

Configurar Visual Studio Code para PHP

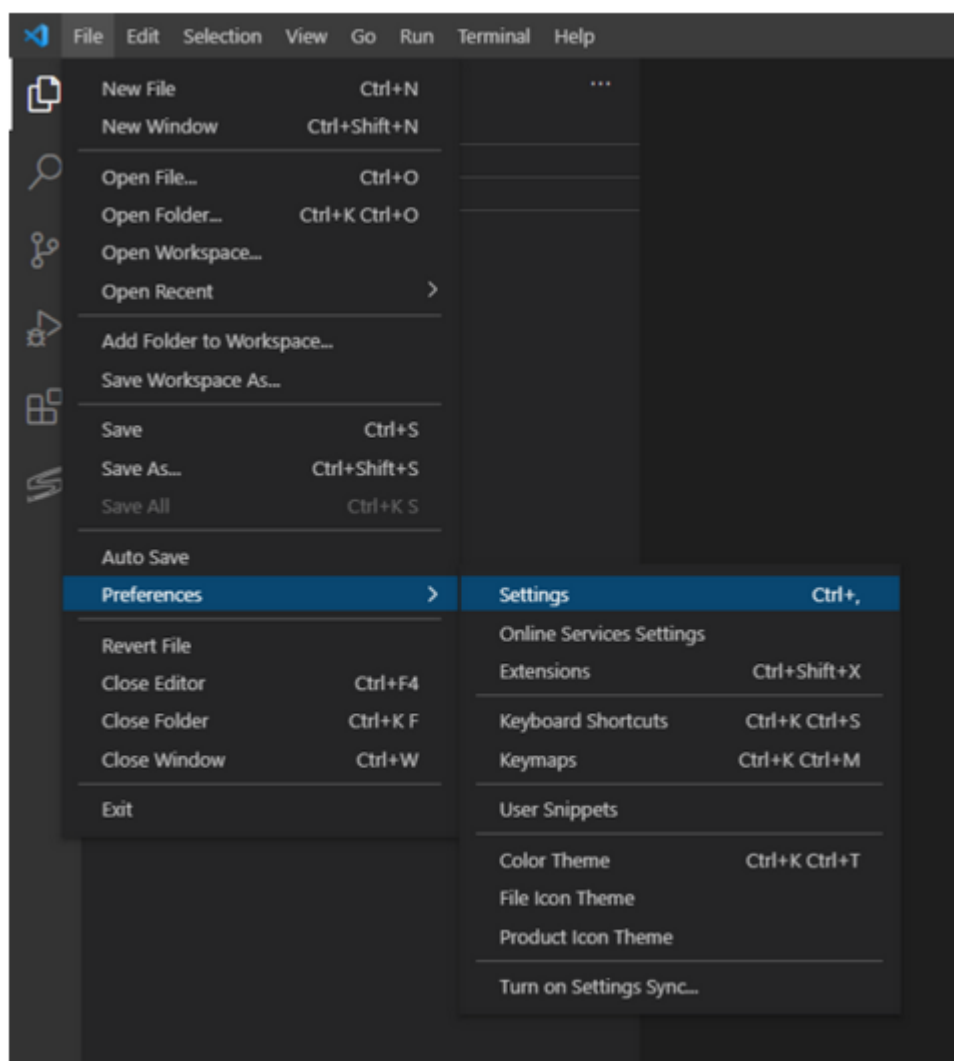
Índice

- [Configuración](#)
- [Extensiones PHP en Visual Studio Code](#)
- [Mi primer Programa en PHP con Visual Studio Code](#)
- [Utilizar un servidor local de PHP](#)

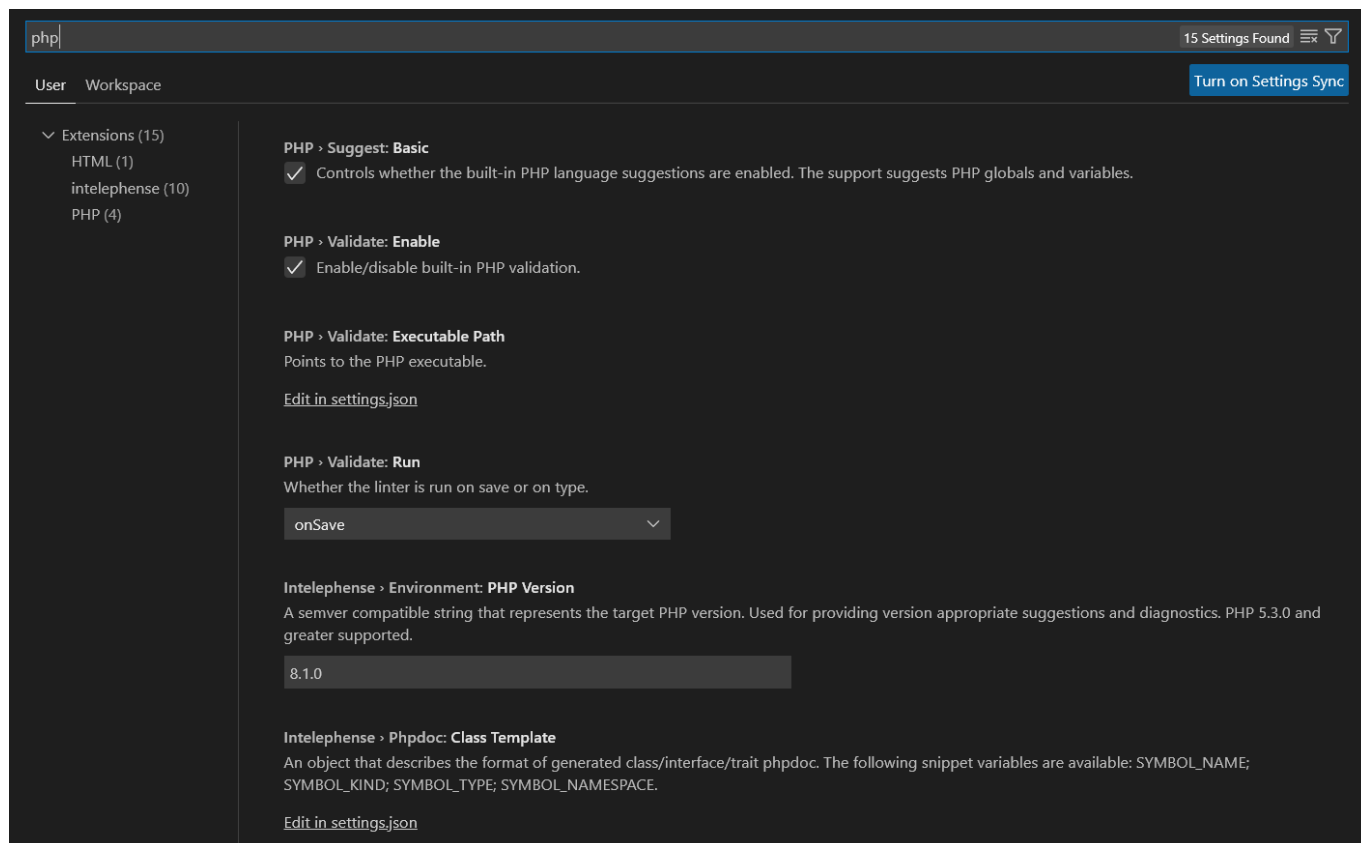
Configuración

Visual Studio Code utiliza el propio motor de PHP para interpretar la sintaxis del lenguaje de programación. Para permitir que nuestro Visual Studio Code nos proporcione toda la potencia de PHP, **lo primero que tenemos que hacer es modificar los Ajustes para decirle a VS Code dónde está instalado PHP**.

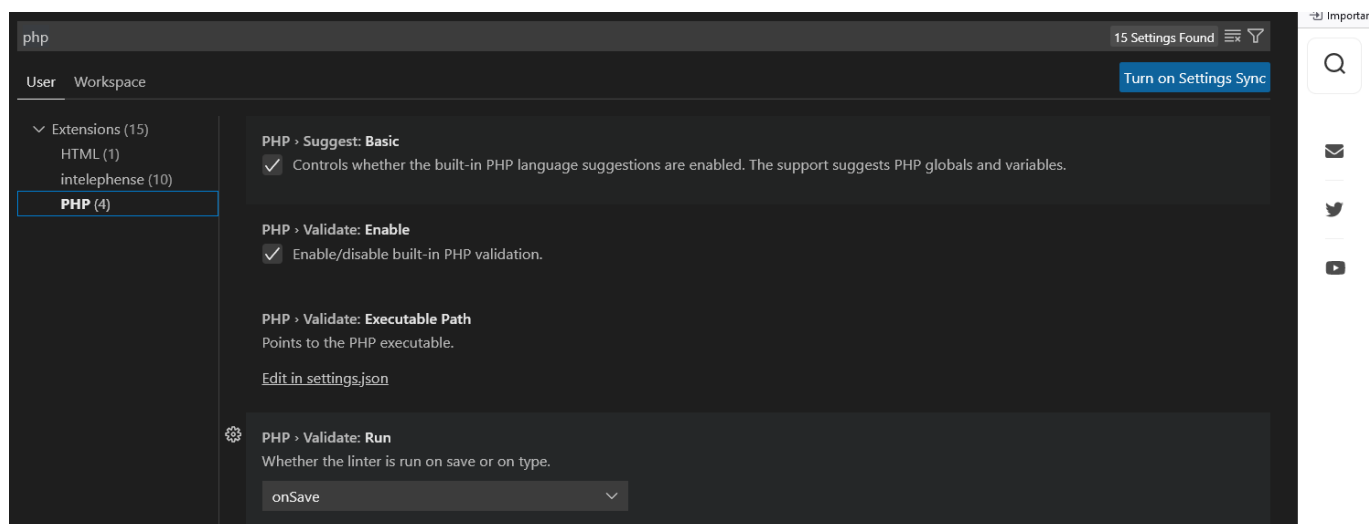
Para ello, una vez abierto Visual Studio Code, vamos a **Archivo -> Preferencias -> Ajustes (File -> Preferences -> Settings)**, o utilizamos el atajo de teclado «**Control+,**». Esto nos abrirá el panel de Ajustes que debemos modificar a continuación.



Escribimos «php» en la barra superior para filtrar los ajustes y sólo mostrar los que necesitamos:



En este panel de ajustes, debemos asegurarnos que las opciones de «Enable» y «Run» estén marcadas. Tras esto, **vamos a modificar el fichero settings.json** para incluir la ruta al ejecutable de PHP que tenemos instalado en nuestro ordenador.



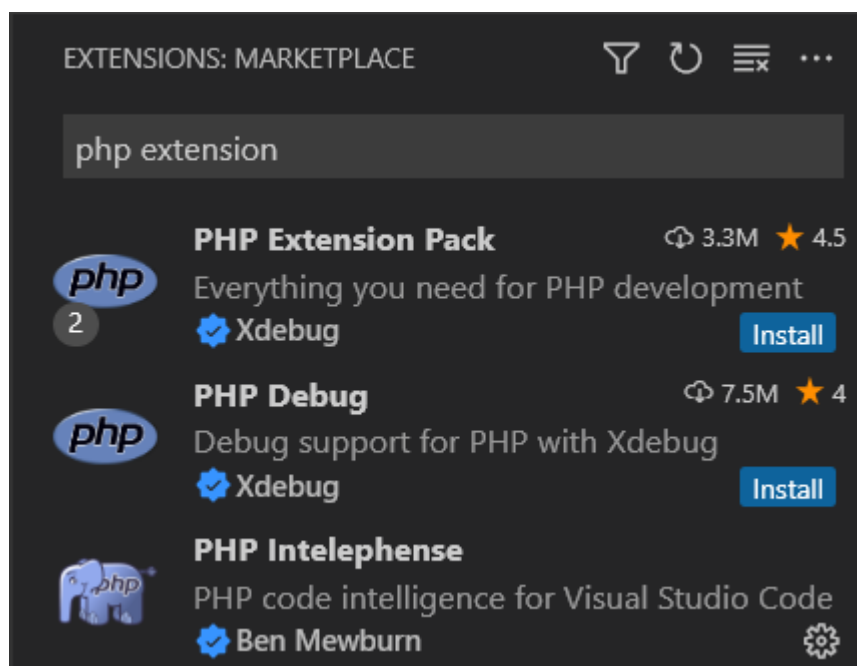
Nota: Si no tienes PHP instalado, necesitarás instalarlo previamente.

```
{
  "php.validate.executablePath": "c:\\xampp\\php\\php.exe",
  "explorer.confirmDelete": false,
  "explorer.confirmDragAndDrop": false
}
```

La ruta dependerá del Sistema Operativo que estés usando (Windows, Linux o Mac).

Extensiones PHP en Visual Studio Code

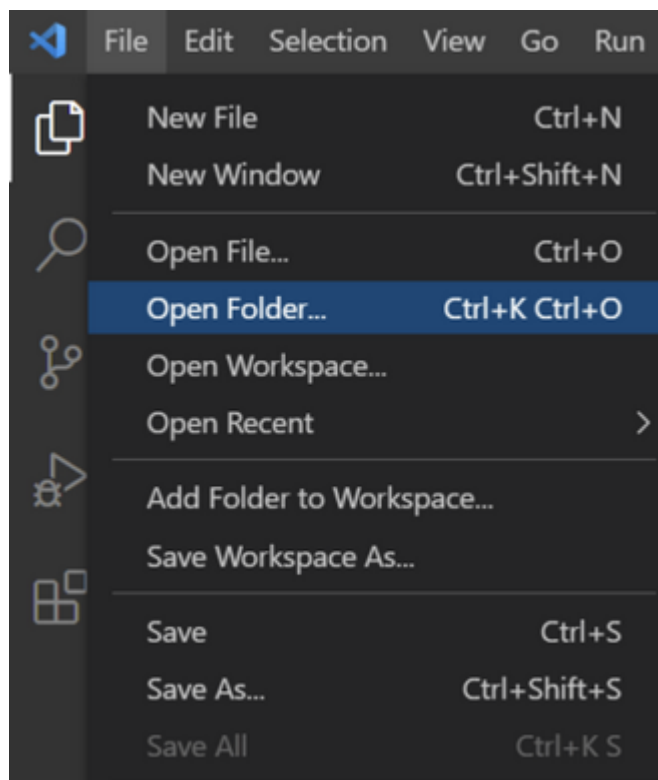
PHP Extension Pack: contiene PHP Debug y PHP Intelephense



Para comprender bien cómo funciona la depuración en VS Code os recomendamos [guía de debug](#)

Mi primer Programa en PHP con Visual Studio Code

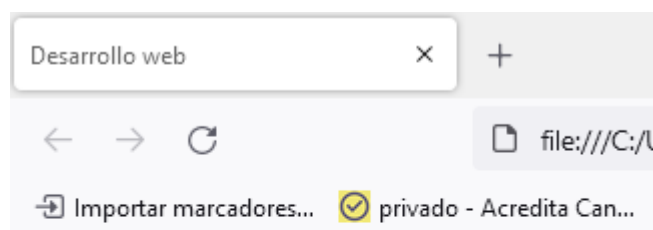
Lo primero que tenemos que hacer, como siempre que utilizamos VS Code, es abrir **una Carpeta o Espacio de trabajo (Workspace)** para comenzar a trabajar.



en mi caso, creo una **nueva carpeta EspacioVSC (de momento vacía)** y la selecciono. A continuación, creamos un nuevo fichero llamado **index.php** y que será el punto de entrada de nuestra web en PHP.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Desarrollo web</title>
  </head>
  <body>
    <h1>Desarrollo en servidor</h1>
  </body>
</html>
```

Para verlo en el navegador, podemos hacer botón derecho sobre el fichero index.php y abrirlo con Firefox. De momento esto nos servirá, ya que aún no hemos añadido código PHP



Desarrollo en servidor

Finalmente, vamos a probar a introducir código PHP en nuestra web, para comprobar que toda la configuración de Visual Studio Code es correcta. Por ejemplo, podemos probar a mostrar en nuestra página web información sobre la instalación de PHP (versión, etc).

```
<?php
    phpinfo();
?>
```

el código PHP se introduce entre las etiquetas `<?php` y `?>`, ya que es la sintaxis típica de PHP. Mientras escribimos dentro de estas etiquetas, tenemos ayuda por parte de la extensión de IntelliSense que hemos activado previamente, de forma que no tenemos que aprendernos de memoria todas las funciones disponibles en PHP.

Utilizar un servidor local de PHP

Tras introducir código PHP en el fichero index, te habrás dado cuenta de que no se visualiza en nuestro navegador. Esto es porque **el código PHP necesita un servidor para poder ser ejecutado**, así que ya no vale simplemente con abrir el fichero index.php con nuestro navegador.

Un workaround rápido es abrir con un **Explorador de Archivos** la carpeta donde está nuestro index.php, y levantar una **Terminal/Consola/Powershell**. A continuación escribimos el comando:

```
php -S localhost:8001
```

Curso22-23 > DESARROLLOWEBENTORNOSERVIDOR > EspacioVSC

Nombre	Fecha de modificación
index.php	16/09/2022 12:15

Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/pscore6>



```
PS C:\Users\cic> cd C:\Users\cic\Documents\Curso22-23\DESARROLLOWEBENTORNOSERVIDOR\EspacioVSC
PS C:\Users\cic\Documents\Curso22-23\DESARROLLOWEBENTORNOSERVIDOR\EspacioVSC>
PS C:\Users\cic\Documents\Curso22-23\DESARROLLOWEBENTORNOSERVIDOR\EspacioVSC> php -S localhost:8001
[Fri Sep 16 12:21:05 2022] PHP 8.1.6 Development Server (http://localhost:8001) started
```

Y en ese momento ya tendremos **un servidor local** que es capaz de **ejecutar nuestra web PHP** en la dirección `http://localhost:8001` (en vez de 8001, puedes elegir el puerto que quieras que no esté ocupado).

Desarrollo web x +

← → ↻ localhost:8001

Desarrollo en servidor

PHP Version 8.1.6 	
System	Windows NT LAPTOP-OC2CV56B 10.0 build 19044 (Windows 10) AMD64
Build Date	May 11 2022 08:52:54
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cmdscriptnologo lejscript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=\\.\.\.\instantclient\shared" "--with-oci8-19=\\.\.\.\instantclient\shared" "--enable-object-out-dir=.obj" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Built-in HTTP server
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902,TS,VS16
PHP Extension Build	API20210902,TS,VS16
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
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, tlsv1.3
Registered Stream Filters	convert.iconv*, string.rot13, string.toupper, string.tolower, convert*, consumed, dechunk, zlib.*, bzip2.*
This program makes use of the Zend Scripting Language Engine: Zend Engine v4.1.6, Copyright (c) Zend Technologies	
	

También podemos realizar la acción de utilizar un servidor local de PHP desde el propio visual studio code abriendo una terminal

