

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS**

Fernando Teures Redondo

**Desenvolvimento de um sistema de priorização
assistido por Inteligência Artificial**

Florianópolis
2018

Fernando Teures Redondo

**Desenvolvimento de um sistema de priorização
assistido por Inteligência Artificial**

Relatório submetido à Universidade Federal de Santa Catarina como requisito para a aprovação na disciplina **DAS 5511: Projeto de Fim de Curso** do curso de Graduação em Engenharia de Controle e Automação.

Orientador: Prof. José Ricardo Rabelo

Florianópolis
2018

Fernando Teures Redondo

Desenvolvimento de um sistema de priorização assistido por Inteligência Artificial

Esta monografia foi julgada no contexto da disciplina DAS5511: Projeto de Fim de Curso e aprovada na sua forma final pelo Curso de Engenharia de Controle e Automação.

Florianópolis, 14 de Dezembro de 2018

Banca Examinadora:

Tiago Emílio da Silva
Orientador na Empresa
Resultados Digitais

Prof. Ricardo José Rabelo
Orientador no Curso
Universidade Federal de Santa Catarina

Prof. Joni da Silva Fraga
Avaliador
Universidade Federal de Santa Catarina

Luis Felipe Pelison
Debatedor
Universidade Federal de Santa Catarina

Guilherme Hammes
Debatedor
Universidade Federal de Santa Catarina

AGRADECIMENTOS

Primeiramente, gostaria de agradecer à minha família por toda educação e instrução que me foram dadas. Sem isso, eu não teria me tornado o profissional, estudante e cidadão que hoje me orgulho de ser.

Um agradecimento mais que especial a Leticia Carvalho Colantuono, minha namorada durante todo o período de faculdade, que me apoiou em todas as decisões e celebrou todas as minhas conquistas. Este documento mesmo só foi possível graças às revisões feitas por ela. Serei eternamente grato por tê-la encontrado.

Em seguida, gostaria de agradecer a todos os meus colegas de curso, com quem pude compartilhar aprendizados em disciplinas e grupos de extensão, em especial os que trabalharam comigo no Centro Acadêmico de Engenharia de Controle e Automação. O aprendizado que tive ao lado dessas pessoas é imensurável.

Ainda referente a meus colegas, gostaria de agradecer todo o apoio de meus amigos próximos, Fernando Barbosa Mota, Fabrício da Silva Hoepers, Alice Ferreira Branco, Fernanda Kasper Ortolan, Pedro Casali, Matheus Valin, Pedro Neves, Victor Hugo Schwab, Matheus Bruhns Bastos, Thiago Bertoldi, Marco Aurélio Geremias, Eduardo Campos Kneipp, Lucas Eduardo Cândido de Oliveira, Gabriel Martins, Stephanie Andreon, Renan Emmanuel Ramos Daquina, Paulo Curado, Leandro Henrique da Silva, Vinicio Virissimo, Vincenzo Bennetti e Gislaine Hoffman. Todos são pessoas muito queridas que marcaram minha passagem durante o período de faculdade.

Agradeço a toda a equipe da Resultados Digitais por todo o crescimento proporcionado, principalmente às minhas lideranças, que me instruíram e mostraram o significado de “A Team”, presente no Culture Code da empresa. Meus agradecimentos são às lideranças Vinicius Martini, Ana Paula Ferreira, Tiago Emílio

da Silva e Sidney Davanso. Ademais, agradeço muito a todos os membros das equipes Customer Heroes e Megazops pelo ensinamento de preocupar-se com o cliente sempre em primeiro lugar. Cabe um agradecimento especial, também, aos Champions de Zendesk das áreas que trabalham junto a mim na governança do nosso Helpdesk de atendimento e aos agentes Vinicius Martini, Thiago Torresani e Vitor do Vale Pereira, que se disponibilizaram a realizar o processo de priorização manual durante o projeto.

Também agradeço ao meu dojo de Aikido, Kawai Shihan Dojo, por me mostrar o caminho do Aikido, arte que renova constantemente minhas energias e me permite perseverar independentemente das circunstâncias. Por esse motivo, sou muito grato aos meus colegas de treino e, em especial, ao meu mestre, Carlos Alberto Grisalt.

Finalmente, um grande agradecimento ao meu orientador Ricardo José Rabelo, pela instrução e pelos questionamentos que me permitiram nortear o projeto. As decisões tomadas e o sucesso da solução só se concretizaram graças a todo esse apoio.

RESUMO

O projeto foi concebido com o intuito de priorizar os chamados de atendimento recebidos pela empresa Resultados Digitais de forma automática e escalável. A necessidade de priorização se deu por um contexto da empresa em que se era necessário entregar mais qualidade ao seu atendimento. Por meio de um estudo anterior ao projeto, percebeu-se que o processo atual de atendimento não entregava agilidade suficiente quando os clientes sofriam com problemas graves. A solução escolhida foi o uso de uma Inteligência Artificial para compreender o impacto do problema no cliente e, uma vez compreendido esse impacto, calcular a prioridade que o chamado deve ter por meio de uma Matriz de Prioridade. Dessa forma, todo o processo de priorização acontece de forma ágil, padronizada e sem nenhuma intervenção humana.

Para viabilizar as escolhas feitas em torno da Matriz de Prioridade e da Inteligência Artificial, foi desenvolvido um sistema distribuído em 3 aplicações web: no Zendesk, são tratadas e armazenadas todas as informações do atendimento; no Zapier, é realizado todo o cálculo da Matriz de Prioridade; e, no Natural Language Classifier da Watson, é feita a interpretação textual utilizando Inteligência Artificial. Em cada módulo foram configuradas regras próprias para que o processo de priorização acontecesse corretamente. O planejamento dessa solução foi elaborado seguindo a arquitetura Model View Controller, garantindo que o sistema se mantivesse coeso frente a sua natureza distribuída. A estrutura é inteiramente conectada via API e todas as trocas de mensagem passam pelo Zapier, para que se tenha registro dessas informações e, também, para garantir que não haja falha na requisição.

Palavras-chave: Inteligência Artificial. Automação de Processos. Priorização. Atendimento ao Cliente.

ABSTRACT

This project was developed with the main purpose of being able to automatically prioritize incoming support tickets in Resultados Digitais. This need appeared in a context where one of the objectives of the company was to deliver a better support experience to the customer. In a study made before the project, it got clear that the current support process wasn't fast enough to clients that were facing a severe issue. The approach used to solve this problem was to apply an Artificial Intelligence to understand the impact of the issue in the customer, after that the impact is given to a Priority Matrix that calculates the issue priority. In this way, the whole prioritization process happens in an agile and standardized way, not needing any human intervention.

To make the choices towards the Prioritization Matrix and Artificial Intelligence feasible, it was developed a system distributed in 3 different web applications: all the information related with the support ticket is handled and stored in Zendesk; the logic behind the Prioritization Matrix is processed in Zapier; and, finally, the text analyses using AI algorithms is made in Watson Natural Language Classifier. Specific rules were created in each module with the intent to guarantee that the whole process execute without error. The solution was planned based in Model View Controller architecture, allowing the system to sustain a cohesive aspect even due its distributed nature. The infrastructure is entirely connected via API and all the messages are intermediated by Zapier, by doing so, logs messages are generated, and the requests are assured to not fail.

Key-words: Artificial Intelligence. Process Automation. Prioritization. Customer Support.

LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo de priorização por categoria	25
Figura 2 - Exemplo de scorecard.....	26
Figura 3 - Exemplo de Matriz de Prioridade.....	27
Figura 4 - Etapas da metodologia em cascata.....	33
Figura 5 - Etapas do framework Scrum	33
Figura 6 - Framework Scrum	35
Figura 7 - Etapas do Projeto	38
Figura 8 - Kanban do Projeto.....	39
Figura 9 - Modelo em BPMN do processo manual	50
Figura 10 - Modelo em BPMN	51
Figura 11 - Diagrama de sequência do processo manual.....	52
Figura 12 - Diagrama de sequência do processo automático.....	53
Figura 13 - Diagrama de implementação do sistema	54
Figura 14 - Exemplo de campo de Impacto	55
Figura 15 - Exemplo de campo de Urgência.....	56
Figura 16 - Regras da visualização para exportação de tickets.....	57
Figura 17 - Configuração de alvo no Zendesk.....	59
Figura 18 - Exemplo de gatilho no Zendesk	60
Figura 19 - Zap do processo manual	62
Figura 20 - Zap do processo automático	63
Figura 21 - Formulário de abertura de chamado na Central de Ajuda.....	65
Figura 22 - Formulário de priorização	66
Figura 23 - Percentual de chamados priorizados antes e depois	70

LISTA DE ABREVIATURAS E SIGLAS

NLP – Natural Language Processing
NLC – Natural Language Classifier
CSAT – Customer Satisfaction
SLA – Service Level Agreement
IA – Inteligência Artificial
AI – Artificial Intelligence
UX – User Experience
FIFO - First In First Out
ITIL – Information Technology Infrastructure Library
MVC – Model View Controller
API - Application Programming Interface
SaaS – Software as a Service
FRT – First Reply Time
NRT – Next Reply Time
RWT – Requester Wait Time
ROI – Return On Investment
RD – Resultados Digitais

SUMÁRIO

1	Introdução.....	15
1.1	Contexto da Resultados Digitais	16
1.2	Contexto na área de Suporte	17
1.3	Justificativa.....	19
1.3.1	Objetivo geral	19
1.3.2	Objetivos específicos	19
2	Fundamentação Teórica	21
2.1	Conceitos básicos de Marketing Digital	21
2.2	Processos de Suporte.....	22
2.2.1	Gestão de fila de atendimento	22
2.2.2	Formas de priorização.....	23
2.2.3	Métodos de priorização	24
2.3	Inteligência Artificial aplicada a textos.....	28
2.3.1	Deep Learning.....	28
2.3.2	Natural Language Processing (NLP).....	29
2.4	Arquitetura de Software	30
2.4.1	Model View Controller (MVC).....	30
2.4.2	APIs e Sistemas Distribuídos	31
2.4.3	Framework Scrum	32
3	Desenho da Solução	37
3.1	Metodologia.....	38
3.2	Matriz de Prioridade	39
3.2.1	Urgência.....	40
3.2.2	Impacto	40
3.2.3	Construção da Matriz	41

3.3	Processo de Priorização.....	42
3.4	Estrutura conceitual do Sistema.....	43
3.4.1	Zendesk.....	44
3.4.2	Zapier.....	45
3.4.3	Natural Language Classifier.....	46
4	Desenvolvimento e Implementação.....	49
4.1	Modelos e diagramas do sistema.....	49
4.1.1	Modelo do processo em BPMN.....	49
4.1.2	Diagrama de Sequência em UML.....	51
4.1.3	Diagrama de Implementação em UML.....	53
4.2	Model (Nível de persistência dos dados).....	54
4.2.1	Dados de “Impacto” e “Urgência” do ticket.....	55
4.2.2	Dados de Treinamento do NLC.....	56
4.3	Controller (controle do fluxo de informação).....	58
4.3.1	Manipulação dos tickets (Zendesk).....	58
4.3.2	Matriz de Prioridade (Zapier).....	61
4.4	View (interface para o usuário).....	64
4.4.1	Formulário de Abertura de Chamados.....	64
4.4.2	Formulário de Priorização.....	66
5	Análise e Resultados.....	69
5.1	Impacto na operação.....	69
5.2	Análise entre processo manual e automático.....	72
5.3	Cálculo de ROI.....	73
6	Considerações Finais e Perspectivas.....	75
6.1	Aprendizados.....	75
6.2	Limitações.....	76

6.3 Planos futuros	76
REFERÊNCIAS	78

1 INTRODUÇÃO

A área de Suporte, hoje, é vista como um recurso estratégico para as empresas de tecnologia. Por muito tempo, esse setor fora negligenciado devido a seu caráter de atendimento que, sob o ponto de vista dos desenvolvedores, seria desnecessário caso a ferramenta fosse “impecável”. Entretanto, nos tempos atuais, é preciso que as empresas cresçam rapidamente e que seus produtos mudem constantemente. Dessa forma, a função do Suporte é instruir e reduzir a fricção que os clientes podem sentir no decorrer dessas mudanças, garantindo que estes estejam acolhidos pela empresa e continuem obtendo sucesso no uso do software.

Tal situação é vista com extrema seriedade em empresas que seguem o modelo Software as a Service (SaaS). É de grande importância que os assinantes mantenham o contrato do produto e não o cancelem, pois isso garante receita recorrente mensalmente para a companhia, gerando um balanço financeiro saudável com perspectivas positivas para o futuro. Nesse sentido, o Suporte é uma peça chave de retenção de clientes e posicionamento da marca, pois é o canal de contato de uso mais frequente.

Como todo o atendimento acontece dentro das ferramentas do tipo Helpdesk, a riqueza de dados comportamentais da base de clientes é muito grande. Isso empodera as análises de dados, que trazem em detalhes a forma como o usuário utiliza a ferramenta e quais são os maiores desafios da empresa para que ele permaneça usando o produto.

Outro recurso importante das ferramentas Helpdesk é viabilizar uma série de otimizações de processo de atendimento, permitindo controlar, na operação, o tempo que se leva para atender a demanda dos clientes e diferenciar os tipos de chamados. Ainda, é possível configurar a prioridade de cada chamado, para que eles tenham tempos de atendimento mais curtos.

1.1 Contexto da Resultados Digitais

A Resultados Digitais é uma empresa que cresce em ritmo acelerado, tanto no mercado nacional, quanto internacional. Presente em mais de 20 países, ela se sustenta em um modelo de negócio SaaS, que é, fundamentalmente, um modelo de receita recorrente, no qual os clientes pagam pelo uso do software ofertado em períodos definidos, usualmente por mês ou por ano. Em função desse modelo, a empresa conta com apenas dois produtos: o RD Station Marketing, ferramenta para planejamento, implementação e análise em Marketing Digital, e o RD Station CRM, ferramenta que ajuda a executar e organizar fluxos de vendas. O produto mais consolidado da empresa é o primeiro, enquanto o segundo foi lançado em 2018 e ainda se encontra em constante mudança.

Devido ao foco em apenas dois produtos, o funcionamento da empresa é voltado ao desenvolvimento e excelência de ambos. Isso permite que melhorias constantes sejam lançadas, garantindo agilidade de desenvolvimento superior às concorrentes que contam com portfólio muito vasto. Outro aspecto relevante trata-se da interconectividade dessas ferramentas, já que marketing e vendas são áreas intrinsecamente ligadas: enquanto o trabalho do marketing, de maneira geral, é atrair interessados, o trabalho de vendas é transformar esses interessados em clientes. Assim, o fluxo operacional de geração de clientes e, conseqüentemente, de geração de receita para a empresa, pode ser acompanhado do começo ao fim quando se opta por contratar e conectar as duas plataformas ofertadas.

Com mais de 12.000 clientes, a Resultados Digitais enfrenta complexos desafios para a manutenção de uma operação saudável frente a tamanho crescimento. Além de seus produtos, a Resultados Digitais é responsável por organizar o maior evento de Marketing e Vendas, inspirado na magnitude desse evento um dos três principais objetivos da empresa para 2018 era o “*RD Summit Everyday*”, que representa a intenção de aumentar substancialmente a qualidade da experiência que os clientes têm com a empresa como um todo, seja usando o produto ou interagindo com qualquer canal de comunicação. Para a companhia, esse objetivo se tornou prioritário a partir da decisão de expandir seu portfólio de

produto e crescer sua presença internacional, já que sua preocupação era garantir que, mesmo com tantas mudanças, seus clientes continuassem se sentindo apoiados sempre que precisassem, de modo a diminuir sua insegurança e evitar que cancelem os contratos.

1.2 Contexto na área de Suporte

Na Resultados Digitais, o setor é visto com a importância que merece, tanto na qualidade do atendimento como no investimento em tecnologias de ponta para apoiar a operação. A área de Suporte é denominada Customer Heroes e é reconhecida por minimizar as fricções causadas por mudança do produto. O time acredita que o sucesso do cliente é a principal prioridade e investe em diversas formas de autoatendimento para que seja possível entregar cada vez mais valor com uma operação de custo cada vez menor.

Com o crescimento da operação do Customer Heroes, se tornou clara a necessidade de criar uma equipe focada na escalabilidade do atendimento e na redução dos custos da operação. Foi assim que, em 2017, nasceu o time de inteligência operacional do Suporte, o Megazops. Originalmente, sua composição era de apenas quatro membros: dois analistas de inteligência operacional, um analista de desenvolvimento e implementação de chatbot e, finalmente, uma coordenadora. Hoje muito maior, o time executa a maior parte dos projetos relacionados a escalabilidade e eficiência operacional do Suporte. O presente projeto, inclusive, foi desenvolvido dentro dele.

Visando ao objetivo de *“RD Summit Everyday”*, diversas mudanças aconteceram na área de Suporte para tornar o processo de atendimento mais completo; dentre elas, destacam-se: o ajuste nos tempos de SLA, a criação de segundo e terceiro níveis de atendimento e o novo processo de CSAT e Close the Loop. A seguir, serão detalhados os projetos de maneira breve, para que se possa ter um melhor entendimento das mudanças.

O ajuste do Service Level Agreement (SLA) foi feito com base em um estudo completo dos tempos médios de atendimento. Sob a perspectiva do Suporte, o SLA

reflete o acordo firmado com o cliente quanto ao tempo necessário para respondê-lo. Com isso, foram redefinidas as metas para o First Reply Time (tempo que a RD leva até a primeira resposta do chamado), Next Reply Time (tempo que a RD leva para uma próxima resposta do chamado) e Requester Wait Time (soma do first reply time com todos os next reply times que o chamado teve). Os novos níveis de atendimento foram criados com o intuito de se obter profundidade na investigação dos chamados, pois eventualmente surgiam demandas técnicas que deviam ser investigadas pelo time de desenvolvimento, o que não mantinha padronizada a experiência do cliente com o Suporte. Por fim, o novo CSAT contemplou formas mais atrativas para o cliente avaliar os chamados. O objetivo em torno de Customer Satisfaction (CSAT) é sempre trazer dados o mais realistas possível sobre o sentimento do cliente em relação à empresa. Para esse projeto de CSAT, a estratégia empregada foi a utilização de formulários de avaliação mais simples e, ao mesmo tempo, mais completos, além da criação de um processo de fechamento do ciclo de comunicação com o cliente, em que um agente específico da operação liga para todos que avaliaram os chamados negativamente.

Ainda assim, um estudo feito pelo setor de qualidade mostrou que alguns casos de insatisfação estavam relacionados ao fato de não se dar a devida prioridade aos atendimentos nos quais os clientes enfrentavam problemas críticos. Além disso, os chamados eram priorizados com critérios subjetivos, pois cada agente tinha um julgamento diferente sobre o que deveria ser priorizado. Finalmente, por não haver padrão quanto a como priorizar os chamados, não se podia definir metas diferentes de SLA para cada valor de prioridade (normal, alta e urgente), o que fazia com que uma estrutura de fila única de atendimento não pudesse ser criada e operacionalizada para os agentes.

Na prática, durante a operação, a prioridade dos chamados era alterada quando o agente julgava necessário. Porém, essa priorização não refletia em uma mudança do tempo máximo para responder o chamado, apenas registrava-se a informação para que os demais agentes soubessem que se tratava de uma demanda possivelmente urgente. Isso gerava um processo dúbio no contexto de atendimento: se um chamado urgente tivesse folga no tempo para ser atendido,

enquanto um de prioridade normal estivesse com pouco tempo, a decisão de qual atender primeiro não ficava clara para o agente. Essa confusão fazia com que, por vezes, alguns chamados priorizados não fossem respondidos em tempo adequado e, por outras, chamados de prioridade normal tivessem uma resposta muito demorada por se responder chamados priorizados que poderiam esperar.

Esse contexto mostrou que, claramente, fazia-se necessário criar um padrão do que deve ser priorizado, bem como homogeneizar ao máximo possível a tomada de decisão no instante da classificação dos chamados. O estudo de priorização dos chamados partiu dessa premissa, visto que um sistema com regras bem definidas garante o padrão desejado, enquanto a homogeneização da priorização é viável a partir do momento em que o texto é interpretado por uma máquina, seja a interpretação feita por Inteligência Artificial ou por qualquer outra técnica cabível.

1.3 Justificativa

De acordo com o contexto apresentado anteriormente, verificou-se que existia uma necessidade latente para que os chamados de atendimento fossem priorizados de forma consistente, padronizada e ágil. Em torno dessa necessidade, então, o projeto de Matriz de Prioridade Assistida por IA foi desenvolvido, tendo sido elaborado para priorizar os chamados de atendimento apenas do RD Station Marketing, por se tratar do produto mais consolidado da empresa.

1.3.1 Objetivo geral

Garantir que os clientes do RD Station Marketing que sofrem de um problema crítico tenham uma resposta rápida e resolutiva. Essa garantia reflete em se ter clareza, na área de Suporte, sobre o que deve ser priorizado e em quanto tempo o chamado deve ser respondido. Além disso, a priorização não deve sobrecarregar os agentes da operação de atendimento e nem diminuir sua produtividade.

1.3.2 Objetivos específicos

- Definir um padrão para a priorização dos chamados de Suporte
- Garantir que todos os chamados técnicos da ferramenta RD Station Marketing estejam sujeitos a esse padrão
- Minimizar o tempo que os agentes consomem executando o processo de priorização
- Evitar causar grandes mudanças no fluxo de trabalho dos agentes

2 FUNDAMENTAÇÃO TEÓRICA

A seguir, serão tratados conceitos importantes para o desenvolvimento da solução. O intuito deste capítulo é consolidar a base teórica existente por trás do que foi desenvolvido durante o projeto.

2.1 Conceitos básicos de Marketing Digital

Como definido anteriormente, o projeto de priorização foi pensado para o suporte ao produto RD Station Marketing. Para adequada compreensão do contexto de atendimento da Resultados Digitais, é necessário entender aspectos relevantes de marketing digital.

É possível analisar o marketing digital, de maneira geral, com base no conceito de funil de vendas. O funil define que, inicialmente, um visitante acessa uma determinada página da web e, ao se interessar por algum conteúdo, troca seus dados pelo material ofertado. Nesse momento, ele passa a ser considerado um lead, que nada mais é que um contato com diversas informações pessoais. Tendo os leads, o objetivo do marketing é fazer com que eles continuem no funil até se tornarem uma oportunidade e, em sequência, uma venda. A ação de fazer os leads progredirem no funil é conhecida como nutrição de leads e, basicamente, consiste em enviar conteúdo de interesse para cada lead e fomentar seu interesse e acesso às páginas do site. Quando o nível de interesse do lead atinge um determinado estágio, ele passa a ser considerado uma oportunidade, momento em que ele é passado para o setor de vendas, que formulará uma oferta adequada ao seu perfil [14].

Para tornar essas operações viáveis, usualmente são utilizadas ferramentas computacionais que permitem registrar os leads e interagir com eles. Algumas opções mais completas também permitem gerir outros aspectos do marketing digital, como posição em ferramentas de busca (Google e Yahoo por exemplo) ou análise de visitantes no site. O RD Station Marketing é uma solução completa de Marketing Digital cujo principal objetivo é atender a todas as necessidades associadas ao funil

de vendas. Portanto, esse é um papel primordial que a ferramenta deve cumprir durante seu funcionamento.

2.2 Processos de Suporte

A questão de priorização de chamados se refere fundamentalmente a uma boa gestão da fila de atendimento. No estudo, ficou evidente que medidas muito mais simples podem ser suficientes, desde que a quantidade de pedidos de Suporte recebidos pela empresa seja pequena. Inicialmente, é importante introduzir o conceito de ticket; um ticket é simplesmente um registro de todas as conversas de qualquer canal: ele agrupa tudo em um só lugar e mantém sua equipe informada [1].

Outro aspecto importante do Suporte é a execução do seu fluxo de trabalho. É primordial que o trabalho seja organizado de maneira clara e bem definida, pois é necessário ser ágil no atendimento e, ao máximo possível, se deve minimizar o tempo que se gasta compreendendo os processos e ferramentas de apoio. A execução do atendimento é feita por uma pessoa que exerce a função de agente de atendimento, este, de maneira geral, têm como função compreender o problema do cliente para, em seguida, resolvê-lo da melhor forma.

Mesmo se tratando da prioridade dos tickets, existem diversas formas de priorização e é importante escolher a que mais se adeque à situação atual da empresa. Entretanto, deve-se tomar cuidado com as priorizações: se não forem definidas metas coerentes para os tempos de atendimento, chamados pouco priorizados podem esperar em excesso.

2.2.1 Gestão de fila de atendimento

Segundo o artigo da Help Scout, toda empresa eventualmente se depara com situações em que o volume de atendimento cresce a ponto de ter que surgir um novo processo [2]. Tais momentos demandam mudanças na forma com que esses chamados são ordenados e distribuídos, pois, caso isso não ocorra, a tendência é que o atendimento ao cliente fique muito demorado.

O artigo apresenta diversas opções para uma boa gestão de fila:

- Ordenar os tickets pelo momento de chegada, onde o primeiro que entra será o primeiro a ser atendido (fila FIFO).
- Especialização de agentes por variedade de assuntos ou profundidade técnica
- Atendimento prioritário baseado no tipo de cliente e/ou tipo de demanda solicitada
- Trabalhar com filas sem responsável pelo chamado; nesses casos, um agente complementa o atendimento do outro
- Diferentes formas de dar o first reply e next reply; a sugestão é de que agentes específicos monitorem todos os chamados e aqueles mais simples recebem uma resposta, mesmo que não sejam o primeiro da fila.

Alguns pontos que o artigo aborda são muito relevantes para o atendimento. Tratando-se especialmente de gerir bem uma fila, é apresentado que a melhor forma de o fazer é empregar a combinação de diversas técnicas, evitando prender-se a somente uma.

2.2.2 Formas de priorização

Com relação especificamente ao tópico de priorização, o artigo [3] apresenta opções variadas para selecionar o que deve ser priorizado ou não. Por mais que, em boa parte, as opções abordadas no artigo sejam mais simples que a Matriz de Priorização, é enfatizado que soluções mais arrojadas potencializam a precisão da classificação do que é prioritário e do que não é. O artigo também destaca a opção de classificar os chamados via IA, retratando que os ganhos dessa abordagem são muito significativos.

Ainda que a priorização seja uma ótima forma de atender com mais agilidade os problemas mais sérios, dois artigos levantam precauções que se deve ter na forma de priorizar os chamados (Como priorizar chamados [4] e Cuidados com a priorização de acordo com a norma ITIL [5]). O padrão ITIL trata-se de uma

convenção internacional de boas práticas no mundo de TI e, conseqüentemente, engloba diversos aspectos da operacionalização de processos de Suporte.

O primeiro ponto levantado é que uma priorização manual normalmente está sujeita a falhas e ineficiência, devendo, portanto, ser evitada. O artigo [4] destaca como sendo os problemas mais comuns:

- Desperdício do tempo do agente, pois ele poderia estar fazendo atividades de maior valor agregado.
- Sujeito a erros. Pessoas não são eficientes todos os dias e podem tomar decisões enviesadas pela fadiga ou momento de pressão
- Pode não ser uma escolha baseada em dados, já que pode envolver diversos aspectos diferentes do ticket, tornando a análise muito longa.

Além disso, o segundo artigo [5] apontou que, mesmo seguindo os padrões ITIL, a categorização pode levar a situações em que somente os chamados de prioridade alta são atendidos em um tempo razoável, estendendo muito o tempo de chamado dos tickets pouco priorizados. Para evitar que isso aconteça, é de fundamental importância que os SLAs de tempo de atendimento sejam definidos de acordo com o volume esperado de cada uma das prioridades.

2.2.3 Métodos de priorização

Priorização simples por categoria

O artigo [6] defende que o primeiro passo para a priorização deve ser algo simples, como divisões em macro categorias: dependendo do escopo que o ticket se encaixar, ele recebe um tipo de prioridade. Um exemplo pode ser visto na imagem a seguir:

Figura 1 - Exemplo de priorização por categoria

#	Pergunta	Resposta	Prioridade
A	O problema/solicitação/dúvida impede que a empresa realize seus processos básicos (faturamento / pagamento)?	Sim	Crítica
B	O problema/solicitação/dúvida impede a execução de algum serviço da empresa?	Sim	Crítica
C	O problema/solicitação/dúvida impede que o usuário realize sua função?	Sim	Crítica
D	O problema/solicitação/dúvida do usuário não o impede de realizar as atividades, mas causa retardo na execução das tarefas?	Sim	Alta
E	O problema/solicitação/dúvida do usuário não o impede de realizar as suas atividades, nem causa retardo, mas o ajudaria a fazer o trabalho melhor?	Sim	Média
F	O problema/solicitação/dúvida impede que o usuário realize funções secundárias?	Sim	Média
G	O problema/solicitação/dúvida não se encaixou em nenhuma das categorias? (!) Aqui é preciso ter cuidado com os falsos positivos. Como nossa estrutura de atribuição de prioridade é simples, pode ser que algum item que caia aqui não seja de prioridade baixa. Fique atento!	Sim	Baixa

Fonte: Referência 6

Por mais que a solução apresentada seja simples e ágil, ainda está sujeita a muito viés por parte do agente que categoriza o chamado. Isso acaba permitindo que, eventualmente, uma categoria muito genérica acabe incorporando tickets que seriam de outras categorias.

Sistema de scorecard

Outra abordagem comum é o sistema de scorecard. Esse sistema levanta quais são os atributos comuns para a maior parte dos tickets que a empresa costuma receber, definindo, a seguir, em quais atributos o chamado se encaixa. Em [7] foi apresentado uma abordagem com uso da técnica de 5W (Who, What, When, Where, Why) para definir os atributos do scorecard, obtendo, assim, os seguintes exemplos para dois cenários:

Figura 2 - Exemplo de scorecard

Ticket 1

Corporate e-mail is down. Ticket number 100		
Who is affected	Company	500 points
What is affected	E-mail	500 points
When is response needed	Immediate	500 points
Where is response needed	Corporate computer room	500 points
Why is response needed	Not working	500 points
Total score		2,500 points

Ticket 2

Remote user cannot access e-mail and will not need access until tomorrow. Ticket number 101		
Who is affected	User	100 points
What is affected	E-mail	500 points
When is response needed	Next day	200 points
Where is response needed	Home office	100 points
Why is response needed	Not working	500 points
Total score		1,400 points

Fonte: Referência 7

Nesse exemplo, como o primeiro ticket possui um score maior que o segundo, ele teria uma prioridade maior. Quanto ao valor da prioridade em si, podem ser definidas faixas de score que correspondam aos valores existentes. Segue o exemplo:

Score < 2000: prioridade normal

Score >= 2000: prioridade alta

Outra opção, caso a ferramenta de Helpdesk suporte, é fazer uma definição de prioridade diretamente no score, o que indicaria que um chamado com score 1401 é mais prioritário que 1400 e o de 1400 é mais prioritário que o de 1399. Entretanto, na prática, é difícil ser tão preciso na categorização de um ticket.

Matriz de prioridade

Finalmente, a última abordagem que aparece nos estudos é a de matriz de prioridade. Essa técnica permite a categorização do chamado cruzando-se informações tanto do cliente quanto do agente, de modo a atenuar viés de ambos os lados. Usualmente, no contexto de atendimento, o cliente é quem fica responsável por definir a urgência do chamado e, o agente, por definir o impacto do chamado.

De maneira sucinta, o documento da Unipampa [18] descreve como construir essa matriz. Um passo importante que deve ser tomado previamente é a definição de como o cliente informará a urgência do chamado e como o agente classificará seu impacto. Uma opção, nesse contexto, é inserir perguntas intermediárias que ajudem a medir suas magnitudes.

Tendo um valor (no caso: muito baixo, baixo, médio, alto e muito alto) para a urgência e para o impacto, constrói-se a seguinte matriz de prioridade, na qual 1 é a prioridade mais alta e 5 é a prioridade mais baixa. Em seguida, essa prioridade deve ser atribuída ao ticket dentro do Helpdesk.

Figura 3 - Exemplo de Matriz de Prioridade

Impacto (Helpdesk)	Muitoalto	4	3	2	1	1
	Alto	4	3	2	2	1
	Médio	4	4	3	3	2
	Baixo	5	4	4	4	3
	Muito baixo	5	5	4	5	4
		Muito baixo	Baixo	Médio	Alto	Muito Alto
		Urgência (Usuário)				

Fonte: Referência 18

2.3 Inteligência Artificial aplicada a textos

Inteligência Artificial é uma área de estudo em grande ascensão na atualidade, com projetos financiados por grandes empresas, como Google, IBM e Microsoft. Dois aspectos muito importantes da Inteligência Artificial são os de generalização e evolução que a abordagem permite. Justamente, uma definição que pode ser dada a ela é: “Inteligência Artificial trata-se da automação do comportamento inteligente” [22].

Enquanto, de maneira geral, sistemas tradicionais desempenham satisfatoriamente nos exatos cenários para os quais foram projetados, sistemas baseados em IA têm potencial para atuar em contextos apenas semelhantes aos originais. Além disso, o constante treinamento dos algoritmos permite melhoria de suas performances ao longo do tempo sem necessidade de edição do código fonte. Devido a essas duas características, aplicações de IA têm sido cada vez mais utilizadas em soluções de engenharia nas quais não há alta padronização no que deve ser processado.

Atualmente, as principais aplicações de IA ocorrem na área da computação, mas muito tem sido desenvolvido também no campo de robótica móvel e veículos autônomos. A perspectiva é de que os setores de P&D e Serviços ao cliente sejam os mais impactados com o crescimento dessa tecnologia. As principais barreiras para seu uso são: infraestrutura de TI precária, falta de profissionais especializados no mercado [8], obsessão com o aspecto acadêmico que muitas vezes a abordagem contempla e, ainda, a negligência do problema que o projeto deve resolver.

2.3.1 Deep Learning

Algoritmos de Deep Learning são fundamentalmente baseados em Redes Neurais, como retratado em [9]. Redes Neurais são algoritmos baseados na estrutura cognitiva de um cérebro, em que existem diversas camadas de neurônios que processam parte da informação total. Deep Learning é uma extensão desse

conceito, permitindo que mais camadas sejam incluídas que na Rede Neural clássica.

O surgimento do Deep Learning permitiu grande avanço nas aplicações da Inteligência Artificial, pois o potencial de abstração e variedade de cenário que podem ser interpretados é significativamente maior nessa abordagem. Isso se deve às diversas camadas de neurônios que podem absorver diferentes aspectos da informação a ser processada. Hoje, os principais campos de aplicação do Deep Learning são para Reconhecimento de Fala, Reconhecimento de imagem, Sistema de Recomendação e Processamento de Linguagem Natural (NLP), e muito também vem sendo estudado para uso do algoritmo em Análise de Dados de maneira geral [10]. Ademais, um aspecto que vale ressaltar da técnica é que, mesmo que sua execução seja rápida, o treinamento do modelo exige bastante esforço computacional, de modo que soluções desse tipo não sejam ideais em casos que o escopo muda completamente com frequência.

Como o artigo resalta, o Deep Learning é um dos grandes responsáveis pelo momento de crescimento acelerado que a IA vive hoje. Graças a ele, diversas empresas estão usando Inteligência Artificial em seus processos de negócio, principalmente nas áreas de Suporte, em que muitos processos possuem escopo bem definido [23].

2.3.2 Natural Language Processing (NLP)

De maneira geral, computadores são bons em processar dados estruturados, como binário ou demais tipos numéricos; entretanto, dados de escrita são desestruturados por natureza, o que dificulta sua interpretação [11]. Ao trabalhar com dados estruturados, tem-se uma compreensão da associação entre os elementos e, assim, pode-se montar modelos que tenham comportamentos específicos sob determinadas circunstâncias.

Contudo, quando o dado é desestruturado, a relação deixa de ser lógica e, portanto, a informação muitas vezes não consegue ser interpretada. Para contornar essa deficiência, surgiram os algoritmos de Natural Language Processing, que a

partir de recursos de Deep Learning buscam interpretar diversos elementos presentes em um texto como contexto, entidades, sentimentos e intenções, de forma que o significado como um todo possa ser interpretado de maneira mais geral.

O algoritmo de NLP se baseia em extrair um sentido de pequenas partes do texto, como uma entidade [12]. Apoiado em uma estrutura de Deep Learning, é possível entender que diversas cadeias de palavras possuem um significado em específico. Ao alimentar o sistema com uma quantidade massiva de exemplos, é possível criar uma generalização em que se atribui o mesmo significado para cadeias de palavras apenas semelhantes. A partir da quebra de diversos aspectos de um texto, pode-se extrair uma compreensão mais completa sobre seu assunto, permitindo que, computacionalmente, textos grandes possam ser interpretados, diferenciados e categorizados [13].

O Natural Language Classifier (NLC) é uma extensão das técnicas de NLP da Watson (segmento de NLP da empresa IBM), que consiste em enquadrar textos em categorias definidas pelo usuário. Tais categorias são informadas em conjunto com uma série de exemplos para cada uma delas, e a categorização é feita de acordo com a comparação de cadeia de palavras conforme o algoritmo de NLP. Uma vez definida a base de dados, treina-se o NLC e, assim, ele ganha a capacidade de identificar qual o percentual de proximidade do texto a alguma das categorias. Essa função é primordial para que se possa categorizar textos através de categorias bem definidas, esse método por exemplo pode ser usado para interpretar se um determinado chamado se enquadra em alguma das categorias consideradas prioritárias em um contexto de atendimento.

2.4 Arquitetura de Software

2.4.1 Model View Controller (MVC)

O MVC é derivado do modelo três camadas e é atualmente uma das abordagens mais usadas para arquitetura de software. Sua principal vantagem é um ganho significativo na modularidade do sistema, o que permite fácil manutenção e

atualização do que foi desenvolvido [15]. Os sistemas baseados em MVC são separados basicamente em três módulos:

- Model – Basicamente, é a camada de persistência dos dados, na qual normalmente se situa o modelo do banco de dados e todas as informações geradas para a aplicação
- View – É o nível de interação direta com o usuário do sistema, muitas vezes conhecido também como interface
- Controller – Onde operam todas as regras da aplicação, é nesta camada que os dados vão ser interpretadas e em função do que foi configurado determinadas ações serão disparadas.

Além de garantir a modularidade, a abordagem em torno do MVC permite que diferentes tecnologias sejam usadas em cada um dos níveis, conferindo muito mais flexibilidade ao desenvolvimento do sistema e permitindo que a ferramenta mais adequada seja alocada na função em que melhor desempenha.

2.4.2 APIs e Sistemas Distribuídos

Com o advento da computação em cloud, diversos sistemas – conhecidos como aplicações web – passaram a ser hospedados na nuvem e acessados diretamente pelos navegadores. O crescimento dessas aplicações foi tamanho que empresas estão deixando de usar sistemas hospedados em seus servidores locais para utilizar sistemas totalmente hospedados em cloud. Para essas empresas, qualquer sistema pode ser acessado e qualquer dado pode estar disponível em um mesmo navegador, o que garante extrema flexibilidade para as áreas operacionais.

Além do acesso humano a esses sistemas, aplicações web permitem uma conexão entre si muito mais simples, pois são minimizadas as barreiras de se ter que lidar com sistemas legados. A abordagem mais usada para conectar aplicações web são as Application Programming Interfaces (APIs) [16]. Basicamente, API é uma interface de comunicação para um dado sistema, no caso de sistemas web ela é desenvolvida como um serviço periférico à aplicação principal que permite entregar e receber dados através de um protocolo HTTP e empacotamento da mensagem em

um formato JSON. Por ser praticamente a abordagem mais usada por esses sistemas, muitas vezes é simples conectar a saída de uma aplicação com a entrada de outra. Isso faz com que a empresa tenha uma rede de sistemas com dados padronizados de ponta a ponta e fácil fluxo de atividade entre áreas.

2.4.3 Framework Scrum

Scrum é um dos frameworks ágeis que derivaram do “agile manifesto” em 1990, o manifesto pregava quatro principais fundamentos a partir dos quais todos os frameworks que surgiram a seguir se inspiraram [19]:

- **Indivíduos e interações** mais que processos e ferramentas
- **Software em funcionamento** mais que documentação abrangente
- **Colaboração com o cliente** mais que negociação de contratos
- **Responder a mudanças** mais que seguir um plano

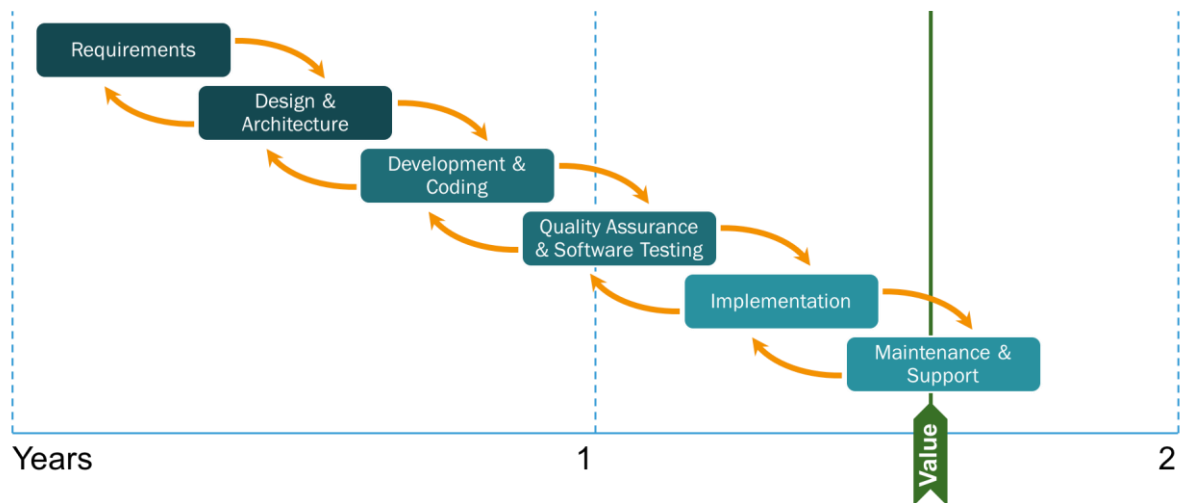
Sob esses quatro pilares, foram criados diversos frameworks, chamados de frameworks ágeis. É importante ressaltar que um framework é diferente de uma metodologia: enquanto metodologias preveem padrões que devem ser seguidos à risca para uma boa execução, frameworks contemplam apenas valores para os papéis dos participantes, permitindo que a execução se adapte à realidade que for necessária [20].

O grande diferencial dos frameworks ágeis é a rapidez na entrega de valor. Enquanto as metodologias tradicionais costumam entregar valor somente na última etapa do projeto, frameworks ágeis buscam encurtar essas entregas e diluí-las em várias pequenas entregas ao longo do desenvolvimento da solução. Isso permite que o cliente já tenha noção do que será entregue logo no início e facilita mudança em aspectos do projeto que não o agradem.

Seguindo esses paradigmas, o Scrum permite que o desenvolvimento de projetos ocorra de forma ágil e flexível. O artigo da DZone [21] explica de forma detalhada o uso do framework; no presente documento, serão detalhados apenas os

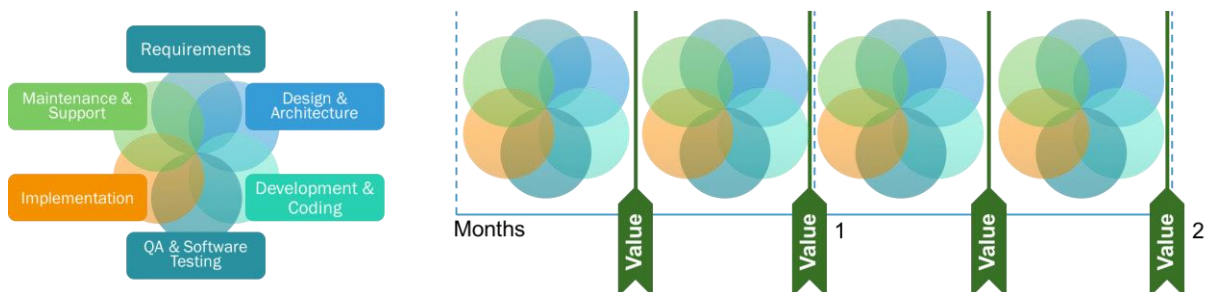
principais conceitos. Assim como boa parte dos frameworks ágeis, o Scrum é baseado no empirismo, ou seja, dentro do framework o problema é sempre encarado na perspectiva de seus efeitos sentidos e nas dores decorrentes de sua existência. Em função disso, o foco é sempre o ganho real da solução, ou seja, o quanto ela agrega valor para quem vai usá-la, e não tanto o valor agregado somente à evolução do projeto. Sob essa perspectiva, o Scrum difere bastante da metodologia em cascata, pois nesta o valor costuma ser somente a entrega final, enquanto naquele são resolvidas partes do problema a cada pequena entrega.

Figura 4 - Etapas da metodologia em cascata



Fonte: Referência 21

Figura 5 - Etapas do framework Scrum



Fonte: Referência 21

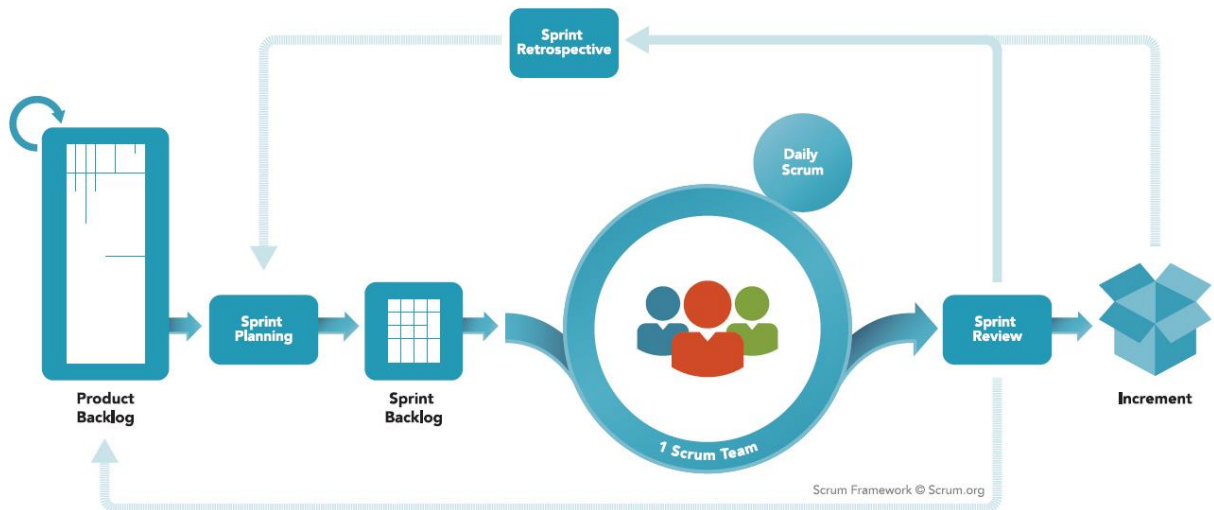
A execução do Scrum é inteiramente baseada em Sprints, que são cada uma das pequenas etapas de desenvolvimento com uma clara entrega de valor ao seu final. Sprints costumam ter periodicidade definida e também, para que sejam

consideradas concluídas, atividades que devem ser cumpridas. Seu planejamento tem o objetivo de que a solução final tenha contemplado todo o backlog de produto (consiste da demanda total do projeto). As Sprints derivadas desse backlog são normalmente definidas pelo Product Owner, membro da equipe responsável por definir e priorizar o que deve ser desenvolvido. Devido à flexibilidade do framework, essa definição pode acontecer também de diferentes formas, podendo o Product Owner decidir sozinho ou em reuniões de priorização com o restante da equipe.

Uma vez definido o que será desenvolvido, o Product Owner recorre ao Scrum Master para verificar a viabilidade de desenvolver o que se deseja dentro da Sprint prevista. É papel do Scrum Master ter plena noção da disponibilidade dos membros da equipe e do quanto irá demandar de recurso desenvolver o que o Product Owner deseja. Caso se tenha capacidade para desenvolver as demandas desejadas, a Sprint é quebrada em atividades (Sprint Backlog) que são alocadas em um Kanban para que a execução possa ser acompanhada. Cada membro da equipe se responsabiliza por suas atividades e, assim, se dá sequência à Sprint.

Figura 6 - Framework Scrum

SCRUM FRAMEWORK



Fonte: Referência 21

Ao se concluir uma Sprint, é feita a Sprint Review, onde se verifica se o valor desejado foi entregue e se ainda existem pendências que não foram entregues nela. Caso exista pendências, o próximo Sprint Planning deverá considerar se essas entrarão na nova Sprint ou serão abandonadas. Esse ciclo iterativo permite uma rápida adequação da solução às mudanças ao longo do projeto, pois, muitas vezes, uma Sprint tem duração média de uma semana, fazendo com que haja valor sendo entregue o tempo todo e, cada vez mais, a solução obtenha a forma que o cliente deseja.

3 DESENHO DA SOLUÇÃO

Em função da justificativa, dos objetivos apresentados anteriormente e do estudo feito em torno da fundamentação teórica, a solução optada para o projeto foi a construção de uma Matriz de Prioridade, na qual a urgência é informada pelo cliente ao fazer sua solicitação e o impacto é classificado por um serviço NLC da Watson. Essa escolha foi pautada nos objetivos específicos que o projeto visa contemplar, visto que a Matriz de Prioridade garante padrão e repetitividade para a solução, já que nela fica claro quais são as categorias que devem ser priorizadas e qual será a prioridade atribuída em cada caso. O NLC contempla os últimos dois objetivos específicos pois, ao tornar o processo de priorização totalmente automático, os agentes não gastarão tempo com a priorização, nem deverão executar tarefas diferentes de sua operação padrão. A automação só é possível devido à IA por trás do NLC, que compreende a categoria em que o problema do cliente se encaixa.

Todavia, dois passos prévios devem ser feitos antes da solução final: primeiramente, devem ser definidos quais tipos de problemas são considerados de alto impacto; em seguida, deve ser feito um processo manual de classificação de Impacto para gerar dados de treinamento para o NLC. Todo esse trabalho prévio deve ser feito pelo fato de que anteriormente a priorização não era padronizada, portanto se fossem usados os dados de históricos, não seria possível treinar o algoritmo de inteligência artificial de forma adequada.

Uma vez definida a prioridade, deve-se atribuir metas de SLA para os tempos de cada um dos níveis prioritários, essa configuração é feita diretamente no Zendesk (ferramenta de Helpdesk usada na Resultados Digitais). Entretanto, por uma limitação dessa ferramenta, as únicas opções de prioridade disponíveis são “baixa”, “normal”, “alta” e “urgente”. Por acreditar que todo problema é relevante, as lideranças de Suporte optaram por não usar o nível “baixo” de prioridade. As metas desses tempos foram definidas por outro membro da equipe de inteligência operacional do Suporte, já que ele havia sido responsável pelo estudo anterior a respeito de tempos de SLA. Por questões de confidencialidade, os números não podem ser compartilhados, mas, para o entendimento do projeto, basta saber que a

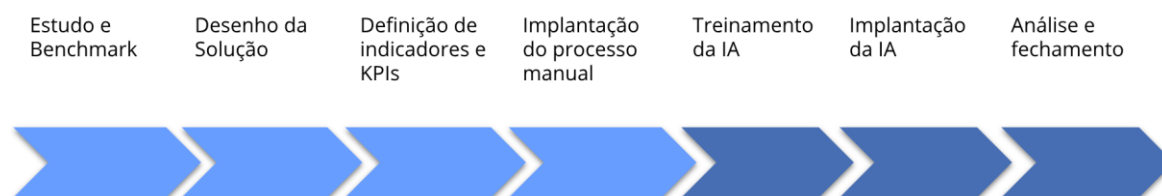
prioridade “urgente” é a que tem tempo mais curto, seguido pela prioridade “alta” e, por último, pela prioridade “normal”.

3.1 Metodologia

O projeto foi desenvolvido como uma maneira adaptada do framework ágil de desenvolvimento de sistemas Scrum. A adaptação foi necessária posto que, por se tratar de um projeto individual, não havia diferentes papéis de Product Owner, Scrum Master e Development Team. Outro aspecto que justifica a adaptação é o fato de que as reuniões de acompanhamento não eram necessárias, já que só uma pessoa estava envolvida no desenvolvimento. Ainda assim, o fluxo de trabalho previsto no Scrum foi respeitado, definindo pequenas entregas semanais como as Sprints do projeto. Ademais, a escolha desse framework se deu pela necessidade de se ter várias entregas intermediárias ao longo do projeto, sem as quais não seria possível treinar o algoritmo final de Inteligência Artificial e o projeto não seria concluído.

De maneira geral, todo o backlog de produto foi definido no começo do projeto, junto às datas cabíveis para cada entrega, visando viabilizar a conclusão do projeto dentro de um tempo hábil. A cada semana, planejava-se a sprint que seria necessária para que as entregas fossem respeitadas. As entregas previstas que compuseram o backlog de produto foram quebradas na seguinte sequência de Sprints:

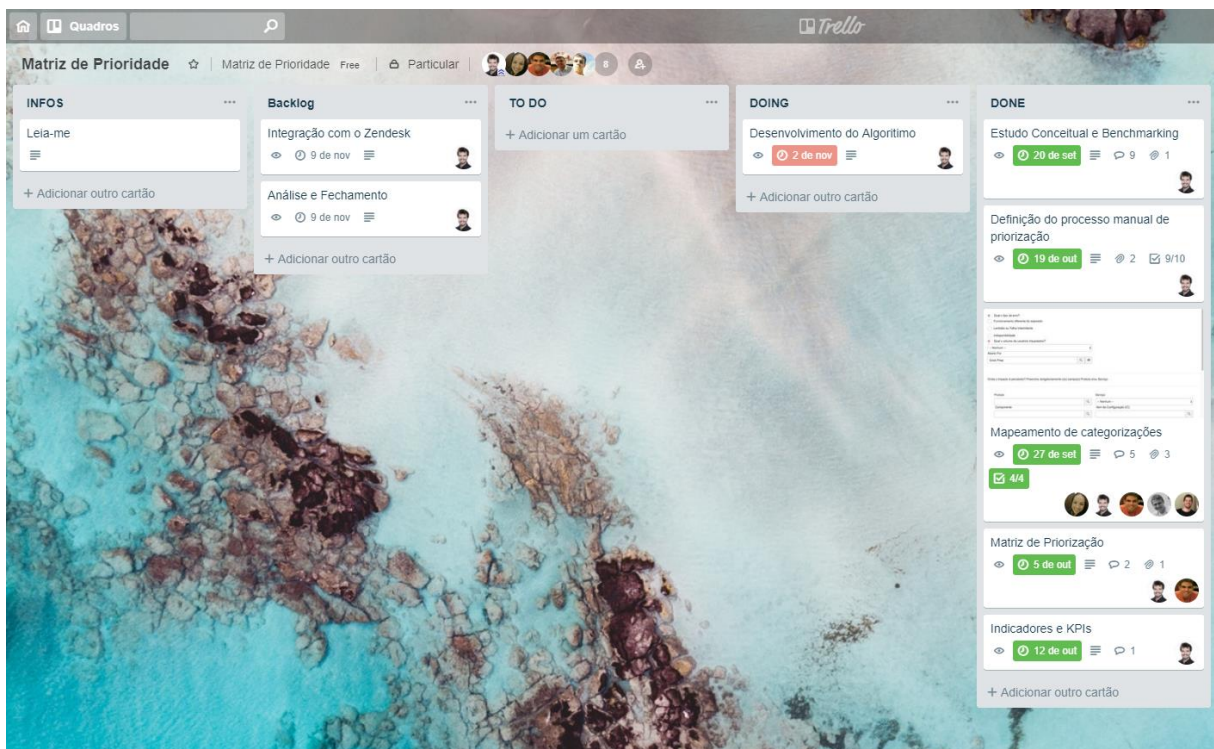
Figura 7 - Etapas do Projeto



Fonte: Arquivo Pessoal

Cada uma dessas Sprints era validada com as lideranças da área de Suporte, para verificar a adequação do projeto à necessidade da empresa. Nem todas as Sprints tiveram a mesma duração, pois algumas demandavam muitas atividades a mais que as outras. Para organizar essas atividades, foi criado um Kanban, no qual se tinha clareza sobre o que deveria ser entregue na Sprint. Cada Sprint gerava tasks (Sprint backlog) dentro dos cards que representavam as etapas do projeto, em forma de check-list. As tasks deviam ser concluídas dentro do tempo da semana; caso não fossem, no início da semana seguinte era avaliado se elas deveriam entrar novamente na Sprint seguinte ou se deveriam ser abandonadas. Esse fluxo permitiu avanços constantes no projeto e diversas oportunidades de repensar as diretrizes iniciais.

Figura 8 - Kanban do Projeto



Fonte: Arquivo Pessoal

3.2 Matriz de Prioridade

Para que se possa desenhar adequadamente a Matriz de Prioridade, deve-se definir quais são as categorias de urgência e quais são as categorias de impacto.

3.2.1 Urgência

A urgência deve expressar o quão crítico o problema é para o cliente. Por meio de benchmarking com outras empresas de SaaS do mercado, verificou-se que a urgência é entregue de maneira mais genérica para o cliente e que, ao mesmo tempo, evita-se usar a palavra “urgência”, já que, muitas vezes, o cliente categoriza como alta apenas para seu chamado ser priorizado. Levando em consideração esses aspectos, foram levantadas as seguintes categorias de urgência:

Impacto nos negócios do cliente	Urgência
Tenho uma dúvida sobre como fazer algo (Limitado)	1
Um recurso não está funcionando como deveria e está atrasando o trabalho (Moderado)	2
Um recurso importante do seu serviço parou de funcionar (Importante)	3
Diversos recursos pararam de funcionar (Crítico)	4

A pergunta feita para o cliente escolher cada uma dessas categorias é: “Qual o impacto do problema para o seu negócio?”. Esse questionamento é feito diretamente no formulário de abertura de chamados na Central de Ajuda.

3.2.2 Impacto

Para possibilitar a definição do Impacto, deve-se compreender quais categorias de problemas são consideradas de alto Impacto no Suporte. Para tal, foi feito um estudo através de uma amostra de 66 chamados de Suporte com alta prioridade, todos sobre diferentes features do produto. Por mais que o número de exemplos fosse pequeno, notou-se claramente que existia um padrão no que era considerado prioritário. Assim, foram definidas as seguintes categorias:

Categorias Mapeadas	Descrição	Pesos
O lead do cliente está impedido de seguir fluxo de nutrição?	O problema está impedindo que o lead do cliente evolua no funil de vendas (lead, lead qualificado, oportunidade, venda) ou o lead não está conseguindo ser enviado para um outro sistema que faria sua nutrição (CRM por exemplo)	200

Está impedindo o cliente de interagir com os leads?	O cliente não consegue fazer com que seus leads vejam/se convertam em campanhas, e-mails, Landing Pages, Formulários ou qualquer forma de contato direto.	200
Afetou a imagem do cliente?	Algum canal onde a marca do cliente está publica para todos acessarem está, ou indisponível, ou com erros	500
Não consegue acessar o RD Station?	A ferramenta está indisponível para o cliente	300

Os pesos foram definidos pelas lideranças do Suporte em função do que consideravam ter maior impacto no sucesso do cliente. Como o cliente pode se encaixar em mais de uma dessas categorias por vez, foram definidas faixas de peso que refletem o impacto do problema:

Faixa do Peso	Impacto
0 – 199	1
200 – 499	2
500 – 799	3
800 ou mais	4

Ao somar o peso gerado por cada categoria em que o chamado se encaixa, obtém-se o impacto.

3.2.3 Construção da Matriz

Definidas as categorias que geram os valores de Impacto e Urgência, define-se também a matriz que, a partir desses dois dados, retorna a prioridade que o chamado deve ter. Esta pode ser vista a seguir:

Urgência	4	Alta	Alta	Urgente	Urgente
	3	Normal	Alta	Alta	Urgente
	2	Normal	Normal	Alta	Alta
	1	Normal	Normal	Normal	Normal
		1	2	3	4
		Impacto			

Cada posição da Matriz teve que ser escolhida adequadamente, pois implica nos valores finais de prioridade que o ticket terá. A lógica que se buscou seguir na construção desses valores foi a seguinte:

- Minimizar a influência da urgência na definição de prioridade. Como não se sabe o comportamento do cliente, é importante evitar que muitos tickets tenham prioridades elevadas e sobrecarreguem o tempo hábil de resposta dos agentes. Na matriz, isso se reflete no fato de que, para quase todos os valores de urgência, a prioridade é normal na primeira coluna, assim como o fato de que na segunda coluna também há pouca influência da urgência.
- As maiores prioridades se concentram, majoritariamente, nos cenários em que o problema do cliente se encaixa em mais de uma categoria de impacto (colunas 3 e 4). Por motivos probabilísticos, isso minimiza a ocorrência de chamados com prioridade alta e urgente.

3.3 Processo de Priorização

Para tornar o uso da Matriz de Prioridade operacional, é preciso definir um processo de priorização adequado à operação de Suporte como um todo. De maneira simplificada, a operação de atendimento começa no nível 1 de atendimento (N1), o qual é responsável por chamados de menor complexidade técnica e, por isso, muitas vezes é o nível de resposta mais rápida. Caso o agente desse nível não tenha conhecimento técnico o suficiente para responder o ticket que está tratando, o atendimento é encaminhado para o nível 2. Como um ticket nunca pode “pular um nível”, todo o esforço de priorização do projeto se deu em N1. Em N1 os agentes atendem aos chamados em uma fila única, onde os tickets são ordenados por proximidade da violação de SLA e agrupados por prioridade. A operação do agente é sempre atribuir para si o primeiro ticket da fila (aquele que está no topo da lista). Até então, independentemente da violação do SLA, os tickets de prioridade alta e urgente eram agrupados no topo da fila e, logo abaixo, vinham os tickets de prioridade normal.

Visando ter uma operação baseada inteiramente apenas no cumprimento das políticas de SLA, a proposta foi de que a fila fosse inteiramente ordenada somente por tempo de violação de SLA, sem agrupamentos. Concomitantemente, tickets com prioridade alta e urgente assumiriam políticas de SLA mais curtas, o que

faria com que subissem na fila automaticamente. Junto a isso, tickets sem prioridade devem aparecer no final da fila de atendimento, pois ainda não se sabe qual política de SLA será atribuída a eles.

Definido esse cenário, foram escalados três agentes para realizar o processo de priorização. Sua função era monitorar os tickets sem prioridade em escalas por período (manhã, tarde e fim da tarde) e, de tempos em tempos, priorizar esses tickets atribuindo um valor de impacto. Sabe-se que esse tipo de ação não gera uma priorização real, já que chamados prioritários podem acabar demorando para aparecer no topo da fila. Ainda assim, o processo foi executado dessa forma, pois o foco da ação é gerar dados relevantes para o algoritmo da IA e não necessariamente otimizar a operação de atendimento. Tendo o valor de Impacto classificado pelo agente, o Helpdesk comunicava-se com a Matriz de Prioridade, que então, atribuía um valor de prioridade ao chamado.

3.4 Estrutura conceitual do Sistema

Buscando executar os processos manual e automático de forma a contemplar os aspectos levantados na Matriz de Prioridade, se faz necessário que o sistema projetado cumpra uma série de requisitos funcionais e não funcionais. Por requisito funcional, entende-se qualquer requisito essencial para o funcionamento adequado do sistema. Já por requisito não funcional, entende-se os aspectos que permitem bom uso do sistema, mas que seu não cumprimento não implica em inoperabilidade do sistema. Ambos os tipos de requisitos possuem igual importância, pois muitas vezes o não cumprimento de um aspecto de uso inviabiliza a boa execução do processo.

Como requisitos funcionais foram definidos os seguintes elementos:

- Classificar precisamente os tipos de chamados que devem ser priorizados
- Viabilizar regras para que a solução seja incorporada naturalmente ao fluxo de atendimento
- Verificar que as regras de priorização não possuem exceções, independentemente do problema informado pelo cliente

- Garantir que o sistema projetado esteja bem dimensionado para o que vai executar, suportando picos de demanda
- Garantir que o sistema projetado tenha fator de falha mínimo ou zero

Quanto aos requisitos não-funcionais, os elementos definidos foram:

- Garantir uma forma intuitiva e clara de priorização para os agentes
- Prover uma solução escalável de fácil manutenção
- Não modificar/deletar o conteúdo ou respostas contidas no chamado.

Os requisitos foram todos derivados dos objetivos definidos no início do projeto e, também, através de condições estabelecidas pelo líder técnico da área de Inteligência Operacional do Suporte. Para cumprir esses requisitos, foram usadas três diferentes aplicações web com diversas funcionalidades: Zendesk, Zapier e Natural Language Classifier. A seguir, explica-se como cada uma dessas aplicações busca cumprir esses requisitos.

Pelo fato de a solução ser construída de forma modular sob aplicações web referência em seus segmentos, o sistema é relativamente simples e escalável, já cumprindo esse requisito não-funcional. Outro requisito que já se cumpre nessa estrutura de solução é o de um bom dimensionamento dos sistemas, pois todas são soluções robustas de mercado, com capacidade de processar altos volumes de dados. Um exemplo disso é o limite de requisições por API: enquanto no projeto é feita uma requisição em média a cada 8 minutos, o Zendesk, que é o sistema com menor limite de requisição na API, permite que sejam feitas 700 requisições em 1 minuto [24].

3.4.1 Zendesk

O Zendesk é a ferramenta líder de mercado na categoria de Help Desk e nela é feita a operação de atendimento da Resultados Digitais. Como ferramenta de Suporte, sua principal função é armazenar, organizar e apresentar para o agente todas as informações necessárias para que um ticket seja resolvido, tudo isso através de uma interface simples e de fácil utilização. A ferramenta ainda permite

disparar ações baseadas em regras condicionais, funcionalidade que viabiliza que uma série de automações no fluxo de atendimento sejam feitas dentro da aplicação.

No contexto do projeto, o Zendesk é primordial para gerir de forma eficiente todas as informações do chamado e garantir que cada chamado esteja adequadamente priorizado. Além disso, é a ferramenta sob a qual acontece todo o processo de atendimento e, portanto, permite a garantia de que a priorização dos chamados acontece de acordo com o fluxo de atendimento já estabelecido. Nela, também, verifica-se que nenhuma importante sensível do chamado será modificada durante a execução do projeto.

Outro aspecto positivo da ferramenta é a familiaridade dos agentes, permitindo que a curva de aprendizado durante o processo manual seja consideravelmente superior e evitando que haja confusões quanto a como usar o sistema para classificar os chamados. Isso garante o requisito de se ter uma forma intuitiva de priorizar os chamados.

3.4.2 Zapier

Zapier é uma ferramenta amplamente utilizada para integrar diversas aplicações web, tendo como principais diferenciais a facilidade de uso e a variedade de integrações nativas. Dessa variedade, a vantagem é a possibilidade de acesso a valores de diversas aplicações sem necessidade de configuração da requisição HTTP. A facilidade é viabilizada através das integrações nativas que a ferramenta possui, nelas já estão configurados todos os aspectos técnicos para viabilizar as trocas de mensagem, entretanto, como cada sistema tem suas características próprias, não é todo aplicativo que possui uma integração nativa.

Sua lógica de funcionamento baseia-se em Zaps, cadeias de ações executadas após a ocorrência de um primeiro “Trigger”. Tais ações podem ser: modificar valores em aplicações com integração nativa, tratar requisições HTTP, manipular dados via script (Python ou Javascript) ou uma série de outras opções. Outra vantagem da ferramenta são os logs gerados a cada instância de execução de

um Zap, o que permite um acompanhamento próximo do funcionamento do sistema e facilita a investigação técnica em casos de falha.

Na solução prevista, o Zapier desempenha função de articulador das mensagens entre as aplicações web, além de permitir a viabilização técnica da Matriz de Prioridade. Como responsável pelo intermédio nas mensagens, seu papel é primordial na garantia de falha mínima, isso porque essa aplicação já é uma ferramenta própria para comunicação entre sistemas. Portanto, diversos protocolos de segurança operam para entregar e receber as mensagens com qualidade, enquanto os logs que a ferramenta gera permitem o rastreamento de possíveis falhas e facilitam sua correção. A viabilização técnica da Matriz de Prioridade também deve ser desenvolvida de tal forma que não exista exceções no processamento da prioridade. Dessa forma, o código que fará esse tratamento deve ter condição de contorno em que, em cenários não previstos, a prioridade aplicada por default é “normal”.

3.4.3 Natural Language Classifier

Como módulo de IA para interpretar os textos e identificar em que categorias ele se encaixa, foi escolhido o serviço Natural Language Classifier da Watson (IBM). Seu funcionamento é extremamente simples, basta introduzir uma planilha do tipo CSV, onde a primeira coluna são os textos e as demais colunas são as categorias em que o texto se encaixa (permitindo que um texto faça parte de diversas categorias), esse arquivo servirá como a base de treinamento do NLC.

Ao prover o arquivo CSV para o NLC, ele automaticamente identifica em que classes se encaixa cada texto, bastando apenas selecionar a opção “treinar” para que o algoritmo inicie seu treinamento. Uma vez treinado, ele disponibiliza uma interface de testes e uma API onde textos podem ser inseridos, e o sistema retorna o percentual de proximidade com cada uma das categorias treinadas.

A escolha dessa aplicação foi feita levando-se em conta a simplicidade tanto de uso e quanto de manutenção do sistema, além disso ela permite alta taxa de

requisição a um baixíssimo preço (mais detalhes de custos no capítulo de resultados). Esse sistema também possui algoritmos de interpretação textual de alta qualidade que poucas ferramentas do mercado têm: o único que entrega resultados similares é o LUIS da Microsoft, porém, falta uma solução específica para categorização, como existe no caso do NLC da Watson. Posto a qualidade da interpretação, essa é a solução que mais contempla o requisito de categorização dos chamados com precisão.

4 DESENVOLVIMENTO E IMPLEMENTAÇÃO

O desenvolvimento do sistema por trás do processo de priorização foi dividido de acordo com o método MVC explicado anteriormente. Tal fato permitiu que diversas aplicações web pudessem ser usadas em diferentes partes da solução. Como a comunicação entre as diferentes aplicações ocorre por meio do Zapier, o sistema preserva a garantia de requisição, mesmo utilizando diversas tecnologias. Nesse capítulo, o intuito é apresentar como se deu a implementação técnica das escolhas feitas em torno das aplicações apresentadas no capítulo anterior.

Inicialmente, todo o sistema foi projetado para viabilizar o processo de priorização manual. Assim que a obtenção de dados foi suficiente para que o algoritmo de NLC convergisse, ele foi implantado. Para suportar a implantação desse novo módulo, algumas mudanças no sistema precisaram ser feitas. Tanto o sistema no cenário anterior, quanto com a implantação do NLC estão descritos nas seções a seguir.

4.1 Modelos e diagramas do sistema

Em sua totalidade, o sistema é composto pela conexão de Zendesk, Zapier e Natural Language Classifier. O funcionamento sincronizado dessas aplicações garante o fluxo correto de trabalho para a priorização dos chamados. Essa seção apresenta os modelos em BPMN e diagramas UML que ilustram tal funcionamento.

4.1.1 Modelo do processo em BPMN

Os processos manual e automático são muito semelhantes. Sua principal diferença é que, no segundo, o agente deixa de ser um ator nas atividades e o NLC passa a exercer sua função. O processo de priorização é bem linear, como se pode perceber pela ausência de gateways no fluxo (com exceção daquele que verifica se o chamado está no processo de priorização). Tal fato é um ótimo sinal para a robustez da solução, visto que, ao se reduzir a variedade de cenários possíveis, os riscos de falha por exceção são minimizados.

A modelagem do processo foi importante para se ter uma clara compreensão sobre todas as funções que o sistema deve exercer. Esse modelo foi criado antes que qualquer desenvolvimento fosse executado nas ferramentas e serviu como guia para compreender se as ações, durante as etapas de priorização, estavam sendo feitas corretamente. Além disso, esse processo foi publicado internamente no servidor da Resultados Digitais e servia como fonte de consulta para que as lideranças do Suporte pudessem ter visão macro de como se dá o processo de priorização.

Figura 9 - Modelo em BPMN do processo manual

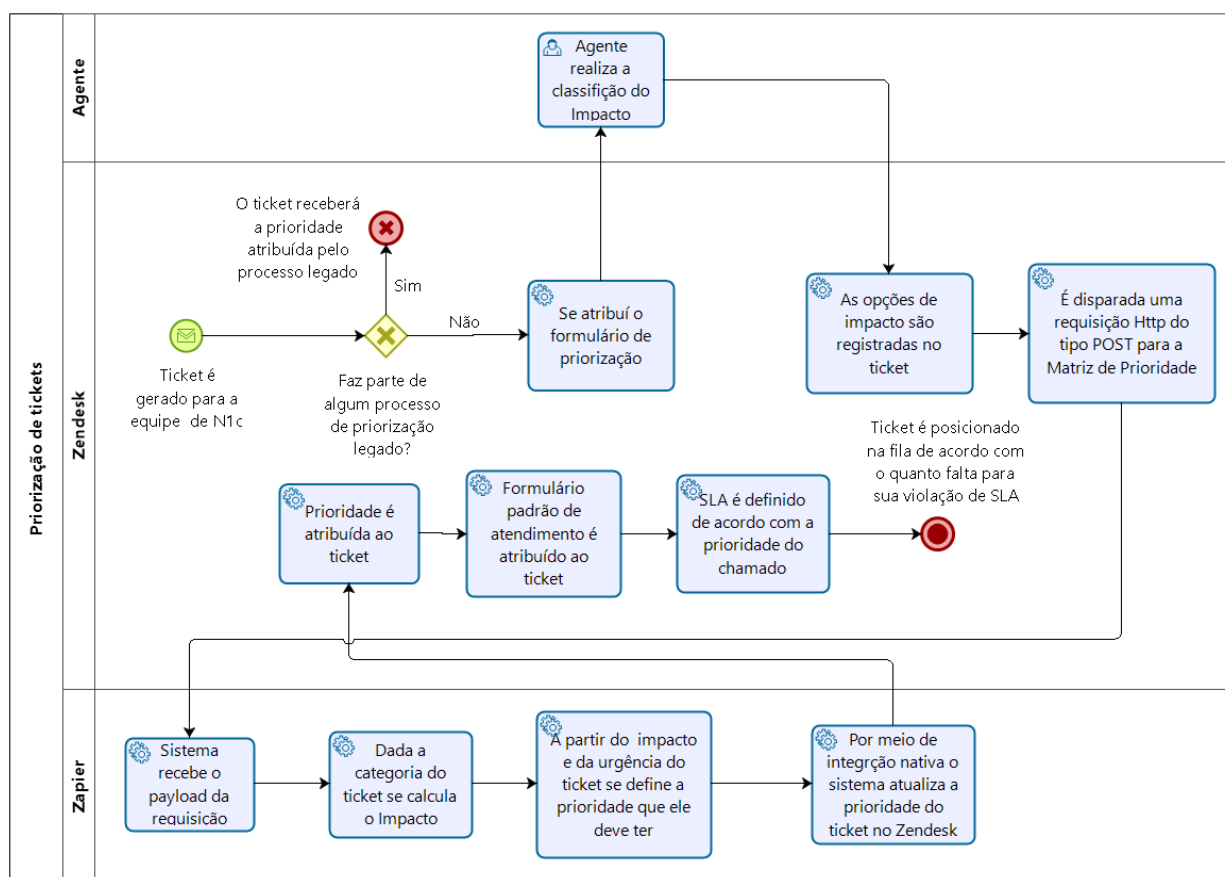
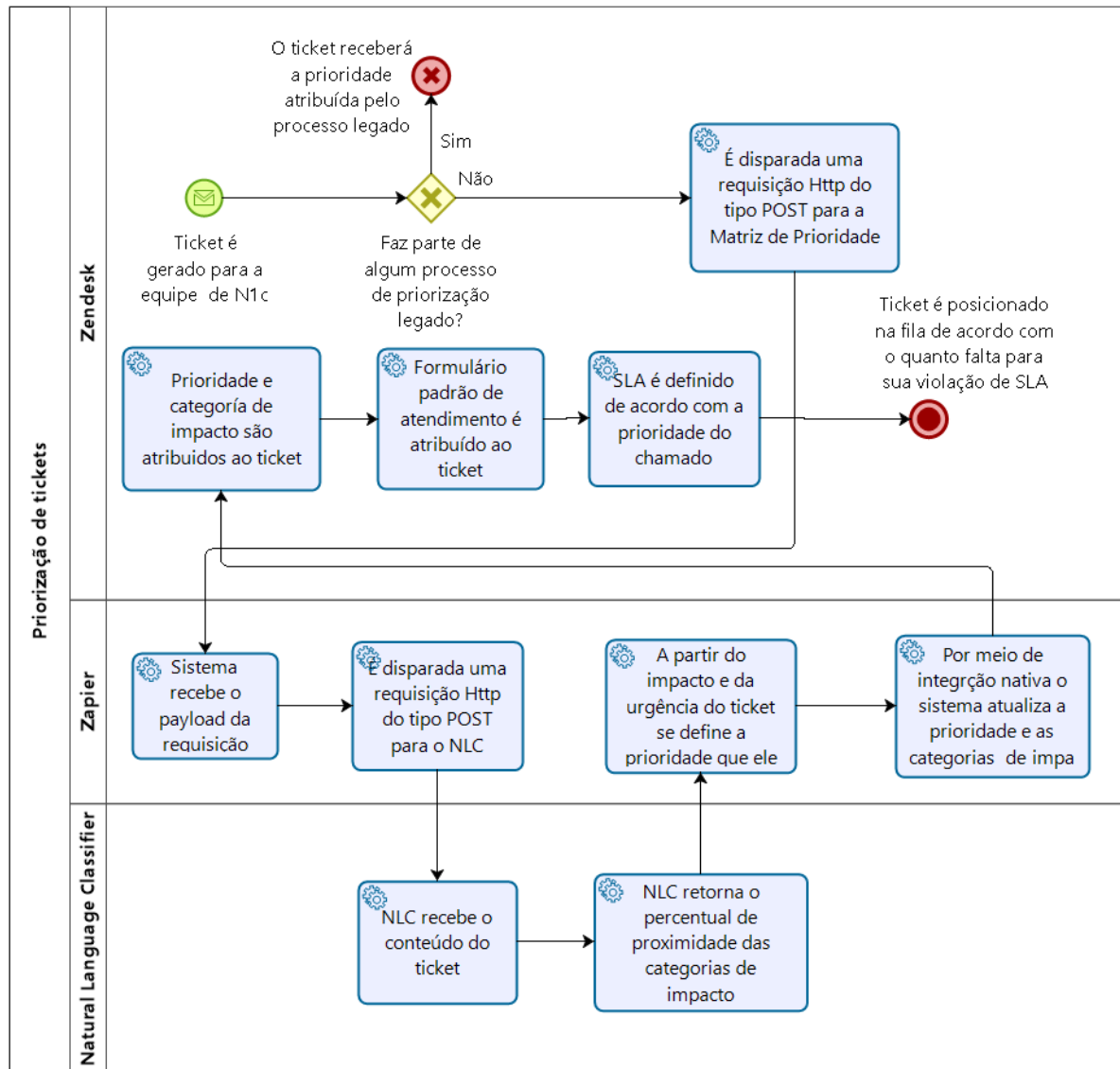


Figura 10 - Modelo em BPMN



Fonte: Arquivo Pessoal

4.1.2 Diagrama de Sequência em UML

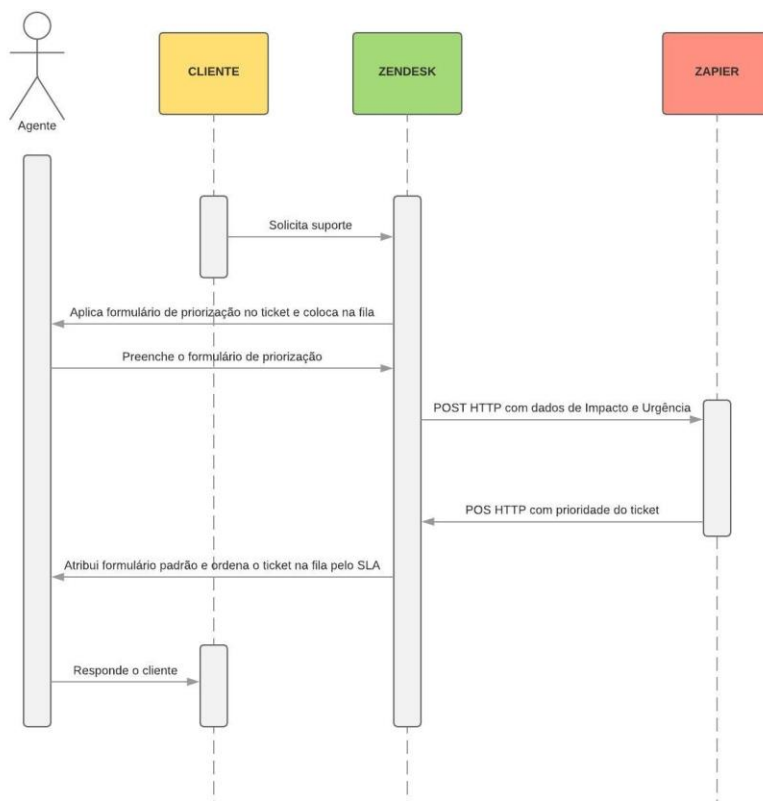
O Diagrama de sequência em UML representa as trocas de mensagens entre as aplicações contempladas na construção do sistema. Para minimizar a complexidade técnica do projeto, todas as aplicações comunicam-se seguindo o protocolo HTTP, por requisições do tipo POST e usando mensagens no formato JSON. As trocas de mensagem com o cliente não fazem parte do sistema, mas

foram incluídas no diagrama de seqüência para facilitar a compreensão do fluxo de informações.

Figura 11 - Diagrama de seqüência do processo manual

DIAGRAMA DE SEQUÊNCIA MANUAL

Fernando Teures Redondo | November 17, 2018

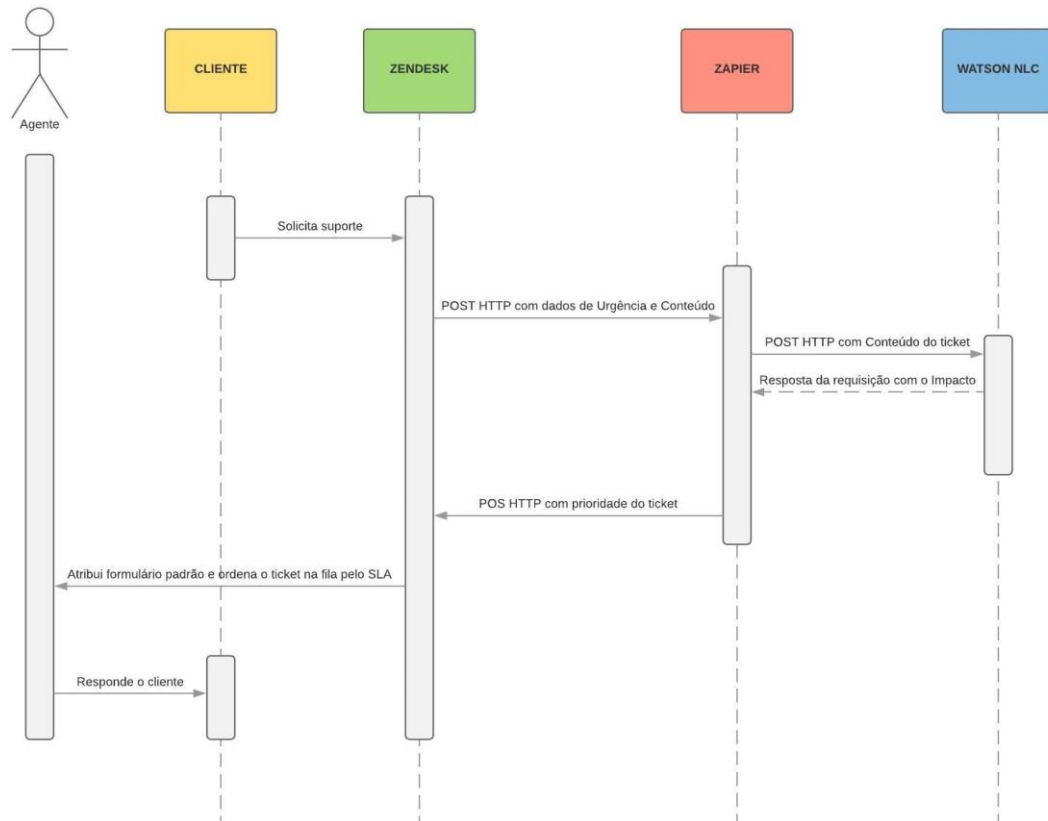


Fonte: Arquivo Pessoal

Figura 12 - Diagrama de sequência do processo automático

DIAGRAMA DE SEQUÊNCIA AUTMÁTICO

Fernando Teures Redondo | November 17, 2018

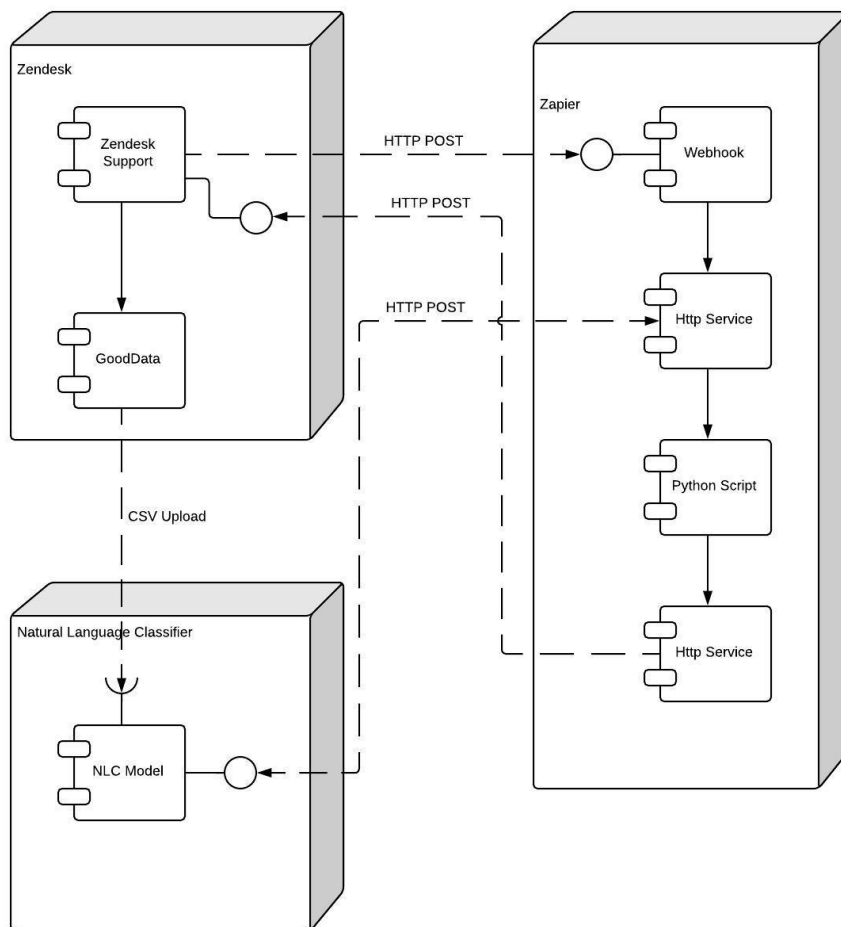


Fonte: Arquivo Pessoal

4.1.3 Diagrama de Implementação em UML

A arquitetura do sistema pode ser representada em um diagrama de implementação. Ele mostra todas as aplicações utilizadas e como elas se comunicam, além de conceder uma visão mais detalhada dos serviços que executam dentro dessas aplicações. No caso do projeto, apenas um diagrama de implementação foi feito: o do sistema final com o NLC já implantado. Essa escolha justifica-se pela pouca diferença em relação ao manual, no qual apenas não haveria a requisição do Zapier para o NLC.

Figura 13 - Diagrama de implementação do sistema



Fonte: Arquivo Pessoal

4.2 Model (Nível de persistência dos dados)

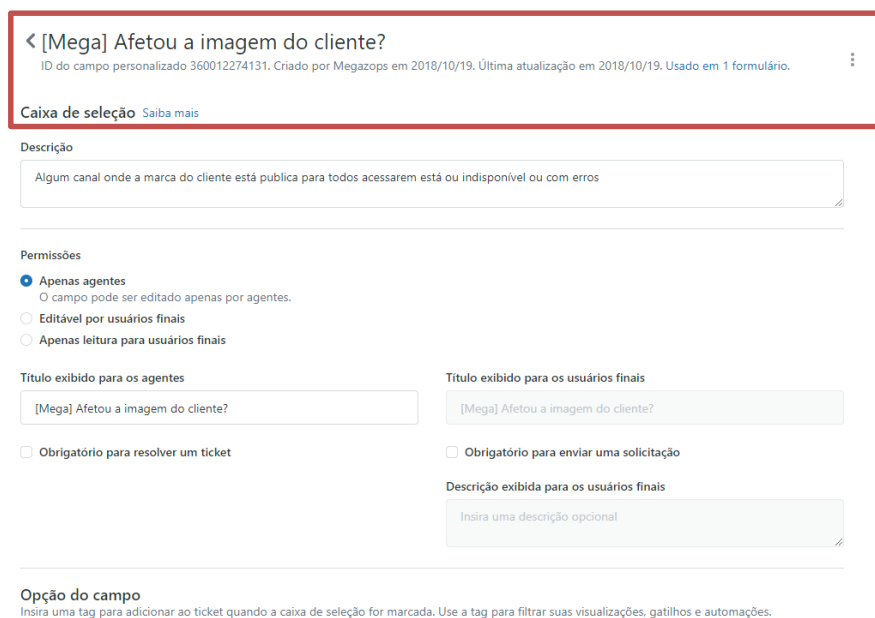
O nível de modelo é responsável por salvar os dados para análises de desempenho do projeto. Essencialmente, o que precisa ser salvo é a prioridade associada aos chamados. Entretanto, no estágio de processo manual, também é importante persistir as categorias de impacto para que o algoritmo de NLC possa ser treinado. O Zendesk já conta com um banco de dados próprio e toda sua gestão é feita pela ferramenta de análise GoodData, que roda sobre esse banco. Assim sendo, toda a persistência de dados do projeto foi feita sob o banco do próprio Zendesk.

4.2.1 Dados de “Impacto” e “Urgência” do ticket

Para persistir os dados de Impacto no banco, foram criados 6 novos atributos na tabela de “tickets” no Zendesk: um para cada uma das 4 categorias de Impacto, um para armazenar o valor de Urgência escolhido pelo cliente e um usado nas regras do Controller. Esses atributos são essencialmente novos campos do ticket.

Para o caso de Impacto, os atributos foram criados como tipo “Caixa de Seleção” (Binário), para que mais de uma categoria pudesse ser marcada como “True”. Já no caso da Urgência, o campo foi criado como tipo “Lista Suspensa”, para que o valor do atributo assumisse somente uma das opções disponíveis na categorização de urgência (capítulo 3). Uma vez criado um campo de ticket, o Zendesk já interpreta sua existência e o deixa disponível para uso nos formulários (View) e gatilhos (Controller).

Figura 14 - Exemplo de campo de Impacto



< [Mega] Afetou a imagem do cliente?
ID do campo personalizado 360012274131. Criado por Megazops em 2018/10/19. Última atualização em 2018/10/19. Usado em 1 formulário.

Caixa de seleção Saiba mais

Descrição

Algun canal onde a marca do cliente está publica para todos acessarem está ou indisponível ou com erros

Permissões

- Apenas agentes
O campo pode ser editado apenas por agentes.
- Editável por usuários finais
- Apenas leitura para usuários finais

Título exibido para os agentes

[Mega] Afetou a imagem do cliente?

Obrigatório para resolver um ticket

Título exibido para os usuários finais

[Mega] Afetou a imagem do cliente?

Obrigatório para enviar uma solicitação

Descrição exibida para os usuários finais

Insira uma descrição opcional

Opção do campo

Insira uma tag para adicionar ao ticket quando a caixa de seleção for marcada. Use a tag para filtrar suas visualizações, gatilhos e automações.

Fonte: Arquivo Pessoal

Figura 15 - Exemplo de campo de Urgência

< [Mega] Urgência
ID do campo personalizado 360012302492. Criado por Megazops em 2018/10/22. Última atualização em 2018/10/22. Usado em 2 formulários.

Lista suspensa Saiba mais

Descrição
Insira uma descrição opcional

Permissões
 Apenas agentes
 Editável por usuários finais
 O campo pode ser editado pelo usuário final ao enviar uma solicitação.
 Apenas leitura para usuários finais

Título exibido para os agentes
[Mega] Urgência

Título exibido para os usuários finais
{{dc.campo_urgencia_qual_o_impacto_do_problema_para_o_seu_negocio}}

Obrigatório para resolver um ticket
 Obrigatório para enviar uma solicitação

Descrição exibida para os usuários finais
Insira uma descrição opcional

Valores de campo [Mostrar tags](#) [Baixar CSV](#)

- [[dc.campo_urgencia_voce_tem_uma_duvida_sobre_como_fazer_algo]]
- [[dc.campo_urgencia_um_recurso_nao_esta_funcionando]]
- [[dc.campo_urgencia_um_recurso_importante_do_seu_servico_parou_de_funcionar]]
- [[dc.campo_urgencia_diversos_recursos_nao_vao_funcionar]]

Valor

Fonte: Arquivo Pessoal

As imagens apresentam como esses campos são criados e configurados dentro do Zendesk. Os textos contidos em chaves duplas são chamados de conteúdo dinâmico. Sua função é apresentar um determinado conteúdo no idioma para o qual o navegador do usuário está configurado. Para o projeto, os campos de ticket criados para o projeto foram:

- [Mega] O lead do cliente está impedido de seguir o fluxo de nutrição?
- [Mega] Está impedindo o cliente de interagir com os leads?
- [Mega] Afetou a imagem do cliente?
- [Mega] Não consegue acessar o RD Station?
- [Mega] Urgência
- [Mega] Revisei a Prioridade

4.2.2 Dados de Treinamento do NLC

Para treinar o algoritmo do NLC é necessário, além dos valores de Impacto e Urgência do ticket, que se tenha acesso ao conteúdo do chamado. Como se trata de

um atributo muito grande, o GoodData não permite acesso direto a esse dado para exportação. Logo, é necessário que a exportação desses dados seja feita de outra forma.

O Zendesk permite que diversos aplicativos desenvolvidos pela própria empresa Zendesk ou por terceiros sejam instalados junto à aplicação principal. Destes, destaca-se o aplicativo “Ticket Exporter”, cuja função, como o nome sugere, é exportar diversos atributos dos tickets, inclusive seu conteúdo. Para exportar os tickets, é preciso que eles façam parte de alguma das filas de atendimento. No Zendesk, estas filas são chamadas de visualizações e as regras que definem quais tickets fazem parte da fila podem ser configuradas livremente. Portanto, para extrair os tickets do projeto de priorização, bastou-se criar uma visualização que contivesse os tickets priorizados durante o processo manual, esses tickets estavam identificados por meio da Tag que era aplicada no instante em que o conteúdo de um ticket era enviado para o Zapier.

Figura 16 - Regras da visualização para exportação de tickets

[Mega] Tickets no processo da Matriz de Priorização

Título da visualização

[Mega] Tickets no processo da Matriz de Priorização

Atende **todas** as seguintes condições:

Ticket: Tags Contém ao menos um dos seguintes

Ticket: Grupo Time CS Operations

Adicionar condição

Atende **qualquer** das seguintes condições:

- Selecione a condição. -

Adicionar condição

Visualizar correspondência das condições acima

Fonte: Arquivo Pessoal

Nesta visualização, são capturados os tickets que possuem etiqueta (mais detalhes na seção Controller) e excluídos os que pertencem ao grupo de CS Operations, pois esse grupo atende somente demandas internas e não faz parte do projeto.

4.3 Controller (controle do fluxo de informação)

No nível controller, são definidas todas as regras que regem o comportamento do sistema, desde a manipulação de dados até o envio de requisições. No projeto de priorização, o controller do sistema fica dividido em duas diferentes aplicações; enquanto as regras referentes aos atributos e manipulação dos tickets foram feitas no Zendesk, as regras referentes ao cálculo da Matriz de Prioridade e acesso ao NLC foram feitas no Zapier. Essa abordagem ocorreu pelo fato de que o Zendesk possui algumas limitações em relação a transformações de valores, como operações de soma e modificação de strings; além disso, ele não gera logs de requisição detalhados como o Zapier.

4.3.1 Manipulação dos tickets (Zendesk)

As regras de negócio no Zendesk são configuradas de duas maneiras: por meio de Gatilhos e Automações. Ambas são muito parecidas, tendo como única diferença o fato de que gatilhos geram ações instantâneas, enquanto automações permitem trabalhar com janelas temporais. Como no projeto desejava-se que todas as ações fossem executadas imediatamente, apenas gatilhos foram utilizados. Gatilhos operam com uma lógica simples de regra: caso as condições sejam cumpridas, uma ou mais ações são realizadas, o que permite manipular livremente uma série de dados existentes no Zendesk.

Para que a ação do gatilho de enviar uma requisição HTTP a um determinado alvo funcione, esse alvo precisa ser configurado dentro do Zendesk. A configuração é simples e basta informar a URL da requisição, o método HTTP e o formato da mensagem em JSON.

Figura 17 - Configuração de alvo no Zendesk

Destino HTTP

Título

URL
A URL alvo, incluindo o protocolo (https ou http é OK) e o caminho.
Exemplos válidos:
• http://somedomain/a/path

Método

Tipo de conteúdo

Autenticação básica Ativado
Adicione as credenciais se o alvo precisar de autenticação de nome de usuário/senha.

Fonte: Arquivo Pessoal

No projeto, foi necessária a criação de dois gatilhos para operacionalizar o processo de priorização manual dos agentes, além de um terceiro que foi utilizado após a implantação do NLC. A seguir, estão listados esses gatilhos e o que fazem:

- [Mega] Atribuir Formulário de Priorização – Aplica o formulário que os agentes usavam para categorizar o impacto em tickets sem prioridade. Usado somente durante o processo de priorização manual
- [Mega] Matriz de Priorização no Zapier – Caso o formulário de priorização esteja aplicado ao chamado e caso o campo “Revisei a Prioridade” esteja marcado (True), ao atualizar o ticket é disparada uma requisição HTTP ao Zapier contendo as informações de Impacto, Urgência, ID do chamado e Conteúdo do chamado. Também empregado somente durante o processo de priorização manual.
- [Mega] Matriz de Priorização no Zapier Automática – Caso o ticket seja criado e pertença ao grupo Customer Heroes (time de N1) ou não tenha grupo, ao atualizar o ticket é disparada uma requisição HTTP ao Zapier contendo as informações de Impacto, Urgência, ID do chamado e Conteúdo do chamado. Esse gatilho foi ativado apenas quando o NLC foi implantado.

Figura 18 - Exemplo de gatilho no Zendesk

< [Mega] Matriz de Priorização no Zapier
Última atualização em 13 de nov de 2018 13:17 • Histórico de revisão

Nome do gatilho*
[Mega] Matriz de Priorização no Zapier

Descrição
Insira uma descrição opcional

Condições
Condições que devem ser atendidas para o gatilho ser executado

TODAS

Formulário É [Mega] Formulário para priorização de chamados

[Mega] Revizei a Prioridade É Selecionado

Adicionar condição

QUALQUER UMA

Adicionar condição

Ações
Ações que ocorrerão se as condições globais forem atendidas

Notificar alvo [Mega] Zapier - Matriz de Priorização

Corpo do JSON

```
1 [{"Ticket ID" : "{{ticket.id}}", "Motivo 1" : "{{ticket.ticket_field_360012223372}}", "Motivo 2" : "{{ticket.ticket_field_360012223392}}", "Motivo 3" : "{{ticket.ticket_field_360012274131}}", "Motivo 4" : "{{ticket.ticket_field_360012274151}}", "Urgencia" : "{{ticket.ticket_field_360012302492}}", "Descricao" : "{{ticket.description}}"}]
```

Fonte: Arquivo Pessoal

Nesse exemplo se mostra a estrutura do Gatilho “[Mega] Matriz de Prioridade no Zapier”, evidenciando as condições necessárias para seu funcionamento e apresentando a mensagem em JSON que é disparada por ele. Os valores entre aspas duplas são placeholders que assumem o valor referente aos atributos do ticket que tiver ativado esse gatilho.

4.3.2 Matriz de Prioridade (Zapier)

Toda a execução no Zapier do projeto foi resumida a somente um Zap, o qual é disparado quando se recebe uma requisição HTTP do gatilho “[Mega] Matriz de Priorização no Zapier”, do Zendesk. No caso de priorização manual, isso faz com que chegue uma mensagem contendo as categorias de Impacto em que o ticket se encaixa, bem como a categoria de urgência do ticket, seu ID e seu conteúdo. Em seguida, o Zap executa a ação de transformar a categoria de urgência em um número de urgência e passa todos esses dados a um script Python. Tal script calcula o peso de Impacto em função das categorias validadas no ticket, calcula o valor numérico do Impacto e identifica, através do Impacto e da Urgência, qual prioridade o chamado deve ter. Uma vez tendo a prioridade, basta atualizar esse valor no ticket através da integração nativa do Zapier com o Zendesk.

Figura 19 - Zap do processo manual

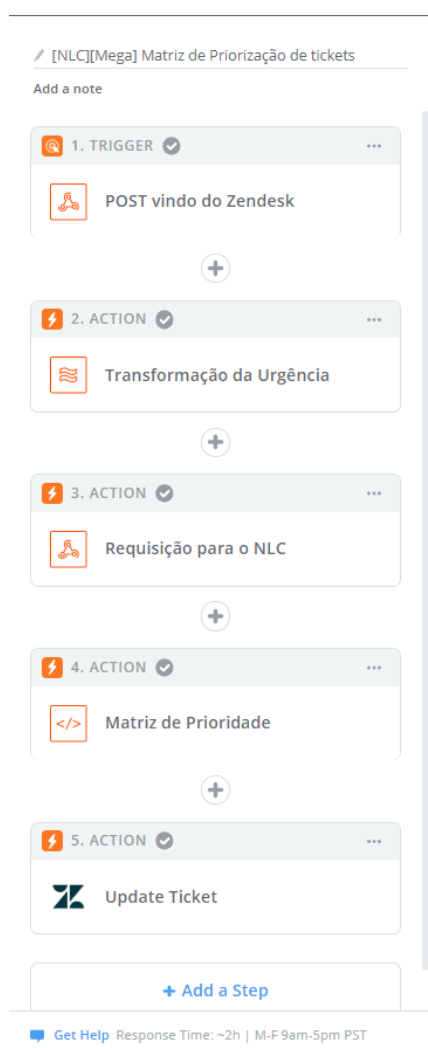


Fonte: Arquivo Pessoal

Para o caso da priorização automática, a execução do Zap é muito semelhante, diferindo com relação à informação de Impacto, que não é passada na mensagem enviada do Zendesk, devendo ser incluída uma ação de requisitar o NLC via HTTP antes do cálculo da Matriz de Prioridade. O NLC da Watson disponibiliza uma API que, ao receber uma requisição POST com um texto, retorna a probabilidade encaixe do texto em cada uma das categorias do modelo; assim, basta que a requisição do Zapier forneça o conteúdo do ticket para que o retorno seja suficiente para compor o impacto. O script Python também precisa ser modificado para interpretar as categorias de impacto em termos percentuais, pois

antes o impacto era recebido como binário do Zendesk; por convenção, definiu-se que a margem de confiabilidade para considerar uma categoria como válida é de 85%. O passo seguinte de atualização da prioridade no ticket no Zendesk continua o mesmo, com a ressalva de que também são atualizados os campos relativos à categoria de Impacto que o NLC calculou, de modo a garantir que dados de impacto sejam gerados frequentemente e que o treinamento do NLC possa ser continuado mesmo durante sua execução.

Figura 20 - Zap do processo automático



Fonte: Arquivo Pessoal

Uma observação importante deve ser feita sobre o script em Python que faz o cálculo da Matriz de Prioridade é a de que ele foi construído usando comandos chamado "Try" e "Except", que permitem fazer com que, a qualquer momento em

que haja uma inconsistência na execução do Try, o código passe a executar o Except. Esse recurso é útil para tratar exceções não previstas na solução que poderiam causar falhas ou deadlocks. No projeto, especificamente, esse era um requisito funcional importante que deveria ser cumprido. O que se fez foi interpretar os dados do NLC sob o comando Try e no caso do Except ser acionado se aplicava uma prioridade “normal” como padrão.

4.4 View (interface para o usuário)

A View do sistema refere-se fundamentalmente à interface que o sistema entrega para seus usuários. No caso do projeto de priorização, a interface do agente só se faz necessária durante a etapa de priorização manual já que, uma vez que o NLC esteja treinado, não existirá interação desses agentes com o sistema de priorização. Por outro lado, a interface do cliente para a classificação de Urgência será sempre necessária.

No Zendesk, a interface para modificar os atributos de um ticket é sempre o formulário de ticket. Esses formulários podem ser configurados para serem editados por usuários finais ou somente por agentes e podem, ainda, ser atribuídos a um ticket a qualquer momento, por meio de um gatilho. Na Resultados Digitais, trabalha-se com apenas um formulário de ticket para todos os clientes, denominado Formulário de Abertura de Chamados. Já os formulários de ticket para agente variam conforme a circunstância do chamado, mas, por default, o Zendesk aplica também o Formulário de Abertura de Chamados.

4.4.1 Formulário de Abertura de Chamados

O formulário de abertura de chamados entrega as quatro opções de urgência para o cliente categorizar seu problema. Assim que o chamado é solicitado, essas opções são registradas no ticket.

Figura 21 - Formulário de abertura de chamado na Central de Ajuda

Enviar uma solicitação

Endereço de email *

Qual é o seu nome? :) *

Sobre o que se trata seu chamado? *

Qual o impacto do problema para o seu negócio?

Qual o ID (identificador) da sua conta?

Sobre qual Funcionalidade se trata seu chamado? *


Assunto

Descrição *

Neste campo você deve escrever sua solicitação. Busque trazer o máximo de informações possíveis sobre sua dúvida ou problema! Recomendamos anexar imagens da tela, pois isso nos ajudará a entender sua questão de forma mais assertiva :)

Anexos

Adicione o arquivo ou arraste até aqui

protegido por reCAPTCHA 

Privacidade - Termos

Fonte: <https://ajuda.rdstation.com.br/hc/pt-br/requests/new>

Essa imagem mostra como o cliente vê o formulário de abertura de chamados. Ao clicar na lista suspensa destacada, as quatro opções de urgência são apresentadas e, ao clicar em uma, ela se torna selecionada.

4.4.2 Formulário de Priorização

O Formulário de Priorização é aplicado assim que um ticket chega à fila de atendimento em N1. Ele apresenta as categorias de Impacto que o agente deve classificar e um campo para confirmar que essa categorização foi feita. Finalmente, o agente pode, também, ver qual urgência foi preenchida pelo cliente para o chamado. Feita a classificação, basta marcar o campo “revisei a prioridade” e atualizar o status do ticket, e o Controller fica responsável pelas demais etapas do processo de priorização.

Figura 22 - Formulário de priorização

The image shows a CRM interface with a priority form on the left and a chat window on the right. The form is titled 'Formulário' and contains the following elements:

- Formulário:** A dropdown menu with the text '[Mega] Formulário para priorização ...'.
- Tags:** A list of tags including 'active', 'formulario_central_de_ajuda', 'funcionalidade_base_de_leads_dados_leads_leads', 'is_partner', 'sobre_rd_station', and 'você_tem_uma_dúvida_sobre_como_fazer_algo'.
- Impact Categories (highlighted with a red box):**
 - [Mega] Está impedindo o cliente de interagir com os leads?
 - [Mega] O lead do cliente está impedido de seguir o fluxo de nutrição?
 - [Mega] Não consegue acessar o RD Station?
 - [Mega] Afetou a imagem do cliente?
 - [Mega] Urgência
 - Dropdown menu: 'Tenho uma dúvida sobre como fazer ...'
- Revisei a Prioridade:** A checkbox labeled '[Mega] Revisei a Prioridade'.
- Prioridade*:** A dropdown menu with the value '-'.
- Habilidades:** A dropdown menu.

The chat window on the right shows a customer message with ID: 16645 | Desengajados, dated Sexta-feira 19:48. The message text is partially redacted. Below the message is a public response field with a 'Resposta pública' tab selected. At the bottom, there is a 'Conversas' dropdown and a 'Tudo' indicator with a notification icon. A 'Aplicar macro' dropdown is visible at the very bottom.

Fonte: Arquivo Pessoal

A imagem destaca os campos do formulário usado pelo agente para categorizar a prioridade do ticket. Após marcar o campo de revisão da prioridade e atualizar o chamado, a prioridade era atribuída em alguns instantes, isso era a confirmação que o agente verificava para garantir que o sistema estava operacional. A urgência não era editada pelo agente, ela estava disponível apenas para o agente saber se a prioridade atribuída estava condizente com os valores de Impacto e Urgência do ticket.

5 ANÁLISE E RESULTADOS

Todas as etapas do projeto foram concluídas dentro do prazo e o sistema final com o funcionamento do NLC pôde ser implementado, permitindo que os chamados pudessem ser priorizados de forma automática. Nesta seção, será analisada a influência do projeto no processo de atendimento, buscando verificar a adesão da solução aos objetivos descritos no início do projeto.

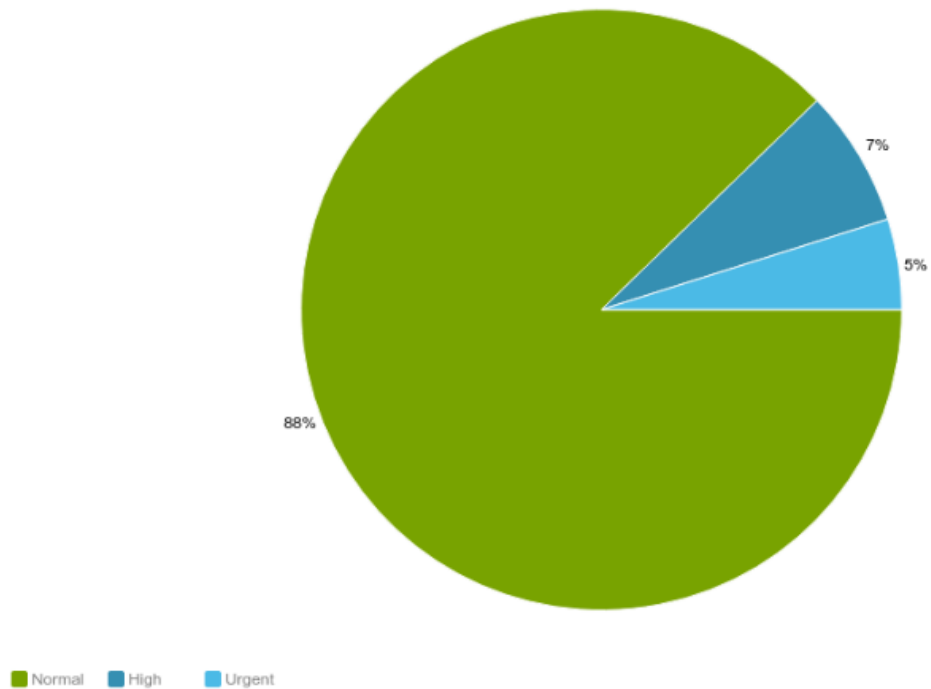
Todas as análises dessa seção foram feitas a partir de informações extraídos diretamente do Good Data, ferramenta que permite analisar e extrair dados de atendimento diretamente do Zendesk, portanto mantendo uma alta confiabilidade nos números.

5.1 Impacto na operação

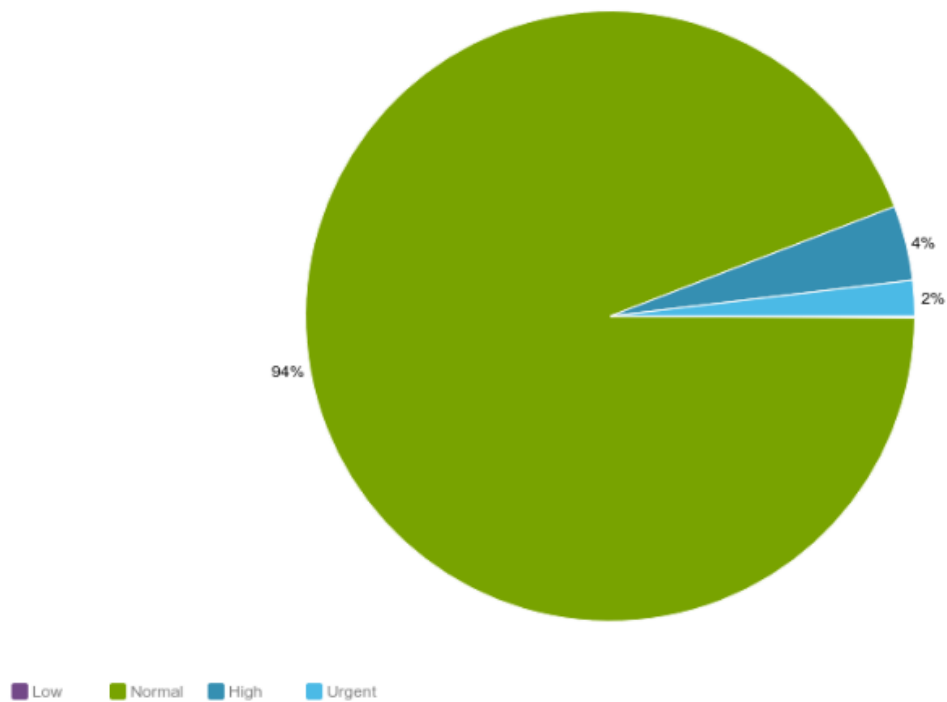
Uma das preocupações do projeto era, inicialmente, que não se afetasse o fluxo de atendimento normal, ou seja, que não fossem exigidos tempos de resposta que os agentes não conseguissem cumprir por conta de SLAs mais curtos de prioridades mais altas. Isso não ocorreu, possivelmente pelo fato de que, ao se ter uma regra mais clara de priorização, menos chamados passaram a ser categorizados com prioridade mais alta. Os gráficos a seguir comparam os chamados priorizados no mês de Setembro (antes do processo de priorização) com os de início de Outubro até início de Novembro (um mês de operação do novo processo de prioridade). Como é possível observar, o percentual de chamados com prioridade alta e urgente diminuiu.

Figura 23 - Percentual de chamados priorizados antes e depois

Percentual de prioridades nos tickets antes do projeto



Percentual de prioridades nos tickets durante o projeto



Fonte: Arquivo Pessoal

Outra preocupação das lideranças do Suporte era o comportamento do cliente em relação ao campo de urgência do chamado. Ao contrário do que se esperava, o cliente soube ponderar bem suas demandas e não preencheu apenas as categorias de maiores urgências. Possivelmente, isso ocorreu pelo fato de o campo de urgência ser apresentado para o cliente sob o nome de “Qual o impacto do problema para o seu negócio?”, levando-o a refletir sobre qual seria, de fato, sua necessidade. A seguir se trazem dados referentes ao atributo de urgência, essa análise deixa evidente que o impacto do projeto foi diferente do esperado.

Valores de urgência	
[Mega] Urgência	# Solved Tickets
Diversos recursos pararam de funcionar	15
Tenho uma dúvida sobre como fazer algo	576
Um recurso importante do seu serviço parou de funcionar	104
Um recurso não está funcionando como deveria e está atrasando o trabalho	369

Entretanto, ficou claro que a operação de priorizar os tickets manualmente consome um tempo produtivo razoável do agente. Em média, foram priorizados 65 tickets por dia, e cada agente levou em torno de 1 minuto para priorizar o chamado. Como havia três agentes na operação de priorização, cada um teve uma perda de produtividade de aproximadamente 22 minutos ou 4,5% do útil de trabalho para cada agente.

Tickets priorizados por dia	
Day of Year (Ticket Solved)	# Solved Tickets
D296	27
D297	54
D298	83
D299	69
D302	75
D303	93
D304	94
D305	112
D309	37
D310	28
D316	31
D317	71
D318	89
D320	43
D323	66
Average	65

5.2 Análise entre processo manual e automático

O processo manual conseguiu atender o objetivo de priorizar os tickets de forma padrão. Contudo, era lento e custoso sob ponto de vista da operação do Suporte, o que pode ser verificado por meio da observação do custo em termos de tempo de atendimento apresentado na seção anterior. Além dos tickets priorizados durante o dia, há tickets gerados fora do horário comercial que se acumulam para serem priorizados no dia seguinte, o que causava certa lentidão na operação de atendimento no início da manhã e aumentava o tempo para priorizar os chamados. Cerca de 17 tickets ficavam pendentes para serem priorizados entre um dia e outro. Este número foi reduzido a zero com a implantação do NLC, devido ao fato de a verificação de prioridade permanecer sempre ativa.

Outra vantagem importante em relação ao processo manual relaciona-se ao tempo de demora para um chamado ser priorizado. Durante a categorização dos agentes, o tempo médio durante o qual um chamado permanecia sem prioridade era de 59,7 minutos. Isso inviabilizava completamente qualquer processo de atendimento com extrema urgência, já que o SLA de primeira resposta da prioridade Urgente era de 2 horas e, portanto, o agente só teria metade do seu tempo real para

investigar a demanda. Ao se implementar o NLC, a urgência foi resolvida, pois o tempo médio de execução do Zap de priorização é de 3 segundos e pode-se dizer que o chamado é priorizado instantaneamente.

5.3 Cálculo de ROI

O cálculo de ROI é fundamental para verificar a viabilidade de qualquer projeto. Ele consiste em avaliar se o ganho do que se está projetando supera seus custos e, assim, o retorno será positivo. O ganho no ROI inclui também economias com custos que deixam de existir. Como o custo por hora de um agente [17] é razoavelmente alto em comparação ao valor pago ao NLC da Watson, o retorno do projeto implementado é bastante positivo. A tabela a seguir analisa detalhadamente todo o retorno previsto para o projeto.

Ganho		
Média salarial de um agente	2.342,51	reais
Tempo de agente necessário para a priorização por dia	62	minutos
Tempo de operação de um agente no dia	540	minutos
dias uteis trabalhados pelos agentes	22	dias
Custo de um agente por dia	106,47	reais
Custo da priorização do agente por dia	12,22	reais
Custo do agente no processo/mês (Ganho total)	268,95	reais
Investimento		
Custo por requisição de API	0,0035	dólares
Cotação atual do dólar	3,76	reais
Requisições Gratuitas por mês oferecidas	1000	requisições
Requisições médias feitas pelo Suporte por mês	2000	requisições
Custo do total de requisição/mês (Custo total)	13,16	
Retorno	R\$ 255,79	por mês

O dado referente à demanda média atendida por mês foi extraída de valores no histórico da operação de atendimento, esses valores foram acessados na ferramenta Good Data que opera em paralelo ao Zendesk.

6 CONSIDERAÇÕES FINAIS E PERSPECTIVAS

6.1 Aprendizados

O projeto conseguiu cumprir sua função de priorizar os chamados do Suporte de forma automática. Além disso, a solução desenhada manteve-se simples, o que permite que sua manutenção seja feita até mesmo por pessoas de áreas menos técnicas que a engenharia. Sob essa perspectiva, a solução trouxe ótimo resultado para a empresa, por reduzir custos operacionais com um sistema que deve gerar poucos gastos no futuro, tanto de funcionamento quanto de governança.

Planejar um projeto tão amplo foi de fato desafiador. As diversas etapas que precederam a solução automática incluíram uma complexidade que inicialmente foi subestimada. Como as definições eram relativamente simples e focadas somente na Matriz de Prioridade, imaginou-se que a execução do processo manual não envolveria uma complexidade técnica grande. Entretanto, pensar um processo que fosse minimamente intrusivo à operação de atendimento e que fosse ao máximo a prova de provas não foi uma tarefa simples. Qualquer solução que é desenvolvida para entrar em produção em uma operação de larga escala sempre trará diversos desafios.

Outro aprendizado importante no projeto foi o de focar no valor que deve ser entregue, buscando sempre simplificar ao máximo a solução em termos de tecnologias usadas e processos. Um exemplo claro disso foi o uso de Inteligência Artificial no projeto. Inicialmente, o planejamento era desenvolver um algoritmo próprio que fosse capaz de analisar os textos por meio das técnicas de IA aprendidas ao longo da graduação. Graças a algumas reuniões com o líder técnico da área, ficou claro que essa escolha não estava sendo feita focando-se na entrega de valor, mas sim no quão profunda tecnicamente a solução seria. O aprendizado principal foi que boas soluções técnicas não necessariamente demandam extensas linhas de código e desenvolvimento exaustivo.

6.2 Limitações

Uma limitação clara da solução foi a precisão do algoritmo. Fazendo-se algumas análises de pequenas amostras de chamados, identificou-se uma taxa de acerto de apenas 72% por cento na categoria mais predominante na base de treinamento. Os demais 28% tickets restantes foram classificados automaticamente como prioridade “normal”, mas dentre eles poderia ter clientes precisando de atendimento urgente e a solução prevista não o contemplaria.

Entretanto, o projeto ainda possui espaço para diversas melhorias, principalmente no que se refere à precisão do algoritmo. Ao se ter mais dados, será possível analisar em detalhes a taxa de acerto e ações que podem ser feitas para melhorá-la. No entanto, a solução trouxe a vantagem de permitir um treinamento constante do NLC, permitindo que o sistema evolua ao se inserir novos dados no algoritmo.

Outro fator de atenção é a divergência de opinião dos agentes quanto a uma prioridade atribuída a um chamado, justamente por ainda ser impreciso, pode se criar um sentimento de insegurança com o sistema e esses agentes podem voltar a classificar os tickets manualmente. Para evitar que isso ocorra a operação deve ter clareza que o algoritmo ainda está em evolução e que possivelmente levará um tempo até que ele seja idealmente preciso. Em paralelo a isso deve se fazer um acompanhamento próximo de agentes de referência na operação para bolar soluções de contorno em casos mais críticos de classificação inadequada.

6.3 Planos futuros

Fato recorrente na Resultados Digitais são as constantes mudanças da empresa, sejam elas específicas ao contexto de Suporte ou não, esse ano por exemplo a empresa lançou um novo produto com uma operação significativamente diferente do anterior. Nesse novo cenário, toda a análise de categorias de impacto precisará ser refeita e, possivelmente, ocorrerão mudanças na estrutura dos Zaps. Contudo, ainda assim, a arquitetura do sistema como um todo continuará a mesma.

Além do contexto do novo produto, se deve ter uma especial atenção com outras mudanças no cenário de atendimento, uma expansão internacional em larga escala pode exigir NLCs em múltiplos idiomas, talvez até mesmo as categorias prioritárias podem mudar nesse contexto. O fato é que como qualquer empresa que cresce em ritmo acelerado, os sistemas internos exigem manutenção constante e eventualmente precisam ser atualizados. É importante que essa manutenção e atualização se aplique também ao sistema desenvolvido no projeto.

REFERÊNCIAS

- [1] Zendesk. Base de Conhecimento. Disponível em: <<https://support.zendesk.com/hc/pt-br/articles/203663536-Principais-conceitos-de-introdu%C3%A7%C3%A3o-ao-Zendesk>>. Acesso em: 16 nov. 2018.
- [2] Help Scout. Conteúdo. Disponível em: <<https://www.helpscout.net/helpu/support-management/>>. Acesso em: 16 nov. 2018.
- [3] Medium. Blog. Disponível em: <<https://medium.com/@Veamly/how-to-prioritize-your-support-tickets-34ebc716bdde>>. Acesso em: 16 nov. 2018.
- [4] Ameyo. Blog. Disponível em: <<https://www.ameyo.com/blog/how-to-delight-customers-streamline-helpdesk-support-with-smart-ticketing>>. Acesso em: 16 nov. 2018.
- [5] Vivantio. Blog. Disponível em: <<https://www.vivantio.com/service-management-the-problem-with-ticket-prioritization/>>. Acesso em: 16 nov. 2018.
- [6] Diferencial TI. Blog. Disponível em: <<https://blog.diferencialti.com.br/como-organizar-o-sla-da-sua-ti/>>. Acesso em: 16 nov. 2018.
- [7] Tech Republic. Article. Disponível em: <<https://www.techrepublic.com/article/how-to-systematically-prioritize-your-support-calls/>>. Acesso em: 16 nov. 2018.
- [8] IGTI. Blog. Disponível em: <<http://igti.com.br/blog/investimento-em-inteligencia-artificial/>>. Acesso em: 16 nov. 2018.
- [9] Nvidia. Blog. Disponível em: <<https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>>. Acesso em: 16 nov. 2018.
- [10] SAS. Article. Disponível em: <https://www.sas.com/pt_br/insights/analytics/deep-learning.html>. Acesso em: 16 nov. 2018.
- [11] Towards Data Science. Article. Disponível em: <<https://towardsdatascience.com/an-easy-introduction-to-natural-language-processing-b1e2801291c1>>. Acesso em: 16 nov. 2018.
- [12] Oxford Academic. Article. Disponível em: <<https://academic.oup.com/jamia/article/18/5/544/829676>>. Acesso em: 16 nov. 2018.
- [13] Algorithmia. Blog. Disponível em: <<https://blog.algorithmia.com/introduction-natural-language-processing-nlp/>>. Acesso em: 16 nov. 2018.

- [14] Resultados Digitais. Blog. Disponível em: <<https://resultadosdigitais.com.br/blog/infografico-exclusivo-como-funciona-o-marketing-digital-de-resultados/>>. Acesso em: 16 nov. 2018.
- [15] Dev Media. Article. Disponível em: <<https://www.devmedia.com.br/desenvolvimento-em-3-camadas-conceitos/22277>>. Acesso em: 16 nov. 2018.
- [16] Canal Tech. Article. Disponível em: <<https://canaltech.com.br/software/o-que-e-api/>>. Acesso em: 16 nov. 2018.
- [17] Catho Empresas. Tabela Salarial. Disponível em: <https://www.catho.com.br/salario/action/site/area_geral.php?state=salario&id_area_especifica=375&id_area_geral=60&id_cargo=3172>. Acesso em: 16 nov. 2018.
- [18] Unipampa. Documento. Disponível em: <https://dtic.unipampa.edu.br/files/2016/07/GLPI-GAUCHA_-_Matriz_de_Prioriza%C3%A7%C3%A3o.pdf>. Acesso em: 16 nov. 2018.
- [19] Agile Manifesto. Disponível em: <<https://agilemanifesto.org/>>. Acesso em: 24 nov. 2018.
- [20] LinkedIn. Article. Disponível em: <<https://www.linkedin.com/pulse/scrum-framework-ou-metodologia-rafael-viana-mba-psm/>>. Acesso em: 24 nov. 2018.
- [21] DZone. Article. Disponível em: <<https://dzone.com/refcardz/scrum?chapter=1>>. Acesso em: 24 nov. 2018.
- [22] Aquarela. Article. Disponível em: <<https://www.aquare.la/o-que-e-inteligencia-artificial/>>. Acesso em: 24 nov. 2018.
- [23] CIO. Article. Disponível em: <<http://cio.com.br/tecnologia/2018/06/13/a-inteligencia-artificial-esta-remodelando-as-operacoes-de-ti/>>. Acesso em: 24 nov. 2018.
- [24] Zendesk. Documentação técnica. Disponível em: <https://developer.zendesk.com/rest_api/docs/support/introduction>. Acesso em: 24 nov. 2018.