



EXAMEN UNIDAD I CREAR UN MAPA WEB CON DOCKER

I. INSTALAR DOCKER DESKTOP

1.1. Comprobación de Requisitos.

Lo primero que haremos será verificar la versión del Sistema Operativo que tenemos instalado, el cual debe ser de 64 bits. Para esto podemos dar clic derecho sobre Equipo y seleccionaremos la opción Propiedades, al inicio de la ventana que se desplegará podremos comprobar la versión del Windows (Figura 1).

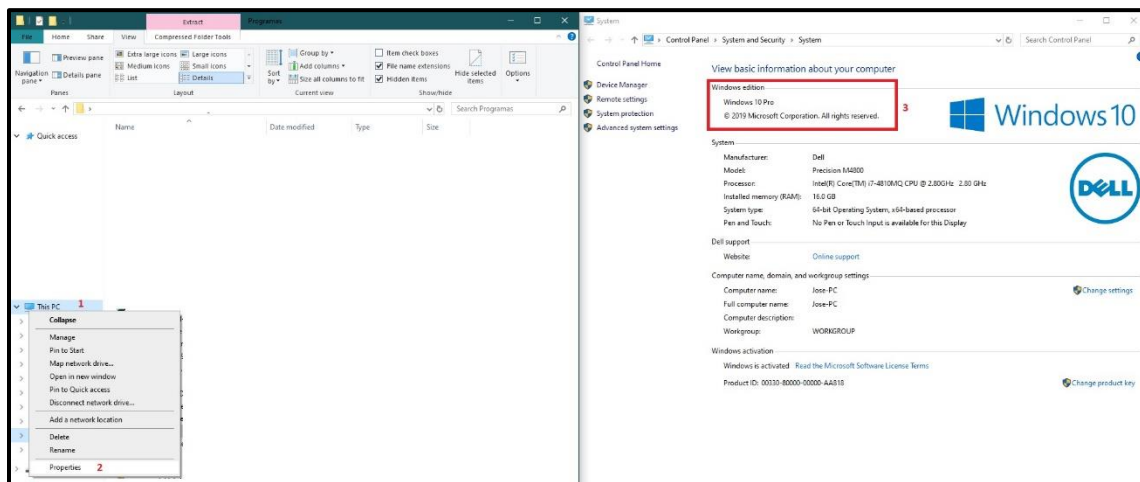


Figura 1. Verificar la versión del Sistema Operativo

Adicionalmente, deberá asegurarse de que su sistema Windows sea compatible con la tecnología de virtualización de hardware y que la virtualización esté habilitada. Para esto debe abrir el Administrador de tareas de Windows, al cual puede acceder desde la combinación de teclas Ctrl+Alt+Supr, o escribiendo el nombre en la barra de búsqueda. Una vez dentro del Administrador de Tareas, navegar a la pestaña Rendimiento y colocarse en la opción CPU, ahí podrá verificar si la virtualización está activada en su Sistema Operativo (Figura 2).

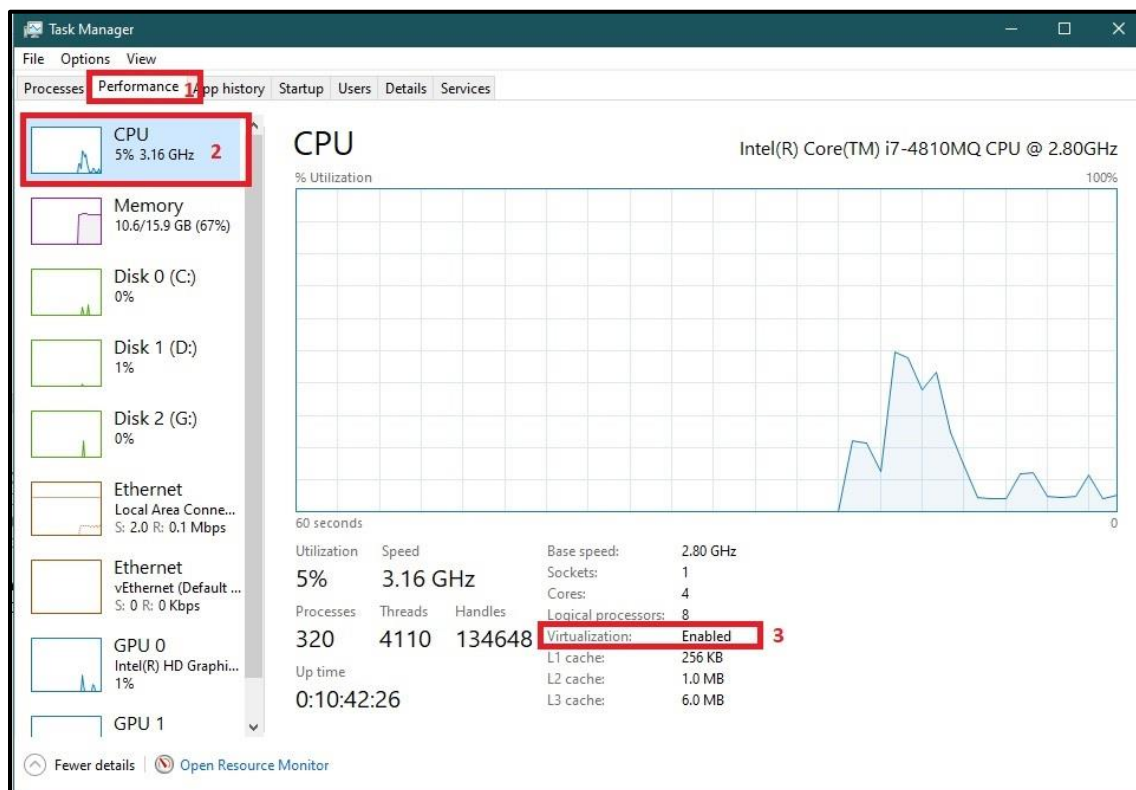


Figura 2. Verificar opción de virtualización

Si la virtualización aparece desactivada, deberá activarla mediante el BIOS de la máquina, y debido a que esto es diferente según el modelo de cada computadora, le sugiero hacer una búsqueda en Google usando el texto “Como activar virtualización” seguido de la marca y/o modelo de su computadora.

Si necesita ayuda en este apartado puede contactarme directamente.

1.1.1. Instalación de Docker en Windows

Docker Desktop está disponible para versiones de Windows Pro y Windows Home (build 19043 o mayor).

Para consultar la versión de su sistema operativo puede presionar las teclas *win+R* y luego escribir en la ventana el comando *winver*, se mostrará algo similar a lo siguiente, donde observará la versión (build) del SO:



Una vez comprobados los requerimientos, procederá a descargar el archivo de instalación desde [este enlace](#).

Una vez descargado en su computadora, dar doble clic al archivo “Docker Desktop Installer.exe”.

Siga las instrucciones del asistente de instalación para aceptar la licencia, autorizar el instalador y continuar con la instalación.

Cuando se le solicite, autorice al instalador de escritorio de Docker con la contraseña de su sistema durante el proceso de instalación. Se necesita acceso privilegiado para instalar componentes de red, enlaces a las aplicaciones Docker y administrar las máquinas virtuales.

Cuando se le solicite, asegúrese de que la opción Habilitar características de Windows de Hyper-V o Instalar los componentes de Windows necesarios para WSL 2 esté seleccionada en la página Configuración.

Haga clic en Finalizar en el cuadro de diálogo de configuración completa e inicie la aplicación Docker Desktop.

Docker Desktop no se inicia automáticamente después de la instalación. Para iniciar Docker Desktop, busque Docker y seleccione Docker Desktop en los resultados de búsqueda.

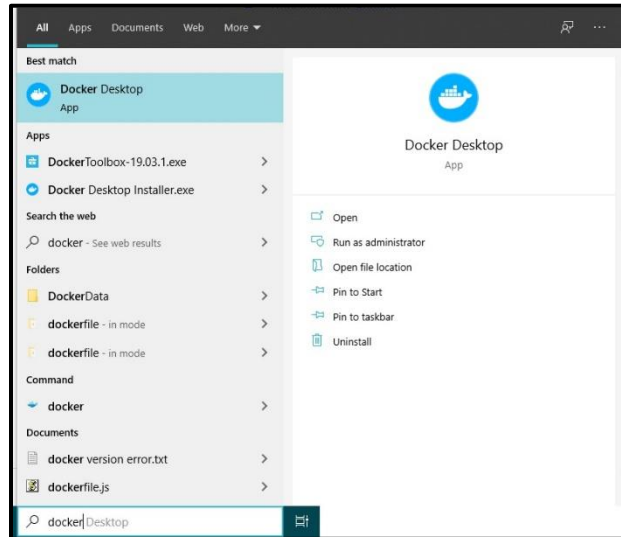


Figura 3. Iniciar Docker Desktop.

Después de instalar la aplicación Docker Desktop, también aparece un mensaje emergente de éxito con los siguientes pasos sugeridos y un enlace a esta documentación.



Figura 4. Post instalación de Docker

Cabe mencionar que no es necesario crear una cuenta en Docker para utilizar la herramienta.



Una vez finalizada la instalación, puede abrir Windows Power Shell para usar la herramienta, deberá ingresar en la consola el siguiente comando:

`docker --version`

Si el resultado coincide como lo que se muestra a continuación, usted instaló correctamente la herramienta.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Jose> docker --version
Docker version 19.03.8, build afacb8b
PS C:\Users\Jose>
```

Figura 5. Comprobar versión de Docker

II. CREAR UN MAPA WEB DESDE QGIS

1.2. Instalar el complemento qgis2web.

Lo primero que haremos será instalar desde QGIS el complemento qgis2web el cual genera un mapa web a partir de su proyecto actual de QGIS, ya sea como OpenLayers, Leaflet o Mapbox GL JS. Replica tantos aspectos del proyecto como sea posible, incluidas capas, estilos (incluidos categorizados y graduados) y extensión. (Figura 1).

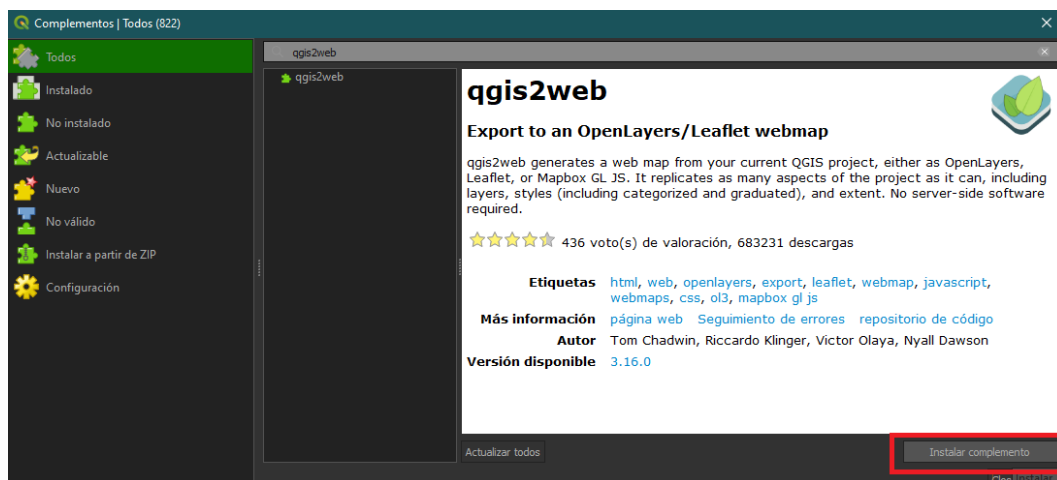


Figura 6. Instalar complemento qgis2web



1.3. Crear mapa en QGIS

Cargar las capas en QGIS que deseamos mostrar en el mapa web, asignado los estilos y demás elementos que consideremos necesarios.

Agregaremos también Open Street Map como mapa base.

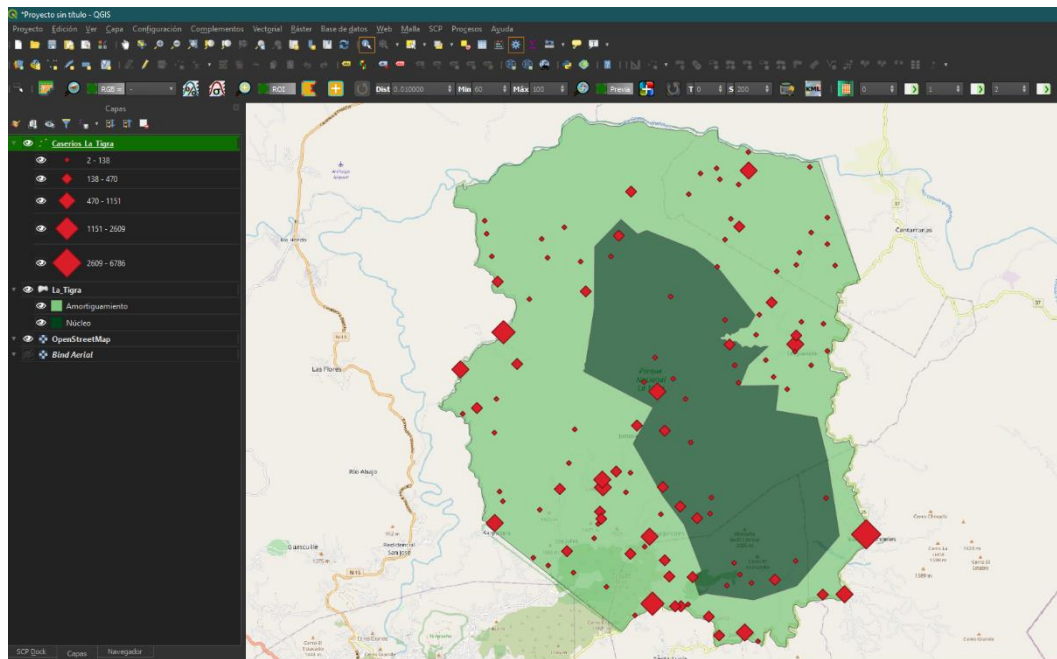


Figura 7. Crear un mapa en QGIS

Luego, accederemos al complemento qgis2web, mediante el menú Web - qgis2web – Create web map

Aquí configuraremos la forma de mostrar las capas dentro del mapa. En la siguiente figura se muestra un mapa con un mapa base (OSM). También se define el estilo de mostrar las consultas de los atributos de las capas.

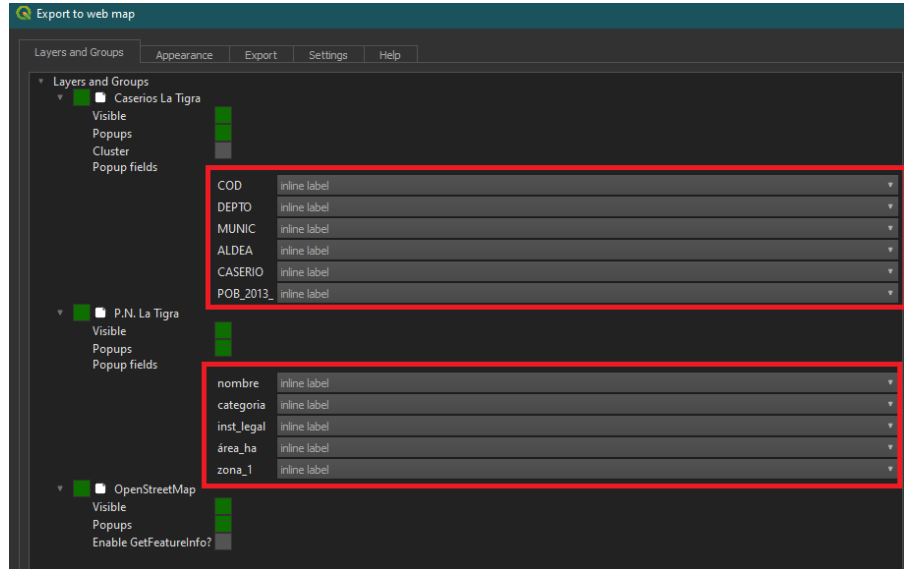


Figura 8. Configuración de capas

Luego, en la siguiente pestaña modificaremos la apariencia del mapa, agregando un botón de búsqueda de y una regla de medición.

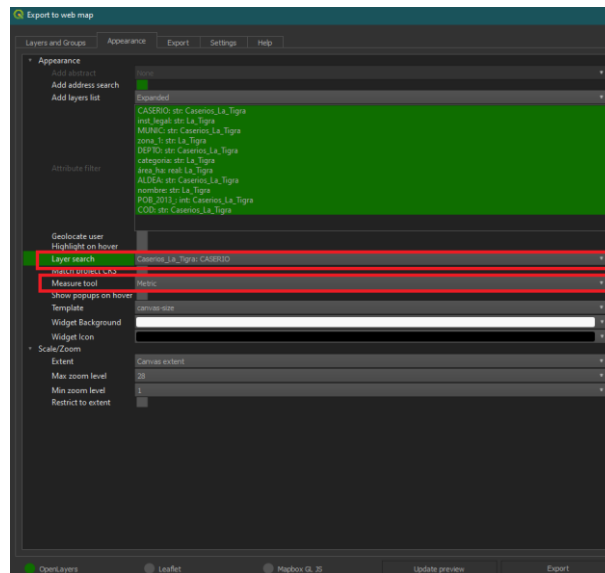


Figura 9. Configuración de apariencia

Luego, pasaremos a las opciones de exportación. Aquí seleccionaremos la opción de “Export to folder” y seleccionaremos la carpeta “C:\CTE-334\Examen_Unidad_I”.

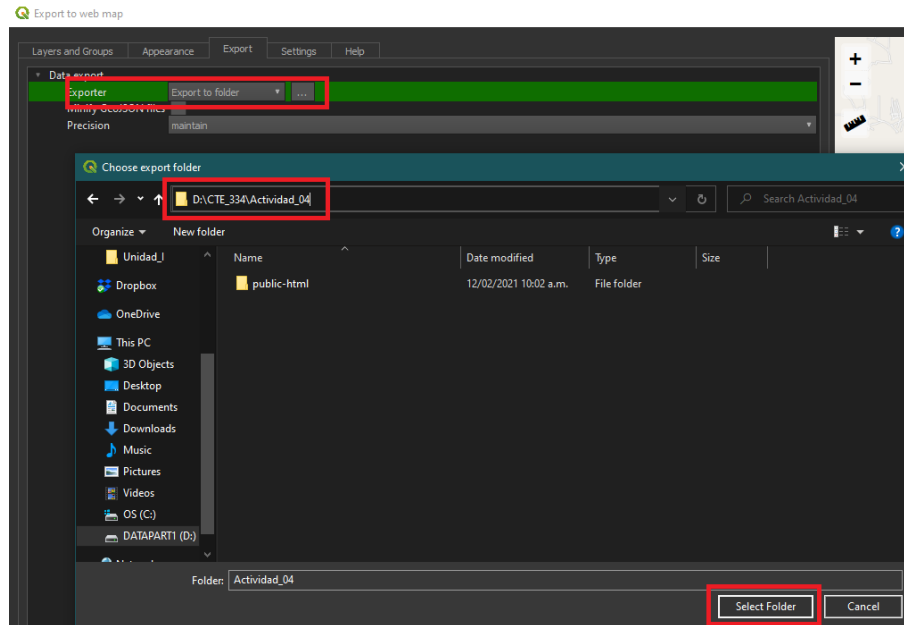


Figura 10. Configurar exportación

Luego daremos clic sobre el botón Export para crear el mapa Web, el cual se mostrará automáticamente en el navegador.

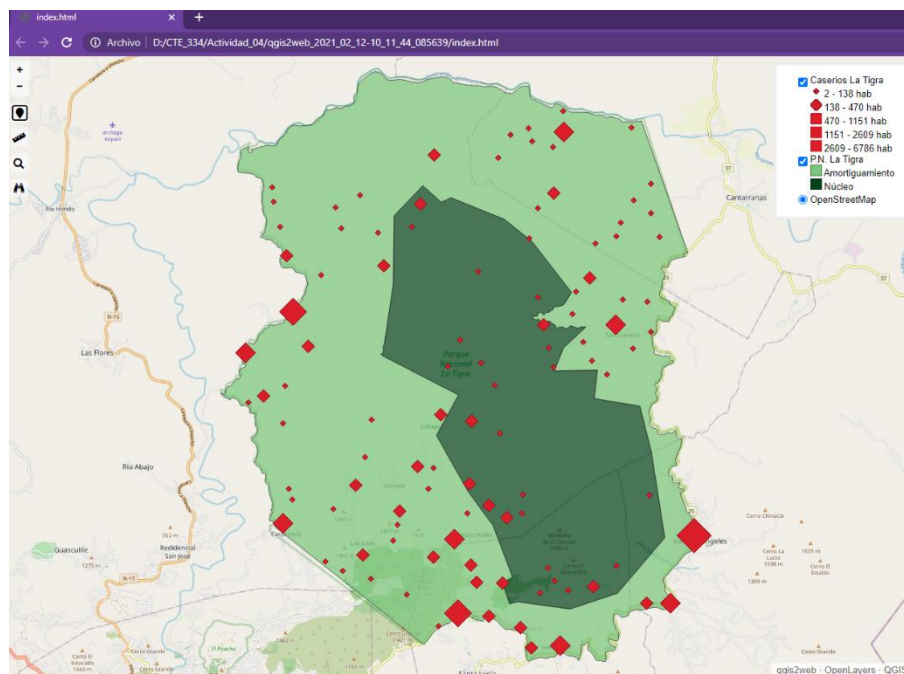


Figura 11. Mapa web creado



Por defecto, la herramienta ha creado una carpeta llamada qgis2web_XXXX, dentro de la carpeta de trabajo, por lo que procederemos a trasladar el contenido de esa carpeta, dentro de la carpeta C:/CTE-334\Examen_Unidad_I\public-html.

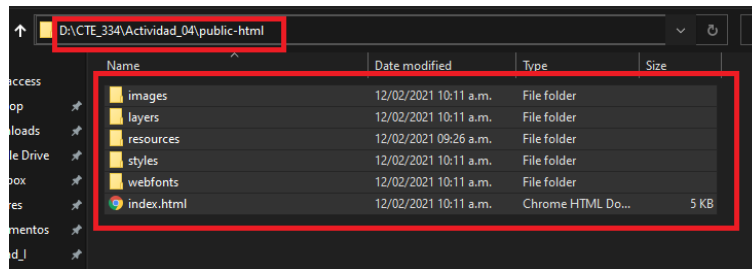


Figura 12. Estructura de los archivos creados

NOTA: El mapa anterior es únicamente ilustrativo del proceso. En su mapa deberá incluir 3 capas de su elección con geometrías diferentes (líneas, puntos y polígonos), y que tengan una coherencia o relación entre ellas, así como el OSM como mapa base.

III. CREAR UN SERVIDOR WEB USANDO DOCKER

Para crear una imagen necesitamos un “Dockerfile”, el cual es un archivo de texto que contiene todos los pasos para crear una imagen. La imagen que contiene el OS, los programas y la aplicación que deseamos usar, y esta imagen se ejecuta mediante un contenedor.

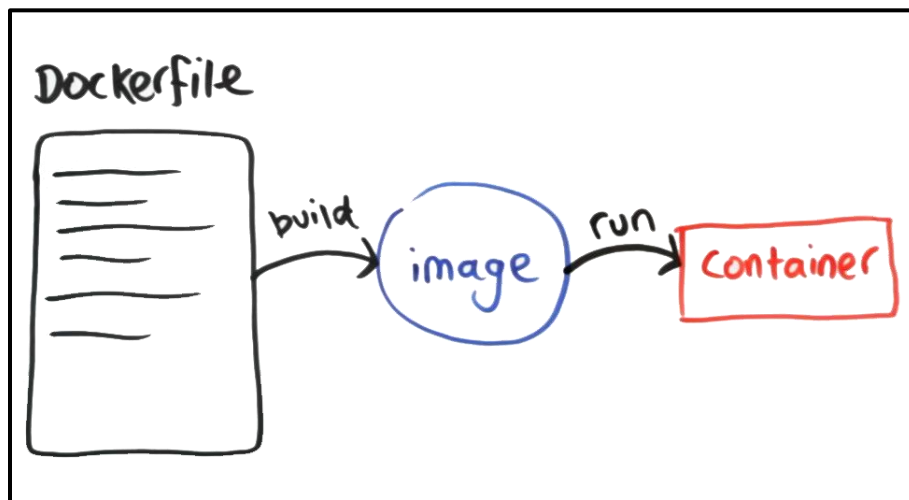


Figura 13. Flujo de trabajo de Docker

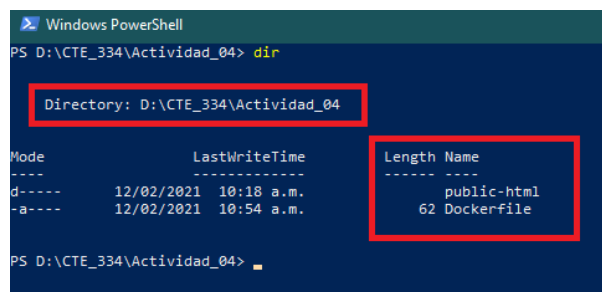
2.1. Crear un servidor HTTP

Dentro de la carpeta C:\CTE-334\Examen_Unidad_I crearemos un archivo de texto llamado “Dockerfile” (sin ninguna extensión y con “D” mayúscula), e incluiremos el siguiente contenido dentro del archivo:

```
FROM httpd:2.4
```

```
COPY ./public-html/ /usr/local/apache2/htdocs/
```

Una vez creado el Dockerfile, abriremos el Powershell y navegaremos hasta esa misma carpeta de trabajo, la cual deberá contener los dos elementos que se muestran a continuación:



```
Windows PowerShell
PS D:\CTE_334\Actividad_04> dir

Directory: D:\CTE_334\Actividad_04

Mode                LastWriteTime         Length Name
----                -
d-----         12/02/2021  10:18 a.m.          public-html
-a----         12/02/2021  10:54 a.m.           62 Dockerfile
```

La imagen muestra una terminal de Windows PowerShell. Se ejecutó el comando 'dir' en la carpeta 'D:\CTE_334\Actividad_04'. El resultado muestra dos elementos: un directorio 'public-html' y un archivo 'Dockerfile' de 62 bytes.

Figura 14. Archivos dentro de carpeta de trabajo



Luego, procederemos a ingresar el siguiente comando en el Powershell:

```
docker build -t apache-server .
```

donde:

- **docker build:** Crea una imagen a partir de un Dockerfile
- **t:** es un parámetro opcional para agregar una etiqueta (tag) a la imagen. En este caso “apache-server”
- **..:** Indica que la localización del Dockerfile está dentro del directorio actual

Finalmente, procederemos a crear el contenedor para la imagen, con el comando:

```
docker run -dit --name webmap -p 8080:80 apache-server
```

donde:

- **docker run:** ejecuta el proceso en un contenedor aislado
- **-d:** ejecuta el contenedor en modo desacoplado
 - **i:** mantiene el estándar input (STDIN) abierto.
- **-t:** crea una pseudo-tty
- **--name:** asigna un nombre al contenedor
- **-p:** crea una conexión entre el puerto del host y el puerto del contenedor HOST:CONTENEDOR

Una vez ejecutado el proceso, procederemos a acceder al servicio desde el navegador, en la dirección:

<http://localhost:8080/>

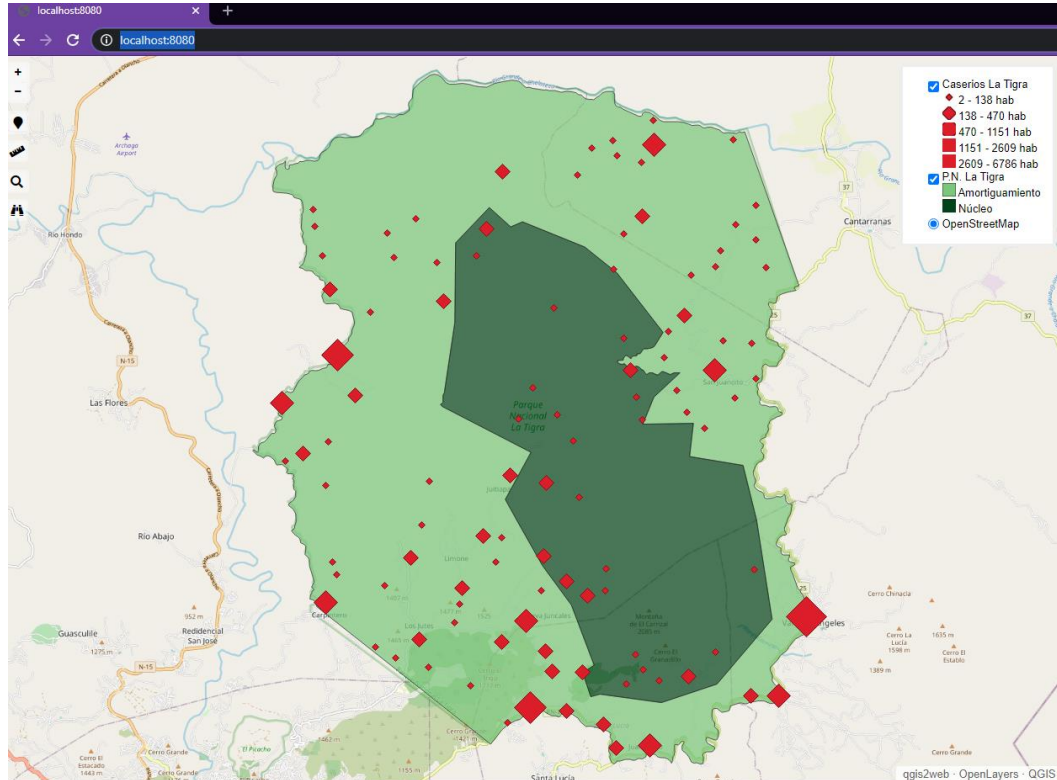


Figura 15. Mapa publicado mediante un servidor Web

IV. CONVERTIR CONTENEDOR DOCKER A UNA NUEVA IMAGEN

Para este último paso, deberemos haber creado una cuenta en [Docker Hub](https://hub.docker.com/).

Una vez tengamos nuestro contenedor en ejecución, podemos ejecutar el siguiente comando:

```
docker ps
```

Este nos mostrará todos los contenedores que estén en ejecución, y lo más importante, podremos localizar el nombre del contenedor que nos interesa:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ad61fdbb7c99	apache-server	"httpd-foreground"	2 minutes ago	Up 2 minutes	0.0.0.0:8081->80/tcp	webmap
02e0fe7dc9be	jcaceres85/tomcat_enee	"catalina.sh run"	7 months ago	Up 5 minutes	0.0.0.0:8088->8080/tcp	tomcat9

Figura 16. Listado de servicios en ejecución

Luego convertiremos el contenedor webmap en una nueva imagen, con el comando:

```
docker commit webmap
```



Luego ejecutaremos el siguiente comando para mostrar todas las imágenes disponibles:

docker images

```
PS D:\CTE-334_old\A04> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
<none>	<none>	e3dc020493e4	2 seconds ago	191MB
apache-server	latest	84c39c9c2faf	8 minutes ago	191MB
node-server	latest	f58bd151ef8a	4 weeks ago	275MB
jcaceres85/tomcat_enee	latest	c8c87b500393	12 months ago	751MB

Observaremos que aparece una imagen con el valor <none>, esta es la imagen que acabamos de crear en el paso anterior, por lo que procederemos a asignar una etiqueta (tag) para poder identificarla, para ello deberemos tomar el valor de la columna IMAGE ID y deberemos vincularlo al usuario que hayan creado en Docker Hub. En mi caso mi usuario es jcaceres85, por lo que deberán sustituir este valor, junto con el IMAGE_ID en el siguiente comando:

docker tag image_ID jcaceres85/webmap

Después de ejecutar el comando, se asignará el nombre a la imagen, por lo que podemos identificarla de manera más fácil, podemos ejecutar nuevamente *docker images*, para ver el cambio:

```
PS D:\CTE-334_old\A04> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
jcaceres85/webmap	latest	e3dc020493e4	4 minutes ago	191MB
apache-server	latest	84c39c9c2faf	13 minutes ago	191MB
node-server	latest	f58bd151ef8a	4 weeks ago	275MB
jcaceres85/tomcat_enee	latest	c8c87b500393	12 months ago	751MB

```
PS D:\CTE-334_old\A04>
```

Finalmente, procederemos a ingresar a Docker Hub, siempre desde la consola, ejecutando el comando:

docker login

Si es la primera vez que ingresamos nos pedirá ingresar nuestro usuario y contraseña. Una vez el ingreso ha sido exitoso, procederemos a subir nuestra imagen al repositorio Docker Hub, con el siguiente comando:



docker push jcaceres85/webmap

Recuerde reemplazar jcaceres85 por su propio usuario.

```
PS D:\CTE-334_old\A04> docker push jcaceres85/webmap
Using default tag: latest
The push refers to repository [docker.io/jcaceres85/webmap]
91df92e00710: Pushed
38c7e09fa4c7: Pushed
cdae29f197b2: Mounted from library/httpd
5ca84525a215: Mounted from library/httpd
e0ba343b5cab: Mounted from library/httpd
2905795cb5b8: Mounted from library/httpd
ec983b166360: Mounted from library/httpd
latest: digest: sha256:37d1cf575cbc481385c8a674196e96680bb7d60f440604a2d2da94ec021cb9de size: 1784
```

PARA ENTREGAR:

1. El enlace del repositorio en Github, con todos los archivos generados (archivos del mapa web y Dockerfile).
2. Nombre de la imagen subida en Docker Hub.