

## **Ordens de Fabrico**

**Artur Pinto nº8230138**

**Luís Garcês nº8230235**

**Wilkie Filho nº8230127**

Maio 2025

Data de Receção	
Responsável	
Avaliação	
Observações	

# **Agradecimentos**

Gostaríamos de expressar a nossa gratidão aos professores Vasco Santos e Bruno Oliveira, da UC Base de Dados, pelo suporte fornecido ao longo deste projeto e pelos conhecimentos transmitidos, que foram fundamentais para o desenvolvimento bem-sucedido deste trabalho.

# Resumo

O presente relatório descreve o desenvolvimento de uma base de dados relacional para apoiar a gestão de ordens de fabrico numa fábrica de calçado, no âmbito da unidade curricular de Bases de Dados. O sistema implementado tem como objetivo registar e controlar os processos/etapas de produção, desde a emissão das ordens de fabrico até à monitorização das operações, das perdas, dos custos e das respetivas subcontractações, respondendo aos requisitos estabelecidos no enunciado do trabalho.

O projeto iniciou-se com a análise detalhada dos requisitos de negócio, tendo sido identificadas as entidades, relacionamentos e atributos essenciais para modelar as etapas de produção. Seguiu-se a construção do modelo conceptual, validado com diagramas ER e tabelas de apoio, e a sua posterior transformação num modelo lógico onde este foi normalizado, para garantir assim a consistência da base de dados, ou seja, a integridade e eficiência dos dados. Foram também implementadas as restrições de integridade, as regras de negócio e os mecanismos automáticos, como os *triggers* e as *stored procedures*, para assegurar o correto funcionamento da base de dados e o cumprimento das validações e restrições exigidas.

A implementação foi realizada utilizando o *SQL Server Developer Edition* da Microsoft, onde foi implementado a criação das tabelas, dos procedimentos de manipulação de dados, as vistas para realizar as consultas necessárias e os *triggers* para o controlo automático de eventos importantes, como os alertas de atrasos e as perdas excessivas. A base de dados permite também efetuar consultas avançadas, como a apresentação de relatórios de desempenho e custos, e a monitorização da eficiência de produção.

O trabalho encontra-se concluído, com todos os requisitos essenciais implementados e validados. O modelo de dados foi testado e demonstra capacidade para suportar o crescimento futuro e adaptações a novos requisitos. O seguinte relatório documenta todas as decisões tomadas na realização do trabalho, justificando as opções de modelação e implementação seguidas, que foram sempre adotadas de acordo com as melhores práticas estudadas nas aulas.

# Índice

## Índice

Agradecimentos.....	iii
Resumo .....	iv
Índice .....	v
Índice de Figuras .....	viii
Índice de Tabelas .....	x
Lista de Siglas e Acrónimos .....	1
1. Introdução.....	1
1.1 Contextualização.....	1
1.2 Apresentação do Caso de Estudo .....	1
1.3 Motivação e Objetivos.....	2
1.4 Estrutura do Relatório .....	2
2. Construção do Modelo Concetual .....	4
2.1 Identificação das Entidades.....	4
2.2 Identificação de relacionamentos e respetiva multiplicidade .....	5
2.3 Identificação e associação de atributos a entidades ou relacionamentos .....	5
2.4 Identificação dos domínios de atributos .....	8
2.5 Escolha das chaves candidatas, primárias e alternativas.....	8
2.6 Verificação de redundância no modelo (Por Concluir) .....	9
2.7 Validação das transações.....	11
3. Construção do modelo lógico .....	12
3.1 Derivação de relações para o modelo de dados lógico .....	12
3.1.1 Entidades .....	12
3.1.2 Relacionamentos .....	16
3.2 Modelo de dados lógico resultante .....	18
3.3 Validar modelo lógico com Normalização .....	19
3.3.1 Normalização das Ordem de Fabrico .....	19
3.3.2 Normalização da Ficha Técnica.....	21
3.3.3 Modelo Lógico após Normalização.....	23

3.4 Validação das transações com o modelo Lógico .....	24
3.5 Identificação Restrições de Integridade dos dados.....	25
3.5.1 Matéria Prima.....	25
3.5.2 Produto.....	25
3.5.3 SubContratados .....	26
3.5.4 Operação .....	27
3.5.4 Ficha Técnica.....	27
3.5.5 Ordem de Fabrico .....	27
3.5.6 Etapa de Produção .....	28
3.5.7 Etapa da Ordem.....	28
3.5.8 Restrições domínios.....	31
3.6 Regras de Negócio .....	31
Gestão Automatizada de Ordens de fabrico.....	31
Controlo de Produção .....	31
Cálculos Automáticos.....	31
Validações de Processo.....	31
Relatórios e Consultas.....	32
3.7 Verificação do provável crescimento futuro F .....	32
4. Construção do modelo físico.....	33
4.1 Criação das tabelas no SQL Server Management Studio.....	33
4.2 Procedures criadas para resolver os problemas de Negocio.....	34
Criar registos de produção para todas as operações da ficha técnica.....	34
Finish Orde stage.....	34
4.2 Criação das procedures de CUD (Create, Update, Delete) no SQL Server Management Studio.....	36
4.3 Views .....	37
Eficiência Media Produção Por Ano .....	37
Eficiência Media Produção Por Mês .....	38
Eficiência Media Produção Por Dia .....	38
Média Perdas de Operação em Percentagem.....	38
Operações pendentes Por Ordem .....	38
Custo de Produção da Ordem de Fabrico .....	39

Custos Comparativos .....	39
Ordens Atrasadas .....	39
Perdas Superiores a 15% .....	40
Operações Problemáticas .....	40
Operações SubContratadas.....	40
Ordens Atrasadas .....	40
Ordens Fabrico Progresso .....	41
Tempo total estimado por ordem .....	41
4.4 Triggers.....	41
Validações operações pendentes Ordem .....	42
Cálculo do custo total da ordem de produção de uma ordem .....	43
Verificação de possibilidade de alterar o status de uma ordem de produção .....	44
4.5 SQL Server Agent.....	44
5. Conclusões e Trabalho Futuro .....	1
Bibliografia.....	1
Referências WWW .....	2

# Índice de Figuras

Figura 1 - 1º Diagrama ER modelo conceptual	5
Figura 2 - Diagrama de transações	11
Figura 3 - Modelo Lógico Resultante	18
Figura 4 - Exemplo de uma Ordem de Fabrico	19
Figura 5 - Legenda dos campos da Ordem de Fabrico	19
Figura 6 - Forma Não Normalizada da Ordem de Fabrico	19
Figura 7 - 2FN Ordem Fabrico Dependencia Parcias	20
Figura 8 - 2FN dependencias parciais detetadas	20
Figura 9 - 3FN Ordem de Fabrico	20
Figura 10 - Exemplo de uma FichaTecnica	21
Figura 11 - Legenda dos campos	21
Figura 12 - Forma Não Normalizada Ficha Técnica	21
Figura 13 - 1FN da Ficha Técnica	21
Figura 14 - 2ºFN da Ficha Técnica	22
Figura 15 - 3º FN da Ficha Tecnica	22
Figura 16 - Modelo Lógico após Normalização	23
Figura 17 - Transações modelo lógico	24
Figura 18 - 1º img Registos de produção	34
Figura 19 - 2º img Registos de produção	34
Figura 20 - verificação ordem das operacoes	35
Figura 21 - alertas de perdas superiores a 15%	35
Figura 22 - soma das perdas da operação	36
Figura 23 - Atualização automatica da ordem	36
Figura 24 - Storage procedures criadas	37
Figura 25 - Eficiencia Media Producao por ano	37
Figura 26 - Tabela da eficiencia media producao por ano	37
Figura 27 - Eficiencia Media Produção por Mês	38
Figura 28 - Tabela da Eficiencia Media Producao por Mês	38
Figura 29 - Eficiência Média Produção por dia	38
Figura 30 - Tabela Eficiência Média Produção por dia	38
Figura 31 - Media de Perdas de Operação em Perdas	38
Figura 32 - Operações Por ordem	38
Figura 33 - Tabela das Operações por Ordem	38
Figura 34 - Custo de Produção da Ordem de Fabrico	39
Figura 35 - Tabela do Custo de Produção da Ordem de Fabrico	39
Figura 36 - Custos Comparativos	39
Figura 37 - Tabela dos Custos Comparativos	39
Figura 38 - Ordens Atrasadas	39



Figura 39 - Tabela das Ordens Atrasadas	39
Figura 40 - Perdas Superiores a 15%	40
Figura 41 - Tabelas de Perdas Superiores a 15%	40
Figura 42 - Operações Problemáticas	40
Figura 43 - Tabela das Operações Problemáticas	40
Figura 44 - Operações SubContratadas	40
Figura 45 - Tabelas Operacoes SubContratadas	40
Figura 46 - Ordens Atrasadas	40
Figura 47 - Tabelas das Ordens Atrasadas	40
Figura 48 - Ordens de Fabrico Progresso	41
Figura 49 - Tabela Progresso das Ordens de Fabrico	41
Figura 50 - Tempo total estimado por ordem	41
Figura 51 - Exemplo do funcionamento da vista Tempo total estimado por ordem	41
Figura 52 - Calculos custos da ordem	42
Figura 53- Validação Status da Ordem	42
Figura 54 - Trigger para cálculo automatizado do custo total da ordem de produção de uma ordem	43
Figura 55 - Trigger que verifica se é possível alterar o status de uma ordem de produção para concluída verificando se todas as etapas dessa ordem já estão concluídas	44
Figura 56 - Storage Procedure ALERT_ORDER_LATE	45
Figura 57 - Tabela de alertas	45
Figura 58 - Nome do job	46
Figura 59 - Etapa do job	47
Figura 60 - Programação do Agente	47

# Índice de Tabelas

Tabela 1 - Lista de Siglas e Acrónimos .....	1
Tabela 2 - Identificação das Entidades .....	4
Tabela 3 - Relacionamentos e multiplicidade .....	5
Tabela 4 - Entidades e atributos .....	8
Tabela 5 - Domínios dos atributos.....	8
Tabela 6 - Chaves Candidatas, Primarias e Alternativas .....	9
Tabela 7 - Diagrama conceptual com chaves primárias .....	9
Tabela 8 - Entidades Fortes .....	14
Tabela 9 - Chaves Primarias e Estrangeiras, entidades Fortes .....	14
Tabela 10 - Entidades Fracas .....	15
Tabela 11 - Chaves Entidades Fracas.....	16

## Lista de Siglas e Acrónimos

Sigla	Designação
BD	Base de dados
SQL	<i>Structured Query Language</i>
PK	Chave Primária ( <i>Primary Key</i> )
FK	Chave Estrangeira ( <i>Foreign Key</i> )
1FN	1º Forma Normal
2FN	2º Forma Normal
3FN	3º Forma Normal

*Tabela 1 - Lista de Siglas e Acrónimos*

# **1. Introdução**

## **1.1 Contextualização**

A indústria do calçado em Felgueiras destaca-se pela sua importância económica, inovação e capacidade de adaptação às exigências do mercado nacional e internacional, revelando-se como uma área de atividade industrial de importante valorização da região de Felgueiras como também do país, valorizando os seus trabalhadores e contribuindo para o crescimento económico do setor e de Portugal. O setor caracteriza-se por uma forte componente produtiva, onde a eficiência, o controlo de qualidade e a gestão rigorosa dos processos são determinantes para a competitividade das empresas. Neste contexto, a gestão eficaz das ordens de produção assume um papel fundamental, visto que permite monitorizar todas as etapas do fabrico e conceção de um artigo (calçado), desde a receção da encomenda até à expedição do produto final, garantindo o controlo dos custos e uma resposta adequada às necessidades dos clientes.

Com a crescente complexidade dos processos produtivos e a necessidade de integração de operações internas e subcontratadas, torna-se fundamental dispor de sistemas de informação robustos que permitam a recolha, armazenamento e análise dos dados obtidos ao longo do ciclo de produção. Neste sentido, a informatização destes processos, através do desenvolvimento e implementação de uma base de dados relacional, permite não só automatizar tarefas relacionadas com as operações e administração, como também fornecer dados sobre o desempenho e a eficiência destas fábricas o que ajuda com base na informação recolhida a tomar melhores decisões para o negócio.

## **1.2 Apresentação do Caso de Estudo**

O caso de estudo apresentado neste trabalho assenta sobre a gestão de ordens de produção numa fábrica de calçado, onde são produzidos diversos artigos (sapatos, botas, sandálias), cada um com processos técnicos e operações específicas. Cada ordem de fabrico define os produtos a fabricar, as quantidades necessárias e o prazo de conclusão, sendo suportada por uma ficha técnica que apresenta a sequência de operações (ex.: corte, costura, montagem, acabamento), as matérias-primas envolvidas e os recursos necessários.

A base de dados a desenvolver deverá ser capaz de :

- Registrar e acompanhar todas as ordens de fabrico, incluindo o estado e progresso das mesmas.
- Controlar as operações realizadas, quantidades produzidas, perdas e custos, tanto em operações internas como subcontratadas.
- Identificar atrasos, perdas excessivas e operações problemáticas.
- Apresentar relatórios , nomeadamente sobre eficiência de produção, custos e desempenho dos parceiros externos , ou seja , dos subcontratados

## 1.3 Motivação e Objetivos

A principal motivação para a realização deste trabalho reside na necessidade de dotar a fábrica de calçado de uma ferramenta informática que permita:

- Melhorar o acesso e o controlo das ordens de fabrico.
- Automatizar a gestão de alertas e restrições dos processos de produção, minimizando erros e desperdícios.
- Disponibilizar informação em tempo útil para apoio à decisão, nomeadamente através de relatórios e indicadores de desempenho.
- Facilitar a integração entre operações internas e subcontratadas, assegurando o cumprimento dos prazos e a otimização dos recursos disponíveis .

Os objetivos específicos do projeto incluem:

- Levantar e analisar os requisitos dos processos de produção de uma ordem de fabrico .
- Desenhar um modelo de dados relacional que represente a realidade da fábrica de calçado .
- Implementar a base de dados em SQL Server, incluindo a criação da tabelas , dos triggers, procedures e as views para suportar as regras de negócio e as consultas exigidas.
- Validar o sistema com dados aleatórios para garantir a consistência e a integridade dos dados e da base de dados

## 1.4 Estrutura do Relatório

O relatório encontra-se organizado e estruturado da seguinte forma:

- O **Capítulo 1** apresenta a contextualização, o caso de estudo, a motivação e os objetivos do trabalho, bem como a estrutura do relatório.
- O **Capítulo 2** descreve a construção do modelo conceptual, incluindo a identificação das entidades, relacionamentos, atributos e validação do modelo conceptual.

- O **Capítulo 3** aborda a transformação do modelo conceptual para o modelo lógico, a normalização das relações, a definição das restrições de integridade e a validação e revisão do modelo lógico com o utilizador.
- O **Capítulo 4** detalha a implementação do modelo físico em SQL Server, a criação das tabelas, procedimentos, triggers e views.
- O **Capítulo 5** apresenta as conclusões, uma apreciação crítica do trabalho realizado e sugestões para desenvolvimentos futuros.
- Por fim, são incluídos anexos com informação adicional, referências bibliográficas e outros elementos de apoio à compreensão e à realização do trabalho

## 2. Construção do Modelo Concetual

### 2.1 Identificação das Entidades

Nome Entidade	Descrição	Pseudónimos	Ocorrências
Operacao	Termo geral que descreve todas as operações internas da empresa	Operações internas das empresas	Uma operação interna é realizada em zero ou mais Fichas Técnicas. Uma operação ela gerar um produto final (que é um material utilizado em outra operação)
SubContratado	Termo geral que descreve todas as operações subContratadas	Operações subcontratadas pela empresa	Uma operação sub Contratada pode ser contratada para zero ou mais Ordens de Fabrico. Um SubContrato gera um produto final (que é um material utilizado em outra operação)
FichaTecnica	Termo geral que descreve todas as fichas técnicas dos produtos	Ficha técnica do produto	Uma ficha técnica é composta pelas etapas de produção (que são um conjunto de operações), todos os materiais utilizados na etapa de produção e vai referenciar um produto
Produto	Termo geral que descreve todos os produtos que a empresa comercializa	Artigos	Um produto possui zero ou mais fichas técnicas.
MateriaPrima	Termo geral que descreve os Materiais utilizados pela empresa nas suas Etapas de Produção	Material	Um material pode ser utilizado em zero ou mais subContratos e em zero ou mais FichasTecnicas
OrdemFabrico	Termo geral que descreve todas as ordens de fabrico	Ordens de Fabrico	Uma ordem de fabrico inclui um ou mais produtos e zero ou mais operações subContratadas

*Tabela 2 - Identificação das Entidades*

## 2.2 Identificação de relacionamentos e respetiva multiplicidade

Nome Entidade	Multiplicidade	Relacionamentos	Multiplicidade	Nome Entidade
Produto	1..1	Descreve_possui	0..*	Ficha Tecnica
FichaTecnica	1..1	EtapasDeProducao	1..*	EtapaProducao
SubContratado	1..1	PodePossuir	0..*	EtapaOrdem
OrdemFabricao	1..1	FazParteDe	1..*	EtapaOrdem
Operacao	0..*	OperacoesNecessarias	1..*	EtapaProducao

Tabela 3 - Relacionamentos e multiplicidade

De maneira que seja mais perceptível a identificação dos relacionamentos e respetiva multiplicidade é possível observar abaixo um pequeno esboço do diagrama do modelo conceptual.

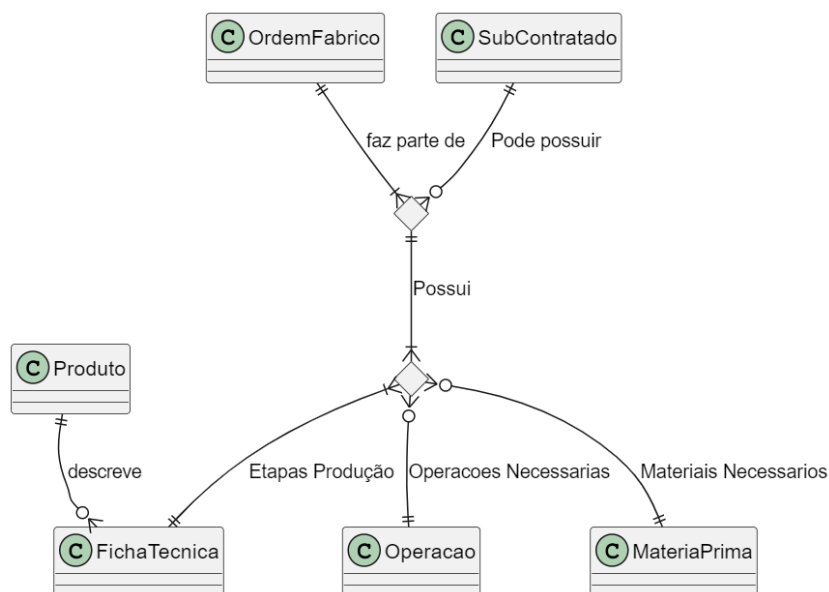


Figura 1 - 1º Diagrama ER modelo conceptual

## 2.3 Identificação e associação de atributos a entidades ou relacionamentos

Após identificarmos todas as entidades e as suas relações chegamos há etapa de definir os atributos de cada uma.

Nome Entidade	Atributos	Descrição	Tipo de Dado e Tamanho	Nulls	Multi-valued
---------------	-----------	-----------	------------------------	-------	--------------



Matéria Prima	MatNo		Codigo único da matéria prima	MatNo	Não	Não
	Name		Nome do Material	50 nvarchar	Não	Não
	PriceUn		Preço unitário desse material	PriceUn	Não	Não
	Stock	quantidade	Quantidade em stock do material	int	Não	Não
		unidadeStock	Unidade de medida desse stock	5 nvarchar	Não	Não
	Obs		Observações sobre a Materia Prima	250 nvarchar	Sim	Não
SubContratadas	SubContratadoNo		Código Único da operação subContratada	SubContratadoNo	Não	Não
	Price		Preço da operação	float	Não	Não
	ExpectedTime		Tempo esperado de duração do subContrato	int	Não	Não
	QuantidadeEnviada		Quantidade de material enviado ao subContratado	int	Não	Não
	QuantidadeRecebida		Quantidade do material final recebida do subContratado	int	Não	Não
	QuantidadePerdas		Quantidade de material defeituoso enviado pelo subContratado	int	Não	Não
Operação	OpNo		Código único da operação	OpNo	Não	Não
	Name		Nome da operação	30 nvarchar	Não	Não
	Obs		Observação sobre a operação	250 nvarchar	Sim	Não
FichaTecnica	FichTecNo		Codigo Único da Ficha Técnica	FichTecNo	Não	Não
	ProductionCost		Custo de Produção da Ficha Técnica	float	Não	Não

	ExpectedQuantity		Quantidade Esperada da produção do produto	int	Não	Não
	MadeAmount		Quantidade real produzida	int	Não	Não
	DiscardedAmount		Quantidade de produto desperdiçado	int	Não	Não
	QuantityMaterial		Quantidade de material utilizado	int	Não	Sim
	Obs		Observações da ficha Técnica	250 nvarchar	Sim	Não
Produto	ProductNo		Código único do produto	ProductNo	Não	Não
	PriceUn		Preço do unitário do produto	float	Não	Não
	Name		Nome do produto	30 nvarchar	Não	Não
	productType		Tipo do produto	20 nvarchar	Não	Não
	Descrição		Descrição do produto	250 nvarchar	Sim	Não
	Tamanhos		Tamanhos fabricáveis do produto (Ex: 41-45)	5 nvarchar	Não	Não
	Peso		Peso total do produto	float	Não	Não
	Género		Identifica o género a quem se identifica o produto	1 char	Não	Não
	Stock	Quantidade	Quantidade em stock do produto	int	Não	Não
		unidade	Unidade da quantidade de stock	5 nvarchar	Não	
OrdemFabrico	ManufacturingNo		Código único da ordem de fabrico	ManufacturingNo	Não	Não
	CostCharged		Custo cobrado ao cliente pelos produtos	float	Não	Não
	Status		Estado atual da ordem	1 nchar	Não	Não
	CreationDate		Data de criação da ordem.	DateTime	Não	Não
	StartDate		Data de início da ordem, quando foi iniciada a produção.	DateTime	Sim	Não

	ExpectedEndDate	Date esperada de termino da ordem de produção	DateTime	Não	Não
	EndDate	Data de termino da ordem de produção	Date	Sim	Não
	Priority	Prioridade da ordem de produção	smallint	Não	Não
	Obs	Observações da ordem de Fabrico	250 nvarchar	Não	Sim

Tabela 4 - Entidades e atributos

## 2.4 Identificação dos domínios de atributos

Nome do Dominio	Descrição	Tipo de Dado e Tamanho	Nulls
MatNo	Código Único do material	7 nchar	Não
OpNo	Código Único de uma Operação	5 nchar	Não
SubContratadoNo	Código Único do SubContratado	5 nchar	Não
FichTecNo	Código Único de uma Ficha Técnica	5 nchar	Não
ProductNo	Código único de um produto	5 nchar	Não
ManufactoringNo	Código único de uma ordem de fabrico	5 nchar	Não

Tabela 5 - Domínios dos atributos

## 2.5 Escolha das chaves candidatas, primárias e alternativas

Entidade	Chave Primaria	Chaves Candidatas
Matéria Prima	MatNo	Nome
Sub Contratado	SubContratadoNo	Nome
Operação	OpNo	Nome
Ficha Técnica	FichTecNo	-
Produto	ProductNo	Nome

Ordem Fabrico	ManufacturingNo	-
---------------	-----------------	---

Tabela 6 - Chaves Candidatas, Primárias e Alternativas

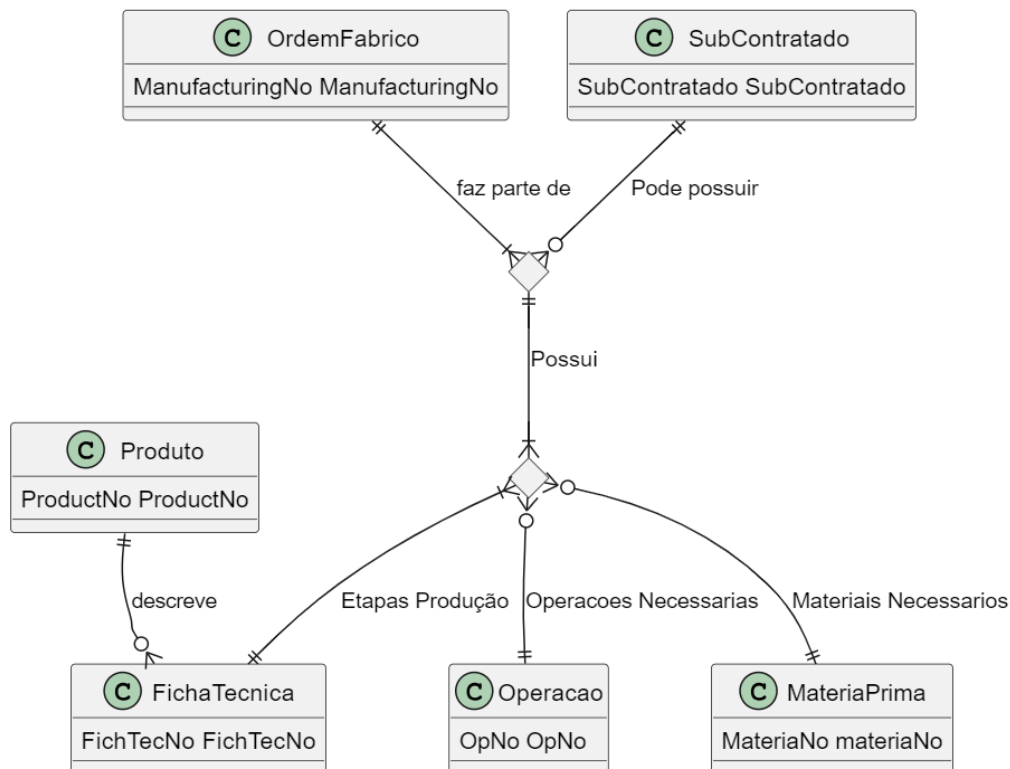


Tabela 7 - Diagrama conceitual com chaves primárias

## 2.6 Verificação de redundância no modelo (Por Concluir)

O processo de verificação de redundância no modelo de dados envolve várias etapas para garantir que não haja duplicações desnecessárias e que a integridade dos dados seja mantida. Este processo pode ser dividido em várias etapas, sendo elas:

### **Etapas 1:** Verificação de Relacionamentos 1 para 1

Nesta primeira etapa verificamos se existem entidades com um relacionamento 1 para 1. No nosso desenho conceitual não foi encontrado nenhum desses tipos de relacionamento, portanto não houve a necessidade de ajustes nesta etapa.

### **Etapas 2:** Remover relacionamentos redundantes

Um relacionamento é redundante se a mesma informação puder ser obtida por meio de outros relacionamentos. Manter o modelo conceitual sucinto, eliminando ligações desnecessárias. Ao rever o nosso diagrama não foi detetado nenhum desses problemas, visto que os únicos cenários possíveis era o relacionamento entre a Ficha Técnica e as Operações e a Ficha Técnica com as Matérias Prima, que já foi resolvido com um relacionamento Ternário, com o nome Etapa de Produção.

E outro possível problema que o relacionamento Binário resolve era o relacionamento entre as Ordens de Fabrico e as Etapas de Produção e as mesmas Etapas com os SubContratados. Logo o nosso diagrama não apresenta nenhum problema de relacionamentos redundantes.

### **Etapas 3: Dimensão Temporal**

A dimensão temporal das relações é essencial para avaliar corretamente a presença de redundâncias no modelo de dados. Esta análise permite compreender o contexto temporal de cada relacionamento, ajudando a evitar a eliminação de relações que, embora pareçam redundantes num dado instante, são fundamentais para manter a coerência histórica e funcional do sistema.

No nosso modelo, identificámos uma limitação ao nível da gestão temporal dos preços. Atualmente, os valores dos materiais e produtos estão armazenados diretamente nas respetivas tabelas (Material e Produto). No entanto, esta abordagem não regista o histórico de preços. Se o preço unitário de um material for alterado, essa alteração afetará retroativamente o custo das ordens de fabrico já emitidas, o que compromete a fidelidade dos dados históricos e pode ter consequências negativas na análise de custos da empresa.

Apesar desta falha pontual, o nosso diagrama evita relacionamentos redundantes e, na generalidade, respeita os princípios da dimensão temporal. Contudo, recomendamos a introdução de uma entidade associativa para preços com validade temporal para garantir a rastreabilidade correta ao longo do tempo.

## 2.7 Validação das transações

Na imagem abaixo, é possível observar o nosso diagrama de transações. Embora a Ficha Técnica e o Produto não estejam diretamente envolvidos em nenhuma transação representada, ambos desempenham um papel fundamental no funcionamento do sistema.

O Produto é essencial, pois sem ele as ordens de fabrico perderiam o seu propósito — não haveria referência sobre o que deve ser produzido. Da mesma forma, a Ficha Técnica fornece os detalhes do processo de produção, como os materiais e operações necessárias para fabricar o produto. Sem essa informação, seria impossível executar corretamente a produção.

Assim, mesmo que não participem diretamente em transações específicas no diagrama, o Produto e a Ficha Técnica são elementos estruturais indispensáveis ao funcionamento coerente e completo do sistema.

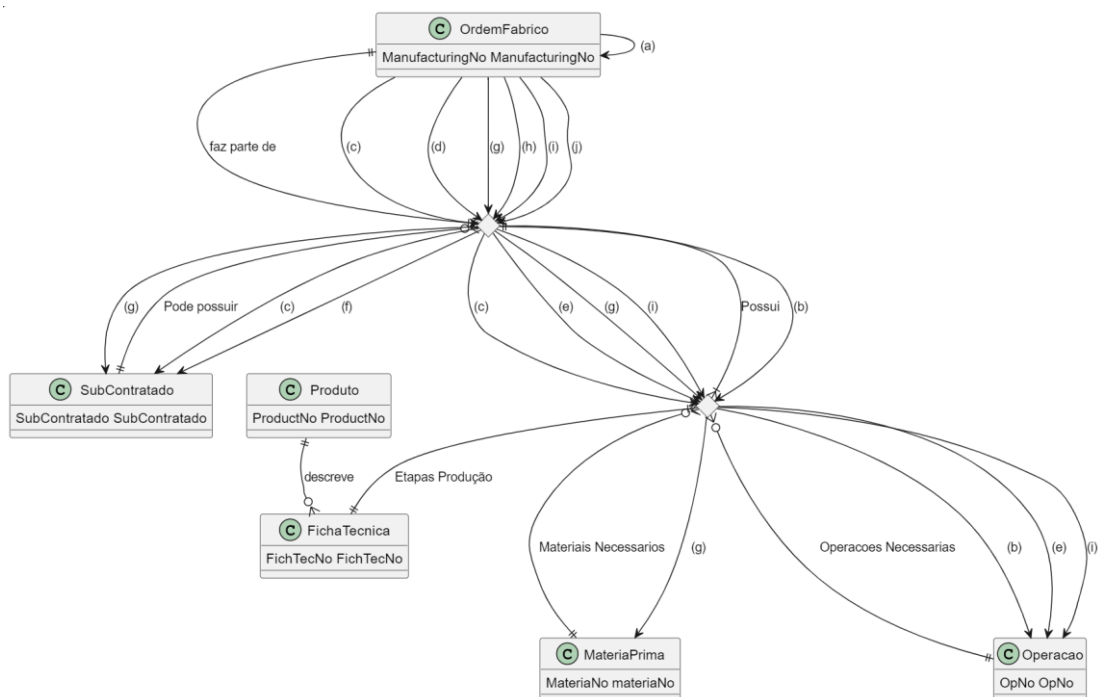


Figura 2 - Diagrama de transações

Legenda do diagrama

- a → Listar Ordens em atraso com o respetivo tempo de atraso;
- b → Listar Operações Problemáticas;
- c → Custos Comparativos entre operação interna e externa;
- d → Consultar as Ordens de fabrico em curso com indicação do progresso;
- e → Relatório de operações com perdas superiores a 15%;
- f → Listar as Operações mais frequentemente subContratadas e respetivos custos;
- g → Determinar o custo total de produção de uma ordem;
- h → Consultar lista de operações pendentes para cada ordem, ordenadas por prioridade;
- i → Estimar o tempo total necessários para concluir uma ordem, com base no tempo médio das operações;
- j → Calcular a eficiência média de produção por período.

### 3. Construção do modelo lógico

#### 3.1 Derivação de relações para o modelo de dados lógico

Neste tópico, vamos derivar os relacionamentos para o modelo lógico com base no modelo conceitual apresentado. Esta derivação inclui a definição de entidades, atributos, chaves primárias, chaves estrangeiras e a tradução de relacionamentos conforme identificado no modelo conceitual.

##### 3.1.1 Entidades

###### 3.1.1 Entidades Fortes

As entidades fortes são um tipo de entidade própria que cuja a sua existência não depende de outra, normalmente refeltida por uma única PK.

Nome Entidade	Atributos	Descrição	Tipo de Dado e Tamanho	Nulls
Matéria Prima	<u>MatNo</u>	Código único da matéria prima	MatNo	Não
	Name	Nome do Material	50 nvarchar	Não
	PriceUn	Preço unitário desse material	float	Não
	QuantityStock	Quantidade em stock do material	Int	Não
	unidadeStock	Unidade de medida desse stock	5 nchar	Não
	Obs	Observações sobre a Materia Prima	250 nvarchar	Sim
SubContratadas	<u>SubContratadoNo</u>	Código Único da operação subContratada	SubContratadoNo	Não
	Name	Nome do SubContratado	30 nvarchar	Não
	Obs	Observação sobre o subContratado	250 nvarchar	Sim
Operação	<u>OpNo</u>	Código único da operação	OpNo	Não

	Name	Nome da operação	30 nvarchar	Não
	Obs	Observação sobre a operação	250 nvarchar	Sim
Ficha Técnica	<u>FichTecNo</u>	Código Único da Ficha Técnica	FichTecNo	Não
	ProductNo	Código único do produto	ProductNo	Não
	Obs	Observações da ficha Técnica	250 nvarchar	Sim
Produto	<u>ProductN</u>	Código único do produto	ProductNo	Não
	PriceUn	Preço do unitário do produto	float	Não
	Name	Nome do produto	30 nvarchar	Não
	productType	Tipo do produto	20 nvarchar	Não
	Description	Descrição do produto	250 nvarchar	Sim
	Sizes	Tamanhos fabricaveis do produto (Ex: 41-45)	5 nchar	Não
	weight	Peso total do produto	float	Não
	gender	Identifica o género a quem se identifica o produto	1 char	Não
	quantityStock	Quantidade em stock do produto	Int	Não
	unityStock	Unidade da quantidade de stock	5 nchar	Não
OrdemFabrico	<u>ManufacturingNo</u>	Código único da ordem de fabrico	ManufacturingNo	Não
	CreationDate	Data de criação da ordem.	DateTime	Não
	StartDate	Data de início da ordem, quando foi iniciada a produção.	DateTime	Sim
	EndDate	Data de termino da ordem de produção	DateTime	Sim
	ExpectedEndDate	Data Esperada da conclusão da Ordem de Fabrico	DateTime	Não



	TotTimeExpected	Tempo Total esperado para a conclusão da Ordem de Fabrico	Int	Não
	TotCostSubContract	Custo total em SubContratados	Int	Não
	Priority	Prioridade da ordem de produção	smallInt	Não
	Status	Estado atual da ordem	char	Não
	Obs	Observações da ordem de Fabrico	250 nvarchar	Não

Tabela 8 - Entidades Fortes

Abaixo é apresentada a tabela com as chaves Primárias de cada uma dessas entidades

Entidade	PK	FK
Produto	ProductNo	-
Operacao	OpNo	-
Materia Prima	MatNo	-
SubContratado	SubContratadoNo	-
Ficha Técnica	FichTecNo	ProductNo
Ordem Fabrico	ManufacturingNo	-

Tabela 9 - Chaves Primarias e Estrangeiras, entidades Fortes

### 3.1.2 Entidades Fracas

As entidades fracas são o oposto das entidades fortes, pois este tipo de entidades para existirem ele necessitam da existência de outras entidades.

Nome Entidade	Atributos	Descrição	Tipo de Dado e Tamanho	Nulls
Etapa de Produção	<u>FichTecNo</u>	Código da Ficha Técnica	FichTecNo	Não
	<u>OpNo</u>	Código da operação	OpNo	Não
	<u>MaterialNo</u>	Código do Material Utilizado na ficha Tecnica	MatNo	Não
	<u>Type</u>	Tipo do Material Utilizado na Etapa de Produção Interna	TypeMaterial	Não
Etapa Ordem de Fabrico	<u>ManufacturingNo</u>	Código da ordem de fabrico	ManufacturingNo	Não
	<u>FichTecNo</u>	Código da Ficha Técnica	FichTecNo	Não
	<u>OpNo</u>	Código da Operação	OpN	Não

	QuantityMaterialOp	Quantidade de material utilizado na Operação ou do material final	Int	Não
	QuantityLostOp	Quantidade de material Perdida na operação ou do material final	Int	Não
	<u>OpSubNo</u>	Código Único da operação subContratada	SubContratadoNo	Sim
	QuantityMaterialSub	Quantidade de material	Int	Sim
	QuantiryReceivedSub	Quantidade de material recebido do subcontratado	Int	Sim
	QuantityLostSub	Quantidade de material Perdido	Int	Sim
	PriceSub	Preço do subContratado	Float	Sim
	<u>MaterialNo</u>	Código do Material	MatNo	Não
	<u>Type</u>	Tipo do Material	TypeMaterial	Não
	TotQuantity	Quantidade Total	Int	Não
	TotQuantityLost	Quantidade Total perdida	Int	Não
	Priority	Prioridade da etapa	Int	Sim
	TimeInMinutes	Tempo estimado da etapa	Int	Sim
	Status	Estado da etapa	1 nchar	Sim
	Ordem	Ordem em que a operação é executada	int	Sim

Tabela 10 - Entidades Fracas

Abaixo é possível ver a tabelas com as chaves primárias e chaves estrangeiras de cada uma destas entidades.

Entidade	PK	FK
Etapa Producao	FichTecNo, OpNo, MatNo	FichTecNo
	Type	OpNo

		MatNo
Etapa Ordem	ManufacturingNo, FichTecNo, OpNo, MatNo, Type	ManufacturingNo
		FichTecNo, OpNo, MatNo, Type
		SubContratadoNo

*Tabela 11 - Chaves Entidades Fracas*

## 3.1.2 Relacionamentos

### 3.1.2.1 Relacionamentos 1 para 1

No nosso diagrama não obtivemos nenhum relacionamento deste tipo

### 3.1.2.2 Relacionamento 1 para muitos

Os relacionamentos de 1 para muitos que possuímos no nosso sistema são os seguintes

- Produto 1.1 -- 0..\*Ficha Técnicas
  - Entidade Pai: Produto
    - PK : ProductNo
  - Entidade Filho: Ficha Técnica
    - PK: FichTecNo
    - FK: ProductNo
- Ficha Técnica 1..1 – 1..\* Etapa Producao
  - Entidade Pai: Ficha Tecnica
    - PK: FichTecNo
  - Entidade Filho: Etapa Producao
    - PK: FichTecNo, OpNo, MaterialNo, Type
    - FK: FichTecNo
- Operacao 1..1 – 0..\* Etapa Producao
  - Entidade Pai: Operacao
    - PK: OpNo
  - Entidade Filho: Etapa Producao
    - PK: FichTecNo, OpNo, MaterialNo, Type
    - FK: OpNo
- Materia Prima 1..1 – 0..\* EtapaProducao
  - Entidade Pai: Materia Prima
    - PK: MatNo
  - Entidade Filho: Etapa Producao
    - PK: FichTecNo, OpNo, MaterialNo, Type
    - FK: MatNo

- SubContratado 0..1 – 0..\* EtapasOrdem
  - Entidade Pai: SubContratado
    - PK: SubContratadoNo
  - Entidade Filho: EtapasOrdem
    - PK: ManufacturingNo, FichTecNo, OpNo, MaterialNo, Type
    - FK: SubContratadoNo
- OrdemFabricao 1..1 – 1..\* Etapas Ordem
  - Entidade Pai: Ordem Fabricao
    - PK: ManufacturingNo
  - Entidade Filho: Etapas Ordem
    - PK: ManufacturingNo, FichTecNo, OpNo, MaterialNo, Type
    - FK: ManufacturingNo
- Etapa Ordem 1..1 – 1..\* Etapa Producao
  - Entidade Filho: Etapa Producao
    - PK: FichTecNo, OpNo, MaterialNo, Type
    - FK: FichTecNo
    - FK: MatNo
    - FK: OpNo
  - Entidade Filha: Etapa Ordem
    - PK: ManufacturingNo, FichTecNo, OpNo, MaterialNo, Type
    - FK: ManufacturingNo
    - FK: SubContratadoNo
    - FK: FichTecNo, OpNo, MaterialNo, Type

### 3.1.2.3 Relacionamento muitos para muitos

No nosso diagrama não obtivemos nenhum relacionamento deste tipo

## 3.2 Modelo de dados lógico resultante

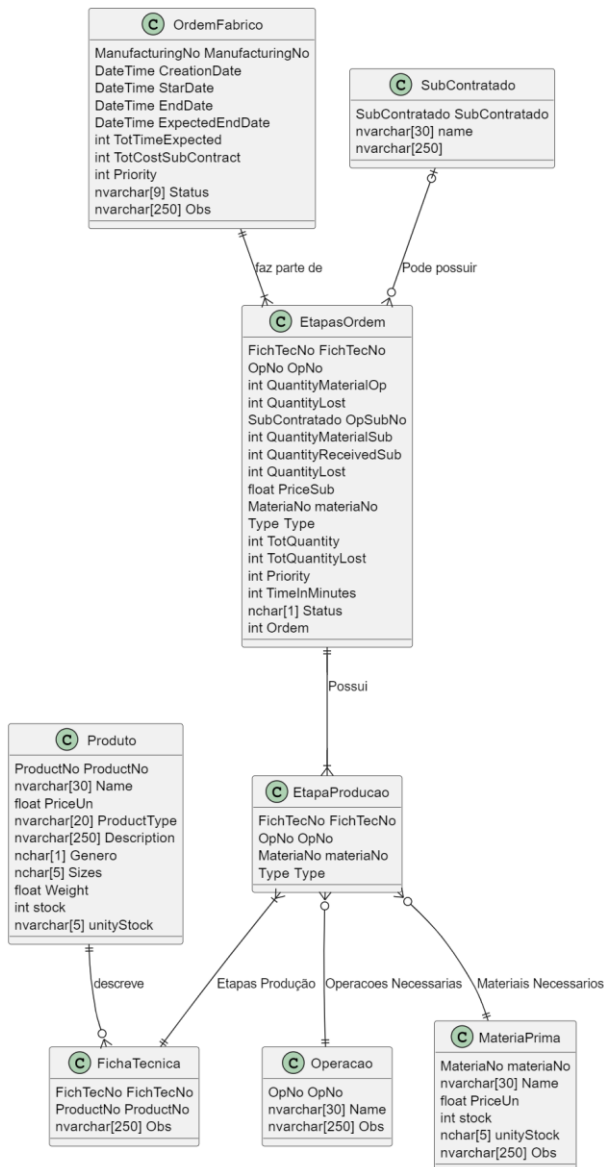


Figura 3 - Modelo Lógico Resultante

### 3.3 Validar modelo lógico com Normalização

#### 3.3.1 Normalização das Ordem de Fabrico

Na imagem abaixo é possível observar um exemplo de um documento de uma ordem de Fabrico

ORDEM DE FABRICO															
Nº Ordem		ORD1													
Data Emis:		10/05/2025													
Data Conc:		30/05/2025													
Quantidad		500													
Estado		Em Produção													
Total		0													
Produtos															
ProductNo	FichTecNo	Name	Obs	Material	Name	Tipo	OpNo	Name	Quantidad	SubContra	Name	Preco	Quantidad	PerdasTot:	Tempo
PROD1	FT1	Ficha Prod	FichaTec	MAT3	Pele Corta	Saída	OP1	Corte	12	SUB1	Corte Sub	20,99	4	12	4
				MAT1	Pele	Entrada			12				3		
				MAT2	Água	Entrada			2				2		

Figura 4 - Exemplo de uma Ordem de Fabrico

Legenda dos campos apresentados

Nº Ordem	A
Data Emissão	B
Data Conclusão	C
Quantidade	D
Estado	E
Total	F
ProductNo	G
FichTecNo	H
Name	I
Obs	J
Material	K
Name	L
Tipo	M
OpNo	N
Name	O
Quantidade	P
SubContratado	Q
Name	S
Preco	T
QuantidadeEnviada/Produzida	U
Tempo	V

Figura 5 - Legenda dos campos da Ordem de Fabrico

#### Forma Não Normalizada

Forma inicial do documento onde há redundância e possíveis grupos repetidos. Vamos normalizar isso passo a passo.

Sem Normalizar															
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
ORD1	10/05/2025	30/05/2025	500	Em Produção	12	PROD1	FT1	Ficha Prod	FichaTec	MAT3	Pele Cortada	Saída	OP1	Corte	12
										MAT1	Pele	Entrada			12
										MAT2	Agua	Entrada			2

Figura 6 - Forma Não Normalizada da Ordem de Fabrico

## 1º Forma Normal

Figura 6 - 1º FN Ordem de Fabrico

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
ORD1	10/05/2025	30/05/2025	500 Em Produção	12 PROD1	FT1	Ficha Prod	FichaTec	MAT3	Pele Cortada	Saida	OP1	Corte	12	SUB1	Corte Sub	20,99		4	12	4	
ORD1	10/05/2026	30/05/2026	500 Em Produção	12 PROD1	FT1	Ficha Prod	FichaTec	MAT1	Pele	Entrada	OP1	Corte	12			20,99		3	12	4	
ORD1	10/05/2027	30/05/2027	500 Em Produção	12 PROD1	FT1	Ficha Prod	FichaTec	MAT2	Agua	Entrada	OP1	Corte	2			20,99		2	12	4	
PX--> A, H, K, N, Q																					

Na 1º forma normal preenchemos a ficha técnica com todos os valores redundantes e identificamos uma PK, que nos permita identificar cada tuplo da ficha Técnica.

## 2º Forma Normal

Na 2º Forma iremos identificar as dependências parciais da PK detetada na 1FN

A -->	B	C	D	E	F
H -->	G	I	J		
K -->	L				
N -->	O				
H, K, M, N -->	P				
A, H, K, N, M -->	U	V	M	Q	

Figura 7 -2FN Ordem Fabrico Dependencia Parciais

Ordem Fabrico	A	B	C	D	E	F															
	PK --> A																				
Ficha Tecnica	H	G	I	J																	
	PK --> H																				
Material	K	L																			
	PK --> K																				
Operacao	N	O																			
	PK --> N																				
Etapas Ficha Tecnica	H	K	N	M	P																
	PK --> H, K, N, M																				
Etapas Ordem Fabrico	A	H	K	N	M	Q	R	S	T	U	V										
	PK --> A, H, K, N, M																				

Figura 8 - 2FN dependencias parciais detetadas

## 3º Forma Normal

Na 3º Forma Normal vamos tentar detetar as dependências transitivas do resultado da 2º FN e as FK de cada uma das dependências transitivas, caso tenham.

Ordem Fabrico	A	B	C	D	E	F															
	PK --> A																				
Ficha Tecnica	H	G	I	J																	
	PK --> H																				
	FK --> G																				
Material	K	L																			
	PK --> K																				
Operacao	N	O																			
	PK --> N																				
SubContratado	Q	R																			
	PK --> Q																				
Etapas Ficha Tecnica	H	K	N	M	P																
	PK --> H, K, N, M																				
	FK --> H																				
	FK --> K																				
	FK --> N																				
Etapas Ordem Fabrico	A	H	K	N	M	Q	U	V													
	PK --> A, H, K, N, M																				
	FK --> A																				
	FK --> H, K, N, M																				
	FK --> Q																				

Figura 9 - 3FN Ordem de Fabrico

### 3.3.2 Normalização da Ficha Técnica

Ficha Técnica						
FichaNo	FT1					
ProdutoNo	PROD1					
Produto	Sapatilha da Nike					
Preço UN	120,99					
Tamanho	41					
Conjunto de Operações e materiais						
OpNo	Nome	Obs	MatNo	Material	Tipo	
OP1	Corte	xxx	MAT1	Sola Cortada	Final	
			MAT1	Sola Cortada	Utilizado	
			MAT1	Água	Utilizado	

Figura 10 - Exemplo de uma Ficha Técnica

Legenda dos campos apresentados

FichaNo	A
ProdutoNo	B
Produto	C
Preço UN	D
Tamanho	E
OpNo	F
Nome	G
Obs	H
MatNo	I
Nome	J
Tipo	K
Obs	L

Figura 11 - Legenda dos campos

#### Forma Não Normalizada

Forma inicial do documento onde há redundância e possíveis grupos repetidos. Vamos normalizar isso passo a passo.

A	B	C	D	E	F	G	H	I	J	K	L
FT1	PROD1	Sapatilha da Nike	120,99	41	OP1	Corte	xxx	MAT1	Sola Corta	Final	as
								MAT2	Sola Corta	Utilizado	as
								MAT3	Água	Utilizado	as

Figura 12 - Forma Não Normalizada Ficha Técnica

#### 1º Forma Normal (1FN)

Na 1ª forma normal preenchemos a ficha técnica com todos os valores redundantes e identificamos uma PK, que nos permita identificar cada tuplo da ficha Técnica

A	B	C	D	E	F	G	H	I	J	K	L
FT1	PROD1	Sapatilha da Nike	120,99	41	OP1	Corte	xxx	MAT1	Sola Corta	Final	as
FT1	PROD1	Sapatilha da Nike	120,99	41	OP1	Corder	xxx	MAT2	Sola Corta	Utilizado	as
FT1	PROD1	Sapatilha da Nike	120,99	41	OP1	Corrte	xxx	MAT3	Água	Utilizado	as

PK --> A, F, I, K

Figura 13 - 1FN da Ficha Técnica

#### 2º Forma Normal (2FN)

Na 2ª Forma iremos identificar as dependências parciais da PK detetada na 1FN



	A --> B, C, D, E				
	F --> G, H				
	I --> J, L				
	A, F, I, K				
Ficha Técnica	<u>A</u>	B	C	D	E
	PK --> A				
Operacao	<u>F</u>	H			
	PK --> F				
Materia Prima	<u>I</u>	J			
	PK --> I				
Etapa de Produção	<u>A</u>	<u>F</u>	<u>I</u>	M	N
	PK --> A, F, I				

Figura 14 - 2ºFN da Ficha Técnica

### 3º Forma Normal

Na 3º Forma Normal vamos tentar detetar as dependências transitivas do resultado da 2º FN e as FK de cada uma das dependências transitivas, caso tenham.

3º FN						
Produto	<u>B</u>	C	D	E		
	PK --> B					
Ficha Técnica	<u>A</u>	B				
	PK --> A					
	FK --> B					
Operacao	<u>F</u>	H				
	PK --> F					
Materia Prima	<u>I</u>	J				
	PK --> I					
Etapa de Produção	<u>A</u>	<u>F</u>	<u>I</u>	K	M	N
	PK --> A, F, I					
	FK --> A					
	FK --> F					
	FK --> I					

Figura 15 - 3º FN da Ficha Técnica

### 3.3.3 Modelo Lógico após Normalização

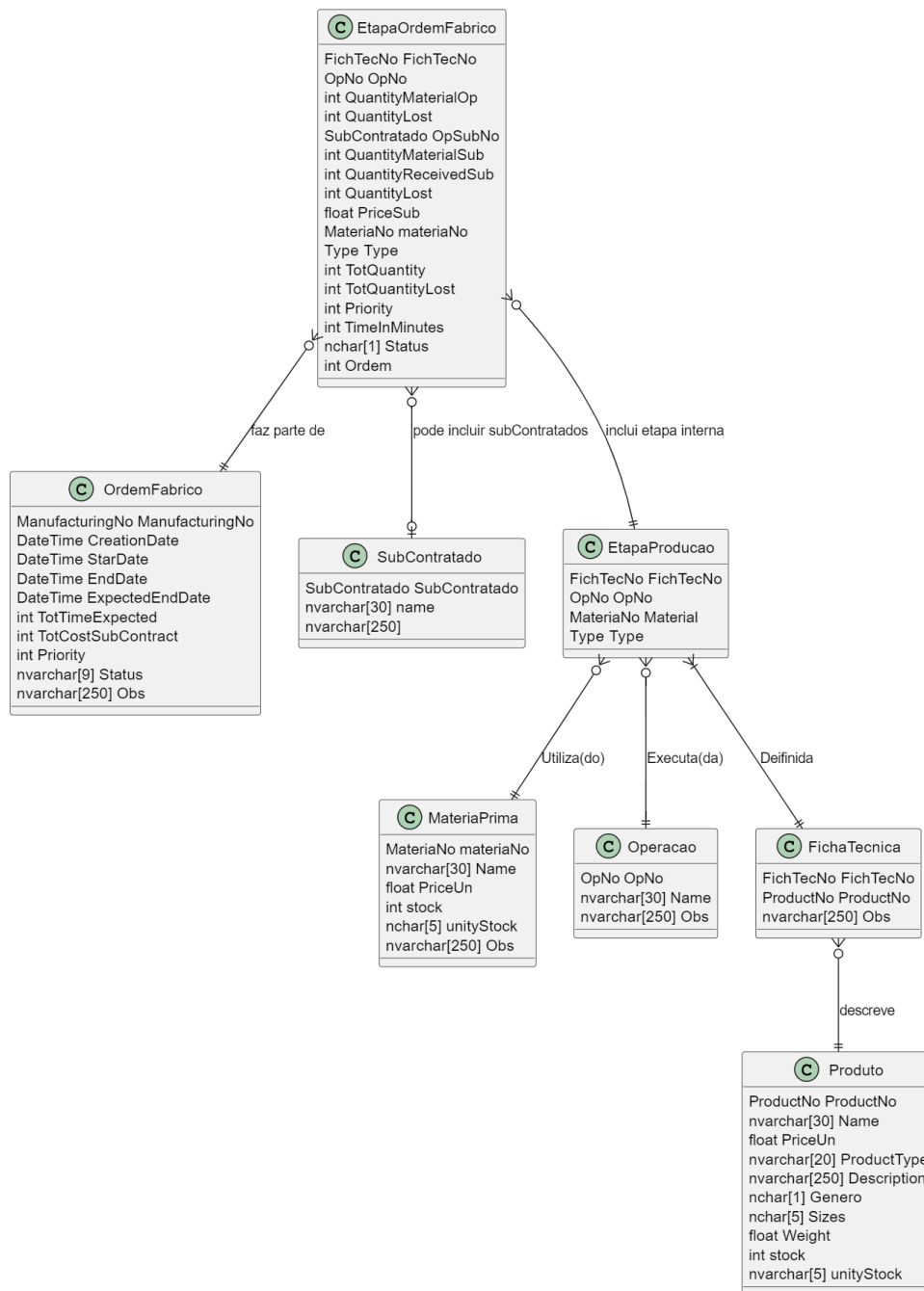


Figura 16 - Modelo Lógico após Normalização

### 3.4 Validação das transações com o modelo Lógico

Na imagem abaixo, é possível observar o nosso diagrama de transações. Embora a Ficha Técnica e o Produto não estejam diretamente envolvidos em nenhuma transação representada, ambos desempenham um papel fundamental no funcionamento do sistema.

O Produto é essencial, pois sem ele as ordens de fabrico perderiam o seu propósito — não haveria referência sobre o que deve ser produzido. Da mesma forma, a Ficha Técnica fornece os detalhes do processo de produção, como os materiais e operações necessárias para fabricar o produto. Sem essa informação, seria impossível executar corretamente a produção.

Assim, mesmo que não participem diretamente em transações específicas no diagrama, o Produto e a Ficha Técnica são elementos estruturais indispensáveis ao funcionamento coerente e completo do sistema.

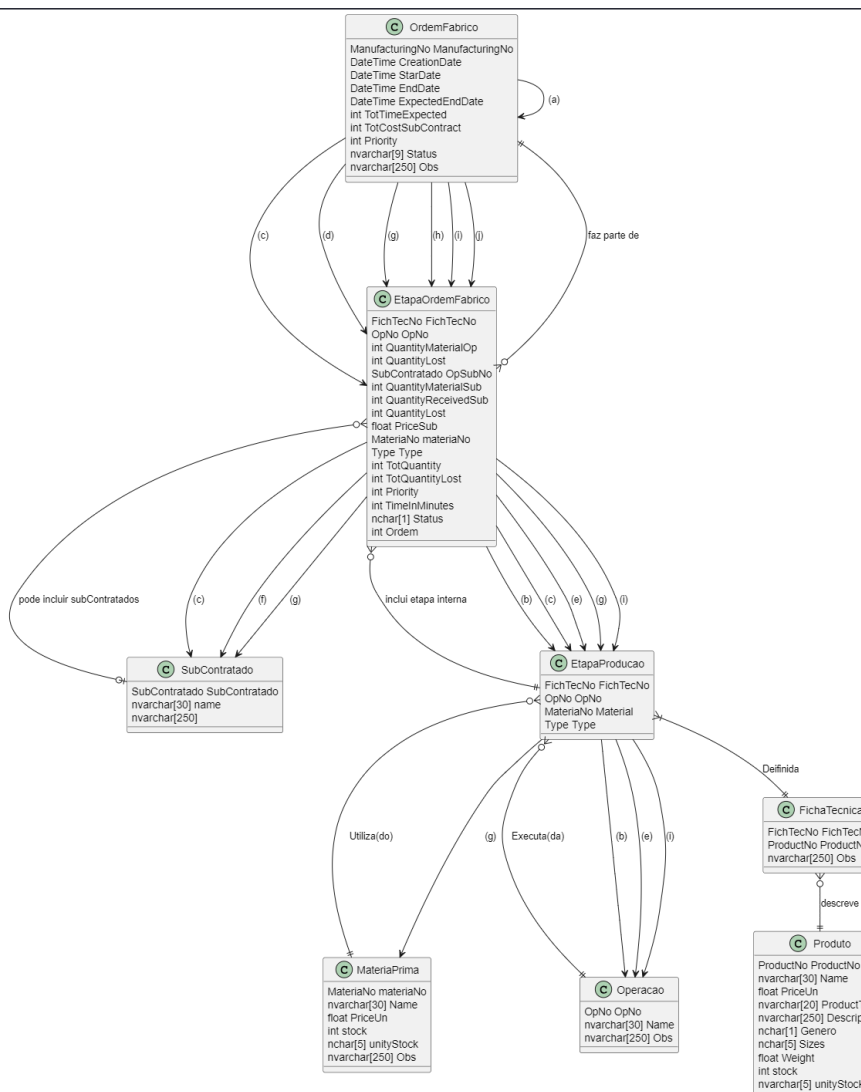


Figura 17 - Transações modelo lógico

Legenda do diagrama

- a → Listar Ordens em atraso com o respetivo tempo de atraso;
- b → Listar Operações Problemáticas;

- c → Custos Comparativos entre operação interna e externa;
- d → Consultar as Ordens de fabrico em curso com indicação do progresso;
- e → Relatório de operações com perdas superiores a 15%;
- f → Listar as Operações mais frequentemente subContratadas e respetivos custos;
- g → Determinar o custo total de produção de uma ordem;
- h → Consultar lista de operações pendentes para cada ordem, ordenadas por prioridade;
- i → Estimar o tempo total necessários para concluir uma ordem, com base no tempo médio das operações;
- j → Calcular a eficiência média de produção por período.

## 3.5 Identificação Restrições de Integridade dos dados

### 3.5.1 Matéria Prima

Atributos	Tipo de Dado e Tamanho	Restrições	Nulls
<u>MatNo</u>	MatNo	-	Não
Name	50 nvarchar	-	Não
PriceUn	float	Tem de ser superior ou igual a 0	Não
QuantityStock	Int	-	Não
unidadeStock	5 nchar	Só pode aceitar um destes valores: <ul style="list-style-type: none"> <li>• M</li> <li>• M^2</li> <li>• Peça</li> <li>• Kg</li> <li>• L</li> <li>• Un</li> </ul>	Não
Obs	250 nvarchar	-	Sim

### 3.5.2 Produto

Atributos	Tipo de dados e Tamanho	Restrições	Nulls
<u>ProductN</u>	ProductNo	-	Não
PriceUn	float	Tem de ser superior ou igual a 0	Não

Name	30 nvarchar	-	Não
productType	20 nvarchar	Tem de ser um dos seguintes valor: <ul style="list-style-type: none"> <li>• “Bota”,</li> <li>• “Sapatilha”,</li> <li>• “Chuteira”,</li> <li>• “Sandália”,</li> <li>• “Chinelo”,</li> <li>• “Outro”</li> </ul>	Não
Description	250 nvarchar	-	Sim
Sizes	5 nchar	-	Não
weight	float	Superior ou igual a 0	Não
gender	1 char	Só pode aceitar os valores: <ul style="list-style-type: none"> <li>• “M”;</li> <li>• “F”;</li> <li>• “U”</li> </ul>	Não
quantityStock	Int	-	Não
unityStock	5 nchar	Só pode aceitar um destes valores: <ul style="list-style-type: none"> <li>• Par</li> <li>• Un</li> <li>• Caixa</li> <li>• Lote</li> </ul>	Não

### 3.5.3 SubContratados

Atributos	Tipo de dados e Tamanho	Restrições	Nulls
<u>SubContratadoNo</u>	SubContratadoNo	-	Não
Name	30 nvarchar	Não pode haver dois subContratados com o mesmo nome <i>Unique Key</i>	Não
Obs	250 nvarchar	-	Sim

### 3.5.4 Operação

Atributos	Tipo de dados e Tamanho	Restrições	Nulls
<u>OpNo</u>	OpNo	-	Não
Name	30 nvarchar	Não pode haver duas operações com o mesmo nome <i>Unique Key</i>	Não
Obs	250 nvarchar		Sim

### 3.5.4 Ficha Técnica

Atributo	Tipo de dados e Tamanho	Restrições	Nulls
<u>FichTecNo</u>	FichTecNo	-	Não
ProductNo	ProductNo	-	Não
Obs	250 nvarchar	-	Sim

### 3.5.5 Ordem de Fabrico

Atributo	Tipo de dados e Tamanho	Restrições	Nulls
<u>ManufacturingNo</u>	ManufacturingNo	-	Não
CreationDate	DateTime	-	Não
StartDate	DateTime	Deve ser superior ou igual há data de criação e inferior ou igual as datas de fim	Sim
EndDate	DateTime	Deve ser superior ou igual as datas de criação e de começo. Deve ser inferior ou igual há data atual	Sim
ExpectedEndDate	DateTime	Deve ser superior há data de criação da Ordem.	Não

		Deve ser inferior ou igual há data criação	
TotTimeExpected	Int	Deve ser superior ou igual a 0	Não
TotCostSubContract	Int	Deve ser superior ou igual a 0	Não
Priority	smallInt	Pode assumir um destes valores: <ul style="list-style-type: none"> <li>• 1: Alta</li> <li>• 2: Medio</li> <li>• 3: Baixo</li> </ul>	Não
Status	char	Somente são permitidos os seguintes valores: <ul style="list-style-type: none"> <li>• P:“Pendente”,</li> <li>• E:“Execução”,</li> <li>• C:“Concluída”</li> </ul>	Não
Obs	250 nvarchar	-	Não

### 3.5.6 Etapa de Produção

Atributo	Tipo de dados e Tamanho	Restrições	Nulls
<u>FichTecNo</u>	FichTecNo	-	Não
<u>OpNo</u>	OpNo	-	Não
<u>MaterialNo</u>	MatNo	-	Não
<u>Type</u>	TypeMaterial	-	Não

### 3.5.7 Etapa da Ordem

Atributo	Tipo de dados e Tamanho	Restrições	Nulls
<u>ManufacturingNo</u>	ManufacturingNo	-	Não
<u>FichTecNo</u>	FichTecNo	-	Não
<u>OpNo</u>	OpN	-	Não

QuantityMaterialOp	Int	Tem de ser superior ou igual a 0.	Não
QuantityLostOp	Int	Tem de ser superior ou igual a 0	Não
<u>OpSubNo</u>	SubContratadoNo	-	Sim
QuantityMaterialSub	Int	Tem de ser Nulo quando a OpSubNo. Caso não seja nulo tem de ser superior ou igual a 0.	Sim
QuantiryReceivedSub	Int	Tem de ser Nulo quando a OpSubNo é nula. Quanto não for nulo não pode ser superior que a quantidade de material enviado	Sim
QuantityLostSub	Int	Tem de ser Nulo se o OpSubNo for nulo Caso não seja null tem de ser superior ou igual a 0	Sim
PriceSub	Float	Tem de ser Nulo se o OpSubNo for nulo Caso não seja null tem de ser superior ou igual a 0	Sim
<u>MaterialNo</u>	MatNo	-	Não
<u>Type</u>	TypeMaterial	-	Não
TotQuantity	Int	O valor deve tem de ser igual há soma dos materiais utilizados pelas operações + subContratados	Não
TotQuantityLost	Int	O valor deve tem de ser igual há soma das perdas das	Não



		operações + subContratados	
Priority	Int	Quando o tipo de material é 'E' tem de ser nulo.  Se não for nulo pode assumir um destes valores: <ul style="list-style-type: none"> <li>• 1 à Alta</li> <li>• 2 à Medio</li> <li>• 3 à Baixo</li> </ul>	Sim
TimeInMinutes	Int	Quando o tipo de material é 'E' tem de ser nulo.  Se não tem de ser superior ou igual a 0	Sim
Status	1 nchar	Quando o tipo de material é 'E' tem de ser nulo.  Se não pode ter 4 valores admissíveis: <ul style="list-style-type: none"> <li>• Pendente</li> <li>• Em Execução</li> <li>• Suspenso</li> <li>• Finalizado</li> </ul>	Sim
Ordem	int	Quando o tipo de material é 'E' tem de ser nulo.  Se não tem de ser superior ou igual a 0 e não pode ter valores repetidos, na mesma ficha técnica de uma ordem de fabrico	Sim

### 3.5.8 Restrições domínios

Domínio	Restrição
TypeMaterial	Só pode aceitar: E: "Entrada" S : "Saída"

## 3.6 Regras de Negócio

### Gestão Automatizada de Ordens de fabrico

- Quando uma ordem de fabrico é concluída (todo o material é recebido após a execução de todas as operações), o seu estado é atualizado
- Se o prazo de conclusão for ultrapassado, deve gerar um alerta, armazenando-o numa tabela específica de controlo

### Controlo de Produção

- Criar registos de produção para todas as operações da ficha técnica.
- Ao registar uma operação como concluída, deve calcular automaticamente as perdas
- Se as perdas forem superiores a 15%, deve registar um alerta de qualidade registando a ocorrência numa tabela específica para o efeito.
- Deve impedir o registo de quantidades recebidas superiores às enviadas

### Cálculos Automáticos

- Calcular o custo total de cada ordem considerando:
  - Operações internas (baseado em tempo estimado)
  - Operações subcontratadas (preço acordado)
- Determinar a eficiência média por operação e por período

### Validações de Processo

- Garantir que a sequência de operações é respeitada
- Validar que não há operações pendentes antes de concluir uma ordem

## **Relatórios e Consultas**

- Listar ordens atrasadas com o respetivo tempo de atraso
- Mostrar operações problemáticas (com maior índice de perdas)
- Apresentar custos comparativos entre produção interna e subcontratada

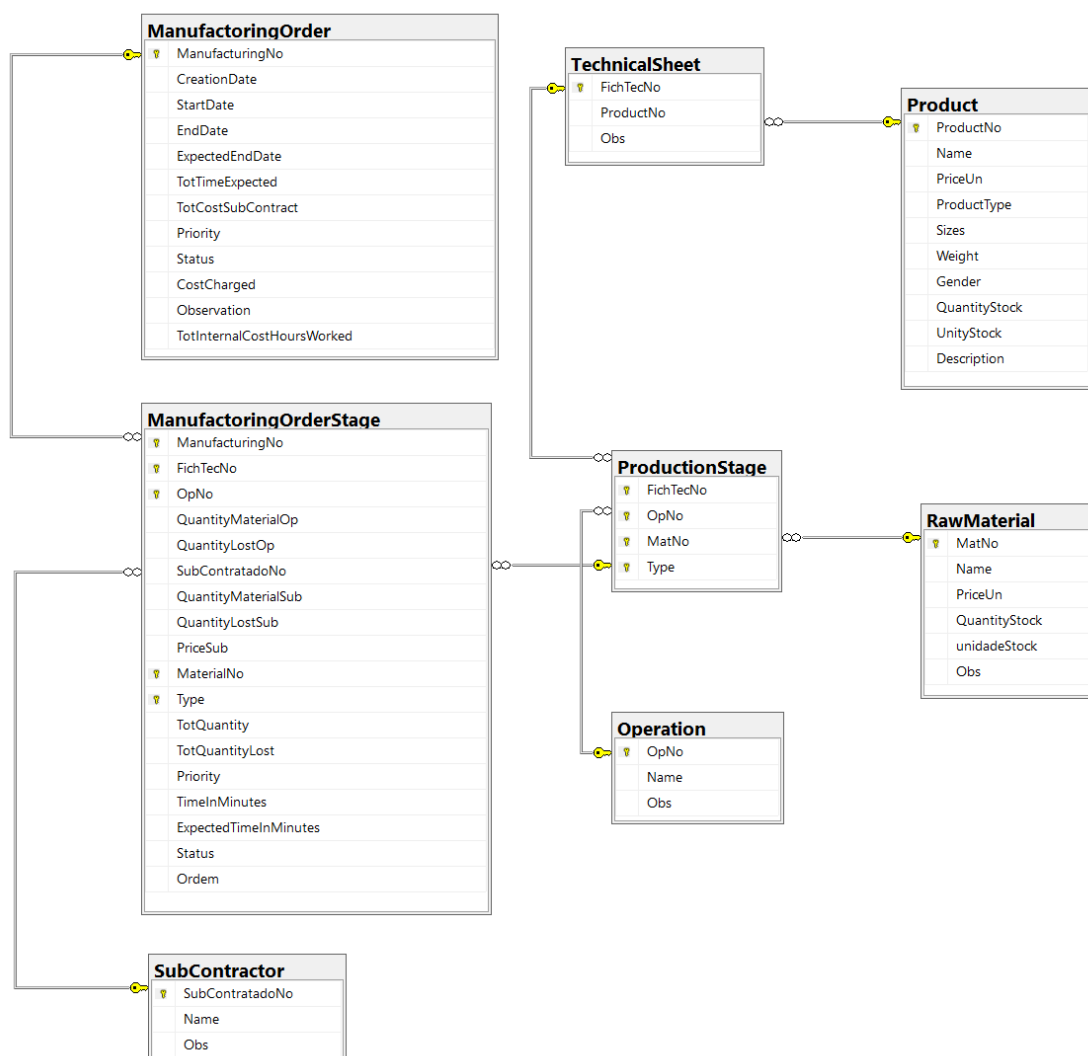
### **3.7 Verificação do provável crescimento futuro F**

Dado que este trabalho foi desenvolvido num contexto académico e responde exatamente aos requisitos do projeto, não se prevê um crescimento significativo para além do que foi solicitado. Ainda assim mantivemos o modelo suficientemente flexível para permitir acréscimos pontuais sem comprometer a estrutura já existente. Desta forma, caso surjam necessidades adicionais no futuro, será possível estender o esquema de dados com alterações mínimas e mantendo a coerência do sistema.

## 4. Construção do modelo físico

### 4.1 Criação das tabelas no SQL Server Management Studio

Após a finalização do modelo lógico passamos para o SQL onde começamos pela criação das tabelas e a realização do diagrama, como dá para observar na imagem abaixo



## 4.2 Procedures criadas para resolver os problemas de Negocio

### Criar registos de produção para todas as operações da ficha técnica

Para tratar deste requisitos criamos esta storage procedure que permite criar uma etapa para a produção de um produto da ordem de fabrico e indicar as quantidades utilizadas internamente e as que forma enviadas aos subcontratados.

```
ALTER PROCEDURE [dbo].[CREATE_MANUFACTURING_ORDER_STAGE]
(
    @ManufacturingNo dbo.ManufacturingNo,
    @FichTecNo dbo.FichTecNo,
    @OpNo dbo.OpNo,
    @QuantityMatOp INT,
    @MaterialNo dbo.MaterialNo,
    @Type dbo.TypeMaterial,
    @OpSubNo dbo.SubContratadoNo = NULL,
    @QuantityMatSub INT = 0,
    @PriceSub INT = NULL,
    @Prioridade INT = NULL,
    @Ordem INT = NULL,
    @ExpectedTimesInMinutes INT = 0
)
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;

        DECLARE
            @Now DATETIME = GETDATE();

        IF @OpSubNo IS NULL
        BEGIN
            IF @Type = 'S'
            BEGIN
                IF @Ordem IS NULL
                    THROW 51000, 'A ordem da Operacao não pode ser nula', 1;

                INSERT INTO ManufacturingOrderStage(ManufacturingNo, FichTecNo, OpNo, QuantityMaterialOp, MaterialNo, Type, Priority, Status, ExpectedTimeInMinutes, Ordem, TotQuantity)
                VALUES
                    (@ManufacturingNo, @FichTecNo, @OpNo, @QuantityMatOp, @MaterialNo, @Type, @Prioridade, 'P', @ExpectedTimesInMinutes, @Ordem, 0);
            END
        ELSE
        BEGIN
            INSERT INTO ManufacturingOrderStage(ManufacturingNo, FichTecNo, OpNo, QuantityMaterialOp, MaterialNo, Type, Priority, TotQuantity)
            VALUES
                (@ManufacturingNo, @FichTecNo, @OpNo, @QuantityMatOp, @MaterialNo, @Type, @Prioridade, @QuantityMatOp);
        END
    END
END
```

Figura 18 - 1º img Registos de produção

```
        ELSE
        BEGIN
            DECLARE
                @Total INT = @QuantityMatOp + @QuantityMatSub;

            IF @Type = 'S'
            BEGIN
                IF @Ordem IS NULL
                    THROW 51000, 'A ordem da Operacao não pode ser nula', 1;

                INSERT INTO ManufacturingOrderStage (ManufacturingNo, FichTecNo, OpNo, QuantityMaterialOp, MaterialNo, Type, QuantityMaterialSub, SubContratadoNo, PriceSub, Priority, Status, ExpectedTimeInMinutes, Ordem, TotQuantity)
                VALUES(@ManufacturingNo, @FichTecNo, @OpNo, @QuantityMatOp, @MaterialNo, @Type, @QuantityMatSub, @OpSubNo, @PriceSub, @Prioridade, 'P', @ExpectedTimesInMinutes, @Ordem, 0);
            END
        ELSE
        BEGIN
            INSERT INTO ManufacturingOrderStage (ManufacturingNo, FichTecNo, OpNo, QuantityMaterialOp, MaterialNo, Type, QuantityMaterialSub, SubContratadoNo, PriceSub, TotQuantity)
            VALUES(@ManufacturingNo, @FichTecNo, @OpNo, @QuantityMatOp, @MaterialNo, @Type, @QuantityMatSub, @OpSubNo, @PriceSub, @Total);
        END
    END

    COMMIT TRANSACTION;
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    DECLARE @ErrMsg NVARCHAR(4000) = ERROR_MESSAGE();
    RAISERROR('Erro ao criar etapa da Ordem de Fabrico: %s', 16, 1, @ErrMsg);
END CATCH;
```

Figura 19 - 2º img Registos de produção

### Finish Orde stage

Esta storage procedure além de permitir atualizar o estado de uma operação de uma ordem fabrico para concluído, que permite:

- antes de atualizar ver se a ordem de operações está a ser cumprida atualizar a ordem caso essa seja a última operação que falta concluir.

```

IF EXISTS (
    SELECT 1
    FROM ManufacturingOrderStage mo
    WHERE ManufacturingNo = @ManufacturingNo
    AND FichTecNo = @FichTecNo
    AND OpNo = @OpNo
    AND MaterialNo = @MaterialNo
    AND Type = @Type
    AND EXISTS (
        SELECT 1
        FROM dbo.ManufacturingOrderStage etapas_anteriores
        WHERE etapas_anteriores.ManufacturingNo = mo.ManufacturingNo
        AND etapas_anteriores.FichTecNo = mo.FichTecNo
        AND etapas_anteriores.Ordem < mo.Ordem
        AND etapas_anteriores.Status IS NULL
    )
)
BEGIN
    PRINT 'Erro: Não é possível registrar esta operação porque existem operações anteriores não realizadas.';
    ROLLBACK TRANSACTION;
    RETURN;
END

```

Figura 20 - verificação ordem das operacoes

- Verificar se essa operação teve perdas superiores a 15% e caso sim registrar um alerta de qualidade e registao numa tabela específica para o efeito, a tabela “error.SuperiorLost”;

```

DECLARE @TotQuantity INT

SELECT @TotQuantity = TotQuantity
FROM ManufacturingOrderStage
WHERE ManufacturingNo = @ManufacturingNo
AND FichTecNo = @FichTecNo
AND OpNo = @OpNo
AND MaterialNo = @MaterialNo
AND Type = @Type
AND (
    (@SubContratadoNo IS NULL AND SubContratadoNo IS NULL) OR
    (@SubContratadoNo IS NOT NULL AND SubContratadoNo = @SubContratadoNo)
)

DECLARE @PercentagemLost FLOAT =
    ((ISNULL(@QuantityLostSub, 0) + ISNULL(@QuantityLostOp, 0)) * 1.0) / NULLIF(@TotQuantity, 0)

IF @PercentagemLost > 0.15
BEGIN
    INSERT INTO error.SuperiorLost (
        ManufacturingNo, FichTecNo, OpNo, SubContratadoNo, MaterialNo, Type, QuantitatyLost
    )
    VALUES (
        @ManufacturingNo, @FichTecNo, @OpNo, @SubContratadoNo, @MaterialNo, @Type, @PercentagemLost
    )
END

```

Figura 21 - alertas de perdas superiores a 15%

- Soma as perdas totais dessa operação, considerando o total de todos as perdas dos seus materiais;

```

UPDATE mos_final
SET TotQuantityLost = TotQuantityLost + perdas.TotalPerdas
FROM ManufacturingOrderStage mos_final
INNER JOIN (
    SELECT
        mos.ManufacturingNo,
        mos.FichTecNo,
        mos.OpNo,
        SUM(ISNULL(mos.TotQuantityLost, 0)) AS TotalPerdas
    FROM ManufacturingOrderStage mos
    WHERE mos.ManufacturingNo = @ManufacturingNo
        AND mos.FichTecNo = @FichTecNo
        AND mos.OpNo = @OpNo
        AND mos.Type = 'E'
        AND ISNULL(mos.Type, '') != 'S'
    GROUP BY mos.ManufacturingNo, mos.FichTecNo, mos.OpNo) AS perdas
ON mos_final.ManufacturingNo = perdas.ManufacturingNo
AND mos_final.FichTecNo = perdas.FichTecNo
AND mos_final.OpNo = perdas.OpNo
AND mos_final.Type = 'S';

DECLARE @countStage INT;

SELECT @countStage = COUNT(stage.Status)
FROM ManufacturingOrderStage AS stage
WHERE stage.ManufacturingNo = @ManufacturingNo
AND stage.Status IN ('P', 'E');

```

Figura 22 - soma das perdas da operação

- Atualizar a ordem de fabrico dessa operação especifica, caso ela seja a última operação da mesma, por concluir.

```

DECLARE @countStage INT;

SELECT @countStage = COUNT(stage.Status)
FROM ManufacturingOrderStage AS stage
WHERE stage.ManufacturingNo = @ManufacturingNo
AND stage.Status IN ('P', 'E');

IF @countStage = 0
BEGIN
    UPDATE ManufacturingOrder
    SET
        Status = 'C',
        EndDate = GETDATE()
    WHERE ManufacturingNo = @ManufacturingNo
END

```

Figura 23 - Atualização automática da ordem

## 4.2 Criação das procedures de CUD (Create, Update, Delete) no SQL Server Management Studio

Para além das stored procedures que criamos para os requisitos especificados criamos um CUD para nos permitir gerir todas as tabelas da base de dados, que não iremos entrar em detalhes neste relatório, mas na imagem abaixo estão os nomes de todas as stored procedures que criamos.

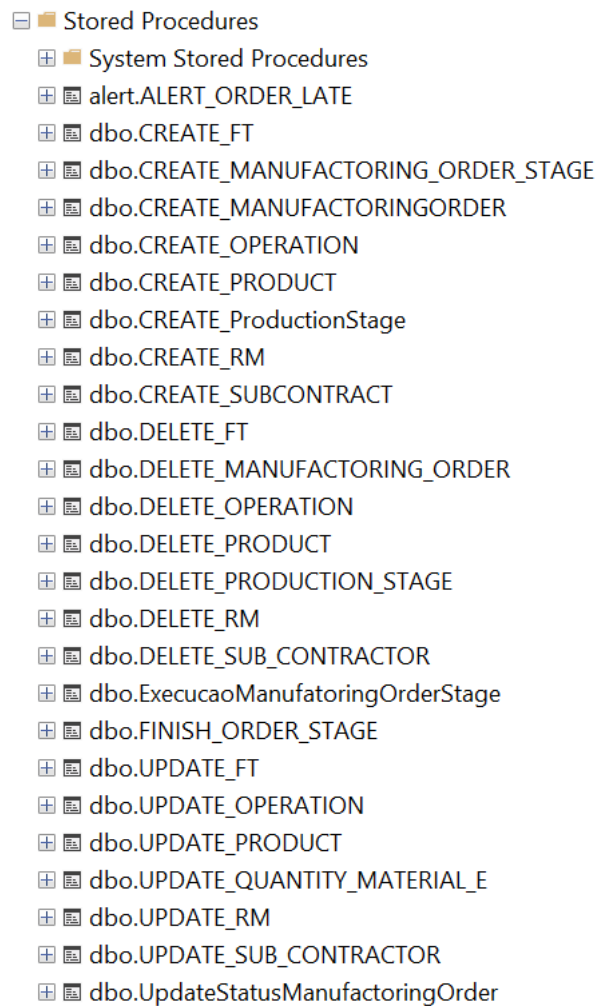


Figura 24 - Storage procedures criadas

## 4.3 Views

### Eficiência Media Produção Por Ano

```
SELECT DATEPART(YEAR, MO.EndDate) AS Ano, AVG(CAST(MOS.TotQuantity - MOS.TotQuantityLost AS DECIMAL(10, 4)) / NULLIF (MOS.TotQuantity, 0) * 100) AS EficienciaMedia
FROM    dbo.ManufacturingOrderStage AS MOS INNER JOIN
        dbo.ManufacturingOrder AS MO ON MOS.ManufacturingNo = MO.ManufacturingNo
WHERE   (MOS.TotQuantity > 0) AND (MOS.Status <> 'P') AND (MOS.TotQuantityLost >= 0)
GROUP BY DATEPART(YEAR, MO.EndDate)
```

Figura 25 - Eficiencia Media Producao por ano

	Ano	EficienciaMedia
1	2025	88.095238095238066

Figura 26 - Tabela da eficiencia media producao por ano



## Eficiência Media Produção Por Mês

```
SELECT DATEPART(MONTH, MO.EndDate) AS Mes, AVG(CAST(MOS.TotQuantity - MOS.TotQuantityLost AS DECIMAL(10, 4)) / NULLIF (MOS.TotQuantity, 0) * 100) AS EficienciaMedia
FROM    dbo.ManufacturingOrderStage AS MOS INNER JOIN
        dbo.ManufacturingOrder AS MO ON MOS.ManufacturingNo = MO.ManufacturingNo
WHERE   (MOS.TotQuantity > 0) AND (MOS.Status <> 'P') AND (MOS.TotQuantityLost >= 0)
GROUP BY DATEPART(MONTH, MO.EndDate)
```

Figura 27 - Eficiência Media Produção por Mês

	Mes	EficienciaMedia
1	5	88.095238095238066

Figura 28 - Tabela da Eficiência Media Produção por Mês

## Eficiência Media Produção Por Dia

```
SELECT DATEPART(WEEK, MO.EndDate) AS Dia, AVG(CAST(MOS.TotQuantity - MOS.TotQuantityLost AS DECIMAL(10, 4)) / NULLIF (MOS.TotQuantity, 0) * 100) AS EficienciaMedia
FROM    dbo.ManufacturingOrderStage AS MOS INNER JOIN
        dbo.ManufacturingOrder AS MO ON MOS.ManufacturingNo = MO.ManufacturingNo
WHERE   (MOS.TotQuantity > 0) AND (MOS.Status <> 'E') AND (MOS.TotQuantityLost >= 0)
GROUP BY DATEPART(WEEK, MO.EndDate)
```

Figura 29 - Eficiência Média Produção por dia

	Dia	EficienciaMedia
1	21	100.000000000000000

Figura 30 - Tabela Eficiência Média Produção por dia

## Média Perdas de Operação em Percentagem

```
SELECT O.OpNo, O.Name AS NomeOperacao,
       AVG(CAST(MOS.TotQuantityLost AS DECIMAL(10, 2)) / NULLIF (MOS.TotQuantity, 0) * 100) AS MediaPercentagemPerda
FROM    dbo.ManufacturingOrderStage AS MOS INNER JOIN
        dbo.ManufacturingOrder AS MO ON MOS.ManufacturingNo = MO.ManufacturingNo INNER JOIN
        dbo.Operation AS O ON MOS.OpNo = O.OpNo
WHERE   (MOS.Status = 'C')
GROUP BY O.OpNo, O.Name
```

Figura 31 - Média de Perdas de Operação em Perdas

## Operações pendentes Por Ordem

Nesta view como forma de contornar o facto de o SQL ignorar a clausula Order By nas view, utilizamos um TOP(100) PERCENT de maneira a mostrar todos os registos e forçar a ordenação.

```
SELECT TOP (100) PERCENT MO.ManufacturingNo AS NumeroOrdem, MO.Priority AS PrioridadeOrdem, MOS.OpNo AS CodigoOperacao,
       O.Name AS NomeOperacao, MOS.Priority AS PrioridadeOperacao, MOS.TotQuantity AS QuantidadeTotal, MOS.Status AS StatusOperacao
FROM    dbo.ManufacturingOrderStage AS MOS INNER JOIN
        dbo.ManufacturingOrder AS MO ON MOS.ManufacturingNo = MO.ManufacturingNo INNER JOIN
        dbo.Operation AS O ON MOS.OpNo = O.OpNo
WHERE   (MOS.Status = 'P') AND (MOS.Status IS NOT NULL)
ORDER BY NumeroOrdem, PrioridadeOrdem, PrioridadeOperacao
```

Figura 32 - Operações Por ordem

	NumeroOrdem	PrioridadeOrdem	CodigoOperacao	NomeOperacao	PrioridadeOperacao	QuantidadeTotal	StatusOperacao
1	M0001	1	OP004	Soldagem MIG	2	0	P

Figura 33 - Tabela das Operações por Ordem

## Custo de Produção da Ordem de Fabrico

```
SELECT mo.ManufacturingNo, SUM(COALESCE (mos.TimeInMinutes, 0)) AS TotTimeInMinutes,
SUM(COALESCE (mos.PriceSub, 0)) AS TotPaidToPartners, SUM(COALESCE (mos.TotQuantity, 0) * COALESCE (rm.PriceUn, 0)) AS TotalMaterialCost
FROM   dbo.ManufacturingOrder AS mo INNER JOIN
      dbo.ManufacturingOrderStage AS mos ON mo.ManufacturingNo = mos.ManufacturingNo INNER JOIN
      dbo.ProductionStage AS ps ON mos.FichTecNo = ps.FichTecNo AND mos.OpNo = ps.OpNo AND mos.MaterialNo = ps.MatNo
      AND mos.Type = ps.Type INNER JOIN
      dbo.RawMaterial AS rm ON ps.MatNo = rm.MatNo
GROUP BY mo.ManufacturingNo
```

Figura 34 - Custo de Produção da Ordem de Fabrico

	ManufacturingNo	TotTimeInMinutes	TotPaidToPartners	TotalMaterialCost
1	M0001	13333333	2312	359,25
2	M0002	0	2312	398,45

Figura 35 - Tabela do Custo de Produção da Ordem de Fabrico

## Custos Comparativos

```
WITH TempoTotalPorOrdem AS (SELECT ManufacturingNo, SUM(TimeInMinutes) AS TotalMinutos
FROM   dbo.ManufacturingOrderStage
GROUP BY ManufacturingNo)
SELECT mo.ManufacturingNo, mo.OpNo, o.Name AS NomeOperacao, mo.TimeInMinutes, mo.PriceSub,
(CAST(m.TotInternalCostHoursWorked AS FLOAT) / NULLIF (tp.TotalMinutos, 0)) AS CustoMinuto, mo.TimeInMinutes * (CAST(m.TotInternalCostHoursWorked AS FLOAT)
/ NULLIF (tp.TotalMinutos, 0)) AS CustolInterno, mo.PriceSub - mo.TimeInMinutes * (CAST(m.TotInternalCostHoursWorked AS FLOAT) /
NULLIF (tp.TotalMinutos, 0)) AS DiferencaCusto
FROM   dbo.ManufacturingOrderStage AS mo INNER JOIN
      dbo.ManufacturingOrder AS m ON mo.ManufacturingNo = m.ManufacturingNo INNER JOIN
      TempoTotalPorOrdem AS tp ON mo.ManufacturingNo = tp.ManufacturingNo LEFT OUTER JOIN
      dbo.Operation AS o ON mo.OpNo = o.OpNo
```

Figura 36 - Custos Comparativos

	ManufacturingNo	OpNo	NomeOperacao	TimeInMinutes	PriceSub	CustoMinuto	CustolInterno	DiferencaCusto
1	M0001	OP003	Dobra de Chapas	13333333	NULL	NULL	NULL	NULL
2	M0001	OP003	Dobra de Chapas	NULL	NULL	NULL	NULL	NULL
3	M0001	OP003	Dobra de Chapas	NULL	NULL	NULL	NULL	NULL
4	M0001	OP004	Soldagem MIG	NULL	2300	NULL	NULL	NULL
5	M0001	OP004	Soldagem MIG	NULL	12	NULL	NULL	NULL
6	M0002	OP003	Dobra de Chapas	NULL	NULL	NULL	NULL	NULL
7	M0002	OP003	Dobra de Chapas	NULL	NULL	NULL	NULL	NULL
8	M0002	OP003	Dobra de Chapas	NULL	NULL	NULL	NULL	NULL
9	M0002	OP004	Soldagem MIG	NULL	2300	NULL	NULL	NULL
10	M0002	OP004	Soldagem MIG	NULL	12	NULL	NULL	NULL

Figura 37 - Tabela dos Custos Comparativos

## Ordens Atrasadas

```
SELECT ManufacturingNo, Priority, ExpectedEndDate, Status
FROM   dbo.ManufacturingOrder
WHERE (ExpectedEndDate < GETDATE()) AND (Status <> 'C')
```

Figura 38 - Ordens Atrasadas

	ManufacturingNo	Priority	ExpectedEndDate	Status
1	M0001	1	2025-01-01 00:00:00.000	P
2	M0005	3	2025-05-22 17:03:11.483	E

Figura 39 - Tabela das Ordens Atrasadas

## Perdas Superiores a 15%

```
SELECT ManufacturingNo, FichTecNo, OpNo, MaterialNo, Type, SUM((COALESCE (QuantityLostSub, 0) + QuantityLostOp) * 1.0 / TotQuantity) AS PercentagemPerdas
FROM   dbo.ManufacturingOrderStage
GROUP BY ManufacturingNo, FichTecNo, OpNo, MaterialNo, Type
HAVING (SUM((COALESCE (QuantityLostSub, 0) + QuantityLostOp) / TotQuantity) > 0.15)
```

Figura 40 - Perdas Superiores a 15%

	ManufacturingNo	FichTecNo	OpNo	MaterialNo	Type	PercentagemPerdas
1	M0002	FT002	OP004	MAT14	S	1.071428571428

Figura 41 - Tabelas de Perdas Superiores a 15%

## Operações Problemáticas

```
SELECT MO.ManufacturingNo, MO.OpNo, O.Name AS NomeOperacao, CASE WHEN MO.QuantityMaterialOp > 0 THEN CAST(MO.QuantityLostOp AS FLOAT) / MO.QuantityMaterialOp ELSE NULL END AS IndicePerdaInterna,
CASE WHEN MO.QuantityMaterialSub > 0 THEN CAST(MO.QuantityLostSub AS FLOAT) / MO.QuantityMaterialSub ELSE NULL END AS IndicePerdaSubContratada
FROM   dbo.ManufacturingOrderStage AS MO LEFT OUTER JOIN
      dbo.Operation AS O ON MO.OpNo = O.OpNo
```

Figura 42 - Operações Problemáticas

	ManufacturingNo	OpNo	NomeOperacao	IndicePerdaInterna	IndicePerdaSubContratada
1	M0001	OP003	Dobra de Chapas	0,12	NULL
2	M0001	OP003	Dobra de Chapas	0	NULL
3	M0001	OP003	Dobra de Chapas	0	NULL
4	M0001	OP004	Soldagem MIG	0	NULL
5	M0001	OP004	Soldagem MIG	0	NULL
6	M0002	OP003	Dobra de Chapas	0	NULL
7	M0002	OP003	Dobra de Chapas	0	NULL
8	M0002	OP003	Dobra de Chapas	0	NULL
9	M0002	OP004	Soldagem MIG	0,8333333333333333	2,5
10	M0002	OP004	Soldagem MIG	0	NULL

Figura 43 - Tabela das Operações Problemáticas

## Operações SubContratadas

```
SELECT MOS.OpNo, O.Name AS NomeOperacao, COUNT(*) AS TotalSubcontratacoes, SUM(ISNULL(MOS.QuantityMaterialSub, 0) * ISNULL(MOS.PriceSub, 0)) AS CustoTotal
FROM   dbo.ManufacturingOrderStage AS MOS INNER JOIN
      dbo.Operation AS O ON MOS.OpNo = O.OpNo
WHERE  (MOS.SubContratadoNo IS NOT NULL)
GROUP BY MOS.OpNo, O.Name
```

Figura 44 - Operações SubContratadas

	OpNo	NomeOperacao	TotalSubcontratacoes	CustoTotal
1	OP004	Soldagem MIG	4	9320

Figura 45 - Tabelas Operacoes SubContratadas

## Ordens Atrasadas

```
SELECT ManufacturingNo, ExpectedEndDate, EndDate, Status, DATEDIFF(DAY, ExpectedEndDate, ISNULL(EndDate, GETDATE())) AS DiasAtraso
FROM   dbo.ManufacturingOrder AS MO
WHERE  (EndDate IS NULL) AND (ExpectedEndDate < GETDATE()) OR
      (EndDate IS NOT NULL) AND (EndDate > ExpectedEndDate)
```

Figura 46 - Ordens Atrasadas

	ManufacturingNo	ExpectedEndDate	EndDate	Status	DiasAtraso
1	M0001	2025-01-01 00:00:00.000	2025-05-19 16:02:03.993	P	138

Figura 47 - Tabelas das Ordens Atrasadas

## Ordens Fabrico Progresso

```
SELECT mo.ManufacturingNo, SUM(mos.TotQuantity) AS QuantidadeMaterialProduzido,
      SUM(mos.QuantityMaterialOp + COALESCE (mos.QuantityMaterialSub, 0)) AS TotalMaterial,
      ROUND(100.0 * SUM(mos.TotQuantity) / SUM(mos.QuantityMaterialOp + COALESCE (mos.QuantityMaterialSub, 0)), 2) AS ProgressoPercentual
FROM   dbo.ManufacturingOrder AS mo LEFT OUTER JOIN
      dbo.ManufacturingOrderStage AS mos ON mos.ManufacturingNo = mo.ManufacturingNo
WHERE  (mos.Type = 'S')
GROUP BY mo.ManufacturingNo
```

Figura 48 - Ordens de Fabrico Progresso

	ManufacturingNo	QuantidadeMaterialProduzido	TotalMaterial	ProgressoPercentual
1	M0001	100	114	87.720000000000
2	M0002	114	114	100.000000000000

Figura 49 - Tabela Progresso das Ordens de Fabrico

## Tempo total estimado por ordem

```
SELECT MO.ManufacturingNo, SUM(ISNULL(MOS.ExpectedTimeInMinutes, 0)) AS TempoTotalEstimado_Minutos
FROM   dbo.ManufacturingOrder AS MO INNER JOIN
      dbo.ManufacturingOrderStage AS MOS ON MO.ManufacturingNo = MOS.ManufacturingNo
GROUP BY MO.ManufacturingNo
```

Figura 50 - Tempo total estimado por ordem

	ManufacturingNo	TempoTotalEstimado_Minutos
1	M0001	24
2	M0002	24

Figura 51 - Exemplo do funcionamento da vista Tempo total estimado por ordem

## 4.4 Triggers

Para realizar o requisito “Calcular o custo total de cada ordem” criamos um trigger onde sempre que a ordem é atualizada é realizado o calculo automático dos seus custos internos e externos

```

ALTER TRIGGER [dbo].[TotalCostCalc]
ON [dbo].[ManufacturingOrder]
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    -- Atualiza os custos apenas para ordens cujo Status foi alterado para 'C'
    UPDATE mo
    SET
        mo.TotCostSubContract = ISNULL(subs.TotalSubContract, 0),
        mo.TotInternalCostHoursWorked = ISNULL(ints.TotalInternalCost, 0)
    FROM dbo.ManufacturingOrder mo
    INNER JOIN inserted i ON mo.ManufacturingNo = i.ManufacturingNo
    INNER JOIN deleted d ON d.ManufacturingNo = i.ManufacturingNo
    LEFT JOIN (
        -- Subcontratação
        SELECT
            ManufacturingNo,
            SUM(ISNULL(PriceSub, 0)) AS TotalSubContract
        FROM dbo.ManufacturingOrderStage
        WHERE SubContratadoNo IS NOT NULL AND PriceSub IS NOT NULL
        GROUP BY ManufacturingNo
    ) subs ON mo.ManufacturingNo = subs.ManufacturingNo
    LEFT JOIN (
        -- Custo interno
        SELECT
            ManufacturingNo,
            SUM(ISNULL(TimeInMinutes, 0) / 60.0 * 5) AS TotalInternalCost
        FROM dbo.ManufacturingOrderStage
        GROUP BY ManufacturingNo
    ) ints ON mo.ManufacturingNo = ints.ManufacturingNo
    WHERE i.Status = 'C' AND d.Status <> 'C';
END;

```

Figura 52 - Calculos custos da ordem

## Validações operações pendentes Ordem

Para cumprir com a requisito “Validar que não há operações pendentes antes de concluir uma ordem”, decidimos criar um Trigger onde sempre que o status da Ordem é atualizado ele verifica se não existem operações por concluir caso haja não deixa com que a ordem seja atualizada como concluída.

```

ALTER TRIGGER [dbo].[Validate_PendeteOp_StageManufacturingOrder]
ON [dbo].[ManufacturingOrder]
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION
        IF NOT UPDATE(Status)
            RETURN;

        SELECT 1
        FROM inserted i
        JOIN dbo.ManufacturingOrderStage st ON st.ManufacturingNo = i.ManufacturingNo
        WHERE i.Status = 'C'
            AND st.Status <> 'C'
            AND st.Status IS NOT NULL
        IF (@@ROWCOUNT > 0)
        BEGIN
            RAISERROR('Não é possível alterar o status. Existem operações não concluídas.', 16, 1);
            ROLLBACK TRANSACTION
        END

        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION
        PRINT 'Surgiu um erro no Trigger para validar a alteração do status da Ordem';
    END CATCH
END

```

Figura 53- Validação Status da Ordem

## Cálculo do custo total da ordem de produção de uma ordem

```
GO
/***** Object: Trigger [dbo].[TotalCostCalc]    Script Date: 23/05/2025 19:42:45 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[TotalCostCalc]
ON [dbo].[ManufacturingOrder]
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    -- Atualiza os custos apenas para ordens cujo Status foi alterado para 'C'
    UPDATE mo
    SET
        mo.TotCostSubContract = ISNULL(subs.TotalSubContract, 0),
        mo.TotInternalCostHoursWorked = ISNULL(ints.TotalInternalCost, 0)
    FROM dbo.ManufacturingOrder mo
    INNER JOIN inserted i ON mo.ManufacturingNo = i.ManufacturingNo
    INNER JOIN deleted d ON d.ManufacturingNo = i.ManufacturingNo
    LEFT JOIN (
        -- Subcontratação
        SELECT
            ManufacturingNo,
            SUM(ISNULL(PriceSub, 0)) AS TotalSubContract
        FROM dbo.ManufacturingOrderStage
        WHERE SubContratadoNo IS NOT NULL AND PriceSub IS NOT NULL
        GROUP BY ManufacturingNo
    ) subs ON mo.ManufacturingNo = subs.ManufacturingNo
    LEFT JOIN (
        -- Custo interno
        SELECT
            ManufacturingNo,
            SUM(ISNULL(TimeInMinutes, 0) / 60.0 * 5) AS TotalInternalCost
        FROM dbo.ManufacturingOrderStage
        GROUP BY ManufacturingNo
    ) ints ON mo.ManufacturingNo = ints.ManufacturingNo
    WHERE i.Status = 'C' AND d.Status <> 'C';
END;
```

Figura 54 - Trigger para cálculo automatizado do custo total da ordem de produção de uma ordem

## Verificação de possibilidade de alterar o status de uma ordem de produção

```
USE [FabricaSapatos]
GO
/***** Object: Trigger [dbo].[Validate_PendeteOp_StageManufacturingOrder]    Script Date: 23/05/2025 19:50:36 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER TRIGGER [dbo].[Validate_PendeteOp_StageManufacturingOrder]
ON [dbo].[ManufacturingOrder]
AFTER UPDATE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION
        IF NOT UPDATE(Status)
            RETURN;

        SELECT 1
        FROM inserted i
        JOIN dbo.ManufacturingOrderStage st ON st.ManufacturingNo = i.ManufacturingNo
        WHERE i.Status = 'C'
            AND st.Status <> 'C'
            AND st.Status IS NOT NULL
        IF (@@ROWCOUNT > 0)
        BEGIN
            RAISERROR('Não é possível alterar o status. Existem operações não concluídas.', 16, 1);
            ROLLBACK TRANSACTION
        END

        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION
        PRINT 'Surgiu um erro no Trigger para validar a alteração do status da Ordem';
    END CATCH
END
```

Figura 55 - Trigger que verifica se é possível alterar o status de uma ordem de produção para concluída verificando se todas as etapas dessa ordem já estão concluídas

## 4.5 SQL Server Agent

Para cumprir a regra de negócio “Se o prazo de conclusão for ultrapassado, deve gerar um alerta, armazenando-o numa tabela específica de controlo”, decidimos utilizar o SQL Server Agent, onde criamos um job que executa a stored procedure que procura por todas as ordens de fabricas atrasadas, ou seja, as ordens cujo o estado não seja concluído e o dia atual seja superior à data de conclusão.

```

ALTER PROCEDURE [alert].[ALERT_ORDER_LATE]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    BEGIN TRY
        BEGIN TRANSACTION
        INSERT INTO alert.LateManufacturingOrder (ManufacturingNo, AlertDate, ExpectedEndDate, Status, Priority)
        SELECT
            mo.ManufacturingNo,
            GETDATE(),
            mo.ExpectedEndDate,
            mo.Status,
            mo.Priority
        FROM dbo.ManufacturingOrder AS mo
        WHERE GETDATE() > mo.ExpectedEndDate AND mo.Status <> 'C';
        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() <> 2627
            THROW;
    END CATCH
END

```

Figura 56 - Storage Procedure ALERT\_ORDER\_LATE

E guarda essas Ordens na tabela de alertas. Como pode ver na imagem abaixo.

	IdAlert	ManufacturingNo	AlertDate	ExpectedEndDate	Status	Priority
1	1	M0001	2025-05-21 16:07:00.760	2025-01-01 00:00:00.000	P	1
2	2	M0001	2025-05-21 16:46:27.307	2025-01-01 00:00:00.000	P	1
3	3	M0001	2025-05-21 16:46:57.427	2025-01-01 00:00:00.000	P	1
4	1002	M0001	2025-05-22 23:00:00.860	2025-01-01 00:00:00.000	P	1
5	1003	M0005	2025-05-22 23:00:00.860	2025-05-22 17:03:11.483	E	3
6	1004	M0001	2025-05-23 10:33:32.090	2025-01-01 00:00:00.000	P	1
7	1005	M0005	2025-05-23 10:33:32.090	2025-05-22 17:03:11.483	E	3

Figura 57 - Tabela de alertas

Para criar esse job, começamos por definir-lhe um nome.



Job Properties - AlertLateManufacturingOrders

Select a page

- General
- Steps
- Schedules
- Alerts
- Notifications
- Targets

Script ? Help

Name: AlertLateManufacturingOrders

Owner: sa

Category: [Uncategorized (Local)]

Description: No description available.

Connection

Server: e3810e0d103c

Connection: sa

[View connection properties](#)

Progress

Ready

☒ Enabled

Source:

Created: 20/05/2025 19:38:51

Last modified: 21/05/2025 12:33:23

Last executed: 23/05/2025 10:33:31

[View Job History](#)

OK Cancel

Figura 58 - Nome do job

De seguida atribuímos um novo passo a esse job, que é a execução do storage Procedure.

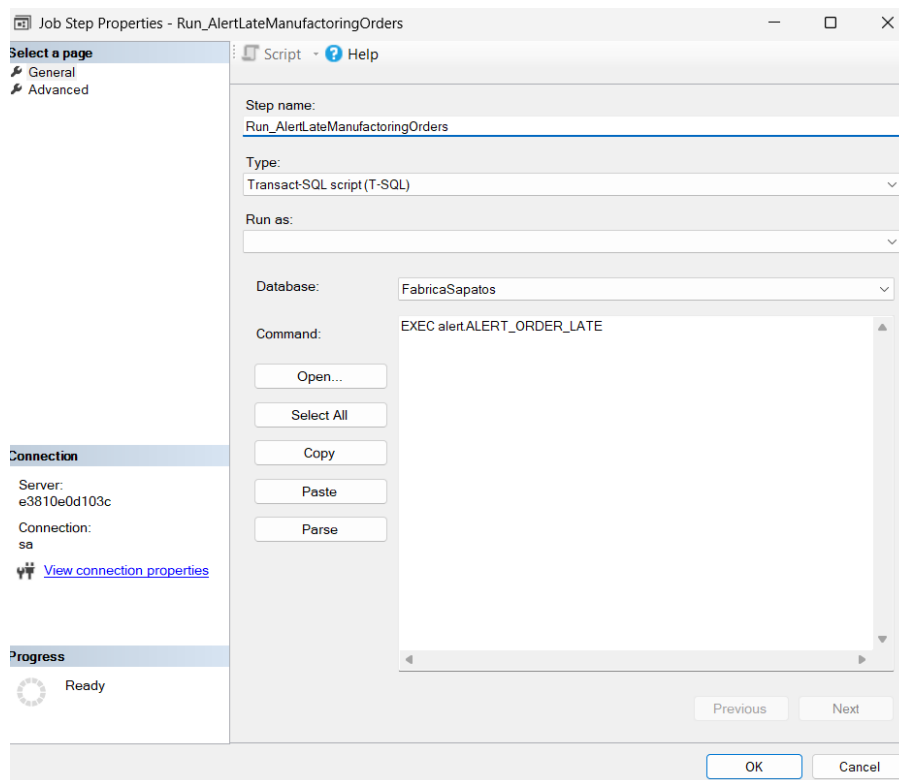


Figura 59 - Etapa do job

Por fim programamos a data em que o job vai ser executado, onde optamos por definir que ele vai ser executado todos os dias as 23:00:00.

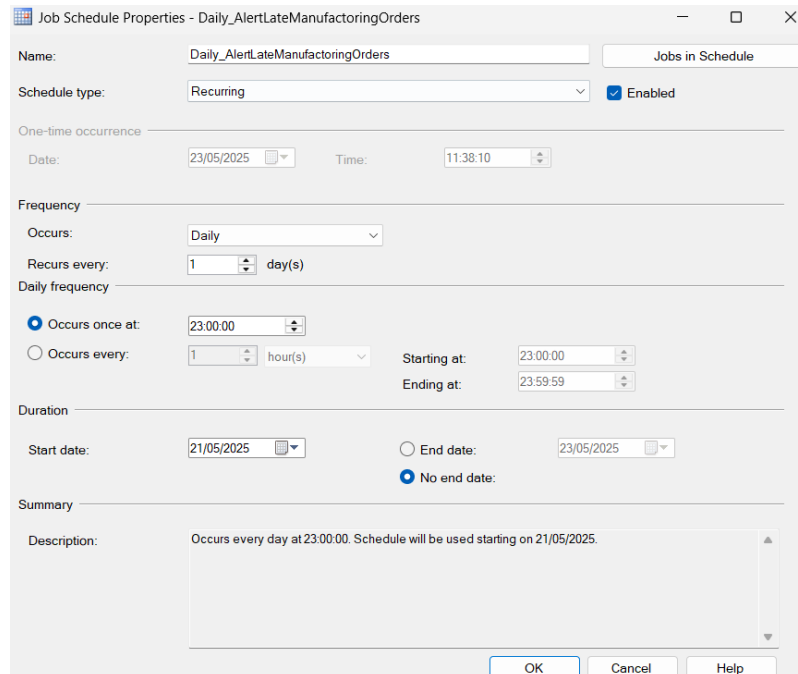


Figura 60 - Programação do Agente

## 5. Conclusões e Trabalho Futuro

O desenvolvimento deste trabalho permitiu abordar de forma integrada e sistemática o desafio da criação de uma base de dados para a gestão de ordens de produção numa fábrica de calçado, respondendo aos requisitos complexos do setor. Assim ao longo do projeto, foi possível aplicar metodologias de análise, modelação e implementação de bases de dados relacionais, com destaque no controlo da qualidade, eficiência das operações e automatização de processos.

Uma vez concluído o trabalho e aplicados todos os conceitos adquiridos durante o semestre na unidade curricular de Base de Dados foi possível obter uma visão mais prática de como é, e como deve funcionar, uma base de dados e todas as etapas da sua criação. Posto isto, também é possível verificar que existem alguns aspetos que devem ser melhorados de maneira que se possa satisfazer totalmente a base de dados em questão.

Entre os principais pontos fortes do trabalho destacam-se:

- **Cobertura dos requisitos do enunciado:** Todos os requisitos de negócio foram analisados e implementados, incluindo mecanismos automáticos para controlo de atrasos, perdas excessivas, validações de processo e cálculos de custos detalhados.
- **Metodologia seguida:** Seguiu-se uma abordagem estruturada, desde a análise de requisitos, passando pela modelação conceptual, normalização e validação, até à implementação física em SQL Server, com documentação detalhada de todas as decisões e justificações.
- **Automatização e controlo:** Foram implementados triggers, procedures e views para garantir o registo automático de alertas, a atualização de estados das ordens e a apresentação de indicadores de desempenho, contribuindo para a redução de erros e para a melhoria do processo de produção.

No entanto, alguns desafios e limitações foram identificados:

- **Complexidade do modelo:** A necessidade de refletir toda a realidade produtiva e de subcontratação resultou num modelo com múltiplas entidades e relacionamentos, exigindo um esforço adicional de validação e documentação para garantir a compreensibilidade e a manutenção futura.
- **Dependência de dados :** A validação do sistema com dados simulados revelou a importância de testar a base de dados com cenários reais e históricos, para ajustar restrições, regras de negócio e relatórios à realidade da fábrica.
- **Integração com outros sistemas:** A solução desenvolvida foca-se no registo e controlo das ordens de produção, mas a sua integração com sistemas de gestão de stocks,

compras ou vendas poderá ser um passo futuro para garantir uma visão ainda mais abrangente e integrada do negócio.

Em conclusão, o trabalho foi concluído satisfatoriamente, abrangendo todos os pontos propostos.

# **Bibliografia**

C. E. B. THOMAS M. CONNOLLY, DATABASE SYSTEMS -A Practical Approach to Design, Implementation, and Management, Pearson, 2015

## Referências WWW

[01] <https://stackoverflow.com/>