

## **Unidad 2: Herramientas de desarrollo de Inteligencia Artificial**

Módulo: “Programación de Inteligencia Artificial”

Curso de Especialización en Inteligencia Artificial  
y Big Data

IES Serra Perenxisa

# Herramientas de desarrollo de Inteligencia Artificial

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.

También veremos productos SaaS disponibles en el mercado relacionados con el desarrollo de inteligencia artificial, especialmente Google Cloud Platform (GCP), Amazon Web Services (AWS) y Microsoft Azure.



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

**Adaptación de los materiales formativos de FP propiedad del Ministerio de Educación y Formación Profesional.**

## Contenido

1.- Jupyter Notebook y sus entornos compatibles.....	4
1.1.- Jupyter Notebook .....	5
1.2.- Google Colab .....	8
1.3.- Binder.....	15
1.4.- Kaggle .....	16
2.- IDEs y editores para proyectos .....	19
2.1.- Visual Studio Code .....	19
2.2.- Pycharm.....	20
2.3.- Replit.....	21
2.4.- Sublime Text .....	22
2.5.- Notepad++ .....	23
3.- Aplicaciones de IA en la nube y APIs.....	24
3.1.- Google Cloud Platform .....	24
3.1.1 – VertexAI y AutoML .....	25
3.1.2 – VisionAI.....	28
3.1.3 – DialogFlow .....	29
3.2.- Amazon Web Services (AWS).....	32
3.2.1 – Amazon SageMaker.....	32
3.2.2 – Amazon Rekognition .....	33
3.2.3 – Amazon Comprehend .....	35
3.3.- Microsoft Azure .....	37
3.3.1 – Cognitive Services .....	37
3.3.2 – AzureML .....	37
3.4.- OpenAI API .....	38

# 1.- Jupyter Notebook y sus entornos compatibles



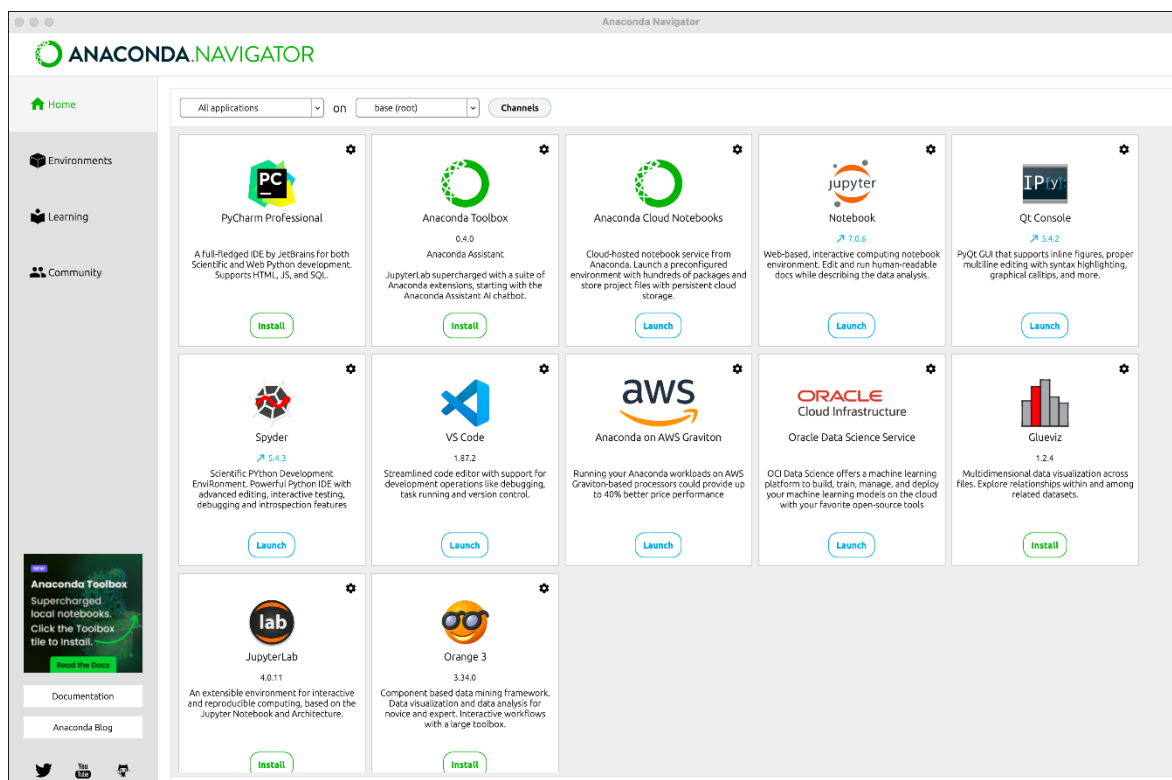
[Jupyter \(https://jupyter.org/\)](https://jupyter.org/) (CC BY-SA (<http://creativecommons.org/licenses/?lang=es>))

El Proyecto Jupyter (<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.

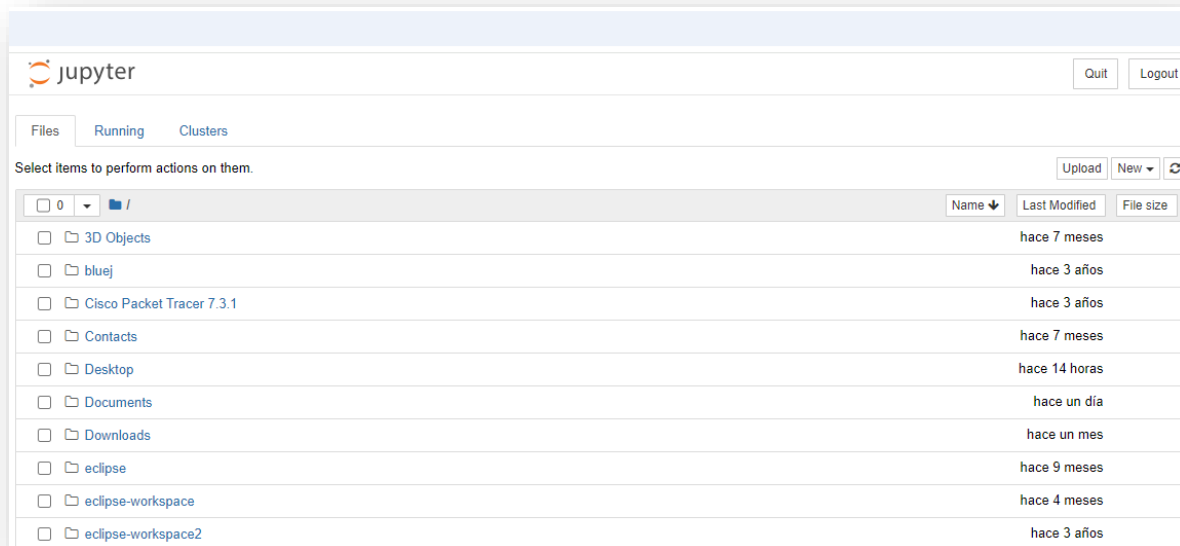
El formato Notebook es especialmente bueno para educación. Existe un buen número de cursos cuyo material didáctico está formado, íntegramente, por notebooks. Como alumno, también es muy útil ir desarrollando y anotando observaciones como en un cuaderno de papel, al tiempo que se puede ir ejecutando código 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 está implementado como interfaz front-end en algunas plataformas cloud para la aplicación del aprendizaje automático a nivel profesional.

## 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.



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.



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).

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](https://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.

```
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.")?
```

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

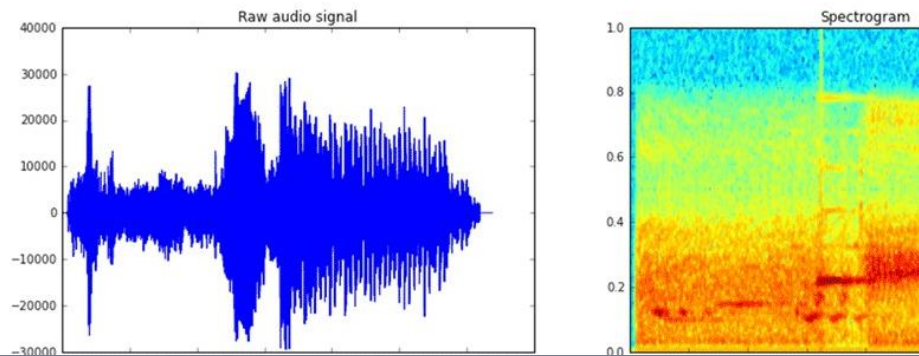
An illustration of the [Discrete Fourier Transform](#)

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

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

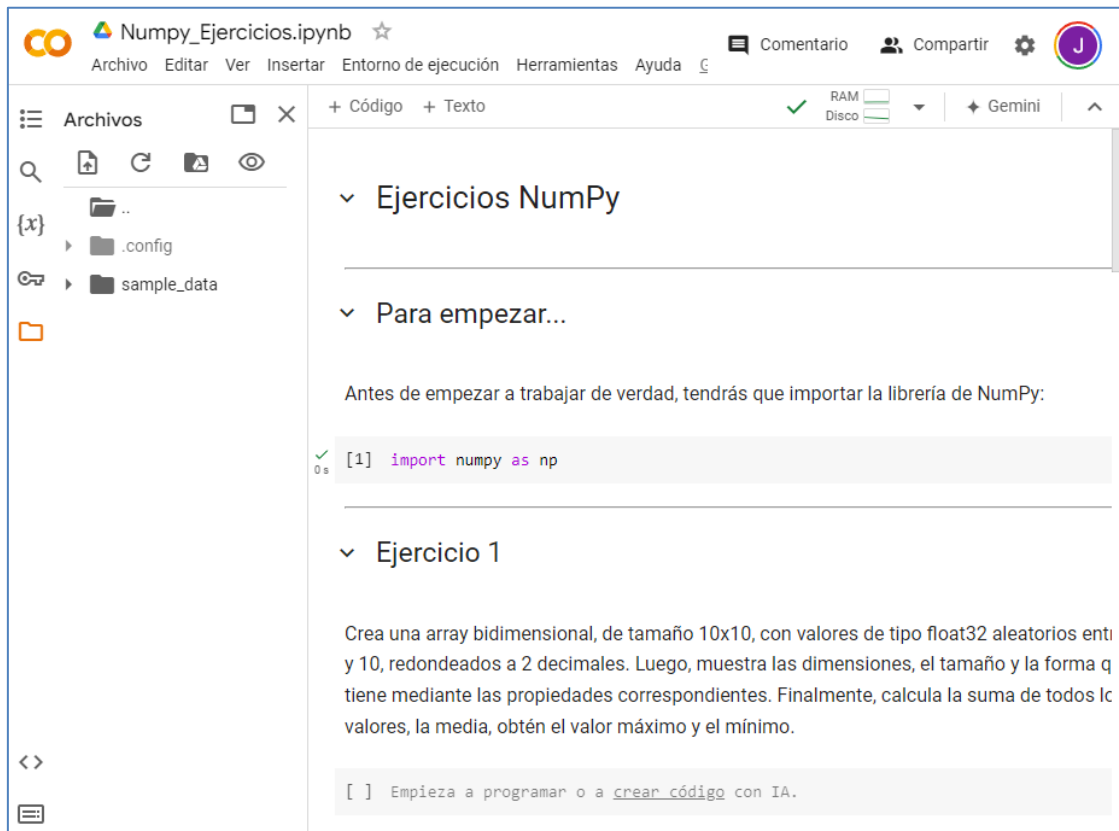
And we can easily view it's spectral structure using matplotlib's builtin specgram routine:

```
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');
```



## 1.2.- Google Colab

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, 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, pero requiere usar la versión de pago Colab Pro porque en la capa gratuita hay pocos minutos para probar.



Para utilizar este entorno de programación, solo hay que entrar en la web de colab:

<https://colab.research.google.com/>

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 el notebook que cuya url vamos a citar más abajo, tienes ejemplos de todas las etiquetas y caracteres que puedes utilizar y la forma de incluir todo tipo de elementos.

[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)

A continuación, se muestra un ejemplo de un notebook con consejos y cosas curiosas que aplicar a los cuadernos en colab. Puedes verlo completo yendo a:

[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 \(/notebooks/markdown\\_guide.ipynb\)](/notebooks/markdown_guide.ipynb).

You can also add math to text cells using [LaTeX \(http://www.latex-project.org/\)](http://www.latex-project.org/) to be rendered by [MathJax \(https://www.mathjax.org\)](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. 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)

<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*
csch@	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

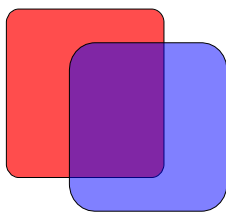
Colaboratory shares the notion of magics from Jupyter. There are shorthand annotations that change how a cell's text is executed. To learn more, see [Jupyter's magics page](http://nbviewer.jupyter.org/github/ipython/ipython/blob/1.x/examples/notebooks/Cell%20Magics.ip) (<http://nbviewer.jupyter.org/github/ipython/ipython/blob/1.x/examples/notebooks/Cell%20Magics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

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-dc3988fd1d2> 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.

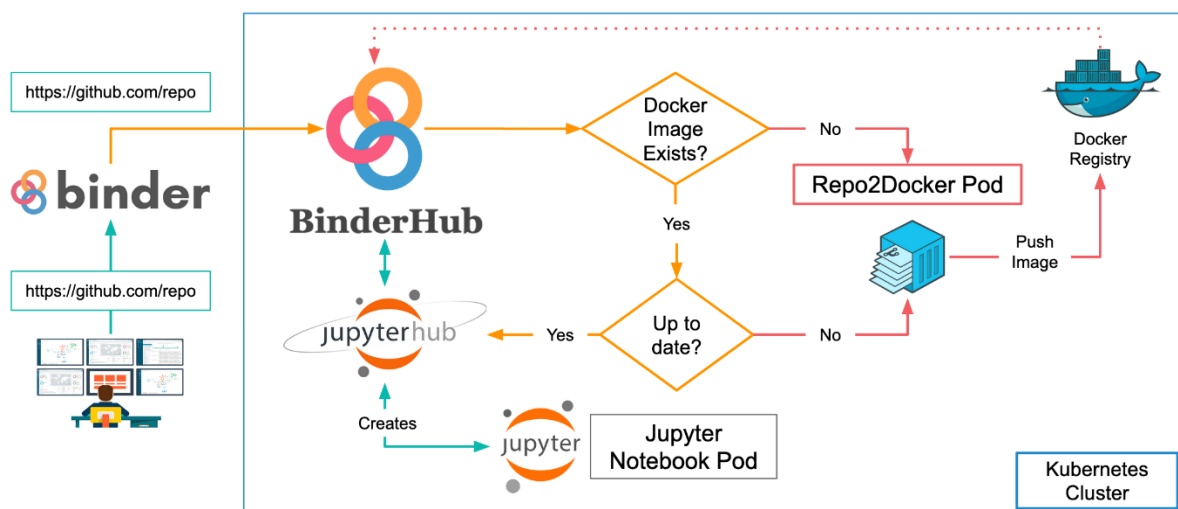
You can resolve and reply to comments, and you can target comments to specific collaborators by typing *+**[email address]*** (e.g., **+user@domain.com** ). Addressed collaborators will be emailed.

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

## 1.3.- Binder

**Binder** 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 quieres 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).



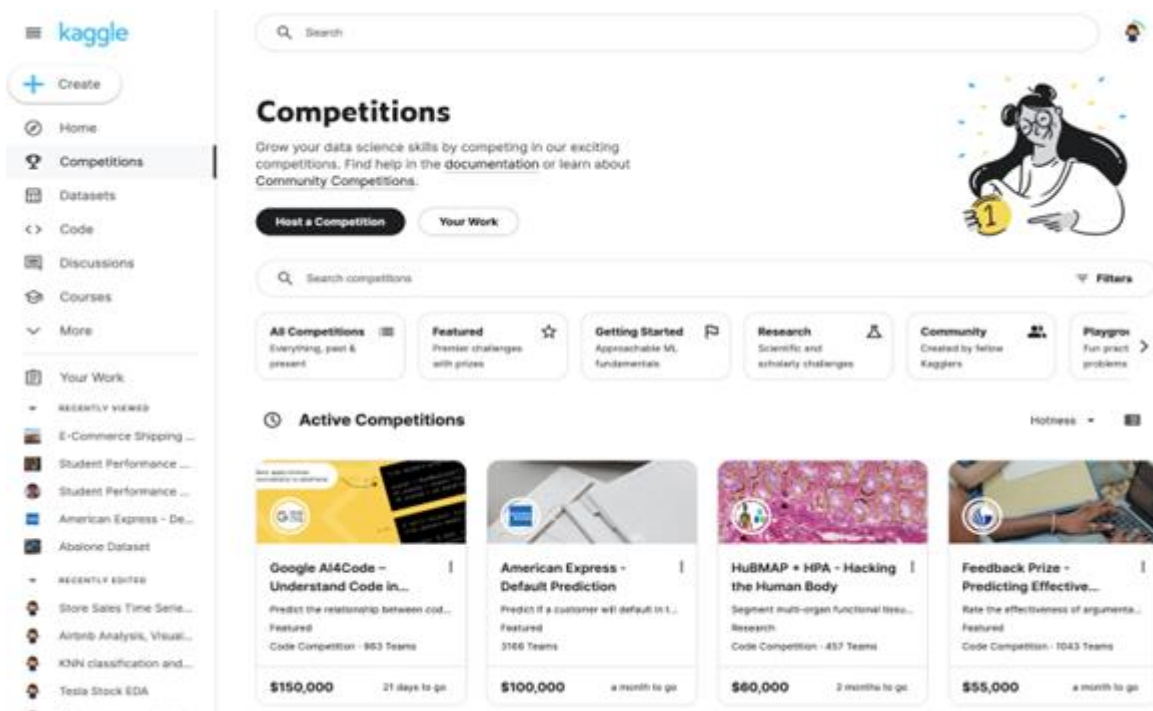
La experiencia del que recibe el enlace, es como la del ejemplo inferior, en el que hay una primera pantalla de inicialización del entorno, y después una interfaz de trabajo con el notebook ya cargado y todos los recursos del repositorio en el menú de la izquierda.

### Cómo funciona

1. Ingresa la información de tu repositorio.  
Proporciona una URL o un repositorio de GitHub que contenga cuadernos Jupyter, así como una rama, etiqueta o hash de confirmación. Launch creará tu repositorio Binder.
2. Creamos una imagen Docker de tu repositorio  
Binder buscará un archivo de dependencia, como requirements.txt o environment.yml, en el directorio raíz del repositorio. Los archivos de dependencia se usarán para crear una imagen Docker.
3. ¡Interactúa con tus cuadernos en un entorno en vivo!  
Un servidor JupyterHub alojará el contenido de tu repositorio. Te ofrecemos un enlace reutilizable y una insignia a tu repositorio en vivo que puedes compartir fácilmente con otros.

## 1.4.- Kaggle

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 más 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 quieres hacer. Desde la pantalla del dataset, si ya te has registrado, puedes pulsar sobre la opción "New notebook"



# 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.

### 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.

```
# 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 notebook
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

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.

notebook44626c1f55 Draft saved

File Edit View Run Add-ons Help

+ Draft Session (0m)

Code

```
# Imports libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# Loading data

df = pd.read_csv('../input/customer-analytics/Train.csv')
df.head()
```

[1]:

	ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Product_importance	Gender
0	1	D	Flight	4	2	177	3	low	
1	2	F	Flight	4	5	216	2	low	
2	3	A	Flight	2	2	183	4	low	
3	4	B	Flight	3	3	176	4	medium	
4	5	C	Flight	2	2	184	3	medium	

+ Code + Markdown

Train.csv

Train.csv (440.46 KiB) ../input/customer-analytics/Train.csv Copy 10 of 12 columns

ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Product_importance
1	D	Flight	4	2	177	3	low
2	F	Flight	4	5	216	2	low
3	A	Flight	2	2	183	4	low
4	B	Flight	3	3	176	4	medium

Input

customer-analytics

Output (60KB / 19.5GB)

/kaggle/working

Settings

Schedule a notebook run

Code Help

FIND CODE HELP

Find Code Help

Search for examples of how to do things

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.

## 2.- IDEs y editores para proyectos

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

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.

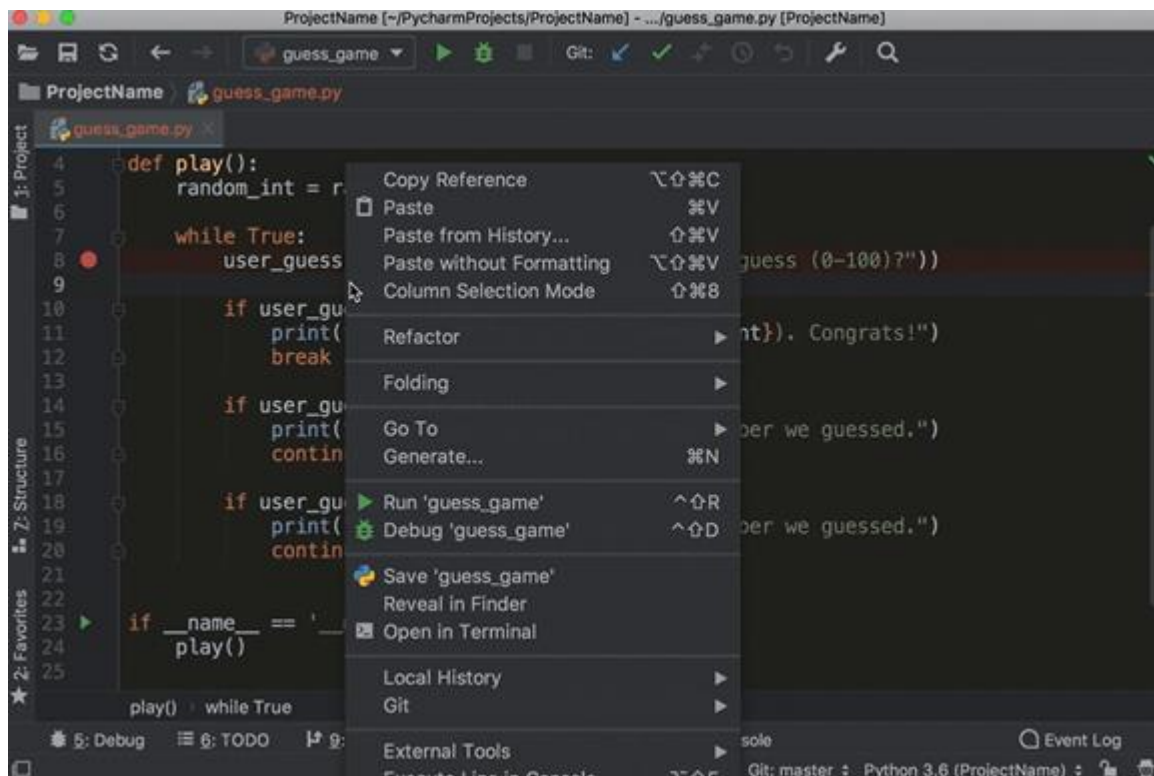
En las últimas actualizaciones de software, se han incluido extensiones para trabajar con Jupyter Notebooks en VSC (Visual Studio Code), pero es importante tener en cuenta que sigue siendo necesario instalar los paquetes y librerías que se quieran usar. Tutorial: <https://code.visualstudio.com/docs/datascience/data-science-tutorial>. 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.

## 2.2.- Pycharm

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.

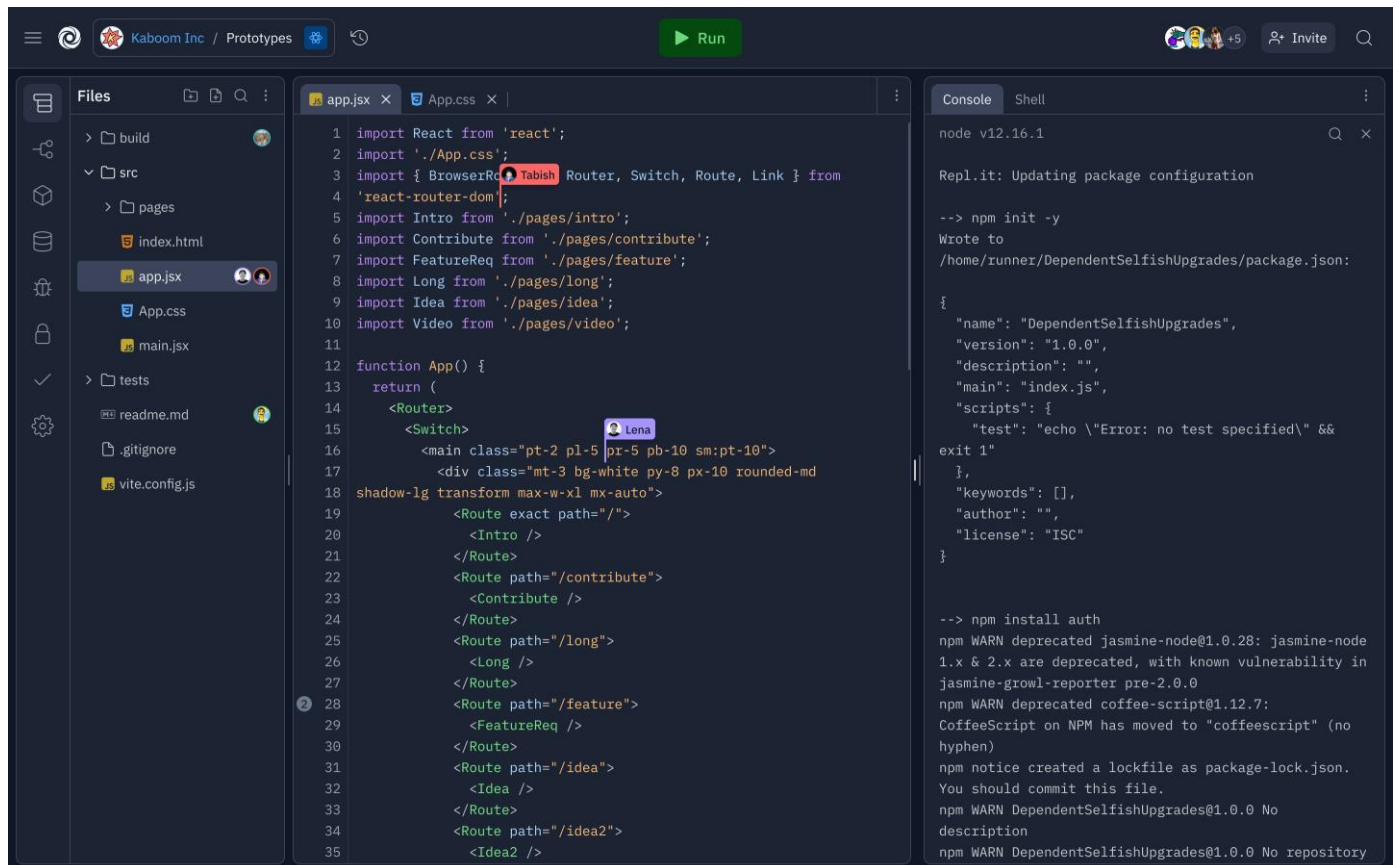
Al igual que VSC (Visual Studio Code), nos va a permitir realizar las acciones necesarias para control de versiones, desde una interfaz gráfica, para hacer commits, push y pull a repositorio. También tienes disponibles Toolkits o plugins para conectar con AWS (Amazon Web Services), GCP (Google Cloud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

<https://realpython.com/pycharm-guide/>

## 2.3.- Replit



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.

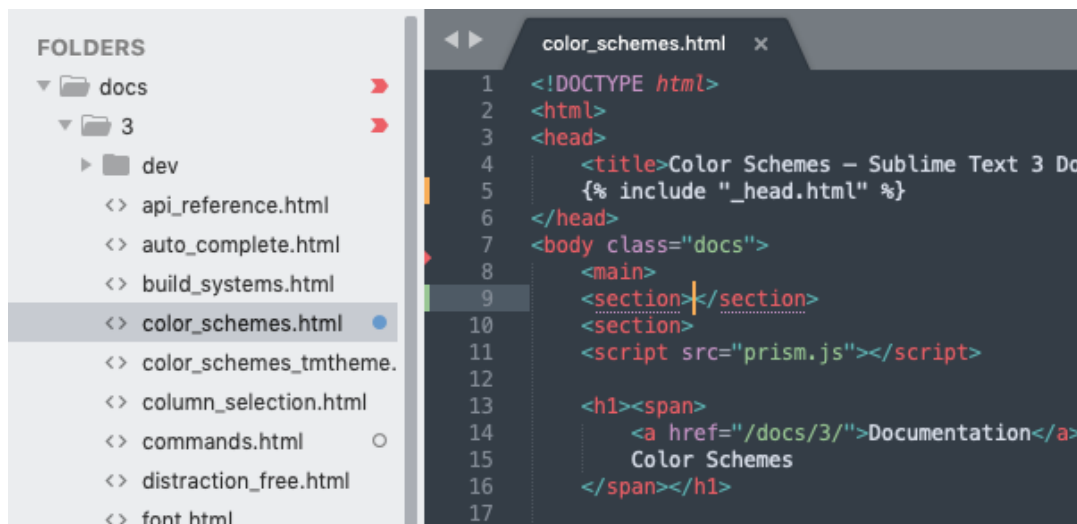
Replit cuenta con un escritorio principal donde puedes gestionar tus proyectos de una forma muy visual.

## 2.4.- Sublime Text

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:

- ✓ Minimapa 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 quieres 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:





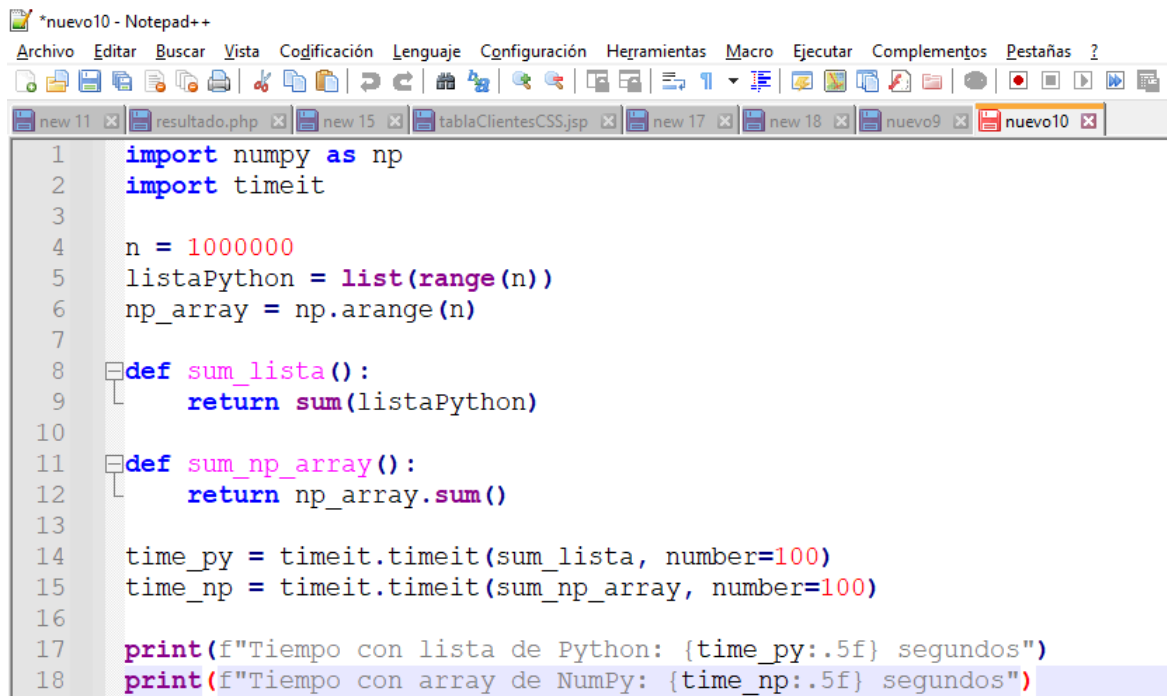
## 2.5.- Notepad++

Es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. Con soporte nativo para Microsoft Windows.

Se parece al Bloc de notas en cuanto al hecho de que puede editar texto sin formato y de forma simple. No obstante, incluye opciones más avanzadas que pueden ser útiles para usuarios avanzados como desarrolladores y programadores. Se distribuye bajo los términos de la licencia GPLv3.

### Características

- ✓ Coloreado y envoltura de sintaxis: si se escribe en un lenguaje de programación o marcado, Notepad++ es capaz de resaltar las expresiones propias de la sintaxis de ese lenguaje para facilitar su lectura.
- ✓ Pestañas: al igual que en muchos navegadores, se pueden abrir varios documentos y organizarlos en pestañas.
- ✓ Resaltado de paréntesis e indentación: cuando el usuario coloca el cursor en un paréntesis, Notepad++ resalta este y el paréntesis correspondiente de cierre o apertura. También funciona con corchetes y llaves.<sup>3</sup>
- ✓ Grabación y reproducción de macros.
- ✓ Soporte de extensiones: incluye algunas por defecto.
- ✓ Compatibilidad con infinidad de lenguajes y formatos.



The screenshot shows the Notepad++ application window titled '\*nuevo10 - Notepad++'. The menu bar includes Archivo, Editar, Buscar, Vista, Codificación, Lenguaje, Configuración, Herramientas, Macro, Ejecutar, Complementos, Pestañas, and ?. The toolbar contains various icons for file operations, editing, and development. The tab bar shows several open files: new 11, resultado.php, new 15, tablaClientesCSS.jsp, new 17, new 18, nuevo9, and nuevo10. The main editor area displays Python code with syntax highlighting. The code compares the performance of a list-based summation function with a NumPy array-based summation function. The list-based function is significantly faster.

```
1 import numpy as np
2 import timeit
3
4 n = 1000000
5 listaPython = list(range(n))
6 np_array = np.arange(n)
7
8 def sum_lista():
9     return sum(listaPython)
10
11 def sum_np_array():
12     return np_array.sum()
13
14 time_py = timeit.timeit(sum_lista, number=100)
15 time_np = timeit.timeit(sum_np_array, number=100)
16
17 print(f"Tiempo con lista de Python: {time_py:.5f} segundos")
18 print(f"Tiempo con array de NumPy: {time_np:.5f} segundos")
```

## 3.- Aplicaciones de IA en la nube y APIs

A lo largo de este apartado veremos productos SaaS disponibles en el mercado relacionados con el desarrollo de inteligencia artificial, con el objetivo de:

- 1.- Conocer el marco habitual en el que un profesional de IA se suele mover a la hora de utilizar la inteligencia artificial en un negocio.
- 2.- Entender la arquitectura de un proyecto que utiliza las distintas utilidades y servicios de una plataforma en la nube.
- 3.- Adquirir nociones de puesta en marcha de un proyecto que incluya uno de estos servicios de IA.
- 4.- Trabajar ejemplos y ejercicios sencillos en torno a estas plataformas:
  - 4.1.- Servicios de IA en Google Cloud Platform.
  - 4.2.- Servicios de IA en Amazon Web Services.
  - 4.3.- Servicios cognitivos y de aprendizaje automático en Microsoft Azure.
  - 4.4.- API para generación de textos e imágenes de Open AI.

El crecimiento que ha experimentado el ámbito de la inteligencia artificial en los últimos años, ha facilitado la aparición y consolidación de herramientas y servicios conectados que permiten a las organizaciones montar las soluciones que necesitan sin necesidad de desarrollar un proyecto de cero. Existen modelos entrenados para tareas concretas, o incluso modelos que se pueden ajustar a ciertos parámetros del problema al que se van a aplicar, y todo esto se controla a través de una interfaz de trabajo, generalmente online y conectada a un entorno en la nube.

### 3.1.- Google Cloud Platform

GCP o Google Cloud Platform es uno de los principales proveedores de computación en la nube que utilizan las empresas y organizaciones. Cuenta con un catálogo de servicios muy extenso, entre los que hay servicios de tipo "llave en mano" o que están preparados para usar en procesos de negocio muy concretos y habituales. Por ejemplo, Document AI permite el análisis y extracción de información de un tipo concreto de documentos, como facturas o impresos de solicitud de hipotecas, y el usuario apenas tiene que preocuparse de la parte de inteligencia artificial.

Para trabajar con GCP es necesario registrarse y crear una cuenta con datos de facturación. La mayoría de los servicios son gratuitos hasta un cierto volumen de utilización, y para usuarios nuevos hay ofertas y créditos de regalo para poder probar durante los primeros meses. Es importante informarse muy bien de las condiciones de facturación de cada servicio, y es recomendable fijar límites de gasto, para evitar sorpresas.

La interfaz de trabajo se denomina "Consola" y cuenta con una pantalla de inicio con tus proyectos recientes:



# Te damos la bienvenida








Estás trabajando en [AutoML project](#)

Número de proyecto: 43084832555 ID del proyecto: automl-project-354011

[Panel](#) [Recomendaciones](#)

[+ Crea una VM](#) [+ Ejecuta una consulta en BigQuery](#) [+ Crea un clúster de GKE](#) [+ Crea un bucket de almacenamiento](#)

## Acceso rápido

 AutoML project  Panel – Vertex AI – AutoML pr...	 AutoML project  Entrenamiento – Vertex AI – A...	 AutoML project  Vertex AI – AutoML project	 Wimbabot  bitcoin_blockchain – BigQuery...
 AutoML project  Cuentas de facturación – Fact...	 AutoML project  Principal – IAM y administraci...	 AutoML project  Configuración del proyecto	 AutoML project  Modelos – Vertex AI – AutoML...

En esta sección de la unidad, vamos a conocer algunos de los servicios de GCP relacionados con la inteligencia artificial y veremos también algunos ejemplos de uso.

El ecosistema de aplicaciones de la nube de Google es inmenso, y vamos a fijarnos en una pequeñísima parte, pero cada proyecto suele requerir utilizar varios servicios en conjunto. Aquí tienes una hoja resumen con casi todos los servicios, pero te recomendamos ir al mapa interactivo de Google Cloud: <https://googlecloudcheatsheet.withgoogle.com/>

### 3.1.1 – VertexAI y AutoML

Vertex AI reúne los servicios de Google Cloud que permiten crear modelos de aprendizaje automático en una interfaz con APIs únicas y unificadas. Se pueden utilizar modelos ya entrenados y listos para usar, o se pueden entrenar modelos adaptados a la lógica de negocio concreta usando AutoML.

Para acceder a ellos, desde la pantalla inicial en la consola de GCP, pinchando en el menú lateral en la parte superior izquierda de la pantalla, debes bajar hasta llegar a la sección de Inteligencia Artificial.

En esta sección, entre los diferentes servicios que proporciona GCP para el desarrollo de la IA, encontramos Vertex AI. Al pinchar en esta opción, se despliega su propio menú, que representa de forma casi ordenada, las fases que se siguen en cualquier proceso de preparación y despliegue de un modelo de aprendizaje automático. En cada apartado del proceso, será necesario fijar los parámetros propios del proyecto.

Una de las grandes ventajas de esta plataforma es que el proceso, desde la carga de datos, hasta la puesta en producción del modelo, es muy fluido y sencillo, gracias al propio flujo de trabajo que han creado, en el que vas configurando, en cada apartado, lo que necesita tu modelo.


Desde el "Panel", se cuenta con un vistazo general del proyecto.

En la sección de "Conjunto de datos", se seleccionan fuentes de datos desde Google Storage, BigQuery, o por subida directa de un archivo csv desde el ordenador. Una vez cargado uno o varios conjuntos de datos, se pueden analizar dichos datos y generar estadísticas para una exploración más completa.


En la sección de "Entrenamiento", se configuran parámetros, tipo de técnica a aplicar, etc. Para poder configurar bien el modelo, es necesario entender y conocer bien el tipo de problema en el que estamos trabajando. La naturaleza de los datos, si los casos están etiquetados o no en función de un campo objetivo para las predicciones, nos permitirá aplicar técnicas de aprendizaje automático supervisado, como la clasificación o la regresión. Las

estadísticas de los datos también nos permiten interpretar algunos comportamientos de las predicciones. Es importante dedicar tiempo a conocer en profundidad el conjunto de datos con el que se está trabajando.


La propia plataforma va a automatizar gran parte de los parámetros y configuración para la creación del modelo y su posterior entrenamiento. Para ello, en la sección de conjunto de datos, presentan los tipos de problemas que tienen categorizados para un proceso guiado.



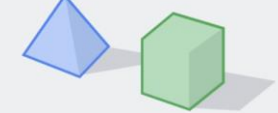
☒ **Clasificación de imágenes (una etiqueta)**  
Predice la etiqueta correcta que deseas asignar a una imagen.



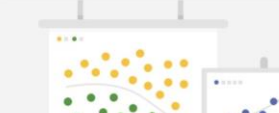
☐ **Clasificación de imágenes (varias etiquetas)**  
Predice todas las etiquetas correctas que deseas asignar a una imagen.




☐ **Detección de objetos de imagen**  
Predice todas las ubicaciones de los objetos en los que tienes interés.




☐ **Segmentación de imágenes**  
Predice áreas por píxel de una imagen con una etiqueta.




☒ **Regresión/clasificación**  
Predice el valor de una columna objetivo. Es compatible con tablas de cientos de columnas y millones de filas.




☐ **Previsión** VISTA PREVIA  
Predice la probabilidad de que sucedan ciertos eventos o demandas.




☒ **Clasificación de texto (una etiqueta)**  
Predice la etiqueta correcta que deseas que se asigne a un documento.



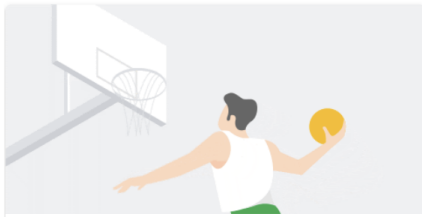
☐ **Clasificación de texto (varias etiquetas)**  
Predice las etiquetas correctas que deseas que se asignen a un documento.



☐ **Extracción de entidades de texto**  
Identifica entidades en tus elementos de texto.



☐ **Análisis de opiniones de texto**  
Comprende la opinión general expresada en un bloque de texto.



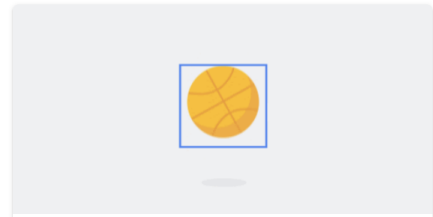
☒ **Reconocimiento de acciones de video**

Identifica los momentos de acción en los videos.



☐ **Clasificación de videos**

Obtén predicciones de etiquetas para fotogramas, imágenes y videos completos.



☐ **Seguimiento de objetos de video**

Obtén etiquetas, seguimientos y marcas de tiempo de los objetos que quieres supervisar en un video.

Cuando el entrenamiento ya ha concluido, se pueden ver, en la sección "Modelo" las métricas del desempeño conseguido.

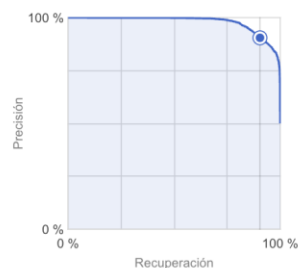
Límite de confianza   0.5

### Todas las etiquetas

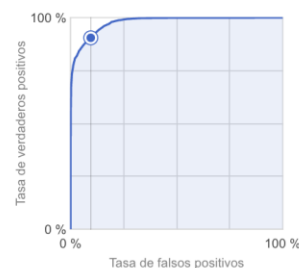
PR AUC 	0.978
ROC AUC 	0.977
Log loss 	0.188
F1 score 	0.9046875
Precisión 	90.5 %
Recuperación 	90.5 %
Fecha de creación	21 jun 2022 20:25:27

Para evaluar tu modelo, configura el umbral de confianza a fin de ver el impacto que tiene sobre la precisión y la recuperación. El mejor umbral de confianza depende de tu caso de uso. Lee algunos [casos de ejemplo](#) para aprender cómo se pueden usar las métricas de evaluación.

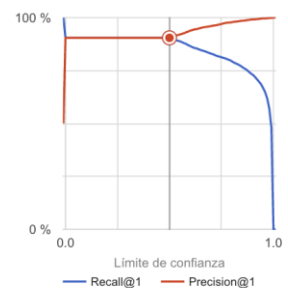
Curva de precisión-recuperación 



Curva ROC 



Precisión-recuperación por umbral 



## Matriz de confusión

En esta tabla, se muestra la frecuencia con la que el modelo clasificó cada

Etiqueta de confianza	Etiqueta predicha	
	1	2
1	96%	4%
2	54%	46%

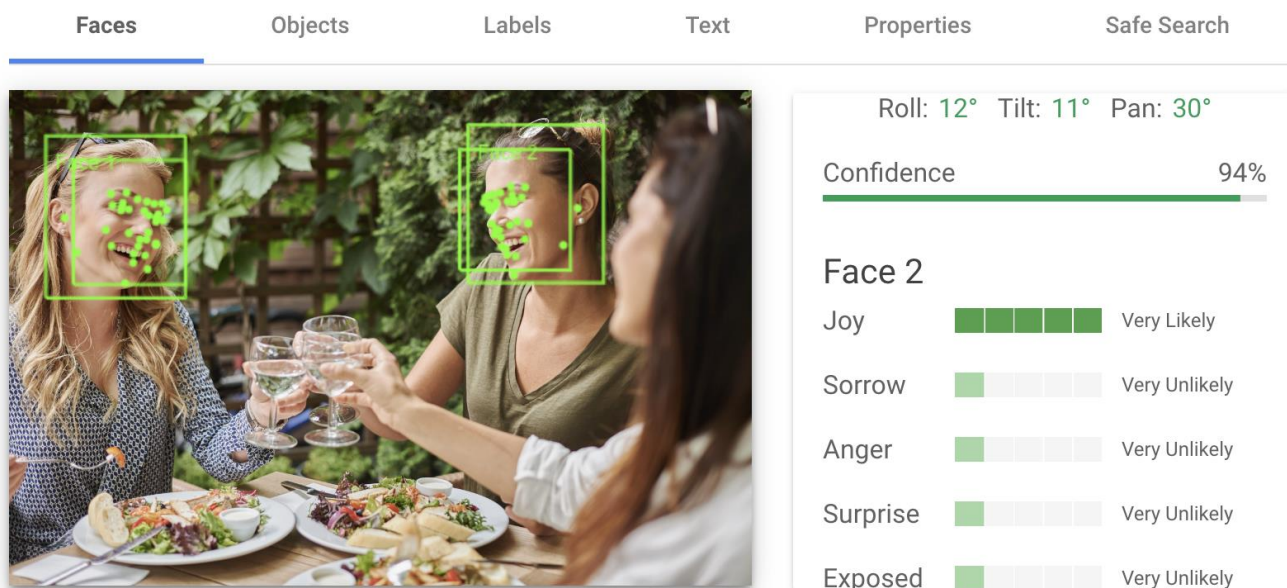
Para poder probar e implementar el modelo, es necesario crear un "endpoint" o acceso al modelo en formato API REST. Esta tarea también la resuelve Vertex AI, prácticamente haciendo click en un botón. Tras tenerlo implementado, podemos hacer una prueba por valores, predicción por lotes y utilizar el "endpoint" o "extremo" en una aplicación.

### 3.1.2 – VisionAI

Google Cloud Platform ya cuenta con un producto de reconocimiento de imagen entrenado genérico que puede ser útil cuando no se cuenta con conjuntos de datos ni experiencia en creación y entrenamiento de modelos. La API de Vision de GCP ofrece:

- ✓ Reconocimiento facial
- ✓ Reconocimiento de objetos en una imagen
- ✓ Identificación de etiquetas para una imagen
- ✓ Extracción de texto de una imagen
- ✓ Detección de elementos no seguros en imagen (violencia, sexo, racismo, etc)

En el caso de reconocimiento facial, el modelo detecta los elementos del rostro y según suposiciones relativas, ha sido entrenado para etiquetar respecto a las principales emociones. También detecta orientación del rostro en la imagen.



También se pueden reconocer objetos en una imagen a través de la detección de bordes y formas, ya que el modelo ha sido entrenado con una gran cantidad de imágenes etiquetadas que incluían objetos de todo tipo. Si necesitásemos que se reconociesen en la imagen objetos peculiares o muy originales, ya tendríamos que entrenar nuestro propio modelo, con imágenes bien etiquetadas que contuviesen dichos objetos.

En muchas aplicaciones, especialmente si van a tener una calificación por edades que garantice un uso seguro por parte de menores, es necesario implementar filtros que bloqueen o adviertan de imágenes que contengan elementos o situaciones sensibles.

Una utilidad muy extendida para esta API es la de detección de texto en imagen. El modelo reconoce los trazos típicos de caracteres escritos y lo entrega por bloques. Es importante tener en cuenta que el modelo solo detecta que hay texto, por lo que después, sería necesario aplicar un modelo de procesamiento de lenguaje para clasificar la intención o significado de dichos textos y así poder clasificarlo para su uso posterior.



## +Block 8

DIEZ OBESO 70

GRANOS , LEGUMBRES  
ANTES

90 Y 92

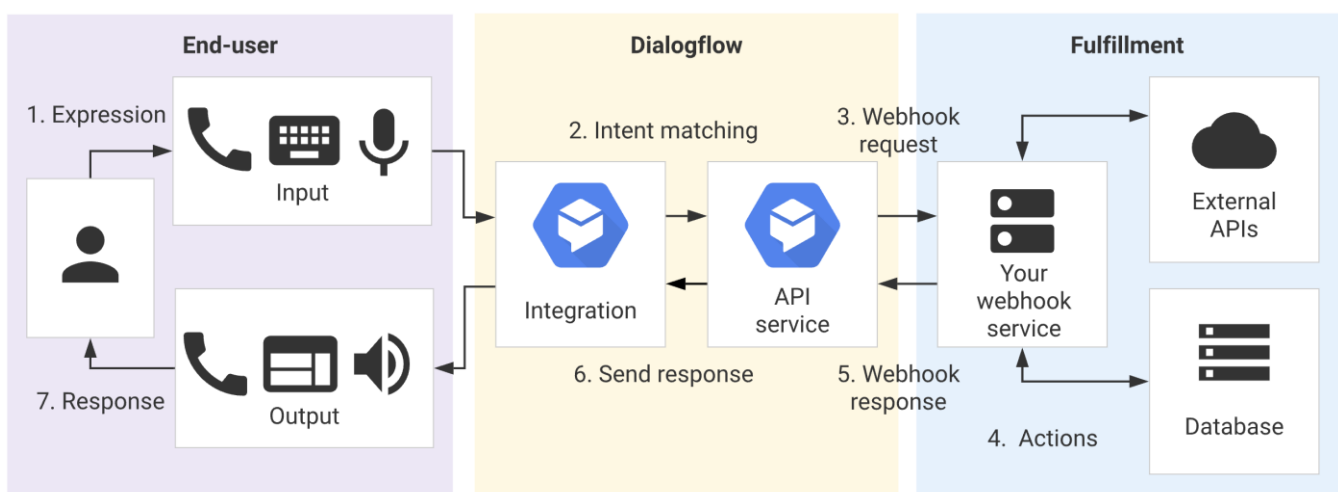
## +Block 9

CASA FUNDADA EN 188  
1

Cuando las funcionalidades de la API de Vision no son suficientes y ya es necesario entrenar modelos de aprendizaje automático para reconocimiento de imagen con parámetros concretos o para detectar elementos muy concretos de un ámbito de actuación, es mejor recurrir a AutoML Vision, integrado o no dentro del entorno de Vertex AI.

### 3.1.3 – DialogFlow

Dialogflow es la plataforma de Google Cloud para aplicaciones de comprensión del lenguaje natural. Está orientada y optimizada para crear una interfaz de usuario de conversación, con la facilidad extra de poder integrarla en todo tipo de aplicaciones, sistemas automatizados, robots, etc. Además, permite la implementación de la extensión de Agent Assist, para trabajar en colaboración con agentes humanos en un centro de atención al cliente. Una de las características más potentes De esta herramienta, es la posibilidad de hacer llamadas a un servicio pre-definido. Por ejemplo, si el usuario desea reservar mesa en tu restaurante, el servicio puede verificar la base de datos y responder acerca de disponibilidad para el día y hora deseados.



Dialogflow acepta entradas de texto o audio, por ejemplo, a través de Google Assistant, o desde una llamada de teléfono. También puede responder a través de texto o de voz sintética. Esta herramienta es parte de la oferta de "Conversational AI" dentro de GoogleCloud, que provee de varios servicios



que complementan Dialogflow para crear una aplicación de negocio.

Crear un asistente conversacional requiere, básicamente, dos cosas:

- ✓ Identificar las intenciones del usuario cuando hace una pregunta o petición
- ✓ Una vez interpretada la pregunta e identificada la intención, recurrir a las respuestas disponibles para entregar la más adecuada al usuario.

En primer lugar, es necesario crear el agente:

2

Dialogflow Essentials US ▼

+ Create new agent

☰ View all agents

3

Agent name CREATE

DEFAULT LANGUAGE ⓘ English - en ▼  
Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE (GMT+1:00) Europe/Madrid ▼  
Date and time requests are resolved using this timezone.

GOOGLE PROJECT Create a new Google project ▼  
Enables Cloud functions, Actions on Google and permissions management.

AGENT TYPE ☒ Set as Mega Agent  
Combine multiple Dialogflow agents (i.e. sub agents) into a single agent (i.e. [mega agent](#)).

Después, daremos de alta los "intents" o intenciones. Existen una serie de intenciones "por defecto", como saludos, bienvenida o preguntas no entendidas:

Settings • Default Welcome Intent SAVE ☰

Contexts

User says Search in user says 🔍

☞ Add user expression

☞ hey

☞ Buenas

☞ Que tal?

☞ Hola

Events ⓘ

WELCOME ⓘ Add event

☹ Fallback intents are triggered if a user's input is not matched by any of the regular intents or enabled domains. [Read more in documentation](#)

Contexts

Events ⓘ

Action input.unknown

Response ⓘ

DEFAULT +

Text response

1 Ups, no he entendido a que te refieres.

2 ¿Podrías repetirlo, por favor?

3 ¿Disculpa?

4 ¿Declaras?

5 ¿Cómo?

6 Enter a text response variant

ADD MESSAGE CONTENT

Try it now

Agent Domains

USER SAYS hola COPY CURL

DEFAULT RESPONSE PLAY

¡Hola!

INTENT Default Welcome Intent

ACTION input.welcome

Y, posteriormente, iremos generando el resto de intentos, añadiendo sus correspondientes "utterances" o cuestiones sugeridas, que son ejemplos de expresiones que podrían utilizarlos usuarios cuando están preguntando o solicitando una intención concreta. Después, serán frases de entrenamiento para el agente. Se recomienda incluir entre 10 y 15 frases por cada intento. Es necesario identificar qué parámetros debemos solicitar al usuario en caso de que no hayan sido indicados por éste en sus preguntas o interacciones, y finalmente, hay que definir las respuestas a dar cuando el agente identifica que el usuario se refiere a esa intención.

● ReservaMesa

” llamaba para reservar una mesa para dos personas

” tienen mesa para dos hoy a mediodía?

” tienen disponibilidad para comer hoy dos personas?

” hay disponible una mesa para hoy?

” Querría una mesa para dos

” quiero reservar una mesa

Action and parameters

Enter action name

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?	PROMPTS ?
<input checked="" type="checkbox"/>	number	@sys.number	\$number	<input checked="" type="checkbox"/>	Para cuántas pe...
<input checked="" type="checkbox"/>	date-time	@sys.date-time	\$date-time	<input checked="" type="checkbox"/>	¿Para qué fecha...
<input checked="" type="checkbox"/>	Restaurante	@Restaurante	\$restaurante	<input checked="" type="checkbox"/>	¿En el Restau... <div>↑ ↓ ⋮</div>
<input type="checkbox"/>	location	@sys.location	\$location	<input type="checkbox"/>	—
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

Responses ?

DEFAULT +

Text Response

1

Su reserva en \$restaurante el día \$date-time ha sido confirmada

2

Reserva realizada para el día \$date-time

3

Enter a text response variant

## 3.2.- Amazon Web Services (AWS)

La plataforma de servicios en la nube de Amazon es una de las más conocidas y utilizadas a nivel mundial. Fue lanzada oficialmente en 2006 a raíz de una propuesta de Chris Pinkham y Benjamin Black, que vieron una oportunidad de negocio en la utilización de recursos sobredimensionados en Amazon.com.

Cuenta con una colección de servicios para desarrolladores y organizaciones muy amplio, que permite que toda la lógica y gestión de negocio esté integrada en un mismo flujo de trabajo. Se accede a través de HTTP, utilizando protocolos REST y SOAP. Fue de las primeras ofertas de pago por uso en infraestructura que se dio en un momento de resurgimiento de propuestas en internet tras la burbuja de las .com y ese formato facilitó la entrada de muchas startups y proyectos que no hubiesen sido viables con los formatos disponibles en el hosting tradicional.

Uno de los servicios más valorados para proyectos emergentes y que necesitan escalabilidad, es la arquitectura sin servidor, utilizando las denominadas funciones "Lambda", lanzado en 2014.

En el ámbito de la inteligencia artificial, cuenta con un servicio administrado para la creación, entrenamiento y despliegue de modelos de aprendizaje automático denominado SageMaker.

Ofrece servicios más concretos y enfocados en aplicaciones concretas de visión artificial, bots conversacionales, y analítica avanzada. Analizamos en las siguientes secciones, con más detalle, los módulos de Rekognition, para reconocimiento de imagen; Comprehend, para análisis de texto; y SageMaker, como entorno de trabajo con modelos de aprendizaje automático.

### 3.2.1 – Amazon SageMaker

Dentro de las opciones que ofrece la plataforma de AWS en el ámbito de la inteligencia artificial, el entorno integrado para la creación, entrenamiento y despliegue de modelos de machine learning en AWS se denomina SageMaker. Contiene varios servicios que se gestionan a través de un panel de control, un IDE personalizado llamado SageMaker Studio, y una interfaz de visualización y presentación llamada Canva. La mayoría del trabajo de un MLops se va a desarrollar dentro de SageMaker Studio, que ofrece la experiencia de programación en formato "notebook" con una serie de herramientas de apoyo extra, y una funcionalidad muy interesante denominada "Autopilot".

Para cada fase del proceso, hay una sección con los servicios más utilizados, donde se pueden ir dando de alta y lanzando el dataset, arquitectura del modelo, modelo predictivo y despliegue.

Como hemos comentado previamente, SageMaker cuenta con una utilidad para automatizarla parte más técnica del proceso de creación y entrenamiento de un modelo de aprendizaje automático. Autopilot analiza el dataset y la morfología de los datos objetivo, y automatiza la selección del tipo de modelo y parámetros asociados al entrenamiento. Pero este servicio tiene un coste mayor que un proyecto normal.



## Recomendación

AWS provee de un entorno para pruebas y aprendizaje con las funcionalidades más básicas y recursos de computación de forma gratuita. Se trata de **SageMaker Studio Lab**. Es necesario registrarse y hay ciertos límites de uso, pero permite utilizar proyectos de prueba de aprendizaje automático dentro de otros proyectos que se tengan desplegados en AWS.

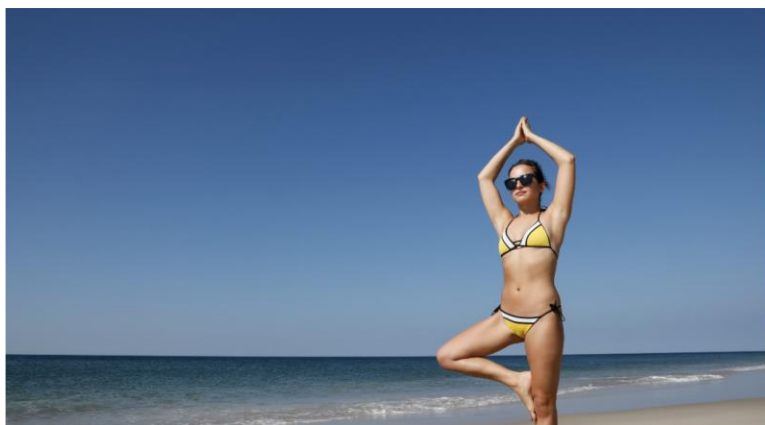
### 3.2.2 – Amazon Rekognition

Rekognition es el servicio de reconocimiento de imagen de AWS ya pre-entrenado y desplegado, que se puede probar a través de una interfaz demostrativa, y que se puede integrar en cualquier proyecto mediante llamadas a su API. Un primer modo, de reconocimiento de "etiquetas" en una imagen, da como resultado los objetos que se podrían encontrar en dicha imagen.



▼ Results	
Car	98.8 %
Automobile	98.8 %
Vehicle	98.8 %
Transportation	98.8 %
Person	98.3 %
Human	98.3 %

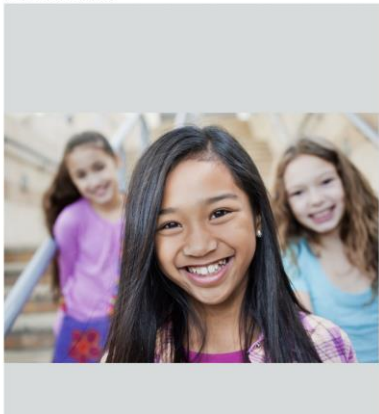
El modo de moderación de imágenes, detecta el grado de probabilidad de que la imagen contenga contenido sensible que deba desenfocarse o que deba activar algún tipo de acción.



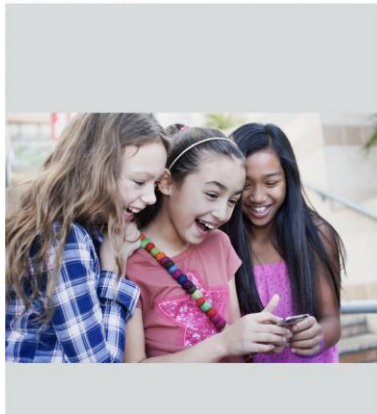
Suggestive	86 %
Female Swimwear Or Underwear	86 %
Revealing Clothes	72.5 %
► Request	
► Response	

En el modo de reconocimiento facial, se reconocen con bastante precisión factores como rasgos masculinos o femeninos, si está sonriendo o demostrando alguna emoción concreta detectable, el rango de edad, etc. Otro modo interesante del módulo Rekognition, es la función de comparación facial, para identificar el rostro de determinadas personas dentro de otras imágenes. En el siguiente ejemplo, la función reconoce la coincidencia del rostro incluso aunque está parcialmente tapado.

Reference face



Comparison faces



## ▼ Results



=



Similarity

99.9 %



≠



Finalmente, también se puede reconocer la presencia de texto en una imagen, identificando letras y números o caracteres especiales, devolviendo la cadena de caracteres reconocida como salida de la función.



## ▼ Results

J389 | NLT

## ► Request

## ► Response

A la vista de la imagen que hay encima, ¿se te ocurre alguna utilidad actual?

En todos los casos, se puede revisar tanto la llamada como la respuesta del ejemplo, que es un archivo tipo JSON con los parámetros "clave-valor" del reconocimiento. Será necesario conocer estos parámetros a la hora de hacer la implementación en un proyecto o aplicación.

Por ejemplo, una solicitud tendría que tener este esquema:

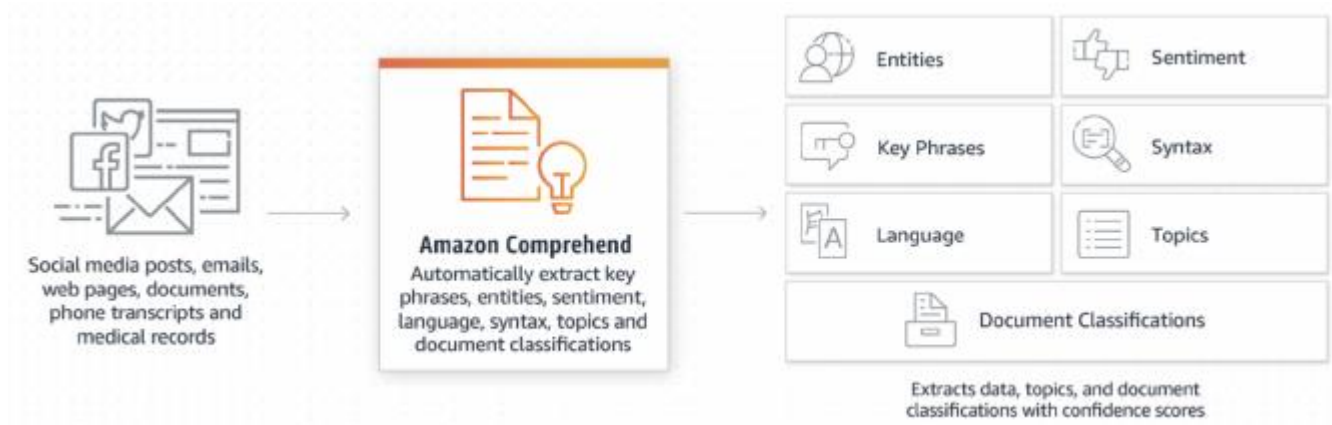
```
{
  "Image": {
    "S3Object": {
      "Bucket": "rekognition-console-sample-images-prod-cmh",
      "Name": "green_car.jpeg"
    }
  }
}
```

Y la respuesta (parcial) de la aplicación, con el texto detectado en la imagen "J389 NLT" de la matrícula y los parámetros del polígono que lo contiene, sería:

```
{
  "TextDetections": [
    {
      "DetectedText": "J389 NLT",
      "Type": "LINE",
      "Id": 1,
      "Confidence": 99.13363647460938,
      "Geometry": {
        "BoundingBox": {
          "Width": 0.1763957142829895,
          "Height": 0.073918916285038,
          "Left": 0.4644404947757721,
          "Top": 0.7082650661468506
        },
        "Polygon": [
          {
            "X": 0.46971258521080017,
            "Y": 0.7082650661468506
          }
        ]
      }
    }
  ]
}
```

### 3.2.3 – Amazon Comprehend

Otro de los módulos preparados para integrar en aplicaciones, es el de procesamiento de lenguaje natural, denominado AWS Comprehend.



Recuerda que el procesamiento de lenguaje natural (NLP) es el conjunto de técnicas computacionales en el ámbito del aprendizaje automático, que tienen como objetivo identificar la intención del interlocutor, contenida en una expresión escrita, o hablada y transformada a texto escrito, y clasificarla o asociarla a opciones o valores de variables de salida. Por ejemplo, un algoritmo puede detectar, a partir de la valoración que hace un usuario de una película, si dicho comentario refleja una actitud positiva o negativa. Es decir, sí le ha gustado o no.

La herramienta de AWS, Comprehend, trabaja bien con los "tokens" o palabras clave presentes en diferentes tipos de texto, pero es necesario configurar el tipo de problema para que el modelo pueda trabajar.

Comprehend, divide el texto en unidades con sentido y las analiza. Por ejemplo, frente a un mensaje cualquiera, la herramienta puede identificar expresiones clave que puedan contribuir a la ejecución de tareas posteriores, como encontrar unidades de información de tipo fechas, cantidades, etc:

Una función muy utilizada en este tipo de modelos es el análisis de sentimiento, que nos puede dar un valor aproximado del grado de positividad o negatividad que hay en un texto. En este ejemplo, vemos como asigna un 99% de negatividad a un comentario negativo en una web de tienda online:

#### Analyzed text

Hice el pedido durante el periodo en el que la sección de productos de menaje estaba de oferta, pero no se me aplicó el descuento. Además, el producto no ha cumplido para nada con mis expectativas. Venía dañado y con el precinto roto. Aunque inicialmente funcionó, tras un par de usos empezó a dar problemas y ahora ya es completamente inservible. He solicitado ayuda al email de atención al cliente, pero no he obtenido respuesta. No recomiendo esta tienda online en absoluto, pues los productos son de mala calidad y el servicio es pésimo.

#### ▼ Results

##### Sentiment

Neutral	Positive	Negative	Mixed
0.00 confidence	0.00 confidence	0.99 confidence	0.00 confidence

#### ▼ Application integration

API call and API response of DetectSentiment API. [Info](#)

##### API call

```
1 {  
2   "Text": "Hice el pedido durante el periodo en el  
   que la sección de productos de menaje estaba  
   de oferta, pero no se me aplicó el descuento.  
   Además, el producto no ha cumplido para nada  
   con mis expectativas. Venía dañado y con el  
   precinto roto. Aunque inicialmente funcionó,  
   tras un par de usos empezó a dar problemas y  
   ahora ya es completamente inservible. He  
   solicitado ayuda al email de atención al  
   cliente, pero no he obtenido respuesta. No  
   recomiendo esta tienda online en absoluto,  
   pues los productos son de mala calidad y el  
   servicio es pésimo.",  
3   "LanguageCode": "es"  
4 }
```

##### API response

```
1 {  
2   "Sentiment": {  
3     "Sentiment": "NEGATIVE",  
4     "SentimentScore": {  
5       "Positive": 0.00007150517194531858,  
6       "Negative": 0.9997981190681458,  
7       "Neutral": 0.00009866517939371988,  
8       "Mixed": 0.00003175950041622855  
9     }  
10  }  
11 }
```

## 3.3.- Microsoft Azure

Microsoft, aunque ha sido mucho más conocida siempre por sus productos de ofimática para usuario final, sistema operativo Windows y algunos productos de software para empresas, ha desplegado también su plataforma de "cloud computing", conocida como Azure. Sus últimos esfuerzos por abrir su comunidad y ofrecer servicios que puedan competir con otras nubes, han dotado a Azure de algunas propuestas muy interesantes.

- ✓ Azure Applied AI Services
- ✓ Azure Cognitive Services
- ✓ Azure Machine Learning
- ✓ Infraestructura de inteligencia artificial

### 3.3.1 – Cognitive Services

Azure Cognitive Services es la suite de modelos listos para usar de Microsoft. Funciona a través de llamada API e implementa servicios de reconocimiento de voz, texto, comprensión de lenguaje natural, reconocimiento de imagen y analítica avanzada de datos para toma de decisiones. Permite utilizar, en un proyecto, funcionalidades de inteligencia artificial, sin necesidad de desarrollo concreto de machine learning.

Servicios de voz:

- ✓ Speech to Text: transcripción de voz en texto legible.
- ✓ Text to Speech: conversión de texto en una locución audible.
- ✓ Speech Translation: traducción simultánea de voz.
- ✓ Speaker recognition: identificación de un interlocutor concreto en audio.

Servicios de lenguaje:

- ✓ Reconocimiento de entidades.
- ✓ Análisis de opiniones.
- ✓ Respuesta a preguntas.
- ✓ Language Understanding
- ✓ Translator Text.

Servicios de visión:

- ✓ API de reconocimiento facial.
- ✓ Computer Vision.
- ✓ Custom Vision.

Servicios para toma de decisiones:

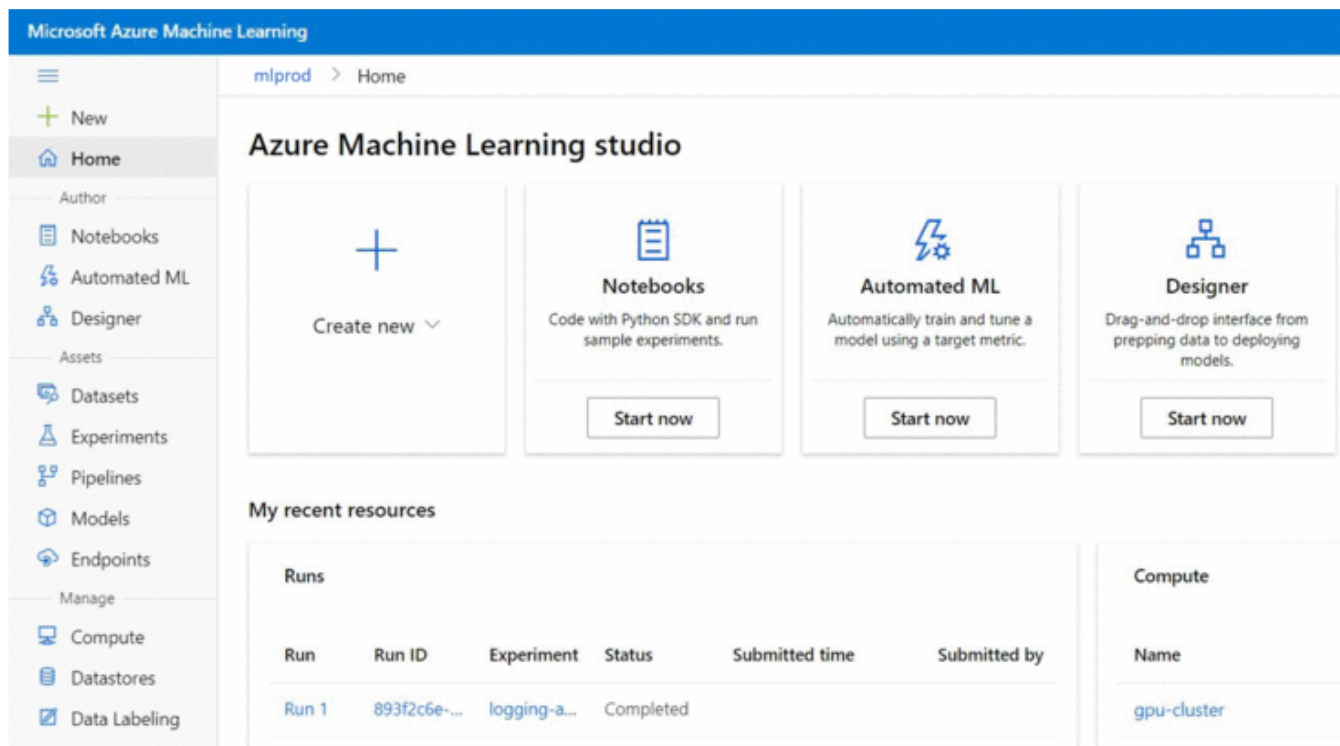
- ✓ Detector de Anomalías: permite hacer una identificación temprana de posibles problemas.
- ✓ Moderador de contenido: detecta contenido potencialmente ofensivo o no deseado.
- ✓ Personalizador.

### 3.3.2 – AzureML

Azure Machine Learning es el servicio para crear y entrenar modelos de cero, con la facilidad de conectarlos al resto de servicios necesarios en un proyecto de forma eficiente e integrada. Cuenta con un escritorio específico de trabajo, denominado Studio, en el que se puede iniciar



y gestionar los modelos, sus implementaciones, métricas e historiales de ejecución.



La interfaz permite trabajar con cuadernos Jupyter Notebook, utilizar librerías para aprendizaje automático y la flexibilidad de configurar los hiperparámetros en el código. Cuenta con una herramienta de depurado y permite la integración con el entorno de programación Visual Studio Code, con lo que se integra con relativa facilidad en proyectos que ya están desplegados en la nube.

### 3.4.- OpenAI API

Aunque la plataforma Open AI y su conocido como "gimnasio" en la nube con recursos para el entrenamiento de modelos de aprendizaje automático son relativamente recientes, su API basada en el algoritmo GPT-3 ha escalado rápidamente, y se ha posicionado como una opción muy valorada para proyectos de inteligencia artificial basados en texto e interacción con las personas.

El modelo está desplegado como una herramienta de auto-completado, básicamente. Es decir, se le pasa un "prompt" que puede ser una pregunta, una instrucción o el inicio de una frase, y el modelo trata de predecir qué es lo más probable que deba aparecer a continuación. De esta manera, emula las respuestas que daría un agente, no como una respuesta realmente, sino como una forma de presentar un texto completo (pregunta + respuesta) tal y como sería si estuviesen hablando dos humanos.

Los parámetros de entrada en la API son:

- ✔ Prompt: es la más importante. Es la instrucción a partir de la cual se configura el completado o respuesta

- ✔ **Temperatura:** es el margen de "riesgo" o libertad que se le da al modelo para ser más creativo o más riguroso. Es un valor entre 0 y 1 en el que el valor de 0 representa ningún margen de libertad. Con la temperatura a 0, el modelo responde de forma muy determinista y nos dará la misma respuesta para la misma entrada.

Se puede probar el comportamiento del modelo según lo que se necesita para una aplicación concreta, en la interfaz de prueba que tiene la web, denominada "playground"

Playground

Load a preset...

Save

View code

Share

...

Dame tres nombres para una joyería que también vende tocados de lujo para mujeres:

Submit

32

Mode

Model

text-davinci-002

Temperature

0.7

Maximum length

256

Playground

Load a preset...

Save

View code

Share

...

Dame tres nombres para una joyería que también vende tocados de lujo para mujeres:

1. "Diamantes y Tocados"

2. "Lujo para Mujeres"

3. "Joyería y Tocados de Lujo"

Submit

32

Mode

Model

text-davinci-002

Temperature

0.7

Maximum length

256

De hecho, para que se pueda apreciar todo el alcance del modelo, hay un catálogo de ejemplos muy variado, y que te recomendamos revisar y probar.



#### Recipe creator (eat at your own risk)

Create a recipe from a list of ingredients.



#### Chat

Open ended conversation with an AI assist...



#### Marv the sarcastic chat bot

Marv is a factual chatbot that is also sarcas...



#### Turn by turn directions

Convert natural language to turn-by-turn dir...



#### Restaurant review creator

Turn a few words into a restaurant review.



#### Create study notes

Provide a topic and get study notes.

En la documentación se puede consultar cómo hacer las llamadas a la API desde una app o backend, pasando como parámetros principales:

- ✔ **model:** el modelo que se quiere usar. Davinci 2 es el más potente, pero también el más caro; lo recomendable es empezar con él y tras ver de lo que es capaz, ir probando con los otros modelos más especializados si logramos más o menos un buen desempeño para nuestra aplicación. Los otros modelos para NLG son Curie, Babbage y Ada. Sus características se pueden consultar aquí:

<https://platform.openai.com/docs/models/overview>

- ✔ **prompt:** la instrucción de entrada.
- ✔ **temperature:** como hemos comentado antes, el valor entre 0 y 1 de la flexibilidad creativa que le permitimos al modelo.
- ✔ **max\_tokens:** el número máximo de tokens que queremos que se generen. Los tokens son conjuntos de caracteres consecutivos que constituyen la unidad de trabajo del modelo. La relación entre tokens y palabras es de 3 a 4 (100 tokens equivalen a unas 75 palabras).

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🍌

Sequences of characters commonly found next to each other may be grouped together: 1234567890

Clear

Show example

Tokens

64

Characters

252

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🍌🍌🍌🍌🍌

Sequences of characters commonly found next to each other may be grouped together: 1234567890

TEXT

TOKEN IDS

Por ejemplo, una llamada de este estilo:



```
curl https://api.openai.com/v1/completions \  
-H "Content-Type: application/json" \  
-H "Authorization: Bearer YOUR_API_KEY" \  
-d '{"model": "text-davinci-002", "prompt": "Say this is a test", "temperature": 0, "max_token
```

Obtendría esta respuesta:

```
{  
  "id": "cmpl-GERzeJQ4lvqPk8SkZu4XMIuR",  
  "object": "text_completion",  
  "created": 1586839808,  
  "model": "text-davinci:002",  
  "choices": [  
    {  
      "text": "\n\nThis is a test",  
      "index": 0,  
      "logprobs": null,  
      "finish_reason": "length"  
    }  
  ]  
}
```