

# Entornos de programación de IA con Python.

## Caso práctico

Han pasado ya algunos meses desde que Miguel empezó a trabajar en Pick&Deliver con la misión de automatizar y optimizar procesos del negocio con inteligencia artificial. Ya hay varias tareas de la empresa que se resuelven de forma rápida y eficiente gracias a aplicaciones de IA (Inteligencia Artificial) que se han implantado.

Las mejoras se han ido notando, y han multiplicado clientes y facturación. El grado de satisfacción de los clientes también ha aumentado bastante, y han recomendado los servicios de Pick&Deliver. Son buenas noticias, pero también constituyen un auténtico reto, porque el incremento rápido de clientes está mostrando que hay algunos cuellos de botella que antes no se habían detectado y que ahora amenazan con hacer a la empresa "morir de éxito".

Miguel se da cuenta de que toca hacer un análisis más profundo y particularizado a las distintas divisiones de la empresa, y para eso debe investigar y utilizar desarrollos de inteligencia artificial propios. ¡Llega el momento de ponerse a programar! Y eso a Miguel le encanta.

Lorena es una estudiante de inteligencia artificial que está haciendo sus prácticas en la empresa, y está ayudando a Miguel en las distintas tareas y desarrollos de Pick&Deliver. Ahora, Miguel le cuenta que van a implementar sus propios modelos de deep learning y que tendrán que decidir con qué herramientas trabajar. Lorena recuerda haber probado, durante su etapa de aprendizaje, distintos tipos de entorno de programación, desde Visual Studio Code, hasta Google Colaboratory. Miguel ha participado en algunas competiciones de la plataforma Kaggle, y está muy acostumbrado al entorno del kernel de Kaggle.

Mientras toman un café, plantean todas las opciones disponibles para elegir cuál será su medio de trabajo para los próximos meses, que se presentan de lo más interesantes. (La siguiente imagen ha sido generada por la API (Application Programming Interfaces) DALL·E 2, de OpenAI, al pedirle una "foto de dos programadores, hombre y mujer, tomando café delante de un ordenador con código en la pantalla").



DALLE2 (<https://openai.com/dall-e-2/>) (Dominio público)

En esta unidad vamos a analizar varios entornos habitualmente utilizados cuando se programa deep learning con Python. La mayoría de ellos admiten, también, otros muchos lenguajes. Vamos a distinguir, principalmente, entre dos categorías:

- ✓ Entornos basados en Jupyter notebook: son interfaces que permiten el tratamiento de ficheros en el lenguaje "notebook", que soporta código ejecutable junto a textos e imágenes. El código está contenido en celdas que se pueden ejecutar de forma independiente.
- ✓ IDE o entornos de desarrollo integrados: se trata de software orientado a la creación de aplicaciones o proyectos. Cuentan con todas las herramientas que puedan ser necesarias para la creación de un producto final.

La principal diferencia entre ellos es que un notebook es un archivo que refleja el planteamiento de un análisis, y un IDE (Integrated Development Environment) cuenta con las funcionalidades y el medio para crear una aplicación.

En ambas categorías, veremos las opciones:

- ✓ Interfaz de escritorio, que es necesario descargar e instalar.
- ✓ Interfaz online, en navegador, basada en servicios en la nube.



Ministerio de Educación y Formación Profesional  
(<https://www.educacionyfp.gob.es/portada.html>) (Dominio público)

## Materiales formativos de FP Online propiedad del Ministerio de Educación y Formación Profesional.

[Aviso Legal](http://www.educacionyfp.gob.es/fpadistancia/comunes/aviso-legal.html) (<http://www.educacionyfp.gob.es/fpadistancia/comunes/aviso-legal.html>)

# 1.- Jupyter Notebook y entornos compatibles.

## Caso práctico

A Miguel le han propuesto dar una serie de ponencias y talleres en una convención internacional y ha propuesto enseñar como caso de éxito las mejoras que han implementado en Pick&Deliver. En la empresa les parece una muy buena forma de darse a conocer como una compañía innovadora y en la vanguardia tecnológica, así que le han dado vía libre para prepararlo y viajar. La tarea de investigar los entornos de tipo notebook para modelos de machine learning adaptados a las necesidades de Pick&Deliver queda en manos de Lorena.



LookStudio (<https://www.freepik.es/autor/lookstudio>) (CC BY-SA  
(<http://creativecommons.org/licenses/?lang=es>))

Lorena ha aprendido a utilizar Jupyter Notebook a lo largo de sus estudios, pero sabe que hay novedades y más opciones, así que empieza a indagar para ver qué herramienta puede ser la más adecuada para la misión que tienen entre manos.



[Jupyter](https://jupyter.org/) (<https://jupyter.org/>) (CC BY-SA (<http://creativecommons.org/licenses/?lang=es>))

El [Proyecto Jupyter](https://jupyter.org/) (<https://jupyter.org/>) se creó en 2014, a partir de IPython, que era una consola para programación en Python, por Fernando Pérez, para dar especial protagonismo al lenguaje Notebook, que es agnóstico y permite el desarrollo con un formato muy práctico en otros lenguajes. Está especialmente enfocado en tres lenguajes para el ámbito de la Ciencia de Datos: Python, R y Julia, de hecho, su nombre viene de unir estos tres: **Ju**-lia, **Pyt**-hon, **R**. Actualmente admite una gran cantidad de lenguajes de programación.

Una de las razones por las que se ha hecho tan famoso, es por ser un software de código abierto, gratuito, con experiencia basada en navegador, y con una filosofía completamente volcada en facilitar el desarrollo de la ciencia de datos y la computación científica, simplificando bastante la parte de configuración y puesta en marcha, pues inicialmente, no estaba orientado a proyecto final, sino más a investigación.

El formato Notebook es especialmente bueno para educación. Existe un buen número de cursos cuyo material didáctico está formado, íntegramente, por notebooks. Como alumno, también es muy útil ir desarrollando y anotando observaciones como en un cuaderno de papel, al tiempo que se puede ir ejecutando código por partes, controlando mejor dónde hay errores o falta algo.

Con el tiempo, su uso se ha extendido y normalizado de tal manera, que ya vimos en la unidad 2 que está implementado como interfaz front-end en algunas plataformas cloud para la aplicación del aprendizaje automático a nivel profesional.

## Autoevaluación

El proyecto Jupyter se lanzó pensando específicamente en facilitar la ciencia de datos en tres lenguajes de programación que tienen gran protagonismo en dicho campo de estudio. Estos lenguajes, son  ,  y  , y la palabra Jupyter es una combinación de parte del nombre de los tres.

Los lenguajes son Julia, Python y R

## 1.1.- Jupyter Notebook.

La mayor parte del flujo de trabajo en la ciencia de datos transcurre en Notebooks, pues este formato ha contribuido a agilizar dicho proceso y a poder evaluar resultados y compartirlos con otros de una forma más visual y clara. Para poder utilizarlo, por ejemplo, con todos los paquetes necesarios para aprendizaje automático con Python, es recomendable recurrir a una distribución que traiga ya todo integrado, como es el caso de Anaconda. Esta distribución ofrece un navegador que mejora mucho la experiencia del usuario con el que se puede configurar y gestionar con comodidad entornos, paquetes y aplicaciones.

The screenshot shows the Anaconda Navigator application window. The top bar includes the Anaconda logo, the title "Anaconda Navigator", and a "Sign in" button. On the left, there's a sidebar with links for "Home", "Environments", "Learning", "Community", "Premium packages and dedicated support.", "Documentation", and "Anaconda Blog". Below the sidebar are social media icons for Twitter, YouTube, and GitHub. The main area is titled "Installed" and contains a table of packages. The columns are "Name", "Description", and "Version". The table lists several packages, each with a green checkmark icon and a brief description. At the bottom of the table, it says "340 packages available".

Name	Description	Version
_ipyw_jlab_nb_ex...	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
alabaster	Configurable, python 2+3 compatible sphinx theme.	0.7.12
anaconda	Simplifies package management and deployment of anaconda	2020.11
anaconda-client	Anaconda cloud command line client library	1.7.2
anaconda-project	Tool for encapsulating, running, and reproducing data science projects	0.8.4
applaunchservices		0.2.1
appnope	Disable app nap on os x 10.9	0.1.0
appscript	Control applescriptable applications from python.	1.1.1
argh	The natural cli.	0.26.2
argon2-cffi	The secure argon2 password hashing algorithm.	20.1.0
asn1crypto	Python asn.1 library with a focus on performance and a pythonic api	1.4.0
astroid	A abstract syntax tree for python with inference support.	2.4.2

La forma más sencilla de iniciarse en la utilización de esta herramienta, es instalar Anaconda y arrancar Jupyter desde el escritorio propio de Anaconda. Lo primero que se muestra es el escritorio con los directorios y notebooks que pueda haber en el directorio principal del entorno.

The screenshot shows the Jupyter interface running on [try.jupyter.org](https://try.jupyter.org). The top navigation bar includes icons for back, forward, and search, followed by the URL "try.jupyter.org". On the right, it says "Hosted by Rackspace" with a logo. Below the header, there are tabs for "Files", "Running", and "Clusters", with "Files" selected. A message "Select items to perform actions on them." is displayed above a file browser window. The browser lists several notebooks with ".ipynb" extensions:

- communities
- datasets
- featured
- Welcome Julia - Intro to Gadfly.ipynb
- Welcome R - demo.ipynb
- Welcome to Haskell.ipynb
- Welcome to Python.ipynb
- Welcome to Spark with Python.ipynb
- Welcome to Spark with Scala.ipynb

At the bottom right of the browser window are buttons for "Upload", "New", and a refresh icon.

[Jupyter](https://docs.jupyter.org/en/latest/running.html) (<https://docs.jupyter.org/en/latest/running.html>). (Dominio público)

Se puede entrar en alguno de los notebooks ya existentes o se puede empezar uno nuevo, pinchando en "New", arriba a la derecha. Como puedes ver, los archivos tienen la extensión .ipynb, que viene a ser un archivo de texto pero con formato JSON (JavaScript Object Notation).

The screenshot shows the Jupyter interface with a different list of notebooks:

- basic
- citations
- custom\_filter
- custom\_latex\_cell\_style
- custom\_preprocessor
- custom\_template
- hr\_cell\_style
- images
- latex\_cell\_style
- notebook\_cell\_style
- rst

The interface is identical to the one in the first screenshot, with a "New" button at the top right.

[Jupyter](https://jupyter-notebook.readthedocs.io/en/stable/notebook.html) (<https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>). (Dominio público)

La interfaz es muy sencilla y tiene como característica principal la organización en celdas de texto o de código. Las celdas se pueden ir ejecutando una a una con el botón del menú o con el atajo de teclado Mayúsculas + Enter. En el ejemplo inferior, sacado de una duda que ha planteado un usuario en stackoverflow.com, puedes ver la interfaz de un notebook en el que aparecen varias celdas de código que se han ejecutado y algunas de ellas dan como resultado un error.

The screenshot shows a Jupyter Notebook interface running on localhost. The title bar indicates the notebook is titled "Loops and Conditionals" and was last checked at 16 minutes ago. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar below has icons for file operations like Open, Save, and Run, along with a "Code" dropdown.

In [3]:

```
mylist = ["a", "b", "c", "d", "E"]
```

In [13]:

```
for item in mylist:  
    print item
```

File "<ipython-input-13-0732ac562580>", line 2  
 print item  
 ^  
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(int item)?

In [9]:

```
mylist
```

Out[9]:

```
['a', 'b', 'c', 'd', 'E']
```

In [11]:

```
[item for item in mylist]
```

Out[11]:

```
['a', 'b', 'c', 'd', 'E']
```

In [14]:

```
for item in mylist:  
    print item + " is a character."
```

File "<ipython-input-14-bc357a683763>", line 2  
 print item + " is a character."  
 ^  
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(int item + " is a character.")?

In [ ]:

[Stackoverflow user9690000 \(https://stackoverflow.com/questions/49997094/looping-in-jupyter-notebook\)](https://stackoverflow.com/questions/49997094/looping-in-jupyter-notebook) (Dominio público)

En el siguiente ejemplo, puedes ver cómo se incluyen, en las celdas de texto, Títulos, párrafos, ecuaciones, enlaces, etc (Etcétera). Además, si el resultado del código es una representación gráfica, ésta aparece también insertada en la celda de ejecución y se mostrará posteriormente cuando se abra el archivo notebook aunque no se ejecute de nuevo.



## Simple spectral analysis

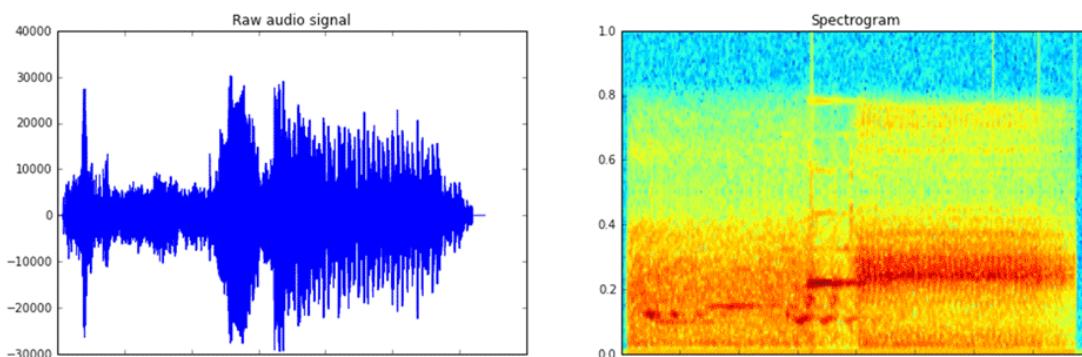
An illustration of the [Discrete Fourier Transform](#)

$$X_k = \sum_{n=0}^{N-1} x_n \exp^{-\frac{2\pi i}{N} kn} \quad k = 0, \dots, N-1$$

```
In [2]: from scipy.io import wavfile  
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin specgram routine:

```
In [5]: fig, (ax1, ax2) = plt.subplots(1,2,figsize(16,5))  
ax1.plot(x); ax1.set_title('Raw audio signal')  
ax2.specgram(x); ax2.set_title('Spectrogram');
```



[Jupyter Notebook \(<https://jupyter.org/>\)](#) (Dominio público)

## Para saber más

La web del proyecto Jupyter contiene varios recursos que te pueden ayudar a iniciarte en el uso de Jupyter Notebook. En concreto, [la documentación \(<https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>\)](#) es bastante completa y organizada. Tenla a mano, porque te será de ayuda al principio.

En el blog de DataQuest hay un [artículo en modo tutorial paso a paso \(<https://www.dataquest.io/blog/jupyter-notebook-tutorial/>\)](#) para empezar con Jupyter Notebook y probar todas sus posibilidades, que es muy recomendable.

## 1.2.- Google Colab.

Te damos la bienvenida a Colaboratory

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Compartir Editar

Índice + Código + Texto Copiar en Drive Conectar Editar

Primeros pasos Ciencia de datos Aprendizaje automático Más recursos Ejemplos destacados Sección

Te damos la bienvenida a Colab

Si ya conoces Colab, echa un vistazo a este vídeo para obtener información sobre las tablas interactivas, la vista del historial de código ejecutado y la paleta de comandos.

3 Cool Google Colab Features

¿Qué es Colaboratory?

Colab, también conocido como "Colaboratory", te permite programar y ejecutar Python en tu navegador con las siguientes ventajas:

- No requiere configuración
- Da acceso gratuito a GPUs
- Permite compartir contenido fácilmente

Colab puede facilitar tu trabajo, ya seas **estudiante, científico de datos o investigador de IA**. No te pierdas el vídeo de [Introducción a Colab](#) para [Google Colab \(\)](#) (Dominio público)

Google Colab, que inicialmente se conocía como Colaboratory, es un entorno interactivo en el que se pueden generar cuadernos ejecutables en los que se combinan celdas de código con celdas de texto enriquecido, imágenes, HTML (HyperText Markup Language), LaTeX, etc (Etcétera). Los cuadernos de Colab son cuadernos de Jupyter alojados en la nube de Google, y por tanto, incluyen muchas de sus características.

Las principales ventajas de Colab sobre Jupyter son:

- ✓ Al tratarse de un servicio en la nube, accesible vía web, no hay que instalar ni configurar nada.
- ✓ Están incorporados los principales paquetes necesarios para Ciencia de Datos, y siempre actualizados a su versión más reciente
- ✓ Se asignan recursos de memoria en disco y RAM (Random Access Memory) más que suficientes en cada sesión, a través de una máquina virtual. El entorno se desconecta pasado un tiempo de inactividad o si se superan las 12 horas. Por otro lado, una cuenta de Google solo puede tener 5 sesiones abiertas simultáneamente. No obstante, Google aplica ciertas penalizaciones de uso si se alcanzan los límites de tiempo de forma sistemática.
- ✓ Permite utilizar GPU (Graphics Processing Unit) para una computación de mayor rendimiento.

Para utilizar este entorno de programación, solo hay que entrar en la [web de colab \(\)](#) y pinchar en la opción de nuevo cuaderno para ponerse a trabajar.

[Examples](#)[Recent](#)[Google Drive](#)[GitHub](#)[Upload](#)

Filter notebooks



Title

Overview of Colaboratory Features



Markdown Guide



Charts in Colaboratory



External data: Drive, Sheets, and Cloud Storage



Getting started with BigQuery

[NEW NOTEBOOK](#)[CANCEL](#)[Google Colab \(<https://colab.research.google.com/>\)](#) (Dominio público)

También se puede importar un notebook que se haya generado en Jupyter o que esté en GitHub. Análogamente, cada cuaderno se almacenará en Drive, en una carpeta específica de Colab, aunque se puede mover a cualquier otra carpeta de Drive y también se puede descargar en formato ipynb o py.

Para aprovechar todas las ventajas de poder documentar con las celdas de texto, te recomendamos que conozcas todas las posibilidades del lenguaje Markdown que se utiliza en ellas. En [este notebook](https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Cheat_sheet_for_Google_Colab.ipynb) ([https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Cheat\\_sheet\\_for\\_Google\\_Colab.ipynb](https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Cheat_sheet_for_Google_Colab.ipynb)) tienes ejemplos de todas las etiquetas y caracteres que puedes utilizar y la forma de incluir todo tipo de elementos.

Company Name	Founder
Microsoft	Bill Gates
Facebook	Mark Zuckerberg
TNP's Hub	Tanu Nanda Prabhu

[Tanu-N-Prabhu \(https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Cheat\\_sheet\\_for\\_Google\\_Colab.ipynb#scrollTo=9P7fUbeb04tP\)](https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Cheat_sheet_for_Google_Colab.ipynb#scrollTo=9P7fUbeb04tP) (CC BY-SA (<http://creativecommons.org/licenses/?lang=es>))

Esto es un ejemplo de un notebook con consejos y cosas curiosas que aplicar a los cuadernos en colab. Puedes probarlo yendo al [propio notebook a través de su enlace en colab](https://colab.research.google.com/notebooks/basic_features_overview.ipynb) ([https://colab.research.google.com/notebooks/basic\\_features\\_overview.ipynb](https://colab.research.google.com/notebooks/basic_features_overview.ipynb)).

## Cells

A notebook is a list of cells. Cells contain either explanatory text or executable code and its output. Click a cell to select it.

### Code cells

Below is a **code cell**. Once the toolbar button indicates CONNECTED, click in the cell to select it and execute the contents in the following ways:

- ✓ Click the **Play icon** in the left gutter of the cell;
- ✓ Type **Cmd/Ctrl+Enter** to run the cell in place;
- ✓ Type **Shift+Enter** to run the cell and move focus to the next cell (adding one if none exists);  
or
- ✓ Type **Alt+Enter** to run the cell and insert a new code cell immediately below it.

There are additional options for running some or all cells in the **Runtime** menu.

In [ ]:

```
a = 10  
a
```

Out[ ]:

10

## Text cells

This is a **text cell**. You can **double-click** to edit this cell. Text cells use markdown syntax. To learn more, see our [markdown guide \(/notebooks/markdown\\_guide.ipynb\)](#).

You can also add math to text cells using [LaTeX \(http://www.latex-project.org/\)](#) to be rendered by [MathJax \(https://www.mathjax.org\)](#). Just place the statement within a pair of \\$ signs. For example  $\sqrt{3x-1} + (1+x)^2$  becomes  $\sqrt{3x-1} + (1+x)^2$ .

## Adding and moving cells

You can add new cells by using the **+ CODE** and **+ TEXT** buttons that show when you hover between cells. These buttons are also in the toolbar above the notebook where they can be used to add a cell below the currently selected cell.

You can move a cell by selecting it and clicking **Cell Up** or **Cell Down** in the top toolbar.

Consecutive cells can be selected by "lasso selection" by dragging from outside one cell and through the group. Non-adjacent cells can be selected concurrently by clicking one and then holding down Ctrl while clicking another. Similarly, using Shift instead of Ctrl will select all intermediate cells.

## Working with python

Colaboratory is built on top of [Jupyter Notebook \(https://jupyter.org/\)](#). Below are some examples of convenience functions provided.

Long running python processes can be interrupted. Run the following cell and select **Runtime -> Interrupt execution** (*hotkey: Cmd/Ctrl-M I*) to stop execution.

In [ ]:

```
import time
print("Sleeping")
time.sleep(30) # sleep for a while; interrupt me!
print("Done Sleeping")
```

Sleeping

```
-----  
KeyboardInterrupt                                Traceback (most recent call last)  
all last)  
<ipython-input-3-626f81edbca4> in <module>()  
      1 import time  
      2 print "Sleeping"  
----> 3 time.sleep(30) # sleep for a while; interrupt me!  
      4 print "Done Sleeping"
```

KeyboardInterrupt:

## System aliases

Jupyter includes shortcuts for common operations, such as ls:

In [ ]:

```
!ls /bin
```

arch@	dmesg*	ls*	pwd*	true*
awk@	dnsdomainname*	lsmod*	readlink*	umount*
basename@	domainname*	mail*	red@	uname*
bash*	echo*	mkdir*	rm*	uncompress*
bunzip2@	ed@	mknod*	rmdir*	usleep*
busybox*	egrep*	mktemp*	run-parts*	ver*
bzip2@	false*	more*	sed*	which*
cat*	fgrep*	mount*	sh@	wrapper_checkp
oints/				
chgrp*	gawk@	mountpoint*	sleep*	zcat*
chmod*	grep*	mv*	sort@	zcmp*
chown*	gunzip*	nc*	stty*	zdiff*
cp*	gzexe*	netcat@	su*	zegrep*
cpio*	gzip*	netstat*	sync*	zfgrep*
csh@	hostname*	nice@	tailf*	zforce*
cut@	igawk@	pidof@	tar*	zgrep*
date*	kill*	ping*	tcsh@	zless*
dd*	ln*	ping6*	tempfile*	zmore*
df*	login*	ps*	touch*	znew*

That !ls probably generated a large output. You can select the cell and clear the output by either:

1. Clicking on the clear output button (x) in the toolbar above the cell; or
2. Right clicking the left gutter of the output area and selecting "Clear output" from the context menu.

Execute any other process using ! with string interpolation from python variables, and note the result can be assigned to a variable:

In [ ]:

```
message = 'Colaboratory is great!'
foo = !echo -e '$message\n$message'
foo
```

Out[ ]:

```
['Colaboratory is great!', 'Colaboratory is great!']
```

## Magics

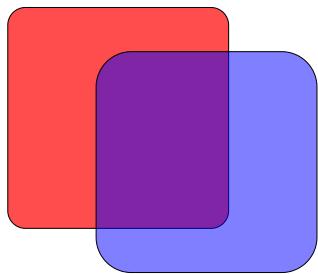
Colaboratory shares the notion of magics from Jupyter. There are shorthand annotations that change how a cell's text is executed. To learn more, see [Jupyter's magics page](http://nbviewer.jupyter.org/github/ipython/ipython/blob/1.x/examples/notebooks/Cell%20Magics.ipynb) (<http://nbviewer.jupyter.org/github/ipython/ipython/blob/1.x/examples/notebooks/Cell%20Magics.ipynb>)

In [ ]:

```
%%html
<marquee style='width: 30%; color: blue;'><b>Whee!</b></marquee>
```

In [ ]:

```
%%html
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 450 400" width="200" height="200">
  <rect x="80" y="60" width="250" height="250" rx="20" style="fill:red; stroke:black; fill-opacity:0.7" />
  <rect x="180" y="110" width="250" height="250" rx="40" style="fill:blue; stroke:black; fill-opacity:0.5;" />
</svg>
```



## Automatic completions and exploring code

Colab provides automatic completions to explore attributes of Python objects, as well as to quickly view documentation strings. As an example, first run the following cell to import the [numpy](http://www.numpy.org) (<http://www.numpy.org>) module.

In [ ]:

```
import numpy as np
```

If you now insert your cursor after `np` and press **Period( . )**, you will see the list of available completions within the `np` module. Completions can be opened again by using **Ctrl+Space**.

In [ ]:

```
np
```

If you type an open parenthesis after any function or class in the module, you will see a pop-up of its documentation string:

In [ ]:

```
np.ndarray
```

The documentation can be opened again using **Ctrl+Shift+Space** or you can view the documentation for method by mouse hovering over the method name.

When hovering over the method name the **Open in tab** link will open the documentation in a persistent pane. The **View source** link will navigate to the source code for the method.

## Exception Formatting

Exceptions are formatted nicely in Colab outputs:

In [ ]:

```
x = 1  
y = 4  
z = y/(1-x)
```

```
-----  
-----  
ZeroDivisionError                                     Traceback (most recent c  
all last)  
<ipython-input-14-dc39888fd1d2> in <module>()  
      1 x = 1  
      2 y = 4  
----> 3 z = y/(1-x)  
  
ZeroDivisionError: integer division or modulo by zero
```

## Rich, interactive outputs

Until now all of the generated outputs have been text, but they can be more interesting, like the chart below.

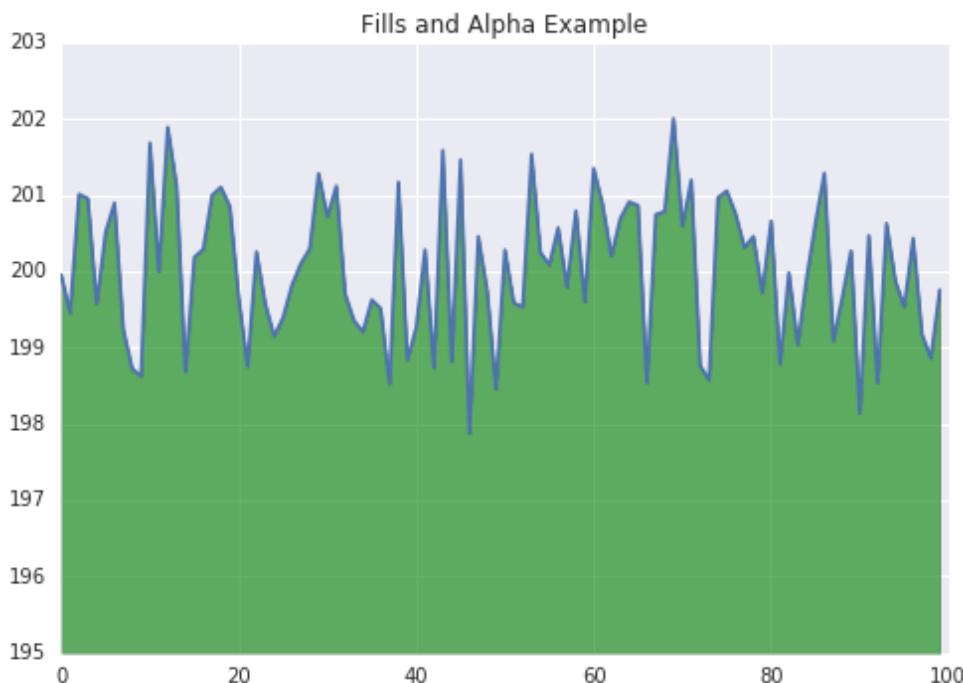
In [ ]:

```
import numpy as np
from matplotlib import pyplot as plt

ys = 200 + np.random.randn(100)
x = [x for x in range(len(ys))]

plt.plot(x, ys, '-.')
plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)

plt.title("Fills and Alpha Example")
plt.show()
```



## Integration with Drive

Colaboratory is integrated with Google Drive. It allows you to share, comment, and collaborate on the same document with multiple people:

- ✓ The **SHARE** button (top-right of the toolbar) allows you to share the notebook and control permissions set on it.
- ✓ **File->Make a Copy** creates a copy of the notebook in Drive.
- ✓ **File->Save** saves the File to Drive. **File->Save and checkpoint** pins the version so it doesn't get deleted from the revision history.
- ✓ **File->Revision history** shows the notebook's revision history.

## Commenting on a cell

You can comment on a Colaboratory notebook like you would on a Google Document. Comments are attached to cells, and are displayed next to the cell they refer to. If you have **comment-only** permissions, you will see a comment button on the top right of the cell when you hover over it.

If you have edit or comment permissions you can comment on a cell in one of three ways:

1. Select a cell and click the comment button in the toolbar above the top-right corner of the cell.
2. Right click a text cell and select **Add a comment** from the context menu.
3. Use the shortcut **Ctrl+Shift+M** to add a comment to the currently selected cell.

You can resolve and reply to comments, and you can target comments to specific collaborators by typing `+[email address]` (e.g., `+user@domain.com`). Addressed collaborators will be emailed.

The Comment button in the top-right corner of the page shows all comments attached to the notebook.

## Para saber más

¿Quieres saber qué volumen de trabajo admite Colab? En [este artículo](https://medium.com/the-agile-crafters-notebook-spanish/17-d%C3%ADas-ejecutando-google-colab-158c44080bf) (<https://medium.com/the-agile-crafters-notebook-spanish/17-d%C3%ADas-ejecutando-google-colab-158c44080bf>), un estudiante lo ha puesto a prueba ejecutando a lo largo de 17 días, cerca de unos 20.000 experimentos. Es muy interesante conocer los problemas que se ha encontrado y algunos consejos valiosos para sacarle partido.

## Autoevaluación

¿Cuál es la principal diferencia entre Jupyter Notebook y Google Colab?

- Jupyter Notebook permite utilizar celdas de texto con lenguaje markdown y Google Colab no

- Jupyter Notebook solo soporta R mientras que Google Colab solo soporta Python

- Jupyter Notebook se instala y ejecuta en local, mientras que Google Colab se ejecuta en la nube y se accede a través de internet.

[Mostrar retroalimentación](#)

# Solución

1. Incorrecto
2. Incorrecto
3. Correcto

## 1.3.- Binder.



Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

New to Binder? Get started with a [Zero-to-Binder tutorial](#) in Julia, Python, or R.

Build and launch a repository

GitHub repository name or URL

GitHub  GitHub repository name or URL

Git ref (branch, tag, or commit)

HEAD

Path to a notebook file (optional)

Path to a notebook file (optional)

Copy the URL below and share your Binder with others:

Fill in the fields to see a URL for sharing your Binder.

Expand to see the text below, paste it into your README to show a binder badge:

[mybinder.org\\_](https://mybinder.org/) (<https://mybinder.org/>) (Dominio público)

[Binder](https://mybinder.org/) (<https://mybinder.org/>) es una web app que proporciona un servidor temporal para ejecutar cuadernos de Jupyter alojados en un repositorio de GitHub. Es gratuito y muy sencillo de utilizar. Es un desarrollo de código abierto, mantenido por la comunidad con el apoyo de empresas como Google, OVH o el Turing Institute.

Solo es necesario dar acceso al repositorio donde están los notebooks, o especificar la ruta de uno concreto. Se habilita un contenedor Docker con la imagen según los requerimientos y dependencias especificados en el repositorio, en la raíz, a partir del archivo requirements.txt o environment.yml. Además, si quieras ofrecer la posibilidad de que otros puedan acceder al notebook, e incluso trabajar sobre él, Binder habilita un enlace al servidor de JupyterHub en el que estarán todos los contenidos del repositorio. Esta opción es muy útil para hacer demos o mostrar desarrollos a otros y poder invitarles a hacer cambios, aportar, etc (Etcétera).

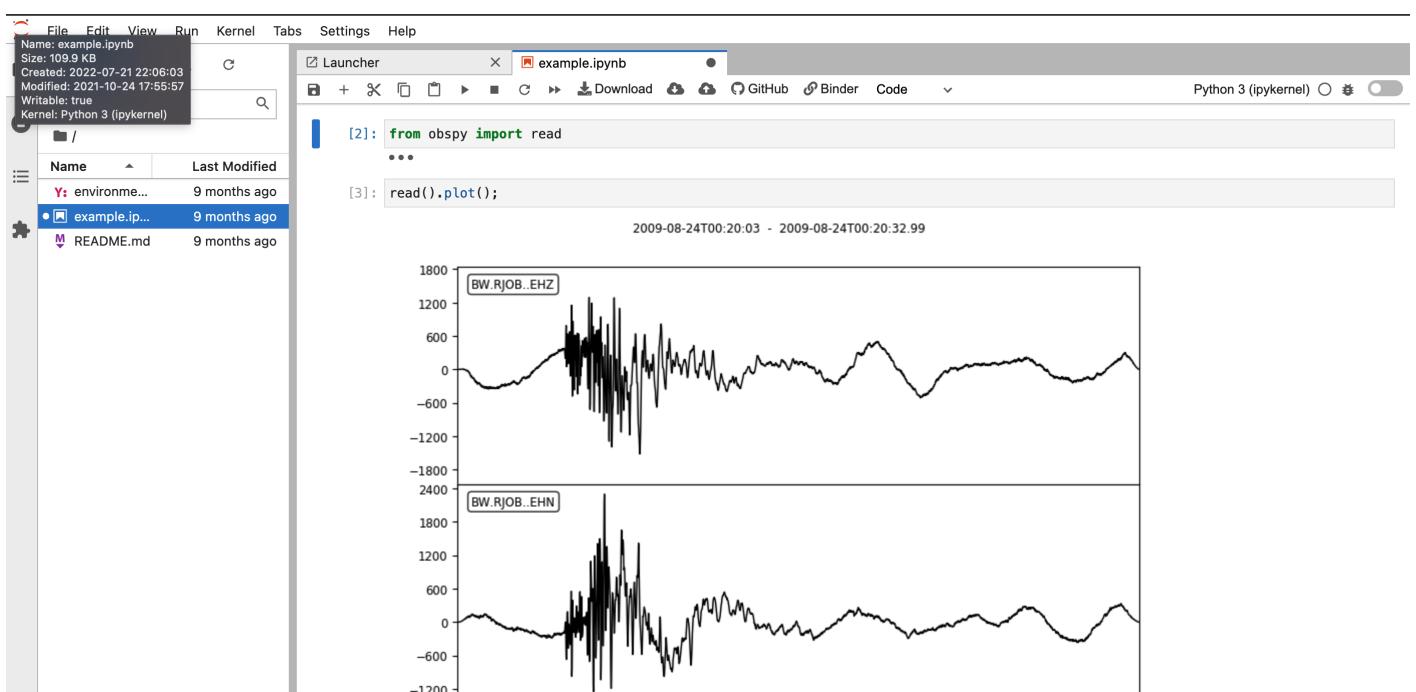
La experiencia del que recibe el enlace, es como la del ejemplo inferior, en el que hay una primera pantalla de inicialización del entorno, y después una interfaz de trabajo con el notebook ya cargado y todos los recursos del repositorio en el menú de la izquierda.



Here's a non-interactive preview on nbviewer while we start a server for you. Your binder will open automatically when it is ready.

This screenshot shows a non-interactive preview of the repository on nbviewer. It features a 'Jupyter nbviewer' header with a logo. Below it is a sidebar titled 'binder-example' containing a list of files: 'obspy's repositories', 'example.ipynb', 'README.md', and 'environment.yml'. The main area displays a Jupyter notebook interface with two code cells. Cell [2] contains the command `from obspy import read`. Cell [3] contains the command `read().plot()`. The output of cell [3] shows two vertically stacked seismogram plots. The top plot is labeled 'BW.RJOB.EHZ' and the bottom plot is labeled 'BW.RJOB.EHN'. Both plots show seismic waveforms with time axes ranging from 2009-08-24T00:20:03 to 2009-08-24T00:20:32.99. The y-axis scales are -1800 to 1800 for the top plot and -1200 to 2400 for the bottom plot.

[obspy](https://github.com/obspy/binder-example) (<https://github.com/obspy/binder-example>) (CC BY-SA (<http://creativecommons.org/licenses/?lang=es>))

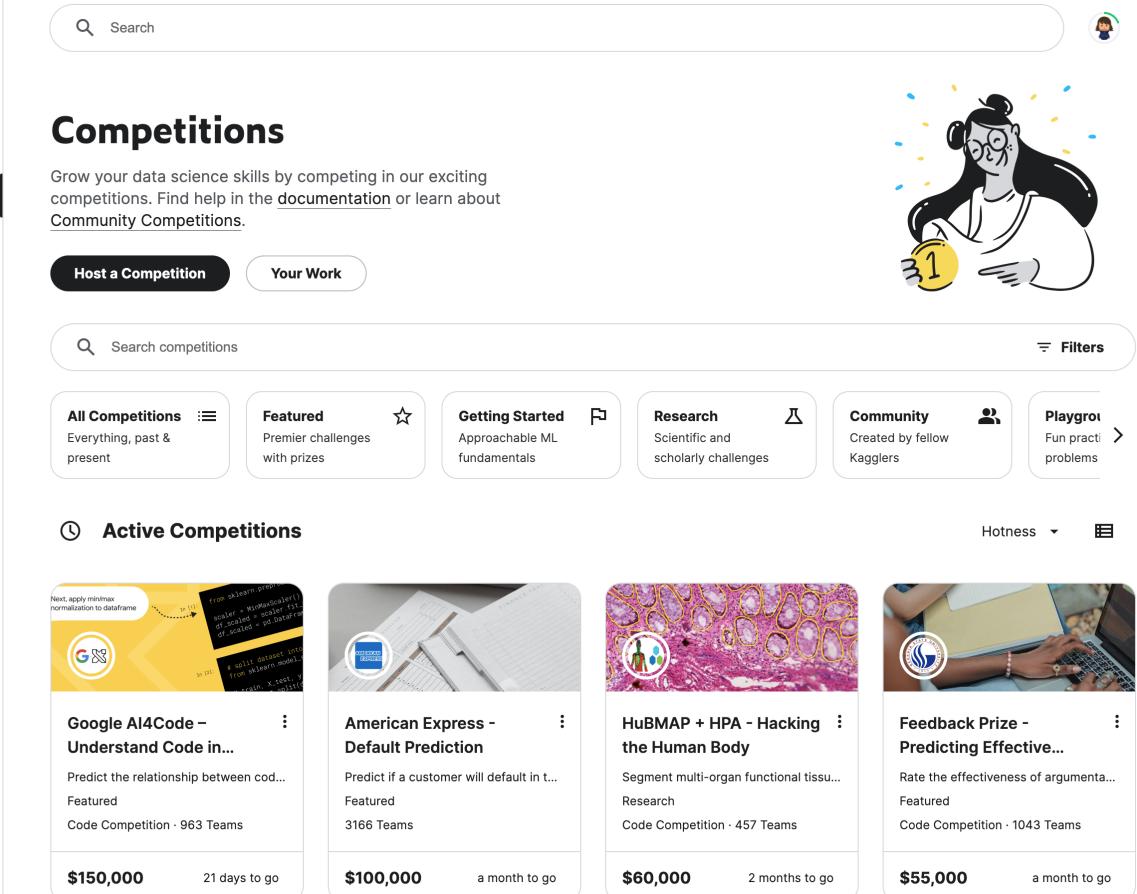


[obspy](https://hub.gke2.mybinder.org/user/obspy-binder-example-dzxfdpdz/lab/tree/example.ipynb) (<https://hub.gke2.mybinder.org/user/obspy-binder-example-dzxfdpdz/lab/tree/example.ipynb>) (CC BY-SA (<http://creativecommons.org/licenses/?lang=es>))

## Para saber más

Puedes acceder a guías de uso y tutoriales en la [documentación de Binder](#). (<https://mybinder.readthedocs.io/en/latest/>) Como hemos comentado, se trata de un proyecto de código abierto y gratuito que depende de la contribución y de toda la comunidad. Puedes contribuir sencillamente mencionándolo en la documentación de tus propios proyectos o compartiendo tu impresión en redes sociales. Así, además, seguramente conectes con otros usuarios con los que seguir aprendiendo.

## 1.4.- Kaggle.



The screenshot shows the Kaggle homepage with the main navigation bar at the top. On the left, there's a sidebar with links for Create, Home, Competitions (which is selected), Datasets, Code, Discussions, Courses, More, Your Work, Recently Viewed, and Recently Edited. The main content area has a search bar at the top right. Below it, a large section titled "Competitions" features a cartoon character holding a gold coin with the number "1". It includes a sub-section for "Active Competitions" with four cards: "Google AI4Code - Understand Code in..." (Featured, \$150,000, 21 days to go), "American Express - Default Prediction" (Featured, \$100,000, a month to go), "HuBMAP + HPA - Hacking the Human Body" (Research, \$60,000, 2 months to go), and "Feedback Prize - Predicting Effective..." (Community, \$55,000, a month to go). There are also sections for "All Competitions", "Featured", "Getting Started", "Research", "Community", and "Playground". A "Filters" button is located on the right side of the competition search bar.

La plataforma kaggle.com surgió en 2010, promovida por Anthony Goldbloom y Jeremy Howard, como una comunidad de científicos e ingenieros de datos dispuestos a participar en competiciones de aprendizaje automático a partir de retos que planteaban diferentes empresas. La idea funcionó bastante bien, y un año después, la empresa ya valía 25 millones de dólares. En 2017 fue adquirida por Google, cuando ya contaba con mas de un millón de usuarios registrados de entre 194 países.

Más allá de la fórmula de las competiciones, el verdadero valor que supone ahora mismo, tanto para la comunidad como para el propio Google, es la riqueza de datasets y propuestas de soluciones de aprendizaje automático que aglutina. Kaggle ha visto generarse en sus "kernels" modelos para el sector financiero, de la salud o para fines sociales.

Es una plataforma clave para todos los que se encuentran aprendiendo o trabajando en el sector de la ciencia de datos, y el nivel de usuario que se va alcanzando según la participación en la misma, es un indicador más a tener en cuenta en el perfil como programador de inteligencia artificial.

Te recomendamos echar un vistazo a sus cursos de machine learning y deep learning, pero, sobre todo, te recomendamos que recurras a su interfaz de programación cuando encuentres un dataset interesante y quieras trabajar con él de forma integrada, dentro de la propia plataforma. Si además forma parte de una competición, aunque no esté activa, puedes subir tu solución y recibir feedback sobre la calidad de tu propuesta según se puntúe y te posiciones en el ranking.

Imagina que encuentras un dataset interesante o que está relacionado con un estudio que quieras hacer. Desde la pantalla del dataset, si ya te has registrado, puedes pulsar sobre la opción "New notebook"

PRACHI GOPALANI · UPDATED A YEAR AGO

▲ 236 New Notebook Download (124 KiB) :

## E-Commerce Shipping Data

Product Shipment Delivered on time or not? To Meet E-Commerce Customer Demand



Data Code (86) Discussion (10) Metadata

### About Dataset

**Context**

An international e-commerce company based wants to discover key insights from their customer database. They want to use some of the most advanced machine learning techniques to study their customers. The company sells electronic products.

[kaggle.com \(https://www.kaggle.com/prachi13/customer-analytics\)](https://www.kaggle.com/prachi13/customer-analytics) (Dominio público)

**Usability** 8.24

**License** Unknown

**Expected update frequency** Never

Tras pulsar en "New Notebook", se abre la interfaz del kernel de kaggle para un nuevo proyecto de código. Cuenta con todos los paquetes necesarios de Python, y las funcionalidades de un entorno tipo notebook. Por defecto, aparece una celda con algunas importaciones y un ejemplo, pero no es necesario conservarla.

notebook44626c1f55 Draft saved

File Edit View Run Add-ons Help

Share Save Version 0

+ Data + Add data

Settings

Schedule a notebook run

Code Help

FIND CODE HELP

Find Code Help

Search for examples of how to do things

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the Kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a new submission using "Save Version".
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

+ Code + Markdown

[kaggle.com \(https://www.kaggle.com/\)](https://www.kaggle.com/) (Dominio público)

Al haber inicializado el notebook a partir de un dataset, lo tenemos vinculado desde la sección de Data, y la ruta será siempre la misma.

The screenshot shows a Jupyter Notebook environment. On the left, there's a sidebar with various icons. The main area has a code cell at the top containing Python code for importing libraries and loading data from a CSV file. Below the code is a data preview table with columns: ID, Warehouse\_block, Mode\_of\_Shipment, Customer\_care\_calls, Customer\_rating, Cost\_of\_the\_Product, Prior\_purchases, Product\_importance, and Gend. The data shows five rows of customer information. At the bottom of the code cell, there are buttons for 'Code' and 'Markdown'. Below the code cell is a preview of the 'Train.csv' file, showing its first few rows with the same columns. The right side of the interface includes sections for 'Data', 'Input' (listing 'customer-analytics'), 'Output (60KB / 19.5GB)', 'Settings', 'Schedule a notebook run', 'Code Help' (with a search bar), and a 'Find Code Help' section.

[kaggle.com](https://kaggle.com) (<https://kaggle.com>) (Dominio público)

A partir de aquí, se trabaja igual que en el resto de entornos notebook, con la excepción de que se va guardando de forma automática en un directorio de trabajo en el que se puede ir haciendo un control básico de versiones.

## Debes conocer

La plataforma kaggle ha sido crisol de importantes avances en el campo del aprendizaje automático y del deep learning. Geoffrey Hinton y George Dahl usaron redes neuronales profundas para ganar una competición lanzada por el laboratorio Merck, lo cual ayudó a demostrar el potencial de esta técnica. De hecho, muchos participantes la utilizarían posteriormente en otras competiciones, consolidando sus uso de forma mucho más acelerada que por las vías académicas convencionales.

Tianqi Chen utilizó la técnica XGBoost, superando con holgura los resultados de Random Forest, que había sido ampliamente utilizada en anteriores competiciones y a la que ha sustituido desde entonces como principal método utilizado en las competiciones.

## Autoevaluación

La plataforma kaggle aporta mucho valor a los científicos de datos porque ofrece datasets muy variados con sus correspondientes ejemplos de notebooks realizados por otros miembros de la comunidad.

Verdadero  Falso

Verdadero

Realmente, los datasets y los notebooks son el verdadero valor de kaggle, más allá de las competiciones.

## 2.- IDEs para proyectos.

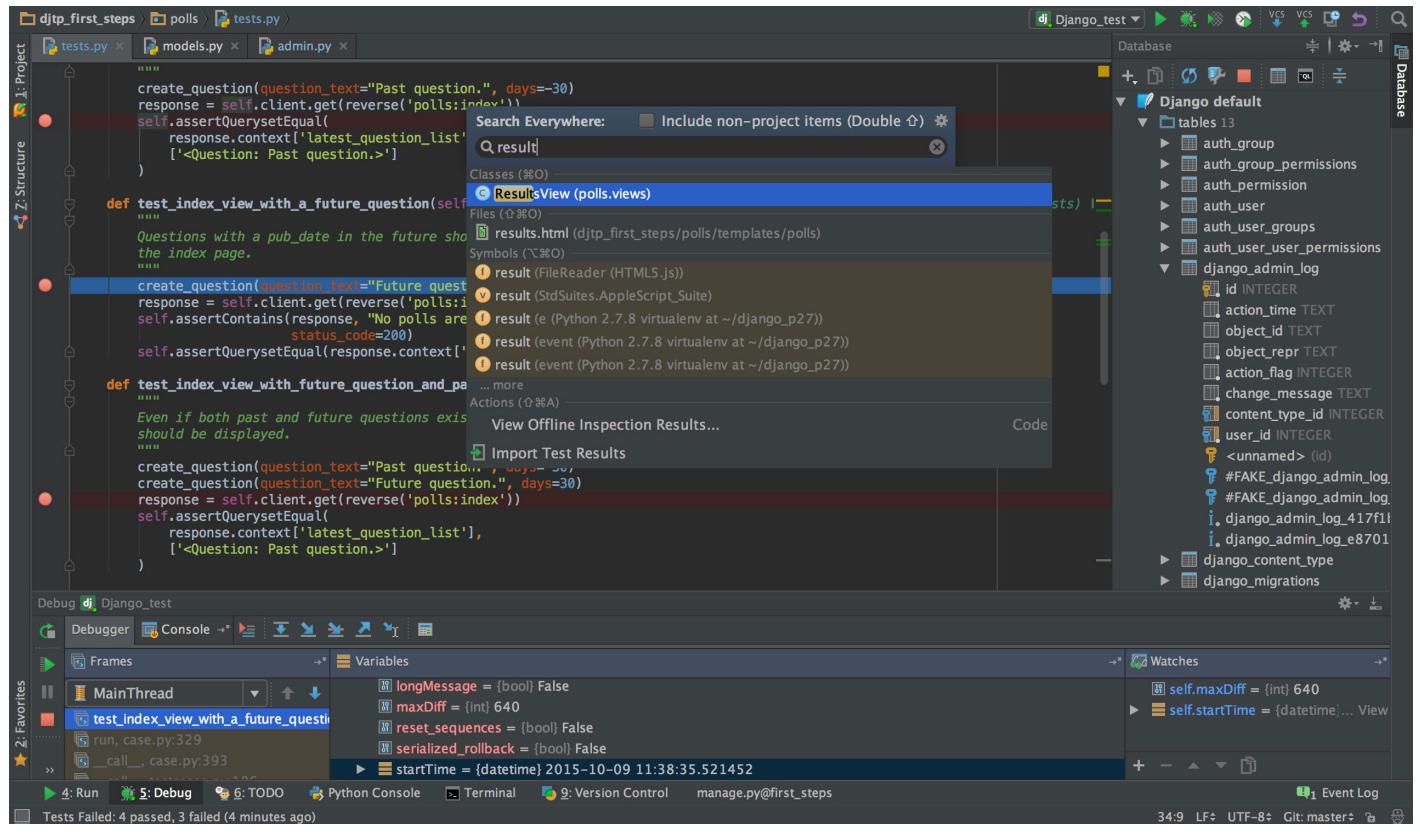
### Caso práctico

Una vez que Lorena ha estado probando las opciones que dan los distintos notebooks, ha decidido documentarse bien para cuando tenga que implementar los modelos que entrenen en una aplicación final. La junta directiva, tras los consejos de Miguel y el impulso incansable de Elena, la gerente de la empresa, ha aceptado que se instalen sensores y dispositivos de IoT (Internet de las Cosas) en distintos puntos de control de las instalaciones de Pick&Deliver para poder monitorizar y analizar en tiempo real. Después, cuando los sistemas estén listos, se podría utilizar la robótica para automatizar ciertas tareas en el almacén, ajustando sus parámetros de funcionamiento, también en tiempo real, en función de las predicciones y alertas que de el sistema de inteligencia artificial.



[LookStudio \(\[https://www.freepik.es/foto-gratis/attractiva-mujer-independiente-riendo-posando-taza-cafe-su-lugar-trabajo-estudiante-chino-camisa-azul-trabaja-documento-campus-amiga-rubia-gafas\\\_10483947.htm#query=lookstudio&position=20&from\\\_view=search\]\(https://www.freepik.es/foto-gratis/attractiva-mujer-independiente-riendo-posando-taza-cafe-su-lugar-trabajo-estudiante-chino-camisa-azul-trabaja-documento-campus-amiga-rubia-gafas\_10483947.htm#query=lookstudio&position=20&from\_view=search\)\)](https://www.freepik.es/foto-gratis/attractiva-mujer-independiente-riendo-posando-taza-cafe-su-lugar-trabajo-estudiante-chino-camisa-azul-trabaja-documento-campus-amiga-rubia-gafas_10483947.htm#query=lookstudio&position=20&from_view=search)  
(CC BY-SA (<http://creativecommons.org/licenses/?lang=es>)))

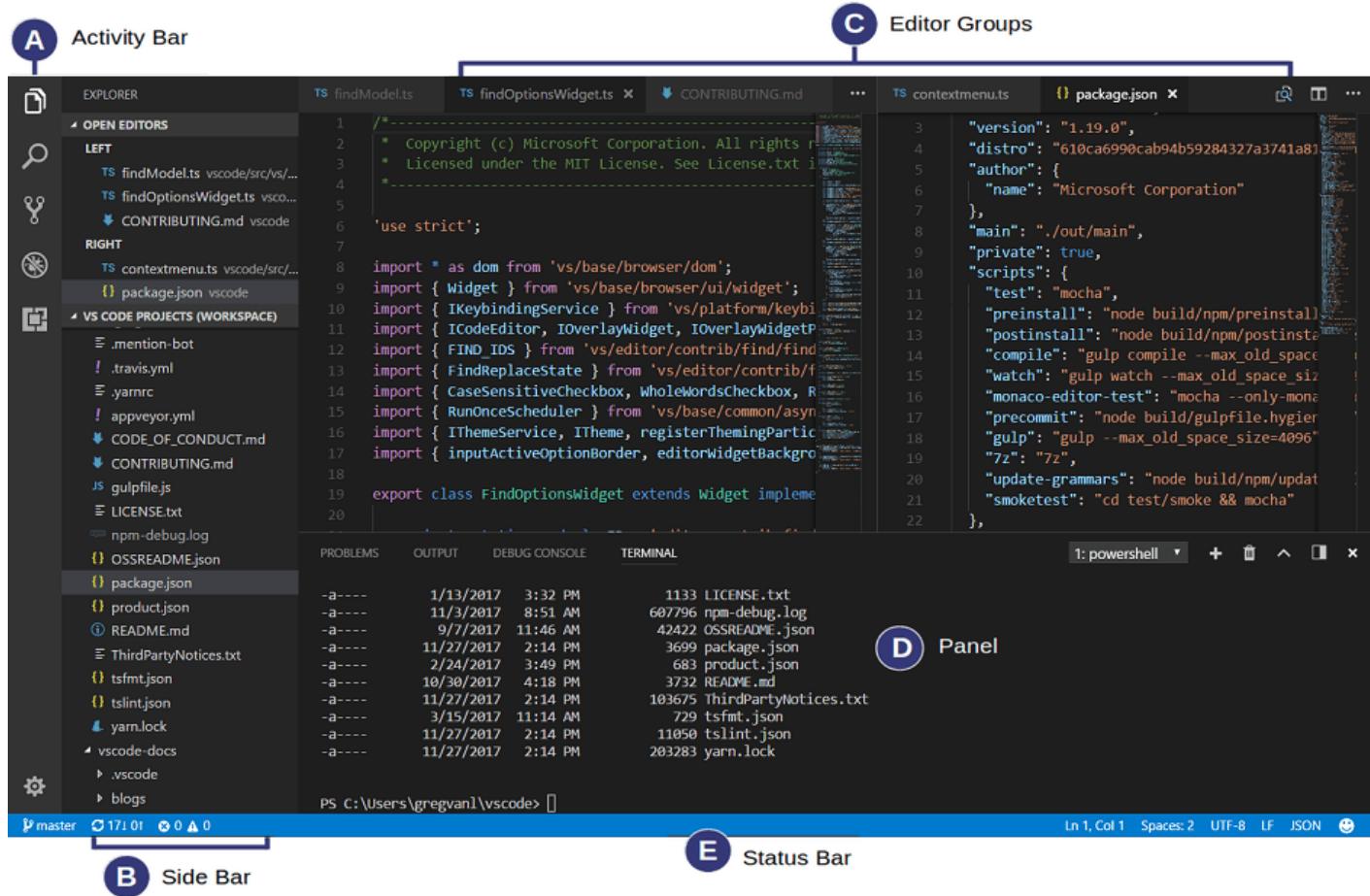
Por el momento, Lorena va a trabajar con algunos IDE (entornos de desarrollo integrados) que ya conoce, y ver qué tal se implementa en ellos todo lo que ha estado probando en los Notebooks.



Aunque el proceso de construcción y entrenamiento de un modelo se va a realizar con un entorno de notebooks con bastante probabilidad, cuando ya tenemos el modelo predictivo, su implementación en un proyecto es más común hacerla utilizando un entorno de programación más orientado a desarrollo de aplicaciones. Para ello, aunque hay una gran cantidad de opciones, vemos algunos [IDE \(Integrated Development Environment\)](#) que presentan distintos paradigmas de proyecto.

## 2.1.- Visual Studio Code.

---



[VSC \(<https://code.visualstudio.com/docs/getstarted/userinterface>\)](https://code.visualstudio.com/docs/getstarted/userinterface) (Dominio público)

Es un **IDE (Integrated Development Environment)**, propiedad de Microsoft, que ha ido ganando aceptación en los últimos años. A pesar de que la descarga oficial está bajo software privativo con opciones de suscripción y servicios extra por parte de Microsoft, está disponible de forma gratuita y en código abierto.

En realidad, está especialmente enfocado en proyectos de .NET y otros lenguajes auspiciados por Microsoft, pero la comunidad de Python lo ha ido adoptando de forma generalizada, y existen un buen número de plugins para Python, integrados en marketplace, que mejoran mucho la experiencia programando en Python.

Es un IDE muy recomendable si se va a trabajar de forma colaborativa, vinculando el IDE a un repositorio de GitHub o GitLab. Implementa varias funcionalidades de Git y sets de herramientas o extensiones para conectar con proyectos en Azure, AWS (Amazon Web Services) o GCP (Google Cloud Platform) para poder desplegar en la nube. La suscripción de pago cuenta con varias herramientas extra a este respecto.

Antes de llevar un proyecto a producción es necesario hacer una batería de test, depurar el código, y con cierta frecuencia, refactorizar para pulir algunas cosas que hayan quedado como soluciones temporales. Este tipo de tareas se llevan a cabo mucho mejor en un IDE.

En las últimas actualizaciones de software, se han incluido extensiones para trabajar con Jupyter Notebooks en VSC (Visual Studio Code), pero es importante tener en cuenta que sigue siendo necesario instalar los paquetes y librerías que se quieran usar. [Aquí tienes un tutorial \(<https://code.visualstudio.com/docs/datascience/data-science-tutorial>\)](https://code.visualstudio.com/docs/datascience/data-science-tutorial) para probar este modo, pero te recomendamos que lo uses cuando ya hayas trabajado durante un tiempo con Jupyter o con Colab, para que te cueste menos identificar los errores debidos a que falta algún paquete frente a los errores típicos de código. En el ejemplo inferior puedes ver un archivo notebook en VSC, con el recurso extra para gestionar el dataset.

```
import pandas as pd
import numpy as np
data = pd.read_csv('data.csv')
```

Name	Type	Size	Value
data	DataFrame	(1309, 14)	pclass survived name \ 0 1 1 <...> ? 1308 ? ? [1]

## Para saber más

Te recomendamos echar un vistazo a la documentación de Visual Studio Code en su sección de [Iniciación](https://code.visualstudio.com/learn) (<https://code.visualstudio.com/learn>) y en la sección de [Guía de usuario](https://code.visualstudio.com/docs/editor/codebasics) (<https://code.visualstudio.com/docs/editor/codebasics>). También es especialmente recomendable que sigas las [instrucciones para dejar VSC a punto para usarlo con Python](https://code.visualstudio.com/docs/languages/python) (<https://code.visualstudio.com/docs/languages/python>).

## Autoevaluación

¿Cuándo se debe utilizar un IDE en un proyecto de inteligencia artificial?

- Cuando ya hemos generado un modelo predictivo que tenga buenas métricas, a la hora de integrarlo en un proyecto.
- Cuando estamos decidiendo qué técnica de aprendizaje utilizar y necesitamos ver y comparar varias
- Cuando se está aprendiendo sobre aprendizaje automático

Opción correcta

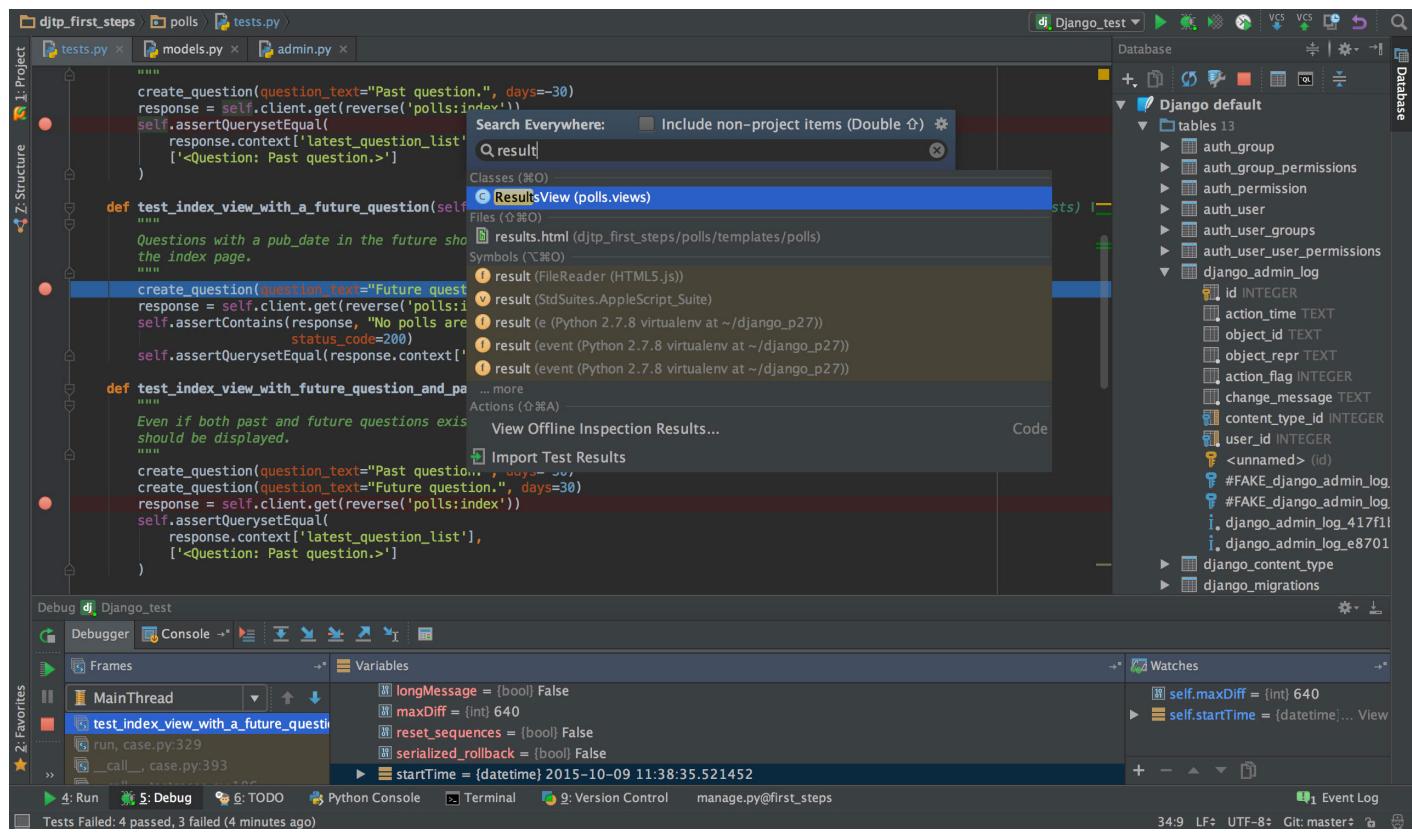
Para hacer este tipo de ejecuciones de forma ágil y cómoda, es mejor utilizar un notebook

El formato notebook es mejor para la etapa de aprendizaje, pues permite tener anotaciones y representaciones integradas entre los distintos resultados de la ejecución de celdas de código.

## Solución

1. Opción correcta
2. Incorrecto
3. Incorrecto

## 2.2.- Pycharm.



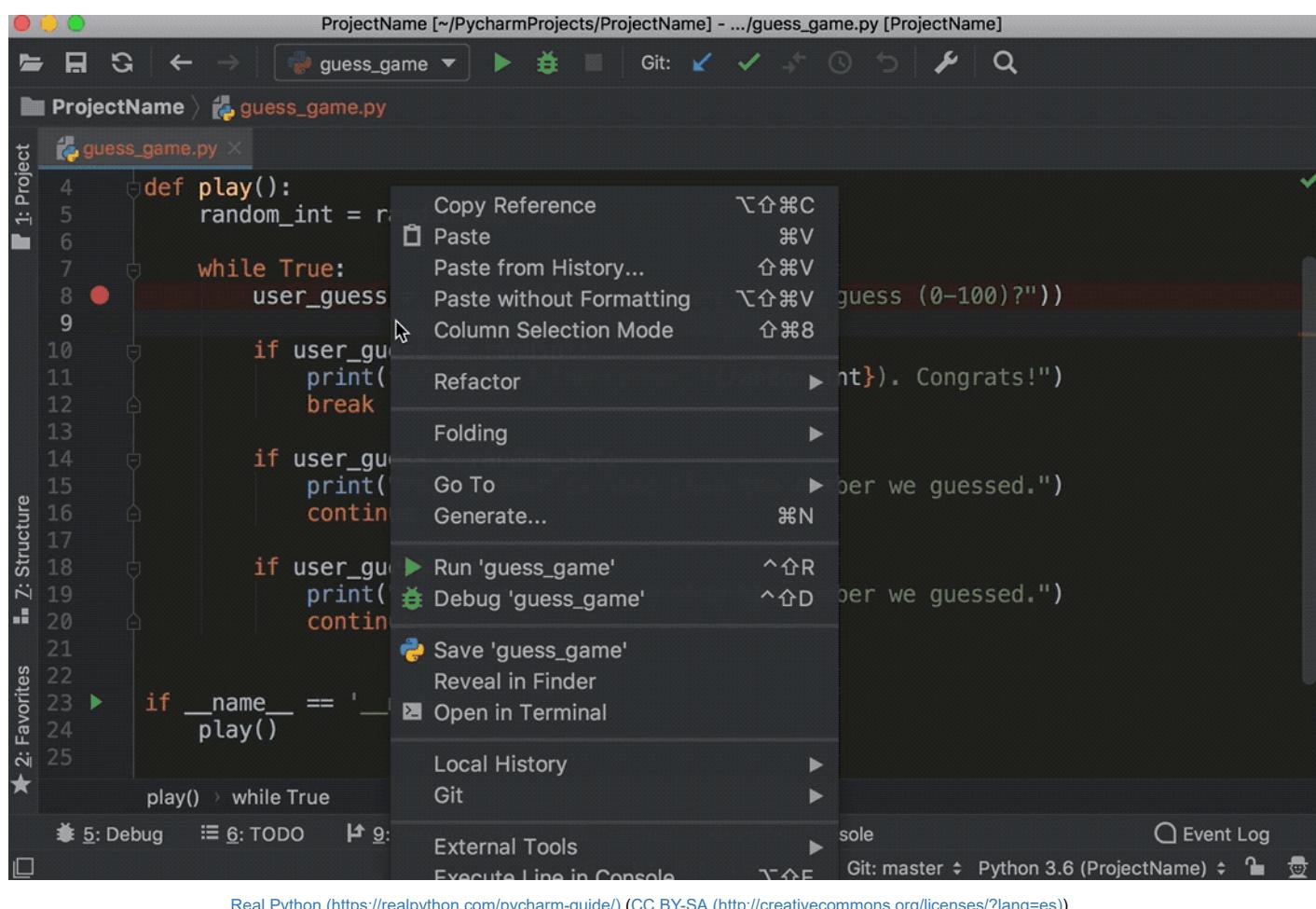
[Pycharm \(https://www.jetbrains.com/es-es/pycharm/\)](https://www.jetbrains.com/es-es/pycharm/) (Dominio público)

Pycharm es un IDE (Integrated Development Environment) enfocado en Python, desarrollado por la empresa JetBrains que incluye varias funcionalidades muy específicas de ese lenguaje. Cuenta con una versión denominada Community, que es gratuita y se publica bajo la Licencia Apache, que incluye:

- ✓ Un entorno virtual para cada proyecto
- ✓ Tiene un editor inteligente que ayuda a detectar errores, verificar sintaxis, completar código y muestra de documentación automática
- ✓ Incluye depurador gráfico, ejecución de tests e inspector de código.

✓ Como todos los IDE, tiene una terminal incorporada.

Al igual que VSC (Visual Studio Code), nos va a permitir realizar las acciones necesarias para control de versiones, desde una interfaz gráfica, para hacer commits, push y pull a repositorio. También tienes disponibles Toolkits o plugins para conectar con AWS (Amazon Web Services), GCP (Google Cloud Platform) o Azure y trabajar con despliegues en dichas nubes.

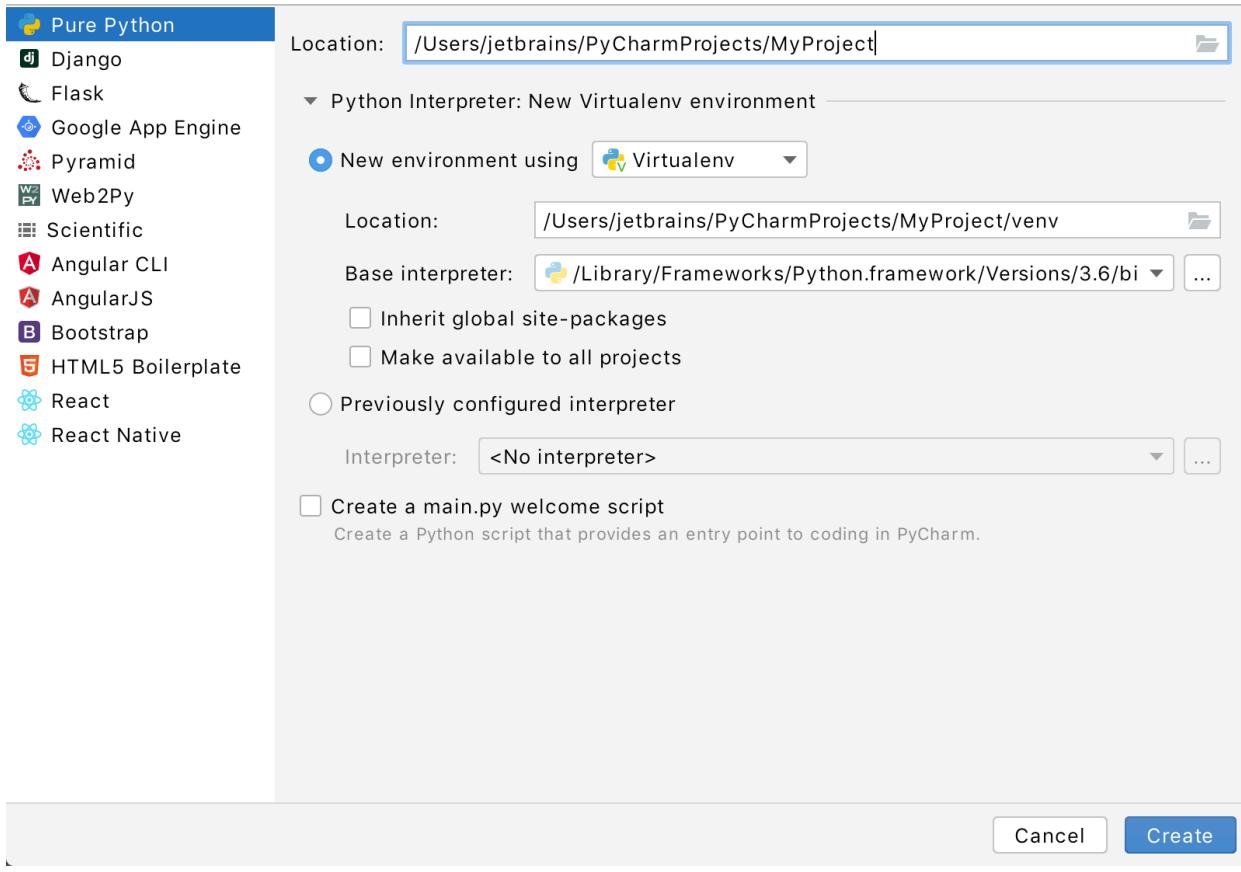


Como puedes ver en el ejemplo de la parte superior, el proceso de depuración del código, está mejor organizado en un IDE, como en este caso, en Pycharm. De hecho, este ejemplo pertenece a una [guía sobre Pycharm](#) (<https://realpython.com/pycharm-guide/>) muy útil en del blog Real Python, que te recomendamos especialmente.

En su versión profesional (suscripción de pago), incluye funcionalidades extra orientadas a la ciencia de datos, como representaciones gráficas o la integración de Conda y cuadernos de Jupyter.

## Para saber más

Pycharm es un IDE muy potente y que utiliza más de un tercio de los desarrolladores de Python. Merece la pena que descubras todas sus ventaja y aprendas a usarlo si crees que vas a pasar muchas horas poniendo en pie proyectos en este lenguaje. La sección de aprendizaje de la propia web de Pycharm contiene varios recursos que te pueden ayudar a empezar, como la [guía de iniciación](#) (<https://www.jetbrains.com/help/pycharm/quick-start-guide.html>) o un [tutorial para crear una primera app](#) (<https://www.jetbrains.com/help/pycharm/creating-and-running-your-first-python-project.html>).



[Pycharm \(<https://www.jetbrains.com/help/pycharm/creating-and-running-your-first-python-project.html#creating-simple-project>\)](https://www.jetbrains.com/help/pycharm/creating-and-running-your-first-python-project.html#creating-simple-project) (Dominio público)

## 2.3.- Replit.

The screenshot shows the Replit IDE interface. On the left is the file explorer with files like build, src (containing pages/index.html, app.jsx, App.css, main.jsx), tests, README.md, .gitignore, and vite.config.js. The central area has two tabs: app.jsx and App.css. The app.jsx tab contains the following code:

```
1 import React from 'react';
2 import './App.css';
3 import { BrowserRouter as Tabish } from 'react-router-dom';
4 import Intro from './pages/intro';
5 import Contribute from './pages/contribute';
6 import FeatureReq from './pages/feature';
7 import Long from './pages/long';
8 import Idea from './pages/idea';
9 import Video from './pages/video';
10
11 function App() {
12   return (
13     <Router>
14       <Switch>
15         <main class="pt-2 pl-5 pr-5 pb-10 sm:pt-10">
16           <div class="mt-3 bg-white py-8 px-10 rounded-md shadow-lg transform max-w-xl mx-auto">
17             <Route exact path="/">
18               <Intro />
19             </Route>
20             <Route path="/contribute">
21               <Contribute />
22             </Route>
23             <Route path="/long">
24               <Long />
25             </Route>
26             <Route path="/feature">
27               <FeatureReq />
28             </Route>
29             <Route path="/idea">
30               <Idea />
31             </Route>
32             <Route path="/idea2">
33               <Idea2 />
34             </Route>
35           </div>
36         </main>
37       </Switch>
38     </Router>
39   );
40 }
41
42 export default App;
```

The right panel shows the console output:

```
node v12.16.1
Repl.it: Updating package configuration
--> npm init -y
Wrote to
/home/runner/DependentSelfishUpgrades/package.json

{
  "name": "DependentSelfishUpgrades",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\"Error: no test specified\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

--> npm install auth
npm WARN deprecated jasmine-node@1.0.28: jasmine-node 1.x & 2.x are deprecated, with known vulnerability in jasmine-growl-reporter pre-2.0.0
npm WARN deprecated coffee-script@1.12.7: CoffeeScript on NPM has moved to "coffeescript" (no hyphen)
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN DependentSelfishUpgrades@1.0.0 No description
npm WARN DependentSelfishUpgrades@1.0.0 No repository
```

[replit.com\\_](https://replit.com/) (<https://replit.com/>) (Dominio público)

Replit es un IDE (Integrated Development Environment) online que ejecuta el código desde el propio navegador. La principal ventaja es lo sencillo que es empezar un proyecto y lanzarlo rápidamente. sus características a destacar son:

- ✓ Soporta muchos lenguajes, más de 50, entre los que está Python.
- ✓ Es tan versátil que hasta se puede utilizar desde un móvil.
- ✓ Permite trabajo colaborativo sobre el mismo proyecto, en tiempo real, aunque es especialmente recomendable hacer la integración con GitHub para una mayor seguridad.
- ✓ Las opciones de despliegue son muy simples pero eficaces. Se puede dar acceso al Repl en diferentes formatos y desde el mismo momento de crearlo.
- ✓ Cuenta con un catálogo de API (Application Programming Interfaces)s y plugins muy variado que enriquecen el proyecto adaptándolo a cualquier necesidad extra.

# Everything you need to launch.

**Host.** Everytime you create a repl, we automatically host it on our servers.

**Deploy.** Your apps are always live with an instantly sharable link. Creating a repl is deploying.

**Run.** Instantly run and share your apps, with no deployment process.



[Replit.com \(https://replit.com/site/hosting\)](https://replit.com/site/hosting) (Dominio público)

Replit cuenta con un escritorio principal donde puedes gestionar tus proyectos de una forma muy visual

The screenshot shows the Replit web interface. On the left is a sidebar with links: + Create, Upgrade, Home (which is selected), Templates, My Repls, Community, Learn, Teams, Curriculum, and a bottom section with links to Blog, About, Careers, Pricing, Discord, and Terms. The main area has a 'Create' section with '+', Python, and HTML, CSS, JS buttons, and a 'See all Templates' link. Below it is a 'Recent' section showing three projects: 'mascota\_virtual' (Python, 5 months ago), 'ejemplo\_denuncia' (HTML, 1 year ago), and 'Diagnosis\_tree' (Python, 1 year ago). There's also a 'See all Repls' link. Underneath is a 'GitHub repos' section with a 'Connect GitHub' button. The final section is 'Trending' with three projects: 'Illegal-Game' (PoopyLulu), 'Parkour Game' (GoodVessel92551), and 'ElectroEscape' (Anitor).

[Replit \(https://replit.com/\)](https://replit.com/) (Dominio público)

Y la interfaz de trabajo es sencilla y práctica, con las herramientas y funcionalidades más necesarias

The screenshot shows the Replit IDE interface. On the left, the 'Files' sidebar lists 'main.py' and 'predict\_diabetes.py'. The main workspace displays the code for 'predict\_diabetes.py', which contains a series of nested if statements for predicting diabetes based on glucose, bmi, and insulin levels. The 'Run' button at the top right is highlighted in green. To the right, the 'Console' tab is active, showing the output: 'Prediction of diabetes diagnosis: False'. Below the code editor, a note credits Carmen Bartolomé ([https://replit.com/@CarmenBartolome/Diagnosistree#predict\\_diabetes.py](https://replit.com/@CarmenBartolome/Diagnosistree#predict_diabetes.py)) and provides a Creative Commons license link.

## Para saber más

Si tienes curiosidad sobre cómo utilizar modelos de aprendizaje automático en Replit, una buena orientación es empezar por la carga y visualización de datos que te proponen en este [tutorial de Data Science en Replit](https://docs.replit.com/tutorials/data-science-and-visualisation-with-repl-it) (<https://docs.replit.com/tutorials/data-science-and-visualisation-with-repl-it>). Por otro lado, a partir del repl compartido por el usuario @GarethDwyer1, Replit ofrece este [tutorial de Introducción al Machine Learning](https://ritza.co/showcase/repl.it/introduction-to-machine-learning-with-python-and-repl-it.html) (<https://ritza.co/showcase/repl.it/introduction-to-machine-learning-with-python-and-repl-it.html>), que forma parte de un libro digital lleno de ejemplos muy útiles que puedes consultar aquí: [Code with Replit](https://www.codewithreplit.com/) (<https://www.codewithreplit.com/>).

Por otro lado, el usuario @ArtemLaptiev1 [ha publicado un artículo](https://replit.com/talk/learn/Building-AI-Neural-Networks-for-beginners/8156) (<https://replit.com/talk/learn/Building-AI-Neural-Networks-for-beginners/8156>) con un guión general para poder trabajar con redes neuronales en el IDE. Como puedes ver, la comunidad en Replit es bastante activa y muy participativa.

## Recomendación

Aunque la versión gratuita de replit es suficiente para descubrirlo y sacarle partido, todos los repls deben ser públicos. Si vas a necesitar tener repls privados, a parte de un mayor almacenamiento, te recomendamos que se solicite el [programa educativo](https://replit.com/site/teams-for-education) (<https://replit.com/site/teams-for-education>) desde tu centro. Tiene también bastantes ventajas para los profesores y es gratuito.

## Autoevaluación

La principal desventaja de Replit es que es necesario descargar la última versión del software e instalarlo en una partición del disco duro de tu ordenador

Verdadero  Falso

Falso

Replit es una aplicación online, que se usa en el navegador. No hay que instalar nada.

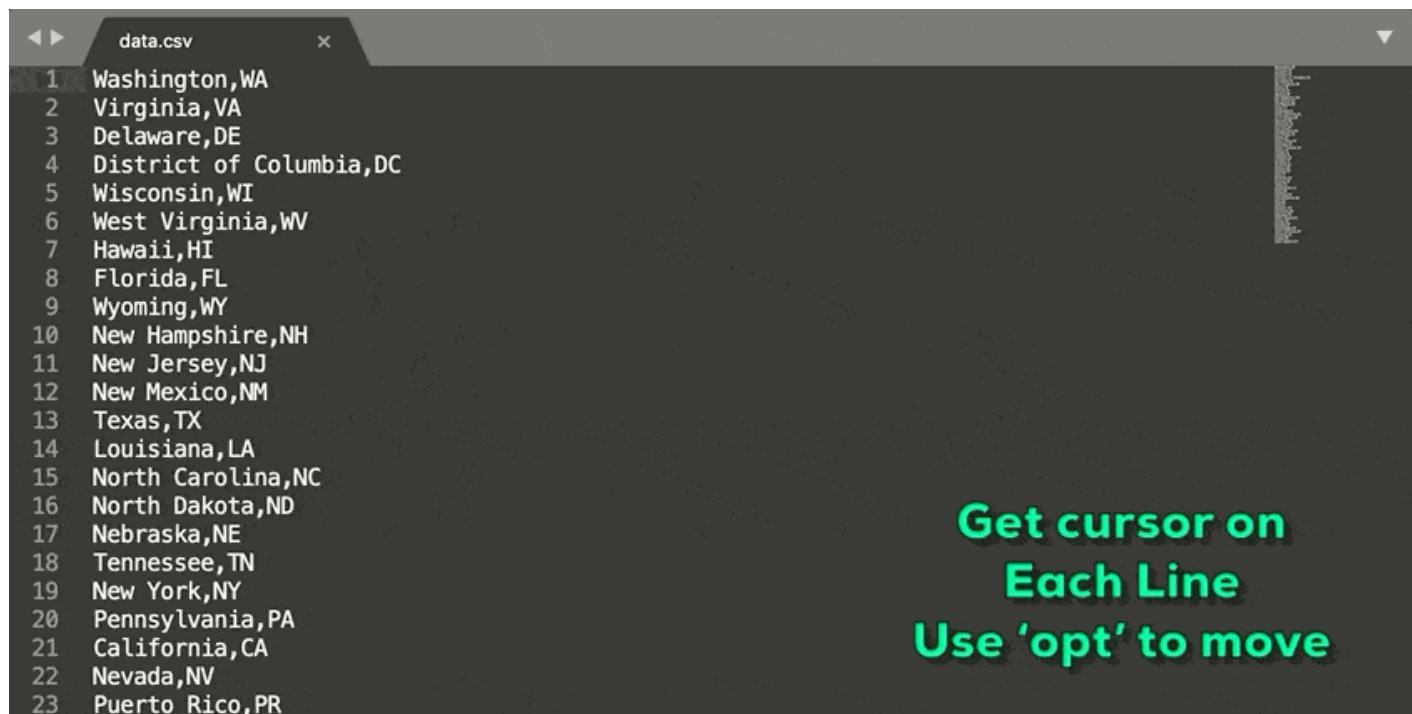
## 2.4.- Sublime Text.

Sublime Text (<https://www.sublimetext.com/>) es un editor de texto y código fuente desarrollado en C++. Es de las opciones más sencillas que vas a ver en cuanto a IDE (Integrated Development Environment), aunque admite plugins de terceros que enriquecen la experiencia. Algunas de las funcionalidades que merece la pena destacar son:

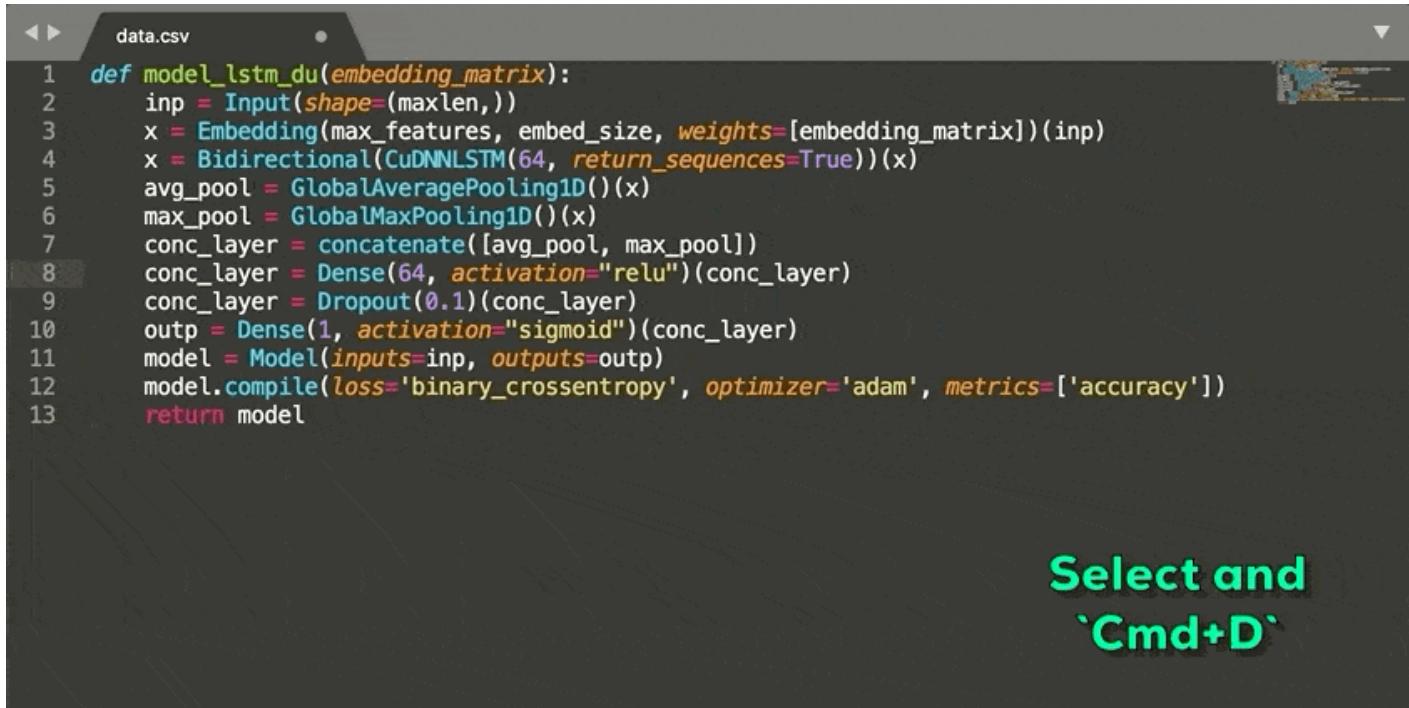
- ✓ Minimap con una previsualización de la estructura del código para moverse por éste
- ✓ Tareas múltiples de selección y cursor.
- ✓ Autocompletado y marcado de llaves.
- ✓ Coloreado de sintaxis
- ✓ Soporte de Snippets y Plugins

Pero el papel que juega en el ámbito de la ciencia de datos es más como herramienta de apoyo o complementaria, que como el entorno para el proyecto. Es como una navaja suiza para el científico de datos o el Mlops que está en pleno desarrollo y pruebas varias. Es un entorno en el que poder visualizar de forma rápida y editar código, ejecutarlo y probar. Cuando llega a ti una nueva idea, o truco y quieras echar un vistazo rápido, Sublime se abre rápidamente y la carga del archivo es sencilla, sin habilitar nada innecesario.

Además, como también admite las tareas de manipulación efectiva de texto que tienen otros IDEs más complejos a través de atajos de teclado, muchos desarrolladores lo prefieren para generar ciertas estructuras de datos o manipular fragmentos de forma rápida sin tocar el código del proyecto en VSC (Visual Studio Code) o Pycharm. Aquí tienes un par de gifs como ejemplo de este tipo de tareas:



@mlwhiz (<https://towardsdatascience.com/why-sublime-text-for-data-science-is-hotter-than-jennifer-lawrence-651afc54d67>) ([CC BY-SA \(<http://creativecommons.org/licenses/?lang=es>\)](http://creativecommons.org/licenses/?lang=es))



The screenshot shows a Sublime Text window with a dark theme. A file named 'data.csv' is open. The code is a Python script for creating a neural network model:

```
1 def model_lstm_du(embedding_matrix):
2     inp = Input(shape=( maxlen, ))
3     x = Embedding(max_features, embed_size, weights=[embedding_matrix])(inp)
4     x = Bidirectional(CuDNNLSTM(64, return_sequences=True))(x)
5     avg_pool = GlobalAveragePooling1D()(x)
6     max_pool = GlobalMaxPooling1D()(x)
7     conc_layer = concatenate([avg_pool, max_pool])
8     conc_layer = Dense(64, activation="relu")(conc_layer)
9     conc_layer = Dropout(0.1)(conc_layer)
10    outp = Dense(1, activation="sigmoid")(conc_layer)
11    model = Model(inputs=inp, outputs=outp)
12    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
13    return model
```

In the bottom right corner of the window, there is a watermark-like text: "Select and 'Cmd+D'".

[@mlwhiz \(<https://creativecommons.org/licenses/?lang=es>\)](https://towardsdatascience.com/why-sublime-text-for-data-science-is-hotter-than-jennifer-lawrence-651afc54d67)

## Para saber más

Puedes consultar la documentación de Sublime Text [aquí](https://www.sublimetext.com/docs) (<https://www.sublimetext.com/docs>). También puedes seguir las novedades y consejos que da [@OdatNurd](https://twitter.com/OdatNurd) (<https://twitter.com/OdatNurd>) en su cuenta de Twitter.

También te recomendamos que eches un vistazo a esta web donde puedes consultar y descargar los [Snippets disponibles para Sublime Text](https://packagecontrol.io/packages/Python%20Data%20Science%20Snippets). (<https://packagecontrol.io/packages/Python%20Data%20Science%20Snippets>)

## Recomendación

Sublime Text ofrece una versión gratuita para probar el IDE, pero, si hay un uso continuado de la misma, el usuario verá una advertencia periódica que le solicita comprar la licencia. La posibilidad de seguir usándolo de forma gratuita no tiene caducidad, y no va más allá de mostrar la advertencia citada. Aun así, si parte de tu trabajo va a depender de la integridad y continuidad de este IDE, te recomendamos que adquieras la licencia o que no utilices este IDE desde el principio.