**MANUAL servir web PHP en Visual Studio CODE usando XAMPP**

Todo esto no es para conectar XAMPP con VSC, es para tener un servidor web de php en vsc gracias a la extensión PHP server. Todo esto que sigue funciona gracias a PHP server, y puedo parar XAMPP y sigue funcionando, de XAMP lo único que usa es el php.exe porque la extensión PHP server no lleva php.exe.

VIDEO 1: <https://www.youtube.com/watch?v=W9cDh_sC2As>

VIDEO 2: https://www.youtube.com/watch?v=jHJGOAZG0oc

Básicamente es configurar el setting.jason con la línea "php.validate.executablePath": "C:\\xampp\\php\\php.exe"

Luego instala 3 extensiones de php (una de ellas PHP server) para VSC y otras 3 de mysql pero creo que las de mysql no hacen falta.

Luego crear una variable de entorno en Windows con la ruta c:\xampp\php.

Ya estaría conectado y en un proyecto php dentro de VSC con botón derecho se le da a php server server proyect y se abre en el navegador web la página creada en php.

Para volver a lanzar hay que pararlo antes dándole a PHP STOP SERVER.

OJO: PHP server no lleva debug.

**RESUMEN DE LO QUE HACE ESTO**: al instalar Xampp (X) se crea en el ordenador, en local, PHP.exe, con lo que ya tenemos PHP en ejecutable. Solo nos falta un servidor para ejecutar un script php con el php.exe. Ese servidor es una extensión que se instala PHP Server. Al instalarla en los setting de VSC en la zona de Extensions/php server configuración aparecen nuevos parámetros como el puerto (3000), el path del php.exe. y el path de php.ini. Con esto, lo que se hace es instalar un servidor web embebido en el propio php, no es un servidor web real. La extensión de VSC PHP server, coge el script y lo ejecuta en el puerto localhost/3000 (servidor web). No hace falta que X esté arrancado ni apache porque el servidor lo crea PHP server con el php.exe. Pero ese PHP server usa el php.exe que tiene X en su carpeta de instalación, que es para lo único que se necesita Xampp.

Pero VSC usa la folder que tiene abierta para poder mapear el scritp con el PHP server, es decir, si el folder de VSC no es el del script que se quiere ejecutar en el PHP server, no va a funcionar porque no se ven, PHP server no encuentra el script para ejecutarlo y da error de no encontrado. Los servidores siempre usan una carpera del pc local para poder comunicarse con el pc que es la carpeta que tiene los script, esa carpeta en este caso es Folder de VSC.

Uno de los parámetros que hay en settings de VSC en la parte de extensión/PHP server es Relative Path: con este parámetro se le indica una ruta local para mapear con el servidor PHP server, indicando una ruta, haríamos que php server cargue script que no esté en el Folder actual. El servidor que crea PHP server ve los archivos de la carpeta donde el propio servidor está corriendo. Si hago un phpinfo() del servidor que crea me sale en algunas variables que usa php que se está ejecutando en la ruta folder de VSC.

**MUY IMPORTANTE: para ejecutar ficheros php en un servidor de PHP server de VSC primero debe estar abierta la folder del fichero a ejecutar en VSC. Ir File, close folder actual y open folder donde está el fichero .php. y así ya se puede usar lo del PHP SERVER:** server proyect. También funciona si el script esta en una carpeta que es subcarpeta de folder. El servidor web ve Folder y todas sus subcarpetas.

INTERESANTE: Tenia el contenedor de PHP en marcha que escucha en el localhosto:8000. En VSC pongo como folder una carpeta www/mis\_pruebas, y el servidor de PHP del contenedor tiene mapeada la carpeta www, con lo que todo lo que se ponga en ella o dentro de subcarpetas lo carga en el contenedor y lo ve el servidor PHP. Pues no me funciona el PHP server de VSC, en localhost:3000/fichero.php error página no encontrada, y es porque hay otro servidor php en marcha en el puerto 8000. Hay conflicto, aunque no debería porque son puertos diferentes y ejecutables diferentes, en el puerto 3000 se está usando el php.exe de XAMPP, en el 8000 el del contenedor. Paro el contenedor de PHP y ya funciona.

Como información cabe decir que teniendo un php.exe es muy sencillo arrancar un servidor web de php (modo línea comandos cli-server), en una ventana de ms-dos, voy a una carpeta que tenga php.exe y escribo **php -S localhost:8000** esto crea un servidor de php en el puerto 8000. Si creo o copio un script en la carpeta donde he creado el servidor ya lo puedo ejecutar en localhosto:8000/nombre.php. (Si en la variable de entorno de Windows Path le pongo la línea c:\xampp\php, como le estoy indicando donde hay un php.exe bastaría con escribir **php -S localhost:8000** en cualquier carpeta de Windows).

Ahora bien para ejecutar script usando el servidor de Xampp hay que copiarlos en la carpeta C:\xampp\htdocs y escribir en el navegador web <http://localhost/script.php> ya que apache de Xampp escucha por el puerto 80 y esta mapeado a la carpeta C:\xampp\htdocs, no tengo que escribir <http://localhost/htdocs>, porque al estar mapeada la carpeta htdocs internamente en Xampp, él ve el contenido de esa carpeta dentro de su servidor web y lo muestra por el puerto 80. Si tuviera mapeada la carpeta c:/pedrito, todo lo que yo metiera en pedrito, archivos HTML o script PHP estarían visibles para ejecutar por el puerto 80 de localhost.

**He hecho phpinfo() de** [**http://localhost/dashboard/phpinfo.php**](http://localhost/dashboard/phpinfo.php) **PHP corriendo en servidor apache de XAMPP y phpinfo() de** [**http://localhost:3000/prueba.php**](http://localhost:3000/prueba.php)**, SERVIDOR de VCS extensión PHP server. Puedo ver que XAMPP está en servidor apache y el otro en servidor llamado: Built-in HTTP server.**

En el parámetro Loaded Configuration File: apache tiene “C:\xampp\php\php.ini” y VSC tiene “C:\xampp\php\php.exe”, por eso no está cargando xdebug.

Le pregunto a la IA y me dice esto:

Edita tu archivo phpinfo.php para mostrar el nombre del entorno de ejecución. Usa esto:

<?php

echo "<hr>";

echo "SAPI: " . php\_sapi\_name(); // tipo servidor donde se ejecuta php

echo 'Ruta actual: ' . \_\_FILE\_\_; //ruta donde se ejecuta el script

?>

Esto es lo que me sale: cli-server

Qué debes ver

* Si ves algo como: SAPI: apache2handler O SAPI: cgi-fcgi → ¡Perfecto! PHP se está ejecutando en modo servidor.
* **Pero si ves: SAPI: cli** → ¡Ahí está el problema! Estás ejecutando PHP en modo CLI (línea de comandos), no como un servidor web. Esto sugiere que el entorno en el que se está ejecutando phpinfo() no es un servidor real, o bien está usando PHP de forma incorrecta (como en modo CLI embebido, no como servidor web). La extensión PHP Server no es un servidor web completo como Apache o Nginx. Lo que hace es ejecutar PHP directamente con el servidor embebido de PHP: Y si no está correctamente configurado, puede acabar simplemente llamando a php en modo CLI, que solo ejecuta el archivo y lo devuelve como texto plano o sin procesar bien. Con php -S localhost:3000 se inicia php en modo cli.

**La IA dice:** El valor cli-server indica que estás ejecutando PHP con su **servidor embebido** (integrado en PHP desde la versión 5.4). Esto confirma que la extensión **PHP Server** de VS Code está usando el comando: php -S localhost: PORT, Y eso está bien, **funciona correctamente para desarrollo local**.

La extensión **PHP Server** de VS Code no está pasando correctamente el parámetro de configuración que le indica a PHP cuál archivo php.ini debe usar. Este comportamiento confirma que **PHP se está ejecutando**, pero **sin detectar ni usar el archivo php.ini**, porque la extensión lo invoca de forma incompleta o incorrecta.

**Interesante prueba. La IA me dice que haga esto:**

1. **Abre una terminal en la carpeta donde tienes tu prueba.php script que ejecutara phinfo()**

Ejecuta: **php -S localhost:8000** esto crea un servidor de php en el puerto 8000.

**Abre en el navegador:** [**http://localhost:8000/phpinfo.php**](http://localhost:8000/phpinfo.php)

**Ahora deberías ver:**

* **Toda la tabla de phpinfo()**
* **Y Loaded Configuration File correctamente apuntando a php.ini, como:**

**Veo que el servidor que ha creado con el CMD** Es de tipo Built-in HTTP server igual que el de PHP server. Así que el problema no es php server si no la configuración de algún parámetro de VSC o PHP server.

|  |  |
| --- | --- |
| Finalmente doy con el problema: en settings de VSC, en el apartado de extensions/PHP Server configuration hay un parámetro llamado **PHP Config Path: ahí es donde se indica la ruta de php.ini dándole el valor C:\xampp\php\php.ini. Lo he probado y ya me carga el debug en PHP. La IA no sabía nada de esto y me ha felicitado.**  **Así que sin usar servidor web apache para PHP de XAMPP, ni servidor web de contenedor yoprogramo, me funciona del debug, pero ojo, no debo tener el fichero launch.jason que me dice de crear la 1º vez que se usa el debug en una Folder, no crearlo y darle a Run and Debug, si le creo el launch.jason, PHP debug se configura en modo listen to xdebug, y el debug escucha por el puerto 9003 peticiones y recibe información de la ejecución del script, entonces falla el debug, para que funcione con el launch.jason hay que arrancar XAMPP o del contenedor.**  **Launch.json siempre es creado en el Folder de VSC actual si VSC ve que no hay ninguno.**  **ESTO DICE LA IA: VS Code genera una configuración temporal (implícita) de depuración**  **Cuando haces clic en:**  **🔷 Run and Debug (Ejecutar y depurar)**  **🔷 Y luego eliges el entorno "PHP" por primera vez**  **VS Code crea internamente una configuración por defecto basada en las extensiones instaladas, en este caso la extensión PHP Debug.**  **🔧 Aunque no veas el archivo launch.json en tu carpeta, VS Code está utilizando una configuración predeterminada como si lo tuviera, algo como esto:**  **{**  **"type": "php",**  **"request": "launch",**  **"name": "Listen for Xdebug",**  **"port": 9003**  **}**  **¿Por qué hace eso? Para facilitarte el arranque. VS Code detecta que tienes: La extensión PHP Debug instalada.**  **Un archivo PHP abierto. Y posiblemente Xdebug ya funcionando en tu entorno.**  **Entonces dice: "No te preocupes, yo me encargo" y crea una configuración interna.**  **📂 ¿Qué pasa si luego creas un launch.json? VS Code dejará de usar la configuración interna.**  **Usará la configuración que tú escribas.**  **Esto es útil cuando necesitas más control: rutas específicas, parámetros, entornos personalizados, etc**  **Para usar PHP debug con XAMPP o contenedor debo primero abrir el folder correspondiente a donde ejecuta los script XAMPP (C:\xampp\htdocs o subcarpeta) o el Folder mapeado con el contenedor de PHP (en mi caso para el contendor yoprogramo es C:\Users\psand\curso-de-php-8-y-mysql-8).**  **Usando PHP server para ejecutar un script que está en la carpeta C:\Users\psand\curso-de-php-8-y-mysql-8, y la instrucción de php** *echo 'Ruta actual: '. \_\_FILE\_\_;* **que nos indica la ruta de ejecución del script me imprime C:\Users\psand\curso-de-php-8-y-mysql-8, con lo que el servidor de PHP y mi fichero están en la misma carpeta, eso es porque PHP server arranca un servidor web en la misma carpeta Folder que tengo cargada, por eso para que funcione el depurador y la orden (botón derecho ratón) de PHP server: PHP proyecto (que ejecuta el script) deben estar el script y el folder en la misma carpeta porque PHP server arranca el servidor en la carpeta Folder y este servidor solo ve los archivos que hay en su carpeta (y subcarpetas).**  **Requisitos para depurar un script PHP en VSC (sin launch.json, sin servidor web XAMPP O CONTENEDOR.**   * Que el script este en la misma carpeta que el Folder de VSC (o subcarpeta) * Que el launch.json no este creado.   PORQUE si se crea, aparece la opción listen to xdebug y esa opción solo funciona para servidores web con php tipo buildin o apache o gnix, servidores web reales.  Pero PHP server crea un servidor tipo cli-server embebido en php, y para eso no se puede usar listen to xdebug, así que no se puede crear el launch.jason**.**  **No es necesario clic botón derecho/PHPsever: server proyect para que se vean las variables.**  Requisito para que funcione el botón derecho PHP server: server proyect que el script y el Folder estén en el mismo sitio. |  |
| En VSC hay un botón similar a Run and Debug, esta en VSC en la ventana del script esquina superior derecha hay esto |  |

Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.

Pinchando en la flechita que hay junto al PLAY sale esto y en Debug PHP

Conclusiones: siempre que le doy a run and debug, sea con servidor creado por PHP server, servidor de XAMPP o servidor en contenedor, se esta usando la extencion PHP debug que escucha por el puerto 9003 a la librería xdebug. Asi que cuando se esta depurando con el Run and Debug sin usar launch.json es porque PHP Debug por defecto escucha por el puerto 9003, y además al darle a Run and Debug, VSC crea el servidor embabido de php con el php.exe de la carpeta de XAMPP, esto esta comprobado porque si cambio el nombre a la carpeta deja de funcionar, sin embargo si cambio el puerto 9003 del pho.ini, no deja de funcionar, y tiene que ser que por defecto xdebug.dll si no se le especifica puerto usara el 9003.

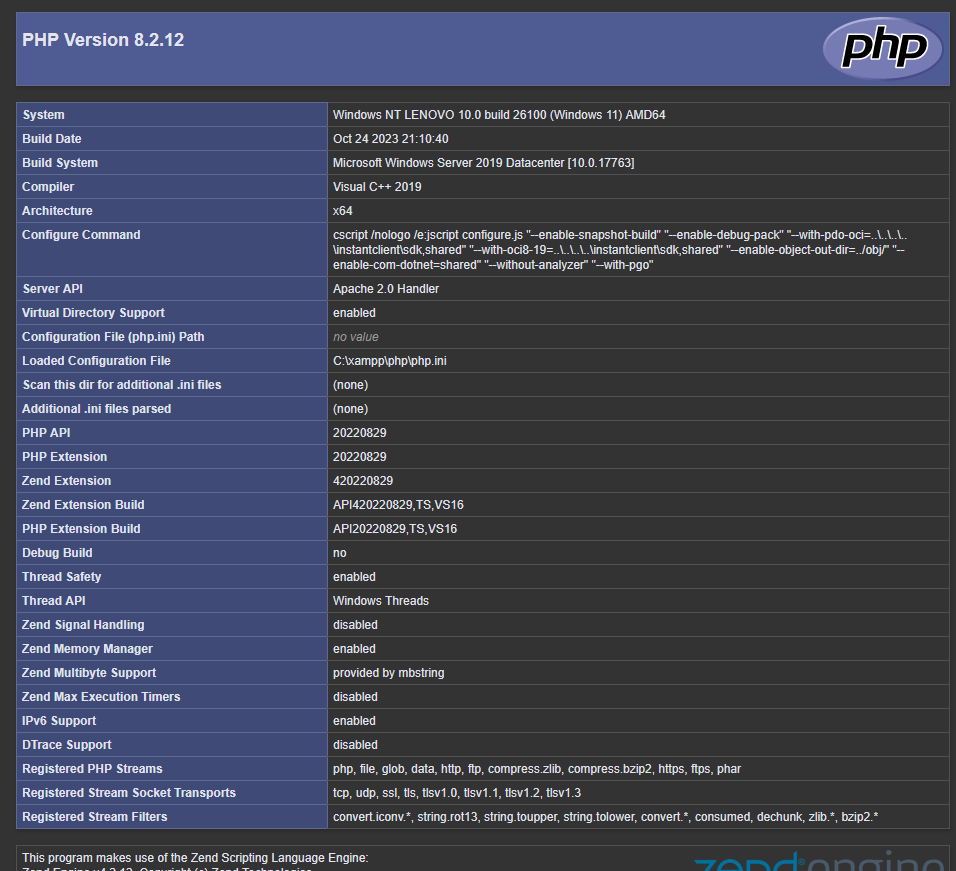
**COMO CONFIGURAR PHP DEBUG EN VSC (para depurar código paso a paso)**

VIDEO 2: https://www.youtube.com/watch?v=uwPobL2jCnw

VIDEO 1: <https://www.youtube.com/watch?v=QxRTw4GPk5Q>

**Aclaración: en estos videos PHP.exe está en una carpeta local y se ejecuta en local, aunque luego hay servidor web que escucha un puerto y si le copias un script.php en su carpeta de trabajo y te vas al navegador y pones localhost:puerto/nombre.php lo va a ejecutar, pero todo está en local. Con lo que la extensión PHPdebug de VSC se comunica con PHP.exe en local, así no se necesita mapping de rutas como sucede cuando PHP.exe se ejecuta en un contenedor. El contenedor no tiene rutas locales, sus rutas están en otra máquina. En XaMPP sí que están en el mismo pc, XAMP es un servidor web pero ejecutado con un php.exe que está en local y usa carpetas locales.**

SE INSTALA PHP DEBUG EN LAS EXTENSIONES DE VSC

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.EN LAS INSTRUCCIONES DE PHP DEBUG (PD) HAY UN LINK PARA USAR UN **ASISTENTE** QUE DICE QUE TIENES QUE HACER, PINCHAR EN EL LINK Y SALE UNA VENTANA COMO ESTA

PHP INFO

ASISTENTE

EN ESA VENTANA PEGAR EL TEXTO (CTRL+A) QUE TE DA LA FUNCION **PHPINFO()** EJECUTADA EN UN FICHERO PHP EN EL SERVIDOR DE PHP (CARPETA CONECTADA CON XAMPP) O ESTO EN EL NAVEGADOR <http://localhost/dashboard/phpinfo.php>

**EN LAS INSTRUCCIONES TE DICE QUE DESCARGUES UNA DLL, HAY QUE DESCARGAR ESA EXACTAMENTE, YO H ESTADO 3 DIAS CON FALLOS Y ES PORQUE DESCARGUE OTRA.**

SE DESCARGA, SE RENOMBRA, SE COPIA EN EL DIRECTORIO QUE DICE Y LUEGO EDITAR PHP.INI ESCRIBIENDO ESTO:

zend\_extension = xdebug

xdebug.mode = debug

xdebug.start\_with\_request = yes

xdebug.remote\_port = 9003

xdebug.idekey = VSCODE

SE GUARDA SE REINICIA XAMPP Y LUEGO VAMOS A ESTE BOTON, EL QUE ARRANCA LA DEPURACION (php DEBUGER) Y LA PRIMERA VEZ SALE QUE SE PINCHE EN CREAR ARCHIVO LAUNCH. JASON, DECIRLE QUE ES PARA ENTORNO PHP. EN LAUNCH.JASON

PONER EL PUESTO 9003 MISMO QUE EL PHP.INI

Una captura de pantalla de un celular con texto e imagen

El contenido generado por IA puede ser incorrecto.A LA HORA DE ARRANCAR EL DEBUG HAY QUE ELEGIR ESTA OPCION LISTEN FOR XDEBUG

SI SE BORRAR ESTE FICHERO LAUNCH.JASON OTRA VEX AL ARRANCAR PD TE PIDE

CREAR EL ARCHIVO LAUNCH.JASON

**HAY QUE TENER EN CUENTA QUE PARA QUE SE VEAN LOS VALORES DE LAS VARIABLES EN EL DEGUG HAY QUE HACER PHP SERVER PROYEC CON EL BOTON DERECHO EN EL SCRITCH PHP CREADO (ojo teniendo script y folder en la misma carpeta), PARA CARGAR el script en** [**http://localhost:3000/prueba.php**](http://localhost:3000/prueba.php) **que es el servidor que crea PHPserver o poner en el navegador** [**http://localhost/prueba.php**](http://localhost/prueba.php) **que hace lo mismo que lo anterior pero en el servidor de XAMPP. Al final para que se ven las variables usando PHPdebug hay que ejecutar el script en algún servidor web de php que tenga xdebug, y como php.ini de ese php.exe tiene puesto que se comunique con el puerto 9003, pues envía a PHPdebug valores de variables. El php.ini es el mismo en XAMP y en PHPserver (La extensión PHP server se ha configurado para que ejecute el php.exe de xampp)**

**Así que para usar PHP debug con XAMPP o contenedor debo primero abrir el folder correspondiente a donde ejecuta los script XAMPP (C:\xampp\htdocs o subcarpeta) o el Folder mapeado con el contenedor de PHP (en mi caso para el contendor yoprogramo es C:\Users\psand\curso-de-php-8-y-mysql-8).**

**XAMPP**

XAMP es una multiherramienta, que lleva servidor web APACHE para PHP, porque PHP por sí solo no tiene servidor web real. Lleva mysql y más cosas.

XAMP (X) simula un servidor web en la 127.0.0.1 o localhost. Al escribir localhost (apache escucha el puerto 80. Para ejecutar script de php copiados en la carpeta C:\xampp\htdocs. tengo que poner localhost:/ htdocs /nombre script. Si pongo solo C:\xampp\htdocs en navegador web sale la página de bienvenida de X, que es el fichero index.php que hay en la carpeta C:\xampp\htdocs.

La carpeta C:\xampp\htdocs es la carpeta que el servidor web usa para contener sus páginas web creadas en PHP. Solo se puede ejecutar scritch php que este en esa carpeta (escribiendo localhost/nombre script ) o en una subcarpeta que este dentro de C:\xampp\htdocs.

Nota: en la página de bienvenida de X hay un link arriba que pone phpinfo que es lo mismo que ejecutar la función de php phpinfo() para ver toda la info de PHP (antes hay que arrancar Apache en el panel de control de X. también hay otro link PHPmyadmin que va a la base de datos de Mysql, habiendo arrancado antes Mysql en el panel de control de X.

**Depurando PHP con Xdebug EN VSC y Docker (PHP CORRIENDO EN CONTENEDOR)**

**Para usar PHP debug con XAMPP o contenedor debo primero abrir el folder correspondiente a donde ejecuta los script XAMPP (C:\xampp\htdocs o subcarpeta) o el Folder mapeado con el contenedor de PHP (en mi caso para el contendor yoprogramo es C:\Users\psand\curso-de-php-8-y-mysql-8).**

LINK DE INSTRUCCIONES: <https://yoprogramo.com/2024/10/12/depurando-php-con-xdebug-y-docker/>

Crear la carpeta web, y dentro crear el archivo Docker docker-compose.yml.

version: '3'

services:

php:

image: yoprogramo/php8.2:1.0.2-xdebug

volumes:

- .:/var/www/Website

ports:

- 8080:80

extra\_hosts:

- host.docker.internal:host-gateway

Ojo con la última instrucción que hay que quitar las dobles comillas. Y en volúmenes quitar web, porque ya hemos creado la carpeta y estamos dentro ya, con lo que no tiene que crear nada. Si le ponemos .web creara la carpeta web dentro de web

Ejecutar ahí dentro Docker-compose ps -d para descargar la imagen.

Ya podemos ejecutar localhosto:8080/prueba.php para que ejecute phpinfo() y se puede ver que este contenedor tiene php con la librería xdebug .

También podemos ver que en esta imagen no crea un contenedor con apache ni ningún servidor web, es porque ya lo lleva integrado la imagen, es una imagen con un servidor web y php y los dos configurados para que se vean.

Luego ir a VSC y añadir la línea a launch.json en la parte de Listen for Xdebug

"pathMappings": {

   "/var/www/Website": "${workspaceRoot}/web"}

Para que se vean las variables hay que ejecutar el script en en el servidor del contenedor localhosto:8080/prueba.php

Ojo si launch.json tiene algún error de semántica no lo VSC lo ignora y su configuración por defecto para PHP Debug. Y tampoco se usara el mapping, con lo que aunque ejecute el script en el contenedor no vere variables.

**El depurador de PHP consiste en un servidor web con PHP con librería Xdebug y ejecutando el script, con lo que hay que mapear una ruta del pc local que tiene el script con una ruta del servidor web de PHP para que se pueda ejecutar el script en dicho servidor web, esto solo si php.exe está en un contendor, porque si está en un servidor tipo XAMPP no hay que mapear, porque el script se copia en la carpeta local del XAMP para que lo carge en su servidor web y así ya se ven el script y php.exe. El servidor web de PHP ejecutando el script, como tiene instalado xdebug.dll y en su php.ini se le ha dicho que tiene que depurar por el puerto 9003, envía la información de las variables al puerto 9003, por eso hay que ejecutar el script para que salgan las variables en VSC, envía información del estado de las variables a un servidor, que sería la extensión PHPdebug de VSC, se las envía al puerto 9003. Le envía por cada instrucción el valor de las variables, y PHPdebug de VSC solo tiene que leer eso que recibe por el 9003 y mostrarlo en VSC de manera ordenada pada vez que se ejecuta una instrucción.**

Requisitos para depurar con php en un contenedor:  "pathMappings": {

             "/var/www/Website": "${workspaceRoot}/web"}

 Esta configuración tiene que estar en launch.json, que es el mapeo de unidades del contenedor y mi pc.

OJO: El contenedor de php tiene vinculada su parpeta de Linux **"/var/www/Website** con la carpeta de mi ordenador **workspaceRoot}/web**. **workspaceRoot** significa el espacio de trabajo de VSC, es decir, la Folder que tenga abierta. Entonces si le estoy diciendo que vincule la carpeta del contenedor con mi folder/web, quiere decir que mi Folder de VSC tiene que ser la carpeta que contiene a /web, porque si le pongo como Folder web, entonces intenta mapear mi carpeta/web(folder)/web y esa ruta no existe.

Otro requisito, como siempre sucede con PHP debug se comunique con quien se comunique, hay que ejecutar el script en el servidor web del contenedor poniendo esto en el navegador <http://localhost:8080/prueba.php>. he probado ejecutarlo con el servidor de PHPserver de VSC con el botón derecho y aunque lo ejecuta pero no carga variables en este paso, el PHP.exe del contenedor no es local y no es el mismo PHP que ejecuta PHP server que ejecuta el php.exe de XAMPP.

**NOTA: según donde se ponga el Breakpoint puede ser que no se inicie el debug, porque hay líneas de código HTML que no son compatibles con debug.**