

Flujo de trabajo con Jupyter Notebook

El toolkit que se presenta es un sistema que permite generar documentación ordenada de manera afable, así como la mantención del sistema de trabajo.

El sistema esta compuesto por: Anaconda, Python, Markdown, Jupyter Notebook y el Terminal. Cada una de las herramientas mencionadas presenta una función específica respecto al toolkit.

Anaconda (*Sistema de administración*)

Anaconda es una distribución de código abierto, diseñada específicamente para tareas asociadas al análisis de datos y computación científica. El objetivo es el sintetizar y unificar el conjunto de buenas prácticas en la mantención, administración y desarrollo de los proyectos.

Para instalar la distribución Anaconda, pueden dirigirse a <https://www.anaconda.com/download/> y escoger la **versión 3.6**.

Esta distribución instala Python 3.6 y Jupyter Notebook, en adición a las principales librerías para Data Science con Python.

De forma alternativa se puede instalar Miniconda, una distribución más liviana con menos prestaciones.

Markdown (*Formato afable de escritura*)

Es un lenguaje de marcas minimalista cuyo eje central es la conversión fácil a `html` y otros formatos. Markdown permite escribir utilizando un lenguaje **fácil de leer y escribir**.

Elemento	Sintaxis Markdown
<code># Título</code>	# Título
<code>## Subtítulo</code>	## Subtítulo
<code>### Subsubtítulo</code>	### Subsubtítulo
<code>#### Párrafo</code>	#### Párrafo
<code>_Cursiva_</code> o <code>*Negrita*</code>	<i>Cursiva</i>
<code>__Negrita__</code> o <code>**Negrita**</code>	Negrita
<code>[text](https://address)</code>	Inserción de hipervínculo
<code>![alt-text](/images/image.jpg)</code>	Inserción de imagen

Las listas pueden ser ordenadas o no ordenadas. Para las listas no ordenadas, precedemos cada elemento con `*`:

- Esta
- es
- una
- lista

Para las listas ordenadas, precedemos cada elemento con un número:

1. Esta
2. es
3. otra
4. lista

Se pueden agregar metadatos mediante un YAML.

Jupyter Notebook (*Python + Markdown*)

Jupyter Notebook (*antes denominado iPython Notebook*) es un ambiente de trabajo que busca unificar las herramientas de documentación y ejecución de código en un mismo archivo. Los notebooks permite la lectura fácil para los humanos, así como la compartimentalización del código para la máquina.

Jupyter acepta dos tipos de *chunks* (códigos): un formato de texto, preferentemente Markdown; y un *kernel*, que es el motor computacional que vincula la interpretación del código en el chunk mediante un lenguaje.

Jupyter expandió su funcionalidad a más de cuarenta lenguajes, destacando Javascript, Ruby, R, Julia, Haskell, entre otros. Aprender su funcionalidad básica es una buena idea para escribir documentación y otros.

Tips para entender Jupyter Notebook

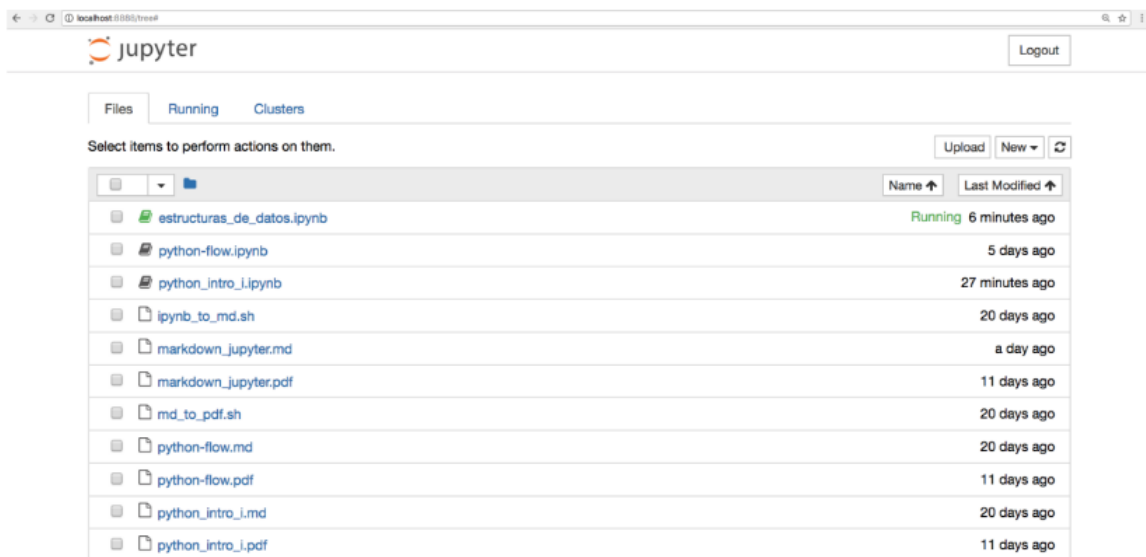
1. Funciona por chunks: Cada chunk puede contener texto o código. Cada función debe estar en el modo de comando (esto significa que deben presionar `Esc` antes de ejecutar).

Combinación de teclas	Acción
<code><Enter></code>	Entrar en modo edición
<code><Shift> + <Enter></code>	Ejecutar chunk y crear nuevo chunk
<code><Control> + <Enter></code>	Ejecutar chunk, no moverse
<code><Tab></code>	Invocar autocompletado (ver todas las opciones)
<code><Shift> + <Tab></code>	Ver documentación sobre la función
<code>M</code>	Cambiar la modalidad del chunk de código a markdown
<code>Y</code>	Cambiar la modalidad del chunk de markdown a código
<code>A</code>	Insertar chunk arriba
<code>B</code>	Insertar chunk abajo
<code>DD</code>	Eliminar chunk

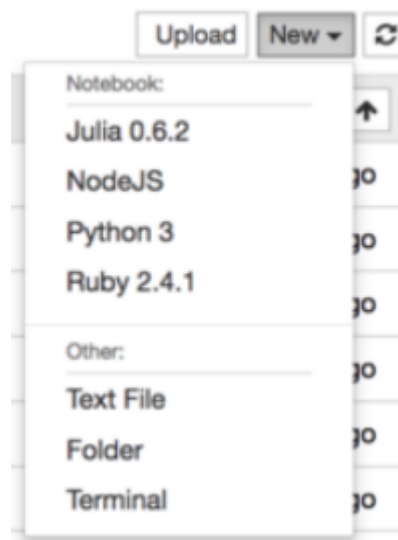
2. Dado que opera en base a chunks, su formato en sí es un poco ininteligible (es básicamente un `.json` con extensión `.ipynb`). Existen algunas alternativas que pueden resultar útiles más adelante:
 - GitHub soporta `.ipynb` y los visualiza en la página.
 - Jupyter Notebook ofrece opciones para convertir `.ipynb` a `.html`, `.doc` y `.pdf`.

Iniciando Jupyter Notebook

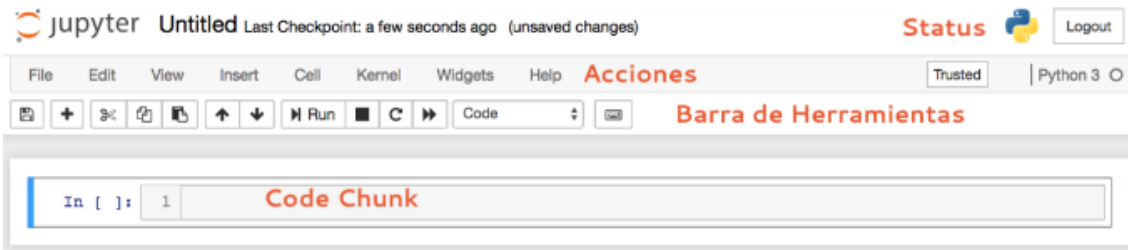
1. Jupyter Notebook funciona desde un navegador web por medio de un servidor. Es un buen consejo familiarizarse con los siguientes pasos:
 - En el terminal, escribir `jupyter notebook`. Aquí se podrá iniciar el servidor de manera local.
 - Por lo general `jupyter notebook` Jupyter Notebook abrirá el navegador web de forma automática. Si por algún motivo el navegador se cierra, Jupyter Notebook seguirá corriendo, solo debes volver al navegador y dirigirte a <https://localhost:8888/>.
2. Una vez dentro de `https://localhost:8888/`, Jupyter Notebook mostrará todos los documentos dentro del working directory, éste se iniciará donde se ejecutó `jupyter notebook`.



3. Para iniciar un notebook, debes hacer clic en *New* y seleccionar `Python 3`.



4. Un notebook nuevo sin contenido tendrá la siguiente apariencia. Dentro de éste se encontrarán varios elementos a considerar:



- En la esquina superior derecha, se encuentran los indicadores de **status** que informa sobre los estados del notebook y el kernel. Acá se indica si existe algún problema en el ambiente de trabajo.
- En la barra de **Acciones**, se encuentran las funciones que permiten administrar el notebook, tales como exportación a otros formatos, guardar notebook, reiniciar el kernel, entre otros.
- La **Barra de Herramientas** contiene atajos a las acciones más utilizadas dentro del notebook.
- Dentro del **Code Chunk** se inserta el código y texto.

5. El **Code Chunk** presenta dos modalidades:

- Modo Python: Se escribe el código que posteriormente será evaluado por el kernel de python.
- Modo Markdown: Se escribe el texto siguiendo el formato Markdown, que posteriormente será evaluado y representado con HTML.



6. Para escribir en el *Modo Python* nos debemos asegurar que el modo `code` esté seleccionado en nuestra barra de herramientas. Si todo funciona adecuadamente, al lado izquierdo del chunk visualizaremos `In []`



- Para ejecutar el chunk, se debe digitar `Shift + Enter`. Si el código no presenta problemas, nos devolverá el resultado interpretado por Python. El indicador `In [1]` alojará un número, que representa el orden de ejecución en el notebook.

7. Para ejecutar Markdown dentro del chunk, debemos cambiar al *Modo Markdown*. Si estamos en lo correcto, el contenido estará destacado siguiendo la sintaxis de Markdown.

```
1 # Este es un chunk dentro de Markdown
2
3 * Podemos hacer listas:
4   - Y otras cosas dentro de las listas
5
6 * También podemos hacer __tablas__
7
8 |Lorem|Ipsum|
9 |-----|-----|
10 |Dolor|Sit|
11 |Amet|
```

8. Ejecutar `shift + Enter` para obtener la versión HTML en el notebook.

```
In [1]: 1 print("Hola Mundo!")
```

Chunk Python

Hola Mundo!

```
1 # Hola Mundo
```

Chunk Markdown