



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Henrique Costa e Silva

**Solução para classificação de contas e seleção de gerentes de atendimento
para serviço de Customer Success**

Florianópolis
2021

Henrique Costa e Silva

**Solução para classificação de contas e seleção de gerentes de atendimento
para serviço de Customer Success**

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Ricardo José Rabelo, Dr.

Supervisora: Paola Andrea Câncio Dias

Florianópolis

2021

Ficha de identificação da obra

A ficha de identificação é elaborada pelo próprio autor.

Orientações em:

<http://portalbu.ufsc.br/ficha>

Henrique Costa e Silva

**Solução para classificação de contas e seleção de gerentes de atendimento
para serviço de Customer Success**

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 22 de fevereiro de 2021.

Banca Examinadora:

Prof. Ricardo José Rabelo, Dr.
Orientador no curso
UFSC/CTC/DAS

Paola Andrea Cândia Dias
Supervisora na Empresa
RD GESTAO E SISTEMAS S.A.

Prof. Rodrigo Castelan Carlson, Dr.
Avaliador
UFSC/CTC/DAS

Prof. Fabio Baldissera, Dr.
Presidente da Banca
UFSC/CTC/DAS

RESUMO

O projeto descrito neste documento possui como objetivo desenvolver uma prova de conceito para um processo de roteamento de clientes que devem atendidos por uma estrutura de Customer Success em uma empresa do modelo SaaS. Tal proposta se mostra relevante para a empresa em que este projeto foi desenvolvido, e para aquelas que implementam uma área de Customer Success em geral, pois a assertividade na seleção de atendimento é um fator crucial para retenção e aumento de receita de clientes, além de ser primordial para manter uma estrutura de atendimento financeiramente estável. O processo original de roteamento também é demasiadamente oneroso para a área responsável por executá-lo e suscetível a erros humanos, dadas as diversas etapas manuais da execução. A metodologia de desenvolvimento do projeto adapta métodos iterativos e incrementais e a metodologia ágil Scrum. A solução consiste de uma rotina automática que utiliza de um *script* em linguagem Python para acessar dados armazenados na nuvem (serviço Amazon S3), tratá-los e calcular critérios relevantes para realizar a seleção automática de atendente mais adequado para cada cliente, com base em parâmetros definidos pela estratégia da empresa. Após a seleção ser realizada, outro *script* em Python – acionado automaticamente pelo código anterior – realiza o registro da relação Cliente-Atendente em um *webservice* interno da empresa, utilizando de uma ferramenta de automação de testes para navegadores web que emula as ações que deveriam ser feitas manualmente e retorna um histórico de registros realizados. Finalmente, parte dos dados consolidados pelo primeiro *script* são utilizados para construir *dashboards* operacionais voltados para o corpo de líderes da área de atendimento em questão, fornecendo informações antes custosas de serem consolidadas. Ao final do projeto, a rotina original que leva cerca de 4 a 5 horas semanais para ser executada agora requer apenas uma média de 90 segundos semanais e roteia clientes para seus gerentes com perfeita aderência aos parâmetros em acordo com a estratégia e política de atendimento da empresa.

Palavras-chave: Customer Success. Atendimento de clientes. SaaS. Python. Amazon S3.

ABSTRACT

This document describes a project that proposes proof-of-concept for the process of routing clients that need to be supported by a Customer Success area in a SaaS company. This proposal is relevant not only for the company in which this project was developed but also for any company that implements a Customer Success area, given that the accuracy of said process is trivial to maintain a financially healthy customer service structure and guarantee client's retention and revenue growth. The original process requires a significant amount of effort to be executed by the team responsible for it, being also highly susceptible to human error given the numerous manual tasks involved. This project's development methodology is adapted from interactive and incremental development strategies and the lean project management framework Scrum. The solution consists of an automatic routine that utilizes a Python script to remotely access a data repository in the cloud (Amazon S3), retrieve and format the information and calculate relevant criteria to automatically select the most adequate attendant for each customer, based on the company's customer service strategy. After the selection phase, another Python script – automatically triggered by the code before – registers the Customer-Attendant relation planned in the company's internal webservice, using a tool that automates browser tests in a programmed routine that emulates the manual actions that should be taken for this step and returns a history of registers made. Finally, a portion of the data from the first script is used to build operational dashboards aimed for the leadership team of the customer service area in question, providing data and predefined analysis that previously required considerable effort to be made available. At the end of the project, the original routing routine that requires 4 to 5 hours weekly to be executed now requires only 90 seconds weekly on average and offers perfect adherence to the company's strategic parameters and policies when allocating attendants to their respective customers.

Keywords: Customer Success. Customer service. SaaS. Python. Amazon S3.

LISTA DE FIGURAS

Figura 1 – Cenário de comportamento de receita recorrente mensal em caso de <i>net churn</i> positivo, de 3%.	16
Figura 2 – Cenário de comportamento de receita recorrente mensal em caso de <i>net churn</i> negativo, de -3%.	16
Figura 3 – 5 passos realizados em um processo de BI segundo TechTarget. . .	18
Figura 4 – Exemplo de <i>dashboard</i> construído na ferramenta Tableau.	19
Figura 5 – Diagramas UML da versão 2.0, com destaque para mudanças em relação à versão 1.0.	20
Figura 6 – Diagrama de sequência que ilustra o uso de um caixa-eletrônico. . .	21
Figura 7 – Exemplo de modelagem de um processo de envio de <i>hardware</i> por um empresa varejista em BPMN.	22
Figura 8 – Visão geral dos componentes da BPMN.	23
Figura 9 – Modelo cascata, com realimentação.	24
Figura 10 – Modelo iterativo e incremental.	24
Figura 11 – Exemplo de regra com capacidade de exportação automática para Amazon S3 pela ferramenta Gainsight.	31
Figura 12 – Digrama de Sequência da rotina automatizada planejada como solução e prova de conceito.	33
Figura 13 – Digrama de Classes representando as categorias macro de dados utilizados pela solução.	34
Figura 14 – Digrama de Implementação da solução proposta.	35
Figura 15 – Digrama de Caso de Uso de Classificação e seleção.	37
Figura 16 – Digrama de Caso de Uso de Registro de Atendimento.	38
Figura 17 – Estrutura de código para acesso ao Amazon S3 pelo SDK Boto3. . .	43
Figura 18 – Exemplo de armazenamento em formato <i>dataframe</i> e formatação de dados pela biblioteca Pandas.	44
Figura 19 – Imagem ilustrativa da função de classificação de clientes em <i>tiers</i> de atendimento.	45
Figura 20 – Imagem ilustrativa da seleção de clientes para roteamento.	46
Figura 21 – Imagem ilustrativa da função de MRR na carteira de um CSM. . . .	46
Figura 22 – Imagem ilustrativa da etapa de chamada de funções pelo método de vetorização.	47
Figura 23 – Imagem ilustrativa da função de seleção CSMs para clientes.	49
Figura 24 – Imagem ilustrativa da etapa de identificação de casos indevidos para o roteamento.	51
Figura 25 – Imagem ilustrativa da etapa de acesso ao portal de registro de gerentes.	53

Figura 26 – Imagem ilustrativa da importação e tratamento de dados na ferramenta Power BI.	54
Figura 27 – Imagem ilustrativa do modelo de <i>dashboards</i> operacionais desenvolvidos.	56
Figura 28 – Visualização disponibilizada para a liderança da área de CSI.	57
Figura 29 – Mapeamento do processo de roteamento de clientes em notação BPMN.	69
Figura 30 – Mapeamento do processo de roteamento de clientes proposto como solução em notação BPMN.	70

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BI	Business Intelligence
BPMN	Business Process Model and Notation
CRM	Customer Relationship Management
CS	Customer Success
CSI	Customer Success Inbound
CSM	Customer Success Manager
IS	Implementation Success
KPI	Key Performance Indicator
MRR	Monthly Reccuring Revenue
RD	RD Station
REST	Representational State Transfer
SaaS	Software as a service
SDK	Software Development Kit
UML	Unified Modeling Language

SUMÁRIO

1	INTRODUÇÃO	11
1.1	MOTIVAÇÃO	11
1.2	OBJETIVOS	12
1.3	ESTRUTURA DO DOCUMENTO	13
2	A EMPRESA	14
3	FUNDAMENTAÇÃO TEÓRICA	15
3.1	MODELO DE ATENDIMENTO DE CUSTOMER SUCCESS EM EMPRESAS SAAS	15
3.2	<i>BUSINESS INTELLIGENCE</i>	17
3.3	<i>DASHBOARDS</i>	18
3.4	<i>SCRIPTING</i>	19
3.5	UML	20
3.6	BPMN	22
3.7	METODOLOGIA ITERATIVA E INCREMENTAL	24
4	ESPECIFICAÇÃO DO PROJETO	26
4.1	PROCESSO ATUAL DE ROTEAMENTO	26
4.2	INFORMAÇÕES RELEVANTES PARA CLASSIFICAÇÃO DE <i>TIERS</i> DE CLIENTES	28
4.3	INFORMAÇÕES RELEVANTES PARA CLASSIFICAÇÃO DE CARTEIRAS DE CSMS	29
4.4	TECNOLOGIAS E FERRAMENTAS EXISTENTES	30
4.5	ESPECIFICAÇÃO DA SOLUÇÃO PROPOSTA	31
4.5.1	Aquisição de dados	35
4.5.2	Tratamento e enriquecimento de tabelas	35
4.5.3	Classificação de níveis de atendimento	36
4.5.4	Seleção de clientes para roteamento	36
4.5.5	Classificação de carteiras de CSMS	36
4.5.6	Seleção de CSMS para atendimento	37
4.5.7	Registros de CSMS em contas	38
4.5.8	<i>Dashboards</i> operacionais	38
4.5.9	Requisitos Funcionais	39
4.5.10	Requisitos Não-Funcionais	41
5	DESENVOLVIMENTO	43
5.1	ACESSO E BUSCA DE DADOS NO AMAZON S3 E TRATAMENTO DE DADOS	43
5.2	CLASSIFICAÇÃO DE CONTAS E CARTEIRAS DE CSMS	44
5.2.1	Classificação de clientes em <i>tiers</i> de atendimento	44

5.2.2	Identificação de contas para roteamento	45
5.2.3	Classificação de carteiras de CSMS	46
5.2.4	Utilização de funções e consolidação de tabelas	47
5.3	ETAPA DE SELEÇÃO GERENTE DE CONTA	47
5.3.1	Parâmetros para seleção de gerente de contas	48
5.3.2	Programação da etapa de seleção de CSM	49
5.4	ETAPA DE REGISTRO DE CSMS	50
5.4.1	Funcionamento geral do <i>script</i>	50
5.4.2	Problemas com o <i>script</i> original e soluções desenvolvidas	50
5.5	DESENVOLVIMENTO DOS <i>DASHBOARDS</i> OPERACIONAIS	54
6	TESTES E METODOLOGIA DE AVALIAÇÃO DO DESEMPENHO	58
6.1	ETAPA DE COLETA E FORMATAÇÃO DE DADOS	58
6.2	ETAPA DE SELEÇÃO DE CLIENTES PARA ROTEAMENTO	58
6.3	ETAPA DE CLASSIFICAÇÃO DE <i>TIERS</i> DE CLIENTES	59
6.4	ETAPA DE CLASSIFICAÇÃO DE CARTEIRAS DE CSMS	59
6.5	ETAPA DE SELEÇÃO DE CSMS	60
6.6	ETAPA DE REGISTRO DE CSMS EM CONTAS	60
6.7	AVALIAÇÃO DA SOLUÇÃO E <i>DASHBOARDS</i>	60
7	CONCLUSÃO	62
7.1	PROPOSTAS DE MELHORIAS	63
	REFERÊNCIAS	64
	ANEXO A – ROTEAMENTO ATUAL DE CLIENTES EM NOTAÇÃO BPMN	69
	ANEXO B – PROCESSO DE ROTEAMENTO DE CLIENTES PRO- POSTO EM NOTAÇÃO BPMN	70

1 INTRODUÇÃO

O mercado de *Software as a service (SaaS)* define um método de distribuição digital de seus produtos e baseia-se em um modelo de assinatura em que clientes pagam um determinado valor pela solução em frequências específicas de tempo (sejam elas mensais ou anuais). Este mercado popularizou a função de *Customer Success (CS)* nas empresas que implementam este método, que visa acompanhar o cliente ao longo de sua jornada com o *software*, o guiando e orientando a respeito do uso deste, garantindo a adoção da solução de forma a maximizar os ganhos obtidos com cada assinatura. No âmbito da RD Station (RD), empresa em que o projeto descrito neste documento foi realizado, esta maximização de ganhos pode ocorrer por meio de uma expansão nas características do produto ou simplesmente por um longa retenção do cliente.

Para garantir que os melhores resultados em ambas estas frentes sejam obtidos, o modelo atual de atendimento da RD rege que *Customer Success Manager (CSM)* (funcionários que atuam na área de CS) específicos sejam alocados a cada cliente de acordo com as características dos produtos obtidos por estes.

O projeto apresentado neste documento explora uma solução que serve como prova de conceito para um método automatizado de seleção de CSMs, visando alocar para cada cliente novo o gerente de contas mais adequado de acordo com o nível de atendimento do cliente (*tier*) e políticas internas da empresa. O escopo do projeto também prevê o desenvolvimento de *dashboards* gerenciais que visam entregar análises predefinidas e simplificadas para o corpo de gerentes da área de Customer Success Inbound (CSI), que é o foco da solução, almejando facilitar a tomada de decisão e diagnóstico a respeito da situação atual da área.

A respeito da metodologia de desenvolvimento, o projeto como um todo é dividido em etapas semanais utilizando de uma metodologia baseada em *sprints*, simplificando o método ágil de gestão de projetos Scrum (SUTHERLAND, 2015). Essa metodologia foi escolhida, principalmente, pois facilita a re-priorização de tarefas ao longo das semanas caso hajam atrasos ou imprevistos, além de favorecer uma cadência mais ágil e segmentada de desenvolvimento da solução, iterativamente testando e aprimorando o que foi desenvolvido.

1.1 MOTIVAÇÃO

O processo de seleção de CSMs e registro destes como responsáveis por cada cliente é comumente chamado de **roteamento** na RD, e é realizado atualmente por meio de um processo extremamente manual e custoso do ponto de vista operacional (como apresentado no capítulo 4). Devido ao baixo nível de automatização da atividade e quantidade de variáveis envolvidas, o processo requer cerca de 4 a 5 horas semanais

para ser executado. Neste período, uma média de 60 clientes de diferentes *tiers* são roteados para seus gerentes de contas, estando esta seleção altamente suscetível a erros humanos ou até mesmo técnicos devido às ferramentas e estratégias utilizadas (também exposto no capítulo 4 deste documento). Dado que os gerentes de contas são alocados em uma fase crucial para a adoção à ferramenta, a escolha equivocada de um CSM tem impacto direto na experiência inicial do cliente, o que por sua vez impacta diretamente na retenção deste na base. Mais imediatamente, há impacto em métricas internas de relevante importância para o modelo de atendimento da empresa que servem como indicadores para evoluções de Monthly Recurring Revenue (MRR), ou receita mensal recorrente, e *churn* (termo utilizado para referenciar o cancelamento da assinatura).

O atual processo e sua característica onerosa também impede que o analista responsável por executá-lo possa desprender uma maior parte do seu tempo realizando atividades mais estratégicas e valiosas para a estrutura de negócios da empresa. O período semanal desprendido, que inclui desde o levantamento de dados de clientes que devem ser roteados até eventuais – e corriqueiras – ações corretivas, toma o espaço que poderia ser dedicado a análises que agregariam valor ao modelo de atendimento e evolução da área de CS e da RD como um todo.

1.2 OBJETIVOS

Como exposto na seção acima, o projeto compreende uma prova de conceito sobre um processo automatizado de roteamento de CSMs para clientes. O planejamento é utilizar da estrutura existente de tecnologias e informações para desenvolver uma rotina em linguagem de programação Python para a classificação de contas e gerentes e seleção destes para cada cliente, adaptando e evoluindo o que for necessário para tornar o processo mais eficiente e preciso. Então, almeja-se utilizar dos dados coletados e tratados para desenvolver *dashboards* operacionais que agreguem valor e facilitem a tomada de decisão do corpo de líderes da área de CSI. A linguagem de programação Python foi escolhida por oferecer poderosas bibliotecas de tratamentos de dados (em formatos nativo de tabelas principalmente) e por ser a linguagem de programação na qual os analistas da área responsável pelo roteamento são mais familiares, tornando o processo de manutenção e evolução da solução proposta mais democrático. Os objetivos gerais do projeto são:

1. Mapeamento do processo atual de roteamento de contas de novos cliente para a área de CSI em notação BPMN;
2. Mapeamento dos critérios utilizados para classificar os *tiers* de atendimento de clientes de CSI;

3. Mapeamento dos critérios utilizados para classificar as carteiras de clientes dos gerentes de contas de CSI;
4. Mapeamento das tecnologias e ferramentas que serão adaptadas para o desenvolvimento do projeto;
5. Programação e testes da etapa da rotina em Python que busca os dados necessários e realiza o tratamento e formatação destes;
6. Programação e testes da rotina em Python de identificação de contas que devem ser roteadas, classificação de *tiers* de clientes e carteiras de clientes de CSM;
7. Definição do método de seleção de CSMs para contas e programação e testes deste na rotina Python desenvolvida;
8. Adaptação e testes do *script* de registro de CSMs em contas no sistema interno da empresa desenvolvido em plataforma web;
9. Mapeamento das análises necessárias e seleção de dados relevantes para os *dashboards* operacionais;
10. Construção dos *dashboards* operacionais;

1.3 ESTRUTURA DO DOCUMENTO

Este documento está dividido em 7 capítulos. Nos capítulos 1 e 2 introduz-se o problema e sua relevância, contextualizando o projeto proposto como prova de conceito para uma solução possível, seu escopo e seus objetivos e apresenta-se a empresa em que o projeto foi desenvolvido. O capítulo 3 traz as bases teóricas para o desenvolvimento e no capítulo 4 são expostos os mapeamentos do processo atual, dados e tecnologias já empregadas no processo vigente que relevantes para a execução do projeto, especificação geral da solução proposta e os requisitos funcionais e não-funcionais do projeto.

No capítulo 5 discorre-se sobre a solução desenvolvida propriamente dita, dividida em tarefas como exposto pelos objetivos descritos na seção anterior. Por fim, o capítulo 6 analisa os resultados obtidos com a solução desenvolvida e no último capítulo destaca-se melhorias e evoluções possíveis para elaborações futuras.

2 A EMPRESA

A RD Station (RD, 2021), popularmente conhecida como Resultados Digitais ou simplesmente RD, é uma empresa do ecossistema tecnológico de Santa Catarina que atua no mercado de SaaS desde de 2011 com o desenvolvimento, distribuição e atendimento de um *software* de referência nacional em marketing digital, o RD Station Marketing (RDSM, 2021). O produto atualmente é o carro chefe do *stack* de soluções da empresa que passa por constante evolução e conta também com uma solução de gerenciamento de vendas, o RD Station CRM (RDSCRm, 2021). Apesar da maior parte da base de clientes da empresa se encontrar em território nacional, ela também se estende a diversos países ao redor do globo como Colômbia, México e Portugal.

Por ser uma empresa do modelo SaaS (KILLHAM, 2021), em que a aquisição e utilização do produto pelo cliente se dá de forma digital por meio de uma assinatura de pagamento recorrente (em frequência usualmente mensal), o emprego de uma área de Customer Success se mostra vantajoso na estratégia de retenção de clientes e expansão de receita recorrente recolhida destes. A área é composta por especialistas em estratégias de marketing digital (CSMs) que guiam o cliente na sua utilização do produto, unindo as funcionalidades do *software* aos objetivos de negócio do cliente em relação a geração de demanda, relacionamento com *leads* e clientes, análise de resultados de marketing e vendas em geral.

A empresa atua com dois canais de vendas para o RD Station Marketing: a área de vendas interna e por meio de agências de marketing parceiras da RD. O *software* de marketing hoje é oferecido em quatro diferentes planos (Light, Basic, Pro e Enterprise) e diferentes bases de contatos (*leads*) para cada um destes, sendo que clientes que adquirem os planos Pro ou Enterprise se classificam para atendimento vitalício de um CSM. Dadas certas características intrínsecas do contratante e plano(s) adquirido(s), a serem exploradas no capítulo seguinte, o atendimento de clientes é dividido em diferentes níveis (ou *tiers*). Tais *tiers* são compostos por CSMs predefinidos, e almejam proporcionar o atendimento mais adequado para o cliente com base nos atributos destes, respeitando diretrizes estratégicas e financeiras estabelecidas pelo modelo de negócio da empresa.

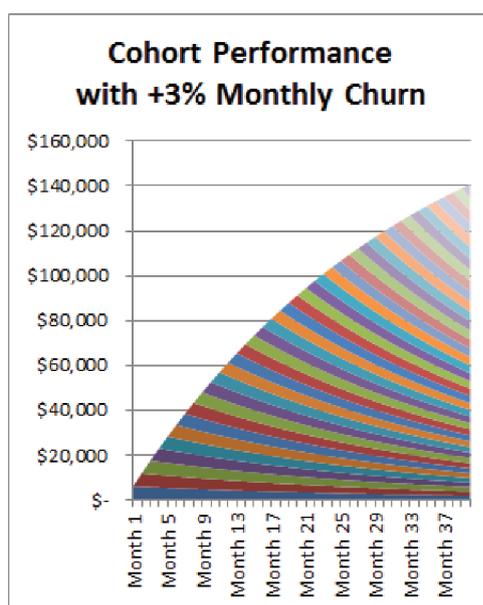
3 FUNDAMENTAÇÃO TEÓRICA

3.1 MODELO DE ATENDIMENTO DE CUSTOMER SUCCESS EM EMPRESAS SAAS

Apesar de estar intrinsecamente ligado às definições estratégicas de cada empresa que utiliza de uma área de Customer Success em sua estrutura, alguns princípios básicos devem ser respeitados para que uma equipe que exerça esta função seja considerada eficiente e financeiramente estável. Como mencionado anteriormente o principal objetivo de uma área de Customer Success, principalmente em uma empresa do modelo SaaS, é de garantir o sucesso e satisfação do cliente com o produto/serviço, almejando obter melhores resultados financeiros com cada assinatura. Estes critérios são atingidos pela adequação do atendimento às necessidades estratégicas de clientes para com o produto adquirido e pela retenção ou expansão de receita recolhida por cliente (DIGITAIS, 2021).

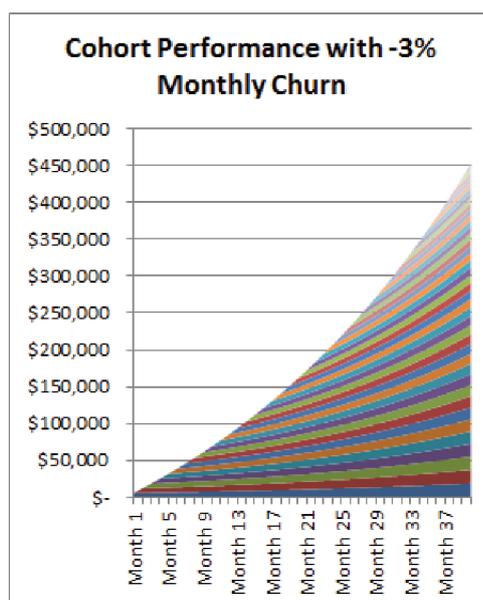
Se em modelos tradicionais de negócio os lucros são obtidos apenas pelas áreas de vendas, em empresas do modelo SaaS (ou aquelas que empregam uma área de Customer Success em geral) pode-se também gerar lucro com o atendimento estratégico de clientes. Isso se dá pois estratégias de atendimento e interações entre gerentes de sucesso – os CSMs – e clientes bem desenvolvidas alavancam os resultados que estes obtêm com o produto, promovendo a retenção de assinaturas e muitas vezes expansão de receita por *upgrades* em características do produto. Especificamente em companhias que atuam no modelo SaaS isso pode ser medido por algumas métricas consolidadas, sendo que as que mais se destacam entre elas são o *churn*, o cancelamento de assinaturas, e o *expansion*, que nada mais é do que o aumento efetivo de receita por cliente. Ambas essas métricas podem ser medidas do ponto de vista de receita (*revenue churn* e *revenue expansion*) ou do ponto de vista de clientes unitários (*logo churn* e *logo expansion*). Quando analisadas em conjunto essas métricas definem o indicador de *net churn*, ou “cancelamento líquido”, sendo que valores positivos de *net churn* podem limitar não apenas a receita adquirida por mês como também o crescimento mês-a-mês de empresas. Já valores negativos de *net churn* trazem crescimentos acelerados, promovendo avanços exponenciais ao longo dos meses. Juntas, essas métricas apontam fatores importantes sobre a saúde não somente da área de Customer Success como também da companhia como um todo, e possuem forte relação com as estratégias de atendimento empregadas (SKOK, 2021). As imagens abaixo ilustram cenários de *net churn* negativo ou positivo, considerando uma situação onde este indicador possui valores de 3% e -3% de *net revenue churn*, respectivamente, e a receita recolhida por mês é de \$6.000,00.

Figura 1 – Cenário de comportamento de receita recorrente mensal em caso de *net churn* positivo, de 3%.



Fonte – (SKOK, 2021)

Figura 2 – Cenário de comportamento de receita recorrente mensal em caso de *net churn* negativo, de -3%.



Fonte – (SKOK, 2021)

3.2 BUSINESS INTELLIGENCE

O conceito de Business Intelligence (BI), ou “inteligência de negócio” traduzido livremente para o português, pode ser definido como a utilização de tecnologias e métodos que arrecadam e armazenam dados e proporcionam gestão de conhecimento, em conjunto com formulações de análises que visam aprimorar o processo de tomada de decisão a respeito de ações relacionadas à estratégias de empresas (BURSTEIN, 2008). Ainda, a empresa global de consultoria e pesquisas Gartner define BI como um termo abrangente que inclui aplicações, infraestrutura e ferramentas e práticas consolidadas que promovem análises para aprimorar e otimizar o processo de tomada de decisão de empresas (ANALYTICS... , 2021).

Para que análises frutíferas e relevantes sejam montadas, é necessário possuir um bom entendimento do negócio da empresa e sua estratégia, de forma que seja possível formular indicadores e métricas significantes dentro do planejamento da companhia. Neste âmbito, diversas metodologias e práticas são empregadas, a mais consolidada entre elas talvez sendo os Key Performance Indicator (KPI). Estes são indicadores de desempenho formulados com base na análise profunda da estratégia da empresa para determinadas ações ou iniciativas, buscando resumir em métricas enxutas o que deve ser medido para que o desempenho das atividades seja avaliado, tornando possível aferir quando alguma ação corretiva ou evolutiva deve ser tomada (5... , 2021). Podem ser resumidos em algumas categorias, como indicadores de produtividade, qualidade, estratégicos, etc.

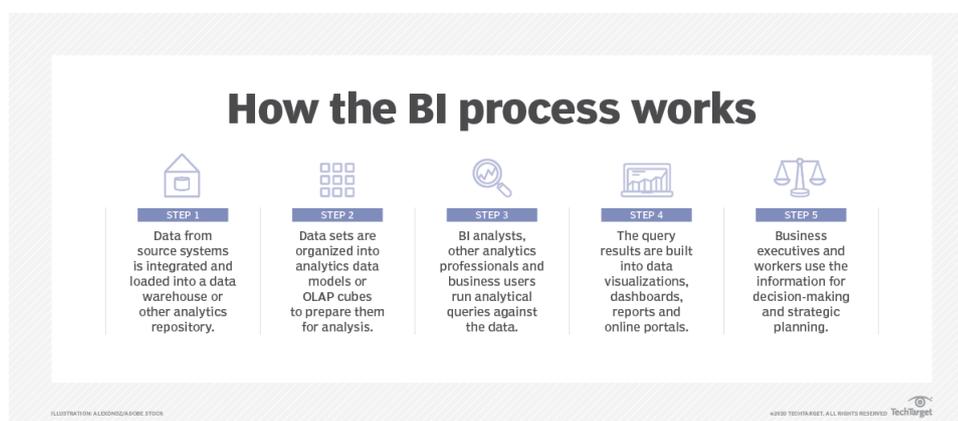
As etapas necessárias para formular análises dentro de um processo de BI incluem (STEDMAN, 2021a):

- Preparação de dados;
- Seleção e tratamento destes dados;
- Formulação e distribuição de KPIs;
- Uso das informações para influenciar e otimizar a tomada de decisão de negócios.

Dentre os tipos de BI existentes, destaca-se a *Operational intelligence*, ou “inteligência operacional” em tradução livre para o português, que é uma forma de análise em “tempo real” que fornece informações a respeito da linha de frente de atuação do negócio (STEDMAN, 2021b). Traçando um paralelo com empresas SaaS que implementam em sua estrutura uma área de Customer Success, análises de inteligência operacional desta área poderiam fornecer dados como quantidade de clientes sendo atendidos, receita gerenciada pela área, clientes com risco de cancelamento, entre outros.

A figura 3 abaixo resume em 5 passos o processo de BI padrão implementado por empresas:

Figura 3 – 5 passos realizados em um processo de BI segundo TechTarget.



Fonte – (STEDMAN, 2021a)

O conceito de BI permeia o desenvolvimento do projeto descrito neste documento dado que para desenvolvê-lo é necessário, em diversas etapas, alinhar as ações a serem tomadas com os objetivos do negócio em questão e os dados disponíveis para tomada de decisão (seja na etapa de escolha de CSMs e definição de nível de atendimento de clientes, ou ainda na etapa de elaboração de *dashboards* operacionais).

3.3 DASHBOARDS

Dashboards podem ser definidos como ferramentas de visualização de dados, construídos com o propósito de facilitar análises a respeito de diferentes tópicos. Estão intimamente relacionados com o tópico anterior de *business intelligence* pois, para que possam ser construídos, é necessário antes realizar um levantamento de dados e indicadores de relevância para o processo ou negócio sendo analisado. Podem ser divididos em diferentes categorias, como estratégicos, analíticos, informacionais ou ainda operacionais (FEW, 2006). Estes, por sua vez, visam fornecer aos analistas ou tomadores de decisão de uma determinada área métricas relevantes para a análise de desempenho da operação em questão, de forma que seja possível identificar riscos eminentes ou oportunidades de aprimoramentos. Se relacionam intimamente com o conceito de *operational intelligence* apresentado acima.

Para que possam ser construídos, pode-se apontar como componentes que se fazem necessários os seguintes (SUTNER, 2021):

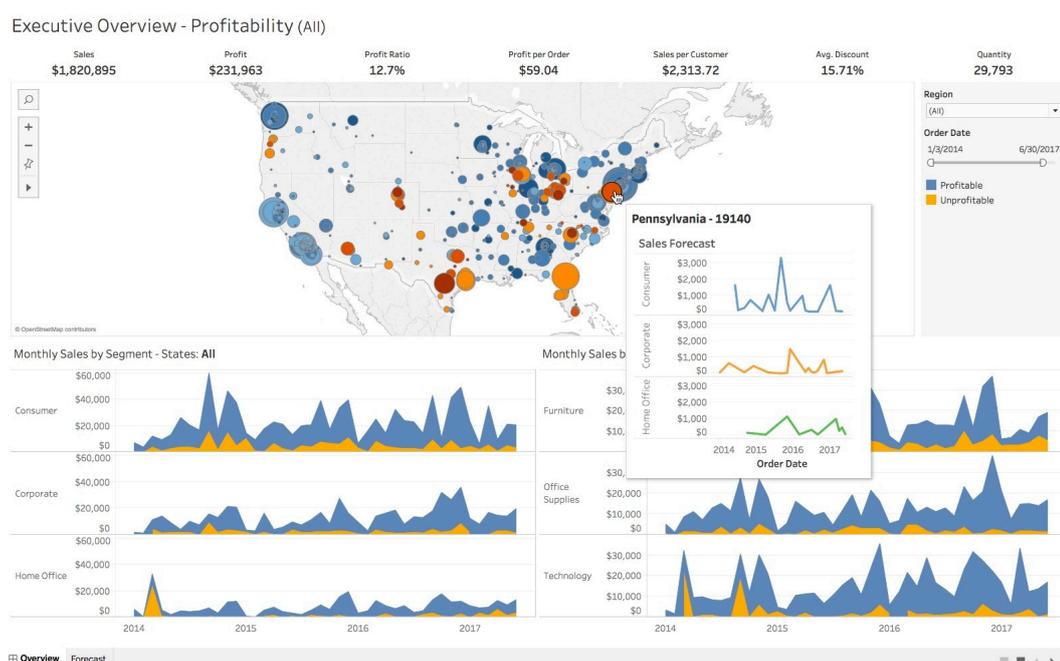
- Conexão à fontes de dados;

- Métodos de visualização de dados (tabelas, gráficos, etc.);
- Filtros de dados;
- Legendas e caixas de texto.

Ainda, alguns hábitos e metodologias podem ser definidos como boas práticas na construção de *dashboards*, de forma a maximizar suas vantagens. Dentre essas boas práticas pode-se destacar o envolvimento de *stakeholders* no desenvolvimento dos *dashboards*, utilizar de uma abordagem iterativa para formular os designs utilizados, identificar de forma precisa os dados que necessitam serem mostrados e não exagerar na quantidade de informações mostradas aos usuários (SUTNER, 2021).

Dashboards podem ser construídos em diversas ferramentas, como o amplamente disseminado Microsoft Excel (MICROSOFT..., 2021) ou ainda ferramentas específicas para visualização de dados com o Tableau (TABLEAU, 2021) (figura 4).

Figura 4 – Exemplo de *dashboard* construído na ferramenta Tableau.



Fonte – (TABLEAU, 2021)

3.4 SCRIPTING

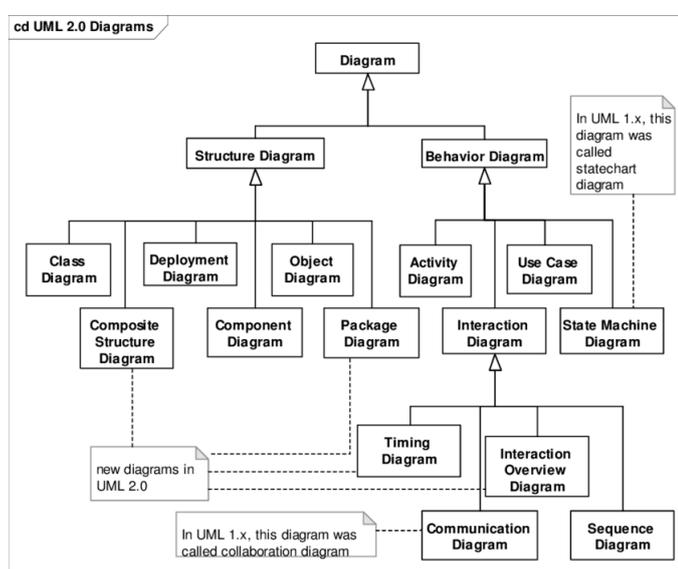
Um *script* é uma forma de programação que visa automatizar a execução de tarefas, que poderiam ser executadas por um agente humano, acessando funções de programas e aplicações e usualmente integrando diferentes sistemas (ECMAScript...,

2021). Linguagens de *script* se diferenciam de outras linguagens de programação por serem interpretadas, e não compiladas, o que significa que seu código é traduzido diretamente para linguagem de máquina linha por linha de forma sequencial sem a necessidade de ser compilado antes (WHAT... , 2021). Esse método de programação é utilizado neste projeto devido à natureza do processo a ser otimizado e a necessidade encontrada de automatizá-lo promovendo ganhos operacionais e assertividade e robustez de classificação de critérios.

3.5 UML

Unified Modeling Language (UML) é a linguagem padrão para modelagem de *softwares*, aplicada largamente no mercado para especificação de projetos e descrição de casos de uso de sistemas (UML, 2021). Em sua versão 2.0, a UML se divide em três categorias, que possuem em conjunto 13 tipos de diagramas. A visão geral dessa hierarquia é apresentada abaixo, na figura 5.

Figura 5 – Diagramas UML da versão 2.0, com destaque para mudanças em relação à versão 1.0.



Fonte – (NEW... , 2021)

Como apresenta a imagem, as três principais categorias de diagramas são (GROFFE, 2021):

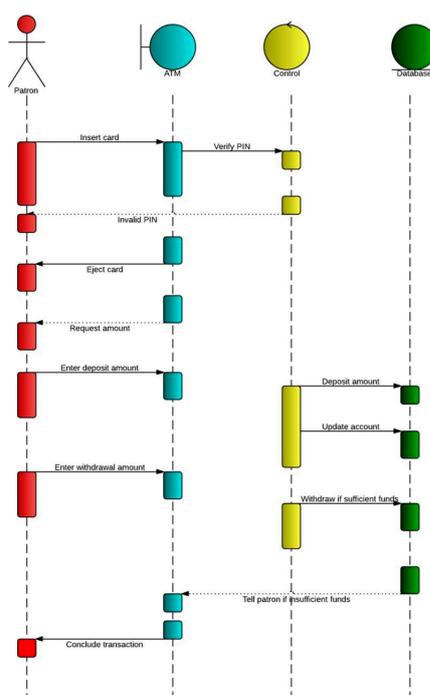
- Diagramas Estruturais: dividem-se em diagramas de classe, de objeto, de componentes, de implantação, de pacotes e de estrutura. Dentre estes, pode-se destacar o diagrama de classes, fundamental pois permite a visualização do

conjunto de classes envolvidas no sistema, com o detalhamento dos atributos (informações sobre as classes) e métodos (funções presentes em cada uma) além das relações estabelecidas entre elas. Outro diagrama notável é o de implantação, que ilustra a estrutura de *hardware* que é aplicada para a implantação do projeto. Nele constam componentes como bancos de dados, servidores, arquivos usados, etc.

- Diagramas Comportamentais: estes são divididos em diagramas de casos de uso, diagramas de atividade e diagramas de transição de estados. Dentre estes, o mais notável é o de casos de uso, que representa as atividades realizadas pelo sistema que interagem com atores externos e definem as ações macro realizadas pelo sistema, ou ainda os propósitos deste.
- Diagramas de Interação: por sua vez dividem-se em diagramas de sequência, de geral iteração, de comunicação e de tempo. Aqui o que mais se destaca é o diagrama de sequências, que apresenta as interações entre os diferentes objetos durante a execução do sistema planejado. Nele ficam claras as ordens em que as ações ocorrem, sendo ainda possível estimar qual objeto é mais utilizado.

A figura 6 ilustra um diagrama de sequência que descreve o processo de uso de um caixa-eletrônico.

Figura 6 – Diagrama de sequência que ilustra o uso de um caixa-eletrônico.

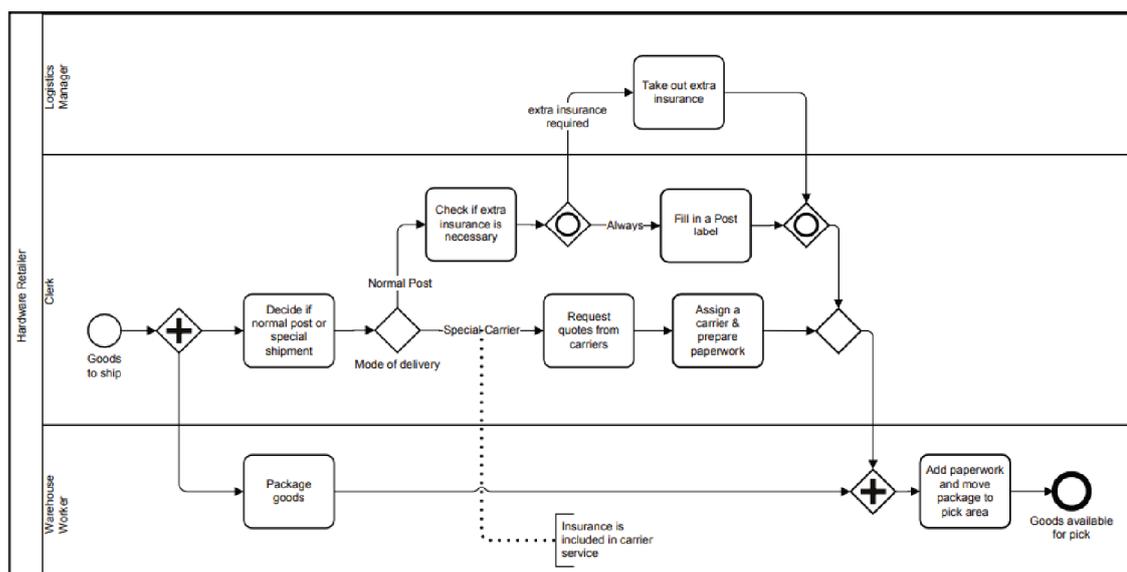


Fonte – (LUCIDCHART, 2021)

3.6 BPMN

A Business Process Model and Notation (BPMN) é amplamente difundida no mercado como o método padrão de modelagem e representação de processos de negócios, principalmente por ser um modelo altamente flexível e eficiente de descrever atividades e fluxos de informação entre etapas ou atores de forma que profissionais de diferentes funções possam obter o mesmo entendimento sobre o processo modelado. Ainda, oferece uma notação que pode facilmente ser traduzida em funcionalidades e linguagem de *software* (BPMN, 2021). DevMedia define BPMN como “uma notação padrão que representa processos de negócios por meio de diagramas de processos de negócio (*Business Process Diagram - BPD*)” e que “esta notação é orientada para uso humano, devido a fácil compreensão do diagrama formado, pois este faz uso de ícones padrões que facilitam o entendimento” (ARANTES, 2021). A figura abaixo ilustra um processo modelado em notação BPMN.

Figura 7 – Exemplo de modelagem de um processo de envio de *hardware* por um empresa varejista em BPMN.



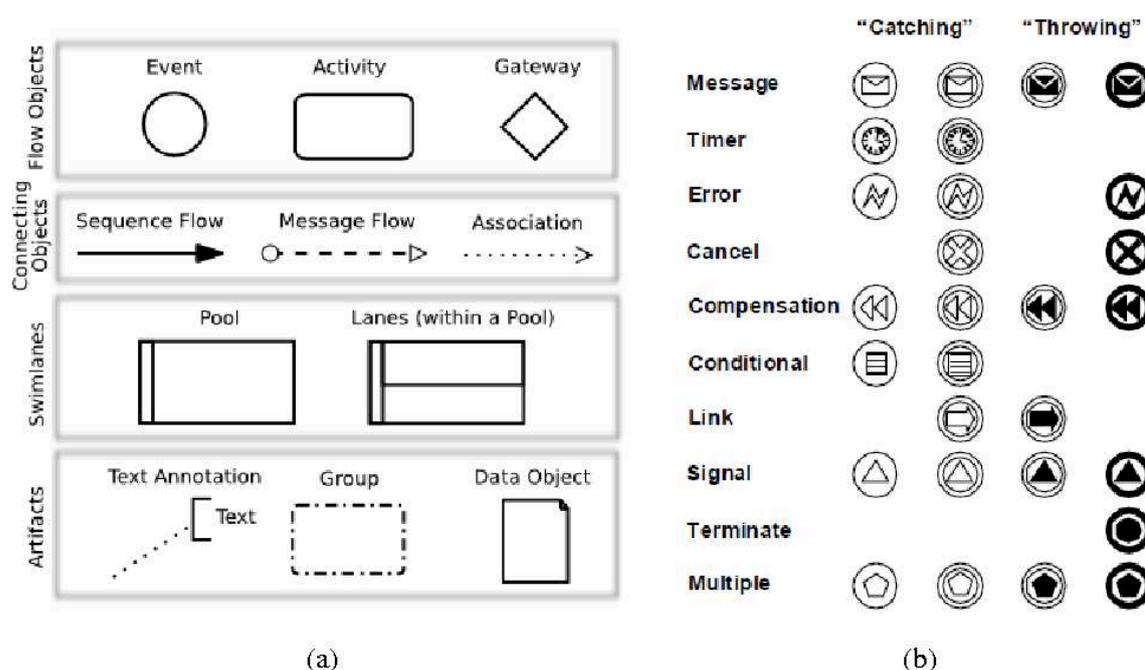
Fonte – (BPMN... , 2021)

Para modelar em BPMN, existem quatro grupos de elementos:

- Objetos de fluxo;
- Objetos de conexão;
- Raias;
- Artefatos.

Dentre estes, **objetos de fluxo** são os principais ao se definir o comportamento de um processo. Eles são divididos em eventos, que sinalizam um início, fim ou ainda fases intermediárias de um processo, atividades, que são os passos lógicos dentro do processo (sejam elas manuais, de serviço, *script*, etc.) ou decisões, que controlam etapas de divergência ou convergência da sequência de um fluxo. A figura 8 apresenta uma visão geral dos elementos da notação.

Figura 8 – Visão geral dos componentes da BPMN.



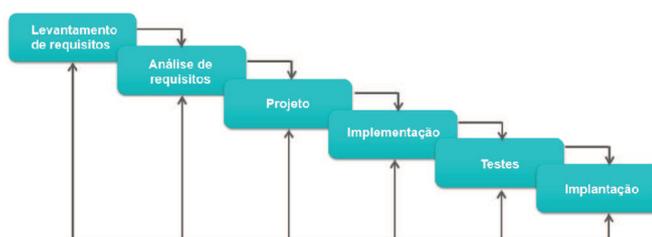
Fonte – (LYNCH, 2021)

A notação BPMN foi a escolhida para descrever os processos original e planejado de roteamento de CSMs para a área de CSI da RD no âmbito deste projeto.

3.7 METODOLOGIA ITERATIVA E INCREMENTAL

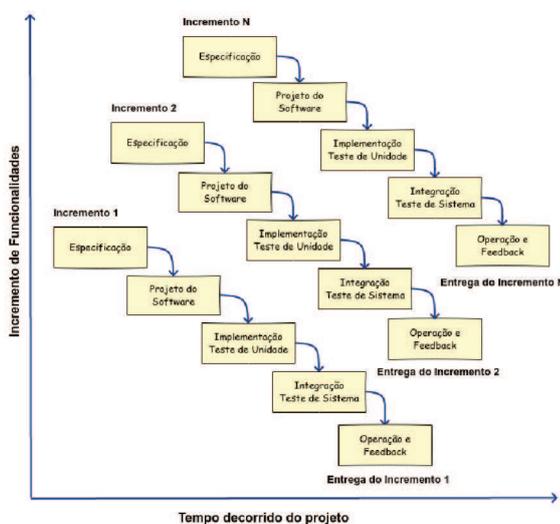
A metodologia de desenvolvimento deste projeto foi adaptada da metodologia iterativa e incremental de desenvolvimento de *softwares*, que rege que o projeto é realizado em etapas iterativamente testadas e utilizadas de forma modular já implementadas, evoluindo através de ciclos de desenvolvimento que avançam para atingir as necessidades especificadas (MEDEIROS, 2021b). Assim como no modelo de desenvolvimento em cascata, em que há uma “abordagem sequencial e sistemática para o desenvolvimento do *software*” (MEDEIROS, 2021a), o modelo incremental atua em seqüências lineares de desenvolvimento, porém é dividido em ciclos lineares menores de forma que após a finalização de cada um há a entrega de uma fração do sistema. Abaixo, ilustra-se os dois métodos mencionados, nas figuras 9 e 10.

Figura 9 – Modelo cascata, com realimentação.



Fonte – (ESTÁCIO, 2021)

Figura 10 – Modelo iterativo e incremental.



Fonte – (DIAS, 2021)

Se comunica intimamente com métodos ágeis de desenvolvimento e gestão de projetos, dentre eles pode-se destacar o Scrum, que também rege que o desenvolvimento de tarefas ou projetos deve ser realizado de forma evolutiva por meio de incrementos e testes, gerando refinamentos sucessivos (COHN, 2021). Unidos, esses métodos permitem que o projeto ou sistema seja entregue em etapas continuamente aprimoradas e integradas, agilizando a entrega de valor às partes envolvidas e aprimorando o conhecimento obtido sobre o funcionamento da solução em questão.

Esta abordagem foi selecionada pois o processo de roteamento possui uma grande interdependência entre suas etapas, sendo que cada uma destas pode ainda ser reutilizada para diferentes processos envolvidos na operação da área de atendimento de Customer Success Inbound. Abordando o projeto de forma incremental e iterativa, pode-se não somente “modularizar” cada etapa de forma que possam entregar valor ao processo de forma individual (automatizando etapas específicas como coleta e tratamento de dados), como também permite-se maior flexibilidade na implantação de melhorias necessárias ao longo do desenvolvimento.

4 ESPECIFICAÇÃO DO PROJETO

Neste capítulo são apresentados mapeamentos feitos para a elaboração do projeto, que serviram como base para o desenho da solução proposta como prova de conceito. Inicia-se apresentando a modelagem do processo original de roteamento de clientes, em notação BPMN. Em seguida são apresentadas as informações que regem a classificação de clientes em seus devidos *tiers* de atendimento e as informações que auxiliam a aferir a situação das carteiras individuais de cada CSM e, por consequência, da área de CSI como um todo, o que será de suma importância para a elaboração dos *dashboards* operacionais desenvolvidos no projeto. Após isso é realizada uma análise das tecnologias e ferramentas já empregadas pelo processo original, que serão utilizadas e adaptadas para desenvolver a prova de conceito da solução proposta. Finaliza-se então com a especificação da solução proposta, ilustrada também por meio de notação BPMN e por diagramas UML, assim como seus requisitos funcionais e não-funcionais.

4.1 PROCESSO ATUAL DE ROTEAMENTO

O processo original de roteamento em sua notação BPMN é apresentado no ANEXO A deste documento. Ele foi elaborado após analisar o processo vigente dividindo-o em atividades executadas pelo analista responsável por ele, enfatizando quais sistemas e fluxos de informações são envolvidos para que o roteamento de contas ocorra. A notação BPMN foi a escolhida dada sua consolidação no mercado como forma padrão de representação de processos de negócio, principalmente por possuir um design intuitivo e preciso o bastante para descrever uma rotina de forma que esta possa ser entendida facilmente por diferentes perfis de profissionais e também traduzida para *softwares* sem grandes dificuldades. O *software* de modelagem escolhido foi o Bizagi (BIZAGI, 2021) pela sua notoriedade no mercado para fins como este devido à usabilidade intuitiva e grande gama de funcionalidades ligadas à modelagem BPMN.

Antes de descrever o processo, é necessário destacar os dois principais critérios que definem quando clientes novos são roteados para a área de atendimento de CSI. São eles (ambos identificados durante o processo mapeado):

1. Aquisição do produto RD Station Marketing nos seus planos Pro ou Enterprise, via área de vendas interna da empresa, sem a contratação um serviço de implementação;
2. Caso tenham adquirido serviço de implementação, finalização de uma quantidade de interações com a área de Sucesso de Implementação – Implementation Success (IS) – predefinida pela estratégia de atendimento da empresa.

Uma vez que um cliente possui um CSM alocado para seu atendimento esta relação se mantém até que o cliente cancele ou mude seu plano de RD Station Marketing para Basic ou Light, ou seja roteado para outro gerente de contas após ter seu nível de atendimento alterado (processo realizado com menor frequência).

Conforme documentado, o roteamento se inicia no acesso à ferramenta de planilhas online Google Sheets, onde a primeira tarefa é a de atualizar os dados de clientes que adquiriram o produto RD Station Marketing sem o serviço de implementação por meio de uma integração nativa entre a ferramenta Google Sheets e o sistema de vendas da RD. Em seguida, na ferramenta de Customer Relationship Management (CRM) da área de CS, são registrados os grupos de contas de clientes com base nas novas vendas. São definidos como “grupos” as contas individuais que pertencem a um mesmo cliente ou entidade contratante em geral.

Ainda na ferramenta de CRM, é baixado em formato CSV um relatório preestabelecido que contém os dados gerais de clientes da área de CS como um todo e possui mais de 200 mil células. Este então é importado à ferramenta de planilhas, atualizando dados já existentes dessa natureza. Após carregado, é necessário realizar a limpeza de dados em casos de contas inválidas, processo realizado manualmente.

No CRM também é processado e baixado o relatório que traz as informações referentes às interações de clientes com a área de IS para aqueles que adquiriram o serviço de implementação. Este *dataset* por sua vez possui cerca de 300 mil células a serem atualizadas e analisadas (por meio de fórmulas) na ferramenta de planilhas. Com estas atividades finalizadas, parte-se então para a filtragem e análise de contas que devem ser propriamente roteadas.

Por meio de uma aferição visual dos critérios carregados em colunas de uma aba específica para cada cliente filtrado na etapa anterior, são classificados os devidos *tiers* de atendimento e já selecionados quais CSMs deverão atendê-los. Com esta relação entre cliente que deve ser roteado e CSM que irá atendê-lo então estabelecida, confecciona-se um arquivo extraído em formato CSV que contenha estas informações. Este arquivo é carregado em um *script* em Python já desenvolvido (que será explorado neste capítulo e então melhorado no desenvolvimento do projeto, como apresentará o capítulo seguinte), que realiza o registro do CSM na conta do cliente no sistema interno de controle de contas da empresa (chamado de RD Contas). O *script* então retorna um relatório de registros realizados, também em formato CSV, que é analisado pelo analista responsável pelo processo para que ele possa realizar qualquer ajuste necessário à marcação automatizada feita. Finalmente, após concluída a etapa de ajustes, finaliza-se o processo.

O mapeamento do processo e sua modelagem em notação BPMN, conforme apresentado acima e no ANEXO A ao final deste documento, cumprem com o primeiro objetivo exposto no capítulo anterior, seção 1.2 (*Mapeamento do processo atual de*

roteamento de contas de novos cliente para a área de CSI em notação BPMN).

4.2 INFORMAÇÕES RELEVANTES PARA CLASSIFICAÇÃO DE TIERS DE CLIENTES

Atualmente, existem 4 níveis de atendimento possíveis para clientes atendidos pela área de Customer Success Inbound, de forma que o *tier* de atendimento de um determinado cliente é definido com base em características inerentes ao produto adquirido e perfil do cliente. Estas características e os valores assumidos por elas que balizam a separação entre os níveis de atendimento configuram informações sigilosas da empresa, portanto a seguir serão apresentadas definições macros e valores fictícios para cada uma delas fazendo com que um panorama suficientemente detalhado possa ser passado, de forma a se compreender o desenvolvimento do projeto descrito ao longo do documento. Estes critérios e seus reais valores são utilizados na etapa do *script* desenvolvido que classifica o nível de atendimento de clientes, de acordo com a lógica descrita na seção dedicada a esta rotina.

1. Tier 1: os critérios a) e b) caracterizam critérios gerais para este *tier*, que é subdividido em dois grupos de atendimento (Pod A e Pod B)
 - a) Contas individuais com MRR menor que R\$1.000,00;
 - b) Contas de grupo com MRR agregado do grupo menor que R\$4.000,00;
 - c) Pod A:
 - i. Clientes entre os meses de fase de adoção à ferramenta;
 - d) Pod B:
 - i. Clientes que já utilizam o produto há um determinado período de tempo, acima da fase de adoção;
2. Tier 2
 - a) Contas individuais com:
 - i. MRR igual ou maior que R\$3.000,00 e menor que R\$4.000,00 ou;
 - ii. Clientes que adquiriram um plano específico com 100.000 contatos ou superior.
 - b) Contas de grupo com MRR agregado do grupo igual ou maior que R\$4.000,00 e menor que R\$5.000,00
3. Tier 3
 - a) Contas individuais com:

- i. MRR igual ou maior que R\$3.000,00 e menor que R\$5.500,00 ou;
 - ii. Plano Y.
 - b) Contas de grupo com MRR agregado do grupo igual ou maior que R\$5.000,00 e menor que R\$15.000.
4. Tier 4
- a) Contas individuais com MRR igual ou maior que R\$5.500,00;
 - b) Contas de grupo com MRR agregado do grupo igual ou maior que R\$15.000,00.

O mapeamento apresentado acima, coletado e sumarizado por meio da análise da estratégia de atendimento da área de CSI, atinge o objetivo de número 2 da seção 1.2 (*Mapeamento dos critérios utilizados para classificar os tiers de atendimento de clientes de CSI*) e será de suma importância para a classificação automática e exata dos níveis de atendimento de clientes a ser discutida no capítulo seguinte.

4.3 INFORMAÇÕES RELEVANTES PARA CLASSIFICAÇÃO DE CARTEIRAS DE CSMS

Neste mapeamento, almejou-se identificar informações que seriam valiosas para determinar qual gerente de contas seria o ideal para um determinado cliente e que também seriam utilizadas na etapa de desenvolvimento dos *dashboards* operacionais voltados para corpo de líderes da área de CSI.

Os seguintes critérios apresentados permitem concluir sobre dois aspectos básicos a respeito dos CSMS, *tiers* e da área de CSI como um todo. O primeiro deles é a carga de trabalho alocada em determinado gerente ou nível de atendimento, medido pelo número de contas da carteira e número de contas em período de adoção à ferramenta (itens 4 e 5, respectivamente). O segundo é sobre a receita gerenciada por determinado CSM ou *tier*, medido pelo critério de MRR gerenciado. O mapeamento apresentado a seguir atinge o objetivo 3 da seção 1.2 (*Mapeamento dos critérios utilizados para classificar as carteiras de clientes dos gerentes de contas de CSI*) e, mais adiante, no capítulo 5, discute-se sobre como esses dados são utilizados efetivamente na elaboração dos *dashboards*.

1. Nome do CSM;
2. Time do CSM;
3. *Tier* de atendimento do CSM;
 - a) Caso pertença ao Tier 1, grupo de atendimento do CSM (Pod A ou Pod B);
4. Número de contas na carteira;

5. Número de contas em período de adoção;
6. Número máximo de contas na carteira (por *tier*) estabelecido pela política de capacidade;
7. Número de contas com risco de cancelamento;
8. MRR gerenciado pelo CSM: soma do MRR das contas na carteira do CSM.

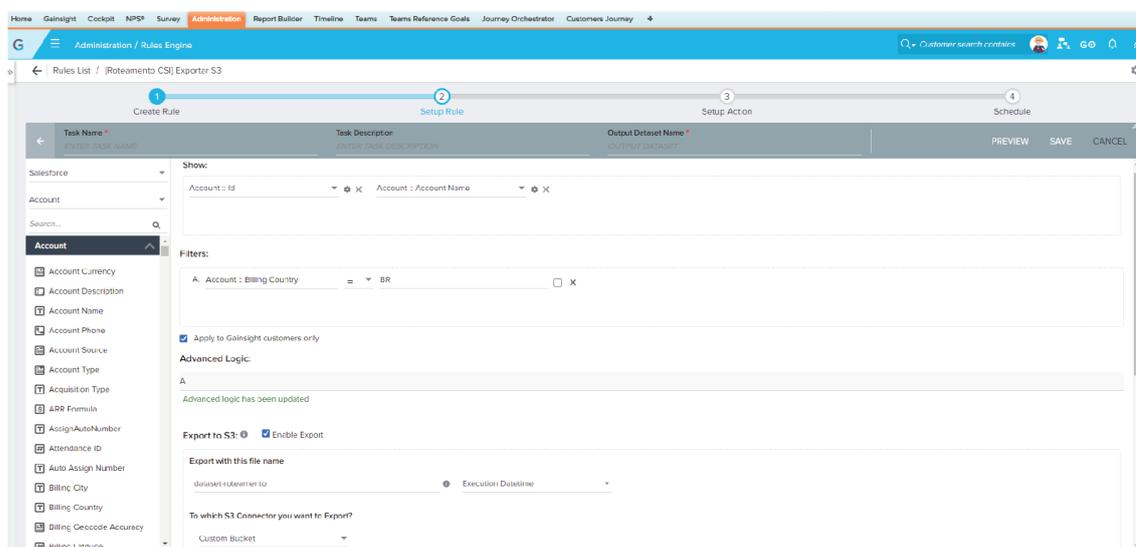
4.4 TECNOLOGIAS E FERRAMENTAS EXISTENTES

De forma a racionar possíveis custos com aquisições de ferramentas novas e também tornar a solução proposta pela prova de conceito mais enxuta, utilizando de fontes de dados e ferramentas já disponíveis, alguns recursos utilizados no processo padrão de roteamento são aproveitados e adaptados no projeto desenvolvido. As duas seguintes tecnologias apresentadas e suas adaptações propostas cumprem com o quarto objetivo exposto na seção 1.2 do capítulo anterior (*Mapeamento das tecnologias e ferramentas que serão adaptadas para o desenvolvimento do projeto*).

O primeiro deles é o CRM utilizado pela área de CS, a ferramenta Gainsight (GAINSIGHT, 2021). Nele, são armazenadas todas as informações pertinentes ao roteamento de clientes, desde MRR por cliente até quais contas fazem parte de grupos. No processo atual de roteamento buscam-se, em diversas etapas da execução da atividade, relatórios preestabelecidos que devem ser manualmente processados e extraídos em formato CSV, para que então possam ser atualizados e tratados na ferramenta Google Sheets. No desenvolvimento do projeto descrito neste documento, para evoluir esta etapa de coleta de dados primordial, utilizou-se de uma funcionalidade nativa do CRM que permite estruturar regras que funcionam como consultas facilitadas ao banco de dados da ferramenta. Estas consultas realizam o trabalho de seleção de dados que são pertinentes ao projeto (tanto para a etapa de roteamento de clientes quanto para a etapa de análise nos *dashboards*), formando tabelas que são armazenadas automaticamente em um horário programado no Amazon S3 e que são acessadas e tratadas pelo *script* como será exposto no capítulo seguinte. A figura 11 a seguir ilustra uma configuração de relatório com capacidade de depósito automatizado no Amazon S3. O Amazon S3 é um serviço web do tipo Representational State Transfer (REST), que pode ser acessado por uma Application Programming Interface (API) REST ou por um Software Development Kit (SDK) especificamente desenvolvidos para este propósito como será exposto na seção 4.5.1. Dentro do serviço, as tabelas são armazenadas dentro de *buckets* (ou repositórios) específicos que só podem ser acessados por chaves de acesso predefinidas (AMAZON, 2021).

Uma outra tecnologia utilizada pelo processo original e adaptada pelo projeto é o *script* preexistente de registro de CSMs nas contas de clientes no *webservice* interno

Figura 11 – Exemplo de regra com capacidade de exportação automática para Amazon S3 pela ferramenta Gainsight.



Fonte – Autor do projeto.

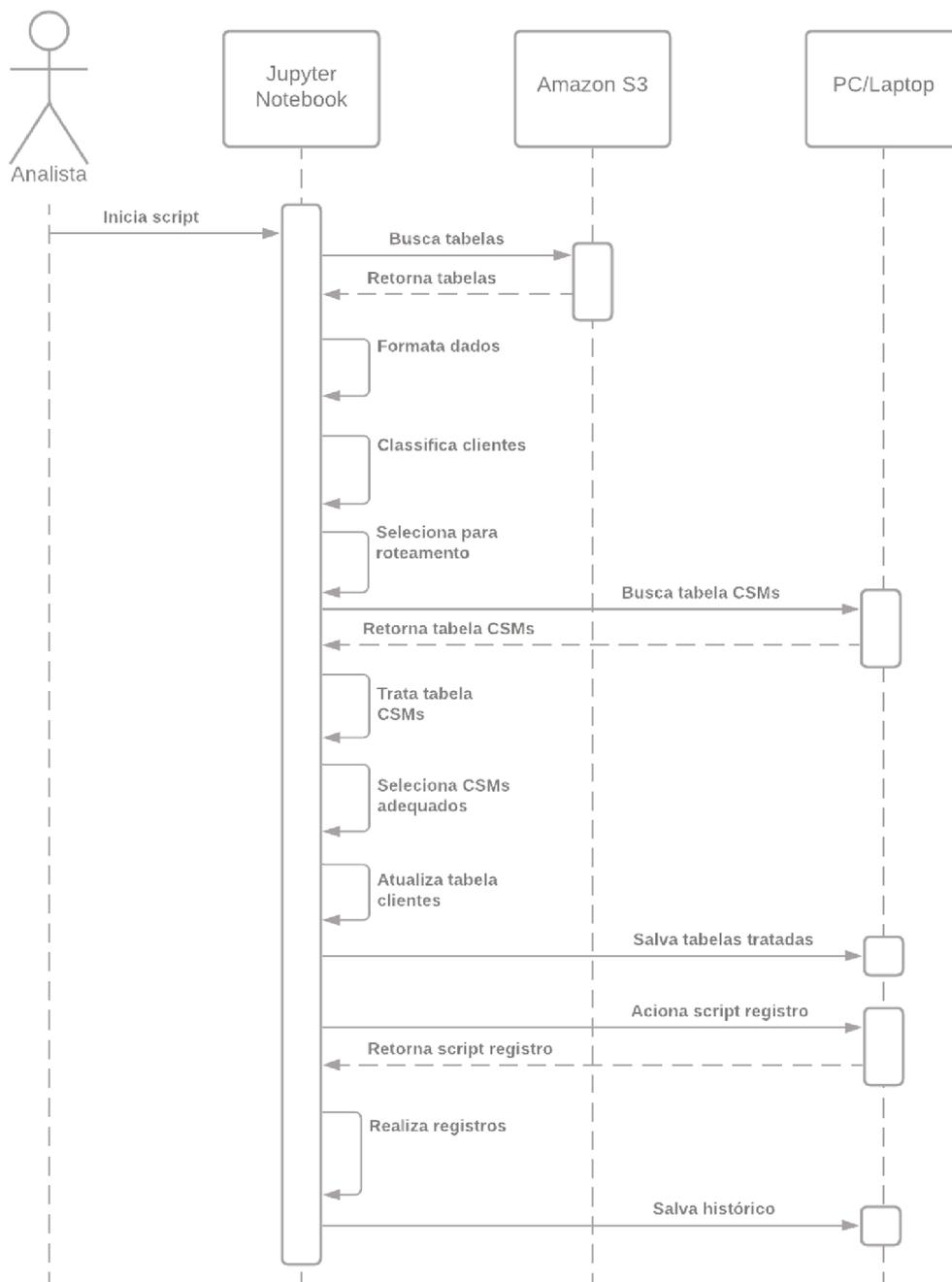
da empresa o RD Contas, que serve como fonte oficial de dados para diversos processos e informações (incluindo a de atendimento de clientes). Antes da execução deste projeto, esta rotina em Python apresentava diversos problemas que causavam marcações indevidas em contas, o que gerava a necessidade de uma etapa de correção de registros ao final do processo de roteamento. Esta correção era feita manualmente pelo analista responsável e a carga de trabalho desprendia para executá-la era diretamente proporcional ao volume de contas roteadas. No capítulo 5 deste documento, detalha-se o funcionamento geral desta etapa e descreve-se quais modificações e melhorias foram realizadas pelo desenvolvedor deste projeto.

4.5 ESPECIFICAÇÃO DA SOLUÇÃO PROPOSTA

Para ilustrar o processo proposto como prova de conceito utiliza-se também da notação BPMN e a modelagem deste é encontrada no ANEXO B deste documento. De forma a elucidar ainda mais a solução, utiliza-se dos diagramas UML de Casos de Uso, Classes, Sequência e Implementação. Os *scripts* desenvolvidos serão implementados no ambiente Jupyter Notebook (JUPYTER. . . , 2021), uma ferramenta da Literate Computing que permite executar códigos de diferentes linguagens no navegador web, sendo possível unir texto, ferramentas gráficas e código em um mesmo ambiente de forma facilitada (VIANA, 2021). Foi escolhido pela sua facilidade de execução de arquivos Python e forte ênfase em ciência de dados e rotinas de BI, e por ser executável em um navegador, minimizando a quantidade de programas que são necessários de

serem baixados para a execução da rotina. Inicia-se pela apresentação do diagrama de Sequência presente na figura 12 abaixo, que ilustra o processo planejado e se comunica proximamente com a modelagem apresentada no ANEXO B.

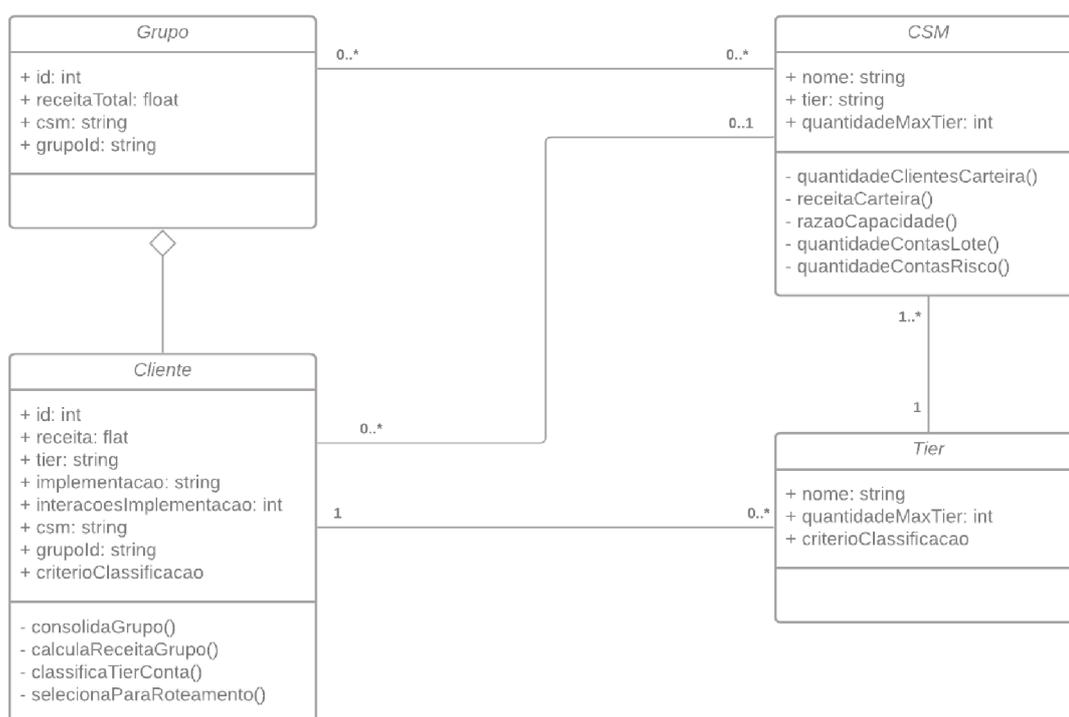
Figura 12 – Digrama de Sequência da rotina automatizada planejada como solução e prova de conceito.



Fonte – Autor do projeto.

Já o diagrama de Classes ilustrado na figura 13 a seguir foi adaptado para representar as categorias de dados tratados. São elas *Clientes*, *CSMs*, *Grupos* (configurados como um conjunto de *Clientes*) e *Tiers*. Os atributos e métodos apresentados foram definidos como as informações mínimas necessárias para cumprir a tarefa de roteamento, não sendo exclusivamente os únicos dados e funções utilizados na rotina de fato implementada apresentada no capítulo 5.

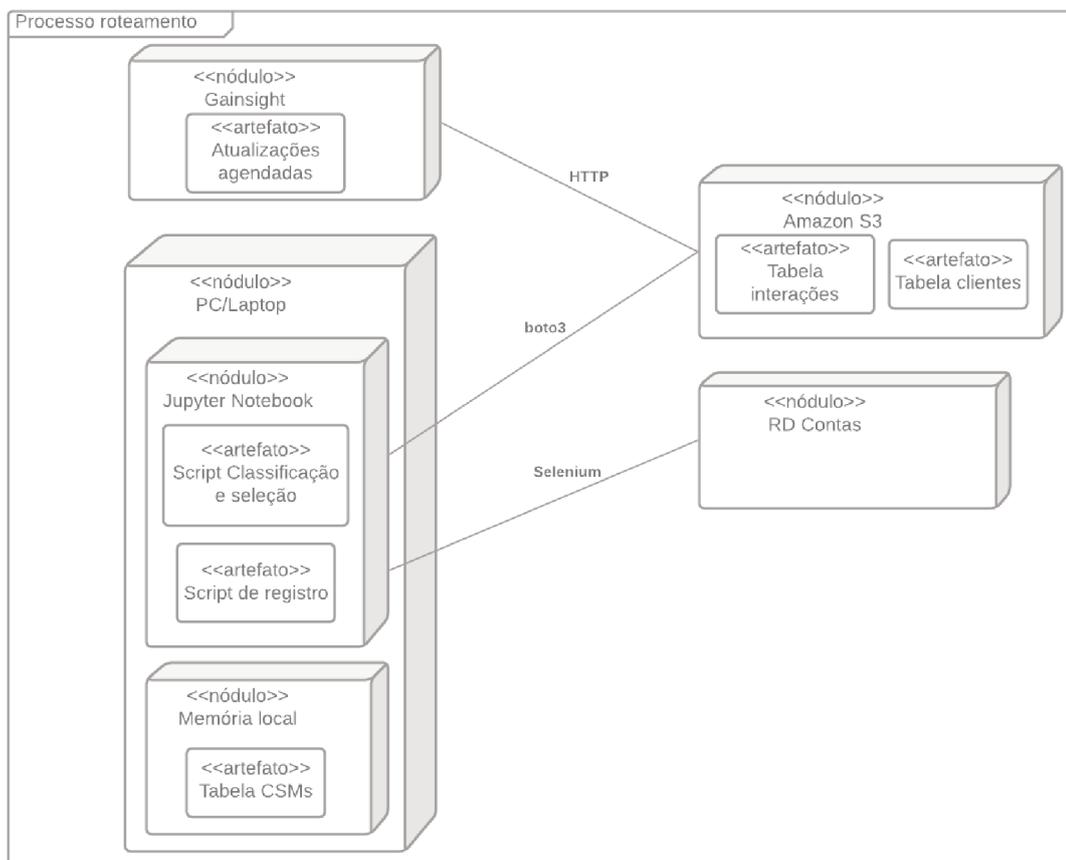
Figura 13 – Diagrama de Classes representando as categorias macro de dados utilizados pela solução.



Fonte – Autor do projeto.

A arquitetura completa da solução proposta é por sua vez ilustrada pelo diagrama de Implementação abaixo, na figura 14. Nela representa-se a atuação do sistema de CRM discutido, que atualiza os dados das tabelas necessárias ao processo no serviço Amazon S3, e como este por sua vez é acessado pelo *script* de classificação e seleção por meio do SDK Boto3 (detalhado na seção seguinte). Ilustra-se também a relação do sistema interno de registro de CSMs em contas de cliente (RD Contas) com o *script* de registro, por meio do pacote Selenium, que foi criado com o propósito de automatizar testes em navegadores web (SELENIUM, 2021) e é adaptado neste código para efetivar os registros necessários no sistema.

Figura 14 – Digrama de Implementação da solução proposta.



Fonte – Autor do projeto.

4.5.1 Aquisição de dados

O processo se inicia no sistema de CRM da área de CS, com o processamento das regras que formam e atualizam as tabelas de dados armazenadas no *webservice* Amazon S3 como apresentado na seção acima. O *script* a ser desenvolvido então deve acessar o repositório específico do serviço que contém os dados relevantes para o roteamento e buscar nele as tabelas atualizadas pela rotina previamente programada do Gainsight. Isso será feito por meio do SDK Boto3, desenvolvido com o intuito de fornecer capacidades de acesso, gerenciamento e configurações de dados armazenados em serviços web da Amazon por simplificar em sua estrutura a API REST necessária para comunicação (BOTO3, 2021).

4.5.2 Tratamento e enriquecimento de tabelas

Após coletar as tabelas de interesse prevê-se a formatação de determinadas colunas que serão usadas para os cálculos de classificação de clientes e carteiras de

CSMs. Em seguida enriquece-se as tabelas que servem de fontes de dados para o processo, adicionando informações que auxiliarão a aferir os níveis de atendimento ideais dos clientes da área de Customer Success Inbound.

4.5.3 Classificação de níveis de atendimento

Com as tabelas propriamente formatadas e enriquecidas, parte-se para a classificação dos níveis de atendimento dos clientes. Esta etapa deve registrar, para cada cliente da base de dados, seu *tier* com base nos critérios definidos pela estratégia de atendimento da área de CSI, apresentados na seção 4.2. Dessa forma, a cada execução da rotina proposta por este projeto será obtida a classificação mais atualizada para cada cliente ativo da base total de contas de Customer Success Inbound. Esta abordagem foi escolhida para permitir possíveis aferições e auditorias de roteamentos, possibilitando identificar alocações indevidas de gerentes de contas. Ainda, é natural – e esperado – que determinadas características dos clientes evoluam ao longo de sua jornada, como MRR ou plano, e por meio da classificação geral da base de clientes agiliza-se o processo de identificação destes casos, conseqüentemente facilitando a realocação de contas entre gerentes quando necessário (processo executado com menor frequência conforme mencionado anteriormente). Dessa forma, essa etapa do código poderia ser reaproveitada em outros processos referentes à área de Customer Success Inbound.

4.5.4 Seleção de clientes para roteamento

A etapa seguinte do processo seleciona quais clientes devem ter seus CSMs alocados para aquela execução do roteamento e armazena-os em uma variável, sendo que estes são identificados conforme os critérios apresentados na seção 4.1 deste capítulo. As tabelas enriquecidas anteriormente são então armazenadas localmente de forma a possibilitar a auditoria de cálculos realizados ou até mesmo utilizá-las para análises a serem entregues ao corpo de líderes da área de CSI, visto que elas já possuirão as classificações mais atualizadas de níveis de atendimento de cliente ativos, informação rica porém custosa de se obter sem o uso do *script* da solução proposta.

4.5.5 Classificação de carteiras de CSMs

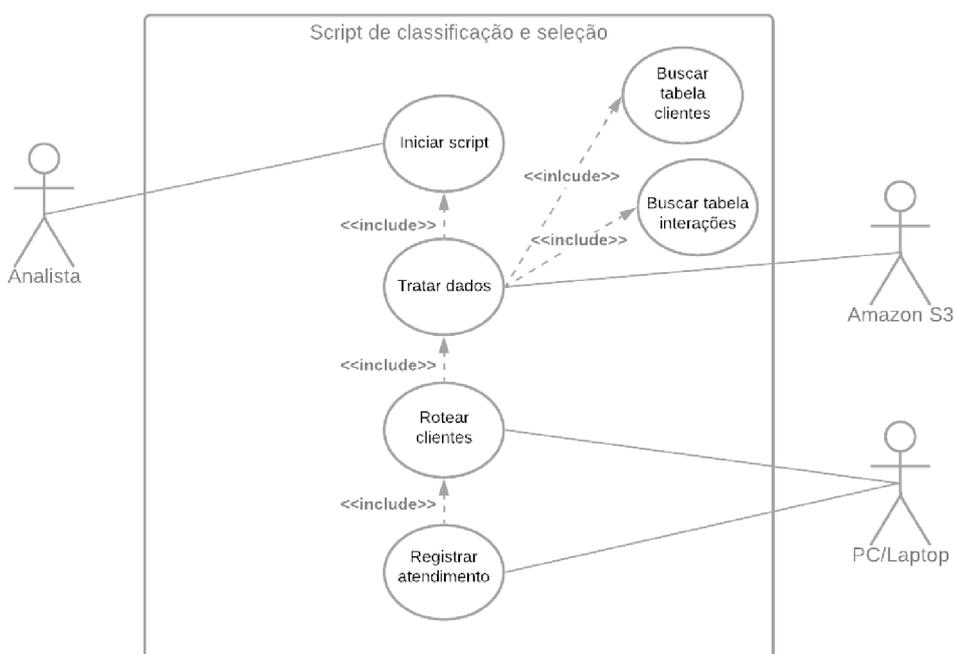
Em seguida o *script* deve buscar e armazenar em uma variável uma tabela previamente construída e armazenada localmente, que contém informações a respeito dos gerentes de atendimento (como nome, *tier* de atendimento ao qual pertence e máximo de contas permitidas pela política de capacidade para seu *tier*). O código então enriquece esta tabela com base em cálculos realizados sobre a tabela de da-

dos de clientes já enriquecida, utilizando desta para aferir informações como receita gerenciada por cada CSM e quantidade de contas totais por CSM.

4.5.6 Seleção de CSMs para atendimento

Esta etapa da rotina utiliza da tabela com dados de CSMs aumentada e das informações de clientes selecionados para roteamento para identificar qual CSM é o mais adequado para cada um destes, retornando uma lista que contenha as alocações definidas. Esta seleção é feita com base nos critérios expostos na seção 4.3, respeitando os parâmetros definidos pela política de capacidade da área de CSI. A tabela com dados de clientes é então atualizada nas entradas referentes aos clientes roteados por aquela execução inserindo o nome do CSM escolhido. O *script* de classificação e seleção até esta etapa pode ser representado pelo diagrama de Caso de Uso abaixo (figura 15). Nele, tem-se como ator primário o analista que iniciará a rotina ao executar o código e como atores secundários tem-se o *webservice* Amazon S3 (responsável por fornecer as tabelas atualizadas para o processo em etapas específicas) e a máquina local do analista em questão, que fornece a tabela de CSMs montada manualmente e recebe as tabelas enriquecidas pelo *script* para armazenamento local.

Figura 15 – Diagrama de Caso de Uso de Classificação e seleção.

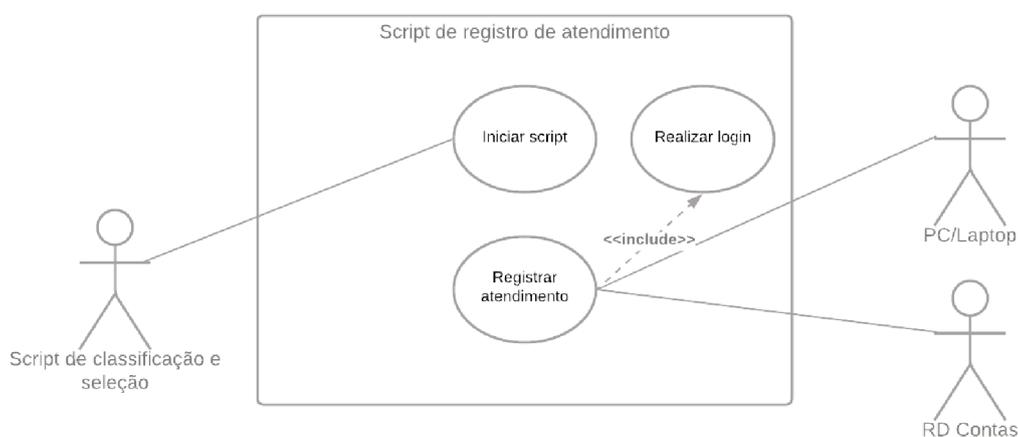


Fonte – Autor do projeto.

4.5.7 Registros de CSMs em contas

Após a atualização da tabela de clientes armazena-se esta localmente, juntamente com a tabela que contém dados de CSMs enriquecida e a lista com as relações entre CSMs e clientes que acabam de ser definidas. Esta última serve como fonte para a etapa final do processo, de registro de gerentes nas contas de seus clientes no sistema interno da empresa por meio do *script* discutido na seção 4.4. Por fim, um histórico dos registros realizados é retornado, de forma que os roteamentos feitos possam ser auditados identificando se foram devidamente executados conforme planejado. Esta etapa pode ser representada pelo diagrama de Caso de Uso abaixo (figura 16), onde o ator primário é o próprio *script* de classificação e seleção como especificado acima que inicia o código de registro ao chamar este em sua última etapa. Como atores secundários tem-se o serviço interno da empresa, que permite capacidade de registros das relações entre cliente e CSM estabelecidas, e a máquina local do analista que executar o processo, que receberá o histórico de registros e fornecerá informações de acesso ao RD Contas.

Figura 16 – Diagrama de Caso de Uso de Registro de Atendimento.



Fonte – Autor do projeto.

4.5.8 Dashboards operacionais

A respeito da elaboração dos *dashboards* operacionais, primeiramente é necessário analisar o método pelo qual os relatórios serão disponibilizados, levando em consideração o atual *stack* de tecnologias disponíveis, ponderando sobre qual ofereceria a melhor navegabilidade e profundidade de detalhes, sendo ainda familiar ao corpo de líderes alvo. A ferramenta escolhida foi o Power BI, produto da Microsoft desenvolvido com foco em produção e publicação de relatórios estratégicos (BI, 2021),

pois além de ser conhecido pelo time de líderes da área de CSI também oferece as ferramentas adequadas para construção dos relatórios planejados, sendo que estes podem ser publicados em um portal online que requer acesso individual (fator importante dada a sigilosidade dos dados trabalhados). Ainda, o sistema é familiar ao time de analistas responsáveis pelo processo de roteamento. Assim, estes poderiam oferecer manutenção e realizar evoluções nos visuais desenvolvidos quando necessário.

Já a respeito dos dados de interesse para as análises do time de líderes, almeja-se disponibilizar informações que permitam diagnosticar possíveis concentrações de receita e riscos de cancelamento em CSMs, assim como concentrações de clientes em carteiras e outras informações que permitam avaliar a carga de trabalho depositada sobre gerentes em geral. Para isso, os seguintes relatórios foram planejados, de forma que cada nível de atendimento teria uma página ou seção dedicada com cada uma das informações abaixo, assim como uma página que analise a área de atendimento em questão como um todo:

1. Total de clientes por *tier*;
2. Total de CSMs por *tier*;
3. Destaque dos gerentes com maior concentração de clientes na carteira;
4. Destaque dos gerentes com maior concentração de receita na carteira;
5. Destaque dos gerentes com maior concentração de risco de cancelamento na carteira;
6. Panorama geral sobre a distribuição de receita entre gerentes no nível de atendimento;
7. Panorama geral sobre a distribuição de clientes entre gerentes no nível de atendimento;
8. Detalhamento sobre a distribuição de risco de cancelamento entre gerentes no nível de atendimento;
9. Detalhamento sobre a distribuição de clientes em período de adoção entre gerentes no nível de atendimento.

4.5.9 Requisitos Funcionais

Os requisitos funcionais do projeto foram elencados ao analisar-se o processo de roteamento original (ANEXO A) e seus principais gargalos (definidos como as etapas que mais desprendiam tempo para serem executadas), aliando a isto o estudo do portfólio atual de tecnologias disponíveis já utilizadas pela empresa e também

pelo time de analistas que possui governança sobre o processo de roteamento. O objetivo principal era automatizar toda a rotina de coleta, tratamento e análise de dados, assim como seleção de gerentes de contas com base em critérios definidos pela estratégia de atendimento da empresa. Abaixo, apresenta-se as funcionalidades mapeadas iniciando-se por aquelas identificadas para o *script* de classificação de clientes e CSMs e seleção de atendimento:

1. Utilização da base de dados disposta pelo CRM da área de atendimento;
2. Acesso automático ao banco de dados;
3. Seleção e recuperação automática das tabelas relevantes para o processo;
4. Normalização de dados;
5. Cálculo do nível de atendimento do cliente com base nos critérios definidos pela estratégia de atendimento da área de Customer Success Inbound;
6. Identificação e recuperação dos clientes que necessitam de um gerente de contas;
7. Classificação das carteiras de gerentes de contas;
8. Seleção do gerente de contas mais adequado para cada cliente, com base nos critérios descritos pela política de capacidade e estratégia da área de Customer Success Inbound;
9. Integração com a etapa de registro de CSMs no portal interno da empresa.

Em seguida, apresenta-se os requisitos mapeados para a etapa de registro de CSMs no portal interno da empresa:

1. Aproveitamento do *script* preexistente para esta etapa;
2. Utilização do resultado retornado pela etapa de seleção de CSMs por clientes;
3. Identificação e pesquisa por meio de ID único;
4. Identificação de clientes inválidos para roteamento;
5. Ignorar as marcações de gerentes em clientes inválidos para roteamento;
6. Normalização da janela do navegador para otimização na identificação de componentes do portal;
7. Retorno de um histórico de registros realizados;

Por fim, elenca-se os requisitos funcionais para os *dashboards* operacionais planejados para utilização do time de líderes da área de atendimento:

1. Utilização das tabelas resultantes do código de busca e tratamento de dados e seleção de CSMs;
2. Atualização de dados facilitada para relatórios;
3. Criação de relatórios com filtros customizáveis pelo usuário;
4. Disponibilização digital por meio de ferramenta com acesso restrito;

4.5.10 Requisitos Não-Funcionais

Os requisitos não-funcionais definem critérios gerais do projeto que baseiam as tomadas de decisões realizadas ao longo do desenvolvimento, buscando aliar os objetivos estratégicos do processo de roteamento e os recursos disponíveis para a elaboração do projeto, evoluindo o desempenho da atividade em relação a tempo necessário para sua execução e assertividade das análises realizadas.

O primeiro requisito não-funcional diz respeito à **democratização do processo de roteamento**, visando não mais onerar um analista em específico para executar o processo. O projeto como um todo foi desenvolvido com o intuito de permitir que qualquer analista da área que exerce governança sobre o processo possa executá-lo com a mesma eficiência e qualidade, tornando o processo como um todo mais robusto e livre de erros humanos independente do agente que o execute.

O segundo requisito não-funcional trata do **aproveitamento e otimização de tecnologias já utilizadas**. Esta estipulação se mostra importante para racionar possíveis custos na aquisição de novas ferramentas, mantendo as tecnologias envolvidas no desempenho do projeto mais próximas possíveis daquelas presentes no leque de conhecimento do time de analistas responsável pela atividade. Dessa forma, possíveis manutenções e evoluções futuras poderiam ser realizadas de forma mais democrática, não necessitando exclusivamente do autor do projeto.

Outro requisito não-funcional é sobre a **eficiência na busca e tratamento de dados**, crucial para otimizar o tempo investido em normalizar as tabelas e informações para então iniciar as análises necessárias. No processo original, a busca e normalização de dados é a etapa mais onerosa, e planeja-se que o *script* desenvolvido no projeto realize estas atividades de forma automática e mais robusta.

Por fim, tem-se o requisito de **assertividade na classificação de clientes**, buscando traduzir para linguagem de código as diretivas determinadas pela estratégia de atendimento da empresa e pelas políticas que regem o funcionamento da área de Customer Success Inbound. Com ele objetiva-se classificar de forma exata os níveis de atendimento de cada cliente, por meio de uma ferramenta e método robusto e ao

mesmo tempo flexível à mudanças ou atualizações de critérios que possam ocorrer de acordo com a evolução da estratégia da área.

O capítulo seguinte trata do desenvolvimento do projeto com base nas especificações expostas neste capítulo.

5 DESENVOLVIMENTO

Este capítulo trata do desenvolvimento efetivo do projeto, apresentando em suas seções as atividades realizadas, seus métodos e justificativas.

5.1 ACESSO E BUSCA DE DADOS NO AMAZON S3 E TRATAMENTO DE DADOS

O *script* de classificação e seleção se inicia com uma rotina de acesso ao Amazon S3 por meio do SDK Boto3, como apresentado no capítulo anterior. Inserindo informações de *login* e diretório de tabelas, é possível resgatar virtualmente do *web-service* as tabelas inseridas neles por meio das regras desenvolvidas no sistema de CRM da área de CS. Assim como no processo original, são recuperadas duas tabelas: uma que contém dados gerais de clientes da área de CS e outra que possui dados sobre as interações de clientes com a área de implementação. Ambas possuem volume e conteúdo correspondentes ao processo original (não prejudicando a qualidade e riqueza de dados do processo), com a vantagem de que não são baixadas localmente para que então possam ser atualizadas em outro local. A figura 17 apresenta um exemplo de acesso ao S3 por meio do SDK utilizado.

Figura 17 – Estrutura de código para acesso ao Amazon S3 pelo SDK Boto3.

```
s3 = boto3.resource(  
    service_name='nome_do_serviço',  
    region_name='região_do_diretório',  
    aws_access_key_id='chave_de_acesso',  
    aws_secret_access_key='chave_de_acesso_secreta'  
)
```

Fonte – Autor do projeto.

Por meio da biblioteca Pandas (PANDAS, 2021), as tabelas são armazenadas localmente e temporariamente no *script* em formato de *dataframes*, variáveis com estrutura semelhante à de uma tabela. Esta biblioteca fornece capacidades de tratamento e análise de dados, com a vantagem de ser possível fazê-lo com dados estruturados facilmente em forma de tabelas (como são nativamente armazenados os dados que servem de base para o processo). Após armazenamento dos dados gerais de clientes e interações de implementação em suas variáveis dedicadas, certos dados são formatados, ajustando valores como números inteiros e decimais, conforme mostra o exemplo da figura 18 a seguir.

Ao final desta etapa, ambas as tabelas se encontram em formatos ideais para serem tratadas nas etapas seguintes da rotina desenvolvida, cumprindo com o objetivo

Figura 18 – Exemplo de armazenamento em formato *dataframe* e formatação de dados pela biblioteca Pandas.

```
print('Buscando tabela de clientes (geral)...')
dados_clientes = pd.read_csv(s3.Bucket('nome_diretorio').Object(nome_tabela).get()['Body'])
print('Tabela de clientes adquirida ({} linhas x {} colunas)\n'.format(dados_clientes.shape[0], dados_clientes.shape[1]))
print('Formatando colunas..')
dados_clientes[['Coluna X', 'Coluna Y']] = dados_clientes[['Coluna X', 'Coluna Y']].apply(pd.to_numeric)
print('Colunas formatadas\n')
```

Fonte – Autor do projeto.

de número 5 definido na seção 1.2 (*Programação e testes da etapa da rotina em Python que busca os dados necessários e realiza o tratamento e formatação destes*). A seção a seguir trata da etapa de classificação de contas de clientes e carteiras de CSMs, que ocorre imediatamente após a coleta e tratamento de dados.

5.2 CLASSIFICAÇÃO DE CONTAS E CARTEIRAS DE CSMs

Após a etapa de coleta e formatação de dados no *webservice* Amazon S3, parte-se para a rotina de classificação de contas de clientes em seus devidos *tiers*, classificação de carteiras de CSMs e identificação de contas que devem ser roteadas. A seguir, apresenta-se o desenvolvimento realizado para cada uma dessas necessidades.

5.2.1 Classificação de clientes em *tiers* de atendimento

Para classificar os clientes em seus devidos níveis de atendimento, utiliza-se apenas da tabela de dados gerais de clientes obtida do Amazon S3. Com base nos dados trazidos por ela, a classificação de clientes ocorre em duas etapas.

A primeira consiste de funções que colhem dados diretamente da tabela e realizam cálculos intermediários para definir padrões importantes para a classificação de *tiers*, visto que estes não são fornecidos diretamente pela fonte (CRM). Mais especificamente, estes dados são referentes aos grupos de contas, e as funções desenvolvidas realizam o trabalho de primeiramente identificar e registrar quais são estes grupos e em seguida consolidar qual o MRR agregado das contas que pertencem a cada um deles.

A segunda etapa por sua vez consiste da função de classificação de níveis de atendimento propriamente dita. Nela, são fornecidos como argumentos os critérios expostos no capítulo 4, seção 4.2, e foi implementada conforme exemplo ilustrado pela figura 19. Por meio dela, é possível realizar a classificação em massa dos clientes da base ativas de contas em conformidade com os parâmetros que definem os níveis de atendimento que resumem a estratégia de CSI. Dessa forma, cada cliente é

devidamente classificado em seu *tier* adequado e conseqüentemente atendido posteriormente por CSMs que atendam suas necessidades específicas, aumentando as chances de expansão de receita por cliente.

Figura 19 – Imagem ilustrativa da função de classificação de clientes em *tiers* de atendimento.

```
def service_level(mrr, group_mrr, plan, base, lifetime):  
    if mrr >= 5500 or group_mrr >= 15000:  
        return "Tier 4"  
    elif (mrr >= 4000 and mrr < 5500) or plan == 'Plano Y' or (group_mrr >=5000 and group_mrr < 15000):  
        return "Tier 3"  
    elif (mrr >=3000 and mrr < 4000) or (plan == 'X' and base >= 100000) or (group_mrr >=4000 and group_mrr < 5500):  
        return "Tier"  
    elif lifetime <= 'X':  
        return "Tier 1 - Pod A"  
    else:  
        return "Tier 1 - Pod B"
```

Fonte – Autor do projeto.

A lógica implementada é um simples “*if-then-else*” entre os parâmetros para cada critério. Esta foi abordagem escolhida dado o fato de que os critérios que balizam os níveis de atendimento são preestabelecidos e fixos e esta abordagem seria a mais simples e eficiente para tratá-los, facilitando ainda a manutenção e modificação do *script* no futuro por agentes que não sejam necessariamente o autor deste projeto.

5.2.2 Identificação de contas para roteamento

Como exposto no capítulo 4, seção 4.1, se qualificam para o atendimento de um gerente de contas na área de CSI aqueles clientes que adquiriram um plano Pro ou Enterprise do produto RD Station Marketing, sendo que caso tenham adquirido junto ao produto um serviço de implementação devem finalizar um determinado número de interações com a área de implementação antes de serem roteados.

Na seção anterior apresenta-se a forma como as tabelas de clientes e interações com a área de implementação são colhidas do serviço Amazon S3, sendo que estas tabelas são então enriquecidas e salvas temporariamente em variáveis. Para realizar a união destas e então filtragem de clientes relevantes para o processo implementam-se funções de união de tabelas (*merge*) e filtragem de dados (*query*) da biblioteca Pandas, como mostra o exemplo figurativo da figura 20 a seguir.

O resultado obtido, ilustrado pela variável “dados_filtrados” no código ilustrado na figura, contém apenas os clientes que devem receber CSMs durante a execução da rotina iniciada, e é disponibilizado também em formato de tabela.

Figura 20 – Imagem ilustrativa da seleção de clientes para roteamento.

```
#filtra clientes hábeis a atendimento CSI
print('Inicia seleção de clientes para roteamento...')
clientes_sem_csm = dados_clientes.query('Nome_do_CSM == Nulo & [...]')
#filtra clientes que possuem X interações de implementações concluídas
iteracao_x_concluida = dados_interacoes.query('Iteração_X_concluida == True')
#une as duas tabelas acima para selecionar quem deve devidamente ser roteado
dados_filtrados = clientes_sem_csm.merge(iteracao_x_concluida, on='ID', how='left').query('Sem_implementacao || Iteracao_OK')
print('Clientes selecionados ({} linhas x {} colunas)\n'.format(dados_filtrados.shape[0], dados_filtrados.shape[1]))
```

Fonte – Autor do projeto.

5.2.3 Classificação de carteiras de CSMs

A etapa do *script* que realiza a classificação das carteiras de CSMs toma como base uma tabela de dados construída manualmente e armazenada localmente na máquina do agente que realizar o processo. Esta tabela possui quatro colunas: nome do CSM, time do CSM, *tier* de atendimento do CSM e máximo de contas indicado por CSM para o *tier*. A abordagem de construir a tabela manualmente foi feita pois dessa forma pode-se lidar mais facilmente com casos de saídas de colaboradores da área, ou outras situações especiais que deveriam impactar no roteamento de contas para um determinado gerente de atendimento, como licença maternidade, férias, etc.

Com base nesta tabela são calculadas informações relevantes para o processo de roteamento e também para a etapa de análise em *dashboards* operacionais ao final do projeto. Estas informações novamente são consolidadas por meio de funções desenvolvidas – que atuam sobre a tabela geral de clientes mencionada na seção 5.1 – que aferem o MRR gerenciado pelo CSM, o número total de clientes em sua carteira e a quantidade de clientes em um período predeterminado de meses que são considerados como o “período de adoção” à ferramenta. Além disso, são consolidados os números de clientes com risco de cancelamento para cada CSM, informação crucial para medir a saúde da base de clientes e que se mostra importante para o trabalho de análises desempenhado pelos líderes da área de atendimento. A figura 21 abaixo ilustra uma função de cálculo de MRR total na carteira de um CSM.

Figura 21 – Imagem ilustrativa da função de MRR na carteira de um CSM.

```
def mrr_csm(nome_csm, tier_csm):
    if tier_csm == 'Tier 1 - Pod A':
        return round(sum(dados_clientes.where((dados_clientes['CSM'] == nome_csm) & (dados_clientes['Lifetime'] <= 'X'),
        inplace = False)['MRR'].dropna()), 2)
    else:
        return round(sum(dados_clientes.where(dados_clientes['CSM'] == nome_csm, inplace = False)['MRR'].dropna()), 2)
```

Fonte – Autor do projeto.

As etapas desenvolvidas conforme descrito pelas seções 5.2.1 à 5.2.3 cumprem com o sexto objetivo apresentado na seção 1.2 a respeito da *Programação e testes da*

rotina em Python de identificação de contas que devem ser roteadas, classificação de tiers de clientes e carteiras de clientes de CSM.

5.2.4 Utilização de funções e consolidação de tabelas

Todas as funções desenvolvidas são executadas com seus respectivos parâmetros utilizando um método de vetorização disponibilizado pela terceira e última biblioteca importada ao projeto, a Numpy, que assim como a biblioteca Pandas permite tratar e realizar cálculos sobre estruturas de dados em formato de *arrays* e matrizes (NUMPY, 2021). Utilizando este método, é possível realizar o cálculo das funções desenvolvidas para cada linha das tabelas fonte em uma rotina semelhante a um *loop for*, colhendo apenas as colunas que correspondem a parâmetros necessários para a função em questão (SARKAR, 2021). A figura 22 abaixo ilustra uma chamada à função com este método, utilizando como exemplo a função que classifica níveis de atendimento de clientes.

Figura 22 – Imagem ilustrativa da etapa de chamada de funções pelo método de vetorização.

```
print('Classificando service level de clientes...')
dados_clientes['Service_Level'] = np.vectorize(service_level, otypes = [object])(dados_clientes['MRR'],
                                                                              dados_clientes['MRR_Grupo'],
                                                                              dados_clientes['Plano'],
                                                                              dados_clientes['Base'],
                                                                              dados_clientes['Lifetime'])
print('Clientes classificados e service levels registrados (Service_Level)\n')
```

Fonte – Autor do projeto.

Cada função desenvolvida ao ser executada gera uma ou mais novas colunas em sua respectiva tabela fonte, complementando tais tabelas que são então baixadas localmente para motivos de auditoria e uso posterior na etapa de elaboração e alimentação de dados aos *dashboards* operacionais. A etapa seguinte do código, que seleciona qual CSM é o mais adequado para uma conta, utiliza das tabelas de clientes e CSMs abordadas nesta seção já expandidas com as colunas calculadas pela funções, e será abordada na seção seguinte.

5.3 ETAPA DE SELEÇÃO GERENTE DE CONTA

Esta seção discorre sobre o método de seleção do CSM mais adequado para uma determinada conta, o raciocínio por trás do método escolhido e como este foi traduzido em linguagem de código para ser inserido no *script* desenvolvido.

5.3.1 Parâmetros para seleção de gerente de contas

Os parâmetros de seleção de um CSM para uma conta específica são dados seguinte os critérios apresentados na seção 4.3 deste documento. Baseado em tais aspectos, ao executar o processo original ilustrado pelo ANEXO A, o analista responsável pelo roteamento de contas realiza uma análise visual dos parâmetros de cada cliente que precisa ser roteado e define qual gerente de contas deve ser registrado para cada um deles. O processo decisório almeja, principalmente, equalizar o volume de clientes entre gerentes de um mesmo nível de atendimento, de forma que a carga de trabalho projetada para cada um seja semelhante. Isto vale não apenas para o número geral de clientes mas também para a quantidade de clientes em cada mês da fase de adoção ao produto. Dessa forma, diminui-se a possível concentração de receita em determinados CSMs e garante-se que todos possuem oportunidades semelhantes de atingimento de resultados, equalizando as oportunidades de cada gerente fornecer um atendimento adequado a seus clientes, conseqüentemente aprimorando resultados de receita colhida por assinatura.

De forma a simular o processo de tomada de decisão feito originalmente, duas formas de avaliação foram definidas. Uma para decidir o gerente de uma conta que se encontra entre os meses de adoção ao produto, e outra para escolher um CSM para um cliente com maior tempo de uso com a ferramenta. Ambas as formas levam em consideração primeiramente o nível de atendimento do cliente, já limitando o espaço de escolhas possíveis para CSMs daquele determinado *tier*.

Para a primeira, a abordagem escolhida foi a formulação de uma razão matemática que pondera quão distante cada CSM está do máximo estabelecido pela política de capacidade da área de CSI para cada *tier* – que por sua vez existe para manter uma estrutura financeiramente estável em relação ao atendimento de clientes na estrutura de pós-venda – avaliando em conjunto o volume de clientes que este CSM possui em determinado mês daqueles definidos como o período de adoção. Esta razão é calculada como exposto na Equação 1 abaixo, de forma que quanto menor o valor obtido por ela “mais ideal” é o CSM para conta sendo roteada:

$$\frac{\text{Total clientes CSM}}{\text{Max total de clientes tier}} \times \text{Total clientes mês } M \quad (1)$$

Já para a segunda forma de avaliação, para clientes além do período de adoção, considera-se apenas a quantidade total de clientes na carteira, sendo que o CSM que possui a menor quantidade é o mais indicado.

Caso haja um empate entre as razões ou quantidade total de clientes por CSM na hora de rotear um determinado cliente, é utilizado o MRR da carteira dos CSMs como critério de decisão, de forma que o CSM com menor receita agregada na carteira é escolhido como o gerente mais adequado.

5.3.2 Programação da etapa de seleção de CSM

A lógica exposta acima foi traduzida em forma de uma função no *script* desenvolvido, que busca na tabela de gerentes de contas – já expandida pelas funções apresentadas na seção 5.2.3 – o CSM mais adequado para um cliente. Ela é implementada sobre a tabela de clientes filtrados para roteamento, apresentada na seção 5.2.2, também por meio de um método de vetorização, de forma que percorre-se a lista de clientes a serem roteados linha por linha para a seleção de seu gerente de contas. A figura 23 abaixo ilustra a função desenvolvida para esta etapa:

Figura 23 – Imagem ilustrativa da função de seleção CSMs para clientes.

```
def seleciona_csm(id_cliente, tier_cliente, lote_cliente):
    tabela_inter = tabela_csms onde 'Tier_csm == tier_cliente'
    if lote_cliente > 'período_de_adoção':
        tabela_inter = tabela_inter onde Total_clientes.ismin()
        if tabela_inter.linhas > 1:
            tabela_inter = tabela_inter onde MRR_CSM.ismin()
            Total_clientes_CSM + 1
            MRR_CSM + MRR_cliente
            print ('Cliente id_cliente roteado para CSM_escolhido')
            retorna CSM_escolhido
        else:
            Total_clientes_CSM + 1
            MRR_CSM + MRR_cliente
            print ('Cliente id_cliente roteado para CSM_escolhido')
            retorna CSM_escolhido
    else:
        tabela_inter = tabela_inter onde Razao_lote_cliente.ismin()
        if tabela_inter.linhas > 1:
            tabela_inter = tabela_inter onde MRR_CSM.ismin()
            Total_clientes_CSM + 1
            MRR_CSM + MRR_cliente
            Clientes do 'lote_cliente' + 1
            atualiza Score do 'lote_cliente'
            print ('Cliente id_cliente roteado para CSM_escolhido')
            retorna CSM_escolhido
        else:
            Total_clientes_CSM + 1
            MRR_CSM + MRR_cliente
            Clientes do 'lote_cliente' + 1
            atualiza Score do 'lote_cliente'
            print ('Cliente id_cliente roteado para CSM_escolhido')
            retorna CSM_escolhido
```

Fonte – Autor do projeto.

Após a seleção de gerentes, a tabela geral de clientes é atualizada com as informações consolidadas pela função da imagem acima, assim a cada nova iteração tem-se uma imagem “real” de como ficariam as carteiras após o roteamento de cada cliente da lista filtrada para roteamento. Esta etapa cumpre por sua vez com o objetivo de número 7 exposto na seção 1.2, sobre a *definição do método de seleção de CSMs para contas e programação e testes deste na rotina Python desenvolvida*

Quando a execução sobre a lista de clientes é finalizada uma nova tabela é montada, contendo duas informações apenas: o ID do cliente e o CSM selecionado para aquele ID. Esta tabela é baixada localmente na máquina do agente que realiza o processo e servirá de base para a próxima e última etapa do roteamento propriamente dito, o registro do gerente de contas na conta do cliente dentro do sistema interno de

gerenciamento de serviços e assinaturas da empresa. Esta etapa final é tratada na seção seguinte.

5.4 ETAPA DE REGISTRO DE CSMS

Como exposto na seção 4.4, e demonstrado pelo ANEXO A, o processo original já conta com um *script* que realiza a marcação de CSMS em seus respectivos clientes no sistema interno da empresa. Contudo, este código apresentava diversos problemas que faziam com que houvesse retrabalho na realização das marcações (representado pela última etapa do processo mapeado pelo ANEXO A). Esta seção descreve com mais detalhes os problemas existentes anteriormente e como o autor do projeto os corrigiu, tornando o processo de marcação de gerentes mais robusto e preciso, atingindo o oitavo objetivo apresentado na seção 1.2 do primeiro capítulo (*adaptação e testes do script de registro de CSMS em contas no sistema interno da empresa desenvolvido em plataforma web*).

5.4.1 Funcionamento geral do *script*

A atividade realizada pelo *script* de registro de gerentes em contas de clientes no *webservice* interno simula o processo manual que deve ser realizado neste, utilizando um pacote de automação de testes em navegadores web que identifica elementos e acessa funções do site da forma como um agente humano faria. A rotina em Python utiliza do pacote Selenium (SELENIUM, 2021), do *webdriver* ChromeDriver (CHROMEDRIVER, 2021) e de um arquivo em formato CSV que contém a relação de clientes e CSMS para acessar no navegador o portal onde os registros de atendentes são realizados, inserindo o usuário e senha do agente que realiza a atividade, e por meio de uma lógica iterativa sobre o arquivo fonte (com a relação de clientes e gerentes) buscando e selecionando o cliente de interesse e finalmente marcando o CSM no espaço dedicado para esta informação.

Para garantir que registros em contas indevidas não serão feitos, o *script* também deve identificar clientes com cancelamento agendado ou inadimplentes, além daqueles que já tiveram seus serviços encerrados. Ainda, é desejável que o processo retorne um arquivo com histórico dos registros realizados pela execução feita de forma a manter-se um *backlog* de informações alteradas para auditoria e possíveis consultas futuras.

5.4.2 Problemas com o *script* original e soluções desenvolvidas

Em sua versão original, o principal problema encontrado pelo *script* era em relação ao ID de cliente utilizado para busca. Frequentemente a busca pelo ID utilizado retornava clientes errados pois a pesquisa era realizada em um campo do portal

que averigua se algum ID dentro de toda a base de clientes continha a informação inserida, e não se algum ID era exatamente igual ao fornecido. Além disso, o ID utilizado para pesquisa definia uma entidade que poderia possuir diversos produtos (como diversos planos RD Station Marketing) sendo que talvez nem todos estes produtos se configurassem para o atendimento de um gerente de contas. Para sanar este empecilho, foi implementada uma busca que utiliza um ID único para cada serviço (informação antes não presente no conjunto de dados utilizado pelo processo original) e ainda realiza a pesquisa deste em um campo específico que retorna apenas um cliente cujo ID de serviço seja exatamente igual ao fornecido.

O *script* original também falhava em identificar casos de clientes que não deveriam ser contemplados pelo processo, como clientes inadimplentes ou com pedido de cancelamento de serviço agendado. Esta etapa é crucial pois afeta diretamente no atendimento e experiência de clientes e ainda em métricas internas de aferição de produtividade de gerentes de contas, visto que caso um cliente em situação de risco de cancelamento tenha um CSM registrado este entraria em contato para realizar um reunião estratégica, gerando conflitos na comunicação e desprendendo um trabalho desnecessário no atendimento do cliente. De forma a solucionar esta questão, a etapa de identificação dos casos mencionados foi amplamente aprimorada para identificar no código fonte do portal (em linguagem HTML) os elementos que indicam que o cliente a ser roteado está em uma situação que o torna inadequado para receber um gerente de contas, ignorando então a marcação prevista. A etapa do código que realiza identificação é ilustrada na figura 24 abaixo.

Figura 24 – Imagem ilustrativa da etapa de identificação de casos indevidos para o roteamento.

```
for e in browser.find_elements_by_xpath("/html/body/section/div/div/div/div[2]/div[2]/div[1]/div[2]/div/div/div[1]"
                                     "/div[1]/span/span[7]/span[contains(text(), 'Cancelamento')]):
    if e.text == 'Cancelamento':
        churnAgendado = True
        status_roteamento = 'CHURN AGENDADO - Status: ignorado'

for e in browser.find_elements_by_xpath("/html/body/section/div/div/div/div[2]/div[2]/div[1]/div[2]/div/div/div[1]"
                                     "/div[1]/span/span[7]/span/span[contains(text(), 'Bloqueado')]):
    if e.text == 'Bloqueado':
        inadimplencia = True
        status_roteamento = 'CLIENTE INADIMPLENTE - Status: ignorado'

for e in browser.find_elements_by_xpath("//*[contains(text(), 'Cancelado (churn)')]):
    if 'Cancelado (churn)' in e.text:
        churn = True
        status_roteamento = 'CHURN - Status: ignorado'

for e in browser.find_elements_by_xpath("//*[contains(text(), 'Encerrado')]):
    if 'Encerrado' in e.text:
        encerrado = True
        status_roteamento = 'ENCERRADO - Status: ignorado'
```

Fonte – Autor do projeto.

Outra melhoria realizada foi em relação ao *layout* da página do *webservice* ao ser acessada pelo *script*. Em sua versão original, ao abrir o portal no navegador a janela deste não ocupava as proporções corretas, o que fazia com que certos ele-

mentos que deveriam ser acessados pelo *script* ficassem dispostos de forma que não era mais possível identificá-los, inviabilizando o processo de marcação automática e encerrando o programa devido ao erro encontrado. Como forma paliativa de resolver esse problema, o agente que realiza o processo devia manualmente maximizar a tela do navegador e excluir um *banner* criado pelo próprio *driver*. No *script* revisado pelo autor do projeto, foi implementada uma etapa de normalização da janela acessada, de forma que esta automaticamente seja maximizada e também livre de qualquer outro componente que interfira na identificação de elementos.

O acesso geral ao arquivo do *script* também foi aprimorado na execução do projeto. Para executá-lo, era necessário antes baixar o arquivo CSV com a relação de clientes e gerentes de contas em um diretório específico, abrir um ambiente de execução de arquivos Python e manualmente executar o código. Agora, o *script* de registros de CSMs é automaticamente chamado ao final da rotina descrita em seções anteriores. Ainda, o método de login ao portal por meio do *script* foi alterado. No arquivo original era necessário fornecer diretamente no código o usuário e senha do analista que estaria realizando o acesso, um formato que expõe informações sigilosas e acaba dificultando a publicação do arquivo em alguma pasta compartilhada de forma que mais agentes possam executá-lo, ferindo o requisito não-funcional de democratização da execução do processo. No *script* aprimorado, as informações de login fornecidas são definidas em um arquivo do tipo JSON e acessadas por meio de variáveis pelo código de acesso ao portal. Dessa forma, o *script* propriamente dito pode ser mantido em uma pasta de acesso compartilhado, sendo que cada agente teria em sua máquina um arquivo JSON com seus dados de acesso predefinidos, mantendo a sigilosidade dos dados. A figura 25 abaixo ilustra esta rotina de acesso e *login* ao portal:

Figura 25 – Imagem ilustrativa da etapa de acesso ao portal de registro de gerentes.

```
fileName = 'relacao de clientes e gerentes'
with open('login.json','r') as f:
    login = json.load(f)

print('AUTENTICAÇÃO\n')
print('Autenticando...')
chrome_options = webdriver.ChromeOptions()
chrome_options.add_experimental_option('prefs', {
    'credentials_enable_service': False,
    'profile': {
        'password_manager_enabled': False
    }
})
chrome_options.add_experimental_option("excludeSwitches", ['enable-automation'])
browser = webdriver.Chrome(options=chrome_options, executable_path=
    r'diretório\chromedriver.exe')
browser.get('endereço do portal')
browser.maximize_window()
user_email = browser.find_element_by_id('user_email')
user_email.send_keys(login['user']['name'])
user_password = browser.find_element_by_id('user_password')
user_password.send_keys(login['user']['password'])
browser.find_element_by_name('commit').click()
print('Iniciando looping...\n')

file = open(fileName, newline='')
reader = csv.reader(file)
header = next(reader)
```

Fonte – Autor do projeto.

O processo de roteamento propriamente dito se encerra quando o *script* tratado nesta seção finaliza sua execução, retornando como resultado um arquivo do tipo CSV com o histórico de ações realizadas, que por sua vez também foi atualizado pelo autor do projeto para conter informações pertinentes à aferição de resultados da rotina, como ID fornecido e ID coletado do portal, CSM previamente registrado no cliente e CSM registrado pelo *script*, estado do registro (“realizado” ou “inorado”, caso o cliente não esteja hábil à marcação) e afins.

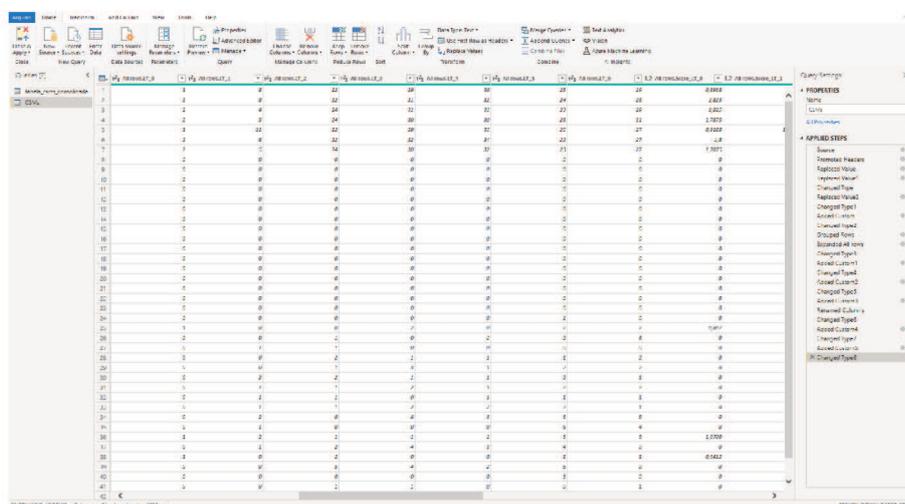
5.5 DESENVOLVIMENTO DOS DASHBOARDS OPERACIONAIS

Esta seção trata da etapa de desenvolvimento dos *dashboards* operacionais que foram planejados para auxiliar a liderança da área de CS Inbound no processo de tomada de decisão, fornecendo informações que seriam valiosas para mensurar possíveis riscos e necessidades na gestão do time de gerentes de atendimento.

O processo de construção dos *dashboards* operacionais se inicia no *script* discutido em seções anteriores, que busca, trata e seleciona CSMs para clientes com base em critérios e lógicas predefinidas. Como exposto na seção 5.2.4, as tabelas utilizadas na rotina são complementadas com colunas calculadas pelo *script* e posteriormente baixadas localmente na máquina do analista que executar o arquivo Python. A tabela de interesse para os *dashboards* operacionais é aquela referente às carteiras dos CSMs.

Ela é importada ao Power BI em formato CSV, que é tratado por uma rotina de passos desenvolvidos para formatar e calcular as campos que serão utilizados pelos relatórios, como ilustra a imagem 26 abaixo.

Figura 26 – Imagem ilustrativa da importação e tratamento de dados na ferramenta Power BI.



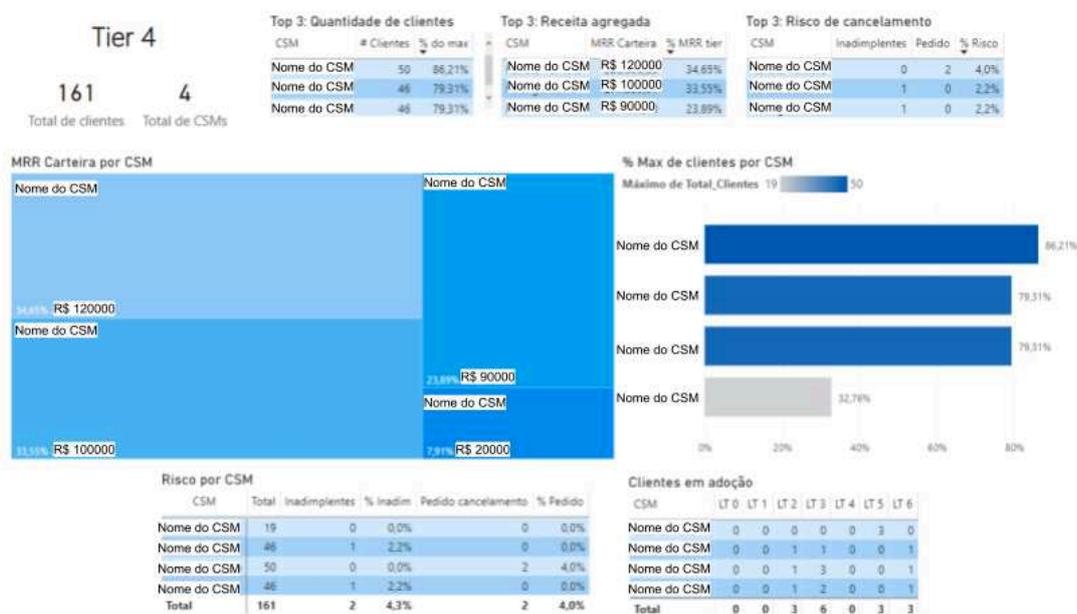
Fonte – Autor do projeto.

A tabela fonte é carregada do arquivo CSV e formatada para que campos de números inteiros, valores verdadeiros ou falsos e outros sejam devidamente identificados nos relatórios a serem desenvolvidos. Após isso, ela é duplicada para que novos cálculos sejam elaborados sobre suas informações sem que a tabela fonte seja modificada. Esta abordagem é considerada uma boa prática caso seja de interesse em desenvolvimentos futuros utilizar da tabela fonte para outros cruzamentos de dados e também como um método de diagnósticos de possíveis erros que possam surgir. As colunas calculadas medem principalmente a representatividade dos dados de cada CSM frente ao seu *tier* e à área de atendimento, como porcentagem de clientes com risco de cancelamento e porcentagem de receita agregada em comparação à carteira de clientes como um todo.

Para as necessidades 1 e 2 expostas na seção 4.5 do capítulo anterior, simples caixas de texto foram utilizadas de forma a destacar a informação de forma direta. Para apresentar os CSMs com maior concentração de receita, clientes e risco de cancelamento, tabelas ordenadas foram o design escolhido, de forma que suas colunas trazem o nome do CSM, o valor absoluto da variável de interesse e quanto este representa frente ao todo. A visualização do panorama geral de distribuição de receita entre CSMs é representada por um gráfico do tipo mapa de árvores, escolhido pela sua capacidade de demonstrar de forma direta proporções de elementos que compõem um grupo pelo área que este elemento ocupa em relação ao todo. Nele, são apresentadas as informações de nome do CSM, MRR por CSM e porcentagem desta receita frente ao seu respectivo *tier* de atendimento. A necessidade 7, sobre o panorama de distribuição de clientes por CSM, é sanada por meio de um gráfico de barras ordenadas de forma decrescente que apresenta a proporção entre a quantidade de clientes na carteira de cada CSM e o máximo estabelecido pela política de capacidade da área para cada *tier*, em forma de porcentagem, sendo que ao topo do gráfico demonstra-se quais são os valores mínimo e máximo absolutos envolvidos naquela visualização. O detalhamento de clientes com risco de cancelamento é exposto por meio de uma tabela que contém os nomes dos CSMs, seus totais de clientes e as informações de clientes inadimplentes ou com pedidos de cancelamento efetuados – em valores absolutos e porcentagens em relação ao total de sua carteira – em suas colunas. Por fim, a necessidade 9, sobre a distribuição de clientes em período de adoção, também é exposta por meio de uma tabela que apresenta os gerentes e suas quantidades de clientes em cada mês do período compreendido como “período de adoção à ferramenta”. Um *mock-up* do design elaborado é exposto na figura 27 abaixo, utilizando de informações fictícias por questões de sigilosidade.

Foram desenvolvidas 5 páginas com as informações expostas acima, uma para cada nível de atendimento e uma para a área de Customer Success Inbound como um todo. Apesar de ser desenvolvido e atualizado localmente na máquina do autor

Figura 27 – Imagem ilustrativa do modelo de dashboards operacionais desenvolvidos.



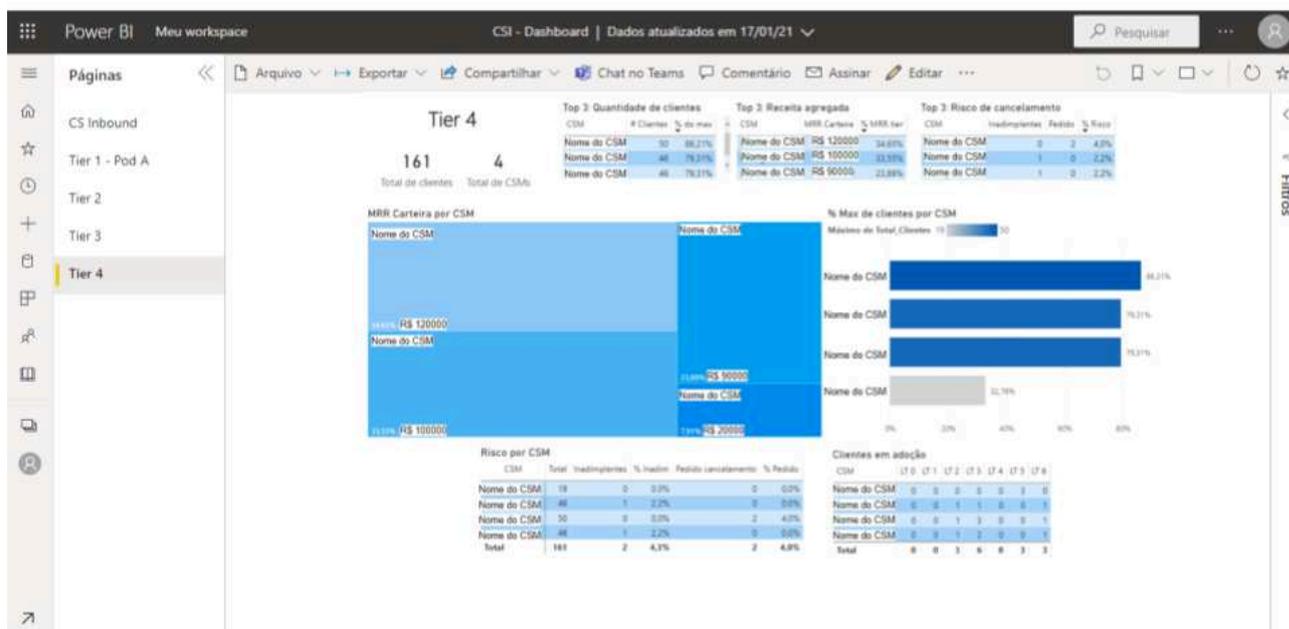
Fonte – Autor do projeto.

deste projeto, os dashboards são disponibilizados no portal online da ferramenta, onde cada líder ao acessar pode realizar seus próprios filtros que selecionem apenas CSMS de interesse, por exemplo. A figura 28 apresenta a visualização disponibilizada para o corpo de líderes da área de CSI no portal online da ferramenta.

Com a construção descrita nesta seção, os dois últimos objetivos apresentados na seção 1.2 são alcançados, finalizando a prova de conceito proposta como escopo do projeto.

O próximo capítulo trata dos testes e validações realizadas para averiguar o desempenho da rotina desenvolvida e apresenta também um conclusão sobre os resultados obtidos com a prova de conceito desenvolvida.

Figura 28 – Visualização disponibilizada para a liderança da área de CSI.



Fonte – Autor do projeto.

6 TESTES E METODOLOGIA DE AVALIAÇÃO DO DESEMPENHO

Os testes para validar o desempenho do processo desenvolvido foram realizados gradualmente ao longo da elaboração e finalização de cada etapa, seguindo o modelo de desenvolvimento em *sprints*.

6.1 ETAPA DE COLETA E FORMATAÇÃO DE DADOS

Iniciou-se validando as informações colhidas do *webservice* Amazon S3 e as formatações realizadas sobre estas pelo *script*. Para isso, as tabelas eram baixadas localmente a partir de duas fontes: o *script*, após a execução de sua etapa de acesso, coleta e formatação de dados, e o CRM que fornece as informações não só para o *webservice* mas também para o processo original. Os testes foram realizados pelo período de uma semana, comparando a cada execução do processo de roteamento original os dados que este utilizava e aqueles retornados pelo *script* em uma ferramenta de planilhas. Para aferir se o código estava executando sua função de forma apropriada, foram utilizados como indicadores fórmulas que comparavam as informações de cada coluna correspondente entre as duas fontes, retornando valores “VERDADEIRO” ou “FALSO” dependendo do resultado da igualdade. Após algumas correções, constatou-se que todas as colunas correspondentes entre os dois métodos eram exatamente iguais, comprovando a eficiência da etapa de busca, armazenamento e formatação de dados do repositório remoto.

6.2 ETAPA DE SELEÇÃO DE CLIENTES PARA ROTEAMENTO

Após isso, validou-se a etapa do código que identificava e filtrava os clientes elegíveis a roteamento, executando o *script* até esta etapa e comparando o resultado retornado com a seleção manual realizada pelo analista responsável no processo original. O indicador utilizado para analisar o sucesso desta etapa foi simplesmente a quantidade de contas corretamente identificadas como hábeis a roteamento – ou seja, novos clientes dos planos Pro ou Enterprise sem CSM alocado, sendo que caso tenham contratado um serviço de implementação devem ter finalizado uma certa quantidade de interações desta – aferido ao realizar, pelo período de uma semana, a investigação de todas as contas identificadas por ambos os métodos de seleção, calibrando qualquer quesito com base nas divergências encontradas. Identificou-se que o código era mais exato na identificação de contas que necessitavam roteamento, estando este livre de erros humanos na análise de dados, permitindo que não houvessem mais atrasos na identificação e conseqüentemente atendimento de clientes novos.

6.3 ETAPA DE CLASSIFICAÇÃO DE *TIERS* DE CLIENTES

Em seguida, foram testadas as fórmulas que classificam os níveis de atendimento de clientes. Para validá-las, novamente as tabelas retornadas pelo código eram baixadas localmente e analisadas em uma ferramenta de planilhas. O indicador utilizado para aferir a exatidão desta etapa foi a quantidade de contas corretamente classificadas em seus devidos níveis de atendimento. Para aferi-lo o método utilizado consistia em construir fórmulas nas planilhas de validação que retornassem o *tier* dos clientes com base nos critérios definidos pela estratégia de atendimento, em uma lógica semelhante àquela utilizada no *script* desenvolvido (ou seja, aferindo os *tiers* de todas as contas ativas da base de clientes de Customer Success Inbound). O resultado destas fórmulas era comparado com nível de atendimento retornado pelo *script*, sendo que esta comparação era avaliada cliente por cliente. Esta etapa foi executada também pelo período de uma semana, onde constatou-se que código desenvolvido atingiu 100% de exatidão em classificar clientes em seus devidos níveis de atendimento. Constatou-se, também, que a execução do *script* até esta etapa leva em média 12 segundos para ser executada, o que demonstra um enorme ganho frente ao tempo que esta leva no processo original (média de 20 minutos).

6.4 ETAPA DE CLASSIFICAÇÃO DE CARTEIRAS DE CSMS

Um método semelhante foi utilizado para testar e validar a etapa que complementa a tabela com a relação de gerentes de contas com dados relevantes para o processo de roteamento, conforme descrito no capítulo 5, seção 5.2.3. Como o detalhamento de informações pelo código era maior que aquele encontrado no processo original foi necessário construir um arquivo que permitisse comparar as informações consolidadas pelo *script* com suas similares, aferidas utilizando das bases de dados do processo original, e fórmulas que simulassem os cálculos realizados pela rotina em Python em uma planilha. Após a comparação dos resultados de cada método de cálculo para cada CSM, uma amostra julgada significativa era colhida e a prova real dos resultados era tirada dos sistemas que fornecem dados ao processo, aferindo se a quantidade de clientes totais por CSM, receita gerenciada e o restante das informações condizia com o que foi calculado. Dessa forma, pelo período de uma semana, utilizou-se dos critérios calculados pela rotina (total de clientes por CSM, MRR gerenciado por CSM, razões matemáticas para os meses de adoção, etc.) como indicadores de sucesso, constatando que estes calculavam com exatidão as informações necessárias e concluindo que a etapa funcionava corretamente.

6.5 ETAPA DE SELEÇÃO DE CSMS

A etapa mais longa dos testes foi referente à seleção de CSMS para clientes que necessitavam de roteamento. Nela, o *script* era executado até esta parte (ou seja, as etapas descritas acima já eram utilizadas) e ao final era retornada a relação entre IDs de clientes e os CSMS que foram selecionados para cada um com base na lógica implementada descrita no capítulo 5, seção 5.3.2. Paralelamente, o processo de roteamento original era executado e ao final de ambos comparava-se a seleção feita pelo analista no processo original e aquela retornada pelo *script*. Para aferir se a lógica programada era válida e atendia os critérios estipulados pela política de capacidade, a função que seleciona o CSM para cada conta era simulada de forma iterativa manualmente pelo autor do projeto, sendo que a cada iteração eram calculados os fatores envolvidos. Assim, foi possível constatar que de fato a escolha do gerente de contas era feita com base na razão desenvolvida (caso o cliente se encontrasse na fase de adoção ao produto) ou com base no total de clientes na carteira (caso o cliente já se encontrasse além da fase de adoção), sendo que para cada uma dessas situações o CSM com menor valor para o critério de relevância era realmente o CSM escolhido.

6.6 ETAPA DE REGISTRO DE CSMS EM CONTAS

Por fim, foram realizados testes com o *script* de acesso e registro de gerentes no *webservice* interno da RD. Sua execução era acompanhada visualmente e de forma iterativa para cada cliente do arquivo base, averiguando se os componentes adequados do site eram acessados, utilizando do relatório de registros feitos que é retornado ao final da execução deste código como indicador de sucesso, analisando se os clientes adequados foram contemplados pela rotina e se os CSMS corretos foram marcados em cada um deles.

6.7 AVALIAÇÃO DA SOLUÇÃO E DASHBOARDS

A rotina como um todo, desde sua etapa de acesso ao repositório virtual do Amazon S3 até a execução do *script* de registro de CSMS no *webservice* interno da empresa, foi também avaliada pela supervisora deste projeto. Além de garantir que o objetivo do projeto estava sendo cumprido (oferecer ganho operacional e exatidão na escolha de gerentes de atendimento para contas de clientes), o foco de sua avaliação foi a respeito das tecnologias utilizadas e lógicas desenvolvidas para selecionar os CSMS ideais para cada cliente. Ambas as avaliações possuem teor qualitativo, onde desejava-se entender se as abordagens implementadas poderiam ser compreendidas por outros membros da equipe responsável pelo roteamento (de forma que pudessem oferecer manutenção e aprimoramentos ao método no futuro, como demanda o requi-

sito não funcional de *democratização do processo de roteamento*) e se os fatores de seleção de gerentes de contas condiziam com a estratégia de atendimento e políticas internas da RD. Após a análise meticulosa da rotina desenvolvida e acompanhamentos de sua execução em conjunto com o autor do projeto, conclui-se que esta atendia de forma satisfatória o objetivo estipulado para a prova de conceito proposta.

A avaliação dos *dashboards* operacionais desenvolvidos também foi realizada pela supervisora deste projeto, e se estendeu também para outros membros da equipe responsável pelo roteamento. Por meio de demonstrações guiadas de uso e análise, o autor do projeto apresentou as funcionalidades dos relatórios, sua base de dados e rotina que consolida as informações. A conclusão atingida é de que o método utilizado para alimentar tais relatórios é seguramente mais eficiente e assertivo que outros utilizados para consolidar as informações mostradas, e que estas também configuram de fato dados extremamente relevantes para uma análise a respeito da área de CSI.

7 CONCLUSÃO

Após a finalização do projeto e a avaliação da equipe de analistas responsável pela execução do processo de roteamento (a qual a supervisora deste projeto coordena), conclui-se que ele cumpriu seu objetivo de oferecer um processo com pouco investimento operacional para ser executado e com nível de precisão na seleção de CSMs não só mais exato quanto também mais robusto, de acordo com as políticas estabelecidas pela empresa.

Como exposto no capítulo 1, seção 1.1, o processo original toma cerca de 4 a 5 horas semanais para ser executado. Já a solução proposta por este projeto realizou as mesmas tarefas com uma média de apenas 18 segundos por execução (cerca de 90 segundos semanais), medidos pelo período de duas semanas onde a mesma média similar de clientes roteados por semana foi constatada. Vale salientar que este tempo compreende o processo desde sua etapa de acesso ao banco de dados do Amazon S3 até seleção de CSMs para cada cliente que precisa ser roteado. A etapa de registros de gerentes no sistema interno não foi mensurada pois apesar das melhorias realizadas, seu tempo de execução não foi alterado, dado que as modificações realizadas pelo autor do projeto influenciam na precisão da execução dos passos necessários para registrar as informações no serviço interno, e não no tempo que estas tomam.

Além do ganho operacional demonstrado pelo tempo de execução da rotina automática, há também o ganho na precisão de escolha de gerentes de contas para clientes. Anteriormente o processo abria brecha para diversos erros humanos dada a natureza manual da execução da maioria de suas etapas. Com a solução proposta, garante-se robustez e exatidão na busca por dados, aferição de critérios e seleção de CSMs pela codificação da lógica que norteia o processo e da automação das atividades deste. Ainda, garante-se que os critérios que balizam a alocação de contas para gerentes, pautados pela política de capacidade da área e estratégia de atendimento definida pelos critérios expostos na seção 4.2, são sempre atendidos de forma imparcial (visto que subtrai-se o fator humano da seleção de gerentes para clientes) e precisa.

Em suma, o projeto se aproxima de uma solução ideal para a necessidade da empresa de rotar os clientes de Customer Success Inbound e para necessidade do time de analistas responsável pelo processo de realizá-lo precisamente sem grandes desprendimentos de recursos. Alguns fatores, porém, ainda impedem que a solução descrita por este projeto se consolide como a forma oficial de roteamento. Estes fatores são expostos na seção seguinte, que trata das propostas de melhorias para o projeto.

7.1 PROPOSTAS DE MELHORIAS

Duas melhorias são propostas para que o projeto descrito neste documento seja integrado nas atividades da área responsável pelo roteamento completamente.

A primeira delas diz respeito ao roteamento de contas que classificam um grupo. Atualmente, é avaliado que estas devem ser atendidas por um mesmo gerente de contas, de forma que cada cliente ou entidade contratante tenha seu atendimento concentrado em apenas um especialista da área de CSI. Na solução desenvolvida e descrita por este documento, garante-se apenas que as contas de um grupo sejam corretamente identificadas e roteadas para CSMs que integram o *tier* de atendimento ao qual este grupo pertence, não sendo todas necessariamente focadas em um gerente apenas. Propõem-se que em possíveis desenvolvimentos futuros a lógica de seleção de CSMs leve esta diretiva estratégica em conta.

A segunda proposta de melhoria diz respeito ao cenário de roteamento de contas em situações em que todos os CSMs de um *tier* tenham atingido o máximo estipulado pela política de capacidade da área de CSI para se respectivo nível de atendimento. Na abordagem atual, seguindo a premissa estratégica de agilizar ao máximo o atendimento de novos clientes, o *script* desenvolvido mantém a avaliação e seleção de CSMs com base nos critérios descritos na seção 5.3.1, de forma a priorizar a agilidade no atendimento de clientes. Porém, propõem-se como melhoria que a rotina trate os casos de inflação de capacidade, seja parando o roteamento completamente e gerando um alerta ou aderindo a critérios de seleção específicos para essas situações.

REFERÊNCIAS

5 indicadores de desempenho para medir seu sucesso. Disponível em:

<https://endeavor.org.br/estrategia-e-gestao/indicadores-de-desempenho/>.

Acesso em: 3 fev. 2021.

AMAZON. **AWS**: O que é o Amazon S3? Disponível em:

https://docs.aws.amazon.com/pt_br/AmazonS3/latest/dev/Welcome.html. Acesso

em: 4 jan. 2021.

ANALYTICS and Business Intelligence (ABI). Disponível em:

[https://www.gartner.com/en/information-technology/glossary/business-](https://www.gartner.com/en/information-technology/glossary/business-intelligence-bi)

[intelligence-bi](https://www.gartner.com/en/information-technology/glossary/business-intelligence-bi). Acesso em: 3 fev. 2021.

ARANTES, Rhaíssa Nogueira. **Introdução ao Business Process Modeling**

Notation (BPMN). Disponível em: [https://www.devmedia.com.br/introducao-ao-](https://www.devmedia.com.br/introducao-ao-business-process-modeling-notation-bpmn/29892)

[business-process-modeling-notation-bpmn/29892](https://www.devmedia.com.br/introducao-ao-business-process-modeling-notation-bpmn/29892). Acesso em: 3 fev. 2021.

BI, Power. **Power BI**. Disponível em:

<https://powerbi.microsoft.com/en-us/what-is-power-bi/>. Acesso em: 4 jan.

2021.

BIZAGI. **Bizagi**. Disponível em: <https://www.bizagi.com/?lang=en>. Acesso em: 3

fev. 2021.

BOTO3. **Boto3**. Disponível em:

<https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>. Acesso

em: 4 jan. 2021.

BPMN. **BPMN**: ABOUT THE BUSINESS PROCESS MODEL AND NOTATION

SPECIFICATION VERSION 2.0. Disponível em:

<https://www.omg.org/spec/BPMN/2.0>. Acesso em: 4 jan. 2021.

BPMN 2.0 by Example Version 1.0. Disponível em:

<https://www.omg.org/cgi-bin/doc?dte/10-06-02.pdf>. Acesso em: 31 jan. 2021.

BURSTEIN, Frada. **Handbook on decision support systems**. Berlin London:

Springer, 2008. ISBN 9783540487159.

CHROMEDRIVER. **ChromeDriver**. Disponível em:
<https://chromedriver.chromium.org/>. Acesso em: 4 jan. 2021.

COHN, Mike. **Agile Needs to Be Both Iterative and Incremental**. Disponível em:
<https://www.mountaingoatsoftware.com/blog/agile-needs-to-be-both-iterative-and-incremental>. Acesso em: 3 fev. 2021.

DIAS, Ricardo Pereira. **O Modelo Incremental**. Disponível em:
<https://medium.com/contexto-delimitado/o-modelo-incremental-b41fc06cac04>. Acesso em: 3 fev. 2021.

DIGITAIS, Resultados. **Tudo sobre Customer Success: o que é, principais métricas e muito mais**. Disponível em:
<https://resultadosdigitais.com.br/blog/o-que-e-customer-success/>. Acesso em: 20 jan. 2021.

ECMASCRIPT Language Specification. Disponível em:
<https://tc39.es/ecma262/#sec-overview>. Acesso em: 31 jan. 2021.

ESTÁCIO. **Aula 8: O desenvolvimento do software em cascata**. Disponível em:
<http://estacio.webaula.com.br/cursos/gra063/aula8.html>. Acesso em: 3 fev. 2021.

FEW, Stephen. **Information dashboard design : the effective visual communication of data**. Beijing Cambridge MA: O'Reilly, 2006. ISBN 9780596100162.

GAINSIGHT. **Gainsight**. Disponível em:
<https://www.gainsight.com/customer-success/>. Acesso em: 4 jan. 2021.

GROFFE, Renato Jose. **Como eu uso UML para desenvolver sistemas?** Disponível em: <https://www.devmedia.com.br/modelagem-de-sistemas-atraves-de-uml-uma-visao-geral/27913>. Acesso em: 3 fev. 2021.

JUPYTER. Disponível em: <https://jupyter.org/>. Acesso em: 3 fev. 2021.

KILLHAM, Evan. **What Is SaaS?** Disponível em:
<https://www.lifewire.com/what-is-saas-5069831>. Acesso em: 4 jan. 2021.

LUCIDCHART. **O que é um diagrama de sequência UML?** Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-sequencia-uml>. Acesso em: 3 fev. 2021.

LYNCH, Warren. **A Comprehensive Guide to BPMN.** Disponível em: <https://warren2lynch.medium.com/a-comprehensive-guide-to-bpmn-5cfd91868a36>. Acesso em: 3 fev. 2021.

MEDEIROS, Higor. **Introdução ao Modelo Cascata.** Disponível em: <https://www.devmedia.com.br/introducao-ao-modelo-cascata/29843>. Acesso em: 3 fev. 2021.

MEDEIROS, Higor. **Introdução aos Processos de Software e o Modelo Incremental e Evolucionário.** Disponível em: <https://www.devmedia.com.br/introducao-aos-processos-de-software-e-o-modelo-incremental-e-evolucionario/29839>. Acesso em: 3 fev. 2021.

MICROSOFT Excel. Disponível em: <https://www.microsoft.com/pt-br/microsoft-365/excel>. Acesso em: 3 fev. 2021.

NEW features in UML syntax and semantics - Scientific Figure on ResearchGate. Disponível em: [https://www.researchgate.net/figure/UML-20-Diagrams-3-Changes-to-previously-known-diagrams-UML-20-introduces-more-detail-to_fig8_228438653%20\[accessed%204%20Feb,%202021\]](https://www.researchgate.net/figure/UML-20-Diagrams-3-Changes-to-previously-known-diagrams-UML-20-introduces-more-detail-to_fig8_228438653%20[accessed%204%20Feb,%202021]). Acesso em: 3 fev. 2021.

NUMPY. **Numpy.** Disponível em: <https://numpy.org/>. Acesso em: 4 jan. 2021.

PANDAS. **Pandas.** Disponível em: <https://pandas.pydata.org/docs/>. Acesso em: 4 jan. 2021.

SARKAR, Tirthajyoti. **Why you should forget loops and embrace vectorization for Data Science.** Disponível em: <https://www.experfy.com/blog/bigdata-cloud/why-you-should-forget-loops-and-embrace-vectorization-for-data-science/>. Acesso em: 4 jan. 2021.

SELENIUM. **Selenium.** Disponível em: <https://www.selenium.dev/>. Acesso em: 4 jan. 2021.

SKOK, David. **SaaS Metrics 2.0**. Disponível em:

<https://www.forentrepreneurs.com/saas-metrics-2/>. Acesso em: 3 fev. 2021.

STATION, RD. **RD Station CRM**. Disponível em: <https://www.rdstation.com/crm/>.

Acesso em: 4 jan. 2021.

STATION, RD. **RD Station Marketing**. Disponível em:

<https://www.rdstation.com/marketing/>. Acesso em: 4 jan. 2021.

STATION, RD. **Resultados Digitais**. Disponível em:

<https://resultadosdigitais.com.br>. Acesso em: 4 jan. 2021.

STEDMAN, Craig. **business intelligence (BI)**. Disponível em:

<https://searchbusinessanalytics.techtarget.com/definition/business-intelligence-BI>. Acesso em: 3 fev. 2021.

STEDMAN, Craig. **operational intelligence (OI)**. Disponível em:

<https://searchbusinessanalytics.techtarget.com/definition/operational-business-intelligence>. Acesso em: 3 fev. 2021.

SUTHERLAND, Jeff. **Scrum: The Art of Doing Twice the Work in Half the Time**.

[S./]: Random House Business, 2015. ISBN 9781847941107.

SUTNER, Shaun. **business intelligence dashboard**. Disponível em:

<https://searchbusinessanalytics.techtarget.com/definition/business-intelligence-dashboard>. Acesso em: 3 fev. 2021.

TABLEAU. **What is Tableau?** Disponível em:

<https://www.tableau.com/why-tableau/what-is-tableau>. Acesso em: 3 fev. 2021.

UML. **WHAT IS UML**. Disponível em: <https://www.uml.org/what-is-uml.htm>.

Acesso em: 3 fev. 2021.

VIANA, Suzana. **Jupyter**. Disponível em:

<https://medium.com/@suzana.svm/por-que-usar-jupyter-notebook-77d5a59b42a1>. Acesso em: 3 fev. 2021.

WHAT is a scripting language and what are the most common ones? Disponível em: <https://en.rockcontent.com/blog/scripting-languages/>. Acesso em: 31 jan. 2021.

ANEXO A – ROTEAMENTO ATUAL DE CLIENTES EM NOTAÇÃO BPMN

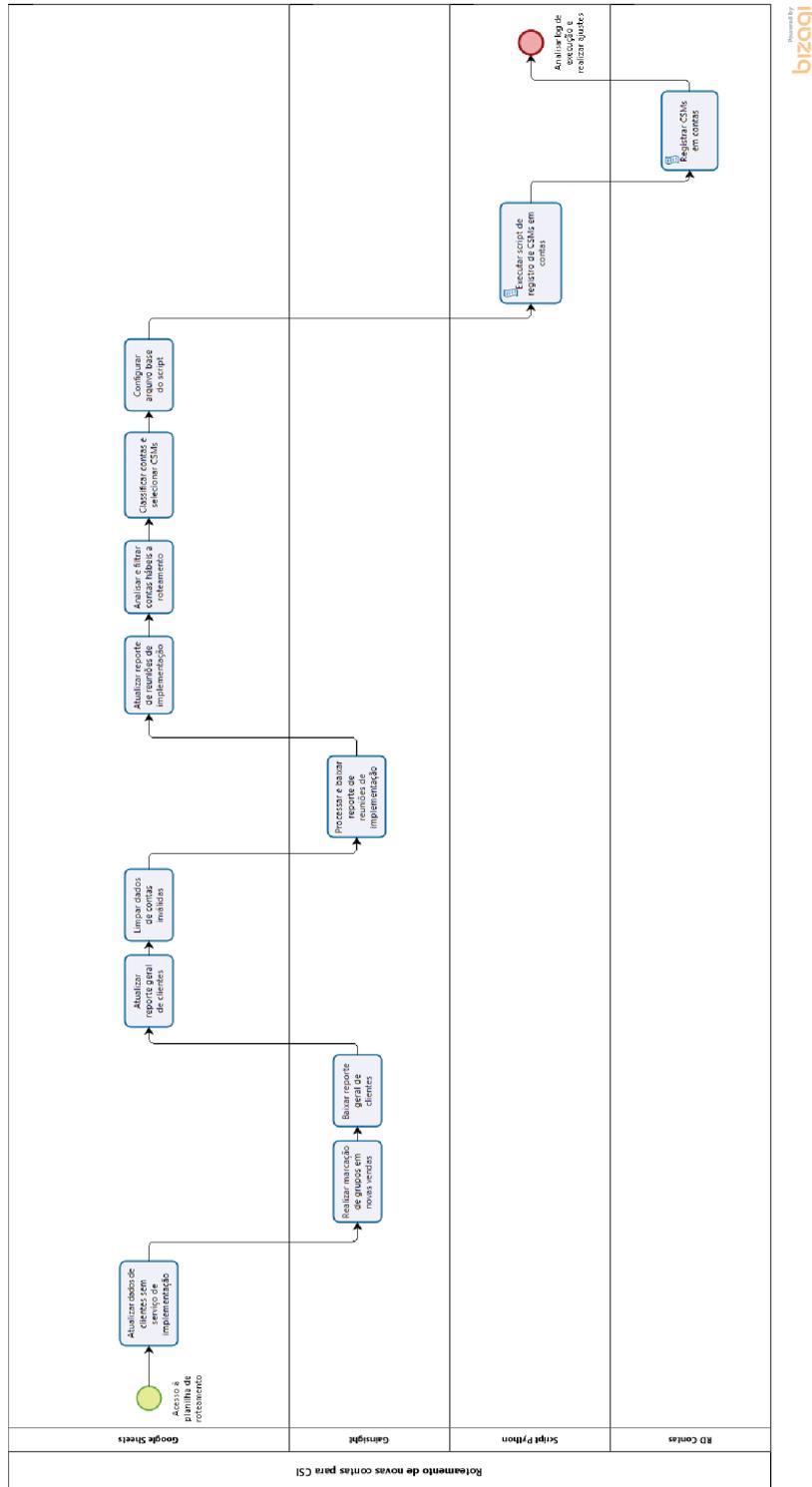


Figura 29 – Mapeamento do processo de roteamento de clientes em notação BPMN.

