

Introducció a Docker

# Part 07. Cas pràctic 01 - Desenvolupant amb Visual Studio Code en un contenidor

---



Autor: Sergi García Barea

Actualitzat setembre 2021

## Llicència




**Reconeixement - No comercial - CompartirIgual**(BY-NC-SA): No es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original.

## Nomenclatura

Al llarg d'aquest tema s'utilitzaran diferents símbols per distingir elements importants dins de l'contingut. Aquests símbols són:

 **Important**

 **atenció**

 **Interessant**

<b>Introducció</b>	<b>3</b>
<b>Visual Studio Code i plugins per Docker</b>	<b>3</b>
<b>Creant i editant aplicació en PHP dins d'un contenidor</b>	<b>3</b>
<b>Bibliografia</b>	<b>6</b>

## UD07. CAS PRÀCTIC 01

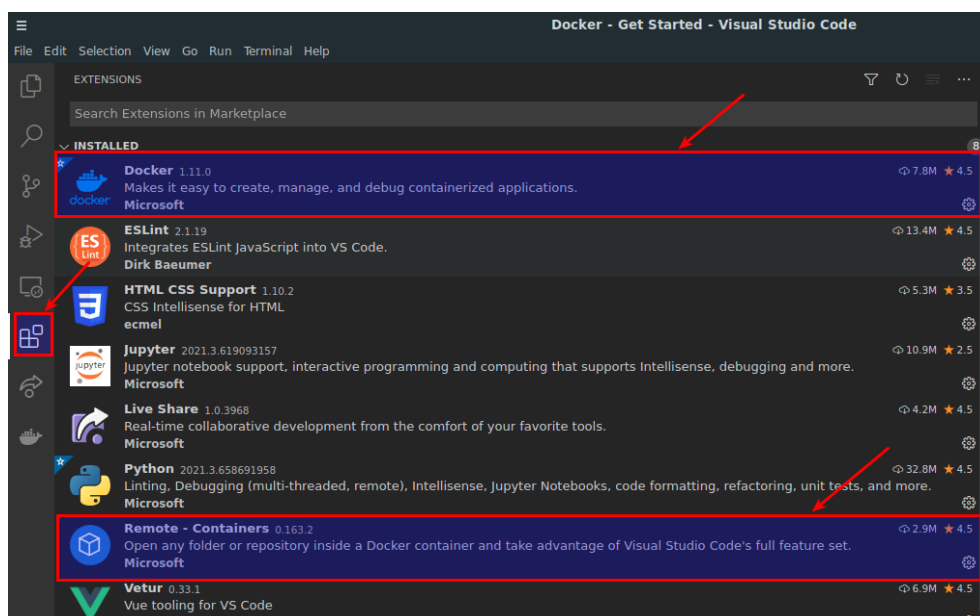
## 1. INTRODUCCIÓ

En aquest cas pràctic anem utilitzar l'editor multiplataforma "Visual **Studio Code**" juntament amb connectors relacionats amb Docker per poder desenvolupar una senzilla aplicació web en un contenidor.

## 2. VISUAL STUDIO CODE I PLUGINS PER DOCKER

Podem descarregar l'editor "Visual Studio Code" en <https://code.visualstudio.com/> i en si no el coneixem i volem saber més podem accedir a <https://code.visualstudio.com/learn>.

En instal·lar-ho, si "Visual Studio Code" detecta Docker instal·lat en el sistema, el mateix editor ens suggerirà una sèrie de plugins. Aquests plugins són els que es poden veure a la imatge:



Els plugins recomanats són:

<https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-docker>

[https://marketplace.visualstudio.com/items?](https://marketplace.visualstudio.com/items?itemName=ms-vscode.remote.remote-containers)

[itemName = ms-vscode.remote.remote-containers](https://marketplace.visualstudio.com/items?itemName=ms-vscode.remote.remote-containers)

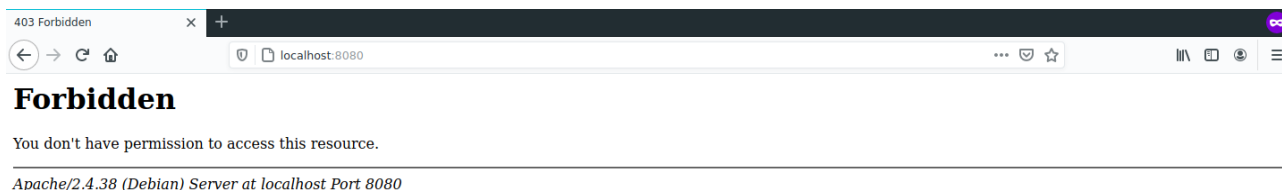
Podem obtenir molta informació de com treballar amb contenidors amb Visual Studio Code en <https://code.visualstudio.com/docs/containers/overview>

## 3. CREANT I EDITANT APLICACIÓ EN PHP DINS D'UN CONTENIDOR

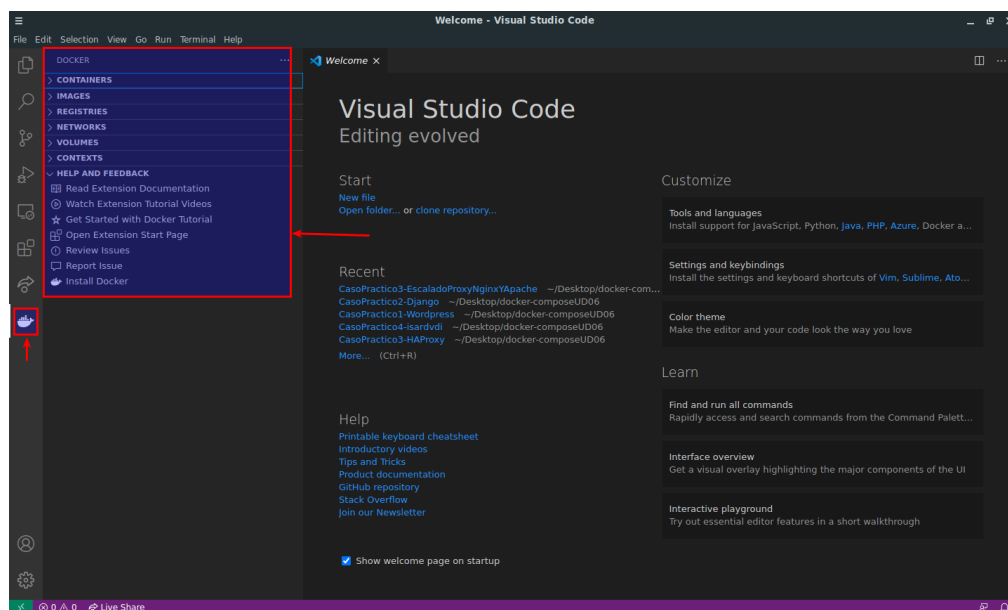
Amb el següent comandament, podem crear un senzill contenidor que es servirà pel port 8080. El contenidor contindrà "**Apache+PHP**" i en el directori **"/var/www/html"** s'emmagatzema el seu lloc web.

```
docker run -d --name servidordesarrollo -p 8080: 80 php: 7.2-apatxe
```

Una vegada creat el nostre contenidor provarem d'accedir a <http://localhost:8080> i observarem alguna cosa similar a:

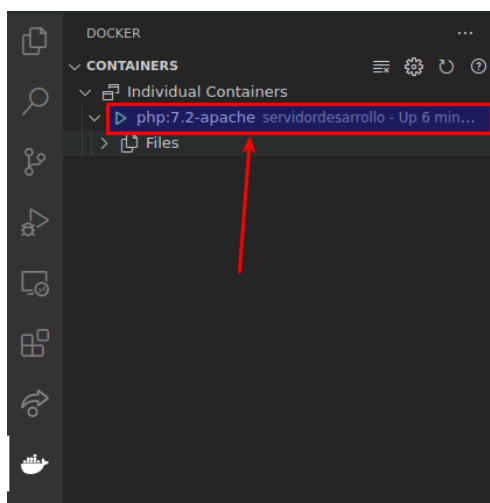


Després d'això, obrirem l'editor Visual Studio Code i accedirem a les utilitats Docker (icona de la balena) i accedirem a un menú amb diferents (opcions) tal com es veu a la imatge:



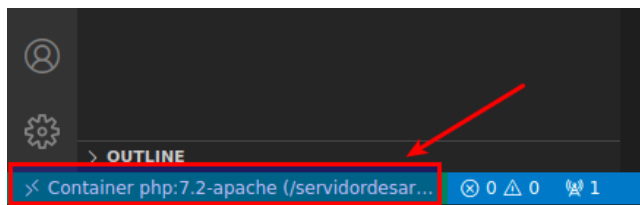
Aquí hi ha diverses pestanyes. Algunes de les més destacades són **"Containers"** (informació dels contenidors), **"Images"** (informació de les imatges disponibles), **"Networks"** (amb informació de les xarxes entre contenidors) o **"Volumes"** (amb informació dels volums creats).

Seguint el cas pràctic, si despleguem la pestanya de menú **"Containers"**, veurem tots els contenidors i podrem operar sobre ells. A la següent imatge, veiem el contenidor que hem creat anteriorment en funcionament.



Utilitzant el botó dret, podrem aturar/iniciar el contenidor, obtenir més informació, etc. També podem observar informació, simplement deixant el cursor a sobre del contenidor.

Per prosseguir amb el nostre cas pràctic, premerem el botó dret sobre el contenidor i seleccionarem l'opció **"Attach Visual Studio Code"**. Això obrirà una nova instància de Visual Studio Code connectada a aquest contenidor. Podem comprovar que és correcte observant a la part inferior de la pantalla de la nova instància alguna cosa similar a:

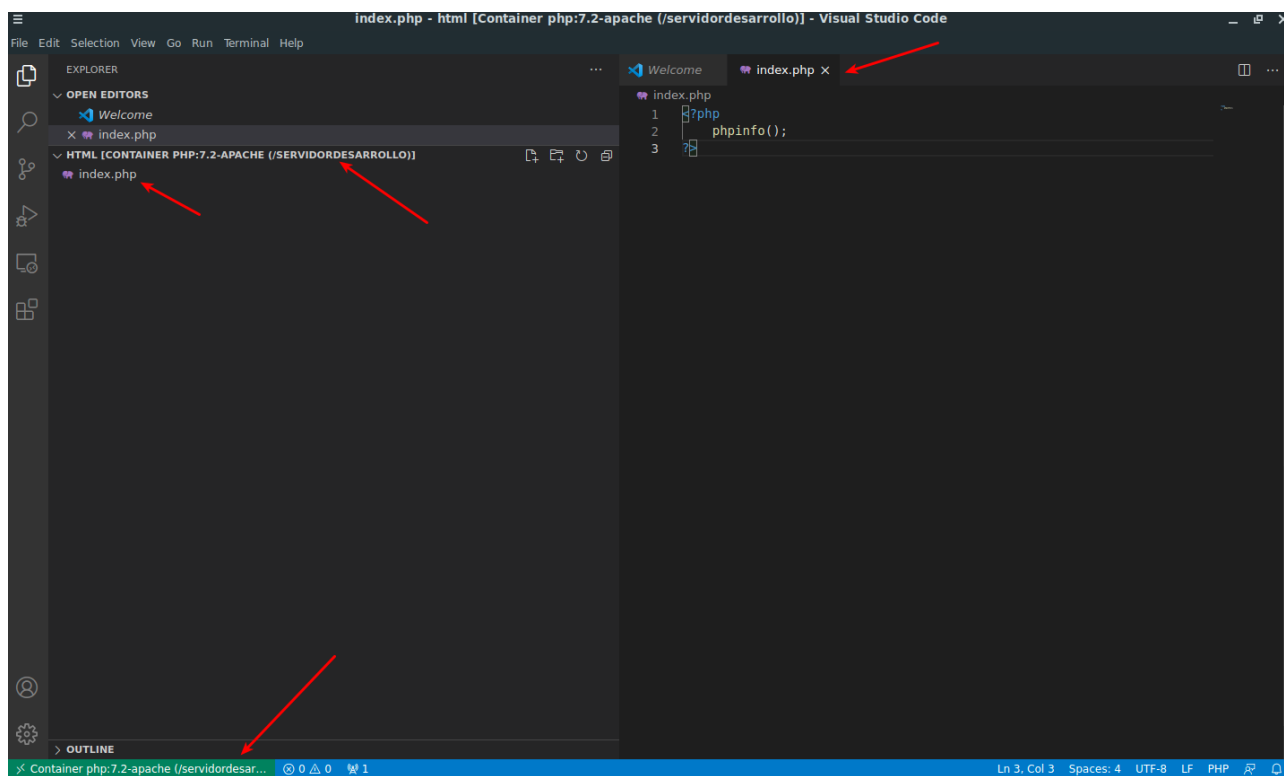


**! Atenció:** aquest pas utilitza internament "remote container". Podríem haver-ho fet manualment havent premut la icona de baix a l'esquerra i havent seleccionat a mà el contenidor.

Una vegada en aquesta instància, podem obrir el directori **"/var/www/html"** i ho editarem com si estiguérem en local. Ací crearem un senzill fitxer **"index.php"** amb el següent contingut

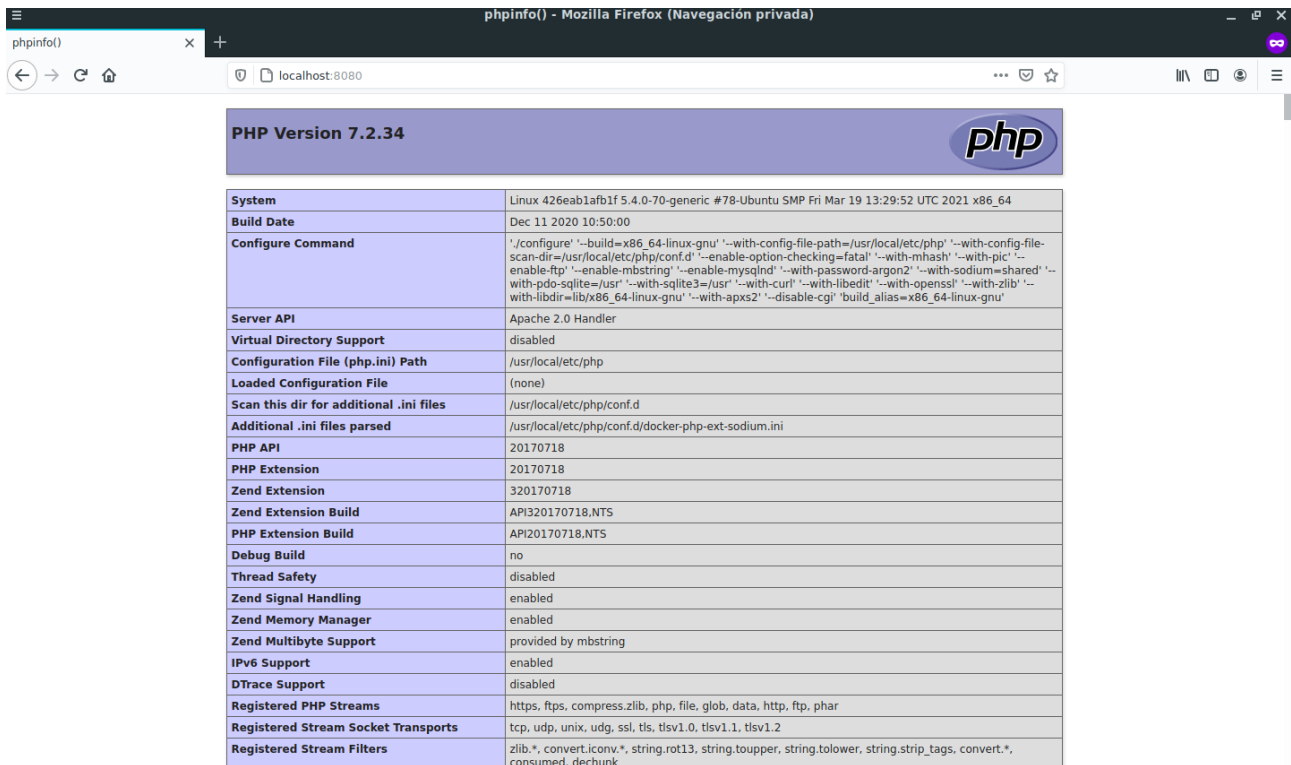
```
<?php
    phpinfo();
?>
```

A l'editor ens quedarà alguna cosa similar a:



Amb això, ja hem modificat el contingut del contenidor, creant un fitxer dins de la carpeta **"/var/www/html"**.

Si ara accedim a <http://localhost:8080> observem que tot funciona correctament i que hem pogut desenvolupar una aplicació dins d'un contenidor de forma gràfica i sense necessitat de mapejar fitxers a la nostra màquina amfitrió.



The screenshot shows a web browser window titled 'phpinfo() - Mozilla Firefox (Navegación privada)' with the address bar showing 'localhost:8080'. The page content is the PHPinfo() output for PHP 7.2.34. It includes a header with the PHP logo and version, followed by a table of system and configuration details.

<b>PHP Version 7.2.34</b>	
<b>System</b>	Linux 426eab1afb1f 5.4.0-70-generic #78-Ubuntu SMP Fri Mar 19 13:29:52 UTC 2021 x86_64
<b>Build Date</b>	Dec 11 2020 10:50:00
<b>Configure Command</b>	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' 'build_alias=x86_64-linux-gnu'
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/usr/local/etc/php
<b>Loaded Configuration File</b>	(none)
<b>Scan this dir for additional .ini files</b>	/usr/local/etc/php/conf.d
<b>Additional .ini files parsed</b>	/usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
<b>PHP API</b>	20170718
<b>PHP Extension</b>	20170718
<b>Zend Extension</b>	320170718
<b>Zend Extension Build</b>	API320170718.NTS
<b>PHP Extension Build</b>	API20170718.NTS
<b>Debug Build</b>	no
<b>Thread Safety</b>	disabled
<b>Zend Signal Handling</b>	enabled
<b>Zend Memory Manager</b>	enabled
<b>Zend Multibyte Support</b>	provided by mbstring
<b>IPv6 Support</b>	enabled
<b>DTrace Support</b>	disabled
<b>Registered PHP Streams</b>	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
<b>Registered Stream Socket Transports</b>	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
<b>Registered Stream Filters</b>	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

#### 4. BIBLIOGRAFIA

- [1] Docker Docs <https://docs.docker.com/>
- [2] Visual Studio Code "Working with contenidors" <https://code.visualstudio.com/docs/containers/overview>