

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Web Client and Backend Development Frameworks.

P1 : Descarga, creación y ejecución de imagen Docker de la página del
minisitio de nuevo ingreso de ESCOM

Profesor: M. en C. José Asunción Enríquez Zárate

Alumna: Mendez Cruz Ivonne

ivonne.mendezw@gmail.com

7CM1

March 6, 2025

Contents

1	Introducción	1
2	Conceptos	1
3	Desarrollo	2
3.0.1	Descargar Docker Desktop	2
3.0.2	Descarga de la pagina web	3
3.0.3	Ejecución del servidor Web NGINX	4
3.0.4	Creación del Dockfile	5
3.0.5	Creación de la imagen	5
4	Resultados	6
5	Conclusión	7
6	Referencias Bibliográficas	9

List of Figures

1	Descargar de Docker Desktop	2
2	Instalación de Docker Desktop	2
3	Configurar el Docker Desktop	3
4	Pagina de nuevo ingreso	3
5	Pagina web ya descargada	4
6	Ejecutando el servidor	4
7	Verificando que el servidor funcione	4
8	Servidor listo	4
9	Sintáxis del Dockerfile	5
10	Lugar del Dockerfile	5
11	Creando la imagen	5
12	Ejecutando el contenedor	5
13	Verificando en Dockerfile	6
14	Resultado final	6
15	Resultado final 2	6
16	Resultado final 3	7
17	Resultado final 4	7

1 Introducción

En el ámbito del desarrollo y despliegue de aplicaciones web, Docker se ha convertido en una herramienta fundamental para la creación de entornos portátiles y reproducibles. Esta práctica tiene como objetivo comprender el proceso de descarga, creación y ejecución de una imagen Docker para el minisitio de nuevo ingreso de ESCOM, permitiendo así la implementación de un contenedor que facilite su distribución y ejecución en cualquier entorno compatible con Docker.

A lo largo de la práctica, se abordarán los pasos esenciales para la gestión de imágenes y contenedores Docker, incluyendo la descarga de la imagen base adecuada, la configuración del entorno mediante un Dockerfile, la construcción de la imagen personalizada y su posterior ejecución. De este modo, se busca desarrollar habilidades prácticas en la administración de contenedores y fortalecer la comprensión de la virtualización a nivel de sistema operativo.

Esta práctica resulta especialmente útil en el contexto de la ingeniería de software y la administración de servidores, ya que el uso de Docker permite un despliegue ágil y eficiente de aplicaciones sin depender de configuraciones específicas del sistema operativo anfitrión.

2 Conceptos

1. Docker: Docker es una plataforma de código abierto que permite desarrollar, enviar y ejecutar aplicaciones dentro de contenedores. Los contenedores son entornos ligeros, portátiles y aislados que incluyen todo lo necesario para ejecutar una aplicación, como el código, las bibliotecas y las dependencias.
2. Imagen Docker: Una imagen en Docker es una plantilla inmutable que contiene todo lo necesario para ejecutar una aplicación. Es una especie de "instantánea" de un sistema de archivos y se utiliza como base para crear contenedores.
3. Contenedor Docker: Un contenedor es una instancia en ejecución de una imagen Docker. Es un entorno aislado que ejecuta una aplicación con todas sus dependencias sin afectar el sistema operativo anfitrión.
4. Dockerfile: Es un archivo de texto que contiene un conjunto de instrucciones para construir una imagen Docker. Define la base del sistema operativo, las aplicaciones y las configuraciones necesarias para la imagen.
5. Docker Hub: Es un servicio de almacenamiento en la nube donde los desarrolladores pueden encontrar, compartir y gestionar imágenes Docker. Contiene imágenes oficiales de software como Nginx, MySQL y más.
6. docker build: Es un comando que se utiliza para construir una imagen a partir de un Dockerfile. Ejemplo: `docker build -t nombre-imagen .` Donde `-t` asigna un nombre a la imagen y `.` indica que el Dockerfile está en el directorio actual.
7. docker run: Es el comando que permite crear y ejecutar un contenedor basado en una imagen. Ejemplo: `docker run -d -p 8080:80 nombre-imagen` Donde `-d` ejecuta el contenedor en segundo plano y `-p 8080:80` mapea el puerto 80 del contenedor al puerto 8080 del host.
8. docker ps: Muestra una lista de los contenedores en ejecución. Para ver todos los contenedores, incluidos los detenidos, se usa: `docker ps -a`
9. docker stop: Este comando se usa para detener un contenedor en ejecución. Ejemplo: `docker stop id contenedor`
10. docker rm : Elimina un contenedor detenido. Ejemplo: `docker rm id contenedor`
11. docker images: Muestra todas las imágenes disponibles en el sistema.
12. docker rmi: Permite eliminar una imagen de Docker. Ejemplo: `docker rmi nombre-imagen`
13. Volumen en Docker: Un volumen en Docker es un método de almacenamiento persistente que permite compartir datos entre contenedores o entre el contenedor y el host. Se usa con `-v` en el comando `docker run`.
14. Red en Docker: Docker permite configurar redes virtuales para que los contenedores se comuniquen entre sí o con el mundo exterior. Se pueden crear redes con `docker network create` y asignarlas a los contenedores con `--network`.
15. Puerto en Docker: Especifica cómo se exponen los servicios de un contenedor al exterior. Se usa `-p` para mapear puertos, como en `-p 8080:80`, donde el puerto 80 del contenedor se enlaza al 8080 del host.

3 Desarrollo

3.0.1 Descargar Docker Desktop

1. Haz clic en el botón "Download for Windows" o "Download for Mac", dependiendo de tu sistema operativo. Si usas Linux, puedes seguir el tutorial específico para tu distribución. El archivo se descargará automáticamente. Si no se inicia la descarga, intenta hacer clic nuevamente en el enlace.

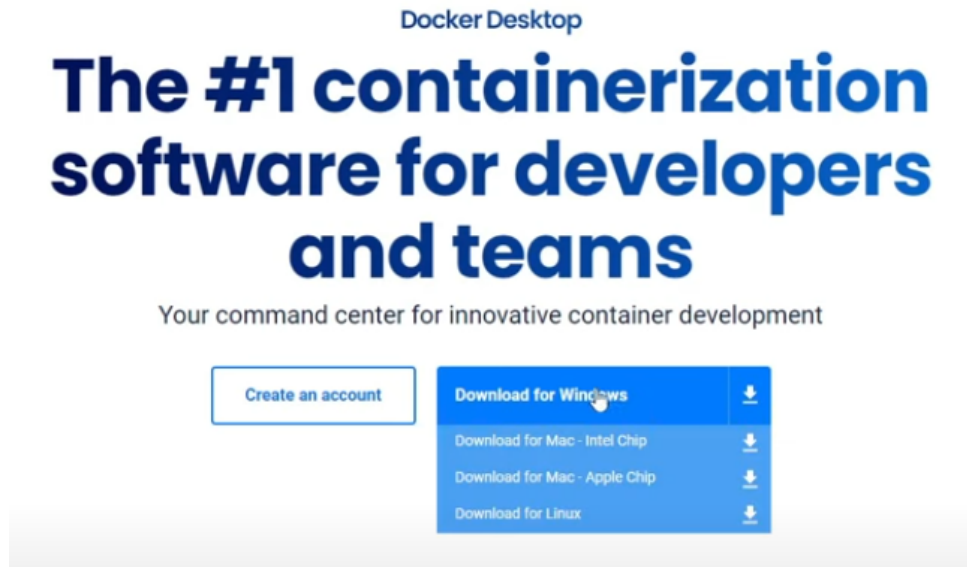


Figure 1: Descargar de Docker Desktop

2. Durante la instalación, el asistente te guiará paso a paso. Asegúrate de que la opción "Install required components" (componentes necesarios) esté seleccionada, como WSL 2 (Windows Subsystem for Linux) si estás en Windows. Si no tienes WSL 2, Docker lo instalará automáticamente.

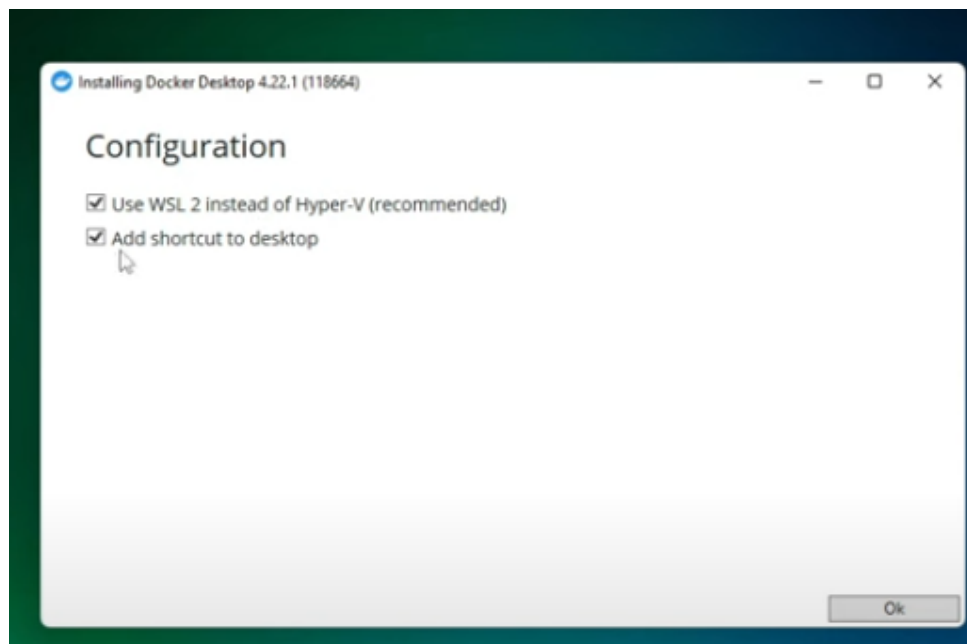


Figure 2: Instalación de Docker Desktop

3. Una vez instalado, busca "Docker Desktop" en tu computadora y abre la aplicación. Si tienes una cuenta de Docker, puedes iniciar sesión. Si no, puedes usar Docker sin necesidad de iniciar sesión.

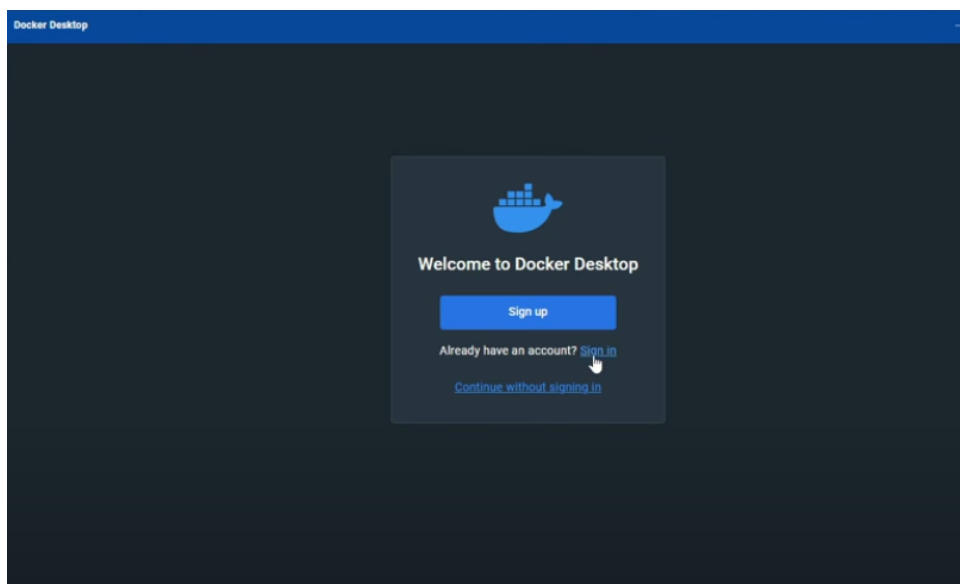


Figure 3: Configurar el Docker Desktop

Ahora Docker Desktop está instalado y listo para ser usado.

3.0.2 Descarga de la pagina web

Lo primero que haremos será acceder a la página web que vamos a utilizar. Para ello, abrimos nuestro navegador de preferencia y en la barra de direcciones, escribimos la URL del sitio web. Una vez cargada la página, buscamos la sección de descargas o la opción que nos permita obtener el archivo que necesitamos.

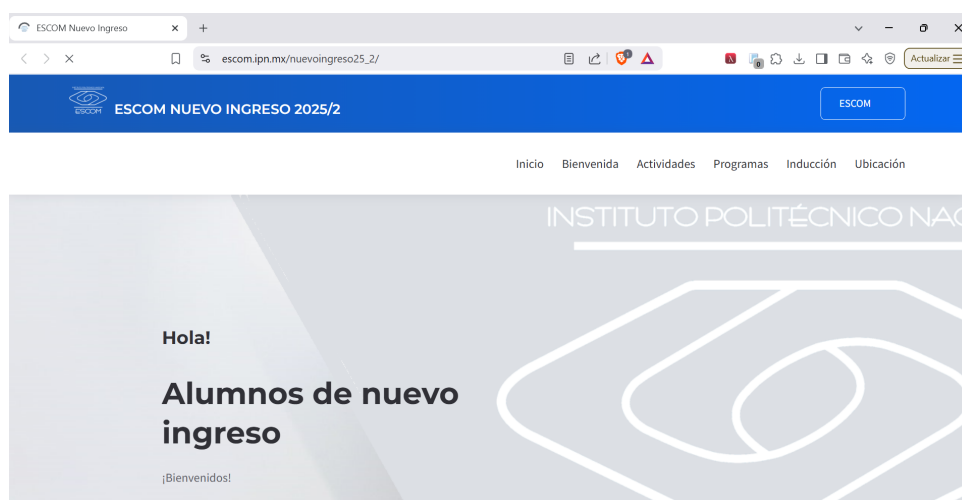


Figure 4: Pagina de nuevo ingreso

Hacemos clic en el enlace correspondiente para iniciar la descarga. Después de hacer clic, nuestro navegador comenzará a descargar el archivo. Es importante asegurarnos de que la descarga se realice correctamente, por lo que debemos verificar que el archivo aparezca en la carpeta de descargas de nuestro sistema. Finalmente, una vez descargado, podremos acceder al archivo desde nuestra computadora.

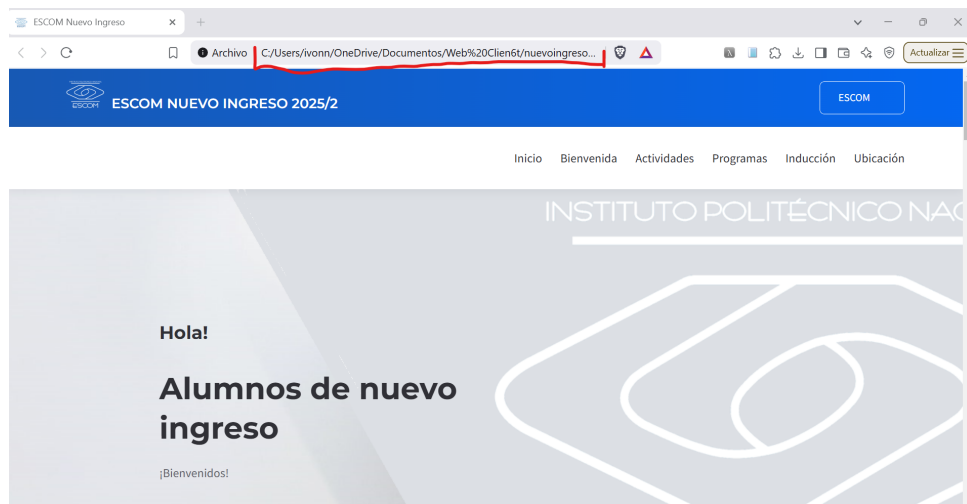


Figure 5: Pagina web ya descargada

3.0.3 Ejecución del servidor Web NGINX

1. Ejecutamos un contenedor a partir de la imagen con docker container run, le asignamos un nombre y publicamos el puerto 80 en el anfitrión:

```
PS> docker container run --name my-nginx -p 80:80 nginx
```

Figure 6: Ejecutando el servidor

Abrimos una nueva consola y comprobamos que está en ejecutándose:

```
PS C:\Users\ivonn> docker container ls
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                    NAMES
194ecd927d7b   mi-sitio-web  "/docker-entrypoint..."  3 hours ago   Up 3 hours   0.0.0.0:8080->80/tcp      mi-web
PS C:\Users\ivonn>
```

Figure 7: Verificando que el servidor funcione

Abrimos el navegador <http://127.0.0.1/> y podemos comprobar que efectivamente se está ejecutando el servidor Web:

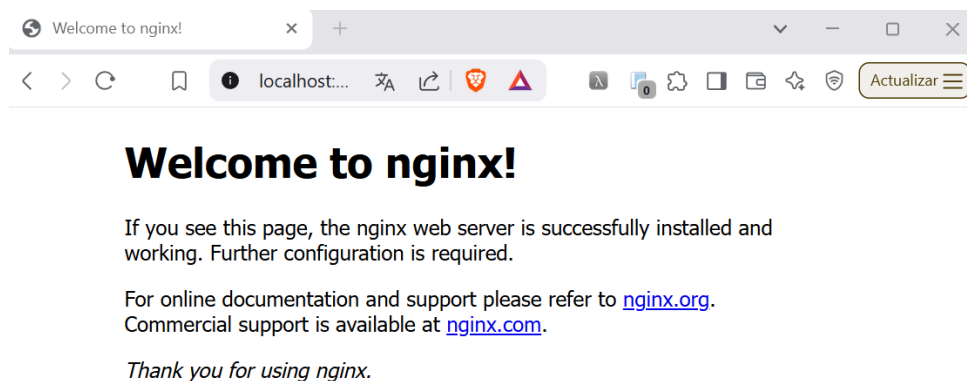


Figure 8: Servidor listo

3.0.4 Creación del Dockerfile

Fichero Dockerfile para crear la imagen:

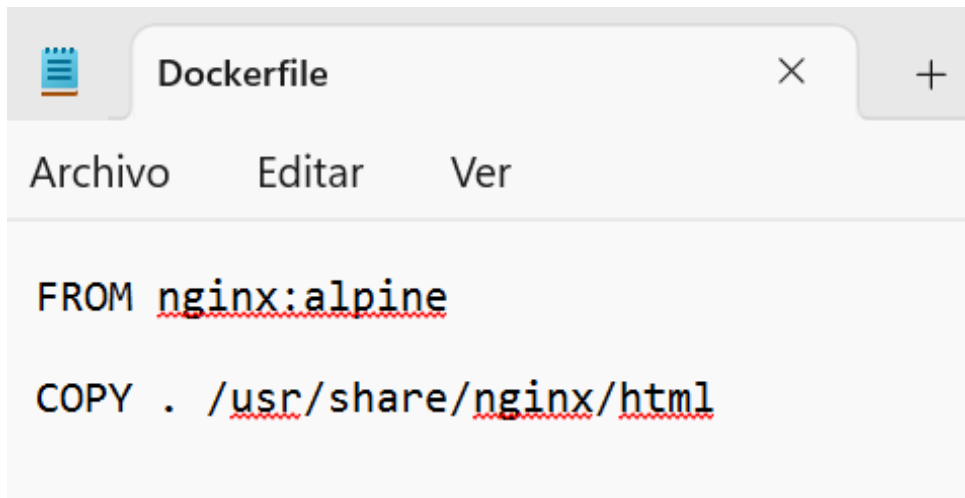


Figure 9: Sintaxis del Dockerfile

Este archivo irá junto a la carpeta que contiene nuestra pagina web y sus assets.

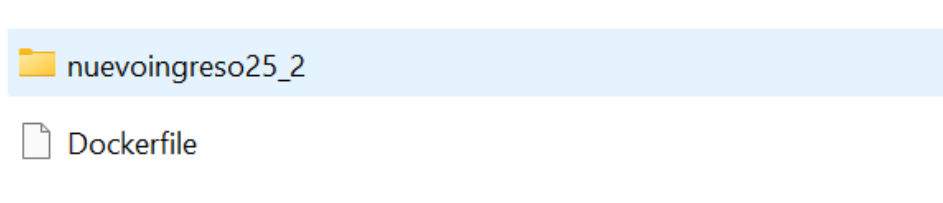


Figure 10: Lugar del Dockerfile

3.0.5 Creación de la imagen

Este comando se ejecuta en PowerShell (PS¿ indica que estás en PowerShell) y se usa para construir una imagen Docker a partir de un Dockerfile ubicado en el directorio actual (.).

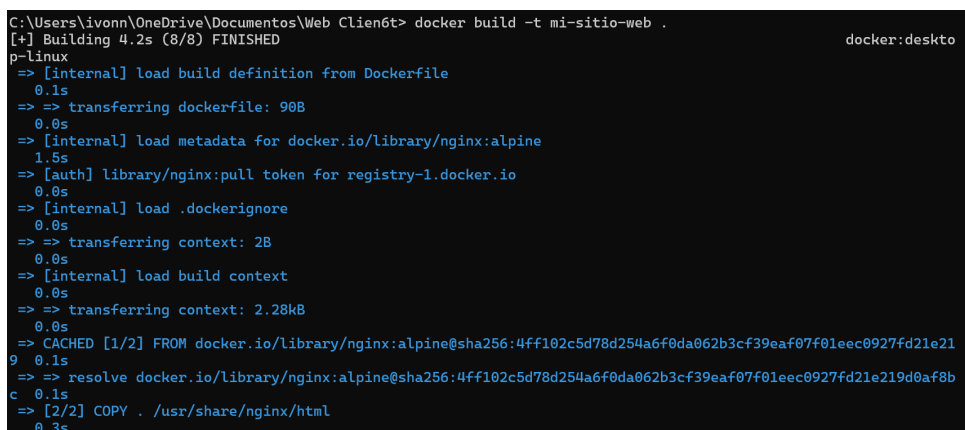


Figure 11: Creando la imagen

Una vez creada nuestra imagen, ahora, ejecutamos un contenedor basado en la imagen creada y mapea el puerto 80 del contenedor al 8080 del host:

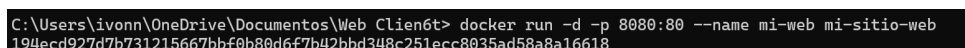


Figure 12: Ejecutando el contenedor

Podemos igual visualizar en nuestro Docker Desktop que nuestro contenedor está activo y que ha sido creado con éxito

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last start	Actions
<input type="checkbox"/>	my-web-contain	86df9e4c0425	my-website	8080:80	0%	13 days a	
<input type="checkbox"/>	my-nginx	9b9e938ca6cc	nginx	80:80	0%	5 hours a	
<input type="checkbox"/>	simple-nginx-rur	69d3e3828048	webserver	8080:80	0%	3 hours a	
<input type="checkbox"/>	mi-web	194ecd927d7b	mi-sitio-we	8080:80	0%	3 hours a	

Figure 13: Verificando en Dockerfile

4 Resultados

Abrimos el navegador y vamos a <http://localhost:8080/nuevoingreso252/index.html>

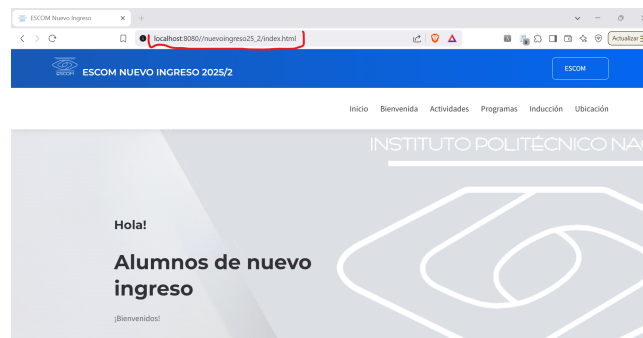


Figure 14: Resultado final

Finalmente tenemos nuestra imagen del sitio web de nuevo ingreso.



Figure 15: Resultado final 2

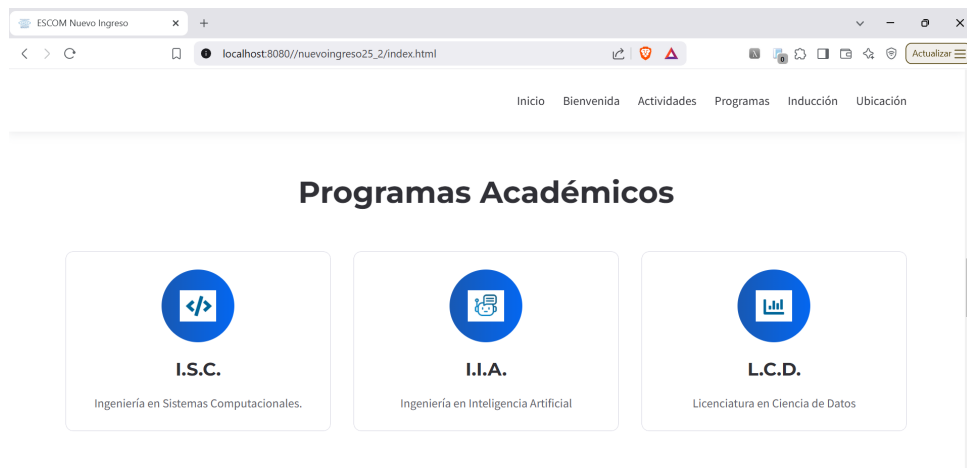


Figure 16: Resultado final 3



Figure 17: Resultado final 4

5 Conclusión

Después de realizar esta práctica sobre la descarga, creación y ejecución de imágenes Docker para desplegar la página del minisitio de nuevo ingreso de ESCOM, puedo decir que he adquirido un conocimiento más profundo sobre el funcionamiento y la utilidad de Docker en el desarrollo y despliegue de aplicaciones.

Al comenzar, comprendí la importancia de Docker como una herramienta que permite empaquetar aplicaciones y sus dependencias en contenedores, garantizando que el software se ejecute de manera consistente en diferentes entornos. Esto es especialmente útil en proyectos que requieren despliegue en servidores o en la nube, ya que evita problemas de compatibilidad entre sistemas operativos y configuraciones locales.

Durante la práctica, seguí los pasos para instalar Docker Desktop, lo cual me permitió familiarizarme con la interfaz y los conceptos básicos, como imágenes, contenedores y volúmenes. Posteriormente, construí una imagen personalizada basada en Nginx, utilizando un Dockerfile que incluía los archivos del minisitio. Esta etapa me ayudó a comprender el proceso de construcción de imágenes y la importancia de definir correctamente las instrucciones dentro del Dockerfile.

Uno de los aspectos más interesantes de esta práctica fue la ejecución del contenedor con la imagen creada y la asignación de puertos para poder acceder a la página web desde el navegador. Al ejecutar el comando `docker run -d -p 8080:80 simple-nginx`, aprendí cómo exponer servicios y hacerlos accesibles externamente. Sin embargo, también me enfrenté a problemas, como el error de asignación de puertos ya en uso, lo que me obligó a investigar soluciones y aprender a listar y detener contenedores en ejecución con `docker ps` y `docker stop`.

Además, entendí la importancia de la reutilización de imágenes y cómo optimizar la construcción de contenedores para que sean más ligeros y eficientes. También pude notar que Docker no solo facilita el despliegue, sino que mejora la gestión y escalabilidad de aplicaciones, ya que permite crear múltiples instancias de un mismo servicio de forma rápida y sencilla.

En conclusión, esta práctica me permitió reforzar mis conocimientos sobre Docker y su aplicación en el desarrollo web. Considero que dominar esta herramienta es clave para cualquier desarrollador, ya que facilita enormemente la administración de entornos y la portabilidad de aplicaciones.

Mendez Cruz Ivonne

6 Referencias Bibliográficas

References

- [1] Docker Hub Container Image Library — App Containerization. Accedido el 4 de marzo de 2025. [En línea]. Disponible: https://hub.docker.com/_/nginx
- [2] “Dockerfile reference”. Docker Documentation. Accedido el 4 de marzo de 2025. [En línea]. Disponible: <https://docs.docker.com/reference/dockerfile/>