



Universidad  
**LATINA**

LAUREATE INTERNATIONAL UNIVERSITIES®

## PROYECTO FINAL

BIT-28 Sistemas Operativos II  
Escuela de Ingeniería en Software  
Periodo: III Cuatrimestre del 2024

**Estudiantes:**

Paulettet Cabal Orellana  
Isabella Lozano Ramos  
Sebastián Velázquez Guillén

**Profesor:**

Carlos Andrés Méndez

## **Tema: Simulación de Redes Virtuales con Mininet.**

Implementación y Configuración de un Entorno de Simulación de Redes utilizando Mininet en Linux

### **Objetivo General:**

- Implementar y configurar un entorno de simulación de redes utilizando Mininet en un sistema operativo Linux para analizar la escalabilidad, disponibilidad, rendimiento del sistema.

### **Objetivos Específicos:**

- Realizar la instalación correcta de Mininet y sus dependencias en un sistema Linux
- Configurar un entorno básico de simulación de red utilizando los componentes de Mininet
- Implementar y evaluar mecanismos de alta disponibilidad para servicios de aplicación.
- Realizar pruebas de escalabilidad horizontal y vertical de servicios.
- Implementar métricas de rendimiento y análisis para servicios de aplicación.
- Documentar el proceso completo de instalación y configuración para futura referencia.

## **Plan del Proyecto**

El desarrollo de este proyecto se llevará a cabo siguiendo un plan estructurado que garantiza una ejecución eficiente, desde la definición de objetivos hasta el análisis y la presentación de resultados obtenidos al configurar la simulación de redes utilizando Mininet en un sistema operativo Linux.

### **Planificación del proyecto:**

#### **1. Definición de objetivos (Semana 12):**

- Identificar y definir los objetivos a tomar en el proyecto.

#### **2. Distribución de responsabilidades (Semana 13):**

- Asignar tareas específicas a cada integrante del equipo, teniendo en cuenta sus habilidades.

#### **3. Implementación de Mininet y sus dependencias (Semana 13):**

- Instalación y configuración inicial exitosa.
- Creación de topologías.
- Ejecución de pruebas en las diferentes topologías.
- Análisis de resultados.

#### **4. Implementación y recopilación de resultados (Semana 14):**

- Crear las topologías de red diseñadas y ejecutar simulaciones en Mininet.
- Recopilar métricas de rendimiento.
- Documentar y analizar los resultados obtenidos.

#### **5. Presentación de resultados (Semana 15):**

- Consolidar los hallazgos y aprendizajes del proyecto en un informe final.
- Preparar una presentación que incluya los resultados obtenidos, lecciones aprendidas y recomendaciones.



## Justificación de la Metodología

Para este proyecto se ha optado por implementar una metodología de investigación aplicada con enfoque experimental y cuantitativo. La investigación aplicada es la más apropiada debido a que el proyecto se centra en la implementación práctica de soluciones tecnológicas y la resolución de problemas específicos en el ámbito de las redes virtuales. El objetivo principal no es generar nuevo conocimiento teórico, sino aplicar el conocimiento existente.

## Enfoque Experimental y Cuantitativo

El componente experimental y cuantitativo constituye un pilar fundamental en nuestra metodología, nos permite desarrollar un estudio del comportamiento de las redes virtuales. A través del enfoque experimental, podemos crear y manipular diferentes topologías de red en un entorno controlado, lo que facilita la realización de pruebas repetibles y la obtención de resultados consistentes. Esto se complementa con una perspectiva cuantitativa que nos permite recopilar información de rendimiento y realizar análisis estadísticos detallados del comportamiento de la red. La combinación de ambos enfoques nos proporciona la capacidad de validar objetivamente el funcionamiento y rendimiento de los servicios de red mediante datos medibles y verificables..

## Fases de la Metodología

### Fase 1: Preparación y Configuración

- Instalación del entorno Linux
- Implementación de Mininet y sus dependencias
- Verificación de la correcta instalación
- Documentación del proceso de configuración

### Fase 2: Diseño Experimental

- Definición de los servicios y aplicaciones a implementar
- Establecimiento de parámetros de medición
- Diseño de casos de prueba

- Definición de criterios de éxito

### Fase 3: Implementación y Pruebas

- Creación de los escenarios de servicios diseñados
- Ejecución de pruebas de rendimiento y disponibilidad
- Medición de métricas de rendimiento
- Documentación de resultados

### Fase 4: Análisis y Evaluación

- Procesamiento de datos recopilados
- Análisis comparativo de resultados
- Evaluación del rendimiento del sistema
- Identificación de áreas de mejora

### Herramientas para la Implementación:

- Mininet : Para simular redes virtuales
- VirtualBox: Para crear máquinas virtuales
- Python: Para automatizar pruebas y análisis de datos
- Herramientas para graficar los resultados

Sistemas operativos ligeros: Alpine Linux o Ubuntu

### Documentación y Referencias:

- Manuales de Mininet
- Artículos sobre sistemas operativos ligeros y métricas de red (IEEE)

## **Artículos Académicos Fundamentales**

### **1. Sobre Mininet y Virtualización de Redes**

Lantz, B., Heller, B., & McKeown, N. (2010). "A network in a laptop: rapid prototyping for software-defined networks." ACM SIGCOMM Workshop on Hot Topics in Networks.

Wang, S. Y., Chou, C. L., & Yang, C. M. (2013). "EstiNet: A network simulator and emulator." IEEE Communications Magazine.

### **2. Implementaciones y Casos de Estudio**

Oliveira, R. L. S., et al. (2014). "Using Mininet for emulation and prototyping Software-Defined Networks." IEEE Colombian Conference on Communications and Computing.

Keti, F., & Askar, S. (2015). "Emulation of Software Defined Networks Using Mininet in Different Simulation Environments." International Conference on Intelligent Systems, Modelling and Simulation.

### **3. Documentación y Guías**

Mininet Walkthrough Commands (<http://mininet.org/walkthrough/>)

Mininet Python API Reference Manual (<http://mininet.org/api/annotated.html>)

GitHub de Mininet (<https://github.com/mininet/mininet>)

### **4. Aspectos de Rendimiento y Escalabilidad**

Kaur, S., Singh, J., & Ghumman, N. S. (2014). "Network programmability using POX controller." International Conference on Communication, Computing & Systems.

Gupta, M., et al. (2013). "Network virtualization and software defined networking for cloud computing: a survey." IEEE Communications Magazine.

### **5. Alta Disponibilidad**

Sharma, S., et al. (2011). "Implementing Quality of Service for the Software Defined Networking enabled future internet." European Workshop on Software Defined Networks.

Lin, P., et al. (2015). "A west-east bridge based SDN inter-domain testbed." IEEE Communications Magazine.

## 6. Métricas y Análisis de Rendimiento

Tootoonchian, A., Ghobadi, M., & Ganjali, Y. (2012). "OpenTM: Traffic matrix estimator for OpenFlow networks." Passive and Active Measurement Conference.

Curtis, A. R., et al. (2011). "DevoFlow: Scaling flow management for high-performance networks." ACM SIGCOMM Computer Communication Review.

## 7. Tendencias Actuales y Futuras

Kreutz, D., et al. (2015). "Software-defined networking: A comprehensive survey." Proceedings of the IEEE.

Nunes, B. A. A., et al. (2014). "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks." IEEE Communications Surveys & Tutorials.



## Marco Teórico

### Introducción a la Simulación de Redes con Mininet

La simulación con Mininet brinda un entorno especializado para probar, entender y analizar topologías de diversas redes, sin necesidad de hardware físico. Esta herramienta es muy útil en escenarios académicos y profesionales, gracias a ellas se puede reducir costos y riesgos asociados con la implementación en entornos reales.

### ¿Qué es Mininet?

#### Mininet Emulador de Redes Virtuales

Mininet es una herramienta diseñada para emular redes completas utilizando namespaces y cgroups de Linux. Este emulador permite la creación de topologías de red realistas en un solo sistema operativo, ofreciendo una plataforma eficiente para evaluar protocolos, servicios y topologías.

#### Características Principales de Mininet

1. **Emulación de Redes Definidas por Software (SDN):** Permite prototipar rápidamente redes SDN mediante la simulación de hosts, switches y enlaces virtuales.
2. **Flexibilidad:** Facilita la creación de entornos personalizados adaptados a necesidades específicas de investigación o a desarrollar.
3. **Escalabilidad:** Ofrece soporte para topologías que van desde redes simples hasta configuraciones complejas.

#### Campos de Uso de Mininet

1. **Investigación académica:**
  - Evaluación de protocolos de red.
  - Desarrollo y prueba de algoritmos de encaminamiento.
2. **Entornos industriales:**
  - Validación de configuraciones antes de implementaciones reales.
  - Pruebas de resiliencia y seguridad.
3. **Educación:**
  - Capacita a estudiantes en diseño y administración de redes.
4. **Optimización de servicios:**
  - Implementación de mecanismos de calidad de servicio (QoS).
  - Pruebas de escalabilidad y alta disponibilidad.

## Configuración de Mininet en Linux

La configuración de Mininet en un entorno Linux requiere:

1. **Instalación del sistema operativo base (Alpine Linux o Ubuntu):** Sistemas ligeros que garantizan un rendimiento óptimo.
2. **Instalación de Mininet y sus dependencias:** Incluyendo controladores SDN y librerías necesarias.
3. **Pruebas iniciales:** Verificar la emulación correcta de topologías básicas.
4. **Automatización:** Uso de Python para generar scripts que gestionen la configuración y pruebas de red.

## Relevancia de Mininet en el Contexto Actual

En un mundo donde las redes virtuales y SDN están en auge, Mininet se posiciona como una herramienta clave para:

- Reducir costos: Evitando la necesidad de hardware físico.
- Acelerar el desarrollo: Facilitando pruebas rápidas de topologías y servicios.
- Fomentar la innovación: Permitiendo explorar nuevas arquitecturas y protocolos sin riesgos.
- Preparar para el futuro: Mejorando la comprensión de entornos virtuales y optimizando el rendimiento

## Conclusión

El uso de herramientas como Mininet en la simulación de redes proporciona una plataforma versátil y accesible para el desarrollo y análisis de infraestructuras de red. Su integración con controladores SDN y su capacidad para emular topologías complejas lo convierten en un recurso invaluable en la investigación y desarrollo de redes virtuales modernas.