

**UNIVERSIDADE DE SÃO PAULO**

Instituto de Ciências Matemáticas e de Computação

## **Análise de Causa Raiz de Chamados de Suporte de Sistemas com aplicação de LLM**

**Fernando dos Santos Ramos**

Monografia - MBA em Inteligência Artificial e Big Data



SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: \_\_\_\_\_

**Fernando dos Santos Ramos**

## **Análise de Causa Raiz de Chamados de Suporte de Sistemas com aplicação de LLM**

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Ricardo Marcondes Marcacini

**Versão original**

**São Carlos**

**2024**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi  
e Seção Técnica de Informática, ICMC/USP,  
com os dados inseridos pelo(a) autor(a)

R175a      Ramos, Fernando dos Santos  
            Análise de Causa Raiz de Chamados de Suporte de  
Sistemas com aplicação de LLM / Fernando dos Santos  
Ramos; orientador Ricardo Marcondes Marcacini. --  
São Carlos, 2024.  
            114 p.

            Trabalho de conclusão de curso (MBA em  
Inteligência Artificial e Big Data) -- Instituto de  
Ciências Matemáticas e de Computação, Universidade  
de São Paulo, 2024.

            1. . I. Marcacini, Ricardo Marcondes, orient.  
II. Título.

**Fernando dos Santos Ramos**

# **Root Cause Analysis of Systems Support Calls with LLM application**

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Prof. Dr. Ricardo Marcondes Marcacini

**Original version**

**São Carlos**

**2024**



*Este trabalho é dedicado a minha família, Mirlene, Lucas e Tiago, por me suportarem todo esse tempo. E a Profa Dr. Solange Rezende, que, num encontro casual, me incentivou a embarcar numa jornada nova, que me revitalizou, despertou meu interesse e paixão e foi mui útil na minha etapa da vida.*





## **AGRADECIMENTOS**

Quero agradecer ao Prof. Dr. Ricardo Marcondes Marcacini, que sempre foi gentil e presente, direcionando o trabalho de forma tão preciosa. Minha gratidão ao Cláudio de Almeida Prado, que, com sua visão sempre inovadora, gentilmente e de forma altruísta, permitiu que o trabalho contasse com uma excelente base de dados. Agradeço também o Tiago Júlio da Silva, que ajudou a dar os parâmetros de negócio para que o trabalho tivesse consistência e direcionamento. E ao Kleber Fonseca e Alyson Feitas, que foram fundamentais para que os dados estivesse prontos e corretos.



## RESUMO

RAMOS, F. S. **Análise de Causa Raiz de Chamados de Suporte de Sistemas com aplicação de LLM**. 2024. 114p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

A sustentação de sistemas em produção é frequentemente feita através de um processo de investigação do problema, o qual é executado por pessoal de suporte especializado. Não é incomum que, uma vez sanado o problema em atendimento em primeiro nível, as descrições que caracterizaram o problema não sejam objeto de análise a posteriori. A investigação é registrada em textos não-estruturados, o que dificulta estudos mais massificados e sistematizados da causa-raiz. Com o advento dos Modelos de Linguagem de Grande Escala (*Large Language Models*, ou LLM) abre-se a possibilidade de fazer uma análise investigativa mais massificada. O trabalho aplicou LLM e estratégias de *prompting* para a entender e classificar a causa-raiz dos problemas, sobre uma base real de chamados de sustentação de sistemas do setor financeiro, com mais de quarenta sistemas em quatro linhas de negócio: bancos, fundos, previdência e consórcio; desde grandes bancos até pequenos gestores de fundos. A base utilizada é um subconjunto dos dados de produção, atualizados até meados de 2023. Os dados contém o número do chamado, sistema, negócio, descrição e comentários. Para classificar erros na operação, propõe-se uma taxonomia com três níveis: grandes grupos de problemas (tecnologia, ambiente, fatores humanos, segurança), subcategorias (software, hardware, dados, infraestrutura) e tipos de falhas específicas para cada uma das subcategorias. A abordagem aplicada foi uma classificação fracamente supervisionada onde os Modelos de Linguagem de Grande Escala foram usados para classificar um sub-conjunto de chamados e um modelo BERT foi treinado com as classificações realizadas pelo LLM. O estudo investigou a aplicação de LLMs como GPT-3.5 e GPT-4, na classificação automatizada de chamados de suporte técnico. O trabalho demonstrou o potencial desses modelos em identificar padrões, compreender o contexto e classificar chamados com alta precisão. O GPT-4, em particular, mostrou superioridade ao GPT-3.5, exibindo uma compreensão mais profunda da linguagem natural e capacidade de inferência. Além disso, a pesquisa explorou a utilização de técnicas como *Bag-of-Words* e agrupamento para analisar as explicações geradas pelos modelos, revelando-se uma abordagem consistente para o processo de análise. O estudo também avaliou a eficácia do modelo BERT, que, após *fine-tuning* demonstrou bom desempenho, mas também apresentou desafios de *overfitting*. Por fim, combinação de LLMs e modelos pré-treinados, como o BERT, pode ser uma abordagem promissora para a classificação automatizada de chamados. No entanto, são necessárias pesquisas adicionais para explorar diferentes arquiteturas, técnicas de pré-treinamento e aumento de dados, visando melhorar a precisão e a generalização dos modelos.

**Palavras-chave:** Classificação de Textos; LLM; Sustentação Sistemas; Setor Financeiro; Bancos



## ABSTRACT

RAMOS, F. S. **Root Cause Analysis of Systems Support Calls with LLM application** . 2024. 114p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

Production system support often relies on an investigative process conducted by specialized support personnel. It's common for problem descriptions, once resolved in first-level support, to not be subject to subsequent analysis. Investigations are recorded in unstructured text, hindering more comprehensive and systematic root cause analyses. The advent of Large Language Models (LLMs) offers the potential for more extensive investigative analysis. This study aimed to apply LLMs and prompting strategies to understand and classify the root causes of problems. A real-world dataset of support tickets from the financial sector was used, encompassing over forty systems from four lines of business: banking, funds, pensions, and consortiums. The dataset, updated until mid-2023, included information such as ticket number, system, business unit, description, and comments. To classify operational errors, a three-level taxonomy was proposed: broad problem groups (technology, environment, human factors, security), subcategories (software, hardware, data, infrastructure), and specific failure types for each subcategory. A weakly supervised classification approach was employed, where LLMs were used to classify a subset of tickets, and a BERT model was trained on the LLM-generated classifications. The study investigated the application of LLMs like GPT-3.5 and GPT-4 in the automated classification of technical support tickets. The work demonstrated the potential of these models to identify patterns, understand context, and classify tickets with high accuracy. GPT-4, in particular, outperformed GPT-3.5, exhibiting a deeper understanding of natural language and inference capabilities. Additionally, the research explored the use of techniques like Bag-of-Words and clustering to analyze the explanations generated by the models, proving to be a consistent approach to the analysis process. The study also evaluated the effectiveness of the BERT model, which, after fine-tuning, showed good performance but also exhibited overfitting challenges. Finally, combining LLMs and pre-trained models like BERT can be a promising approach for automated ticket classification. However, further research is needed to explore different architectures, pre-training techniques, and data augmentation to improve model accuracy and generalization.

**Keywords:** Text Classification; LLM; Support Systems; Financial Sector; Bank



## LISTA DE FIGURAS

Figura 1 – Tipologia de <i>Prompting</i> - Adaptado de (LIU <i>et al.</i> , 2023a) . . . . .	31
Figura 2 – Abordagem de Execução . . . . .	34
Figura 3 – Extração e Limpeza dos Dados . . . . .	34
Figura 4 – Classificação LLM na taxonomia . . . . .	35
Figura 5 – Classificação Fracamente Supervisionada . . . . .	36
Figura 6 – Análises Visuais e Comparativas . . . . .	36
Figura 7 – Distribuição da Classificação feita pelo GPT 4 . . . . .	42
Figura 8 – Distribuição da Classificação feita pelo GPT 3.5 . . . . .	43
Figura 9 – Relação de Contagem das Classificações no GPT-3.5 e GPT-4: Primeiros 20 elementos exceto o item 113 Problemas Funcionais . . . . .	44
Figura 10 – GPT 4 e GPT 3.5 comparados com Jaccard e Sorensen-Dice . . . . .	45
Figura 11 – Distribuição Classificação dos Chamados pelo GPT 3.5 Sem Item 113 Problemas Funcionais . . . . .	47
Figura 12 – Distribuição Classificação dos Chamados pelo GPT 4 Sem o Item 113 <i>Problemas Funcionais</i> . . . . .	49
Figura 13 – Grafo de agrupamento com as mensagens de explicação da classificação da taxonomia . . . . .	49
Figura 14 – Descrição dos 20 primeiros agrupamentos e número de mensagens . . . . .	50
Figura 15 – Distribuição de Similaridade com e sem índice numérico . . . . .	54
Figura 16 – Mapa de Calor das duas classificações para um texto de descrição de chamado . . . . .	56
Figura 17 – Histograma Número de Palavras nos Chamados de Sustentação 1o Lote com 2.000 chamados . . . . .	57
Figura 18 – Histograma de número de palavras segundo lote de classificação, com 78.680 chamados . . . . .	59
Figura 19 – Métricas treinamento <i>fine-tuning</i> do BERT . . . . .	62
Figura 20 – Métricas de <i>fine-tuning</i> do BERT para dados consolidados . . . . .	67
Figura 21 – Distribuição das 20 classificações mais frequentes com a fusão de dados originais com os dados sintéticos . . . . .	70
Figura 22 – Distribuição de número de palavras do conjunto de dados ampliado . . . . .	71
Figura 23 – Métricas de <i>fine-tuning</i> com dados ampliados Ciclo 3 . . . . .	75
Figura 24 – Gráfico de Violino para a Similaridade de Jaccard . . . . .	76
Figura 25 – Índice de Sorensen-Dice . . . . .	76
Figura 26 – Distribuição da soma das métricas de Jaccard com simulação de classi- ficação aleatória . . . . .	77
Figura 27 – Evolução treino e validação por época Ciclo 6 . . . . .	88





## LISTA DE TABELAS

Tabela 1 – Taxonomia para Causa-Raiz de Chamados de Sustentação . . . . .	39
Tabela 2 – 10 Primeiros Elementos Contagem de Itens Classificados pelo GPT 3.5 sem o Item 113 Problemas Funcionais . . . . .	48
Tabela 3 – Comparação mensagens GPT 3.5 e GPT 4: classificação errônea por falta de interpretação . . . . .	51
Tabela 4 – Comparação mensagens GPT 3.5 e GPT 4: maior inferência sobre o texto . . . . .	52
Tabela 5 – Comparação Mensagens GPT 3.5 e GPT 4: :Leitura e Interpretação do <i>stack</i> de erro . . . . .	53
Tabela 6 – Classificação de Vazamento de Dados . . . . .	55
Tabela 7 – Contagem de palavras 2.000 chamados do 1o lote . . . . .	58
Tabela 8 – Dados Balanceados para submissão aprendizado BERT . . . . .	61
Tabela 9 – Precisão, Revocação, Pontuação F1 e Suporte para Cada Classe . . . .	64
Tabela 10 – Classificação e suas probabilidades associadas . . . . .	65
Tabela 11 – Tabela de Comparação LLM e BERT e Probabilidades Retornadas . .	66
Tabela 12 – Categorias Consolidadas e Quantidades . . . . .	67
Tabela 13 – Relatório de classificação <i>fine-tunning</i> com dados agrupados . . . . .	68
Tabela 14 – Classes e Probabilidades . . . . .	68
Tabela 15 – Curva de aprendizado com 5 e 20 épocas . . . . .	69
Tabela 16 – Lista de Problemas e Contagem . . . . .	71
Tabela 17 – Precisão, revocação, F1, Suporte conjunto ampliado de dados . . . . .	74
Tabela 18 – Comparação Qualitativa entre GPT 4 e BERT . . . . .	80
Tabela 19 – Comparação GPT 4 e BERT com os menores índices de Jaccard e Sorensen Dice . . . . .	82
Tabela 20 – Comparação da distribuição classificação feita pelo BERT e GPT 4 . .	83
Tabela 21 – Comparação das principais classificações feitas por BERT e GPT-4 . .	83
Tabela 22 – Métricas do <i>fine-tunning</i> com maxlen=256 e Features = 15.000 . . . .	85
Tabela 23 – Dados de treino e validação por época Ciclo 6 . . . . .	87
Tabela 24 – Métricas de desempenho por classe . . . . .	88
Tabela 25 – Diferença de Classificações GPT4 e GPT 4 Mini e BERT (Ciclo 6) . .	89
Tabela 26 – Exemplo de Chamado e Resultado GPT-4 Turbo . . . . .	106
Tabela 27 – Contagem de Problemas GPT-4 . . . . .	107
Tabela 28 – Contagem de Problemas GPT-3.5 . . . . .	108
Tabela 29 – Contagem de Itens Classificados pelo GPT 3.5 sem o Item 113 Problemas <i>Funcionais</i> . . . . .	109

Tabela 30 – Contagem de Itens Classificados pelo GPT 4 sem o Item <i>113 Problemas Funcionais</i> . . . . .	110
Tabela 31 – Exemplos de Classificação com e sem <i>113 Problemas Funcionais</i> . . . .	111
Tabela 32 – Contagem de classificação do segundo lote para aprendizado modelo BERT . . . . .	112
Tabela 33 – Resultados do Treinamento BERT Usando a Política OneCycle com LR Máxima de 2e-05 com limpeza de mensagem, <i>dropout</i> 20% e <i>l2</i> (0.01)	113
Tabela 34 – Ajuste Fino sem Texto da Classificação: Desempenho por Classe: Precisão, Recall, F1-Score e Suporte . . . . .	113
Tabela 35 – Comparação dos Resultados para Diferentes Conjuntos de Dados e Configurações . . . . .	114

## LISTA DE ABREVIATURAS E SIGLAS

LLM	Large Language Model
BERT	Bidirectional Encoder Representations from Transformers
GPT-4	Generative Pre-Trained Transformer, quarta geração
GPT-3.5	Generative Pre-Trained Transformer, terceira geração e meia.
UTF-8	Unicode Transformation Format - 8 bits
Regex	Regular Expression



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>25</b>
<b>1.1</b>	<b>Sobre a empresa</b>	<b>26</b>
<b>1.2</b>	<b>Sobre a base de Dados</b>	<b>26</b>
<b>1.3</b>	<b>Objetivos</b>	<b>27</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA E DIREÇÕES PARA O TRABALHO</b>	<b>29</b>
<b>2.1</b>	<b>Análise de textos não-estruturados e aprendizado não-supervisionado ou semi-supervisionado</b>	<b>29</b>
<b>2.2</b>	<b>Modelos de Linguagem de Grande Escala</b>	<b>30</b>
<b>2.3</b>	<b>Estudo de adequação da engenharia de <i>prompt</i> para o domínio de dados de sustentação</b>	<b>30</b>
<b>2.4</b>	<b>A base de dados e domínio de informação</b>	<b>32</b>
<b>2.5</b>	<b>Abordagem proposta</b>	<b>33</b>
2.5.1	Extração dos Dados	33
2.5.2	Classificação na Taxonomia com a aplicação de LLM	34
2.5.3	Classificação com Modelo BERT	35
2.5.4	Análise da classificação	36
<b>3</b>	<b>TAXONOMIA PARA ANÁLISE DOS TEXTOS E IDENTIFICAÇÃO DA CAUSA RAIZ</b>	<b>37</b>
<b>3.1</b>	<b>Proposta de taxonomia</b>	<b>37</b>
<b>3.2</b>	<b>Aplicação da taxonomia</b>	<b>39</b>
<b>4</b>	<b>ANÁLISE QUALITATIVA DA CLASSIFICAÇÃO, RESULTADOS E OBSERVAÇÕES</b>	<b>41</b>
<b>4.1</b>	<b>Análise qualitativa e coerência na classificação</b>	<b>41</b>
4.1.1	Classificação com o uso do GPT-4 Turbo	41
4.1.2	Classificação com o uso do GPT-3.5 Turbo	42
4.1.3	Comparação das 20 primeiras classificações	44
<b>4.2</b>	<b>Similaridade de Jaccard e Sorensen-Dice para comparar os resultados do GPT 4 e GPT 3.5</b>	<b>44</b>
<b>4.3</b>	<b>Análises de potenciais alterações na classificação dada mudança de ordem e exclusão de elementos da taxonomia</b>	<b>46</b>
4.3.1	Reclassificação com GPT 3.5 Turbo excluindo item 113 Problemas Funcionais	46
4.3.2	Reclassificação com GPT 4 Turbo excluindo item 113 Problemas Funcionais	46
<b>4.4</b>	<b>Análise das mensagens de explicação associada a taxonomia classificada</b>	<b>48</b>

4.4.1	Observações sobre as Mensagens geradas pelo GPT 3.5 e pelo GPT4 . . .	50
4.4.2	Similaridade entre a classificação e a mensagem de explicação vinda do GPT 4	52
4.4.3	Análise de itens com menor número de classificações . . . . .	54
4.4.4	Relação entre o primeiro item e o segundo item classificados de uma mesma mensagem . . . . .	54
<b>5</b>	<b>ANÁLISE QUANTITATIVA COM APLICAÇÃO DO MODELO BERT, FINE-TUNNING E PROBABILIDADES ASSOCIADAS . . .</b>	<b>57</b>
<b>5.1</b>	<b>Lotes classificação . . . . .</b>	<b>57</b>
5.1.1	O segundo lote de Classificação . . . . .	58
<b>5.2</b>	<b>O Modelo BERT . . . . .</b>	<b>58</b>
5.2.1	Considerações sobre <i>fine-tuning</i> do modelo BERT com dados desbalanceados	58
5.2.2	Análise do modelo BERT usado . . . . .	59
5.2.3	<b><i>Fine-tuning</i> do modelo BERT para tratar dos chamados . . . . .</b>	<b>60</b>
5.2.4	Pré-processamento das mensagens para submissão ao BERT . . . . .	61
<b>5.3</b>	<b>Ciclo 1 de submissão ao modelo BERT, com dados originais . . . .</b>	<b>61</b>
5.3.1	Notas sobre o primeiro treinamento . . . . .	63
5.3.2	Observação inicial sobre predição e quantificação da probabilidade . . . . .	64
<b>5.4</b>	<b>Agrupamento das classificações com aumento dos dados para um conjunto menor de classes e novo treinamento (Ciclo 2) . . . . .</b>	<b>65</b>
5.4.1	Resultados do <i>fine-tuning</i> com as classificações consolidadas . . . . .	66
5.4.2	Indicadores de Precisão, revocação, F1 e Suporte . . . . .	68
5.4.3	<i>fine-tuning</i> com 5 e 20 épocas . . . . .	69
<b>5.5</b>	<b>Uso de dados sintéticos para o <i>fine-tuning</i> do BERT (Ciclo 3) . .</b>	<b>69</b>
5.5.1	Resultados do <i>fine-tuning</i> com base de mensagens ampliada . . . . .	72
5.5.2	Impacto dos Dados Sintéticos . . . . .	73
5.5.3	Considerações . . . . .	73
5.5.4	Análise dos Resultados de Treinamento e Validação e Similaridade de Jaccard e Sorensen-Dice . . . . .	74
5.5.5	Teste de Hipótese para descartar classificador aleatório . . . . .	77
5.5.6	Análise comparativa da classificação GPT 4 e BERT maiores e menores índices de Jaccard e Sorensen Dice . . . . .	78
5.5.7	Classificação de 78.348 Chamados com o BERT com modelo ajustado Ciclo 3	82
<b>5.6</b>	<b>Aplicação de Regularização L2 e <i>Dropout</i> e nova rodada de <i>fine-tuning</i> (Ciclo 4) . . . . .</b>	<b>84</b>
5.6.1	Análise do <i>fine-tuning</i> do BERT com ajustes dos hiperparâmetros . . . . .	84
<b>5.7</b>	<b><i>fine-tuning</i> com aumento de palavras e características (Ciclo 5) .</b>	<b>84</b>
5.7.1	Evolução nas métricas de precisão, revocação e F1 nos Ciclo 3, 4 e 5 . . . .	86
<b>5.8</b>	<b>Experimento para avaliar se dados balanceados geram diferença no treinamento e resultados (Ciclo 6) . . . . .</b>	<b>86</b>

5.8.1	Resultados do experimento . . . . .	87
5.8.2	Comparação Ciclo 3 e Ciclo 6 . . . . .	88
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>91</b>
<b>6.1</b>	<b>Comparação GPT 3.5 e GPT 4 na classificação de chamados de suporte . . . . .</b>	<b>92</b>
6.1.1	Capacidades de classificação de chamados de suporte . . . . .	92
6.1.2	Análise das mensagens de explicação da classificação com <i>Bag-Of-Words</i> e Agrupamento . . . . .	93
6.1.3	Aplicação da Taxonomia . . . . .	93
<b>6.2</b>	<b>Análise Quantitativa com aplicação do modelo BERT . . . . .</b>	<b>93</b>
6.2.1	Resultados . . . . .	94
6.2.2	Recomendações para trabalhos futuros . . . . .	95
	<b>REFERÊNCIAS . . . . .</b>	<b>97</b>
	<b>ANEXOS</b>	<b>103</b>
	<b>ANEXO A – ANEXOS . . . . .</b>	<b>105</b>





## 1 INTRODUÇÃO

O setor financeiro no Brasil é caracterizado pela sua complexidade tanto em termos de produtos quanto nas questões regulatórias. Soma-se a isso o fato de que é cada vez mais notório o crescimento da importância para a satisfação dos clientes nos canais digitais, implicando o atendimento e disponibilidade 24x7 das operações dos bancos e outras instituições financeiras (LONGARAY *et al.*, 2021). A sustentação desses sistemas consiste na atividade de manter os sistemas em funcionamento, provendo resolução em tempo previsto em contrato de problemas e falhas. As falhas interrompem a prontidão necessária de processamento, e podem comprometer a governança e segurança do ambiente de tecnologia. (SAHIBUDIN; SHARIFI; AYAT, 2008) evidencia esses impactos quando propõem uma correlação entre ITIL, COBIT e ISO/IEC 2700. Quando registro dessa falha ou incidente é feita por meio de estruturas de atendimento humano, pois a falha normalmente não tem sua causa raiz imediatamente identificada, não é feita uma classificação no início do atendimento. Dessa forma, a intervenção necessária para dar continuidade a operação requer um processo de investigação, com perguntas, diagnósticos iniciais, descrição do erro e até mesmo uma cópia da pilha de mensagens de erro provida pelo sistema operacional e pela linguagem de programação. Esse processo de investigação é essencialmente não estruturado, pois o registro dos problemas é feito nas conversas entre atendente, ou mesmo desenvolvedor, e o usuário do sistema. A característica intrínseca da falta de estrutura faz com que, após o incidente ter sido resolvido, no contexto da empresa, exista pouca investigação da causa raiz do problema, não somente com o foco no erro do sistema em si, mas no seu entorno de execução. Ilustra-se o ponto acima com falta de parametrização do ambiente, treinamento inadequado de usuário, incompletude das funcionalidades implementadas, infra-estrutura obsoletas, problemas de integração, entre outros.

Frequentemente, os elementos mencionados anteriormente causam falhas que não foram testadas e não estavam previstas na codificação do sistema. Por isso, os chamados abertos – registros numerados e formais da ocorrência da falha que são criados em uma central de atendimento e suporte – não trazem categorização prévia do tipo de erro, mesmo por que ainda será feito um processo de investigação do problema.

Os Modelos de Linguagem de Grande Escala (*Large Language Models* ou abreviadamente LLM) podem trazer uma excelente oportunidade nesse questão. Com seu potencial de entender linguagem natural os LLM ou "Modelos Fundacionais" (BOMMASANI *et al.*, 2021) pode extrair informações sobre os textos de diversas formas (LIU *et al.*, 2023a). No cenário do trabalho proposto, o LLM podem extrair os principais itens que causadores da falha.

E, eventualmente, com aplicação de aprendizado não-supervisionado será possível

agrupar em categorias para aplicação de gestão do negócio, a qual pode indicar grupos de causa-raiz para análises mais profundas. Respostas típicas informariam, por exemplo, se a instalação do pacote de software tem sido negligente com relação a ajustar as variáveis de ambiente da infraestrutura. Ou até, sugerir que determinada sigla de sistema de fato carece de evolução funcional.

## **1.1 Sobre a empresa**

A empresa que gentilmente cedeu as bases de dados tem mais de 25 anos no mercado, e acompanha e fez parte das inúmeras transformações do mercado financeiro. Fundada em 1996 a fez 23 aquisições ao longo da sua trajetória e passou por momentos fundamentais da economia brasileira. A história passa pela estabilização do real e da economia brasileira, à abertura do mercado com a flexibilização do BACEN para abertura de novas instituições, passando pelo surgimento das *Fintechs*, Bancos Digitais, PIX, LGPD e com a mais recente demanda do mercado, o *Open Banking*.

Companhia tem o capital aberto desde 2012, negociados no Novo Mercado da B3 desde 2017. O capital da empresa era pulverizado, com mais de 83% de suas ações ordinárias circulando livremente pelo mercado de capitais. Em 2023 foi adquirida por uma das maiores empresas de tecnologia do setor financeiro mundial.

## **1.2 Sobre a base de Dados**

A base de dados usada é um subconjunto dos dados de produção do suporte e atendimento da empresa, atualizada até meados de 2023, com dados reais. O sistema que executa a função de gerenciamento de incidentes e problemas é o sistema Jira, um sistema líder de mercado (GARTNER, 2022).

Os dados foram obtidos por meio de chamadas de interface com ambiente de desenvolvimento, que foi atualizado em novembro de 2023 com os dados produtivos. Portanto, é necessários o tratamento adequado para não expor os dados dos clientes, que são instituições financeiras do mercado brasileiro.

Foi feita uma extração de um subconjunto desses dados, somente com as informações mais relevantes para esse estudo, como número do chamado, sigla de sistema, negócio (bancário, fundos, previdência e consórcio), descrição do chamado e comentários.

É importante notar que, como descrição e comentários são campos para livre escrita, ambos não tem um padrão nem estrutura definidas, além de aceitarem desde descrição do problema até partes de mensagens de erro produzidas pelos sistemas no seu momento de falha.

Os dados também refletem a realidade da empresa, que tem uma abrangência bastante relevante de clientes do setor financeiro – desde grandes bancos até pequenas

gestoras de ativos – e diversidade dos sistemas, tanto em linhas de negócios distintas quanto na sobreposição de sistemas com o mesmo objetivo de negócio, resultado do histórico de aquisições da empresa.

### 1.3 Objetivos

O objetivo geral deste estudo é investigar o desempenho dos Modelos de Linguagem de Grande Escala abertos para classificar o atendimento e resolução de incidentes do setor financeiro. Em particular, serão propostos e avaliados *promptings* de LLMs em textos provenientes de chamados de suporte, visando identificar padrões, categorizar problemas e, assim, aprimorar a capacidade de resolução de falhas de forma mais eficiente.

O objetivo geral pode ser organizado nos seguintes objetivos específicos:

- Investigar os desafios enfrentados na sustentação de sistemas, com ênfase na falta de estrutura nos processos de resolução de incidentes e na subsequente falta de investigação da causa raiz;
- Avaliar o potencial dos Modelos de Linguagem de Grande Escala na automação e melhoria dos processos de investigação, categorização e análise de chamados de suporte, organizando em temas e subtemas;
- Desenvolver e aplicar *prompts* (instrução) específicos para o modelo de linguagem, utilizando o conhecimento de domínio do problema para extrair informações relevantes das descrições e comentários dos chamados;
- Utilizar técnicas de sumarização de texto para extrair a essência dos problemas descritos nos chamados, buscando proporcionar uma visão mais estruturada para análises posteriores.



## 2 FUNDAMENTAÇÃO TEÓRICA E DIREÇÕES PARA O TRABALHO

Os LLM são modelos complexos, grandes, variados e exaustivamente treinados (ZHAO *et al.*, 2023), e o advento destes grandes modelos de linguagem trouxe à luz uma gama de novas possibilidades de aplicação no ciclo de desenvolvimento de sistemas. Estudos de aplicação sobre a análise de registros de processamento e sumarização em programas (SRIDHARA; MAZUMDAR *et al.*, 2023; SUN *et al.*, 2023) com LLM indicam que eles podem trazer uma visão mais analíticas do que está ocorrendo. E a capacidade de identificação e correção de erros de programa, (JESSE *et al.*, 2023; LIVENTSEV *et al.*, 2023; WENG; ANDRZEJAK, 2023; LIU *et al.*, 2024; ZHANG *et al.*, 2023) potencializam expectativas interessantes para entendimento e correção dos problemas que causam falhas. Entretanto, para identificar a causa raiz de problemas tendo como fundação o entendimento de descrições que não foram feitas por sistemas ou máquinas e sim registros com conteúdos gerado por analistas de sistemas e usuários tem desafios intrínsecos. Com a ressalva no foco em conteúdos de mídia social, os estudos de (KRUMM; DAVIES; NARAYANASWAMI, 2008; WYRWOLL; WYRWOLL, 2014; NAAB; SEHL, 2017) ajudam a iluminar os desafios de conteúdo relativos à relatos humanos. Tanto analistas de sistemas em suporte, quanto os próprios clientes, ao abrirem chamados, descrevem os problemas em sua forma mais humana, sem necessariamente ter categorias prévias de classificação.

### 2.1 Análise de textos não-estruturados e aprendizado não-supervisionado ou semi-supervisionado

O domínio dos textos de chamados de suporte é bastante peculiar. Sendo uma base privada, o domínio tem características *sui generis*. Técnica de aprendizado supervisionado seriam exaustivas e talvez não trouxessem a luz novos conhecimentos, potencialmente ficando enviesada pelos diversos especialistas dos sistemas envolvidos. Portanto, é interessante buscar a aplicação de técnicas não-supervisionadas para a sustentação do trabalho. Em (REZENDE; MARCACINI; MOURA, 2011), é possível entender as técnicas de algoritmos de aprendizagem não-supervisionados aplicados no contexto de textos não estruturados. (ROSSI; MARCACINI; REZENDE, 2013) avança e aplica diversas técnicas de aprendizado não supervisionado em um amplo conjunto de coleções de textos, estabelecendo uma referência importante no assunto. (WEISS; INDURKHYA; ZHANG, 2015), Capítulo 1, afirma que mineração de texto busca fazer classificação e predição, e traz também uma espécie de abordagem para que esses dois objetivos sejam atingidos. Já (ROSSI; MARCACINI; REZENDE, 2014) faz uma proposição de um *framework* (acabouço) para extração de informação em palavras-chaves, permitindo a incrementação de *clusters* (agrupamentos). Para avançar no aspecto de entendimento de questões subjetivas, como

sentimento, (MARCACINI *et al.*, 2018) introduz um conceito de transferência de aprendizado entre domínios diferentes para combinar características rotuladas, não-rotuladas e características linguísticas, de forma a contornar limitações de treinamento necessários para um novo domínio. (SILVA; MARCACINI, 2021) traz um estudo da aplicação de modelos BERT (*Bidirectional Encoder Representations from Transformers*) para análise de sentimentos. (ARAUJO; GÔLO; MARCACINI, 2022) mostra que os métodos tradicionais de aprendizado de máquina para análise de opinião em textos curtos são eficazes, chegando próximo a técnicas com redes neurais, como Modelos Neurais de Linguagem (*Neural Language Models* - NLM).

## 2.2 Modelos de Linguagem de Grande Escala

Recentemente, a aplicação de Modelos de Linguagem de Grande Escala tem se mostrado bastante interessante e com resultados que surpreendem (BROWN *et al.*, 2020; CHOWDHERRY *et al.*, 2023; CHUNG *et al.*, 2022; HOFFMANN *et al.*, 2022; LIU *et al.*, 2023b) e sinalizam um horizonte interessante no âmbito do entendimento e extração de informação mais subjetiva de textos. E esses modelos tendem a aumentar e ganhar ainda mais poder de tratar problemas dos mais diversos (KAPLAN *et al.*, 2020). Mesmo sem treinamento específico no domínio do problema, os LLMs trazem grandes e novas oportunidades de análise (BROWN *et al.*, 2020; BOMMASANI *et al.*, 2021; IYER *et al.*, 2022). Para os chamados de sustentação, existe o potencial de aplicação das LLMs para trazer informação relevante que até então não seria possível de forma massiva. Informações como recuperação sintética do que foi o chamado, inferência da causa-raiz, sumarização dos problemas podem levar a agilidade na resolução, análise gerencial de ações mais estruturantes e mesmo a proposição de novas capacidades e sistemas.

## 2.3 Estudo de adequação da engenharia de *prompt* para o domínio de dados de sustentação

Para este trabalho se faz necessário entender como melhor aplicar a técnica de *prompting*. (LIU *et al.*, 2023a) traz um estudo abrangente e sistemático das formas de elaborar *prompts*, introduzindo uma tipologia mostrada na Figura 1. Para o trabalho é necessário um desenvolvimento de *prompt* que possa:

- (a) ser entendido pelo negócio e, se necessário, ajustado com a participação da equipe;
- (b) possibilitar o acesso simplificado aos LLMs, sem necessidade de conversões automatizadas de *tokens* (símbolos) ou outras soluções complementares;
- (c) ser eficaz para o domínio de dados do problema;
- (d) poder ser ajustado para várias técnicas sem engessar em uma técnica de *prompt* somente.

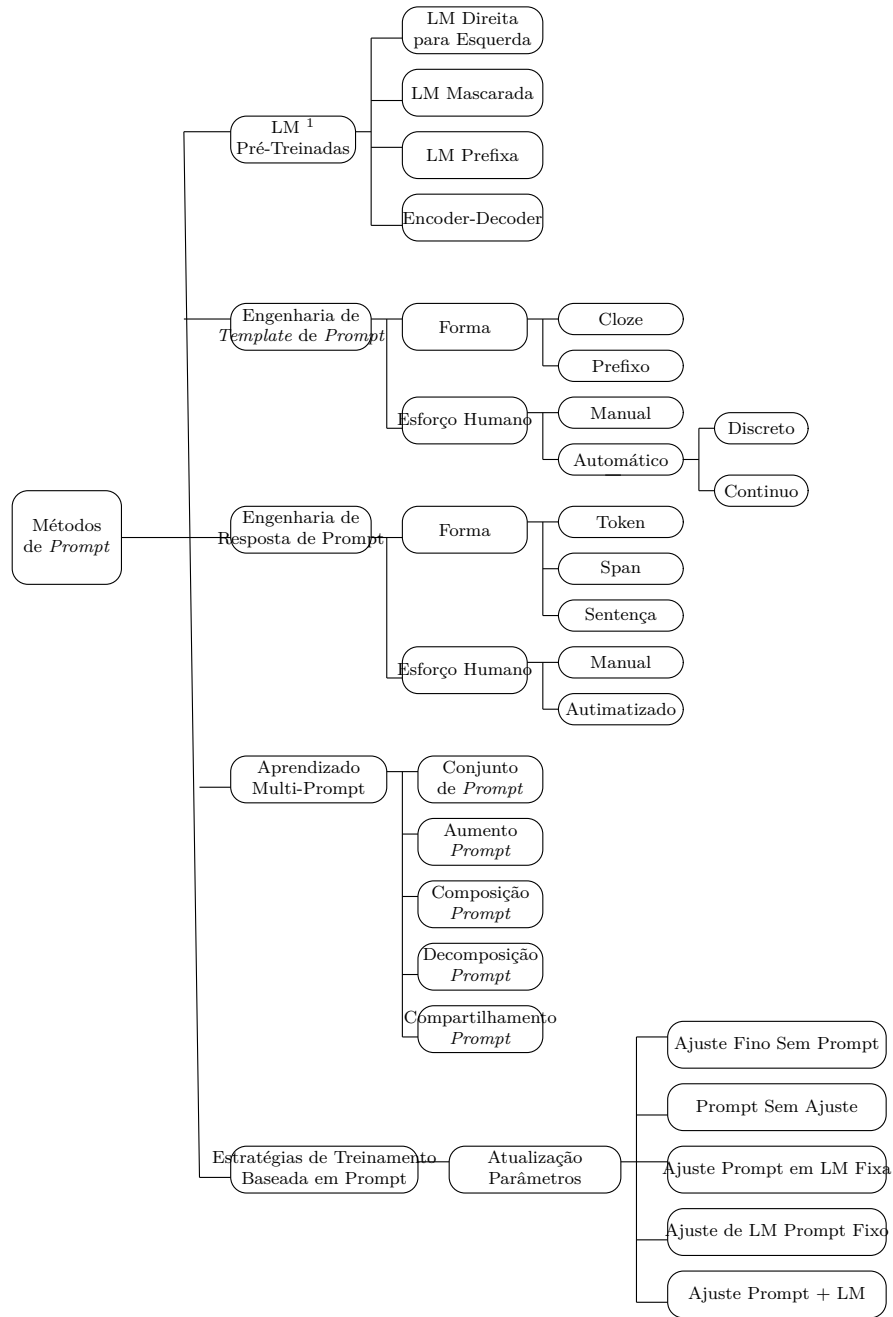


Figura 1 – Tipologia de *Prompting* - Adaptado de (LIU *et al.*, 2023a)

Isto posto, a forma de elaboração do *prompt* será manual, com declarações diretas, buscando aplicar as técnicas de *zero-shot*, *few-shot* e *chain-of-thoughts*. A capacidade dos modelos pré-treinados de aprender a sumarizar, responder questões, sem um treinamento específico foi apontado em (RADFORD *et al.*, 2019). Esse aprendizado instantâneo (*Zero-Shot Learning*), segundo (WEI *et al.*, 2021), é uma técnica que pode trazer bons resultados para modelos de linguagens aprimorados. O estudo também traz uma análise da aplicação dessa técnica comparando FLAN com o GPT-3, que de uma certa maneira, com FLAN tendo um claro desempenho a frente do GPT-3 com a técnica de *Zero-Shot*. O estudo de (BROWN *et al.*, 2020) mostra que os LLM são aprendizes de poucos minutos (*few-shot*

*learners*) e tem capacidades poderosas para produzir resultados sem necessariamente serem treinadas para um domínio de conhecimento ou dados. Os LLMs pode expandir, sintetizar e concluir sobre um domínio de informação para qual sequer foram apresentadas. (SCHICK; SCHÜTZE, 2020a) mostra que, apontando as tarefas que devem ser executadas e combinando com aprendizado supervisionado com alguma configurações de *few-shot*, alavanca-se o conhecimento contido em um modelo pré-treinado. Os mesmos autores em (SCHICK; SCHÜTZE, 2020b) evidenciam que mesmo modelos de linguagem menores podem ter um desempenho otimizado quando aplicada técnicas de *few-shot*. Outra forma de obter resultados melhores para classificação de textos na esfera de uso de modelos pré-treinados é o emprego de padrões de construção de exploração do conhecimento de tais modelos (SCHICK; SCHÜTZE, 2021). Outro aspecto é o ajuste fino dos *prompts* para *few-shot* sendo, através de modelos ou padrões como descritos em (GU *et al.*, 2021; HAN *et al.*, 2022; LIU *et al.*, 2023b; LESTER; AL-RFOU; CONSTANT, 2021; LI; LIANG, 2021).

Já solicitar tarefas explicitamente, colocando a necessidade da exposição da linha de raciocínio (*chain-of-thought*), apura a capacidade de resolução de tarefa envolvida, com consequente aumento da taxa de acerto (CHUNG *et al.*, 2022; WEI *et al.*, 2022b). Em (WANG *et al.*, 2022) é proposta uma forma de auto consistência, que gera várias respostas e cujo passo final do método é escolher a resposta mais provável, dado o conjunto de respostas. Contudo, o método só pode ser aplicado quando a resposta final é uma resposta fechada. Por fim, para o entorno do trabalho de análise de causa-raiz dos chamados de sustentação, pode-se ter algum tipo de análise distinta do esperado, capacidades emergentes (WEI *et al.*, 2022a) podem se manifestar, dado que o texto que descreve o chamado é bastante diverso, com comentários e até mesmo a pilha de mensagens de erro. No obstante, vale a observação do contraponto sobre propriedade emergente feita em (SCHAEFFER; MIRANDA; KOYEJO, 2024).

## 2.4 A base de dados e domínio de informação

Foram extraídos dois grupos de dados. O primeiro contendo dois mil chamados os quais foram usados para uma análise qualitativa, para avaliar se a resposta gerada pelo LLM é coerente com uma potencial causa raiz, o que foi feito por um especialista de domínio. Outra lote de dados foi extraído, contendo setenta e oito mil registros, para que fossem submetidos ao modelo BERT. A base contém tanto descrições quanto, em alguns casos, a cópia da pilha de mensagens de erro. Sobre cada campo, foi aplicada uma preparação dos dados com a eliminação de caracteres especiais, para evitar problemas de leitura. Outrossim, mesmo fazendo parte dos campos, os textos da pilha das mensagens de erro não serão explorados nesse trabalho. Os registros extraídos foram organizados em:

- id: é o identificador do número do chamado;



- `description`: traz a descrição do chamado de forma resumida;
- `comments`: aglutina todas as informações textuais do chamado, conversas, solicitações e até mesmo a pilha de mensagens de erro do sistema. Esse campo consta somente do lote de dois mil dados para análise qualitativa.
- `siglaSistema`: é sigla do sistema e traz a sigla de um dos 44 sistemas envolvidos;
- `areaSistema`: area de negócio que são: Fundos, Previdência, Bancos e Consórcio.

Os erros são descritos tanto por profissionais de atendimento de suporte, que nesse caso são analistas de sistemas, como o próprio cliente que pode abrir um chamado. Essa diversidade de sistemas, negócios e formas de abertura geram uma diversidade ampla no registro do problema. (JANCZAREK; SOSNOWSKI, 2015) evidencia que quanto mais refinado é o relato do problema, melhor é a qualidade da ação para dirimir a falha. No âmbito de textos feito por humanos, que relatam falhas de sistemas, a granularidade fina descrição infortunosamente não tem uniformidade em detalhes (GIGCH, 1986).

## 2.5 Abordagem proposta

Em resumo, a abordagem proposta aplica modelos de Linguagem em Larga Escala para fazer uma classificação dos chamados dentro de uma taxonomia sugerida. Espera-se que o LLM tenha capacidade suficiente de interpretação e inferência para mapear a descrição do chamado em definições pré-estabelecidas, mesmo quando não estiver claro uma única causa provável. Nesse caso, seguramente mais de uma causa pode ser apontada, que de certa forma pode fazer sentido em uma análise preliminar do conteúdo relativamente abstrato do texto do referido chamado. Com a rotulagem em uma parte dos chamados, será treinado um modelo de classificação, e nesse caso o modelo escolhido foi o BERT (DEVLIN *et al.*, 2018), para rotular os chamados restantes. Com o resultado geral, serão feitas análises de acuracidade, comparações de rotulagens por grupos de sistemas, ou mesmo linhas de negócio, e espera-se ter padrões comparáveis entre os chamados rotulados pelo LLM e pelo BERT dentro dos grupos de sistemas e linhas de negócio.

### 2.5.1 Extração dos Dados

A primeira etapa do trabalho é extrair as informações do sistema Jira. O banco de dados disponibilizado contém todos os campos de um chamado e muitos campos auxiliares de controles de fluxo. A extração inicial dos dados teve a seleção somente dos campos indispensáveis para o trabalho. Além disso, houve a limpeza de caracteres especiais com o uso de expressões regulares. Outra limpeza foi a remoção de comentário padronizados e sistematicamente repetidos nas descrições e que não agregam informação semântica.

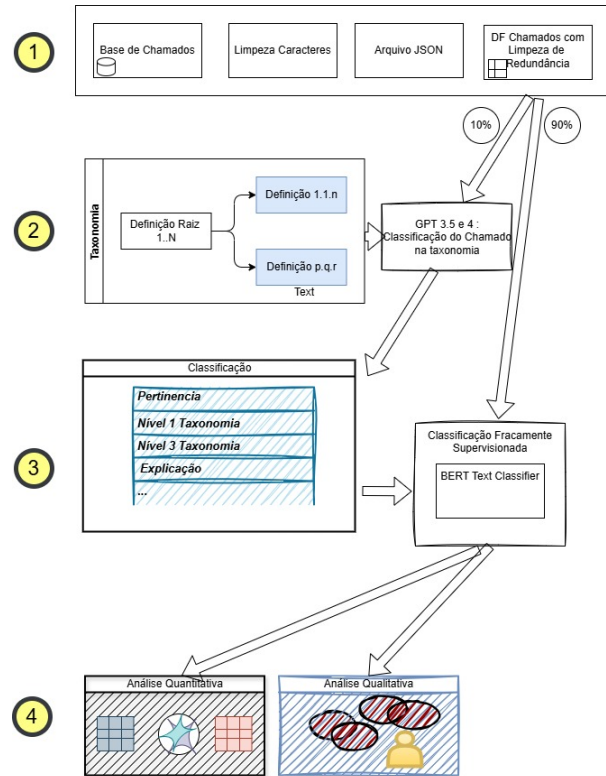


Figura 2 – Abordagem de Execução

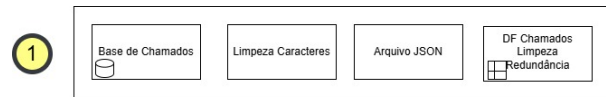


Figura 3 – Extração e Limpeza dos Dados

### 2.5.2 Classificação na Taxonomia com a aplicação de LLM

Com a taxonomia sendo parte do *prompt* é feita a submissão para a *LLM*, nesse caso com o uso do GPT 3.5 (BROWN *et al.*, 2020) e GPT 4, do texto descritivo do chamado para a classificação da causa-raiz do problema para uma posterior análise gráfica ou gerencial. A forma para tratamento de maior consistência é perguntar diretamente para a *LLM* a classificação no primeiro nível da taxonomia e no terceiro nível. A classificação foi feita na seguinte estrutura:

[Nível de pertinência, Item, Número do Item Enumerado, Explicação]

Nível de pertinência  $x \in Alto, Médio, Baixo$

Exemplo: ['Alto', '4 Segurança', '4.2.3 Vazamento de dados', 'O texto mostra que houve problemas com dados que estavam sendo acessados indevidamente'] Espera-se mais de uma classificação, com diferentes níveis de pertinência, dado o nível abstrato de descrição do chamado.

O primeiro lote de dados, dois mil chamados, é subetido integralmente. À esse lote,

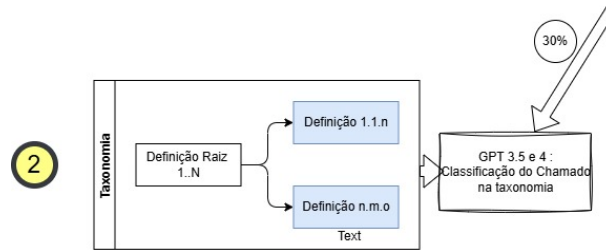


Figura 4 – Classificação LLM na taxonomia

são aplicadas as variações de *prompt* e LLM. Os *prompts* serão alterados na ordem da taxonomia proposta, para verificar se não há viés de resposta dada a ordem. E o LLM será variada, ao menos entre GPT-3 e GPT-4 para conferir se há de fato mudança estrutural na resposta provida por tais modelos. O segundo lote, de setenta e oito mil chamados foram extraídas amostras aleatórias, num total de cinco mil chamados e estas submetidas ao LLM. Dada o nível abstrato de descrição do chamado, o LLM retorna mais de uma potencial explicação para a causa raiz. Esse fato tem seu lado positivo, pois a priori, em uma análise inicial, é possível ter-se mais do que uma potencial causa de problema. A multiplicidade de potenciais causas-raiz pode ser usada para fazer o ajuste fino no modelo BERT. Outra questão relevante para essa etapa da abordagem é que nem sempre os LLMs retornam precisamente no formato sugerido no *prompt*. Isto posto, nessa etapa também é feito um tratamento do retorno dos LLMs, buscando:

- (a) eliminar eventuais caracteres especiais;
- (b) colocar na mesma forma e tipo de variável os dados obtidos;
- (c) filtrar e ordenar por pertinência Alta e Média e eliminar pertinência baixa;
- (d) verificar se os dados retornados pertencem ao conjunto de itens da taxonomia definida.

As ações acima objetivam deixar os dados preparados para serem submetidos ao modelo BERT e que, mesmo com uso de diferentes LLMs - e conseqüente variação nos dados retornados por elas -, os dados sejam tratados e se tenha um conjunto com qualidade adequada para outras análises, tais como a consistência da explicação feita para uma classificação sugerida pelo LLM.

### 2.5.3 Classificação com Modelo BERT

Uma vez classificados pelo GPT 4, o conjunto de chamados é submetido ao BERT (DEVLIN *et al.*, 2018) passando o texto descritivo do chamado e a classificação da causa-raiz. O uso do BERT foi proposto pelas seguintes razões:

- (a) é uma forma mais econômica financeira e computacionalmente que a rotulação com LLM em toda a base de dados.

- (b) provê uma maneira de medir a acuracidade pelas probabilidades associadas na classificação feita pelo BERT.
- (c) pode ajudar a estabelecer critério mais objetivos de comparação e limiares para tomada de decisão baseadas nas probabilidades.

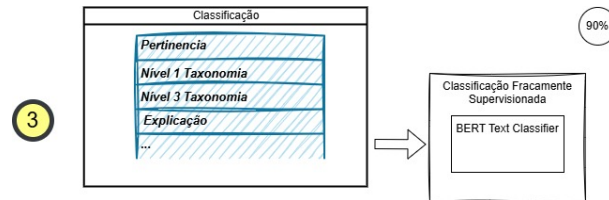


Figura 5 – Classificação Fracamente Supervisionada

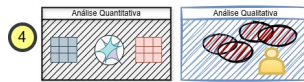


Figura 6 – Análises Visuais e Comparativas

#### 2.5.4 Análise da classificação

As análises objetivam responder ou dar indícios se a classificação feita pelo LLM traz de fato informação relevante para a solução do chamado. São feitas duas grandes linha de análise. A primeira delas é avaliar o quão coerente é uma classificação feita dado o texto de entrada. Para isso, é feito uma observação de especialista em um conjunto limitado de chamados classificados bem como uma comparação direta entre os dois modelos fundacionais usados para o trabalho, o GPT 3.5 e o GPT 4. A outra análise é buscar mais dados quantitativos para avaliar a acuracidade de classificação, mesmo que de forma indireta. A classificação com o BERT enseja trazer mais luz nesse sentido. A análise visa calcular métricas como precisão, recall, F1-score e acurácia para cada classe. Métricas gerais, como a acurácia, para observar se as classes estão balanceadas no conjunto de dados, pois desequilíbrios podem afetar o desempenho do modelo. Será aplicada Jaccard e índice de Sørensen-Dice para avaliar similaridade de classificação entre GPT 4 e BERT.

### 3 TAXONOMIA PARA ANÁLISE DOS TEXTOS E IDENTIFICAÇÃO DA CAUSA RAIZ

De forma análoga a análise de sentimentos, para melhorar a mineração da causa-raiz baseada em características pode ser profícuo o uso de ontologias para abordar o problema em questão (PENALVER-MARTINEZ *et al.*, 2014). A ontologia fornece a estrutura filosófica e conceitual e direcionamento mais formal necessário para estabelecer as relações hierárquicas inerentes às taxonomias dentro de um determinado domínio de conhecimento (REES, 2003; GUARINO; WELTY, 2000). Todavia, mesmo considerando relevante para elaborações futuras, este trabalho não trará uma discussão ontológica, focando, neste momento, apenas na taxonomia. Nessa seção, é proposta uma taxonomia específica para esse estudo. Essa estrutura taxonômica pretende promover a abertura das possibilidades de aplicação das técnicas de *prompt*, *fine-tuning*, padrões e auto-consistência descritas nas subseções anteriores. Um dos primeiros estudos sobre qual seria uma estrutura adequada para buscar uma taxonomia que pudesse ajudar a entender por que os sistemas falham é trazida em (GIGCH, 1986), que define um metamodelo para taxonomias de falhas em sistemas. Em (RALPH, 2018) as diretrizes e estudos sobre taxonomias de processos engenharia de software sintetizam conceitos metodológicos para geração de tais taxonomias, ampliando um pouco mais a abrangência e uso no contexto do trabalho. (USMAN *et al.*, 2017) conduz um estudo interessante sobre o crescente número de taxonomias que foram criadas até a data do artigo, indicando que não há uma única forma universal para tal construção. Como objetivo do trabalho é analisar a causa raiz dos problemas, outra fonte possível é buscar base em formas propostas como Classificação Ortogonal de Defeitos (CHILLAREGE *et al.*, 1992). Em (SÖYLEMEZ; TARHAN, 2018) há uma síntese de estudos sobre Classificação Ortogonal de Defeitos e propõem um método em forma de processo para realizar tal análise, incluindo um Modelo de Formulário de atributos para a análise. Apesar da descrição dos atributos de análise contribuírem para a taxonomia desse trabalho, o artigo aborda as causas endógenas de falha. E o processo, método e os atributos sugeridos não ajudam a compor uma taxonomia mais completa que possa ajudar a classificar a causa em potencial do problema deste trabalho, que podem ser também exógenas ao sistema.

#### 3.1 Proposta de taxonomia

Para buscar classificar os erros ou problemas na operação de sustentação, é sugerida uma taxonomia que busca compreender os aspectos mais relevantes bem como abrangente do ponto de vista de tipos de falhas. A Tabela 1, é o resultado da análise das referências citadas em conjunto com as demandas de negócio, e apresenta categorias de falhas típicas de sistemas, software, hardware e itens que são relativos a operação do sistema pelos

usuários finais. A taxonomia proposta para o trabalho consiste em três níveis:

- 1o Nível: grandes grupos problemas, separando temas de tecnologia, ambiente de processamento, fatores humanos e segurança
- 2o Nível: grupos de Classes de Falhas, discernindo as sub-categorias, como software hardware, dados, infra-estrutura, etc
- 3o Nível: tipos de falhas pertinentes aos grupos de classes de falhas.

O objetivo do trabalho é classificar a falha ou problema no 3o Nível. Contudo, é possível que as classificações intermediárias no Nível 2 sejam necessárias para ter maior acuracidade e auto-consistência da resposta.

1o Nível	2o Nível	3o Nível
1. Questões técnicas	1.1 Falhas de Software  1.2 Falhas de Hardware  1.3 Falhas em Dados	1.1.1 Erros de Codificação 1.1.2 Problemas de Integração 1.1.3 Problemas Funcionais 1.1.4 Versões Não Suportadas 1.1.5 Versão de Software Desatualizada 1.1.6 Software Obsoleto  1.2.1 Mau Funcionamento da Rede 1.2.2 Problemas de Conectividade 1.2.3 Recursos de Hardware Insuficientes 1.2.4 Outras Falhas de Hardware  1.3.1 Falta de Dados 1.3.2 Dados Corrompidos 1.3.3 Dados Duplicados 1.3.4 Dados Inconsistentes
2. Fatores Ambientais	2.1 Infraestrutura  2.2 Dependências Externas	2.1.1 Atualizações/Mudanças de Rede 2.1.2 Migração de Servidor 2.1.3 Versão do Banco de Dados 2.1.4 Migração de Banco de Dados 2.1.5 Versão do Sistema Operacional 2.1.6 Configuração Incorreta  2.2.1 Alterações na API de Terceiros 2.2.2 Alterações em Software de Terceiros 2.2.3 Falhas Externas ao Sistema
3. Fatores Humanos	3.1 Erros do Usuário  3.2 Operação por Técnico Não Qualificado	3.1.1 Uso Incorreto do Sistema 3.1.2 Falta de Treinamento dos Usuários 3.1.3 Configurações Incorretas do Usuário 3.1.4 Falta de Conhecimento do Usuário Sobre Processos de Negócios  3.2.1 Problemas de Configuração 3.2.2 Problemas de Instalação 3.2.3 Erros de Parâmetros 3.2.4 Falta de Conhecimento Técnico Sobre Software 3.2.4 Falta de Conhecimento Técnico Sobre Hardware
4. Incidentes de Segurança	4.1 Ataques Cibernéticos  4.2 Violações de Dados	4.1.1 <i>Malware</i> 4.1.2 Negação de Serviço (DoS)  4.2.1 Acesso Não Autorizado 4.2.2 Vazamento de Dados

Tabela 1 – Taxonomia para Causa-Raiz de Chamados de Sustentação

### 3.2 Aplicação da taxonomia

A análise e classificação de textos em grandes volumes de dados apresenta desafios significativos que demandam uma abordagem equilibrada e eficiente. Esta subseção explora a intersecção entre diferentes metodologias de processamento textual, propondo uma

solução que combina técnicas tradicionais de mineração de dados com a aplicação de taxonomia. (REZENDE, 2003) mostra um leque de técnicas e abordagens combinadas para fazer mineração de textos, as quais podem ser recompostas objetivando o tratamento do domínio de dados do trabalho aqui apresentado. (WEISS; INDURKHYA; ZHANG, 2015) é também uma interessante fonte de recomendações de abordagem e métodos de mineração de dados, considerando algoritmos de classificação não-supervisionados para volumes massivos de dados. Nesse momento, é importante a ponderação entre a aplicação exclusiva de Modelos de Linguagem de Larga Escala vis-a-vis a aplicação de métodos não-supervisionados ou semi-supervisionados de classificação. LLMs, como perscrutado nas sessões anteriores, são modelos poderosos, mas têm seus custos. Não obstante, modelos de classificação não supervisionados ou semi supervisionados podem oferecer uma bom aprumo entre custo e benefício. Nesse intuito-far-se-á uma proposição de uma ponte entre essas abordagens com a aplicação de uma taxonomia. Análogo a obtenção de sentimentos por características descritas em (FELDMAN, 2013), a análise da causa-raiz de problemas em chamados pode se beneficiar da mesma abordagem proposta em (PENALVER-MARTINEZ *et al.*, 2014), que, ao definir uma estrutura ontológica e aplicá-la, promove vantagens como a ampliação de uso de algoritmos distintos de aprendizado de máquina bem como ser desnecessário adotar treinamentos específicos. Soma-se isso a composição de *prompts*, cujas técnicas como *few-shot* ou *chain-of-thoughts* podem ser melhor exploradas com as definições da taxonomia proposta.



## 4 ANÁLISE QUALITATIVA DA CLASSIFICAÇÃO, RESULTADOS E OBSERVAÇÕES

Para a análise qualitativa, que se baseia na avaliação humana e de especialista, a comparação entre os dois modelos fundacionais usados foi equânime com a submissão dos mesmos 2.000 chamados para o GPT 3.5 e para o GPT 4. As análises e comparações abaixo foram feitas:

- (a) distribuição de classificação, tendo em vista que foi selecionado a classificação com pertinência mais alta dentro do conjunto de classificações feitas para um chamado;
- (b) diferenças de classificação entre GPT 3.5 e GPT 4;
- (c) mudança de ordem da classificação escrita no *prompt*;
- (d) categorias mais classificadas e viés na classificação feita pelo LLM;
- (e) similaridade entre as causas apontadas e as mensagens de explicação;
- (f) acuracidade na classificação feita pelo modelo BERT;

### 4.1 Análise qualitativa e coerência na classificação

O primeiro passo da análise é entender como foram feitas as classificações o quão coerente é a resposta dos dois modelos. Para isso, a próximas subseções mostram a distribuição da classificação dos dois modelos, avaliam se as classificações foram feitas dentro da taxonomia proposta, e as diferenças mais relevantes entre os dois modelos. A subseção também traz se há de fato diferença na classificação quando a ordem da taxonomia do *prompt* é completamente invertida. Por fim, é feita uma análise das explicações feitas pelo LLM, e aqui foi escolhido o resultado do GPT 4 por conta do melhor resultado que esse modelo trouxe. É avaliado se essas explicações dadas para a classificação são coerentes com o texto do item enumerado, para isso aplicou-se técnica de *bag of words*, agrupamento e similaridade. A seção também traz tabelas com os exemplos para ilustrar as diferentes situações encontradas e detalhes maiores podem ser vistos nos anexos.

#### 4.1.1 Classificação com o uso do GPT-4 Turbo

A primeira rodada de classificação com o uso de LLM foi feita com o GPT-4 (ACHIAM *et al.*, 2023). A extração dos dados foi feita da seguinte forma: Os dois mil chamados foram submetidos tanto com o *prompt* seguindo a ordem da taxonomia apresentada na Tabela 1 quanto com o *prompt* completamente invertido na taxonomia. Como pode ser visto na Tabela 26, para cada chamado submetido a classificação, existem

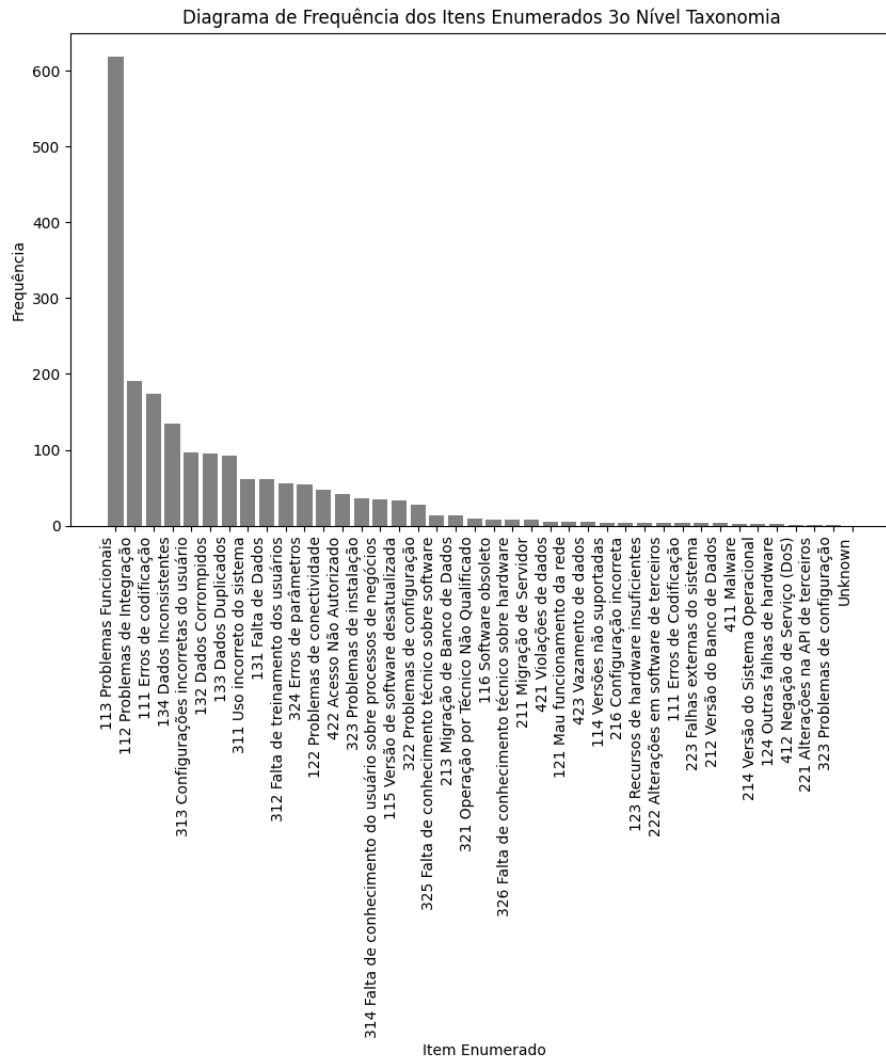


Figura 7 – Distribuição da Classificação feita pelo GPT 4

um ou mais itens classificados. Esse fato, per si, é desejável e o *prompt* construído assim o solicitou. No início da investigação do chamado para entender a causa-raiz, existem possibilidades distintas de causas e é importante que essas possibilidades seja apontadas. Contudo, para fins de *fine-tuning* do modelos BERT, foi feito um filtro selecionando um único item enumerado pertencente ao Nível 3 da Taxonomia. Essa seleção traz o primeiro elemento que teve a classificação de pertinência mais alta, entre as possíveis Alta, Média e Baixa. Feito esse filtro, tanto na submissão com o *prompt* seguindo a ordem da inicial taxonomia quanto na ordem invertida, onde foram invertidos o primeiro nível e invertidos os itens do 3o nível, mantendo-se os grupos, para os 2.000 chamados analisados, a classificação foi rigorosamente a mesma. A figura 7 ilustra a distribuição dessa classificação.

#### 4.1.2 Classificação com o uso do GPT-3.5 Turbo

No caso do modelo GPT-3.5 (ACHIAM *et al.*, 2023) a submissão pode ser feita com os mesmos dois mil chamados. Foi necessário um pequeno pré-processamento para



Figura 8 – Distribuição da Classificação feita pelo GPT 3.5

tratar o retorno do GPT-3.5, que por vezes variou no formato de resposta bem como nas marcações de início *strings*.

A Figura 8 mostra que o item 113 Problemas Funcionais é também dominante na classificação no GPT-3.5, assim como o foi no GPT-4. É possível verificar que, em alguns chamados, não foi feita a classificação, mas colocada diretamente a explicação para a classificação solicitada no *prompt*. O fato ocorreu de forma rigorosamente igual tanto na ordem original da taxonomia quanto na ordem invertida, ou seja, foi colocada a mesma frase.

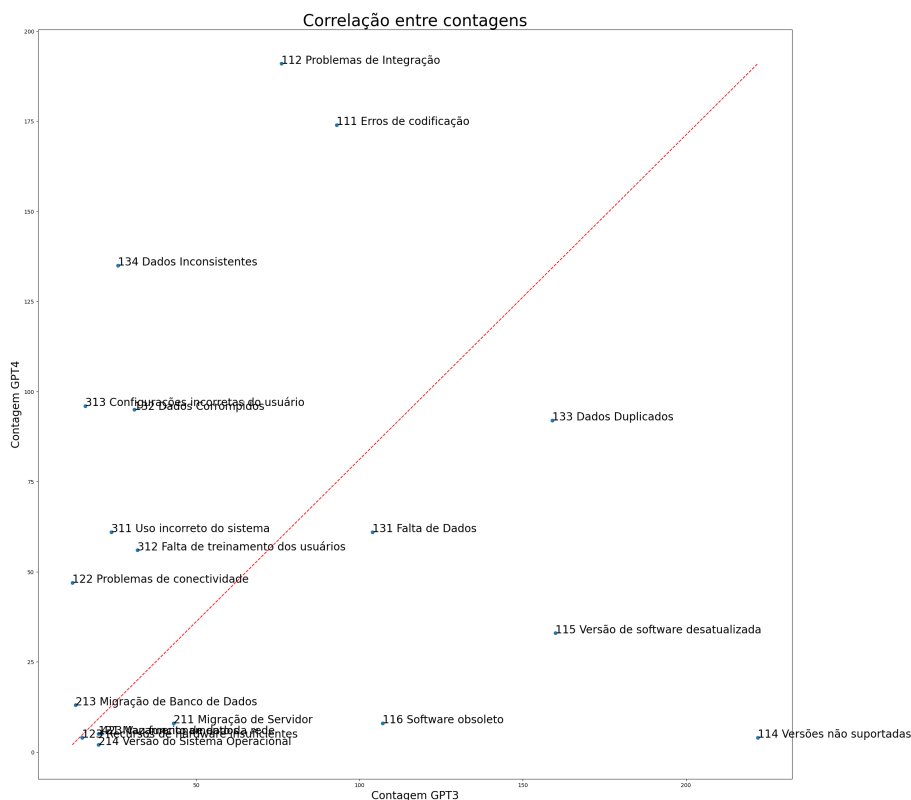


Figura 9 – Relação de Contagem das Classificações no GPT-3.5 e GPT-4: Primeiros 20 elementos exceto o item 113 Problemas Funcionais

#### 4.1.3 Comparação das 20 primeiras classificações

Um fenômeno interessante observado foi a alta ocorrência da classificação no item 113 Problemas Funcionais. Este item teve 619 classificações no GPT-4 e 662 no GPT-3.5, mas somente 319 chamados foram classificados no item 113. Em comum entre os dois modelos fundacionais houveram somente 439 chamados com a mesma classificação. Ressalva-se aqui que foi elencado um único item, de pertinência Alta, do conjunto de itens que foram classificados. Nos itens subsequentes, existe uma variação relevante, como observado na Figura 9

## 4.2 Similaridade de Jaccard e Sorensen-Dice para comparar os resultados do GPT 4 e GPT 3.5

As medidas de similaridade, como os coeficientes de Jaccard e Sorensen-Dice, desempenham um papel fundamental na comparação de conjuntos de dados, especialmente em tarefas de classificação automática, como a categorização de chamados por modelos de linguagem natural. Essas métricas quantificam o grau de sobreposição entre dois conjuntos, oferecendo uma perspectiva robusta sobre a similaridade entre as classificações realizadas por diferentes versões de modelos, como o GPT-3.5 e o GPT-4. O coeficiente de Jaccard é uma medida que avalia a similaridade entre dois conjuntos ao calcular a razão entre

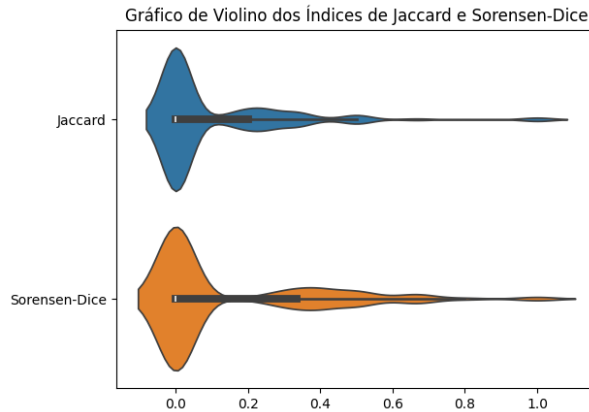


Figura 10 – GPT 4 e GPT 3.5 comparados com Jaccard e Sorensen-Dice

o número de elementos em comum (interseção) e o total de elementos distintos nos dois conjuntos (união). Formalmente, para dois conjuntos  $A$  e  $B$ , o coeficiente de Jaccard é dado por:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Essa métrica varia de 0 a 1, onde 0 indica que os conjuntos não compartilham nenhum elemento e 1 indica que os conjuntos são idênticos. Aplicada à comparação das classificações de chamados por GPT-3.5 e GPT-4, o coeficiente de Jaccard ajuda a identificar o quão semelhantes são as categorias atribuídas por ambos os modelos, considerando tanto as coincidências quanto as divergências nas classificações. A métrica simétrica, o que significa que a Similaridade de Jaccard entre  $A$  e  $B$  é a mesma que entre  $B$  e  $A$ , tornando-a independente da ordem dos modelos. A métrica é sensível tanto às diferenças quanto às semelhanças, proporcionando uma visão equilibrada da concordância entre os modelos. A Similaridade de Jaccard pode ser aplicada independentemente do número de categorias ou da natureza dos dados, desde que as classificações possam ser representadas como conjuntos. O coeficiente de **Sorensen-Dice**, por sua vez, também mede a similaridade entre dois conjuntos, mas dá maior peso à interseção em relação à soma dos tamanhos dos conjuntos. A fórmula para dois conjuntos  $A$  e  $B$  é:

$$DSC(A, B) = \frac{2 \times |A \cap B|}{|A| + |B|}$$

Assim como o Jaccard, o Sorensen-Dice varia de 0 a 1, com valores maiores indicando maior similaridade. No contexto da comparação entre GPT-3.5 e GPT-4, essa métrica é especialmente útil para enfatizar as coincidências nas classificações, sendo particularmente sensível a mudanças na interseção dos conjuntos, o que pode ser relevante quando se deseja observar a consistência entre as versões dos modelos. A Figura 10 mostra o resultado da comparação das classificações feitas pelo GPT 3.5 e GPT 4, novamente considerando-se as classificações com pertinência "Alta" e "Média".

A distribuição do coeficiente de Jaccard apresenta uma concentração considerável em torno de valores baixos, com uma parte significativa da distribuição próxima de 0, e uma diminuição gradual em direção a valores maiores. Isso indica que, em muitos casos, há pouca interseção entre as classificações feitas pelos dois modelos, o que sugere uma baixa similaridade entre as categorias atribuídas pelo GPT-3.5 e GPT-4. O coeficiente de Sorensen-Dice também mostra uma tendência semelhante, com uma concentração em valores baixos. No entanto, devido à natureza da métrica, há uma ênfase ligeiramente maior em valores intermediários (entre 0.2 e 0.4). Mesmo assim, a distribuição reflete que a similaridade entre as classificações é geralmente baixa, com poucas coincidências substanciais entre as categorias atribuídas pelos dois modelos.

### **4.3 Análises de potenciais alterações na classificação dada mudança de ordem e exclusão de elementos da taxonomia**

Essa sessão faz uma breve avaliação de alguns cenários de mudança na ordem ou eliminação de elementos da taxonomia para avaliar se essas alterações comprometem a qualidade e resultado.

#### **4.3.1 Reclassificação com GPT 3.5 Turbo excluindo item 113 Problemas Funcionais**

Para verificar como é comportamento da classificação sem o item de maior frequência, tanto no GPT-4 quanto no GPT-3.5, o *prompt* foi modificado com a simples remoção da respectiva linha que continha o item 113 Problemas Funcionais. A Figura 11 mostra claramente outra distribuição de classificações bastante distinta.

Observa-se também que a qualidade da classificação deteriorou consideravelmente com o GPT 3.5. Pode-se notar que, mesmo sem o Item *113 Problemas Funcionais*, houve a classificação nesse item com a troca de descrição ou mesmo uma alucinação por parte do LLM. Os itens a) *113 Processo Travado* e; b) *113 Valor de lançamento incorreto* sequer constam da taxonomia. Já os itens a) *113 Dados Inconsistentes* e; b) *113 Versões não suportadas* constam na taxonomia mas não tem essa numeração. Outro fato interessante de se constatar é nova distribuição ordenada pela frequência, como mostra a Tabela 2. Como pode ser observado, não houve uma predominância tão substancial de um item, havendo agora uma distribuição um pouco mais equilibrada aos menos nos três primeiros itens. A observação de que a remoção de um item de maior frequência resultou em uma degradação na qualidade das respostas é notável.

#### **4.3.2 Reclassificação com GPT 4 Turbo excluindo item 113 Problemas Funcionais**

O mesmo exercício foi feito usando o GPT 4, mas com resultados diferentes. No GPT 4, conseguiu lidar melhor com a falta desse item de classificação. A Figura 12 mostra



Figura 11 – Distribuição Classificação dos Chamados pelo GPT 3.5 Sem Item 113 Problemas Funcionais

Item	Contagem
115 Versão de software desatualizada	312
133 Dados Duplicados	255
131 Falta de Dados	206
114 Versões não suportadas	162
116 Software obsoleto	130
132 Dados Corrompidos	122
111 Erros de codificação	116
121 Mau funcionamento da rede	100
112 Problemas de Integração	90
212 Versão do banco de dados	55

Tabela 2 – 10 Primeiros Elementos Contagem de Itens Classificados pelo GPT 3.5 sem o Item 113 Problemas Funcionais

melhor balanceamento das causas classificadas bem como não traz nenhuma alucinação e reporta um aumento significativo de apontamentos relacionados a dados.

#### 4.4 Análise das mensagem de explicação associada a taxonomia classificada

Uma questão importante é ter um entendimento se as mensagens de explicação de certa forma poderiam trazer informação relevante para o discernimento da classificação e serem fontes confiáveis para ações relacionadas a correção na atividade de suporte de sistemas. No *prompt*, intencionalmente, foi requisitada uma explicação do porquê determinada classificação pertencente a taxonomia era pertinente. Essa informação da mensagem gerada pelo LLM é aberta, não pertencendo a um conjunto fechado e, portanto, passível de variações. A primeira análise foi saber se de fato as explicações geradas teriam alguma coerência entre elas, dada as classificações feitas e já ilustradas nas seções anteriores. Para essa análise, foi usada a forma combinada de mineração de texto com visualização com grafos (VAZIRGIANNIS; MALLIAROS; NIKOLENTZOS, 2018), de forma a entender se existiam *clusters* que de alguma forma são coerentes com as classificações. Para essa tarefa, foram selecionadas todas as classificações que tiveram "Alta" ou "Média" pertinência, resultando em um número de 5.883 mensagens geradas dos 2.000 chamados submetidos ao GPT 4. Não foram consideradas as explicações quando a pertinência foi classificada como "Baixa" por que nota-se que estas contém com raras exceções, a mesma mensagem sem conteúdo significativo, como por exemplo: [*Baixo*, '4 Segurança', '422 Acesso Não Autorizado', 'Embora segurança seja sempre uma preocupação, não há nenhuma menção direta de atividades de acesso não autorizado ou problemas de segurança no documento.']]

Também houve a necessidade de remoção de palavras geradas nas mensagens que não agregavam ou mesmo distorciam a formação dos *clusters*, que foram: "documento", "menciona", "descreve", "mencionado", "ticket", "pode", "claramente", "caracteriza". É possível perceber a formação clara de agrupamento na Figura 13 mostrando uma consistência entre as mensagens, sendo possível usar essa técnica para também se obter um entendimento gerencial das grandes causas raízes dos problemas reatados nos chamado. É interessante observar também que foram criados agrupamentos distintos, mesmo que a



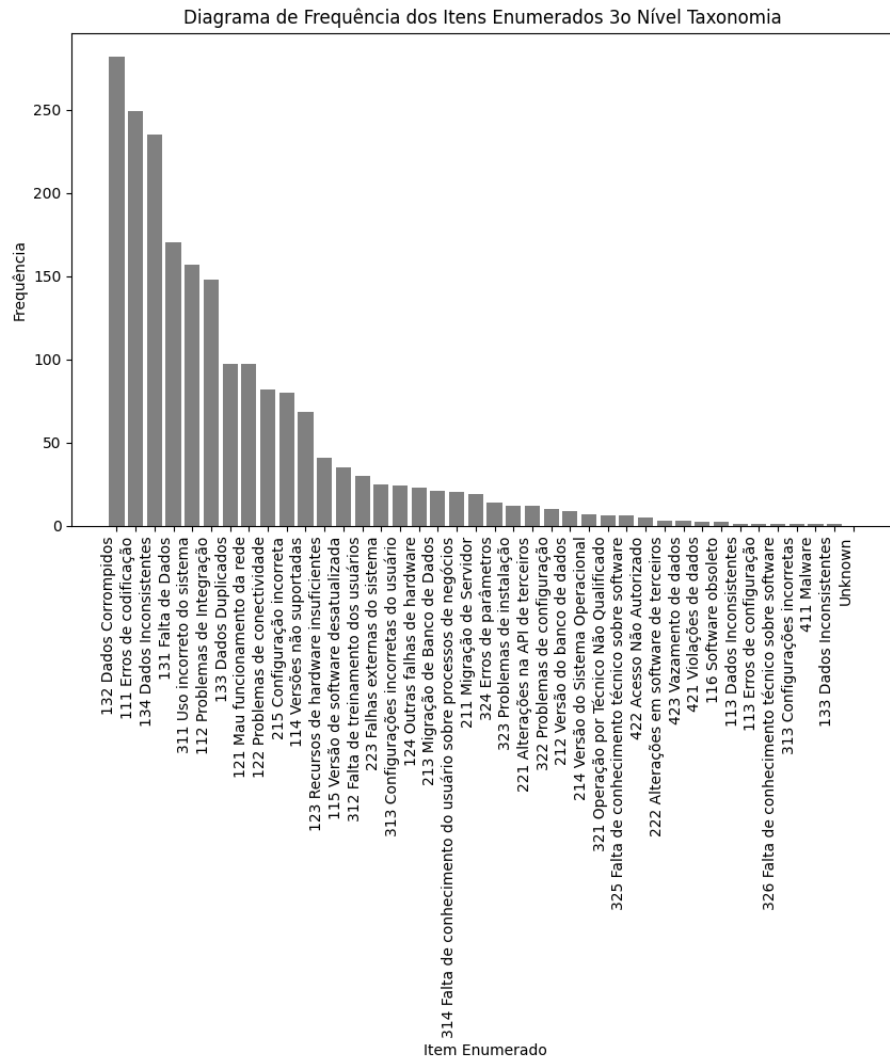


Figura 12 – Distribuição Classificação dos Chamados pelo GPT 4 Sem o Item 113 *Problemas Funcionais*

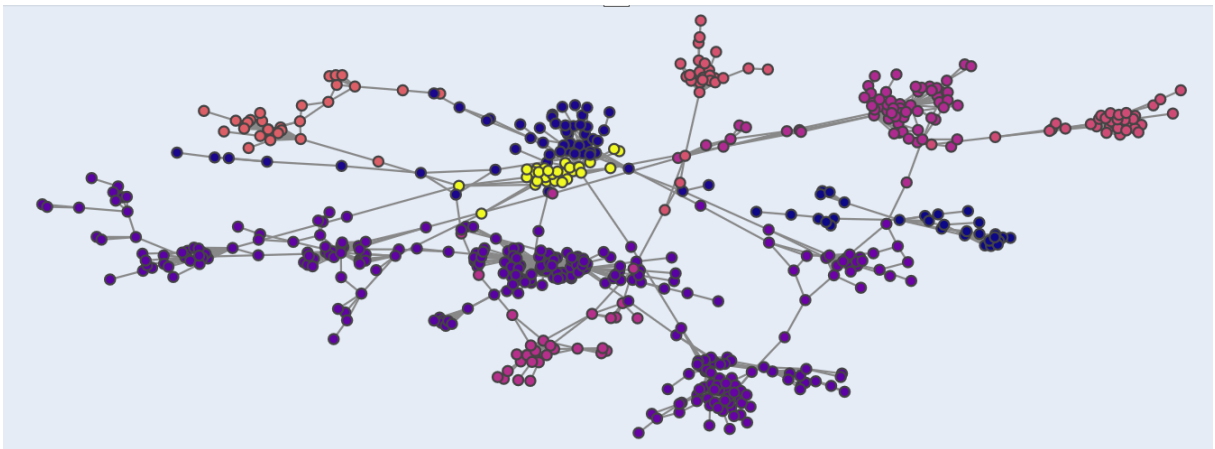


Figura 13 – Grafo de agrupamento com as mensagens de explicação da classificação da taxonomia

explicação contivesse menção a "funcional", como pode ser visto na Figura 14

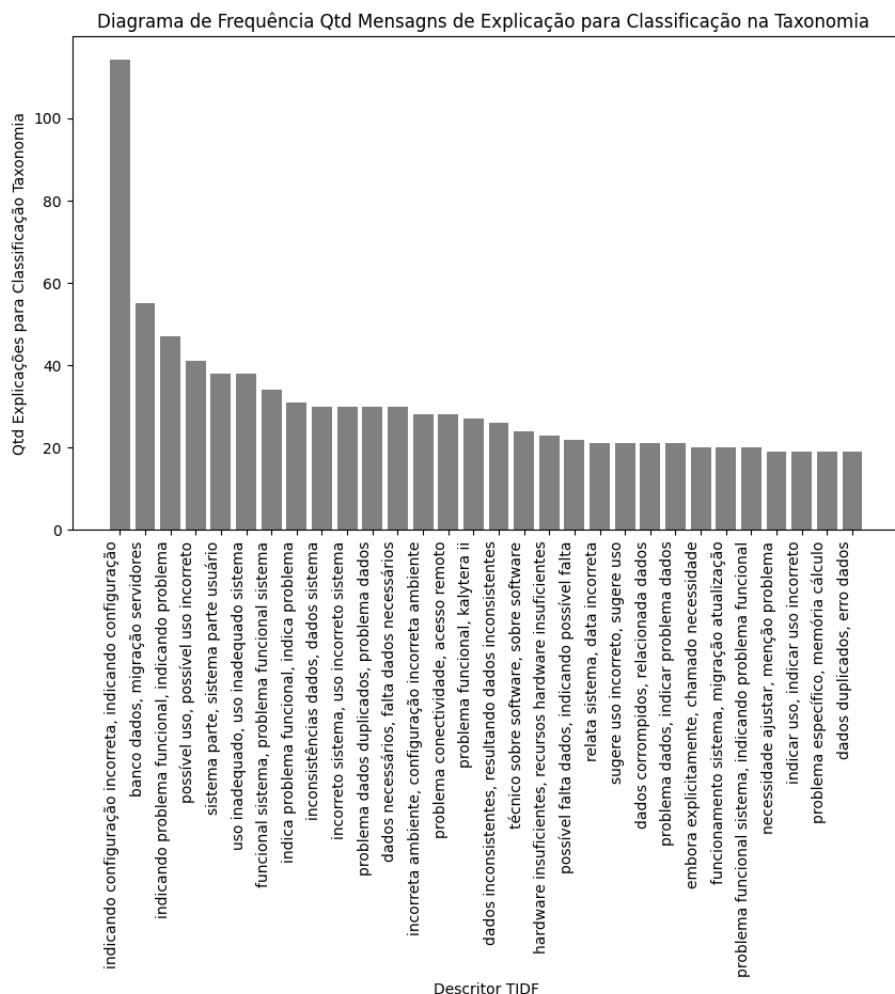


Figura 14 – Descrição dos 20 primeiros agrupamentos e número de mensagens

#### 4.4.1 Observações sobre as Mensagens geradas pelo GPT 3.5 e pelo GPT4

Outro aspecto relevante nessa investigação é entender se há diferenças fundamentais entre o GPT3.5 e GPT 4 na explicação dos porquês de classificação. É perceptível a diferença entre o GPT 3.5 e GPT 4 na capacidade de explicar melhor qual o motivo pelo qual um item da taxonomia foi escolhido para classificar o chamado. As tabelas a seguir ilustram três pontos observados de capacidade ampliada de inferência e leitura pelo GPT 4. Esse análise é de sobremaneira importante para os fins do estudo aqui descrito, dados os significativos esforços de evoluções de investigação e detecção da falha a ser resolvida em tempo hábil. Na Tabela 3 traz o chamado, e as duas classificações feitas. Claramente o chamado é um teste, sem qualquer menção a uma potencial falha. E as classificações feitas divergem, sendo que a classificação feita pelo GPT 4 é deliberadamente mais adequada para o chamado. Há de se ponderar o fato de que não houve uma explicação coerente com o tipo de chamado pelo GPT 4. Mesmo assim as classificações dadas pelo GPT 4, como "Baixa" foram mais

adequadas, tendo em mente que, para este trabalho, a classificação "Baixa" não foi usada para como um substrato de informação, e, pelos motivos vistos anteriormente, faz-se jus colocar que foi válida.

Chamado	Classificação GPT 3.5 e GPT 4
O ticket SD162759 Ticket de teste SSDENG NÃO EXCLUIR O ticket foi criado pela organização Banco Abc para o produto FundosPortal Auditor A descrição do ticket é Ticket de teste SSDENG NÃO EXCLUIR O ticket foi criado por Fabio Vintorin em 20231020 1639510 O status atual do ticket é Novo Resumo do tempo em cada status Novo 0 horas 00	<b>GPT 3.5:</b> [['Alto', '1 Técnica', '115 Versão de software desatualizada', 'A descrição do ticket menciona que o produto FundosPortal Auditor pode estar desatualizado, o que pode estar causando o mau funcionamento do sistema.'], ['Médio', '3 Humano', '313 Configurações incorretas do usuário', 'É possível que o problema relatado no ticket seja causado por configurações incorretas feitas pelos usuários. É importante investigar se houve alguma configuração incorreta que possa ter impactado no funcionamento do software.']] <b>GPT 4:</b> [['Baixo', '1 Técnica', '123 Recursos de hardware insuficientes', 'O documento não menciona nenhum problema relacionado a recursos de hardware.'], ['Baixo', '1 Técnica', '121 Mau funcionamento da rede', 'Não há indícios de mau funcionamento da rede no documento.'], ['Baixo', '2 Ambiente', '212 Versão do banco de dados', 'O ticket criado não faz referência a questões relacionadas à versão do banco de dados.'], ['Baixo', '3 Humano', '311 Uso incorreto do sistema', 'Nenhuma menção a uso incorreto do sistema é feita no documento.'], ['Baixo', '4 Segurança', '421 Violações de dados', 'Não há referências a violações de dados no documento.']]

Tabela 3 – Comparação mensagens GPT 3.5 e GPT 4: classificação errônea por falta de interpretação

A Tabela 4 traz outro aspecto interessante. A constatação do GPT 3.5 *GPT 3.5* [['Alto', '1 Técnica', '114 Versões não suportadas', 'O registro não está integrando ao SGR devido a problemas de integração com versões não suportadas dos componentes envolvidos']] fica insustentável quando se lê o conteúdo do chamado. É ululante que a falha foi devida a um problema de integração, apropriadamente colocada na taxonomia como item '1 Técnica', '112 Problemas de Integração'. O GPT 4 é francamente afirmativo nesse sentido, inclusive pontuando com 'Alto' a pertinência. Vale notar também que o GPT 4 aponta o *swagger* na resposta como um item explicatório da classificação. De fato, *swagger* é um sistema de integração de mercado. Ressalva-se aqui que não é claro se essa explanação foi devido ao entendimento de *swagger* como um sistema de integração ou se foi devido ao contexto da palavra.

O último exemplo sobre a capacidade de inferir do GPT 4 está na Tabela 5. Como pode ser constatado, a classificação na taxonomia para o GPT 4 foi em absoluto, mais adequada. Apesar das dos dois modelos conseguirem aprofundar no entendimento e ler a pilha de mensagens de falhas, como consta no chamado, o GPT 3.5 conclui que a classificação é: ['Médio', '4 Segurança', '412 Negação de Serviço (DoS)', 'Ao receber a mensagem "104 Connection reset by peer", pode indicar uma tentativa de Negação de Serviço (DoS) ao serviço de envio de PIX.']. Claramente, a pilha de mensagens de erro não condiz com essa classificação, somando-se ao fato de que foi colocado como 'Médio' e o último tópico do resultado da submissão. Ao passo que, para o GPT 4, a classificação foi: ['Alto', '1 Técnica', '122 Problemas de conectividade', 'O documento descreve uma série

Chamado	Classificação GPT 3.5 e GPT 4
O ticket SD162758 URGENTE INC0331256 Registro não está integrando O ticket foi criado pela organização Banco XPTO Brasil SA para o produto SGR Gerenciamento de Recebíveis A descrição do ticket é Boa noite no dia 25082023 tivemos um registro que não foi integrado ao SGR Tentamos realizar o procedimento via swagger sem sucesso A integração gravou na Tabela_X entretanto não gravou na Tabela_Y Solicitamos auxilio para entendermos a falha na integração Abrimos um call para tratarmos o assunto na segunda as 0900	<p><b>GPT 3.5</b> [['Alto','1 Técnica','114 Versões não suportadas','O registro não está integrando ao SGR devido a problemas de integração com versões não suportadas dos componentes envolvidos'], ['Médio','2 Ambiente','215 Configuração incorreta','A integração gravou na tabela errada devido a configurações incorretas'], ['Baixo','3 Humano','312 Falta de treinamento dos usuários','Pode ser que a falta de treinamento dos usuários tenha levado a erros na integração']]</p> <p><b>GPT 4</b> [['Alto','1 Técnica','112 Problemas de Integração','O documento relata que um registro não foi integrado ao SGR, mesmo após tentativa de integração via swagger'], ['Alto','1 Técnica','132 Dados Corrompidos','A descrição menciona que a integração gravou na tabela Tabela_X mas não gravou na tabela Tabela_Y, sugerindo possível corrupção de dados ou falha no processo'], ['Médio','3 Humano','311 Uso incorreto do sistema','Os usuários parecem estar realizando tentativas de integração sem sucesso, o que pode indicar um uso incorreto ou falta de entendimento do sistema']]</p>

Tabela 4 – Comparação mensagens GPT 3.5 e GPT 4: maior inferência sobre o texto

*de exceções relacionadas a problemas de conexão ao enviar requisições HTTP no serviço de envio de PIX, como conexão resetada pelo peer e problemas ao ler dados da conexão de transporte']*. Notadamente, além de ser mais assertiva e alinhada com o que se lê na pilha de mensagens, ela faz duas inferências interessantes, uma sobre o problema: a) Fala sobre uma série de exceções da conexão via HTTP; b) exemplifica a conexão *resetada* pelo *peer* e; c) Interpreta que houve erro no dados de transporte, que pode ser lido na pilha de mensagens. Outra corroboração é o fato de que, mesmo as palavras estando emendadas pelo advento do tratamento inicial de dados, houve o entendimento da pilha de mensagens. Com o intuito de eliminar caracteres tipo símbolo e reduzir o volume de dados, como descrito na subseção 2.5.1, em vários trechos houve junção de palavras, mas aparentemente, isso não foi uma restrição. O fato de conseguir entender a descrição humana do erro e a pilha de mensagens é promissor para ampliar a constatação da capacidade de interpretar registros de processamento, ou *log*, como descrito em (LE; ZHANG, 2023).

#### 4.4.2 Similaridade entre a classificação e a mensagem de explicação vinda do GPT 4

Outra averiguação feita foi a relação entre a classificação obtida *vis-a-vis* o texto de classificação. Foi considerado o uso de similaridade de cosseno por que a similaridade de cosseno mede o ângulo entre dois vetores, não a sua magnitude. Isso é útil porque os textos são de tamanhos diferentes e podem ainda assim ser muito semelhantes em termos de conteúdo. Essa tarefa foi feita usando *sentence transformer* com o model *distiluse-base-multilingual-cased-v1* (SANH *et al.*, 2019) que é uma versão destilada do *Universal Sentence Encoder*. O modelo é baseado na arquitetura do *transformer*, mas foi destilado para ser menor e mais rápido, mantendo um desempenho razoável e foi treinado em dados multilíngues oferecendo suporte ao português. O valor da coluna Similaridade é a similaridade de cosseno entre a classificação e a mensagem correspondentes de cada linha Esta similaridade indica o quão próximas são as duas representações textuais . A Figura 15 ilustra a aplicação da similaridade com a classificação composta do índice e seu

Chamado	Classificação GPT 3.5 e GPT 4
<p>O ticket SD162757 Erro acusado para serviço de envio de PIX O ticket foi criado pela organização Banco QWERT para o produto PIX Tree A descrição do ticket é O erro apresentado abaixo é apresentado com certa reincidência no serviço de envio de PIX Favor analisar</p> <pre> EvalPixExceptionsSPICryptoException Exceção ao enviar requisição à uri httpscompirsnfnetbr16422apivlin28127603 msgshhttpscompirsnfnetbr16422apivlin28127603msgs método POST timeout 20000n SystemNetHttpHttpRequestException An error occurred while sending the requestn SystemIOIOEx- ception Unable to read data from the transport connection Connection reset by peern SystemNetSocketsSocketException 104 Connection reset by peern End of inner exception stack trace n at SystemNetSocketsSocketAwaitableSoc- ketAsyncEventArgsThrowExceptionSocketError error CancellationToken cancellationToken in _srcSystemNet- SocketssrcSystemNetSocketsSocketTaskscsline 1107n__ at SystemNetSocketsSocketAwaitableSocket.AsyncEventArgs- GetResultInt16 token in _srcSystemNetSocketssrcSystem- NetSocketsSocketTaskscsline 1076n at SystemNetSecurityS- slStreamFillBufferAsync__InternalFillBufferAsync215_- 0TReadAdapterTReadAdapter adap ValueTask1 task Int32 min Int32 initial in _srcSystemNetSecuritysrcSys- temNetSecuritySslStreamImplementationcsline 1479n at SystemNetSecuritySslStreamReadAsyncInternalTReadAdap- terTReadAdapter adapter Memory1 buffer in _srcSystemNet- SecuritysrcSystemNetSecuritySslStreamImplementationcsline 1356n__ at SystemNetHttpHttpConnectionFillAsync in _srcSystemNetHttpsrcSystemNetHttpSocketsHttpHan- dlerHttpConnectioncsline 1437n at SystemNetHttpHttp- ConnectionReadNextResponseHeaderLineAsyncBoolean foldedHeadersAllowed in _srcSystemNetHttpsrcSystem- NetHttpSocketsHttpHandlerHttpConnectioncsline 1343n__ at SystemNetHttpHttpConnectionSendAsyncCoreHttpRequest- Message request CancellationToken cancellationToken in _srcSystemNetHttpsrcSystemNetHttpSocketsHttpHan- dlerHttpConnectioncsline 513n End of inner exception stack trace n at SystemNetHttpHttpConnectionSendAsync- CoreHttpRequestMessage request CancellationToken cancellationToken in _srcSystemNetHttpsrcSystemNetHttp- SocketsHttpHandlerHttpConnectioncsline 728n__ at SystemNetHttpHttpConnectionPoolSendWithNtConnectio- nAuthAsyncHttpConnection connection HttpRequestMessage request Boolean doRequestAuth CancellationToken cancellationToken in _srcSystemNetHttpsrcSystemNetHttp- SocketsHttpHandlerHttpConnectionPoolcsline 566n at SystemNetHttpHttpConnectionPoolSendWithRetryAsyn- cHttpRequestMessage request Boolean doRequestAuth CancellationToken cancellationToken in _srcSystemNetHttpsrcSystemNetHttp- SocketsHttpHandlerHttpConnectionPoolcsline 537n__ at SystemNetHttpRedirectHandlerSendAsyncHttpRequest- Message request CancellationToken cancellationToken in _srcSystemNetHttpsrcSystemNetHttpSocketsHttpHan- dlerRedirectHandlercsline 33n at SystemNetHttpDecom- pressionHandlerSendAsyncHttpRequestMessage request CancellationToken cancellationToken in _srcSystem- NetHttpsrcSystemNetHttpSocketsHttpHandlerDecompres- sionHandlercsline 57n__ at SystemNetHttpHttpClientFi- nishSendAsyncBufferedTask1 sendTask HttpRequestMessage request CancellationTokenSource cts Boolean disposeCts in _srcSystemNetHttpsrcSystemNetHttpHttpClientcsline 521n at EvalPixInternalHttpRequestGetResponseHttpClient client HttpRequestMessage requestMessage String methodn at EvalPixInternalHttpRequestSendRequestString method String uri NameValueCollection headers Byte contentn End of inner exception stack trace n at EvalPixInternalHttpRequestSendRe- questString method String uri NameValueCollection headers Byte contentn at EvalPixSPICryptoClientSendRequestString httpMethod String uri NameValueCollection requestHeaders Byte contentPropertiesSourceContextEvalPixSPICryptoClient O ticket foi criado por rogernerissunicombr em 20230825 2025280 O status atual do ticket é Novo </pre>	<p><b>GPT 3.5</b> [['Alto', '1 Técnica', '114 Versões não suporta- das', 'O problema apresentado está relacionado a uma exceção ao enviar uma requisição para uma URI, o que pode indicar incompatibilidade com a versão do software suportada.'], ['Médio', '4 Segurança', '412 Ne- gação de Serviço (DoS)', 'Ao receber a mensagem "104 Connection reset by peer", pode indicar uma tentativa de Negação de Serviço (DoS) ao serviço de envio de PIX.']]</p> <p><b>GPT 4</b> [['Alto', '1 Técnica', '122 Problemas de conecti- vidade', 'O documento descreve uma série de exceções relacionadas a problemas de conexão ao enviar requisi- ções HTTP no serviço de envio de PIX, como conexão resetada pelo peer e problemas ao ler dados da conexão de transporte'], ['Alto', '1 Técnica', '123 Recursos de hardware insuficientes', 'O documento menciona espe- cificamente a exceção "System.IO.IOException: Unable to read data from the transport connection", que pode estar associada a limitações ou problemas no hardware de rede responsável pela comunicação'], ['Alto', '2 Am- biente', '215 Configuração incorreta', 'A recorrência de erros e a especificação detalhada das exceções sugerem que pode haver problemas na configuração do ambi- ente como servidores, segurança SSL, ou configuração do próprio serviço PIX'], ['Médio', '3 Humano', '313 Configurações incorretas do usuário', 'O erro frequente e a natureza das exceções podem indicar também que há configurações sendo feitas de forma incorreta pelos administradores do sistema ou pelos operadores do ser- viço']]</p>

Tabela 5 – Comparação Mensagens GPT 3.5 e GPT 4: :Leitura e Interpretação do *stack* de erro

texto, como 113 Problema Funcional e sem o índice, restando apenas Problema Funcional. É perceptível a melhora na similaridade de cosseno quando a informação no número é removida da texto, como pode ser observado na Figura 15

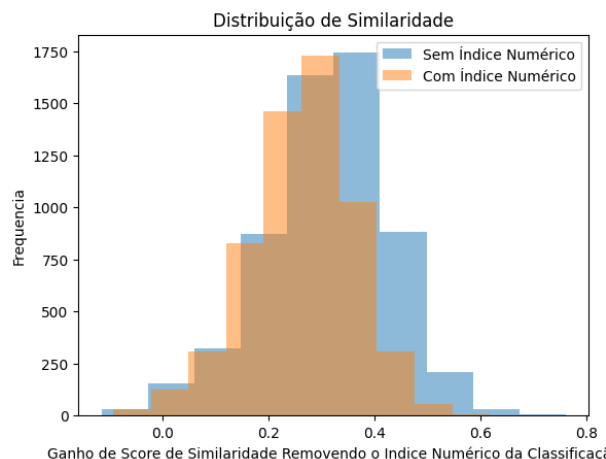


Figura 15 – Distribuição de Similaridade com e sem índice numérico

#### 4.4.3 Análise de itens com menor número de classificações

Não menos importante, essa subseção traz uma rápida análise das classificações que obtiveram as menores classificações para entender se essas classificações foram feitas de forma correta. O foco foram nas classificação de Violação e Vazamento de Dados e que é crítica do pontos de vista das leis de proteção de dados (CARVALHO *et al.*, ). Como pode ser visto, houve uma correta classificação nesse item e em uma análise entre os chamados classificados de forma igual entre GPT 3.5 e GPT 4 também houve coerência. Vale observar que esse problema, quando descrito em um chamado, se distingue dos demais de forma clara, promovendo uma classificação mais inequívoca.

#### 4.4.4 Relação entre o primeiro item e o segundo item classificados de uma mesma mensagem

A Figura 16 traz o mapa de calor, cruzando o primeiro item classificado e o segundo item classificado para se ter um panorama de como o GPT 4 está interpretando o texto de descrição do chamado e correlacionados as possíveis causas.

As combinações (113 Problemas Funcionais, 311 Uso incorreto do sistema), (134 Dados Inconsistentes, 311 Uso incorreto do sistema) e (113 Problemas Funcionais, 132 Dados Corrompidos indicam que o GPT 4, ao ler os textos, interpreta uma forte relação entre esses dois tipos de problemas. Também fica notório que 113 Problemas Funcionais e 134 Dados Inconsistentes aparecem com alta frequência isoladamente, o que indica uma classificação definitiva, sem outras causas. Auto-referências (ex: Problemas Funcionais com

Chamado	Classificação
<p>O ticket SD162585 Emissão de extrato financeiro para beneficiário com informações de outros beneficiários O ticket foi criado pela organização OOO para o produto PrevNetBenefício A descrição do ticket é Ao emitir o extrato financeiro de beneficiários o sistema não permite emitir individualmente e nesse caso é algo que não pode ocorrer pois são dados sensíveis expostos O ticket foi criado por Fabio Vintorin em 20230824 1722310 Comentário criado por marianerosaneosprevenciacombr em Thu Aug 24 172251 BRT 2023 EXTRATO MODELOpngthumbnail Comentário criado por Sinqia em Thu Aug 24 174359 BRT 2023 Prezada marianerosaneosprevenciacombr boa tarde</p> <p>Em análise a sua solicitação informamos que a geração do relatório atualmente trás a informação por matrícula considerando que o saldo é efetivamente vinculado a matrícula e a busca da geração do relatório é feito pela matrícula e não por beneficiário</p> <p>Para que seja alterado será necessário alterar a forma como o sistema faz essa leitura o que demanda uma customização que é tratada pela equipe de desenvolvimento dedicada da OOO</p> <p>colorde350bExiste atualmente uma demanda já sendo tratada pela equipe de desenvolvimento para corrigir esse mesmo relatório que está inclusive em etapa de homologação pela OOO</p> <p>SD149551</p> <p>Solicito que seja analisado na homologação da entrega do relatório essa observação da necessidade de alterar a leitura para geração por beneficiário</p> <p>Por favor validar a correçãoorientação se corrigiu o problema fechar o chamado e para que possamos aprimorar o nosso atendimento favor responder a pesquisa de satisfação disponibilizada após finalização do chamado A pesquisa refere-se ao atendimento específico desse chamado</p>	<p>[[['Alto', '4 Segurança', '423 Vazamento de dados', 'O documento mostra que houve um problema de exposição de dados sensíveis ao emitir o extrato financeiro de beneficiários.'], ['Alto', '1 Técnica', '113 Problemas Funcionais', 'O chamado relata que o sistema não permite a emissão de extratos financeiros individualmente, o que resulta na exposição de informações de outros beneficiários.'], ['Médio', '3 Humano', '311 Uso incorreto do sistema', 'A análise da solicitação indica que a geração do relatório é feita por matrícula ao invés de por beneficiário, o que pode ser considerado um uso inadequado do sistema para essa finalidade específica.']]</p>
<p>O ticket SD161812 Uso do canal seguro cipher na comunicação com MQ O ticket foi criado pela organização Banco QWERT Brasil para o produto CCO Contas Correntes A descrição do ticket é Uso do canal seguro cipher na comunicação com MQ O ticket foi criado por rodrigomarquesbcom em 20230818 1452530 O status atual do ticket é Em Andamento</p>	<p>[[['Médio', '1 Técnica', '112 Problemas de Integração', 'O ticket menciona o uso de canal seguro cipher na comunicação com MQ, indicando potenciais problemas na integração de sistemas que precisam ser analisados e corrigidos.'], ['Alto', '4 Segurança', '423 Vazamento de dados', 'O uso de um canal seguro cipher na comunicação com MQ reflete uma preocupação com a segurança e a prevenção de vazamento de dados, sendo altamente pertinente para garantir a integridade e confidencialidade das informações.']]</p>
<p>O ticket SD161052 Mascaramento das informações da agência e conta da porto na impressão do boleto O ticket foi criado pela organização Porto Seguro Consórcio para o produto Pleno no módulo Gestão de Consórcio keySQLGRkey A descrição do ticket é Objetivo do Projeto Mascaramento das informações da agência e conta da porto na impressão do boleto avulso e dos boletos gerados via canais URA APP e portal do cliente Oportunidade ou Problema Cenário Atual Atualmente é recebido pagamentos via transferência bancária e pix diretamente na conta da porto dificultando assim a baixa do pagamento na cota do cliente pois esses entram como DNI Visão de Negócio Cenário Proposto Precisamos que seja feito o mascaramento das informações da agência e conta nos boletos avulso gerado via pleno e os boletos gerados também via canais Este deve ser feito somente na impressão no campo agência código cedente no boleto avulso</p>	<p>[[['Médio', '1 Técnica', '113 Problemas Funcionais', 'A necessidade de mascarar informações na impressão dos boletos indica um problema funcional no software de geração de boletos.'], ['Alto', '4 Segurança', '423 Vazamento de dados', 'O mascaramento das informações da agência e conta nas impressões dos boletos visa evitar possíveis vazamentos de dados sensíveis.'], ['Médio', '1 Técnica', '114 Versões não suportadas', 'O status atual indica que estão trabalhando em uma atualização, o que pode sugerir que as versões anteriores do software não suportam essa funcionalidade de mascaramento.']]</p>

Tabela 6 – Classificação de Vazamento de Dados

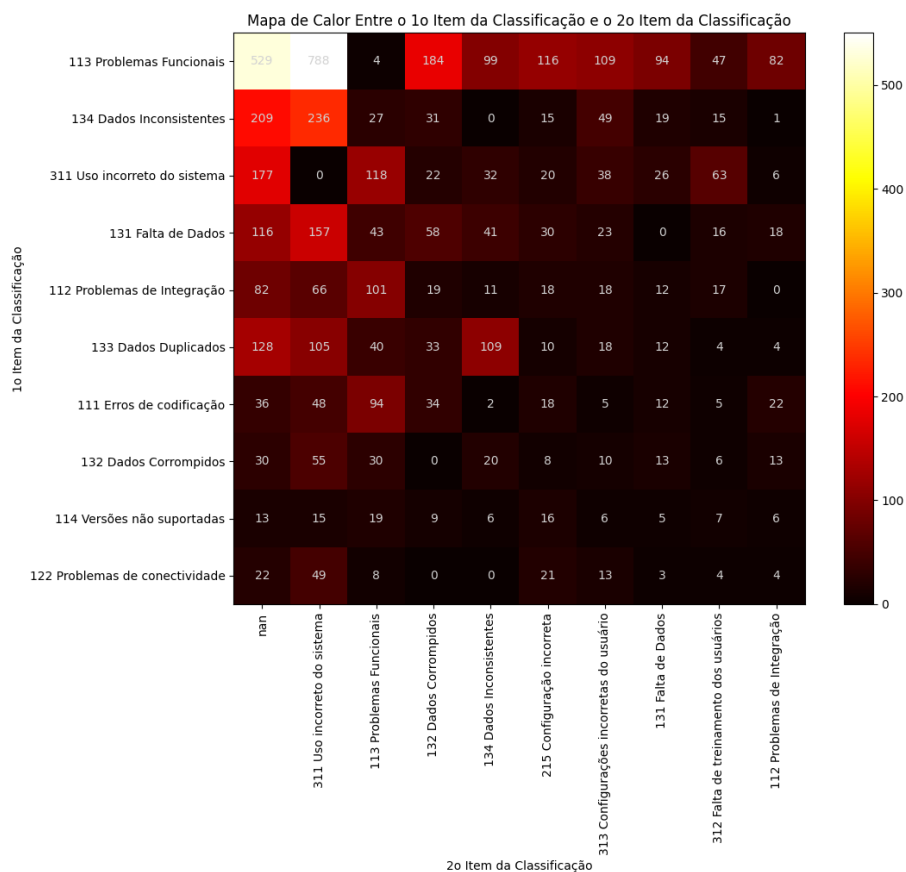


Figura 16 – Mapa de Calor das duas classificações para um texto de descrição de chamado

Problemas Funcionais) são raras ou inexistentes (4 ocorrências), o que pode indicar que a classificação tem sido assertiva ao elencar o item da taxonomia.



## 5 ANÁLISE QUANTITATIVA COM APLICAÇÃO DO MODELO BERT, *FINE-TUNNING* E PROBABILIDADES ASSOCIADAS

O objetivo da aplicação do modelo BERT é buscar uma forma de quantificar a assertividade da classificação bem como ter uma alternativa que abrisse possibilidade avaliação de custos mais comedidos a uma contratação de um LLM. Dada a análise qualitativa feita, com resultado melhor apresentado pelo GPT 4, entendeu-se que as mensagens classificadas pelo GPT 4 poderiam ser consideradas como entrada para o *fine-tuning* do BERT. Essa sessão está organizada pelos ciclos de treinamento feitos, que no total soma-se 6 ciclos. Foram feitas variações de épocas, quantidades de mensagens, ajuste em hiper-parâmetros e uso de dados sintéticos para ter um direcionamento sobre qual é a melhor abordagem para o treinamento do BERT para esse contexto e domínio de dados.

### 5.1 Lotes classificação

O primeiro passo foi analisar se faria sentido usar o primeiro lote de dois mil chamados classificados e entender como seria a composição dos chamados do ponto de vista de palavras e tamanho em caracteres desses chamados. Essa avaliação é importante para entender se, ao aplicar o modelo BERT, que tem limites de entrada de 512 palavras, haverá cortes relevantes que poderiam comprometer o resultado.

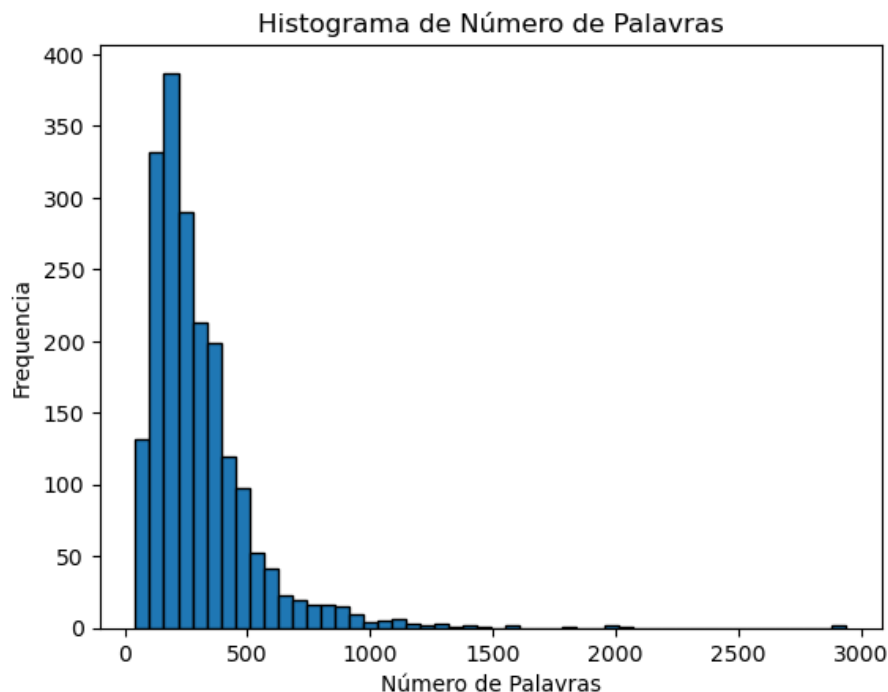


Figura 17 – Histograma Número de Palavras nos Chamados de Sustentação 1o Lote com 2.000 chamados

Na Tabela 5.1 é possível notar que em grande medida os chamados estão dentro dos limites para a classificação através do uso do modelo BERT. Precisamente, são 1.777 chamados ou 88,85% dos chamados são passíveis de serem submetidos ao modelo em sua integridade. Tratando do limite de 256 palavras, neste caso, existem 1.055 chamados, 52,75%. Por fim, sendo 128 palavras o limite do modelo BERT, tem-se 255 chamados, ou 12,75% do total. Outra constatação foi o desbalanceamento das classificações, ilustrado nas sessões anteriores, o que suscitou numa indagação sobre a possibilidade de usar dados sintéticos. Observando-se esse perfil do primeiro lote de classificação, objeto das análises qualitativas, é possível constatar que faz sentido ter outro grupo de chamados classificados considerando apenas a descrição do chamado, e com isso ter um perfil de tamanho de palavras mais adequado para encaixar no limite de 128 palavras, o que proporciona um esforço computacional menor, sem perder a qualidade da classificação

Medida	Valor
Qtd Chamados	2.000
Média Palavras	305,27
Desvio Padrão	230,21
Mínimo	44,00
25%	166,00
50%	244,00
75%	372,00
Máximo	2.940,00

Tabela 7 – Contagem de palavras 2.000 chamados do 1o lote

### 5.1.1 O segundo lote de Classificação

Com foco na submissão para o BERT, foi feito um segundo lote de classificação, inicialmente com 4.835 chamados foram classificados, vindos de uma segunda extração do Jira, sendo o total de chamados 78.680. Nesse submissão para o GPT 4 foi usado somente o campo de descrição do chamado. Isso resultou em uma amostra mais interessante para o BERT, desde o ponto de vista de tamanho da mensagem a serem submetidas, que ficou abaixo das 128 palavras. A Figura 18 mostra a distribuição das classificações, que ficou um pouco mais balanceada nas categorias subsequentes ao item "113 Problemas Funcionais", como pode ser vista na Tabela 32 do Anexo.

## 5.2 O Modelo BERT

### 5.2.1 Considerações sobre *fine-tuning* do modelo BERT com dados desbalanceados

Como mostrado nas Figura 7, os dados classificados e serem usado para o modelo são claramente desbalanceados. (BUDA; MAKI; MAZUROWSKI, 2018) e (HAIXIANG *et al.*, 2017) discutem sobremaneira forma de lidar com o problema de dados desbalanceados.

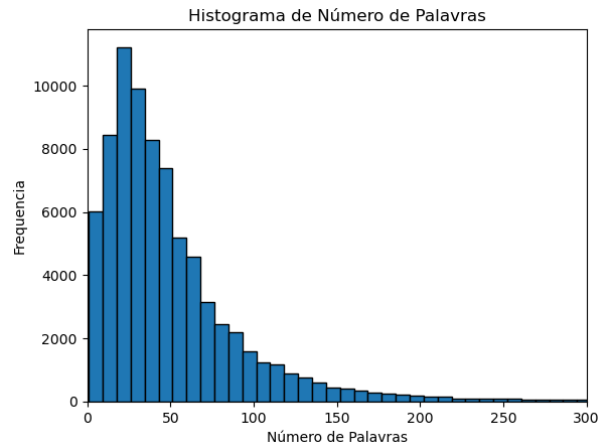


Figura 18 – Histograma de número de palavras segundo lote de classificação, com 78.680 chamados

(MADABUSHI; KOCHKINA; CASTELLE, 2020) oferece uma forma de atribuir pesos diferentes às diferentes classes, dando mais importância às classes minoritárias, e indica que, embora o modelo BERT tenha capacidade para lidar com classes desequilibradas sem a necessidade de dados adicionais para aumento, sua capacidade de generalização é limitada quando há uma disparidade significativa entre os dados de treinamento e de teste. (LI *et al.*, 2023) ressalta que o desempenho dos modelos treinados em dados sintéticos diminui significativamente em tarefas de classificação que envolvem altos níveis de subjetividade, bem como em instâncias onde a subjetividade é mais pronunciada. Dadas as características do domínio de informação dos chamados, que como já explanado contém alto grau de subjetividade, optou-se por de fato gerar um novo conjunto maior de chamados classificados.

### 5.2.2 Análise do modelo BERT usado

O modelo BERT `multi_cased_L-12_H-768_A-12`<sup>1</sup> é uma variação específica do modelo BERT original, com características e capacidades únicas que se fazem, para esse trabalho, necessários para o domínio de dados, o qual tem ampla diversidade de textos, estilos de escrita, tamanho, variedade de dados e assuntos. A seguir, uma breve descrição dos termos que definem o nome do modelo:

- **multi\_cased:** Indica que o modelo foi treinado em um corpus de texto que inclui tanto letras maiúsculas quanto minúsculas, tornando-o mais adaptável a diferentes estilos de escrita.
- **L-12:** Representa o número de camadas (*layers*) de transformadores no modelo. Neste caso, são 12 camadas, o que indica uma profundidade considerável na compreensão

<sup>1</sup> `bert_multi_cased_L-12_H-768_A-12` in TF2 SavedModel format · Issue #493 · tensorflow/hub · GitHub

do texto.

- **H-768:** Refere-se ao tamanho do vetor oculto (*hidden size*) de cada camada, que é de 768 dimensões. Um vetor maior geralmente permite capturar informações mais complexas e nuances do texto.
- **A-12:** Indica o número de cabeças de atenção (*attention heads*) em cada camada. Com 12 cabeças de atenção, o modelo pode capturar diferentes tipos de relações entre as palavras.

O modelo BERT multi\_cased\_L-12\_H-768\_A-12, por ser uma variante do BERT, herda as principais capacidades do modelo original:

- **Compreensão profunda do contexto:** Graças às suas múltiplas camadas e mecanismos de atenção, o modelo consegue entender o contexto de uma palavra com base em todas as outras palavras na frase, sua característica bidirecional.
- **Adaptação a diversas tarefas:** Assim como o BERT original, este modelo pode ser facilmente adaptado a diversas tarefas de PLN através de *fine-tuning*.
- **Gerenciamento de ambiguidades:** O modelo é capaz de lidar com palavras que possuem múltiplos significados, considerando o contexto em que elas são utilizadas.
- **Capacidade de lidar com diferentes estilos de escrita:** Devido ao treinamento em um corpus com letras maiúsculas e minúsculas, o modelo é mais robusto em relação a diferentes estilos de escrita.

### 5.2.3 *Fine-tuning* do modelo BERT para tratar dos chamados

Para executar o *fine-tuning* do modelo BERT foi definido o comprimento máximo das palavras de texto <sup>2</sup>, bem em linha com o perfil de palavras do conjunto de dados. Textos mais longos devem ser truncados, e textos mais curtos são preenchidos com *padding*. O limite máximo de palavras únicas que são consideradas no vocabulário foi ajustado para 10.000. Palavras raras além deste limite são ignoradas. Também foi usada a política "One Policy" proposta em (SMITH, 2017) que envolve aumentar gradualmente a taxa de aprendizagem de um valor inicial pequeno para um valor máximo e depois diminuí-la gradualmente até um valor final pequeno ao longo de um único ciclo de formação. Foi demonstrado que este ajuste cíclico da taxa de aprendizagem melhora a velocidade de convergência e o desempenho de generalização em muitas arquiteturas de redes neurais. (SMITH; TOPIN, 2019) discute que as taxas de aprendizagem devem ser significativamente mais altas do que as normalmente usadas para uma convergência mais rápida. Tradicionalmente, eram preferidas taxas de aprendizagem mais baixas para evitar instabilidade e *overfitting*. No

---

<sup>2</sup> ktrain.text API documentation

entanto o trabalho sobre super convergência mostraram que o uso de taxas de aprendizagem maiores pode, na verdade, acelerar o treinamento e melhorar o desempenho.

#### 5.2.4 Pré-processamento das mensagens para submissão ao BERT

Para submeter ao BERT, foram eliminadas todas os chamados que por ventura não continham texto e classificação relevante. É natural na base de chamados serem encontrados chamados sem mensagem explícita, chamados de teste, mensagem nulas e chamados com baixíssimo nível de descrição. Ao serem submetidos o LLM, a classificação retornada é incoerente, mas o GPT 4 conseguiu sempre nesses casos colocar a pertinência como "Baixa", o que permitiu um filtro objetivo e consistente. O segundo estágio de pré-processamento foi balancear o conjunto de mensagem para que as classificações ficassem relativamente uniformes nas suas quantidades. A Tabela 32 mostra a ordem e a quantidade de classificação por item e a Tabela 8 mostra o corte para os dez primeiros itens de maior classificação bem como a redução aleatória dos chamados classificados como 113 Problemas Funcionais e a eliminação do número identificador. O número foi removido pois, na análise quantitativa descrita anteriormente, percebeu-se que, ao calcular a similaridade de cosseno do texto da classificação com a mensagem de explicação associada, quando o número é removido, a similaridade melhora, como mostra a Figura 15.

Item	Quantidade
Problemas Funcionais	512
Uso incorreto do sistema	440
Dados Inconsistentes	429
Falta de Dados	386
Problemas de Integração	323
Dados Duplicados	271
Erros de codificação	213
Dados Corrompidos	166
Versões não suportadas	128
Problemas de conectividade	118

Tabela 8 – Dados Balanceados para submissão aprendizado BERT

### 5.3 Ciclo 1 de submissão ao modelo BERT, com dados originais

Como mostrado na Tabela 8, são aplicadas dez categorias para a classificação. O treinamento tem 10 épocas que significa uma passagem completa por todos os dados de treinamento, com 66 lotes de dados cada uma delas. Os dados obtidos apresentam o desempenho do modelo BERT ao longo das 10 épocas de treinamento. As métricas de treinamento chave são:

- **Perda** =  $-\frac{1}{N} \left( \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c}) \right)$  Erro médio do modelo durante o treinamento.

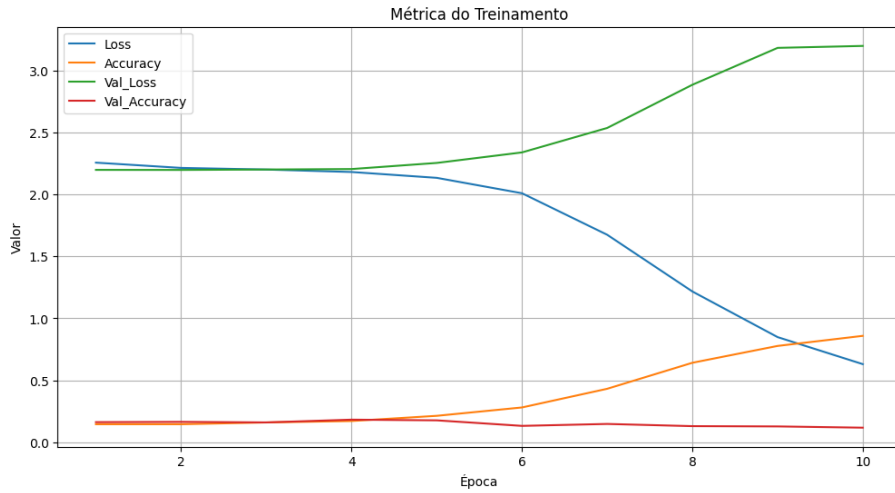


Figura 19 – Métricas treinamento *fine\_tunning do BERT*

- **Acurácia** =  $\frac{1}{N} \left( \sum_{i=1}^N \mathbf{1}(\hat{y}_i = y_i) \right)$  Porcentagem de exemplos classificados corretamente durante o treinamento.
- **Perda Validação** =  $-\frac{1}{M} \left( \sum_{i=1}^M \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c}) \right)$  Erro médio do modelo em um conjunto de dados de validação separado.
- **Acurácia Validação** =  $\frac{1}{M} \left( \sum_{i=1}^M \mathbf{1}(\hat{y}_i = y_i) \right)$  Porcentagem de exemplos classificados corretamente no conjunto de validação.

Época	Perda	Acurácia	Val Perda	Val Acurácia
1	2.2565	0.1450	2.1981	0.1618
2	2.2141	0.1450	2.1977	0.1641
3	2.2008	0.1593	2.2010	0.1596
4	2.1810	0.1708	2.2048	0.1819
5	2.1338	0.2129	2.2542	0.1763
6	2.0095	0.2804	2.3391	0.1317
7	1.6743	0.4311	2.5365	0.1473
8	1.2164	0.6411	2.8853	0.1295
9	0.8481	0.7770	3.1830	0.1272
10	0.6299	0.8584	3.1987	0.1172

### Análise do *Overfitting*:

*Overfitting* (sobreajuste) ocorre quando um modelo se ajusta demais aos dados de treinamento, perdendo a capacidade de generalizar para novos dados. Para identificar o *overfitting*, é observada a divergência entre as métricas de treinamento e validação.

- **Decréscimo gradual:** idealmente, tanto a *Loss* quanto a *Val\_Loss* devem diminuir

gradualmente ao longo das épocas de treinamento. Isso indica que o modelo está aprendendo e se ajustando aos dados.

- **Overfitting:** se a Loss continuar diminuindo, mas a Val\_Loss começar a aumentar, isso é um sinal de *overfitting*. O modelo está memorizando os dados de treinamento em vez de generalizar para novos dados.
- **Underfitting:** se ambas as métricas permanecerem altas, o modelo pode estar *underfitting* (subajuste), ou seja, não está aprendendo as características suficientes dos dados.

**Accuracy e Val\_Accuracy:** a *Accuracy* (acurácia) mede a proporção de exemplos que o modelo classifica corretamente durante o treinamento. A *Val\_Accuracy* mede a acurácia do modelo no conjunto de validação.

- **Aumento gradual:** a *Accuracy* e a *Val\_Accuracy* devem aumentar à medida que o modelo é treinado. No entanto, a *Val\_Accuracy* pode não aumentar tanto quanto a *Accuracy*, especialmente em casos de *overfitting*.
- **Lacuna entre Accuracy e Val\_Accuracy:** Uma grande lacuna entre a *Accuracy* e a *Val\_Accuracy* indica *overfitting*.

### 5.3.1 Notas sobre o primeiro treinamento

- **Loss:** diminui consistentemente ao longo das épocas, indicando que o modelo está aprendendo.
- **Accuracy:** aumenta significativamente ao longo do treinamento, o que é esperado.
- **Val\_Loss:** começa a aumentar a partir da época 5, enquanto a *Val\_Accuracy* começa a diminuir a partir da época 6.

O modelo evolui muito bem nos dados de treinamento, mas não generaliza bem para novos dados. Com base nas métricas, é evidente que o modelo está sofrendo de *overfitting* a partir da época 5. A partir desse ponto, o modelo começa a memorizar o conjunto de treinamento, perdendo a capacidade de generalizar para novos dados. Para traçar as ações possíveis, (HOWARD; RUDER, 2018) propõem um método para fazer *fine tuning* universal e discute o *fine-tuning* de modelos de linguagem pré-treinados para tarefas de classificação de texto, abordando técnicas de regularização, como *dropout* e *early stopping*. (RADFORD *et al.*, 2019) , apresenta o modelo GPT-2 e discute algumas das técnicas de regularização utilizadas, como *dropout* e Regularização L2. (IOFFE; SZEGEDY, 2015) coloca que Normalização Batch potencialmente eliminar a necessidade de *dropout*, que é

uma técnica preconizada em (SRIVASTAVA *et al.*, 2014). Para este trabalho, podem ser aplicadas quatro técnicas, isoladamente ou combinadas:

1. **Parar o treinamento precocemente:** a melhor época para interromper o treinamento seria antes da época 5, quando a *Val\_Loss* ainda estiver diminuindo.
2. **Regularização:** implementar técnicas como *dropout* ou Regularização L1/L2 para reduzir a complexidade do modelo e evitar *overfitting*.
3. **Aumentar o conjunto de dados:** um conjunto de dados maior pode ajudar o modelo a generalizar melhor.
4. **Ajustar hiper-parâmetros:** experimentar com diferentes valores de hiper-parâmetros, como a taxa de aprendizado, pode melhorar o desempenho do modelo.

Classe	Precisão	Revocação	Pont. F1	Suporte
0	0.04	0.02	0.03	45
1	0.12	0.09	0.10	82
2	0.10	0.14	0.12	113
3	0.07	0.06	0.06	68
4	0.12	0.16	0.14	126
5	0.18	0.15	0.17	170
6	0.10	0.18	0.13	84
7	0.00	0.00	0.00	34
8	0.12	0.11	0.12	133
9	0.11	0.02	0.04	41
Acurácia	0.12			
Média Macro	0.10	0.09	0.09	896
Média Ponderada	0.12	0.12	0.11	896

Tabela 9 – Precisão, Revocação, Pontuação F1 e Suporte para Cada Classe

### 5.3.2 Observação inicial sobre predição e quantificação da probabilidade

Um dos grandes objetivos para a aplicação do modelo BERT neste trabalho foi buscar uma forma quantitativa de mensurar a classificação e que tem três itens relevantes nesse contexto:

- **Confiança do modelo:** as probabilidades retornadas indicam o nível de confiança que o modelo tem em relação a cada classe. Valores mais próximos de 1 indicam maior confiança. O número objetivo não foi possível com o aplicação dos LLM;
- **Aplicação neste contexto específico:** no trabalho de investigação de problemas em produção, é importante não apenas saber a classe prevista, mas também o nível de confiança (probabilidade) associado a essa previsão;



- **Limiar Personalizado:** é possível com isso definir um limiar de confiança personalizado para aceitar ou rejeitar previsões. Por exemplo, se a probabilidade de uma classe não atingir um certo nível, pode não fazer sentido tomar uma decisão automática.

Um exemplo: para o texto de chamado "*O sistema está gerando cálculo de movimentos retroativos ao clonar projeção produto por terem a mesma base de partida de cenário 30112022 porém para projetar o mês seguinte o sistema deve desprezar as movimentações anteriores e iniciar o processamento da base realizada cargas uma vez que sempre será clonada a opção projeção produto*", o resultado de classificação com as probabilidades está na Tabela 10

Descrição	Probabilidade
Dados Duplicados	0.408267
Problemas Funcionais	0.311407
Dados Inconsistentes	0.135321
Uso incorreto do sistema	0.057727
Dados Corrompidos	0.032641
Versões não suportadas	0.015339
Falta de Dados	0.013353
Problemas de Integração	0.012462
Problemas de conectividade	0.011325
Erros de codificação	0.002158

Tabela 10 – Classificação e suas probabilidades associadas

#### 5.4 Agrupamento das classificações com aumento dos dados para um conjunto menor de classes e novo treinamento (Ciclo 2)

O Ciclo 2 de *fine-tuning* buscou aumentar o conjunto de dados através da consolidação de classificações para um nível intermediário da taxonomia. Ao analisar os dados qualitativamente, comparando como o LLM classificou com o resultado da classificação pelo BERT, é possível perceber as divergências, como mostra a Tabela 11 que contém: a) mensagem do chamado; b) o que o LLM classificou; c) o que o BERT classificou (as 3 primeiras classes com maior probabilidade) e d) Probabilidades, coluna onde são listadas as probabilidades em ordem decrescente, cujas três primeiras correspondem às três classes que estão listadas na coluna BERT. Além disso, outro objetivo é o aumento de dados para um conjunto menor de classes, para tratar *overfitting*.

Apesar da divergência aparente, é possível notar que de uma certa forma, como semântica análoga, existem quatro grandes categorias que pode ser usadas para ter mais exemplos de classificação para o *fine tuning*, a saber: Dados, Funcional, Integração, Uso e Outros. Abaixo, segue a aglutinação das classificações feitas em Nível 3 para este

Chamado	LLM	BERT	Probabilidade
<p>Durante o início do processo de integração do arquivo Imbarq no ambiente de produção observamos a ausência de alguns jobs no SQCONN Por isso gostaria de tirar algumas dúvidas para realizar a criação desses jobs</p> <p>As informações de integração dos DEMAIS ATIVOS e SWAPCETIP também são extraídas do arquivo Imbarq</p> <p>Precisamos de ajuda para realizar a criação dos jobs de Demais Ativos e SWAPCETIP com os parâmetros corretos</p>	Problemas de Integração, Erros de parâmetros	Problemas Funcionais, Dados Inconsistentes, Falta de Dados	0.7174, 0.1580, 0.0484, 0.0279, 0.0247, 0.0080, 0.0070, 0.0053, 0.0026, 0.0009
<p>Pessoal</p> <p>Inicialmente tratamos essa questão através do chamado SD165131 porém o mesmo foi finalizado por inatividade após a Sinqia sinalizar que não tinha sido possível simular o erro O time refez os testes e identificou que ainda existe erro quando o processo é feito com a flag de Vinculação automática ativa</p> <p>Anexo evidências</p>	Problemas Funcionais, Uso incorreto do sistema	Problemas de Integração, Falta de Dados, Erros de codificação	0.5505, 0.1583, 0.1249, 0.0618, 0.0524, 0.0201, 0.0143, 0.0080, 0.0072, 0.0024
<p>Tivemos problema para gerar o EFDReinf da competência de Abril2022 Só temos um prestador de serviço que tem o evento R2010 que é a Serviceline Fizemos o pagamento dos serviços prestados em abril2022 em 04052022 mas no lançamento a competência é 29042022 ao tentar gerar as informações do evento R2010 para o mês 042022 apareceu a mensagem anexa</p> <p>Favor verificar a inconsistência o mais breve possível pois temos até 10052022 para enviar o arquivo com a obrigação Estamos disponibilizando uma base no ftp</p>	Dados Inconsistentes, Uso incorreto do sistema	Problemas de Integração, Uso incorreto do sistema, Dados Inconsistentes	0.3078, 0.2399, 0.1365, 0.0820, 0.0746, 0.0509, 0.0410, 0.0288, 0.0245, 0.0139

Tabela 11 – Tabela de Comparação LLM e BERT e Probabilidades Retornadas

nível intermediário e a Tabela 12 traz as quantidades de mensagens rotuladas para o *fine tuning*.

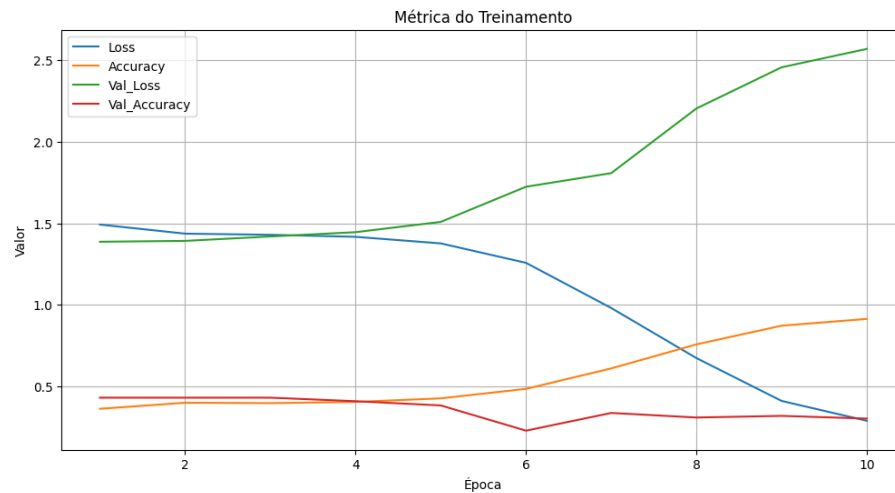
- **Dados** = ['Dados Corrompidos', 'Dados Duplicados', 'Dados Inconsistentes', 'Falta de Dados']
- **Funcional** = ['Erros de codificação', 'Problemas Funcionais', 'Configuração incorreta']
- **Integração** = ['Problemas de Integração', 'Problemas de conectividade', 'Mau funcionamento da rede']
- **Uso** = ['Uso incorreto do sistema', 'Falta de conhecimento do usuário sobre processos de negócio', 'Configurações incorretas do usuário', 'Falta de treinamento dos usuários']
- **Outros** = Todos as demais classificações feitas

#### 5.4.1 Resultados do *fine-tuning* com as classificações consolidadas

Foram submetidas 4.630 chamados e descartados 205, por questões de chamados classificados como Baixa pertinência ou campo nulo. É possível observar que o problema

Categoria Consolidada	Quantidade
Funcional	1902
Dados	1252
Uso	665
Integração	476
Outros	335

Tabela 12 – Categorias Consolidadas e Quantidades

Figura 20 – Métricas de *fine-tuning* do BERT para dados consolidados

de *overfitting* também acontece com as mesmas características. A consolidação de classes buscou ampliar o conjunto de dados para treinamento em cada classe, uma das ações listadas na subseção anterior. Contudo, é possível perceber que o comportamento das variáveis *Loss*, *Val\_Loss*, *Accuracy* e *Val\_Accuracy* é bastante parecido com o primeiro ciclo de treinamento.

Observando também a Tabela 13 é notório avaliar que o desempenho geral foi ruim. A acurácia geral de 0,30 e a média ponderada de F1 de 0,30 indicam um desempenho abaixo do esperado. Isso sugere que o modelo está tendo dificuldades em classificar corretamente as amostras. Há desequilíbrio entre as classes, onde as classes 2 e 3 possuem um número muito menor de amostras (suporte) em comparação com as outras classes. Isso pode estar afetando o desempenho do modelo, especialmente para essas classes minoritárias. O baixo F1 para todas as classes indica que o modelo está tendo dificuldades tanto em identificar corretamente os exemplos positivos (revocação baixo) quanto em evitar classificar exemplos negativos como positivos (precisão baixa). A diferença entre a média macro e a média ponderada corroboram que modelo está tendo um desempenho significativamente pior nas classes minoritárias.

### 5.4.2 Indicadores de Precisão, revocação, F1 e Suporte

A Tabela 13 apresenta uma avaliação detalhada do modelo BERT treinado com dados agrupados, utilizando métricas de precisão, revocação e F1 para cada classe de saída.

**Precisão:** A proporção de exemplos positivos classificados corretamente entre todos os exemplos classificados como positivos. **Revocação:** A proporção de exemplos positivos classificados corretamente entre todos os exemplos positivos reais. **F1:** A média harmônica de precisão e revocação, oferecendo um bom indicador geral do desempenho do classificador. As métricas de precisão para esse ciclo de *fine-tuning* claramente mostram que o modelo não teve sucesso em aprender a classificar.

Classe	Precisão	Revocação	Pont. F1	Suporte
0	0.28	0.33	0.30	378
1	0.43	0.43	0.43	598
2	0.10	0.08	0.09	139
3	0.03	0.03	0.03	78
4	0.15	0.13	0.14	196
<b>Acurácia</b>	0.30			
<b>Média Macro</b>	0.20	0.20	0.20	1389
<b>Média Pond.</b>	0.29	0.30	0.30	1389

Tabela 13 – Relatório de classificação *fine-tuning* com dados agrupados

Para comparar preliminarmente as probabilidades auferidas no caso das dez classes e depois no caso consolidado em cinco classes, para o mesmo texto submetido com o modelo BERT re-treinado "*O sistema está gerando cálculo de movimentos retroativos ao clonar projeção produto por terem a mesma base de partida de cenário 30112022 porém para projetar o mês seguinte o sistema deve desprezar as movimentações anteriores e iniciar o processamento da base realizada cargas uma vez que sempre será clonada a opção projeção produto*" agora com as classes consolidadas, as distribuição de probabilidade é mostarda na Tabela 14, ficando mais manifesta a classificação relacionada a Dados.

Classes	Probabilidade
Dados	0.8589955
Funcional	0.10510544
Integração	0.00175545
Outros	0.03259172
Uso	0.00155189

Tabela 14 – Classes e Probabilidades

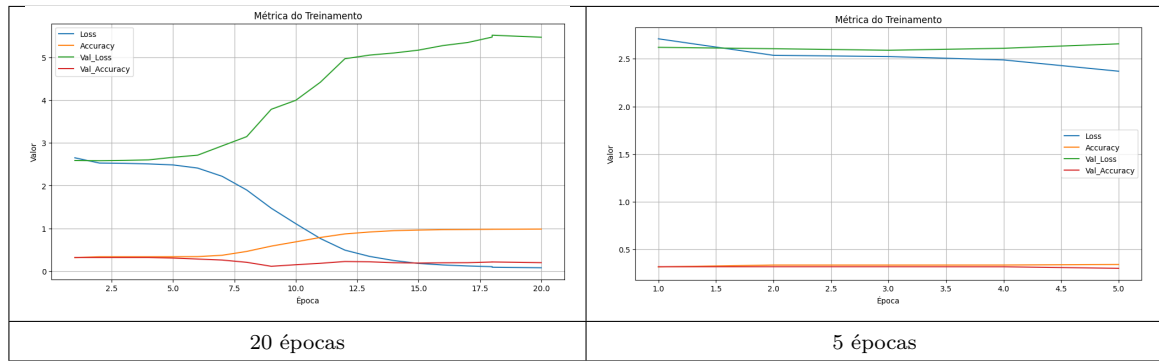


Tabela 15 – Curva de aprendizado com 5 e 20 épocas

#### 5.4.3 *fine-tuning* com 5 e 20 épocas

Foi experimentado uma variação de épocas, com uma forma relativamente bruta de *early stop* com cinco épocas e um extensão das épocas para vinte. Em ambos os casos, o comportamento foi relativamente similar, e novamente com *overfitting* para as vinte épocas e sem evolução para as cinco épocas como pode ser visto na Tabela 15

### 5.5 Uso de dados sintéticos para o *fine-tuning* do BERT (Ciclo 3)

O Ciclo 3, em sua vez, busca ampliar os dados com o uso de dados sintéticos. Dadas as limitações para fazer o *fine-tuning* do BERT para o contexto desse trabalho, uma alternativa que se levanta é a aplicação de dados sintéticos. (LI *et al.*, 2023) coloca que, embora os *LLMs* sejam capazes de gerar dados realistas, sua eficácia na geração de dados para classificação de texto é variável e depende de fatores como a subjetividade da tarefa, a qualidade dos dados de treinamento e a complexidade da tarefa. Por fim, o artigo sugere que a geração de dados sintéticos com *LLMs* pode ser uma ferramenta útil para aprimorar a classificação de texto, mas é importante considerar suas limitações e escolher cuidadosamente as tarefas adequadas. (TANG *et al.*, 2023) descreve que para tarefas específicas de mineração de texto, como reconhecimento de entidade nomeada biológica e extração de relações o uso de dados sintéticos com ChatGPT, apesar de apresentar desempenho limitado, ao gerar uma grande quantidade de dados sintéticos de alta qualidade com rótulos utilizando, trouxe ganhos de aprendizagem. Para o conjunto de dados sintéticos, foram usadas as mensagens de explicação geradas pelo LLM que sustentam a classificação, cuja análise qualitativa foi feita na Seção 4.4 que se mostraram coerentes com a classificação. Foram somente incluídas as mensagens que estavam relacionadas a classificações cuja pertinência foi 'Alta' ou 'Média'. O conjunto de chamados classificados foi ampliando para 9.521 textos classificados e foram adicionadas 24.981 mensagens geradas para diversas classificações das mensagens originais, o tamanho do conjunto de dados fica em 34.502 textos com respectivas classificações. A Figura 21 mostra 20 itens mais frequentes, o que leva a ponderação sobre balanceamento de dados. Finalmente, com a

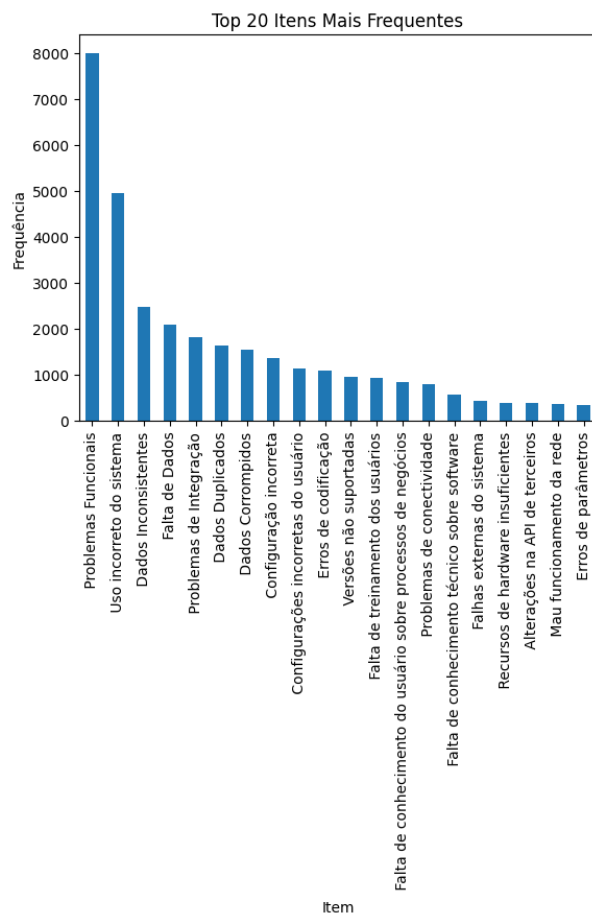


Figura 21 – Distribuição das 20 classificações mais frequentes com a fusão de dados originais com os dados sintéticos

limpeza de chamados com classificação exclusivamente de pertinência "Baixo" ou problemas de mensagens nulas, chegou-se a 31.123 textos com suas respectivas classificações.

(LEEVY *et al.*, 2018) apresenta duas técnicas que incluem métodos de nível de dados (por exemplo, amostragem de dados) e métodos de nível de algoritmo (por exemplo, sensíveis a custo e híbrido/conjunto). Os métodos de amostragem de dados são populares no tratamento de desequilíbrio de classe, com métodos de amostragem aleatória geralmente apresentando melhores resultados gerais. (DEVLIN *et al.*, 2018), apesar de não mencionar claramente, o artigo destaca a capacidade do BERT de aprender representações contextuais de um conjunto de dados grande e diversificado, o que pode torná-lo mais resiliente a desequilíbrios de classe. Outro aspecto importante é saber se as mensagens que serão submetidas tem um número de palavras que se encaixa no limite definido para este trabalho de 128 para fazer o *fine-tuning* do BERT. A Figura 22 ilustra que a grande maioria das mensagens, precisamente 33.894 mensagens ou 98,23% do conjunto total se encaixam nesse limite.

A submissão para *fine-tuning* é feita com os 20 itens classificados com maior frequência, o que conta com 32.123 ou 93,10% das mensagens classificadas, como mostrado

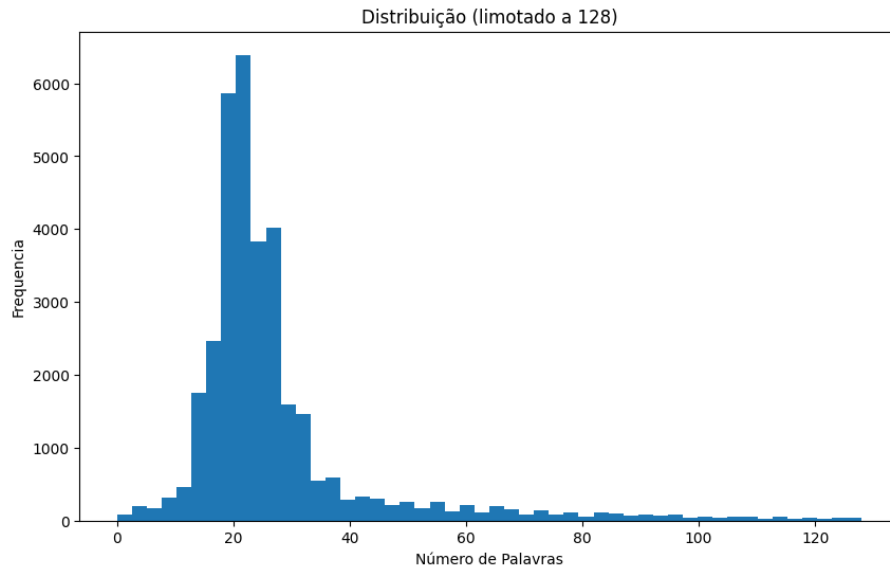


Figura 22 – Distribuição de número de palavras do conjunto de dados ampliado

na Tabela 16. Para esse cenário de *fine-tuning*, não foram aplicadas as técnicas de *dropout* e regularização.

Item Enumerado	Cont.
Problemas Funcionais	8001
Uso incorreto do sistema	4949
Dados Inconsistentes	2486
Falta de Dados	2093
Problemas de Integração	1814
Dados Duplicados	1639
Dados Corrompidos	1543
Configuração incorreta	1357
Configurações incorretas do usuário	1134
Erros de codificação	1096
Versões não suportadas	964
Falta de treinamento dos usuários	929
Falta de conhecimento do usuário sobre processos de negócios	850
Problemas de conectividade	796
Falta de conhecimento técnico sobre software	571
Falhas externas do sistema	422
Alterações na API de terceiros	397
Recursos de hardware insuficientes	397
Mau funcionamento da rede	355
Erros de parâmetros	330

Tabela 16 – Lista de Problemas e Contagem

### 5.5.1 Resultados do *fine-tuning* com base de mensagens ampliada

As métricas de desempenho (precisão, revocação, F1 e suporte) mostradas na Tabela 17 dão uma indicação clara de quão bem o modelo aprendeu a classificar as classes dadas.

- **Acurácia:** A acurácia geral do modelo é de 0,73, o que é aceitável, mas sugere que há espaço para melhorias. Dada a complexidade e diversidade dos dados, uma acurácia acima de 0,70 indica que o modelo capturou padrões significativos nos dados.
- **Métricas de Média Ponderada:** As médias ponderadas para precisão, revocação e F1 estão todas em torno de 0,73, o que sugere que o modelo tem um desempenho consistente em todas as diferentes classes. No entanto, essas métricas abaixo de 0,8 sugerem que o modelo enfrenta dificuldades com algumas classes.
- **Macro Avg** (média não ponderada entre as classes):
  - As médias macro para precisão, revocação e F1 são ligeiramente inferiores às médias ponderadas, o que sugere que o desempenho do modelo varia entre as diferentes classes.
- **Weighted Avg:** Como as médias ponderadas são ligeiramente superiores, isso indica que o modelo tem um desempenho melhor nas classes com mais exemplos, que estão mais fortemente representadas no cálculo ponderado.
- **Classes de Alto Desempenho:** As classes *Falhas Externas ao Sistema*, *Falta de conhecimento técnico sobre software*, *Falta de treinamento dos usuários* e *Uso incorreto do sistema* apresentam um desempenho consistente em todas as métricas, indicando que o modelo aprendeu bem as características dessas classes. A seguir, uma análise mais individual para explicar os porquês:
  - Classe *Falhas externas do sistema* (F1: 0,84), *Falta de conhecimento técnico sobre software* (F1: 0,82) e *Uso incorreto do sistema* (F1: 0,82) demonstram forte desempenho. Essas classes têm alta precisão e revocação, indicando que o modelo pode identificar essas classes de forma precisa e consistente, com pouca confusão.
  - A alta precisão nessas classes sugere que, quando o modelo prevê essas classes, geralmente está correto, e o alto revocação indica que ele está identificando uma grande parte das instâncias reais dessas classes.
- **Classes de Baixo Desempenho:** As classes *Dados Duplicados*, *Dados Inconsistentes*, *Erros de codificação*, *Falta de Dados* e *Problemas de Integração* têm um



desempenho inferior, especialmente em termos de precisão. Isso sugere que o modelo pode estar tendo dificuldades em classificar corretamente os exemplos dessas classes.

- Classes como *Dados Duplicados* (F1: 0,62), *Erros de codificação* (F1: 0,60), *Problemas de Integração* (F1: 0,60) e *Falta de Dados* (F1: 0,63) apresentam desempenho inferior. Os F1s mais baixos indicam problemas com precisão, revocação ou ambos. *Dados Duplicados* tem um revocação de 0,63, o que significa que o modelo está perdendo cerca de 0,37 das instâncias reais dessa classe. *Problemas de Integração* mostra uma precisão de 0,55, indicando que quase metade das instâncias previstas como sendo dessa classe estão incorretas, sugerindo que o modelo está confundindo essa classe com outras. Isso indica que essas classes são mais desafiadoras para o modelo aprender, potencialmente devido a dados de treinamento insuficientes ou ruidosos, ou porque essas classes compartilham semelhanças significativas com outras classes.

### 5.5.2 Impacto dos Dados Sintéticos

O uso de dados sintéticos no *fine-tuning* pode ter induzido as classificações. Isso pode explicar algumas das classes de menor desempenho, onde o modelo tem dificuldade em classificar instâncias corretamente. No entanto, os dados sintéticos também aumentaram o desempenho fornecendo exemplos adicionais. Nesse contexto, parece que para algumas classes, o modelo se beneficiou dos dados sintéticos (por exemplo, Classe *Falhas externas do sistema* com um alto F1), enquanto para outras, os dados sintéticos podem não ter sido tão eficazes.

- **Desbalanceamento de Classes:** A variação no suporte (número de instâncias por classe) indica que o desbalanceamento de classes gerou ruído no resultado, o que pode contribuir para o desempenho inconsistente do modelo em diferentes classes.

### 5.5.3 Considerações

O *fine-tuning* do BERT neste conjunto de dados mostra um sucesso moderado, com uma acurácia geral de 0,73 e desempenho razoavelmente forte em algumas classes. No entanto, os F1s mais baixos para várias classes sugerem que há áreas para melhoria, particularmente no tratamento do desbalanceamento de classes e na garantia da qualidade e representatividade dos dados sintéticos. Um ajuste mais refinado, potencialmente com foco nas classes de menor desempenho e na melhoria da qualidade dos dados, poderia aprimorar o desempenho do modelo. Contudo, se comparado com os resultados anteriores, os quais não contavam com os dados sintéticos, o resultado deste ciclo se mostra significativamente melhor.

Classe	Precisão	Revocação	Pont. F1	Suporte
Alterações na API de terceiros	0.83	0.68	0.75	111
Configuração incorreta	0.83	0.86	0.84	385
Configurações incorretas do usuário	0.83	0.76	0.79	342
Dados Corrompidos	0.76	0.77	0.76	464
Dados Duplicados	0.61	0.63	0.62	473
Dados Inconsistentes	0.67	0.65	0.66	773
Erros de codificação	0.61	0.59	0.60	309
Erros de parâmetros	0.81	0.78	0.80	93
Falhas externas do sistema	0.92	0.78	0.84	129
Falta de Dados	0.61	0.66	0.63	642
Falta de conhecimento do usuário sobre processos de negócios	0.77	0.67	0.72	260
Falta de conhecimento técnico sobre software	0.87	0.78	0.82	171
Falta de treinamento dos usuários	0.82	0.77	0.79	269
Mau funcionamento da rede	0.83	0.67	0.74	106
Problemas Funcionais	0.71	0.74	0.73	2459
Problemas de Integração	0.55	0.66	0.60	525
Problemas de conectividade	0.86	0.65	0.74	239
Recursos de hardware insuficientes	0.76	0.73	0.75	116
Uso incorreto do sistema	0.82	0.82	0.82	1488
Versões não suportadas	0.72	0.69	0.71	283

Tabela 17 – Precisão, revocação, F1, Suporte conjunto ampliado de dados

#### 5.5.4 Análise dos Resultados de Treinamento e Validação e Similaridade de Jaccard e Sorensen-Dice

Como mostrado na Figura 23, a perda de treinamento (*Loss*) ao longo das 10 épocas apresentou uma clara tendência de queda, indicando que o modelo está aprendendo a minimizar o erro ao longo do tempo. A perda começou em 1.82 na primeira época e caiu consistentemente até 0.26 na décima época. Isso demonstra que o modelo está melhorando sua capacidade de prever corretamente durante o treinamento. Por outro lado, a perda de validação (*Val\_Loss*) apresentou uma tendência diferente. Inicialmente, houve uma pequena redução, indo de 0.94 na primeira época para 0.82 na segunda época. No entanto, a partir da terceira época, a *Val\_Loss* começou a aumentar gradualmente, atingindo 1.26 na décima época. Este aumento sugere que o modelo pode estar começando a superestimar os padrões dos dados de treinamento, resultando em *overfitting*. A acurácia de treinamento (*Accuracy*) também mostrou uma melhoria constante, aumentando de 0,5042 na primeira época para 0,92.52 na décima época. Este aumento contínuo reflete que o modelo está se tornando progressivamente mais eficiente em classificar corretamente os exemplos de treinamento. Em contrapartida, a acurácia de validação (*Val\_Accuracy*) mostrou um comportamento estável, com pequenas flutuações. Começando em 0,7495 na primeira época, a *Val\_Accuracy* atingiu seu pico de 0,7743 na segunda e na sexta épocas, mas começou a declinar ligeiramente a partir da sétima época, terminando em 0,726 na décima época. Essa tendência reforça a hipótese de *overfitting*, onde o modelo se torna menos eficaz em dados não vistos, mesmo que continue melhorando em dados conhecidos.

Como etapa final é interessante comparar as classificações feitas pelo BERT e pelo GPT4. Dadas as diferenças intrínsecas em suas arquiteturas e abordagens de aprendizado,

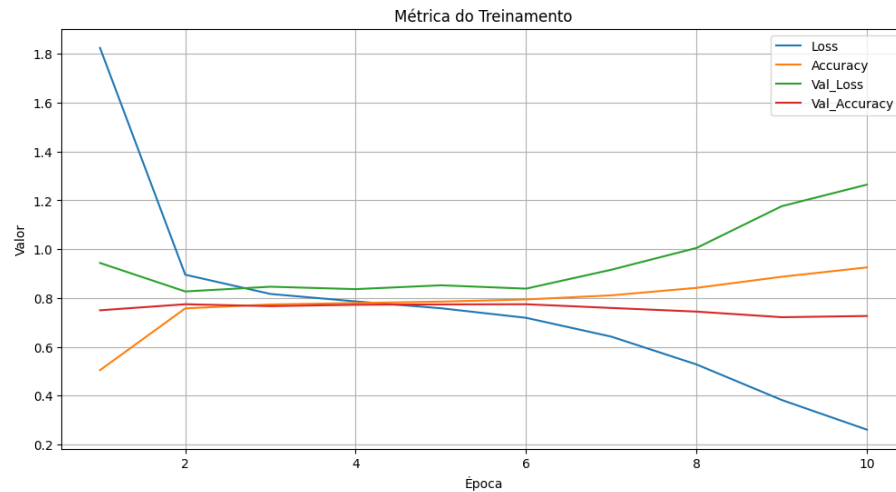


Figura 23 – Métricas de *fine-tunning* com dados ampliados Ciclo 3

as classificações geradas por esses modelos podem variar significativamente. Avaliar a similaridade entre as classificações fornecidas por diferentes modelos é essencial para validar a consistência e a eficácia das soluções propostas. Neste cenário, a Similaridade de Jaccard se apresenta como uma métrica quantitativa robusta para medir a concordância entre as classificações geradas por BERT e GPT-4. Outrossim, não foram, neste momento, usadas as probabilidades de classificação geradas pelo BERT como fator de ponderação ou ajuste de métrica.

Para aplicar a Similaridade de Jaccard na comparação das classificações de chamados realizadas pelos modelos BERT e GPT-4, foram feitos os seguintes passos:

1. **Chamados classificados pelo GPT 4:** foram extraídas aleatoriamente 500 novos chamados da base original e classificados com o GPT 4.
2. **Representação das Classificações:** cada chamado classificado pelos modelos BERT e GPT-4 é representado como um conjunto de categorias. Por exemplo, um chamado pode ser classificado como pertencente às categorias “*Falta de Dados*”, “*Problemas de Conectividade*” pelo BERT, e como “*Falta de Dados*”, “*Problemas Funcionais*” pelo GPT-4.
3. **Cálculo da Similaridade de Jaccard:** para cada chamado, foi calculada a Similaridade de Jaccard entre os conjuntos de categorias gerados pelo BERT e pelo GPT-4. Isso é feito utilizando a fórmula apresentada anteriormente, onde a interseção representa as categorias comuns entre os dois modelos, e a união representa o total de categorias identificadas por ambos os modelos.

O gráfico de Violino, Figura 24 mostra a distribuição das métricas de similaridade de Jaccard entre as mensagens de erro classificadas. A forma do "violino" ilustra a densidade

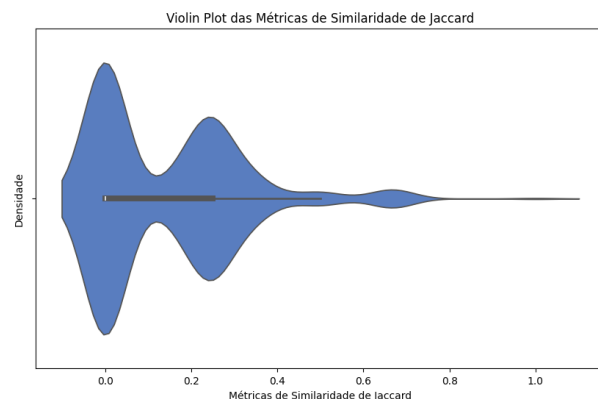


Figura 24 – Gráfico de Violino para a Similaridade de Jaccard

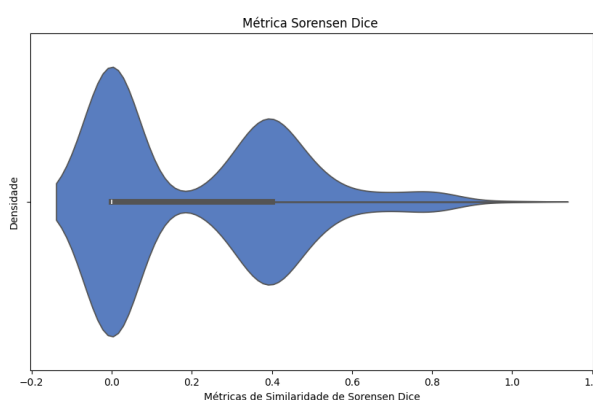


Figura 25 – Índice de Sorensen-Dice

dos dados em diferentes intervalos de valores. O "violino" tem múltiplos picos, o que indica que há vários grupos distintos de similaridade entre as mensagens. A largura do violino em diferentes pontos ao longo do eixo das métricas de similaridade representa a densidade dos valores. Uma largura maior sugere uma maior densidade de valores naquela faixa, enquanto uma largura menor indica uma menor densidade. A parte mais espessa do violino, próxima à mediana, indica que a maioria dos pares de conjuntos apresenta um nível de similaridade intermediário. A presença de um "bigode" mais longo à direita da mediana indica que há um número considerável de pares de conjuntos com alta similaridade. Isso pode ser interpretado como a existência de grupos de conjuntos bastante semelhantes entre si. Também existe uma alta concentração de elementos à esquerda do gráfico, o que indica um grupo relevante com baixa similaridade.

O índice de Sorensen-Dice dá um peso maior aos elementos que estão presentes em ambos os conjuntos, enquanto o índice de Jaccard trata a presença e a ausência de elementos de forma mais simétrica.

A mediana da distribuição parece estar próxima de 0,2, indicando que, em média, os conjuntos comparados possuem uma baixa similaridade. A cauda longa à direita sugere que há um número considerável de pares de conjuntos com alta similaridade. A

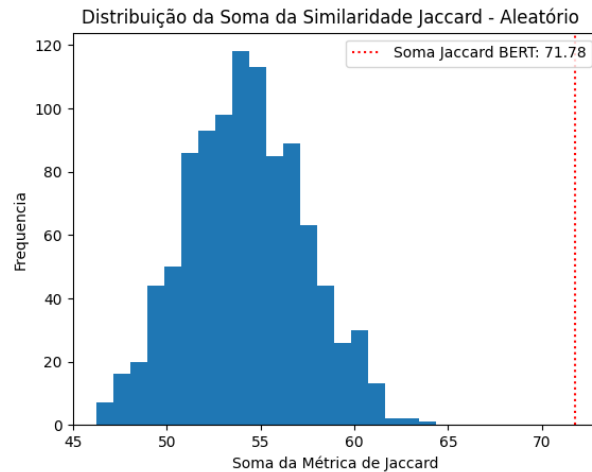


Figura 26 – Distribuição da soma das métricas de Jaccard com simulação de classificação aleatória

Figura 25, como esperado, tem aspecto similar a Figura 24, e evidencia a concordância intermediária de classificações entre os dois os modelos de linguagem. Em síntese, o BERT conseguiu classificar e ter uma similaridade intermediária no geral, mas com um conjunto relativamente expressivo com baixa similaridade, se traduzindo em uma concordância média entre GPT 4 e BERT.

#### 5.5.5 Teste de Hipótese para descartar classificador aleatório

Por fim foi feita uma simulação de classificação para verificar se o BERT treinado estava classificando de forma aleatória. Foram geradas classificações aleatórias nas mensagens com as mesmas classes usadas para o treinamento e calculado os índices de Jaccard para cada classificação e posteriormente foram somados esses valores, resultando em um total geral para cada uma das 1.000 iterações. A distribuição dessa soma para 1.000 simulações está na Figura 26.

##### Hipóteses:

- **Hipótese Nula (H0):** BERT é um classificador aleatório. A soma do Jaccard obtida pelo BERT não é significativamente diferente da média das somas obtidas nas simulações aleatórias.
- **Hipótese Alternativa (H1):** BERT não é um classificador aleatório. A soma do Jaccard obtida pelo BERT é significativamente diferente da média das somas obtidas nas simulações aleatórias.

##### Média e desvio padrão das classificações aleatórias:

$$\mu = 54.211$$

$$\sigma = 3.132$$

**Valor observado (soma da classificação do BERT):**

$$x = 71.7833$$

**Estatística do teste Z:**

$$Z = \frac{x - \mu}{\sigma}$$

Com  $Z$ : 5.611 e  $P$ -value: 1.0083161443823844e-08 a hipótese nula pode ser rejeitada. O classificador BERT NÃO é aleatório.

#### 5.5.6 Análise comparativa da classificação GPT 4 e BERT maiores e menores índices de Jaccard e Sorensen Dice

A Tabela 18 mostra o resultado das classificações que obtiveram as maiores índices de Jaccard e Sorensen-Dice. É interessante notar que o GPT 4 classifica em maior número "Uso incorreto do sistema" e "Problemas Funcionais" em quase todas as descrições e que o BERT, por outro lado, apresenta uma maior diversidade de classificações. Além de "Uso incorreto do sistema" e "Problemas Funcionais", ele identifica "Configurações incorretas do usuário", "Problemas de Integração", "Falta de Dados", entre outros. É interessante notar também que as probabilidades distribuídas variam bastante entre o primeiro exemplo da tabela a o último.

Description	GPT 4	BERT	Prob.	Jaccard	Sor. Dice
"Usuario esta homologando processo de contestação junto a CIP porem o processo esta apontando erro na geração dos dados Peço que analisem e nos ajudem a identificar o motivo da ocorrência em anexo o email do usuario com a tela do Erro"	Problemas Funcionais, Uso incorreto do sistema	Uso incorreto do sistema, Configurações incorretas do usuário, Problemas Funcionais	0.9411, 0.0220, 0.0176	0,67	0,80

Description	GPT 4	BERT	Prob.	Jaccard	Sor. Dice
<p>"Ao gerar o formulário de recadastramento do sistema de benefícios prevnet observamos que não possui o termo de Pessoa Politicamente Exposta Gentileza incluir no verso do formulário</p> <p>Na oportunidade e com a finalidade de atender a LGPD gentileza incluir 03 três campos para inclusão de textos a serem enviados pelo SERGUS no final do formulário</p> <p>Para ajudar na ilustração segue anexo word como modelo</p> <p>Estamos à disposição</p> <p>Atenciosamente</p> <p>Jessica"</p>	Uso incorreto do sistema, Problemas Funcionais	Problemas Funcionais, Problemas de Integração, Uso incorreto do sistema	0.8438, 0.0784, 0.0278	0,67	0,80
<p>"Foram realizadas cancelamento de duas cotas contempladas 2201138 0 lance embutido valor R 503265 e 62104290 termo variável descontado do credito no valor R 163713 feita exclusão da contemplação antes de estornar o lance e as variáveis por este motivo o valor não retornou ao grupo e apresentou diferença</p> <p>Como as cotas foram excluídas não é possível resgatar os valores para o grupo Não encontramos permissões no sistema</p> <p>No caso não foi feito o primeiro processo antes da exclusão da contemplação que seria no financeirocrédito despesa o estorno do lance e da variáveis</p> <p>Segue imagens em anexo"</p>	Problemas Funcionais, Uso incorreto do sistema	Falta de Dados, Uso incorreto do sistema, Problemas Funcionais	0.4144, 0.3437, 0.1281	0,67	0,80
<p>"Prezados</p> <p>Na homologação da nova versão da CAPITALIZAÇÃO referente ao resgate verificamos que o resgate parcelado está com erro o sistema está calculando todo o saldo na hora de calcular a próxima parcela mesmo erro que existia em produção SD52527 a tela de consulta de resgate não mostra os dados bancários do participante"</p>	Problemas Funcionais, Falta de Dados	Falta de Dados, Problemas Funcionais, Dados Corrompidos	0.4485, 0.3121, 0.1944	0,67	0,80
<p>"Boa tarde</p> <p>Cliente com erro na troca de crédito evidencia em anexo</p> <p>i156 423 CPF 26046077869 Simone Bello de Souza Garcia Código do cliente 1464620"</p>	Uso incorreto do sistema, Problemas Funcionais	Problemas Funcionais, Uso incorreto do sistema, Erros de codificação	0.3041, 0.2304, 0.1556	0,67	0,80

Description	GPT 4	BERT	Prob.	Jaccard	Sor. Dice
"Falha ao usa a função ANEXAR documentos via XDOC para o contrato 12056000196977 outro contrato que abre anexo normalmente 12056000216244 Chamado 4780221"	Problemas Funcionais, Uso incorreto do sistema	Erros de codificação, Uso incorreto do sistema, Problemas Funcionais	0.2278, 0.1846, 0.1683	0,67	0,80
"Boa tarde Assim como recomendado no chamado anterior SD84549 o log de transação está sendo gerado para avaliação de qual foi a ação que ocorreu na cota que indique a exclusão do lançamento Mas ainda não sabemos como resolver esta questão quais os procedimentos para a resolução da questão Creio que o atendimento ainda não esteja solucionado"	Dados Corrompidos, Uso incorreto do sistema	Uso incorreto do sistema, Dados Corrompidos, Dados Inconsistentes	0.1826, 0.1586, 0.1219	0,67	0,80

Tabela 18 – Comparação Qualitativa entre GPT 4 e BERT

A Tabela 19 mostra as mensagens com nenhuma classificação coincidente entre GPT 4 e BERT. Uma observação interessante é que tanto GPT-4 quanto BERT identificam consistentemente problemas funcionais em todos os casos. Isso sugere que ambos os modelos têm uma compreensão dos principais problemas funcionais descritos nos textos, que ficam classificados como "Erros de codificação"(GPT-4) e "Problemas Funcionais"(BERT), o que indica que uma potencial consolidação, como sugerida em seções anteriores. Essa visão aponta para uma possível complementaridade entre os dois modelos: enquanto GPT-4 oferece mais precisão nas análises, o BERT pode fornecer uma visão complementar e talvez mais geral que abrange diversas categorias. Dependendo da aplicação e do ajuste do processo de negócio, uma combinação dos resultados de ambos os modelos poderia oferecer uma classificação mais robusta e completa.

Description	GPT-4	BERT	Prob.	Jaccard	Sor. Dice
-------------	-------	------	-------	---------	--------------



Description	GPT-4	BERT	Prob.	Jaccard	Sor. Dice
<p>Prezados boa tarde</p> <p>Ao emitirmos memória de cálculo para pagamento de portabilidade referente ao participante Inscrição 624 verificamos que os valores da memória de cálculo não estão considerando a última devolução conforme demonstra evidências anexas Data da devolução 08042021 Último fechamento do sistema 032021 Vigente para pagamentos e arrecadações ao longo do mês 042021 Assim solicitamos que seja realizado a correção da memória para que considere a devolução para que não ocorra pagamento a maior</p> <p>Permanecemos à disposição para demais esclarecimentos</p> <p>Obrigada</p> <p>Atenciosamente Janesca Iurkevicz</p>	Dados Inconsistentes, Dados Corrompidos	Problemas Funcionais, Versões não suportadas, Problemas de Integração	0.9926, 0.0021, 0.0014	0,00	0,00
<p>Prezados boa tarde</p> <p>Os seguintes fundos estão dando erro no processamento</p> <p>Segue log em anexo</p>	Erros de codificação	Problemas Funcionais, Versões não suportadas, Problemas de Integração	0.9888, 0.0024, 0.0021	0,00	0,00
Operação processando gentileza interromper e reiniciar	Mau funcionamento da rede	Problemas de Integração, Uso incorreto do sistema, Problemas de conectividade	0.9866, 0.0034, 0.0026	0,00	0,00
<p>GENLATPLA2999 GestorME</p> <p>Mensagens Feedings Enrichment Alterar o Controle IF de ERC20240214XXXXXXXXXX</p> <p>Para NrOperCam XXXXXXXXXXXX</p> <p>InclusãoAlteração</p>	Falta de conhecimento do usuário sobre processos de negócios, Versões não suportadas	Problemas Funcionais, Dados Inconsistentes, Falta de Dados	0.9851, 0.0072, 0.0041	0,00	0,00
Peço a gentileza de alterar o email anterior que estava cadastrado no relay para o email atual conforme telas anexo	Problemas de instalação	Problemas Funcionais, Erros de codificação, Dados Duplicados	0.9842, 0.0052, 0.0049	0,00	0,00
Bom dia Favor verificar e ajustar as alíquotas regressivas Veja no anexo que a matrícula 000105178 000109857 possui um saldo de contas e na distribuição das alíquotas o saldo está maior	Dados Inconsistentes	Problemas Funcionais, Falta de Dados, Versões não suportadas	0.9815, 0.0068, 0.0030	0,00	0,00

Description	GPT-4	BERT	Prob.	Jaccard	Sor. Dice
Olá Hoje o sistema nos permite cadastrar duas pessoas em uma mesma cota com mesmo número de contrato aquela opção de 50 para cada um Mas no APP de vendas não tem essa opção Como procedemos	Dados Duplicados, Uso incorreto do sistema	Versões não suportadas, Problemas de conectividade, Mau funcionamento da rede	0.9808, 0.0042, 0.0029	0,00	0,00

Tabela 19 – Comparação GPT 4 e BERT com os menores índices de Jaccard e Sorensen Dice

### 5.5.7 Classificação de 78.348 Chamados com o BERT com modelo ajustado Ciclo 3

Os histogramas apresentados na Tabela 20 oferecem uma visão comparativa das distribuições de classificação de chamados de suporte realizadas por dois modelos de linguagem, BERT e GPT-4, com a nota que a frequência representada do GPT 4 é relativa ao segundo lote feito para as análises quantitativas. A análise a seguir visa destacar as principais similaridades e diferenças entre as classificações realizadas por cada modelo, com foco nos 5 primeiros itens de maior frequência em cada distribuição. Nota importante. Não foi feita uma análise para descartar a hipótese de que o treinamento com dados desbalanceados tenha gerado viés no modelo BERT, mas concomitante à essa ressalva, também vale ponderar que talvez os dados desbalanceados favorecem ao BERT ter classificação em perfil similar. Se a classificação pelo GPT 4 foi considerada aderente pela análise qualitativa, um potencial viés pode ser positivo para um perfil de uso da informação de classificação em mais alto nível. Ambos os modelos classificaram que "Problemas Funcionais" com maior frequência, o que pode ser um resultado esperado dado que esse item sempre despontou como o mais classificado. Enquanto "Falta de Dados" ocupa a segunda posição na distribuição do BERT (9,49%), ela se encontra em quarto lugar na distribuição do GPT-4 (7,78%). A categoria "Dados Inconsistentes" apresenta uma frequência semelhante em ambos os modelos.

Ao se ponderar pelo o uso mais gerencial das informações de classificação e considerando a análise das colunas da Tabela 21 sob a ótica gerencial, e o fato de que o GPT-4 demonstrou consistência na classificação de chamados de suporte, a recomendação para o uso de um BERT ajustado pode ser avaliada como alternativa com algumas considerações. O BERT, conforme os dados apresentados, consegue identificar as principais categorias de forma similar ao GPT-4. As classificações como "Problemas Funcionais", "Falta de Dados", e "Uso incorreto do sistema" aparecem com alta frequência em ambos os modelos. Há uma inversão de posições entre as categorias "Falta de Dados" e "Uso incorreto do sistema" entre os dois modelos. O BERT classifica "Falta de Dados" como a segunda categoria mais frequente, enquanto o GPT-4 classifica "Uso incorreto do sistema" em segundo

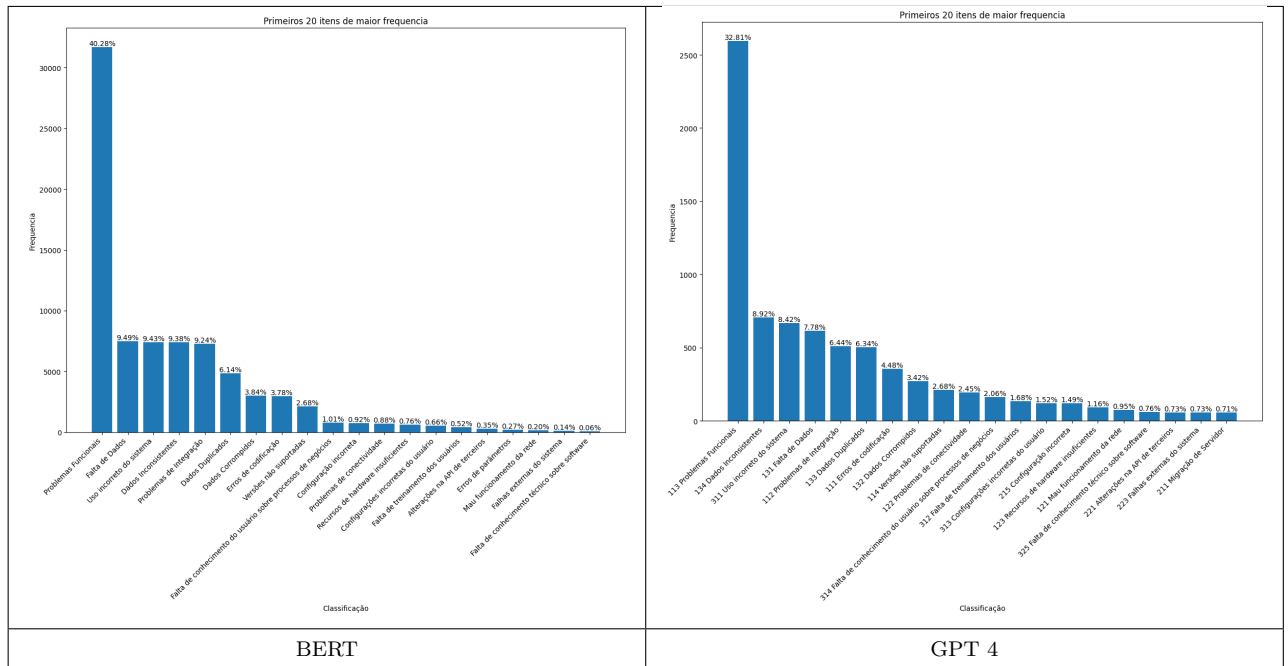


Tabela 20 – Comparação da distribuição classificação feita pelo BERT e GPT 4

lugar. O BERT apresentou uma distribuição relativamente próxima à do GPT-4 em várias categorias, o que sugere que, com um *fine-tuning*, ele pode se tornar uma ferramenta eficaz para a classificação. Se a consistência e precisão na classificação dos chamados forem prioridades, e dado que o GPT-4 já demonstrou um desempenho sólido, o uso de BERT, mesmo ajustado, deve ser considerado com ressalvas.

BERT	%	GPT-4	%
Problemas Funcionais	40.28%	Problemas Funcionais	32.81%
Falta de Dados	9.49%	Dados Inconsistentes	8.92%
Uso incorreto do sistema	9.43%	Uso incorreto do sistema	8.42%
Dados Inconsistentes	9.38%	Falta de Dados	7.78%
Problemas de Integração	9.24%	Problemas de Integração	6.44%
Dados Duplicados	6.14%	Dados Duplicados	6.34%
Dados Corrompidos	3.84%	Erros de codificação	4.48%
Erros de codificação	3.78%	Dados Corrompidos	3.42%
Versões não suportadas	2.68%	Versões não suportadas	2.68%
Falta de conhecimento do usuário sobre processos de negócios	1.01%	Problemas de conectividade	2.45%

Tabela 21 – Comparação das principais classificações feitas por BERT e GPT-4

## 5.6 Aplicação de Regularização L2 e *Dropout* e nova rodada de *fine-tuning* (Ciclo 4)

Foi feito mais um *fine-tuning* com o conjunto sintético tendo este substituídas palavras contidas na classificação da descrição da mensagem. Por exemplo "Problema Funcional", no singular, foi trocado por "Falha" na mensagem de texto sintético submetida ao treinamento. Isso visa evitar um viés relacionado diretamente às palavras da classificação ao estarem no texto a ser classificado. Como pode ser visto nos diversos exemplos, quando o GPT 4 dá uma explicação do porquê da classificação, ele explicitamente menciona o próprio texto da classificação na mensagem. Contudo, os resultados não mudaram substancialmente como pode ser visto no Anexo, Tabela 33.

### 5.6.1 Análise do *fine-tuning* do BERT com ajustes dos hiperparâmetros

Após a terceira época, feita com OneCycle, LR Máxima de  $2e-05$ , limpeza de mensagem, *dropout* 20% e  $l2(0.01)$ , a perda de validação começou a aumentar novamente, chegando a 1.4391 na décima época. Esse aumento na perda de validação pode indicar que o modelo está começando a sofrer de *overfitting*. E os demais índices também não melhoraram substancialmente. A acurácia geral do modelo é de 0,68, o que indica que aproximadamente dois terços das previsões estão corretas. Esse valor é moderado e sugere que há espaço para melhorias. A média ponderada para precisão, revocação e F1 é de 0.68, indicando um desempenho equilibrado, mas não excelente, em todas as classes. Os resultados podem ser vistos na Tabela 34 e mostram que não houve ganho, ou mesmo houve piora, no modelo com a estratégia de remover as palavras da classificação no conjunto de texto das mensagens sintéticas. Embora não ofereça uma interface direta para aplicar *dropout* e L2 regularização, pode-se manipular o modelo subjacente para alcançar o mesmo efeito. Essa abordagem requer um pouco mais de código, mas permite um controle mais preciso sobre o processo de treinamento. Foi aplicada regularização L2 com 0,01 e *dropout* com 0,2. Além disso, no conjunto das explicações, que se tornaram dados sintéticos para reforçar o treinamento, foram excluídas as menções e o próprio texto da classificação feita, afim de verificar se existe potencial viés de classificação com melhora nos índices de acuracidade e perda. Contudo, como pode ser observado na Figura 28, não houve melhora significativa no modelo ajustado.

## 5.7 *fine-tuning* com aumento de palavras e características (Ciclo 5)

Para o quinto ciclo de treinamento, foram ampliadas a quantidade de palavras de entrada para 256 palavras e vocabulário com 15.000 itens. O tamanho maior permite que o modelo considere sequências mais longas de texto. Isso é útil quando as entradas possuem informações importantes distribuídas ao longo de um texto maior. O `max_features` maior significa que o modelo tem acesso a um vocabulário mais extenso. Isso pode melhorar

o desempenho ao permitir que o modelo reconheça uma maior variedade de palavras e expressões, capturando nuances linguísticas e termos específicos que seriam ignorados com um vocabulário menor.

Classe	Precisão	Revocação	Pont. F1	Suporte
0	0.91	0.69	0.79	111
1	0.87	0.84	0.85	385
2	0.85	0.77	0.81	342
3	0.82	0.78	0.80	464
4	0.62	0.63	0.62	473
5	0.65	0.67	0.66	773
6	0.66	0.59	0.62	309
7	0.84	0.77	0.80	93
8	0.92	0.79	0.85	129
9	0.64	0.63	0.64	642
10	0.78	0.68	0.73	260
11	0.89	0.79	0.84	171
12	0.83	0.77	0.80	269
13	0.74	0.66	0.70	106
14	0.69	0.77	0.73	2459
15	0.62	0.65	0.64	525
16	0.76	0.66	0.71	239
17	0.85	0.70	0.77	116
18	0.83	0.83	0.83	1488
19	0.83	0.71	0.76	283
<b>Acurácia</b>	0.74 (9637)			
<b>Média Macro</b>	0.78	0.72	0.75	9637
<b>Média Pond.</b>	0.74	0.74	0.74	9637

Tabela 22 – Métricas do *fine-tuning* com maxlen=256 e Features = 15.000

Comparado com o modelo do Ciclo 3, esse novo modelo, com ajustes como *dropout*, regularização L2 256 palavras e vocabulário aumentado para 15.000, Tabela 22 apresenta, em geral, melhorias nas métricas de precisão, revocação e pontuação F1 em várias classes. A precisão aumentou em várias classes, como nas classes 0 (0.83 para 0.91), 1 (0.83 para 0.87), e 10 (0.77 para 0.78). Isso indica que o modelo atualizado tem menos falsos positivos em várias classes. A revocação também foi melhorada em algumas classes, como na classe 9 (0.66 para 0.63) e 18 (0.82 para 0.83), sugerindo que o modelo atualizado tem menos falsos negativos em geral. Para classes com menos suporte, como 13 e 17, houve uma estabilização e, em alguns casos, melhora na pontuação F1. Por exemplo, a classe 17 foi de 0.75 para 0.77, e a classe 13 se manteve estável (apesar de uma leve queda de 0.74 para 0.70). O modelo, com parâmetros ajustados, mostra uma melhoria geral no desempenho, especialmente em termos de precisão e pontuação F1. O aumento do `max_len` e `max_features`, junto com as técnicas de regularização, parecem ter contribuído para um melhor equilíbrio entre falsos positivos e falsos negativos, resultando em um modelo mais robusto. Contudo, ainda persiste uma tendência de *overfitting*. A perda de treinamento diminui consistentemente ao longo das épocas, começando em 1.74 e chegando a 0.32, o que é esperado durante o treinamento. A perda de validação, entretanto, inicialmente diminui, mas começa a aumentar após a terceira época. Esse comportamento sugere que, após esse ponto, o modelo começa a ajustar-se excessivamente aos dados de treinamento, em vez de generalizar bem para novos dados. A acurácia de validação, no entanto, atinge um pico em

torno da terceira época (77%) e depois começa a declinar ligeiramente, estabilizando-se em torno de 73-74%. Isso sugere que o modelo não está melhorando em sua capacidade de generalizar para novos dados após as primeiras épocas. A divergência entre a perda de treinamento e a perda de validação, juntamente com a estabilização e leve queda na acurácia de validação, são sinais de *overfitting*.

#### 5.7.1 Evolução nas métricas de precisão, revocação e F1 nos Ciclos 3, 4 e 5

Como pode ser observado na Figura 28, houve significativa melhora entre o Ciclo 3 e o Ciclo 5. Observa-se um aumento geral na precisão, com valores entre 0.62 e 0.92, e classes como "Configuração incorreta" e "Versões não suportadas" apresentando melhorias notáveis. As melhorias na revocação são menos pronunciadas em comparação com a precisão, o que indica que o modelo BERT, embora mais preciso, ainda pode ter dificuldades para capturar todas as instâncias verdadeiras em algumas classes. No entanto, o aumento em algumas classes sugere uma adaptação do modelo para capturar mais instâncias verdadeiras após mais treinamento. A F1 geral mostra melhorias em várias classes no Ciclo 5, sugerindo que o treinamento adicional do BERT resultou em um modelo que não só é mais preciso, mas também mais equilibrado em termos de Revocação. A melhora na F1 indica que o modelo é mais robusto e consistente na classificação das classes. Comparando o Ciclo 3 com o Ciclo 5, vemos um progresso no desempenho do modelo BERT. O treinamento adicional melhorou significativamente a precisão, e a F1 também mostra melhorias, o que é um indicativo de que o modelo está se tornando mais confiável em suas previsões. As melhorias, especialmente na precisão, sugerem que o modelo está se tornando mais eficaz em reduzir falsos positivos, o que é crucial no entendimento de chamados de sustentação.

### 5.8 Experimento para avaliar se dados balanceados geram diferença no treinamento e resultados (Ciclo 6)

A título de experimento mais livre para avaliar como é o comportamento de treinamento, foi feito o último *fine-tuning* do BERT, nesse caso, balanceando-se os dados igualmente nas dez categorias mais frequentes e aplainados em 1.500 mensagens classificadas para o treinamento. Esse corte foi definido por conta da limitação de dados. Foi usado para esse experimento o GPT 4 Mini, modelo `gpt-4o-mini-2024-07-18`, que segundo a OpenAI é obtém 82% no MMLU e atualmente supera o GPT-4 nas preferências de chat no *leaderboard LMSYS*, além de ser mais barato. As mensagens de explicação resultantes dessa submissão ao GPT 4 Mini também foram incluídas. Não houve uma análise qualitativa se o GPT4 Mini faz uma classificação adequada, como foi feito para o GPT 4. Houve também a mistura com os chamados classificados pelo GPT 4 segundo lote para maximizar o número de mensagens para o treinamento. Um grupo de 1.500 mensagens foi escolhido aleatoriamente do conjunto total de dados de cada um dos dez itens de maior frequência de classificação. Como nesse caso os dez itens mais frequentes representam

73,80% do total, isso significa que é possível haver distorções na classificação. Portanto, o objetivo desse experimento é ampliar o conjunto de dados, ter a mesma quantidade de exemplos para as categorias e verificar se comportamento de classificação pelo BERT segue aproximadamente o padrão de frequência visto em diversos histogramas ao longo do trabalho, onde o item "Problemas Funcionais" aparece sempre e substancialmente a frente. O treinamento foi executado com *dropout*, ajuste de L2, 256 palavras e 15.000 itens de vocabulário, com dados originais vindos de 31.123 chamados, extraíndo 1.500 aleatoriamente dos 10 itens mais frequentes e formando o conjunto de 15.000 chamados.

### 5.8.1 Resultados do experimento

O treinamento mostrou um comportamento tendendo rapidamente ao *overfitting*. Nota-se que a diferença entre a acurácia de treinamento e validação cresce significativamente ao longo do tempo. Enquanto a acurácia de treinamento alcança surpreendentes 98.20%, a de validação permanece estagnada em torno de 68%. Essa divergência é mais um sinal de que o modelo está se especializando demais nos dados de treinamento bem como após a terceira época, a perda de validação começa a aumentar de forma consistente, enquanto a perda de treinamento continua a diminuir, indicação de *overfitting*, como mostrado na Tabela 23 e a Figura 27

Época	Perda Treino	Acurácia Treino	Perda Val	Acurácia Val
1	1.8304	0.3710	1.1567	0.6351
2	1.0170	0.6686	0.9906	0.6722
3	0.8820	0.7030	0.9592	0.6827
4	0.7878	0.7311	0.9541	0.6856
5	0.6723	0.7702	1.0703	0.6736
6	0.5322	0.8213	1.1095	0.6713
7	0.3380	0.8870	1.3485	0.6700
8	0.1941	0.9352	1.4585	0.6669
9	0.1100	0.9692	1.5180	0.6847
10	0.0690	0.9820	1.5913	0.6796

Tabela 23 – Dados de treino e validação por época Ciclo 6

Observando a Tabela 24 é percebido que classe "Configuração incorreta" apresenta o melhor desempenho geral, com uma alta precisão e revocação, com Precisão 0.86 Revocação 0.82 e F1 0.84. "Uso Incorreto do Sistema" é outra classe com excelente desempenho, com precisão e revocação altas, 0.82 e 0,79 respectivamente. Já "Erros de Codificação" tem o desempenho pior em termos de precisão e revocação, com 0.54 e 0,56. E por fim, e notavelmente, "Problemas Funcionais" com precisão 0.56 e revocação 0.63, teve um desempenho baixo.

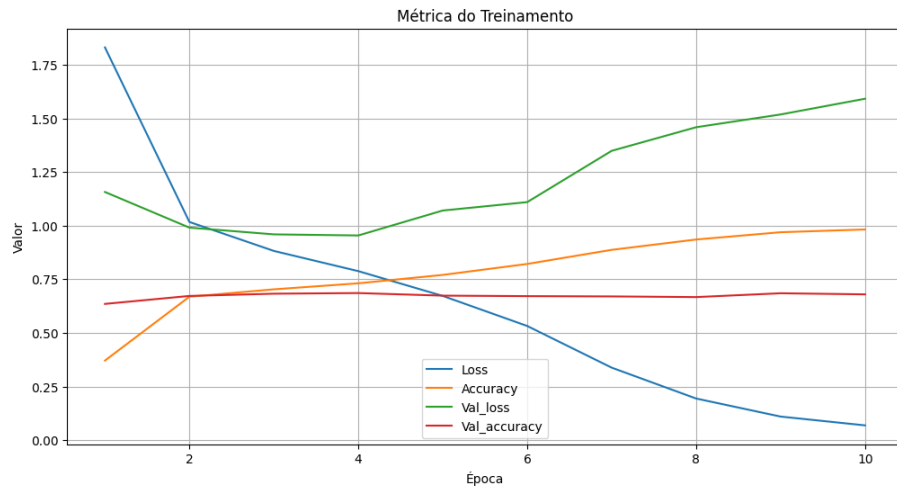


Figura 27 – Evolução treino e validação por época Ciclo 6

Classe	Precisão	Revocação	F1	Suporte
Configuração incorreta	0.86	0.82	0.84	431
Dados Corrompidos	0.75	0.72	0.74	456
Dados Duplicados	0.62	0.65	0.63	432
Dados Inconsistentes	0.71	0.71	0.71	443
Erros de codificação	0.54	0.56	0.55	474
Falta de Dados	0.62	0.58	0.60	468
Problemas Funcionais	0.56	0.63	0.59	452
Problemas de Integração	0.78	0.73	0.76	452
Uso incorreto do sistema	0.82	0.79	0.80	451
Versões não suportadas	0.60	0.62	0.61	441
<b>Acurácia</b>			0.68	4500
<b>Média Macro</b>	0.69	0.68	0.68	4500
<b>Média Ponderada</b>	0.68	0.68	0.68	4500

Tabela 24 – Métricas de desempenho por classe

### 5.8.2 Comparação Ciclo 3 e Ciclo 6

A comparação entre os dois resultados de treinamento (Ciclo 6 e Ciclo 3) revela diferenças e semelhanças nas métricas de desempenho para as classes em comum. O foco será nas classes: "Configuração incorreta", "Uso incorreto do sistema", "Problemas de Integração" e "Erros de codificação". Ambos os ciclos apresentam resultados semelhantes para "Configuração incorreta", com F1 idêntico (0.84), a precisão (0.86) é ligeiramente superior e a revocação (0.82) é melhor. Para "Uso Incorreto do Sistema", O Ciclo 3 apresenta uma F1 ligeiramente superior (0.82 contra 0.80 no Ciclo 6), com uma revocação equilibrada em relação à precisão (ambas 0.82) e revocação é um pouco menor (0.79). "Problemas de Integração" é consideravelmente melhor no Ciclo 6. No Ciclo 6, o modelo apresenta uma precisão significativamente superior (0.78 contra 0.55 no Ciclo 3) e uma F1 maior (0.76 contra 0.60). A performance para "Erros de Codificação" é melhor no Ciclo 3. A precisão



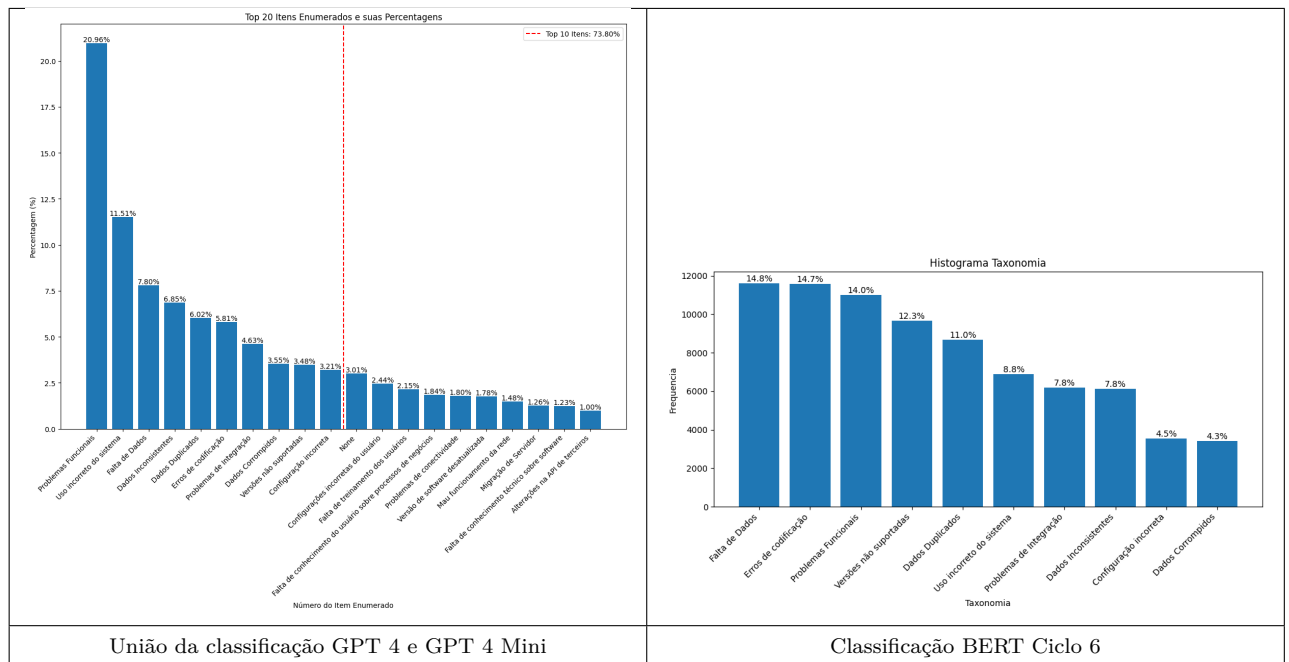


Tabela 25 – Diferença de Classificações GPT4 e GPT 4 Mini e BERT (Ciclo 6)

(0.61) e revocação (0.59) são superiores no Ciclo 3 em comparação com o Ciclo 6, onde a precisão é de apenas 0.54 e a revocação é de 0.56. Para "Problemas Funcionais" houve uma surpreendente diferença. No Ciclo 6, a precisão é de 0.56 e já no Ciclo 3, a precisão é significativamente maior, em 0.71. A revocação no Ciclo 6 é de 0.63 e Ciclo 3, a revocação é melhor, com 0.74, com F1 em 0,59 Ciclo 6 e Ciclo 3, o F1 é de 0.73. Ter essa variação no item "Problema Funcional" é um caso que chama atenção. Esse foi o item mais observado ao longo desse trabalho, pela frequência que apareceu nas diversas classificações e pela abrangência da sua interpretação, o que pode implicar na necessidade de muito mais casos para melhorar a efetiva classificação do BERT nesse item. A Tabela 25 mostra a frequência da classificação dos conjuntos de chamados feitos pelo GPT 4 e somados ao GPT 4 Mini e o resultado da classificação feita pelo BERT treinado com os dados balanceados. Note que já era esperado que haveria alguma distorção por conta da seleção dos dez mais frequentes representar 73,80% mas não ter a classe "Problema Funcional" a frente chama a atenção e é destoante de todas as outras classificações mostradas nesse trabalho.

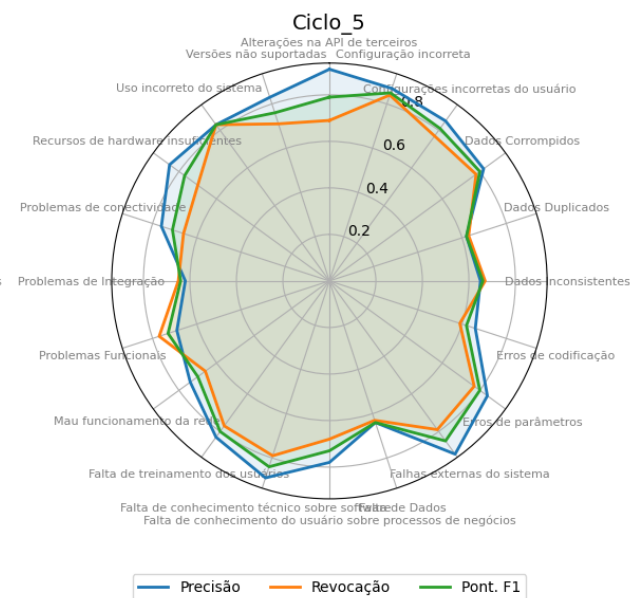
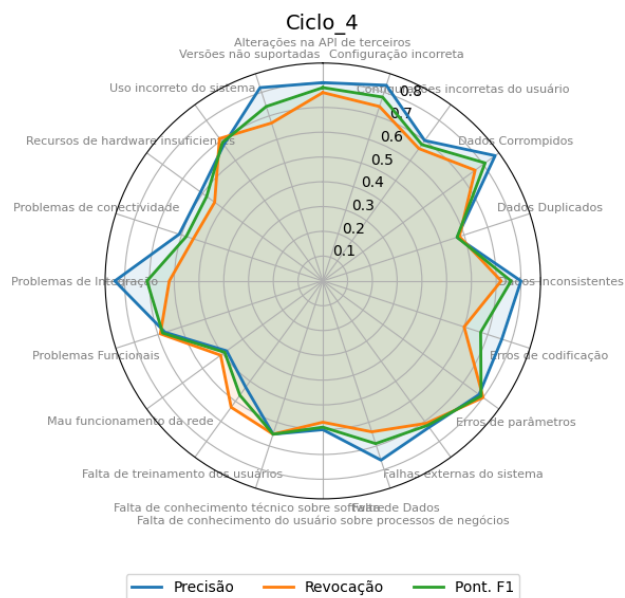
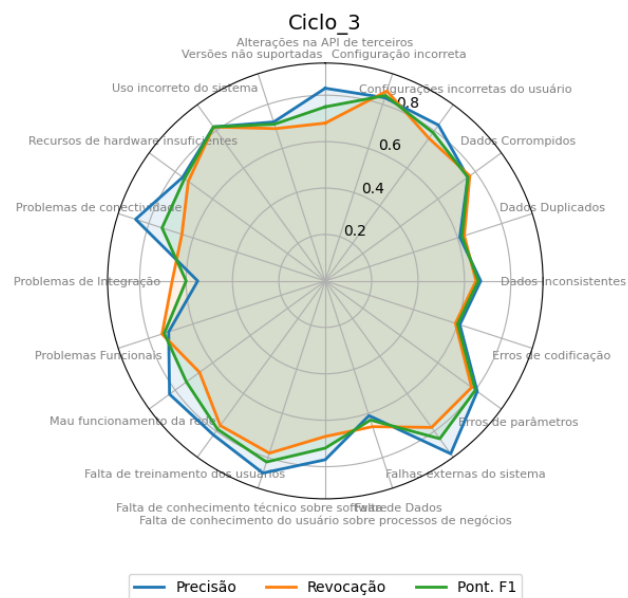


Figura 28 – Gráfico de radar Ciclo 3, 4 e 5

## 6 CONCLUSÃO

O trabalho apresentado traz uma proposta para a análise de chamados de suporte em sistemas de produção, utilizando modelos de linguagem de grande escala (LLMs). Tradicionalmente, a análise de chamados de suporte é realizada de forma manual e individual, dificultando a identificação de padrões e a compreensão das causas raiz dos problemas. O estudo trouxe uma forma de automatizar esse processo, utilizando LLMs para classificar os chamados de acordo com uma taxonomia pré-definida que abrange falhas em sistemas, software, hardware e operação por usuários. O trabalho contou com uma base de dados real de chamados de uma empresa do setor financeiro, abrangendo diversos sistemas e linhas de negócio. Os dados foram tratados e preparados para serem utilizados como entrada para os modelos de linguagem. A classificação foi realizada em duas etapas: inicialmente, um LLM foi utilizado para classificar um subconjunto dos dados, e posteriormente, um modelo BERT foi treinado com essas classificações para aprimorar a precisão. Os resultados desse trabalho podem trazer benefícios interessantes para as empresas no seus processos de negócio de sustentação:

- **Melhoria na identificação das causas raiz dos problemas:** ao classificar os chamados de forma automatizada, é possível identificar rapidamente os principais problemas e suas causas.
- **Otimização dos processos de suporte:** com uma melhor compreensão dos problemas, é possível otimizar os processos de suporte e reduzir o tempo de resolução dos chamados.
- **Tomada de decisões mais informada:** as informações geradas pela análise dos chamados podem ser utilizadas para tomar decisões estratégicas sobre investimentos em tecnologia e treinamento de equipes.
- **Roteamento de chamados:** rotear automaticamente os chamados para os agentes de suporte mais adequados.
- **Resolução automática de problemas:** em alguns casos, ter a capacidade de resolver problemas simples de forma autônoma, reduzindo a carga de trabalho dos agentes de suporte.
- **Criação de bases de conhecimento:** criar bases de conhecimento mais completas e precisas, facilitando a busca por informações relevantes.

O estudo demonstra o potencial dos LLMs para transformar a forma como as empresas lidam com a análise de chamados de suporte, proporcionando conhecimento

valioso para a melhoria contínua dos sistemas e processos. O trabalho teve uma primeira parte qualitativa, que permitiu uma análise mais profunda e detalhada das capacidades e limitações de cada modelo, com análise de casos específicos de classificação, em que os modelos geram resultados diferentes, buscando identificar essa variação e análise de conteúdo gerado pelos modelos para identificar padrões, nuances e diferenças. Também foram aplicadas técnicas como *Bag-Of-Words* e Agrupamento para avaliar se as mensagens de explicação da classificação apresentavam certa coerência. A segunda parte do trabalho visou realizar várias sessões de ajuste fino no modelo de linguagem BERT para buscar usar um modelo alternativo, mais simples e que possa ser usado de forma massificada. Também houve o objetivo de se ter um indicador numérico da precisão da classificação, que são as probabilidades associadas a cada classificação feita pelo BERT.

## 6.1 Comparação GPT 3.5 e GPT 4 na classificação de chamados de suporte

Pode ser percebido no trabalho que modelos de linguagem GPT 3.5 e GPT 4 representam avanços significativos na área de processamento de linguagem natural. Ambos foram capazes de gerar texto relativos aos problemas reportados nos chamados, com diferentes tipos de conteúdo. No entanto, o GPT 4, sendo a versão mais recente, apresenta melhorias substanciais, especialmente em tarefas que exigem um entendimento mais profundo e nuançado da linguagem bem como demonstrou capacidade de inferir melhor com a pilha de erro.

### 6.1.1 Capacidades de classificação de chamados de suporte

A classificação de chamados de suporte é uma tarefa complexa que exige não apenas a identificação de palavras-chave, mas também a compreensão do contexto e da intenção do usuário. Ambos os modelos, GPT 3.5 e GPT 4, demonstram capacidades promissoras nessa área, mas com nuances importantes.

- **GPT 3.5:** Foi razoavelmente bem ao identificar padrões e correlações nas mensagens. Ele pôde classificar chamados aparentemente usando textos específicos, mas teve dificuldade em lidar com a estrutura de dado solicitada para a classificação na taxonomia, o que exigiu um tratamento da resposta do GPT 3.5. Outrossim, mostrou limitação com chamados que exigem mais inferência. O GPT 3.5 teve dificuldades em generalizar problemas, especialmente quando se deparam com chamados ou com expressões idiomáticas, jargões e outros dados, como a pilha de erro.
- **GPT 4:** Demonstrou um entendimento mais profundo do contexto, permitindo-lhe classificar chamados com base não apenas em palavras-chave, mas também no significado geral da mensagem. O modelo foi mais capaz de generalizar problemas, identificando a raiz do problema mesmo quando este é expresso de forma diferente e

o GPT 4 mostrou uma melhor compreensão da semântica do problema reportado, permitindo-lhe identificar sinônimos e diferentes formas de expressar a mesma ideia.

Em resumo, o GPT 4 foi um avanço significativo em relação ao GPT 3.5 na tarefa de classificação de chamados de suporte. Sua capacidade de entender o contexto, generalizar problemas e lidar com a semântica o tornam uma ferramenta mais poderosa e precisa para automatizar processos de atendimento a chamados de sustentação. É notória também a discrepância de classificações feitas pelo GPT 3.5 e o GPT 4. Mesmo com um domínio de dados de chamados muito arenoso, seria de se esperar mais alinhamento entre as classificações desses dois modelos.

#### 6.1.2 Análise das mensagens de explicação da classificação com *Bag-Of-Words* e Agrupamento

Foi feito também uma atividade para analisar as mensagens de explicação geradas pelo modelo. Essas mensagens, que descrevem a lógica por trás da classificação, são uma rica fonte de informação sobre a capacidade de inferência do modelo GPT 4. Foram usadas as mensagens geradas por esse modelo exatamente pelas capacidades ampliadas de processamento em linguagem natural demonstradas no decorrer do trabalho. Para realizar a análise foram utilizadas duas técnicas, *Bag of Words* e **Agrupamento KNN**. Com as duas técnicas, de fato, observou-se a formação de agrupamentos claros e significativos, sendo essa uma abordagem se mostrou também possível para ter uma visão das causas de problemas. Ou seja, a combinação de *Bag of Words* e Agrupamento KNN para analisar as mensagens de explicação do GPT-4 é uma abordagem promissora para entender melhor as explicações e se elas são coerentes com as classificações feitas.

#### 6.1.3 Aplicação da Taxonomia

A combinação de uma taxonomia bem estruturada com a capacidade das LLMs de fornecer explicações para suas classificações resultou em uma classificação de suporte mais preciso, eficiente e robusta, em linha com as validações expostas em (WANG *et al.*, 2022), (DHULIAWALA *et al.*, 2023) e (WEI *et al.*, 2022b). Ao solicitar uma explicação, o modelo demonstra sua compreensão do problema e justifica a classificação escolhida, de forma a permitir identificar e avaliar possíveis erros ou vieses. Um resultado colateral, mas que foi muito importante para o trabalho, foi o uso das mensagens de explicação para ajudar no ajuste fino do modelo, o que encoraja o uso desse método em outras circunstâncias.

### 6.2 Análise Quantitativa com aplicação do modelo BERT

Esta parte do trabalho teve como objetivo avaliar a eficácia do modelo de linguagem pré-treinado BERT, com ajuste fino, na tarefa de classificação de chamados de suporte. O foco principal foi ter a aplicação de um modelo que possibilitasse visualizar as probabilidades

de classificação, características que o diferenciam do uso de modelos de linguagem de grande porte como o GPT-4. Também foi analisado o fenômeno do *overfitting* e formas de contorná-lo nesse domínio de conhecimento e dados. Utilizou-se um conjunto de dados real de chamados de suporte técnico, contendo informações de descrição do problema complementado com a mensagens de explicação da classificação feita pelo GPT 4, sendo esses dados sintéticos. Vale ressaltar que essas mensagens, mesmo sendo geradas pela LLM, continham parte do texto original do chamado, o que deu mais proximidade com uma mensagem real. O modelo BERT foi escolhido por sua capacidade de capturar o contexto das palavras em um texto e com o conjunto de dados de chamados foi feito o ajuste fino, refinando seus parâmetros para a tarefa específica de classificação. Para avaliar o *overfitting*, foram utilizadas técnicas como validação cruzada e curvas de aprendizado. Por fim o modelo BERT, ao contrário de modelos como GPT-4, permite a visualização das probabilidades atribuídas a cada classe, o que auxilia na compreensão da incerteza da classificação.

### 6.2.1 Resultados

O experimento demonstrou que o modelo BERT, com ajuste fino, dados não balanceados, reais e somados aos dados sintéticos, num total 31.123 chamados - ciclo (3) -, apresentou um desempenho geral consistente, com F1-Score acima de 0.7 para a maioria das classes. Um fato interessante foi que o modelo (4) com *dropout*, ajuste de L2, e com texto da classificação removido do corpo da descrição da mensagem de explicação (dado sintético) apresenta um desempenho mais estável, com menos variação entre as classes como pode ser observado na Tabela 35, mas um pouco pior que o modelo do ciclo (3) nas métricas de Acuracidade, Revocação e F1. A tabela traz uma síntese geral dos quatro ciclos de ajuste fino e suas características, incluindo os primeiros ciclos de treinamento, que foram muito infrutíferos. O modelo do ciclo (3), ao observar-se as métricas de Jaccard e Sorensen-Dice, mostrou um desempenho razoável - até melhor que o GPT 3.5 - na classificação de chamados, ao comparar-se a classificação feita pelo GPT 4. Finalmente, o quinto modelo (Ciclo 5), com LR 2e-5, 10 épocas, com *dropout*, L2, mas agora permitindo até 256 palavras com vocabulário estendido para 15.000 elementos e treinados no mesmo conjunto de 31.123 chamados foi o que teve as métricas de Acurácia, Revocação e F1 melhores, como evidenciado na Figura 28. No entanto, em todos os ciclos de treinamento, o *overfitting* foi um problema resistente. O *overfitting* é um desafio comum em modelos de aprendizado de máquina e houve a tentativa de mitigar através de técnicas como regularização, *dropout* e *early stopping*, mas não houve melhora aplicando esses dois métodos. Além disso, foi aplicada a técnicas de aumento de dados para ajudar a aumentar a generalização do modelo, com o uso de dados sintéticos das próprias mensagens de explicação. Essa ação resultou de fato em uma melhora dos números de acuracidade do modelo após o ajuste fino.

O uso de dados sintéticos é geralmente controverso e pode limitar a capacidade de generalização dos modelos de linguagem. Contudo, para esse trabalho, o uso de dados sintéticos, mas que continham parte do texto real, permitiu um avanço no ajuste fino do BERT. Longe do ideal, mas dentro de um cenário possível, essa estratégia ajudou a ter um modelo melhor ajustado e que trouxe um resultado razoável para o domínio de mensagens e contexto.

Uma descrição de chamados é muito variada, sendo um tema discutido e mostrado extensivamente no decorrer do trabalho. Isso parece ser um fator de dispersão para uma classificação eficaz. Diversidade, falta de estrutura e textos tão díspares são um desafio mesmo para o olhar e interpretação humano. O trabalho mostrou que a classificação de chamados por uma LLM poderosa, como o GPT 4, de fato aproxima-se da capacidade de uma pessoa para entender o problema em uma triagem inicial. Técnicas complementares como *cluster* e *bag-of-words*, se bem ajustadas dentro dos processos de resolução de chamados, podem oferecer formas complementares de análise eficazes e econômicas. O trabalho demonstrou que o modelo BERT, com ajuste fino, é uma ferramenta razoável para a classificação de chamados de suporte, obtendo-se uma concordância média entre GPT 4 e BERT. A capacidade de visualizar as probabilidades de classificação é uma vantagem significativa em relação a outros modelos de linguagem. No entanto, é fundamental adotar medidas para mitigar o *overfitting* e garantir a generalização do modelo para que as taxas de similaridade com modelos como o GPT4 seja maiores.

### 6.2.2 Recomendações para trabalhos futuros

Este trabalho abre caminho para futuras pesquisas na área de classificação de chamados utilizando modelos de linguagem pré-treinados no domínio de conhecimento de chamados de suporte. A compreensão das vantagens e limitações desses modelos é fundamental para o desenvolvimento de soluções mais eficazes nessa arena. São exemplos de trabalhos futuros:

- Avaliar se a ampliação massiva de dados de treinamento, combinados com dados sintéticos, pode trazer mais acuracidade ao modelo BERT bem como fazer um pré-processamento nas descrições dos chamados, para deixá-los somente com a mensagem principal mais consistente e com menos informação desnecessária;
- Explorar diferentes arquiteturas de BERT e variações de treinamento, com mais palavras de entrada e vocabulário mais ampliado;
- Realizar variações das técnicas de aumento de dados para melhorar a generalização do modelo, com dados sintéticos gerados individualmente, a partir da própria mensagem de classificação, e expandidos por uma LLM mais capaz como GPT 4;

- Analisar e comparar o resultado das classificações feita pelo BERT a luz da concentração ou dispersão das probabilidades associadas, sendo essas um potencial fator ponderador e avançar na aplicação de técnicas estatísticas de validação dos resultados.
- Trabalhar forma combinadas entre modelos de LLM e modelos como BERT nos processos de suporte para ter maior eficiência com perfil de custos mais adequados;
- Forma de classificação de chamados e extração de informação ampliada, considerando a pilha de mensagens, para um eventual direcionamento eficaz de solução e até forma de auto-correção para erros mais simples.



## REFERÊNCIAS

- ACHIAM, J. *et al.* Gpt-4 technical report. **arXiv preprint arXiv:2303.08774**, 2023.
- ARAUJO, A. F.; GÔLO, M. P.; MARCACINI, R. M. Opinion mining for app reviews: an analysis of textual representation and predictive models. **Automated Software Engineering**, Springer, v. 29, p. 1–30, 2022.
- BOMMASANI, R. *et al.* On the opportunities and risks of foundation models. **arXiv preprint arXiv:2108.07258**, 2021.
- BROWN, T. *et al.* Language models are few-shot learners. **Advances in neural information processing systems**, v. 33, p. 1877–1901, 2020.
- BUDA, M.; MAKI, A.; MAZUROWSKI, M. A. A systematic study of the class imbalance problem in convolutional neural networks. **Neural networks**, Elsevier, v. 106, p. 249–259, 2018.
- CARVALHO, V. M. de *et al.* Lei geral de proteção de dados e direito da concorrência.
- CHILLAREGE, R. *et al.* Orthogonal defect classification-a concept for in-process measurements. **IEEE Transactions on software Engineering**, v. 18, n. 11, p. 943–956, 1992.
- CHOWDHERRY, A. *et al.* Palm: Scaling language modeling with pathways. **Journal of Machine Learning Research**, v. 24, n. 240, p. 1–113, 2023.
- CHUNG, H. W. *et al.* Scaling instruction-finetuned language models. **arXiv preprint arXiv:2210.11416**, 2022.
- DEVLIN, J. *et al.* Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.
- DHULIAWALA, S. *et al.* Chain-of-verification reduces hallucination in large language models. **arXiv preprint arXiv:2309.11495**, 2023.
- FELDMAN, R. Techniques and applications for sentiment analysis. **Communications of the ACM**, ACM New York, NY, USA, v. 56, n. 4, p. 82–89, 2013.
- GARTNER. **Enterprise Agile Planning Tools Reviews and Rating**. 2022. Available at: <https://www.gartner.com/reviews/market/enterprise-agile-planning-tools>.
- GIGCH, J. P. V. Modeling, metamodeling, and taxonomy of system failures. **IEEE transactions on reliability**, IEEE, v. 35, n. 2, p. 131–136, 1986.
- GU, Y. *et al.* Ppt: Pre-trained prompt tuning for few-shot learning. **arXiv preprint arXiv:2109.04332**, 2021.
- GUARINO, N.; WELTY, C. Ontological analysis of taxonomic relationships. *In*: SPRINGER. **International Conference on Conceptual Modeling**. [S.l.: s.n.], 2000. p. 210–224.

- HAIXIANG, G. *et al.* Learning from class-imbalanced data: Review of methods and applications. **Expert systems with applications**, Elsevier, v. 73, p. 220–239, 2017.
- HAN, X. *et al.* Ptr: Prompt tuning with rules for text classification. **AI Open**, Elsevier, v. 3, p. 182–192, 2022.
- HOFFMANN, J. *et al.* Training compute-optimal large language models. **arXiv preprint arXiv:2203.15556**, 2022.
- HOWARD, J.; RUDER, S. Universal language model fine-tuning for text classification. **arXiv preprint arXiv:1801.06146**, 2018.
- IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In: PMLR. International conference on machine learning*. [S.l.: s.n.], 2015. p. 448–456.
- IYER, S. *et al.* Opt-impl: Scaling language model instruction meta learning through the lens of generalization. **arXiv preprint arXiv:2212.12017**, 2022.
- JANCZAREK, P.; SOSNOWSKI, J. Investigating software testing and maintenance reports: Case study. **Information and Software Technology**, Elsevier, v. 58, p. 272–288, 2015.
- JESSE, K. *et al.* Large language models and simple, stupid bugs. **arXiv preprint arXiv:2303.11455**, 2023.
- KAPLAN, J. *et al.* Scaling laws for neural language models. **arXiv preprint arXiv:2001.08361**, 2020.
- KRUMM, J.; DAVIES, N.; NARAYANASWAMI, C. User-generated content. **IEEE Pervasive Computing**, IEEE, v. 7, n. 4, p. 10–11, 2008.
- LE, V.-H.; ZHANG, H. Log parsing: How far can chatgpt go? *In: IEEE. 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. [S.l.: s.n.], 2023. p. 1699–1704.
- LEEVY, J. L. *et al.* A survey on addressing high-class imbalance in big data. **Journal of Big Data**, Springer, v. 5, n. 1, p. 1–30, 2018.
- LESTER, B.; AL-RFOU, R.; CONSTANT, N. The power of scale for parameter-efficient prompt tuning. **arXiv preprint arXiv:2104.08691**, 2021.
- LI, X. L.; LIANG, P. Prefix-tuning: Optimizing continuous prompts for generation. **arXiv preprint arXiv:2101.00190**, 2021.
- LI, Z. *et al.* Synthetic data generation with large language models for text classification: Potential and limitations. **arXiv preprint arXiv:2310.07849**, 2023.
- LIU, J. *et al.* Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. **Advances in Neural Information Processing Systems**, v. 36, 2024.
- LIU, P. *et al.* Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. **ACM Computing Surveys**, ACM New York, NY, v. 55, n. 9, p. 1–35, 2023.

LIU, X. *et al.* Gpt understands, too. **AI Open**, Elsevier, 2023.

LIVENTSEV, V. *et al.* Fully autonomous programming with large language models. **arXiv preprint arXiv:2304.10423**, 2023.

LONGARAY, A. A. *et al.* Study about the evaluation of internet banking and mobile banking users' satisfaction in brazil. **REMark**, Universidade Nove de Julho (UNINOVE), PPGA, v. 20, n. 1, p. 27, 2021.

MADABUSHI, H. T.; KOCHKINA, E.; CASTELLE, M. Cost-sensitive bert for generalisable sentence classification with imbalanced data. **arXiv preprint arXiv:2003.11563**, 2020.

MARCACINI, R. M. *et al.* Cross-domain aspect extraction for sentiment analysis: A transductive learning approach. **Decision Support Systems**, Elsevier, v. 114, p. 70–80, 2018.

NAAB, T. K.; SEHL, A. Studies of user-generated content: A systematic review. **Journalism**, SAGE Publications Sage UK: London, England, v. 18, n. 10, p. 1256–1273, 2017.

PENALVER-MARTINEZ, I. *et al.* Feature-based opinion mining through ontologies. **Expert Systems with Applications**, Elsevier, v. 41, n. 13, p. 5995–6008, 2014.

RADFORD, A. *et al.* Language models are unsupervised multitask learners. **OpenAI blog**, v. 1, n. 8, p. 9, 2019.

RALPH, P. Toward methodological guidelines for process theories and taxonomies in software engineering. **IEEE Transactions on Software Engineering**, IEEE, v. 45, n. 7, p. 712–735, 2018.

REES, R. V. Clarity in the usage of the terms ontology, taxonomy and classification. **Cib Report**, CIB, v. 284, n. 432, p. 1–8, 2003.

REZENDE, S. O. **Sistemas inteligentes: fundamentos e aplicações**. [S.l.: s.n.]: Editora Manole Ltda, 2003.

REZENDE, S. O.; MARCACINI, R. M.; MOURA, M. F. O uso da mineração de textos para extração e organização não supervisionada de conhecimento. **Revista de Sistemas de Informação da FSMA**, n. v. 7, n. 1, p. 7–21, 2011.

ROSSI, R. G.; MARCACINI, R. M.; REZENDE, S. O. Benchmarking text collections for classification and clustering tasks. 2013.

ROSSI, R. G.; MARCACINI, R. M.; REZENDE, S. O. Analysis of domain independent statistical keyword extraction methods for incremental clustering. **Learning and Nonlinear Models**, v. 12, n. 1, p. 17–37, 2014.

SAHIBUDIN, S.; SHARIFI, M.; AYAT, M. Combining itil, cobit and iso/iec 27002 in order to design a comprehensive it framework in organizations. *In*: IEEE. **2008 Second Asia International Conference on Modelling & Simulation (AMS)**. [S.l.: s.n.], 2008. p. 749–753.

SANH, V. *et al.* Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. **arXiv preprint arXiv:1910.01108**, 2019.

SCHAEFFER, R.; MIRANDA, B.; KOYEJO, S. Are emergent abilities of large language models a mirage? **Advances in Neural Information Processing Systems**, v. 36, 2024.

SCHICK, T.; SCHÜTZE, H. Exploiting cloze questions for few shot text classification and natural language inference. **arXiv preprint arXiv:2001.07676**, 2020.

SCHICK, T.; SCHÜTZE, H. It's not just size that matters: Small language models are also few-shot learners. **arXiv preprint arXiv:2009.07118**, 2020.

SCHICK, T.; SCHÜTZE, H. Few-shot text generation with natural language instructions. *In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. [S.l.: s.n.], 2021. p. 390–402.

SILVA, E. H. d.; MARCACINI, R. M. Aspect-based sentiment analysis using bert with disentangled attention. *In: Proceedings*. [S.l.: s.n.], 2021.

SMITH, L. N. Cyclical learning rates for training neural networks. *In: IEEE. 2017 IEEE winter conference on applications of computer vision (WACV)*. [S.l.: s.n.], 2017. p. 464–472.

SMITH, L. N.; TOPIN, N. Super-convergence: Very fast training of neural networks using large learning rates. *In: SPIE. Artificial intelligence and machine learning for multi-domain operations applications*. [S.l.: s.n.], 2019. v. 11006, p. 369–386.

SÖYLEMEZ, M.; TARHAN, A. Challenges of software process and product quality improvement: catalyzing defect root-cause investigation by process enactment data analysis. **Software Quality Journal**, Springer, v. 26, p. 779–807, 2018.

SRIDHARA, G.; MAZUMDAR, S. *et al.* Chatgpt: A study on its utility for ubiquitous software engineering tasks. **arXiv preprint arXiv:2305.16837**, 2023.

SRIVASTAVA, N. *et al.* Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

SUN, W. *et al.* Automatic code summarization via chatgpt: How far are we? **arXiv preprint arXiv:2305.12865**, 2023.

TANG, R. *et al.* Does synthetic data generation of llms help clinical text mining? **arXiv preprint arXiv:2303.04360**, 2023.

USMAN, M. *et al.* Taxonomies in software engineering: A systematic mapping study and a revised taxonomy development method. **Information and Software Technology**, Elsevier, v. 85, p. 43–59, 2017.

VAZIRGIANNIS, M.; MALLIAROS, F. D.; NIKOLENTZOS, G. Graphrep: boosting text mining, nlp and information retrieval with graphs. *In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. [S.l.: s.n.], 2018. p. 2295–2296.

- 
- WANG, X. *et al.* Self-consistency improves chain of thought reasoning in language models. **arXiv preprint arXiv:2203.11171**, 2022.
- WEI, J. *et al.* Finetuned language models are zero-shot learners. **arXiv preprint arXiv:2109.01652**, 2021.
- WEI, J. *et al.* Emergent abilities of large language models. **arXiv preprint arXiv:2206.07682**, 2022.
- WEI, J. *et al.* Chain-of-thought prompting elicits reasoning in large language models. **Advances in Neural Information Processing Systems**, v. 35, p. 24824–24837, 2022.
- WEISS, S. M.; INDURKHYA, N.; ZHANG, T. **Fundamentals of predictive text mining**. [*S.l.: s.n.*]: Springer, 2015.
- WENG, G.; ANDRZEJAK, A. Automatic bug fixing via deliberate problem solving with large language models. *In*: IEEE. **2023 IEEE 34th International Symposium on Software Reliability Engineering Workshops (ISSREW)**. [*S.l.: s.n.*], 2023. p. 34–36.
- WYRWOLL, C.; WYRWOLL, C. **User-generated content**. [*S.l.: s.n.*]: Springer, 2014.
- ZHANG, Q. *et al.* A critical review of large language model on software engineering: An example from chatgpt and automated program repair. **arXiv preprint arXiv:2310.08879**, 2023.
- ZHAO, W. X. *et al.* A survey of large language models. **arXiv preprint arXiv:2303.18223**, 2023.



## **ANEXOS**





**ANEXO A – ANEXOS**

“ “

Chamado	Resultado GPT 4
<p>O sistema está gerando cálculo de movimentos retroativos ao clonar projeção produto por terem a mesma base de partida de cenário 30112022 porém para projetar o mês seguinte o sistema deve desprezar as movimentações anteriores e iniciar o processamento da base realizada cargas uma vez que sempre será clonada a opção projeção produto O ticket SD162723 IN9040941 Erro de projeção com movimentação retroativa com clone de Projeção Produto partindo de cenário com mesma data base O ticket foi criado pela organização Seguradora para o produto FundosProjeção Financeira A descrição do ticket é O sistema está gerando cálculo de movimentos retroativos ao clonar projeção produto por terem a mesma base de partida de cenário 30112022 porém para projetar o mês seguinte o sistema deve desprezar as movimentações anteriores e iniciar o processamento da base realizada cargas uma vez que sempre será clonada a opção projeção produto O ticket foi criado por valeriamelobradescoseguroscombr em 20230825 1634350 O status atual do ticket é Em Andamento Resumo do tempo em cada status Em Andamento 0 horas 001 Novo 0 horas 00 Triagem 0 horas 00</p>	<p>["['Alto','1 Técnica','113 Problemas Funcionais','O documento descreve um problema funcional no sistema que gera cálculos indevidos de movimentos retroativos ao clonar projeções.'],'",["['Médio','2 Ambiente','215 Configuração incorreta','O problema pode estar relacionado a uma configuração incorreta no sistema que não está descartando as movimentações anteriores conforme necessário.'],'",["['Baixo','3 Humano','312 Falta de treinamento dos usuários','Embora não seja o foco principal, pode haver uma falta de conhecimento dos usuários sobre como o sistema deve operar para evitar esse tipo de problema.']]"]</p>
<p>Bom dia Estamos calculando a folha e está calculando imposto com valor abaixo de 1000 reais por favor poderiam verificar Anexo exemplo Atenciosamente Talita O ticket SD162639 Imposto abaixo de 1000 reais O ticket foi criado pela organização Energia para o produto Produto de Benefício A descrição do ticket é Bom dia Estamos calculando a folha e está calculando imposto com valor abaixo de 1000 reais por favor poderiam verificar Anexo exemplo Atenciosamente Talita O ticket foi criado por talitagarciaenergiasaprevcombr em 20230825 1003230 Comentário criado por talitagarciaenergiasaprevcombr em Fri Aug 25 100329 BRT 2023 imagepngthumbnail Comentário criado por Empresa em Fri Aug 25 164619 BRT 2023 Motivo Aguardando Evidência Talita boa tarde Poderia informar qual versão do SPB vocês estão utilizando e também nos enviar o trace que você ativa no momento de calcular a folha do participante Além disso pode nos enviar também um print da tela do Corporativo menu APOIO IRRF Faixa salarial do IRRF Att Rafaela Comentário criado por talitagarciaenergiasaprevcombr em Fri Aug 25 165622 BRT 2023 Boa tarde Versão 08150008003</p>	<p>["['Alto','1 Técnica','113 Problemas Funcionais','O documento menciona que o cálculo do imposto na folha está apresentando valores abaixo do esperado, indicando um problema funcional no sistema de cálculo.'],'",["['Médio','1 Técnica','114 Versões não suportadas','Foi solicitado à usuária que informe a versão do software SPB, sugerindo que a versão pode estar relacionada ao problema.'],'",["['Médio','3 Humano','311 Uso incorreto do sistema','Foi solicitado um trace e um print da tela de configurações do IRRF, indicando que pode haver um uso incorreto ou configuração inadequada do sistema.']]"]</p>

Tabela 26 – Exemplo de Chamado e Resultado GPT-4 Turbo

Item	Contagem
113 Problemas Funcionais	619
112 Problemas de Integração	191
111 Erros de codificação	174
134 Dados Inconsistentes	135
313 Configurações incorretas do usuário	96
132 Dados Corrompidos	95
133 Dados Duplicados	92
311 Uso incorreto do sistema	61
131 Falta de Dados	61
312 Falta de treinamento dos usuários	56
324 Erros de parâmetros	55
122 Problemas de conectividade	47
422 Acesso Não Autorizado	41
323 Problemas de instalação	36
314 Falta de conhecimento do usuário sobre processos de negócios	34
115 Versão de software desatualizada	33
322 Problemas de configuração	27
325 Falta de conhecimento técnico sobre software	14
213 Migração de Banco de Dados	13
321 Operação por Técnico Não Qualificado	9
116 Software obsoleto	8
326 Falta de conhecimento técnico sobre hardware	8
211 Migração de Servidor	8
421 Violações de dados	5
121 Mau funcionamento da rede	5
423 Vazamento de dados	5
114 Versões não suportadas	4
216 Configuração incorreta	4
123 Recursos de hardware insuficientes	4
222 Alterações em software de terceiros	4
111 Erros de Codificação	3
223 Falhas externas do sistema	3
212 Versão do Banco de Dados	3
411 Malware	2
214 Versão do Sistema Operacional	2
124 Outras falhas de hardware	2
412 Negação de Serviço (DoS)	1
221 Alterações na API de terceiros	1
323 Problemas de configuração	1

Tabela 27 – Contagem de Problemas GPT-4

Item	Contagem
113 Problemas Funcionais	662
114 Versões não suportadas	222
115 Versão de software desatualizada	160
133 Dados Duplicados	159
116 Software obsoleto	107
131 Falta de Dados	104
111 Erros de codificação	93
112 Problemas de Integração	76
211 Migração de Servidor	43
212 Versão do banco de dados	35
312 Falta de treinamento dos usuários	32
132 Dados Corrompidos	31
134 Dados Inconsistentes	26
311 Uso incorreto do sistema	24
423 Vazamento de dados	21
214 Versão do Sistema Operacional	20
215 Configuração incorreta	20
121 Mau funcionamento da rede	20
313 Configurações incorretas do usuário	16
123 Recursos de hardware insuficientes	15
213 Migração de Banco de Dados	13
122 Problemas de conectividade	12
421 Violações de dados	11
321 Operação por Técnico Não Qualificado	6
322 Problemas de configuração	5
314 Falta de conhecimento do usuário sobre processos de negócios	5
124 Outras falhas de hardware	4
221 Alterações na API de terceiros	3
325 Falta de conhecimento técnico sobre software	2
422 Acesso Não Autorizado	2
Não há menção de vazamento de dados no ticket	1
111 Erros de Codificação	1
O documento descreve problemas na forma como as operações estão sendo unificadas dentro do xml o que indica possíveis problemas funcionais no sistema	1
223 Falhas externas do sistema	1
136 Software obsoleto	1
O texto indica que houve um problema com a configuração para o disparo automático das carteiras o que pode estar relacionado a recursos de hardware insuficientes que impossibilitaram o envio	1
135 Inconsistência no Cadastro	1
326 Falta de conhecimento técnico sobre hardware	1
134 Dados Inconsistentes	1
Problemas Funcionais	1
116 Software Obsoleto	1
Versões não suportadas	1
412 Negação de Serviço (DoS)	1
114 Versão de software desatualizada	1
O texto descreve claramente um problema funcional na rotina de fechamento do dia que resultou em um erro de consulta de dados no INFOBANK. Isso impactou o fechamento do SINQIA SGR	1
212 Versão do Banco de Dados	1
141 Versão de software desatualizada	1
135 Dados Inconsistentes	1
Unknown	0

Tabela 28 – Contagem de Problemas GPT-3.5

Item	Contagem
115 Versão de software desatualizada	312
133 Dados Duplicados	255
131 Falta de Dados	206
114 Versões não suportadas	162
116 Software obsoleto	130
132 Dados Corrompidos	122
111 Erros de codificação	116
121 Mau funcionamento da rede	100
112 Problemas de Integração	90
212 Versão do banco de dados	55
211 Migração de Servidor	53
134 Dados Inconsistentes	51
123 Recursos de hardware insuficientes	31
312 Falta de treinamento dos usuários	27
214 Versão do Sistema Operacional	24
213 Migração de Banco de Dados	24
122 Problemas de conectividade	22
423 Vazamento de dados	21
313 Configurações incorretas do usuário	19
215 Configuração incorreta	15
311 Uso incorreto do sistema	15
124 Outras falhas de hardware	13
113 Problemas de Integração	10
421 Violações de dados	10
314 Falta de conhecimento do usuário sobre proc...	7
322 Problemas de configuração	6
422 Acesso Não Autorizado	6
221 Alterações na API de terceiros	5
412 Negação de Serviço (DoS)	3
321 Operação por Técnico Não Qualificado	3
111 Erros de Codificação	3
113 Falta de Dados	2
411 Malware	2
Erros de codificação	2
324 Erros de parâmetros	2
113 Versões não suportadas	2
115 Versão do software desatualizada	1
Migração de Servidor	1
323 Problemas de instalação	1
113 Versão de software desatualizada	1
222 Alterações em software de terceiros	1
113 Processo Travado	1
216 Configuração incorreta	1
114 Versões não suportadas	1
113 Valor de lançamento incorreto	1
	1
111 Erro de codificação	1
135 Falta de Dados	1
115 Falta de Dados	1
2 Ambiente	1
135 Falha na Atualização de Saldo	1
315 Falta de conhecimento do usuário sobre proc...	1
A necessidade de realizar manualmente a extraçã...	1
113 Dados Inconsistentes	1
121 Erros de codificação	1
116 Software Obsoleto	1
325 Falta de conhecimento técnico sobre software	1
None	0

Tabela 29 – Contagem de Itens Classificados pelo GPT 3.5 sem o Item *113 Problemas Funcionais*

Item	Contagem
132 Dados Corrompidos	282
111 Erros de codificação	249
134 Dados Inconsistentes	235
131 Falta de Dados	170
311 Uso incorreto do sistema	157
112 Problemas de Integração	148
133 Dados Duplicados	97
121 Mau funcionamento da rede	97
122 Problemas de conectividade	82
215 Configuração incorreta	80
114 Versões não suportadas	68
123 Recursos de hardware insuficientes	41
115 Versão de software desatualizada	35
312 Falta de treinamento dos usuários	30
223 Falhas externas do sistema	25
313 Configurações incorretas do usuário	24
124 Outras falhas de hardware	23
213 Migração de Banco de Dados	21
314 Falta de conhecimento do usuário sobre proc...	20
211 Migração de Servidor	19
324 Erros de parâmetros	14
323 Problemas de instalação	12
221 Alterações na API de terceiros	12
322 Problemas de configuração	10
212 Versão do banco de dados	9
214 Versão do Sistema Operacional	7
321 Operação por Técnico Não Qualificado	6
325 Falta de conhecimento técnico sobre software	6
422 Acesso Não Autorizado	5
222 Alterações em software de terceiros	3
423 Vazamento de dados	3
421 Violações de dados	2
116 Software obsoleto	2
113 Dados Inconsistentes	1
113 Erros de configuração	1
326 Falta de conhecimento técnico sobre software	1
313 Configurações incorretas	1
411 Malware	1
133 Dados Inconsistentes	1
None	0

Tabela 30 – Contagem de Itens Classificados pelo GPT 4 sem o Item 113 *Problemas Funcionais*

Ticket	Taxonomia Sem 113	Taxonomia Com 113
<p>SD162754 Erro no relatório Estatístico O ticket foi criado pela organização Pro para o produto PIX Tree A descrição do ticket é Solicito correção para a data passada para as procedures dos 2 relatórios estatísticos Hoje a data passada é no formato AAAADIAMES e para data em formato ISO está incorreto É necessário modificar para AAAAMESDIA para que as procedures aceitem a data corretamente Referente a incidente 4471908 O ticket foi criado por rogernerissysunicombr em 20230825 2003150 Comentário criado por rogernerissysunicombr em Fri Aug 25 200432 BRT 2023 Favor corrigir a identificação do menu para Relatórios no plural porque são 2 relatórios O status atual do ticket é Novo Resumo do tempo em cada status Novo 0 horas 00</p>	<p>[[‘Alto’, ‘1 Técnica’, ‘111 Erros de codificação’, ‘O ticket descreve um problema de formatação de data que precisa ser corrigido nas procedures dos relatórios estatísticos, indicando um erro no código.’], [‘Médio’, ‘1 Técnica’, ‘132 Dados Corrompidos’, ‘O problema de formato de data incorreto pode resultar em dados corrompidos ao serem passados para as procedures.’], [‘Médio’, ‘3 Humano’, ‘311 Uso incorreto do sistema’, ‘A solicitação inclui um pedido de mudança na interface do usuário (menu para Relatórios no plural), o que pode estar relacionado ao uso inadequado do sistema ou design confuso.’]]</p>	<p>[[‘Alto’, ‘1 Técnica’, ‘113 Problemas Funcionais’, ‘O documento descreve um erro funcional relacionado ao formato incorreto da data passada para as procedures dos relatórios estatísticos. Isso impede que o software funcione corretamente.’], [‘Médio’, ‘1 Técnica’, ‘134 Dados Inconsistentes’, ‘A mudança no formato da data de AAAADIAMES para AAAAMESDIA é necessária para garantir a consistência dos dados nos relatórios estatísticos.’], [‘Baixo’, ‘3 Humano’, ‘313 Configurações incorretas do usuário’, ‘Embora mencione a necessidade de correção na identificação do menu para "Relatórios" no plural, este é um problema menor e facilmente resolvível com uma configuração correta.’]]</p>
<p>SD162753 APISGR Erro na consulta PIX no cadastro de boletos ITBRRTB011224 O ticket foi criado pela organização Banco XPTO Brasil para o produto SGR Gerenciamento de Recebíveis A descrição do ticket é Erro na consulta PIX no cadastro de boletos Log em anexo O ticket foi criado por Ricardo Sarcedo de Mesquita em 20230825 1914040 Comentário criado por Ricardo Sarcedo de Mesquita em Fri Aug 25 191410 BRT 2023 imagepngthumbnail O status atual do ticket é Novo Resumo do tempo em cada status Novo 0 horas 00</p>	<p>[[‘Alto’, ‘1 Técnica’, ‘111 Erros de codificação’, ‘O documento descreve um erro específico na consulta PIX no cadastro de boletos, o que indica um problema técnico de codificação.’], [‘Médio’, ‘3 Humano’, ‘312 Falta de treinamento dos usuários’, ‘Embora menos provável, a falta de treinamento ou conhecimento do usuário sobre o uso correto do sistema poderia ser uma causa subjacente para o erro relatado.’]]</p>	<p>[[‘Alto’, ‘1 Técnica’, ‘113 Problemas Funcionais’, ‘O documento descreve um erro específico relacionado à consulta PIX no cadastro de boletos, o que configura um problema funcional do sistema.’], [‘Médio’, ‘3 Humano’, ‘311 Uso incorreto do sistema’, ‘Considerando que o problema possa ter sido reportado por um usuário que não seguiu os procedimentos corretos, embora isso não esteja explicitamente mencionado no documento’], [‘Baixo’, ‘2 Ambiente’, ‘221 Alterações na API de terceiros’, ‘É possível, embora menos provável, que o erro esteja relacionado a alterações externas, como na API de terceiros usada para integração’], [‘Baixo’, ‘4 Segurança’, ‘422 Acesso Não Autorizado’, ‘Não há menção de problemas de segurança como acesso não autorizado ou violação de dados no documento’]]</p>
<p>SD162751 Processo 392 Tabela de Faixa por Faixa Não está registrando inclusão O ticket foi criado pela organização Banco Duck para o produto Controle de Ativos e Passivos A descrição do ticket é Processo 392 Tabela de Faixa por Faixa Não está registrando inclusão O ticket foi criado por pricardobbbcombr em 20230825 1820200 O status atual do ticket é Novo Resumo do tempo em cada status Novo 0 horas 00</p>	<p>[[‘Médio’, ‘1 Técnica’, ‘131 Falta de Dados’, ‘O ticket descreve que a "Tabela de Faixa por Faixa" não está registrando inclusão, o que pode estar relacionado à falta de dados.’], [‘Médio’, ‘1 Técnica’, ‘132 Dados Corrompidos’, ‘Embora não explicitamente mencionado, a não inclusão de dados pode ser um indicativo de dados corrompidos.’], [‘Alto’, ‘3 Humano’, ‘311 Uso incorreto do sistema’, ‘A origem do problema pode estar relacionada ao uso incorreto do sistema por parte dos usuários, especialmente porque a descrição é vaga.’]]</p>	<p>[[‘Alto’, ‘1 Técnica’, ‘113 Problemas Funcionais’, ‘O ticket relata que o processo "Tabela de Faixa por Faixa" não está registrando inclusão, caracterizando um problema funcional.’]]</p>

Tabela 31 – Exemplos de Classificação com e sem 113 *Problemas Funcionais*

Item	Contagem
113 Problemas Funcionais	1612
311 Uso incorreto do sistema	440
134 Dados Inconsistentes	429
131 Falta de Dados	386
112 Problemas de Integração	323
133 Dados Duplicados	271
111 Erros de codificação	213
132 Dados Corrompidos	166
114 Versões não suportadas	128
122 Problemas de conectividade	118
314 Falta de conhecimento do usuário sobre pro...	92
215 Configuração incorreta	77
312 Falta de treinamento dos usuários	68
313 Configurações incorretas do usuário	65
123 Recursos de hardware insuficientes	44
221 Alterações na API de terceiros	43
325 Falta de conhecimento técnico sobre software	42
213 Migração de Banco de Dados	39
324 Erros de parâmetros	39
121 Mau funcionamento da rede	35
115 Versão de software desatualizada	29
211 Migração de Servidor	28
223 Falhas externas do sistema	28
322 Problemas de configuração	19
422 Acesso Não Autorizado	18
323 Problemas de instalação	14
421 Violações de dados	12
214 Versão do Sistema Operacional	10
321 Operação por Técnico Não Qualificado	9
212 Versão do banco de dados	8
124 Outras falhas de hardware	6
423 Vazamento de dados	6
222 Alterações em software de terceiros	4
412 Negação de Serviço (DoS)	3
116 Software obsoleto	2
133 Dados Inconsistentes	2
O documento menciona um alinhamento prévio par...	1
411 Malware	1
323 Problemas de Instalação	1
O documento relata um problema relacionado ao ...	1
O documento menciona uma solicitação no email ...	1
326 Falta de conhecimento técnico sobre software	1
A solicitação de adição de um novo tipo de rec...	1
None	0

Tabela 32 – Contagem de classificação do segundo lote para aprendizado modelo BERT



Época	Perda	Acurácia (%)	Perda de Val.	Acurácia de Val. (%)
1	2.0570	42.36	1.3522	63.09
2	1.1839	67.34	1.0483	70.38
3	1.0061	71.59	1.0067	71.51
4	0.9290	73.43	1.0388	71.37
5	0.8678	74.90	1.0211	71.75
6	0.7977	76.80	1.0590	71.28
7	0.6959	79.71	1.0890	70.93
8	0.5772	83.12	1.2031	69.93
9	0.4509	86.58	1.3363	68.46
10	0.3295	90.41	1.4391	67.77

Tabela 33 – Resultados do Treinamento BERT Usando a Política OneCycle com LR Máxima de 2e-05 com limpeza de mensagem, *dropout* 20% e l2(0.01)

Classe	Precisão	Revocação	Pont. F1	Suporte
0	0.80	0.76	0.78	414
1	0.83	0.74	0.78	337
2	0.70	0.66	0.68	269
3	0.86	0.76	0.81	174
4	0.57	0.58	0.57	636
5	0.80	0.72	0.76	249
6	0.76	0.60	0.67	109
7	0.78	0.80	0.79	1461
8	0.73	0.71	0.72	296
9	0.76	0.64	0.69	124
10	0.60	0.57	0.59	509
11	0.65	0.65	0.65	453
12	0.53	0.63	0.57	742
13	0.48	0.51	0.49	320
14	0.67	0.69	0.68	103
15	0.84	0.62	0.71	108
16	0.61	0.55	0.58	528
17	0.61	0.54	0.58	230
18	0.68	0.71	0.69	2457
19	0.82	0.67	0.74	121
<b>Acurácia</b>	0.68			
<b>Macro Média</b>	0.70	0.65	0.68	9640
<b>Média Ponderada</b>	0.68	0.68	0.68	9640

Tabela 34 – Ajuste Fino sem Texto da Classificação: Desempenho por Classe: Precisão, Recall, F1-Score e Suporte

LR 2e-5, 10 épocas (1) Dados semi-balanc, 2.986 chamados					LR 2e-5, 10 épocas (2) Dados não balanc., 4.630 chamados					LR 2e-5, 10 épocas (3) Não balanc. Sintéticos, 31.123 chamados					LR 2e-5, 10 épocas, drpout+L2 sem txt (4) Não balanc., Sintét., 31.123 chamados				
Classe	Precisão	Revocação	Pont. F1	Suporte	Classe	Precisão	Revocação	Pont. F1	Suporte	Classe	Precisão	Revocação	Pont. F1	Suporte	Classe	Precisão	Revocação	Pont. F1	Suporte
0	0.04	0.02	0.03	45	0	0.28	0.33	0.30	378	0	0.83	0.68	0.75	111	0	0.80	0.76	0.78	414
1	0.12	0.09	0.10	82	1	0.43	0.43	0.43	598	1	0.83	0.86	0.84	385	1	0.83	0.74	0.78	337
2	0.10	0.14	0.12	113	2	0.10	0.08	0.09	139	2	0.83	0.76	0.79	342	2	0.70	0.66	0.68	269
3	0.07	0.06	0.06	68	3	0.03	0.03	0.03	78	3	0.76	0.77	0.76	464	3	0.86	0.76	0.81	174
4	0.12	0.16	0.14	126	4	0.15	0.13	0.14	196	4	0.61	0.63	0.62	473	4	0.57	0.58	0.57	636
5	0.18	0.15	0.17	170						5	0.67	0.65	0.66	773	5	0.80	0.72	0.76	249
6	0.10	0.18	0.13	84						6	0.61	0.59	0.60	309	6	0.76	0.60	0.67	109
7	0.00	0.00	0.00	34						7	0.81	0.78	0.80	93	7	0.78	0.80	0.79	1461
8	0.12	0.11	0.12	133						8	0.92	0.78	0.84	129	8	0.73	0.71	0.72	296
9	0.11	0.02	0.04	41						9	0.61	0.66	0.63	642	9	0.76	0.64	0.69	124
										10	0.77	0.67	0.72	260	10	0.60	0.57	0.59	509
										11	0.87	0.78	0.82	171	11	0.65	0.65	0.65	453
										12	0.82	0.77	0.79	269	12	0.53	0.63	0.57	742
										13	0.83	0.67	0.74	106	13	0.48	0.51	0.49	320
										14	0.71	0.74	0.73	2459	14	0.67	0.69	0.68	103
										15	0.55	0.66	0.60	525	15	0.84	0.62	0.71	108
										16	0.86	0.65	0.74	239	16	0.61	0.55	0.58	528
										17	0.76	0.73	0.75	116	17	0.61	0.54	0.58	230
										18	0.82	0.82	0.82	1488	18	0.68	0.71	0.69	2457
										19	0.72	0.69	0.71	283	19	0.82	0.67	0.74	121

Tabela 35 – Comparação dos Resultados para Diferentes Conjuntos de Dados e Configurações