### JupyterLab

[](https://1.bp.blogspot.com/-2Eo9n9fie-k/WsUcZ9XN2EI/AAAAAAAADPc/2G6hZkPVC7cfmJsRWTKa4uJJVwDeKZmpwCLcBGAs/s1600/jupyter.png)

**Índice**

[Primeros pasos con JupyterLab](https://python-para-impacientes.blogspot.com.es/2018/03/primeros-pasos-con-jupyterlab.html)

* Introducción
* Los cuadernos (Notebooks) y la aplicación JupyterLab
* Instalar JupyterLab
* Iniciar la aplicación JupyterLab
* Crear un cuaderno
* Modificar un cuaderno
* Cerrar un cuaderno y la aplicación JupyterLab

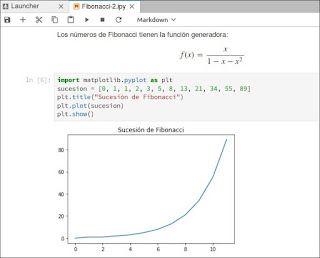
[Elementos de un cuaderno JupyterLab](https://python-para-impacientes.blogspot.com.es/2018/04/elementos-de-un-cuaderno-de-jupyterlab.html)

* Insertar una imagen y un texto
* Insertar una tabla con datos de un archivo CSV
* Insertar una ecuación
* Insertar un gráfico de líneas
* Insertar un enlace
* Insertar un vídeo de Youtube

Enlaces: <https://python-para-impacientes.blogspot.com/>

<https://www.adictosaltrabajo.com/2018/01/18/primeros-pasos-con-jupyter-notebook/>

## Primeros pasos **con** JupyterLab

[](https://2.bp.blogspot.com/-tWk0uyPLtLY/Wr-G7diipnI/AAAAAAAADNM/C7VMfdWWOd0glQgDS8cRQiV-dqaKSij-gCLcBGAs/s1600/jupyterlab0.jpg)

### Introducción

El proyecto **Jupyter** tiene su origen en la aplicación web **[Ipython Notebook](https://es.wikipedia.org/wiki/IPython" \l "Notebook" \t "_blank)**que permite registrar las sesiones de trabajo con Python en cuadernos electrónicos, que pueden incluir otros contenidos además del código fuente.

En el año 2014 se anunció un proyecto separado de **Ipython** para facilitar el desarrollo de **Notebook** y de otras herramientas para crear y compartir con facilidad este tipo de cuadernos. El proyecto ha evolucionado en el actual **JupyterLab** con un nuevo interfaz de usuario; estando en desarrollo su primera versión pero con betas disponibles para su instalación y disfrute.

Un **cuaderno de trabajo** está formado por un conjunto de celdas que pueden contener código fuente, textos explicativos, ecuaciones, gráficos, controles interactivos (*widgets*) y otros elementos. El código de una celda se puede ejecutar y el resultado que se obtiene se muestra a continuación de la celda como una parte del cuaderno, facilitando al usuario tanto el cambio de los datos como el análisis de la información resultante en tiempo real.

En la presente entrada mostraremos cómo se instala **JupyterLab** y los fundamentos para trabajar con los cuadernos.

### Los cuadernos (Notebooks) y la aplicación JupyterLab

Los cuadernos de **JupyterLab** gracias a su formato flexible se pueden modificar con facilidad para evaluar sobre la marcha nuevos resultados. Asimismo, resulta sencillo producir nuevas versiones de un mismo cuaderno, convertirlo a otros formatos y compartirlo con otras personas, favoreciendo esta característica el trabajo colaborativo.  
  
**JupyterLab** es una aplicación web con una arquitectura cliente-servidor que permite editar y ejecutar los cuadernos y otros tipos de archivos utilizando un navegador de Internet. La aplicación se puede ejecutar en cualquier equipo, incluso sin conexión a Internet. **JupyterLab** es compatible con las últimas versiones de **Firefox**, **Chrome** y **Safari**.  
  
El término **Jupyter** es un acrónimo que identifica a los lenguajes de programación Julia, Python y R que fueron los primeros en ser aceptados por la aplicación en sus comienzos. Hoy día afortunadamente se pueden usar [más lenguajes de programación](https://github.com/jupyter/jupyter/wiki/Jupyter-kernels).

### Instalar JupyterLab

Para instalar JupyterLab con el instalador **Pip**:

**$ pip3 install jupyterlab**

Existen también varias distribuciones Python como [**Anaconda**](https://anaconda.org/anaconda), **[WinPython](http://winpython.github.io/" \t "_blank)** o [**Python Anywhere**](https://www.pythonanywhere.com/) que incluyen IPython, Jupyter Lab y/o Jupyter Notebook y otros módulos de uso bastante extendido.  
Para más información relacionada con la instalación de JupyterLab:

* <https://jupyterlab.readthedocs.io/en/stable/index.html>
* <https://github.com/jupyterlab/jupyterlab>

Antes de iniciar la aplicación Jupyterlab crearemos una carpeta llamada "**Cuadernos**" para los ejemplos de esta guía, accediendo a la línea de comandos y ejecutando:

**$ mkdir Cuadernos**

**$ cd Cuadernos**

### Iniciar la aplicación Jupyterlab

Como arrancamos Jupyter desde la terminal, se abrirá el workspace en la carpeta en la que estemos (que generalmente será nuestro home de usuario). Es conveniente saber que no permite navegar a carpetas superiores a la inicial. Con el botón New, podemos crear:

* Carpetas (recomendable para organizar todo).
* Archivos de texto (con un editor bastante simple).
* Terminales.
* Cuadernos, que es la opción que vamos a estudiar en este tutorial.

Cuando creamos un cuaderno, se nos abre en una pestaña nueva. La interfaz es muy sencilla, con una barra de menú y herramientas y una celda vacía, que es la unidad básica de trabajo con Jupyter. En una celda, podremos escribir una o varias líneas de código, y ejecutarla. También podemos escribir texto plano, y eliminar la celda del flujo de ejecución del cuaderno o hacer que se muestre de manera diferente si queremos que funcione como título, separador de sección, etc. Lo veremos más adelante.

Es importante remarcar que tenemos dos modos de trabajo en Jupyter:

Edición, que permite modificar el contenido de las celdas, como si fuera un editor de texto. La celda que tengamos seleccionada se muestra en verde. Podemos entrar en el mismo seleccionando una celda y pulsando *Enter*.

Mando, que nos permite ejecutar celdas o modificar el cuaderno y su estructura. La celda seleccionada se muestra en azul. Podemos volver al modo de mando pulsando *Esc*.



Para ejecutar una celda, haremos *Control + Enter*, o si queremos pasar automáticamente a la siguiente después de ejecutarla, podemos hacer *Shift + Enter*. Para saber cuándo una celda se ha ejecutado, está en ello o ya ha terminado, podemos mirar la cabecera de la celda:

Si solamente aparece **In [ ]**, quiere decir que aún no se ha ejecutado.

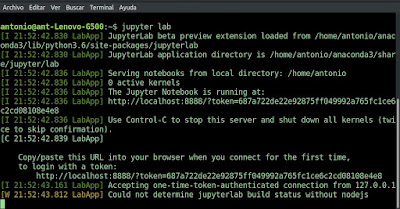
Si lo que tenemos es **In [\*]**, entonces está en proceso de ejecución.

Si aparece un número entre los corchetes, ya ha terminado de ejecutar. La única *pega* es que podemos haber hecho cambios posteriores a su ejecución, y esto no se indica de ninguna manera.

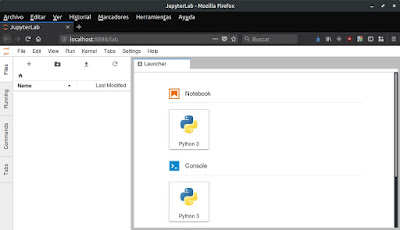
Todos los atajos de teclado disponibles están en el botón 

Continuamos con la creación de un sencillo cuaderno para ir conociendo la aplicación. Para iniciar la aplicación Jupyterlab ejecutar:

**$ jupyter lab**

[](https://3.bp.blogspot.com/-egkELpFhGp0/Wr-VHqTjogI/AAAAAAAADNc/Hh6C6_4lV20dEljmwFgxkkJJ03fzgO1fwCLcBGAs/s1600/jupyterlab2.jpg)

A continuación, veremos que en la consola se ejecuta un servidor web y que la aplicación **JupyterLab** se abre en el navegador Internet predeterminado en la dirección [http://localhost:8888](http://localhost:8888/), quedando disponible para su uso:

[](https://4.bp.blogspot.com/-gV1kkRRmXXw/Wr-DxiJgCiI/AAAAAAAADMw/y4YeTSRZ-301CUWG1xYzKf-wQrU2hWpLgCLcBGAs/s1600/jupyterlab3.jpg)

**JypiterLab** consta de una barra con el menú típico de opciones en la parte superior. Debajo de este menú la ventana se divide en dos grandes áreas:

* En el **área de la izquierda**: en "**Files**" se pueden gestionar los archivos y carpetas de la aplicación; en "**Running**" hacer un seguimiento de las sesiones y si es necesario detener núcleos (kernels), en "**Commands**" encontrar y ejecutar cualquier comando de JupyterLab y en "**Tabs**" seleccionar un área de trabajo entre las disponibles.
* En el **área de la derecha** llamada "**Launcher**" se muestran, por defecto, varios accesos directos para editar cuadernos, iniciar sesiones en las distintas consolas y editar de textos.

Las áreas son dimensionables y la de la derecha puede extenderse o no a la anchura del navegador presionando la tecla [**Control+B**] o con la opción "**Show Left Area**" del menú "**View**".

### Crear un cuaderno

Para crear un cuaderno hacer clic sobre el acceso directo "**Notebooks Python 3**" en el área "**Launcher**" o seleccionar la opción "**New Notebook**" del menú "**File**".

A continuación, aparecerá en el área de la derecha una nueva pestaña con el nombre del cuaderno "***Untitled***", una barra de herramientas y una celda vacía esperando a que se introduzca algún contenido.

Para empezar cambiaremos el nombre del cuaderno por "**Fibonacci**" seleccionando la opción "**Rename Notebook**" del menú "**File**" o con la opción "**Rename**" del menú contextual que aparece al presionar con el botón derecho del ratón sobre el nombre del archivo en el área de la izquierda. Después, escribir el nuevo nombre y aceptar el cambio con la tecla [**Enter**]. La aplicación le agrega la extensión **.ipynb** que es la que se utiliza para los cuadernos.

Para escribir texto en la primera celda seleccionar "**[Markdown](http://python-para-impacientes.blogspot.com.es/2014/09/markdown-para-python-i.html" \t "_blank)**" en la lista de selección de la barra de herramientas donde aparece la palabra "**Code**". Después, hacer clic en la celda y escribir el siguiente texto:

|  |
| --- |
| ## La sucesión de Fibonacci  La sucesión de \*\*Fibonacci\*\* comienza con los números 0 y 1, y a partir de estos, cada término es la suma de los dos anteriores. El siguiente código Python muestra los números de la serie (hasta 100): |

Después hacer clic sobre el botón "**+**" ("**Insert A Cell Bellow**") o presionar [**Alt+Enter**] para insertar una nueva celda después de la actual y escribir en ella el siguiente código Python:

def fib(n):

a, b = 0,1

while a < n:

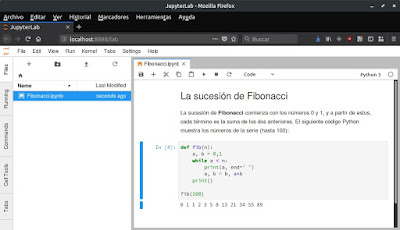
print(a, end=' ')

a, b = b, a+b

print()

fib(100)

Cuando hayas finalizado se puede ejecutar el contenido de ambas celdas de una vez eligiendo la opción "**Run All Cells**" en el menú "**Run**". El texto descriptivo mejorará su apariencia gracias a los caracteres de formato de **[Markdown](http://python-para-impacientes.blogspot.com.es/2014/09/markdown-para-python-i.html" \t "_blank)** que se incluyen en él y el resultado de ejecutar el código Python se mostrará debajo del mismo: ***0 1 1 2 3 5 8 13 21 34 55 89***.

[](https://3.bp.blogspot.com/-RU5euVCbCbM/Wr-FMbXf30I/AAAAAAAADNA/0gUgHRIKBfIhJuF_hOQt_QvrNXRr2xD-gCLcBGAs/s1600/jupyterlab4.jpg)

Para guardar hacer clic en el icono de guardar "**Save the Notebook...**" o seleccionar la opción equivalente en el menú "**File**", o bien, presionar la combinación de teclas [**Control+S**]. Cuando exista algún contenido pendiente de guardar aparecerá en la pestaña un círculo a la derecha del nombre del cuaderno, que cambiará a "**X**" cuando todo esté guardado.  
  
Para exportar el cuaderno en **HTML**, **Markdown**, **PDF**, etc. elegir el formato en la opción "**Export Notebook As...**" del menú "**File**".

Para cerrar el cuaderno seleccionar "**Close Notebook**" en el menú "**File**" y ya hemos terminado nuestro primer cuaderno.

### Modificar un cuaderno

Cuando sea necesario abrir el cuaderno hacer doble clic sobre su nombre o seleccionar la opción "**Open**" del menú contextual. En ambos casos primero se abrirá el cuaderno y con posterioridad se ejecutarán todas las celdas. Después, para editar el contenido de alguna celda: hacer doble clic sobre una celda con texto o clic sobre una celda con código y reescribir su contenido.

Por ejemplo, para obtener los números de Fibonnaci que existen hasta el número 234 hacer doble clic sobre el número "100" de la función ***fib()*** en la última línea de la celda del código Python para seleccionar el número y escribir 234 para sustituir dicho valor.

Para editar cualquier línea de código lo normal es hacer clic a la altura de la línea y del carácter a modificar. Para volver a ejecutar la celda actual presionar [**Control+Enter**]. Como se puede apreciar con Jupyterlab es fácil modificar cuadernos para comprobar nuevos resultados. Para concluir, volvemos a reponer el valor "100" en la función para dejar el código como estaba antes, deshaciendo el último cambio con [**Control+Z**].

En el menú "**Edit**" se encuentran todas las opciones para operar con celdas. Básicamente, una o más celdas se pueden seleccionar o no, cortar, copiar, pegar y mover. Con la opción "**Split Cell**" se puede dividir una celda en dos. La división se efectuará por la posición que ocupe el cursor. También, dos o más celdas seleccionadas se pueden agrupar en una con la opción "**Merge Selected Cells**".

Para borrar las salidas que se situán en el espacio contiguo a una celda y que resultan de ejecutar código fuente Python seleccionar las opciones "**Clear Output**" o "**Clear All Outputs**" en el menú "**Edit**".

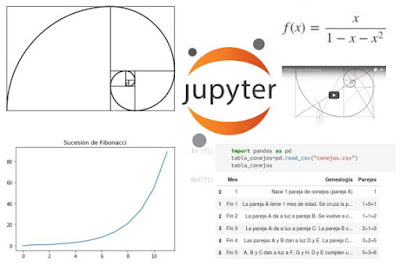
También, en la barra de herramientas del cuaderno hay botones con las opciones de edición de más uso.  
  
Por último, comentar que existen varias opciones para ejecutar el contenido de las celdas en el menú "**Run**". Para ejecutar celdas individuales o todas la celdas; para ejecutar una celda e insertar una nueva a continuación y para ejecutar las celdas anteriores o posteriores a una celda determinada.

### Cerrar el cuaderno y la aplicación JupyterLab

Para cerrar el cuaderno y finalizar la sesión de trabajo seleccionar "**Close and Shutdown Notebook**" en el menú "**File**" y cerrar la pestaña del navegador.

Para detener el servidor web acceder a la consola donde se inició JupyterLab y presionar [**Control+C**]. Después, responder al siguiente mensaje: "***Shutdown this notebook server (y/[n])?***" presionando la tecla [**y**].

## Elementos de un cuaderno de JupyterLab

[](https://2.bp.blogspot.com/-VmSF01gnTKQ/WsFQWt9z3-I/AAAAAAAADNs/Y5b1ml4Txm8VRyk3dtN6JehYKWCoW2YRgCLcBGAs/s1600/elementos.jpg)

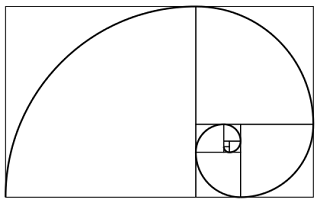
En esta entrada crearemos otro cuaderno a partir del archivo "Fibonacci.ipynb" de la [entrada anterior](http://python-para-impacientes.blogspot.com/2018/03/primeros-pasos-con-jupyterlab.html) en el que insertaremos nuevos elementos. En el cuaderno se agregarán celdas con textos, una imagen, una tabla construida con los datos de un archivo CSV, una ecuación, un gráfico de líneas, un enlace web y un vídeo de Youtube.

Para comenzar con la tarea, iniciar Jupyterlab ejecutando desde la línea de comandos:

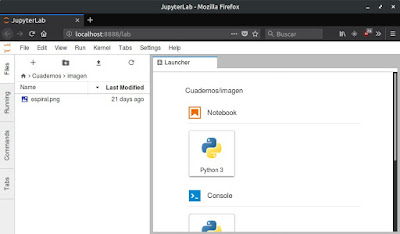
**$ jupyter lab**

Una vez que arranque la aplicación web accederemos a la carpeta "**Cuadernos**" desde "**Files**" y crearemos una carpeta llamada "imagen" haciendo clic en el icono "**New Folder**" de la barra de herramientas.

Después, descargaremos la siguiente imagen de la famosa Espiral de Fibonacci a dicha carpeta:

[](https://2.bp.blogspot.com/-ChVZ6snCJUc/WsFR19vwbQI/AAAAAAAADN0/eXH9H5wriEoEr9HG3sgfzP7qh5Eo8tVaQCLcBGAs/s1600/espiral.png)

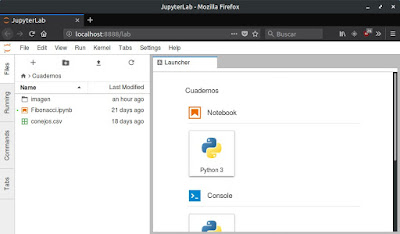
Cualquier archivo del disco local o de un recurso compartido de una red se puede incorporar a la carpeta actual utilizando la opción "**Upload Files**" (se encuentra a la derecha de "**New Folder**" en la barra de herramientas).

[](https://2.bp.blogspot.com/-wetQdhh9GRo/WsFaxeEHtdI/AAAAAAAADOc/HunAY3ns5_ATd1Dp4mnor19QdzVyJzaVwCLcBGAs/s1600/jupyterlab2-6.jpg)

También, para mostrar más adelante cómo insertar una tabla con los datos de una archivo **CSV** crearemos en el editor de textos del panel "Launcher" un archivo de textos "***untitled.txt***" en la carpeta "**Cuadernos**" con el siguiente contenido:

|  |
| --- |
| **Mes,Genealogia,Parejas** 1,Nace 1 pareja de conejos (pareja A),1 Fin 1,La pareja A tiene 1 mes de edad. Se cruza la pareja A,1+0=1 Fin 2,La pareja A da a luz a pareja B. Se vuelve a cruzar pareja A,1+1=2 Fin 3,La pareja A da a luz a pareja C. La pareja B cumple 1 mes. Se cruzan las parejas A y B.,2+1=3 Fin 4,"Las parejas A y B dan a luz D y E. La pareja C cumple 1 mes. Se cruzan parejas A, B y C",3+2=5 Fin 5,"A, B y C dan a luz a F, G y H. D y E cumplen un mes. Se cruzan A, B, C, D y E",5+3=8 |

Para salvar los datos presionar [**Control+S**]. Para finalizar los preparativos renombrar el archivo "**untitled.txt**" con "**conejos.csv**" con la opción "**Rename**" del menú contextual que aparece al hacer clic con el botón derecho sobre el nombre del archivo.

[](https://2.bp.blogspot.com/-V5k41p0cJGE/WsFbrds_7DI/AAAAAAAADOo/pCrNl7Ms2RYO0khQhwzSUK0r5vNdMZ3twCLcBGAs/s1600/jupyterlab2-5.jpg)

Un archivo **CSV** (de Valores Separados por Comas) es un tipo de documento que representa los datos de forma parecida a una tabla, es decir, organizando la información en filas y columnas.

En un archivo **CSV** los datos de las diferentes columnas son separados habitualmente por un signo de puntuación (una coma, un punto y coma, etc.) u otro carácter que actúa como separador. En cambio, las filas suelen separarse por un salto de línea. Además, muchas veces los datos suelen ir precedidos por un encabezado con los nombres de campos o identificadores de columnas, como puede apreciarse en el ejemplo.

Un archivo CSV almacena la información en un formato accesible que facilita el intercambio entre aplicaciones. Es frecuente en muchos programas que manejan datos contar con una opción para importar o exportar la información en dicho formato. También, las aplicaciones ofimáticas de hojas de cálculos (Calc, Google Sheets, Gnumeric, Calligra Sheets, Excel, etc.) incorporan opciones para abrir y guardar la información en formato CSV.

A continuación, hacer clic con el botón derecho sobre el nombre del archivo "Fibonacci.ipynb" para mostrar el menú contextual y seleccionar la opción "Duplicate" para crear una copia del mismo: "Fibonacci-Copy1.ipynb". Después, volver a mostrar el menú contextual y seleccionar "Rename" para renombrar el cuaderno como "Fibonacci-2.ipynb". Por último, abrir la copia del cuaderno haciendo doble clic sobre su nombre.

### Escribiendo código

Es importante, a la par que evidente, tener instalados los paquetes de Python que queramos utilizar. Una vez hecho, podemos importarlos al cuaderno. Para los ejemplos utilizados en este tutorial, será necesario instalar el paquete matplotlib

|  |  |
| --- | --- |
| def busqueda\_binaria(array, valor):  ary = array  a = 0  b = len(array)  while a < b:  m = (a + b) // 2  if array[m] == valor:  return m  elif array[m] < valor:  a = m + 1  else:  b = m  return None  x1 = [0,1,2,3,4,5,6,7]  x2 = [1,4,8,12,41]  x3 = [0,1,2,4,8,16,32] | |
| Busqueda\_binaria(x1,3)  3 |
| Busqueda\_binaria(x1,3)  3 |
| Busqueda\_binaria(x1,3)  3 |
| Busqueda\_binaria(x1,3)  3 |
| Busqueda\_binaria(x1,3)  3 |

#### 5.1. LaTeX

Para mostrar código LaTeX en nuestro cuaderno, simplemente tenemos que escribir entre dólares (simples o dobles, según queramos), y marcar la celda como texto plano (Markdown).

#### 5.2. Otros lenguajes

Para poder trabajar en Jupyter con otros lenguajes, se han desarrollado diferentes \_kernels\_. Los más utilizados, aparte del propio python, son:

* Haskell: kernel *ihaskell*.
* Javascript: kernel *ijavascript*.
* Julia: kernel *ijulia*.

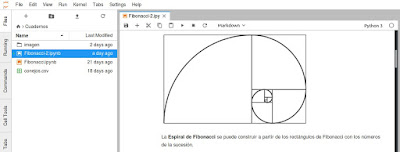
Para utilizarlos previamente deberíamos añadirlos

Una vez instalados, simplemente seleccionamos el kernel deseado en la barra de menú de Jupyter.

### Insertar una imagen y un texto

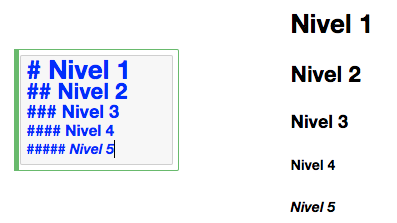
Para insertar la imagen de la espiral de Fibonnaci con una descripción seleccionar la última celda y agregar una nueva al final del cuaderno del tipo "**Markdown**" haciendo clic en el botón "**+**". En ella incluir el siguiente código que ejecutaremos con [**Control+Enter**]:

### 

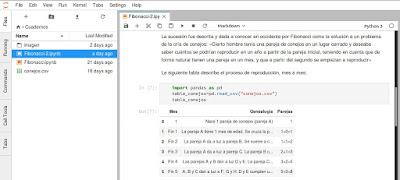
[](https://1.bp.blogspot.com/-qR-1ybDovoE/WsFZTLGMRnI/AAAAAAAADOI/FT9oFRyXvmg7wXH2GoHoVOj5VPu5HhAqwCLcBGAs/s1600/jupyterlab2-1.jpg)Para insertar la imagen de la espiral de Fibonnaci con una descripción seleccionar la última celda y agregar una nueva al final del cuaderno del tipo "**Markdown**" haciendo clic en el botón "**+**". En ella incluir el siguiente código que ejecutaremos con [**Control+Enter**]:

|  |
| --- |
| ![title](imagen/espiral.png "Espiral de Fibonacci")  La \*\*Espiral de Fibonacci\*\* se puede construir a partir de los rectángulos de Fibonacci con los números de la sucesión. |

Varias cositas referentes a la escritura de texto plano:

* Como se ha mencionado antes, se debe indicar que el tipo de celda es Markdown.
* Podemos separar el contenido del documento con cabeceras para secciones, subsecciones, subsubsecciones… Es tan fácil como añadir de una a cinco almohadillas (#) al principio de la línea, tantas como el nivel que queramos tenga el título.
* Podemos añadir imágenes, indicando anchura, alineación, etc., con el comando: <img src='...' align="...">
* Si queremos mostrar una lista, con *guión + espacio* se formateará correctamente.
* ¡Podemos incluso añadir emojis!

### Insertar una tabla con datos de un archivo CSV

[](https://4.bp.blogspot.com/-QEjwPgHIu3o/WsFZfake2CI/AAAAAAAADOM/EE6mMqY05BQv0GmqezLOGylX2OTU3q9gQCLcBGAs/s1600/jupyterlab2-2.jpg)

Para insertar una tabla con los datos del archivo "**conejos.csv**" agregar una celda al final del cuaderno del tipo "**Markdown**" haciendo clic en el botón "**+**" con el siguiente texto y ejecutar con [**Control+Enter**]:

|  |
| --- |
| La sucesión fue descrita y dada a conocer en occidente por Fibonacci como la solución a un problema de la cría de conejos: «Cierto hombre tenía una pareja de conejos en un lugar cerrado y deseaba saber cuántos se podrían reproducir en un año a partir de la pareja inicial, teniendo en cuenta que de forma natural tienen una pareja en un mes, y que a partir del segundo se empiezan a reproducir»  La siguiente tabla describe el proceso de reproducción, mes a mes: |

Después, hacer clic en el botón "**+**" o presionar la tecla [**B**] para insertar una nueva celda al final del cuaderno del tipo "**CODE**" con el siguiente código Python y ejecutar con [**Control+Enter**]:

import pandas as pd

tabla\_conejos=pd.read\_csv("conejos.csv")

print(tabla\_conejos)

En el código anterior se importa **Pandas**, una biblioteca para manipulación y análisis de datos para Python. Es bastante eficiente en el manejo de archivos CSV, permitiendo obtener los datos de un número de filas desde el comienzo o final de la tabla, seleccionar columnas, filtrar la información, etc.. como en los siguientes ejemplos:

# Obtener las tres primeras filas desde el comienzo de la tabla

tabla\_conejos=pd.read\_csv('conejos.csv').head(3)

tabla\_conejos

# Obtener sólo los datos de las columnas "Mes" y "Parejas"

tabla\_conejos=pd.read\_csv('conejos.csv', usecols=['Mes', 'Parejas'])

tabla\_conejos

### Insertar una ecuación

Para incluir ecuaciones en un cuaderno utilizaremos la codificación **[LaTex](https://es.wikipedia.org/wiki/LaTeX" \t "_blank)** dentro de texto Markdown o con código Python generando las ecuaciones con el módulo **Latex**:

Para insertar una ecuación en el cuaderno agregar una celda de tipo "**Markdown**" al final del cuaderno (presionando la tecla [**B**] con el siguiente texto y ejecutar con [**Control+Enter**]:

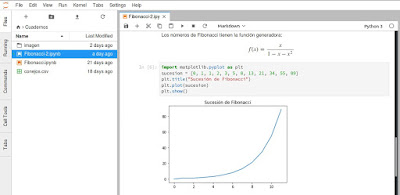
|  |
| --- |
| Los números de Fibonacci tienen la función generadora:  $$f(x) = \frac{x}{1 - x - x^2}$$ |

El carácter **$** delimita el comienzo y el final de la ecuación. Además, en este ejemplo el segundo carácter **$** en cada extremo hace que cuando se ejecute la celda se muestre la fórmula en el centro de la línea con respecto a los márgenes izquierdo y derecho.  
  
El siguiente código genera la misma fórmula con **Latex** pero con código Python:

from IPython.display import display, Math, Latex

display(Math(r'f(x) = \frac{x}{1 - x - x^2}'))

### Insertar un gráfico de líneas

[](https://2.bp.blogspot.com/-uisUDFcUZjE/WsFZ6z76FHI/AAAAAAAADOU/5aabhkoKos8i7h_NdF5MLI5BQIZrnhdOwCLcBGAs/s1600/jupyterlab2-3.jpg)

Para insertar un gráfico de líneas que representa la sucesión de Fibonacci agregar una nueva celda del tipo "**CODE**" haciendo clic en el botón "**+**", incluir el siguiente código Python y ejecutar con [**Control+Enter**]:

import matplotlib.pyplot as plt

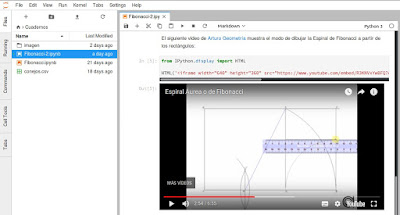
sucesion = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

plt.title("Sucesión de Fibonacci")

plt.plot(sucesion)

plt.show()

### Insertar un enlace

[](https://4.bp.blogspot.com/-XIDur3aTsCg/WsFaNJAMLdI/AAAAAAAADOY/KzBNt1EOfUwdYb6WMJ5L4PExQA8jzY2cQCLcBGAs/s1600/jupyterlab2-4.jpg)

Para insertar un texto que incluye un enlace web agregar una nueva celda al final del cuaderno del tipo "**Markdown**" haciendo clic en el botón "**+**". En ella incluir el siguiente código y ejecutar con [**Control+Enter**]:

|  |
| --- |
| El siguiente video de [Arturo Geometría](https://www.youtube.com/user/ArturoGeometria) muestra el modo de dibujar la Espiral de Fibonacci a partir de los rectángulos: |

### Insertar un video de Youtube

Para insertar un vídeo de **Youtube** agregar una nueva celda del tipo "**CODE**" haciendo clic en el botón "**+**". En ella incluir el siguiente código Python que se ha contruído copiando el código de inserción que se obtiene en la propia página del video dentro de la función **HTML()** y ejecutar con [**Control+Enter**]:

|  |
| --- |
| from IPython.display import HTML  HTML('<iframe allow="autoplay; encrypted-media" allowfullscreen="" frameborder="0" height="360" src="https://www.youtube.com/embed/R3KNVxYwBFQ?ecver=1" width="640"></iframe>') |

Otra posibilidad es utilizar la [función mágica](http://python-para-impacientes.blogspot.com.es/2014/08/funciones-magicas-de-ipython.html) **%%HTML** en una celda con el código **HTML** que incrusta el vídeo:

|  |
| --- |
| %%html <iframe allow="autoplay; encrypted-media" allowfullscreen="" frameborder="0" height="360" src="https://www.youtube.com/embed/R3KNVxYwBFQ?ecver=1" width="640"></iframe> |

Para cerrar el cuaderno y finalizar la sesión de trabajo seleccionar "**Close and Shutdown Notebook**" en el menú "**File**". Para detener el servidor web acceder a la consola donde se inició JupyterLab y presionar [**Control+C**]. Después, cuando aparezca el siguiente mensaje: "**Shutdown this notebook server (y/[n])?**" presionar la tecla [**y**].