



## **Desenvolvimento Full Stack**

**Felipe Marques de Almeida - 202208291929**

**Polo Rodovia Br 407 - Juazeiro - BA**

**RPG0015 - Vamos manter as informações!– Turma 22.3 – 3º Semestre**

**GitHub: [Kiffom2108/Missao-Pratica-Nivel-2-Mundo-3](https://github.com/Kiffom2108/Missao-Pratica-Nivel-2-Mundo-3) (github.com)**

### **Objetivo da Prática**

Identificar os requisitos de um sistema e transformá-los no modelo adequado.

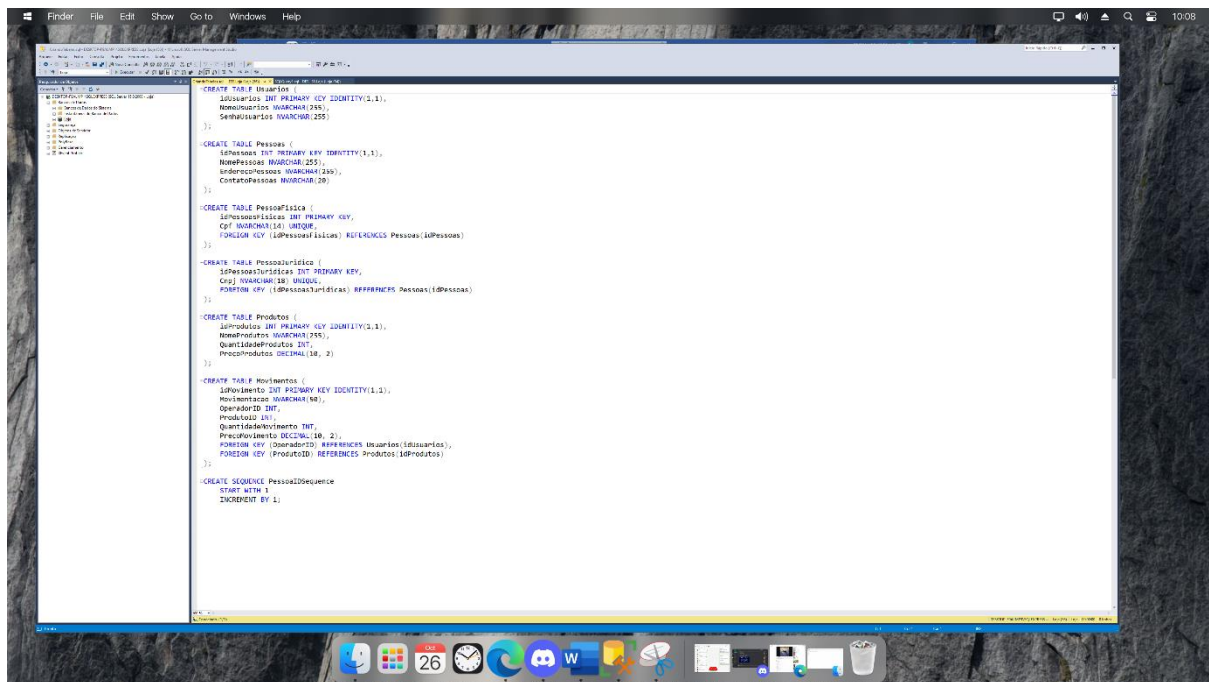
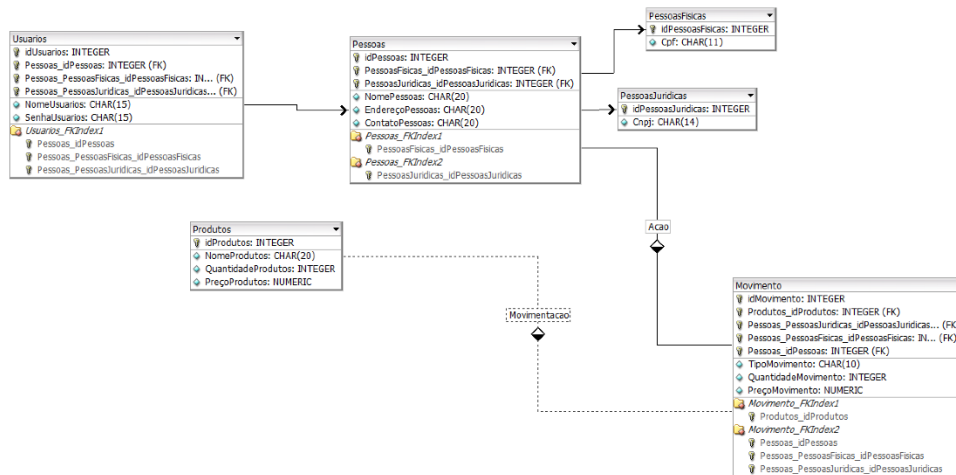
Utilizar ferramentas de modelagem para bases de dados relacionais.

Explorar a sintaxe SQL na criação das estruturas do banco (DDL).

Explorar a sintaxe SQL na consulta e manipulação de dados (DML)

No final do exercício, o aluno terá vivenciado a experiência de modelar a base de dados para um sistema simples, além de implementá-la, através da sintaxe SQL, na plataforma do SQL Server

## Códigos Desenvolvidos:



## Como são implementadas as diferentes cardinalidades, basicamente 1X1, 1XN ou NxN, em um banco de dados relacional?

Cardinalidade 1X1 é representada quando uma tabela se relaciona com outra de forma exclusiva, utilizando uma chave estrangeira única.

Cardinalidade 1XN é implementada quando uma tabela se relaciona com várias linhas de outra tabela, usando uma chave estrangeira na tabela com a cardinalidade "1".

Cardinalidade NxN é representada por meio de tabelas de junção que conectam duas ou mais tabelas.

### **Que tipo de relacionamento deve ser utilizado para representar o uso de herança em bancos de dados relacionais?**

Para representar herança em bancos de dados relacionais, pode-se usar a técnica de herança por tabela, onde uma tabela filha herda campos e relacionamentos de uma tabela pai. Outra abordagem é a herança por classe, que envolve a criação de uma tabela para cada classe na hierarquia de herança, e essas tabelas compartilham uma chave primária comum.

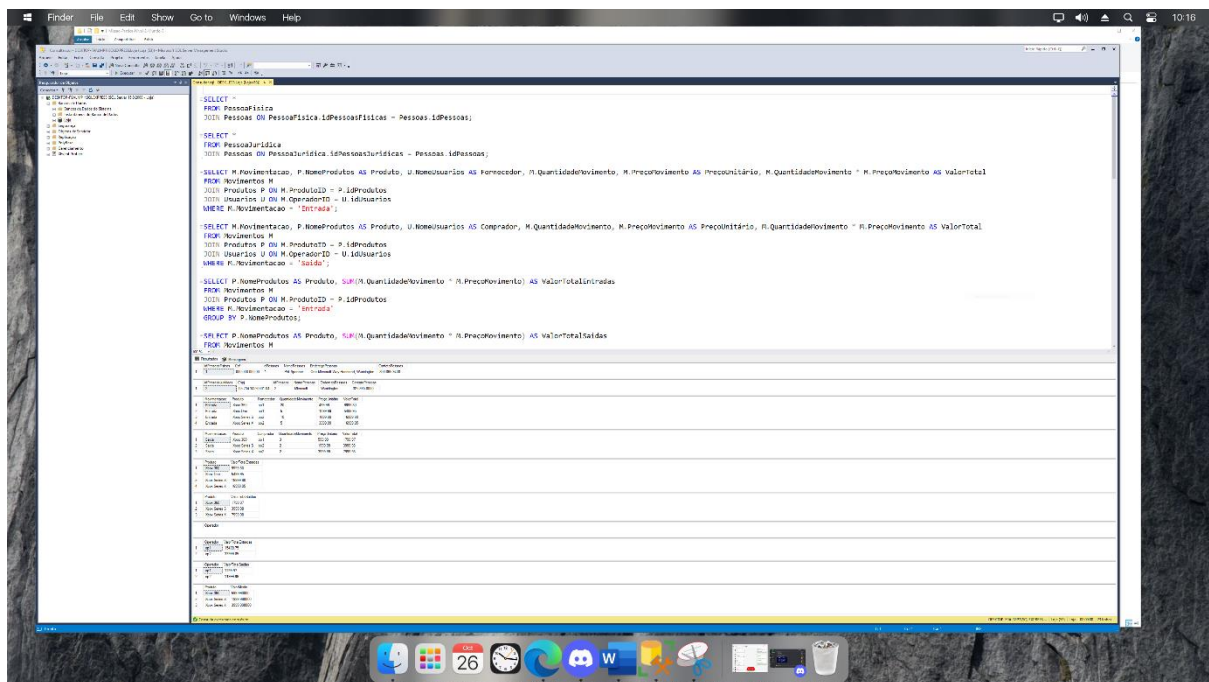
### **Como o SQL Server Management Studio permite a melhoria da produtividade nas**

### **tarefas relacionadas ao gerenciamento do banco de dados?**

O SQL Server Management Studio (SSMS) melhora a produtividade no gerenciamento de bancos de dados através de suas diversas funcionalidades. Ele oferece uma interface gráfica amigável para gerenciar bancos de dados, permite a escrita e execução de consultas SQL de forma eficiente, fornece ferramentas de monitoramento de desempenho, e facilita a administração de permissões e backups, tornando o gerenciamento do banco de dados mais eficaz e conveniente para os administradores e desenvolvedores.

### **Alimentando a Base**





## Diferenças entre o uso de Sequence e Identity:

O uso de Sequence e Identity está relacionado à geração de valores automáticos para colunas em um banco de dados. Aqui estão as principais diferenças entre eles:

### Sequence:

Uma sequência (Sequence) é um objeto do banco de dados que gera valores em sequência.

Pode ser utilizado para fornecer valores exclusivos para colunas de várias tabelas.

Oferece maior flexibilidade, permitindo que você defina os valores iniciais, incrementos e limites.

Pode ser compartilhado entre várias tabelas.

### Identity:

O Identity é uma propriedade de coluna que gera automaticamente valores exclusivos.

Normalmente associado a uma coluna de chave primária para garantir a unicidade.

A configuração de inicialização, incremento e limites é gerenciada automaticamente pelo sistema de gerenciamento de banco de dados.

Limitado a uma coluna por tabela.

## **Importância das Chaves Estrangeiras para a Consistência do Banco:**

As chaves estrangeiras (Foreign Keys) são fundamentais para a consistência do banco de dados, pois estabelecem relacionamentos entre tabelas. A importância delas inclui:

**Integridade Referencial:** As chaves estrangeiras garantem que os dados em uma tabela estejam relacionados a dados em outra tabela. Isso mantém a integridade referencial, evitando referências a dados inexistentes.

**Consistência dos Dados:** Ao usar chaves estrangeiras, você garante que os relacionamentos entre tabelas sejam mantidos, prevenindo informações inconsistentes ou órfãs.

**Manutenção da Qualidade dos Dados:** As chaves estrangeiras ajudam a manter a qualidade dos dados, uma vez que restringem a inserção de valores inválidos em colunas de referência.

## **Operadores do SQL na Álgebra e Cálculo Relacional:**

Álgebra Relacional (Operadores):

Seleção ( $\sigma$ ): Filtra linhas com base em condições.

Projeção ( $\pi$ ): Seleciona colunas específicas de uma tabela.

União ( $\cup$ ): Combina duas tabelas e remove duplicatas.

Interseção ( $\cap$ ): Retorna as linhas comuns a duas tabelas.

Diferença ( $-$ ): Retorna as linhas que estão em uma tabela, mas não na outra.

Produto Cartesiano ( $\times$ ): Combina todas as linhas de duas tabelas.

Junção ( $\bowtie$ ): Combina dados de várias tabelas com base em uma condição.

Divisão ( $\div$ ): Retorna registros que têm associações com todos os valores em outra tabela.

Cálculo Relacional:

O cálculo relacional é mais uma notação matemática para recuperar dados e não possui operadores com nomes como os da álgebra relacional.

Utiliza expressões matemáticas e predicados lógicos para especificar consultas.

## **Agrupamento em Consultas e Requisito Obrigatório:**

O agrupamento em consultas é realizado usando o operador GROUP BY. Para agrupar, você deve especificar uma ou mais colunas pelas quais deseja agrupar os resultados. O requisito obrigatório ao usar GROUP BY é que todas as colunas na lista de seleção devem ser uma coluna de grupo ou uma função de agregação. Além disso, colunas não agregadas não podem ser incluídas na lista de seleção, a menos que façam parte da cláusula GROUP BY.

O agrupamento permite que você resuma dados com base em valores comuns nas colunas de grupo e aplique funções de agregação, como SUM, COUNT, AVG, etc., para calcular valores resumidos para cada grupo.

Em resumo, o uso adequado de GROUP BY é fundamental para realizar agregações de dados e resumir informações em consultas SQL de forma eficaz.