



# Tutorial de Eclipse y PyDev

---

En este tutorial usted se describe paso a paso y en detalle el proceso de instalación de las herramientas de desarrollo a utilizar en este curso: el intérprete de Python, Eclipse y PyDev. Además, el tutorial también muestra los principales procedimientos necesarios para crear y ejecutar programas.

## 1. Instalación de Python en Linux.

En Linux se tiene instalado Python por defecto, la versión de éste dependerá de la distribución de Linux con la que se está trabajando. Por ejemplo, para las distribuciones Ubuntu 12.x se tiene por defecto Python 2.7 y para Ubuntu 14.x se tienen instaladas por defecto Python 2.7 y Python 3.4. Para comprobar esto, basta abrir una terminal, escribir Python y presionar la tecla Tab para observar las versiones que se tienen.

### 1.1. Instalación de Python 3.4 (solo para Ubuntu 12.x).

1. Abrir una terminal con la combinación de teclas **Alt + Ctrl + T**.
2. Agregamos el repositorio desde el cual bajaremos nuestro software: **sudo add-apt-repository ppa:frull/deadsnakes**. (Puede copiar y pegar en el terminal, para pegar se hace mediante la combinación **Ctrl + shift + v**).
3. Debemos actualizar los repositorios: **sudo apt-get update**
4. Instalar Python: **sudo apt-get install python3.4**

## 2. Instalación de Eclipse en Linux

Siga los pasos mencionados a continuación para instalar Eclipse en una distribución de Linux derivada de Debian. (Ubuntu, Mint, Kubuntu, etc); si tiene una versión no derivada de Debian, probablemente este tutorial le sea de ayuda.

1. Abrir una terminal: Dependiendo de la distribución de Linux que tenga instalada, esta aplicación se puede encontrar en diferentes partes del menú. Si no la encuentra, puede intentar presionar **Alt + Ctrl + T**.
2. Instalar Java: Como Eclipse es una aplicación desarrollada en Java, requiere que una máquina virtual de Java esté instalada en el sistema. Ingrese en una terminal de Linux el siguiente comando: **sudo apt-get install openjdk-7-jre-headless**. Antes de iniciar la instalación, la terminal le pedirá que lo confirme. Ingrese Yes o Si (dependiendo si está en inglés o en español) para proceder con la instalación.
3. Descargar Eclipse: Una vez termine la instalación de Java, abra un navegador y copie uno de los siguientes enlaces para descargar Eclipse IDE for Java Developers de acuerdo a si su sistema es de 32 o de 64 bits:
  - 32 bits:  
[https://eclipse.org/downloads/download.php?file=/technology/epp/downloads/releases/mars/R/eclipse-java-mars-R-linux-gtk.tar.gz&mirror\\_id=576](https://eclipse.org/downloads/download.php?file=/technology/epp/downloads/releases/mars/R/eclipse-java-mars-R-linux-gtk.tar.gz&mirror_id=576)
  - 64 bits:  
[https://eclipse.org/downloads/download.php?file=/technology/epp/downloads/releases/mars/R/eclipse-java-mars-R-linux-gtk-x86\\_64.tar.gz&mirror\\_id=576](https://eclipse.org/downloads/download.php?file=/technology/epp/downloads/releases/mars/R/eclipse-java-mars-R-linux-gtk-x86_64.tar.gz&mirror_id=576)

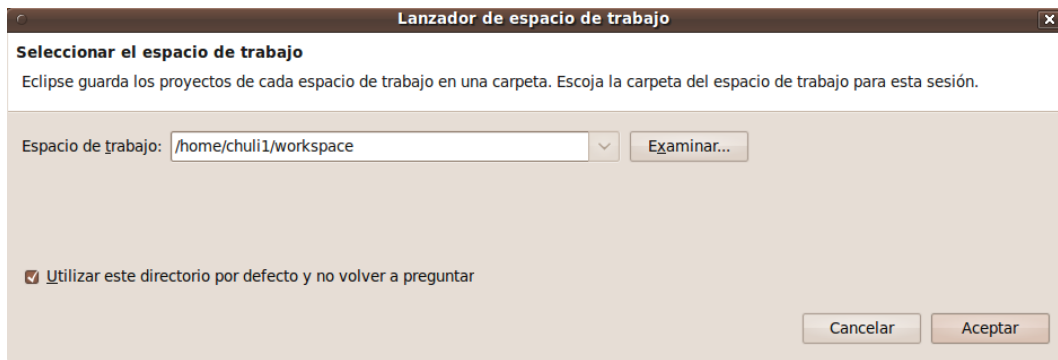
Para determinar si su versión de Ubuntu es de 32 o 64 bits, ejecute el comando **uname -m** en una terminal de Linux. Si el resultado es i386, quiere decir que su sistema operativo es de 32 bits. Si el resultado es x86\_64, quiere decir que su sistema es de 64 bits.

4. Instalar Eclipse: Luego que Eclipse haya terminado de descargarse como se indicó en el paso anterior, abra una terminal y ejecute los siguientes comandos, en su orden:

- Descomprimir:
  - o Para 64 bits
    - `tar xzvf Descargas/eclipse-java-mars-R-linux-gtk-x86_64.tar.gz`
  - o Para 32 bits
    - `tar xzvf Descargas/eclipse-java-mars-R-linux-gtk.tar.gz`
- Mover a carpeta apropiada: `sudo mv eclipse/ /opt`
- Crear acceso directo: `sudo ln -s /opt/eclipse/eclipse /usr/local/bin`

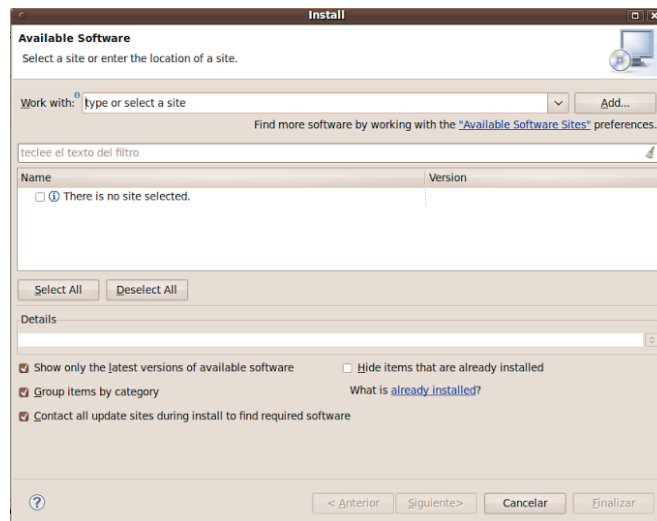
## 2.1 Configuración de PyDev en Eclipse.

Después de tener Eclipse correctamente instalado en nuestro equipo, debemos configurar el plugin PyDev para trabajar con Python. Lo primero que hacemos es ejecutar Eclipse. Para esto, abra una terminal, escriba `eclipse` y presione enter. Al abrirse el programa se nos pedirá un directorio que será utilizado como espacio de trabajo. Podemos dejar el que viene por defecto marcando la casilla de como se muestra en la Figura 1.



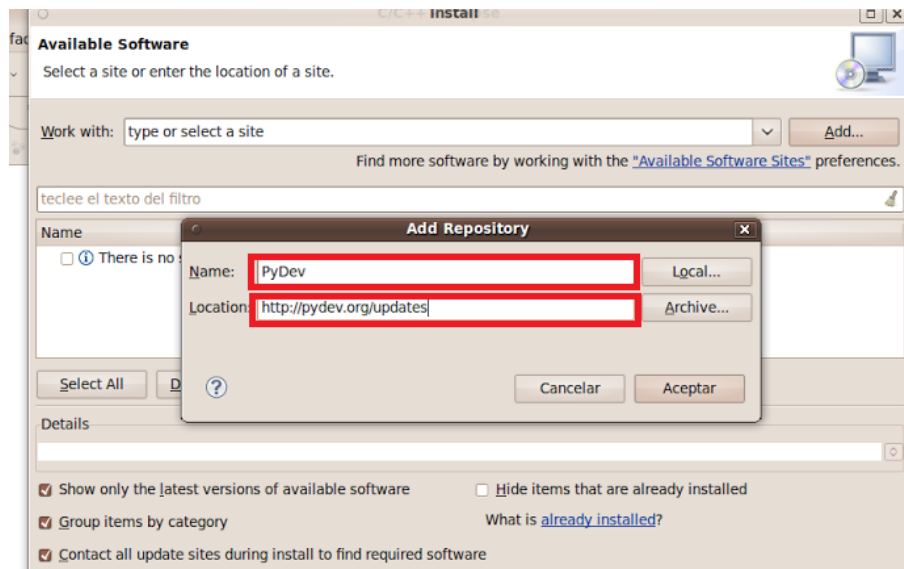
**Figura 1. Configuración de PyDev en Eclipse.**

Estando en la ventana principal de Eclipse vamos a help/ayuda y escogemos la opción de Instalar nuevo software.



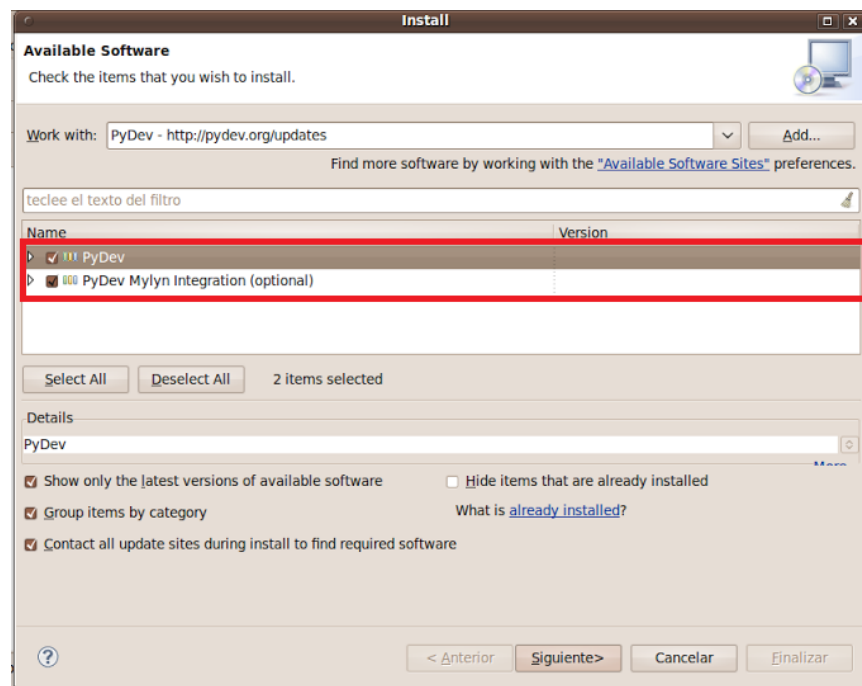
**Figura 2. Configuración de PyDev en Eclipse**

En la Figura 2. Hacemos clic en “Add...” para añadir el repositorio de PyDev, escribimos tal cual el sitio del cual obtendremos el repositorio, este es: **<http://pydev.org/updates>**



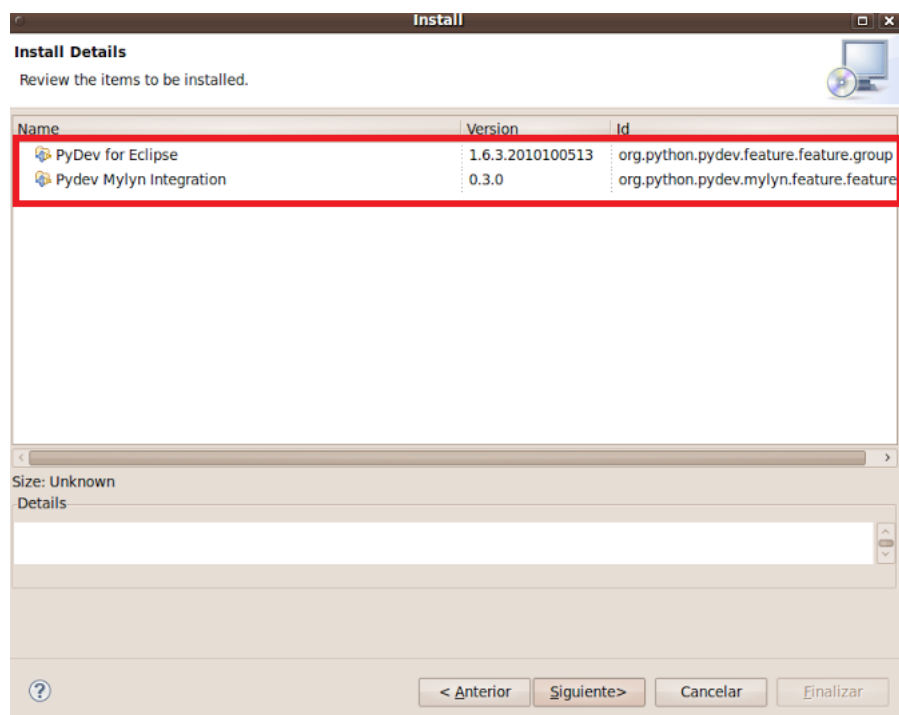
**Figura 3. Configuración de PyDev en Eclipse.**

Ahora marcamos las dos casillas que aparecen (PyDev / PyDev Mylin Integration) y pulsamos siguiente:



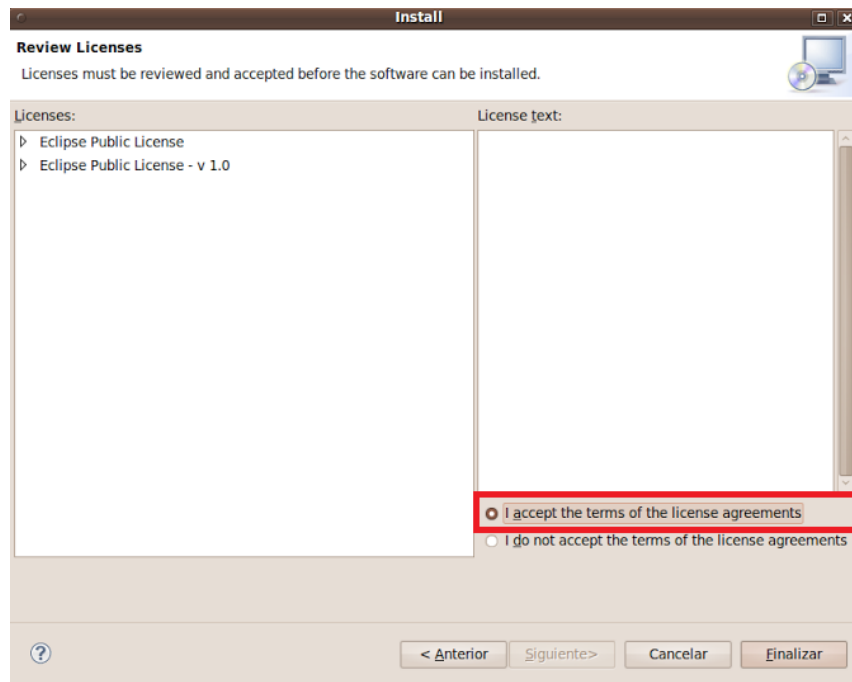
**Figura 4. Configuración de PyDev en Eclipse.**

Nos saldrá una lista detallada de las herramientas que se van a instalar como se muestra en la Figura 5, presionamos siguiente.



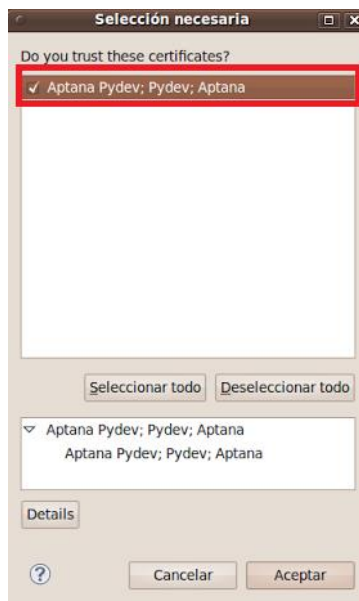
**Figura 5. Configuración de PyDev en Eclipse.**

Aceptamos la licencia y clic en finalizar.



**Figura 6. Configuración de PyDev en Eclipse.**

Comenzará el proceso de instalación, que puede tardar algunos minutos, además saldrá un mensaje de advertencia que debemos aceptar, luego de finalizada la instalación debemos marcar los certificados de PyDev.



**Figura 7. Configuración de PyDev en Eclipse.**

Aceptamos y por último se nos solicitara reiniciar Eclipse, ya estamos listos para crear nuestro primer proyecto en lenguaje Python.

### 3. Crear un proyecto nuevo

Cuando se desarrolla software con un IDE, cada programa que se hace debe pertenecer a un proyecto. El término proyecto se refiere a un conjunto de archivos de configuración que están ocultos al usuario y que se utilizan para la correcta ejecución del programa que está implementado en uno o varios archivos fuente. Para abrir la aplicación puede buscar el icono correspondiente en el menú, o en una terminal ingresar la palabra Eclipse y dar Enter, cuando la aplicación inicie, dé clic en aceptar al primer cuadro de diálogo que aparece, luego vaya al menú **Archivo** → **Nuevo** → **Project** → **PyDev Project**. Darle clic en next, como se ve en la Figura 8.

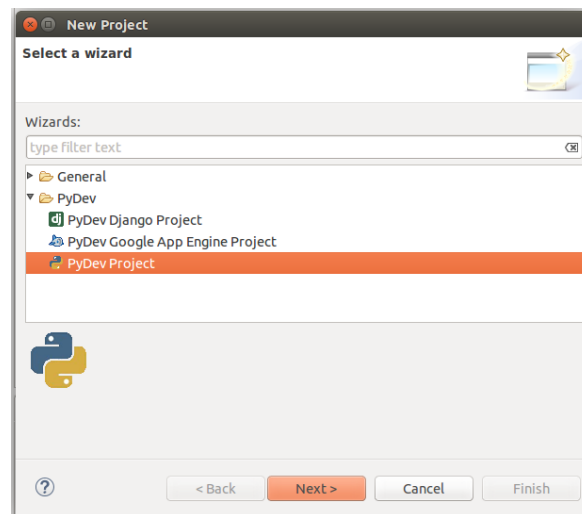


Figura 8. Creación de un nuevo proyecto

Si esta es la primera vez que se utilizara el software, debemos configurar el intérprete necesario para trabajar correctamente con Python (esto solo se debe hacer la primera vez, en caso de no tener el intérprete configurado), por lo que daremos clic en la sección de color rojo como se observa en la Figura 9.

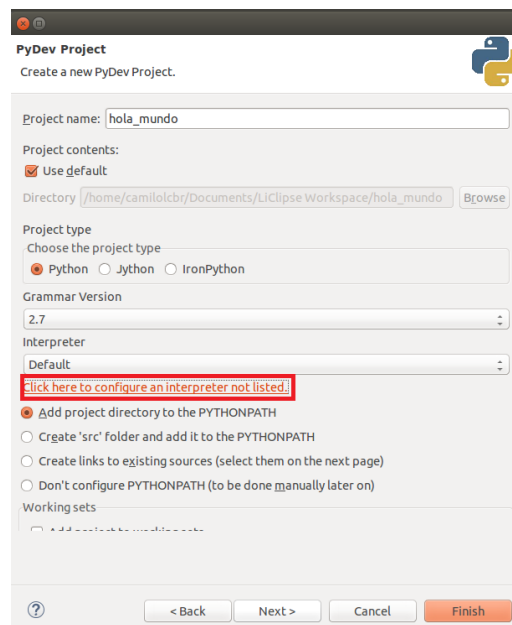


Figura 9. Configuración del intérprete de Python.

Debemos configurar un nuevo intérprete, dado que no tenemos ninguno hasta el momento, luego se podrán configurar los intérpretes que queramos dependiendo de la versión de Python que utilizamos, en nuestro caso trabajaremos con Python 3.4. Hacemos clic en **Advanced Auto-config** como se muestra en la Figura 10.

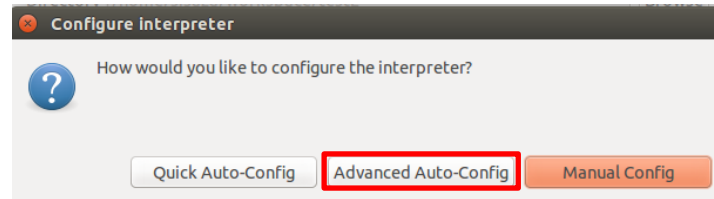


Figura 10. Configuración del intérprete de Python

Nos aparecerá una ventana con todos los intérpretes que tenemos instalados en nuestro sistema, seleccionamos Python 3.4.

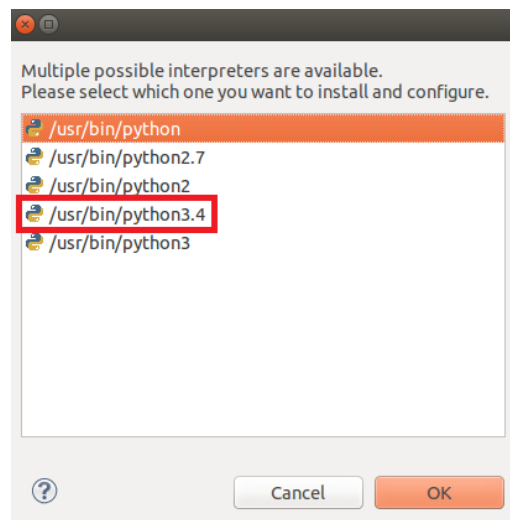


Figura 11. Configuración del intérprete de Python

Damos en aceptar, verificamos que todas las casillas estén seleccionadas, esto para hacer uso de todas las librerías disponibles, como se muestra en la Figura 12 y nuevamente presionamos en aceptar.

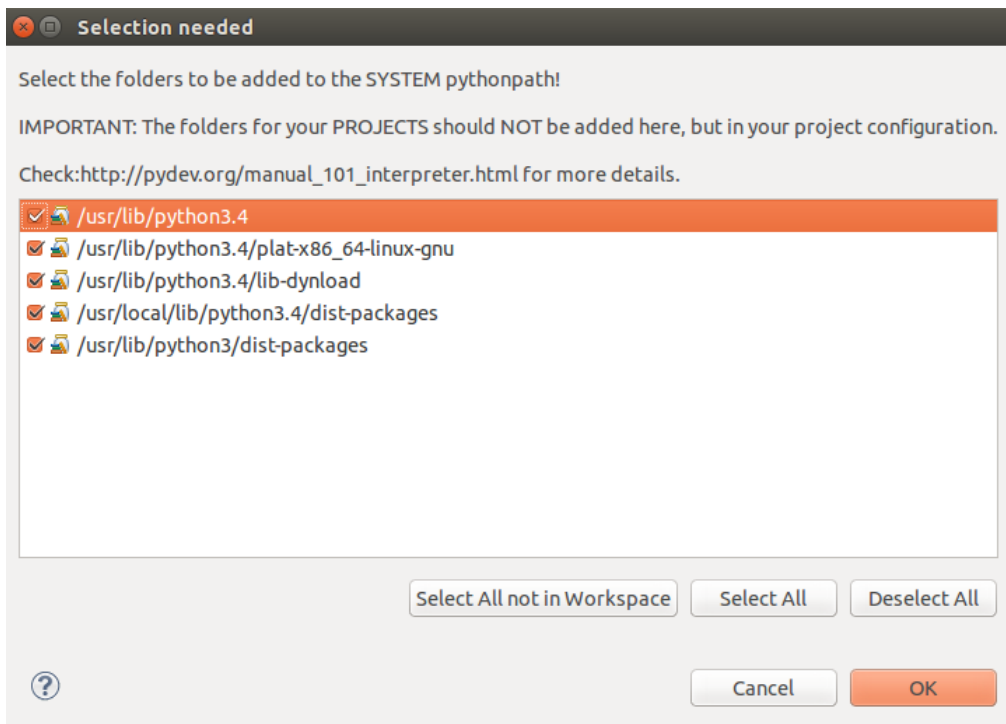


Figura 12. Configuración del intérprete de Python.

En este punto ya podemos asignar el nombre a nuestro proyecto, seleccionamos **Grammar Version 3** y presionamos en finalizar.

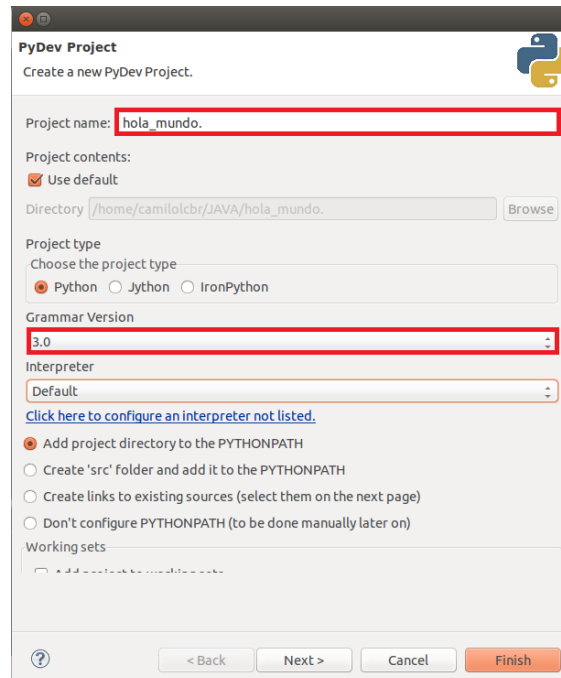


Figura 13. Asignación del nombre al proyecto.

Ahora debemos adicionar nuestro primer archivo fuente al proyecto, este es el archivo donde escribiremos nuestro código, para esto nos paramos justo encima del nombre de nuestro proyecto y damos clic derecho, **New** → **PyDev module**, indicamos el nombre del archivo a crear y presionamos finalizar, aparecerá un cuadro de dialogo que se debe aceptar.



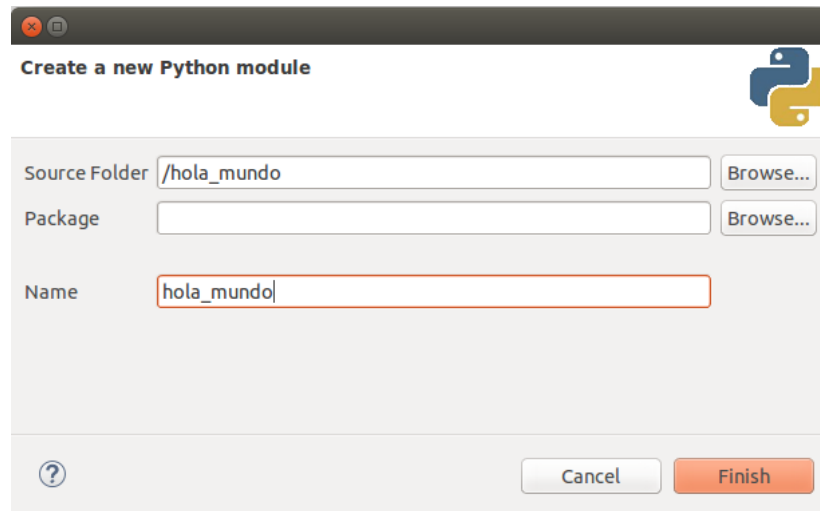


Figura 14. Adicionar archivo fuente.

En este punto debe aparecer nuestro archivo fuente, justo debajo del nombre de nuestro proyecto, y al lado derecho se tiene nuestro editor de texto donde escribiremos el código, como se observa en la Figura 15.

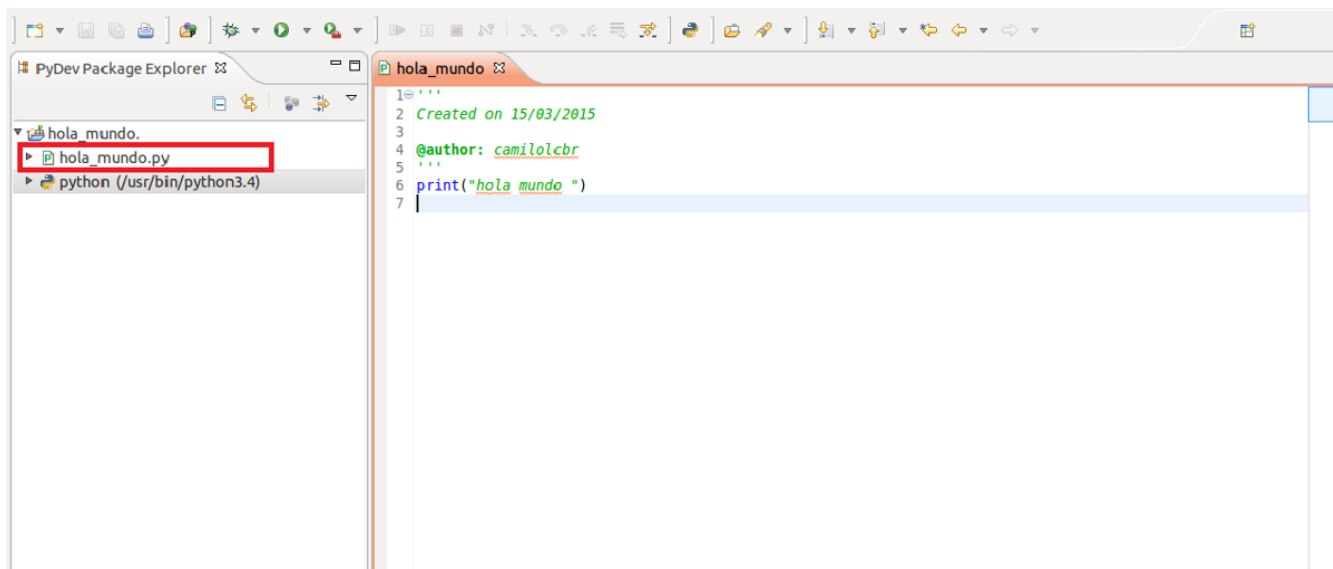


Figura 15. Adicionar archivo fuente.

#### 4. Ejecución de un programa en Eclipse

Luego de crear un proyecto y escribir las instrucciones del programa en un archivo fuente, el siguiente paso en el flujo de trabajo es la ejecución del programa. Para esto, presionamos el botón con el ícono de Python (como se muestra en la Figura 16), o simplemente, con el cursor activo en el código fuente, tecleamos la combinación Ctrl+Alt+Enter. Al hacer esto por primera vez en un proyecto, se desplegará una ventana como la mostrada en la Figura 17, donde debemos seleccionar la consola a utilizar: Python console.

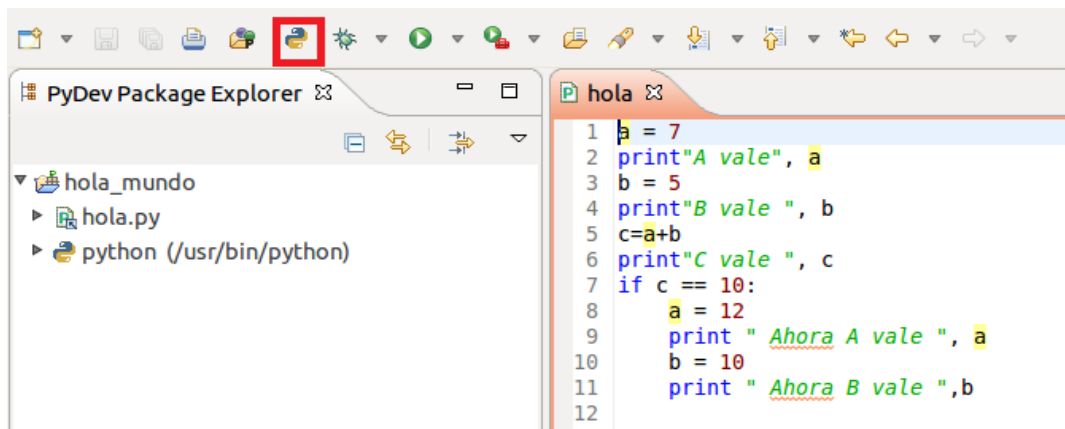


Figura 16. Botón para ejecución de un programa.

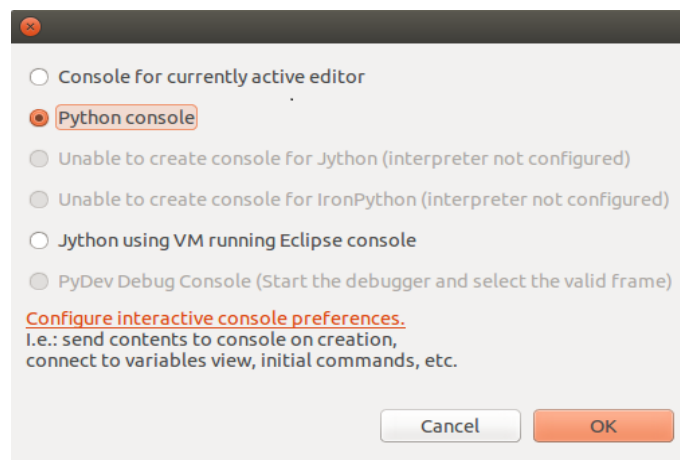
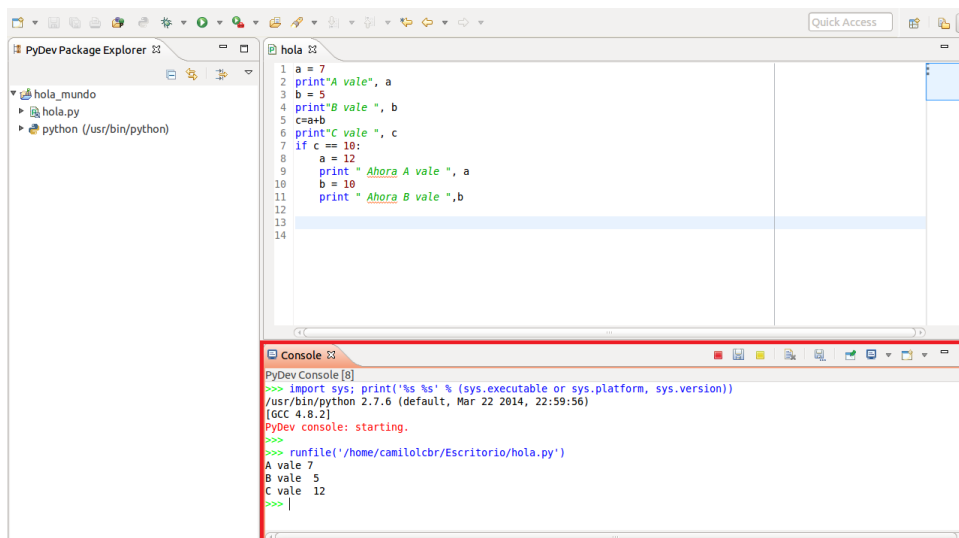


Figura 17. Seleccionar la consola interactiva.

La consola interactiva se desplegará en la parte inferior como se muestra en la Figura 18. Luego, ejecutaremos nuestro código, de nuevo con la combinación de teclas **Ctrl+Alt+Enter**.

La consola interactiva de Eclipse, permite tener un mayor control sobre nuestro programa, permitiendo hacer modificaciones y pruebas sencillas, sin necesidad de ir a modificar nuestro código cada vez (ver Sección 0).



**Figura 18. Ejecutar un programa en la consola interactiva.**

Cada que se hagan cambios en el código fuente, debemos guardar los cambios antes de ejecutar nuevamente el programa modificado.

#### *Ejecución de un programa Python en la terminal de Linux*

La ejecución de un programa en Eclipse es útil en el proceso de desarrollo de un programa. Sin embargo, cualquier programa Python puede ser ejecutado directamente en la terminal de Linux sin la intervención de Eclipse. Para ejecutar un programa Python en la terminal siga estos simples pasos:

- Abra una terminal de Linux
- Ubíquese en la carpeta donde se encuentra el archivo Python (.py)
- Ingrese el comando: `python3.4 nombre_programa.py`

#### *La consola interactiva de Python*

La consola interactiva de Python, puede ser invocada, no solo desde Eclipse, como se vio en la sección anterior, sino que también está disponible en cualquier terminal Linux aunque no se tenga instalado Eclipse ni PyDev. Solo es necesario que el intérprete de Python esté instalado, el cual viene por defecto en cualquier sistema Linux. Aparte de poder ejecutar un programa Python, a la consola se le llama interactiva porque le permite al programador interactuar con la ejecución del programa.

Abra una terminal de Linux como se explicó anteriormente e invoque la terminal interactiva de Python ejecutando el comando **Python3.4**

Ejecute en la terminal interactiva instrucciones como las que se muestran en la Figura 19 para experimentar algunas de las cosas que se pueden hacer en la consola interactiva de Python.

```
sisaza@sisaza-VirtualBox: ~  
sisaza@sisaza-VirtualBox:~$ python3.4  
Python 3.4.0 (default, Apr 11 2014, 13:05:18)  
[GCC 4.8.2] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> 2+1  
3  
>>> a=5  
>>> b=4  
>>> a*b  
20  
>>> c=a*b+6  
>>> c  
26  
>>> print('El valor de la variable c es ', c)  
El valor de la variable c es 26  
>>> 
```

Figura 19. Consola interactiva de Python en la terminal de Linux.

## 5. Depuración de un programe en Eclipse

En ocasiones cuando realizamos un código, o estamos analizando alguno previamente implementado, una buena opción es hacer la depuración del código, esto es analizar que ocurre línea por línea cuando se ejecuta el programa y detenerlo en el momento que queramos, y así poder detectar fallas que se pueden presentar.

Para ingresar al modo de depuración, simplemente damos clic en la parte superior de la pantalla sobre el botón que se muestra a continuación, o simplemente después de establecer nuestros checkpoints, damos clic derecho estando en cualquier parte de nuestro código, **Debug as → python run**,

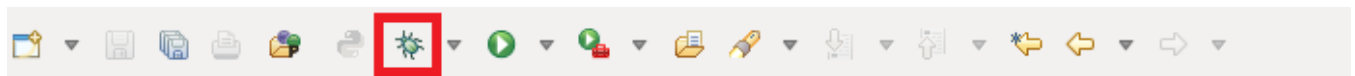


Figura 20. Ingresar el modo de depuración.

Al hacer esto, Eclipse pedirá cambiar al ambiente de depuración, en el que las ventanas con las diferentes herramientas cambian de posición en la aplicación y aparecen algunas ventanas con funcionalidades especiales.

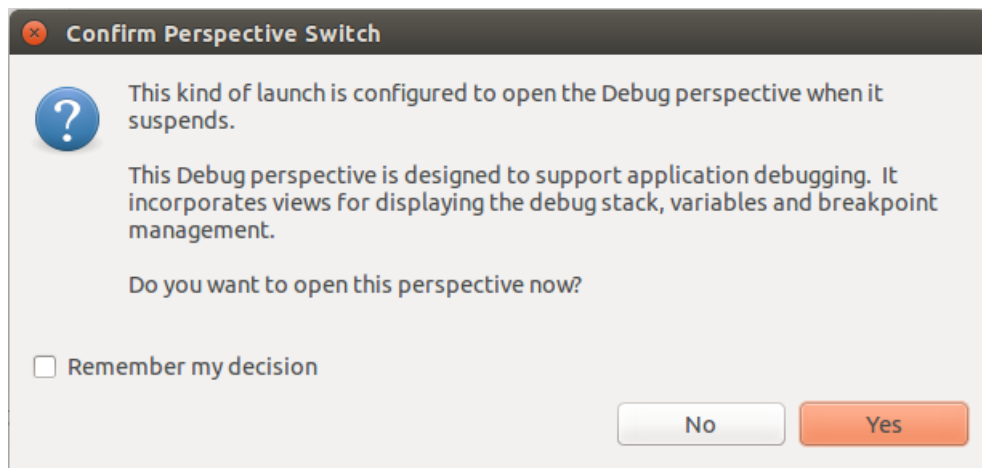


Figura 21. Cambiar al modo de depuración.

Se confirma y se debe tener en cuenta que para cambiar entre el ambiente de depuración y el de edición (PyDev), basta hacer clic en el botón con forma de tabla que se encuentra en la parte superior derecha y luego escoger el ambiente deseado

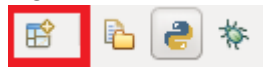


Figura 22. Cambiar el ambiente de visualización.

En este momento, debemos seleccionar la línea o las líneas en las que queremos detener la ejecución del programa, para visualizar lo que ha ocurrido hasta dicho momento, es decir agregar un 'Breakpoint'. Esto lo hacemos colocando el curso sobre la línea en la que queremos detenernos y dar clic entre los números del editor y el borde de este, hasta que aparezca una llave de color verde como se muestra señalado en la Figura 23 , este será nuestro breakpoint.

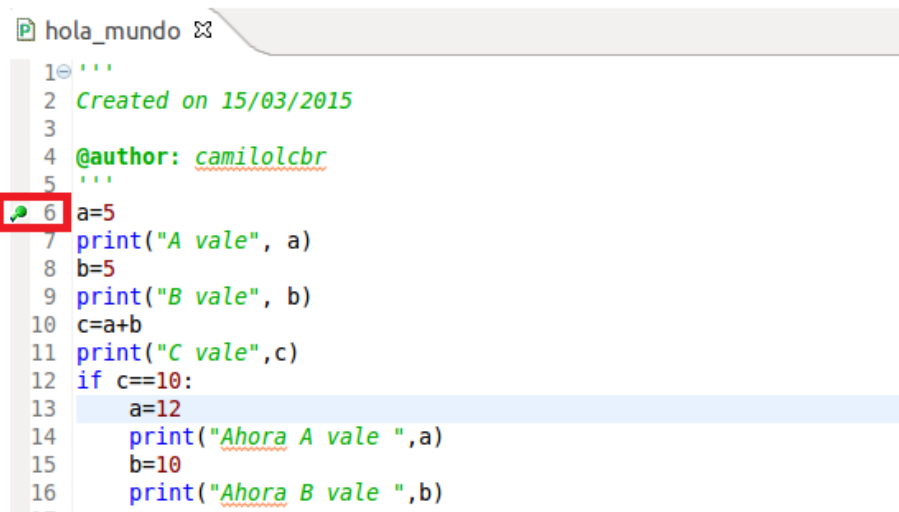
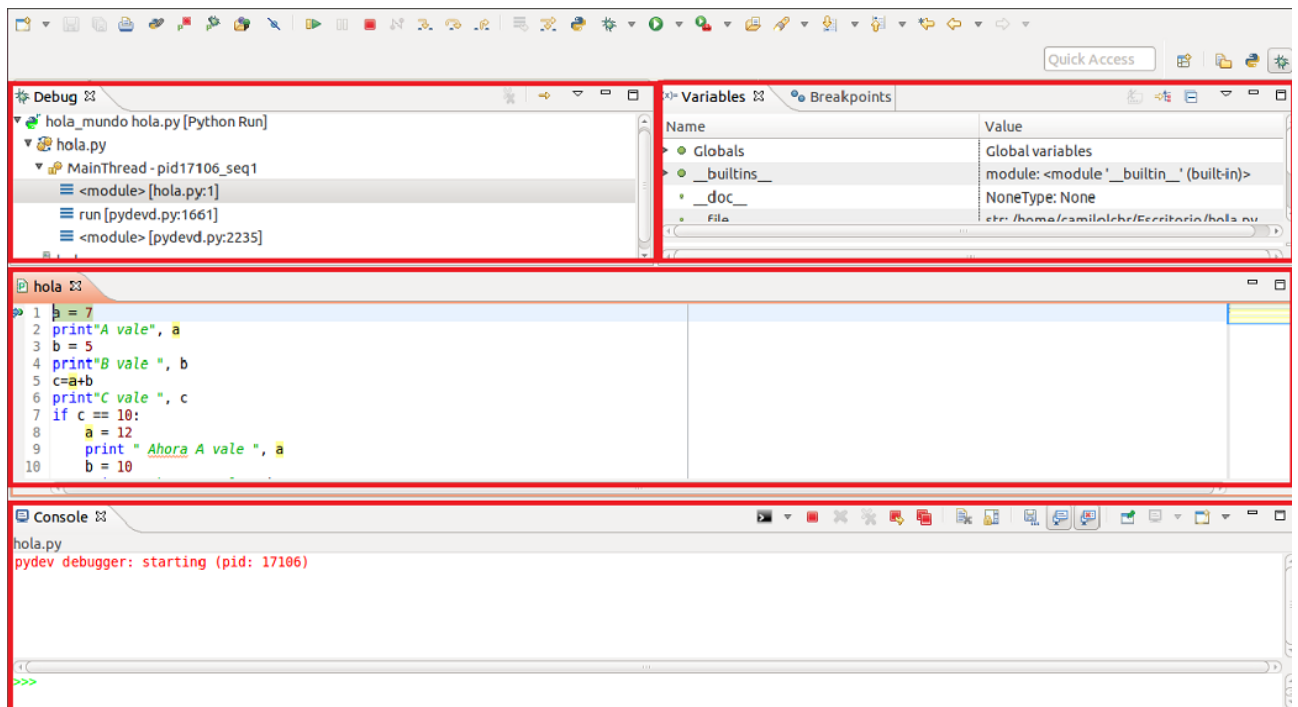


Figura 23. Agregar un breakpoint a nuestro programa.

El ambiente de depuración muestra un entorno similar al siguiente, Figura 24:

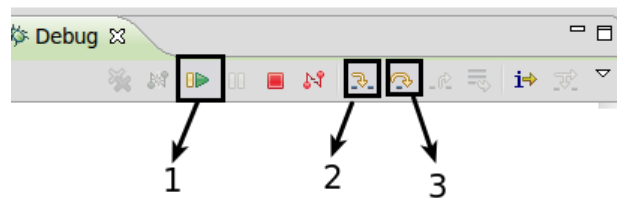


**Figura 24. Ambiente de depuración.**

- **Zona Debug:** Este cuadro presenta información acerca del proceso que se está depurando, que función se está ejecutando y cuál es su dirección, así como cuál hilo estamos depurando en dicho momento. En el caso de programas sencillos, esta ventana presenta información poco relevante, por lo tanto se puede ignorar por el momento.
- **de variables y breakpoints:** En este cuadro, se presenta el nombre de la variable, el tipo de dato de esta, así como el valor actual, también se muestran cuáles serán los puntos en los que el programa se detendrá (Breakpoint) Mas adelante veremos cómo funciona mejor.
- **Zona del código:** En la ventana del medio, se tiene nuestro código listo para correr con sus respectivos breakpoint se observa el código fuente del archivo que estamos ejecutando, más adelante profundizaremos en esto.
- **Zona de consola:** En este cuadro, se observan los mensajes que el programa imprime, este cuadro también lo utilizamos para ingresar información por el teclado al programa cuando lo requiere.

Luego de hacer clic en el botón con forma de insecto, el programa se ubica en la primera línea de código. La línea que está resaltada indica cuál es la línea que va a ser ejecutada en el siguiente paso.

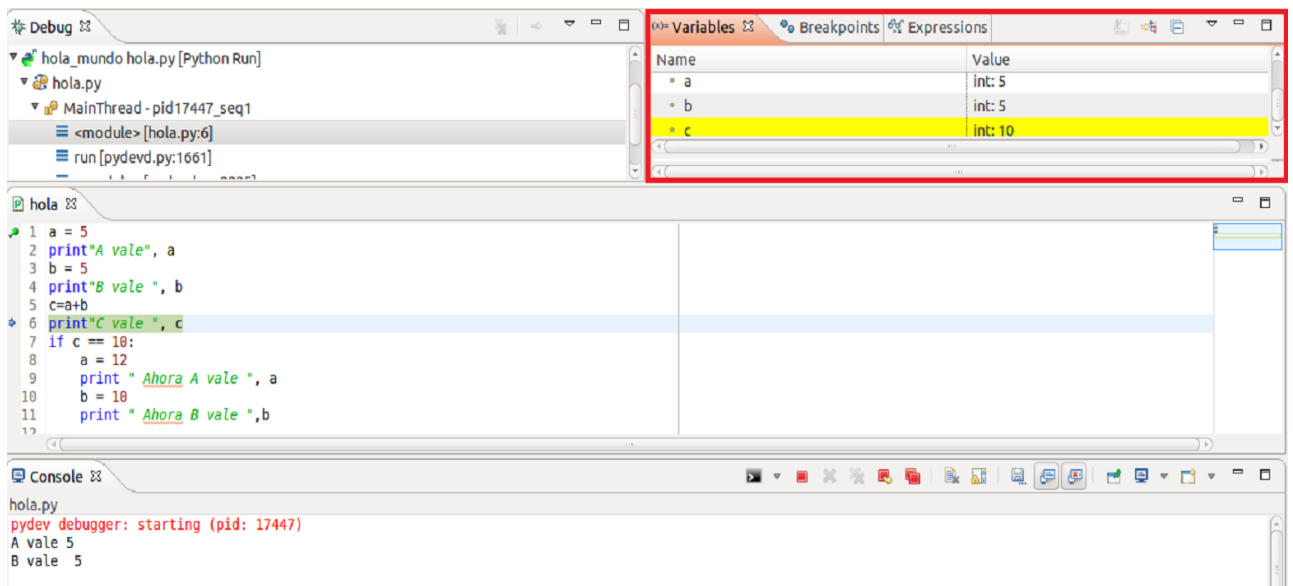
Existen diversas formas de continuar con la ejecución del programa



**Figura 25. Continuar con la depuración**

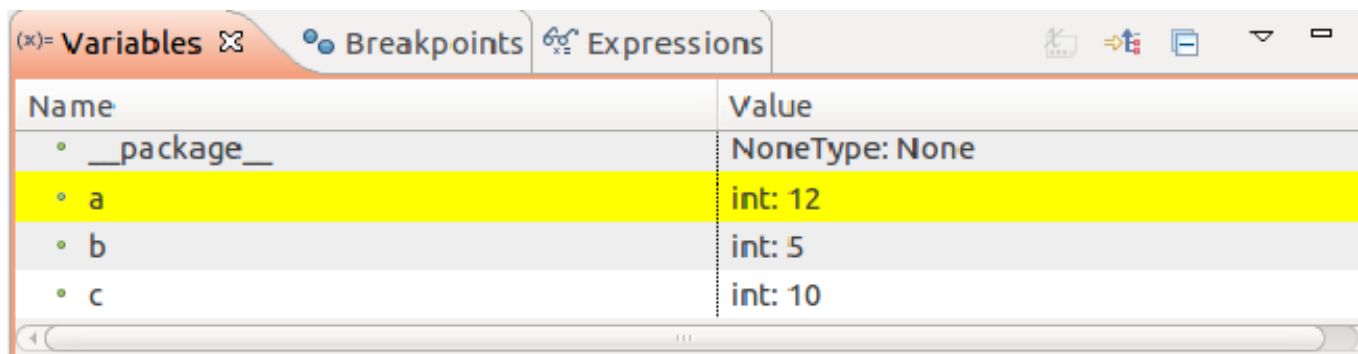
1. Resumir: Continúa la ejecución del programa hasta que encuentra el siguiente breakpoint o hasta el final del programa.
2. Step Into: Ejecuta la siguiente línea de código y en caso de que sea una función, entra al contenido de esta.
3. Step Over: Ejecuta la siguiente línea de código y en caso de que sea una función, NO entra al contenido de esta, simplemente la ejecuta toda.

Para ejecutar la primera línea de código haga clic en el botón Step Over, observará como la línea seleccionada pasa a la siguiente, y mientras tanto en la zona de variables se puede observar cómo estas van cambiando de valor a medida que se ejecuta cada línea.



**Figura 26. Realizar la depuración.**

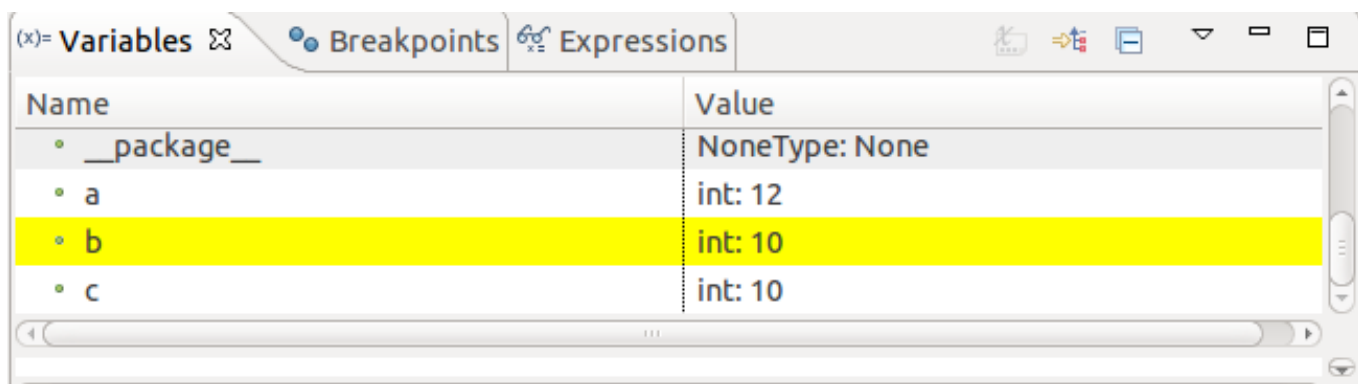
Como en este caso solo se tiene un breakpoint y se puso justo en la primera línea del código, la ejecución solo se detendrá cuando nosotros queramos hacerlo presionando el botón de stop en la parte superior de la pantalla, o cuando el programa termine, a medida que usted presiona Step Over, cada línea se ira ejecutando y usted podrá observar en la zona de variables que está sucediendo con cada una de estas, como en este caso que las variables 'a' y 'b' cambian su valor mientras que 'c' permanece constante, como se ve en la Figura 27 y en la Figura 28 .



Name	Value
• <code>__package__</code>	NoneType: None
• <code>a</code>	int: 12
• <code>b</code>	int: 5
• <code>c</code>	int: 10

Figura 27. Zona de variables.

Después de presionar de nuevo el botón de Step Over:



Name	Value
• <code>__package__</code>	NoneType: None
• <code>a</code>	int: 12
• <code>b</code>	int: 10
• <code>c</code>	int: 10

Figura 28. Zona de variables.

## 6. Programas multiarchivo.

En muchos casos un programa estará conformado por más de un archivo fuente. Para agregar otros archivos al proyecto se debe dar clic derecho sobre el nombre de nuestro proyecto, New → File, escribir el nombre de nuestro nuevo archivo e indicar la extensión que tendrá, para nuestro caso (.py).

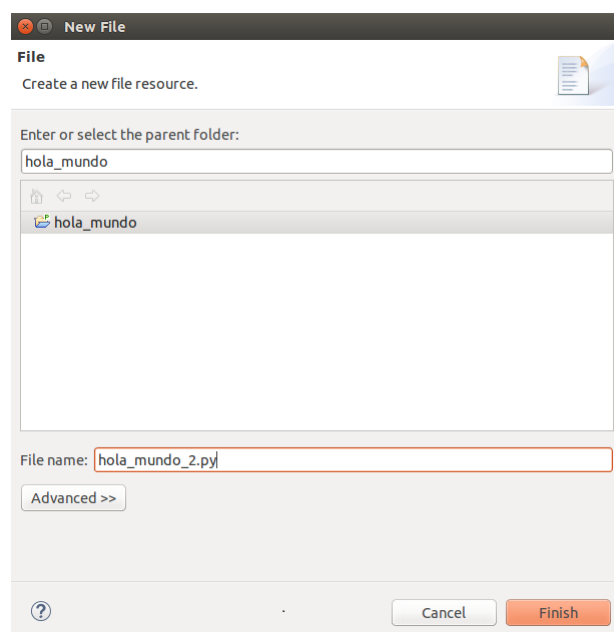
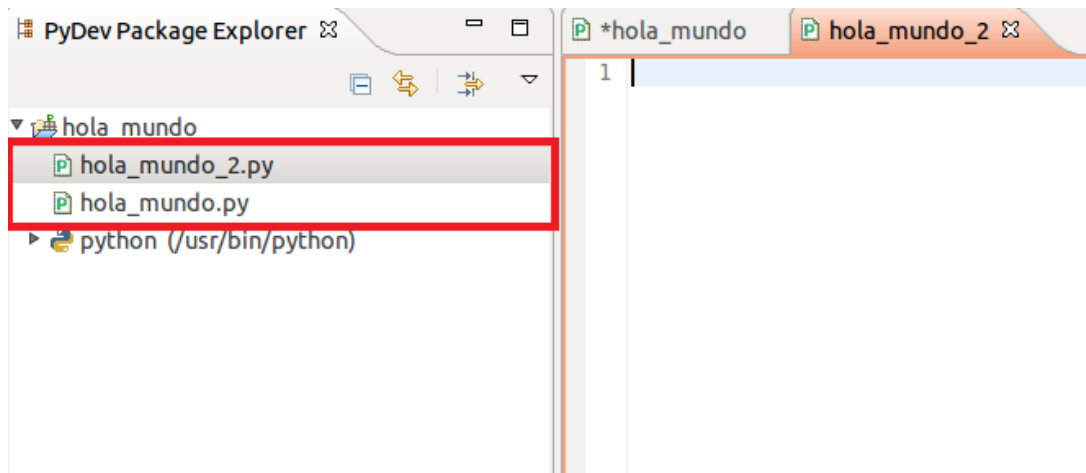


Figura 29. Adicionar un nuevo archivo al proyecto.

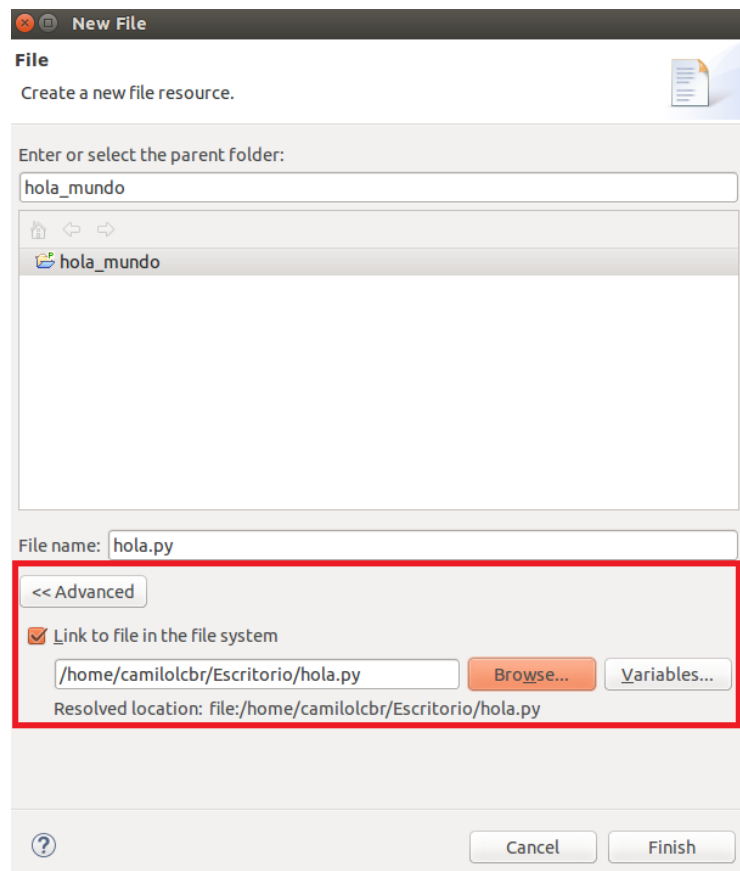


Presionamos en finalizar y podemos observar el nuevo archivo asociado a nuestro proyecto.



**Figura 30. Adicionar un nuevo archivo al proyecto.**

También, es posible agregar a un proyecto, archivos fuente ya existentes que quieran ser reutilizados o modificados. Para esto, damos clic derecho sobre el nombre de nuestro proyecto, New → File, Pero en este caso debemos presionar la opción avanzadas y seleccionamos la ubicación del archivo que queremos asociar a nuestro proyecto, como se muestra en la **Figura 31**.



**Figura 31. Adicionar un archivo existente a nuestro proyecto.**

Aceptamos y nuevamente podemos ver los diferentes archivos asociados a nuestro proyecto principal.

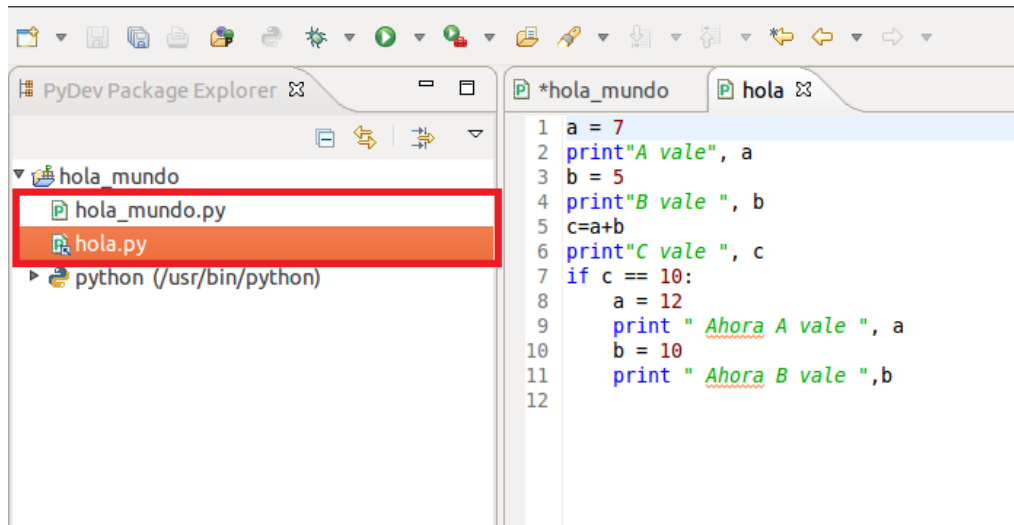


Figura 32. Adicionar un archivo existente a nuestro proyecto.