



Universidade Estácio  
Campus Manaus Curso de Desenvolvimento Full Stack  
Relatório da Missão Prática 4 - Mundo 4

Disciplina: RPG 0026 - Tirando proveito da nuvem para projetos de software

Tirando proveito da nuvem para projetos de software

1. Missão Prática: “Tirando proveito da nuvem para projetos de software”

2. Objetivos da Prática

- Demonstrar habilidade na criação e gerenciamento de recursos na Nuvem Azure, adquirindo conhecimento sobre a estrutura básica da plataforma Azure
- Utilizar e efetivamente o portal Azure para criar e reconfigurar uma Máquina Virtual (VM), demonstrando compreensão dos recursos e suas funções.
- Configurar regras de rede e grupos de segurança, adquirindo conhecimento sobre a estrutura das regras de rede na Nuvem Azure.
- Importar um arquivo .bacpac para um banco de dados no Banco de Dados SQL do Azure;
- Criar e configurar um aplicativo web no Azure, demonstrando compreensão do mecanismo de hospedagem e implantação de aplicativos web.

Endereço do repositório: <https://github.com/finntroll89/mundo-4-missao-pratica-4.git>

3. Configuração da máquina virtual e do banco de dados na nuvem Azure

Fig.1: Criação da máquina virtual na nuvem Azure.

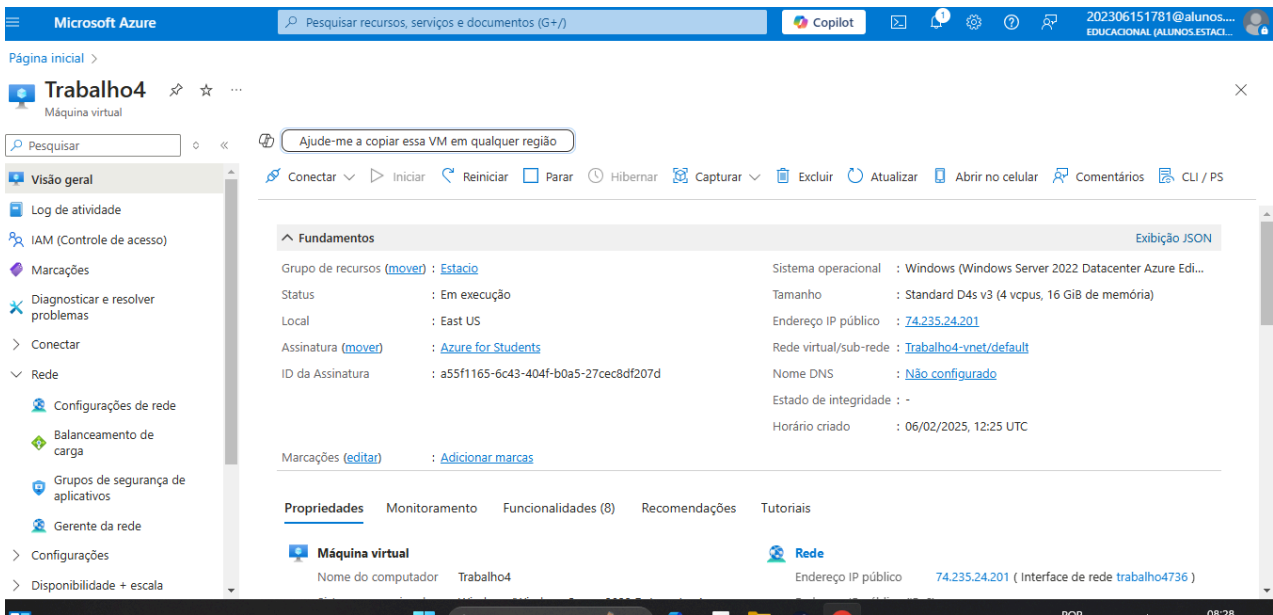
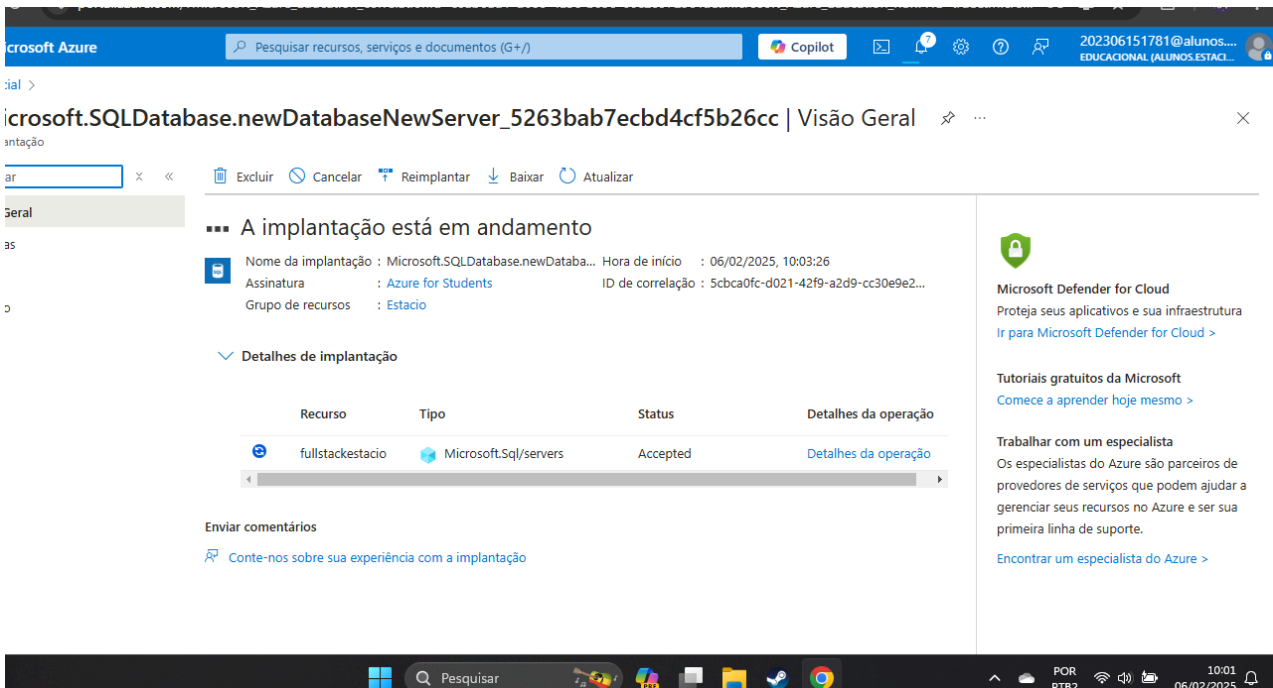


Fig.2: Criação do banco de dados SQL na nuvem Azure.



#### 4. Script do banco de dados

String de conexão: sqlcmd -S tcp:sqlstacio.database.windows.net,1433 -d "LogiMove Transportes" -U alexpaz -P "pppp3talliC@" -N -I 30

Comandos SQL:

Criar a tabela de Motoristas

```
CREATE TABLE Drivers (  
    DriverID INT PRIMARY KEY,  
    Nome VARCHAR(100),  
    CNH VARCHAR(20),  
    Endereço VARCHAR(200),  
    Contato VARCHAR(50)  
);  
GO
```

-- Criar a tabela de Clientes

```
CREATE TABLE Clients (  
    ClientID INT PRIMARY KEY,  
    Nome VARCHAR(100),  
    Empresa VARCHAR(100),  
    Endereço VARCHAR(200),  
    Contato VARCHAR(50)  
);  
GO
```

-- Criar a tabela de Pedidos

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY,  
    ClientID INT,  
    DriverID INT,  
    DetalhesPedido TEXT,  
    DataEntrega DATE,  
    Status VARCHAR(50),  
    FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),  
    FOREIGN KEY (DriverID) REFERENCES Drivers(DriverID)  
);  
GO
```

-----  
----

-- Inserir motoristas

```
INSERT INTO Drivers (DriverID, Nome, CNH, Endereço, Contato)  
VALUES  
    (1, 'Pedro Almeida', '9876543210', 'Av. Paulista, 1500, SP', '11987456321'),  
    (2, 'Juliana Costa', '0123456789', 'Rua do Ouvidor, 50, RJ', '21965412387');  
GO
```

-- Inserir clientes

```
INSERT INTO Clients (ClientID, Nome, Empresa, Endereço, Contato)  
VALUES  
    (1, 'Maria Santos', 'Comércio Mineiro', 'Av. Afonso Pena, 2000, MG', '31912345678'),  
    (2, 'Lucas Pereira', 'Indústria Gaúcha', 'Rua da Praia, 350, RS', '51987654321');  
GO
```

```
-- Inserir pedidos
INSERT INTO Orders (OrderID, ClientID, DriverID, DetalhesPedido, DataEntrega,
Status)
VALUES
    (1, 1, 1, 'Entrega de 10 caixas', '2025-02-10', 'Pendente'),
    (2, 2, 2, 'Transporte de carga refrigerada', '2025-02-12', 'Em andamento');
GO
```

```
-----
-- Listar motoristas
SELECT * FROM Drivers;
GO
```

```
-- Listar clientes
SELECT * FROM Clients;
GO
```

```
-- Listar pedidos
SELECT * FROM Orders;
GO
```

```
-- Consultar pedidos de um cliente específico
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;
GO
```

```
-----
-- Atualizar status do pedido 1 para "Entregue"
UPDATE Orders SET Status = 'Entregue' WHERE OrderID = 1;
GO
```

```
-- Excluir um cliente (caso não tenha pedidos relacionados)
DELETE FROM Clients WHERE ClientID = 2;
GO
```

```
-- Verificar se a tabela de pedidos está vazia
SELECT COUNT(*) FROM Orders;
GO
```

```
-----
-- Inserir novo cliente
INSERT INTO Clients (ClientID, Nome, Empresa, Endereço, Contato)
VALUES (99, 'João Silva', 'Translog', 'Rua das Flores, 123', '(11) 99999-9999');
GO
```

Fig. 3: Resultado da execução dos scripts de inserção.

```

1> INSERT INTO Drivers (DriverID, Nome, CNH, Endereço, Contato) VALUES (1, 'Pedro Almeida', '9876543210', 'Av. Paulista, 1500, SP', '11987456321');
2> go
Msg 2627, Level 14, State 1, Server sqlestacio, Line 1
Violation of PRIMARY KEY constraint 'PK_Drivers__F1B1CD2491ABB52D'. Cannot insert duplicate key in object 'dbo.Drivers'. The duplicate key value is (1).
The statement has been terminated.
1>

```

Fig. 4: Resultado da execução dos scripts de consulta.

```

1> SELECT * FROM Clients;
2> GO
ClientID      Nome                                     Endereço                                     Empresa
-----
1  Maria Santos                               Av. Afonso Pena, 2000, MG                               Comércio Mineiro
31912345678
2  Lucas Pereira                               Rua da Praia, 350, RS                               Indústria Gaúcha
51987654321
(2 rows affected)
1>

```

```

2> SELECT * FROM Orders;
3> GO
OrderID      ClientID      DriverID      DetalhesPedido                                     DataE
ntrega      Status
-----
1            1            1            1 Entrega de 10 caixas
2025-02-10  Pendente
2            2            2            2 Transporte de carga refrigerada
2025-02-12  Em andamento
(2 rows affected)
1>

```

Micro atividades .

Fig. 5: Criação banco de dados .

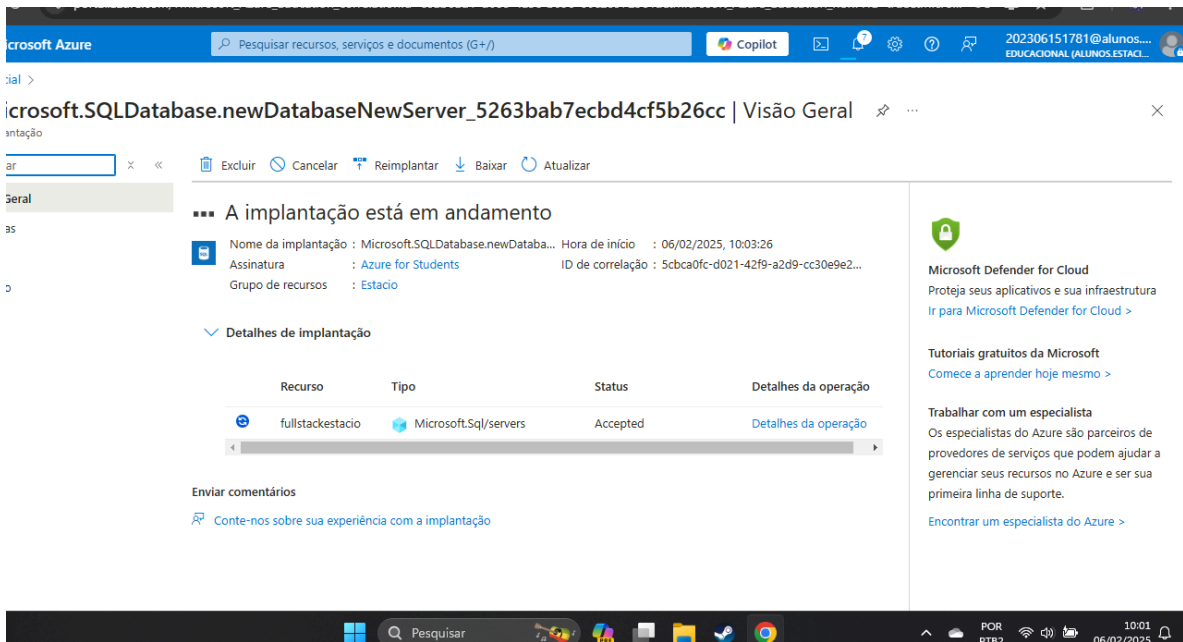


Fig. 6: Configuração de postas:

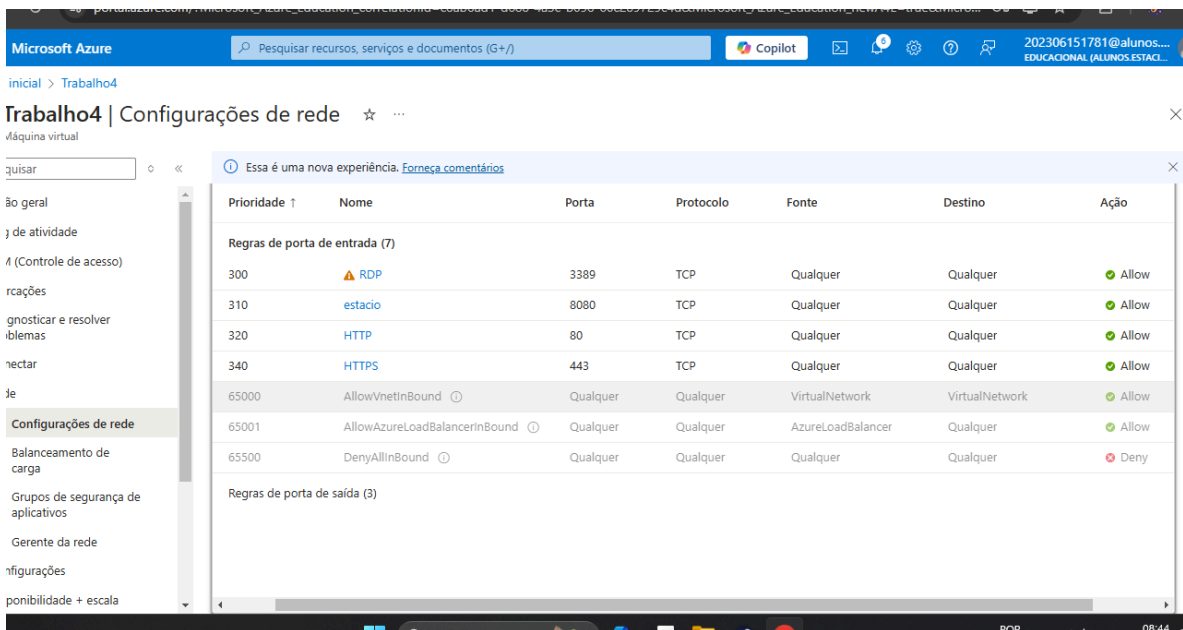


Fig. 7: Criando Maquina virtual.

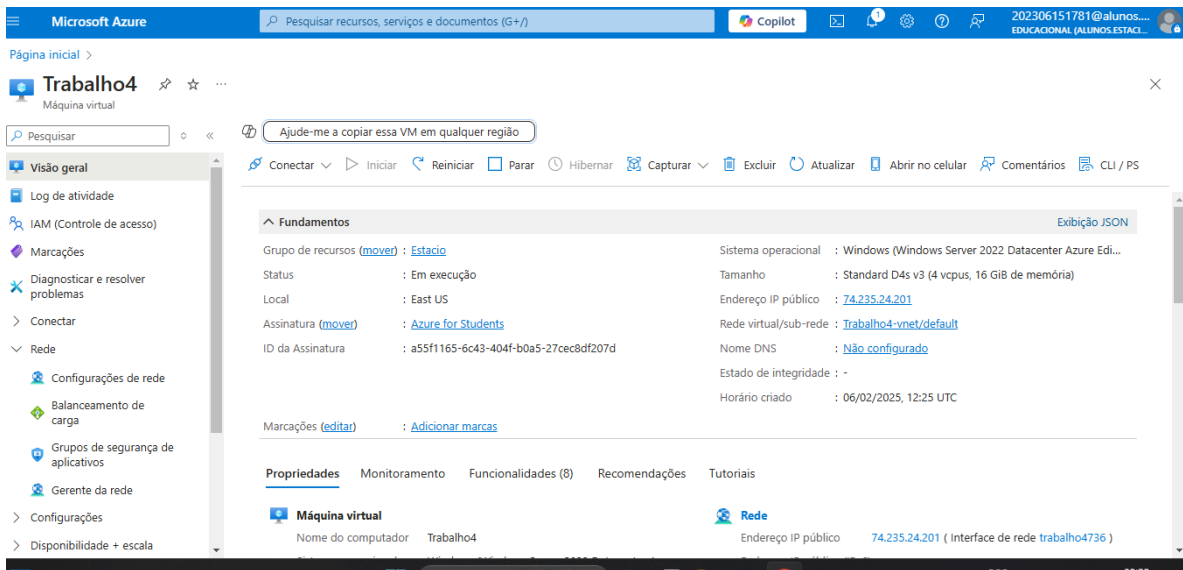


Fig. 8: Conexão com sql .

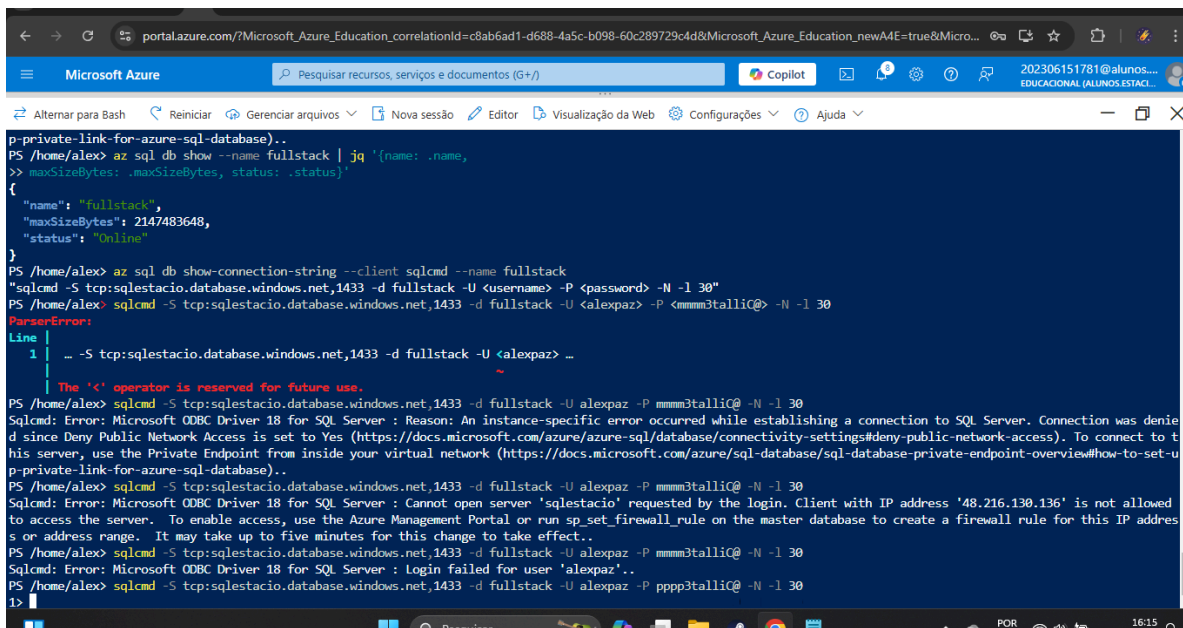


Fig. 9: Inserção

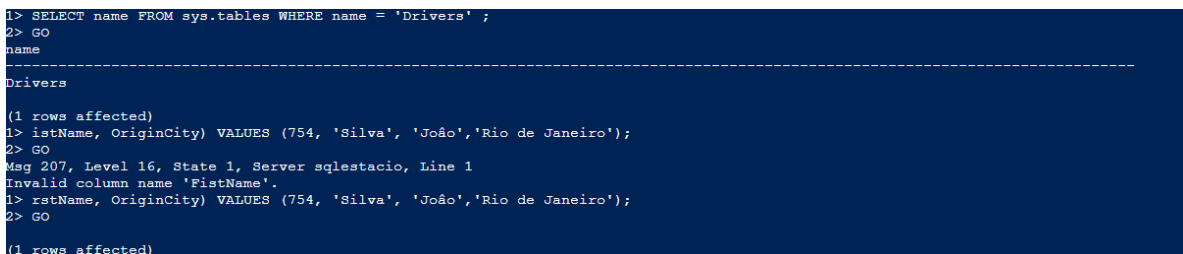


Fig. 10: Consultar sql .

```
VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/alex> az configure --defaults group=[resource-group] sql-server=[server-name]
PS /home/alex>
PS /home/alex> az configure --defaults group=Estacio sql-server=sqlestacio
PS /home/alex> az sql db list
{
  "autoPauseDelay": null,
  "availabilityZone": "NoPreference",
  "catalogCollation": "SQL_Latin1_General_CP1_CI_AS",
  "collation": "SQL_Latin1_General_CP1_CI_AS",
  "createMode": null,
  "creationDate": "2025-02-06T17:44:51.783000+00:00",
  "currentBackupStorageRedundancy": "Local",
  "currentServiceObjectiveName": "Basic",
  "currentSku": {
    "capacity": 5,
    "family": null,
    "name": "Basic",
    "size": null,
    "tier": "Basic"
  },
  "databaseId": "703064f2-cb00-4d94-acf2-0fc6d524347c",
  "defaultSecondaryLocation": "centralus",
  "earliestRestoreDate": "2025-02-06T17:58:41+00:00",
  "edition": "Basic",
  "elasticPoolId": null,
```

Fig. 11: Deletar .

```
(1 rows affected)
1> DELETE FROM Drivers WHERE DriverID = 745;
2> GO

(1 rows affected)
```

Fig. 12: Leitura .

```
(1 rows affected)
1> SELECT DriverID, OriginCity FROM Drivers;
2> GO
DriverID      OriginCity
-----
754 Rio de Janeiro

(1 rows affected)
1> 
```

Fig. 13: Deletar e checar lista .



```
(1 rows affected)
1> DELETE FROM Drivers WHERE DriverID = 745;
2> GO

(1 rows affected)
1> SELECT COUNT(*) FROM Drivers;
2> GO

-----
          1

(1 rows affected)
1> tName, OriginCity) VALUES (745, 'Souza', 'Carlos', 'Belo Horizonte');
2> GO

(1 rows affected)
1> e, FirstName, OriginCity) VALUES (746, 'Basto', 'Raimundo', 'Belem');
2> GO

(1 rows affected)
1> SELECT COUNT(*) FROM Drivers;
2> GO

-----
          3

(1 rows affected)
```