



Microsoft TRELIS.2

TRELIS.2 es un modelo generativo de IA de 4.000 millones de parámetros diseñado para profesionales del desarrollo de videojuegos, efectos visuales (VFX) y diseño industrial. Permite transformar una única imagen 2D en activos 3D de alta fidelidad con materiales PBR completos (metalicidad, rugosidad, opacidad) en segundos. Es ideal para departamentos de arte 3D que buscan acelerar el prototipado de assets complejos y estructuras internas mediante una arquitectura de vóxeles dispersos.

[Visitar Sitio Oficial](#) [Preguntar a ChatGPT](#) [Preguntar a Claude](#) [Preguntar a Grok](#)

Contenido del Dossier

- [Información de la Herramienta](#)
- [Consejos de Implantación](#)
- [Preguntas Frecuentes](#)
- [Contratos y Condiciones](#)

INFORMACIÓN DE LA HERRAMIENTA

Qué y para quién es

TRELLIS.2 es un modelo generativo de inteligencia artificial a gran escala (4.000 millones de parámetros) desarrollado por Microsoft, diseñado específicamente para la creación de activos 3D de alta fidelidad a partir de una única imagen 2D (Image-to-3D).

En el ámbito profesional, está dirigido a sectores como el desarrollo de videojuegos, efectos visuales (VFX), diseño industrial y comercio electrónico. Es una herramienta para departamentos de arte 3D y diseño de producto que buscan acelerar la fase de prototipado y creación de "assets" listos para entornos profesionales (PBR).

Principal ventaja profesional

La capacidad de generar modelos con topología compleja (superficies abiertas como hojas o ropa, y estructuras internas) y materiales PBR completos (metalicidad, rugosidad, opacidad) en segundos, eliminando la necesidad de procesos de optimización manual lentos que suelen requerir otras herramientas de IA.

Para quién no es

Profesionales que operen exclusivamente en entornos Windows o macOS (requiere Linux para ejecución local), o aquellos que no dispongan de hardware gráfico de alta gama (mínimo 24GB de VRAM). No es apto para empresas que busquen una solución "llave en mano" sin perfiles técnicos capaces de gestionar entornos de desarrollo Python/CUDA.

Funcionalidades clave

- **Generación en múltiples resoluciones:** Capacidad de crear activos desde 512³ hasta 1536³.
- **Estructura O-Voxel:** Una representación de vóxeles dispersos que permite manejar geometrías no estructuradas y superficies abiertas sin pérdida de detalle.
- **Materiales PBR nativos:** Generación automática de mapas de color base, rugosidad (roughness), metalicidad y opacidad.
- **Eficiencia extrema:** Generación de modelos en tiempos récord (desde 3 segundos en resolución 512³ hasta 1 minuto en 1536³ usando hardware H100).
- **Conversión bidireccional instantánea:** Transformación de mallas a vóxeles en <10s y viceversa en <100ms.

Precios

- **Versión gratuita:** Open Source bajo licencia MIT. El código, los pesos del modelo y la documentación están disponibles de forma gratuita para investigación y uso académico.
- **Hugging Face Demo:** Existe una versión gratuita limitada en la nube para pruebas rápidas.
- **Uso en producción (API externa):** Plataformas como Runware ofrecen acceso vía API con costes basados en el tiempo de procesamiento (ej.: aprox. 0.025\$ por generación a 1024px).

Perfil del usuario

- Empresas de videojuegos (Indie y AAA) para creación rápida de props y elementos de entorno.
- Estudios de arquitectura y diseño de interiores para visualización rápida de mobiliario a partir de bocetos.
- Departamentos de marketing que requieran modelos 3D rápidos para visores web o realidad aumentada.
- **Perfiles profesionales:** Artistas 3D, Desarrolladores de IA, Diseñadores de productos y Artistas técnicos.

Nivel técnico requerido

- **Uso:** Medio (requiere entender conceptos de 3D y materiales PBR).
- **Instalación/Configuración:** Muy alto (requiere experiencia en Linux, Docker, gestión de entornos Conda, compilación de kernels CUDA y manejo de PyTorch).
- **Necesidades de soporte:** Requiere intervención de perfiles de sistemas o ingenieros de Machine Learning para el despliegue local.
- **Conocimientos necesarios:** Python, CUDA Toolkit 12.4, gestión de modelos en Hugging Face.

Ejemplos de uso profesional

- Generación de objetos decorativos (props) para escenarios de videojuegos a partir de arte conceptual.
- Creación de activos para catálogos de e-commerce en formato GLB con materiales realistas.
- Prototipado rápido de calzado o prendas de ropa con texturas de cuero o tela realistas.

Uso y distribución

- **Versión web:** Demo oficial en Hugging Face Spaces.
- **Versión escritorio:** No disponible como aplicación nativa; requiere ejecución mediante scripts de Python o contenedores Docker.
- **CLI:** Interfaz de línea de comandos disponible para entrenamiento e inferencia masiva.

Open source

El proyecto es de código abierto. Incluye el código de inferencia, el código de entrenamiento y los pesos del modelo pre-entrenado (4B).

Integraciones

- **Facilidad de integración:** Media-Alta para desarrolladores (vía código Python).
- **API propia:** No dispone de una API comercial directa de Microsoft, pero es integrable en flujos de trabajo mediante la librería oficial.
- **Integraciones nativas:** Existe un wrapper para **ComfyUI**, lo que permite integrarlo en flujos de trabajo de Stable Diffusion para generar 3D directamente desde prompts de texto.
- **Formatos de exportación:** Exporta nativamente a GLB (con soporte de transparencia) y PLY.

Notas finales

Información legal, licencias y contratos

- **Licencia:** MIT para el código y modelo (permite uso, copia y modificación).
- **Disclaimer:** Microsoft especifica que los materiales están destinados principalmente a propósitos académicos y de investigación. Se debe tener precaución con la propiedad intelectual de las imágenes utilizadas como entrada.

Otros

- El requisito de hardware es estricto: se recomienda NVIDIA A100 o H100, aunque puede funcionar en GPUs de consumo como la RTX 3090/4090 (24GB VRAM) con optimizaciones de memoria específicas.

Para más información:

- [Sitio web oficial](#)
- [Github](#)
- [Precios \(vía API Runware\)](#)
- [Hugging Face Spaces](#)

CONSEJOS DE IMPLANTACIÓN

Aplicación profesional

- Sectores clave: Desarrollo de videojuegos (AAA e Indie), estudios de efectos visuales (VFX), diseño industrial, arquitectura y comercio electrónico avanzado.
- Presupuesto: Bajo en licenciamiento (Open Source), pero elevado en infraestructura de hardware o consumo de APIs externas.
- Puntos clave: Reducción drástica del tiempo de modelado manual de "props" y activos secundarios; generación de modelos con topología compleja y materiales PBR listos para renderizado profesional.

Madurez digital requerida

- Usuarios y equipo: Artistas 3D familiarizados con flujos de trabajo PBR y diseñadores técnicos con capacidad para integrar modelos generados en motores de renderizado (Unreal Engine, Unity, Blender).
- Empresa y departamentos: Departamentos de IT/Sistemas con experiencia en entornos Linux y gestión de contenedores; departamentos de arte con disposición a adoptar flujos de trabajo híbridos (IA + retoque manual).

Plan orientativo de implantación

Pasos necesarios y estimaciones

- Tiempos estimados de despliegue: De 1 a 3 semanas para una implementación local estable.
- Evaluación inicial: Auditoría de hardware disponible (GPU NVIDIA con >24GB VRAM) y definición de los formatos de salida requeridos para el pipeline existente (GLB/PLY).
- Prueba de concepto (PoC): Uso de la demo en Hugging Face o implementación mediante ComfyUI para validar la calidad de los activos generados con el estilo artístico de la empresa.
- Configuración y personalización: Despliegue en servidores locales o instancias cloud (A100/H100) utilizando Docker y configuración de la integración con herramientas de diseño.
- Formación y adaptación: Capacitación del equipo de arte en la selección de imágenes de entrada óptimas y en la post-producción de los modelos generados.

Necesidades de formación del equipo

- Capacitación en el manejo de modelos generativos 3D y comprensión de la arquitectura O-Voxel.
- Formación técnica para la limpieza y optimización de mallas automáticas en software como Blender o ZBrush.

Perfiles necesarios

- Perfiles técnicos: Ingeniero de Machine Learning o Ingeniero de Software con experiencia en Python, CUDA y PyTorch.
- Personal externo: Consultores de implementación de IA o proveedores de infraestructura GPU en la nube.
- Otros: Un "Technical Artist" para asegurar que los materiales PBR generados cumplen con los estándares de calidad del proyecto.

Retorno de la inversión

- Tiempos: Reducción del tiempo de creación de activos base de horas/días a menos de un minuto.
- KPIs: Tiempo medio de creación por asset, coste por generación frente al coste de modelado manual, y tasa de activos generados por IA que superan el control de calidad sin necesidad de correcciones mayores.

Otros

- Requisitos técnicos críticos: El sistema requiere estrictamente Linux y controladores CUDA 12.4; no existe soporte nativo para Windows fuera de WSL2 (Windows Subsystem for Linux), lo que puede complicar la integración en estaciones de trabajo estándar de Windows.
- Optimización de recursos: Para empresas sin hardware dedicado, el uso de APIs externas como Runware permite escalar la producción sin inversión CAPEX en servidores.

PREGUNTAS FRECUENTES

¿Qué es TRELIS.2 y en qué se diferencia de otros generadores 3D?

Es un modelo generativo de 4.000 millones de parámetros desarrollado por Microsoft que transforma imágenes 2D en activos 3D de alta fidelidad. A diferencia de otros modelos, utiliza una arquitectura basada en O-Voxels (vóxeles dispersos) que permite representar geometrías complejas, como superficies abiertas, estructuras internas delgadas y materiales PBR detallados, evitando las limitaciones de las mallas tradicionales en la generación automática.

¿Cuáles son los requisitos técnicos de hardware para ejecutarlo localmente?

El sistema demanda un hardware gráfico de alto rendimiento debido a su tamaño de parámetros y la complejidad de las operaciones CUDA. Se requiere una GPU con un mínimo de 24 GB de VRAM (como la NVIDIA RTX 3090 o 4090), aunque para un rendimiento óptimo en entornos profesionales se recomienda el uso de hardware de grado empresarial como NVIDIA A100 o H100.

¿Es una herramienta de código abierto?

Sí, TRELIS.2 se distribuye bajo la licencia MIT. Esto implica que tanto el código de inferencia como los pesos del modelo y la documentación están disponibles de forma gratuita para su uso, modificación y distribución, orientándose principalmente a la investigación y al desarrollo académico y técnico.

¿Cómo se integra en flujos de trabajo de diseño profesional ya existentes?

La integración se realiza principalmente mediante scripts de Python o a través de integraciones comunitarias como el nodo para ComfyUI, lo que permite conectarlo con flujos de trabajo de Stable Diffusion. Los activos finales se exportan en formatos estándar como GLB (con soporte para transparencia y materiales PBR) y PLY, facilitando su importación en motores de juegos como Unreal Engine o Unity, y software de modelado como Blender.

¿Qué tipo de materiales y texturas genera automáticamente?

El modelo genera mapas de flujo de trabajo PBR (Physically Based Rendering) nativos. Esto incluye el mapa de color base (albedo), rugosidad (roughness), metalicidad (metallicity) y mapas de opacidad, lo que permite que los objetos reaccionen de forma realista a la iluminación en entornos de renderizado profesional sin necesidad de texturizado manual adicional.

¿Es posible utilizar TRELIS.2 sin disponer de un servidor propio o hardware de alta gama?

Sí, existen dos alternativas principales. Para pruebas rápidas y experimentación, está disponible una demo gratuita en Hugging Face Spaces. Para implementaciones en producción que requieran escalabilidad, plataformas de terceros como Runware ofrecen acceso vía API, permitiendo la generación de modelos mediante un modelo de pago por uso basado en el tiempo de procesamiento.

¿Qué nivel de conocimientos técnicos se requiere para la instalación?

El nivel técnico requerido para la configuración inicial es muy alto. El profesional debe tener experiencia avanzada en entornos Linux, gestión de contenedores Docker, configuración de entornos Conda y compilación de kernels específicos de NVIDIA CUDA 12.4, además de manejo de la biblioteca PyTorch.

¿Cumple con normativas de privacidad y propiedad intelectual?

A ser una herramienta que se ejecuta localmente o mediante infraestructura privada, el control de los datos de entrada reside en el usuario. No obstante, Microsoft advierte que el uso de imágenes protegidas por derechos de autor como entrada para la generación de modelos 3D es responsabilidad del usuario, instando a verificar la propiedad intelectual antes de procesar activos en el modelo.

¿Cuánto tiempo tarda en generar un modelo 3D completo?

La eficiencia varía según la resolución deseada y el hardware utilizado. En una GPU H100, la generación en resolución básica (512³) tarda aproximadamente 3 segundos, mientras que la generación de activos de alta resolución (1536³) puede completarse en un tiempo estimado de un minuto.

CONTRATOS Y CONDICIONES

Informe técnico descriptivo

Principales recomendaciones

- Evaluar el hardware disponible: requiere GPUs de gama alta (tipo NVIDIA H100, A100 o RTX 3090/4090 con 24GB de VRAM) y entorno Linux.
- Verificar las licencias de las librerías dependientes (nvdiffrast y nvdiffrast), ya que aunque el modelo central es abierto, estas extensiones de terceros pueden tener restricciones específicas para uso comercial.
- Evitar el uso de imágenes con derechos de autor de terceros como entrada (input) para generar modelos 3D, ya que el resultado podría considerarse una obra derivada no autorizada.
- Implementar un proceso de revisión humana para filtrar posibles sesgos o contenidos inapropiados generados por el modelo, dado que ha sido entrenado con datos públicos de internet.

Ley de Inteligencia Artificial (AI Act)

- Clasificación de riesgo: Bajo (IA de propósito general). No se encuadra en sistemas de alto riesgo al ser una herramienta de creación de activos 3D, siempre que no se use para fines biométricos o de vigilancia.
- Transparencia: Al ser un modelo generativo, la empresa debe informar que el contenido 3D ha sido generado por IA, especialmente si se utiliza en interfaces con usuarios finales.
- Documentación técnica: El cumplimiento se facilita al ser un modelo con documentación técnica pública y pesos disponibles (4B), lo que permite la trazabilidad exigida por la normativa de la UE para modelos de propósito general.

Privacidad y protección de datos

- Ubicación de los datos: En ejecución local (on-premise), los datos permanecen bajo control total de la empresa española. Si se usa vía API de terceros o demos (Hugging Face), los datos se procesan en servidores externos.
- Responsabilidades: La empresa española actúa como Responsable del Tratamiento si utiliza imágenes de personas físicas para generar modelos 3D, debiendo contar con base legal (consentimiento o interés legítimo).
- Derechos ARCO: Se deben facilitar mecanismos para que los interesados puedan ejercer sus derechos si su imagen personal ha sido utilizada en el proceso de entrenamiento o generación.

Propiedad intelectual

- Propiedad de datos: El usuario mantiene la propiedad de las imágenes de entrada suministradas al modelo.
- Propiedad del resultado: Según la legislación española y de la UE, las obras creadas exclusivamente por IA sin intervención humana significativa podrían carecer de protección por derecho de autor. Se recomienda la edición posterior por artistas 3D para consolidar la autoría.
- Licencia del modelo: Distribuido bajo MIT License, lo que permite copia, modificación y uso comercial, siempre que se incluya el aviso de copyright original de Microsoft.

Usos y prohibiciones

- Usos admitidos: Generación de activos para videojuegos, prototipado industrial, comercio electrónico (formatos GLB con materiales PBR) y fines de investigación académica.
- Usos prohibidos: Generación de contenido dañino, ilegal o que infrinja los derechos de propiedad intelectual de terceros según las políticas de uso responsable de Microsoft.

Seguridad y certificaciones

- Seguridad: Al ser software de código abierto ejecutable bajo Python y CUDA, se recomienda realizar auditorías de seguridad sobre las dependencias externas para evitar vulnerabilidades en la infraestructura local.
- Certificaciones: No cuenta con certificaciones tipo ISO o SOC2 de forma nativa al ser un proyecto de investigación, por lo que la seguridad depende del entorno de despliegue de la empresa.

Otros

- El modelo se presenta como un proyecto de investigación ("Research Project"). Microsoft incluye un descargo de responsabilidad (Disclaimer) indicando que los materiales no están destinados originalmente para explotación comercial directa, a pesar de la licencia MIT permisiva.

Fuentes consultada:

- [Contratos y condiciones](#)
- [Licencias \(GitHub\)](#)
- [Documentación técnica y código](#)

Para más información y herramientas:

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

[Inicio](#) [Todas las herramientas](#) [Categorías](#)

Este documento ofrece recomendaciones generadas mediante análisis humano y sistemas de IA automatizados. La información tiene carácter meramente informativo y no constituye asesoramiento legal, profesional ni garantía de resultados. Las marcas, logotipos y nombres comerciales pertenecen a sus respectivos propietarios y se utilizan únicamente con fines identificativos.