

# Ejemplos - Optimización de capas

---

Arturo Silvelo

Try New Roads

Este ejemplo muestra cómo optimizar el uso de capas en un Dockerfile para aprovechar la cache y reducir el tiempo de construcción.

- Agrupa instrucciones que no cambian frecuentemente (por ejemplo, instalación de dependencias).
- Usa `.dockerignore` para evitar copiar archivos innecesarios.

## Dockerfiles incluidos y explicación

- **Dockerfile.1-mal-optimizado:** Todas las instrucciones están separadas y no se aprovecha la cache. La imagen es más grande y lenta de construir.
  - Ejecución:

```
docker build -f Dockerfile.1-mal-optimizado -t capas-mal .  
docker run --rm capas-mal
```

- **Dockerfile.2-agrupa-run:** Se agrupan instrucciones `RUN` para reducir el número de capas y mejorar la cache.

- Ejecución:

```
docker build -f Dockerfile.2-agrupa-run -t capas-agrupadas .  
docker run --rm capas-agrupadas
```

- **Dockerfile.3-limpieza:** Se eliminan archivos temporales y dependencias de build al final del proceso para reducir el tamaño de la imagen.
  - Ejecución:

```
docker build -f Dockerfile.3-limpieza -t capas-limpias .  
docker run --rm capas-limpias
```

- **Dockerfile.4-reordenado:** Las capas se ordenan para maximizar el uso de la cache, poniendo primero las instrucciones que menos cambian (por ejemplo, dependencias).
  - Ejecución:

```
docker build -f Dockerfile.4-reordenado -t capas-reordenadas .  
docker run --rm capas-reordenadas
```
- **.dockerignore:** Evita copiar archivos innecesarios al contexto de construcción, lo que reduce el tamaño y mejora la cache.

## Recomendación

Construye y ejecuta cada imagen para comparar el tamaño, el tiempo de construcción y el uso de la cache. Observa cómo los cambios en el Dockerfile afectan el resultado final.