

Docker Desktop

- Disponible para Windows, macOS y Linux.
- Proporciona una interfaz gráfica y CLI para gestionar contenedores.
- Permite crear, ejecutar y detener contenedores fácilmente desde tu máquina local.
- Integra Kubernetes opcionalmente para orquestación.

Ventajas de usar contenedores en desktop

- Portabilidad: el mismo contenedor corre igual en tu PC y en la nube.
- Aislamiento: cada proyecto tiene sus dependencias encapsuladas.
- Facilidad de pruebas: puedes levantar y destruir entornos en segundos.
- Aprendizaje: es el paso previo antes de orquestar en Kubernetes o en servidores.

Instalación en Windows (Docker Desktop)

Requisitos Previos Clave:

- Windows 10/11 de 64 bits (Pro, Enterprise o Education, o Home con WSL 2).
- Tener activado WSL 2 o la función Hyper-V de Windows.

Paso	Descripción
1. Descarga	Descarga el instalador Docker Desktop Installer.exe desde la página oficial de Docker.
2. Ejecutar	Haz doble clic en el archivo para iniciar el asistente de instalación.

3. Configuración	En la pantalla de configuración, asegúrate de que la opción "Use WSL 2 instead of Hyper-V" esté seleccionada (es la recomendada).
4. Finalizar	Sigue las instrucciones del asistente hasta que finalice y selecciona Cerrar.
5. Iniciar	Inicia Docker Desktop desde el menú de inicio y espera a que el icono en la barra de tareas se ponga verde (indicando que está en funcionamiento).

<https://docs.docker.com/desktop/setup/install/windows-install/>

Instalación en macOS (Docker Desktop)

Requisitos Previos Clave:

- Versión reciente de macOS (consulta la documentación oficial para la versión mínima requerida).
- Procesador Apple Silicon (M-series) o Intel.

Paso	Descripción
1. Descarga	Descarga el archivo .dmg de Docker Desktop para macOS (asegúrate de elegir la versión para tu tipo de chip: Apple Silicon o Intel).
2. Instalar	Abre el archivo .dmg y arrastra el icono de Docker a la carpeta de Aplicaciones (/Applications).
3. Iniciar	Haz doble clic en Docker.app en tu carpeta de Aplicaciones para iniciarlo.
4. Aceptar Términos	Acepta el Acuerdo de Servicio de Suscripción de Docker si es la primera vez que lo ejecutas.

5. Configurar	Sigue los pasos del asistente de configuración inicial para conceder los permisos necesarios.
---------------	---

<https://docs.docker.com/desktop/setup/install/mac-install/>

Instalación en Linux (Docker Engine)

Para Linux, la forma más habitual y recomendada es instalar Docker Engine desde los repositorios oficiales de Docker, especialmente para servidores. El proceso varía ligeramente por distribución, pero aquí están los pasos genéricos para Ubuntu o Debian:

Paso	Comando o Acción
1. Actualizar	Actualiza la lista de paquetes del sistema: <code>sudo apt update</code>
2. Instalar dependencias	Instala los paquetes necesarios para usar repositorios HTTPS: <code>sudo apt install ca-certificates curl gnupg lsb-release</code>
3. Añadir Clave GPG	Agrega la clave GPG oficial de Docker: <code>`curl -fsSL https://download.docker.com/linux/ubuntu/gpg`</code>
4. Añadir Repositorio	Agrega el repositorio estable de Docker a la lista de fuentes de APT.
5. Instalar Docker	Instala los paquetes del motor y la CLI: <code>sudo apt install docker-ce docker-ce-cli containerd.io</code>
6. Verificar	Comprueba que Docker se esté ejecutando: <code>sudo systemctl status docker</code> y luego ejecuta la imagen de prueba: <code>sudo docker run hello-world</code>

7. Post-instalación (Opcional)	Para ejecutar comandos de Docker sin <code>sudo</code> , agrega tu usuario al grupo <code>docker</code> : <code>sudo usermod -aG docker \$USER</code> . Cierra y vuelve a iniciar sesión.
--------------------------------	---



<https://docs.docker.com/desktop/setup/install/linux/>

Conceptos Fundamentales de Docker

Para trabajar con Docker de manera efectiva, es crucial entender la relación entre tres elementos principales: la Imagen, el Contenedor y el Dockerfile.

1. La Relación Imagen ↔ Contenedor

La mejor analogía para entender Docker es la siguiente:

-  Imagen (Image): La Plantilla o Receta.
 - Es el paquete inmutable (de solo lectura) que contiene todo lo necesario para que tu aplicación se ejecute: el código, el entorno de ejecución (ej. Node.js o Python), las dependencias del sistema operativo y las librerías.
 - Función: Es lo que subes y descargas de registros como Docker Hub.
-  Contenedor (Container): La Instancia Ejecutable.
 - Es una instancia en ejecución de una Imagen. Cuando ejecutas una Imagen, se convierte en un Contenedor.
 - Función: Es el entorno ligero y aislado donde tu aplicación corre en tiempo real, interactúa con la red, y utiliza los recursos de la CPU y la RAM de tu máquina host.

2. El Dockerfile: La Receta del Constructor

El `Dockerfile` es un archivo de texto simple que contiene una lista de instrucciones que Docker lee para construir la Imagen.

Instrucción Clave	Propósito	Ejemplo
FROM	Especifica la imagen base de la que partirá tu aplicación (ej. una imagen oficial de Python o Ubuntu).	FROM node:18-alpine
WORKDIR	Establece el directorio de trabajo predeterminado dentro del contenedor.	WORKDIR /app

COPY	Copia archivos (código fuente) desde tu máquina local al Contenedor.	COPY . .
RUN	Ejecuta un comando durante el proceso de construcción de la imagen (ej. instalar dependencias).	RUN npm install
CMD	Define el comando que se ejecutará al iniciar el Contenedor. Solo puede haber uno por Dockerfile .	CMD ["npm", "start"]

3. Docker Hub: El Catálogo Público

- Es el registro oficial (el "GitHub de las imágenes") donde desarrolladores y empresas almacenan y comparten sus imágenes de Docker.
- Imágenes Oficiales: Contiene imágenes mantenidas por la comunidad (como [python](#), [nginx](#), [mysql](#)) que sirven como punto de partida confiable para la mayoría de los proyectos.

III. Comandos de Docker Esenciales

Una vez que tengas tu [Dockerfile](#) listo, estos comandos son la clave para el flujo de trabajo de Docker: Construir, Ejecutar y Administrar.

1. Flujo de Trabajo Principal

Comando	Acción
docker build	Construye la Imagen. Toma tu Dockerfile y el código de tu proyecto, y crea la Imagen lista para usar.
docker run	Crea y Ejecuta el Contenedor. Toma una Imagen y la convierte en un proceso aislado en ejecución.
docker push	Sube la Imagen. Manda tu Imagen recién construida al Docker Hub o a un registro privado para compartirla.

2. Comandos de Administración Comunes

Comando	Alias	Descripción
docker ps	ps	Muestra los Contenedores en ejecución (procesos). Usa <code>docker ps -a</code> para ver también los detenidos.
docker stop	stop	Envía una señal para detener limpiamente un Contenedor en ejecución.
docker rm	rm	Elimina uno o más Contenedores detenidos de tu disco.
docker rmi	rmi	Elimina una o más Imágenes de tu disco local.
docker logs	logs	Muestra la salida de registro (logs) de un Contenedor, útil para la depuración (debugging).
docker exec	exec	Ejecuta un nuevo comando dentro de un Contenedor que ya está en funcionamiento.

IV. Primeros Pasos Prácticos

El primer comando para probar tu instalación y familiarizarte con el flujo pull -> run es:

Bash

None

```
docker run hello-world
```

Explicación del Comando `hello-world`:

1. `docker run`: Le dice a Docker que inicie un Contenedor.
2. `hello-world`: Es el nombre de la Imagen a utilizar.
3. Proceso: Docker primero verifica si tienes la imagen `hello-world` localmente. Si no la tienes, automáticamente la descarga (pull) desde Docker Hub.
4. Luego, crea y ejecuta un Contenedor a partir de esa Imagen, que simplemente imprime un mensaje de confirmación y luego se detiene.

<https://www.freecodecamp.org/espanol/news/guia-de-docker-para-principiantes-como-crear-tu-primera-aplicacion-docker/>

<https://www.datacamp.com/es/tutorial/docker-tutorial>

<https://kamilinux.com/guia-completa-de-docker/>