



# UNIVERSIDAD AUTONOMA DEL ESTADO DE HIDALGO

Área Académica de Ciencias y Tecnología

Licenciatura de Ciencias Computacionales

GESTIÓN DE GIMNASIOS

Manual Técnico

Manual para Nivel de Licenciatura

SQL SLAYERS

Presentado por:

Barón Salinas Mauricio Valentín

Cristóbal Silverio Cristian

Gómez Daniel Aimar Jair

Hernández Reyes Magaly

Sánchez Carrasco Monserrat

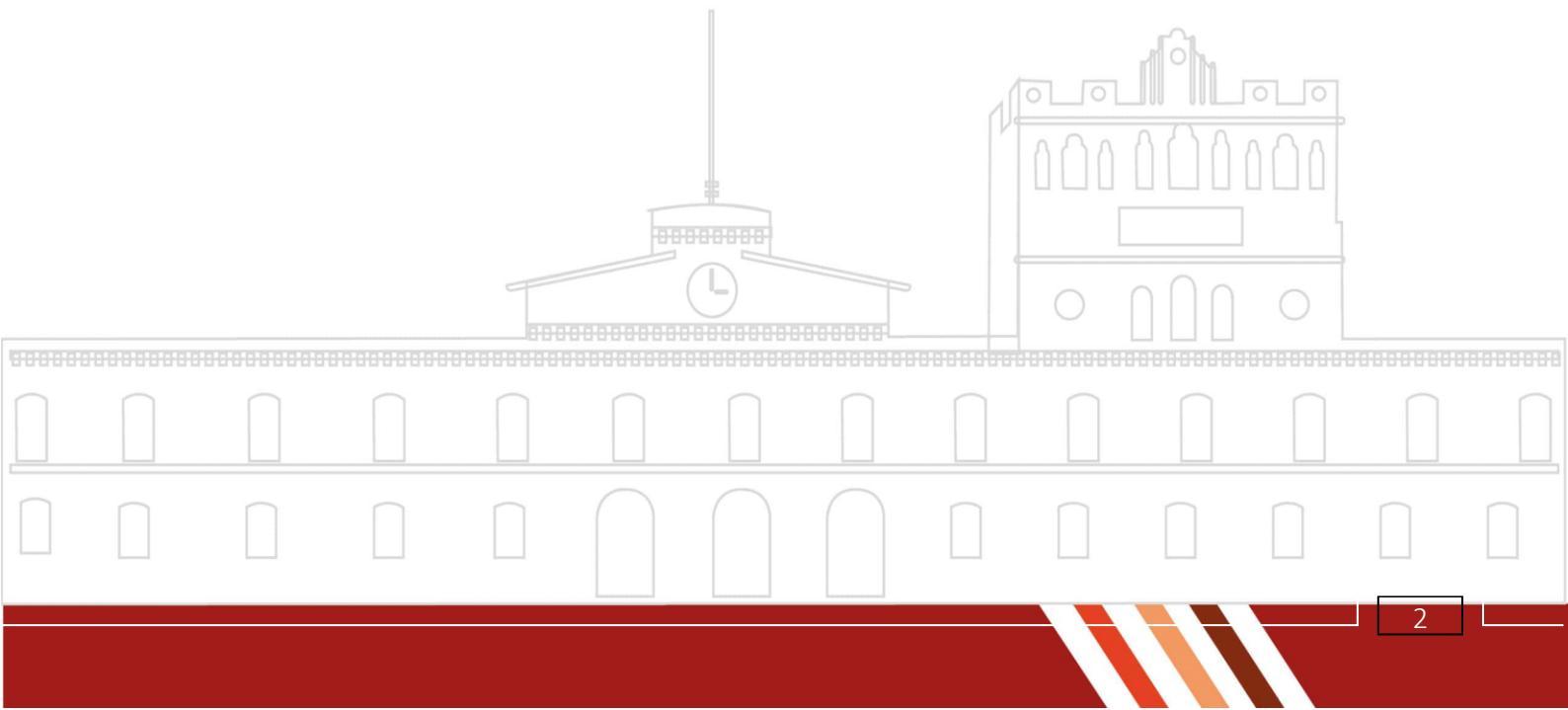
Ramírez Suárez Gerardo

Docente: Dr. Eduardo Cornejo Velázquez

Asignatura: Base de Datos Distribuida  
Pachuca de Soto, Hidalgo; Noviembre 2024

# Tabla de contenido

<b>Introducción</b>	<b>3</b>
Introduction	3
<b>Herramientas software</b>	<b>4</b>
Docker y Contenedорización	4
Ventajas de Docker	4
MongoDB y Bases de Datos NoSQL	4
Características de MongoDB	4
Visual Studio Code (VS Code)	4
Beneficios de VS Code	5
<b>Manual de Instalación de Docker Desktop y WSL</b>	<b>6</b>
Requisitos previos	6
Paso 1: Instalar y configurar WSL 2	6
Paso 2: Instalar Docker Desktop	7
Paso 3: Probar la instalación	7
Paso 4: Configuración adicional (opcional)	8
Solución de problemas	8
<b>Manual de Instalación: MongoDB, Mongo Shell y MongoDB Tools en Windows</b>	<b>9</b>
Requisitos previos	9
Paso 1: Descargar MongoDB Community Server	9
Paso 2: Instalar MongoDB	9
Paso 3: Instalar MongoDB Shell y MongoDB Tools	9
Paso 4: Configurar variables de entorno	10
Paso 5: Verificar la instalación	10
Paso 6: Iniciar MongoDB	11
Paso 7: Conectarse a MongoDB	11
Solución de problemas	11
<b>Para agregar una imagen de Docker que contiene un contenedor con MongoDB y configurarlo en tu entorno de desarrollo, sigue estos pasos:</b>	<b>12</b>
Paso 1: Crear un Archivo Dockerfile	12
Paso 2: Construir la Imagen Docker	12
Paso 3: Ejecutar el Contenedor de MongoDB	12
Paso 4: Configurar Visual Studio Code para Conectarse al Contenedor	13
Paso 5: Conectar Diferentes Laptops al Contenedor	13
<b>Conclusión</b>	<b>14</b>



## Introducción

La virtualización y los contenedores han revolucionado la forma en que se desarrollan y despliegan las aplicaciones modernas. Docker es una plataforma líder en contenedorización que permite a los desarrolladores empaquetar aplicaciones y sus dependencias en contenedores, asegurando que funcionen de manera consistente en diferentes entornos.

MongoDB, una base de datos NoSQL orientada a documentos, se integra perfectamente con Docker para ofrecer soluciones de almacenamiento de datos escalables y flexibles. Visual Studio Code (VS Code), un editor de código ligero pero poderoso, complementa este ecosistema al proporcionar un entorno de desarrollo robusto.

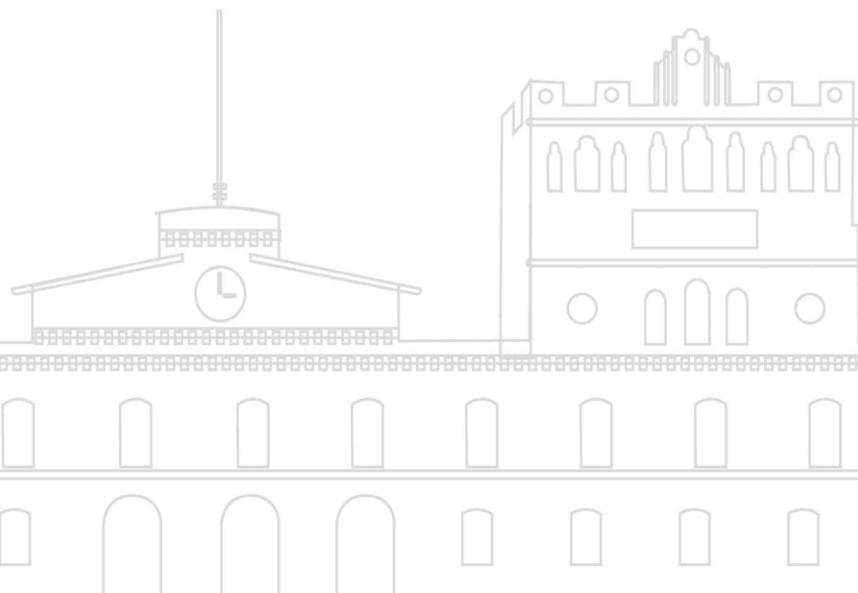
Este marco teórico aborda la implementación de Docker con MongoDB y su integración con VS Code, así como la configuración para conectarse a diferentes laptops en un entorno colaborativo.

## Introduction

Virtualization and containers have revolutionized the way modern applications are developed and deployed. Docker is a leading containerization platform that allows developers to package applications and their dependencies into containers, ensuring that they work consistently across different environments.

MongoDB, a document-oriented NoSQL database, integrates seamlessly with Docker to provide scalable and flexible data storage solutions. Visual Studio Code (VS Code), a lightweight yet powerful code editor, complements this ecosystem by providing a robust development environment.

This theoretical framework addresses the implementation of Docker with MongoDB and its integration with VS Code, as well as the configuration to connect to different laptops in a collaborative environment.



## Herramientas software

### Docker y Contenedорización

Docker es una plataforma de código abierto que automatiza la implementación de aplicaciones dentro de contenedores ligeros. Los contenedores son unidades estándar de software que encapsulan el código, las dependencias y el entorno necesario para ejecutar una aplicación, lo que garantiza la coherencia entre los entornos de desarrollo, prueba y producción. Docker facilita la gestión de dependencias y la portabilidad, permitiendo a los desarrolladores concentrarse en escribir código sin preocuparse por las discrepancias del entorno.



#### Ventajas de Docker

- **Portabilidad:** Los contenedores Docker pueden ejecutarse en cualquier sistema con Docker instalado.
- **Consistencia:** Asegura que la aplicación se ejecute de la misma manera en todos los entornos.
- **Aislamiento:** Los contenedores funcionan de manera aislada, lo que evita conflictos de dependencias.

### MongoDB y Bases de Datos NoSQL

MongoDB es una base de datos NoSQL orientada a documentos que utiliza un formato de almacenamiento JSON flexible llamado BSON. A diferencia de las bases de datos SQL tradicionales, MongoDB no requiere un esquema predefinido, lo que permite la inserción de datos sin restricciones de estructura. Esto lo hace ideal para aplicaciones que requieren agilidad y escalabilidad.



#### Características de MongoDB

- **Escalabilidad Horizontal:** Facilita la distribución de datos a través de múltiples servidores.
- **Modelado Flexible:** Permite cambios rápidos en la estructura de datos.
- **Consultas Ricas:** Ofrece un poderoso lenguaje de consultas.

### Visual Studio Code (VS Code)

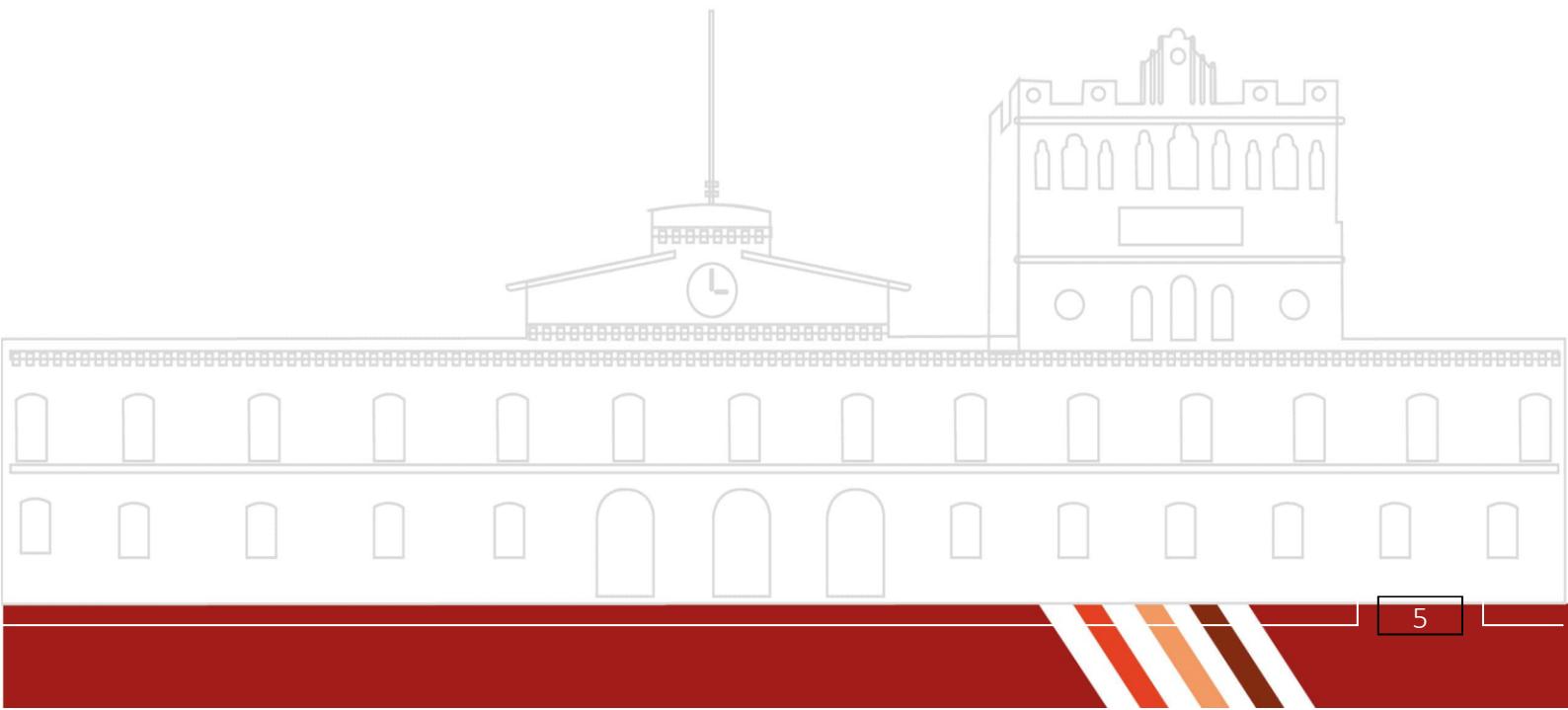
Visual Studio Code es un editor de código fuente desarrollado por Microsoft que se ha convertido en una herramienta esencial para los desarrolladores debido a su ligereza y extensibilidad. VS Code ofrece características avanzadas como autocompletado inteligente, depuración integrada y extensiones para una amplia variedad de lenguajes y herramientas de desarrollo.



# Visual Studio Code

## *Beneficios de VS Code*

- **Ligero y Rápido:** Ideal para el desarrollo ágil.
- **Extensible:** Amplia funcionalidad a través de extensiones.
- **Integración con Docker:** Facilita la gestión de contenedores y la interacción con bases de datos como MongoDB.



# Manual de Instalación de Docker Desktop y WSL

## Requisitos previos

Antes de comenzar, asegúrate de cumplir con los siguientes requisitos:

1. **Sistema operativo compatible:**
  - o Windows 10 64 bits (Build 1903 o superior) o Windows 11.
2. **Hardware:**
  - o Procesador de 64 bits con soporte para virtualización (Habilitar en BIOS/UEFI).
  - o Al menos 4 GB de RAM.
3. **Permisos administrativos** en tu PC para realizar configuraciones y habilitar características.

## Paso 1: Instalar y configurar WSL 2

WSL (Windows Subsystem for Linux) es una capa de compatibilidad que permite ejecutar un entorno Linux en Windows.

### 1.1 Habilitar las características de Windows necesarias

1. Abre una ventana de **PowerShell** con permisos de administrador. Para hacerlo:
  - o Busca "PowerShell" en el menú de inicio, haz clic derecho y selecciona **Ejecutar como administrador**.
2. Ejecuta los siguientes comandos para habilitar WSL y la máquina virtual requerida:

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```
3. Reinicia tu computadora para aplicar los cambios.

### 1.2 Configurar WSL 2 como la versión predeterminada

1. Descarga el paquete de actualización del kernel de Linux para WSL 2 desde el siguiente enlace:  
[Paquete de actualización de kernel WSL](#)
2. Instala el paquete descargado siguiendo las instrucciones en pantalla.
3. En PowerShell, establece WSL 2 como predeterminado:  
`wsl --set-default-version 2`

### 1.3 Instalar una distribución de Linux

1. Abre Microsoft Store y busca una distribución de Linux como **Ubuntu**, **Debian** o **Fedora**.
2. Haz clic en **Obtener** para instalar la distribución.
3. Inicia la distribución desde el menú de inicio y sigue las instrucciones para configurar tu cuenta (usuario y contraseña).

## Paso 2: Instalar Docker Desktop

### 2.1 Descargar Docker Desktop

1. Ve al sitio oficial de Docker: [Descargar Docker Desktop](#).
2. Haz clic en **Download for Windows** y guarda el archivo ejecutable.

### 2.2 Instalar Docker Desktop

1. Ejecuta el instalador descargado y sigue los pasos del asistente:
  - Acepta los términos de uso.
  - Asegúrate de seleccionar **Usar el motor WSL 2 como backend predeterminado** durante la instalación.
2. Una vez finalizada la instalación, reinicia tu computadora si se solicita.

### 2.3 Iniciar Docker Desktop

1. Abre Docker Desktop desde el menú de inicio.
2. Verifica que Docker esté utilizando WSL 2:
  - Haz clic en el ícono de configuración (⚙️) en Docker Desktop.
  - Ve a **Resources > WSL Integration**.
  - Asegúrate de que la integración con WSL 2 esté habilitada y selecciona las distribuciones de Linux que deseas usar con Docker.

## Paso 3: Probar la instalación

### 3.1 Verificar WSL

1. Abre una terminal de Linux (por ejemplo, Ubuntu desde el menú de inicio).

Ejecuta el comando:  
`wsl --list --verbose`
2. Asegúrate de que la versión predeterminada sea **2**.

### 3.2 Verificar Docker

1. Abre una terminal de Linux o PowerShell.

Ejecuta el comando:

```
docker --version
```

2. Esto debería devolver la versión instalada de Docker.

Prueba ejecutando un contenedor simple:

```
docker run hello-world
```

3. Si todo está configurado correctamente, verás un mensaje indicando que Docker está funcionando.

## Paso 4: Configuración adicional (opcional)

### 4.1 Actualizar Docker Desktop

Mantén Docker Desktop actualizado para obtener nuevas funcionalidades y correcciones de errores. Puedes verificar las actualizaciones desde la configuración de Docker Desktop.

### 4.2 Usar Docker Compose

Docker Desktop incluye **Docker Compose**, una herramienta para definir y ejecutar aplicaciones Docker multi-contenedor. Verifica su instalación con:

```
docker-compose --version
```

### 4.3 Configurar redes o volúmenes personalizados

Si planeas usar Docker en proyectos complejos, puedes configurar redes y volúmenes personalizados para adaptarlo a tus necesidades.

## Solución de problemas

### Problema: Docker no detecta WSL 2

- Asegúrate de que WSL 2 esté habilitado y sea la versión predeterminada.
- Verifica que las distribuciones de Linux estén correctamente integradas con Docker Desktop.

### Problema: Error "Virtualización deshabilitada"

- Habilita la virtualización en la configuración del BIOS/UEFI de tu computadora. Consulta el manual de tu placa base o fabricante para más detalles.



# Manual de Instalación: MongoDB, Mongo Shell y MongoDB Tools en Windows

## Requisitos previos

Antes de comenzar, asegúrate de:

- Tener Windows 10 o superior (64 bits).
- Tener permisos administrativos para realizar la instalación.
- Contar con al menos 512 MB de RAM y 200 MB de espacio en disco.

## Paso 1: Descargar MongoDB Community Server

1. Ve al sitio oficial de MongoDB:  
[Descargar MongoDB](#).
2. Selecciona las siguientes opciones en la página de descarga:
  - Version: Elige la versión más reciente.
  - Platform: Windows.
  - Package: MSI.
3. Haz clic en Download para descargar el instalador.

## Paso 2: Instalar MongoDB

1. Ejecuta el archivo .msi descargado.
2. Sigue las instrucciones del instalador:
  - Selecciona Complete para una instalación completa.
  - En la sección "Service Configuration", deja la configuración predeterminada (MongoDB se ejecutará como un servicio).
  - Habilita la opción Install MongoDB Compass si deseas instalar la interfaz gráfica (opcional).
3. Finaliza la instalación.

## Paso 3: Instalar MongoDB Shell y MongoDB Tools

### 3.1 MongoDB Shell

MongoDB Shell (mongosh) es una herramienta interactiva para trabajar con bases de datos.

1. Ve a la página oficial de MongoDB Shell:  
[Descargar MongoDB Shell](#).
2. Descarga el instalador para Windows y sigue las instrucciones para instalarlo.

### 3.2 MongoDB Tools

MongoDB Tools incluye herramientas como `mongodump`, `mongorestore`, `mongoexport`, y más.

1. Descarga MongoDB Database Tools desde el sitio oficial:  
[Descargar MongoDB Tools](#).
2. Extrae el contenido del archivo ZIP en una carpeta de tu elección, como C:\Program Files\MongoDB\Tools.

## Paso 4: Configurar variables de entorno

1. Abre el Panel de Control y ve a:
  - o Sistema > Configuración avanzada del sistema > Variables de entorno.
2. En Variables del sistema, selecciona la variable `Path` y haz clic en Editar.
3. Agrega las siguientes rutas (según tu instalación):
  - o Para MongoDB Server:  
C:\Program Files\MongoDB\Server\<versión>\bin
  - o Para MongoDB Tools:  
C:\Program Files\MongoDB\Tools\bin
  - o Para MongoDB Shell:  
C:\Program Files\MongoDB\Shell\<versión>\bin
4. Haz clic en Aceptar para guardar los cambios.

## Paso 5: Verificar la instalación

### 5.1 Verificar MongoDB Server

1. Abre una ventana de PowerShell o Símbolo del sistema.

Ejecuta el siguiente comando:

```
mongod --version
```

2. Deberías ver la versión instalada de MongoDB.

### 5.2 Verificar MongoDB Shell

Abre una terminal y ejecuta:

```
mongosh
```

1. Si la instalación es correcta, deberías acceder a la consola interactiva de MongoDB Shell.

### 5.3 Verificar MongoDB Tools

Ejecuta cualquiera de las herramientas, por ejemplo:

```
mongodump --version
```

Esto confirmará que MongoDB Tools está instalado correctamente.

## Paso 6: Iniciar MongoDB

### 6.1 Iniciar MongoDB como servicio

MongoDB debería iniciarse automáticamente al instalarlo como servicio. Para verificarlo:

1. Abre Servicios (puedes buscar "Servicios" en el menú de inicio).
2. Busca el servicio llamado MongoDB y asegúrate de que esté en ejecución.

### 6.2 Iniciar MongoDB manualmente

Si necesitas iniciar MongoDB manualmente:

1. Navega al directorio donde se instaló MongoDB (por ejemplo, `C:\Program Files\MongoDB\Server\<versión>\bin`).
2. Ejecuta el siguiente comando:  
`mongod`

## Paso 7: Conectarse a MongoDB

1. Abre MongoDB Shell con:

`mongosh`

2. Por defecto, se conectará al servidor local (`localhost`) en el puerto `27017`.

## Solución de problemas

Error: 'mongod' no se reconoce como un comando interno

- Asegúrate de que las rutas de las herramientas estén correctamente configuradas en las variables de entorno.

Error al iniciar el servicio MongoDB

- Verifica los permisos de usuario.
- Asegúrate de que no haya otro proceso utilizando el puerto `27017`.

El servicio no se inicia automáticamente

- Inicia el servicio manualmente desde la herramienta Servicios de Windows.

Para agregar una imagen de Docker que contiene un contenedor con MongoDB y configurarlo en tu entorno de desarrollo, sigue estos pasos:

#### Paso 1: Crear un Archivo Dockerfile

- Crea un archivo llamado Dockerfile en tu directorio de trabajo.
- Escribe el siguiente contenido en el Dockerfile para especificar cómo se debe construir la imagen Docker:

#### Utilizar la imagen base oficial de MongoDB

```
FROM mongo:latest
```

Definir el directorio de trabajo dentro del contenedor

```
WORKDIR /data/db
```

Exponer el puerto 27017 para MongoDB

```
EXPOSE 27017
```

Comando para ejecutar MongoDB

```
CMD ["mongod"]
```

#### Paso 2: Construir la Imagen Docker

##### 2. Construir la Imagen:

- Abre una terminal y navega al directorio donde se encuentra el Dockerfile.
- Ejecuta el siguiente comando para construir la imagen Docker:

```
docker build -t mi-mongodb .
```

Este comando construirá una imagen Docker llamada mi-mongodb utilizando las instrucciones del Dockerfile.

#### Paso 3: Ejecutar el Contenedor de MongoDB

##### 3. Ejecutar el Contenedor:

- Una vez que la imagen está construida, puedes ejecutar un contenedor basado en esa imagen con el siguiente comando:

```
docker run -d -p 27017:27017 --name mi-contenedor-mongodb mi-mongodb
```

Este comando iniciará un contenedor en segundo plano (-d) y mapeará el puerto 27017 del host al puerto 27017 del contenedor, lo que permitirá acceder a MongoDB desde tu máquina local.

#### Paso 4: Configurar Visual Studio Code para Conectarse al Contenedor

##### 4. Configurar la Conexión en VS Code:

- Abre Visual Studio Code.
- Instala las extensiones de Docker y MongoDB desde el marketplace de VS Code.
- Configura la conexión a MongoDB en VS Code agregando la siguiente configuración en el archivo de configuración de MongoDB:

```
{  
  "mongo.connections": [  
    {  
      "name": "Local MongoDB",  
      "connectionString": "mongodb://localhost:27017"  
    }  
  ]  
}
```

#### Paso 5: Conectar Diferentes Laptops al Contenedor

##### 5. Configurar la Conexión Remota:

- Si necesitas que diferentes laptops se conecten al contenedor MongoDB, asegúrate de que Docker esté configurado para aceptar conexiones remotas y que el puerto 27017 esté abierto en el firewall.

##### 6. Conexión desde otras Laptops:

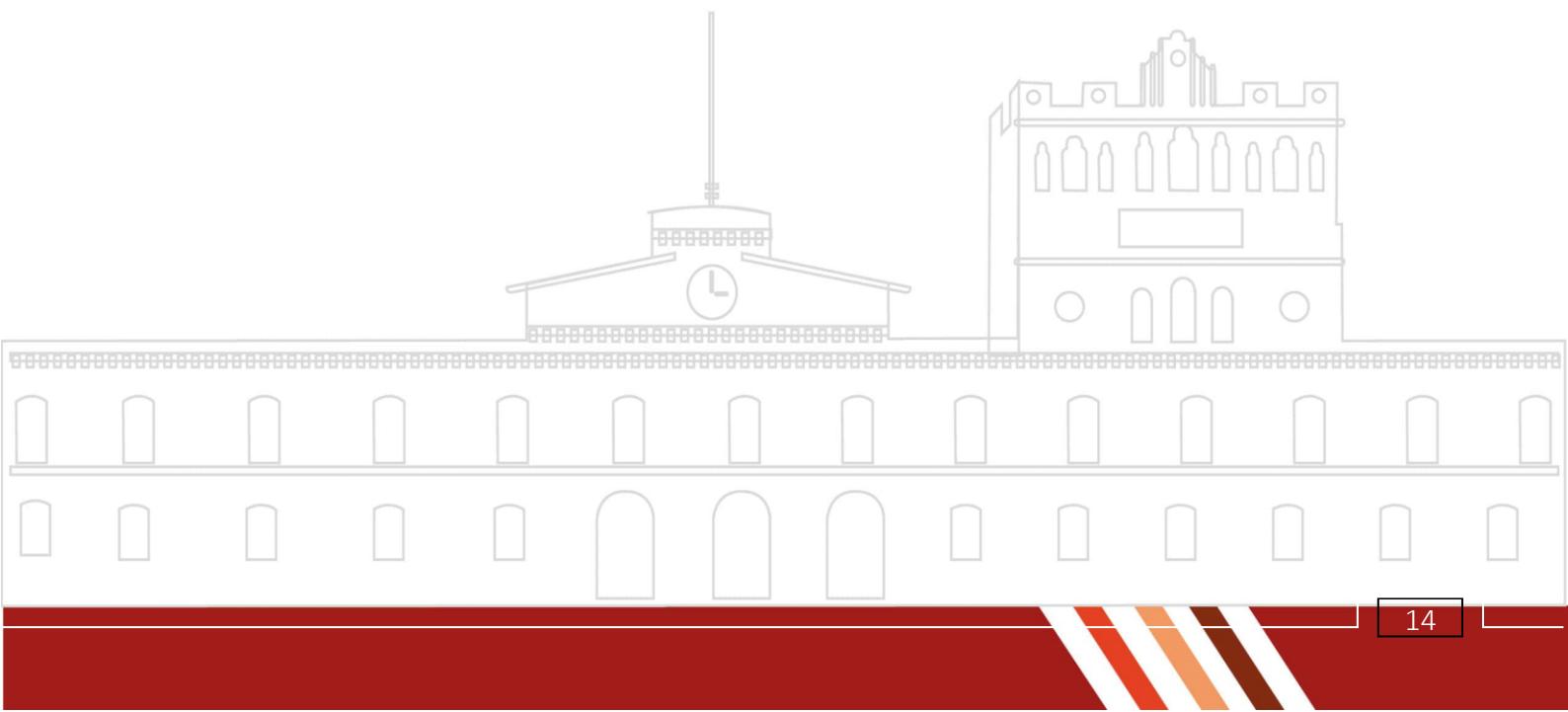
- En cada laptop, configura la conexión a MongoDB en VS Code utilizando la dirección IP del host donde se está ejecutando el contenedor:

```
{ "mongo.connections": [  
  { "name": "Remote MongoDB",  
    "connectionString": "mongodb://<IP-del-host>:27017"  
  } ]  
}
```

Reemplaza <IP-del-host> con la dirección IP del host que ejecuta el contenedor.

## Conclusión

La integración de Docker con MongoDB y Visual Studio Code ofrece una solución robusta y flexible para el desarrollo y despliegue de aplicaciones. La contenedorización asegura la portabilidad y consistencia del entorno, mientras que MongoDB proporciona una base de datos escalable y flexible. VS Code complementa este ecosistema con sus poderosas herramientas de desarrollo. La configuración adecuada de estos componentes y la conexión entre diferentes laptops facilitan un entorno colaborativo eficiente y productivo.



## Referencias

- **Configurar y desplegar un contenedor de MongoDB con Docker** - Un tutorial detallado sobre cómo instalar y configurar MongoDB en un contenedor Docker.  
[Configurar y desplegar un contenedor de MongoDB con Docker | Diego Orozco](#)
- **Cómo instalar y configurar MongoDB en Docker** - Un artículo de Serverspace que profundiza en el proceso de configuración de MongoDB en Docker.  
[MongoDB Docker. Cómo instalar y configurar MongoDB en Docker?](#)
- **Instalación de MongoDB con Docker** - Un tutorial de CodigoElectronica que guía paso a paso la instalación y configuración de MongoDB usando Docker  
<http://codigoelectronica.com/blog/installacion-de-mongodb-con-docker>
- **Documentación Oficial de Docker** - La documentación oficial de Docker es una excelente fuente para aprender sobre la instalación, configuración y uso de contenedores Docker.  
[Docker Docs](#)
- **Documentación Oficial de MongoDB** - La documentación oficial de MongoDB ofrece guías, tutoriales y referencias detalladas sobre su uso y configuración  
[MongoDB Documentation](#)

