

**MATERIAL UTILIZABLE:**

para la parte TEST, ningún material

para la parte práctica:

- todo el material de este módulo profesional y consulta a internet
  - la parte práctica se realizará sobre el entorno de desarrollo Visual Studio Code, disponible en una máquina virtual (se podrá elegir entre entorno en Windows o entorno en GNU/Linux Ubuntu).
- durante todo el tiempo del examen se grabará la pantalla con el software OBS previamente instalado.

**PARTE 1: TEST****(2 puntos)**

**Realizar el test en la plataforma FPADISTANCIA:** 20 preguntas

Cada pregunta respondida correctamente en cada tema puntúa: 0,1 (= 2 puntos / 20 preguntas)

Cada pregunta respondida incorrectamente en cada tema puntúa: **-0,05** (= (-1/2) \* 2 puntos / 20 preguntas)

La nota que muestra la plataforma al terminar el test NO es la correcta, pues esta plataforma no aplica los descuentos por fallos. La fórmula aplicar, para cada tema, es: nº de aciertos \* 0,10 - nº fallos \* (1/2) \* 0,10

**PARTE 2: PREGUNTAS PRÁCTICAS****(8 puntos)**

**Al terminar sube por favor este fichero .odt con tus respuestas a FPadistancia.**

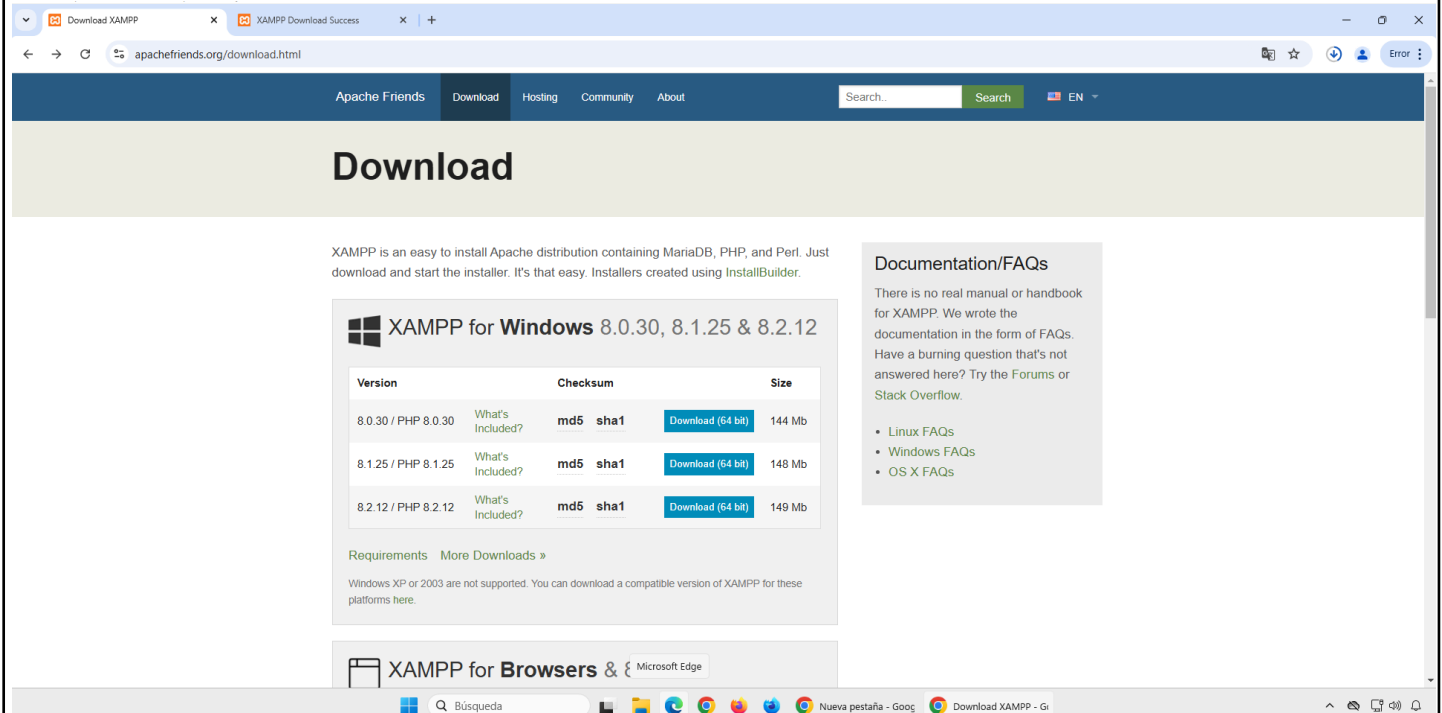
**PREGUNTA 2.1.** Supón que tienes que configurar tu entorno de desarrollo (XAMPP) bajo Windows y Visual Studio Code para desarrollar una aplicación EXAMEN1 y quieres tener todo el código disponible en la carpeta C:\TAREA1\EXAMEN1 y hacer pruebas de su funcionamiento en la URL <http://examen1.localhost>.

Enumera los pasos principales que seguirías y pon capturas de pantalla de los elementos de configuración esenciales (no hace falta documentar todo el proceso detallando los subpasos de cada paso principal).

Al final, pon una captura de pantalla de un navegador web abriendo cualquier archivo que crees de esa aplicación en la URL indicada, para demostrar que funciona tu configuración.

**RESPUESTA:** (4 puntos)

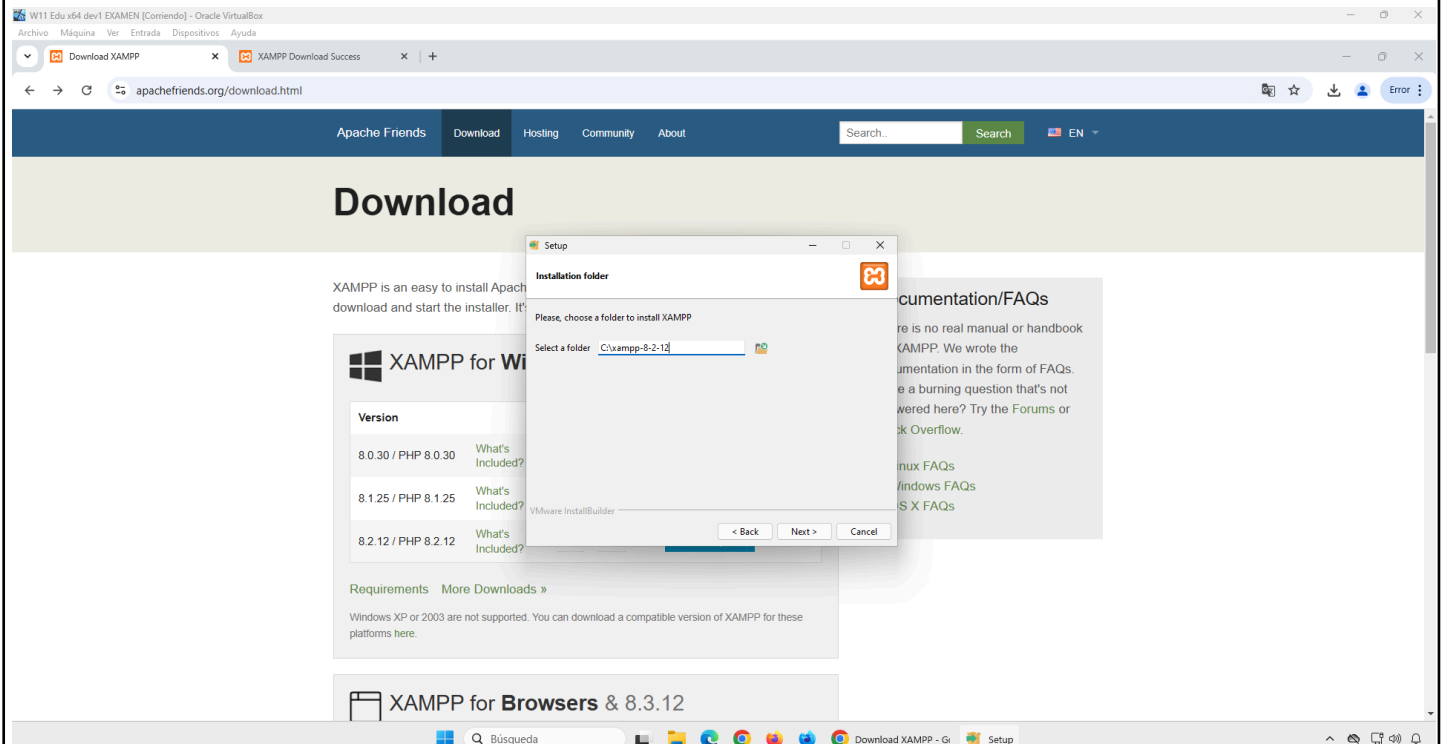
Acceder a <https://www.apachefriends.org/download.html> y descargar el instalador con la última versión de PHP.



The screenshot shows the Apache Friends website's 'Download' page. The page has a dark blue header with navigation links: Apache Friends, Download, Hosting, Community, and About. A search bar is on the right. The main content area is titled 'Download' and features a table of XAMPP versions for Windows. The table lists three versions: 8.0.30, 8.1.25, and 8.2.12, each with a 'Download (64 bit)' button. To the right of the table is a 'Documentation/FAQs' section. Below the table, there are links for 'Requirements' and 'More Downloads'. At the bottom, there is a section for 'XAMPP for Browsers & Microsoft Edge'.

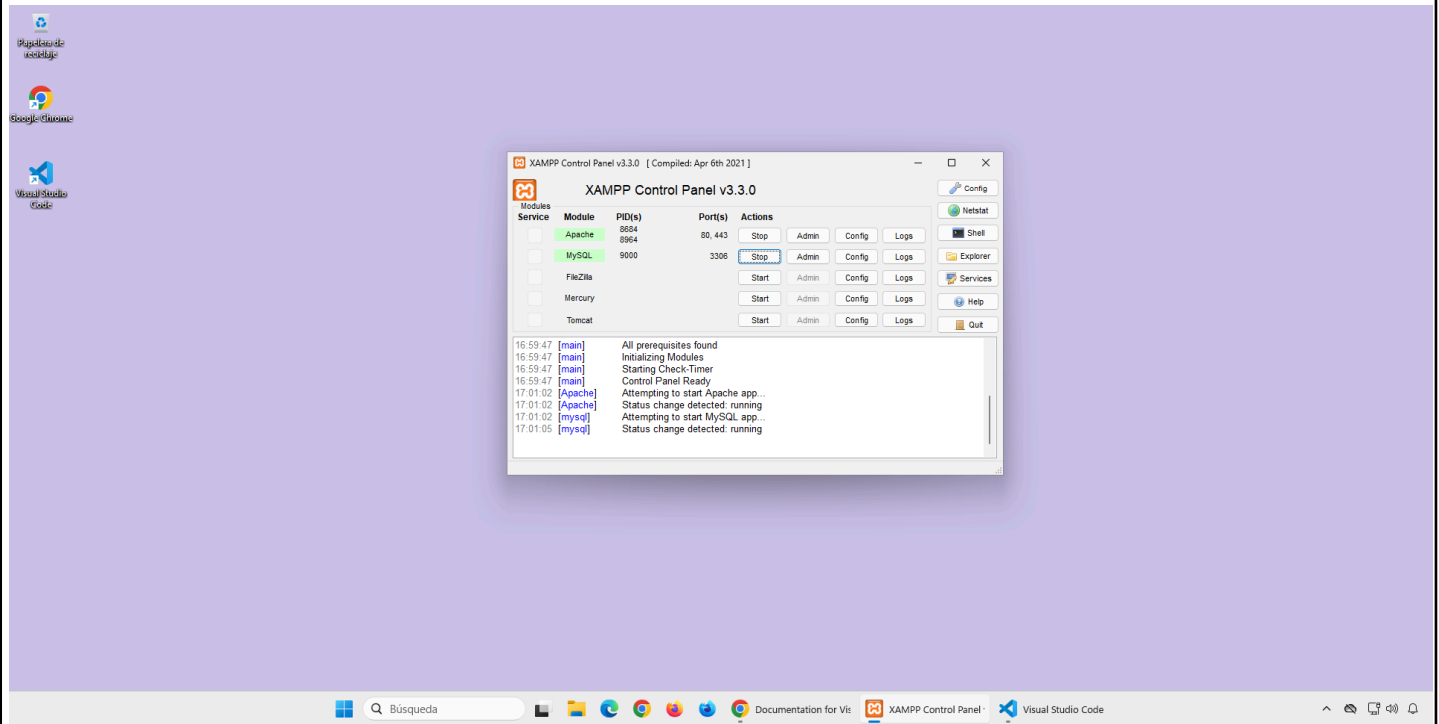
Version	Checksum	Size
8.0.30 / PHP 8.0.30	md5 sha1	144 Mb
8.1.25 / PHP 8.1.25	md5 sha1	148 Mb
8.2.12 / PHP 8.2.12	md5 sha1	149 Mb

Indicar la versión de PHP instalada en la carpeta de instalación de XamPP durante la instalación.



The screenshot shows the XAMPP installation setup window. The window is titled 'Setup' and has a 'Next >' button. The 'Installation folder' section is active, showing a text box with the path 'C:\xampp-8-2-12'. Below the text box is a 'Select a folder' button. The background shows the same Apache Friends Download page as the previous screenshot.

Al terminar la instalación, si abrimos Apache tenemos los siguientes servicios: Apache (servidor web), MySQL (servidor de base de datos), Filezilla (servidor FTP), Mercury (servidor de correo electrónico) y Tomcat (servidor de servlets para Java). Arrancamos los servidor Apache y MySQL (dando permisos al firewall si lo requiere aunque para hacer pruebas en localhost no sería necesario).

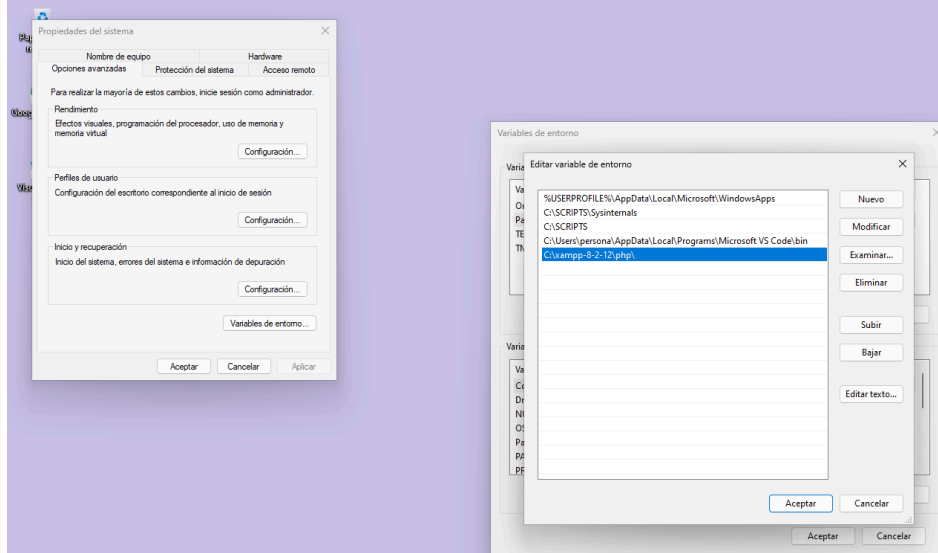


Acceder a <http://localhost> para comprobar que el servidor web se ha iniciado correctamente y funciona.

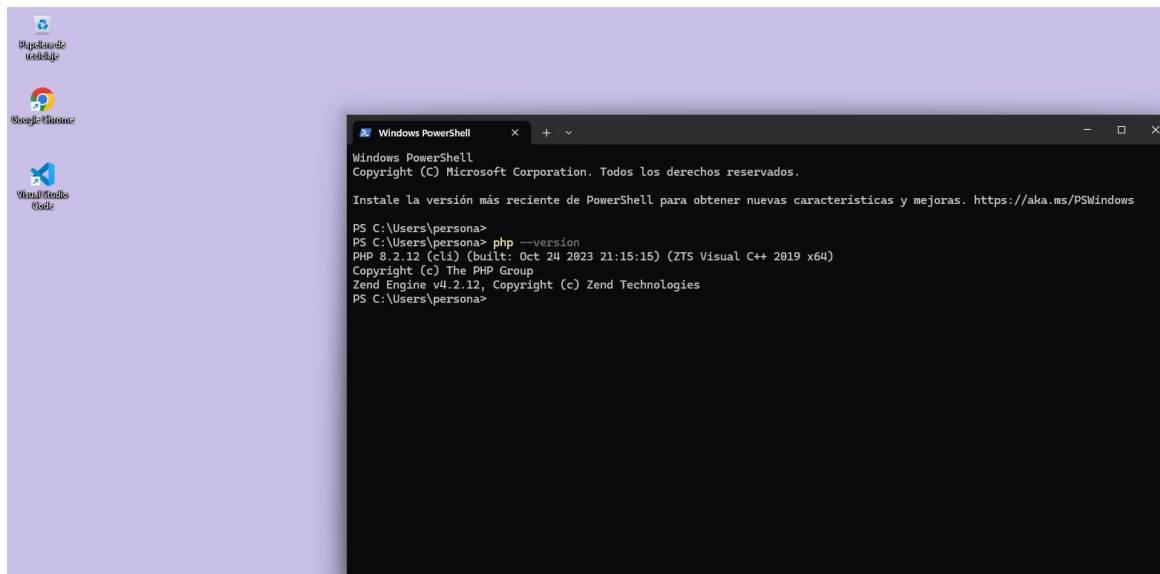


Añadir las variables de entorno PHP.

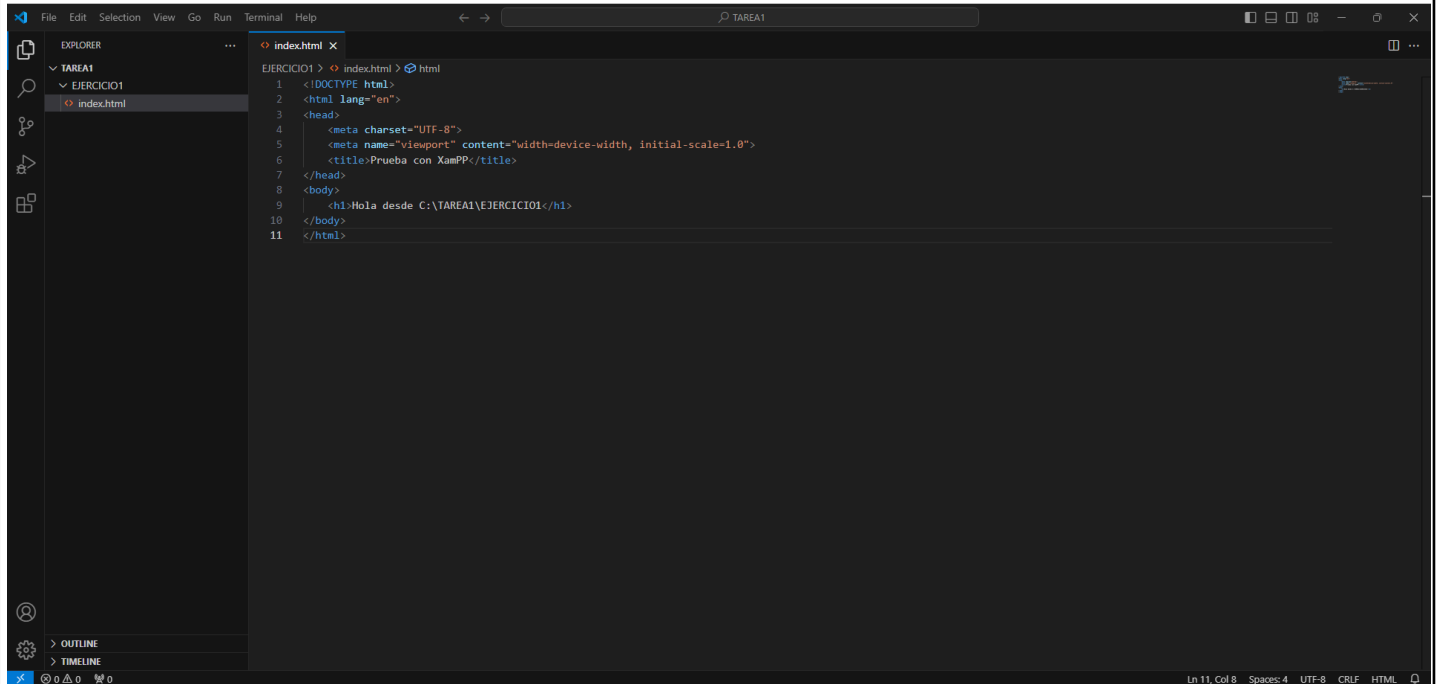
`C:\xampp-8-2-12\php\`



Al hacer esto ya podríamos ejecutar comandos PHP desde la consola.



Crear C:\TAREA1\EJERCICIO1\index.html para mi nueva página.



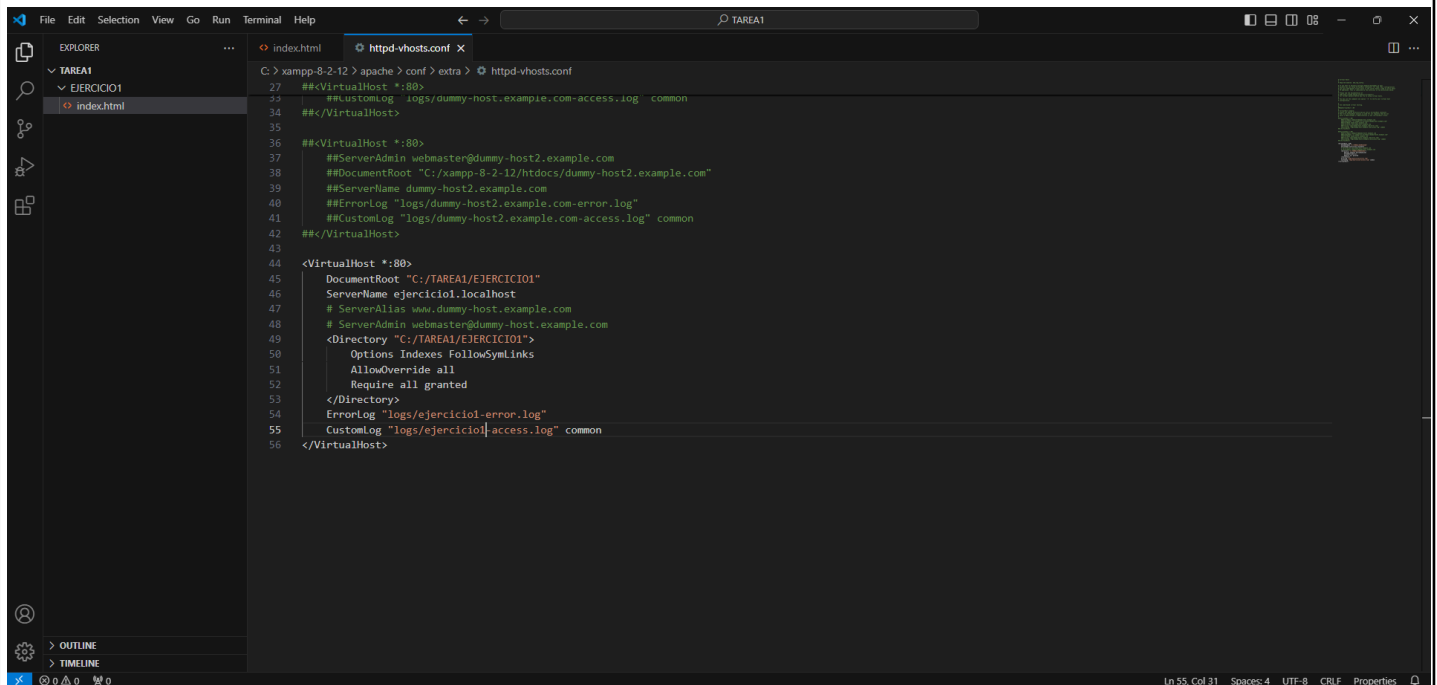
The screenshot shows the Visual Studio Code interface with the Explorer pane on the left showing the file structure: TAREA1 > EJERCICIO1 > index.html. The main editor displays the content of index.html, which is a basic HTML5 document template.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Prueba con XamPP</title>
7 </head>
8 <body>
9   <h1>Hola desde C:\TAREA1\EJERCICIO1</h1>
10 </body>
11 </html>
  
```

The status bar at the bottom indicates the file is at line 11, column 8, using UTF-8 encoding with CRLF line endings.

Configurar un virtual host. Para ello debemos agregar lo siguiente al archivo C:\xampp-8-2-12\apache\conf\extra\httpd-vhosts.conf:



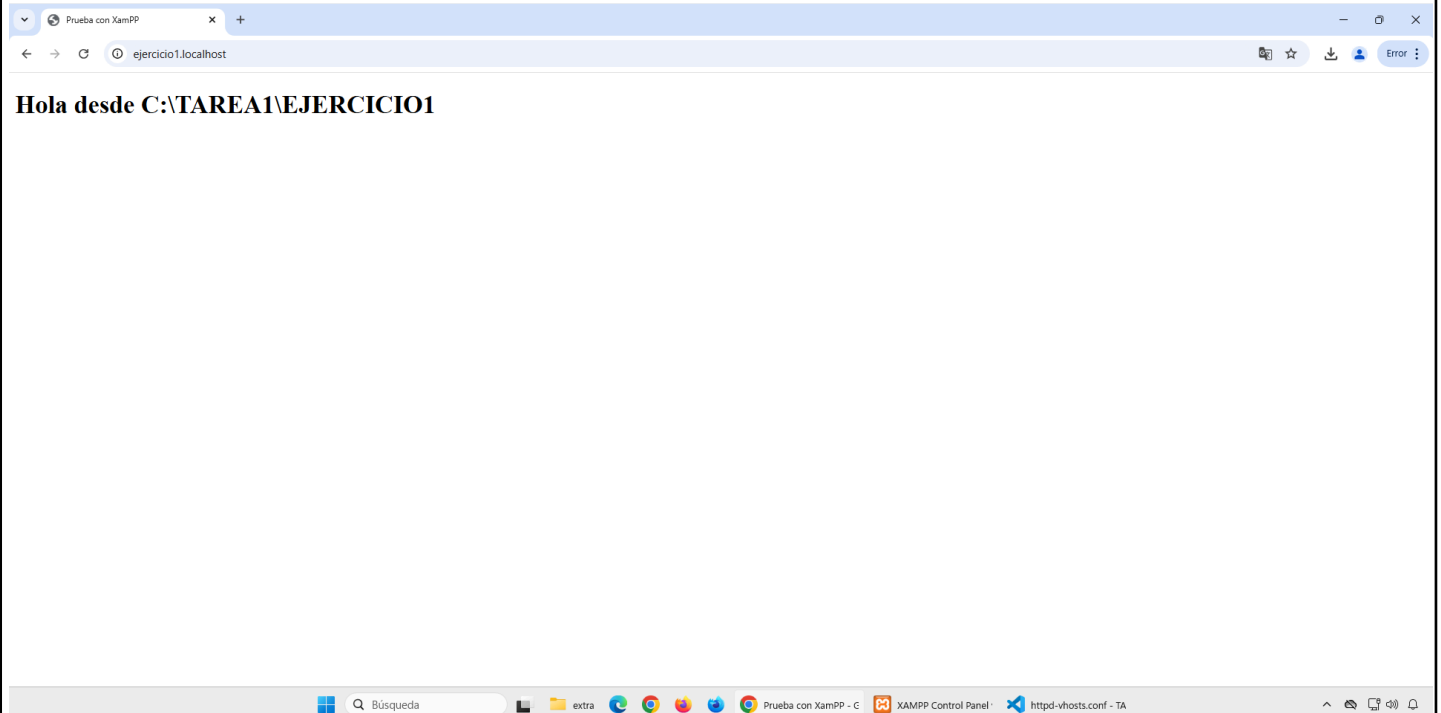
The screenshot shows the Visual Studio Code interface with the Explorer pane on the left showing the file structure: TAREA1 > EJERCICIO1 > index.html. The main editor displays the content of httpd-vhosts.conf, which is a configuration file for Apache's virtual hosts.

```

27 ##<VirtualHost *:80>
28 #ServerRoot "/usr/local/apache2"
29 #ServerName www.example.com:80
30 #ServerAdmin webmaster@dummy-host2.example.com
31 #DocumentRoot "/usr/local/apache2/htdocs"
32 #ErrorLog "logs/dummy-host2.example.com-error.log"
33 #CustomLog "logs/dummy-host2.example.com-access.log" common
34 ##</VirtualHost>
35
36 ##<VirtualHost *:80>
37 #ServerRoot "/usr/local/apache2"
38 #ServerName www.example.com:80
39 #ServerAdmin webmaster@dummy-host2.example.com
40 #DocumentRoot "/usr/local/apache2/htdocs"
41 #ErrorLog "logs/dummy-host2.example.com-error.log"
42 #CustomLog "logs/dummy-host2.example.com-access.log" common
43 ##</VirtualHost>
44
45 <VirtualHost *:80>
46   DocumentRoot "C:/TAREA1/EJERCICIO1"
47   ServerName ejercicio1.localhost
48   # ServerAlias www.dummy-host.example.com
49   # ServerAdmin webmaster@dummy-host.example.com
50   <Directory "C:/TAREA1/EJERCICIO1">
51     Options Indexes FollowSymLinks
52     AllowOverride all
53     Require all granted
54   </Directory>
55   ErrorLog "logs/ejercicio1-error.log"
56   CustomLog "logs/ejercicio1-access.log" common
57 </VirtualHost>
  
```

The status bar at the bottom indicates the file is at line 55, column 31, using UTF-8 encoding with CRLF line endings.

Al acceder a `http://ejercicio1.localhost` el resultado sería el mismo.



**PREGUNTA 2.** Ahora necesitas configurar la depuración en VSC para desarrollo con aplicaciones basadas en PHP.

Enumera los pasos de configuración principales que seguirías y pon capturas de pantalla de los elementos de configuración esenciales.

Al final, pon una captura de pantalla de Visual Studio Code en la que se vea que el depurador está parado en un punto de código del programa en PHP que elijas.

**RESPUESTA:** (4 puntos)

Descargar la .dll adecuada: <https://xdebug.org/docs/install>. Crear el fichero index.php con phpinfo(). En mi caso ejecuto directamente php -i en la terminal y ya obtengo la información necesaria.

```
Windows PowerShell
PS C:\Users\persona>
PS C:\Users\persona> php -i
phpinfo()
PHP Version => 8.2.12

System => Windows NT DESKTOP-25RMGNB 10.0 build 22631 (Windows 11) AMD64
Build Date => Oct 24 2023 21:10:40
Build System => Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler => Visual C++ 2019
Architecture => x64
Configure Command => cscript /nologo /e:javascript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=.\\..\\..\\instantclient\\sdk,shared" "--with-oci8-19=.\\..\\..\\instantclient\\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API => Command Line Interface
Virtual Directory Support => enabled
Configuration File (php.ini) Path =>
Loaded Configuration File => C:\xampp-8-2-12\php\php.ini
Scan this dir for additional .ini files => (none)
Additional .ini files parsed => (none)
PHP API => 20220829
PHP Extension => 20220829
Zend Extension => 420220829
Zend Extension Build => API20220829,TS,VS16
PHP Extension Build => API20220829,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
Zend Max Execution Timers => disabled
IPv6 Support => enabled
DTrace Support => disabled

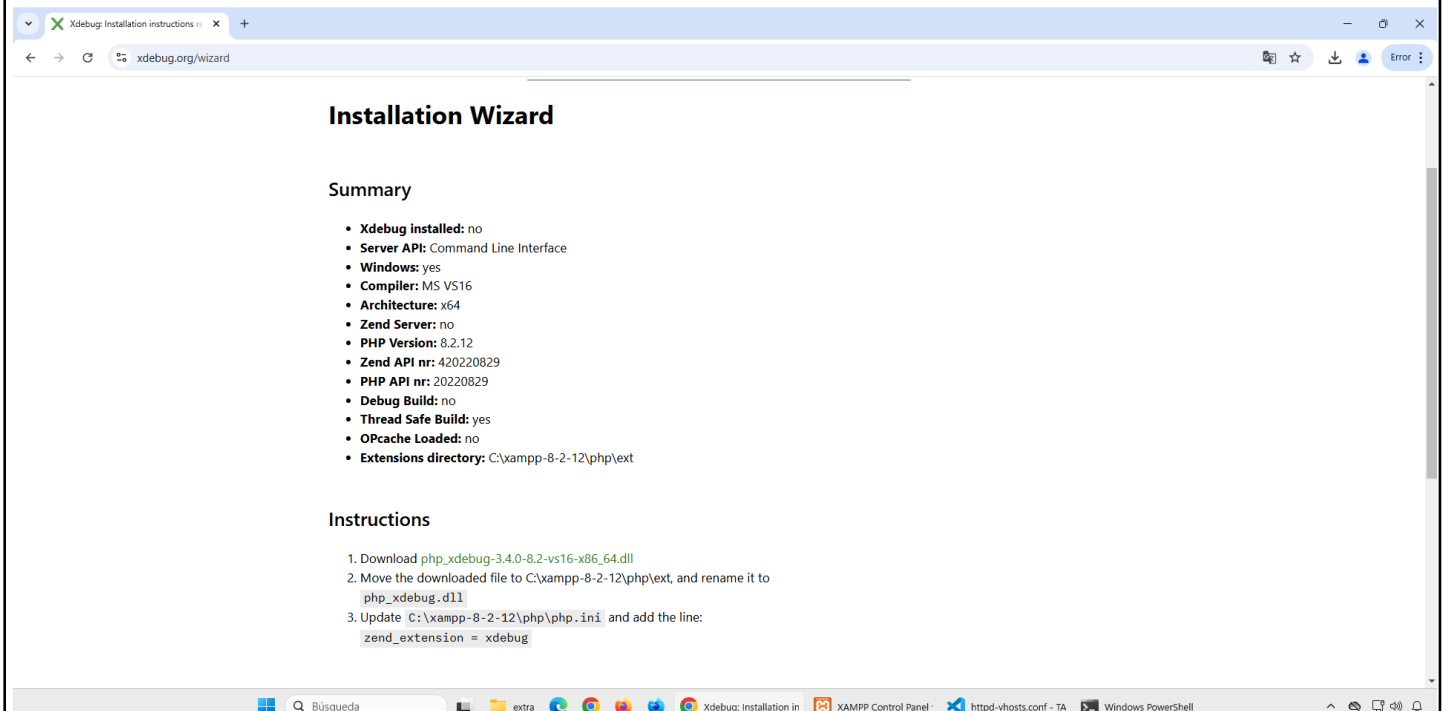
Registered PHP Streams => php, file, glob, data, http, ftp, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports => tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters => convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, zlib.*, bzip2.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.2.12, Copyright (c) Zend Technologies

-----

Configuration
```

Subir al wizzard (asistente) y ver el resultado el análisis de nuestra instalación PHP, así como la versión dll recomendada para nuestro caso.



The screenshot shows the Xdebug Installation Wizard web page. The page has a title "Installation Wizard" and a "Summary" section. The summary lists the following information:

- Xdebug installed: no
- Server API: Command Line Interface
- Windows: yes
- Compiler: MS VS16
- Architecture: x64
- Zend Server: no
- PHP Version: 8.2.12
- Zend API nr: 420220829
- PHP API nr: 20220829
- Debug Build: no
- Thread Safe Build: yes
- OPcache Loaded: no
- Extensions directory: C:\xampp-8-2-12\php\ext

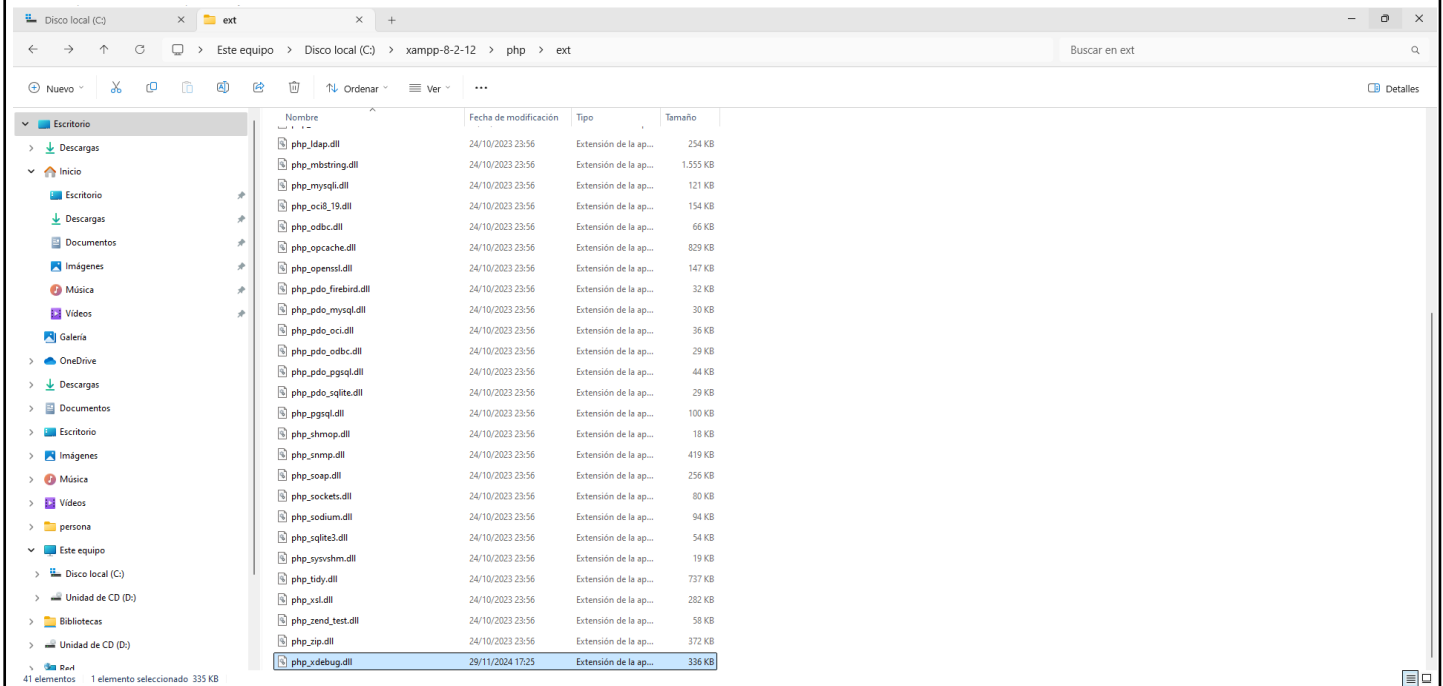
Below the summary, there is an "Instructions" section with three steps:

- Download [php\\_xdebug-3.4.0-8.2-vs16-x86\\_64.dll](#)
- Move the downloaded file to C:\xampp-8-2-12\php\ext, and rename it to [php\\_xdebug.dll](#)
- Update C:\xampp-8-2-12\php\php.ini and add the line:  
`zend_extension = xdebug`

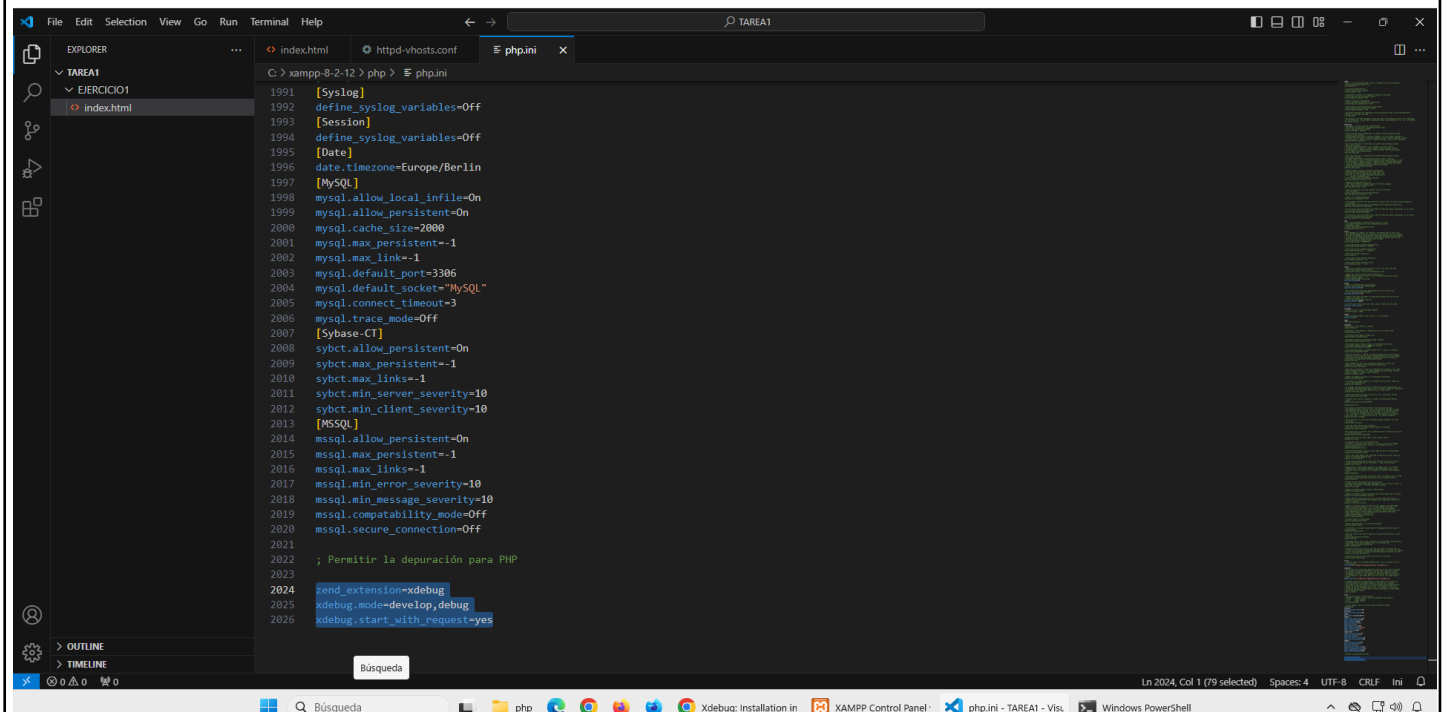
The page is displayed in a web browser with the address bar showing "xdebug.org/wizard". The browser's taskbar at the bottom shows various open applications, including "Xdebug: Installation in", "XAMPP Control Panel", and "httpd-vhosts.conf - TA".



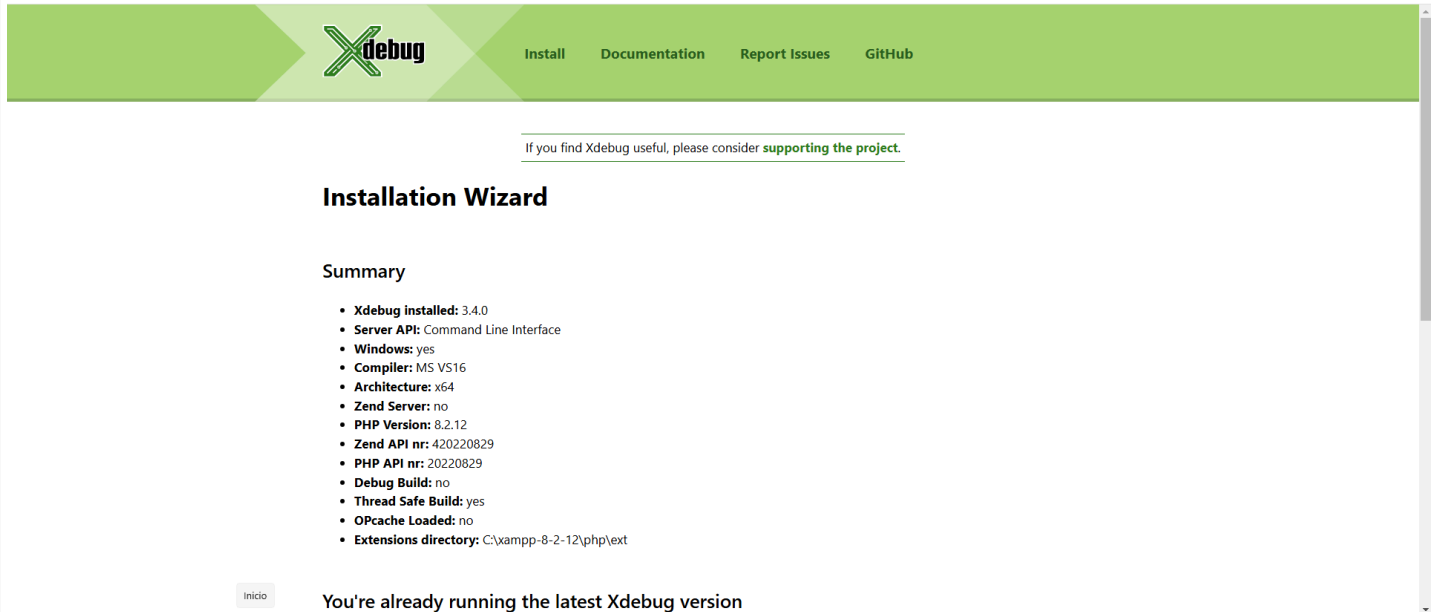
Descargar la dll, moverla a C:\xampp-8-2-12\php\ext y cambiarle el nombre a php\_xdebug.dll.



Configurar el fichero C:\xampp-8-2-12\php\php.ini para la nuva extensión. Añadir las siguientes líneas al final del archivo:



Utilizo el asistente nuevamente para comprobar que se ha instalado correctamente, aunque podría también utilizar la función `phpinfo()`.

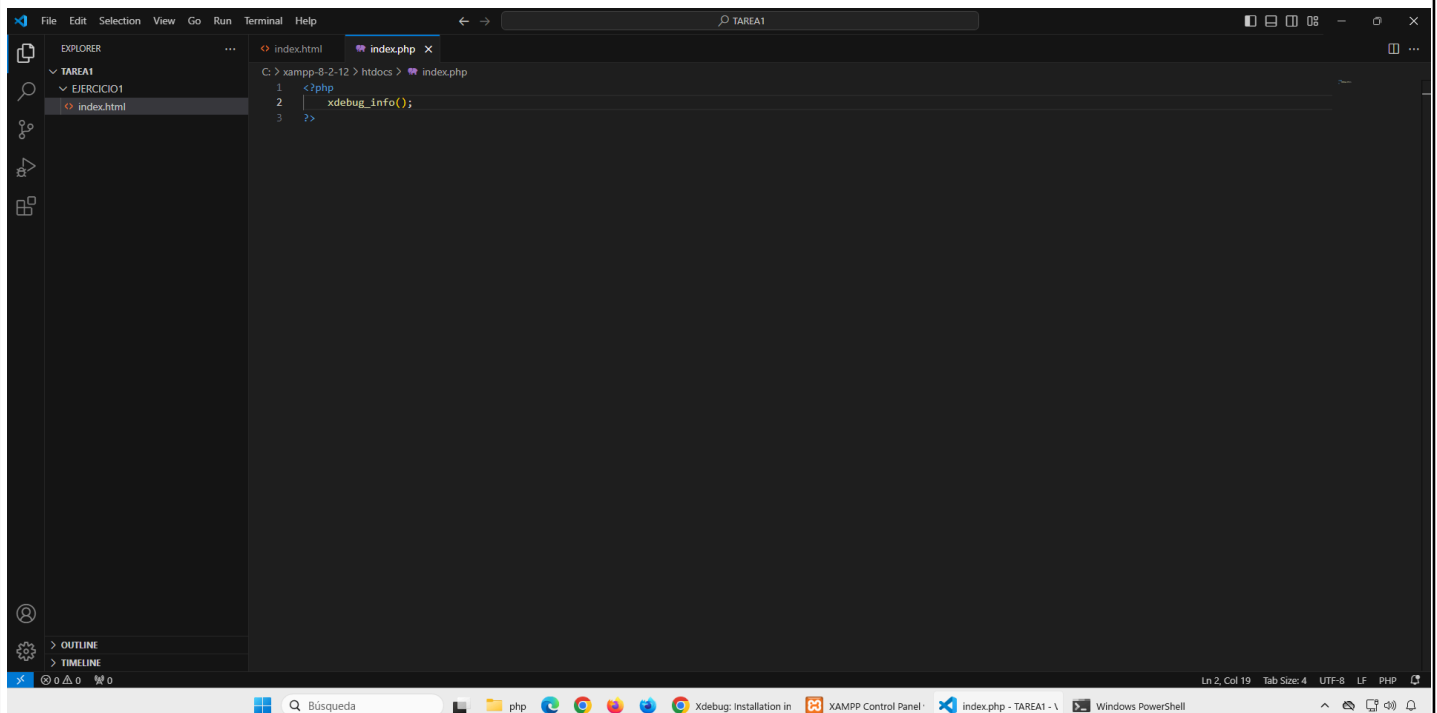


The screenshot shows the Xdebug Installation Wizard interface. At the top, there's a green header with the Xdebug logo and links for Install, Documentation, Report Issues, and GitHub. Below the header, a message says: "If you find Xdebug useful, please consider [supporting the project](#)." The main section is titled "Installation Wizard" and contains a "Summary" of the installation status. The summary lists the following details:

- Xdebug installed: 3.4.0
- Server API: Command Line Interface
- Windows: yes
- Compiler: MS VS16
- Architecture: x64
- Zend Server: no
- PHP Version: 8.2.12
- Zend API nr: 420220829
- PHP API nr: 20220829
- Debug Build: no
- Thread Safe Build: yes
- OPcache Loaded: no
- Extensions directory: C:\xampp-8-2-12\php\ext

At the bottom, there is a button labeled "Inicio" and a message: "You're already running the latest Xdebug version".

Comprobar que funciona: función `xdebug_info()`. Añado lo siguiente al archivo `index.php` del directorio `htdocs` para comprobarlo:



The screenshot shows a Visual Studio Code editor window with the file `index.php` open. The code in the editor is as follows:

```
1 <?php
2     xdebug_info();
3 ?>
```

The Explorer sidebar on the left shows the project structure with `TAREA1` and `EJERCICIO1` folders, and `index.html` file. The status bar at the bottom indicates the current file is `index.php` in the `TAREA1` workspace, using UTF-8 encoding and LF line endings.

**Xdebug**

Version: 3.4.0  
Support Xdebug on Patreon, GitHub, or as a business

Enabled Features (through 'xdebug.mode' setting)		
Feature	Enabled/Disabled	Docs
Development Helpers	✓ enabled	
Coverage	X disabled	
GC Stats	X disabled	
Profiler	X disabled	
Step Debugger	✓ enabled	
Tracing	X disabled	

Optional Features	
Compressed File Support	yes (gzip)
Clock Source	GetSystemTimePreciseAsFileTime
'xdebug://gateway' pseudo-host support	no
'xdebug://nameserver' pseudo-host support	no
Systemd Private Temp Directory	not enabled

**Diagnostic Log**

Message	Docs
[Step Debug] Time-out connecting to debugging client, waited: 200 ms. Tried: localhost:9003 (through xdebug_client_host/xdebug_client_port).	

**Step Debugging**

Debugger	Not Active	Docs

**PHP**

Build Configuration

Instalamos la extensión XDebug para PHP.

**PHP Debug** v1.35.0  
Xdebug xdebug.org 12,798,241 ★★★★★ (135) Sponsor

Debug support for PHP with Xdebug

Details Features Changelog

**PHP Debug Adapter for Visual Studio Code**

vs marketplace v1.35.0 downloads 49M rating 4.1/5 (135) build https://github.com/badges/shields/issues/8671 codecov 6.7%

code style prettier semantic release

```

1 <?php
2
3 define('TEST_CONSTANT', 123);
4
5 echo "Hello World!";
6
7 $variableThatIsNotSet;
8 $dataArray = array(1, 'test' => 2);
9 $largeArray = array_fill(0, 100, 'test');
10 $arrayWithSpaceKey = array('space key' => 1);
11 $object = new DateTime;
12 $boolean = true;
13 $float = 1.23;
14 $string = '123';
15 $null = null;
16 $variableThatIsNotSet = uninitialized;
17 $superGlobal;
18
19 // User Defined Constants
20
21 function a_function($parameter)
22 {
23     echo "function breakpoint";
24 }
25
26 // Notice (undefined index)
27 echo $dataArray['undefined_index'];
28

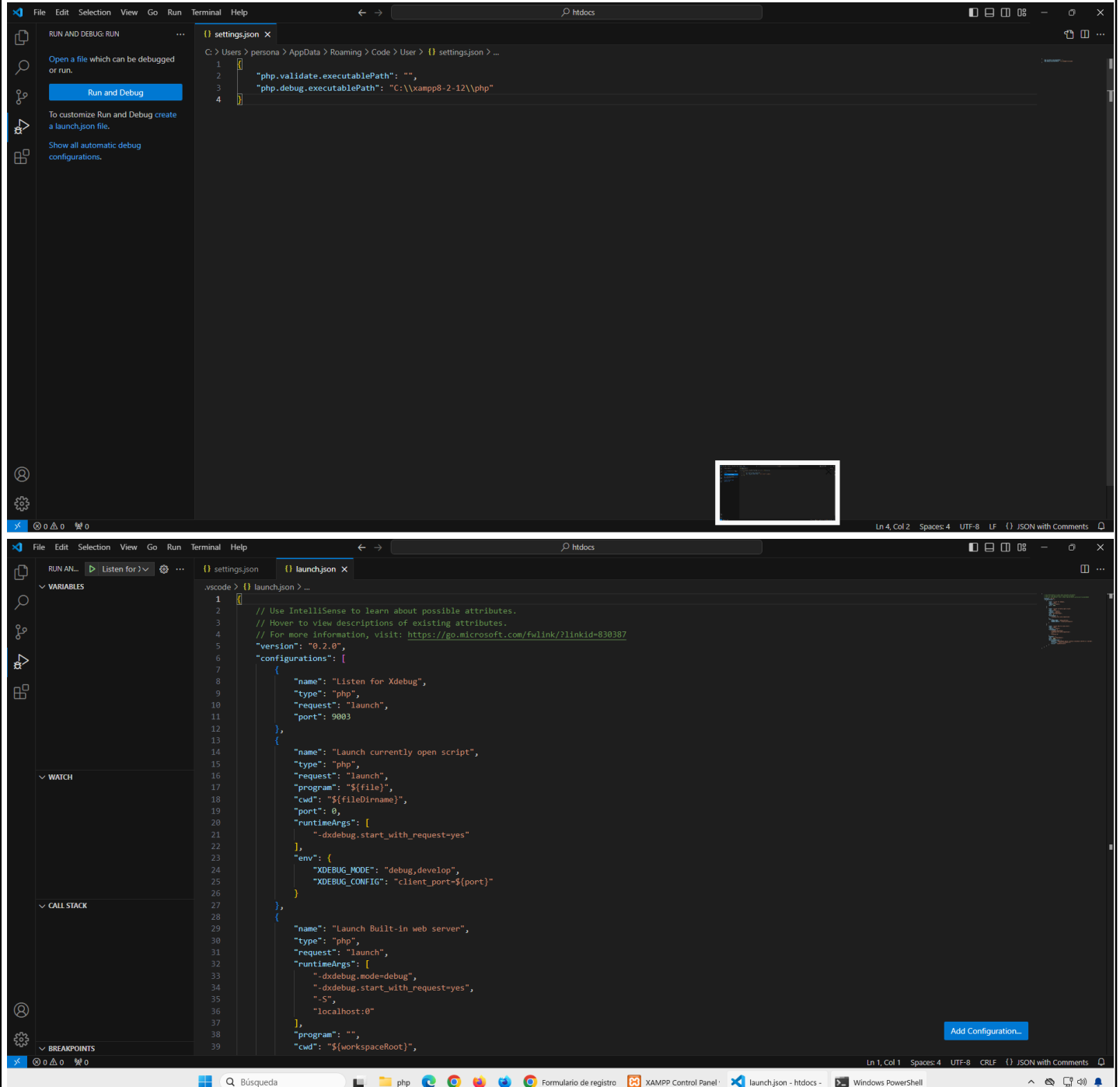
```

Categories: Debuggers

Resources: Marketplace, Issues, Repository, License, Xdebug

More Info: Published: 2015-12-29, 17:55:50; Last released: 2024-07-15, 23:27:33; Identifier: xdebug.php-debug

Crear el archivo launch.json para añadir los modos de debug por defecto. Se genera de forma automática pulsando en el enlace que aparece tras instalar la extensión. Ahora ya podemos debuggear la aplicación PHP en los siguientes modos:



Creo una aplicación sencilla en PHP.

```

formulario.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Formulario de registro</title>
7 </head>
8 <body>
9 <h2>Formulario de registro</h2>
10 <form action="procesarFormulario.php" method="post">
11 <label for="nombre">Nombre</label>
12 <input type="text" id="nombre" name="nombre" required>
13 <label for="email">Email</label>
14 <input type="email" id="email" name="email" required>
15 <button type="submit">Registrar</button>
16 </form>
17 </body>
18 </html>

procesarFormulario.php
1 <?php
2 if (isset($_POST['nombre']) && isset($_POST['email'])) {
3     $nombre = htmlspecialchars($_POST['nombre']);
4     $email = htmlspecialchars($_POST['email']);
5     echo "Hola $nombre, tu email $email ha sido registrado";
6 } else {
7     echo "Por favor, completa el formulario"
8 }
9 ?>
    
```

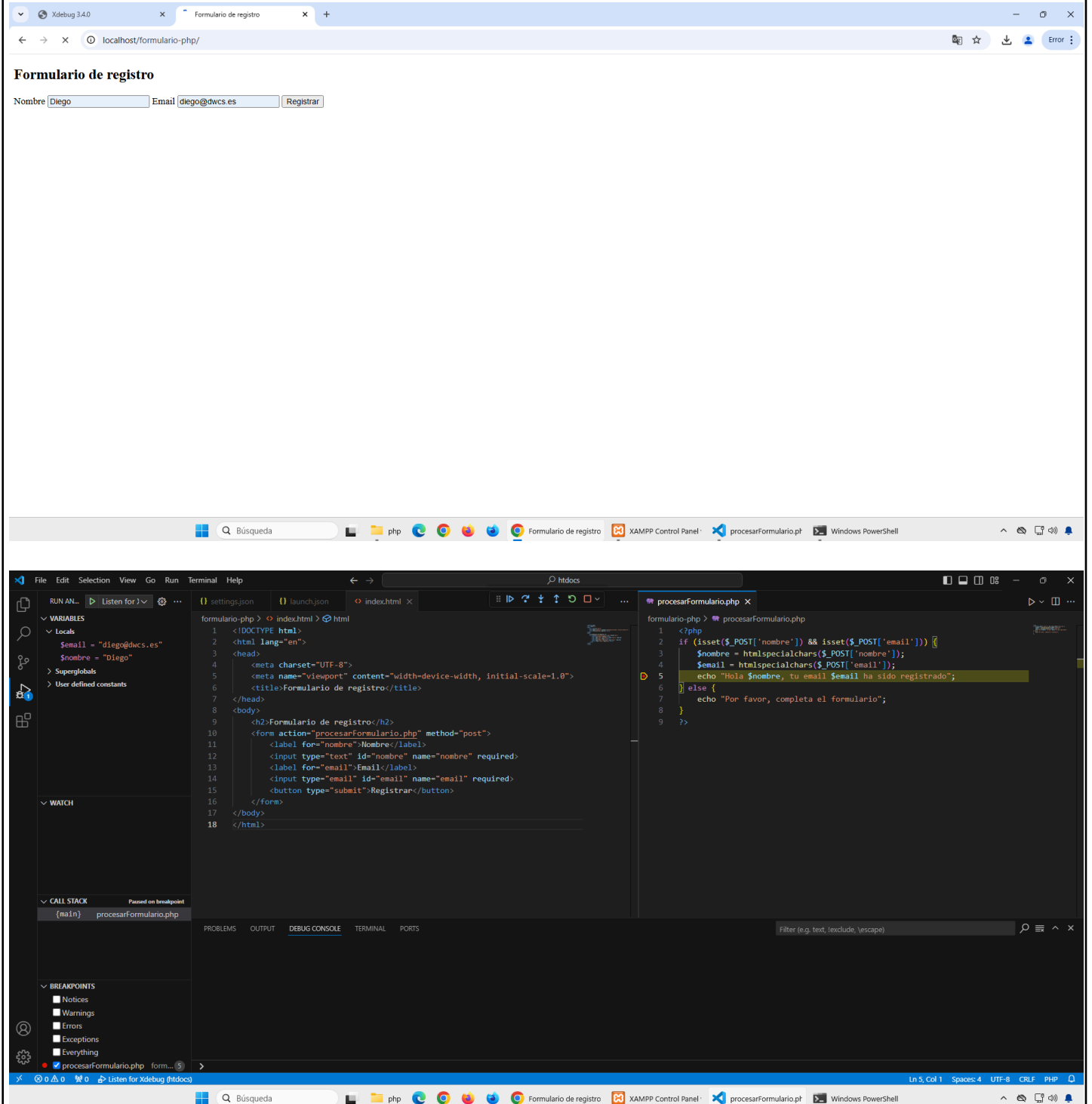
Añadir un breakpoint para comprobar que funciona (se detiene la ejecución en ese punto) e iniciar el modo debug haciendo clic en el siguiente botón (Play).

```

formulario.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Formulario de registro</title>
7 </head>
8 <body>
9 <h2>Formulario de registro</h2>
10 <form action="procesarFormulario.php" method="post">
11 <label for="nombre">Nombre</label>
12 <input type="text" id="nombre" name="nombre" required>
13 <label for="email">Email</label>
14 <input type="email" id="email" name="email" required>
15 <button type="submit">Registrar</button>
16 </form>
17 </body>
18 </html>

procesarFormulario.php
1 <?php
2 if (isset($_POST['nombre']) && isset($_POST['email'])) {
3     $nombre = htmlspecialchars($_POST['nombre']);
4     $email = htmlspecialchars($_POST['email']);
5     echo "Hola $nombre, tu email $email ha sido registrado";
6 } else {
7     echo "Por favor, completa el formulario"
8 }
9 ?>
    
```

Al enviar los datos desde el formulario se detiene en el el breakpoint.



The image shows a web browser window at the top and a code editor window at the bottom. The browser window displays a registration form titled "Formulario de registro" with fields for "Nombre" (containing "Diego") and "Email" (containing "diego@dwcs.es"), and a "Registrar" button. The code editor window shows the PHP file "procesarFormulario.php" with the following code:

```

1 <?php
2 if (isset($_POST['nombre']) && isset($_POST['email'])) {
3     $nombre = htmlspecialchars($_POST['nombre']);
4     $email = htmlspecialchars($_POST['email']);
5     echo "Hola $nombre, tu email $email ha sido registrado";
6 } else {
7     echo "Por favor, completa el formulario";
8 }
9 ?>

```

The code editor also shows the HTML file "index.html" with the form structure:

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Formulario de registro</title>
7 </head>
8 <body>
9     <h2>Formulario de registro</h2>
10    <form action="procesarFormulario.php" method="post">
11        <label for="nombre">Nombre</label>
12        <input type="text" id="nombre" name="nombre" required>
13        <label for="email">Email</label>
14        <input type="email" id="email" name="email" required>
15        <button type="submit">Registrar</button>
16    </form>
17 </body>
18 </html>

```

The code editor also shows the "CALL STACK" and "BREAKPOINTS" panels. The "CALL STACK" panel shows the function "procesarFormulario.php" and the "BREAKPOINTS" panel shows a breakpoint set at line 5 of "procesarFormulario.php".

Le doy a continuar desde VSC y se muestran los datos en el navegador.

