

# Comandos Avanzados de **find** - Guía Teórica con Ejemplos

## 1. Introducción

**find** es una herramienta poderosa de Unix/Linux para buscar archivos y directorios basándose en múltiples criterios y ejecutar acciones sobre los resultados.

## 2. Sintaxis Básica

bash

```
find [ruta] [opciones] [expresión] [acción]
```

## 3. Búsqueda por Nombre y Tipo

### Por nombre (case-sensitive)

bash

*# Buscar archivos exactos*

```
find /home -name "archivo.txt"
```

*# Usar comodines*

```
find /var -name "*.log"
```

*# Buscar en directorio actual*

```
find . -name "config*"
```

### Por nombre (case-insensitive)

bash

```
find /home -iname "*.PDF"
```

### Por tipo de archivo

bash

```
find /tmp -type f # archivos regulares
```

```
find /tmp -type d # directorios
```

```
find /tmp -type l # enlaces simbólicos
```

```
find /tmp -type s # sockets
```

```
find /tmp -type p # pipes (FIFO)
```

```
find /tmp -type b # dispositivos de bloque
```

```
find /tmp -type c # dispositivos de caracteres
```

## 4. Búsqueda por Tamaño

bash

*# Archivos mayores a 100MB*

```
find /var -size +100M
```

*# Archivos menores a 10KB*

```
find /home -size -10k
```

*# Archivos de exactamente 512 bytes*

```
find . -size 512c
```

*# Unidades: c(bytes), k(KB), M(MB), G(GB)*

```
find /logs -size +1G -size -5G # Entre 1GB y 5GB
```

## 5. Búsqueda por Tiempo

### Modificación (mtime)

bash

*# Modificados hace exactamente 7 días*

```
find /backup -mtime 7
```

*# Modificados en los últimos 7 días*

```
find /backup -mtime -7
```

*# Modificados hace más de 30 días*

```
find /tmp -mtime +30
```

### Acceso (atime)

bash

*# Accedidos en las últimas 24 horas*

```
find /var -atime -1
```

*# No accedidos en 90 días*

```
find /home -atime +90
```

### Cambio de metadatos (ctime)

bash

*# Cambios de permisos/propietario últimos 5 días*

```
find /etc -ctime -5
```

## Tiempo en minutos

bash

*# Modificados en los últimos 60 minutos*

`find /tmp -mmin -60`

*# Creados hace más de 120 minutos*

`find . -cmin +120`

## 6. Búsqueda por Permisos

bash

*# Permisos exactos (777)*

`find /var -perm 777`

*# Al menos estos permisos*

`find /bin -perm -755`

*# Cualquiera de estos permisos*

`find /tmp -perm /222`

*# Archivos con SUID*

`find / -perm -4000`

*# Archivos con SGID*

`find / -perm -2000`

*# Archivos con sticky bit*

`find /tmp -perm -1000`

## 7. Búsqueda por Usuario y Grupo

bash

*# Por usuario*

`find /home -user juan`

*# Por UID*

`find / -uid 1000`

*# Por grupo*

`find /var -group developers`

*# Por GID*

`find / -gid 100`

*# Archivos sin propietario (huérfanos)*

`find / -nouser`

*# Archivos sin grupo*

```
find / -nogroup
```

## 8. Operadores Lógicos

### AND (por defecto o -a)

bash

*# Archivos .txt mayores a 1MB*

```
find . -name "*.txt" -size +1M
```

*# Equivalente explícito*

```
find . -name "*.txt" -a -size +1M
```

### OR (-o)

bash

*# Archivos .txt o .pdf*

```
find . -name "*.txt" -o -name "*.pdf"
```

*# Archivos .log o .tmp mayores a 10MB*

```
find . \( -name "*.log" -o -name "*.tmp" \) -size +10M
```

### NOT (!)

bash

*# Todo excepto .txt*

```
find . ! -name "*.txt"
```

*# Directorios que no sean .git*

```
find . -type d ! -name ".git"
```

## 9. Profundidad de Búsqueda

bash

*# Solo en el directorio actual (sin subdirectorios)*

```
find . -maxdepth 1 -name "*.sh"
```

*# Mínimo 2 niveles de profundidad*

```
find /var -mindepth 2 -name "*.log"
```

*# Entre 2 y 4 niveles*

```
find /usr -mindepth 2 -maxdepth 4 -type f
```

## 10. Acciones sobre Resultados

### Imprimir (por defecto)

bash

```
find . -name "*.txt" -print
```

### Ejecutar comandos (-exec)

bash

*# Eliminar archivos encontrados*

```
find /tmp -name "*.tmp" -exec rm {} \;
```

*# Cambiar permisos*

```
find /var/www -type f -exec chmod 644 {} \;
```

*# Múltiples archivos en un comando (más eficiente)*

```
find . -name "*.txt" -exec rm {} +
```

*# Copiar archivos*

```
find . -name "*.conf" -exec cp {} /backup/ \;
```

### Confirmación interactiva (-ok)

bash

*# Pide confirmación antes de eliminar*

```
find /tmp -name "*.log" -ok rm {} \;
```

### Ejecutar con xargs

bash

*# Más eficiente para múltiples archivos*

```
find . -name "*.txt" | xargs grep "error"
```

*# Manejo seguro de nombres con espacios*

```
find . -name "*.txt" -print0 | xargs -0 grep "error"
```

### Ejecutar comandos complejos

bash

*# Comprimir archivos encontrados*

```
find /logs -name "*.log" -mtime +30 -exec gzip {} \;
```

*# Mover a directorio específico*

```
find . -name "*.bak" -exec mv {} /backups/ \;
```

*# Ejecutar script personalizado*

```
find /data -type f -size +100M -exec ./procesar.sh {} \;
```

## 11. Ejemplos Avanzados Combinados

### Limpiar archivos temporales antiguos

bash

```
find /tmp -type f -mtime +7 -name "*.tmp" -delete
```

### Buscar archivos duplicados por tamaño

bash

```
find /home -type f -size +10M -exec md5sum {} \; | sort | uniq -d -w32
```

### Archivos modificados recientemente excluyendo directorios

bash

```
find /var/log -type f -mtime -1 ! -path "*/archive/*" -name "*.log"
```

### Buscar y listar con detalles

bash

```
find /etc -name "*.conf" -exec ls -lh {} \; 2>/dev/null
```

### Comprimir logs antiguos

bash

```
find /var/log -name "*.log" -mtime +90 -exec gzip {} \;
```

### Encontrar archivos vacíos

bash

```
find /home -type f -empty
```

```
find /var -type d -empty # directorios vacíos
```

### Buscar por contenido y tamaño

bash

```
find /var/www -type f -name "*.php" -exec grep -l "password" {} \;
```

## 12. Optimización y Buenas Prácticas

### Limitar búsqueda con -prune

bash

*# Excluir directorio .git de la búsqueda*

```
find . -path "./.git" -prune -o -name "*.js" -print
```

## Usar -delete en lugar de -exec rm

bash

*# Más eficiente*

```
find /tmp -name "*.tmp" -delete
```

## Búsqueda más rápida con -quit

bash

*# Detener tras encontrar el primero*

```
find / -name "needle.txt" -quit
```

# 13. Casos de Uso Reales

## Backup de archivos modificados hoy

bash

```
find /project -type f -mtime 0 -exec cp --parents {} /backup/ \;
```

## Encontrar archivos grandes que ocupan espacio

bash

```
find / -type f -size +500M -exec ls -lh {} \; 2>/dev/null | sort -k5 -h
```

## Cambiar permisos recursivamente solo en archivos

bash

```
find /var/www -type f -exec chmod 644 {} \;
```

```
find /var/www -type d -exec chmod 755 {} \;
```

## Buscar código vulnerable

bash

```
find /var/www -name "*.php" -exec grep -H "eval(" {} \;
```

Este comando es extremadamente versátil y potente cuando se combinan sus diferentes opciones. La clave está en entender los operadores lógicos y saber encadenar criterios de búsqueda efectivamente.