

Cómo depurar código PHP Con Visual Studio Code

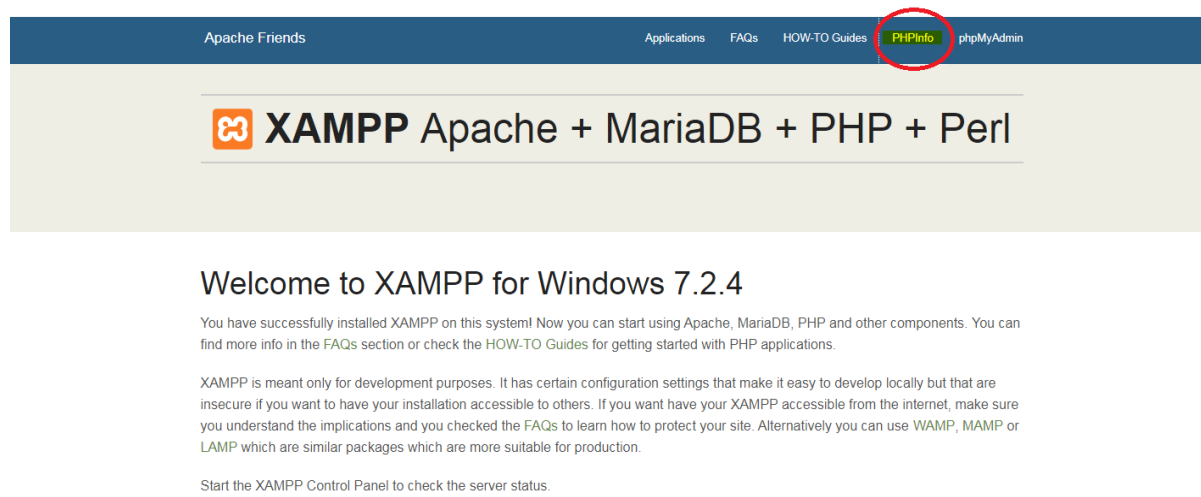
(extraído de <https://www.espai.es/blog/2018/09/como-depurar-codigo-php-con-visual-studio-code/>)

Para depurar código PHP, utilizaremos la extensión Xdebug, distribuida bajo **The PHP License 3**, que proporciona un soporte muy completo para la depuración de nuestros scripts.

Instalación de Xdebug

Para ello, necesitaremos descargar la versión de Xdebug, adecuada a nuestra versión de PHP. Hay que tener en cuenta que necesitaremos tener PHP 7.0 o superior.

Empezaremos averiguando que versión de Xdebug nos corresponde para la instalación de PHP de nuestra máquina. Si tenemos instalado el XAMPP es muy sencillo, tan solo tenemos que abrir el navegador y escribiremos en la barra de direcciones localhost, y en el menú de la página que aparecerá pulsaremos la opción de PHPInfo.



Si estamos usando Windows, con `ctrl+a`, seleccionaremos todo el texto y con `ctrl+c` lo copiaremos en el portapapeles.

PHP Version 7.2.4



System	Windows NT ESPAIB-A204 6.1 build 7601 (Windows 7 Professional Edition Service Pack 1) i586
Build Date	Mar 28 2018 04:23:32
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x86
Configure Command	cmdscript /nologo configure.js "--enable-snapshot-build"--enable-debug-pack* "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\v86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\v86\instantclient_12_1\sdk,shared" "--enable-object-out-dir=.obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,TS,VC15
PHP Extension Build	API20170718,TS,VC15
Debug Build	no
Thread Safety	enabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
OTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, zlib.*

Seguidamente en una nueva pestaña iremos a la siguiente dirección <https://xdebug.org/download.php> y haremos click en el enlace de custom installation instructions.

← → Es seguro | <https://xdebug.org/download.php>



XDEBUG EXTENSION FOR PHP | DOWNLOADS

[home](#) | [updates](#) | [download](#) | [documentation](#) | [contributing](#) | [license](#) | [support](#) | [issue tracker](#)

SOURCE

The Xdebug extension helps you debugging your script by providing a lot of valuable debug information. The debug information that Xdebug can provide includes the following:

Xdebug is hosted in GIT. The source code can be browsed through [github](#) and can be checked out with:

```
git clone git://github.com/xdebug/xdebug.git
```

RELEASES

The Windows binaries generally work for every mini release for the mentioned PHP version, although the extension is built against the most current PHP version at that time. The VCx marker tells with which compiler the extension was built, and Non-thread-safe whether ZTS was disabled. Those qualifiers need to match the PHP version you're using. If you don't know which one you need, please refer to the custom installation instructions.

If you like Xdebug, please consider giving a donation.

[BECOME A PATRON](#)

Nos aparecerá la siguiente pantalla, en la que copiaremos el contenido del portapapeles anterior, y pulsaremos el botón Analyse my phpinfo() output.

TAILORED INSTALLATION INSTRUCTIONS

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. Do **not** paste the raw HTML (from view-source) into the form.

```
contributors are listed in the manual.
PHP Quality Assurance Team
Ilia Alshanetsky, Joerg Behrens, Antony Dovgal, Stefan Esser, Moriyoshi
Koizumi, Magnus Maatta, Sebastian Nohn, Derick Rethans, Melvyn Sopacua, Jani
Taskinen, Pierre-Alain Joye, Dmitry Stogov, Felipe Pena, David Soria Parra,
Stanislav Malyshev, Julien Pauli, Stephen Zarkos, Anatol Belski, Remi
Collet, Ferenc Kovacs
Websites and Infrastructure team
PHP Websites Team      Rasmus Lerdorf, Hannes Magnusson, Philip Olson,
Lukas Kahwe Smith, Pierre-Alain Joye, Kalle Sommer Nielsen, Peter Cowburn,
Adam Harvey, Ferenc Kovacs, Levi Morrison
Event Maintainers      Damien Seguy, Daniel P. Brown
Network Infrastructure   Daniel P. Brown
Windows Infrastructure   Alex Schoenmaker
PHP License
This program is free software; you can redistribute it and/or modify it
under the terms of the PHP License as published by the PHP Group and
included in the distribution in the file: LICENSE

This program is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.

If you did not receive a copy of the PHP license, or have any questions
about PHP licensing, please contact license@php.net.
```

The information that you upload will not be stored. The script will only use a few regular expressions to analyse the output and provide you with instructions. You can see the code [here](#).

Analyse my phpinfo() output

If you like Xdebug, please consider giving a [donation](#).

BECOME A PATRON

Esto nos mostrará una pantalla con las instrucciones que necesitaremos para usar la extensión Xdebug.

XDEBUG EXTENSION FOR PHP | DOCUMENTATION | INSTALLATION

[home](#) | [updates](#) | [download](#) | [documentation](#) | [contributing](#) | [license](#) | [support](#) | [issue tracker](#)

TAILORED INSTALLATION INSTRUCTIONS

SUMMARY

- **Xdebug installed:** no
- **Server API:** Apache 2.0 Handler
- **Windows:** yes - Compiler: MS VC15 - Architecture: x86
- **Zend Server:** no
- **PHP Version:** 7.2.4
- **Zend API nr:** 320170718
- **PHP API nr:** 20170718
- **Debug Build:** no
- **Thread Safe Build:** yes
- **Configuration File Path:** C:\Windows
- **Configuration File:** C:\xampp\php\php.ini
- **Extensions directory:** C:\xampp\php\ext

INSTRUCTIONS

1. Download `php_xdebug-2.6.1-7.2-vc15.dll`
2. Move the downloaded file to C:\xampp\php\ext
3. Edit `C:\xampp\php\php.ini` and add the line
`zend_extension = C:\xampp\php\ext\php_xdebug-2.6.1-7.2-vc15.dll`
4. Restart the webserver

If you like Xdebug, please consider giving a [donation](#).

BECOME A PATRON

Tal y como nos indican, lo primero que hay que hacer es descargar el archivo que nos sugieren y copiarlo en la carpeta que nos indica en Extensions directory (en mi caso, la carpeta C:\xampp\php\ext).

Ahora editaremos el fichero C:\xampp\php\php.ini, y añadiremos al final las siguientes líneas.

[XDebug]


```
xdebug.remote_enable = 1
```

```
xdebug.remote_autostart = 1
```

```
zend_extension = C:\xampp\php\ext\php_xdebug-2.6.1-7.2-vc15.dll
```


NOTA: el valor de zend_extension debe de ser el que nos indica la página de Xdebug en sus instrucciones.

Para verificar que todo el proceso anterior ha tenido éxito, consultaremos otra vez la opción de PHPInfo, y comprobaremos que en el texto que sale al lado de la imagen de zendengine, aparece “with Xdebug” tal y como se muestra en la siguiente imagen (otra forma de comprobarlo es buscar el texto xdebug en esta página).

PHP Version 7.2.4

System	Windows NT ESPAIB-A204 6.1 build 7601 (Windows 7 Professional Edition Service Pack 1) i586
Build Date	Mar 28 2018 04:23:32
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x86\instantclient_12_1sdk;shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x86\instantclient_12_1sdk;shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,TS,VC15
PHP Extension Build	API20170718,TS,VC15
Debug Build	no
Thread Safety	enabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, zlib.*, bzip2.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.2.0 Copyright (c) 1998-2018 Zend Technologies
with Xdebug v2.6.1 Copyright (c) 2002-2018, by Derick Rethans

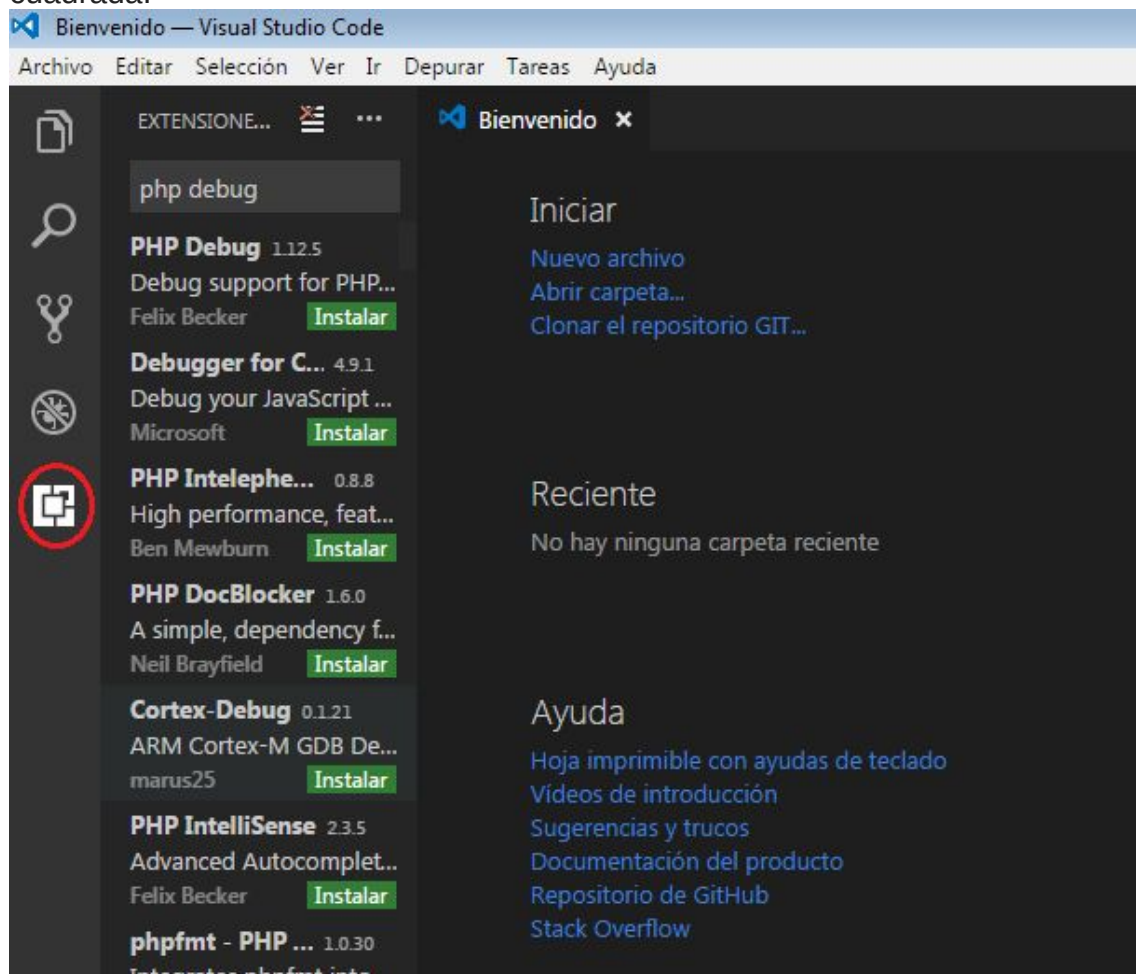


Configuración de Visual Studio Code para depurar código

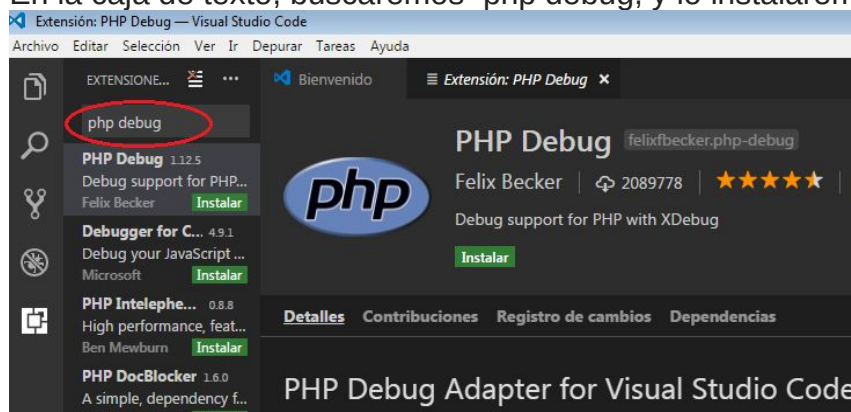
PHP

Ahora ya estamos en condiciones para poder depurar con Visual Studio Code. Para ello debemos asegurarnos de tener instalado en este editor la extensión de «PHP Debug».

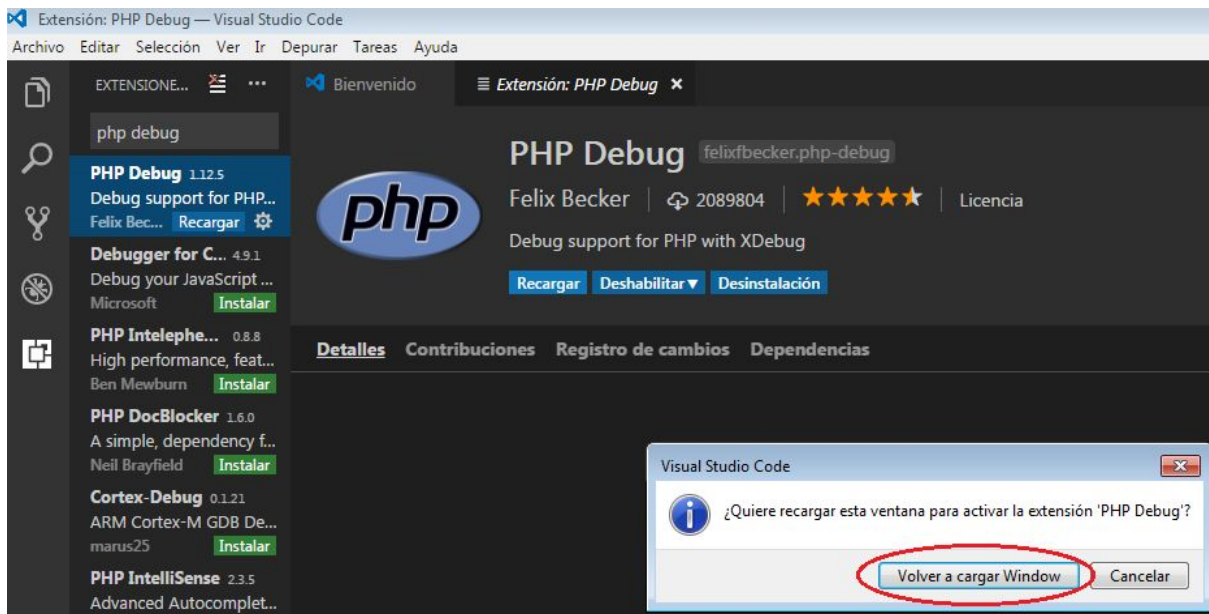
Las extensiones las buscaremos usando el icono del menú de la izquierda con forma cuadrada.



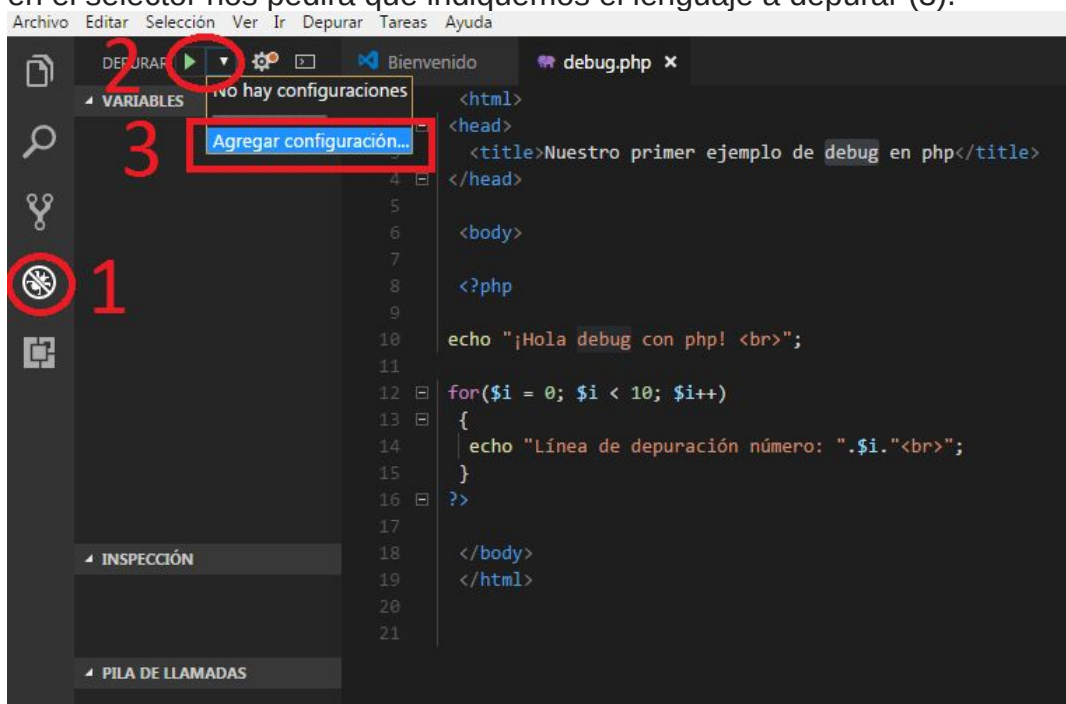
En la caja de texto, buscaremos php debug, y lo instalaremos.



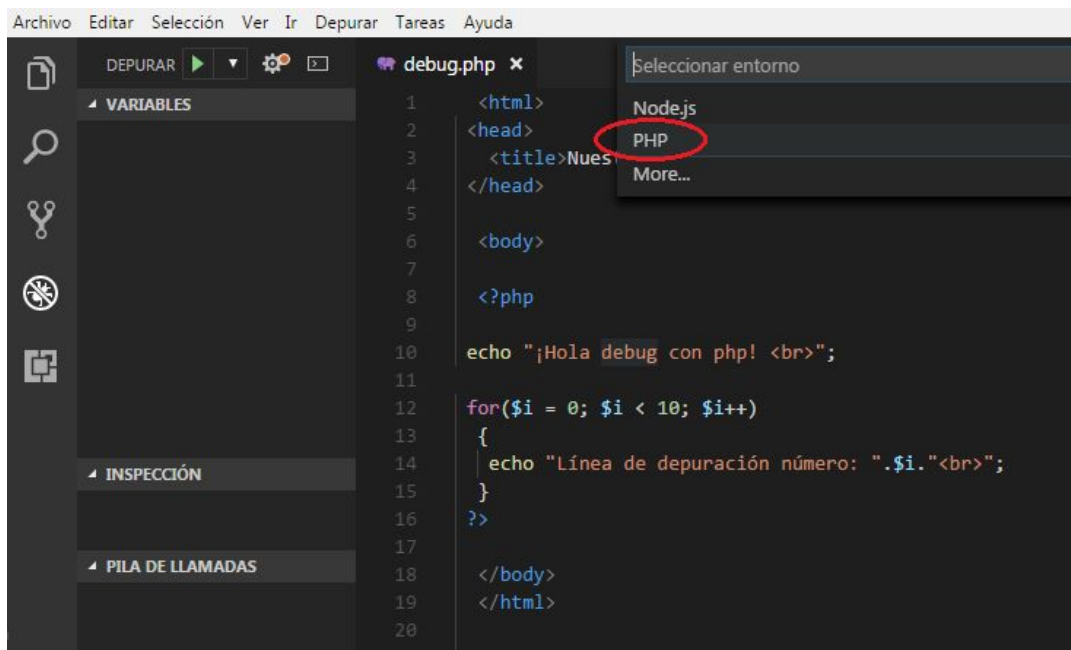
Una vez instalado, para asegurarnos de que funciona, pulsaremos el botón azul Recargar. Y si nos aparece, también pulsaremos el botón con el mensaje «Volver a cargar Window».



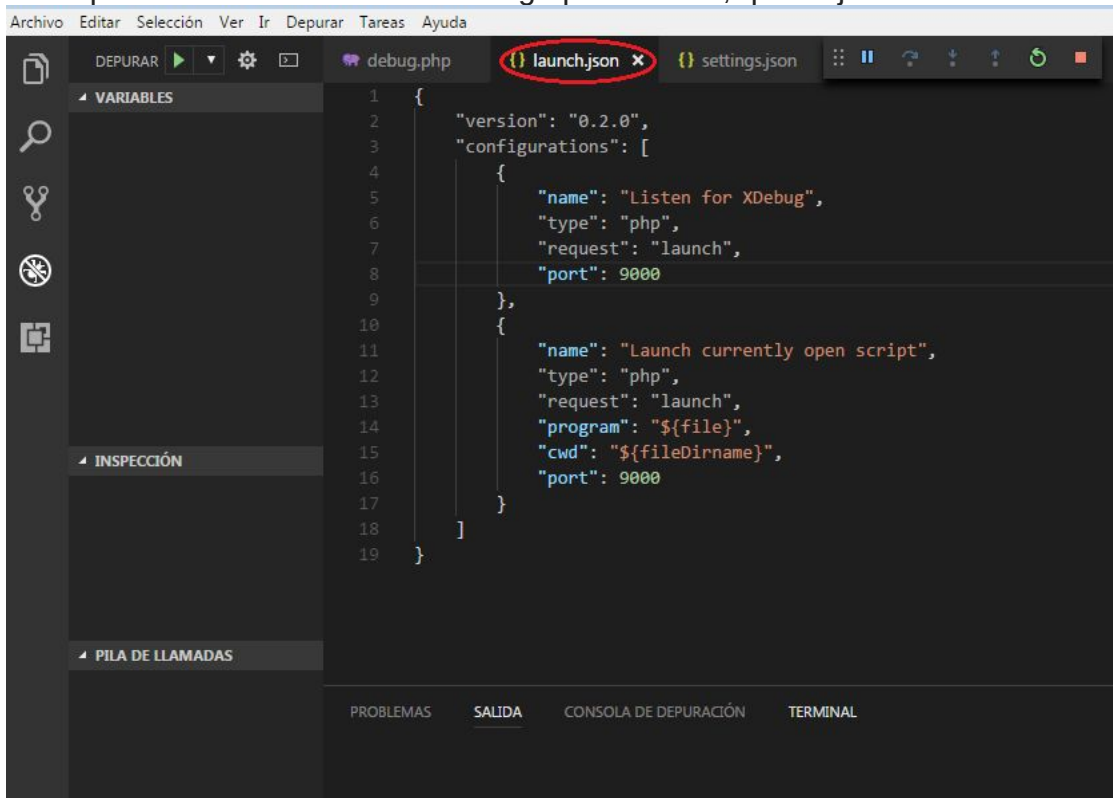
En este momento, es cuando abriremos nuestro proyecto (Archivo > Abrir Carpeta). Y después, accederemos a la opción de depuración en el cuarto icono del menú de la izquierda (1). Lo primero que haremos es agregar una configuración para depurar PHP con Xdebug. En la parte superior hay un selector (2), y a su lado, un engranaje. Al pulsar en el selector nos pedirá que indiquemos el lenguaje a depurar (3).



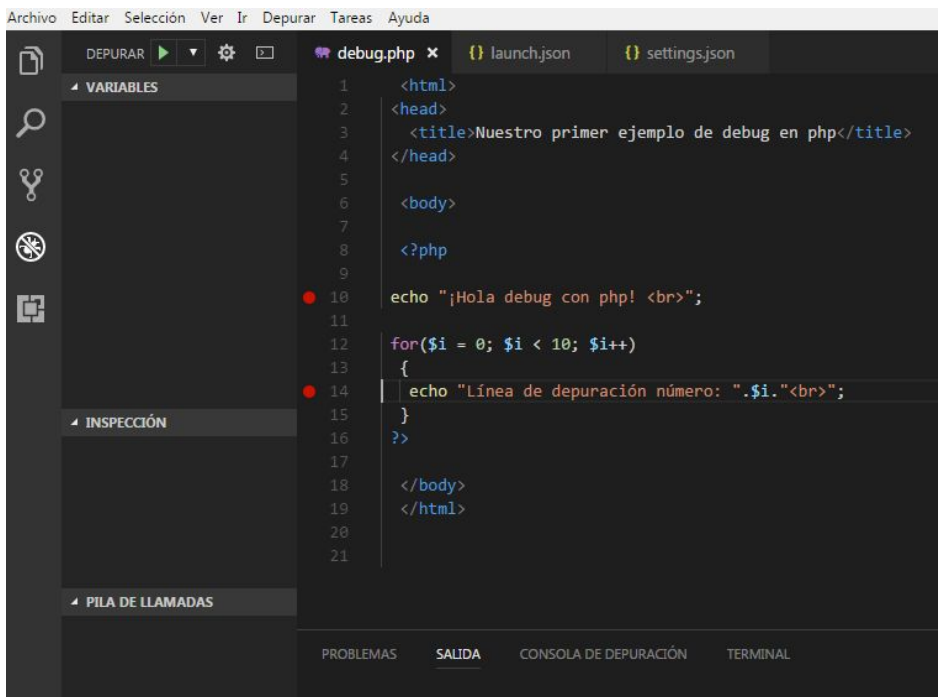
Seleccionaremos el PHP.



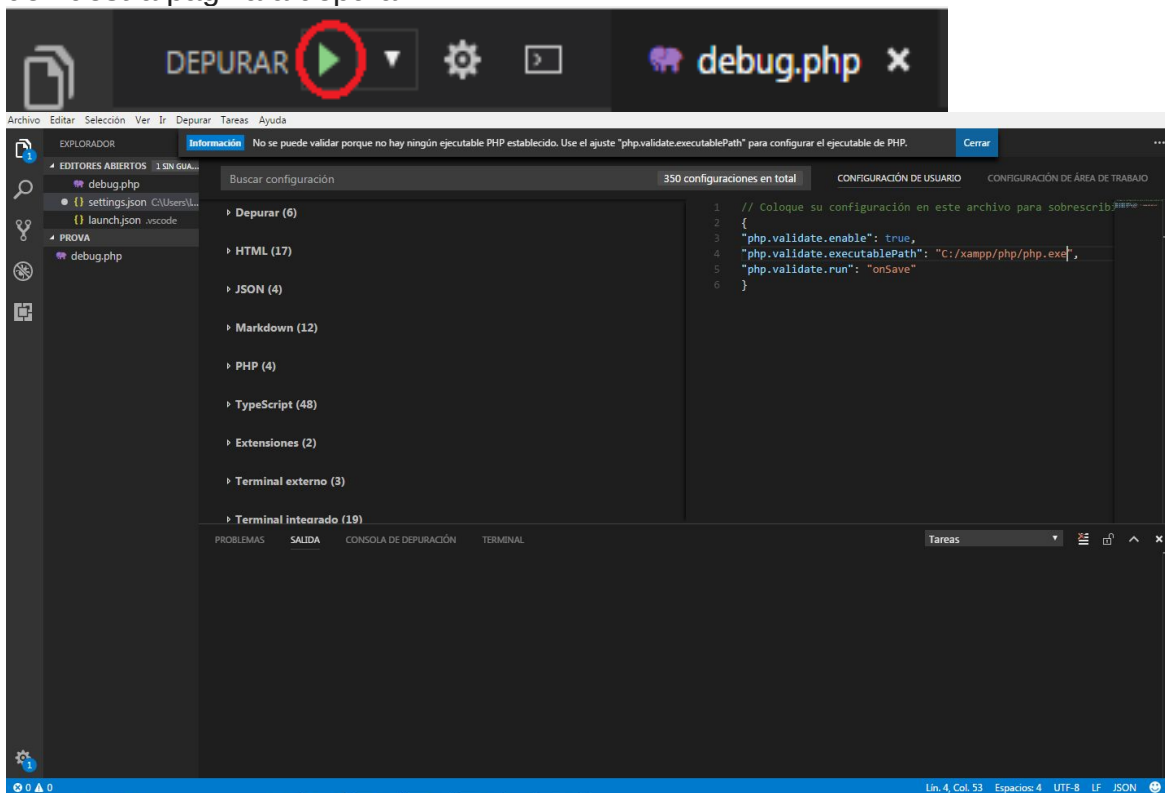
Nos aparecerán unas líneas de código per defecto, que dejaremos tal cual.



En nuestro código PHP, agregaremos puntos de interrupción haciendo click a la izquierda del número de línea. Veremos cómo aparecen puntitos rojos en cada línea con un punto de interrupción.



Pulsamos el botón de Iniciar depuración, y después abrimos el navegador con la dirección de nuestra página a depurar.



Cuando el navegador empiece a cargar la página, Visual Studio Code tomará el control y nos permitirá ir depurando paso a paso usando las típicas herramientas de depuración. Al añadir la configuración para la depuración, Visual Studio Code habrá creado una carpeta `.vscode` dentro de la carpeta de nuestro proyecto con la configuración de para la depuración. A partir de entonces, cada vez que abramos nuestro proyecto, tan solo tendremos que ir a la opción de depuración, poner algún punto de interrupción y darle a la

flechita verde. Si por error borramos esa carpeta, tendremos que volver a repetir los pasos para volver a añadir la configuración.