

## **EVIDENCIA 2**

### **Entornos de trabajo para Python 3**

Nombre de la Unidad de Aprendizaje: Introducción a la programación

Nombre del proyecto: Actividad 2

Programa educativo: Licenciado en Tecnologías de información

Semestre: 1 Grupo: 11

Nombre del maestro: **ALVARO FRANCISCO SALAZAR GONZALEZ**

Nombre del estudiante:

CORTEZ CASTRO ELIUD **1904452**

BAUTISTA CRUZ MELANY GERALDIN **1956151**

TREVIÑO ZAMARRIPA ROBLE ALEJANDRO **1964751**

REYNA URESTI ISAAC **1923148**

# INSTRUCCIONES

Elaborar, en equipo, un análisis comparativo respecto a las siguientes herramientas con las que se puede codificar código para Python 3 (CADA MIEMBRO DEL EQUIPO DEBERÁ SUBIR SU PROPIA COPIA DE LA ACTIVIDAD A LA PLATAFORMA EN LA FECHA INDICADA EN FORMATO PDF) y deberá incluir debidamente identificado el siguiente contenido:

- 1) Entornos/Herramientas (agregar dos adicionales que se identifiquen mediante una búsqueda en Internet).
  - a. PyCharm
  - b. VsCode
  - c. Thonny
  - d. Sublime 3
  - e. Jupyter
- 2) Sitios oficiales y de descarga para cada uno de ellos
- 3) Propietarios de cada uno de los entornos presentados.
- 4) Breve historia de cada entorno presentado.
- 5) Cuadro comparativo con las principales características de cada uno de los entornos estudiados
- 6) Pros y contras de cada uno de acuerdo con comentarios de la comunidad de desarrollo 7) Opinión preliminar de la mejor opción para codificar en un curso básico de Python (Argumente)
- 8) Opinión preliminar de la mejor opción para codificar en Python de manera profesional (Argumente)

Recursos: <https://morioh.com/p/ec413b3341f5>

# ACTIVIDAD

## Sitios oficiales y de descarga para cada uno de ellos:

- PyCharm: <https://www.jetbrains.com/es-es/pycharm/>
- VsCode: <https://code.visualstudio.com/>
- Sublime Text: <https://www.sublimetext.com/>
- Thonny : <https://thonny.org>
- Jupyter : <https://jupyter.org/>
- Atom: <https://atom.io>
- Spyder Python: <https://pypi.org/project/spyder/#downloads>

## Propietarios de cada uno de los entornos presentados.

- PyCharm: JetBrains
- VsCode: Microsoft
- Sublime Text: Jon Skinner
- Thonny : Aivar Annamaa y contribuyentes
- Jupyter : [Fernando Pérez](#), [Brian Granger](#)
- Atom: GitHub
- Spyder Python :. Pierre Raybaut

## Breve historia de cada entorno presentado.

### PyCharm:

La versión beta se lanzó en julio de 2010 y la 1.0 llegaría 3 meses después. La versión 2.0 se publicó el 13 de diciembre de 2011, la versión 3.0 el 24 de septiembre de 2013 y la versión 4.0 el 19 de noviembre de 2014. PyCharm Community Edition, la versión de código abierto de PyCharm, estuvo disponible el 22 de octubre de 2013.

### VsCode:

Visual Studio Code fue anunciado el 29 de abril de 2015 por Microsoft en la conferencia Build de 2015. Una versión preliminar se lanzó poco después.

El 18 de noviembre de 2015, Visual Studio Code fue lanzado bajo la licencia MIT y su código fuente fue publicado en GitHub. También fue anunciada una nueva capacidad para agregar extensiones.<sup>1</sup>

El 14 de abril de 2016, Visual Studio Code graduó la etapa de vista previa pública y se lanzó a la web.

### **Sublime Text:**

se lanzó el 18 de enero del 2008 su última versión estable fue el 3.2.2 el 1 de octubre de 2019, Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo vi llamado Vintage mode. Se puede descargar y evaluar de forma gratuita. Sin embargo, no es software libre o de código abierto y se debe obtener una licencia para su uso continuado, aunque la versión de evaluación es plenamente funcional y no tiene fecha de caducidad. Está disponible para OS X, Windows y Linux. Sublime Text utiliza un paquete de herramientas de interfaz de usuario personalizado, optimizado para la velocidad y la belleza, aprovechando la funcionalidad nativa de cada plataforma. Cuenta con una potente API de complementos basada en Python. Junto con la API, cuenta con una consola integrada en Python para experimentar interactivamente en tiempo real. Una de sus últimas versiones en prueba es Build 3210

### **Thonny:**

Thonny es un entorno de desarrollo integrado (IDE). Desarrollado por la Universidad de Tartu en Estonia, este software ha sido creado para aprender y enseñar programación que puede hacer que la visualización de programas sea una parte natural del flujo de trabajo de los principiantes. Tiene una versión estable: 3.2.7 / 11 de marzo de 2020 se puede utilizar en Microsoft Windows, GNU/Linux, macOS Entre sus características destacadas se encuentran las diferentes formas de recorrer el código, la evaluación paso a paso de la expresión, la visualización intuitiva de la pila de llamadas y el modo para explicar los conceptos de referencias y montón. Apoya la investigación educativa al registrar las acciones del usuario para reproducir o analizar el proceso de programación. Es de uso gratuito y se puede ampliar.

### **Jupyter:**

Jupyter Notebook es uno de los editores de portátiles IPython más utilizados en la industria de la ciencia de datos. Aprovecha al máximo el hecho de que Python es un lenguaje interpretado, lo que significa que las líneas de código de Python pueden ejecutarse una línea a la vez y no es necesario compilar todo el conjunto como C / C ++. Esto hace que los portátiles IPython sean ideales para escribir y crear prototipos de modelos de aprendizaje automático. Dado que inicialmente se realiza una cantidad significativa de preprocesamiento, y después de eso, hay un proceso de ajuste repetido de hiperparámetros y creación de prototipos de

modelos, la capacidad de ejecutar una celda (un grupo de líneas) juntas a la vez les da a los científicos de datos la capacidad de sintonizar sus modelos fácilmente.

### **Atom:**

Un framework que permite crear aplicaciones de escritorio multiplataforma usando Chromium y Node.js.<sup>678</sup> Está escrito en CoffeeScript y Less. También puede ser utilizado como un entorno de desarrollo integrado (IDE). Atom liberó su beta en la versión 1.0, el 25 de junio de 2015.

Sus desarrolladores lo llaman un «editor de textos hackeable para el siglo XXI».

### **Spyder python:**

Originalmente creado y desarrollado por Pierre Raybaut en 2009, desde 2012 Spyder ha sido mantenido y mejorado continuamente por un equipo de desarrolladores científicos de Python y la comunidad.

## **Cuadro comparativo con las principales características de cada los entornos estudiados**

<i>Entornos</i>	<i>Características</i>
<b>PyCharm</b>	<ul style="list-style-type: none"><li>• Editor inteligente: Resaltado de sintaxis, sangría automática y formateo de código, finalización de código, formateador de código, resaltado de errores sobre la marcha, análisis del código a medida que se escribe y detector de código duplicado.</li><li>• Depurador gráfico en local.</li><li>• Refactorizaciones rápidas y seguras válidas para nombres de fichero, funciones, constantes, clases, propiedades, métodos, parámetros y variables locales y globales.</li><li>• Navegación inteligente por el código: Buscar en todas partes, encontrar usos de cualquier símbolo, ir a clase/archivo/símbolo, ir a declaración, marcadores personalizados</li><li>• Integración con sistemas de control de versiones: Git, Mercurial, CVS, Subversion, GitHub.</li></ul>

	<ul style="list-style-type: none"> <li>• Ejecución de un plan de pruebas.</li> <li>• Integración con documentación: reStructuredText y Google.</li> <li>• Integración con PyQt y PyGTK.</li> <li>• Gestión de paquetes</li> <li>• Integración nativa con Virtualenv/Buildout.</li> <li>• Consola de Python.</li> <li>• Soporte para XML, HTML, YAML, JSON, RelaxNG.</li> <li>• Terminal local.</li> </ul>
<b><i>VsCode</i></b>	<p>El código combina la interfaz de usuario optimizada de un editor moderno con asistencia y navegación de código enriquecido y una experiencia de depuración integrada, sin la necesidad de un IDE completo. Visual Studio Code, cuenta con herramientas de Debug hasta opciones para actualización en tiempo real de nuestro código en la vista del navegador y compilación en vivo de los lenguajes que lo requieran (por ejemplo en el caso de SASS a CSS). Además de las extensiones, tendremos la posibilidad de optar por otros themes o bien configurarlo a nuestro gusto. Para modificar el esquema de colores y los iconos</p>
<b><i>Sublime Text</i></b>	<ul style="list-style-type: none"> <li>• Simplicidad: Posé Una interfaz muy sencilla, pero no por eso carece de funciones.</li> <li>• Sublime Text tiene una potente API de Python que permite que los complementos aumenten la funcionalidad incorporada</li> <li>• Compatibilidad: Sublime Text 3 es flexible y soporta varios lenguajes de programación</li> <li>• Mini mapa: consiste en una previsualización de la estructura del código, es muy útil para desplazarse por el archivo cuando se conoce bien la estructura de este.</li> </ul>

- Multi Selección: Hace una selección múltiple de un término por diferentes partes del archivo.
- Syntax Highlight configurable: El remarcado de sintaxis es completamente configurable a través de archivos de configuración del usuario.
- Búsqueda Dinámica: Se puede hacer búsqueda de expresiones regulares o por archivos, proyectos, directorios, una conjunción de ellos o todo a la vez.
- Auto completado y marcado de llaves: Se puede ir a la llave que cierra o abre un bloque de una forma sencilla.
- Soporte de Snippets y Plugins: Los snippets son similares a las macros o los bundles además de la existencia de multitud de plugin.
- Configuración total de Keybindings: Todas las teclas pueden ser sobrescritas a nuestro gusto.
- Acceso rápido a línea o archivo: Se puede abrir un archivo utilizando el conjunto de teclas CMP en Mac OS X o Ctrl+P en Windows y Linux y escribiendo el nombre del mismo o navegando por una lista. También se puede ir a una línea utilizando los dos puntos ":" y el número de línea.
- Paleta de Comandos: Un intérprete de Python diseñado solo para el programa con el cual se puede realizar infinidad de tareas.
- Coloreado y envoltura de sintaxis: Si se escribe en un lenguaje de programación o marcado, resalta las

	<p>expresiones propias de la sintaxis de ese lenguaje para facilitar su lectura.</p> <ul style="list-style-type: none"> <li>• Pestañas: Se pueden abrir varios documentos y organizarlos en pestañas.</li> </ul>
<p><b>Thonny</b></p>	<ul style="list-style-type: none"> <li>• Fácil de empezar. Thonny viene con Python 3.7 integrado, por lo que solo se necesita un instalador simple y está listo para aprender a programar. (También puede usar una instalación de Python separada, si es necesario). La interfaz de usuario inicial está despojada de todas las características que pueden distraer a los principiantes.</li> <li>• Variables sin complicaciones. Una vez que haya terminado con hello-worlds, seleccione Ver → Variables y vea cómo sus programas y comandos de shell afectan las variables de Python.</li> <li>• Depurador simple. Simplemente presione Ctrl + F5 en lugar de F5 y podrá ejecutar sus programas paso a paso, sin necesidad de puntos de interrupción. Presione F6 para un gran paso y F7 para un pequeño paso. Los pasos siguen la estructura del programa, no solo las líneas de código.</li> <li>• Paso a través de la evaluación de expresiones. Si usa pequeños pasos, puede incluso ver cómo Python evalúa sus expresiones. Puede pensar en esta caja celeste como una hoja de papel donde Python reemplaza las subexpresiones con sus valores, pieza por pieza.</li> <li>• Representación fiel de llamadas a funciones. Entrar en una llamada de función abre una nueva ventana con una tabla de variables locales y un puntero de código separados. Una buena comprensión de cómo funcionan las llamadas a funciones es especialmente importante para comprender la recursividad.</li> </ul>



	<ul style="list-style-type: none"> <li>• Destaca los errores de sintaxis. Las comillas sin cerrar y los paréntesis son los errores de sintaxis más comunes para principiantes. El editor de Thonny hace que sean fáciles de detectar</li> <li>• Explica los alcances. Resaltar ocurrencias de variables le recuerda que el mismo nombre no siempre significa la misma variable y ayuda a detectar errores tipográficos. Las variables locales se distinguen visualmente de las globales.</li> <li>• Modo para explicar referencias. Las variables se presentan inicialmente de acuerdo con el modelo simplificado (nombre → valor) pero puede cambiar a un modelo más realista (nombre → dirección / id → valor).</li> <li>• Finalización de código. Los estudiantes pueden explorar las API con la ayuda de la finalización de código.</li> <li>• Carcasa del sistema amigable para principiantes. Seleccione Herramientas → Abrir shell del sistema para instalar paquetes adicionales o aprender a manejar Python en la línea de comandos. Thonny se ocupa de PATH y de los conflictos con otros intérpretes de Python.</li> <li>• GUI pip simple y limpio. Seleccione Herramientas → Administrar paquetes para una instalación aún más fácil de paquetes de terceros.</li> </ul>
<i>Jupyter</i>	<ul style="list-style-type: none"> <li>• Compatibilidad con Markdown y LaTeX: Jupyter Notebooks, además de poder escribir código Python, admite documentación y comentarios con formato de texto a través de Markdown Editor. Cada celda se puede convertir para usar Markdown o Code. Además, Jupyter Notebook, al ser una herramienta científica, también admite comandos LaTeX para escribir ecuaciones en cualquier celda del cuaderno.</li> </ul>

	<ul style="list-style-type: none"> <li>• Pantalla dedicada para DataFrame y Plots: dado que los datos son el componente central de Data Science y Machine Learning, Jupyter Notebook tiene una pantalla incorporada para tablas de datos o pandas DataFrames. Además, la visualización de datos es un proceso importante del análisis exploratorio de datos del canal de ciencia de datos. Por lo tanto, Jupyter Notebooks tiene una pantalla integrada para gráficos y diagramas para que el desarrollador no tenga que lidiar con gráficos emergentes.</li> <li>• Soporte de desarrollo remoto: Jupyter Notebook es una aplicación basada en servidor que, cuando se ejecuta localmente, crea un servidor localhost antes de abrirse a través de un navegador web. Pero lo mismo se puede utilizar para el desarrollo remoto, ya que el portátil se puede ejecutar en un servidor remoto al que luego se puede conectar para ejecutar el portátil localmente en el navegador web, mientras que el procesamiento se realiza en el lado del servidor.</li> <li>• Acceso directo al símbolo del sistema o al shell de Linux desde el interior del portátil: dado que el portátil se puede utilizar como una herramienta de desarrollo remoto, el portátil permite al desarrollador acceder directamente al shell de Linux o al símbolo del sistema de Windows directamente desde el propio portátil sin tener que abrir el shell o símbolo del sistema. Esto se logra agregando un signo de exclamación ("!") Antes de escribir el comando de shell.</li> <li>• Compatibilidad con varios idiomas: Jupyter Notebook es compatible con Python y R. R es también un lenguaje de programación utilizado popularmente por científicos de datos y estadísticos.</li> </ul>
<i>Atom</i>	<ul style="list-style-type: none"> <li>• Sirve para trabajar en cualquier sistema operativo (Windows, OS X o Linux)</li> <li>• Puedes instalarle paquetes o crear los tuyos propios</li> <li>• Autocompletado inteligente que permite escribir código más rápido</li> <li>• Buscar y reemplazar texto de una forma sencilla</li> </ul>
<i>Spyder python</i>	<ul style="list-style-type: none"> <li>• El editor que integra este IDE es multilenguaje. Posé un navegador de función/clase, funciones de análisis</li> </ul>

de código (pyflakes y pylint son actualmente compatibles), la opción de finalización de código, división horizontal y vertical y definición goto.

- Consola interactiva. Las consolas Python o IPython son un espacio de trabajo y soporte de depuración para evaluar al instante el código escrito en el Editor. También viene con la integración de figuras Matplotlib.
- Posé un visor de documentación. El programa nos va a poder mostrar documentación para cualquier llamada de clase o función realizada en el Editor o en una consola.
- Vamos a poder explorar las variables creadas durante la ejecución de un archivo. Va a ser posible editarlas con varios editores basados en GUI, como un diccionario y los de matriz Numpy.

## **Pros y contras de cada uno de acuerdo con comentarios de la comunidad de desarrollo**

### **PyCharm:**

- Pros: En base a los comentarios encontrados en la comunidad de Capterra, nos hemos percatado que a los usuarios les gusta utilizar el editor de código PyCharm por diversas razones como pueden ser el autocompletado del código, el depurador, también les gusta que existen extensiones y complementos los cuales se van actualizando constantemente, otra cualidad de PyCharm es la facilidad para la búsqueda y la instalación de paquetes y que les permite realizar el seguimiento de un proyecto muy grande. Hemos notado que en general les gusta utilizar PyCharm gracias a que se adapta muy bien a la hora de programar en Python y que les ayuda a mantener un código legible y ordenado.
- Contrás: La comunidad piensa que PyCharm consume demasiados recursos y que para poder utilizarlo se necesitara de una computadora lo suficientemente potente como para poder brindarles la experiencia adecuada, de lo contrario se vuelve molesto y lento el utilizarlo. Otro punto negativo que hemos detectado es que la versión de paga es algo costosa y

en comparación hay opciones gratuitas que funcionan muy bien. También piensan que la depuración es un proceso muy lento el cual podría mejorarse. Otro comentario con el cual nos hemos encontrado bastante seguido es que al utilizar PyCharm durante mucho tiempo se terminan acostumbrando mucho a el y a las facilidades que les brinda y al momento de utilizar otro editor se vuelve complicado acostumbrarse.

### **VsCode:**

- Pros: A la gente les suele gustar mucho Visual Studio Code y al ser respaldado por una empresa tan grande como lo es Microsoft se convierte en una de sus mayores cualidades ya que constantemente se esta actualizando. Una de las características que mas les gusta a los usuarios es la herramienta IntelliSense ya que les permite autocompletar código de manera inteligente. Otra característica que les gusta mucho a los usuarios es que viene vinculada con Git desde un principio y este le facilita mucho a la hora de trabajar en conjunto con otros desarrolladores. Un punto clave son las extensiones ya que tiene una gran variedad y es muy sencillo el poder instalarlas.
- Contras: En general no encontramos comentarios negativos al respecto de Visual Studio Code, lo mas recalable es que mencionan los usuarios que los atajos del teclado cambian dependiendo de en que plataforma te encuentres, ya sea Windows, Mac o cualquier distribución de Linux. Otro punto que han mencionado los usuarios es que no les gusta que trabaje sobre Electron ya que consideran que debido a esto no da un rendimiento adecuado en algunas ocasiones. En general fue difícil encontrar comentarios negativos sobre este editor de código.

### **Sublime text:**

Pros:

- Muy liviano, fácil de instalar y tiene una versión portable, resalta todo tipo de lenguaje con colores para visualmente detectar fallos a simple vista.
- Para aprender es una buena opción, porque te ayuda, pero no te lo da todo hecho.
- Funciona tanto en Windows como en Mac y Linux.
- Sublime Text es el editor de texto más rápido y liviano de la competencia y, sin embargo, es lo suficientemente funcional como para usarse como un IDE. Proporciona una combinación única de versatilidad y funcionalidad, que es verdaderamente única.
- Su rapidez en la ejecución

- Es muy ligero, funcionalidades comparables a los entornos de desarrollo integrados
- Multilenguaje y multiplataforma
- Cursor múltiple
- Gestión de proyectos completos
- Posibilidad de ejecutar y depurar código sin salir del editor
- Comunidad de usuarios cada vez mayor

Contras:

- Es totalmente configurable, pero si queremos una configuración avanzada es un poco difícil de hacer para usuarios novatos.
- No es gratuito, tiene una versión registrada, aunque vale la pena. Si no lo tenemos registrado, no pierde ninguna funcionalidad, aunque aparece un mensaje de compra cada cierto tiempo al grabar un archivo.
- ser un editor de texto, a pesar de que compensa su falta de funcionalidad incorporada a través de complementos y complementos, al final del día, sigue siendo un editor de texto y carece de algunas características clave que poseen los IDE dedicados.
- Difícil de aprender y configurar

Thonny

Pros:

- ❖ amigable con el novato, y como ya se ha comentado que este lenguaje es ideal para iniciarse en el sector, se lo podría considerar como una herramienta ideal para aquellos que quieran empezar a aprender a programar, ofreciendo además versiones para GNU/Linux, Windows y Mac.
- ❖ El programa funciona en Windows, macOS y Linux. Está disponible como paquete binario que incluye un intérprete de Python reciente o un paquete instalable en pip. Se puede instalar a través del administrador de paquetes del sistema operativo en Debian, Raspberry Pi, Ubuntu y Fedora.
- ❖ diseñado principalmente para facilitar la vida de los principiantes en Python proporcionándoles un IDE simple y ligera, pero con excelentes características, es un poco como el kit de principiantes. Por lo tanto, este programa informático es particularmente adecuado para los principiantes que desean comenzar a programar y desarrollar en Python y, por lo tanto, no es en absoluto adecuado para los expertos en desarrollo.

Contras:

- Si usted es un desarrollador experimentado, este software no es para usted.
- Sólo las funcionalidades básicas.

- Su principal inconveniente, común a otros IDEs en mayor o menor medida, en el consumo de recursos del sistema.
- También que carece de mucho soporte para webapps (.war, jsp y servlets) tal y como lo hace netbeans. (los plugins por lo general no son ni tan potentes ni tan sencillos como el módulo que en Netbeans viene preinstalado).

### **Jupyter:**

Pros: la principal ventaja es la conveniencia de usarlo en I + D y en la creación de prototipos para el aprendizaje automático y problemas científicos. Reduce significativamente el tiempo necesario para la creación de prototipos y el ajuste de modelos de aprendizaje automático en comparación con otros IDE.

Contras: La principal desventaja es que este IDE no es compatible con todo el proceso de desarrollo y es ideal solo para la creación de prototipos. No tiene las herramientas o funcionalidades adicionales que hacen que otros IDE sean ideales para la implementación de programas y scripts.

### **Atom:**

Pros:

- 1.- Lo mejor de otros mundos en uno solo: Atom, así como otros editores de texto, permite integrar características que son útiles en otros programas a su propia configuración. Es decir que si lo deseas puedes integrar un paquete que mejore el cierre automático de las etiquetas o un paquete que te ayude a integrar Bootstrap de forma sencilla y con unas cuantas teclas o tal vez un paquete que te resalte los errores en el código, por mencionar algunos.
- 2.- Paquetes y más paquetes: Como lo mencionamos en el punto anterior puedes 'armar' Atom a tu gusto integrando algún paquete (package) desarrollado por alguien de la comunidad de GitHub, lo cual garantiza que al menos el 99.9% de ellos son gratuitos. Por aquí puedes conocer más al respecto.
- 3.- Interfaz amigable a la vista: Atom integra diferentes temas que puedes alternar conforme la hora del día, es decir que de día puedes optar por un tema luminoso y de noche por alguno más oscuro. No más ojos irritados por pasar días horas escribiendo código.

- 4.- Ligereza que tu RAM apreciará: Atom se actualiza constantemente corrigiendo diferentes bugs y mejorando sus funciones, esto no significa que vaya a pesar más y más cada vez. Podemos garantizar que es de lo más ligero que existe en cuanto a editores.
- 5.- Te hará aprender más: Cuando nos acostumbramos a un programa damos algunas cosas por hecho, tal es el caso de Dreamweaver que tiene montones de ventanas las cuales resultan muy cómodas, pero en definitiva no te obligan a aprender el código que generas con ellas.

Contras:

- A veces débil en el extremo rendimiento
- La documentación es irregular en lugares

### **Spyder Phyton:**

Pros:

- Gratis
- Depuración línea por línea
- Similar a Rstudio

Contras:

- Interfaz antigua
- No es fácil cambiar ENV

### **Opinión preliminar de la mejor opción para codificar en un curso básico de Python (Argumente):**

Observando y analizando la gran variedad de editores de texto que existen llegamos a la conclusión de que la mejor alternativa para empezar a programar en el lenguaje Python es sin duda Thonny, lo que nos llevo a tomar esta decisión es que este editor de texto está pensado específicamente para desarrollar con Python y la ventaja que representa es viene integrado con la versión 3.7 de Python y solo se necesita instalar el editor, esto beneficia al usuario porque le permite solo enfocarse en aprender la lógica y la sintaxis de Python. Otra razón por la cual creemos que Thonny es la mejor opción es que tiene la opción de correr el programa paso a paso

y esto permite ver el flujo del algoritmo y observar como se van evaluando cada una de las expresiones y esto le ayuda al estudiante a comprender mejor. También tiene la opción de mostrar una tabla con las variables del algoritmo y ver como estas van cambiando a lo largo del programa.

### **Opinión preliminar de la mejor opción para codificar en Python de manera profesional (Argumente):**

Para alguien mas experimentado en el mundo de la programación existen muchas otras alternativas y realmente todas son muy buenas y creemos que depende mas de los gustos personales de cada usuario, a fin de cuentas son herramientas que nos sirven para realizar un objetivo, pero creemos que uno de los mejores editores de texto es sin pensarlo Visual Studio Code y esto mas que nada porque tiene el respaldo de una empresa tan grande como lo es Microsoft, además de que viene integrado con Git lo que hace que sea mucho mas ameno realizar trabajo en conjunto con diversas personas.