Construimos el software **que heredará su equipo** y lo escribimos para que pueda hacerlo.

Arquitectura, backends, frontends, sitios web, mapas, sistemas de diseño y más de un idioma: desde Copenhague.

Engineer company
Software

Software construido para mantenerse, no solo para entregarse

+45 50 10 61 84 · Copenhague, Dinamarca

Creado con esmero y cariño por D.

Backends, frontends y la arquitectura que los une, escritos para que la siguiente persona que abra el repositorio pueda modificarlos. Los registros de trabajo describen la situación inicial, lo realizado y el resultado.

Empiece con un flujo de trabajo, una integración o una parte de una aplicación. Podemos acordar una entrega pequeña con comprobaciones de aceptación, interfaces documentadas y un traspaso que permita a su equipo mantener lo entregado.

Trabajos de software · Precios y modalidades de contratación · Hablemos de su proyecto

Empiece por el resultado que necesita

Cuéntenos qué necesita conseguir su empresa. Conectamos software, datos e infraestructura en torno a las personas que los utilizan.

Explore nuestros servicios Póngase en contacto →

Vea cómo abordamos el trabajo

Encuentre el servicio que encaja con su objetivo y vea ejemplos de lo que se entregó y cómo.

Casos prácticos Póngase en contacto →

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

Vea el catálogo completo

Un solo colaborador para **11 servicios** de la arquitectura a la última pantalla.

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.

Construir para durar

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

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.

{ 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

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 →

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 →

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 →

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

Esta dirección muestra solo los trabajos de software. La empresa también trabaja con datos, opera infraestructura y asesora a equipos: esa parte se encuentra en engineer.company.
Vea todo lo que hace Engineer ApS →

Empiece con una mejora concreta

Un flujo de trabajo, un flujo de datos o un sistema que necesita mejorar puede ser una primera tarea útil. Podemos acordar su alcance juntos.

Vea cuánto cuesta Póngase en contacto →

Prepare la primera conversación

Indíquenos el resultado que necesita, qué existe ya y cuándo le gustaría tenerlo listo. Con esa información podemos hablar del siguiente paso.

Explore nuestros servicios

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

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



Ubuntu



Debian

Cómo operamos los servidores



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://software.engineer.company/es/