



Neste artigo veremos como executar o [SQL Server](#) em um Contêiner Docker no ambiente [Linux](#).

Se você ainda não sabia o Microsoft SQL Server é suportado no ambiente Linux no [Ubuntu](#), [SUSE Linux Enterprise Server \(SLES\)](#) e [Red Hat Enterprise Linux \(RHEL\)](#). Ele também tem suporte como uma [imagem do Docker](#), que pode ser executado no mecanismo do Docker no Linux ou no Docker para Windows ou Mac.



Para o SQL Server 2017 os requisitos mínimos para Linux são:

Memória	2 GB
Sistema de Arquivos	XFS ou EXT4 (outros sistemas de arquivos, como BTRFS, não têm suporte)
Espaço em disco	6 GB
Velocidade do processador	2 GHz
Núcleos de processador	2 núcleos
Tipo de processador	compatível com x64 somente

Assim você pode instalar o SQL Server no Linux; veja [neste link](#) as instruções para instalar no [Ubuntu](#).



Neste primeiro contato com o SQL Server no ambiente Linux veremos como executar este banco de dados em um [contêiner Docker](#) usando a ferramenta [sqlcmd](#) para realizar algumas tarefas no SQL Server.

O ambiente Linux usado foi o seguinte:

1. [Linux - \(Distribuição Ubuntu LTS 18.04 Bionic Beaver\)](#)
2. [Docker](#)
3. [.NET Core SDK 2.1](#)

Todos esses recursos são gratuitos e são multiplataforma, e, a instalação em geral é bem simples.

Nota: Se você não conhece o Docker acompanhe a série de artigos : [Introdução básica ao Docker](#)

Baixando a imagem do SQL Server e criando o Contêiner

Vamos iniciar baixando a imagem do SQL Server que iremos executar no contêiner.

Obs: As informações foram obtidas em : [Guia de Início rápido: Imagens de contêiner executar o SQL Server no Docker](#)

As imagens para o SQL Server podem ser obtidas a partir do repositório [Docker hub](#) neste link: <https://hub.docker.com/r/microsoft/mssql-server-linux/>

Uma nova página contendo as imagens oficiais para o Microsoft SQL Server no Linux para Docker também esta disponível neste link: <https://hub.docker.com/r/microsoft/mssql-server>

Abra um terminal de comandos no Linux e digite o comando :

`docker pull microsoft/mssql-server-linux:2017-latest`

```
macoratti@linux: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
macoratti@linux:~$ docker pull microsoft/mssql-server-linux:2017-latest  
2017-latest: Pulling from microsoft/mssql-server-linux  
59ab41dd721a: Downloading 17.95MB/42.22MB  
57da90bec92c: Download complete  
06fe57530625: Download complete  
5a6315cba1ff: Download complete  
739f58768b3f: Download complete  
0b751601bca3: Download complete  
bcf04a22644a: Downloading 4.267MB/69.51MB  
6b5009e4f470: Downloading 4.291MB/281.1MB  
a9dca2f6722a: Waiting
```

Ao final a imagem será baixada no seu repositório local:

```
macoratti@linux: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
macoratti@linux:~$ docker pull microsoft/mssql-server-linux:2017-latest  
2017-latest: Pulling from microsoft/mssql-server-linux  
59ab41dd721a: Pull complete  
57da90bec92c: Pull complete  
06fe57530625: Pull complete  
5a6315cba1ff: Pull complete  
739f58768b3f: Pull complete  
0b751601bca3: Pull complete  
bcf04a22644a: Pull complete  
6b5009e4f470: Pull complete  
a9dca2f6722a: Pull complete  
Digest: sha256:9b700672670bb3db4b212e8aef841ca79eb2fce7d5975a5ce35b7129a9b90ec0  
Status: Downloaded newer image for microsoft/mssql-server-linux:2017-latest  
macoratti@linux:~$
```

Para visualizar informações da imagem digite no terminal o comando: **docker image ls** ou **docker images**

```

macoratti@linux: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
macoratti@linux:~$ docker image ls
REPOSITORY                TAG                IMAGE ID           CREATED            SIZE
microsoft/mssql-server-linux 2017-latest       314918ddaedf      2 days ago        1.35GB
microsoft/dotnet             2.1-aspnetcore-runtime db366d73508b      4 weeks ago       253MB
mysql                        5.7               ae6b78bedf88      4 weeks ago       372MB
macoratti@linux:~$

```

Vemos a imagem baixada com um tamanho de 1.35 GB.

Vamos agora criar um contêiner Docker para poder executar o [MS SQL Server](#).

No terminal de comandos digite (*em uma única linha*) :

```
docker run -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=Numsey@Password!' -p 1401:1433 -d
--name=mac_sql microsoft/mssql-server-linux:2017-latest
```

```

macoratti@linux: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
macoratti@linux:~$ docker run -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=Numsey@Password!' -p 1401:1433 -d --name=mac_sql
microsoft/mssql-server-linux:2017-latest

```

Destinchando o comando temos que:

<code>docker run</code>	Cria e executa o contêiner Docker
<code>-e 'ACCEPT_EULA=Y'</code>	Aceita os termos de licença da Microsoft
<code>-e 'SA_PASSWORD=Numsey@Password!'</code>	Define uma senha para o usuário SA
<code>-p 1401:1433</code>	mapeia a porta 1433 do contêiner para porta 1401 do host
<code>-d</code>	Executa o contêiner em segundo plano
<code>--name=mac_sql</code>	Define o nome do contêiner
<code>microsoft/mssql-server-linux:2017-latest</code>	Nome da imagem usada para criar o contêiner

Nota: Por padrão o MS SQL Server escuta na porta 1433

Ao executar o comando será criado o contêiner que fica em execução em segundo plano.

Para obter informações do contêiner criado digite no terminal: **docker container ps**

```
macoratti@linux: ~  
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda  
macoratti@linux:~$ docker run -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=Numsey@Password!' -p 1401:1433 -d --name=mac_sql microsoft/mssql-server-linux:2017-latest  
3126004e0089a7e0d77e7b4c149a4c3e57ef5c05383938c329cacca34eb5f146  
macoratti@linux:~$ docker container ps  
CONTAINER ID        IMAGE                                COMMAND                  CREATED             STATUS              PORTS  
3126004e0089        microsoft/mssql-server-linux:2017-latest  "/opt/mssql/bin/sqls..."  15 seconds ago      Up 12 seconds      0.0.0.0:1401->1433/tcp   mac_sql  
macoratti@linux:~$
```

Vemos o contêiner **mac_sql** criado a partir da imagem do MS SQL Server pronto para ser usado, e, agora podemos acessar o MS SQL Server no contêiner.

Vamos entrar no contêiner e acessar o SQL Server.

Digite o seguinte comando no terminal:

docker exec -it mac_sql /opt/mssql-tools/bin/sqlcmd -S localhost -U sa

```
macoratti@linux: ~  
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda  
macoratti@linux:~$ docker exec -it mac_sql /opt/mssql-tools/bin/sqlcmd -S localhost -U sa  
Password:  
1>
```

Destrinchando o comando:

<code>docker exec</code>	Executa um comando no contêiner em execução
<code>-it</code>	Usa o modo iterativo e anexa um terminal
<code>mac_sql</code>	Nome do contêiner

/opt/mssql-tools/bin/sqlcmd	Caminho da ferramenta de linha de comando sqlcmd no contêiner
-S localhost	Servidor localhost
-U sa	Usuário sa

Será solicitado para você informar a senha que foi definida na criação do contêiner : [Numsey@Password! \(lemgra!\)](#)

Vemos então aparecer prompt de comandos do **sqlcmd** : 1>

Podemos agora usar comandos **Transact-SQL** para testar o acesso ao MS SQL Server.

Digite:

1> **SELECT @@VERSION**

2> **GO**

Ao teclar **Enter** teremos o seguinte resultado:

```
macoratti@linux: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda

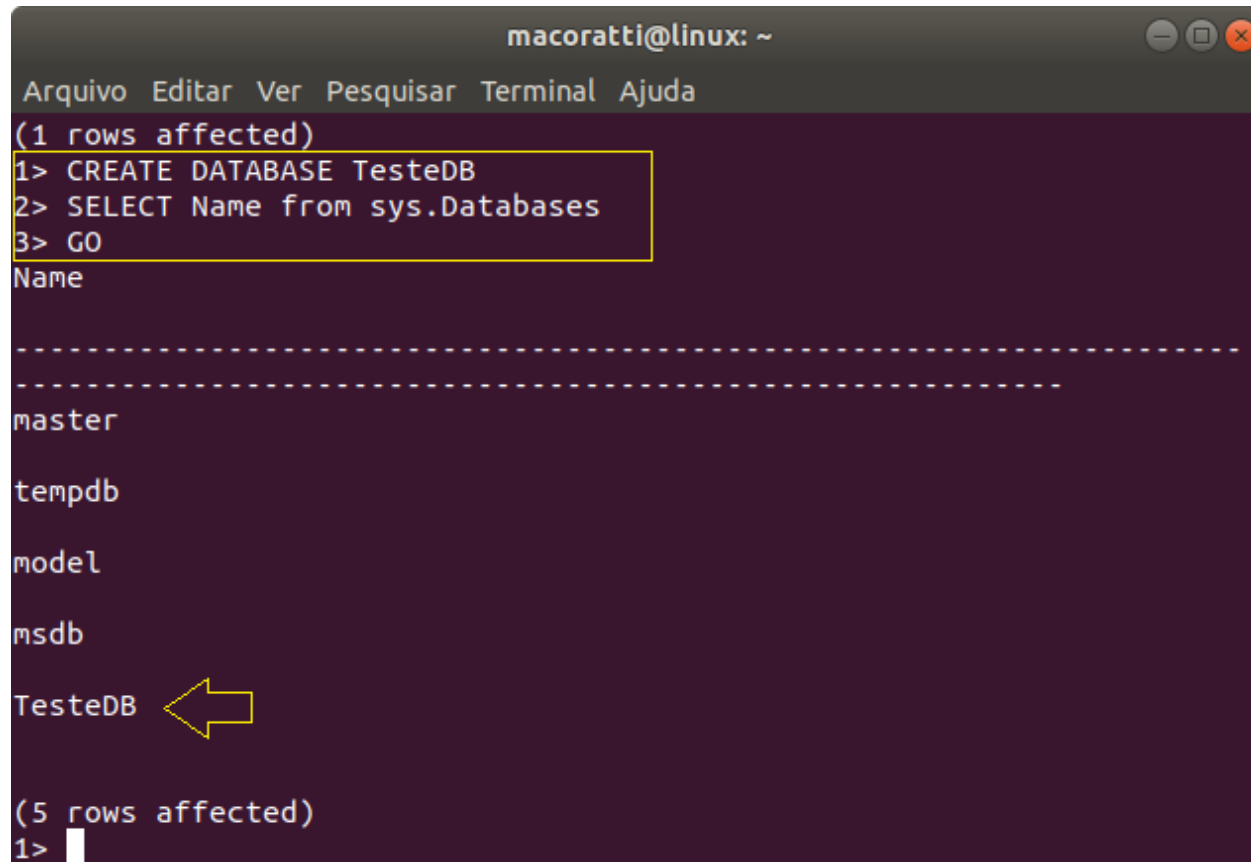
macoratti@linux:~$ docker exec -it mac_sql /opt/mssql-tools/bin/sqlcmd -S localhost -U sa
Password:
1> SELECT @@VERSION
2> GO

-----
Microsoft SQL Server 2017 (RTM-CU13) (KB4466404) - 14.0.3048.4 (X64)
Nov 30 2018 12:57:58
Copyright (C) 2017 Microsoft Corporation
Developer Edition (64-bit) on Linux (Ubuntu 16.04.5 LTS)

(1 rows affected)
1> 
```

Agora vamos criar um banco de dados e a seguir exibir os nomes de todos os bancos de dados no servidor usando os seguintes comandos Transact-SQL:

```
1> CREATE DATABASE TesteDB
2> SELECT Name from sys.Databases
3> GO
```

A screenshot of a terminal window titled 'macoratti@linux: ~'. The window has a menu bar with 'Arquivo', 'Editar', 'Ver', 'Pesquisar', 'Terminal', and 'Ajuda'. The terminal shows the execution of SQL commands. The first command is '1> CREATE DATABASE TesteDB', followed by '2> SELECT Name from sys.Databases', and '3> GO'. The output shows '(1 rows affected)' followed by a list of database names: 'Name', 'master', 'tempdb', 'model', 'msdb', and 'TesteDB'. A yellow arrow points to 'TesteDB'. Below this, the output shows '(5 rows affected)' and a new prompt '1>' with a cursor.

```
macoratti@linux: ~
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
(1 rows affected)
1> CREATE DATABASE TesteDB
2> SELECT Name from sys.Databases
3> GO
Name
-----
master
tempdb
model
msdb
TesteDB
(5 rows affected)
1> 
```

Vemos que o banco de dados **TesteDB** foi criado.

Vamos criar uma tabela chamada **Produtos** neste banco de dados com 3 campos:

- id - tipo int
- nome - tipo nvarchar(50)
- quantidade - int

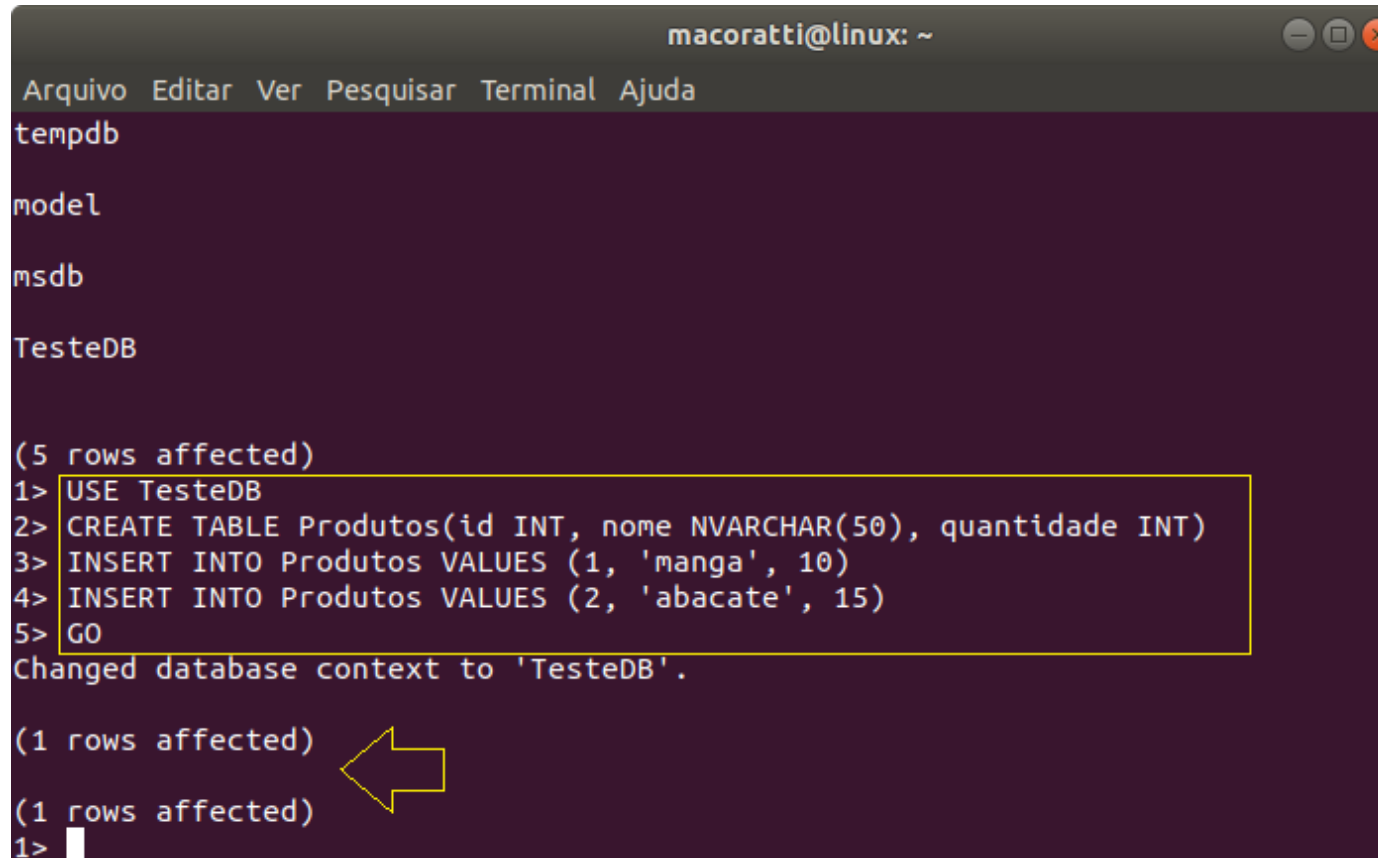
A instrução SQL usada será:

```
CREATE TABLE Produtos(id INT, nome NVARCHAR(50), quantidade INT)
```

A seguir vamos incluir dois produtos na tabela **Produtos** usando o comando **INSERT INTO**:

```
INSERT INTO Produtos VALUES (1, 'manga', 10);  
INSERT INTO Produtos VALUES (2, 'abacate', 15);
```

Para isso digite os comandos na linha de comando **sqlcmd** no contêiner:



```
macoratti@linux: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
tempdb  
model  
msdb  
TesteDB  
  
(5 rows affected)  
1> USE TesteDB  
2> CREATE TABLE Produtos(id INT, nome NVARCHAR(50), quantidade INT)  
3> INSERT INTO Produtos VALUES (1, 'manga', 10)  
4> INSERT INTO Produtos VALUES (2, 'abacate', 15)  
5> GO  
Changed database context to 'TesteDB'.  
  
(1 rows affected)  
(1 rows affected)  
1>
```

Para exibir os dados da tabela **Produtos** digite o comando abaixo no prompt do **sqlcmd** no contêiner:

```
SELECT * FROM Produtos WHERE quantidade > 5;  
GO
```



```
macoratti@linux: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda

(5 rows affected)
1> USE TesteDB
2> CREATE TABLE Produtos(id INT, nome NVARCHAR(50), quantidade INT)
3> INSERT INTO Produtos VALUES (1, 'manga', 10)
4> INSERT INTO Produtos VALUES (2, 'abacate', 15)
5> GO
Changed database context to 'TesteDB'.

(1 rows affected)

(1 rows affected)
1> SELECT * FROM Produtos WHERE quantidade > 5;
2> GO
id          nome          quantidade
-----
          1  manga          10
          2  abacate          15

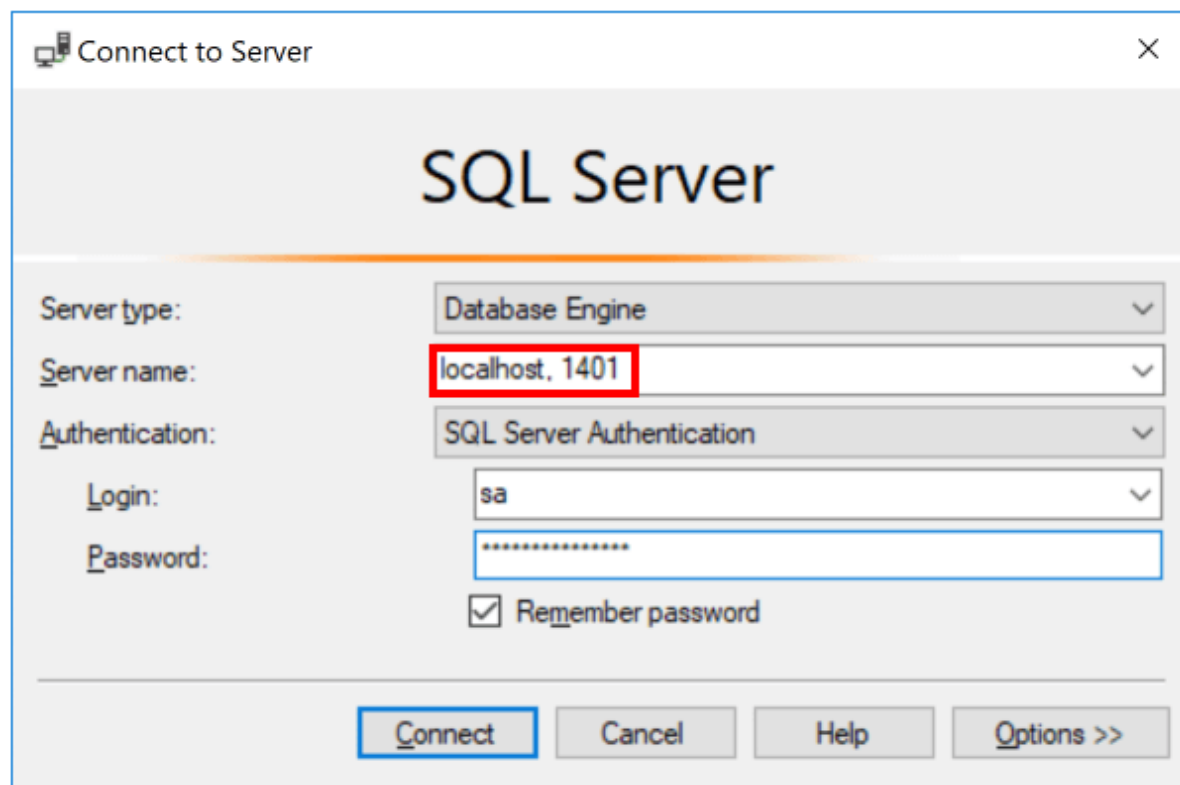
(2 rows affected)
1> 
```

Temos a exibição dos dados incluídos via comandos SQL usando a ferramenta [sqlcmd](#).

Para sair digite [QUIT](#).

Podemos acessar o contêiner via [SQL Server Management Studio](#) informando em [Server name](#) o servidor: [localhost](#) e a porta [1401](#), e , as credenciais de acesso nos campos [Login](#) e [Password](#).

Ficaria assim :



Existe também uma ferramenta chamada [SQL Operations Studio](#) que pode ser usada para testar o acesso no MS SQL Server no contêiner.

Dessa forma acessamos o SQL Server em um contêiner Docker no ambiente Linux.

Para parar e remover o contêiner podemos usar os seguintes comandos no terminal:

```
sudo docker container stop mac_sql
```

```
sudo docker container rm mac_sql
```

Em um próximo artigos vamos continuar a mostrar como usar o SQL Server no Linux.



Nota: Acompanhe a introdução ao Docker [nesta série de artigos](#) sobre o assunto.

"Eu sou o Alfa e o Ômega, o princípio e o fim, diz o Senhor, que é, e que era, e que há de vir, o Todo-Poderoso. "
[Apocalipse 1:8](#)

[Veja os Destaques e novidades do SUPER DVD Visual Basic \(sempre atualizado\) : clique e confira !](#)

Quer migrar para o VB .NET ?

- Veja mais sistemas completos para a plataforma .NET no [Super DVD .NET](#) , confira...
- [Curso Básico VB .NET - Vídeo Aulas](#)

Quer aprender C# ??

- Chegou o [Super DVD C#](#) com exclusivo material de suporte e vídeo aulas com curso básico sobre C#.
- [Curso C# Basico - Video Aulas](#)

Quer aprender os conceitos da Programação Orientada a objetos ?

- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) 

Quer aprender o gerar relatórios com o ReportViewer no VS 2013 ?

- [Curso - Gerando Relatórios com o ReportViewer no VS 2013 - Vídeo Aulas](#)

Gostou ?  [Compartilhe no Facebook](#)  [Compartilhe no Twitter](#)

Referências:

- [Seção VB .NET do Site Macoratti.net](#)
- [Super DVD .NET - A sua porta de entrada na plataforma .NET](#)
- [Super DVD Vídeo Aulas - Vídeo Aula sobre VB .NET, ASP .NET e C#](#)
- [Seção C# do site Macoratti.net](#)
- [Super DVD C#](#)
- [Super DVD Visual Basic](#)

- [Curso Básico VB .NET - Vídeo Aulas](#)
- [Curso C# Básico - Vídeo Aulas](#)
- [.NET - Apresentando o AngularJS - Macoratti.net](#)
- [ASP.NET MVC - Usando o AngularJS - I - Macoratti.net](#)
- [Entity Framework - Conceitos Básicos - Uma visão geral - Macoratti](#)
- [Seção Entity Framework - Macoratti .NET](#)
- <https://docs.microsoft.com/en-us/ef/core/>
- <https://docs.microsoft.com/en-us/ef/core/get-started/netcore/new-db-sqlite>
- <https://www.microsoft.com/net/download/linux-package-manager/ubuntu18-04/sdk-2.1.403>
- <https://www.ubuntu.com/download/desktop>
- <https://www.virtualbox.org/>

[José Carlos Macoratti](#)