

Para usuarios de Windows y Mac OS: entorno de desarrollo con Docker y Visual Studio Code

Índice

1	Instalación de Docker Desktop	1
2	Instalación de Visual Studio Code	3
3	Estructura de una carpeta de proyecto	3
3.1	Nota para los usuarios de GNU/Linux	3
4	Conexión al contenedor	3
5	Proceso de desarrollo, compilación y ejecución	4
6	Bibliografía	5

Este documento es un manual breve para poder desplegar el entorno de desarrollo utilizando Visual Studio Code, Docker y la extensión Dev - Containers. Está destinado a usuarios de los sistemas operativos Windows y Mac OS. En GNU/Linux no es necesario usar contenedores en esta asignatura.

El archivo [entorno-vscode-docker.tar.gz](https://github.com/entorno-vscode-docker) contiene una serie de ficheros para la configuración de un proyecto inicial, cuya estructura y contenido se explica más adelante.

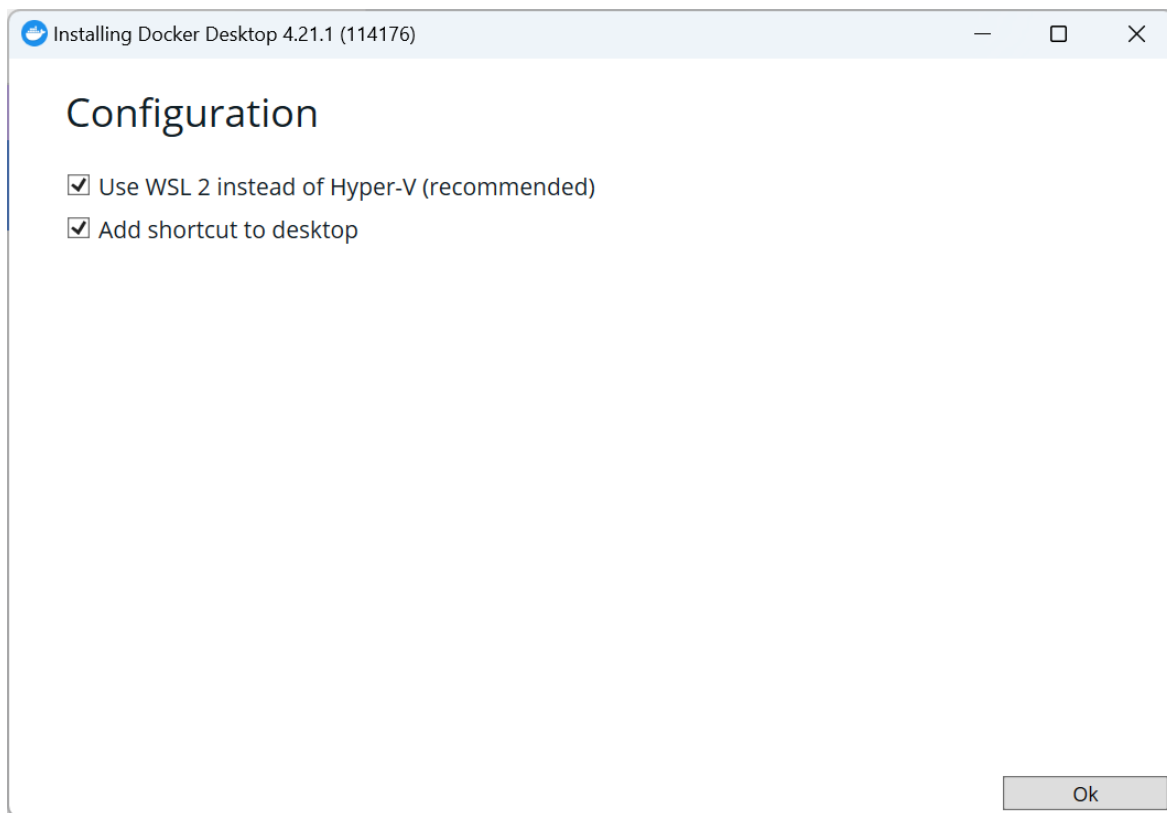
1 Instalación de Docker Desktop

- Descarga e instala Docker Desktop desde el [sitio web](https://docs.docker.com/docker-for-mac/install/) oficial de Docker para Windows o Mac OS, según el sistema operativo que utilices.
- Asegúrate de que Docker Desktop esté ejecutándose y funcione correctamente.

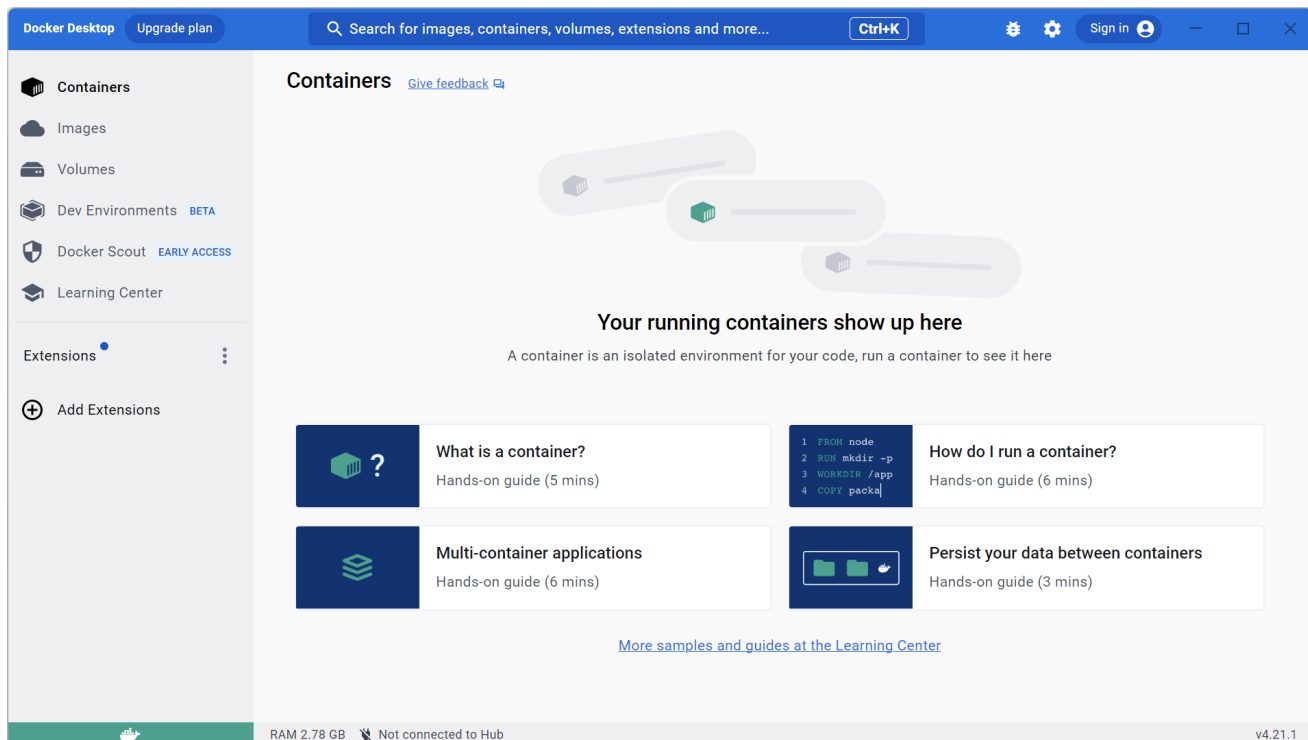
Nota para usuarios de MS Windows: Al arrancar Docker puede que te dé avisos importantes.

- Seguramente tengas que habilitar la virtualización en tu equipo desde la BIOS.
- También te puede pedir que actualices el Windows Subsystem for Linux (WSL): para ello abre una Power Shell y ejecuta los comandos: `wsl --install` (por si no está instalado) y `wsl --update`.

Aquí puedes ver una captura con Docker instalándose:



Y otra captura con Docker en ejecución:



2 Instalación de Visual Studio Code

- Descarga e instala Visual Studio Code desde el [sitio web](#) oficial.
- Asegúrate de tener instalada la extensión `Dev Containers` en Visual Studio Code. Puedes instalarla desde la sección de Extensiones de Visual Studio Code buscando “Dev Containers” y haciendo clic en “Install”.

3 Estructura de una carpeta de proyecto

Para que VSCode funcione contra un contenedor debemos tener una carpeta de proyecto con la siguiente estructura:

- `proyecto`: carpeta de proyecto, con el nombre que deseemos.
 - `.devcontainer`: subcarpeta de configuración del contenedor y del entorno (**fíjate en el punto inicial**).
 - * `Dockerfile`: fichero que configura la imagen del contenedor
 - * `devcontainer.json`: fichero que configura la apariencia y plugins iniciales del VSCode que se ejecuta en el contenedor
 - ... y dentro de `proyecto`, el código fuente con la estructura que queramos.

Esta estructura de directorio, junto con los ficheros `Dockerfile` y `devcontainer.json` y un archivo `.c` de prueba se proporcionan en el fichero comprimido [entorno-vscode-docker.tar.gz](#), mencionado más arriba.

3.1 Nota para los usuarios de GNU/Linux

Si abrimos un proyecto con una carpeta `.devcontainer` y el complemento `Dev Containers` está instalado, pueden ocurrir varias cosas, por ejemplo:

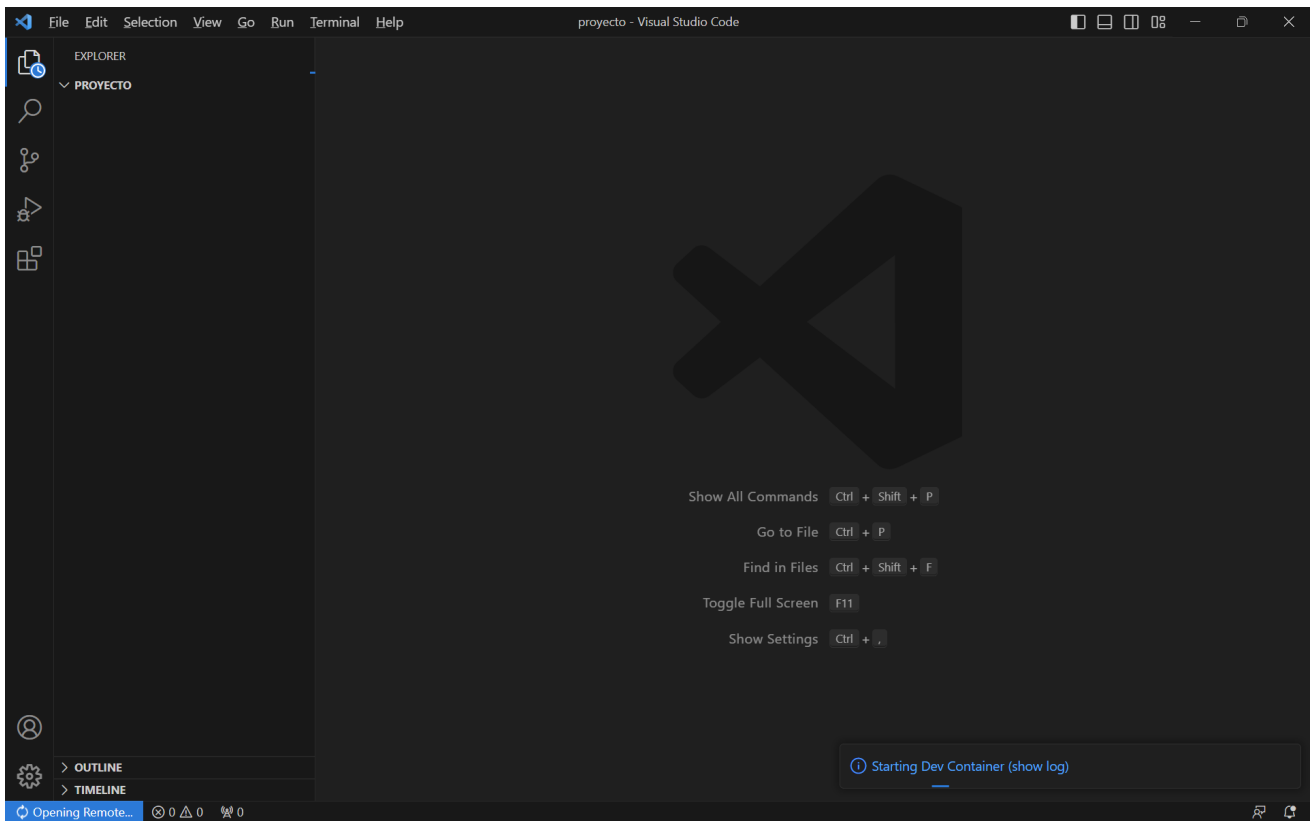
1. Si tenemos Docker instalado: se puede crear el contenedor si lo permitimos, cosa que es totalmente **innecesaria** en GNU/Linux, pues podemos compilar de forma nativa.
2. Si no tenemos Docker instalado o no tenemos permisos para crear contenedores: Visual Studio puede darnos alguna advertencia o mensaje de error, aunque podríamos trabajar con normalidad.

La recomendación si se trabaja en un sistema mixto es que al pasar los archivos de Windows/MacOS a GNU/Linux no se copie la carpeta `.devcontainer`.

4 Conexión al contenedor

- Una vez que tengas la estructura completa de la carpeta de trabajo `proyecto`, con su carpeta `.devcontainer`, abre la carpeta de trabajo con Visual Studio Code.
- Te preguntará si abrirlo con Dev Containers. Responde que sí.
- Visual Studio Code comenzará a construir y configurar el contenedor según las especificaciones del `Dockerfile` y `devcontainer.json`. Esto puede llevar unos minutos la primera vez que se ejecute.
- Una vez que se complete el proceso, la barra de estado de Visual Studio Code mostrará “Dev Container” junto al nombre de la carpeta de trabajo. Esto indica que estás trabajando dentro del contenedor.
- Los cambios realizados en la carpeta de trabajo se reflejarán dentro del contenedor y viceversa. Esto ocurre porque el contenedor sincroniza el directorio `workspace/usuario` del contenedor con el directorio de trabajo en el host.

Esta es una captura de pantalla con VSCode creando el contenedor



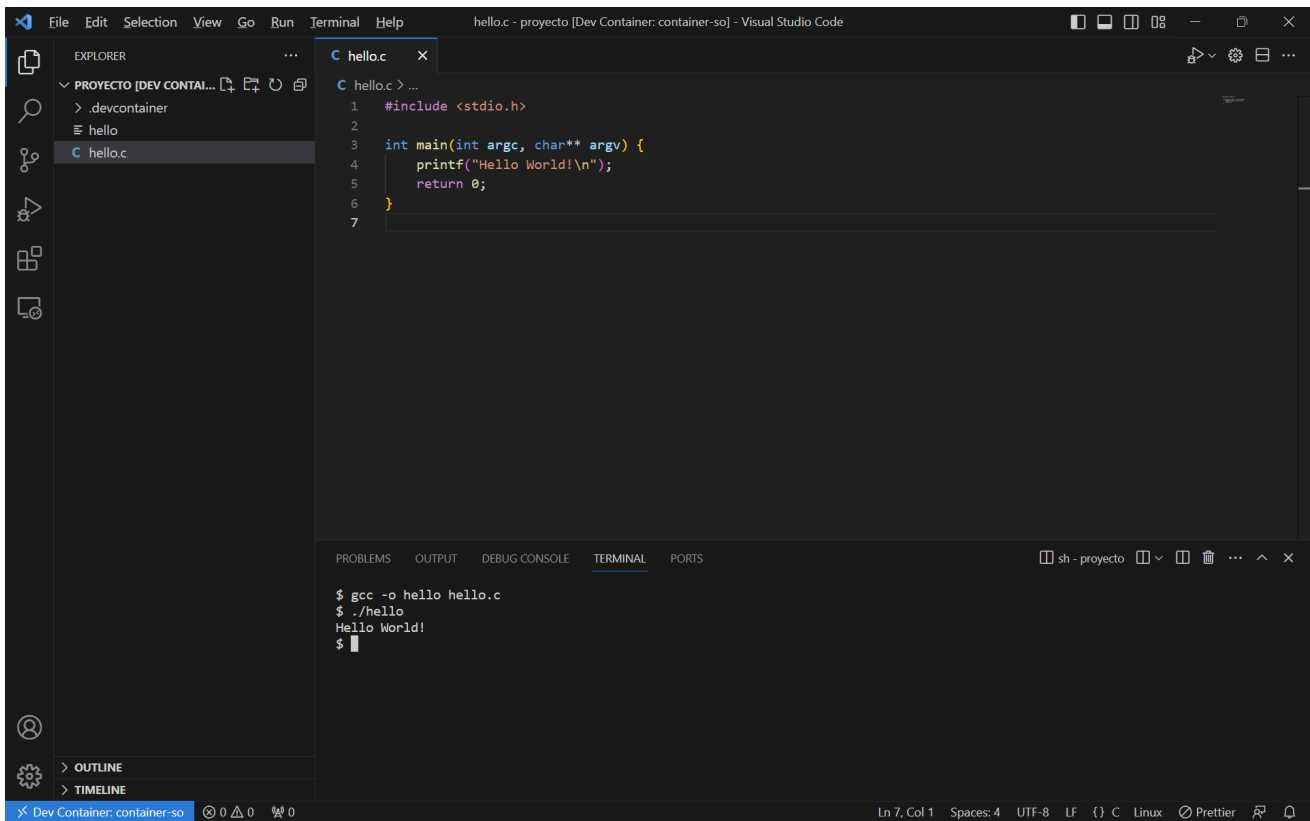
5 Proceso de desarrollo, compilación y ejecución

- El directorio de trabajo contiene un archivo llamado `hello.c`. Examínalo (ver figura más abajo):
- Haz click en **Terminal**, **New Terminal**.
- Introduce las siguientes órdenes:

```
gcc -o hello hello.c
./hello
```

- Deberías ver en pantalla `Hello World!`.
- Prueba a modificar `hello.c`. Los cambios se deberían efectuar también en tu equipo local.

Aquí se puede ver una captura de pantalla con VSCode ejecutando el programa `hello`:



6 Bibliografía

La documentación completa para cada sistema operativo se puede encontrar en las siguientes fuentes:

1. [Developing inside a Container](#): página de Microsoft orientada a la instalación de todo el kit de desarrollo, de principio a fin y con variantes para cada uno de los dos sistemas operativos. Ver apartado **Installation**, y sigue las instrucciones para Windows o MacOS, dependiendo de tu sistema operativo.
2. [Install Docker Desktop on Windows](#): página de Docker con instrucciones para instalar Docker Desktop en Windows.
3. [Install Docker Desktop on Mac](#): página de Docker con instrucciones para instalar Docker Desktop en MacOS.