

# Examen Parcial (Teórica)

## CC312 Administración de Redes

Fecha: 19/10/2024 Duración: 30 minutos Ciclo: 2024-II

Apellidos: Pacheco Taboada Nombres: André Joaquin

1. [1 pto.] Describa qué es una dirección MAC.

Una dirección MAC (Media Access Control) es un identificador único de 48 bits asignado a interfaces de red para comunicaciones en la capa de enlace de datos del modelo OSI. Características clave:

- Formato: 12 dígitos hexadecimales, agrupados en 6 pares (ejemplo: 00:1A:2B:3C:4D:5E)
- Estructura: Primeros 24 bits (OUI) identifican al fabricante, últimos 24 bits son únicos para el dispositivo
- Uso: En protocolos como Ethernet, Wi-Fi, y Bluetooth para identificación de dispositivos en la red local
- Inmutabilidad: Generalmente fija, aunque algunas interfaces permiten su modificación (MAC spoofing)
- Relevancia en redes: Esencial para la comunicación a nivel de enlace y en procesos como ARP (Address Resolution Protocol)

2. [1 pto.] ¿Cómo se pueden activar (triggered)/ detener / controlar los trabajos de Jenkins mediante programación?

Los trabajos de Jenkins pueden gestionarse programáticamente de varias formas:

a) API REST:

```
python
1  import requests
2
3  jenkins_url = "http://jenkins.example.com"
4  job_name = "my-job"
5  token = "my-api-token"
6
7  # Activar un trabajo
8  requests.post(f"{jenkins_url}/job/{job_name}/build", auth=("user", token))
9
10 # Detener un trabajo
11 requests.post(f"{jenkins_url}/job/{job_name}/lastBuild/stop", auth=("user", token))
```

b) Jenkins CLI:

```
java -jar jenkins-cli.jar -s http://jenkins.example.com -auth user:token build my-job
```

c) Groovy script (vía Script Console)

d) Pipeline as Code (Jenkinsfile):

```

pipeline {
    agent any
    stages {
        stage('Trigger another job') {
            steps {
                build job: 'my-other-job', wait: false
            }
        }
    }
}

```

e) Webhooks: Configurar un trigger en Jenkins que responda a eventos HTTP POST.

f) Docker API: Controlar Jenkins desde contenedores Docker. Vía Docker CLI:

```

docker exec -it jenkins-master /bin/bash -c "java -jar /var/jenkins_home/war/WEB-INF/jenkins-cli.jar -s http://localhost:8080/ build my-job"

```

### 3. [1 pto.] Defina qué es un Hypervisor.

Un Hypervisor es una capa de software o firmware que virtualiza recursos de hardware, permitiendo múltiples sistemas operativos (guests) en un único host físico. Características principales:

- Tipos:
    1. Tipo 1 (bare-metal): Ejecutado directamente sobre el hardware (ej. VMware ESXi, Xen)
    2. Tipo 2 (hosted): Ejecutado sobre un sistema operativo (ej. VirtualBox, VMware Workstation)
  - Funciones clave:
    - Aislamiento de VMs
    - Asignación y gestión de recursos (CPU, memoria, almacenamiento, red)
    - Snapshots y migración de VMs
  - Tecnologías de optimización:
    - Virtualización asistida por hardware (Intel VT-x, AMD-V)
    - Paravirtualización
    - Memory ballooning y page sharing
  - Relevancia en redes: Fundamental en la implementación de NFV (Network Function Virtualization) y en la gestión de infraestructuras de red virtualizadas.
4. [1 pto.] ¿Qué es Jenkins y cuál es su uso?

Jenkins es una plataforma de automatización de código abierto para Integración Continua (CI) y Entrega Continua (CD). Características y usos principales:

- Automatización de builds, tests y despliegues
- Integración con sistemas de control de versiones (Git, SVN)
- Extensibilidad mediante plugins
- Pipeline as Code usando Groovy DSL
- Distribución de builds en múltiples agentes
- Soporte para contenedores (Docker) y entornos cloud

Relevancia en administración de red e s:

- Automatización de configuraciones de red
- Despliegue continuo de servicios de red
- Pruebas automáticas de infraestructura

5. [1 pto.] Describa brevemente qué es Flask

Flask es un microframework web para Python, diseñado para ser simple y extensible. Características principales:

- Servidor de desarrollo integrado y depurador
- Enrutamiento de URLs mediante decoradores
- Motor de plantillas Jinja2 integrado
- Soporte para cookies seguras (sesiones del lado del cliente)
- Compatible con WSGI 1.0
- Extensible mediante Flask Extensions

Para el curso de redes nos sirve para:

- Desarrollo rápido de interfaces web para herramientas de gestión de red
- Creación de APIs RESTful para interactuar con dispositivos de red
- Integración fácil con bibliotecas de networking de Python (ej. Netmiko, NAPALM)