

PE01 - Servidor Web LAMP

Peso: 25% de la nota

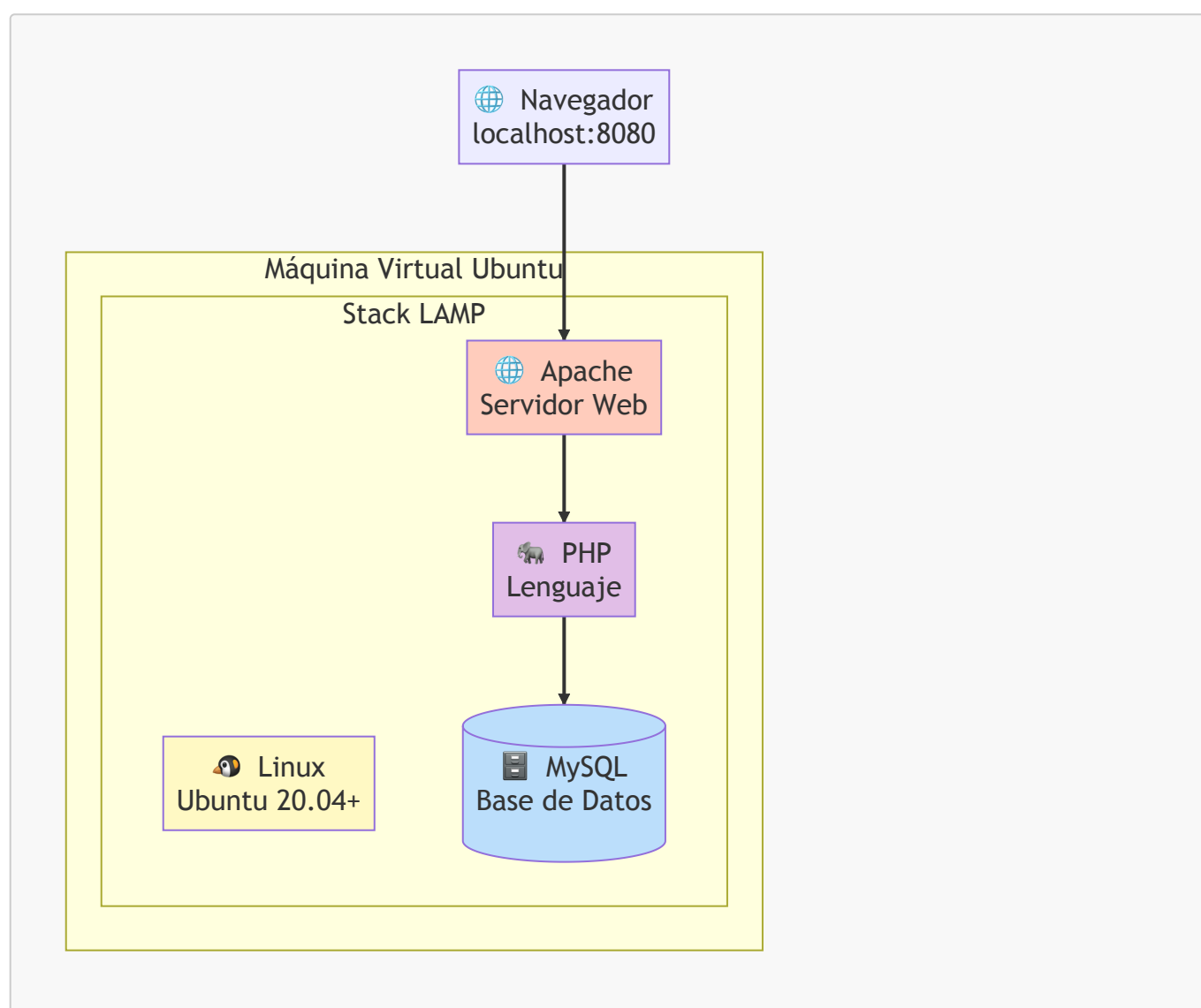
Duración: 2-3 horas

Dificultad: ★ ★ Media

Objetivo

Crear un servidor LAMP (Linux, Apache, MySQL, PHP) completamente funcional y automatizado con Vagrant, que incluya una aplicación web de prueba.

Arquitectura LAMP



Requisitos mínimos (5 puntos)

1. **VM con Ubuntu 20.04 o superior** (0.5 puntos)
2. **Apache instalado y funcionando** (1 punto)
3. **MySQL instalado y configurado** (1 punto)

4. **PHP instalado con extensiones básicas** (0.5 puntos)
 5. **Provisioning automatizado** (2 puntos)
-

Requisitos avanzados (3 puntos)

6. **IP privada fija** (0.5 puntos)
 7. **Puerto HTTP redirigido** (0.5 puntos)
 8. **Base de datos creada con usuario** (1 punto)
 9. **Aplicación web PHP funcionando** (1 punto)
-

Requisitos extras (2 puntos)

10. **Virtual host personalizado** (0.5 puntos)
 11. **Scripts de provisioning separados** (0.5 puntos)
 12. **README completo** (0.5 puntos)
 13. **Página de inicio con phpinfo()** (0.5 puntos)
-

Especificaciones técnicas

Configuración de la VM

```
Hostname: lamp-server
Box: ubuntu/focal64 o superior
RAM: 2048 MB
CPUs: 2
IP privada: 192.168.56.100
Puerto host: 8080 → VM:80
```

Software a instalar

- Apache 2.4
- MySQL 8.0
- PHP 7.4 o superior con extensiones:
 - php-mysql
 - php-curl
 - php-gd
 - php-mbstring
 - php-xml

Base de datos

```
Nombre: lamp_db
Usuario: lamp_user
```

```
Password: lamp_pass
Permisos: ALL en lamp_db
```

Estructura del proyecto

```
PE01-LAMP/
├── Vagrantfile
├── README.md
├── scripts/
│   ├── install-apache.sh
│   ├── install-mysql.sh
│   └── install-php.sh
└── www/
    └── index.php
```

Aplicación web de prueba

Crear un `index.php` que muestre:

1. Título "Servidor LAMP"
2. Información del servidor (hostname, IP, SO)
3. Versiones instaladas (Apache, MySQL, PHP)
4. Conexión a base de datos (exitosa/fallida)
5. Lista de extensiones PHP cargadas

Entregables

1. **Vagrantfile** completo y comentado
2. **Scripts de provisioning** organizados
3. **README.md** con:
 - Descripción del proyecto
 - Requisitos previos
 - Instrucciones de uso
 - Accesos (URLs, credenciales)
 - Capturas de pantalla
4. **Archivo index.php** funcional
5. **(Opcional)** Archivo de configuración del virtual host

Criterios de evaluación

Aspecto	Puntos	Descripción
Funcionalidad	5	Todo funciona correctamente
Automatización	2	Provisioning sin intervención manual

Aspecto	Puntos	Descripción
Organización	1	Código limpio y estructurado
Documentación	1	README completo y claro
Extras	1	Funcionalidades adicionales

Guía de implementación

1. Crear estructura

```
mkdir PE01-LAMP
cd PE01-LAMP
mkdir scripts www
touch Vagrantfile README.md
```

2. Vagrantfile básico

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/focal64"
  config.vm.hostname = "lamp-server"

  # Red
  config.vm.network "private_network", ip: "192.168.56.100"
  config.vm.network "forwarded_port", guest: 80, host: 8080

  # Recursos
  config.vm.provider "virtualbox" do |vb|
    vb.name = "LAMP-Server"
    vb.memory = "2048"
    vb.cpus = 2
  end

  # Carpeta compartida
  config.vm.synced_folder "./www", "/var/www/html",
    create: true,
    owner: "www-data",
    group: "www-data"

  # Provisioning (completar)
  # ...
end
```

3. Script install-apache.sh

```
#!/bin/bash
set -e

echo "=== Instalando Apache ==="
apt-get install -y apache2

# Habilitar módulos
a2enmod rewrite
a2enmod ssl

systemctl enable apache2
systemctl start apache2

echo "Apache instalado correctamente"
```

4. Script install-mysql.sh

```
#!/bin/bash
set -e
export DEBIAN_FRONTEND=noninteractive

echo "=== Instalando MySQL ==="
apt-get install -y mysql-server

echo "=== Configurando MySQL ==="
# Crear base de datos y usuario
mysql <<EOF
CREATE DATABASE IF NOT EXISTS lamp_db;
CREATE USER IF NOT EXISTS 'lamp_user'@'localhost' IDENTIFIED BY
'lamp_pass';
GRANT ALL PRIVILEGES ON lamp_db.* TO 'lamp_user'@'localhost';
FLUSH PRIVILEGES;
EOF

echo "MySQL configurado correctamente"
```

5. index.php de ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Servidor LAMP</title>
  <style>
    body { font-family: Arial; margin: 40px; }
    .success { color: green; }
    .error { color: red; }
    table { border-collapse: collapse; margin: 20px 0; }
```

```

        td, th { border: 1px solid #ddd; padding: 8px; }
    </style>
</head>
<body>
    <h1>Servidor LAMP Funcionando</h1>

    <h2>Información del Servidor</h2>
    <table>
        <tr><td>Hostname</td><td><?php echo gethostname(); ?></td></tr>
        <tr><td>IP</td><td><?php echo $_SERVER['SERVER_ADDR']; ?></td>
    </tr>
        <tr><td>S0</td><td><?php echo php_uname(); ?></td></tr>
    </table>

    <h2>Versiones Software</h2>
    <table>
        <tr><td>Apache</td><td><?php echo apache_get_version(); ?></td>
    </tr>
        <tr><td>PHP</td><td><?php echo phpversion(); ?></td></tr>
        <tr><td>MySQL</td><td>
            <?php
            try {
                $conn = new PDO("mysql:host=localhost", "root", "");
                echo $conn->query('SELECT VERSION()')->fetchColumn();
            } catch (PDOException $e) {
                echo "N/A";
            }
            ?>
        </td></tr>
    </table>

    <h2>Conexión a Base de Datos</h2>
    <?php
    try {
        $conn = new PDO(
            "mysql:host=localhost;dbname=lamp_db",
            "lamp_user",
            "lamp_pass"
        );
        echo '<p class="success">✅ Conexión exitosa a lamp_db</p>';
    } catch (PDOException $e) {
        echo '<p class="error">❌ Error: ' . $e->getMessage() . '</p>';
    }
    ?>

    <h2>Extensiones PHP</h2>
    <p><?php echo implode(', ', get_loaded_extensions()); ?></p>

    <hr>
    <p><a href="info.php">Ver phpinfo() completo</a></p>
</body>
</html>

```

Pruebas

```
# Levantar servidor
vagrant up

# Verificar servicios
vagrant ssh -c "systemctl status apache2"
vagrant ssh -c "systemctl status mysql"

# Probar en navegador
http://localhost:8080

# Probar conexión a BD
vagrant ssh -c "mysql -u lamp_user -plamp_pass lamp_db -e 'SELECT 1;'"
```

Ejemplo de README.md

Servidor LAMP con Vagrant

Servidor Linux + Apache + MySQL + PHP completamente automatizado.

Requisitos

- VirtualBox 7.0+
- Vagrant 2.4+
- 2 GB RAM libres
- 5 GB disco

Instalación

```
\\`\\`\\`bash
git clone <tu-repo>
cd PE01-LAMP
vagrant up
\\`\\`\\`
```

Acceso

- ****Web****: http://localhost:8080
- ****IP privada****: http://192.168.56.100
- ****SSH****: `vagrant ssh`

Credenciales

MySQL

- Usuario: `lamp_user`
- Password: `lamp_pass`
- Base de datos: `lamp_db`

Estructura

- `Vagrantfile`: Configuración de la VM
- `scripts/`: Scripts de provisioning
- `www/`: Contenido web

Capturas

```
! [Captura 1] (screenshots/web.png)
! [Captura 2] (screenshots/phpinfo.png)
```

Consejos

1. Prueba cada script por separado antes de integrarlo
2. Usa `set -e` en bash para detectar errores
3. Comenta tu código
4. Prueba con `vagrant destroy && vagrant up` para verificar reproducibilidad
5. Documenta todos los accesos y credenciales

Rúbrica detallada

Criterio	0 pts	1-2 pts	3-4 pts	5 pts
Provisioning	No funciona	Funciona parcialmente	Funciona con intervención manual	Totalmente automático
Aplicación web	No carga	Muestra errores	Funciona parcialmente	Todo funciona perfectamente
Organización	Desorganizado	Estructura básica	Bien organizado	Excelente estructura
Documentación	Sin README	README básico	README completo	README ejemplar con capturas

Fecha de entrega

[Definir según calendario del curso]

Preguntas frecuentes

P: ¿Puedo usar Debian en lugar de Ubuntu?

R: Sí, pero adapta los comandos de instalación.

P: ¿Es obligatorio separar los scripts?

R: No, pero suma puntos extras.

P: ¿Puedo usar Ansible en lugar de Shell?

R: Sí, se valorará positivamente.

Siguiente práctica: [PE02 - Infraestructura WordPress](#)