



Configuración depurador Xdebug

Introducción:

A la hora de desarrollar código PHP sobre un servidor, en este caso de Moodle es clave el uso de un depurador. No solo permite ahorrar mucho tiempo a la hora de resolver problemas, sino que también aporta un atisbo del funcionamiento interno de un programa para facilitar una mejor comprensión de este.

En este documento se detalla cómo realizar la instalación y uso del depurador Xdebug, sobre el entorno de desarrollo Eclipse y el navegador Google Chrome.

Para comenzar es importante remarcar que se deben tener en cuenta:

- El servidor que aloja nuestro código (En este caso la máquina virtual con Moodle)
- El entorno de desarrollo (En este caso Eclipse)
- El navegador (Usaremos Chrome, se pueden utilizar otros navegadores)

Instalación inicial de Xdebug (Windows)

Para realizar la instalación inicial de Xdebug primero se debe conocer qué tipo de versión es compatible con la versión actual de Windows. Para ello, sobre la consola de Windows (Para acceder introduce “cmd” sobre la barra del explorador de archivos de Windows) debemos realizar el comando “php -i”. Es importante tener instalado PHP y configurado adecuadamente sus variables de entorno.

```
C:\Users\fjavh>php -i
phpinfo()
PHP Version => 8.0.2

System => Windows NT PC 10.0 build 18363 (Windows 10) AMD64
Build Date => Feb  3 2021 18:29:02
Build System => Microsoft Windows Server 2016 Standard [10.0.14393]
Compiler => Visual C++ 2019
Architecture => x64
```

Figura 1. Salida por consola del comando `php -i`.



Una vez hemos realizado ese comando vamos a acceder a la página:

<https://xdebug.org/wizard>

Posteriormente copiaremos toda la información obtenida en la consola de Windows y la introduciremos en el cuadro de texto de la página

Installation Wizard

This page helps you finding which file to download, and how to configure PHP to get Xdebug running. Please paste the **full** output of phpinfo() (either a copy & paste of the HTML version, the HTML source or `php -i` output) and submit the form to receive tailored download and installation instructions.

```
phpinfo()
PHP Version => 8.0.2

System => Windows NT BIOTUX-PC 10.0 build 18363 (Windows 10) AMD64
Build Date => Feb 3 2021 18:29:02
Build System => Microsoft Windows Server 2016 Standard [10.0.14393]
Compiler => Visual C++ 2019
Architecture => x64
Configure Command => cscript /nologo /e:javascript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-snap-build\dep-aux\oracle\x64\instantclient_19_9\sdk,shared" "--with-oci8-12c=c:\php-snap-build\dep-aux\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-19=c:\php-snap-build\dep-aux\oracle\x64\instantclient_19_9\sdk,shared" "--enable-object-out-dir=.\obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgsql"
Server API => Command Line Interface
Virtual Directory Support => enabled
Configuration File (php.ini) Path =>
Loaded Configuration File => C:\xampp\php\php.ini
Scan this dir for additional .ini files => (none)
Additional .ini files parsed => (none)
PHP API => 20200930
PHP Extension => 20200930
Zend Extension => 420200930
Zend Extension Build => API420200930,TS,VS16
```

Figura 2. Wizard de instalación de Xdebug.

Una vez la página ha analizado nuestra información nos devolverá la versión recomendada que debemos instalar de Xdebug y sus instrucciones necesarias.

Installation Wizard

Summary

- Xdebug installed: 3.0.2
- Server API: Command Line Interface
- Windows: yes
- Compiler: MS VS16
- Architecture: x64
- Zend Server: no
- PHP Version: 8.0.2
- Zend API nr: 420200930
- PHP API nr: 20200930
- Debug Build: no
- Thread Safe Build: yes
- OPcache Loaded: no
- Extensions directory:

Instructions

1. Download `php_xdebug-3.0.3-8.0-vs16-x86_64.dll`
2. Move the downloaded file to
3. Update `C:\xampp\php\php.ini` and change the line
`zend_extension = php_xdebug-3.0.3-8.0-vs16-x86_64.dll`

Figura 3. Información de instalación de Xdebug.



Nota: Al buscar documentación sobre la instalación de Xdebug, mucha de la información sobre la instalación pide cambiar más parámetros en “php.ini”, si utilizas la versión 3 o superior esto no será necesario.

Instalación inicial de Xdebug (Linux)

Para realizar la instalación de Xdebug en Linux deberemos ejecutar en la terminal el comando adecuado para la distribución que se utilice:

- **Alpinelinux:**
`sudo apk add php7-pecl-xdebug`
- **Arch Linux:**
`sudo pacman -Sy xdebug`
- **CentOS:**
`sudo yum install php-xdebug`
- **CentOS (Remi Repro):**
`sudo yum install php74-php-xdebug`
- **Debian** (9/stretch, testing/buster/bullseye/sid):
`sudo apt-get install php-xdebug`
- **Fedora** (32):
`sudo yum install php-xdebug`
- **Fedora** (Remi Repro):
`sudo yum install php74-php-xdebug`
- **Gentoo:**
`emerge dev-php/xdebug`
- **Manjaro** (20.1/Mikah):
`sudo pacman -S xdebug`
- **RHEL:**
`sudo yum install php-xdebug`
- **RHEL** (Remi Repro):
`sudo yum install php74-php-xdebug`
- **SUSE** (openSUSE, Enterprise):
`sudo zypper in php7-xdebug`
- **Ubuntu** (18.04 LTS/Bionic, 20.04 LTS/Focal):
`sudo apt-get install php-xdebug`
- **Ubuntu** (Ondřej Surý's PPA):
`sudo apt-get install php7.4-xdebug`

Figura 4. Comandos de instalación de Xdebug sobre Linux.

Nota: Puede que la versión de PHP en el equipo o la de Xdebug presenten incompatibilidades, es importante asegurarse de que son compatibles en el siguiente enlace:

<https://xdebug.org/docs/compat#supported-versions>



Configuración de Eclipse

En primer lugar, es necesario utilizar Eclipse PHP o el Eclipse estándar con los plug-ins para desarrollo PHP.

Posteriormente entraremos en la barra de herramientas a la opción Window y una vez allí a Preferencias. Una vez ahí entramos a la opción Debug dentro de la sección PHP. Una vez ahí veremos lo siguiente.

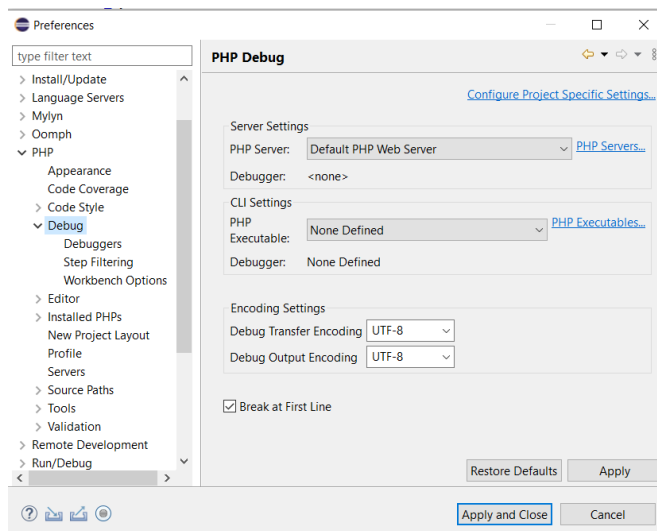


Figura 5. Configuración del debugger sobre Eclipse.

Lo primero será cambiar las opciones del servidor pulsando en PHP Servers y editando el servidor que utiliza localhost, se introducen los siguientes datos. Es importante remarcar que el puerto a utilizar es aquel que utilice nuestro servidor para conectarse (En mi caso el 443)

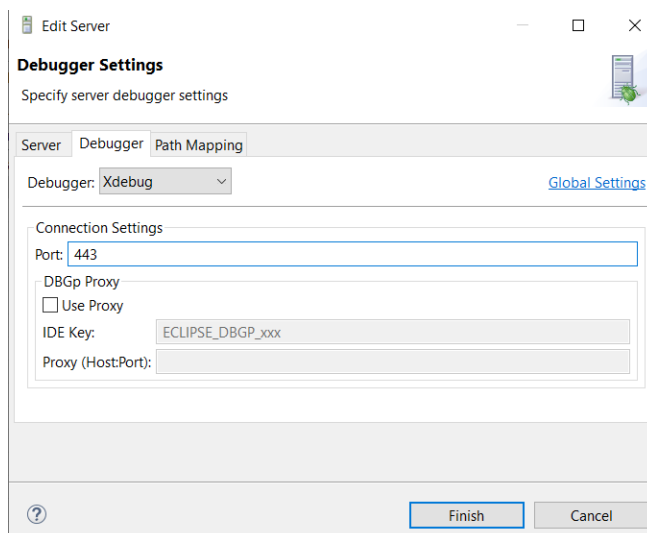


Figura 6. Configuración de servidor y puertos del debugger.



Posteriormente se deberá editar algunos parámetros en la sección Debuggers

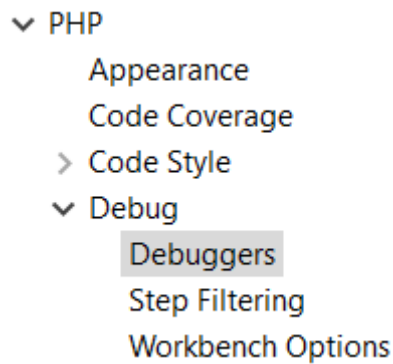


Figura 7. Sección Debuggers en configuración de Eclipse.

Una vez seleccionada esa opción se selecciona Xdebug y se entra en Configure. Es importante asegurarse que el puerto indicado es 9000. Nota: Puede que alguna aplicación ya este usando en el equipo el puerto 9000, por lo tanto, puede ser necesario cambiar este valor por otro puerto como 9001.

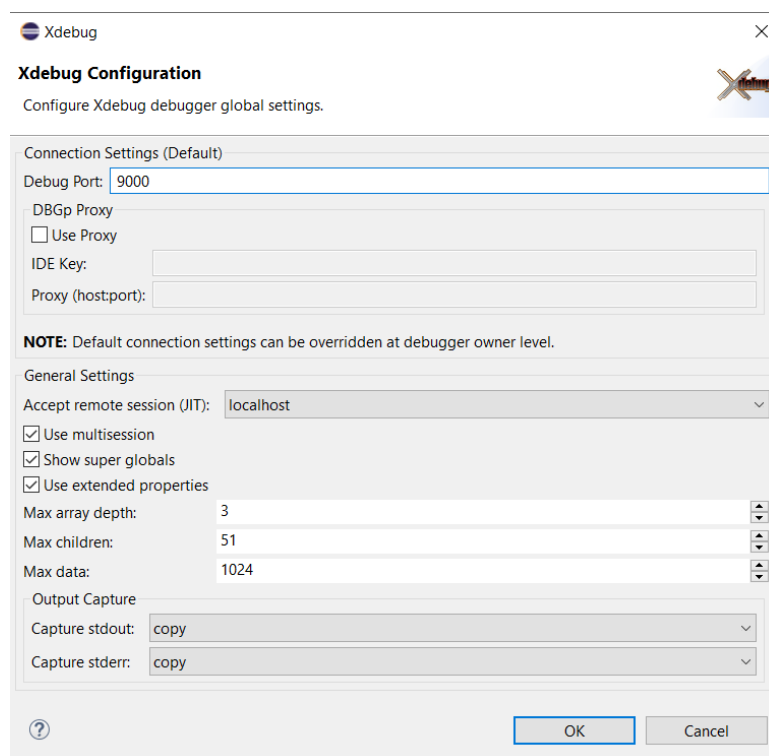


Figura 8. Configuración de Xdebug.



Instalación de la extensión de navegador (Chrome)

La extensión de Xdebug para navegador puede ser útil a la hora de ejecutar el servidor sobre el navegador y necesitar breakpoints o ciertos datos al realizar una ejecución real del programa.

Para descargarlo, en la tienda de aplicaciones de Chrome buscaremos una extensión llamada Xdebug Helper y la añadiremos al navegador.

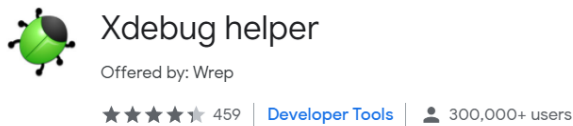


Figura 9. Extensión de Xdebug en Chrome.

Una vez añadido al navegador y se quiera hacer debug sobre un código se cambia el modo en la extensión por Debug pulsando en la barra de extensiones de Chrome.

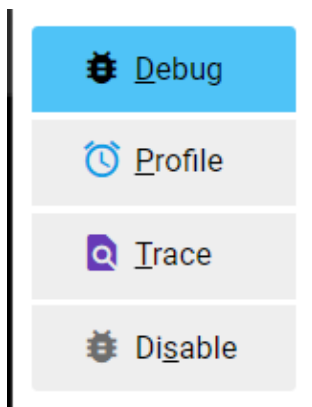


Figura 10. Modos de ejecución de la extensión sobre navegador.



Ejemplo de uso

Una vez realizado toda la instalación para realizar una depuración simplemente se deberá poner un breakpoint en donde nos interese y pulsar Debug on server.

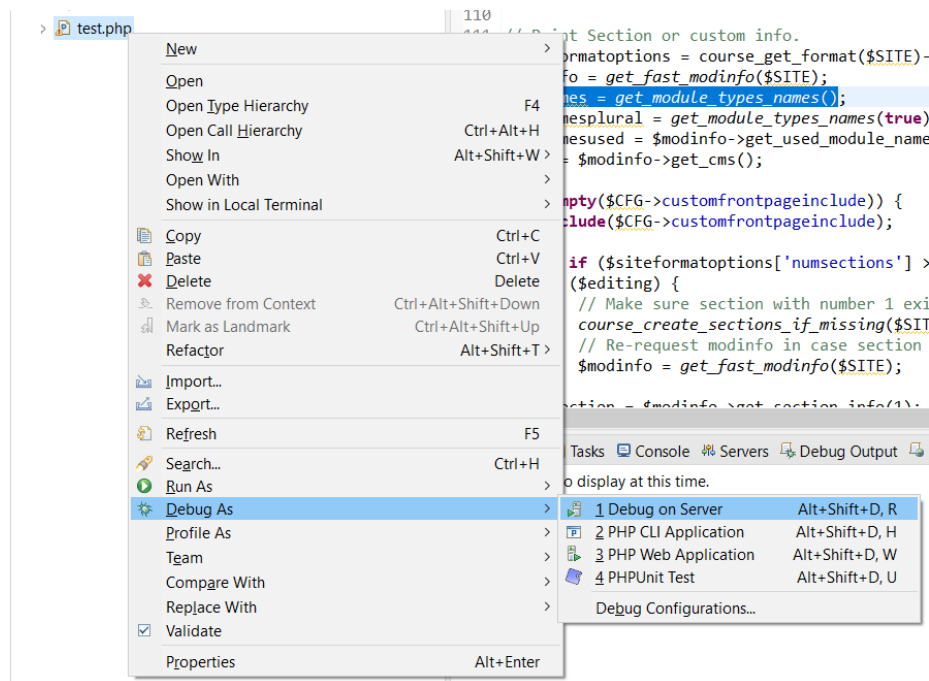


Figura 11. Ejemplo de uso.

Al realizar esto entraremos en el modo Debug de Eclipse cuando se ejecute el código que se quiera analizar mientras se navega por la página.

Configuración de carpeta compartida desde VM VirtualBox

Para realizar la programación de Evalcode es bastante importante realizar la ejecución del código desde los archivos que se encuentran desde el servidor, para ello se va a emplear una carpeta compartida para poder entrar desde el sistema operativo que utilice el entorno de desarrollo a los ficheros del servidor.

En primer lugar, se debe Insertar imagen de CD de Guest Additions, viene indicado en la siguiente figura:

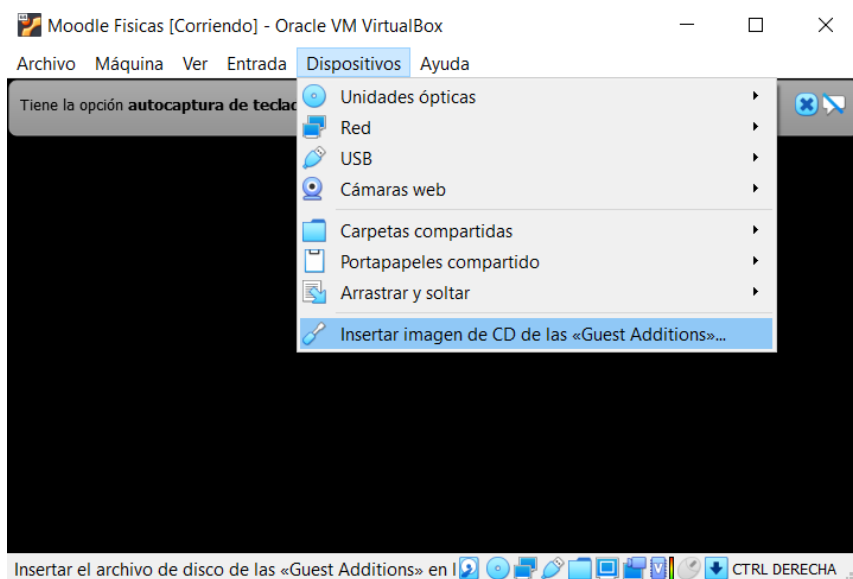


Figura 12. Insertar Guest Additions

Una vez hecho esto se debe acceder a la sección de carpetas compartidas y crear una carpeta:

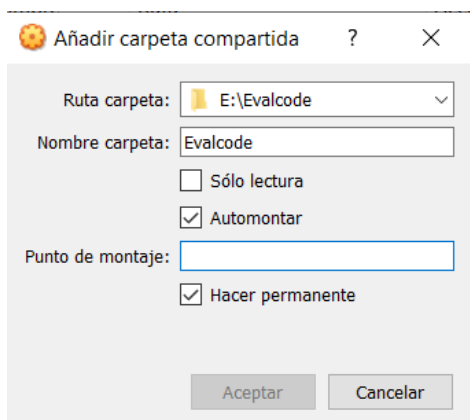
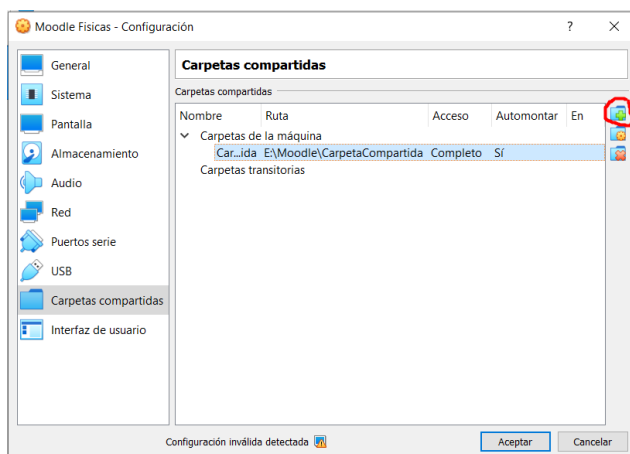


Figura 13 y 14. Creación de carpetas compartidas en VM VirtualBox



IMPORTANTE: Antes de hacer este paso es recomendable guardar el contenido de la carpeta sobre la que se desea montar del guest, puesto que el contenido de la carpeta host pasará a estar dentro del sistema de ficheros del guest, es decir, se borrará.

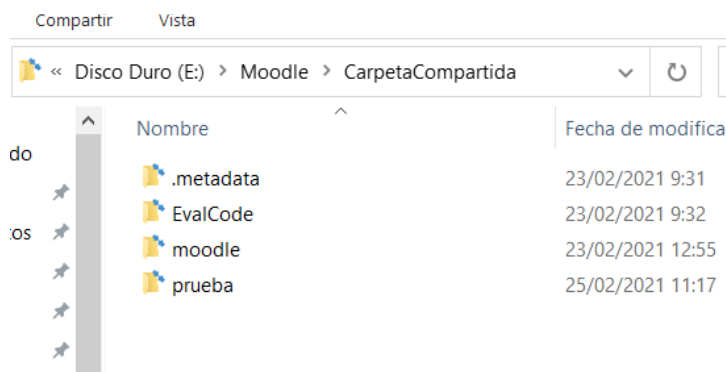
Una vez hecho esto se configura la carpeta con la Ruta en el host y el nombre que nos interese, es importante remarcar que, si seleccionamos la opción Automontar, la carpeta se montará sobre un directorio por defecto en el sistema operativo, en caso de querer utilizar una carpeta del sistema de ficheros Guest, es decir, utilizar la carpeta donde se está ejecutando Evalcode dentro del servidor se debe montar la carpeta utilizando el siguiente comando:

```
sudo mount -t vboxsf [[nombre de la carpeta virtual]] [[directorio del guest donde estará la carpeta compartida]]
```

Ejemplo:

```
sudo mount -t vboxsf CarpetaCompartida /www/html/moodle
```

Una vez realizado esto, se puede copiar desde el guest lo que había anteriormente en la carpeta sobre la que se ha montado y se podrá empezar con el desarrollo utilizando el workspace de Eclipse sobre la carpeta compartida.



```
moodle@moodle:/var/www/html$ ls
index.php moodle moodle-latest-33.zip prueba.php
```

Figura 15 y 16. Carpeta Moodle compartida entre Windows y Ubuntu.



Bibliografía

- [1] Especificación de Xdebug sobre Eclipse: https://wiki.eclipse.org/Debugging_using_XDebug
- [2] Documentación e instalación: <https://xdebug.org/>