copias y recuperación ante desastres como código

Configuré la retención de las copias de seguridad de la base de datos como infraestructura como código, audité la capacidad de recuperación y documenté el procedimiento de restauración, dejando explícitas las carencias pendientes para evitar descubrirlas durante un incidente.

### 600 flujos ETL/ELT en PL/pgSQL

Desarrollé 600 flujos ETL/ELT basados en PL/pgSQL para simplificar procesos complejos de tratamiento de datos en varios entornos PostgreSQL de desarrollo y producción.

**Ver los 9 logros**

{ Datos y bases de datos }

## Acelerar las partes que van lentas

Optimización del rendimiento de bases de datos

Detectar las consultas y los índices que faltan en un sistema cada año más pesado y corregir lo que realmente importa.

### Claves ordenadas temporalmente para un modelo de datos en crecimiento

Adopté identificadores UUID v7 ordenados temporalmente (PostgreSQL 18) como claves de las entidades para reducir la fragmentación de los índices B-tree y acelerar las consultas.

### PostgreSQL 10 veces más rápido

Multipliqué por 10 el rendimiento de PostgreSQL mediante índices, particionamiento y optimización de consultas, mejorando la eficiencia de la base de datos para datos de usuarios, organizaciones cliente, información geoespacial y series temporales eléctricas.

### 100 almacenes PostgreSQL, MS SQL y BigQuery

Diseñé la arquitectura, creé y administré 100 bases de datos de almacenes de datos en PostgreSQL, MS SQL y Google BigQuery, principalmente con datos GIS y series temporales, y optimicé su rendimiento y escalabilidad.

**Ver los 6 logros**

{ Datos y bases de datos }

# Dejar atrás el sistema antiguo

Migración y modernización de bases de datos

Trasladar los datos de un sistema envejecido a otro con soporte, manteniendo el negocio en funcionamiento durante el proceso.

### Todas las cadenas en un árbol de contenido

Trasladé todas las cadenas visibles para el usuario de Python a un árbol de 874 archivos de contenido en tres idiomas tras encontrar traducciones muertas cuya inutilidad no se veía y una comprobación que evaluaba silenciosamente solo un tercio de los logros.

### 1.022 migraciones con control de coherencia

Mantuve la coherencia del esquema durante 1.022 migraciones con un control CI que crea la base de datos por dos vías — una instalación nueva y una instalación seguida de todas las migraciones — y falla si los resultados difieren.

### Mapas GIS 10 veces más rápidos con PostgreSQL

Multipliqué por 10 el rendimiento de una aplicación de mapas geográficos mediante una migración estratégica de MSSQL a PostgreSQL, incluida la transición de los datos, y optimicé el procesamiento y la seguridad de los datos.

{ Datos y bases de datos }

# Que sus datos vayan de A a B por sí solos

Desarrollo de flujos de datos (ETL/ELT)

Las cifras llegan donde hacen falta según un calendario, sin que nadie tenga que copiar una hoja de cálculo al terminar el mes.

### 495 páginas generadas desde SQLite

Generé 495 páginas de logros en tres idiomas mediante una exportación SQLite de solo lectura, con las direcciones de página definidas como datos para que corregir una frase dejara de mover la página y romper su enlace.

### Analítica propia dentro de PostgreSQL

Desarrollé analítica de producto propia en PostgreSQL — 47 funciones sobre tablas de eventos particionadas que se depuran automáticamente —, seudonimizada con una sal rotatoria y supeditada al consentimiento del visitante.

### Un producto cartográfico de principio a fin

Diseñé la arquitectura, desarrollé y mantuve infraestructura, procesamiento de datos y una aplicación de mapas durante 2 años, cubriendo tanto entregas de software como mantenimiento operativo.

**Ver los 14 logros**

{ Datos y bases de datos }

## Un lugar donde todas las cifras coinciden

Diseño de almacenes de datos

Cifras de ventas, operaciones y finanzas reunidas para que dos departamentos no lleven dos respuestas distintas a la misma reunión.

### 100 almacenes PostgreSQL, MS SQL y BigQuery

Diseñé la arquitectura, creé y administré 100 bases de datos de almacenes de datos en PostgreSQL, MS SQL y Google BigQuery, principalmente con datos GIS y series temporales, y optimicé su rendimiento y escalabilidad.

### Diseño y operación de 6 flujos ETL/ELT

Diseñé, implementé y administré 6 flujos ETL/ELT con Google BigQuery, MSSQL, PostgreSQL, scripts de Shell, PL/pgSQL y Transact-SQL, integrando los datos para su procesamiento eficiente mediante una API en Python.

### Flujo de datos geográficos 50 veces más rápido

Registré una mejora de 50 veces en el flujo de procesamiento de datos geográficos en el trabajo de la fundadora para Utiligize mediante cambios en SQL y en los modelos de datos en PostgreSQL, MS SQL y Google Cloud BigQuery.

{ Datos y bases de datos }

# Ver qué está ocurriendo

Análisis de datos y paneles de BI

Paneles e informes con las pocas cifras de las que realmente depende una decisión, actualizados sin tener que pedirlo.

### Analítica propia dentro de PostgreSQL

Desarrollé analítica de producto propia en PostgreSQL — 47 funciones sobre tablas de eventos particionadas que se depuran automáticamente —, seudonimizada con una sal rotatoria y supeditada al consentimiento del visitante.

### Sistema de gestión del tiempo e informes

Diseñé un sistema de gestión del tiempo y elaboración de informes que permaneció en producción durante años sin modificaciones importantes.

### Análisis e informes periódicos de progreso

Entregué análisis e informes periódicos al CEO, convirtiendo los KPI de ingeniería y el estado de las entregas en información para tomar decisiones.

**Ver los 11 logros**

{ Datos y bases de datos }

# Confiar en las cifras

Gobernanza y calidad de datos

Reglas y comprobaciones que detectan datos incorrectos al entrar y aclaran quién responde de cada cifra y de dónde procede.

### 13 direcciones medidas sin rastreo

Medí trece direcciones únicamente a partir del registro de accesos del servidor web — sin cookies, rastreadores ni servicios de terceros —, enmascarando la red del cliente al escribir cada línea e incorporando el resultado a un archivo permanente de 21 recuentos diarios y 20 facetas mensuales.

### Conocer la licencia de todo lo que se publica

Audité la compatibilidad de licencias de 644 crates Rust en cada compilación y probé que el control se activa al cambiar la licencia de una crate y al modificar la revisión fijada del núcleo sin regenerar los datos.

### Comprobación de cifras entre idiomas

Desarrollé una comprobación de contenido entre idiomas que falla si una traducción omite una cifra del inglés o usa una notación ajena al idioma; detectó dos descripciones danesas sin una métrica y dieciséis pasajes ucranianos con comillas del estilo inglés.

**Ver los 16 logros**

{ Datos y bases de datos }

## Datos que la IA puede utilizar de verdad

Preparación de datos para IA

Sus registros al alcance de los agentes de IA, una única versión de cada cliente, acceso limitado a lo que corresponde y cada respuesta vinculada a su fuente.

### 963 bloques de metadatos por un único grafo

Sustituí 963 bloques anónimos de datos estructurados que repetían la empresa 1.671 veces por un único grafo enlazado de 16 tipos con identificadores de origen estables.

### 495 páginas generadas desde SQLite

Generé 495 páginas de logros en tres idiomas mediante una exportación SQLite de solo lectura, con las direcciones de página definidas como datos para que corregir una frase dejara de mover la página y romper su enlace.

### 1.022 migraciones con control de coherencia

Mantuve la coherencia del esquema durante 1.022 migraciones con un control CI que crea la base de datos por dos vías — una instalación nueva y una instalación seguida de todas las migraciones — y falla si los resultados difieren.

**Ver los 6 logros**

{ Nube, infraestructura y DevOps }

# El lugar donde funciona el software

Infraestructura en la nube y migración

Servidores dimensionados para el uso real, trasladados a la nube o fuera de ella, con la factura mensual calculada de antemano.

## Segundo servidor aprovisionado desde código

Automaticé el aprovisionamiento de un segundo servidor en un segundo proveedor de nube y creé el cortafuegos antes que la máquina para que naciera protegida, con los cortafuegos de ambos proveedores configurados directamente mediante sus API REST para evitar una colección de terceros.

## Contenedores con Podman y Quadlet

Desplegué una plataforma de contenedores con Podman y Quadlet bajo systemd en lugar de Docker, porque la publicación predeterminada de puertos en redes bridge de Docker puede eludir las reglas UFW, y asigné a los despliegues un usuario sin privilegios con una única orden fija, en vez de root.

## Servicios públicos en un servidor pequeño

Alojé los servicios públicos de la empresa en un host de 512 MB y un solo núcleo — una forja git, un servidor web para siete dominios, Tor, dos servidores de protocolos alternativos y bloqueo de intrusiones — tomando los 464 MB de memoria utilizable como restricción de diseño; la automatización de copias de seguridad es una implementación aparte.

**Ver los 9 logros**

{ Nube, infraestructura y DevOps }

# Servidores que se pueden reconstruir a partir de una configuración versionada

Infraestructura como código

Toda la configuración escrita en archivos para poder recrearla con exactitud y evitar que exista solo en la memoria de una persona.

## Espejo onion de Tor con una dirección legible

Publiqué el sitio como espejo onion de Tor en una dirección legible que comienza por engineer, con la clave generada fuera del servidor para que nunca llegara a este repositorio ni a un portátil, y lo mantuve como la única de las cuatro puertas de entrada que nunca se contabiliza.

## Ensayo del hook CI: 2 defectos detectados

Ensayé el hook CI del servidor de la forja y encontré dos defectos imposibles de alcanzar leyendo el archivo: una alternativa que introducía un argumento irresoluble en la entrada del hook y una variable de entorno de git que acompañaba al control hasta el checkout y hacía fallar 19 pruebas.

## Secretos operativos separados por servidor

Separé los cuatro secretos operativos de cada servidor tras constatar que dos servidores que comparten una señal de actividad alertan menos que dos señales independientes, y que una contraseña compartida de copias de seguridad puede exponer las copias de ambos servidores, incluso en repositorios separados, si alguien la obtiene y puede leer los datos cifrados y los archivos de claves de ambos repositorios.

Ver los 17 logros

{ Nube, infraestructura y DevOps }

# Publicar cambios sin sobresaltos

DevOps y automatización CI/CD

Cambios probados, publicados y reversibles, para que lleguen a sus clientes un martes por la tarde en vez de un viernes por la noche.

### Un criterio de historial para 4 repositorios

Apliqué un único criterio de historial a cuatro repositorios — commits convencionales, sin emojis ni líneas de atribución y con un hook de mensajes de commit —, junto con 48 documentos de instrucciones que regulan la forma de trabajar.

### Conocer la licencia de todo lo que se publica

Audité la compatibilidad de licencias de 644 crates Rust en cada compilación y probé que el control se activa al cambiar la licencia de una crate y al modificar la revisión fijada del núcleo sin regenerar los datos.

### La mitad sin utilizar del núcleo, implementada

Implementé la mitad del núcleo de mensajería que la aplicación no utilizaba — transferencia de copias de seguridad, mensajes temporales, edición y reenvío de mensajes, invitaciones verificadas, proxies y política de cifrado —, ejecutando cada prueba contra la biblioteca real, sin simulaciones.

**Ver los 43 logros**

{ Nube, infraestructura y DevOps }

# Software que funciona igual en todas partes

Contenedores y orquestación

Software empaquetado para comportarse igual en un portátil, un servidor de pruebas y producción, y ampliar su capacidad cuando aumente la actividad.

### Contenedores con Podman y Quadlet

Desplegué una plataforma de contenedores con Podman y Quadlet bajo systemd en lugar de Docker, porque la publicación predeterminada de puertos en redes bridge de Docker puede eludir las reglas UFW, y asigné a los despliegues un usuario sin privilegios con una única orden fija, en vez de root.

### Menor exposición en cada unidad systemd

Reduje la exposición de aislamiento systemd en todas las unidades que instala la plataforma — un servicio de monitorización por ausencia de señal pasó de 9,6 UNSAFE a 1,5 y una forja git expuesta a Internet de 8,3 EXPOSED a 1,5 — y añadí una comprobación del analizador durante la convergencia tras encontrar una directiva mal escrita que se ignoraba silenciosamente en tres plantillas de unidades.

### Flujos CI/CD en GitHub Actions

Desarrollé flujos CI/CD en GitHub Actions con una imagen de producción distroless para la interfaz y promoción entre varios entornos.

**Ver los 5 logros**

{ Nube, infraestructura y DevOps }

## Saberlo antes que sus clientes

Fiabilidad de sistemas y monitorización

Monitorización y alertas que avisan cuando un fallo aún es pequeño, y cifras que muestran con qué frecuencia se mantiene así.

### Seis recolectores declaran lo que omitieron

Hice que seis recolectores de informes declararan la parte de sus entradas que nunca habían leído — archivos no abiertos, sondeos no instalados y líneas del diario no reconocidas — y ajusté los umbrales a 116 muestras medidas tras constatar que una recolección ciega y un servidor sano generan la misma página.

### 13 direcciones medidas sin rastreo

Medí trece direcciones únicamente a partir del registro de accesos del servidor web — sin cookies, rastreadores ni servicios de terceros —, enmascarando la red del cliente al escribir cada línea e incorporando el resultado a un archivo permanente de 21 recuentos diarios y 20 facetas mensuales.

### Un flujo de eventos limitado para una aplicación macOS

Limité los búferes de eventos y la generación de registros tras un incidente de macOS documentado con un crecimiento de 41 MB por segundo y 111 GB de páginas comprimidas en una máquina de 36 GB; una ejecución completa de pruebas documentada redujo las líneas de registro del núcleo de 610.996 a 1.411.

**Ver los 23 logros**

{ Nube, infraestructura y DevOps }

# Recuperarlo todo después de un mal día

Copias de seguridad y recuperación ante desastres

Copias de seguridad que se prueban en vez de darse por buenas y un plan escrito para la mañana en que un servidor no vuelva a encenderse.

## Secretos operativos separados por servidor

Separé los cuatro secretos operativos de cada servidor tras constatar que dos servidores que comparten una señal de actividad alertan menos que dos señales independientes, y que una contraseña compartida de copias de seguridad puede exponer las copias de ambos servidores, incluso en repositorios separados, si alguien la obtiene y puede leer los datos cifrados y los archivos de claves de ambos repositorios.

## Automatización de copias con un método explícito de restauración

Implementé automatización de copias con restic, retención, comprobaciones de integridad y un flujo mensual de prueba de restauración; documenté la activación del almacenamiento y la verificación de restauración real como pasos operativos separados.

## Copias y recuperación ante desastres como código

Configuré la retención de las copias de seguridad de la base de datos como infraestructura como código, audité la capacidad de recuperación y documenté el procedimiento de restauración, dejando explícitas las carencias pendientes para evitar descubrirlas durante un incidente.

Ver los 5 logros

{ Nube, infraestructura y DevOps }

# Mantenerlo todo en funcionamiento

Administración de sistemas

El cuidado diario de servidores, correo, cuentas y actualizaciones para que los servicios de los que depende todo el mundo sigan funcionando sin sorpresas.

### Espejo onion de Tor con una dirección legible

Publiqué el sitio como espejo onion de Tor en una dirección legible que comienza por engineer, con la clave generada fuera del servidor para que nunca llegara a este repositorio ni a un portátil, y lo mantuve como la única de las cuatro puertas de entrada que nunca se contabiliza.

### Seis recolectores declaran lo que omitieron

Hice que seis recolectores de informes declararan la parte de sus entradas que nunca habían leído — archivos no abiertos, sondeos no instalados y líneas del diario no reconocidas — y ajusté los umbrales a 116 muestras medidas tras constatar que una recolección ciega y un servidor sano generan la misma página.

### Ensayo del hook CI: 2 defectos detectados

Ensayé el hook CI del servidor de la forja y encontré dos defectos imposibles de alcanzar leyendo el archivo: una alternativa que introducía un argumento irresoluble en la entrada del hook y una variable de entorno de git que acompañaba al control hasta el checkout y hacía fallar 19 pruebas.

Ver los 28 logros

{ Nube, infraestructura y DevOps }

# Conectar sus oficinas y a su personal

Configuración de redes y VPN

Wi-Fi, cortafuegos y acceso remoto configurados para trabajar desde cualquier lugar y permitir el acceso solo a las personas autorizadas.

### Segundo servidor aprovisionado desde código

Automaticé el aprovisionamiento de un segundo servidor en un segundo proveedor de nube y creé el cortafuegos antes que la máquina para que naciera protegida, con los cortafuegos de ambos proveedores configurados directamente mediante sus API REST para evitar una colección de terceros.

### Zona DNS de 20 registros como código

Concilié declarativamente una zona DNS de 20 registros con la API de Cloudflare mediante puntos de entrada separados para auditoría y exportación BIND, y desactivé el proxy CDN por motivos de privacidad después de implementarlo.

### Reglas de cortafuegos verificadas por posición

Verifiqué las reglas del cortafuegos por su posición, además de por su existencia, leyendo la lista numerada de reglas y la cadena activa del filtro de paquetes, porque una regla presente puede no recibir nunca un paquete.

**Ver los 11 logros**

{ Nube, infraestructura y DevOps }

## Decidir quién puede acceder a qué

Seguridad y gestión de accesos

Cuentas, contraseñas y permisos organizados para que cada persona tenga exactamente el acceso que necesita y lo pierda el mismo día que deja la empresa.

### Espejo onion de Tor con una dirección legible

Publiqué el sitio como espejo onion de Tor en una dirección legible que comienza por engineer, con la clave generada fuera del servidor para que nunca llegara a este repositorio ni a un portátil, y lo mantuve como la única de las cuatro puertas de entrada que nunca se contabiliza.

### 13 direcciones medidas sin rastreo

Medí trece direcciones únicamente a partir del registro de accesos del servidor web — sin cookies, rastreadores ni servicios de terceros —, enmascarando la red del cliente al escribir cada línea e incorporando el resultado a un archivo permanente de 21 recuentos diarios y 20 facetas mensuales.

### Conocer la licencia de todo lo que se publica

Audité la compatibilidad de licencias de 644 crates Rust en cada compilación y probé que el control se activa al cambiar la licencia de una crate y al modificar la revisión fijada del núcleo sin regenerar los datos.

**Ver los 31 logros**

{ Nube, infraestructura y DevOps }

# Que su IA actúe solo donde le corresponde

Seguridad de agentes de IA

El acceso de cada agente definido y limitado a lo necesario, cada acción registrada donde el agente no puede alterarla y cada protección comprobada en lugar de darse por segura.

## Secretos operativos separados por servidor

Separé los cuatro secretos operativos de cada servidor tras constatar que dos servidores que comparten una señal de actividad alertan menos que dos señales independientes, y que una contraseña compartida de copias de seguridad puede exponer las copias de ambos servidores, incluso en repositorios separados, si alguien la obtiene y puede leer los datos cifrados y los archivos de claves de ambos repositorios.

## Regla de alcance tras una fuga de credenciales

Incorporé al repositorio una regla de alcance tras una reorganización que había introducido el inventario, los permisos de cortafuegos y textos de otra empresa, y mantuve los residuos en cuarentena bajo el detector de secretos, sin excluirlos.

## Comprobación de secretos en un segundo

Añadí una comprobación previa que ejecuta en aproximadamente un segundo el mismo código de convergencia sobre secretos locales del operador, después de que una ejecución de producción parcialmente aplicada fallara en su novena tarea con los ajustes de intercambio ya escritos en el servidor activo.

**Ver los 9 logros**

{ Consultoría y dirección }

# Una ingeniera sénior en la reunión

Dirección técnica y consultoría

Alguien de su lado de la mesa cuando una decisión sobre proveedores, presupuesto o arquitectura necesita una segunda opinión informada.

### Un criterio de historial para 4 repositorios

Apliqué un único criterio de historial a cuatro repositorios — commits convencionales, sin emojis ni líneas de atribución y con un hook de mensajes de commit —, junto con 48 documentos de instrucciones que regulan la forma de trabajar.

### Pruebas reducidas a 8,9 segundos

Pasé de seis pruebas que superaban un límite de sesenta segundos a 135 pruebas satisfactorias en 8,9 segundos al analizar el hilo principal y eliminar las dos llamadas en las que permanecía durante 3.989 de 4.017 muestras.

### Concurrencia estricta Swift 6 sobre C

Adopté la concurrencia estricta completa de Swift 6 sin actores, conectando un bucle de eventos C bloqueante con el actor principal mediante un productor, un consumidor y un único orden, tras constatar que una tarea por evento pierde el orden del que depende la interfaz.

**Ver los 26 logros**

{ Consultoría y dirección }

# Hacer crecer el equipo que ya tiene

Desarrollo de equipos y mentoría

Selección, incorporación y acompañamiento para que el conocimiento se quede con su personal en vez de marcharse con una consultora.

### Dirección de un equipo de soporte informático

Dirigí un equipo que ofrecía soporte informático, recuperación de datos y reparación de equipos a más de 1.000 clientes, manteniendo la calidad del servicio.

### 150 horas al mes ahorradas en 30 proyectos

Mejoré la eficiencia de 30 proyectos y ahorré 150 horas al mes al optimizar los flujos de trabajo y la gestión de recursos.

### Creación de un departamento de soporte técnico

Creé un departamento de soporte técnico que atendió a más de 10.000 clientes con servicios de soporte informático y resolución de incidencias, y registré las solicitudes en un CRM.

**Ver los 9 logros**

{ Consultoría y dirección }

## Llevar el proyecto a término

Gestión ágil de proyectos

Una persona que responda del alcance, las fechas y las conversaciones difíciles, para que el proyecto tenga una responsable en lugar de cinco opiniones.

### Un criterio de historial para 4 repositorios

Apliqué un único criterio de historial a cuatro repositorios — commits convencionales, sin emojis ni líneas de atribución y con un hook de mensajes de commit —, junto con 48 documentos de instrucciones que regulan la forma de trabajar.

### Analizador propio para la documentación

Sometí la documentación a un analizador propio — límite de líneas por archivo que solo puede reducirse, índice obligatorio y comprobación de rutas reales — que detectó seis de nueve planes marcados como terminados con tareas pendientes y catorce tareas obligatorias para commits que no figuraban en ningún documento.

### Desarrollo organizado en 51 iniciativas escritas

Organicé el desarrollo del sitio en 51 iniciativas escritas repartidas en 54 documentos de planificación, cada uno con tareas pendientes, mediciones y decisiones descartadas, incluidas seis de ocho conclusiones de preparación para agentes rechazadas con su motivo registrado.

**Ver los 16 logros**

{ Consultoría y dirección }

# Decidir qué construir

Estrategia de producto y requisitos

Convertir las necesidades del negocio en algo que se pueda desarrollar y decir con claridad qué no merece la pena construir.

## Errores que dejan de culpar al usuario

Reescribí los mensajes de error de la aplicación conforme a un criterio de tono documentado después de que un inicio de sesión rechazado culpara al usuario de escribir mal la contraseña cuando el proveedor exigía una específica para aplicaciones, y lo cubrí con una prueba que identifica al proveedor.

## Desarrollo organizado en 51 iniciativas escritas

Organicé el desarrollo del sitio en 51 iniciativas escritas repartidas en 54 documentos de planificación, cada uno con tareas pendientes, mediciones y decisiones descartadas, incluidas seis de ocho conclusiones de preparación para agentes rechazadas con su motivo registrado.

## Control de commits con 22 analizadores

Desarrollé un control obligatorio de commits con 22 analizadores de una línea y cinco que requieren una explicación, sin nivel de advertencias ni supresiones en línea permitidas, que cubre HTML, CSS, JavaScript, Python, YAML, Markdown, Shell, enlaces, ortografía, secretos y tipografía.

Ver los 19 logros

{ Consultoría y dirección }

## Dejarlo escrito para quien venga después

Documentación técnica

Traspasos, procedimientos operativos y manuales para que su personal pueda operar el sistema sin llamar a quien lo construyó.

### Seis recolectores declaran lo que omitieron

Hice que seis recolectores de informes declararan la parte de sus entradas que nunca habían leído — archivos no abiertos, sondeos no instalados y líneas del diario no reconocidas — y ajusté los umbrales a 116 muestras medidas tras constatar que una recolección ciega y un servidor sano generan la misma página.

### Un criterio de historial para 4 repositorios

Apliqué un único criterio de historial a cuatro repositorios — commits convencionales, sin emojis ni líneas de atribución y con un hook de mensajes de commit —, junto con 48 documentos de instrucciones que regulan la forma de trabajar.

### Errores que dejan de culpar al usuario

Reescribí los mensajes de error de la aplicación conforme a un criterio de tono documentado después de que un inicio de sesión rechazado culpara al usuario de escribir mal la contraseña cuando el proveedor exigía una específica para aplicaciones, y lo cubrí con una prueba que identifica al proveedor.

**Ver los 19 logros**

{ Soporte de TI y equipos }

## Ayuda informática para el día a día

Soporte informático y asistencia técnica

Resolver con rapidez y lenguaje claro los problemas del portátil, la impresora, el acceso o el correo, sin hacer sentir mal a nadie.

### Diagnóstico y reparación de portátiles

Diagnostiqué y reparé hardware de portátiles — pantallas, bisagras, teclados, paneles táctiles, placas base y alimentación — y resolví problemas de software en Linux, Windows y Mac.

### Datos recuperados de SD, HDD, SSD y RAID

Recuperé datos de una amplia variedad de medios — tarjetas SD, HDD, SSD, matrices RAID, unidades externas y sistemas Mac.

### Soporte 24/7 para unos 1.000 servidores IPTV/OTT

Presté soporte de infraestructura 24/7 a una plataforma de distribución IPTV/OTT, administrando unos 1.000 servidores y sistemas de los clientes para usuarios de China, Estados Unidos y Alemania.

**Ver los 7 logros**

{ Soporte de TI y equipos }

## Rescatar archivos de un equipo averiado

Reparación de equipos y recuperación de datos

Reparar lo que merece la pena y recuperar fotografías,
facturas o planos de un disco que ha dejado de funcionar.

### Diagnóstico y reparación de portátiles

Diagnostiqué y reparé hardware de portátiles — pantallas, bisagras, teclados, paneles táctiles, placas base y alimentación — y resolví problemas de software en Linux, Windows y Mac.

### Datos recuperados de SD, HDD, SSD y RAID

Recuperé datos de una amplia variedad de medios — tarjetas SD, HDD, SSD, matrices RAID, unidades externas y sistemas Mac.

### Gestión de 4.000 reparaciones de ordenadores

Gestioné 4.000 reparaciones de ordenadores para resolver con rapidez y eficacia problemas de hardware y software.

**Ver los 4 logros**

{ Marca y marketing }

# Que le encuentren y reconozcan su identidad

Marca, marketing y SEO

Un nombre, una imagen y un sitio web que aparezcan al buscar lo que vende y transmitan lo mismo allí donde esté presente.

### Identidad visual creada a partir de 2 dibujos

Desarrollé los conjuntos de iconos y favicons de la empresa a partir de dos dibujos originales, con todos los recursos derivados regenerados mediante scripts y una comprobación que falla si se confirma un archivo derivado anterior a su fuente.

### Sistema de diseño macOS de 48 tokens

Desarrollé un sistema visual de 48 tokens sobre el material de vidrio propio de la plataforma y escribí el analizador que rechaza números sin significado declarado, tamaños de fuente fijados directamente y animaciones que ignoran la preferencia de movimiento reducido.

### Sitemap corregido: fechas que eran iguales

Corregí un sitemap en el que 172 de 176 URL compartían una misma fecha de modificación, tomando la fecha del historial git tras constatar que la exportación reescribe todos los archivos en cada ejecución.

**Ver los 11 logros**

## Cuéntenos qué está construyendo.

Sin formulario, sin embudo, sin secuencia de ventas: solo un correo electrónico a una persona que lo leerá.

**Inicie la conversación**

## En el servidor, trabajamos con **Linux de código abierto** — sin licencias que pagar ni nada oculto.



Ubuntu



Debian

**Cómo operamos los servidores**

### Elija cómo se hace el trabajo

Elija un colaborador de ingeniería por cómo se hará el trabajo. Conectamos las personas, las decisiones y la entrega que su proyecto necesita.

**Explore nuestros servicios** Póngase en contacto →

### Haga preguntas concretas sobre la experiencia

Lea los trabajos y las formas de trabajar junto con «Presentación completa». Le proporcionan preguntas concretas sobre experiencia y responsabilidad.

**Casos prácticos** Póngase en contacto →

### Deje claras las responsabilidades

Empiece por identificar el resultado y las decisiones aún abiertas. Podemos determinar dónde aportaría valor nuestra participación y qué debería conservar su equipo.

**Vea cuánto cuesta** Póngase en contacto →

## Encuentre una forma de trabajar juntos

Cuéntenos quién necesita el resultado y quién se responsabilizará después de él. Podemos hablar de una forma de trabajar que encaje en su organización.

**Explore nuestros servicios**

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

**Creado con esmero y cariño por D.**

**CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N**

**Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.**

**Impreso desde https://engineer.company/es/overview/**