

Anexo	Lista de cotejo	Modo:	Ordinario
Materia:	Automatización de Infraestructura Digital	Unidad:	Unidad I.- Entornos de desarrollo en la automatización de redes.

Objetivo

El estudiante implementará un entorno de desarrollo para la automatización de redes.

Criterios

Criterios	Saber	Hacer
ESTRUCTURA CORRECTA	1.0	0.0
PROCEDIMIENTO CORRECTO	0.0	5.0
IDENTIFICACIÓN DE CONCEPTOS DE UNIDAD	3.0	0.0
SER	0.0	1.0

Instrucciones

El presente instrumento consta de dos actividades:

1. Archivo README en forma de reporte
2. Configuración de una aplicación.

Actividad 1, Archivo README en forma de reporte (Total 5 pts.):

Elaborar y presentar un archivo **README** en forma de reporte sobre el procedimiento de instalación y configuración de las herramientas de automatización de redes que incluya:

- Descripción de las herramientas de desarrollo.
- Evidencia clara de instalación de las herramientas.
- Lista de verificación del funcionamiento del entorno de desarrollo.
- Anexo con los recursos utilizados de la comunidad de colaboración.

Instrucciones:

- Introducción. - Indica el contenido del reporte en un mínimo de media cuartilla y máxima una cuartilla. Mínimo 200 palabras.
- Desarrollo. - Es la parte más grande del informe donde se plasmará todo lo que se hizo durante la unidad.
 - Descripción de las herramientas utilizadas para automatización: **Saber 1.5 puntos.**
 - Docker Engine
 - Docker Compose.
 - Docker Swagger.

- Procedimiento de instalación: **Saber 1.5 puntos.**
 - Instalación técnica de herramientas necesarias (VSCode, Plugins, etc.).
 - Instalación técnica de docker.
 - Instalación técnica de Git.
- Evidencia de pruebas de verificación de funcionamiento. **Saber Hacer 1.0 puntos.**
 - Ejecutar la imagen “hello-world” para verificar el funcionamiento de docker.
 - Ejecutar un archivo “.YML” para verificar el funcionamiento de contenedores.
- Conclusión - Cada estudiante indicando el nombre creará un resumen de los puntos abordados en el trabajo, donde se exponen los resultados y se destacan los hallazgos más importantes, en un mínimo de media cuartilla. **Saber 0.5 puntos.**
- Bibliografía estilo APA7 - La bibliografía empieza en una página nueva inmediatamente después del texto, sigue con la numeración de páginas del documento, es preciso justificar el texto y poner sangría francesa. Entre las citas bibliográficas no se deja ninguna línea en medio. **Saber 0.5 puntos.**

Nota:

Subir el archivo README a la cuenta de GitHub.

A continuación se muestra un ejemplo:

```

1  # Commandos básico y utiles para el curso
2
3  ## Índice
4
5  1. [**Docker**](#docker)
6  |   - [Docker Compose](#docker-compose)
7
8
9  ### Docker Compose
10
11 - **Iniciar contenedores definidos en un
   archivo `docker-compose.yml`:**
12 |   ```shell
13 |   docker-compose up
14 |   ```
15
16 - **Ver logs de servicios en Docker Compose:**
17 |   ```shell
18 |   docker-compose logs servicio
19 |   ```

```

Commandos básico y utiles para el curso

Índice

1. **Docker**
 - Docker Compose

Docker Compose

- **Iniciar contenedores definidos en un archivo `docker-compose.yml`:**

```
docker-compose up
```

- **Ver logs de servicios en Docker Compose:**

```
docker-compose logs servicio
```

Actividad 2 - Configuración de una aplicación (Total 4 pts.):

El estudiante en base a un caso de estudio proporcionado por el profesor(a) deberá realizar un proyecto para generar el despliegue de una aplicación utilizando docker (compose y/o swarm):

Instrucciones:

- A partir de los siguientes recursos (<https://bit.ly/3tLNqMy>) realizar lo siguiente:
- Crear un archivo YML con el siguiente nombre: **stack-iniciales-alumno.yml**
- Agregar y mostrar el proceso de la creación del servicio de PhpMyAdmin utilizando la imagen oficial de Docker Hub (<https://hub.docker.com/>) de acuerdo a la siguiente configuración: **Saber Hacer 1.0 puntos**

PHPMyAdmin

- **image:** phpmyadmin latest
- **name:** phpmyadmin
- **External Port:** 9090
- **Internal Port:** 80
- **Environment:**
 - **PMA_ARBITRARY=1**

- Agregar y mostrar el proceso de la creación del servicio de MySQL Server utilizando la imagen oficial de Docker Hub (<https://hub.docker.com/>) de acuerdo a la siguiente configuración: **Saber Hacer 1.0 puntos**

MySQL

- **Image:** mysql v8.0
- **name:** mysql-db
- **External Port:** 3306
- **Internal Port:** 3306
- **volumes:** dbfiles
- **Environment:**
 - **Root Password:** qwerty
 - **Database:** sistema_ventas_db
 - **User:** admin
 - **Password:** admin

- Crear el archivo **Dockerfile** para configurar el servicio backend
 - Usar la versión de **node 18.17.1-alpine**
 - Crear un directorio en **/usr/src/app**
 - Copiar los archivos **package.json** y **package-lock.json** al directorio raíz **./**
 - Instalar las dependencias: Ejecutar el comando **npm install**
 - Copiar todo el directorio del proyecto al contenedor.
 - Exponer el puerto 3000
 - Agregar el comando **CMD** [**"node"**, **"index.js"**]
- A partir del archivo **Dockerfile** creado para el backend se debe construir el contenedor de acuerdo a la siguiente configuración: **Saber Hacer 1.0 puntos**

Backend

- Image: Node V18.17.1-alpine
- name: sistema-ventas-back
- External Port: 8080
- Internal Port: 3000
- Depends on: mysql-db

- Crear el archivo **Dockerfile** para configurar el servicio frontend
 - Usar la versión de nginx:stable-alpine.
 - Crear un volumen con el nombre /temp
 - Eliminar de la imagen el contenido de la carpeta **/usr/share/nginx/html/***
 - Copiar el contenido de la configuración de **nginx.conf**, **mime.types** y el directorio **sistema-ventas-front** a sus respectivas carpetas.
 - Exponer el puerto 80.
 - Ejecutar el comando: **CMD["nginx", "-g", "daemon off;"]**
- A partir del archivo **Dockerfile** creado para el frontend se debe construir el contenedor de acuerdo a la siguiente configuración: **Saber Hacer 1.0 puntos**

Frontend

- WebServer: Nginx latest
- Name: sistema-ventas-front
- Ports: 80
- depends_on: sistema-ventas-back

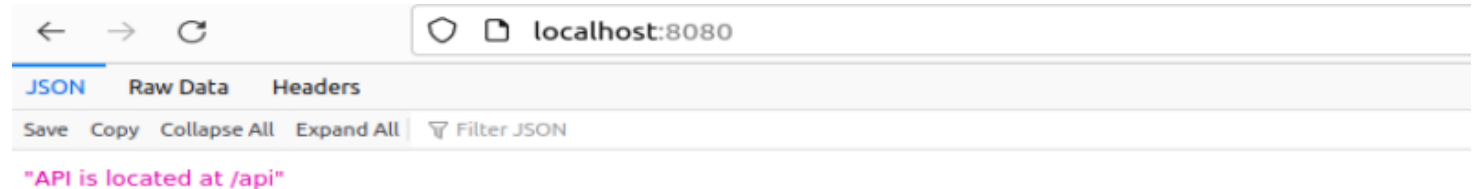
El resultado final debe ser el siguiente:

Frontend:

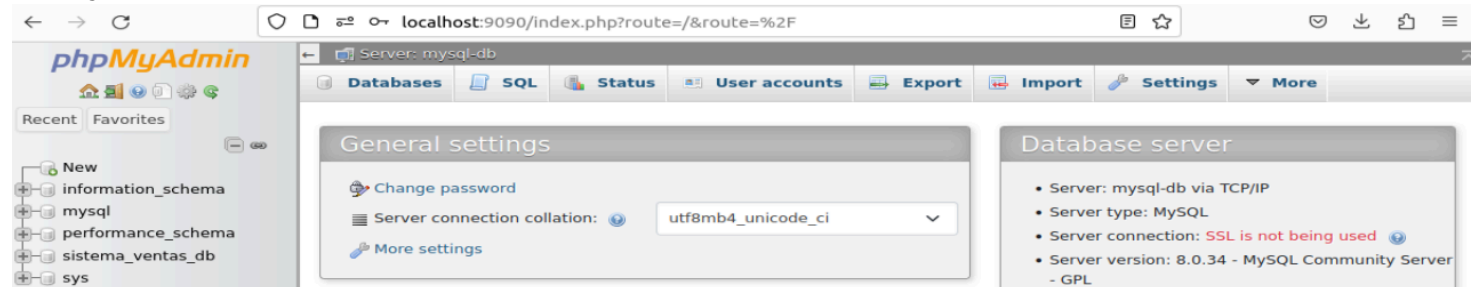
- usuario: **admin**
- Contraseña: **12345678**



Backend:



PhpMyAdmin



Especificaciones del archivo README

Entregar un archivo README en la plataforma (**elección del profesor(a)**) con las siguientes características:

- Portada: Nombre y logo de la Institución, Carrera, Grupo, Unidad, Nombre, Núm. Control (cada integrante) y fecha.
- Títulos
- SubTítulos
- Interlineado 1.15 y justificado.
- Subir el archivo a GitHub.

Ser - VALOR 1.0 PTS.
<ul style="list-style-type: none">• Asistencias totales a clase.• Portafolio de evidencias.
Fecha de Entrega
La entrega será especificado por cada facilitador(a)

Referencias

Bell, Peter. *Introducing GitHub: A Non-technical Guide*. O'Reilly Media. Accessed 4 June 2025.

Gift, Noah, et al. *Python for DevOps: Learn Ruthlessly Effective Automation*. O'Reilly Media, 2019. Accessed 4 June 2025.

Hillar, Gaston C. *Building RESTful Python Web Services*. Packt Publishing, Limited, 2016. Accessed 4 June 2025.

Jackson, Chris, et al. *Cisco Certified DevNet Associate DEVASC 200-901 Official Cert Guide*. Cisco Press, 2020. Accessed 4 June 2025.

Lenz, Moritz. *Python Continuous Integration and Delivery: A Concise Guide with Examples*. Apress, 2018. Accessed 4 June 2025.

Tsitoara, Mariot. *Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer*. Apress, 2019. Accessed 4 June 2025.