

# JupyterLab: Análisis y experimentación con Python

---

## Introducción

JupyterLab es un entorno web interactivo para trabajar con notebooks, código, texto técnico, tablas, gráficas y resultados en una misma interfaz. Dentro del stack se usa como entorno auxiliar de desarrollo y análisis, pensado para pruebas rápidas, procesamiento de datos, revisión de logs y experimentación con Python sin instalar dependencias directamente sobre el sistema anfitrión.

El despliegue se basa en la imagen oficial `quay.io/jupyter/datascience-notebook`, perteneciente a Jupyter Docker Stacks. Esta familia de imágenes proporciona entornos Docker ya preparados para ejecutar Jupyter Server con JupyterLab y herramientas de computación interactiva.

---

## Enfoque general

El servicio se despliega como un contenedor independiente llamado `JupyterNotebookDS`.

La configuración mantiene una separación clara entre el entorno de ejecución y los datos de trabajo:

- El contenedor aporta JupyterLab, Python y librerías de ciencia de datos;
- El directorio `./work` conserva notebooks, scripts, datasets y resultados;
- El token de acceso se define mediante `.env`;
- La interfaz web se expone desde el puerto interno `8888` hacia el puerto externo `15930`;
- El servicio queda aislado en la red Docker `Jupyter_NET`.

Este enfoque permite usar Jupyter como entorno reproducible y desechable. El contenedor puede actualizarse o recrearse sin perder el contenido de trabajo, siempre que el volumen persistente se conserve.

---

## Desarrollo

# Qué se hizo y por qué

Se desplegó JupyterLab usando la imagen `quay.io/jupyter/datascience-notebook:latest`.

La elección de esta imagen evita construir una imagen propia para un uso generalista. Incluye un entorno preparado para análisis de datos, notebooks y pruebas con Python, además de librerías habituales como pandas, NumPy, SciPy, Matplotlib o scikit-learn, según la versión publicada de la imagen.

El directorio de trabajo se monta en `/home/jovyan/work`, que es la ruta habitual usada por las imágenes oficiales de Jupyter. Además, el arranque fuerza `--ServerApp.root_dir=/home/jovyan/work`, de forma que la interfaz queda limitada al directorio persistente definido para el servicio.

El acceso se protege mediante token configurado desde `.env`. Esto evita dejar el token escrito directamente en el `docker-compose.yml` y permite sustituirlo sin modificar la definición principal del servicio.

## Configuración utilizada

La configuración del despliegue está versionada en Gitea:

[Configuración de JupyterLab en Gitea](#)

El despliegue utiliza dos archivos principales:

- `.env`, con la variable `JUPYTER_TOKEN`.
- `docker-compose.yml`, con la definición del contenedor, volumen, puerto, red y comando de arranque.

---

## Validación

La validación básica del servicio consiste en comprobar que el contenedor arranca correctamente, que Jupyter escucha en el puerto esperado y que el acceso requiere el token configurado.

El contenedor debe aparecer activo con el nombre `JupyterNotebookDS`.

La interfaz debe quedar disponible en el puerto externo `15930`, redirigido al puerto interno `8888` del contenedor.

El directorio `./work` debe reflejar los cambios realizados desde JupyterLab. La creación de notebooks, scripts o archivos desde la interfaz web debe persistir en el host tras recrear el contenedor.

También debe comprobarse que el directorio inicial mostrado por JupyterLab corresponde a `/home/jovyan/work`, evitando navegar fuera del espacio de trabajo previsto.

---

## Decisiones importantes

El despliegue usa `latest`, lo que simplifica actualizaciones pero puede introducir cambios no controlados en librerías, versiones de Python o comportamiento de JupyterLab. Para un entorno más estable conviene fijar una etiqueta concreta de la imagen cuando el uso pase de experimental a dependiente de trabajos recurrentes.

El token se externaliza mediante `.env`, pero sigue siendo una credencial sensible. No debe publicarse con un valor real en repositorios accesibles.

El servicio no se plantea como entorno de producción ni como backend permanente de aplicaciones. Su función dentro del stack es servir como laboratorio técnico: análisis puntual, pruebas de código, revisión de datos, generación de gráficas y documentación ejecutable.

---

## Resumen breve

JupyterLab queda desplegado en Docker como entorno web de análisis y experimentación basado en `quay.io/jupyter/datascience-notebook`.

El servicio conserva el trabajo en `./work`, expone la interfaz en el puerto `15930`, usa token desde `.env` y queda aislado en la red `Jupyter_NET`.

La configuración prioriza simplicidad, persistencia del espacio de trabajo y separación entre sistema anfitrión y entorno Python.

---

## Referencias

[Documentación oficial de Jupyter Docker Stacks](#)

[Selección de imágenes en Jupyter Docker Stacks](#)

[Ejecución de contenedores Jupyter Docker Stacks](#)

---

Revision #2

Created 2026-06-06 12:13:05 UTC by Juan Francisco

Updated 2026-06-06 12:30:42 UTC by Juan Francisco