INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




# EGITRON Metrics - Solução para business intelligence (BI) do ecossistema EGITRON

**DANIEL REIS COELHO**
julho de 2024

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

# EGITRON METRICS

## Solução para Business Intelligence (BI) do ecossistema EGITRON

**Daniel Reis Coelho**

**Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Sistemas Computacionais**

**Orientador:** Alexandre Bragança

Porto, Junho 2024

# Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade.

Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Portanto, o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P. PORTO.

ISEP, Porto, 21 de julho de 2024

# Dedicatória

Dedico esta dissertação aos meus pais e irmão, pelo apoio incondicional e pela confiança depositada em mim ao longo de toda a minha jornada académica. Agradeço também aos meus amigos e colegas, pelo encorajamento e pela ajuda nos momentos de dificuldade. Aos meus professores, orientadores e empresa, pela orientação e pelos ensinamentos que foram essenciais para a concretização deste trabalho.

Esta conquista é um reflexo do suporte e carinho de todos vocês. Muito obrigado.

# Resumo

As empresas que possuem o software EGITRON Quality Control, da empresa EGITRON, um software que desempenha um papel central na gestão e controlo de qualidade de produtos em diversas unidades industriais, tanto a nível nacional quanto internacional, deparam-se com a impossibilidade de realizar uma análise abrangente e estratégica para otimizar os processos de produção e o desejo de aprimorar a tomada de decisões e impulsionar a inovação das empresas industriais. Devido a isso existe a necessidade de desenvolver uma solução, e a solução está diretamente relacionada com a utilização de um armazém de dados para armazenar de forma consolidada todo o tipo de informação que é possível extrair do processo produtivo diário das unidades industriais.

Um armazém de dados (DW), em vez de ser apenas um conceito, é um sistema projetado para armazenar as informações relacionadas às atividades de uma organização de maneira consolidada, funcionando como um ponto único de verdade para qualquer relatório ou análise a ser realizada. Ele possibilita a análise de grandes quantidades de informações que geralmente provêm dos sistemas transacionais da organização (OLTP).

Com o aumento crescente de dados, surgiu também a necessidade de os analisar. No entanto, os sistemas de armazém de dados atuais não têm a capacidade de lidar com a enorme quantidade de dados que está a ser produzida e precisa de ser gerenciada ou analisada. Nesta dissertação, é investigada a importância e os objetivos da criação de um DW no caso em questão, analisadas diferentes arquiteturas e modelos dimensionais, e avaliadas diversas ferramentas. No decorrer do documento no que toca à implementação é detalhado o passo a passo do processo de ETL, desde a extração de dados de diversas fontes, passando pela transformação e limpeza, até ao carregamento no DW.

Os resultados obtidos com a implementação demonstram uma melhoria significativa na capacidade de análise e na qualidade das decisões a serem tomadas. A solução implementada permite uma atualização incremental eficiente dos dados, garante a integridade e segurança das informações, e oferece alta disponibilidade e facilidade de manutenção. Estes benefícios confirmam que o armazém de dados é uma solução eficaz para as necessidades das empresas que utilizam o software EGITRON Quality Control.

**Palavras-chave**: Data Warehouse, ETL, modelo dimensional

# Abstract

Companies that use the EGITRON Quality Control software, developed by EGITRON, face challenges in performing comprehensive and strategic analyses to optimize production processes and enhance decision-making and innovation in industrial enterprises. Due to this, there is a need to develop a solution directly related to the use of a data warehouse to consolidate all types of information that can be extracted from the daily production processes of industrial units.

A data warehouse (DW), rather than being just a concept, is a system designed to store information related to an organization's activities in a consolidated manner, serving as a single point of truth for any report or analysis to be conducted. It enables the analysis of large amounts of information that typically come from the organization's transactional systems (OLTP).

With the increasing volume of data, the need to analyze it has also grown. However, current data warehouse systems are not capable of handling the enormous amount of data being produced that needs to be managed or analyzed. This dissertation investigates the importance and objectives of creating a DW in this specific case, analyzes different architectures and dimensional models, and evaluates various tools. Throughout the document, the implementation process is detailed step by step, covering the ETL process from data extraction from various sources, through transformation and cleaning, to loading into the DW.

The results obtained from the implementation demonstrate a significant improvement in the capacity for analysis and the quality of decision-making. The implemented solution allows for efficient incremental data updates, ensures the integrity and security of information, and offers high availability and ease of maintenance. These benefits confirm that the data warehouse is an effective solution for the needs of companies using the EGITRON Quality Control software.

**Keywords**: Data Warehouse, ETL, dimensional model

# Agradecimentos

Em primeiro lugar, agradeço aos meus pais e irmão, que me apoiaram em todos os momentos, sempre disponíveis para me ajudar, ouvir e aconselhar, nunca deixando de acreditar nas minhas capacidades. Agradeço também aos meus colegas de mestrado e a todos os meus amigos que me acompanharam desde o primeiro ano da licenciatura até ao presente, por me apoiarem e ajudarem a superar todas as adversidades do meu percurso académico.

Um agradecimento especial à minha entidade patronal, EGITRON, e aos colegas e amigos de profissão, pela compreensão e facilidade com que me permitiram conciliar o mestrado com a minha vida profissional. Agradeço igualmente à instituição ISEP, pela transmissão de conhecimentos adquiridos no Mestrado em Sistemas Computacionais.

Por fim, expresso a minha gratidão ao meu orientador, Alexandre Bragança, por todo o suporte prestado. A sua disponibilidade para a revisão do trabalho, bem como as críticas construtivas e sugestões fornecidas, foram cruciais para o resultado e qualidade da dissertação final.

# Índice

**1 Introdução ..... 1**

1.1 Interpretação analítica, crítica e ética ..... 1
1.1.2 Interpretação analítica ..... 2
1.1.3 Interpretação crítica ..... 3
1.1.4 Interpretação ética ..... 3

1.2 Contexto ..... 4

1.3 Problema ..... 5

1.4 Objetivos ..... 6

1.5 Processo de pesquisa ..... 7
1.5.1 Métodos de Pesquisa ..... 7
1.5.2 Questões de Pesquisa ..... 8
1.5.3 Critérios de Inclusão e Exclusão ..... 8

1.6 Plano de Trabalho ..... 9

1.7 Estrutura do Documento ..... 11

**2 Software EGITRON Quality Control ..... 13**

2.1 Contexto Software ..... 13

2.2 EQC – Ponto Atual ..... 16

2.3 EQC – Necessidades / Pretensões ..... 17

2.4 Análise de Visão da Solução ..... 18

2.5 Sumário ..... 20

**3 Estado da Arte ..... 23**

3.1 Data Warehouse ..... 24
3.1.1 Data Warehouse vs Dados Operacionais ..... 25

3.2 Modelação de Dados ..... 27
3.2.1 A Importância da Modelação de Dados ..... 27
3.2.2 Técnicas de Modelação de Dados ..... 28

3.3 Arquiteturas Data Warehouse ..... 33
3.3.1 Contexto ..... 33
3.3.2 Enterprise Data Warehouse ..... 34
3.3.3 Data Mart Independente ..... 35
3.3.4 Data Mart Dependente ..... 36

3.4 DW Tradicional versus DW na Cloud ..... 36
3.4.1 Análise Comparativa ..... 37
3.4.2 Discussão / Análise ..... 38

3.5 Ferramentas Data Warehouse ..... 39
3.5.1 Snowflake ..... 39
3.5.2 Amazon Redshift ..... 41
3.5.3 Discussão / Análise ..... 43
3.6 ETL e ELT ..... 44
3.6.1 ETL e ELT: Definições ..... 44
3.6.2 ETL e ELT: Comparação de processos ..... 44
3.6.3 Escolher entre ETL e ELT ..... 45
3.6.4 Ferramentas ETL ..... 46
3.6.5 Discussão / Análise ..... 47
3.7 Sumário ..... 48
**4 Desenvolvimento ..... 49**
4.1 Requisitos Funcionais e Não Funcionais ..... 50
4.1.1 Requisitos Funcionais ..... 50
4.1.2 Requisitos Não Funcionais ..... 50
4.2 Casos de Uso ..... 51
4.3 Análise de Alternativas de Design ..... 53
4.3.1 Apache Nifi com Data Warehouse no Azure ..... 53
4.3.2 Talend com Data Warehouse on-premises ..... 53
4.3.3 Apache Nifi com Google BigQuery ..... 54
4.3.4 Comparação e Seleção da Alternativa ..... 54
4.4 Diagrama de Componentes da Nova Solução ..... 55
4.5 Principais Processos da Solução ..... 56
4.6 Especificação do Sistema ..... 58
4.6.1 Endpoints ..... 58
4.6.2 Apache Nifi ..... 60
4.6.3 Data Warehouse ..... 60
4.7 Implementação ..... 61
4.7.1 Data Warehouse ..... 61
4.7.2 Apache Nifi ..... 63
4.7.3 Procedimento ..... 66
4.8 Implementação e Colocação em Produção do Apache Nifi ..... 74
4.8.1 Escolha da Infraestrutura de Produção ..... 75
4.8.2 Preparação do Ambiente de Produção ..... 75
4.8.3 Instalação do Apache Nifi no Servidor ..... 75
4.8.4 Deploy dos Fluxos de Dados ..... 76
4.8.5 Monitorização e Manutenção ..... 76
4.8.6 Discussão / Análise ..... 76
4.9 Sumário ..... 76
**5 Avaliação da Solução ..... 77**
5.1 Objetivos da Avaliação ..... 77
5.2 Metodologia de Avaliação ..... 78

5.3 Resultados da Avaliação ..... 78
5.3.1 Comparação de Dados Entre o Data Warehouse e Base de dados de Produção 79
5.3.2 Complexidade das Consultas ..... 82
5.3.3 Execução dos Processos de Validação ..... 83
5.3.4 Análise de Desempenho ..... 83
5.3.5 Validação de Endpoints ..... 85
5.3.6 Discussão / Análise ..... 86
5.3.7 Sumário ..... 87

**6 Conclusão ..... 89**

6.1 Considerações Finais ..... 89

6.2 Melhorias Futuras ..... 90
6.2.1 Suporte para Múltiplas Bases de Dados de Produção Simultâneas ..... 91
6.2.2 Desacoplamento das Ligações Diretas às Bases de Dados de Produção ..... 91
6.2.3 Automatização e Monitorização Avançada ..... 91
6.2.4 Otimização da Performance do Data Warehouse ..... 92

6.3 Sumário ..... 92

# Lista de Figuras

Figura 1 - Diagrama de *Gantt* do cronograma e das tarefas do projeto ..... 9
Figura 2 - EQC - Exemplos de ensaios pré-definidos ..... 14
Figura 3 - EQC - Exemplo de configuração de um modelo de controlo ..... 15
Figura 4 - EQC - Exemplo de um relatório ..... 15
Figura 5 - EQC - Exemplo de um relatório de controlo de qualidade em PDF ..... 16
Figura 6 - EQC - Planeamento da visão da solução do problema ..... 20
Figura 7 - EQC - Fluxo de dados para o Data Warehouse (Schio, 2006) ..... 25
Figura 8 - Um modelo E/R típico ..... 28
Figura 9 - Um esquema em estrela / modelo dimensional (Ballard, 2012) ..... 30
Figura 10 - Estrutura tabela de factos ..... 32
Figura 11 - Tipos de modelos dimensionais ..... 33
Figura 12 - Arquiteturas de um Data Warehouse (Ballard, 2012) ..... 34
Figura 13 - Arquitetura Data Warehouse empresarial (Ballard, 2012) ..... 35
Figura 14 - Arquitetura Data Mart independente (Ballard, 2012) ..... 36
Figura 15 - Ilustração arquitetura Snowflake (Snowflake, 2023) ..... 40
Figura 16 - Ilustração arquitetura Amazon Redshift (AWS, 2023) ..... 42
Figura 17 - Diagrama de Componentes ..... 55
Figura 18 - Processo de Extração de Dados ..... 56
Figura 19 - Processo de Extração e Limpeza de Dados ..... 57
Figura 20 - Processo de Carregamento no Data Warehouse ..... 57
Figura 21 - Swagger UI ..... 59
Figura 22 - Diagrama de fluxos do Apache Nifi ..... 60
Figura 23 - Diagrama da base de dados ..... 61
Figura 24 - Diagrama Resumo do Fluxo Implementado ..... 67
Figura 25 - Conexão a Base De Dados ..... 68
Figura 26 - Após obtenção da última atualização ..... 68
Figura 27 - Consulta SQL para obter unidades industriais ..... 69
Figura 28 - Inserção ou atualização de uma unidade industrial ..... 69
Figura 29 - JSON obtido do endpoint metrics/report ..... 70
Figura 30 - Filtragem de Dados do Relatório ..... 71
Figura 31 - Dois Caminhos a Seguir Para Preencher Estatísticas ..... 72
Figura 32 - Implementação para preencher tabela temporária ..... 72
Figura 33 - Tabela temporária preenchida ..... 73
Figura 34 - Preenchimento da Tabela De Factos ..... 74
Figura 35 - Tabela de Factos Preenchida ..... 74
Figura 36 - Comparação de Relatórios ..... 79
Figura 37 - Comparação de Modelos de Controlo ..... 80
Figura 38 - Comparação de Produtos ..... 80
Figura 39 - Validação de Campos de Dados ..... 81
Figura 40 - Validação de Tabela de Facto ..... 81

Figura 41 - Validação de Integridade Referencial ..... 82
Figura 42 - Performance - DW vs. Produtivo ..... 84

# Lista de Tabelas

Tabela 1 - Base de Dados Operacional vs Data Warehouse ..... 26
Tabela 2 - DW tradicional vs. DW na cloud (Rehman, 2018) ..... 38

# Acrónimos e Símbolos

## Lista de Acrónimos

| | |
|---|---|
| **BI** | Business Intelligence |
| **PRISMA** | Preferred Reporting Items for Systematic Reviews and Meta-Analyses |
| **EQC** | EGITRON Quality Control |
| **IBM** | International Business Machines Corporation |
| **DW** | Data Warehouse |
| **ETL** | Extract Transform Load |
| **ELT** | Extract Load Transform |
| **OLTP** | Online Transaction Processing |
| **OLAP** | Online Analytic Processing |
| **E/R** | Entidade-Relacionamento |
| **EDW** | Enterprise data warehouse |
| **TI** | Tecnologia da Informação |
| **MPP** | Processamento Massivamente Paralelo |
| **SSL** | Secure Sockets Layer |
| **HTTPS** | Hyper Text Transfer Protocol Secure |

# 1 Introdução

Esta dissertação de mestrado foi elaborada no âmbito da unidade curricular DIMEI do segundo ano do Mestrado em Engenharia Informática do Instituto Superior de Engenharia do Porto (ISEP). O presente documento constitui uma dissertação que visa desenhar e implementar um protótipo de uma solução de data warehouse no contexto do software EGITRON Quality Control.

Neste capítulo, será apresentada uma visão geral dos fundamentos teóricos e metodológicos que guiarão a análise deste estudo. Inicialmente, são abordados os diferentes tipos de interpretação - analítica, crítica e ética, fornecendo uma base para a compreensão das diversas perspetivas e abordagens. Em seguida, o problema em estudo é contextualizado, delineando os objetivos e justificando a sua relevância. É descrito o processo de pesquisa adotado, incluindo os métodos utilizados, as questões que norteiam a investigação e os critérios de inclusão e exclusão. Por fim, são apresentados o plano de trabalho e a estrutura do documento, orientando sobre como o estudo está organizado e o que esperar dos capítulos subsequentes.

## 1.1 Interpretação analítica, crítica e ética

Esta secção aborda de forma abrangente a interpretação analítica, crítica e ética no contexto de uma dissertação sobre o software EQC em empresas industriais. A análise analítica destaca a lacuna entre a geração de dados e sua aplicação efetiva, atendendo às necessidades dos *stakeholders*. A descrição clara do problema central, relacionado à falta de uma solução de

Business Intelligence (BI), evidencia a capacidade de comunicar os desafios enfrentados pelas empresas.

A interpretação crítica destaca a importância da conformidade legal e ética na escolha de plataformas de implementação, examinando a falta de aproveitamento integral dos dados do EQC. A análise ética é evidente na consideração dos impactos sociais e na preocupação com a legalidade e responsabilidade social.

Em suma, a secção estabelece uma base sólida para a dissertação, integrando análises analíticas, críticas e éticas para compreender as complexidades do problema e as suas implicações.

### 1.1.2 Interpretação analítica

Ao considerar os *stakeholders* e seus interesses, a dissertação aborda de maneira abrangente as necessidades das empresas industriais que fazem uso do software EQC. O foco na lacuna entre a geração de dados e a sua efetiva utilização demonstra uma atenção aos interesses dos gestores, operadores de chão de fábrica e demais partes interessadas envolvidas nas operações industriais.

A descrição clara do problema evidencia a capacidade em comunicar eficazmente os desafios enfrentados pelas empresas no contexto do software EQC. A falta de uma solução integral de Business Intelligence é apresentada como o cerne da questão, proporcionando um entendimento claro do problema central a ser abordado.

A consideração dos possíveis impactos sociais decorrentes das tentativas de resolver o problema é aparente na análise da necessidade de otimizar a utilização dos dados do EQC. Reconhece-se a importância não apenas para as empresas envolvidas, mas também para o aprimoramento das práticas no processo produtivo, sugerindo uma visão ampla dos impactos potenciais na sociedade.

As consequências de não resolver o problema são explicadas, destacando a subutilização de informações estratégicas e estatísticas fundamentais, prejudicando a eficiência operacional e limitando a capacidade de inovação das empresas. As contribuições esperadas da resolução do problema estão claramente delineadas, visando uma gestão mais informada e estratégica, promovendo a tomada de decisões baseada em dados sólidos e impulsionando a excelência operacional. A colaboração com um funcionário da empresa (gestor de projetos do software EQC) na estruturação dos dados dos endpoints do software EQC é fundamental para garantir a integridade e eficácia da análise. Esta colaboração limita-se única e exclusivamente ao escopo de trabalho descrito, assegurando que todas as etapas do processo respeitem as políticas internas de privacidade e segurança de dados da empresa. É permitido um acesso regular aos backups atualizados das bases de dados da empresa e a respetiva autorização para os trabalhar, desempenhando um papel fundamental na garantia da privacidade e integridade dos dados analisados nesta dissertação.

### 1.1.3 Interpretação crítica

Considerando as preocupações legais e sociais, reconhece-se a necessidade de uma plataforma robusta para análise de dados, respeitando as normas éticas e regulamentações vigentes no contexto industrial. A escolha cuidadosa das plataformas de implementação é apresentada como uma medida crítica para garantir alinhamento com tais normas.

A análise de múltiplas perspetivas é evidente na abordagem crítica da falta de solução de BI. São examinados os processos produtivos, identificadas áreas de melhoria e destacadas a necessidade de uma plataforma de análise de dados abrangente. A crítica estende-se à falta de aproveitamento integral dos dados do EQC, revelando uma abordagem analítica que procura identificar inconsistências e preconceitos na utilização dessas informações.

A base científica é reforçada pela análise de dados e estatísticas relacionadas ao problema. A geração massiva de dados é reconhecida como uma característica inerente à produção moderna, fundamentando a necessidade de uma solução abrangente de BI na indústria.

### 1.1.4 Interpretação ética

Ao avaliar os *stakeholders* e seus interesses, evidencia-se a consideração ética ao priorizar as necessidades das empresas e demais partes interessadas e inclui o compromisso com a proteção de dados pessoais e empresariais. A clareza na descrição do problema demonstra um compromisso ético em comunicar efetivamente os desafios enfrentados pelas empresas industriais. A consideração dos possíveis impactos sociais reflete uma abordagem ética, reconhecendo a responsabilidade de otimizar o uso dos dados do EQC não apenas para benefício das empresas, mas também para contribuir para práticas aprimoradas no processo produtivo em geral. A análise das consequências de não resolver o problema é guiada por uma preocupação ética, destacando os prejuízos à eficiência operacional e à capacidade de inovação.

A análise crítica e ética estende-se à preocupação com a legalidade e responsabilidade social, reforçando a importância de uma solução de BI alinhada às normas e regulamentações. O compromisso ético é evidente na procura por uma gestão mais informada e estratégica, respeitando as diretrizes éticas para a manipulação e armazenamento de dados sensíveis no contexto industrial. Não há uso de materiais de outros autores que envolva questões de direitos autorais, pois todo o conteúdo produzido será original e baseado nos dados fornecidos pela empresa e respetivo conhecimento obtido com o decorrer do desenvolvimento da dissertação.

## 1.2 Contexto

O surgimento do chão-de-fábrica após a Revolução Industrial marcou a transição das pequenas oficinas artesanais para a produção em larga escala nas indústrias. Antes, o conhecimento sobre a fabricação estava nas mãos dos artesãos, mas a industrialização separou esse conhecimento do processo produtivo. Isso deu origem ao proletariado e ao chão-de-fábrica como é conhecido hoje. Atualmente, o chão-de-fábrica é altamente organizado e tecnologicamente avançado, demandando profissionais qualificados. Apesar da geração massiva de dados durante a produção, muitas empresas não aproveitam totalmente essas informações. O uso de ferramentas de análises de dados e Business Intelligence (BI), oferece uma solução para processar e interpretar esses dados. Empresas como Oracle, IBM, Seagate e Microsoft disponibilizam softwares ajustáveis às necessidades, proporcionando às gestões do chão-de-fábrica informações valiosas para melhorar a qualidade, produtividade e satisfação do cliente. Business Intelligence (BI) tem como objetivo apoiar uma melhor tomada de decisões empresariais (Elena, 2011).

Na indústria atual, a gestão eficiente de dados é crucial para a tomada de decisões estratégicas. A EGITRON, empresa portuguesa de engenharia e automação industrial cujo principal objetivo é fornecer aos seus clientes soluções de excelência para o controlo de qualidade e inspeção dos seus produtos emprega o software EQC em diversas empresas, incluindo líderes nacionais como Amorim e Logoplaste, para otimizar processos, desde o controlo da matéria-prima até à expedição final. O EQC assume um papel central ao otimizar uma variedade de processos críticos. Desde o controlo rigoroso da receção de matéria-prima dos fornecedores até à expedição final para os clientes, o EQC demonstra a sua adaptabilidade e eficácia. Além disso, desempenha um papel vital a garantir que os padrões e requisitos específicos da engenharia de produtos sejam atendidos e aprovados durante o processo produtivo, solidificando o seu lugar como uma ferramenta integral ao longo de toda a cadeia produtiva (para maior detalhe sobre o EQC, ver capítulo 2).

No entanto, embora essas empresas tenham integrado com sucesso o EQC nas suas operações diárias, surge um desafio significativo na ausência de uma solução abrangente de Business Intelligence (BI). A capacidade de extrair *insights* estratégicos dos dados gerados pelo EQC é subutilizada, privando as organizações de uma análise profunda e estatísticas fundamentais para aprimorar a eficiência operacional e impulsionar a inovação. “Business Intelligence (BI) converte os dados em informação útil e, através da análise humana, em conhecimento” (Negash, 2004).

## 1.3 Problema

Apesar dos avanços tecnológicos na indústria contemporânea, observa-se uma lacuna considerável na maximização do potencial dos dados gerados durante o processo de produção. A geração massiva de dados, inerente à produção moderna, ainda não é plenamente explorada para aprimorar a eficiência operacional, qualidade e inovação. Mesmo com a presença da EGITRON, uma empresa dedicada à engenharia e automação industrial, e o uso eficaz do software EQC por empresas líderes como a Amorim e Logoplaste, depara-se com um desafio significativo devido à ausência de uma solução integral de Business Intelligence (BI). No âmago desta dissertação, destaca-se uma questão crucial que permeia as operações das empresas que fazem uso do software EQC: a disparidade entre a notável capacidade desse sistema em gerar dados significativos e a ausência de uma solução abrangente de BI. Esta lacuna compromete substancialmente a habilidade dessas organizações em extrair *insights* profundos e estatísticas essenciais para influenciar a tomada de decisões estratégicas. Para competir no mercado global de hoje, as empresas precisam de ter mais conhecimento do que antigamente e, ainda, para obter sucesso, elas precisam de saber mais sobre os seus clientes, mercados, tecnologias e processos, e precisam de ter essas informações antes dos seus concorrentes (Heinrichs, 2003).

O software EQC, desenvolvido pela EGITRON e analisado mais detalhadamente no capítulo 2, apresenta-se como uma ferramenta fundamental para o registo e controlo de ensaios de qualidade ao longo do processo produtivo. A sua capacidade altamente parametrizável, permitindo a criação de ensaios personalizados e a configuração de modelos de controlo específicos para cada cliente, destaca-se como uma ferramenta essencial no controlo de qualidade industrial. No entanto, a não integração efetiva dos dados gerados por esse software numa solução de BI abrangente resulta na subutilização de informações estratégicas cruciais. "O chão-de-fábrica gera hoje grande quantidade de dados que, por estarem dispersos ou desorganizados, não são utilizados em todo o seu potencial como fonte de informação" (Fortulan, 2005). A falta de uma plataforma robusta para análise de dados não apenas limita a compreensão aprofundada dos processos produtivos, mas também prejudica a efetiva utilização do potencial informacional proporcionado pelo EQC. Com a EGITRON a desempenhar um papel crucial no fornecimento de soluções de automação e engenharia industrial, a resolução desse problema emerge como uma necessidade imperativa. Capacitar empresas como Amorim e Logoplaste a superar essa disparidade entre a geração de dados e a sua transformação em conhecimento valioso torna-se crucial para promover uma tomada de decisão mais fundamentada e, consequentemente, impulsionar a competitividade no complexo cenário industrial contemporâneo.

Diante desse cenário desafiador, a proposta de um planeamento e implementação de um data warehouse eficiente e adaptável, integrando os dados do software EQC, surge como um passo fundamental para superar as limitações atuais. A necessidade de maximizar a eficiência e a inovação transcende fronteiras, considerando que o software EQC é amplamente utilizado em diversos pontos tanto a nível nacional quanto mundial. Essa solução é concebida com o

intuito de viabilizar uma análise profunda e estratégica, procurando otimizar os processos de produção, aprimorar a tomada de decisões e impulsionar a inovação nas empresas industriais.

Com o efetivo desenvolvimento do data warehouse, vislumbra-se a criação de uma visão abrangente e integrada dos dados ao longo da cadeia produtiva. Esta iniciativa permitirá não apenas o cruzamento global de dados entre diferentes indústrias à escala mundial, mas também facilitará a comparação entre unidades industriais. A capacidade de identificar discrepâncias e padrões de desempenho tornar-se-á uma ferramenta valiosa para aprimorar a eficiência operacional. Além disso, a análise de erros em relatórios ganhará uma nova dimensão, oferecendo a capacidade de comparar e identificar unidades industriais com maior ou menor incidência de falhas. O acompanhamento da evolução do produto ao longo dos anos proporcionará *insights* valiosos para contínuas melhorias, enquanto por exemplo a análise da variação do químico TCA em relação às condições climáticas permitirá uma compreensão mais profunda das influências ambientais nos resultados. A proposta de cruzamento de dados para identificação de possíveis falhas em diferentes etapas do processo produtivo representa uma abordagem proativa na gestão da qualidade. A integração holística de todas as informações, por sua vez, proporcionará uma compreensão global do desempenho, facilitando a identificação de áreas específicas que podem ser aprimoradas.

Assim, essa abordagem não apenas viabilizará uma gestão mais informada e estratégica, mas também permitirá uma tomada de decisões alicerçada em dados sólidos. O resultado almejado é a promoção da excelência operacional em todas as unidades industriais, impulsionando as empresas a enfrentar os desafios de um cenário industrial cada vez mais orientado por dados e inovação.

## 1.4 Objetivos

A presente dissertação tem como objetivo central o planeamento e desenvolvimento de uma solução, no caso, um data warehouse (DW), especificamente direcionado para empresas que fazem uso do software EQC. O mesmo visa preencher a lacuna identificada entre a geração massiva de dados do EQC e a subutilização dessas informações para aprimorar a eficiência operacional, qualidade e inovação nas operações de produção. O objetivo é desenhar e implementar um protótipo de data warehouse que integre os dados gerados pelo EQC, permitindo uma análise sistemática das operações de produção. Pretende-se desenvolver um processo de ETL (Extração, Transformação, Carregamento) que assegure a correta integração dos dados, garantindo a sua qualidade e consistência, e criar uma arquitetura de data warehouse robusta e escalável, adaptável às necessidades específicas de cada empresa. Além disso, será avaliado o desempenho da solução implementada, através de testes e validações que comparem os dados entre o data warehouse e as bases de dados de produção, analisando a complexidade das consultas e a eficiência do sistema. Com este objetivo, a dissertação procura não apenas desenvolver uma solução técnica, mas também proporcionar

uma compreensão aprofundada das melhores práticas e tecnologias relacionadas com data warehouses. Além disso, visa promover uma utilização estratégica dos dados gerados pelo EQC, permitindo às empresas obter insights valiosos que possam ser utilizados para tomar decisões informadas e estratégicas, impulsionando assim a inovação e a competitividade no setor industrial.

## 1.5 Processo de pesquisa

Para a realização desta dissertação, foi utilizada a metodologia PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses), amplamente reconhecida e utilizada em revisões sistemáticas de literatura. A metodologia PRISMA foi aplicada de forma a garantir a transparência, precisão e reprodutibilidade dos processos de pesquisa e seleção das fontes bibliográficas utilizadas na implementação de um Data Warehouse.

A pesquisa desempenha um papel crucial nesta investigação, centrada no desenvolvimento de uma solução de Data Warehouse (DW) orientada para o controlo de qualidade no processo industrial, especificamente para o software EQC. Esta dissertação procura estabelecer uma estrutura metodológica robusta que não apenas compreenda a complexidade da geração e utilização de dados industriais, mas também oriente a criação de soluções inovadoras para otimizar processos, melhorar a tomada de decisões e impulsionar a inovação nas empresas industriais.

Inicialmente, a pesquisa focará na explicação e compreensão das funcionalidades do software EQC, identificando os pontos-chave da obtenção de dados e a capacidade parametrizável do sistema. Será também realizada uma análise detalhada sobre como os dados são atualmente utilizados e quais as lacunas presentes na sua plena exploração para aprimorar a eficiência operacional.

Tendo em conta que se tem como alvo a maximização do potencial informacional do EQC, será feita uma análise mais específica sobre a integração de dados num data warehouse. Será analisado como os dados podem ser agregados de maneira eficiente e adaptável para possibilitar uma análise abrangente e estratégica.

### 1.5.1 Métodos de Pesquisa

O método de pesquisa utilizado inicia-se pela identificação da literatura relevante sobre ETL e Data Warehousing. Para isso, foram exploradas fontes variadas e específicas. Inicialmente, foram consultadas bases de dados acadêmicas reconhecidas, como Google Scholar, IEEE Xplore e ACM Digital Library (sendo Google Scholar a mais utilizada das três), visando artigos científicos e conferências relevantes na área de Data Warehousing, Business Intelligence e processos de ETL. Além disso, foi utilizada a plataforma *b-on* para acesso a periódicos científicos e a Elsevier para revisão de artigos de revisão e estudos de caso.

A fim de aprofundar a pesquisa, foram identificadas palavras-chave essenciais que abrangem os principais aspetos dos temas investigados, tais como "Data Warehouse", "Business Intelligence", "ETL", entre outras específicas para os subtópicos de interesse. A escolha e combinação adequada dessas palavras-chave foram fundamentais para direcionar a procura e refinar os resultados obtidos.

A seleção dos artigos para leitura foi baseada em critérios criteriosos, incluindo a relevância do conteúdo, a data de publicação mais recente para assegurar atualidade, e o enfoque em estudos empíricos, revisões sistemáticas e meta-análises. Priorizei trabalhos publicados após os anos 2000 para garantir a pertinência e a aplicabilidade dos dados e conclusões na atualidade.

Este processo de pesquisa estruturada e criteriosa não apenas possibilitou identificar lacunas e desafios na gestão de dados pelo EQC, mas também proporcionou uma base sólida para a formulação de questões de pesquisa precisas e relevantes, essenciais para o desenvolvimento deste estudo.

### 1.5.2 Questões de Pesquisa

A questão central desta pesquisa é compreender como é possível estruturar e integrar os dados do software EQC num data warehouse para extrair informação e conhecimentos significativos para o controlo de qualidade no processo industrial. Esta pesquisa visa responder a perguntas como: “Quais são as principais funcionalidades do software EQC na gestão de qualidade de produtos industriais?”, "Como é possível otimizar a gestão de dados do EQC através de um data warehouse para promover inovação, obtenção de *insights* e melhorias no controlo de qualidade?", “O que caracteriza um data warehouse e como ele difere dos dados operacionais?”, “Por que é que a modelação de dados é considerada importante no contexto de um data warehouse?”, “Como o ETL (Extract, Transform, Load) é definido no contexto de um data warehouse e qual o impacto e importância do mesmo?”. A resposta a estas e outras questões permitirá a implementação de uma solução de um sistema de Data warehouse adaptado ao contexto do software EQC, proporcionando uma base sólida para a extração de *insights* valiosos.

### 1.5.3 Critérios de Inclusão e Exclusão

Foram estabelecidos critérios específicos para a inclusão e exclusão de estudos na revisão:

**Critérios de Inclusão:**

- ✓ Estudos que abordem a implementação de data warehouses.
- ✓ Publicações em português e inglês.
- ✓ Estudos publicados depois de 2000.

**Critérios de Exclusão:**

- ✓ Artigos publicados antes de 1990.
- ✓ Publicações em línguas diferentes de português e inglês.

## 1.6 Plano de Trabalho

Esta dissertação tem duas etapas principais: as entregas P1 e P2. A entrega P1 incide sobre o estudo e planeamento do projeto, realizando um levantamento do estado da arte. Por outro lado, a entrega de P2 incide sobre a implementação do desenho final da solução e o seu desenvolvimento, ou seja, a programação da solução e respetivos testes, seguindo a metodologia de experimentação e avaliação estabelecida. No final, são apresentadas as conclusões do projeto e são refletidos aspetos a ter em conta como trabalho futuro.

Para garantir que todas as tarefas desde o início até ao final do projeto foram concluídas dentro do prazo especificado para as entregas P1 e P2, foi criado um diagrama de *Gantt*, demonstrado na figura abaixo.

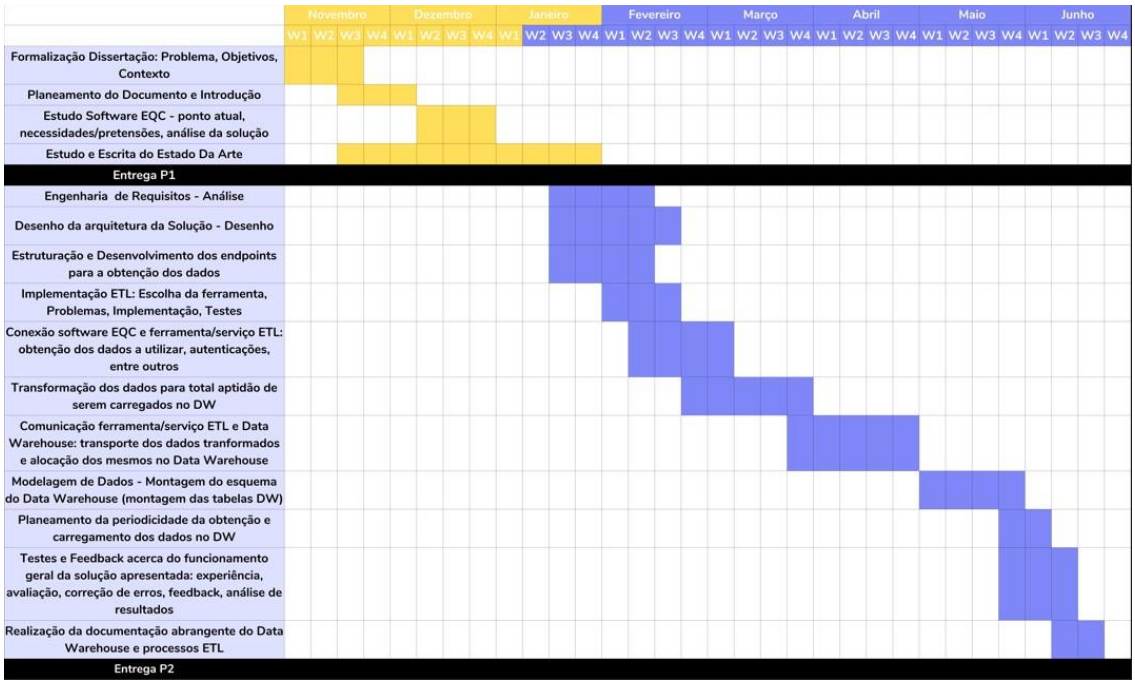



Figura 1 - Diagrama de *Gantt* do cronograma e das tarefas do projeto

Este gráfico, ilustrado na Figura 1, em conjunto com o plano de trabalho, explicado mais detalhadamente abaixo, permite uma melhor compreensão da forma como o tempo vai ser gerido para garantir que todas as prioridades sejam atendidas, enquanto ajuda o dono do projeto a acompanhar o seu progresso e a ajustar o plano, se necessário. De realçar que todos estes tópicos são contextualizados e abordados mais detalhadamente na análise da solução na secção 2.4, e consequentemente no capítulo 3, do estado da arte.

**Engenharia de Requisitos – Análise:**
A partir dos objetivos do trabalho e da descoberta do estado da arte, será especificado um conjunto de requisitos a atingir.

**Desenho da Arquitetura da Solução:**
Elaborar uma arquitetura de alto nível para a solução proposta, incluindo o desenho do processo de ETL, o modelo de dados do DW, entre outros.

**Estruturação e Desenvolvimento dos endpoints para a obtenção dos dados:**
Fase de desenvolvimento que visa a obtenção dos dados que se irão tratar/trabalhar. Sendo que os dados estão do lado do software EQC é necessário obtê-los de alguma maneira. Em parceria com a empresa EGITRON foi acordada a estruturação e desenvolvimento de endpoints responsáveis por enviar todos os dados necessários, de forma estruturada. Tudo isto será desenvolvido numa fase inicial da realização da dissertação.

**Implementação ETL: Escolha da tecnologia, problemas, implementação, testes:**
Para obter os dados ao chamar os endpoints atrás indicados, é necessário ter algum serviço ou ferramenta para o fazer, daí a análise da escolha da tecnologia e possíveis problemas a surgir com a mesma.

**Conexão software EQC e ferramenta/serviço ETL: obtenção dos dados a utilizar, autenticações, entre outros:**
Relacionado ao ponto anterior, mas focado sobretudo no processo prático de obter os dados efetuando a chamada dos endpoints com todas as autenticações e parâmetros necessários num pedido.

**Transformação dos dados para total aptidão de serem carregados no DW:**
Abordando o segundo passo do processo ETL, os dados após serem obtidos serão transformados, se necessário, para estarem aptos a serem carregados no data warehouse.

**Comunicação ferramenta/serviço ETL e Data Warehouse:**
Transporte dos dados transformados e alocação dos mesmos no data warehouse. Abordando o terceiro passo do processo ETL, os dados após serem obtidos e transformados serão carregados no Data Warehouse.

**Modelação de Dados - Montagem do esquema do Data Warehouse (montagem das tabelas DW):**
Devido a um data warehouse, geralmente utilizar um modelo dimensional, com tabelas de factos e dimensões, permitindo uma estrutura mais otimizada para consultas analíticas, é necessário desenvolver uma nova arquitetura de forma planeada e estruturada.

**Testes e Feedback acerca do funcionamento geral da solução apresentada: experiência, avaliação, correção de erros, feedback, análise de resultados:**
Chegando a este ponto, entra-se numa fase de introspeção do que foi feito, avalia-se e testa-se o funcionamento geral da solução e são tomadas decisões em relação a isso. Corrigir erros,

testes de desempenho, testes de carga, testes de atualização, validação de dados, melhorar eficiência e analisar os resultados serão alguns pontos a realizar.

**Realização da documentação abrangente do Data Warehouse e processos ETL:**
Numa fase final, irá ser realizada uma pequena documentação de tudo o que é necessário para proceder ao funcionamento desta solução.

## 1.7 Estrutura do Documento

A secção 1.1 apresenta uma interpretação analítica, crítica e ética da dissertação e do contexto em si tentando por exemplo evidenciar as implicações sociais, legais e éticas, visando contribuir para uma gestão mais informada, estratégica e responsável. A secção 1.2 aborda um pouco a indústria do passado e do presente e procura destacar a importância do software EQC na otimização dos processos industriais, enfatizando a necessidade de uma solução abrangente de Business Intelligence para extrair insights estratégicos dos dados gerados, visando aprimorar a eficiência operacional e impulsionar a inovação na indústria. O problema central que motivou esta pesquisa é então apresentado na secção 1.3, delineando as lacunas identificadas e a relevância da investigação proposta.

Os objetivos do documento estão claramente definidos na secção 1.4 e 1.5, onde, além do objetivo geral, são abordados os processos, métodos e questões de pesquisa e delineados objetivos práticos que guiarão a abordagem metodológica. A estrutura e o plano de trabalho são detalhados na secção 1.6, fornecendo uma visão do percurso que será percorrido ao longo da resolução da dissertação.

No capítulo 2, está presente uma parte essencial para a precessão em detalhe do contexto e problema do software em questão, onde o software EQC é explicado ao pormenor. Nessa mesma secção é feita uma análise da solução fazendo sempre e por várias vezes, a ligação de uma matéria específica com a abordagem e estudo da mesma no estado da arte.

No capítulo 3, mais especificamente 3.1 a 3.3, é apresentado o estado da arte, explorando conceitos chave relacionados ao data warehouse. Desde a definição fundamental do que constitui um data warehouse e a importância e técnicas de modelação de dados até às distintas arquiteturas existentes, a secção oferece uma visão abrangente do panorama atual nesta área. De seguida, da secção 3.4 a 3.5 é feita a comparação entre DW tradicional e implementações na nuvem que fornece uma perspetiva crítica e contextualizada, seguida por uma análise sobre como escolher a solução mais adequada para as necessidades específicas e por fim a análise de algumas ferramentas existentes no mercado.

A secção 3.6, dedicada aos processos de ETL e ELT, apresenta uma introdução abrangente seguida por uma análise comparativa, destacando vantagens e desvantagens associadas a ambas as abordagens, abordando também algumas ferramentas. A seleção entre ETL e ELT é discutida em profundidade.

Ao longo deste documento, procura-se não apenas abordar conceitos teóricos, mas também oferecer insights práticos que possam ser aplicados no contexto específico do problema do software EGITRON Quality Control, como o realizado na secção 2.4, onde a cada nova matéria abordada como parte integrante da possível solução era feita a indicação do seu estudo num ponto específico do estado da arte.

No capítulo 4, será detalhado o desenvolvimento do sistema proposto, desde a sua especificação até à implementação, validação e transição para o ambiente de produção. Será apresentada uma visão geral sobre os componentes do sistema, incluindo os pontos de interação e a infraestrutura utilizada. Serão descritos os passos práticos realizados para a construção do sistema, incluindo a configuração e integração das diversas tecnologias envolvidas. A avaliação do sistema será abordada no capítulo 5, detalhando os métodos utilizados para assegurar que os dados e as operações são corretas e que o desempenho é adequado. Será feita uma análise comparativa e de desempenho para garantir que o sistema está a funcionar conforme o esperado. Por fim, no capítulo 6, serão discutidas possíveis melhorias futuras na implementação, com foco em suportar múltiplas bases de dados simultâneas, melhorar a integração e monitorização, e otimizar o desempenho geral do sistema.

# 2 Software EGITRON Quality Control

Neste capítulo, será explorado o software EGITRON Quality Control (EQC), uma ferramenta fundamental para a gestão da qualidade em unidades industriais. É iniciado com uma introdução geral ao software, seguida de uma análise técnica detalhada das suas funcionalidades e conceitos. Em seguida, será discutido o estado atual do EQC, identificando as suas capacidades e áreas de melhoria. Também serão analisadas as necessidades e pretensões futuras do software, destacando as demandas a serem atendidas. Por fim, será apresentada uma visão da solução proposta, delineando como o software pode evoluir para satisfazer essas necessidades e aprimorar a sua eficiência.

## 2.1 Contexto Software

Numa visão mais técnica e abordando alguns conceitos presentes no software, o EQC permite primeiramente a criação de várias unidades industriais que pertençam à mesma empresa (quando, por exemplo, uma empresa tem várias fábricas em localizações diferentes) e utilizadores que podem aceder a uma ou mais dessas unidades industriais. Em cada uma dessas unidades industriais, o EQC é utilizado para registar os ensaios de controlo de qualidade que são efetuados ao produto final e posteriormente a emissão do relatório de qualidade que deve acompanhar o produto na sua entrega ao cliente.

Os ensaios são testes de garantia de qualidade efetuados ao produto. É um processo que as empresas usam para garantir que os seus produtos e/ou serviços alcançam regulamentos e padrões específicos através de técnicas e medições para evitar a ocorrência de problemas e garantir a satisfação do cliente final. O EQC é altamente parametrizável no que concerne ao seu uso, pelo que, é possível criar e configurar qualquer ensaio normativo. O conceito de "Ensaio Personalizado" no EQC permite aos utilizadores aceder a uma ampla gama de medições, análises estatísticas e cálculos automáticos de uma forma organizada e fácil. Esses dados podem ser adquiridos automaticamente através do EGITRON HUB, simplificando ainda mais o processo. Dada a versatilidade do EQC, o cliente pode criar inúmeras características por ensaio e, inclusivamente, características que dependam de fórmulas. É dada a

possibilidade de criação de fórmulas com base em características medidas. Para maior agilidade, é possível criar “Grupos de Ensaios”. Esta funcionalidade permite atribuir a uma determinada gama de produtos, ensaios pré-definidos.

Na Figura 2 é possível ver exemplos de ensaios pré-definidos. Estes são alguns exemplos de ensaios pré-definidos em maior detalhe:

- **Dimensional:** Teste para verificar as dimensões do produto, incluindo comprimento, largura, altura e espessura, assegurando a sua conformidade com as especificações.
- **Massa:** Teste físico para avaliação do peso e da massa volúmica do produto, garantindo que este atenda aos padrões estabelecidos.
- **Humidade:** Teste químico com o objetivo de verificar a quantidade de água impregnada no produto, assegurando níveis aceitáveis de humidade conforme os requisitos.
- **TCA:** Teste químico realizado na cortiça para verificar a presença do composto *Tricloroanisol*.

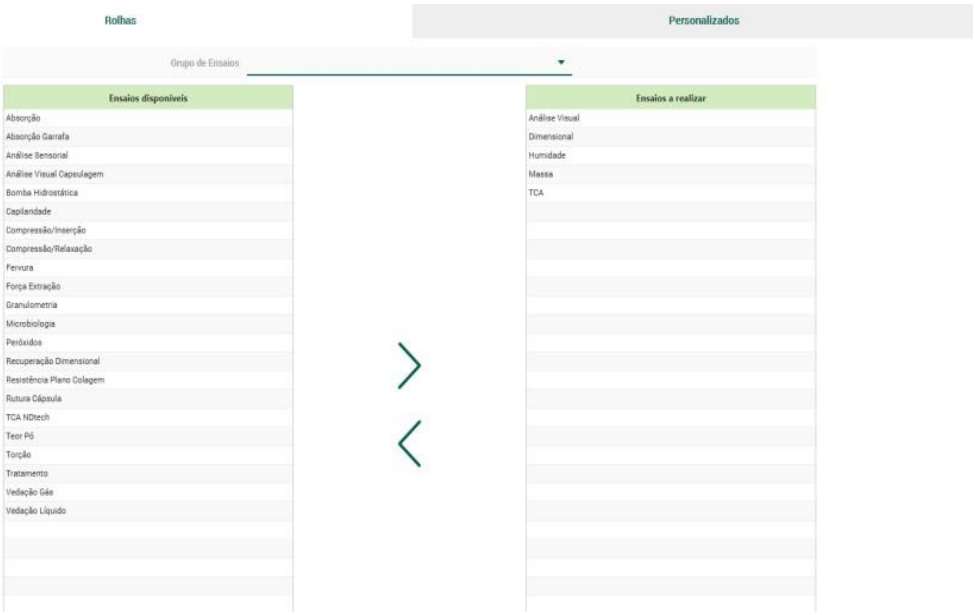



Figura 2 - EQC - Exemplos de ensaios pré-definidos

Cada cliente tem as suas exigências, assim como cada produto. Nesse sentido, o software EQC apresenta o conceito de "Modelo de Controlo" (Figura 3). O modelo de controlo serve como *template* para a criação de relatórios de qualidade. Cada cliente tem a liberdade de requerer de um produto produzido por uma unidade industrial determinados ensaios, distintos de outro cliente, e as suas próprias especificações para esses ensaios.

O modelo de controlo facilita a seleção e pré-configuração desses ensaios, simplificando o processo para os utilizadores na unidade industrial, onde basta selecionar o produto e o cliente para gerar automaticamente o relatório com os ensaios necessários.

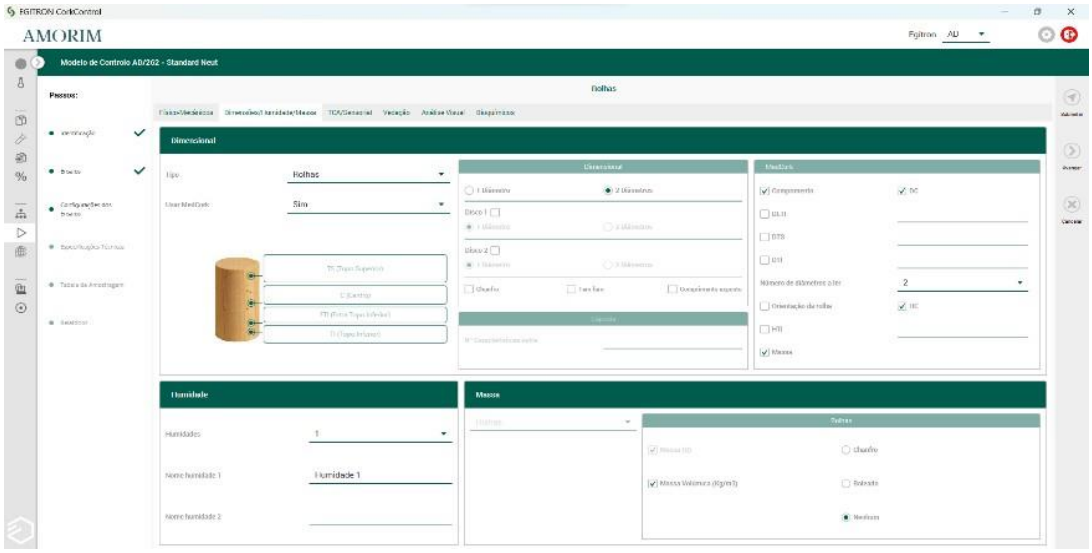



Figura 3 - EQC - Exemplo de configuração de um modelo de controlo

A criação de um relatório, como visualizado na Figura 4, é sempre efetuada a partir de um modelo de controlo, o que nos indicará a que ensaios o produto ou produtos estarão sujeitos. Todos estes testes realizados ao produto são realizados em máquinas especificas ou indicadas à situação. O EQC, neste momento, fornece a integração com outros softwares da EGITRON que permitem a aquisição desses valores em tempo real e a apresentação dos mesmos no *software* (várias estatísticas ou comparação de limites de valores são também possíveis de visualizar). Depois da produção estar concluída e todos os testes essenciais serem realizados, o EQC permite a criação de relatórios de qualidade. Esses relatórios, contendo os resultados dos ensaios de garantia de qualidade, são incorporados à documentação da venda do produto. Isso oferece aos clientes uma visão clara e transparente sobre os altos padrões de qualidade atingidos pelo produto.

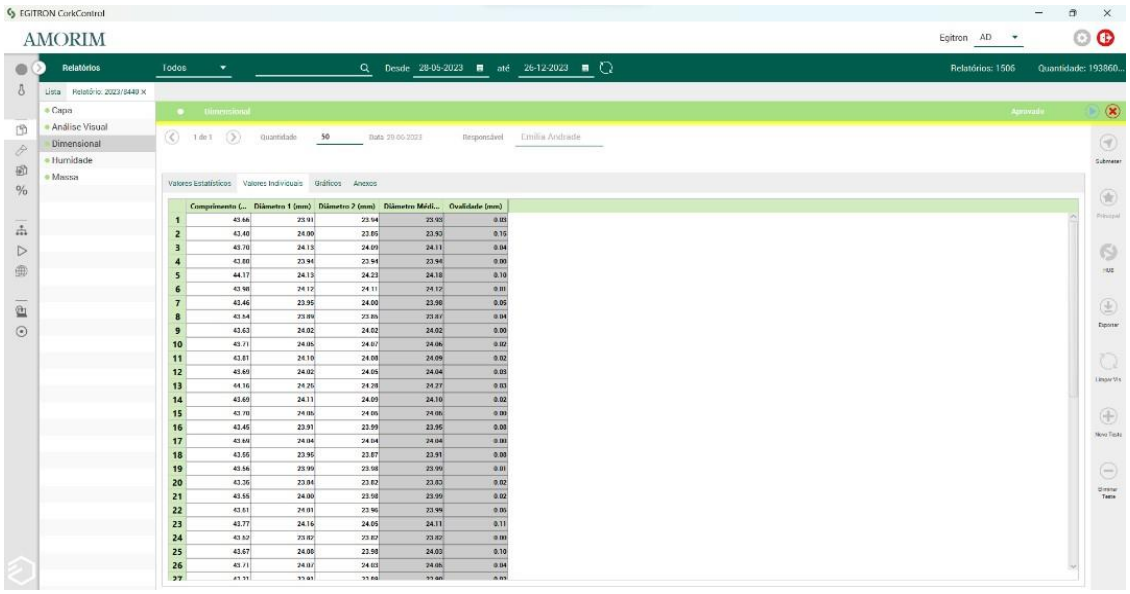



Figura 4 - EQC - Exemplo de um relatório

O EQC facilita a partilha eficiente de relatórios ao longo da cadeia de produção. Por exemplo, na Amorim Florestal, onde placas de cortiça são obtidas das árvores, são gerados relatórios para essas mesmas placas (produto final da Amorim Florestal) que por sua vez são enviadas para a Amorim Cork, onde a partir delas são criadas rolhas de cortiça. Os relatórios (exemplo de Relatório de Controlo de Qualidade na Figura 5), para além de serem enviados com a encomenda (em papel) são compartilhados internamente com a Amorim Cork. Isto permite a reutilização dos valores de ensaios já efetuados, evitando a repetição de testes desnecessários. Da mesma forma, quando as rolhas são enviadas posteriormente para a Amorim Top Series (ATS) para serem capsuladas (rolhas para garrafas de vinho do porto ou garrafas de *whiskey* por exemplo), o processo de partilha de relatórios é repetido, garantindo eficiência e consistência ao longo da cadeia produtiva.

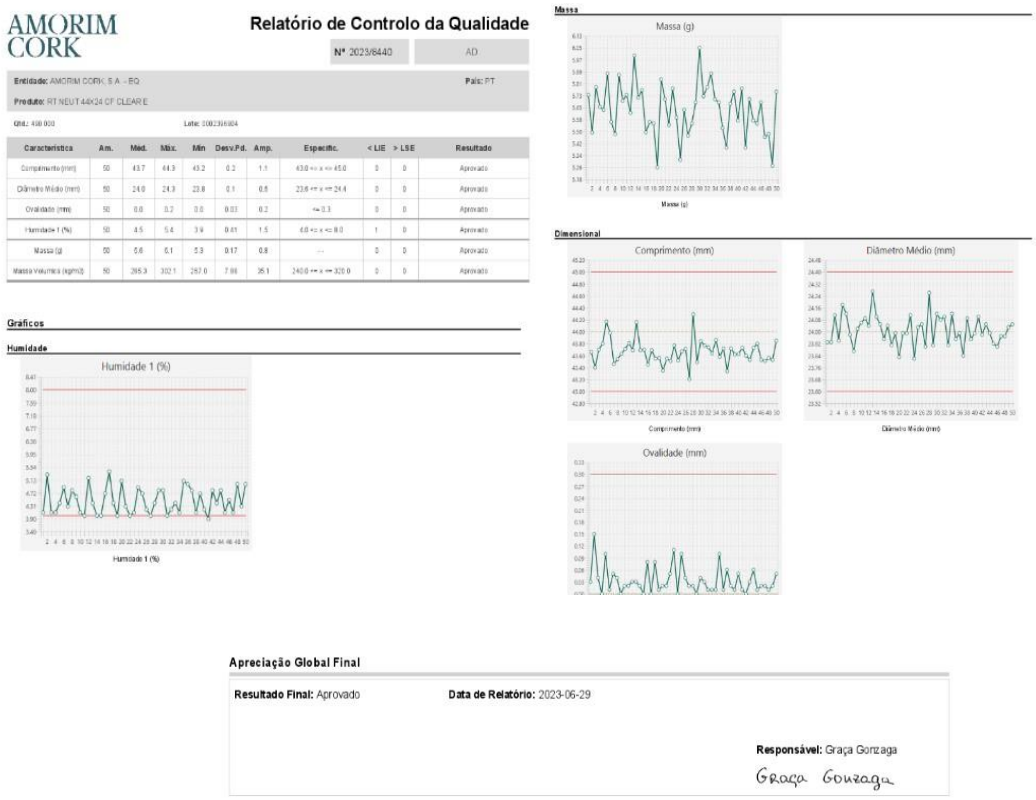



Figura 5 - EQC - Exemplo de um relatório de controlo de qualidade em PDF

## 2.2 EQC – Ponto Atual

O software EQC desempenha um papel central na gestão e controlo de qualidade de produtos em diversas unidades industriais, tanto a nível nacional quanto internacional. O processo iniciasse quando uma unidade industrial recebe um produto, como uma placa de cortiça, e o introduz no software EQC.

Este software oferece uma variedade de ensaios aos quais o produto pode ser submetido. Cada ensaio representa um conjunto específico de testes, como o ensaio de Humidade, o ensaio de Resistência ou o ensaio de Absorção. Ao criar um "Modelo de Controlo", como exemplificado na Figura 3, é possível associar num só modelo vários ensaios, podendo esse mesmo modelo futuramente ser usado para outros mais produtos. De seguida é possível criar um "Relatório" (Figura 4), onde simplesmente é associado o modelo de controlo ao produto desejado (modelo esse que contem os ensaios específicos).

O EQC possui a capacidade de se conectar a diversas máquinas ou dispositivos *hardware*, facilitando a obtenção automática dos valores resultantes dos testes realizados no produto. Com todos os dados reunidos/obtidos, o relatório é então avaliado, resultando na aprovação ou rejeição do produto. No caso de aprovação, o produto está pronto para ser expedido. Acompanhando cada envio, há um "Relatório de Controlo de Qualidade" (conforme ilustrado na Figura 5), que inclui todas as métricas associadas aos testes realizados no produto. Esse documento proporciona uma visão abrangente da qualidade do produto, oferecendo uma garantia clara e transparente aos clientes e parceiros comerciais.

Atualmente o *software* EQC apresenta um *backend* desenvolvido na linguagem JAVA em parceria com a *framework* Spring Boot e respeitando a arquitetura REST, atuando numa base de dados de modelo relacional robusta (MySQL). Essa arquitetura, adaptada para ambientes industriais, viabiliza a manipulação ágil e constante de dados em tabelas, assegurando eficiência nas operações de inserção, remoção e atualização. É um sistema em evolução contínua, sempre atendendo às demandas dinâmicas do cenário industrial.

## 2.3 EQC – Necessidades / Pretensões

Como mencionado anteriormente na secção 2.2, é notável que o *software* se tornou um elemento crucial no quotidiano das empresas, especialmente no que diz respeito ao controlo de qualidade dos seus produtos. Em diversas localidades ao redor do mundo, a quantidade de dados e valores presentes nas bases de dados de cada unidade industrial aumenta a cada dia, devido à constante inserção e atualização desses dados durante o processo produtivo.

Empresas proeminentes, como a Corticeira Amorim, líder mundial no setor da cortiça e uma das principais utilizadoras do software EQC, enfrentam o desafio da impossibilidade de realizar uma análise abrangente e estratégica para otimizar os processos de produção, aprimorar a tomada de decisões e impulsionar a inovação nas empresas industriais. Existem diversos exemplos de informações, dados e insights que a empresa deseja aceder. Relacionando um pouco com o contexto do software, explicado na secção 2.1, é possível indicar alguns desses exemplos:

- ✓ **Cruzamento global de dados:** Integração de dados entre diferentes unidades industriais à escala mundial.

- ✓ **Comparação entre unidades industriais:** Facilidade em comparar o mesmo produto em diversas unidades industriais, identificando discrepâncias e padrões de desempenho.

- ✓ **Análise de erros em relatórios:** Capacidade de analisar e comparar relatórios, identificando unidades industriais com mais ou menos erros.

- ✓ **Evolução do produto ao longo dos anos:** Acompanhamento da evolução do produto ao longo do tempo para *insights* sobre melhorias contínuas.

- ✓ **Variação do químico TCA em relação às condições climáticas e localização geográfica:** Análise e comparação da variação do químico TCA ao longo do tempo, considerando as condições climáticas ou localização geográfica.

- ✓ **Comparação ou cruzamento de N características:** Ovalidade, massa volúmica, diâmetro médio, peso, humidade, torção, entre muitos outros, cruzando-os uns com os outros (ex: massa volúmica x humidade) parametrizando com dados como intervalo de tempo, localização/região, unidades industriais, máximos e mínimos dos valores, limites, etc)

- ✓ **Identificação de possíveis falhas:** Cruzamento de dados para identificar falhas em diferentes etapas do processo produtivo.

- ✓ **Análise holística:** Integração de todas as informações para compreensão holística do desempenho global, facilitando a identificação de áreas de melhoria.

Com isto, prende-se a necessidade de planear e desenvolver uma solução que possa responder a estas necessidades/pretensões. Em termos simples, é necessário ter um enorme armazém de dados para armazenar de forma consolidada todo o tipo de informação que é possível extrair do processo produtivo diário das unidades industriais. Essa abordagem permitiria uma gestão mais informada e estratégica, possibilitando a tomada de decisões baseada em dados sólidos e promovendo a excelência operacional em todas as unidades industriais.

## 2.4 Análise de Visão da Solução

Todos os dados existentes atualmente, pertencem ao software EQC, desde informação dos utilizadores, produtos e ensaios ou até mesmo dos valores obtidos nas realizações dos testes desses mesmos ensaios como explicado de forma sucinta na secção 2.2. De alguma forma, esses mesmos dados precisam de ser obtidos, se necessário transformados e por fim colocados num armazém de dados.

A este fenómeno dá-se o nome de ETL (Extração, Transformação, Carregamento) e o mesmo é estudado/analisado no capítulo do estado da arte. Encontra-se em planeamento e estruturação com a colaboração da EGITRON o desenvolvimento de alguns recursos REST (endpoints), separados por responsabilidades especificas do software, para através desses mesmos endpoints obter os dados estruturados no formato JSON e após, se necessário algumas transformações, inserir esses mesmos dados no armazém de dados. Por "responsabilidades específicas do *software*", entende-se a divisão de dados de forma organizada por cada *endpoint*. Por exemplo, um dos *endpoints* estaria responsável por retornar todos os utilizadores e gestores do *software*, outro *endpoint* retornaria todos os produtos existentes, outro *endpoint* os relatórios de qualidade realizados nos produtos com os valores obtidos nos testes. Todos esses *endpoints* estão interligados (exemplo: *endpoint* que retorna relatórios existentes relaciona-se com o *endpoint* dos produtos pois um relatório não existe sem um produto associado) e são parametrizáveis no que toca à sua chamada.

Com isto, é possível obter uma fonte de dados minimamente filtrada e estruturada, mas ainda não transformada na totalidade de forma a conseguir inserir diretamente no armazém de dados. Na Figura 6, é possível perceber o procedimento. Vai existir uma camada (identificada como "ETL" na Figura 5) responsável por chamar estes recursos REST disponibilizados pelo software EQC, transformá-los e inseri-los no armazém de dados. No que toca à periodicidade da obtenção e carregamento desses mesmos dados no armazém de dados, o mesmo tem de ser pensado e planeado apesar de no caso em questão não ser crítico o desatualizar momentâneo do armazém de dados. A atualização diária a uma hora mais indicada, a atualização semanal ou até mesmo mensal são possibilidades em cima da mesa.

Como é possível ver em maior detalhe na secção 3.2.2 e 3.3 do estado da arte, um data warehouse, no que toca à sua arquitetura difere daquela encontrada em bases de dados relacionais convencionais, como a existente no software EQC, principalmente pela sua orientação para análises complexas e consultas *ad hoc*. O data warehouse, geralmente utiliza um modelo dimensional, com tabelas de factos e dimensões, permitindo uma estrutura mais otimizada para consultas analíticas, enquanto as bases de dados relacionais seguem uma abordagem mais normalizada, adequada para operações transacionais e rotineiras. Devido a isso, é crucial planear e desenvolver uma nova arquitetura baseada no modelo dimensional, alinhando-a com os dados e conteúdos presentes na atual base de dados do software EQC.

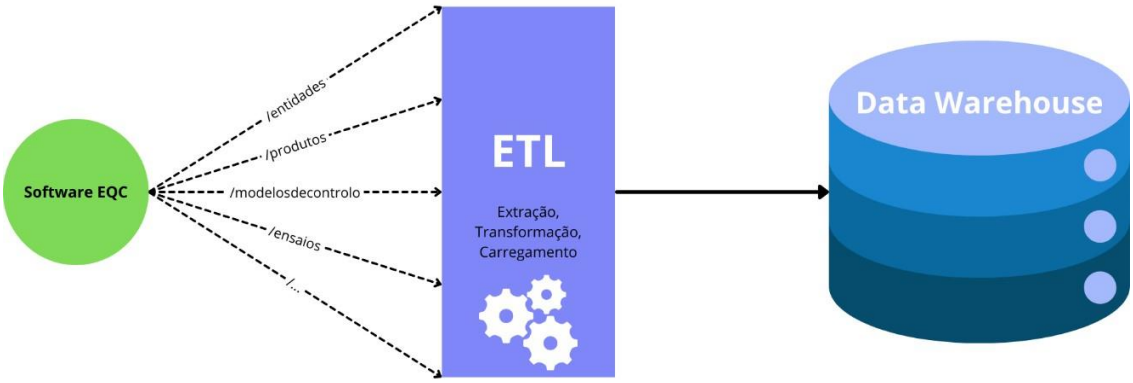



Figura 6 - EQC - Planeamento da visão da solução do problema

Com a nova arquitetura, baseada no modelo dimensional, planeada e desenvolvida (montagem e ligação de todas as tabelas necessárias), é necessário com base em alguns fatores selecionar a melhor solução de data warehouse como abordado na secção 3.6 do estado da arte.

A melhor solução é estudada e analisada da forma mais "correta" e independente, não sendo condicionada pela opinião da empresa em questão. Em função do contexto do trabalho e da implementação da solução poderá fazer sentido limitar o que foi apresentado no estado da arte acerca da escolha da ferramenta, podendo ter de se optar por considerar ferramentas *open source* ou de licença gratuita.

Em suma, o processo planeado é extenso e encontra-se planeado na secção 1.6 (plano de trabalho

## 2.5 Sumário

Neste capítulo, foi abordado detalhadamente o software EGITRON Quality Control (EQC). Primeiramente foi fornecida uma visão geral do software e a sua importância na gestão da qualidade. Em seguida, na secção 2.1, foram discutidas as funcionalidades técnicas do EQC, como a criação de unidades industriais, o registo de ensaios de controle de qualidade e a emissão de relatórios de qualidade. Também foi explicada a parametrização do software e o conceito de "Ensaio Personalizado", além da integração com o EGITRON HUB para aquisição automática de dados.

Na secção 2.2, foi avaliado o estado atual do software, destacando a sua arquitetura *backend* desenvolvida em Java com Spring Boot e sua base de dados relacional MySQL. Foi discutido como o software EQC conecta-se a diversos dispositivos para obter automaticamente os valores dos testes e como os relatórios de qualidade são gerados e partilhados ao longo da cadeia produtiva.

Na secção 2.3, foram identificadas as demandas futuras do software, como a necessidade de um armazém de dados para uma análise abrangente e estratégica. Foram apontados exemplos específicos de análises desejadas pelas empresas, como a comparação de dados entre unidades industriais e a identificação de possíveis falhas no processo produtivo.

Na seção 2.4, foi discutido o planeamento da solução para atender às necessidades identificadas. Foi explicado o processo de ETL (Extração, Transformação e Carregamento) para obter, transformar e inserir dados no armazém de dados. Foi abordada a necessidade de uma nova arquitetura baseada em um modelo dimensional e a seleção da melhor solução de data warehouse.

# 3 Estado da Arte

A escolha e estruturação das secções presentes no estado da arte são fundamentais para fornecer um panorama completo e relevante sobre os tópicos abordados. A razão pela qual cada secção foi incorporada neste estudo é guiada pela necessidade de estabelecer uma base sólida de compreensão dos conceitos-chave, especificamente no contexto do software EGITRON Quality Control.

Neste capítulo, será abordado o estado da arte em relação aos conceitos e tecnologias fundamentais para o desenvolvimento de um sistema de data warehouse (DW). O capítulo é iniciado com uma definição clara do que é um data warehouse e a sua distinção em relação aos dados operacionais. Em seguida, é discutida a importância da modelação de dados, apresentando as principais técnicas utilizadas neste processo.

Também serão exploradas diferentes arquiteturas de data warehouse, incluindo o contexto em que cada uma se aplica e as suas variações. Irá ser feita uma análise comparativa entre os DW tradicionais e os DW na *cloud*, destacando as vantagens e desvantagens de cada abordagem.

Adicionalmente, será realizada uma avaliação das principais ferramentas de data warehouse disponíveis no mercado. Por fim, são discutidos os processos de ETL (Extração, Transformação e Carregamento) e ELT (Extração, Carregamento e Transformação), comparando-os e oferecendo orientações sobre como escolher a abordagem mais adequada para diferentes cenários.

## 3.1 Data Warehouse

O conceito central do data warehouse reside na capacidade de responder a perguntas que ultrapassam as transações individuais, abrangendo todo o processo de negócios. Distinto dos sistemas OLTP, o desenho do data warehouse reflete a visão dos especialistas sobre o negócio, sendo esse o diferencial crucial entre ambos. De acordo com (Ferreira, 2005), "*Warehousing*" é uma técnica utilizada para recuperar e integrar dados de fontes distribuídas, autônomas e, possivelmente, heterogêneas. Esses dados são armazenados num amplo depósito denominado data warehouse, onde são resumidos e organizados em dimensões, facilitando consultas e análises por meio de sistemas de suporte à decisão.

Os data warehouses são amplamente adotados pelas empresas devido à sua capacidade de fornecer uma base sólida para a integração de dados corporativos e históricos, fundamentais para análises gerenciais. A sua construção segue uma abordagem passo a passo, organizando e armazenando dados com uma perspetiva a longo prazo. A análise de tendências a partir de dados históricos básicos é uma vantagem inerente. No entanto, a fase mais complexa é o processo de carga, onde dados de diferentes ambientes operacionais são selecionados, padronizados, limpos e transferidos para o data warehouse.

Após a criação inicial, o data warehouse recebe cargas incrementais refletindo o ambiente operacional ao longo do tempo, transformando-se num vasto repositório para sistemas de apoio à decisão. Distingue-se pela não-volatilidade, uma vez que os dados, uma vez carregados, não sofrem mais alterações. A modelação dimensional é a técnica fundamental para obter um modelo de data warehouse que identifique e represente informações cruciais para o negócio. Quando bem aplicada, a modelação dimensional reflete a mentalidade dos gestores e otimiza o processo de tomada de decisões. Data warehousing engloba um conjunto de tecnologias de suporte à decisão, com o objetivo de capacitar o profissional do conhecimento (executivo, gestor, analista) a tomar decisões melhores e mais rápidas (Chaudhuri, 1997). Um data warehouse corporativo apresenta características distintivas:

- ✓ Separado dos sistemas transacionais e alimentado por estes.
- ✓ Totalmente disponível para consultas dos utilizadores de negócios.
- ✓ Integrado como uma base única e padrão para o modelo da empresa.
- ✓ Associado a informações temporais e períodos definidos.
- ✓ Orientado por assunto, descrevendo o desempenho do negócio.
- ✓ Acessível a utilizadores com conhecimento limitado de sistemas computacionais.

A Figura 7 demonstra o fluxo de dados de vários sistemas OLTP, que seguem o modelo relacional para o data warehouse, que disponibiliza uma base de dados para consultas utilizadas por ferramentas específicas.

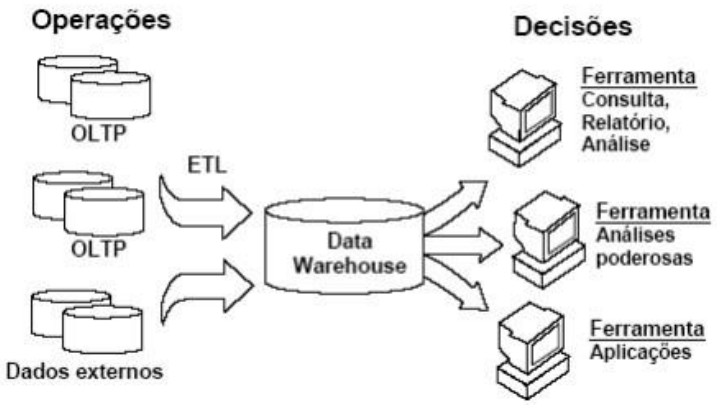



Figura 7 - EQC - Fluxo de dados para o Data Warehouse (Schio, 2006)

### 3.1.1 Data Warehouse vs Dados Operacionais

Um data warehouse é uma coleção de dados "orientada a assuntos, integrada, variável no tempo e não volátil, utilizada principalmente na tomada de decisões organizacionais." (Inmon, 2005). Geralmente, o data warehouse é mantido separadamente das bases de dados operacionais da organização, e há várias razões para essa separação. O data warehouse oferece suporte ao processamento analítico online (OLAP), cujos requisitos funcionais e de desempenho diferem significativamente daqueles das aplicações de processamento transacional online (OLTP) tradicionalmente mantidas pelas bases de dados operacionais.

As aplicações OLTP automatizam tarefas de processamento de dados burocráticas, como entrada de pedidos e transações bancárias, que constituem as operações quotidianas fundamentais de uma organização. Essas tarefas são estruturadas, repetitivas e consistem em transações curtas, atômicas e isoladas. As transações exigem dados detalhados e atualizados e leem ou atualizam poucos (dezenas de) registos acedidos geralmente pelas suas chaves primárias. As bases de dados operacionais tendem a ter centenas de *megabytes* a *gigabytes* de tamanho. A consistência e recuperabilidade da base de dados são críticas, e a maximização do *throughput* da transação é a métrica chave de desempenho.

Em contraste, os data warehouses são direcionados para o suporte à decisão. “Os dados históricos, resumidos e consolidados são mais importantes do que os registos individuais e detalhados” (Chaudhuri, 1997). As cargas de trabalho são intensivas em *querys*, principalmente querys *ad hoc* complexas que podem aceder milhões de registos e realizar muitas junções e agregações. A capacidade de processar consultas rapidamente (*query throughput*) e os tempos de resposta das consultas são mais cruciais do que a capacidade de processar transações rapidamente (*transaction throughput*). (Chaudhuri, 1997).

Dado que as base de dados operacionais são ajustadas para suportar cargas de trabalho OLTP conhecidas, tentar executar consultas OLAP complexas nas bases de dados operacionais resultaria num desempenho inaceitável. Além disso, o suporte à decisão requer dados que podem estar ausentes nas bases de dados operacionais; por exemplo, compreender tendências ou fazer previsões requer dados históricos, enquanto as bases de dados operacionais armazenam apenas dados atuais. O suporte à decisão geralmente exige consolidar dados de muitas fontes heterogêneas. As diferentes fontes podem conter dados de qualidade variada ou usar representações, códigos e formatos inconsistentes, que precisam de ser conciliados.

Além disso, a arquitetura do data warehouse é otimizada para consultas analíticas e relatórios, utilizando esquemas dimensionais como estrela ou floco de neve. Esses esquemas facilitam a modelação e a recuperação eficiente de dados para análises, permitindo uma visão multidimensional dos dados. Enquanto as bases de dados operacionais geralmente seguem um modelo relacional normalizado para garantir a consistência e integridade dos dados transacionais, os data warehouses adotam estratégias de modelação mais flexíveis para atender às necessidades complexas de análise.

Para se compreender o conceito de data warehouse, muitas vezes é útil distingui-lo dos sistemas operacionais convencionais, dos quais ele obtém todas as suas informações. A Tabela 1 lista algumas diferenças entre OLTP e armazenamento de dados de data warehouse.

Tabela 1 - Base de Dados Operacional vs Data Warehouse

| **Características** | **Base de Dados Operacional** | **Data Warehouse** |
|---|---|---|
| **Esquema** | Muitas tabelas, esquema normalizado. | Menos tabelas, esquema desnormalizado. |
| **Tipo de operações dos utilizadores** | Adição, atualização e exclusão de registos. | Consulta de registos para leitura. |
| **Acesso de utilizadores** | Muitos utilizadores acedendo registos. | Poucos utilizadores acedendo registos. |
| **Uso de logs** | Logs de transação/reversão utilizados. | Sem logs de transação/reversão necessários. |
| **Granularidade dos dados** | Muitas linhas de detalhe. | Linhas consolidadas, resumidas. |

| | | |
|---|---|---|
| **Índices** | Índices menores para atualizações rápidas. | Índices vastos para consultas otimizadas. |
| **Exatidão dos dados** | Dados sempre exatos atualmente; podem ser atualizados. | Exato para um momento específico no tempo. |

## 3.2 Modelação de Dados

A modelação de dados é uma prática essencial no desenvolvimento de sistemas de informação, oferecendo uma visão abstrata e precisa dos dados necessários para o negócio. Esta secção explora a importância da modelação de dados, destacando as principais técnicas utilizadas: a modelação Entidade-Relacionamento (E/R) e a modelação dimensional, cada uma com suas vantagens e contextos de aplicação específicos. Esta secção também detalha as características e usos dos modelos dimensionais, como os esquemas em estrela, floco de neve e multi-estrela, que são desenhados para otimizar a análise e o reporte de dados.

### 3.2.1 A Importância da Modelação de Dados

"Um modelo é uma abstração e reflexão do mundo real." (Ballard, 2012). A modelação proporciona a capacidade de visualizar o que ainda não se conseguiu concretizar. Isso aplica-se também à modelação de dados. O objetivo principal de um modelo de dados é garantir que todos os objetos de dados necessários para o negócio sejam representados com precisão e integralidade.

Do ponto de vista do negócio, um modelo de dados pode ser facilmente verificado, pois é construído usando notações e linguagem fáceis de entender e decifrar. No entanto, do ponto de vista técnico, o modelo de dados é detalhado o suficiente para servir como um plano para o administrador de base de dados ao construir a base de dados física. Por exemplo, o modelo pode ser facilmente utilizado para definir os elementos-chave, como chaves primárias, chaves estrangeiras e tabelas que serão usadas no design da estrutura de dados. Tradicionalmente, os modeladores de dados têm utilizado o diagrama Entidade-Relacionamento (E/R) como parte do processo de modelação de dados, como meio de comunicação com os analistas de negócios. O foco do modelo E/R é capturar as relações entre várias entidades da organização ou processo para os quais se está a projetar o modelo. O modelo de entidade relacionamento tem por base a descrição de algo do mundo real realizada através de um conjunto de objetos chamados "entidades" e pelo conjunto de "relacionamentos" entre esses objetos. (Dantas,

2016). Por outro lado, o modelo dimensional tem o seu foco no negócio. A modelação dimensional proporciona uma capacidade aprimorada de visualizar as questões abstratas que os analistas de negócios precisam de responder. Usando a modelação dimensional, os analistas podem entender e navegar facilmente na estrutura de dados e explorar completamente os dados.

Embora relacionados, os modelos E/R e dimensionais são extremamente diferentes. Enquanto os modelos E/R geralmente são normalizados, os modelos dimensionais são desnormalizados. Há muito debate sobre qual método é melhor e em quais condições selecionar uma técnica específica. O modelo E/R é frequentemente utilizado ao projetar aplicações altamente orientadas para transações (OLTP), enquanto o modelo dimensional é mais adequado para consultas e análises *ad hoc* em aplicações de armazenamento de dados.

### 3.2.2 Técnicas de Modelação de Dados

Nesta secção, serão abordadas diversas técnicas de modelação de dados, destacando-se as abordagens de Modelação E/R e Modelo Dimensional, com o objetivo de ilustrar as suas características, vantagens e desvantagens em diferentes contextos de utilização.

#### 3.2.2.1 Modelação E/R

A modelação E/R é uma técnica de design na qual se armazena os dados de forma altamente normalizada dentro de uma base de dados relacional. A Figura 8 mostra uma visualização de um modelo E/R normalizado, que simplesmente representa como as várias tabelas de um modelo E/R conectam-se e inter-relacionam-se. Isso é chamado de estrutura normalizada.

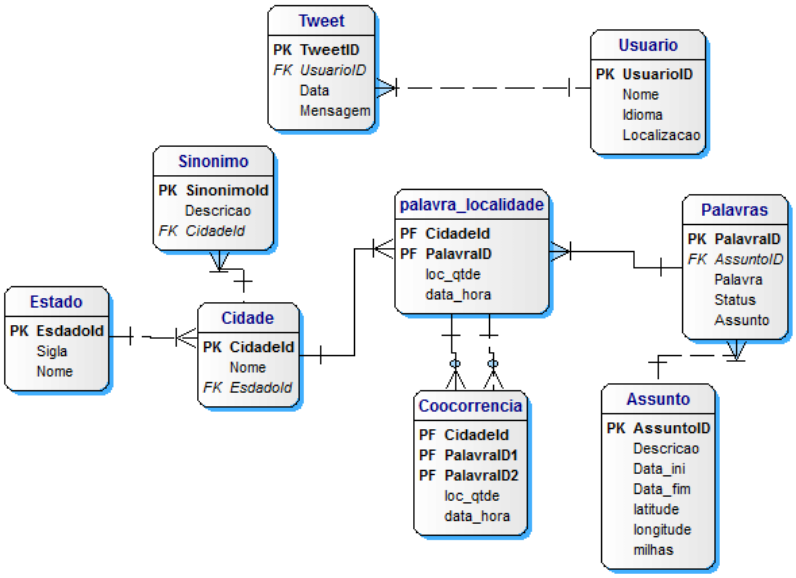



Figura 8 - Um modelo E/R típico

O objetivo da normalização é minimizar a redundância, não armazenando os mesmos dados em várias tabelas. Como resultado, a normalização pode minimizar problemas de integridade,

pois as atualizações SQL precisam de ser aplicadas apenas a uma tabela. No entanto, consultas, especialmente aquelas envolvendo tabelas muito grandes que incluem a junção de dados armazenados em várias tabelas normalizadas, podem exigir esforço adicional e programação para obter desempenho aceitável.

Os diagramas Entidade-Relação e as técnicas de normalização são muito utilizados para a conceção de bases de dados em ambientes OLTP. No entanto, as conceções de bases de dados recomendadas pelos diagramas E/R são inadequadas para sistemas de apoio à decisão em que a eficiência na consulta e no carregamento de dados (incluindo carregamentos incrementais) é importante. (Chaudhuri, 1997). Embora os dados em tabelas normalizadas sejam uma forma muito pura de dados e minimizem a redundância, pode ser desafiador para os analistas navegar. Por exemplo, se um analista precisar de navegar por um modelo de dados que requer a junção de 15 tabelas, pode ser difícil e pouco intuitivo. Isso é mitigado com um modelo dimensional, que possui dimensões padronizadas e independentes. Aplicações de armazenamento de dados são principalmente de leitura e, portanto, geralmente podem beneficiar da desnormalização. A desnormalização é uma técnica que envolve a duplicação de dados em uma ou mais tabelas para minimizar ou eliminar junções demoradas. Nessas situações, controlos adequados devem ser implementados para garantir que os dados duplicados estejam sempre consistentes em todas as tabelas, evitando problemas de integridade de dados. Segundo (Ballard, 2012), o modelo E/R concentra-se basicamente em três elementos: entidades, atributos e relacionamentos. Uma entidade é uma categoria de um objeto de interesse para o negócio, com uma definição correspondente que estabelece os seus limites. Cada entidade tem atributos associados, que são características de interesse para o negócio. As relações entre entidades descrevem como elas interagem, geralmente expressas como um verbo.

As vantagens da técnica de modelação E/R são:

- ✓ Eliminação de dados redundantes, economizando espaço de armazenamento e facilitando a aplicação de restrições de integridade.

- ✓ Comandos *INSERT*, *UPDATE* e *DELETE* num modelo E/R normalizado são mais rápidos do que num modelo desnormalizado devido a ter menos fontes redundantes de dados.

- ✓ A técnica de modelação E/R ajuda a capturar as inter-relações entre várias entidades para as quais se está a projetar a base de dados, sendo eficaz na representação de relacionamentos.

Uma desvantagem do modelo E/R é que ele não é tão eficiente ao realizar consultas muito extensas envolvendo várias tabelas, sendo mais adequado para processamento *INSERT*, *UPDATE* ou *DELETE* do que para *SELECT*.

#### 3.2.2.1.1 Modelo Dimensional

Para superar problemas de desempenho em consultas extensas no data warehouse, recorre-se a modelos dimensionais. “A modelação multidimensional é a técnica estruturada desenvolvida para a obtenção de modelos de dados de simples entendimento e alta performance de acesso aos dados.” (Machado, 2004). A abordagem de modelação dimensional proporciona uma melhoria no desempenho das consultas para relatórios sumarizados sem comprometer a integridade dos dados. No entanto, essa melhoria de desempenho vem com o custo de requerer mais espaço de armazenamento.

Um modelo dimensional é comumente conhecido como esquema em estrela. Este tipo de modelo é amplamente utilizado em data warehousing, pois pode oferecer um desempenho de consulta muito melhor, especialmente em consultas muito extensas, em comparação com um modelo E/R (Entidade/Relacionamento). Além disso, tem a grande vantagem de ser mais fácil de compreender. Geralmente, consiste numa grande tabela de factos, envolvida por várias outras tabelas contendo dados descritivos, denominadas dimensões. “Cada tabela de dimensão consiste em colunas que correspondem a atributos da dimensão”. (Chaudhuri, 1997). Quando é representado graficamente, assemelha-se à forma de uma estrela, daí o nome. A Figura 9 ilustra um exemplo de esquema em estrela.

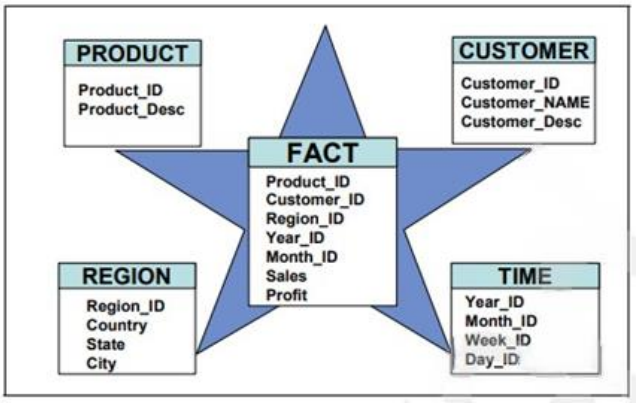



Figura 9 - Um esquema em estrela / modelo dimensional (Ballard, 2012)

Essa abordagem de modelação é especialmente eficaz para otimizar o desempenho das consultas em relatórios que necessitam de agregações de dados, tornando-se uma escolha valiosa em ambientes de data warehousing, mesmo considerando o aumento no requisito de espaço de armazenamento.

#### 3.2.2.1.2 Tipos de Tabelas

O modelo dimensional é composto por dois tipos de tabelas com características distintas. São elas:

- ✓ Tabela de factos
- ✓ Tabela de dimensões

(Ballard, 2012) fornece-nos um conjunto de características sobre os dois tipos de tabelas:

**Tabela de factos:**

- ✓ "Os factos são fundamentais para os DW. Mostram factos reais do mundo real e podem ser vistos como processos que geram dados ao longo do tempo. São caracterizados por propriedades." (Tryfona, 1999).
- ✓ A tabela de factos contém valores numéricos daquilo que é medido.
- ✓ Cada tabela de factos contém as chaves para as tabelas de dimensões associadas. Estas são chamadas chaves estrangeiras na tabela de factos.
- ✓ As tabelas de factos contêm normalmente um número pequeno de colunas.
- ✓ Em comparação com as tabelas de dimensão, as tabelas de factos têm um grande número de linhas.
- ✓ A informação numa tabela de factos tem características, tais como:
    - São numéricas e usadas para gerar agregados e resumos.
    - Os valores dos dados precisam de ser aditivos, ou semi-aditivos, para permitir o resumo de um grande número de valores.
    - Todos os factos no Segmento 2 devem referir-se diretamente às chaves de dimensão no Segmento 1 da estrutura, como se pode ver na Figura 10. Isto permite o acesso a informações adicionais das tabelas de dimensão.

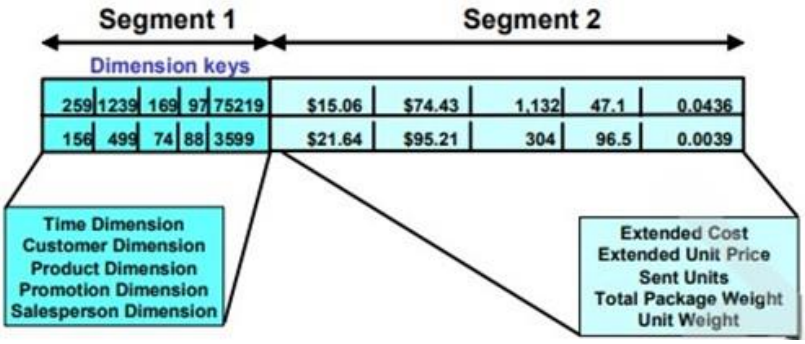



Figura 10 - Estrutura tabela de factos

**Tabela dimensional:**

- ✓ As tabelas dimensionais contêm os detalhes sobre os factos. Isso, por exemplo, permite que os analistas de negócios entendam melhor os dados e os seus relatórios.

- ✓ As tabelas dimensionais contêm informações descritivas sobre os valores numéricos na tabela de factos. Ou seja, contêm os atributos dos factos. Por exemplo, as tabelas dimensionais para uma aplicação de análise de *marketing* podem incluir atributos como período de tempo, região de *marketing* e tipo de produto.

- ✓ Como os dados numa tabela dimensional são desnormalizados, ela normalmente tem um grande número de colunas.

- ✓ As tabelas dimensionais normalmente contêm significativamente menos linhas de dados do que a tabela de factos.

- ✓ Os atributos numa tabela de dimensão são normalmente usados como cabeçalhos de linha e coluna num relatório ou exibição de resultados de consulta. Por exemplo, as descrições textuais num relatório provêm de atributos de dimensão.

**Tipos de modelos dimensionais:**

Existem três tipos básicos de modelos dimensionais (Figura 11), que são:

- ✓ **Modelo estrela (*Star model*):** Os esquemas em estrela têm uma tabela de factos e várias tabelas de dimensões. As tabelas de dimensão não são desnormalizadas.

- ✓ **Modelo de floco de neve (*Snowflake model*):** A normalização e expansão adicionais das tabelas dimensionais num esquema em estrela resultam na implementação de um design *snowflake*. A dimensão é considerada *snowflake* quando as colunas de baixa cardinalidade na dimensão foram removidas para separar tabelas normalizadas que depois ligam-se novamente à tabela de dimensão original.

- ✓ **Modelo de várias estrelas (*Multi-star model*):** Um modelo multi-estrelas é um modelo dimensional que consiste em múltiplas tabelas de factos, unidas através de dimensões.

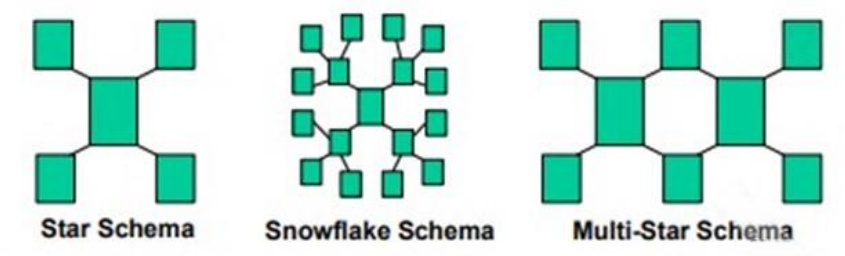



Figura 11 - Tipos de modelos dimensionais

## 3.3 Arquiteturas Data Warehouse

Esta secção explora diferentes arquiteturas de data warehousing, discutindo as suas características e implicações na modelação de dados. Compreender essas arquiteturas é essencial para escolher a estrutura mais adequada para atender às necessidades específicas de uma organização.

### 3.3.1 Contexto

Nesta secção, são exploradas três abordagens arquiteturais para data warehousing, cuja escolha impacta os requisitos de modelação de dados. A arquitetura do data warehouse determinará, ou será determinada pela localização dos data warehouses e data marts, bem como onde o controlo é centralizado. Por exemplo, os dados podem residir centralmente num local gerenciado centralmente, ou podem estar distribuídos local e/ou remotamente, sendo gerenciados centralmente ou de forma independente.

(Ballard, 2012), indica e explica três abordagens arquiteturais, conforme listadas abaixo e representadas na Figura 12:

- ✓ Enterprise Data Warehouse (EDW)
- ✓ Data Marts Independentes
- ✓ Data Marts Dependentes

Essas abordagens arquiteturais podem ser combinadas, sendo o exemplo mais comum a combinação de EDW e data marts dependentes. Nesse caso, os data marts recebem dados do EDW, em vez de diretamente da área de preparação. Essas escolhas arquiteturais têm implicações significativas nos requisitos de modelação de dados e na organização do

ambiente de armazenamento de dados. A compreensão dessas abordagens é fundamental para tomar decisões informadas sobre a estrutura do seu sistema de data warehousing.

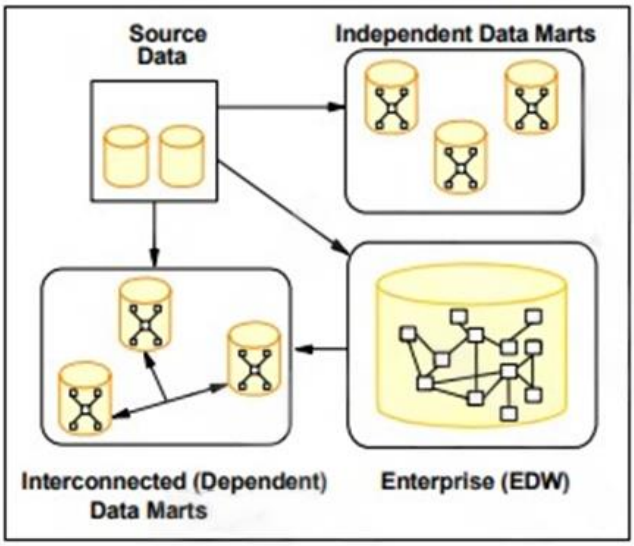



Figura 12 - Arquiteturas de um Data Warehouse (Ballard, 2012)

### 3.3.2 Enterprise Data Warehouse

Um data warehouse empresarial (EDW) é aquele que visa atender a totalidade ou grande parte dos requisitos de negócios para um ambiente de armazenamento de dados totalmente integrado, com um alto grau de acesso e uso de dados entre departamentos ou linhas de negócios. Em outras palavras, o data warehouse é projetado e construído com base nas necessidades globais da empresa. Pode ser considerado um repositório comum para dados de suporte à decisão disponíveis em toda a organização ou numa grande parte desses dados. “Um data warehouse empresarial (EDW) permite uma análise multifuncional que não só descobre o que está a acontecer na empresa, mas também permite aos utilizadores descobrir porque é que certas coisas estão a acontecer.” (Gardner, 1998). O termo "Empresa" é usado aqui para refletir o alcance de acesso e uso de dados, não a estrutura física.

A Figura 13 apresenta um diagrama arquitetural para um data warehouse empresarial.

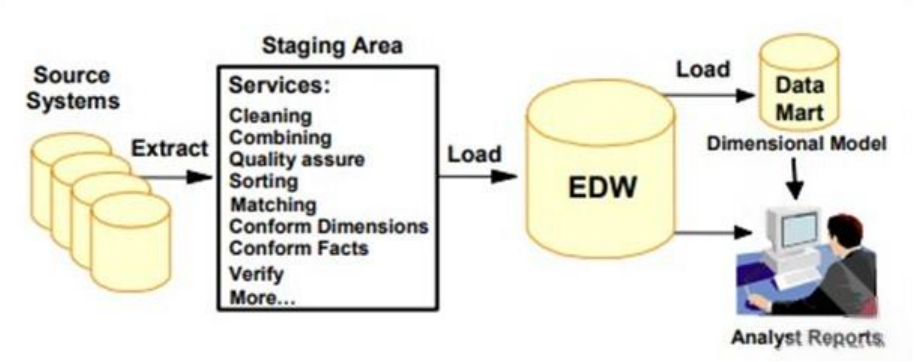



Figura 13 - Arquitetura Data Warehouse empresarial (Ballard, 2012)

Este tipo de data warehouse é caracterizado pela gestão centralizada de todos os dados. No entanto, a centralização não implica necessariamente que todos os dados estejam num único local ou num ambiente de sistemas comum. Ou seja, é centralizado, mas de forma lógica, em vez de ser centralizado fisicamente. O ponto chave é que o ambiente é gerenciado como uma única entidade integrada.

### 3.3.3 Data Mart Independente

Uma arquitetura de data mart independente, como o próprio nome sugere, consiste em data marts autônomos controlados por grupos de trabalho específicos, departamentos ou linhas de negócios. Geralmente, são construídos exclusivamente para atender às necessidades particulares desses grupos de trabalho, departamentos ou linhas de negócios específicos. Embora possa haver exceções, normalmente não há conectividade com data marts noutros grupos de trabalho, departamentos ou linhas de negócios. Portanto, esses data marts não compartilham dimensões e factos conformados entre si. Isso torna-se uma preocupação ao utilizar data marts independentes. Os dados em cada um podem estar em níveis de atualização diferentes, e as definições de dados podem não ser consistentes, mesmo para elementos de dados com o mesmo nome.

Por exemplo, considere-se data mart #1 e data mart #2, conforme exemplificado na Figura 14, ambos com uma dimensão de cliente. No entanto, como eles não partilham dimensões conformadas, isso significa que esses dois data marts precisam de implementar cada um a sua própria versão da dimensão do cliente. São essas decisões que podem resultar, por exemplo, em fontes de dados inconsistentes e desatualizadas nos data marts independentes de uma empresa. Isso pode levar a fontes de dados imprecisas, resultando em tomadas de decisão inconsistentes e não confiáveis.

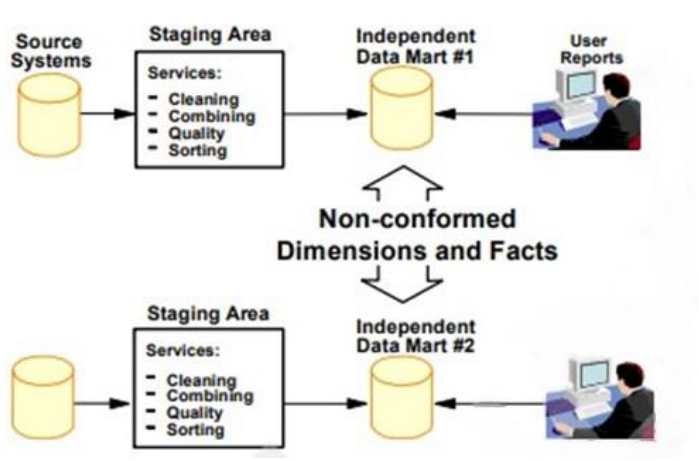



Figura 14 - Arquitetura Data Mart independente (Ballard, 2012)

### 3.3.4 Data Mart Dependente

Um sistema de data mart interconectado é, essencialmente, uma implementação distribuída. Embora os data marts individuais sejam implementados num grupo de trabalho, departamento ou linha de negócios específicos, eles são integrados, ou interconectados, para oferecer uma visão mais global dos dados. Esses data marts estão conectados entre si utilizando, por exemplo, dimensões e factos conformados. Suponha-se que um data mart #1 e um data mart #2 utilizam uma dimensão de cliente. Quando se afirma que esses data marts partilham dimensões conformadas, significa que ambos implementam a mesma versão comum de uma dimensão de cliente. Além disso, cada um desses data marts tipicamente possui uma área de preparação comum. No nível mais elevado de integração, a combinação de todos esses data marts dependentes pode ser considerada como um data warehouse empresarial distribuído.

Na implementação mencionada anteriormente, onde as arquiteturas de EDW (Enterprise Data Warehouse) e data marts dependentes são combinadas, a área de preparação é essencialmente substituída pelo EDW.

## 3.4 DW Tradicional versus DW na Cloud

Esta secção compara as principais diferenças entre data warehouses tradicionais e data warehouses na cloud, destacando os benefícios e desafios de cada abordagem, assim como as situações onde um modelo pode ser preferível ao outro.

### 3.4.1 Análise Comparativa

Um data warehouse na *cloud* oferece "armazenamento como serviço". Um data warehouse tradicional requer muito tempo para configurar o *hardware*, o *software* e a infraestrutura. Também leva muito tempo para otimizar e gerir o sistema. Um data warehouse na nuvem foi concebido para tirar partido de um maior número de utilizadores e aplicações. A escalabilidade num ambiente tradicional é uma tarefa tediosa e consome muitos recursos, enquanto o data warehouse na nuvem permite escalonar para cima ou para baixo instantaneamente, sem complicações. Usando a nuvem, os dados podem ser dimensionados rapidamente, ao contrário do ambiente tradicional, onde isso não é instantâneo conforme as necessidades comerciais evoluem. A implementação de um data warehouse na nuvem é rápida e fácil, ao passo que a implementação de um data warehouse tradicional leva muito tempo.

Os data warehouse em nuvem são eficientes para análises devido ao uso de armazenamento colunar e processamento massivamente paralelo (MPP), resultando num melhor desempenho para consultas complexas. Enquanto o data warehouse tradicional concentra-se na gestão de dados, o data warehouse em nuvem permite que as empresas desloquem o seu foco da gestão de sistemas para a análise. Além disso, o data warehouse em nuvem oferece benefícios financeiros ao eliminar custos iniciais significativos. Não há necessidade de investir em hardware, salas de servidores, problemas de pessoal de TI ou despesas operacionais para manter o data warehouse.

Dessa forma, o data warehouse em nuvem reduz o custo e a complexidade do gerenciamento de sistemas locais, permitindo que os clientes se concentrem em extrair valor dos seus dados, em vez de se preocuparem com a manutenção da infraestrutura de *hardware* e *software*. Na Tabela 2, são indicadas segundo (Rehman, 2018), as principais diferenças entre o data warehouse tradicional e o data warehouse na nuvem.

Tabela 2 - DW tradicional vs. DW na cloud (Rehman, 2018)

| Data warehouse tradicional | Cloud Data warehouse |
|---|---|
| O planeamento de um data warehouse tradicional pode ser uma tarefa grande e exigente. | Não é necessário planear o data warehouse utilizando a *cloud* (nuvem). |
| Um data warehouse tradicional é inflexível e pode resultar em excesso de aprovisionamento e pagamentos excessivos. | Um data warehouse na *cloud* é flexível e cresce automaticamente quando a necessidade aumenta. Utilizar o data warehouse na *cloud* reduz os custos. |
| Um data warehouse tradicional afeta as consultas quando os dados crescem. | Um data warehouse na *cloud* não afeta as consultas quando os dados crescem. |
| Um data warehouse tradicional não "encolhe" quando é subutilizado. | Um data warehouse na *cloud* é automaticamente reduzido para poupar custos quando é subutilizado. |
| Utilizando um data warehouse tradicional, não é possível aumentar ou diminuir os dados a qualquer altura. Demora horas ou dias a configurar *hardware*, *software* e infraestruturas.<br>Também não consegue lidar com o crescente número de utilizadores. | O data warehouse criado para a *cloud* pode ser "ampliado e reduzido a qualquer altura". Não demorará horas ou dias a configurar *hardware*, *software* e infraestruturas. Além disso, permite gerir um número crescente de utilizadores. |
| Um data warehouse tradicional obriga a compra de computação e armazenamento em conjunto. | O data warehouse na *cloud* permite redimensionar os clusters de computação (pagar apenas o que precisa, quando precisa). |
| Um data warehouse tradicional é caro e não é fácil de escalar quando os dados aumentam. | Um data warehouse criado para a *cloud* é fácil e económico de dimensionar e escalar. |

### 3.4.2 Discussão / Análise

O data warehouse atual está a transformar o panorama dos *big data* e *business intelligence*, oferecendo formas cómodas, mas poderosas, de acompanhar as novas tendências. Os rápidos avanços na obtenção de dados e na velocidade de processamento, juntamente com a expansão das capacidades de armazenamento, são de facto impressionantes. Ao longo do tempo, o *hardware* e o desempenho continuam a melhorar. A manutenção, a dimensão ou a atualização de um data warehouse na nuvem é notoriamente mais simples em comparação com data warehouse tradicionais. Ao contrário dos homólogos tradicionais, os data

warehouse na nuvem eliminam a necessidade de tarefas como a manutenção de índices, a limpeza de ficheiros ou a atualização de meta dados. A infraestrutura de nuvem fornece armazenamento económico e capacidades de computação a pedido, aumentando ainda mais a eficiência. Isto reduziu significativamente a complexidade e o tempo de obtenção de valor, limitações que estavam associadas à adoção e utilização bem-sucedida da tecnologia tradicional de data warehouse.

Além disso, os data warehouse em nuvem permitem o escalonamento independente de computação, armazenamento e serviços. Essencialmente, estão a substituir os data warehouse tradicionais como a principal fonte de apoio à decisão e de análise empresarial. A mudança para os data warehouse na nuvem não só facilita os processos de manutenção simplificados, como também contribui para a redução de custos, tornando-se assim uma opção atraente para as organizações que pretendem melhorar os processos de tomada de decisões e as capacidades analíticas (Rehman, 2018). No entanto, há cenários onde os data warehouse tradicionais ainda são essenciais e superiores. Por exemplo, para empresas que preferem manter os seus dados e operações internamente por razões de segurança, conformidade regulatória ou controlo total sobre a infraestrutura, os data warehouse on-premises são indispensáveis. Essas empresas podem ter políticas rigorosas contra o uso de serviços em nuvem ou podem operar em setores onde a segurança dos dados é crítica e a presença física dos servidores é uma exigência. Além disso, para organizações com infraestruturas de TI já estabelecidas e robustas, o custo de transição para a nuvem pode ser proibitivo, tornando os data warehouse tradicionais uma escolha mais prática e econômica. Portanto, apesar das vantagens significativas dos data warehouse na nuvem, os tradicionais ainda mantêm um papel crucial em cenários específicos onde controlo, segurança e conformidade são prioritários.

## 3.5 Ferramentas Data Warehouse

No atual cenário empresarial, a gestão eficiente e estratégica de dados desempenha um papel crucial no sucesso das organizações. Nesse contexto, o uso de ferramentas de Data Warehouse torna-se imperativo para armazenar, processar e analisar grandes volumes de dados estruturados e semiestruturados. Esta secção explora duas das principais ferramentas de Data Warehouse na nuvem, Snowflake e Amazon Redshift, destacando as suas arquiteturas, desempenho, escalabilidade, capacidades de carregamento de dados e implementação.

### 3.5.1 Snowflake

Inicialmente construído com base em AWS (Amazon Web Services), o Snowflake é um armazém de dados na nuvem com tudo incluído para dados estruturados e semiestruturados, fornecido como *"software-as-a-service"* (SaaS). Como cliente, não se precisa de selecionar,

instalar ou gerir qualquer hardware virtual ou físico, exceto para configurar o tamanho e o número de clusters de computação. O resto das tarefas de manutenção são realizadas pelo Snowflake, o que torna esta solução praticamente sem servidor. Ao contrário das ofertas tradicionais de armazenamento, o Snowflake fornece soluções analíticas e de armazenamento de dados mais flexíveis, mais rápidas e mais fáceis de utilizar.

**Arquitetura:**
A arquitetura do Snowflake foi concebida nativamente para a nuvem e combinada com um motor de consulta SQL inovador. O Snowflake inclui três camadas principais, como é possível ver na Figura 15, tais como armazenamento de bases de dados, processamento de consultas e serviços de nuvem. Existe um repositório de dados centralizado para uma única cópia de dados que pode ser acedida a partir de todos os nós de computação independentes.

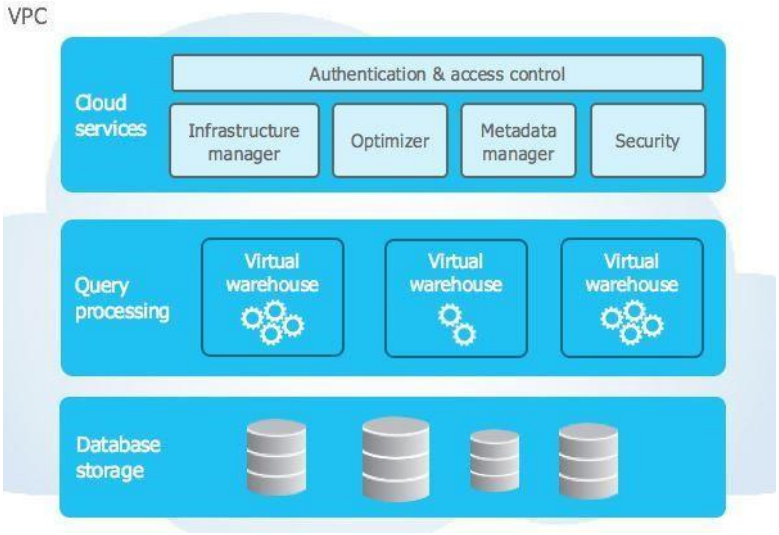



Figura 15 - Ilustração arquitetura Snowflake (Snowflake, 2023)

**Desempenho:**
Graças ao conceito de computação e armazenamento separados, o Snowflake permite cargas de trabalho simultâneas, o que significa que os utilizadores podem executar várias consultas ao mesmo tempo. As cargas de trabalho não se afetam mutuamente, o que resulta num desempenho mais rápido (de acordo com um dos *benchmarks* (Tridant, 2020), o Snowflake é capaz de processar 6 a 60 milhões de linhas de dados de 2 a 10 segundos).

**Escalabilidade**:
O Snowflake permite um escalonamento ininterrupto e sem interrupções (tanto horizontal como vertical), alimentado por uma arquitetura de dados partilhados em vários clusters. Não requer o envolvimento de um operador ou administrador de base de dados para escalar, uma vez que o *software* trata de todo o escalonamento automaticamente de acordo com a procura comercial. Esta é uma enorme vantagem para as empresas mais pequenas com recursos limitados.

**Carregamento de dados:**
O Snowflake suporta as abordagens de integração de dados ELT e ETL, o que significa que a transformação de dados pode ocorrer durante ou após o carregamento. A abordagem ELT ajuda a capturar dados brutos e, em seguida, a encontrar o melhor caso de utilização para os mesmos.

**Implementação:**
O Snowflake é considerado um dos produtos de data warehouse mais intuitivos e mais simples de utilizar e evoca uma experiência sem servidor. Tendo herdado muitas características das bases de dados relacionais e combinando-as com os princípios da nuvem, o serviço promete um arranque rápido e fácil. As dimensões do cluster de computação por armazém podem ser configuradas em pormenor. Ainda assim a configuração do Snowflake ainda requer conhecimentos e competências sólidos em SQL, bem como uma boa compreensão da arquitetura do armazém de dados.

**Adequado para:** Empresas que procuram um DW de fácil implementação com escalonamento automático quase ilimitado e desempenho de alto nível beneficiarão da utilização do Snowflake.

### 3.5.2 Amazon Redshift

Parte da plataforma de computação em nuvem da Amazon, o Redshift é um software de armazenamento de dados baseado em nuvem para empresas. A plataforma permite o processamento rápido de conjuntos de dados maciços. Não só é adequada para a análise de dados de qualidade, como também fornece consultas automáticas de simultaneidade de acordo com a procura de carga de trabalho. O Redshift é uma solução mais autogerida, o que significa que os engenheiros terão de dedicar tempo à gestão de recursos e servidores.

**Arquitetura:**
O Redshift foi concebido com a arquitetura MPP (Processamento Massivamente Paralelo) sem partilha (Figura 16). Inclui clusters de armazém de dados com nós de computação divididos em fatias de nós. Os nós de computação individuais são atribuídos com o código pelo nó líder. O sistema comunica com as aplicações cliente utilizando controladores JDBC e ODBC padrão da indústria. A tecnologia pode ser integrada com a maioria das aplicações cliente baseadas em SQL, ETL, BI, análise de dados e ferramentas de extração de dados existentes.

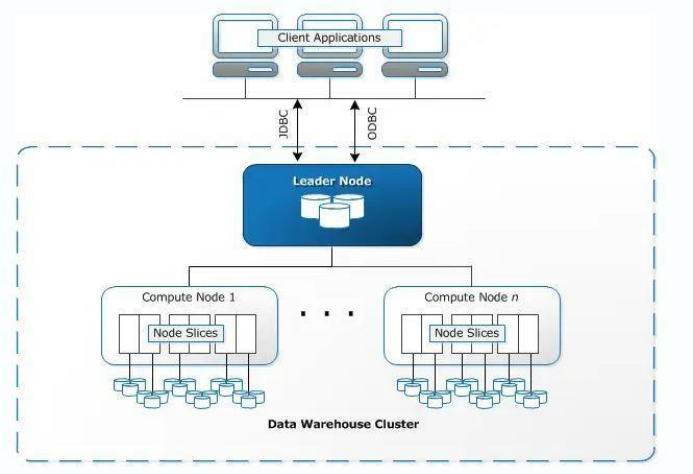



Figura 16 - Ilustração arquitetura Amazon Redshift (AWS, 2023)

**Desempenho:**
Embora o desempenho seja globalmente bom na maioria dos tipos de dados, é bastante baixo quando se utilizam dados semiestruturados (por exemplo, ficheiros JSON). Para um desempenho ótimo, recomenda-se que os utilizadores optem pelo conceito de chaves de distribuição. São colunas utilizadas para definir um segmento de base de dados que armazena uma determinada linha de dados.

**Escalabilidade:**
O Redshift permite o escalonamento horizontal e vertical. Está pré-configurado para suportar 500 ligações simultâneas e até 50 consultas simultâneas a serem executadas ao mesmo tempo num cluster. Isto significa que até 500 utilizadores podem executar até 50 consultas em qualquer momento num cluster. Caso seja necessário processar mais consultas de leitura simultâneas, o Redshift fornece a funcionalidade de escalonamento de simultaneidade que adiciona automaticamente a capacidade de outro cluster.

**Carregamento de dados:**
Embora suporte abordagens ELT/ETL e comandos DML (linguagem de manipulação de dados) padrão, incluindo *INSERT*, o Redshift (AWS, 2023) afirma que a maneira mais eficaz de carregar dados nas suas tabelas é usar o comando *COPY*. Com este comando, é possível trabalhar simultaneamente com vários fluxos de dados e ler dados de vários ficheiros de dados. A plataforma também oferece recursos de *streaming* de dados em tempo real.

**Implementação:**
O Redshift automatiza algumas das responsabilidades de configuração de clusters mais demorosas, tais como cópias de segurança de dados, correções, replicações e a facilitação da administração de um armazém. De uma forma ou de outra, deve haver um certo nível de experiência em engenharia de dados e gestão de servidores para trabalhar com o Redshift: Um *DevOps* ou engenheiro de dados deve configurar clusters e definir a alocação de

memória, computação e armazenamento. Por esses motivos, o Redshift enquadra-se mais no lado autogerenciado do espetro.

**Adequado para:**
O Redshift foi inicialmente concebido para o armazenamento de grandes volumes de dados. Se a empresa lida com dados de grande escala e as consultas necessitam de respostas rápidas, deve-se definitivamente considerar seriamente o Redshift. A plataforma também é adequada para empresas que procuram um armazém de dados com um modelo de preços transparente e com poucos ou nenhuns custos administrativos gerais.

### 3.5.3 Discussão / Análise

Em resumo, ao analisar as ferramentas de data warehouse Snowflake e Amazon Redshift, observa-se duas soluções robustas, cada uma com as suas próprias características distintas. O Snowflake destaca-se pela sua abordagem praticamente sem servidor, oferecendo uma experiência intuitiva e fácil implementação. A sua arquitetura nativa na nuvem, com computação e armazenamento separados, proporciona desempenho rápido e escalabilidade contínua, adaptando-se automaticamente à demanda comercial. Suporte às abordagens ELT e ETL, junto com a capacidade de processar cargas de trabalho simultâneas, tornam o Snowflake adequado para empresas que procuram flexibilidade e desempenho de alto nível.

Por outro lado, o Amazon Redshift, integrado à plataforma de computação em nuvem da Amazon, apresenta uma arquitetura MPP sem partilha. Oferece escalabilidade horizontal e vertical, sendo inicialmente concebido para armazenar grandes volumes de dados. Embora possua um desempenho globalmente bom, destaca-se pela eficiência quando lidando com dados estruturados. O Redshift demanda uma gestão mais autogerida, exigindo conhecimentos sólidos em engenharia de dados e administração de servidores. Ambas as soluções suportam abordagens ELT/ETL, proporcionando flexibilidade na transformação de dados. Enquanto o Snowflake se destaca pela facilidade de uso e escalabilidade automática, o Redshift foca-se em grandes volumes de dados e consultas rápidas, demandando maior intervenção na gestão por parte dos engenheiros.

Em última análise, a escolha entre Snowflake e Amazon Redshift dependerá das necessidades específicas de cada empresa. O Snowflake pode ser a escolha ideal para organizações que valorizam a simplicidade, escalonamento automático e desempenho robusto, enquanto o Redshift pode ser mais apropriado para empresas que lidam com grandes volumes de dados estruturados, com a disposição para gerenciar mais ativamente os recursos do Data Warehouse.

## 3.6 ETL e ELT

Num mundo orientado por dados, as organizações dependem cada vez mais de volumes consideráveis de informações provenientes de diversas fontes para influenciar decisões fundamentadas. A integração de dados tornou-se um aspeto crucial na gestão dessa informação, e duas abordagens populares surgiram para atender a essas necessidades: Extração, Transformação, Carregamento (ETL) e Extração, Carregamento, Transformação (ELT).

### 3.6.1 ETL e ELT: Definições

Dentro das fases da construção de um DW, o ETL/ELT é uma das tarefas com maior custo, tanto para o tempo do projeto quanto para os recursos como associar e unificar os dados de diferentes fontes, com formatos de estruturas diferentes, onde a variedade de fontes e os esquemas dificultam o trabalho. (Kimball, 2008).

**ETL:** Extração, Transformação, Carregamento

O ETL é um processo de integração de dados que envolve extrair informações de diversas fontes, transformá-las num formato consistente e padronizado, para então carregá-las num armazém de dados alvo, como um data warehouse. Esse procedimento garante que os dados estejam limpos, precisos e prontos para análise e relatórios.

**ETL:** Extração, Carregamento, Transformação

O ELT é uma abordagem alternativa à integração de dados que também envolve a extração, transformação e carregamento de dados. No entanto, no ELT, os dados são inicialmente carregados no sistema de destino e, em seguida, transformados dentro do próprio sistema. Essa abordagem tira proveito das capacidades de processamento das plataformas de dados modernas para realizar transformações de maneira mais eficiente.

### 3.6.2 ETL e ELT: Comparação de processos

"A principal questão quando se trata de comparar ETL e ELT é obviamente a sequência em que as etapas de extração, carregamento e transformação são executadas numa pipeline de dados." (Myrianthous, 2023).

- ✓ **Extração**

Tanto no ETL quanto no ELT, o processo de extração é o mesmo, envolvendo a obtenção de dados brutos de diferentes fontes, como bases de dados, *spreadsheets*, APIs ou provedores externos de dados.

- ✓ **Transformação**

A localização da transformação de dados é a principal diferença entre ETL e ELT. (Singhal, 2022)

- No ETL, os dados são transformados antes de serem carregados no sistema de destino, geralmente utilizando uma ferramenta ou mecanismo separado para realizar limpeza, filtragem, agregação e enriquecimento de dados.

- No ELT, os dados são inicialmente carregados no sistema de destino, sendo a transformação realizada dentro do próprio sistema. Isso aproveita a potência de processamento e a escalabilidade de plataformas de dados modernas, como data lakes ou data warehouses baseados em nuvem.

✓ **Carregamento**

Tal função requer a verificação da integridade dos dados, otimização do processo de carga e suporte às necessidades do processo de carga, como a eliminação e inclusão de índices. (Hokama, 2004)

- No ETL, os dados transformados são carregados no sistema de destino, podendo envolver carregamento em lote ou carregamento incremental, dependendo dos requisitos.

- No ELT, os dados brutos são carregados diretamente no sistema de destino, e o processo de transformação ocorre após o carregamento dos dados.

### 3.6.3 Escolher entre ETL e ELT

Ao decidir entre ETL e ELT, segundo (Halder, 2023) as organizações devem considerar vários fatores, incluindo:

**Volume de Dados:**
Organizações com grandes volumes de dados podem beneficiar do ELT, pois ele pode fornecer melhor desempenho e escalabilidade ao aproveitar o poder de processamento das plataformas de dados modernas.

**Requisitos de Qualidade dos Dados:**
Se a qualidade dos dados for uma alta prioridade, o ETL pode ser a escolha preferida, pois geralmente envolve etapas mais robustas de validação e limpeza de dados. No entanto, organizações ainda podem implementar medidas de qualidade de dados dentro de um processo ELT, se necessário.

**Requisitos de Processamento:**
Para processamento de dados em tempo real ou quase real, o ELT pode ser mais adequado, pois pode oferecer carregamento e transformação de dados mais rápidos. Os processos ETL,

por outro lado, podem ser mais lentos devido à necessidade de transformar dados antes de carregá-los no sistema de destino.

**Compatibilidade do Sistema:**
Organizações com data warehouse *legacy* ou sistemas que não possuem as capacidades de processamento necessárias para transformações dentro do sistema podem ser mais adequadas aos processos ETL. No entanto, organizações que utilizam plataformas de dados modernas, como data lakes ou data warehouses baseados em nuvem, podem beneficiar das vantagens de desempenho e escalabilidade do ELT.

### 3.6.4 Ferramentas ETL

O processo de ETL (Extração, Transformação e Carregamento) é fundamental no desenvolvimento de um armazém de dados, permitindo a integração, limpeza e formatação de dados provenientes de várias fontes antes de serem carregados no sistema de armazenamento central. Diversas ferramentas ETL estão disponíveis no mercado, cada uma com características específicas que podem atender a diferentes necessidades e contextos. Esta secção apresenta algumas das principais ferramentas ETL, destacando as suas funcionalidades e características.

#### 3.6.4.1 Apache Nifi
Apache NiFi é uma plataforma robusta de integração de dados que automatiza o fluxo de dados entre diferentes sistemas de software. Apache NiFi é amplamente reconhecida pela sua capacidade de mover dados de forma eficiente, segura e em tempo real.

**Características Principais:**

- ✓ **Interface de Arrastar e Soltar:** Possui uma interface gráfica intuitiva que permite aos utilizadores projetar fluxos de dados complexos com facilidade.
- ✓ **Processamento em Tempo Real:** Suporta o processamento de dados em tempo real, essencial para ambientes onde a latência baixa é crítica.
- ✓ **Gestão de Fluxo de Dados:** Oferece controlo granular sobre o fluxo de dados, incluindo priorização, roteamento e transformação.
- ✓ **Segurança:** Implementa funcionalidades robustas de segurança, como autenticação, autorização e criptografia.
- ✓ **Escalabilidade:** Pode ser escalado horizontalmente para lidar com grandes volumes de dados.

**Pontos Fracos:**

- ✓ **Curva de Aprendizado:** Pode ser complexo para novos utilizadores, especialmente aqueles sem experiência em integração de dados.
- ✓ **Consumo de Recursos:** Pode consumir muitos recursos do sistema, especialmente em configurações complexas ou de alta carga.

- ✓ **Documentação:** Embora a comunidade seja ativa, a documentação pode ser menos completa e detalhada em comparação com algumas ferramentas comerciais.

#### 3.6.4.2 Talend

Talend é uma plataforma de integração de dados *open-source* que oferece uma ampla gama de funcionalidades ETL. É conhecida pela sua interface amigável e suporte a diversas fontes de dados, facilitando o desenvolvimento e manutenção de processos ETL complexos.

**Características Principais:**

- ✓ **Conectividade Extensiva:** Suporte a uma vasta gama de fontes de dados, incluindo bases de dados, APIs e serviços web.
- ✓ **Componentes Reutilizáveis:** Disponibilidade de uma biblioteca de componentes reutilizáveis que aceleram o desenvolvimento.
- ✓ **Escalabilidade:** Capacidade de lidar com grandes volumes de dados, adequada para ambientes corporativos.

**Pontos Fracos:**

- ✓ **Desempenho:** Pode apresentar problemas de desempenho em cenários de dados extremamente grandes ou complexos.
- ✓ **Licenciamento:** A versão *open-source* possui limitações em comparação com a versão paga, que pode ser cara.
- ✓ **Complexidade de Configuração:** Pode ser complexa de configurar e otimizar para utilizadores menos experientes.

### 3.6.5 Discussão / Análise

No complexo ecossistema da integração de dados, a dicotomia entre ETL e ELT reflete a diversidade de desafios enfrentados pelas organizações modernas na gestão e análise de grandes volumes de informações. A escolha entre essas abordagens, longe de ser uma decisão binária, revela-se uma jornada adaptativa, onde a estratégia ideal muitas vezes reside na sinergia entre processos ETL e ELT. A adoção de uma abordagem híbrida, que integra elementos de ambas as metodologias, emerge como uma resposta inteligente às nuances específicas enfrentadas por diferentes organizações. Ao realizar algumas transformações durante o processo ETL e outras dentro do sistema de destino através do ELT, as empresas podem alcançar um equilíbrio harmonioso entre a qualidade dos dados, desempenho e escalabilidade, alinhando-se precisamente com as suas necessidades únicas.

As ferramentas e tecnologias desempenham um papel fundamental nesse cenário dinâmico, oferecendo um arsenal diversificado para a construção de pipelines de dados robustos. Ao considerar fatores cruciais, como volume de dados, requisitos de qualidade, necessidades de processamento e compatibilidade do sistema, as organizações podem tomar decisões informadas sobre a estratégia de integração de dados mais adequada para os seus objetivos

específicos. A seleção criteriosa de ferramentas e a implementação de uma estratégia de integração bem alinhada não apenas capacitam as organizações a gerenciar efetivamente os seus dados, mas também a desbloquear *insights* valiosos que impulsionarão o seu sucesso.

Assim, ao enfrentar os desafios e oportunidades presentes no cenário atual, a integração de dados emerge não apenas como uma necessidade operacional, mas como um catalisador estratégico para a obtenção de conhecimentos profundos e informados, fundamentais para a excelência num mundo orientado por dados.

## 3.7 Sumário

A secção 3.1 destina-se a introduzir o conceito essencial de data warehouse (DW), esclarecendo o que é e delineando as diferenças cruciais entre dados operacionais e dados armazenados num DW. Esta escolha é motivada pela centralidade do DW no contexto de gestão de dados, sendo uma peça-chave na estrutura que sustenta a tomada de decisões informadas no controlo de qualidade. A secção 3.2 aborda a modelação de dados, destacando a importância desse processo e apresentando técnicas que contribuem para uma representação eficiente e compreensível dos dados. Este tópico é crucial, pois a qualidade do modelo de dados influencia diretamente a eficácia das análises realizadas no DW. A secção 3.3 explora as diferentes arquiteturas de data warehouse, desde o contexto geral até a distinção entre Enterprise Data Warehouse, Data Mart Independente e Data Mart Dependente. Esta abordagem visa proporcionar uma compreensão mais profunda das diferentes estruturas que podem ser implementadas, possibilitando uma análise mais criteriosa na escolha da arquitetura mais adequada. A secção 3.4 compara o DW tradicional com o DW na nuvem, oferecendo uma análise crítica das vantagens e desvantagens associadas a cada abordagem. Esta comparação é relevante, uma vez que a escolha entre implementações tradicionais e baseadas na nuvem tem implicações significativas na visão e estratégia de desenvolvimento.

A secção 3.5 explora o processo de escolha de um data warehouse adequado e possíveis ferramentas a utilizar, considerando os fatores críticos que devem ser ponderados. Esta secção é essencial para a visão da solução, pois a escolha de um DW alinhado com as necessidades específicas do *software* EGITRON Quality Control é crucial para o sucesso da implementação proposta. A secção 3.6, dedicada aos processos de ETL e ELT, proporciona uma compreensão profunda das diferentes abordagens para a integração de dados, destacando as suas vantagens e desvantagens. Esta análise é crucial para a visão da solução, pois influencia diretamente a forma como os dados são tratados, transformados e carregados no DW, afetando a qualidade final dos dados disponíveis para análises.

Cada secção do estado da arte foi escolhida com base na sua contribuição para a construção de um conhecimento sólido e contextualizado. A interligação entre estas secções é essencial para formar uma visão abrangente que possa orientar eficazmente a abordagem à solução do problema proposto.

# 4 Desenvolvimento

Conforme especificado na secção 3.4.2 e 3.5 e tendo em conta as tecnologias que acarretam custos adicionais, assim como as preferências de algumas empresas clientes da EGITRON, que optam por manter os seus dados e bases de dados em servidores locais, foi desenvolvida uma solução utilizando tecnologias gratuitas e implementando o armazenamento dos dados localmente.

Neste capítulo, irá ser detalhado o desenvolvimento do sistema proposto, abordando desde a sua especificação até à implementação, validação e transição para o ambiente de produção. Este desenvolvimento é fundamental para garantir que o sistema atende aos requisitos estabelecidos e funciona de forma eficiente e eficaz.

Inicialmente, será apresentada uma visão geral sobre os componentes do sistema, incluindo os pontos de interação e a infraestrutura utilizada. Em seguida, serão descritos os passos práticos realizados para a construção do sistema, incluindo a configuração e integração das diversas tecnologias envolvidas. A validação do sistema será abordada, detalhando os métodos utilizados para assegurar que os dados e as operações são corretas e que o desempenho é adequado. Será feita uma análise comparativa e de desempenho para garantir que o sistema está a funcionar conforme o esperado.

Posteriormente, serão descritas as etapas seguidas para preparar o ambiente de produção e colocar o sistema em operação, incluindo a monitorização e manutenção contínua para garantir a sua estabilidade e eficiência. Por fim, serão discutidas possíveis melhorias futuras na implementação, com foco em suportar múltiplas bases de dados simultâneas, melhorar a integração e monitorização, e otimizar o desempenho geral do sistema.

Este capítulo visa fornecer uma visão abrangente e detalhada de todo o processo de desenvolvimento, garantindo que cada etapa seja bem compreendida e documentada, facilitando a replicação e possíveis aprimoramentos futuros.

## 4.1 Requisitos Funcionais e Não Funcionais

Para garantir o sucesso da implementação do data warehouse utilizando Apache NiFi e os endpoints criados, foram estabelecidos requisitos funcionais e não funcionais detalhados. Esta secção descreve esses requisitos, fundamentais para alcançar os objetivos do projeto.

### 4.1.1 Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades específicas que o sistema deve oferecer para atender às necessidades de negócio identificadas. Estes incluem:

1. **Extração de Dados:**
   O sistema deve ser capaz de extrair dados das bases de dados de produção e endpoints de uma API.
2. **Transformação e Limpeza de Dados:**
   Os dados extraídos devem ser transformados conforme necessário, utilizando processadores para remodelar dados JSON e extrair as informações específicas e necessárias.
3. **Carregamento no Data Warehouse:**
   Após a transformação, os dados devem ser carregados no data warehouse utilizando consultas SQL para inserção e atualização nas tabelas dimensionais e de factos.
4. **Atualização Incremental:**
   Implementar mecanismos para atualização incremental dos dados no Data Warehouse, permitindo apenas a inserção de novos registos ou atualização de registos existentes desde a última carga.

### 4.1.2 Requisitos Não Funcionais

Os requisitos não funcionais definem as características que o sistema deve possuir para além das funcionalidades específicas. Estes incluem (e prosseguindo com a numeração):

5. **Performance:**
   Garantir que o sistema possa lidar com grandes volumes de dados. Assegurar que se pode obter os mesmos resultados no data warehouse em um tempo menor do que diretamente na base de dados de produção.
6. **Integridade dos Dados:**
   Garantir a integridade referencial dos dados, especialmente ao inserir dados na tabela

de factos após terem sido corretamente associados às tabelas dimensionais correspondentes.

7. **Segurança:**
   Implementar medidas de segurança para proteger dados sensíveis durante o movimento e armazenamento, utilizando conexões seguras e criptografia quando necessário, especialmente no que toca à comunicação com os endpoints realizados.
8. **Facilidade de Manutenção:**
   O sistema deve ser fácil de manter e gerir, com monitorização contínua e capacidade de ajustar a configuração conforme necessário sem interrupções significativas.
9. **Compatibilidade:**
   Garantir compatibilidade com diferentes sistemas operativos e ambientes de infraestrutura, utilizando padrões abertos e componentes de software amplamente suportados.
10. **Documentação:**
    Fornecer documentação abrangente do sistema, incluindo arquitetura, processos de ETL, configuração do Apache NiFi e esquema dimensional, para facilitar a manutenção e entendimento do sistema por parte dos utilizadores e administradores.

## 4.2 Casos de Uso

Para ilustrar a aplicação prática do sistema proposto e demonstrar como a implementação de um armazém de dados (DW) e um processo de ETL resolvem os problemas identificados, foram definidos vários casos de uso. Estes casos de uso detalham as interações entre os atores principais e o sistema, descrevendo os passos necessários para executar as funcionalidades essenciais. Através dos casos de uso, é possível visualizar como o sistema extrai, transforma e carrega dados de diversas fontes, assegurando a integridade e atualização incremental dos dados no DW. Este processo garante uma análise abrangente e estratégica dos dados, contribuindo para a otimização dos processos de produção e melhorando a tomada de decisões nas empresas que utilizam o software EGITRON Quality Control.

1. **Extração de Dados de Fontes Heterogéneas**

- ✓ **Ator:** Administrador de dados
- ✓ **Descrição:** O sistema extrai dados de várias fontes de dados, como bases de dados de produção e endpoints de API.
- ✓ **Pré-condições:** Conexões estabelecidas com as fontes de dados.
- ✓ **Pós-condições:** Dados extraídos disponíveis para transformação.
- ✓ **Fluxo Principal:**
  - O Administrador de dados inicia o processo de extração.
  - O sistema conecta-se às diferentes fontes de dados.
  - Dados são extraídos e armazenados temporariamente para transformação.
- ✓ **Fluxos Alternativos:**
  - Falha na conexão com uma fonte de dados.

2. **Transformação e Limpeza de Dados**

- ✓ **Ator:** Administrador de dados
- ✓ **Descrição:** O sistema transforma e limpa os dados extraídos para adequá-los ao modelo dimensional do DW.
- ✓ **Pré-condições:** Dados extraídos disponíveis.
- ✓ **Pós-condições:** Dados transformados prontos para carregar.
- ✓ **Fluxo Principal:**
  - O Administrador de Dados define as regras de transformação.
  - O sistema aplica as regras de transformação e limpeza.
- ✓ **Fluxos Alternativos:**
  - Erro nas regras de transformação.

3. **Carregamento de Dados no Data Warehouse**

- ✓ **Ator:** Administrador de dados
- ✓ **Descrição:** O sistema carrega os dados transformados no DW, atualizando as tabelas dimensionais e de factos.
- ✓ **Pré-condições:** Dados transformados disponíveis na área de *staging*.
- ✓ **Pós-condições:** Dados carregados no DW.
- ✓ **Fluxo Principal:**
  - O Administrador de dados inicia o processo de carregamento.
  - O sistema insere e atualiza os dados nas tabelas dimensionais.
  - O sistema insere e atualiza os dados nas tabelas de factos.
- ✓ **Fluxos Alternativos:**
  - Conflito de integridade referencial.

4. **Atualização Incremental dos Dados**

- ✓ **Ator:** Administrador de dados
- ✓ **Descrição:** O sistema atualiza incrementalmente os dados no DW, inserindo novos registos e atualizando os existentes.
- ✓ **Pré-condições:** Dados anteriores carregados no DW.
- ✓ **Pós-condições:** DW atualizado com novos dados.
- ✓ **Fluxo Principal:**
  - O Administrador de dados define a frequência de atualização incremental.
  - O sistema identifica novos registos e alterações desde a última atualização.
  - Dados novos e atualizados são carregados no DW.
- ✓ **Fluxos Alternativos:**
  - Falha na atualização incremental.

5. **Validação da Integridade dos Dados**

- ✓ **Ator:** Analista de dados
- ✓ **Descrição:** O sistema valida a integridade dos dados carregados no DW, assegurando que correspondem aos dados de produção.
- ✓ **Pré-condições:** Dados carregados no DW.
- ✓ **Pós-condições:** Dados validados e relatórios gerados.
- ✓ **Fluxo Principal:**
  - O Analista de Dados solicita a validação da integridade dos dados.
  - O sistema compara os dados do DW com os dados de produção.
  - Relatórios de integridade são gerados e analisados pelo Analista de dados.
- ✓ **Fluxos Alternativos:**
  - Discrepância nos dados.

## 4.3 Análise de Alternativas de Design

Nesta secção, são analisadas várias alternativas de design para a solução de ETL. O objetivo é identificar a solução mais adequada que atenda às necessidades do projeto e das partes interessadas.

### 4.3.1 Apache Nifi com Data Warehouse no Azure

- ✓ **Descrição:** A primeira alternativa considera o uso do Apache Nifi para o processo ETL e um data warehouse hospedado na plataforma Azure. O Apache Nifi é uma ferramenta robusta para automação de fluxo de dados, enquanto o Azure oferece um ambiente escalável e integrado para armazenamento e análise de dados.
- ✓ **Vantagens:**
  - **Integração:** Boa integração com outras soluções da Microsoft.
  - **Escalabilidade:** A capacidade de escalar recursos conforme a demanda.
  - **Alta Disponibilidade:** Serviços gerenciados com alta disponibilidade garantida.
- ✓ **Desvantagens:**
  - **Custo:** Custos recorrentes associados ao uso da nuvem.
- ✓ **Contexto:** Ideal para organizações que já utilizam tecnologias Microsoft e procuram uma solução integrada.

### 4.3.2 Talend com Data Warehouse on-premises

- ✓ **Descrição:** A segunda alternativa envolve o uso do Talend para o processo ETL e um data warehouse hospedado *on-premises*. Talend é uma plataforma ETL poderosa e flexível, e um Data Warehouse *on-premises* oferece controlo total sobre os dados.

- ✓ **Vantagens:**
  - **Controlo Total:** Maior controlo sobre o ambiente e os dados.
  - **Custos:** Sem custos recorrentes de nuvem, apenas custos iniciais de *setup*.
- ✓ **Desvantagens:**
  - **Manutenção:** Necessidade de manutenção contínua da infraestrutura.
  - **Custo Inicial:** Investimento inicial elevado em *hardware* e *software*.
- ✓ **Contexto:** Adequado para organizações com infraestrutura de TI robusta e equipa de suporte disponível.

### 4.3.3 Apache Nifi com Google BigQuery

- ✓ **Descrição:** A terceira alternativa propõe o uso do Apache Nifi para orquestração de ETL e Google BigQuery como data warehouse. Apache Nifi é uma plataforma de orquestração de *workflow* altamente configurável, e Google BigQuery é um data warehouse escalável e performático.
- ✓ **Vantagens:**
  - **Escalabilidade:** BigQuery oferece escalabilidade quase ilimitada.
  - **Performance:** Alta performance para consultas de grandes volumes de dados.
  - **Integração:** Boas integrações com outras ferramentas do Google Cloud.
- ✓ **Desvantagens:**
  - **Curva de Aprendizagem:** Apache Nifi pode ter uma curva de aprendizagem íngreme.
- ✓ **Contexto:** Ideal para empresas que já estão na nuvem do Google ou que planeiam migrar para lá.

### 4.3.4 Comparação e Seleção da Alternativa

- ✓ **Comparação**:
  - **Custo:** *On-premises* pode ter menor custo a longo prazo, enquanto nuvem oferece menor custo inicial.
  - **Escalabilidade:** Soluções na nuvem (Azure, BigQuery) são mais escaláveis.
  - **Facilidade de Uso:** Apache Nifi e Talend são conhecidos pelas suas interfaces amigáveis.
  - **Integração:** Depende do ecossistema tecnológico da organização (Microsoft, Google).
  - **Suporte e Manutenção:** Soluções na nuvem têm menos necessidade de manutenção de infraestrutura.
- ✓ **Seleção Justificada:**

  A solução selecionada é a combinação de Apache Nifi com data warehouse *on-premises*. Essa escolha é baseada na necessidade de controlo total sobre os dados e a infraestrutura, assim como as preferências de algumas empresas clientes da EGITRON, que optam por manter os seus dados e bases de dados em servidores

locais, além de uma análise de custo-benefício que favorece a ausência de custos recorrentes de nuvem. A robustez do Apache Nifi como ferramenta ETL, junto com a flexibilidade e controlo de um data warehouse *on-premises*, oferecem uma solução equilibrada para os requisitos do projeto.

## 4.4 Diagrama de Componentes da Nova Solução

O diagrama de componentes apresentado na Figura 17 ilustra a nova solução proposta, destacando os principais elementos do sistema e o fluxo de dados entre eles. Este diagrama é essencial para entender a estrutura da implementação e as interações entre os diversos componentes, proporcionando uma visão clara de como os dados são extraídos, transformados, carregados e finalmente utilizados para análise e geração de relatórios.

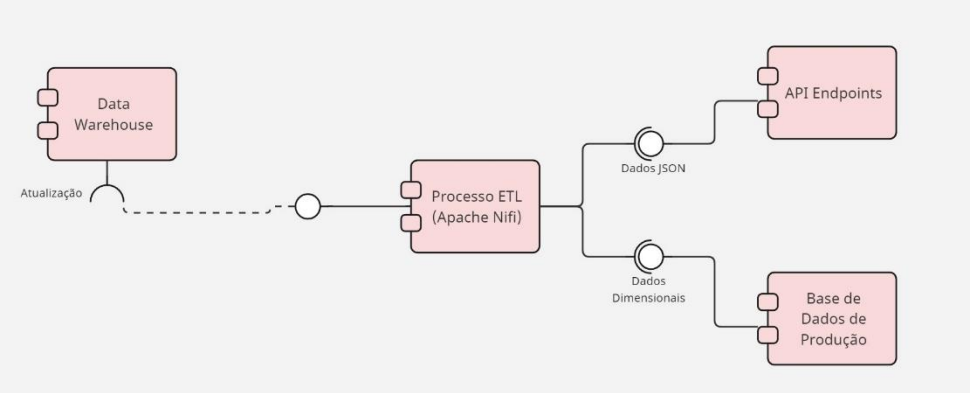



Figura 17 - Diagrama de Componentes

**Componentes:**

- ✓ **Processo ETL (Apache Nifi):**
  - **Descrição:** O processo de ETL é o núcleo da solução, responsável por extrair dados de várias fontes, realizar as transformações necessárias e carregá-los no data warehouse.
  - **Interação:** Recebe dados das fontes de dados e transfere-os para o data warehouse.
- ✓ **Fontes de Dados:**
  - **Descrição:** Incluem as bases de dados de produção e endpoints da API de onde os dados são extraídos.
  - **Interação:** Fornecem dados ao processo de ETL.
- ✓ **Data Warehouse:**
  - **Descrição:** Armazena os dados transformados de forma estruturada, permitindo análise e relatórios eficientes.
  - **Interação:** Recebe dados do processo de ETL.

Cada um dos componentes desempenha um papel crucial na solução ETL, garantindo a integração eficiente e eficaz de dados de várias fontes, a sua transformação conforme necessário e sua disponibilização para análise e relatórios. Este diagrama facilita a compreensão das interações e dependências entre os componentes, servindo como uma base sólida para a implementação e manutenção da solução.

## 4.5 Principais Processos da Solução

A implementação da solução para a integração e análise de dados do software EGITRON Quality Control envolve diversos processos essenciais que garantem a extração, transformação, carregamento e atualização dos dados no data warehouse (DW). Abaixo, são apresentados os principais processos da solução, detalhando as suas etapas por meio de diagramas de atividade. Esses diagramas são fundamentais para entender o fluxo de dados e as operações realizadas em cada etapa do processo ETL (Extração, Transformação, Carregamento).

**Extração de Dados:**
O processo de extração de dados, ilustrado na Figura 18, é o primeiro passo onde os dados são obtidos de várias fontes heterogéneas, incluindo bases de dados de produção e *endpoints* de API. Esse processo garante que os dados necessários estejam disponíveis para as etapas subsequentes de transformação e carregamento.

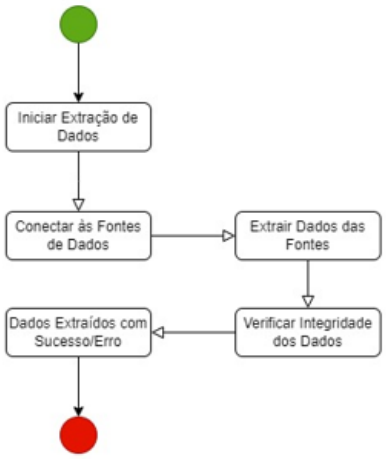



Figura 18 - Processo de Extração de Dados

**Transformação e Limpeza de Dados:**
Após a extração, os dados precisam de ser transformados e limpos para garantir a sua qualidade e adequação ao formato do data warehouse. O processo representado na Figura 19 envolve a aplicação de regras de transformação, a remoção de duplicações e a correção de erros, assegurando que apenas dados devidamente transformados e limpos sejam carregados.

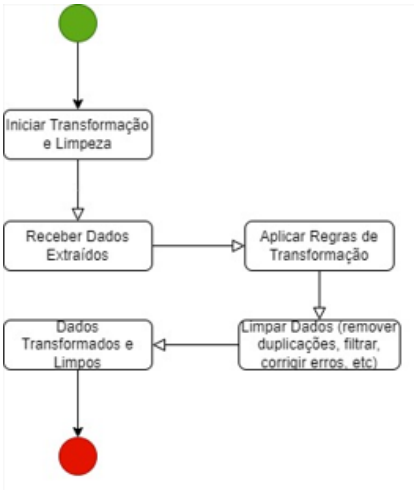



Figura 19 - Processo de Extração e Limpeza de Dados

**Carregamento no Data Warehouse:**

O carregamento dos dados no data warehouse é uma etapa crítica onde os dados transformados são inseridos ou atualizados nas tabelas dimensionais e de factos. O processo representado na Figura 20 envolve a preparação das consultas de inserção, a verificação da estrutura do DW e a garantia do sucesso nas validações dos dados. A inserção/atualização é realizada item a item, sendo a mesma concluída com sucesso ou insucesso, não comprometendo a continuação da inserção/atualização.

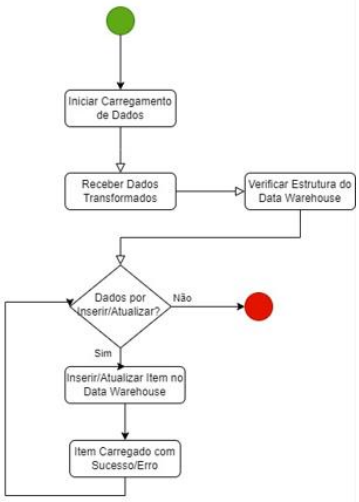



Figura 20 - Processo de Carregamento no Data Warehouse

Esses processos são fundamentais para garantir que os dados do EGITRON Quality Control sejam integrados de forma eficiente e eficaz, permitindo uma análise abrangente e estratégica para a otimização dos processos de produção e a tomada de decisões informadas.

## 4.6 Especificação do Sistema

De acordo com a Figura 6, é possível dividir a implementação em 3 partes: os *endpoints* responsáveis pela disponibilização dos dados, a ferramenta de ETL e o data warehouse.

Começando pelos *endpoints*, foram desenvolvidos como já enunciado na secção 2.4, na análise da solução, alguns recursos REST, separados por responsabilidades especificas do software, para através desses mesmos *endpoints* obter os dados estruturados no formato JSON e após, se necessário, algumas transformações inserir esses mesmos dados no armazém de dados. Os *endpoints* foram desenvolvidos na linguagem JAVA, tal como o próprio software EQC é. Com isto, conseguiu-se ter disponível através de alguns *endpoints*, toda a informação possível existente no *software*.

De seguida passando pela ferramenta ETL em questão, foi escolhida o Apache Nifi. O Apache NiFi foi selecionado devido à sua robustez e flexibilidade na integração de diferentes fontes de dados, bem como na capacidade de realizar transformações e limpezas nos dados conforme necessário. Com o Apache NiFi, foram estabelecidas pipelines de dados eficientes, permitindo a extração dos dados dos *endpoints* Java em formato JSON, seguida de transformações conforme especificado nos requisitos do projeto. Estas transformações podem incluir filtragem de dados, normalização de formatos, enriquecimento com informações adicionais, entre outras operações. O Apache NiFi facilita a configuração destas transformações de forma visual e intuitiva, permitindo uma rápida implementação e manutenção dos pipelines de dados.

Por fim, o data warehouse, que é o destino final dos dados processados pelo Apache NiFi. A base de dados do DW foi estabelecida localmente com a ajuda do software SQL Server Management Studio, que é utilizado para configurar, gerir e administrar o sistema.

Em resumo, a especificação do sistema compreende a implementação de *endpoints* em Java para disponibilização dos dados, a utilização do Apache NiFi como ferramenta de ETL para extração, transformação e carga dos dados, e a utilização de um data warehouse como destino final dos dados processados. Este conjunto de tecnologias forma uma arquitetura robusta e escalável para atender às necessidades de integração e análise de dados do projeto.

### 4.6.1 Endpoints

Com a realização destes *endpoints*, o objetivo era disponibilizar todo o conteúdo existente no software EQC. Devido a isso foram criados alguns *endpoints* cada um contendo diferentes informações. Abordando mais a fundo os mesmos, estes são alguns dos *endpoints* criados:

- ✓ **metrics/knowledge:** *Endpoint* que enviando uma data de início e uma de fim retorna várias listas de ID's de componentes criados ou atualizados entre essas mesmas datas.
- ✓ **metrics/product:** *Endpoint* que enviando uma lista de ID's de produtos retorna uma lista de produtos com vários dados referentes aos mesmos.
- ✓ **metrics/reporttemplate:** *Endpoint* que enviando uma lista de ID's de modelos de controlo retorna uma lista de modelos de controlo com vários dados referentes aos mesmos.
- ✓ **metrics/report:** *Endpoint* que enviando uma lista de ID's de relatórios retorna uma lista de relatórios com vários dados referentes aos mesmos.
- ✓ **metrics/assay:** *Endpoint* que enviando uma lista de ID's de relatórios ou de ensaios retorna vários dados e métricas referentes aos ensaios presentes no relatório.

Na Figura 21 é possível ver através do *Swagger* (uma ferramenta e uma estrutura de software usada para desenvolver, documentar e consumir APIs) alguns destes endpoints.

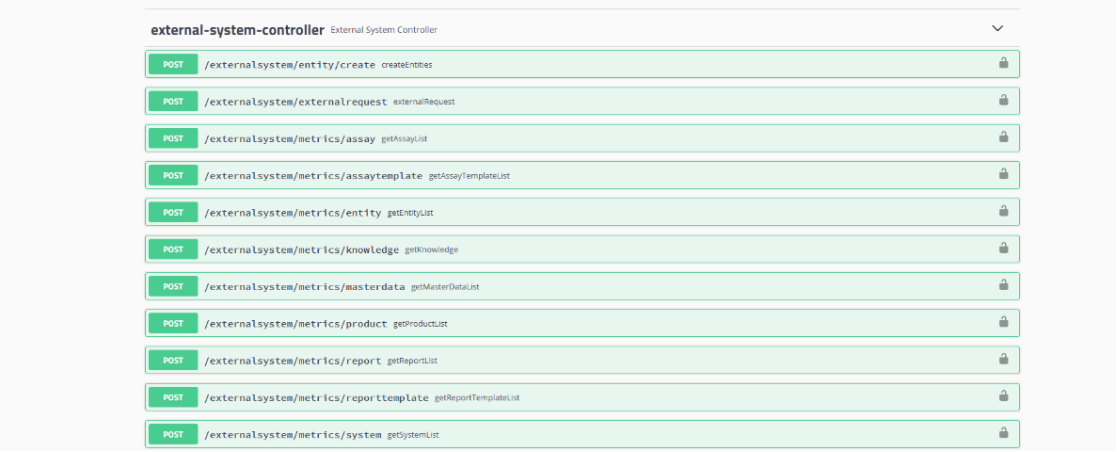



Figura 21 - Swagger UI

### 4.6.2 Apache Nifi

Através do Apache Nifi, foi possível sobretudo obter então os dados disponibilizados pelos *endpoints*, fazer as alterações necessárias aos JSONs recebidos, e carregar esses mesmos dados no Data Warehouse. Na Figura 22, é possível visualizar parte do esquema realizado para a implementação.

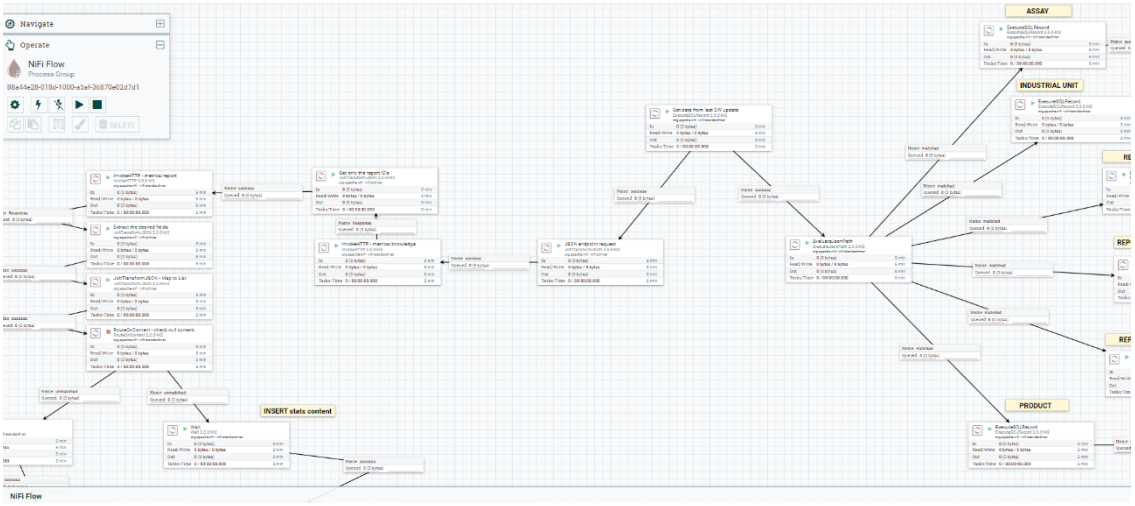



Figura 22 - Diagrama de fluxos do Apache Nifi

### 4.6.3 Data Warehouse

O data warehouse, como indicado anteriormente foi criado localmente com a ajuda do software SQL Server Management Studio. Esta ferramenta ofereceu uma plataforma robusta para a conceção e implementação do data warehouse, proporcionando uma variedade de funcionalidades que facilitaram a modelação dos dados, a criação de esquemas adequados e a gestão eficiente do armazenamento.

Na Figura 23, é apresentado o diagrama da base de dados do data warehouse. Este diagrama ilustra a estrutura e a organização das tabelas, as relações entre elas e as chaves primárias e estrangeiras que asseguram a integridade referencial dos dados. Através desta representação visual, é possível compreender melhor a arquitetura do sistema de armazenamento de dados, facilitando a análise e a tomada de decisões informadas.

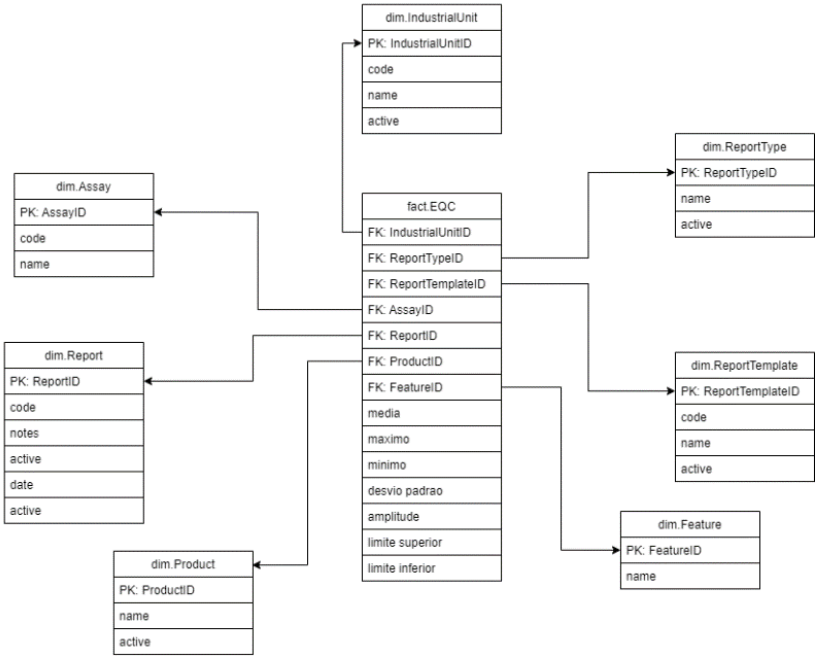



Figura 23 - Diagrama da base de dados

## 4.7 Implementação

Nesta secção, serão abordados os detalhes da implementação do data warehouse e do Apache NiFi. Inicialmente, a estruturação e implementação do esquema da base de dados do data warehouse serão descritas, seguidas pelo processo de instalação, configuração e utilização do Apache NiFi para o processamento e integração de dados. Esta secção detalha as etapas de compreensão dos requisitos, determinação da granularidade, identificação de dimensões, definição de medidas e criação das tabelas de factos e dimensões. Em seguida, a integração e o fluxo de dados utilizando o Apache NiFi são explorados, destacando os processadores utilizados e o procedimento passo a passo para a implementação do sistema.

### 4.7.1 Data Warehouse

Antes de abordar todo o processo realizado no Apache Nifi é necessário abordar a estruturação e implementação do esquema da base de dados do Data Warehouse. Estes são alguns dos passos realizados durante o processo de implementação:

**Compreensão dos Requisitos:**
Antes de iniciar o processo de modelagem, é necessário escolher os processos relevantes para análise na área de assunto a ser modelada. Isso envolve identificar as atividades-chave que impactam o desempenho do software e que serão incorporadas ao modelo dimensional.

**Determinação da Granularidade das Tabelas de Factos:**
O passo seguinte é determinar a granularidade das tabelas de factos. Isso significa decidir o nível de detalhe necessário para as métricas que serão armazenadas nas tabelas de factos, levando em consideração as necessidades de análise e os processos de negócios selecionados. No caso, como o software EQC gira muito em volta dos imensos relatórios que são realizados e das métricas/estatísticas dos próprios, a granularidade da tabela de factos é por relatório. O tamanho da base de dados com granularidade por semana seria muito menor do que por exemplo, como no caso, por relatório. Mas se por um lado se obtêm uma base de dados muito grande por outro pode significar que os dados não são detalhados o suficiente para que os utilizadores realizem consultas significativas à base de dados.

**Identificação de Dimensões e Hierarquias:**
Cada tabela de factos deve estar associada a dimensões relevantes que fornecerão contextos para as métricas. É importante identificar as dimensões específicas e suas hierarquias para cada tabela de factos, permitindo uma análise mais detalhada e abrangente dos dados. Seguindo a ideia do que foi abordado nas secções 2.2 e 2.3, é relativamente fácil identificar as dimensões existentes que irão fazer parte do esquema final da base de dados. Unidades Industriais, Relatórios, Modelos de Controlo, Ensaios (Dimensional, Humidade, Massa, etc), entre outros são algumas das dimensões necessárias.

**Definição de Medidas:**
Na tabela de factos, é necessário identificar as medidas que serão armazenadas e analisadas. Essas medidas representam as métricas quantitativas que serão avaliadas em relação às dimensões, oferecendo insights sobre o desempenho do *software* e dos processos de negócios.

**Determinação de Atributos Para as Tabelas de Dimensão:**
Cada tabela de dimensão deve conter atributos relevantes que descrevam as características das dimensões identificadas. Esses atributos fornecem contextos adicionais para as análises e são essenciais para a compreensão dos dados pelos utilizadores.

Como abordado na secção 3.2.2.1.2, o modelo dimensional é composto por dois tipos de tabelas: tabelas de factos e tabelas de dimensões. Para o caso em questão, foi necessária só a criação de uma tabela de factos. Conectadas a essa tabela estão as tabelas de dimensão que no caso são: Unidades Industriais, Modelos de Controlo, Relatórios, Produtos, entre outros. Todas essas tabelas estão representadas na tabela de factos através da respetiva FK. As medidas utilizadas na tabela são sobretudo medidas estatísticas comumente obtidas e analisadas pelos utilizadores do *software*, são elas: média, máximo, mínimo, desvio padrão, amplitude, limite superior e limite inferior, entre outras. Estas medidas, recaem sobretudo sobre os valores obtidos pelos ensaios realizados aos produtos e emitidos em relatório através do *software*. Com esta estrutura de base de dados, consigo ter acesso a vários dados estatísticos que cruzam dimensões como: Unidades Industriais, Modelos de Controlo, Relatórios, Tipos de Relatório, Ensaios, Características por Ensaios, Produtos e possivelmente

mais alguns surgiram.

Basicamente devido à estrutura que a base de dados apresenta e de as estatísticas já serem também previamente calculadas e não ser necessário estar constantemente a fazer os mesmos cálculos, é possível muito rapidamente ter acesso em milésimos de segundo a dados estatísticos (média, desvio padrão, máximos, mínimos, etc) que cruzam várias dimensões possíveis como por exemplo:

- ✓ Média da Amorim Distribuição no *Ensaio Dimensional* nos produtos de A a F na característica ovalidade.
- ✓ Desvio Padrão na Amorim Irmãos e Amorim Distribuição no ensaio de humidade em todas as características nos relatórios do tipo amostra no produto rolha x nos modelos de controlo A a D.

Tudo isto também por intervalo de datas se necessário para qualquer dimensão, ou seja, são N os novos dados e estatísticas que se pode aceder com esta nova implementação.

### 4.7.2 Apache Nifi

Após a disponibilização dos dados estar devidamente finalizada, através da realização dos *endpoints* enunciados na secção 4.6.1, e a estrutura da base de dados realizada, explicada na secção 4.6.3, é necessário receber esses mesmos dados e trabalhá-los devidamente para colocá-los no data warehouse.

#### 4.7.2.1 Instalação

O Apache NiFi é uma plataforma de processamento e distribuição de dados em tempo real, projetada para automatizar o movimento de dados entre diferentes sistemas. Nesta secção, além de se abordar o processo de instalação e configuração do NiFi, irão ser exploradas as suas capacidades e como pode ser utilizado para atender às necessidades de processamento de dados.

Antes de começar, é necessário certificar-se de que o sistema atende aos seguintes requisitos mínimos:

- ✓ Sistema operativo compatível com o Apache NiFi (Linux, Windows, MacOS, etc.).
- ✓ Java Development Kit (JDK) instalado (versão compatível com o Apache NiFi).
- ✓ Espaço suficiente em disco para armazenar os dados processados e os próprios arquivos do NiFi.

O primeiro passo é fazer o download do Apache NiFi a partir do site oficial (https://nifi.apache.org/download.html) e escolher a versão mais recente e compatível com o sistema operativo. Após o download, são extraídos os arquivos do Apache NiFi para uma pasta à escolha no sistema.

Agora, é necessário configurar algumas variáveis de ambiente necessárias para o funcionamento do Apache NiFi. Adiciona-se o diretório binário do NiFi ao PATH e define-se a variável JAVA_HOME para apontar para a instalação do JDK.

Para iniciar o Apache NiFi, é necessário aceder ao diretório binário do NiFi e executar o script nifi.sh (ou nifi.bat em sistemas Windows). Isso iniciará o servidor do NiFi dando acesso à interface do utilizador através do navegador usando o endereço: *https://localhost:8080/nifi*.

#### 4.7.2.2 Capacidades do Nifi

O Apache NiFi oferece uma ampla gama de funcionalidades para processar e movimentar dados. Algumas das capacidades principais são:

**Fluxo de Dados Visual:**
O NiFi fornece uma interface gráfica intuitiva para projetar e visualizar o fluxo de dados. Os utilizadores podem arrastar e soltar componentes para criar pipelines de dados complexos com facilidade.

**Processamento de Dados em Tempo Real:**
O NiFi suporta o processamento de dados em tempo real, permitindo o processamento imediato de dados à medida que são recebidos. Isso é essencial para cenários onde a baixa latência é crítica.

**Conectividade Extensível:**
Com uma ampla variedade de processadores pré-construídos e suporte para integração com outros sistemas através de APIs e protocolos padrão, o NiFi pode-se integrar facilmente a sistemas de armazenamento, bases de dados, sistemas de mensagens, e muito mais.

**Monitoramento e Gestão Centralizados:**
O NiFi oferece recursos robustos de monitoramento e gestão, permitindo que os administradores acompanhem o desempenho do fluxo de dados, identifiquem gargalos e realizem ajustes conforme necessário.

#### 4.7.2.3 Processadores Utilizados

No contexto da manipulação e processamento de dados no Apache NiFi, uma variedade de processadores desempenha um papel fundamental na transformação, integração e monitoramento dos fluxos de dados. Cada processador tem a sua própria função específica e contribui para a eficiência e flexibilidade do ambiente de processamento de dados. Estes foram os processadores utilizados:

**JoltTransformation:**
O processador JoltTransformation no Apache NiFi é usado para transformar dados JSON de um formato para outro. Aplica transformações baseadas em especificações Jolt (JSON Object Level Transformation), que são definidas usando uma linguagem de especificação JSON para descrever como os dados de entrada devem ser transformados para os dados de saída desejados. É possível fazer transformações como renomear campos, extrair valores, adicionar ou remover campos, entre outras operações, tudo isso sem a necessidade de escrever código

personalizado. Em suma, o JoltTransformation é uma ferramenta poderosa para manipular e remodelar dados JSON de maneira eficiente e flexível dentro do fluxo de dados do Apache NiFi.

**ExecuteSQL**:
O ExecuteSQL é um processador no Apache NiFi que permite executar consultas SQL em bases de dados. É útil para interagir com bases de dados, como MySQL, PostgreSQL, Oracle, SQL Server, entre outros. Com o ExecuteSQL, é possível enviar consultas SQL diretamente para uma base de dados e receber os resultados dentro do fluxo de dados do NiFi. Isso permite integrar facilmente dados de bases de dados relacionais com os respetivos fluxos de dados, realizar transformações baseadas em consultas SQL e até mesmo escrever dados de volta na base de dados, se necessário. Em resumo, o ExecuteSQL facilita a interação entre o Apache NiFi e as bases de dados, proporcionando uma maneira eficiente de trabalhar com dados armazenados nesses sistemas.

**InvokeHttp**:
O processador InvokeHTTP no Apache NiFi é utilizado para efetuar pedidos HTTP a serviços na web. Pode enviar pedidos GET, POST, PUT, DELETE e outros para URLs específicas. Isto permite que o NiFi interaja com sistemas externos, como APIs da web, para enviar ou receber dados. Em suma, o InvokeHTTP é uma porta de comunicação entre o NiFi e serviços *web* externos, permitindo a troca de informações.

**EvaluateJsonPath:**
O processador EvaluateJsonPath no Apache NiFi é utilizado para avaliar caminhos em documentos JSON. Pode ser usado para extrair dados específicos de um documento JSON, identificando e selecionando os valores associados a determinados caminhos ou chaves. Isso permite que o NiFi processe e manipule dados JSON de forma eficiente, selecionando apenas as informações relevantes para posterior processamento ou envio. Em resumo, o EvaluateJsonPath é uma ferramenta útil para trabalhar com dados estruturados em formato JSON no NiFi.

**DetectDuplicate:**
O processador DetectDuplicate no Apache NiFi é utilizado para detetar duplicados em fluxos de dados. Analisa os dados recebidos e verifica se existem registos ou mensagens idênticas que já foram processadas anteriormente. Isso é útil para evitar o processamento redundante de dados, especialmente em situações em que é importante evitar duplicações, como processamento de registos de eventos ou eliminação de mensagens repetidas. Em suma, o DetectDuplicate é uma ferramenta que ajuda a garantir a integridade e eficiência do fluxo de dados ao identificar e lidar com duplicações.

**MonitorActivity**:
O processador MonitorActivity no Apache NiFi é usado para monitorizar a atividade dentro do fluxo de dados. Ele recolhe estatísticas sobre os fluxos de dados que passam por ele, como o número de registos processados, o tamanho médio dos dados, entre outros. Isso permite aos utilizadores monitorizar e analisar o desempenho do fluxo de dados, identificar possíveis

gargalos ou problemas e otimizar o processo conforme necessário. Em resumo, o MonitorActivity é uma ferramenta útil para obter insights sobre o fluxo de dados e garantir um desempenho eficiente do sistema.

**Wait & Notify:**
O processador Wait pausa o fluxo de dados que passa por ele por um determinado período ou até que uma condição específica seja satisfeita. Por exemplo, pode ser configurado para aguardar uma determinada quantidade de tempo antes de permitir que os dados continuem a fluir pelo fluxo. O processador Notify é utilizado para notificar outros componentes do fluxo de dados quando ocorre um evento específico. Por exemplo, após uma condição ser satisfeita em outro ponto do fluxo, este processador pode ser configurado para enviar uma notificação para que o processamento continue em outro ramo do fluxo.

Relacionando os dois, o processador Wait pára o fluxo até que uma condição seja atendida, enquanto o processador Notify pode ser usado para notificá-lo quando essa condição é satisfeita, permitindo assim a continuação do processamento.

### 4.7.3 Procedimento

Após a instalação, surge a página de login do Apache Nifi. A configuração padrão gera um nome de utilizador e palavra-passe aleatórios, os quais são registados no ficheiro logs/nifi-app.log, localizado na pasta de instalação do NiFi. Ao preencher todos os campos necessários, é apresentada a página inicial, disponibilizando de imediato todos os recursos e ferramentas disponíveis. Com tudo configurado, é altura de estruturar e planear o diagrama e o fluxo de dados a seguir. Levando em consideração o que foi discutido na secção 4.6.1 sobre os *endpoints* criados e toda a teoria abordada em relação ao modelo dimensional de uma base de dados, estabelece-se um plano a seguir.

O plano resumido é ilustrado na Figura 24 e consiste inicialmente na criação e preenchimento das tabelas dimensionais, pois sem elas a tabela de factos não pode ser criada, uma vez que ela contém as chaves estrangeiras dessas tabelas dimensionais. Em seguida, é necessário invocar alguns dos *endpoints* criados para obter toda a informação necessária e, se necessário, efetuar algumas transformações antes de inserir os dados na tabela de factos. Para facilitar este processo, conta-se com uma tabela temporária que serve para a migração e verificação de alguns dados, e também com uma tabela de histórico no data warehouse, onde será registada uma nova entrada a cada atualização, permitindo que apenas os dados a partir dessa data até hoje sejam requisitados na próxima atualização.

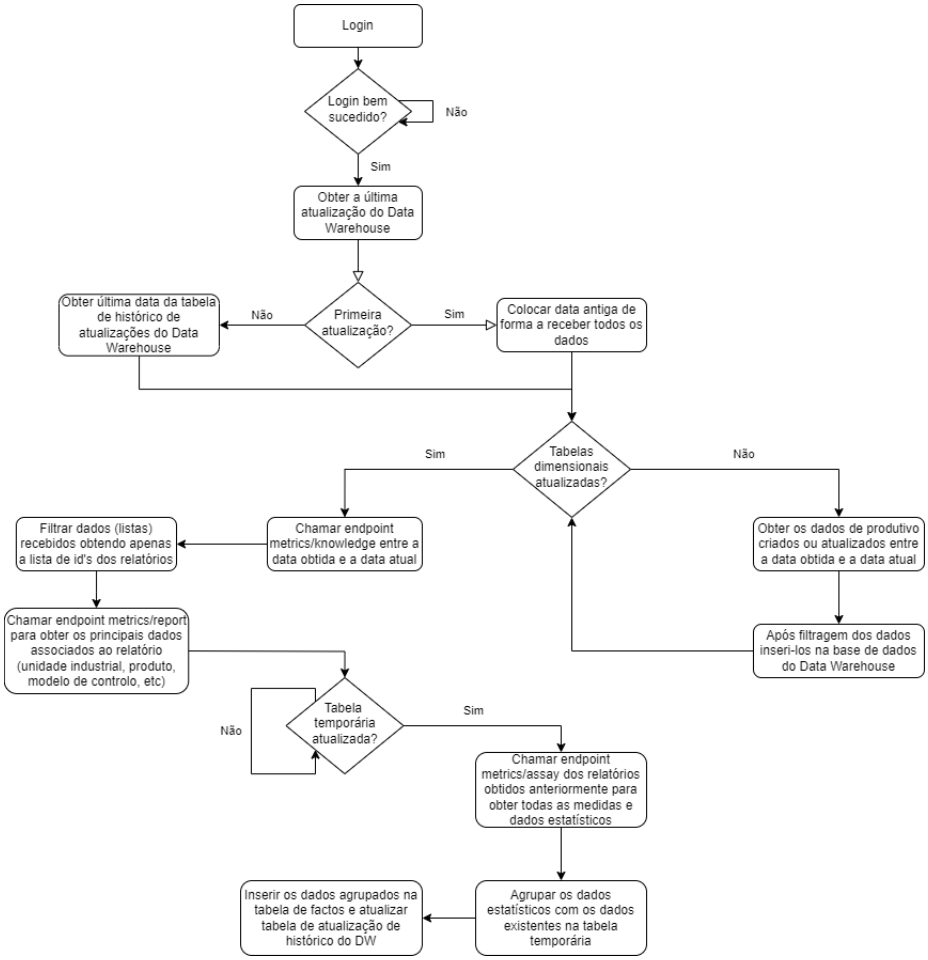



Figura 24 - Diagrama Resumo do Fluxo Implementado

Na implementação, é essencial conectar o Apache Nifi à base de dados de produção e ao data warehouse. Para isso, utiliza-se o processador DBCPConnectionPool, que estabelece e gerência *pools* de conexão com bases de dados usando a biblioteca Apache Commons DBCP (Database Connection Pooling). Conforme ilustrado na Figura 25, é necessário preencher diversos campos, como o URL de conexão à base de dados, o nome da classe do *driver* da base de dados, as credenciais de utilizador da base de dados e, possivelmente, uma *query* de validação.

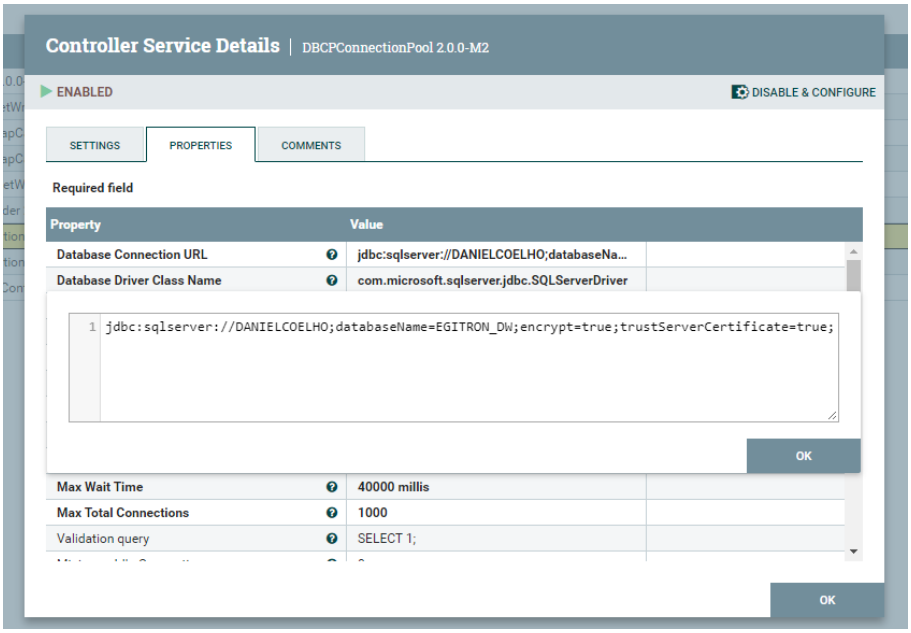



Figura 25 - Conexão a Base De Dados

Com o Apache Nifi devidamente conectado às duas bases de dados, é iniciado o diagrama. A primeira etapa consiste em obter a data da última atualização realizada no data warehouse. Para isso, utiliza-se o processador ExecuteSQL, ao qual é atribuída a conexão à base de dados de produção previamente configurada, e uma consulta SQL que obtém a data mais recente. Caso não haja registos, é devolvida uma data que garanta a inclusão de todos os dados da base de dados de produção.

Segue-se o diagrama com dois fluxos distintos, ilustrado na Figura 26. Por um lado, o caminho para a criação da tabela de factos e, por outro, o processo relacionado às tabelas dimensionais.

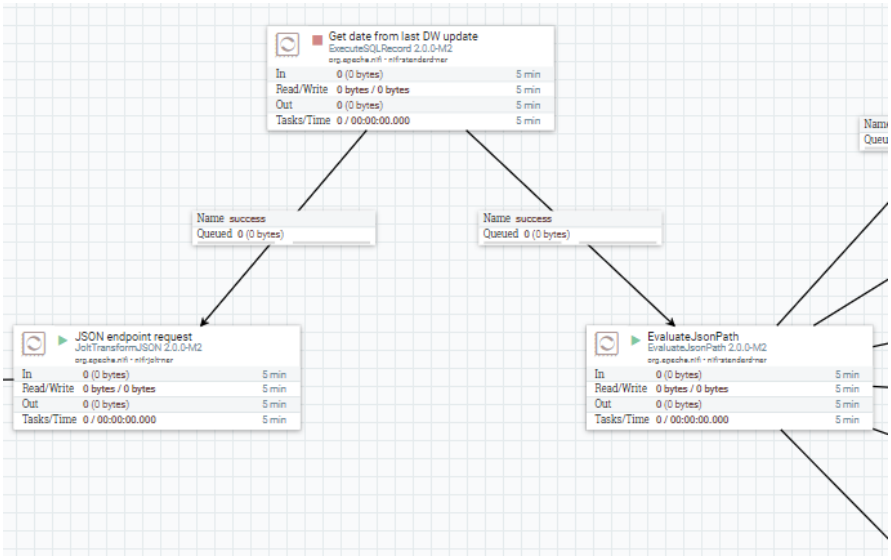



Figura 26 - Após obtenção da última atualização

#### 4.7.3.1 Tabelas Dimensionais

No fluxo das tabelas dimensionais, opta-se por migrar os dados diretamente da base de dados de produção para o data warehouse. Para exemplificar, considere-se a tabela das unidades industriais. Como é possível ver na Figura 27, recorre-se novamente ao processador ExecuteSQL, onde uma consulta SQL é configurada para obter todas as unidades industriais criadas ou atualizadas entre a última atualização do data warehouse e a data atual.

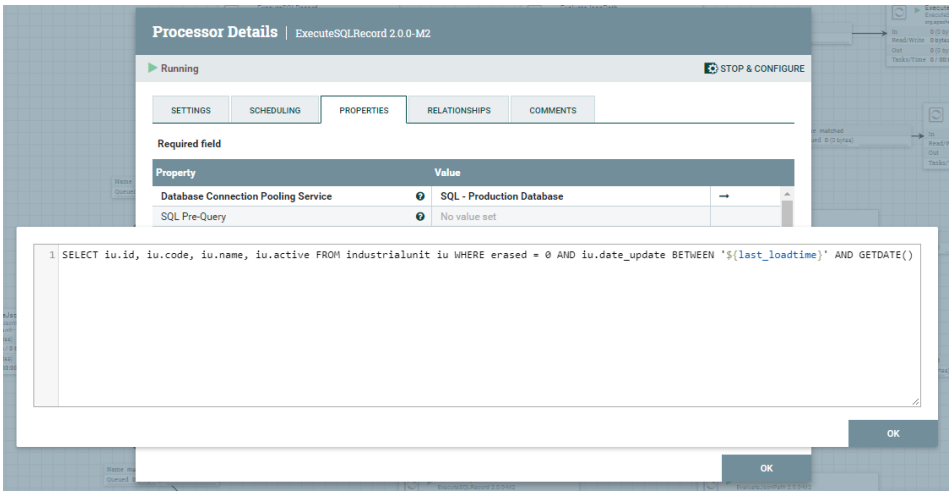



Figura 27 - Consulta SQL para obter unidades industriais

Obtendo todos os campos necessários, recorre-se novamente ao processador ExecuteSQL (Figura 28) para, por meio de uma *query*, criar uma unidade industrial se esta ainda não existir, ou atualizar uma unidade industrial caso já exista na tabela dimensional.

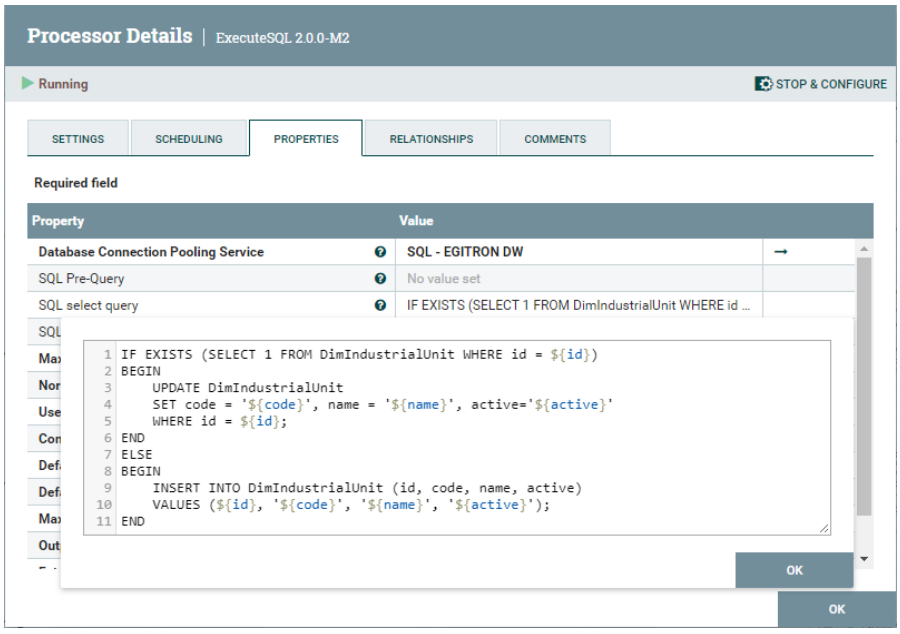



Figura 28 - Inserção ou atualização de uma unidade industrial

Repetindo o mesmo procedimento para todas as outras tabelas dimensionais, finaliza-se a criação ou atualização das mesmas. É importante observar que, devido à estrutura intencionalmente concebida dessa maneira, nunca será possível inserir qualquer dado na tabela de factos sem que todas as tabelas dimensionais tenham sido corretamente criadas ou atualizadas.

#### 4.7.3.2 Tabela de Factos

No fluxo das tabelas de factos, inicia-se o processo de recolha de dados de alguns dos *endpoints* previamente criados e mencionados na secção 4.6.1. Após a construção do JSON necessário para efetuar o pedido ao *endpoint*, utilizando o processador JoltTransformationJSON e incluindo as datas inicial e final pertinentes, procede-se à chamada ao *endpoint* metrics/knowledge. Conforme explicado na secção 4.6.1, este *endpoint* irá devolver os IDs de tudo o que foi criado ou atualizado entre as datas especificadas. Após receber a resposta, extrai-se apenas a lista dos IDs dos relatórios, uma vez que é em torno dos dados e estatísticas destes que o Data Warehouse se concentra.

Com isso, avança-se para a obtenção dos principais dados sobre os relatórios. Para isso, é realizada uma chamada ao *endpoint* metrics/report, de onde são retornados, como é possível ver na Figura 29, todos os tipos de dados relativos ao respetivo relatório, exceto os valores e estatísticas dos testes presentes no mesmo.

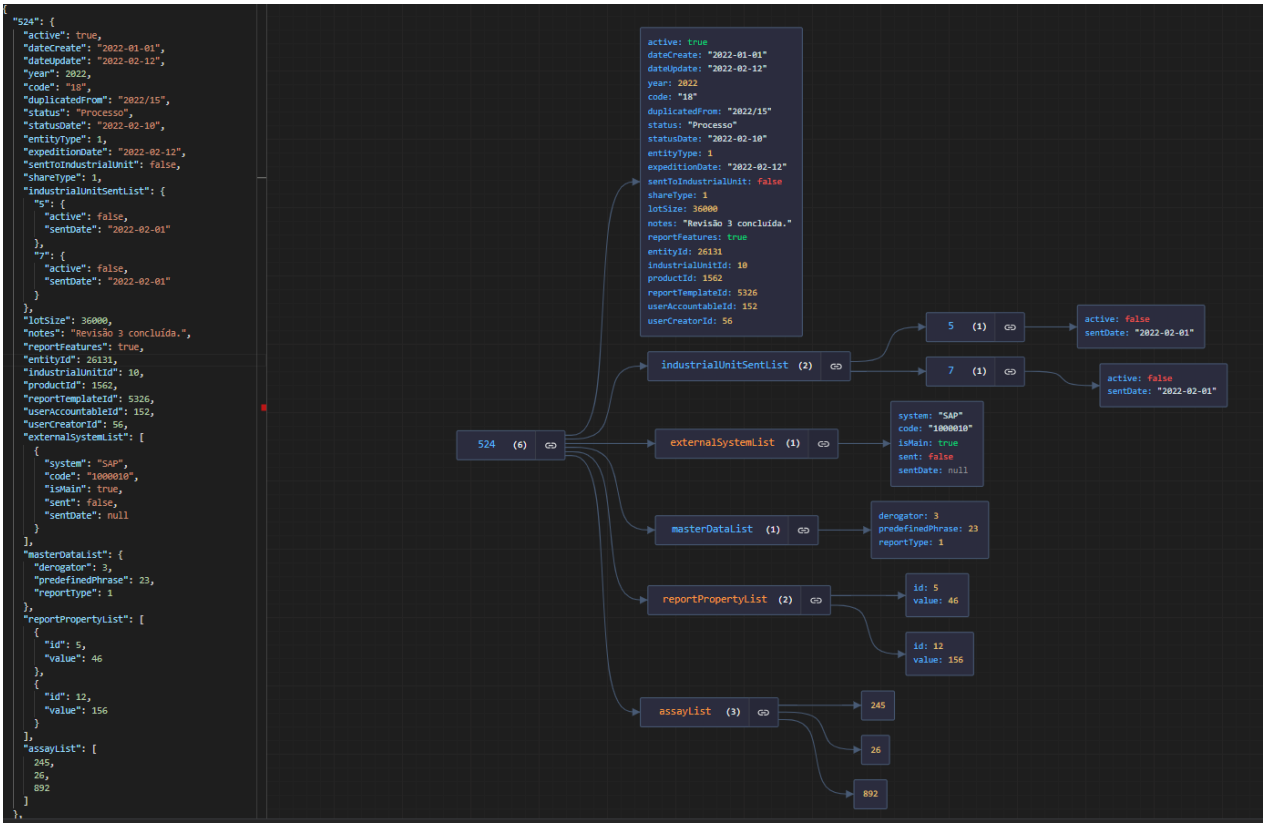



Figura 29 - JSON obtido do endpoint metrics/report

Utilizando o processador JoltTransformationJSON, são selecionados apenas os campos desejados do JSON de resposta, tendo em conta que são fornecidas todas as informações do relatório e que, no futuro, mais campos poderão ser extraídos. Conforme apresentado na Figura 30, são extraídas informações como o código e ano do relatório, a unidade industrial que o criou, o tipo de relatório, o produto incluído no relatório, o modelo de controlo associado e até mesmo o estado do relatório. É importante notar que a maioria destes dados anteriormente mencionados correspondem às chaves estrangeiras existentes na tabela de factos.

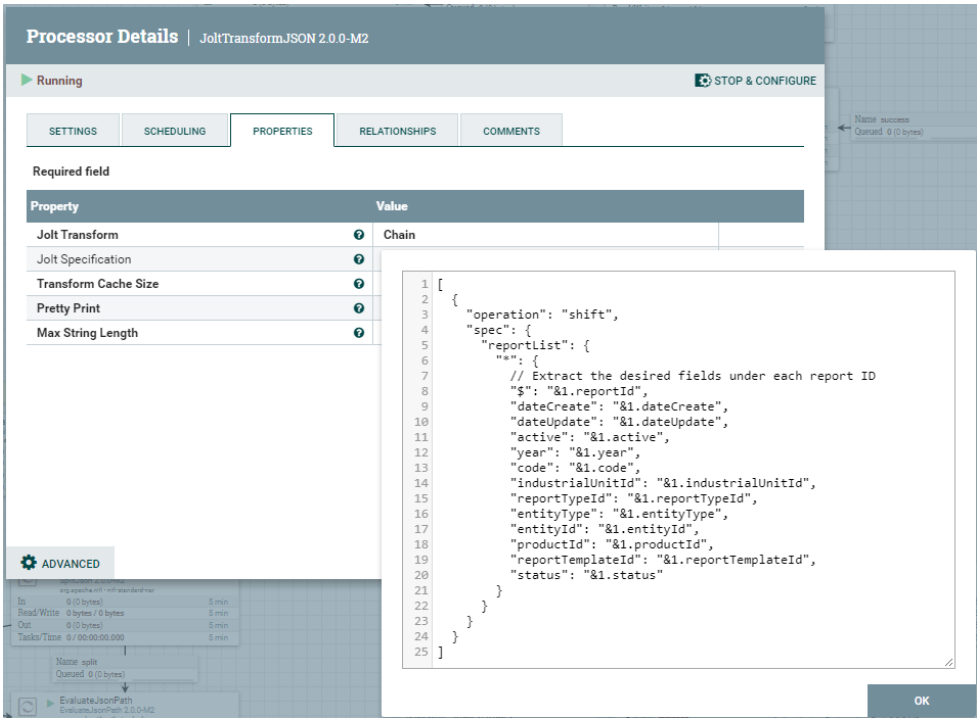



Figura 30 - Filtragem de Dados do Relatório

Neste ponto, foi decidido primeiramente inserir as chaves estrangeiras (FK's) na tabela de factos temporária, que será útil para a transição, modificação e confirmação de dados. Em seguida, é chamado o *endpoint* metrics/assay, onde se obtém os valores e estatísticas referentes aos testes realizados em cada ensaio presente em cada relatório (média, desvio padrão, amplitude, entre outros). Após esta obtenção e o agrupamento com os dados já existentes na tabela temporária, a tabela de factos é preenchida completamente.

Assim, explicando passo a passo o que foi enunciado anteriormente, é possível visualizar através da Figura 31 que o diagrama segue novamente por dois caminhos diferentes.

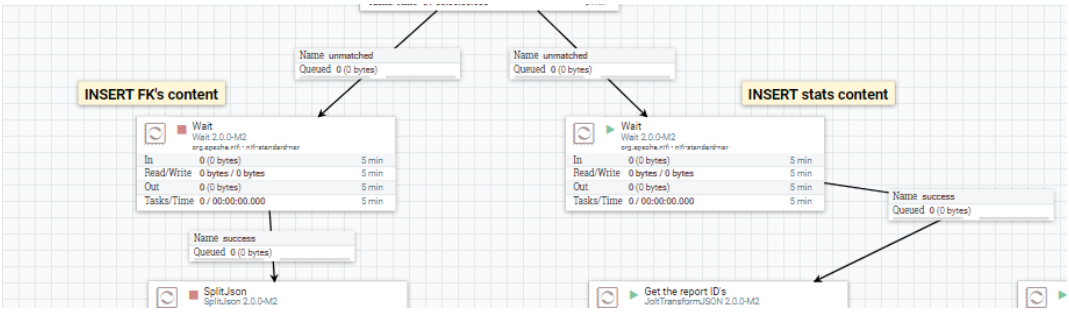



Figura 31 - Dois Caminhos a Seguir Para Preencher Estatísticas

**Inserção das FK's**

Seguindo a decisão de inserir primeiramente as chaves estrangeiras (FKs) na tabela temporária, obtidas através do *endpoint* metrics/report, e estando num ponto onde se tem a lista de todos os relatórios com toda a informação referente aos mesmos, é utilizado o processador SplitJson para dividir o JSON por cada posição ou, no caso, por cada relatório. Em seguida, as variáveis são configuradas corretamente no processador ExecuteSQL e inseridas na tabela temporária o respetivo conteúdo de cada relatório. Na Figura 32 é possível ver em maior detalhe o sucedido e na Figura 33 a tabela temporária devidamente preenchida.

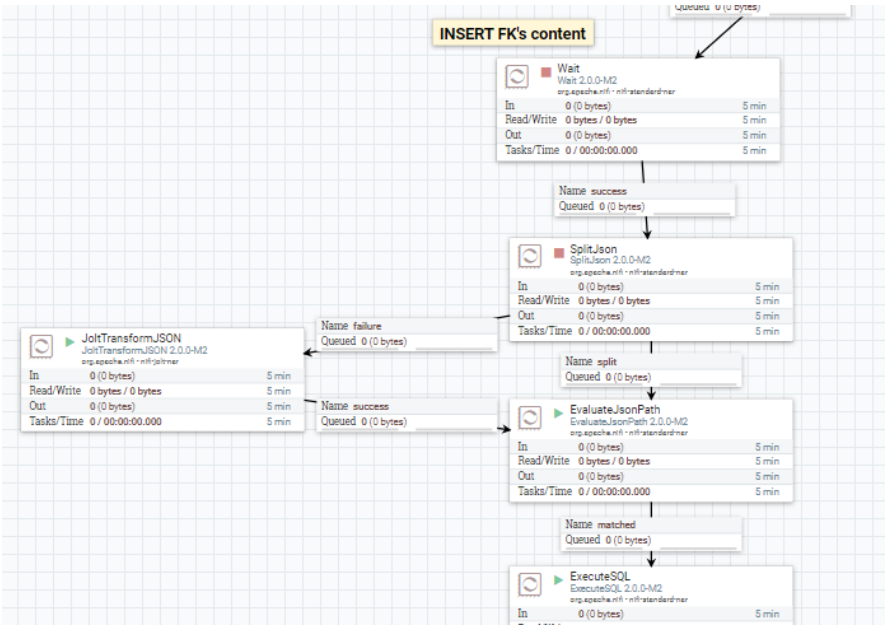
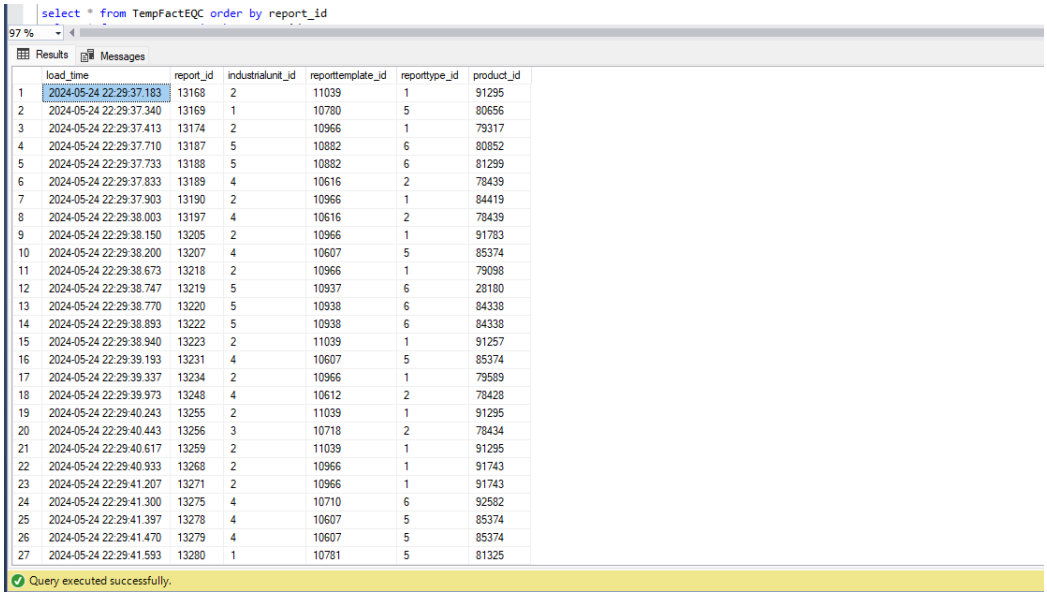



Figura 32 - Implementação para preencher tabela temporária



```
select * from TempFactEQC order by report_id
```

97 %

Results | Messages

| | load_time | report_id | industrialunit_id | reporttemplate_id | reporttype_id | product_id |
|---|---|---|---|---|---|---|
| 1 | 2024-05-24 22:29:37.183 | 13168 | 2 | 11039 | 1 | 91295 |
| 2 | 2024-05-24 22:29:37.340 | 13169 | 1 | 10780 | 5 | 80656 |
| 3 | 2024-05-24 22:29:37.413 | 13174 | 2 | 10966 | 1 | 79317 |
| 4 | 2024-05-24 22:29:37.710 | 13187 | 5 | 10882 | 6 | 80852 |
| 5 | 2024-05-24 22:29:37.733 | 13188 | 5 | 10882 | 6 | 81299 |
| 6 | 2024-05-24 22:29:37.833 | 13189 | 4 | 10616 | 2 | 78439 |
| 7 | 2024-05-24 22:29:37.903 | 13190 | 2 | 10966 | 1 | 84419 |
| 8 | 2024-05-24 22:29:38.003 | 13197 | 4 | 10616 | 2 | 78439 |
| 9 | 2024-05-24 22:29:38.150 | 13205 | 2 | 10966 | 1 | 91783 |
| 10 | 2024-05-24 22:29:38.200 | 13207 | 4 | 10607 | 5 | 85374 |
| 11 | 2024-05-24 22:29:38.673 | 13218 | 2 | 10966 | 1 | 79098 |
| 12 | 2024-05-24 22:29:38.747 | 13219 | 5 | 10937 | 6 | 28180 |
| 13 | 2024-05-24 22:29:38.770 | 13220 | 5 | 10938 | 6 | 84338 |
| 14 | 2024-05-24 22:29:38.893 | 13222 | 5 | 10938 | 6 | 84338 |
| 15 | 2024-05-24 22:29:38.940 | 13223 | 2 | 11039 | 1 | 91257 |
| 16 | 2024-05-24 22:29:39.193 | 13231 | 4 | 10607 | 5 | 85374 |
| 17 | 2024-05-24 22:29:39.337 | 13234 | 2 | 10966 | 1 | 79589 |
| 18 | 2024-05-24 22:29:39.973 | 13248 | 4 | 10612 | 2 | 78428 |
| 19 | 2024-05-24 22:29:40.243 | 13255 | 2 | 11039 | 1 | 91295 |
| 20 | 2024-05-24 22:29:40.443 | 13256 | 3 | 10718 | 2 | 78434 |
| 21 | 2024-05-24 22:29:40.617 | 13259 | 2 | 11039 | 1 | 91295 |
| 22 | 2024-05-24 22:29:40.933 | 13268 | 2 | 10966 | 1 | 91743 |
| 23 | 2024-05-24 22:29:41.207 | 13271 | 2 | 10966 | 1 | 91743 |
| 24 | 2024-05-24 22:29:41.300 | 13275 | 4 | 10710 | 6 | 92582 |
| 25 | 2024-05-24 22:29:41.397 | 13278 | 4 | 10607 | 5 | 85374 |
| 26 | 2024-05-24 22:29:41.470 | 13279 | 4 | 10607 | 5 | 85374 |
| 27 | 2024-05-24 22:29:41.593 | 13280 | 1 | 10781 | 5 | 81325 |

Query executed successfully.

Figura 33 - Tabela temporária preenchida

**Inserção das medidas**

Após a inserção do conteúdo principal de um relatório na tabela temporária, é altura de inserir os dados e estatísticas que farão parte da tabela de factos. Neste ponto, foi utilizado o *endpoint* metrics/assay mencionado anteriormente na secção 4.6.1. Enviando a lista de ID's dos relatórios no pedido, são retornados os dados e estatísticas relativos aos relatórios. Após algumas transformações e filtragens de conteúdo utilizando os processadores EvaluateJSONPath e JoltTransformation, chega-se a um ponto onde apenas temos uma lista das medidas/estatísticas de cada relatório. Novamente, é utilizado o processador SplitJson para dividir o JSON por cada posição ou, no caso, por cada relatório. Com isso, é procedido simplesmente à inserção desses dados na tabela de factos.

No entanto, é realizada uma *query* específica onde primeiro é verificado se o relatório existe na tabela temporária e se não existe na tabela de factos. Se sim, é feito o agrupamento dos dados da tabela temporária com os novos e inseridos na tabela de factos. Se não, é feita uma inserção direta na tabela de factos, indicando que apenas ocorreu uma edição nos valores do relatório. Na Figura 34 é possível ver em maior detalhe o sucedido e na Figura 35 a tabela de factos devidamente preenchida.

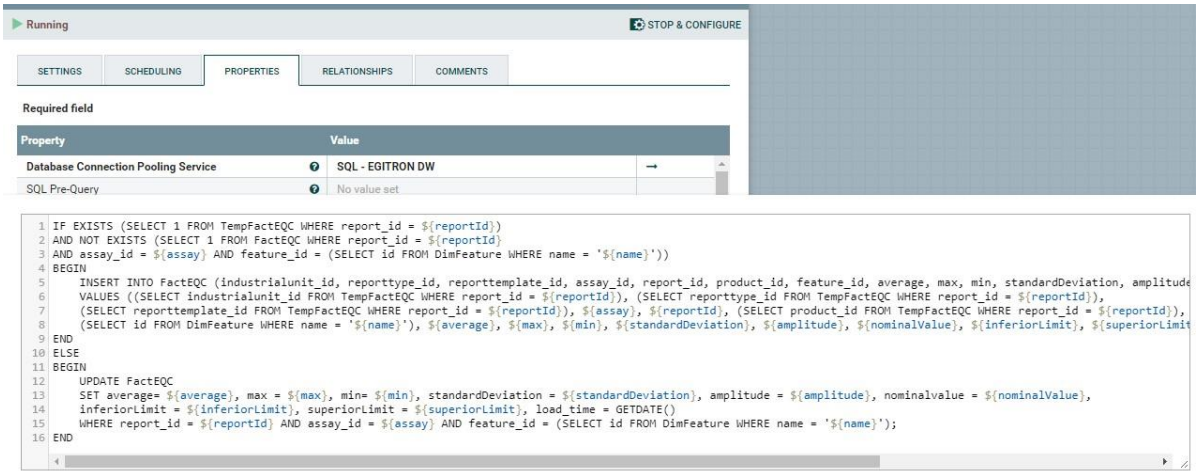



Figura 34 - Preenchimento da Tabela De Factos

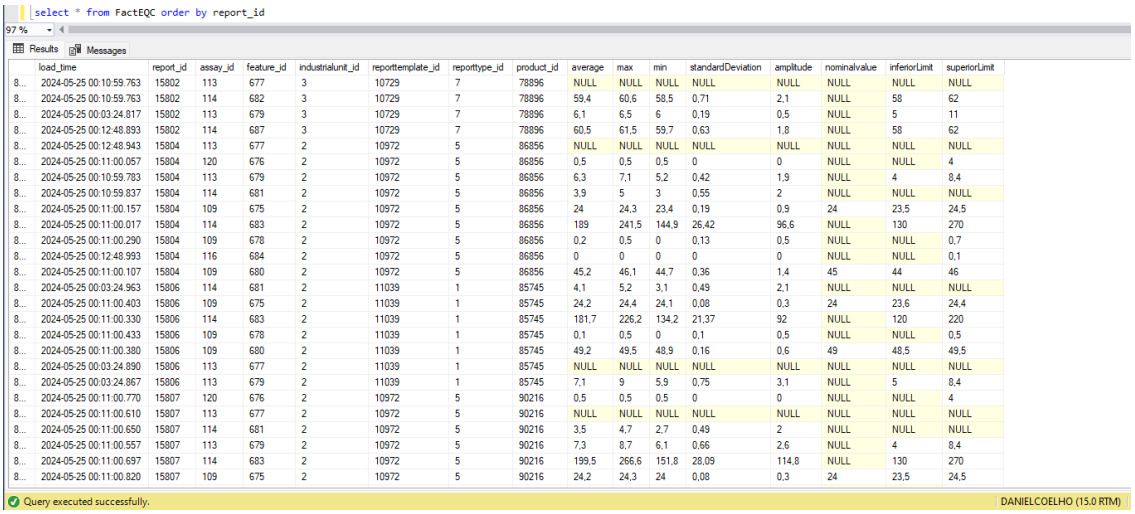



Figura 35 - Tabela de Factos Preenchida

## 4.8 Implementação e Colocação em Produção do Apache Nifi

A transição de soluções de um ambiente de desenvolvimento local para um ambiente de produção é uma etapa crítica na garantia da fiabilidade, disponibilidade e acessibilidade de aplicações para os utilizadores finais. Esta secção detalha o processo de implementação do Apache NiFi, uma ferramenta robusta para a automação e gestão de fluxos de dados, desde a escolha da infraestrutura de produção até a monitorização contínua.

A secção inicia-se com a análise das diversas opções de infraestrutura para hospedar o Apache NiFi em produção, incluindo servidores físicos, servidores virtuais em serviços de nuvem como AWS, Azure ou Google Cloud Platform, e containers utilizando Docker e Kubernetes. A escolha da infraestrutura impacta diretamente no desempenho e na escalabilidade do sistema.

Em seguida, são descritos os passos necessários para preparar o ambiente de produção, como a configuração do sistema operativo e a instalação das dependências necessárias. Um guia passo a passo para a instalação e configuração do Apache NiFi no servidor de produção é apresentado, cobrindo desde o download e extração do *software* até as configurações de segurança, incluindo a habilitação de HTTPS e a configuração de autenticação e autorização.

### 4.8.1 Escolha da Infraestrutura de Produção

Para hospedar o Apache NiFi em produção, diversas opções de infraestrutura podem ser consideradas, tais como:

**Servidores Físicos:** Adequados para ambientes onde é necessário controlo total sobre o hardware.

**Servidores Virtuais:** Hospedagem em serviços de nuvem como AWS, Azure ou Google Cloud Platform.

**Containers:** Utilização do Docker para empacotar o NiFi e Kubernetes para orquestração.

### 4.8.2 Preparação do Ambiente de Produção

Os passos necessários para preparar o servidor de produção incluem:

**Configuração do Sistema Operativo:** Instalação de um sistema operativo adequado, como Ubuntu Server ou CentOS.

**Instalação de Dependências:** Garantir que todas as dependências necessárias para o Apache NiFi estejam instaladas (por exemplo, Java).

### 4.8.3 Instalação do Apache Nifi no Servidor

Segue-se um guia passo a passo para a instalação e configuração do Apache NiFi no servidor de produção:

**Download e Extração:**

```
wget https://archive.apache.org/dist/nifi/1.15.3/nifi-1.15.3-bin.tar.gz
tar -xzvf nifi-1.15.3-bin.tar.gz
```

**Configuração Inicial:**

- ✓ Editar os ficheiros de configuração, como nifi.properties, para ajustar portas, caminhos de armazenamento, etc.
- ✓ Configurar a memória e outros parâmetros de desempenho conforme necessário.

**Segurança:**

- ✓ Habilitar HTTPS configurando certificados SSL.
- ✓ Configurar autenticação e autorização para garantir que apenas utilizadores autorizados possam aceder e modificar os fluxos.

### 4.8.4 Deploy dos Fluxos de Dados

Para migrar os fluxos de dados configurados no ambiente local para o ambiente de produção, os seguintes passos devem ser seguidos:

- ✓ **Exportar e Importar Templates:** Utilizar *templates* para exportar os fluxos do ambiente local e importar no servidor de produção.
- ✓ **Configurar Conexões:** Garantir que todas as conexões (bases de dados, APIs, etc.) estejam configuradas corretamente no ambiente de produção.

### 4.8.5 Monitorização e Manutenção

A monitorização e manutenção contínua são essenciais para assegurar o funcionamento adequado do Apache NiFi em produção:

- ✓ **Logs e Alertas:** Configurar alertas para erros e monitorizar continuamente os logs para identificar e solucionar problemas rapidamente.

### 4.8.6 Discussão / Análise

A implementação e colocação em produção do Apache NiFi envolve uma série de passos críticos, desde a escolha da infraestrutura até à realização de testes de produção. Este processo garante que a solução é fiável, escalável e segura, proporcionando benefícios significativos em termos de acessibilidade e desempenho para os utilizadores finais.

## 4.9 Sumário

O capítulo 4 foi focado no desenvolvimento do sistema proposto, desde a especificação inicial até à implementação. Foi iniciado pela especificação detalhada dos componentes do sistema, como endpoints, Apache Nifi e data warehouse. A implementação foi descrita em etapas, incluindo a configuração do data warehouse e a instalação do Apache Nifi, seguida por um procedimento detalhado.

# 5 Avaliação da Solução

A implementação de um data warehouse (DW) representa um marco significativo na gestão e análise de grandes volumes de dados, permitindo uma otimização substancial na recuperação e manipulação de informações. No entanto, a etapa final e crucial deste processo é a validação da integridade e precisão dos dados migrados. Esta validação é essencial para garantir que o DW oferece uma representação fiel e confiável dos dados originais armazenados na base de dados de produção.

## 5.1 Objetivos da Avaliação

Conforme delineado na secção 1.4, os objetivos específicos deste trabalho incluem:

1. Desenho e implementação de um protótipo de DW para empresas que utilizam o software EQC.
2. Desenvolvimento de um processo de ETL que assegure a correta integração dos dados, garantindo a sua qualidade e consistência.
3. Criação de uma arquitetura de DW robusta e escalável, adaptável às necessidades específicas de cada empresa.
4. Avaliação do desempenho da solução implementada através de testes e validações comparativas.
5. Promoção de uma utilização estratégica dos dados gerados pelo EQC para obter *insights* valiosos.

Cada uma dessas metas será avaliada através de metodologias específicas que garantem uma validação abrangente e precisa da solução implementada.

## 5.2 Metodologia de Avaliação

A avaliação da solução foi conduzida em várias frentes, seguindo uma metodologia estruturada que inclui:

1. **Validação de Dados:**
- ✓ **Comparação de Dados entre o DW e a Base de Dados de Produção:** Verificação do número total de registos e precisão dos dados.
- ✓ **Validação de Contagem de Registos:** Comparação das tabelas dimensionais e de fatos.
- ✓ **Validação de Campos de Dados:** Comparação dos valores de campos específicos.
- ✓ **Verificação de Integridade Referencial:** Garantia de que todas as chaves estrangeiras no DW correspondem a chaves primárias válidas.

2. **Avaliação de Desempenho:**
- ✓ **Complexidade das Consultas:** Comparação da complexidade e eficiência das consultas entre a base de dados produtiva e o DW.
- ✓ **Execução dos Processos de Validação:** Validação inicial, periódica e pós-mudança.
- ✓ **Análise de Desempenho:** Medição do tempo de execução das consultas para avaliar a eficiência do sistema.

3. **Validação de *Endpoints*:**
- ✓ **Testes Manuais dos *Endpoints*:** Execução de casos de teste para verificar a extração, transformação e carregamento corretos dos dados.

Cada uma dessas etapas foi documentada e os resultados foram analisados para assegurar que a solução atende aos objetivos estabelecidos.

## 5.3 Resultados da Avaliação

Esta secção aborda a comparação meticulosa entre os dados contidos no data warehouse (DW) e aqueles na base de dados de produção. A comparação envolve a criação de consultas específicas para verificar o número total de registos em tabelas correspondentes e para analisar a precisão dos dados. Além disso, a secção destaca as diferenças na complexidade das consultas entre as duas bases de dados, evidenciando a eficiência do DW na execução de consultas mais simples e rápidas. A análise de desempenho das consultas, medida através de testes comparativos, ilustra os benefícios tangíveis em termos de eficiência e simplicidade.

### 5.3.1 Comparação de Dados Entre o Data Warehouse e Base de dados de Produção

A fase final e fundamental na implementação de um data warehouse é a validação da integridade e precisão dos dados migrados. Essa validação é crucial para assegurar que o data warehouse ofereça uma representação fiel e confiável dos dados do sistema de produção.

Para realizar essa comparação, foram adotadas abordagens meticulosas que envolveram uma análise comparativa entre os dados armazenados no data warehouse e os dados originais mantidos na base de dados de produção. Para alcançar este objetivo, foram elaboradas consultas específicas com o intuito não apenas de comparar o número total de registos entre as tabelas das duas bases de dados, mas também de analisar os dados para garantir uma maior precisão e veracidade dos mesmos.

#### 5.3.1.1 Validação de Contagem de Registos

Começando pelas tabelas dimensionais, que são as primeiras a serem populadas, foram criadas três *queries*. Uma delas corresponde ao número total de registos na tabela específica da base de dados do data warehouse, outra ao número total de registos na tabela correspondente da base de dados de produção (origem), e por fim uma que agrupa as duas tabelas das diferentes bases de dados por uma determinada característica e obtém o número total.

Nas imagens seguintes, da Figura 36 a 38, é possível visualizar a comparação abordando as tabelas que contêm, por exemplo, os relatórios, produtos e modelos de controlo, verificando assim a veracidade e qualidade do processo de migração de dados.

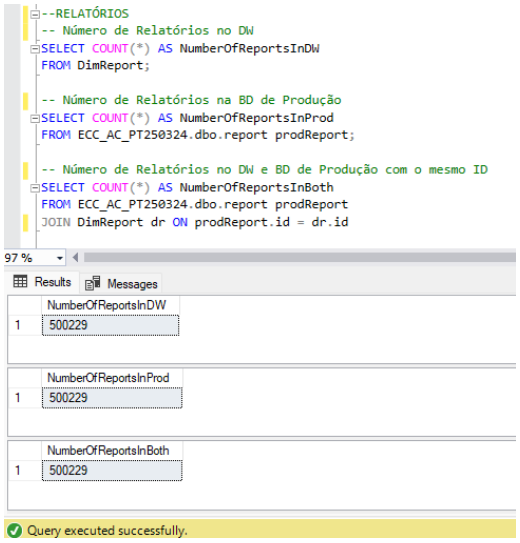



Figura 36 - Comparação de Relatórios

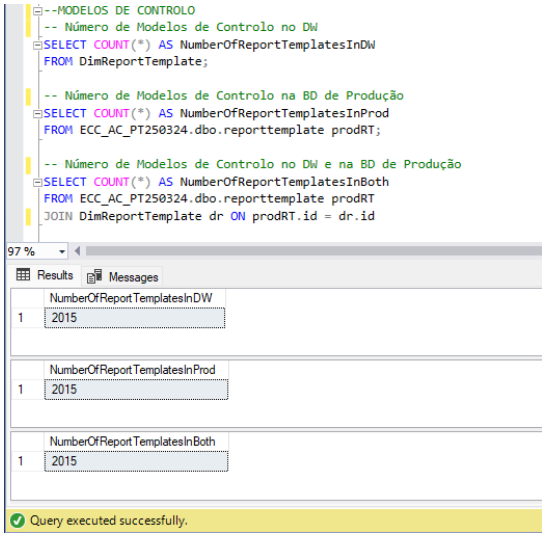



Figura 37 - Comparação de Modelos de Controlo

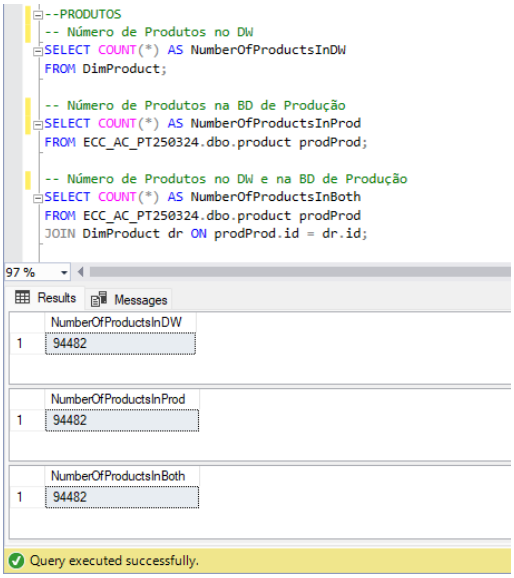



Figura 38 - Comparação de Produtos

#### 5.3.1.2 Validação de Campos de Dados

De seguida é necessário comparar os valores de campos específicos entre a tabela de produção e a tabela do data warehouse para garantir que não haja discrepâncias. De realçar que o número total de linhas de cada *query* executada é igual ao número total de linhas da tabela específica no DW ou produção (se a última *query* da Figura 39 retornar 10 linhas indica-me que apenas 10 unidades industriais estão presentes nas duas BD's com o código igual, se o número total de unidades industriais na BD de produtivo for 10, é sinal de que todas as unidades industriais foram passadas e todas elas apresentam dados corretos, no caso o código). Na Figura 39 é possível verificar o sucedido.

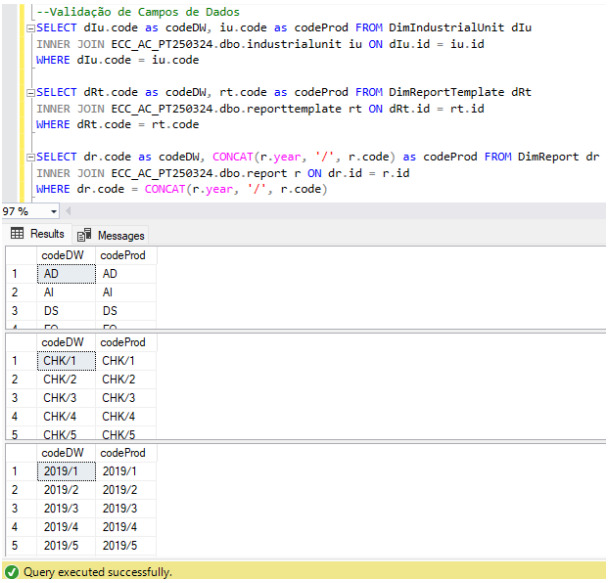



Figura 39 - Validação de Campos de Dados

Por outro lado, foi também realizada uma *query* onde se verifica se as FK's existentes na tabela de factos estão de facto corretamente bem agrupadas. Na Figura 40 é possível verificar que o número de itens retornado é igual ao número total de linhas da tabela de facto, ou seja, confirma-se a validade do conteúdo presente na tabela.

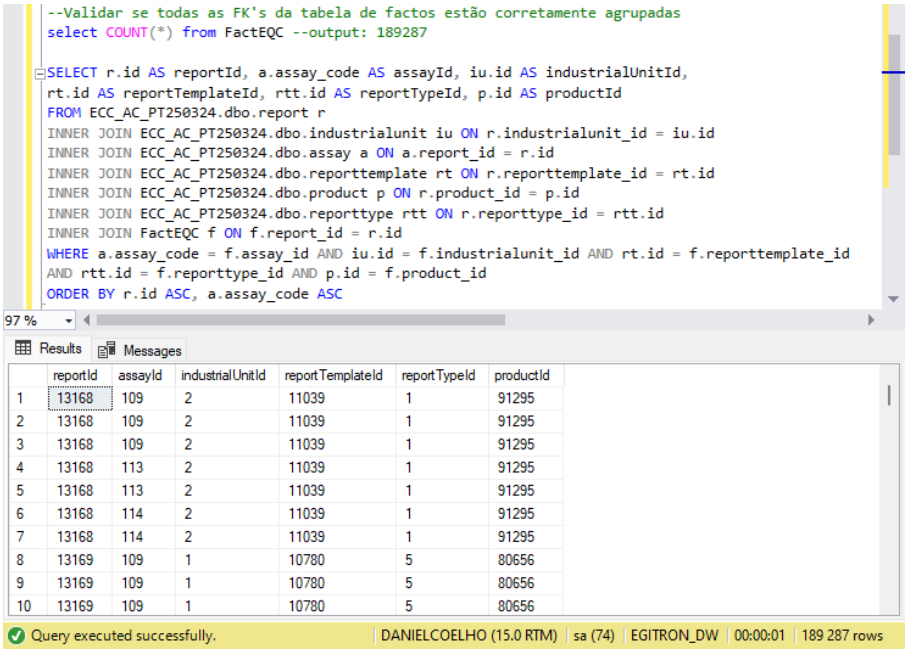



Figura 40 - Validação de Tabela de Facto

#### 5.3.1.3 Verificação de Integridade Referencial

É necessário também assegurar que as relações entre tabelas no data warehouse respeitam a integridade referencial definida no modelo de dados. No caso, serão verificadas se todas as

chaves estrangeiras no data warehouse correspondem a chaves primárias válidas nas tabelas relacionadas. Na Figura 41 é possível visualizar que não existem chaves estrangeiras inválidas.

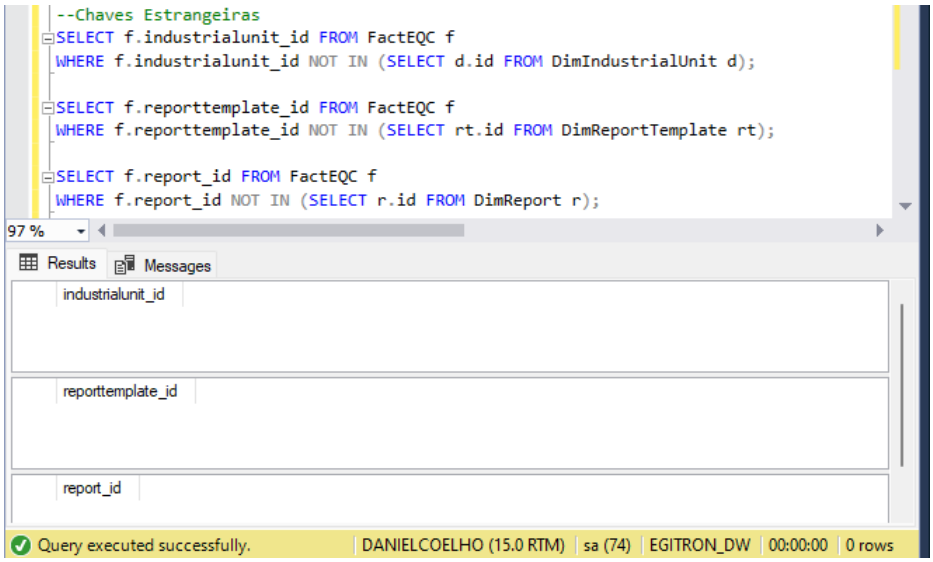



Figura 41 - Validação de Integridade Referencial

### 5.3.2 Complexidade das Consultas

Uma das principais diferenças entre uma base de dados produtiva e um data warehouse é a complexidade das consultas necessárias para obter informações. Nas bases de dados produtivas, as consultas tendem a ser mais complexas devido à necessidade de realizar múltiplas junções entre tabelas e aplicar diversas condições para filtrar os dados.

Considere-se a seguinte *query* realizada numa base de dados produtiva:

```
SELECT CONCAT(r.year,'/',r.code) FROM report r
INNER JOIN industrialunit iu ON r.industrialunit_id = iu.id
INNER JOIN reporttemplate rt ON r.reporttemplate_id = rt.id
INNER JOIN reporttype rtt ON r.reporttype_id = rtt.id
INNER JOIN product p ON r.product_id = r.product_id
INNER JOIN language_product lp ON lp.product_id = p.id
INNER JOIN assay a ON a.report_id = r.id
INNER JOIN assaytest at ON at.assay_id = a.id
INNER JOIN assay_csdimensional_values dv ON dv.assaytest_id = at.id
INNER JOIN assayvalue_diameter av ON av.id = dv.diameter_id
WHERE lp.name = 'RT NAT 49X24 2 (W 4S) CL0 E' AND a.assay_code = 109 AND
r.active = 1
GROUP BY CONCAT(r.year,'/',r.code)
```

Esta *query* requer várias junções e condições para obter o resultado desejado, refletindo a complexidade e a carga computacional necessária para extrair a informação. No data warehouse, os dados são organizados de maneira a otimizar o desempenho das consultas. Estruturas como tabelas de factos e dimensões permitem que as informações sejam pré-calculadas e armazenadas de forma mais eficiente.

A mesma informação pode ser obtida com uma *query* significativamente mais simples no data warehouse:

```
SELECT r.code FROM FactEQC f
INNER JOIN DimReport r ON r.id = f.report_id
INNER JOIN DimProduct p ON p.id = f.product_id
INNER JOIN DimAssay a ON a.id = f.assay_id
WHERE p.name = 'RT NAT 49X24 2 (W 4S) CL0 E' AND a.id = 109
GROUP BY r.code
```

Nesta *query*, o número de junções e condições é consideravelmente reduzido, o que demonstra a eficiência na recuperação dos dados. As tabelas de dimensão e factos permitem uma organização dos dados que simplifica as consultas e melhora o desempenho.

### 5.3.3 Execução dos Processos de Validação

Os processos de validação foram executados em diferentes condições, incluindo:

- ✓ **Validação Inicial:** Durante a carga inicial dos dados.
- ✓ **Validação Periódica:** Em intervalos regulares para garantir a continuidade da integridade dos dados.
- ✓ **Validação Pós-Mudança:** Após qualquer alteração significativa no processo ETL ou na estrutura dos dados.

**Total de Processos Executados:**

- ✓ **Validação Inicial:** 10 processos
- ✓ **Validação Periódica:** 20 processos (realizados semanalmente ao longo de 4 semanas)
- ✓ **Validação Pós-Mudança:** 10 processos

Os resultados dos processos de validação confirmam que os dados no data warehouse são verdadeiros e consistentes com os dados de produção. Nenhuma discrepância significativa foi encontrada, e todas as regras de negócio e transformações foram aplicadas corretamente.

### 5.3.4 Análise de Desempenho

A implementação final foi avaliada com base nos objetivos inicialmente estabelecidos e nos requisitos funcionais e não funcionais definidos. Esta avaliação considera tanto a capacidade do sistema para cumprir as suas funcionalidades quanto a sua performance geral.

Para quantificar a performance do data warehouse em comparação com a base de dados produtiva, foram realizadas cinco tentativas de execução de cada uma das *queries* apresentadas na secção 5.3.2, medindo-se o tempo necessário para obter os resultados. Os tempos obtidos (em segundos) são apresentados no gráfico da Figura 42:

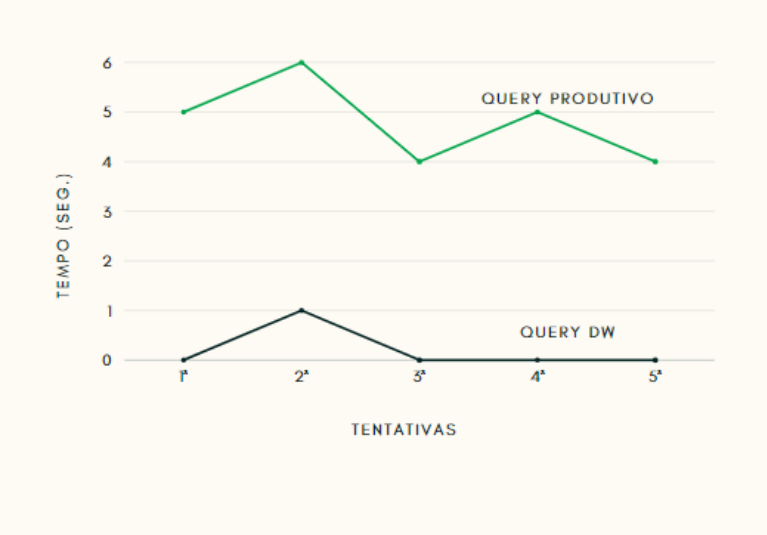



Figura 42 - Performance - DW vs. Produtivo

A comparação entre as duas *queries* ilustra vários benefícios do data warehouse:

**Eficiência**: Consultas mais rápidas devido à organização otimizada dos dados.

**Simplicidade**: *Queries* menos complexas, facilitando a manutenção e a compreensão do código.

**Desempenho**: Redução da carga computacional nas consultas, resultando em melhor desempenho geral do sistema.

**Pré-calculação:** Muitos dos cálculos e agregações são feitos antecipadamente, o que acelera a recuperação da informação.

Enumerando mais alguns dos requisitos funcionais e não funcionais propostos na secção 4.1, no que toca ao ponto 1, o sistema demonstrou uma capacidade robusta para extrair dados de diversas fontes heterogéneas, como bases de dados de produção e *endpoints* de API, graças à utilização do Apache NiFi. Os processadores do Apache NiFi facilitaram a integração de dados provenientes de múltiplas origens, assegurando uma extração de dados eficiente e sem falhas.

Já no ponto 2, a transformação e limpeza dos dados extraídos foram realizadas com sucesso, utilizando processadores do Apache Nifi personalizados para manipular dados JSON e garantir que estivessem no formato adequado antes de serem carregados no data warehouse. No que diz respeito ao carregamento dos dados no data warehouse (ponto 3), o sistema conseguiu realizar a inserção e atualização dos dados nas tabelas dimensionais e de factos de forma eficaz como foi possível visualizar na Figura 35 por exemplo.

A integridade referencial dos dados (ponto 6) foi mantida através de rigorosas verificações e validações implementadas nos processos de ETL e nas validações realizadas no durante e pós implementação como foi possível verificar na secção 5.3.1.3.

Além disso, a implementação de mecanismos para atualização incremental (ponto 4) permitiu a inserção de novos registos e a atualização de registos existentes de maneira eficiente, reduzindo significativamente o tempo de carga e evitando redundância de dados. A utilização de *querys* capazes de inserir ou atualizar conforme o necessário foi essencial para atingir o objetivo deste requisito (Figura 28 e 34).

No que toca ao ponto 5, o sistema mostrou uma alta capacidade de processamento de grandes volumes de dados sem comprometer a performance. Para atingir o objetivo do ponto 5, foram realizados testes de desempenho que indicaram que os resultados obtidos no data warehouse foram alcançados num tempo menor comparado ao da base de dados de produção, demonstrando eficiência superior (Figura 42). Como abordado no ponto 7, medidas de segurança foram rigorosamente implementadas para proteger dados sensíveis durante o movimento e armazenamento, incluindo a utilização de criptografia e conexões seguras (SSL/TLS). Estas medidas garantiram a proteção adequada dos dados, atendendo aos padrões de segurança exigidos.

A facilidade de manutenção do sistema (ponto 8) foi alcançada graças à interface gráfica intuitiva do NiFi, que facilitou a monitorização contínua e ajustes de configuração sem interrupções significativas. Esta característica simplificou a gestão do sistema e permitiu uma manutenção eficiente. A compatibilidade com diferentes sistemas operativos e ambientes (ponto 9) foi garantida pela utilização de componentes de software amplamente suportados e padrões abertos. O sistema demonstrou ser flexível e adaptável a diversas configurações de infraestrutura.

Por fim, foi fornecida uma documentação detalhada do sistema (ponto 10), abrangendo a arquitetura, processos de ETL, configuração do Apache NiFi e o esquema dimensional. Esta documentação facilitou o entendimento e a manutenção do sistema por parte dos utilizadores e administradores, atendendo plenamente a este requisito.

### 5.3.5 Validação de Endpoints

Na implementação de sistemas complexos de armazém de dados, a realização de testes é crucial para garantir a integridade e a confiabilidade dos dados. Embora a dissertação se tenha concentrado principalmente na implementação do ETL e na construção do armazém de dados, a validação dos *endpoints* através de testes automáticos é uma prática essencial que garante a correta extração, transformação e carregamento dos dados.

**Metodologia dos Testes:**
Apesar de não terem sido implementados testes automáticos completos durante o desenvolvimento deste projeto, foi seguido um rigoroso processo de validação manual dos *endpoints*. A metodologia envolveu os seguintes passos:

**Identificação dos Endpoints:**
Foram identificados todos os *endpoints* críticos utilizados para a extração de dados das fontes de produção e APIs.

**Definição de Casos de Teste:**
Para cada *endpoint*, foram definidos casos de teste que abrangem cenários típicos de uso, bem como condições extremas e de erro. Por exemplo:

- ✓ Consulta de dados válidos.
- ✓ Consulta com parâmetros inválidos.
- ✓ Verificação de respostas em casos de ausência de dados.
- ✓ Testes de performance para grandes volumes de dados.

**Execução de Testes Manuais:**
Cada caso de teste foi executado manualmente para verificar a resposta do *endpoint*. Isso incluiu:

- ✓ Verificação da estrutura da resposta (JSON/XML).
- ✓ Validação dos dados retornados contra os dados esperados.
- ✓ Teste de tempos de resposta para garantir a performance.

**Documentação dos Resultados:**
Todos os resultados dos testes foram documentados, incluindo as condições de teste, os dados de entrada, as respostas recebidas e as observações sobre qualquer discrepância encontrada.

**Correção de Erros:**
Em casos onde foram identificados problemas ou discrepâncias, os *endpoints* foram ajustados conforme necessário, e os testes foram re-executados para garantir que as correções estavam eficazes.

Embora a implementação de testes automáticos não tenha sido realizada nesta fase do projeto, os testes manuais extensivos garantiram a funcionalidade e a integridade dos *endpoints* críticos. Para futuras iterações do projeto, é recomendada a implementação de um conjunto de testes automáticos usando *frameworks* como Postman, JUnit, ou similares, para automatizar e ampliar a cobertura dos testes de *endpoints*, assegurando a qualidade contínua e facilitando a deteção e correção de erros de forma mais eficiente.

### 5.3.6 Discussão / Análise

A implementação final do sistema não apenas atendeu aos objetivos do projeto, mas também cumpriu todos os requisitos funcionais e não funcionais estabelecidos. A avaliação do desempenho confirmou que o sistema é eficiente, seguro, altamente disponível, fácil de manter, compatível com diferentes ambientes e bem documentado. Dessa forma, o sistema

está bem posicionado para oferecer suporte contínuo às necessidades analíticas e de negócios da organização.

### 5.3.7 Sumário

A implementação de um data warehouse (DW) e a subsequente validação da integridade e precisão dos dados migrados são etapas cruciais para garantir que o DW ofereça uma representação fiel e confiável dos dados originais. Esta secção relembrou os objetivos da avaliação e a respetiva metodologia de avaliação. Além disso, abordou a comparação detalhada entre os dados contidos no DW e aqueles na base de dados de produção, incluindo a criação de consultas específicas para verificar o número total de registos e a precisão dos dados.

Foram descritas várias etapas do processo de validação, como a comparação do número total de registos entre tabelas correspondentes, a verificação da precisão dos valores de campos específicos e a garantia de que todas as chaves estrangeiras correspondem a chaves primárias válidas. A complexidade das consultas também foi abordada, comparando a complexidade das consultas em bases de dados produtivas e no DW, demonstrando a eficiência e simplicidade das consultas no DW. Os processos de validação foram executados em diferentes condições, incluindo validação inicial, periódica e pós-mudança, assegurando a integridade contínua dos dados. A análise de desempenho comparou a performance entre a base de dados produtiva e o DW, mostrando benefícios em termos de eficiência, simplicidade e desempenho.

A validação de *endpoints* foi realizada através de uma metodologia de testes manuais, incluindo a definição de casos de teste, execução de testes e documentação dos resultados. A conclusão destacou que a implementação final do sistema atendeu aos objetivos do projeto e aos requisitos estabelecidos, confirmando a eficiência, segurança, facilidade de manutenção, compatibilidade e documentação detalhada do sistema.

Em suma, a secção demonstrou a importância de um processo rigoroso de validação para assegurar a integridade dos dados no DW, destacando a eficiência e simplicidade das consultas no DW em comparação com a base de dados de produção, e recomendando a implementação futura de testes automáticos para aprimorar a qualidade contínua do sistema.

# 6 Conclusão

Este capítulo apresenta as considerações finais sobre a dissertação e as futuras melhorias propostas. São discutidos os principais objetivos alcançados, as limitações encontradas e o potencial impacto do data warehouse desenvolvido, assim como as áreas de aprimoramento que podem ser exploradas para otimizar e expandir o sistema.

## 6.1 Considerações Finais

A presente dissertação teve como objetivo central o planeamento e desenvolvimento de uma solução de data warehouse (DW) para empresas que utilizam o software EGITRON Quality Control (EQC). Este projeto visou preencher a lacuna existente entre a geração massiva de dados pelo EQC e a subutilização dessas informações para melhorar a eficiência operacional, a qualidade e a inovação nas operações de produção.

O primeiro objetivo foi desenhar e implementar um protótipo de DW que integrasse os dados gerados pelo EQC, permitindo uma análise sistemática das operações de produção. Para tal, desenvolveu-se um processo de ETL (Extração, Transformação, Carregamento) que assegurou a correta integração dos dados, garantindo a sua qualidade e consistência. Criou-se uma arquitetura de DW robusta e escalável, adaptável às necessidades específicas de cada empresa. Ao longo da implementação, foram executados diversos testes e validações para comparar os dados entre o DW e as bases de dados de produção. Analisou-se a complexidade das consultas e a eficiência do sistema, demonstrando que a organização dos dados no DW permite consultas mais simples e rápidas, em comparação com as bases de dados de produção.

Os principais objetivos da dissertação foram atingidos. O protótipo de DW desenvolvido mostrou-se eficaz na integração e análise dos dados gerados pelo EQC. O processo de ETL implementado garantiu a qualidade e consistência dos dados, e a arquitetura do DW provou ser robusta e escalável. Os testes e validações realizados confirmaram a eficiência da solução, com consultas menos complexas e mais rápidas do que nas bases de dados de produção. Na fase final e fundamental da implementação, foi realizada uma validação meticulosa da integridade e precisão dos dados migrados. Esta comparação envolveu a criação de consultas específicas para verificar o número total de registos em tabelas correspondentes e para analisar a precisão dos dados. As tabelas dimensionais e de factos foram cuidadosamente comparadas, assegurando a veracidade e qualidade do processo de migração de dados. Além disso, foi verificada a integridade referencial, confirmando que todas as chaves estrangeiras no DW correspondiam a chaves primárias válidas nas tabelas relacionadas. A análise de desempenho das consultas, medida através de testes comparativos, ilustrou os benefícios tangíveis em termos de eficiência e simplicidade. A organização dos dados no DW resultou em consultas menos complexas e mais rápidas, reduzindo a carga computacional e melhorando o desempenho geral do sistema.

No entanto, o trabalho realizado também apresentou algumas limitações. Conforme especificado na secção 3.4.2 e 3.5 e considerando as tecnologias que acarretam custos adicionais, bem como as preferências de algumas empresas clientes da EGITRON que optam por manter os seus dados e bases de dados em servidores locais, foi desenvolvida uma solução utilizando tecnologias gratuitas e implementando o armazenamento dos dados localmente, sendo essas as duas principais limitações da implementação. Além disso, a implementação não incluiu testes automáticos completos para a validação dos *endpoints*, embora tenham sido realizados testes manuais extensivos. A introdução de testes automáticos em futuras iterações do projeto seria benéfica para aumentar a cobertura dos testes e facilitar a deteção e correção de erros de forma mais eficiente. Além disso, a análise de desempenho foi realizada com um conjunto limitado de *queries* e dados, sendo recomendada uma avaliação mais extensa e variada em futuras iterações. Subjetivamente, este projeto demonstrou o potencial significativo de um DW bem implementado para transformar a forma como as empresas industriais utilizam os dados gerados pelos seus processos de produção. A capacidade de integrar e analisar grandes volumes de dados de forma eficiente pode proporcionar *insights* valiosos, permitindo decisões mais informadas e estratégicas. Esta compreensão aprofundada das melhores práticas e tecnologias relacionadas com data warehouses poderá impulsionar a inovação e a competitividade no setor industrial.

## 6.2 Melhorias Futuras

Embora a implementação atual do data warehouse (DW) tenha sido bem-sucedida, foram identificadas diversas áreas que podem ser aprimoradas para otimizar e expandir o sistema. Esta secção delineia várias melhorias futuras que podem ser introduzidas para melhorar a funcionalidade, segurança e eficiência do DW. Aborda-se a necessidade de suporte para

múltiplas bases de dados de produção simultâneas, o desacoplamento das ligações diretas às bases de dados de produção, a automatização e monitorização avançada, e a otimização da performance do data warehouse. Essas melhorias visam não apenas incrementar a robustez e a eficiência do DW, mas também assegurar que o sistema esteja preparado para crescer e se adaptar às futuras necessidades das unidades industriais.

### 6.2.1 Suporte para Múltiplas Bases de Dados de Produção Simultâneas

Atualmente, o data warehouse (DW) está configurado para funcionar com uma única base de dados de produção. Uma melhoria significativa seria a capacidade de o DW integrar dados de múltiplas bases de dados de produção ao mesmo tempo. Para atingir este objetivo, seria necessário:

- ✓ **Escalabilidade do Sistema:** Garantir que o DW é escalável e pode lidar com o aumento do volume de dados e das conexões simultâneas.
- ✓ **Desenvolvimento de Conectores Dinâmicos:** Implementar conectores dinâmicos no Apache NiFi que possam ser facilmente configurados para diferentes bases de dados sem necessitar de reconfigurações manuais extensivas.

### 6.2.2 Desacoplamento das Ligações Diretas às Bases de Dados de Produção

No diagrama de fluxo atual do Apache NiFi, existem ligações diretas às bases de dados de produção. No entanto, para aumentar a segurança e a flexibilidade, seria benéfico eliminar essas ligações diretas e utilizar os *endpoints* criados como intermediários para a transferência de dados. As vantagens incluem:

- ✓ **Segurança Melhorada:** Redução do risco de exposição direta das bases de dados de produção a possíveis vulnerabilidades.
- ✓ **Manutenção Simplificada:** Facilitar a manutenção e atualização dos fluxos de dados sem a necessidade de modificações diretas nas conexões de base de dados.
- ✓ **Flexibilidade Operacional:** Permitir mudanças rápidas e eficientes na configuração dos fluxos de dados, adaptando-se melhor às necessidades dinâmicas das operações industriais.

### 6.2.3 Automatização e Monitorização Avançada

Para melhorar a eficiência operacional e a confiabilidade do sistema, a implementação de mecanismos avançados de automatização e monitorização é crucial:

- ✓ **Alertas e Notificações:** Configurar sistemas de alerta que notifiquem os administradores sobre falhas ou anomalias no fluxo de dados em tempo real.

- ✓ **Monitorização Contínua:** Implementar ferramentas de monitorização contínua que forneçam visibilidade sobre o desempenho e a integridade dos fluxos de dados, permitindo ações proativas para resolver problemas antes que afetem o sistema.

### 6.2.4 Otimização da Performance do Data Warehouse

A performance do DW é crucial para garantir que os dados estejam disponíveis em tempo hábil para análise. Algumas estratégias de otimização incluem:

- ✓ **Indexação e Particionamento:** Utilizar técnicas avançadas de indexação e particionamento das tabelas para melhorar a velocidade das consultas.
- ✓ **Armazenamento de Dados em Camadas:** Implementar uma abordagem de armazenamento em camadas, onde dados mais recentes e frequentemente acedidos são armazenados em camadas de acesso rápido, enquanto dados mais antigos são movidos para camadas de armazenamento mais lentas e económicas.

## 6.3 Sumário

O capítulo discute melhorias futuras para otimizar e expandir a implementação atual do data warehouse (DW). Propõe-se suporte para múltiplas bases de dados de produção simultâneas, desacoplamento das ligações diretas às bases de dados para melhorar segurança e flexibilidade, automatização avançada e monitorização contínua para eficiência operacional, além da otimização da performance do DW através de indexação, particionamento e armazenamento em camadas. Essas medidas visam fortalecer a robustez, segurança e eficiência do sistema, preparando-o para crescer e adaptar-se às necessidades industriais futuras.

# Referências


Elena, C. (2011). *Business intelligence. Journal of Knowledge Management, Economics and Information Technology.* [Online]
Available at: http://www.scientificpapers.org/wp-content/files/1102_Business_intelligence.pdf

Negash, S. (2004*). Business intelligence. Communications of the association for information systems.* [Online]
Available at: https://aisel.aisnet.org/cgi/viewcontent.cgi?article=3234&context=cais

Heinrichs, J. H., & Lim, J. S. (2003). *Integrating web-based data mining tools with business models for knowledge management. Decision Support Systems* [Online]
Available at:
https://www.sciencedirect.com/science/article/pii/S0167923602000982

Fortulan, M. R., & Gonçalves Filho, E. V. (2005) *Uma proposta de aplicação de Business*
*Intelligence no chão-de-fábrica. Gestão & Produção* [Online]
Available at: https://www.scielo.br/j/gp/a/ydtVGxxBtD65zcx4VmJDJGw/?lang=pt

Ferreira, J. E., ITALIANO, I., & TAKAI, O. (2005). Introdução a Banco de Dados. [Online]
Available at: https://www.ime.usp.br/~jef/apostila.pdf

Chaudhuri, S., & Dayal, U. (1997). *An overview of data warehousing and OLAP technology. ACM Sigmod record*. [Online]
Available at: https://dl.acm.org/doi/pdf/10.1145/248603.248616

Schio, L. C. (2006). *Implementação de Cubo OLAP sobre um Data Warehouse* (Doctoral dissertation, Universidade do Vale do Paraíba) [Online]
Available at: https://biblioteca.univap.br/dados/000044/00004479.pdf

Inmon, W. H. (2005). *Building the data warehouse. John wiley & sons*. [Online]
Available at: https://ia800202.us.archive.org/

Ballard, C., Farrell, D. M., Gupta, A., Mazuela, C., & Vohnik, S. (2012). *Dimensional Modeling: In a Business Intelligence Environment. IBM Redbooks*. [Online]
Available at: https://www.redbooks.ibm.com/redbooks/pdfs/sg247138.pdf

Tryfona, N., Busborg, F., & Borch Christiansen, J. G. (1999, November). starER: A conceptual model for data warehouse design. In Proceedings of the 2nd ACM international workshop on Data warehousing and OLAP (pp. 3-8). [Online]
Available at: https://dl.acm.org/doi/abs/10.1145/319757.319776

Machado, F. N. R. (2004). Tecnologia e projeto de Data Warehouse. Saraiva Educação SA.

Available at: https://books.google.com.br/books?hl=pt-PT&lr=&id=1YqwDwAAQBAJ&oi=fnd&pg=PT2&dq=data+warehouse&ots=P6xNlEFTT6&sig=2dzjBoLgfF1F8m4DfFuEsS_zj5g

Dantas, C. M., Cordula, F. R., & Araújo, W. J. (2016). *Análise da representação da informação em modelos entidade relacionamento com base em metadados.* Archeion Online, João Pessoa, 4(1), 40-63. [Online]
Available at: https://www.researchgate.net/profile/Flavio-Cordula/publication/331787461_ANALISE_DA_REPRESENTACAO_DA_INFORMACAO_EM_MODELOS_ENTIDADE_RELACIONAMENTO_COM_BASE_EM_METADADOS/links/5c8bb4a6a6fdcc381755bbb5/ANALISE-DA-REPRESENTACAO-DA-INFORMACAO-EMMODELOS-ENTIDADE-RELACIONAMENTO-COM-BASE-EM-METADADOS.pdf

Wells, D. (2003). *Making Sense of the Methodology Debate. FlashPoint TDWI, The Data Warehouse Institute*. [Online]
Available at: http://www.dw-institute.com/research/

Gardner, S. R. (1998). Building the data warehouse. Communications of the ACM, 41(9), 52-60. [Online]
Available at: https://dl.acm.org/doi/abs/10.1145/285070.285080

Myrianthous, G. (2023). *Transitioning from ETL to ELT.* Medium [Online]
Available at: https://towardsdatascience.com/from-etl-to-elt-908ce414e39e

Rehman, K. U. U., Ahmad, U., & Mahmood, S. (2018). *A Comparative Analysis of Traditional and Cloud Data Warehouse* [Online]
Available at: http://www.dw-institute.com/research/

Twilio Segment (n.d). *How to choose the right data warehouse* [Online]
Available at: https://segment.com/academy/choosing-stack/how-to-choose-the-right-data-warehouse

Roy, K. (2021). *How to Choose the Right Data Warehouse Storage?* Medium [Online] Available at: https://medium.com/datatobiz/how-to-choose-the-right-data-warehouse-storage-6c30690273cb

Kimball, R., Ross, M., Thorthwaite, W., Becker, B., & Mundy, J. (2008). *The data warehouse lifecycle toolkit.* John Wiley & Sons. [Online]
Available at: https://books.google.com.br/books?hl=pt-PT&lr=&id=XoS2oy1IcB4C&oi=fnd&pg=PA1&dq=The+Data+Warehouse+Lifecycle+Toolkit&ots=1EEdjEjJaA&sig=gqivmC5fEh_4jNRGuxaQUIoRtzY

Hokama, D. D. B., Camargo, D., Fujita, F., Fogliene, J. L. V., & JL, V. (2004). *A modelagem de dados no ambiente Data Warehouse*. São Paulo, 32. [Online]

Available at:
http://meusite.mackenzie.com.br/rogerio/tgi/2004ModelagemDW.pdf

Singhal, B., & Aggarwal, A. (2022, December). ETL, ELT and Reverse ETL: A business case Study. In 2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering [Online] Available at: https://ieeexplore.ieee.org/abstract/document/10046997

Bruzarosco, D. C., Castoldi, A. V., & dos Santos Pacheco, R. C. (2000). Criando data warehouse com o modelo dimensional. Acta Scientiarum, 22(5), 1389-1397. [Online] Available at: https://www.academia.edu/download/85747860/3099-Article_Text9074-1-10-20080513.pdf

Halder, N. (2023). *ETL vs. ELT: A Comprehensive Comparison and Guide to Modern Data Integration Strategies.* Medium [Online]
Available at: https://medium.com/analysts-corner/etl-vs-elt-a-comprehensive-comparison-and-guide-to-modern-data-integration-strategies-f2968bc64651

Tridant (2020). *Benchmark Testing of Snowflake using Tableau* [Online]
Available at: https://www.tridant.com/benchmark-testing-of-snowflake-using-tableau/

Snowflake (2023). *Snowflake* [Online]
Available at: https://www.snowflake.com/en/

AWS (2023). *Amazon Redshift* [Online]
Available at: https://aws.amazon.com/redshift/