

UNIVERSIDAD AUTONOMA DE SINALOA

UNIDAD ACADEMICA FACULTAD DE INGENIERIA MOCHIS

LICENCIATURA EN INGENIERIA DE SOFTWARE



ASIGNATURA:

Asministración de sistemas

DOCENTE:

Dr. Herman Geovany Ayala Zuñiga

ALUMNO:

Alberto Torres Chaparro

GRUPO:

3-01

TAREA 2:

Automatización y Gestión del Servidor DHCP

Objetivo de la tarea

Diseñar e implementar una solución automatizada mediante scripts (**Bash** para Oracle Linux y **PowerShell** para Windows Server) que permita instalar, configurar y monitorear un servidor DHCP de manera desatendida. El sistema garantiza la asignación dinámica de parámetros de red (IP, Máscara, Gateway, DNS) a clientes en una red interna, asegurando la integridad de los datos y reduciendo el error humano.

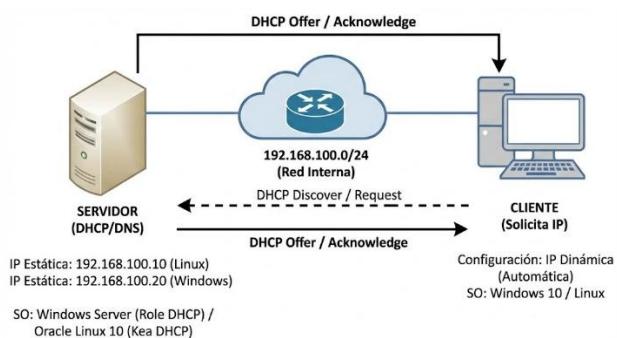
Historial de cambios del repositorio

ID	Fecha	Autor	Descripción del Cambio (Commit Message)
833988	06/02/2026	Alberto Torres Chaparro	Script inicial de verificación del servicio kea DHCP en Oracle Linux
e7c54e6	06/02/2026	Alberto Torres Chaparro	Implementación de la instalación desatendida del servicio kea DHCP
d5eac8d	06/02/2026	Alberto Torres Chaparro	Implementación de configuración y validación estricta de parámetros DHCP
d6b6f9e	06/02/2026	Alberto Torres Chaparro	Mejora visual de los clientes conectados
656c327	06/02/2026	Alberto Torres Chaparro	Finalización del script de Linux agregando un menú de corrección selectiva y un monitoreo...
e8b3d77	07/02/2026	Alberto Torres Chaparro	Verificación e instalación del rol DHCP en Windows Server
a4a02a5	07/02/2026	Alberto Torres Chaparro	Agregue la configuración del rol DHCP con validaciones estrictas y la instalación de dependencias...
7c8630e	07/02/2026	Alberto Torres Chaparro	Script finalizado. Agregados: Módulo de monitoreo en tiempo real y ajustes visuales.

Diagrama de topología

La práctica se ejecuta en un entorno virtualizado (VirtualBox) configurado en "Red Interna", asegurando el aislamiento del tráfico.

- **Servidor A (Oracle Linux 10):**
 - o IP: 192.168.100.10
 - o Software: Kea DHCP Server.
- **Servidor B (Windows Server):**
 - o IP: 192.168.100.20
 - o Software: Microsoft DHCP Role.
- **Cliente:**
 - o Configuración: IP Dinámica (DHCP).
 - o SO: Windows 10 / Linux.
- **Segmento de Red: 192.168.100.0 /24.**



Requisitos Previos

- **Conexión a Internet:** Necesario para la descarga de paquetes.
- **Privilegios:** Root (Linux) o Administrador (Windows).

Instrucciones de Ejecución

En Linux

- 1- Diríjase al directorio donde está ubicado el archivo
- 2- Ejecute el script con el siguiente comando:

```
Sudo bash servicio_kea.sh
```

En Windows Server

- 1- Ejecute Powershell como administrador
- 2- Diríjase al directorio donde está ubicado el archivo
- 3- Ejecute el script

```
.\servicio_dhcp.ps1
```

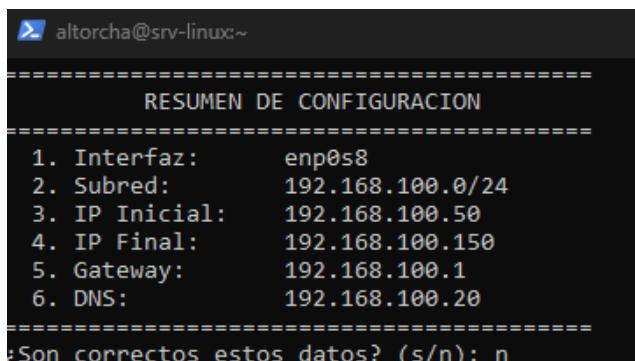
Flujo de Interacción

Ambos scripts fueron diseñados con un enfoque interactivo. Antes de aplicar cualquier cambio en el sistema, presentan un **Resumen de Configuración** que el administrador debe confirmar (s/n). A continuación, se detallan los parámetros específicos solicitados según la arquitectura del Sistema Operativo.

Oracle Linux (Script Bash)

Debido a la arquitectura de Kea *DHCP*, este script solicita explícitamente la tarjeta de red física.

- 1. Interfaz:** Se debe ingresar el identificador del adaptador de red.
- 2. Subred:** La red base en formato CIDR (ej. 192.168.100.0/24).
- 3. Rango de IPs:** Definición de la primera y última dirección a asignar.
- 4. Parámetros de Red:** Gateway y DNS que se entregarán a los clientes.



```
alotorcha@srv-linux:~  
=====  
RESUMEN DE CONFIGURACION  
=====  
1. Interfaz:      enp0s8  
2. Subred:       192.168.100.0/24  
3. IP Inicial:   192.168.100.50  
4. IP Final:    192.168.100.150  
5. Gateway:     192.168.100.1  
6. DNS:          192.168.100.20  
=====  
:Son correctos estos datos? (s/n): n
```

Windows Server (Script PowerShell)

El script de PowerShell abstrae la capa física y se centra en la definición lógica del ámbito (Scope).

- 1. Nombre del Ámbito:** Etiqueta descriptiva para identificar la red en la consola de administración (ej. **Nettest** o Red_Sistemas). A diferencia de Linux, aquí no se pide interfaz física.
- 2. Red ID y Máscara:** Identificadores de la subred (192.168.100.0 con máscara 255.255.255.0).
- 3. Rango de IPs:** Mismo rango de direcciones que en el escenario Linux (.50 a .150).
- 4. Validación de DNS:** El script intenta validar la IP del DNS (.20). Como se observa en la evidencia, si el servicio DNS local aún está inicializando, el script captura la advertencia y fuerza la configuración para asegurar la continuidad.

```
Administrator: C:\Windows\system32\cmd.exe
=====
RESUMEN DE CONFIGURACION
=====
1. Nombre:      Nettest
2. Red ID:      192.168.100.0
3. Mascara:     255.255.255.0
4. IP Inicial: 192.168.100.50
5. IP Final:   192.168.100.150
6. Gateway:    192.168.100.1
7. DNS:        192.168.100.20
=====
Son correctos estos datos? (s/n): s
```

Bitácora de Desarrollo

Desarrollo del Script para Oracle Linux (Bash + Kea DHCP)

El desarrollo en Linux se centró en la modularidad y el control explícito de los archivos de configuración de Kea.

- **Detección e Instalación de Dependencias:** Se implementó una función *check_install* que verifica la existencia del binario *kea-dhcp4*. Si el servicio no es detectado, el script ejecuta automáticamente *dnf install -y isc-kea-dhcp4-server*, garantizando que el entorno esté listo sin intervención manual.
- **Manipulación de Archivos de Configuración (JSON):** A diferencia del antiguo ISC-DHCP, Kea utiliza formato JSON (*/etc/kea/kea-dhcp4.conf*). El script no sobrescribe el archivo a ciegas; en su lugar, utiliza herramientas de edición de texto (*sed* y *echo*) para inyectar la configuración de la subred, el rango de IPs (pool) y la interfaz específica (*interfaces-config*) solicitada al usuario.

- **Validación de Interfaz de Red:** Debido a que Kea requiere declarar explícitamente la interfaz de escucha, el script solicita este dato. Esto es una medida de seguridad para evitar que el servicio DHCP responda en interfaces no deseadas (como la WAN), lo cual podría causar conflictos en redes externas.

Desarrollo del Script para Windows Server (PowerShell)

El script de Windows se diseñó bajo el paradigma de "Infraestructura como Código" (IaC), priorizando la experiencia de usuario y la robustez ante errores del sistema.

- **Instalación Silenciosa ("Ninja Mode"):** Para mejorar la experiencia de usuario, se utilizó el cmdlet Start-Job para instalar los roles DHCP y DNS en segundo plano.
 - *Lógica:* Start-Job -ScriptBlock { Install-WindowsFeature ... }. Esto permite mostrar un indicador de "Cargando..." en lugar de bloquear la consola con barras de progreso nativas de Windows, reduciendo el tiempo de percepción de espera.
- **Manejo de Excepciones en DNS (Try/Catch):** Durante las pruebas, se detectó que Windows Server intenta validar la conectividad del servidor DNS al momento de configurarlo (Set-DhcpServerv4OptionValue -OptionId 6). Si el servicio DNS local (192.168.100.20) está recién instalado y aún no responde, el comando falla y detiene el script.
 - *Solución:* Se encapsuló este comando en un bloque try { ... } catch { ... }. Esto permite capturar el error de validación y forzar la aplicación de la configuración, asegurando que el script continúe su ejecución exitosamente.
- **Idempotencia y Sobrescritura de Ámbitos:** Se implementó lógica para evitar conflictos de duplicidad. Antes de crear un nuevo ámbito, el script verifica si el Scopeld ya existe.

Código:

```
if (Get-DhcpServerv4Scope -Scopeld $RedID) { Remove-DhcpServerv4Scope ... }.
```

Esto permite re-ejecutar el script múltiples veces en el mismo servidor (por ejemplo, para corregir un error de dedo) sin que el sistema arroje errores fatales por recursos duplicados.

- **Monitor de Clientes en Tiempo Real:** Se desarrolló un módulo final que convierte la consola en un panel de monitoreo. Utiliza un bucle infinito while(\$true) que consulta Get-DhcpServerv4Lease cada 3 segundos. La salida se formatea utilizando cadenas de formato de .NET ("{0,-16}" -f ...) para alinear perfectamente las columnas de IP, Hostname y MAC Address, facilitando la lectura visual de las conexiones entrantes.

Evidencias de validación

Validación de parámetros en Linux

```
altorcha@srv-linux:~  
=====  
RESUMEN DE CONFIGURACION  
=====  
1. Interfaz:      enp0s8  
2. Subred:        192.168.100.0/24  
3. IP Inicial:   192.168.100.50  
4. IP Final:     192.168.100.150  
5. Gateway:       192.168.100.1  
6. DNS:           192.168.100.20  
=====  
¿Son correctos estos datos? (s/n):
```

Validación de parámetros en Windows

```
Administrator: C:\Windows\system32\cmd.exe  
=====  
RESUMEN DE CONFIGURACION  
=====  
1. Nombre:        Prueba  
2. Red ID:        192.168.100.0  
3. Mascara:       255.255.255.0  
4. IP Inicial:   192.168.100.50  
5. IP Final:     192.168.100.150  
6. Gateway:       192.168.100.1  
7. DNS:           192.168.100.20  
=====  
Son correctos estos datos? (s/n): -
```

Pruebas de funcionamiento del script

Oracle Linux

```
altorcha@srv-linux:~  
=====  
MONITOR DHCP KEA - 14:02:44  
(Presione Ctrl+C para salir)  
=====  
DIRECCION IP      | MAC ADDRESS      | HOSTNAME  
-----|-----|-----  
192.168.100.62    | 08:00:27:6b:54:7b | cte-windows.  
-
```

Windows Server

The screenshot shows two windows side-by-side. The left window is titled 'Srv-Windows-Sistemas (Servidor windows limpio) [Corriendo] - Oracle VirtualBox' and contains a command prompt window. The right window is titled 'Cte-Windows-Sistemas (Cliente windows limpio) [Corriendo] - Oracle VirtualBox' and also contains a command prompt window. Both windows show the output of 'ipconfig /release' and 'ipconfig /renew' commands, displaying network configuration details for Ethernet adapters.

```
Administrator: C:\Windows\system32\cmd.exe
Administrator: PS C:\Windows\system32> ipconfig /release
Administrator: PS C:\Windows\system32> ipconfig /renew
```

```
Administrator: C:\Windows\system32> ipconfig /release
Administrator: C:\Windows\system32> ipconfig /renew
```

Conclusiones

La realización de esta práctica permitió comprobar la eficacia de la **Automatización de Infraestructura** frente a la configuración manual tradicional. A través del desarrollo de scripts en *Bash* y *PowerShell*, se logró reducir el tiempo de despliegue de un servidor DHCP de varios minutos a cuestión de segundos, eliminando prácticamente el error humano en la asignación de parámetros críticos como la Puerta de Enlace y el DNS.

Un hallazgo técnico significativo fue la diferencia arquitectónica entre ambos sistemas operativos:

Oracle Linux (Kea DHCP): Demostró un enfoque de "**Seguridad por Diseño**", obligando al administrador a declarar explícitamente la interfaz física de escucha (`enp0s8`). Esto ofrece un control granular y evita fugas de tráfico DHCP en interfaces no deseadas.

Windows Server: Presentó un modelo de **Abstracción**, donde el servicio gestiona automáticamente la vinculación con las tarjetas de red basándose en los Ámbitos lógicos. Si bien esto simplifica la gestión, requiere mecanismos de validación robustos (como los bloques try/catch implementados) para manejar dependencias como el servicio DNS.

Referencias

- **Internet Systems Consortium (ISC).** (2024). *Kea Administrator Reference Manual*. Recuperado de: <https://kea.readthedocs.io/en/latest/>
- **Microsoft.** (2024). *DhcpServer Module | Microsoft Learn*. Recuperado de: <https://learn.microsoft.com/en-us/powershell/module/dhcpserver/>
- **Oracle.** (2024). *Oracle Linux 9/10 Networking Guide*. Recuperado de: <https://docs.oracle.com/en/operating-systems/oracle-linux/>