

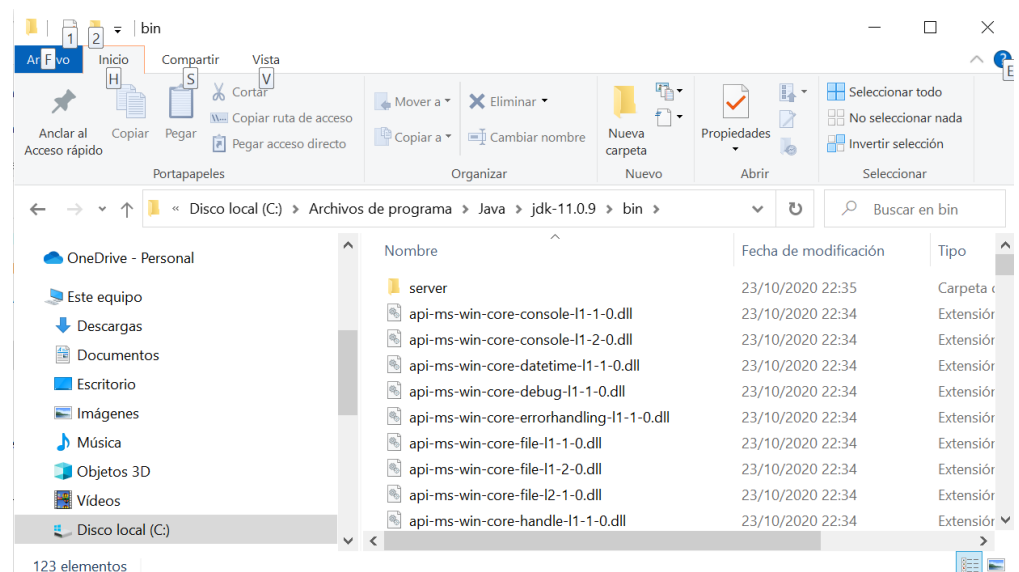
Configuración de Netbeans para el desarrollo de proyectos PHP

Paso 1: Instalación JDK 11 de Java

Primero nos instalaremos o actualizaremos la versión de Java de nuestro PC.

<https://www.oracle.com/es/java/technologies/javase/jdk11-archive-downloads.html>

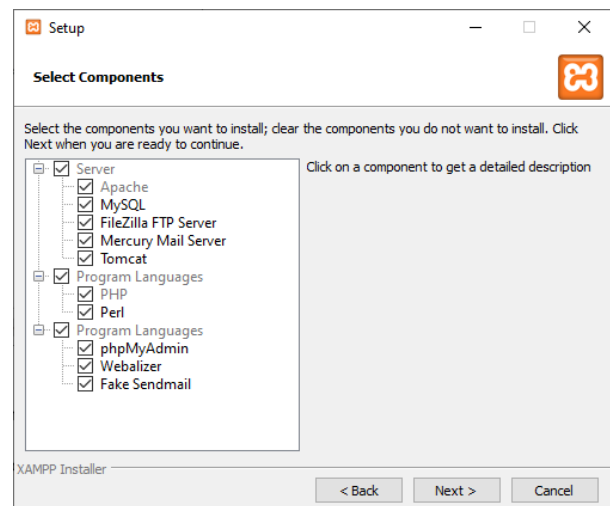
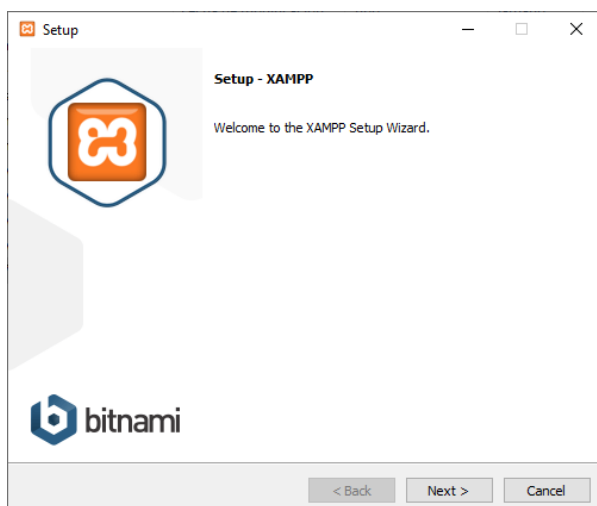
Descargamos y ejecutamos el instalador: **jdk-11.0.15_windows-x64_bin.exe** (será necesario registrarse, pero es totalmente gratuito). Una vez tenemos Java instalado. Podemos acceder a sus ficheros en esta carpeta:



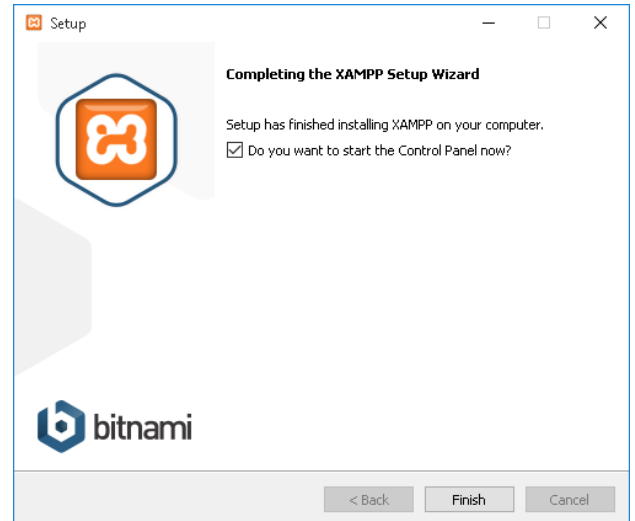
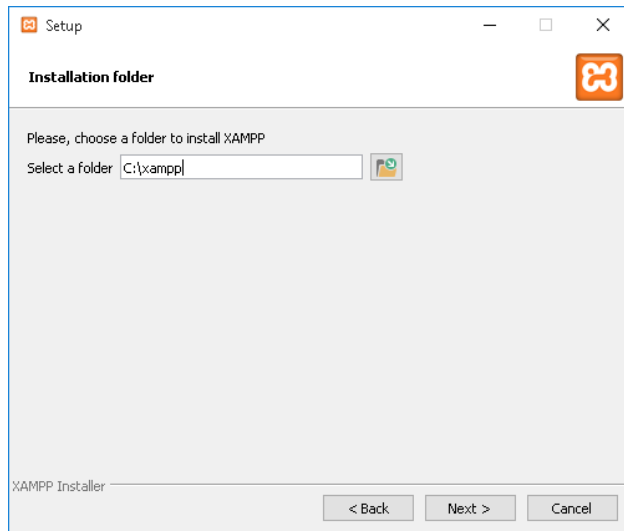
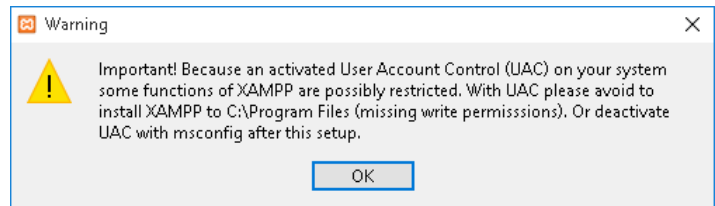
Paso 2: Instalación del servidor web local XAMPP

Como no pretendemos implantar en este momento un sistema de producción real, sino un entorno de desarrollo donde probar nuestros avances con el código, podemos trabajar con XAMPP como servidor web local para nuestro aprendizaje basado en proyectos: <https://www.apachefriends.org/es/index.html>
https://www.apachefriends.org/es/faq_windows.html

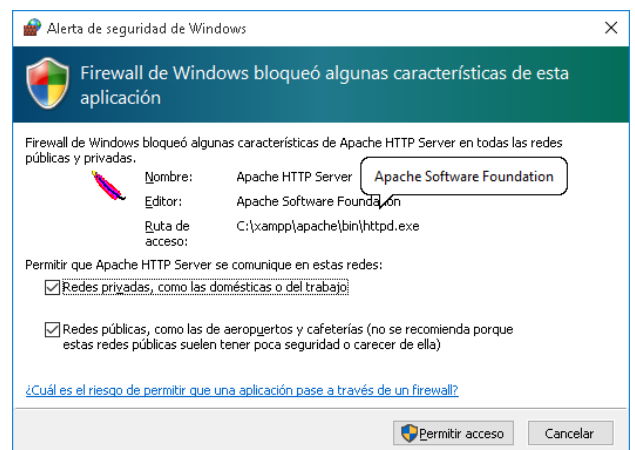
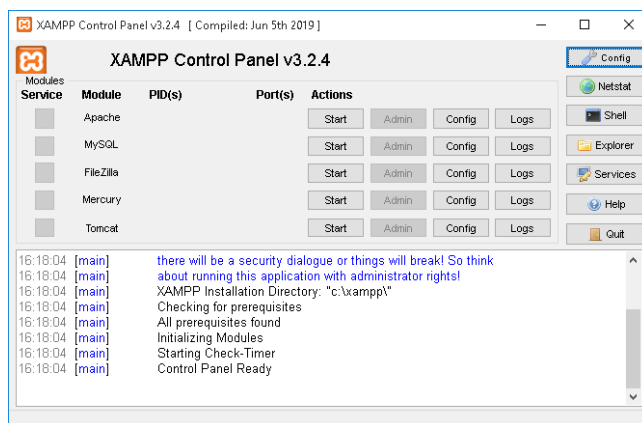
La instalación es un proceso muy simple donde no es necesario configurar ningún parámetro:



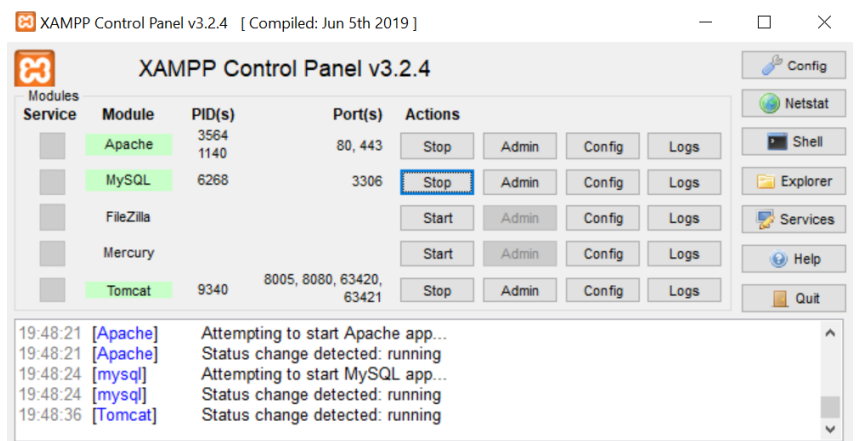
Aparecerá la siguiente advertencia sobre el control de cuentas de usuario, que aceptaremos para continuar



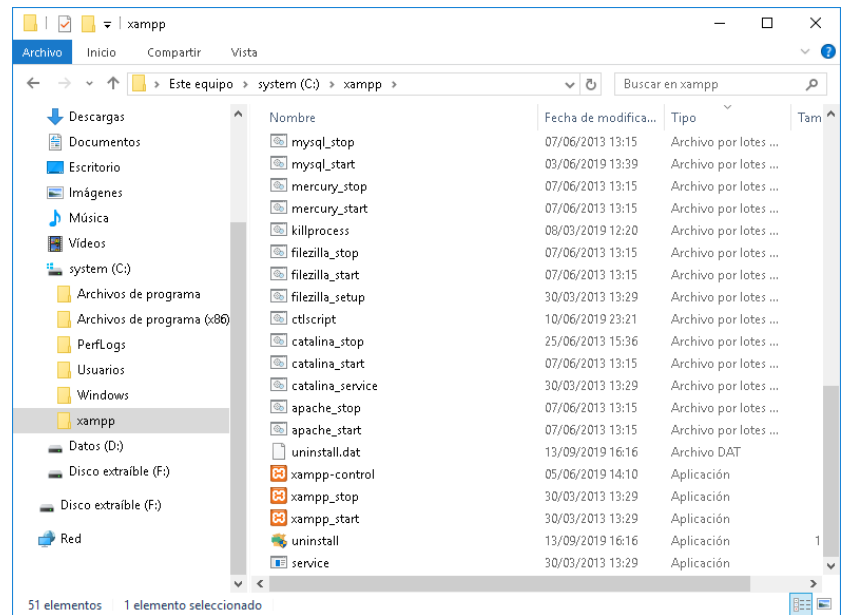
Como hemos dejado marcado el arranque del **panel de control de XAMPP** nos aparecerá de esta manera y procederemos a arrancar los servicios **Apache**, **MySQL** y **Tomcat** para comprobar que la instalación se ha realizado correctamente (se nos pedirá, para cada servicio, autorización para crear las reglas respectivas que le permitan atravesar el Firewall de Windows):



Una vez aceptadas las 3 peticiones comprobaremos que no ha habido ningún problema en el arranque de los servicios.



Para arrancar individualmente servicios o el propio panel de control de XAMPP encontraremos los ejecutables en su carpeta:



Una vez instalado el servidor web local, ya podemos trabajar de forma local como si estuviéramos conectados a un servidor web. Buscaremos la carpeta **htdocs**, ya que en ella ubicaremos nuestra página principal o home, sobre la que dirigiremos las funcionalidades que estemos desarrollando con PHP.

También crearemos en esta carpeta subdirectorios donde ubicaremos los diferentes proyectos que iremos realizando y podremos probar su funcionamiento independientemente.

Servicios incluidos en XAMPP

Apache. Al activar Apache podemos tener acceso al servidor **localhost** y el contenido de la carpeta **htdocs** será mostrado en el navegador para comprobar su aspecto y funcionalidad.

MySQL. Activa la conexión con la base de datos MySQL, realmente en las últimas versiones es MariaDB que es prácticamente idéntica, pero es código libre y no propiedad de Oracle. Si ya tenemos nuestra base de datos MySQL instalada y configuradas simplemente se trata de no activar este servicio.

FileZilla. Servidor de ficheros

Mercury. Servidor de correo

Tomcat. Servidor o contenedor de servlets y JSP's, funcionalidad basada en Java, no es un servidor de aplicaciones al estilo JBoss o JOnAS.

Ejemplos de cómo usar PHP o cambiar la configuración de Apache:

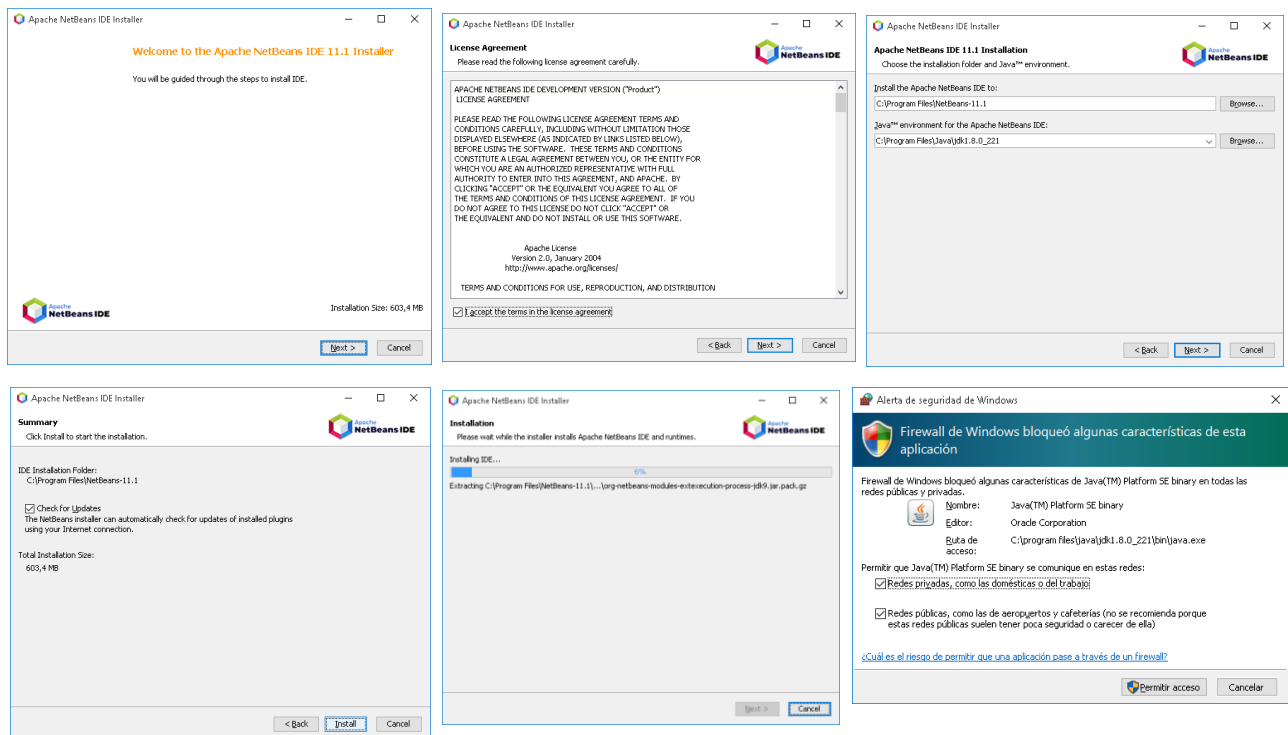
<http://www.mclibre.org/consultar/php/index.html>

Paso 3: Instalación del IDE Netbeans

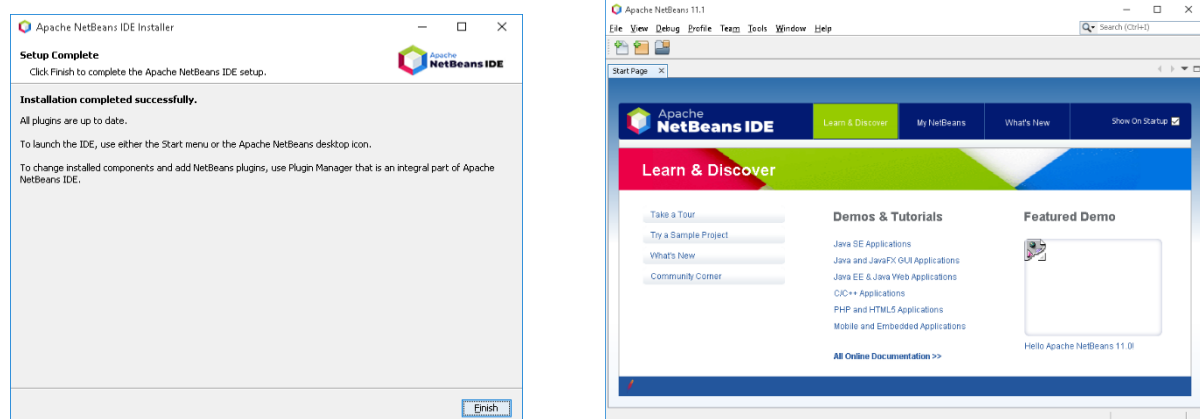
NetBeans es un entorno de desarrollo integrado (IDE) muy extendido en los proyectos profesionales, sin alcanzar el nivel de Eclipse se trata de un entorno cómodo para introducirse con los proyectos de desarrollo. Permite trabajar con proyectos en múltiples lenguajes de programación, nosotros lo utilizaremos en este curso para desarrollar en lenguaje PHP.

Podemos descargar las últimas versiones desde el siguiente enlace, descargaremos e instalaremos la última versión para Windows 64: <https://netbeans.apache.org/download/index.html>

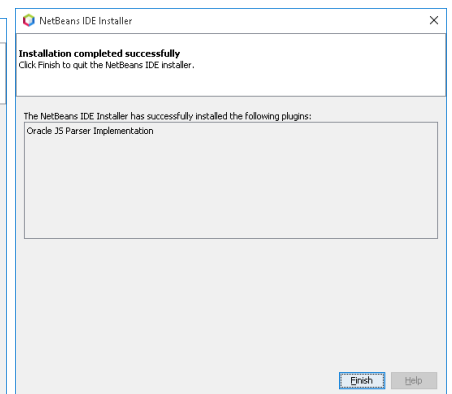
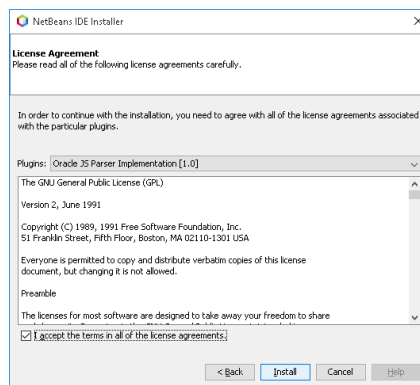
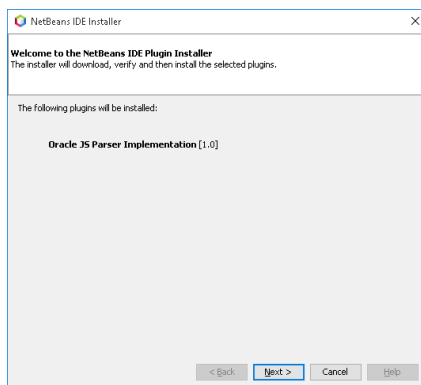
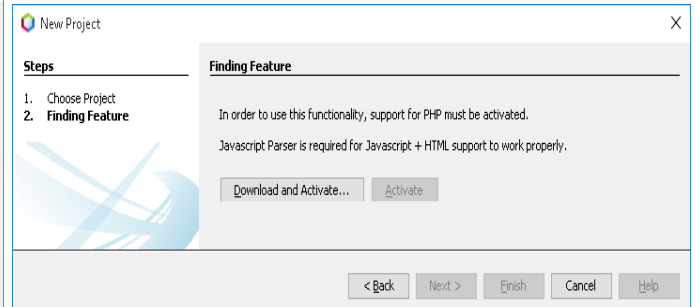
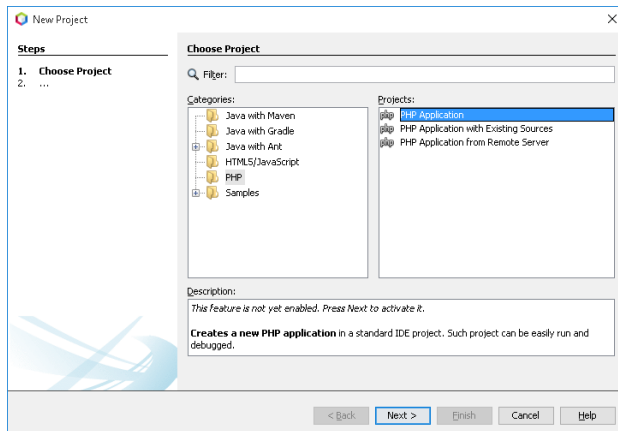
La instalación es muy simple y sin necesidad de configurar más que la autorización para atravesar el Firewall



Una vez aparece la pantalla del asistente confirmando la instalación lanzaremos la ejecución del IDE desde el icono creado en nuestro escritorio.

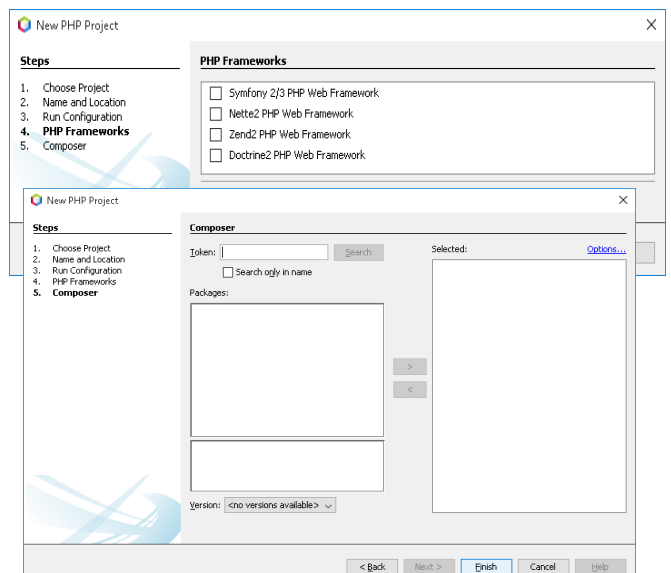
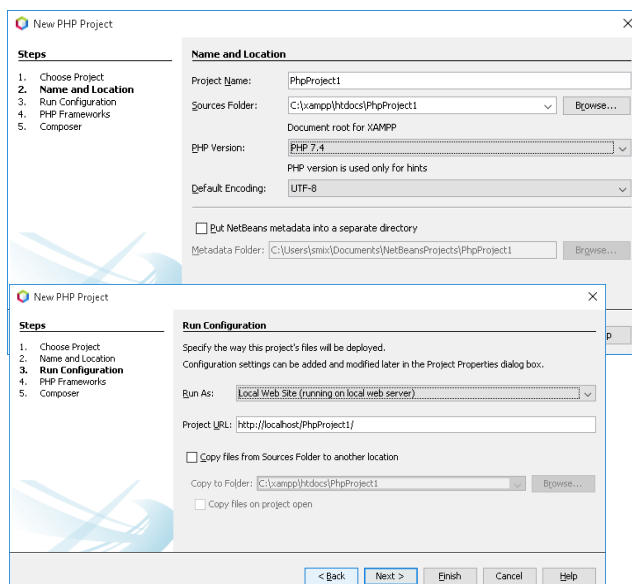


Una vez arrancado crearemos un proyecto. Iremos a **File / New Project** y buscaremos en la categoría **PHP**, donde elegiremos **PHP Application**, como por defecto NetBeans no incorpora el soporte para proyectos PHP, nos pedirá la primera vez añadir algunos pluggins y activar algunas características, en los siguientes proyectos este paso intermedio ya no será necesario.



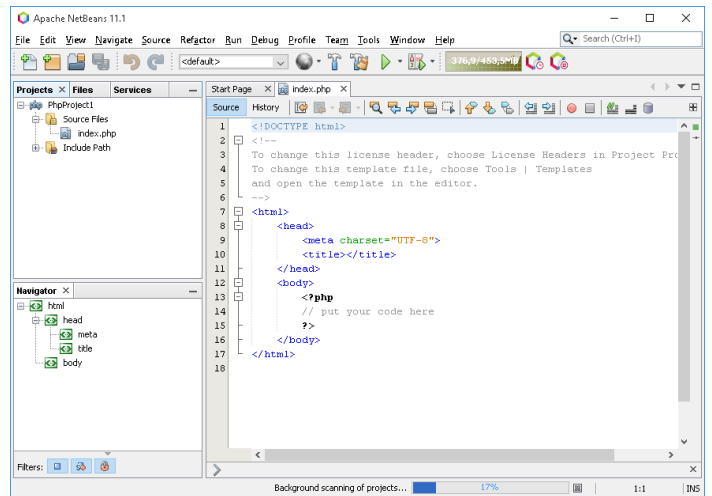
Las ventanas anteriores muestran el proceso de descarga y activación de pluggins características, solo las veremos la primera vez, a partir de este momento el asistente presentará algunas diferencias.

Donde ya nos aparece un nombre y una carpeta por defecto donde crear nuestro primer proyecto de prueba. Cambiaremos nombre y carpeta a donde queramos, pero es muy importante guardar los proyectos en C:\xampp\htdocs para que el servidor web local los ejecute en el navegador.

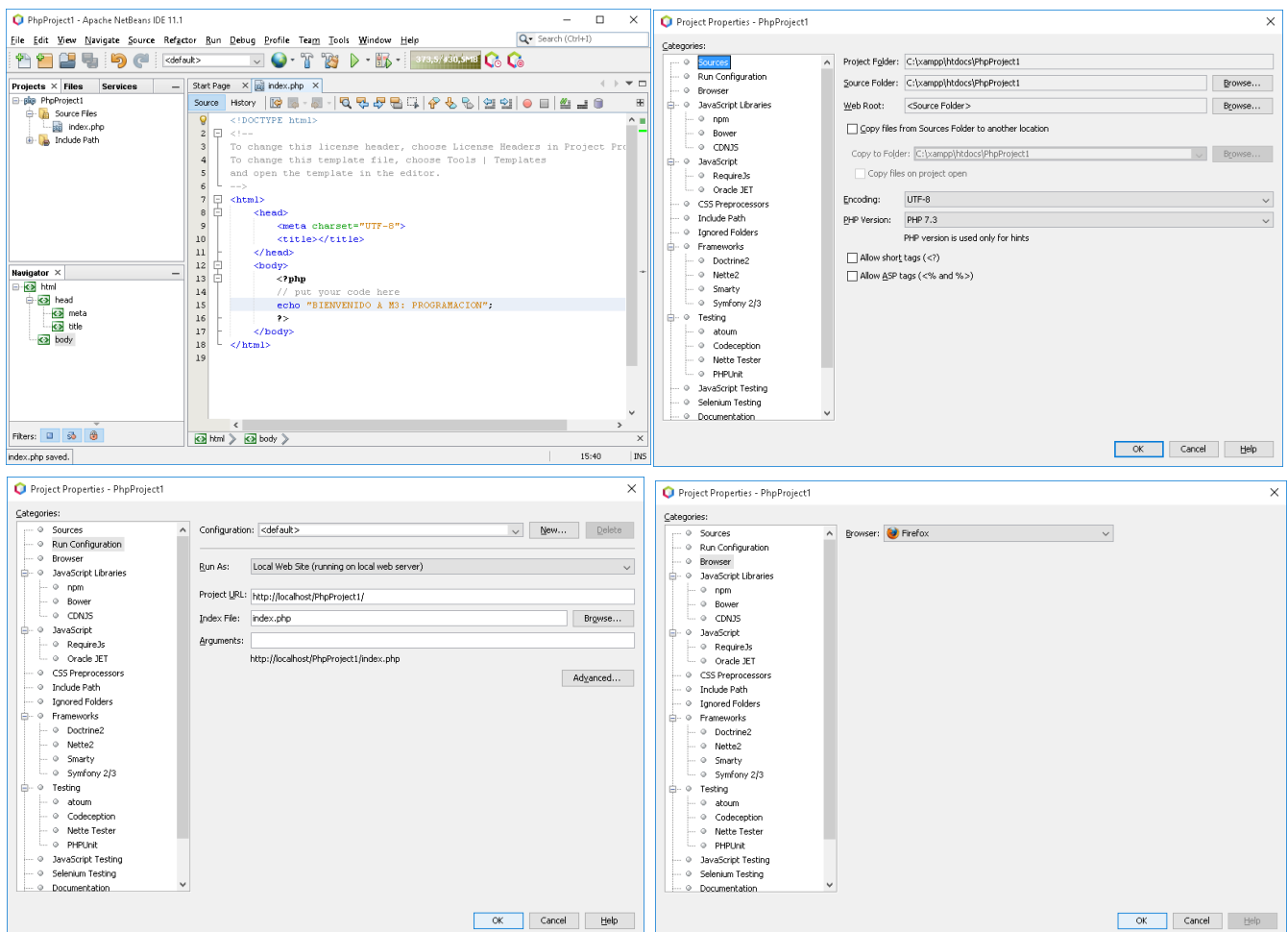


Mantendremos inicialmente la configuración de ejecución de la aplicación a sus valores por defecto, y para un proyecto básico no necesitamos el apoyo de los **Frameworks** profesionales ni del **Composer**

Y nos aparece cargado nuestro primer proyecto, con su estructura de carpetas y ficheros por defecto:

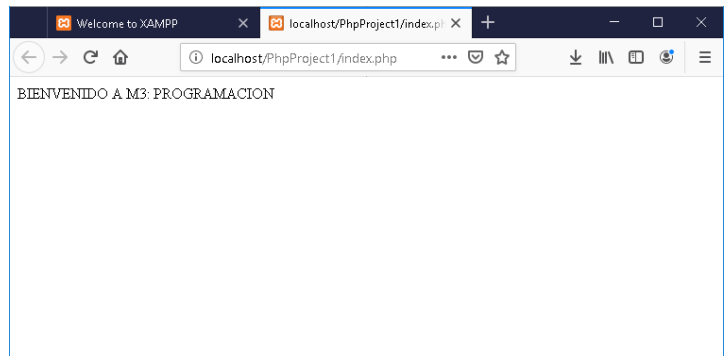


Añadimos una línea básica para que nos salude al ejecutarse, pero antes de ejecutarlo echaremos un rápido vistazo a su configuración. Hacer click con el botón derecho sobre el nombre del proyecto e ir a la última opción: **Properties**



Donde podemos revisar las configuraciones del Proyecto y modificar las que sean necesarias, lo dejaremos todo igual, excepto el navegador donde se ejecutara nuestro código, en este caso el elegido es Firefox.

Aceptamos el cambio y desde el IDE picando sobre el botón de Play de color verde veremos el resultado



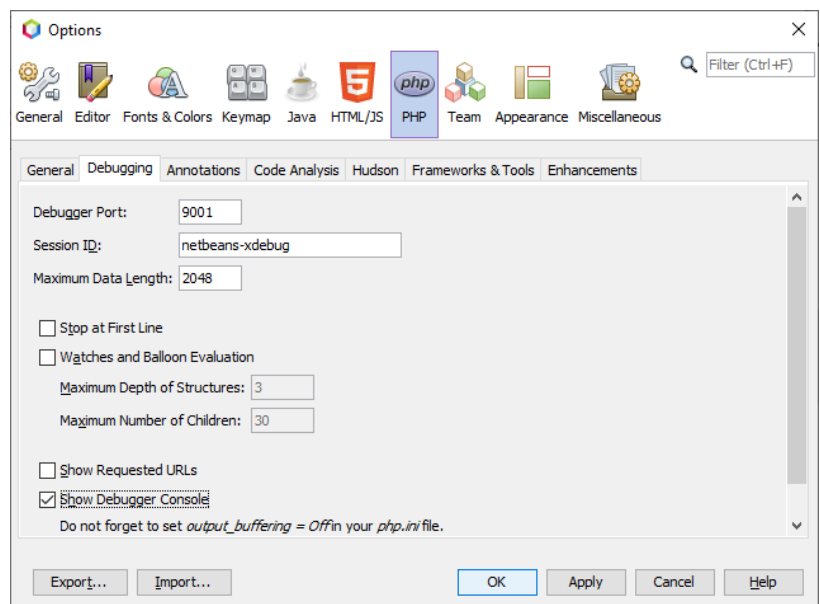
Con esto validamos todo el procedimiento realizado para obtener el entorno de desarrollo. Ya tenemos un servidor web local y un IDE para la creación de proyectos PHP que lo utiliza, para en este caso ejecutar su código sobre el navegador Firefox.

Paso 4: Instalación de XDebug para la depuración de Proyectos PHP

Una necesidad importante durante el desarrollo es la de poder depurar nuestro código.

Si vamos al menú **Options** y nos dirigimos al apartado PHP a la pestaña **Debugging** observaremos que ya está preparado para la depuración a través de **Xdebug**.

En nuevas versiones ha cambiado el puerto a 9003 y activaremos la presentación por consola **recordando modificar el parámetro indicado en php.ini**.



Pero eso no significa ni que este descargado ni configurado. Eso es lo que tendremos que hacer nosotros:

- En la página oficial de **Xdebug** (<https://xdebug.org/download.php>) encontramos las versiones disponibles. Optamos por la versión para sistemas Windows 64 bits es: **PHP 8.1 VC15 (64 bit)**.
- Copiar la dll descargada, por ejemplo **php_xdebug-3.1.5-8.1-vs16-x86_64.dll** en **C:\xampp\php\ext**
- A continuación, editar el archivo **C:\xampp\php\php.ini** (nos sirve el mismo bloc de notas, pero os recomendaría que empezaraís a probar **Notepad++** o **Sublime** para este tipo de acciones en el futuro), añadiendo al final del fichero las siguientes líneas:

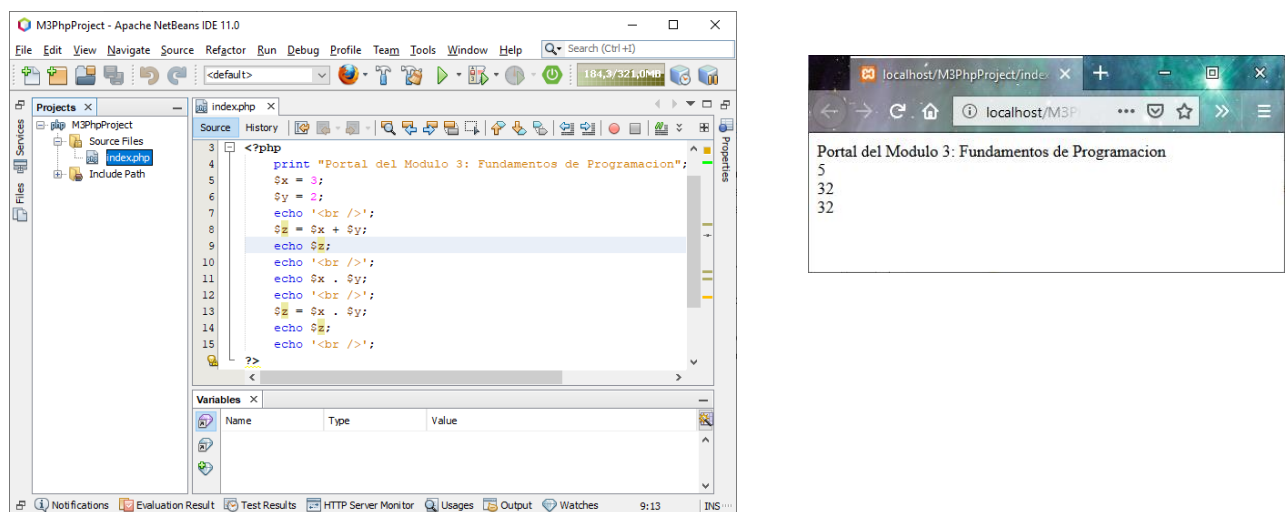
```
[XDEBUG]
zend_extension = "C:\xampp\php\ext\php_xdebug-3.1.5-8.1-vs16-x86_64.dll"
xdebug.mode=debug
xdebug.client_host=localhost
xdebug.client_port=9003
xdebug.idekey="netbeans-xdebug"
```

- Reiniciamos el servidor web y ya podemos probar su funcionamiento.

Necesitamos depurar nuestros proyectos para solucionar los errores que surgen, rastreando el origen del problema. Tradicionalmente en el proceso de depuración intervienen algunas acciones típicas como:

- Ejecutar el programa línea a línea.
- Observar la ventana de Variables o Watches, para ir siguiendo los valores que toman las variables tras cada ejecución
- Establecer un punto de breakpoint para comenzar la depuración a partir de una determinada línea

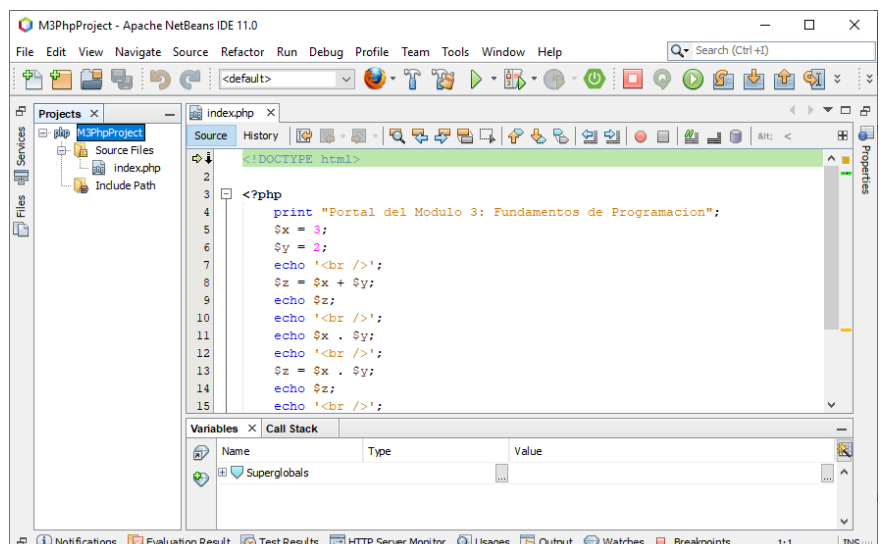
Un simple ejemplo: Programa que opera con 2 variables y el resultado de su ejecución en el navegador



El programa no presenta ningún error, pero vemos el operador . y puede que no entendamos que sucede realmente con las variables. Lo que haremos será ejecutar línea a línea su código para ir viendo que sucede con las variables y sus valores.

La depuración no se realiza con el botón verde de Play, sino el que tenemos a su derecha (**Debug Project**).

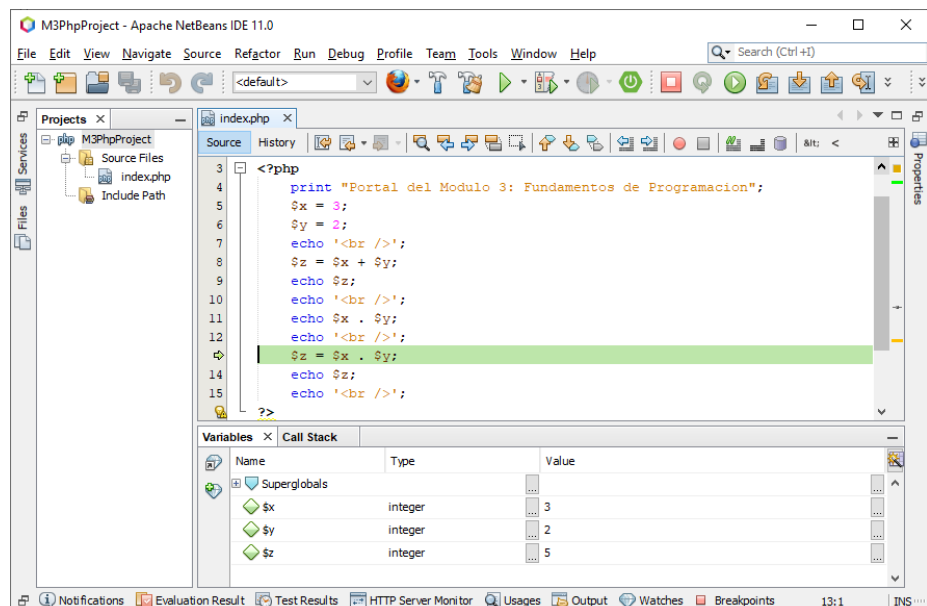
Se abrirá una sesión de depuración en nuestro navegador por defecto. Podremos avanzar línea a línea por nuestro código con F7 ejecutándolas y observando la ventana inferior **Variables**, el resultado interno de nuestras operaciones.



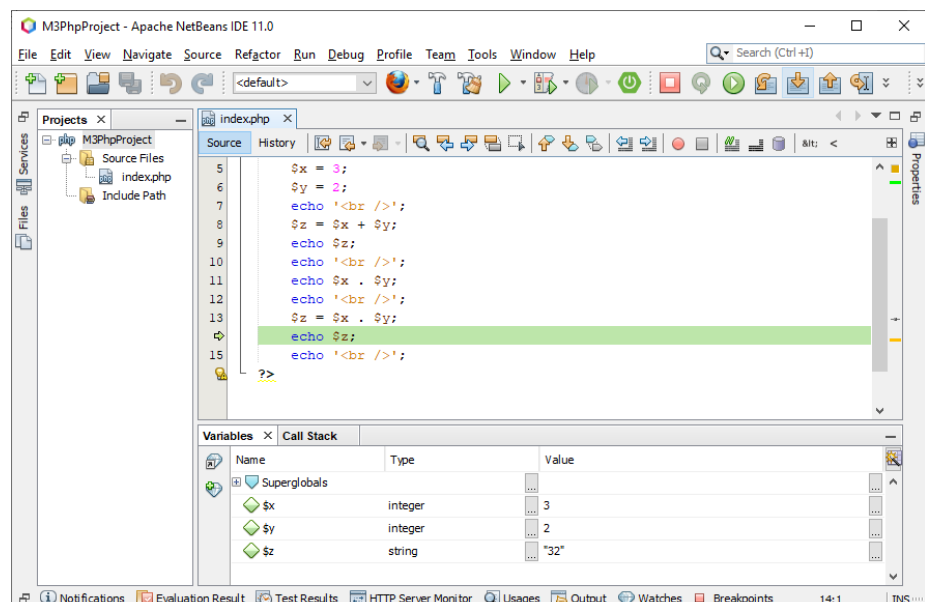
En la captura anterior podemos observar varios cambios:

- El programa se encuentra detenido justo en la línea de fondo verde, la que es apuntada por la flechita de la izquierda, esta línea aún no se ha ejecutado.
- A la derecha del botón de Debug Project se han activado un conjunto de botones que antes no estaban, que nos permiten parar la depuración, y hacerla avanzar de diferentes maneras. En lugar de avanzar línea a línea con F7 podemos hacerlo a través del botón que muestra una flechita vertical hacia abajo.
- Aparece una ventana inferior horizontal que nos permitirá observar el valor y tipo de las variables.

Comenzamos la ejecución línea a línea y observamos como ha medida que avanzamos se van creando las variables definidas con el valor y tipo implícito asignado. Nos detenemos justo antes de la instrucción que nos interesa ¿Cómo es que hemos unido dos valores enteros y los ha guardado como si se tratara de dos caracteres?



Cuando ejecutamos la línea esto es lo que observamos:

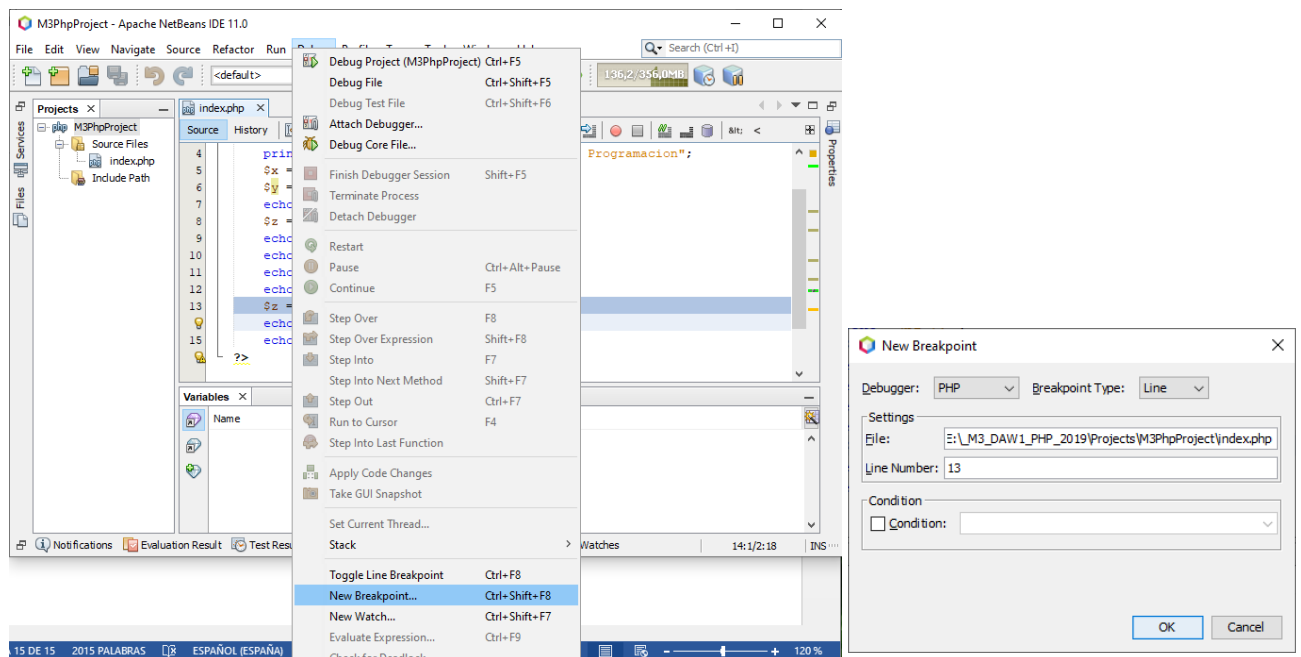


Ya sabemos que ha pasado internamente en nuestro programa, la variable \$z ha cambiado automáticamente su tipo en tiempo de ejecución para adaptarse a la operación de unión especificada por el . y por eso ahora su contenido es "32", las comillas representan un String o cadena de texto.

Para terminar con este sencillo ejemplo de proceso de depuración, faltar conocer como se establece un punto de ruptura o breakpoint. Su utilidad es la de permitir que el programa se ejecute hasta la línea establecida como breakpoint, para no tener que ejecutar una a una cada línea hasta llegar a dicho punto.

Establecimiento de un punto de ruptura y ejecución del programa hasta dicho punto:

Situaremos el cursor sobre la línea donde queremos que se detenga la ejecución, y a través del menú Debug, opción New Breakpoint o directamente con la combinación de teclas indicada establecemos el breakpoint.



Cada uno de los breakpoints añadidos quedará reflejado con un fondo de línea rosado y un cuadradito que identificará su posición. El ejecutar el **Debug Project** no observamos ninguna diferencia, pero debemos ir al nuevo icono redondo de Play de color verde y símbolo en blanco para avanzar hasta detenernos justo en el primer breakpoint establecido, y partir de ahí continuamos el proceso de depuración como siempre.

