



UNIVERSIDAD ORT
Uruguay

2021

Taller de
servidores Linux
OBLIGATORIO

ANSIBLE

Docentes

Sebastián Orrego
Roberto Wagner
Andrés Tarallo

Alumnos

Pacheco, Martín - 263651
Sánchez, Ricardo - 255864
Sandoval, Damián - 205106

Tabla de Contenidos

Acerca del documento	2
Escenario planteado	2
Desarrollo de la solución	3
Punto 1 - Preparación de los ambientes	3
Punto 2 – Ansible	6
2.1 – Creación y preparación del Repositorio	6
2.2 y 2.3 – Corrección y adaptación de playbooks	7
Referencias bibliográficas.....	10
Declaración de autoría	11
Aporte Individual	11

Acerca del documento

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

Escenario planteado

Prerrequisitos:

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 Anfitrión” 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 del 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 RedHat (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.

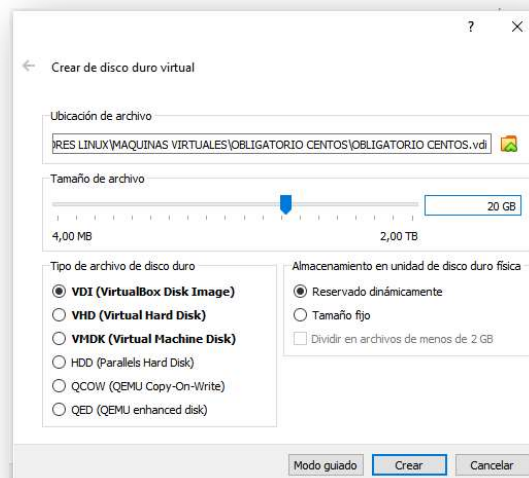
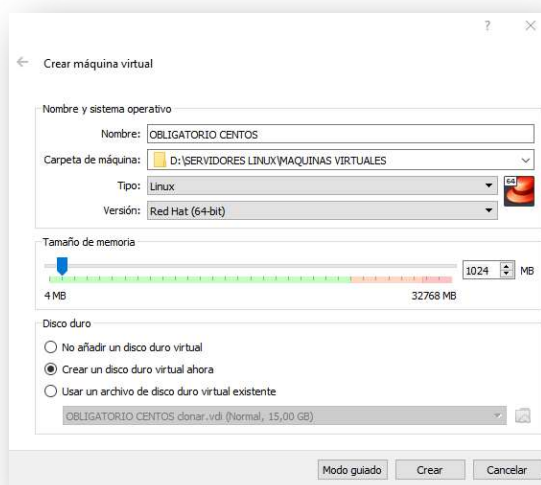
Para realizar el despliegue se necesita contar con un equipo bastión con **ansible** instalado. Para las pruebas realizadas se trabajó con el componente **Windows Subsystem for Linux**, presente como característica extendida de Windows 10. De esta manera se cuenta con un ambiente de terminal de **Ubuntu**.

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

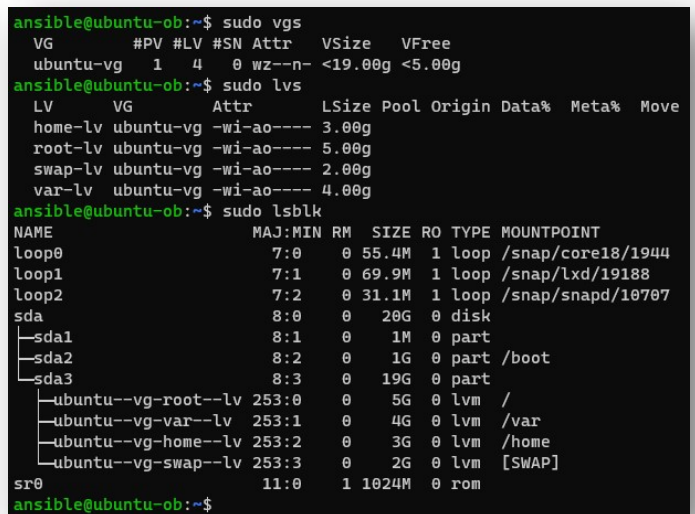
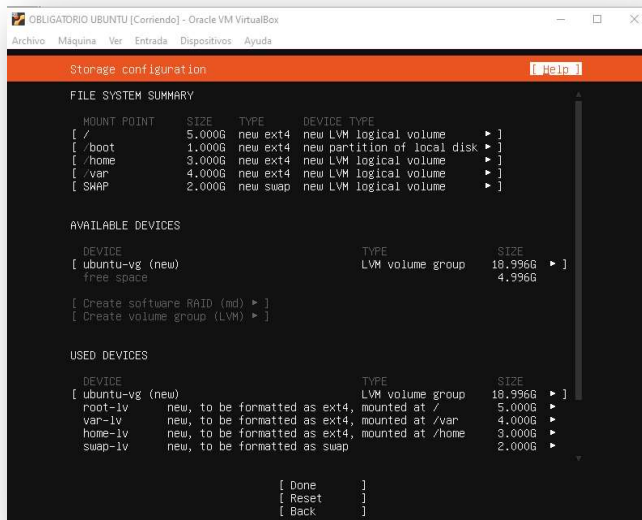
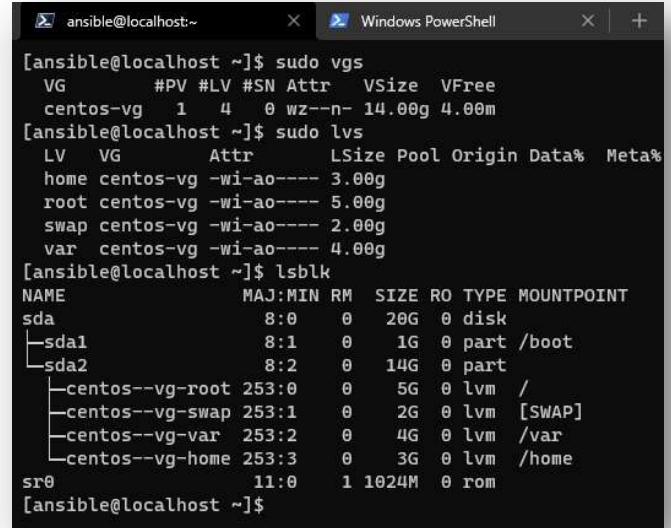
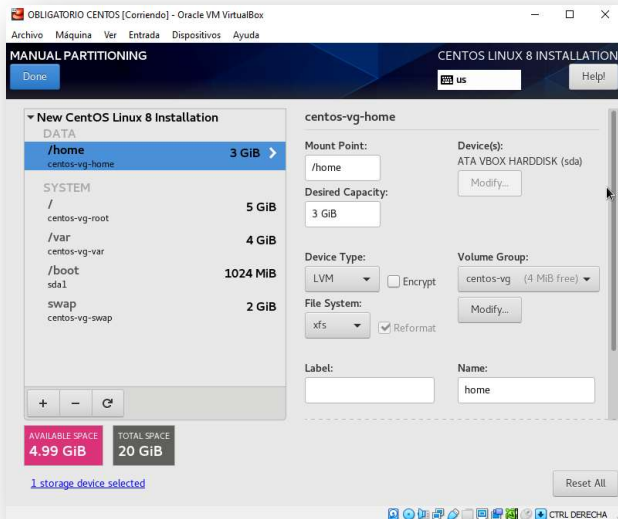
- /boot – 1 GB

Volúmenes lógicos

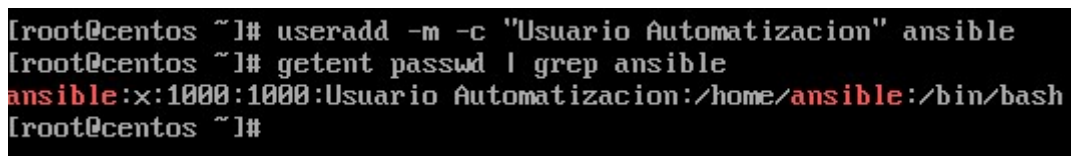
- / (root) – 5 GB
- /var – 4 GB
- /home – 3 GB
- SWAP – 2 GB

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:

```
## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
ansible ALL=(ALL) NOPASSWD:ALL
```

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

```
/mnt/c/Users/willd$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/damian/.ssh/id_rsa):
/home/damian/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/damian/.ssh/id_rsa
Your public key has been saved in /home/damian/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:MtCYkttbt8AsQ046oeBjuhvyDV9sWLB4fn/6Kd3L52A
The key's randomart image is:
+---[RSA 3072]-----+
|
| o . +
| = + .
| . = +
|o o o X S
|+o X * .
|*+. + = . . .E
|+++. + . . o+...|
|+o. o.o. .o=o o+.|
+---[SHA256]-----+
```

```
rsanchez c $ ssh-copy-id root@192.168.56.101
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/rsanchez/.ssh/id_rsa.pub"
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ECDSA key fingerprint is SHA256:/Ama6FvQqb9QUS/IKEeL2BcvJES++NgiEmGnCEaxEXE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@192.168.56.101's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@192.168.56.101'"
and check to make sure that only the key(s) you wanted were added.
```

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

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:

```
~ $ git clone https://github.com/damiansandoval/obligatorio_2021_08.git
Cloning into 'obligatorio_2021_08'...
remote: Enumerating objects: 325, done.
remote: Counting objects: 100% (325/325), done.
remote: Compressing objects: 100% (214/214), done.
remote: Total 325 (delta 107), reused 278 (delta 63), pack-reused 0
Receiving objects: 100% (325/325), 2.32 MiB | 5.11 MiB/s, done.
Resolving deltas: 100% (107/107), done.
```

```
# For a full list of available options, run ansible-config list or see the
# documentation: https://docs.ansible.com/ansible/latest/reference_appendices/config.html.

[defaults]
inventory = ./inventory/hosts
```

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

```
[RedHat]
centos-ob

[Debian]
ubuntu-ob
```

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

A continuación, se detallan los cambios realizados sobre el código original forkeado de [emverdes](#)

- En todas tareas se cambió el usuario de ejecución de **root** a **ansible**
- Se agrego el parametro `become: yes` para permitir al usuario ansible tener privilegios elevados.
- Para los playbooks **web** y **db** se modificó el parámetro `hosts` para que se ejecuten en todos los equipos que contenga el inventario, independientemente de su distribución (derivados **RedHat** o **Debian**)
- Originalmente el inventario tenía 2 grupos, uno para **webserver**s y otro para **dbserver**s. Se cambio por **RedHat** y **Debian**, para así correr los playbook indistinto de la distribución de linux
- El rol **common** se quitaron los tags
- Se cambia el servicio **NTP** por **Chrony**
- Se hace uso del `when: ansible_facts['os_family'] == "DISTRO"` para así poder tener bloques de código que se ejecuten en una distro en particular.
- En el caso de **RedHat** se agregó el repositorio **Epel**
- Para **Debian** se optó por instalar **MySQL** mientras que en **RedHat** se utilizó **MariaDB**
- Se agrega el componente **pip** para instalar las dependencias necesarias para la conexión entre python, php y las bases de datos
- eliminación de usuario **anónimo** en las bases de datos.
- Se unifican varios playbooks de servidor web en uno solo
- Configuraciones de Selinux y Firewall para que tanto el servidor web como la base de datos funcionen correctamente
- Se modifico la estructura de carpetas del proyecto
- Se agrega un `ansible.cfg` en la raíz del proyecto
- Se agrega un documento de entrega en el obligatorio
- Se modifica los playbooks para que tomen variables de entorno de ejecución desde una carpeta, indicando que solo tome de ahí los archivos con extensión **yml**

```
- name: Incluir variables
  include_vars:
    dir: ../vars/
  extensions:
    - 'yml'
```

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):


```

18 # Segundo role. Instala configura Apache, con soporte para contenido PHP
19 - name: Desplegar Servidores Web
20   hosts: Debian
21   remote_user: ansible
22   become: yes
23
24   roles:
25     - web
26
27
28 # Tercer role. Despliegue de servidor de bases de datos
29 - name: Desplegar Servidores de bases de datos
30   hosts: RedHat
31   remote_user: ansible
32   become: yes
33
34   roles:
35     - db

```

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 referente a cada servidor remoto:

```

rsanchez inventory $ ansible -i centos-ob, all -m ansible.builtin.setup
centos-ob | SUCCESS => {
  "ansible_facts": {
    "ansible_all_ipv4_addresses": [
      "10.0.2.15",
      "192.168.56.101"
    ],

```

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**):

```

20 # Instalacion de chrony en derivados de RedHat
21 - name: RedHat - Instalar Chrony
22   yum:
23     name: chrony
24     state: present
25     when: ansible_facts['os_family'] == "RedHat"
26
27 # Instalacion de chrony en derivados de Debian
28 - name: Debian - Instalar Chrony
29   apt:
30     name: chrony
31     state: present
32     when: ansible_facts['os_family'] == "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**):

```
PLAY [Aplicar configuraciones basicas en los servidores] *****

TASK [Gathering Facts] *****
ok: [ubuntu-ob]
ok: [centos-ob]

TASK [common : RedHat - Instalar Epel] *****
skipping: [ubuntu-ob]
changed: [centos-ob]
```

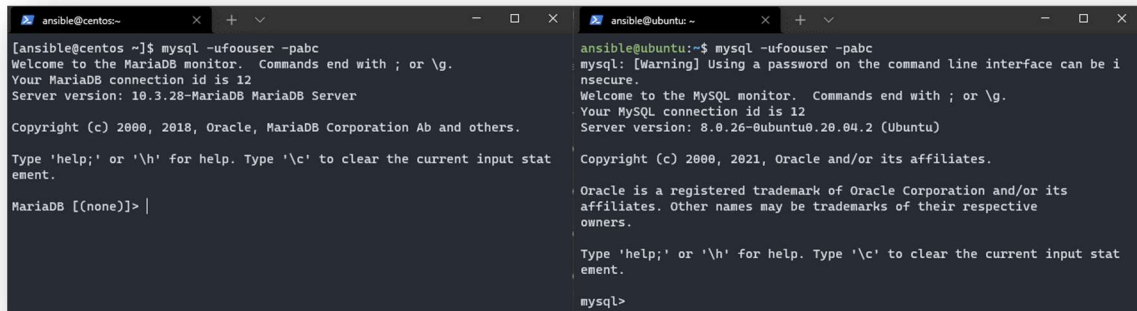
Esta condicional es la que permite que el playbook sea compatible con más de una distribución. En vista de que algunos componentes y servicios cambian de nombre según la distribución, algunas tareas debieron ser duplicadas para que puedan funcionar en cada sistema.

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**.

```
PLAY RECAP *****
centos-ob      : ok=31  changed=22  unreachable=0  failed=0  skipped=17  rescued=0  ignored=0
ubuntu-ob     : ok=29  changed=19  unreachable=0  failed=0  skipped=19  rescued=0  ignored=0
```



The image shows two terminal windows side-by-side. The left window is titled 'ansible@centos:~' and shows the command 'mysql -ufooouser -pabc' being executed. It displays the MariaDB monitor interface with version 10.3.28. The right window is titled 'ansible@ubuntu:~' and shows the command 'mysql -ufooouser -pabc'. It displays the MySQL monitor interface with version 8.0.26-0ubuntu0.20.04.2, including a warning about using a password on the command line.

```
ansible@centos:~]$ mysql -ufooouser -pabc
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 10.3.28-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> |

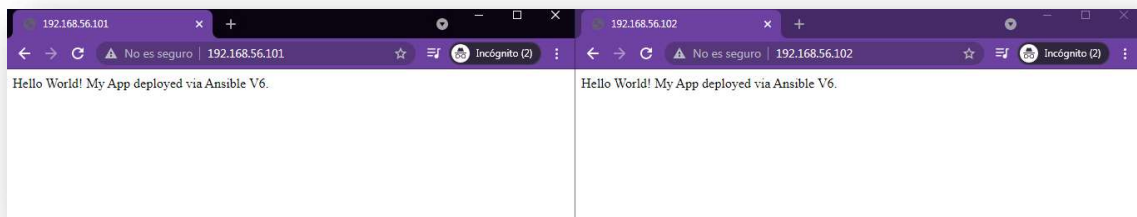
ansible@ubuntu:~$ mysql -ufooouser -pabc
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.26-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```



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 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.