

# IaC – Network Automation

Temario Curso Network Automation I - Año 2025 - By Edgardo Scrimaglia

---

## Introducción: ¿Qué es IaC?

La **Infraestructura como Código (IaC)** es un enfoque que permite **gestionar y aprovisionar infraestructura mediante código en lugar de procesos manuales**, lo que facilita la creación de entornos coherentes, reproducibles y versionados. Al definir y concebir la configuración de redes, servidores y recursos de Cloud en bloques de código, se logra automatizar despliegues de múltiples dispositivos, reducir errores humanos y aplicar buenas prácticas como control de versiones y despliegues repetibles. IaC es fundamental tanto en DevOps como en cloud y network automation porque permite a los equipos tratar la infraestructura con la misma disciplina que el desarrollo de software

---

## Objetivo del curso

Formar profesionales capaces de diseñar, desarrollar y poner en marcha soluciones de **automatización de redes** mediante herramientas como **Python, Netmiko, Ansible, TextFSM, Jinja2** y **REST API** aplicando buenas prácticas en entornos simulados con **GNS3** y abordando casos reales de administración de infraestructuras de red.

---

## Estructura general

-  **Clases:** 16 sesiones de 3 h
  -  **Laboratorios:** 7 sesiones prácticas (intercaladas)
  -  **Modalidad:** Presencial + Remota (preferentemente remota)
  -  **Laboratorio virtual:** GNS3
-

## Temas del curso

1. **Instalación del entorno de automatización**
    - Creación de un entorno virtual
    - Instalación de lenguajes (Python, Ansible, Jinja2)
    - Instalación de VMware Fusion
    - Instalación del simulador GNS3
  2. **Fundamentos de scripting en Python y acceso a dispositivos**
    - Introducción a Python (tipos, estructuras, funciones)
    - Conexión SSH con **Netmiko**
    - Automatización de comandos CLI básicos
  3. **Netmiko aplicado a la red**
    - Biblioteca Python que simplifica conexiones SSH a dispositivos de red
    - Establecer sesiones SSH (ConnectHandler), enviar comandos show y de configuración, y manejar sesiones privilegiadas y de múltiples dispositivos
    - Uso de `send_config_set`, `send_command`, `send_command_timing`, `send_config_from_file`, `enable()`, `disconnect()` y uso de TextFSM integrado para parseo
  4. **Parseo y estructuración de datos**
    - Uso de **TextFSM** para convertir salida CLI en datos estructurados
    - Extracción y manipulación de información para análisis y reporte
  5. **Generación automatizada de configuraciones**
    - Plantillas con **Jinja2** (variables, loops, filtros)
    - Uso de Jinja2 en Python y en Ansible para configurar dispositivos de forma dinámica
  6. **Automatización a escala y APIs de red**
    - **Ansible**: inventarios, playbooks, módulos `net_*`, roles, idempotencia e integración con Jinja2
    - **REST API / XML**: consumo con Python `requests`, parseo XML
-

## Temario por sesión

1 <b>Introducción &amp; Python – fundamentos</b>	Entorno, versiones, variables, tipos, estructuras, scripts.
2 <b>Python – conexiones de red</b>	SSH con Netmiko básico, automatización de comandos show.
3 <b>Lab 1: Python → dispositivos</b>	Automatización vía Netmiko: show, config, backup básico en GNS3.
4 <b>Python avanzado y NETMIKO</b>	Manejo de prompting, control de flujo, factor delay, configuración.
5 <b>Templating: TextFSM</b>	Parseo de salida CLI, crear templates, extracción estructurada de datos.
6 <b>Lab 2: Parsing con TextFSM</b>	Uso de TextFSM en Python para extraer datos desde salida CLI simulada.
7 <b>Templating: Jinja2</b>	Plantillas de configuración, variables, loops y filtros.
8 <b>Lab 3: Generación de configs con Jinja2</b>	Crear configuraciones parametrizadas y desplegarlas en GNS3.
9 <b>Introducción a Ansible</b>	Arquitectura, instalación, inventario, módulos básicos, playbooks YAML.
10 <b>Ansible &amp; Network Modules</b>	Uso de net_, ios_ y cli_config; pruebas de idempotencia y handlers.
11 <b>Lab 4: Ansible básico</b>	Playbook para obtener información y cambiar configuración en dispositivos Cisco.
12 <b>Ansible avanzado + Jinja2</b>	Roles, conditionals, loops, includes, Variables y templating en playbooks.
13 <b>Lab 5: Conjunto Ansible + Jinja2</b>	Plantillas y roles aplicados a configuración avanzada en GNS3.
14 <b>REST API &amp; XML</b>	Usar Python “requests” con RESTCONF/NETCONF, parseo XML, APIs de dispositivos.
15 <b>Lab 6: REST API</b>	Interacción con API simulada, consulta y modificación.
16 <b>Lab 7: Integración total + repaso</b>	Proyecto final: automatizar config, parsing, clean-up con Python y Ansible.

---

## Laboratorios (7)

1. Python + Netmiko – show/config en GNS3
2. TextFSM – extracción organizada de datos CLI
3. Jinja2 – generación y despliegue de configs
4. Ansible básico – show, config vía módulos
5. Ansible + Jinja2 – roles y plantillas
6. REST API – consumo de APIs de red
7. Integración completa – Python + Ansible + APIs

---

## Flujo del curso

- Primero afianzamos **Python** y **Netmiko**, luego avanzamos a **TextFSM** y **Jinja2**
- Posteriormente incorporamos **Ansible**, y cerramos con **REST API/XML**
- Se integran laboratorios cada 2 sesiones teóricas, garantizando práctica continua

---

## Ventajas y adaptación

- Cubre todos los temas requeridos: Python, Ansible, Netmiko, TextFSM, Jinja2, REST-API y XML
- Diseño modular, práctico, equilibrado entre teoría y práctica
- Uso de GNS3 permite simular entornos reales de forma remota
- Laboratorios orientados a resolver problemas reales de automatización de red

---

## Prerrequisitos

- **Conocimientos básicos de Python**, incluyendo estructuras de control, funciones y manejo de paquetes (por ejemplo, `pip`).
- **Nivel de networking equivalente a certificación CCNA**, es decir: subredes IP, VLANs, enrutamiento y switching básicos, SSH.
- **Equipo propio**: computadora con acceso a internet, entorno Python 3.12 o mayor instalado preferentemente, y privilegios para instalar librerías (Netmiko, Jinja2, TextFSM, Ansible).
- **GNS3 instalado**, para realizar todos los laboratorios de simulación.