

**2021**

**Docentes**

Alejandro Pazos

Javier Cros

**Alumnos**

Pacheco, Martin - 263651

Sánchez, Ricardo - 255864

Sandoval, Damián - 205106

****

**Docentes**

Sebastián Orrego

Roberto Wagner

Andrés Tarallo

**OBLIGATORIO**

Taller de servidores Linux

**Alumnos**

Pacheco, Martín - 263651

Sánchez, Ricardo - 255864

Sandoval, Damián - 205106

Tabla de Contenidos

[Acerca del documento 2](#_Toc79350408)

[Escenario planteado 2](#_Toc79350409)

[Desarrollo de la solución 3](#_Toc79350410)

[Punto 1 - Preparación de los ambientes 3](#_Toc79350411)

[Punto 2 – Ansible 5](#_Toc79350412)

[2.1 – Creación de repositorio y preparación 5](#_Toc79350413)

[2.2 y 2.3 – Corrección y adaptación de playbooks 6](#_Toc79350414)

[Referencias bibliográficas 9](#_Toc79350415)

[Declaración de autoría 9](#_Toc79350416)

[Aporte Individual 10](#_Toc79350417)

# Acerca del documento

El documento presentado a continuación constituye la entrega final del obligatorio del Taller de Servidores Linux.

# Escenario planteado

Prerequisitos:

Debe tener un equipo bastión (o controlador) con ansible instalado, un usuario para ejecutar los playbooks (recomendamos el usuario ansible) con su clave pública generada.

A) 30%

Instalar un servidor CentOS y otro Ubuntu con 1 GB de memoria RAM y 15 GB de Disco (o más),

Configure la instalación con el siguiente esquema de particionamiento

* Una partición de 1G para /boot
* usando LVM: un volumen lógico de 5GB para /, 4 GB para /var y 3 GB para /home y 2 GB de Swap.

Debe tener 2 interfaces de red, 1 conectada a NAT y la otra a una red Interna o “Solo Anfritrión” que le permita conectarse al equipo bastión con Ansible.

Agregue un usuario ansible, dele permisos con SUDO sin contraseña. Desde el equipo bastión o controlador, copie la clave pública para poder conectarse al servidor.

B) 60 % Tareas a realizar con Ansible

1) Haga un Fork de el repositorio https://github.com/emverdes/obligatorio\_2021\_08. En ese repositorio se encuentra un rol básico para la configuración de un stack LAMP. Agregue un archivo ansible.cfg y un inventario. Estos roles y playbooks están hechos para distribuciones de la familia Red Hat (REDHAT o CENTOS)

2) Corrija los errores que haya en los playbooks

3) Además de Centos o RedHat, los playbooks se deben poder ejecutar también en servidores con UBUNTU o DEBIAN.

C) 10% Documentación

Actualice el archivo README.md explicando los cambios realizados y como utilizar el rol.

Debe documentar cada paso de las tareas realizadas mediante capturas de pantalla. En el caso del playbook debe mostrar la ejecución correcta de todas las tareas.

Agregue la documentación al repositorio. Descargue el repositorio como archivo zip y súbalo a Gestión como entrega.

# Desarrollo de la solución

Punto 1 - Preparación de los ambientes  
  
Con el objetivo de contar con un ambiente que permita hacer despliegue de un stack LAMP en servidores GNU/Linux, lo primero que se debe hacer es prepararlo para asegurar una correcta ejecución de las tareas.

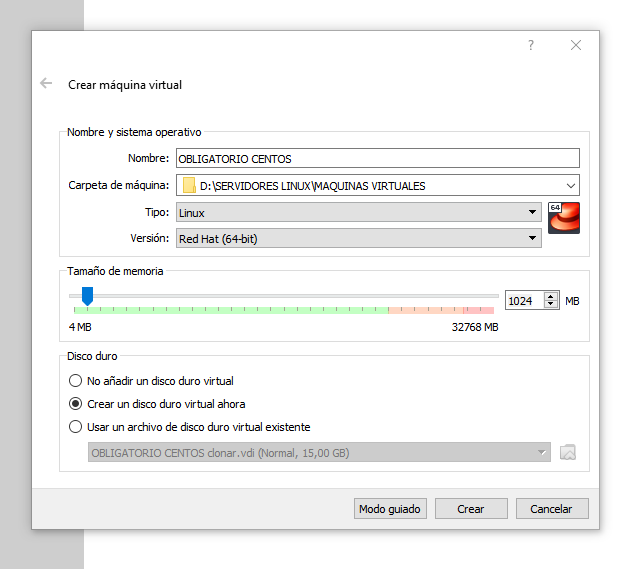
Para realizar el despliegue se necesita contar con un equipo bastión con **ansible** instalado. Para las pruebas realizadas se trabajó con el componente **Windows Subsystem for Linux**, presente como característica extendida de Windows 10. De esta manera se cuenta con un ambiente de terminal de **Ubuntu**.

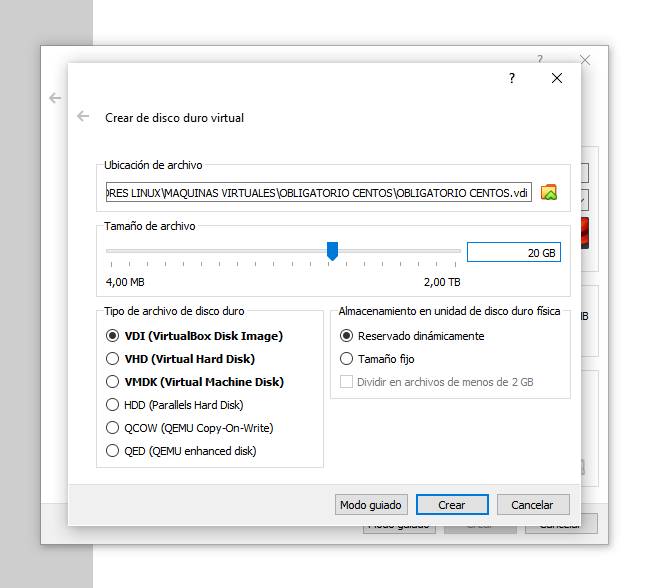
Con el comando sudo apt install ansible se instalan los paquetes y dependencias requeridas para poder utilizar ansible. Se repite el mismo procedimiento para **git** (sudo apt install git).

Luego se crearon dos máquinas virtuales, una corriendo Ubuntu Server 20.04 y la otra con CentOS 8.

Ambos servidores cuentan con dos interfaces de red, una configurada como **NAT** (para navegación a internet) y la otra definida como red **Sólo Anfitrión** (para comunicación con el equipo bastión), y fueron instalados con las siguientes características:

* 1 GB de memoria RAM
* 20 GB de disco duro





Al momento de hacer la instalación de los Sistemas Operativos se creó el siguiente esquema de particionamiento:

**Particiones físicas**

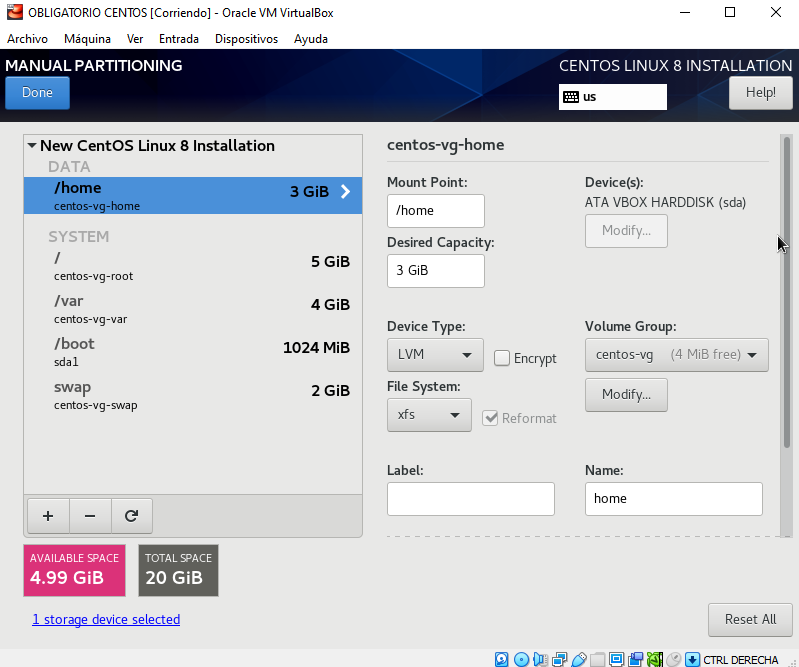
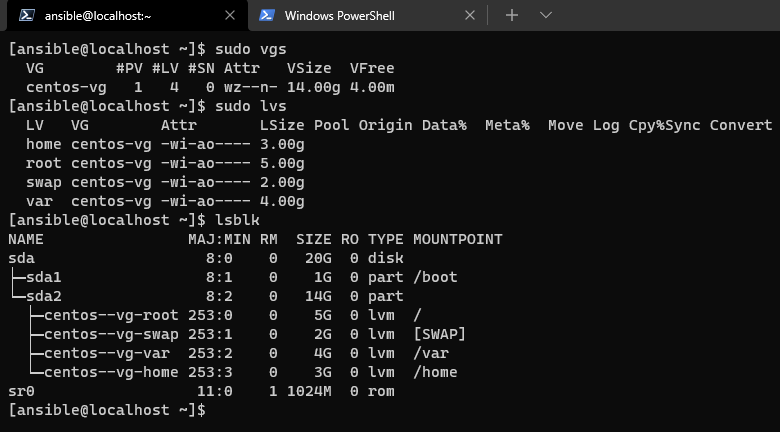
* /boot – 1 GB

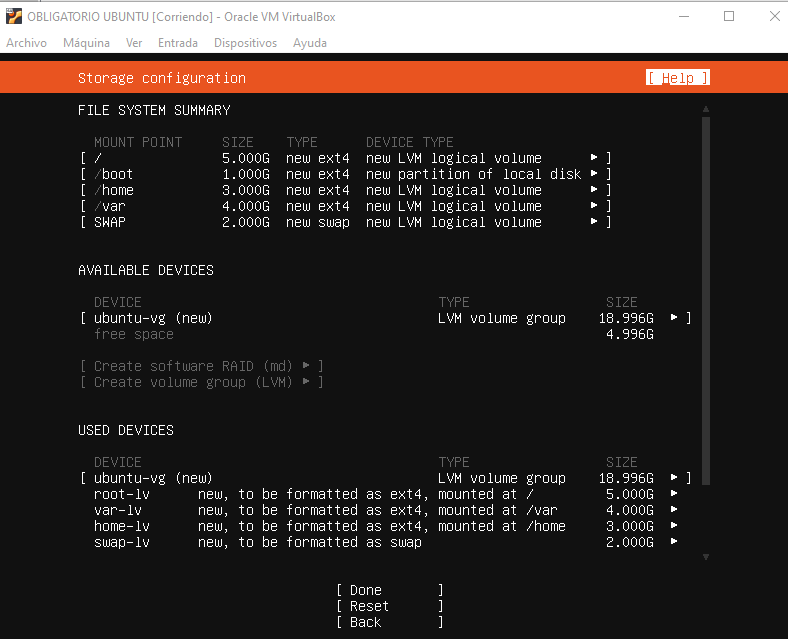
**Volúmenes lógicos**

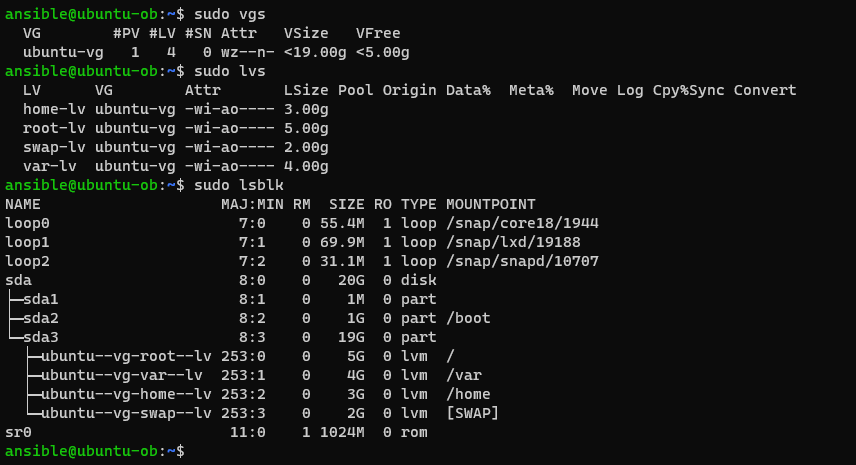
* / (root) – 5 GB
* /var – 4 GB
* /home – 3 GB
* SWAP – 2 GB

Todos los volúmenes y particiones fueron generados en el asistente de instalación de los sistemas. En CentOS 8 la configuración se realiza por medio de interfaz gráfica, y en el caso de **Ubuntu Server** se utiliza la interfaz de texto.

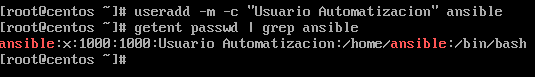
Los volúmenes lógicos fueron creados con **LVM** y pertenecen a un mismo grupo de volúmenes (**Volume Group**).



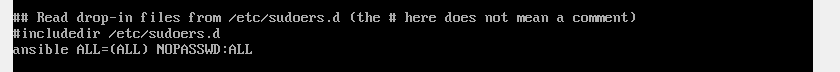




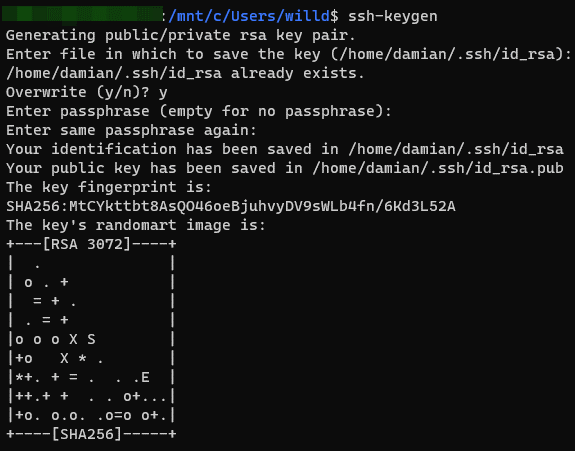
Para estandarizar las tareas a realizar y no utilizar el usuario **root** para la ejecución de tareas, en todos los servidores a configurar se creó el usuario **ansible**:

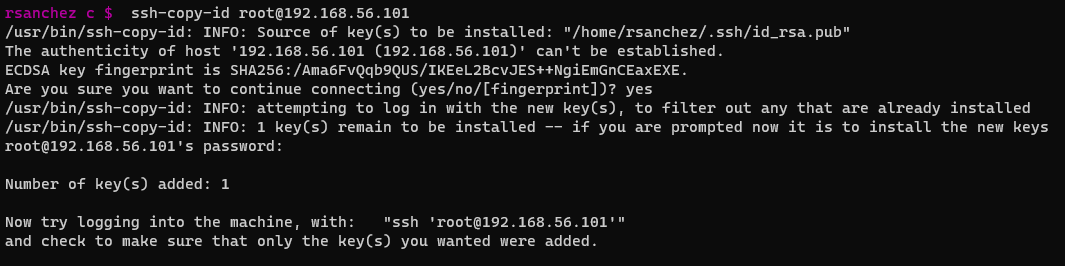


Adicionalmente, se le dieron permisos para elevar privilegios (sudo) sin contraseña. Esto se logra agregando la siguiente línea al final del archivo /etc/sudoers:



Una vez asegurada la conexión entre el bastión y los equipos a configuar, se copian las claves públicas hacia cada uno de los servidores, para poder establecer la conexión vía **SSH:**





De esta forma, se podrá acceder vía SSH a los servidores sin necesidad de indicar la contraseña del usuario **ansible**.

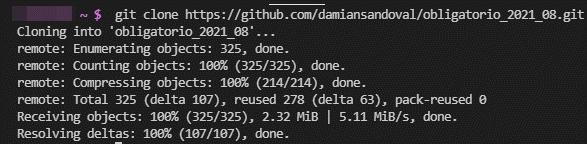
## Punto 2 – Ansible

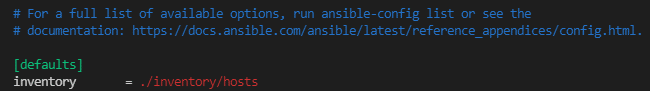
### 2.1 – Creación y preparación del Repositorio

Para poder trabajar de manera colaborativa con el código y contar con un sistema seguro de gestión de versiones se trabajó con la herramienta **git**.

En primer lugar se realizó un fork del repositorio [**https://github.com/emverdes/obligatorio\_2021\_08**](https://github.com/emverdes/obligatorio_2021_08)

Luego se clonó el nuevo repositorio en el equipo bastión y se creó, para la ejecución de los playbooks, un archivo de configuración (**ansible.cfg)** el cual contiene, entre otras configuraciones comunes, la ruta del archivo de inventario:





El archivo de inventario contiene una lista de los servidores donde se realizará la instalación y configuración del stack LAMP:



Para esta ejecución se tomaron en cuenta dos familias de distribuciones GNU/Linux: **RedHat** y **Debian**. Para hacer el despliegue se puede definir en cuáles servidores se realizará la instalación de cada uno de los componentes.

Los playbooks y roles contenidos en este repositorio están en capacidad de dejar operativo el stack tanto en derivados de **RedHat** como en derivados de **Debian**, permitiendo al administrador decidir en cuál distribución desea implementar cuáles componentes, según sus necesidades particulares.

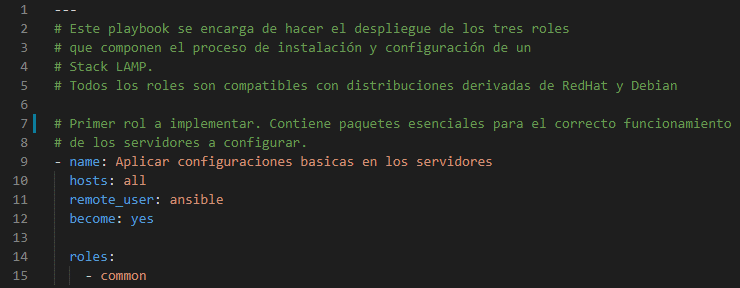
Para ejecutar los playbooks en más equipos, simplemente deberán ser incluidos en el archivo de inventario, bajo el grupo que corresponde a su familia de distribuciones.

### 2.2 y 2.3 – Corrección y adaptación de playbooks

Gracias al repositorio que nos fue provisto contamos con una base para la elaboración de nuestro repositorio. Inicialmente contenía un playbook, llamado **site.yml**, el cual se encarga de hacer el despliegue de tres roles (common, web y db).

Inicialmente el rol **common** era el único que estaba pensado para ser compatible con **RedHat** y **Debian**, mientras que el resto contemplaban sólo uno de los dos.

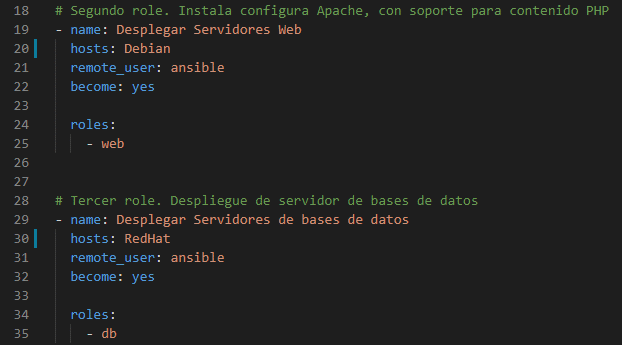
El playbook se encarga de definir, para cada rol, cuáles serán las condiciones de su ejecución:



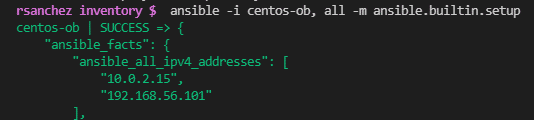
Las configuraciones realizadas en el primer punto del presente documento permiten la ejecución de los playbooks con el usuario **ansible,** elevando privilegios sin necesidad de ingresar su contraseña en el proceso.

El rol **common** contiene paquetes básicos para el correcto funcionamiento de los servidores, por lo que la recomendación es que siempre sea ejecutado en ambas distribuciones.

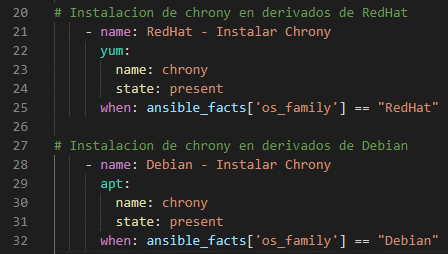
El resto de los roles pueden ser modificados en caso de que se necesite hacer la instalación de cada componente en diferentes distribuciones (servidores web en **Ubuntu** y servidores de Bases de Datos en **CentOS**, por ejemplo):



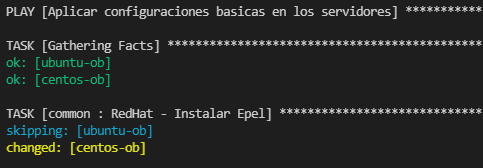
A nivel de roles, lo primero que se debe hacer es prepararlos para que sean compatibles con ambas distribuciones. Esto se puede lograr con el uso de **ansible facts**, una serie de parámetros que obtiene ansible referentes a cada servidor remoto:



Una vez obtenidos los datos necesarios, se optó por usar como criterio el atributo **os\_family**, que indica el tipo de distribución de los servidores (en este caso, **RedHat** y **Debian**):



Esto, en conjunto con el atributo **when,** permite indicarle a una tarea del playbook que se ejecute únicamente en los servidores remotos que cumplan con dicha característica. En los servidores donde no se cumpla dicha condición la tarea no será realizada (**skip**):

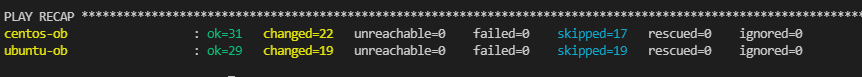


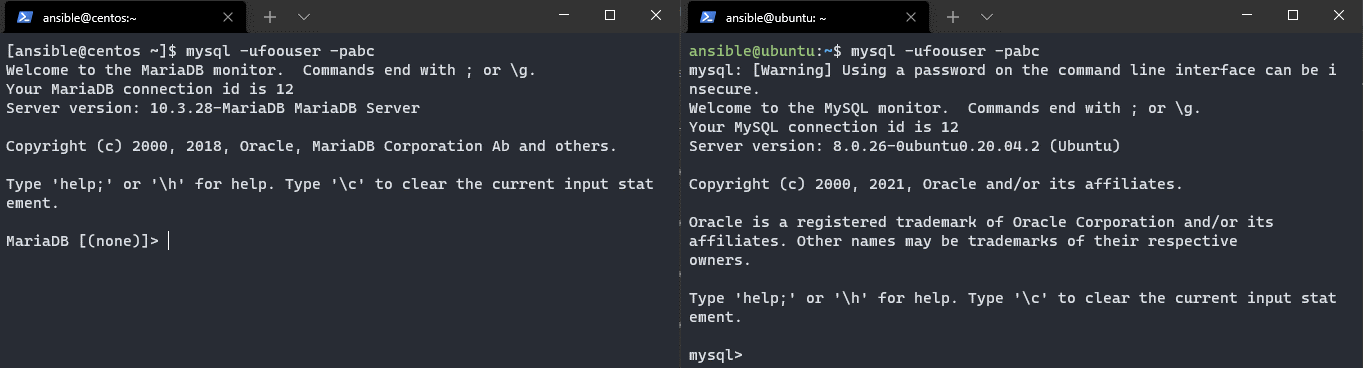
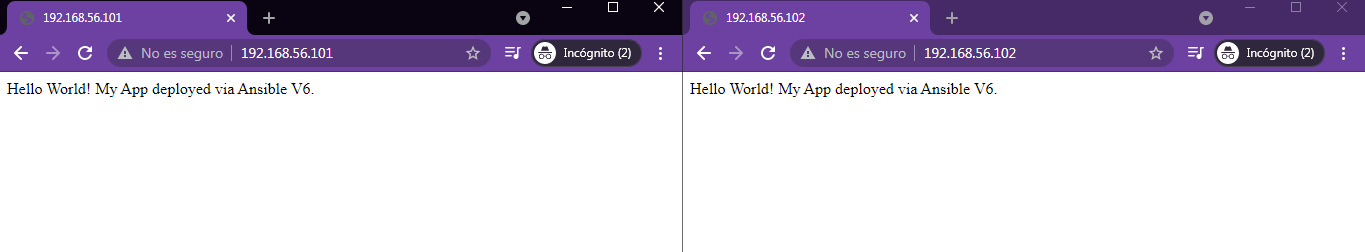
Esta condicional es la que permite que el playbook sea compatible con más de una distribución. En vista de que algunos componentes y servicios cambian de nombre según la distribución, algunas tareas debieron ser duplicadas para que puedan funcionar en cada sistema.

Diferencias principales entre **RedHat** y **Debian**:

|  |  |  |
| --- | --- | --- |
| **Componente** | **Nombre/servicio en RedHat** | **Nombre/servicio en Debian** |
| Manejador de paquetes | dnf / yum | apt |
| Firewall | firewalld | ufw |
| Servidor de Base de Datos | mariadb | mysql |
| Servidor web Apache | httpd | apache2 |

Una vez realizada la ejecución de todos los roles que componen este repositorio, se estará en capacidad de contar con un Servidor web y un Servidor de Bases de Datos totalmente operativos, independientemente de si corren un Sistema Operativo derivado de **RedHat** o **Debian.**



# Referencias bibliográficas

**Documentación oficial ansible**

<https://docs.ansible.com/>

**Documentación oficial git**

https://git-scm.com/doc

# Declaración de autoría

Por la siguiente, Martín Pacheco, Ricardo Sánchez y Damián Sandoval con números de estudiante 263651, 255864 y 205106 respectivamente, estudiantes de la carrera “Analista en Infraestructura Informática” en relación con el trabajo obligatorio de fin de semestre presentado para su evaluación y defensa, declaramos que asumimos la autoría de dicho documento entendida en el sentido de que no se han utilizado fuentes sin citarlas debidamente.

|  |  |  |
| --- | --- | --- |
| Martin Pacheco | Ricardo Sánchez | Damián Sandoval |
|  |  |  |

# Aporte Individual

La implementación de soluciones, pruebas de funcionamiento, evaluación de requerimientos y documentación de lo anterior fueron repartidas equitativamente entre los tres integrantes del proyecto, lo que hace complejo discriminar las tareas realizadas por cada uno.

Debido a la dinámica de trabajo del equipo, se llevaron a cabo varias sesiones de trabajo en conjunto, realizando todas las tareas entre los tres miembros del grupo.

En todo momento se mantuvo una rotación de las tareas, para así garantizar que los tres estuviéramos al tanto de todas las modificaciones realizadas tanto a nivel de documentación como de implementación de la maqueta. De esta forma logramos estar alineados en lo que refiere al diagramado y armado de este proyecto.

Asimismo, todos los cambios realizados a la maqueta eran replicados en las máquinas de todos los integrantes con el fin de garantizar que todos pudiésemos realizar pruebas usando los mismos ambientes.