

# Entornos de Programación para Python



Septiembre 2025

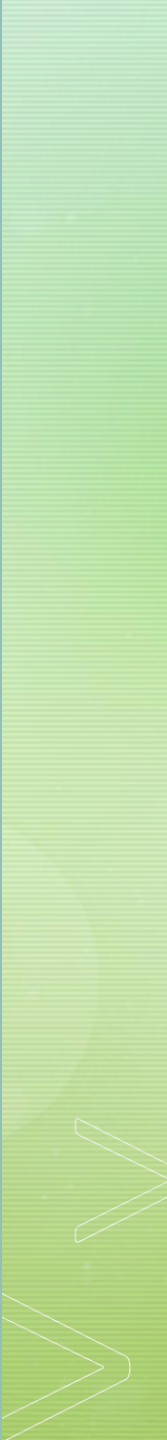
Juan Romero Sánchez

IES Trassierra





# Introducción

- Python es un lenguaje versátil y ampliamente utilizado.
  - Antes de programar, debemos conocer las opciones de entornos disponibles.
- 

# Tipos de Entornos

| Alternativas para Comenzar a Programar en Python |  |
|--|--|
| Opción 1   | <b>Intérprete y Editor de Texto Puro:</b> Solo se instala el intérprete. Se usa un editor de texto simple (ej: Notepad, no Word). Recomendado para fanáticos de la consola o acceso remoto sin escritorio.                       |
| Opción 2   | <b>Entorno de Desarrollo Integrado (IDE):</b> La mejor opción para empezar a programar en serio. Ofrecen herramientas que facilitan el desarrollo: correctores, depuradores, escritura predictiva, etc.                          |
| Opción 3   | <b>Entorno de Desarrollo Web:</b> Permite trabajar desde un navegador. El código se ejecuta en un servidor (en la nube o localmente). Ideal para colaboración y acceso simple. Ejemplos: Replit, Jupyter Notebook, Google Colab. |
| Recomendación Inicial                            | Empezar con un <b>IDE ligero y esencial</b> como <b>Thonny</b> .   |

# IDEs Populares

## PyCharm (El Gigante Profesional)

| PyCharm: El Estándar Profesional |  |
|----------------------------------|--|
| Desarrollador                    | JetBrains  |
| Características Clave            | <ul style="list-style-type: none"><li>- <b>Soporte Avanzado:</b> Autocompletado inteligente, refactorización, depuración robusta, integración con Git/Docker.</li><li>- <b>Análisis de Código:</b> Detección de errores en tiempo real y sugerencias de optimización.</li><li>- <b>Personalización:</b> Temas, <i>plugins</i> extensos, multiplataforma.</li></ul> |
| Ventajas (Pro)                   | Ideal para proyectos grandes/complexos (web, ML). Herramientas avanzadas para colaboración.  |
| Desventajas (Pro)                | Alto consumo de recursos (RAM/CPU). Curva de aprendizaje inicial.  |
| Versiones y Precio               | <b>Community</b> (Gratuita) para aprendizaje/proyectos pequeños. <b>Professional</b> (Suscripción) para uso profesional.   |

# IDEs Populares

## Visual Studio Code (VS Code): El Editor Multilenguaje

| Visual Studio Code (VS Code): El Editor Multilenguaje |   |
|---|---|
| Desarrollador   | Microsoft   |
| Características Clave                                 | <ul style="list-style-type: none"><li>- <b>Extensiones:</b> Gran ecosistema de complementos (oficial de Python, IntelliSense, Jupyter). –</li><li>- <b>Ligereza:</b> Bajo consumo de recursos, rápido.</li><li>- <b>Integración:</b> Git, Docker, soporte para múltiples lenguajes (ideal para proyectos mixtos).</li></ul> |
| Ventajas (General)                                    | Altamente personalizable. Excelente para desarrollo colaborativo (Live Share) y DevOps. Gratuito.   |
| Desventajas (General)                                 | Requiere configuración inicial (extensiones, linters, entornos virtuales). Menos integrado nativamente que PyCharm para Python puro.  |
| Recomendado   | Desarrolladores que usan <b>múltiples lenguajes</b> o prefieren un entorno <b>ligero y modular</b> .<br><b>VSCodium</b> es la alternativa 100% <i>Open-Source</i> .   |



# IDEs Populares

## Visual Studio Code (VS Code): El Editor Multilenguaje

| Visual Studio Code (VS Code): El Editor Multilenguaje |   |
|---|---|
| Desarrollador   | Microsoft   |
| Características Clave                                 | <ul style="list-style-type: none"><li>- <b>Extensiones:</b> Gran ecosistema de complementos (oficial de Python, IntelliSense, Jupyter). –</li><li>- <b>Ligereza:</b> Bajo consumo de recursos, rápido.</li><li>- <b>Integración:</b> Git, Docker, soporte para múltiples lenguajes (ideal para proyectos mixtos).</li></ul> |
| Ventajas (General)                                    | Altamente personalizable. Excelente para desarrollo colaborativo (Live Share) y DevOps. Gratuito.   |
| Desventajas (General)                                 | Requiere configuración inicial (extensiones, linters, entornos virtuales). Menos integrado nativamente que PyCharm para Python puro.  |
| Recomendado   | Desarrolladores que usan <b>múltiples lenguajes</b> o prefieren un entorno <b>ligero y modular</b> .<br><b>VSCodium</b> es la alternativa 100% <i>Open-Source</i> .   |

# IDEs Populares

## Entornos para Ciencia de Datos: Jupyter y Spyder

| Entornos para Ciencia de Datos: Jupyter y Spyder |  |
|--|--|
| Jupyter Notebook / JupyterLab                    | <ul style="list-style-type: none"><li>- <b>Enfoque:</b> Interactivo y documentado. Ideal para prototipado, análisis de datos y ML.</li><li>- <b>Características:</b> Celdas de código y texto (Markdown), visualizaciones integradas.</li><li>- <b>Ventaja:</b> Ideal para compartir (GitHub, Colab).</li></ul>  |
| Spyder   | <ul style="list-style-type: none"><li>– <b>Enfoque:</b> Programación científica. Integrado en la distribución Anaconda.</li><li>– <b>Características:</b> Interfaz similar a MATLAB/RStudio.</li><li>– <b>Explorador de variables</b> en tiempo real (crucial en ciencia de datos).</li><li>– <b>Ventaja:</b> Ligero, excelente para análisis estadístico y visualización.</li></ul> |
| Anaconda (Distribución)                          | Distribución gratuita de Python y R. Incluye <b>Conda</b> (gestor de paquetes/entornos) y librerías clave (NumPy, pandas). Ideal para preconfigurar entornos para ciencia de datos.  |

# IDEs Populares

## Entornos para Principiantes y Educativos

| Opciones para Principiantes y Educación |   |
|---|---|
| Thonny                                  | <ul style="list-style-type: none"><li>- <b>Puede ser el ideal para Empezar el Curso</b></li><li>- <b>Enfoque:</b> Diseñado para la enseñanza.</li><li>- <b>Características:</b> Depuración sencilla (visualización paso a paso), gestión de entornos simplificada, interfaz amigable.</li><li>- <b>Recomendado:</b> Principiantes absolutos y educadores.</li></ul> |
| IDLE                                    | <ul style="list-style-type: none"><li>– <b>El Básico Integrado</b></li><li>– <b>Enfoque:</b> Viene preinstalado con Python.</li><li>– <b>Características:</b> Editor simple con resaltado de sintaxis y <i>shell</i> interactivo.</li><li>– <b>Uso:</b> Pruebas rápidas o primeros pasos, carece de herramientas avanzadas.</li></ul>                               |
| Mu                                      | <ul style="list-style-type: none"><li>– <b>El Editor Mínimo</b></li><li>– <b>Enfoque:</b> Simple, más que un editor, menos que un IDE.</li><li>– <b>Uso:</b> Ideal para iniciación y sistemas embebidos (micro:bit, ESP32).</li></ul>   |



# IDEs Populares

## Entornos para Principiantes y Educativos

| Opciones para Principiantes y Educación |   |
|---|---|
| Thonny                                  | <ul style="list-style-type: none"><li>- <b>Puede ser el ideal para Empezar el Curso</b></li><li>- <b>Enfoque:</b> Diseñado para la enseñanza.</li><li>- <b>Características:</b> Depuración sencilla (visualización paso a paso), gestión de entornos simplificada, interfaz amigable.</li><li>- <b>Recomendado:</b> Principiantes absolutos y educadores.</li></ul> |
| IDLE                                    | <ul style="list-style-type: none"><li>– <b>El Básico Integrado</b></li><li>– <b>Enfoque:</b> Viene preinstalado con Python.</li><li>– <b>Características:</b> Editor simple con resaltado de sintaxis y <i>shell</i> interactivo.</li><li>– <b>Uso:</b> Pruebas rápidas o primeros pasos, carece de herramientas avanzadas.</li></ul>                               |
| Mu                                      | <ul style="list-style-type: none"><li>– <b>El Editor Mínimo</b></li><li>– <b>Enfoque:</b> Simple, más que un editor, menos que un IDE.</li><li>– <b>Uso:</b> Ideal para iniciación y sistemas embebidos (micro:bit, ESP32).</li></ul>   |

# IDEs Populares

## Entornos de Desarrollo Web/Online

| Entornos de Desarrollo Web/Online |  |
|-----------------------------------|--|
| Google Colab                      | <ul style="list-style-type: none"><li>- <b>El Notebook en la Nube de Google</b></li><li>- <b>Base:</b> Adaptación y evolución de Jupyter Notebook.</li><li>- <b>Ventajas Clave:</b> Integración con Google Drive. Soporte para <b>GPU/TPU gratuitas</b> (ideal para Machine Learning). Permite compartir entornos personalizados completos (código, librerías, datos).</li><li>- <b>Recomendado:</b> Colaboración, ML, uso de infraestructura potente sin depender del equipo local.</li></ul> |
| Replit                            | <ul style="list-style-type: none"><li>– <b>Plataforma Multi-lenguaje en la Nube</b></li><li>– <b>Ventajas:</b> Permite programar Python sin instalación local. Soporte para despliegue de aplicaciones.</li><li>– <b>Recomendado:</b> Experimentación, proyectos rápidos.</li></ul>  |
| Ventajas Generales                | Accesibilidad (cualquier navegador), colaboración fácil, sin necesidad de configuración local.   |
| Desventajas                       | Dependencia de conexión a internet. Menos control sobre el entorno y recursos limitados en planes gratuitos.   |

# Resumen y Conclusión (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   |

# Resumen y Conclusión (Recomendaciones)

| Resumen y Conclusión (Recomendaciones) |  |
|--|--|
| Profesionales                          | <ul style="list-style-type: none"><li>- <b>PyCharm (Professional):</b> Desarrollo <i>full-stack</i> y proyectos complejos.</li><li>- <b>VS Code:</b> Versatilidad, multi-lenguaje, flujos personalizados.</li><li>- <b>Jupyter/Spyder:</b> Imprescindibles para Ciencia de Datos y Análisis.</li></ul> |
| No Profesionales                       | <ul style="list-style-type: none"><li>– <b>Thonny/IDLE:</b> Ideal para principiantes absolutos.</li><li>– <b>VS Code / Plataformas Online (Colab):</b> Buenas opciones para estudiantes que avanzan y experimentan.</li></ul>  |
| <b>Tendencias 2025</b>                 | Accesibilidad (cualquier navegador), colaboración fácil, sin necesidad de configuración local.   |
| <b>Desventajas</b>                     | Auge de la integración de <b>IA</b> (Copilot) y plataformas <b>en la nube</b> (Colab, Replit), haciendo el desarrollo más accesible y colaborativo.  |