



# 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 para um sistema de **locadora de jogos**. Este relatório detalha as etapas seguidas, os métodos utilizados e os resultados alcançados durante o processo.

---

## 2. 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.
- 

## 3. CONFIGURANDO O AMBIENTE AZURE

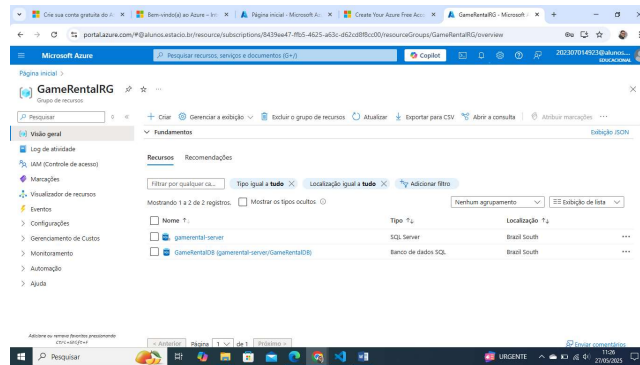
### 3.1 Criação da Conta no Azure

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.

### 3.2 Configuração do Azure SQL Database

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

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



## 4. DESIGN DO BANCO DE DADOS

### 4.1 Estrutura do Banco de Dados

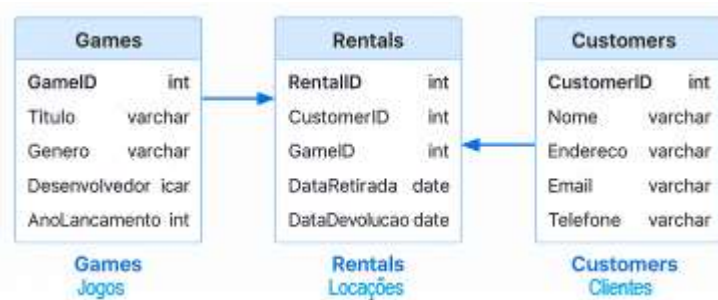
O banco de dados foi projetado para incluir as seguintes tabelas principais:

- **Games (Jogos):** Tabela para armazenar informações sobre os jogos, como título, gênero, desenvolvedor e ano de lançamento.
- **Customers (Clientes):** Tabela para armazenar informações dos clientes, como nome, endereço, e-mail e telefone.
- **Rentals (Locações):** Tabela para armazenar informações das locações, incluindo o cliente, o jogo alugado, data de retirada, data de devolução e status (devolvido, pendente, atrasado).

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

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

- **Customers (1:N) Rentals:** Um cliente pode ter várias locações.
- **Games (1:N) Rentals:** Um jogo pode estar presente em várias locações ao longo do tempo.



## 5. IMPLEMENTAÇÃO DO BANCO DE DADOS

### 5.1 Criação das Tabelas

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

### Código 1: Tabela de Jogos (Games)

sql

```
CREATE TABLE Games (  
    GameID INT PRIMARY KEY,  
    Titulo VARCHAR(100) NOT NULL,  
    Genero VARCHAR(50) NOT NULL,  
    Desenvolvedor VARCHAR(100),  
    AnoLancamento INT  
);
```

```
CREATE TABLE Games (  
    GameID INT PRIMARY KEY,  
    Titulo VARCHAR(100) NOT NULL,  
    Genero VARCHAR(50) NOT NULL,  
    Desenvolvedor VARCHAR(100),  
    AnoLancamento INT  
);
```

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

sql

```
CREATE TABLE Customers (  
    CustomerID INT PRIMARY KEY,  
    Nome VARCHAR(100) NOT NULL,  
    Endereco VARCHAR(200),  
    Email VARCHAR(100),  
    Telefone VARCHAR(20)  
);
```

```
CREATE TABLE Customers (  
    CustomerID INT PRIMARY KEY,  
    Nome VARCHAR(100) NOT NULL,  
    Endereco VARCHAR(200),  
    Email VARCHAR(100),  
    Telefone VARCHAR(20)  
);
```

### Código 3: Tabela de Locações (Rentals)

sql

```
CREATE TABLE Rentals (  
    RentalID INT PRIMARY KEY,  
    CustomerID INT NOT NULL,  
    GameID INT NOT NULL,  
    DataRetirada DATE,  
    DataDevolucao DATE,  
    Status VARCHAR(50),  
    FOREIGN KEY (CustomerID) REFERENCES  
Customers(CustomerID),  
    FOREIGN KEY (GameID) REFERENCES Games(GameID)  
);
```

```
CREATE TABLE Rentals (  
    RentalID INT PRIMARY KEY,  
    CustomerID INT NOT NULL,  
    GameID INT NOT NULL,  
    DataRetirada DATE,  
    DataDevolucao DATE,  
    Status VARCHAR(50),  
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),  
    FOREIGN KEY (GameID) REFERENCES Games(GameID)  
);
```

---

## 6. INSERÇÃO E GESTÃO DE DADOS

### 6.1 Inserção de Dados de Teste

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

### Código 4: Exemplo de Inserção de Dados

sql

CopiarEditar

```
INSERT INTO Games (Titulo, Genero, Desenvolvedor, AnoLancamento)  
VALUES ('The Legend of Zelda', 'Aventura', 'Nintendo', 1986);
```

```
INSERT INTO Customers (Nome, Endereco, Email, Telefone)  
VALUES ('Maria Oliveira', 'Rua B, 200', 'maria@example.com', '21999999999');
```

```
INSERT INTO Games (Titulo, Genero, Desenvolvedor, AnoLancamento)
VALUES ('The Legend of Zelda', 'Aventura', 'Nintendo', 1986);

INSERT INTO Customers (Nome, Endereco, Email, Telefone)
VALUES ('Maria Oliveira', 'Rua B, 200', 'maria@example.com', '2199999999');
```

---

## 7. EXECUÇÃO E VALIDAÇÃO DE CONSULTAS

### 7.1 Consultas T-SQL

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

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

```
sql
CopiarEditar
SELECT Rentals.RentalID, Customers.Nome AS Cliente, Games.Titulo AS Jogo,
Rentals.Status
FROM Rentals
JOIN Customers ON Rentals.CustomerID = Customers.CustomerID
JOIN Games ON Rentals.GameID = Games.GameID;
```

```
SELECT Rentals.RentalID, Customers.Nome AS Cliente, Games.Titulo AS Jogo, Rentals.Status
FROM Rentals
JOIN Customers ON Rentals.CustomerID = Customers.CustomerID
JOIN Games ON Rentals.GameID = Games.GameID;
```

#### Código 6: Atualização do Status da Locação

```
sql

UPDATE Rentals
SET Status = 'Devolvido'
WHERE RentalID = 1;
```

```
UPDATE Rentals
SET Status = 'Devolvido'
WHERE RentalID = 1;
```

#### Código 7: Exclusão de uma Locação

```
sql
CopiarEditar
DELETE FROM Rentals
WHERE RentalID = 2;
```

```
DELETE FROM Rentals
WHERE RentalID = 2;
```

---

## 8. 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.

---

## 9. 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.

### 9.1 Próximos Passos

- Explorar a integração do banco de dados com aplicações web ou móveis.
- Realizar testes de desempenho e otimização para cenários de alta demanda.
- Ampliar o banco de dados para incluir funcionalidades adicionais, como relatórios de locações, histórico dos clientes e análises de jogos mais alugados.