

**Universidad de Costa Rica**  
Escuela de Ciencias de la Computación e Informática

# **Ingeniería de Software**

## **CI-0126**

### **Laboratorio 4**

Creando un proyecto de backend conectado a base de datos

#### **Repositorio del laboratorio**

<https://github.com/MarioCordero/my-labs-ing-sof-CI-0126>

**Profesora:** Msc. Rebeca Obando Vásquez

**Estudiante:** Mario Cordero

**Carnet:** C22306

4 de septiembre de 2025

# 1. Conexión e Inserción de Datos en la Base de Datos

Para la configuración del entorno, se utilizó un contenedor de **SQL Server** en **Docker**.

```
mario@mario-EXT-Linux:~$ sudo docker run -e "ACCEPT_EULA=Y" \
-e "MSSQL_SA_PASSWORD=MyStrongPassword123!" \
-p 1433:1433 \
--name sqlserver2025 \
-v ~/sqlserver data 2025:/var/opt/mssql \
-d mcr.microsoft.com/mssql/server:2025-latest
75b1580ab456409563cb923a2cacb8b6458ffedad2646e41901ea7e744547a8
mario@mario-EXT-Linux:~$ sudo docker ps -a # Ver contenedores
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
75b1580ab456   mcr.microsoft.com/mssql/server:2025-latest  "/opt/mssql/bin/laun..."  7 seconds ago  Up 6 seconds  0.0.0.0:1433->1433/tcp, [::]:1433->1433/tcp  sqlserver2025
1f608d678b9c   hello-world                                "/hello"                 5 days ago     Exited (0) 5 days ago                                     epic_moore
mario@mario-EXT-Linux:~$
```

Figura 1: Creación del contenedor en docker

Posteriormente, se estableció la conexión a la base de datos desde **Azure Data Studio**, tal como se muestra en la Figura 2:

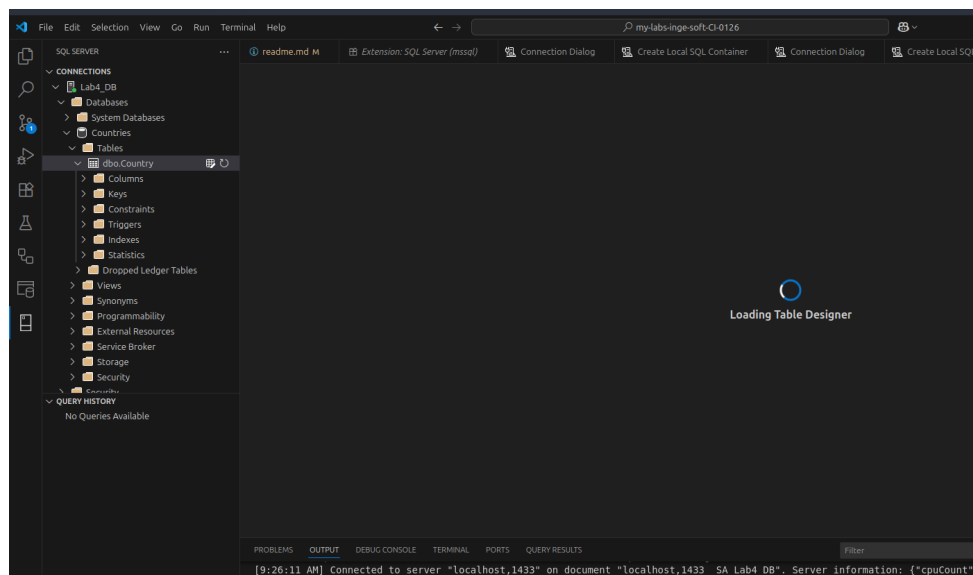


Figura 2: Conexión a la base de datos desde Azure Data Studio.

Una vez conectados, se procedió a la creación de la tabla en la base de datos, como se aprecia en la Figura 3:

```
mylab@mylab:~/EXT-Linux$ sqlcmd -S localhost,1433 -U SA -P "MyStrongPassword123!" -d Countries
1> CREATE TABLE Country(
2>   Id INT IDENTITY(1,1) PRIMARY KEY,
3>   Name VARCHAR(200) NOT NULL,
4>   Language VARCHAR(200) NOT NULL,
5>   Continent NVARCHAR(100) NOT NULL
6> );
7> GO
1> INSERT INTO Country (Name, Language, Continent) VALUES
2> INSERT INTO Country (Name, Language, Continent) VALUES
3>
4> ('Costa Rica', 'Español', 'América'),
5> ('Argentina', 'Español', 'América'),
6> ('Canada', 'Inglés/Francés', 'América'),
7> ('Francia', 'Francés', 'Europa'),
8> ('España', 'Español', 'Europa');
9> GO
Msg 136, Level 15, State 1, Server 75b1500ab456, Line 2
Incorrect syntax near the keyword 'INSERT'.
1> INSERT INTO Country (Name, Language, Continent) VALUES
2> ('Costa Rica', 'Español', 'América'),
3> ('Argentina', 'Español', 'América'),
4> ('Canada', 'Inglés/Francés', 'América'),
5> ('Francia', 'Francés', 'Europa'),
6> ('España', 'Español', 'Europa');
7> GO
(5 rows affected)
1> SELECT * FROM Country;
2> GO
10
Name
-----
1 Costa Rica
2 Argentina
3 Canada
4 Francia
5 España
(5 rows affected)
1>
```

Figura 3: Creación de tabla en la base de datos.

Después, se insertaron registros en la tabla Country y se verificaron mediante un SELECT \*, como se observa en la Figura 4:

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Lab4\_DB' database is expanded, showing the 'Country' table. The 'Query History' pane shows a list of queries, including 'SELECT \* FROM Country;'. The main pane displays the 'USE Countries; SELECT \* FROM dbo.Country;' query. The 'Query Results' pane at the bottom shows the results of the query, which is a table with 5 rows and 5 columns: Id, Name, Language, Continent, and Continent.

Id	Name	Language	Continent	Continent
1	Costa Rica	Español	América	América
2	Argentina	Español	América	América
3	Canada	Inglés/Francés	América	América
4	Francia	Francés	Europa	Europa
5	España	Español	Europa	Europa

Figura 4: Verificación de datos insertados con SELECT.

Finalmente, se ejecutó la aplicación y se utilizó **Swagger** para probar el endpoint `/country`, como se muestra en la Figura 5:

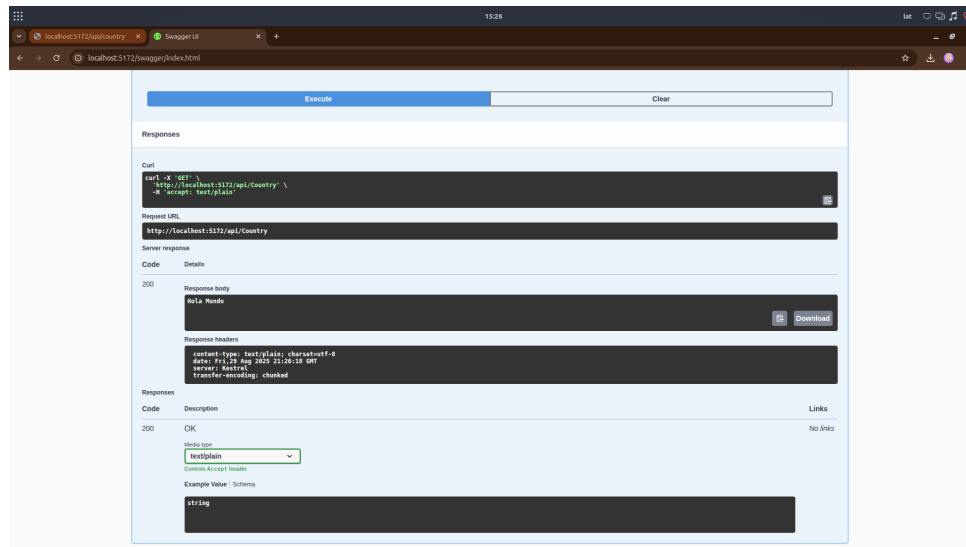


Figura 5: Prueba del endpoint `/country` en Swagger.

## 2. Investigación Técnica

### 2.1. Entity Framework vs Dapper

**Entity Framework** es un ORM (Object-Relational Mapping) completo desarrollado por Microsoft que permite a los desarrolladores trabajar con bases de datos utilizando objetos .NET en lugar de tener que escribir consultas SQL directamente. EF proporciona características como mapeo objeto-relacional, consultas LINQ, seguimiento de cambios, y migraciones de base de datos (Microsoft, 2023).

**Dapper** es un micro-ORM desarrollado por el equipo de Stack Overflow que se centra en el rendimiento y la simplicidad. A diferencia de Entity Framework, Dapper no proporciona todas las características de un ORM completo, pero ofrece un mapeo extremadamente rápido entre objetos .NET y resultados de bases de datos (Stack Overflow, 2023).

Entity Framework	Dapper
ORM completo	Micro-ORM
Mayor abstracción	Menor abstracción
Consultas LINQ	SQL tradicional
Migraciones automáticas	Sin migraciones
Mayor overhead	Alto rendimiento
Ideal para desarrollo rápido	Ideal para máximo rendimiento

Cuadro 1: Comparativa entre Entity Framework y Dapper

## 2.2. Postman

**Postman** es una plataforma de colaboración para el desarrollo de APIs que permite diseñar, probar, documentar y monitorizar APIs. Proporciona una interfaz gráfica para enviar solicitudes HTTP y analizar respuestas, lo que facilita el testing de endpoints durante el desarrollo (Postman, 2023).

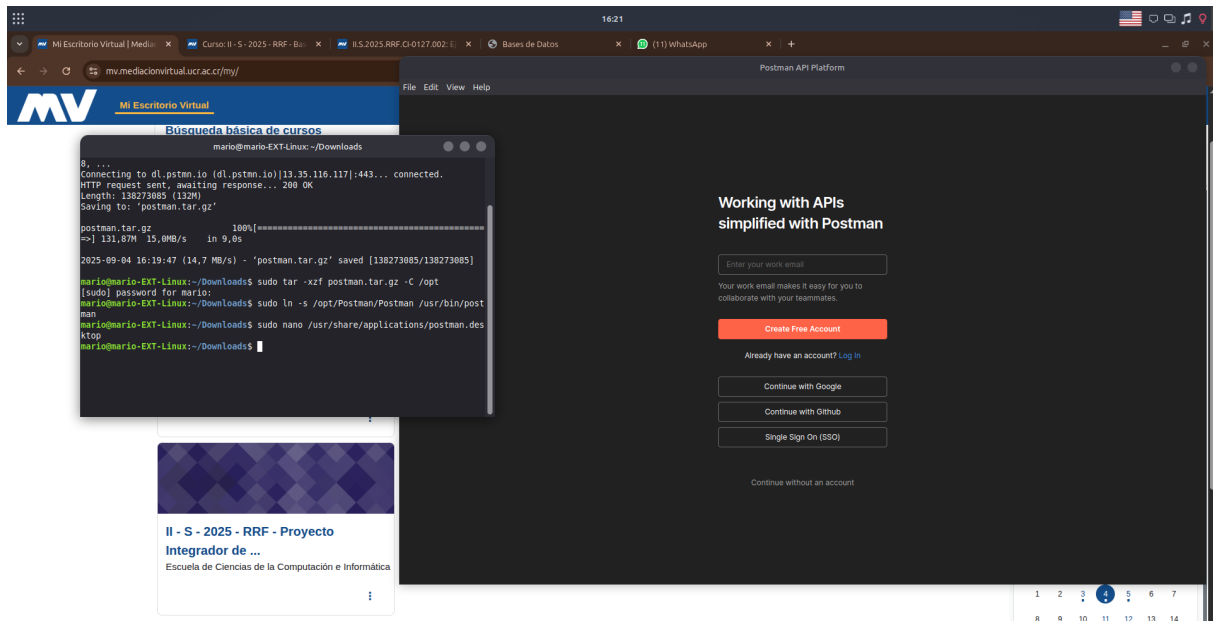


Figura 6: Instalación de Postman en Linux Ubuntu

Para la instalación en Linux se usaron los siguientes comandos:

### Terminal

```
cd /Downloads
wget https://dl.pstmn.io/download/latest/linux64 -O postman.tar.gz
sudo tar -xzf postman.tar.gz -C /opt
sudo ln -s /opt/Postman/Postman /usr/bin/postman
```

Luego se hizo un acceso directo en Linux, no obstante es un poco más elaborado y no tiene que ver con la descarga del programa. Postman será utilizado en clases posteriores para probar los endpoints del API desarrollada en este laboratorio.

### 3. Referencias

Microsoft. (2023). *Entity Framework Core*. Documentación oficial de Microsoft. Recuperado de <https://learn.microsoft.com/es-es/ef/core/>

Stack Overflow. (2023). *Dapper: A simple object mapper for .NET*. Repositorio oficial de GitHub. Recuperado de <https://github.com/StackExchange/Dapper>

Postman. (2023). *Postman Learning Center*. Recuperado de <https://learning.postman.com/>