### **Trabajo Integrador 1 – Arquitectura y Sistemas Operativos**

* **Tema**: Virtualización
* **Alumnos**: Matías Roda - matute.22@live.com.ar

Matías Rodríguez - xmatiasrcx@gmail.com

* **Materia**: Arquitectura y Sistemas Operativos.
* **Comisión**: Nro. 9
* **Profesor/a**: Osvaldo Falabella.
* **Tutor/a**: Ana Valeria Celerier.
* **Fecha de Entrega**:

# Índice

1. Introducción
2. Marco Teórico
3. Caso Práctico
4. Metodología Utilizada
5. Resultados Obtenidos
6. Conclusiones
7. Bibliografía
8. Anexos

# Introducción

En el presente Trabajo Integrador se abordará la virtualización de sistemas informáticos, entendida como la técnica que permite crear múltiples entornos operativos sobre un único equipo físico (“computadoras dentro de computadoras”). Para ello se empleará VirtualBox, que facilita la creación y ejecución de máquinas virtuales, y Ubuntu como sistema operativo invitado en un entorno Linux. Para el técnico en programación, dominar la virtualización permite el desarrollo y pruebas en un entorno aislado sin comprometer nuestro sistema operativo.

Los objetivos de este trabajo son:

1. Documentar el proceso de instalación y configuración de VirtualBox en Ubuntu, así como la creación y gestión de máquinas virtuales.
2. Evaluar las ventajas de la virtualización en términos de eficiencia, aislamiento y escalabilidad.
3. Crear y ejecutar una aplicación sencilla en Python, demostrando el uso práctico de este entorno virtualizado.

# Marco Teórico

A continuación se exponen de manera concreta los conceptos clave que fundamentan la virtualización de sistemas y las herramientas empleadas (VirtualBox y Ubuntu):

* **Virtualización:** es la técnica que abstrae los recursos físicos de un solo equipo (CPU, memoria, almacenamiento, red) para crear MV “máquinas virtuales” que funcionan como sistemas independientes. Cada MV incluye su propio sistema operativo (“guest”) y aplicaciones, pero comparte el hardware subyacente con el sistema anfitrión (“host”).
* **Sandbox:** Cada MV actúa como un entorno aislado o “sandbox”, de modo que las operaciones realizadas en ella no afectan al host. Este aislamiento garantiza seguridad y reproducibilidad en pruebas y desarrollo.
* **Hipervisor (Hypervisor):** Un hypervisor es un software que permite crear y gestionar MV. Actúa como intermediario entre el hardware físico y los sistemas operativos que se ejecutan en las VM, asegurando que cada VM tenga acceso controlado a los recursos. Hay dos tipos
* **Tipo 1 (bare‐metal):** Corre directamente sobre el hardware, sin SO anfitrión.
* **Tipo 2 (hosteado):** Se ejecuta sobre un sistema operativo existente. Este es el que usaremos para el Trabajo Integrador.
* **Host (Anfitrión):** Equipo físico y sistema operativo principal, sobre el cual se instala el hipervisor.
* **Guest (Invitado):** Máquina virtual con su propio sistema operativo y aplicaciones.
* **VirtualBox:** es un hipervisor de tipo 2, de código abierto y multiplataforma.
* **Ubuntu:** Es una distribución Linux basada en Debian, de código abierto.

Este apartado contiene la fundamentación conceptual del tema tratado. Debe incluir definiciones, clasificaciones, jerarquías, estructuras y sintaxis si corresponde.

* + Incluir gráficos, esquemas o tablas si son útiles para la comprensión.
  + Mencionar autores o documentación oficial si se toman definiciones o explicaciones.

# Caso Práctico

💻 Aquí se debe presentar un problema o situación concreta que haya sido desarrollada o simulada para aplicar el contenido del trabajo.

# Incluye:

* + Breve descripción del problema a resolver.
  + Capturas de pantalla si corresponde.
  + Validación del funcionamiento.

# Metodología Utilizada

 Describe los pasos seguidos durante el desarrollo del trabajo. Se pueden incluir:

* + Investigación previa (fuentes utilizadas).
  + Etapas de diseño y prueba del código.
  + Herramientas y recursos utilizados (IDE, librerías, control de versiones, etc.).
  + Trabajo colaborativo (reparto de tareas en el grupo de trabajo).

# Resultados Obtenidos

📊 Detalla qué se logró con el caso práctico, qué aspectos funcionaron correctamente y qué dificultades se presentaron.

Se pueden incluir:

* + Casos de prueba realizados.
  + Errores corregidos.

# Conclusiones

🎯 Reflexión final del grupo de trabajo. Aquí se recomienda incluir:

* + Qué se aprendió al hacer el trabajo.
  + Posibles mejoras o extensiones futuras.
  + Dificultades que surgieron si las hubo y cómo se resolvieron.

# Bibliografía

📚 Listado de fuentes consultadas, utilizando normas básicas APA u otro formato consistente. Se sugiere incluir:

* + Libros.
  + Documentación oficial.
  + Sitios web con fecha de acceso.
  + Artículos o materiales recomendados por el docente. Ejemplo:
  + Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts. John Wiley & Sons. Capítulo 3, "Procesos".
  + Stallings, W. (2023). Operating Systems: Internals and Design Principles.

# 📥 Recomendaciones para la presentación

* + Formato del archivo: **.pdf**
  + Tipografía: Arial o Calibri, tamaño 11 o 12
  + Interlineado: 1,5
  + Márgenes estándar (2.5 cm)
  + Portada opcional con el logo de la institución (si se requiere)
  + Entrega digital mediante plataforma institucional o por correo