

# Docker Básico

---

**Arturo Silvelo**

Try New Roads

# Imágenes

## ¿Qué son las Imágenes de Docker?

---

- Las imágenes de Docker son plantillas de solo lectura utilizadas para crear contenedores.
- Contienen el sistema de archivos y todas las dependencias necesarias para ejecutar una aplicación.
- Se pueden compartir y distribuir fácilmente a través de registros como Docker Hub.
- Para crear imágenes personalizadas, se utiliza un `Dockerfile`, que es un archivo de texto con instrucciones para construir la imagen.

## ¿Qué es un Dockerfile?

---

- Un Dockerfile es un archivo de texto que define los pasos necesarios para construir una imagen de Docker, como instalar dependencias, copiar archivos y configurar el entorno.
- Se utiliza con el comando `docker build` para generar una imagen personalizada y reproducible.
- Un Dockerfile se construye usando una serie de instrucciones o palabras reservadas, cada una de las cuales genera una nueva capa en la imagen.
- Las capas son almacenadas de manera eficiente, y Docker reutiliza las capas que no han cambiado, lo que acelera las construcciones subsecuentes.
- La capa final de la imagen es el contenedor que se ejecutará, proporcionando el entorno listo para ejecutar la aplicación.

# Instrucciones Comunes en Dockerfile

---

- **FROM** : Define la imagen base a partir de la cual se construye la nueva imagen.

```
FROM nginx:alpine
```

- **WORKDIR** : Cambia el directorio donde se ejecutarán los siguientes comandos.
- **ENV** : Define las variables de entorno.
- **RUN** : Ejecuta un comando dentro de la imagen durante su construcción.

```
RUN apk update && apk add --no-cache iputils nano
```

- **VOLUME** : Crea un punto de montaje para volúmenes.

```
VOLUME ["/etc/nginx/conf.d"]
```

- **COPY / ADD**: Copia archivos o directorios del host al contenedor.

```
COPY ./index.html /usr/share/nginx/html/index.html
```

- **EXPOSE**: Informa a Docker que el contenedor escucha en un puerto específico.  
No mapea automáticamente el puerto al host.

```
EXPOSE 80
```

- **CMD**: Especifica el comando que se ejecutará cuando se inicie el contenedor. Es esencial para definir el comportamiento del contenedor.

```
CMD ["nginx", "-g", "daemon off;"]
```

Más información: <https://docs.docker.com/reference/dockerfile/>



## .dockerignore

---

- El archivo `.dockerignore` especifica qué archivos o directorios no deben ser copiados a la imagen de Docker.
- Similar al `.gitignore`, ayuda a evitar archivos innecesarios en la imagen.
- Mejora la eficiencia al reducir el tamaño de la imagen y acelera el proceso de construcción.
- Protege la seguridad al evitar incluir archivos sensibles en la imagen.

## ¿Cómo crear una imagen Docker?

---

- Para crear una imagen Docker solo necesitas un archivo llamado Dockerfile con las instrucciones para construir el entorno y la aplicación.
- Utiliza el comando `docker build` para generar la imagen a partir del Dockerfile y el contexto, que es la carpeta donde están los archivos necesarios.

```
docker build -t nombre-imagen:tag .
```

# Opciones avanzadas de `docker build`

- Puedes usar **varios** `-t` para etiquetar la imagen con diferentes nombres/tags:

```
docker build -t miapp:latest -t miapp:v1.0 .
```

- Para indicar un `Dockerfile` con otro nombre o en otra ruta, usa `-f`:

```
docker build -f Dockerfile.dev -t miapp:dev .
```

- Puedes construir imágenes desde cualquier carpeta como contexto:

```
docker build -t miapp:prod ./directorio-con-el-dockerfile
```