

# **Framework para Verificação e Correção Automatizada da Conformidade Sintática de Requisitos de Software baseada em Modelos com Uso de IA**

**Arthur H. M. de Oliveira, Johnny Marques**

Instituto Tecnológico de Aeronáutica (ITA)  
Pós-Graduação em Ciências e Tecnologias Espaciais – Doutorado  
São José dos Campos – SP – Brazil

Doutorado - Início: Mar/2025 - Término: Out/2028 - Qualificação: Set/2026

hendricks@ita.br, johnny@ita.br

**Abstract.** This article presents the progress of a doctoral research project focused on developing a framework for the automated verification and correction of requirement syntax. The goal is to ensure compliance with Requirement Models widely adopted in the industry, improving the accuracy of specifications in software development processes. The methodology applies Natural Language Processing (NLP) techniques and Large Language Models (LLMs) to automate a process previously dependent on manual intervention. A Systematic Literature Mapping (SLM) identified existing approaches, and the research is currently exploring relevant NLP techniques and tools for the framework.

**Resumo.** Este artigo apresenta o progresso de uma pesquisa de doutorado voltada ao desenvolvimento de um framework para verificação e correção automatizadas da sintaxe de requisitos. O objetivo é assegurar a conformidade com Modelos de Requisitos amplamente adotados pela indústria, aprimorando a precisão das especificações em processos de desenvolvimento de software. A metodologia aplica técnicas de Processamento de Linguagem Natural (PLN) e Modelos de Linguagem de Grande Porte (LLMs) para automatizar um processo antes dependente de intervenção manual. Um Mapeamento Sistemático da Literatura (MSL) identificou abordagens existentes, e a pesquisa atualmente explora técnicas e ferramentas de PLN relevantes ao framework. Link: <https://youtu.be/amL-WAbUt6U>.

## **1. Contexto e Motivação da Pesquisa**

A transformação digital tem impulsionado a necessidade de agilidade no desenvolvimento de *software*, tornando a Engenharia de Requisitos (ER) uma disciplina crucial para a qualidade dos sistemas [Almendra et al. 2022]. Por se tratar de um processo intensivo em conhecimento, os processos de ER partem de dados dispersos em múltiplas fontes – como as partes interessadas, regulamentações, produtos já existentes e a experiência da equipe, entre outros, e resultam em especificações estruturadas e decisões identificadas, compreendidas e negociadas entre as partes interessadas [Marques and da Cunha 2019].

Essas atividades frequentemente demandam intervenção humana, sendo a linguagem natural a notação predominante para expressar requisitos na prática industrial

[Zhao et al. 2021]. Contudo, requisitos formulados em linguagem natural irrestrita são intrinsecamente imprecisos e suscetíveis a ambiguidades [Arora et al. 2015]. Essa inerente imprecisão e o potencial para múltiplas interpretações podem gerar problemas significativos que comprometem o desenvolvimento e a qualidade dos sistemas de *software*, caso não sejam endereçados em fases iniciais [Balwani et al. 2024].

Em resposta a esses desafios, Modelos de Requisitos têm sido amplamente empregados na indústria como uma estratégia para estruturar requisitos e mitigar a ambiguidade [Balwani et al. 2024]. Atrelado a isso, técnicas e ferramentas de Processamento de Linguagem Natural (PLN) e, mais recentemente, os Grandes Modelos de Linguagem (LLMs), subcampos da Inteligência Artificial (IA), têm oferecido soluções promissoras para a detecção e mitigação de ambiguidades em requisitos de maneira automatizada [Arora et al. 2015] [Arora et al. 2024]. Assim, o problema central endereçado nesta pesquisa é: **“Como as técnicas e ferramentas de Processamento de Linguagem Natural (PLN) e os Grandes Modelos de Linguagem (LLMs) podem ser integrados para desenvolver um *framework* automatizado eficiente na verificação e correção da conformidade sintática dos requisitos face aos Modelos de Requisitos?”**

Dessa forma, o objetivo desta pesquisa visa realizar uma análise abrangente dos modelos de requisitos discutidos na literatura existente, identificar e catalogar *frameworks* relacionados, explorar e determinar as melhores técnicas e ferramentas de Processamento de Linguagem Natural (NLP) para o *framework* proposto, explorar e determinar os Grandes Modelos de Linguagem (LLMs) e suas configurações ideais para os melhores desempenhos do *framework* proposto, e executar casos de estudo em sistemas reais dos setores de Defesa e Aeroespacial.

## 2. Fundamentação Teórica

A fim de estabelecer uma base conceitual eficaz para a discussão proposta nesta pesquisa, são apresentadas as definições de termos cruciais.

### 2.1. Modelos de Requisitos

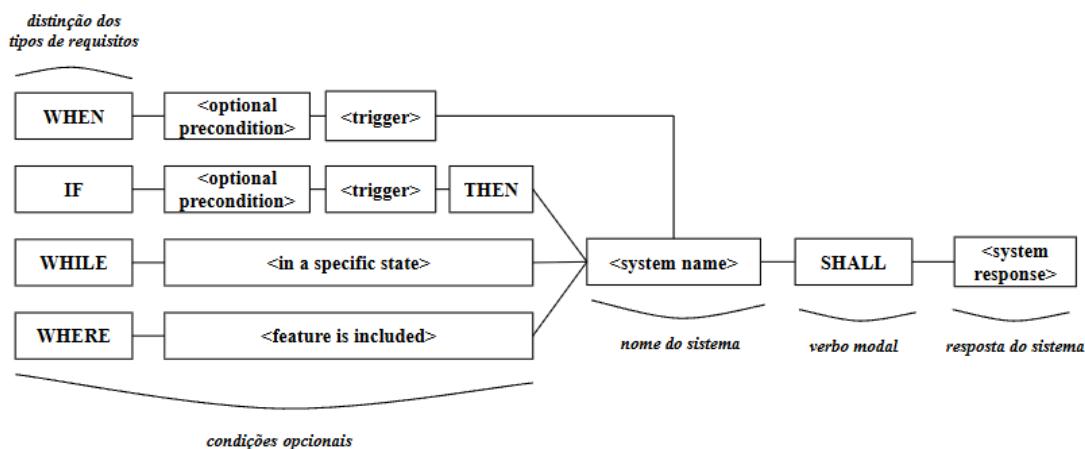
Modelos de Requisitos são ferramentas amplamente empregadas na indústria para auxiliar na estruturação da sintaxe de requisitos [Balwani et al. 2024]. Esses modelos geralmente utilizam verbos e palavras-chave para designar as funcionalidades e ações do sistema em espaços predefinidos, visando aumentar a clareza e a precisão na captura desses elementos. Inicialmente, os modelos RUPP [Pohl and Rupp 2021] e EARS [Mavin et al. 2009] foram incluídos no escopo desta pesquisa devido à sua relevância e reconhecimento na indústria e literatura.

#### 2.1.1. Easy Approach to Requirements Syntax (EARS)

A Figura 1 apresenta o modelo *Easy Approach to Requirements Syntax* (EARS) [Mavin et al. 2009], que é composto por quatro partes: condições opcionais no início, o nome do sistema, um verbo modal e a resposta do sistema representando o comportamento do sistema, podendo distinguir até cinco tipos de requisitos:

- **Requisitos Ubíquos:** não têm pré-condição e estão sempre ativos.

- **Requisitos Orientados a Eventos:** começam com a palavra-chave *WHEN* e indicam um evento de gatilho.
- **Requisitos de Comportamento Indesejado:** começam com a palavra-chave *IF*, seguido por *THEN*, antes do nome do sistema, e expressam situações indesejáveis.
- **Requisitos Orientados a Estados:** começam com a palavra-chave *WHILE* e estão ativos em um estado definido.
- **Requisitos de recursos opcionais:** começam com a palavra-chave *WHERE* e precisam ser atendidos quando certos recursos opcionais estão presentes.
- **Requisitos complexos:** usam uma combinação de mais de um tipo de condição.



**Figura 1. Easy Approach to Requirements Syntax (EARS).**

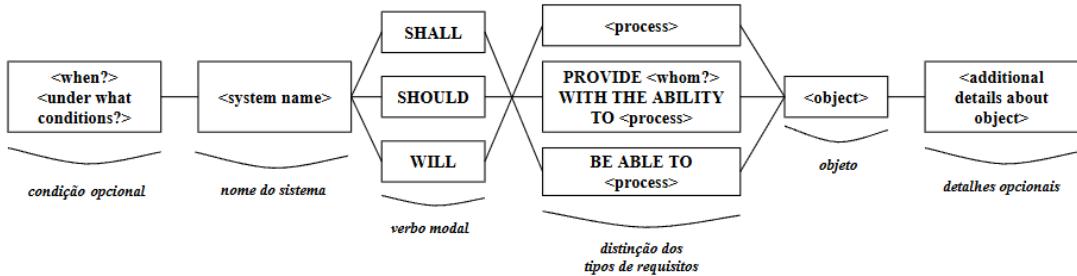
## 2.2. Rupp

A Figura 2 apresenta o modelo Rupp [Pohl and Rupp 2021], que é composto por até seis características distintas: uma condição opcional, o nome do sistema, um verbo modal, a funcionalidade de processamento necessária, o objeto para o qual a funcionalidade é necessária e detalhes opcionais, podendo distinguir até três tipos de requisitos:

- **Requisitos Autônomos:** descrevem a funcionalidade que o sistema executa independentemente da interação do usuário. Utilizam o formato *<process>*.
- **Requisitos de Interação com o Usuário:** referem-se à funcionalidade que o sistema oferece a usuários específicos. O formato é “*PROVIDE <whom?> WITH THE ABILITY TO <process>*”.
- **Requisitos de Interface:** definem a funcionalidade que o sistema realiza em resposta a eventos acionados por outros sistemas. O formato é “*BE ABLE TO <process>*”.

## 2.3. Processamento de Linguagem Natural para Engenharia de Requisitos

A aplicação do PLN na Engenharia de Requisitos, ou o termo do inglês *Natural Language Processing for Requirements Engineering* (NLP4RE), é uma área de pesquisa que busca aplicar tecnologias (técnicas, ferramentas e recursos) de PLN ao processo de ER para auxiliar analistas humanos em diversas tarefas de análise linguística em documentos de requisitos textuais [Zhao et al. 2021]. Esses três tipos de tecnologias se diferenciam, podendo ser amplamente categorizados em:



**Figura 2. Modelo Rupp.**

- Técnicas de Processamento de Linguagem Natural: métodos, processos ou procedimentos para executar uma tarefa particular de PNL.
- Ferramentas de Processamento de Linguagem Natural: sistemas de *software* ou bibliotecas que oferecem suporte a uma ou mais técnicas de PNL.
- Recursos de Processamento de Linguagem Natural: dados linguísticos que dão suporte a técnicas ou ferramentas de PNL, que pode ser um léxico de linguagem ou um *corpus* (coleção de textos).

#### 2.4. Modelos de Linguagem de Grande Porte para Engenharia de Requisitos

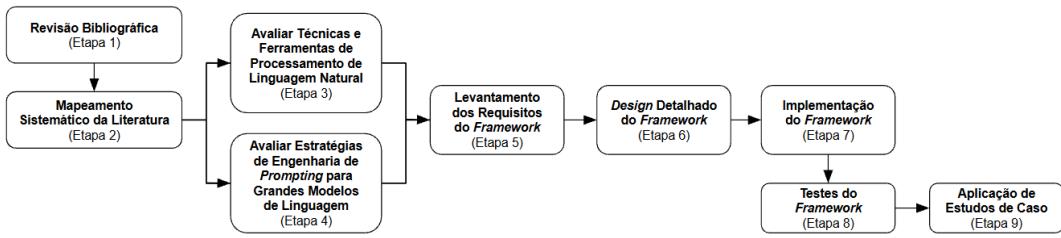
Modelos de Linguagem de Grande Porte, popularmente conhecidos pelo termo do inglês *Large Language Models* (LLMs), surgiram como ferramentas poderosas de PLN e têm sido amplamente aplicados na ER [Arora et al. 2024]. Diversas estratégias de Engenharia de *Prompting* têm sido aplicadas na literatura:

- *Zero-shot learning*: Abordagem onde o LLM executa uma atividade sem exemplos explícitos no *prompt*, baseando-se apenas em seu conhecimento pré-existente.
- *Few-shot learning*: Envolve o fornecimento de alguns exemplos relacionados à atividade no *prompt* do LLM.
- *One-shot learning*: Uma variação do *few-shot learning* na qual apenas um único exemplo é fornecido ao LLM para a execução de uma tarefa.
- *Chain-of-thought prompting*: Adiciona várias etapas de raciocínio ao fornecer exemplos para um LLM, permitindo que o modelo gere o raciocínio junto com a saída relacionada.

### 3. Métodos de Pesquisa

A pesquisa será conduzida em 9 etapas, conforme mostra o Fluxograma da Figura 3.

A Etapa 1 consiste na Revisão Bibliográfica abrangente da Literatura de Modelos de Requisitos, com o objetivo de estabelecer uma compreensão sólida dos modelos de requisitos existentes. Na Etapa 2 será realizado um Mapeamento Sistemático da Literatura (MSL), visando avaliar a pesquisa relevante sobre modelos de requisitos e técnicas de verificação e correção automatizada. Em paralelo, na Etapa 3 será realizado um estudo para avaliar técnicas e ferramentas de Processamento de Linguagem Natural (PLN), visando identificar as que apresentarem maior precisão no processo de verificação de requisitos. De forma complementar, a Etapa 4 incluirá um estudo para avaliar estratégias



**Figura 3. Fluxograma.**

de Engenharia de *Prompting* para Grandes Modelos de Linguagem (LLMs), explorando como otimizar as interações com LLMs para corrigir a sintaxe dos requisitos.

Com base nos conhecimentos adquiridos, será realizado nas Etapas 5, 6 e 7 o levantamento dos requisitos, o *design* detalhado e a implementação do *framework* proposto. Posteriormente, serão realizados testes com o *framework* na Etapa 8 e será realizada a aplicação de estudos de caso dos setores de Defesa e Aeroespacial na Etapa 9.

#### 4. Resultados Preliminares

Após a conclusão da Revisão Bibliográfica sobre Modelos de Requisitos, foi realizado um Mapeamento Sistemático da Literatura (MSL), seguindo as diretrizes estabelecidas por Petersen et al. (2015), seguindo o Fluxograma da Figura 4.

**Questões de Pesquisa do MSL** Foram levantadas 4 Questões de Pesquisas no MSL

**QP1.** Quais ferramentas que aplicam automação nos processos de verificação e/ou correção de requisitos, visando sua conformidade sintática com modelos de requisitos, são descritas na literatura?

**QP2.** Quais tipos de modelos de requisitos são utilizados pelas ferramentas?

**QP3.** Quais as técnicas e ferramentas de processamento de linguagem natural são utilizadas para a verificação da sintaxe de requisitos?

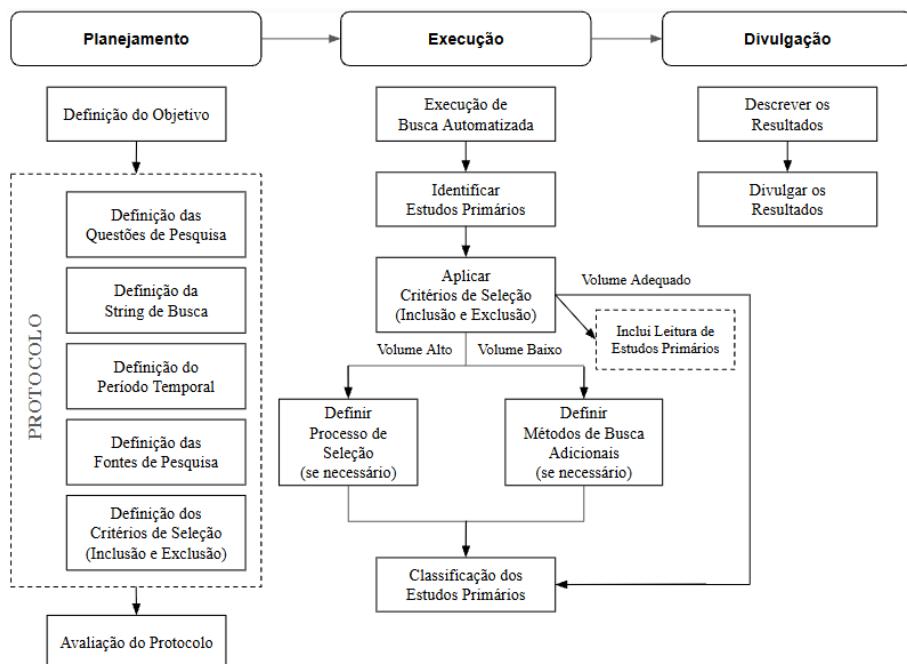
**QP4.** Existem ferramentas que aplicam grandes modelos de linguagem e técnicas de engenharia de *prompting* para a correção da sintaxe de requisitos?

**Período Temporal e Fontes de Pesquisa do MSL** A faixa temporal utilizada no MSL foi de 1 de Janeiro de 2008 até 14 de Junho de 2025, com foco em duas fontes de pesquisa:

- IEEE Xplore Digital Library (Xplore)
- ACM Digital Library (ACM)

**String de Busca do MSL** A Tabela 1 apresenta a *string* de busca utilizada no MSL

Os trabalhos obtidos na extração automática foram analisados e selecionados com base nos Critérios de Inclusão (CI) e Critérios de Exclusão (CE) definidos pelos autores. A Tabela 2 apresenta uma síntese dos resultados numéricos após execução do MSL.



**Figura 4.** Etapas do Mapeamento Sistemático da Literatura.

**Tabela 1.** String de Busca do Mapeamento Sistemático da Literatura.

Título	Conectivo	Resumo	Conectivo	Resumo
Requirements templates		Requirements templates		Natural language processing
Requirements boilerplates		RTs		NLP
Requirements checking	AND	Requirements boilerplates		Large language models
Requirements syntax		Requirements syntax		LLMs
Requirements formalism				Artificial Intelligence
Requirements formalisation				Machine Learning
				Automation
				Automatically

**Trabalhos Relacionados** A Tabela 3 apresenta um sumário com os trabalhos relacionados encontrados após a execução do MSL e leitura abrangente dos artigos, incluindo ferramentas propostas e modelos de requisitos abordados. Foram encontrados 5 trabalhos relacionados, 5 ferramentas propostas e 2 modelos de requisitos utilizados.

**Tabela 2. Resultados da Execução do Mapeamento Sistemático da Literatura.**

Bases de Dados	Extração Automática	Após Aplicação dos Critérios
IEEE Xplore	17	7
ACM Digital Library	8	2
<b>Total</b>	<b>25</b>	<b>9</b>

[Arora et al. 2013] apresenta RUBRIC, uma ferramenta que verifica a conformidade sintática de requisitos com o modelo Rupp através de técnicas de PLN. [Arora et al. 2015] apresenta RETA, uma evolução do RUBRIC, que contempla os modelos Rupp e EARS. [Lúcio et al. 2017] apresenta EARS-CTRL, que verifica a conformidade sintática de requisitos com base apenas no modelo EARS. [Tiwari et al. 2022] apresenta NL2RT, que traduz texto em linguagem natural para requisitos nos formatos EARS e RUPP, utilizando a biblioteca spaCy. [Balwani et al. 2024] apresenta AutoReco, uma ferramenta que, além de verificar a conformidade de requisitos com os modelos EARS e RUPP, fornece recomendações com versões corrigidas.

**Tabela 3. Trabalhos Relacionados.**

Trabalho	Ferramenta	Modelos de Requisitos
[Arora et al. 2013]	RUBRIC	Rupp
[Arora et al. 2015]	RETA	EARS e Rupp
[Lúcio et al. 2017]	EARS-CTRL	EARS
[Tiwari et al. 2022]	NL2RT	EARS e Rupp
[Balwani et al. 2024]	AutoReco	EARS e Rupp
<b>Total</b>	<b>5</b>	<b>2</b>

**Avaliação de Ferramentas de PLN** Experiments iniciais estão sendo conduzidos para avaliar ferramentas de PLN com bibliotecas em Python – Stanza, NLTK e SpaCy, considerando o modelo Rupp. Inicialmente, uma base de dados com 977 requisitos de *software* da plataforma Kaggle (<https://www.kaggle.com/datasets/iamvaibhav100/software-requirements-dataset>) está sendo utilizada nos experimentos, dos quais 521 são requisitos funcionais. Os resultados iniciais apontam para uma vantagem da biblioteca NLTK em métricas de acurácia, precisão e tempo de execução, conforme mostra a Tabela 4.

**Tabela 4. Resultados Iniciais para Avaliação de Ferramentas de PLN.**

Métrica	Stanza	NLTK	SpaCy
Acurácia	0.8791	0.8887	0.8733
Precisão	0.7879	0.8014	0.7800
Tempo de Execução (s)	629.8088	0.5129	5.7852

## 5. Considerações Finais e Próximos Passos

As etapas dos Métodos de Pesquisa concluídas incluem a Revisão Bibliográfica abrangente da Literatura de Modelos de Requisitos e o Mapeamento Sistemático da Literatura (MSL). Atualmente a pesquisa está na Etapa 3 que consiste na avaliação de técnicas e ferramentas de Processamento de Linguagem Natural (PLN) considerando bibliotecas em

Python – Stanza, NLTK e SpaCy, e será seguida pela Etapa 4 de avaliação de Grandes Modelos de Linguagem (LLMs).

## Referências

- Almendra, C., Silva, C., Martins, L. E. G., and Marques, J. (2022). How assurance case development and requirements engineering interplay: a study with practitioners. *Requirements Engineering*, 27(2):273–292.
- Arora, C., Grundy, J., and Abdelrazek, M. (2024). Advancing requirements engineering through generative ai: Assessing the role of llms. In *Generative AI for Effective Software Development*, pages 129–148. Springer.
- Arora, C., Sabetzadeh, M., Briand, L., and Zimmer, F. (2015). Automated checking of conformance to requirements templates using natural language processing. *IEEE transactions on Software Engineering*, 41(10):944–968.
- Arora, C., Sabetzadeh, M., Briand, L., Zimmer, F., and Gnaga, R. (2013). Automatic checking of conformance to requirement boilerplates via text chunking: An industrial case study. In *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 35–44. IEEE.
- Balwani, S., Tiwari, S., Dasgupta, S., and Sharma, A. (2024). An approach for providing recommendation for requirements non-conformant with requirement templates (rts). In *Proceedings of the 17th Innovations in Software Engineering Conference*, pages 1–11.
- Lúcio, L., Rahman, S., Cheng, C.-H., and Mavin, A. (2017). Just formal enough? automated analysis of ears requirements. In *NASA Formal Methods Symposium*, pages 427–434. Springer.
- Marques, J. and da Cunha, A. M. (2019). A set of requirements for certification of air-borne military software. In *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, pages 1–7. IEEE.
- Mavin, A., Wilkinson, P., Harwood, A., and Novak, M. (2009). Easy approach to requirements syntax (ears). In *2009 17th IEEE international requirements engineering conference*, pages 317–322. IEEE.
- Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and software technology*, 64:1–18.
- Pohl, K. and Rupp, C. (2021). *Basiswissen requirements engineering: Aus-und Weiterbildung nach IREB-Standard zum certified professional for requirements engineering foundation level*. dpunkt. verlag.
- Tiwari, S., Shah, P., and Khare, M. (2022). Nl2rt: A tool to translate natural language text into requirements templates (rts). In *2022 IEEE 30th International Requirements Engineering Conference (RE)*, pages 262–263. IEEE.
- Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K. J., Ajagbe, M. A., Chioasca, E.-V., and Batista-Navarro, R. T. (2021). Natural language processing for requirements engineering: A systematic mapping study. *ACM Computing Surveys (CSUR)*, 54(3):1–41.