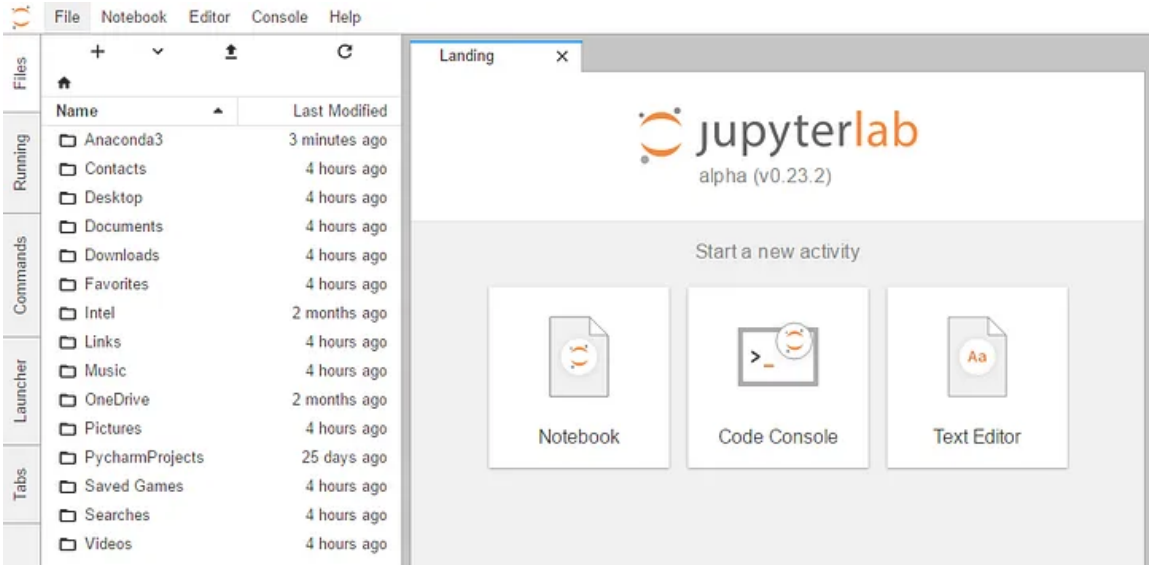
JupyterLab: La evolución de Jupyter Notebook

JupyterLab es la interfaz de usuario nueva generación para el proyecto Jupyter. Básicamente es un IDE con todas las funciones que tiene todo lo que siempre se quiso tener en los notebooks de Jupyter que le permite trabajar con documentos y actividades tal y como en Notebook, además de editores de texto, terminales y componentes personalizados de manera flexible, integrada y extensible.



Las principales características de JupyterLab son:

• *Arrastrar y soltar*:

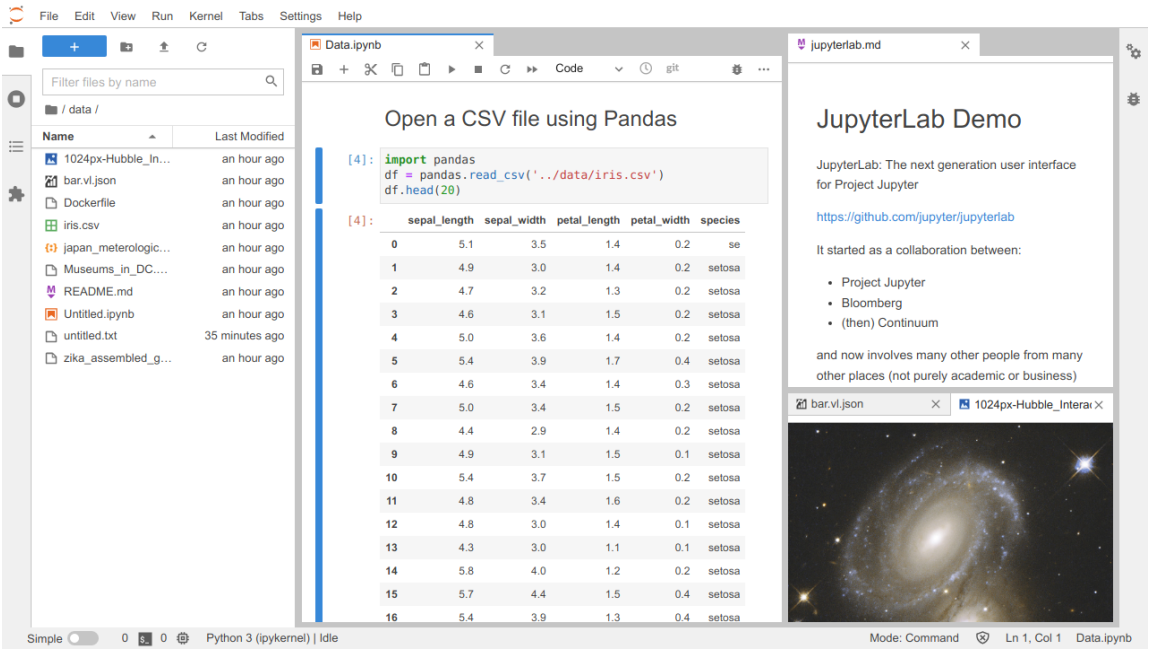
La capacidad de reordenar celdas sin cortar y pegar es poderosa. También se siente más natural arrastrar y soltar dado que el código está organizado en celdas en notebooks.

• *Múltiples notebooks y kernels*:

La ejecución de varios notebooks al mismo tiempo ya existe con los Juypyter Notebook. Sin embargo, estos notebooks tenían que abrirse en varias ventanas del navegador. En JupyterLab, puede tener varios notebooks abiertos al mismo tiempo y en la misma ventana del navegador. Además, puede organizar sus Notebook como desee, lo que le brinda más flexibilidad. Otra buena característica es que es posible hacer que cada notebook se ejecute en su propio kernel, esto es poderoso cuando se ejecutan varios notebooks al mismo tiempo haciendo cosas diferentes.

• *Editor de markdown en tiempo real*

Con esta nueva función, puedo editar y ver en tiempo real la actualización de mis archivos markdown en JupyterLab. Esto acelera el proceso de edición y agiliza el trabajo.

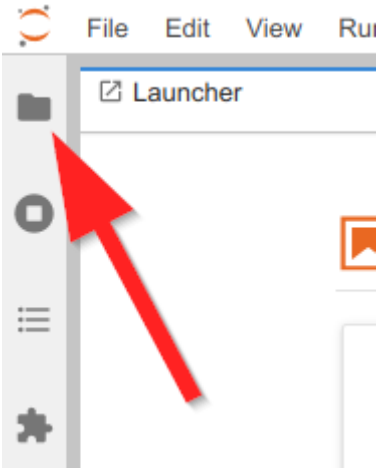


• *Múltiples ventanas*

Con varias ventanas abiertas al mismo tiempo, puedo tener varios notebooks en las que estoy trabajando y luego usar una terminal dentro de JupyterLab

• *Explorador de archivos completo*

El explorador de archivos y el menú Archivo le permiten trabajar con archivos y directorios en su sistema. Esto incluye abrir, crear, eliminar, renombrar, descargar, copiar y compartir archivos y directorios.



• *Gestionar distintos kernel y terminales*

• *Búsqueda de comandos*

- *Visor rápido de archivos CSV*
- ...

Ejecutar JupyterLab de forma remota con Slurm

Al igual que Jupyter Notebook, JupyterLab se puede iniciar localmente y acceder a los archivos locales, o se pueden iniciar en una máquina remota.

De forma predeterminada, **Labs no es seguro y potencialmente expone los archivos locales de los usuarios a usuarios no deseados**. Recomendamos leer la sección *Ejecutar Jupyter Notebook con Slurm* donde puede encontrar varios consejos para usar Jupyter Notebook y que son de aplicación a JupyterLab.

Recuerda:

No ejecute Jupyter en los nodos de login

La razón

Internet no está disponible en los nodos de cómputo

¿Qué solución hay?

No se puede acceder directamente a los nodos de cómputo

¿Por qué?

Esta implementación no es el servidor multiusuario que andas buscando

¿Qué significa esto?

Puede utilizar JupiterLab de dos maneras diferentes:

- Crear un entorno conda o pip
- Uso como módulo

Crear un entorno Conda para instalar JupyterLab

Crear un entorno Conda

Aquí tenemos varias recomendaciones para trabajar con entornos de conda que recomendamos leer. Para simplificar:

```
conda activate jupyter-labenv
conda install jupyterlab matplotlib ipykernel ipywidgets  ipympl --channel conda-forge -y
```

Usar JupyterLab como módulo

Para buscar el módulo escribir:

```
module spider jupyter
```

```
----------------------------------------------------------------------------
  JupyterLab: JupyterLab/3.1.6
----------------------------------------------------------------------------
    Description:
      JupyterLab is the next-generation user interface for Project Jupyter offering all the familiar
building blocks of the classic Jupyter Notebook (notebook, terminal, text editor, file browser, rich
outputs, etc.) in a flexible and powerful user interface.
      JupyterLab will eventually replace the classic Jupyter Notebook.


    You will need to load all module(s) on any one of the lines below before the "JupyterLab/3.1.6"
module is available to load.

    GCCcore/11.2.0
...
```

Cómo iniciar JupyterLab

Depende de cómo quieras iniciarlo. Si prefieres usarlo como módulo sólo tienes que cambiar la línea que activa el entorno por la que carga el módulo.

```
conda activate jupyter-labenv
```

with

```
module load GCCcore/11.2.0 JupyterLab/3.1.6
```

Ejecutar JupyterLab en un nodo de cómputo mediante *salloc*

```
salloc --nodes=1 --partition batch --ntasks=1 --mem=20G --time=12:00:00
# or salloc -N 1 -p batch
# or srun -p batch --pty bash

conda activate jupyter-labenv
jupyter lab --no-browser --port=8890 --ip=127.0.0.1
```

Por defecto JupyterLab no es seguro y expone sus archivos locales a usuarios no deseados.

Ejecutar Labs de la forma habitual utiliza conexiones HTTP inseguras. En esta guía, presentamos una guía para ejecutarlo de forma más segura. Más info aquí

Tip

Tenga en cuenta que el valor predeterminado es que JupyterLab abra automáticamente un navegador, pero no podemos hacerlo en un servidor remoto, por lo que omitimos esa función con el parámetro *--no-browser*).

Crear túnel SSH

Aquí hablamos sobre las razones por las que es necesario crear el túnel ssh y cómo hacerlo.

Inicia una segunda terminal, conecta a uno de los nodos de login y ejecuta crea un tunel entre el nodo login y el de ejecución.

```
ssh -N -L localhost:8889:localhost:8889 yourUser@nodeXXXX-Y.hpc.iter.es
```

En tu máquina local, establece el tunel contra el nodo de login como a continuación:

```
ssh -N -L localhost:8890:localhost:8890 yourUser@loginX.hpc.iter.es
```

!!! tip "Puedes usar los DNS del nodo o la IP

```
Change _nodeXXXX-X.hpc.iter.es_ or loginX.hpc.iter.es by node ip.
```

Introduce la dirección que te da Lab en tu navegador favorito

http://127.0.0.1:8890/lab?token=4c7eddd5770e27195808f4615d8e6f3de48b45c57169b69e

Recomendamos securizar Jupyter Labs

Ejecutar JupyterLab en un nodo de cómputo mediante *sbatch*

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --mem=20G
#SBATCH --partition=batch
#SBATCH --constrains=sandy      # sandy, ilk (icelake)... arquitecture
#SBATCH --time=01:00:00
#SBATCH --job-name=jupyter-lab
#SBATCH --output=jupyter-lab-%j.out
#SBATCH --error=jupyter-lab-%j.err
################################################################
# Enable modules profile

# Enable conda on bash console
eval "$(conda shell.bash hook)"

# load modules or conda environments here
module load  Miniconda3/4.9.2
conda activate jupyter-labenv

# get tunneling info
XDG_RUNTIME_DIR=""
node=$(hostname -s)
user=$(whoami)
cluster="teide-hpc"
port=8890

# print tunneling instructions jupyter-log
echo -e "
Command to create ssh tunnel from login node to compute node:
ssh -N -L localhost:${port}:localhost:${port} ${user}@nodeXXXX-Y.hpc.iter.es

Command to create ssh tunnel from your local machine to any login node:
ssh -N -L localhost:${port}:localhost:${port} ${user}@login1(2).hpc.iter.es

Use a Browser on your local machine to go to:
localhost:${port}  (prefix w/ https:// if using password)
"

# Run Jupyter
jupyter lab --no-browser --port=${port} --ip=127.0.0.1
```

Para ejecutarlo

```
sbatch jupyter-lab.sbatch
```

Para acceder a JupyterLab navegar a http://localhost:8890/

Notas importantes

Establecer una contraseña segura en JupyterLab

De forma predeterminada, Lab inicia el servidor con la autenticación de token habilitada y este token se registra en el terminal, de modo que puede copiar y pegar la URL en su navegador. **Este token se puede usar sólo una vez** y se usa para configurar una cookie para su navegador una vez que se conecta. Después de que su navegador haya realizado su primera solicitud con este token único, el token se descarta y se establece una cookie en su navegador.

Como alternativa a la autenticación por token puede establecer una contraseña para su servidor. En posteriores inicios se le pedirá una contraseña y esta se almacenará cifrada en su archivo de configuración *jupyter_server_config.json* que generalmente se encuentra en **~/.jupyter**. Puede configurar su contraseña escribiendo:

```
jupyter server password
```

```
Enter password:
Verify password:
[JupyterPasswordApp] Wrote hashed password to /home/vdominguez/.jupyter/
jupyter_server_config.json
```

You can access now with password instead of token autentication on http://localhost:8890/lab

Más consejos de seguridad aquí

https://jupyter-server.readthedocs.io

Cómo cambiar el directorio de inicio de Jupyter

Si los archivos de su notebook no están en el directorio donde está ejecutando Jupyter, puede pasar la ruta del directorio de trabajo como argumento al iniciar JupyterLab de la siguiente forma:

```
jupyter lab --notebook-dir=/home/yourUser/data/notebooks/ --preferred-dir /home/yourUser/data/
myapp
```

Hay que tener en cuenta que la sesión de Jupyter siempre reside en un workspace. El workspace por defecto es la URL */lab* :

`bash http(s)://<server:port>/<lab-location>/lab`

Asegúrese de cerrar JupyterLab cuando haya terminado

Para hacer esto, si está utilizando el modo *salloc*, puede volver a iniciar sesión en la sesión de *screen* que inició anteriormente donde se ejecuta el cuaderno jupyter y use *ctrl-C* para cerrar el servidor de Jupyter y *exit* para cerrar la sesión con el nodo.

Si está utilizando el modo *sbatch*, puede usar el comando *scancel* .

JupyterLab Extensions Manager

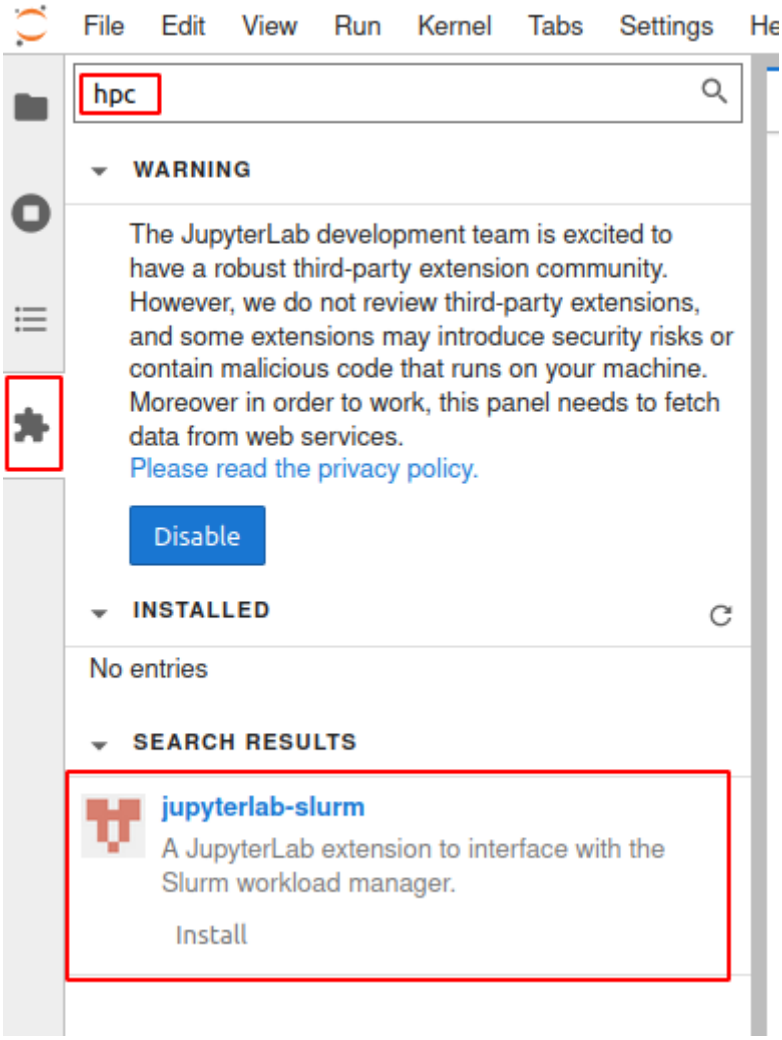
El gestor de extensiones de JupyterLab es simplemente un complemento plug-and-play que hace posibles más cosas que puede que necesite.

Técnicamente, la extensión JupyterLab es un paquete de JavaScript que puede agregar todo tipo de características interactivas a la interfaz de JupyterLab.

Hay un montón de extensiones de JupyterLab que tal vez quieras usar. Entre las más conocidas podemos destacar:

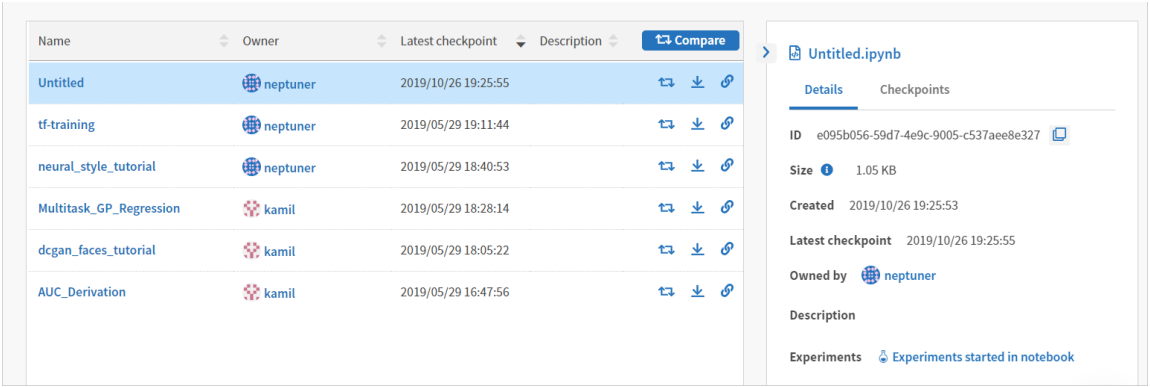
Jupyterlab-slurm

Una extensión de JupyterLab para interactuar con el administrador de carga de trabajo de Slurm.



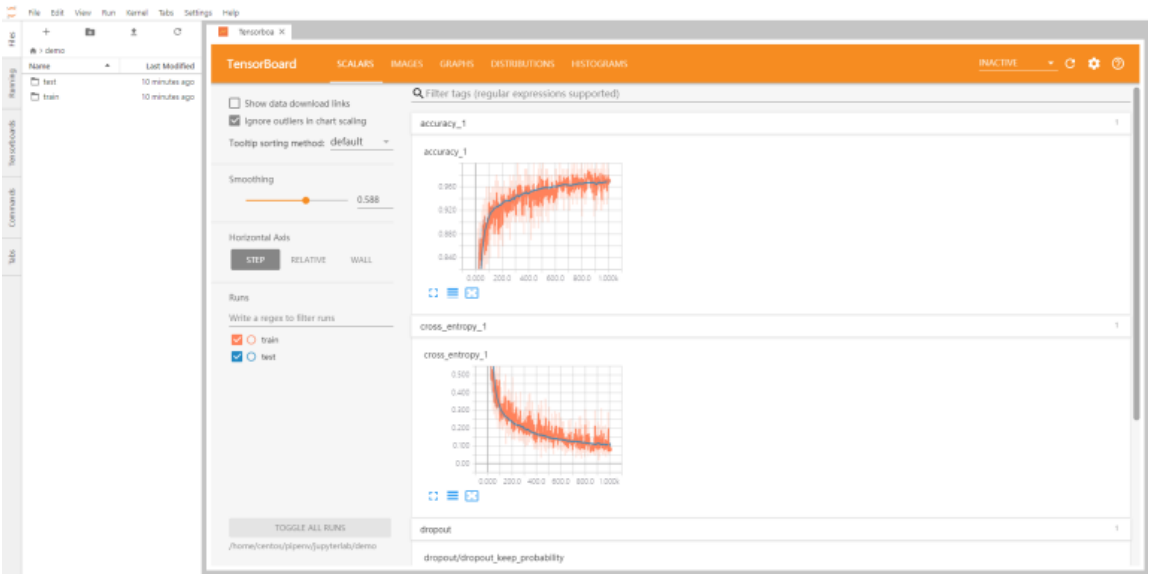
Neptune-notebooks

Neptune es una herramienta para el seguimiento de experimentos, el registro de modelos, el control de versiones de datos y la supervisión de modelos en vivo.



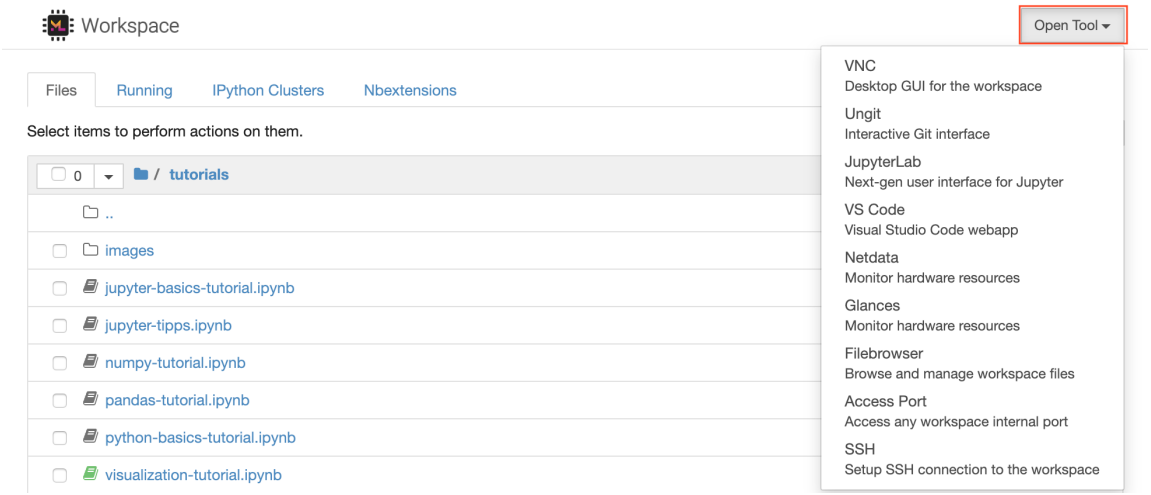
JupyterLab TensorBoard

JupyterLab TensorBoard es una extensión de interfaz para tensorboard en jupyterlab. Ayuda a colaborar entre jupyter notebook y tensorboard (una herramienta de visualización para tensorflow) al proporcionar una interfaz gráfica de usuario para iniciar, administrar y detener tensorboard en la interfaz de jupyter.



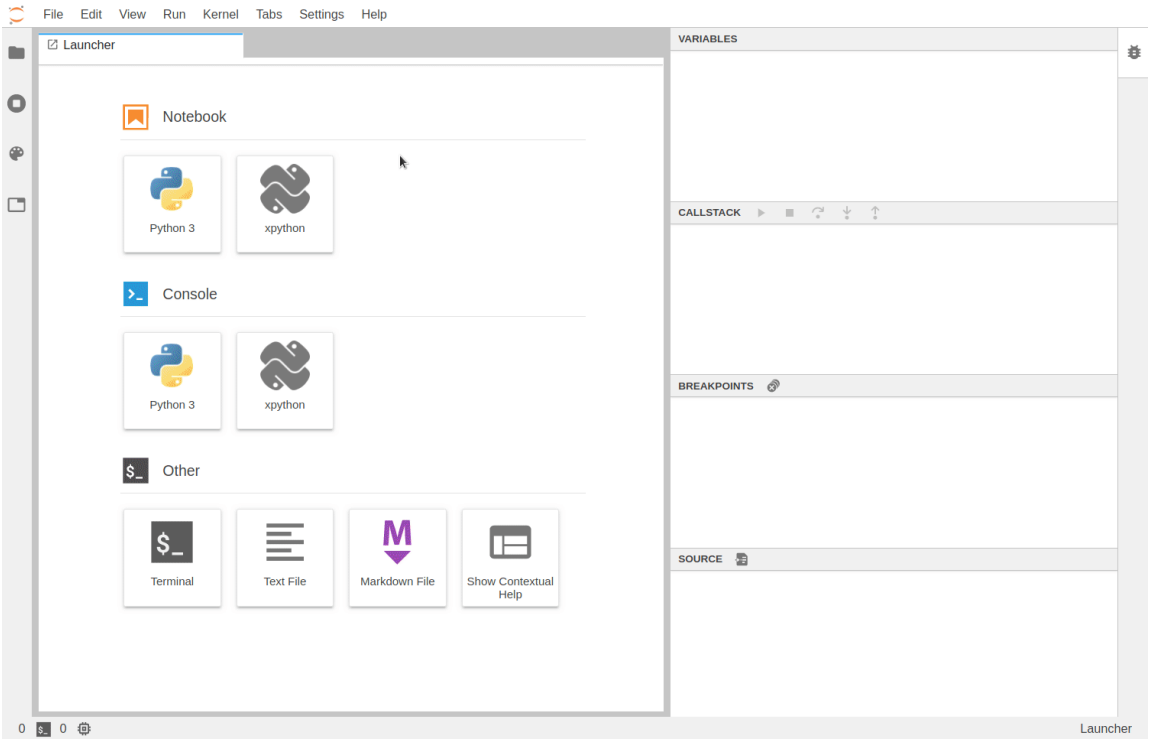
Jupyter ML-workspace

ML workspace es un entorno de desarrollo integrado todo en uno basado en la web dedicado al aprendizaje automático y la ciencia de datos.



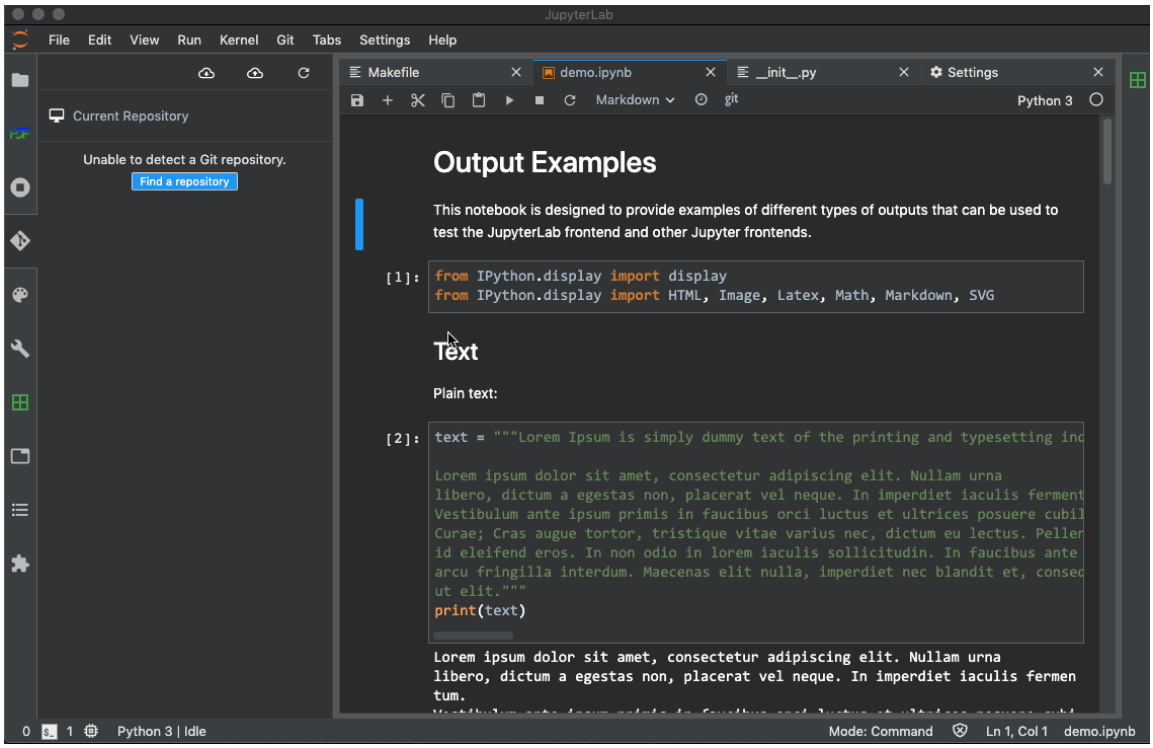
JupyterLab Debugger

Debugger es una extensión de JupyterLab que funciona como un depurador visual para notebooks, consolas y archivos fuente de Jupyter. Puede ayudarlo a identificar y corregir errores.



JupyterLab Git

Esta es una extensión de JupyterLab para Git, un sistema de control de versiones distribuido gratuito y de código abierto. Le permite controlar la versión. Simplemente lo usa abriendo la extensión Git desde la pestaña Git en el panel izquierdo.



Otras extensiones destacables:

- JupyterLab LaTeX
- JupyterLab variableInspector
- JupyterLab plotly
- JupyterLab bokeh
- Jupyter Dash
- JupyterLab Table of Contents
- JupyterLab SQL