P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO
POLITÉCNICO
DO PORTO

M

MESTRADO
ENGENHARIA INFORMÁTICA

Framework BPMN para a Modelação de Processos de ETL

Ana Letícia Martins Ribeiro

11/2022

Framework BPMN para a Modelação de Processos de ETL
Ana Letícia Martins Ribeiro
11/2022

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO
POLITÉCNICO
DO PORTO

M

MESTRADO
ENGENHARIA INFORMÁTICA

Framework BPMN para a Modelação de Processos de ETL

Ana Letícia Martins Ribeiro

Professor Bruno Oliveira

Professor Óscar Oliveira

Agradecimentos

A realização deste trabalho tornou-se possível através do contributo e apoio de várias pessoas, às quais expresso o meu sincero agradecimento.

Aos meus orientadores, o professor Bruno Oliveira e o professor Óscar Oliveira, pela sua orientação, acompanhamento do trabalho desenvolvido, pelas correções e sugestões relevantes que foram feitas durante a orientação.

Aos meus diretores de curso, a professora Dorabela Gamboa e o professor Rui Soares, pela sua disponibilidade, incentivo e prontidão.

À minha família, pelo seu continuo apoio e incentivo durante todo o meu percurso académico.

Por último, a todas as pessoas que de uma forma ou de outra se disponibilizaram a ajudar para que atingisse este objetivo.

Resumo

O *Extract-Transform-Load* (ETL) é um componente crítico nos Sistemas de *Data Warehousing* (SDW) sendo responsável por extrair, transformar e carregar dados para apoiar os requisitos de tomada de decisão. Devido à complexidade da gestão dos dados, estes processos consomem grande parte dos recursos necessários na implementação dos SDW. Sendo um componente crítico que pode comprometer a adequação do sistema, se não fornecer garantias na qualidade de dados, a confiança no sistema é comprometida. Apesar da sua importância, o desenvolvimento de sistemas de ETL é essencialmente ad-hoc, o que não contribui para garantir o seguimento de práticas sólidas que garantam a coerência e coesão do desenvolvimento dos sistemas. Nos últimos anos, a *Business Process Model and Notation* (BPMN) tem sido proposta e utilizada para suportar os modelos conceptuais de ETL. O BPMN é uma linguagem expressiva que permite diferentes abordagens para representar os requisitos de povoamento dos processos de ETL. Neste trabalho, é explorada a utilização de BPMN para modelação conceptual de ETL, analisando as abordagens existentes e propondo um conjunto de diretrizes para utilizar o BPMN de uma forma mais consistente.

Palavras-chave: Data Warehousing, Business Process Model and Notation, Extract-Transform-Load, modelação conceptual ETL.

Abstract

The Extract-Transform-Load (ETL) is a critical component in Data Warehousing Systems (SDW) being responsible for extracting, transforming, and loading data to support decision-making requirements. Due to the complexity of data management, these processes consume a large part of the resources needed in the implementation of SDW. Being a critical component that can compromise the suitability of the system, if it does not provide guarantees in data quality, trust in the system is compromised. Although its importance, the development of ETL systems is essentially ad-hoc, which does not contribute to guaranteeing the follow-up of solid practices that guarantee the coherence and cohesion of the development of the systems. In recent years, the Business Process Model and Notation (BPMN) has been proposed and used to support the conceptual models of ETL. BPMN is an expressive language that allows different approaches to represent the population requirements of ETL processes. In this work, the use of BPMN for conceptual modeling of ETL is explored, analyzing the existing approaches, and proposing a set of guidelines to use BPMN in a standardized way.

Keywords: Data Warehousing, Business Process Model and Notation, Extract-Transform-Load, conceptual modeling ETL

Índice

Índice de Figuras

Abreviaturas

ETL – Extract-Transform-Load

BPMN – Business Process Modelling and Notation

OMG – Object Management Group

BI – Business Intelligence

DW – Data Warehouse

SDW – Sistema de *Data Warehousing*

BPEL – Business Process Execution Language

XML – Extensible Markup Language.

XPDL – *Process Definition Language*

BPMS – Business Process Management System

BD – Base de Dados

EPC – Event-driven Process Chain

OLTP – Online Transactional Processing

OLAP – Online Analytical Processing

BI – Business Intelligence

CDC – Change Data Capture

SSIS – SQL Server Integration Services

SGBDs – Sistemas de Gestão de Bases de Dados

UML – Unified Modeling Language

MDA – Model Driven Architecture

PIM – Platform Independent Model

PSM – Platform Specific Model

QVT – Query View Transformation

SysML – Systems Modeling Language

DFM – Dimensional Fact Model

WWI – Wide World Importers

SK – Surrogate Key

SCD – Slowly Changing Dimension

SKP – Surrogate Key Pipelining

SQL – Structured Query Language

Capítulo 1 - Introdução

A necessidade de capturar e analisar dados está em constante evolução, uma vez que as organizações expandem frequentemente a sua gama de parceiros e atividades de negócio [1]. As tendências de pesquisa atuais remontam a abordagens analíticas e de armazenamento de dados que visam lidar com dados menos estruturados que vêm em larga escala [2]. No entanto, possuir grandes quantidades de dados não garante decisões informadas [3], sendo necessário garantir que o núcleo (de dados) preserva os requisitos de qualidade para poder ser utilizado no apoio aos processos de tomada de decisão.

As necessidades organizacionais estão em constante evolução e a forma em que os dados chegam para fins de análise também sofreram grandes mudanças, pois os dados não estruturados, produzidos por diversos agentes, revolucionou a forma como os dados são tratados. Para além das fontes de dados tradicionais (ex.: sistemas de base de dados tradicionais, folhas de cálculo, etc.) existe agora uma necessidade para a integração de dados não estruturados (ex.: comentários de redes sociais ou imagens de câmaras), o que eleva ainda mais a complexidade de implementação dos sistemas de Extração, Transformação e Carregamento [4], [5] conhecidos como *Extract-Transform-Load* (ETL).

1.1 Contextualização

Para além da análise de dados, *Business Intelligence* (BI) permite também a entrega e integração de informação útil através da monitorização de eventos de negócio de modo a melhorar os processos de tomada de decisão, considerando todos os níveis de gestão. BI envolve tecnologias e estratégias que permite aos utilizadores finais analisar os dados (atuais e históricos), criar relatórios e realizar tarefas de mineração de dados. Neste contexto, o *Data Warehouse* (DW) tem sido utilizado para armazenar e gerir os dados analíticos. A sua premissa é relativamente simples: representar a única fonte de verdade de uma organização. Isto é possível através de um trabalho complexo de extração de dados (de fontes possivelmente heterogéneas e tipicamente corporativas) e alinhar a sua estrutura aos requisitos analíticos definidos pelo DW. Um DW representa um único repositório especialmente construído para suportar requisitos analíticos, podendo fornecer *insights* cruciais para os processos de tomada de decisão. Adicionalmente, o DW permite a captura dados operacionais em vários pontos do tempo visando a preservação do histórico de

operações. A preservação de dados históricos é um dos aspetos mais importantes de um DW [6]. Todas as alterações relevantes efetuadas nos dados de origem são preservadas. O ETL é reconhecidamente o componente mais crítico de qualquer DW sendo responsável por extrair, transformar, conciliar e carregar dados de fontes de dados (internas e externas) para estruturas de dados especialmente configuradas para suportar os requisitos de tomada de decisão [4]. Dada a complexidade da gestão de dados, os processos de ETL são responsáveis por consumir os principais recursos na implementação de DW. O ETL representa um componente crítico que compromete a adequação de qualquer sistema: se falham no fornecimento de qualidade dos dados, a confiança do sistema ficará comprometida [4], [7].

As soluções ETL tradicionais extraem os dados de bases de dados, transformam os dados de acordo com os requisitos associados aos processos de tomada de decisão, carregando-os posteriormente para um DW de destino e/ou para as soluções analíticas específicas.

O DW é um recurso caro, pois é uma solução extremamente personalizada. Na área de desenvolvimento de software é comum a utilização de *frameworks* com soluções pré-construídas para simplificar o desenvolvimento de software. No entanto, é notório que no desenvolvimento de um DW é muito difícil reutilizar componentes de software. Normalmente, para dados específicos de clientes, surgem diferentes formas de tomar decisões, que tipicamente requerem modelos dimensionais diferentes e, consequentemente, conjuntos de dados diferentes [8]. Assim, mesmo que seja adotada a mesma solução padrão para implementar um DW (o que acontece muitas vezes na área de venda a retalho deste tipo de sistemas), as pessoas têm diferentes formas de pensar e necessidades, o que significa que os modelos de dados originais pré-construídos numa solução devem ser ajustados de acordo com os requisitos dos decisores. Os sistemas suportados por DW podem demorar meses e serem construídos e podem ter um grande impacto a nível financeiro e estratégico. Isso ocorre porque a equipa de ETL precisa de entender o negócio para o qual o DW será implementado e conhecer todas as especificidades dos dados armazenados. Os esforços de manutenção também são muito elevados devido às variáveis inesperadas às mudanças nos processos de negócio, onde grandes esforços de desenvolvimento são necessários.

1.2 Motivação e Objetivos

A qualidade dos dados só pode ser garantida se os procedimentos e metodologias utilizados para criar processos de integração e conciliação de dados fornecerem padrões e práticas sólidas para que os processos de desenvolvimento possam melhorar a confiança na qualidade do sistema [9].

O componente de ETL é o coração do DW ou das soluções de análise de dados. Alterar um processo de ETL (para cumprir, por exemplo, com uma mudança nos requisitos organizacionais) é um desafio extremamente complexo [10]. Os sistemas ETL são reconhecidos como um componente crítico e que consome cerca de 80% dos recursos necessário à implementação de um DW [4].

A equipa de desenvolvimento de um DW deve ter um conhecimento sólido da atividade do negócio para reduzir a ambiguidade e mal-entendidos. Além disso, estas equipas devem também identificar problemas com os dados (como dados ausentes, incoerentes ou incompatíveis). Em contraste com as fontes de dados operacionais, os sistemas analíticos não impõem os princípios de normalização de dados nem mesmo garantem as transações comerciais. Eles são criados apenas para fins analíticos e, por isso, se os dados forem de baixa qualidade, todo o sistema fica comprometido. A qualidade e a complexidade dos dados de origem têm um impacto significativo no desenvolvimento do sistema ETL.

Na maioria dos casos, a equipa ETL concentra-se em ferramentas e tecnologias, em vez de práticas fundamentais. O ETL é um componente extremamente técnico que captura a implementação de tarefas muito granulares. Normalmente, o desenvolvimento de ETL é orientado por práticas ad-hoc que, ao contrário do que tradicionalmente acontece com o desenvolvimento de software, não adere a uma metodologia sólida que oriente e documente o processo de desenvolvimento de uma representação conceptual para uma implementação física [11]. Algumas ferramentas ETL ajudam nesse aspeto, no entanto, cada ferramenta fornece seu próprio método com sua própria notação e metodologia. Normalmente estas ferramentas, apenas cobrem o nível físico do ETL, sem fornecer primitivas (conceptuais) que possam ser utilizadas para descrever o sistema de forma mais conceptual. Portanto, quando é necessário alterar a ferramenta de ETL, o sistema precisa, normalmente, ser todo reimplementado.

Considerando os processos ETL como um componente crítico para a implementação do DW, a definição de um modelo detalhado será benéfica ao longo de todo o desenvolvimento, instalação e validação do processo, além de fornecer aos arquitetos e engenheiros um guia prático e acessível. O desenvolvimento do ETL ainda é fortemente baseado em características físicas utilizadas para suportar sua execução, ou seja, os processos desenvolvidos são implementados através do recurso a ferramentas e linguagens específicas restritas ou enquadradas por características arquiteturais. Embora a maioria dessas ferramentas seja muito poderosa para a execução de ETL, não fornecem os recursos necessários para documentar e representar processos de integração num nível maior de abstração, além de limitarem a comunicação com utilizadores não técnicos pela utilização de notações específicas. Dessa forma, o sistema torna-se complexo e de difícil compreensão por não possuir uma representação mais direta.

A *Business Process Modelling and Notation*[1] (BPMN) [12] é uma notação de modelação padrão que fornece um conjunto de artefactos que permitem modelar processos de negócio. A BPMN foca dois pontos-chave: a gestão e o planeamento de um *workflow* de um processo; e a modelação da arquitetura de implementação [13]. A BPMN é independente das ferramentas de implementação utilizadas e pode ser mapeado para primitivas de execução como *Business Process Execution Language* (BPEL) [13] ou através da utilização da semântica de execução BPMN 2.0 [14]. Nos últimos anos, o BPMN tem sido utilizado como uma forma de representar processos ETL a um nível de abstração mais elevado, permitindo que a equipa de desenvolvimento se concentre nos aspetos fundamentais do fluxo de trabalho [15]. A escolha da notação BPMN para modelação ETL dá-se principalmente pela sua simplicidade na representação e modelação de processos de negócio e ao seu poder de expressividade (o que torna essa notação facilmente aplicada no contexto de processos ETL). Esta abordagem de representação possui várias vantagens que podem minimizar alguns dos problemas tipicamente encontrados no desenvolvimento de ETL, nomeadamente:

- Utilização de procedimentos típicos do processo de desenvolvimento de ETL, através da identificação de componentes de sistema reutilizáveis;

[1] Em português, Notação de Modelação de Processos de Negócio.

- Melhorar a compreensão do sistema e consequentemente a comunicação com a equipa de desenvolvimento, uma vez que se trabalha com componentes de alto-nível que escondem especificidades relacionadas com o processamento e tratamento de dados;
- Encapsular as melhores práticas para resolver problemas recorrentes;
- Disponibilizar um modelo de alto-nível que garanta flexibilidade no desenvolvimento de ETL;
- Permitir representar o processo ETL independentemente da plataforma de implementação. Uma clara distinção e independência entre modelos abstratos e a sua representação física abstrai os utilizadores dos detalhes de implementação, contribuindo para a compreensibilidade do processo, e melhora o planeamento, a comunicação e a produtividade.

Em vários cenários, a ambiguidade da notação é uma grande vantagem uma vez que permite explorar o desenho dos processos sobre várias perspetivas, o que aumenta a expressividade do processo principalmente numa fase inicial de desenvolvimento. No entanto, para a implementação de processos a partir de especificações BPMN, esta ambiguidade é um problema dado que dificulta a interpretação dos processos a nível de execução. Assim, com o desenvolvimento deste trabalho, pretende-se:

- Identificar padrões de desenho (*design patterns*) na modelação de processos ETL;
- Propor uma metodologia de representação para o processo ETL utilizando BPMN, tendo em conta as melhores práticas e diretrizes identificadas através do estudo de casos de estudo. Em concreto pretende-se selecionar cuidadosamente os artefactos da notação BPMN que devem ser utilizados para a caracterização de processos de ETL considerando diferentes níveis de abstração;
- Exemplificar a metodologia de modelação proposta recorrendo a um caso de estudo.

1.3 Contributos Inovadores

A qualidade dos dados apenas pode ser garantida se, as metodologias e procedimentos utilizados na criação dos processos de integração de dados forem desenvolvidos com base em padrões comuns e práticas sólidas. Para melhorar a confiança na qualidade do sistema deve ser utilizada uma metodologia de desenho que cubra as principais fases de desenvolvimento de ETL, desde a modelação conceptual à modelação física.

Os benefícios potenciais deste trabalho não só podem contribuir diretamente para minimizar os problemas atuais em sistemas de ETL, mas também podem contribuir para abrir novas oportunidades para gestão e análise de dados.

O desenvolvimento deste trabalho tem como premissa propor uma metodologia de desenvolvimento ETL que promova as melhores práticas e diretrizes na fase de modelação conceptual, permitindo o desenvolvimento de modelos que possam ser utilizados como base para a implementação dos processos nas ferramentas comerciais.

Como contributo inovador, será proposta uma metodologia para modelação conceptual de ETL, onde será desenvolvido um conjunto de orientações que suporte a criação de modelos conceptuais. Esta representação visa proporcionar uma perceção do processo a qualquer tipo de utilizador, seja o programador do sistema, seja o analista do negócio ou os responsáveis pelos processos de tomada de decisão.

1.4 Estrutura da Dissertação

Para além deste capítulo, este documento encontra-se estruturado da seguinte forma:

- O Capítulo 2 apresenta a notação BPMN para a modelação de processos de negócio, apresentando não só os vários artefactos de modelação como também os diferentes tipos de processos de negócio que podem ser representados. Neste capítulo, é ainda apresentada uma visão geral das principais ferramentas de modelação de processos BPMN.
- O Capítulo 3 apresenta os principais conceitos associados ao desenvolvimento de processos ETL, assim como as principais abordagens de modelação, não só conceptual, mas também a nível lógico.
- O Capítulo 4 apresenta uma análise acerca da utilização da BPMN no desenvolvimento de modelos conceptuais de processos de ETL.
- O Capítulo 5 apresenta uma proposta de documento de convenções expondo as melhores práticas de modelação e apresenta uma proposta de três níveis de modelação de processos ETL através da utilização da notação BPMN.
- O Capítulo 6 apresenta um caso de estudo em que a metodologia de desenvolvimento de ETL com BPMN é aplicada, passo a passo.

- O Capítulo 7 apresenta as principais conclusões do trabalho realizado, assim como uma análise crítica. São ainda exploradas as principais linhas orientadoras de trabalho futuro.

Capítulo 2 – A notação BPMN

A *Business Process Model and Notation* (BPMN) é uma notação gráfica para modelação processos de negócio, muito semelhante ao conceito de fluxograma que existe desde a década de 1980. Foi desenvolvida pelo *Object Management Group* (OMG) em 2004 com o intuito de descrever e documentar os processos de negócio [16]. Surgiu como substituta aos fluxogramas que consistiam em pequenas caixas e setas de fluxo e ao diagrama *Unified Modeling Language* (UML) o primeiro método estruturado que surgiu para o fluxo de um processo [17]. Os fluxogramas apresentam alguns problemas, entre os quais a incapacidade de atender facilmente aos requisitos de representação de processos extensos com todos os seus aspetos relevantes, como regras de divisão, eventos, unidades organizacionais, fluxos de dados, entre outros [18]. Este (histórico) problema dos fluxogramas - em que cada modelador de processos podia inventar os seus próprios significados e interpretações aos diagramas que produzia - teve a consequência indesejada de que nem todos interpretariam os processos da mesma forma promovendo dificuldades na comunicação dos processos [16].

Para colmatar estas dificuldades, seria necessária uma notação que se apresentasse mais adequada à representação de processos. Uma notação para modelação de processos é uma linguagem padronizada para a descrição de processos de negócio, onde qualquer interveniente familiarizado com a linguagem pode compreender os modelos criados por outra pessoa. Além disso, os processos podem ser sistematicamente analisados e o seu comportamento dinâmico pode ser simulado com base numa representação padronizada. Os modelos também fornecem uma base para o desenvolvimento de sistemas de informação, para a execução e suporte de processos de negócios e precisam de uma estrutura padronizada que contenha todas as informações relevantes para o desenvolvimento do sistema.

Os processos suportados pelo sistema são cada vez mais controlados por *Business Process Management Systems* (BPMS). Eles contêm motores de processo que controlam diretamente os fluxos de trabalho através da utilização de modelos de processos apropriados ou descrições formais de processos. Como resultado, diferentes *softwares* podem partilhar e editar processos projetados com BPMN. Para tal, os modelos têm de atender a exigências muito restritas, pois não são convertidos num programa de computador por um ser humano, mas são diretamente processados por uma máquina [19].

A BPMN foi projetada para ser fácil de utilizar e de ser entendida por todos os envolvidos no processo de negócio, por exemplo, analistas de negócio, programadores e/ou técnicos responsáveis pela implementação da tecnologia que executará esses processos e os empresários responsáveis por gerir e monitorizar esses processos. A BPMN tem sido amplamente utilizada em vários domínios para a modelação de vários tipos processos, tendo ganho popularidade devido à sua simplicidade e expressividade para representar os mais diversos fluxos, intervenientes, condições e eventos associados.

A especificação mais recente da notação é o BPMN 2.0[2], publicada em 2011, desde então tornou-se a linguagem mais popular de modelação de processos de negócios pois muitos profissionais de negócios sentem-se confortáveis com a sua utilização para visualizar fluxos de trabalho. É também uma notação compreensível para programadores, preenchendo assim a lacuna entre os utilizadores técnicos e não técnicos. As suas principais características são a versatilidade para modelar diversos cenários, a utilização de entendimento difundido e o facto de ser suportada por BPMS. Estas características podem ser observadas através dos níveis de modelação. Devido ao crescente número de conceitos em BPMN foram definidos três níveis de representação [20]:

- Nível 1: Modelação descritiva – este nível de modelação consiste numa representação inicial do processo de negócio demonstrando as suas principais atividades e tem como objetivo a documentação simples do fluxo do processo, ou seja, o mapeamento e descrição de processos do negócio e a respetiva documentação do fluxo do processo. É o tipo de modelação sobre o qual a maioria dos consultores de modelação de processos de negócio se refere. É de alto-nível onde ocasionalmente as regras de validação dos diagramas BPMN são ignoradas, mas em contrapartida facilita a comunicação em toda a organização. É geralmente utilizado para descrever o trabalho rotineiro ou demonstrar os processos existentes numa organização sem nenhum nível de especificidades [20].
- Nível 2: Modelação analítica – este nível de modelação consiste numa evolução do nível anterior, onde são acrescentados regras e resultados, permitindo uma modelação mais precisa e detalhada. Promove a interação entre diferentes setores e pessoas envolvidas no projeto, como analistas de negócio, técnicos e gestores. Neste

[2] https://www.bpmn.org

nível as atividades do processo e os seus objetivos tornam-se mais claros e são elaboradas representações que possam descrever detalhadamente o fluxo das atividades. Neste nível também podem ser incluídas análises específicas de desempenho para fins de otimização. A modelação analítica deve ser validada de acordo com as regras de especificação do BPMN e organizadas de forma eficaz através de representações hierárquicas do processo de negócio. É por exemplo utilizado nas áreas de recursos humanos, de logística e de compras [20].

- Nível 3: Modelação executável – consiste numa evolução do nível anterior em que a representação assume mais precisão e consequentemente mais detalhe no processo. Tem como vantagem a capacidade de execução do processo através de simulações. As simulações geram informações que acercam o desempenho do processo e validam a execução do processo segundo as regras de modelação BPMN. Este nível de execução requer uma descrição detalhada dos atributos do processo, descrevendo as duas propriedades e características. As representações continuam compreensíveis e manipuláveis por analistas e arquitetos de negócio, porém, objetiva-se a conversão dos diagramas para *softwares* e tecnologias atreladas a BPMN 2.0 que automatizem os processos, onde os modelos gráficos podem ser transformados em especificações baseadas em *Extensible Markup Language* (XML) que impulsionam os mecanismos de processo, nomeadamente a XML *Process Definition Language* (XPDL). Ao contrário do BPMN que é uma notação gráfica disponível apenas em ferramentas que adotam o padrão, o XPDL é uma especificação textual que é independente de ferramenta. A especificação XPDL possui no seu conteúdo a relação de todos os elementos de um processo. Nele são detalhadas as principais características de um processo: os seus elementos, as informações de entrada e saída e a definição do fluxo de trabalho. Os principais elementos são as atividades do processo, os eventos ou os estados do processo em determinada fase de execução, as condições existentes dentro de um processo e os responsáveis pelas atividades [20], [21].

Estes três níveis de representação de processos diferem no que diz respeito à abstração, informação, precisão, complexidade, utilidade, padronização de elementos da notação BPMN e evolução e enriquecimento do desenho alinhado ao propósito para o qual é desenvolvido e a escolha do modelo a implementar deve ter em conta esses fatores.

2.1 Elementos BPMN

A notação BPMN representa um conjunto de artefactos que podem ser agrupados em diferentes categorias, nomeadamente, *Flow Objects*, *Connecting Objects*, *Swimlanes, Data objects* e *Artifacts* [22], [19].

Os *Flow Objects* (objetos de fluxo) são ligados num diagrama através dos *Connecting Objects* (objetos de ligação) para criar a estrutura básica de um processo de negócio. Os *Flow Objects podem ser*: *Events* (eventos)*, Activities* (atividades) ou *Gateways* (decisões). Relativamente aos objetos de ligação, estes podem ser: *Sequence Flows* (fluxos de Sequência)*, Message Flows* (fluxos de mensagens) ou *Associations* (associações)*.* Os objetos *swimlanes* são elementos estruturantes do BPMN que permitem diferenciar entre as seções de um diagrama BPMN rotulando o nome de cada processo. Os modelos BPMN são focados em fluxos de sequência e de mensagem e em troca de dados, os dados são representados pelos *Data Objects* (objetos de dados)*.* Se existirem outros aspetos relevantes para um processo de negócio que precisem de ser mapeados, é possível recorrer à utilização de *Artifacts* (artefactos) [19].

2.1.1 Atividades

Segundo [19] as atividades são componentes básicos do BPMN 2.0, e sem elas o processo de negócio não existe. Os modelos BPMN podem conter vários tipos de atividades como ilustrado na figura 1.

Figura 1 - Tipos de atividades base

Os tipos de atividade podem ser:

- Tarefa - é o nível mais granular de um processo. É uma ação que ocorre num processo de negócio, ou seja, é um termo genérico que descreve o trabalho que a empresa

realiza. Essa tarefa pode ser executada por uma pessoa ou por um sistema, por exemplo, o envio de uma carta é uma tarefa. A sua representação é feita por um retângulo de cantos arredondados com a descrição da tarefa no centro desse retângulo (ver figura 1).

- Subprocessos - são distinguidos das tarefas através da colocação do símbolo “+” (mais) no centro inferior da forma (ver figura 1). Então é elaborado um diagrama à parte correspondente ao subprocesso. Algumas ferramentas de modelação também permitem expandir e contrair subprocessos, para que o contexto de todo o processo permaneça visível, assim como os detalhes do um subprocesso. Esta representação é feita através de um retângulo à volta de todo o subprocesso. A visão expandida de subprocessos só é razoável até um determinado tamanho de modelo. Caso contrário, os diagramas ficarão muito grandes e complexos e será difícil compreendê-los [19]. Podem ainda existir processos hierárquicos através de subprocessos contidos noutros subprocessos. Os eventos dos subprocessos podem ser implícitos os explícitos, não sendo obrigados a seguir a mesma representação do processo principal.
- Ciclos – Representa uma atividade que é repetida várias vezes num processo. Um ciclo é representado por uma seta circular na parte inferior do retângulo (ver figura 1). Os subprocessos também podem ser assinalados como ciclos, tanto na forma expandida como na forma contraída. Para que o ciclo não seja repetido indefinidamente deverá ser descrito com que frequência o ciclo será repetido, seja pela utilização de uma condição de saída, ou pela repetição do ciclo até que ou enquanto a condição definida seja verdadeira. As condições podem ser expressas através de anotações (ver subcapítulo 2.1.8). As condições devem ser escritas como declarações textuais de fácil compreensão.
- Multi-instância – atividade representada com três linhas paralelas na parte inferior do retângulo (ver figura 1) e é utilizada para lidar com uma coleção inteira de objetos que precisam de ser processados. Uma atividade multi-instância só faz sentido quando os dados da instância do processo contêm algum tipo de coleção, como itens de um pedido. A atividade multi-instância é realizada várias vezes, uma para cada objeto da coleção, onde o número de repetições é determinado pelo número de itens. A atividade multi-instância difere da atividade de ciclo pois o número de repetições é conhecido e na atividade de ciclo é necessário testar a condição de saída a cada repetição. Tal como nos ciclos podem ser utilizadas anotações.

As tarefas realizadas num processo podem ser divididas em diferentes tipos. A especificação BPMN define os seguintes tipos:

- Tarefa de serviço - Uma tarefa de serviço é uma função automatizada, por exemplo a chamada de uma função de uma aplicação ou serviço da *web*.
- Tarefa de receção - Uma tarefa de receção recebe uma mensagem. Corresponde a um evento de receção de mensagem.
- Tarefa de envio - Uma tarefa de envio envia uma mensagem. Corresponde a um evento de lançamento de mensagem.
- Tarefa do utilizador - Uma tarefa do utilizador espera por um *input* do utilizador. Este é um tipo de tarefa tipicamente chamada “fluxo de trabalho de interação humana”.
- Tarefa de regra de negócio - Numa tarefa de regra de negócios, uma ou mais regras de negócio são aplicadas para produzir um resultado ou para tomar uma decisão.
- Tarefa de *script* - Esta tarefa consiste num *script* que contém instruções que são processadas diretamente pelo mecanismo de processo.
- Tarefa manual - Uma tarefa manual é realizada sem suporte a tecnologias de informação.
- Tarefa abstrata - É uma tarefa sem tipo definido.

Com a exceção da tarefa abstrata, as diferentes tarefas são marcadas com ícones, como ilustrado na figura 2. Com a exceção da tarefa abstrata, a BPMN 2.0 apresenta marcadores no canto superior-esquerdo das tarefas que permitem definir quais destas são automatizadas ou manuais. O marcador de uma tarefa manual é representado pelo símbolo de uma “mão”, enquanto o marcador de tarefa de utilizador é representado por um símbolo de utilizador, sendo ambas consideradas tarefas manuais uma vez que é necessária a intervenção de um utilizador para a sua realização. As tarefas automatizadas são marcadas com 5 subtipos que representam a forma pela qual a automatização é realizada. Esses subtipos são: tarefa de *script*, nas quais o BPMS executa um código internamente; tarefa de serviço, são aquelas que o BPMS invoca um serviço externo para realizar a atividade; tarefa de regra de negócio, que é utilizada para aplicar uma ou mais regras do negócio; tarefa de envio, que envia uma mensagem para outra *pool* (interveniente de processo) do modelo; e tarefa de "receção", que recebe uma mensagem enviada por outra *pool* do modelo.

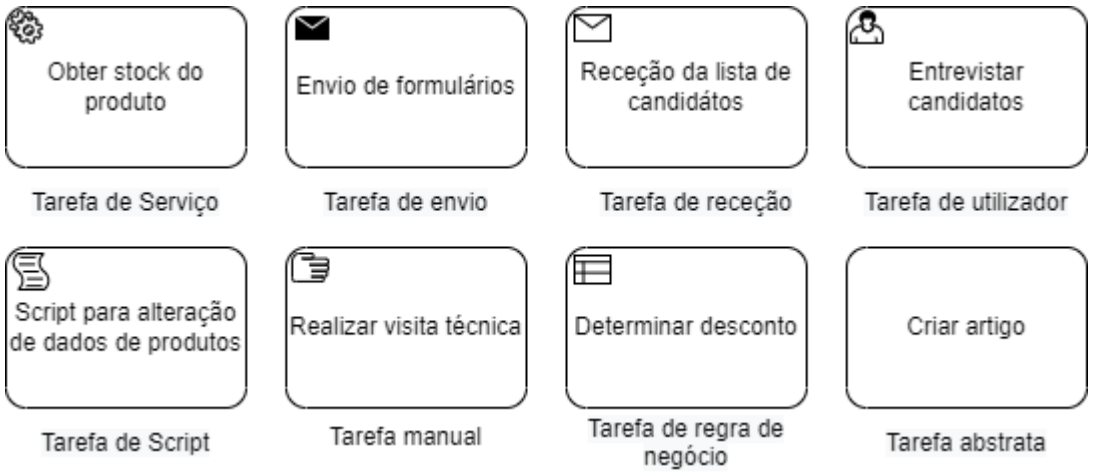

Figura 2 - Exemplos de tipos de tarefas

2.1.2 *Gateways*

Um *gateway* é utilizado para controlar a divergência e convergência de *Sequence Flows* e é representado por uma forma de diamante (Ver figura 3). Um *gateway* determina as decisões tradicionais, bem como a bifurcação, divisão e junção de caminhos. Os marcadores internos dos *gateway*s indicam o tipo de controlo de que é realizado. A utilização do *gateway* está tipicamente associada à tomada de decisões para avaliar o estado do processo de negócio e, com base na condição, delimita o fluxo num ou mais caminhos. Deve ter-se atenção ao utilizar um *gateway* para dividir ou juntar, pois não pode ser utilizado numa combinação de ambos. Existem quatro tipos de *gateways* mais utilizados: o *gateway* exclusivo, o *gateway* paralelo, o *gateway* inclusivo e o *gateway* complexo.

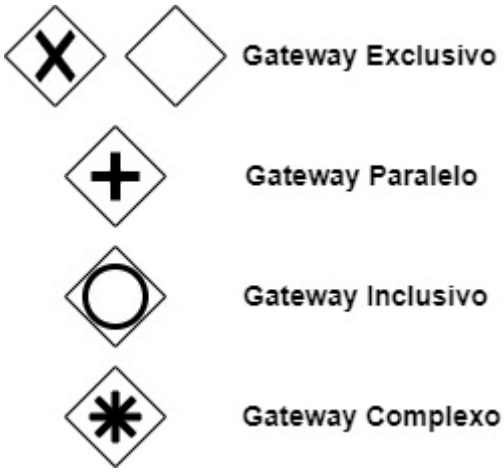

Figura 3 - Tipos de Gateway

Existem quatro tipos de *gateways* mais utilizados:

- *Gateway* exclusivo **-** O *gateway* exclusivo é utilizado para modelar caminhos alternativos e pode ser aplicado como uma divisão ou como uma união. Estes podem ser representados de duas formas: em branco ou com um "X" no seu interior (ver figura 4). É recomendada a utilização consistente de apenas uma das variantes [19]. Deve considerar-se que um *gateway* representa apenas lógica, ou seja, nenhuma atividade é realizada e nenhum tempo passa durante a execução de um *gateway*. Se for modelada uma atividade que deve tomar uma decisão, ela é representada como uma atividade seguida de um *gateway* exclusivo, com várias saídas que representam as várias opções de decisão. As condições são anotadas em texto simples nos fluxos de sequência existentes. Ou então a questão pode também ser colocada no símbolo do *gateway*, e os fluxos de sequência trazem apenas as respostas "sim" e "não". Se o texto for muito longo ou for utilizado o símbolo com "X", a pergunta também pode ser anotada ao lado do símbolo do *gateway*. Uma das saídas pode ser definida como saída padrão e apenas é selecionada se nenhuma condição das saídas restantes for verdadeira. Esta saída é marcada com uma pequena barra/traço diagonal.

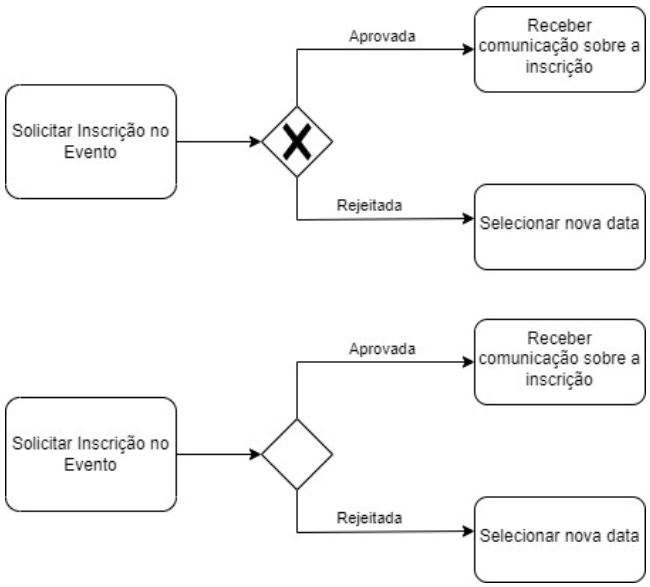

Figura 4 - Gateways exclusivos

- *Gateway paralelo* - Um *gateway* paralelo pode dividir um fluxo de sequência em dois ou mais caminhos paralelos, o que corresponde a um *And* (E) lógico. Os caminhos paralelos também são unidos por um *gateway* paralelo. Este *gateway* pode ser

utilizado para a realização de atividades simultaneamente. É representado por uma forma de diamante com um sinal de “+” no interior. A figura 5 demostra a utilização do *gateway* paralelo. O processo inicia-se pela escrita da notícia. De seguida, é utilizado o *gateway* que permite seguir dois caminhos distintos, que são as atividades “Publicar na página oficial” e “Publicar nas redes sociais” respetivamente.

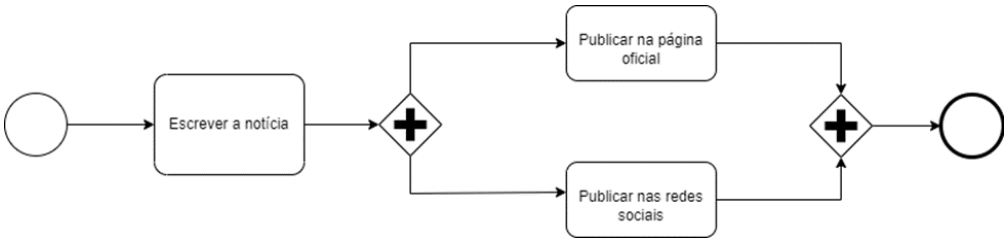

Figura 5 - Gateway paralelo

- *Gateway* inclusivo - Um *gateway* inclusivo seleciona ou junta um ou mais caminhos, onde qualquer combinação com pelo menos uma das opções é possível. Um *gateway* inclusivo, representa, portanto, a lógica de um *Or* (Ou) e é representado por uma forma de diamante com um círculo no interior. Assim como um *gateway* exclusivo, um *gateway* inclusivo também pode ter uma das suas saídas marcada com uma pequena barra diagonal como fluxo de sequência padrão, que será selecionado automaticamente caso nenhuma condição dos outros fluxos de sequência seja verdadeira. Isto garante a seleção de pelo menos um fluxo de sequência. Em contraste com os outros fluxos de sequência, o fluxo de sequência padrão não pode ser selecionado em combinação com outros fluxos de sequência, porque o padrão só será selecionado se nenhuma das outras condições se aplicar. A figura 6 ilustra a utilização do *gateway* inclusivo, onde um de dois caminhos deve ser escolhido. Depois da atividade “Determinar distância” segue-se o *gateway* inclusivo que existe como um Ou. Ou é escolhido ir pelo caminho da atividade “Ir de carro” se forem 300 km ou menos, ou pelo caminho da atividade “Ir de avião” se forem mais de 300 km de distância.

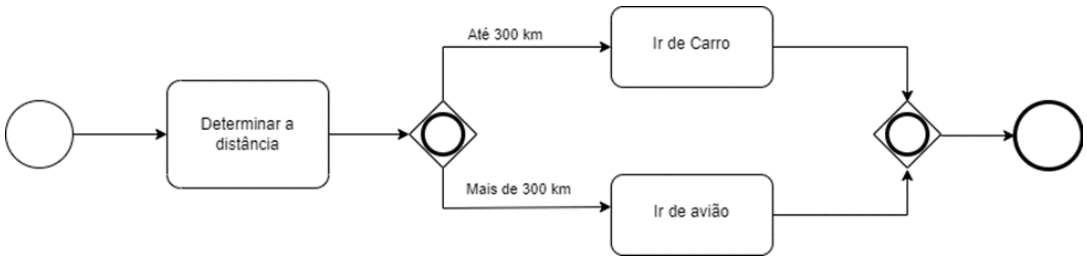

Figura 6 - Gateway inclusivo

- *Gateway* complexo - Um *gateway* complexo é representado na forma de um diamante com um asterisco. É utilizado para combinar e dividir fluxos de sequência e é bastante útil para representar fluxos de sequência complexos. Este tipo de *gateway* deve ser rotulado de forma clara e compreensível de modo a melhorar a legibilidade do modelo. Este *gateway* deve ser utilizado quando existe a necessidade de simplificar o processo, por exemplo, quando estão a ser utilizados muitos *gateways*, estes devem ser substituídos por um *gateway* complexo. A figura 7 ilustra uma utilização do *gateway* complexo tem início na tarefa “Solicitar questionários”. Seguidamente o fluxo de sequência do processo é dividido em três fluxos paralelos por um *gateway* paralelo, onde são solicitados três questionários, embora apenas dois dos questionários sejam necessários para a análise dos resultados. O *gateway* complexo é utilizado seguindo o conjunto de tarefas relacionadas com a gestão de questionários para que seja efetuada uma verificação especial, ou seja, quando 2 de 3 questionários forem entregues o processo continua e seguir-se-á a atividade “Analisar resultados”.

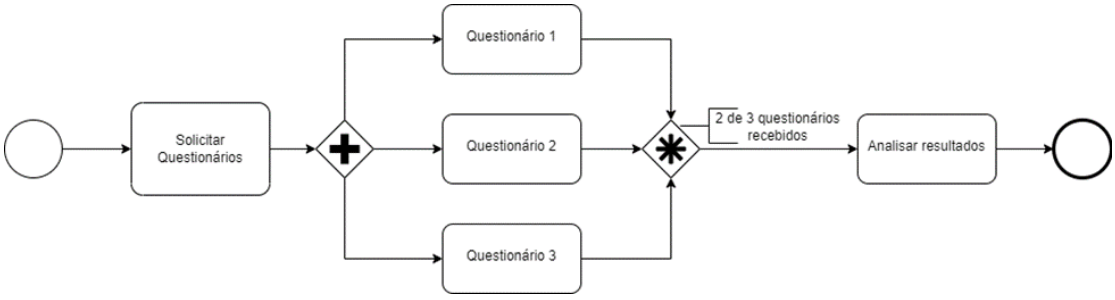

Figura 7 - Gateway Complexo

2.1.3 *Splitting* e *Merging* sem *gateways*

É possível conduzir vários fluxos de sequência para uma atividade sem a utilização de *gateways* levando a uma representação mais concisa.

- *Splitting* (Divisão) – Para modelar uma divisão inclusiva existem duas possibilidades, ou através de um *gateway* inclusivo, como referido anteriormente, ou através de fluxos de sequência condicionais onde a atividade apresenta várias saídas. Os fluxos de sequência condicional são representados por pequenos diamantes condicionais colocados no início da condição. Após concluída uma atividade são ativados os fluxos que possuem condições verdadeiras. A condição deve ser expressa de forma que pelo menos uma condição se aplique, assim sendo a lógica de representação é a mesma de um *gateway* inclusivo. Para modelar uma divisão exclusiva, além do *gateway* de divisão exclusivo podem ser utilizados fluxos de sequência condicionais que saem de uma atividade. As condições devem ser definidas de forma que seja sempre exatamente uma condição verdadeira. Quando as atividades são seguidas por fluxos de sequência condicional, um dos fluxos de sequência pode ser marcado como fluxo padrão, que será escolhido automaticamente se nenhuma condição dos outros fluxos de sequência se aplicar e é representado por uma barra.
- *Merging* (Junção) – A junção de caminhos alternativos também pode ser modelada sem a utilização de *gateways.* São utilizados os fluxos de sequência alternativa que neste caso vão diretamente para a próxima atividade.

As funções de *splitting* e *merging* com e sem *gateways* podem ser combinadas entre si. É possível, por exemplo, ter vários fluxos de sequência condicional a sair de uma atividade e posteriormente juntá-los com um *gateway* exclusivo. As divisões e junções correspondentes são mais fáceis de detetar se foram utilizados simetricamente os mesmos tipos de *gateway*, especialmente em casos mais complexos com vários *gateways* combinados, para ser mais fácil de compreender a lógica do fluxo [21].

Na figura 8 é possível verificar a possibilidade de modelação de caminhos com *gateway* e sem *gateway*, para tal é utilizado o fluxo de sequência condicional.

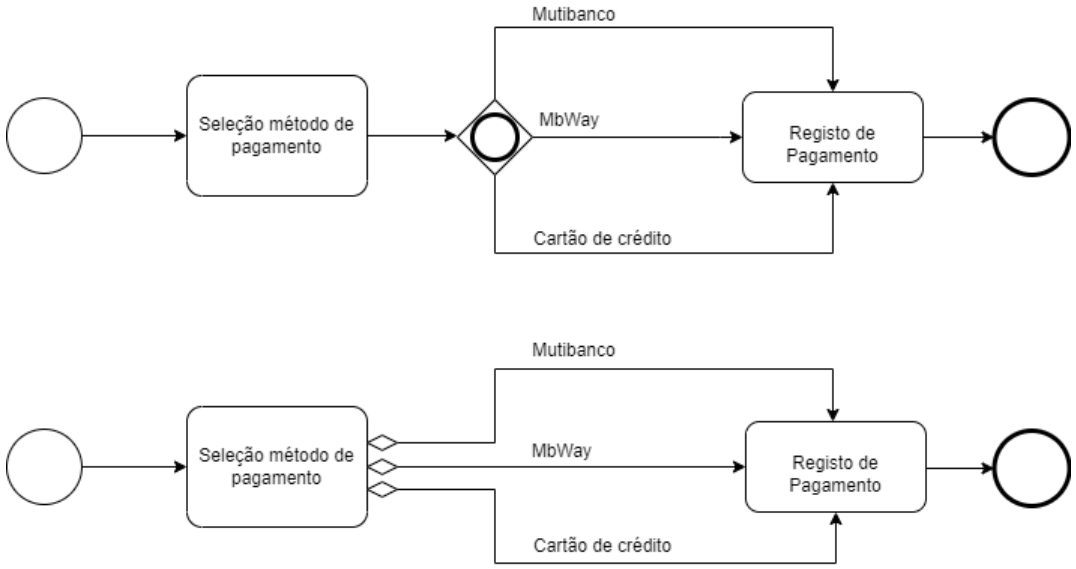

Figura 8 - Decisão com e sem utilização de gateway

2.1.4 Eventos

Um Evento é utilizado para expressar algo que acontece durante o curso de um processo de negócio e é representado por um círculo. Os eventos afetam o fluxo do processo e geralmente têm um *trigger* (gatilho) ou um resultado. Ao modelar um evento deve ter-se em conta dois aspetos que são: a causa e o seu efeito no processo. Eventos são círculos com centros abertos que possibilitam adicionar marcadores internos que diferenciem diferentes *triggers* e/ou resultados. Existem três tipos de eventos (ver figura 9):

Figura 9 - Tipos de eventos

- Evento de Início - Existem três tipos de eventos na medida em que afetam o fluxo: Início, Intermediário e Fim. O processo começa pelo evento de início que consiste num círculo simples com uma linha única no início da *lane*. Na maior parte dos casos é utilizado apenas um evento de início em cada processo. Relativamente aos eventos

de início, também é possível que ocorram a partir de um acontecimento específico, dessa forma devem ser representados pelo círculo com um símbolo no interior de forma a ilustrar o evento adequadamente. Por exemplo o término de uma tarefa pode indicar o início de um novo processo.

- Evento intermediário – Estes eventos são representados por um círculo com linha dupla. Caso seja necessário, os símbolos básicos apresentados podem ser complementados com ícones conforma a situação em que sejam utilizados, como por exemplo mensagens ou temporizadores. Os eventos intermédios podem ser utilizados para vários cenários, por exemplo:
    - O envio ou receção de uma mensagem, uma carta, ou um email;
    - Quando é alcançado um certo período, ou quando termina um certo período;
    - Uma condição tornar-se verdadeira;
    - Quando ocorre um erro.
- Evento de fim - Representa o último evento do processo, que consiste num círculo semelhante ao evento inicial, mas com uma borda mais espessa. Além do evento de fim sem tipo, também podem ser adicionados símbolos, assim como os eventos de início e os intermediários, por exemplo a colocação do símbolo de mensagem no evento de fim indica que o processo termina e é enviada uma mensagem. Podem existir vários eventos de fim num processo, onde cada um termina o fluxo de sequência na instância em que é colocado. Se um desses eventos for do tipo evento de fim com o círculo preenchido no interior, todos os fluxos de sequência são terminados e o processo será finalizado [23].

2.1.5 *Connecting Objects*

Existem três objetos principais de *Connecting Objects* (ver figura 10):

- *Sequence Flow* (Fluxo de Sequência) - utilizado para mostrar a ordem ou sequência pela qual as atividades são executadas num processo e é representado por uma linha sólida com uma ponta de seta sólida.
- *Message Flow* (Fluxo de mensagens) - utilizado para mostrar a troca de mensagens entre dois participantes do processo separados por entidades, ou por funções de negócio, que as enviam e que as recebem, e é representado por uma linha tracejada com uma ponta de seta aberta. Um fluxo de mensagens representa qualquer tipo de

troca de informações, por exemplo, um email, uma carta, uma chamada telefónica e qualquer tipo de troca de dados eletrónica.

- *Association* (Associação) - utilizada para associar dados, texto e outros artefactos a objetos de fluxo e é representada por uma linha pontilhada com uma seta de linha, ou apenas pela linha pontilhada. As associações são utilizadas para mostrar as entradas e saídas das atividades [24].

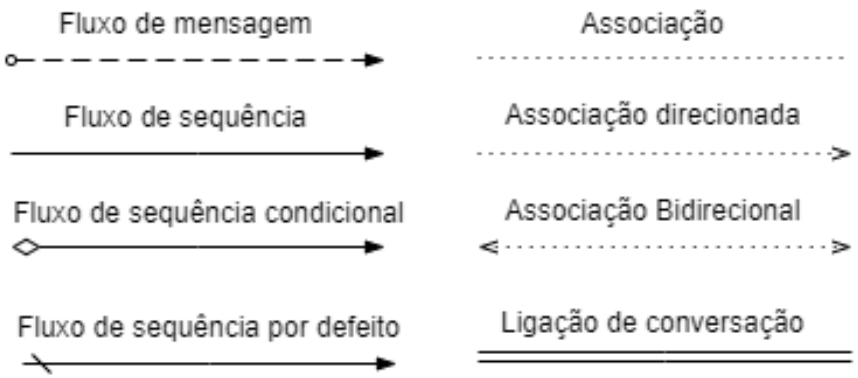

Figura 10 - Objetos de Fluxo

2.1.6 *Swimlanes*

Todos os processos BPMN com mais de um participante são desenvolvidos dentro de uma *pool* e rotulados com o nome do processo. Os processos de participante único por vezes não têm a pool representada visualmente pois não existe a necessidade de representar interação entre participantes. Os objetos *swimlanes* são elementos estruturantes do BPMN que permitem diferenciar as seções de um diagrama BPMN. São caixas retangulares que representam os participantes de um processo de negócio. Cada *swimlane* pode conter objetos de fluxo que são executados por esse participante, e podem ser dispostas horizontalmente ou verticalmente (ver figura 11). Para *swimlanes* horizontais, o processo flui da esquerda para a direita, enquanto o processo nas *swimlanes* verticais flui de cima para baixo. Existem dois tipos de *swimlanes*:

- *Pools* - representam os participantes de um processo de negócio. Pode ser uma entidade específica, por exemplo um departamento, ou uma função por exemplo, um gerente, um assistente, um estudante, um fornecedor. A utilização de vários *pools* é particularmente interessante quando estamos a modelar uma colaboração, que

consiste na interação dos processos de vários parceiros, em que o processo de cada parceiro é representado num *pool* separado (ver subcapítulo 2.2.1) [19].

- *Lanes* – são sub-partições das *pools* e são utilizadas para organizar e categorizar as atividades. Por exemplo, para a pool estudante podes existir as *lanes*: horário, disciplina e sala. Assim como a *pool* é possível utilizar as *lanes* para representar entidades ou funções específicas que estão envolvidas num processo. A utilização de vários pools é particularmente interessante quando estamos a modelar uma colaboração, que consiste na interação dos processos de vários parceiros, em que o processo de cada parceiro é representado num *pool* separado (ver subcapítulo 2.2.1) [19]

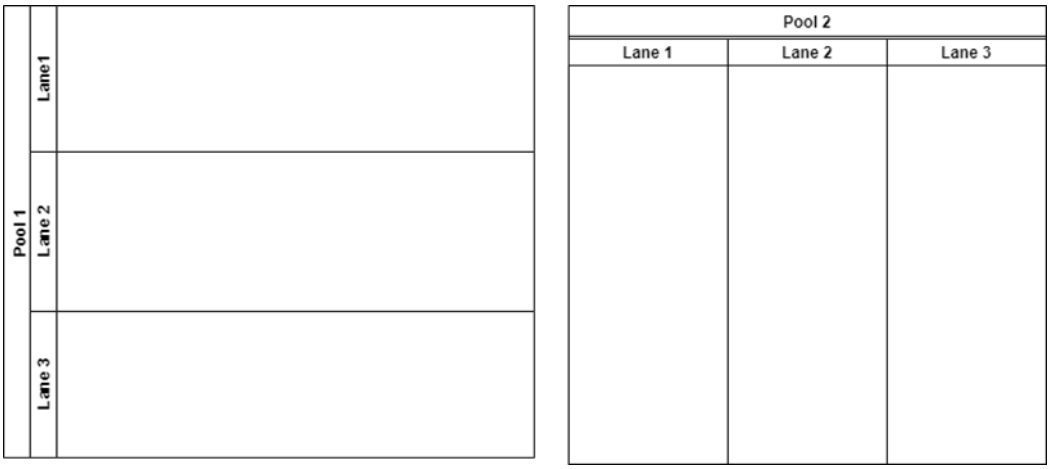

Figura 11 - Pool horizontal e pool vertical

2.1.7 Artefactos

Os modelos BPMN são focados em fluxos de sequência e de mensagem e em troca de dados. Se existirem outros aspetos relevantes para um processo de negócio que precisem de ser mapeados, é possível recorrer à utilização de artefactos (ver figura 12).

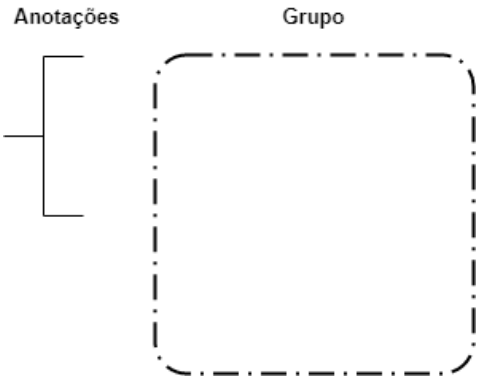

Figura 12 - Artefactos

A especificação BPMN define dois tipos padrão de artefactos:

- Anotações - utilizada para adicionar explicações e/ou comentários a um determinado elemento do modelo de forma a aumentar a compreensão de um modelo e não tem nenhum efeito na lógica do fluxo de controlo do modelo. Por outro lado, as anotações podem ser utilizadas para documentar as condições de saída de um ciclo, por exemplo, onde as próprias condições de saída têm naturalmente um efeito na lógica do fluxo de controle do processo. Em BPMN, as condições de saída são definidas, por valores de atributo dos respetivos elementos de modelação, o que faz com que a apresentação no modelo por anotações seja apenas para fins de documentação. Uma anotação é representada por um parêntese reto que pode ser ligado a qualquer elemento num modelo BPMN através de uma linha, por exemplo, para adicionar um comentário ou explicação ao elemento em questão. Anotações são um mecanismo pelo qual o modelador fornece informações de texto adicionais para o leitor de um diagrama BPMN. É utilizado muitas vezes para mostrar as entradas e saídas das atividades no processo. No entanto, a estrutura básica do processo, conforme determinado pelas Atividades, *Gateways* e Fluxo de Sequência, não é alterada com a adição de Artefactos ao diagrama.
- Grupo - desenhado como um retângulo arredondado em que a borda consiste numa linha de pontos e traços. Os pontos e traços devem ser claramente visíveis, para não ser confundido com um subprocesso. Como todos os artefactos, um grupo é puramente um objeto gráfico sem qualquer relevância para a lógica de um diagrama BPMN. Portanto, é possível utilizá-lo sem quaisquer restrições, para destacar partes interessantes de um modelo, ou para agrupar elementos que se relacionam entre si.

Também é permitido desenhar grupos além das margens de *pools* e *lanes*. Como um elemento puramente gráfico, um grupo não pode ser a origem ou o destino de uma sequência ou fluxo de mensagens. Os fluxos podem atravessar as fronteiras do grupo conforme desejado.

Os modeladores e os fornecedores de ferramentas têm liberdade para definir seus próprios artefactos, complementando assim o BPMN com construções necessárias.

2.1.8 Objetos de dados

A BPMN foi projetado para permitir aos modeladores e ferramentas de modelação alguma flexibilidade em estender a notação básica e fornecer a capacidade de adicionar ao contexto apropriado, ou seja, a uma situação de modelação específica. Os objetos de dados são um mecanismo que permite mostrar de que forma os dados são necessários ou produzidos pelas atividades. Eles são ligados às atividades através de associações [24]. Geralmente durante um processo são utilizados e criados dados, informações, ficheiros, documentos, etc. Durante o fluxo sequencial que existe entre atividades são frequentes a transferência de dados. Se os dados criados ou utilizados forem ficheiros ou documentos são representados em BPMN por um “Objeto de dados”, para modelar dados persistentes, deve ser utilizado um objeto “Base de Dados” (ver figura 13).

Figura 13 - Objetos de Dados

Geralmente durante um processo são utilizados e criados dados, informações, ficheiros, documentos, etc. Durante o fluxo sequencial que existe entre uma atividade e outra, ele é frequentemente acompanhado pela transferência de dados. Se os dados criados ou utilizados forem ficheiros ou documentos são representados em BPMN por um “Objeto de dados”, para modelar dados persistentes, deve ser utilizado um objeto “Base de Dados”.

2.1.9 Exceções

Capturar e documentar casos especiais e exceções ao fluxo normal do processo é complicado e trabalhoso embora o BPMN contenha várias construções específicas para a modelação e o tratamento de exceções (ver figura 14).

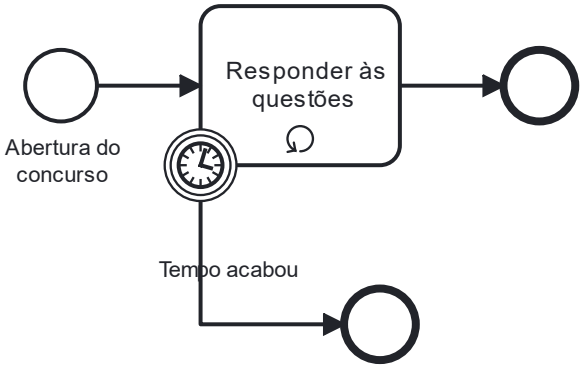

Figura 14 - Uma atividade de ciclo é abortada por um evento intermediário

A especificação BPMN define as seguintes construções:

- Eventos intermediários - Durante a realização de uma atividade, pode ocorrer um evento que resulta num termino precoce dessa atividade. Um exemplo desse evento é o temporizador, ou seja, quando o tempo termina a atividade é interrompida. Para modelar este género de exceções é colocado um evento de interrupção no canto inferior esquerdo da atividade pretendida, ou seja, um círculo de borda dupla, com o símbolo correspondente ilustrado no interior. Como este evento ocorre durante a execução do processo, é considerado um evento intermediário. A atividade passa então a ter mais um fluxo de sequência que se inicia no evento intermediário. O evento intermediário só tem efeito se ocorrer enquanto a respetiva atividade está a ser executada, caso ocorra antes ou depois será ignorado. Quando é necessário reagir a um evento, mas continuar a atividade em andamento é utilizado um evento intermediário sem interrupção. Um dos exemplos da utilização deste evento é o envio de notificações. A modelação deste evento em BPMN assemelha-se ao anterior, com a exceção da linha dupla do círculo do evento ser desenhada em tracejado. O evento de envio de mensagem é também um exemplo de uma variante sem interrupção, onde a mensagem é processada em paralelo com o decorrer das restantes atividades. Os eventos intermediários sem interrupção podem ser anexados não apenas a tarefas,

mas também a subprocessos. Além dos eventos de temporizador e de mensagem, outros tipos de eventos intermediários podem ser anexados às atividades: eventos de sinal, eventos condicionais, eventos múltiplos e eventos múltiplos paralelos. Cada um deles pode ser utilizado como um evento de interrupção ou não interrupção [19].

- Tratamento de Erros - O tratamento de exceções é frequentemente necessário devido a erros. Para isso, podem ser utilizados eventos intermediários do tipo “erro”, que são representados pelo símbolo de um “raio”. Em contraste com os tipos anteriores de eventos intermediários, os eventos intermediários de erro não podem ser utilizados em fluxos de sequência normal. Eles só podem ser anexados aos limites da atividade. Como nos outros tipos de eventos, o ícone preenchido representa um evento de lançamento, enquanto um evento de captura é representado por um ícone em branco. Os eventos de erro de lançamento são sempre eventos finais, porque abortam completamente o subprocesso circundante. Os eventos de captura de erro, por outro lado, são sempre eventos intermediários que só podem ser anexados aos limites de tarefas ou subprocessos [19].
- Eventos de escalonamento - Os eventos de escalonamento são semelhantes aos eventos de erro. Embora os eventos de erro sejam utilizados principalmente para problemas técnicos, os eventos de escalonamento representam principalmente problemas num nível de negócios, por exemplo, se uma tarefa não for concluída, uma meta não for atingida ou um acordo necessário não for alcançado. Este tipo de eventos pode ser utilizado em subprocessos. Um escalonamento é definido por um evento intermediário de escalonamento, anexado à borda do subprocesso e este evento intermediário aciona um fluxo de exceção. Em contraste com os erros que terminam sempre a atividade, os escalonamentos podem ser representados em ambas as variantes: o evento de interrupção e o evento de não interrupção. Os eventos de escalonamento sem interrupção são considerados o padrão. Se um escalonamento for necessário para abortar toda a atividade, os limites do evento intermediário anexado são desenhados com linhas sólidas. Nesse caso, o evento de escalonamento de lançamento no subprocesso deve ser um evento final. Se um evento intermediário de escalonamento de lançamento fosse utilizado, o fluxo de sequência subsequente nunca poderia ocorrer porque o subprocesso seria abortado pelo escalonamento [19].
- Transações e compensações - O termo “transação” é utilizado em diferentes contextos. Por exemplo, existem transações comerciais, como uma transferência de

dinheiro de uma conta bancária para outra. As transações de bases de dados relacionais são bem conhecidas. Os diferentes tipos de transações têm em comum o facto de serem unidades completas de trabalho. A forma de modelar as transações através de BPMN é desenhar a borda do subprocesso com uma linha dupla, marcando assim o subprocesso como uma transação. Uma transação deve ser realizada na sua totalidade ou não ser sequer realizada. Se ocorrer algo de errado, a transação deve ser revertida, isso significa que a interrupção de uma transação reverte automaticamente os efeitos das atividades que já foram executadas. Para permitir isso, uma atividade de compensação é atribuída a cada atividade que requer compensação. O símbolo de retrocesso (duas pontas de seta), identifica a tarefa ou o subprocesso como atividades de compensação. Os eventos intermediários de captura do tipo de compensação também são utilizados para essa finalidade. Estes eventos estão sempre ligados a uma atividade e têm sempre associações de compensação de saída em execução para a atividade de compensação, essa associação de compensação é representada por uma linha tracejada com uma seta. As compensações também podem ser modeladas sem transações. Para isso, podem ser utilizados eventos finais de compensação ou intermediários que contêm símbolos de retrocesso preenchidos [19].

2.2 Tipos de Diagramas

Uma das características mais importantes da notação BPMN é a sua expressividade na representação de processos. Esta característica possibilita a representação dos processos de várias formas diferentes, adaptando-se ao contexto e ao propósito em que se está a ser utilizada. Para isso o BPMN contempla três tipos de diagramas.

O diagrama de processo ou colaboração é o tipo de diagrama utilizado com mais frequência, o que faz com que algumas ferramentas e bibliografia de BPMN apenas contemplem este tipo de diagrama. Embora seja sem dúvida o tipo mais importante, existem também áreas de aplicação úteis para os outros tipos de diagramas [19].

2.2.1 Diagrama de processo ou colaboração

Neste tipo de diagrama, pode ser modelado o fluxo do processo, incluindo atividades, divisões, fluxos paralelos, etc. Também é possível mostrar a colaboração entre dois ou mais

processos com mensagens trocadas entre eles. Os diagramas de processo e os diagramas de colaboração são do mesmo tipo de diagrama, a diferença é que um diagrama com apenas um processo é frequentemente chamado de diagrama de processo (ver figura 15), enquanto um diagrama com vários processos interagindo entre eles é um diagrama de colaboração (ver figura 16).

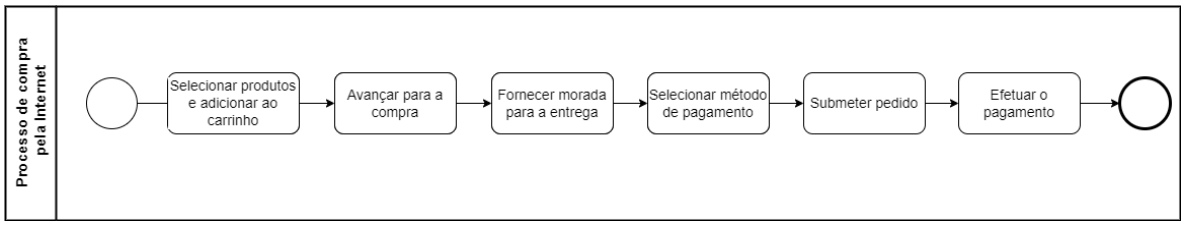

Figura 15 - Exemplo de diagrama de processo

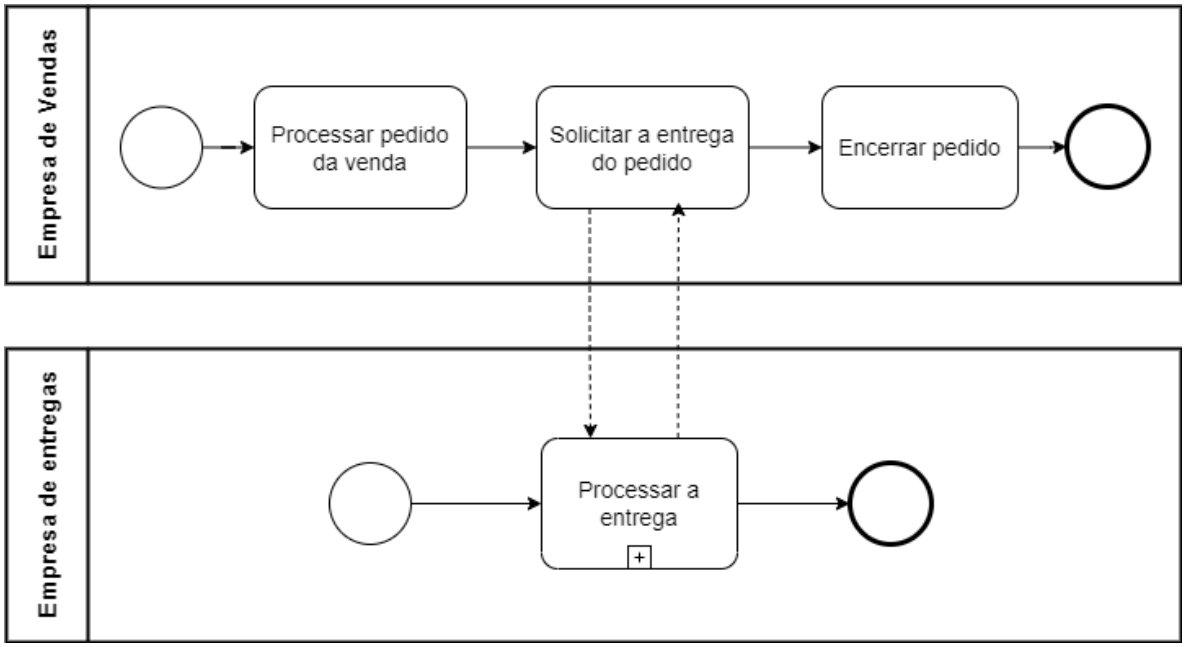

Figura 16 - Exemplo de diagrama de colaboração

Os diagramas de colaboração são especialmente úteis para documentar a interação entre diferentes processos. Por exemplo, o processo de vendas e o processo de entregas são dois processos independentes e de empresas distintas, no entanto estão interligados, pois no

processo de vendas é enviado um pedido que aciona o processo de entregas, existindo assim uma comunicação entre ambos. As colaborações são modeladas com dois ou mais *pools*, onde cada *pool* contém um processo separado e é também modelada a comunicação entre eles através dos fluxos de mensagens.

2.2.2 Diagrama de coreografia

Neste tipo de diagrama são modeladas as interações entre diferentes processos. No entanto, cada troca de dados é modelada como uma atividade. É possível visualizar divisões, ciclos, etc. para representar processos de troca complexos. Os diagramas de coreografia fornecem outra possibilidade para modelar a sequência temporal e lógica dos fluxos de mensagens. Nesses diagramas, o foco está nas próprias trocas de mensagens que são modeladas como atividades coreográficas. Uma atividade de coreografia representa a troca de uma ou mais mensagens entre dois ou mais parceiros. Nos casos mais simples, consiste em enviar apenas uma mensagem de um parceiro para outro. Cada atividade é iniciada por um dos parceiros, que envia a primeira mensagem. Alguns elementos de modelação de processos não são significativos em diagramas de coreografia, portanto, eles não são permitidos nestes diagramas. Por exemplo, não existem eventos de mensagem dentro do fluxo de sequência normal, porque as trocas de mensagens estão contidas nas atividades de coreografia por definição. Da mesma forma, os *gateways* baseados em eventos não são seguidos por eventos, mas por atividades coreográficas, o que significa que o caminho é selecionado pela atividade de coreografia que é iniciada pela sua mensagem inicial.

Tal como exemplificado na figura 17, para serem conhecidas as mensagens que são trocadas em cada atividade de coreografia, podem ser adicionados pequenos símbolos de envelope e ligados à borda do respetivo parceiro. Os envelopes devem ter as mesmas cores das bordas parceiras. Um envelope branco significa uma mensagem que inicia uma atividade coreográfica. Os símbolos do envelope das mensagens do outro participante são sombreados.

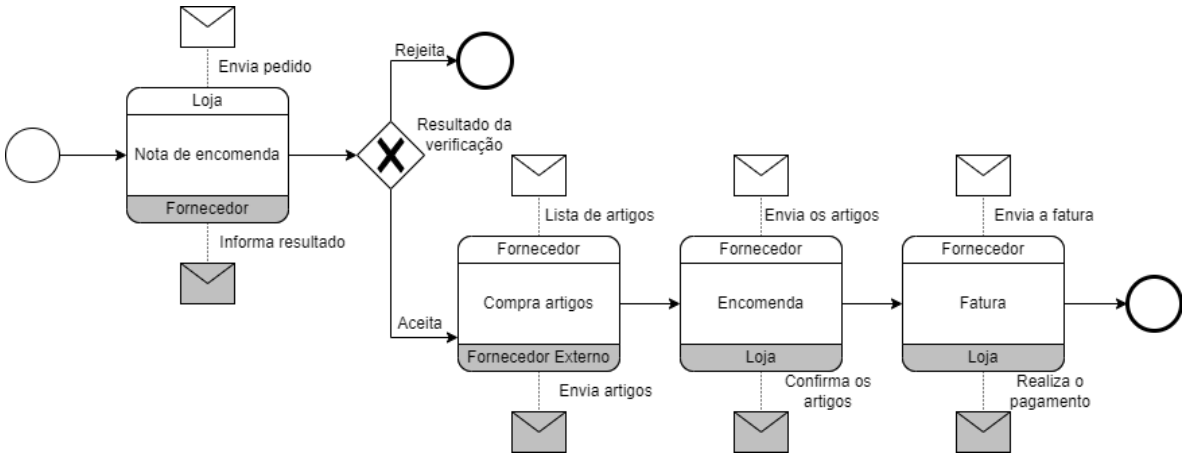

Figura 17 - Diagrama de coreografia - Exemplo

2.2.3 Diagrama de conversação

Um diagrama de conversação é uma visão geral de vários parceiros e das suas inter-relações, ou seja, quais parceiros de um determinado domínio cooperam em as tarefas associadas. Existe uma conversação entre os intervenientes que é realizada por uma série de fluxos de mensagens. Os detalhes podem ser modelados, através de num diagrama de coreografia ou um diagrama de colaboração e é também possível combinar os fluxos de mensagens de duas ou mais conversações num diagrama. A ligação entre uma conversação e um participante é chamada de *link* de conversação. Uma conversação está sempre ligada a dois ou mais participantes (ver figura 18).

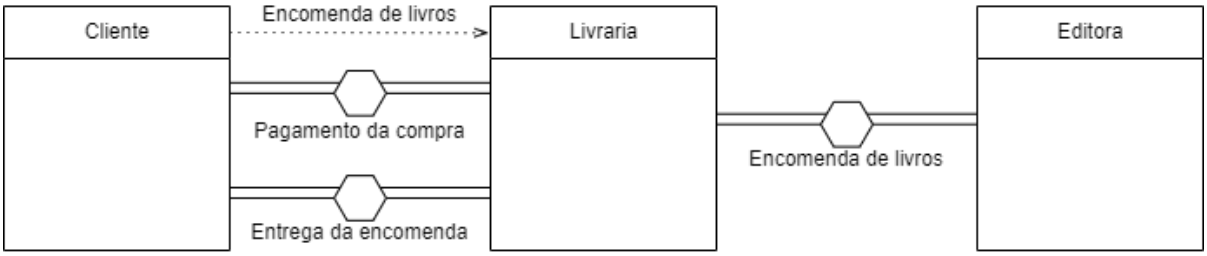

Figura 18 - Diagrama de conversação - Exemplo

2.3 Exemplo

Na figura 18 está ilustrado um exemplo de um modelo que colaboração de um concurso televisivo, onde existem três processos: o processo de inscrição, o processo de reserva de sala e o processo de avaliação.

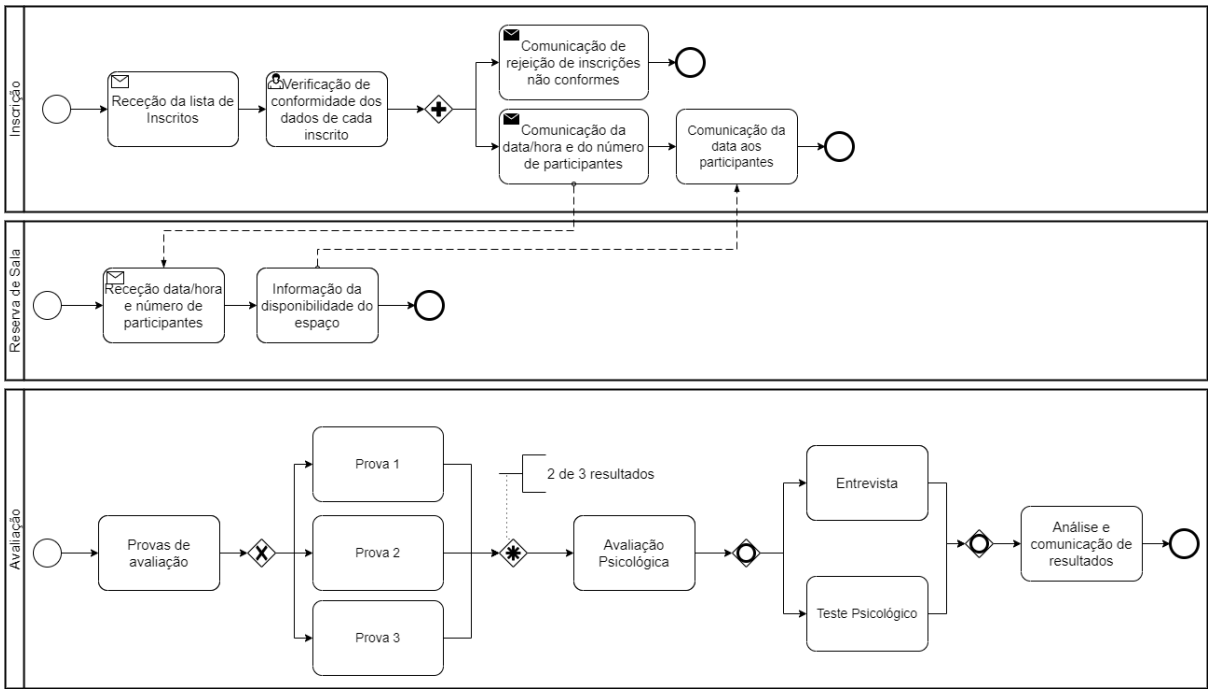

Figura 19 - Diagrama de processo/colaboração de um concurso

Os processos representados são:

- O processo de "Inscrição" inicia-se pela tarefa de receção "Receção da lista de inscritos". Na segunda tarefa deste processo, a "Validação dos participantes" são verificados dos dados dos utilizadores inscritos classificando-os como rejeitados ou aceites para a participação no concurso. Nesta tarefa está ilustrado também o fluxo de dados para a base de dados "Participantes", onde são registados os participantes do concurso, que são os utilizadores classificados como aceites. O processo segue-se com um *gateway* paralelo que é utilizado para dividir o fluxo de sequência em dois caminhos paralelos, ou seja duas tarefas são executadas simultaneamente. São as tarefas "Comunicação de rejeição de inscrições não conformes", onde os inscritos não selecionados para participação no concurso serão informados da sua rejeição e a tarefa "Reserva de sala" onde este processo comunica com o processo "Reserva de

Sala" para receber a informação da data disponível para o concurso. Após o recebimento da informação, tem lugar a tarefa "Comunicação da data aos participantes", que se segue por um *gateway* paralelo e do evento de fim que termina o processo.

- O processo "Reserva de Sala" inicia-se da mesma forma que o processo anterior. A tarefa "Receção da data/hora e número de participantes" é uma tarefa de receção e corresponde à receção do número de participantes. A segunda tarefa é uma tarefa de serviço "Consulta de salas disponíveis" e está ligada pelo fluxo de dados à base de dados, onde são consultadas as salas disponíveis. Seguidamente temos um *gateway* paralelo que divide o fluxo em dois caminhos paralelos para as tarefas "Envio de confirmação da reserva" e "Reserva da Sala".
- O processo de "Avaliação" inicia-se pela atividade "Verificação dados do participante" através da consulta à base de dados correspondente. Segue-se pela atividade "Admissão às provas de avaliação", de seguida temos um *gateway* exclusivo e um *gateway* complexo que são utilizados para modelar caminhos alternativos, ou seja, cada participante terá de ter aprovação a 2 de 3 provas. Posteriormente temos a tarefa "Admissão à prova de avaliação psicológica" que consiste no apuramento dos candidatos que passam à fase seguinte. Depois dessa tarefa temos um *gateway* inclusivo que permite optar por um dos dois caminhos disponíveis consoante a validação anterior. Se o candidato obteve uma média igual ou superior a 75% será direcionado para a tarefa "Entrevista" que consiste numa tarefa manual, ou seja, será necessária a intervenção de uma pessoa para realizar essa tarefa. Se o candidato obteve uma avaliação inferior a 75% será direcionado para a tarefa "Teste Psicológico" que consiste numa tarefa de serviço, pois é um processo automatizado. A última tarefa deste processo consiste na tarefa "Análise e comunicação de resultados", que consiste em comunicar os resultados aos participantes.

2.4 Ferramentas

Para apoiar a modelação de processos de negócio, existem os chamados *Business* BPMS, que são um tipo especial de sistema que permite automatizar um processo de negócio. Os BPMS podem ser utilizados para a definição, execução e gestão dos processos de desenvolvimento de software, proporcionando melhorias nos processos. Estas ferramentas podem ser consideradas sistemas de gestão de fluxos de trabalho. Um fluxo de trabalho pode

ser definido como a automatização de um processo de negócio, ou parte dele, durante o qual documentos, informações e tarefas são enviadas de um participante para outro através de ações.

A crescente popularidade do padrão BPMN dá origem a um aumento do número de ferramentas que suportam a modelação de processos nesta notação. A existência de padrões como o BPMN não garante que todas as ferramentas possuam suporte para a implementação desse padrão. Além desta questão existem diferenças no que diz respeito ao *design*, aos recursos de armazenamento, às políticas de licenciamento entre outros aspetos. Perante essa grande oferta existente torna-se difícil entender as suas diferenças funcionais sem as experimentar.

Existem diversas ferramentas disponíveis no mercado que permitem a modelação de processos BPMN das quais podemos destacar:

- BPMM.IO[3] - Para desenhar processos de forma intuitiva e simples, o BPMN.io é uma excelente ferramenta. Foi desenvolvido por uma empresa alemã e uma das suas maiores vantagens é o fato de ser extremamente leve. Baseado totalmente na *web*, esta ferramenta funciona diretamente no *browser* e não é necessário efetuar nenhum tipo de download. Permite criar diagramas gráficos, sem muita informação adicional, pois não possui recursos complementares. Os diagramas criados podem ser transferidos para o computador no formato padrão BPMN, que posteriormente pode ser recarregado e editado e também no formato de imagem *Portable Network Graphics* (PNG).
- DRAW.IO[4] - O Draw.io é uma ferramenta baseada em nuvem que permite criar qualquer tipo de diagrama. Esta ferramenta é do mesmo tipo que o BPMN.io, um pouco mais rica nas funcionalidades visuais, pois permite utilizar cores, mas com menos recursos específicos para a modelação, como a ausência de validação. Os diagramas criados nesta ferramenta são puramente desenhos vetoriais, não há nenhum tipo de validação ou verificação se a modelação está correta.

[3] https://bpmn.io/

[4] https://draw.io/

- Aris express[5] - O ARIS Express é uma versão *light* da plataforma de análise de processos, o ARIS *Platform*. Foi criada originalmente como parte da plataforma que desenvolveu a notação *Event-driven Process Chain* (EPC), teve de adequar-se à nova realidade do mercado e incorporar ao seu conjunto de diagramas o BPMN. A execução da aplicação é efetuada no computador local assim como o armazenamento dos ficheiros. Suporta os elementos principais do BPMN, como todos os tipos de tarefas existentes, os diversos tipos de *gateway* e vários tipos de fluxos. Relativamente à parte do desenho é possível referir que o Aris Express é uma ferramenta intuitiva e de fácil aprendizagem.
- HEFLO[6] - O HEFLO possui uma *interface* leve e agradável e é *online* e o que permite criar diagramas de processo sem necessitar de instalar nenhuma aplicação. O editor é totalmente *web*, possui uma interface bastante fácil de utilizar e funciona diretamente no browser do computador. É muito fácil criar diagramas pois possui uma grande aderência à notação BPMN para o diagrama de processos e de orquestração. Os diagramas criados são gravados na nuvem própria do produto. Além disso, é possível exportá-los para o formato padrão BPMN, imagem PNG ou de documentação, como o *Portable Document Format* (PDF).
- Visio[7] - O Visio inclui um modelo que contém os elementos gráficos descritos pela especificação BPMN 2.0. É um *software* que permite desenhar uma grande variedade de diagramas, que incluem fluxogramas, projetos, plantas, diagramas de fluxo de dados, diagramas de fluxo de processos, modelação de processos de negócio, mapas 3D e muitos outros. É um produto da Microsoft, vendido como um complemento ao Microsoft Office. Permite a validação do diagrama, ou seja, verificar se existem problemas no diagrama de acordo com as regras de validação atuais. A validação é utilizada para garantir que os diagramas são construídos em conformidade com os procedimentos gerais. Possibilita a modelação dos elementos gráficos fundamentais da notação BPMN, como as tarefas especificas, os diversos tipos de *gateway* e vários tipos de fluxos. Permite a exportação do diagrama no formato compatível com o

[5] https://www.ariscommunity.com/aris-express

[6] https://www.heflo.com/pt-br/

[7] https://www.microsoft.com/pt-pt/microsoft-365/visio/flowchart-software

Microsoft Word, PDF e formatos de imagem. Tem como desvantagem o facto de na versão *Web* não dar suporte à criação de diagramas BPMN. O Microsoft Visio é uma ferramenta de desenho, mais especificamente para a criação de diagramas. É possível desenhar processos com o Visio, porém não se trata de uma ferramenta específica para isso e não pode ser considerada um BPMS.

As ferramentas referidas permitem construir fluxos de trabalho com a notação BPMN, através das quais é possível representar de forma padronizada e intuitiva as características do processo, por exemplo as suas atividades, a ordem de execução e os intervenientes. Estas ferramentas de modelação permitem descrever tarefas genéricas e a maior parte não suporta todos os artefactos BPMN enquanto, ferramentas como o BizAgi[8], e já obrigam a utilizar tarefas mais concretas, como por exemplo, a tarefa *service task*. Estas ferramentas vão além do desenho, ou seja, permitem a execução os processos. Este subcapítulo visa fornecer um estudo comparativo desta categoria de ferramentas, assim como das suas características.

A definição dos *softwares* a serem avaliados foi qualificada através de uma pesquisa sobre os *softwares* mais utilizados que implementam a notação BPMN, que disponibilizam uma versão académica ou trial para possibilitar a respetiva avaliação. A partir desta lista foram consultadas todas as páginas *web* dos respetivos fabricantes de forma a obter os ficheiros de instalação dos *softwares* e a utilização no *browser* daqueles que não requerem instalação em máquina local.

As ferramentas classificadas que consideram a perspetiva de execução, ou seja, que as permitem que o processo de modelação seja operacionalizado foram:

- Bizagi - O Bizagi é uma solução de gestão de processos de negócio desenvolvido pela empresa Bizagi que permite aos utilizadores modelar e executar processos de negócio através de um ambiente gráfico. É uma plataforma constituída por três componentes: Modeler, Studio e Automation. Cada um dos componentes facilita uma etapa fundamental na transformação e automatização dos processos. O Bizagi Modeler foca-se na parte do desenho. Serve para modelar, documentar, simular e otimizar os processos e é um componente gratuito ao contrário do Bizagi Studio e

[8] https://www.bizagi.com/pt

Bizagi Automation. Permite a importação e exportação dos diagramas nos formatos de imagem, Visio, XPDL, BPMN e XML. O Bizagi Studio serve para automatizar os processos através de pouca codificação é uma a ferramenta de desenvolvimento de processos. Os processos modelados no Bizagi Modeler são complementados com mais informações, ou seja, são criados formulários que apoiam o processo, assim como tudo o que for necessário desenvolver para que o processo seja automatizado é realizado no Bizagi Studio. O Bizagi Automation permite orquestrar processos e construir interfaces personalizadas. É ainda importante referir em relação ao Bizagi Studio que além das características mencionadas acima é uma aplicação *desktop* que permite a validação do diagrama, ou seja, verificar se existem erros de modelação. Suporta os elementos fundamentais do BPMN, como os 8 tipos de tarefas especificas, os diversos tipos de *gateway* e vários tipos de fluxos.

- Modelio[9] - Esta ferramenta de modelação de processos é um software do tipo *open-source* que requer instalação numa máquina local, cujo objetivo original seria a modelação de diagramas UML, mas que foi estendida para criar diagramas de processos em BPMN. Quem a utiliza tem a sensação de estar a trabalhar numa ferramenta de programador, pois foi desenvolvida com base no Eclipse. Esta ferramenta possibilita a elaboração de diagramas de processos BPMN e diagramas de colaboração BPMN. Através da análise da ferramenta não foi possível observar a modelação de diferentes tipos de tarefas, diferentes tipos de fluxos e diferentes tipos de *gateway*s, apenas a modelação dos elementos básicos do BPMN, sem definir os tipos de elementos. Embora a ferramenta permita a verificação de erros do modelo, em termos da notação, tem outras desvantagens pelo facto de que criar um diagrama ser trabalhoso e pouco produtivo.
- Sydle[10] - Esta ferramenta brasileira além de ser um editor de diagramas em BPMN é uma suíte para gestão de processos, logo possui outras funcionalidades que vão além da modelação do processo. O modelador é *web*, portanto não é necessário baixar ou instalar nenhum programa. Para criar modelos no Sydle é necessário criar uma conta, que pode ser na versão gratuita (*Community*) da ferramenta. A utilização de outras funcionalidades além da modelação pode exigir a escolha de uma licença paga. O

[9] https://www.modelio.org/

[10] https://www.sydle.com/br/

foco principal da ferramenta é a modelação de processos para serem automatizados através do próprio produto e estão disponíveis para a modelação apenas os elementos de BPMN implementados na automatização. Não possui nenhuma funcionalidade específica para fazer a validação do modelo no editor, mas algumas das regras básicas são verificadas automaticamente.

- Bonita BPM[11] - O modelador do Bonita BPM é uma ferramenta integrada do BPMS da Bonitasoft, ou seja, é necessário ser instalada a suite completa da Bonita BPM *Community* para que se possa testar o modelador. A ferramenta possui os itens padrão, uma paleta de elementos BPMN, um conjunto de separadores para documentação e configuração dos elementos dos modelos de processos. A paleta de elementos não contém todos os elementos da notação. Não contém, por exemplo, as tarefas manuais, as tarefas de regra de negócios, os subprocessos embutidos/incorporados. Para hierarquização de processos terão de ser utilizados subprocessos identificados na ferramenta como atividades de chamada. Não contém também nenhuma opção de fluxo, ou seja, limita-se ao fluxo de sequência. Esta falta de elementos para o desenvolvimento de modelos de BPMN não possibilita usufruir de todo o poder de expressão da notação. É uma ferramenta de modelação eficaz, pois apesar da limitação dos elementos indisponíveis da notação, eficiente, com fluidez na tarefa de modelação dos processos. Permite também construir e executar o processo, após ser desenvolvido o formulário do utilizador.
- Adonis[12] - O Adonis é uma ferramenta fabricada e comercializada pelo BOC Group utilizada para documentação, análise e otimização de processos de negócio. É uma aplicação baseada na *web*, compatível com a notação BPMN 2.0, embora só permita a modelação de dois tipos de *gateways*, o *gateway* exclusivo e o *gateway* paralelo. O Adonis permite adicionar objetos rapidamente ao diagrama e ligar todos os objetos automaticamente. Possui uma funcionalidade de arrastar e soltar que reduz consideravelmente o tempo e esforço de modelação. Permite a validação do diagrama de acordo com os critérios da notação e permite a exportação do modelo nos formatos de imagem e PDF. O ADONIS fornece uma variedade de funcionalidades diferentes, não se limitando à modelação de processos de negócio. Essas funcionalidades são

[11] https://www.bonitasoft.com/

[12] https://www.adonis-community.com/en/

nomeadamente: recursos de análise, relatórios gráficos, simulação de processos, insights orientados por dados, publicação e trabalho colaborativo e também automatização de processos de processos BPMN 2.0.

Através da análise das ferramentas de modelação BPMN foi possível observar que a ferramenta Bizagi é uma ferramenta que se destaca das restantes ferramentas que foram avaliadas, pois permite modelar, validar e executar os processos, permite a exportação dos modelos no formato padrão BPMN e suporta os elementos fundamentais do BPMN, como os 8 tipos de tarefas especificas, os diversos tipos de *gateway* e os vários tipos de fluxos existentes.

Através desta análise foi possível ainda observar que a principal diferença entre ferramentas académicas e industriais é que as ferramentas industriais normalmente fazem parte de uma maior suite BPM que fornece suporte para simulação de processos, automatização e gestão dos processos, enquanto as ferramentas académicas se concentram apenas na modelação dos processos.

A crescente popularidade do padrão BPMN dá origem a um aumento do número de ferramentas que suportam a modelação de processos nesta notação. A existência de padrões como o BPMN não garante que todas as ferramentas possuam suporte para a implementação desse padrão. Além desta questão existem diferenças no que diz respeito ao design, aos recursos de armazenamento, às políticas de licenciamento entre outros aspetos.

Capítulo 3 - O sistema de ETL

Para suportar a tomada a decisão é necessário extrair conhecimento através da informação existente. Para isso, várias etapas têm de ser abordadas e envolvem (entre outros procedimentos) a recolha dos dados utilizados para suportar os processos de negócio existentes numa organização.

Diariamente são geradas grandes quantidades de dados nas empresas resultantes das mais diversas atividades. Os dados gerados podem ser provenientes de atividades comerciais associadas ao domínio de negócio da organização (por exemplo, vendas, encomendas, etc.), mas também de atividades que de alguma forma se relacionam com o domínio de negócio (por exemplo, dados comportamentais relacionados com as interações dos utilizadores com um website). Para além de a estrutura e representação dos dados variar (podemos estar perante dados estruturados, semiestruturados ou mesmo não estruturados), os dados encontram-se tipicamente fragmentados pelos diversos departamentos da organização, o que envolverá processos de extração de dados complexos, condicionados pela disponibilidade dos sistemas fonte e das tecnologias que os suportam.

Os dados gerados por processos de negócio representam as transações diárias da empresa, onde são realizadas operações de inserção, atualização e eliminação de dados sobre uma ou mais bases de dados tipicamente relacional. Este tipo de sistema é conhecido por *Online Transactional Processing* (OLTP) e é aquele que armazena a informação mais valiosa para a empresa. Os sistemas operacionais registam os dados do processo de negócio da organização tendo em consideração os requisitos e decisões operacionais.

Além dos sistemas operacionais, existem os sistemas analíticos, que são construídos tendo em vista suportar as decisões de negócio, ou seja, as decisões táticas e estratégicas. Estes sistemas utilizam os dados provenientes dos sistemas operacionais para atender aos requisitos de análise específicos de cada organização. Geralmente são sistemas *Online Analytical Processing* (OLAP) que são estruturas multidimensionais de dados otimizadas para obter e manipular grandes quantidades de dados sob múltiplas perspetivas de uma forma fácil e seletiva dando suporte à tomada de decisão.

O *Business Intelligence* (BI) oferece suporte à tomada de decisão pois consiste na recolha, organização, análise e gestão de informações. Segundo a empresa Gartner[13], líder em consultoria e pesquisa no mercado das novas tecnologias, BI é um termo que se refere a aplicações, infraestruturas, ferramentas e práticas que permitem o acesso e análise de dados e assim otimizar decisões e desempenho. Inicia-se pela integração de dados de uma ou mais fontes utilizando um repositório de armazenamento único, específico e consistente dos dados, por exemplo, um *Data Warehouse* [25] . Os dados armazenados neste repositório são adicionados à medida que novos dados ou alterações são introduzidas nos sistemas fonte e tendo em vista uma representação da informação ao nível histórico. Posteriormente estes dados são analisados através de ferramentas específicas, que permitem desenvolver *dashboards* e relatórios utilizados no suporte à tomada de decisão.

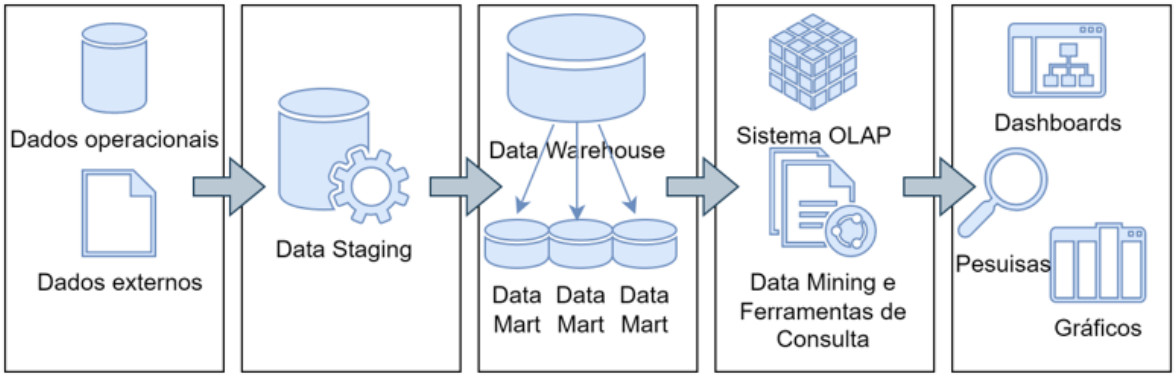

Figura 20 - Arquitetura de um Sistema de BI

Um sistema típico de BI tem como premissa fornecer uma visão geral do processo de negócio de forma a suportar a tomada de decisão. A figura acima ilustra uma possivel arquitetura geral de um sistema de BI [25] cuja estrutura é constituída por cinco camadas ou cinco componentes, onde cada camada corresponde a um ambiente completo de BI (ver figura 20):

[13] https://www.gartner.com/en

- Fontes de dados: o ambiente de fontes de dados corresponde aos locais onde se encontram as bases de dados, seja uma ou mais fontes internas ou externas à organização, que darão suporte ao sistema;
- Sistema de integração de dados: o ambiente de integração de dados é onde se realiza o processo *Extract-Transform-Load*[14] (ETL). Os dados estão extraídos das diversas fontes, são transformados e carregados para os repositórios de dados.
- Ambiente de *Data Warehouse*: é o repositório central de dados, onde os dados são carregados após passar pelo processo de ETL.
- Análise de dados: os mecanismos de análise de dados fornecem como recurso várias técnicas, como OLAP, *Data Mining* e ferramentas de consulta, que permitem que os dados sejam trabalhados para que as informações relevantes possam ser e disponibilizadas aos gestores para auxiliar na tomada de decisão;
- Visualização de dados: o componente de visualização de dados disponibiliza diversas aplicações que permitem aceder à informação atendendo os requisitos específicos da organização para que os gestores possam extrair valor, acompanhar o desempenho do negócio, através de relatórios e *dashboards*.

3.1 O Sistema de Data Warehouse

Os sistemas de *Data Warehouse* (DW) [26] assumem um papel vital no planeamento das atividades empresariais, permitindo o desenvolvimento de aplicações otimizadas para suporte à decisão servindo como ferramenta de análise de dados permitindo que os utilizadores construam os seus próprios caminhos de exploração dos dados de forma intuitiva e autónoma. A elaboração de um DW consiste em agregar informação proveniente de uma ou mais Bases de Dados (BD), ou de outras fontes como documentos ou folhas de cálculo, para posteriormente ser tratada, formatada e consolidada numa única estrutura de dados. Um sistema DW (SDW) consiste numa BD com um grande volume de dados que são extraídos das fontes heterogéneas mencionadas. A informação, após ser armazenada, fica disponível no DW/*Data Marts* (DM) para consulta que visa ajudar nas tomadas de decisão. A estrutura do um DW e dos DM é desenvolvida de forma a facilitar a análise e consulta desses

[14] Extração, Transformação e Carregamento, em português.

dados. Devido ao custo elevado, o DW muitas vezes é dividido em partes menores, nomeadamente os DM que consolidam apenas as informações de uma determinada área do negócio. Após a sua criação, os vários DM podem ser unidos para formarem um único DW [27].

Para a construção de um DW são necessários vários passos quer ao nível da extração quer ao nível do processamento de dados. O processo ETL destina-se à extração transformação dos dados e termina com a inclusão destes no DW. Nesta fase realizam-se os procedimentos de limpeza, integração e transformação de dados e segundo a literatura este é o processo mais crítico e demorado na construção de um DW.

Uma das ferramentas mais utilizadas para o acesso e a análise dos dados de um DW é o OLAP. Através desta ferramenta é possível realizar o tratamento dos dados provenientes das diferentes fontes em tempo real e permite também utilizar uma grande variedade de ferramentas de visualização e organizar os dados segundo os critérios de seleção pretendidos. A maior vantagem do OLAP é a capacidade de realizar análises multidimensionais dos dados, associadas a cálculos complexos.

3.2 O processo de ETL

O *Extract-Transform-Load* (ETL) é um sistema de povoamento de um DW, que é responsável pela extração de dados de várias fontes, a sua limpeza, otimização e inserção desses dados num DW (ver figura 21).

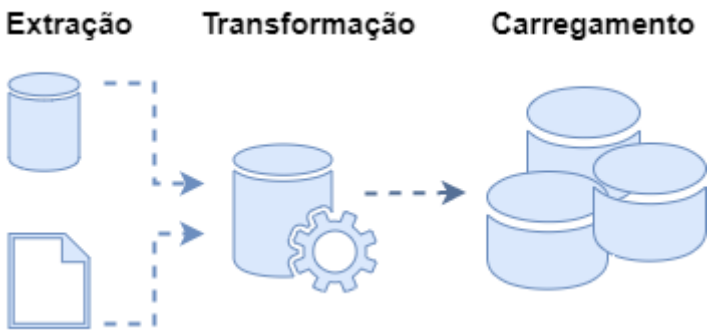

Figura 21 - Processo ETL

O processo de ETL inicia-se pela **extração** dos dados a partir das fontes de dados operacionais, que na maioria dos casos são bases de dados relacionais, embora também

possam ser de outro formato como bases de dados NoSQL, documentos de texto ou folhas de cálculo. Para o carregamento inicial dos dados no DW, todos os dados das fontes de dados são extraídos e encaminhados para as restantes atividades de ETL. Esses dados são obtidos por rotinas de extração que fornecem informação igual ou modificada, relativamente à fonte original. Quanto maior for o volume de dados, mais tempo será consumido na execução da atividade de extração. Após o carregamento inicial, a extração costuma ser feita de forma incremental, com o objetivo de manter os dados do DW atualizados. Desta forma, a atividade de extração tem como permissão de identificar, nas fontes de dados, as alterações que ocorreram desde a extração anterior, ou seja, os novos dados inseridos ou dados que foram alterados e de seguida essas diferenças são extraídas e encaminhadas para uma área de retenção para a atividade de transformação dos dados [28]. Esta a área de retenção, chamada de *Data Staging Area* (DSA), é tipicamente utilizada uma área especial para o tratamento de dados. O processo de ETL inicia-se pela extração dos dados a partir das fontes de dados operacionais, que na maioria dos casos são bases de dados relacionais, embora também possam ser de outro formato como bases de dados NoSQL, documentos de texto ou folhas de cálculo. Para o carregamento inicial dos dados no DW, todos os dados das fontes de dados são extraídos e encaminhados para as restantes atividades de ETL. Esses dados são obtidos por rotinas de extração que fornecem informação igual ou modificada, relativamente à fonte original. Quanto maior for o volume de dados, mais tempo será consumido na execução da atividade de extração. Após o carregamento inicial, a extração costuma ser feita de forma incremental, com o objetivo de manter os dados do DW atualizados. Desta forma, a atividade de extração tem como permissão de identificar, nas fontes de dados, as alterações que ocorreram desde a extração anterior, ou seja, os novos dados inseridos ou dados que foram alterados e de seguida essas diferenças são extraídas e encaminhadas para o DSA para a atividade de transformação dos dados [28].

A segunda atividade diz respeito à **transformação** dos dados. Os dados brutos, que foram extraídos das diversas fontes de informação, passam pelo processo de transformação para que posteriormente, sejam armazenados no DW. A transformação envolve as tarefas de limpeza, conversão e integração dos dados. A transformação visa compatibilizar os dados brutos extraídos das diversas fontes de dados com o esquema proposto para o DW. Para não interferir com as fontes de dados e com os sistemas que as alimentam, geralmente a tarefa de transformação é executada na DSA do ambiente do DW. A transformação dos dados visa resolver conflitos quer a nível de esquema, quer a nível de conformidade dos dados. A

limpeza dos dados corresponde a uma série de tarefas para a modificação ou a eliminação de dados indesejáveis, redundantes, duplicados, inconsistentes ou inválidos, de forma a melhorar a qualidade dos dados [29].

A última etapa do processo ETL consiste no **carregamento** dos dados no DW. Os dados de entrada da atividade de carregamento foram previamente extraídos das diversas fontes de dados e passaram por tarefas de transformação e limpeza. O carregamento de dados envolve desafios técnicos que devem ser resolvidos pelos técnicos envolvidos no processo, por exemplo a utilização de processos otimizados, para o carregamento de dados, de forma a estruturar o processo de povoamento. Esta etapa inclui primeiramente o carregamento das tabelas de dimensões seguido do carregamento das tabelas de factos.

Segundo alguns autores [4], [30], a elaboração de um processo ETL incide maioritariamente sobre o mapeamento dos atributos dos dados de uma ou várias fontes para os atributos das tabelas do DW. Num DW, os dados normalmente utilizados estão localizados em BD multidimensionais. É importante referir que as alterações nos dados não afetam as fontes originais, mas sim, os dados no momento de extração para o repositório da DW. Assim, depois do processo de transformação ocorre o processo de carregamento. Neste processam-se os mapeamentos sintáticos e semânticos entre os esquemas, respeitando as restrições de integridade e criando assim uma visão concretizada e unificada das fontes. Este processo é dos mais árduos e complexos de executar devido a sua complexidade que dependerá da heterogeneidade das fontes.

Existe um conjunto de diretrizes para a realização do planeamento do processo ETL, que podemos agrupar nas seguintes quatro fases, nomeadamente, 1) Definição de fontes de dados e repositório de destino, 2) *Source-to-target-mapping*, 3) Descrição do sistema de povoamento, e 4) Modelação conceptual [15], [10], [31], [32].

A primeira etapa do planeamento do processo ETL (**Definição de fontes de dados e repositório de destino**) consiste em caracterizar as fontes de dados de origem, assim definir o repositório de destino. Este processo consiste em analisar as fontes de informação, através da observação das suas características e quais as relações de interesse com o sistema de DW.

A segunda fase do planeamento do processo ETL [6] consiste em efetuar o mapeamento entre as fontes e o repositório (***Source-to-target-mapping*** *ou* ***Logical Map***[15]). Ao mover os dados de um sistema para outro, é difícil existir uma situação em que a origem e o sistema de destino possuam a mesma estrutura. Portanto, existe a necessidade de um mecanismo que permita aos utilizadores mapear os seus atributos no sistema de origem para os atributos no sistema de destino. Este processo pode ser definido como o conjunto de descrições sobre as atividades de transformação de dados necessárias entre o(s) sistema(s) de origem na estrutura e no conteúdo necessários para o sistema de destino.

A terceira fase (**Descrição do sistema de povoamento**) consiste em definir a arquitetura e principais estratégias do sistema de povoamento. Normalmente o processo de ETL extrai os dados oriundos dos sistemas operacionais e normaliza a estrutura dos dados de acordo com os requisitos e estrutura de armazenamento de destino. Existem três tipos de extração: total, diferencial e selecionada. Na extração total, todos os dados são extraídos das fontes e são utilizados para o povoamento inicial do DW, ou seja, consiste numa extração em lote. A extração diferencial consiste num povoamento incremental, onde primeiramente são extraídos todos os dados da fonte e de seguida é realizada uma comparação dos dados extraídos com os dados do último povoamento, sendo dessa forma acrescentados ao DW os dados novos. A extração selecionada refere-se ao processo de *Change Data Capture* (CDC) que consiste num conjunto de procedimentos utilizados para identificar e capturar mudanças nos dados da base de dados. Relativamente à transformação e limpeza dos dados, existem alguns processos comumente utilizados para assegurar a qualidade e a integridade dos mesmos, como por exemplo, a normalização e transformação de valores, eliminação de duplicados, conciliação dados que existam em múltiplas fontes, geração de chaves de substituição entre outros.

Finalmente, a última etapa (**Modelação conceptual**) antes da implementação física do projeto consiste na modelação conceptual do processo ETL. De uma forma geral, a modelação conceptual do ETL tem como objetivo propor e organizar as várias tarefas, funções e mapeamentos necessários para conciliar os dados das fontes de dados com a

[15] Em português - Mapeamento de origem para destino

estrutura proposta para o DW, assim como documentar todas as tarefas envolvidas na execução do processo.

3.3 Ferramentas de ETL

As ferramentas de ETL são *softwares* responsáveis pela extração de dados de diversas fontes, a limpeza e customização (transformação) e inserção (carregamento) num DW. Existem no mercado diversas ferramentas capazes de executar processos de ETL. As ferramentas de ETL disponíveis atualmente encontram-se bem preparadas para o processo de extração, transformação e carregamento. Uma boa ferramenta de ETL deve ser capaz de comunicar com as diversas BD e ler diferentes formatos de dados. Atualmente a oferta de ferramentas ETL é elevada. As ferramentas ETL de software empresarial mais utilizadas, ou seja, ferramentas proprietárias, são: IBM Data Stage[16], Oracle Data Integrator[17], SAS Data Management[18], Talend[19], e Informatica Power Center[20]. O SQL Server Integration Services[21] (SSIS) - possui uma edição developer gratuita, sendo esta utilizada a nível académico. E é possível referir ainda a ferramenta gratuita Pentaho Data Integrator[22], também conhecida por Kettle.

A seleção de uma ferramenta de ETL adequada é uma decisão muito importante a ser tomada, uma vez que é necessário avaliar para o projeto em questão quais são os principais requisitos de extração de dados de múltiplas fontes e a sua transformação.

---

[16] https://www.ibm.com/products/datastage

[17] https://www.oracle.com/pt/middleware/technologies/data-integrator.html

[18] https://www.sas.com/en_us/software/data-management.html

[19] https://www.talend.com/

[20] https://docs.informatica.com/pt_pt/data-integration/powercenter/10-2.html

[21] https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver16

[22] https://www.hitachivantara.com/en-us/pdf/legal/pentaho-data-integration-open-source-software-packages.pdf

Ao contrário de outros componentes de uma arquitetura de *Data Warehousing,* é muito difícil mudar a ferramenta ETL, uma vez que cada ferramenta utiliza o seu próprio modelo de representação e de estruturação do processo, o que está naturalmente associado às decisões de arquitetura consideradas no desenvolvimento de cada ferramenta.

É importante referir ainda, que existem abordagens *code-first* para ETL, que por serem ao nível do código saem do escopo deste trabalho.

3.4 Abordagens de Modelação

Os DWs integram diferentes fontes de dados para dar uma visão multidimensional dos mesmos ao tomador de decisão. Para tal, os processos de ETL (Extração, Transformação e Carregamento) são responsáveis por extrair dados das fontes de dados operacionais heterogéneas, a sua transformação (conversão, limpeza, padronização, etc.) e a sua carga no DW. O ETL representa um componente chave do DW. Não só devido à enorme quantidade de recursos necessários à sua implementação, mas também porque os processos se relevarem inadequados, todo o sistema de DW fica comprometido.

Nas últimas décadas existiram várias abordagens propostas para a modelação conceptual do processo ETL. Esses trabalhos de pesquisa podem ser classificados considerando as linguagens de modelação que os autores utilizaram como o UML e o BPMN, que foram estudadas e/ou estendidas para incorporar os conceitos específicos no domínio do ETL, ou então *frameworks* especificas desenvolvidas considerando as características de um sistema de ETL. Este subcapítulo contém uma breve discussão sobre essas técnicas.

Segundo a revisão efetuada pelos autores [33] é amplamente reconhecido que o projeto e a manutenção adequados dos processos de ETL são fatores-chave para o sucesso dos projetos de DW e que, se os processos de ETL não forem bem projetados, dados errados levarão a decisões incorretas e todo o projeto de DW provavelmente falhará. Nos últimos anos, várias abordagens de modelação conceptual foram propostas para projetar processos de ETL.

Há muitas formas diferentes de especificar processos ETL. Por exemplo, a utilização de ferramentas especializadas relacionadas a Sistemas de Gestão de Bases de Dados (SGBDs) tradicionais como o IBM WebSphere Application Server, o SQL Server Integration Services e

o Oracle Warehouse Builder. No entanto, a nível conceptual essas soluções têm algumas desvantagens importantes: fornecem notação proprietária diretamente associada às características de arquitetura de cada ferramenta; é necessário conhecer pormenores técnicos associados à ferramenta para desenhar ETL (não aplicável para modelos conceptuais); e toda a implementação fica ligada à ferramenta, o que torna a mudança para uma nova ferramenta muito difícil, senão mesmo impossível. Essas características dificultam a integração em ambientes heterogêneos de DWs e a curva de aprendizagem aumenta. Portanto, muitas organizações preferem desenvolver os seus próprios processos de ETL através de um desenvolvimento manual de código ad-hoc específico (por exemplo, Structured Query Language (SQL)). No entanto, esta forma de proceder implica alguns problemas, devido ao seu alto custo de manutenção.

A literatura mais relevante sobre o tema defende a modelação conceptual de processos ETL, pois essas propostas de modelação conceptual melhoram o desenvolvimento dos processos ETL, ao serem muito úteis para documentar os processos ETL e apoiar as tarefas do *designer*. No entanto, a maior parte das abordagens carecem de mecanismos de geração automática de código para executar o processo de ETL em plataformas específicas, de modo que seja possível tirar partido das especificações conceptuais. Essa etapa é necessária para evitar falhas e ganhar tempo no desenvolvimento na implementação de processos ETL complexos.

Nas subseções seguintes são apresentadas as abordagens de modelação de processos ETL.

3.4.1 Modelação UML e SysML

A primeira tentativa de modelação conceptual de ETL foi estabelecida em [34]. Os autores focaram no problema da definição de atividades de ETL e forneceram fundamentos formais para a sua representação conceptual. O modelo conceptual proposto pelos autores consiste na personalização do mapeamento de relacionamentos entre atributos e as respetivas atividades de ETL nas etapas iniciais de um projeto de DW. É enriquecido com uma 'palete' de um conjunto de atividades ETL frequentemente utilizadas, como a atribuição de chaves substitutas, a verificação de valores nulos, entre outros. E foi construído de uma forma personalizável e extensível, para que o *designer* possa enriquecê-lo com os seus próprios padrões recorrentes para atividades de ETL. Os autores propuseram um processo genérico personalizável e forneceram um conjunto de notações para representar as atividades de ETL.

Através deste modelo estabeleceram o relacionamento entre os atributos de origem e os atributos do DW.

Os autores enriqueceram ainda mais seu trabalho em [35] propondo uma metodologia que mostra o procedimento passo a passo desde a seleção da fonte até ao povoamento do DW, juntamente com o mapeamento de relacionamento de atributos e anotação do diagrama com restrições de tempo de execução.

Os autores em [36] projetaram o fluxo de trabalho de ETL com base na abordagem de modelação UML. Esta foi a primeira abordagem de projeto de modelo conceptual a utilizar a notação UML padrão. Os autores utilizam o diagrama de classes UML para estabelecer o relacionamento da base de dados e dos seus atributos. Através da utilização da UML, é possível modelar diferentes aspetos de uma arquitetura DW, como fontes de dados operacionais, o esquema conceptual do DW de destino e processos ETL de maneira integrada, utilizando sempre a mesma notação. Neste trabalho, os autores fornecem os mecanismos necessários para uma especificação fácil e rápida das operações comuns definidas nos processos de ETL, como a integração de diferentes fontes de dados, a transformação entre os atributos de origem e destino, a geração de chaves substitutas entre outros. Outra das vantagens da proposta é a utilização da UML (padronização, facilidade de uso e funcionalidade) e a integração perfeita do design dos processos ETL com o esquema conceptual do DW. Nesta abordagem de modelação de processos ETL, os autores definiram um conjunto de estereótipos UML que representam as tarefas ETL mais comuns, por exemplo: a integração de diferentes fontes de dados, a transformação entre os atributos de origem e destino, a geração de chaves substitutas. Os vários processos de transformação, como agregação, conversão, filtro, junção, etc., são suportados por sua modelação com facilidade de *zoom in* e *zoom out* para diferentes níveis de design. Os autores propõem-se em trabalhos futuros, a desenvolver uma metodologia que permita integrar todos os modelos e esquemas que forem utilizados num projeto de DW (o esquema DW de destino, esquemas de fonte de dados, processos ETL, etc.) numa abordagem formal.

Outro esforço de pesquisa que utilizou o UML foi proposto em [37] onde os autores destacaram apenas a fase de extração. Eles identificaram seis classes e mostraram o diagrama de classes, o diagrama de caso de uso e o diagrama de sequência para a fase de extração através da utilização da notação UML padrão. Os autores não incluem a fase de transformação e nem a fase de carregamento no referido trabalho.

Em [38] os autores modelaram um processo ETL completo através da utilização de diagramas de atividades UML. A atividade envolvida no processo de ETL é expressa através de um diagrama com sequência de fluxo de controlo que suporta várias atividades de transformação. Após isso os autores enriqueceram o seu trabalho em [33] propondo a geração automática de código a partir de modelos conceptuais, apoiando a arquitetura orientada a modelos - *Model Driven Architecture* (MDA) para projetar processos ETL. Projetaram um modelo conceptual baseado no seu trabalho anterior [38] através da utilização de *Platform Independent Model* (PIM) com suporte aos recursos UML. O PIM pode oferecer uma visão funcional do sistema sem se preocupar com a plataforma. Diferentes *Platform Specific Model* (PSM) que mostram a visualização do modelo lógico podem ser produzidos a partir do PIM. O código de criação automática da estrutura de dados é gerado a partir do PSM individual. A transformação do modelo PIM para o modelo PSM é feita pela linguagem *Query View Transformation* (QVT).

Em [39] os autores propuseram uma abordagem de modelação estendida da UML, a *Systems Modeling Language* (SysML), que oferece mais facilidades em relação à UML. Os autores consideraram que implementar qualquer modelo SysML é mais adequado para os programadores e outros técnicos, pois é desenvolvido do ponto de vista da engenharia de sistemas. Traz outra vantagem em relação à UML, que consiste no facto de ser mais flexível e expressiva, ser focada na análise de requisitos e não em visões centrados no software. Em [40] os autores ampliaram seu trabalho, ao apresentar uma automatização para a validação do modelo SysML.

3.4.2 Modelação notações próprias

No artigo [41], os autores aprofundam o projeto lógico de cenários ETL e fornecem uma estrutura genérica e personalizável para apoiar o *designer* do DW na sua tarefa. Primeiramente, apresentam um metamodelo especialmente padronizado para a definição de atividades de ETL. Seguem uma abordagem do fluxo de trabalho, onde a saída de uma determinada atividade pode ser armazenada de forma persistente ou passada para uma atividade subsequente. No referido trabalho é ainda utilizada uma linguagem de programação de base de dados declarativa, LDL, para definir a semântica de cada atividade. Os autores referem que o metamodelo é genérico o suficiente para capturar qualquer atividade ETL. Na procura de maior reusabilidade e flexibilidade, especializaram o conjunto de metamodelos genéricos com uma paleta de atividades de ETL frequentemente utilizada, à qual

denominaram *templates*. Os conceitos de design que os autores apresentaram foram implementados numa ferramenta especifica, o ARKTOS II. Os autores começaram por definir um metamodelo formal com uma abstração dos processos ETL ao nível lógico. São definidos os repositórios de dados, as atividades e suas partes constituintes. Uma atividade é definida como uma entidade com possivelmente mais de um esquema de entrada, um esquema de saída e um esquema de parâmetro, de modo que a atividade seja preenchida a cada vez com seus valores de parâmetro apropriados. O fluxo de dados dos produtores para seus consumidores é feito através da utilização de relacionamentos que mapeiam os atributos dos primeiros para os respetivos atributos dos segundos. Em segundo lugar, fornecem uma estrutura de reutilização que complementa a generalidade do metamodelo. Isso é obtido a partir de um conjunto de especializações "embutidas" das entidades da camada de metamodelo, especificamente adaptadas para os elementos mais frequentes de cenários ETL. Essa paleta de atividades de modelo será chamada de camada de modelo e é caracterizada pela sua extensibilidade. Por último, discutem questões de implementação e apresentam uma ferramenta gráfica, ARKTOS II, que facilita o desenho de cenários ETL, com base no modelo apresentado.

Em [42] os autores aprofundam a otimização lógica dos processos ETL, modelando-a como um problema de pesquisa no espaço de estados. Consideraram cada fluxo de trabalho ETL como um estado e produziram o espaço de estados através de um conjunto de transições de estado corretas. Forneceram também algoritmos para a minimização do custo de execução de um fluxo de trabalho ETL. Primeiramente os autores apresentam uma declaração formal teórica para o problema, modelando-o assim como um problema de busca no espaço de estados, com cada estado representando um projeto particular do fluxo de trabalho como um grafo. Os vértices do grafo representam atividades e dados armazenados, e as arestas capturam o fluxo de dados entre os vértices. Através da modelação do problema de procura no espaço de estados, definiram transições de um estado para outro que estendem as técnicas tradicionais de otimização de consultas. Provaram a correção das transições introduzidas e também forneceram detalhes sobre como os estados são gerados e as condições sob as quais as transições são permitidas.

Por fim, apresentaram algoritmos para a otimização de processos de ETL. Primeiro, utilizaram um algoritmo exaustivo para explorar o espaço de procura na sua totalidade e encontrar o fluxo de trabalho ETL ideal. Um fluxo de trabalho ETL é modelado como um grafo acíclico direcionado. Os vértices do grafo compreendem atividades e conjuntos de registos. Um

conjunto de registos é qualquer armazenamento de dados que pode fornecer um esquema de registo simples possivelmente através de uma interface de *gateway*; no restante do artigo, os autores trataram principalmente dos dois tipos mais populares de conjuntos de registos, ou seja, tabelas relacionais e ficheiros. Através da utilização de algoritmos de procura intensa e heurística para reduzir o espaço de pesquisa que exploraram e demonstraram a eficiência da abordagem através de um conjunto de resultados experimentais.

As abordagens apresentadas que focam em notações específicas podem causar um esforço extra para a equipa de desenvolvimento de ETL aprender as especificidades da notação, assim como dificulta a compreensão pelos utilizadores não técnicos.

3.4.3 Modelação BPMN

O BPMN (ver Capítulo 2 – A notação BPMN) consiste num conjunto de elementos gráficos padrão que ajudam a entender os processos de negócios dentro de uma organização como abordado e demonstrado no capítulo 2. A primeira tentativa de utilizar a notação BPMN na modelação conceptual ETL foi proposta em [15]. Os autores afirmam que o processo ETL pode ser considerado um tipo particular de processo de negócio e que pode facilitar a comunicação tanto com o pessoal técnico como não técnico. Como o BPMN é uma notação amplamente utilizada para modelação e execução de processos de negócios, não é raro vê-la usada para auxiliar em outros cenários. Os autores descreveram o processo de formação do modelo conceptual e a conversão de BPMN para *Business Process Execution Language* (BPEL). Esta conversão foi feita para executar o modelo projetado, bem como implementar as relações com *web services.* Seguindo o mesmo propósito, os autores apresentaram em[10] uma atualização ao modelo juntamente com os fatores de manutenção necessários e o projeto de um metamodelo BPMN para visão conceptual de ETL. Numa continuação do seu trabalho, os autores, apresentaram em [43] um metamodelo BPMN independente de fornecedor baseado em *Model-Driven Development* (MDD) onde o metamodelo é criado e a geração automática de código para qualquer plataforma específica de fornecedor é proposta. Após esse trabalho, os autores prosseguiram com o tema, em [44] propondo a transformação de modelo para texto e de modelo para modelo para geração de código.

Ainda seguindo a proposta de modelação através de BPMN em [32], os autores projetaram um metamodelo ETL generalizado para algumas tarefas específicas através da notação BPMN o que possibilita a modelação e validação dos processos, antes de se proceder à sua

implementação. Este trabalho complementa e estende o trabalho de. incorporando construções de modelos conceptuais específicos como padrões BPMN para atividades de ETL como 'captura de dados alterados, 'dimensões de mudança lenta', *'pipelining* de chave substituta' ou melhoria de qualidade de dados., entre outros. Os autores utilizaram um Diagrama de Colaboração para representar a interação de componentes independentes (padrões), fornecendo uma primeira abordagem para um sistema ETL multicamadas usando BPMN, mostrando de forma mais simples como os padrões ETL podem ser utilizados para suportar a modelação conceptual ETL com BPMN. Essa base pode ser estendida para criar mais padrões BPMN cobrindo todas as atividades de um fluxo de trabalho ETL, o que resulta em ajudar a desenvolver um fluxo de trabalho ETL de alta qualidade, diminuindo o número de erros e tornando o processo mais eficiente. Por fim, os autores validaram o seu modelo utilizando um estudo de caso através da criação de esqueletos ETL que consistem em sistemas ETL que geraram a partir de uma especificação de modelo conceptual em BPMN. Além disso, os autores avançaram no seu trabalho para o processo de geração automática de modelos conceptuais para físicos em [45] e [46].

Mais recentemente em [47] os autores deram continuidade a esta abordagem através da representação da modelação conceitual ETL em diferentes camadas, cada uma representando um detalhe de processo diferente. Esta abordagem tem em vista fornecer à equipa de desenvolvimento de ETL ferramentas específicas para a comunicação em diferentes fases de desenvolvimento do ETL. Cada camada representa um novo nível de detalhe aplicado a uma construção específica descrita na camada anterior. Isso contribui para uma abordagem de desenvolvimento mais ágil, pois os modelos podem enriquecer os requisitos do sistema de forma incremental. De forma a demonstrar a utilização da técnica os autores exploraram um subprocesso específico de um cenário ETL e demonstraram como as especificidades de ETL podem ser representadas num nível conceptual de uma maneira eficaz.

3.5 Exemplo

De forma a ilustrar a problemática inerente a este trabalho é apresentado um exemplo comum do mundo real da área de vendas que aborda todas as etapas ETL num esquema de DW. O modelo do DW apresentado na figura 22 integra quatro tabelas de dimensão Produto, Cliente, Funcionário, e Calendário e uma tabela de factos Vendas que permitem analisar os dados

das vendas do conjunto de lojas ao longo do tempo. O modelo conceptual apresentado na figura 22 consiste num modelo conceptual chamado *Dimensional Fact Model* (DFM) proposto por Golfareli [26]. O esquema é composto por medidas: a quantidade vendida, a margem de lucro e o valor dos impostos. O grão consiste na venda de um produto de uma determinada marca, modelo e cor, a um cliente, por um determinado funcionário, numa determinada loja, numa determinada freguesia, numa certa cidade, num determinado país, numa determinada data.

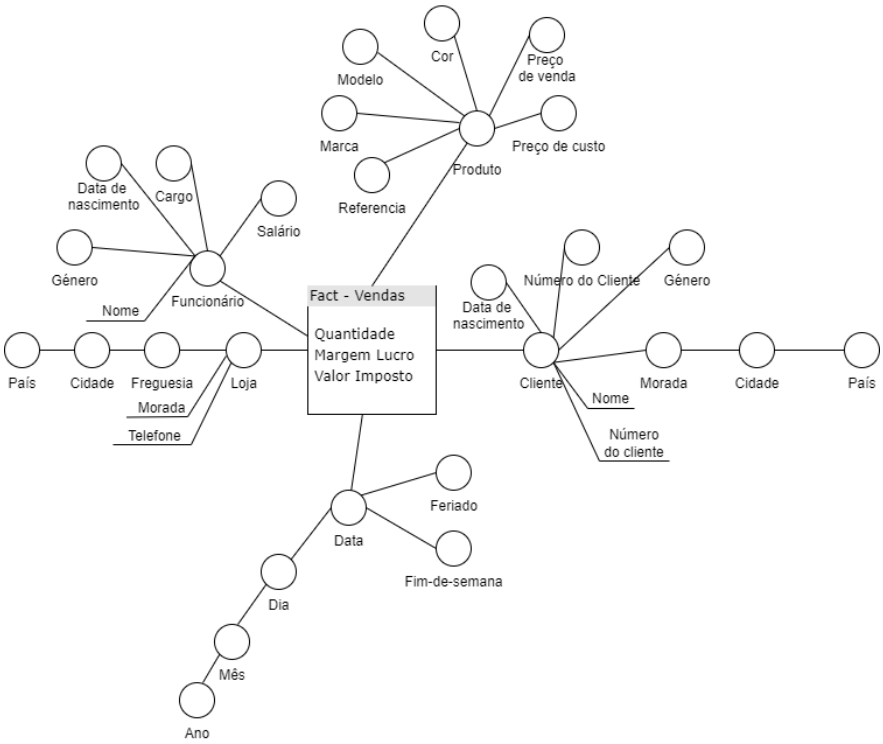

Figura 22 - Esquema em estrela – Vendas

Depois de elaborado o esquema dimensional, é necessário definir o processo ETL adequando-o à estrutura do DW e às fontes de origem, pois uma especificação incorreta ou incompleta das atividades de ETL afetará a qualidade do DW. O exemplo apresentado, embora sendo um exemplo simples acarreta a realização de várias tarefas. Ao analisar cada tarefa individualmente é possível identificar várias subtarefas. Na figura 23 está apresentado o *workflow* ETL necessário ao preenchimento da dimensão Clientes, implementado no SSIS. Este processo inicia-se pela extração dos dados dos clientes da fonte que consiste num ficheiro *clientes.csv*. Posteriormente os dados são transformados e integrados, através de uma transformação dos valores da coluna género que não se encontra uniformizada na fonte,

ou seja, existem valores irregulares como “f”, “M”, “Feminino”. Essa transformação é realizada através da ferramenta “*Derived Column”* e representa uma operação simples que pode ser reutilizada noutras situações semelhantes. Por último os dados são carregados do DW de destino.

Ao efetuar uma análise ao processo de ETL representado na figura 23 que ilustra o processo implementado através da notação gráfica da ferramenta SSIS é possível verificar que existe a necessidade de estabelecer uma ponte entre a definição do processo ETL e a sua implementação física, ou seja, a modelação conceptual. A utilização de um modelo abstrato torna-se vantajosa tanto para os utilizadores técnicos que podem focar todo o esforço na implementação física tanto para os utilizadores não técnicos pois podem concentrar-se nos detalhes operacionais e nos requisitos do negócio. Essa abordagem facilita também a comunicação entre os utilizadores técnicos e não técnicos e ainda possibilita a otimização de todo o processo caso seja alterada a ferramenta de implementação de ETL.

Figura 23 - Implementação da dimensão Clientes SSIS

Capítulo 4 – BPMN para ETL

A elaboração de um DW surge da necessidade das empresas obterem respostas rápidas perante o grande volume de dados existente. O DW torna-se um repositório consolidado e centralizado que permite o acesso fácil às informações armazenadas. Os DW são elaborados com base nas atividades operacionais da organização e a partir das quais se definem as métricas a analisar. Estes visam responder às questões *como* e *o quê,* i.e., os requisitos a partir dos quais são construídos as dimensões e os factos. A integração dos dados no DW efetua-se através do processo ETL que consiste numa tarefa de grande importância na entrega de dados conformes e de qualidade.

Um processo ETL pode ser considerado um tipo específico de processo de negócio. Tal como nos processos de negócios tradicionais, não existe um modelo padrão para definir os processos de ETL, cada uma das ferramentas existentes fornece seu próprio modelo. Geralmente são modelos detalhados pois consideram os vários problemas de implementação e as especificidades das ferramentas. Neste capítulo é abordada a representação conceptual do desenvolvimento do processo ETL através da utilização da notação BPMN.

A notação BPMN tornou-se um padrão para modelação de processos de negócio devido à sua simplicidade na forma de representação que permite uma rápida compreensão pelos especialistas do negócio e na facilidade de comunicação entre os vários intervenientes do processo e entre as várias áreas do negócio, aliada ainda a tecnologias que permitem a automatização dos processos modelados. Neste capítulo será abordado como o BPMN vem sendo utilizado para projetar processos de ETL, assim como os respetivos componentes [15].

4.1 Estado da arte BPMN para ETL

A primeira tentativa de utilizar a notação BPMN na modelação conceptual ETL foi proposta em [15]. Os autores afirmam que o processo ETL pode ser considerado um tipo particular de processo de negócio e que pode facilitar a comunicação tanto com o pessoal técnico como não técnico. Os autores descreveram o processo de formação do modelo conceptual e a conversão de BPMN para *BPEL*. Esta conversão foi feita para executar o modelo projetado, bem como implementar os relacionamentos com *web services.* Seguindo o mesmo propósito, os autores apresentaram em [10] uma atualização ao modelo com um metamodelo BPMN

para suportar a visão conceptual de ETL. Numa continuação do seu trabalho, os autores, apresentaram em [43] um metamodelo BPMN independente baseado em *Model-Driven Development* (MDD) onde é suportada a geração automática de código para plataformas comerciais. Após esse trabalho, os autores prosseguiram com o tema em [44] explorando transformações *Model-to-Text* e *Model-to-Model*.

Ainda seguindo a proposta de modelação através de BPMN em [48], [32] , os autores projetaram um metamodelo ETL generalizado para algumas tarefas específicas através da notação BPMN, o que possibilita a modelação e validação dos processos, antes de se proceder à sua implementação. Este trabalho complementa e estende o trabalho de [48], [49] incorporando modelos conceptuais específicos como padrões para atividades de ETL como CDC, *Slowly Changing Dimensions* (SCD), *Surrogate Key Pipelining* (SKP) ou *Data Quality Enhancement* (DQE). Os autores utilizaram um Diagrama de Colaboração BPMN para representar a interação de componentes independentes (padrões), fornecendo uma primeira abordagem para um sistema ETL multicamadas usando BPMN, e mostrando de forma mais simples como os padrões ETL podem ser utilizados para suportar a modelação concetual ETL com BPMN. Essa base pode ser estendida para criar mais padrões BPMN cobrindo todas as atividades de um fluxo de trabalho ETL, o que ajuda no desenvolvimento de um fluxo de trabalho ETL de alta qualidade, diminuindo o número de erros e tornando o processo mais eficiente. Por fim, os autores validaram o modelo com um caso de estudo através da criação de esqueletos ETL que consistem em sistemas ETL que foram gerados a partir de um modelo conceptual em BPMN. Além disso, os autores avançaram no seu trabalho para o processo de tradução automática de modelos conceptuais para a sua implementação em [45] e [46].

Mais recentemente em, [47] os autores deram continuidade a esta abordagem através da representação da modelação concetual de ETL em diferentes camadas, cada uma representando um detalhe de processo diferente. Esta abordagem tem em vista fornecer à equipa de desenvolvimento de ETL ferramentas específicas para a comunicação em diferentes fases de desenvolvimento do ETL. Esta abordagem contribui para uma abordagem de desenvolvimento mais ágil, já que os modelos podem enriquecer os requisitos do sistema de forma incremental. De forma a demonstrar a utilização da técnica, os autores exploraram um subprocesso específico de um cenário ETL e demonstraram como as especificidades de ETL podem ser representadas num nível concetual.

4.2 Aplicação dos elementos BPMN ao ETL

Como referido no capítulo 2, a BPMN é composta por quatro tipos de elementos: objetos de fluxo, objetos de ligação, *swimlanes* e artefactos. Neste subcapítulo é apresentada a utilização e aplicabilidade destes elementos ao contexto ETL.

4.2.1 Objetos de fluxo

Uma **atividade ETL** representa uma unidade de trabalho, ou seja, é o objeto central que descreve tudo o que é executado no fluxo de trabalho. Pode ser modelada em BPMN como uma tarefa, quando é uma atividade unitária que não é subdividida, ou um subprocesso para representar atividades compostas ou uma hierarquia de tarefas no fluxo de trabalho.

Para suportar diferentes níveis de abstração, podemos usar *collapsed subprocesses* (subprocessos recolhidos) BPMN que podem ser aplicados com sucesso a processos ETL. Isso proporciona a conceptualização do processo, uma vez que tarefas mais complexas são decompostas em diferentes níveis. Isso é muito vantajoso para apresentar, discutir e entender as principais tarefas do processo [50].

Para representar um ciclo é utilizada uma *task* ou *subprocess* com o respetivo símbolo. Um ciclo é um recurso de controlo que leva à execução de uma tarefa ETL repetidamente enquanto a condição for verdadeira. As condições podem ser verificadas antes ou depois da atividade. Para declarar a condição é utilizado o elemento de anotação. O ciclo é útil no contexto ETL, especialmente para tarefas de *pipelining* que são executadas linha por linha.

Os ***gateways*** são utilizados para controlar a sequência de atividades num processo ETL com base nas condições. Podem incorporar a condição ou estar vinculados a um artefacto de condição de *gateway* que inclui a condição. O BPMN define vários tipos de *gateway*, como exclusivos, inclusivos, baseados em eventos ou baseados em dados; além disso, eles podem ser de divisão ou de junção. Segundos os autores em [15] os tipos mais usados num contexto ETL são os *gateways* exclusivos baseados e os *gateways* paralelos. Um *gateway* exclusivo modela uma decisão: dependendo de uma condição de dados, onde o *gateway* ativa uma ou mais das suas ramificações de saída. Um *gateway* paralelo expressa a sincronização entre os fluxos de entrada como uma condição para ativar os fluxos de saída.

Os **eventos** representam algo que acontece que afeta a sequência e o tempo das atividades do fluxo de trabalho. Esses eventos podem ser internos ou externos à atividade em consideração. Podem ser categorizados em eventos iniciais, intermediários ou finais. Os eventos podem ser colocados ao longo do *workflow* do processo, como uma atividade, ou no limite da forma da atividade ou do subprocesso. Existem utilizações típicas de eventos num contexto ETL. Por exemplo, um evento de início do tipo *Timer*, pode ser utilizado para representar a execução periódica de um processo ETL, enquanto um evento de início normal pode ser utilizado se o processo for simplesmente disparado pelo final de seu processo predecessor. Além disso, outros eventos comuns incluem eventos de erro, mensagem, cancelamento e compensação.

O tratamento de erros pode ser representado por um evento intermediário específico. Caso surjam erros no processo, a sua notificação é realizada através de uma ação explícita: a atividade é cancelada e é enviada uma mensagem; ou por uma ação implícita: que será definida nas próximas etapas do processo de desenvolvimento. Além de detetar erros, um evento de compensação (*Compensation Event*) também pode ser utilizado para recuperar erros, lançando atividades de compensação específicas, que estão vinculadas ao evento de compensação com o objeto de ligação da associação. Por exemplo, um evento de erro pode enviar um alerta por e-mail a notificar a falha de uma tarefa ou processo, enquanto um evento de compensação tentará corrigir um erro ao executar uma atividade adicional antes de reiniciar a execução à parte interessada do fluxo de trabalho [15].

4.2.2 *Connecting Objects*

Os objetos de fluxo ou ligação têm vindo a ser utilizados no contexto ETL como o padrão referido pela OMG na notação BPMN. Permitem modelar o fluxo entre as diversas atividades, ou seja, nada mais foi acrescentado para a utilização destes elementos na modelação de ETL.

Como referido anteriormente no subcapítulo 2.1.5, existem em BPMN, três tipos de objetos de ligação: fluxo de sequência, fluxo de mensagens e associações. Um fluxo de sequência representa as restrições de sequenciamento entre os objetos de fluxo. É o objeto de ligação básico e essencial num fluxo de trabalho ETL. Na figura 24 estão ilustradas cinco atividades que estão ligadas através de fluxos de sequência, a atividade de destino começará somente quando a de origem for concluída, por exemplo, a atividade “Load DimState” só tem início

após a conclusão da atividade “Load DimCountry”. Se vários fluxos de sequência emergem de uma atividade, todos eles serão ativados após sua execução. Caso haja a necessidade de controlar um fluxo de sequência, é possível adicionar uma condição ao fluxo de sequência através da utilização de um fluxo de sequência condicional ou de um *gateway*.

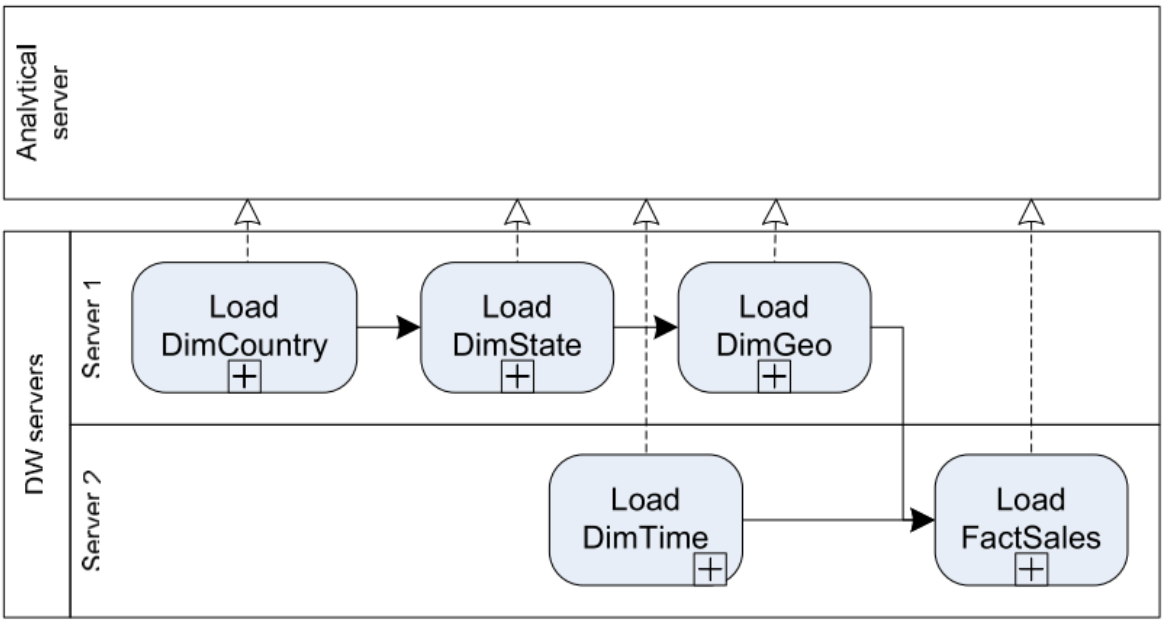

Figura 24 - Exemplificação dos objetos de ligação [15]

Na figura 24 está ilustrada a utilização dos fluxos de mensagens que são usados para representar o envio e a receção de mensagens entre as duas *swimlanes*. Num diagrama BPMN, *swimlanes* separadas representam entidades diferentes. No exemplo ilustrado existem as *swimlanes* “DW servers” e “Analitical server”. O fluxo de mensagens é o único objeto de ligação permitido para atravessar os limites de uma *swimlane*. Todas as atividades representadas comunicam com a *swimlane* “Analitical server” através dos fluxos de mensagens.

Uma associação serve para relacionar anotações a objetos de fluxo, e no processo ETL pode ser utilizado, por exemplo, para ligar as anotações à respetiva atividade .

4.2.3 *Swimlanes*

Uma *swimlane* como referido no subcapítulo 2.1.6 é um objeto estruturante que permite a definição dos limites do processo. No entanto, uma *pool* pode ser subdividida em várias *lanes*,

que representam funções ou serviços na empresa. Os autores [15], referem que *swimlanes* permitem a organização e a hierarquização de vários processos ETL diferentes e multiníveis para povoar ou atualizar um DW. As *lanes* permitem que os processos ETL sejam organizados de acordo com várias estratégias [15]:

- Arquitetura técnica - por exemplo, localização de tarefas em servidores, aplicações, interfaces;
- Por perfil de utilizador - por exemplo, um gestor, analista ou designer que possui direitos de acesso especiais relativamente a outros utilizadores, ou
- Por entidades comerciais - por exemplo, um departamento ou empresa.

4.2.4 Artefactos

Uma **anotação** é utilizada para expressar a semântica sobre objetos de fluxo. Segundo a abordagem dos autores [15], as anotações são especializadas em dois dos objetos já referidos: anotações de dados e condições de *gateway*. As anotações de dados são utilizadas para tornar explícita a semântica de uma tarefa ETL. Através das anotações é feita a menção a: dados de entrada e saída da tarefa, parâmetros, que se referem a dados adicionais utilizados pela tarefa, pré e pós-condições, comentários e qualquer outra semântica útil. Dependendo se a tarefa é uma operação de linha ou conjunto de linhas, os dados de entrada e de saída podem ser uma tabela ou uma linha.

É possível referir que a utilização de artefactos é meramente descritiva, ainda assim é de elevada importância, pois na grande maioria das vezes o nome da atividade e o tipo de tarefa não é suficiente para documentar realmente em que consiste a atividade e de que forma é executada. Para melhorar a documentação do processo, os autores [15] utilizam nas anotações de texto associadas a uma atividade: *input*, *output* e condições (ver figura 25).

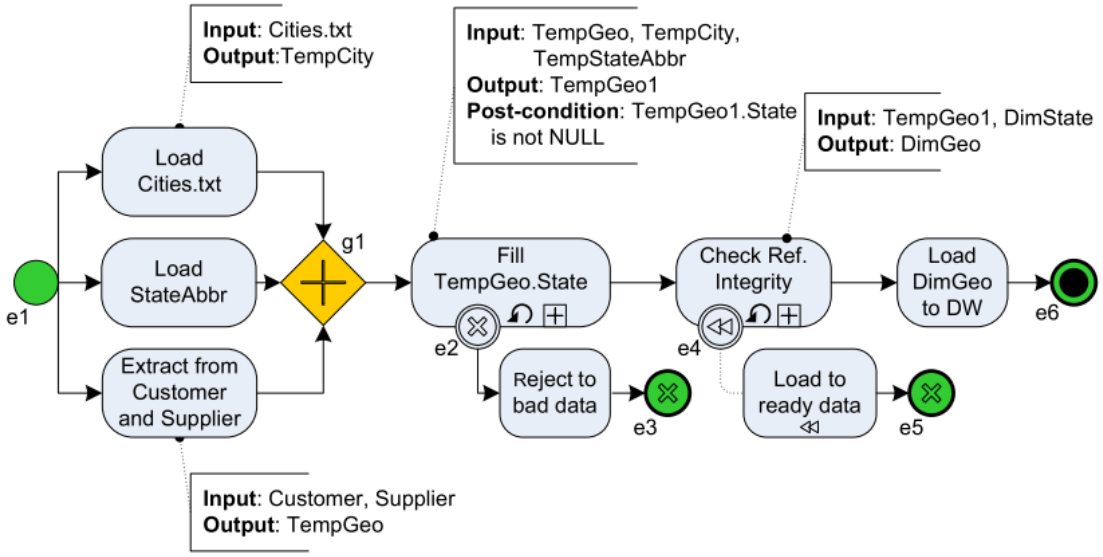

Figura 25 - Exemplo do fluxo de trabalho ETL [15]

Relativamente aos *gateways* os autores [15] acrescentam ainda que uma condição de um *gateway* indica uma expressão condicional que impõe algum controlo sobre o fluxo, por exemplo, decisão ou sincronização. Em alguns casos, a semântica do *gateway* é expressa graficamente e não precisa de uma condição de *gateway* explícita. Referem que as condições do *gateway* incluem dois recursos: condição, declarando a expressão condicional do gateway, e comentários, que permitem expressar qualquer aspeto útil do *gateway*.

4.2.5 Objetos de dados

Segundo os autores em [15] um objeto de dados é utilizado tradicionalmente para representar documentos em trânsito entre tarefas por exemplo: faturas e/ou contratos. No contexto ETL, os objetos de dados devem ser utilizados para representar bases de dados, ficheiros e documentos, quer sejam eles fontes de dados, dados temporários ou bases de dados do DW.

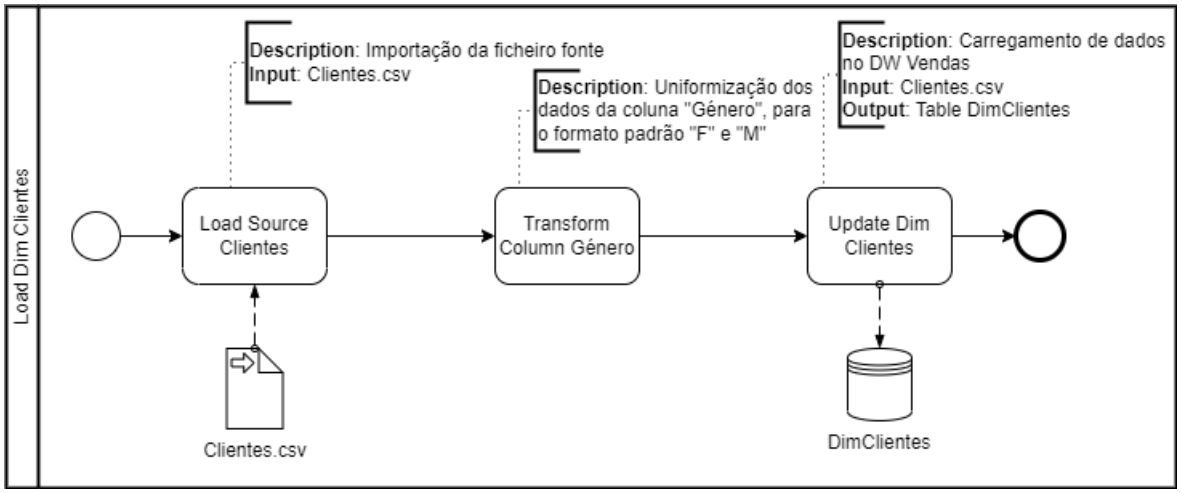

Figura 26 - Modelação BPMN do exemplo apresentado em 3.3

Seguindo o exemplo apresentado no subcapítulo 3.3, que se refere à implementação de um DW de Vendas nomeadamente na implementação da dimensão Cliente, na figura 26 é proposta a modelação através de BPMN do processo ETL para o povoamento da dimensão clientes tendo como base os trabalhos apresentados para o estado da arte BPMN para ETL referidos em 4.1. O processo de povoamento da dimensão Cliente desenvolve-se dentro de uma *swimlane* e inicia-se pelo evento de início. A primeira tarefa ETL deste processo consiste na extração dos dados do ficheiro “Clientes.csv” e está representado no diagrama BPMN pela tarefa “Load Source Clientes” e pelo objeto de dados “Clientes.csv”. estes dois elementos estão ligados pelo fluxo de mensagem. Através do elemento anotação é feita uma descrição acerca da tarefa e o input de dados necessário à sua execução. A ligação do elemento anotação à tarefa é realizado através do fluxo de associação. A segunda tarefa consiste na uniformização dos dados da coluna “Género” onde os dados são transformados para “F” e “M”. A última tarefa ETL deste processo consiste no *Update* da Dimensão Clientes, ou seja, o carregamento dos dados já uniformizados para a base dade dados “*DimClientes*”. O processo termina com o evento de fim.

4.3 Modelação ETL em camadas

Em [50] os autores apresentam uma abordagem de modelação conceptual BPMN para modelação de processos ETL através da utilização de três diferentes camadas de abstração. Tendo em conta a expressividade do BPMN, que pode ser muito útil para a representação de ETL, os modelos conceptuais de ETL podem variar significativamente. Os autores referem

que nos últimos anos a modelação de ETL baseada em BPMN está focada na semântica da linguagem e não numa metodologia que possa ser utilizada para representar o desenvolvimento de ETL em diferentes fases. Esta abordagem resulta em processos BPMN onde são misturados vários níveis de detalhe, o que pode dificultar a interpretação e compreensão do processo. Nesse sentido, os autores abordam a representação conceptual ETL em diferentes camadas, cada uma representando um nível de detalhe de processo diferente que é aplicado a uma construção específica descrita na camada anterior.

Abstraction level	Purpose	Main BPMN artifacts
Process	Representing system abstraction and providing process dependencies description.	Subprocesses are used to represent data flows for DW populating processes.
Pattern	Representing the macro activities presented in the ETL system regarding extraction, quality and load tecnhiques.	Subprocesses represent common ETL procedures/sub systems.
Task	Represent the elementary level for ETL representation using (mainly) atomic tasks. Processes are represented in a row-by-row processing.	Task are a predominant modelling artifact

Figura 27 - Resumo das camadas de abstração para modelação conceptual ETL[50]

Na figura 27 os autores apresentam uma visão geral das três diferentes camadas de abstração. Na primeira coluna identificam o nível em que os processos ETL podem ser representados. No nível de Processo os autores definem uma visão geral dos principais processos do sistema ETL, o que pode representar apenas as dependências entre dimensões e processos de preenchimento de tabela de fatos e a descrição de subprocessos relacionados a cada objeto de dados. Definem para uma possível utilização, a identificação da necessidade de aplicar de uma técnica *Slowly Changing Dimension* (SCD) ou para identificar as restrições aplicadas à utilização de tabelas ponte.

No nível Padrão, a equipa utiliza componentes configuráveis adicionais, conhecidos como padrões que deve ser incluída na documentação do projeto. Tem como objetivo fornecer uma forma mais direta de descrever os principais componentes ETL sem especificar como esses procedimentos serão implementados. Por exemplo, neste nível pode ser identificada a

necessidade de aplicar um mecanismo específico de *Change Data Capture* (CDC) ou SCD, sem detalhar como será tratado.

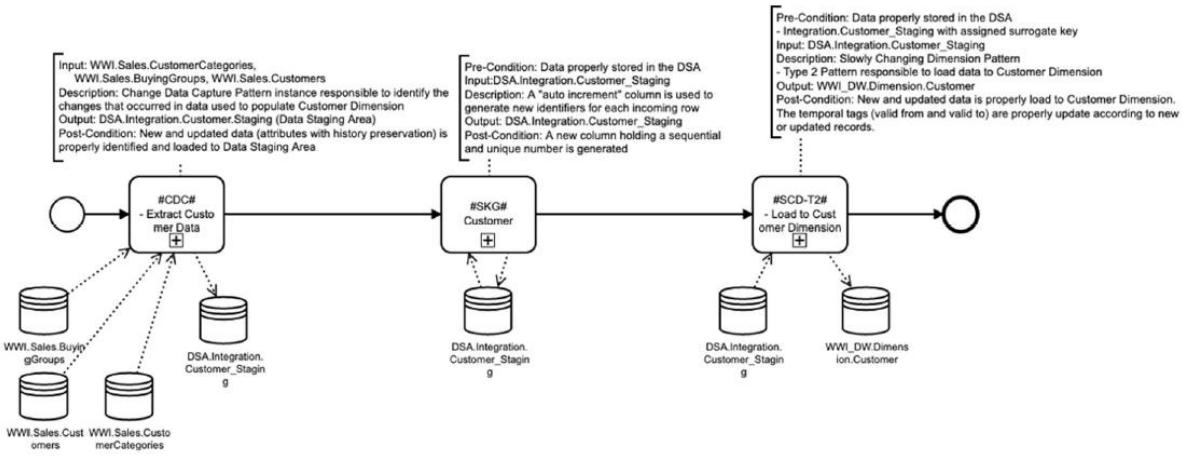

Figura 28 - Modelo conceptual BPMN com representação de padrões [50]

Nesse nível (ver figura 28), as atividades são documentadas para identificar entradas, saídas e possíveis abordagens de tratamento de erros numa visão de alto nível. Os objetos de dados também são identificados, para revelar mais detalhes a complexidade do sistema ETL e são utilizadas técnicas específicas utilizadas (por exemplo, para CDC ou SCD).

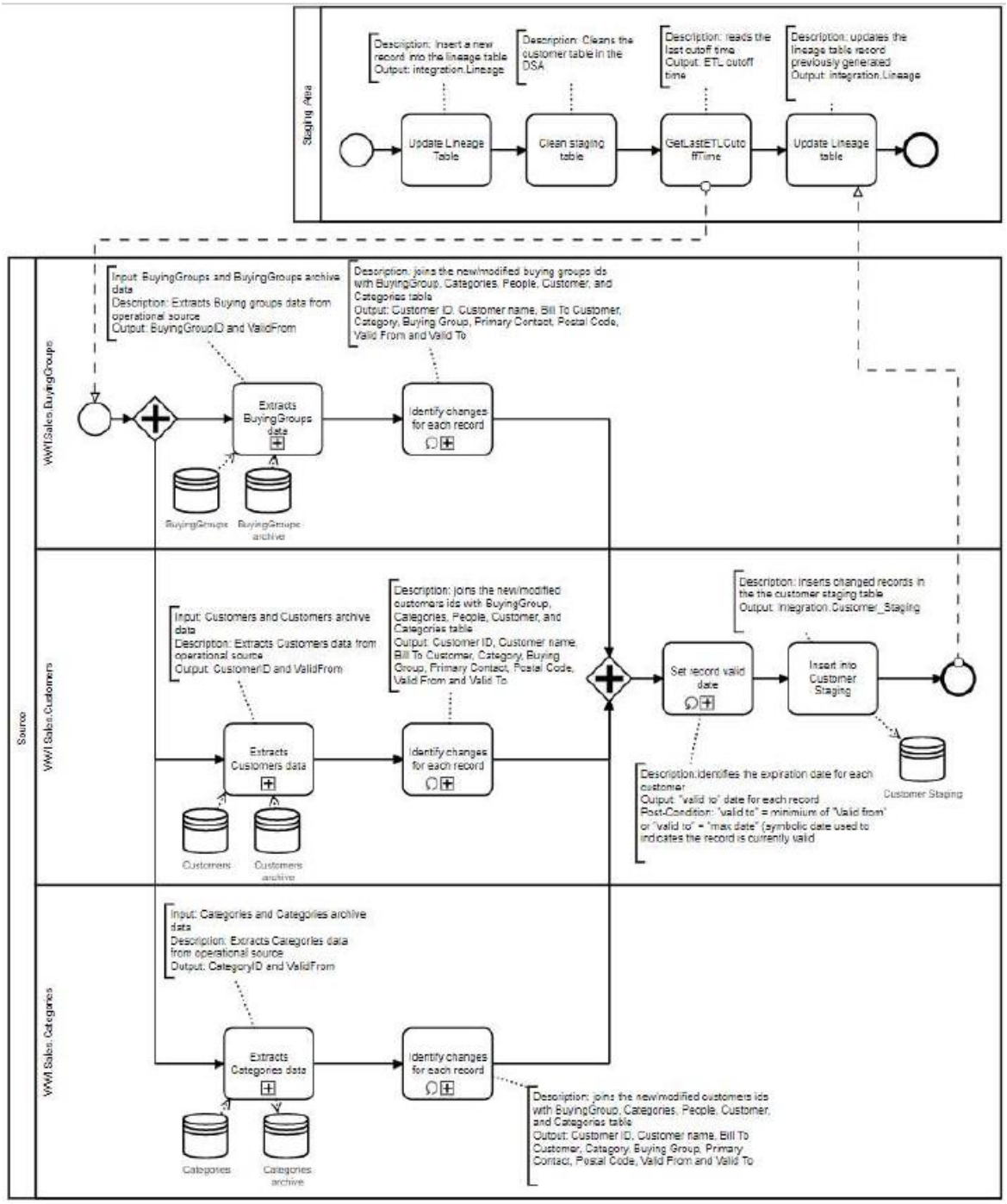

Figura 29 - Nível de processo ETL elementar [50]

Por último, o nível Tarefa (ver figura 29) representa principalmente as tarefas BPMN que descrevem o algoritmo para implementar os padrões identificados no nível Padrão. As tarefas de representação como junções, ou seleções podem ser utilizadas num nível lógico para descrever como cada uma das tarefas será implementada.

Capítulo 5 – Proposta de modelação

A apresentação de convenções para a modelação de processos ETL com BPMN, expõe as melhores práticas de modelação através de uma proposta baseada em três níveis. A elaboração da proposta visa gerar *templates* gerais que descrevem como um problema deve ser resolvido independentemente do contexto em que será aplicado e independentemente da ferramenta de implementação física.

A elaboração de modelos abstratos permite melhorar a compreensão do processo, por todas as partes envolvidas sejam empresários, analistas de negócio ou utilizadores mais técnicos [15]. Principalmente nas fases iniciais de desenvolvimento, os modelos conceptuais desempenham um papel extremamente importante, onde os utilizadores validam os requisitos de negócio. O BPMN fornece uma notação muito simples e poderosa para a representação de processos que se adequa perfeitamente a processos como o de ETL e além do seu grande poder de expressividade disponibiliza ainda mecanismos que possibilitam a sua execução. Além disso, é possível identificar como vantagem o facto de os utilizadores de negócios já estarem familiarizados com a notação e o facto dos processos de negócio já implementados nas empresas poderem ser aproveitados para entender a lógica dos processos e do fluxo de dados [51].

No entanto, a adoção de uma notação para a modelação de processos ETL com BPMN não é direta e devem ser adotadas convenções que forneçam uma orientação para guiar o processo de modelação, evitando que existam diferentes representações do processo que contribuam para uma interpretação inadequada.

Como referido no capítulo 2, devido ao crescente número de conceitos em BPM foram utilizados três níveis de representação [52]: modelação descritiva, modelação analítica e modelação executável. Esta abordagem descreve o nível através do qual cada processo está representado, permitindo uma modelação progressiva e cada vez mais rica dependendo das necessidades em cada fase do projeto. Considerando estas características e a adequabilidade da utilização de BPMN no contexto de ETL (discutida e apresentada anteriormente), é apresentado um estudo da aplicação desta abordagem no contexto de ETL. A aplicação desta metodologia visa aplicar os princípios de criação de modelos BPMN ao ETL em diferentes camadas, com objetivos de clareza, expressividade e consistência na especificação e consequente implementação de processos.

Seguindo a abordagem descrita em [52], é proposta uma abordagem de modelação BPMN para ETL *top-down*, resultando em modelos estruturados hierarquicamente. Através do estudo e análise dos métodos de modelação de processos BPMN apresentados pelo autor foi verificada a aplicabilidade dos métodos e regras ao processo ETL. Após essa análise foram selecionados e adaptados os métodos e as regras e ajustáveis ao processo ETL e identificados os elementos que constituem cada nível de modelação. Foram reajustados os níveis de modelação retirando o nível de modelação executável, tendo em conta que não se torna particularmente útil no contexto de ETL. A aplicação de princípios básicos de composição e a utilização de elementos visa implementar um conjunto de “melhores práticas”, de modelação conceptual ETL.

5.1 Nível 1 - Modelação Descritiva

O conjunto de elementos de Nível 1 representa as formas e símbolos tipicamente utilizados do fluxograma tradicional e é suficiente para descrever a maioria dos comportamentos do processo de uma forma compacta e amigável.

O Nível 1 visa implementar uma documentação simples do fluxo do processo e consiste numa modelação de alto-nível. Este nível é composto por 3 componentes: a palete de elementos, o método e o estilo de *design*, através dos quais são desenvolvidos os diagramas de modelação descritiva.

5.1.1 Conjunto de elementos do Nível 1

Ao conjunto de formas e símbolos que fazem parte da palete de Nível 1, o BPMN 2.0 denomina de subclasse de conformidade de modelação de processos descritiva. Através da palete de nível 1 é possível modelar praticamente qualquer processo ETL, com exceção da representação de fluxos de mensagem externos.

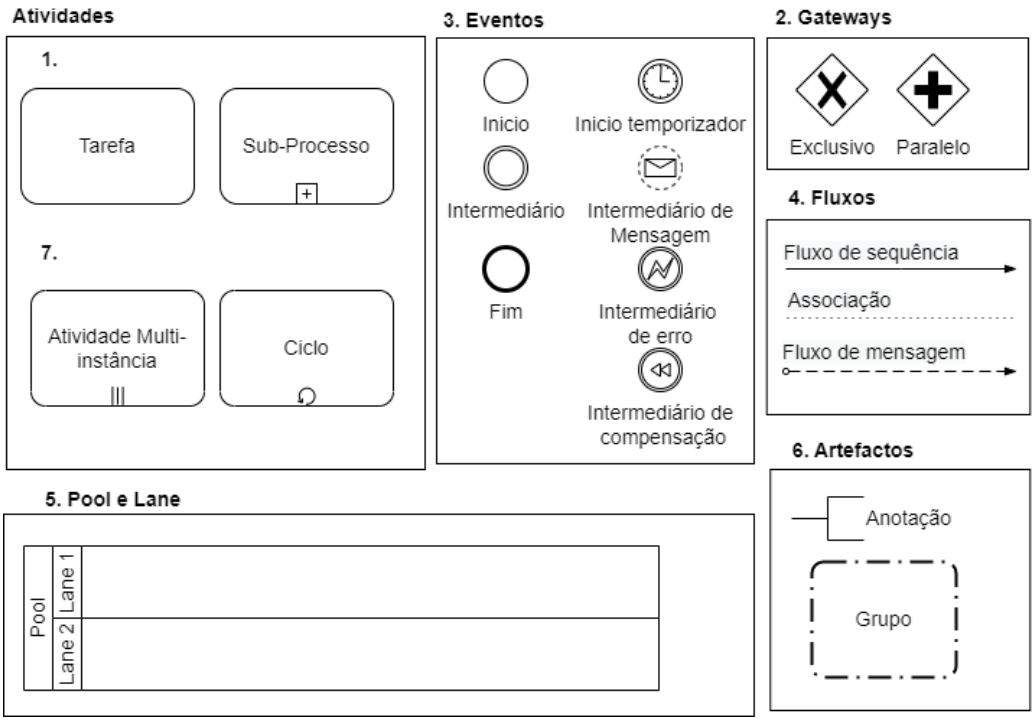

Figura 30 - Conjunto de elementos do Nível 1

O conjunto de elementos apresentado na figura 30 consiste no grupo de elementos BPMN que constituem o conjunto de elementos de Nível 1 [52]:

1. Atividades: Tarefa e subprocesso.

Para representar atividades de ETL atómicas é utilizado o elemento BPMN “tarefa” e para representar atividades compostas é utilizado o elemento “subprocesso”. Ambos podem ser enriquecidos através da adição de marcadores (ver capítulo 2) para descrever atividades especificas. Além disso, os subprocessos podem devem ser utilizados para expressar uma estrutura hierárquica no fluxo de trabalho, para que a representação de processos de se torne mais legível.

2. *Gateways*

Os *gateways* (Exclusivo e Paralelo) devem ser utilizados para controlar o fluxo de sequência do processo ETL com base nas condições específicas estabelecidas. Um *gateway*, um elemento em forma de diamante, “controla” o fluxo do processo, dividindo-o em caminhos alternativos. O *gateway* Paralelo é utilizado para identificar caminhos independentes e que podem ser desencadeados em paralelo, enquanto que o *gateway* Exclusivo ou *gateway* XOR

significa que apenas um dos fluxos de saída pode ser executado como resultado da validação de uma condição.

3. Eventos de início, eventos intermediários e eventos de fim.

Os eventos devem ser utilizados para descrever algo que acontece e afeta o comportamento das atividades no fluxo de trabalho ETL. Neste nível devem ser utilizados os eventos de início, eventos intermediários e eventos de fim interligados às atividades anteriores e subsequentes através do fluxo de sequência. Os eventos de início e fim como servem para dar início ao processo e terminar o processo com os respetivos nomes indicam. Os eventos intermediários ocorrem após o início de um processo e antes do seu final e permitem descrever comportamentos adicionais, como erros ou exceções que ocorrem durante o processo ETL.

O significado preciso de um evento intermediário depende dos detalhes da sua representação. O ícone do elemento e a cor desse ícone significam diferentes comportamentos acionados para um determinado sinal de acionamento. Os eventos intermediários a serem utilizados neste nível são o: o evento intermediário de *throwing* e o evento intermediário de *catching,*

4. Fluxo de sequência, fluxo de mensagem e associação

O fluxo de sequência e o fluxo de mensagem devem ser utilizados para estabelecer relações entre os elementos BPMN. Servem para descrever a sequência pela qual as atividades são executadas e os possíveis caminhos que podem ser percorridos.

Uma associação deve ser utilizada para relacionar as anotações de texto aos objetos de fluxo no processo ETL.

5. *Pool* e *Lane*

Os *Pools* e *Lanes* são utilizados para representar os diferentes papéis do processo ETL. Esses papéis permitem a representação do processo através de múltiplas estratégias. Por exemplo: representar o processo de transformação de dados associados a diferentes repositórios; a representação de diferentes entidades num processo; ou cobrindo uma tarefa ETL especifica que careça de representação a esse nível, como é o exemplo a modelação de um processo ETL onde é utilizada uma *lane* para cada fase ETL (extração, transformação e carregamento).

6. Artefactos: anotação de texto básica e grupo.

O artefacto “Anotação de texto” deve ser utilizado para adicionar detalhes extra aos objetos BPMN que representam o fluxo de trabalho ETL. Por exemplo para descrever as condições de um *gateway* ou para rotular os diferentes eventos de fim.

Um fluxo de sequência de um processo pode conter muitos componentes, o que dificulta a sua visualização no ecrã da ferramenta de ETL, como por exemplo na ferramenta SSIS. No ecrã de desenho desta ferramenta, nomeadamente no separador Data Flow, onde é implementado todo o fluxo do processo ETL, é possível agrupar uma série de elementos do processo, através da utilização do objeto “Grupo”. Para efetuar a modelação desse elemento em BPMN é utilizado o artefacto grupo.

7. Atividade de *loop* (ciclo) e atividade multi-instância.

Como referido no capítulo 2, as atividades de ciclo e as atividades multi-instância consistem num tipo específico de atividade que é utilizada para lidar com coleções de objetos ou tarefas que carecem de repetição. Por exemplo, o elemento ciclo" pode ser utilizado para descrever um processamento de dados linha a linha tendo como termino ou continuação uma condição de verificação. O marcador de múltipla instância pode ser utilizado para representar várias instâncias de uma atividade que lidam com objetos de dados específicos dos quais a quantidade de objetos a tratar é previamente conhecida.

Relativamente ao esquema dos diagramas não existe uma necessidade de especificação gráfica em termos de cores, fontes e tamanhos dos vários elementos, até porque cada ferramenta de modelação possui um esquema específico diferindo de ferramenta para ferramenta. O importante a nível de esquema é manter um padrão consistente a nível de representação dos vários processos, ou seja, deve ser selecionado um padrão e mantê-lo durante a representação dos vários processos. Desta forma é garantida a uniformização e a compreensibilidade dos modelos.

5.1.2 Método

Através dos elementos de trabalho já definidos para o Nível 1, é possível lidar com os requisitos de modelação da maioria dos processos. Não existem métodos oficiais para suportar a modelação BPMN, uma vez que a sua utilização se destina a uma ampla variedade

de cenários por pessoas com interesses e habilidades divergentes. À semelhança disso encontra-se a modelação de processos ETL para a qual têm sido propostas várias opções de modelação e nenhuma delas foi até ao momento adotada como padrão ou pelo menos se revele como uma abordagem de eleição entre a comunidade de investigação e profissional associada ao desenvolvimento de sistemas de ETL.

Os utilizadores que modelam os processos ETL através de BPMN têm como objetivo a elaboração de representações visuais (diagramas) que transmitam a lógica do processo e os requisitos subjacentes. Mas o facto de cada utilizador poder aplicar os seus próprios significados e interpretações nos diagramas que produz tem a desvantagem de que nem todos os utilizadores os interpretem da mesma forma o que promove dificuldades na comunicação dos processos [16]. A proposta de modelação apresentada neste capítulo visa colmatar essa dificuldade através de uma tentativa de padronizar a modelação de processos ETL. A implementação deste método torna-se uma mais-valia para maximizar o entendimento dos diagramas dos processos ETL pois consiste numa estrutura organizada que contém todas as informações relevantes para o desenvolvimento do sistema.

O método consiste na elaboração de um modelo completo, consistente e bem estruturado, baseado num estilo de modelação hierárquica. O autor refere em [52] que os objetivos do método devem seguir os princípios do “bom BPMN”, que são nomeadamente:

- Completude: os elementos essenciais da lógica do processo de ponta a ponta devem ser capturados no diagrama, incluindo como o processo é iniciado, os seus estados finais, o que a instância representa e sua interação com entidades externas.
- Clareza: os detalhes do fluxo do processo, as atividades condicionais, as atividades que são executadas em paralelo e como são tratadas as exceções. Além disso, todos os elementos do diagrama devem ter uma interpretação clara, mesmo para aqueles que não estão familiarizados com seu processo nem com sua terminologia.
- Partilha entre negócios: o BPMN como linguagem pode ser partilhada por utilizadores de negócios, analistas de negócios e programadores.
- Consistência estrutural: todos os modeladores devem, idealmente, seguir a mesma estrutura geral na criação de modelos.

Modelação Hierárquica *Top-Down*

O método descrito descreve um estilo de modelação hierárquica *top-down*. O modelo hierárquico representa graficamente o processo completo como um conjunto de diagramas de processos vinculados que representam níveis de processos distintos. Ou seja, um subprocesso recolhido no diagrama de nível "pai" é expandido num diagrama de nível "filho" separado. Os subprocessos recolhidos nesse nível "filho" podem ser expandidos ainda mais noutro diagrama com um relacionamento "neto" com o primeiro. Um único diagrama de nível-superior fica no topo da hierarquia e o número de níveis aninhados abaixo dele é ilimitado. Em contraste, um modelo de processo plano coloca todas as etapas do processo, mesmo os detalhes mais subtis, num único diagrama [52].

A modelação hierárquica significa que o modelo é representado visualmente por vários diagramas. Os diagramas não são modelos separados, apenas visões separadas e parciais de um único modelo.

Top-down significa que a modelação do processo começa por ilustrar o processo completo, enumerando as suas principais etapas num mapa de alto-nível. A partir daí, é detalhada a lógica interna de cada atividade num diagrama de nível filho, revelando apenas os detalhes necessários para sua finalidade. A abordagem *top-down* força o *designer* a começar com o quadro geral, adicionando apenas os detalhes necessários para o propósito imediato.

Passo 1 – Determinar o escopo do processo

A modelação *top-down* começa com a definição do escopo do processo, ou seja, onde o processo se inicia e onde termina. Um processo ETL tem uma estrutura semelhante, ou seja, um conjunto de atividades bem definidas com início e fim determinados. Para auxiliar na definição do processo deve ser considerado:

- O início do processo e como este será iniciado. Tipicamente, no contexto dos processos ETL, os processos iniciam-se de acordo com uma periodicidade (por exemplo, todos os dias a uma determinada hora ou semanalmente).
- Fim do processo: Consiste em determinar quando o processo está completo. Um processo de ETL termina, principalmente neste nível de conceptualização, com o povoamento de uma dimensão ou de uma tabela de factos, o que não implica (pelo

menos na maioria dos cenários) a necessidade de executar processos externos relacionados.

- Objetivo do processo: Pode ser associado ao povoamento de um objeto de dados (dimensão ou tabela de factos), representando todas as etapas de alto nível necessárias para atingir esse objetivo.
- Identificação das diferentes formas de terminar o processo: Consiste em definir se o processo contém mais de um estado final. Por exemplo, o processo de integração de dados pode falhar devido à existência de dados não conformes para os quais não foi determinada uma exceção. Nesse caso devemos dizer que o processo terá dois estados finais, ou seja, o evento final pré-determinado e outro evento final para a ocorrência de erros. Portanto devem ser definidos os eventos finais necessários conforme a operação o necessite.

Neste passo é criado um diagrama. As considerações apresentadas são discutidas pela equipa e são devidamente documentadas.

Para ilustrar o método é possível apresentar um processo de povoamento de uma dimensão "Cliente" enquadrada num *Data Mart* direcionado a suportar os principais processos de tomada de decisão relacionadas com vendas a retalho.

1. Início do processo: O processo de povoamento na dimensão cliente é realizado diariamente às 23:00 horas.
2. Fim do processo: O processo conclui-se após o carregamento dos dados na dimensão cliente no *Data Mart* de vendas.
3. Objetivo do processo: O processo consiste na extração de dados relacionados com clientes da fonte operacional, a aplicação de processos de uniformização e limpeza de dados de forma a garantir não só os requisitos de qualidade dos dados, mas também da sua estrutura, de acordo com os requisitos definidos na construção da dimensão Cliente.
4. Nomeação das diferentes formas de terminar o processo: Para o processo descrito apenas existe um evento final pré-determinado que se segue após a conclusão da última tarefa do processo.

Passo 2 – O mapa de alto-nível

O próximo passo no método passa por definir mapa de alto-nível que consiste na identificação das principais atividades do processo, idealmente dez ou menos que posteriormente serão utilizadas para a elaboração do diagrama BPMN.

Como o mapa de alto-nível representa apenas uma lista das principais atividades do processo. As atividades no mapa de alto-nível representam *containers* em que detalhes mais específicos serão posteriormente adicionados. As atividades identificadas nesta fase correspondem a atividades BPMN, identificando ações realizadas repetidamente com um início e fim bem definido.

Para ilustrar o segundo passo do método é apresentado um exemplo da elaboração do mapa de alto-nível para o processo de povoamento da dimensão cliente. A elaboração deste mapa nada mais é do que a identificação do cenário. Diz respeito ao nome no processo, ou seja, a definição do objetivo para o qual é criado e a identificação das suas principais atividades, referindo o objetivo de cada uma delas e de que forma terminam.

Cenário: Povoamento da dimensão Cliente

Atividades identificadas:

- Extrair dados de clientes: Extrair os dados das tabelas: Customer, Address, City e Country.

Estados finais: “Dados de clientes e das suas moradas extraídos da fonte”

- Limpar e converter dados: Limpeza e uniformização de dados. Esta atividade só é realizada quando os clientes forem extraídos e é responsável por conversão dos valores de texto para minúsculas, e de seguida a tarefa responsável pela concatenação do atributo primeiro_nome com o atributo ultimo_nome.

Estado final: “Dados uniformizados com sucesso”

- Gerar *Surrogate key* (SK): Geração de chaves de substituição. Esta atividade só é realizada quando os dados forem uniformizados.

Estado final: “Chaves de substituição geradas”.

- Carregar DW: Armazenar os dados na dimensão cliente. Esta atividade começa logo que a atividade geração de SK for concluída.

Estado final: “Carregamento concluído”

Passo 3 – diagrama de processo *top-level*

Após a elaboração do mapa de alto-nível, podemos transformá-lo num diagrama BPMN de alto-nível. Cada atividade do mapa de alto-nível torna-se um subprocesso no diagrama. No estilo de modelação hierárquica, cada uma dessas atividades é posteriormente expandida em diagramas de nível filho que mostram os detalhes de cada etapa.

Se existirem atividades condicionais no mapa de alto-nível, cada uma delas é representada seguida de um *gateway* que verifica o estado final da atividade anterior. Se o *gateway* tiver duas saídas, deve rotular-se o *gateway* com um dos nomes dos estados finais da atividade anterior e uma saída com “sim” e a outra “não”.

Se uma atividade for executada simultaneamente com outras atividades no mapa de alto-nível, pode dividir-se o fluxo em caminhos paralelos ou através um *gateway* paralelo ou simplesmente com dois fluxos de sequência da atividade anterior. Se uma atividade exigir a conclusão de duas ou mais atividades paralelas, deve utilizar-se o respetivo *gateway*.

Para ilustrar o terceiro passo do método é apresentado um exemplo da elaboração do diagrama de nível-superior para o processo de povoamento da dimensão cliente em duas fases.

- Fase 1 - Seguindo o mesmo exemplo dos passos 1 e 2 do processo de povoamento da dimensão cliente o resultado do terceiro método é apresentado na figura 31. Cada atividade é modelada com um subprocesso e o início e fim estão bem definidos.

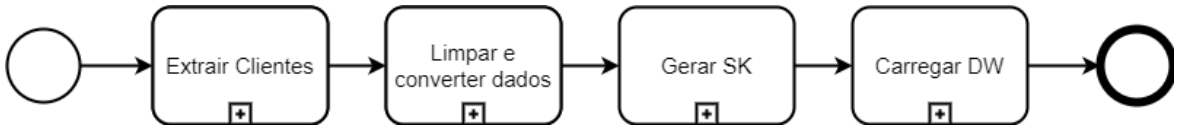

Figura 31 - Diagrama BPMN de nível-superior versão 1

- Fase 2 - Para ilustrar o método de forma abrangente, isto é, apresentar caminhos de exceção, é necessário realizar algumas alterações ao exemplo mencionado. Supõe-

se que o processo de povoamento da dimensão clientes é executado diariamente. Num processo de povoamento incremental é necessário verificar de a SK já foi gerada. Desta forma é necessário modelar essa verificação através de um *gateway* exclusivo. O *gateway* exclusivo é colocado antes da atividade “Gerar SK” para efetuar essa verificação O resultado desta versão é apresentado na figura 32 e a alteração que possui relativamente à versão anterior diz respeito à adição de uma condição antes da atividade “Gerar SK”. O estado final “Chaves geradas” e o estado final “Chaves não geradas”. Desta forma o *gateway* que antecede a atividade é rotulado com uma pergunta e os dois caminhos do *gateway* correspondem aos estados finais.

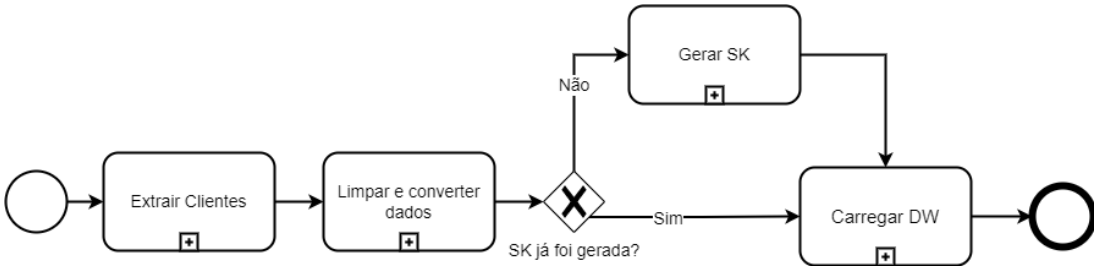

Figura 32 - Diagrama BPMN de nível-superior versão 2

Uma nota importante é referir que muitas vezes é preferível omitir *swimlanes* no diagrama de nível-superior e colocá-las apenas nos diagramas de nível filho, para que o diagrama não se torne demasiado complexo. A figura 33 consiste na mesma representação do processo, mas através da utilização de *swimlanes*, representação esta, que mantém a semântica do modelo.

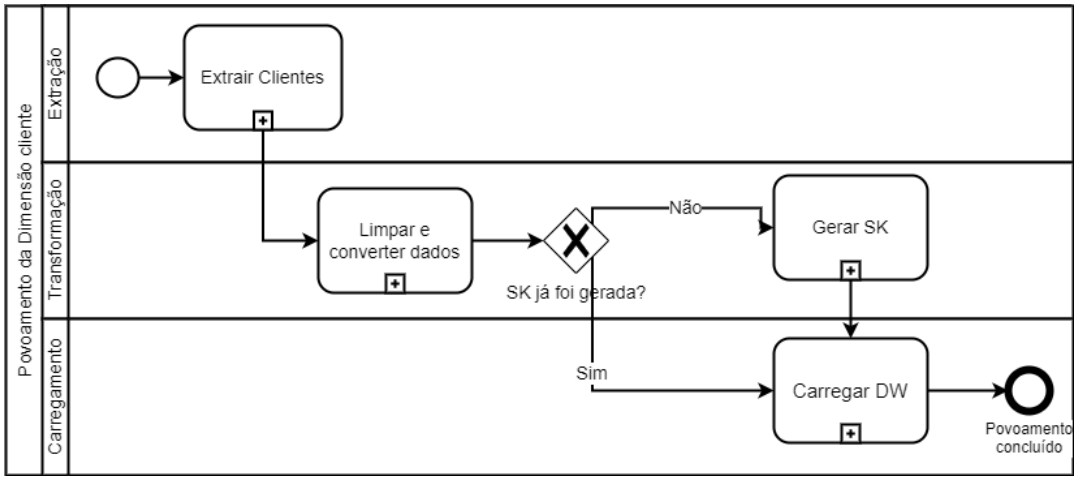

Figura 33 - Diagrama BPMN de nível-superior versão 2 com swimlanes

Passo 4 – expansão *child-level* (nível-filho)

O diagrama de nível-superior tem como objetivo informar como o processo começa e termina assim como das principais atividades associadas. No entanto revela pouco sobre os detalhes internos de cada atividade. Para isso, é necessário mostrar a expansão de nível-filho de cada atividade de nível-superior. Na modelação hierárquica, cada um deles é desenhado num diagrama separado, vinculado a uma atividade de subprocesso recolhido no diagrama de nível-superior.

A expansão do nível-filho deve ter um evento de início sem tipo e as suas atividades na expansão podem incluir também subprocessos que são expandidos noutro diagrama. Pode também enquadrar-se o processo de nível-filho no contexto de uma *pool*, devendo neste caso ser identificada com o mesmo nome do subprocesso recolhido. As *lanes* são definidas nos vários tipos de diagramas independentemente do nível de processo.

De forma a ilustrar o passo número 4 do método foi considerada a atividade “Limpar e converter dados” representada na figura 33. Esta atividade consiste num subprocesso representado na figura 34. Possui duas tarefas, a conversão dos atributos de texto para minúsculas, e de seguida a tarefa responsável pela concatenação do atributo primeiro_nome com o atributo ultimo_nome e termina com o evento final “Dados tratados”.

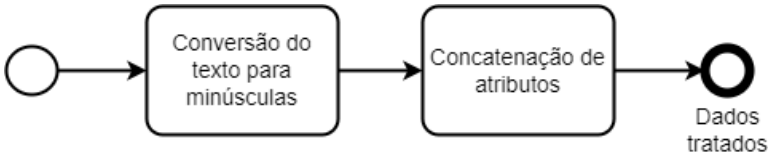

Figura 34 - Subprocesso de nível-filho "Tratar e converter dados"

Passo 5 – repetição da etapa 4

A arquitetura dos processos deve considerar níveis hierárquicos sempre que possível, através da implementação de subprocessos. Os subprocessos permitem manter o controlo do fluxo evitando que o processo se torne complexo. Os detalhes de uma tarefa mais complexa são muitas vezes tarefas específicas que não são necessárias ao conhecimento de todos os intervenientes do negócio, sendo mais direcionadas a utilizadores técnicos e ao serem documentados no subprocesso aumentam significativamente a compressibilidade do modelo.

O passo número 5 do método consiste em repetir a etapa 4 com níveis aninhados adicionais, se existirem. Este passo deve ser realizado para as atividades do tipo subprocesso que ainda careçam de ser expandidas em subníveis. A expansão dos subprocessos torna-se útil para criar diferentes níveis de abstração e simplificar a representação dos processos que constituem o modelo ETL seguindo a abordagem de modelação *top-down.*

Resumo dos passos do método:

1. Analisar o escopo do processo, quando inicia e termina, o que o processo representa e os possíveis estados finais.
2. Enumerar as principais atividades num mapa de alto-nível. Definir os possíveis estados finais de cada atividade.
3. Criar um diagrama BPMN de nível-superior através da organização das atividades identificadas no mapa de alto-nível. Este diagrama de processo deve possuir um evento final de nível-superior que indique o estado final do processo. E devem ser utilizados g*ateways* para mostrar os caminhos condicionais e simultâneos necessários.
4. Expandir cada subprocesso de nível superior num diagrama de nível-filho.
5. Repetir a etapa 4 com níveis aninhados adicionais, se existirem.

5.1.3 Regras de estilo

O Método apresentado no ponto anterior ajuda a estabelecer consistência na estrutura dos diagramas ETL modelados através de BPMN, mas por si só não garante que os diagramas revelem a lógica do processo clara e completa sem a necessidade de documentação complementar. Para maximizar a compreensão dos diagramas é requerida a aplicação de convenções adicionais chamadas estilo de *design*. Esse estilo consiste nas melhores práticas recomendadas, já que são convenções não exigidas pela especificação BPMN 2.0. As melhores práticas de modelação ETL com BPMN consistem na aplicação das regras de estilo ao conjunto de elementos do Nível 1 que tornam a lógica do processo clara. No entanto, os diagramas também devem obedecer às regras oficiais da especificação BPMN, pois um bom modelo começa por aderir às regras da especificação.

Algumas regras de estilo são princípios básicos, enquanto outras são regras específicas de utilização. Segundo [52], as regras de estilo importantes aplicáveis à modelação de Nível 1 são:

1. Utilização de ícones e etiquetas para simplificar a lógica do processo no diagrama. Deve-se por isso documentar todas as atividades, os subprocessos, os estados finais, os fluxos de sequência, os *gateways* e os fluxos de mensagens. Devem ainda ser identificados os tipos de tarefas e as condições que despoletam os eventos. Deve-se ainda utilizar anotações de texto para clarificar a lógica de processo, o que no caso do ETL se revela particularmente útil dada as especificadas e a tecnicidade do processo.
2. Elaboração de modelos hierárquicos. Esse princípio consiste em utilizar uma metodologia que resulte numa decomposição hierárquica. O diagrama de nível superior deve abranger todo o processo. Cada subprocesso num nível de processo deve ser expandido num diagrama de nível filho separado.
3. Utilizar um pool de caixa preta para representar um solicitante externo ao processo do qual não se conhece a lógica. Este tipo de representação pode ser especialmente útil que o designer de ETL pretender omitir certos aspetos, considerandos como intervenientes externos que completam a lógica de processo. Por exemplo, a pool de caixa preta poderá ser utilizada para representar um componente de monitorização ou mesmo de tratamento de erros.
4. Iniciar processos por solicitação através de eventos de início de mensagem. No contexto de um processo de ETL, a inclusão de vários intervenientes não será, à partida, uma prática comum uma vez que que é um processo mais técnico que não requer intervenção manual. No entanto, pode ser utilizada para representar um cenário em que (por exemplo) um componente de tratamento de erros desencadeia um processo de tratamento de dados para lidar com os dados armazenados na área de quarentena.
5. Se for possível devem modelar-se as atividades ETL que estão diretamente ligadas, dentro de uma única *pool* e não em *pools* separadas. As *pools* separadas representam processos diferentes o que no contexto ETL poderá não ser o mais adequado. Representar cada tarefa ETL como um conjunto separado de processos figura 37 geralmente é incorreto, pois transmite a ideia de que cada processo é uma unidade independente dos outros, e não parte do processo. A representação correta está ilustrada na figura 33 em que as três tarefas ETL são definidas dentro de uma única *pool*.

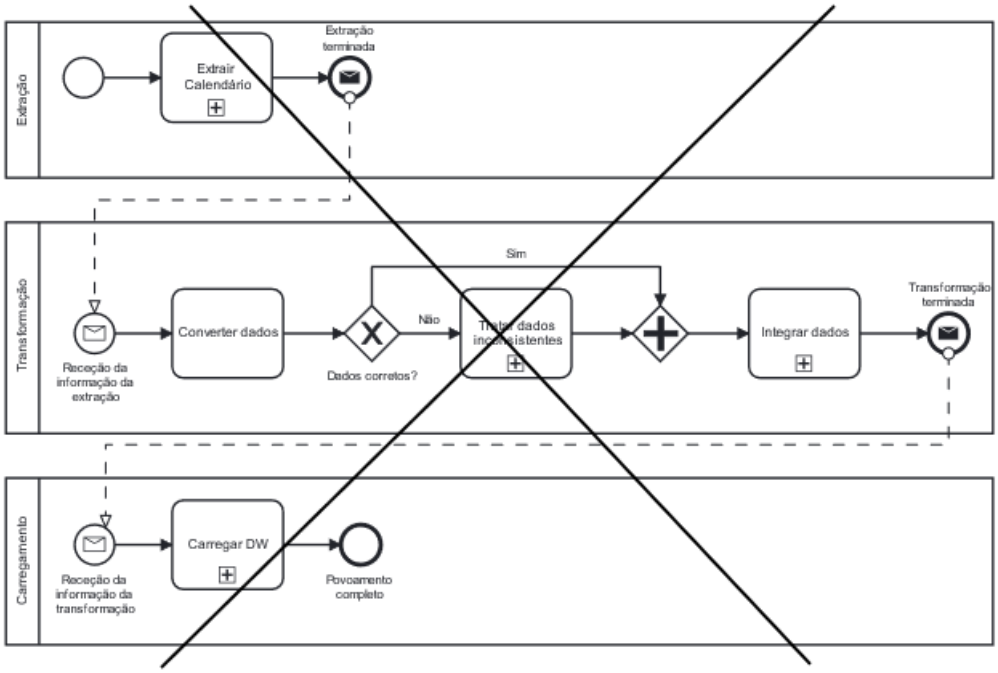

Figura 35 - Representação incorreta da regra número 5

6. Documentar *pools* de processos com o nome do processo e documentar *pools* de caixa preta com a função que presentam no processo. Devem ser utilizados termos genéricos e garantir que não se encontram repetidos.
7. Indicar os estados finais de sucesso e exceção de um processo ou subprocesso, com eventos finais separados e documentar cada um com a indicação do seu estado final. Se o estado final tiver influência no fluxo subsequente, é importante apresentar os estados finais relevantes. Por exemplo, a falha no processo de povoamento de uma dimensão impede o carregamento dos factos na respetiva tabela.
8. Nomes das atividades. Os nomes das atividades devem ser escritos utilizando o infinitivo como forma verbal. Por exemplo “Extrair dados”, “Combinar caracteres” ou “Povoar dimensão”.
9. Definição do tipo de evento de início para indicar como o processo se inicia. No caso dos processos de ETL um evento temporizado indica que o processo irá ser executado periodicamente em intervalos bem definidos. Um evento de início sem tipo pode ser utilizado quando o processo é iniciado manualmente.
10. Se um subprocesso for seguido por um *gateway* rotulado como uma pergunta, o subprocesso deve ter múltiplos eventos finais, e um deles deve corresponder à

etiqueta do *gateway*. Esta regra é importante nos modelos mais complexos, ajudando ao enquadramento da lógica do processo através da hierarquia do diagrama.

11. Mostrar fluxo de mensagens de todos os eventos de Mensagem. Os fluxos de mensagens são opcionais em BPMN, mas numa representação de atividades de ETL é preferível apresentar o fluxo de mensagens ligado a todos os eventos de Mensagem.
12. Fluxos de mensagens em níveis hierárquicos. A segunda regra de rastreabilidade *top-down* requer a aplicação no diagrama de nível-filho de todos os fluxos de mensagens ligadas a um subprocesso colapsado.
13. No caso de existirem vários intervenientes no processo, indicar o nome do fluxo de mensagem através do nome da mensagem para clarificar a comunicação existente entre cada um.
14. Não devem existir eventos finais com o mesmo nome. Caso representem o mesmo estado final, devem ser combinados num único evento final.
15. Não devem existir atividades com o mesmo nome. Se representarem a mesma atividade, deve ser utilizada uma *call activity*. Se representarem atividades diferentes, deve atribuir-se nomes diferentes.
16. Um subprocesso deve conter apenas um único evento de início sem tipo. Num processo de nível superior, são utilizados eventos de início múltiplos para representar diferentes ações que despoletam os processos. No contexto de um subprocesso, deve existir apenas um.

5.2 Nível 2 – Modelação Analítica

O Nível 2 expande o conjunto de formas e elementos apresentada no Nível 1 – Modelação Descritiva. O foco principal está na utilização de eventos. Na prática pretende-se descrever como um processo responde a um evento e como o processo gera um evento baseado numa determinada condição.

No Nível 1, cada etapa do processo é acionada pela conclusão da etapa anterior. Quando uma atividade é concluída, o fluxo de sequência dela proveniente dá início à próxima etapa do processo. Existem, no entanto, outros eventos que ocorrem durante a realização de uma atividade que permitem modelar comportamentos adicionais. O BPMN fornece uma linguagem visual para comportamentos que ocorrem enquanto uma atividade está em execução, o que permite terminar a execução de uma atividade e iniciar uma outra como

resposta. Esta característica é bastante importante para os processos de ETL dado que em várias situações a execução de uma tarefa pode determinar um evento de compensação para impedir que a tarefa termine de forma inesperada. Este cenário ocorre, por exemplo, em situações em que a normalização de nomes é realizada recorrendo a um conjunto pré-determinado de valores expetáveis que são tipicamente armazenados numa tabela de dicionário. Quando surge um valor inesperado, o registo ou o conjunto de registos podem ser redirecionados para uma tabela de quarentena para que posteriormente possam ser analisados.

5.2.1 Conjunto de elementos do Nível 2

O conjunto de elementos de Nível 2 adiciona ao processo os eventos intermediários de limite representados graficamente por um círculo com linha dupla colocados na borda de uma atividade. Os três eventos principais são o evento temporizador (*Timer*), o evento de mensagem (*Message*) e o evento de erro (*Error*). De seguida, é apresentada a lista completa dos elementos (figura 36) que são acumulados ao conjunto de elementos de Nível 1, constituindo assim o conjunto de elementos de Nível 2.

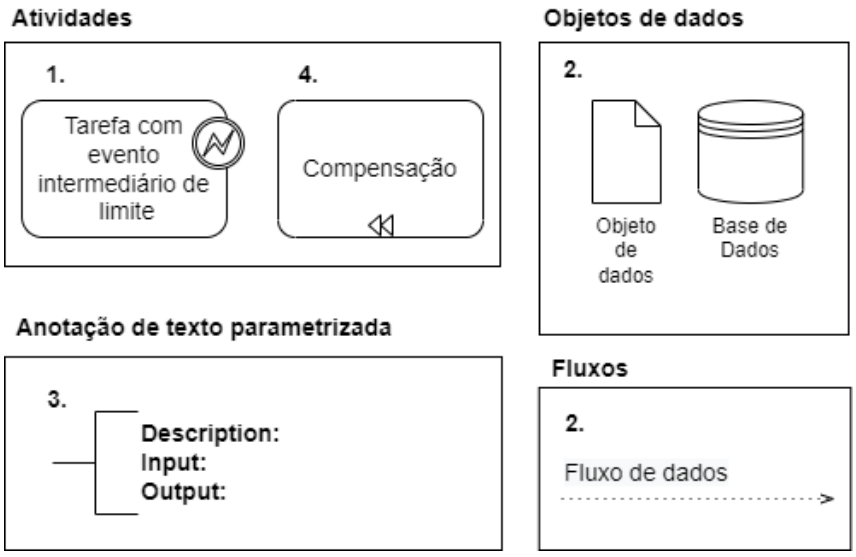

Figura 36 - Conjunto de elementos do Nível 2

A Figura 36 apresenta os seguintes elementos:

1. Tarefa com evento intermediário de limite (*boundary*)

No decorrer de uma atividade os eventos intermediários de limite são especialmente úteis pois permitem modelar comportamentos adicionais como erros ou exceções que ocorram durante uma atividade.

O comportamento despoletado por um evento refere-se a ações do processo que são iniciadas imediatamente após a ocorrência de um sinal específico. Assim como abordado no Nível 1, os eventos criam um fluxo de processo seguido do evento acionado [52]. A instanciação desse novo fluxo ocorre imediatamente após a deteção do comportamento que origina o evento.

Para modelar este tipo de evento num processo, deve ser desenhado um evento intermediário de *catching* no limite da uma atividade pretendida. Este evento de limite não significa espera, significa que enquanto a atividade está em execução, o processo fica à escuta desse sinal. Se o sinal ocorrer antes da conclusão da atividade será acionado o fluxo de sequência do evento, que se denomina de fluxo de exceção. Por outro lado, se a atividade for concluída sem a ocorrência do sinal do evento de limite, o fluxo de exceção é ignorado e o processo continua pelo fluxo normal, ou seja, o fluxo de sequência fora da atividade.

2. Objetos de dados e fluxo de dados

Os “objetos de dados” devem representar os repositórios de dados, que fluem dentro do processo, ou seja, devem ser utilizados para modelar entradas de dados, saídas de dados e o armazenamento de dados. O fluxo sequencial de um processo ETL que existe entre uma atividade e outra, é acompanhado pela transferência de dados, as seja, as atividades utilizam, alteram e criam dados. Se os dados corresponderem a ficheiros ou documentos devem ser modelados em BPMN por um “Objeto de dados”, para modelar objetos referentes a dados persistentes, deve ser utilizado o objeto “Base de Dados”.

Para ligar uma atividade aos objetos de dados deve ser utilizada uma associação direcionada que indique como os dados fluem para a tarefa, indicando se representam o *input* ou o *output* de dados.

3. Anotação de texto parametrizada

Neste nível de modelação, a anotação de texto além da função desempenhada no Nível 1 pode ser parametrizada, nomeadamente na sua utilização em atividades, através da utilização palavras-chave que permitem expressar requisitos específicos, reduzindo as ambiguidades e permitindo a sua melhor interpretação para a posterior implementação a nível físico do processo ETL. As palavras-chave que podem ser utilizadas são: *Description*, *Input* e *Output*, de forma a descrever o objetivo da tarefa, os dados que fluem como input e os dados que representam o resultado de execução da tarefa.

4. Atividade de compensação

O elemento "Compensação" deve ser utilizado para incorporar processos de compensação para desfazer ou compensar ações realizadas por uma atividade ETL. A atividade de compensação utiliza-se para corrigir um erro, consistindo numa atividade adicional que é executada à parte do fluxo de sequência, e após estar terminada o processo segue o seu fluxo de trabalho normal. Deve estar vinculada ao evento de compensação que lhe dá origem através do objeto de ligação associação.

5.2.2 Método

O método a aplicar no contexto ETL consiste nos seguintes passos:

Passo 1 – Adição de objetos de dados

Os “objetos de dados” devem representar os estados dos dados, que fluem dentro de um processo, ou seja, devem ser utilizados para modelar entradas de dados, saídas de dados e o armazenamento de dados.

De forma a ilustrar o primeiro passo, foi considerado o processo “Povoamento da Dimensão Clientes” representado na figura 37. Ao processo anteriormente definido foram acrescentados os objetos de dados necessários à execução do processo ETL. O processo apresentado na figura 37 inicia-se pela extração dos dados do ficheiro “clientes.csv” para a base de dados *Data Staging Area* (DSA). Depois segue-se a limpeza e conversão dos dados e a geração de SK, e por último os dados são armazenados na base de dados “Dim Clientes”. As bases de dados e ficheiros necessários estão representados na figura através dos objetos de dados do BPMN.

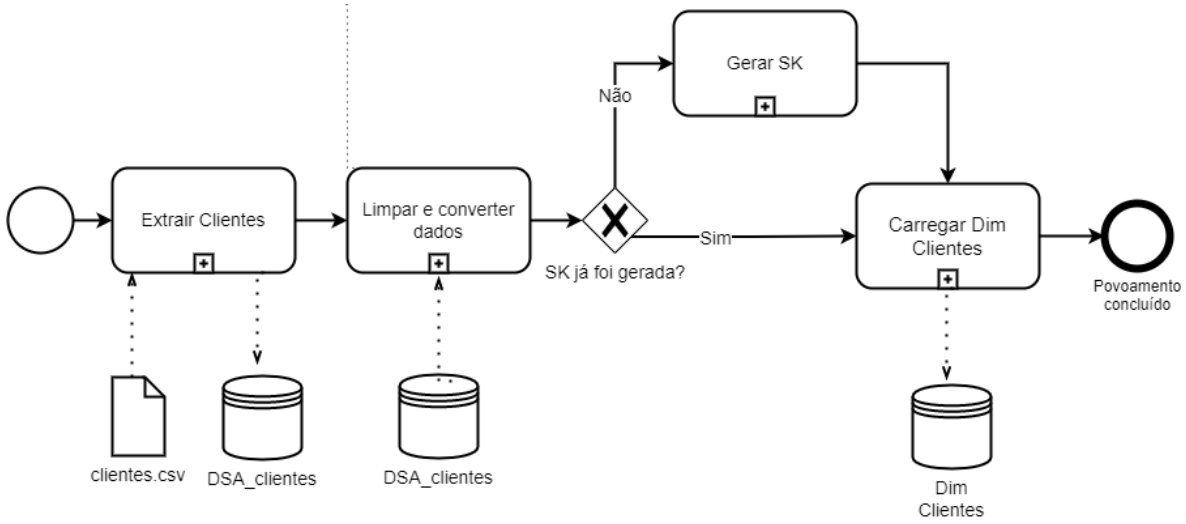

Figura 37 - Adição de objetos de dados

Passo 2 – Adição de anotações parametrizadas

No Nível 1 as anotações de texto desempenham funções como a descrição de condições de *gateways* e a rotulação de elementos como por exemplo o evento de fim. Neste nível as anotações desempenham funções que vão além das descritas para o nível 1, pois possuem uma função parametrizada, nomeadamente na sua utilização, através da utilização palavras-chave em atividades. Desta forma a sua utilização permite expressar requisitos específicos, reduzindo as ambiguidades e permite uma melhor interpretação para a posterior implementação a nível físico do processo ETL.

Para ilustrar o segundo passo do método é seguido o mesmo exemplo para o “Povoamento da Dimensão Clientes” ao qual são adicionados os elementos “anotação” às atividades ETL. Na figura 38 é demonstrada a aplicação do passo em questão onde é possível verificar as palavras-chave que devem ser utilizadas: *description*, *input* e *output*. O elemento *description,* é utilizado para efetuar uma descrição acerca da função para a qual a atividade está desenvolvida, o elemento *input para* descrever as fontes de dados e o elemento *output* para mencionar os dados de destino.

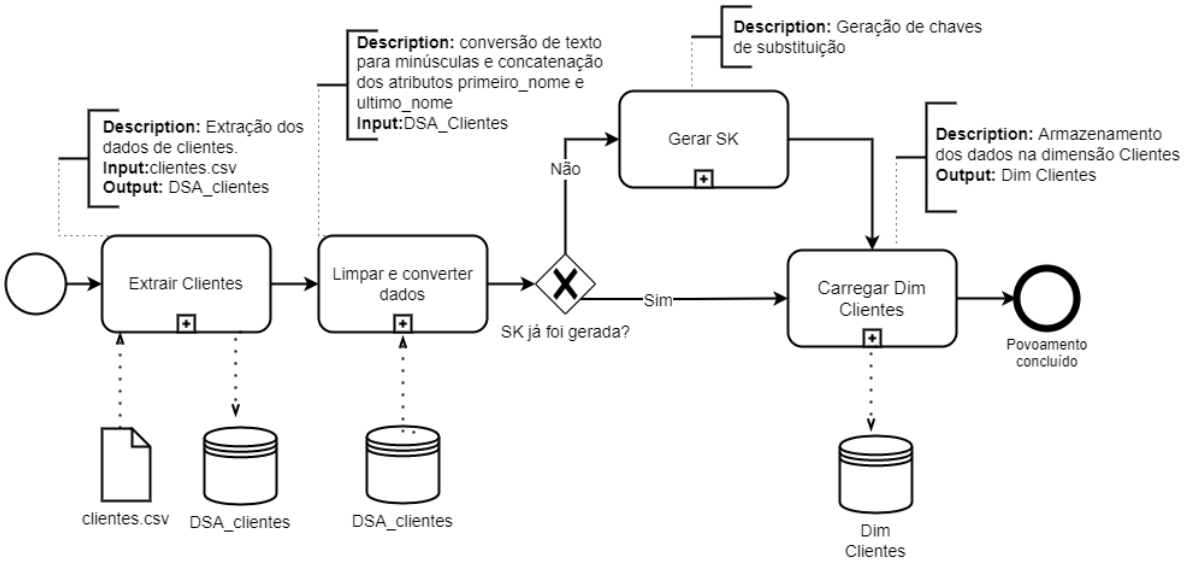

Figura 38 - Adição dos elementos description, input e output

Passo 3 – Adição de eventos intermediários de limite

As tarefas ETL em execução por vezes geram eventos, esses eventos são denominados eventos intermediários de limite. Esses eventos geram comportamentos adicionais como erros ou exceções que ocorram durante a atividade. Para modelar esses comportamentos através da notação BPMN é adicionado um evento intermediário de limite à borda da atividade pretendida.

De forma a ilustrar o terceiro passo do método a atividade “Limpar e converter dados” contém um evento de erro que ilustra de que forma os erros são tratados no decorrer do processo. Se durante a atividade ocorrer algum erro, este evento gera um fluxo de sequência alternativo denominado fluxo de exceção que liga à atividade “Tratamento de erros”. Por outro lado, se a atividade for concluída sem a ocorrência do sinal do evento de limite de erro, o fluxo de exceção é ignorado e o processo continua pelo fluxo normal (ver figura 39).

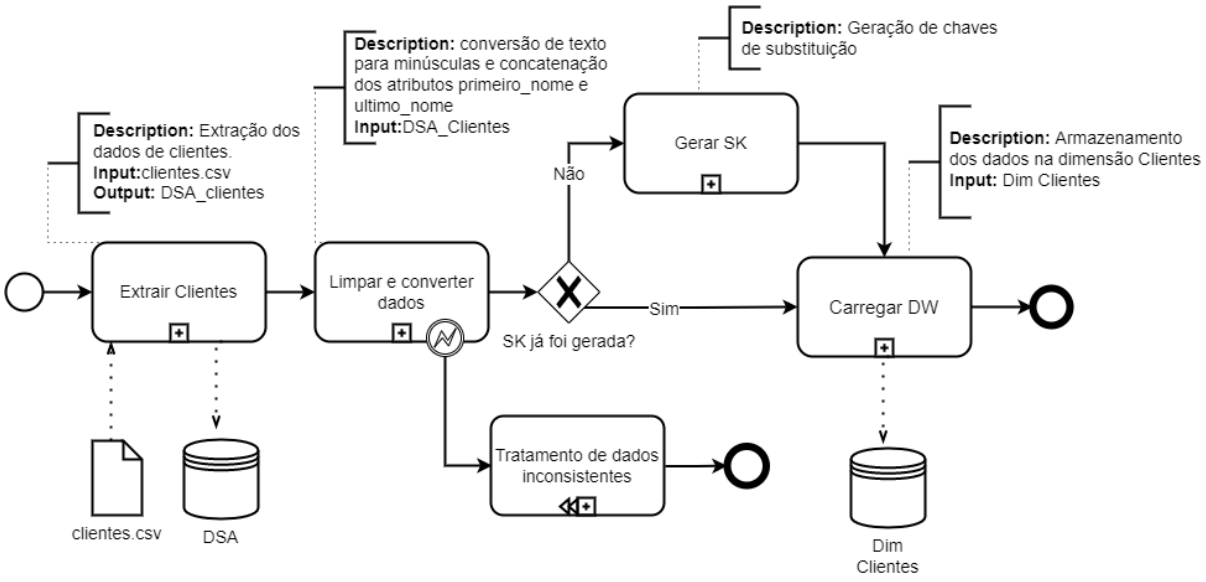

Figura 39 - Adição do evento intermediário de limite

5.2.3 Regras de estilo

As regras de estilo mais importantes aplicáveis à modelação processos de Nível 2 enquadrados com as caraterísticas do ETL são [52]:

Marcação

1. O nome da atividade deverá ser preferência um verbo no modo impessoal.
2. Duas atividades no mesmo processo não devem ter o mesmo nome, a menos que ambas sejam atividades de chamada.
3. O evento de início deve ter um nome que indique a condição do acionamento.
    - Um evento de início de mensagem deve ter um nome como "Receber [nome da mensagem]".
    - Um evento de início do temporal deve ter um nome que indique a programação do processo.
    - Um evento de início de sinal deve ter um nome que indique o nome do sinal.
    - Um evento de início condicional deve ter um nome que indique a condição de o despoleta.
4. No caso dos eventos de limite, o nome de um evento de limite de erro num subprocesso deve corresponder ao nome de um evento de fim de erro no nível filho.
5. Um evento intermediário de *throwing* ou *catching* deve ter um nome.

6. Um evento final deve ter um nome correspondente ao seu estado final.
7. O nome do diagrama de nível filho deve corresponder ao nome do subprocesso.

Eventos de fim

8. Dois eventos finais no mesmo processo devem conter nomes diferentes. Se esses eventos significarem o mesmo estado final devem ser combinados.
9. Se um subprocesso for seguido por um *gateway* “sim/não”, o evento final do subprocesso deve ter o nome de forma corresponder ao rótulo do *gateway*.

Expansão do subprocesso

10. Em cada subprocesso apenas pode ser utilizado um evento de início.
11. Uma expansão de nível filho não deve ser incluída numa forma de subprocesso expandida se os níveis de processo pai e filho forem representados em diagramas separados.

Nas abordagens de modelação Descritiva e Analítica, o foco passa por representar visualmente as especificidades do processo, descrevendo a lógica do processo de uma forma humanamente compreensível. A ênfase está no diagrama e na representação visual da lógica do processo. A serialização XML serve principalmente o propósito de troca de modelos entre ferramentas. Num processo executável, um mecanismo de *software* automatiza o fluxo de execução do modelo desde a instanciação do processo até a conclusão. Isso requer que sejam especificados detalhes adicionais para cada elemento BPMN, como os dados de entrada e de saída, a definição de eventos e a elaboração de expressões condicionais [52]. Esses detalhes são invisíveis no diagrama, mas o BPMN 2.0 fornece elementos XML para os especificar. Assim, no contexto deste trabalho, o termo “BPMN executável” refere-se à capacidade que uma ferramenta possui para especificar e exportar detalhes relacionados à execução.

O método e estilo BPMN trata de expor a lógica do processo claramente no diagrama através da utilização de formas e rótulos, enquanto o BPMN executável trata da definição e mapeamento dos dados do processo. Alinhar o *design* executável com método e estilo implica uma ligação específica entre as formas e regras no diagrama e as variáveis, mensagens, entradas de dados, saídas de dados e mapeamentos no modelo executável. A modelação executável não se torna particularmente útil no contexto ETL visto que um modelo exportado

no formato XML contém apenas a descrição do que o processo deve fazer, ou seja, a modelação conceptual. Não possui as características especificas, como as ligações à base de dados e operações especificas, para que possa ser interpretado por uma ferramenta como por exemplo o *Microsoft Integration Services*. Para tal seria necessário enriquecer essa informação com diversos dados que formariam um esqueleto passível de importação através de uma ferramenta ETL. Vários trabalhos têm surgido nesse âmbito com o intuito de mapear os modelos conceptuais para modelos físicos: [53], [15], [54].

Através de uma ferramenta de modelação BPMN é possível que qualquer utilizador, possa representar um processo ETL através de BPMN com relativa rapidez, mesmo não estando familiarizado com a notação. O resultado pode ser na maior parte dos casos um modelo abstrato, inconsistente e que não otimizaria a implementação física do processo ETL. A elaboração de modelos consistentes e otimizados tornam o conhecimento do processo transparente. Para fornecer orientação aos *designers* de ETL durante a criação de diagramas BPMN e orientação aos leitores no esforço que fazem para entender esses diagramas, a definição e aplicação de regras de modelação específicas são muito úteis. Não apenas por uma questão de semântica, mas também porque a aplicação de regras consistentes deve resultar na modelação de um processo consistente e de alta qualidade, bem como eliminar interpretações erradas que de outra forma seriam difíceis de evitar. A proposta de modelação apresentada preenche essa lacuna e garante que a modelação de processos ETL seja realizada de forma padronizada, consistente e normalizada.

Capítulo 6 – Caso de Estudo

O caso de estudo apresentado neste capítulo demonstra a aplicação da BPMN à modelação de um exemplo conhecido da Microsoft o Wide World Importers[23] (WWI). Visa demonstrar como a abordagem de desenvolvimento de ETL com BPMN é aplicada, passo a passo.

A WWI é uma importadora e distribuidora de produtos inovadores que vende para clientes de retalho nos Estados Unidos, incluindo lojas especializadas, supermercados, lojas de informática, lojas de atrações turísticas e clientes individuais. O esquema dimensional do DW é composto por oito dimensões e seis tabelas de factos que compõem vários "módulos" referentes a eventos gerados por processos de negócios específicos.

O processo de povoamento inicia-se com a dimensão temporal (data). Este processamento garante que todas as datas do ano atual são preenchidas na tabela. De seguida, cada uma das tabelas de dimensão é povoada. E por último são carregadas as tabelas de factos.

Para este caso de estudo foram selecionados os processos ETL relacionados com o povoamento do esquema de vendas. Esse esquema integra uma tabela de factos "Venda" que representa as vendas faturadas a clientes, e as dimensões "Data", "Cidade", "Funcionário", "Cliente" e "Item de stock" que suportam o processo de negócio.

6.1 Estrutura Dimensional do *Data Warehouse*

Um DW deve ser estruturado de forma a suportar os principais processos de tomada de decisão, para tal, as tabelas e os seus relacionamentos devem ser modelados orientados ao assunto e considerando as necessidades de análise para um determinado processo. Segundo os fundamentos da modelação dimensional [55], as tabelas de factos representam os processos de negócio e as suas medidas, enquanto as tabelas de dimensões representam as perspetivas de análise acerca desses processos.

[23] https://docs.microsoft.com/en-us/sql/samples/wide-world-importers-what-is?view=sql-server-ver15

O esquema do DW de vendas referente ao caso de estudo consiste num esquema em estrela e está representado na figura 40.

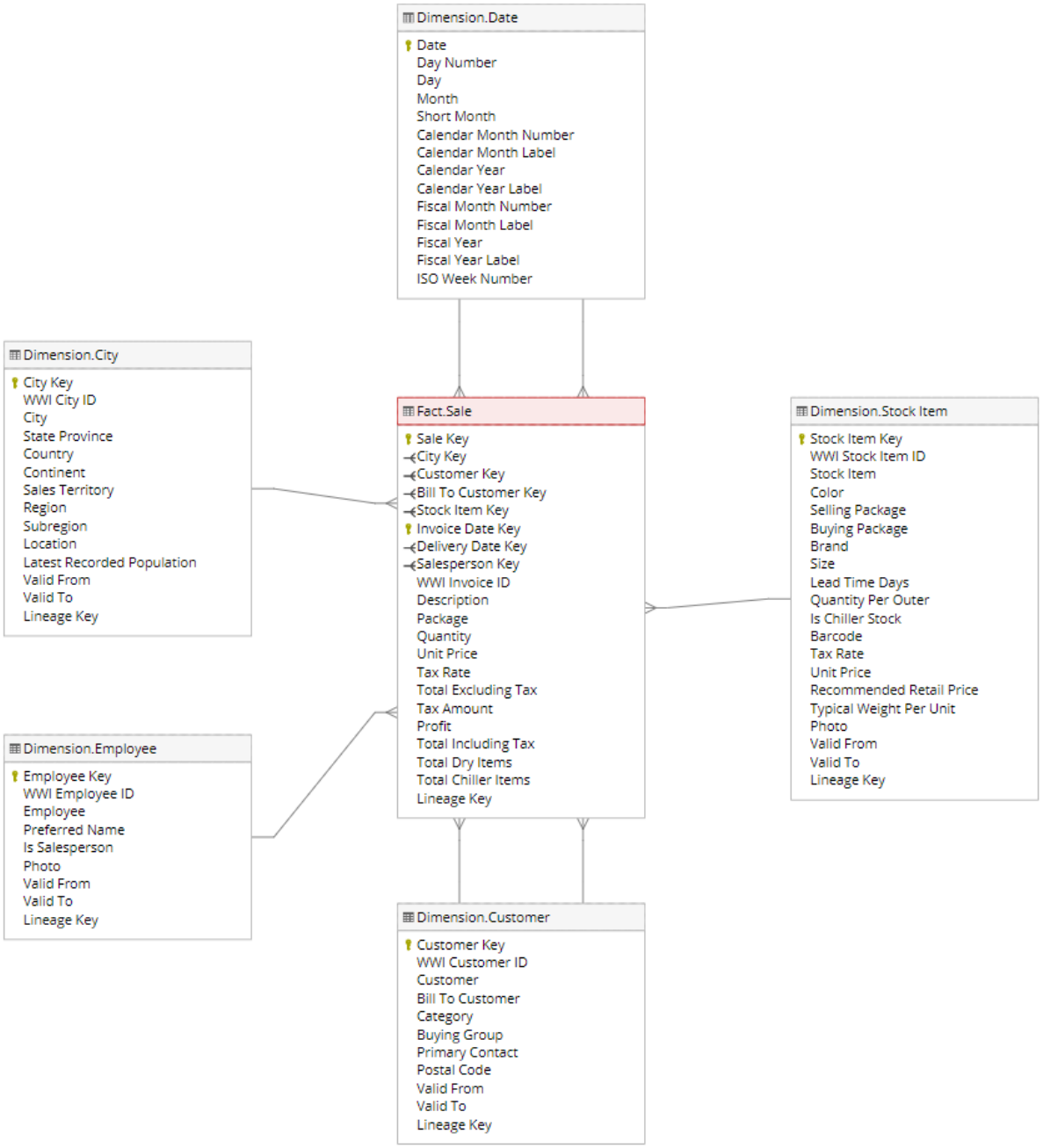

Figura 40 - Esquema dimensional do DW Vendas

O DW referido na figura 40 é constituído por uma fabela de factos “Vendas”. Cada entrada armazenada nessa tabela de factos representa uma venda de um item em *stock*, a um cliente, por um funcionário numa certa cidade numa determinada data. As dimensões presentes neste caso de estudo que integram o DW Vendas são:

- **Date** (Data) – dimensão temporal com o grão relativo ao dia que tendo dois papeis (*Role-Playing dimension*) no relacionamento com a tabela de factos, relativamente à data da venda e a data de entrega de um item.
- **Stock Item** (Artigo) – dimensão que representa a informação relativa a um item em *stock*/artigo.
- **Customer** (Cliente) – dimensão que representa a informação sobre clientes envolvidos em vendas, tendo dois papeis (*Role-Playing dimension*) no relacionamento com a tabela de factos, relativamente ao cliente que envolvido na venda e ainda o cliente para o qual foi emitida a fatura.
- **Employee** (Funcionário) – dimensão representa a informação relativa a funcionários.
- **City** (Cidade) – dimensão que representa a informação sobre dados geográficos.
- **WWI Invoice ID, description** e **package –** dimensões degeneradas que representam o código da fatura, descrição da venda e ainda o tipo de pacote utilizado na venda.

Para cada facto são definidas um conjunto de medidas. As medidas utilizadas no esquema Vendas são:

- **Unit Price** (Preço unitário) – preço unitário cobrado pelo artigo.
- **Tax Rate** (taxa do imposto) – valor do imposto aplicado ao artigo
- **Quantity** (Quantidade) – quantidade vendida.
- **Total Excluding Tax** (Valor excluindo o imposto) – Valor total da venda excluindo o imposto.
- **Tax Amount** (Valor do imposto) – Valor total do imposto.
- **Profit** (Lucro) – Valor total do lucro.
- **Total Dry Items** (Total de itens secos) – Número total de artigos secos.
- **Total Chiller Items** (Total de itens refrigerados) – Número total de artigos refrigerados.

6.2 Modelação conceptual do processo ETL com BPMN – Nível 1

Com base na abordagem seguida no capítulo 5, é apresentada uma proposta de modelação de processos de ETL abordando os dois primeiros níveis (Nível 1- Modelação Descritiva e Nível 2 – Modelação Analítica) de modelação através da utilização da notação BPMN. Como referido no capítulo 5, o método de modelação consiste na elaboração de um modelo completo, consistente e bem estruturado baseado num estilo de modelação hierárquica.

A modelação de Nível 1, a Modelação Descritiva, consiste numa representação inicial do processo ETL através da demonstração das suas principais atividades e tem como objetivo a documentação simples do fluxo do processo.

Através do conjunto de elementos de Nível 1 e da aplicação do método apresentado para esse nível, é proposta uma abordagem de modelação *top-down*, ou seja, a elaboração de um mapa de alto-nível de forma a conhecer todo o processo. Através do mapa de alto-nível é criado um diagrama BPMN de nível-superior. Posteriormente esses diagramas são expandidos para os diagramas de nível-filho e por último são adicionados os fluxos de mensagem.

6.1.1 Âmbito do processo

O primeiro passo para a modelação do processo ETL consiste na definição do âmbito do processo. Neste primeiro passo, como referido anteriormente, não é elaborado diagrama. O processo em questão consiste no povoamento do esquema Vendas (Figura 40). Para a definição do escopo do processo foram considerados:

1. Inicio do processo: O processo tem início programado regularmente para execução diária, às 23:00 horas.
2. Fim do processo: O processo conclui-se após o carregamento dos dados na tabela de factos de venda - **Sale**.
3. O que cada instância do processo representa: Cada instância representa um processo de povoamento do esquema Vendas.
4. Nomeação das diferentes formas de terminar o processo: Para este processo apenas existe um evento final pré-determinado que se segue após a conclusão da última tarefa do processo.

6.1.2 Mapa de alto-nível

O segundo passo do método consiste na definição do mapa de alto nível que consiste na elaboração de uma lista das principais atividades do processo de povoamento do esquema de Vendas.

Atividades identificadas:

- **Carregamento Dimensão Data.** Processo de geração de registos de datas considerando o período 2010-2022.

Estado final: Geração de dados concluída.

- **Carregamento Dimensão Cidade:** Processo de povoamento da dimensão cidade. Esta atividade começa logo que a atividade carregamento da dimensão data for concluída (de acordo com o processo definido para o exemplo WWI). Consiste numa tarefa responsável por extrair os dados da fonte operacional, tratar os mesmos e efetuar a carga na dimensão de destino (*City*).

Estado final: carregamento concluído.

- **Carregamento Dimensão Cliente:** Processo de povoamento da dimensão cliente. Esta atividade começa logo que a atividade carregamento da dimensão cidade for concluída. Esta tarefa consiste no povoamento do Dimensão *Customer* através de uma série de tarefas de fluxo de dados, a partir dos dados de clientes oriundos do ambiente operacional.

Estado final: carregamento concluído.

- **Carregamento Dimensão Funcionário:** Processo de povoamento da dimensão funcionário. Esta atividade começa logo que a atividade carregamento da dimensão cliente for concluída. Diz respeito à extração, transformação e carregamento da Dimensão *Employee* a partir dos dados obtidos da fonte operacional.

Estado final: carregamento concluído.

- **Carregamento Dimensão Itens de stock:** Processo de povoamento da dimensão itens de stock. Esta atividade começa logo que a atividade carregamento da dimensão

funcionário for concluída. Consiste no povoamento da dimensão *Stock Item* pela extração dos dados da fonte operacional, seguindo-se a sua transformação e o por último o respetivo carregamento.

Estado final: carregamento concluído.

- **Carregamento Facto Venda:** Processo de povoamento da tabela de factos venda. A tabela venda (*Sale*) consiste nas vendas faturadas a clientes. Esta atividade começa logo que a atividade carregamento da dimensão itens de stock for concluída. Começa por extrair os dados da tabela de origem, depois obtém as chaves estrangeiras para os registos das dimensões “DimDate”, “DimCity”, “DimCustomer”, “DimEmployee” e “DimStockItem”. Posteriormente é executado o processo de transformação e por último são inseridos os registos transformados na tabela de facto *Sale*.

Estado final: carregamento concluído.

6.1.3 Diagrama de processo de nível-superior

Através do mapa de alto-nível elaborado no passo anterior, este é transformado num diagrama BPMN de nível-superior. A elaboração do diagrama de nível superior para o processo de povoamento do esquema de Vendas é realizada em duas fases. O diagrama da Fase 1 está apresentado na figura 41, onde cada atividade é representada por um subprocesso e modelada sequencialmente. Ou seja, o processo é modelado conforme a sequência de preenchimento das tabelas do esquema definido pelo exemplo WWI.

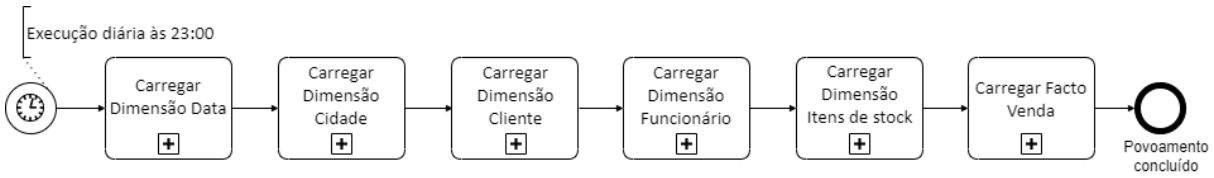

Figura 41 - Representação do processo ETL de nível superior - Fase 1

A Fase 2 consiste na elaboração do diagrama apresentado na figura 42 onde as atividades ETL que correspondem ao povoamento das dimensões são executadas paralelamente, pois não têm dependência entre si, e posteriormente é executada a atividade que corresponde ao povoamento do facto **vendas**.

De acordo com a abordagem seguida e considerando as características do processo apresentado na figura 42, existem atividades, nomeadamente, o povoamento das dimensões, em que é possível dividir o fluxo de povoamento em caminhos paralelos usando um *gateway* paralelo. Neste caso, o povoamento da tabela de factos depende da conclusão do povoamento das dimensões, já que existe uma dependência de integridade referencial. Por isso, é utilizado um *gateway* paralelo convergente para indicar que só após a conclusão dos fluxos de povoamento das dimensões, o fluxo de povoamento da tabela de factos pode iniciar.

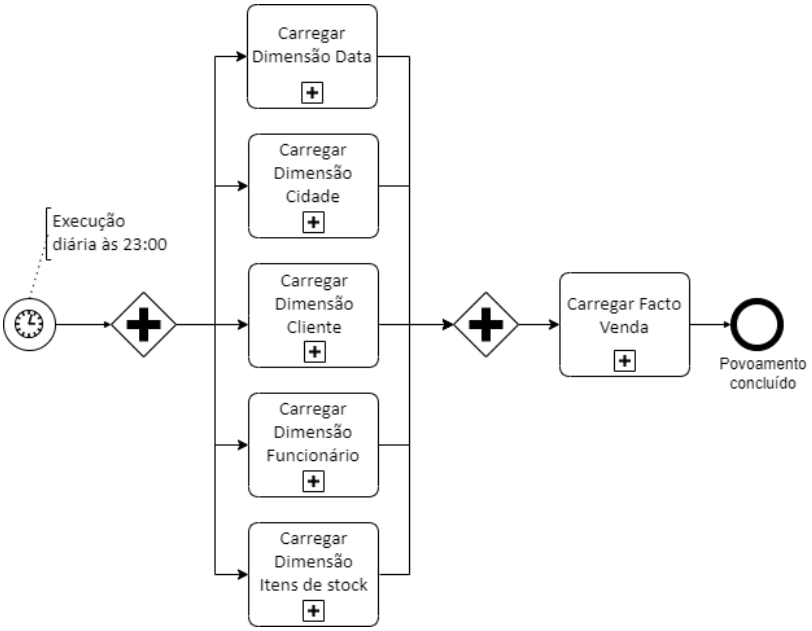

Figura 42 - Representação do processo ETL de nível superior - fase 2

6.1.4 Expansão *child-level* (nível-filho)

O diagrama de nível-superior apresentado na figura 41 e na figura 42 tem como objetivo informar como o processo começa e como termina, encapsulando a complexidade de cada tarefa. Na modelação hierárquica, cada processo *child-level* é desenhado num diagrama separado, associado a uma atividade de subprocesso recolhido no diagrama de nível-superior.

Para o caso de estudo em questão foi selecionado o subprocesso **"Carregar Dimensão Cliente"** para a elaboração do diagrama de nível-filho apresentado na figura 43.

Figura 43 - Diagrama nível-filho "Carregar dimensão cliente"

O processo de carregamento da dimensão cliente inicia-se pela atividade **"Atualizar a Tabela Lineage"** responsável por registar quando o processo de alimentação da tabela iniciou e será atualizado posteriormente após a conclusão da execução da instância do processo. A segunda atividade **"Limpar a tabela Staging"** consiste em num procedimento de limpeza da tabela de *staging* para a preparar para novos dados. A terceira atividade **"GetLastETLCutoffTime"** consiste na execução de um procedimento: que é responsável por identificar o *cutofftime* (data de corte), que consiste na data de início do povoamento para a tabela *customer* (cliente). A quarta atividade **"Extrair dados de clientes"** – consiste num *Data Flow* para extrair os dados sobre clientes, onde os novos dados da fonte são lidos e copiados para a tabela de *staging* do DW de destino. São identificadas:

- As mudanças nos grupos de compra;
- As mudanças na categoria dos clientes que ocorreram desde o último povoamento;
- As mudanças dos clientes que ocorreram desde o último povoamento.
  De seguida é povoada a tabela de *staging* do cliente.

A quinta atividade **"Carregar Dimensão Cliente"** – diz respeito à incorporação dos dados nas tabelas no esquema de destino.

6.1.5 Repetição da etapa 4

Este passo apenas deve ser implementado para as atividades que contenham níveis adicionais aninhados, ou seja, para as atividades do tipo subprocesso que ainda careçam de ser expandidas em subníveis. Os subprocessos BPMN podem ser úteis para criar diferentes níveis de abstração e simplificar a representação do modelo seguindo desta forma a proposta apresentada no capítulo 5, uma abordagem de modelação *top-down*.

No caso de estudo em questão existem dois subprocessos nessa situação: **“Extração dos dados do cliente”** e **“Carregar Dimensão Cliente”**, pelo que são apresentados nas figuras seguintes os respetivos diagramas de nível-filho.

A expansão do subprocesso **“Extrair dados de clientes”** é apresentado na figura 44 e é composto por quatro subprocessos que dizem respeito à identificação das mudanças que ocorreram relativas aos clientes (nos grupos de compra, nas categorias de clientes e nos clientes) e à tarefa de carregamento da tabela “Customer Staging”, com os dados novos e/ou alterados, que foram identificados nos subprocessos anteriores – **“Carregar staging customer”.**

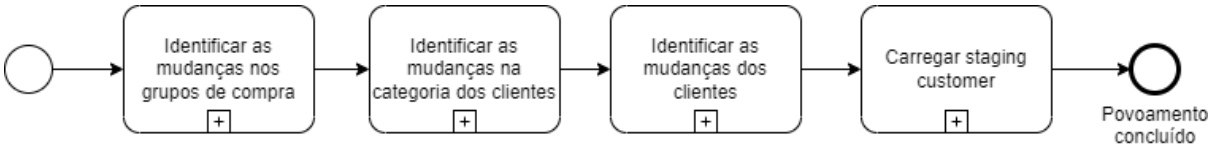

Figura 44 - Subprocesso “Extrair dados de cliente”

A expansão do subprocesso **“Carregar Dimensão Cliente”** representada na figura 45 é composta por duas atividades responsáveis por atualizar as tabelas “Lineage” e “ETLCutoff” e uma atividade responsável por carregar a tabela de dimensão “Customer” através dos dados provenientes da tabela “Customer Staging”.

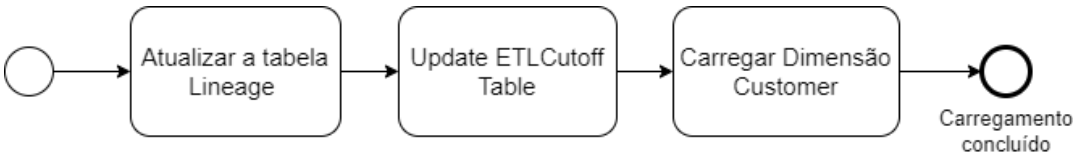

Figura 45 - Subprocesso “Carregar Dimensão Cliente” fase 1

O modelo apresentado na figura 45 referente à atividade **“Carregar Dimensão Cliente”** também pode ser modelado segundo a fase 2, em que as atividades **“Atualizar tabela Lineage”** e **“Update ETLCutoffTable”** podem ser executadas paralelamente e a atividade **“Carregar Dimensão Customer”** é executada posteriormente. Este subprocesso **“Carregar Dimensão Cliente**” está ilustrado na figura 46.

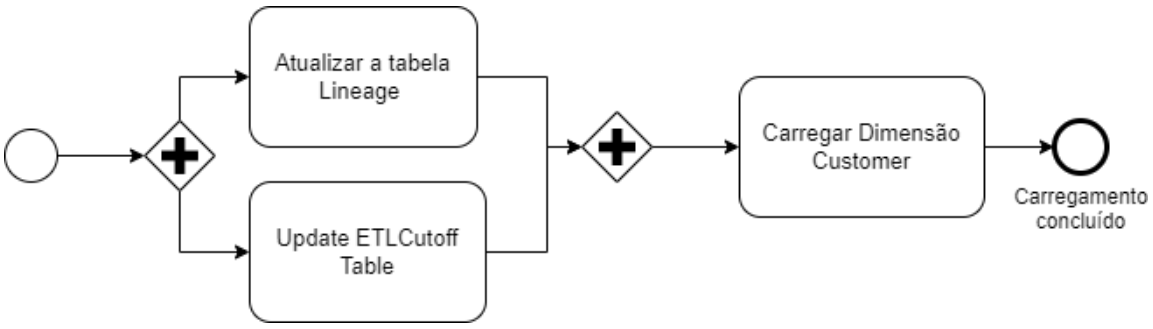

Figura 46 - Subprocesso "Carregar Dimensão Cliente" fase 2

Sempre que uma tarefa seja passível de ser decomposta esta deve ser modelada através de um subprocesso que posteriormente seja expandido, ao contrário de uma atividade atómica, que deve ser modelada com uma tarefa.

6.3 Modelação conceptual do processo ETL com BPMN – Nível 2

A modelação de Nível 2, a Modelação Analítica, consiste numa representação pormenorizada do processo ETL destinada à interpretação por utilizadores avançados.

A cada nível de abstração a complexidade do modelo vai aumentando, e de forma a detalhar os pormenores do fluxo ETL vão sendo adicionados novos elementos, como as anotações e os objetos de dados. Os elementos de anotação devem incluir os dados de entrada na atividade (*input*), uma descrição acerca do funcionamento da tarefa (*description*) os dados de saída da atividade (*output*).

Através do conjunto de elementos de nível 2 e da aplicação do método apresentado para esse nível é seguida a modelação do caso de estudo em questão. Cada processo deve ser documentado ao maior nível de detalhe possível, desta forma facilitará bastante o processo de implementação a nível físico facilitará a interpretação do processo.

6.3.1 Adição de objetos de dados

O primeiro passo do método de nível 2 consiste na adição de objetos de dados aos diagramas já implementados, que devem representar os estados dos dados, que fluem dentro de um processo, sejam entradas de dados, saídas de dados ou armazenamento de dados.

O processo apresentado na figura 47 consiste no subprocesso **"Carregar Dimensão Customer"** apresentado anteriormente ao qual são adicionados os objetos de base de dados correspondentes às fontes e aos destinos dos dados utilizados pelas atividades constituintes do processo.

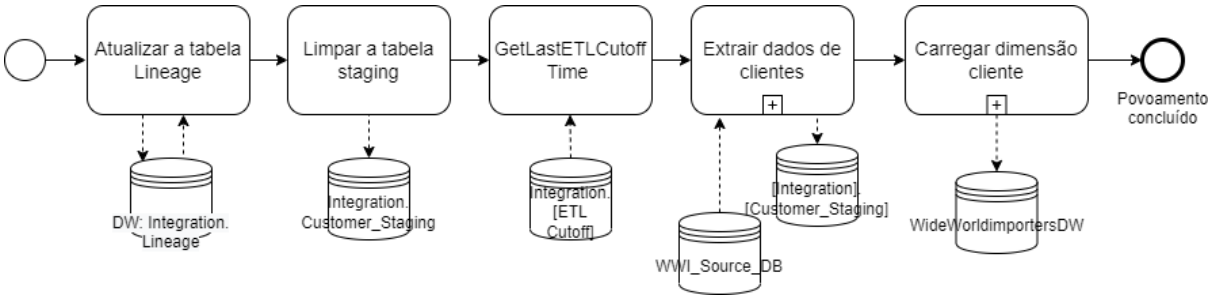

Figura 47 - Carregar Dimensão Customer com a inclusão de objetos de dados

Assim como na figura 47, na figura 48 está representado o subprocesso **"Extração dos dados do cliente"** com adição dos objetos de dados integrantes do processo.

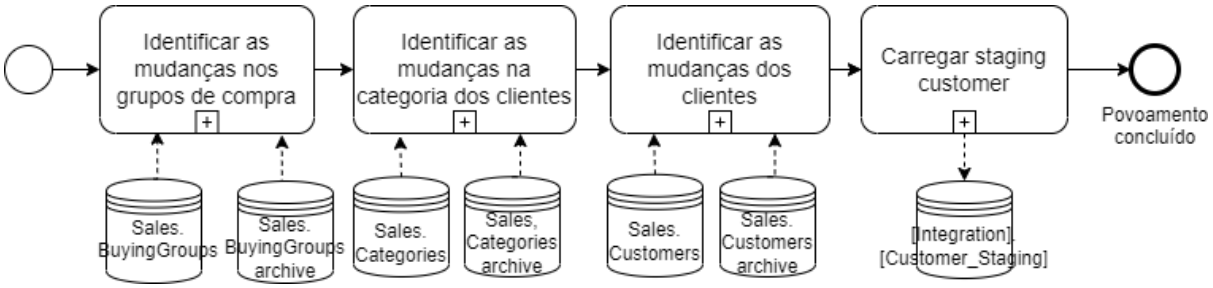

Figura 48 - Extração dos dados do cliente com adição de objetos de dados

Assim como na figura 47, na figura 49 está representado o subprocesso **"Carregar Dimensão Cliente"** com adição dos objetos de dados utilizados pelo processo.

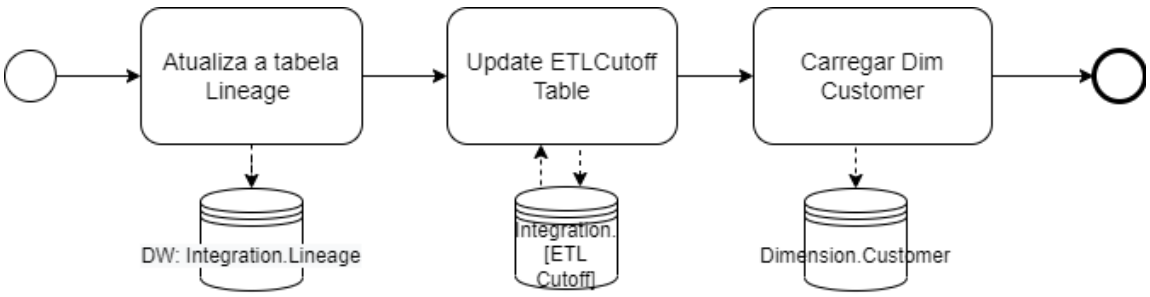

Figura 49 - Carregar Dimensão Cliente com adição de objetos de dados

6.3.2 Adição de anotações parametrizadas

As anotações parametrizadas devem ser utilizadas a partir da expansão *child-level* (nível-filho) e visam adicionar palavras-chave às atividades ETL modeladas de forma a permitir uma melhor interpretação dos processos pelos implementadores de ETL a nível físico. Na figura 50 está representado o subprocesso **"Carregar Dimensão Customer"** ao qual foram adicionadas anotação parametrizadas. As anotações parametrizadas adicionam informação extra às atividades, nomeadamente uma descrição do intuito para o qual cada atividade está criada, os dados de input das atividades e os dados de output.

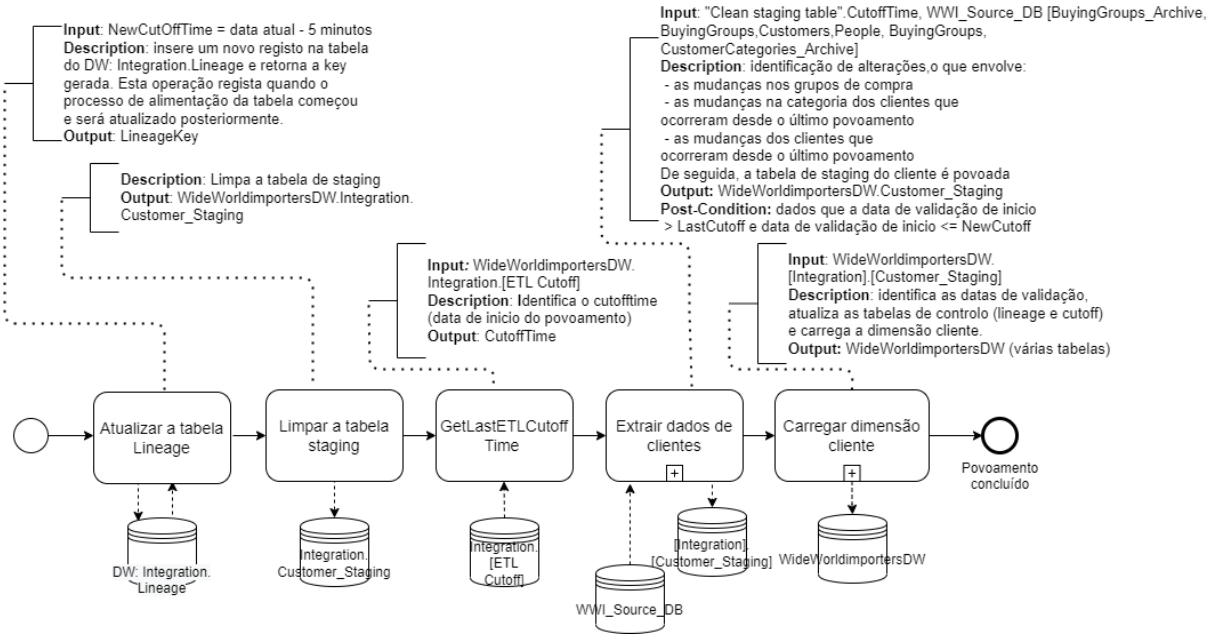

Figura 50 - Carregar Dimensão Customer com anotações parametrizadas

Assim como na figura 50, na figura 51 está representado o subprocesso **"Extração dos dados do cliente"** com anotações parametrizadas.

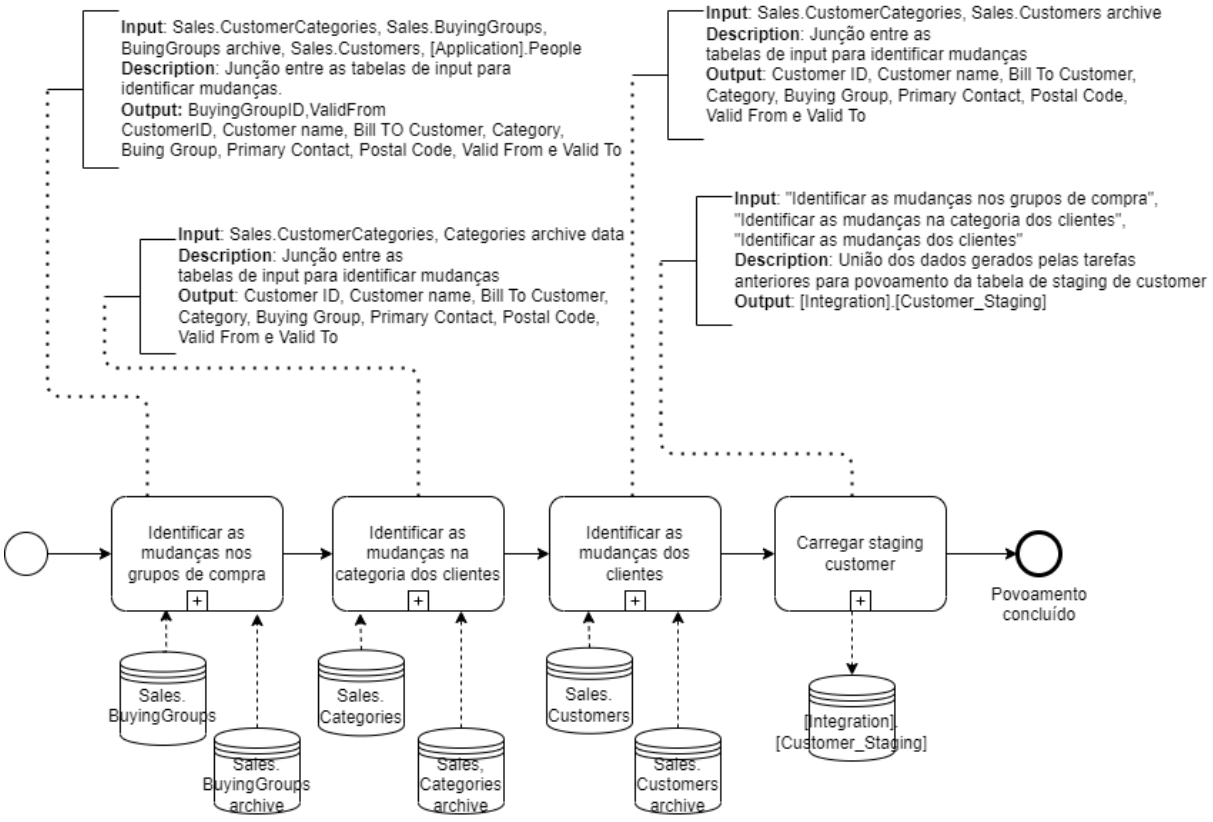

Figura 51 - Extrair dados de clientes com anotações parametrizadas

Assim como descrito na figura 50, na figura 52 está representado o subprocesso **"Carregamento Dimensão Cliente"** com anotações parametrizadas.

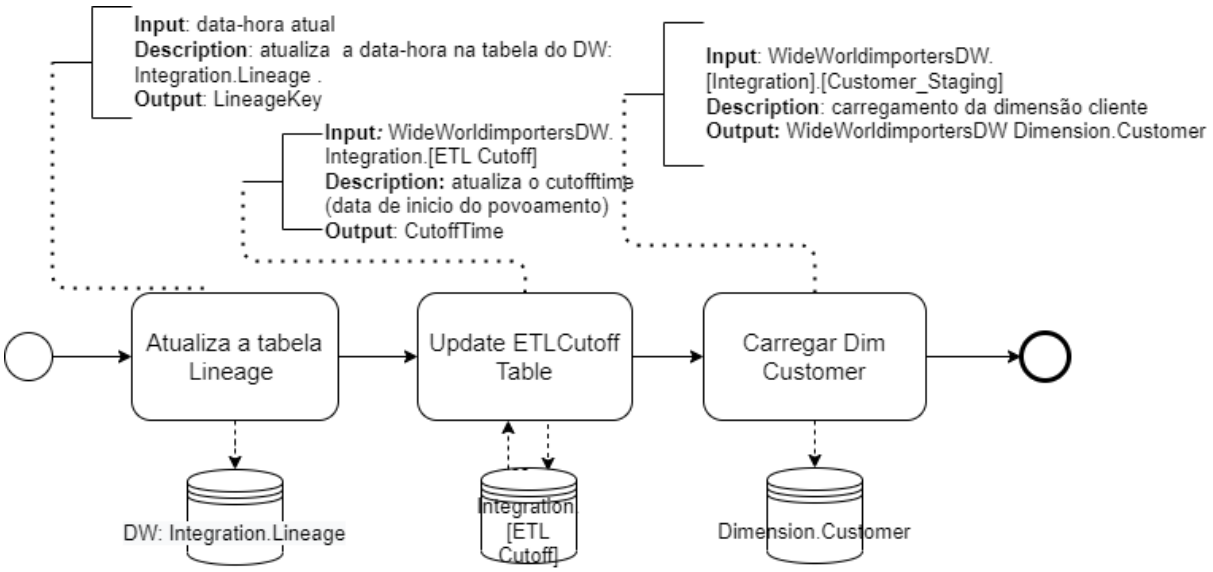

Figura 52 - Carregar Dimensão Cliente com anotações parametrizadas

6.3.3 Adição de eventos intermediários de limite

Neste caso de estudo não foi necessária a utilização de todos os elementos do conjunto de elementos de nível 1 e de nível 2, como os eventos intermediários de limite. Ainda assim é possível referir que os eventos intermediários de limite são especialmente úteis para lidar com exceções e erros em processos ETL.

Como referido anteriormente no capítulo 5, o nível de modelação executável não tem relevância num projeto deste tipo visto que a exportação do modelo de cada processo em XML contém apenas a modelação conceptual e não possui as características específicas como as operações efetuadas pelas atividades e as ligações a base de dados, para ser passível de importação por uma ferramenta de implementação de ETL. Nas ferramentas BPMN tradicionais, a serialização do processo é útil para partilhar especificações conceptuais e também para partilhar pormenores de implementação/execução do processo considerando características base do BPMN que podem ser facilmente interpretáveis por várias ferramentas. No entanto, a execução de processos ETL é uma aplicação que se distingue essencialmente pelas características do domínio. Por exemplo, a intervenção humana (típica em processos de negócio comuns) é quase inexistente neste tipo de processo dados que são processo técnicos executadas diretamente por uma ferramenta. No limite, podemos interpretar o processo ETL com um conjunto de comandos SQL orquestrados num

determinado contexto. Por isso, a execução do processo será algo mais específico, seja utilizando ferramentas “visuais” como o Microsoft Integration Services, ou utilizando uma abordagem *code-first* como por exemplo, o Apache Airflow[24] ou Apache Spark[25]. Por isso, a serialização do processo em XML teria de ser realizada sobre a perspetiva de ETL de forma a aproveitar a especificação conceptual desenvolvida. A maior dificuldade reside essencialmente na especificação das tarefas documentadas a nível conceptual, que num domínio técnico como o do ETL dificulta a geração de mecanismos automatizados para gerar modelos físicos. Em [56] os autores abordaram esta questão, mas utilizam o conceito de padrão para limitar o âmbito das tarefas e assim definir procedimentos que podem ser instanciados de acordo com o âmbito de cada tarefa modelada em BPMN.

[24] https://airflow.apache.org
[25] https://spark.apache.org

Capítulo 7 – Conclusões e Trabalho Futuro

7.1 Conclusões

O ETL é o processo mais complexo e moroso na construção de um sistema DW. Atualmente são disponibilizadas diversas ferramentas de ETL no mercado, cada uma com as suas particularidades. As capacidades destas ferramentas no tratamento e manipulação de informação, aliadas à facilidade e simplicidade de utilização, tornam-nas uma referência no desenvolvimento do processo ETL.

Um fluxo de trabalho ETL compreende as atividades de extração, transformação e carregamento de dados. Essas atividades podem ser representadas e executadas de formas distintas, por exemplo, usando operadores relacionais ou funções definidas pelo utilizador, sendo implementadas em várias linguagens de programação, o que normalmente resulta num *design* complexo do fluxo de trabalho ETL. Projetar um fluxo de trabalho ETL eficiente e sem erros é uma tarefa complexa e dispendiosa. Nos últimos anos têm sido vários os trabalhos de pesquisa relacionados com este tema e apesar desse esforço, ainda não existe nenhum consenso ou recomendação sobre as melhores práticas a serem aplicadas na modelação de sistemas de ETL.

Existem diversas técnicas para o desenvolvimento de um fluxo de trabalho ETL. Ao nível concetual, existem métodos que envolvem a notação UML, a notação BPMN e ainda outras notações proprietárias. Ainda não existe nenhuma estrutura que defina um fluxo de trabalho ETL. Perante isso existe a necessidade de uma estrutura ETL que reduza o trabalho do programador, numa perspetiva de *design*, otimização e manutenção. Essa estrutura deve fornecer recomendações sobre um *design* eficiente, um fluxo de trabalho que esteja de acordo com os requisitos de negócio e métodos que melhorem o desempenho do fluxo de trabalho ETL. O BPMN possui um método sistemático que traduz os requisitos de negócios num modelo concetual que visa implementar um projeto independente de fornecedor para o desenho do fluxo de trabalho ETL.

Apesar das contribuições mencionadas, continua a existir a necessidade de um único modelo de representação dos requisitos do fluxo ETL. Além disso, todas as abordagens discutidas neste trabalho exigem que o desenvolvedor de ETL forneça extensivamente alguma entrada

durante a fase de *design* de um fluxo de trabalho ETL, bem como exige conhecimento técnico de utilizadores de negócios para entender e validar um *design* ETL. Além disso, apesar do fato de que várias abordagens foram propostas, a comunidade de pesquisa ainda não concordou com a notação e abordagem padrão para representar um modelo concetual de um fluxo de trabalho ETL que possa ser progressivamente enriquecido com mais detalhes de acordo com as necessidades e fases do projeto.

A modelação de processos ETL com base no BPMN simplifica a representação dos fluxos, utilizando uma linguagem conhecida e de fácil compreensão que já é amplamente utilizada. A BPMN fornece um conjunto de primitivas que cobrem os requisitos de processos ETL utilizados com mais frequência e este conjunto de primitivas pode ser estendido aos requisitos de aplicações específicas. Uma das vantagens da utilização desta notação deve-se ao BPMN já ser utilizado para especificar processos de negócios no geral, portanto, os utilizadores finais não necessitam de aprender outra notação para especificar os processos ETL [15]. Outra das vantagens inerente ao BPMN é o facto de fornecer uma especificação conceptual independente de implementação de tais processos. Ao ocultar detalhes técnicos na modelação conceptual do processo, permite que os utilizadores finais e *designers* se concentrem nas características essenciais desses processos. Apesar da sua expressividade ser importante no desenvolvimento de modelos conceptuais, acarreta vários problemas de consistência e de interpretação, dado que o mesmo processo pode ser modelado de várias formas distintas. Para reduzir as redundâncias na forma como um processo pode ser modelado, estudou-se uma abordagem de modelação consistente e clara para vários tipos de utilizadores que suporta a modelação de processos de forma estruturada. Com este trabalho, foi validada a adequabilidade desta abordagem quando enquadrada com processos de ETL, que embora sejam processos de negócio, têm características específicas.

A utilização da abordagem proposta visa disponibilizar regras e métodos gerais que descrevem como um problema deve ser resolvido, independentemente do contexto em que será usada e independentemente da ferramenta de implementação física. Serve como ponte entre o esquema concetual e a implementação física dos processos. Esta abordagem fornece documentação detalhada de todo o processo quer para os utilizadores a nível do negócio quer para os utilizadores mais avançados responsáveis pela implementação, dando a estes últimos, orientações fundamentais, que agilizarão todo o processo ETL. A proposta de modelação teve como base outros trabalhos pioneiros na aplicação de BPMN ao domínio de ETL [15], [10], que tem sido explorado outros autores [47]. Com este trabalho, a abordagem

apresentada em [54] foi explorada, utilizando três níveis de representação: Modelação Descritiva, Modelação Analítica e Modelação Executável. O trabalho desenvolvido focou-se num estudo e experimento utilizando um processo de engenharia reversa no qual os processos ETL implementados fisicamente foram analisados e modelados conceptualmente em BPMN.

7.2 Trabalho futuro

A adoção de um modelo e abordagem que suporte o planeamento e desenvolvimento de processos ETL revela-se, cada vez mais também imprescindível para o sucesso global no desenvolvimento de sistemas de *data warehousing* e sistemas analíticos em geral.

A utilização da notação BPMN proporciona uma grande vantagem uma vez que permite explorar o desenho dos processos sobre várias perspetivas, o que aumenta a expressividade do processo principalmente numa fase inicial de desenvolvimento. No entanto, para a implementação de processos a partir de especificações BPMN, pode surgir ambiguidade o que dificulta a interpretação dos processos a nível de execução. Este trabalho visou reduzir essa ambiguidade através da implementação de uma metodologia de representação ETL fornecendo um conjunto de diretrizes e melhores práticas demonstradas através de um caso de estudo. Foram selecionados cuidadosamente os artefactos da notação BPMN que devem ser utilizados para a caracterização de processos de ETL considerando diferentes níveis de abstração.

Embora os objetivos planeados para este trabalho tenham sido alcançados, várias linhas de pesquisa podem ser seguidas para o enriquecimento e aprimoramento da abordagem. Possíveis trabalhos complementares e futuros passam por: modelar outros casos de estudo seguindo a abordagem proposta de modo a validar o método; e também aumentar o âmbito de validação do modelo através da modelação de cenários ETL reais.

A abordagem apresentada permite reduzir a distância na tradução de um modelo conceptual para um plano físico, atenuando um pouco a distância que existe entre a conceção e a implementação real. A modelação concetual de uma atividade é bastante útil, para a compreensão e implementação de qualquer sistema. A descrição de todas as atividades ETL e do seu comportamento contribuem para que o esforço de construção de modelos seja mais proveitoso, não só na fase de esboço, discussão ou de estudo dos inúmeros aspetos de um

sistema de ETL, mas também posteriormente na fase de implementação, onde muito do material utilizado na especificação pode ser utilizado como uma base solida na implementação física. Neste trabalho foi definido um método de modelação que futuramente pode ser descrito de forma comportamental e traduzido num pacote ETL que pode ser importado numa ferramenta ETL e enriquecido apenas com pormenores mais “físicos” (como a definição de ligações aos repositórios de dados), deixando a lógica de processo para níveis de representação de mais alto nível.

Bibliografia

[1] I. Yaqoob *et al.*, «Big Data: From Beginning to Future», *Int J Inf Manage*, vol. 36, n. 6, pp. 1231–1247, Dez. 2016, doi: 10.1016/j.ijinfomgt.2016.07.009.

[2] B. Inmon, «Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump», 2016.

[3] M. Janssen, H. van der Voort, e A. Wahyudi, «Factors influencing big data decision-making quality», *J Bus Res*, vol. 70, pp. 338–345, Jan. 2017, doi: 10.1016/j.jbusres.2016.08.007.

[4] R. Kimball e J. Caserta, «The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data», 2004, Acedido: Fev. 08, 2021. [Em linha]. Available: http://dl.acm.org/citation.cfm?id=1201627

[5] «Fluxo de Informação em ambiente organizacional. - E-LIS repository». http://eprints.rclis.org/33914/ (acedido Fev. 27, 2021).

[6] R. Kimball e M. Ross, *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, Second. Wiley, 2002, p. 464.

[7] T. Guo, C. Xu, B. Shi, C. Xu, e D. Tao, «Learning from Bad Data via Generation», Set. 2019.

[8] M. Weske, W. M. P. van der Aalst, e H. M. W. Verbeek, «Advances in business process management», em *Data and Knowledge Engineering*, Jul. 2004, vol. 50, n. 1, pp. 1–8. doi: 10.1016/j.datak.2004.01.001.

[9] J. Merino, I. Caballero, B. Rivas, M. Serrano, e M. Piattini, «A Data Quality in Use model for Big Data», *Future Generation Computer Systems*, vol. 63, pp. 123–130, Out. 2016, doi: 10.1016/j.future.2015.11.024.

[10] Z. el Akkaoui, J.-N. N. Mazón, A. Vaisman, e E. Zimányi, «BPMN-Based Conceptual Modeling of ETL Processes», *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7448, pp. 1–14, 2012, doi: 10.1007/978-3-642-32584-7_1.

[11] R. Iru *et al.*, «Conceptual modeling for ETL processes», *Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP - DOLAP '02*, pp. 14–21, 2002, doi: 10.1145/583890.583893.

[12] G. Aagesen e J. Krogstie, «Bpmn 2.0 for modeling business processes», em *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*, Springer Berlin Heidelberg, 2015, pp. 219–250. doi: 10.1007/978-3-642-45100-3_10.

[13] W. M. P. van der Aalst, «Business Process Execution Language», em *Encyclopedia of Database Systems*, Boston, MA: Springer US, 2009, pp. 288–289. doi: 10.1007/978-0-387-39940-9_1194.

[14] F. Barbier, «Moving Process Models from the Conceptual to the Executable», *Reactive Internet Programming*, p. 85, 2016, [Em linha]. Available: https://www.omg.org/oceb-2/documents/ExecutionSemantics-Moving-BPMN-toExecution-FinalPosting.pdf

[15] Z. el Akkaoui e E. Zimányi, «Defining ETL worfklows using BPMN and BPEL», em *Proceeding of the ACM twelfth international workshop on Data warehousing and OLAP DOLAP 09*, 2009, pp. 41–48. doi: 10.1145/1651291.1651299.

[16] B. S. Reale, «BPMN 2.0 for beginners», pp. 1–23, 2019.

[17] OMG, «Business Process Model and Notation (BPMN)», 2014. [Em linha]. Available: https://www.omg.org/

[18] M. Owen e J. Raj, «BPMN and Business Process Management», 2003. [Em linha]. Available: www.oasis-open.org

[19] T. Allweyer, «BPMN 2.0 Introduction to the standard for business process modeling», 2016.

[20] S. Bruce, «BPMN Method and Style», pp. 1–345, 2009.

[21] R. F. Pucpr, M. Antonio, e B. de Paula, «Traduzindo a definição de processo em xpdl para modelos em redes de petri», *Encontro Nacional de Engenharia de Produção*, 2009.

[22] «BPMN and Business Process Management», 2003. [Em linha]. Available: www.oasis-open.org

[23] A. Mesa e M. S. Tabares, «Comparativo entre herramientas BPMN Christian Lochmuller», 2014, doi: 10.14508/sdp.2014.6.12.95-108.

[24] S. A. White, «Introduction to BPMN», 2004. [Em linha]. Available: www.bptrends.com

[25] R. Kimball, M. Ross, B. Becker, J. Mundy, e W. Thornthwaite, *Relentlessly Practical Tools for Data Warehousing and Business Intelligence*. 2016.

[26] M. Golfarelli e S. Rizzi, «A methodological framework for data warehouse design», *DOLAP: Proceedings of the ACM International Workshop on Data Warehousing and OLAP*, vol. Part F1292, pp. 3–9, 1998, doi: 10.1145/294260.294261.

[27] J. Ferreira, M. Miranda, A. Abelha, e J. Machado, «O Processo ETL em Sistemas Data Warehouse», *INForum 2010 - II Simpósio de Informática*, n. January 2010, pp. 757–765, 2010.

[28] P. Vassiliadis, A. Simitsis, P. Georgantas, M. Terrovitis, e S. Skiadopoulos, «A generic and customizable framework for the design of ETL scenarios», *Inf Syst*, vol. 30, n. 7, pp. 492–525, Nov. 2005, doi: 10.1016/j.is.2004.11.002.

[29] P. Vassiliadis, «A Survey of Extract-Transform-Load A Survey of Extract – Transform – Load Technology», vol. 5, n. May, pp. 1–27, 2009.

[30] P. Vassiliadis, «Gulliver in the land of data warehousing: practical experiences and observations of a researcher», *P.Vassiliadis*, vol. 12, n. 1, 2000, [Em linha]. Available: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-28/

[31] B. Oliveira e O. Belo, «BPMN Patterns for ETL Conceptual Modelling and Validation», *The 20th International Symposium on Methodologies for Intelligent Systems: Lecture Notes in Artificial Intelligence*, vol. 7661 LNAI, pp. 445–454, 2012, doi: 10.1007/978-3-642-34624-8_50.

[32] B. Oliveira e O. Belo, «ETL Standard Processes Modelling A Novel BPMN Approach», pp. 120–127, 2013, doi: 10.5220/0004418301200127.

[33] L. Mũoz, J. N. Mazón, e J. Trujillo, «Automatic generation of ETL processes from conceptual models», *International Conference on Information and Knowledge Management, Proceedings*, pp. 33–40, 2009, doi: 10.1145/1651291.1651298.

[34] P. Vassiliadis, A. Simitsis, e S. Skiadopoulos, «Conceptual modeling for ETL processes», pp. 14–21, 2002.

[35] A. Simitsis e P. Vassiliadis, «A Methodology for the Conceptual Modeling of ETL Processes», em *CAiSE'03: Proceedings of the 15th International Conference on Advanced Information Systems Engineering*, 2003, pp. 305–316.

[36] J. Trujillo, I. Y. Song, S. Luja, S. Luján-Mora, J. Trujillo, e I. Y. Song, «A UML profile for multidimensional modeling in data warehouses», *Data Knowl Eng*, vol. 59, n. 3 SPEC. ISS., pp. 725–769, 2006, doi: 10.1016/j.datak.2005.11.004.

[37] M. Mrunalini, T. V. S. Kumar, e K. R. Kanth, «Simulating secure data extraction in Extraction Transformation Loading (ETL) processes», *EMS 2009 - UKSim 3rd European Modelling Symposium on Computer Modelling and Simulation*, pp. 142–147, 2009, doi: 10.1109/EMS.2009.111.

[38] L. Muñoz, J. N. Mazón, J. Pardillo, e J. Trujillo, «Modelling ETL processes of data warehouses with UML activity diagrams», *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5333, pp. 44–53, 2008, doi: 10.1007/978-3-540-88875-8_21.

[39] N. Biswas, S. Chattopadhyay, G. Mahapatra, S. Chatterjee, e K. C. Mondal, «SysML based conceptual ETL process modeling», em *Communications in Computer and Information Science*, 2017, vol. 776, pp. 242–255. doi: 10.1007/978-981-10-6430-2_19.

[40] N. Biswas, S. Chattapadhyay, G. Mahapatra, S. Chatterjee, e K. C. Mondal, «A new approach for conceptual extraction-transformation-loading process modeling», *International Journal of Ambient Computing and Intelligence*, vol. 10, n. 1, pp. 30–45, Jan. 2019, doi: 10.4018/IJACI.2019010102.

[41] P. Vassiliadis, A. Simitsis, P. Georgantas, M. Terrovitis, e S. Skiadopoulos, *A generic and customizable framework for the design of ETL scenarios*, vol. 30, n. 7. Pergamon, 2005, pp. 492–525. doi: 10.1016/j.is.2004.11.002.

[42] N. Tech, «Optimizing ETL Processes in Data Warehouses Alkis Simitsis», *Data Engineering*, n. Icde, pp. 564–575, 2005, [Em linha]. Available: http://www.computer.org/portal/web/csdl/doi/10.1109/ICDE.2005.103

[43] Z. el Akkaoui *et al.*, «A model-driven framework for ETL process development», em *International Conference on Information and Knowledge Management, Proceedings*, 2011, pp. 45–52. doi: 10.1145/2064676.2064685.

[44] Z. el Akkaoui, E. Zimányi, J.-N. N. Mazón, e J. Trujillo, «A BPMN-based design and maintenance framework for ETL processes», *International Journal of Data Warehousing and Mining*, vol. 9, n. 3, pp. 46–72, Jul. 2013, doi: 10.4018/jdwm.2013070103.

[45] O. Belo, C. Gomes, B. Oliveira, R. Marques, e V. Santos, «Automatic generation of ETL physical systems from BPMN conceptual models», em *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9344, pp. 239–247. doi: 10.1007/978-3-319-23781-7_19.

[46] B. Oliveira, V. Santos, e O. Belo, «Pattern-based ETL conceptual modelling», em *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, vol. 8216 LNCS, pp. 237–248. doi: 10.1007/978-3-642-41366-7_20.

[47] B. Oliveira, Ó. Oliveira, e O. Belo, «Using BPMN for ETL conceptual modelling: A case study», *Proceedings of the 10th International Conference on Data Science, Technology and Applications, DATA 2021*, n. Section 3, pp. 267–274, 2021, doi: 10.5220/0010575702670274.

[48] B. Oliveira e O. Belo, «BPMN Patterns for ETL Conceptual Modelling and Validation», *The 20th International Symposium on Methodologies for Intelligent Systems: Lecture Notes in Artificial Intelligence*, vol. 7661 LNAI, pp. 445–454, 2012, doi: 10.1007/978-3-642-34624-8_50.

[49] B. Oliveira e O. Belo, «An Ontology for Describing ETL Patterns Behavior», n. Data, pp. 102–109, 2016, doi: 10.5220/0005974001020109.

[50] B. Oliveira, Ó. Oliveira, e O. Belo, «Using BPMN for ETL conceptual modelling: A case study», *Proceedings of the 10th International Conference on Data Science, Technology and Applications, DATA 2021*, pp. 267–274, 2021, doi: 10.5220/0010575702670274.

[51] B. Oliveira e O. Belo, «Task clustering on ETL systems: A pattern-oriented approach», *DATA 2015 - 4th International Conference on Data Management Technologies and Applications, Proceedings*, pp. 207–214, 2015, doi: 10.5220/0005559302070214.

[52] B. Silver, *BPMN method and style : with BPMN implementer's guide*. 2011.

[53] M. Guimarães e O. Belo, «Geração Automática de Esqueletos para Sistemas ETL», *XIX Jornadas de Ingeniería del Software y Bases de Datos (JISBD'2014)*, pp. 27–32, 2014.

[54] B. Oliveira e O. Belo, «Approaching ETL conceptual modelling and validation using BPMN and BPEL», *DATA 2013 - Proceedings of the 2nd International Conference on Data Technologies and Applications*, n. Section 2, pp. 191–198, 2013, doi: 10.5220/0004563001910198.

[55] R. Kimball e M. Ross, *The Data Warehouse Toolkit The Definitive Guide to Dimensional Modeling*. 2013.

[56] B. Oliveira, V. Santos, C. Gomes, R. Marques, e O. Belo, «Conceptual-physical bridging - From BPMN models to physical implementations on kettle», *CEUR Workshop Proc*, vol. 1418, pp. 55–59, 2015.