

AVISO: Las tareas son trabajo individual y no deben ser compartidas con otros compañer@s. Si se detecta o se sospecha que una tarea es copia total o parcial de otra, serán anuladas las 2 tareas, independientemente de quién realizara la versión original. La nota de ambas tareas será de 0 puntos

En los apartados en los que es necesario entregar las capturas de pantalla, éstas deben tener como fondo de pantalla la plataforma de fpadistancia con tu usuario mostrando claramente la foto de tu perfil. Aquellos apartados/subapartados que no cumplan esta condición no serán corregidos. Por ejemplo:

CONSEJO PARA NOVAT@s: RECUERDA IR GUARDANDO TU TRABAJO (HACIENDO CLIC EN EL DISKETE DE LA BARRA DE HERRAMIENTAS) CADA POCO TIEMPO PARA NO PERDERLO.

1. CASO PRÁCTICO (4 puntos)

Caso práctico

Juan debe explicar a Carlos las necesidades que tendrá un nuevo proyecto para controlar direcciones de correo, para ello le pide que piense detenidamente que tecnología serían las más óptimas y cuáles no. Carlos con los conceptos que le ha explicado se pone manos a la obra.



Quieres programar una aplicación web para gestionar una lista de direcciones de correo. En una primera aproximación, tu aplicación se compondrá de tres páginas:

Una página de presentación, donde explicas el cometido de la aplicación y su funcionamiento. Contiene un enlace que te lleva a la siguiente página.

Una página de introducción de datos, donde cualquiera puede darse de alta introduciendo su nombre y su dirección de correo. Contiene un formulario que, una vez llenado, envía los datos y te lleva a la siguiente página.

Una página de visualización de datos, en la que se muestra la lista, nombres y direcciones de correo, de todos los que se han anotado en la aplicación.

Responde a las siguientes preguntas sobre la aplicación que vas a desarrollar:

- 1) ¿Qué tipo de páginas, estáticas o dinámicas, utilizarás para programar cada una de las páginas que componen tu aplicación? ¿Por qué?

RESPUESTA: (0,40 puntos)

..

- Página de presentación. Usaría una página estática porque el contenido es principalmente información explicativa sobre la aplicación y no cambia nunca, solo contiene un enlace a otra página.
 - Página de introducción de datos: Usaría una página dinámica porque tiene un formulario que recoge datos y los envía a un servidor para su procesamiento, y éstos no son siempre los mismos.
 - Página de visualización de datos: Usaría una página dinámica porque debe mostrar una lista actualizada con los datos de las personas que han sido registradas, lo cual cambia a medida que los usuarios se van inscribiendo.
- 2) Si en la página de introducción de datos quieras comprobar, antes de enviar los datos, que el correo electrónico introducido cumple unas ciertas normas (por ejemplo, que tiene una @), ¿qué tecnología/lenguaje utilizarás?

RESPUESTA: (0,40 puntos)

Usaría JavaScript para validar el formato del correo electrónico en el lado del cliente.

- 3) Si en esa misma página, ahora quieras comprobar que el correo electrónico introducido no se haya introducido anteriormente y ya figure en la lista, ¿qué tecnología/lenguaje utilizarás?

RESPUESTA: (0,40 puntos)

Usaría PHP o cualquier otro lenguaje del lado del servidor para hacer una consulta a la base de datos y comprobar si existe algún usuario con el correo electrónico que introducido.

- 4) ¿Qué arquitecturas puedes usar en el servidor para ejecutar la aplicación? ¿Cómo es el o los lenguajes que se usa en cada una de esas arquitecturas: de guiones, compilado a código nativo o compilado a código intermedio?

RESPUESTA: (0,80 puntos)

- 5) ¿Qué parámetros debes tener en cuenta para decidirte por usar una arquitectura u otra?

RESPUESTA: (0,40 puntos)

- ¿Qué tamaño tiene el proyecto?
- ¿Qué lenguajes de programación conozco? ¿Vale la pena el esfuerzo de aprender uno nuevo?
- ¿Voy a usar herramientas de código abierto o herramientas propietarias? ¿Cuál es el coste de utilizar soluciones comerciales?
- ¿Voy a programar la aplicación yo solo o formaré parte de un grupo de programadores?
- ¿Cuento con algún servidor web o gestor de base de datos disponible o puedo decidir libremente utilizar el que crea necesario?
- ¿Qué tipo de licencia voy a aplicar a la aplicación que desarrolle?

- 6) Si te decides por utilizar una arquitectura AMP para la aplicación ¿qué componentes necesitas instalar en tu servidor para ejecutar la aplicación? Indica algún producto concreto para cada componente.

RESPUESTA: (0,80 puntos)

Para una arquitectura AMP (como LAMP o WAMP) se necesita:

- Sistema operativo. Linux (en LAMP) o Windows (en WAMP).
- Servidor web. Apache procesa las solicitudes HTTP.
- Motor de base de datos. MySQL o MariaDB almacenan los datos.
- Lenguaje de servidor. PHP, Perl o Python procesan la lógica de la aplicación.

- 7) ¿Qué necesitas instalar en tu ordenador para poder desarrollar la aplicación?

RESPUESTA: (0,40 puntos)

Para desarrollar la aplicación se necesita:

- Un servidor local como XAMPP (contiene Apache, MySQL y PHP).
- Un editor de código o IDE, como Visual Studio Code o PHPStorm.
- Navegador web para probar la aplicación.

- 8) Si utilizas el lenguaje PHP para programar la aplicación, ¿cuál será el tipo de datos se utilizará para manipular cada una de las direcciones de correo?

RESPUESTA: (0,40 puntos)

Utilizaría el tipo de dato string (secuencias de caracteres).

2. EJERCICIO PRÁCTICO (6 puntos)

2.1. Supón que tienes que configurar tu entorno de desarrollo (XAMPP) bajo Windows y Visual Studio Code para desarrollar una aplicación EJERCICIO1 y quieres tener todo el código disponible en la carpeta C:\TAREA1\EJERCICIO1 y hacer pruebas de su funcionamiento en la URL <http://ejercicio1.localhost>.

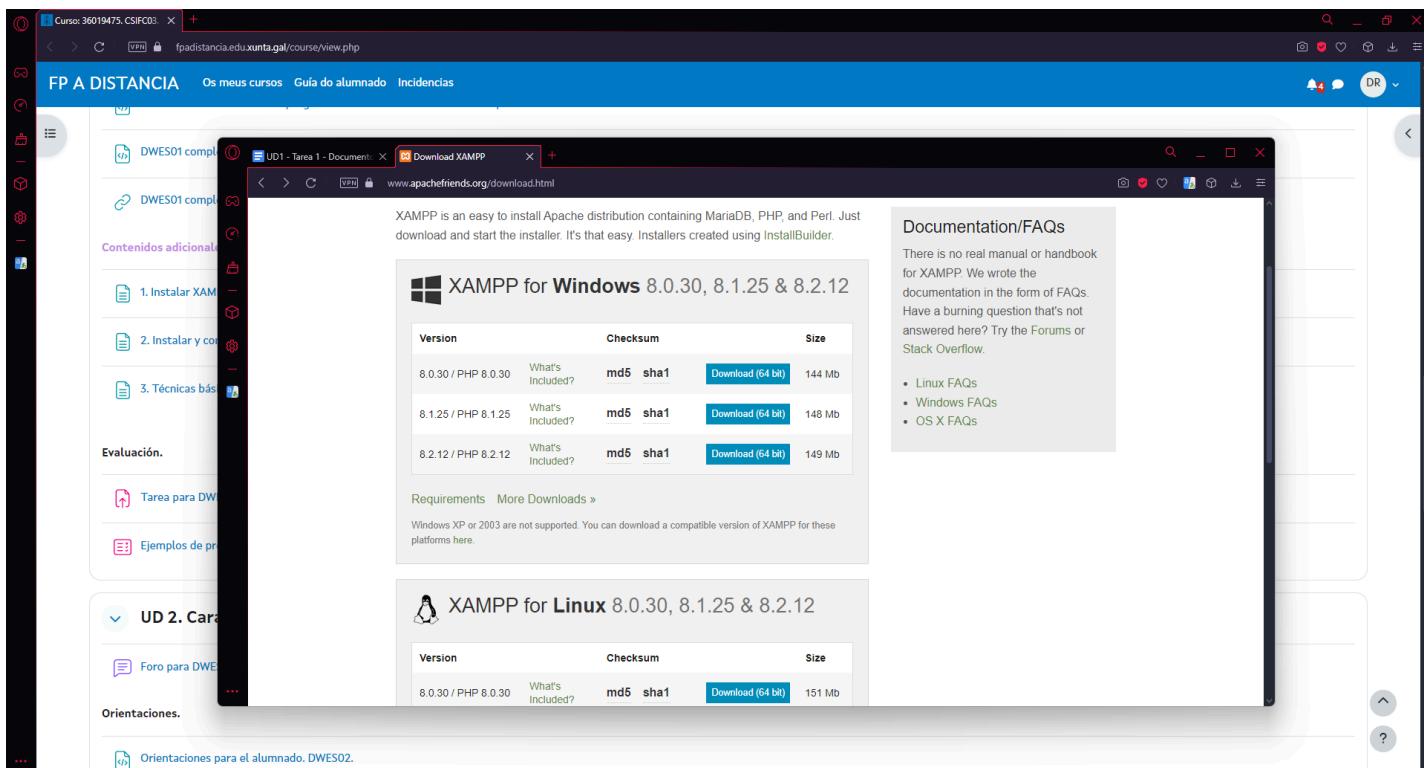
Enumera los pasos principales que seguirías y pon capturas de pantalla de los elementos de configuración esenciales (no hace falta documentar todo el proceso detallando os subpasos de cada paso principal).

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.

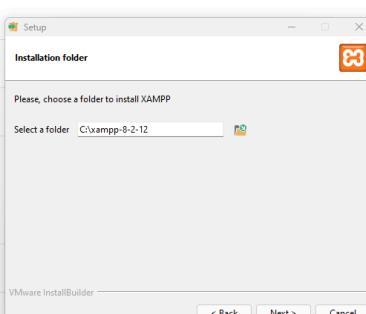
RESPUESTA: (3 puntos)

1- Instalar XAMPP.

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



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



FP A DISTANCIA Os meus cursos Guía do alumnado Incidencias

DWES01 completa. Versión imprimible para uso offline.

DWES01 completa (VERSIÓN EDITABLE FORMATO LIBREOFFICE)

Contenidos adicionales

1. Instalar XAMPP y comprender archivos principales
2. Instalar y configurar Xdebug y comprobar en VSC
3. Técnicas básicas de depuración (debug) en PHP

Evaluación.

Tarea para DWES01.

Ejemplos de preguntas tipo test para examen UD1

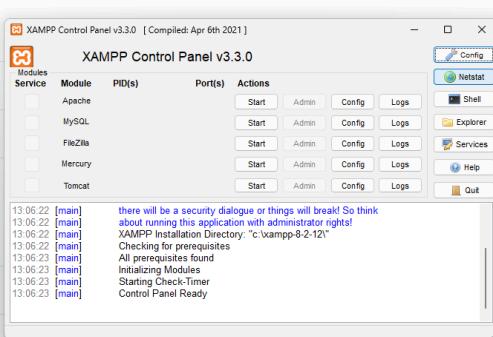
UD 2. Características del lenguaje PHP

Foro para DWES02.

Orientaciones.

Orientaciones para el alumnado. DWES02.

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).



FP A DISTANCIA Os meus cursos Guía do alumnado Incidencias

DWES01 completa. Versión imprimible para uso offline.

DWES01 completa (VERSIÓN EDITABLE FORMATO LIBREOFFICE)

Contenidos adicionales

1. Instalar XAMPP y comprender archivos principales
2. Instalar y configurar Xdebug y comprobar en VSC
3. Técnicas básicas de depuración (debug) en PHP

Evaluación.

Tarea para DWES01.

Ejemplos de preguntas tipo test para examen UD1

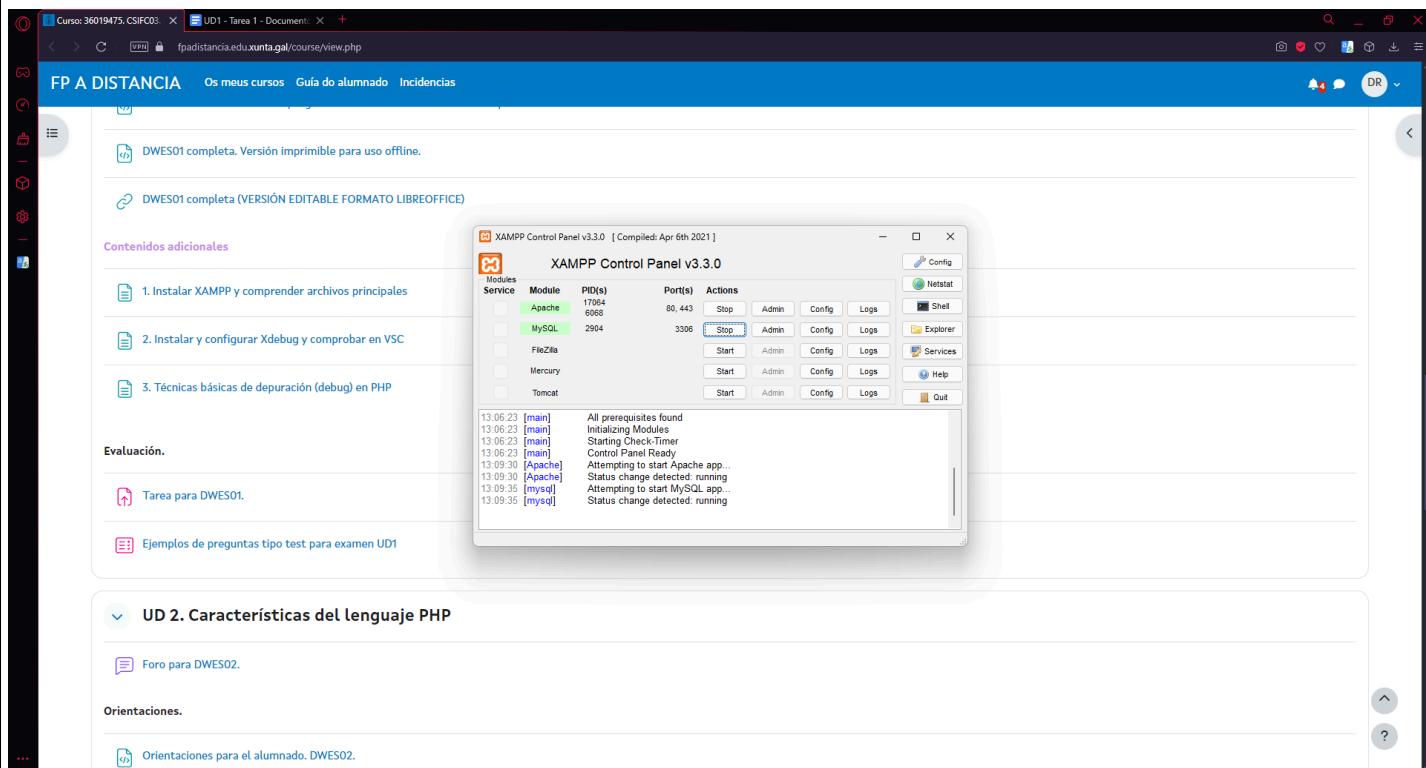
UD 2. Características del lenguaje PHP

Foro para DWES02.

Orientaciones.

Orientaciones para el alumnado. DWES02.

Arrancar los servidores Apache y MySQL (dando permisos al firewall si lo requiere aunque para hacer pruebas en localhost no sería necesario).



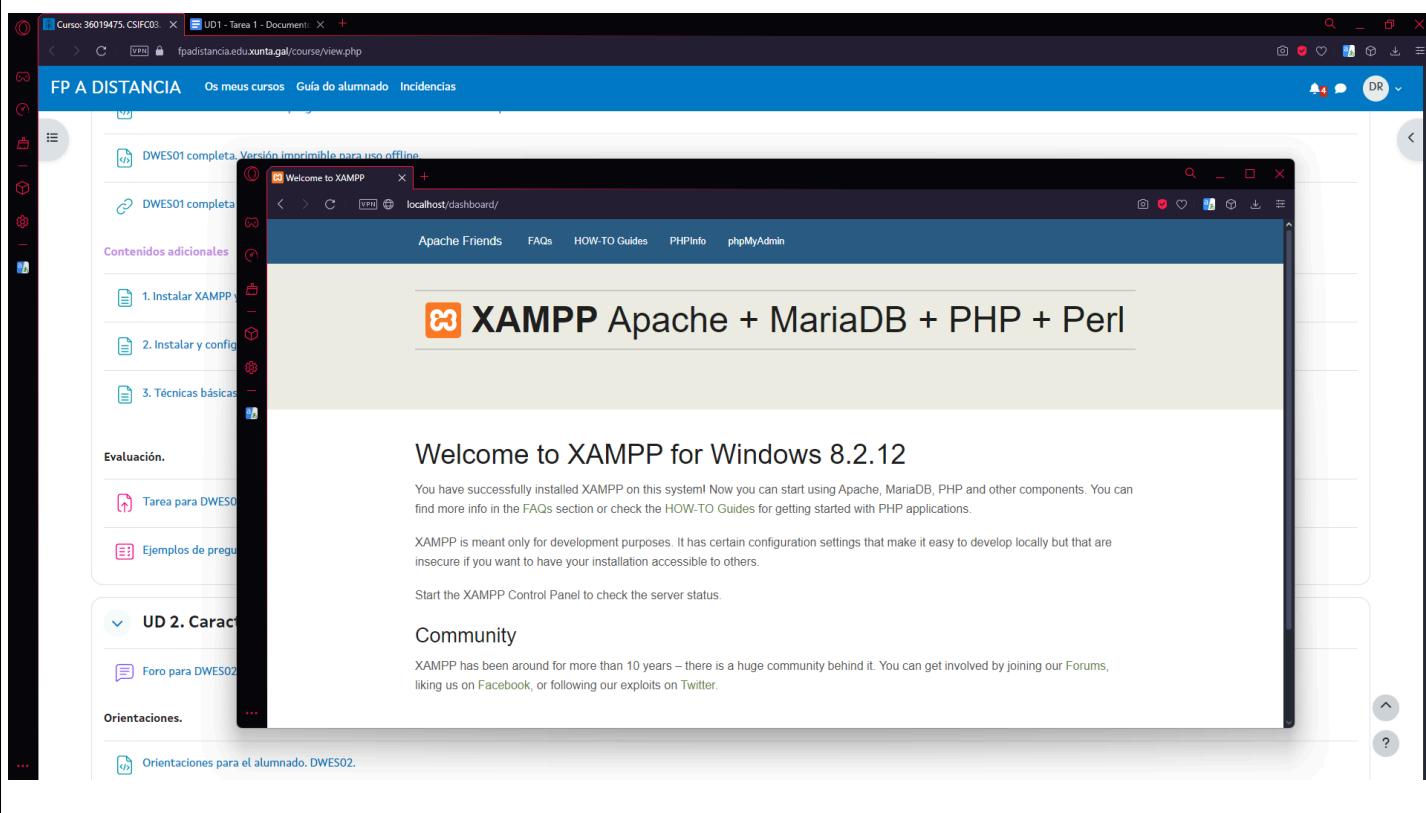
The screenshot shows a web browser window with a course navigation bar at the top. A modal window titled "XAMPP Control Panel v3.3.0" is overlaid on the page. The modal displays logs for the startup of Apache and MySQL services. The logs show:

```

13:06:22 [main] All prerequisites found
13:06:23 [main] Initializing Modules
13:06:23 [main] Starting Check-Timer
13:06:23 [main] Control Panel Ready
13:09:30 [Apache] Attempting to start Apache app...
13:09:30 [Apache] Status change detected: running...
13:09:35 [MySQL] Attempting to start MySQL app...
13:09:35 [MySQL] Status change detected: running...

```

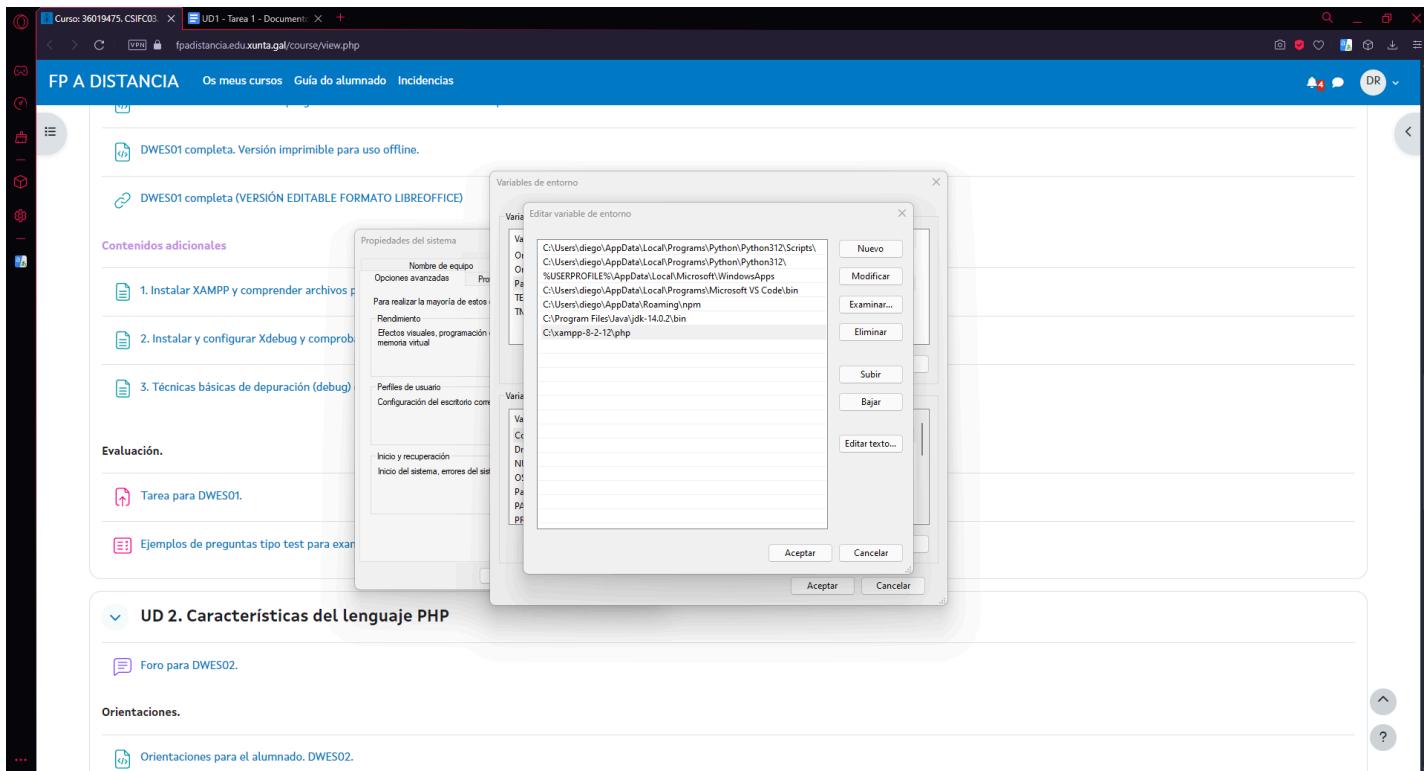
Acceder a <http://localhost> para comprobar que el servidor web se ha iniciado correctamente y funciona.



The screenshot shows a web browser window with a course navigation bar at the top. A modal window titled "Welcome to XAMPP" is overlaid on the page. The modal displays the XAMPP dashboard with the title "XAMPP Apache + MariaDB + PHP + Perl". The dashboard includes links for "Apache Friends", "FAQs", "HOW-TO Guides", "PHPInfo", and "phpMyAdmin". Below the title, it says "Welcome to XAMPP for Windows 8.2.12". It provides information about the successful installation and the purpose of XAMPP. It also encourages users to start the XAMPP Control Panel to check the server status and to engage with the community.

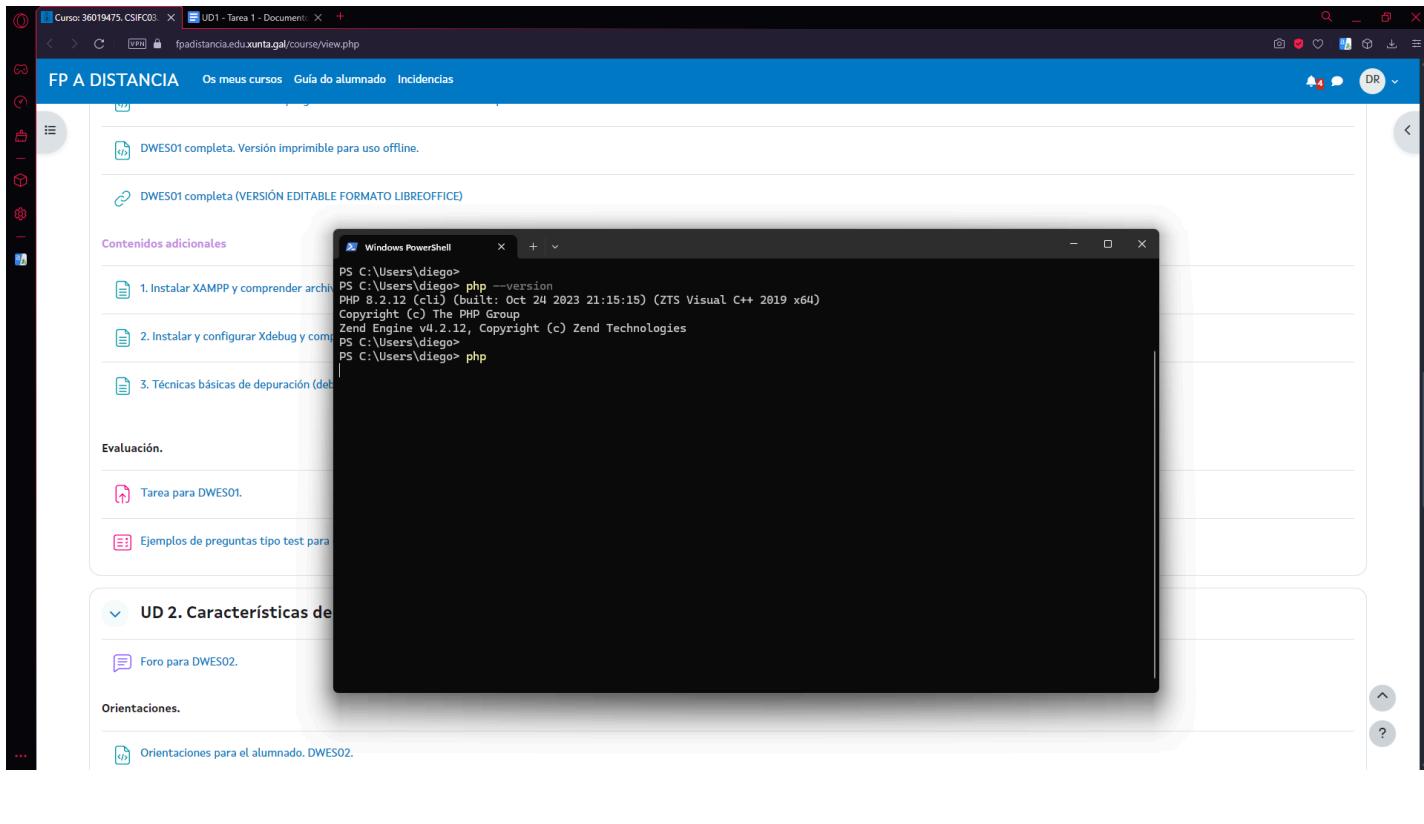
Añadir las variables de entorno PHP

C:\xampp-8-2-12\php\php.exe



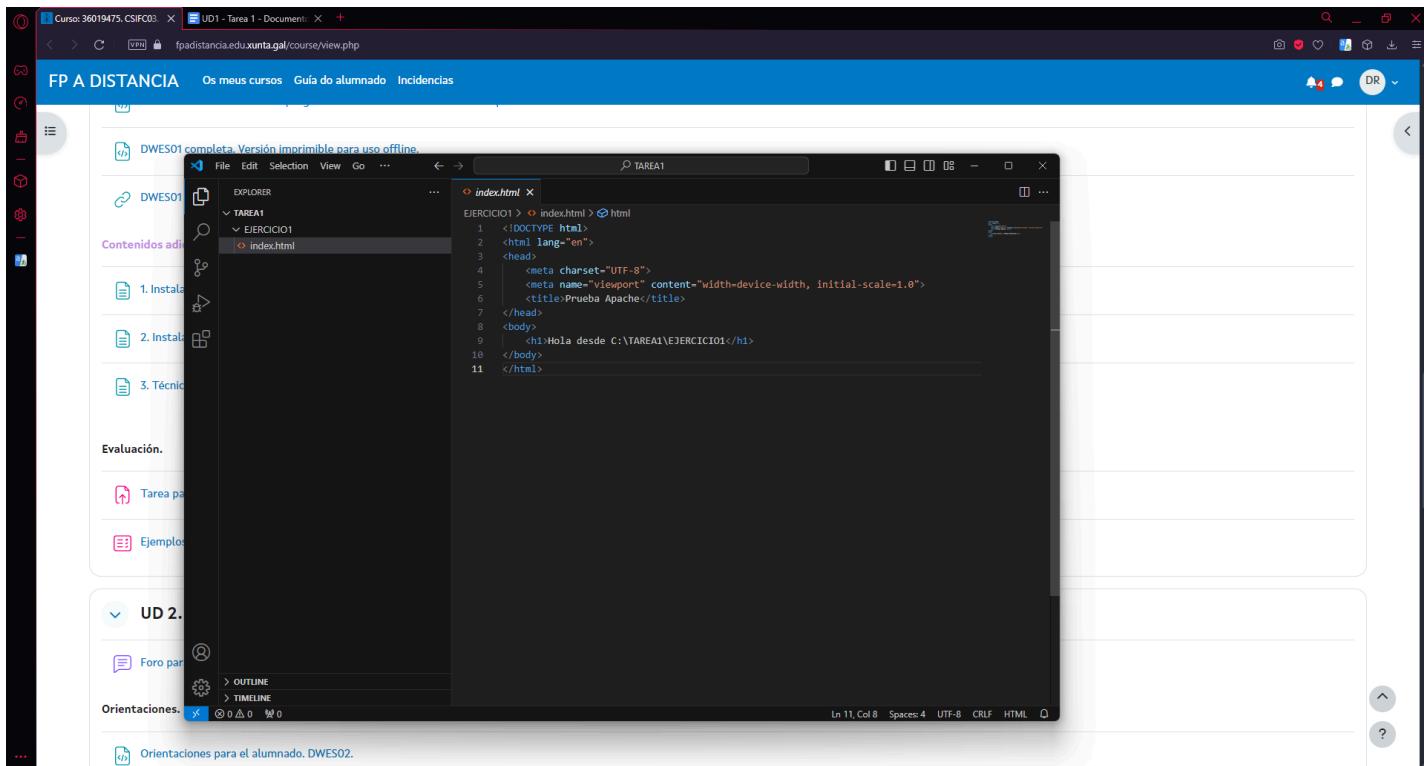
The screenshot shows a web browser window with a course page from 'FP A DISTANCIA'. A modal dialog box titled 'Variables de entorno' (Environment Variables) is open in the foreground. The dialog lists several environment variables, including 'Path' which contains the path 'C:\xampp-8-2-12\php'. The dialog has buttons for 'Nuevo' (New), 'Modificar' (Modify), 'Examinar...' (Browse...), 'Eliminar' (Delete), 'Subir' (Move up), 'Bajar' (Move down), and 'Aceptar' (Accept). The background shows course content like 'DWES01 completa' and 'UD 2. Características del lenguaje PHP'.

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



The screenshot shows a web browser window with a course page. A Windows PowerShell window is embedded within the page, showing the output of a PHP command. The command 'php --version' was run, and the output shows PHP 8.2.12 (cli) built on October 24, 2023. The PowerShell window has a dark theme and is titled 'Windows PowerShell'.

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



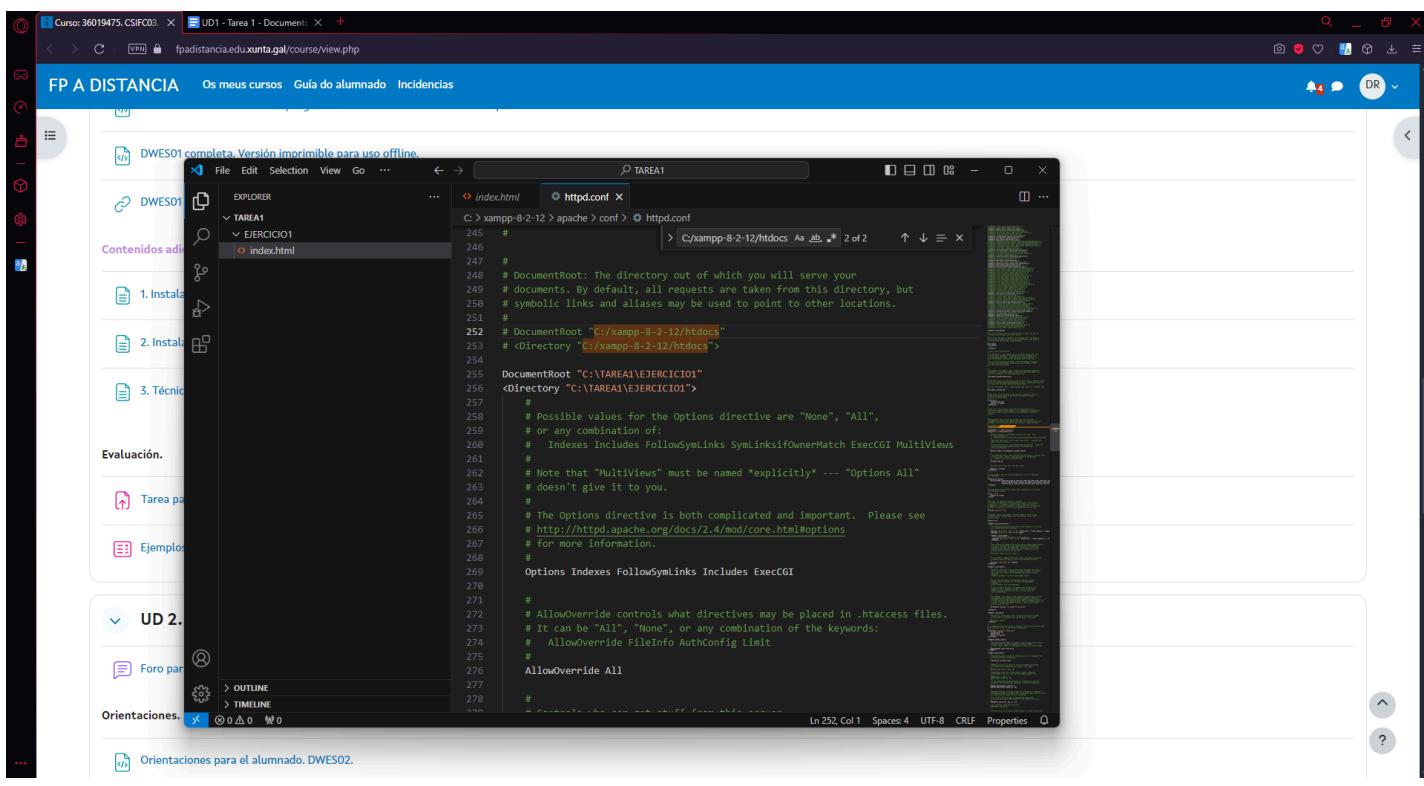
```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prueba Apache</title>
</head>
<body>
    <h1>Hola desde C:\TAREA1\EJERCICIO1</h1>
</body>
</html>

```

Cambiar el DocumentRoot en el archivo de configuración de Apache.

C:\xampp-8-2-12\apache\conf\httpd.conf



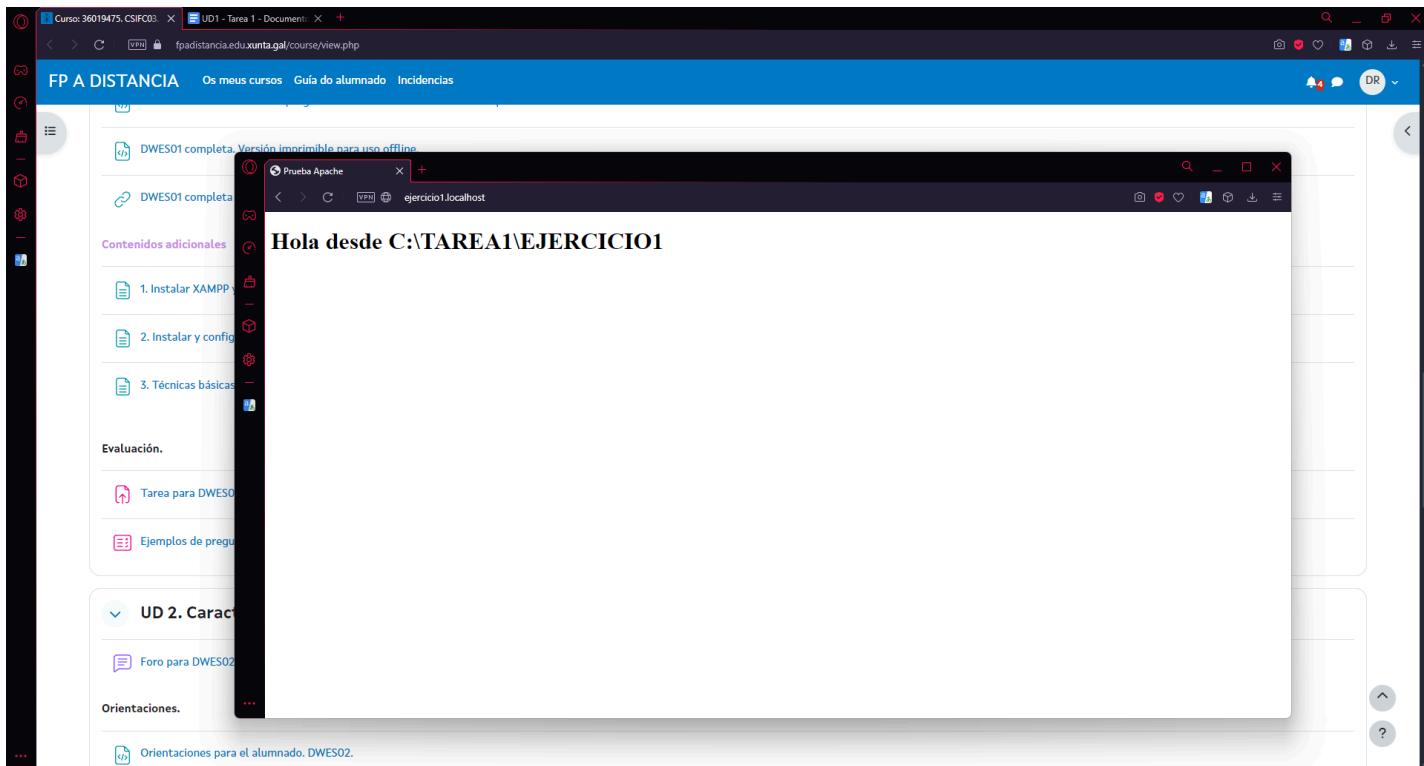
```

# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
# DocumentRoot "C:/xampp-8-2-12/htdocs"
# <Directory "C:/xampp-8-2-12/htdocs">

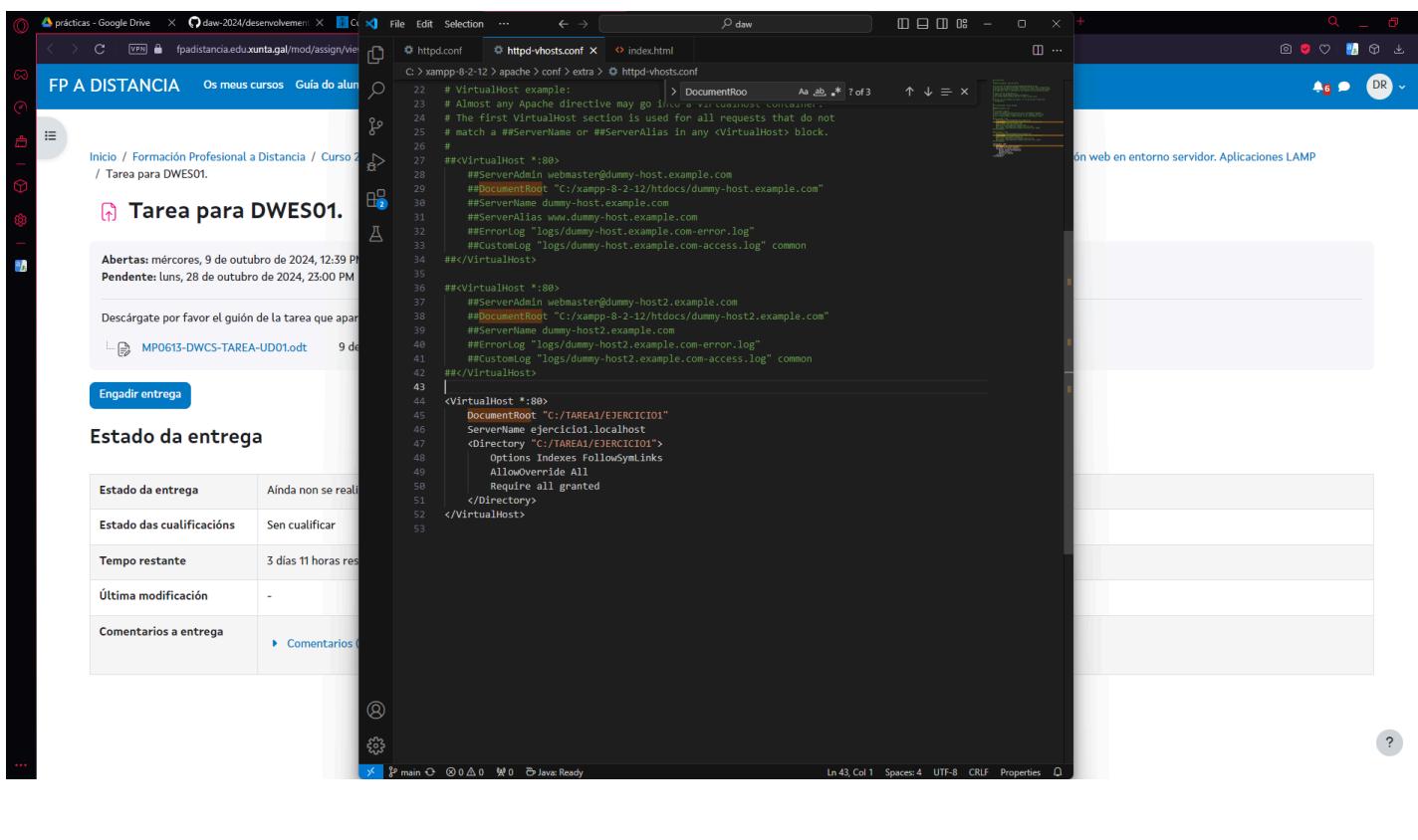
DocumentRoot "C:\TAREA1\EJERCICIO1"
<Directory "C:\TAREA1\EJERCICIO1">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymlinks SymLinksIfOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important. Please see
    # http://httpd.apache.org/docs/2.4/mod/core.html#options
    # for more information.
    #
    Options Indexes FollowSymLinks Includes ExecCGI
    #
    # AllowOverride controls what directives may be placed in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
    #   AllowOverride FileInfo AuthConfig Limit
    #
    AllowOverride All

```

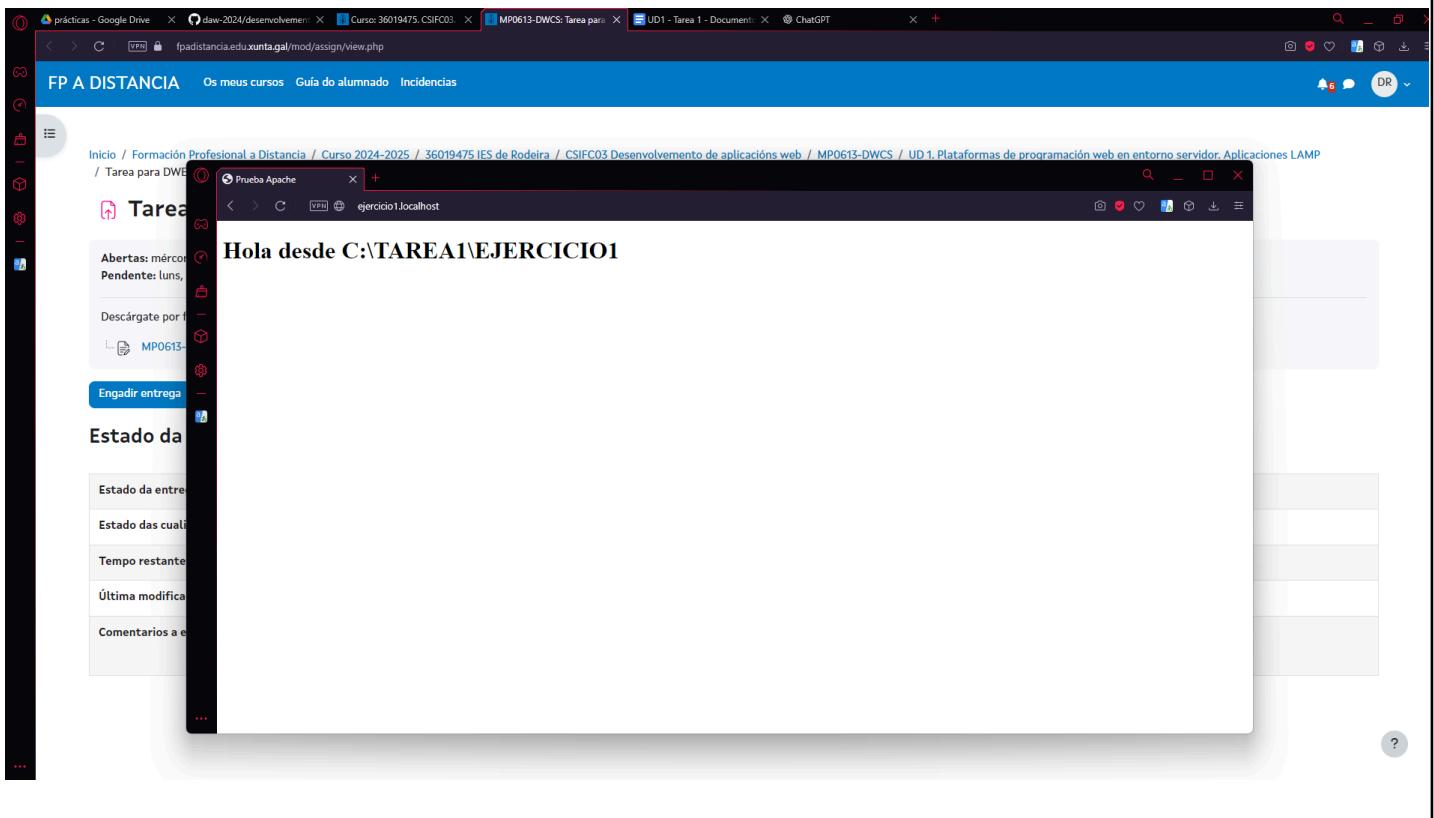
Comprobar que al acceder a <http://ejercicio1.localhost> visualizo mi página como si se estuviese sirviendo desde *htdocs*.



Aún así, lo mejor sería configurar un virtual host. Para ello debemos dejar el *DocumentRoot* como estaba en *C:\xampp-8-2-12\apache\conf\httpd.conf* y agregar lo siguiente al archivo *C:\xampp-8-2-12\apache\conf\extra\httpd-vhosts.conf*.



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



2.2. Ahora necesitas configurar la depuración en VSC para desarrollo con aplicaciones basadas en PHP.

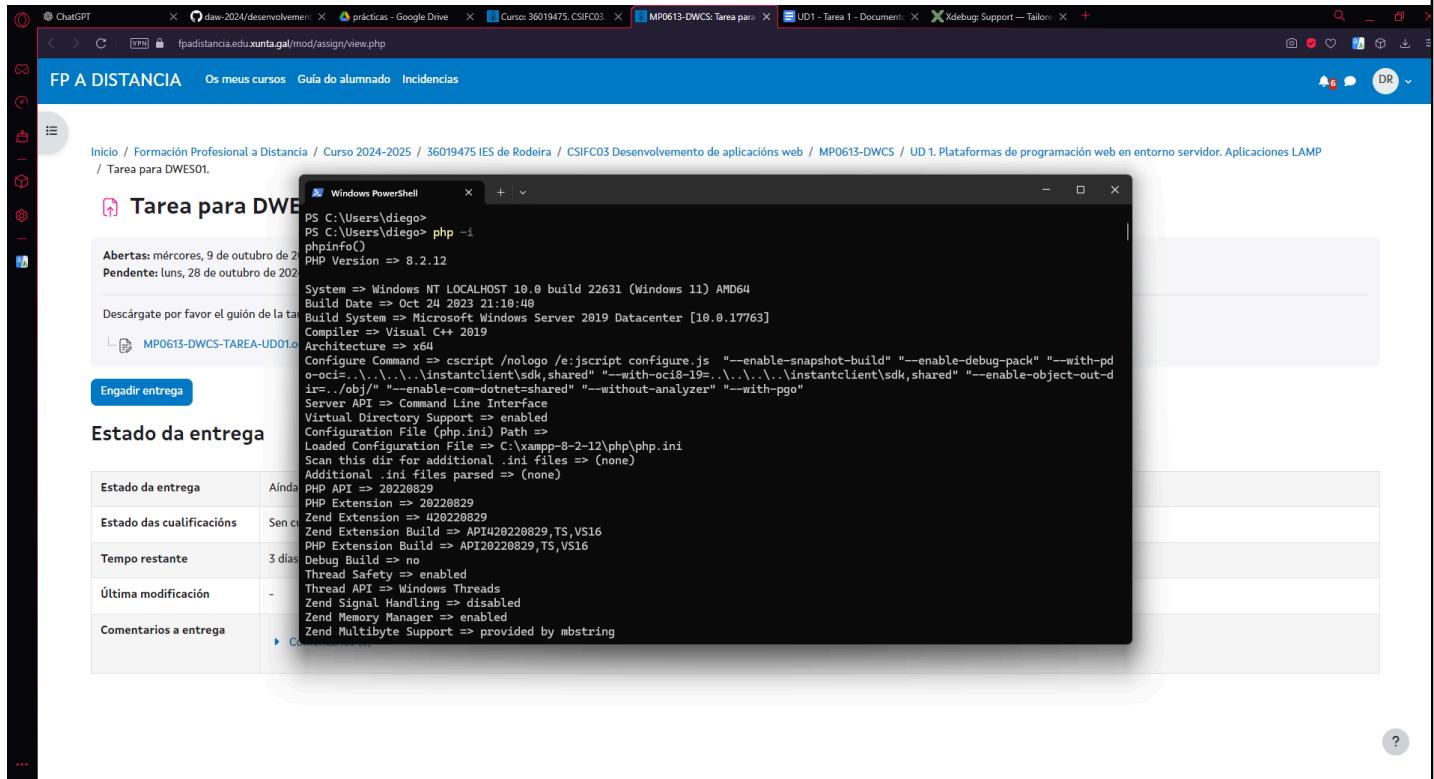
Enumera los pasos de configuración principales que seguirías y pon capturas de pantalla de los elementos de configuración esenciales.

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 (vale el usado en el vídeo de la explicación de la sección correspondiente).

RESPUESTA: (3 puntos)

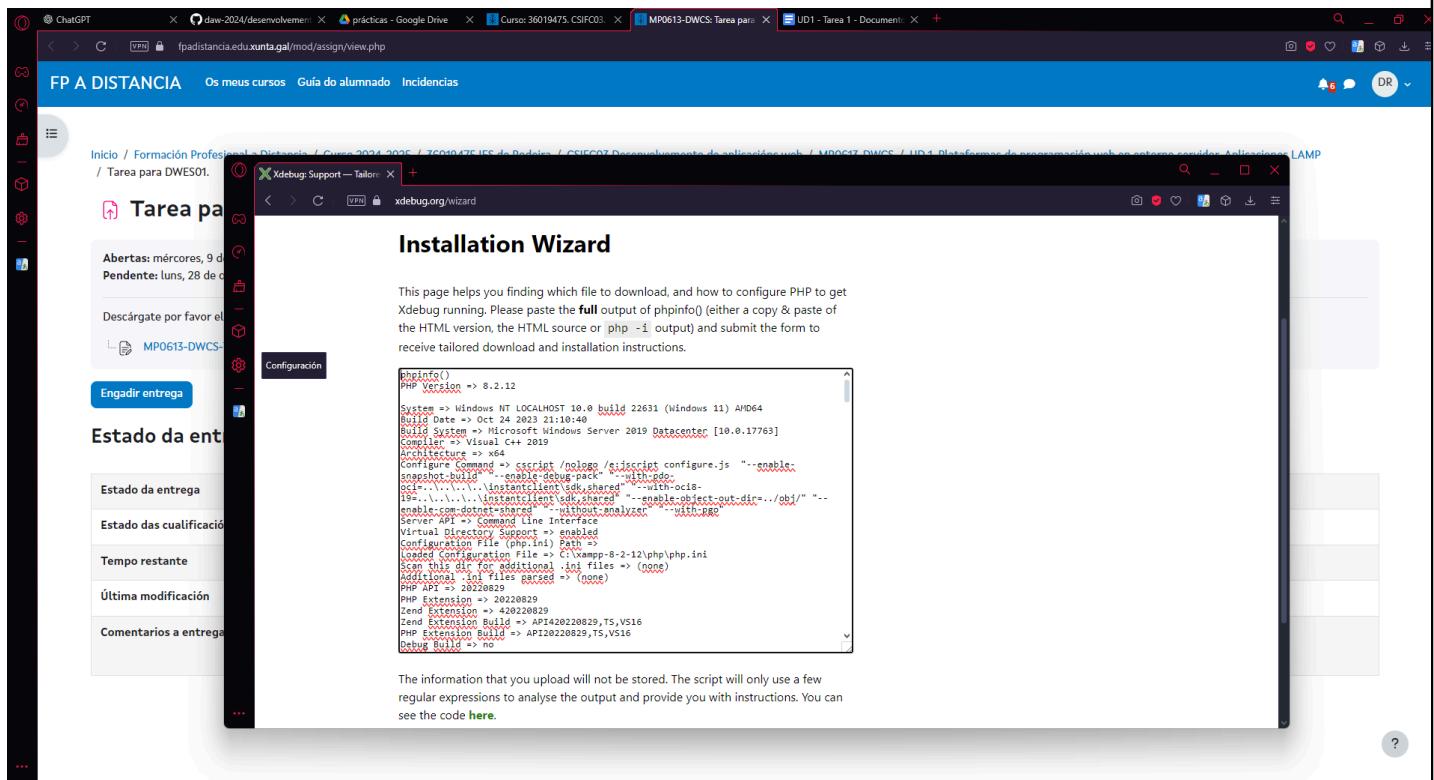
1. Descargar la .dll adecuada: <https://xdebug.org/docs/install>

- a. Crear el fichero *index.php* con *phpinfo()*. En mi caso ejecuto directamente *php -i* en la terminal y ya obtengo la información necesaria.



The screenshot shows a browser window with several tabs open. The main content area displays a task submission form for "MP0613-DWCS-TAREA-UD010". A PowerShell window is overlaid on the page, showing the output of the command "php -i". The output includes details about the PHP version (8.2.12), system (Windows NT LOCALHOST 10.0 build 22631 (Windows 11) AMD64), and various configuration settings like Zend extensions and PHP API.

- b. 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 a browser window with a modal dialog titled "Installation Wizard". This dialog is part of the Xdebug Support Tailor tool. It contains a text area with the output of the "phpinfo()" command, which is identical to the one shown in the previous screenshot. Below the text area, there is a note stating that the uploaded information will not be stored and that the script will only use regular expressions to analyze the output.

FP A DISTANCIA Os meus cursos Guía do alumnado Incidencias

[Inicio / Formación Profesional a Distancia / Curso 2024-2025 / 7C010475 IES de Rodeira / CFGS Desarrollo de aplicaciones web / MP0613-DWCS / UD1 - Plataforma de programación web en entornos servidos. Aplicaciones LAMP / Tarea para DWES01.](#)

[xdebug.org/wizard](#)

Tarea para DWES01

Summary

- 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

Instructions

1. Download [php_xdebug-3.3.2-8.2-vs16-x86_64.dll](#)
2. Move the downloaded file to C:\xampp-8-2-12\php\ext, and rename it to `php_xdebug.dll`
3. Update `C:\xampp-8-2-12\php\php.ini` and add the line:
`zend_extension = xdebug`

Enabling Features

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

FP A DISTANCIA Os meus cursos Guía do alumnado Incidencias

[Inicio / Formación Profesional a Distancia / Curso 2024-2025 / 7C010475 IES de Rodeira / CFGS Desarrollo de aplicaciones web / MP0613-DWCS / UD1 - Plataforma de programación web en entornos servidos. Aplicaciones LAMP / Tarea para DWES01.](#)

[xdebug.org/wizard](#)

Tarea para DWES01

Summary

Guardar como

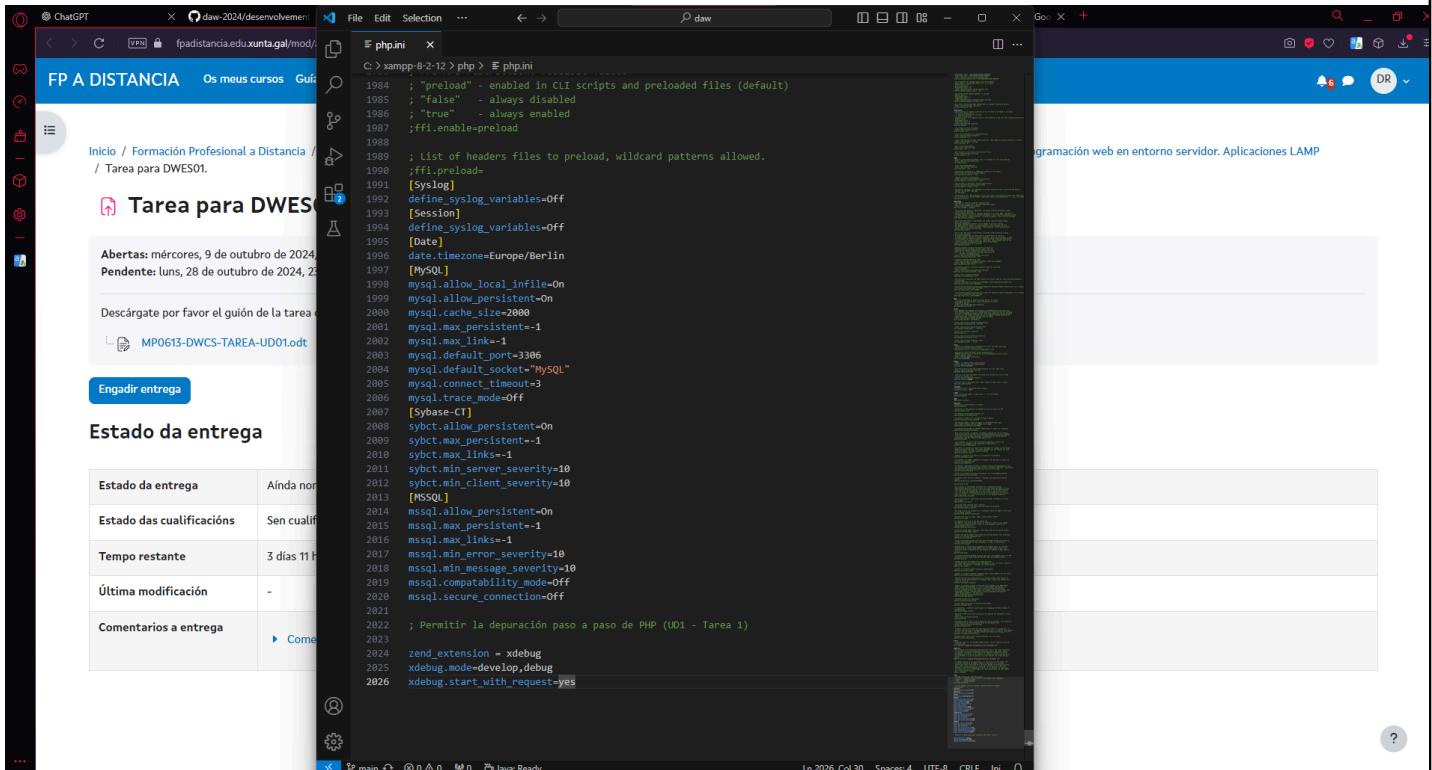
Nombre	Fecha de modificación	Tipo	Tamaño
php_bz2.dll	24/10/2023 23:56	Extensión de la aplicación	85 KB
php_com_dotnet.dll	24/10/2023 23:56	Extensión de la aplicación	91 KB
php_curl.dll	24/10/2023 23:56	Extensión de la aplicación	665 KB
php_dba.dll	24/10/2023 23:56	Extensión de la aplicación	152 KB
php_dl_test.dll	24/10/2023 23:56	Extensión de la aplicación	16 KB
php_enchant.dll	24/10/2023 23:56	Extensión de la aplicación	25 KB
php_exif.dll	24/10/2023 23:56	Extensión de la aplicación	71 KB
php_ffmpeg.dll	24/10/2023 23:56	Extensión de la aplicación	167 KB

Nombre: `php_xdebug.dll`
 Tipo: Todos los archivos

Guardar Cancelar

Enabling Features

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



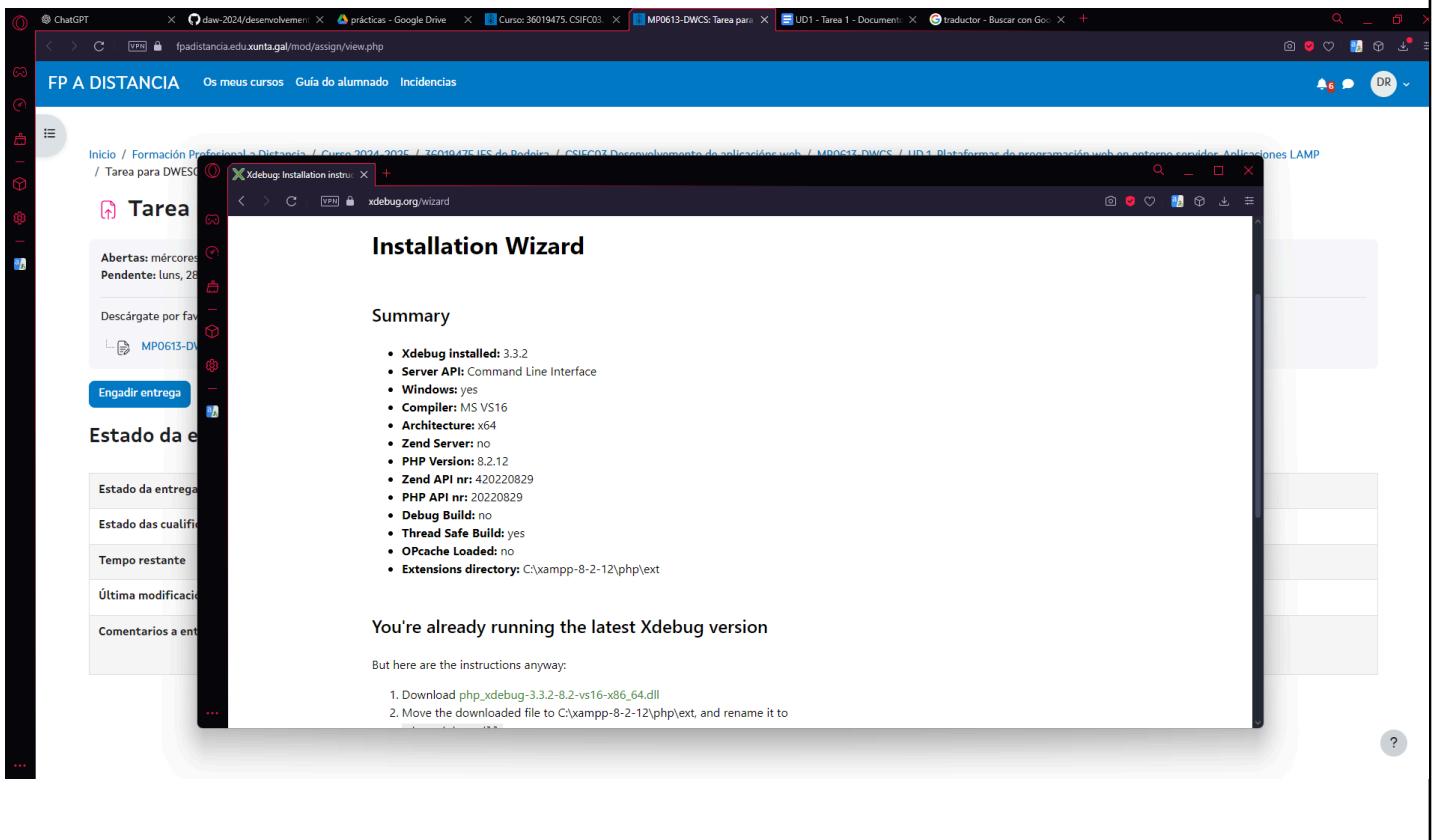
The screenshot shows a web-based file manager interface for a course assignment. On the left, there's a sidebar with course navigation and a file tree. The main area is a code editor with the file `php.ini` open. The code editor has syntax highlighting for PHP configuration. At the bottom of the code editor, there are status bars showing file statistics like 'In 2026, Col 30' and encoding 'UTF-8 CRLF'. The right side of the screen shows a preview of the assignment page, which includes a table for task status and a note about enabling Xdebug.

```

1984 ; "preload" - enabled in CLI scripts and preloaded files (default)
1985 ; "false" - always disabled
1986 ; "true" - always enabled
1987 ;ffl.enable=preload
1988
1989 ; List of headers files to preload, wildcard patterns allowed.
1990 ;[Syslog]
1991 define_syslog_variables=Off
1992 ;[Session]
1993 define_syslog_variables=Off
1994 ;[Date]
1995 date.timezone=Europe/Berlin
1996 ;[MySQL]
1997 mysql.allow_local_infile=On
1998 mysql.allow_persistent=On
1999 mysql.cache_size=2000
2000 mysql.max_persistent=-1
2001 mysql.max_links=-1
2002 mysql.default_port=3306
2003 mysql.default_socket=""MySQL"
2004 mysql.connect_timeout=3
2005 mysql.trace_mode=Off
2006 ;[Sybase-CT]
2007 sybct.allow_persistent=On
2008 sybct.max_persistent=-1
2009 sybct.max_links=-1
2010 sybct.min_server_severity=10
2011 sybct.min_client_severity=10
2012 ;[MSSQL]
2013 mssql.allow_persistent=On
2014 mssql.max_persistent=-1
2015 mssql.max_links=-1
2016 mssql.min_error_severity=10
2017 mssql.min_message_severity=10
2018 mssql.compatibility_mode=Off
2019 mssql.secure_connection=Off
2020
2021 zend_extension = xdebug
2022 xdebug.mode=develop,debug
2023 xdebug.start_with_request=yes
2024
2025
2026

```

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



The screenshot shows a web-based file manager interface for a course assignment. On the left, there's a sidebar with course navigation and a file tree. The main area is a modal window titled 'Xdebug: Installation instructions' from 'xdebug.org/wizard'. The window contains the 'Installation Wizard' summary, which lists various Xdebug installation details. It also includes a note stating 'You're already running the latest Xdebug version' and provides instructions for manual download if needed.

Summary

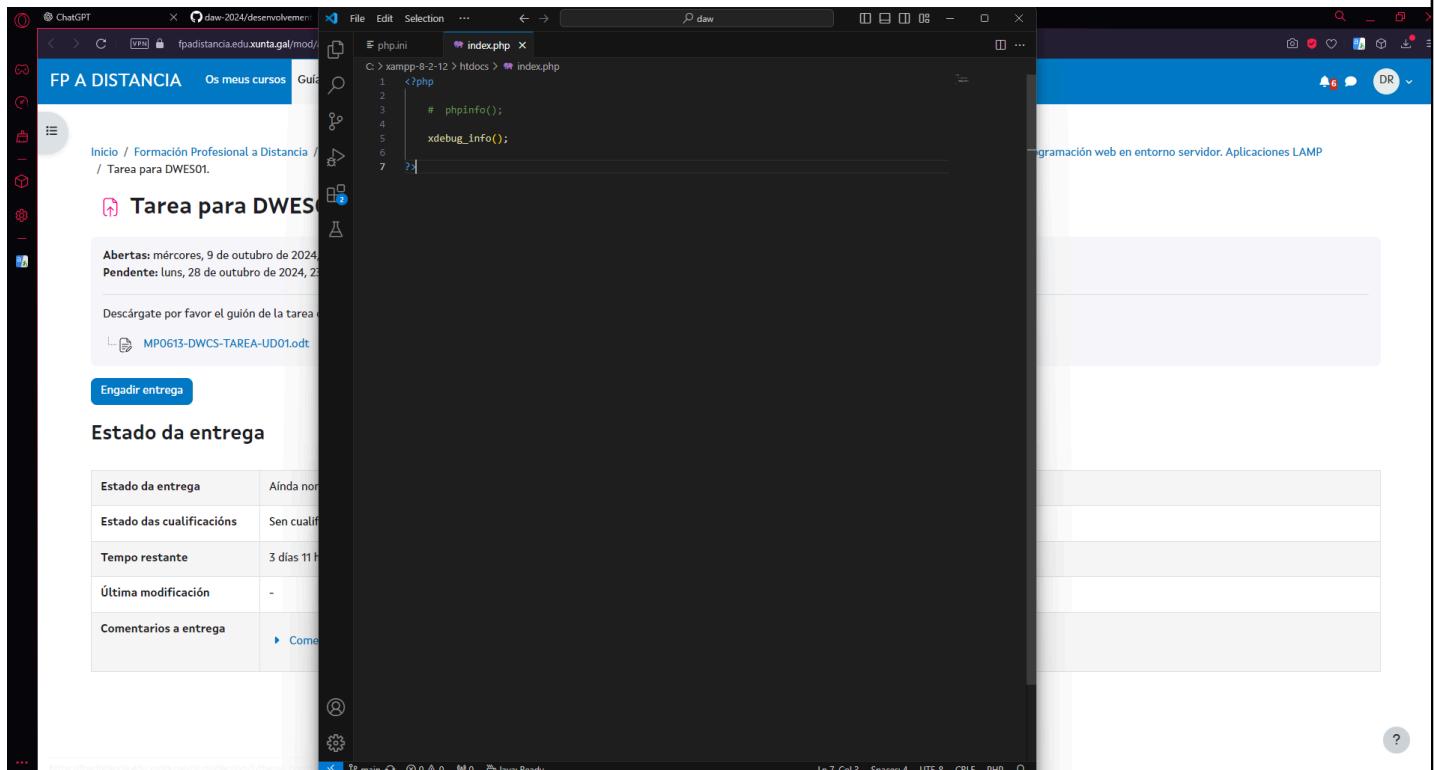
- Xdebug installed: 3.3.2
- Server API: Command Line Interface
- Windows: yes
- Compiler: MS VS16
- Architecture: x64
- Zend Server: no
- PHP Version: 8.2.12
- Zend API nr: 420220829
- PHP API nr: 20220829
- Debug Build: no
- Thread Safe Build: yes
- OPcache Loaded: no
- Extensions directory: C:\xampp-8-2-12\php\ext

You're already running the latest Xdebug version

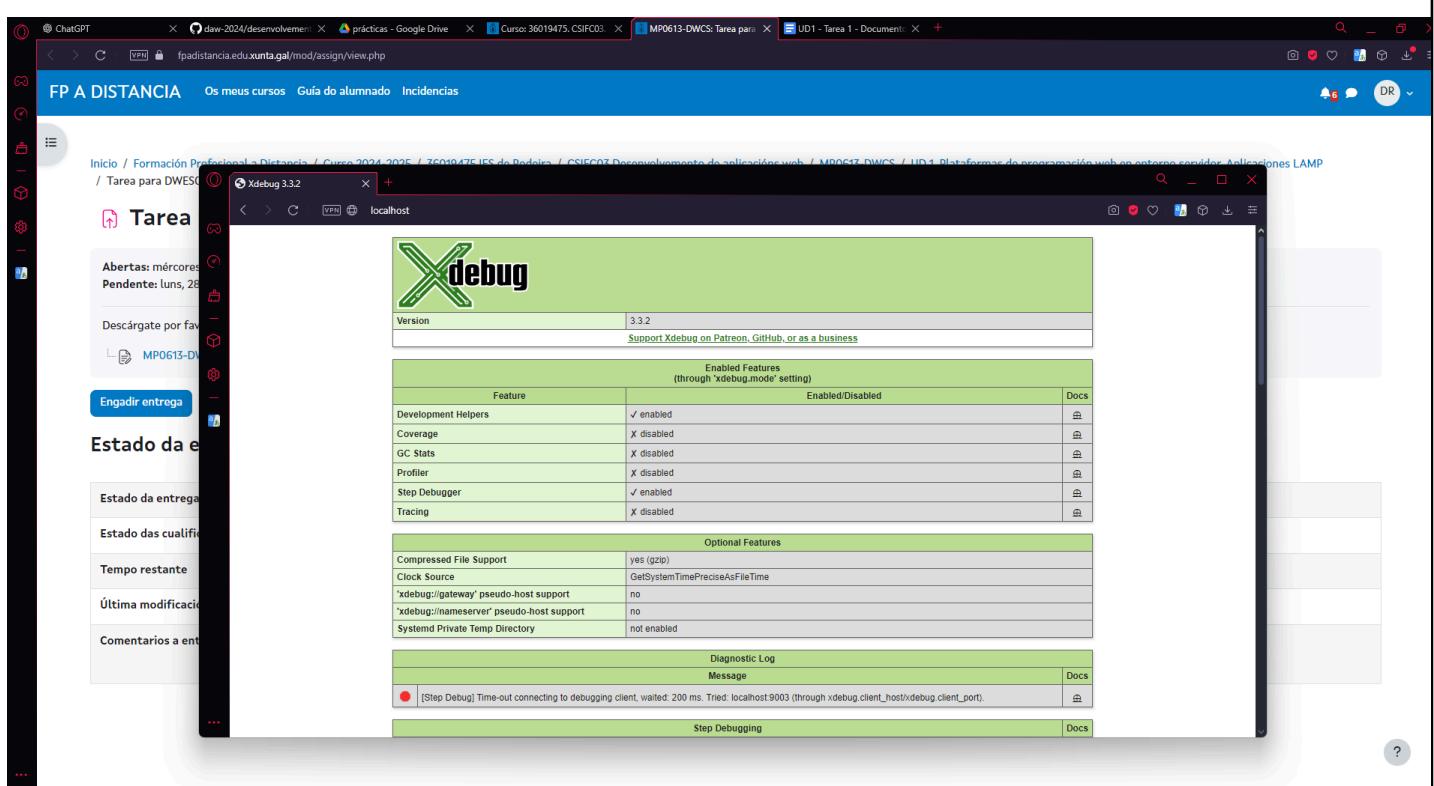
But here are the instructions anyway:

- Download [php_xdebug-3.3.2-8.2-vs16-x86_64.dll](#)
- Move the downloaded file to C:\xampp-8-2-12\php\ext, and rename it to `xdebug.dll`

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



```
C:\ > xampp-8-2-12 > htdocs > index.php
1 <?php
2
3 # phpinfo();
4
5 xdebug_info();
6
7 ?>
```



The terminal output shows the Xdebug configuration:

```
Version: 3.3.2
Support Xdebug on Patreon, GitHub, or as a business

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

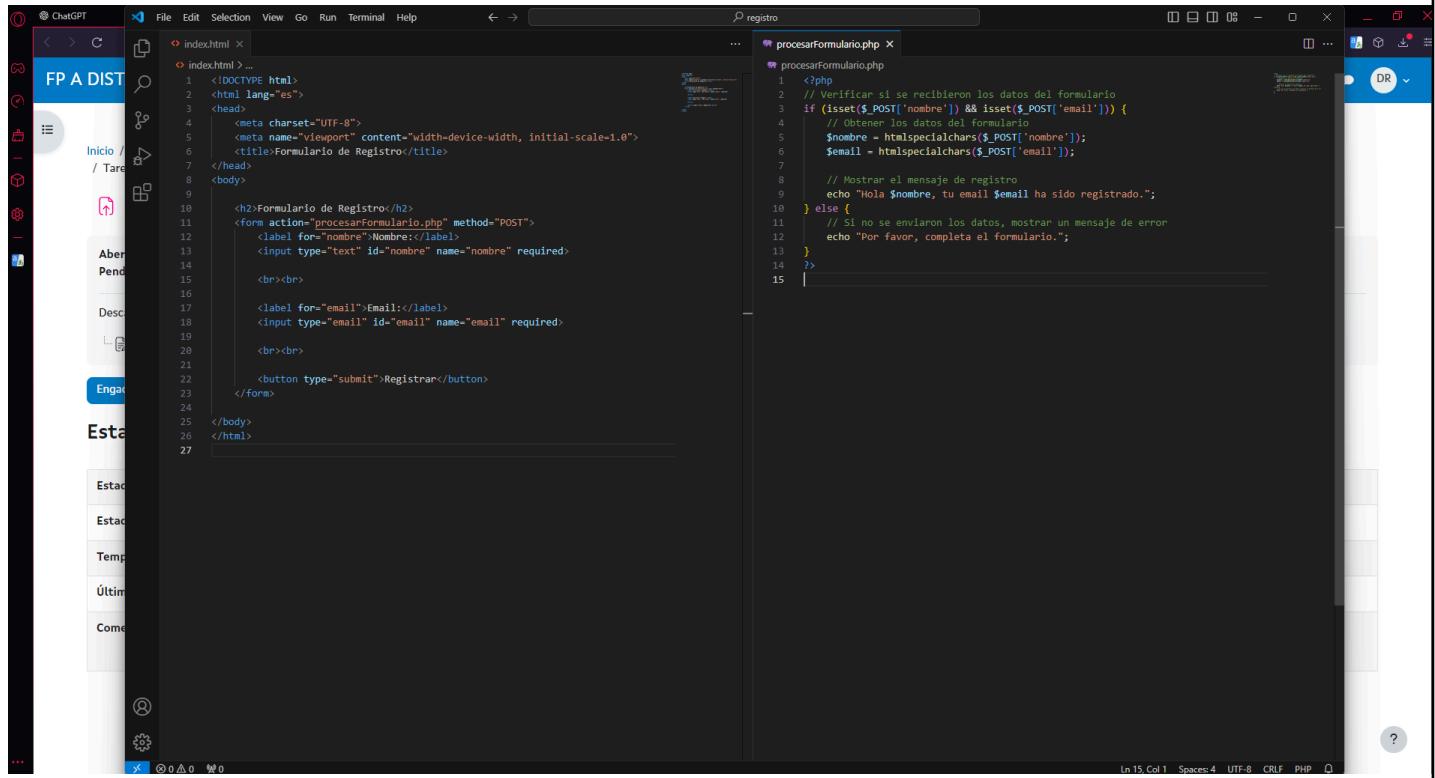
Optional Features
Compressed File Support yes (gzip)
Clock Source GetSystemTimePreciseAsFileTime
'xdebug://gateway' pseudo-host support no
'xdebug://nameserver' pseudo-host support no
Systemd Private Temp Directory not enabled

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
Docs
```

4. Comprobar desde Visual Studio Code.

Creo una aplicación sencilla en PHP.



```

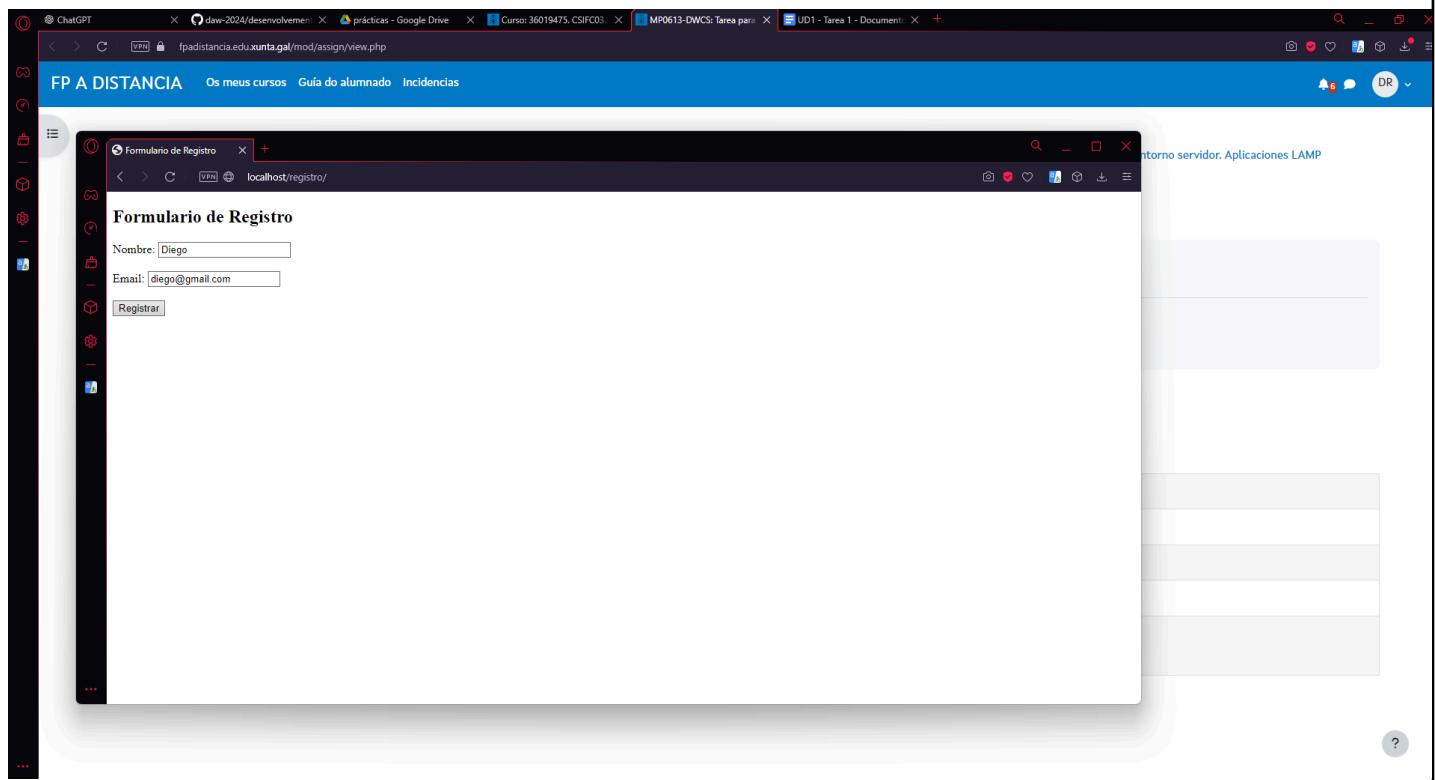
index.html
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Formulario de Registro</title>
7   </head>
8   <body>
9
10
11   <h2>Formulario de Registro</h2>
12   <form action="procesarFormulario.php" method="POST">
13     <label for="nombre">Nombre:</label>
14     <input type="text" id="nombre" name="nombre" required>
15
16     <br><br>
17
18     <label for="email">Email:</label>
19     <input type="email" id="email" name="email" required>
20
21     <br><br>
22
23     <button type="submit">Registrar</button>
24
25   </body>
26 </html>
27

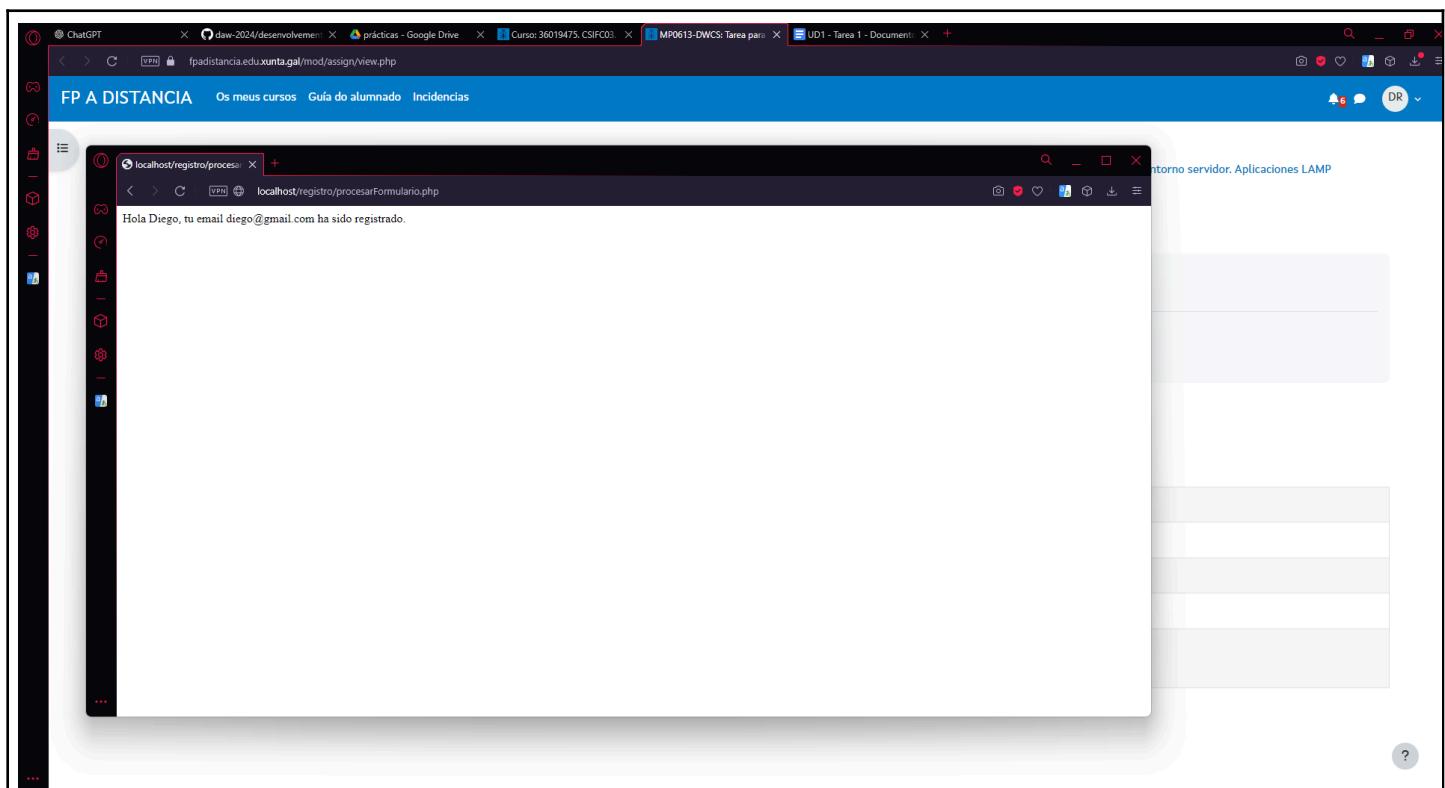
```

```

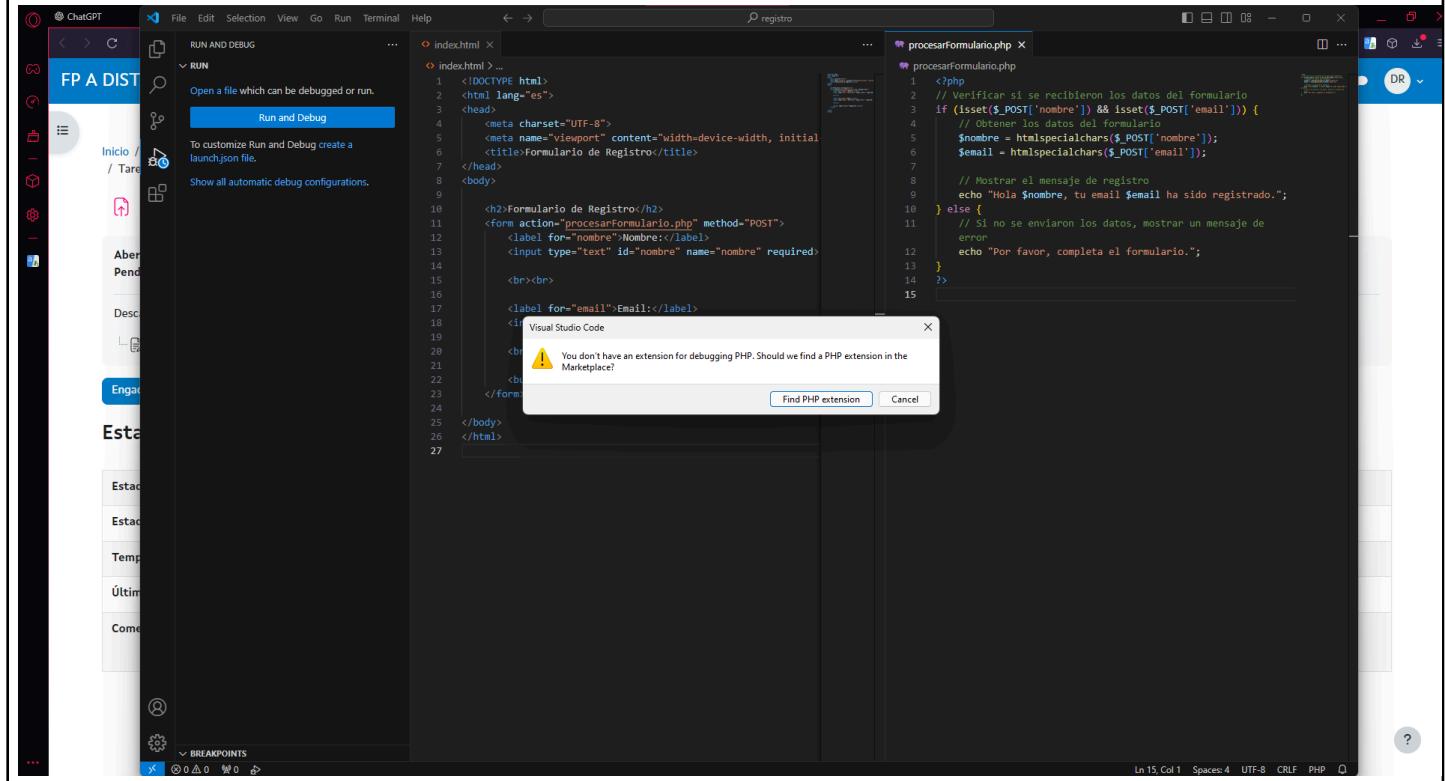
procesarFormulario.php
1 <?php
2   // Verificar si se recibieron los datos del formulario
3   if (isset($_POST['nombre']) && isset($_POST['email'])) {
4     // Obtener los datos del formulario
5     $nombre = htmlspecialchars($_POST['nombre']);
6     $email = htmlspecialchars($_POST['email']);
7
8     // Mostrar el mensaje de registro
9     echo "Hola $nombre, tu email $email ha sido registrado.";
10  } else {
11    // Si no se enviaron los datos, mostrar un mensaje de error
12    echo "Por favor, completa el formulario.";
13  }
14
15

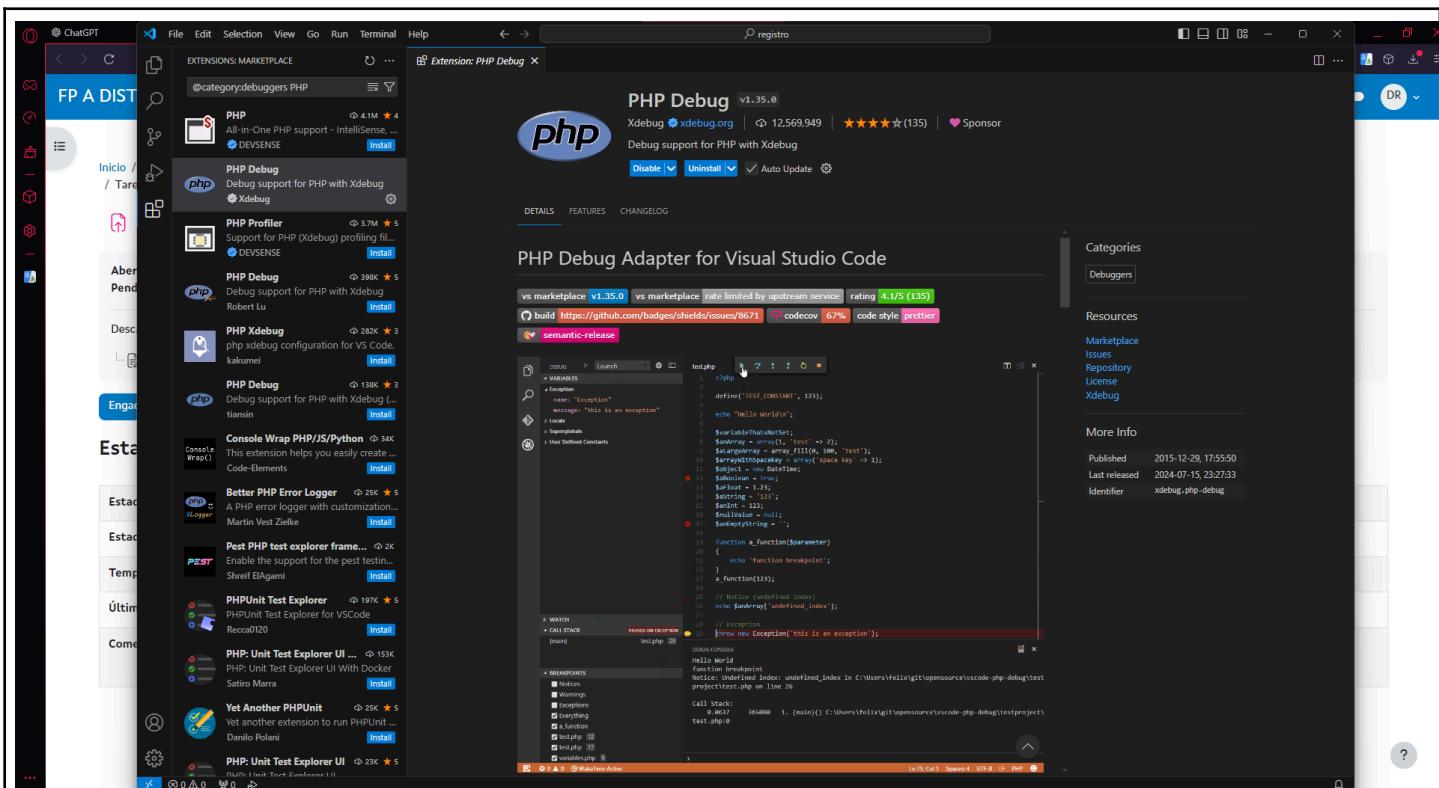
```



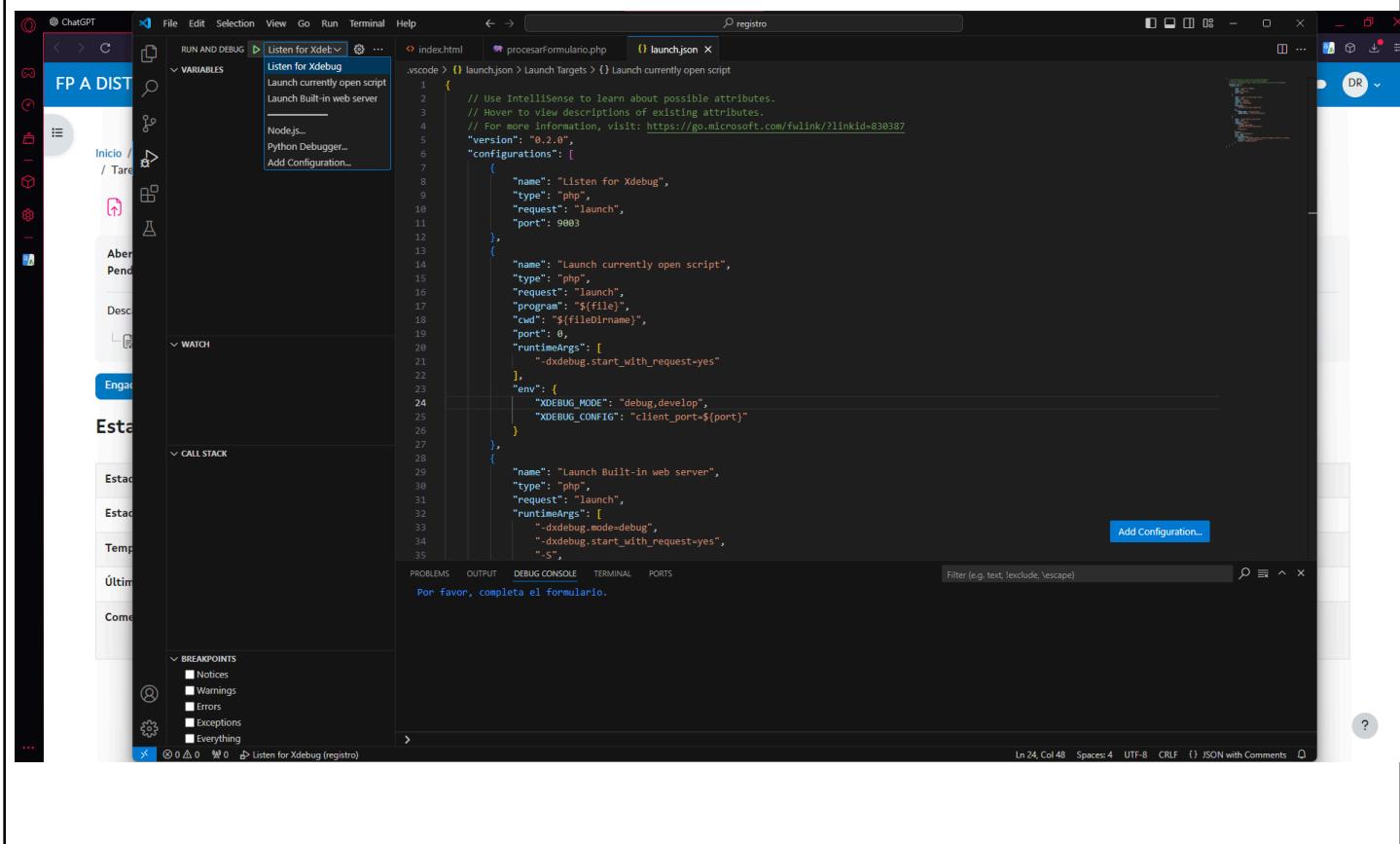


Instalamos la extensión XDebug para PHP.

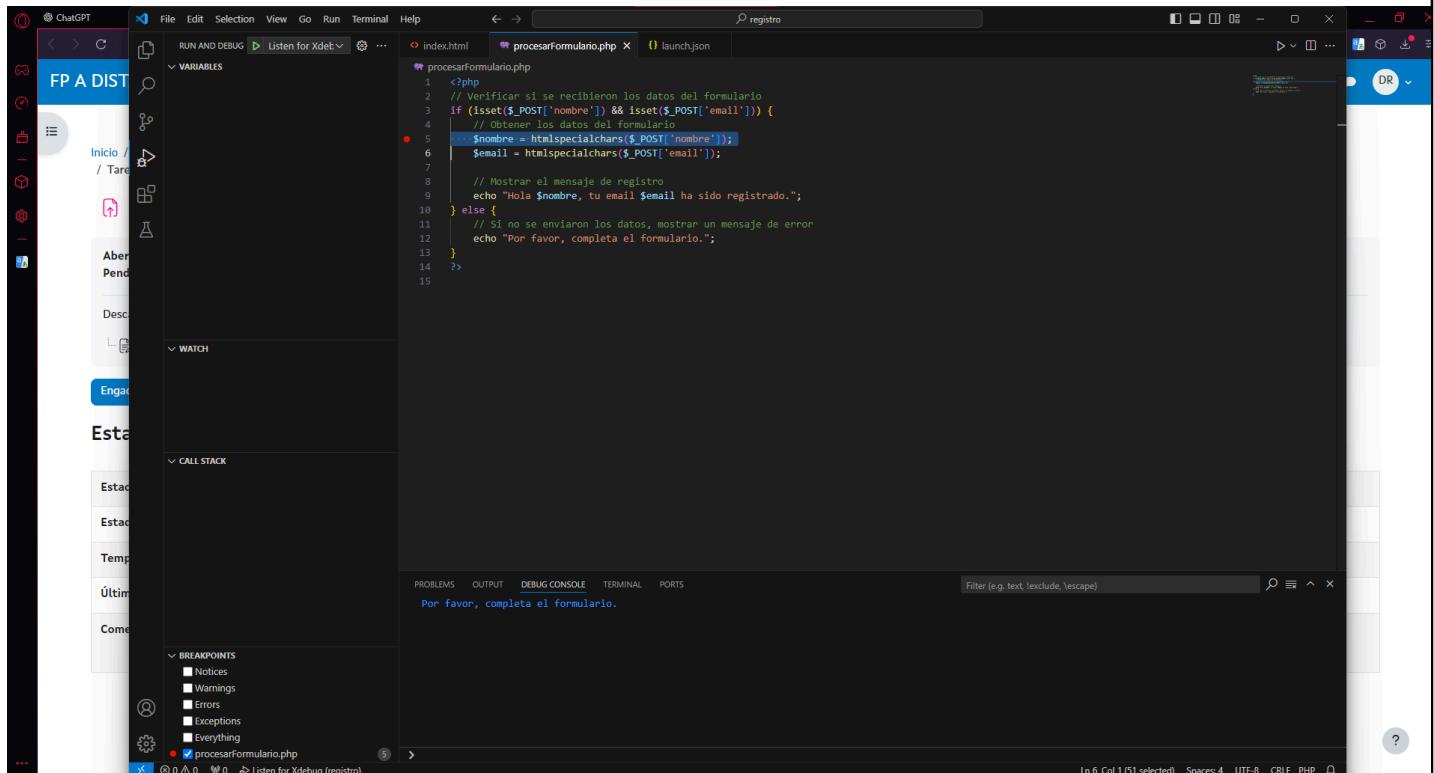




Crear el archivo *launch.json* para añadir los modos de debug por defecto. Se genera de forma automática pulsando en el enlace que aparece tras instalar la extensión. Ahora ya podemos debuggear la aplicación PHP en los siguientes modos:



Añadir un breakpoint para comprobar que funciona (se detiene la ejecución en ese punto).



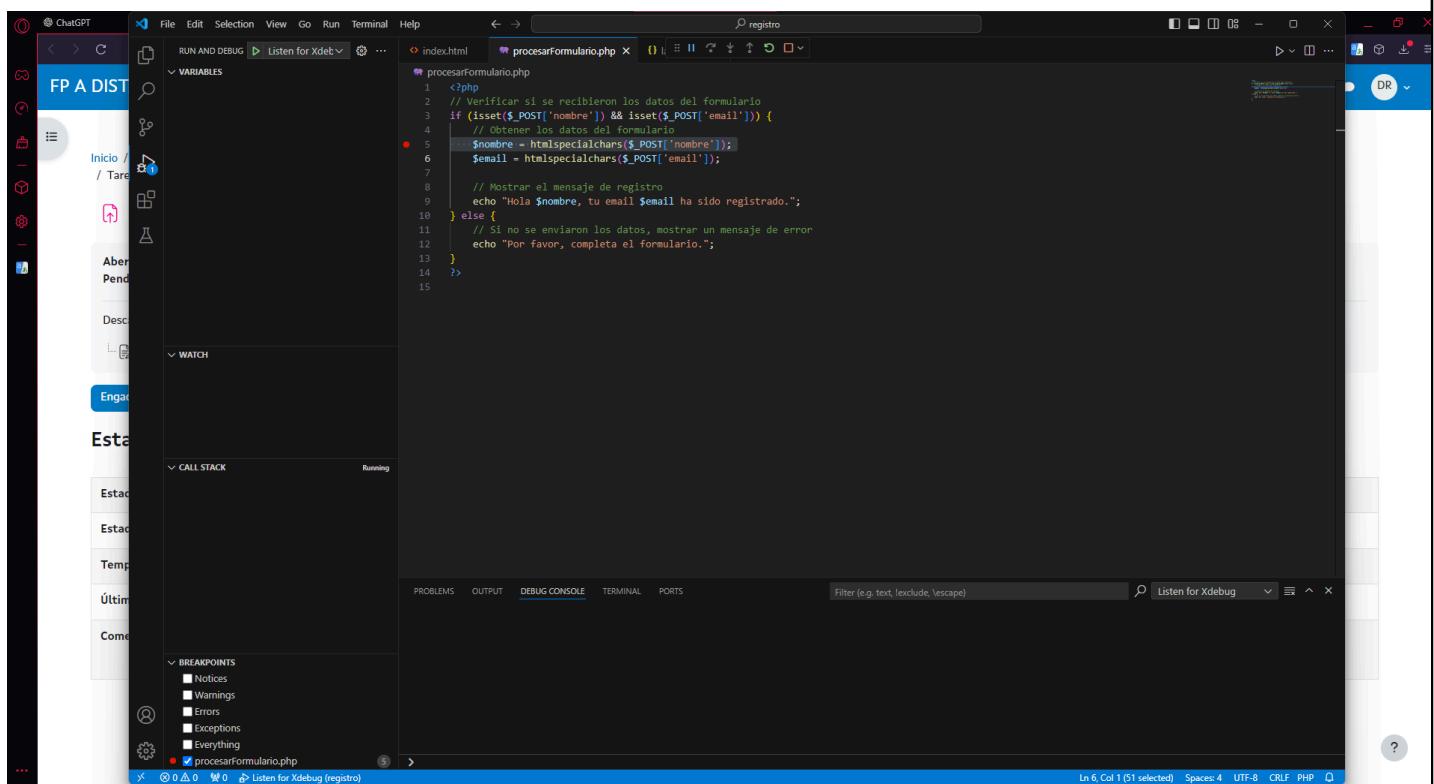
```

<?php
// Verificar si se recibieron los datos del formulario
if (isset($_POST['nombre']) && isset($_POST['email'])) {
    // Obtener los datos del formulario
    $nombre = htmlspecialchars($_POST['nombre']);
    $email = htmlspecialchars($_POST['email']);

    // Mostrar el mensaje de registro
    echo "Hola $nombre, tu email $email ha sido registrado.";
} else {
    // Si no se enviaron los datos, mostrar un mensaje de error
    echo "Por favor, completa el formulario.";
}
?>

```

Iniciar el modo debug haciendo clic en el siguiente botón (*Play*).



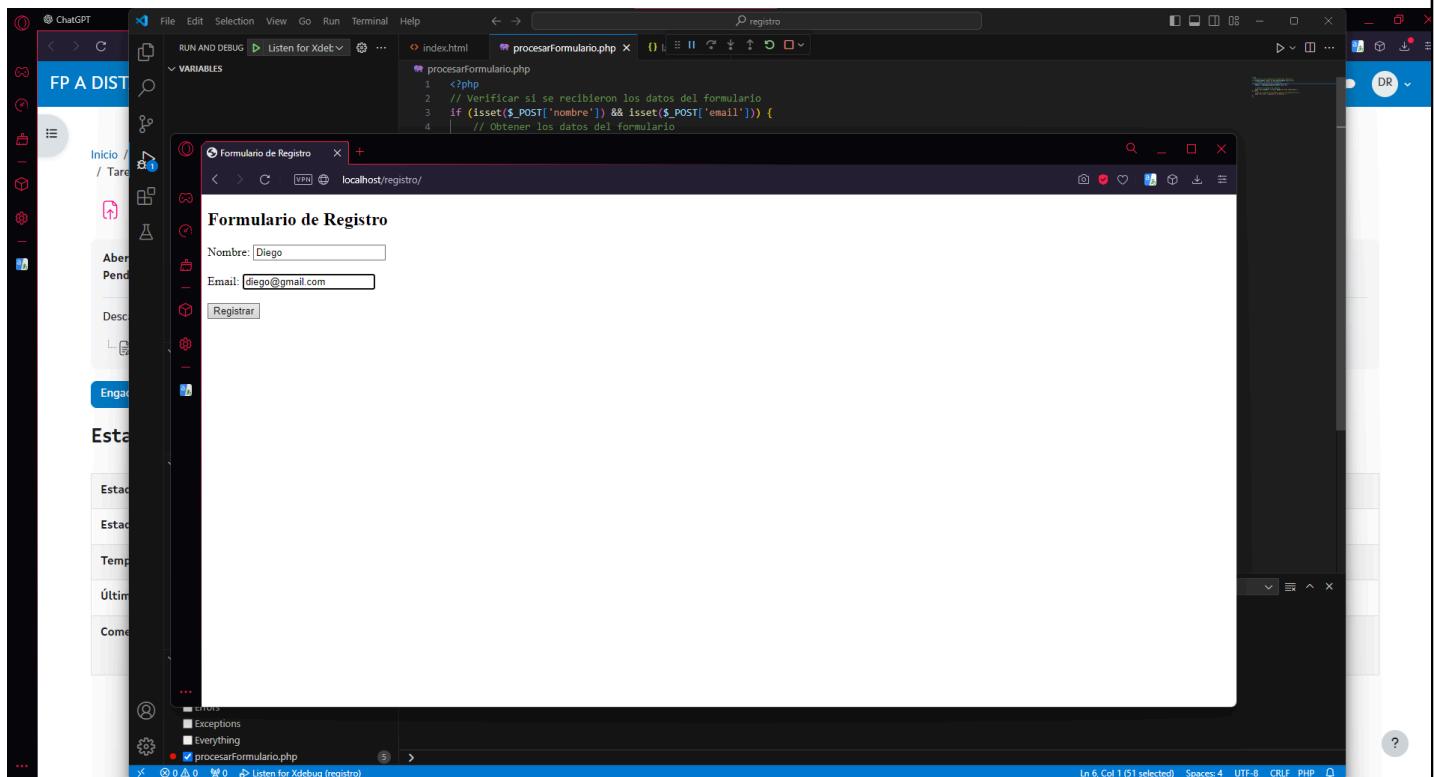
```

<?php
// Verificar si se recibieron los datos del formulario
if (isset($_POST['nombre']) && isset($_POST['email'])) {
    // Obtener los datos del formulario
    $nombre = htmlspecialchars($_POST['nombre']);
    $email = htmlspecialchars($_POST['email']);

    // Mostrar el mensaje de registro
    echo "Hola $nombre, tu email $email ha sido registrado.";
} else {
    // Si no se enviaron los datos, mostrar un mensaje de error
    echo "Por favor, completa el formulario.";
}
?>

```

Relleno el formulario nuevamente.

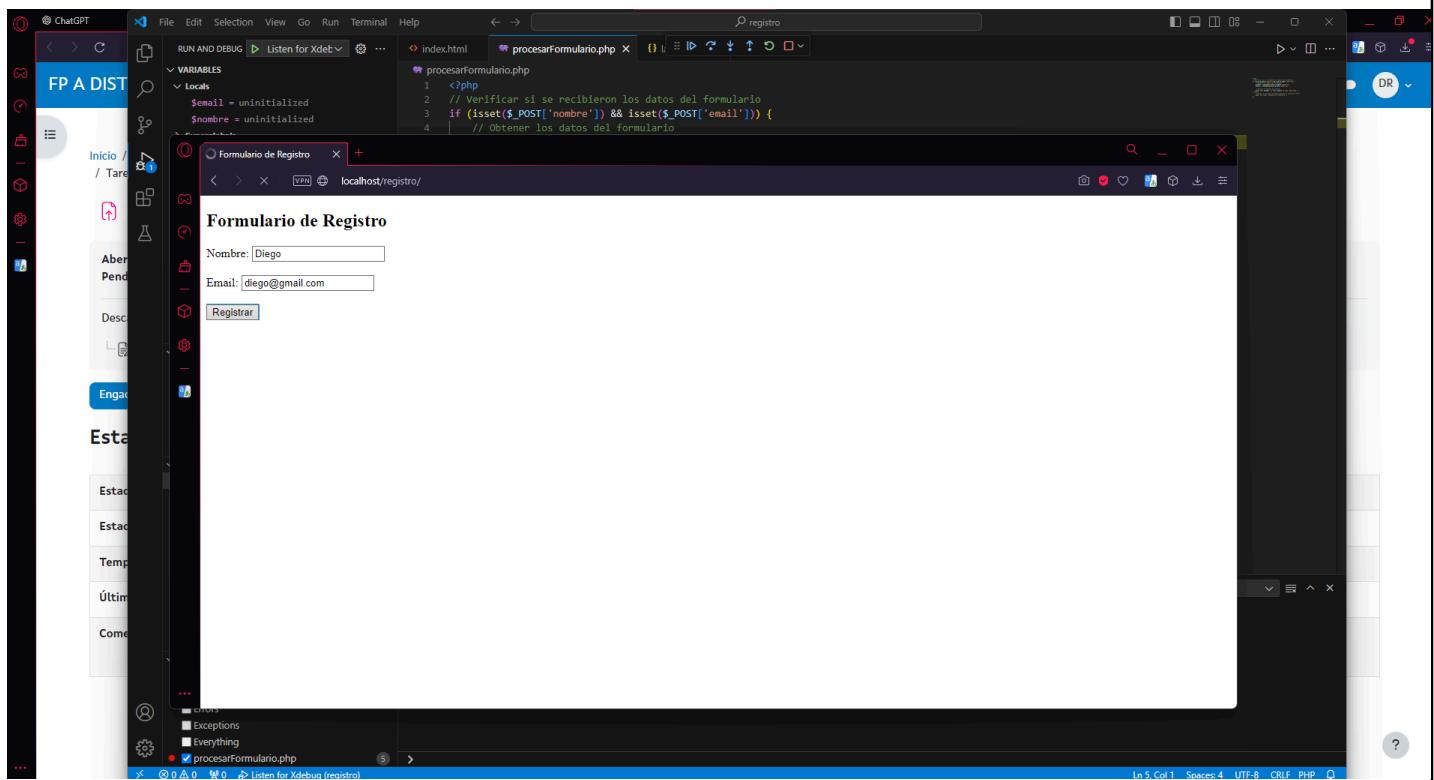


The screenshot shows a Visual Studio Code (VS Code) interface. On the left is the sidebar with various icons and sections like 'FP A DIST', 'Abre', 'Pend', 'Desc', 'Enga', 'Estad', 'Estad', 'Temp', 'Últim', 'Comen'. The main area has tabs for 'index.html' and 'procesarFormulario.php'. The code editor shows the following PHP code:

```
<?php
// Verificar si se recibieron los datos del formulario
if (!isset($_POST['nombre']) && !isset($_POST['email'])) {
    // Obtener los datos del formulario
}
```

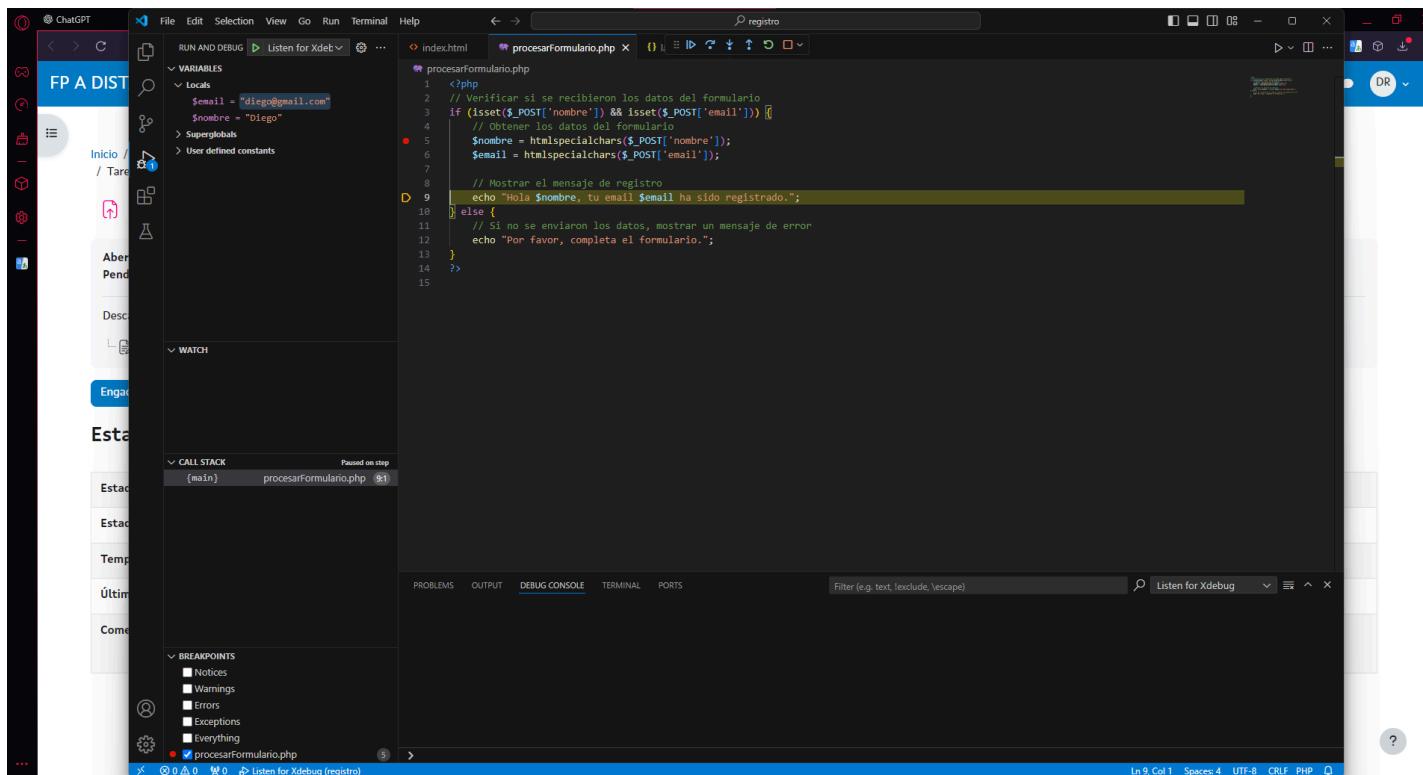
Below the code editor is a browser window titled 'Formulario de Registro' showing a simple form with fields for 'Nombre' (Diego) and 'Email' (diego@gmail.com), and a 'Registrar' button. The status bar at the bottom indicates 'Ln 6, Col 1 (51 selected) Spaces: 4 UTF-8 CRLF PHP'.

Al enviar los datos se detiene en el el breakpoint, por eso las variables aparecen sin valor en el debug de VSC.



This screenshot is identical to the one above, but the 'Variables' section in the sidebar is expanded, showing 'Locals' with '\$email = uninitialized' and '\$nombre = uninitialized'. The rest of the interface and code are the same.

Le doy a avanzar, la ejecución continua y las variables toman su valor.



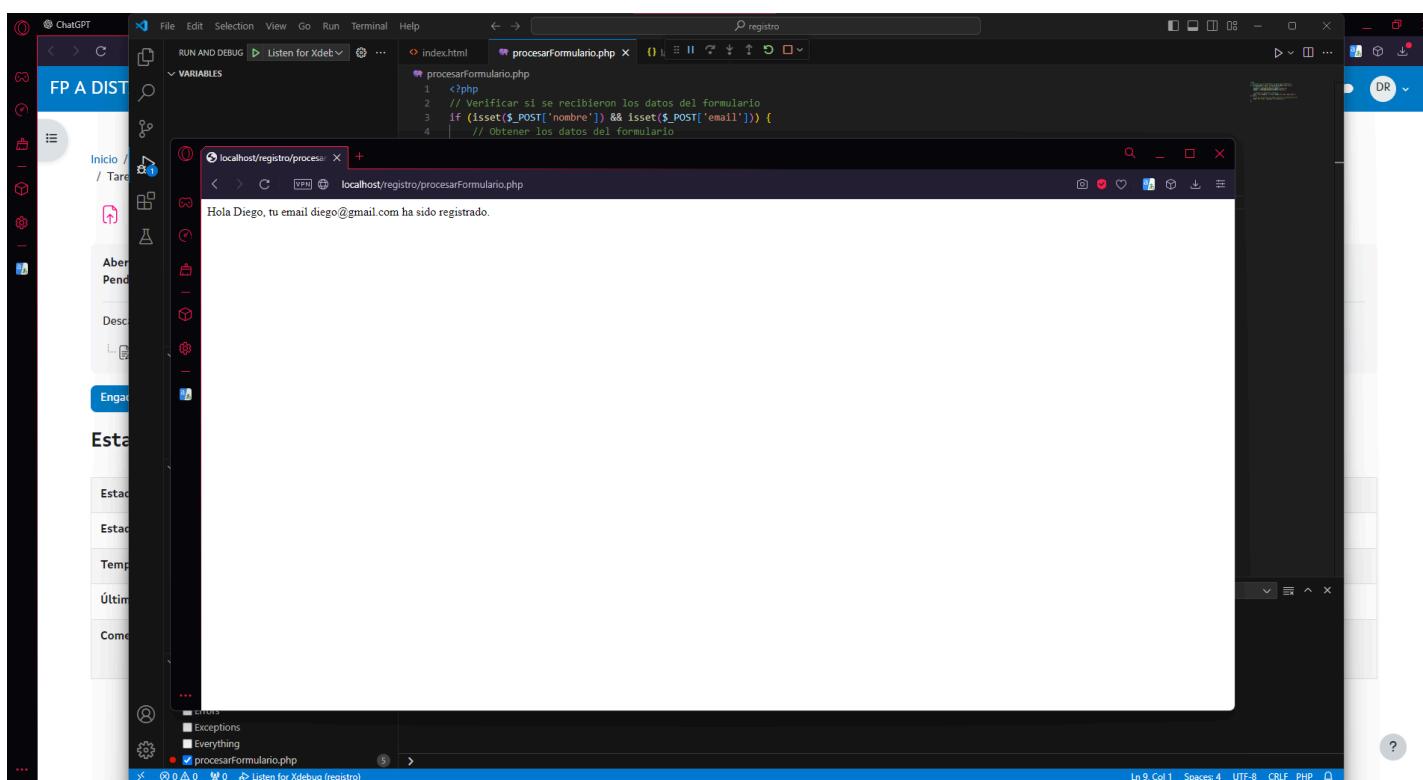
```

<?php
// Verificar si se recibieron los datos del formulario
if (!isset($_POST['nombre']) && !isset($_POST['email'])) {
    // Obtener los datos del formulario
    $Nombre = htmlspecialchars($_POST['nombre']);
    $email = htmlspecialchars($_POST['email']);

    // Mostrar el mensaje de registro
    echo "Hola $Nombre, tu email $email ha sido registrado.";
} else {
    // Si no se enviaron los datos, mostrar un mensaje de error
    echo "Por favor, completa el formulario.";
}
?>

```

Los datos se procesan y se muestra el resultado de la ejecución.



Hola Diego, tu email diego@gmail.com ha sido registrado.