

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

ANÁLISE DE DADOS PARA USUÁRIOS NÃO TÉCNICOS
UTILIZANDO O MÉTODO SSBI

GEOVANI FINOTI LEITÃO

BLUMENAU
2018

GEOVANI FINOTI LEITÃO

ANÁLISE DE DADOS PARA USUÁRIOS NÃO TÉCNICOS

UTILIZANDO O MÉTODO SSBI

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Profa. Simone Erbs da Costa, Especialista– Orientadora

**BLUMENAU
2018**

ANÁLISE DE DADOS PARA USUÁRIOS NÃO TÉCNICOS

UTILIZANDO O MÉTODO SSBI

Por

GEOVANI FINOTI LEITÃO

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente: _____
Prof. Simone Erbs Costa, Especialista – Orientadora, FURB

Membro: _____
Prof. Gabriele Jennrich Bambineti, Especialista – FURB

Membro: _____
Prof. Roberto Heinzle, Doutorado – FURB

Blumenau, 11 de julho de 2018

RESUMO

Este trabalho apresenta a especificação, desenvolvimento e operacionalidade da aplicação intitulada SELBST, que tem como objetivo disponibilizar uma aplicação para montar análises financeiras dinâmicas de forma integrada. A aplicação visa auxiliar o usuário nas tomadas de decisões referente ao seu negócio, disponibilizando dados bancários, para que ele possa visualizar e analisar essas informações de forma rápida e eficiente. A aplicação em si é composta por uma Application Programming Interface (API) de integração e a interface de interação com usuário. A API é responsável por realizar a comunicação entre o banco de dados do cliente e o banco de dados da aplicação, sendo ela codificada utilizando a linguagem de programação JavaScript com NodeJs. A interface, por sua vez, é responsável pela experiência de interação do usuário para com a aplicação, desenvolvida utilizando o React no qual se interliga com o componente Plotly, responsável por disponibilizar a área de trabalho para o usuário, bem como utilizar os serviços do Firebase de controle de usuário e de banco de dados. Por meio da avaliação de usabilidade, elaborada seguindo as heurísticas de Nielsen, aplicada em três participantes chave da aplicação, foi possível constatar que a aplicação obteve um índice satisfatório no que se diz respeito a usabilidade da aplicação, cumprindo assim seus objetivos principais. Além disso, também foi possível elencar alguns pontos a serem melhorados na aplicação, com base no feedback dos participantes, visando evoluir cada vez mais a experiência do usuário no uso da aplicação.

Palavras-chave: SELBST. NodeJs. REACT. Plotly. Firebase.

ABSTRACT

This work presents the specification, development and operation of the SELBST application, which aims to provide a tool to assemble dynamic financial analyzes in an integrated way. The system aims to assist the user in making decisions regarding his business, providing banking data so that he can view and analyze this information quickly and efficiently. The application itself is composed of an integration Application Programming Interface (API) and the user interface. The API is responsible for performing the communication between the client database and the application database, and is coded using the JavaScript programming language with Nodejs. The interface, in turn, is responsible for the entire user interaction experience with the application, developed using React in which it interconnects with the Plotly component, responsible for making the wokspace available to the user, and also using the Firebase user control and database services. Through the usability evaluation, elaborated following the heuristics of Nielsen, applied in three key participants of the system, it was possible to verify that the application obtained a satisfactory index with respect to the usability of the application, thus fulfilling its main objectives. In addition, it was also possible to list some points to be improved in the application, based on the feedback of the participants, aiming to evolve more and more the experience of the user in the use of the system.

Key-words: SELBST. NodeJs. REACT. Plotly. Firebase.

LISTA DE FIGURAS

Figura 1 – Etapas do DW	15
Figura 2 – Modelagem Dimensional do BI	17
Figura 3 – Dashboard	18
Figura 4 – Modelo de Trabalho SSBI.....	19
Figura 5 - SAS WorkSpace	21
Figura 6- DV Dados	22
Figura 7 - DV Gráfico	23
Figura 8 - DV Análise	23
Figura 9 - Tableau WorkSpace.....	24
Figura 10 - Tableau Filtro.....	25
Figura 11 - Tableau Filtro.....	26
Figura 12- Diagrama de casos de uso do SELBST.....	30
Figura 13 - Diagrama de atividades: criar dashboard.....	32
Figura 14 - Diagrama de componentes da aplicação SELBST	33
Figura 15- Diagrama de Arquitetura da aplicação SELBST.....	34
Figura 16 - Diagrama de Deploy da aplicação SELBST	35
Figura 17 - Tela de login.....	51
Figura 18 - Header.....	52
Figura 19 - Tela de gerenciamento de usuário.....	52
Figura 20 - Tela de cadastro de usuário.....	52
Figura 21 - Tela de gerenciamento de regras	53
Figura 22 - Tela de cadastro de regras	54
Figura 23 - Lista de Cubos.....	54
Figura 24 – Tela de Área de trabalho	55
Figura 25 - Lista de visualizações.....	56
Figura 26 - Lista de fórmulas.....	56
Figura 27 - Lista de informações	57
Figura 28 - Selecionando informações.....	57
Figura 29 - Visualização das informações.....	58
Figura 30 - Aplicação de filtros	58

Figura 31 - Tela de filtros	59
Figura 32 - Legenda.....	59
Figura 33 - Funcionalidades da visualização.....	60
Figura 34 – Resposta referente ao Papel dos Participantes.....	63
Figura 35 – Resposta referente ao Perfil dos Participantes.....	64
Figura 36 - Resposta referente ao Feedback do sistema	66
Figura 37 - Resposta referente ao Feedback do cubo selecionado.....	67
Figura 38 - Resposta referente aos Conceitos Familiares	67
Figura 39 - Resposta referente a Ordem lógica	68
Figura 40 - Resposta referente ao Sair da opção selecionada.....	68
Figura 41 - Resposta referente as Ações do sistema.....	69
Figura 42 - Resposta referente ao Estilo e padrões consistentes	69
Figura 43 - Resposta referente a Ocorrência de erros	70
Figura 44 - Resposta referente as Instruções visíveis.....	70
Figura 45 - Resposta referente as Imagens ou ícones.....	71
Figura 46 - Resposta referente aos Diálogos do sistema.....	71
Figura 47 - Resposta referente a Linguagem simples	72
Figura 48 - Resposta referente as Opções de ajuda	72
Figura 49 – Gráfico de Índice de atendimento e impacto das heurísticas	73
Figura 50 - Informações do Participante	82
Figura 51 – Heurística H1 - Visibilidade do Sistema.....	83
Figura 52 – Heurística H2 - Correspondência entre o sistema e o mundo real.....	84
Figura 53 – Heurística H3 - Controle e liberdade do usuário.....	85
Figura 54 – Heurística H4 - Consistência e padrões	86
Figura 55 – Heurística H5 - Prevenção de erros.....	87
Figura 56 - Heurística H6 - Reconhecimento no lugar de memorização.....	87
Figura 57 - Heurística H7 - Flexibilidade e eficiência de uso.....	88
Figura 58 - Heurística H8 - Desenho estético e minimalista.....	88
Figura 59 - Heurística H9 - Auxílio no reconhecimento, diagnóstico e recuperação de erros	89

Figura 60 - Heurística H10 - Ajuda e Documentação.....	90
Figura 61 - Roteiro de Teste.....	91
Figura 62 - Termo de consentimento Livre e Esclarecido.....	92

LISTA DE QUADROS

Quadro 1 - Comparativo entre os trabalhos correlatos	26
Quadro 2- Matriz de rastreabilidade de Requisitos Funcionais X UC.....	31
Quadro 3 – Codificação das rotinas Rotas.....	37
Quadro 4 – Codificação das rotinas Auxiliares.....	38
Quadro 5 – Codificação da rotina updateDatabase	39
Quadro 6 - Codificação das rotinas de acesso a aplicação	40
Quadro 7- Codificação das rotinas antes de iniciar o acesso a aplicação.....	40
Quadro 8 – Codificação da rotina iniciando o acesso a aplicação.....	41
Quadro 9 – Codificação da rotina de configuração de campo de formulário...41	
Quadro 10 – Codificação da rotina formulário do login.....	42
Quadro 11 – Codificação da validação dos campos do login.....	43
Quadro 12 – Codificação da rotina inicialização do login.....	43
Quadro 13 – Codificação das rotinas de administrador.....	44
Quadro 14 – Codificação da rotina que busca informações do usuário.....	45
Quadro 15 – Codificação da rotina de Alteração de usuários.....	46
Quadro 16 – Codificação da rotina de Alteração de regras	47
Quadro 17 – Codificação da rotina Mudança de Cubo.....	47
Quadro 18 – Codificação da rotina inicialização do Dashboard.....	48
Quadro 19 – Codificação da rotina criação do Header	49
Quadro 20 – Codificação da rotina criação do workspace.....	50
Quadro 21- Comparativo entre os trabalhos correlatos e a aplicação SELBST	61
Quadro 22 – Relação das Heurísticas com as Perguntas do Questionário65	
Quadro 23 – Relação das Heurísticas e impactos.....	66
Quadro 24 - Caso de uso UC4 - Processo Batch	79
Quadro 25 - Caso de uso UC2 - CRUD usuário	79
Quadro 26 - Caso de uso UC3 - CRUD Regras.....	79
Quadro 27 – Caso de uso UC5 - Selecionar Cubo de dados.....	80
Quadro 28 - Caso de uso UC6 - Alterar forma de visualizar as informações.....	80

Quadro 29 - Caso de uso UC7 - Aplicar fórmula nas informações disponíveis.....	80
Quadro 30 - Caso de uso UC8 - Arrastar campos para o Workspace.....	80
Quadro 31 - Conjunto básico das Heurísticas de Nielsen	81

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface

BI – Business Intelligence

CRUD – Create, Read, Update, Delete

DV – Data Visualization

DW – Data Warehouse

IDE – Ambiente de Desenvolvimento Integrado

OLAP – Online Analytical Processing

PJ – Pessoa Jurídica

RF – Requisito Funcional

RNF – Requisito Não Funcional

SSBI – Self-Service Business Intelligence

TCLE – Termo de Consentimento Livre e Esclarecido

TI – Tecnologia da Informação

UC – Use Case

UML – Unified Modeling Language

VA – Visual Analytics

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 OBJETIVOS.....	13
1.2 ESTRUTURA.....	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 DATA WAREHOUSE.....	15
2.2 BUSINESS INTELLIGENCE.....	16
2.3 DASHBOARD	17
2.4 SELF-SERVICE.....	19
2.5 TRABALHOS CORRELATOS	20
2.5.1 Visual Analytics	20
2.5.2 Data Visualization.....	21
2.5.3 Tableau Desktop.....	24
2.5.4 Comparação entre os trabalhos correlatos.....	26
3 DESENVOLVIMENTO	28
3.1 LEVANTAMENTO DE INFORMAÇÕES	28
3.1.1 Requisitos.....	29
3.2 ESPECIFICAÇÃO	29
3.2.1 Diagrama de casos de uso	30
3.2.2 Matriz de rastreabilidade dos RFs X UC	31
3.2.3 Digrama de atividade para criar dashboard.....	31
3.2.4 Diagrama de componentes	32
3.2.5 Diagrama de Arquitetura.....	33
3.2.6 Diagrama de Deploy	34
3.3 IMPLEMENTAÇÃO	35
3.3.1 Técnicas e ferramentas utilizadas.....	35
3.3.2 Implementação do desenvolvimento.....	36
3.3.3 Operacionalidade da implementação	50
3.4 RESULTADOS E DISCUSSÕES.....	60
3.4.1 Comparativo entre os trabalhos correlatos e a aplicação SELBST.....	60
3.4.2 Avaliação de usabilidade.....	62
4 CONCLUSÕES.....	74

4.1 EXTENSÕES	75
REFERÊNCIAS	76
APÊNDICE A – DETALHAMENTO DOS CASOS DE USO	79
APÊNDICE B – CONJUNTO BÁSICO DAS HEURÍSTICAS DE NIELSEN	81
APÊNDICE C – QUESTIONÁRIO DE USABILIDADE	82
APÊNDICE D – ROTEIRO DE TESTE	91
APÊNDICE E – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)	92

1 INTRODUÇÃO

O uso da tecnologia no setor bancário é relevante para o negócio, principalmente, segundo Leite (2007), no que se refere à redução de custos e competitividade no mercado. Nesse sentido, Leite, Diniz e Jayo (2009) afirmam que com a evolução bancária, as instituições financeiras têm dado maior importância para a eficiência operacional, com o intuito de obter sucesso. Contudo, a eficácia na gestão operacional das agências é um aspecto que se torna cada vez mais crítico, no que se refere ao contexto de produtividade, custo e qualidade de atendimento, pois a gestão em si, se mostra complexa, devido ao tamanho da rede, volume de dados envolvidos e a necessidade de agilidade nas tomadas de decisões.

Assim, de acordo com Leite, Diniz e Jayo (2009), uma das possíveis soluções para aperfeiçoar a tomada de decisão em ambientes dinâmicos é fazer uso de Business Intelligence (BI), que nestes casos, tem-se mostrado satisfatório para a gestão operacional de agências em grandes bancos. Entretanto, conforme colocado pelos autores (LEITE; DINIZ; JAYO, 2009), ainda existem muitos desafios entre o BI e a gestão de agências bancárias para serem superados.

Atualmente, existem algumas tendências para o BI. Entretanto, Castro (2016) observa que é importante saber o contexto em que elas serão aplicadas e quanto serão utilizadas para determinar o seu sucesso em uma organização. Apesar das possibilidades propiciadas pelo BI na tomada de decisões, se faz necessário ter um profissional apropriado e uma infraestrutura tecnológica estruturada, para um funcionamento adequado do sistema.

O Self-Service Business Intelligence (SSBI) é uma dessas tendências que têm chamado à atenção do mundo corporativo de BI, devido, segundo Oliveira (2014), a democratização dos dados. Dessa forma, os usuários passam a ter maior facilidade em acessar as visões de dados fornecidas pela equipe de Tecnologia da Informação (TI), tornando o processo de elaboração de painéis mais autossuficiente. Nesse sentido, Castro (2016), explica que os conceitos de SSBI vieram justamente para desmitificar a ideia de que o BI é exclusivo de área de TI, aproximando cada vez mais o usuário de negócio nesse mundo de dados.

Diante deste cenário, este trabalho se propõe disponibilizar uma ferramenta para usuários que tenham uma conta Pessoa Jurídica (PJ) nas instituições financeiras, com um foco em especial para os serviços oferecidos pela Viacredi, possam montar análises dinâmicas de maneira simples e ágil de forma integrada.

1.1 OBJETIVOS

O objetivo geral deste trabalho é desenvolver uma aplicação para montar análises financeiras dinâmicas de forma integrada.

Os objetivos específicos são:

- a) integrar o banco de dados com a instituição financeira de estudo, Viacredi;
- b) identificar os focos de análise, bem como as dimensões que serão atendidas e os indicadores disponibilizados;
- c) prover aos usuários PJ da Viacredi uma camada analítica para análise financeira dos dados;
- d) facilitar a descoberta da relação entre os dados por interface disponibilizada e avaliada por meio de questionário de usabilidade;
- e) implementar os processos necessários à extração, transformação e carga dos dados;
- f) construir e disponibilizar mecanismo analítico por meio de *dashboard* e cubos OLAP.

1.2 ESTRUTURA

O trabalho desenvolvido está dividido em quatro capítulos. O primeiro capítulo apresenta a introdução do trabalho desenvolvido, o problema, a justificativa, os objetivos desenvolvidos e a definição de sua estrutura. No segundo capítulo é apresentada a fundamentação teórica utilizada durante o projeto, sendo abordados os conceitos extração, manipulação e visualização dos dados, bem como uma explanação sobre o conceito de *self-service business intelligence*. O terceiro capítulo apresenta o desenvolvimento do trabalho e nele são detalhados os requisitos do modelo desenvolvido e suas diagramações, a sua implementação e as técnicas e ferramentas utilizadas durante o processo. Ademais, é detalhado o processo de avaliação do trabalho, assim como as discussões e resultados obtidos durante o projeto. Por fim, no quarto e último capítulo as conclusões do trabalho desenvolvido são apresentadas, bem como, as extensões planejadas para serem implementadas no futuro.

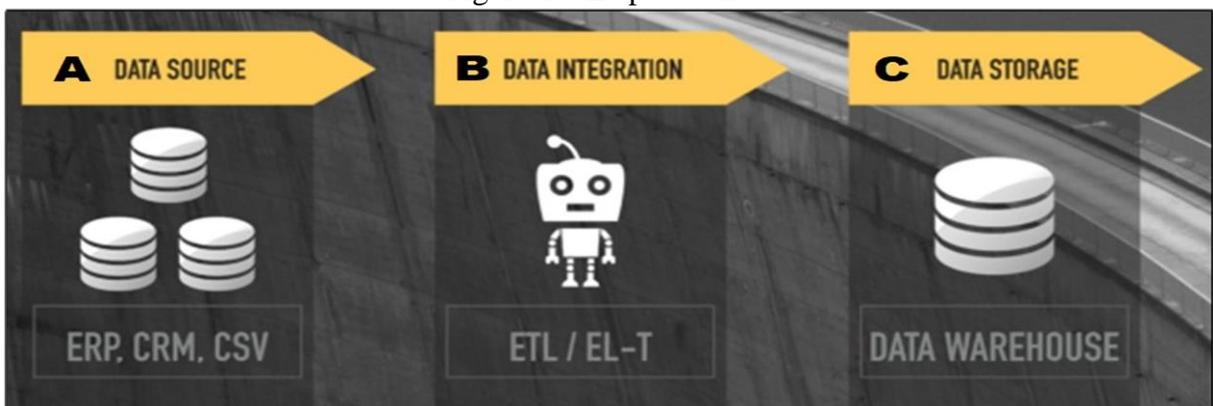
2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo descrever brevemente, os principais assuntos para realização deste trabalho. A seção 2.1 abordará o Data WareHouse (DW) e a necessidade de armazenar as informações. Na seção 2.2 é apresentado o Business Intelligence (BI) e a importância da informação no processo de tomada de decisão. Na seção 2.3 é demonstrado o Dashboard juntamente com a importância da visualização da informação. A seção 2.4 aborda o Self-Service Business Intelligence (SSBI), juntamente com o conceito e tendência do mercado em sua utilização. Por fim, na seção 2.5 é demonstrado os trabalhos correlatos selecionados como base para desenvolver a aplicação estudada e a correlação entre eles.

2.1 DATA WAREHOUSE

Uma das mais importantes tecnologias que complementam o desenvolvimento do BI é o Data WareHouse (DW) (ELIAS, 2014). O propósito do DW é armazenar dados consolidados de várias fontes em um único local, denominando dados estruturados, que visam à qualidade e a precisão dos mesmos (CETAX, 2016). O DW tem seu processo de manipulação dos dados segmentados em três principais etapas, sendo elas demonstradas pela Figura 1: Data Source (letra A), Data Integration (letra B) e Data Storage (letra C), que individualmente atuam de forma diferenciada sobre o dado para que no final do processo ele esteja preparado para ser utilizado (PITON, 2017).

Figura 1 – Etapas do DW



Fonte: adaptado de Piton (2017).

A primeira etapa refere-se ao Data Source, local físico da origem dos dados a serem modelados e futuramente utilizados no BI. Data Integration é a segunda etapa do DW, em que os dados são preparados e modelados, aplicando todos os cálculos e regras necessárias para obter o resultado desejado. Por fim, a última etapa é o Data Storage, local físico referente ao armazenamento das informações geradas pela e na etapa anterior, e conseqüentemente, a origem de dados dos relatórios e *dashboards* a serem utilizados pelo usuário (PITON, 2017).

De acordo com Dionizio (2013), atualmente as empresas de grande e médio porte tendem a gerar e armazenar grande volume de informações em diversos locais diferentes. Essa ampla quantidade de dados em conjunto com a tecnologia a fim de extrair corretamente certas informações é um fator chave para o sucesso de uma corporação, tendo em vista o mercado de trabalho competitivo. Seguindo nessa linha tecnológica, o Data Warehouse (DW) surge para suprir essa necessidade de integrar e armazenar dados corporativos de diferentes origens (RIBEIRO, 2011).

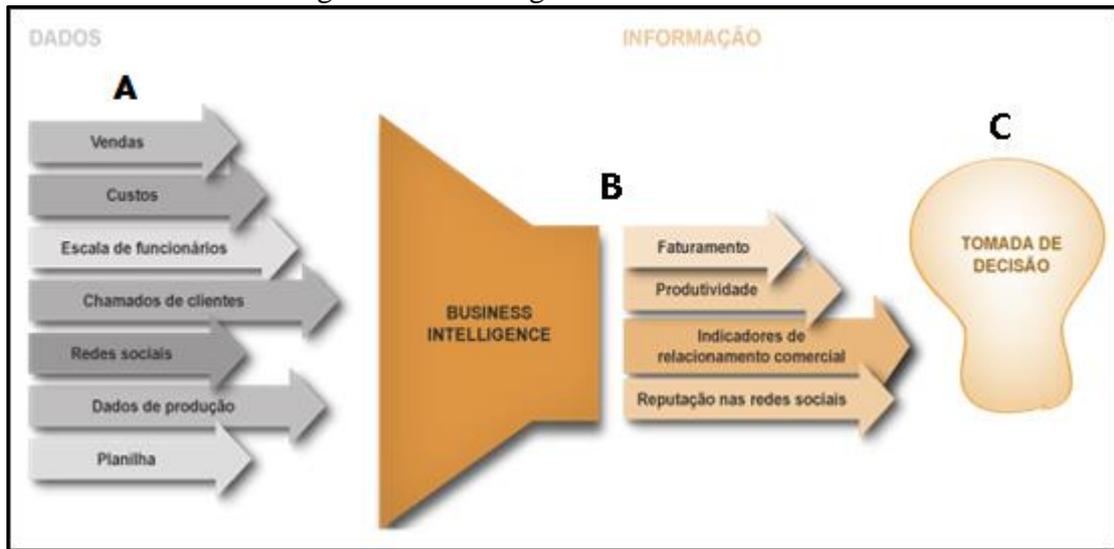
O DW é um depósito de dados digitais que armazena informações detalhadas de forma consolidada. Além disso, o mesmo possibilita que as corporações mantenham o histórico de todas as informações importantes para o negócio, auxiliando na criação de padrões com base nos dados coletados (SIGNIFICADOS, 2017). Nesse sentido, segundo Elias (2014), o DW possui característica e estrutura para suportar análise com grande volume de dados, que sem sombra de dúvidas, são necessárias para a agilidade no processo de tomada de decisão de qualquer corporação. Dessa forma, Piton (2017) coloca que o DW pode até ser confundido com os conceitos do BI, entretanto, conforme Elias (2014), ele apenas complementa o desenvolvimento do BI.

2.2 BUSINESS INTELLIGENCE

O Business Intelligence (BI) pode ser traduzido como Inteligência de Negócios, com o intuito de auxiliar as empresas e instituições a tomarem as decisões de forma inteligente. Dessa forma, o processo do BI consiste em ter fundamentos para evitar decisões precipitadas, sendo segundo Castro (2016), um dos grandes objetivos do BI, ou seja, fornecer informações adequadas e significativas a qualquer momento. Novato (2017) coloca como objetivo principal do BI facilitar a descoberta de informações significativas para as organizações, tendo como origem os dados brutos gerados pelo BI (AUSLAND, 2015).

Segundo Novato (2017), o BI auxilia na compreensão da nova geração de consumidores que já nasceu conectada ao mundo da internet, gerando e consumindo um grande volume de dados. Novato (2017) observa ainda, que esses dados são fundamentais nas tomadas de decisões para as grandes organizações, podendo ser melhor compreendido por sua modelagem Dimensional representada pela Figura 2 na captação (letra A), organização (letra B) e análise de dados (letra C) (KNOW SOLUTIONS, 2017).

Figura 2 – Modelagem Dimensional do BI



Fonte: adaptado de Know Solutions (2017).

Na Figura 2, a captação (letra A) se refere ao meio pelo qual as informações são adquiridas e armazenadas em um DW (seção 2.2), possibilitando dessa forma integrar um grande volume de dados oriundos de outros sistemas da organização como de Enterprise Resource Planning (ERPs) e planilhas (KNOW SOLUTIONS, 2017). A organização (letra B) refere à visualização da informação em diferentes perspectivas denominadas cubo de dados, possibilitando a criação de hierarquias de informação e a combinação de dados na elaboração das análises, simplificando assim o processo de consultas complexas (RP INFO, 2017). Por fim, a análise de dados (letra C) refere à visualização da informação por meio de relatórios e *dashboards* (seção 2.3), auxiliando o entendimento do negócio e facilitando no processo de tomada de decisão (ATOM, 2015).

2.3 DASHBOARD

Segundo Buchsbaum (2012), mesmo com muitas opções no mercado, ainda há empresas que visualizam suas informações de negócio por meio de planilhas. Buchsbaum (2012) ainda observa que esse procedimento ocasiona muitos imprevistos, sejam eles por erro de informação ou pelo tempo de elaboração, já que o processo é feito manualmente. Visando melhorar esses e outros aspectos, o *dashboard* auxilia o usuário dar a devida atenção para o que realmente importa na análise de dados, ou seja, a informação consolidada (PROJECT BUILDER, 2017).

Em geral, o *dashboard* nada mais é que a forma visual de apresentação das informações das informações importantes e até mesmo necessárias para o negócio (ATOM, 2015). A facilidade da descoberta das informações pode ser melhor compreendida pela Figura 3, sendo possível observar o Desempenho dos Vendedores por vistas mais intuitivas e destacadas

(ÁVILA, 2014). A letra A da Figura 3 apresenta as informações em um gráfico do tipo velocímetro destacando o desempenho em uma escala, que vai de ótimo a ruim, representada por cores. A letra B traz informações referentes às vendas por vendedores utilizando um gráfico de barras comum, em que as linhas representam as vendas e as colunas representam os vendedores, facilitando o entendimento dos valores apresentados. Já na letra C é demonstrado os dados que dizem respeito ao volume de vendas, utilizando um gráfico de barras em sua versão 3D, que segue o mesmo princípio de utilização de sua versão comum sendo o dado disponibilizado na coluna na cor azul, o melhor resultado; e o dado disponibilizado na coluna na cor vermelho o pior resultado (ÁVILA, 2014).

Figura 3 – Dashboard



Fonte: adaptado de Ávila (2014).

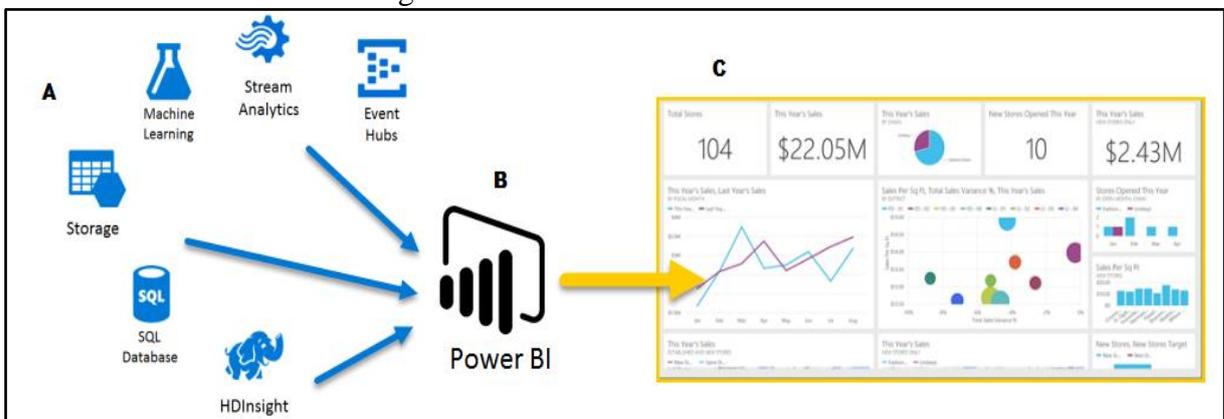
Para Buchsbaum (2012), o BI representa uma visão panorâmica do negócio, com inúmeras possibilidades de interação, servindo como ponto de partida para o processo de tomada de decisão. Enquanto para Ávila (2014), organiza e apresenta as informações de forma mais intuitiva, deixando as informações mais importantes em destaque, para o usuário não perder tempo procurando a informação, e sim as utilizando.

2.4 SELF-SERVICE

O Self-Service BI (SSBI) tem como premissa fornecer ao usuário a capacidade de desenhar e recolher os seus próprios relatórios dentro de uma ferramenta (CASTRO, 2016). Para Rouse (2016), é uma abordagem da análise de dados, que permite aos usuários acessar e trabalhar com dados corporativos, mesmo sem ter um perfil técnico ou analítico. Castro (2016) complementa que o SSBI possibilita aos usuários criar suas próprias pesquisas e análises, tomando possíveis decisões com base no que foi visto nos resultados.

Ramos (2016) observa que ultimamente muito se tem discutido sobre SSBI, fazendo com que relevantes desenvolvedores de software, trabalhem intensamente para disponibilizar ferramentas cada vez mais intuitivas e descomplicadas. Nesses casos, ainda segundo Ramos (2016), o SSBI mostra um ganho significativo na agilidade de obtenção de dados e tomadas de decisões ao fornecer aos gestores uma autossuficiência no desenvolvimento de painéis ou até mesmo tirar dúvidas sobre algum ponto, sem necessidade de intervenção da área de Tecnologia da Informação (TI). Essa agilidade abordada por Ramos (2016) deve-se ao modelo de trabalho adotado pelo SSBI, exemplificado na Figura 4.

Figura 4 – Modelo de Trabalho SSBI



Fonte: adaptado de Power BI (2017).

Na Figura 4, a origem dos dados está representada pela letra A, sendo apresentada pelas diversas fontes de dados que podem ser utilizadas como origem para a ferramenta trabalhar as informações; a ferramenta que integra e manipula os dados estão representados pela letra B, que demonstra a ferramenta que irá fazer a integração, a intercessão e manipulação dos dados provenientes da origem, disponibilizando para o usuário as informações para que ele elabore suas análises; e a análise obtida por meio da ferramenta está representado pela letra C, apresentando a análise em si, feita pelo o usuário com as informações dispostas e trabalhadas pela ferramenta (POWER BI, 2017).

Um cenário interessante para a atuação do SSBI seria de que se uma pessoa está prestes a fechar um negócio, necessitando de uma informação prévia, no modelo tradicional do BI, o usuário teria que abrir uma requisição para área de TI e esperar que ela fique pronta. Contudo, com SSBI, essas informações podem ser consultadas em tempo real (FAROL, 2017). Conforme ressaltado por Rouse (2016), o processo de tomada de decisão se torna menos burocrático e muito mais ágil, entregando todo o potencial de análise na mão do usuário (CASTRO, 2016).

2.5 TRABALHOS CORRELATOS

Neste capítulo são apresentados três trabalhos que apresentam semelhança com os principais objetivos deste trabalho. Na subseção 2.5.1 aborda-se a ferramenta Visual Analytics (SAS INSTITUTE INC, 2016), que possibilita interligar fontes de dados de forma simplificada. Na subseção 2.5.2 apresenta-se a ferramenta Oracle Data Visualization (DV) de Oracle (2016a), permitindo ao usuário explorar dados de maneira simples e descobrir informações que auxiliam na análise de dados. Na subseção 2.5.3 aborda-se a ferramenta Tableau Desktop, que fornece ao usuário uma ferramenta de análise de dados intuitiva e de simples manuseio (TABLEAU SOFTWARE, 2017a). Por fim, a subseção 2.5.4 traz um quadro comparativo entre os trabalhos correlatos citados, comentando as principais diferenças de funcionalidades.

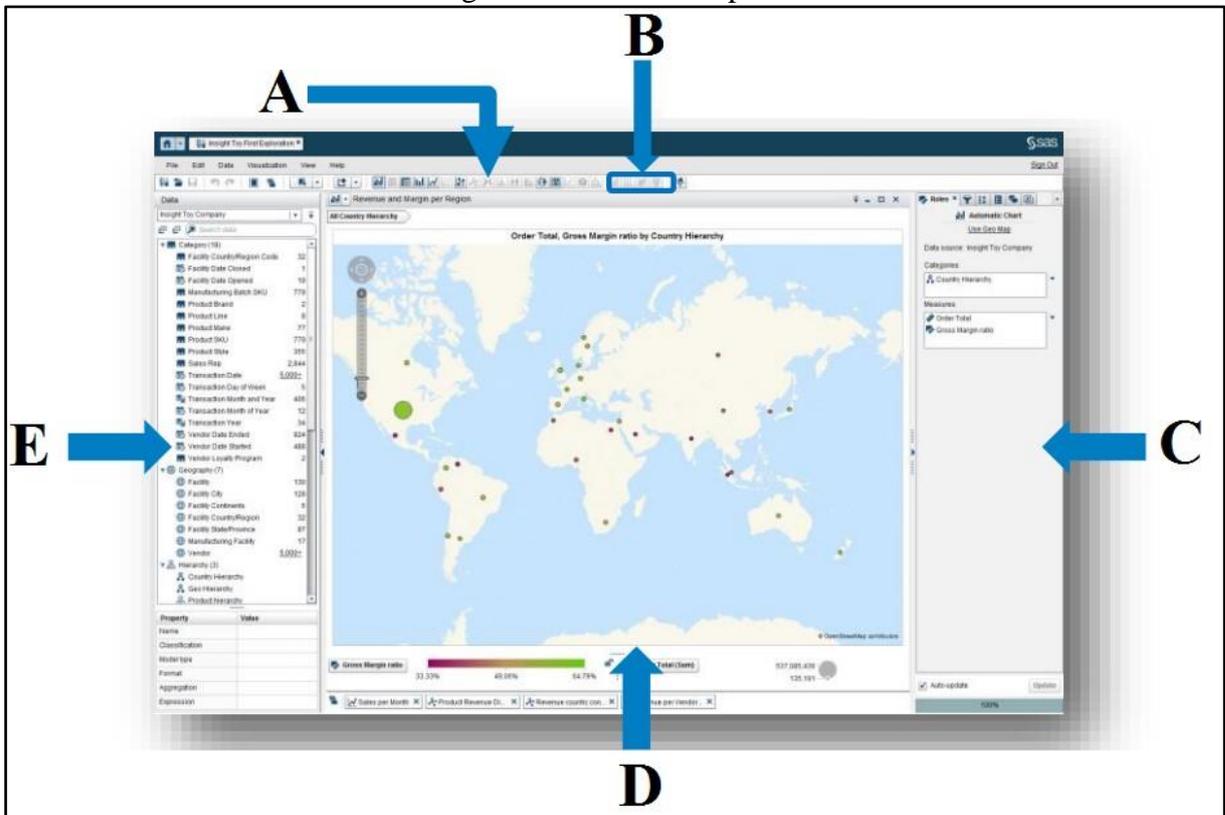
2.5.1 Visual Analytics

O Visual Analytics (VA) é uma ferramenta de análise de dados self-service. Uma das características é não necessitar de um perfil técnico para utilizar o potencial e funcionalidades da ferramenta, pois não é necessário qualquer tipo de codificação para que funcionem. A ferramenta é disponibilizada nas plataformas Web, Android e iOS. Nelas o usuário pode interagir com a análise, adicionar filtros e fazer detalhamento da informação (SAS INSTITUTE INC, 2016). Dentre as funcionalidades da ferramenta está à possibilidade de realizar análises como: árvore de decisões, diagrama de redes, análises de precisão, entre outros. Além disso, independentemente do tamanho do banco de dados, de acordo com SAS Institute Inc (2016), algumas das características disponibilizadas pela ferramenta são (Figura 5):

- a) visualização fácil e rápida dos dados, possibilitando simplificar o descobrimento de oportunidades ocultas;
- b) identificação de relações entre causas e feito, auxiliando na tomada de decisões;
- c) visualização interativa e de fácil compartilhamento;
- d) *autocharting*, escolhe automaticamente o gráfico mais adequado para exibir os dados selecionados;

- e) dados multidimensionais podem ser fatiados e explorados, aplicando filtros em qualquer nível de uma hierarquia;
- f) novas métricas podem ser calculadas e adicionadas em qualquer modo de exibição.

Figura 5 - SAS WorkSpace



Fonte: adaptado de SAS Institute Inc. (2014).

A Figura 5 apresenta um *dashboard* construído pela ferramenta VA (SAS INSTITUTE INC, 2014). A letra A, indica uma caixa de ferramentas que possibilita ao usuário, selecionar a forma de explorar e visualizar os dados. A opção em destaque pela letra B, somente é habilitada se o usuário possuir a ferramenta SAS Visual Statistics. O painel apontado na letra C permite que o usuário altere as propriedades da visualização, filtros de datas, criando comentários para que outros possam visualizar. As letras D e E, indicam à origem e destino das informações desejadas pelo usuário. A letra D, a WorkSpace do usuário e disponibiliza as informações na maneira que desejar, podendo ser gráficos, tabelas ou mapas. A letra E, indica o painel de dados disponibilizados por meio de cubos, possibilitando ao usuário a criação de suas análises (SAS INSTITUTE INC, 2014).

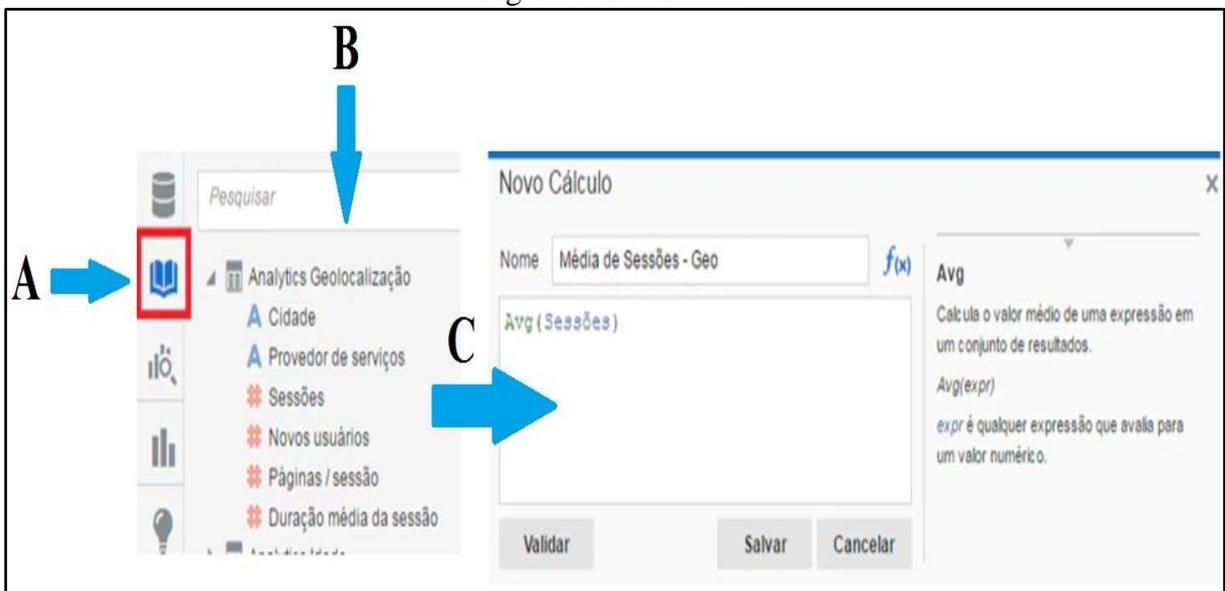
2.5.2 Data Visualization

O Data Visualization (DV) é uma ferramenta que possibilita a apresentação visual de dados abstratos, auxiliando o usuário a encontrar novos padrões ou tendências (ORACLE,

2016c). A ferramenta DV é disponibilizada nas plataformas Web, Android e iOS (ORACLE, 2016a). Uma das funcionalidades da ferramenta (ORACLE, 2016c) é a importação e utilização de metadados, necessitando apenas arrastar alguns atributos para a WorkSpace, sendo modificável pelo usuário. Além disso, algumas das funcionalidades fornecidas pela ferramenta são (Figura 6):

- a) instalar plug-ins disponíveis na loja da Oracle;
- b) criar scripts em linguagem R incorporados a análise;
- c) visualizar interatividade e de fácil compartilhamento;
- d) pesquisar dados por meio de linguagem natural;
- e) pesquisar dados por voz, por meio de aplicativo móvel.

Figura 6- DV Dados



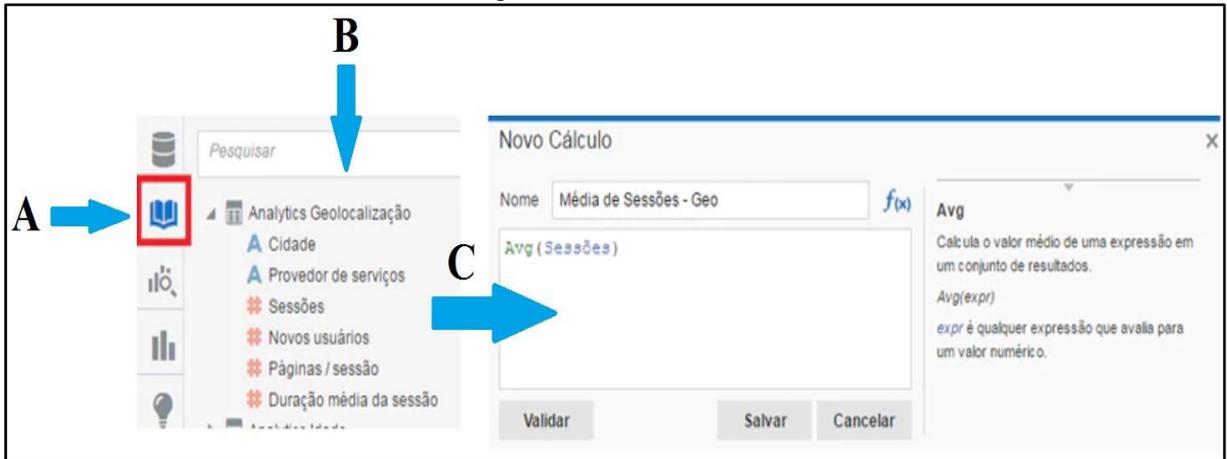
Fonte: adaptado de Oracle (2016b).

A Figura 6 demonstra o cubo de informações selecionado pelo usuário na ferramenta DV. Na letra A, um menu de ícones permite ao usuário navegar de forma simples pelo sistema. Esse exemplo contempla o menu de atributos que disponibiliza informações a serem utilizadas nas análises. A letra B indica as colunas de dados já definidas entre atributos e métricas, para serem utilizados nas visualizações. Por fim, a letra C, com a possibilidade de o usuário utilizar fórmulas de agregação nos atributos disponibilizados em um campo texto. O exemplo apresentado se refere à utilização da fórmula Average (AVG) com a funcionalidade de calcular a média do valor desejado (ORACLE, 2016b).

A Figura 7 demonstra a WorkSpace do DV. Na letra A da referida figura podem ser vistas as opções de gráficos disponibilizados para o usuário, que auxiliará na exploração e visualização dos dados. No exemplo, foi utilizado o gráfico de barras. Na letra B, é indicado o

local no qual será montada a visualização da análise, o usuário poderá arrastar qualquer atributo ou componente disponibilizado pela ferramenta, para a visualização das informações (ORACLE, 2016b).

Figura 7 - DV Gráfico



Fonte: adaptado de Oracle (2016b).

A Figura 8 demonstra a visualização montada pelo usuário a partir dos dados disponibilizados para ele. Na letra A, indica a lista de *insights* de dados já realizado pelo usuário, enquanto na letra B, se localiza os filtros de dados que podem ser utilizados dinamicamente na visualização em questão, sendo que essa funcionalidade pode ser usufruída por qualquer usuário que tenha acesso a ele. Por fim, na letra C, a funcionalidade que permite que o usuário salve a análise em questão, para ser compartilhada para outros usuários ou até mesmo, ser consultada posteriormente (ORACLE, 2016b).

Figura 8 - DV Análise



Fonte: adaptado de Oracle (2016b).

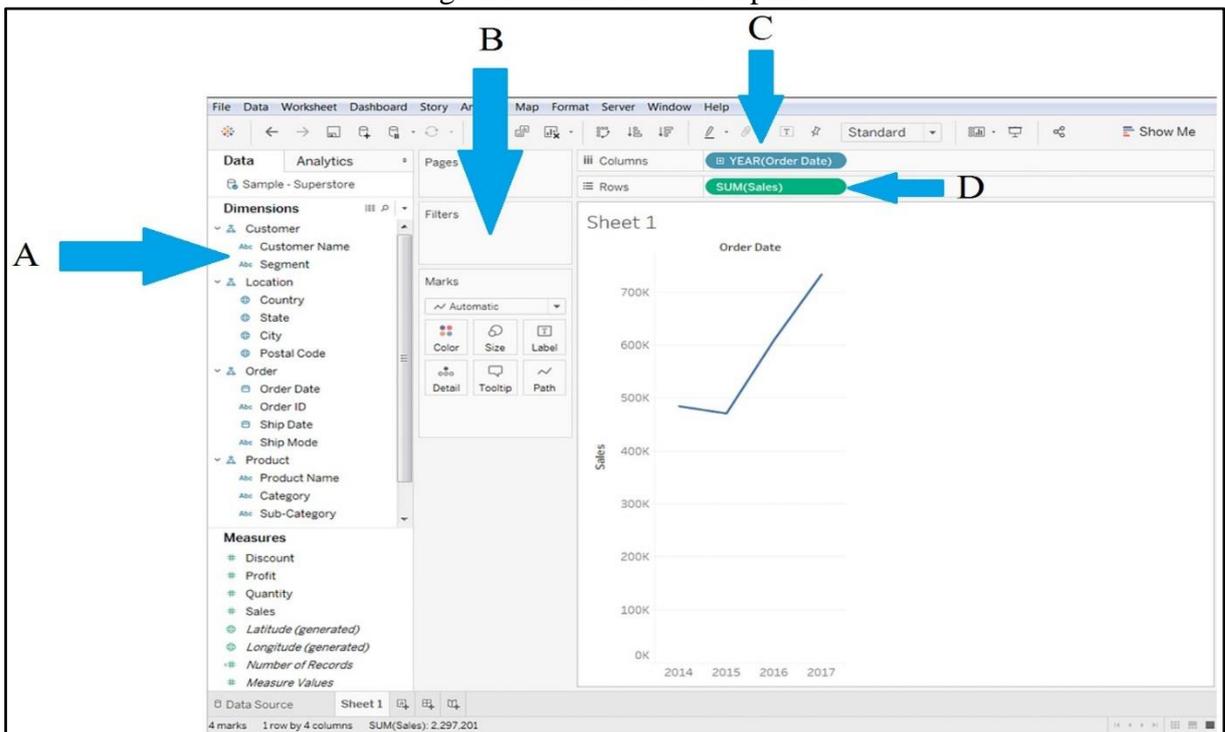
2.5.3 Tableau Desktop

O Tableau Desktop é uma solução de análise de dados, que permite ao usuário que não possui conhecimento técnico, criar visualizações interativas em tempo real (FIVE ACTS, 2017). A ferramenta é disponibilizada apenas na plataforma Desktop. Essa ferramenta possibilita que o usuário possa se conectar com fontes de dados (FIVE ACTS, 2017). Possibilitando a conexão rápida com bases diferentes e com volume de informações significativas. Algumas das funcionalidades fornecidas pela ferramenta são:

- visualização fácil e rápida dos dados;
- conexão direta na base de dados, tornando a análise atualizada em tempo real;
- fácil compartilhamento de informações com outros usuários;
- visualização de dados off-line, com o recurso *in-memory*.

A Figura 9 demonstra a WorkSpace da ferramenta em questão, na letra A, está localizada as tabelas com seus atributos e métricas do cubo selecionado pelo o usuário para análise. A letra B está indicando os filtros a serem aplicados na análise, gráficos disponíveis para a visualização das informações e lista de páginas da análise, que está sendo elaborada pelo o usuário. Nas letras C e D estão localizados os atributos a serem exibidos pelo gráfico selecionado previamente pelo usuário; a letra C representa a coluna do gráfico e a letra D, a sua linha, apresentando um valor mensurável (TABLEAU SOFTWARE, 2017b).

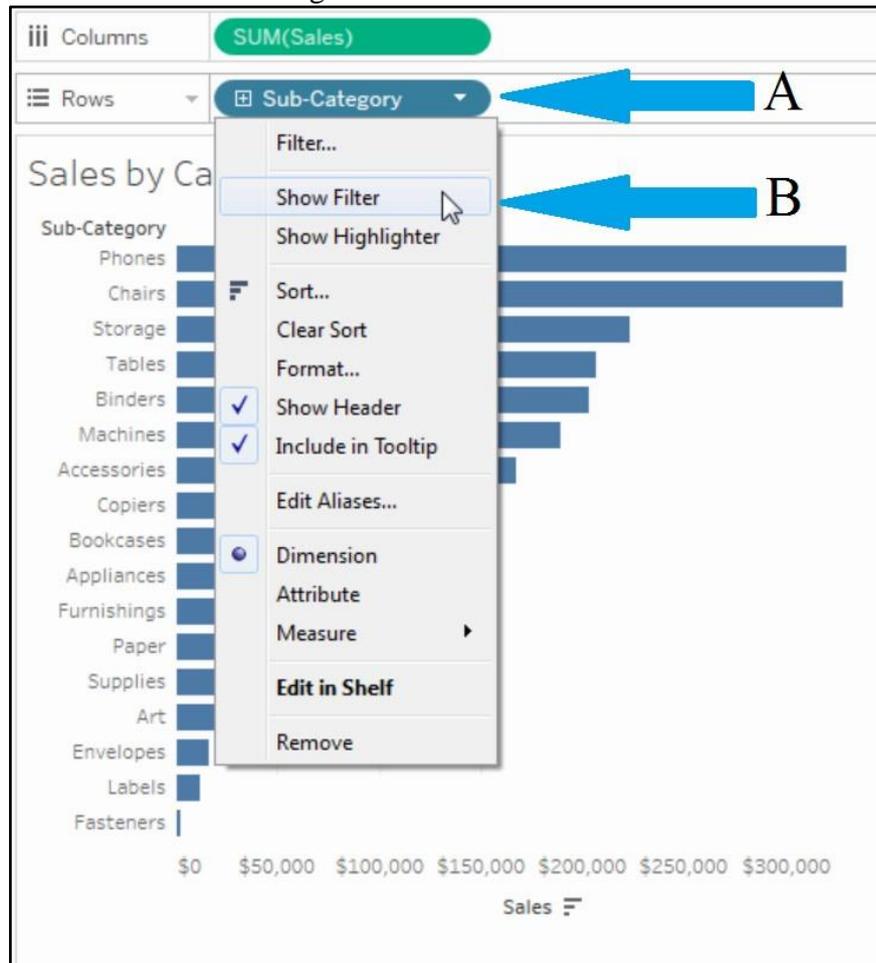
Figura 9 - Tableau WorkSpace



Fonte: adaptado de Tableau Software (2017b).

Na Figura 10 é demonstrada a aplicação de filtros dinâmicos nos gráficos gerados pelo usuário. A letra A indica os *labels* do gráfico da análise, se o usuário clicar nele irá aparecer o menu com as opções de personalizações para tal atributo. A letra B indica a opção de mostrar filtros, que no caso, irá habilitar uma lista de filtros referente ao atributo (TABLEAU SOFTWARE, 2017c), ilustrado melhor na Figura 11.

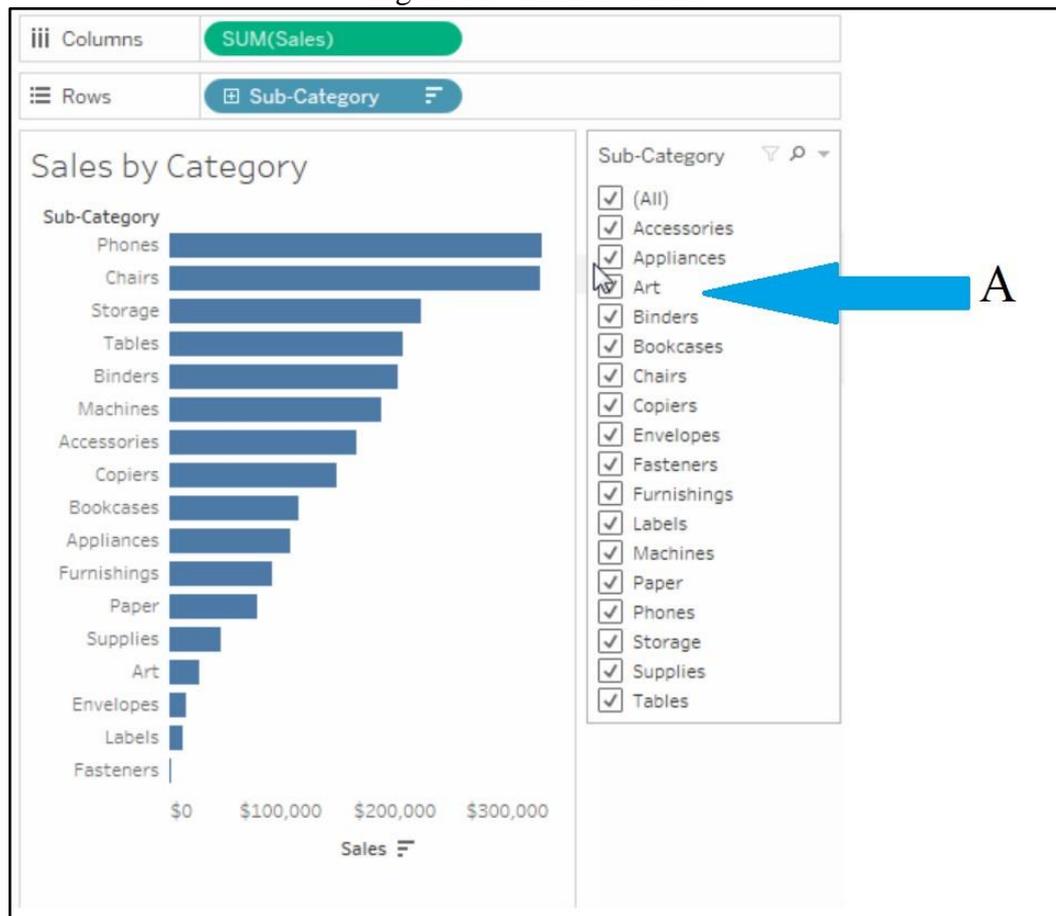
Figura 10 - Tableau Filtro



Fonte: adaptada de Tableau Software (2017c).

A Figura 11 apresenta uma lista de filtros apontados pela letra A, conforme mencionado anteriormente na Figura 10. Essa lista é apresentada em forma de caixa de seleção, em que o usuário pode desmarcar, com o intuito de ocultá-la do gráfico, ou marcar para que a mesma apareça novamente a qualquer momento, de forma dinâmica e rápida (TABLEAU SOFTWARE, 2017c).

Figura 11 - Tableau Filtro



Fonte: adaptado de Tableau Software (2017c).

2.5.4 Comparação entre os trabalhos correlatos

No Quadro 1 é apresentado um comparativo entre os trabalhos correlatos, elencando algumas de suas funcionalidades, dadas como principais recursos das ferramentas dispostas. O referido quadro tem o objetivo de indicar as semelhanças e as diferenças entre os trabalhos relacionados, dizendo respeito a essas funcionalidades, sendo, que as linhas representam as características e as colunas os trabalhos correlatos.

Quadro 1 - Comparativo entre os trabalhos correlatos

Características	Correlatos		
	Tableau Desktop Tableau Software (2017a)	Data Visualization Oracle (2016)	Visual Analytics SAS Institute Inc (2016)
Compartilhamento de Painéis	✓	✓	✓
Painéis Interativos	✓	✓	✓
Conexão com várias fontes de dados	✓	✓	✓
Autocharting	X	X	✓
Pesquisa em linguagem natural	X	✓	X
Recurso <i>in-memory</i>	✓	X	X
Plataformas (Android/iOS/Web)	X	✓	✓

Fonte: elaborado pelo autor.

Conforme demonstrado no Quadro 1, os trabalhos correlatos trazem as ferramentas Tableau Desktop de Tableau Software (2017a); Data Visualization de Oracle (2016) e Visual Analytics de SAS Institute Inc (2016), que se destacam por possibilitar o compartilhamento de painéis desenvolvido pelo usuário. Essa funcionalidade permite que o usuário forneça de forma fácil, o painel que ele desenvolveu, para outros usuários visualizarem. Além dessa característica, as ferramentas correlatas ao estudo possuem em comum os painéis interativos. Dessa forma, possibilita ao usuário filtrar dados no painel de forma rápida e dinâmica, sem a necessidade de desenvolver um painel específico para tal situação. Por fim, a última característica em comum entre os três trabalhos relacionados, diz respeito à conexão com várias fontes de dados, possibilitando ao usuário, buscar seus dados tanto de um banco de dados Oracle como de uma planilha em Excel.

Relacionado ao *Autocharting*, apenas a ferramenta Visual Analytics oferece a opção de sugerir ao usuário no momento da construção da análise a melhor forma de visualizar o gráfico, com base no tipo de dados utilizado por ele. Referente à pesquisa realizada em linguagem natural, a ferramenta Data Visualization, possibilita ao usuário digitar um texto ou por meio de voz, no caso de dispositivo móvel, as informações desejadas, em que a ferramenta ficará responsável de processar a informação.

E referente ao *in-memory*, a ferramenta Tableau Desktop permite que o usuário consulte os dados mesmo estando off-line, devido a essa característica armazena dados em memória, para uso posterior. Por fim, tem-se as plataformas em que as ferramentas são disponibilizadas para o acesso dos usuários, no caso, apenas Data Visualization e Visual Analytics possuem aplicativo tanto para as plataformas Web quanto para as plataformas móveis, Android e iOS. Já o Tableau Desktop, como o próprio nome já diz, possui aplicativo apenas para a versão Desktop, tendo outras ferramentas para quem deseja explorar os dados em outras plataformas.

3 DESENVOLVIMENTO

Este capítulo tem como objetivo detalhar as principais etapas relacionadas ao processo de desenvolvimento da aplicação SELBST desenvolvida. A seção 3.1 aborda o levantamento de informações da aplicação. Na seção 3.2 é apresentada a especificação da aplicação. Na seção 3.3 é demonstrada a implementação do sistema. Por fim, a seção 3.4 traz a correlação dos trabalhos correlatos e a aplicação SELBST, bem como traz a avaliação de usabilidade e apresenta os seus resultados e as discussões obtidas por meio do questionário aplicado junto aos sujeitos participantes da avaliação.

3.1 LEVANTAMENTO DE INFORMAÇÕES

Este trabalho apresenta o desenvolvimento da aplicação SELBST, que visa auxiliar as tomadas de decisões de um empreendedor quanto ao seu negócio, fornecendo-o informações minuciosas referente aos produtos e serviços consumidos por ele na instituição financeira Viacredi. Por parte das instituições financeiras, a aplicação SELBST é vista como um diferencial no mercado perante seus concorrentes, podendo atrair o público empreendedor de maneira mais espontânea. Ademais, poderá ser utilizada como modo de divulgar seu portfólio de produtos e serviços de uma forma mais direta.

A aplicação em si possui basicamente dois atores que atuam diretamente no sistema, sendo um deles mais técnico, representado pelo `analista de sistema` e o outro com perfil voltado ao negócio, representado pelo `usuário`. O analista atuará no sistema primeiramente na modelagem de dados, responsável por buscar as informações de diversas fontes de dados e centraliza-las em um único local, realizando um breve tratamento. Esta abordagem se refere ao tema do DW, abordado na seção 2.1. Enquanto, o passo seguinte diz respeito ao BI propriamente dito (seção 2.2), ou seja, na criação de regras e cubos de informações, com base na modelagem descrita no passo anterior, que serão disponibilizados para o `usuário final` poder fazer seus `dashboards` e realizar suas `consultas`.

O `usuário` atuará apenas no consumo dessas informações preparadas e fornecidas pelo `analista de sistema` por meio da aplicação, estando apto a criar `dashboards` com variados tipos de gráficos e tabelas (seção 2.3), bem como cruzar esses dados da maneira que achar mais pertinente ao seu negócio. Ademais, principalmente fazer uso dessa informação coletada para auxiliar nas tomadas de decisões do dia a dia e também na gestão do seu empreendimento (seção 2.4), possibilitando ganhar uma vantagem competitiva perante a seus concorrentes e consequentemente se destacar mais no mercado.

3.1.1 Requisitos

A seguir são detalhados os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) atendidos pelo sistema: O sistema deverá:

- a) permitir que o usuário execute a ação *drag and drop* para utilizar os atributos do cubo selecionado, para dentro da WorkSpace (Requisito Funcional - RF);
- b) destacar os cubos selecionados pelo usuário (RF);
- c) permitir que o usuário aplique fórmulas nas informações de métricas (RF);
- d) sugerir gráficos de apresentação para o usuário utilizar em suas análises (RF);
- e) permitir que o analista de sistema cadastre um usuário (RF);
- f) permitir que o analista de sistema cadastre uma regra (RF);
- g) permitir que o processo de atualização de dados seja disparado por uma requisição (RF);
- h) permitir que o usuário mude a forma de visualização das informações (RF);
- i) permitir que o usuário se autentique na aplicação (RF);
- j) propiciar a visualização dos dados fornecidos da instituição em estudo (Requisito Não Funcional - RNF);
- k) fazer integração com Application Programming Interface (API) para consulta de dados da instituição em estudo (RNF);
- l) utilizar o banco de dados do Firebase (RNF);
- m) utilizar a linguagem React no desenvolvimento do front-end (RNF);
- n) utilizar o Firebase para fazer o cadastro e controle de usuários (RNF);
- o) utilizar o componente Plotly para disponibilizar o WorkSpace na aplicação (RNF);
- p) utilizar a linguagem JavaScript com NodeJS no desenvolvimento do *back-end* (RNF).

3.2 ESPECIFICAÇÃO

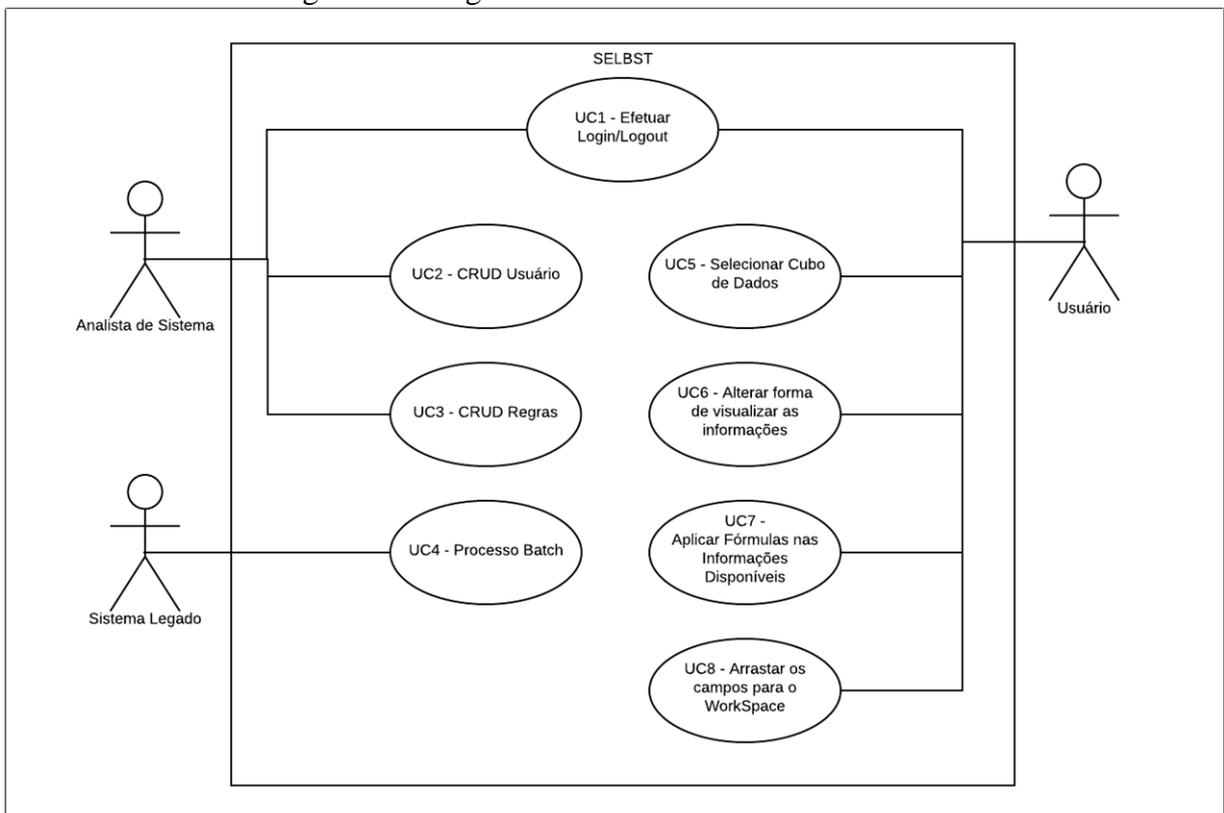
Esta seção tem como objetivo detalhar brevemente os principais diagramas, feitos com base no padrão Unified Modeling Language (UML), ajudando a compreender melhor o que compõem o desenvolvimento da aplicação. A subseção 3.2.1 aborda o diagrama de casos de uso do sistema e as funções com base nos atores. Na subseção 3.2.2 é apresentado a matriz de rastreabilidade entre os requisitos funcionais e os casos de uso. Na subseção 3.2.3 é abordado o diagrama de atividade da criação de um *dashboard* junto com a explicação de cada passo. A subseção 3.2.4 demonstra o diagrama de componentes da aplicação e os seus relacionamentos. A subseção 3.2.5 apresenta o diagrama de arquitetura da aplicação e a divisão de componentes

por frente do sistema. Por último, a subsecção 3.2.6 aborda o diagrama de *deploy* e a comunicação entre os servidores e os componentes.

3.2.1 Diagrama de casos de uso

A Figura 12 exhibe o Diagrama de Casos de Uso (DCU) do SELBST para os atores Sistema Legado, Analista de Sistema e Usuário. O Caso de Uso (UC) UC4 - Processo Batch, que com base nas regras criadas pelo ator Analista de Sistema no UC3 - CRUD Regras busca informações do sistema legado e insere no Firebase Database. No caso de uso UC5 - Selecionar Cubo de Dados permite que o ator Usuário, com base nas informações disponibilizadas pelo UC3 - CRUD Regras, selecione o grupo de informações nas quais ele deseja realizar análises. Já nos casos de usos UC6 - Alterar forma de visualizar as informações, UC7 - Aplicar formulas nas informações disponíveis e UC8 - Arrastar os campos para Workspace permite que o ator Usuário trabalhe com as informações selecionadas aplicando filtros, fórmulas e gráficos diferentes para que atenda sua demanda. O detalhamento dos casos de usos está disponível no Apêndice A.

Figura 12- Diagrama de casos de uso do SELBST



Fonte: elaborado pelo autor.

3.2.2 Matriz de rastreabilidade dos RFs X UC

O Quadro 2 apresenta a matriz de rastreabilidade entre os requisitos especificados na seção 3.1.1 e os casos de uso apresentados na Figura 12.

Quadro 2– Matriz de rastreabilidade de Requisitos Funcionais X UC

REQUISITOS FUNCIONAIS	UC
RF a: permitir que o usuário execute a ação drag and drop para utilizar os atributos do cubo selecionado, para dentro da WorkSpace.	UC8
RF b: destacar os cubos selecionado pelo usuário.	UC5
RF c: permitir que o usuário aplique fórmulas nas informações de métricas.	UC7
RF e: permitir que o analista de sistema cadastre um usuário.	UC2
RF f: permitir que o analista de sistema cadastre uma regra.	UC3
RF g: permitir que o processo de atualização de dados seja disparado por uma requisição.	UC4
RF h: permitir que o usuário mude a forma de visualização das informações	UC6
RF i: permitir que o usuário se autentique na aplicação	UC1

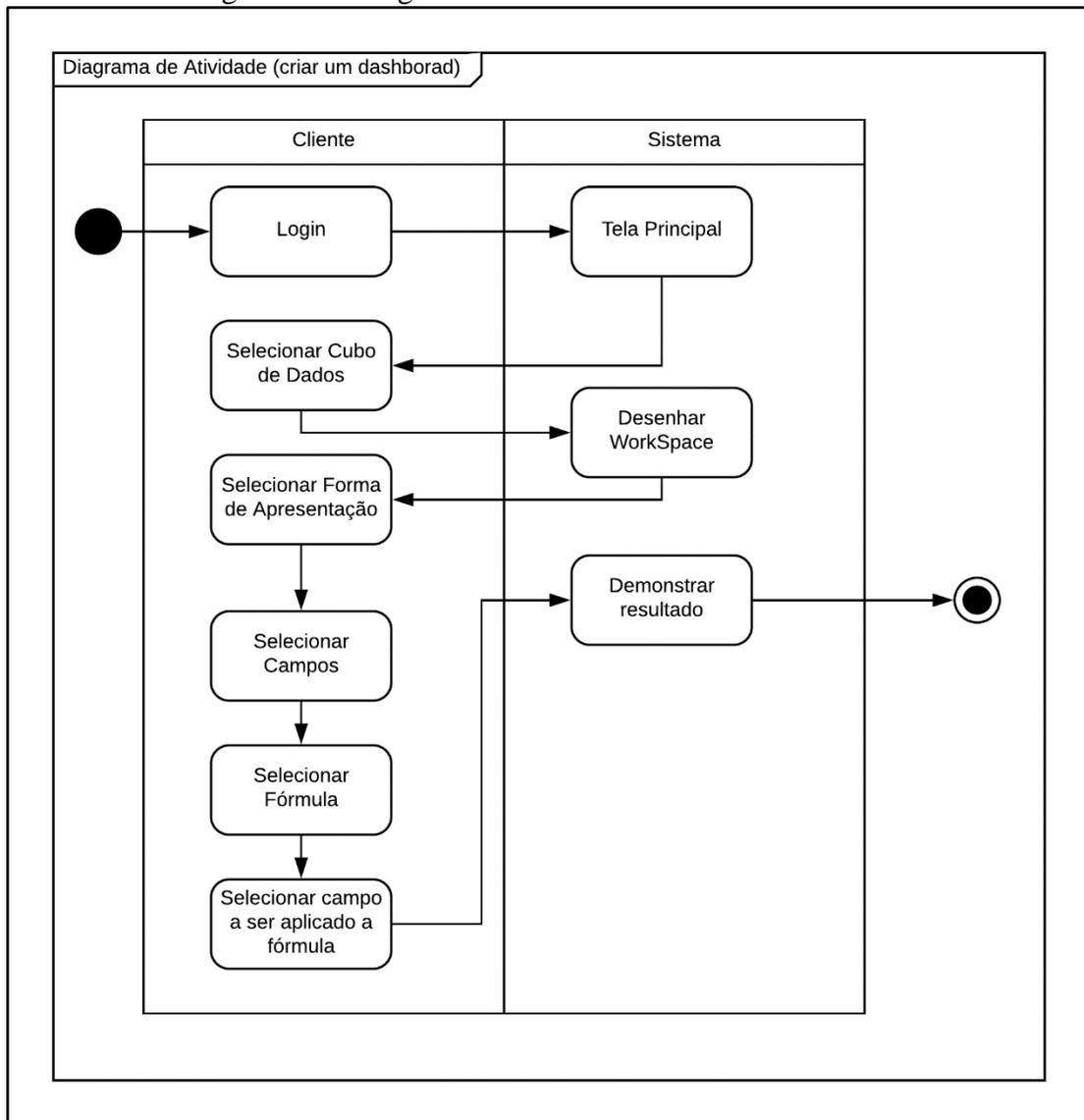
Fonte: elaborado pelo autor.

3.2.3 Digrama de atividade para criar dashboard

A Figura 13 apresenta um diagrama de atividades, no qual exemplifica passo a passo as ações SELBST, que devem ser tomadas para que um usuário elabore um dashboard na aplicação. O primeiro passo a ser efetuado pelo usuário, após fazer o login, é realizado a atividade de selecionar o cubo de dados no qual ele deseja trabalhar, com isso o sistema irá desenhar a WorkSpace e disponibilizará as informações referente ao cubo selecionado.

No passo seguinte o usuário deve selecionar o tipo de gráfico ou tabela em que as informações devem ser apresentadas, ou seja, demonstrar resultados, bem como selecionar fórmulas a serem aplicadas nos campos de métricas. Posteriormente, a aplicação demonstra o resultado no dashboard para que o usuário tenha a liberdade de trabalhar da forma que lhe for mais apropriado.

Figura 13 - Diagrama de atividades: criar dashboard



Fonte: elaborado pelo autor.

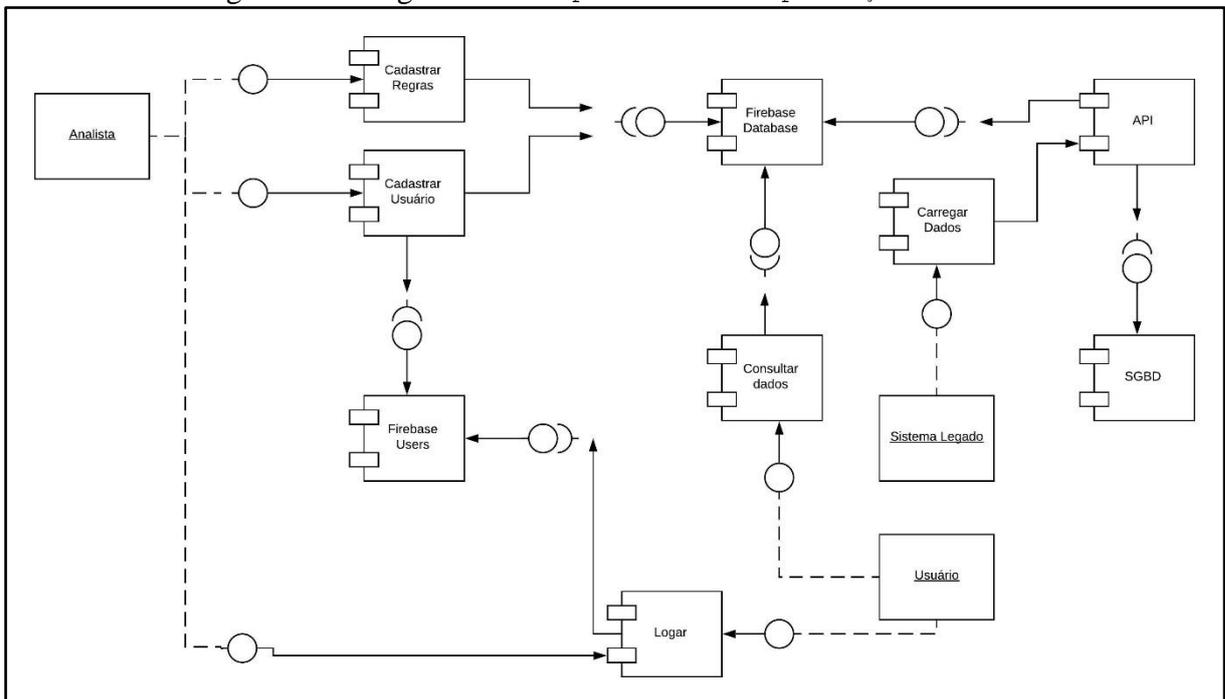
3.2.4 Diagrama de componentes

A Figura 14 demonstra o diagrama de componentes da aplicação expondo o relacionamento de cada elemento do sistema um para com o outro, separados pelas ações que podem ser realizadas pelo Analista, pelo Usuário e pelo Sistema Legado. Primeiramente, demonstrando o sistema pela visão do Analista, possui duas ações principais, que são Cadastrar Regras e Cadastrar Usuário. Quando selecionado a opção Cadastrar Regras, o sistema se comunica diretamente com o módulo do Firebase Database, persistindo as informações preenchidas no formulário. Quando selecionado a opção Cadastrar Usuário, o sistema além de se comunicar com o módulo do Firebase Database para persistir informações referente ao seu perfil, também se comunica com o módulo Firebase Users para salvar as suas credenciais e dar permissão de acesso ao sistema.

Logo em seguida, explorando o sistema pela visão do *Usuário*, existem duas ações por ele definidas, *Consultar dados* e *Logar*, sendo esta última compartilhada com o *Analista*. Quando selecionado a opção *Logar* o sistema se comunica com o módulo do *Firestore Users*, verificando as credencias inseridas e se bem-sucedido permite que ele acesse a aplicação. Quando selecionado a opção *Consultar dados* o sistema se comunica com o módulo *Firestore Database*, buscando as informações registradas, com base nas credencias logadas e as tratando para serem apresentadas ao *Usuário* na aplicação.

Por último, demonstrando o sistema pela visão do *Sistema Legado*, possui apenas uma ação, que é a de *Carregar Dados*, sendo esta uma das principais ações do sistema. Esta ação consiste em carregar as informações do banco de dados relacional do cliente, com base nas regras cadastradas, para serem consumidas pela aplicação por meio do *Firestore Database*. Essa comunicação é feita por meio do módulo *API*, que por sua vez se comunica com o *SGBD* do cliente, buscando as informações necessárias e retornando a *API*, permitindo se conectar ao módulo *Firestore Database* e persistindo os dados resgatados.

Figura 14 - Diagrama de componentes da aplicação SELBST



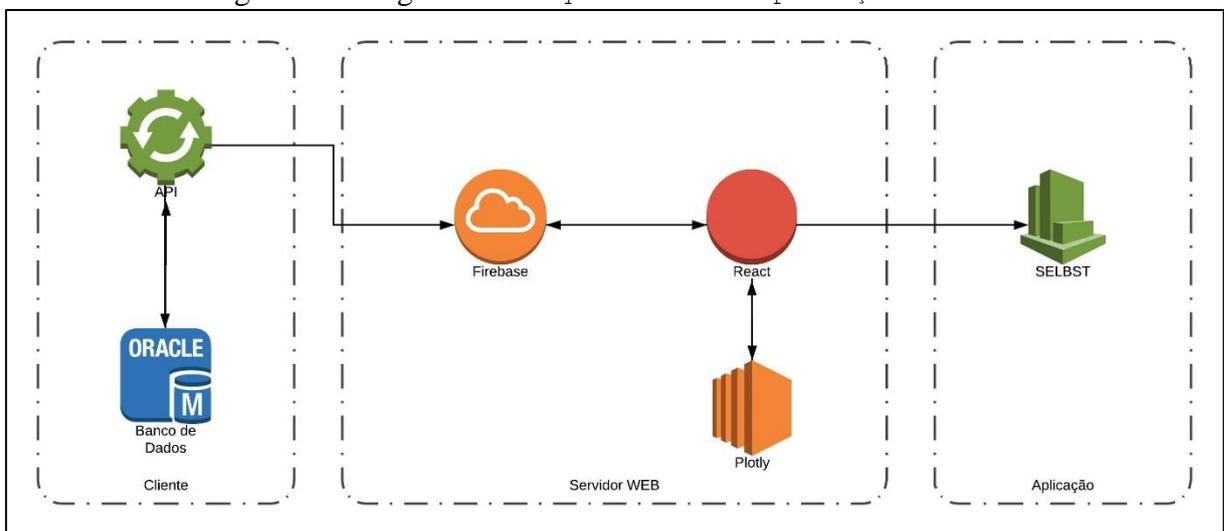
Fonte: elaborado pelo autor.

3.2.5 Diagrama de Arquitetura

Na Figura 15 é apresentado o diagrama da arquitetura da aplicação SELBST, demonstrando os componentes do sistema em suas três principais frentes, que são a do Cliente, a do Servidor Web e a da Aplicação. Os componentes Banco de Dados e API,

que tem sua correlação explicada na subseção 3.2.4, local que são hospedados no ambiente de preferência do Cliente, sendo deste, a responsabilidade de manter ambos acessíveis e operante para um funcionamento perfeito da aplicação desenvolvida. Já os componentes Firebase, React e Plotly ficam hospedados em um Servidor Web do próprio Firebase, isentando o Cliente de responsabilidades. As correlações desses componentes se baseiam no módulo React, que busca informações contidas no Firebase e repassa para o módulo Plotly; que por sua vez trata essas informações retornando-as para o React, ficando responsável por demonstrar o resultado para o usuário final na Aplicação.

Figura 15- Diagrama de Arquitetura da aplicação SELBST



Fonte: elaborado pelo autor.

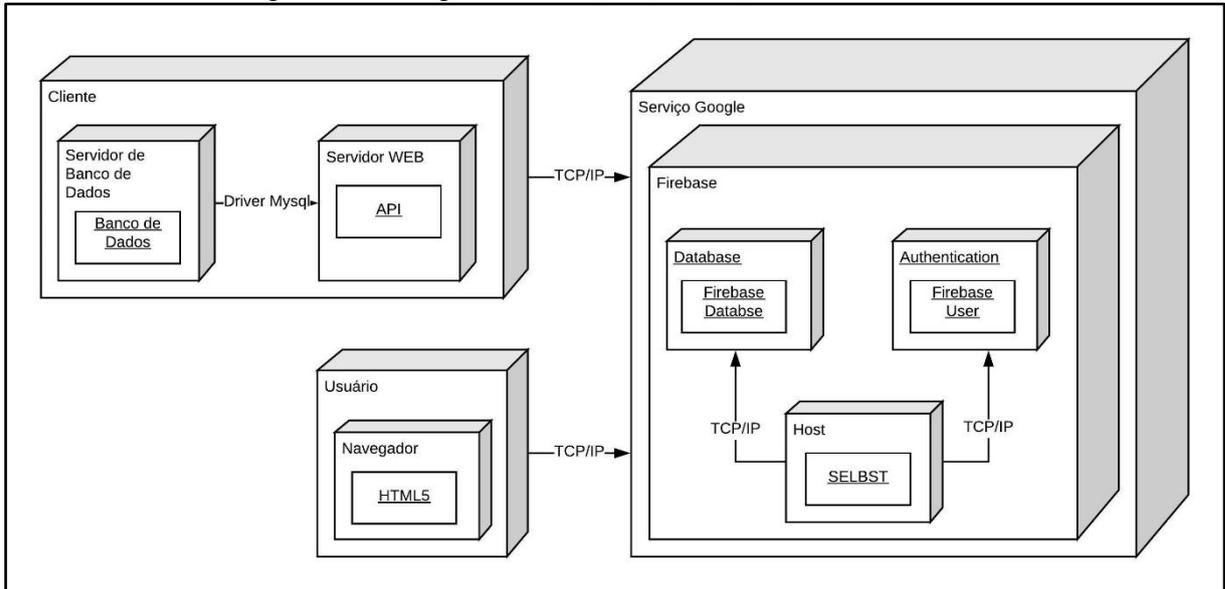
3.2.6 Diagrama de Deploy

A Figura 16 representa o diagrama de deploy da aplicação SELBST, no qual demonstra a comunicação entre os ambientes em que o sistema se encontra, sendo eles o do Cliente, o do Serviço Google e o do Usuário, além de demonstrar a comunicação entre os componentes dentro de um ambiente único. No ambiente do Cliente encontra-se os componentes Banco de Dados, localizado dentro do Servidor de Banco de Dados e a API localizada dentro do Servidor WEB, sendo que a comunicação entre esses dois servidores é feita por meio do Driver Mysql.

Enquanto no ambiente do Serviço Google, no qual se comunica com o ambiente do Cliente por meio do protocolo TCP/IP, têm-se os componentes Firebase Database, localizado dentro de Database, Firebase User localizado dentro de Authentication e a própria aplicação SELBST localizada dentro de Host, comunicando-se com os outros dois servidores por meio do protocolo TCP/IP. Já no ambiente do Usuário, tem-se o componente

HTML5 dentro do serviço do Navegador, comunicando-se com o Serviço Google também por meio do protocolo TCP/IP.

Figura 16 - Diagrama de Deploy da aplicação SELBST



Fonte: elaborado pelo autor.

3.3 IMPLEMENTAÇÃO

Este capítulo tem como objetivo detalhar a codificação das principais rotinas da aplicação de estudo, assim como explicar suas funcionalidades disponibilizadas em tela para a interação do usuário com a aplicação. A subseção 3.3.1 aborda as técnicas e ferramentas utilizadas durante todo o processo de desenvolvimento. Na subseção 3.3.2 é apresentado os códigos referentes ao *back-end* da aplicação em si. Por fim, a subseção 3.3.3 exemplifica a operacionalidade do sistema, demonstrando suas telas e funcionalidades.

3.3.1 Técnicas e ferramentas utilizadas

Esta subseção traz as técnicas e ferramentas utilizadas no desenvolvimento da aplicação SELBST, tendo sido descrita de uma maneira mais específica nos diagramas de componentes (subseção 3.2.4), arquitetura (subseção 3.2.5) e *deploy* (subseção 3.2.6), em que são elencados os componentes que fazem parte do sistema, bem como demonstrando suas correlações e a forma que se comunicam entre eles.

Para auxiliar o desenvolvimento da aplicação SELBST, foi utilizado o padrão UML com o intuito de modelar e compreender a amplitude do sistema, nos quais são descritos de uma maneira mais ampla na subseção 3.2. Na confecção do código fonte foi utilizada o Ambiente

de Desenvolvimento Integrado (IDE) Visual Code¹, de licença Freeware, disponibilizado pela Microsoft, auxiliando o desenvolvedor tanto pela sua interface minimalista e intuitiva quanto pela quantidade de *plug-ins* disponíveis em sua comunidade. Enquanto para hospedar o código da aplicação desenvolvida foi utilizado o sistema de hospedagem Bitbucket², disponibilizado pela Atlassian, que fornece um plano de hospedagem para manter o repositório privado de maneira gratuita.

Referente a arquitetura do sistema foi adotada a estrutura *cliente-servidor*, tendo a parte do *servidor* hospedada em dois locais diferentes, uma localizada no *servidor do cliente*, comunicando-se diretamente com o banco de dados *Firestore Database*; e outra hospedada no *Firestore* que se comunica com o usuário. A API, hospedada no *servidor do cliente* foi desenvolvida na linguagem de programação *JavaScript* com *NodeJs*³ e utilizando o *framework Express*, que visa facilitar o gerenciamento das rotas de acesso à aplicação. Já comunicação entre o banco de dados do cliente e o sistema foi realizada pelo *driver MySQL*, diferentemente da comunicação entre a API e o *Firestore*⁴, banco da aplicação, que é realizada pela *requisição http*.

A aplicação acessada pelos usuários foi desenvolvida com base no ecossistema do *React*⁵, com suas variadas bibliotecas e *frameworks*, visando agilizar e facilitar o desenvolvimento da aplicação, sendo utilizada na elaboração do *front-end* e para intermediar a comunicação entre os demais componentes. Um deles é o *framework Plotly*, que é responsável por confeccionar o *dashboard* utilizado pelo usuário para criar e gerenciar suas análises. Ademais, foram utilizados componente relacionados ao *Firestore*, sendo eles o *database*, em que ficam guardados as informações da aplicação e o *authentication* para realizar o controle de usuários.

3.3.2 Implementação do desenvolvimento

Para demonstrar a implementação da aplicação, o Quadro 3 apresentado o arquivo em que é criada a rota de acesso a API, em que nas linhas 1 a 3 é destinado a importação

¹ IDE Visual Code: Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 25 jun. 2018.

² Bitbucket: Disponível em: <<https://bitbucket.org/>>. Acesso em: 25 jun. 2018.

³ Nodejs: Disponível em: <<https://nodejs.org/en/>>. Acesso em: 25 jun. 2018.

⁴ Firestore: Disponível em: <<https://firebase.google.com/>>. Acesso em: 25 jun. 2018.

⁵ React: Disponível em: <<https://reactjs.org/>>. Acesso em: 25 jun. 2018.

de bibliotecas auxiliares e nas linhas 5 a 7 é exibido a criação da rota em si, que no caso está chamando a rotina de atualização de dados nomeada de `updateDatabase`.

Quadro 3 – Codificação das rotinas Rotas

```
1  const express = require('express');
2  const router = express.Router();
3  const routerController = require('../controller/routerController.js');
4
5  router.get('/firebase/process', (req, res) => {
6    |   routerController.updateDatabase();
7  })
8
9  module.exports = router;
```

Fonte: elaborado pelo autor.

No Quadro 4 é demonstrada a lógica das rotinas complementares da função `updateDatabase`, em que nas linhas 6 e 7 é criada a rotina `getClientInformation`, responsável por buscar as informações no banco de dados do cliente com base na regra delimitado pelo parâmetro. Nas linhas 14 a 20 é criada a rotina `setClientRulesApplication`, responsável por inserir as informações passadas por parâmetro no banco de dados da aplicação. A rotina `getChildsApplication`, entre as linhas 22 e 28, responsável por buscar as informações contidas no banco de dados da aplicação. E nas linhas 30 a 32 é criada a rotina `getDatabaseReference`, responsável por identificar a referência do banco de dados da aplicação a ser trabalhada na API.

Quadro 4 – Codificação das rotinas Auxiliares

```

5  module.exports = {
6    getClientInformation: function(query) {
7      var df = Q.defer();
8      dbConnection.runQuery(query).then(function(info) {
9        df.resolve(info);
10     });
11     return df.promise;
12   },
13
14   setClientRulesApplication: function(rule, ref) {
15     var df = Q.defer();
16     dbFirebaseConnection.insertQuery(rule, ref).then(function(value) {
17       df.resolve(value);
18     });
19     return df.promise;
20   },
21
22   getChildsApplication: function(ref) {
23     var df = Q.defer();
24     dbFirebaseConnection.selectChild(ref).then(function(value) {
25       df.resolve(value);
26     });
27     return df.promise;
28   },
29
30   getDatabaseReference(reference) {
31     return dbFirebaseConnection.getDatabaseReference(reference);
32   },
33
34   updateDatabse() {
35     this.getChildsApplication(this.getDatabaseReference('RULES')).then(

```

Fonte: elaborado pelo autor.

No Quadro 5 é demonstrada a rotina `updateDatabse`, sendo está a rotina principal da API, responsável por buscar os dados do banco de dados do cliente e inseri-los no banco de dados da aplicação. O processo é iniciado na chamada da rotina `getChildsApplication`, linha 35, utilizando a referência `RULES`, que é responsável por buscar as regras cadastradas no banco de dados da aplicação. Posteriormente, na linha 38, é chamado a rotina `getChildsApplication` novamente, passando a referência `USERS`, para buscar as informações referente aos usuários cadastrados na aplicação. Da linha 40 a 56 é feito um laço percorrendo todos os usuários encontrados no passo anterior, buscando as informações `CDCOOPER` e `NRDCONTA`, nas linhas 43 e 44, para serem consideradas nas regras, linha 45, buscadas pela rotina executada na linha 35. Por último, na linha 46 é chamada a rotina `getClientInformation`, que busca as informações do cliente com base na regra já tratada anteriormente e insere no banco de dados da aplicação utilizando a rotina `setClientRulesApplication` na linha 48.

Quadro 5 – Codificação da rotina updateDatabase

```

34  updateDatabase() {
35      this.getChildsApplication(this.getDatabaseReference('RULES')).then(
36          rules => {
37              Object.keys(rules).forEach(async rule => {
38                  this.getChildsApplication(this.getDatabaseReference('USERS')).then(
39                      async users => {
40                          Object.keys(users).forEach(async user => {
41                              this.getChildsApplication(this.getDatabaseReference('USERS/' + user)).then(
42                                  async key => {
43                                      const cdcooper = key.cdcooper;
44                                      const nrdconta = key.nrdconta;
45                                      const rul = eval('`'+rules[rule]+'`');
46                                      await this.getClientInformation(rul).then(
47                                          async results => {
48                                              await this.setClientRulesApplication(
49                                                  results,
50                                                  this.getDatabaseReference('USERS/' + user + '/' + rule)
51                                              );
52                                          }
53                                  );
54                              });
55                          });
56                      }
57                  );
58              });
59          }
60      );
61  }
62  };

```

Fonte: elaborado pelo autor.

Seguindo para a aplicação web, já de início pode-se perceber que um dos pontos chave do sistema é a segurança da informação, demonstrada pelo Quadro 6, dizendo respeito às rotinas responsáveis por fazerem o controle de acesso a aplicação. As linhas 12 a 27 contêm a rotina `PrivateRoute`, responsável por verificar se a sessão do usuário está ativa, conforme linha 17, direcionando-o para o componente solicitado, ou se necessário, direcionando-o para a tela de login para que ele se autentique novamente. Já nas linhas 29 a 38 consta a rotina `PublicRoute`, responsável por direcionar todas as requisições feitas antes da autenticação do usuário para a tela de login, para que o usuário insira suas credenciais e possa adentrar ao sistema.

Quadro 6 - Codificação das rotinas de acesso a aplicação

```

12  function PrivateRoute({ component: Component, authed, ...rest }) {
13    return (
14      <Route
15        {...rest}
16        render={props =>
17          authed === true ? (
18            <Component {...props} />
19          ) : (
20            <Redirect
21              to={{ pathname: '/login', state: { from: props.location } }}
22            />
23          )
24        )
25      />
26    );
27  }
28
29  function PublicRoute({ component: Component, authed, ...rest }) {
30    return (
31      <Route
32        {...rest}
33        render={props =>
34          authed === false ? <Component {...props} /> : <Redirect to="/" />
35        }
36      />
37    );
38  }

```

Fonte: elaborado pelo autor.

Entre as linhas 45 e 59 do Quadro 7 está a rotina `componentDidMount`, que após verificar se o usuário está autenticado na linha 47 atualiza as variáveis de verificações de acordo com a condição realizada linha 47. Já nas linhas 60 a 62 é codificada a rotina `componentWillUnmount`, responsável por destruir a sessão usuário após ele fechar o componente `componentWillUnmount`.

Quadro 7- Codificação das rotinas antes de iniciar o acesso a aplicação

```

45  componentDidMount() {
46    this.removeListener = firebaseAuth().onAuthStateChanged(user => {
47      if (user) {
48        this.setState({
49          authed: true,
50          loading: false
51        });
52      } else {
53        this.setState({
54          authed: false,
55          loading: false
56        });
57      }
58    });
59  }
60  componentWillUnmount() {
61    this.removeListener();
62  }

```

Fonte: elaborado pelo autor.

O Quadro 8 contempla a rotina `render`, dispostas entre as linhas 63 e 88, sendo a rotina principal do componente, responsável por inicializá-lo, chamando as rotinas auxiliares definidas anteriormente no escopo. Na linha 70 é verificado se a requisição está sendo feita por um usuário autenticado, chamando assim a rotina `PrivateRoute`, entre as linhas 77 e 81, caso contrário, chamará a rotina `PublicaRoute` entre as linhas 72 e 76.

Quadro 8 – Codificação da rotina iniciando o acesso a aplicação

```

63  render() {
64      return this.state.loading === true ? (
65          <SpinnerFlex>
66              <Spinner large intent="primary"/>
67          </SpinnerFlex>
68      ) : (
69          <BrowserRouter>
70              <React.Fragment>
71                  <Switch>
72                      <PublicRoute
73                          authed={this.state.authed}
74                          path="/login"
75                          component={Login}
76                      />
77                      <PrivateRoute
78                          authed={this.state.authed}
79                          path="/"
80                          component={Dashboard}
81                      />
82                      <Route render={() => <h3>No Match</h3>} />
83                  </Switch>
84              </React.Fragment>
85          </BrowserRouter>
86      );
87  }
88  }

```

Fonte: elaborado pelo autor.

Seguindo para a criação do formulário da tela de login, sendo apresentado no Quadro 9 as regras a serem aplicadas nos campos presentes na tela. O campo de e-mail possui validação de não nulo, linha 8 e validação de e-mail válido, linha 9; e o campo senha possui apenas a validação de não nulo, codificado na linha 12.

Quadro 9 – Codificação da rotina de configuração de campo de formulário

```

6  const config = {
7      email: {
8          isRequired: 'Não pode ser nulo',
9          isEmail: 'Deve ser um endereço de e-mail válido'
10     },
11     password: {
12         isRequired: 'Não pode ser nulo'
13     }
14 };

```

Fonte: elaborado pelo autor.

Entre as linhas 16 e 50 do Quadro 10 é apresentado a criação do formulário do login, com todos os campos e validações apresentadas no passo anterior. As linhas 20 a 32 é referente a criação e definição do campo de e-mail em que as linhas 20 a 25 são utilizadas para definir o label do campo e as linhas 26 a 31 são utilizadas para definir o campo em si. Logo abaixo, nas linhas 33 a 46, é referente a criação e a definição do campo senha em que as linhas 34 a 38 é para definir o label do campo e as linhas 41 a 44 são utilizadas para definir o campo em si.

Quadro 10 – Codificação da rotina formulário do login

```

16  const LoginForm = props => (
17    <Validation config={config} initialValues={props.fields}>
18      ({ errors, setField, fields, submitted }) => (
19        <Fragment>
20          <FormGroup
21            helperText={errors.email}
22            label="E-mail"
23            labelFor="email"
24            intent="danger"
25          >
26            <input
27              id="email"
28              name="email"
29              placeholder="e-mail@dominio.com"
30              type="email"
31            />
32          </FormGroup>
33          <FormGroup
34            helperText={errors.password}
35            placeholder="Senha"
36            label="Senha"
37            labelFor="password"
38            intent="danger"
39          >
40            <input
41              id="password"
42              name="password"
43              placeholder="*****"
44              type="password"
45            />
46          </FormGroup>
47        </Fragment>
48      )
49    </Validation>
50  );

```

Fonte: elaborado pelo autor.

Com o formulário pronto, passando para a tela de login, entre as linhas 19 e 30 do Quadro 11, está a rotina responsável por retornar uma mensagem de erro ao usuário caso as credenciais inseridas no formulário sejam inválidas. Na linha 22 é feita a validação do usuário pelo método `login`, que em caso de inconsistência nas credenciais retornará um código de erro. Na linha 23 é verificado se o código de retorno é referente ao erro de usuário não encontrado, retornando para a aplicação a mensagem `Usuário não existe`. Enquanto na linha 25 é verificado se o código de retorno é referente ao erro de senha incorreta, retornando então para a aplicação a mensagem `Senha incorreta`.

Quadro 11 – Codificação da validação dos campos do login

```

19 export default class Login extends Component {
20   handleSubmit = ({ errors, fields, isValid }) => {
21     if (isValid) {
22       login(fields.email, fields.password).catch(error => {
23         if (error.code === 'auth/user-not-found') {
24           AppToaster.show({ message: 'Usuário não existe' });
25         } else if (error.code === 'auth/wrong-password') {
26           AppToaster.show({ message: 'Senha incorreta' });
27         }
28       });
29     }
30   };

```

Fonte: elaborado pelo autor.

O Quadro 12 traz a rotina render, na qual dispõem o formulário de login para o usuário, demonstrado nas linhas 37 a linha 46 e também chama função handleSubmit demonstrada no Quadro 11, na linha 34, para verificar se o usuário e senha inseridos são válidos.

Quadro 12 – Codificação da rotina inicialização do login

```

32   render() {
33     return (
34       <Form onSubmit={this.handleSubmit}>
35         <FormContainerFlex>
36           <ControlGroup>
37             <LoginFormGrid>
38               <div className="login">
39                 <img
40                   className="login-img-title"
41                   alt="login-img-tittle"
42                   src={imgLogoTitle}
43                 />
44                 <h2 className="login-title"> SELBST </h2>
45               </div>
46               <LoginForm />
47               <Button
48                 className="pt-intent-success"
49                 text="Entrar"
50                 type="submit"
51               />
52             </LoginFormGrid>
53           </ControlGroup>
54         </FormContainerFlex>
55       </Form>
56     );
57   }
58 }

```

Fonte: elaborado pelo autor.

No Quadro 13 estão as rotinas que dizem respeito ao principal fonte da aplicação, responsável por centralizar as demais telas e rotinas da aplicação. Já de início, entre as linhas 11 e 24, é codificado a rotina AdminRoute que após verificar se o usuário autenticado é um administrador, linha 16, libera os componentes Usuários e Regras, relacionados a esse perfil de usuário.

Quadro 13 – Codificação das rotinas de administrador

```

11 function AdminRoute({ component: Component, authenticated, ...rest }) {
12   return (
13     <Route
14       {...rest}
15       render={props =>
16         authenticated === true ? (
17           <Component {...props} />
18         ) : (
19           <Redirect to={{ pathname: '/', state: { from: props.location } }} />
20         )
21       )
22     />
23   );
24 }

```

Fonte: elaborado pelo autor.

O Quadro 14 traz a codificação de buscar as informações do cliente. Logo abaixo, entre as linhas 37 e 139 está a rotina `componentWillMount`, responsável por validar e carregar os dados um pouco antes da tela ser criada, visando a performance da aplicação. A primeira rotina a ser executada, conforme Quadro 14, é responsável por salvar as informações referente ao usuário autenticado nos *arrays* `dataSourceCharts`, `dataSourceKey` e *array* `dataSourceProfile`. Sendo que entre as linhas 45 e 49, são salvas as chaves, do objeto trabalho e seus respectivos valores no *array* `dataSourceCharts`, que será utilizado posteriormente pelo componente que gerencia as informações a serem utilizadas pelo usuário final. Já entre as linhas 50 e 53 são salvos apenas as chaves do objeto trabalho no *array* `dataSourceKey`, que é utilizado para demonstrar os cubos de informações disponíveis para o usuário trabalhar. E por fim, entre as linhas 59 e 60 é atribuído ao *array* `dataSourceProfile`, `TRUE`, caso o usuário autenticado possuir um perfil de administrador, utilizado posteriormente para liberar acesso a algumas funcionalidades da aplicação específicas para esse tipo de perfil.

Quadro 14 – Codificação da rotina que busca informações do usuário

```

39   ref
40   .child(`USERS/${firebaseAuth().currentUser.uid}`)
41   .on('value', function(snapshot) {
42     snapshot.forEach(function(childSnapshot) {
43       if (typeof childSnapshot.val() === 'object') {
44         if (childSnapshot.key !== 'PROFILE') {
45           self.setState({
46             dataSourceCharts: [
47               ...self.state.dataSourceCharts,
48               { key: childSnapshot.key, val: childSnapshot.val() }
49             ],
50             dataSourceKey: [
51               ...self.state.dataSourceKey,
52               { key: childSnapshot.key }
53             ]
54           });
55         } else if (
56           childSnapshot.key === 'PROFILE' &&
57           childSnapshot.val().isAdmin
58         ) {
59           self.setState({
60             dataSourceProfile: { hasPermision: true }
61           });
62         }
63       }
64     });
65   });

```

Fonte: elaborado pelo autor.

Em seguida, no Quadro 15 estão as rotinas responsáveis por identificar as alterações feitas na tela Usuário, armazenando essas informações no *array* `dataSourceUsers`, para que posteriormente seja possível atualizar a tela sem a necessidade de fazer uma nova consulta no banco de dados, bem como melhorar a performance da aplicação. É possível verificar que a rotina codificada entre as linhas 67 e 79, verifica se foi adicionado um usuário novo, adicionando suas informações no *array* `dataSourceUsers`. Já a rotina codificada entre as linhas 81 e 86, identifica se um usuário foi removido, pegando todos os usuários que tiverem o identificador diferente do usuário excluído e atribui o resultado obtido ao *array* `dataSourceUsers`. E por fim, a rotina codificada entre as linhas 88 e 102, responsável por verificar se um usuário foi modificado, trazendo as novas informações para serem atribuídas ao *array*.

Quadro 15 – Codificação da rotina de Alteração de usuários

```

67     ref.child('USERS').on('child_added', snapshot => {
68         self.setState({
69             dataSourceUsers: [
70                 ...self.state.dataSourceUsers,
71                 {
72                     agency: snapshot.val().cdcooper,
73                     account: snapshot.val().nrdconta,
74                     email: snapshot.val().PROFILE.email,
75                     uid: snapshot.key
76                 }
77             ]
78         });
79     });
80
81     ref.child('USERS').on('child_removed', snapshot => {
82         self.state.dataSourceUsers = self.state.dataSourceUsers.filter(
83             e => e.uid !== snapshot.key
84         );
85         self.setState(self.state);
86     });
87
88     ref.child('USERS').on('child_changed', snapshot => {
89         self.state.dataSourceUsers = self.state.dataSourceUsers.map(e => {
90             if (e.uid === snapshot.key) {
91                 return {
92                     agency: snapshot.val().cdcooper,
93                     account: snapshot.val().nrdconta,
94                     email: snapshot.val().PROFILE.email,
95                     uid: snapshot.key
96                 };
97             } else {
98                 return e;
99             }
100         });
101         self.setState(self.state);
102     });

```

Fonte: elaborado pelo autor.

No Quadro 16, é demonstrada as rotinas que são responsáveis por identificar as alterações feitas na tela Query, na qual seguem a mesma lógica das rotinas apresentadas no Quadro 15, em que nas linhas 104 a 116 é codificada a rotina responsável por identificar as regras adicionadas, logo abaixo, possui a rotina responsável por verificar se alguma regra foi removida, entre as linhas 118 a 122; e por último, nas linhas 124 a 139, é demonstrada a rotina responsável por verificar se alguma regra foi alterada, sempre atribuindo os valores consultados ao *array* dataSource, utilizado para atualizar a tela Query posteriormente.

Quadro 16 – Codificação da rotina de Alteração de regras

```

104     ref.child('RULES').on('child_added', snapshot => {
105         self.setState({
106             dataSource: [
107                 ...self.state.dataSource,
108                 {
109                     rules: snapshot.key,
110                     query: snapshot.val(),
111                     delete: snapshot.key,
112                     edit: snapshot.key
113                 }
114             ]
115         });
116     });
117
118     ref.child('RULES').on('child_removed', snapshot => {
119         self.state.dataSource = self.state.dataSource.filter(e => e.rules !== snapshot.key);
120     });
121     self.setState(self.state);
122 });
123
124     ref.child('RULES').on('child_changed', snapshot => {
125         self.state.dataSource = self.state.dataSource.map(e => {
126             if (e.rules === snapshot.key) {
127                 return {
128                     rules: snapshot.key,
129                     query: snapshot.val(),
130                     delete: snapshot.key,
131                     edit: snapshot.key
132                 };
133             } else {
134                 return e;
135             }
136         });
137         self.setState(self.state);
138     });
139 }

```

Fonte: elaborado pelo autor.

O Quadro 17 traz a rotina `switchChartState`, responsável por alterar o cubo de informações quando solicitado pelo o usuário, atribuindo os valores selecionados as variáveis `selectedDataSource` e utilizada para informar ao componente que gerencia os gráficos, quais informações devem ser trabalhadas. Conforme demonstrada na linha 145, e `selectedDataSourceKey`, utilizada para demonstrar ao usuário em qual cubo ele está trabalhando, conforme demonstrado na linha 146.

Quadro 17 – Codificação da rotina Mudança de Cubo

```

141     switchChartState = name => {
142         this.state.dataSourceCharts.map(val => {
143             if (val.key === name) {
144                 this.setState({
145                     selectedDataSource: val.val,
146                     selectedDataSourceKey: val.key
147                 });
148                 this.props.history.replace('/');
149             }
150         });
151     };

```

Fonte: elaborado pelo autor.

O Quadro 18 traz a rotina `render`, codificada entre as linhas 153 a 189, em que nas linhas 156 a 161 é chamado o `Header` do programa, passando como parâmetro os `arrays` `dataSourceKey`, `dataSourceProfile`, `selectedDataSourceKey`, `switchChartState`, carregados anteriormente na rotina. Logo abaixo, entre as linhas 162 a 178, é demonstrada a chamada da rotina `AdminRoute`, que libera acesso aos componentes `Usuário` e `Regras`, caso o usuário autenticado possuir acesso de administrador. E por último é exibido a chamada do componente `Pivottable`, responsável por criar o `dashboard`, local que o usuário edita suas análises, passando como parâmetro o `array` `selectedDataSource`, indicando qual cubo o usuário escolheu para trabalhar.

Quadro 18 – Codificação da rotina inicialização do `Dashboard`

```

153   render() {
154     return (
155       <Fragment>
156         <Header
157           chartsButton={this.state.dataSourceKey}
158           hasPermission={this.state.dataSourceProfile.hasPermission}
159           buttonClick={this.state.selectedDataSourceKey}
160           click={this.switchChartState.bind(this)}
161         />
162         <AdminRoute
163           authed={this.state.dataSourceProfile.hasPermission}
164           path="/query"
165           component={() => (
166             <Query data={this.state.dataSource} history={this.props.history} />
167           )}
168         />
169         <AdminRoute
170           authed={this.state.dataSourceProfile.hasPermission}
171           path="/usuarios"
172           component={() => (
173             <Users
174               data={this.state.dataSourceUsers}
175               history={this.props.history}
176             />
177           )}
178         />
179         <Route
180           path="/"
181           exact
182           component={() => (
183             <Pivottable selectedDataSource={this.state.selectedDataSource} />
184           )}
185         />
186       </Fragment>
187     );
188   }
189 }

```

Fonte: elaborado pelo autor.

O Quadro 19 traz a codificação das rotinas responsáveis por modelar o `Header`, em que nas linhas 20 a 22, possui o nome da aplicação com um link redirecionando para a página inicial. Já nas linhas 24 a 37, possui os componentes direcionados ao perfil de Administrador, verificado na linha 24, sendo eles `Usuário` e `Regras`, que são responsáveis por redirecionar para suas respectivas telas, liberadas anteriormente pelo componente `Dashboard`. Entre as linhas 38 e 58 está o componente responsável por listar os `Cubos` de

Dados, para que o usuário escolha em qual deles ele deseja trabalhar. Por último, nas linhas 60 a 67 está o componente responsável por efetuar o *logout* da aplicação SELBST.

Quadro 19 – Codificação da rotina criação do Header

```

17  const Header = props => (
18    <Navbar>
19      <NavbarGroup align={Alignment.LEFT}>
20        <Link to={'/'}>
21          <NavbarHeading>SELBST</NavbarHeading>
22        </Link>
23        <NavbarDivider />
24        {props.hasPermision && (
25          <Fragment>
26            <Link to={'/usuarios'}>
27              <Button className="pt-minimal" icon="person" text="Usuários" />
28            </Link>
29            <Link to={'/query'}>
30              <Button
31                className="pt-minimal"
32                icon="manually-entered-data"
33                text="Regras"
34              />
35            </Link>
36          </Fragment>
37        )}
38        <Popover
39          content={
40            <Menu>
41              {props.chartsButton.map(keyValue => {
42                return (
43                  <MenuItem
44                    key={keyValue.key}
45                    text={keyValue.key}
46                    onClick={() => props.click(keyValue.key)}
47                  />
48                );
49              }}
50            </Menu>
51          }
52        >
53          <Button
54            className="pt-minimal"
55            icon="more"
56            text={props.buttonClick || 'Selecione um Cubo'}
57          />
58        </Popover>
59      </NavbarGroup>
60      <NavbarGroup align={Alignment.RIGHT}>
61        <Button
62          onClick={logout}
63          className="pt-minimal"
64          icon="log-out"
65          text="Sair"
66        />
67      </NavbarGroup>
68    </Navbar>
69  );

```

Fonte: elaborado pelo autor.

Por fim, no Quadro 20 é demonstrada as rotinas responsáveis pela criação da área de trabalho do usuário, denominada como *workspace*, em que entre as linhas 25 e 30, dentro da rotina `render`, é chamado o componente `Plotly`, responsável por gerenciar o *workspace*, passando como parâmetro os dados do `cubo` selecionado pelo usuário, linha 26, o estado do componente atual, linha 27 e também os componentes necessários para a criação visual da *workspace*, linha 28. Ademais, caso o usuário não tenha selecionado nenhum `cubo` para ser trabalhado, será disposto na tela inicial a mensagem `Selecione algum Cubo`, conforme linha 34.

Quadro 20 – Codificação da rotina criação do *workspace*

```

21  render() {
22    return (
23      <PivottableFlex>
24        {this.state.selectedDataSource.length > 0 ? (
25          <PivotTableUI
26            data={this.state.selectedDataSource}
27            onChange={s => this.setState(s)}
28            renderers={Object.assign({}, TableRenderers, PlotlyRenderers)}
29            {...this.state}
30          />
31        ) : (
32          <NonIdealStateFlex>
33            <NonIdealState
34              title="Selecione algum Cubo"
35              visual="series-search"
36            />
37          </NonIdealStateFlex>
38        )}
39      </PivottableFlex>
40    );
41  }
42 }

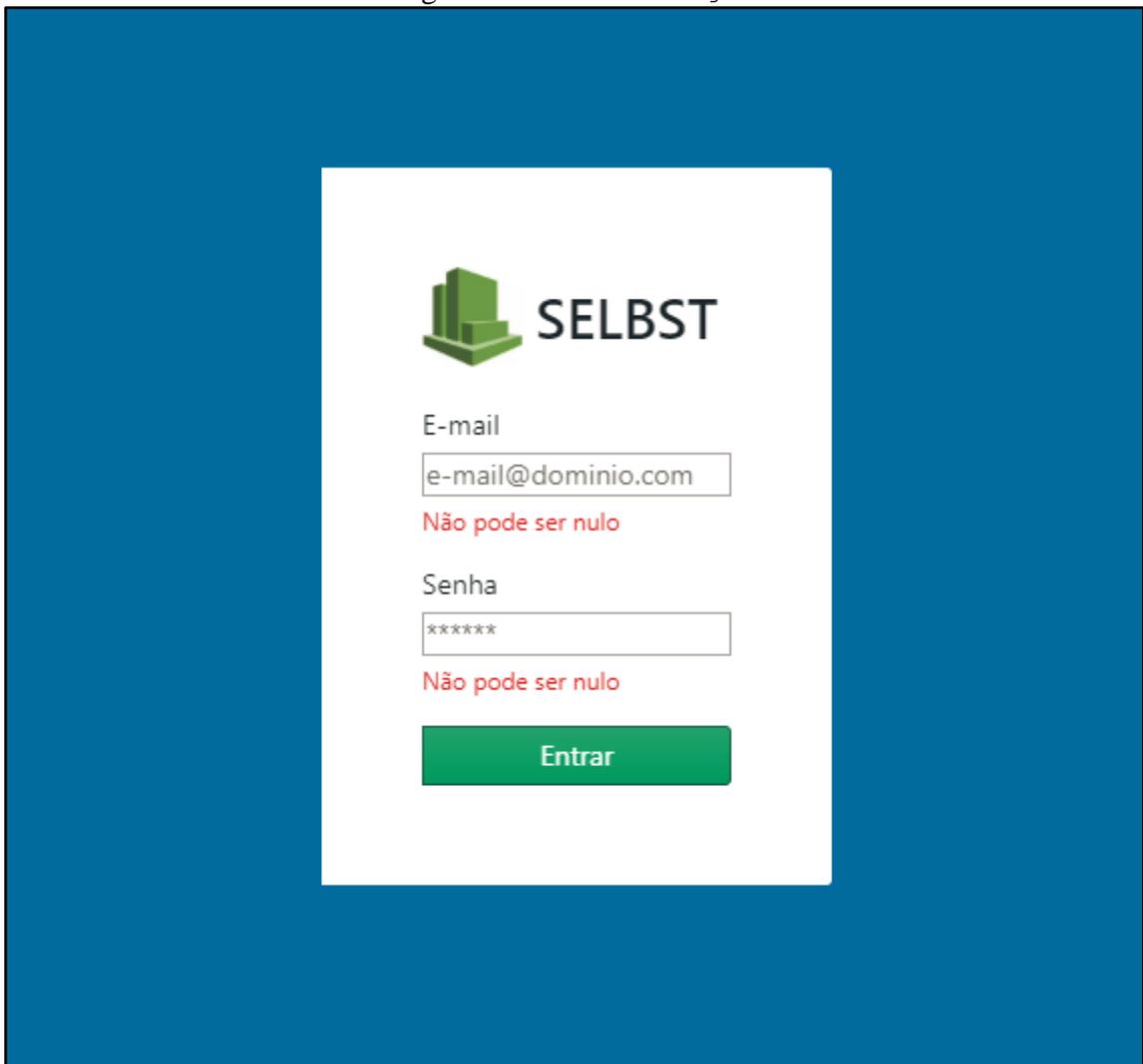
```

Fonte: elaborado pelo autor.

3.3.3 Operacionalidade da implementação

A operacionalidade da aplicação inicia com o usuário solicitando para a instituição financeira Viacredi, o acesso a aplicação `SELBST`. Logo após ser liberado, o usuário poderá se autenticar pela tela de `login`, informando seu respectivo `E-mail` e `Senha`, repassados pela instituição financeira e clicando na opção `Entrar`, para que a aplicação valide suas credenciais, conforme Figura 17.

Figura 17 - Tela de login



Logo do sistema SELBST, composto por um ícone de três blocos verdes e o texto "SELBST".

E-mail

Não pode ser nulo

Senha

Não pode ser nulo

Fonte: elaborado pelo autor.

Após efetuar o *login*, o usuário será redirecionado para a tela principal da aplicação, conforme Figura 17, em que na demarcação da letra A, o usuário tem a funcionalidade de retornar a tela inicial da aplicação, de qualquer parte da aplicação em que ele esteja. As funcionalidades demarcadas pela letra B, só estarão disponíveis para os usuários que possuem um perfil de administrador, em que possui as opções de ir para a tela Usuário ou ir para a tela Regras. Na opção demarcada pela letra C, é habilitada para todos os usuários, e tem a funcionalidade de selecionar o cubo de informações em que o usuário deseja confeccionar suas análises, o qual será detalhado mais abaixo na Figura 18; e por último, está a funcionalidade demarcada pela letra D, responsável por finalizar a sessão da aplicação para o usuário autenticado.

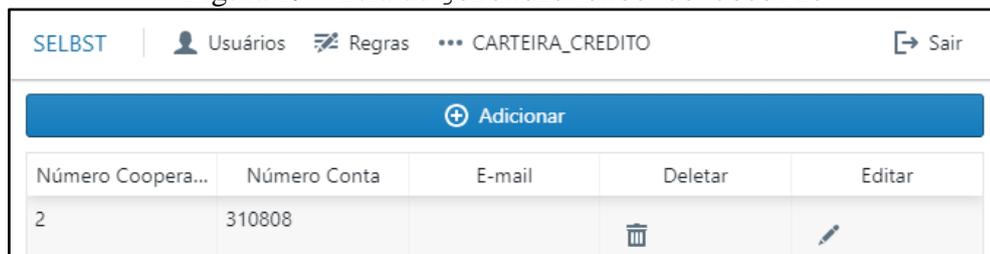
Figura 18 - Header



Fonte: elaborado pelo autor.

A opção **Usuários** e a opção **Regras**, conforme demarcação da letra **B** na Figura 18, possuem a mesma sistemática, alterando apenas a informação disposta, em que quando selecionado a opção **Usuários** habilita a tela de gerenciamento de usuários, conforme Figura 19, que de início mostra todos os usuários cadastrados com suas respectivas informações. Já o campo **Deletar** tem a funcionalidade de remover o usuário selecionado. No campo **Editar**, possui a opção de alterar as informações do usuário selecionado. E mais acima, no botão **Adicionar**, tem a opção de adicionar um novo usuário.

Figura 19 - Tela de gerenciamento de usuário



Fonte: elaborado pelo autor.

Tanto na opção **Editar** quanto a **Adicionar**, a aplicação habilita a tela de cadastro de usuário, conforme Figura 20, porém quando o usuário de perfil administrador deseja alterar informações do usuário normal, essa tela já vem preenchida com as informações referente ao usuário escolhido, podendo adicionar ou alterar os valores referentes a Cooperativa, Conta e-mail e senha.

Figura 20 - Tela de cadastro de usuário

Fonte: elaborado pelo autor.

Quando selecionado a opção *Regras*, conforme demarcação da letra B na Figura 18, a aplicação habilita a tela de gerenciamento de regras, conforme Figura 21, que de início mostra todos as regras cadastrados com suas respectivas informações. No campo *Deletar* é demonstrada a funcionalidade de *remover* a regra selecionada; enquanto o campo *Editar* possui a opção de *alterar* as informações da regra selecionada; e mais acima, o botão *Adicionar* contém a opção de *adicionar* uma nova regra.

Figura 21 - Tela de gerenciamento de regras

Nome da regra	Query	Deletar	Editar
CARTEIRA_CREDITO	select dia.dtreferencia ...		

Fonte: elaborado pelo autor.

Seguindo o mesmo padrão do gerenciamento de usuário, as opções *Editar* e *Adicionar*, compactuam da tela de cadastro de regras, demonstrada na Figura 22, em que quando o usuário de perfil *administrador* deseja alterar alguma regra, essa tela vem preenchida com as informações referente a regra escolhida. A tela em si é composta apenas por duas informações, sendo a primeira o *Nome da regra*, em que o *administrador* rotula esse grupo de informações, a modo que fique de fácil entendimento para o usuário final; e a *query* da regra, que será utilizada para buscar as informações na base de dados da instituição financeira e inseri-las na base da aplicação.

Figura 22 - Tela de cadastro de regras

SELBST | Usuários | Regras | ... Selecione um Cubo | Sair

Nome da regra
Nova Regra

Query para a regra
1 select * from dual;

Voltar
Criar

Fonte: elaborado pelo autor.

Voltando para a tela principal, é demonstrado o detalhamento demarcado pela letra C, demonstrada anteriormente pela Figura 18, que é a opção *Selecione um Cubo*. Ao clicar no componente selecionado no cubo é demonstrado para o usuário uma lista de cubos de dados disponíveis na aplicação SELBST, conforme Figura 23, possibilitando selecionar um cubo e iniciar suas análises.

Figura 23 - Lista de Cubos

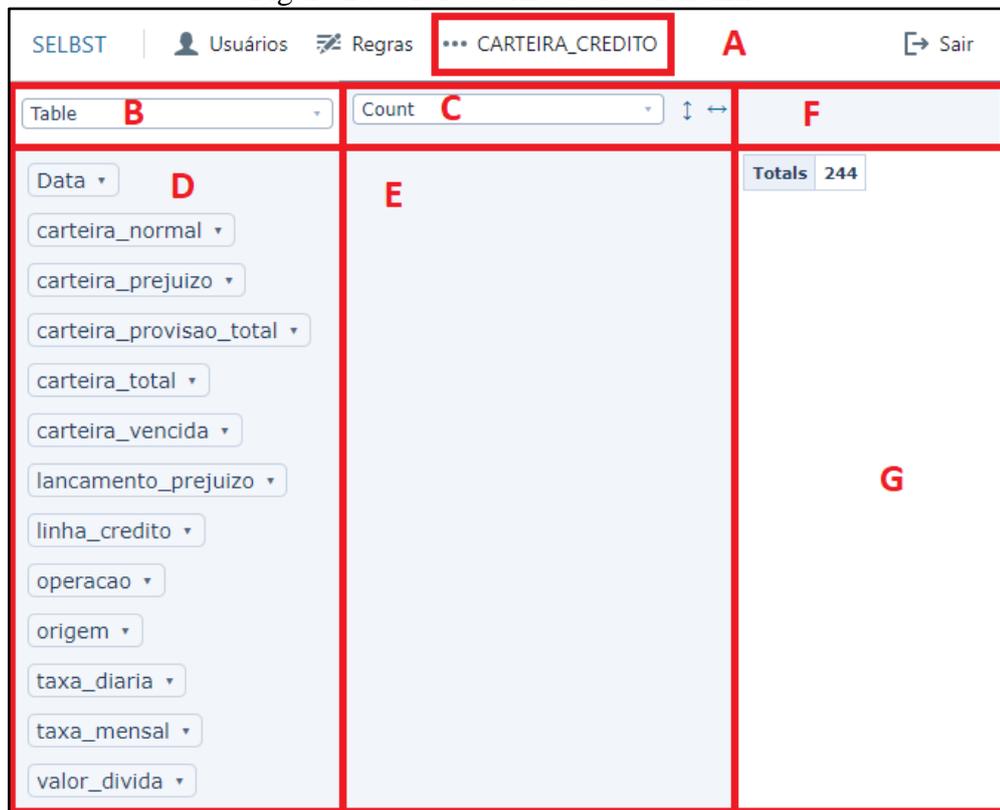
... Selecione um Cubo

- CARTEIRA_CREDITO
- INDICADOR_COBRANCA
- INDICADOR_RECIPROCIDADE
- LANCAMENTOS
- STATUS_COBRANCA

Fonte: elaborado pelo autor.

A Figura 24 traz a Tela da Área de Trabalho. Após selecionar o cubo de dados a ser manuseado (letra A da Figura 24), a aplicação SELBST habilita a Workspace, demonstrado pela Figura 24, em que na letra B é disponibilizado os tipos de visualizações que podem ser aplicadas nas análises a serem feitas pelo usuário. Na letra D, são listadas as informações em que o usuário poderá utilizar para confeccionar suas análises. Já na letra C, pode-se aplicar fórmulas aritméticas em determinadas informações para complementar alguma informação. Nas letras E e F, que representam os eixos Y e X respectivamente, é o local em que o usuário arrasta as informações para que elas sejam exibidas em suas análises, independentemente do tipo de visualização escolhida. Ademais, na letra G, são apresentados os resultados das regras aplicadas pelos usuários nas informações disponíveis.

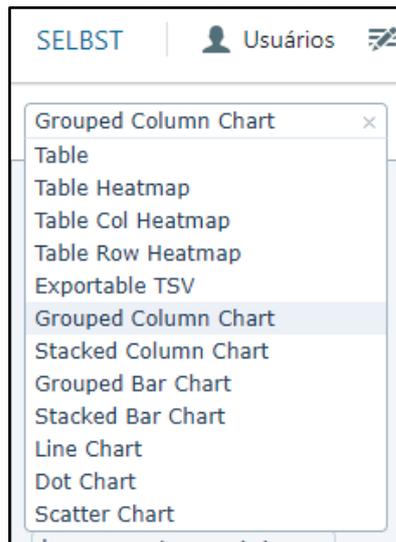
Figura 24 – Tela de Área de trabalho



Fonte: elaborado pelo autor.

Com o Workspace disponível, um dos primeiros passos a ser seguido, é definir qual a forma em que ele deseja visualizar as informações em seu painel. Sendo assim, conforme demarcado pela letra B na Figura 24, o usuário seleciona a lista de possíveis visualizações, no qual mostrará uma gama de gráficos e tabelas, como por exemplo Table, Grouped Column Chart, Line chart, entre outros, que podem ser utilizadas, selecionando a que melhor lhe atender no momento, conforme Figura 25.

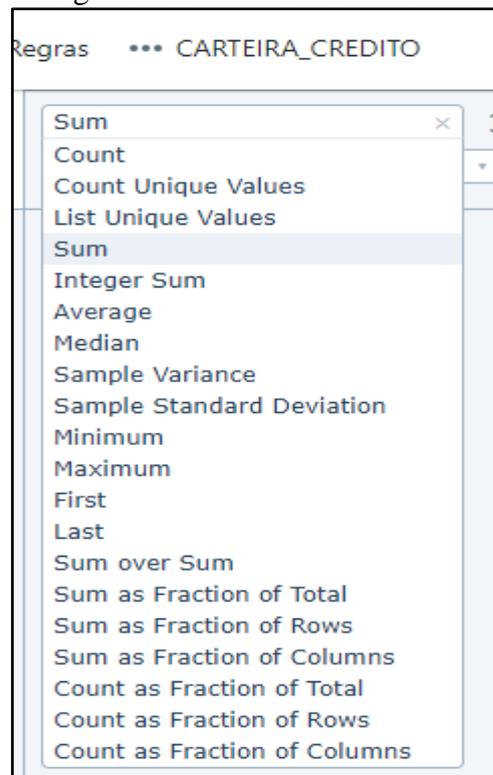
Figura 25 - Lista de visualizações



Fonte: elaborado pelo autor.

O próximo passo a ser seguido é selecionar a fórmula aritmética a ser utilizada no painel, em que após selecionar o campo demarcado pela letra C na Figura 24, habilitará uma lista de fórmulas que poderão ser aplicadas na análise feita pelo usuário, conforme Figura 26.

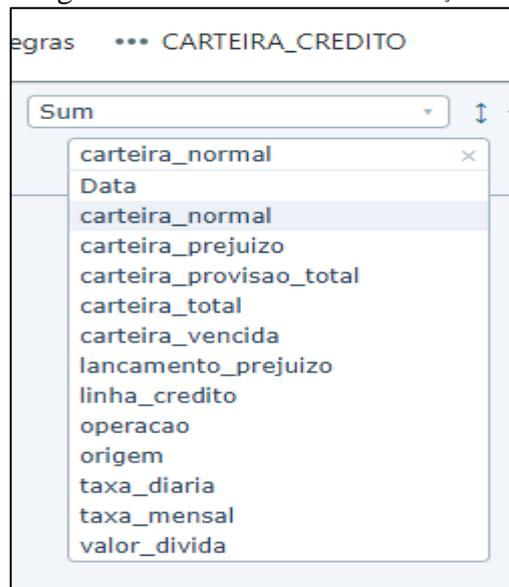
Figura 26 - Lista de fórmulas



Fonte: elaborado pelo autor.

Com base na escolha da fórmula feita no passo anterior, a aplicação habilitará uma lista de informações disponíveis para que o usuário selecione uma delas e aplique a fórmula em questão, conforme Figura 27. Com isso a aplicação irá fazer o cálculo dessas informações e irá disponibiliza-las no workspace na visualização escolhida anteriormente pelo usuário.

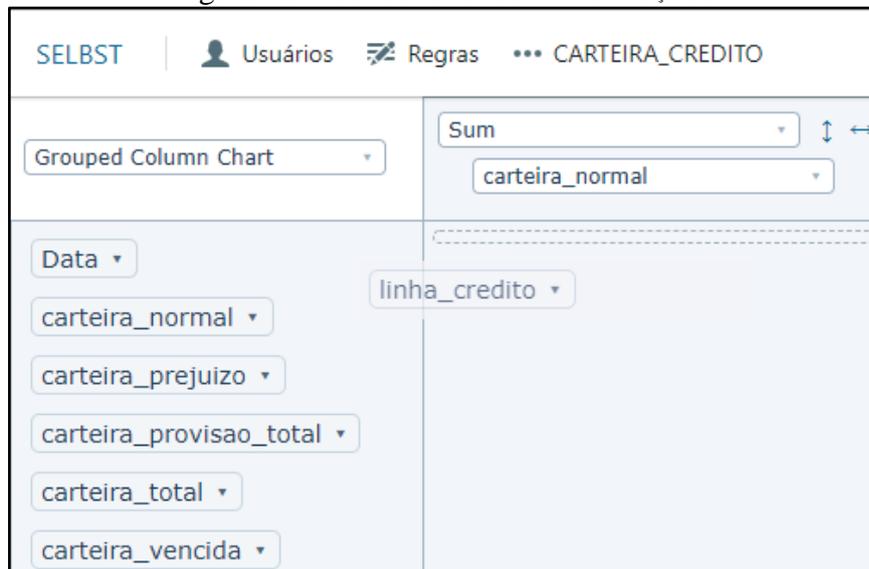
Figura 27 - Lista de informações



Fonte: elaborado pelo autor.

Com o campo de métrica já definido e calculado, o usuário pode complementar sua análise arrastando as informações dispostas na área demarcada pela letra D na Figura 24, para os eixos Y e X, demarcados pelas letras E e F, respectivamente; também na Figura 24, para por exemplo, segregar o valor calculado por alguma descrição, conforme sugere a Figura 28.

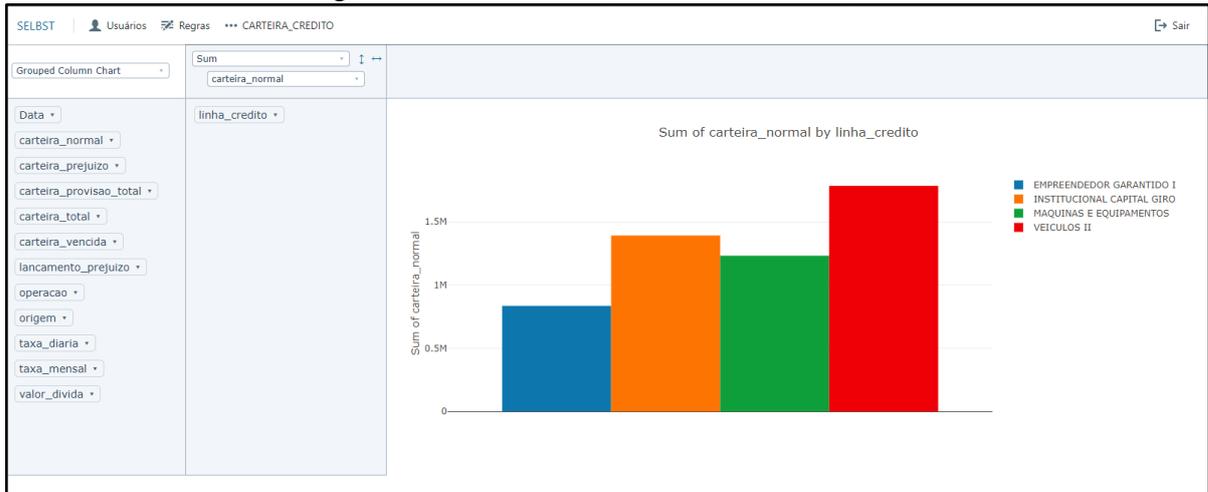
Figura 28 - Selecionando informações



Fonte: elaborado pelo autor.

Após definir o tipo de visualização junto com as métricas e as descrições das informações a serem dispostas, a aplicação monta o resultado desse cruzamento de dados e das regras, na área demarcada pela letra G na Figura 24, conforme Figura 29.

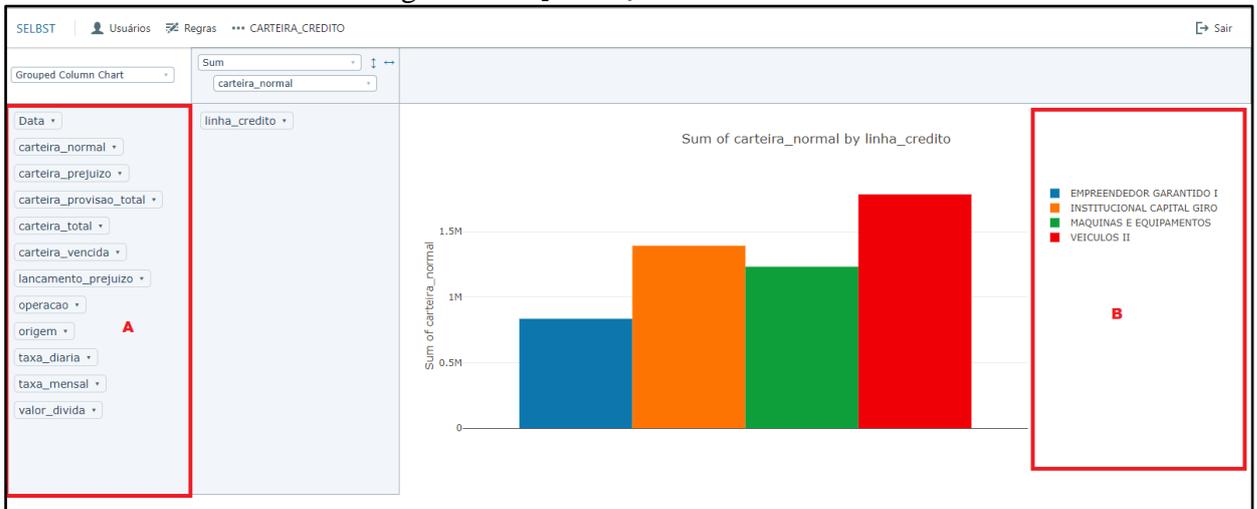
Figura 29 - Visualização das informações



Fonte: elaborado pelo autor.

Com o painel montado, o usuário pode aplicar *filtros* nas informações dispostas para restringir os dados na apresentação. Conforme Figura 30, os filtros podem ser aplicados em duas áreas diferentes. Na área demarcada pela letra A, na qual aplica o filtro em toda a análise ou na legenda, demarcado pela letra B, em que aplica o filtro apenas na apresentação.

Figura 30 - Aplicação de filtros



Fonte: elaborado pelo autor.

Para a aplicar o filtro na análise toda, o usuário deve escolher o campo no qual deseja fazer um filtro, na área demarcada pela letra A da Figura 30 e clicar na *seta* ao lado do nome do campo, conforme Figura 31, selecionando os valores no qual deseja restringir.

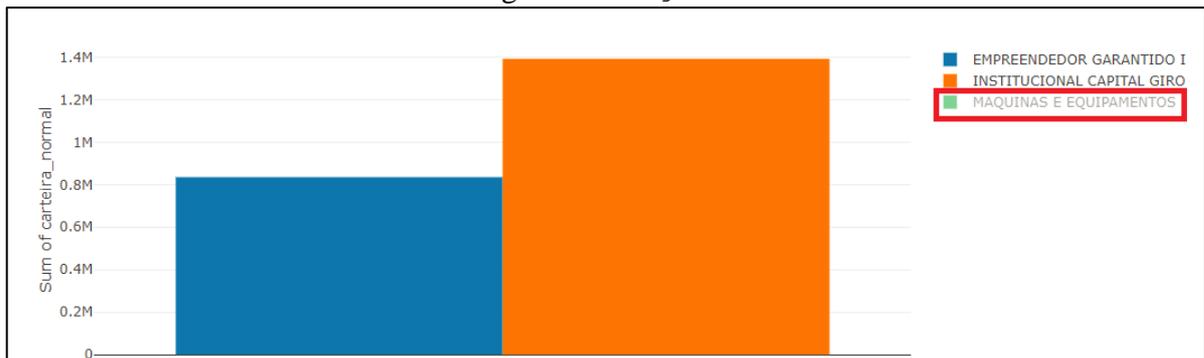
Figura 31 - Tela de filtros



Fonte: elaborado pelo autor.

O usuário poderá restringir o dado apenas na apresentação, para tal, ele deve clicar na informação localizada na legenda ao lado da área de trabalho, demarcado pela letra B na Figura 30, conforme sugere Figura 32.

Figura 32 - Legenda



Fonte: elaborado pelo autor.

A Figura 33 traz as Funcionalidades de Visualização da aplicação SELBST, que ficam dispostas no canto superior direito do painel e são habilitadas quando o tipo de visualização escolhida pelo usuário for um tipo de gráfico. A área de trabalho do usuário permite realizar zoom in e zoom out nos gráficos, letra C da Figura 33, possibilitando visualizar alguma informação separadamente. Ademais, permite selecionar um grupo de resultados, letra B da Figura 33, ficando evidenciados no gráfico e facilitando talvez alguma análise de resultado de um grupo específico. Por último, caso o usuário queira utilizar o painel criado em alguma apresentação, a aplicação fornece a funcionalidade para gerar uma imagem estática do painel no seu estado atual, bem como permitir baixar de forma automática para o computador do usuário, letra A da Figura 33.

Figura 33 - Funcionalidades da visualização



Fonte: elaborado pelo autor.

3.4 RESULTADOS E DISCUSSÕES

Esta seção estruturada da seguinte forma: na subseção 3.4.1 é realizada a comparação entre os trabalhos correlatos e a aplicação desenvolvida neste trabalho; e a subseção 3.4.2 traz a avaliação de usabilidade realizada junto aos usuários especialistas da aplicação, bem como os resultados obtidos são apresentados e as discussões são realizadas.

3.4.1 Comparativo entre os trabalhos correlatos e a aplicação SELBST

Nesta subseção é realizada uma comparação entre os trabalhos correlatos apresentados na seção 2.5 e a aplicação SELBST desenvolvida neste trabalho. Esta correlação está disposta no Quadro 21, distribuindo os correlatos entre as colunas e as características dispostas entre as linhas. Quadro 21- Comparativo entre os trabalhos correlatos e a aplicação SELBST. Pelo referido quadro é possível perceber pela correlação da aplicação SELBST com as características destacadas na última coluna do quadro (da esquerda para direita), que o trabalho em questão alcançou seu objetivo.

Quadro 21- Comparativo entre os trabalhos correlatos e a aplicação SELBST

Características \ Correlatos	Tableau Desktop Tableau Software (2017a)	Data Visualization Oracle (2016)	Visual SAS Institute Inc (2016)	SELBST (2018)
Compartilhamento de Painéis	✓	✓	✓	X
Painéis Interativos	✓	✓	✓	✓
Conexão com várias fontes de dados	✓	✓	✓	X
<i>Autocharting</i>	X	X	✓	X
Pesquisa em linguagem natural	X	✓	X	X
Recurso <i>in-memory</i>	✓	X	X	X
Plataformas (Android/iOS/Web)	X	✓	✓	X
Gratuita	X	X	X	✓
Restrição de dados	X	X	X	✓

Fonte: elaborado pelo autor.

Analisando o Quadro 21, nota-se que o SELBST, assim como todos os outros correlatos, possui a característica *Painéis Interativos*, sendo esta a principal funcionalidade das aplicações de análise de dados. É possível verificar também que esta é a única característica em comum entre o SELBST e os correlatos. Isso se deve ao fato de que a aplicação desenvolvida visa diminuir a complexidade no uso das funcionalidades disponibilizadas para o usuário e também garantir que para a instituição financeira que o usuário está consumindo apenas os dados disponibilizados por ela. Esse ponto se faz importante para o contexto do trabalho pois a aplicação tem que ser a mais minimalista e intuitiva o possível, pelo fato de que grande parte do grupo de usuários foco da aplicação tem um conhecimento amplo do negócio porém nunca utilizaram uma ferramenta para analisar seus dados, e agregar funcionalidades desnecessárias só iria piorar a experiência com a aplicação em primeiro momento. Outro ponto relevante no que se diz respeito a ausência de funcionalidades contidas nos correlatos é de que só dessa maneira a instituição financeira consegue ter total controle de quais informações o usuário está manipulando na aplicação, ficando assim mais fácil para tirar possíveis dúvidas e auxiliá-lo a manipular certa informação.

Contudo a aplicação possui outras características que não são disponibilizadas pelos correlatos e se fazem bastante relevantes tanto para as instituições financeiras quanto para os usuários. Uma delas seria a ausência de cobrança para a utilização da aplicação, sendo ela totalmente *Gratuita* para o usuário e também para a instituição financeira, visando familiarizar as pessoas jurídicas com a análise de dados e demonstrar a sua relevância para o negócio. E por último outra característica em destaque da aplicação é a *Restrição de dados*, onde o analista de sistema, quando for criar uma regra, consegue facilmente definir restrições de dados por

usuário evitando que dados sigilosos sejam acessados por outras pessoas sem a devida autenticação.

3.4.2 Avaliação de usabilidade

Com objetivo de certificar a eficácia e atestar a usabilidade da aplicação desenvolvida, bem como garantir a completude da aplicação, foi realizado uma avaliação de usabilidade, tendo como base a experiência do usuário em sua utilização. A avaliação foi realizada por meio de questionário elaborado de forma quantitativa e qualitativa; e de maneira on-line. A ferramenta utilizada para realização do questionário foi o Google Formulários⁶ e disponibilizada para os participantes da pesquisa por e-mail.

Foi utilizada uma amostra de três usuários especialistas da instituição Viacredi, de três áreas diferentes. A avaliação foi dividida em duas partes: a primeira parte dizendo respeito a identificação dos usuários especialistas da avaliação e a segunda parte relacionada a avaliação de usabilidade em si, referente a aplicação SELBST desenvolvida. Cabe destacar, que a avaliação utilizou princípios do Método User Requirements and User Communicability Assessment to Groupware (M3C-URUCAg) (COSTA, 2018).

Foram aplicadas as seguintes considerações do método na avaliação: as perguntas serem elaboradas fazendo a relação das heurísticas de Nielsen (Apêndice B) com os requisitos da aplicação; que o momento de aplicar a avaliação deve ocorrer após o uso do usuário com a aplicação; a forma de elaborar as repostas para melhor medir o grau de satisfação do usuário devem utilizar escala de severidade (Apêndice C); as tarefas a serem realizadas pelos usuários especialistas devem seguir um roteiro elaborado de forma específica (Apêndice D), para guiar o usuários especialista nas tarefas a serem realizadas, bem como para melhor entendimento da avaliação e no uso da aplicação; disponibilizar a avaliação de usabilidade para o usuário participante da pesquisa de forma on-line, contendo Termo de Consentimento Livre e Esclarecido (TCLE) (Apêndice E), roteiro a seguido e o questionário de avaliação (COSTA, 2018).

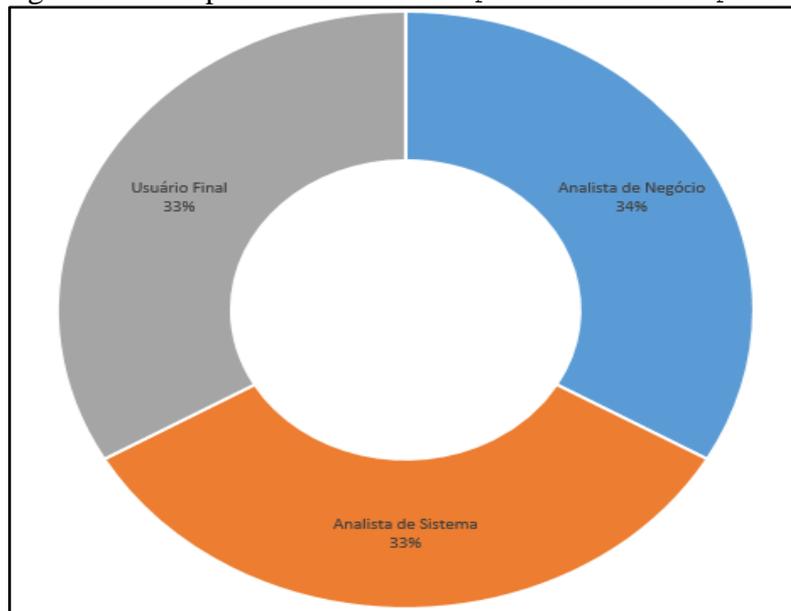
Antes de aplicar a avaliação aos usuários especialistas foi repassado o objetivo do trabalho em si e uma breve introdução sobre a aplicação desenvolvida, comentando sobre suas principais funcionalidades e seu foco de mercado. Logo após, foi encaminhado o link de acesso a uma página do Google Formulários, contendo: o TCLE, convidando o usuário participante a

⁶ Disponível em: <https://docs.google.com/forms/d/e/1FAIpQLSfCgIHI7HaBCwElizRJp1N--D1LC3K2I3-LKrgi0QROEqz5pg/viewform?usp=pp_url>. Acesso em: 19 jun. 2018.

colaborar com o trabalho de conclusão de curso, bem como são apresentados os possíveis danos e os direitos ao participar da avaliação. Ademais, o TCLE contém uma orientação que caso o usuário participante prosseguisse para a seção do roteiro, ele aceitava o termo estabelecido. Ao prosseguir, o formulário exibia um roteiro disponibilizando o link da aplicação SELBST e nove passos para serem seguidos pelos usuários participantes, a fim de conhecerem melhor a aplicação SELBST. No final do roteiro, caso ele se dirigisse para a próxima seção o usuário participante aceitava responder o questionário de avaliação.

Referente ao questionário da avaliação em si, ele era composto por duas perguntas voltadas a conhecer o perfil do participante e quatorze perguntas pertinente a usabilidade da aplicação. A primeira parte do questionário é referente a conhecer melhor os usuários participantes da avaliação, contendo duas perguntas. A Figura 34 traz a primeira pergunta referente ao Papel do Participante na empresa Viacredi, pelo fato de que cada participante selecionado representa um papel significativo para a avaliação aplicação SELBST. Dois dos participantes, equivalendo a 66,66% dos usuários participantes da avaliação representam um papel ativo dentro da aplicação desenvolvida, sendo eles o usuário final e o analista de sistema, enquanto o terceiro participante, representado pelo papel de analista de negócio, equivalendo a 33,33% dos usuários participantes da avaliação, faz uso de uma aplicação com objetivos semelhantes aos da aplicação desenvolvida, o que auxiliou e muito no quesito comparativo.

Figura 34 – Resposta referente ao Papel dos Participantes

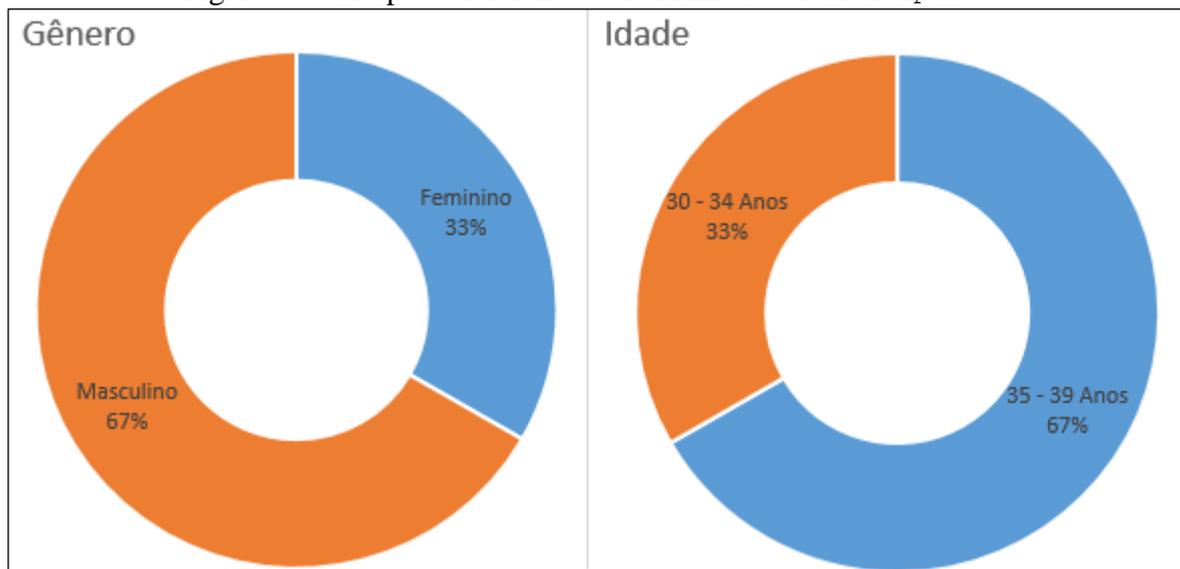


Fonte: elaborado pelo autor.

Na primeira parte do questionário os usuários participantes responderam às perguntas referentes ao seu perfil, em que com base nas informações adquiridas pode-se perceber pelo

gráfico demarcado pelo Gênero na Figura 35, que o sexo predominante foi o masculino, representando 67% dos usuários participantes, e o 33%, era uma representante feminina. Referente a idade, também possui uma predominância, sendo que dois dos participantes possuem uma faixa etária entre os 35 e 39 anos, equivalente a 67% dos usuários participantes e apenas um tem a faixa etária entre 30 e 34 anos, equivalendo a 33% dos participantes, conforme gráfico demarcado pela Idade também na Figura 35.

Figura 35 – Resposta referente ao Perfil dos Participantes



Fonte: elaborado pelo autor.

A segunda parte do questionário diz respeito a avaliação de usabilidade e experiência do usuário. O objetivo das perguntas de usabilidade da aplicação foi identificar a satisfação do usuário em relação a utilização dos mecanismos fornecidos pelo sistema. As perguntas relacionadas a usabilidade da aplicação, nas quais foram fundamentadas nas heurísticas de Nielsen juntamente com o método de avaliação de impacto, em que cada pergunta, quando respondida negativamente, estava relacionada em que eram medidos de zero, sendo este pouco impactante, a quatro, sendo muito impactante. As perguntas em si, elencadas no Quadro 22, visavam capturar a experiência obtida pelos usuários participantes ao usar a aplicação, analisando se atende os padrões lógicos de uma aplicação web, bem como se as funcionalidades atendem completamente o propósito da aplicação.

Quadro 22 – Relação das Heurísticas com as Perguntas do Questionário

HEURÍSTICA	PERGUNTAS DA AVALISAÇÃO
H1	O sistema possui um feedback rápido indicando o que você está fazendo na interface no momento?
	O sistema possui um feedback rápido indicando o cubo que você selecionou?
H2	O sistema utiliza palavras, termos, expressões e conceitos familiares para você?
	As informações aparecem em uma ordem lógica e natural para o seu entendimento?
H3	O sistema possibilita que você saia da opção selecionada a qualquer momento?
H4	As palavras, situações ou ações no sistema são claras, possibilitando o seu entendimento?
	O sistema possui padrões e estilos consistentes?
H5	O sistema possui ícones que ajudam a impedir a ocorrência de erros?
H6	O sistema possui instruções, ações e opções visíveis ou facilmente recuperáveis (sempre que apropriado para o uso)?
H7	O sistema possui teclas, imagens ou ícones para facilitar o seu uso?
H8	Os diálogos do sistema trazem informações relevantes ou na maioria das vezes necessárias?
H9	O sistema possui mensagens de erros com linguagem simples?
H10	O sistema possui opção de ajuda?

Fonte: elaborado pelo autor.

Na coleta de informações baseadas nas respostas do questionário, os participantes levantaram alguns problemas relacionado as heurísticas H4, H6, H9 e H10. Elencada por todos os participantes, a heurística H10, o sistema possui opção de ajuda, poderia ser melhor trabalhada na aplicação, podendo ter talvez alguma tela em específico com informações gerais sobre as funcionalidades da aplicação. Relacionada pelos participantes Usuário Final e Analista de Sistema, a heurística H6, o sistema possui instruções, ações e opções visíveis ou facilmente recuperáveis (sempre que apropriado para o uso), poderia dispor algumas descrições de funcionalidades, visando facilitar o entendimento das mesmas.

Referente a heurística H4, o sistema possui padrões e estilos consistentes, o participante Analista de Sistema comentou que a aplicação poderia explorar mais a estética, deixando uma aparência mais harmônica. Ademais, relacionada pelos participantes Analista de Negócio e Usuário Final, a heurística H9, o sistema possui mensagens de erros com linguagem simples, a aplicação poderia apresentar mensagens um pouco mais amigáveis para o usuário, facilitando seu entendimento. Para facilitar o entendimento da informação disposta acima, o Quadro 23 elenca as heurísticas diagnosticadas (dispostas nas linhas) e os problemas encontrados pelos usuários (dispostas nas colunas), juntamente com o impacto que as mesmas infringem na aplicação.

Quadro 23 – Relação das Heurísticas e impactos

	Analista de Sistema	Analista de Negócio	Usuário Final
H1			
H2			
H3			
H4	0		
H5			
H6	1		2
H7			
H8			
H9		0	2
H10	4	0	4

Fonte: elaborado pelo autor.

Em seguida serão demonstrados detalhadamente os resultados obtidos por meio da avaliação. Referente a primeira heurística H1 foram elaboradas duas perguntas, dispostas nas Figura 36 e Figura 37. A Figura 36 traz o resultado obtido na primeira pergunta relacionada a H1 é se o sistema possui um feedback rápido indicando o que você está fazendo na interface no momento? Sendo possível observar pela referida figura, que 100% dos usuários participantes da avaliação, nos papéis do Analista de Sistema, Analista de Negócio e do Usuário Final, demonstram que a aplicação responde de maneira rápida e fluida os comandos executados, oferecendo uma navegação agradável para o usuário.

Figura 36 - Resposta referente ao Feedback do sistema



Fonte: elaborado pelo autor.

A Figura 37 traz o resultado obtido pela segunda pergunta relacionada a H1 - se o sistema possui um feedback rápido indicando o cubo que você selecionou? Sendo possível observar pelos resultados obtidos conforme dispostos na referida figura, que 100% dos usuários participantes da avaliação demonstram que a aplicação indica de maneira rápida a informação em que o usuário da aplicação escolheu.

Figura 37 - Resposta referente ao Feedback do cubo selecionado



Fonte: elaborado pelo autor.

Referente a segunda heurística H2 foram elaboradas duas perguntas, dispostas nas Figura 38 e Figura 39. A Figura 38 traz o resultado obtido na primeira pergunta relacionada a H2 - se o sistema utiliza palavras, termos, expressões e conceitos familiares para você? Sendo possível observar pela referida figura, que 100% dos usuários participantes da avaliação, demonstram que a aplicação possui termos e nomenclaturas das funções e componentes de fácil entendimento.

Figura 38 - Resposta referente aos Conceitos Familiares



Fonte: elaborado pelo autor.

A Figura 39 traz o resultado obtido pela segunda pergunta relacionada a H2 - se as informações aparecem em uma ordem lógica e natural para o seu entendimento? Sendo possível observar pela figura em questão, que 100% dos usuários participantes da avaliação, demonstram que as funcionalidades da aplicação estão dispostas de maneira fácil e clara para o uso do usuário.

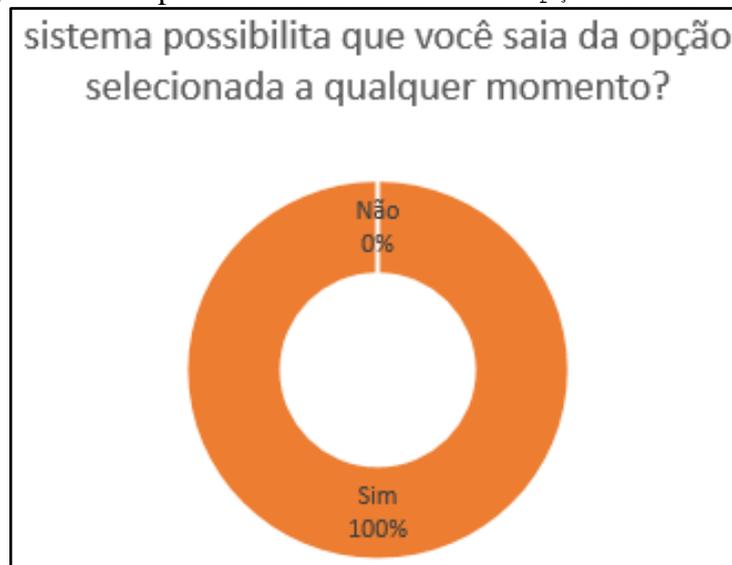
Figura 39 - Resposta referente a Ordem lógica



Fonte: elaborado pelo autor.

Referente a heurística H3 foi elaborada apenas uma pergunta, disposta na Figura 40, trazendo o resultado obtido na pergunta H3 - Se o sistema possibilita que você saia da opção selecionada a qualquer momento? Sendo possível observar pela referida figura, que 100% dos usuários participantes da avaliação, demonstram que a aplicação possui opção para que o usuário a qualquer momento saia da aplicação ou da tela em que se encontra.

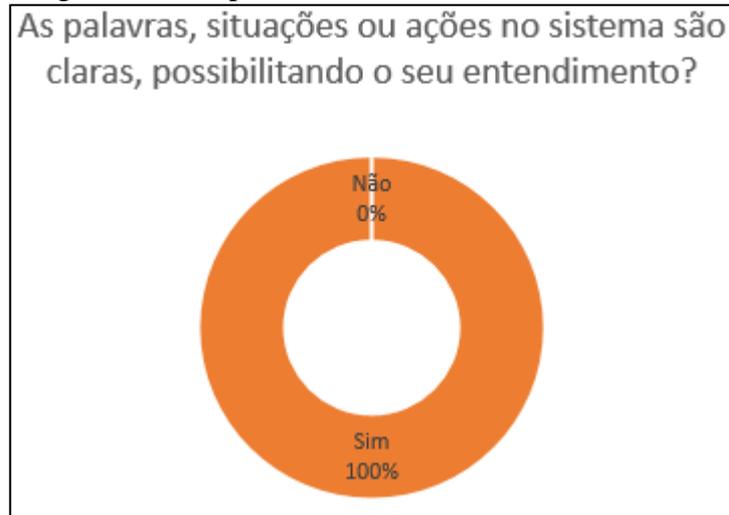
Figura 40 - Resposta referente ao Sair da opção selecionada



Fonte: elaborado pelo autor.

Referente a heurística H4 foi elaborada duas perguntas, disposta na Figura 41 e Figura 42. A Figura 41 traz o resultado obtido na primeira pergunta relacionada a H4 - se as palavras, situações ou ações no sistema são claras, possibilitando o seu entendimento? Sendo possível observar pela referida figura, que 100% dos usuários participantes da avaliação, demonstram que a aplicação possui informações de fácil entendimento para o usuário.

Figura 41 - Resposta referente as Ações do sistema



Fonte: elaborado pelo autor.

A Figura 42 traz o resultado obtido pela segunda pergunta relacionada a H4 - se o sistema padrões e estilos consistentes? Sendo possível observar pela figura em questão, que 67% dos usuários participantes da avaliação, demonstram que a aplicação possui um aspecto visual agradável e harmonioso para o uso, e apenas 33% dos usuários demonstram insatisfação no que se diz respeito a esse aspecto.

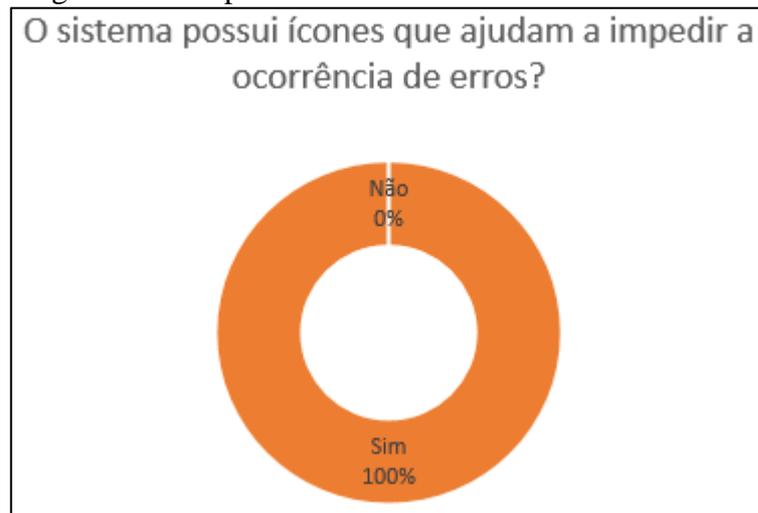
Figura 42 - Resposta referente ao Estilo e padrões consistentes



Fonte: elaborado pelo autor.

Referente a heurística H5 foi elaborada apenas uma pergunta, disposta na Figura 43, trazendo o resultado obtido na pergunta H5 - se o sistema possui ícones que ajudam a impedir a ocorrência de erros? Sendo possível observar pela referida figura, que 100% dos usuários participantes da avaliação demonstram que a aplicação possui ícones que auxiliam a identificação das funcionalidades, evitando possíveis confusão.

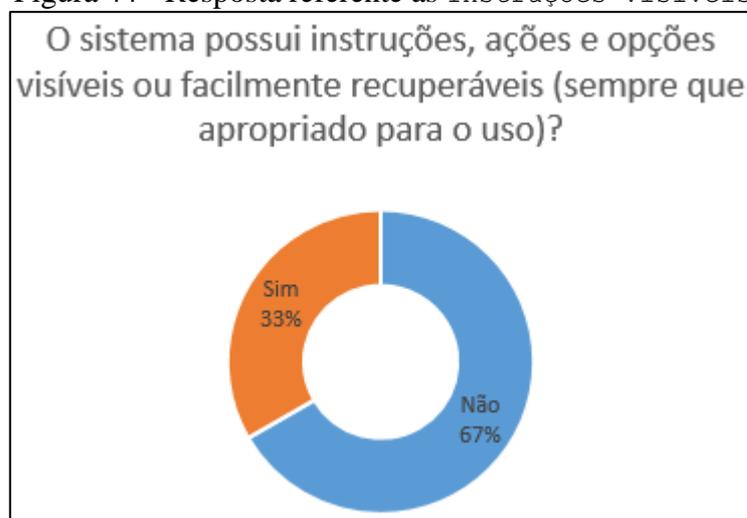
Figura 43 - Resposta referente a Ocorrência de erros



Fonte: elaborado pelo autor.

Referente a heurística H6 foi elaborada apenas uma pergunta, disposta na Figura 44, trazendo o resultado obtido na pergunta H6 - se o sistema possui instruções, ações e opções visíveis ou facilmente recuperáveis (sempre que apropriado para o uso)? Sendo possível observar pela referida figura, que 67% dos usuários participantes da avaliação, demonstram que a aplicação não possui instruções de fácil acesso ou entendimento, e apenas 33% dos usuários demonstram que a aplicação atende esses aspectos.

Figura 44 - Resposta referente as Instruções visíveis



Fonte: elaborado pelo autor.

Referente a heurística H7 foi elaborada apenas uma pergunta, disposta na Figura 45, trazendo o resultado obtido na pergunta H7 - se o sistema possui teclas, imagens ou ícones para facilitar o seu uso? Sendo possível observar pela referida figura, que 100% dos usuários participantes da avaliação, demonstram que a aplicação possui imagens e ícones de fácil entendimento atribuídos as funcionalidades.

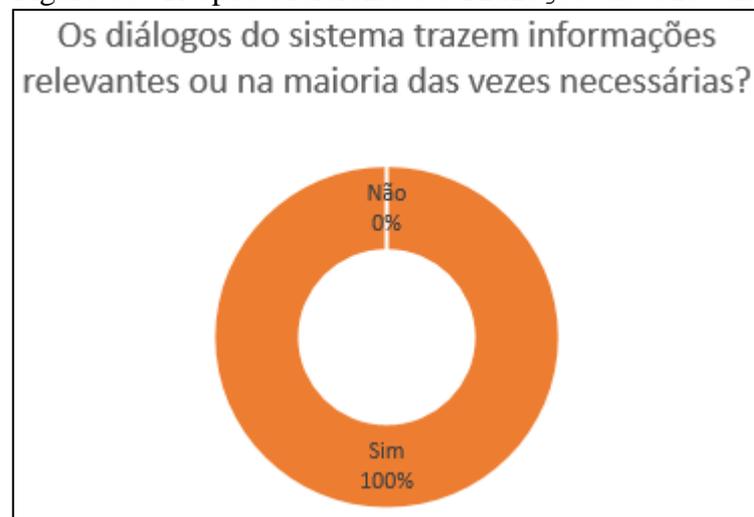
Figura 45 - Resposta referente as Imagens ou ícones



Fonte: elaborado pelo autor.

Referente a heurística H8 foi elaborada apenas uma pergunta, disposta na Figura 46, trazendo o resultado obtido na pergunta H8 - se os diálogos do sistema do sistema trazem informações relevantes ou na maioria das vezes necessárias? Sendo possível observar pela referida figura, que 100% dos usuários participantes da avaliação, demonstram que a aplicação possui mensagens de aviso para o usuário quando há alguma inconformidade na ação.

Figura 46 - Resposta referente aos Diálogos do sistema



Fonte: elaborado pelo autor.

Referente a heurística H9 foi elaborada apenas uma pergunta, disposta na Figura 47, trazendo o resultado obtido na pergunta H9 - se o sistema possui mensagens de erros com linguagem simples? Sendo possível observar pela referida figura, que 67% dos usuários participantes da avaliação, demonstram que a aplicação não possui mensagens de aviso em uma

linguagem de fácil entendimento para o usuário, e apenas 33% dos usuários demonstram que a aplicação atende esse aspecto.

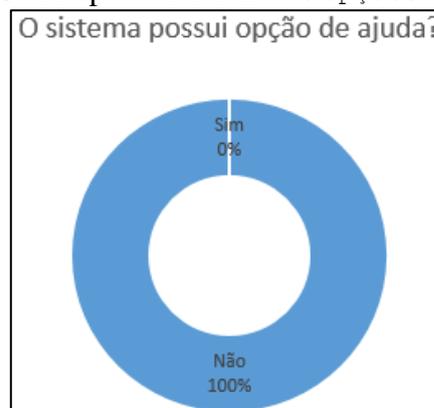
Figura 47 - Resposta referente a Linguagem simples



Fonte: elaborado pelo autor.

Devido a aplicação ter funcionalidades intuitivas, não foi visto a necessidade de elaborar uma opção destinado a uma documentação com descrições detalhadas de cada uma delas. Em decorrência disso, na pergunta referente a heurística H10, o sistema possui opção de ajuda? 100% participantes foram desfavoráveis a ausência de uma opção destinada a documentação dentro da aplicação, conforme demonstrado na Figura 48. Cabe destacar, que a heurística H10 se refere a aplicação ser intuitiva ao ponto de não necessitar de uma tela em específico para ajuda, contudo, usar um agente ou ícone ou avatar para ajudar é medida interessante a ser adotada. A sugestão então foi incluir na aplicação uma página destinada a descrever detalhadamente essas funcionalidades, afim sanar as principais dúvidas nos primeiros usos do usuário.

Figura 48 - Resposta referente as Opções de ajuda

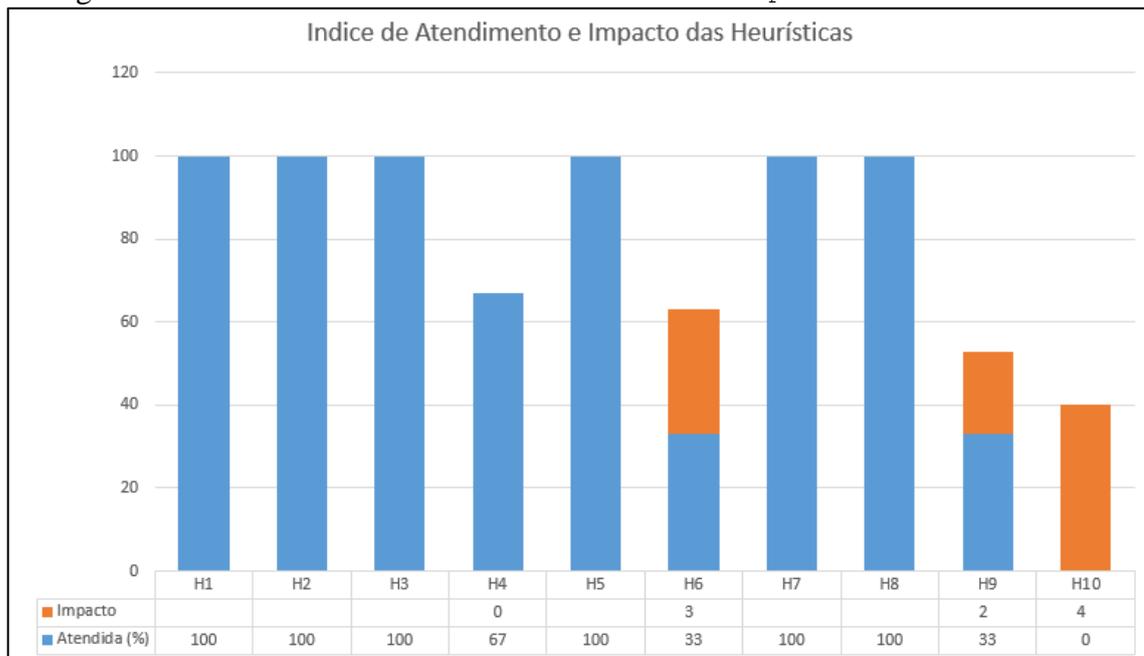


Fonte: elaborado pelo autor.

Com base na análise dos resultados obtidos é possível verificar que a aplicação possui um alto grau de satisfação por todos os usuários participantes na avaliação referente a usabilidade da aplicação, no qual eles relatam se sentir confortável em utilizar a mesma após uma breve explicação das funcionalidades. Outro ponto positivo a ser destacado na aplicação é

referente a nomenclatura adotada pela e a disposição das funcionalidades na aplicação, que segundo os participantes, auxilia na obtenção da informação por manter as opções em fácil acesso e nomeá-las de maneira clara e de fácil entendimento. Para auxiliar a compreensão das informações, foi disposta a Figura 49, contendo informações referente as heurísticas atendidas e mencionadas logo acima.

Figura 49 – Gráfico de Índice de atendimento e impacto das heurísticas



Fonte: elaborado pelo autor.

Porém, com a análise dos resultados obtidos, também pode-se identificar pontos em que a aplicação deve ser aprimorada, visando melhorar a experiência do usuário com a aplicação. Um deles, sendo um dos mais impactantes e indicado pelos três usuários especialistas foi referente a documentação das funcionalidades disponíveis na aplicação (Figura 49); segundo eles, é necessário que a aplicação forneça uma área em que seja possível visualizar informações detalhadas sobre as funcionalidades, afim de auxiliar os primeiros contatos de um usuário com a aplicação. Ademais, pode-se perceber pela Figura 49 que os pontos a serem melhorados na aplicação são de sua maioria relacionados a ausência de informações instrutivas, não impactando diretamente a usabilidade e a funcionalidade da aplicação. Por fim, os usuários especialistas colocaram que a aplicação tem grande potencial de auxiliar os usuários em suas análises diárias e em suas tomadas de decisões.

4 CONCLUSÕES

Este trabalho se justifica devido ao desafio por parte das instituições financeiras em fornecer ferramentas para os usuários Pessoa Jurídica (PJ) visualizarem informações de suas respectivas contas bancárias. Desta forma, se fez necessário o desenvolvimento da aplicação SELBST, no qual é focada nos recursos a serem utilizados por esse grupo específico de usuários. A aplicação visa auxiliá-los a visualizar de forma rápida e eficiente, seus dados bancários, assim como fornecer para as instituições financeiras, uma forma de disponibilizar essas informações, com um baixo custo operacional.

O objetivo geral da aplicação de desenvolver uma aplicação para montar análises financeiras dinâmica de forma integrada (seção 1.1) foi em sua completude atendido, conforme resultado obtido pelos feedbacks dos usuários especialistas, apresentados na seção 3.4. Conseqüentemente, os objetivos específicos também foram atendidos, dos quais os de integrar o banco de dados e o de implementar um processo de extração de dados foram atingidos por meio do desenvolvimento da API; assim como o de identificar os focos de análise, que foi contemplado com a disponibilização da tela de cadastro de regras, no qual o usuário analista disponibiliza as informações que forem convenientes para a instituição. E os de prover ao usuário PJ uma camada analítica, de facilitar a descoberta da relação entre os dados e o de construir e disponibilizar um mecanismo analítico foram contemplados por meio da disponibilização da área de trabalho do usuário, em que ele pode trabalhar as informações da maneira em que desejar.

Com base nas características apresentadas na seção 2.5, nota-se que o trabalho desenvolvido se destaca pela questão de nenhuma das ferramentas apresentadas serem focadas em certo perfil de usuário, podendo assim concluir que as mesmas possuem funcionalidades irrelevantes para o perfil em foco do estudo, podendo até mesmo atrapalhar a experiência com as mesmas. Partindo desse pressuposto, a aplicação de estudo, abstrai as funcionalidades irrelevantes para o perfil do usuário em questão, auxiliando na experiência no uso da aplicação.

Além disso, pode-se destacar a carência de informações que um usuário PJ tem perante seus dados contidos nas instituições financeiras, como no caso da Viacredi. Pelos resultados obtidos, a questão de pesquisa colocada de Conjectura-se que a ferramenta auxilie a suprir esse déficit de informações, trazendo conseqüências positivas tanto para os colaboradores, quanto para a instituição financeira é possível afirmar, que sim, destacando-se as seguintes contribuições:

- a) gerar imagem positiva da instituição financeira, com algo inovador ao usuário;
- b) gerar vantagem competitiva perante a concorrência, tanto para o usuário, quanto

para a instituição financeira;

- c) contribuir socialmente com o usuário, instigando a buscar novos conhecimentos para melhor utilizar os recursos oferecidos pela ferramenta;
- d) contribuir para o empreendimento do usuário, auxiliando nos controles financeiros.

Dessa maneira, a aplicação desenvolvida auxilia brandamente o usuário e seu empreendimento, com informações e visões de negócio até então desconhecidas. Assim como, auxiliar a instituição financeira em conquistar a confiança do perfil de usuário PJ, estreitando o relacionamento entre as partes, para que possam usufruir mais serviços ofertados pela instituição financeira, assim como indicar para outros colaboradores.

4.1 EXTENSÕES

Como extensão deste trabalho sugere-se:

- a) pesquisa de informação por comandos de voz, afim de facilitar a busca por dados específicos, sem que o usuário tenha que confeccionar o *dashboard*;
- b) ampliar a gama de fórmulas e visualizações disponibilizadas na aplicação, possibilitando maior variedade de análises;
- c) permitir que o usuário salve as análises confeccionadas por ele, possibilitando uma consulta posterior nas informações sem ter que as refazer.

REFERÊNCIAS

- ATOM. **O que é dashboard.** [S.I.], 2015. Disponível em: <<http://www.atomsail.com/pt/o-que-e-dashboard-saas-atomsail.php>>. Acesso em: 24 jun. 2018.
- AUSLAND. **Você sabe o que significa Business Intelligence.** [S.I.], 2015. Disponível em: <<http://ausland.com.br/conceito-de-business-intelligence/>>. Acesso em: 24 jun. 2018.
- ÁVILA, Rafael. **Dicas importantes para elaborar um dashboard útil e profissional.** [S.I.], 2014. Disponível em: <<https://blog.luz.vc/excel/dicas-importantes-para-elaborar-um-dashboard-util-e-profissional/>>. Acesso em: 24 jun. 2018.
- BUCHSBAUM, Paulo. **DASHBOARDS: introdução ao conceito. In: Fórum de Varejo, AZOV, 2012.** Disponível em: <<http://www.greatsolutions.com.br/images/BI.pdf>>. Acesso em: 24 jun. 2018.
- CASTRO, Jorge Miguel Loureiro Torres de. **Tendências de Business Intelligence: SSBI como foco principal de estudo.** 2016. 73 f. Dissertação (Mestrado em Gestão de Informação) - Curso de Pós-Graduação, Universidade Nova de Lisboa, Lisboa, 2016.
- CETAX. **Qual a diferença entre Data Warehouse, Big Data e Business Intelligence?** [S.I.], 2016. Disponível em: <<https://www.cetax.com.br/qual-a-diferenca-entre-data-warehouse-big-data-e-business-intelligence/>>. Acesso em: 24 jun. 2018.
- COSTA, Simone Erbs da. **Avaliação de Usabilidade pelas Heurísticas de Nielsen: Método M3C-URUCAg.** iLibrasCollaborative, 2018. Disponível em: <<https://www.ilibrascollaborative.com/noticias-e-recursos>>. Acesso em: 24 jun. 2018.
- DIONÍZIO, Leandro. **Data Warehouse.** [S.I.], 2013. Disponível em: <<http://www.devmedia.com.br/data-warehouse/12609#comentariosArtigo>>. Acesso em: 24 jun. 2018.
- ELIAS, Diego. **Conhecendo a arquitetura de Data Warehouse.** [S.L.], 2014. Disponível em: <<https://canaltech.com.br/business-intelligence/conhecendo-a-arquitetura-de-data-warehouse-19266/>>. Acesso em: 24 jun. 2018.
- FAROL. **Afinal, o que é business intelligence?** [S.I.], 2017. Disponível em: <<http://farolbi.com.br/afinal-o-que-e-self-service-bi/>>. Acesso em: 24 jun. 2018.
- FIVE ACTS. **Análise visual para todos.** [S.I.], 2017. Disponível em: <<http://fiveacts.com.br/solucoes/tableau/>>. Acesso em: 24 jun. 2018.
- KNOW SOLUTIONS. **O que é Business Intelligence (BI)?** [S.I.], 2017. Disponível em: <<http://knowsolution.com.br/o-que-e-business-intelligence-bi/>>. Acesso em: 24 jun. 2018.
- LEITE, Fabiano Luiz Caldas. **Utilização de Business Intelligence para gestão da área operacional de agências bancárias: um estudo de caso.** 2007. 167 f. Dissertação (Mestrado) Curso de Administração, Análise e Tecnologia da Informação, Fundação Getúlio Vargas Escola de Administração de Empresas de São Paulo, São Paulo, 2007.
- LEITE, Fabiano Luiz Caldas; DINIZ, Eduardo Henrique; JAYO, Martin. Utilização de Business Intelligence para gestão operacional de agências bancárias: um estudo de caso. **Revista Eletrônica de Sistemas de Informação**, v. 8, n. 2, p. 1677-3071, [S.I.], 2009.
- NIELSEN, J. Usability engineering. [S.l.]: Elsevier, 1994.
- NOVATO, Douglas. **O que é Business Intelligence?** [S.I.], 2017. Disponível em: <<https://www.oficinadanet.com.br/post/13153-o-que-e-business-intelligence>>. Acesso em: 24 jun. 2018.

OLIVEIRA, Roberto. **O que é self-service BI?** [S.I.], 2014. Disponível em: <<http://smalldata.com.br/o-que-e-self-service-bi/>>. Acesso em: 24 jun. 2018.

ORACLE. **About.** [S.I.], 2016a. Disponível em: <<http://www.oracle.com/webfolder/technetwork/dataviz/index.html>>. Acesso em: 24 jun. 2018.

_____. **Insights com Oracle Data Visualization Desktop.** [S.I.], 2016b. Disponível em: <<http://www.oracle.com/technetwork/pt/articles/bi/insights-data-visualization-desktop-3045139-ptb.html>>. Acesso em: 24 jun. 2018.

_____. **Oracle Data Visualization.** [S.I.], 2016c. Disponível em: <<https://www.oracle.com/solutions/business-analytics/data-visualization/capabilities.html>>. Acesso em: 24 jun. 2018.

PITON, Rafael. **Data Warehouse: o que é?** [S.I.], 2017. Disponível em: <<http://rafaelpiton.com.br/data-warehouse-o-que-e/>>. Acesso em: 24 jun. 2018.

POWER BI. **Azure e Power BI** [S.I.], 2017. Disponível em: <<https://powerbi.microsoft.com/pt-br/documentation/powerbi-azure-and-power-bi/>>. Acesso em: 24 jun. 2018.

PROJECT BUILDER. **8 coisas que você precisa saber sobre dashboard.** [S.I.], 2017. Disponível em: <<https://projectbuilder.com.br/blog-pb/entry/dicas/8-coisas-que-voce-precisa-saber-sobre-dashboard>>. Acesso em: 13 set. 2017.

RAMOS, Wilson. **BI Self-Service, tradicional ou modelo híbrido.** [S.I.], 2016. Disponível em: <<https://www.portalgsti.com.br/2016/11/bi-self-service-tradicional-ou-modelo-hibrido.html>>. Acesso em: 24 jun. 2018.

RIBEIRO, Viviane. **O que é Data Warehouse?** [S.I.], 2011. Disponível em: <<https://vivianeribeiro1.wordpress.com/2011/03/30/o-que-e-data-warehouse/>>. Acesso em: 24 jun. 2018.

ROUSE, Margaret. **Self-service Business Intelligence (BI).** [S.I.], 2016. Disponível em: <<http://www.rpinfo.com.br/businessintelligence>>. Acesso em: 24 jun. 2018.

RP INFO. **Business Intelligence BI.** [S.I.], 2017. Disponível em: <<http://searchbusinessanalytics.techtarget.com/definition/self-service-business-intelligence-BI>>. Acesso em: 24 jun. 2018.

SAS INSTITUTE INC. **SAS Visual Analytics.** [S.I.], 2016. Disponível em: <https://www.sas.com/pt_br/software/business-intelligence/visual-analytics.html>. Acesso em: 24 jun. 2018.

_____. **SAS Visual Analytics: user's guide.** [S.I.], 2014. Disponível em: <https://www.sas.com/content/dam/SAS/en_us/doc/other1/sas-visualanalytics-startup%20guide.pdf>. Acesso em: 24 jun. 2018.

SIGNIFICADOS. **Significado de Data Warehouse.** [S.I.], 2017. Disponível em: <<https://www.significados.com.br/data-warehouse/>>. Acesso em: 24 jun. 2018.

TABLEAU SOFTWARE. **Visão geral.** [S.I.], 2017a. Disponível em: <<https://www.tableau.com/pt-br/products/desktop>>. Acesso em: 24 jun. 2018.

_____. **Etapa 2: arrastar e soltar para ter uma visão preliminar.** [S.I.], 2017b. Disponível em: <<http://onlinehelp.tableau.com/current/guides/get-started-tutorial/pt-br/get-started-tutorial-drag.html>>. Acesso em: 24 jun. 2018.

_____. **Filtrar dados da sua exibição.** [S.I.], 2017c. Disponível em:
<<http://onlinehelp.tableau.com/current/pro/desktop/pt-br/filtering.html#OrderOps>>. Acesso em: 24 jun. 2018.

APÊNDICE A – Detalhamento dos casos de uso

A seguir são descritos detalhadamente os casos de uso da aplicação web, os quais estão ordenados por ordem de importância. Cada detalhamento possui a descrição, um cenário, as pré e pós-condições do caso de uso e um cenário de exceção. O caso de uso UC4 - Processo Batch está descrito no Quadro 24.

Quadro 24 - Caso de uso UC4 - Processo Batch

Número	4
Caso de Uso	Processo Batch
Ator	Sistema Legado
Cenário Principal	1. Ator faz requisição http.
Cenário de Exceção	No passo 1, caso houver alguma falha na integração, será retornado um Json com a descrição técnica do erro.
Pós-condições	O conteúdo será persistido

Fonte: elaborado pelo autor.

O caso de uso UC2 - CRUD usuário está descrito no Quadro 24.

Quadro 25 - Caso de uso UC2 - CRUD usuário

Número	2
Caso de Uso	CRUD Usuário
Ator	Analista de Sistema
Cenário Principal	<ol style="list-style-type: none"> 1. Ator clica na opção Usuário localizada no Header. 2. Ator clica na opção Adicionar. 3. Ator preenche as informações referente ao usuário. 4. Ator clica na opção Criar.
Cenário de Exceção	No passo 4, caso não tenha sido informado, será exibido uma mensagem Campo não pode ser nulo abaixo do campo.
Pós-condições	O conteúdo será persistido

Fonte: elaborado pelo autor.

O caso de uso UC3 - CRUD Regras está descrito no Quadro 26.

Quadro 26 - Caso de uso UC3 - CRUD Regras

Número	3
Caso de Uso	CRUD Regras
Ator	Analista de Sistema
Cenário Principal	<ol style="list-style-type: none"> 1. Ator clica na opção Regras localizada no Header. 2. Ator clica na opção Adicionar. 3. Ator preenche as informações referente a regra. 4. Ator clica na opção Criar.
Cenário de Exceção	No passo 4, caso não tenha sido informado, será exibido uma mensagem Campo não pode ser nulo abaixo do campo.
Pós-condições	O conteúdo será persistido

Fonte: elaborado pelo autor.

O caso de uso UC5 - Selecionar Cubo de dados está descrito no Quadro 27.

Quadro 27 – Caso de uso UC5 – Selecionar Cubo de dados

Número	5
Caso de Uso	Selecionar Cubo de dados
Ator	Usuário
Cenário Principal	<ol style="list-style-type: none"> 1. Ator clica na opção <i>Selecione um Cubo</i> localizada no Header. 2. Ator clica em algum dos itens listados.
Cenário de Exceção	Nenhum
Pós-condições	Será disponibilizado para o usuário as informações referentes ao cubo escolhido.

Fonte: elaborado pelo autor.

O caso de uso UC6 - Alterar forma de visualizar as informações está descrito no Quadro 28.

Quadro 28 - Caso de uso UC6 - Alterar forma de visualizar as informações

Número	6
Caso de Uso	Alterar forma de visualizar as informações
Ator	Usuário
Cenário Principal	<ol style="list-style-type: none"> 1. Ator clica no campo que lista as formas de visualizações. 2. Ator clica em algum dos itens listados.
Cenário de Exceção	Nenhum
Pós-condições	Será apresentado no WorkSpace a forma escolhida.

Fonte: elaborado pelo autor.

O caso de uso UC7 - Aplicar fórmula nas informações disponíveis está disponível no Quadro 29.

Quadro 29 - Caso de uso UC7 - Aplicar fórmula nas informações disponíveis

Número	7
Caso de Uso	Aplicar fórmula nas informações disponíveis
Ator	Usuário
Cenário Principal	<ol style="list-style-type: none"> 1. Ator clica no campo que lista as fórmulas. 2. Ator clica em algum dos itens listados. 3. Ator seleciona a informação que irá aplicar a fórmula
Cenário de Exceção	No passo 3, caso o tipo de informação selecionado não for uma métrica, a aplicação não exibirá nenhum valor.
Pós-condições	Será apresentado no WorkSpace o resultado da fórmula aplicada.

Fonte: elaborado pelo autor.

O caso de uso UC8 - Arrastar campos para o WorkSpace está disponível no Quadro 30.

Quadro 30 - Caso de uso UC8 - Arrastar campos para o WorkSpace

Número	8
Caso de Uso	Arrastar campos para o WorkSpace
Ator	Usuário
Cenário Principal	<ol style="list-style-type: none"> 1. Ator seleciona a informação que deseja exibir. 2. Ator arrasta a informação selecionada para o WorkSpace.
Cenário de Exceção	Nenhum
Pós-condições	Será apresentado no WorkSpace as informações selecionadas.

Fonte: elaborado pelo autor.

APÊNDICE B – Conjunto básico das heurísticas de Nielsen

Este Apêndice contém o conjunto básico das heurísticas de Nielsen, que foram utilizados na avaliação da aplicação desenvolvida, assim como serviram de guias na usabilidade propiciada pela aplicação (Quadro 31).

Quadro 31 - Conjunto básico das Heurísticas de Nielsen

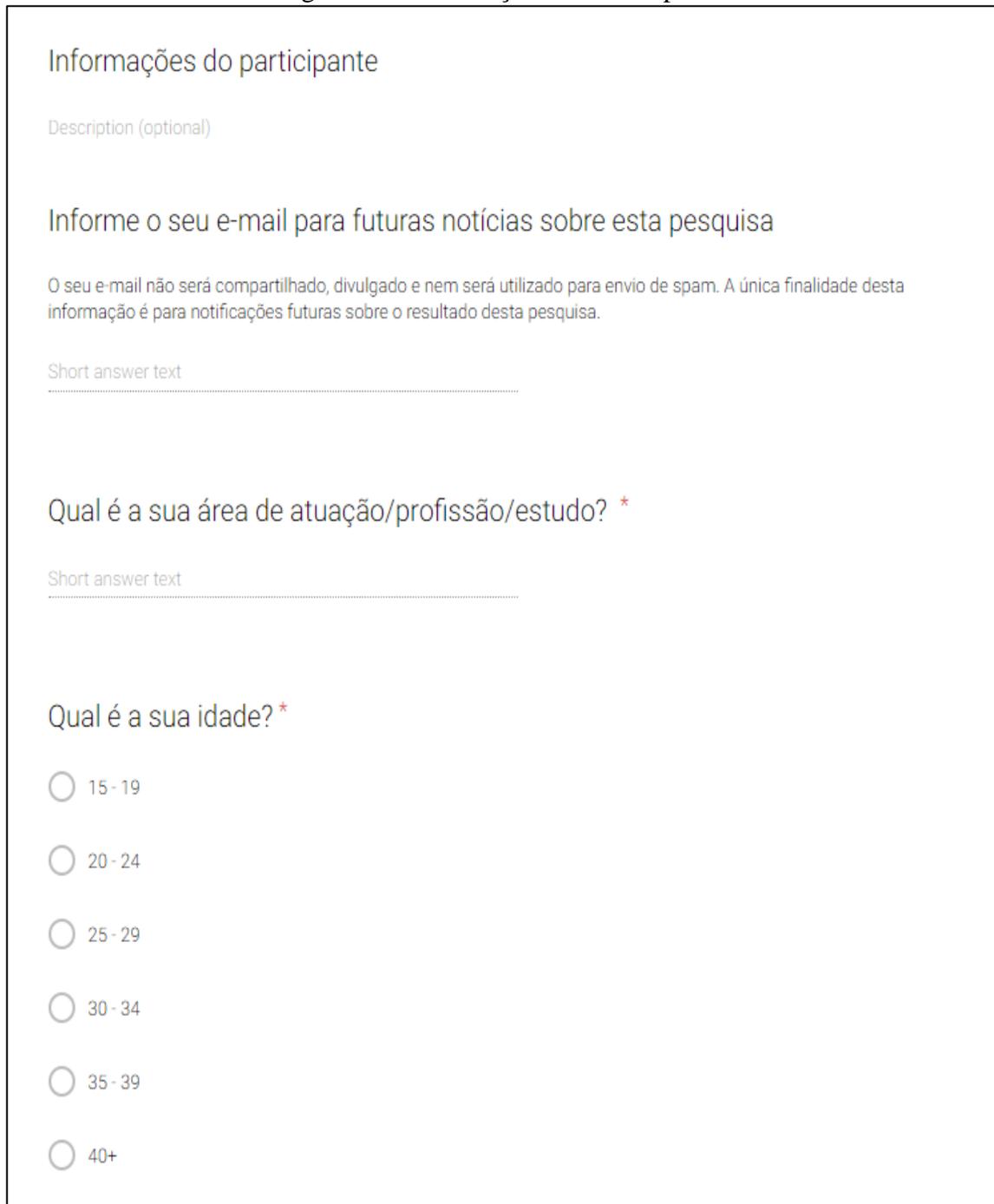
NRO	HEURÍSTICA	CARACTERÍSTICA
H1	Visibilidade do estado do sistema.	O dispositivo deve manter o usuário informado sobre todos os processos e mudanças de estado através do uso de um tipo específico de feedback, em um tempo razoável.
H2	Compatibilidade entre o sistema e o mundo real	O dispositivo deve falar a linguagem dos usuários com palavras, frases e conceitos familiares ao usuário, ao invés de conceitos e/ou técnicas orientados ao sistema. O dispositivo deve seguir as convenções do mundo real, exibindo as informações em uma ordem lógica e natural.
H3	Controle e liberdade do usuário	O dispositivo deve permitir ao usuário desfazer e refazer suas ações, e deve proporcionar "saídas de emergência" para sair do estado indesejado. Essas opções devem ser claramente apontadas, de preferência por meio de um botão físico ou similar; o usuário não deve ser obrigado a passar por um diálogo extenso.
H4	Consistência e padrões	O dispositivo deve seguir as convenções estabelecidas, de forma que o usuário deve ser capaz de fazer as coisas de uma forma familiar, padrão e consistente.
H5	Prevenção de erro	O dispositivo deve ter um design de interface cuidadoso, a fim de evitar erros. As funcionalidades não disponíveis devem ser escondidas ou desativadas e o usuário deve ser capaz de obter informações adicionais sobre todas as funcionalidades disponíveis. Os usuários devem ser avisados quando os erros são prováveis de ocorrer.
H6	Reconhecimento em lugar de lembrança	O dispositivo deve minimizar a carga de memória do usuário, tornando os objetos, ações e opções visíveis. O usuário não deve ter que lembrar de informações de uma parte do diálogo para outra. Instruções para a utilização do sistema devem ser visíveis ou facilmente recuperáveis sempre que necessário.
H7	Flexibilidade e eficiência de uso	A ineficiência das tarefas de usuário pode reduzir a eficácia do usuário e causar-lhes frustrações.
H8	Estética e design minimalista	O dispositivo deve evitar a exibição de informações irrelevantes ou raramente necessárias. Cada informação extra reduz o desempenho do sistema.
H9	Ajuda aos usuários para reconhecer, diagnosticar e se recuperar de erros	Mensagens de erro no dispositivo devem ser expressas em linguagem simples (sem códigos), indicando precisamente o problema e construtivamente sugerindo uma solução.
H10	Ajuda e documentação	O dispositivo deve fornecer uma documentação e ajuda fácil de encontrar, centrada na atividade atual do usuário. Uma lista de passos concretos (e não muito grandes) para realizar deve ser fornecida.

Fonte: elaborado pelo autor com base em Nielsen, 1994.

APÊNDICE C – Questionário de Usabilidade

A seguir, na Figura 50, é apresentado o questionário de usabilidade elaborado utilizando o Google Forms, o qual três participantes responderam após o teste da aplicação do SELBST.

Figura 50 - Informações do Participante



The image shows a Google Form titled "Informações do participante". It includes a description field, a request for an email address with a privacy notice, and three required questions: "Qual é a sua área de atuação/profissão/estudo?", "Qual é a sua idade?", and a list of age ranges (15-19, 20-24, 25-29, 30-34, 35-39, 40+).

Informações do participante

Description (optional)

Informe o seu e-mail para futuras notícias sobre esta pesquisa

O seu e-mail não será compartilhado, divulgado e nem será utilizado para envio de spam. A única finalidade desta informação é para notificações futuras sobre o resultado desta pesquisa.

Short answer text

Qual é a sua área de atuação/profissão/estudo? *

Short answer text

Qual é a sua idade? *

15 - 19

20 - 24

25 - 29

30 - 34

35 - 39

40+

Fonte: elaborado pelo autor.

Figura 51 – Heurística H1 - Visibilidade do Sistema

Section 4 of 13

H1 - Visibilidade do Sistema

Description (optional)

1 - O sistema possui um feedback rápido indicando o que você está fazendo na interface no momento? *

Sim

Não

1 - Impacto

0 1 2 3 4

Pouco Impactante Muito Impactante

2 - O sistema possui um feedback rápido indicando o cubo que você selecionou? *

Sim

Não

2 - Impacto

0 1 2 3 4

Pouco Impactante Muito Impactante

Fonte: elaborado pelo autor.

Figura 52 – Heurística H2 - Correspondência entre o sistema e o mundo real

Section 5 of 13

H2 - Correspondência entre o Sistema e o Mundo Real

Description (optional)

3 - O sistema utiliza palavras, termos, expressões e conceitos familiares para você? *

Sim

Não

3 - Impacto

	0	1	2	3	4	
Pouco Impactante	<input type="radio"/>	Muito Impactante				

4 - As informações aparecem em uma ordem lógica e natural para o seu entendimento? *

Sim

Não

4 - Impacto

	0	1	2	3	4	
Pouco Impactante	<input type="radio"/>	Muito Impactante				

Fonte: elaborado pelo autor.

Figura 53 – Heurística H3 - Controle e liberdade do usuário

Section 6 of 13

H3 - Controle e Liberdade do Usuário

Description (optional)

5 - O sistema possibilita que você saia da opção selecionada a qualquer momento? *

Sim

Não

5 - Impacto

	0	1	2	3	4	
Pouco Impactante	<input type="radio"/>	Muito Impactante				

Fonte: elaborado pelo autor.

Figura 54 – Heurística H4 - Consistência e padrões

Section 7 of 13

H4 - Consistência e Padrões

Description (optional)

6 - As palavras, situações ou ações no sistema são claras, possibilitando o seu *
entendimento?

Sim

Não

6 - Impacto

	0	1	2	3	4	
Pouco Impactante	<input type="radio"/>	Muito Impactante				

7 - O sistema possui padrões e estilos consistentes? *

Sim

Não

7 - Impacto

	0	1	2	3	4	
Pouco Impactante	<input type="radio"/>	Muito Impactante				

Fonte: elaborado pelo autor.

Figura 55 – Heurística H5 - Prevenção de erros

Section 8 of 13

H5 - Prevenção de Erros

Description (optional)

8 - O sistema possui ícones que ajudam a impedir a ocorrência de erros? *

Sim

Não

8 - Impacto

	0	1	2	3	4	
Pouco Impactante	<input type="radio"/>	Muito Impactante				

Fonte: elaborado pelo autor.

Figura 56 - Heurística H6 - Reconhecimento no lugar de memorização

Section 9 of 13

H6 - Reconhecimento no Lugar de Memorização

Description (optional)

9 - O sistema possui instruções, ações e opções visíveis ou facilmente recuperáveis (sempre que apropriado para o uso)? *

Sim

Não

9 - Impacto

	0	1	2	3	4	
Pouco Impactante	<input type="radio"/>	Muito Impactante				

Fonte: elaborado pelo autor.

Figura 57 - Heurística H7 - Flexibilidade e eficiência de uso

Section 10 of 13

H7 - Flexibilidade e Eficiência de Uso

Description (optional)

11 - O sistema possui teclas, imagens ou ícones para facilitar o seu uso? *

Sim

Não

11 - Impacto

	0	1	2	3	4	
Pouco Impactante	<input type="radio"/>	Muito Impactante				

Fonte: elaborado pelo autor.

Figura 58 - Heurística H8 - Desenho estético e minimalista

Section 11 of 13

H8 - Desenho Estético e Minimalista

Description (optional)

12 - Os diálogos do sistema trazem informações relevantes ou na maioria das vezes necessárias? *

Sim

Não

12 - Impacto

	0	1	2	3	4	
Pouco Impactante	<input type="radio"/>	Muito Impactante				

Fonte: elaborado pelo autor.

Figura 59 - Heurística H9 - Auxílio no reconhecimento, diagnóstico e recuperação de erros

Section 12 of 13

H9 - Auxílio no Reconhecimento, Diagnóstico e Recuperação de Erros

Description (optional)

13 - O sistema possui mensagens de erros com linguagem simples? *

Sim

Não

13 - Impacto

	0	1	2	3	4	
Pouco Impactante	<input type="radio"/>	Muito Impactante				

Fonte: elaborado pelo autor.

Figura 60 - Heurística H10 - Ajuda e Documentação

Section 13 of 13

H10 - Ajuda e Documentação

Description (optional)

14 - O sistema possui opção de ajuda? *

Sim

Não

14 - Impacto

	0	1	2	3	4	
Pouco Impactante	<input type="radio"/>	Muito Impactante				

Fonte: elaborado pelo autor.

APÊNDICE D – Roteiro de teste

Este Apêndice contém a apresentação do roteiro de teste repassado para o usuário testar a aplicação SELBST e prestar sua avaliação (Figura 61).

Figura 61 - Roteiro de Teste

<p>1 - Faça login na aplicação com os dados abaixo: usuário = usuario2@uso.com.br senha = senha123</p> <p>2 - Selecione a área de assunto CARTEIRA_CREDITO</p> <p>Visualizar a soma da carteira normal no período de um mês agrupados por linha de crédito:</p> <p>3 - Utilize a opção de visualização Table Heatmap</p> <p>3.1 - Utilize a fórmula SUM selecionando o campo carteira_normal para gerar a visualização</p> <p>3.2 - No eixo Y arraste o campo Data</p> <p>3.2.1 - Selecione o período de um mês clicando na seta ao lado do campo Data</p> <p>3.3 - No eixo X arraste o campo linha de crédito</p> <p>Visualizar a soma da carteira total por linha de crédito nos fechamentos de mês</p> <p>4 - Utilize a opção de visualização Grouped Column Chart</p> <p>4.1 - Utilize a fórmula SUM selecionando o campo carteira_total para gerar a visualização</p> <p>4.2 - No eixo Y arraste o campo Data</p> <p>4.2.1 - Selecione três dias de fechamento de mês clicando na seta ao lado do campo Data</p> <p>4.3 - No eixo X arraste o campo linha de crédito</p> <p>4.4 - Utilize a legenda para ocultar dados de algum dia em específico clicando em cima do mesmo</p> <p>4.5 - Invertar os campos de eixo</p> <p>4.6 - Coloque os campos no mesmo eixo</p> <p>Visualizar a soma da provisão total da carteira por linha de crédito nos fechamentos de mês</p> <p>5 - Utilize a opção de visualização Stacked Bar Chart</p> <p>5.1 - Utilize a fórmula SUM selecionando o campo carteira_provisao_total para gerar a visualização</p> <p>5.2 - No eixo Y arraste o campo Data</p> <p>5.2.1 - Selecione três dias de fechamento de mês clicando na seta ao lado do campo Data</p> <p>5.3 - No eixo X arraste o campo linha de crédito</p> <p>5.4 - Utilize a legenda para ocultar dados de algum dia em específico clicando em cima do mesmo</p> <p>5.5 - Invertar os campos de eixo</p> <p>5.6 - Coloque os campos no mesmo eixo</p> <p>Visualizar a evolução da soma do valor da dívida por linha de crédito nos fechamentos de mês</p> <p>6 - Utilize a opção de visualização Line Chart</p> <p>6.1 - Utilize a fórmula SUM selecionando o campo valor_divida para gerar a visualização</p> <p>6.2 - No eixo X arraste o campo Data</p> <p>6.2.1 - Selecione três dias de fechamento de mês clicando na seta ao lado do campo Data</p> <p>6.3 - Também no eixo X arraste o campo linha de crédito</p> <p>Explorar as opções disponíveis na aplicação</p> <p>7 - Utilize as funcionalidades disponíveis no canto superior direito do gráfico para poder trabalhar com o mesmo</p> <p>8 - Explore outras áreas de assuntos</p> <p>9 - E também outras formas de visualização de informações</p>

Fonte: elaborado pelo autor.

APÊNDICE E – Termo de consentimento Livre e Esclarecido (TCLE)

Este apêndice contém o termo de consentimento disponibilizado para o usuário antes de iniciar a avaliação de usabilidade da aplicação (Figura 62).

Figura 62 - Termo de consentimento Livre e Esclarecido

<p>TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO</p> <p>O(a) senhor(a) está sendo convidado a participar de uma pesquisa de graduação de uma aplicação intitulada SELBST. A aplicação avaliada é um projeto desenvolvido pelo acadêmico Geovani Finoti Leitão com o intuito de desenvolver uma ferramenta para montar análises financeiras dinâmicas de forma integrada. A pesquisa busca avaliar a aplicação SELBST a partir da realização de tarefas predefinidas e, posteriormente, da realização da avaliação de usabilidade e experiência da aplicação. Esta avaliação servirá como base para as futuras melhorias e mudanças que a aplicação avaliada possa sofrer, além de levantar a viabilidade da continuação do projeto.</p> <p>O professor regente fará a explicação da forma que será aplicado a avaliação. Como esta é uma participação voluntária, o(a) Senhor(a) e seu/sua acompanhante não terão despesas e nem serão remunerados pela participação na pesquisa. Todas as despesas decorrentes de sua participação serão ressarcidas. Em caso de danos, decorrentes da pesquisa será garantida a indenização. Os possíveis desconfortos e riscos decorrentes do estudo, levando-se em conta que é uma pesquisa, e os resultados positivos ou negativos somente serão obtidos após a sua realização. Assim, estou sujeito a realização de tarefas pré-definidas e especificadas no formulário de avaliação. Além disso, a minha avaliação poderá ou não ser considerada no resultado final do aplicativo, dependendo da forma que eu estarei respondendo minha avaliação.</p> <p>Estou ciente que minha privacidade será respeitada, ou seja, meu nome ou qualquer outro dado ou elemento que possa, de qualquer forma, me identificar, será mantido em sigilo. Também fui informado que eu posso me recusar a participar do estudo, ou retirar meu consentimento a qualquer momento, sem precisar justificar, e que, por desejar sair da pesquisa, não sofrerei qualquer prejuízo.</p> <p>Os pesquisadores envolvidos no estudo são, Geovani Finoti Leitão, que posso entrar em contato pelo e-mail geovani.leitao@gmail.com e da orientadora da pesquisa Simone Erbs da Costa, que posso entrar em contato pelo e-mail simoneerbsdacosta@gmail.com. Além disso, é assegurada toda assistência durante toda a pesquisa, bem como me é garantido o livre acesso a todas as informações e esclarecimentos adicionais sobre o estudo e suas consequências, ou seja, tudo que eu queria saber antes, durante e depois da minha participação.</p> <p>Dessa forma, tendo sido orientado quanto ao teor de todo aqui mencionado e compreendido a natureza e o objetivo do referido estudo, manifesto meu livre consentimento em participar, estando totalmente ciente de que não existe nenhum valor econômico, a receber ou a pagar, por minha participação. Caso exista algum dano decorrente a minha participação no estudo, serei devidamente indenizado conforme determina a lei.</p> <p>Em caso de reclamação ou qualquer outra denúncia sobre esse estudo devo entrar em contato com a orientadora da pesquisa Simone Erbs da Costa pelo e-mail simoneerbsdacosta@gmail.com. Os benefícios e vantagens em participar deste estudo estão relacionados a contribuir para o uso de novas metodologias em salas de aula, tornando o ambiente de ensino aprendizagem mais próximo da realidade que estou inserido. A pessoa que estará acompanhando os procedimentos será o professor regente da disciplina que o questionário está sendo aplicado. O(a) senhor(a) poderá se retirar do estudo a qualquer momento, sem qualquer tipo de constrangimento. Solicitamos a sua autorização para o uso de seus dados para a produção de artigos técnicos e científicos.</p> <p>A sua privacidade será mantida por meio da não-identificação do seu nome. Caso esteja de acordo, com o termo basta seguir o roteiro da pesquisa na próxima seção e posteriormente responder as perguntas do questionário da avaliação.</p>
--

Fonte: elaborado pelo autor.