

# Store Stocker

Cristoffer Martins

Arthur Ghizi

Gabriel Jorge

Gabriel Custódio

João Vitor



# Store Stocker

Simplificando o gerenciamento  
do seu estoque.







## LINGUAGEM

PHP



## FRAMEWORK

LARAVEL

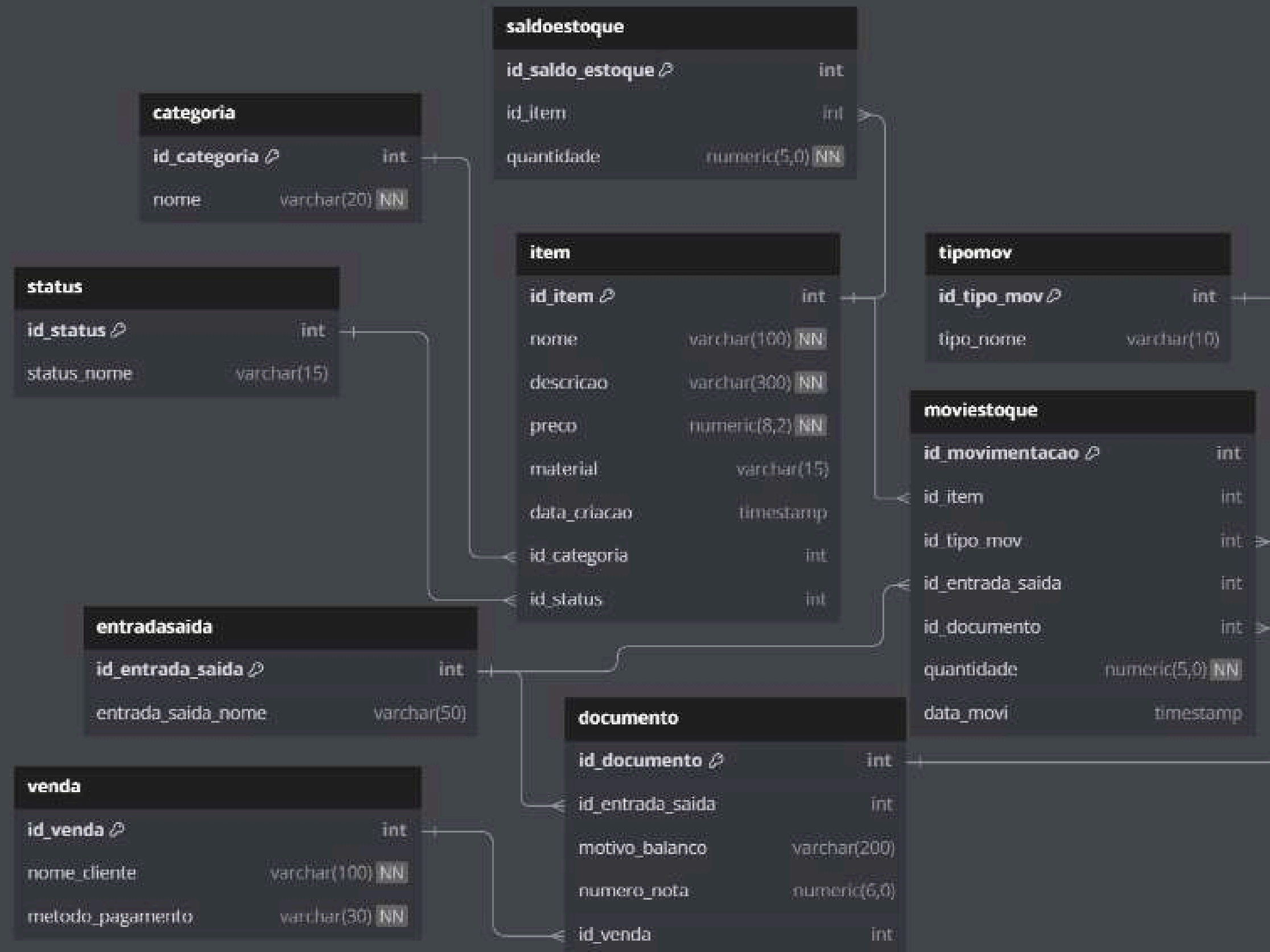


## BANCO

PostgreSQL  
(host no Supabase)



# MODELO



# FÍSICO



# SCRIPT FUNC

```
CREATE OR REPLACE FUNCTION consultar_estoque_categoria(id_categoria_param integer)
  RETURNS TABLE(item character varying, saldo numeric)
  LANGUAGE plpgsql
AS $function$
BEGIN
  -- Retorna os itens da categoria e o saldo disponível
  RETURN QUERY
  SELECT i.nome, s.quantidade
  FROM item i
  JOIN saldoestoque s ON i.id_item = s.id_item
  WHERE i.id_categoria = id_categoria_param;
END;
$function$
;

SELECT * FROM estoque_teste.consultar_estoque_categoria(1);
```



# SCRIPT PROC

```
CREATE OR REPLACE PROCEDURE estoque_teste.registrar_movimentacao(IN p_id_item integer, IN p_id_tipo_mov integer, IN p_quantidade integer, IN p_tipo_documento character varyi
    LANGUAGE plpgsql
AS $procedure$
DECLARE
    v_id_documento INT;
    v_id_venda INT;
BEGIN
    -- Verifica se a quantidade é válida
    IF p_quantidade <= 0 THEN
        RAISE EXCEPTION 'A quantidade deve ser maior que zero.';
    END IF;

    -- Caso 'nota_fiscal'
    IF p_tipo_documento = 'nota_fiscal' THEN
        IF p_numero_nota IS NULL THEN
            RAISE EXCEPTION 'O número da nota fiscal deve ser informado para este tipo de documento.';
        END IF;

        INSERT INTO documento (id_entrada_saida, numero_nota)
        VALUES (3, p_numero_nota)
        RETURNING id_documento INTO v_id_documento;
```



# SCRIPT PROC

```
-- Caso 'balanco'
ELSIF p_tipo_documento = 'balanco' THEN
    IF p_motivo_balanco IS NULL THEN
        RAISE EXCEPTION 'O motivo do balanço deve ser informado para este tipo de documento.';
    END IF;

    INSERT INTO documento (id_entrada_saida, motivo_balanco)
    VALUES (1, p_motivo_balanco)
    RETURNING id_documento INTO v_id_documento;

-- Caso 'venda'
ELSIF p_tipo_documento = 'venda' THEN
    IF p_nome_cliente IS NULL OR p_metodo_pagamento IS NULL THEN
        RAISE EXCEPTION 'Nome do cliente e método de pagamento devem ser informados para vendas.';
    END IF;

    -- Primeiro insere os dados na tabela 'venda'
    INSERT INTO venda (nome_cliente, metodo_pagamento)
    VALUES (p_nome_cliente, p_metodo_pagamento)
    RETURNING id_venda INTO v_id_venda;

    -- Em seguida, insere os dados na tabela 'documento'
    INSERT INTO documento (id_entrada_saida, id_venda)
    VALUES (2, v_id_venda)
    RETURNING id_documento INTO v_id_documento;
```



# SCRIPT PROC

```
ELSE
    RAISE EXCEPTION 'Tipo de documento inválido: %', p_tipo_documento;
END IF;

-- Insere os dados na tabela 'moviestoque'
INSERT INTO moviestoque (id_item, id_tipo_mov, id_entrada_saida, id_documento, quantidade)
VALUES (
    p_id_item,
    p_id_tipo_mov,
    CASE p_tipo_documento
        WHEN 'nota_fiscal' THEN 3
        WHEN 'balanco' THEN 1
        WHEN 'venda' THEN 2
    END,
    v_id_documento,
    p_quantidade
);

RAISE NOTICE 'Movimentação registrada com sucesso. Documento ID: %, Item ID: %', v_id_documento, p_id_item;

END;
$procedure$
;
```



# CHAMAR PROC

```
-- Nota fiscal
CALL estoque_teste.registrar_movimentacao(
    p_id_item := 1,
    p_id_tipo_mov := 1,
    p_quantidade := 5,
    p_tipo_documento := 'nota_fiscal',
    p_numero_nota := 1001
);

-- Balanço
CALL estoque_teste.registrar_movimentacao(
    p_id_item := 1,
    p_id_tipo_mov := 1,
    p_quantidade := 50,
    p_tipo_documento := 'balanco',
    p_motivo_balanco := 'Ajuste de estoque'
);

-- Venda
CALL estoque_teste.registrar_movimentacao(
    p_id_item := 1,
    p_id_tipo_mov := 2,
    p_quantidade := 3,
    p_tipo_documento := 'venda',
    p_nome_cliente := 'João da Silva',
    p_metodo_pagamento := 'Cartão de Crédito'
);
```



# SCRIPT TRIGGER

```
-- Trigger: atualizar saldo de estoque automatico
CREATE OR REPLACE FUNCTION update_saldo_estoque_fn()
RETURNS TRIGGER AS $$
BEGIN
    -- Verifica se há estoque insuficiente para saída
    IF EXISTS (
        SELECT 1
        FROM saldoestoque s
        WHERE s.id_item = NEW.id_item
        AND NEW.id_tipo_mov = 2
        AND NEW.quantidade > s.quantidade
    ) THEN
        RAISE EXCEPTION 'Estoque insuficiente para saída.';
    END IF;

    -- Atualiza saldo se o item já existe
    IF EXISTS (SELECT 1 FROM saldoestoque s WHERE s.id_item = NEW.id_item) THEN
        IF NEW.id_tipo_mov = 1 THEN
            UPDATE saldoestoque
            SET quantidade = quantidade + NEW.quantidade
            WHERE id_item = NEW.id_item;
        END IF;
    -- Insere novo saldo se o item não existe
    ELSE
        IF NEW.id_tipo_mov = 1 THEN
            INSERT INTO saldoestoque (id_item, quantidade)
            VALUES (NEW.id_item, NEW.quantidade);
        END IF;
    END IF;
END IF;
```



# SCRIPT TRIGGER

```
-- Processa saída de estoque
IF NEW.id_tipo_mov = 2 THEN
    UPDATE saldoestoque
    SET quantidade = quantidade - NEW.quantidade
    WHERE id_item = NEW.id_item;
END IF;

RETURN NEW;
END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER trigger_update_saldo_estoque
AFTER INSERT OR UPDATE ON moviestoque
FOR EACH ROW
EXECUTE FUNCTION update_saldo_estoque_fn();
```



# CRUD

+ Adicionar Joia

Nome	Material	Descrição	Status	Preço	
Anel de Ouro 18k	Ouro	Anel delicado de ouro 18k com detalhes minimalistas.	ativo	R\$ 1.500,00	<div> Editar</div> <div> Excluir</div>
Pulseira de Prata	Prata	Pulseira de prata com fecho ajustável.	ativo	R\$ 250,00	<div> Editar</div> <div> Excluir</div>
Colar com Pingente de Pérola	Dourado	Colar dourado com um pingente de pérola natural.	ativo	R\$ 900,00	<div> Editar</div> <div> Excluir</div>
Brinco de Diamante	Ouro Branco	Brinco de ouro branco com diamantes lapidados.	ativo	R\$ 2.500,00	<div> Editar</div> <div> Excluir</div>
Relógio de Aço Inoxidável	Aço	Relógio resistente com design moderno.	ativo	R\$ 300,00	<div> Editar</div> <div> Excluir</div>



# CRUD

## Adicionar Joia

Tipo de Material:

Ex: Ouro, Prata

Descrição:

Descrição do item

Nome da Joia:

Ex: Anel de Ouro

Preço (R\$):

Ex: 1500.00

Categoria:

Selecione a categoria

Status:

Selecione o status

Salvar

Voltar



# CRUD

## Editar Joia

Edite os detalhes da joia

Nome da Joia

Relógio de Aço Inoxidável

Preço

300,00

Status



Ativo



Desativado

Descrição

Relógio resistente com design moderno.

Salvar Alterações

[Voltar](#)



# 1. Qual o item mais movimentado nos últimos 30 dias (independente do tipo de movimentação)?

```
WITH MovimentacoesRecentes AS (  
    SELECT  
        ms.id_item,  
        tm.tipo_nome, -- Tipo de movimentação (entrada ou saída)  
        SUM(ms.quantidade) AS total_movimentado  
    FROM  
        moviestoque ms  
    JOIN  
        tipomov tm ON ms.id_tipo_mov = tm.id_tipo_mov  
    WHERE  
        ms.data_movi >= CURRENT_DATE - INTERVAL '30 days'  
    GROUP BY  
        ms.id_item, tm.tipo_nome  
)  
SELECT  
    i.nome AS item_nome,  
    m.tipo_nome AS tipo_movimentacao,  
    m.total_movimentado  
FROM  
    MovimentacoesRecentes m  
JOIN  
    item i ON m.id_item = i.id_item  
ORDER BY  
    m.total_movimentado DESC  
LIMIT 1;
```



## 2. Qual é o histórico de movimentações de cada item do estoque?

```
SELECT
    i.nome AS item_nome,
    tm.tipo_nome AS tipo_movimento,
    ms.quantidade,
    ms.data_movi
FROM
    moviestoque ms
JOIN
    item i ON ms.id_item = i.id_item
JOIN
    tipomov tm ON ms.id_tipo_mov = tm.id_tipo_mov
WHERE
    ms.id_item IN (SELECT id_item FROM item)
ORDER BY
    ms.data_movi;
```



### 3. Qual o item mais movimentado, do tipo SAÍDA, nos últimos 30 dias?

```
WITH MovimentacoesRecentes AS (  
    SELECT  
        ms.id_item,  
        SUM(ms.quantidade) AS total_saida  
    FROM  
        moviestoque ms  
    WHERE  
        ms.data_movi >= CURRENT_DATE - INTERVAL '30 days'  
        AND ms.id_tipo_mov = 2  
    GROUP BY  
        ms.id_item  
),  
RankedMovimentacoes AS (  
    SELECT  
        m.id_item,  
        m.total_saida,  
        ROW_NUMBER() OVER (ORDER BY m.total_saida DESC) AS rank  
    FROM  
        MovimentacoesRecentes m  
)  
SELECT  
    i.nome AS item_nome,  
    r.total_saida  
FROM  
    RankedMovimentacoes r  
JOIN  
    item i ON r.id_item = i.id_item  
WHERE  
    r.rank = 1;
```



## 4. Qual é a média de movimentação de entrada e saída por item no último ano?

```
WITH MediaMovimentacao AS (  
    SELECT  
        ms.id_item,  
        AVG(CASE WHEN ms.id_tipo_mov = 1 THEN ms.quantidade ELSE 0 END) AS media_entrada,  
        AVG(CASE WHEN ms.id_tipo_mov = 2 THEN ms.quantidade ELSE 0 END) AS media_saida  
    FROM  
        moviestoque ms  
    WHERE  
        ms.data_movi >= CURRENT_DATE - INTERVAL '1 year'  
    GROUP BY  
        ms.id_item  
)  
SELECT  
    i.nome AS item_nome,  
    m.media_entrada,  
    m.media_saida  
FROM  
    MediaMovimentacao m  
JOIN  
    item i ON m.id_item = i.id_item  
ORDER BY  
    (m.media_entrada + m.media_saida) DESC  
LIMIT 5;
```



# 5. Quais os 5 itens em menor quantidade dentro do estoque?

```
WITH SaldoEstoque AS (  
    SELECT  
        ms.id_item,  
        i.nome AS item_nome,  
        SUM(CASE WHEN ms.id_tipo_mov = 1 THEN ms.quantidade ELSE -ms.quantidade END) AS saldo  
    FROM  
        moviestoque ms  
    JOIN  
        item i ON ms.id_item = i.id_item  
    GROUP BY  
        ms.id_item, i.nome  
)  
,  
RankingEstoque AS (  
    SELECT  
        item_nome,  
        saldo,  
        DENSE_RANK() OVER (ORDER BY saldo ASC) AS ranking  
    FROM  
        SaldoEstoque  
)  
|  
SELECT  
    item_nome,  
    saldo,  
    ranking  
FROM  
    RankingEstoque  
WHERE  
    ranking <= 5  
ORDER BY  
    saldo ASC;
```



## 6. Quais os 5 itens com maior variação de saldo (entrada/saída) nos últimos 30 dias?

```
WITH MovimentacoesVariação AS (  
  SELECT  
    ms.id_item,  
    SUM(CASE WHEN ms.id_tipo_mov = 1 THEN ms.quantidade ELSE 0 END) AS total_entradas,  
    SUM(CASE WHEN ms.id_tipo_mov = 2 THEN ms.quantidade ELSE 0 END) AS total_saidas  
  FROM  
    moviestoque ms  
  WHERE  
    ms.data_movi >= CURRENT_DATE - INTERVAL '30 days'  
  GROUP BY  
    ms.id_item  
)  
,  
RankingMovimentacoes AS (  
  SELECT  
    i.nome AS item_nome,  
    (m.total_entradas - m.total_saidas) AS variacao_saldo,  
    RANK() OVER (ORDER BY (m.total_entradas - m.total_saidas) DESC) AS ranking  
  FROM  
    MovimentacoesVariação m  
  JOIN  
    item i ON m.id_item = i.id_item  
)  
SELECT  
  item_nome,  
  variacao_saldo,  
  ranking  
FROM  
  RankingMovimentacoes  
WHERE  
  ranking <= 5  
ORDER BY  
  ranking;
```



# OBRIGADO!

Cristoffer Martins

Arthur Ghizi

Gabriel Jorge

Gabriel Custódio

João Vitor