

# **DOCUMENTO “LOGIMOVE”**

Desenvolvedor : André Luis Gonçalves Carvalhal

Matrícula: 202203185403

## **BRIEFING:**

Esta atividade tem por objetivo desenvolver um banco de dados no Azure SQL para a LogiMove Transportes, uma empresa de logística. O objetivo é migrar de um sistema baseado em papel para uma solução digital, utilizando autenticação digital para melhor coordenação e rastreamento de remessas. O banco de dados armazenará informações sobre motoristas, clientes e pedidos.

### **1. Configuração do Ambiente Azure :**

- Criar uma conta no Azure.

- Configurar uma instância do Azure SQL Database.

- Estabelecer os parâmetros de segurança, como firewalls e regras de acesso.

### **2. Design do Banco de Dados:**

Definir a arquitetura do banco de dados considerando as necessidades da empresa.

Criar um diagrama de entidade-relacionamento (ER) para visualizar as relações entre as tabelas.

### **3. Implementação do Banco de Dados:**

Utilizar T-SQL para criar tabelas, definir chaves primárias, chaves estrangeiras e índices.

## 1. Configuração do Ambiente Azure

A conta de aluno do Microsoft Azure foi utilizada para a criação do banco de dados.

Servidor criado com acesso público (Para caso o professor deseje executar testes), com conexão via login de SQL.

Endereço do Servidor: *logimovedb.database.windows.net*

Porta: *1433*

Usuário: *logimove*

Senha: *@estacio123*

### LINK JDBC:

*jdbc:sqlserver://logimovedb.database.windows.net:1433;database=LogiMove*

*DB;user=logimove@logimovedb;password={@estacio123};encrypt=true;trustServerCertificate=false;hostNameInCertificate=\*.database.windows.net;loginTimeout=30;*

### LINK ODBC:

*Driver={ODBC Driver 18 for SQL*

*Server};Server=tcp:logimovedb.database.windows.net,1433;Database=LogiMove*

*DB;Uid=logimove;Pwd={@estacio123};Encrypt=yes;TrustServerCertificate=no;Connection Timeout=30;*

Regra de Firewall adicionada, IP local do desenvolvedor.

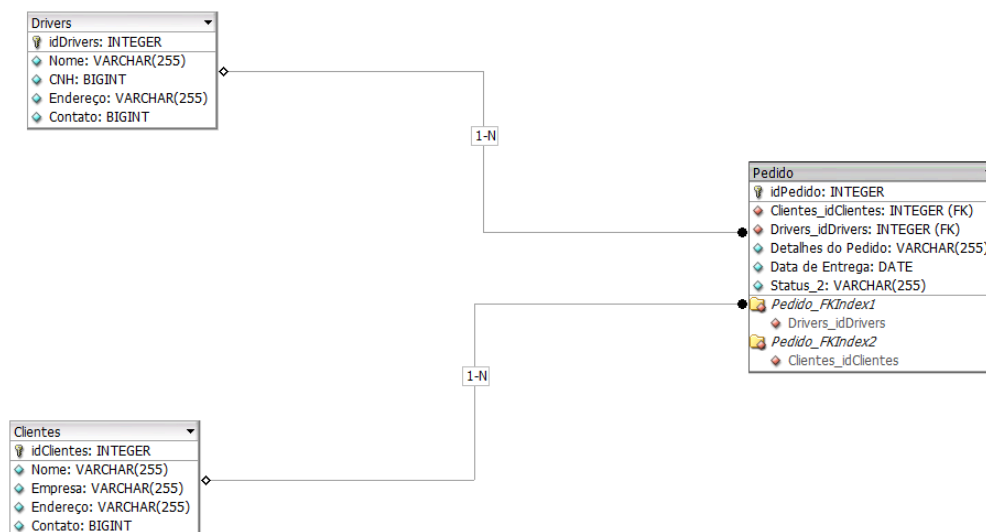
## 2. Design do Banco de Dados:

Analisando o projeto. Tendo em vista que os pedidos recebem tem motoristas que podem se repetir em diversos deles, criando um padrão de 1 para N. Também no caso de clientes, que podem ter N pedidos. Então o projeto se adequa a um padrão relacional. E também por sugestão do roteiro, o modelo de arquitetura

**RELACIONAL** foi utilizado.

# Preview do arquivo *Diagrama ER.xml*

feito com DBDesigner, encontra-se no diretório do Github.



### ***T-SQL Commands para implementação:***

```
sqlcmd -S tcp:logimovedb.database.windows.net,1443 -d "LogiMove DB"  
-U logimove -P "@estacio123" -N -I 30
```

```
CREATE TABLE Drivers (idDrivers int PRIMARY KEY IDENTITY(1,1),  
Nome varchar(255) NOT NULL, CNH BIGINT NOT NULL, Endereço  
varchar(255) NOT NULL, Contato BIGINT NOT NULL);
```

```
CREATE TABLE Clientes (idClientes int PRIMARY KEY IDENTITY(1,1),  
Nome varchar(255) NOT NULL, Empresa varchar(255) NOT NULL,  
Endereço varchar(255) NOT NULL, Contato BIGINT NOT NULL);
```

```
CREATE TABLE Pedido (idPedido int PRIMARY KEY IDENTITY(1,1), Data  
DATE NOT NULL, Detalhes_do_Pedido varchar(255) NOT NULL,  
idClientes int, idDrivers int, CONSTRAINT FK_idClientes FOREIGN KEY  
(idClientes) REFERENCES Clientes(idClientes), CONSTRAINT  
FK_idDrivers FOREIGN KEY (idDrivers) REFERENCES  
Drivers(idDrivers));
```

### ***Resultado:***

```
1> SELECT name FROM sys.tables  
2> GO  
name  
-----  
Drivers  
Clientes  
Pedido  
  
(3 rows affected)  
1> █
```

Ao criar IDs como PRIMARY KEY já foi criado um Index Clusterizado para as tabelas. Dispensando a criação de um novo index.

### ***T-SQL Commands dos Testes:***

Executados no Azure Data Studio para melhor visualização.

```
INSERT INTO Drivers (Nome, CNH, Endereço, Contato) VALUES  
( 'Teste', 12345678911, 'Rua Teste', 21999999999);
```

```
SELECT * FROM Drivers;
```

Results		Messages			
	idDrivers	Nome	CNH	Endereço	Contato
1	1	Teste	12345678911	Rua Teste	21999999999

```
INSERT INTO Clientes (Nome, Empresa, Endereço, Contato) VALUES  
( 'Cliente Teste', 'Empresa Teste', 'Endereço Teste', 21999999999);  
SELECT * FROM Clientes;
```

Results		Messages			
	idClientes	Nome	Empresa	Endereço	Contato
1	1	Cliente Teste	Empresa Teste	Endereço Teste	21999999999

```
INSERT INTO Pedido (Data, Detalhes_do_Pedido, idClientes, idDrivers)  
VALUES ( '2024-05-31', '1 Saco de Cimento', 1, 1);  
SELECT * FROM Pedido;
```

	idPedido	Data	Detalhes_do_Pedido	idClientes	idDrivers
1	1	2024-05-31	1 Saco de Cimento	1	1