1. Programas
2. Docker Desktop

https://www.docker.com/products/docker-desktop/

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En el caso que les pida instalar el parche de WSL, deben descargarlo e instalarlo, luego reiniciar la computadora

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

[Pasos de instalación manual para versiones anteriores de WSL | Microsoft Docs](https://docs.microsoft.com/es-es/windows/wsl/install-manual#step-4---download-the-linux-kernel-update-package)

Instalar

<https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Iniciar Servidor de base de datos:**

Abrir un powershell en modo administrador, ir a la ruta de “**Infrastructure\sql2019**” y ejecutar el comando “**docker compose up -d**”

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

1. SSMS (Query Analizer SQL Server 2019)

<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. NodeJs

<https://nodejs.org/es/>

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

1. Visual Studio 2022 (Marcar la opción de desarrollo web)

Interfaz de usuario gráfica, Texto, Aplicación, Word, Correo electrónico

Descripción generada automáticamente

1. CLI Angular

* Ejecutar en un powershell en modo administrador, el comando “**Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser**”
* Ejecutar en un powershell en modo administrador, el comando “**npm install -g @angular/cli**”
* Ejecutar en un powershell en modo administrador, el comando “**ng version**”

Texto

Descripción generada automáticamente

1. Crear una base de datos SQL.
   1. Click derecho en la carpeta de base de datos y seleccionar nueva base de datos.

Graphical user interface, application

Description automatically generated

* 1. Ingresamos el nombre de la Base de datos “**BikeStores**” y pulsamos OK.

Graphical user interface, text, application

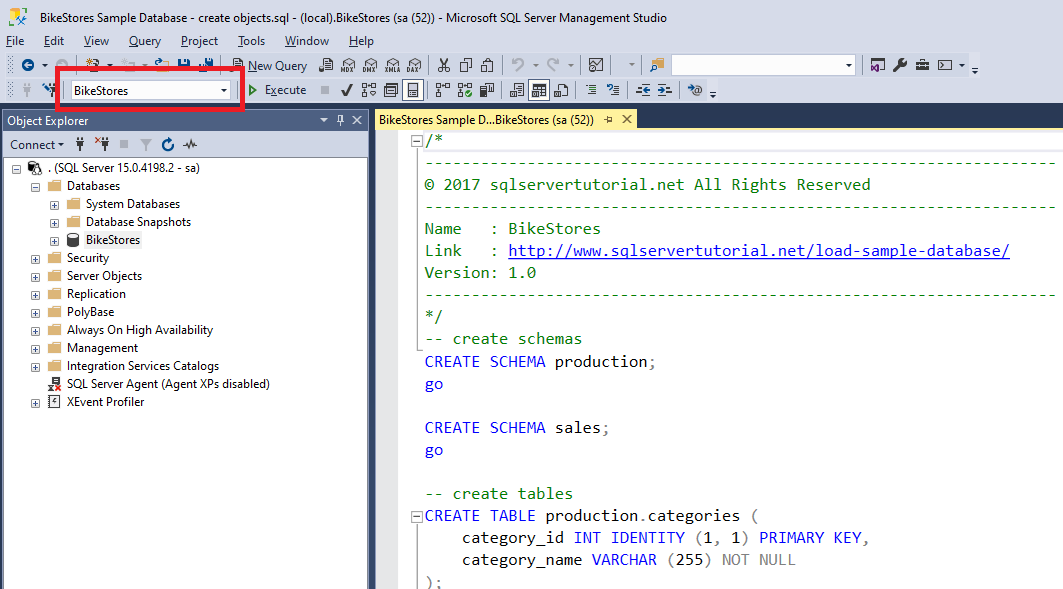
Description automatically generated

* 1. Se crea la BD.

Graphical user interface, text, application, email

Description automatically generated

1. Crear la estructura de tablas
   1. Abrir el archivo “**BikeStores Sample Database - create objects.sql**” con el SSMS, y la base de datos seleccionada debe ser “**BikeStores**”



* 1. Pulsar en el botón ejecutar (Play verde) y al final les dará el resultado de ejecución correcta, solo se puede ejecutar 1 vez, si se ejecuta mas de una vez les dará un error de que la base de datos ya existe.

Graphical user interface, text, application, email

Description automatically generated

1. Cargar los datos para la Base de datos.
   1. Abrir el archivo “**BikeStores Sample Database - load data.sql**” con el SSMS y la base de datos seleccionada debe ser “**BikeStores**”

Graphical user interface, text, application

Description automatically generated

* 1. Pulsar en el botón ejecutar (Play verde) y al final les dará el resultado de ejecución correcta, solo se puede ejecutar 1 vez, si se ejecuta más de una vez les dará un error de que la base de datos ya existe.

Graphical user interface, text, application

Description automatically generated

1. Ejecuta la aplicación base.

Prerrequisito:

* Abrir un powershell en modo administrador en la ruta de “**BikeStores.Client**” y ejecutar el comando “**npm install**” (va a obtener los paquetes para el cliente angular)
* Abrir un powershell en modo administrador en la ruta de “**BikeStores.Client**” y ejecutar el comando “**ng build**” (va a generar un compilado inicial)

1. Abrir el visual studio 2022 en modo administrador.
2. Abrimos la solución “**BikeStores.sln**”

Graphical user interface, text, application

Description automatically generated

1. Debe existir 2 proyecto, uno API (**BikeStores.API**) y el otro cliente (**BikeStores.Client**)

Graphical user interface, application

Description automatically generated

1. Configurar ejecución de aplicación, hacer click derecho en la solución y seleccionar “**Establecer proyectos de inicio**”

Graphical user interface, text, application

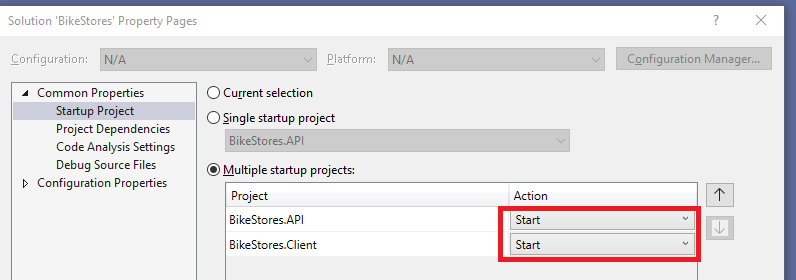
Description automatically generated

1. Seleccionar “**Iniciar** múltiples proyectos”

Graphical user interface, application

Description automatically generated

1. Establecer en “**Iniciar**” a los proyectos listados



1. Ejecutar la solución (Pulsar el botón start)



1. Este seria el resultado de la ejecución

Graphical user interface

Description automatically generated

1. Implementar las acciones CRUD de Brand
   1. Implementar la acción “**Listar**”, Ir al componente **list.componente.ts** del módulo **Brands** del feature module **bike-stores-manager** e implementar el código en el método **listBrands**.

Graphical user interface, text, application

Description automatically generated

* 1. El resultado de la ejecución seria :

Graphical user interface, application, Teams

Description automatically generated

* 1. Implementar la acción “**Editar**”, Ir al componente **edit.componente.ts** del módulo **Brands** del feature module **bike-stores-manager** e implementar el código en los métodos **getBrandByBrandId** (obtiene los datos del Brand)y **onFormSubmit** (graba los datos del Brand editado).

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application, website

Description automatically generated

* 1. Implementar la acción “**Nuevo**”, Ir al componente **new.componente.ts** del módulo **Brands** del feature module **bike-stores-manager** e implementar el código en el método **onFormSubmit** (graba los datos del Brand nuevo).

Graphical user interface, text, application

Description automatically generated

* 1. Implementar la acción “**Eliminar**”, Ir al componente **delete.componente.ts** del módulo **Brands** del feature module **bike-stores-manager** e implementar el código en los métodos **getBrandByBrandId** (obtiene los datos del Brand)y **onFormSubmit** (graba los datos del Brand editado).

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated