



**UNIVERSIDADE ESTÁCIO DE SÁ  
POLO ALCANTARA - SÃO GONÇALO/RJ**

**TECNOLOGIA EM DESENVOLVIMENTO FULL STACK  
RPG0026- TIRANDO PROVEITO DA NUVEM**

**RELATÓRIO DA MISSÃO PRÁTICA**

**1 INTRODUÇÃO**

Nesta missão prática, o objetivo foi aplicar conhecimentos sobre desenvolvimento de bancos de dados utilizando o Azure SQL. A prática envolveu desde a configuração do ambiente no Azure até a implementação e validação de um banco de dados com tabelas relacionadas. Este relatório detalha as etapas seguidas, os métodos utilizados e os resultados alcançados durante o processo.

**OBJETIVOS**

- Configurar e provisionar um banco de dados no Azure SQL.
- Projetar e implementar um banco de dados relacional utilizando T-SQL.
- Realizar operações CRUD (Create, Read, Update, Delete) para manipulação de dados.

**CONFIGURAÇÃO DO AMBIENTE AZURE**

**.1 Criação da Conta no Azure**

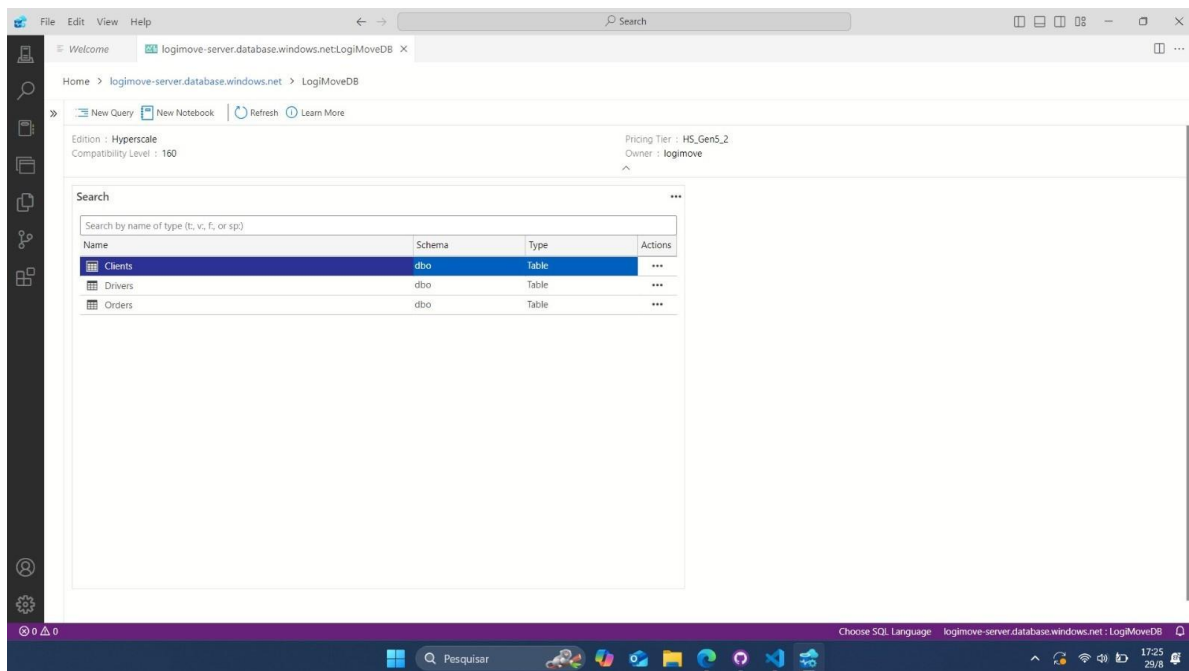
Foi necessário criar uma conta no Azure para acessar os recursos de nuvem. A configuração da conta seguiu os passos básicos de cadastro na plataforma.

## .2 Configuração do Azure SQL Database

Uma instância do Azure SQL Database foi provisionada com as seguintes características:

- ✧ Nome do Banco de Dados: LogiMoveDB
  - ✧ Servidor: logimove-server
  - ✧ Região: Brasil
- 
- ✧ Nível de Preço: Configurado inicialmente no nível Básico, adequado para práticas e testes.

**Imagem 1: Banco de dados no Azure Data Studio**



## DESIGN DO BANCO DE DADOS

### 1.2 Estrutura do Banco de Dados

O banco de dados foi projetado para incluir as seguintes tabelas principais:

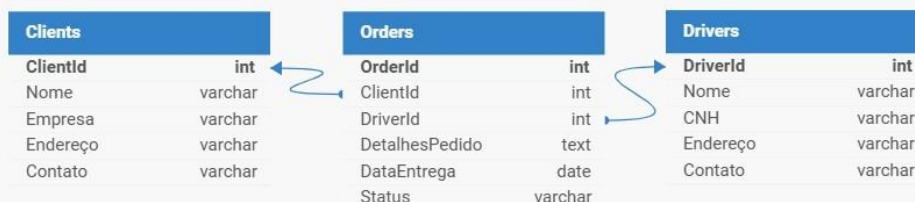
- ✧ Drivers (Motoristas): Tabela para armazenar informações de motoristas, como nome, CNH, endereço e contato.
- ✧ Clients (Clientes): Tabela para armazenar informações de clientes, como nome, empresa, endereço e contato.
- ✧ Orders (Pedidos): Tabela para armazenar informações de pedidos, incluindo cliente, motorista, detalhes do pedido, data de entrega e status.

### 1.3 Diagrama de Entidade-Relacionamento (ER)

Foi desenvolvido um diagrama ER para visualizar e planejar as relações entre as tabelas:

- ✧ Drivers (1:N) Orders: Um motorista pode estar associado a vários pedidos.
- ✧ Clients (1:N) Orders: Um cliente pode ter vários pedidos registrados.

**Imagem 2: Modelagem do banco**



## IMPLEMENTAÇÃO DO BANCO DE DADOS

### 1.2 Criação das Tabelas

As tabelas foram criadas utilizando T-SQL com a seguinte estrutura:

#### Código 1: Tabela de Motoristas (Drivers)

```
CREATE TABLE Drivers (  
    DriverID INT PRIMARY KEY,  
    Nome VARCHAR(100) NOT NULL,  
    CNH VARCHAR(20) NOT NULL,  
    Endereço VARCHAR(200),  
    Contato VARCHAR(50)  
);
```

#### Código 2: Tabela de Clientes (Clients)

```
CREATE TABLE Clients (  
    ClientID INT PRIMARY KEY,  
    Nome VARCHAR(100) NOT NULL,  
    Empresa VARCHAR(100),  
    Endereço VARCHAR(200),  
    Contato VARCHAR(50)  
);
```

#### Código 3: Tabela de Pedidos (Orders)

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY,  
    ClientID INT NOT NULL,  
    DriverID INT NOT NULL,  
    DetalhesPedido TEXT,  
    DataEntrega DATE,  
    Status VARCHAR(50),  
    FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),  
    FOREIGN KEY (DriverID) REFERENCES Drivers(DriverID)  
);
```

## INSERÇÃO E GESTÃO DE DADOS

### 1.3 Inserção de Dados de Teste

Foram inseridos dados de teste nas tabelas para simular cenários práticos.

#### Código 4: Exemplo de Inserção de Dados

```
INSERT INTO Drivers (Nome, CNH, Endereço, Contato) VALUES  
('João Silva', '123456789', 'Rua A, 100', 'joao@example.com');
```

## EXECUÇÃO E VALIDAÇÃO DE CONSULTAS

### 1.4 Consultas T-SQL

Consultas SQL foram executadas para validar o funcionamento das tabelas e dos relacionamentos criados:

#### Código 5: Consulta de Pedidos por Cliente

```
SELECT Orders.OrderID, Clients.Nome AS Cliente, Drivers.Nome AS  
Motorista, Orders.Status  
FROM Orders  
JOIN Clients ON Orders.ClientID = Clients.ClientID  
JOIN Drivers ON Orders.DriverID = Drivers.DriverID;
```

### Código 6: Atualização do Status de Pedido

```
UPDATE Orders  
SET Status = 'Entregue'  
WHERE OrderID = 1;
```

### Código 7: Exclusão de Pedido

```
DELETE FROM Orders  
WHERE OrderID = 2;
```

## OPERAÇÕES CRUD

As operações CRUD (Create, Read, Update, Delete) foram executadas com sucesso para garantir que o banco de dados está funcionando corretamente. Essas operações permitiram a manipulação dos dados inseridos, confirmando a integridade e a funcionalidade do banco de dados.

## CONCLUSÃO

A missão prática foi concluída com sucesso. O banco de dados no Azure SQL foi configurado, implementado e validado de acordo com os objetivos propostos. As operações CRUD foram testadas e funcionaram conforme esperado, demonstrando a eficiência e a confiabilidade do banco de dados.

### 1.5 Próximos Passos

- ✧ Explorar a integração do banco de dados com aplicações web ou móveis.
- ✧ Realizar testes de desempenho e otimização para cenários de alta carga.
- ✧ Ampliar o banco de dados para incluir funcionalidades adicionais, como relatórios e análises.