



**Contemporânea**

*Contemporary Journal*

3(8): 12268-12296, 2023

ISSN: 2447-0961

Artigo

# **MODELANDO O CONHECIMENTO PARA APLICAÇÕES EM AMBIENTE DE DESENVOLVIMENTO DE SOFTWARE**

MODELING KNOWLEDGE FOR APPLICATIONS IN A SOFTWARE DEVELOPMENT ENVIRONMENT

DOI: 10.56083/RCV3N8-131

Recebimento do original: 18/07/2023

Aceitação para publicação: 18/08/2023

## **Bárbara Beatriz Bueno de Oliveira**

Graduada em Sistemas de Informação

Instituição: F1RST Digital Services

Endereço: Rua Professora Alice J Danna Juliana, 46, Jardim Paraíso, CEP: 13561-080

E-mail: ba.bueno.oliveira@gmail.com

## **Dildre Georgiana Vasques**

Doutora em Ciências de Computação e Matemática Computacional

Instituição: Universidade de São Paulo

Endereço: Av. Prof. Luciano Gualberto, 158, Butantã, São Paulo - SP, CEP: 05508-900

E-mail: dildre.vasques@gmail.com

## **Franciene Duarte Gomes**

Doutora em Tecnologia

Instituição: Universidade Federal de Mato Grosso do Sul (UFMS)

Endereço: Av. Ranulpho Marques Leal, 3484, Três Lagoas - MS, CEP: 79613-000

E-mail: franciene.gomes@ufms.br

**RESUMO:** A tomada de decisão se torna mais eficiente em um ambiente de desenvolvimento de software quando está pautada no uso e na assimilação do conhecimento. As tomadas de decisões são embasadas pelos requisitos funcionais uma vez que os mesmos descrevem o que o sistema deve fazer. Quando esses requisitos são mal compreendidos, especificados e gerenciados, ocorrerão problemas nos produtos de software. Assim, foram pesquisados trabalhos relacionados ao tema que mostram a automatização ou semi-automatização do processo de construção de diagramas, porém nenhum dos trabalhos focam na extração do conhecimento e na pré-



modelagem para domínio dos requisitos. Nesse contexto, o objetivo deste trabalho foi aplicar uma ferramenta focada na aquisição do conhecimento. Essa ferramenta é baseada em técnicas de processamento de linguagem natural e foi utilizada para construir uma pré-modelagem de software que apoiasse a extração do conhecimento. Assim, utilizou-se a ferramenta para processar um texto de visão de negócio de um sistema de livraria virtual. Com base na tabela de fatos e no mapa conceitual gerados pela ferramenta, identificou-se os principais objetos de negócio e como estes se relacionam. A partir dessas informações, elaborou-se uma versão inicial de um diagrama de caso de uso e de um diagrama de classes. A aplicação da ferramenta mostrou-se capaz de colaborar com a identificação dos conceitos fundamentais que constituíam o domínio. Esse resultado facilitou o levantamento e entendimento dos requisitos funcionais de software, necessários para a satisfação das necessidades do usuário.

**PALAVRAS-CHAVE:** Engenharia de Requisitos, Extração de Conhecimento, Gestão do Conhecimento, Modelagem de Software.

**ABSTRACT:** Decision-making becomes more efficient in a software development environment when using and assimilating knowledge. Functional requirements that describe what the system must do influence decision-making. When these requirements need to be better understood, specified, and managed, problems will occur in software products. Thus, the literature shows the automation or semi-automation of the diagram construction process. Generally, the studies do not focus on knowledge extraction and pre-modeling to master the requirements. In this context, the objective of this work was to apply a tool focused on acquiring knowledge. Our proposal is based on natural language processing techniques and was used to build pre-modeling software to support knowledge extraction. Thus, the study uses the tool to process a business vision text from a virtual bookstore system. The study identified the main business objects and how they are related based on the fact table and the conceptual map. The work creates an initial version of a use case diagram and a class diagram from this information. The application of the tool proved to be able to collaborate with the identification of the fundamental concepts that constituted the domain. This result facilitated the survey and understanding of the functional software requirements necessary to satisfy the user's needs.

**Keywords:** Knowledge Extraction, Knowledge management, Requirements Engineering, Software Modeling.



Artigo está licenciado sob forma de uma licença  
Creative Commons Atribuição 4.0 Internacional.



## **1. Introdução**

A tomada de decisão em um ambiente de desenvolvimento de software representa um processo fundamental para assegurar as estratégias e o cumprimento dos objetivos. Para que esta tomada de decisão seja eficiente, ela deve estar atrelada ao uso e a assimilação do conhecimento. Nesse sentido, ferramentas e práticas de Gestão do Conhecimento (GC) representam tecnologias que se tornam indispensáveis para o bom desempenho de um processo de desenvolvimento de software [PAULA and TATTO 2020]. O uso dessas ferramentas e práticas podem proporcionar o domínio dos conceitos fundamentais e este domínio facilita o desenvolvimento do processo [HANSEN and BIRKINSHAW 2007].

O processo de desenvolvimento de software diz respeito a um conjunto de atividades que são utilizadas para definir, desenvolver, testar e manter um software [SOMMERVILLE 2019]. Os recursos de GC, por sua vez, têm como objetivo gerar, codificar, coordenar e transferir o conhecimento existente [DAVENPORT and PRUSAK 1998], podendo ajustar-se perfeitamente ao processo de desenvolvimento de software, já que este é adaptável e possibilita a equipe de desenvolvedores a seleção e escolha de ações e tarefas apropriadas para construir um produto de qualidade para os seus usuários [PRESSMAN 2021].

Esse processo está pautado em algumas atividades essenciais comuns ligadas ao levantamento de requisitos, análise de requisitos, projeto, implementação, testes e implantação [DEVMEDIA 2007]. Nesse cenário, os requisitos de software representam uma atividade de extrema importância para o desenvolvimento de um produto de software de qualidade. A importância em compreender os requisitos é tão grande a ponto de existir a Engenharia de Requisitos (ER), uma disciplina dentro da engenharia de software com o objetivo de identificar as necessidades das partes



interessadas. Em síntese, a ER estabelece uma ponte entre o projeto e a construção do software [PRESSMAN 2021].

Uma das principais causas de problemas nesses produtos está ligada a falhas nessa variável, o que leva a geração de requisitos mal compreendidos, mal especificados e mal gerenciados [REINEHR 2020]. Esses problemas podem afetar o desempenho de produtos de software e conduzem projetos ao fracasso. No planejamento de um projeto de software a identificação dos requisitos está relacionada diretamente as necessidades do cliente [MORAIS and ZANIN 2020].

A documentação na ER estabelece a comunicação entre todos os envolvidos, nivelando o conhecimento sobre o projeto. Desse modo, a documentação funciona como um ativo específico para a transmissão e compartilhamento do conhecimento [SOUZA et al. 2019]. Portanto, diferentes trabalhos veem sendo desenvolvidos com o objetivo de facilitar a identificação dos requisitos e a modelagem de software baseada na geração de diagramas da Linguagem de Modelagem Unificada (UML) com base nos requisitos [DAWOOD et al. 2017]; [VASQUES et al. 2019a]. Um diagrama UML é composto por elementos relacionados entre si, representando um modo de ilustrar uma linguagem de notação para se usar em projetos de sistemas. Desse modo, o escopo se torna mais claro, já que esse modelo de visualização é capaz de centralizar um determinado conceito e revelar os seus relacionamentos.

Atualmente, nota-se uma particular ênfase na utilização de Processamento de Linguagem Natural (PLN) como método de auxílio à geração de diagramas UML [ELALLAOUI et al. 2018]; [HAMZA and HAMMAD 2019]; [BASHIR et al. 2021]. PLN é uma vertente da Inteligência Artificial que tem por objetivo ajudar os computadores a interpretar e manipular a linguagem humana [NADEKARNI et al. 2011].



Os trabalhos relacionados listados na seção 3 mostram que a automatização ou semi-automatização do processo de construção de diagramas UML podem melhorar o levantamento dos requisitos, fazendo com que esse processo se torne menos laborioso e gaste um tempo menor para a sua execução. No entanto, nenhum desses trabalhos utilizou uma abordagem focada na extração do conhecimento e na sua pré-modelagem como uma ferramenta de auxílio para o entendimento e o domínio dos conceitos fundamentais dos requisitos e de todos os seus relacionamentos.

Nesse cenário, o objetivo deste trabalho foi utilizar uma ferramenta de aquisição do conhecimento, capaz de extrair e modelar o conhecimento capturado em tabelas estruturadas e mapas conceituais causais, para apoiar a modelagem de diagramas UML na fase de pré-modelagem do software. O experimento realizado teve também como escopo mostrar a importância do uso de ferramentas de Gestão do Conhecimento no apoio as atividades de ER.

O artigo está dividido do seguinte modo: a Seção 2 apresenta alguns conceitos fundamentais para a compreensão da área de conhecimento investigada; na Seção 3 traz a descrição de alguns trabalhos relacionados; na Seção 4 é apresentada a metodologia utilizada; a Seção 5 traz a aplicação da ferramenta em um experimento e a análise dos resultados e, por fim, na Seção 6 são apresentadas as conclusões.

## **2. Fundamentação Teórica**

A seção 2 foi subdividida entre as seções de Engenharia de Requisitos, Modelagem de Software e Processamento de Linguagem Natural (PNL).



## 2.1 Engenharia de Requisitos

A Engenharia de Requisitos (ER) é o processo de descobrir, analisar, documentar e verificar as descrições sobre aquilo que o sistema deve fazer de acordo com aquilo que os usuários (clientes) necessitam [SOMMERVILLE 2019]. Segundo [SOMMERVILLE 2019], na engenharia de requisitos existem três fases fundamentais para o desenvolvimento das atividades, descritas a seguir:

- **Elicitação de Requisitos:** visa compreender o trabalho realizado pelos usuários para entender como o novo sistema apoiaria o trabalho deles. Nesta fase são levantados, entre outros itens, o domínio da aplicação, os serviços, as características e as limitações do sistema.
- **Especificação de Requisitos:** tem a finalidade de transcrever as informações dos requisitos dos usuários e do sistema que foram coletadas para o documento de requisitos.
- **Validação de Requisitos:** busca verificar se os requisitos adquiridos se encaixam no sistema que o cliente solicitou. Nesta fase, quando encontrada algum erro, este deve ser rapidamente corrigido, evitando problemas futuros.

Para [SOMMERVILLE 2019], na ER pode haver dois tipos de descrição de "requisitos de usuário". Antes de começar a implementar o sistema, é necessário identificar a qual tipologia estes pertencem, ou seja, se são requisitos funcionais ou não funcionais. Os requisitos funcionais são aqueles que especificam o que um sistema deve fazer de acordo com o tipo de software e o público alvo, além de descrever a abordagem que a organização adotou para chegar aos requisitos. Desse modo, o objetivo dos requisitos funcionais é traçar as funções do sistema, entradas, saídas e exceções.

Para que o sistema seja desenvolvido de maneira coerente é necessário que nos requisitos funcionais também estejam detalhados os serviços e



informações que o cliente solicitou. Os requisitos não funcionais, por sua vez, não possuem uma relação direta com os serviços que o sistema deve oferecer. Esses requisitos estão relacionados à confiabilidade do sistema, tempo de resposta e uso da memória. Esses requisitos podem definir quais serão as restrições que o sistema apresentará, como a capacidade dos dispositivos de E/S (entrada e saída) ou as representações dos dados que a interface utilizará.

## 2.2 Modelagem de Software

A modelagem de software é uma maneira de representar um sistema usando algum tipo de notação gráfica como os diagramas UML ou os modelos formais (matemáticos). A seguir são descritos dois modelos de diagramas UML citados e utilizados no desenvolvimento deste trabalho.

- **Diagrama de Caso de Uso:** O diagrama de caso de uso representa um comportamento formado por uma sequência de ações produzidas pelo sistema [BOOCH et al. 2012]. A modelagem do diagrama de caso de uso é utilizada para auxiliar na obtenção de informações detalhadas sobre determinado assunto.
- **Diagrama de Classes:** O diagrama de classes mostra os atributos e operações de uma classe e as restrições que se aplicam ao modo como os objetos são conectados [BOOCH et al. 2012]. A classe é uma descrição de um conjunto de objetos e são identificadas por substantivos ou frases nominais. O diagrama de classes pode ser dividido em modelo de análise e modelo de projeto. O diagrama de classe modelo projeto é usado para visualizar uma perspectiva de software e neste diagrama as classes são apresentadas como componentes do software. Já o modelo de análise (ou domínio) é usado para visualizar uma perspectiva conceitual e neste modelo é possível ter uma abstração de conceitos ou objetos do mundo real [LARMAN



2008].

### 2.3 Processamento de Linguagem Natural

O Processamento de Linguagem Natural é uma área de pesquisa e aplicação que busca usar os computadores para entender e manipular texto ou fala, em linguagem natural, para realizar diversas tarefas desejadas [NADEKARNI et al. 2011]. Assim, pode-se dizer que o PLN é o estudo científico das linguagens a partir da perspectiva computacional [KUMAR 2013].

O volume abundante de texto em linguagem natural produzido diariamente no mundo conectado faz com que a disseminação do conteúdo de conhecimento ali presente se torne cada vez mais difícil. O PLN automatizado objetiva realizar esta tarefa de modo eficaz e com precisão [CHOWDHARY 2020].

Atualmente, essa área de pesquisa empresta saberes de vários campos científicos, exigindo que os pesquisadores ampliem significativamente sua base de conhecimento mental [NADEKARNI et al. 2011]. Isso ocorre porque os fundamentos do PLN estão em várias disciplinas, como informática, linguística, matemática, engenharia elétrica, inteligência artificial, psicologia, entre outras.

Em síntese, o objetivo último do PLN é extrair significado do texto, ou seja, a sua semântica. Desse modo, os pesquisadores do PLN reúnem conhecimento sobre como os seres humanos entendem e usam a linguagem para que ferramentas e técnicas apropriadas possam utilizar esse conhecimento para a realização das mais diversas tarefas [NADEKARNI et al. 2011].





### 3. Revisão da Literatura

Esta revisão aborda trabalhos que mostram processos, técnicas, métodos e ferramentas com foco na compreensão do conhecimento na fase de levantamento de requisitos de software. Para isso foi feita uma pesquisa bibliográfica entre o período de 2017 a 2022, utilizando os seguintes termos: “*Software Modeling*”, “*Software Pre-Modeling*”, “*Conceptual Level*”, “*Extraction*”, “*Requirements Analysis*”, “*Use Case Diagram*”, “UML” “*Natural Language Processing*”. A pesquisa foi executada nas bases de dados: Science Direct e Google Academics. A partir da metodologia PRISMA [PRISMA 2009], que é um conjunto de itens baseado em evidências para relato em revisões sistemáticas e metanálises, foi realizada uma revisão bibliográfica.

Os trabalhos relacionados mostram como a GC e em particular, o processo de extração do conhecimento baseado em PLN, pode contribuir de maneira efetiva no desenvolvimento de produtos de software. Como exemplo desse tipo de pesquisa, o trabalho de [VASQUES et al. 2019a] traz a aplicação de uma abordagem de Aprendizagem Baseada em Problemas (PBL) [MOYA 2017], implementando um processo educativo baseado em GC para a extração de requisitos. Os autores realizaram um experimento, dividido em duas fases, com um grupo de 32 alunos de uma universidade pública. Na primeira fase, os alunos extraíram os requisitos e criaram um diagrama de caso de uso UML conforme o modelo tradicional (lendo a história do usuário) e, na segunda fase, utilizaram como suporte um processo de aquisição de conhecimento baseado em semântica verbal (processo Verbka) para uma pré-modelagem dos diagramas. Os resultados obtidos das respostas dos questionários aplicados aos alunos, ao final do experimento, mostraram que a utilização do processo de extração de conhecimento na fase da pré-modelagem facilitou a extração dos requisitos e o desenvolvimento do diagrama de caso de uso.

Em um outro experimento realizado pelos mesmo autores [VASQUES



et al. 2019b] foi apresentado um modo de facilitar a criação de diagramas de caso de uso por meio da identificação dos atores envolvidos nos requisitos de negócios. Com o auxílio do processo Verbka os alunos foram capazes de capturar o conhecimento contido nos textos (não estruturados, escritos em linguagem livre) e descrevê-lo de modo estruturado. Concluiu-se que o processo auxiliou na pré-modelagem e conseqüentemente na obtenção de uma lista de requisitos, atores, casos de uso e funcionalidades de casos de uso, mostrando como todas essas variáveis se relacionam entre si. Esses resultados foram obtidos de forma mais sistemática e eficiente se comparada à interação direta com informações textuais por meio da leitura, reduzindo erros e possibilitando a economia de tempo e esforços na análise e extração de requisitos.

[DAWOOD et al. 2017], por meio de um questionário distribuído mundialmente entre diversos grupos selecionados, como empresas de TI, universidades e grupos de pesquisas, puderam conhecer como as organizações geraram e utilizavam os requisitos e os seus respectivos diagramas. Analisaram mais de 90 respostas, as quais indicaram que apenas 13% dos entrevistados geravam diagramas UML de modo automático e cerca de 23,9% de maneira semiautomática, enquanto 40,2% geravam diagramas UML manualmente. O processo manual, mesmo gerando bons resultados, é laborioso e um tempo considerável é gasto para a sua execução. Nesse sentido, estudos com foco em PLN aplicado à geração de Diagramas UML vêm ganhando um espaço cada vez maior no campo da ER. Esses estudos têm por objetivo melhorar o processo de construção de Diagramas UML a partir do conhecimento extraído dos requisitos.

Em [VEMURI et al. 2017], o objetivo foi desenvolver uma ferramenta que aprendesse a partir de padrões de requisitos, reduzindo a dependência de regras gramaticais para identificar casos de uso. Foi utilizado um algoritmo para identificar os atores e casos de uso de qualquer texto através do método de aprendizado supervisionado. Foram utilizados três documentos



de requisitos do usuário para o teste final e obtiveram um resultado de 70% de precisão. Os autores concluíram que além da identificação dos requisitos funcionais, também é necessário lidar com os requisitos não funcionais e que melhorias nesta abordagem poderiam automatizar completamente a fase de análise.

[ELALLAOUI et al. 2018] apresentaram a integração de ferramentas de PLN em requisitos ágeis em histórias de usuários específicas para geração de modelos UML. Segundo os autores, o método possibilita uma interpretação clara das histórias do usuário, reduzindo o tempo de desenho de diagramas de caso de uso e melhorando o fluxo de trabalho. Por fim, concluíram que foram obtidos bons resultados quando as histórias de usuários possuem uma sintaxe simples e compreensível, no entanto, perceberam que, quando as sentenças são mais complexas, o método apresenta algumas falhas.

[HAMZA and HAMMAD 2019] usaram a verificação ortográfica dos textos de requisitos e a identificação dos elementos do diagrama de caso de uso em conjunto com o PLN. Os autores utilizaram quatro estudos de caso para a identificação e a extração dos requisitos e obtiveram como resultados uma média de precisão de 72%. Os autores concluíram que para se obter melhores resultados seria necessário o uso de ferramentas de *chunking* para combinar uma sequência de palavras em uma *tag*.

Na pesquisa de [Ahmed et al. 2021] foi utilizada uma abordagem automatizada de PLN para gerar Diagramas de Entidade-Relacionamento (ER) a partir dos requisitos iniciais do software. Os autores seguiram a metodologia do PLN com o diferencial de novas regras de divisão de sentenças, tokenização e marcações PoS (etiquetar gramaticalmente elementos textuais na identificação de palavras). Utilizou-se a ferramenta Entity Relationship Diagram (ERD-Gen), desenvolvida como parte do estudo de pesquisa para gerar o modelo ERD automaticamente a partir do conjunto preliminar de requisitos.

Já na pesquisa de [BASHIR et al. 2021], foi proposto um sistema para



análise de projeto em ER, denominado READ. O objetivo desse sistema é gerar diagramas de classe UML usando PLN juntamente com técnicas de ontologia de domínio. A ferramenta proposta foi capaz de obter resultados satisfatórios em seus testes, com desempenho de 94% de recall e 69% de precisão.

Na pesquisa de [SANTOS 2021], foi desenvolvido um protótipo de uma ferramenta Web para a implementação de um framework que tem a capacidade de proporcionar a automatização na estruturação, validação e representação de requisitos de diferentes contextos. A autora teve como principal metodologia a construção de diversas fases para a implementação e desenvolvimento do framework. Para a fase de extração e estruturação do conhecimento, a autora utilizou a ferramenta AutRQ. A autora concluiu que esta ferramenta mostra eficiência em diferentes contextos da ER.

Todos esses trabalhos buscaram um modo de aplicar ferramentas que pudessem contribuir de maneira eficiente para a geração de algum tipo de diagrama UML com base em textos escritos em língua natural. Buscaram também uma maneira de conseguir extrair e modelar todos os requisitos apresentados pelos usuários por meio da análise linguística e/ou semântica dos conceitos e seus relacionamentos. A maioria dos trabalhos buscou um meio automatizado, através do uso de alguma ferramenta, para realizar a modelagem e a extração dos requisitos, além de utilizarem o PLN para auxiliar nesse processo. Porém nenhum trabalho utilizou uma abordagem focada na extração do conhecimento e na pré-modelagem para o entendimento dos requisitos funcionais e dos relacionamentos.

Portanto, assim como no trabalho de [SANTOS 2021], o experimento proposto neste trabalho utiliza a mesma ferramenta Verbka, com o intuito de extrair, compreender e modelar o conhecimento contido em textos de modo automatizado. O que difere esse trabalho dos trabalhos relacionados que utilizam ferramentas automatizadas para a construção de diagramas UML e extração de requisitos, está no foco dirigido à fase de pré-modelagem



do conhecimento, como um meio de capturar o conhecimento de modo sistemático, a fim de revelar as causas e efeitos presentes nos diversos relacionamentos entre os conceitos. Além disso, acredita-se que a representação do conhecimento explícito em mapas conceituais pode facilitar a compreensão holística do assunto em pauta e contribuir também para a identificação de relacionamentos implícitos.

Com base nos trabalhos relacionados descritos acima, foi gerada a Tabela 1 que apresenta as seguintes colunas: título do trabalho, o que foi desenvolvido (ferramenta ou processo), qual diagrama foi utilizado, a principal metodologia e qual o requisito foi abordado.

Tabela 1. Comparação dos Trabalhos Relacionados

<b>Trabalho</b>	<b>Processo</b>	<b>Ferramenta</b>	<b>Tipos de Diagramas</b>	<b>Principal Metodologia</b>	<b>Tipo de Requisitos</b>
Dawood (et al. 2017)	Processo de Identificar Tipos de Extração de Requisitos	-	Diagramas UML em geral	Questionários	Requisitos Funcionais
Vemuri (et al. 2017)	Processo de Extrair Autores e Caso de Uso	-	Diagrama de Caso de Uso	Algoritmo com Aprendizado Supervisionado	Requisitos Funcionais e Não Funcionais
Elallaoui (et al. 2018)	-	Ferramentas de PNL	Diagrama de Caso de Uso	Processamento de Linguagem Natural	Requisitos Funcionais
Hamza and Hammad (2019)	-	Processo de Linguagem Natural	Diagrama de Caso de Uso	Processamento de Linguagem Natural	Requisitos Funcionais
Vasques (et al. 2019a)	Verbka	-	Diagrama de Caso de Uso	Aprendizado Baseado em Problemas (PBL)	Requisitos Funcionais
Vasques (et al. 2019b)	Verbka	-	Diagrama de Caso de Uso	Revisão	Requisitos Funcionais
Ahmed (et al. 2021)	-	Entity Relationship Diagram (ERD-Gen)	Diagrama de Entidade-Relacionamento	Processamento de Linguagem Natural	Requisitos Funcionais



Bashir (et al. 2021)	-	Ferramenta READ	Diagrama de Classes	Implementação de Ferramenta	Requisitos Funcionais
Santos (2021)	-	Ferramenta AutRQ	Diagrama de Caso de Uso	Implementação de Ferramenta	Requisitos Funcionais
Oliveira (et al. 2022)	-	Verbka	Diagrama de Caso de Uso e Classes	Aplicação da Ferramenta	Requisitos Funcionais

Fonte: Autores

#### 4. Metodologia

Este trabalho segue uma abordagem qualitativa. Esse tipo de abordagem tem como intuito analisar e interpretar textos e imagens, a fim de apontar comportamentos e tendências com base nos dados gerados [CRESWELL and CRESWELL 2021]. Para a extração dos requisitos e construção do diagrama de classes [WAZLAWICK 2004], foi utilizada a ferramenta Verbka [VASQUES 2021].

Essa ferramenta foi desenvolvida com base em um processo de aquisição de conhecimento baseado em semântica verbal e regras gramaticais, usado para extrair, modelar e representar o conhecimento em tabelas estruturadas e mapas conceituais [VASQUES 2016]. Esse processo possui como metodologia a decodificação de um texto escrito em língua natural e a sua reorganização e estruturação em cadeias de relacionamentos causais. Por meio da semântica contida nas orações, o processo mostra que é possível reunir as informações presentes em um texto e apresentá-las visualmente em Grafos e/ou Mapas Conceituais Causais. O processo Verbka possui uma sequência de etapas (Figura 1), descritas a seguir:

1. **Seleção Textual:** são selecionados textos que transmitem algum significado para o processo, que demonstre claramente o que é necessário para o desenvolvimento do projeto ou qual o problema que deve ser resolvido.

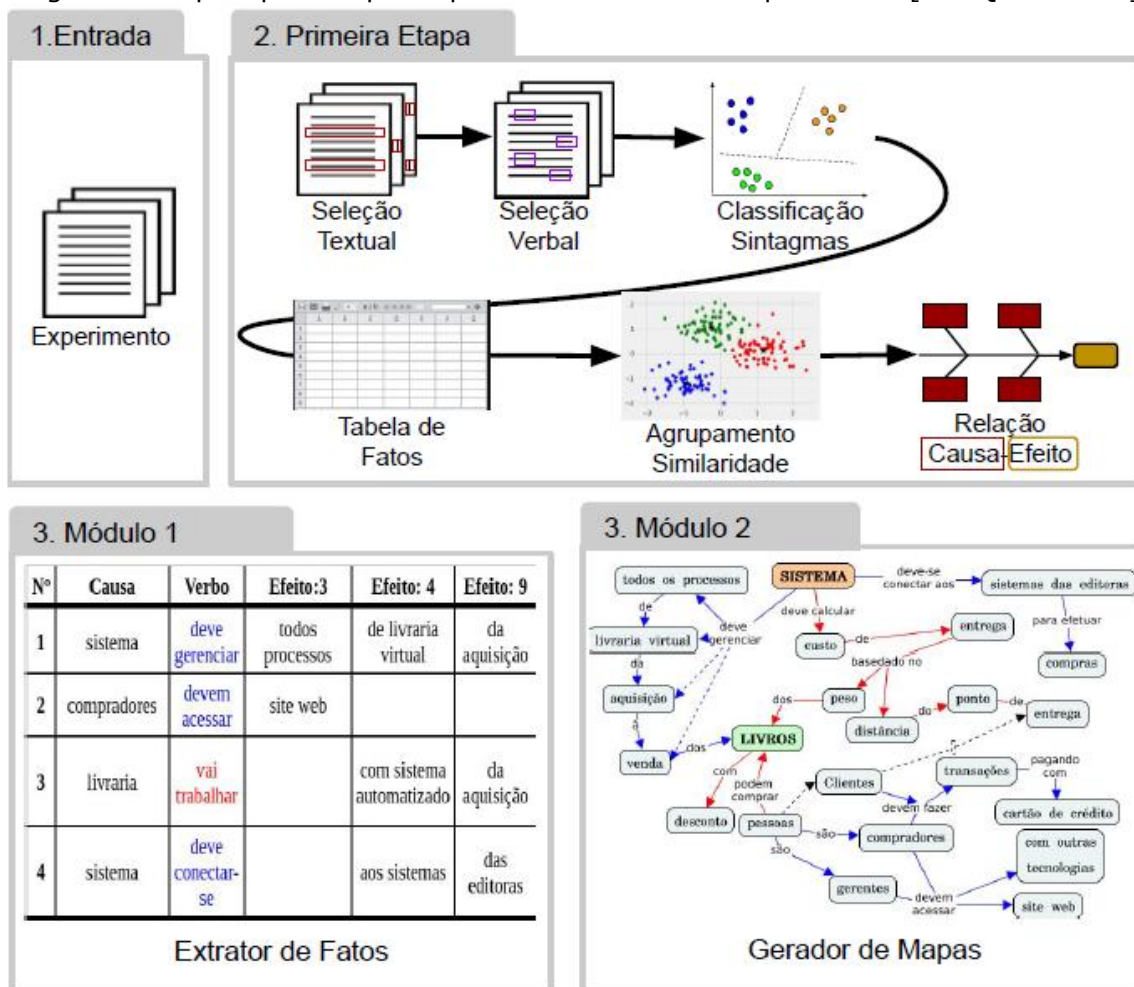








Figura 1. Etapas que compõe o processo do Verbka Inspirado em [VASQUES 2016]



Fonte: Autores

- Conversão de voz passiva em voz ativa:** caso existam frases escritas na voz passiva, essas são convertidas em sua forma ativa, a fim de revelar qual conceito é a causa e qual é o efeito da ação verbal. Considere a frase "O jogo foi visto por vários torcedores" que, após a aplicação das regras necessárias, se torna "Vários jogadores assistiram ao jogo". No caso de um sujeito indeterminado, insere-se o conceito genérico "pessoas". Tome como exemplo a seguinte frase: "O jogo foi visto ontem à noite" que, após a aplicação das regras, se tornou "Pessoas assistiram ao jogo ontem à noite".



O módulo "Gerador de mapas" é responsável por modelar informações na forma de grafos ou mapas conceituais a partir da tabela de fatos gerada. Esse modelo de visualização tem a capacidade de representar o conhecimento relacionado a algum conceito específico, gerando uma representação interativa e dinâmica do conhecimento. Nos mapas conceituais, cada termo é colocado dentro de um círculo ou caixa (vértices) e está relacionado a outros termos por meio de arcos (arestas) que representam as conexões (verbos ou preposições). Os conceitos conectados por um arco dão origem a um fato, que é a característica particular dos mapas conceituais [NOVAK and CANÃS 2010].

## **5. Aplicação e Resultados**

Com a aplicação da ferramenta Verbka sobre o texto do estudo de caso "Sistema Livir: Livraria Virtual" [WAZLAWICK 2004], foi gerada uma tabela de fatos (ou proposições) e um mapa conceitual. A partir da tabela de fatos, foi elaborado um diagrama de caso de uso e o diagrama de classes [SOMMERVILLE 2019]. Uma análise visual foi feita por meio do mapa conceitual gerado pela ferramenta para apoiar na identificação dos conceitos (objetos de negócio) e nas ligações definidas nos diagramas de caso de uso e de classes criados a partir dos requisitos. As etapas para o desenvolvimento deste experimento são descritas a seguir.

### **5.1 Processando o Texto de Entrada**

Para a execução da abordagem proposta, primeiramente foi selecionado um texto de entrada para o processamento pela ferramenta Verbka. Utilizou-se um texto sobre a "Visão Geral do Sistema" de uma livraria virtual fictícia, contida em um livro didático [WAZLAWICK 2004]. Segundo [WAZLAWICK 2004], esse tipo de documento tem o intuito de descrever



pontos relevantes referentes ao sistema que o cliente deseja que seja desenvolvido. No Quadro 1 é apresentado o texto utilizado para este experimento.

Quadro 1. Estudo de Caso do Sistema Livir

### **Sistema Livir: Livraria Virtual**

#### Visão Geral do Sistema

O sistema deve gerenciar todos os processos de uma livraria virtual, desde a aquisição até a venda dos livros. O acesso dos compradores e gerentes deve ser feito através de um site Web e possivelmente com outras tecnologias. Os compradores fazem as transações pagando com cartão de crédito.

Existem promoções eventuais pelas quais os livros podem ser comprados com desconto. De início, a livraria vai trabalhar apenas com livros novos a serem adquiridos de editoras que tenham sistema automatizado de aquisição. O sistema a ser desenvolvido deve conectar-se aos sistemas das editoras para efetuar as compras.

O sistema deve calcular o custo de entrega baseado no peso dos livros e na distância do ponto de entrega. Eventualmente pode haver promoções do tipo "entrega gratuita" para determinadas localidades. O sistema deve permitir a um gerente emitir relatórios de livros mais vendidos e de compradores mais assíduos, bem como sugerir compras para compradores baseadas em seus interesses anteriores. Quando um livro é pedido, se existe em estoque, é entregue imediatamente, senão o livro é solicitado ao fornecedor, e um prazo compatível é informado ao comprador.

Fonte: WAZLAWICK 2004

## 5.2 Gerando Tabela de Fatos e Mapa Conceitual

A partir do processamento do texto de entrada (Quadro 1), a ferramenta gerou a tabela de fatos (Tabela 2), que tem a função de especificar os componentes semânticos sobre quem realiza uma determinada ação (CAUSA) e quais os efeitos de cada uma dessas ações (EFEITO). A primeira coluna da tabela representa o conceito CAUSA, a segunda coluna o Verbo e as colunas subsequentes são referentes aos conceitos EFEITO.

Como exemplo de relacionamentos extraídos e reportados na Tabela 2, temos o fato de número 7 "pessoas podem comprar livros com desconto", no qual o conceito "pessoas" representa a CAUSA da ação, os verbos "podem

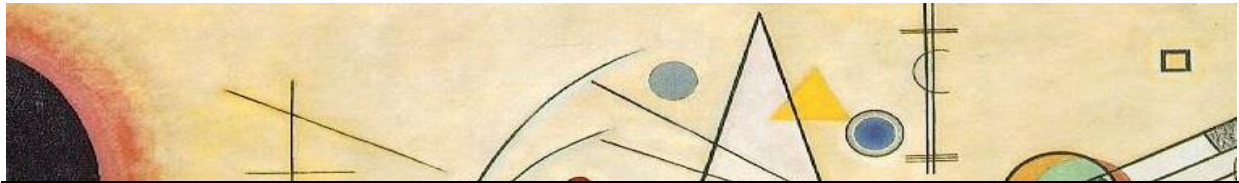


comprar”, representam a AÇÃO, e os conceitos “livros” e “com desconto”, por sua vez, representam os EFEITOS da ação. As cores dos verbos representam a sua tipologia semântica.

Os componentes apresentados nas colunas da Tabela 2 representam os diferentes conceitos que compõem cada um dos fatos (ou proposições), formando assim uma linha. Identificou-se que as linhas de número 02, 03, 04 e 05 representam requisitos não funcionais, pois são classificados como requisitos ambientais, segundo [SOMMERVILLE 2019]. A linha de número 09 também é considerada como um requisito não funcional, classificado como requisito de processo operacional ainda segundo [SOMMERVILLE 2019]. Portanto, essas linhas não serão consideradas para a construção dos diagramas de caso de uso e dos diagramas de classes, já que estes visam a modelagem apenas dos requisitos funcionais.

Tabela 2. Proto-Papeis

Nº	Causa	Verbo	Efeito:3	Efeito:4	Efeito:9	Efeito:10	Efeito:11	Efeito:12
01	sistema	deve gerenciar	todos processos	de livraria virtual	da aquisição	à venda	dos livros	
02	peessoas	devem acessar	compradores	por um site web				
03	peessoas	devem acessar	compradores	com outras tecnologias				
04	peessoas	devem acessar	gerentes	por um site web				
05	peessoas	devem acessar	gerentes	com outras tecnologias				
06	compradores	devem fazer	transações pagando	com cartão de crédito				
07	peessoas	podem comprar	livros	com desconto				
08	livraria	vai trabalhar		com sistema automatizado	de aquisição apenas	com livros novos	à ser adquiridos	de editoras
09	sistema	deve conectar-se		aos sistemas	das editoras	para pedidos para efetuar as compras		
10	sistema	deve calcular	custo	de entrega baseado	no peso	dos livros		



11	sistema	deve calcular	custo	de entrega baseado	na distância	do ponto	de entrega	
----	---------	------------------	-------	-----------------------	-----------------	----------	------------	--

Fonte: Autores

Em seguida, a partir da Tabela 2, foi gerado pela ferramenta o mapa conceitual causal (Figura 2) a partir da seleção dos fatos (das linhas) que o usuário selecionou para a modelagem. Esse mapa mostra os relacionamentos de causa-efeito entre todos os conceitos ali presentes, fazendo com que as diferentes causas e os diferentes efeitos se tornem fontes e receptores de diversas saídas e entradas resultantes dos fluxos de ações.

No mapa conceitual da Figura 2, observa-se a existência de três cores de setas [VASQUES 2016]: setas azuis são usadas para indicar um relacionamento de causa-efeito-causa, ou seja, uma ação parte de um conceito causa, que realiza uma ação e está ação, por sua vez, volta e atinge o mesmo conceito realizador da ação (por exemplo: "pessoas podem comprar livros", onde o verbo "poder" representa uma ação vinculada ao próprio conceito "pessoas", não significando que o livro foi comprado). As setas vermelhas são usadas para indicar um relacionamento de causa-efeito, no qual existe um conceito que realiza uma ação e um outro conceito que sofre essa ação (por exemplo: "livraria vai trabalhar com sistema automatizado", na qual "livraria" atua sobre "sistema automatizado" de modo efetivo). Por fim, as setas pretas representam a especificação ou qualidade de um conceito, não implicando em nenhuma afetação ou relação causa-efeito (por exemplo: "sistema é um sistema automatizado"). A análise das relações de causa e efeito entre os conceitos e o apoio desse entendimento é fundamental para a construção conceitual [VASQUES 2016].





sistema, que nesse contexto representa uma livraria virtual. Esse mapa conceitual ajuda ainda a entender os requisitos e a agrupar alguns conceitos (como: "livraria virtual"), o que viabiliza o aprendizado colaborativo [NÉSIO et al. 2020]. Nesse contexto, o mapa também é utilizado para a extração de conhecimento implícito e para ajudar a sanar dúvidas que, eventualmente, possam surgir. Exemplos de relacionamentos implícitos identificados no mapa são mostrados por meio das setas tracejadas, como: "sistema deve gerenciar aquisição", "pessoas são clientes", entre outros.

### 5.3 Modelando Diagramas de Caso de Uso e Diagramas de Classes

Para a elaboração do diagrama de Caso de Uso, mostrado na Figura 3, considerou-se a informação de que o sistema iria gerenciar os processos da livraria, assim como a livraria iria adquirir os livros, além disso, o sistema também deveria ser capaz de calcular os custos dos livros incluindo os valores do peso e a distância de entrega.

A Tabela 3 mostra um exemplo de uma lista com três dos principais requisitos funcionais extraídos. Cada conceito apresentado na tabela de fatos (Tabela 2) pode ser identificado como um requisito. Os casos de uso representam os requisitos funcionais baseados na descrição da visão de negócio, assim como as classes que representam os objetos de negócio (conceitos) no diagrama de classes. Esses requisitos, compõem elementos do modelo conceitual nos diagramas das Figuras 3 e 4.



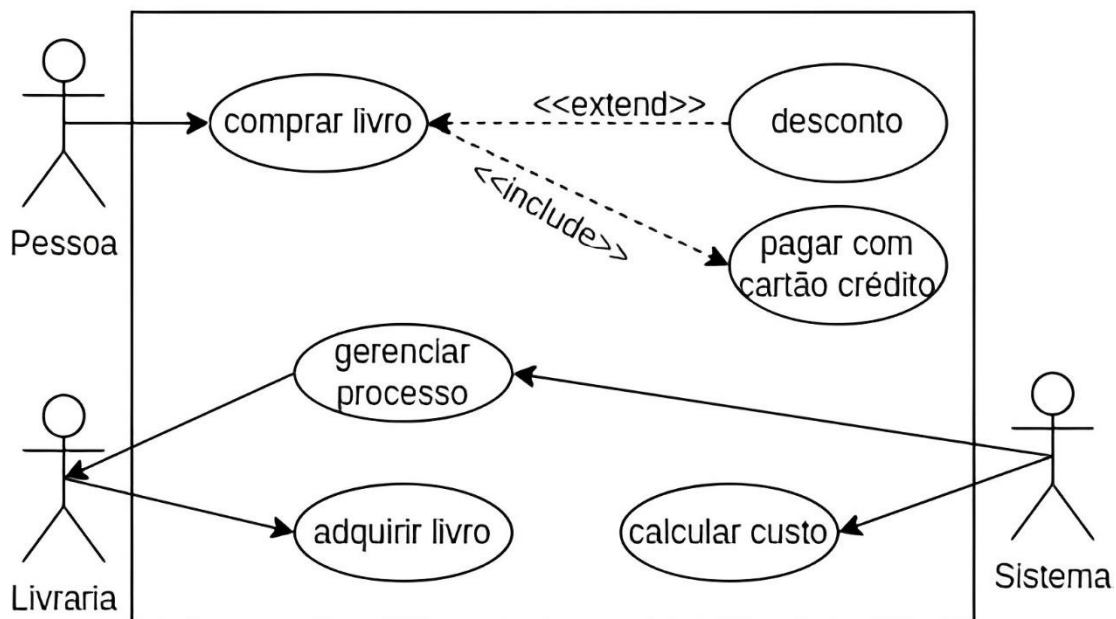
Tabela 3. Exemplo de lista de requisitos funcionais

Nº	Requisito	Descrição
1	Pessoas podem comprar livros	As pessoas são responsáveis pelas compras dos livros online.
2	Livraria vai trabalhar com aquisição de livros novos	A livraria é responsável pela compra de livros para serem vendidos online.
3	Sistema deve calcular custo	O sistema é responsável pelos cálculos dos custos para entrega do livro.

Fonte: Autores

No diagrama da Figura 3 também é considerado que uma pessoa seria capaz de comprar um livro e que, eventualmente, poderia receber um desconto e teria somente a opção de pagamento pelo cartão de crédito. A partir dessas informações e com base na Tabela 2, construiu-se o diagrama de caso de uso.

Figura 3. Diagrama de Caso de Uso referente ao Sistema de Livraria Virtual  
**Sistema Livir: Livraria Virtual**



Fonte: Autores

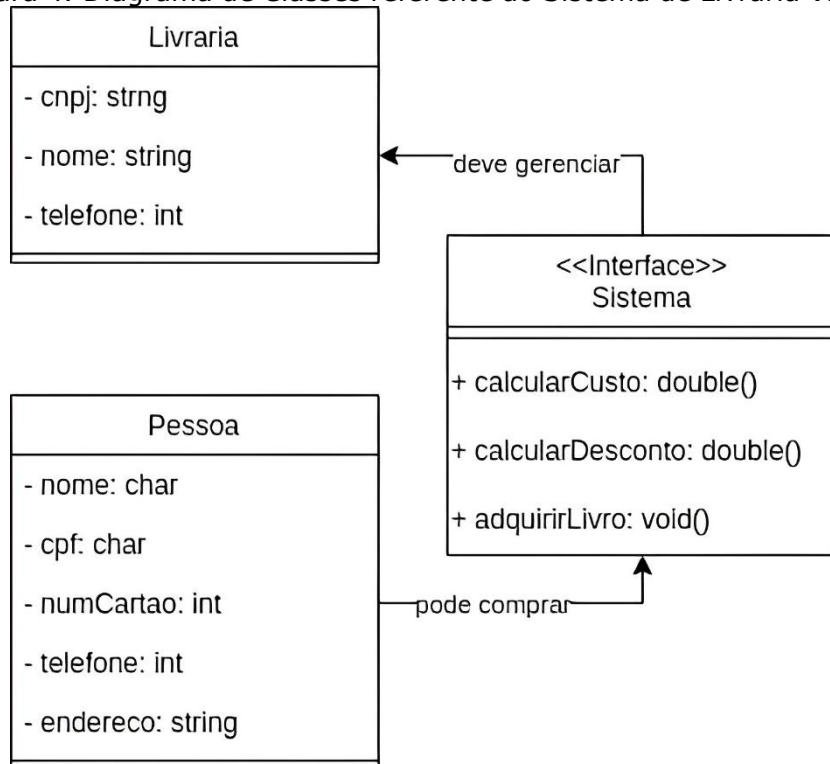
O diagrama de classes apresentado na Figura 4 é um diagrama no modelo de análise. Esse diagrama também foi elaborado com base na tabela de fatos (Tabela 2). Para a elaboração do diagrama da Figura 4, foi considerada a informação de que o sistema calcularia os custos e os





descontos, além de gerenciar a livraria. Considerou-se também que as pessoas poderiam comprar livros através do sistema usando apenas o cartão de crédito. Assim, os atores “Livraria” e “Pessoa” representados na Figura 3 passam a representar classes no sistema da Figura 4.

Figura 4. Diagrama de Classes referente ao Sistema de Livraria Virtual



Fonte: Autores

Pode-se analisar que, tanto no mapa conceitual apresentado na Figura 2, quanto nos diagramas apresentados nas Figuras 3 e 4, existe uma relação de causa-efeito entre os conceitos “sistema” e os “custos dos livros” e entre “sistema” e “gerenciamento dos processos da livraria”. É importante ressaltar que os diagramas das Figuras 3 e 4 representam uma versão inicial na pré- modelagem de software. Em um projeto de desenvolvimento de software, esses modelos devem ser analisados e refinados a cada etapa do desenvolvimento do projeto.



## **6. Conclusões**

Os resultados do experimento mostraram que o uso de um processo de Gestão do Conhecimento, especificamente a utilização de técnicas de extração do conhecimento e a sua visualização em tabelas estruturadas e mapas conceituais pode auxiliar no entendimento e na identificação dos casos de uso e das classes em um projeto de software, além de facilitar o levantamento dos requisitos. O uso do mapa conceitual colaborou com a análise qualitativa, ajudou no entendimento dos conceitos principais, dos seus relacionamentos explícitos e implícitos e das situações mais complexas, mostrando que também poderia ser utilizado como uma ferramenta colaborativa.

O uso da ferramenta de aquisição do conhecimento facilitou o domínio dos conceitos fundamentais, simplificando o desenvolvimento do processo de modelagem de diagramas UML. Assim como no trabalho de [VASQUES et al. 2019b], o uso do processo Verbka neste trabalho possibilitou a extração dos requisitos necessários para o desenvolvimento dos diagramas de forma simples e objetiva. Todavia, a diferença do trabalho de [VASQUES et al. 2019b], este experimento foi realizado com o auxílio da ferramenta que automatiza o processo, vale ressaltar que os requisitos não funcionais não entraram em análise e a não atenção aos requisitos não funcionais constitui uma das maiores causas de abandono de produtos de software.

Como trabalho futuro, pretende-se usar a ferramenta Verbka também para apoiar a especificação de requisitos não funcionais e a sua modelagem, já que esses requisitos estão ligados diretamente aos atributos de qualidade de software.





HANSEN, M. T. and BIRKINSHAW, J. (2007). The innovation value chain. *Harvard business review*, 85(6):121–130. <https://bityli.com/NUGMxaD>.

KUMAR, E. (2013). Natural language processing. IK International Pvt Ltd. <https://bityli.com/OeXCHC>.

LARMAN, C. (2007-2008). Utilizando UML e padrões: uma introdução a análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. Porto Alegre, RS: Bookman, 3 edition.

MORAIS, I. S. and ZANIN, A. (2020). Engenharia de Software. Porto Alegre: SAGAH, 1 edition.

MOYA, E. C. (2017). Using active methodologies: The students view. *Procedia - Social and Behavioral Sciences*, 237:672–677. <https://www.sciencedirect.com/science/article/pii/S187704281730040X>.

NADEKARNI, P. M., OHNO-MACHADO, L., and CHAPMAN, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551. <https://bityli.com/VnHqydBf>.

NE´SIO, R., FREITAS, P. F., and FREITAS, G. M. (2020). Laboratório de projeto concepção, gestão e implementação de projetos aplicados ao ensino de engenharia de sistemas. In Congresso Brasileiro de Automática-CBA, volume 2. [https://www.sba.org.br/open\\_journal\\_systems/index.php/cba/article/view/1600/1350](https://www.sba.org.br/open_journal_systems/index.php/cba/article/view/1600/1350).

NOVAK, J. D. and CAN˜AS, A. J. (2010). A teoria subjacente aos mapas conceituais e como elaborá-los e usá-los. *Práxis educativa*, 5(1):9–29. <https://dialnet.unirioja.es/servlet/articulo?codigo=3251296>.

PAULA, L. S. d. and TATTO, L. (2020). Diagnostico de práticas e ferramentas de gestão de conhecimento na Industria de software: Estudo de caso múltiplo. *Revista Científica Multidisciplinar Núcleo do Conhecimento*, 03:69–90. <https://www.nucleodoconhecimento.com.br/administracao/industria-de-software>.

PRESSMAN, Roger S.; MAXIM, B. R. (2021). Engenharia de Software: uma abordagem profissional. Porto Alegre: AMGH, 9 edition.

PRISMA (2009). Relatório transparente prisma. <http://www.prisma-statement.org/>. REINEHR, S. (2020). Engenharia de Requisitos. Grupo A, 1 edition.



SANTOS, G. S. d. (2021). Um processo de representação estruturada e validação de requisitos de software para mitigar problemas semânticos. In UNIVERSIDADE ESTADUAL DE CAMPINAS - Faculdade de Tecnologia, page 140. UNICAMP.

SOMMERVILLE, I. (2019). Software engineering. Editora Pearson, 10 edition.

SOUZA, L., MIRANDA, E., LUCENA, M., and GOMES, A. (2019). Desafios e práticas da engenharia de requisitos no contexto de fábrica de software com foco na documentação e gestão do conhecimento. Cadernos do IME-Série Informática, 42:98. <https://www.e-publicacoes.uerj.br/index.php/cadinf/article/view/47530/32365>.

VASQUES, D. (2016). Dissertação de mestrado. verbka: processo para aquisição de conhecimento baseado em semântica verbal. page 168. [https://www.researchgate.net/publication/309311907\\_Dissertacao\\_de\\_Mestrado\\_Verbka\\_processo\\_para\\_aquisicao\\_de\\_conhecimento\\_baseado\\_em\\_semantica\\_verbal](https://www.researchgate.net/publication/309311907_Dissertacao_de_Mestrado_Verbka_processo_para_aquisicao_de_conhecimento_baseado_em_semantica_verbal).

VASQUES, D. G. (2021). Uma abordagem híbrida para detecção de relacionamentos causais aplicados à descoberta baseada em literatura. PhD thesis. [https://www.teses.usp.br/teses/disponiveis/55/55134/tde-20042022-141047/publico/DildreVasques\\_revisada.pdf](https://www.teses.usp.br/teses/disponiveis/55/55134/tde-20042022-141047/publico/DildreVasques_revisada.pdf).

VASQUES, D. G., GALINDO, J. F., dos SANTOS, G. S., GOMES, F. D., GARGIA-NUNES, P. I., and MARTINS, P. S. (2019a). An educational process for requirements extraction and use case modeling based on problem-based learning and knowledge acquisition. In Brazilian Symposium on Information Systems, pages 1–8. <https://dl.acm.org/doi/abs/10.1145/3330204.3330276>.

VASQUES, D. G., SANTOS, G. S., GOMES, F. D., GALINDO, J. F., and MARTINS, P. S. (2019b). Use case extraction through knowledge acquisition. In Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pages 0624–0631. IEEE. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8936279>.

VEMURI, S., CHALA, S., and FATHI, M. (2017). Automated use case diagram generation from textual user requirement documents. In Canadian Conference on Electrical and Computer Engineering (CCECE), pages 1–4. IEEE. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7946792>.

WAZLAWICK, R. S. (2004). Análise e projeto de sistemas de informação orientados a objetos., volume 1. Rio de Janeiro: Campus.