

EJEMPLOS DETALLADOS PARA CLASE

Ejemplo 1: Diagnostico Completo de Conectividad

Escenario: "Los usuarios reportan que no pueden acceder a nuestra aplicación web"

1. *Crear archivo*
touch dig.sh
2. *Abrirlo con nano*
nano dig.sh
3. *Escribir el siguiente script*

```
#!/usr/bin/env bash
set -euo pipefail
```

```
TIMEOUT=5
HOST_TEST="google.com"
HTTP_TEST_URL="http://httpbin.org/get"
```

```
echo "=== DIAGNÓSTICO DE CONECTIVIDAD COMPLETO ==="
echo "Objetivo: Verificar acceso a aplicación web"
echo
```

```
# 1. Verificar conectividad básica
echo "1. TEST DE CONECTIVIDAD BÁSICA"
echo "Probando conectividad a servidor..."
if ping -c 3 -W "$TIMEOUT" "$HOST_TEST" > /dev/null 2>&1; then
    echo "✓ Conectividad a Internet: OK"
else
    echo "✗ Conectividad a Internet: FALLO"
    echo " → Verificar cables de red, WiFi, o ISP"
fi
```

```
# 2. Verificar DNS
echo -e "\n2. TEST DE RESOLUCIÓN DNS"
echo "Resolviendo nombres de dominio..."
if command -v nslookup >/dev/null 2>&1 && nslookup "$HOST_TEST" >
/dev/null 2>&1; then
    echo "✓ Resolución DNS: OK"
    DNS_SERVER=$(nslookup "$HOST_TEST" 2>/dev/null | grep -m1 'Server' |
awk -F: '{print $2}')
    [[ -n "${DNS_SERVER:-}" ]] && echo " DNS Server: $DNS_SERVER"
else
    echo "✗ Resolución DNS: FALLO"
    echo " → Verificar configuración DNS (8.8.8.8, 1.1.1.1)"
fi
```

```
# 3. Verificar ruta de red
echo -e "\n3. ANÁLISIS DE RUTA DE RED"
echo "Trazando ruta a $HOST_TEST..."
if command -v traceroute >/dev/null 2>&1; then
    traceroute -n -m 5 -w 2 -q 1 "$HOST_TEST" 2>/dev/null | head -8 \
        || echo " (No se alcanzó el destino en 5 saltos; posible filtrado ICMP)"
elif command -v tracepath >/dev/null 2>&1; then
    tracepath -n "$HOST_TEST" 2>/dev/null | head -8 \
        || echo " (tracepath no pudo completar; puede ser normal)"
else
    echo " Ni traceroute ni tracepath instalados. Saltando."
fi
```

```
# 4. Verificar servicio web específico
echo -e "\n4. TEST DE SERVICIO WEB"
echo "Probando acceso HTTP: $HTTP_TEST_URL"
if curl -s -I -L --max-time "$TIMEOUT" "$HTTP_TEST_URL" | grep -qE
'^HTTP/. * (200|3[0-9])'; then
    echo "✓ Servicio HTTP: OK"
else
    echo "X Servicio HTTP: FALLO"
fi
```

```
echo -e "\n=== RESUMEN ==="
echo "Diagnóstico completado: $(date)"
```

```
4. Dar permisos al archivo y ejecutarlo
chmod +x dig.sh
./dig.sh | tee diagnostico.log
```

Ejemplo 2: Análisis de Puertos y Servicios

1. *Crear archivo*
touch puertos.sh
2. *Abrirlo con nano*
nano puertos.sh
3. *Escribir el siguiente script*

```
#!/bin/bash
echo "=== ANÁLISIS DE PUERTOS Y SERVICIOS ==="
```

```
# Lista de puertos comunes para verificar
declare -a ports=(
    "22:SSH"
    "53:DNS"
```

```

    "80:HTTP"
    "443:HTTPS"
    "3000:Development"
    "8080:Alt-HTTP"
)

echo "Verificando puertos locales en escucha..."
for port_info in "${ports[@]}"; do
    port="${port_info%%.*}"
    service="${port_info##*}"

    if netstat -tln 2>/dev/null | grep ":$port " > /dev/null; then
        echo "✓ Puerto $port ($service): ABIERTO"
    else
        echo "○ Puerto $port ($service): cerrado"
    fi
done

# Cuenta conexiones TCP en estado ESTABLISHED
echo -e "\n=== CONEXIONES ACTIVAS ==="
echo "Conexiones establecidas:"
netstat -tn 2>/dev/null | grep ESTABLISHED | wc -l | xargs echo "Total:"

#Top 5 de IPs remotas con más conexiones activas
echo -e "\nTop 5 conexiones por IP remota:"
netstat -tn 2>/dev/null | grep ESTABLISHED | \
    awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -nr | head -5

4. Dar permisos al archivo y ejecutarlo
    chmod +x puertos.sh
    ./puertos.sh

```

Ejemplo 3: Test de Performance de DNS

```

1. Crear archivo
    touch dns_bench.sh
2. Abrirlo con nano
    nano dns_bench.sh
3. Escribir el siguiente script

#!/usr/bin/env bash
set -u

echo "=== BENCHMARK DE SERVIDORES DNS ==="

# Lista de servidores DNS populares
declare -a dns_servers=(

```

```

"8.8.8.8:Google"
"1.1.1.1:Cloudflare"
"208.67.222.222:OpenDNS"
"192.168.1.1:Router-Local"
)

test_domain="github.com"

echo "Probando velocidad de resolución DNS para $test_domain"
printf "%-20s %-12s %-8s\n" "Servidor DNS" "Tiempo(ms)" "Estado"
echo
"=====

for dns_info in "${dns_servers[@]"; do
    dns="${dns_info%%:*}"
    name="${dns_info##*:}"

    # Medir tiempo de resolución (ms desde epoch)
    start_time=$(date +%s%3N)
    if nslookup -timeout=2 -retry=1 "$test_domain" "$dns" > /dev/null 2>&1; then
        end_time=$(date +%s%3N)
        elapsed=$((end_time - start_time))
        printf "%-20s %-12s %-8s\n" "$name ($dns)" "${elapsed}" "OK"
    else
        printf "%-20s %-12s %-8s\n" "$name ($dns)" "-" "TIMEOUT"
    fi
done

echo -e "\n=== ANÁLISIS DE REGISTROS DNS ==="
echo "Registros A para $test_domain:"
# La 1.ª línea de nslookup muestra el servidor; filtramos las IP del dominio
nslookup -type=A "$test_domain" 2>/dev/null \
| grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}' \
| sed '1d' | sort -u

echo -e "\nServidores de nombres autoritativos (NS):"
nslookup -type=NS "$test_domain" 2>/dev/null \
| awk '/nameserver/ {print $NF}' | sed 's/\.$//'

```

4. *Dar permisos al archivo y ejecutarlo*

```

chmod +x dns_bench.sh
./dns_bench.sh

```