NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

# Agricultura Biológica em Portugal

A importância da utilização de ferramentas de Business Intelligence na integração e visualização de dados

Francisco Pereira Gomes Fausto da Costa

Trabalho de Projeto apresentado como requisito parcial para obtenção do grau de Mestre em Gestão de Informação

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

**NOVA Information Management School**
**Instituto Superior de Estatística e Gestão de Informação**
Universidade Nova de Lisboa

# AGRICULTURA BIOLÓGICA EM PORTUGAL: A IMPORTÂNCIA DA UTILIZAÇÃO DE FERRAMENTAS DE BUSINESS INTELLIGENCE NA INTEGRAÇÃO E VISUALIZAÇÃO DE DADOS

por

Francisco Pereira Gomes Fausto da Costa

Trabalho de Projeto apresentado como requisito parcial para a obtenção do grau de Mestre em Gestão de Informação, Especialização em Gestão do Conhecimento e Business Intelligence

**Orientador/Coorientador:** Professor Doutor Miguel de Castro Neto

Novembro 2018

# AGRADECIMENTOS

A excelência no desenvolvimento de um projeto e o potencial do mesmo só fazem sentido se conseguirmos partilhar o sucesso com quem contribuiu para a sua realização.

Assim, quero agradecer, em primeiro lugar, à minha família e namorada, toda a dedicação e empenho para que este projeto se realizasse e a força que me transmitiram para que não desistisse, estando presentes durante toda a realização do projeto.

Em segundo lugar, quero agradecer ao Professor Miguel de Castro Neto pela oportunidade que me deu de desenvolver um projeto com imenso potencial de exploração de dados e que me permitiu melhorar o meu conhecimento técnico sobre várias ferramentas. Acima de tudo, a sua confiança no meu trabalho e acreditar que seria possível é algo que não vou esquecer.

Em terceiro lugar, quero agradecer a todos os amigos e colegas que fizeram este percurso comigo e que com as suas sábias lições me indicaram qual poderia ser um caminho possível para problemas aparentemente sem solução.

Por último, mas não menos importante, quero agradecer à Direção-Geral da Agricultura e Desenvolvimento Rural (DGADR) pela amabilidade que tiveram comigo, em especial à Dr.ª Filipa Osório e à Dr.ª Cristina Hagatong, por acreditarem que seria possível melhorar um processo que facilita-se o seu trabalho diário e se desse evolutivo na concretização do Observatório Nacional de Produção Biológica (ONPB).

A realização deste trabalho foi, e citando Steve Jobs, "*Have the courage to follow your heart and intuition. They somehow already know what you truly want to become*".

Um grande obrigado a todos.

Francisco Pereira Costa

# RESUMO

O presente relatório de projeto tem como objetivo demonstrar uma aplicação prática da implementação de um novo sistema tecnológico de gestão de dados para a Direção-Geral da Agricultura e Desenvolvimento Rural (DGADR), nomeadamente para o Observatório Nacional de Produção Biológica (ONPB). Este sistema de gestão de base de dados é denominado por *Data Warehouse*, isto é, a organização de dados de forma integrada e concebida para otimizar a sua análise.

Devido ao papel que ONPB assume no contexto agrícola português, uma vez que o seu principal prepósito é recolher, tratar e divulgar a informação disponível sobre produção biológica, a implementação de um *Data Warehouse* sobre recolha de dados operacionais em conjunto com o desenvolvimento de um processo de extração, transformação e carregamento (ETL), permitirá um aumento no dinamismo e na forma como se lida com a informação recolhida, tornando possível obter vantagens competitivas. Desta forma, há que destacar algumas das melhorias que serão obtidas deste projeto como a melhoria no processo de recolha de dados e na qualidade dos mesmos e a potencialidade de criação de sistemas analíticos de informação, que funcionem como sistemas de apoio à decisão dos utilizadores.

# PALAVRAS-CHAVE

*Business Intelligence*; *Data Warehouse*; Agricultura Biológica; ETL

# ABSTRACT

This project report aims to show a practical application of the implementation of a new technological data management system for the Direção-Geral da Agricultura e Desenvolvimento Rural (DGADR), specifically for the Observatório Nacional de Produção Biológica (ONPB). This database management system is called *Data Warehouse*, that is, the organization of data in an integrated way and designed to optimize its analysis.

Due to the important role that ONPB plays in the Portuguese organic agricultural context since its main purpose is to collect, process and disseminate available information on the production, the implementation of a *Data Warehouse* on existing operational data collection and the development of an extraction, load and transform (ETL) process will allow an increase in dynamism and in the way it deals with the information collected, making it possible to obtain competitive advantages. Saying that, we must highlight some of the improvements that will be obtained from this project like the improvement in the data collection process and quality of data and the potential for the creation of information reporting systems that function as decision support systems for the users.

# KEYWORDS

*Business Intelligence*; *Data Warehouse*; *Organic Agriculture*; ETL

# ÍNDICE

# ÍNDICE DE FIGURAS

# ÍNDICE DE TABELAS

# LISTA DE SIGLAS E ABREVIATURAS

| | |
|---|---|
| **BI** | Business Intelligence |
| **DGADR** | Direção-Geral da Agricultura e Desenvolvimento Rural |
| **DM** | Data Mart |
| **DW** | Data Warehouse |
| **ENAB** | Estratégia Nacional para a Agricultura Biológica |
| **ETL** | Extração, Transformação e Carregamento (*Extract, Transform and Load*) |
| **OLAP** | Online Analytical Processing |
| **OLTP** | Online Transaction Processing |
| **ONPB** | Observatório Nacional de Produção Biológica |
| **SA** | Staging Area |
| **SQL** | Structure Query Language |

# 1. INTRODUÇÃO

Num ambiente em constante mudança e competitividade, a agricultura tem vindo a assistir a uma evolução acelerada das tecnologias disponíveis para a sua prática (Antle, Jones, & Rosenzweig, 2017). A evolução que se tem assistido no campo das tecnologias de informação e comunicação tem disponibilizado no mercado capacidades computacionais crescentes, permitindo que as máquinas passassem a realizar operações que até agora eram apenas realizadas por seres humanos, tais como regular automaticamente a temperatura ambiente e definir a melhor estratégia para proteção dos diferentes cultivos (Gan & Lee, 2018; Wolfert, Ge, Verdouw, & Bogaardt, 2017).

Os sistemas agrícolas produzem dados que permitem aos investigadores considerar problemas complexos ou tomar decisões agrícolas informadas (Zhao et al., 2018). Desta forma, surge a necessidade de se desenvolverem sistemas integrados de informação que consigam responder às necessidades do mercado em tempo real, oferecendo reatividade e competitividade. Os modelos são necessários para compreender e prever o desempenho geral de um sistema (Janssen et al., 2017), tendo os dados um papel fundamental no desenvolvimento, avaliação e execução de modelos para que possam ser tomadas decisões baseadas em informação real.

Segundo a Comissão Europeia (2018), a agricultura biológica é um sistema agrícola que procura fornecer ao consumidor, alimentos frescos, saborosos e autênticos, respeitando os processos naturais de ciclo de vida. Os dados associados à agricultura biológica têm vindo a sofrer um aumento ao longo dos últimos anos. Segundo estatísticas do *The World of Organic Agriculture* (Willer, H. and Lernoud, 2016), cerca de 1% de toda a terra agrícola foi considerada biológica. Por região, as maiores parcelas biológicas de terras agrícola estão na Oceânia (4,1%) e na Europa (2,4%), sendo que na União Europeia, 5,7% de toda a terra agrícola é biológica. No entanto, alguns países têm proporções muito mais altas: Falkland (36,3%), Liechtenstein (30,9%), Áustria (19,4%). Em 2014, existiam 2,3 milhões de produtores biológicos sendo que 40% da produção biológica mundial está na Ásia, seguidos de África (26%) e da América Latina (17%). Os países com mais produtores são a Índia (650.000), Uganda (190.552) e México (169.703). Mais de um quarto das terras agrícolas biológicas do mundo (11,7 milhões de hectares) e mais de 86% (1,9 milhões) dos produtores estavam em países em desenvolvimento e mercados emergentes em 2014.

Devido ao enorme volume de dados e à crescente expansão da agricultura biológica em todo o mundo, surge a necessidade de se utilizarem ferramentas de *Business Intelligence*, uma vez que conseguem fornecer informação de vários níveis organizacionais que seja oportuna, relevante e fácil de compreender (Dooley, Levy, Hackney, & Parrish, 2018; Popovič, Hackney, Coelho, & Jaklič, 2014).

Neste contexto, surge o Observatório Nacional de Produção Biológica, que tem por objetivo fornecer informações dos vários operadores de agricultura biológica em Portugal.

Tentando responder ao desafio da constante evolução tecnológica, pretende-se uniformizar a recolha, promover a integração e implementar novas formas de tratamento dos dados de agricultura biológica. Assim, através da utilização dos recursos fornecidos pela DGADR, através do Professor Doutor Miguel de Castro Neto, orientador desta tese de mestrado, pretende-se colaborar no processo de recolha de dados de operadores de controlo, ajudando no tratamento dos mesmos; implementar um processo de ETL, assente num *Data Warehouse* por forma a alcançar-se informação integrada e padronizada; construção de um processo de visualização de dados que gere conhecimento e, consequentemente, colabore no suporte à tomada de decisão.

## 1.1 OBJETIVO DO TRABALHO

O presente projeto tem como principal objetivo a implementação de um modelo de *Data Warehouse* para dados de agricultura biológica, num sistema informático de recolha de informação pouco desenvolvido até ao momento. Como forma de atingir este objetivo, foi construído um processo de ETL, que permitirá um uso mais eficiente do sistema de gestão de bases de dados, otimizando o processo de consulta da mesma. Ao mesmo tempo, foi também melhorada a fonte de recolha de dados, de forma a haver uma verificação dos dados inseridos e reduzir o enviesamento dos mesmos, através de uma limpeza de dados. Para além disso, foi desenvolvido um processo de visualização de dados através da implementação de *dashboards* dinâmicos que permitam analisar a informação em tempo real. Por último, o projeto tem o objetivo de promover os benefícios inerentes a um aproveitamento da informação disponibilizada pelos organismos de controlo que reportam à DGADR.

## 1.2 ENQUADRAMENTO DE NEGÓCIO

O âmbito deste projeto foi construído com base num enquadramento dos objetivos, missão e descrição do negócio, neste caso, agricultura biológica. Este projeto terá como objetivo a implementação de um modelo de *Data Warehouse* para DGADR, no âmbito da criação do ONPB, pelo que se torna relevante definir no que consiste este mesmo organismo.

### 1.2.1 DIREÇÃO-GERAL DA AGRICULTURA E DESENVOLVIMENTO RURAL

A DGADR tem como missão contribuir para a execução das medidas aplicadas no âmbito das atividades e exploração agrícola, dos recursos genéticos agrícolas, da qualificação dos agentes rurais, entre outros.

Tem como principais atribuições contribuir para a formulação da estratégia, das prioridades e objetivos nas áreas da sua missão, tentando ao longo dos anos contribuir para criar e manter atualizado um sistema de informação sobre o regadio e sobre as infraestruturas que o sustentam.

É o organismo competente por fazer executar e implementar a Estratégia Nacional para a Agricultura Biológica em Portugal, apresentando semestralmente relatórios de progresso ao membro do Governo responsável. Na figura 1 é possível observarmos o organograma atual da DGADR.

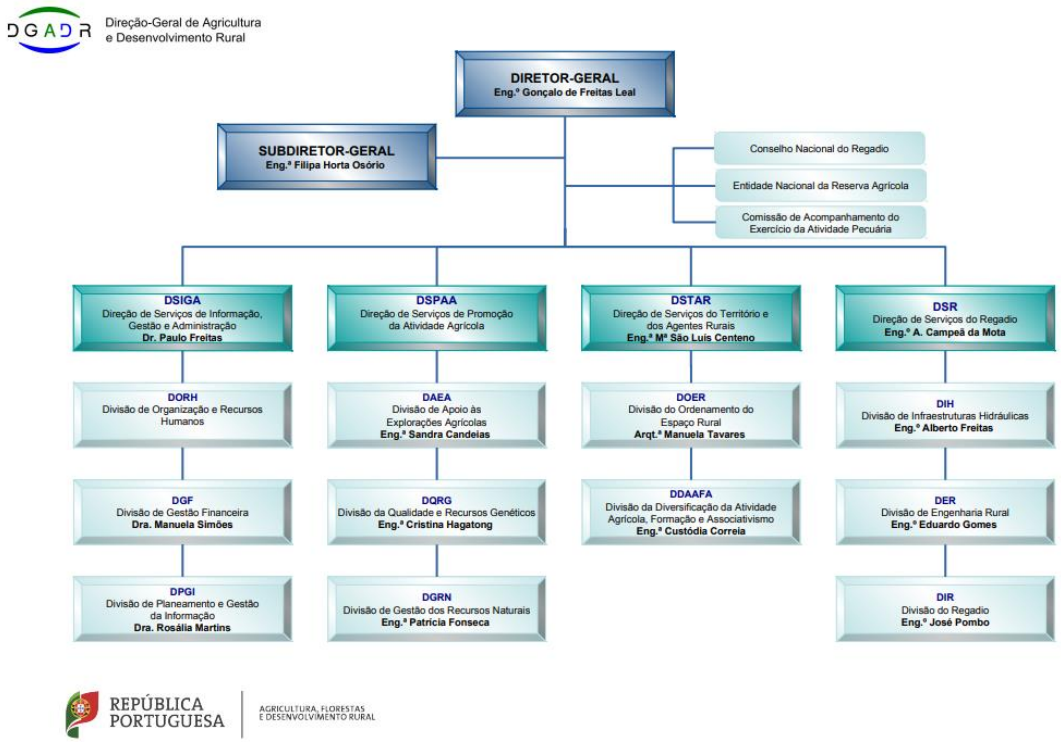


**Figura 1.1 - Organograma DGADR**

### 1.2.2 OBSERVATÓRIO NACIONAL DE PRODUÇÃO BIOLÓGICA

O XXI Governo Constitucional assumiu no seu programa o compromisso de definir uma Estratégia Nacional para a Agricultura Biológica e pôr em execução um plano de ação para a produção e promoção de produtos agrícolas e géneros alimentícios biológicos (DGADR, 2017b). Um dos pontos que esta estratégia assume é o de criar o ONPB. Através deste organismo, pretende-se recolher, tratar e divulgar a informação disponível sobre produção, transformação e comercialização de produtos biológicos, incluindo sobre o seu consumo e sobre os vários mercados existentes.

Desta forma, com este projeto pretende-se ir de encontro aos pressupostos do ONPB e apoiar a criação e o desenvolvimento de um portal de dados sobre agricultura biológica que reúna num único ponto a informação obtida no âmbito das atividades dos organismos de controlo.

### 1.2.3 CARACTERIZAÇÃO DA AGRICULTURA E DA PRODUÇÃO BIOLÓGICA EM PORTUGAL

No presente capítulo pretende-se enquadrar e descrever os principais aspetos que caracterizam a agricultura e a produção biológica em Portugal. A informação dos próximos subcapítulos tem em conta o Diário da República, 1.ª Série - N.º 144 de 27 de julho de 2017 (DGADR, 2017a).

#### 1.2.3.1 SUPERFÍCIE CULTIVADA

Segundo a DGADR, a superfície em agricultura biológica (AB) em Portugal continental é de 239.864 hectares, sendo a região do Alentejo (cerca de 64% - 152.969 hectares) a que possui maior área de ocupação, como é possível verificar pela figura 2. As áreas de pastagens representam cerca de 70% desta superfície, onde as culturas com maior representatividade são o olival (9%), as culturas forrageiras (cerca de 8%) e os frutos secos (4%). A ocupação cultural pode ser vista na figura 3.

**Agricultura biológica — Área total, n.º de produtores agrícolas e área média das explorações de agricultura biológica**

| Regiões | Área | Produtores | Área média |
|---|---|---|---|
| | ha | nº | ha |
| Continente | 239.864 | 3.820 | 63 |
| Entre-Douro e Minho | 8.799 | 476 | 18 |
| Trás-os-Montes | 17.176 | 966 | 18 |
| Beira Litoral | 2.279 | 244 | 9 |
| Beira Interior | 44.547 | 716 | 62 |
| Ribatejo e Oeste | 11.276 | 360 | 31 |
| Alentejo | 152.969 | 959 | 160 |
| Algarve | 2.818 | 99 | 28 |

**Figura 1.2 - Explorações de Agricultura Biológica** (DGADR, 2017a)

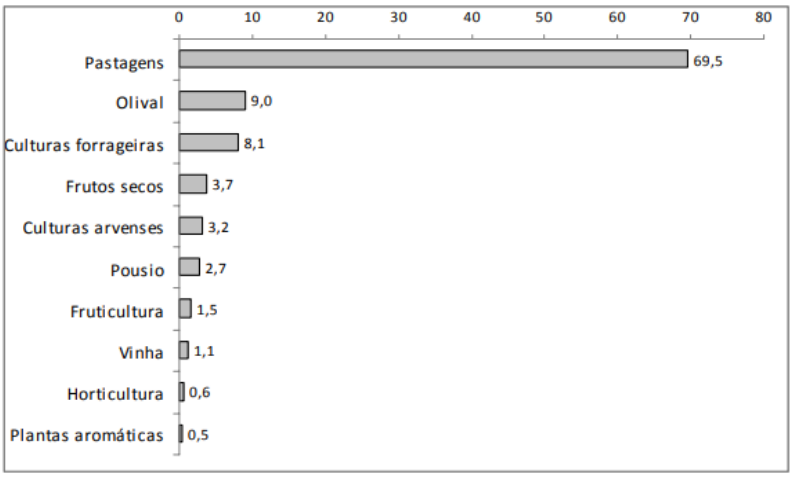


**Figura 1.3 - Ocupação cultural da superfície em Agricultura Biológica** (DGADR, 2017a)

### 1.2.3.2 Efetivos Pecuários

O efetivo pecuário biológico no ano de 2015 representa um total de 96.876 cabeças de bovinos, 108.337 de ovino, enquanto que as aves atingem o valor de 61.062 bicos. Ao nível da apicultura registam-se 55.000 colmeias. Verificam-se ainda alguns efetivos de suínos, caprinos e equídeos, contudo sem grande expressividade.

**Efetivo pecuário em agricultura biológica, por espécies — Continente**

Unidade: nº de cabeças

| Ano | Bovinos | Suínos | Caprinos | Ovinos | Equídeos | Aves | Apicultura (nº colmeias) |
|---|---|---|---|---|---|---|---|
| **2002** | 8.202 | 3.091 | 1.440 | 38.072 | 107 | 7.024 | 130 |
| **2003** | 18.329 | 3.507 | 2.341 | 63.026 | 103 | 12.164 | 248 |
| **2004** | 36.653 | 5.495 | 3.551 | 94.119 | 145 | 37.573 | 738 |
| **2005** | 56.896 | 5.487 | 5.219 | 114.085 | 126 | 46.438 | 1.439 |
| **2006** | 58.968 | 5.578 | 6.301 | 115.068 | 155 | 70.584 | 1.499 |
| **2007** | 68.768 | 8.369 | 5.801 | 111.021 | 388 | 44.557 | 3.608 |
| **2008** | 69.097 | 9.499 | 6.525 | 106.682 | 278 | 41.998 | 6.122 |
| **2009** | 62.376 | 4.165 | 5.894 | 79.903 | 301 | 53.440 | 9.494 |
| **2010** | 65.524 | 4.381 | 6.838 | 96.874 | 274 | 57.002 | 15.927 |
| **2011** | 65.291 | 3.304 | 7.952 | 93.523 | 200 | 46.071 | 26.397 |
| **2012** | **68.004** | 2.636 | 8.765 | **90.665** | 192 | 44.611 | 32.409 |
| **2013** | 68.310 | 2.009 | 6.512 | 88.405 | 167 | 45.208 | 33.916 |
| **2014** | 73.359 | 1.721 | 6.554 | 91.085 | 154 | 56.910 | 47.043 |
| **2015** | **96.876** | 829 | 6.467 | **108.337** | 177 | 61.062 | 55.001 |

**Figura 1.4 - Efetivos Pecuários em Agricultura Biológica, por Espécie** (DGADR, 2017a)

### 1.2.3.3 PRODUTORES AGRÍCOLAS

No ano de 2015, atingiram-se os 3.837 de produtores biológicos, o que corresponde ao maior número existente no Continente, no período entre 1994 e 2015. Esta evolução pode ser vista na figura 5.

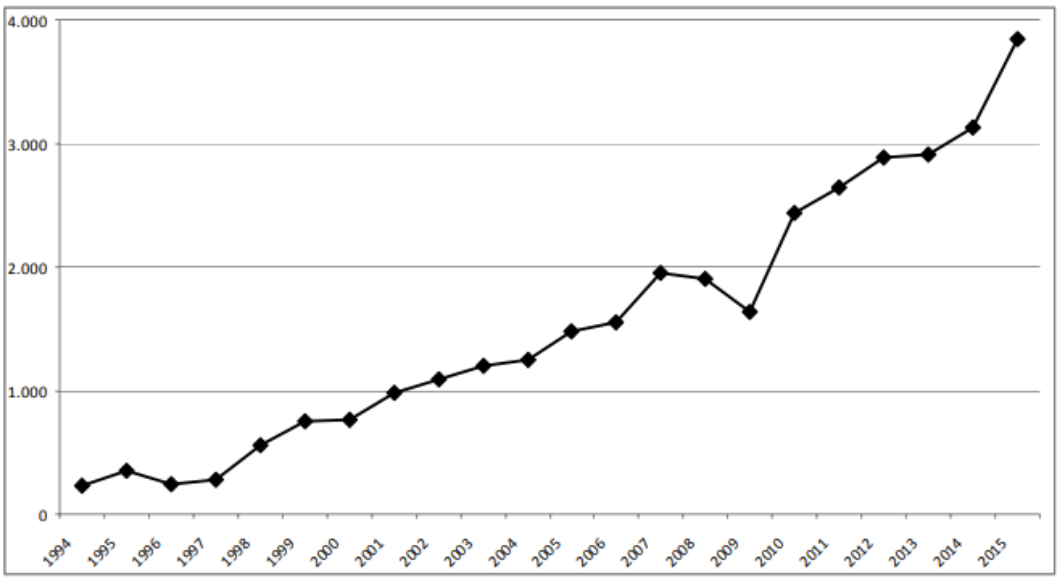


**Figura 1.5 - Número Total de Produtores** (DGADR, 2017a)

### 1.2.3.4 PRODUTORES PECUÁRIOS

O número total de produtores pecuários biológicos era de 446 no ano de 2004, enquanto que esse valor quase triplicou, atingindo, em 2015, os 1324 produtores.

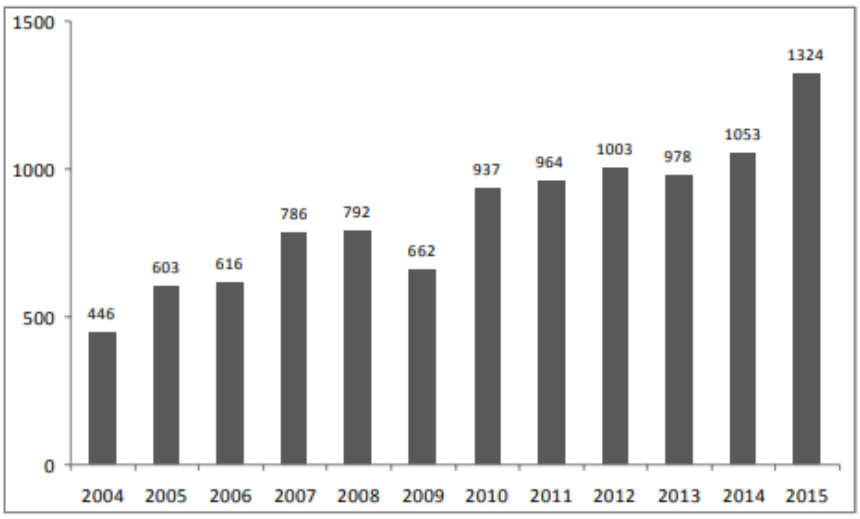


**Figura 1.6 - Número de Produtores Pecuários Biológicos** (DGADR, 2017a)

### 1.2.3.5 IMPORTADORES

No que respeita aos importadores de produtos biológicos é possível realizar uma análise mais detalhada sobre a evolução desta atividade, com base nas validações dos certificados de importação de produtos biológicos que entraram em Portugal.

Pode verificar-se que entre 2014 e 2016, houve um aumento do número de operadores nesta atividade, numa variação de 125%. O maior número de operadores traduziu-se num aumento exponencial, de 2014 para 2016, tanto do número de importações (variação de 450%), como das quantidades importadas (variação de 732%), como é possível observar pela figura 8.

| | 2014 | 2015 | 2016* | Variação 2014/2016 % |
|---|---|---|---|---|
| Quantidade importada (kg) | 46.674 | 45.870 | 388.181 | 732% |
| Número de importações | 8 | 50 | 44 | 450% |
| Número de importadores | 4 | 6 | 9 | 125% |

*dados apurados até setembro de 2016

**Figura 1.7 - Importações de Produtos Biológicos** (DGADR, 2017a)

Em relação aos países de origem da importação de produtos biológicos, como se pode verificar pela figura 9, a maior quantidade importada provém da China e do Equador, responsáveis por cerca de 76% do volume total importado ao longo dos 3 anos.

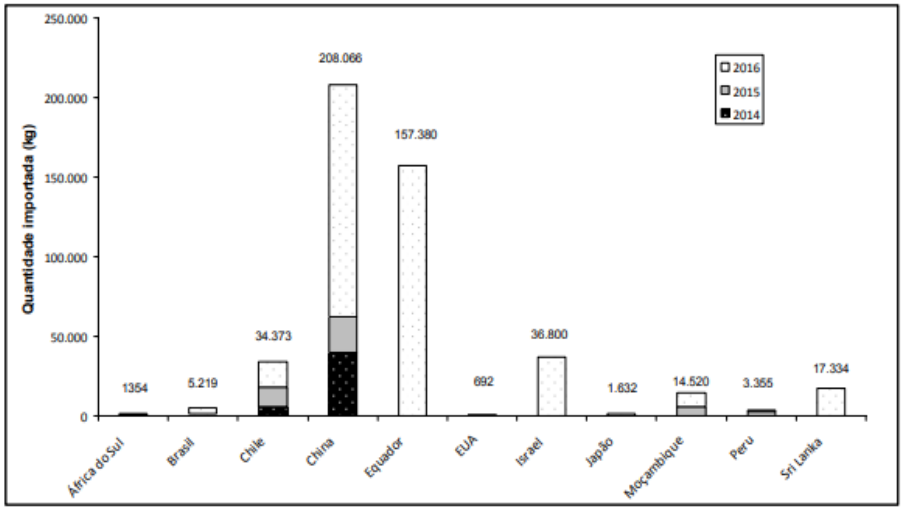


**Figura 1.8 - Países de Origem da Importação de Produção Biológica** (DGADR, 2017a)

## 1.3 ESTRUTURA DO RELATÓRIO

O presente relatório seguirá a seguinte estrutura:

| CAPÍTULO | DESCRIÇÃO |
|---|---|
| **INTRODUÇÃO** | Capítulo introdutório onde é descrito qual é o objetivo do projeto e um enquadramento do negócio do cliente e da sua importância a nível nacional. |
| **REVISÃO DE LITERATURA** | Capítulo onde são explicados os principais conceitos como *Business Intelligence*, *Data Warehouse*, ETL e *Data Visualization*. |
| **METODOLOGIA** | Capítulo é explicada a estrutura das diferentes bases de dados envolvidas e a metodologia de construção da *Staging Area* e do *Data Warehouse*. É também descrito o processo de extração, transformação e carregamento (ETL) implementado, tanto de passagem do ficheiro fonte para a *Staging Area* como da *Staging Area* para o *Data Warehouse*. |
| **CONCLUSÕES** | Capítulo onde são explicadas as conclusões retiradas da implementação do projeto. |
| **LIMITAÇÕES** | Capítulo que expõe as limitações encontradas na implementação do projeto. |
| **RECOMENDAÇÕES PARA TRABALHOS FUTUROS** | Capítulo onde são elaboradas algumas propostas de trabalho futuro. |

**Tabela 1 - Estrutura do Relatório**

# 2. REVISÃO DE LITERATURA

Neste capítulo serão descritas as tecnologias utilizadas, assim como estudos relevantes, com o intuito de contextualizar o projeto, de modo a compreender-se melhor o problema em estudo.

## 2.1 *BUSINESS INTELLIGENCE*

*Business Intelligence* engloba uma ampla variedade de ferramentas, aplicações e metodologias que permitem às organizações recolher dados de sistemas internos e fontes externas, transformá-los de acordo com as necessidades de negócio, criar relatórios analíticos e disponibilizar a informação de forma a facilitar a tomada de decisão organizacional (Chen, Chiang, & Storey, 2012; Larson & Chang, 2016).

O esforço despendido na construção de aplicações de *Business Intelligence*, sem o suporte de dados multidimensionais e sem processos de extração, transformação e carregamento (ETL), torna-se moroso e ineficaz (Trieu, 2017). Deste modo, o *Business Intelligence* surge como uma ferramenta capaz de providenciar conhecimento aos utilizadores, a vários níveis organizacionais, com informação útil e fácil de usar (Popovič et al., 2014).

*Business Intelligence* permite aceder, explorar e estruturar as informações armazenadas num *Data Warehouse* ou *Data Mart*. Esta capacidade, conjugada com as ferramentas tecnológicas adjacentes, determinou a forma como as tecnologias de informação são encaradas atualmente (Dooley et al., 2018), ou seja, *Business Intelligence* é vista como uma ferramenta necessária de implementar nas organizações como catalisador da competitividade de uma empresa (Larson & Chang, 2016; Trieu, 2017). A figura 10 mostra os principais componentes de um ambiente de *Business Intelligence* desenvolvidos para o presente projeto.

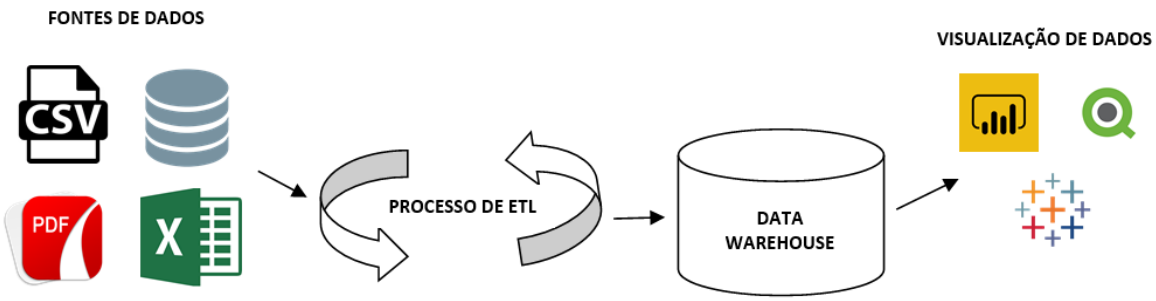


**Figura 2.1 - Processo de Business Intelligence**

Conforme é possível observar, um ambiente de *Business Intelligence* é composto por um sistema de bases de dados transacional, por um processo de ETL que carrega os dados no *Data*

*Warehouse* e, por fim, por um conjunto de ferramentas que permitem explorar os dados. Estas ferramentas efetuam análises designadas por *Online Analytical Processing* (OLAP) e utilizam metodologias como o *Data Mining* que visam, por exemplo, a identificação de padrões implícitos nos dados (Hema & Malik, 2010).

## 2.2 *DATA WAREHOUSE*

Os sistemas fonte, normalmente, representam estruturas de dados heterogéneas. Estas podem residir na organização ou ser providenciadas por um fornecedor externo. Quem trabalha com *Data Warehouse* extrai dados dos sistemas de origem e transforma-os, de forma a que sejam importantes no suporte à tomada de decisão. Um exemplo disto é a possibilidade de todos os registos de vários sistemas poderem ser combinados e consolidados com base num número de identificação (Baars & Kemper, 2008).

Conseguir obter dados relevantes é um dos aspetos mais desafiante, exigindo cerca de 80% do tempo e esforço e gerando mais de 50% dos custos inesperados de um projeto de *Business Intelligence* (Watson & Wixom, 2007). Existem várias causas, como a baixa qualidade de dados nos sistemas de origem ou a política de propriedade de dados e ferramentas desatualizadas. Os dados, quando armazenados, devem ser orientados por processo de negócio, variantes no tempo e não voláteis (Popovi, Coelho, & Jakli, 2009). Dependendo da arquitetura, o *Data Warehouse* pode alimentar *Data Marts*, que têm um âmbito mais restrito concentrando-se, sobretudo, numa área funcional, região geográfica, aplicação ou divisão organizacional específica.

A manutenção de dados de suporte num *Data Warehouse* ou *Data Mart* tem de garantir "uma única versão da verdade" (Dyché & Levy, 2006). Desta forma, os metadados desempenham um papel fundamental, uma vez que descrevem os valores de cada campo, os tamanhos, as definições e os processos de transformação dos mesmos, fornecendo transparência à medida que os dados são transferidos do sistema de origem para o utilizador final.

O *Data Warehouse* deve tornar as informações de uma organização facilmente acessíveis e o conteúdo deve ser compreensível. Os dados devem ser intuitivos e óbvios para quem os vais utilizar e não só para quem desenvolveu o sistema e as ferramentas de acesso devem ser simples e fáceis de usar, retornando os resultados de consulta de forma rápida e eficiente. Os dados devem ser cuidadosamente montados a partir de diferentes fontes. Contudo, devem passar por um processo de limpeza e utilizados apenas quando estiverem aptos para o utilizador final (Kimball, Reeves, Ross, & Thornthwaite, 2008).

As informações de um processo de negócio devem corresponder às informações de outro. Se duas medidas de desempenho tiverem o mesmo nome devem tem o mesmo significado. Por outro lado, se duas medidas não significam o mesmo, devem ser identificadas de forma diferente (Baars & Kemper, 2008; Kimball et al., 2008). Informações consistentes significam informações de alta qualidade. Isso significa que todos os dados são contabilizados e completos. A consistência também implica que definições comuns para o conteúdo do *Data Warehouse* estejam disponíveis para o utilizador.

Uma vez que não conseguimos evitar a mudança, um *Data Warehouse* deve ser adaptável e resiliente. As alterações no mesmo devem ser fáceis, o que significa que não invalidam dados ou aplicações existentes. Se os dados descritivos forem modificados, devemos contabilizar as alterações adequadamente (Di Tria, Lefons, & Tangorra, 2017).

### 2.2.1 NORMALIZAÇÃO DE DADOS

A normalização numa base de dados aumenta substancialmente a integridade dos dados inseridos, uma vez que reduz o impacto das transações processadas (Kimball et al., 2008). Por outro lado, o aumento na normalização de uma base de dados reduz a performance da mesma, ou seja, a rapidez de acesso aos dados. De acordo com os objetivos que se pretendem atingir com a implementação de um *Data Warehouse*, a definição standard deste sistema é equivalente à de desnormalização, ou seja, à da eliminação da complexidade nas pesquisas, mas colocando entraves à modificação dos dados (Kimball & Ross, 2011). De um modo geral, as bases de dados relacionais, utilizadas para o processamento transacional de dados (OLTP) são tipicamente normalizadas, sendo que as bases de dados utilizadas para o processamento analítico de dados (OLAP) são tendencialmente desnormalizadas.

As regras de normalização estão divididas nas seguintes formas:

- **Primeira Forma Normal (1NF)**: Contém tabelas onde os atributos têm valor atómico ou singular, armazenados no mesmo domínio e onde a ordem de carregamento não importa.
- **Segunda Forma Normal (2NF)**: Todos os atributos que não são chave são totalmente funcionais e dependem de uma chave primária. É, normalmente, utilizada para o processamento analítico de dados (OLAP).
- **Terceira Forma Normal (3NF)**: Todos os atributos que não são chave são independentes uns dos outros, ao mesmo tempo que devem ser dependentes, exclusivamente, da chave primária. É, normalmente, utilizada para o processamento transacional de dados (OLTP).

### 2.2.2 MODELO DIMENSIONAL

A modelação dimensional é amplamente aceite como a técnica mais utilizada para apresentar dados analíticos (Kimball & Ross, 2011), uma vez que tem em conta os seguintes requisitos:

- Fornece informação que é compreensível e relevante para o negócio;
- Retorna os resultados das pesquisas de uma forma rápida e eficiente.

Há mais de cinco décadas, as organizações e consultores de negócio migraram, de forma natural, para uma estrutura dimensional simples de forma a ir ao encontro da necessidade humana de simplicidade. Isto é fundamental porque garante que os utilizadores possam compreender facilmente os dados, assim como permite que se navegue e entregue resultados de forma célere e eficaz (Claudia, Nicholas, & Geiger, 2003; Kimball & Ross, 2011).

A abordagem mais utilizada e sugerida por Kimball e Ross (2011), consiste num modelo dimensional constituído por tabelas de fatos e pelas dimensões associadas. A tabela de fatos é constituída por um conjunto de chaves estrangeiras que vêm das tabelas de dimensão - *Surrogate Keys* - conjuntamente com métricas calculadas nos processos de transformação. No entanto, existem outras abordagens à modelação dimensional. As metodologias mais comuns são a arquitetura em estrela (*Star Schema*) e arquitetura em foco de neve (*Snowflake Schema*) (Levene & Loizou, 2003).

A arquitetura em estrela consiste numa tabela central (tabela de fatos) que se relaciona com as restantes (tabelas de dimensão), tal como representado na figura 11. Esta é uma arquitetura que apresenta algumas vantagens como a redução de redundância e um rácio elevado de desnormalização, permitindo que a execução de *queries* tenha maior performance. Por outro lado, apresenta-se como mais complexa, sendo necessário clareza e consistência na formulação das tabelas, para não se gerarem problemas de redundância de dados.

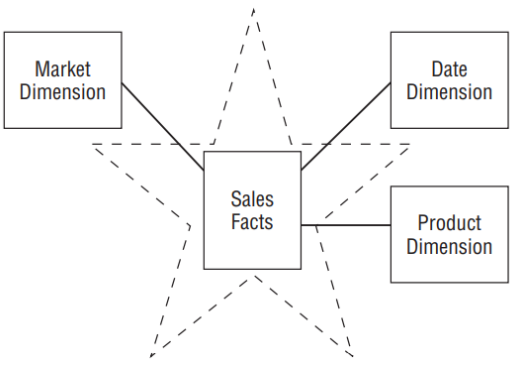


**Figura 2.2 – Arquitetura em Estrela (Kimball & Ross, 2011)**

## 2.3 ETL

O processo de extrair dados de vários sistemas, transformá-los para atender às necessidades de negócio e carregá-los numa ferramenta de destino é denominado de ETL, que significa extração, transformação e carregamento(Salaki, Waworuntu, & Tangkawarow, 2016).

O ETL viu pela primeira vez um aumento de popularidade durante a década de 1970, quando as organizações começaram a usar bases de dados para armazenar diferentes tipos de informação (Kakish & Kraft, 2012). Rapidamente, tornou-se o principal método para obter dados de diferentes fontes, transformá-las e carregá-las em diversas ferramentas (Kimball & Caserta, 2014). Algumas décadas depois, os *Data Warehouses* tornaram-se a grande novidade, fornecendo uma base de dados que integrava informações de vários sistemas. A fim gerir a constante mudança da tecnologia digital nos últimos anos, o número de sistemas de dados, fontes e formatos aumentou exponencialmente, mas a necessidade de processos de ETL permaneceu tão importante quanto a estratégia de integração de dados de uma organização (Bergamaschi, Guerra, Orsini, Sartori, & Vincini, 2011).

A sigla ETL (*Extract, Transform and Load*) tem uma denominação transversal do processo que descreve. Este engloba todas as operações relacionadas com a extração, limpeza, transformação e carregamento de informação transacional num *Data Warehouse*. Segundo Todman (2001), ETL é utilizado para descrever "o processo de extração de dados de um sistema fonte e posteriormente de modificação ou transformação desses mesmos dados para um formato que seja mais aceitável para o *Data Warehouse* do que o seu formato inicial".

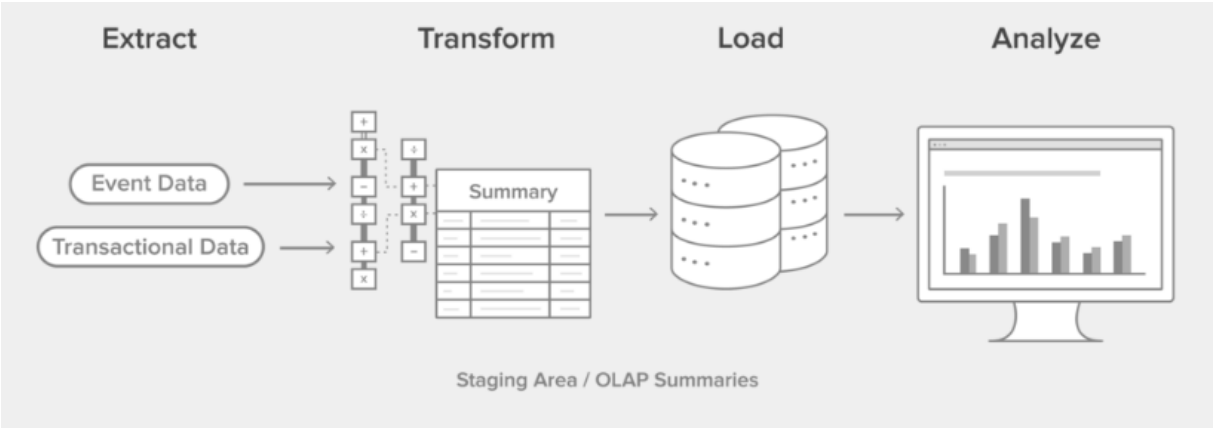


**Figura 2.3 - Processo de ETL**

Um sistema de ETL é dividido em três partes principais: a extração, transformação e o carregamento.

A primeira etapa é a extração de dados. Esta etapa é responsável por extrair dados dos sistemas de origem. Durante a extração de dados de diferentes fontes, deve entender-se a estrutura dos mesmos e saber lidar com fontes de diferentes naturezas. É também, considerada como uma etapa periódica que deve ser atualizada de acordo com as necessidades de negócio, extraindo apenas os dados alterados desde a última execução do processo de ETL.

A segunda etapa é a transformação de dados. Nesta etapa é feita a limpeza e uniformização dos dados recebidos de forma a obter dados mais precisos e consistentes e menos ambíguos. É nesta etapa que se define todas as regras de transformação.

Carregar dados para o *Data Warehouse* é a etapa final de um processo de ETL. Nesta fase, os dados extraídos e transformados são carregados para as estruturas dimensionais, isto é, os dados são carregados paras as tabelas de fato e de dimensões. Este processo despoleta um conjunto de ações na base de dados, desde atualizações a verificações referenciais e de integridade dos dados (El-Sappagh, Hendawi, & El Bastawissy, 2011; Kimball & Caserta, 2014).

## 2.4 VISUALIZAÇÃO DE DADOS

Um sistema interativo de visualização de dados de negócio, organizados de acordo com o modelo dimensional, permite combinar dados de várias fontes de dados e apresentá-los de forma simples e intuitiva. Este é um dos principais objetivos do *Business Intelligence* - facilitar o acesso aos dados e criar condições de análise que colaborem no suporte à tomada de decisão (Ballantyne, 2001).

Existe cada vez mais a necessidade de utilização de ferramentas de visualização de dados que ajudem os utilizadores a compreendê-los. Neste contexto surgem os *dashboards*. Um exemplo de uma definição de um *dashboard* de *Business Intelligence* é "*a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance*" (Few, 2006).

Apesar de ser possível mostrar quase tudo num *dashboard*, existe pelo menos uma característica que deve ser respeitada quando abordamos este tópico – a informação exposta deve ser sucinta e fácil de compreender. Um *dashboard* deve ser capaz de indicar quais os principais pontos que merecem atenção e que podem exigir algum tipo de ação. Não é necessário que indique todos os detalhes necessários para se agir, mas deve ser tão fácil e transparente quanto possível obter essa informação, levando a que por vezes uma mudança na forma como olhamos para o mesmo, retire diferentes conclusões. Posto isto, torna-se necessário ter uma perspetiva diferente em determinados tópicos, usando forma de pesquisa como o *drill down*, para analises mais detalhadas (Few, 2006).

Os *dashboards* precisam ser adaptados de forma a evidenciar melhor o que é relevante dentro de grandes volumes de dados e que esteja de acordo com as necessidades de negócio. Este deve ser um processo claro não só para quem o constrói, mas também para quem o observa (Elias, Aufaure, & Bezerianos, 2013).

# 3. METODOLOGIA

## 3.1 REQUISITOS INICIAIS

Um requisito é uma declaração sobre uma necessidade de negócio que é acordada pelas partes envolventes no sentido de resolver ou melhorar um determinado problema. É a base para o desenvolvimento de uma ideia e validação de qualquer produto (Abai, Yahaya, & Deraman, 2013; Kiritani & Ohashi, 2015). Desta forma, torna-se fundamental para definir o propósito e processo de um projeto, ajudando a analisar e a gerir o mesmo. Os requisitos de qualidade são essenciais para o desenvolvimento e execução de qualquer projeto.

No desenvolvimento do presente projeto foram definidos e priorizados os seguintes requisitos:

- Desenvolvimento e melhoria do processo de recolha de dados, de forma a facilitar o tratamento dos mesmos;
- Desenvolvimento de um processo de ETL que possibilite a transformação dos dados e o carregamento dos mesmos num *Data Warehouse*;
- Desenvolvimento de um *dashboard* que possibilite analisar a informação de uma forma simples e objetiva que suporte a tomada de decisão.

A figura 12 ilustra o fluxo das várias tarefas que foram desenvolvidas ao longo do presente projeto.

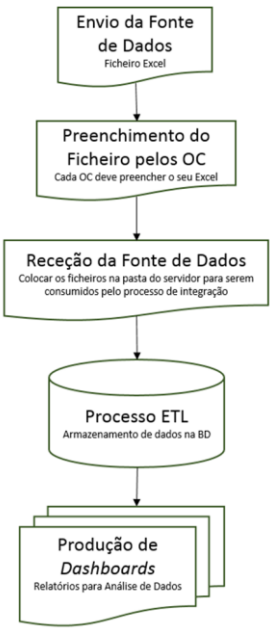


**Figura 3.1 - Fluxo de Projeto**

## 3.2 FONTE DE DADOS

Todos os anos, de forma a controlar a atividade de agricultura biológica dos vários operadores é recolhida informação dos vários organismos de controlo que reportam à DGADR, no sentido de se compreender a evolução da agricultura biológica no território português.

A recolha é feita através de um ficheiro de Excel que é enviado para cada um dos organismos de controlo. Neste ficheiro são recolhidas informações que permitem analisar os operadores de cada organismo de controlo, assim como os contratos e certificados associados aos mesmos. Para além disto, é possível complementar com alguma informação adicional, nomeadamente, a produção animal, a produção vegetal, atividades de importação e exportação, tendo em vista o enriquecimento do relatório anual de controlo.

Com o presente projeto, propôs-se desenvolver para a componente de fonte de dados, os seguintes pontos:

- ✓ ***IMPLEMENTAÇÃO DE UM PROCESSO DE LOGIN DE UTILIZADORES (CADA OC TERÁ O SEU PRÓPRIO UTILIZADOR E EXISTIRÁ UM ADMINISTRADOR PARA RESOLVER QUESTÕES TÉCNICAS RELACIONADAS COM A GESTÃO DOS MESMOS). DEVE SER ENVIADO UM EXCEL PARA CADA OC QUE DEVE FAZER LOGIN COM O SEU UTILIZADOR.***

Para este desenvolvimento, utilizou-se a componente de programador do Excel, onde através de código em Visual Basic foi possível criar um ecrã de login como ilustrado na figura 13.

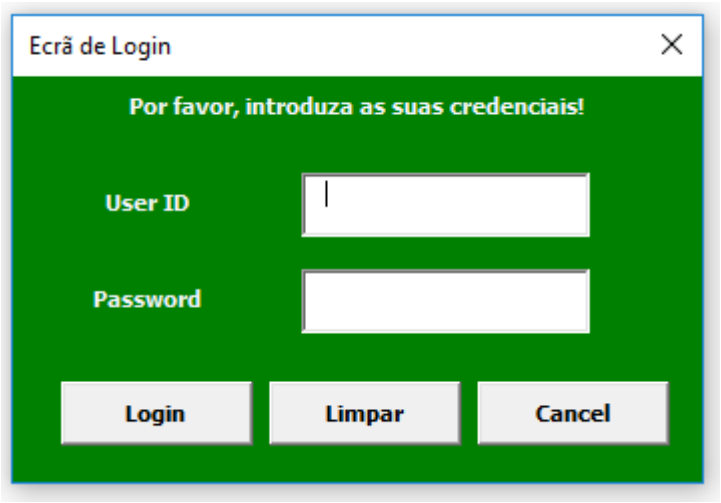


**Figura 3.2 - Ecrã de Login**

O utilizador deve inserir o user ID e password que lhe foram atribuidos de forma a conseguir aceder ao ficheiro de Excel. Ao longo do processo vão surgindo várias janelas de pop up que indicam se a informação foi inserida corretamente ou não e qual a ação que o utilizador deve realizar, como inserir o seu user ID (figura 14) ou inserir a sua password (figura 15).

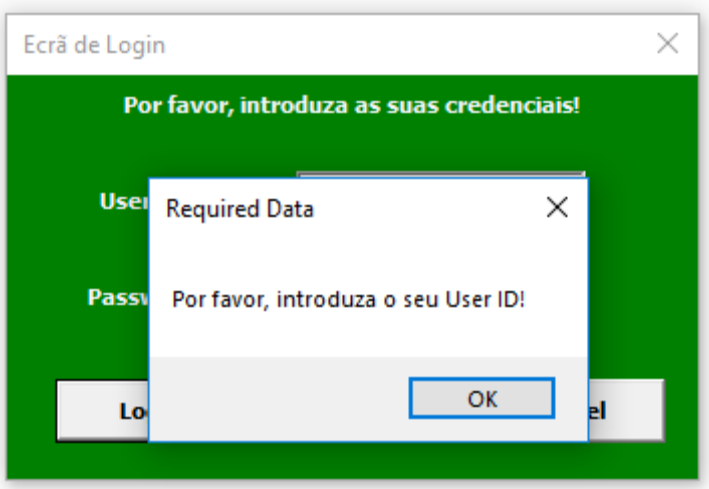


**Figura 3.3 - Introduzir user ID**

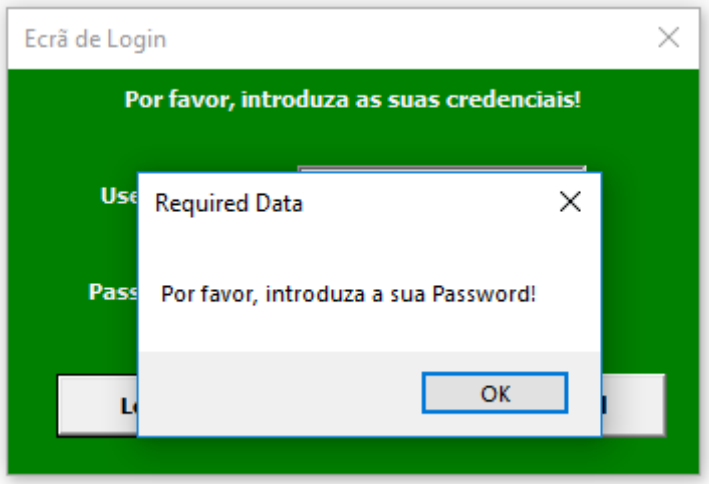


**Figura 3.4 - Introduzir Password**

Caso os dados inseridos pelo utilizador se encontrem incorretos, o pop up representado na figura 16 aparacerá e a informação será eliminada para que o processo possa ser iniciado de novo.

Por outro lado, se os dados forem inseridos corretamente, o pop up representado pela figura 17 aparacerá e o utlizador conseguirá entrar no ficheiro.

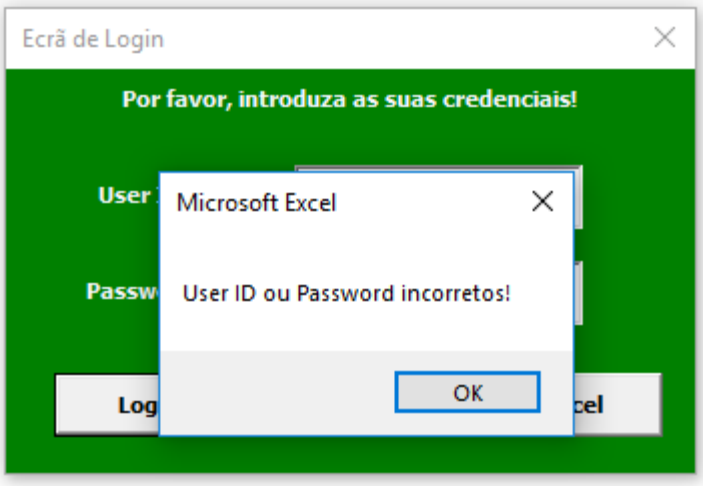


**Figura 3.5 - Dados Introduzidos Incorretos**

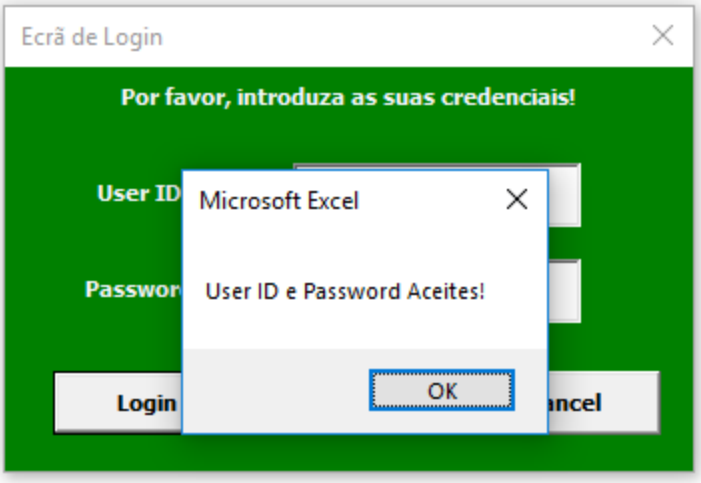


**Figura 3.6 - Dados Introduzidos Aceites**

Para o caso de o utilizador não se recordar do seu user ID ou password, foi criado um utilizador de administração que tem acesso a uma página de login onde consegue verificar todos os utilizadores existentes e as respetivas passwords (figura 18). Deste modo, sempre que um utilizador necessitar de ajuda técnica com o login deve questionar a DGADR que entrará com o utilizador de administração e fornecerá a informação necessária.

# CONTROLO DE UTILIZADORES

| ORGANISMO DE CONTROLO (OC) | USER ID | PASSWORD |
|---|---|---|
| ADMINISTRADOR | admin | admin |
| AGRICERT | agricert | 1234 |
| APCER | apcer | 4502 |
| BEIRA TRADIÇÃO | beiratra | 9975 |
| CERTIPLANET | certiplanet | 6686 |
| CERTIS | certis | 4677 |
| CODIMACO | codimaco | 2078 |
| COMISSÃO TÉCNICA DE CERTIFICAÇÃO E CONTROLO | comtec | 5737 |
| CONFRARIA DO QUEIJO SÃO JORGE | saojorge | 2152 |
| CONTROLVET | controlvet | 8663 |
| ECOCERT PORTUGAL | ecocert | 5499 |
| NATURALFA | naturalfa | 7356 |
| SAGILAB | sagilab | 5612 |
| SATIVA | sativa | 1479 |
| SGS ICS | sgsics | 5896 |
| TRADIÇÃO E QUALIDADE | tradicao | 4303 |
| | | |
| | | |
| | | |
| | | |

**Figura 3.7 - Controlo de Utilizadores**

Para que um utlizador que pertença a dois organismos de controlo consiga mudar rapidamente de utilizador, foi criado um link (figura 19) em todas as páginas que permite a mudança de utilizador através do ecrã de login explicado anteriormente.

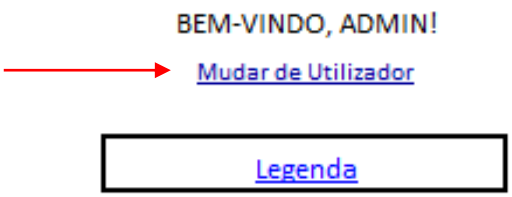


**Figura 3.8 - Mudança de Utilizador**

- ✓ ***Implementação de um formulário inicial para o preenchimento dos campos Regime, Denominação e Responsável. Cada regime tem a sua especificidade em termos de preenchimento, uma vez que para alguns é necessário haver denominação.***

Para este desenvolvimento, utilizou-se a componente de programador do Excel, onde através de código em Visual Basic foi possível criar um formulário como ilustrado na figura 20.

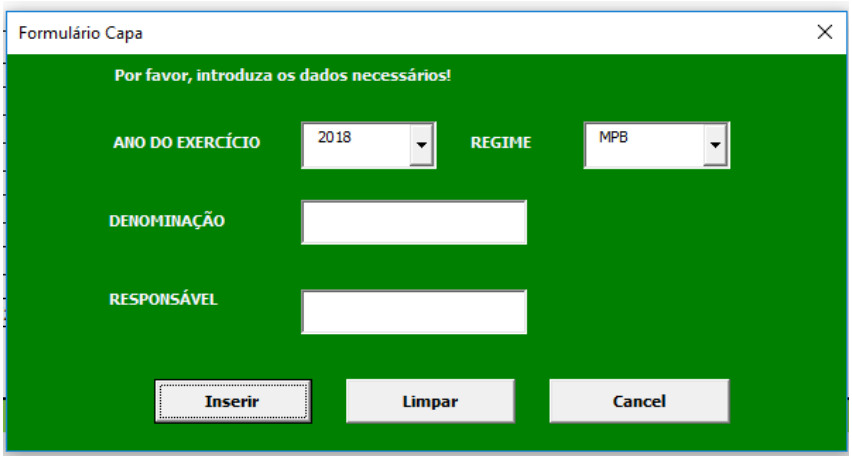


**Figura 3.9 - Formulário de Capa**

Este formulário tem como principais particularidades facilitar o preenchimento do ano do exercício, uma vez que o ano é calculado de forma dinâmica com base no ano atual, permitindo escolher qualquer ano dos últimos cinco. É também possível selecionar qual o regime para o qual se vai realizar o registo de controlo, tendo em atenção que para alguns regimes é necessário preencher a denominação (figura 21). O campo do responsável não é obrigatório sendo por isso um campo de texto livre.

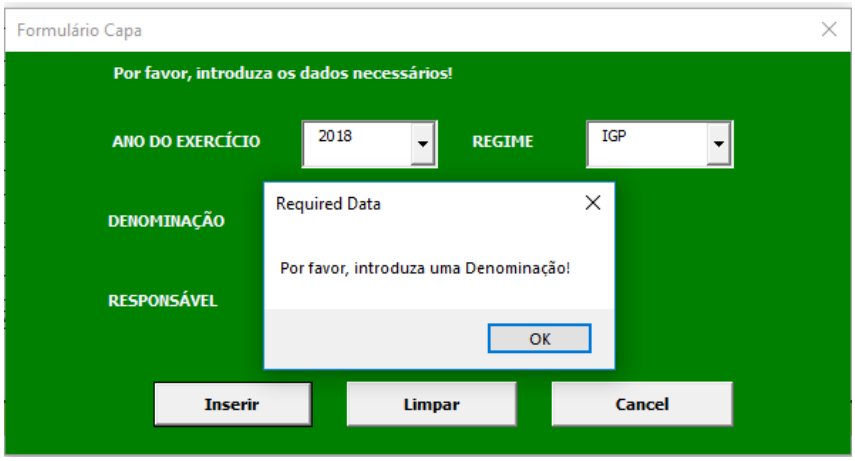


**Figura 3.10 - Introdução Obrigatória da Denominação**

- ✓ ***VALIDAÇÃO DO TIPO DE DADOS INSERIDOS PARA CADA CAMPO, COM DESTAQUE A VERMELHO PARA OS QUE FORAM PREENCHIDOS INCORRETAMENTE.***

Para este desenvolvimento, utilizou-se a componente de programador do Excel, onde através de código em Visual Basic foi possível criar regras de validação como ilustrado na figura 22.


| NIF | NOME | MORADA FISCAL | CONCELHO | E-MAIL | OBSERVAÇÕES |
|---|---|---|---|---|---|
| 999206517 | MARIA ALIC | R. DA MISERICÓR | CARRAZEDA DE ANSIÃES e TORRE DE MONCORVO | XPTO@ABC.DEF | |
| 999428450 | MARIA AMÉL | RUA DR ROLÃO PR | IDANHA A NOVA | XPTO@ABC.DEF | |
| 999235777 | AIDA DOS P | AVELOSO 6430-01 | MEDA | XPTO@ABC.DEF | |
| 999810811 | MARIA ALCI | RUA MANUEL ANTÓ | ALIJÓ e CARRAZEDA DE ANSIÃES | XPTO@ABC.DEF | |
| 999648107 | JOSÉ DOS S | RUA DE SANTO AN | TAROUCA e TRANCOSO | XPTO@ABC.DEF | |
| 999879075 | CARLOS MAN | RUA DE SANTIAGO | CASTELO BRANCO e VILA VELHA DE RÓDÃO | XPTO@ABC.DEF | |
| 999454026 | MARIA VIOL | LARGO MANUEL AN | MIRANDELA e VILA FLOR | XPTO@ABC.DEF | |
| 999817303 | JOÃO CORRE | RUA TRAVESSA DA | IDANHA A NOVA | XPTO@ABC.DEF | |

**Figura 3.11 - Dados Concelho Incorretos**

Para se conseguir criar uma validação simples e objetiva dos campos que compõem as diferentes folhas do ficheiro de Excel, foram criadas varias funções que validam a informação inserida para uma determinada coluna em função do seu tipo. Assim, para as colunas do tipo numérico foi criada uma função que valida se a informação inserida é apenas composta por números e para as colunas do tipo texto foi criada uma função que valida se a informação inserida não é apenas composta por números. Foram, também, desenvolvidas funções para tratar outros tipos de dados, nomeadamente de data e email, assim como foi criada uma função que permite validar se o valor inserido faz ou não parte de uma lista de valores. O código utilizado para a criação das funções pode ser visto em anexo.

Todos os valores que forem inseridos incorretamente aparecerão a vermelho e a coluna será filtrada tendo em conta a cor vermelha, como podemos observar pela figura 22.

Este processo não é visível para o utilizador, uma vez que é executado através do botão de "Validação de Dados" existente em todas as páginas (figura 23).

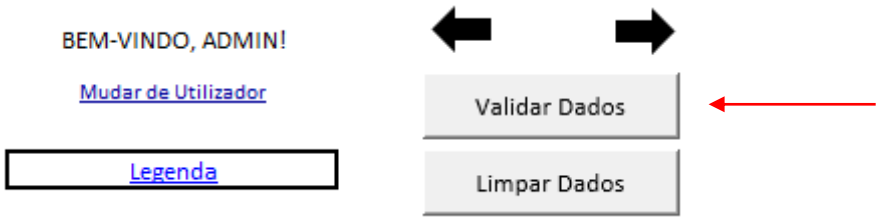


**Figura 3.12 - Validação de Dados**

Após todas as folhas de Excel terem sido preenchidas e validadas, isto é, não existirem campos a vermelho por não estarem de acordo com as regras de preenchimento, o utilizador deve utilizar o botão de "Criar Excel Final" para gerar um novo ficheiro Excel que servirá de fonte ao processo de ETL.

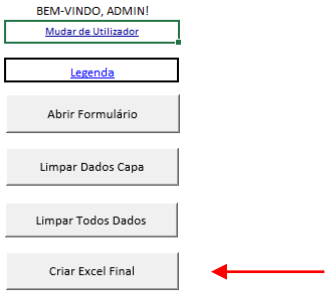


**Figura 3.13 - Criar Excel Final**

- ✓ ***MELHORIA DO ASPETO GRÁFICO E DINAMIZAÇÃO DO PRÓPRIO FICHEIRO.***

Para que o processo de recolha de dados seja simples e intuitivo para o utilizador, o aspeto gráfico do ficheiro de Excel foi modificado, tentando que as cores e a forma como os campos estão construídos tornem mais simples a função do utilizador. Desta forma, as cores utilizadas tentam contrastar com o fundo para facilitar a leitura do conteúdo, assim como a navegação pelas folhas de Excel pretende ser fácil e percetível.

A título de exemplo, a figura 25 e 26 representam a melhoria gráfica das folhas do ficheiro de Excel, uma vez que todas apresentam a mesma estrutura com exceção da capa.

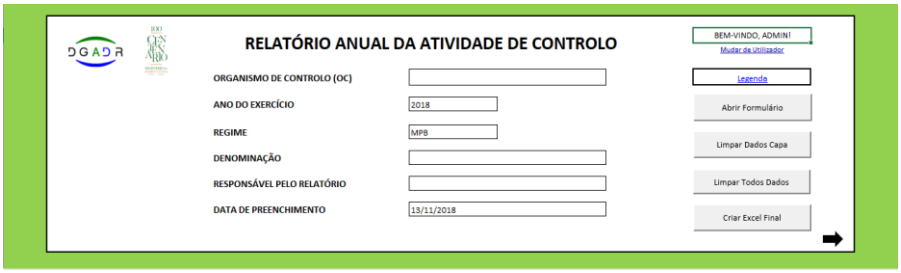


**Figura 3.14 - Estrutura da Folha da Capa**

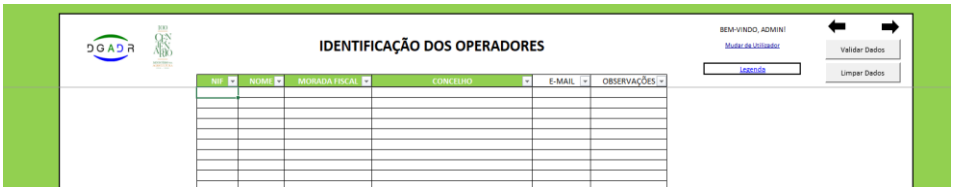


**Figura 3.15 - Estrutura da Folha dos Operadores**

- ✓ ***CRIAÇÃO DE UM FICHEIRO DE ERROS PARA AJUDAR NA IDENTIFICAÇÃO DE PROBLEMAS TÉCNICOS.***

Para ajudar o utilizador quando um erro técnico ocorre ao executar uma ação dentro do ficheiro de Excel foi desenvolvida, através da componente de programador do Excel e utilizando código em Visual Basic, a criação de um ficheiro de log de erros para se conseguir identificar mais facilmente qual pode ser o problema na execução.

A figura 27 ilustra a mensagem de erro que aparece quando um erro é detetado, impedindo que o utilizador aceda diretamente ao código e indicando que deve enviar o ficheiro de erros gerado à equipa técnica de suporte.

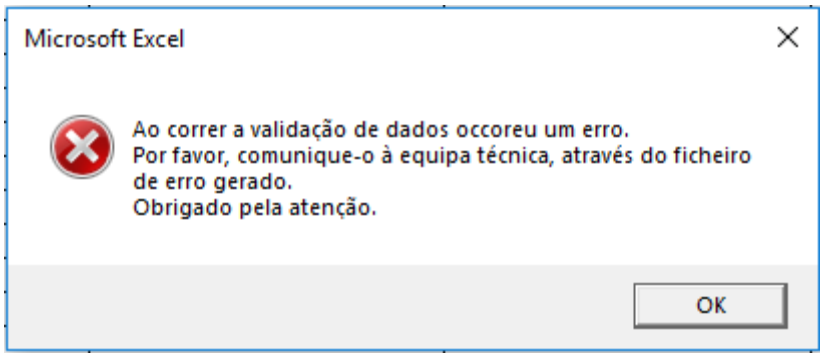


**Figura 3.16 - Mensagem de Erro Técnico**

## 3.3 PROCESSO DE INTEGRAÇÃO

Concluído o processo de recolha de dados, surge a necessidade de armazenar a informação recolhida e realizar análises sobre a mesma, de forma a gerar informação que suporte a tomada de decisão. Assim, e devido à inexistência de um local de armazenamento de dados que facilite o tratamento e análise dos mesmos, foi proposto o desenvolvimento de uma base de dados em SQL que armazene a informação recolhida pelo ficheiro de Excel anteriormente referido, assim como o desenvolvimento de um processo de ETL que carregue os dados para um *Data Warehouse*.

Com o presente projeto, propôs-se desenvolver para a componente de integração de dados, os seguintes pontos:

- ✓ ***DESENVOLVIMENTO DE UM MODELO DE BASE DE DADOS DESNORMALIZADO, CENTRADO NO OPERADOR, COM AS DIMENSÕES QUE O COMPLEMENTAM.***

Para o desenvolvimento desta componente, começou por se criar um modelo relacional com o desenho de todas as tabelas e respetivos atributos. Deste modo, foram desenvolvidas 14 tabelas, onde cada uma corresponde a uma folha do ficheiro de Excel fonte. Os diagramas desenvolvidos para a construção da *Staging Area* e do *Data Warehouse* podem ser vistos em anexo.

A escolha do tipo de modelo multidimensional deve ter em conta a complexidade e a redundância dos diferentes tipos de modelo. Para o presente projeto, o modelo que mais se ajusta às necessidades de negócio é a arquitetura em estrela (*Star Schema*). Este modelo abrange a complexidade desejada e, através da divisão do modelo dimensional em fatos e dimensões, permite uma análise detalhada sobre o objetivo de negócio.

O modelo foi desenvolvido tendo por base uma tabela de fatos (Operador) e as tabelas de dimensões que a suportam. Cada tabela de dimensão liga-se à tabela de fatos pela mesma chave OLTP – NIF e Organismo de Controlo. Contudo para as transações efetuadas durante o processo de ETL são utilizadas as *surrogate keys*, isto é, uma chave numérica que é utilizada para melhorar a performance da relação entre as tabelas.

Para a construção do modelo também foi tida em conta a forma como se lidam com os dados operacionais, isto é, apesar de os mesmos não serem possíveis de visualizar a nível relacional, são possíveis de obter a nível analítico tendo em conta questões chave do negócio como quantos são os operadores do concelho de Lisboa que são produtores.

- ✓ ***CONSTRUÇÃO DE UM PROCESSO DE ETL QUE CARREGUE A INFORMAÇÃO RECOLHIDA PELO PROCESSO DE RECOLHA DE DADOS PARA UMA STAGING AREA E PARA UM DATA WAREHOUSE.***

Para a construção do processo de ETL foram desenvolvidas duas componentes principais: a *Staging Area* (SA) e o *Data Warehouse* (DW). Todo o processo de ETL foi desenvolvido com recurso à tecnologia da Microsoft, *SQL Server Integration Services*. Os diagramas desenvolvidos para o processo de ETL da *Staging Area* e do *Data Warehouse* podem ser vistos em anexo.

| FONTE EXCEL | TABELA SA | TABELA DW |
|---|---|---|
| **2 - OPERADOR** | STG_OPERADOR | OPERADOR |
| **3 - CONTRATO** | STG_CONTRATO | CONTRATO |
| **4 - CERTIFICADO** | STG_CERTIFICADO | CERTIFICADO |
| **5 - CONTROLOS** | STG_CONTROLOS | CONTROLOS |
| **6 - AREA_VEGETAL** | STG_AREA_VEGETAL | AREA_VEGETAL |
| **7 - PROD_VEGETAL** | STG_PROD_VEGETAL | PROD_VEGETAL |
| **8 - EF_ANIMAL** | STG_EF_ANIMAL | EF_ANIMAL |
| **9 - PROD_ANIMAL** | STG_PROD_ANIMAL | PROD_ANIMAL |
| **10 - PREPARACAO** | STG_PREPARACAO | PREPARACAO |
| **11 - DISTRIBUICAO** | STG_DISTRIBUICAO | DISTRIBUICAO |
| **12 - IMPORTACAO** | STG_IMPORTCAO | IMPORTACAO |
| **13 - EXPORTACAO** | STG_EXPORTCAO | EXPORTACAO |
| **14 - ANALISES** | STG_ANALISES | ANALISES |
| **15 - OCORRENCIAS** | STG_OCORRENCIAS | OCORRENCIAS |

**Tabela 2 - Mapeamento entre fonte e destino**

### 3.3.1 *STAGING AREA*

As tabelas de dimensões vão descrever os objetos que pertencem à tabela de fatos. No presente projeto, tentou manter-se uma certa coerência na importação dos dados, de forma a melhorar a integridade dos mesmos, nunca pondo de lado a performance com que os mesmos são carregados. Deste modo, as dimensões apresentam o mesmo conteúdo dentro de cada *Sequence Container*, onde podemos destacar:

- O *delete* que é feito inicialmente às tabelas para que elas possuam apenas dados "limpos", sem duplicados, garantindo maior integridade dos mesmos;

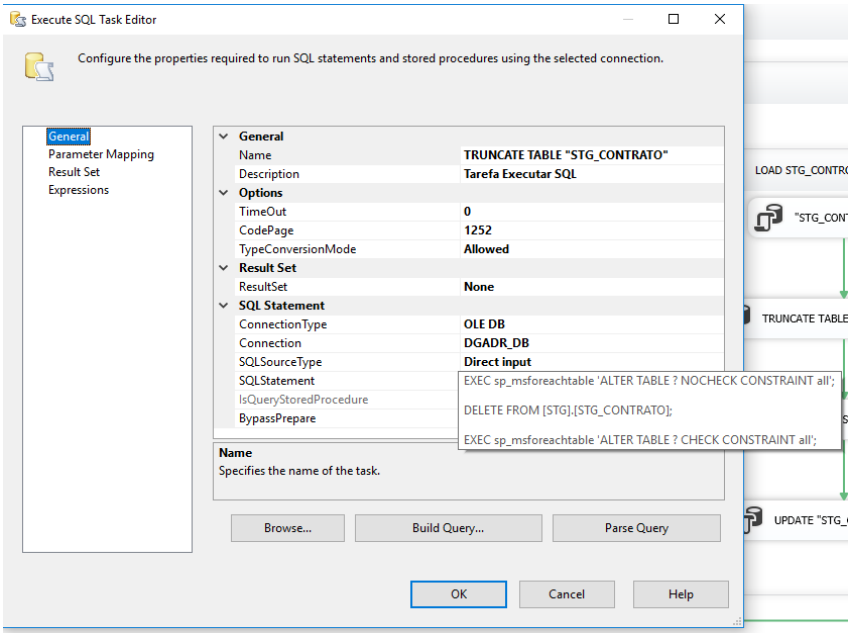


**Figura 3.17 - Tarefa de SQL para apagar dados**

- A execução da *stored procedure* para se realizar a auditoria do carregamento e de execução de uma determinada tarefa. É preciso atualizar a tabela de auditoria com o tempo de execução final, sendo por isso que existem dois *Execute SQL Tasks*, um que lança a *stored procedure* inicialmente e outro que atualiza a tabela com o tempo final, duração e número de linhas inseridas ou filtradas.

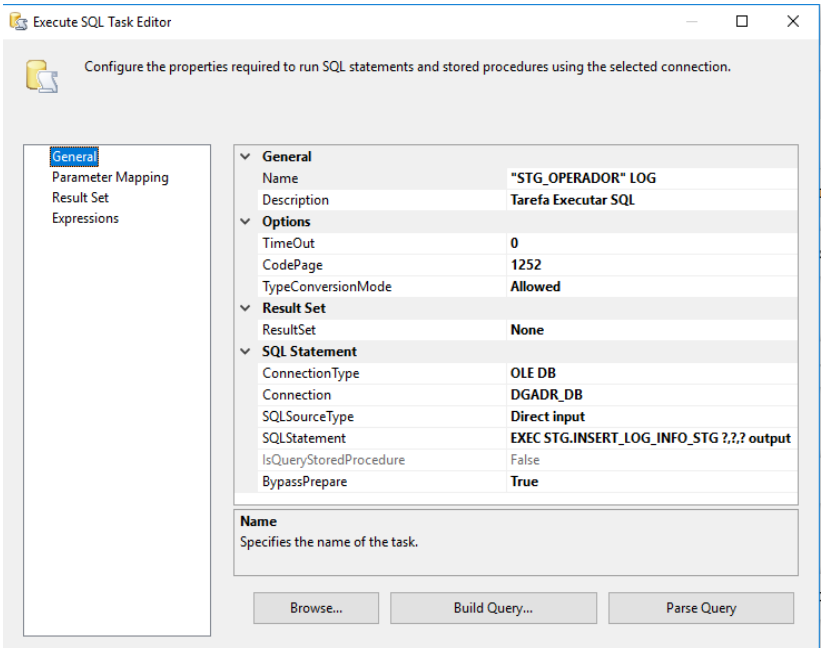


**Figura 3.18 - Tarefa de SQL para inserir linha na tabela de auditória**

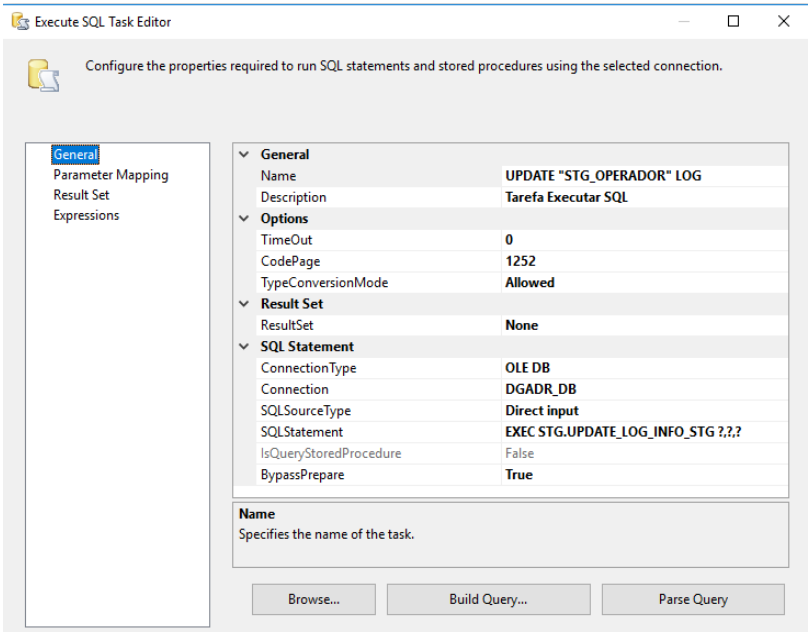


**Figura 3.19 - Tarefa de SQL para atualizar a tabela de auditória**

- O *Data Flow* de carregamento dos dados que faz a importação dos dados de uma fonte (*SRC*) para um destino (*TGT*);

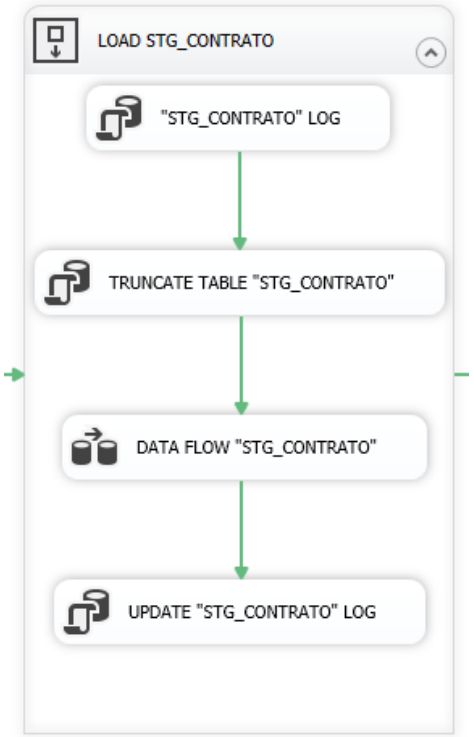


**Figura 3.20 - *Data Flow Staging Area***

Relativamente ao *Data Flow* destacam-se os seguintes pontos:

- A importação dos dados por via de uma fonte (*SRC*), que é feita de forma direta através do carregamento de dados das folhas de Excel do ficheiro fonte;
- A verificação de preenchimento dos campos obrigatórios através de uma transformação de *Conditional Split*, isto é, todos os campos que são obrigatórios devem vir preenchidos para se evitarem erros de violação da chave primária.
- Um *Row Count* que permite guardar o número de linhas que vão ser inseridas na tabela de destino e o *Row Count Filter* que permite guardar o número de linha filtradas, ou seja, o número de linhas cujo os campos obrigatórios não vêm preenchidos. Ambos os valores de número de linhas são guardados em variáveis criadas previamente para o efeito;
- A integração dos dados numa tabela de destino (*TGT*), através do mapeamento dos campos.

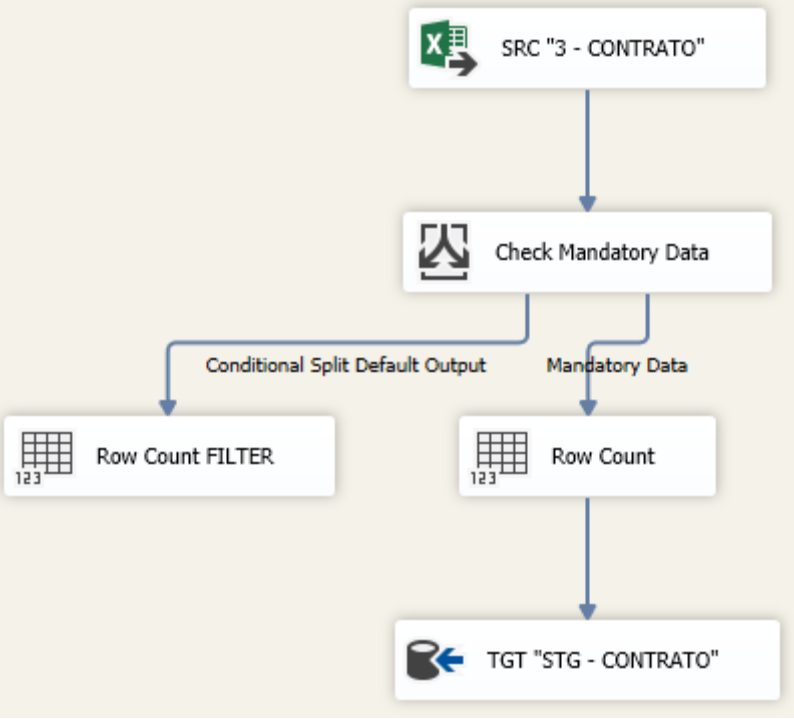


**Figura 3.21 - Processo de carregamento da *Staging Area***

**NOTA:** O processo de carregamento de dados nas dimensões da *Staging Area* é feito de uma forma semelhante, sendo por isso que se referenciou como é que é realizado o processo de uma forma geral e não de uma forma particular.

Tipicamente, as tabelas de fatos guardam a informação quantitativa (métricas) que vão depois ser descritas pelas dimensões. Contudo, no presente projeto, a tabela considerada como fato (Operador) não possui métricas uma vez que aquilo que queremos compreender é atividade dos vários operadores e aquilo que os descreve.

Neste projeto, a tabela de fatos apresenta um conteúdo semelhante às tabelas de dimensões dentro do *Sequence Container*, uma vez que não é de carácter quantitativo, mas sim mais descritivo:

- O *delete* que é feito inicialmente às tabelas para que elas possuam apenas dados "limpos", sem duplicados, garantindo maior integridade dos dados;

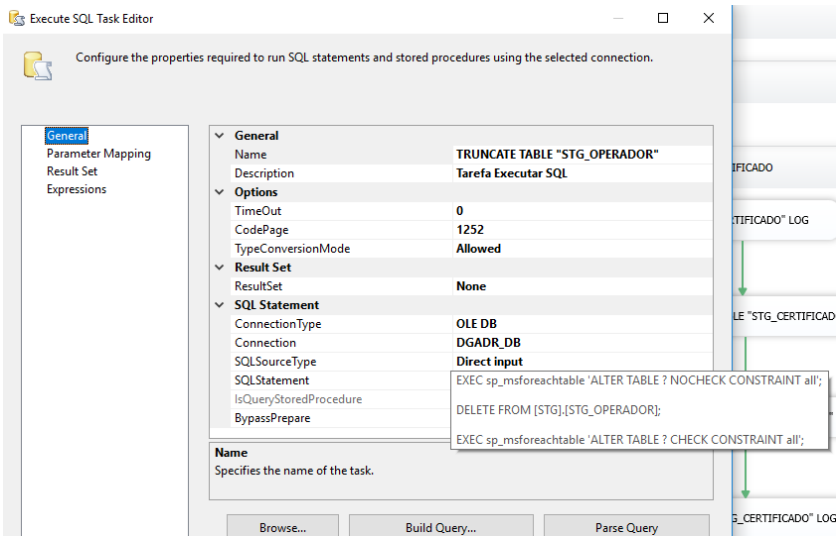


**Figura 3.22 - Tarefa de SQL para apagar dados (Operador)**

- A execução da *Stored Procedure* para se realizar a auditoria do carregamento e de execução de uma determinada tarefa. É preciso atualizar a tabela de auditoria, com o tempo de execução final, sendo por isso que existem dois *Execute SQL Tasks*, um que lança a *Stored Procedure* inicialmente e outro que atualiza a tabela com o tempo final, duração e número de linhas inseridas.

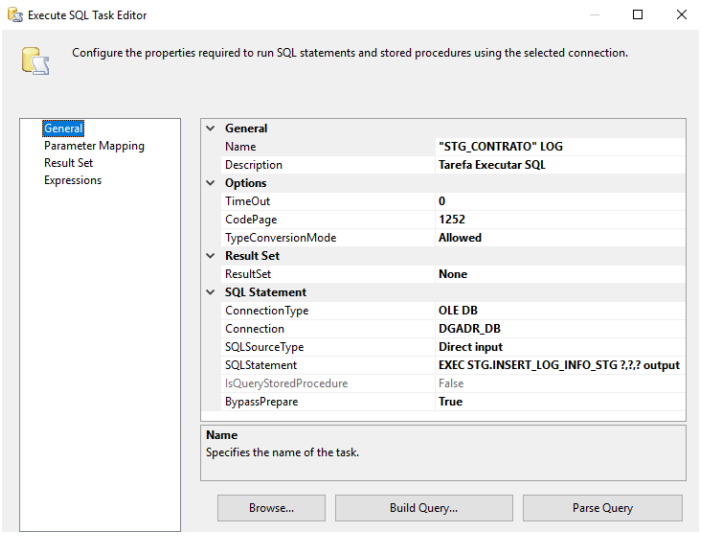


**Figura 3.23 - Tarefa de SQL para inserir linha na tabela de auditória da *Staging Area* (Operador)**

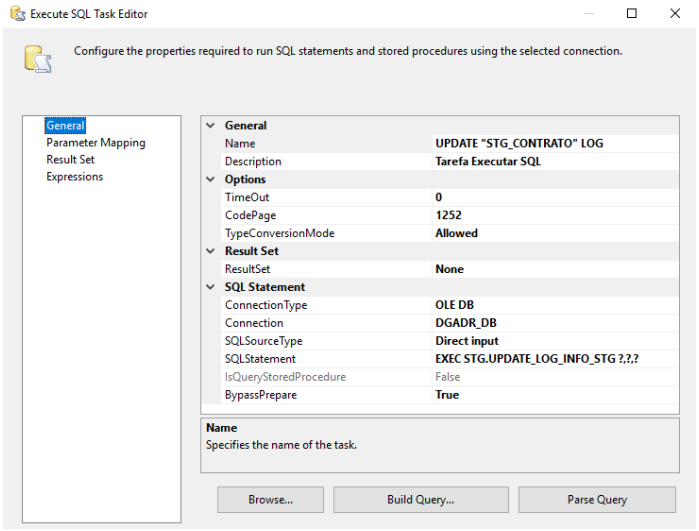


**Figura 3.24 - Tarefa de SQL para atualizar a tabela de auditória da *Staging Area* (Operador)**

- O *Data Flow* de carregamento dos dados que faz a importação dos dados de uma fonte (*SRC*) para um destino (*TGT*);

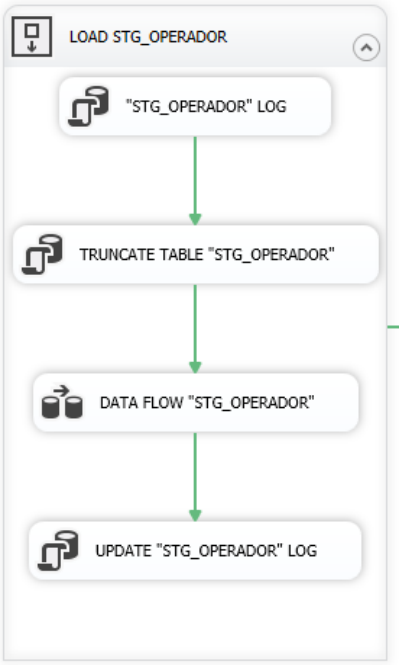


**Figura 3.25 - *Data Flow Staging Area* (Operador)**

Relativamente ao *Data Flow* destacam-se os seguintes pontos:

- A importação dos dados por via de uma fonte (*SRC*), que é feita de forma direta através do carregamento de dados das folhas de Excel do ficheiro fonte;
- A verificação de preenchimento dos campos obrigatórios através de uma transformação de *Conditional Split*, isto é, todos os campos que são obrigatórios devem vir preenchidos para se evitarem erros de violação da chave primária.
- *Join* com todas as tabelas de dimensão para o carregamento das *surrogate keys*
- Um *Row Count* que permite guardar o número de linhas que vão ser inseridas na tabela de destino e o *Row Count Filter* que permite guardar o número de linha filtradas, ou seja, o número de linhas cujo os campos obrigatórios não vêm preenchidos. Ambos os valores de número de linhas são guardados em variáveis criadas previamente para o efeito;
- A integração dos dados numa tabela de destino (*TGT*), através do mapeamento dos campos.

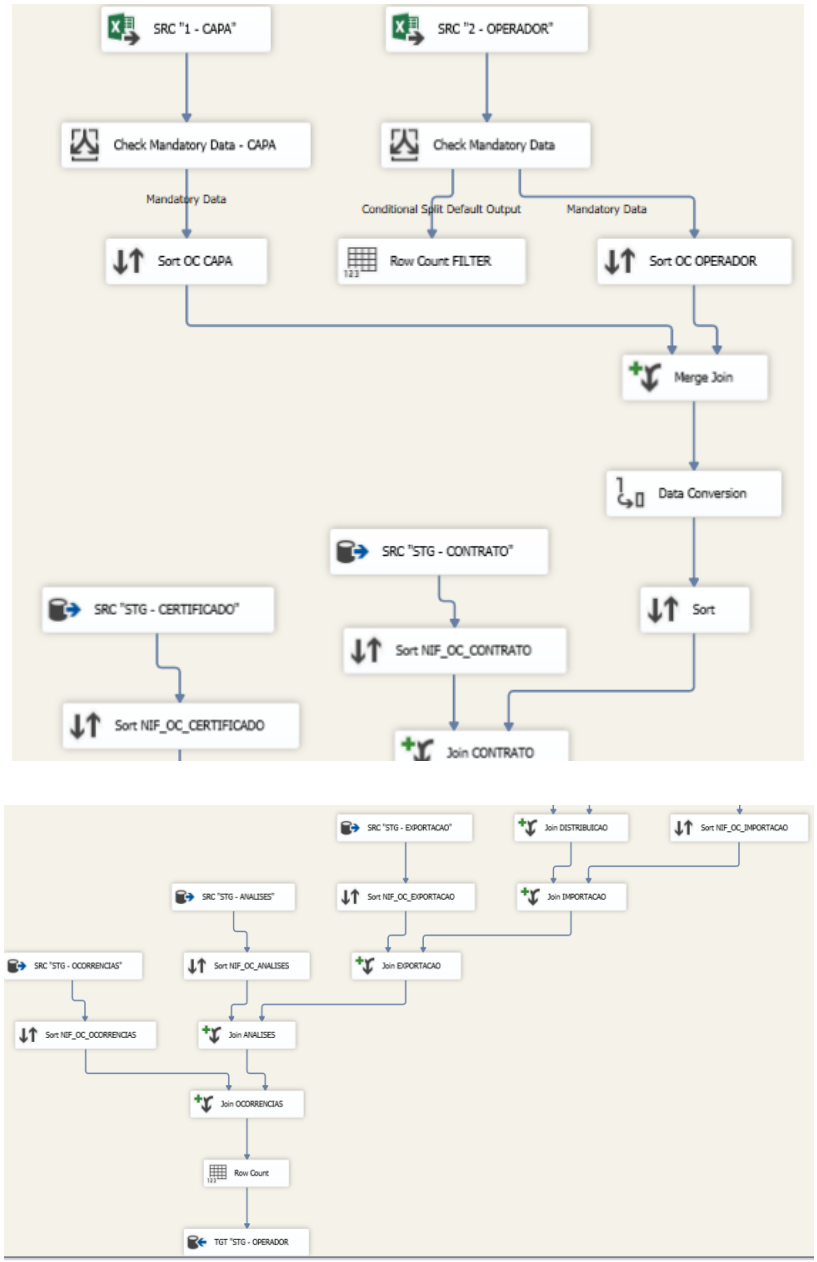


**Figura 3.26 - Processo de carregamento da *Staging Area* (Operador)**

Para que o processo seja dinâmico, incluiu-se a possibilidade de processar múltiplos ficheiros de Excel que venham dos diferentes organismos de controlo ao mesmo tempo. Contudo, é necessário que contenham a máscara (nome de ficheiro) que foi previamente definida e que cumpram os requisitos do ponto de vista de metadados, isto é, garantir que todos os ficheiros possuem o mesmo número de folhas e o mesmo número de atributos.

De forma a que se consiga controlar os ficheiros que já foram processados, no final do processo, os mesmos são armazenados numa pasta de arquivo, sendo-lhes atribuído um ID para identificação da ordem pelo qual foram carregados.


**Figura 3.27 - Transformações utilizadas para arquivar os ficheiros**

### 3.3.2 *DATA WAREHOUSE*

À semelhança do que acontece na *Staging Area*, as tabelas de dimensões vão descrever os objetos que pertencem à tabela de fatos. Tentou manter-se a coerência na importação dos dados, de forma a melhorar a integridade dos mesmos, nunca pondo de lado a performance com que os mesmos são carregados.

As dimensões apresentam o mesmo conteúdo dentro de cada *Sequence Container*, onde se pode destacar:

- Não é feito nenhum *delete* às tabelas, visto que os dados apenas são atualizados ou adicionados, de forma a criar histórico, mas nunca apagados das mesmas;
- A execução da *Stored Procedure* para se realizar a auditoria do carregamento e de execução de uma determinada tarefa. É preciso atualizar a tabela de auditoria, com o tempo de execução final, sendo por isso que existem dois *Execute SQL Tasks*, um que lança a *stored procedure* inicialmente e outro que atualiza a tabela com o tempo final, duração e número de linhas inseridas ou filtradas;

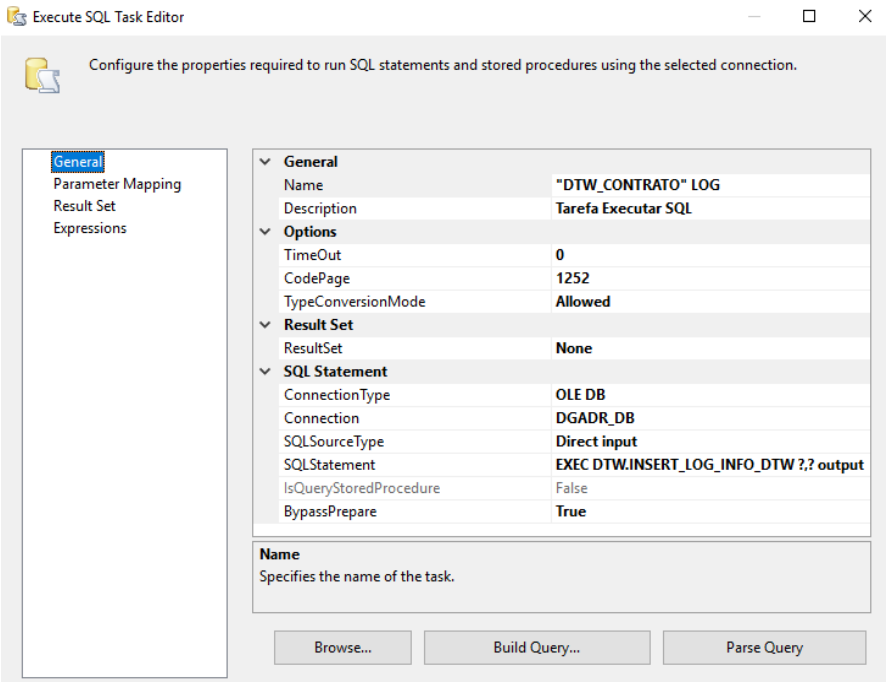


**Figura 3.28 - Tarefa de SQL para inserir linha na tabela de auditória do *Data Warehouse***

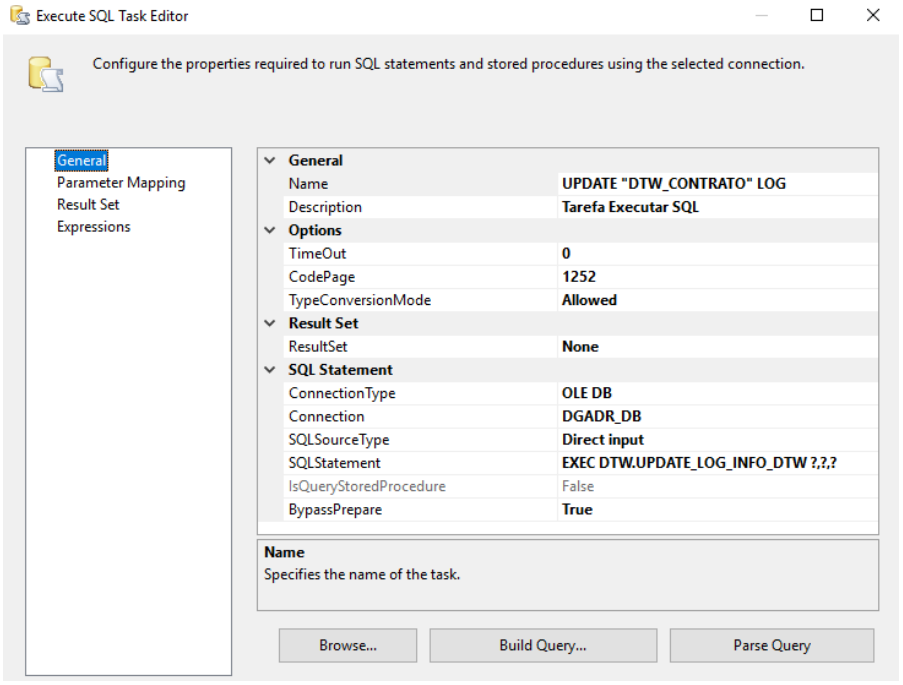


**Figura 3.29 - Tarefa de SQL para atualizar a tabela de auditória do *Data Warehouse***

- O *Data Flow* de carregamento dos dados que faz a importação dos dados de uma fonte (*SRC*) para um destino (*TGT*).

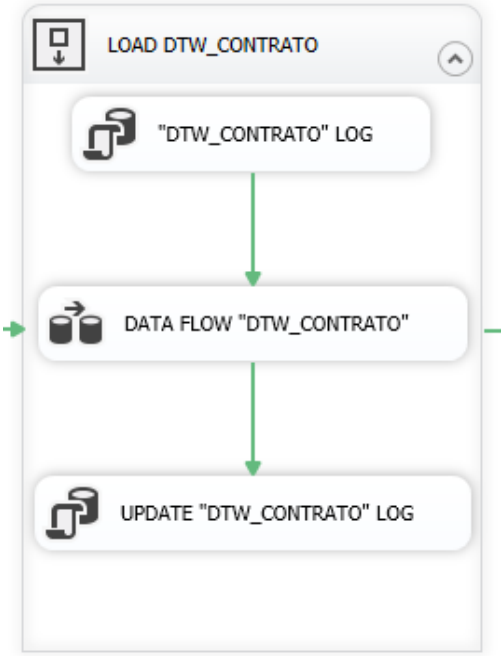


**Figura 3.30 - *Data Flow Data Warehouse***

Relativamente ao *Data Flow* destacam-se os seguintes pontos:

- A importação dos dados por via de uma fonte (*SRC*), que pode ser feita de forma direta através de um *Table Load*;
- A criação de um código MD5 (código de 32 caracteres) para verificação se o registo já foi ou não inserido na tabela de destino;
- Caso o valor do MD5 não exista, é feita a validação do NIF e Organismo de Controlo de forma a compreender se é um registo novo ou se é uma atualização de um já existente;
- Criação de campos de auditoria para compreender quando é que um registo foi inserido (*INSERT_DATE*), quando é que foi atualizado (*UPDATE_DATE*) e se é a versão mais atualizada (*IS_LAST_VERSION*);
- Um *Row Count* que permite guardar o número de linhas que vão ser inseridas na tabela de destino ou o número de linhas que foram atualizadas. O número de linhas é guardado numa variável criada previamente para o efeito;
- A integração dos dados numa tabela de destino (*TGT*), através do mapeamento dos campos;

- A utilização de comandos em SQL para se realizarem *queries* que permitam atualizar os campos necessários.

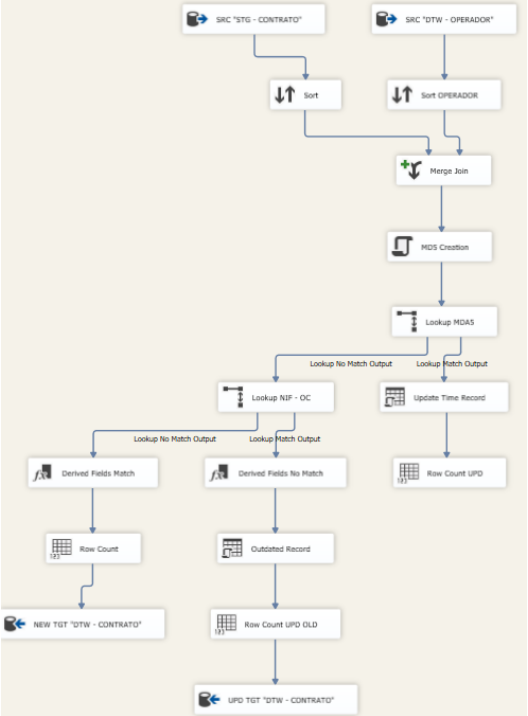


**Figura 3.31 - Processo de carregamento do *Data Warehouse***

**NOTA:** À semelhança da *Staging Area*, o processo de carregamento de dados nas dimensões do *Data Warehouse* é feito de uma forma semelhante, sendo por isso que se referenciou como é que é realizado o processo de uma forma geral e não de uma forma particular.

Neste projeto, a tabela de fatos apresenta um conteúdo semelhante às tabelas de dimensões dentro do *Sequence Container*, uma vez que não é de carácter quantitativo, mas sim mais descritivo:

- Não é feito nenhum *delete* às tabelas, visto que os dados apenas são atualizados ou adicionados à tabela de forma a criar histórico, mas nunca apagados da mesma;

- A execução da *Stored Procedure* para se realizar a auditoria do carregamento e de execução de uma determinada tarefa. É preciso atualizar a tabela de auditoria, com o tempo de execução final, sendo por isso que existem dois *Execute SQL Tasks*, um que lança a *Stored Procedure* inicialmente e outro que atualiza a tabela com o tempo final, duração e número de linhas inseridas ou filtradas;

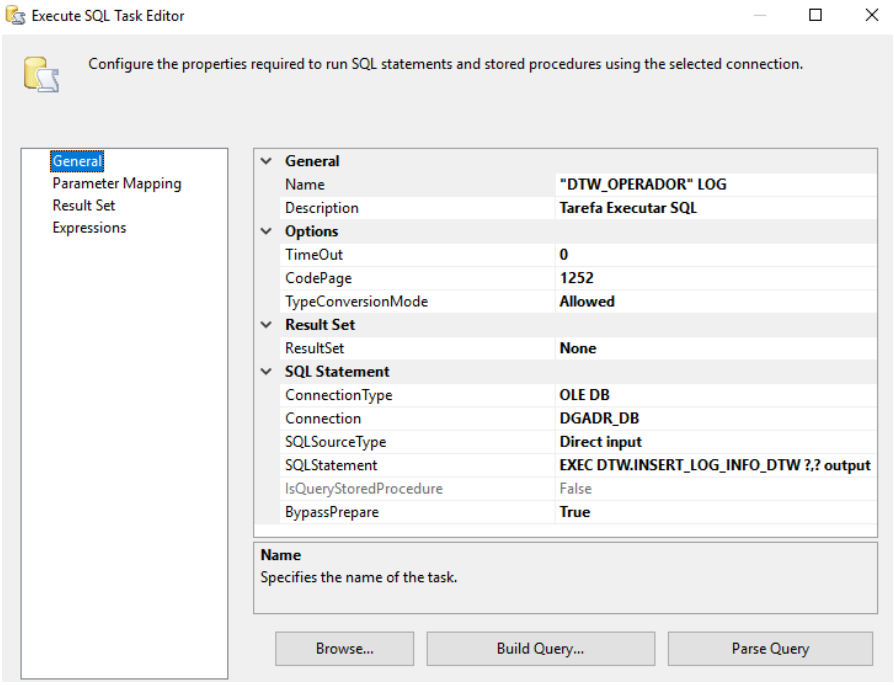


**Figura 3.32 - Tarefa de SQL para inserir linha na tabela de auditória do *Data Warehouse* (Operador)**

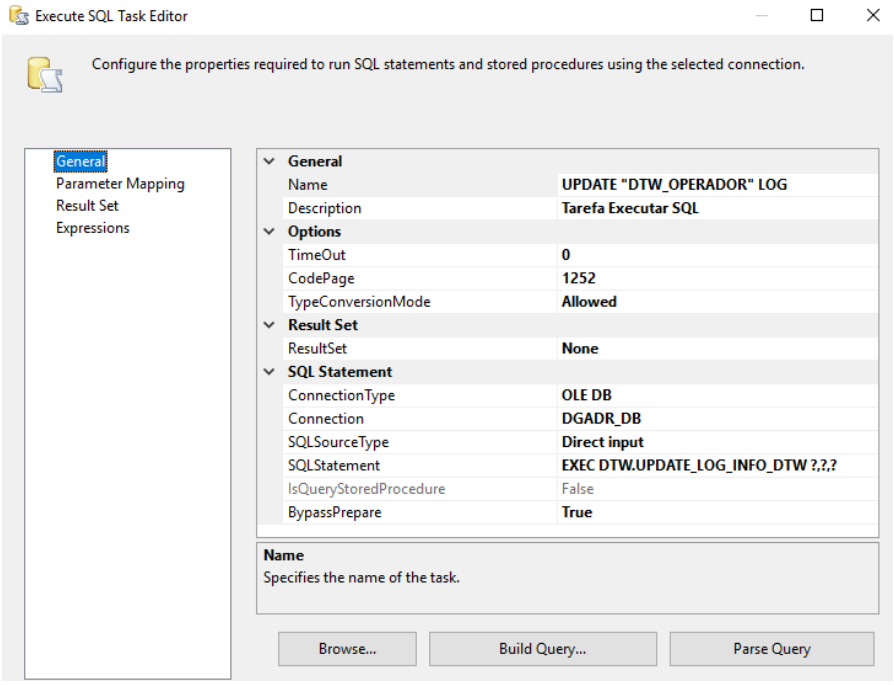


**Figura 3.33 - Tarefa de SQL para atualizar a tabela de auditória do *Data Warehouse* (Operador)**

- O *Data Flow* de carregamento dos dados que faz a importação dos dados de uma fonte (SRC) para um destino (DTN);

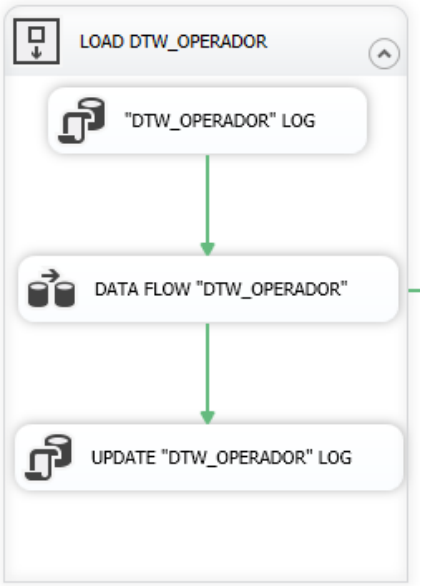


**Figura 3.34 - *Data Flow Data Warehouse* (Operador)**

Em relação ao *Data Flow* propiamente dito, podemos destacar:

- A importação dos dados por via de uma fonte (*SRC*), que pode ser feita de forma direta através de um *Table Load*;
- *Join* com todas as tabelas de dimensão para o carregamento das *surrogate keys*. Apenas são selecionados os registos das dimensões da última versão seja igual 1 (IS_LAST_VERSION = 1).
- A criação de um código MD5 (código de 32 caracteres) para verificação se o registo já foi ou não inserido na tabela de destino;
- Caso o valor do MD5 não exista, é feita a validação do NIF e Organismo de Controlo de forma a compreender se é um registo novo ou se é uma atualização a um já existente.
- Criação de campos de auditoria para compreender quando é que um registo foi inserido (*INSERT_DATE*), quando é que foi atualizado (*UPDATE_DATE*) e se é a versão mais atualizada (*IS_LAST_VERSION*);
- Um *Row Count* que permite guardar o número de linhas que vão ser inseridas na tabela de destino ou o número de linhas que foram atualizadas. O número de linhas é guardado numa variável criada previamente para o efeito;
- A integração dos dados numa tabela de destino (*TGT*), através do mapeamento dos campos;

- A utilização de comandos em SQL para se realizarem *queries* que permitam atualizar os campos necessários.

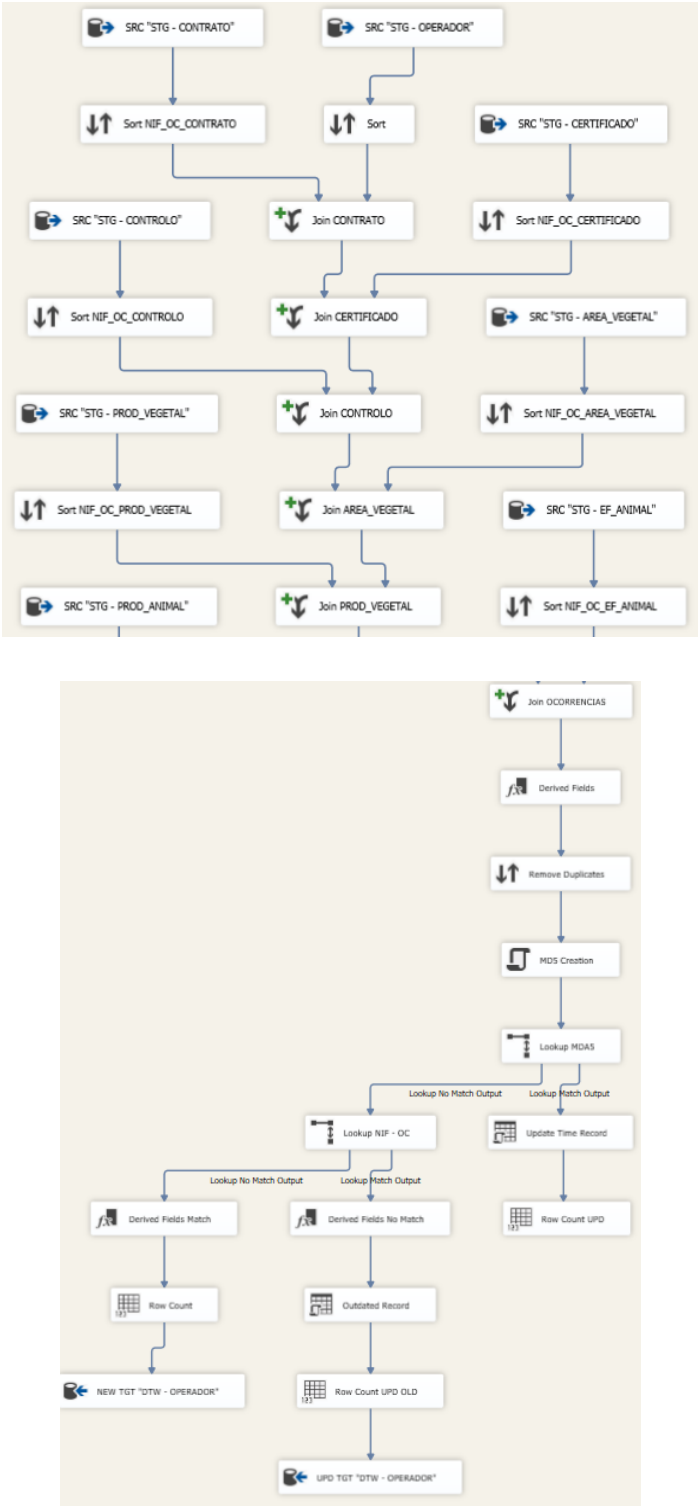


**Figura 3.35 - Processo de carregamento do *Data Warehouse* (Operador)**

- ✓ ***CRIAÇÃO DE UMA TABELA DE LOG QUE PERMITA COMPREENDER A PERFORMANCE DO PRÓPRIO SISTEMA E O NÚMERO DE DADOS RECOLHIDOS PARA CADA EXECUÇÃO***.

Foi criada uma tabela de auditoria, tanto para a *Staging Area* como para o *Data Warehouse*, que guarda informação sobre o número de registos inseridos, quando se inicia o processo e quando termina o processo para uma tarefa em particular, fazendo referência também à duração, expressa em segundos, que o processo demorou. Esta tabela foi carregada com recurso a uma *stored procedure*, para que o processo seja dinâmico, isto é, sempre que uma tarefa é iniciada no processo de ETL é adicionado um novo registo com a informação previamente mencionada.

| | MASTER_ID | BATCH_ID | TASK_NAME | FILE_NAME | RECORD_COUNT_INSERT | RECORD_COUNT_FILTER | START_TIME | END_TIME | DURATION |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | "STG_OPERADOR" LOG | Relatorio_2017_sativa_20181113.xlsx | 8 | 0 | 2018-11-13 16:08:57.400 | 2018-11-13 16:08:58.983 | 1.583 |
| 2 | 2 | 1 | "STG_CONTRATO" LOG | Relatorio_2017_sativa_20181113.xlsx | 0 | 52 | 2018-11-13 16:08:59.040 | 2018-11-13 16:09:00.010 | 0.970 |
| 3 | 3 | 1 | "STG_CERTIFICADO" LOG | Relatorio_2017_sativa_20181113.xlsx | 0 | 0 | 2018-11-13 16:09:00.077 | 2018-11-13 16:09:01.020 | 0.944 |
| 4 | 4 | 1 | "STG_CONTROLO" LOG | Relatorio_2017_sativa_20181113.xlsx | 8 | 0 | 2018-11-13 16:09:01.070 | 2018-11-13 16:09:15.053 | 13.983 |
| 5 | 5 | 1 | "STG_AREA_VEGETAL" LOG | Relatorio_2017_sativa_20181113.xlsx | 0 | 125 | 2018-11-13 16:09:15.127 | 2018-11-13 16:09:16.353 | 1.227 |
| 6 | 6 | 1 | "STG_PROD_VEGETAL" LOG | Relatorio_2017_sativa_20181113.xlsx | 0 | 2 | 2018-11-13 16:09:16.403 | 2018-11-13 16:09:17.393 | 0.990 |
| 7 | 7 | 1 | "STG_EF_ANIMAL" LOG | Relatorio_2017_sativa_20181113.xlsx | 0 | 27 | 2018-11-13 16:09:17.440 | 2018-11-13 16:09:18.480 | 1.040 |
| 8 | 8 | 1 | "STG_PROD_ANIMAL" LOG | Relatorio_2017_sativa_20181113.xlsx | 0 | 0 | 2018-11-13 16:09:18.530 | 2018-11-13 16:09:19.470 | 0.940 |
| 9 | 9 | 1 | "STG_PREPARACAO" LOG | Relatorio_2017_sativa_20181113.xlsx | 0 | 0 | 2018-11-13 16:09:19.520 | 2018-11-13 16:09:20.497 | 0.976 |
| 10 | 10 | 1 | "STG_DISTRIBUICAO" LOG | Relatorio_2017_sativa_20181113.xlsx | 0 | 0 | 2018-11-13 16:09:20.547 | 2018-11-13 16:09:21.470 | 0.924 |
| 11 | 11 | 1 | "STG_IMPORTACAO" LOG | Relatorio_2017_sativa_20181113.xlsx | 0 | 0 | 2018-11-13 16:09:21.520 | 2018-11-13 16:09:22.447 | 0.926 |
| 12 | 12 | 1 | "STG_EXPORTACAO" LOG | Relatorio_2017_sativa_20181113.xlsx | 0 | 0 | 2018-11-13 16:09:22.493 | 2018-11-13 16:09:23.413 | 0.920 |
| 13 | 13 | 1 | "STG_ANALISES" LOG | Relatorio_2017_sativa_20181113.xlsx | 0 | 1 | 2018-11-13 16:09:23.463 | 2018-11-13 16:09:24.437 | 0.973 |
| 14 | 14 | 1 | "STG_OCORRENCIAS" LOG | Relatorio_2017_sativa_20181113.xlsx | 0 | 7 | 2018-11-13 16:09:24.483 | 2018-11-13 16:09:25.500 | 1.017 |

**Figura 3.36 - Tabela de Auditoria**

- ✓ ***CRIAÇÃO DE UMA TABELA DE ERROS QUE PERMITA MONITORIZAR QUAL FOI O ERRO QUE OCORREU E QUE LEVOU A QUE O PROCESSO NÃO TERMINASSE COM SUCESSO.***

Foi criada uma tabela de erros, cujo principal objetivo é guardar a informação que esteja relacionada com os erros que possam ocorrer durante o processo de ETL, quer estejamos a carregar a *Staging Area* ou o *Data Warehouse*. Esta tabela, à semelhança da tabela de auditoria, tem por base uma *stored procedure* que atribui dinamismo ao processo, isto é, quando existe um erro durante o processo, a identificação do mesmo, assim como um ID e a tarefa onde esse erro ocorreu são guardados na tabela de erros.

| | MASTER_ID | BATCH_ID | TASK_NAME | ERROR |
|---|---|---|---|---|
| 1 | 3 | 6 | ERROR "DTW_IMPORTACAO" TABLE | Executing the query "EXEC DTW.INSERTT_LOG_INFO_DTW ?,? output" failed with the following error: "Could not find stored procedure 'DTW.INSERTT_LOG_INFO_DT... |

**Figura 3.37 - Tabela de Erros**

## 3.4 PROCESSO DE VISUALIZAÇÃO DE DADOS

Após o tratamento dos dados é necessário disponibilizar a informação de uma forma clara com o propósito de aumentar o conhecimento sobre os dados em estudo e, consequentemente, suporte à tomada de decisão. Deste modo, propõe-se desenvolver para a componente de visualização de dados, os seguintes pontos:

- ✓ ***CRIAÇÃO DE DASHBOARDS QUE DISPONIBILIZEM A INFORMAÇÃO EM DIFERENTES COMPONENTES ATRAVÉS DA FERRAMENTA POWER BI.***

Power BI é uma ferramenta de análise de negócio que pertence à Microsoft. Esta ferramenta permite uma visualização interativa onde os utilizadores podem criar os seus próprios *reports* e *dashboards* sem dependerem de qualquer tipo de administração de bases de dados ou de departamento de IT. É uma solução que possibilita aos utilizadores ter uma perspetiva diferente dos dados, uma vez que se consegue aceder aos mesmos de forma rápida e dinâmica. Assim, conseguem-se retirar conclusões destes, atribuindo uma vantagem competitiva à organização.

Para esta componente foram criados três *dashboards* com informação sobre os operadores, a produção vegetal e a produção animal. É possível filtrar pelo ano de exercício, pelo organismo de controlo e por concelho de forma a obtermos diferentes resultados consoante o objetivo da análise.

Na figura 49 é possível visualizar o *dashboard* dos operadores que tem informação sobre o número de operadores ativos, assim como a distribuição de operadores pelo tipo de atividade ou por concelho. Desta forma, observa-se que o concelho onde existe um maior número de operadores ativos é o concelho de Vila Flor onde o número de operadores por atividade por atividade é semelhante.

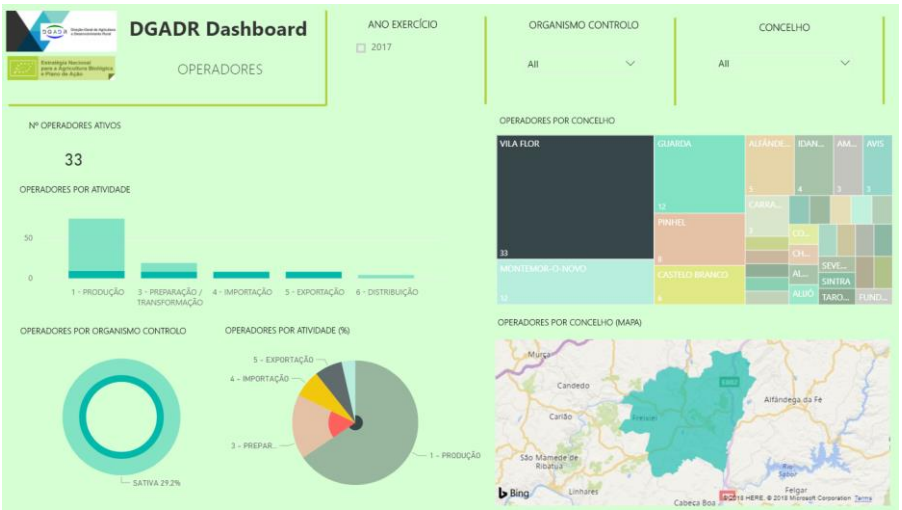


**Figura 3.38 - *Dashboard* Operadores**

Na figura 50 é possível visualizar o *dashboard* sobre a produção vegetal que tem informação sobre o total de área de produção biológica, assim como o número de operadores por cultura. Devido à reduzida amostra de dados apenas foi possível mostrar a produção vegetal para dois operadores, que produzem pequenos frutos e citrinos nos concelhos de Sintra e Almada.

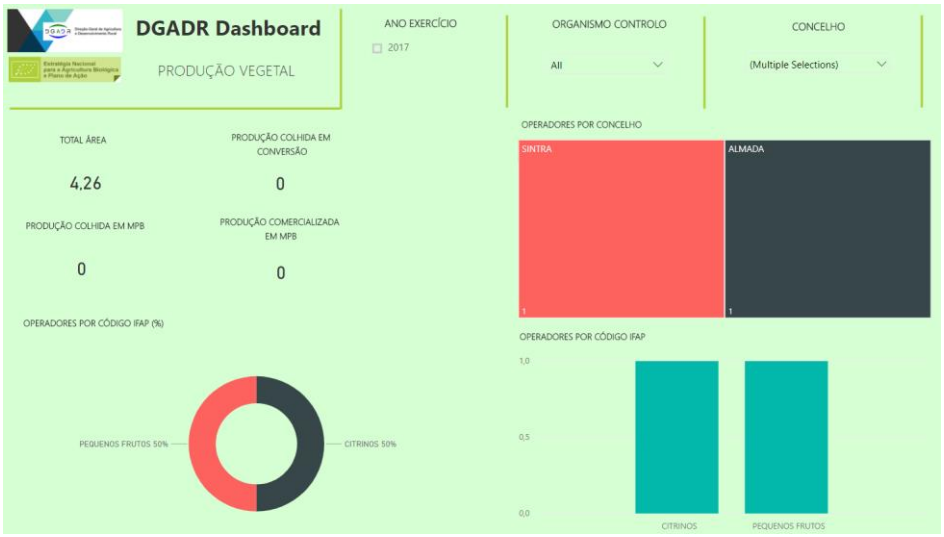


**Figura 3.39 - *Dashboard* Produção Vegetal**

Na figura 51 é possível observar o *dashboard* sobre a produção animal que tem informação sobre o número de animais produzidos em conversão ou em modo de produção biológica (MPB). Para além disso, consegue-se verificar a distribuição de operadores por tipo de produção ou por concelho. A título de exemplo é possível observar que dos operadores ativos, 10.62% produzem ovelhas reprodutoras num total de 856 animais produzidas de forma biológica.

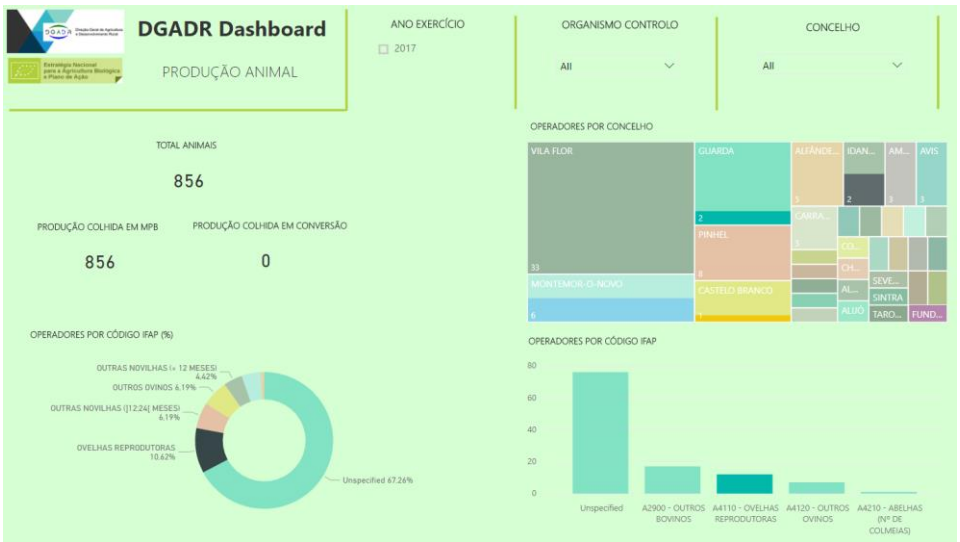


**Figura 3.40 - *Dashboard* Produção Animal**

# 4. CONCLUSÕES

Os modelos de sistemas de informação aplicados à agricultura tornaram-se ferramentas importantes com a capacidade preditiva e de avaliação, colaborando na tomada de decisão, nos setores público e privado. A implementação de um projeto de integração de agricultura biológica, em parceria com a DGADR, fez com que fossem possíveis diversas melhorias no atual processo de controlo dos operadores de produção biológica. A possibilidade de se realizarem validações no processo de recolha de dados fez com que existisse menor enviesamento e maior integração dos mesmos, facilitando a tarefa de quem verifica a recolha dos dados. Através do armazenamento da informação recolhida num *Data Warehouse* ao invés de ficheiros de Excel, tornou-se possível haver uma maior rapidez na consulta e análise dos dados que até então era morosa e complexa, assim como a utilização do processo de ETL fez com que os dados passassem a ser carregados de forma automática, com menor probabilidade de erro. Por último, a criação de *dashboards* permitiu que a forma como se analisam os dados se torne mais simples e percetível, conseguindo, de uma forma rápida e em tempo real, dar informação aos utilizadores sobre a evolução dos operadores de produção biológica em Portugal.

# 5. LIMITAÇÕES

Durante a realização do presente projeto foram encontradas algumas limitações, tais como:

- Foram levantadas questões sobre o tipo de ferramentas utilizadas devido ao licenciamento necessário e à falta de conhecimento sobre o funcionamento de determinadas tecnologias, sendo necessário haver alguma formação para a completa implementação do projeto;
- A possibilidade de se conseguir contornar a criação do ficheiro do ficheiro fonte para o processo de ETL, sabendo que nem todos os campos foram validados, levando a que a informação possa vir enviesada ou mesmo errada;
- A falta de dados consistentes para o desenvolvimento da componente de visualização de dados (*dashboards*) e validação do processo de recolha de dados;
- A falta de conhecimento concetual da base de dados de notificações levou a que não fosse possível fazer-se uma completa integração com o processo de controlo de operadores.

# 6. RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Ao longo do desenvolvimento do presente projeto de implementação foram encontrados alguns pontos de melhoria que se prepõem da seguinte forma:

- Existir uma maior integração com a base de dados de notificações, onde seja possível identificar outras características do operador que não estejam no ficheiro de Excel de controlo de operadores;
- Através do aumento do conhecimento das várias dimensões que sustentam o operador, conseguir tornar o *Data Warehouse* mais desnormalizado e criar hierarquias que possibilitem aprofundar melhor os dados;
- Implementação de uma ferramenta de *reporting* para futuros desenvolvimentos a nível de modelação analítica - OLAP;
- Adequação da capacidade de manutenção do processo de integração, tendo em conta o grau de crescimento inerente ao *Data Warehouse*.

# 7. BIBLIOGRAFIA


Abai, N. H. Z., Yahaya, J. H., & Deraman, A. (2013). User Requirement Analysis in Data Warehouse Design: A Review. *Procedia Technology*, *11*, 801–806. http://doi.org/10.1016/J.PROTCY.2013.12.261

Antle, J. M., Jones, J. W., & Rosenzweig, C. E. (2017). Next generation agricultural system data, models and knowledge products: Introduction. *Agricultural Systems*, *155*, 186–190. http://doi.org/10.1016/J.AGSY.2016.09.003

Baars, H., & Kemper, H.-G. (2008). Management Support with Structured and Unstructured Data—An Integrated Business Intelligence Framework. *Information Systems Management*, *25*(2), 132–148. http://doi.org/10.1080/10580530801941058

Ballantyne, S. F. J. C. (2001, May 10). Interactive business data visualization system. Retrieved from https://patents.google.com/patent/US6995768B2/en

Bergamaschi, S., Guerra, F., Orsini, M., Sartori, C., & Vincini, M. (2011). A semantic approach to ETL technologies. *Data & Knowledge Engineering*, *70*(8), 717–731. http://doi.org/10.1016/J.DATAK.2011.03.003

Chen, H., Chiang, R. H. L., & Storey, V. C. (2012). Business Intelligence and Analytics: From Big Data to Big Impact. *MIS Quarterly*, *36*(4), 1165. http://doi.org/10.2307/41703503

Claudia, I., Nicholas, G., & Geiger, J. G. (2003). *Mastering Data Warehouse Design: Relational and Dimensional Techniques*. *Wiley Publishing, Inc.*

Comissão Europeia. (2018). What is organic farming? | Agricultura biológica. Retrieved November 1, 2018, from https://ec.europa.eu/agriculture/organic/organic-farming/what-is-organic-farming_pt

DGADR. (2017a). Diário da República, 1.ª Série - N.º 144 - 27 de julho de 2017. Diário da República. Retrieved from https://dre.pt/application/file/a/107761804

DGADR. (2017b). Diário da República, 2.ª série - N.º 199 - 16 de outubro de 2017. Diário da República. Retrieved from https://dre.pt/application/file/a/108309044

Di Tria, F., Lefons, E., & Tangorra, F. (2017). Cost-benefit analysis of data warehouse design methodologies. *Information Systems*, *63*, 47–62. http://doi.org/10.1016/J.IS.2016.06.006

Dooley, P. P., Levy, Y., Hackney, R. A., & Parrish, J. L. (2018). Critical Value Factors in Business Intelligence Systems Implementations (pp. 55–78). Springer, Cham. http://doi.org/10.1007/978-3-319-58097-5_6

Dyché, J., & Levy, E. (2006). *Customer Data Integration: Reaching a Single Version of the Truth*. John Wiley & Sons.

El-Sappagh, S. H. A., Hendawi, A. M. A., & El Bastawissy, A. H. (2011). A proposed model for data warehouse ETL processes. *Journal of King Saud University - Computer and Information Sciences*, *23*(2), 91–104. http://doi.org/10.1016/J.JKSUCI.2011.05.005

Elias, M., Aufaure, M.-A., & Bezerianos, A. (2013). Storytelling in Visual Analytics Tools for Business Intelligence (pp. 280–297). Springer, Berlin, Heidelberg. http://doi.org/10.1007/978-3-642-40477-1_18

Few, S. (2006). *Information Dashboard Design*. *O'Reilly Press*. http://doi.org/10.1017/S0021849904040334

Gan, H., & Lee, W. S. (2018). Development of a Navigation System for a Smart Farm. *IFAC-PapersOnLine*, *51*(17), 1–4. http://doi.org/10.1016/J.IFACOL.2018.08.051

Hema, R., & Malik, N. (2010). Data Mining and Business Intelligence. In *Proceedings of the 4th National Conference*. Retrieved from http://www.bvicam.ac.in/news/INDIACom 2011/177.pdf

Janssen, S. J. C., Porter, C. H., Moore, A. D., Athanasiadis, I. N., Foster, I., Jones, J. W., & Antle, J. M. (2017). Towards a new generation of agricultural system data, models and knowledge products: Information and communication technology. *Agricultural Systems*, *155*, 200–212. http://doi.org/10.1016/J.AGSY.2016.09.017

Kakish, K., & Kraft, T. A. (2012). ETL Evolution for Real-Time Data Warehousing. In *Proceedings of the Conference on Information Systems Applied Research*. Retrieved from www.aitp-edsig.org

Kimball, R., & Caserta, J. (2014). *The Data Warehouse ETL Toolkit*. *Igarss 2014*. http://doi.org/10.1017/CBO9781107415324.004

Kimball, R., Reeves, L., Ross, M., & Thornthwaite, W. (2008). *The Data Warehouse Lifecycle Toolkit*. *WILEY*. http://doi.org/10.1017/CBO9781107415324.004

Kimball, R., & Ross, M. (2011). *The Data Warehouse Toolkit: the Definitive Guide to Dimensional Modelling*. *WILEY*. http://doi.org/10.1145/945721.945741

Kiritani, K., & Ohashi, M. (2015). The Success or Failure of the Requirements Definition and Study of the Causation of the Quantity of Trust Existence Between Stakeholders. *Procedia Computer Science*, *64*, 153–160. http://doi.org/10.1016/J.PROCS.2015.08.476

Larson, D., & Chang, V. (2016). A review and future direction of agile, business intelligence, analytics and data science. *International Journal of Information Management*, *36*(5), 700–710. http://doi.org/10.1016/J.IJINFOMGT.2016.04.013

Levene, M., & Loizou, G. (2003). Why is the snowflake schema a good data warehouse design? *Information Systems*, *28*(3), 225–240. http://doi.org/10.1016/S0306-4379(02)00021-2

Popovi, A., Coelho, P. S., & Jakli, J. (2009). *The impact of business intelligence system maturity on information quality*. Retrieved from http://ssrn.com/abstract=1625573Electroniccopyavailableat:http://ssrn.com/abstract=1625573

Popovič, A., Hackney, R., Coelho, P. S., & Jaklič, J. (2014). How information-sharing values influence the use of information systems: An investigation in the business intelligence systems context. *The Journal of Strategic Information Systems*, *23*(4), 270–283. http://doi.org/10.1016/J.JSIS.2014.08.003

Salaki, R. J., Waworuntu, J., & Tangkawarow, I. R. H. T. (2016). Extract transformation loading from OLTP to OLAP data using pentaho data integration. In *IOP Conference Series: Materials Science and Engineering*. http://doi.org/10.1088/1757-899X/128/1/012020

Todman, C. (2001). *Designing A Data Warehouse: Supporting Customer Relationship Management*. *Prentice Hall*.

Trieu, V.-H. (2017). Getting value from Business Intelligence systems: A review and research agenda. *Decision Support Systems*, *93*, 111–124. http://doi.org/10.1016/J.DSS.2016.09.019

Watson, H. J., & Wixom, B. H. (2007). The Current State of Business Intelligence. *Computer*. http://doi.org/10.1109/MC.2007.331

Willer, H. and Lernoud, J. (2016). *The World of Organic Agriculture 2016: Statistics and Emerging Trends*. *the World of Organic Agriculture*. http://doi.org/10.4324/9781849775991

Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M.-J. (2017). Big Data in Smart Farming – A review. *Agricultural Systems*, *153*, 69–80. http://doi.org/10.1016/J.AGSY.2017.01.023

Zhao, K., Sun, R., Deng, C., Li, L., Wu, Q., & Li, S. (2018). Visual Analysis System for Market Sales Data

of Agricultural Products. *IFAC-PapersOnLine*, *51*(17), 741–746. http://doi.org/10.1016/J.IFACOL.2018.08.107

# 8. ANEXOS

```
'FUNÇÃO PARA VALIDAÇÃO DO NIF

Function ValidateNIF(tbl_Range As Range)

For Each i In tbl_Range
    If IsNumeric(i) And Len(i) = 9 Then

        i.Value = UCase(i)
        i.Font.Name = "Calibri"
        i.Font.Size = 10
        i.HorizontalAlignment = xlCenter
        i.Interior.Color = RGB(255, 255, 255)

        With i.Borders
            .LineStyle = xlContinuous
            .Color = vbBlack
            .Weight = xlThin
        End With

    ElseIf i.Value = "" Then

        i.Font.Name = "Calibri"
        i.Font.Size = 10
        i.HorizontalAlignment = xlCenter
        i.Interior.Color = RGB(255, 255, 255)

        With i.Borders
            .LineStyle = xlContinuous
            .Color = vbBlack
            .Weight = xlThin
        End With

        Exit For

    Else

        i.Font.Name = "Calibri"
        i.Font.Size = 10
        i.HorizontalAlignment = xlCenter
        i.Interior.Color = RGB(255, 0, 0)

        With i.Borders
            .LineStyle = xlContinuous
            .Color = vbBlack
            .Weight = xlThin
        End With

    End If
Next

End Function
```

**Figura 8.1 - Função para Validação do NIF**

```
'FUNÇÃO PARA VALIDAÇÃO DE UM CAMPO NÚMERICO

Function ValidateNumeric(tbl_Range As Range)

For Each i In tbl_Range
    If IsNumeric(i) And i.Value <> "" Then

        i.Value = UCase(i)
        i.Font.Name = "Calibri"
        i.Font.Size = 10
        i.HorizontalAlignment = xlCenter
        i.Interior.Color = RGB(255, 255, 255)

        With i.Borders
            .LineStyle = xlContinuous
            .Color = vbBlack
            .Weight = xlThin
        End With

    ElseIf i.Value = "" Then

        i.Font.Name = "Calibri"
        i.Font.Size = 10
        i.HorizontalAlignment = xlCenter
        i.Interior.Color = RGB(255, 255, 255)

        With i.Borders
            .LineStyle = xlContinuous
            .Color = vbBlack
            .Weight = xlThin
        End With

        Exit For

    Else

        i.Font.Name = "Calibri"
        i.Font.Size = 10
        i.HorizontalAlignment = xlCenter
        i.Interior.Color = RGB(255, 0, 0)

        With i.Borders
            .LineStyle = xlContinuous
            .Color = vbBlack
            .Weight = xlThin
        End With

    End If
Next

End Function
```

**Figura 8.2 - Função para Validação de um Campo Numérico**

```
'FUNÇÃO PARA VALIDAÇÃO DE UM CAMPO ALFANÚMERICO

Function ValidateString(tbl_Range As Range)

For Each i In tbl_Range
    If Not IsNumeric(i) Then

        i.Value = UCase(i)
        i.Font.Name = "Calibri"
        i.Font.Size = 10
        i.HorizontalAlignment = xlCenter
        i.Interior.Color = RGB(255, 255, 255)

        With i.Borders
            .LineStyle = xlContinuous
            .Color = vbBlack
            .Weight = xlThin
        End With

    ElseIf i.Value = "" Then

        i.Font.Name = "Calibri"
        i.Font.Size = 10
        i.HorizontalAlignment = xlCenter
        i.Interior.Color = RGB(255, 255, 255)

        With i.Borders
            .LineStyle = xlContinuous
            .Color = vbBlack
            .Weight = xlThin
        End With

        Exit For

    Else

        i.Font.Name = "Calibri"
        i.Font.Size = 10
        i.HorizontalAlignment = xlCenter
        i.Interior.Color = RGB(255, 0, 0)

        With i.Borders
            .LineStyle = xlContinuous
            .Color = vbBlack
            .Weight = xlThin
        End With

    End If
Next

End Function
```

**Figura 8.3 - Função para Validação de um Campo Alfanumérico**

```
'FUNÇÃO PARA VALIDAÇÃO DE UMA LOOKUP

Function ValidateLKP(tbl_Range As Range, lkp_Range As Range)

For Each i In tbl_Range

lkp_value = Application.WorksheetFunction.CountIf(lkp_Range, i.Value) > 0

    If Not IsNumeric(i) And lkp_value Then

        i.Value = UCase(i)
        i.Font.Name = "Calibri"
        i.Font.Size = 10
        i.HorizontalAlignment = xlCenter
        i.Interior.Color = RGB(255, 255, 255)

        With i.Borders
            .LineStyle = xlContinuous
            .Color = vbBlack
            .Weight = xlThin
        End With

     ElseIf i.Value = "" Then

        i.Font.Name = "Calibri"
        i.Font.Size = 10
        i.HorizontalAlignment = xlCenter
        i.Interior.Color = RGB(255, 255, 255)

        With i.Borders
            .LineStyle = xlContinuous
            .Color = vbBlack
            .Weight = xlThin
        End With

        Exit For

     Else

        i.Font.Name = "Calibri"
        i.Font.Size = 10
        i.HorizontalAlignment = xlCenter
        i.Interior.Color = RGB(255, 0, 0)

        With i.Borders
            .LineStyle = xlContinuous
            .Color = vbBlack
            .Weight = xlThin
        End With

     End If
Next

End Function
```

**Figura 8.4 - Função para Validação de uma Lookup**

```
'FUNÇÃO PARA VALIDAÇÃO DE UM CAMPO DE EMAIL

Function ValidateEmailAddress_F(ByVal strEmailAddress As String) As Boolean

    Dim objRegExp As New RegExp
    Dim blnIsValidEmail As Boolean

    objRegExp.IgnoreCase = True
    objRegExp.Global = True
    objRegExp.Pattern = "^([a-zA-Z0-9_\-\.]+)@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})$"

    blnIsValidEmail = objRegExp.Test(strEmailAddress)
    ValidateEmailAddress_F = blnIsValidEmail

    Exit Function

End Function
```

```
Function ValidateEmailAddress(tbl_Range As Range)

    For Each i In tbl_Range

        If ValidateEmailAddress_F(i) Then

            i.Value = UCase(i)
            i.Font.Name = "Calibri"
            i.Font.Size = 10
            i.HorizontalAlignment = xlCenter
            i.Interior.Color = RGB(255, 255, 255)

            With i.Borders
                .LineStyle = xlContinuous
                .Color = vbBlack
                .Weight = xlThin
            End With

        ElseIf i.Value = "" Then

            i.Font.Name = "Calibri"
            i.Font.Size = 10
            i.HorizontalAlignment = xlCenter
            i.Interior.Color = RGB(255, 255, 255)

            With i.Borders
                .LineStyle = xlContinuous
                .Color = vbBlack
                .Weight = xlThin
            End With

            Exit For

        Else

            i.Font.Name = "Calibri"
            i.Font.Size = 10
            i.HorizontalAlignment = xlCenter
            i.Interior.Color = RGB(255, 0, 0)

            With i.Borders
                .LineStyle = xlContinuous
                .Color = vbBlack
                .Weight = xlThin
            End With


        End If
    Next

End Function
```

**Figura 8.5 - Função para Validação de um Campo de Email**

```
Function ValidateDate(tbl_Range As Range)

    For Each i In tbl_Range

        If ValidateDate_F(i) Then

            i.Value = UCase(i)
            i.Font.Name = "Calibri"
            i.Font.Size = 10
            i.HorizontalAlignment = xlCenter
            i.Interior.Color = RGB(255, 255, 255)

            With i.Borders
                .LineStyle = xlContinuous
                .Color = vbBlack
                .Weight = xlThin
            End With

         ElseIf i.Value = "" Then

            i.Font.Name = "Calibri"
            i.Font.Size = 10
            i.HorizontalAlignment = xlCenter
            i.Interior.Color = RGB(255, 255, 255)

            With i.Borders
                .LineStyle = xlContinuous
                .Color = vbBlack
                .Weight = xlThin
            End With

            Exit For

         Else

            i.Font.Name = "Calibri"
            i.Font.Size = 10
            i.HorizontalAlignment = xlCenter
            i.Interior.Color = RGB(255, 0, 0)

            With i.Borders
                .LineStyle = xlContinuous
                .Color = vbBlack
                .Weight = xlThin
            End With


         End If
    Next

End Function
```

**Figura 8.6 - Função para Validação de um Campo de Data**


**Figura 8.7 - Diagrama do *Data Warehouse***

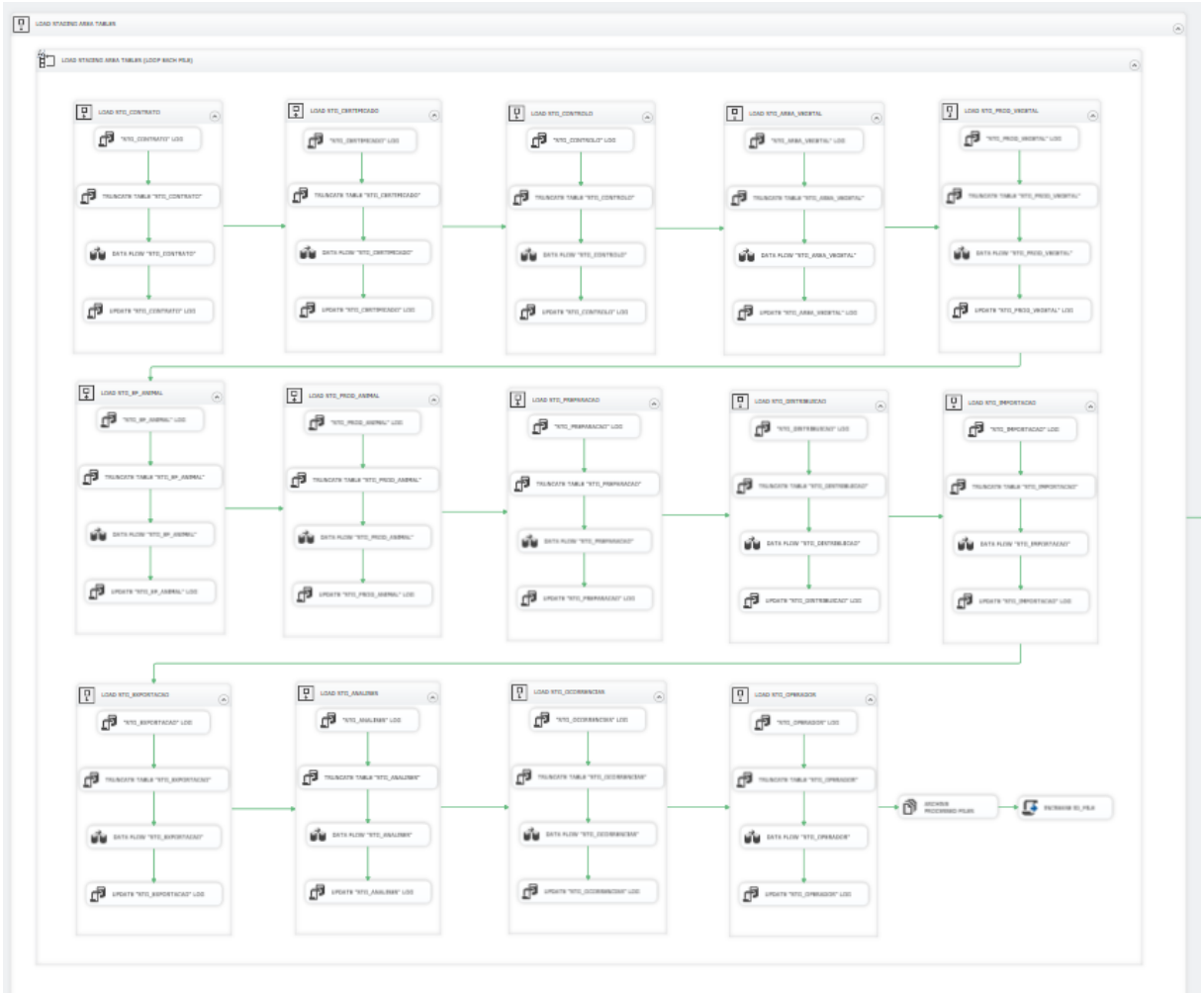

**Figura 8.8 - Diagrama do Processo de ETL (*Staging Area*)**

**Figura 8.9 - Diagrama do Processo de ETL (*Data Warehouse*)**