



ESCOLA  
SUPERIOR  
DE TECNOLOGIA  
E GESTÃO

## **Disciplina de Bases de Dados**

Ano Letivo de 2024 / 2025

# **Desenvolvimento e Implementação de uma Base de Dados para Gestão de uma Fábrica de Calçado**

**Pedro Antunes, 8230068**

**Ruben Nunes, 8230069**

**João Ribeiro, 8130157**

Maio, 2025



# Agradecimentos

Gostaria de expressar o meu sincero agradecimento a todas as entidades e pessoas que, de diferentes formas, contribuíram para a realização deste projeto.

À empresa **Guimarshoe**, pela disponibilidade em nos receber nas suas instalações, permitindo-nos observar o funcionamento real de uma fábrica de calçado, bem como pela partilha das fichas técnicas e pela abertura do próprio responsável da empresa em explicar o processo produtivo com detalhe e clareza.

A dois familiares do Ruben (Paula, Casemiro) que forneceram conhecimento do métodos de processos na **Ecco®**, pelo contributo valioso ao proporcionar uma visão global sobre o funcionamento de uma grande empresa do setor do calçado, ajudando-nos a compreender a realidade industrial numa escala diferente.

Ao professor **Vasco Santos** pelo acompanhamento prestado e pela disponibilidade em analisar o nosso diagrama Entidade-Relacionamento, orientando-nos na sua melhoria.

Aos colegas do **CTeSP – Curso Técnico Superior Profissional em Cibersegurança, Redes e Sistemas Informáticos**, bem como aos colegas da licenciatura em **Segurança Informática em Redes de Computadores (LSIRC)**, pelo seu contributo com ideias, conselhos técnicos e partilhas que enriqueceram significativamente o nosso trabalho.

A todos, o nosso muito obrigado.

# Resumo

O presente relatório descreve o trabalho desenvolvido no âmbito da disciplina de Base de Dados, com o objetivo principal de criar uma base de dados para suportar a gestão das Ordens de Produção de uma fábrica. A primeira fase do projeto envolveu a pesquisa e análise de requisitos, incluindo a interação com o contexto de uma fábrica para compreender o fluxo das ordens de fabrico. Seguiu-se o planeamento e discussão detalhada dos requisitos para garantir a sua concretização.

O desenvolvimento contemplou inicialmente a criação de um modelo conceptual, apresentado para feedback. Com base neste retorno, o modelo foi refinado, focando a base de dados na gestão detalhada das ordens de fabrico e das operações associadas, resultando numa estrutura mais clara e direta. O resultado desta fase é um modelo de dados relacional, representado através de um Diagrama de Entidade-Relacionamento (ER), que estabelece a estrutura para o armazenamento e processamento da informação.

Durante a fase de implementação, a base de dados foi criada recorrendo a comandos SQL para a definição das tabelas, relações, *constraints* e *triggers*, assegurando a integridade e automatização de processos como o cálculo de perdas, custos, atualização de estados e registo do consumo de materiais. Foram ainda desenvolvidas diversas *views* e consultas analíticas, permitindo gerar relatórios de apoio à decisão, como o progresso das ordens, eficiência das operações e identificação de perdas anormais.

Foram realizados testes funcionais e de integridade para garantir o correto funcionamento da base de dados, incluindo simulações completas do ciclo de vida de uma ordem de fabrico, desde a sua emissão até à conclusão. Adicionalmente, foi recolhido feedback de utilizadores externos, que permitiu validar a clareza dos dados apresentados nos relatórios e a fiabilidade dos automatismos implementados.

Em conclusão, o projeto cumpriu os objetivos propostos com sucesso, fornecendo uma base de dados robusta e funcional, capaz de acompanhar com precisão o processo produtivo de uma fábrica. A solução construída não só suporta a gestão diária das operações, como também abre portas a futuras melhorias, como a integração com sistemas externos, a inclusão de indicadores de desempenho em tempo real, ou a construção de uma interface gráfica para utilizadores finais.

# Índice

Agradecimentos .....	3
Resumo .....	4
Índice .....	5
Índice de Figuras.....	8
Índice de Tabelas .....	9
<b>1. Introdução .....</b>	<b>10</b>
1.1 Contextualização .....	10
1.2 Apresentação do Caso de Estudo .....	10
1.3 Motivação e Objetivos .....	11
1.4 Estrutura do Relatório.....	12
<b>2. Metodologia.....</b>	<b>13</b>
2.1 Planeamento e Recolha de Informação .....	13
2.2 Modelação Conceptual .....	13
2.3 Modelação Lógica .....	14
2.4 Implementação Técnica .....	15
2.5 Testes e Validações .....	15
<b>3. Modelo de Dados .....</b>	<b>17</b>
3.1 Modelo Conceptual .....	17
3.2.2 Entidades Principais .....	18
3.2 Domínio dos Atributos .....	20
3.3 Modelo Lógico .....	26
3.3.1 Objetivos .....	26
3.3.2 Componentes do Modelo Lógico .....	26
3.3.3 Diagrama do Modelo Lógico .....	27
3.3 Modelo Físico .....	28
<b>4. Implementação .....</b>	<b>29</b>
4.1 Povoamento da Base de Dados.....	29
4.2.1 Objetivos do Povoamento da Base de Dados .....	29
4.2.2 Etapas do Povoamento .....	29
4.2.3 Conceito do Povoamento na Prática .....	30
<b>5. Regras de Negócio Implementadas .....</b>	<b>31</b>
<b>6. Consultas SQL .....</b>	<b>34</b>

6.1	Consultas .....	34
6.1.1	As ordens de fabrico emitidas e o seu estado atual .....	34
6.1.2	Quantidades de materiais enviados para produção.....	35
6.1.3	Quantidade de produtos semi-acabados ou acabados.....	36
6.1.4	Percentagem de perdas (materiais defeituosos ou desperdiçados).....	37
6.1.5	Os subcontratados envolvidos e os custos associados .....	38
6.1.6	Consultar as ordens de fabrico em curso, com indicação do progresso .....	39
6.1.7	Identificar autom. Ordens com atrasos .....	41
6.1.8	Calcular a percentagem média de perdas por operação .....	41
6.1.9	Gerar relatórios de operações com perdas superiores a 15% .....	43
6.1.10	Listar operações mais freq. subcontratadas e os respetivos custos .....	43
6.1.11	Determinar o custo tot. de prod. de uma ordem .....	44
6.1.12	Consultar a lista de operações pendentes para cada ordem.....	46
6.1.13	Estimar o tempo tot. para concluir uma ordem (tempo médio das operações) ....	47
6.1.14	Calc. a eficiência média de produção (materiais aproveitados, por período) .....	48
6.1.15	Geração de alertas, armazenando-o numa tabela específica de controlo .....	49
6.1.16	Determinar a eficiência média por operação e por período .....	49
6.1.16	Listar ordens atrasadas com o respetivo tempo de atraso .....	50
6.1.17	Operações problemáticas (com maior índice de perdas) .....	51
6.1.18	Custos comparativos entre produção interna e subcontratada .....	52
6.1.19	Geração de alertas de qualidade, se a perda for sup. a 15%.....	53
6.2	Procedures .....	54
6.2.1	Criar reg. de execução para as operações e ordens da ficha técnica .....	54
<b>7.</b>	<b>Conclusão .....</b>	<b>56</b>
<b>8.</b>	<b>Anexos .....</b>	<b>57</b>
8.1	CREATE TABLE.....	57
8.1.1	Produto .....	57
8.1.2	Operacao .....	57
8.1.3	Material .....	58
8.1.4	Subcontratado.....	58
8.1.5	FichaTecnica .....	59
8.1.6	DetalheFichaTecnica .....	60
8.1.7	OrdemFabrico .....	60
8.1.8	ItemOrdemFabrico.....	61
8.1.9	ExecucaoOperacao.....	62
8.1.10	AlertaAtraso.....	62

8.1.11	AlertaQualidade .....	63
8.1.12	ConsumoMaterialReal.....	63
8.2	CREATE TRIGGERS.....	64
8.2.1	Produto .....	64
8.2.2	Subcontratado.....	65
8.2.3	FichaTecnica .....	65
8.2.4	DetalheFichaTecnica .....	67
8.2.5	ItemOrdemFabricao.....	68
8.2.6	OrdemFabricao .....	69
8.2.7	AlertaAtraso.....	71
8.2.8	AlertaQualidade.....	72
8.2.9	ExecucaoOperacao.....	73
8.2.10	ConsumoMaterialReal.....	78
8.3	CREATE PROCEDURE .....	79
8.5	CREATE VIEW .....	81
8.4	INSERT INTO .....	87
8.4.1	Produto .....	87
8.4.2	Operacao .....	88
8.4.3	Material .....	88
8.4.4	Subcontratado.....	89
8.4.5	FichaTecnica .....	89
8.4.6	DetalheFichaTecnica .....	90
8.4.7	OrdemFabricao .....	93
8.4.8	ItemOrdemFabricao.....	93
8.4.9	MaterialConsumoReal .....	94
<b>9.</b>	<b>Referências .....</b>	<b>97</b>

# Índice de Figuras

Figura 1 - Diagrama ER (Modelo Conceptual)	17
Figura 2 - Diagrama ER (Modelo Lógico)	27
Figura 3 - Estrutura do Modelo Físico	28
Figura 4 - Resultado da consulta [6.1.1]	34
Figura 5 - Resultado da consulta [6.1.2]	35
Figura 6 - Resultado da consulta [6.1.3]	36
Figura 7 - Resultado da consulta [6.1.4]	37
Figura 8 - Resultado da consulta [6.1.5]	38
Figura 9 - Resultado da consulta [6.1.6]	40
Figura 10 - Resultado da consulta [6.1.7]	41
Figura 11 - Resultado da consulta [6.1.8]	42
Figura 12 - Resultado da consulta [6.1.9]	43
Figura 13 - Resultado da consulta [6.1.10]	44
Figura 14 - Resultado da consulta [6.1.11]	45
Figura 15 - Resultado da consulta [6.1.12]	46
Figura 16 - Resultado da consulta [6.1.13]	47
Figura 17 - Resultado da consulta [6.1.14]	48
Figura 18 - Resultado da consulta [6.1.16]	50
Figura 19 - Resultado da consulta [6.1.16]	50
Figura 20 - Resultado da consulta [6.1.17]	51
Figura 21 - Resultado da consulta [6.1.18]	52
Figura 22 - Resultado da consulta [6.1.19]	53
Figura 23 - Resultado do procedimento [6.2.1]	54
Figura 24 - Resultado do trigger [6.2.1]	55



# Índice de Tabelas

Tabela 1 - Domínio da Tabela "Produto"	20
Tabela 2 - Domínio da Tabela "Operacao"	21
Tabela 3 - Domínio da Tabela "Material"	21
Tabela 4 - Domínio da Tabela "Subcontratado"	21
Tabela 5 - Domínio da Tabela "FichaTecnica"	22
Tabela 6 - Domínio da Tabela "DetalheFichaTecnica"	22
Tabela 7 - Domínio da Tabela "OrdemFabricao"	23
Tabela 8 - Domínio da Tabela "ItemOrdemFabricao"	23
Tabela 9 - Domínio da Tabela "ExecucaoOperacao"	24
Tabela 10 - Domínio da Tabela "AlertaAtraso"	24
Tabela 11 - Domínio da Tabela "AlertaQualidade"	25
Tabela 12 - Domínio da Tabela "ConsumoMaterialReal"	25

# 1. Introdução

O projeto envolveu levantamento de requisitos, modelação de dados, implementação em *SQL Server*, testes, consultas *SQL* e mecanismos automáticos como *procedures*, *triggers* e *views*. O sistema resultante permite gerir eficientemente as ordens de fabrico, subcontractações, controlo de perdas, custos e eficiência produtiva.

## 1.1 Contextualização

O presente trabalho foi desenvolvido no âmbito da disciplina de Base de Dados e foca-se na conceção de um sistema relacional que responda às necessidades específicas de uma fábrica de calçado no que diz respeito à organização das ordens de produção, operações associadas e processos de subcontractação. Para enquadrar o projeto, partiu-se de um cenário típico de uma fábrica de calçado, onde é comum a realização de diversas operações como o **corte, a costura, a montagem e o acabamento**.

Embora não tenhamos qualquer experiência direta neste setor, procurámos compreendê-lo melhor através do contacto com algumas fábricas locais, bem como da consulta de recursos disponíveis online. Esta fase inicial foi fundamental para orientar a modelação dos dados e garantir que a base de dados concebida responde a situações realistas no contexto do fabrico de calçado.

## 1.2 Apresentação do Caso de Estudo

Este trabalho surgiu da necessidade de aplicar os conhecimentos adquiridos ao longo do semestre na unidade curricular de **Base de Dados**. Tendo despertado interesse em compreender mais de perto o funcionamento de uma fábrica de calçado, considerámos ser relevante explorar como os processos produtivos poderiam beneficiar de uma maior **organização e automatização** através de uma base de dados estruturada.

Durante a análise inicial, verificámos que muitos dos processos relacionados com **ordens de fabrico, controlo de materiais e operações** são, em muitos casos, tratados de **forma manual** ou pouco sistematizada. Assim, tornou-se evidente o potencial valor de um sistema relacional que permita **armazenar, consultar e gerir** eficientemente toda a informação envolvida na produção, contribuindo para uma maior eficiência e produtividade nas fábricas que adotem este modelo

## 1.3 Motivação e Objetivos

Depois de termos contactado duas fábricas com realidades bastante distintas — uma que ainda utilizava registos em papel e outra que já recorria a métodos digitais — surgiu-nos o interesse em desenvolver um sistema funcional que substituísse o uso do papel por uma solução mais digital e evoluída. Acreditámos que uma maior digitalização dos processos poderia contribuir para uma gestão mais eficiente e, consequentemente, para um aumento da produtividade.

A ideia de desenvolver uma base de dados relacional para suportar a gestão de ordens de fabrico revelou-se uma oportunidade interessante para aplicar e aprofundar conhecimentos **como modelação de dados, integridade referencial, normalização e criação de consultas orientadas** para o apoio à decisão.

O principal objetivo deste projeto consistiu na conceção e implementação de uma base de dados que permitisse:

- Gerir ordens de fabrico e respetivo estado;
- Registar todas as operações realizadas em cada ordem, com indicação dos materiais envolvidos, quantidades enviadas e recebidas, e perdas ocorridas;
- Controlar subcontratações, identificando os parceiros externos envolvidos e os respetivos custos;
- Gerar alertas automáticos em caso de atrasos ou perdas excessivas;
- Calcular indicadores como o progresso da produção, custos totais, percentagem de perdas e eficiência por operação;
- Apoiar a gestão através de consultas que identifiquem situações críticas ou padrões relevantes na produção.

Estes objetivos orientaram todas as fases do trabalho, desde a análise inicial até à implementação prática da base de dados, procurando garantir que a solução desenvolvida respondesse de forma **funcional, fiável e organizada** às necessidades típicas de um processo produtivo realista.

## 1.4 Estrutura do Relatório

Este relatório está organizado em sete capítulos, de forma a refletir todas as etapas envolvidas no desenvolvimento do projeto:

**Capítulo 1 – Introdução:** apresenta o enquadramento geral do trabalho, incluindo o contexto do caso de estudo, a motivação e os objetivos definidos.

**Capítulo 2 – Metodologia:** descreve as fases seguidas no desenvolvimento do projeto, desde a recolha de informação até à modelação e implementação.

**Capítulo 3 – Modelo de Dados:** expõe o modelo Entidade-Relacionamento elaborado com base nos requisitos identificados, bem como as decisões tomadas durante a sua construção.

**Capítulo 4 – Implementação:** foca-se no processo de **criação e povoamento da base de dados**, descrevendo os objetivos do povoamento, as fontes de dados utilizadas (reais ou simuladas), e as etapas realizadas para garantir a coerência e a integridade dos dados introduzidos.

**Capítulo 5 – Regras de Negócio Implementadas:** documenta as regras de negócio aplicadas através de restrições, *triggers* e *procedures*, assegurando o cumprimento das regras funcionais do sistema.

**Capítulo 6 – Consultas SQL:** apresenta um conjunto selecionado de *views* e *queries* desenvolvidas para análise, monitorização e apoio à decisão. Inclui também descrições de algumas *triggers* e vistas associadas a cálculos de desempenho, alertas e relatórios operacionais.

**Capítulo 7 – Conclusão:** reflete sobre os principais resultados obtidos, os desafios enfrentados e as aprendizagens desenvolvidas ao longo do projeto

**Capítulo 8 – Anexos:** reúne os scripts SQL de criação da base de dados, *triggers*, *procedures* e consultas mais relevantes, bem como eventuais diagramas ou elementos de apoio adicionais.

**Capítulo 9 – Referências:** apresenta as fontes bibliográficas, técnicas e digitais que fundamentaram o trabalho realizado.

## 2. Metodologia

O desenvolvimento deste projeto seguiu uma abordagem faseada, dividida em várias etapas que permitiram organizar o trabalho de forma lógica e progressiva, desde a recolha inicial de informação com visitas a uma fábrica e consulta do enunciado até à implementação final da base de dados.

### 2.1 Planeamento e Recolha de Informação

Numa fase inicial, e por sugestão do professor, realizámos visitas a duas fábricas locais com o objetivo de compreender melhor o funcionamento de uma fábrica de calçado. Estas visitas, aliadas à consulta de algumas fontes online, permitiram-nos identificar as principais fases do processo de produção, os tipos de operações envolvidas, o modo como funcionam as operações subcontratadas e também aspetos relacionados com a gestão de stock.

Tivemos ainda a oportunidade de perceber, de forma mais concreta, como o negócio funciona no seu todo — desde a entrada de uma encomenda até à finalização e entrega do produto — o que foi essencial para fundamentar o modelo de dados a desenvolver.

### 2.2 Modelação Conceptual

Com base nas informações recolhidas durante as visitas às fábricas e na pesquisa realizada, começámos por identificar os principais elementos envolvidos no processo produtivo, o que nos levou a definir um conjunto inicial de entidades e relações. Esta primeira versão do modelo conceptual incluía várias tabelas, tentando representar todos os aspetos possíveis da produção, desde a fase em que o cliente fazia a encomenda com a fábrica até à sua distribuição.

No entanto, após a apresentação inicial ao professor, recebemos feedback que nos ajudou a perceber que o modelo estava excessivamente complexo, com várias entidades desnecessárias para os objetivos definidos. Foi então sugerido que nos focássemos exclusivamente na gestão das ordens de fabrico e operações associadas, deixando de lado componentes menos relevantes para esta fase do projeto, como por exemplo maquinaria usada, empregos envolvidos nos processos entre outras coisas.

Com base nessas orientações, redesenhámos o modelo conceptual, tornando-o mais direto, funcional e ajustado ao caso de estudo. As entidades finais incluíram apenas os elementos essenciais, como **Produto**, **Ficha Técnica**, **Operação**, **Ordem de Fabrico**, **Execução de Operação**, **Material** e **Subcontratado**, entre outras diretamente ligadas ao processo produtivo.

O modelo *Entidade-Relacionamento (ER)* foi elaborado com recurso a ferramentas digitais de modelação, permitindo representar graficamente as entidades com as suas respetivas *Primary Keys* e as relações entre elas.

## 2.3 Modelação Lógica

Após a definição do modelo conceptual, avançámos para a modelação lógica, com o objetivo de converter as entidades e relações identificadas em tabelas relacionais compatíveis com a linguagem SQL. Esta etapa teve como principal foco a criação de uma estrutura coerente e eficiente para posterior implementação no sistema de gestão de base de dados.

Cada entidade foi convertida numa tabela com os respetivos atributos, tendo sido definida uma chave primária (*PK*) para cada tabela, de forma a garantir a identificação única de cada registo. Para além disso, foram estabelecidas chaves estrangeiras (*FK*) que asseguram a ligação entre as várias tabelas, com base nas relações do modelo conceptual. Por exemplo, a tabela **FichaTecnica** contém uma *FK* que referencia o **Produto**, e a tabela **ExecucaoOperacao** liga-se simultaneamente ao **ItemOrdemFabrico**, **Operacao** e **Subcontratado**.

Na definição dos atributos, foram escolhidos tipos de dados adequados ao seu conteúdo, tendo em conta tanto o feedback obtido nas fábricas visitadas como as especificidades do sistema de base de dados utilizado (*SQL Server*). Por exemplo, campos numéricos como quantidades e custos foram definidos como *INT* ou *DECIMAL*, datas como *DATE*, e campos de texto como *VARCHAR* com tamanhos apropriados.

Durante este processo, foram ainda aplicados os princípios de normalização, nomeadamente até à terceira forma normal (*3FN*), com o objetivo de evitar redundâncias, garantir a integridade dos dados e facilitar a manutenção da base de dados.

Esta modelação lógica serviu de base direta para a criação do *script SQL* responsável pela implementação física da base de dados.

## 2.4 Implementação Técnica

Após a definição do modelo lógico, procedeu-se à implementação física da base de dados utilizando o sistema de gestão *SQL Server*. Para isso, foi criado um *script SQL* com todas as instruções necessárias para a criação das tabelas, definição de chaves primárias e estrangeiras, e aplicação de restrições relevantes, como tipos de dados e regras de integridade.

O *script* foi executado no *SQL Server Management Studio (SSMS)*, ambiente onde foi criada a base de dados com a estrutura previamente planeada. A criação das tabelas foi realizada de forma sequencial, respeitando a ordem de dependências entre as chaves estrangeiras, garantindo assim a integridade referencial desde o início.

Após a criação da estrutura, foram inseridos dados de teste com o objetivo de validar o correto funcionamento da base de dados. Estes dados permitiram simular casos típicos de produção, como a emissão de ordens de fabrico, a execução de operações (*internas e subcontratadas*), o registo de perdas e a geração de alertas automáticos.

Durante esta fase, foram também realizados ajustes pontuais ao modelo lógico, sempre que se verificou a necessidade de melhorar a estrutura ou corrigir pequenos detalhes. No final da implementação, a base de dados encontrava-se totalmente funcional e pronta para suportar as consultas e validações definidas nos objetivos do projeto

## 2.5 Testes e Validações

Após a implementação da base de dados, foram realizados diversos testes com o objetivo de validar a estrutura, a integridade dos dados e o cumprimento dos requisitos funcionais definidos. Para tal, foram inseridos dados fictícios nas principais tabelas, simulando situações reais como a criação de produtos, emissão de ordens de fabrico, execução de operações e envolvimento de subcontratados.

Um dos principais focos dos testes foi verificar o correto funcionamento das chaves estrangeiras, garantindo que apenas dados válidos pudessem ser introduzidos e que as relações entre as tabelas fossem corretamente mantidas. Para além disso, foram testadas restrições de integridade como tipos de dados, campos obrigatórios e valores válidos.

Também foram executadas consultas *SQL* com o objetivo de obter indicadores relevantes, como a percentagem de perdas, custos de produção e identificação de ordens com atraso. Estas consultas permitiram verificar se os dados estavam a ser processados corretamente e se os resultados correspondiam às expectativas definidas.

Por fim, os testes permitiram identificar pequenos ajustes a fazer na estrutura e confirmar que a base de dados estava preparada para responder aos casos de uso definidos. Esta fase foi essencial para garantir a fiabilidade e utilidade prática do sistema desenvolvido.



## 3. Modelo de Dados

### 3.1 Modelo Conceptual

Para que fosse possível a criação desta base de dados, seguimos os passos ensinados ao longo do semestre, complementados pelas definições do manual recomendado pelo professor.

O primeiro passo no desenvolvimento foi o **levantamento de requisitos**, cujo objetivo é compreender que informação o sistema deve armazenar e como os dados serão utilizados. Para isso, recolhemos o máximo de informação possível através de pesquisa online e, principalmente, através da visita à fábrica *GuimaShoe*, o que nos permitiu perceber melhor o funcionamento real do processo de fabrico de calçado.

Com base nessas informações, avançámos para a **modelação conceptual**, que visa criar uma visão abstrata da estrutura dos dados, ainda sem ligação a qualquer tecnologia ou sistema de gestão específico. Nesta fase, identificámos as entidades principais como **OrdemFabrico**, **Produto**, **Operacao**, **Material** e **Subcontratado**, entre outras.

A partir daí, começámos a esboçar os relacionamentos entre estas entidades. Por exemplo:

- Uma **OrdemFabrico** pode incluir vários produtos em diferentes quantidades, o que originou a entidade associativa **ItemOrdemFabrico**.
- Cada **Produto** está associado a uma ou mais **FichaTecnica**, que descrevem as etapas e materiais necessários à sua produção.
- Cada **FichaTecnica** inclui várias **Operacoes**, cada uma possivelmente relacionada com um ou mais **Materiais**.
- As **Operacoes** podem ser executadas **internamente**, por **Subcontratados** e também podemos ser subcontratados para fazer certas operações na fábrica, o que motivou a criação de relações que permitam distinguir esse cenário.

O modelo conceptual serviu como ponto de partida fundamental para entender a estrutura geral dos dados e facilitou a comunicação da arquitetura inicial do sistema. Este modelo foi representado graficamente através de um **Diagrama Entidade-Relacionamento (ER)** utilizando a notação **Crow's Foot**, onde se definiram as entidades, atributos e cardinalidades.

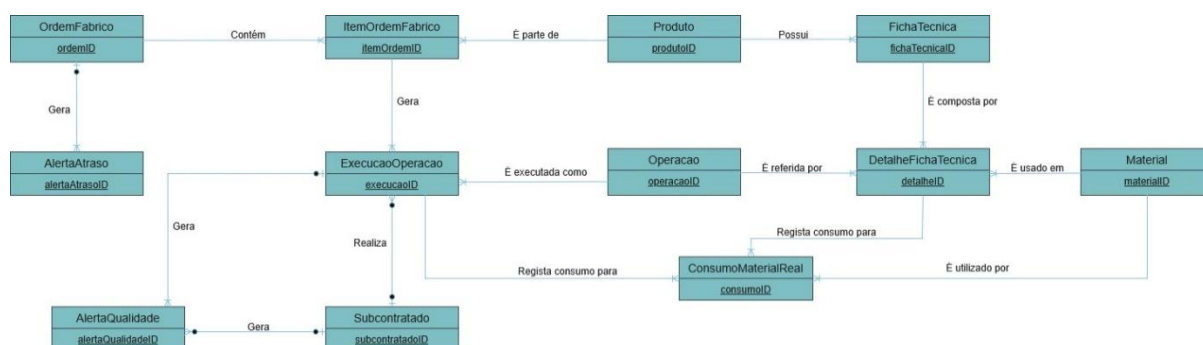


Figura 1 - Diagrama ER (Modelo Conceptual)

Nesta fase, foi evidente que a *OrdemFabricao* é a entidade central do sistema, representando um pedido de produção. Através da entidade *ItemOrdemFabricao*, relaciona-se com um ou mais Produtos, cada um com a sua respetiva *FichaTecnica*, que detalha as operações necessárias.

O relacionamento entre *Operacao* e *Material* foi modelado como muitos-para-muitos, indicando que uma operação pode envolver vários materiais e que um mesmo material pode ser utilizado em diferentes operações. Por fim, os Subcontratados foram tratados como entidades externas que podem ser responsáveis pela execução de certas operações, representando assim uma relação um-para-muitos entre Subcontratado e *ExecucaoOperacao*.

### 3.2.2 Entidades Principais

- **Produto:** Define os produtos fabricados, com detalhes como modelo, variante, cor e datas de registo.
  - Atributos: produtoID, modelo, variante, cor, nome, descricao, dataCriacao, dataAtualizacao
- **Operacao:** Representa cada tipo de operação realizada no processo produtivo.
  - Atributos: operacaoID, nome, descricao
- **Material:** Armazena a informação dos materiais utilizados nas operações de produção, incluindo tipo, unidade e custo.
  - Atributos: materialID, nome, tipo, unidadeMedida, custoUnitario
- **Subcontratado:** Identifica entidades externas responsáveis por executar determinadas operações produtivas.
  - Atributos: subcontradoID, nome, morada, contacto, custoServico
- **FichaTecnica:** Detalha os passos e recursos necessários à produção de um produto específico.
  - Atributos: fichaTecnicaID, *produtoID* (FK para **Produto**), descricao, dataCriacao, dataAtualizacao
- **DetalheFichaTecnica:** Especifica as operações e materiais de uma ficha técnica, com ordem, quantidades e custos.
  - Atributos: detalheID, *fichaTecnicaID* (FK para **FichaTecnica**), *operacaoID* (FK para **Operacao**), *materialID* (FK para **Material**), ordemExecucao, quantidade, duracaoUnitariaMin, custoMinuto

- **OrdemFabricao:** Representa uma ordem de produção emitidas pela fábrica, com datas e estado de progresso.
  - Atributos: ordemID, dataEmissao, dataPrevistaConclusao, estado
- **ItemOrdemFabricao:** Indica os produtos a serem produzidos numa ordem de fabrico, com quantidade e ficha técnica associada
  - Atributos: itemOrdemID, *ordemID* (FK para **OrdemFabricao**), *produtoID* (FK para **Produto**), quantidadePlaneada, *fichaTecnicaID* (FK para **FichaTecnica**), estadoProducao
- **ExecucaoOperacao:** Regista a execução de cada operação, com controlo de quantidades, perdas e custos
  - Atributos: execucaoID, *itemOrdemID* (FK para **ItemOrdemFabricao**), *operacaoID* (FK para **Operacao**), *subcontratadoID* (FK para **Subcontratado**), quantidadeEnviada, quantidadeRecebida, dataExecucao, custo, quantidadePerdida
- **AlertaAtraso:** Guarda alertas gerados automaticamente sempre que uma ordem ultrapassa o prazo
  - Atributos: alertaID, *ordemID* (FK para **OrdemFabricao**), dataGeracao, motivo
- **AlertaQualidade:** Regista alertas de qualidade sempre que há perdas superior ao aceitável (acima de 15% de perdas)
  - Atributos: alertaQualidadeID, *execucaoID* (FK para **ExecucaoOperacao**), *subcontratadoID* (FK para **Subcontratado**), descricao, dataGeracao
- **ConsumoMaterialReal:** Representa o registo da quantidade real de material utilizada e a percentagem de perda numa execução específica de operação
  - Atributos: consumoID, *execucaoID* (FK para **ExecucaoOperacao**), *materialID* (FK para **Material**), quantidadeUtilizada, dataRegisto, percentagemPerdaMaterial

## 3.2 Domínio dos Atributos

O processo de criação de uma base de dados relacional para uma fábrica de calçado requer uma análise cuidada e preparação rigorosa, de forma a garantir que o sistema seja robusto, eficiente e adequado às necessidades reais da produção. Um projeto mal planeado pode resultar em atrasos na entrega, baixa performance, falhas de integridade nos dados ou insatisfação por parte dos utilizadores.

Após levantamento dos requisitos, a definição dos processos e a modelação conceptual, foi elaborado um documento de domínio dos atributos. Este documento específico, para cada tabela do sistema, os seus respetivos atributos, os tipos de dados utilizados, os valores permitidos (quando aplicável), e as restrições definidas – como chaves primárias, obrigatoriedade de preenchimento (**Not Null**), valores únicos (**Unique**), ou geração automática de identificadores (**Automatic Identity**).

Esta estrutura permitiu garantir consistência na implementação e suporte às regras de negócio definidas, como o controle de operações, registo de perdas, subcontratação de serviços, monitorização de atrasos e avaliação de desempenho.

Posto isto foram propostas as seguintes tabelas:

Atributo	Tipo de Dados	Domínio / Exemplo	NN	UQ	AI	K
<b><u>produtoID</u></b>	INT	Identificador do produto (1, 2, 3, ...)	x	x	x	<b>PK</b>
<i>modelo</i>	VARCHAR(50)	Modelo do artigo (ex: 'X100')	x	x		
<i>variante</i>	VARCHAR(50)	Variante (design, etc.)		x		
<i>cor</i>	VARCHAR(50)	Cor do produto (ex: 'Preto')	x			
<i>nome</i>	VARCHAR(50)	Nome comercial (ex: 'Bota Urbana')	x			
<i>descricao</i>	VARCHAR(255)	Descrição livre				
<i>dataCriacao</i>	DATETIME2	Gerado automaticamente ( <i>default</i> )				
<i>dataAtualizacao</i>	DATETIME2	Atualizado por <i>trigger</i>				
<i>precoVenda</i>	DECIMAL(10,2)					

Tabela 1 - Domínio da Tabela "Produto"

- A chave primária do **Produto** foi definida para ser gerado de forma automática para facilitar na atribuição de identificadores aos produtos. Os campos *modelo*, *cor*, *nome* são obrigatórios para que cada produto seja bem identificado, nos quais *modelo* e *variante* garantem uma unicidade no **Produto**. Cada vez que seja feita uma atualização (*update*) em relação ao produto, é registado uma data.

Atributo	Tipo de Dados	Domínio / Exemplo	NN	UQ	AI	K
<u>operacaoID</u>	INT	Identificador da operação (1, 2, 3, ...)	x	x	x	<b>PK</b>
nome	VARCHAR(50)	'Corte', 'Costura', 'Montagem', 'Acabamento'	x			
descricao	VARCHAR(255)	Descrição da operação				

Tabela 2 - Domínio da Tabela "Operacao"

- A tabela **Operacao** descreve todas as etapas envolvidas na produção de um produto. Cada operação possui um nome e uma descrição. Estas informações são fundamentais para a estimativa de custos, tempos de fabrico e controlo de qualidade de acordo com a operação. No campo *nome* os nomes válidos são só aqueles que estão no exemplo.

Atributo	Tipo de Dados	Domínio / Exemplo	NN	UQ	AI	K
<u>materialID</u>	INT	Identificador do material (1, 2, 3, ...)	x	x	x	<b>PK</b>
nome	VARCHAR(50)	Nome do material (ex: 'Couro')	x	x		
tipo	VARCHAR(50)	Tipo do material (ex: 'Natural')				
unidadeMedida	DECIMAL(10,2)	Unidade de medida (ex: cm, m, l, etc.)				
custoUnitario	DECIMAL(10,2)	Preço por unidade (ex: 3.50)	x			

Tabela 3 - Domínio da Tabela "Material"

- A tabela **Material** armazena todos os tipos de materiais utilizados no fabrico dos produtos. Cada material tem um identificador único e atributos como o *nome*, *tipo* (ex: couro, tecido), *medida* e o seu custo unitário. Esta informação é essencial para o cálculo do custo de produção e para a gestão de inventário.

Atributo	Tipo de Dados	Domínio / Exemplo	NN	UQ	AI	K
<u>subcontratadoID</u>	INT	Identificador do subcontratado	x	x	x	<b>PK</b>
nome	VARCHAR(100)	Nome da empresa/parceiro	x			
morada	VARCHAR(255)	Endereço completo				
contacto	VARCHAR(50)	Telefone, E-mail, etc.		x		
custoServico	DECIMAL(10, 2)	Custo por unidade de serviço prestado	x			

Tabela 4 - Domínio da Tabela "Subcontratado"

- A tabela **Subcontratado** guarda os dados dos parceiros externos responsáveis por realizar operações subcontratadas. Esta entidade é essencial para gestão de custos e rastreamento da qualidade das execuções terceirizadas de operações.

Atributo	Tipo de Dados	Domínio / Exemplo	NN	UQ	AI	K
<b><u>fichaTecnicaID</u></b>	INT	Identificador da ficha (1, 2, 3, ...)	x	x	x	<b>PK</b>
<i>produtoID</i>	INT	FK para <b>Produto</b>	x			<i>FK</i>
<i>origem</i>	VARCHAR(50)	Natureza da ficha se é interna, ou externa ( <i>default</i> )	x			
<i>descricao</i>	VARCHAR(255)	Descrição geral da ficha				
<i>dataCriacao</i>	DATETIME2	Gerado automaticamente ( <i>default</i> )				
<i>dataAtualizacao</i>	DATETIME2	Atualizado por <i>trigger</i>				
<i>estado</i>	VARCHAR(50)	'ativo', 'inativo'	x			

Tabela 5 - Domínio da Tabela "FichaTecnica"

- Cada **FichaTecnica** representa a especificação de produção de um determinado produto. Inclui informações como a descrição do processo, o estado da ficha (ativa ou inativa) e está associada a um **Produto** através de uma chave estrangeira. Os campos *estado* e *produtoID* são obrigatórios para garantir a integridade das associações e do ciclo de vida do produto o qual possui uma **FichaTecnica**.

Atributo	Tipo de Dados	Domínio / Exemplo	NN	UQ	AI	K
<b><u>detalheID</u></b>	INT	Identificador da ficha detalhada (1, 2, 3, ...)	x	x	x	<b>PK</b>
<i>fichaTecnicaID</i>	INT	FK para <b>FichaTecnica</b>	x	x		<i>FK</i>
<i>operacaoID</i>	INT	FK para <b>Operacao</b>	x			<i>FK</i>
<i>ordemExecucao</i>	DECIMAL(10,2)	Ordem da operação no processo (ex: 1, 2, 3)	x	x		
<i>vezesPorUnidade</i>	DECIMAL(10,2)	Quantidade de material usada	x			
<i>descricao</i>	VARCHAR(255)	Observações adicionais				
<i>duracaoUnitariaMin</i>	DECIMAL(5,2)	Tempo estimado, em minutos para executar a operação				
<i>custoMinuto</i>	DECIMAL(10,2)	Preço da mão de obra por minuto e por cada execução de operação				

Tabela 6 - Domínio da Tabela "DetalheFichaTecnica"

- A tabela **DetalheFichaTecnica** representa a estrutura detalhada da produção de um produto. Cada registo associa uma operação (como "Costura" ou "Montagem") a uma ficha técnica, podendo incluir um material necessário, a quantidade usada e a ordem de execução no processo. Isto permite definir, passo a passo, o ciclo completo de fabrico. Para aspeto financeiro e análise de eficiência os campos *duracaoUnitariaMin* e *custoMinuto* são definidos se necessário.

Atributo	Tipo de Dados	Domínio / Exemplo	NN	UQ	AI	K
<u>ordemID</u>	INT	Identificador da ordem de fabrico (1, 2, 3)	x	x	x	<b>PK</b>
dataEmissao	DATE	Data de criação da ordem de fabrico (ex: '2025-05-01')	x			
dataPrevistaConclusao	DATE	Prazo estimado para terminar a produção	x			
estado	VARCHAR(20)	'planeada', 'em_execucao', 'concluida', 'cancelada'	x			

Tabela 7 - Domínio da Tabela "OrdemFabrico"

- A tabela **OrdemFabrico** representa cada ordem de produção emitida pela fábrica. Guarda dados essenciais como a data de emissão, a data prevista para conclusão e o estado atual da ordem. O campo *estado* permite acompanhar o progresso da produção (ex: planeada, em execução, concluída ou cancelada).

Atributo	Tipo de Dados	Domínio / Exemplo	NN	UQ	AI	K
<u>itemOrdemID</u>	INT	Identificador da ordem do produto (1, 2, 3)	x	x	x	<b>PK</b>
ordemID	INT	FK para <b>OrdemFabrico</b>	x	x		<i>FK</i>
produtoID	INT	FK para <b>Produto</b>	x	x		<i>FK</i>
quantidadePlaneada	INT	Quantidade a ser produzida	x			
fichaTecnicaID	INT	FK para <b>FichaTecnica</b>	x			<i>FK</i>
estadoProducao	INT	Progresso da produção do produto (‘não iniciado’, ‘Cortado’, ‘Costurado’, 'Montado', 'Acabado', 'em progresso')				

Tabela 8 - Domínio da Tabela "ItemOrdemFabrico"

- Esta tabela **ItemOrdemFabrico** liga os produtos às ordens de fabrico, permitindo especificar quais produtos serão fabricados em cada ordem e em que quantidade. É associado a cada item de produção a sua respetiva ficha técnica, garantindo que o processo de fabrico correto será seguido. Também o progresso da produção do produto pode ser acompanhado através do campo *estadoProducao*.

Atributo	Tipo de Dados	Domínio / Exemplo	NN	UQ	AI	K
<b><u>execucaoID</u></b>	INT	Identificador da execução	x	x	x	<b>PK</b>
<i>itemOrdemID</i>	INT	FK para <b>ItemOrdemFabricao</b>	x	x		<i>FK</i>
<i>operacaoID</i>	INT	FK para <b>Operacao</b>	x	x		<i>FK</i>
<i>subcontratadoID</i>	INT	FK para <b>Subcontratado</b> (se aplicável)				<i>FK</i>
<i>precoPorUnidade</i>	DECIMAL(10,2)	Valor para execução de uma única unidade				
<i>quantidadeEnviada</i>	INT	Quantidade enviada para produção	x			
<i>quantidadeRecebida</i>	INT	Quantidade recebida após produção	x			
<i>dataExecucao</i>	DATE	Data da execução				
<i>custo</i>	DECIMAL(10,2)	Custo da operação (interno ou subcontratado)	x			
<i>quantidadePerdida</i>	INT	Informa a quantidade perdida após a execução				

Tabela 9 - Domínio da Tabela "ExecucaoOperacao"

- Esta tabela **ExecucaoOperacao** regista a execução de cada operação associada a um item de ordem de fabrico. Para além das quantidades enviadas e recebidas, regista também o custo, a data de execução e a quantidade de perdas. Também identifica, se aplicável, o subcontratado responsável pela operação. Estes dados são fundamentais para controlo da produção e avaliação de desempenho.

Atributo	Tipo de Dados	Domínio / Exemplo	NN	UQ	AI	K
<b><u>alertaAtrasoID</u></b>	INT	Identificador único do alerta	x	x	x	<b>PK</b>
<i>ordemID</i>	INT	FK para <b>OrdemFabricao</b>	x			<i>FK</i>
<i>dataGeracao</i>	DATE	Data em que o alerta foi gerado	x			
<i>motivo</i>	VARCHAR(255)	Motivo do alerta				

Tabela 10 - Domínio da Tabela "AlertaAtraso"

- A tabela **AlertaAtraso** regista todos os alertas emitidos automaticamente quando uma ordem de fabrico ultrapassa a data prevista de conclusão. Serve como controlo de prazos e auxilia na identificação de ordens problemáticas.



Atributo	Tipo de Dados	Domínio / Exemplo	NN	UQ	AI	K
<b><u>alertaQualidadeID</u></b>	INT	Identificador do único do alerta	x	x	x	<b>PK</b>
<i>execucaoID</i>	INT	FK para <b>ExecucaoOperacao</b>	x	x		<i>FK</i>
<i>subcontratadoID</i>	INT	FK para <b>Subcontratado</b> (se aplicável)				<i>FK</i>
<i>descricao</i>	VARCHAR(255)	Motivo do alerta (ex: "perdas > 15%")				
<i>dataGeracao</i>	DATE	Data em que o alerta foi gerado	x			

Tabela 11 - Domínio da Tabela "AlertaQualidade"

- A tabela **AlertaQualidade** regista alertas relacionados com perdas elevadas em operações de produção, normalmente superiores a 15%. Está associada a uma execução específica e pode também identificar o subcontratado responsável. Este controlo permite detetar e corrigir problemas de qualidade no processo de fabrico.

Atributo	Tipo de Dados	Domínio / Exemplo	NN	UQ	AI	K
<b><u>consumoID</u></b>	INT	Identificador do consumo de recursos para a produção	x	x	x	<b>PK</b>
<i>execucaoID</i>	INT	FK para <b>ExecucaoOperacao</b>	x			<i>FK</i>
<i>detalheID</i>	INT	FK para <b>DetalheFichaTecnica</b>	x			<i>FK</i>
<i>materialID</i>	INT	FK para <b>Material</b>	x			<i>FK</i>
<i>quantidadePlaneada</i>	DECIMAL(10,2)	Quantidade de recursos planeados	x			
<i>quantidadeUtilizada</i>	DECIMAL(10,2)	Quantidade de recursos utilizados				
<i>dataRegisto</i>	DATETIME2	Data de registo do controlo de recursos				
<i>percentagemPerdaMaterial</i>	DECIMAL(5,2)	Perca de recursos durante a produção				

Tabela 12 - Domínio da Tabela "ConsumoMaterialReal"

- A tabela **ConsumoMaterialReal** armazena os registos dos materiais consumidos durante a execução de operações produtivas, permitindo o controlo real de consumo e perdas por material, por operação e por execução.

## 3.3 Modelo Lógico

O modelo lógico representa uma versão detalhada da estrutura da base de dados, servindo como **ponte entre o modelo conceptual e a implementação física** no sistema de gestão de base de dados – neste caso, o **SQL Server**. Através deste modelo, o sistema ganha forma relacional, com tabelas, campos e restrições que asseguram a integridade e coerência dos dados armazenados.

### 3.3.1 Objetivos

- **Conversão do modelo conceptual em estrutura relacional:**  
Transformar as entidades e relações identificadas na fase conceptual em tabelas relacionais, compatíveis com o SGBD utilizado.
- **Definições dos tipos de dados:**  
Atribuir tipos de dados apropriados a cada atributo, como *INT*, *VARCHAR*, *DECIMAL*, *DATE*, de forma a garantir a integridade da informação e a eficiência de armazenamento.
- **Estabelecimento de relações:**  
Definir **chaves estrangeiras** que ligam as tabelas entre si, assegurando correspondência entre registos relacionados, como por exemplo entre **Produto** e **FichaTecnica**.
- **Aplicações de restrições de integridade:**  
Implementar **chaves primárias**, restrições de domínio (*CHECK*), e outras regras de validação, como a obrigatoriedade de certas colunas (*NOT NULL*) ou a verificação da ordem de execução das operações.

### 3.3.2 Componentes do Modelo Lógico

- **Tabelas (Entidades):**

Foram criadas tabelas para representar cada entidade identificada, como **Produto**, **Operacao**, **FichaTecnica**, **OrdemFabrico**, **ConsumoMaterialReal**, entre outras.

- **Colunas (Atributos):**

Cada tabela inclui atributos específicos com os respetivos tipos de dados definidos de acordo com o seu conteúdo. Por exemplo, *percentagemPerdas* foi definido como *DECIMAL(5,2)*, *dataCriacao* como *DATETIME2*, ou até mesmo o uso do *IDENTITY(1,1)* para permitir a incrementação de uma *seed* interna (espécie de *counter*) em certas chaves primárias.

- **Chaves Primárias (PK):**

Garantem a unicidade de cada registo em todas as tabelas. Ex.: produtoID, fichaTecnicaID, execucaoID

- **Chaves Estrangeiras (FK):**

Estabelecem as ligações entre as tabelas. Exemplo: a tabela **ExecucaoOperacao** contém referências para **ItemOrdemFabrico**, **Operacao** e opcionalmente **Subcontratado**.

### 3.3.3 Diagrama do Modelo Lógico

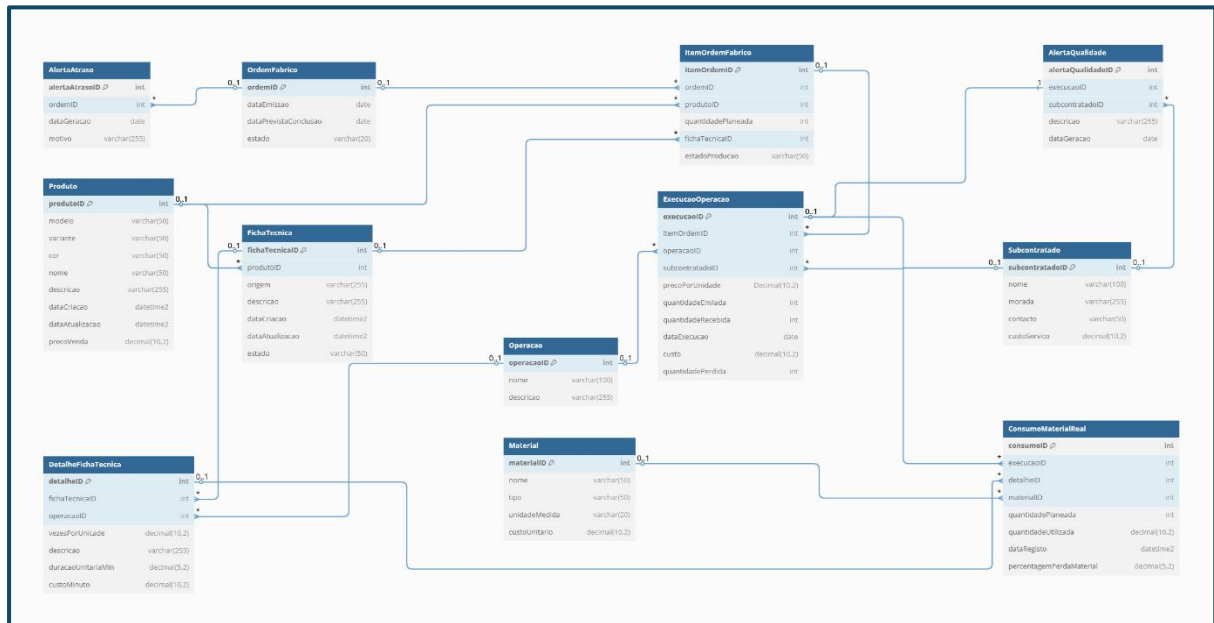


Figura 2 - Diagrama ER (Modelo Lógico)

### 3.3 Modelo Físico

O modelo físico corresponde à implementação concreta do modelo lógico no sistema de gestão de base de dados utilizado — neste caso, o SQL Server. Nesta fase, foram definidas todas as tabelas, com os respetivos campos, tipos de dados, chaves primárias e estrangeiras, bem como as restrições de integridade necessárias ao correto funcionamento do sistema.

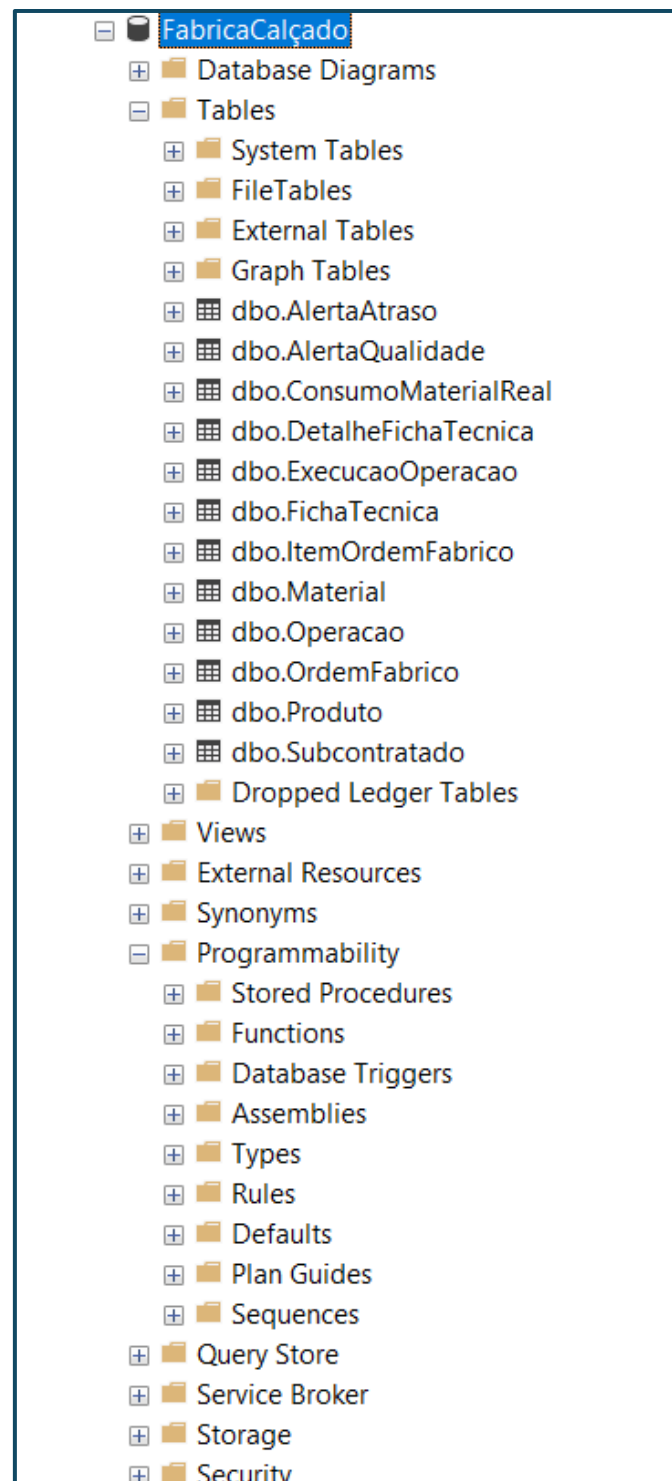


Figura 3 - Estrutura do Modelo Físico

## 4. Implementação

### 4.1 Povoamento da Base de Dados

Para que fosse possível a realização dos testes na base de dados, tanto de consulta, inserção, atualização povoamos a nossa base de dados com dados fictícios. Esta etapa é crucial para assegurar que todas as funcionalidades (consulta, inserção, atualização e entre outras) sejam testadas de forma realista e eficiente permitindo simular cenários propostos no enunciado. A utilização de dados representativos possibilita ainda a identificação antecipada de possíveis inconsistências, garantindo a robustez e confiabilidade do sistema.

O povoamento não se limitou à inserção de dados: incluiu também a criação de **objetos auxiliares** como *triggers*, *procedures*, e *views*, os quais são fundamentais para suportar regras de negócio e automatizações na base de dados.

#### 4.2.1 Objetivos do Povoamento da Base de Dados

- **Inicialização:** foram fornecidos dados exemplo que permitiram a inicialização da base de dados.
- **Testes de funcionalidades:** assegurar que todas as operações CRUD possam ser testadas de forma eficaz.
- **Validação de regras de negócio:** verificar se as regras de negócio implementadas estão a ser cumpridas conforme o esperado com os automatismos implementados via *triggers* e *procedures*.

#### 4.2.2 Etapas do Povoamento

- **Definição da Estrutura:** Antes de inserir os dados, garantimos que as relações entre tabelas estejam bem definidas e implementadas de acordo com a base do modelo de dados projetado.
- **Ordem de inserção:** Foram garantidas ordens de inserção para que os dados inseridos obedecem à lógica do negócio respeitando assim a relação entre tabelas
- **Script de inserção:** Utilizamos um *script SQL*, para inserir os dados iniciais. Esses *scripts* forma devidamente documentados e organizados para futuras manutenções/atualizações.

### 4.2.3 Conceito do Povoamento na Prática

A ordem de povoamento foi a seguinte:

1. **Produto** (não tem dependências)
2. **Operacao** (não tem dependências)
3. **Material** (não tem dependências)
4. **Subcontratado** (não tem dependências)
5. **FichaTecnica**
6. **DetalheFichaTecnica**
7. **OrdemFabricao**
8. **ItemOrdemFabricao**
9. **ConsumoMaterialReal**

O *script* completo para o povoamento da base de dados encontra-se disponível nos anexos ("SCRIPT/Povoamento por ordem ou SCRIPT/ALL\_IN\_ONE/1\_all\_in\_one\_tables.sql").

## 5. Regras de Negócio Implementadas

A base de dados desenvolvida para o sistema de produção da fábrica de calçado não se limita apenas à estruturação de tabelas e relações entre entidades. Foi também fundamental incorporar um conjunto de regras de negócio que asseguram o correto funcionamento do processo produtivo, evitando inconsistências e automatizando tarefas repetitivas ou críticas.

Estas regras foram implementadas através de *triggers* (gatilhos), *constraints* e *cálculos automáticos*, permitindo validar dados, controlar o fluxo de produção, calcular custos e perdas, gerar alertas de qualidade e atualizar automaticamente os estados dos elementos envolvidos.

A seguir, são descritas as principais regras de negócio implementadas, bem como o seu impacto na integridade e eficiência do sistema.

As regras aplicadas para cada tabela foram as seguintes:

### 1. Produto

- A data de criação (*dataCriacao*) não pode ser alterada após a inserção inicial, garantindo assim que os dados não possam ser adulterados.
- A data de atualização (*dataAtualizacao*) é automaticamente atualizada sempre que há uma modificação no produto.
- O preço de venda (*precoVenda*) é recalculado automaticamente com base no custo da ordem de fabrico concluída, adicionando uma margem de lucro de 30% (definido por nós, pode ser alterado no futuro).

### 2. Subcontratado

- Um subcontratado **não pode ser excluído** se tiver associado a uma operação.

### 3. Ficha Técnica

- A data de criação (*dataCriacao*) é imutável após a criação.
- A data de atualização (*dataAtualizacao*) é atualizada automaticamente após alterações.
- Não é possível **inativar uma ficha técnica** se ela estiver associada a ordens de fabrico ativas.
- Fichas do tipo "**interna**" devem conter **exatamente quatro operações**: Corte, Costura, Montagem e Acabamento, com ordens de execução de 1 a 4, garantindo assim que todas as operações têm de seguir uma obrigatoriedade das 4 operações.
- Fichas do tipo "**externa**" não podem ter ordem de execução superior ao número total de operações, isto faz com que por exemplo se somos subcontratados para fazer X operações pode não haver necessidade de incluir obrigatoriamente todas as operações, mas somente 1 pelo menos.
- Operações associadas a fichas técnicas **ativas não podem ser excluídas**.

#### 4. Item de Ordem de Fabrico

- Só pode utilizar fichas técnicas **ativas**.
- Ao inserir um item, são automaticamente criadas execuções para as operações correspondentes.

#### 5. Ordem de Fabrico

- A **data de emissão** não pode ser modificada.
- A ordem só pode ser marcada como **concluída** se todas as operações associadas tiverem sido executadas com sucesso.
- Ao receber a primeira execução de operação, a ordem muda automaticamente para o estado *em\_execucao*.
- Quando todas as operações forem concluídas (estado no *ItemOrdemFabrico* "Acabado"), a ordem muda automaticamente para concluída.
- Um **alerta de atraso** é gerado se a data prevista de conclusão for ultrapassada sem que a ordem tenha sido concluída.

#### 6. Execução de Operação

- Não é permitido:
  - Alterar manualmente os campos custo ou quantidadePerdida.
  - Registrar uma quantidade recebida maior que a enviada.
  - Definir *precoPorUnidade* nulo ou  $\leq 0$  em fichas externas/parciais sem subcontratado, garantindo assim que todas as operações ao qual a nossa fábrica é subcontratada, tem um valor por unidade a ser cobrado.
  - Saltar operações (ordem incorreta de execução).
  - Enviar quantidades maiores que a recebida na operação anterior, se na operação de corte forem enviadas para lá 50 unidades e saírem 50, não faz sentido na operação a seguir serem enviadas 100 peças se na operação anterior não saíram 100 peças.
- Ao registar a primeira execução com quantidade recebida  $> 0$ :
  - A *dataExecucao* é automaticamente preenchida.
  - É bloqueada qualquer tentativa de modificar essa data para nulo ou retroativamente.
- Cálculo automático:
  - *quantidadePerdida* (diferença entre enviada e recebida).
  - custo com base na origem da ficha técnica (interna ou externa).
  - Consumo real de material e sua perda percentual.
- Se a perda de material exceder 15%, um **alerta de qualidade** é gerado automaticamente.



## 7. Alertas

- **Não é possível excluir alertas** de atraso ou de qualidade.
- A data de geração (*dataGeracao*) dos alertas não pode ser alterada.

## 8. Consumo de Material

- Não pode ser excluído após registo.
- A perda de material é calculada automaticamente com base na quantidade planeada vs. Utilizada.

## 6. Consultas SQL

Neste capítulo apresentam-se as principais consultas SQL implementadas no sistema, com o objetivo de responder aos requisitos funcionais e apoiar a gestão da produção. As *queries* visam fornecer informação útil, identificar problemas e calcular indicadores relevantes

### 6.1 Consultas

#### 6.1.1 As ordens de fabrico emitidas e o seu estado atual

```
SELECT * FROM OrdemFabrico ORDER BY dataEmissao DESC;
```

	ordemID	dataEmissao	dataPrevistaConclusao	estado
1	1	2025-05-20	2025-05-21	em_execucao
2	2	2025-05-19	2025-05-20	planeada
3	3	2025-05-18	2025-05-22	planeada
4	4	2025-05-17	2025-05-21	planeada
5	5	2025-05-16	2025-05-22	planeada
6	6	2025-05-15	2025-05-23	planeada
7	7	2025-05-14	2025-05-22	planeada
8	8	2025-05-13	2025-05-23	planeada
9	9	2025-05-12	2025-05-23	planeada
10	10	2025-05-11	2025-05-23	concluida

Figura 4 - Resultado da consulta [6.1.1]

- Todas as **OrdemFabrico**, permitindo visualizar todos os dados disponíveis para cada registo. Ordenando os resultados pela coluna *dataEmissao* em ordem decrescente, o que significa que as ordens de fabrico mais recentes aparecem primeiro nesta consulta.

## 6.1.2 Quantidades de materiais enviados para produção

```
SELECT
    ofa.ordemID,
    p.nome AS Produto,
    o.nome AS Operacao,
    eo.execucaoID,
    m.nome AS Material,
    cmr.quantidadeUtilizada AS QuantidadeUtilizada,
    cmr.quantidadePlaneada AS QuantidadePlaneadaPorUnidade,
    ROUND(cmr.quantidadePlaneada * eo.quantidadeEnviada, 2) AS
QuantidadePlaneadaTotal,
    cmr.percentagemPerdaMaterial,
    cmr.dataRegisto
FROM ConsumoMaterialReal cmr
JOIN ExecucaoOperacao eo ON cmr.execucaoID = eo.execucaoID JOIN ItemOrdemFabrico io ON
eo.itemOrdemID = io.itemOrdemID
JOIN OrdemFabrico ofa ON io.ordemID = ofa.ordemID
JOIN Produto p ON io.produtoID = p.produtoID
JOIN Operacao o ON eo.operacaoID = o.operacaoID
JOIN Material m ON cmr.materialID = m.materialID
WHERE cmr.quantidadeUtilizada IS NOT NULL
ORDER BY ofa.ordemID, eo.execucaoID, m.nome;
```

	ordemID	Produto	Operacao	execucaoID	Material	QuantidadeUtilizada	QuantidadePlaneadaPorUnidade	QuantidadePlaneadaTotal	percentagemPerdaMaterial	dataRegisto
1	1	Bota de Couro Cano Curto Castanha	Corte	60	Atacadores	50.00	1.00	50.00	0.00	2025-05-23 10:52:08.3265406
2	1	Bota de Couro Cano Curto Castanha	Corte	60	Etiquetas	50.00	1.00	50.00	0.00	2025-05-23 10:52:08.3265406
3	1	Bota de Couro Cano Alto Preta	Corte	64	Pele Genuína	59.00	1.00	50.00	18.00	2025-05-23 01:06:41.3883855
4	1	Bota de Couro Cano Alto Preta	Corte	64	Tecido Sintético	102.00	2.00	100.00	2.00	2025-05-23 01:06:41.3883855

Figura 5 - Resultado da consulta [6.1.2]

- Esta *query* tem como objetivo listar os materiais efetivamente utilizados em cada execução de operação numa ordem de fabrico, comparando com o valor planeado ao realmente utilizado (que posteriormente o utilizador altera caso os valores sejam maiores que). A *query* vai buscar as execuções de operações que já têm materiais consumidos, relaciona esses dados com a ordem de fabrico, o **Produto**, a **Operacao** e o **Material** e compara com o que estava planeado com o que foi realmente usado.

### 6.1.3 Quantidade de produtos semi-acabados ou acabados

```
SELECT
    o.ordemID,
    p.nome AS produto,
    op.nome AS operacao,
    eo.quantidadeEnviada,
    eo.quantidadeRecebida,
    eo.dataExecucao
FROM ExecucaoOperacao eo
JOIN ItemOrdemFabrico iof ON eo.itemOrdemID = iof.itemOrdemID
JOIN OrdemFabrico o ON iof.ordemID = o.ordemID
JOIN Produto p ON iof.produtoID = p.produtoID
JOIN Operacao op ON eo.operacaoID = op.operacaoID
WHERE eo.quantidadeRecebida > 0 OR eo.quantidadeEnviada > 0
ORDER BY o.ordemID, iof.itemOrdemID, eo.operacaoID;
```

	ordemID	produto	operacao	quantidadeEnviada	quantidadeRecebida	dataExecucao
1	1	Bota de Couro Cano Alto Preta	Corte	50	50	2025-05-23
2	1	Bota de Couro Cano Alto Preta	Costura	50	50	2025-05-23

Figura 6 - Resultado da consulta [6.1.3]

- Esta consulta dá-nos todas as **Operacao** já realizadas para cada **OrdemFabrico**, indicando o **Produto** associado, o tipo de operação (Corte, Costura,...) a quantidade de unidades enviadas para a operação (quantidade de sapatos neste caso) e a quantidades recebidas de volta.

### 6.1.4 Percentagem de perdas (materiais defeituosos ou desperdiçados)

```
SELECT
    o.ordemID,
    p.nome AS produto,
    op.nome AS operacao,
    eo.quantidadeEnviada,
    eo.quantidadeRecebida,
    eo.quantidadeEnviada - eo.quantidadeRecebida AS quantidadePerdida,
    ROUND(
        CASE
            WHEN eo.quantidadeEnviada = 0 THEN 0
            ELSE 100.0 * (eo.quantidadeEnviada - eo.quantidadeRecebida) /
eo.quantidadeEnviada
        END, 2
    ) AS percentagemPerdaProduto,
    m.nome AS material, cmr.quantidadeUtilizada,
    ROUND(
        CASE
            WHEN eo.quantidadeEnviada = 0 THEN NULL
            ELSE 100.0 * (cmr.quantidadeUtilizada / eo.quantidadeEnviada)
        END, 2
    ) AS consumoMedioPorUnidade,
    eo.dataExecucao
FROM ExecucaoOperacao eo
JOIN ItemOrdemFabrico iof ON eo.itemOrdemID = iof.itemOrdemID
JOIN OrdemFabrico o ON iof.ordemID = o.ordemID
JOIN Produto p ON iof.produtoID = p.produtoID
JOIN Operacao op ON eo.operacaoID = op.operacaoID
LEFT JOIN ConsumoMaterialReal cmr ON eo.execucaoID = cmr.execucaoID
LEFT JOIN Material m ON cmr.materialID = m.materialID
WHERE eo.quantidadeEnviada > 0
ORDER BY o.ordemID, p.nome, eo.execucaoID, m.nome;
```

Results		Messages									
	ordemID	produto	operacao	quantidadeEnviada	quantidadeRecebida	quantidadePerdida	percentagemPerdaProduto	material	quantidadeUtilizada	consumoMedioPorUnidade	dataExecucao
1	1	Bota de Couro Cano Alto Preta	Corte	50	50	0	0.000000000000	Pele Genuína	30.00	60.00000000000000	2025-05-23
2	1	Bota de Couro Cano Alto Preta	Corte	50	50	0	0.000000000000	Tecido Sintético	12.00	24.00000000000000	2025-05-23
3	1	Bota de Couro Cano Alto Preta	Costura	50	50	0	0.000000000000	Linha de Costura	3.00	6.00000000000000	2025-05-23

Figura 7 - Resultado da consulta [6.1.4]

- Esta *query* apresenta, por **OrdemFabrico** e **Operacao**, a eficiência da produção, calculando a percentagem de perda de **Produto** com base nas *quantidades enviadas e recebidas*. Além disso, exibe o consumo médio de material por *unidade* produzida, bem como a *data de execução* da **Operacao**. Também são incluídos o *nome* do **Produto**, o *nome* da **Operacao** e o **Material** utilizado, caso exista.

### 6.1.5 Os subcontratados envolvidos e os custos associados

```
SELECT
    o.ordemID,
    p.nome AS produto,
    op.nome AS operacao,
    sc.nome AS subcontratado,
    eo.quantidadeEnviada,
    eo.quantidadeRecebida,
    eo.custo,
    eo.dataExecucao
FROM ExecuciaoOperacao eo
JOIN ItemOrdemFabrico iof ON eo.itemOrdemID = iof.itemOrdemID
JOIN OrdemFabrico o ON iof.ordemID = o.ordemID
JOIN Produto p ON iof.produtoID = p.produtoID
JOIN Operacao op ON eo.operacaoID = op.operacaoID
JOIN Subcontratado sc ON eo.subcontratadoID = sc.subcontratadoID
WHERE eo.quantidadeEnviada > 0
ORDER BY o.ordemID, p.nome, eo.execucaoID;
```

	ordemID	produto	operacao	subcontratado	quantidadeEnviada	quantidadeRecebida	custo	dataExecucao
1	1	Bota de Couro Cano Alto Preta	Montagem	Oficina de Costura Silva	50	50	75.00	2025-05-23
2	1	Bota de Couro Cano Alto Preta	Acabamento	Oficina de Costura Silva	50	40	75.00	2025-05-23

Figura 8 - Resultado da consulta [6.1.5]

- Esta *query* lista todas as operações executadas por **Subcontratado**, apresentando a ordem de fabrico, o **Produto** e a **Operacao** correspondente. Também exhibe o *nome* do **Subcontratado** envolvido, as *quantidades enviadas* e *recebidas*, o *custo* associado à **Operacao** e a *data de execução*. Trata-se de uma ferramenta útil para a análise de custos externos e para o acompanhamento da produção terceirizada.

## 6.1.6 Consultar as ordens de fabrico em curso, com indicação do progresso

```
ALTER VIEW vw_OrdensFabrico_Progresso AS
WITH UltimaOperacaoExecutada AS (
    SELECT
        eo.itemOrdemID,
        MAX(dft.ordemExecucao) AS ultimaEtapa
    FROM ExecuciaoOperacao eo
    JOIN DetalheFichaTecnica dft ON eo.operacaoID = dft.operacaoID
    WHERE eo.quantidadeRecebida > 0
    GROUP BY eo.itemOrdemID
),
UltimaExecucaoFinal AS (
    SELECT *
    FROM (
        SELECT
            eo.itemOrdemID,
            eo.quantidadeRecebida,
            ROW_NUMBER() OVER (
                PARTITION BY eo.itemOrdemID
                ORDER BY eo.execucaoID DESC
            ) AS rn
        FROM ExecuciaoOperacao eo
        JOIN DetalheFichaTecnica dft ON eo.operacaoID = dft.operacaoID
        JOIN UltimaOperacaoExecutada uo
            ON eo.itemOrdemID = uo.itemOrdemID AND dft.ordemExecucao = uo.ultimaEtapa
        ) AS ult
    WHERE ult.rn = 1
)
SELECT
    ofa.ordemID,
    ofa.dataEmissao,
    ofa.dataPrevistaConclusao,
    io.itemOrdemID,
    p.nome AS nomeProduto,
    CONCAT(p.modelo, ' ', p.cor, ' ', p.variante) AS produtoDetalhado,
    io.quantidadePlaneada,
    ISNULL(uef.quantidadeRecebida, 0) AS quantidadeProduzida,
    ROUND(
        100.0 * ISNULL(uef.quantidadeRecebida, 0) / NULLIF(io.quantidadePlaneada, 0),
        2
    ) AS percentagemProgresso,
    io.estadoProducao
FROM OrdemFabrico ofa
JOIN ItemOrdemFabrico io ON io.ordemID = ofa.ordemID
JOIN Produto p ON io.produtoID = p.produtoID
LEFT JOIN UltimaExecucaoFinal uef ON uef.itemOrdemID = io.itemOrdemID;
```

100 %										
Results Messages										
	ordemID	dataEmissao	dataPrevistaConclusao	itemOrdemID	nomeProduto	produtoDetalhado	quantidadePlaneada	quantidadeProduzida	percentagemProgresso	estadoProducao
1	1	2025-05-10	2025-05-15	1	Bota de Couro Cano Alto Preta	Bota Preta Cano Alto	100	40	40.000000000000	Acabado
2	1	2025-05-10	2025-05-15	2	Bota de Couro Cano Curto Castanha	Bota Castanha Cano Curto	50	0	0.000000000000	nao iniciado
3	2	2025-05-12	2025-05-18	3	Sandália de Couro Verão	Sandália Castanha Verão	200	0	0.000000000000	nao iniciado
4	3	2025-05-15	2025-05-20	4	Tênis Casual Urbanos	Tênis Preto/Branco Casual	150	0	0.000000000000	nao iniciado
5	4	2025-05-23	2025-05-30	5	Sapato Social Clássico Preto	Sapatos Preto Social	75	0	0.000000000000	nao iniciado
6	5	2025-05-24	2025-06-01	6	Chinelos de Casa Confortáveis	Chinelos Cinzento Casa	300	0	0.000000000000	nao iniciado
7	6	2025-05-18	2025-05-22	7	Bota de Couro Cano Alto Preta	Bota Preta Cano Alto	50	0	0.000000000000	nao iniciado
8	6	2025-05-18	2025-05-22	8	Sabrinhas Clássicas Pretas	Sabrinhas Pretas Clássicas	100	0	0.000000000000	nao iniciado
9	7	2025-05-20	2025-05-25	9	Bota de Couro Cano Curto Castanha	Bota Castanha Cano Curto	75	0	0.000000000000	nao iniciado
10	8	2025-05-11	2025-05-16	10	Sandália de Couro Verão	Sandália Castanha Verão	120	0	0.000000000000	nao iniciado
11	9	2025-05-22	2025-05-29	11	Tênis Casual Urbanos	Tênis Preto/Branco Casual	90	0	0.000000000000	nao iniciado
12	10	2025-05-13	2025-05-19	12	Sapato Social Clássico Preto	Sapatos Preto Social	40	0	0.000000000000	nao iniciado
13	11	2025-05-01	2025-05-07	13	Chinelos de Casa Confortáveis	Chinelos Cinzento Casa	200	0	0.000000000000	nao iniciado
14	12	2025-05-05	2025-05-11	14	Bota Tática Impermeável Preta	Bota Preta Tática	60	0	0.000000000000	nao iniciado
15	13	2025-05-08	2025-05-14	15	Sandália Desportiva Ajustável	Sandália Azul/Cinza Desportiva	180	0	0.000000000000	nao iniciado
16	14	2025-05-25	2025-06-05	16	Tênis Desportivo Treino	Tênis Cinzento/Verde Desportivo	110	0	0.000000000000	nao iniciado
17	15	2025-05-28	2025-06-08	17	Sapato Oxford Formal Preto	Sapatos Preto Oxford	55	0	0.000000000000	nao iniciado
18	16	2025-05-19	2025-05-21	18	Alpargatas de Tecido Azul Marinho	Alpargatas Azul Marinho Casual	250	0	0.000000000000	nao iniciado
19	17	2025-05-22	2025-05-27	19	Sabrinhas Coloridas Vermelhas	Sabrinhas Vermelhas Coloridas	80	0	0.000000000000	nao iniciado

Figura 9 - Resultado da consulta [6.1.6]

- Esta *view* permite consultar o progresso atual de cada item de ordem de fabrico, comparando a *quantidade produzida* com a *quantidade planeada*. Exibe dados relevantes da ordem, como o *ID* e as *datas*, além do *nome* e detalhes do produto. Apresenta ainda a *quantidade planeada* versus a produzida, a percentagem de progresso e o estado atual da produção *estadoProducao*. É uma ferramenta útil para o monitoramento em tempo real da produção, a gestão de prazos e a análise de desempenho.



### 6.1.7 Identificar autom. Ordens com atrasos

```
CREATE VIEW vw_OrdensAtrasadas AS
SELECT
    ordemID,
    dataPrevistaConclusao,
    estado,
    DATEDIFF(DAY, dataPrevistaConclusao, CAST(GETDATE() AS DATE)) AS diasAtraso
FROM OrdemFabrico
WHERE
    estado NOT IN ('concluida', 'cancelada')
    AND dataPrevistaConclusao < CAST(GETDATE() AS DATE);
```

	ordemID	dataPrevistaConclusao	estado	diasAtraso
1	1	2025-05-15	em_execucao	8
2	2	2025-05-18	planeada	5
3	3	2025-05-20	planeada	3
4	6	2025-05-22	planeada	1
5	8	2025-05-16	planeada	7
6	10	2025-05-19	planeada	4
7	11	2025-05-07	planeada	16
8	12	2025-05-11	planeada	12
9	13	2025-05-14	planeada	9
10	16	2025-05-21	planeada	2
11	18	2025-05-09	planeada	14
12	20	2025-05-20	planeada	3

Figura 10 - Resultado da consulta [6.1.7]

- Esta view retorna todas as ordens de fabrico que se encontram em atraso, ou seja, aquelas cuja data prevista de conclusão já passou e que ainda não foram concluídas nem canceladas. Apresenta informações como o ID da ordem, a data prevista de conclusão, o estado atual da ordem e o número de dias de atraso. É especialmente útil para a gestão de prazos, emissão de alertas operacionais e tomada de ações corretivas de forma proativa.

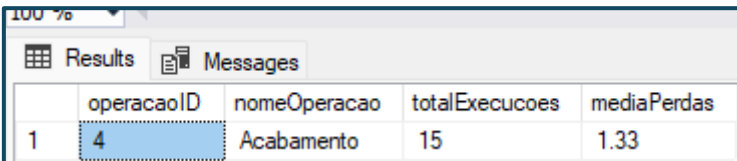
### 6.1.8 Calcular a percentagem média de perdas por operação

```
CREATE VIEW vw_OperacoesProblematicas AS
SELECT
    eo.operacaoID,
    op.nome AS nomeOperacao,
    COUNT(*) AS totalExecucoes,
    CAST(
        AVG(
            CASE
                WHEN eo.quantidadeEnviada > 0
                THEN 100.0 * (eo.quantidadeEnviada - eo.quantidadeRecebida) /
eo.quantidadeEnviada
                ELSE 0
            END
        ) AS DECIMAL(5,2)
```

```

) AS mediaPerdas
FROM ExecuciaoOperacao eo
JOIN Operacao op ON eo.operacaoID = op.operacaoID
GROUP BY eo.operacaoID, op.nome
HAVING AVG(
    CASE
        WHEN eo.quantidadeEnviada > 0
        THEN 100.0 * (eo.quantidadeEnviada - eo.quantidadeRecebida) /
eo.quantidadeEnviada
        ELSE 0
    END
) > 0;

```



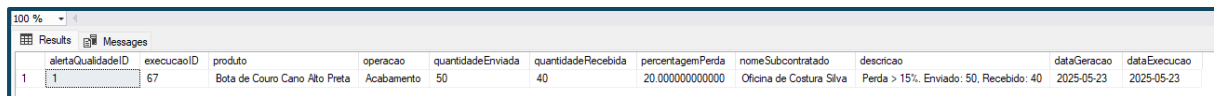
	operacaoID	nomeOperacao	totalExecucoes	mediaPerdas
1	4	Acabamento	15	1.33

Figura 11 - Resultado da consulta [6.1.8]

- Esta view identifica as operações com maior índice médio de perdas ao longo das execuções. Apresenta o nome da operação, o número total de execuções e a média percentual de perda de produto, calculada com base na diferença entre as quantidades enviadas e recebidas. É uma ferramenta útil para a análise da eficiência operacional, a identificação de gargalos e a definição de ações de melhoria nos processos de produção.

### 6.1.9 Gerar relatórios de operações com perdas superiores a 15%

```
CREATE VIEW [dbo].[vw_RelatorioAlertasQualidade] AS
SELECT
    aq.alertaQualidadeID,
    aq.execucaoID,
    p.nome AS produto,
    op.nome AS operacao,
    eo.quantidadeEnviada,
    eo.quantidadeRecebida,
    ROUND(
        100.0 * (eo.quantidadeEnviada - eo.quantidadeRecebida) /
        NULLIF(eo.quantidadeEnviada, 0), 2
    ) AS percentagemPerda,
    s.nome AS nomeSubcontratado,
    aq.descricao,
    aq.dataGeracao,
    eo.dataExecucao
FROM AlertaQualidade aq
JOIN ExecucaoOperacao eo ON aq.execucaoID = eo.execucaoID
JOIN Operacao op ON eo.operacaoID = op.operacaoID
JOIN ItemOrdemFabrico iof ON eo.itemOrdemID = iof.itemOrdemID
JOIN Produto p ON iof.produtoID = p.produtoID
LEFT JOIN Subcontratado s ON aq.subcontratadoID = s.subcontratadoID;
```



alertaQualidadeID	execucaoID	produto	operacao	quantidadeEnviada	quantidadeRecebida	percentagemPerda	nomeSubcontratado	descricao	dataGeracao	dataExecucao
1	67	Bota de Couro Cano Alto Preta	Acabamento	50	40	20.0000000000000	Oficina de Costura Silva	Perda > 15%. Enviado: 50, Recebido: 40	2025-05-23	2025-05-23

Figura 12 - Resultado da consulta [6.1.9]

- Esta *view* apresenta um relatório detalhado de alertas de qualidade gerados durante a produção, sinalizando potenciais problemas no processo produtivo. Inclui informações como o produto e a operação onde ocorreu a perda, as quantidades enviadas e recebidas, a percentagem de perda de produção, o nome do subcontratado (quando aplicável), a descrição do alerta, bem como as datas de geração do alerta e de execução da operação. Trata-se de uma ferramenta essencial para o controlo de qualidade e para a identificação de falhas ou desperdícios significativos no processo produtivo.

### 6.1.10 Listar operações mais freq. subcontratadas e os respetivos custos

```
CREATE VIEW vw_OperacoesMaisSubcontratadas AS
SELECT
    eo.operacaoID,
    op.nome AS nomeOperacao,
    COUNT(*) AS nExecucoesSubcontratadas,
    SUM(ISNULL(so.custoServico, 0) * eo.quantidadeEnviada) AS custoTotalSubcontratado,
    COUNT(DISTINCT eo.subcontratadoID) AS nSubcontratados
FROM ExecucaoOperacao eo
JOIN Operacao op ON eo.operacaoID = op.operacaoID
LEFT JOIN Subcontratado so ON eo.subcontratadoID = so.subcontratadoID
WHERE eo.subcontratadoID IS NOT NULL
GROUP BY eo.operacaoID, op.nome;
```

Results		Messages			
	operacaoID	nomeOperacao	nExecucoesSubcontratadas	custoTotalSubcontratado	nSubcontratados
1	3	Montagem	1	75.00	1
2	4	Acabamento	1	75.00	1

Figura 13 - Resultado da consulta [6.1.10]

- Esta *view* apresenta uma análise das operações mais frequentemente subcontratadas, permitindo avaliar o impacto económico da subcontratação no processo produtivo. Inclui o nome da operação, o número total de execuções realizadas por subcontratados, o custo total associado a essas subcontratações — calculado com base na quantidade enviada e no custo do serviço — e o número de subcontratados diferentes envolvidos por operação. É uma ferramenta útil para a tomada de decisões estratégicas sobre outsourcing, o controlo de custos e a avaliação da dependência externa nas operações produtivas.

### 6.1.11 Determinar o custo tot. de prod. de uma ordem

```
-- Calcular o custo total de cada ordem considerando: Operações internas (baseado em
tempo estimado) Operações subcontratadas (preço acordado)
CREATE VIEW [dbo].[vw_CustoTotalPorOrdemFabricao] AS
WITH CustosInternos AS (
    SELECT
        ofa.ordemID,
        SUM(eo.quantidadeEnviada * ISNULL(dft.duracaoUnitariaMin, 0) *
ISNULL(dft.custoMinuto, 0)) AS custoInterno
    FROM OrdemFabricao ofa
    JOIN ItemOrdemFabricao io ON io.ordemID = ofa.ordemID
    JOIN ExecucacaoOperacao eo ON eo.itemOrdemID = io.itemOrdemID
    JOIN DetalheFichaTecnica dft
        ON io.fichaTecnicaID = dft.fichaTecnicaID
        AND eo.operacaoID = dft.operacaoID
    WHERE eo.subcontratadoID IS NULL
    GROUP BY ofa.ordemID
),
CustosSubcontratados AS (
    SELECT
        ofa.ordemID,
        SUM(eo.quantidadeEnviada * ISNULL(s.custoServico, 0)) AS custoSubcontratado
    FROM OrdemFabricao ofa
    JOIN ItemOrdemFabricao io ON io.ordemID = ofa.ordemID
    JOIN ExecucacaoOperacao eo ON eo.itemOrdemID = io.itemOrdemID
    JOIN Subcontratado s ON eo.subcontratadoID = s.subcontratadoID
    GROUP BY ofa.ordemID
),
CustosMateriais AS (
    SELECT
        ofa.ordemID,
        SUM(cmr.quantidadeUtilizada * ISNULL(m.custoUnitario, 0)) AS custoMaterial
    FROM OrdemFabricao ofa
    JOIN ItemOrdemFabricao io ON io.ordemID = ofa.ordemID
    JOIN ExecucacaoOperacao eo ON eo.itemOrdemID = io.itemOrdemID
    JOIN ConsumoMaterialReal cmr ON cmr.execucacaoID = eo.execucacaoID
    JOIN Material m ON cmr.materialID = m.materialID
```

```

GROUP BY ofa.ordemID
)

SELECT
    ofa.ordemID,
    ofa.dataEmissao,
    ofa.dataPrevistaConclusao,
    ofa.estado,
    ISNULL(ci.custoInterno, 0) AS custoInternoTotal,
    ISNULL(cs.custoSubcontratado, 0) AS custoSubcontratadoTotal,
    ISNULL(cm.custoMaterial, 0) AS custoMaterialTotal,
    ISNULL(ci.custoInterno, 0) + ISNULL(cs.custoSubcontratado, 0) +
    ISNULL(cm.custoMaterial, 0) AS custoTotal
FROM OrdemFabrico ofa
LEFT JOIN CustosInternos ci ON ci.ordemID = ofa.ordemID
LEFT JOIN CustosSubcontratados cs ON cs.ordemID = ofa.ordemID
LEFT JOIN CustosMateriais cm ON cm.ordemID = ofa.ordemID;

```

	ordemID	dataEmissao	dataPrevistaConclusao	estado	custoInternoTotal	custoSubcontratadoTotal	custoMaterialTotal	custoTotal
1	1	2025-05-10	2025-05-15	em_execucao	175.0000	150.00	351.0500	676.05
2	2	2025-05-12	2025-05-18	planeada	0.0000	0.00	0.0000	0.00
3	3	2025-05-15	2025-05-20	planeada	0.0000	0.00	0.0000	0.00
4	4	2025-05-23	2025-05-30	planeada	0.0000	0.00	0.0000	0.00
5	5	2025-05-24	2025-06-01	planeada	0.0000	0.00	0.0000	0.00
6	6	2025-05-18	2025-05-22	planeada	0.0000	0.00	0.0000	0.00
7	7	2025-05-20	2025-05-25	planeada	0.0000	0.00	0.0000	0.00
8	8	2025-05-11	2025-05-16	planeada	0.0000	0.00	0.0000	0.00
9	9	2025-05-22	2025-05-29	planeada	0.0000	0.00	0.0000	0.00
10	10	2025-05-13	2025-05-19	planeada	0.0000	0.00	0.0000	0.00
11	11	2025-05-01	2025-05-07	planeada	0.0000	0.00	0.0000	0.00
12	12	2025-05-05	2025-05-11	planeada	0.0000	0.00	0.0000	0.00
13	13	2025-05-08	2025-05-14	planeada	0.0000	0.00	0.0000	0.00
14	14	2025-05-25	2025-06-05	planeada	0.0000	0.00	0.0000	0.00
15	15	2025-05-28	2025-06-08	planeada	0.0000	0.00	0.0000	0.00
16	16	2025-05-19	2025-05-21	planeada	0.0000	0.00	0.0000	0.00
17	17	2025-05-22	2025-05-27	planeada	0.0000	0.00	0.0000	0.00
18	18	2025-05-03	2025-05-09	planeada	0.0000	0.00	0.0000	0.00
19	19	2025-05-26	2025-06-02	planeada	0.0000	0.00	0.0000	0.00

Figura 14 - Resultado da consulta [6.1.11]

- Esta *view* calcula o custo total de cada ordem de fabrico, agregando diferentes componentes de despesa. São considerados os custos internos, calculados com base na quantidade enviada, na duração estimada das operações e no custo por minuto; os custos subcontratados, definidos pelo valor do serviço acordado por operação; e os custos de materiais, apurados segundo a quantidade efetivamente utilizada e o respetivo custo unitário. A *view* apresenta a identificação e as datas da ordem, o estado atual, o detalhe dos custos por tipo (interno, subcontratado e material), bem como o custo total acumulado da ordem. Trata-se de uma ferramenta essencial para a análise de rentabilidade, controlo de custos e planeamento estratégico da produção.

## 6.1.12 Consultar a lista de operações pendentes para cada ordem

```
-- Consultar a lista de operações pendentes para cada ordem, ordenadas por prioridade.
CREATE VIEW vw_OperacoesPendentesPorPrioridade AS
SELECT
    ofa.ordemID,
    io.itemOrdemID,
    p.nome AS nomeProduto,
    CONCAT(p.modelo, ' ', p.cor, ' ', ISNULL(p.variante, '')) AS produtoDetalhado,
    o.operacaoID,
    o.nome AS nomeOperacao,
    dft.ordemExecucao AS prioridade,
    io.quantidadePlaneada,
    ISNULL(eoTotal.totalRecebido, 0) AS quantidadeRecebida,
    CASE
        WHEN ISNULL(eoTotal.totalRecebido, 0) = 0 THEN 'pendente'
        WHEN ISNULL(eoTotal.totalRecebido, 0) < io.quantidadePlaneada THEN 'parcial'
        ELSE 'concluída'
    END AS estadoOperacao
FROM OrdemFabrico ofa
JOIN ItemOrdemFabrico io ON io.ordemID = ofa.ordemID
JOIN Produto p ON io.produtoID = p.produtoID
JOIN DetalheFichaTecnica dft ON io.fichaTecnicaID = dft.fichaTecnicaID
JOIN Operacao o ON dft.operacaoID = o.operacaoID
OUTER APPLY (
    SELECT SUM(eo.quantidadeRecebida) AS totalRecebido
    FROM ExecucaoOperacao eo
    WHERE eo.itemOrdemID = io.itemOrdemID
    AND eo.operacaoID = dft.operacaoID
) eoTotal
WHERE ISNULL(eoTotal.totalRecebido, 0) < io.quantidadePlaneada;
```

	ordemID	itemOrdemID	nomeProduto	produtoDetalhado	operacaoID	nomeOperacao	prioridade	quantidadePlaneada	quantidadeRecebida	estadoOperacao
1	1	1	Bota de Couro Cano Alto Preta	Bota Preta Cano Alto	1	Corte	1	100	50	parcial
2	1	1	Bota de Couro Cano Alto Preta	Bota Preta Cano Alto	2	Costura	2	100	50	parcial
3	1	1	Bota de Couro Cano Alto Preta	Bota Preta Cano Alto	3	Montagem	3	100	50	parcial
4	1	1	Bota de Couro Cano Alto Preta	Bota Preta Cano Alto	4	Acabamento	4	100	40	parcial
5	1	2	Bota de Couro Cano Curto Castanha	Bota Castanha Cano Curto	1	Corte	1	50	0	pendente
6	1	2	Bota de Couro Cano Curto Castanha	Bota Castanha Cano Curto	2	Costura	2	50	0	pendente
7	1	2	Bota de Couro Cano Curto Castanha	Bota Castanha Cano Curto	3	Montagem	3	50	0	pendente
8	1	2	Bota de Couro Cano Curto Castanha	Bota Castanha Cano Curto	4	Acabamento	4	50	0	pendente
9	2	3	Sandália de Couro Verão	Sandália Castanha Verão	1	Corte	1	200	0	pendente
10	2	3	Sandália de Couro Verão	Sandália Castanha Verão	2	Costura	2	200	0	pendente
11	2	3	Sandália de Couro Verão	Sandália Castanha Verão	3	Montagem	3	200	0	pendente
12	2	3	Sandália de Couro Verão	Sandália Castanha Verão	4	Acabamento	4	200	0	pendente
13	3	4	Ténis Casual Urbanos	Ténis Preto/Branco Casual	1	Corte	1	150	0	pendente
14	3	4	Ténis Casual Urbanos	Ténis Preto/Branco Casual	2	Costura	2	150	0	pendente
15	3	4	Ténis Casual Urbanos	Ténis Preto/Branco Casual	3	Montagem	3	150	0	pendente
16	3	4	Ténis Casual Urbanos	Ténis Preto/Branco Casual	4	Acabamento	4	150	0	pendente
17	4	5	Sapato Social Clássico Preto	Sapatos Preto Social	1	Corte	1	75	0	pendente
18	4	5	Sapato Social Clássico Preto	Sapatos Preto Social	2	Costura	2	75	0	pendente
19	4	5	Sapato Social Clássico Preto	Sapatos Preto Social	3	Montagem	3	75	0	pendente

Figura 15 - Resultado da consulta [6.1.12]

- Esta view lista todas as operações de produção que ainda se encontram pendentes ou incompletas, agrupando-as por ordem de fabrico e ordenando-as de acordo com a sua prioridade na sequência de execução, indicada pelo campo *ordemExecucao*. Inclui informações sobre a ordem e o produto associado, a operação e a sua respetiva prioridade, a quantidade planeada e a quantidade já recebida, bem como o estado atual da operação — podendo ser pendente, parcial ou concluída. É uma ferramenta fundamental para o controlo

da produção em tempo real, permitindo identificar atrasos, monitorizar o progresso das operações e planear com precisão os próximos passos da linha de produção.

### 6.1.13 Estimar o tempo tot. para concluir uma ordem (tempo médio das operações)

```
CREATE VIEW vw_EstimativaTempoConclusao AS
SELECT
    o.ordemID,
    o.dataEmissao,
    o.dataPrevistaConclusao,
    SUM(ROUND(
        ISNULL(dft.duracaoUnitariaMin, 0) * io.quantidadePlaneada, 2
    )) AS tempoEstimadoMinutos
FROM OrdemFabrico o
JOIN ItemOrdemFabrico io ON o.ordemID = io.ordemID
JOIN DetalheFichaTecnica dft ON io.fichaTecnicaID = dft.fichaTecnicaID
GROUP BY o.ordemID, o.dataEmissao, o.dataPrevistaConclusao;
```

	ordemID	dataEmissao	dataPrevistaConclusao	tempoEstimadoMinutos
1	1	2025-05-10	2025-05-15	9900.00
2	2	2025-05-12	2025-05-18	7600.00
3	3	2025-05-15	2025-05-20	8700.00
4	4	2025-05-23	2025-05-30	6750.00
5	5	2025-05-24	2025-06-01	8400.00
6	6	2025-05-18	2025-05-22	8000.00
7	7	2025-05-20	2025-05-25	4350.00
8	8	2025-05-11	2025-05-16	4560.00
9	9	2025-05-22	2025-05-29	5220.00
10	10	2025-05-13	2025-05-19	3600.00
11	11	2025-05-01	2025-05-07	5600.00
12	12	2025-05-05	2025-05-11	2400.00
13	13	2025-05-08	2025-05-14	7200.00
14	14	2025-05-25	2025-06-05	3850.00
15	15	2025-05-28	2025-06-08	3300.00
16	16	2025-05-19	2025-05-21	6250.00
17	17	2025-05-22	2025-05-27	4080.00
18	18	2025-05-03	2025-05-09	2550.00

Figura 16 - Resultado da consulta [6.1.13]

- Esta view calcula a estimativa total de tempo, em minutos, necessária para concluir cada ordem de fabrico, tendo como base a duração unitária prevista das operações e a quantidade planeada de produção. Apresenta a ordem de fabrico juntamente com as suas datas de emissão e previsão de conclusão, bem como o tempo total estimado para a execução de todas as operações associadas. Esta informação é particularmente útil para o planeamento da

capacidade produtiva, a gestão de prazos e a análise da viabilidade do cronograma de produção.

#### 6.1.14 Calc. a eficiência média de produção (materiais aproveitados, por período)

```
CREATE VIEW vw_EficienciaMaterial_DiaSemanaMes AS
WITH Base AS (
    SELECT
        cmr.materialID,
        m.nome AS nomeMaterial,
        CAST(cmr.dataRegistro AS DATE) AS dataDia,
        SUM(cmr.quantidadePlaneada) AS totalPlaneado,
        SUM(cmr.quantidadeUtilizada) AS totalUtilizado
    FROM ConsumoMaterialReal cmr
    JOIN Material m ON cmr.materialID = m.materialID
    WHERE cmr.dataRegistro IS NOT NULL
    GROUP BY cmr.materialID, m.nome, CAST(cmr.dataRegistro AS DATE)
)
SELECT
    b.materialID,
    b.nomeMaterial,
    b.dataDia AS periodoDiario,
    CAST(b.totalPlaneado / NULLIF(b.totalUtilizado, 0) AS DECIMAL(5,4)) AS
    eficienciaDiaria,
    -- Semana e mês
    CAST(DATEPART(YEAR, b.dataDia) AS VARCHAR) + '-' +
    RIGHT('0' + CAST(DATEPART(WEEK, b.dataDia) AS VARCHAR), 2) AS semana,
    FORMAT(b.dataDia, 'yyyy-MM') AS mes
FROM Base b;
```

	materialID	nomeMaterial	periodoDiario	eficienciaDiaria	semana	mes
1	1	Pele Genuína	2025-05-23	0.1233	2025-21	2025-05
2	2	Sola de Borracha	2025-05-23	0.2000	2025-21	2025-05
3	3	Atacadores	2025-05-23	NULL	2025-21	2025-05
4	4	Linha de Costura	2025-05-23	0.1477	2025-21	2025-05
5	5	Cola de Sapato	2025-05-23	0.1200	2025-21	2025-05
6	6	Palmilha	2025-05-23	0.0400	2025-21	2025-05
7	7	Tecido Sintético	2025-05-23	0.1333	2025-21	2025-05

Figura 17 - Resultado da consulta [6.1.14]

- Esta view permite analisar a eficiência do uso de materiais ao longo do tempo, com base na razão entre a quantidade planeada e a quantidade realmente utilizada por material. Apresenta o material e a data de registro, a eficiência diária calculada pela relação *planeado ÷ utilizado*, o número da semana do ano e o mês correspondente. Trata-se de uma ferramenta ideal para monitorizar desperdícios, acompanhar a evolução da eficiência no consumo de recursos e apoiar decisões orientadas para a melhoria contínua nos processos produtivos.



### 6.1.15 Geração de alertas, armazenando-o numa tabela específica de controlo

	alertaAtrasoID	ordemID	dataGeracao	motivo
1	1	20	2025-05-23	A ordem ultrapassou o prazo de conclusão
2	2	18	2025-05-23	A ordem ultrapassou o prazo de conclusão
3	3	16	2025-05-23	A ordem ultrapassou o prazo de conclusão
4	4	13	2025-05-23	A ordem ultrapassou o prazo de conclusão
5	5	12	2025-05-23	A ordem ultrapassou o prazo de conclusão
6	6	11	2025-05-23	A ordem ultrapassou o prazo de conclusão
7	7	10	2025-05-23	A ordem ultrapassou o prazo de conclusão
8	8	8	2025-05-23	A ordem ultrapassou o prazo de conclusão
9	9	6	2025-05-23	A ordem ultrapassou o prazo de conclusão
10	10	3	2025-05-23	A ordem ultrapassou o prazo de conclusão
11	11	2	2025-05-23	A ordem ultrapassou o prazo de conclusão
12	12	1	2025-05-23	A ordem ultrapassou o prazo de conclusão

### 6.1.16 Determinar a eficiência média por operação e por período

```
CREATE VIEW vw_EficienciaOperacao_Periodos AS
SELECT
    op.operacaoID,
    op.nome AS nomeOperacao,

    -- Dia (data completa)
    CONVERT(VARCHAR(10), eo.dataExecucao, 120) AS dia,

    -- Semana formatada corretamente: AAAA-SS
    CAST(DATEPART(YEAR, eo.dataExecucao) AS VARCHAR) + '-' +
    RIGHT('0' + CAST(DATEPART(WEEK, eo.dataExecucao) AS VARCHAR), 2) AS semana,

    -- Mês: AAAA-MM
    FORMAT(eo.dataExecucao, 'yyyy-MM') AS mes,

    -- Eficiência em percentagem
    CAST(
        AVG(
            CASE
                WHEN eo.quantidadeEnviada > 0
                THEN 1.0 * eo.quantidadeRecebida / eo.quantidadeEnviada
                ELSE NULL
            END
        ) * 100 AS DECIMAL(5,2)
    ) AS eficienciaPercentual
FROM ExecucaoOperacao eo
JOIN Operacao op ON eo.operacaoID = op.operacaoID
WHERE eo.dataExecucao IS NOT NULL
GROUP BY
    op.operacaoID,
    op.nome,
    eo.dataExecucao;
```

	operacaoID	nomeOperacao	dia	semana	mes	eficienciaPercentual
1	1	Corte	2025-05-23	2025-21	2025-05	95.56
2	2	Costura	2025-05-23	2025-21	2025-05	94.87
3	3	Montagem	2025-05-23	2025-21	2025-05	100.00
4	4	Acabamento	2025-05-23	2025-21	2025-05	93.33

Figura 18 - Resultado da consulta [6.1.16]

- Esta view calcula a eficiência média de cada operação ao longo do tempo, permitindo análises em diferentes escalas — diária, semanal e mensal. Inclui o nome e o ID da **Operacao**, a data de execução (por dia), o número da semana no formato AAAA-SS e o mês no formato AAAA-MM. A eficiência percentual da **Operacao** é calculada com base na razão entre a quantidade recebida e a quantidade enviada, ou vice-versa, conforme o critério adotado. Trata-se de uma ferramenta valiosa para a monitorização contínua da performance operacional, possibilitando a detecção de desvios e a identificação de tendências de melhoria ou degradação ao longo do tempo.

### 6.1.16 Listar ordens atrasadas com o respectivo tempo de atraso

```
-- Identificar automaticamente ordens com atrasos (data prevista de conclusão
ultrapassada).
CREATE VIEW vw_OrdensAtrasadas AS
SELECT
    ordemID,
    dataPrevistaConclusao,
    estado,
    DATEDIFF(DAY, dataPrevistaConclusao, CAST(GETDATE() AS DATE)) AS diasAtraso
FROM OrdemFabrico
WHERE
    estado NOT IN ('concluida', 'cancelada')
    AND dataPrevistaConclusao < CAST(GETDATE() AS DATE);
```

	ordemID	dataPrevistaConclusao	estado	diasAtraso
1	3	2025-05-20	planeada	3
2	6	2025-05-22	planeada	1
3	8	2025-05-16	planeada	7
4	10	2025-05-19	planeada	4
5	11	2025-05-07	planeada	16
6	12	2025-05-11	planeada	12
7	13	2025-05-14	planeada	9
8	16	2025-05-21	planeada	2
9	18	2025-05-09	planeada	14

Figura 19 - Resultado da consulta [6.1.16]

- Esta view identifica automaticamente todas as ordens de fabrico que ultrapassaram a data prevista de conclusão e que ainda não foram concluídas nem canceladas. Apresenta o ID da

ordem de fabrico, a data prevista de conclusão, o estado atual da ordem e o número de dias de atraso em relação à data atual. É uma ferramenta útil para a gestão da produção, o controlo de prazos e o acompanhamento de ordens críticas, permitindo atuar de forma rápida e eficaz sobre atrasos operacionais.

### 6.1.17 Operações problemáticas (com maior índice de perdas)

```
-- Mostrar operações problemáticas (com maior índice de perdas)
CREATE VIEW vw_OperacoesProblematicas AS
SELECT
    eo.operacaoID,
    op.nome AS nomeOperacao,
    COUNT(*) AS totalExecucoes,
    CAST(
        AVG(
            CASE
                WHEN eo.quantidadeEnviada > 0
                THEN 100.0 * (eo.quantidadeEnviada - eo.quantidadeRecebida) /
eo.quantidadeEnviada
                ELSE 0
            END
        ) AS DECIMAL(5,2)
    ) AS mediaPerdas
FROM ExecucacaoOperacao eo
JOIN Operacao op ON eo.operacaoID = op.operacaoID
GROUP BY eo.operacaoID, op.nome
HAVING AVG(
    CASE
        WHEN eo.quantidadeEnviada > 0
        THEN 100.0 * (eo.quantidadeEnviada - eo.quantidadeRecebida) /
eo.quantidadeEnviada
        ELSE 0
    END
) > 0;
```

	operacaoID	nomeOperacao	totalExecucoes	mediaPerdas
1	1	Corte	15	0.89
2	2	Costura	17	0.90
3	4	Acabamento	15	1.33

Figura 20 - Resultado da consulta [6.1.17]

- Esta *view* identifica as operações com maiores índices médios de perda durante a produção. Inclui o *nome* da **Operacao**, o *número total* de **ExecucaoOperacao** registadas e a média percentual de perda de **Produto**, calculada com base na diferença entre a quantidade enviada e a quantidade recebida. Apenas são apresentadas as operações que registaram perdas reais, ou seja, superiores a 0%, permitindo focar naquelas com maior impacto negativo. Trata-se de uma ferramenta útil para a análise da eficiência, a priorização de melhorias nos processos produtivos e o controlo da qualidade.

## 6.1.18 Custos comparativos entre produção interna e subcontratada

```
-- Apresentar custos comparativos entre produção interna e subcontratada
CREATE VIEW vw_CustosComparativos_Interno_Subcontratado AS
SELECT
    dados.tipoProducao,
    SUM(dados.quantidadeRecebida) AS totalProduzido,
    SUM(dados.custo) AS custoTotal,
    ROUND(SUM(dados.custo) / NULLIF(SUM(dados.quantidadeRecebida), 0), 2) AS
custoMedioUnidade,
    ROUND(
        100.0 * SUM(dados.quantidadeRecebida) / NULLIF(
            (SELECT SUM(quantidadeRecebida) FROM ExecuciaoOperacao WHERE
quantidadeRecebida > 0), 0
        ), 2
    ) AS percentagemProducao
FROM (
    SELECT
        quantidadeRecebida,
        custo,
        CASE
            WHEN subcontratadoID IS NULL THEN 'Interna'
            ELSE 'Subcontratada'
        END AS tipoProducao
    FROM ExecuciaoOperacao
    WHERE quantidadeRecebida > 0
) AS dados
GROUP BY dados.tipoProducao;
```

	tipoProducao	totalProduzido	custoTotal	custoMedioUnidade	percentagemProducao
1	Interna	740	987.00	1.330000	70.48000000000000
2	Subcontratada	310	210.50	0.680000	29.52000000000000

Figura 21 - Resultado da consulta [6.1.18]

- Esta view apresenta uma comparação entre a produção interna e a produção **Subcontratada**, considerando diversos indicadores. São exibidas a *quantidade* total produzida por *tipo* de produção, o *custo total* associado, o *custo médio* por unidade produzida e a percentagem que cada tipo representa em relação ao total produzido. Esta análise permite compreender o peso e o impacto económico de cada modelo produtivo dentro da fábrica, sendo fundamental para apoiar decisões estratégicas relacionadas com a escolha entre produção interna ou externa.

### 6.1.19 Geração de alertas de qualidade, se a perda for sup. a 15%

	alertaQualidadeID	execucaoID	subcontratadoID	descricao	dataGeracao
1	1	67	1	Perda > 15%. Enviado: 50, Recebido: 40	2025-05-23
2	2	57	NULL	Perda > 15%. Enviado: 130, Recebido: 110	2025-05-23

Figura 22 - Resultado da consulta [6.1.19]

## 6.2 Procedures

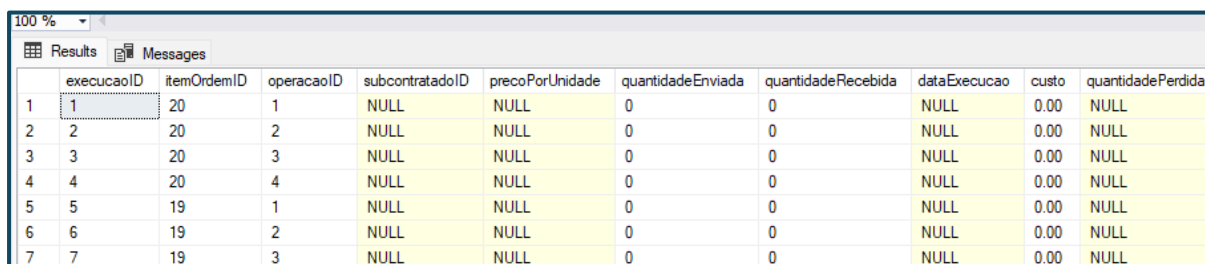
### 6.2.1 Criar reg. de execução para as operações e ordens da ficha técnica

```
-- Criar registos execucao por item
ALTER PROCEDURE sp_CriarRegistosExecucaoPorItem
    @itemOrdemID INT
AS
BEGIN
    SET NOCOUNT ON

    DECLARE @fichaTecnicaID INT;

    -- Aqui obtemos a ficha técnica associada à ordem
    SELECT @fichaTecnicaID = fichaTecnicaID
    FROM ItemOrdemFabricao
    WHERE itemOrdemID = @itemOrdemID;

    -- Inserimos um registo de execução vazio para cada operação de ficha técnica
    INSERT INTO ExecucaoOperacao ( itemOrdemID, operacaoID, subcontratadoID,
    quantidadeEnviada,quantidadeRecebida, dataExecucao, custo)
    SELECT
        @itemOrdemID,
        d.operacaoID,
        NULL,
        0,
        0,
        NULL, -- 0 presente nao vem antes do passado.
        0.00
    FROM DetalheFichaTecnica d
    WHERE d.fichaTecnicaID = @fichaTecnicaID;
END;
```



	execucaoID	itemOrdemID	operacaoID	subcontratadoID	precoPorUnidade	quantidadeEnviada	quantidadeRecebida	dataExecucao	custo	quantidadePerdida
1	1	20	1	NULL	NULL	0	0	NULL	0.00	NULL
2	2	20	2	NULL	NULL	0	0	NULL	0.00	NULL
3	3	20	3	NULL	NULL	0	0	NULL	0.00	NULL
4	4	20	4	NULL	NULL	0	0	NULL	0.00	NULL
5	5	19	1	NULL	NULL	0	0	NULL	0.00	NULL
6	6	19	2	NULL	NULL	0	0	NULL	0.00	NULL
7	7	19	3	NULL	NULL	0	0	NULL	0.00	NULL

Figura 23 - Resultado do procedimento [6.2.1]

- Este *procedure* é responsável por inicializar os registos de **ExecucaoOperacao** para um item de ordem de fabrico específico, com base na respetiva ficha técnica. Para isso, obtém a ficha técnica associada ao @itemOrdemID e cria automaticamente, para cada operação nela definida, um registo correspondente na tabela **ExecucaoOperacao**. Esses registos são gerados com as quantidades e o custo a zero, e com o estado inicial sem execução, ou seja, com *dataExecucao* = NULL. Este procedimento é essencial para garantir que cada item de produção tem as operações devidamente preparadas para registo, facilitando o acompanhamento da produção desde o seu início.

```

-- Criar Execuções na ItemOrdemFabricao
ALTER TRIGGER tr_ItemOrdemFabricao_CriarExecucoes
ON ItemOrdemFabricao
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @itemOrdemID INT;

    DECLARE cur CURSOR FOR
        SELECT itemOrdemID FROM inserted;

    OPEN cur;

    FETCH NEXT FROM cur INTO @itemOrdemID;

    WHILE @@FETCH_STATUS = 0
    BEGIN
        EXEC sp_CriarRegistosExecucaoPorItem @itemOrdemID;
        FETCH NEXT FROM cur INTO @itemOrdemID;
    END;

    CLOSE cur;
    DEALLOCATE cur;
END;

```

	execucaoID	itemOrdemID	operacaoID	subcontratadoID	precoPorUnidade	quantidadeEnviada	quantidadeRecebida	dataExecucao	custo	quantidadePerdida
49	49	5	2	NULL	NULL	0	0	NULL	0.00	NULL
50	50	5	3	NULL	NULL	0	0	NULL	0.00	NULL
51	51	5	4	NULL	NULL	0	0	NULL	0.00	NULL
52	52	4	1	NULL	NULL	0	0	NULL	0.00	NULL
53	53	4	2	NULL	NULL	0	0	NULL	0.00	NULL
54	54	4	3	NULL	NULL	0	0	NULL	0.00	NULL
55	55	4	4	NULL	NULL	0	0	NULL	0.00	NULL
56	56	3	1	NULL	NULL	150	130	2025-05-23	120...	20
57	57	3	2	NULL	NULL	130	110	2025-05-23	156...	20

Figura 24 - Resultado do trigger [6.2.1]

- Este *trigger* é executado automaticamente após a inserção de um novo **ItemOrdemFabricao**. Para cada novo *itemOrdemID* inserido, ele chama o procedimento *sp\_CriarRegistosExecucaoPorItem*, o qual cria todos os registos de execução correspondentes às operações definidas na ficha técnica do produto. Com isso, garante-se que, ao adicionar um item a uma ordem de fabrico, o sistema gere de forma imediata os registos de execução necessários, preparando o ciclo de produção sem necessidade de intervenção manual.

## 7. Conclusão

A realização deste projeto de criação e gestão da base de dados para a produção numa fábrica de calçado revelou-se um processo essencial e eficaz para estruturar e automatizar a gestão das operações industriais. A solução desenvolvida permite acompanhar com precisão cada etapa do processo produtivo, desde a definição da ficha técnica até à execução das operações, registo de consumo de materiais, controlo de perdas e conclusão das ordens de fabrico.

A construção do modelo conceptual, lógico e físico da base de dados permitiu organizar de forma coerente todas as entidades relevantes, como produtos, ordens de fabrico, operações, subcontratados, materiais e alertas de qualidade. A utilização de comandos DDL possibilitou a criação de tabelas e respetivas restrições de integridade, enquanto os comandos DML facilitaram a inserção e manipulação de dados reais, simulando cenários de produção industrial.

Foram desenvolvidas *triggers* que automatizam ações cruciais do processo, como o cálculo de custos, a deteção de perdas acima de 15%, o registo do consumo real de materiais e a atualização automática dos estados de produção dos itens e das ordens. Estas automações eliminaram grande parte da necessidade de intervenção manual, reduzindo erros e promovendo maior consistência e eficiência no registo dos dados.

As *views* e consultas SQL criadas permitiram extrair relatórios completos sobre a produção, eficiência por operação, identificação de etapas problemáticas, progresso das ordens, materiais utilizados e subcontratados envolvidos. Estes relatórios proporcionam à gestão uma visão clara e fundamentada do estado atual da produção, apoiando a tomada de decisões operacionais e estratégicas.

O projeto foi desenvolvido com base numa metodologia iterativa, permitindo testar, validar e refinar continuamente cada componente da base de dados. Esta abordagem garantiu a robustez, escalabilidade e adaptabilidade da solução final.

Em suma, o projeto atingiu todos os objetivos propostos, resultando numa base de dados sólida, funcional e capaz de acompanhar a dinâmica e as necessidades da produção numa fábrica de calçado. A solução implementada não só facilita a gestão diária do processo produtivo como também constitui uma base fiável para futuras melhorias e integrações.



## 8. Anexos

### 8.1 CREATE TABLE

#### 8.1.1 Produto

```
-- Tabela Produto
CREATE TABLE Produto (

    produtoID INT IDENTITY(1,1) NOT NULL,
    modelo VARCHAR(50) NOT NULL,
    variante VARCHAR(50),
    cor VARCHAR(50) NOT NULL,
    nome VARCHAR(50) NOT NULL,
    descricao VARCHAR(255),
    dataCriacao DATETIME2 DEFAULT SYSDATETIME(),
    dataAtualizacao DATETIME2,

    -- Restrição de unicidade no modelo e a variante do Produto
    CONSTRAINT UQ_Produto_modelo_variante
    UNIQUE NONCLUSTERED (modelo, variante),

    -- Chave primária
    CONSTRAINT PK_Produto PRIMARY KEY CLUSTERED (produtoID)
);
```

#### 8.1.2 Operacao

```
-- Tabela Operacao
CREATE TABLE Operacao (

    operacaoID INT IDENTITY(1,1) NOT NULL,
    nome VARCHAR(20) NOT NULL,
    descricao VARCHAR(255),

    -- Restrição de nome válidos
    CONSTRAINT CHK_Operacao_Nome_Valido
    CHECK (nome IN ('Acabamento', 'Montagem', 'Costura', 'Corte')),

    -- Chave primária
    CONSTRAINT PK_Operacao PRIMARY KEY CLUSTERED (operacaoID)
);
```

### 8.1.3 Material

```
-- Tabela Material
CREATE TABLE Material (

    materialID INT IDENTITY(1,1) NOT NULL,
    nome VARCHAR(50) NOT NULL,
    tipo VARCHAR(50) NULL,
    unidadeMedida VARCHAR(20) NULL,
    custoUnitario DECIMAL(10,2) NOT NULL,

    -- Restrição de unicidade no nome do Material
    CONSTRAINT UQ_Material_Nome
    UNIQUE NONCLUSTERED (nome),

    -- Chave primária
    CONSTRAINT PK_Material PRIMARY KEY CLUSTERED (materialID),
)
```

### 8.1.4 Subcontratado

```
-- Tabela Subcontratado
CREATE TABLE Subcontratado (

    subcontratadoID INT IDENTITY(1,1) NOT NULL,
    nome VARCHAR(100) NOT NULL,
    morada VARCHAR(255),
    contacto VARCHAR(50),
    custoServico DECIMAL(10,2) NOT NULL,

    -- Restrição de unicidade no contacto do Subcontratado
    CONSTRAINT UQ_Subcontratado_contacto
    UNIQUE NONCLUSTERED (contacto),

    -- Chave primária
    CONSTRAINT PK_Subcontratado
    PRIMARY KEY CLUSTERED (subcontratadoID)
);
```

## 8.1.5 FichaTecnica

```
-- Tabela FichaTecnica
CREATE TABLE FichaTecnica (

    fichaTecnicaID INT IDENTITY(1,1) NOT NULL,
    produtoID INT NOT NULL,
    origem VARCHAR(50) NOT NULL DEFAULT 'interna',
    descricao VARCHAR(255),
    dataCriacao DATETIME2 DEFAULT SYSDATETIME(),
    dataAtualizacao DATETIME2,
    estado varchar(50) NOT NULL,

    -- Restrição de estado válidos
    CONSTRAINT CHK_FichaTecnica_estado_valido
    CHECK (estado IN ('inativa', 'ativa')),

    -- Restrição de origem válidos
    CONSTRAINT CHK_FichaTecnica_origem_valida
    CHECK (origem IN ('interna', 'externa')),

    -- Chave Estrangeira que refrencia a tabela Produto
    CONSTRAINT FK_FichaTecnica_Produto
    FOREIGN KEY (produtoID) REFERENCES Produto(produtoID),

    -- Chave primária
    CONSTRAINT PK_FichaTecnica PRIMARY KEY CLUSTERED (fichaTecnicaID)
);
```

### 8.1.6 DetalheFichaTecnica

```
-- Tabela DetalheFichaTecnica
CREATE TABLE DetalheFichaTecnica (

    detalheID INT IDENTITY(1,1) NOT NULL,
    fichaTecnicaID INT NOT NULL,
    operacaoID INT NOT NULL,
    materialID INT,
    ordemExecucao INT NOT NULL,
    vezesPorUnidade DECIMAL(10,2) NOT NULL,
    descricao VARCHAR(255),
    duracaoUnitariaMin decimal(5,2),
    custoMinuto decimal(10,2),

    -- Restrição de unicidade no ficha tecnica e na ordem de execução de
    DetalheFichaTecnica
    CONSTRAINT UQ_DetalheFichaTecnica_FichaTecnica_OrdemExecucao
    UNIQUE NONCLUSTERED (fichaTecnicaID, ordemExecucao),

    -- Chave Estrangeira que refrencia a tabela FichaTecnica
    CONSTRAINT FK_DetalheFichaTecnica_FichaTecnica
    FOREIGN KEY (fichaTecnicaID) REFERENCES FichaTecnica(fichaTecnicaID),

    -- Chave Estrangeira que refrencia a tabela Operacao
    CONSTRAINT FK_DetalheFichaTecnica_Operacao
    FOREIGN KEY (operacaoID) REFERENCES Operacao(operacaoID),

    -- Chave primária
    CONSTRAINT PK_DetalheFichaTecnica
    PRIMARY KEY CLUSTERED (detalheID)
);
```

### 8.1.7 OrdemFabricao

```
-- Tabela OrdemFabricao
CREATE TABLE OrdemFabricao (

    ordemID INT IDENTITY(1,1) NOT NULL,
    dataEmissao date DEFAULT SYSDATETIME() NOT NULL,
    dataPrevistaConclusao date NOT NULL,
    estado VARCHAR(20) NOT NULL DEFAULT 'planeada',

    -- Restrição de estado válidos
    CONSTRAINT CHK_OrdemFabricao_estado
    CHECK (estado IN ('cancelada', 'concluida', 'em_execucao', 'planeada')),

    -- Chave primária
    CONSTRAINT PK_OrdemFabricao
    PRIMARY KEY CLUSTERED (ordemID)
);
```

### 8.1.8 ItemOrdemFabricao

```
-- Tabela ItemOrdemFabricao
CREATE TABLE ItemOrdemFabricao (

    itemOrdemID INT IDENTITY(1,1) NOT NULL,
    ordemID INT NOT NULL,
    produtoID INT NOT NULL,
    quantidadePlaneada INT NOT NULL,
    fichaTecnicaID INT NOT NULL,
    estadoProducao VARCHAR(50) DEFAULT 'nao inciado',

    -- Restrição de unicidade no ordem e no produto do ItemOrdemFabricao
    CONSTRAINT UQ_ItemOrdemFabricao_ordemID_produtoID
    UNIQUE NONCLUSTERED (ordemID, produtoID),

    -- Chave Estrangeira que referencia a tabela FichaTecnica
    CONSTRAINT FK_ItemOrdemFabricao_FichaTecnica
    FOREIGN KEY (fichaTecnicaID) REFERENCES FichaTecnica(fichaTecnicaID),

    -- Chave Estrangeira que referencia a tabela OrdemFabricao
    CONSTRAINT FK_ItemOrdemFabricao_OrdemFabricao
    FOREIGN KEY (ordemID) REFERENCES OrdemFabricao(ordemID),

    -- Chave Estrangeira que referencia a tabela Produto
    CONSTRAINT FK_ItemOrdemFabricao_Produto
    FOREIGN KEY (produtoID) REFERENCES Produto(produtoID),

    -- Chave Primária
    CONSTRAINT PK_ItemOrdemFabricao
    PRIMARY KEY CLUSTERED (itemOrdemID),

);
```

### 8.1.9 ExecucacaoOperacao

```
-- Tabela ExecucacaoOperacao
CREATE TABLE ExecucacaoOperacao (

    execucaoID INT IDENTITY(1,1) NOT NULL,
    itemOrdemID INT NOT NULL,
    operacaoID INT NOT NULL,
    subcontratadoID INT,
    precoPorUnidade DECIMAL(10,2),
    quantidadeEnviada INT NOT NULL,
    quantidadeRecebida INT NOT NULL,
    dataExecucao date,
    custo decimal(10,2) NOT NULL,
    quantidadePerdida INT,

    -- Restrição para quantidades válidas
    CONSTRAINT CHK_ExecucacaoOperacao_quantidades_naoNegativa
    CHECK (quantidadeEnviada >= 0 AND quantidadeRecebida >= 0),

    -- Restrição de unicidade no ordem do item e na operação da ExecucacaoOperacao
    CONSTRAINT UQ_ExecucacaoOperacao
    UNIQUE NONCLUSTERED (itemOrdemID, operacaoID),

    -- Chave Estrangeira que referencia a tabela ItemOrdemFabricao
    CONSTRAINT FK_ExecucacaoOperacao_ItemOrdemFabricao
    FOREIGN KEY (itemOrdemID) REFERENCES ItemOrdemFabricao(itemOrdemID),

    -- Chave Estrangeira que referencia a tabela Operacao
    CONSTRAINT FK_ExecucacaoOperacao_Operacao
    FOREIGN KEY (operacaoID) REFERENCES Operacao(operacaoID),

    -- Chave Estrangeira que referencia a tabela Subcontratado
    CONSTRAINT FK_ExecucacaoOperacao_Subcontratado
    FOREIGN KEY (subcontratadoID) REFERENCES Subcontratado(subcontratadoID),

    -- Chave Primária
    CONSTRAINT PK_ExecucacaoOperacao
    PRIMARY KEY CLUSTERED (execucaoID),
);
```

### 8.1.10 AlertaAtraso

```
-- Tabela AlertaAtraso
CREATE TABLE AlertaAtraso (

    alertaAtrasoID INT IDENTITY(1,1) NOT NULL,
    ordemID INT NOT NULL,
    dataGeracao DATE NOT NULL,
    motivo VARCHAR(255),

    -- Chave Estrangeira que referencia a tabela OrdemFabricao
    CONSTRAINT [FK_AlertaAtraso_OrdemFabricao]
    FOREIGN KEY (ordemID) REFERENCES OrdemFabricao(ordemID),

    -- Chave primária
    CONSTRAINT PK_AlertaAtraso
```

```
PRIMARY KEY CLUSTERED (alertaAtrasoID),  
);
```

### 8.1.11 Alerta Qualidade

```
-- Tabela AlertaQualidade  
CREATE TABLE AlertaQualidade (  
  
    alertaQualidadeID INT IDENTITY(1,1) NOT NULL,  
    execucaoID INT NOT NULL,  
    subcontratadoID INT,  
    descricao VARCHAR(255),  
    dataGeracao DATE NOT NULL,  
  
    -- Restrição de unicidade na execução de AlertaQualidade  
    CONSTRAINT UQ_AlertaQualidade_execucao  
    UNIQUE NONCLUSTERED (execucaoID),  
  
    -- Chave Estrangeira que referencia a tabela ExecucaoOperacao  
    CONSTRAINT FK_AlertaQualidade_ExecucaoOperacao  
    FOREIGN KEY (execucaoID) REFERENCES ExecucaoOperacao(execucaoID),  
  
    -- Chave Estrangeira que referencia a tabela Subcontratado  
    CONSTRAINT FK_AlertaQualidade_Subcontratado  
    FOREIGN KEY (subcontratadoID) REFERENCES Subcontratado(subcontratadoID),  
  
    -- Chave Primária  
    CONSTRAINT PK_AlertaQualidade  
    PRIMARY KEY CLUSTERED (alertaQualidadeID),  
);
```

### 8.1.12 ConsumoMaterialReal

```
CREATE TABLE ConsumoMaterialReal (  
  
    consumoID INT IDENTITY(1,1) NOT NULL,  
    execucaoID INT NULL,  
    detalheID INT NOT NULL,  
    materialID INT NOT NULL,  
    quantidadePlaneada DECIMAL(10,2) NOT NULL,  
    quantidadeUtilizada DECIMAL(10,2) NULL,  
    dataRegistro DATETIME2 DEFAULT SYSDATETIME(),  
    percentagemPerdaMaterial DECIMAL(5,2),  
  
    -- Restrição de quantidade utilizada válida  
    CONSTRAINT CK_ConsumoMaterialReal_quantidadeUtilizada_pos  
    CHECK (quantidadeUtilizada IS NULL OR quantidadeUtilizada >= 0)  
  
    -- Restrição de quantidade planeada válida  
    CONSTRAINT CK_ConsumoMaterialReal_quantidadePlaneada_pos  
    CHECK (quantidadePlaneada >= 0),  
  
    -- Chave Estrangeira que referencia a tabela ExecucaoOperacao  
    CONSTRAINT FK_ConsumoMaterialReal_ExecucaoOperacao  
    FOREIGN KEY (execucaoID) REFERENCES ExecucaoOperacao(execucaoID),  
  
    -- Chave Estrangeira que referencia a tabela Material
```

```

CONSTRAINT FK_ConsumoMaterialReal_Material
FOREIGN KEY (materialID) REFERENCES Material(materialID),

-- Chave Estrangeira que referencia a tabela DetalheFichaTecnica
CONSTRAINT FK_ConsumoMaterialReal_DetalheFichaTecnica
FOREIGN KEY (detalheID) REFERENCES DetalheFichaTecnica(detalheID),

-- Chave Primária
CONSTRAINT PK_ConsumoMaterialReal
PRIMARY KEY CLUSTERED (consumoID),
);

```

## 8.2 CREATE TRIGGERS

### 8.2.1 Produto

```

-- Atualizar data no Produto
CREATE TRIGGER tr_Produto_AtualizarData
ON Produto
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE Produto
    SET dataAtualizacao = SYSDATETIME()
    FROM Produto p
    INNER JOIN inserted i ON p.produtoID = i.produtoID;
END;

```

```

-- Bloquear data criação no Produto
CREATE TRIGGER tr_Produto_BloquearDataCriacao
ON Produto
INSTEAD OF UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN deleted d ON i.produtoID = d.produtoID
        WHERE i.dataCriacao <> d.dataCriacao
    )
    BEGIN
        RAISERROR('A data de criação do produto não pode ser alterada.', 16, 1);
        RETURN;
    END;
    -- Continua o update normalmente (preserva dataCriacao original)
    UPDATE p
    SET
        modelo = i.modelo,
        variante = i.variante,
        cor = i.cor,
        nome = i.nome,
        descricao = i.descricao,
        dataAtualizacao = i.dataAtualizacao,

```



```

        precoVenda = i.precoVenda
    FROM Produto p
    JOIN inserted i ON p.produtoID = i.produtoID;
END;

```

## 8.2.2 Subcontratado

```

-- Bloquear delete no Subcontratado
CREATE TRIGGER tr_Subcontratado_BloquearDelete
ON Subcontratado
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM deleted d
        JOIN ExecucaoOperacao eo ON eo.subcontratadoID = d.subcontratadoID
    )
    BEGIN
        RAISERROR('Não é possível eliminar um subcontratado que já foi usado em
operações.', 16, 1);
        RETURN;
    END;

    DELETE FROM Subcontratado
    WHERE subcontratadoID IN (SELECT subcontratadoID FROM deleted);
END;

```

## 8.2.3 FichaTecnica

```

-- Atualizar data na FichaTecnica
CREATE TRIGGER tr_FichaTecnica_AtualizarData
ON FichaTecnica
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE f
    SET dataAtualizacao = SYSDATETIME()
    FROM FichaTecnica f
    JOIN inserted i ON f.fichaTecnicaID = i.fichaTecnicaID;
END;

-- Bloquear data criação na FichaTecnica
CREATE TRIGGER tr_FichaTecnica_BloquearDataCriacao
ON FichaTecnica
INSTEAD OF UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    -- Verifica se alguém tentou mudar a dataCriacao
    IF EXISTS (

```

```

        SELECT 1
        FROM inserted i
        JOIN deleted d ON i.fichaTecnicaID = d.fichaTecnicaID
        WHERE i.dataCriacao <> d.dataCriacao
    )
    BEGIN
        RAISERROR('Não é permitido alterar a data de criação da ficha técnica.', 16,
1);
        RETURN;
    END

    -- Permite o resto da atualização, mas preservando a dataCriacao original
    UPDATE f
    SET
        f.produtoID = i.produtoID,
        f.descricao = i.descricao,
        f.dataAtualizacao = i.dataAtualizacao,
        f.estado = i.estado
    FROM FichaTecnica f
    JOIN inserted i ON f.fichaTecnicaID = i.fichaTecnicaID;
END;

```

```

-- Bloquear inativacao da FichaTecnica
CREATE TRIGGER tr_FichaTecnica_BloquearInativacao
ON FichaTecnica
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    -- Verifica se o estado foi alterado para 'inativa'
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN deleted d ON i.fichaTecnicaID = d.fichaTecnicaID
        WHERE i.estado = 'inativa' AND d.estado <> 'inativa'
    )
    BEGIN
        -- Verifica se há ordens de fabrico ainda ativas com essa ficha
        IF EXISTS (
            SELECT 1
            FROM ItemOrdemFabrico io
            JOIN inserted i ON io.fichaTecnicaID = i.fichaTecnicaID
            JOIN OrdemFabrico ofa ON io.ordemID = ofa.ordemID
            WHERE ofa.estado <> 'concluida'
        )
        BEGIN
            RAISERROR('Não é permitido inativar uma Ficha Técnica com Ordens de
Fabrico ainda em execução ou planeadas.', 16, 1);
            ROLLBACK;
            RETURN;
        END
    END
END;

```

## 8.2.4 DetalheFichaTecnica

```
-- Bloquear delete se ficha tiver ativa na DetalheFichaTecnica
CREATE TRIGGER tr_DetalheFichaTecnica_BloquearDeleteSeAtiva
ON DetalheFichaTecnica
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM deleted d
        JOIN FichaTecnica f ON d.fichaTecnicaID = f.fichaTecnicaID
        WHERE f.estado = 'ativa'
    )
    BEGIN
        RAISERROR('Não é permitido eliminar operações de uma ficha técnica ativa.',
16, 1);
        RETURN;
    END;

    DELETE FROM DetalheFichaTecnica
    WHERE detalheID IN (SELECT detalheID FROM deleted);
END;
```

```
-- Validar operações por ficha interna ou ficha externa
CREATE TRIGGER tr_DetalheFichaTecnica_ValidarInternaExterna
ON DetalheFichaTecnica
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    -- Validação para INTERNAS
    IF EXISTS (
        SELECT dft.fichaTecnicaID
        FROM DetalheFichaTecnica dft
        JOIN FichaTecnica ft ON ft.fichaTecnicaID = dft.fichaTecnicaID
        JOIN Operacao o ON o.operacaoID = dft.operacaoID
        WHERE ft.origem = 'interna'
        GROUP BY dft.fichaTecnicaID
        HAVING
            COUNT(DISTINCT o.nome) <> 4
            OR
            SUM(CASE WHEN o.nome NOT IN ('Corte', 'Costura', 'Montagem', 'Acabamento')
THEN 1 ELSE 0 END) > 0
            OR
            COUNT(DISTINCT dft.ordemExecucao) <> 4
            OR MIN(dft.ordemExecucao) <> 1
            OR MAX(dft.ordemExecucao) <> 4
    )
    BEGIN
        RAISERROR('Ficha Técnica interna deve conter exatamente 4 operações (Corte,
Costura, Montagem, Acabamento) com ordemExecucao de 1 a 4.', 16, 1);
        ROLLBACK;
        RETURN;
    END
END
```

```

-- Validação para EXTERNAS / PARCIAIS
IF EXISTS (
    SELECT dft.fichaTecnicaID
    FROM DetalheFichaTecnica dft
    JOIN FichaTecnica ft ON ft.fichaTecnicaID = dft.fichaTecnicaID
    WHERE ft.origem IN ('externa', 'parcial')
    GROUP BY dft.fichaTecnicaID
    HAVING MAX(dft.ordemExecucao) > COUNT(*)
)
BEGIN
    RAISERROR('Ficha Técnica externa/parcial: ordemExecucao não pode ser maior que
o número total de operações.', 16, 1);
    ROLLBACK;
    RETURN;
END
END;

```

## 8.2.5 ItemOrdemFabricao

```

-- Criar Execuções na ItemOrdemFabricao
CREATE TRIGGER tr_ItemOrdemFabricao_CriarExecucoes
ON ItemOrdemFabricao
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @itemOrdemID INT;

    DECLARE cur CURSOR FOR
        SELECT itemOrdemID FROM inserted;

    OPEN cur;

    FETCH NEXT FROM cur INTO @itemOrdemID;

    WHILE @@FETCH_STATUS = 0
    BEGIN
        EXEC sp_CriarRegistrosExecucaoPorItem @itemOrdemID;
        FETCH NEXT FROM cur INTO @itemOrdemID;
    END;

    CLOSE cur;
    DEALLOCATE cur;
END;

```

```

-- Validar ficha que estejam ativas na itemOrdemFabricao
CREATE TRIGGER tr_ItemOrdemFabricao_ValidarFichaTecnicaAtiva
ON ItemOrdemFabricao
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;

```

```

IF EXISTS (
    SELECT 1
    FROM inserted i
    JOIN FichaTecnica ft ON i.fichaTecnicaID = ft.fichaTecnicaID
    WHERE ft.estado = 'inativa'
)
BEGIN
    RAISERROR('Ficha Técnica está inativa e não pode ser usada em ordens de
    fabrico.', 16, 1);
    ROLLBACK;
    RETURN;
END
END;

```

## 8.2.6 OrdemFabrico

```

-- Atualiza preco venda consoante o custo da ordem
CREATE TRIGGER tr_OrdemFabrico_AtualizarPrecoVendoQuandoOrdemConcluida
ON OrdemFabrico
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @margemLucro DECIMAL(5,2) = 0.30;

    -- Atualizar precoVenda apenas se houver produção final (última operação)
    UPDATE p
    SET p.precoVenda = ROUND(
        (dados.custoTotal / NULLIF(dados.qtdFinal, 0)) * (1 + @margemLucro),
        2
    )
    FROM Produto p
    JOIN (
        SELECT
            io.produtoID,
            SUM(eo.custo + ISNULL(cmr.quantidadeUtilizada * m.custoUnitario, 0)) AS
custoTotal,
            SUM(
                CASE
                    WHEN dft.ordemExecucao = ultimas.ordemMax THEN
eo.quantidadeRecebida
                    ELSE 0
                END
            ) AS qtdFinal
        FROM inserted i
        JOIN ItemOrdemFabrico io ON io.ordemID = i.ordemID
        JOIN ExecucaoOperacao eo ON eo.itemOrdemID = io.itemOrdemID
        JOIN DetalheFichaTecnica dft
            ON dft.fichaTecnicaID = io.fichaTecnicaID
            AND dft.operacaoID = eo.operacaoID
        LEFT JOIN ConsumoMaterialReal cmr ON cmr.execucaoID = eo.execucaoID
        LEFT JOIN Material m ON m.materialID = cmr.materialID
        JOIN (
            SELECT fichaTecnicaID, MAX(ordemExecucao) AS ordemMax
            FROM DetalheFichaTecnica
            GROUP BY fichaTecnicaID
        ) AS ultimas ON ultimas.fichaTecnicaID = dft.fichaTecnicaID
    )

```

```

        WHERE i.estado = 'concluida'
        GROUP BY io.produtoID
    ) AS dados ON dados.produtoID = p.produtoID
    WHERE dados.qtdFinal > 0;
END;

```

```

-- Gerar alerta atraso na OrdemFabricao
CREATE TRIGGER tr_OrdemFabricao_GerarAlertaAtraso
ON OrdemFabricao
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO AlertaAtraso(ordemID, dataGeracao, motivo)
    SELECT
        i.ordemID,
        SYSDATETIME(),
        'A ordem ultrapassou o prazo de conclusão'
    FROM inserted i

    -- prazo ultrapassado e ainda nao esta concluida a ordem se nao existir faz
    alerta, evitando duplicação de alerta
    WHERE
        i.dataPrevistaConclusao < SYSDATETIME() AND i.estado <> 'concluida'
        AND NOT EXISTS (
            SELECT 1
            FROM AlertaAtraso a
            WHERE a.ordemID = i.ordemID
        );
END;

```

```

-- Validar conclusao na OrdemFabricao
CREATE TRIGGER tr_OrdemFabricao_ValidarConclusao
ON OrdemFabricao
INSTEAD OF UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    -- 1. Impedir alteração da data de emissão
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN OrdemFabricao o ON i.ordemID = o.ordemID
        WHERE i.dataEmissao <> o.dataEmissao
    )
    BEGIN
        RAISERROR('Não é permitido alterar a data de emissão da ordem.', 16, 1);
        RETURN;
    END;

    -- 2. Validar se o novo estado é 'concluida' e se há operações pendentes
    IF EXISTS (
        SELECT 1
        FROM inserted i
        WHERE i.estado = 'concluida'
    )

```

```

BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN ItemOrdemFabricao io ON io.ordemID = i.ordemID
        JOIN DetalheFichaTecnica d ON d.fichaTecnicaID = io.fichaTecnicaID
        WHERE NOT EXISTS (
            SELECT 1
            FROM ExecuciaoOperacao eo
            WHERE eo.itemOrdemID = io.itemOrdemID
            AND eo.operacaoID = d.operacaoID
            AND eo.quantidadeRecebida > 0 -- garantir que foi feita
        )
    )
    BEGIN
        RAISERROR('Não é possível concluir a ordem pois existem operações
pendentes.', 16, 1);
        RETURN;
    END
END;

-- 3. Aplicar o update normal
UPDATE o
SET
    o.dataPrevistaConclusao = i.dataPrevistaConclusao,
    o.estado = i.estado
FROM OrdemFabricao o
JOIN inserted i ON o.ordemID = i.ordemID;
END;

```

## 8.2.7 AlertaAtraso

```

-- Bloquear data Geracao no AlertaAtraso
CREATE TRIGGER tr_AlertaAtraso_BloquearDataGeracao
ON AlertaAtraso
INSTEAD OF UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN deleted d ON i.alertaAtrasoID = d.alertaAtrasoID
        WHERE i.dataGeracao <> d.dataGeracao
    )
    BEGIN
        RAISERROR('A data de geração do alerta não pode ser alterada.', 16, 1);
        RETURN;
    END;

    -- Permitir outras alterações válidas
    UPDATE a
    SET
        ordemID = i.ordemID,
        motivo = i.motivo
    FROM AlertaAtraso a
    JOIN inserted i ON a.alertaAtrasoID = i.alertaAtrasoID;

```

```
END;
```

```
-- Proibir Delete no AlertaAtraso
CREATE TRIGGER tr_AlertaAtraso_ProibirDelete
ON AlertaAtraso
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;

    RAISERROR('Não é permitido eliminar alertas de atraso, por motivos de
preservação de histórico', 16, 1);
END;
```

## 8.2.8 AlertaQualidade

```
-- Bloquear data geracao no AlertaQualidade
CREATE TRIGGER tr_AlertaQualidade_BloquearDataGeracao
ON AlertaQualidade
INSTEAD OF UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN deleted d ON i.alertaQualidadeID = d.alertaQualidadeID
        WHERE i.dataGeracao <> d.dataGeracao
    )
    BEGIN
        RAISERROR('A data de geração do alerta de qualidade não pode ser alterada.',
16, 1);
        RETURN;
    END;

    -- Permite alterações aos restantes campos
    UPDATE a
    SET
        descricao = i.descricao,
        execucaoID = i.execucaoID,
        subcontratadoID = i.subcontratadoID
    FROM AlertaQualidade a
    JOIN inserted i ON a.alertaQualidadeID = i.alertaQualidadeID;
END;
```

```
-- Proibir delete no AlertaQualidade
CREATE TRIGGER tr_AlertaQualidade_ProibirDelete
ON AlertaQualidade
INSTEAD OF DELETE
AS
BEGIN
```



```

SET NOCOUNT ON;

RAISERROR('Não é permitido eliminar alertas de qualidade', 16, 1);
END;

```

## 8.2.9 Execuc aoOperacao

```

-- Atualizar data execu  o na Execuc aoOperacao
CREATE TRIGGER tr_Execuc aoOperacao_AtualizarDataExecuc ao
ON Execuc aoOperacao
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE eo
    SET dataExecuc ao = SYSDATETIME()
    FROM Execuc aoOperacao eo
    JOIN inserted i ON eo.execuc aoID = i.execuc aoID
    WHERE
        eo.dataExecuc ao IS NULL AND
        (
            i.quantidadeRecebida > 0 OR
            (i.quantidadeEnviada > 0 AND i.quantidadeRecebida <= i.quantidadeEnviada)
        );
END;

```

```

-- Bloquear delete na Execuc aoOperacao
CREATE TRIGGER tr_Execuc aoOperacao_BloquearDelete
ON Execuc aoOperacao
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;

    RAISERROR('N o   permitido eliminar execu  es de opera  es.', 16, 1);
END;

```

```

-- Controlo de produ  o na Execuc aoOperacao
CREATE TRIGGER tr_Execuc aoOperacao_ControloProducao
ON Execuc aoOperacao
INSTEAD OF UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    -- BLOQUEAR ALTERA  O MANUAL DOS CAMPOS CALCULADOS
    IF EXISTS (
        SELECT 1 FROM inserted i
        JOIN Execuc aoOperacao eo ON eo.execuc aoID = i.execuc aoID
        WHERE i.custo <> eo.custo OR ISNULL(i.quantidadePerdida, 0) <>
        ISNULL(eo.quantidadePerdida, 0)
    )

```

```

BEGIN
    RAISERROR('Não é permitido alterar manualmente os campos custo ou
quantidadePerdida.', 16, 1);
    RETURN;
END;

-- IMPEDIR RECEBIDO > ENVIADO
IF EXISTS (SELECT 1 FROM inserted WHERE quantidadeRecebida > quantidadeEnviada)
BEGIN
    RAISERROR('Quantidade recebida não pode ser maior que a enviada.', 16, 1);
    RETURN;
END;

-- BLOQUEAR EXECUÇÃO SEM PREÇO EM FICHAS EXTERNAS/PARCIAIS
IF EXISTS (
    SELECT 1
    FROM inserted i
    JOIN ItemOrdemFabrico io ON i.itemOrdemID = io.itemOrdemID
    JOIN FichaTecnica ft ON io.fichaTecnicaID = ft.fichaTecnicaID
    WHERE LOWER(ft.origem) IN ('externa', 'parcial')
    AND i.subcontratadoID IS NULL
    AND (i.precoPorUnidade IS NULL OR i.precoPorUnidade <= 0)
)
BEGIN
    RAISERROR('Fichas externas/parciais requerem precoPorUnidade se não forem
subcontratadas.', 16, 1);
    RETURN;
END;

-- BLOQUEAR SALTOS DE EXECUÇÃO
IF EXISTS (
    SELECT 1 FROM inserted i
    JOIN ExecucaoOperacao eo ON eo.itemOrdemID = i.itemOrdemID AND eo.execucaoID <
i.execucaoID
    WHERE eo.quantidadeRecebida IS NULL OR eo.quantidadeRecebida = 0
)
BEGIN
    RAISERROR('Execuções devem seguir a ordem correta.', 16, 1);
    RETURN;
END;

-- BLOQUEAR ENVIO EXCESSIVO VS OPERAÇÃO ANTERIOR
IF EXISTS (
    SELECT 1
    FROM inserted i
    JOIN ItemOrdemFabrico io ON io.itemOrdemID = i.itemOrdemID
    JOIN DetalheFichaTecnica atual ON atual.fichaTecnicaID = io.fichaTecnicaID AND
atual.operacaoID = i.operacaoID
    JOIN DetalheFichaTecnica anterior ON anterior.fichaTecnicaID =
atual.fichaTecnicaID
    AND anterior.ordemExecucao = (
        SELECT MAX(ordemExecucao)
        FROM DetalheFichaTecnica
        WHERE fichaTecnicaID = atual.fichaTecnicaID AND ordemExecucao <
atual.ordemExecucao
    )
    JOIN ExecucaoOperacao eoAnterior ON eoAnterior.itemOrdemID = io.itemOrdemID
AND eoAnterior.operacaoID = anterior.operacaoID
    WHERE i.quantidadeEnviada + ISNULL((
        SELECT SUM(eo2.quantidadeEnviada)
        FROM ExecucaoOperacao eo2

```

```

        WHERE eo2.itemOrdemID = i.itemOrdemID AND eo2.operacaoID = i.operacaoID
AND eo2.execucaoID <> i.execucaoID
    ), 0) > eoAnterior.quantidadeRecebida
)
BEGIN
    RAISERROR('Quantidade enviada excede a recebida na operação anterior.', 16,
1);
    RETURN;
END;

-- UPDATE PRINCIPAL
UPDATE eo
SET
    eo.quantidadeEnviada = i.quantidadeEnviada,
    eo.quantidadeRecebida = i.quantidadeRecebida,
    eo.subcontratadoID = i.subcontratadoID,
    eo.dataExecucao = i.dataExecucao,
    eo.precoPorUnidade = i.precoPorUnidade
FROM ExecucaoOperacao eo
JOIN inserted i ON eo.execucaoID = i.execucaoID;

-- CALCULAR PERDAS
UPDATE eo
SET eo.quantidadePerdida =
    CASE WHEN i.quantidadeEnviada >= i.quantidadeRecebida THEN i.quantidadeEnviada
- i.quantidadeRecebida ELSE 0 END
FROM ExecucaoOperacao eo
JOIN inserted i ON eo.execucaoID = i.execucaoID;

-- ALERTA DE QUALIDADE (> 15%)
INSERT INTO AlertaQualidade (execucaoID, subcontratadoID, descricao, dataGeracao)
SELECT
    i.execucaoID,
    i.subcontratadoID,
    CONCAT('Perda > 15%. Enviado: ', i.quantidadeEnviada, ', Recebido: ',
i.quantidadeRecebida),
    SYSDATETIME()
FROM inserted i
WHERE i.quantidadeEnviada > 0
AND (1.0 * (i.quantidadeEnviada - i.quantidadeRecebida) /
NULLIF(i.quantidadeEnviada, 0)) > 0.15
AND NOT EXISTS (
    SELECT 1 FROM AlertaQualidade aq WHERE aq.execucaoID = i.execucaoID
);

-- REGISTRAR CONSUMO DE MATERIAL REAL
INSERT INTO ConsumoMaterialReal (
    execucaoID, detalheID, materialID, quantidadePlaneada, quantidadeUtilizada,
dataRegistro
)
SELECT
    i.execucaoID,
    cmr.detalheID,
    cmr.materialID,
    cmr.quantidadePlaneada,
    ROUND(cmr.quantidadePlaneada * i.quantidadeEnviada, 2),
    SYSDATETIME()
FROM inserted i
JOIN ItemOrdemFabrico io ON i.itemOrdemID = io.itemOrdemID
JOIN DetalheFichaTecnica dft ON io.fichaTecnicaID = dft.fichaTecnicaID AND
dft.operacaoID = i.operacaoID

```

```

JOIN ConsumoMaterialReal cmr ON cmr.detalheID = dft.detalheID AND cmr.execucaoID
IS NULL
WHERE i.quantidadeEnviada > 0
AND NOT EXISTS (
    SELECT 1 FROM ConsumoMaterialReal r
    WHERE r.execucaoID = i.execucaoID AND r.materialID = cmr.materialID AND
r.detalheID = cmr.detalheID
);

-- CUSTO INTERNO
UPDATE eo
SET eo.custo = ROUND(i.quantidadeEnviada * ISNULL(dft.duracaoUnitariaMin, 0) *
ISNULL(dft.custoMinuto, 0), 2)
FROM ExecucaoOperacao eo
JOIN inserted i ON eo.execucaoID = i.execucaoID
JOIN ItemOrdemFabrico io ON i.itemOrdemID = io.itemOrdemID
JOIN DetalheFichaTecnica dft ON io.fichaTecnicaID = dft.fichaTecnicaID AND
dft.operacaoID = i.operacaoID;

-- CUSTO EXTERNO
UPDATE eo
SET eo.custo = ROUND(i.quantidadeEnviada * ISNULL(s.custoServico, 0), 2)
FROM ExecucaoOperacao eo
JOIN inserted i ON eo.execucaoID = i.execucaoID
JOIN Subcontratado s ON i.subcontratadoID = s.subcontratadoID
WHERE i.subcontratadoID IS NOT NULL;

-- ATUALIZAR ESTADO DE PRODUÇÃO NO ITEM CONFORME ÚLTIMA OPERAÇÃO EXECUTADA
WITH EstadoProducaoPorItem AS (
    SELECT
        eo.itemOrdemID,
        estadoProducao =
            CASE MAX(CASE op.nome
                WHEN 'Corte' THEN 1
                WHEN 'Costura' THEN 2
                WHEN 'Montagem' THEN 3
                WHEN 'Acabamento' THEN 4
                ELSE 0 END)
            WHEN 1 THEN 'Cortado'
            WHEN 2 THEN 'Costurado'
            WHEN 3 THEN 'Montado'
            WHEN 4 THEN 'Acabado'
            ELSE 'não iniciado'
        END
    FROM ExecucaoOperacao eo
    JOIN inserted i ON i.execucaoID = eo.execucaoID
    JOIN DetalheFichaTecnica dft ON eo.operacaoID = dft.operacaoID AND
dft.fichaTecnicaID = (
        SELECT fichaTecnicaID FROM ItemOrdemFabrico WHERE itemOrdemID = eo.itemOrdemID
    )
    JOIN Operacao op ON op.operacaoID = eo.operacaoID
    WHERE eo.quantidadeRecebida > 0
    GROUP BY eo.itemOrdemID
)

UPDATE iof
SET iof.estadoProducao = ep.estadoProducao
FROM ItemOrdemFabrico iof
JOIN EstadoProducaoPorItem ep ON iof.itemOrdemID = ep.itemOrdemID;
UPDATE o
SET o.estado = 'em_execucao'
FROM OrdemFabrico o

```

```

WHERE o.estado = 'planeada'
AND EXISTS (
    SELECT 1
    FROM inserted i
    JOIN ItemOrdemFabricao io ON io.itemOrdemID = i.itemOrdemID
    WHERE io.ordemID = o.ordemID
    AND i.quantidadeRecebida > 0
);

UPDATE o
SET o.estado = 'concluida'
FROM OrdemFabricao o
WHERE o.estado <> 'concluida'
AND NOT EXISTS (
    SELECT 1
    FROM ItemOrdemFabricao io
    WHERE io.ordemID = o.ordemID
    AND io.estadoProducao <> 'Acabado'
);
END;

```

```

-- Proteger data execução na ExecucaoOperacao
CREATE TRIGGER tr_ExecucaoOperacao_ProtegerDataExecucao
ON ExecucaoOperacao
AFTER UPDATE
AS

BEGIN
    SET NOCOUNT ON;
    -- dataExecucao inicia como NULL antes da primeira execução real
    -- Após o primeiro registo de quantidade recebida, é preenchida automaticamente
    via trigger
    -- Este trigger impede que dataExecucao seja posteriormente alterada para NULL ou
    para uma data posterior,
    -- preservando a integridade do histórico de execução, evitando fraude ou enganos

    -- 1. Impede que dataExecucao seja apagada (tornada NULL) se já existia valor
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN deleted d ON i.execucaoID = d.execucaoID
        WHERE d.dataExecucao IS NOT NULL AND i.dataExecucao IS NULL
    )
    BEGIN
        RAISERROR('Não é permitido remover a data de execução de uma operação já
registada.', 16, 1);
        ROLLBACK;
        RETURN;
    END

    -- 2. Impede que a dataExecucao seja atualizada para uma data anterior à atual
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN deleted d ON i.execucaoID = d.execucaoID
        WHERE
            i.dataExecucao IS NOT NULL
            AND i.dataExecucao < SYSDATETIME()
            AND d.dataExecucao IS NOT NULL -- só valida se for uma atualização real
    )

```

```

        AND i.dataExecucao <> d.dataExecucao
    )
    BEGIN
        RAISERROR('Não é permitido definir uma data de execução anterior ao momento
        atual.', 16, 1);
        ROLLBACK;
        RETURN;
    END
END;

```

## 8.2.10 ConsumoMaterialReal

```

-- Bloquear delete no ConsumoMaterialReal
CREATE TRIGGER tr_ConsumoMaterialReal_BloquearDelete
ON ConsumoMaterialReal
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;

    RAISERROR('Não é permitido eliminar registos de consumo de material.', 16, 1);
END;

```

```

-- Calcular perda material no ConsumoMaterialReal
CREATE TRIGGER tr_ConsumoMaterialReal_CalcularPerda
ON ConsumoMaterialReal
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE cmr
    SET percentagemPerdaMaterial =
        CASE
            WHEN eo.quantidadeEnviada IS NULL OR eo.quantidadeEnviada = 0 THEN NULL
            WHEN cmr.quantidadeUtilizada IS NULL THEN NULL
            ELSE ROUND(
                100.0 * (cmr.quantidadeUtilizada - (cmr.quantidadePlaneada *
                eo.quantidadeEnviada))
                / NULLIF(cmr.quantidadePlaneada * eo.quantidadeEnviada, 0),
                2
            )
        END
    FROM ConsumoMaterialReal cmr
    JOIN inserted i ON i.consumoID = cmr.consumoID
    JOIN ExecucaoOperacao eo ON cmr.execucaoID = eo.execucaoID
    WHERE cmr.quantidadeUtilizada IS NOT NULL;
END;

```

## 8.3 CREATE PROCEDURE

```
-- Criar registos execucao por item
CREATE PROCEDURE sp_CriarRegistosExecucaoPorItem
    @itemOrdemID INT
AS
BEGIN
    SET NOCOUNT ON

    DECLARE @fichaTecnicaID INT;

    -- Aqui obtemos a ficha técnica associada à ordem
    SELECT @fichaTecnicaID = fichaTecnicaID
    FROM ItemOrdemFabricao
    WHERE itemOrdemID = @itemOrdemID;

    -- Inserimos um registo de execução vazio para cada operação de ficha técnica
    INSERT INTO ExecucaoOperacao ( itemOrdemID, operacaoID, subcontratadoID,
quantidadeEnviada,quantidadeRecebida, dataExecucao, custo)
    SELECT
        @itemOrdemID,
        d.operacaoID,
        NULL,
        0,
        0,
        NULL, -- 0 presente nao vem antes do passado.
        0.00
    FROM DetalheFichaTecnica d
    WHERE d.fichaTecnicaID = @fichaTecnicaID;
END;
```

```
-- Registrar execuções
/*
Exemplo de uso:
EXEC dbo.sp_RegistarExecucaoOperacao
    @execucaoID = 7,
    @quantidadeEnviada = 120,
    @quantidadeRecebida = 115,
    @subcontratadoID = NULL (No caso de externa, definir)
    @precoPorUnidade = NULL; (No caso de externo, definir)
*/
CREATE PROCEDURE sp_RegistarExecucaoOperacao
    @execucaoID INT,
    @quantidadeEnviada INT,
    @quantidadeRecebida INT,
    @subcontratadoID INT = NULL,
    @precoPorUnidade DECIMAL(10,2)
AS
BEGIN
    SET NOCOUNT ON;

    -- Validação: existe a execução?
    IF NOT EXISTS (SELECT 1 FROM ExecucaoOperacao WHERE execucaoID = @execucaoID)
    BEGIN
        RAISERROR('Execução não encontrada.', 16, 1);
        RETURN;
    END;
```

```

-- Validação: quantidades
IF @quantidadeEnviada < 0 OR @quantidadeRecebida < 0
BEGIN
    RAISERROR('Quantidade enviada ou recebida não pode ser negativa.', 16, 1);
    RETURN;
END;

-- Validação: subcontratado existe (se fornecido)
IF @subcontratadoID IS NOT NULL AND NOT EXISTS (
    SELECT 1 FROM Subcontratado WHERE subcontratadoID = @subcontratadoID
)
BEGIN
    RAISERROR('Subcontratado inválido.', 16, 1);
    RETURN;
END;

-- Atualização (dataExecucao será definida no trigger ou default)
UPDATE ExecucaoOperacao
SET
    quantidadeEnviada = @quantidadeEnviada,
    quantidadeRecebida = @quantidadeRecebida,
    subcontratadoID = @subcontratadoID,
    precoPorUnidade = @precoPorUnidade
WHERE execucaoID = @execucaoID;

SELECT 'Execução atualizada com sucesso.' AS mensagem;
END;

```

```

-- Cancelar ordem de fabrico
CREATE PROCEDURE sp_CancelarOrdemFabrico
    @ordemID INT
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE OrdemFabrico
    SET estado = 'cancelada'
    WHERE ordemID = @ordemID;

    DELETE FROM ExecucaoOperacao
    WHERE itemOrdemID IN (
        SELECT itemOrdemID FROM ItemOrdemFabrico WHERE ordemID = @ordemID
    );
END;

-- Relatorio mensal
CREATE PROCEDURE sp_RelatorioResumoProducao
AS
BEGIN
    SET NOCOUNT ON;

    -- Resumo geral por mês
    SELECT
        FORMAT(eo.dataExecucao, 'yyyy-MM') AS periodo,
        SUM(eo.quantidadeRecebida) AS totalProduzido,
        SUM(ISNULL(eo.quantidadePerdida, 0)) AS totalPerdido,
        CAST(

```



```

        100.0 * SUM(eo.quantidadeRecebida) / NULLIF(SUM(eo.quantidadeRecebida +
ISNULL(eo.quantidadePerdida, 0)), 0)
        AS DECIMAL(5,2)
    ) AS eficienciaMediaGlobalPercent,

    -- Subqueries inline para contar ordens
    (SELECT COUNT(*) FROM OrdemFabrico WHERE estado = 'concluida') AS
ordensConcluidas,
    (SELECT COUNT(*) FROM OrdemFabrico WHERE estado = 'em_execucao') AS
ordensEmExecucao

FROM ExecucaoOperacao eo
WHERE eo.dataExecucao IS NOT NULL
GROUP BY FORMAT(eo.dataExecucao, 'yyyy-MM')
ORDER BY periodo DESC;
END;

```

## 8.5 CREATE VIEW

```

-- Apresentar custos comparativos entre produção interna e subcontratada
CREATE VIEW vw_CustosComparativos_Interno_Subcontratado AS
SELECT
    dados.tipoProducao,
    SUM(dados.quantidadeRecebida) AS totalProduzido,
    SUM(dados.custo) AS custoTotal,
    ROUND(SUM(dados.custo) / NULLIF(SUM(dados.quantidadeRecebida), 0), 2) AS
custoMedioUnidade,
    ROUND(
        100.0 * SUM(dados.quantidadeRecebida) / NULLIF(
            (SELECT SUM(quantidadeRecebida) FROM ExecucaoOperacao WHERE
quantidadeRecebida > 0), 0
        ), 2
    ) AS percentagemProducao
FROM (
    SELECT
        quantidadeRecebida,
        custo,
        CASE
            WHEN subcontratadoID IS NULL THEN 'Interna'
            ELSE 'Subcontratada'
        END AS tipoProducao
    FROM ExecucaoOperacao
    WHERE quantidadeRecebida > 0
) AS dados
GROUP BY dados.tipoProducao;

```

```

-- Calcular o custo total de cada ordem considerando: Operações internas (baseado em
tempo estimado) Operações subcontratadas (preço acordado)
CREATE VIEW vw_CustoTotalPorOrdemFabrico AS
WITH CustosInternos AS (
    SELECT
        ofa.ordemID,
        SUM(eo.quantidadeEnviada * ISNULL(dft.duracaoUnitariaMin, 0) *
ISNULL(dft.custoMinuto, 0)) AS custoInterno
    FROM OrdemFabrico ofa

```

```

JOIN ItemOrdemFabrico io ON io.ordemID = ofa.ordemID
JOIN ExecuciaoOperacao eo ON eo.itemOrdemID = io.itemOrdemID
JOIN DetalheFichaTecnica dft
    ON io.fichaTecnicaID = dft.fichaTecnicaID
    AND eo.operacaoID = dft.operacaoID
WHERE eo.subcontratadoID IS NULL
GROUP BY ofa.ordemID
),
CustosSubcontratados AS (
    SELECT
        ofa.ordemID,
        SUM(eo.quantidadeEnviada * ISNULL(s.custoServico, 0)) AS custoSubcontratado
    FROM OrdemFabrico ofa
    JOIN ItemOrdemFabrico io ON io.ordemID = ofa.ordemID
    JOIN ExecuciaoOperacao eo ON eo.itemOrdemID = io.itemOrdemID
    JOIN Subcontratado s ON eo.subcontratadoID = s.subcontratadoID
    GROUP BY ofa.ordemID
),
CustosMateriais AS (
    SELECT
        ofa.ordemID,
        SUM(cmr.quantidadeUtilizada * ISNULL(m.custoUnitario, 0)) AS custoMaterial
    FROM OrdemFabrico ofa
    JOIN ItemOrdemFabrico io ON io.ordemID = ofa.ordemID
    JOIN ExecuciaoOperacao eo ON eo.itemOrdemID = io.itemOrdemID
    JOIN ConsumoMaterialReal cmr ON cmr.execucaoID = eo.execucaoID
    JOIN Material m ON cmr.materialID = m.materialID
    GROUP BY ofa.ordemID
)

SELECT
    ofa.ordemID,
    ofa.dataEmissao,
    ofa.dataPrevistaConclusao,
    ofa.estado,
    ISNULL(ci.custoInterno, 0) AS custoInternoTotal,
    ISNULL(cs.custoSubcontratado, 0) AS custoSubcontratadoTotal,
    ISNULL(cm.custoMaterial, 0) AS custoMaterialTotal,
    ISNULL(ci.custoInterno, 0) + ISNULL(cs.custoSubcontratado, 0) +
    ISNULL(cm.custoMaterial, 0) AS custoTotal
FROM OrdemFabrico ofa
LEFT JOIN CustosInternos ci ON ci.ordemID = ofa.ordemID
LEFT JOIN CustosSubcontratados cs ON cs.ordemID = ofa.ordemID
LEFT JOIN CustosMateriais cm ON cm.ordemID = ofa.ordemID;

```

-- Determinar a eficiência média por operação e por período, neste caso a escolha foi mensalmente porque é a que faz mais sentido segundo o negocio

```

CREATE VIEW vw_EficienciaMediaPorOperacaoPeriodoMensal AS
SELECT
    eo.operacaoID,
    op.nome AS nomeOperacao,
    FORMAT(eo.dataExecucao, 'yyyy-MM') AS periodo,
    CAST(SUM(eo.quantidadeRecebida) * 1.0 / NULLIF(SUM(eo.quantidadeEnviada), 0) AS
    DECIMAL(5,4)) AS eficienciaMedia
FROM ExecuciaoOperacao eo
JOIN Operacao op ON eo.operacaoID = op.operacaoID
GROUP BY
    eo.operacaoID,
    op.nome,

```

```
FORMAT(eo.dataExecucao, 'yyyy-MM');
```

```
-- Calcular a eficiência média de produção (percentagem de materiais aproveitados) por
período (dia, semana, mês).
CREATE VIEW vw_EficienciaOperacao_DiaSemanaMes AS
WITH Ef_Diaria AS (
    SELECT
        eo.operacaoID,
        CAST(eo.dataExecucao AS date) AS periodo,
        CAST(SUM(eo.quantidadeRecebida * 1.0) / NULLIF(SUM(eo.quantidadeEnviada), 0)
AS DECIMAL(5,4)) AS eficienciaDiaria
    FROM ExecucaoOperacao eo
    GROUP BY eo.operacaoID, CAST(eo.dataExecucao AS date)
),
Ef_Semanal AS (
    SELECT
        eo.operacaoID,
        FORMAT(eo.dataExecucao, 'yyyy-ww') AS periodo,
        CAST(SUM(eo.quantidadeRecebida * 1.0) / NULLIF(SUM(eo.quantidadeEnviada), 0)
AS DECIMAL(5,4)) AS eficienciaSemanal
    FROM ExecucaoOperacao eo
    GROUP BY eo.operacaoID, FORMAT(eo.dataExecucao, 'yyyy-ww')
),
Ef_Mensal AS (
    SELECT
        eo.operacaoID,
        FORMAT(eo.dataExecucao, 'yyyy-MM') AS periodo,
        CAST(SUM(eo.quantidadeRecebida * 1.0) / NULLIF(SUM(eo.quantidadeEnviada), 0)
AS DECIMAL(5,4)) AS eficienciaMensal
    FROM ExecucaoOperacao eo
    GROUP BY eo.operacaoID, FORMAT(eo.dataExecucao, 'yyyy-MM')
)
SELECT
    d.operacaoID,
    op.nome AS nomeOperacao,
    d.periodo AS periodoDiario,
    d.eficienciaDiaria,
    s.eficienciaSemanal,
    m.eficienciaMensal
FROM Ef_Diaria d
LEFT JOIN Ef_Semanal s
    ON s.operacaoID = d.operacaoID
    AND FORMAT(d.periodo, 'yyyy-ww') = s.periodo
LEFT JOIN Ef_Mensal m
    ON m.operacaoID = d.operacaoID
    AND FORMAT(d.periodo, 'yyyy-MM') = m.periodo
JOIN Operacao op ON op.operacaoID = d.operacaoID;
```

```
-- Lucro por ordem de fabrico
CREATE VIEW vw_LucroPorOrdemFabrico AS
SELECT
    ofa.ordemID,
    ofa.dataEmissao,
    ofa.estado,
    p.modelo,
    p.nome AS nomeProduto,
    p.precoVenda,
```

```

io.quantidadePlaneada,

-- Receita esperada
ROUND(p.precoVenda * io.quantidadePlaneada, 2) AS receitaPrevista,

-- Custo total
ROUND(SUM(
    ISNULL(eo.custo, 0) + ISNULL(cmr.quantidadeUtilizada * m.custoUnitario, 0)
), 2) AS custoTotalReal,

-- Lucro estimado
ROUND(
    (p.precoVenda * io.quantidadePlaneada)
    - SUM(ISNULL(eo.custo, 0) + ISNULL(cmr.quantidadeUtilizada * m.custoUnitario,
0)),
    2
) AS lucroPrevisto

FROM OrdemFabrico ofa
JOIN ItemOrdemFabrico io ON io.ordemID = ofa.ordemID
JOIN Produto p ON p.produtoID = io.produtoID
JOIN ExecuciaoOperacao eo ON eo.itemOrdemID = io.itemOrdemID
LEFT JOIN ConsumoMaterialReal cmr ON cmr.execucaoID = eo.execucaoID
LEFT JOIN Material m ON m.materialID = cmr.materialID

GROUP BY
    ofa.ordemID,
    ofa.dataEmissao,
    ofa.estado,
    p.modelo,
    p.nome,
    p.precoVenda,
    io.quantidadePlaneada;

```

```

-- Lucro da produção
CREATE VIEW vw_LucroPorTipoProducao AS
SELECT
    tipo.tipoProducao,
    COUNT(DISTINCT io.itemOrdemID) AS totalItens,
    SUM(io.quantidadePlaneada * p.precoVenda) AS receitaTotal,

    -- Custo das execuções (interno ou subcontratado)
    SUM(eo.custo) AS custoExecucaoTotal,

    -- Custo de materiais usados
    SUM(ISNULL(cmr.quantidadeUtilizada * m.custoUnitario, 0)) AS custoMaterialTotal,

    -- Custo total
    SUM(eo.custo + ISNULL(cmr.quantidadeUtilizada * m.custoUnitario, 0)) AS
custoTotal,

    -- Lucro
    SUM(io.quantidadePlaneada * p.precoVenda) -
    SUM(eo.custo + ISNULL(cmr.quantidadeUtilizada * m.custoUnitario, 0)) AS lucroTotal

FROM ExecuciaoOperacao eo
JOIN ItemOrdemFabrico io ON eo.itemOrdemID = io.itemOrdemID
JOIN Produto p ON p.produtoID = io.produtoID
LEFT JOIN ConsumoMaterialReal cmr ON cmr.execucaoID = eo.execucaoID

```

```

LEFT JOIN Material m ON m.materialID = cmr.materialID

-- Distingue produção interna vs subcontratada
CROSS APPLY (
    SELECT
        CASE
            WHEN eo.subcontratadoID IS NULL THEN 'Interna'
            ELSE 'Subcontratada'
        END AS tipoProducao
    ) AS tipo
GROUP BY tipo.tipoProducao;

```

```

-- Listar as operações mais frequentemente subcontratadas e os respectivos custos.
CREATE VIEW vw_OperacoesMaisSubcontratadas AS
SELECT
    eo.operacaoID,
    op.nome AS nomeOperacao,
    COUNT(*) AS nExecucoesSubcontratadas,
    SUM(ISNULL(so.custoServico, 0) * eo.quantidadeEnviada) AS custoTotalSubcontratado,
    COUNT(DISTINCT eo.subcontratadoID) AS nSubcontratados
FROM ExecucaoOperacao eo
JOIN Operacao op ON eo.operacaoID = op.operacaoID
LEFT JOIN Subcontratado so ON eo.subcontratadoID = so.subcontratadoID
WHERE eo.subcontratadoID IS NOT NULL
GROUP BY eo.operacaoID, op.nome;

```

```

-- Consultar a lista de operações pendentes para cada ordem, ordenadas por prioridade.
CREATE VIEW vw_OperacoesPendentesPorPrioridade AS
SELECT
    ofa.ordemID,
    io.itemOrdemID,
    p.nome AS nomeProduto,
    CONCAT(p.modelo, ' ', p.cor, ' ', ISNULL(p.variante, '')) AS produtoDetalhado,
    o.operacaoID,
    o.nome AS nomeOperacao,
    dft.ordemExecucao AS prioridade,
    io.quantidadePlaneada,
    ISNULL(eoTotal.totalRecebido, 0) AS quantidadeRecebida,
    CASE
        WHEN ISNULL(eoTotal.totalRecebido, 0) = 0 THEN 'pendente'
        WHEN ISNULL(eoTotal.totalRecebido, 0) < io.quantidadePlaneada THEN 'parcial'
        ELSE 'concluída'
    END AS estadoOperacao
FROM OrdemFabrico ofa
JOIN ItemOrdemFabrico io ON io.ordemID = ofa.ordemID
JOIN Produto p ON io.produtoID = p.produtoID
JOIN DetalheFichaTecnica dft ON io.fichaTecnicaID = dft.fichaTecnicaID
JOIN Operacao o ON dft.operacaoID = o.operacaoID
OUTER APPLY (
    SELECT SUM(eo.quantidadeRecebida) AS totalRecebido
    FROM ExecucaoOperacao eo
    WHERE eo.itemOrdemID = io.itemOrdemID
        AND eo.operacaoID = dft.operacaoID
    ) eoTotal
WHERE ISNULL(eoTotal.totalRecebido, 0) < io.quantidadePlaneada;

```

```

-- Mostrar operações problemáticas (com maior índice de perdas)
CREATE VIEW vw_OperacoesProblematicas AS
SELECT
    eo.operacaoID,
    op.nome AS nomeOperacao,
    COUNT(*) AS totalExecucoes,
    CAST(
        AVG(
            CASE
                WHEN eo.quantidadeEnviada > 0
                THEN 100.0 * (eo.quantidadeEnviada - eo.quantidadeRecebida) /
eo.quantidadeEnviada
            ELSE 0
            END
        ) AS DECIMAL(5,2)
    ) AS mediaPerdas
FROM ExecuciaoOperacao eo
JOIN Operacao op ON eo.operacaoID = op.operacaoID
GROUP BY eo.operacaoID, op.nome
HAVING AVG(
    CASE
        WHEN eo.quantidadeEnviada > 0
        THEN 100.0 * (eo.quantidadeEnviada - eo.quantidadeRecebida) /
eo.quantidadeEnviada
        ELSE 0
        END
    ) > 0;

```

```

-- Identificar automaticamente ordens com atrasos (data prevista de conclusão
ultrapassada).
CREATE VIEW vw_OrdensAtrasadas AS
SELECT
    ordemID,
    dataPrevistaConclusao,
    estado,
    DATEDIFF(DAY, dataPrevistaConclusao, CAST(GETDATE() AS DATE)) AS diasAtraso
FROM OrdemFabrico
WHERE
    estado NOT IN ('concluida', 'cancelada')
    AND dataPrevistaConclusao < CAST(GETDATE() AS DATE);

```

```

-- Consultar as ordens de fabrico em curso, com indicação do progresso (quantidade
produzida vs. quantidade planeada).
CREATE VIEW vw_OrdensFabrico_Progresso AS
SELECT
    ofa.ordemID,
    ofa.dataEmissao,
    ofa.dataPrevistaConclusao,
    io.itemOrdemID,
    p.nome AS nomeProduto,
    CONCAT(p.modelo, ' ', p.cor, ' ', p.variante) AS produtoDetalhado,
    io.quantidadePlaneada,
    ISNULL(SUM(eo.quantidadeRecebida), 0) AS quantidadeProduzida,
    ROUND(

```

```

        100.0 * ISNULL(SUM(eo.quantidadeRecebida), 0) / NULLIF(io.quantidadePlaneada,
0),
        2
    ) AS percentagemProgresso,
    io.estadoProducao
FROM OrdemFabrico ofa
JOIN ItemOrdemFabrico io ON io.ordemID = ofa.ordemID
JOIN Produto p ON io.produtoID = p.produtoID
LEFT JOIN ExecuciaoOperacao eo ON eo.itemOrdemID = io.itemOrdemID
WHERE ofa.estado = 'em_execucao'
GROUP BY
    ofa.ordemID,
    ofa.dataEmissao,
    ofa.dataPrevistaConclusao,
    io.itemOrdemID,
    p.nome,
    p.modelo, p.cor, p.variante,
    io.quantidadePlaneada,
    io.estadoProducao

```

## 8.4 INSERT INTO

### 8.4.1 Produto

```

-- Inserir dados na tabela Produto (focado em calçado)
INSERT INTO Produto (modelo, variante, cor, nome, descricao) VALUES
('Bota', 'Cano Alto', 'Preta', 'Bota de Couro Cano Alto Preta', 'Bota de couro genuíno
preta com cano alto.'),
('Bota', 'Cano Curto', 'Castanha', 'Bota de Couro Cano Curto Castanha', 'Bota de couro
genuíno castanha com cano curto.'),
('Bota', 'Tática', 'Preta', 'Bota Tática Impermeável Preta', 'Bota tática resistente e
impermeável em preto.'),
('Bota', 'Montanha', 'Castanha/Preta', 'Bota de Montanha Aventura', 'Bota robusta para
caminhadas e montanhismo.'),
('Sandália', 'Verão', 'Castanha', 'Sandália de Couro Verão', 'Sandália de couro
confortável para os dias quentes.'),
('Sandália', 'Desportiva', 'Azul/Cinza', 'Sandália Desportiva Ajustável', 'Sandália
desportiva com tiras ajustáveis para maior suporte.'),
('Sandália', 'Plataforma', 'Bege', 'Sandália Plataforma Elegante', 'Sandália de
plataforma bege para ocasiões especiais.'),
('Ténis', 'Corrida', 'Branco/Azul', 'Ténis de Corrida Performance', 'Ténis leves e com
bom amortecimento para corrida.'),
('Ténis', 'Casual', 'Preto/Branco', 'Ténis Casual Urbanos', 'Ténis pretos e brancos
para uso diário com estilo.'),
('Ténis', 'Desportivo', 'Cinzento/Verde', 'Ténis Desportivo Treino', 'Ténis
confortáveis para diversas atividades físicas.'),
('Sapatos', 'Social', 'Preto', 'Sapato Social Clássico Preto', 'Sapato social de couro
preto elegante.'),
('Sapatos', 'Social', 'Castanho', 'Sapato Social Clássico Castanho', 'Sapato social de
couro castanho sofisticado.'),
('Sapatos', 'Mocassim', 'Castanho', 'Mocassim de Couro Confortável', 'Mocassim de
couro macio para um look casual elegante.'),
('Sapatos', 'Oxford', 'Preto', 'Sapato Oxford Formal Preto', 'Sapato Oxford preto
ideal para eventos formais.'),
('Chinelos', 'Piscina', 'Azul', 'Chinelos de Piscina Antiderrapantes', 'Chinelos azuis
antiderrapantes para uso em piscinas e praias.'),

```

```
( 'Chinelos', 'Casa', 'Cinzento', 'Chinelos de Casa Confortáveis', 'Chinelos cinzentos e quentinhos para usar em casa.' ),
( 'Espadrilles', 'Verão', 'Bege', 'Espadrilles de Lona Bege', 'Espadrilles leves de lona bege com sola de corda.' ),
( 'Alpargatas', 'Casual', 'Azul Marinho', 'Alpargatas de Tecido Azul Marinho', 'Alpargatas de tecido azul marinho para um look descontraído.' ),
( 'Sabrinas', 'Clássicas', 'Pretas', 'Sabrinas Clássicas Pretas', 'Sabrinas pretas elegantes e confortáveis.' ),
( 'Sabrinas', 'Coloridas', 'Vermelhas', 'Sabrinas Coloridas Vermelhas', 'Sabrinas vermelhas vibrantes para um toque de cor.' );
```

## 8.4.2 Operacao

```
-- Inserir dados na tabela Operacao (respeitando a restrição de nome)
INSERT INTO Operacao (nome, descricao) VALUES
( 'Corte', 'Operação de corte dos materiais seguindo os moldes.' ),
( 'Costura', 'Operação de costura das diferentes partes do calçado.' ),
( 'Montagem', 'Operação de montagem das solas, palmilhas e outras partes do calçado.' ),
( 'Acabamento', 'Operação final de limpeza, inspeção e embalagem do calçado.' );
```

## 8.4.3 Material

```
-- Inserir dados na tabela Material (com unidades de medida específicas para calçado em português)
INSERT INTO Material (nome, tipo, unidadeMedida, custoUnitario) VALUES
( 'Pele Genuína', 'Couro', 'Pé Quadrado', 2.80 ),
( 'Sola de Borracha', 'Solas', 'Par', 3.20 ),
( 'Atacadores', 'Acessórios', 'Par', 0.50 ),
( 'Linha de Costura', 'Fios', 'Carretel', 1.10 ),
( 'Cola de Sapato', 'Químicos', 'Litro', 7.80 ),
( 'Palmilha', 'Componentes Internos', 'Par', 1.50 ),
( 'Tecido Sintético', 'Tecidos', 'Metro Linear', 1.75 ),
( 'Fivelas Metálicas', 'Acessórios', 'Unidade', 0.35 ),
( 'Fecho Éclair', 'Acessórios', 'Unidade', 0.90 ),
( 'Ilhós', 'Acessórios', 'Milheiro', 2.15 ),
( 'Reforço Lateral', 'Componentes Internos', 'Par', 0.80 ),
( 'Etiquetas', 'Embalagem', 'Unidade', 0.06 ),
( 'Caixas de Sapatos', 'Embalagem', 'Unidade', 0.45 ),
( 'Não Tecido', 'Tecidos', 'Metro Linear', 0.58 ),
( 'Camurça', 'Couro', 'Pé Quadrado', 3.30 ),
( 'Verniz', 'Couro', 'Pé Quadrado', 3.10 ),
( 'Elástico', 'Acessórios', 'Metro Linear', 0.25 ),
( 'Velcro', 'Acessórios', 'Metro Linear', 0.65 ),
( 'TNT (Tecido Não Tecido)', 'Tecidos', 'Metro Linear', 0.36 ),
( 'Pelúcia Sintética', 'Tecidos', 'Metro Linear', 2.50 ),
( 'Pele Ecológica', 'Couro Sintético', 'Pé Quadrado', 1.75 ),
( 'Sola de Poliuretano (PU)', 'Solas', 'Par', 4.10 ),
( 'Reforço de Biqueira', 'Componentes Internos', 'Par', 0.55 ),
( 'Contraforte', 'Componentes Internos', 'Par', 0.70 ),
( 'Cola de Poliuretano', 'Químicos', 'Litro', 9.50 ),
( 'Tachas', 'Acessórios', 'Milheiro', 13.00 ),
( 'Membrana Impermeável', 'Tecidos', 'Metro Linear', 3.70 ),
( 'Forro Têxtil', 'Componentes Internos', 'Metro Linear', 0.65 ),
( 'Palmilha Ortopédica', 'Componentes Internos', 'Par', 2.80 ),
```



```
( 'Fio de Nylon', 'Fios', 'Carretel', 1.50),
( 'Spray Impermeabilizante', 'Químicos', 'Unidade', 5.20),
( 'Papel de Embrulho', 'Embalagem', 'Metro', 0.10),
( 'Sacos de Proteção', 'Embalagem', 'Unidade', 0.20),
( 'Microfibra', 'Tecidos', 'Metro Linear', 1.36),
( 'Lona', 'Tecidos', 'Metro Linear', 1.43),
( 'Cortiça', 'Solas', 'Placa', 11.00),
( 'EVA (Etileno Vinil Acetato)', 'Solas', 'Placa', 12.00),
( 'Fita Adesiva', 'Embalagem', 'Rolo', 1.90),
( 'Cartão', 'Embalagem', 'Folha', 0.03),
( 'Feltro', 'Tecidos', 'Metro Linear', 0.90);
```

#### 8.4.4 Subcontratado

```
-- Inserir dados na tabela Subcontratado
INSERT INTO Subcontratado (nome, morada, contacto, custoServico) VALUES
( 'Oficina de Costura Silva', 'Rua das Flores, 123, Lisboa', '911223344', 1.50),
( 'Montagem Rápida Lda.', 'Avenida Central, 45, Porto', '933445566', 0.80),
( 'Cortes Precisos Unipessoal', 'Travessa da Indústria, 789, Braga', '966778899',
0.50),
( 'Acabamentos de Qualidade SA', 'Largo do Comércio, 10, Setúbal', '922334455', 0.30),
( 'Colagens Industriais Oliveira', 'Zona Industrial, Lote 5, Aveiro', '955667788',
0.25),
( 'Pespontos Modernos', 'Rua Nova, 234, Guimarães', '944556677', 0.40),
( 'Embalagens Seguras Lda', 'Parque Empresarial, Edifício A, Faro', '977889900', 0.15),
( 'Controlo de Qualidade Total', 'Rua Direita, 567, Coimbra', '988990011', 0.20),
( 'Etiquetas Personalizadas', 'Avenida dos Descobrimentos, 89, Lagos', '912345678',
0.05),
( 'Subcontratação de Mão de Obra Especializada', 'Rua da Liberdade, 112, Évora',
'934567890', 2.00),
( 'Serviços de Costura Fina', 'Rua Augusta, 321, Viana do Castelo', '967890123', 1.75),
( 'Unidade de Montagem Avançada', 'Avenida Marginal, 654, Matosinhos', '923456789',
0.90),
( 'Corte a Laser Industrial', 'Zona Industrial da Maia, Lote 10, Maia', '956789012',
0.60),
( 'Acabamentos Manuais Profissionais', 'Rua do Bonfim, 43, Vila do Conde', '947890123',
0.35),
( 'Colagem de Precisão', 'Estrada Nacional 109, Figueira da Foz', '978901234', 0.30),
( 'Pespontos Criativos', 'Rua da Igreja, 76, Barcelos', '919012345', 0.45),
( 'Embalagens Ecológicas', 'Parque Industrial de Famalicão, Lote 8, Vila Nova de
Famalicão', '930123456', 0.18),
( 'Inspeção de Qualidade Rigorosa', 'Rua do Souto, 98, Braga', '961234567', 0.22),
( 'Etiquetagem Automática', 'Avenida da República, 150, Espinho', '924567891', 0.07),
( 'Fornecedor de Serviços de Subcontratação', 'Rua da Alfândega, 200, Lisboa',
'957891234', 1.20);
```

#### 8.4.5 FichaTecnica

```
-- Inserir dados na tabela FichaTecnica (relacionados com os produtos criados)
INSERT INTO FichaTecnica (produtoID, origem, descricao, estado) VALUES
(1, 'interna', 'Ficha técnica para Bota de Couro Cano Alto Preta.', 'ativa'),
(2, 'interna', 'Especificações da Bota de Couro Cano Curto Castanha.', 'ativa'),
(3, 'externa', 'Instruções da Bota Tática Impermeável Preta.', 'ativa'),
(4, 'interna', 'Processo de montagem da Bota de Montanha Aventura.', 'ativa'),
(5, 'interna', 'Ficha técnica da Sandália de Couro Verão.', 'ativa'),
(6, 'externa', 'Especificações da Sandália Desportiva Ajustável.', 'ativa'),
```

```
(7, 'interna', 'Ficha de produção da Sandália Plataforma Elegante.', 'inativa'),
(8, 'interna', 'Especificações do Tênis de Corrida Performance.', 'ativa'),
(9, 'interna', 'Detalhes da produção do Tênis Casual Urbano.', 'ativa'),
(10, 'externa', 'Ficha técnica do Tênis Desportivo Treino.', 'ativa'),
(11, 'interna', 'Processo de fabrico do Sapato Social Clássico Preto.', 'ativa'),
(12, 'interna', 'Especificações do Sapato Social Clássico Castanho.', 'ativa'),
(13, 'interna', 'Detalhes da montagem do Mocassim de Couro Confortável.', 'ativa'),
(14, 'externa', 'Ficha técnica do Sapato Oxford Formal Preto.', 'ativa'),
(15, 'interna', 'Processo de fabrico dos Chinelos de Piscina Antiderrapantes.', 'ativa'),
(16, 'interna', 'Especificações dos Chinelos de Casa Confortáveis.', 'ativa'),
(17, 'interna', 'Detalhes da produção das Espadrilles de Lona Bege.', 'ativa'),
(18, 'externa', 'Ficha técnica das Alpargatas de Tecido Azul Marinho.', 'ativa'),
(19, 'interna', 'Processo de fabrico das Sabrinas Clássicas Pretas.', 'ativa'),
(20, 'interna', 'Especificações das Sabrinas Coloridas Vermelhas.', 'ativa');
```

#### 8.4.6 DetalheFichaTecnica

```
-- Detalhes para a Ficha Técnica 1 (produtoID 1, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(1, 1, 1, 1.00, 'Corte do couro.', 15.00, 0.10),
(1, 2, 2, 1.00, 'Costura das partes.', 25.00, 0.08),
(1, 3, 3, 1.00, 'Montagem da sola.', 20.00, 0.12),
(1, 4, 4, 1.00, 'Acabamento e inspeção.', 10.00, 0.05);

-- Detalhes para a Ficha Técnica 2 (produtoID 2, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(2, 1, 1, 1.00, 'Corte do couro.', 12.00, 0.10),
(2, 2, 2, 1.00, 'Costura das partes.', 20.00, 0.08),
(2, 3, 3, 1.00, 'Montagem da sola.', 18.00, 0.12),
(2, 4, 4, 1.00, 'Acabamento e inspeção.', 8.00, 0.05);

-- Detalhes para a Ficha Técnica 3 (produtoID 3, externa)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(3, 3, 1, 1.00, 'Montagem completa (externo).', 40.00, 0.20);

-- Detalhes para a Ficha Técnica 4 (produtoID 4, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(4, 1, 1, 1.00, 'Corte dos materiais.', 18.00, 0.11),
(4, 2, 2, 1.00, 'Costura das partes.', 30.00, 0.09),
(4, 3, 3, 1.00, 'Montagem da sola.', 22.00, 0.14),
(4, 4, 4, 1.00, 'Acabamento e inspeção.', 15.00, 0.07);

-- Detalhes para a Ficha Técnica 5 (produtoID 5, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(5, 1, 1, 1.00, 'Corte das tiras.', 8.00, 0.10),
(5, 2, 2, 3.00, 'Costura das tiras.', 15.00, 0.08),
(5, 3, 3, 1.00, 'Montagem da sola.', 10.00, 0.12),
(5, 4, 4, 1.00, 'Acabamento.', 5.00, 0.05);

-- Detalhes para a Ficha Técnica 6 (produtoID 6, externa)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
```

```

(6, 2, 1, 1.00, 'Costura das tiras (externo).', 25.00, 0.10),
(6, 3, 2, 1.00, 'Montagem final (externo).', 15.00, 0.15);

-- Detalhes para a Ficha Técnica 7 (produtoID 7, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(7, 1, 1, 1.00, 'Corte dos materiais.', 10.00, 0.12),
(7, 2, 2, 1.00, 'Costura das partes.', 20.00, 0.09),
(7, 3, 3, 1.00, 'Montagem da plataforma.', 18.00, 0.15),
(7, 4, 4, 1.00, 'Acabamento.', 7.00, 0.06);

-- Detalhes para a Ficha Técnica 8 (produtoID 8, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(8, 1, 1, 1.00, 'Corte do tecido.', 10.00, 0.10),
(8, 2, 2, 1.00, 'Costura das partes.', 20.00, 0.08),
(8, 3, 3, 1.00, 'Montagem da sola.', 15.00, 0.13),
(8, 4, 4, 1.00, 'Acabamento.', 7.00, 0.05);

-- Detalhes para a Ficha Técnica 9 (produtoID 9, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(9, 1, 1, 1.00, 'Corte do tecido.', 12.00, 0.09),
(9, 2, 2, 1.00, 'Costura das partes.', 22.00, 0.07),
(9, 3, 3, 1.00, 'Montagem da sola.', 16.00, 0.11),
(9, 4, 4, 1.00, 'Acabamento.', 8.00, 0.05);

-- Detalhes para a Ficha Técnica 10 (produtoID 10, externa)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(10, 3, 1, 1.00, 'Montagem completa (externo).', 35.00, 0.18);

-- Detalhes para a Ficha Técnica 11 (produtoID 11, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(11, 1, 1, 1.00, 'Corte do couro.', 18.00, 0.10),
(11, 2, 2, 1.00, 'Costura das partes.', 35.00, 0.08),
(11, 3, 3, 1.00, 'Montagem da sola.', 25.00, 0.15),
(11, 4, 4, 1.00, 'Acabamento e lustro.', 12.00, 0.06);

-- Detalhes para a Ficha Técnica 12 (produtoID 12, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(12, 1, 1, 1.00, 'Corte do couro.', 17.00, 0.11),
(12, 2, 2, 1.00, 'Costura das partes.', 33.00, 0.09),
(12, 3, 3, 1.00, 'Montagem da sola.', 23.00, 0.14),
(12, 4, 4, 1.00, 'Acabamento e lustro.', 11.00, 0.07);

-- Detalhes para a Ficha Técnica 13 (produtoID 13, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(13, 1, 1, 1.00, 'Corte do couro.', 14.00, 0.09),
(13, 2, 2, 1.00, 'Costura das partes.', 28.00, 0.07),
(13, 3, 3, 1.00, 'Montagem da sola.', 20.00, 0.12),
(13, 4, 4, 1.00, 'Acabamento e palmilha.', 9.00, 0.06);

-- Detalhes para a Ficha Técnica 14 (produtoID 14, externa)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(14, 2, 1, 1.00, 'Costura (externo).', 40.00, 0.12),

```

```

(14, 3, 2, 1.00, 'Montagem (externo).', 20.00, 0.20);

-- Detalhes para a Ficha Técnica 15 (produtoID 15, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(15, 1, 1, 2.00, 'Corte do material.', 5.00, 0.05),
(15, 3, 2, 1.00, 'Montagem da sola.', 8.00, 0.10),
(15, 4, 3, 1.00, 'Acabamento.', 3.00, 0.04),
(15, 2, 4, 1.00, 'Costura (se aplicável).', 6.00, 0.03); -- Adicionando costura, pode
ser opcional

-- Detalhes para a Ficha Técnica 16 (produtoID 16, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(16, 1, 1, 2.00, 'Corte do material.', 8.00, 0.08),
(16, 2, 2, 1.00, 'Costura das partes.', 12.00, 0.06),
(16, 3, 3, 1.00, 'Montagem da sola.', 5.00, 0.10),
(16, 4, 4, 1.00, 'Acabamento.', 3.00, 0.04);

-- Detalhes para a Ficha Técnica 17 (produtoID 17, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(17, 1, 1, 1.00, 'Corte da lona.', 7.00, 0.07),
(17, 2, 2, 1.00, 'Costura da lona.', 15.00, 0.06),
(17, 3, 3, 1.00, 'Montagem da sola.', 18.00, 0.11),
(17, 4, 4, 1.00, 'Acabamento.', 6.00, 0.05);

-- Detalhes para a Ficha Técnica 18 (produtoID 18, externa)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(18, 3, 1, 1.00, 'Montagem completa (externo).', 25.00, 0.10);

-- Detalhes para a Ficha Técnica 19 (produtoID 19, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(19, 1, 1, 1.00, 'Corte da pele.', 9.00, 0.08),
(19, 2, 2, 1.00, 'Costura das partes.', 18.00, 0.07),
(19, 3, 3, 1.00, 'Montagem da sola.', 12.00, 0.10),
(19, 4, 4, 1.00, 'Acabamento.', 6.00, 0.05);

-- Detalhes para a Ficha Técnica 20 (produtoID 20, interna)
INSERT INTO DetalheFichaTecnica (fichaTecnicaID, operacaoID, ordemExecucao,
vezesPorUnidade, descricao, duracaoUnitariaMin, custoMinuto) VALUES
(20, 1, 1, 1.00, 'Corte da pele.', 10.00, 0.09),
(20, 2, 2, 1.00, 'Costura das partes.', 20.00, 0.08),
(20, 3, 3, 1.00, 'Montagem da sola.', 14.00, 0.11),
(20, 4, 4, 1.00, 'Acabamento.', 7.00, 0.06);

```

### 8.4.7 OrdemFabrico

```
-- Inserir dados na tabela OrdemFabrico
INSERT INTO OrdemFabrico (dataEmissao, dataPrevistaConclusao) VALUES
('2025-05-10', '2025-05-15'), -- Passado
('2025-05-12', '2025-05-18'), -- Passado
('2025-05-15', '2025-05-20'), -- Passado
('2025-05-23', '2025-05-30'), -- Futuro
('2025-05-24', '2025-06-01'), -- Futuro
('2025-05-18', '2025-05-22'), -- Passado
('2025-05-20', '2025-05-25'), -- Futuro
('2025-05-11', '2025-05-16'), -- Passado
('2025-05-22', '2025-05-29'), -- Futuro
('2025-05-13', '2025-05-19'), -- Passado
('2025-05-01', '2025-05-07'), -- Passado
('2025-05-05', '2025-05-11'), -- Passado
('2025-05-08', '2025-05-14'), -- Passado
('2025-05-25', '2025-06-05'), -- Futuro
('2025-05-28', '2025-06-08'), -- Futuro
('2025-05-19', '2025-05-21'), -- Passado
('2025-05-22', '2025-05-27'), -- Futuro
('2025-05-03', '2025-05-09'), -- Passado
('2025-05-26', '2025-06-02'), -- Futuro
('2025-05-16', '2025-05-20'); -- Passado
```

### 8.4.8 ItemOrdemFabrico

```
-- Inserir dados na tabela ItemOrdemFabrico
INSERT INTO ItemOrdemFabrico (ordemID, produtoID, quantidadePlaneada, fichaTecnicaID)
VALUES
(1, 1, 100, 1), -- Ordem 1: 100 Botas Cano Alto Pretas (Ficha Técnica 1)
(1, 2, 50, 2), -- Ordem 1: 50 Botas Cano Curto Castanhas (Ficha Técnica 2)
(2, 5, 200, 5), -- Ordem 2: 200 Sandálias de Couro Verão (Ficha Técnica 5)
(3, 9, 150, 9), -- Ordem 3: 150 Tênis Casual Urbanos (Ficha Técnica 9)
(4, 11, 75, 11), -- Ordem 4: 75 Sapatos Sociais Clássicos Pretos (Ficha Técnica 11)
(5, 16, 300, 16), -- Ordem 5: 300 Chinelos de Casa Confortáveis (Ficha Técnica 16)
(6, 1, 50, 1), -- Ordem 6: 50 Botas Cano Alto Pretas (Ficha Técnica 1)
(6, 19, 100, 19), -- Ordem 6: 100 Sabrinas Clássicas Pretas (Ficha Técnica 19)
(7, 2, 75, 2), -- Ordem 7: 75 Botas Cano Curto Castanhas (Ficha Técnica 2)
(8, 5, 120, 5), -- Ordem 8: 120 Sandálias de Couro Verão (Ficha Técnica 5)
(9, 9, 90, 9), -- Ordem 9: 90 Tênis Casual Urbanos (Ficha Técnica 9)
(10, 11, 40, 11), -- Ordem 10: 40 Sapatos Sociais Clássicos Pretos (Ficha Técnica 11)
(11, 16, 200, 16), -- Ordem 11: 200 Chinelos de Casa Confortáveis (Ficha Técnica 16)
(12, 3, 60, 3), -- Ordem 12: 60 Botas Táticas Impermeáveis Pretas (Ficha Técnica 3)
(13, 6, 180, 6), -- Ordem 13: 180 Sandálias Desportivas Ajustáveis (Ficha Técnica 6)
(14, 10, 110, 10), -- Ordem 14: 110 Tênis Desportivos Treino (Ficha Técnica 10)
(15, 14, 55, 14), -- Ordem 15: 55 Sapatos Oxford Formais Pretos (Ficha Técnica 14)
(16, 18, 250, 18), -- Ordem 16: 250 Alpargatas de Tecido Azul Marinho (Ficha Técnica 18)
(17, 20, 80, 20), -- Ordem 17: 80 Sabrinas Coloridas Vermelhas (Ficha Técnica 20)
(18, 4, 30, 4); -- Ordem 18: 30 Botas de Montanha Aventura (Ficha Técnica 4)
```

## 8.4.9 MaterialConsumoReal

```
-- INSERIR AO FIM PARA PROPOSITOS DEMONSTRATIVOS
-- Inserir dados na tabela ConsumoMaterialReal
-- Estes dados representam o consumo planeado de materiais
-- para operações específicas detalhadas nas Fichas Técnicas.
-- A coluna quantidadeUtilizada é deixada como NULL, pois as operações ainda não foram
realizadas.
-- Assumimos que os operacaoID's são 1=Corte, 2=Costura, 3=Montagem, 4=Acabamento.
-- Os detalheID's são sequenciais com base nas inserções anteriores de
DetalheFichaTecnica.

-- Detalhes para a Ficha Técnica 1 (Bota de Couro Cano Alto Preta - Interna)
-- DetalheID's assumidos: 1 a 5
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(1, 1, 0.50), -- Corte: Pele Genuína (0.50 Pé Quadrado por unidade)
(1, 7, 0.20), -- Corte: Tecido Sintético (para forro, 0.20 Metro Linear por unidade)
(2, 4, 0.05), -- Costura (Etapa 1): Linha de Costura (0.05 Carretel por unidade)
(3, 4, 0.07), -- Costura (Etapa 2): Linha de Costura (0.07 Carretel por unidade)
(4, 2, 1.00), -- Montagem: Sola de Borracha (1 Par por unidade)
(4, 6, 1.00), -- Montagem: Palmilha (1 Par por unidade)
(4, 5, 0.01), -- Montagem: Cola de Sapato (0.01 Litro por unidade)
(5, 3, 1.00), -- Acabamento: Atacadores (1 Par por unidade)
(5, 12, 1.00); -- Acabamento: Etiquetas (1 Unidade por unidade)

-- Detalhes para a Ficha Técnica 2 (Bota de Couro Cano Curto Castanha - Interna)
-- DetalheID's assumidos: 6 a 9
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(6, 1, 0.40), -- Corte: Pele Genuína
(7, 4, 0.04), -- Costura: Linha de Costura
(8, 2, 1.00), -- Montagem: Sola de Borracha
(8, 5, 0.008), -- Montagem: Cola de Sapato
(9, 13, 1.00); -- Acabamento: Caixas de Sapatos

-- Detalhes para a Ficha Técnica 3 (Bota Tática Impermeável Preta - Externa)
-- DetalheID assumido: 10
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(10, 26, 1.00), -- Montagem completa: Membrana Impermeável (1 Metro Linear por
unidade, se fornecido por nós)
(10, 22, 1.00); -- Montagem completa: Sola de Poliuretano (1 Par por unidade, se
fornecido por nós)

-- Detalhes para a Ficha Técnica 4 (Bota de Montanha Aventura - Interna)
-- DetalheID's assumidos: 11 a 14
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(11, 1, 0.60), -- Corte: Pele Genuína
(11, 26, 0.30), -- Corte: Membrana Impermeável
(12, 4, 0.08), -- Costura: Linha de Costura
(13, 2, 1.00), -- Montagem: Sola de Borracha
(13, 5, 0.015); -- Montagem: Cola de Sapato

-- Detalhes para a Ficha Técnica 5 (Sandália de Couro Verão - Interna)
-- DetalheID's assumidos: 15 a 18
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(15, 1, 0.20), -- Corte: Pele Genuína
(16, 4, 0.03), -- Costura: Linha de Costura
(17, 2, 1.00), -- Montagem: Sola de Borracha
(17, 5, 0.005); -- Montagem: Cola de Sapato

-- Detalhes para a Ficha Técnica 6 (Sandália Desportiva Ajustável - Externa)
-- DetalheID's assumidos: 19 a 20
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
```

```

(19, 7, 0.25), -- Costura: Tecido Sintético (se fornecido)
(20, 22, 1.00); -- Montagem: Sola de Poliuretano (se fornecido)

-- Detalhes para a Ficha Técnica 7 (Sandália Plataforma Elegante - Interna)
-- DetalheID's assumidos: 21 a 24
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(21, 8, 0.10), -- Corte: Camurça
(21, 9, 0.05), -- Corte: Glitter
(22, 4, 0.06), -- Costura: Linha de Costura
(23, 10, 1.00), -- Montagem: Plataforma
(23, 2, 1.00), -- Montagem: Sola
(24, 11, 1.00); -- Acabamento: Fivela

-- Detalhes para a Ficha Técnica 8 (Tênis de Corrida Performance - Interna)
-- DetalheID's assumidos: 25 a 28
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(25, 7, 0.35), -- Corte: Tecido Sintético Respirável
(25, 14, 0.10), -- Corte: Reforço Lateral
(26, 4, 0.07), -- Costura: Linha de Costura
(27, 15, 1.00), -- Montagem: Entressola Amortecedora
(27, 2, 1.00), -- Montagem: Sola de Borracha
(28, 3, 1.00); -- Acabamento: Atacadores

-- Detalhes para a Ficha Técnica 9 (Tênis Casual Urbano - Interna)
-- DetalheID's assumidos: 29 a 32
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(29, 7, 0.30), -- Corte: Tecido Lona
(29, 16, 0.05), -- Corte: Detalhe em Couro Sintético
(30, 4, 0.06), -- Costura: Linha de Costura
(31, 2, 1.00), -- Montagem: Sola de Borracha
(32, 3, 1.00); -- Acabamento: Atacadores

-- Detalhes para a Ficha Técnica 10 (Tênis Desportivo Treino - Externa)
-- DetalheID assumido: 33
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(33, 27, 1.00); -- Montagem completa: Tênis Desportivo (se fornecemos algum
componente)

-- Detalhes para a Ficha Técnica 11 (Sapato Social Clássico Preto - Interna)
-- DetalheID's assumidos: 34 a 37
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(34, 1, 0.45), -- Corte: Pele Genuína
(35, 4, 0.08), -- Costura: Linha de Costura
(36, 17, 1.00), -- Montagem: Sola de Couro
(36, 18, 1.00), -- Montagem: Salto
(37, 19, 1.00); -- Acabamento: Graxa para Couro

-- Detalhes para a Ficha Técnica 12 (Sapato Social Clássico Castanho - Interna)
-- DetalheID's assumidos: 38 a 41
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(38, 1, 0.45), -- Corte: Pele Genuína
(39, 4, 0.08), -- Costura: Linha de Costura
(40, 17, 1.00), -- Montagem: Sola de Couro
(40, 18, 1.00), -- Montagem: Salto
(41, 19, 1.00); -- Acabamento: Graxa para Couro

-- Detalhes para a Ficha Técnica 13 (Mocassim de Couro Confortável - Interna)
-- DetalheID's assumidos: 42 a 45
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(42, 1, 0.35), -- Corte: Pele Genuína
(43, 4, 0.06), -- Costura: Linha de Costura
(44, 17, 1.00), -- Montagem: Sola de Couro

```



```

(45, 20, 1.00); -- Acabamento: Palmilha Confortável

-- Detalhes para a Ficha Técnica 14 (Sapato Oxford Formal Preto - Externa)
-- DetalheID assumido: 46
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(46, 28, 1.00); -- Montagem completa: Sapato Oxford (se fornecemos algum componente)

-- Detalhes para a Ficha Técnica 15 (Chinelos de Piscina Antiderrapantes - Interna)
-- DetalheID's assumidos: 47 a 50
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(47, 21, 0.15), -- Corte: Borracha Antiderrapante
(48, 21, 0.10), -- Corte: Tira de Borracha
(49, 5, 0.003), -- Montagem: Cola
(50, 29, 1.00); -- Acabamento: Embalagem

-- Detalhes para a Ficha Técnica 16 (Chinelos de Casa Confortáveis - Interna)
-- DetalheID's assumidos: 51 a 54
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(51, 23, 0.20), -- Corte: Tecido Felpudo
(51, 24, 0.15), -- Corte: Espuma
(52, 4, 0.03), -- Costura: Linha de Costura
(53, 25, 1.00), -- Montagem: Sola de EVA
(54, 29, 1.00); -- Acabamento: Embalagem

-- Detalhes para a Ficha Técnica 17 (Espadrilles de Lona Bege - Interna)
-- DetalheID's assumidos: 55 a 58
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(55, 7, 0.30), -- Corte: Lona
(56, 4, 0.05), -- Costura: Linha de Costura
(57, 30, 1.00), -- Montagem: Sola de Corda
(58, 31, 1.00); -- Acabamento: Etiqueta

-- Detalhes para a Ficha Técnica 18 (Alpargatas de Tecido Azul Marinho - Externa)
-- DetalheID assumido: 59
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(59, 32, 1.00); -- Montagem completa: Alpargatas (se fornecemos algum componente)

-- Detalhes para a Ficha Técnica 19 (Sabrinas Clássicas Pretas - Interna)
-- DetalheID's assumidos: 60 a 63
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(60, 1, 0.25), -- Corte: Pele Genuína
(61, 4, 0.04), -- Costura: Linha de Costura
(62, 2, 1.00), -- Montagem: Sola de Borracha
(63, 29, 1.00); -- Acabamento: Embalagem

-- Detalhes para a Ficha Técnica 20 (Sabrinas Coloridas Vermelhas - Interna)
-- DetalheID's assumidos: 64 a 67
INSERT INTO ConsumoMaterialReal (detalheID, materialID, quantidadePlaneada) VALUES
(64, 33, 0.25), -- Corte: Pele Sintética Colorida
(65, 4, 0.04), -- Costura: Linha de Costura
(66, 2, 1.00), -- Montagem: Sola de Borracha
(67, 29, 1.00); -- Acabamento: Embalagem

```



## 9. Referências

Connolly, T & Begg, C. (2015, April 17).

*Database systems: A practical approach to design, implementation and management* (6th ed.).

Pearson. <https://eu.pearson.com/products/9781292061849r365> ISBN: 9780132943260

dbdiagram. (n.d.). *DBML – Full syntax docs*. <https://dbml.dbdiagram.io/docs/>

dbdiagram. (n.d.). *Dbdiagram docs*. <https://docs.dbdiagram.io/>

Ecco®. (n.d.). *Coleção para homem*. <https://pt.ecco.com/pt-PT/Homem>

Gornik, D. (2023, June 11). - *Entity relationship modeling with UML*.

IBM. <https://www.uml.org.cn/UMLTools/pdf/ermodeling.pdf>

Ramakrishnan, R., & Gehrke, J. (2022).

*Database management systems* (3rd ed.).

McGraw-Hill. ISBN: 9780072465631

SPARX Systems. (n.d.) - *Database modeling in UML*.

<https://sparxsystems.com/resources/tutorials/uml/datamodel.html>

SPARX Systems. (n.d.) - *UML tutorial* - <https://sparxsystems.com/resources/tutorials/uml/part1.html>

SPARX Systems. (n.d.) - *UML 2 tutorial* - <https://sparxsystems.com/resources/tutorials/uml2/>

Stack Overflow. (n.d.).

*Entity relationship (ER) diagram enum values*.

<https://stackoverflow.com/questions/74615345/entity-relationship-er-diagram-enum-values>

Wikipedia. (n.d.) - *ISO 9000* - [https://pt.wikipedia.org/wiki/ISO\\_9000](https://pt.wikipedia.org/wiki/ISO_9000)