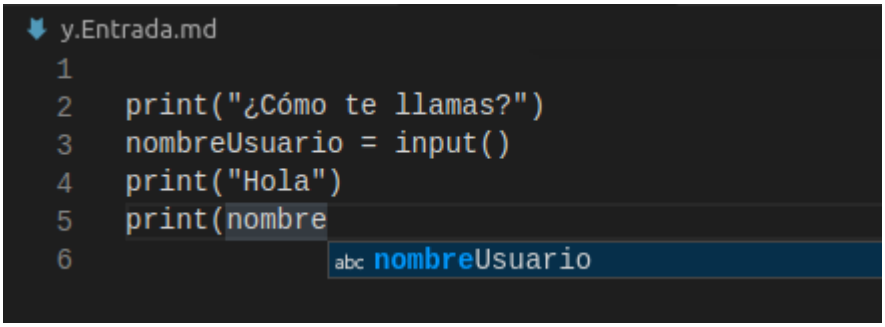
Entornos de programación para Python

Ya que estamos convencidos de la utilidad de Python, vamos a ver las diferentes formas de usarlo antes de comenzar con su instalación.

Tenemos 3 alternativas para trabajar con Python:

- Instalar **sólo el intérprete** y usar cualquier editor de texto puro (no vale un Word o similar porque introducen caracteres de formato y el resultado no lo entendería el intérprete) para escribir nuestros programas, o
- Instalar un **Entorno de desarrollo integrado** (IDE) que nos facilite el trabajo. Para empezar a programar en serio debemos usar un IDE, que nos aporta ventajas y herramientas que nos facilitan el desarrollo como: correctores, escritura predictiva, depuradores, plantillas, etc.



- Utilizar un **entorno de desarrollo web** como Replit, Jupyter Notebook o Google Colab (que es la integración y evolución de Jupyter con Google Drive) que nos permite trabajar desde un navegador web. Nuestro código se ejecutará en un servidor, en nuestra máquina con Jupyter o en la nube de Google en el caso de Colab. En ambos casos tenemos integrado un gestor de ficheros donde trabajamos y guardamos nuestros ficheros de código o el texto/documentación de nuestro proyecto.

Salvo que seas un fanático de la consola de comandos o sólo puedas acceder a tu ordenador remotamente y sin escritorio visual, la mejor opción ahora mismo sería un IDE ligero y con la funcionalidad esencial.

Vamos a empezar usando un IDE sencillo, Thonny, que tiene todo lo necesario, pero que no nos abruma con un excesivo número de menús, ni de opciones que no necesitamos.

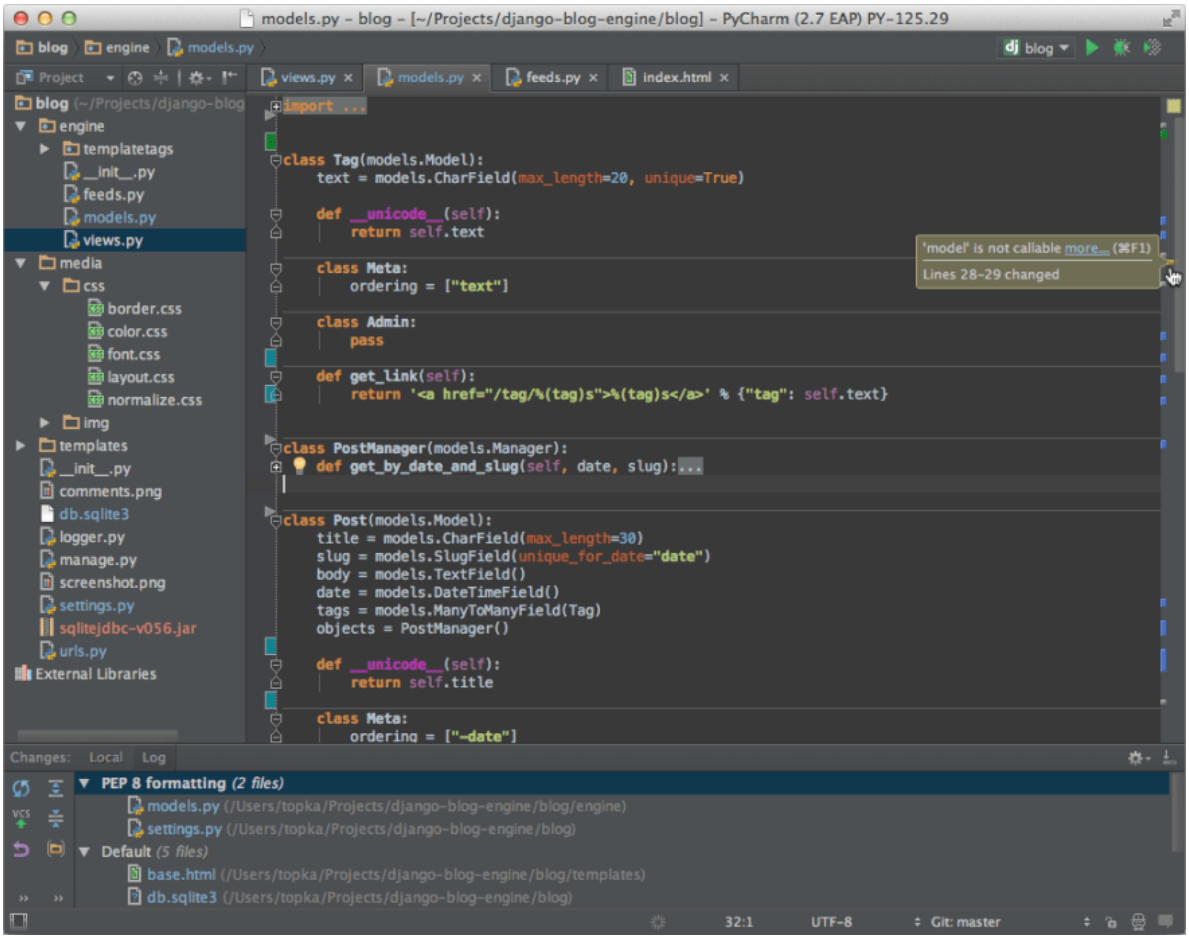
A lo largo del curso iremos instalando y aprendiendo a usar otros IDEs más avanzados, a medida que los vayamos necesitando.

Son muchos los IDEs disponibles, veamos algunos de ellos.

PyCharm

A día de hoy, PyCharm, desarrollado por JetBrains, es un IDE potente y completo, y es el más usado a nivel profesional para trabajar con Python. Como corresponde a una opción profesional, es de pago, aunque podemos usar una versión gratuita (con menos opciones, eso sí).

Actualmente cuando lo descargas e instalas comienzas una prueba de la versión Pro de 30 días y pasado ese tiempo si no lo pagas, se desactivan las opciones Pro.



Características Principales

- **Soporte avanzado**: Autocompletado inteligente, refactorización, depuración, integración con Git, soporte para Django, Flask, y bases de datos.
- **Análisis de código**: Detección de errores en tiempo real, sugerencias de optimización.
- **Integración**: Virtualenv, Conda, Docker, y herramientas de testing (pytest, unittest).
- **Personalización**: Temas, plugins extensos, y soporte multiplataforma (Windows, macOS, Linux).

Para Profesionales

- **Ventajas**:
    - Ideal para proyectos grandes y complejos (web, ciencia de datos, machine learning).
    - Herramientas avanzadas para colaboración en equipo (Git, revisiones de código).
    - Soporte para frameworks empresariales (Django, FastAPI).
    - Depuración robusta y estudio de rendimiento (profiling).
- **Desventajas**:

    - Consumo elevado de recursos (RAM, CPU), especialmente en la edición Professional.
    - Curva de aprendizaje inicial para configuraciones avanzadas.
- **Recomendado para**: Desarrolladores de software, ingenieros de datos, y equipos trabajando en proyectos a gran escala.

Para No Profesionales

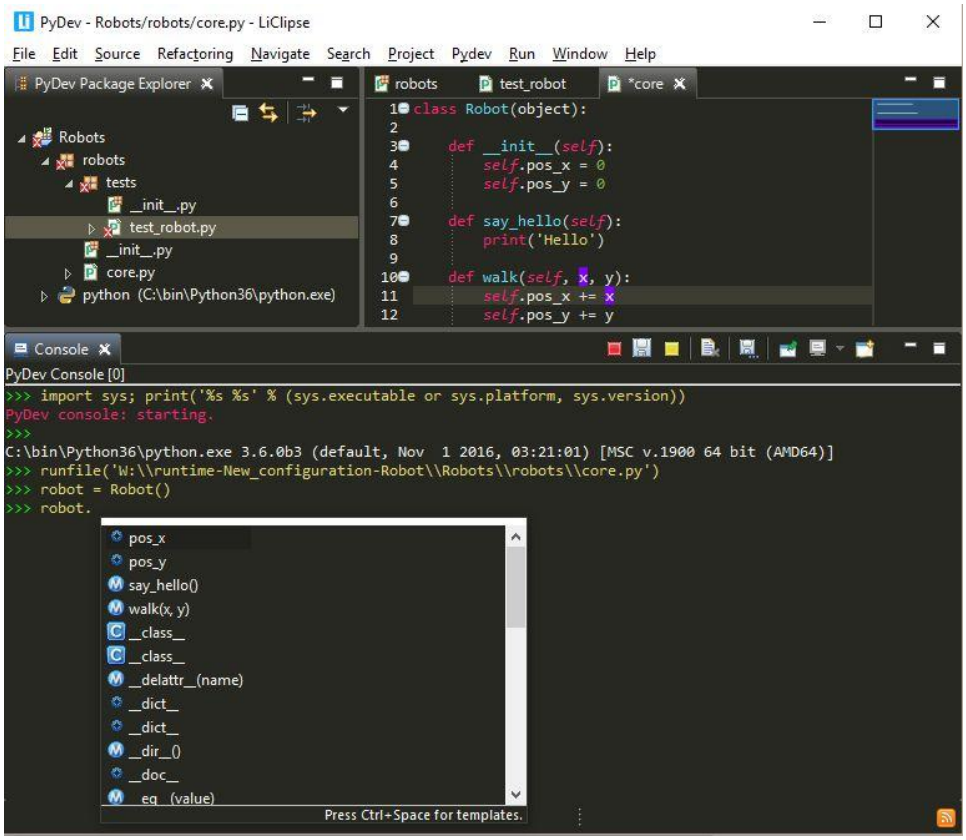
- **Ventajas**:
    - La edición Community es gratuita y suficiente para proyectos pequeños o aprendizaje.
    - Interfaz amigable con guías contextuales.
- **Desventajas**:
    - Puede resultar abrumador para principiantes debido a la cantidad de opciones.
    - Requiere configuración inicial (entornos virtuales, intérpretes).
- **Recomendado para**: Estudiantes avanzados o aficionados que quieran un entorno completo para aprender.

Precio

- Community: Gratuito.
- Professional: Suscripción (~$199/año, con descuentos para estudiantes).

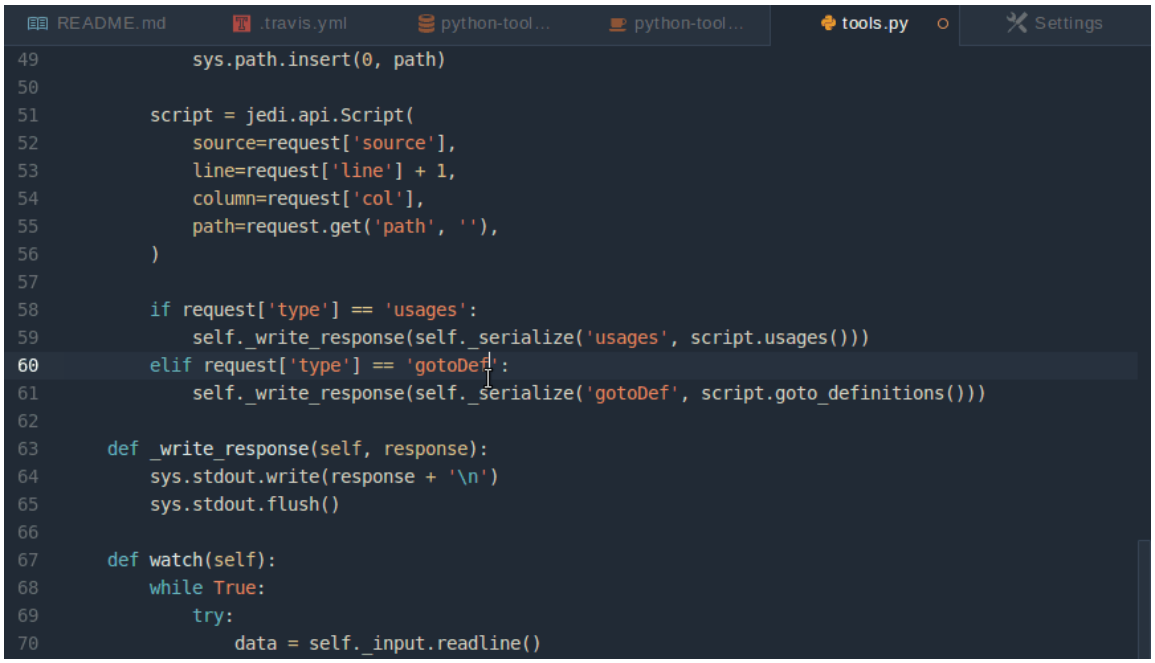
PyDev

Es un complemento para el entorno de programación Eclipse que nos permite programar Python, al igual que Eclipse, es un IDE gratuito.



Visual Studio Code

VS Code, creado por Microsoft, es un excelente IDE multilenguaje, multientorno y con miles de extensiones/complementos desarrollados por Microsoft y por terceros. Al ser multilenguaje es una elección natural para los que trabajamos con varios lenguajes.



En mi caso lo uso para trabajar con Arduino, Python y para editar documentación (las primeras versiones del material de este curso se editaron con él).

Es mi elección para el desarrollo con distintos lenguajes y por las muchas herramientas, pero quizás un poco difícil para empezar. Lo usaremos a partir de la mitad del curso, cuando los proyectos empiecen a ser más complejos.

Características Principales

- **Extensiones**: Extensión oficial de Python (Microsoft) con soporte para IntelliSense, depuración, Jupyter Notebooks, y linters (Pylance, Flake8).
- **Ligereza**: Bajo consumo de recursos, rápido incluso en máquinas modestas.
- **Integración**: Git, Docker, y soporte para múltiples lenguajes.
- **Personalización**: Temas, atajos de teclado, y configuración flexible.

Para Profesionales

- **Ventajas**:
    - Altamente personalizable para flujos de trabajo específicos.
    - Excelente para desarrollo colaborativo (Live Share) y DevOps.
    - Compatible con ciencia de datos (Jupyter) y desarrollo web.
    - Comunidad activa y gran cantidad de extensiones.
- **Desventajas**:
    - Requiere configuración inicial, extensiones, linters (verificadores de estilo), entornos virtuales.
    - Menos integrado que PyCharm para proyectos complejos de Python.
- **Recomendado para**: Desarrolladores que trabajan en múltiples lenguajes o prefieren un entorno ligero y modular.

Para No Profesionales

- **Ventajas**:
    - Gratuito y fácil de instalar.
    - Interfaz limpia y moderna, ideal para principiantes.
    - Tutoriales y extensiones que facilitan el aprendizaje.
- **Desventajas**:
    - Configurar entornos de Python puede ser complicado para novatos.
    - Dependencia de extensiones para funcionalidades avanzadas.
- **Recomendado para**: Estudiantes y aficionados que buscan un entorno ligero y versátil.

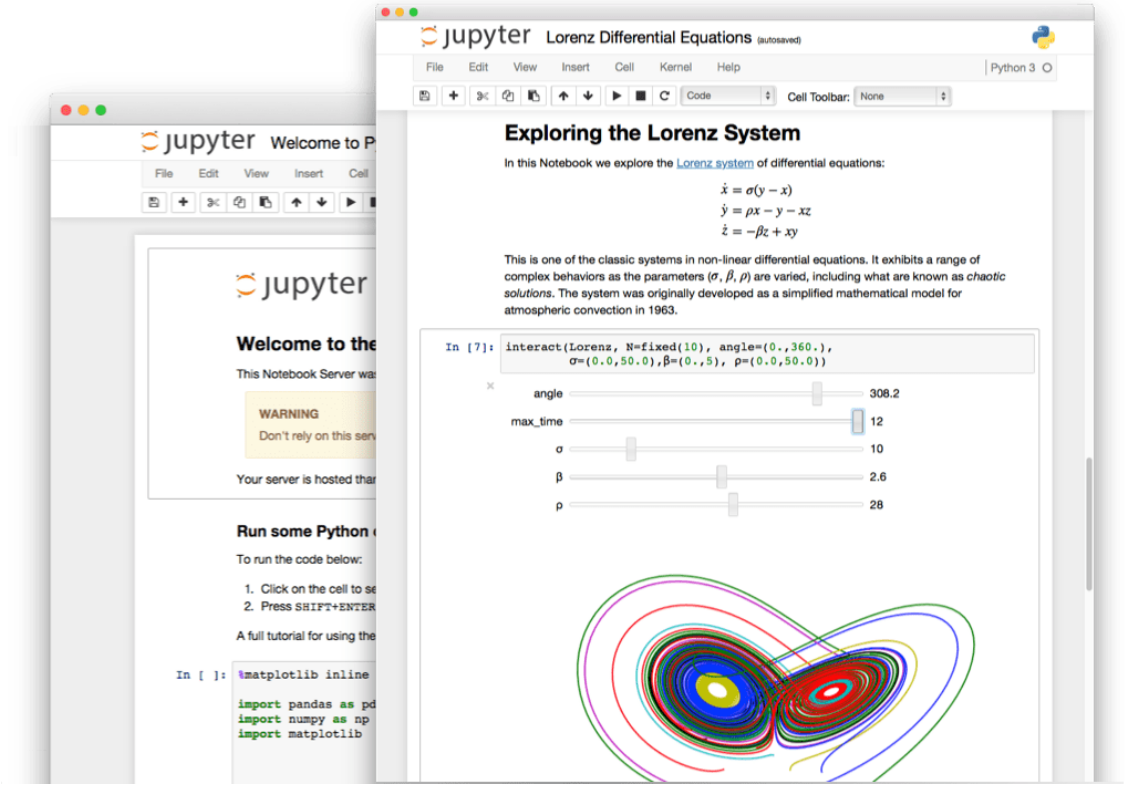
Precio

- Gratuito.

Si eres muy fanático del Open-Source, existe una alternativa que lo es al 100%, VSCodium que sólo integra la parte opensource liberada por Microsoft, sin usar algunas librerías propietarias que Microsoft utiliza para obtener información del uso que hacemos.

Jupyter Notebook

Jupyter Notebook es una aplicación enfocada en trabajar en un solo documento a la vez mientras JupyterLab es una aplicación Web mucho más avanzada, moderna y más adecuado para proyectos complejos o colaborativos la anterior (la ejecutamos en nuestra máquina pero sobre un navegador web), diseñada para crear al mismo tiempo documentación y código pensando sobre todo en compartirlo posteriormente. Resulta ideal para ciencia de datos, machine learning y visualización de datos.



Como se puede ver en la imagen está muy orientada hacia el mundo científico. La usaremos en los últimos temas del curso, sobre todo para el tema de gráficos, su entorno natural.

Características Principales

- **Interactividad**: Celdas de código, markdown, y visualizaciones integradas.
- **Soporte para ciencia de datos**: Integración con NumPy, pandas, matplotlib, y TensorFlow.
- **JupyterLab**: Interfaz más moderna con soporte para múltiples archivos y extensiones.
- **Colaboración**: Fácil de compartir (notebooks en GitHub, Google Colab).

Para Profesionales

- **Ventajas**:
    - Ideal para prototipado rápido, análisis de datos y machine learning.

    - Integración con herramientas como Dask, Spark, y cloud (AWS, Google Cloud).
    - Soporte para kernels múltiples (R, Julia, etc.).
- **Desventajas**:
    - No es ideal para desarrollo de aplicaciones o proyectos estructurados.
    - Control de versiones limitado (Al usar un formato JSON es complicado gestionar los cambios).
- **Recomendado para**: Científicos de datos, ingenieros de machine learning, y analistas.

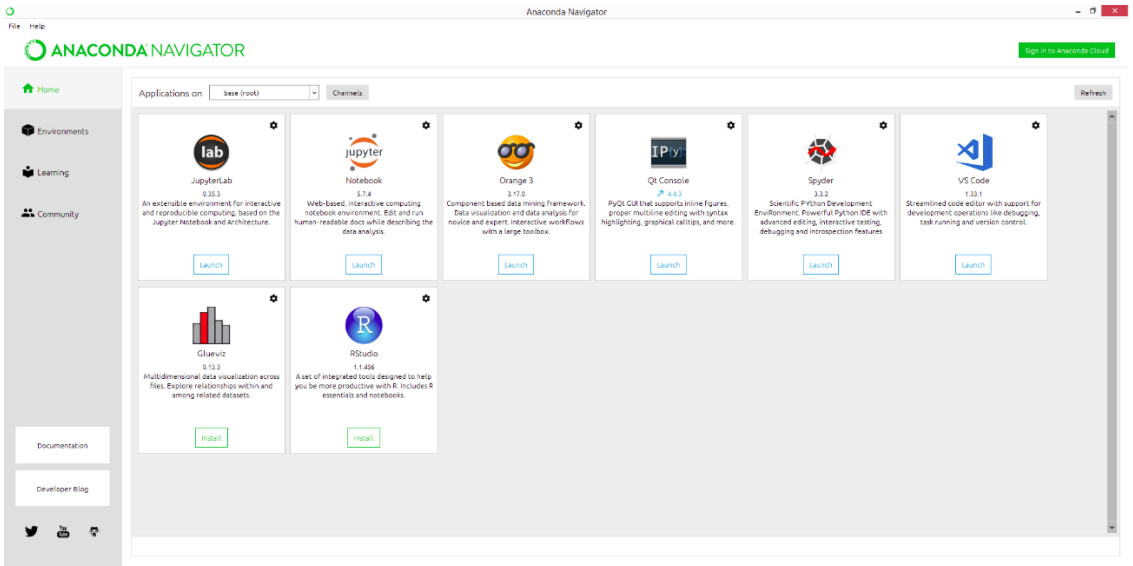
Para No Profesionales

- **Ventajas**:
    - Fácil de usar para aprender Python, especialmente en ciencia de datos.
    - Interfaz intuitiva para experimentar con código y visualizaciones.
    - Disponible en plataformas en línea como Google Colab (gratuito).
- **Desventajas**:
    - No es adecuado para aprender programación estructurada o desarrollo de software.
    - Requiere conocimientos básicos de Python para aprovecharlo.
- **Recomendado para**: Estudiantes de ciencia de datos o principiantes interesados en análisis de datos.

Precio

- Gratuito (open-source).

[Anaconda]



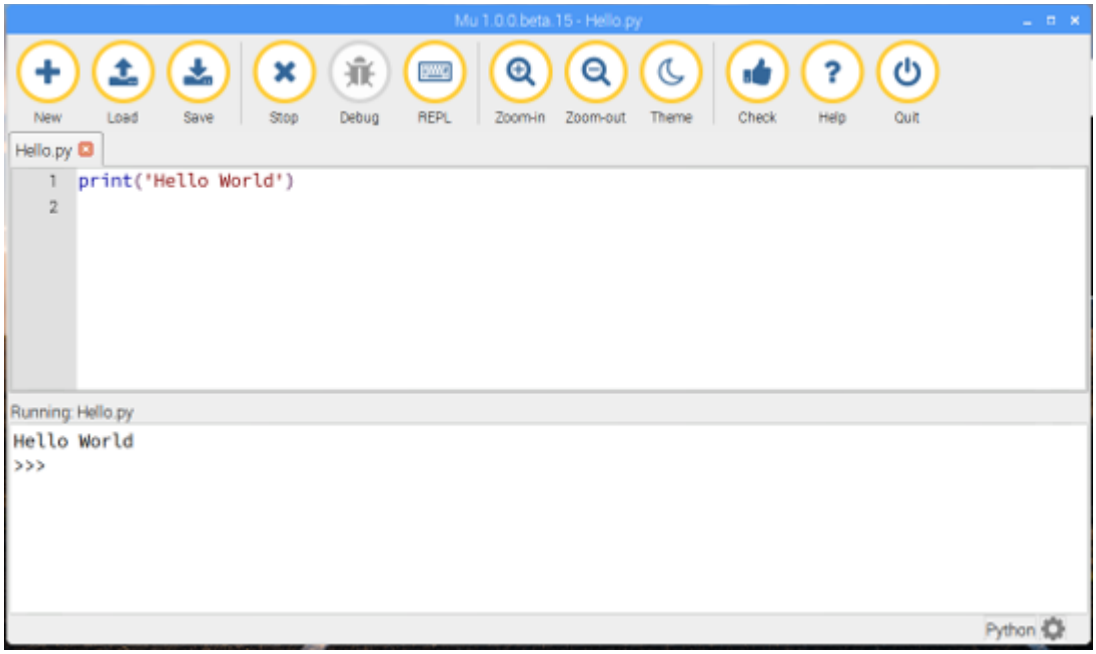
Anaconda es una distribución gratuita y de código abierto de Python y R ,diseñada para simplificar la gestión de paquetes y entornos, especialmente para ciencia de datos, aprendizaje automático, y análisis de datos.

Incluye Python y R (lenguaje de programación orientado a los trabajas con estadística), Conda, un administrador de paquetes y entornos y muchas Librerías populares, como NumPy, pandas, Matplotlib, Jupyter Notebook, y más.

Es ideal para principiantes y profesionales que desean un entorno preconfigurado para análisis de datos.

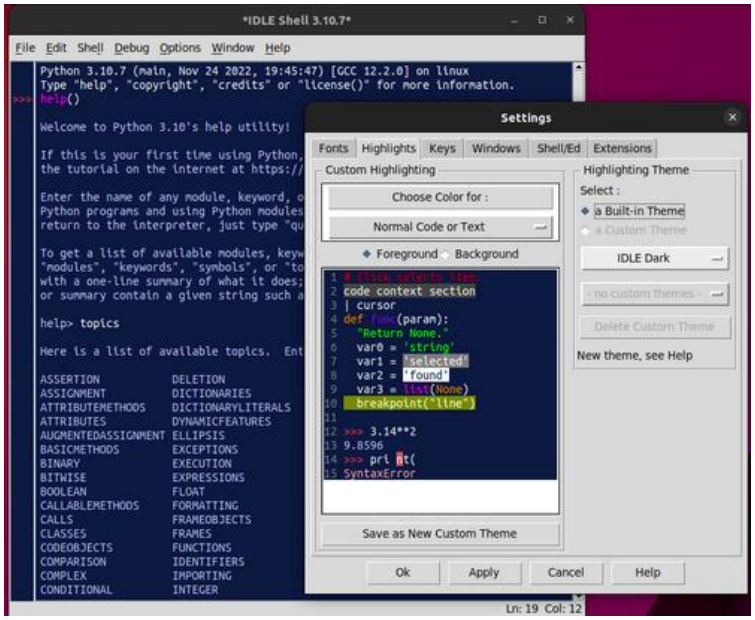
Mu

Es algo más que un editor pero mucho menos que un IDE. Se puede usar como iniciación, pero enseguida se queda corto. Sí que es cómodo para trabajar sistemas embebidos, tipo micro:bit o ESP32 con Python.



Está escrito en Python, lo cual es un plus.

IDLE



IDLE es el entorno de desarrollo integrado que viene preinstalado con Python. Es simple y está diseñado para principiantes.

Características Principales

- **Simplicidad**: Editor básico con resaltado de sintaxis y shell interactivo.
- **Ligereza**: Consumo mínimo de recursos.
- **Integrado**: No requiere instalación adicional si Python está instalado.

Para Profesionales

- **Ventajas**:
    - Útil para pruebas rápidas o scripts pequeños.
    - No requiere configuración.
- **Desventajas**:
    - Carece de herramientas avanzadas (depuración, refactorización, integración con Git).
    - No es adecuado para proyectos grandes o colaborativos.
- **Recomendado para**: Uso ocasional o entornos con recursos limitados.

Para No Profesionales

- **Ventajas**:
    - Ideal para principiantes absolutos por su simplicidad.
    - Incluido con Python, sin necesidad de instalar software adicional.
- **Desventajas**:
    - Funcionalidades muy limitadas.

    - Interfaz anticuada y poco personalizable.
- **Recomendado para**: Principiantes que están aprendiendo los fundamentos de Python.

Precio

- Gratuito (incluido con Python).

Thonny

Thonny es un IDE ligero, diseñado específicamente para principiantes, con un enfoque en la enseñanza de Python. Es el entorno elegido para la primera parte del curso.

Características Principales

- **Depuración sencilla**: Visualización paso a paso de variables y ejecución.
- **Gestión de entornos**: Instalación y gestión de paquetes simplificada.
- **Interfaz amigable**: Resaltado de sintaxis y errores para principiantes.

Para Profesionales

- **Ventajas**:
    - Útil con fines educativos debido a su simplicidad.
    - Bajo consumo de recursos.
- **Desventajas**:
    - No es adecuado para proyectos complejos o desarrollo profesional.
    - Falta de herramientas avanzadas (integración con Git, soporte para frameworks).
- **Recomendado para**: Profesores o desarrolladores que enseñan Python.

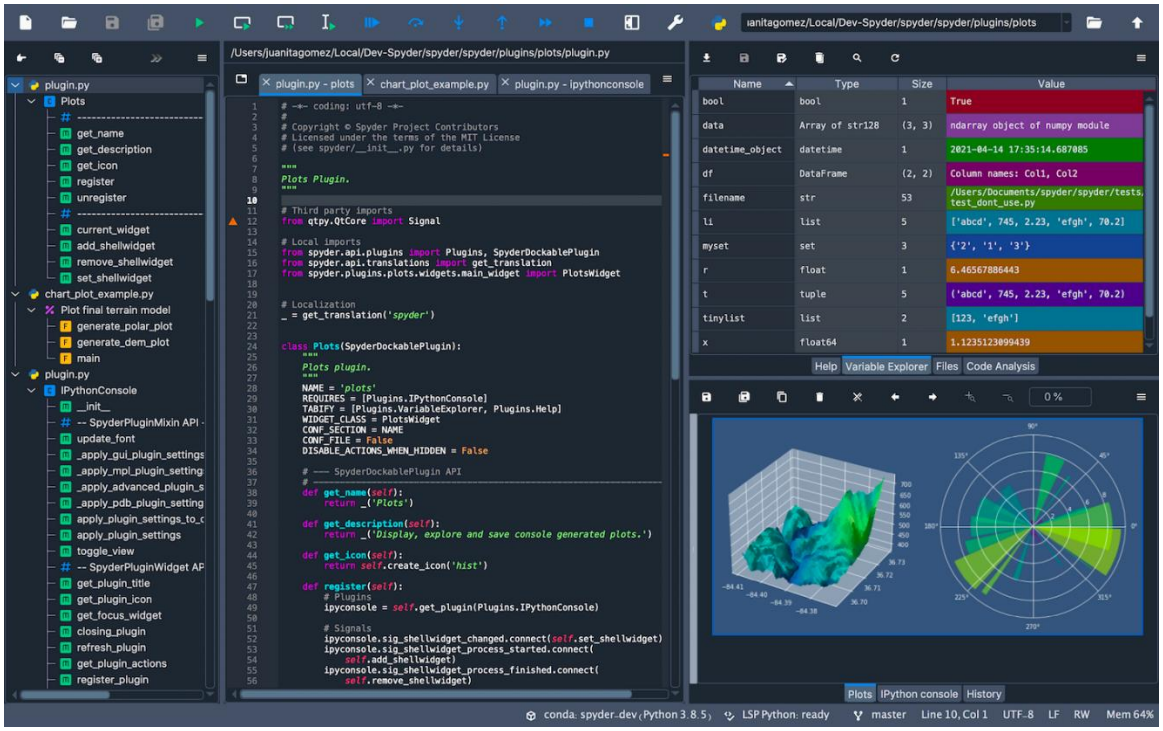
Para No Profesionales

- **Ventajas**:
    - Excelente para principiantes por su interfaz clara y herramientas de depuración.
    - Gestión automática de entornos virtuales.
- **Desventajas**:
    - Limitado para proyectos más allá del aprendizaje básico.
    - Menos soporte para extensiones o personalización.
- **Recomendado para**: Estudiantes y principiantes que quieran un entorno educativo.

Precio

- Gratuito.

[Spyder]



Spyder (Scientific Python Development Environment) es un IDE de código abierto diseñado específicamente para ciencia de datos y programación científica en Python. Integrado en la distribución Anaconda, es popular entre científicos y analistas de datos.

Características Principales

- **Enfoque científico**: Integración nativa con bibliotecas como NumPy, pandas, SciPy, y matplotlib.
- **Explorador de variables**: Permite inspeccionar y editar datos en tiempo real.
- **Consola IPython**: Ejecución interactiva de código y soporte para Jupyter Notebooks.
- **Depuración**: Usa el depurador PDB con breakpoints y exploración de variables.
- **Interfaz**: Similar a MATLAB y RStudio, con paneles para código, consola y gráficos.

Para Profesionales

- **Ventajas**:
    - Ideal para ciencia de datos, análisis estadístico y visualización.
    - Ligero en comparación con PyCharm, con menor consumo de recursos (~200-300 MB).
    - Explorador de variables para inspeccionar grandes volúmenes de datos (DataFrames), útil en machine learning.
    - Integración con Git y soporte para pruebas unitarias (pytest, unittest).
- **Desventajas**:

    - Menos robusto que PyCharm para proyectos no científicos (p. ej., desarrollo web).
    - Limitado a Python, sin soporte nativo para otros lenguajes.
    - Puede ser lento con grandes conjuntos de datos o al iniciar.
- **Recomendado para**: Científicos de datos, ingenieros y analistas que trabajan con bibliotecas científicas.

Para No Profesionales

- **Ventajas**:
    - Interfaz intuitiva, especialmente para usuarios de MATLAB o RStudio.
    - Fácil de usar para principiantes en ciencia de datos gracias al explorador de variables.
    - Gratuito y preinstalado con Anaconda, sin configuraciones complejas.
- **Desventajas**:
    - Curva de aprendizaje para usuarios sin experiencia en Python o ciencia de datos.
    - Menos adecuado para aprender programación general o estructurada.
- **Recomendado para**: Estudiantes de ciencia de datos o principiantes interesados en análisis y visualización.

Precio

- Gratuito (open-source, la parte de Anaconda).

Otros Editores

Existen otros editores que admiten complementos para trabajar con Python (y con otros lenguajes) como pueden ser Atom o Sublime Text, este último de pago, ofrece una versión gratuita con limitaciones.

Entornos online

Existen páginas que nos permiten trabajar con un intérprete de Python (y de más lenguajes) online.

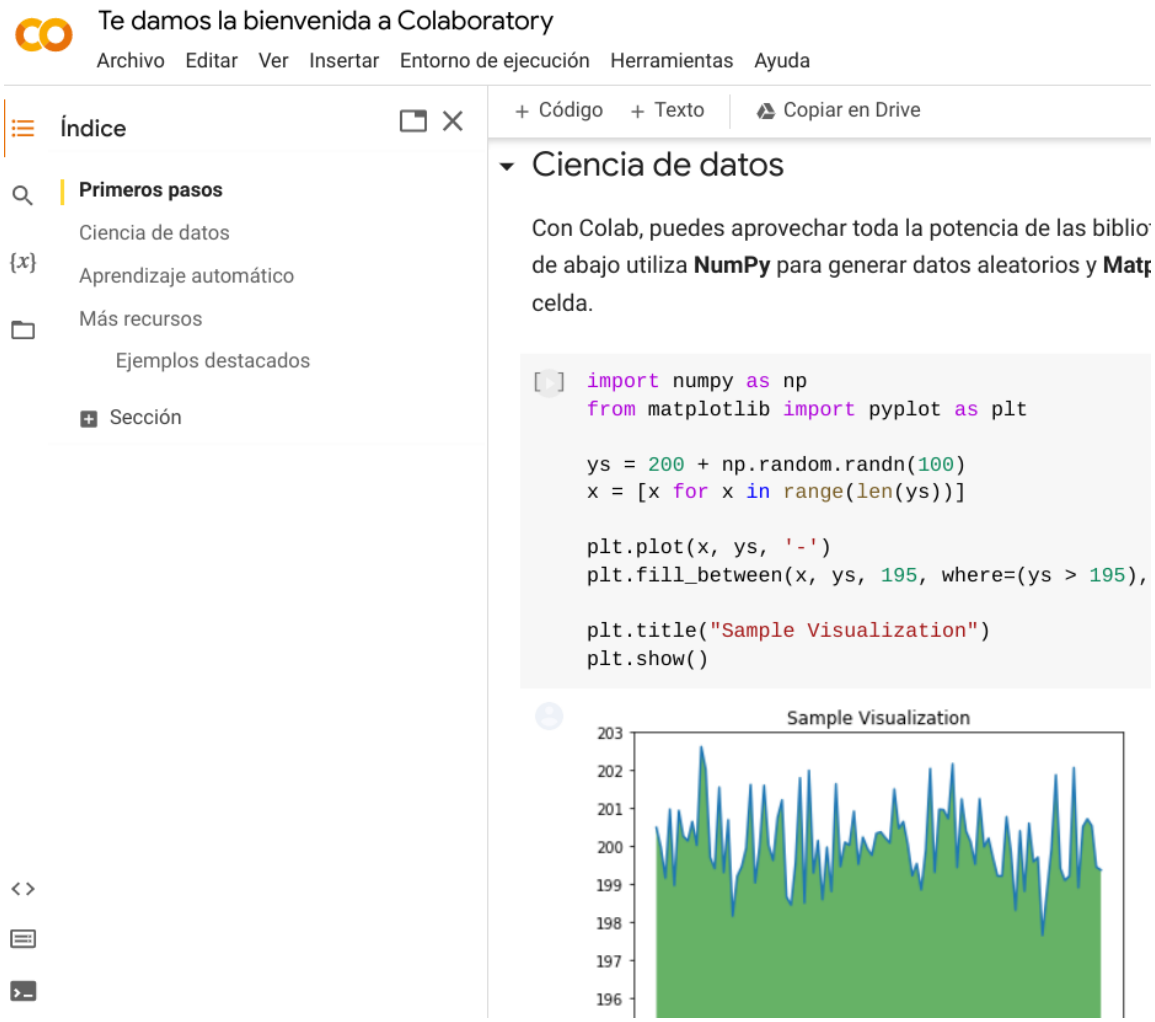
Son cómodos de usar y pueden ayudarnos a resolver algún problema puntual de acceso o para tomar contacto con el sistema.

Otra página donde podemos practicar con nuestro código es create.withcode

Google Colab

Google Colab es una adaptación de Jupyter Notebook por parte de Google y su integración con Google Drive, nos permite compartir fácilmente con otros usuarios y también utilizar toda su infraestructura de la nube. Su nombre viene de Colaboratory lo que nos muestra claramente su vocación de trabajar colaborativamente.

Al igual que Jupyter Notebook, nos permite mezclar texto y código de manera que podemos trabajar en un artículo o en documento interactivo y explicar, programar, calcular y al mismo tiempo, presentar los resultados. También integra herramientas de procesado, visualización de datos y en general data mining, incluso de aprendizaje automático o machine learning.



Otra gran ventaja de utilizar Google Colab es que nos permite crear un entorno de ejecución totalmente personalizado para nuestro código Python, independiente de nuestro ordenador. Así podemos instalar componentes o módulos que necesitemos, teniendo acceso a un entorno virtual donde se está ejecutando nuestro código.

Los cuadernos de Colab ejecutan código en los servidores en la nube de Google, lo que te permite aprovechar toda la potencia del hardware de Google, independientemente de la potencia de nuestro equipo

Un ejemplo típico de uso de Colab, donde realmente aprovechamos muchas de sus posibilidades, sería usar un documento donde ya tenemos preparado todo lo necesario para trabajar con un sistema de inteligencia artificial, con la instalación de las librerías necesarias, el código para usarlo, incluso todas aquellas imágenes u objetos que queremos que nuestra inteligencia artificial utilice.

De esa forma, cuando compartimos nuestro documento con texto y con código, se está compartiendo la instalación, los ficheros, las librerías, imágenes, etc, en definitiva todo lo que necesitamos para trabajar, aprovechándose de la infraestructura de Google Drive.

Otra enorme ventaja es que estamos también delegando en los servidores de la nube de Google todo el procesamiento, nuestro ordenador no tiene que realizar aquí ese procesamiento sino que se hace todo directamente en los ordenadores de la nube de Google.

Características Principales

- **Accesibilidad**: Funcionan en cualquier navegador, sin instalación.
- **Colaboración**: Fácil compartir proyectos.
- **Google Colab**: Integración con Google Drive, soporte para GPU/TPU.

Para Profesionales

- **Ventajas**:
    - Ideal para prototipos rápidos o pruebas en la nube.
    - Google Colab es potente para machine learning con acceso a GPUs gratuitas.
- **Desventajas**:
    - Dependencia de conexión a internet.
    - Limitaciones en recursos y personalización en planes gratuitos.
- **Recomendado para**: Científicos de datos.

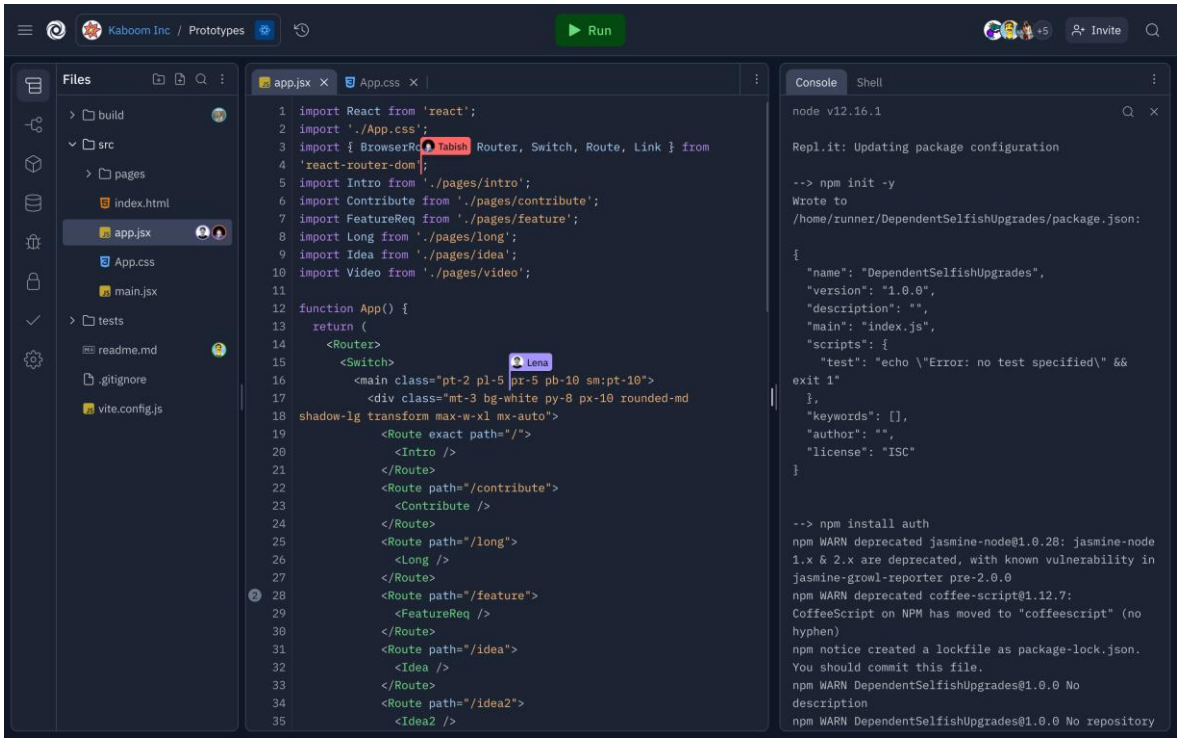
Para No Profesionales

- **Ventajas**:
    - Sin necesidad de configurar entornos locales.
    - Interfaz intuitiva y acceso inmediato.
    - Google Colab es ideal para aprender ciencia de datos.
- **Desventajas**:
    - Menos control sobre el entorno (versiones de Python, paquetes).
    - Algunas funcionalidades avanzadas requieren suscripciones.
- **Recomendado para**: Principiantes que quieran experimentar sin instalar software.

Precio

- Google Colab: Gratuito (Pro/Pro+ con costos ~$9.99-$49.99/mes).

Replit



Plataformas basadas en la nube como Replit permiten programar en Python sin instalar software localmente.

Características Principales

- **Accesibilidad**: Funcionan en cualquier navegador, sin instalación.
- **Colaboración**: Fácil compartir proyectos.
- **Replit**: Entorno multi-lenguaje con soporte para despliegue de aplicaciones.

Para No Profesionales

- **Ventajas**:
    - Sin necesidad de configurar entornos locales.
    - Interfaz intuitiva y acceso inmediato.
- **Desventajas**:
    - Menos control sobre el entorno (versiones de Python, paquetes).
    - Algunas funcionalidades avanzadas requieren suscripciones.
- **Recomendado para**: Principiantes que quieran experimentar sin instalar software.

Precio

- Replit: Gratuito con limitaciones y (planes premium ~$7-$20/mes).

Resumen y Recomendaciones

Entorno	Profesionales	No Profesionales	Casos de Uso Principales
PyCharm	Proyectos grandes, desarrollo web, ML	Aprendizaje avanzado	Desarrollo profesional, proyectos complejos
VS Code	Flujos personalizados, multi-lenguaje	Aprendizaje versátil	Desarrollo general, ciencia de datos
Jupyter	Ciencia de datos, ML, prototipado	Aprendizaje de datos	Análisis de datos, visualización
IDLE	Pruebas rápidas, scripts simples	Primeros pasos en Python	Aprendizaje básico
Thonny	Enseñanza, tutorías	Principiantes absolutos	Educación, aprendizaje inicial
Spyder	Ciencia de datos, análisis, visualización	Aprendizaje de ciencia de datos	Programación científica, análisis de datos
Plataformas en línea	Prototipado, colaboración, ML (Colab)	Experimentación sin instalación	Proyectos rápidos, aprendizaje en la nube

Conclusión

- **Profesionales**: PyCharm (Professional) es la mejor opción para proyectos complejos y desarrollo full-stack, mientras que VS Code es ideal para flujos de trabajo personalizados. Spyder destaca en ciencia de datos y entornos ligeros, siendo una alternativa eficiente a Jupyter para proyectos científicos estructurados. Jupyter es imprescindible para prototipado y análisis de datos.
- **No Profesionales**: Thonny y IDLE son ideales para principiantes absolutos. Spyder es excelente para estudiantes de ciencia de datos debido a su explorador de variables y similitud con MATLAB/RStudio. VS Code y plataformas en línea como Google Colab son adecuadas para estudiantes que avanzan hacia proyectos más complejos.
- **Tendencias 2025**: La integración de herramientas de IA (como GitHub Copilot en VS Code o PyCharm) y el auge de plataformas en la nube (Colab, Replit) están transformando el desarrollo en Python, haciendo que la programación sea más accesible y colaborativa. Spyder sigue siendo una opción sólida para ciencia de datos, especialmente para usuarios que prefieren un entorno más tradicional que Jupyter.