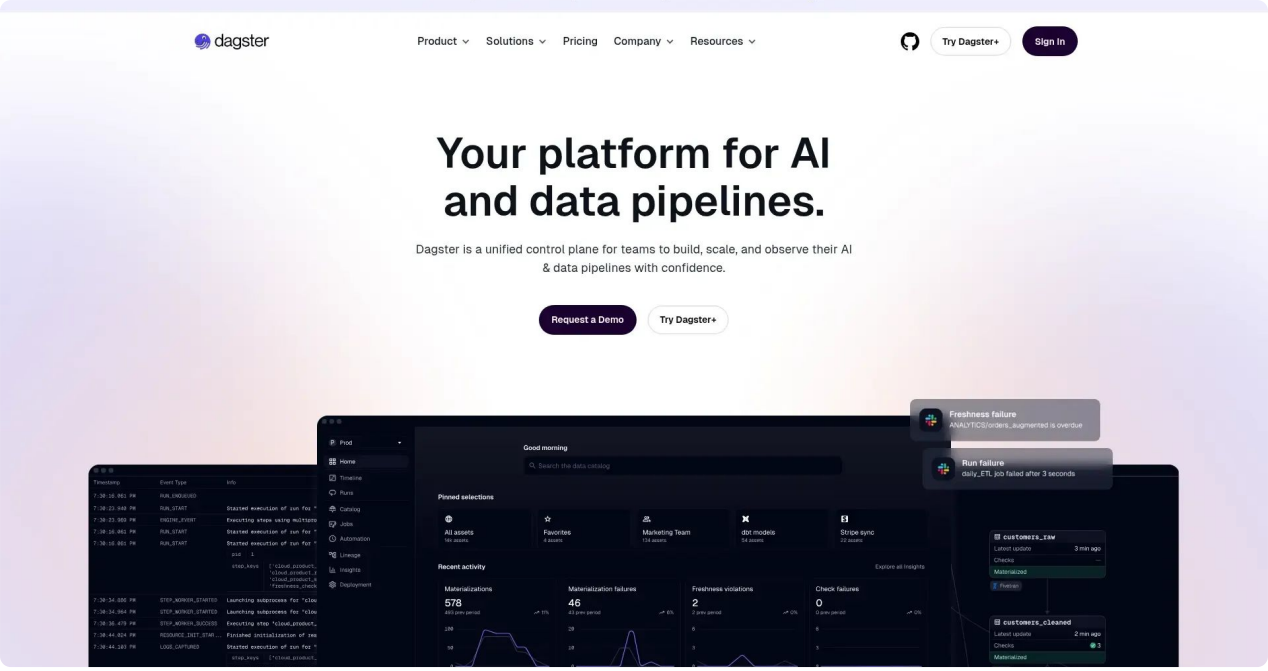

Dagster

Dagster es una plataforma de orquestación de datos de nueva generación diseñada para ingenieros, analistas y científicos de datos que buscan profesionalizar su infraestructura. A diferencia de los sistemas tradicionales, se centra en activos de datos como tablas y modelos de ML, permitiendo definir dependencias reales y estados finales. Es ideal para equipos que utilizan Python y requieren alta testabilidad, linaje detallado y observabilidad en flujos de trabajo complejos de ETL, ELT e IA.

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

Contenido del Dossier

INFORMACIÓN DE LA HERRAMIENTA

Qué y para quién es
Dagster es una plataforma de orquestación de datos de nueva generación diseñada para el desarrollo, producción y observación de flujos de trabajo de datos. A diferencia de los orquestadores tradicionales basados en tareas (como Airflow), Dagster se centra en los **activos de datos** (tablas, archivos, modelos de ML).
Está dirigido a ingenieros de datos, analistas y científicos de datos dentro de empresas que buscan profesionalizar su infraestructura de datos, priorizando la testabilidad, el linaje de datos y la capacidad de ejecución local.

Principal ventaja profesional
Su enfoque **"Asset-Centric"** (centrado en activos). Permite definir el estado final deseado de los datos y sus dependencias, facilitando un linaje claro y una detección de errores mucho más rápida que en sistemas basados simplemente en una secuencia de tareas inconexas.

Para quién no es
No es adecuado para equipos que no utilicen Python como lenguaje principal o para empresas con flujos de trabajo extremadamente simples (donde un simple cron job sea suficiente). También puede ser rechazado por departamentos que busquen herramientas estrictamente no-code o que no estén dispuestos a asumir la curva de aprendizaje de un modelo de programación declarativo.

Funcionalidades clave
- **Orquestación basada en activos:** Define dependencias entre conjuntos de datos reales, no solo entre scripts.
- **Entorno de desarrollo local (Dagster UI):** Interfaz web potente para visualizar, ejecutar y depurar pipelines localmente antes del despliegue.
- **Declarative Scheduling:** Los activos se actualizan automáticamente basándose en políticas de "frescura" de datos.
- **Observabilidad y Linaje:** Seguimiento detallado de cómo se transforma el dato desde el origen hasta el destino final.
- **Pruebas y tipado:** Facilita la creación de tests unitarios para los pipelines de datos, algo complejo en otros orquestadores.
- **Catálogo de datos integrado:** Permite buscar y entender el estado de cada tabla o modelo generado.

Precios
Dagster ofrece un modelo híbrido entre código abierto y servicios gestionados bajo la marca **Dagster+**.
- **Versión Gratuita (Open Source):** Completa y bajo licencia Apache 2.0. Permite el uso total de la herramienta pero requiere que la empresa gestione su propia infraestructura (Kubernetes, Docker, etc.).
- **Solo Plan:** ~10$ al mes (7.5k créditos incluidos). Ideal para proyectos individuales o validaciones técnicas iniciales.
- **Starter Plan:** ~100$ al mes (30k créditos incluidos). Incluye control de acceso basado en roles (RBAC) y búsqueda en catálogo.
- **Pro / Enterprise:** Precio bajo presupuesto. Ofrece despliegues ilimitados, soporte personalizado via Slack, cumplimiento de SLAs y seguridad avanzada (SAML, auditorías).
Nota: El sistema de créditos se basa en la ejecución de opeciones y materialización de activos ($0.03 por crédito excedente).

Perfil del usuario
- **Empresas:** Scale-ups tecnológicas, departamentos de datos en corporaciones con stacks modernos (Modern Data Stack) y empresas con fuertes necesidades de IA/ML.
- **Perfiles:** Data Engineers, Analytics Engineers, ML Ops y Arquitectos de Datos.

Nivel técnico requerido
- **Uso:** Alto. Requiere dominio fluido de **Python** y conceptos de bases de datos/ETL.
- **Instalación/Configuración:** Medio-Alto (para la versión OSS se requiere conocimiento de Docker, Kubernetes o despliegue en nubes como AWS/GCP).
- **Competencias necesarias:** Desarrollo de software (Git, CI/CD), SQL y manejo de APIs.

Ejemplos de uso profesional
- **Ciclo ELT/ETL:** Coordinación de la ingesta de datos desde SaaS (Fivetran/Airbyte) hacia almacenes de datos (Snowflake/BigQuery) y transformaciones posteriores con dbt.
- **Entrenamiento de Modelos ML:** Orquestar el re-entrenamiento de modelos de IA asegurando que las características (features) de entrada estén actualizadas.
- **Reporting y BI:** Asegurar que los cuadros de mando en herramientas como Looker o PowerBI reflejen datos válidos y alertar automáticamente si un activo crítico no se ha actualizado.

Uso y distribución
- **Versión web:** Panel de control centralizado (Cloud o auto-alojado).
- **Versión escritorio:** Herramienta visual Dagster UI (antes Dagit) para ejecución local.
- **CLI:** Interfaz de línea de comandos completa para gestión y despliegue.
- **Librerías:** Paquete Python disponible vía PyPI (pip install dagster).

Open source
El núcleo de Dagster es **Open Source** (Apache License 2.0). Todo el código necesario para orquestar y visualizar pipelines está disponible en su repositorio oficial de GitHub.

Integraciones
- **Facilidad de integración:** Media (requiere código Python para configurar los conectores).
- **API propia:** Dispone de una API de GraphQL muy robusta para interactuar con el orquestador de forma programática.
- **Integraciones nativas:** Amplio ecosistema que incluye:
- **Almacenamiento:** Snowflake, BigQuery, Redshift, Databricks.
- **Transformación:** dbt (integración de primer nivel con linaje a nivel de columna).
- **Ingesta:** Fivetran, Airbyte.
- **Infraestructura:** Kubernetes, Docker, AWS (S3, Lambda, ECS), GCP, Azure.

Notas finales
Información legal e infracción
La versión Dagster+ (Cloud) opera bajo términos de servicio de SaaS estándar con opciones para cumplimiento de **SOC2 Type II, HIPAA** y cifrado AES-256 en reposo. El código abierto es libre para uso comercial sin coste de licencia.

Otros
Es importante destacar que Dagster está desplazando a Airflow en muchas organizaciones modernas debido a que soluciona problemas de "deuda técnica" en los pipelines, permitiendo que el código de orquestación sea parte del ciclo de vida del desarrollo de software (pruebas unitarias y entornos de stagging reales).

Para más información:
- Sitio web oficial: https://dagster.io
- Documentación técnica: https://docs.dagster.io
- Precios y planes: https://dagster.io/pricing
- Github: https://github.com/dagster-io/dagster
- Discord de la comunidad: https://dagster.io/community

CONSEJOS DE IMPLANTACIÓN

Aplicación profesional
- **Tipos de empresa:** Scale-ups tecnológicas, departamentos de datos en corporaciones con arquitecturas modernas (Modern Data Stack) y empresas con alta carga en IA/ML que requieren linaje estricto.
- **Presupuesto:**
- **Open Source:** Gratuito (requiere inversión en personal para mantenimiento de infraestructura).
- **Cloud (Dagster+):** Plan Starter desde ~100$/mes; Plan Pro/Enterprise escalable según consumo de créditos (materialización de activos).
- **Puntos clave:** Sustituye el modelo tradicional de "tareas" por un modelo de "activos" (Asset-centric), permitiendo que el orquestador entienda qué dato se está creando y no solo qué script se está ejecutando.

Madurez digital requerida
- **Usuarios:** Ingenieros de datos y perfiles técnicos con dominio fluido de **Python**. No es una herramienta para usuarios de negocio o perfiles no-code.
- **Empresa:** Organizaciones que ya han superado la etapa de scripts aislados y buscan profesionalizar su ciclo de vida de datos (DataOps), implementando CI/CD, pruebas unitarias y observabilidad avanzada.

Plan orientativo de implantación
Pasos necesarios y estimaciones
- **Fase 1: Evaluación y PoC (1-2 semanas):** Identificar un pipeline crítico pero contenido (ej. flujo dbt o ingesta de una API). Configuración del entorno local con dagster dev.
- **Fase 2: Configuración de Infraestructura (2 semanas):**
- En OSS: Despliegue en Kubernetes mediante Helm charts o Docker Compose.
- En Dagster+: Configuración de agentes híbridos para ejecutar código en la infraestructura propia del cliente manteniendo el plano de control en la nube.
- **Fase 3: Migración Incremental (Variable):** Utilización de herramientas como **Airlift** para observar DAGs existentes de Airflow desde la interfaz de Dagster sin migrar el código inmediatamente (estrategia Peer-Observe-Migrate).
- **Fase 4: Capacitación y estándares (2-3 semanas):** Definición de "Asset Factories" (patrones reutilizables) para que otros equipos puedan crear pipelines de forma estandarizada.

Necesidades de formación del equipo
- **Desarrollo de Software:** Formación en pruebas unitarias específicas para datos y uso de Resources y I/O Managers para separar la lógica de computación del almacenamiento.
- **Cambio de Paradigma:** Entrenamiento en el modelo declarativo (definir el estado deseado del activo) frente al imperativo tradicional (secuencia de pasos).

Perfiles necesarios
- **Data Engineers:** Perfil principal para la definición de activos y lógica de transformación.
- **Platform/DevOps Engineer:** Si se opta por la versión Open Source, para la gestión del clúster y escalado de workers.
- **Analytics Engineers:** Para integrar modelos de dbt dentro del grafo global de activos.

Retorno de la inversión (ROI)
- **Tiempos:** Casos documentados muestran reducciones en el tiempo de creación de pipelines de **1-2 días a solo 1 hora** mediante automatización y CLI.
- **Reducción de costes:** Ahorros de hasta el **50% en costes de Data Warehouse** al optimizar qué activos necesitan ser materializados realmente (evitando cómputo innecesario).
- **KPIs:**
- Reducción del tiempo medio de detección de errores (MTTD) gracias al linaje integrado.
- Disminución de intervenciones manuales en horas de guardia (on-call).
- Tasa de reuso de código mediante el catálogo de activos compartido.

Otros
- **Airlift:** Herramienta clave para empresas que vienen de Airflow; permite una migración progresiva "tarea por tarea" evitando el riesgo de grandes migraciones "Big Bang".
- **Interoperabilidad:** Integración de "primer nivel" con **dbt**, permitiendo visualizar el linaje a nivel de columna y ejecutar tests de dbt directamente desde la interfaz de Dagster.

PREGUNTAS FRECUENTES

¿Qué diferencia a Dagster de otros orquestadores tradicionales como Apache Airflow?

A diferencia de los modelos basados en tareas (tasks), Dagster utiliza un enfoque centrado en activos de datos (Software-Defined Assets). Esto permite definir el estado final deseado de los datos y sus dependencias, facilitando un linaje claro y permitiendo que el sistema comprenda qué tablas o modelos de ML se están produciendo, en lugar de simplemente ejecutar scripts en una secuencia ciega.

¿Es Dagster una tecnología de código abierto?

Sí, el núcleo de Dagster es open source y está distribuido bajo la licencia Apache 2.0. El código fuente completo, incluyendo la interfaz de usuario y las herramientas de ejecución, está disponible para su descarga y contribución en GitHub.

¿Qué costes implica el uso de Dagster en un entorno profesional?

Existen dos vías: la versión Open Source es totalmente gratuita pero requiere que la empresa gestione su propia infraestructura. La versión gestionada, Dagster+, ofrece un plan 'Solo' desde 10$ mensuales, un plan 'Starter' de 100$ y planes 'Enterprise' bajo presupuesto. El coste en la nube se calcula mediante créditos basados en la ejecución de operaciones y materialización de activos.

¿Cómo aborda Dagster el cumplimiento normativo y la seguridad de los datos?

En su modalidad Dagster+, la plataforma cumple con estándares de seguridad avanzados incluyendo la certificación SOC2 Type II y compatibilidad con HIPAA para datos sensibles. Implementa cifrado AES-256 en reposo y ofrece opciones de autenticación empresarial mediante SAML y sistemas de control de acceso basado en roles (RBAC).

¿Qué nivel técnico se requiere para implementar y mantener la herramienta?

El perfil de uso es técnico y requiere un dominio fluido de Python y SQL. Para la versión de código abierto, se necesitan competencias en ingeniería de software y DevOps para la configuración de contenedores (Docker) y orquestación de infraestructura (Kubernetes, AWS o GCP).

¿Es posible realizar pruebas unitarias sobre los pipelines de datos?

Sí, una de las ventajas competitivas de Dagster es su capacidad para facilitar el testeo. Al separar la lógica de negocio de la infraestructura mediante recursos y definiciones de activos, los ingenieros pueden ejecutar pruebas unitarias y de integración localmente, lo que reduce errores en producción.

¿Cómo se integra Dagster con herramientas de transformación como dbt?

Dagster ofrece una integración de primer nivel con dbt, permitiendo importar proyectos de dbt como activos dentro de su grafo. Esto proporciona una visibilidad completa del linaje de datos a nivel de columna, permitiendo rastrear el origen y destino de la información a través de ambos sistemas de forma unificada.

¿Para qué escenarios no se recomienda el uso de esta plataforma?

No es la herramienta adecuada para equipos que no utilicen Python como lenguaje base o que busquen soluciones estrictamente 'no-code'. Tampoco es eficiente para procesos extremadamente simples que pueden resolverse con tareas programadas básicas (cron jobs), dada la curva de aprendizaje de su modelo programático.

¿Cuenta con una interfaz para la observación y depuración de flujos de trabajo?

Sí, dispone de Dagster UI, una consola web profesional que permite visualizar el linaje de los activos, monitorizar ejecuciones en tiempo real, lanzar re-ejecuciones parciales y consultar un catálogo de datos integrado para entender el estado de salud de toda la plataforma de datos.

CONTRATOS Y CONDICIONES

Informe técnico descriptivo

Principales recomendaciones
- **Privacidad desde el diseño**: Al integrar Dagster con herramientas de terceros (Fivetran, Airbyte, Snowflake), asegure que el filtrado de datos personales se realice en el origen para evitar que información sensible sea procesada innecesariamente.
- **Configuración de Residencia**: Si utiliza la versión Cloud (Dagster+), es imperativo seleccionar explícitamente la región **eu-north-1** (Estocolmo) durante la configuración para garantizar que el plano de control y los metadatos permanezcan en territorio de la UE.
- **Minimización de Metadatos**: Evite incluir datos personales (nombres, emails, DNIs) en los nombres de los activos, logs o parámetros de configuración, ya que estos sí se sincronizan con la infraestructura del proveedor.
- **Gestión de Secretos**: Utilice siempre el sistema de gestión de secretos de Dagster o servicios externos (AWS Secrets Manager, HashiCorp Vault) para las credenciales de conexión; nunca las incluya directamente en el código Python de los pipelines.

Ley de Inteligencia Artificial (AI Act)
- **Clasificación de riesgo**: Como orquestador, Dagster no es un sistema de IA "per se", pero si se utiliza para entrenar modelos de "Alto Riesgo" (ej. RRHH, infraestructuras críticas), la empresa debe documentar el **linaje de datos** y la transparencia del entrenamiento que Dagster facilita.
- **Gobernanza de datos**: Las funcionalidades de "Asset-Centric" ayudan a cumplir con los requisitos de calidad de datos del AI Act, permitiendo auditar el origen y las transformaciones aplicadas a los sets de datos usados para el entrenamiento.

Privacidad y protección de datos
- **Responsabilidades**: La empresa usuaria actúa como **Responsable del Tratamiento**. Dagster Labs (Elementl, Inc.) actúa como **Encargado del Tratamiento** únicamente respecto a los metadatos operativos y de configuración.
- **Ubicación de los datos**:
- **Arquitectura Híbrida**: Sus datos de negocio nunca salen de su propia infraestructura (AWS, GCP, Azure o On-premise).
- **Plano de Control**: Existe una versión específica para la UE donde los metadatos se almacenan en centros de datos europeos (Región eu-north-1).
- **Transferencia internacional**: Al ser una empresa con sede en EE.UU. (California), el uso de Dagster+ implica una transferencia internacional de metadatos. Se recomienda verificar la adhesión del proveedor al EU-U.S. Data Privacy Framework o la firma de Cláusulas Contractuales Tipo (SCCs).
- **Derechos ARCO**: Al ser un orquestador que no almacena los datos finales, el ejercicio de derechos (acceso, supresión) debe realizarse sobre las bases de datos de origen o destino (Snowflake, BigQuery, etc.), no en Dagster.

Propiedad intelectual
- **Propiedad de datos**: La empresa usuaria mantiene la propiedad total sobre los datos procesados y el código Python desarrollado para los pipelines.
- **Licencia Open Source**: El núcleo de la herramienta usa la licencia **Apache 2.0**, permitiendo uso comercial, modificación y distribución sin costes de royalties, siempre que se mantengan los avisos de copyright originales.

Usos y prohibiciones
- **Usos prohibidos**: Está estrictamente prohibido el envío de **Datos Prohibidos** ("Prohibited Data") al plano de control de Dagster+, incluyendo: números de tarjetas de crédito, contraseñas, datos de salud (HIPAA fuera de contratos específicos) y categorías especiales del RGPD.
- **Usos admitidos**: Orquestación de flujos de trabajo de ingeniería de datos, entrenamiento de modelos ML, sincronización de bases de datos y monitorización de activos de datos.

Seguridad y certificaciones
- **Seguridad**: Cifrado AES-256 en reposo para metadatos y uso obligatorio de HTTPS para todas las comunicaciones.

- **Certificaciones**: Dispone de informe **SOC 2 Type II** y cumplimiento con **HIPAA** (bajo acuerdos específicos).
- **Control de Acceso**: Soporta RBAC (Control de acceso basado en roles) y autenticación vía SAML en planes empresariales.

Otros
- **Impacto Legal Legal (Medio/Bajo)**: Es bajo si se usa la versión Open Source (auto-alojada), ya que el control es total del cliente. Es medio si se usa Dagster+ debido a la exportación de metadatos de configuración a terceros.

Fuentes consultadas:
- Contratos y Términos de Servicio
- Política de Privacidad
- Seguridad y Cumplimiento
- Documentación Dagster+ UE
- Licencia Apache 2.0 (GitHub)