

Modelos listos para explorar

Hugging Face reúne modelos, datos, demos y herramientas en un mismo ecosistema.



Hugging Face

El Hub permite descubrir recursos; su calidad y condiciones se revisan uno a uno.

El repositorio sirve como punto de encuentro para publicar, documentar, versionar y reutilizar artefactos de aprendizaje automático. Descargar un modelo no garantiza que sea adecuado para el dominio, la licencia o los recursos disponibles.

Cuatro piezas conectadas

Cada pieza resuelve una parte distinta del trabajo con modelos.



Model Hub y Dataset Hub almacenan recursos versionados. Transformers ofrece clases y pipelines para arquitecturas compatibles. Spaces permite publicar interfaces, habitualmente con Gradio o Streamlit.

De la búsqueda a tu aplicación

Primero exploras, después pruebas y solo entonces integras.

1. Buscar

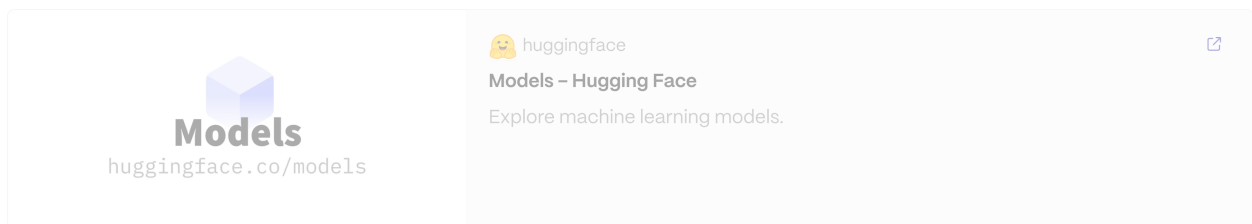
Filtra por tarea, licencia y librería.

2. Probar

Lee la ficha y usa una demo o notebook.

3. Integrar

Fija versión, mide y documenta.



El catálogo se filtra por tarea y metadatos; la decisión se toma con evidencia propia.

Empieza por la tarea

El filtro reduce candidatos; la documentación decide cuáles merece la pena probar.

Tarea y tipo de salida que necesitas.

Dominio de las imágenes de uso real.

Entorno de hardware, latencia y memoria.

Buscas un clasificador para imágenes médicas. ¿Qué debes hacer primero?

- ☐ **A** Elegir el modelo con más descargas.
 - ☐ **B** Filtrar por tarea y revisar la model card con el dominio y licencia en mente.
 - ☐ **C** Descargar todos los modelos y comparar solo su tamaño.
-

Respuesta B. Exacto: la tarea reduce la búsqueda y la documentación permite juzgar la adecuación.

Las descargas, los likes y la fecha de actualización ayudan a priorizar la inspección, pero no sustituyen una evaluación en el dominio objetivo.

La ficha cuenta la historia

Una model card útil explica origen, uso, resultados y límites.

Uso Tarea, entradas, código y casos previstos.	Evidencia Datos, métricas y condiciones de evaluación.	Riesgos Sesgos, fallos conocidos y licencia.
--	--	--

Conviene registrar el identificador y la revisión exacta del repositorio. Dos versiones con el mismo nombre pueden cambiar archivos, configuración o resultados.

Descargar no significa poder usar

Código, pesos, datos y aplicación final pueden tener condiciones distintas.

Permisos	Obligaciones	Restricciones
Uso comercial, modificación y redistribución.	Atribución, avisos, publicación o uso responsable.	Ámbitos excluidos, investigación o cláusulas específicas.

Apache-2.0, MIT y BSD suelen ser permisivas, pero hay que leer el texto concreto. OpenRAIL y licencias personalizadas pueden añadir restricciones de uso. Las variantes Creative Commons y las licencias “research only” no son intercambiables. Para una decisión jurídica o comercial, revisar los términos del recurso y del caso de uso.

Revisa el artefacto completo

La etiqueta del repositorio es el inicio de la revisión, no su conclusión.

Identifica la versión y todos sus archivos.

Comprueba términos de código, pesos y datos.

Registra la decisión y las obligaciones.

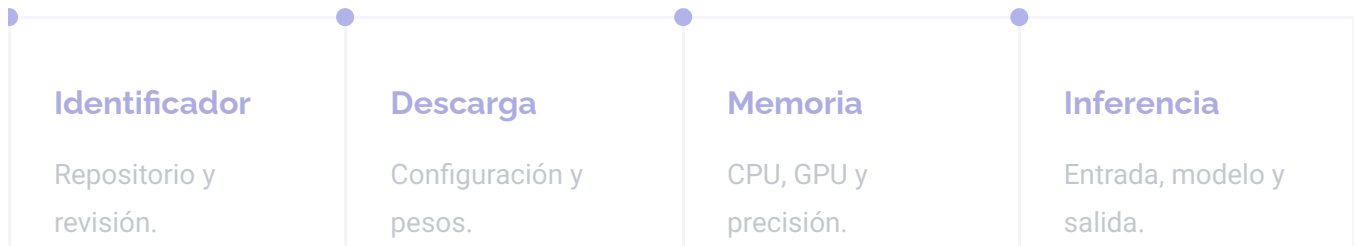
Un repositorio tiene código permisivo, pero los pesos incluyen otra licencia. ¿Qué debes revisar?

-
- ☐ **A** Solo la licencia del código.
-
- ☐ **B** Las condiciones de código, pesos, datos y uso de la versión concreta.
-
- ☐ **C** Nada si el modelo se puede descargar sin pagar.
-

Respuesta B. Exacto: cada recurso puede añadir términos distintos.

Del Hub a la memoria

La primera ejecución descarga archivos; las siguientes pueden reutilizar la caché.



La caché local evita repetir transferencias, pero no reduce la memoria necesaria para ejecutar el modelo. En entornos reproducibles conviene fijar una revisión y controlar la ubicación de caché.

Cada archivo cumple una función

La configuración describe la arquitectura; los pesos contienen lo aprendido.

config.json Arquitectura, tamaños y etiquetas.	model.safetensors Parámetros, a veces repartidos en fragmentos.	Preprocesador Resolución, normalización o tokenización.
--	---	---

Safetensors evita ejecutar código de deserialización basado en pickle y permite leer los pesos de forma eficiente. Aun así, se debe revisar el repositorio y cualquier código personalizado solicitado por el modelo.

Una API, muchas arquitecturas

`from_pretrained` reconstruye el procesador y el modelo desde su repositorio.

```
from transformers import AutoImageProcessor, AutoModelForImageClassification

model_id = "google/vit-base-patch16-224"
processor = AutoImageProcessor.from_pretrained(model_id)
model = AutoModelForImageClassification.from_pretrained(model_id)
```

Las clases Auto eligen una implementación compatible con la configuración. Para reproducibilidad se puede indicar una revisión concreta. El dispositivo y el tipo numérico se eligen según memoria, latencia y soporte del modelo.

Una prueba en pocas líneas

Un pipeline reúne preprocesado, inferencia y posprocesado para una tarea común.

```
from transformers import pipeline

detector = pipeline(
    "object-detection",
    model="PekingU/rtdetr_r50vd"
)
results = detector("imagen.jpg")
```

¿Qué simplifica un pipeline de Transformers?

- ☐ A La preparación de entrada y la ejecución de una tarea común.
- ☐ B La necesidad de conocer la licencia y la model card.
- ☐ C El entrenamiento de cualquier modelo sin datos.

Respuesta A. Exacto: el pipeline agrupa pasos frecuentes para una primera prueba.

El pipeline es ideal para explorar. En producción, las clases específicas permiten controlar mejor el batching, el dispositivo, la precisión, el preprocesado y el formato de salida.

Busca y prueba un modelo

Compara la documentación con lo que observas al ejecutar el cuaderno.

1. Guarda una copia en Drive.
2. Ejecuta el modelo propuesto.
3. Elige otro y justifica la decisión.

Abrir en Colab ↗

Para que la comparación sea reproducible, anota el identificador y la revisión del modelo, la tarea, la licencia, el preprocesador, el dispositivo y el consumo aproximado de memoria. Las descargas o likes sirven para descubrir candidatos, no para decidir cuál funciona mejor en tu dominio.