

UNIVERSIDADE ESTADUAL DO SUDOESTE DA
BAHIA CAMPUS VITÓRIA DA CONQUISTA
DEPARTAMENTO CIÊNCIAS EXATAS
CURSO CIÊNCIA DA COMPUTAÇÃO

Lucas Santos Figueiredo

Design System: Uma abordagem que traz consistência visual em Sistemas de Software Interativo.

Vitória da Conquista
2025

Lucas Santos Figueiredo

Design System: Uma abordagem que traz consistência visual em Sistemas de Software Interativo.

Trabalho de Conclusão de Curso submetido ao curso de Ciência da Computação do Campus Vitória da Conquista da Universidade Estadual do Sudoeste da Bahia como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Gidevaldo Novais dos Santos.

Vitória da Conquista

2025

Lucas Santos Figueiredo

Design System: Uma abordagem que traz consistência visual em Sistemas de Software Interativo.

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel e aprovado em sua forma final pelo Curso Ciência da Computação.

Local: Vitória da Conquista, 09 de Julho de 2025.

Banca examinadora

Prof. Gidevaldo Novais do Santos, Dr
Orientador

Prof.(a) Maísa Soares dos Santos Lopes, Dr.(a)
Instituição Universidade Estadual do Sudoeste da Bahia

Prof.(a) Maria Luísa Ghizone González, Mestra
Instituição Universidade Estadual do Sudoeste da Bahia

Vitória da Conquista, 2025

DEDICATÓRIA

Dedico esse trabalho primeiramente a minha família, em especial aos meus pais, por todo o sacrifício que fizeram para que eu tivesse as melhores condições de estudo. Além da minha família, gostaria de dedicar também aos meus amigos Enzo Lucca, João Pedro, Aleksander Souza e Rodrigo Carmo que foram fundamentais para chegar nessa etapa do curso, o companheirismo de vocês levarei para o resto da vida. E por fim, a minha esposa, Bruna Maria, por ter acompanhado todo o processo de confecção deste trabalho mesmo não tendo nenhuma relação com a área.

AGRADECIMENTOS

Agradeço ao orientador Gidevaldo Novais dos Santos por suas instruções durante a confecção deste trabalho, sem os teus direcionamentos sobre qual metodologia seria melhor e métodos para avaliar os resultados eu não teria conseguido concluir esta etapa do curso de Ciência da Computação de forma tão exemplar.

Além do meu orientador gostaria de agradecer também aquelas pessoas que foram referências e me ajudaram a estudar sobre o assunto de Design System. Como o querido Filipe Nzongo, que sempre está contribuindo para o avanço deste tema tanto no ambiente acadêmico e corporativo. Anna Takahashi e Poe Bellentani com seus conteúdos em vídeo sobre Design Tokens e Design System para vários temas. Tais profissionais impactaram fundamentalmente para o meu entendimento e aplicação do assunto.

RESUMO

Em cenários corporativos onde são desenvolvidos softwares interativos com o objetivo de resolver problemas específicos, as questões que envolvem o negócio e a viabilidade técnica do que está sendo desenvolvido ganham prioridades acima de fatores que impactam o usuário final, podendo assim levar problemas de consistência visual, usabilidade e affordance nas interfaces gráficas do usuário. Dessa forma, essa pesquisa busca resolver a seguinte questão em um contexto específico: *Como manter a consistência visual em um Sistema de Software Interativo, através da abordagem de Design System, para empresas privadas que investem em transformação digital?*, tendo como método a Design Science Research. Com esse objetivo guiando a pesquisa, questões foram levantadas em relação a como garantir que os processos feitos na fase do projeto estivessem conectados com os processos da camada de engenharia e também como iríamos chegar em uma linguagem comum quanto aos componentes utilizados para construir as interfaces gráficas do usuário. Através da criação de um Design System conseguimos chegar nesse ponto comum, trazendo em um primeiro momento impacto imediato na fase do projeto com melhorias em processos e também na fase da confecção em engenharia, garantindo um código mais coeso, e a sua reutilização.

Palavras-chave: Design System; Interface Gráfica do Utilizador; Interação Humano-Máquina; Front-end; Engenharia de Software.

ABSTRACT

In corporate environments where interactive software is developed to address specific problems, business considerations and the technical feasibility of ongoing development often take precedence over factors that directly impact the end user. Such prioritization can result in issues concerning visual consistency, usability, and affordance within graphical user interfaces. This study seeks to address the following research question in a specific context: *How can visual consistency be maintained in an Interactive Software System, through the Design System approach, within private companies investing in digital transformation?* The research is grounded in the Design Science Research methodology. Guided by this objective, the investigation explored how to ensure that processes undertaken during the design phase remained aligned with those in the engineering layer, and how to establish a single source of truth regarding the components used to construct graphical user interfaces. The creation of a Design System enabled the establishment of this common ground, yielding an immediate impact during the design phase through process improvements, and subsequently enhancing the engineering phase by ensuring more cohesive code and facilitating its reusability.

Keywords: Design System; Graphical User Interface; Human-Machine Interaction; Front-end; Software Engineering.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Atomic Design..... | 20 |
| Figura 2 – Poster, München Olympics, 1972, Otl Aicher..... | 21 |
| Figura 3 – Interface do GEDES..... | 24 |
| Figura 4 – Xerox Star..... | 28 |
| Figura 5 – Iconografia confeccionada por David Canfield Smith | 29 |
| Figura 6 – Menu Superior Google Docs..... | 31 |
| Figura 7 – Menu Superior Google Presentation..... | 31 |
| Figura 8 – Menu Superior Google Sheets..... | 32 |
| Figura 9 – Página inicial site Kabum!..... | 32 |
| Figura 10 – Página do produto site Kabum!..... | 32 |
| Figura 11 – Relação entre as camadas de um Design System..... | 37 |
| Figura 15 – Textfield do Carbon Design System, DS da IBM..... | 41 |
| Figura 19 – Metodologia proposta nesta pesquisa..... | 52 |
| Figura 20 – Mapa de Stakeholders..... | 55 |
| Figura 21 – Demonstrativo visual dos componentes na biblioteca de engenharia..... | 67 |
| Figura 22 – Aspecto de contraste do componente Button e Badge..... | 68 |
| Figura 23 – Comparativo do componente Searchfield entre Engenharia e Design..... | 69 |
| Figura 24 – Comparativo do componente Toast entre Engenharia e Design..... | 69 |
| Figura 25 – Comparativo do componente Textfield entre Engenharia e Design..... | 70 |
| Figura 26 – Comparativo do componente Tooltip entre Engenharia e Design..... | 70 |
| Figura 27 – Comparativo do componente Badge entre Engenharia e Design..... | 71 |
| Figura 28 – Comparativo do componente Button entre Engenharia e Design..... | 71 |
| Figura 29 – Comparativo do componente Radiobutton entre Engenharia e Design..... | 72 |
| Figura 30 – Comparativo do componente Checkbox entre Engenharia e Design..... | 72 |
| Figura 31 – Roadmap execução das demandas com foco em Design..... | 74 |
| Figura 32 – Roadmap execução das demandas com foco em Design e Engenharia..... | 74 |
| Figura 33 – Roadmap execução das demandas com foco em Avaliação..... | 75 |
| Figura 35 – Interface Gráfica do Tokens Studio..... | 80 |
| Figura 36 – Estrutura de JSON construída pelo Tokens Studio..... | 81 |
| Figura 37 – Conteúdo do arquivo JSON de exemplo..... | 82 |
| Figura 38 – Definindo configuração de transformação..... | 82 |
| Figura 39 – Importando construtor..... | 83 |
| Figura 40 – Arquivo com o conteúdo inicial convertido..... | 83 |
| Figura 41 – Definindo configuração de transformação para flutter..... | 84 |
| Figura 42 – Definindo configuração de transformação para android e ios..... | 84 |
| Figura 43 – Definindo configuração de transformação para reac app..... | 85 |
| Figura 44 – Processo de transformação dos Design Tokens em variáveis específicas..... | 85 |
| Figura 45 – Interações no componente Button em Carbon Design System..... | 87 |
| Figura 46 – Interações no componente Button em Dell Design System..... | 87 |
| Figura 47 – Apresentação das propriedades do Componente Button..... | 88 |
| Figura 48 – Opções de variações de tamanho..... | 88 |
| Figura 49 – Opções de variações de estados de interação..... | 89 |

| | |
|---|-----|
| Figura 50 – Representação dos Design Tokens utilizados no componente Button..... | 90 |
| Figura 51 – Recorte das propriedades do componente Button..... | 90 |
| Figura 52 – Recorte anatomia documentado para o componente Button..... | 91 |
| Figura 53 – Recorte Estados de Interação documentados para o componente Button..... | 92 |
| Figura 54 – Conexão entre Design e Engenharia..... | 93 |
| Figura 55 – Componentes entregues na primeira versão..... | 112 |
| Figura 56 – Componentes antes da primeira versão..... | 112 |
| Figura 57 – Atualização visual do Design System..... | 113 |

LISTA DE QUADROS

| | |
|---|-----|
| Quadro 1 – Categorias de Stakeholders..... | 54 |
| Quadro 2 – Dificuldades evidenciadas pelas Designers..... | 58 |
| Quadro 3 – Necessidades evidenciadas pelas Designers..... | 59 |
| Quadro 4 – Expectativas evidenciadas pelas Designers..... | 59 |
| Quadro 5 – Dificuldades evidenciadas pelas Engenheiros..... | 59 |
| Quadro 6 – Contexto atual na visão de Engenharia..... | 61 |
| Quadro 7 – Contexto atual relacionado ao Marketing da empresa..... | 62 |
| Quadro 8 – Exemplo prático da composição da nomenclatura dos Design Tokens..... | 64 |
| Quadro 9 – Lista de componentes levantados na Auditoria..... | 66 |
| Quadro 10 – Lista de opções para cada nível de nomenclatura dos Design Tokens, parte 1... 76 | |
| Quadro 11 – Lista de opções para cada nível de nomenclatura dos Design Tokens, parte 2 77 | |
| Quadro 12 – Exemplo prático da composição da nomenclatura dos Design Tokens..... | 77 |
| Quadro 14 – Hipóteses para validação..... | 97 |
| Quadro 15 – Frequência de ocorrências de cada tópico..... | 100 |
| Quadro 16 – Validação das hipóteses..... | 102 |
| Quadro 17 – Hipóteses para validação..... | 106 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|-----|-------------------------|
| DS | Design System |
| DSR | Design Science Research |

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO..... | 16 |
| 2 ESTADO DA ARTE..... | 20 |
| 3 REFERENCIAL TEÓRICO E METODOLÓGICO..... | 25 |
| 3.1 SISTEMA DE SOFTWARE..... | 25 |
| 3.2 INTERAÇÃO, INTERFACE, USABILIDADE E AFFORDANCE..... | 27 |
| 3.2.1 INTERAÇÃO E INTERFACE..... | 27 |
| 3.2.2 USABILIDADE E AFFORDANCE..... | 29 |
| 3.3 CONSISTÊNCIA E PADRONIZAÇÃO..... | 30 |
| 3.3.1 CONSISTÊNCIA INTERNA E EXTERNA..... | 31 |
| 3.3.2 CAMADAS DE CONSISTÊNCIA..... | 33 |
| 3.4 DESIGN SYSTEM..... | 35 |
| 3.4.1 O ECOSISTEMA DE UM DESIGN SYSTEM..... | 36 |
| 1. CORE DESIGN SYSTEM (NÚCLEO DO SISTEMA DE DESIGN)..... | 37 |
| a. DESIGN TOKENS (TOKENS DE DESIGN)..... | 38 |
| b. ÍCONES..... | 39 |
| c. COMPONENTES DE UI..... | 39 |
| 2. TECHNOLOGY-SPECIFIC IMPLEMENTATION (IMPLEMENTAÇÃO ESPECÍFICA DE TECNOLOGIA)..... | 42 |
| 3. CAMADA DE “RECEITA” (RECIPES) E COMPONENTES INTELIGENTES (SMART COMPONENTS)..... | 43 |
| 4. CAMADA DE PRODUTO..... | 44 |
| 3.4.2 CONSIDERAÇÕES DO ECOSISTEMA DE DESIGN..... | 44 |
| 4 METODOLOGIA..... | 46 |
| 4.1 DEFINIÇÃO DO PROBLEMA..... | 46 |
| 4.2 ANÁLISE E DIAGNÓSTICO..... | 47 |
| 4.3 DEFINIÇÃO DOS RESULTADOS ESPERADOS..... | 48 |
| 4.4 PROJETO E DESENVOLVIMENTO..... | 48 |
| 4.5 INTERVENÇÃO..... | 50 |
| 4.6 APRENDIZAGEM E AVALIAÇÃO..... | 50 |
| 4.7 COMUNICAÇÃO..... | 50 |
| 5 ANÁLISE E RESULTADOS..... | 52 |
| 5.1 PREPARAÇÃO..... | 53 |
| 5.1.1 ENTREVISTAS SEMIESTRUTURADAS..... | 56 |
| 1. PESSOAS UTILIZADORAS DO DESIGN SYSTEM..... | 57 |
| a. DESIGNERS..... | 58 |
| b. ENGENHEIRAS..... | 59 |
| 2. PESSOAS CONSTRUTORAS..... | 60 |
| a. ENGENHEIRA..... | 60 |
| b. REPRESENTANTE DE MARKETING..... | 61 |
| 3. ANÁLISE DAS ENTREVISTAS SEMIESTRUTURADAS..... | 63 |
| a. PRÉ-ANÁLISE..... | 64 |
| b. EXPLORAÇÃO DO MATERIAL (CODIFICAÇÃO & CATEGORIZAÇÃO)..... | 64 |

| | |
|---|------------|
| c. TRATAMENTO, INFERÊNCIA E INTERPRETAÇÃO..... | 65 |
| d. RECOMENDAÇÕES INICIAIS..... | 66 |
| 5.1.2 AUDITORIA DOS COMPONENTES..... | 66 |
| 5.1.3 ROADMAP DO PROJETO..... | 73 |
| 5.2 CONSTRUÇÃO..... | 75 |
| 5.2.1 DEFINIÇÃO DA LINGUAGEM ÚNICA..... | 75 |
| 5.2.2 CONEXÃO DA LINGUAGEM ÚNICA..... | 79 |
| 5.2.3 CONSTRUINDO COMPONENTES..... | 86 |
| 1. CONTEXTO DE DESIGN..... | 86 |
| 2. CONTEXTO DE ENGENHARIA..... | 92 |
| 5.3 USO E AVALIAÇÃO DO DESIGN SYSTEM..... | 94 |
| 5.3.1 CONTEXTO DE DESIGN..... | 94 |
| 1. INTRODUÇÃO À INTERVENÇÃO..... | 94 |
| 2. AVALIAÇÃO DA INTERVENÇÃO..... | 95 |
| a. Fase 1 – Pré-análise..... | 96 |
| b. Fase 2 – Exploração do material (codificação e categorização)..... | 99 |
| c. Fase 3 – Tratamento dos resultados, inferência e interpretação..... | 100 |
| i. Síntese interpretativa por categoria..... | 100 |
| d. Fase 4 - Considerações sobre a análise..... | 103 |
| 5.3.2 CONTEXTO DE ENGENHARIA..... | 104 |
| 1. INTRODUÇÃO À INTERVENÇÃO..... | 104 |
| 2. AVALIAÇÃO DA INTERVENÇÃO..... | 105 |
| a. Pré-análise..... | 105 |
| b. Exploração do material (codificação e categorização)..... | 107 |
| c. Tratamento, inferência e interpretação..... | 108 |
| d. Validação das hipóteses..... | 108 |
| e. Recomendações imediatas..... | 109 |
| 5.3.3 ANÁLISE DE AMBOS OS CONTEXTOS APÓS USO DA INTERVENÇÃO.... | 110 |
| 6 CONCLUSÃO..... | 112 |
| REFERÊNCIAS..... | |
| APÊNDICE A - EXEMPLO DE APLICAÇÃO DA NOMENCLATURA DEFINIDA..... | |
| APÊNDICE B - ROTEIRO DE PERGUNTAS DA PESQUISA INICIAL..... | |
| APÊNDICE C - ROTEIRO DE PERGUNTAS DA AVALIAÇÃO NO CONTEXTO DE DESIGN | |
| 124 | |
| APÊNDICE D - ROTEIRO DE PERGUNTAS DA AVALIAÇÃO..... | |

1 INTRODUÇÃO

Tratando-se de empresas privadas inseridas na economia brasileira, a transformação digital tem obtido cada vez mais importância para as mesmas, tendo como objetivo alcançar um diferencial competitivo em uma economia cada vez mais globalizada (Rafael Ribeiro, analista de Produtividade e Inovação da ABDI). Como podemos notar na pesquisa realizada pela Agência Brasileira de Desenvolvimento Industrial (ABDI) e a Fundação Getúlio Vargas (FGV), a “Sondagem sobre Transformação Digital das Empresas”, evidencia que as empresas estão aumentando e continuando os seus investimentos em digitalização, onde das 4 mil empresas que responderam a pesquisa, mais de 24% delas relataram um aumento nos investimentos em digitalização no segundo semestre de 2023.

Com esse comportamento esperado das empresas, conseguimos perceber que boa parte desses investimentos são direcionados para Sistemas de Software Interativo, os quais são “compostos por hardware, software e meios de comunicação desenvolvidos para interagirem com pessoas” (Barbosa, Simone. 2010, p. 02), que dependendo de cada empresa, possa vir a ser um produto digital, software como serviço e até mesmo serviços potencializados por software, como o aplicativo Uber, Ifood e Zé Delivery. Além desses tipos, pode haver também os Softwares para sustentação de processos internos de tais empresas, comumente chamado de *backoffice* e academicamente de Sistemas de Informação.

O fato das empresas estarem investindo mais em se transformarem digitalmente não é garantia de que conseguirão atingir seus objetivos, dado que o contexto de desenvolvimento de software envolve diversos fatores, como é evidenciado por Barbosa, Simone & Silva Bruno, 2010, no livro Interação Humano Computador. Para eles há uma diferença sutil, mas importante, entre o que um sistema interativo deve permitir fazer, o que o sistema de fato permite fazer e a maneira como ele é utilizado. Ou seja, há uma relação de três principais atores nesse contexto, quem é o dono do sistema, quem constroi o sistema e quem utiliza no dia a dia, dependendo de como seja a abordagem das relações entre esses três atores, o Sistema de Software Interativo poderá ser concebido de forma qualificada ou não.

Da relação entre o dono do sistema - empresas privadas - e quem constroi o sistema - equipe multidisciplinar, contendo Engenheiros de Software, Designers, Gestores de Produto e Agilistas, é perceptível o quão próxima foi se tornando essa relação, visando a adequação de processos de desenvolvimento de softwares nas expectativas de quem contrata, como exemplo disso, temos o manifesto ágil, que teve como objetivo simplificar e tornar mais palpável tais processos para o contexto de negócio.

Porém quando introduzimos o terceiro ator - quem irá utilizar o software no dia a dia - tanto a empresa e quanto quem constroi o software, acabam que não se aprofundam o suficiente na visão deste ator, e assim realizando uma abordagem “de dentro para fora, concebendo primeiro representações de dados, algoritmos que processam esses dados, arquitetura do sistema e tudo mais que permite um sistema Interativo funcionar” (Barbosa, Simone & Silva Bruno, 2010). E assim deixando de lado critérios de Usabilidade de Nielsen, Acessibilidade e *Affordance*, os quais estão dentro da Natureza da Interação do Humano-Máquina.

Ao deixar de lado, por exemplo, o critério de usabilidade, o Sistema a ser desenvolvido começará a possuir problemas, levando o usuário a não conseguir aprender a usar o sistema, ter menos eficiência na tarefa realizada, diminuição da satisfação, não recordar a maneira de uso e não se sentir seguro a usar o sistema, (Nielsen, 1988). Um fator que impacta diretamente na usabilidade é a 4ª heurística de Nielsen, sobre consistências e padrões, onde o autor afirmou a primeira vez em seu artigo “10 Usability Heuristics for User Interface Design” que os usuários não devem se questionar se tal ação, situação ou palavras diferentes são a mesma coisa em vários contexto do software. Ou seja, é esperado pelas pessoas usuárias certo nível de padrão na aplicação em desenvolvimento e quando não há essa abordagem de fora para dentro, (Barbosa, Simone & Silva Bruno, 2010), a tendência é o software não estar consistente para as pessoas que estarão usando no dia-a-dia.

E assim, nesse cenário, começamos a perceber que a relação entre empresas privadas e equipe multidisciplinar pode começar a ter problemas no nível das expectativas em cima do que irá ser desenvolvido, pois no final das contas, a métrica que importa para as empresas brasileiras é a relação de retorno sobre investimento. E a partir do momento em que a empresa analisa que o software

desenvolvido não está trazendo o retorno sobre o que foi investido para se ter o software, a pressão sobre a equipe multidisciplinar aumenta.

Seguindo nesse cenário em que empresas privadas brasileiras seguem investindo em suas transformações digitais, realizando o desenvolvimento de software de dentro para fora, e tendo como resultado dessa abordagem problemas de consistência na aplicação, surge então um problema a ser lidado, o qual tal trabalho monográfico buscará responder a seguinte questão:

- Como manter a consistência visual em um Sistema de Software Interativo, através da abordagem de Design System, para empresas privadas que investem em transformação digital?

É com essa questão norteadora que este trabalho irá explorar tal cenário e analisar quais boas práticas dos fundamentos de IHC, Design e Design System, podem ser utilizadas para contornar esses problemas.

Ao analisar a questão central desta pesquisa, é suposto que há uma forma de deixar softwares consistentes visualmente, fazendo com que as empresas, em momento de transformação digital, tenham softwares que garantam critérios de usabilidade, affordance e acessibilidade, e assim obtendo uma satisfação positiva dos seus usuários, que por consequência levará a empresa à um retorno sobre o investimento feito para se ter a aplicação.

Dada a hipótese desse trabalho monográfico, o objetivo geral que irá guiar este estudo é:

- Estruturar um Design System que centralize decisões de Design de Interfaces Gráficas, conectando times de Design e Engenharia em uma forma única de comunicação.

Já os objetivos específicos que irão guiar para o geral, são:

- Avaliar o cenário dos processos atuais das pessoas que atuam como designers e desenvolvedores, identificando quais ferramentas e tecnologias são utilizadas nos processos de construção de interfaces gráficas.

- Mapear os elementos mais utilizados para a criação de interfaces gráficas de Softwares;

- Caracterizar uma estratégia para a construção desses componentes de maneira que estejam conectados entre o que é usado pelo time de Design e o que é usado pelo time de Engenharia, os design tokens.

- Criar os componentes mais utilizados na construção de interfaces gráficas na ferramenta utilizada pelo time de Design e no Framework de tecnologia de Front-end utilizado pelo time de Engenharia.

Ao realizar tal pesquisa será possível observar uma evolução no assunto discutido de forma acadêmica, dado que não se é encontrado conteúdos brasileiros tratando tais cenários. Além disso, contribuirá para a disseminação de boas práticas de atuação no mercado de trabalho envolvendo desenvolvimento de softwares. Outro ator que terá impacto com esse trabalho são as empresas brasileiras que podem estar passando por esse momento de transformação, tendo assim um guia de como garantir que tenham usabilidade, affordance e acessibilidade de forma adequada para as pessoas usuárias e assim investindo melhor.

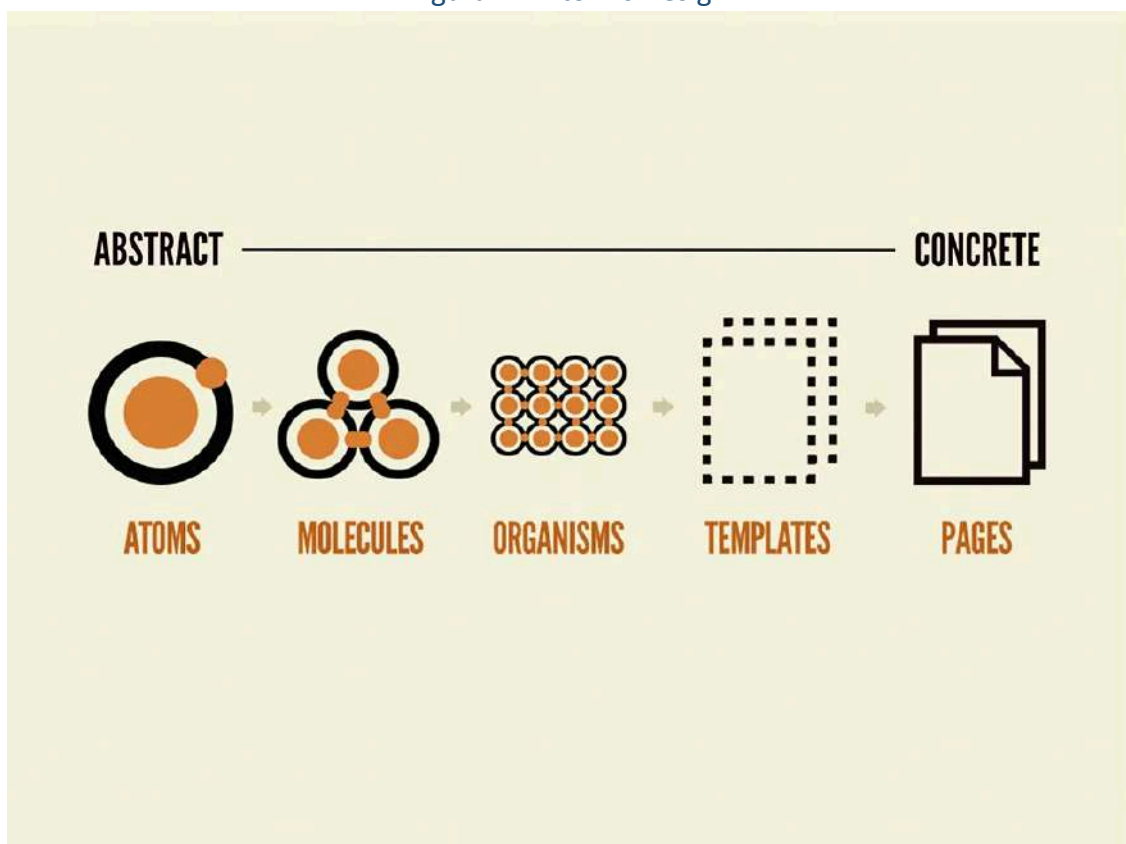
Na seção seguinte é apresentado como está o atual cenário relacionado a essa problemática e como outras pessoas e empresas lidam com tal questão, posteriormente será possível ler toda a base teórica que serviu de base para a estruturação e desenvolvimento desta pesquisa. Outras duas sessões que estão disponíveis neste trabalho são: Metodologia e Análise e Resultado, onde haverá a descrição de como foi utilizado o método Design Science Research, além da análise e os resultados colhidos após a aplicação de tal método. Por fim, ao final será possível encontrar as conclusões obtidas a partir desse estudo, o que foi validado e o que foi refutado, além de pontos de melhoria para o futuro.

2 ESTADO DA ARTE

Analisando o momento do Estado da Arte em relação a como manter consistência visual em Sistemas de Software Interativo, somos levados ao tópico com mais enfoque: a abordagem e estruturação de um Design System como produto que servirá de suporte na construção de tais softwares.

O termo “Design System” começou a ser explorado e discutido recentemente após a publicação do livro *Atomic Design* no ano de 2016 pelo autor Brad Frost, onde o mesmo aborda como estruturar Sistemas de Design através do método que tem como objetivo guiar a maneira de se estruturar elementos de interfaces gráficas, categorizando e dividindo tais elementos em grupos específicos, fazendo assim uma sistematização dos componentes como, botões, ícones, campos de texto e muitos outros. A Figura 1 foi extraída do próprio artigo proposto pelo autor e exemplifica como é feita essa categorização.

Figura 1 – Atomic Design

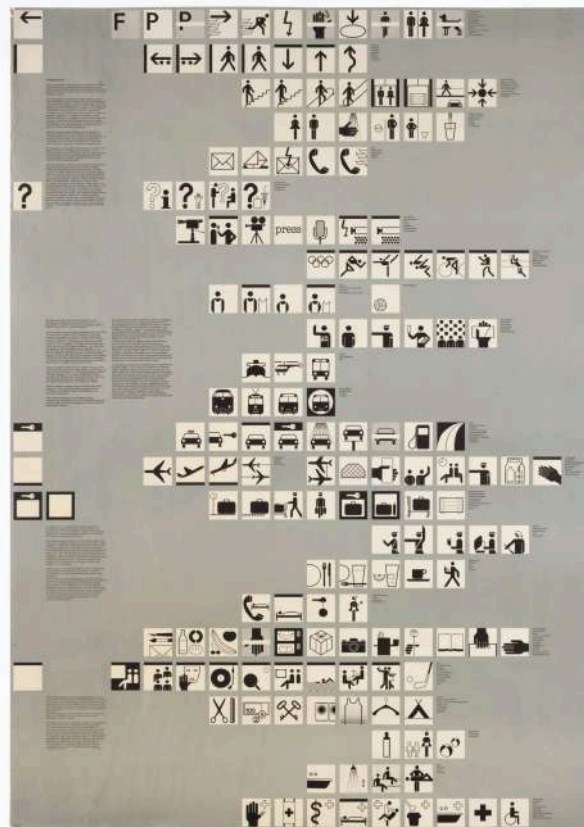


Fonte: [Frost](#),(2023)

Mas apesar do termo ter ganhado força depois de 2016 o conceito de Design System é percebido em outras áreas além da tecnologia, como podemos observar

no trabalho do Designer Otl Aicher para as Olimpíadas de Munique em 1972, onde o próprio criou toda uma documentação e guia de uso sobre a identidade visual, dos pictogramas e de todo o sistema de sinalização para os jogos olímpicos. Veja abaixo um exemplo de um dos artefatos produzidos por Aicher:

Figura 2 – Poster, München Olympics, 1972, Otl Aicher



Fonte: [NDGA](#), (2021)

Como podemos ver, se aproxima da base de pensamento exposta por Frost (2016) em seu livro, onde há elementos visuais categorizados seguindo o contexto de aplicação.

Seguindo então para o contexto de aplicação de Sistemas de Design no âmbito da tecnologia, podemos observar grandes empresas que utilizam dessa abordagem para garantir a consistência visual em suas aplicações de software, como o Google faz com o Material Design, o Design System open-source estruturado pela empresa. Esse material está disponível na internet e é uma grande documentação de como é estruturado o Design System do Google. Essa

documentação deu-se início em 2014 e desde lá a empresa tem contribuído com vários padrões e boas práticas referentes a elementos de interface gráfica.

No meio acadêmico podemos observar alguns trabalhos que vão de encontro com a abordagem de Design System para prover consistência visual em softwares. Como a pesquisa de Souza, Sergio (2023), intitulada “Design System Em Uma Holding de Ensino Superior”. Em resumo, a sua pesquisa analisa o papel do Design System na construção de uma identidade visual consistente e eficiente, explorando as metodologias de design e sua relação com o Design System. Construindo assim um estudo de caso sobre a implementação do Design System “Lyft”, onde o mesmo mostra através de processos de Design Thinking e métodos ágeis como estruturar um Design System, porém o ponto sobre a conexão das decisões de design entre Designers e Engenheiros não foi apresentada com detalhes de aplicação prática e ferramentas que poderiam apoiar nessa etapa do processo. Esse é um dos pontos que esta pesquisa irá abordar nas próximas seções, sendo assim um diferencial ainda não explorado no meio acadêmico.

Outro trabalho que podemos observar, publicado por Silva, Matheus (2019), intitulado como “Atingindo interfaces de usuário consistentes por composição e sistemas de tokens”, trás uma visão de engenharia para solucionar o problema de inconsistência em interfaces de usuário quanto ao código dessas interfaces, abordando a forma como a consistência é um pilar fundamental para a usabilidade dos usuários. O autor expõe as seguintes problemáticas nesse contexto, ao reconhecer as limitações do CSS tradicional na organização e gerenciamento de estilos, em projetos complexos. Além disso, a popularização da arquitetura de componentes trouxe novos desafios, como a proliferação de arquivos CSS e a dificuldade em manter a consistência visual.

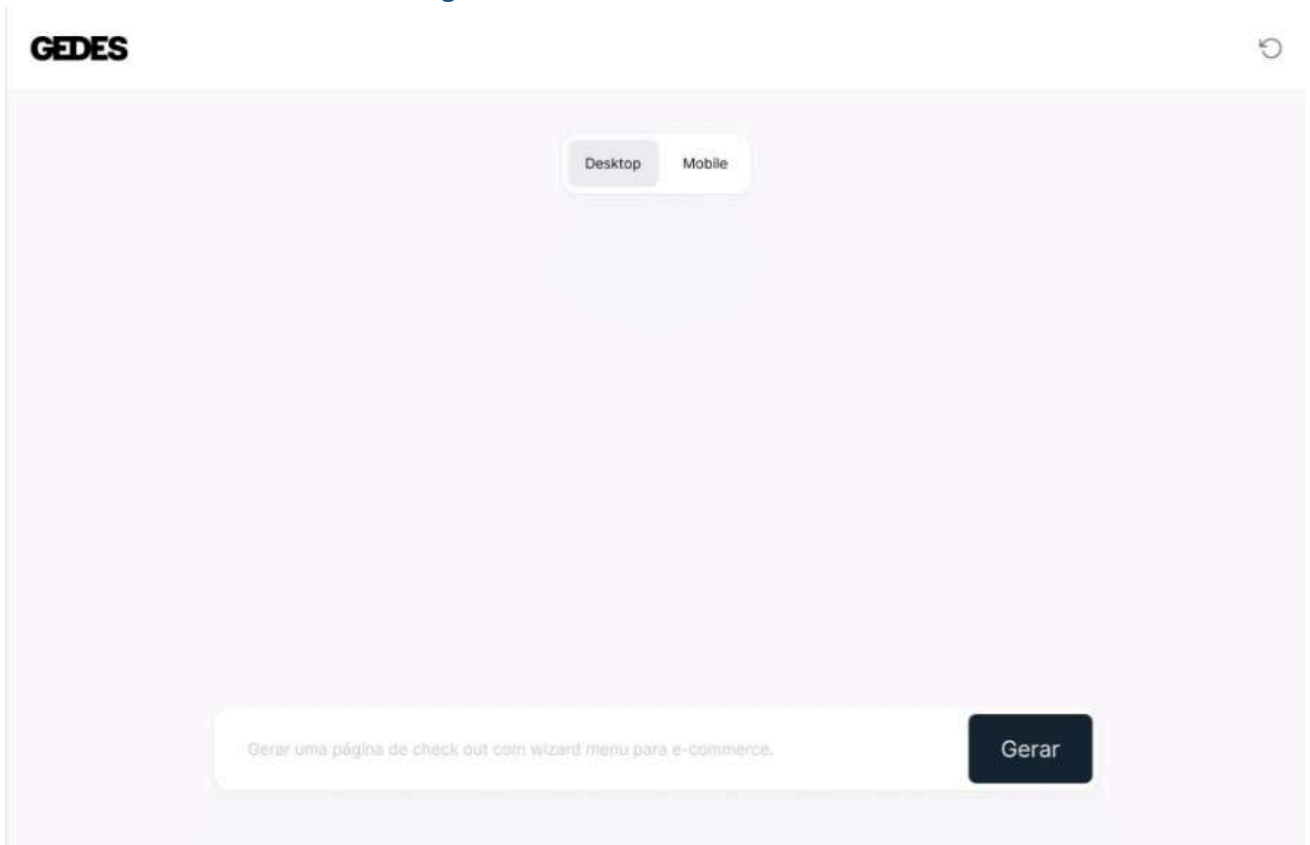
E com essa base ele mostra uma possível solução através de composição de código e sistematização de tokens, construindo duas bibliotecas de código, a “**styled-tokens**” e a “**adapt-ui**”, onde a primeira permite a criação de sistemas de tokens que encapsulam valores de estilo reutilizáveis e adaptáveis a mudanças no design e a segunda, oferece um conjunto de tokens pré-definidos e componentes estilizados que utilizam esses tokens, facilitando a construção de interfaces consistentes. Essa forma de solucionar tal problema vai de encontro com o que será

explorado nesse trabalho monográfico, porém além da forma de se solucionar esse problema do lado da engenharia, também será abordado aqui como garantir que essa consistência persista em Design também.

O último trabalho acadêmico observado nesta pesquisa foi o artigo publicado por Pereira, Marcus (2024) sobre “Design Systems: Um Estudo Exploratório Sobre a Documentação de Componentes para Interfaces de Usuário”. A pesquisa apresenta uma análise detalhada sobre a importância da documentação em design systems, com base na revisão de literatura, estudo de caso do Carbon Design System e pesquisa com usuários, definindo assim alguns critérios para se ter uma documentação coesa, conectada com código, com fácil navegação, acessível, fácil de manter e com histórico de mudanças. Os pontos apresentados no trabalho são de extrema importância no contexto de garantir a consistência de interfaces gráficas de usuário, porém essa é apenas uma parte de toda série de processos e fatores que serão mostradas neste trabalho em questão.

Podemos então ver que há vários estudos e evolução do assunto, apesar do pouco tempo de discussão sobre Sistemas de Design envolvendo desenvolvimento de softwares, o tema em questão é explorado constantemente e segue se adaptando conforme a tecnologia vai se desenvolvendo, podemos perceber isso no trabalho de Pós-Graduação do Filipe Nzongo, intitulado como “Sistema de Design Generativo, Como Meio para Aumentar a Eficiência Operacional, Reduzir Custos e Explorar Novos Horizontes de Interface.”, o autor explora o assunto indo além e trazendo o tema de Inteligência Artificial Generativa como forma de materializar os princípios de Design e decisões de como construir os elementos de uma interface gráfica. Tendo então como proposta final um Sistema de Design o “GEDES” que gera interfaces de usuário, usando como parâmetros os princípios de design definidos previamente.

Figura 3 – Interface do GEDES.



Fonte: Nzongo, (2022)

Em resumo, o Design System se mostra um método bastante explorado seja de maneira acadêmica, mas também por grandes empresas, como Google e IBM. Podendo então ser explorado e implementado para garantir a consistência visual em sistemas de software iterativo que estão sendo desenvolvidos por empresas em processo de transformação digital, sendo assim uma forma de garantir que o investimento dessas empresas no desenvolvimento de tais aplicações tenha um certo nível de segurança, dado que com o Design System garantindo a consistência visual através da plataforma desenvolvida, irá trazer segurança de uso as pessoas que estarão utilizando.

3 REFERENCIAL TEÓRICO E METODOLÓGICO

Para que seja possível manter a consistência visual em um Sistema de Software Interativo, se faz necessário o entendimento sobre qual tipo de Sistema está sendo trabalhado e como a abordagem de desenvolvimento impacta para chegar nessa consistência. Dentre as abordagens que envolvem esse contexto de problema, esse trabalho irá focar na estruturação de um Design System, comumente utilizado por empresas como Google e IBM, além de ser discutido inicialmente por profissionais, como Brad Frost e Alla Kholmatova.

3.1 SISTEMA DE SOFTWARE

Segundo Sommerville (2011) em seu livro Engenharia de Software, um sistema de software está além de ser apenas um programa de computador, é um composto do programa em si, da documentação vinculada e a forma como deve ser configurado para que o programa funcione corretamente. Além disso, o autor distingue alguns tipos de sistema de software conforme o contexto em que será desenvolvido, alguns deles são: sistemas embutidos, Sistemas de Sistemas, Produtos sob encomenda e o Sistema de Informação.

Em específico nesta pesquisa o foco está no tipo de Sistema de Informação que para o autor é um “sistema cujo objetivo principal é gerenciar e fornecer acesso a um banco de dados de informações”, em geral esse tipo de sistema é utilizado para questões de gerenciamento interno das empresas privadas. Sommerville trás também alguns critérios para que um sistema de software seja considerado profissional, os quais são:

- **Qualidade:** o software deve garantir dois pontos requeridos pelos usuários: a **funcionalidade** e o **desempenho**, sendo então confiável, fácil de manter e usar. Além disso, em um olhar de qualidade voltado para o programa em si é analisado como é o comportamento do software durante a execução, a sua estrutura, organização e a documentação.

- **Aceitabilidade:** o software deve ser aceitável para o tipo de usuário para o qual foi projetado, sendo compreensível, usável e compatível com outros sistemas utilizados por ele.
- **Profissionalismo:** se faz necessário que o desenvolvimento inclua técnicas para especificação, tenha um projeto estruturado e que o software esteja apto a evolução.

No livro "Engenharia de Software Moderna" de Marco Tulio Valente (2020) são abordadas questões que envolvem o conceito de sistema de software. Para o autor, o software é um produto complexo de engenharia que necessita de abordagens sistemáticas e que deve atender a requisitos específicos envolvendo um desenvolvimento em equipe. Devido a tal complexidade o autor busca enfatizar a diversidade de sistemas de software e como cada tipo deve ter seu método próprio para a abordagem de desenvolvimento, assim, baseando-se na classificação proposta por Meyer (2013), existem três tipos principais de software:

- **Sistemas A (Acute):** são sistemas críticos, onde falhas podem gerar consequências graves.
- **Sistemas B (Business):** são aqueles relacionados ao contexto corporativo, que envolvem questões financeiras, recursos humanos, logística, vendas, contabilidade, etc. São também sistemas web e sistemas de software básico.
- **Sistemas C (Casuais):** são sistemas em que a qualidade não é exigida, não sendo necessário o uso de scripts e programas de uso único.

Como é possível observar tanto Sommerville (2011) quanto Marco Tulio (2020) reforçam conceitos relacionados a sistemas de software na perspectiva da programação, porém há uma abordagem diferente para tal definição no livro "Interação Humano Computador" dos autores Barbosa e Silva (2010). Os sistemas computacionais interativos são "compostos por hardware, software e meios de comunicação desenvolvidos para interagirem com pessoas". Para eles o software é um artefato, fruto de processos de análise e síntese, sobre uma situação-problema levando a um conjunto específico de soluções. O artefato só alcança o seu sucesso caso os seus usuários consigam realizar as suas tarefas dentro do sistema linguístico no qual o artefato é codificado.

Essa abordagem de Barbosa e Silva (2010) nos mostra uma perspectiva diferente, com mais foco na situação-problema em que o software precisa ser desenvolvido, trazendo o sucesso do mesmo vinculado a forma como o usuário irá utilizar o sistema. Isso nos leva a questões que envolvem a interface de interação do sistema.

3.2 INTERAÇÃO, INTERFACE, USABILIDADE E AFFORDANCE.

Para analisarmos as questões que envolvem a interação do usuário com o sistema se faz necessário entender o que compõe e quais as boas práticas para garantir que durante o processo de uso o usuário não tenha problemas que o impeça de realizar o seu objetivo com o sistema proposto. Começando pela definição do que é interação e a interface.

3.2.1 INTERAÇÃO E INTERFACE

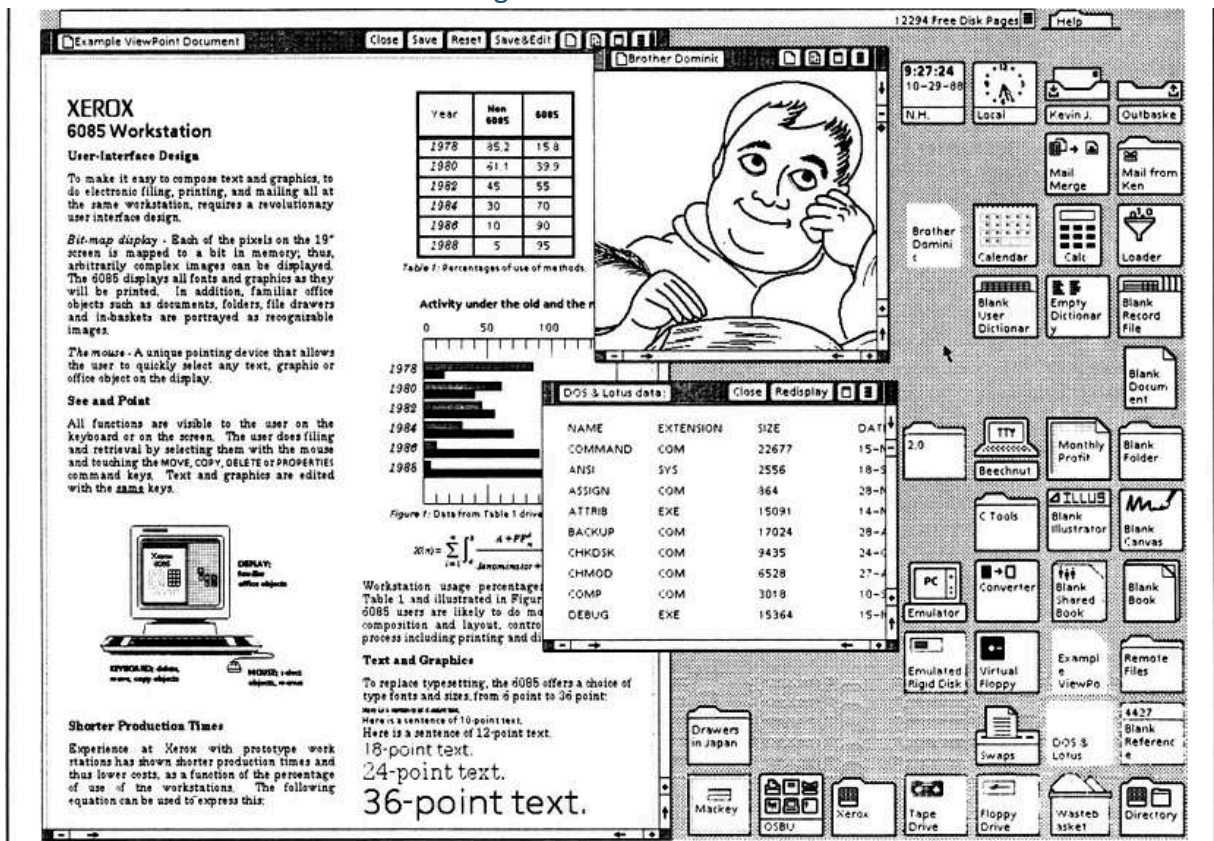
Segundo Norman (1986), a interação é resultado da intenção e planejamento das ações do usuário sobre a interface, onde o mesmo interpreta e percebe as respostas do sistema, avaliando se o seu objetivo foi alcançado. Já de Souza (2005), enfatiza que a interação usuário-sistema é como um processo de comunicação entre pessoas, mediada por sistemas computacionais. Dessa forma, a interação é o ponto chave em que o usuário irá dizer se o sistema em questão é útil ou não. O modo como essa interação é definida pela a equipe multidisciplinar envolvida no projeto se torna crucial, sendo necessário assim entender os requisitos necessários para o contexto da situação-problema e como será a interface onde irá acontecer tal interação.

A interface de um sistema interativo é a relação de todo o sistema com o qual o usuário realiza algum contato, seja motor, perceptível ou conceitual (Moran, 1981). Envolve desde o hardware para captar entradas de interação do usuário quanto dispositivos de saída como monitores que apresentam o que está sendo feito no software, e o próprio programa, quando por exemplo o usuário clica com o mouse em um botão para voltar à uma página anterior. É a interface que limita as ações do usuário com o sistema, ou seja quando se é projetado a interface, é necessário se

atentar a forma como ela será definida, pois pode ser que o caminho definido não seja o melhor para que o usuário realize os seus objetivos.

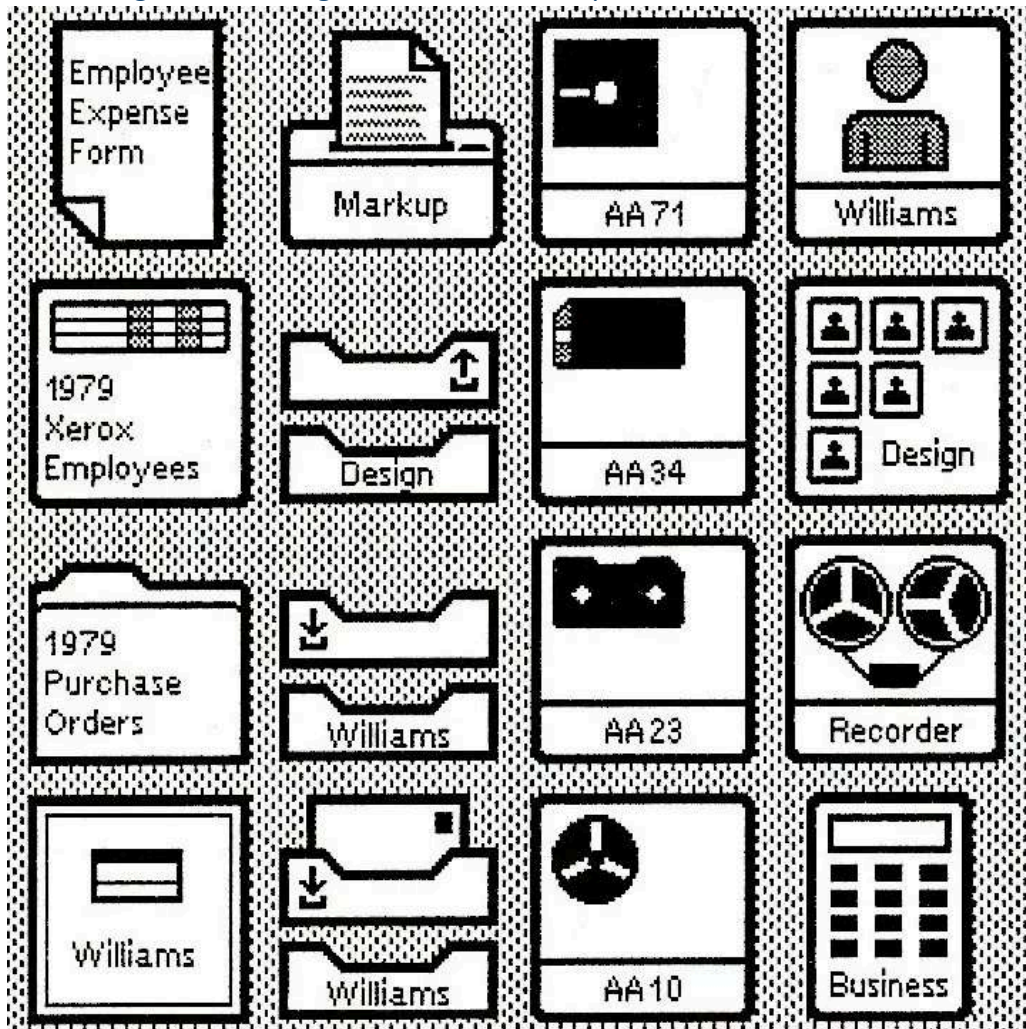
Um dos tipos de interface mais utilizados em sistemas de software corporativos são as interfaces gráficas do usuário, comumente chamadas de GUI (*Graphical User Interfaces*) com o modelo de interação WIMP (*Windows, Icons, Menu, and Pointing devices*) que faz uma metáfora ao ambiente de trabalho em escritórios, esse estilo de interface foi desenvolvido pelo departamento de pesquisa Xerox Palo Alto Research Center (PARC), a partir de 1980, para ser implementado no Sistema Operacional Xerox Star, tendo Alan Kay como um dos principais responsáveis por conceber os conceitos que são vistos até os dias de hoje. Veja abaixo algumas imagens da primeira interface gráfica do usuário:

Figura 4 – Xerox Star.



Fonte: Prosthetic knowledge, (2025)

Figura 5 – Iconografia confeccionada por David Canfield Smith .



Fonte: Inside Design, (2025)

3.2.2 USABILIDADE E AFFORDANCE

Esses dois conceitos são indispensáveis no contexto de interfaces gráficas do usuário, pois são responsáveis por guiar as boas práticas para conceber interfaces que possam levar os usuários ao objetivo desejado dentro do sistema. Quando falamos em Affordance em interfaces gráficas, estamos nos referindo ao fato de que os objetos possuem características que são capazes de evidenciar aos usuários as manipulações e operações possíveis de se realizar com ele (Norman, 1988). Ou seja, a forma como os elementos da interface gráfica são desenhados irá guiar o usuário e moldar a sua interação com o sistema, caso haja um cenário de falsa

affordance o usuário poderá se confundir com a forma de interação que o mesmo deseja realizar, entrando assim na questão da usabilidade do sistema.

A usabilidade de um sistema de software depende dos três conceitos apresentados até aqui, pois é no momento da interação com o sistema, através de uma interface gráfica desenvolvida com elementos com características conhecidas, affordance, que a pessoa que está utilizando o sistema dirá se o próprio possui uma usabilidade adequada. Para Norman (1993), a usabilidade está relacionada com a facilidade de aprendizado e uso da interface, bem como a satisfação do usuário em decorrência desse uso. E é com esse critério de satisfação que poderemos avaliar se uma interface tem uma boa usabilidade ou não, porém como foi evidenciado nesse estudo, em determinados cenários essa usabilidade pode não ter sido concebida da melhor forma, impactando toda essa cadeia relacionada a interação em interfaces gráficas, tendo como principal causa desse problema a falta de consistência visual gerando uma usabilidade ruim.

3.3 CONSISTÊNCIA E PADRONIZAÇÃO

Em 1994 Jacob Nielsen publicou o seu primeiro artigo intitulado “10 Usability Heuristics for User Interface Design”, onde definiu com base em pesquisas, 10 princípios gerais sobre usabilidade em softwares. Dentre estes 10 princípios há a “Consistência e Padrões”, baseada na Lei de Jacob sobre experiência do usuário na internet, onde afirma que os usuários passam mais tempo em outros sites do que o site feito por uma empresa específica, ou seja, os usuários estão mais acostumados com outros sites do que um em específico, o que faz as expectativas dos usuários serem baseadas por outros sites. E não é somente Nielsen que traz evidências sobre essa relação das expectativas dos usuários, Norman (1988) e Tognazzini (2003) também acreditam que a consistência mais importante é com as expectativas dos usuários.

Dessa forma, tais especialistas recomendam que as ações, os resultados das ações, o layout dos diálogos e as visualizações de informação devam ser padronizados em todo o sistema, analisando quais elementos possuem ações diferentes, fazendo com que os mesmos tenham um visual diferente, é o exemplo dado por Barbosa, Simone & Silva Bruno (2010) em que “um elemento de interface utilizado para selecionar uma opção deve ser distinto de um elemento utilizado para

disparar uma ação do sistema”. Mas é nesse ponto que esse estudo monográfico adentra, pois devido a abordagem de desenvolvimento olhando “de dentro para fora” (Barbosa, Simone & Silva Bruno, 2010), não tendo o usuário como ator participante do processo, pode acabar surgindo inconsistências visuais dentro do sistema, podendo haver assim elementos que possuem a mesma característica, porém com visuais distintos, fazendo com que a expectativa da pessoa que irá utilizar o sistema não seja atingida, e como consequência não sendo confiável, Cooper (1999).

O instituto Nielsen Norman Group aborda esse tema sobre manter a consistência em sistemas de informação no artigo publicado por Rachel Krause, (2021) intitulado de “*Maintain Consistency and Adhere to Standards*” (Manter a consistência e aderir aos padrões). A autora menciona sobre como manter uma consistência contínua, ao olhar internamente para um software ou externamente para outros softwares do mesmo setor, além disso, discute quais são as camadas da consistência e os impactos que ela pode trazer para o ecossistema do software.

3.3.1 CONSISTÊNCIA INTERNA E EXTERNA

A consistência interna está relacionada aos padrões internos de um software ou produto ou família de produtos, como pacote de aplicações para escritório do Google Workspace, Google Docs, Google Sheets e Google Presentation. conseguimos visualizar essa consistência interna quando analisamos a mesma ordenação de menu entre tais produtos, sempre disponibilizando “Arquivo, Editar, Ver” e demais opções.

Figura 6 – Menu Superior Google Docs.



Fonte: Google Docs, (2025)

Figura 7 – Menu Superior Google Presentation.



Fonte: Google Docs, (2025)

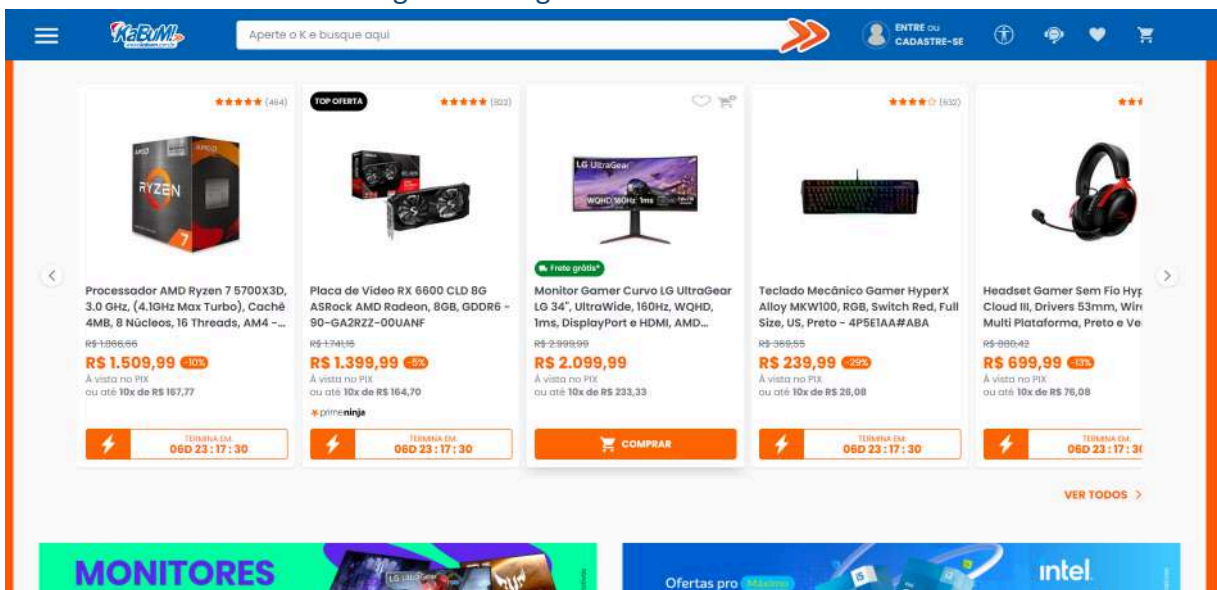
Figura 8 – Menu Superior Google Sheets.



Fonte: Google Docs, (2025)

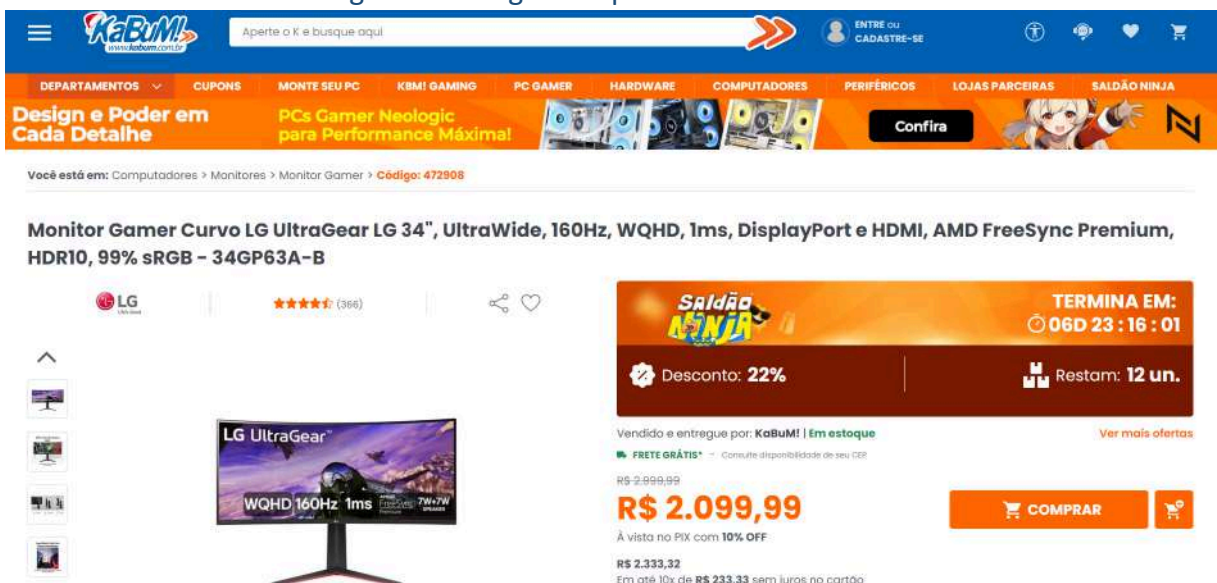
Já em sistemas web como o ecommerce da Kabum é mais simples visualizarmos a consistência nos layouts das páginas e cores aplicadas a elementos da interface gráfica como botões, onde sempre temos o botão principal utilizando a mesma cor em vários pontos da aplicação.

Figura 9 – Página inicial site Kabum!.



Fonte: <https://www.kabum.com.br/>, (2025)

Figura 10 – Página do produto site Kabum!.



Fonte: <https://www.kabum.com.br/>, (2025)

Nas imagens acima podemos observar esse comportamento relacionado ao botão “Comprar”, tanto na tela inicial e na tela do produto o botão possui as mesmas características, tornando assim para o usuário o fácil entendimento que independente da página onde ele esteja, será possível realizar a compra através daquele botão. Caso houvesse uma diferença entre o mesmo botão para cada página o usuário teria que para e compreender onde estaria essa ação.

Já a consistência externa se refere aos padrões definidos dentro da categoria do software ou da web em geral. Rachel (2021) trás um exemplo de padrão definido na web, o qual é o “link da página inicial”, onde em todos os sites são referenciados pela logo do site, tornando um acesso rápido para à tela inicial. Outro ponto abordado por Rachel são os níveis de consistência necessários para se observar e assim conseguir manter entre todo o software.

3.3.2 CAMADAS DE CONSISTÊNCIA

Rachel (2021) apresenta quatro níveis de consistência para se analisar e assim atingir o objetivo de conseguir manter o sistema de informação consistente, os quais são: visual, layout da página e botão, dados inseridos pelo usuário e conteúdo.

- **Visual:** na camada visual da consistência, são avaliadas através da iconografia, imagens, símbolos, cores e formatos dos elementos da interface gráfica. Manter o mesmo padrão desses elementos em todo o software fará com que os usuários não precisem se questionar toda vez que estiverem realizando tarefas diferentes dentro da mesma aplicação.
- **Layout da página e botão:** aqui a base é manter a mesma estrutura mesmo havendo diferentes páginas em seu sistema, trazendo a consistência nos elementos da interface, como o botão, tendo uma decisão definida pelo time de Design que o botão principal possua a cor laranja e esteja sempre posicionado à direita, fará com que o usuário aprenda apenas uma vez essa disposição e

espere que essa interação seja a mesma independente da tarefa em que esteja realizando.

- **Dados inseridos pelo usuário:** na web em geral alguns formatos de preenchimento de dados já possuem um padrão estabelecido, como a forma de preencher dados relacionados a Datas, a maior parte das aplicações utilizam um componente de seletor de data ao invés de um campo de texto livre. Utilizar esse padrão já estabelecido, fará com que o usuário não gaste mais tempo entendendo que ele precisa digitar uma data ao invés de selecioná-la.
- **Conteúdo:** o nível de consistência para conteúdos está bem relacionado com a forma em que é escrito na interface, se durante uma experiência o usuário se depara com textos com diferentes tons de voz, o mesmo começará a achar estranho, trazendo uma desconfiança e desentendimento se continua na mesma plataforma.

Ao ter como guia estes quatro pontos de consistência para ser mantido entre todo o sistema, poderá observar os impactos em duas instâncias, quanto às expectativas dos usuários e a sua curva de aprendizagem. Pois, quando o usuário se depara com um software ele já trás consigo uma bagagem de outras experiências em softwares semelhantes, esperando então ver certa semelhança em alguns pontos da experiência, manter essa consistência externa, fará com que as expectativas dos usuários sejam atendidas. Além disso, os usuários não precisarão gastar mais tempo para entender como funciona tal parte da experiência, diminuindo assim a curva de aprendizagem ao interagir com a aplicação.

Porém como estruturar essa consistência é um fator importante, para Rachel (2021) a consistência interna do software pode ser potencializada com a construção de um Design System, o qual trará benefícios para os usuários finais, mas também para as equipes de Design e Engenharia na construção das interfaces gráficas. E é esse ponto que vamos observar no próximo tópico, como é a anatomia de um Design System e seus benefícios.

3.4 DESIGN SYSTEM

Antes de evidenciar do que é composto um Design System ou Sistema de Design, precisamos analisar como o mesmo é definido. Esse ponto de definição ainda é amplamente discutido pelas pessoas que trabalham com desenvolvimento de softwares, existem muitas definições acerca do tema, mas algumas vem ganhando relevância, tais definições trazidas por especialistas em executar projetos de estruturação de um Design System. Começando por Frost (2021), para ele Design System é “a história oficial de como uma organização projeta e constroi interfaces digitais”, tal definição traz um foco mais no impacto que um Design System deve trazer para a empresa que esteja construindo, pois segundo ele, “a única coisa que realmente importa no final do dia é a experiência real do usuário com a qual os seres humanos interagem”, Frost (2023).

Uma segunda definição que encontramos vem da Alla Kholmatova (2017), para ela um Design System é um conjunto de *Padrões Conectados e Práticas Compartilhadas*, organizados de forma coerente para atender aos propósitos de um produto digital, nessa definição a autora complementa onde o DS deve impactar e define alguns artefatos que precisam estar contidos nessa estrutura de *Padrões e Práticas Compartilhadas*. Em seu livro intitulado: *Design Systems: A Practical Guide*, ela aborda sobre esses *Padrões Conectados* sendo os elementos reutilizáveis da interface, diferenciando entre padrões funcionais (comportamentos) e perceptuais (marca e estética), quanto às *Práticas Compartilhadas* se diz à conexão entre Projeto e Engenharia, tornando esses padrões em códigos reutilizáveis e que são compartilhados entre todos que trabalham na equipe que estão desenvolvendo o sistema.

Outra definição que podemos observar é a do Nathan Curtis. Em seu artigo publicado em 2017, intitulado de “Defining Design Systems”, o autor aborda três aspectos que definem um DS, primeiro deles é que um DS precisa conter um “UI kit de peças reutilizáveis e interconectadas”, o UI kit é a representação de todas as peças de elementos de Interface de Usuário que são mais utilizadas para compor as páginas da interface gráfica de um sistema, o fato de estarem interconectadas vai de encontro com que os autores anteriores definiram também, esse artefato precisa estar conectado com a Engenharia através de códigos reutilizáveis. O segundo ponto evidenciado por Nathan é que o DS precisa ser “um conjunto de produtos

coesos e interconectados”, ou seja, o Design System precisa ser tratado como um produto que atenderá seus clientes, clientes estes sendo as pessoas designers e engenheiras que utilizam o sistema para construir as interfaces gráficas e os usuários finais que irão utilizar tais interfaces, seguindo essa ideia, a equipe responsável pela construção do Design System conseguirá entender as necessidades e pensar de forma estratégica para que o projeto traga impacto para a empresa em questão. Já o terceiro ponto que define um DS segundo o autor é: “um DS é uma comunidade de colaboradores interconectados”, dessa forma, podemos observar a importância que o autor atribui ao fato da construção de um DS envolver várias pessoas, chegando à um nível de uma comunidade, pois muitas decisões precisam ser refletidas, decisões estas que influenciam o visual da marca da empresa e como isso será comunicado nas interfaces de usuário, sendo assim necessário a colaboração durante a execução do projeto.

Por fim, como uma referência do nosso mercado brasileiro, Filipe Nzongo define Design System como “uma combinação de instrumentos que podem produzir diferentes sonoridades, desde Jazz à música eletrônica. Do mesmo jeito um sistema de design consistente pode produzir desde uma *Landing Page* à um *Dashboard* complexo.”. A definição trazida pelo autor trás mais o foco do que é possível fazer e a diversidade de tipos de interface quando se possui um Design System bem estruturado. Dessa forma, mesmo que todas essas definições enfatizam a importância de um sistema de design em diferentes pontos, conseguimos perceber que quando falamos de um Design System, estamos nos referindo à um projeto de um produto que tem o foco exclusivo em construir interfaces de usuário de forma reutilizável e que as definições que caracterizam os elementos da interface sejam compartilhadas entre as pessoas que consomem tal produto.

3.4.1 O ECOSISTEMA DE UM DESIGN SYSTEM

Agora que temos uma definição do que é um Design System, se faz necessário entender do que é composto um Design System e como ele é estruturado. Tendo como base o artigo publicado por Frost em 2023 intitulado de “The Design System Ecosystem” podemos observar que há todo um ecossistema na relação entre o DS e o software que o mesmo serve para construir as interfaces, o

autor argumenta que há várias camadas nessa relação e que tais camadas moldam e definem como o Sistema de Design é construído, veja a imagem abaixo gerada pelo próprio autor:

Figura 11 – Relação entre as camadas de um Design System.



Fonte: Frost, (2023)

Como podemos ver, as camadas *Core Design System*, *Technology-specific implementation*, *Recipes* e *Smart components* alimentam a camada do produto/sistema e por outro lado a camada de produto devolve necessidades e melhorias que precisam ser feitas para cada camada. Seguindo essa abordagem do autor podemos entender melhor o que há de artefato em cada uma dessas camadas, começando pelo núcleo de um Sistema de Design.

1. CORE DESIGN SYSTEM (NÚCLEO DO SISTEMA DE DESIGN)

Para o autor, com base nas consultorias fornecidas para várias empresas, o núcleo de um Sistema de Design é composto por padrões, princípios, convenções e componentes, todos esses pontos se referindo às interfaces gráficas do usuário. Tais fatores possuem o objetivo de ajudar a empresa privada que está investindo em desenvolvimento de software à: “contar a história oficial de como o DS projeta e

constroi interfaces de usuário e por consequência fornecer os materiais de construção para fazer tais interfaces”. Ou seja, é no núcleo do Design System que moram as decisões de design que compõem os elementos de UI mais comuns para a construção do Sistema de Software em questão. E para se ter esse núcleo o autor acrescenta quais são os ativos/artefatos que darão vida a essa camada.

a. DESIGN TOKENS (TOKENS DE DESIGN)

O conceito de Design Tokens surgiu na empresa estadunidense Salesforce, responsável por desenvolver softwares por demanda, o conceito definido em 2014 a partir do problema sobre como manter consistente as decisões de design (cores, *border radius*, espaçamentos) com a necessidade de construir para vários tipos de plataformas diferentes (IOS, Android, Web) mantendo a eficiência em momentos de alteração nas decisões de design (Rohde, 2014). Com esse problema em mente eles chegaram à seguinte definição: manter as decisões de design em um único lugar, usando um sistema que leve essas decisões para todas as demais plataformas, sendo basicamente uma série de arquivos JSON contendo um par nome-valor, sendo de diferentes categorias, como cores de texto, fundo de botões, bordas, tamanho de fontes, espaçamento e outras micro decisões visuais. ([Sönke Rohde, 2014](#))

O objetivo então é usar os tokens no lugar do valor bruto, para que assim consigamos atingir um nível agnóstico do DS desenvolvido, pois assim não criamos nenhum vínculo à uma plataforma específica, linguagem de código ou framework. Dessa forma é possível ter uma **única fonte da verdade** com todas essas decisões em um repositório de código compartilhado, tendo então essas decisões em pares nome-valor em arquivos JSON. A partir desses arquivos podemos transformar esses JSONs brutos para o formato específico da linguagem escolhida para construir o Front-End do Sistema.

Quanto a esse ativo, Frost recomenda que essa seja uma camada interna do núcleo separada dos componentes de UI “para desbloquear a tematização das interfaces, criar uma separação de preocupações entre a linguagem da marca e os componentes da IU e criar linguagens de marca de versão independentes dos

componentes da IU.”. Sendo assim, ele sugere que essa camada seja organizada em uma biblioteca na ferramenta de Design em que o DS esteja sendo construído, isso para o time de Design, já para a engenharia, os tokens de design estarão em um repositório, seguindo o conceito estabelecido pela salesforce, e por último que essas decisões de design estejam empacotadas e publicadas em um registro de software para facilitar o compartilhamento entre os desenvolvedores.

b. ÍCONES

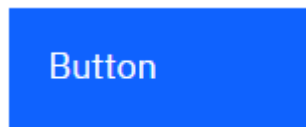
Os ícones possuem um objetivo fundamental na composição de interfaces gráficas do utilizador, pois são utilizados para auxiliar e complementar o entendimento da pessoa usuária sobre o sistema. Como boas práticas de construção de interface, a iconografia está presente em botões de ação, menus e elementos de formulário. Dessa forma, fazem parte do núcleo central do DS, para Frost uma boa abordagem para ganhar escalabilidade com os ícones é gerenciar/versionar ícones como uma camada própria, da mesma forma feita com os Design Tokens, pois assim empresas que possuem sistema em várias plataformas, web e mobile, conseguem gerenciar de uma forma mais facilitada.

c. COMPONENTES DE UI

Os componentes de UI são um conjunto de elementos que compõem a interface gráfica do sistema de software, sendo desenvolvido pela empresa privada. Esse conjunto é composto pelos elementos que mais aparecem em toda a estrutura do sistema de software interativo, esse conjunto só é definido pela equipe que está estruturando o DS, após uma análise de todas as telas que estão no sistema. Segue abaixo alguns exemplos de componentes de UI que podem compor esse conjunto:

i. Botão

Figura 12 – Botão do Carbon Design System, DS da IBM

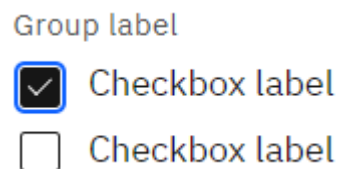


Fonte: Carbon Design System, (2025)

O botão é o responsável por ser o gatilho das ações principais a serem executadas pelos usuários dentro do software. O exemplo acima mostra como a IBM construiu o botão para o seu contexto de aplicação.

ii. Caixa de seleção (*Checkbox*)

Figura 13 – Checkbox do Carbon Design System, DS da IBM



Fonte: Carbon Design System, (2025)

O *checkbox* serve para apresentar opções a serem escolhidas pela pessoa usuária em contextos de preenchimento de informações como em formulários. O exemplo acima segue sendo o Carbon Design System e mostra como os elementos possuem entre si certa consistência.

iii. Botão de alternância (*Toggle*)

Figura 14 – Toggle do Carbon Design System, DS da IBM



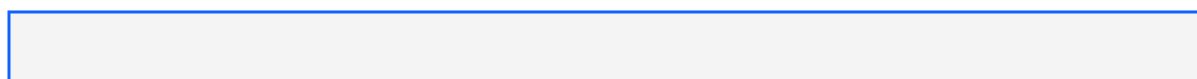
Fonte: Carbon Design System, (2025)

O toggle é utilizado em contextos que o usuário tem o controle de apontar o estado de alguma funcionalidade do sistema, entre habilitada ou desabilitada. Na figura 14 podemos ver que a IBM utiliza elementos de cores para mostrar essa alternância e assim deixar mais claro ao usuário o que ele está realizando.

iv. Campo de texto (*Textfield*)

Figura 15 – Textfield do Carbon Design System, DS da IBM

Text input label

A diagram of a text input field. It consists of a horizontal rectangle with a thin blue border and a light gray fill. The rectangle is positioned between the text 'Text input label' above and 'Optional help text' below.

Optional help text

Fonte: Carbon Design System, (2025)

O campo de texto é um elemento utilizado para dar a possibilidade do usuário inserir informações para o sistema. Da mesma forma que os elementos anteriores, a IBM mantém decisões de design em comum, como a cor azul dando destaque no que está ativo ou selecionado, tipo de fonte tipográfica utilizada e espaçamento dos elementos. Esses fatores são compartilhados entre todos os componentes de UI presentes no Carbon Design System.

Para Frost, esse conjunto de componentes de UI devem estar disponíveis para designers, que estão construindo os protótipos de interfaces gráficas do sistema de software, na ferramenta de escolha da equipe. A ferramenta mais utilizada atualmente no mercado é o Figma, dentro dela é possível construir e gerenciar essa lista de componentes e disponibilizar versões para as equipes que irão utilizar o Design System, essa lista é comumente chamada de biblioteca de componentes. Já para engenheiros de software, esse conjunto de componentes é disponibilizado através de um repositório de código exclusivo da biblioteca, Frost enfatiza que “a verdadeira fonte da verdade para um sistema de design é uma biblioteca de componentes codificados que criam produtos digitais reais. Afinal, apesar de toda essa confusão de ecossistema, a única coisa que realmente importa no final do dia é a experiência real do usuário com a qual os seres humanos

interagem.”. O formato recomendado pelo autor para a disponibilização dessa biblioteca é o de Web Components, pois são um padrão da própria plataforma web, caso o Sistema de Software esteja sendo construído em alguma tecnologia específica, poderemos ver mais a frente que através dos design tokens, conseguimos converter os componentes e suas estruturas para essas tecnologias específicas.

Os últimos ativos que fazem parte dessa camada de componentes de UI são as documentações dos componentes para Designers e para Engenheiros de Software. Para cada ator tem se uma abordagem diferente, para os engenheiros o Frost recomenda utilizar uma ferramenta chamada Storybook, o qual dá a possibilidade de documentar, revisar e testar os componentes, com os Designers a estratégia é ter um site de referência servindo como uma vitrine com documentação, suporte e todos os elementos que compõem o DS, uma ferramenta recomendada é o Zeroheight que possibilita todas essas questões mencionadas.

2. TECHNOLOGY-SPECIFIC IMPLEMENTATION (IMPLEMENTAÇÃO ESPECÍFICA DE TECNOLOGIA)

Como o núcleo central do DS é composto por elementos da interface codificados em Web Components, pode vir a ser necessário, com base no contexto de desenvolvimento do Sistema de Software, a tradução desses componentes para outro tipo de tecnologia de construção no Front-end do software. Para o Frost, essa camada de tradução/envolvimento pode ajudar desenvolvedores de aplicativos que trabalham em pilhas de tecnologia específica, como React, Angular, Vue, cenários desenvolvimento nativo como Android e iOS e outros como totens de atendimento e affins, a manejar facilmente os elementos do DS.

O auxílio funciona da seguinte maneira: os Web Components do DS são envolvidos em uma sintaxe específica, nesse exemplo de um framework React, então podemos chamar uma instância do botão Web Component através do código `<ds-button variant="primary">`, com a camada de tradução/envolvimento, podemos envolver o botão no framework React e

expor aos desenvolvedores como `<DsButton variant="primary">`. Facilitando assim para os engenheiros que estão acostumados com certa nomenclatura.

Sendo assim, essa camada ficaria responsável por realizar essa transformação, podendo ter subcamadas para frameworks, desenvolvimento nativo e outras tecnologias. A lógica aqui segundo o autor é que para cada subcamada ter o seu próprio repositório de tradução/envolvimento (chamado de wrapper), pacote de compartilhamento e documentação independente dos mesmo ativos que moram no núcleo central do DS.

3. CAMADA DE “RECEITA” (RECIPES) E COMPONENTES INTELIGENTES (SMART COMPONENTS)

Estas duas camadas são opcionais para uma estrutura de um Design System, pois tais camadas cuidam de contextos mais específicos e relacionados ao sistema de software sendo desenvolvido. Quando Frost menciona a camada de *recipes*, é a camada que serve para dar vazão a necessidades específicas de uma equipe específica que está desenvolvendo o software, no contexto atual de desenvolvimento é comum haver várias equipes cuidando de partes diferentes do sistema, que por sua vez pode ter necessidades de elementos de UI específicos que não estão no núcleo central do DS. Dessa forma, surge tal camada para que o próprio time possa construir tais componentes usando a base compartilhada do núcleo do sistema de design.

Já a camada de componentes inteligentes, cuida da conexão do front-end com back-end do sistema, atuando no *back-of-the-front-end*. Tais elementos contém a lógica e funcionalidade de negócios comuns como, envio de formulários, componentes de pagamento, elementos de análise de comportamento e muito mais.

4. CAMADA DE PRODUTO

Esta é a camada onde o Sistema de Design se torna realidade, pois traz toda a infraestrutura mencionada anteriormente para ser utilizada na construção do sistema de software interativo. Na visão do autor, é nesse nível “que o sucesso de um sistema de design pode ser verdadeiramente medido”, pois toda a lógica construída previamente irá provar os seus benefícios na camada de produto, trazendo benefícios em reutilização de código para pessoas engenheiras e economia de tempo para designers na construção de protótipos de interfaces.

Além disso, o usuário final irá se beneficiar de um sistema mais consistente e com questões de acessibilidade validadas desde o núcleo do design system, através das decisões de design que compõem os Design Tokens e por final, a empresa privada verá que tais benefícios terão o impacto financeiro vindo da economia de tempo dos profissionais, tendo os mesmo mais tempo para a realização de tarefas mais valiosas e também o aumento da satisfação dos usuários ao usarem o sistema de software interativo.

3.4.2 CONSIDERAÇÕES DO ECOSISTEMA DE DESIGN

Toda essa visão em camadas abordada por Frost nos trás uma dimensão do quão grande pode ser um projeto de estruturação de um Sistema de Design, sendo algo complexo de executar. Porém o próprio autor reitera que quase todas as camadas são opcionais e vão depender das necessidades da empresa privada em questão, e recomenda começar o Design System de forma simples, utilizando como base a Lei de Gall (1975): “Um sistema complexo desenhado do zero nunca funciona e não pode ser remendado para fazer funcionar”, ou seja, as camadas necessárias para a arquitetura de um DS vão depender da complexidade do cenário, podendo então começar de forma simples e conforme o sistema for evoluindo, fazer incrementos das demais camadas citadas.

Sendo assim, começamos a entrar no assunto relacionado aos métodos para chegarmos na definição da arquitetura do ecossistema de um Design System. No próximo tópico serão abordadas as referências que guiaram a abordagem metodológica para avaliar se a hipótese em aberto de como manter a consistência visual em um sistema de software interativo pode ser validada ou não.

4 METODOLOGIA

Como forma de obtermos um conhecimento de qualidade, que seja útil para profissionais em empresas e reconhecido pela academia, o uso da Design Science como pensamento epistemológico se encaixa perfeitamente para o direcionamento do trabalho proposto nesta pesquisa. Sendo então uma forma de se estudar o artificial, visando a criação de tal artefato de forma que seja relevante para as empresas e mantendo o rigor que a academia exige. É com essa base que vários autores de áreas como sistema de informação, engenharia, arquitetura e artes propuseram métodos para se executar tal pensamento a fim de se definir o meio de resolução de problemas criando artefatos artificiais. (EEKELS; ROOZENBURG, 1991; NUNAMAKER et al., 1991) (WALLS et al., 1992).

Esse método é conhecido como Design Science Research que guiou o atual trabalho, em específico os métodos apresentados por VAN AKEN *et al.* (2012) e PEFFERS *et al.* (2007), a junção das ideias apresentadas por esses autores teve como objetivo garantir que seja relevante para o contexto das organizações e cumprir com o rigor acadêmico. Dessa forma, nos próximos tópicos serão apresentados as fases do método e as metodologias utilizadas em cada fase, definindo uma maneira de garantir a consistência visual de sistemas de software interativo através da abordagem de Design System, para empresas privadas que investem em transformação digital.

4.1 DEFINIÇÃO DO PROBLEMA

Para VAN AKEN *et al.* (2012) e PEFFERS *et al.* (2007), essa fase é responsável pelo entendimento do contexto do problema, evidenciando a relevância da pesquisa. Dado o contexto de desenvolvimento de software de forma ágil e em times multidisciplinares, se faz necessário realizar entrevistas com as pessoas que estão no processo de construção de interfaces gráficas, como designers e pessoas engenheiras de software, através do tipo de pesquisa qualitativa. Dado que para Triviños (1987) a pesquisa qualitativa tem como base a percepção do fenômeno dentro do seu contexto, compreendendo o significado, o objetivo de tais entrevistas

está relacionado ao levantamento de dores em relação a tal processo, assim, com os dados coletados servirão para a próxima fase.

Além das entrevistas com o foco em entender como é feito o processo de criação de interfaces gráficas, outro processo importante é o mapeamento de pessoas interessadas e impactadas pela proposta de solução para o problema em questão, comumente chamado de Mapa de Stakeholders. Esse processo se torna essencial na compreensão do impacto das pessoas envolvidas na proposta da solução e como elas serão impactadas pela mesma. (FREEMAN, 2010).

Para finalizar essa fase de definição do problema e auxiliar no entendimento do que precisa ser melhorado, é feita uma auditoria das interfaces gráficas construídas na fase do projeto, por designers, e as interfaces construídas na fase final pelos engenheiros, disponível para as pessoas usuárias do sistema. Esse processo tem como objetivo mapear onde estão as inconsistências visuais nas interfaces gráficas e compreender quais elementos são os mais afetados, além de compreender os tipos de inconsistências. Todos esses processos servirão de insumo para a próxima fase do método, a qual será explicitada no próximo tópico.

4.2 ANÁLISE E DIAGNÓSTICO

Para Van Aken *et al.* (2012) esta fase é o momento de entender ao máximo o contexto e o ambiente em que ocorre o problema, buscando o entendimento do que pode estar causando tal problemática. Com o levantamento de informações da fase anterior relacionado às entrevistas, o mapa de stakeholders e a análise das interfaces gráficas é possível realizar a análise da situação atual da empresa, e condensar essas informações utilizando o método de Análise Temática. Segundo Maria Rosala (2022), esse método consiste em decompor e organizar dados marcando observações e citações individuais com códigos apropriados, e assim facilitando a descoberta de temas significativos, como dores do processo, pessoas relevantes e quais elementos apresentam mais problemas de consistência visual. Tendo essa visão da situação atual da empresa, é possível seguir para a próxima etapa definindo o que pode se esperar dos resultados e impacto da solução no contexto.

4.3 DEFINIÇÃO DOS RESULTADOS ESPERADOS

Esse é o momento que para Peffers *et al.* (2007) se define por meio de metas quantitativas ou qualitativas para o problema em resolução. É uma oportunidade para dar visibilidade às pessoas não técnicas dos próximos passos na construção da proposta para solução do problema e quais objetivos irão guiar. Para isso, é possível utilizar a construção de um Roadmap, essa técnica consiste em uma lista de trabalhos futuros com o objetivo de alinhar, priorizar e deixar evidente tais trabalhos. (Sarah Gibbons, 2020). Conectando assim ao Roadmap as principais entregas que impactam as metas alinhadas nessa fase. Com isso definido e o contexto do problema analisado e compreendido, chegamos a fase de construção do artefato de solução.

4.4 PROJETO E DESENVOLVIMENTO

Conforme foi apresentado no método DSR exposto por Peffers *et al.* (2007) e Van Aken, *et al.* (2012) o objetivo para essa fase é a confecção do artefato para resolver o problema estudado, definindo suas funcionalidades desejadas e a arquitetura, tendo como base os conhecimentos teóricos existentes. Dessa forma, para conseguirmos uma maneira de manter a consistência visual em um sistema de software interativo, vamos usar como base a estrutura do ecossistema de um Design System, exposta por Frost (2023).

Em um primeiro momento para termos a camada central (*Core*) do sistema, podemos utilizar do processo de pesquisa de mercado, através de um benchmarking com foco no visual dos elementos que compõem um Design System, apesar de um benchmarking ser utilizado para comparar métricas com um padrão significativo (Kate Moran, 2020), nesse caso será uma comparação entre soluções disponíveis no mercado, como o Carbon Design System da IBM, Material 3 do Google, Human Interface Guidelines da Apple. Ao ter como base essas referências mundiais, é possível entender padrões já estabelecidos no mercado, podendo assim incorporar para a proposta de solução em questão.

Além desse processo, é necessário definir quais componentes estão no *core* do DS, esse inventário é definido com base na auditoria das interfaces gráficas construídas na fase do projeto, onde há todos os componentes que mais se repetem no sistema. Essa lista tem como objetivo conter os componentes que mais são reutilizados para que assim não seja necessário a construção dos mesmos em todo momento de definição de uma nova interface gráfica para o sistema. Com essa lista definida, é possível construir os Design Tokens, sendo estes a ponte de contato entre as decisões de design para os componentes na camada *core* e engenharia. Essa construção pode ser feita de forma colaborativa entre as pessoas que construirão a solução e quem estará usando o DS no dia a dia.

A próxima etapa desta fase do método é a construção dos componentes *core* na ferramenta utilizada pelos designers para a construção dos protótipos de interface gráfica. Essa construção define cada formato e comportamento de como deve ser tal elemento que compõem a interface, utilizando as decisões visual contidas nos design tokens. É através dessa construção que habilitamos a reutilização de tais elementos pelos designers que trabalham na construção do sistema de software, havendo essa biblioteca construída, servirá de base para a construção da mesma no ambiente de código usado pelos engenheiros.

Para isso é necessário a análise de qual tecnologia é utilizada para a construção do visual da interface, se é mobile ou web, e quais frameworks de construção são utilizados, pois com essas informações, pode se construir a lógica de transformação dos design tokens em variáveis em cada tipo de tecnologia e também como os componentes *core* serão envolvidos por essas tecnologias específicas para serem utilizados na construção do sistema de software. Essa lógica de disponibilização é utilizada para garantir que o Design System se mantenha agnóstico, não sendo dependente de tecnologia, facilitando assim possíveis mudanças futuras.

E para finalizar a fase de projeto e desenvolvimento, se faz necessário a construção da documentação que apoia e auxilia os designers e desenvolvedores a utilizar o design system em cada contexto. Tendo isso como objetivo para garantir o

entendimento das pessoas que serão impactadas pela intervenção do problema em questão.

4.5 INTERVENÇÃO

Segundo Van Aken *et al.* (2012), é neste momento que ocorre a implementação da proposta de solução para o problema em questão na empresa, para termos essa implementação é necessário disponibilizar a biblioteca construídas para os designers e a biblioteca em código para o time de engenharia, essa distribuição pode ser feita de forma gradual ou imediata para todas as pessoas envolvidas. Essa decisão irá depender de como é a organização da empresa quanto a distribuição das equipes que constroem a interface gráfica do sistema.

4.6 APRENDIZAGEM E AVALIAÇÃO

Tanto para Peffers *et al.* (2007) e Van Aken *et al.* (2012) é nessa fase após a implementação que o pesquisador deve avaliar como está o contexto do problema após ter a solução implementada. Para avaliarmos essa situação, podemos utilizar de entrevistas como método qualitativo de pesquisa para compreender as percepções das pessoas que foram impactadas pela solução, vale destacar aqui que os resultados da pesquisa realizada na primeira etapa do processo serve de base comparativa do antes e depois da aplicação do Design System. Além disso, podemos avaliar de forma quantitativa através do levantamento do retorno sobre investimento, onde podemos realizar o cálculo do quanto o Design System está trazendo de ganho para a empresa em horas reduzidas de trabalho para construção de tela, menos o valor investido para a implementação da solução e sua manutenção, dividido pelo valor investido, o resultado representará a proporção de para cada valor investido o quanto obteve de retorno.

4.7 COMUNICAÇÃO

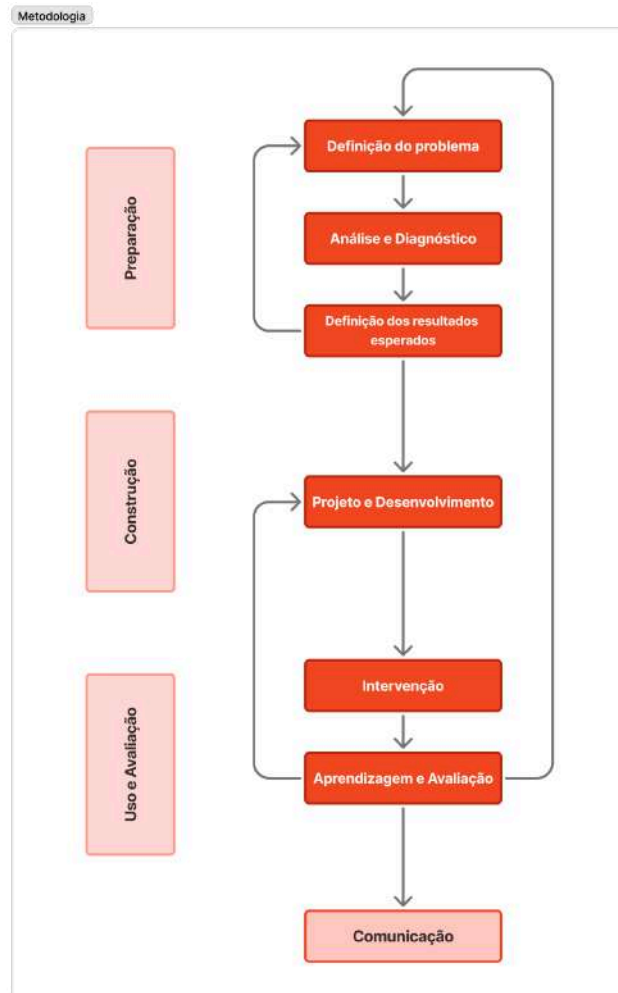
Como última fase do processo, tendo como referência Peffers *et al.* (2007), deve se fazer a comunicação para as pessoas não técnicas dos impactos gerados pela a solução proposta, mostrando os resultados da avaliação e os métodos

utilizados. Além disso, nesse momento o pesquisador pode evidenciar se a solução precisa de melhorias, dado a avaliação feita na etapa anterior, tendo assim a liberdade de propor um novo ciclo de iteração para a metodologia proposta nesta pesquisa.

5 ANÁLISE E RESULTADOS

Essa seção contém a análise e resultados obtidos a partir da intervenção proposta para a questão central desta pesquisa. A qual trata-se sobre como manter a consistência visual em um sistema de software interativo, através da abordagem de Design System, sendo aplicado em uma empresa privada brasileira, do setor de marketing digital, tal aplicação seguiu as etapas da metodologia apresentadas na seção anterior.

Figura 19 – Metodologia proposta nesta pesquisa



Fonte: Autoria própria, (2025)

A aplicação desta metodologia se deu em três momentos: Preparação, Construção e Uso e Avaliação da solução. Durante a **Preparação** acontece a **Definição do Problema**, onde foi feito todo o entendimento do contexto através do

mapeamento de stakeholders, entrevistas e a auditoria de componentes para identificar quais são os pontos de problema ao construir as interfaces gráficas. Há também a **Análise e Diagnóstico**, onde foi analisado e categorizado os problemas evidenciados nas entrevistas executadas na etapa anterior, e por último, tendo todos esses insumos sobre o problema e o contexto, foi realizado uma **Definição Dos Resultados Esperados** pela empresa em questão para o desenvolvimento do Design System dentro do prazo de 3 meses.

O segundo momento **Construção** está relacionado a etapa metodológica do **Projeto e Desenvolvimento**, onde foi realizado a materialização do Design System, para isso foi necessário a definição da lista de componentes *Core*, a estrutura de *Design Tokens*, responsável pela conexão entre Design e Engenharia, a construção da biblioteca para Designers e a construção da biblioteca em código para os desenvolvedores. No terceiro momento, acontece a **Intervenção** e a **Avaliação** do Design System, sendo aqui o primeiro contato dos Designers e Engenheiros com a solução proposta, no contexto em questão as avaliações com cada grupo foram feitas após 3 meses de uso, sendo captadas através de entrevistas qualitativas, comparando com as entrevistas feitas no início da pesquisa, além disso, foi feita uma estimativa do retorno sobre o investimento realizado na solução em questão. Os 3 momentos serão descritos com mais detalhes nas seções 5.1, 5.2 e 5.3.

5.1 PREPARAÇÃO

Para dar início ao entendimento do contexto da empresa de marketing brasileira quanto ao processo de construção de interfaces gráficas para o desenvolvimento do software interativo de uso interno, foi necessário a construção de um artefato chamado mapa de stakeholders, com o objetivo de entender quais pessoas estavam relacionadas e seriam impactadas pelo projeto de construção do Design System, assim realizando a etapa metodológica de *Definição do problema* da DRS de VAN AKEN *et al.* (2012) e PEFFERS *et al.* (2007).

Para realizar a construção de tal artefato foi necessário categorizar em dois grupos de interesse, stakeholders primário e secundário e sua posição à

organização, interna ou externa. (United States Agency International Development (USAID), 2018). Veja abaixo o quadro com o detalhamento de cada categoria:

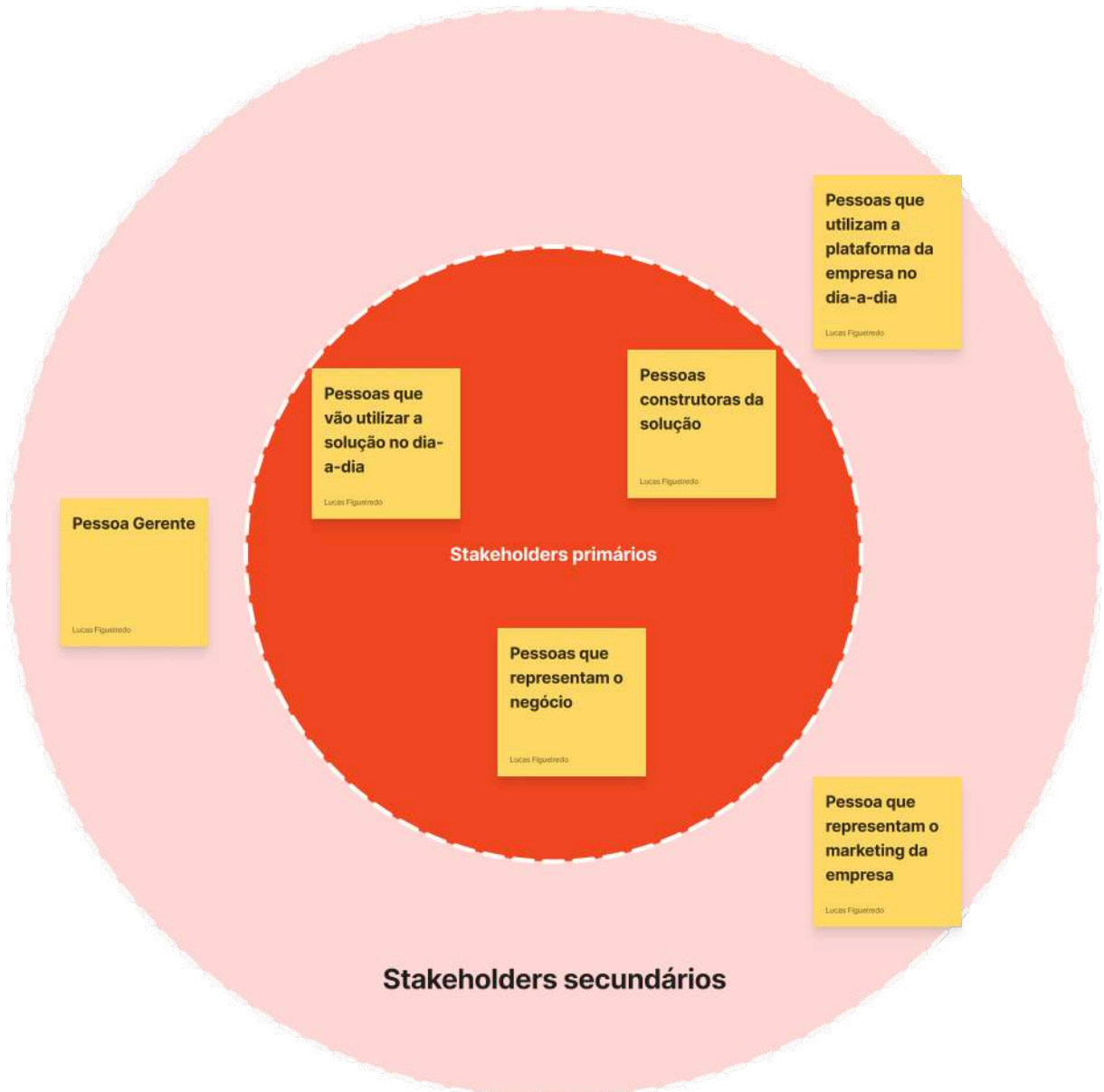
Quadro 1 – Categorias de Stakeholders

| Categorias | Definições |
|-------------------------------|--|
| Stakeholder Primário | São as pessoas envolvidas de forma direta ao projeto, tendo interesse direto e imediato no que é feito para a realização do projeto e impactam nas decisões tomadas. |
| Stakeholder Secundário | São pessoas que estão envolvidas de forma indireta, serão impactadas mas não influenciam diretamente as decisões tomadas. |
| Interno | São as pessoas stakeholders que estão inseridas no contexto interno da empresa |
| Externo | São as pessoas stakeholders que não estão inseridas no contexto interno da empresa |

Fonte: Autoria própria, (2025)

Com base em uma conversa inicial com as pessoas que representam a diretoria da organização e com a compreensão das pessoas que compõem a equipe que trabalha no desenvolvimento de software dentro da empresa, chegamos a conclusão da seguinte disposição dos stakeholders para este projeto:

Figura 20 – Mapa de Stakeholders



Fonte: Autoria própria, (2025)

Dentro do círculo que representa as pessoas stakeholders primárias internas, existem 3 grupos: as pessoas que irão utilizar o Design System, divididas entre time de design e desenvolvimento; As pessoas construtoras, representantes técnicos de design e desenvolvimento que irão atuar na construção da proposta de solução;

Decisores, representantes da diretoria da empresa, atuando no acompanhamento da solução e garantindo que os objetivos estejam alinhados com a direção da empresa. Já o grupo de pessoas stakeholders secundárias internas, ficou composto pela pessoa agilista do projeto e o representante da equipe de marketing da empresa, por fim, o último grupo composto pelas pessoas stakeholders secundárias externas, representou as pessoas usuárias do software desenvolvido pela empresa.

Tendo então esse mapa de stakeholders em mãos é possível observar qual tipo de informação deve ser lavada pelos consultores ao se referir ao projeto, por exemplo, as pessoas decisoras serão informadas com o andamento do projeto, quanto tempo será necessário para a execução das demandas, e outras questões que conversam mais com o planejamento e objetivos de negócio, do que como será executado. Por um outro lado, o como será executado pode impactar as pessoas utilizadoras do Design System, sendo fundamental a compreensão de como é o contexto de construção de interfaces gráficas que elas seguem. Dessa forma, como complemento a esse artefato, foi realizada uma entrevista semiestruturada como levantamento de informações qualitativas sobre o contexto em que existiam problemas. No próximo tópico será exposto como foram realizadas tais entrevistas e seus resultados.

5.1.1 ENTREVISTAS SEMIESTRUTURADAS

Nesta seção é apresentado os resultados em relação às entrevistas realizadas com os principais stakeholders nesta fase de *Análise e Diagnóstico*, sendo as pessoas utilizadoras, construtoras e a representante da equipe de marketing. Foram no total 6 entrevistas, obtendo assim um panorama geral do contexto do problema na visão de tais pessoas. Para se chegar nesse resultado foi definido o objetivo da entrevista, o roteiro de perguntas que se encontra no Apêndice B desta pesquisa e o público alvo. O objetivo foi guiado pela fase da Análise e Diagnóstico do ciclo para resolução de problemas de Van Aken *et al.* (2012), visando a busca pelo entendimento completo do contexto e ambiente em que o problema acontece. Sendo assim, o objetivo definido foi: identificar as principais dificuldades no contexto de construção das interfaces gráficas e as expectativas das pessoas externas ao projeto.

Com o objetivo definido, criou-se o roteiro para as entrevistas, para cada grupo de pessoas entrevistadas houve a confecção de um roteiro com perguntas diferentes, sendo para o grupo de pessoas que iriam utilizar o Design System, divididas entre designers e desenvolvedoras, as construtoras, em específico da parte de desenvolvimento de código e a representante de marketing da empresa. As perguntas podem ser vistas por completo no Apêndice B ao final desta pesquisa. Após a execução das entrevistas, foi realizada a análise temática dos fatos evidenciados pelas pessoas entrevistadas, categorizando assim entre grupo entrevistado por tipos de dificuldades, necessidades, contexto atual e expectativas. Na próxima seção será evidenciado todos os detalhes da análise temática.

1. PESSOAS UTILIZADORAS DO DESIGN SYSTEM

Após a realização das entrevistas com duas pessoas da área de Design e duas pessoas Engenheiras, ficou mais explícito o cenário que estava acontecendo no processo de construção das interfaces gráficas para software em desenvolvimento para a empresa de marketing. Tal processo era composto pela confecção de protótipos por parte dos designers para serem feitos testes de usabilidade e assim que as telas estivessem validadas, o protótipo serviria de artefato para a construção e composição de user stories para as pessoas engenheiras responsáveis. Porém para as pessoas Designers, já havia um kit de elementos de interface pré-prontos para utilizarem nesse processo, mas esse kit não era refletido em engenharia, o que havia em código era uma biblioteca sem consistência visual e operacional com o kit, havendo assim componentes duplicados e sem padronização na construção. E é a partir desse cenário de inconsistência entre os processos que as interfaces gráficas chegavam às pessoas usuárias finais com inconsistências visuais, comprometendo assim a consistência interna do sistema e as expectativas dos usuários. (Rachel Krause, 2021; Norman, 1988; Tognazzini, 2003.).

Além desse contexto descrito nas entrevistas, houveram outros pontos mencionados pelos dois grupos entrevistados, em relação a dificuldades,

necessidades e expectativas que tinham quanto à proposta de solução. Nas próximas duas seções estarão explícitas tais dizeres captados nas entrevistas.

a. DESIGNERS

O grupo em questão era composto por 2 profissionais, dos quais relataram terem experiência ao utilizar componentes construídos na plataforma de confecção de protótipos, o Figma. Além disso, cada pessoa trabalhava em uma parte específica do sistema. A seguir estão as tabelas com os pontos evidenciados por tais pessoas.

Quadro 2 – Dificuldades evidenciadas pelas Designers

| | Uso de componentes | Documentação | Nomenclatura e Organização |
|---------------------|--|---|--|
| Dificuldades | Componente de Tabela é difícil de adaptar para o contexto de uso | Text field e dropdown são os componentes com mais dificuldade para encontrar o padrão de altura e largura | Encontrar um componente específico na organização que temos atualmente |
| | Dúvida sobre variação de altura e largura dos componentes | Identificar como os componentes e atributos de estilos devem ser utilizados | Encontrar ícones utilizando o idioma Português brasileiro. |
| | Dúvida sobre como usar o atributo de elevação nos componentes | | |

Fonte: Dados da pesquisa, (2025)

Quadro 3 – Necessidades evidenciadas pelas Designers

| | | | |
|---------------------|---|-------------------------|--|
| Necessidades | ícones com escala padronizada de tamanhos | Mais variações de cores | Componentes para contexto de representação de gráficos dentro do sistema |
|---------------------|---|-------------------------|--|

Fonte: Dados da pesquisa, (2025)

Quadro 4 – Expectativas evidenciadas pelas Designers

| | | | |
|---------------------|---|--|--|
| Expectativas | Documentação de processos quanto a publicação dos componentes | Documentação de como utilizar o componente | Componentes para contexto de representação de gráficos dentro do sistema |
|---------------------|---|--|--|

. Fonte: Dados da pesquisa, (2025)

b. ENGENHEIRAS

Já o grupo de pessoas engenheiras era composto de 2 pessoas, das quais relataram como era o processo de construção das interfaces gráficas no ambiente de desenvolvimento, ambas atuavam em equipes diferentes e tinham diferentes níveis de experiência. Segue abaixo as tabelas com os pontos evidenciados:

Quadro 5 – Dificuldades evidenciadas pelas Engenheiras

| | Inconsistência | Construção de componentes | Disponibilização dos componentes |
|--|--|---|--|
| | Muitos componentes não foram atualizados para o visual atual | Falta de padrão em uso de biblioteca auxiliares para construção dos componentes | Falta de definição se um componente deve ser local ou global na estrutura de arquivo de acesso |

| | | | |
|----------------------|--|---|--|
| Más prácticas | que temos nas telas do Figma | | |
| | Falta de processo de atualização dos componentes | Componentes construídos de forma específica, sem ser genérico ao contexto | Falta de padronização na construção de componentes |
| | | | Componente com textos fixos, sem a possibilidade de editar |

Fonte: Dados da pesquisa, (2025)

2. PESSOAS CONSTRUTORAS

O grupo de pessoas construtoras do Design System eram compostas por uma pessoa Designer e outra Engenheira, a representante de Designer foi a responsável pela aplicação das entrevistas e executar o processo descrito nesta pesquisa. Já a representante de Engenharia foi responsável pela construção de toda a arquitetura em código para que o Design System fosse construído, seguindo a visão de ecossistema evidenciada por Frost (2023). Esta etapa de entrevista foi apenas aplicada com a pessoa engenheira, pelo fato da representante de Design estar no contexto de aplicação e execução da metodologia. Na próxima seção será demonstrado os principais pontos de preocupação na visão da pessoa construtora de engenharia.

a. ENGENHEIRA

A entrevista teve um foco mais nas restrições no contexto de observação, experiências anteriores, tecnologia utilizada, processos atuais e os objetivos pessoais para o projeto. Para assim entender melhor as preocupações na visão do construtor. No quadro 6 e 7 constam todos os pontos levantados em torno destas questões.

Quadro 6 – Contexto atual na visão de Engenharia

| | Documentação | Limitações | Experiência |
|-----------------------|--|--|--|
| Contexto Atual | Documentação atual não dá visibilidade de todos os componentes que existem | Construir componentes do zero pode tornar o projeto mais complicado pois tem que pensar na adaptação dos componentes em diferentes telas | Trabalhou em contextos usando bibliotecas auxiliares como Bootstrap. |
| | Poucos componentes documentados | Não existe nenhuma limitação pelo React para construir os componentes propostos no Figma | |
| | Nomenclatura dos componentes diferente do Figma | Componentes atuais construídos em de forma específica, impedindo a aplicação em outros contextos, não sendo um componente genérico. | |

Fonte: Dados da pesquisa, (2025)

b. REPRESENTANTE DE MARKETING

A entrevista com a pessoa representante de marketing foi realizada para compreender como eram as regras de aplicação da marca da empresa e como poderíamos trazer esses detalhes para dentro do sistema em desenvolvimento. Além disso, compreendemos também como era o processo de construção das interfaces gráficas voltadas para as vendas, tais interfaces ficavam em um escopo separado do sistema de software interativo analisado nesta pesquisa, para entender se haveria algumas oportunidades do Design System proposto apoiar nesse

processo. No quadro 7 abaixo estão os principais fatos levantados através dessa entrevista.

Quadro 7 – Contexto atual relacionado ao Marketing da empresa

| | Marca da Empresa | Processo de Construção | Ferramentas Utilizadas |
|-----------------------|---|---|--|
| Contexto Atual | Não existe uma equipe dedicada à marca da empresa | Não existe um guia de estilo definido dentro da ferramenta de prototipação, Figma | GreatPages, ferramenta usada para a publicação da página na Web. |
| | Uso da cor vermelha para questões principais. Iconografia deve ser apenas com bordas. | Uso de vários elementos criativos, como imagens e ilustrações | Figma, ferramenta usada para a prototipação das páginas |
| | Não usar flat design em ilustrações. | Existem modelos diferentes de páginas conforme o tipo de venda | |

Fonte: Dados da pesquisa, (2025)

Dado o contexto atual evidenciado pela entrevista, ficou claro que a proposta de solução não encaixam no escopo de trabalho da equipe de marketing, e o principal ponto de convergência, que garante a aceitação da proposta do ponto de vista quanto a marca da empresa, foi o uso das cores principais da marca, o vermelho e o preto, além de manter os estilos de iconografia e o não uso do flat design em ilustrações.

3. ANÁLISE DAS ENTREVISTAS SEMIESTRUTURADAS

Para garantir uma maior solidez epistemológica em cima desta investigação, recorreremos ao método de Bardin (2016), o qual é reconhecido por oferecer um roteiro metodológico rigoroso para a análise dos dados qualitativos captados nas entrevistas. Conforme a autora salienta, o método visa transformar um conteúdo de dados brutos – entrevistas, documentos ou interações – em informação estruturada capaz de sustentar inferências válidas sobre fenômenos sociais (BARDIN, 2016).

Tendo então como guia este método, iniciamos pela pré-análise, sendo a fase de aproximação flutuante que delinea o corpus, evidencia os objetivos de codificação e estabelece as unidades de registro e de contexto. Essa preparação – comparável a “um mergulho inicial que ancora todo o restante do percurso” (BARDIN, 2016, p. 123), garantiu a pertinência entre as perguntas da pesquisa e as categorias a priori, sem desconsiderar a emergência de temas inesperados.

Em seguida, partimos para a exploração do material, etapa em que o texto é fracionado, codificado e reagrupado em categorias mutuamente exclusivas e exaustivas. Aqui, foi adotado um procedimento híbrido, combinando categorias dedutivas (derivadas das hipóteses) e indutivas (derivadas da própria fala das participantes). Tal estratégia respeita a recomendação de Bardin (2016) de permitir que “o material fale” antes de qualquer forçosa adequação a esquemas prévios, garantindo fidelidade às nuances do discurso.

E a última etapa, o tratamento dos resultados – com interpretação analítica e quantificação auxiliar de frequências – possibilitou confrontar cada categoria aos objetivos de pesquisa, produzindo inferências que dialogam com questões como dores de uso, documentação, Nomenclatura & Organização, Processos & Governança e Critérios de Sucesso. De acordo com Bardin (2016), é nessa fase que o pesquisador atua, ao fazer inferências, convertendo dados em conhecimento acionável, exatamente o que se buscou ao propor recomendações práticas para a proposta deste trabalho.

a. PRÉ-ANÁLISE

Constituição do corpus – foram consideradas as transcrições completas das entrevistas com: duas pessoas projetistas da equipe de Design, três desenvolvedores front-end, um profissional de marketing e a pessoa construtora em engenharia sendo o “construtor” do DS. Esse material corresponde a 12 páginas de texto contínuo.

Leitura flutuante – uma leitura exploratória identificou sete temas reiterados: dores ao construir telas, limitações da biblioteca atual, documentação, nomenclatura/organização, necessidades de novos recursos, processos pouco claros e fatores de sucesso.

Objetivo de codificação – responder: “Quais barreiras, necessidades e critérios de êxito são percebidos antes da adoção de um Design System na Empresa Privada de Marketing?”

Unidades – unidade de registro: enunciado semântico (frase ou sintagma opinativo); unidade de contexto: bloco temático específico (ex.: “Dificuldades – Documentação”).

b. EXPLORAÇÃO DO MATERIAL (CODIFICAÇÃO & CATEGORIZAÇÃO)

Foi adotada codificação aberta, seguida de agrupamento em categorias mistas (a priori + emergentes). O quadro abaixo sintetiza 46 unidades de registro:

Quadro 8 – Exemplo prático da composição da nomenclatura dos Design Tokens

| Macro-categoria | Subcategorias (exemplos) | n |
|-----------------|--|----|
| Dores de Uso | Tabela, Dropdown, tamanhos, collapsed components | 13 |

| | | |
|----------------------------|---|---|
| Documentação | Falta de guia único, ausência de exemplos, tokens obscuros | 8 |
| Nomenclatura & Organização | Ícones PT/EN, nomes divergentes entre Figma e código | 5 |
| Processos & Governança | Publicação, manutenção parada, pessoa engenheira dedicada | 6 |
| Necessidades Funcionais | Ícones escalonados, mais cores, gráficos, Dialogs | 7 |
| Critérios de Sucesso | Processos claros, documentação robusta, componentes genéricos | 4 |
| Restrições / Limitações | Aderência a bibliotecas externas, falta de experiência prévia | 3 |

Fonte: Dados da pesquisa, (2025)

c. TRATAMENTO, INFERÊNCIA E INTERPRETAÇÃO

- **Dores de Uso** predominam (28 % do corpus). Designers relatam dificuldades com o componente de tabela e dimensões padronizadas; pessoas engenheiras apontam componentes engessados ou obsoletos que levam à criação local.
- **Documentação** é vista como ausente ou fragmentada (“não está muito documentada”, “preciso saber o padrão definido”).
- **Nomenclatura & Organização** surgem como gargalo transversal: nomes diferentes entre Figma e React impedem reaproveitamento.
- **Processos & Governança** – Não há fluxo visível para propor, aprovar ou manter componentes; devs mencionam “biblioteca parada”.
- **Critérios de Sucesso** – Entrevistados concordam que DS bem-sucedido exigirá documentação abrangente, componentes genéricos, canal único de comunicação e manutenção contínua.

d. RECOMENDAÇÕES INICIAIS

- **Mapa de componentes existentes** (Figma × Front) para eliminar duplicidades e alinhar nomenclaturas.
- **Guia de contribuição** (processo de design → engenharia → publicação) versionado e visível.
- **Documentação living-style-guide** com exemplos de código, tokens e boas práticas.
- **Roadmap de manutenção** com responsável dedicado e ciclos de revisão.

Como evidenciado anteriormente, o cenário antes da aplicação da intervenção era de **fragmentação estrutural**: ausência de documentação, nomenclaturas divergentes e processos inexistentes minavam a adoção da biblioteca anterior tanto pela a equipe de Design quanto por desenvolvedores. Esse quadro inicial justificou a implantação do Design System e ofereceu um mapa de prioridades que serão descritas nas próximas seções.

5.1.2 AUDITORIA DOS COMPONENTES

A confecção da auditoria foi realizada com o objetivo de fazer o levantamento dos componentes mais utilizados no sistema de software e as inconsistências que haviam neles, tendo assim a lista inicial de *ui components* da camada *Core* do design system, e um artefato visual provando as inconsistências. (FROST, 2022). No Quadro 9 e Figura 22 estão todos os componentes mapeados. Além disso, serão mostradas as evidências de inconsistências encontradas.

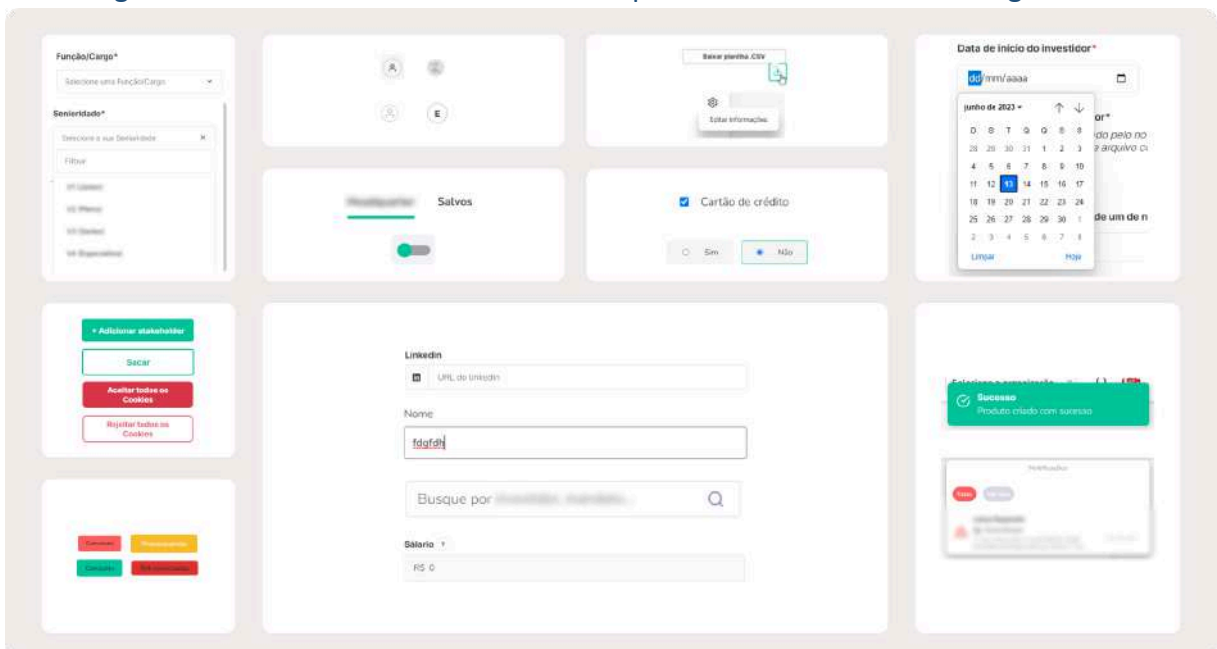
Quadro 9 – Lista de componentes levantados na Auditoria

| Nome | Categoria | Kit de elementos no Figma | Biblioteca de Componentes em Código |
|--------|-----------|---------------------------|-------------------------------------|
| Button | Action | Sim | Sim |
| Avatar | Others | Sim | Sim |

| | | | |
|-------------|------------|-----|-----|
| Badge | Others | Sim | Sim |
| Toast | Dialog | Sim | Sim |
| Tooltip | Dialog | Sim | Sim |
| Tab | Navigation | Sim | Sim |
| Textfield | Form | Sim | Sim |
| Searchfield | Form | Sim | Sim |
| Dropdown | Form | Sim | Sim |
| Calendar | Form | Sim | Sim |
| Radiobutton | Form | Sim | Sim |
| Checkbox | Form | Sim | Sim |
| Switch | Action | Sim | Sim |
| Breadcrumb | Navigation | Sim | Sim |

Fonte: Autoria própria, (2025)

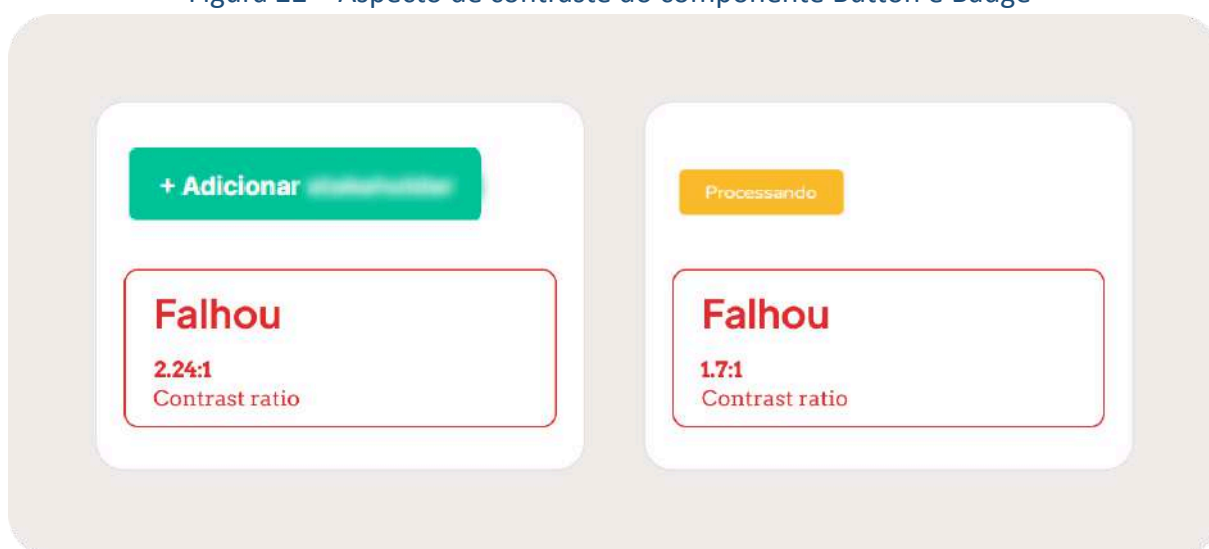
Figura 21 – Demonstrativo visual dos componentes na biblioteca de engenharia



Fonte: Autoria própria, (2025)

Foram mapeados cerca de 14 componentes, os mesmos apareciam em quase todas as telas do sistema interativo. Além disso, conseguimos mapear as principais inconsistências que haviam entre componentes de mesmo nome, porém dependendo da tela em que estivesse, o mesmo componente estava diferente, comprometendo assim a consistência interna do software em desenvolvimento. (RACHEL, 2021). Outro fator comprometido em alguns dos componentes, como a Badge e Button, foi o critério 1.4.3 - Contraste (Mínimo) (Nível AA), estipulado pela a organização W3C responsável pela criação do guia de acessibilidade na Web WCAG. Dessa forma, impedindo que boa parte das pessoas pudessem ler o conteúdo de texto presente no componente. Veja na figura 23 o aspect ratio dos dois componentes.

Figura 22 – Aspecto de contraste do componente Button e Badge



Fonte: Autoria própria, (2025)

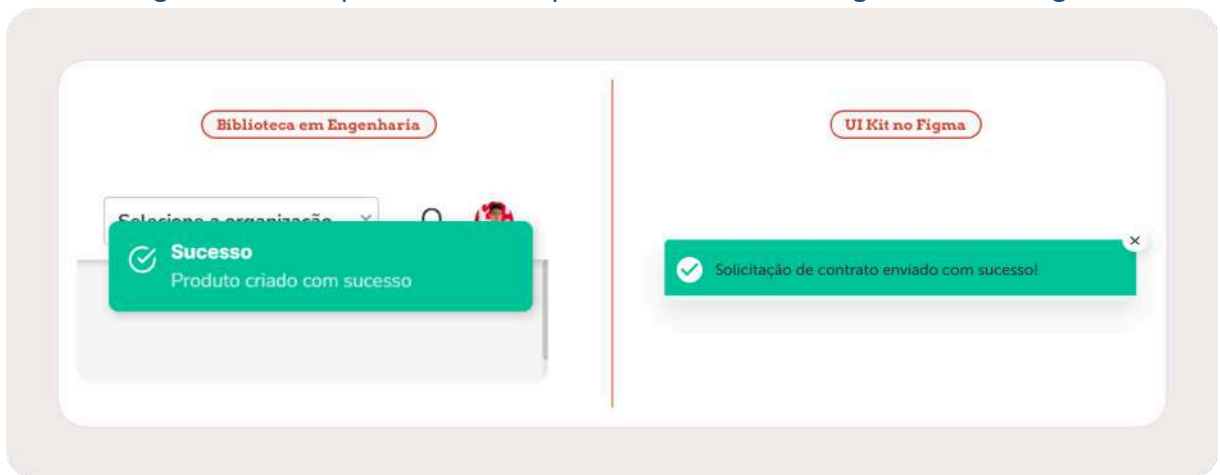
A recomendação para o critério de contraste mínimo segunda a W3C é de 4.5:1 para textos menores que 18 pontos. E para complementar a auditoria, as próximas figuras mostram a inconsistência entre os componentes validados pela equipe de Design e os componentes construídos em Engenharia.

Figura 23 – Comparativo do componente Searchfield entre Engenharia e Design



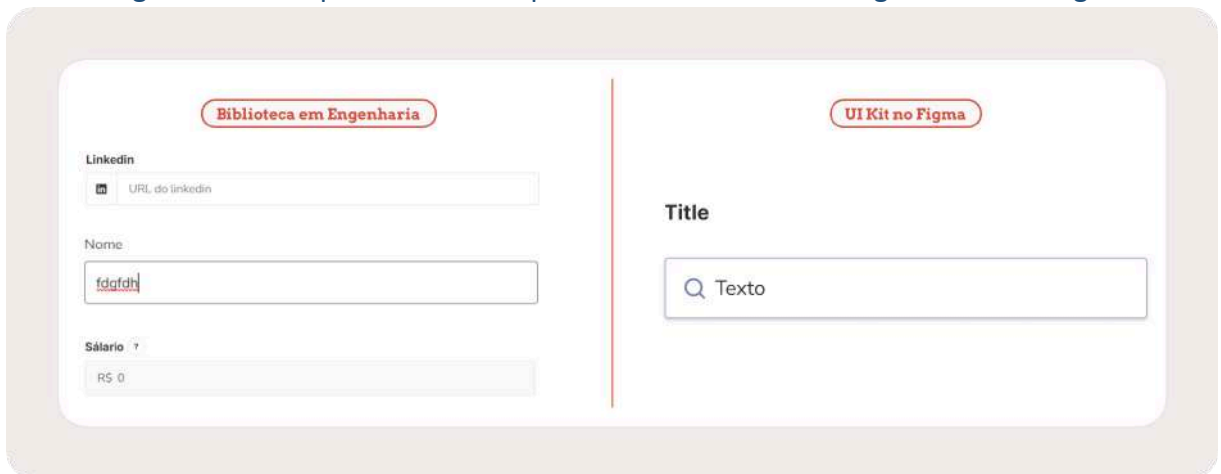
Fonte: Autoria própria, (2025)

Figura 24 – Comparativo do componente Toast entre Engenharia e Design



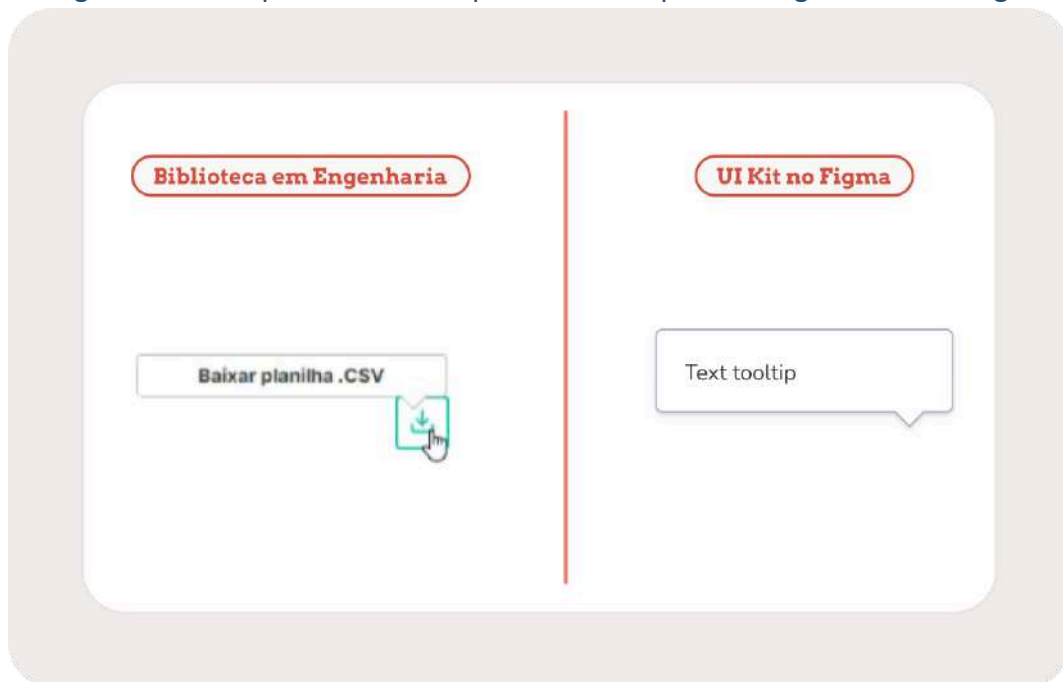
Fonte: Autoria própria, (2025)

Figura 25 – Comparativo do componente Textfield entre Engenharia e Design



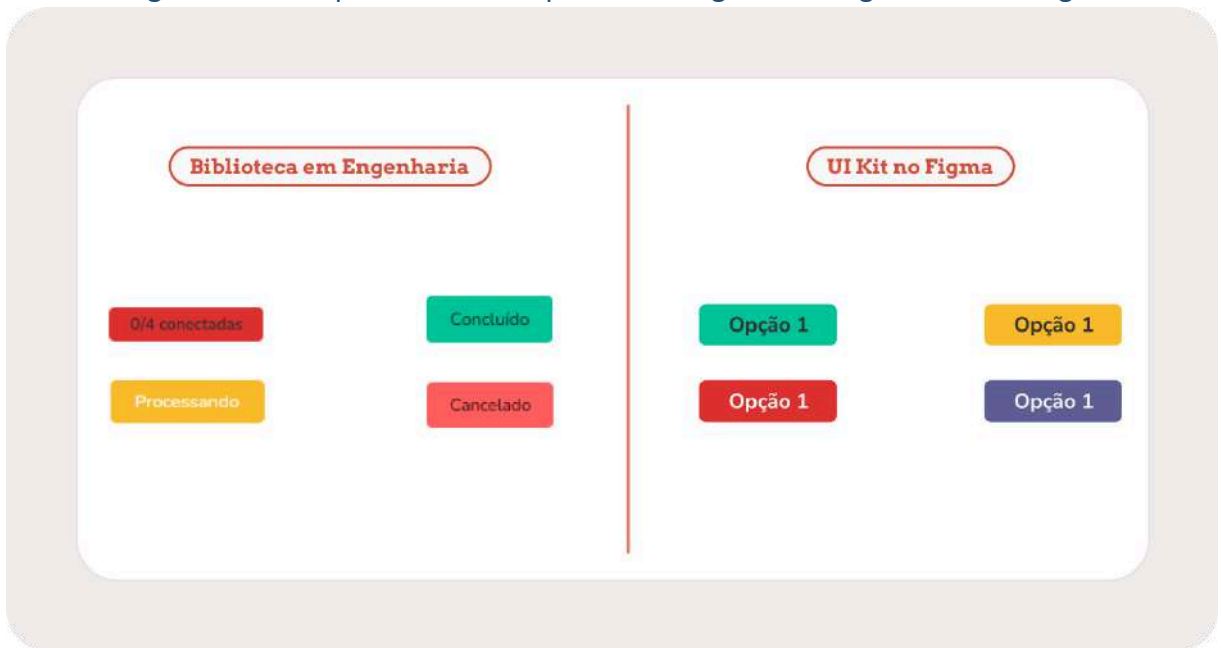
Fonte: Autoria própria, (2025)

Figura 26 – Comparativo do componente Tooltip entre Engenharia e Design



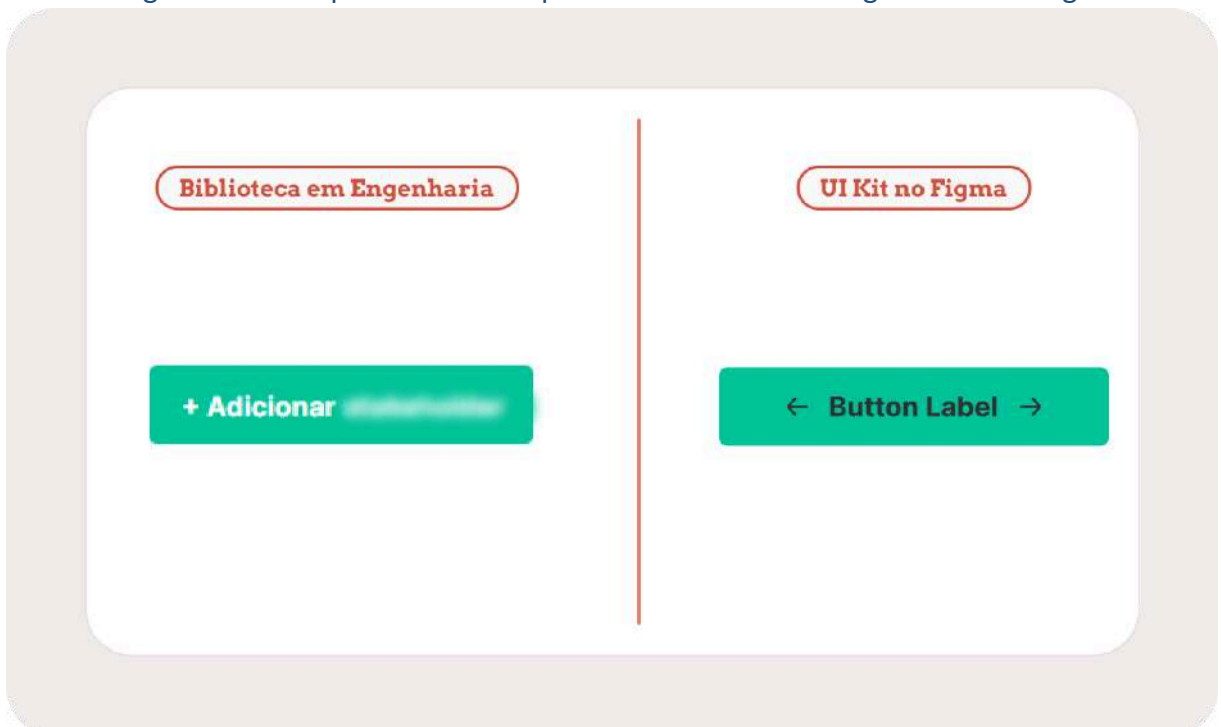
Fonte: Autoria própria, (2025)

Figura 27 – Comparativo do componente Badge entre Engenharia e Design



Fonte: Autoria própria, (2025)

Figura 28 – Comparativo do componente Button entre Engenharia e Design



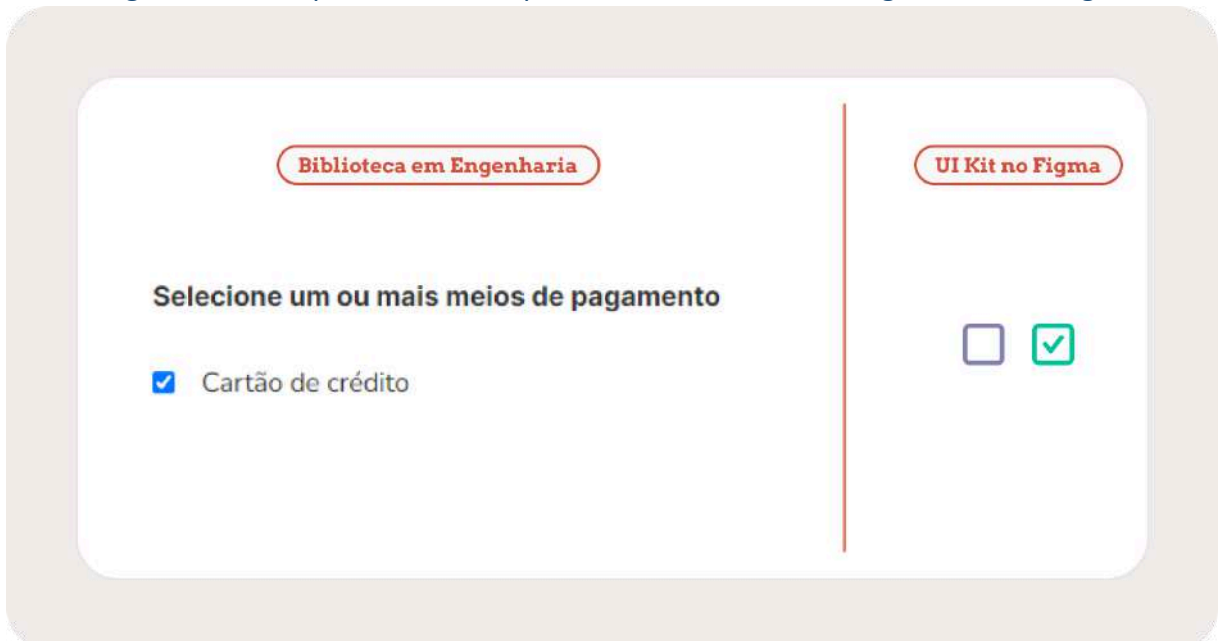
Fonte: Autoria própria, (2025)

Figura 29 – Comparativo do componente Radiobutton entre Engenharia e Design



Fonte: Autoria própria, (2025)

Figura 30 – Comparativo do componente Checkbox entre Engenharia e Design



Fonte: Autoria própria, (2025)

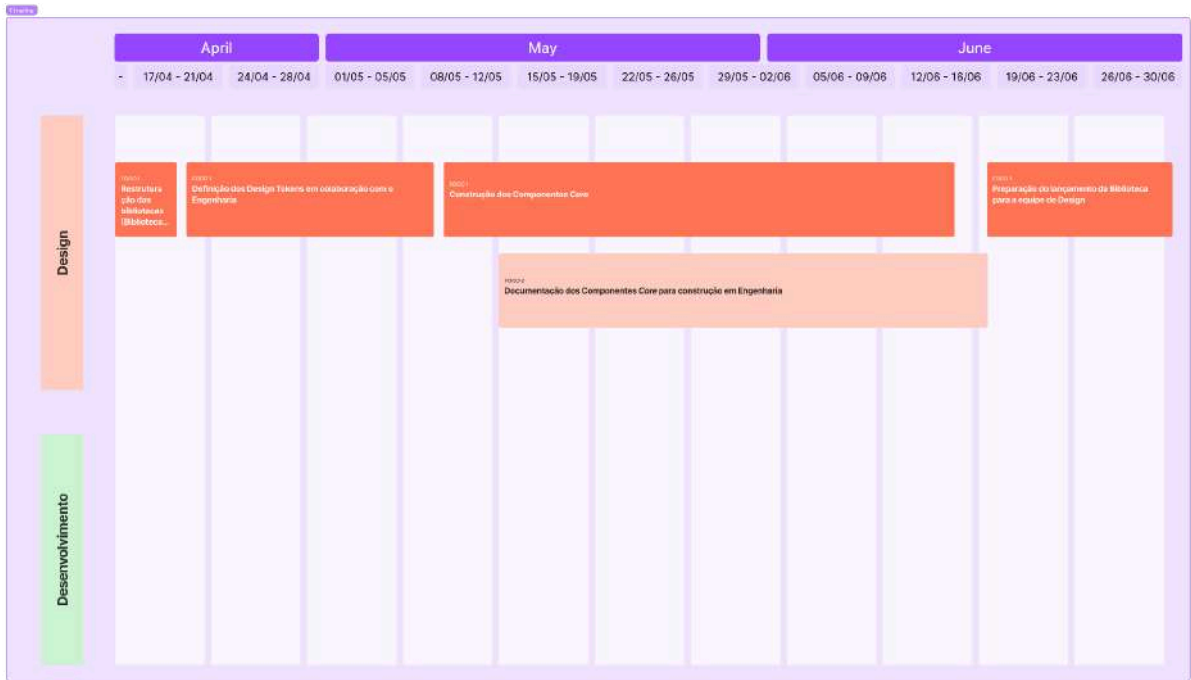
Havendo assim todas as evidências de problemas e o entendimento sobre o contexto, realizado com as entrevistas e a auditoria, conseguimos então ir para a

última fase do momento de *Preparação à Definição dos Resultados Esperados*, tal definição foi decidida juntamente com os stakeholders primários decisores, para que assim o projeto estivesse alinhado com os objetivos da empresa, para assim termos uma solução que contemplasse todos os stakeholders do projeto. Na próxima seção haverá os detalhes do que foi decidido nesta etapa.

5.1.3 ROADMAP DO PROJETO

O roadmap do projeto foi um formato utilizado para conectarmos os resultados esperados pelo tempo de execução, onde foi mapeado uma estimativa de quando cada etapa de construção estaria finalizada por cada nível de construtor, demandas envolvendo Design e posteriormente Engenharia. Dessa forma, ficou esperado que após 3 meses do início da construção do projeto, o Design System estaria com sua camada *Core* estruturada para a equipe de Design e todos os componentes *Core* estariam documentados para que a pessoa construtora de Engenharia começasse a estruturar o Design System para tal contexto, e assim após mais 3 meses, a estrutura de Design System deveria estar pronta na camada de engenharia. Para chegarmos então a esse objetivo, foi quebrada em etapas as demandas que seriam executadas no contexto de Design, havendo apoio de Engenharia, durante esse período de 3 meses e também as etapas de execução para o período consequente. As Figuras 31, 32 e 33 representam as demandas realizadas em cada período de tempo.

Figura 31 – Roadmap execução das demandas com foco em Design



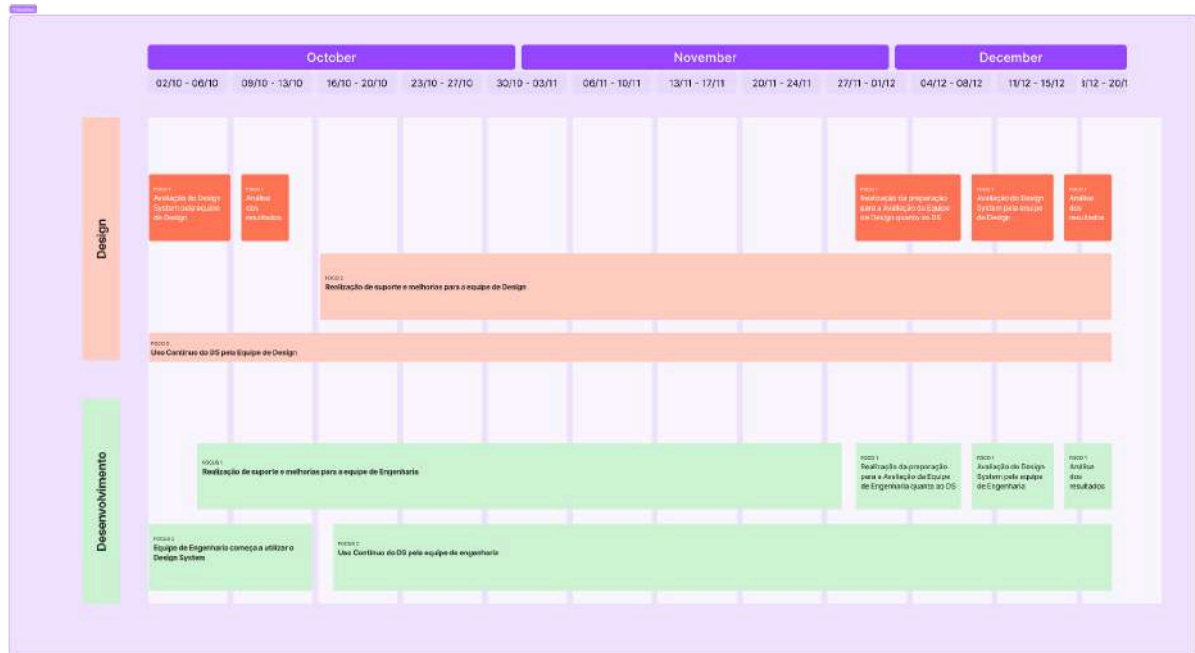
Fonte: Autoria própria, (2025)

Figura 32 – Roadmap execução das demandas com foco em Design e Engenharia



Fonte: Autoria própria, (2025)

Figura 33 – Roadmap execução das demandas com foco em Avaliação



Fonte: Autoria própria, (2025)

Com as demandas definidas, tínhamos o alinhamento dos resultados esperados para o projeto e assim o passo a passo de como realizar a construção da solução proposta para os problemas mapeados anteriormente. Na próxima seção será explicitado os detalhes feitos para o momento de *Construção*.

5.2 CONSTRUÇÃO

A partir dos detalhes sobre o contexto atual na construção de interfaces gráficas na empresa brasileira de marketing digital, foi possível entender os principais problemas desse processo e mapear as demandas para a construção do artefato interventivo para a solução. Havendo assim a definição de quais componentes fariam parte da camada *Core*, a próxima etapa é a construção da subcamada de Design Tokens, definindo assim a nomenclatura para as variáveis agnósticas que representam os atributos visuais que compõem os elementos da interface gráfica. (SÖNKE ROHDE, 2014).

5.2.1 DEFINIÇÃO DA LINGUAGEM ÚNICA

Esse processo foi realizado em conjunto com a pessoa construtora de engenharia em uma discussão livre de quais atributos seriam considerados como Design Tokens. Como o objetivo desta pesquisa está focado na construção de uma linguagem única entre Design e Engenharia, será apresentado apenas como foi definido essa nomenclatura. A organização e lógica de utilização dos design tokens não será detalhada.

No Quadro 10 e 11 é possível observar a lista de opções para cada nível que compõem um Design Token. E no Quadro 8 é evidenciado os tipos de Design Tokens e sua nomenclatura composta pelas opções de cada nível.

Já a Figura 34 do Apêndice A exemplifica toda essa lógica de níveis da nomenclatura e os tipos de Design Tokens, baseado no estudo de Nathan Curtis (2020), sobre como nomear tais variáveis agnósticas.

Quadro 10 – Lista de opções para cada nível de nomenclatura dos Design Tokens, parte 1

| Sistema | Conceito | Componente | Grupo | Categoria | Elemento |
|-----------------|-----------|----------------------|------------|------------|----------|
| Nome do Sistema | core | nome do componente 1 | body | size | icon |
| | semantic | nome do componente 2 | heading | space | |
| | component | nome do componente n | typography | color | |
| | | | brand | border | |
| | | | | shadow | |
| | | | | dimension | |
| | | | | opacity | |
| | | | | text | |
| | | | | font | |
| | | | | decoration | |

Fonte: Autoria própria, (2025)

Quadro 11 – Lista de opções para cada nível de nomenclatura dos Design Tokens, parte 2

| Propriedade | Variante | Estado | Escala |
|-------------|-----------|----------|--------|
| family | pill | default | xxs |
| weight | primary | hover | xs |
| line-height | secondary | active | s |
| radius | cloudy | press | sm |
| width | contrast | focus | m |
| padding | | disabled | ml |
| gap | | loading | l |
| bg | | success | xl |
| border | | warning | xxl |
| | | error | xxxl |

Fonte: Autoria própria, (2025)

Quadro 12 – Exemplo prático da composição da nomenclatura dos Design Tokens

| | Sistema | Conceito | Componente | Grupo | Categoria | Propriedade | Variante | Estado | Escala |
|-----------------|---------|----------|------------|-------|------------|-------------|----------|--------|--------|
| Valor Bruto | #00C49A | | | | | | | | |
| Tipo Primitivo | LabDS | core | color | red | 50 | | | | |
| Tipo Semântico | LabDS | semantic | selectable | color | secondary | default | | | |
| Tipo Componente | LabDS | comp | button | color | background | secondary | default | | |

Fonte: Autoria própria, (2025)

O Quadro acima mostra um exemplo de como pode ser utilizada a nomenclatura para os Design Tokens que irão compor o Design System, e em

específico, como podemos referenciar a decisão de qual cor estará vinculada ao fundo do componente botão. Dessa forma, temos o valor bruto em hexa decimal #00C49A o qual é armazenado no Design Token \$LabDS.core.color.red.50, essa primeira variável está ainda em um tipo chamado primitivo, não tendo ainda um significado da sua aplicação, para deixar mais explícito onde essa variável é utilizada, criamos outra variável chamada \$LabDS.semantic.selectable.color.secondary.default e vinculamos a anterior no valor desta, chegando assim no tipo semântico. Com mais entendimento de onde pode ser aplicada tal cor, podemos ainda evoluir esse pensamento para um último tipo, chamado de componente, onde novamente aplicamos a variável anterior como valor da nova variável chamada \$LabDS.comp.button.color.background.secondary.default, dessa maneira criamos uma série de Design Tokens aninhados com o valor bruto inicial, facilitando assim no futuro uma possível mudança de decisão, pois se caso queira mudar a cor de fundo do componente botão, basta mudar a referência ao valor bruto.

O principal está no nome dado a cada variável, para isso fizemos uma categorização para como definir tais nomes através de níveis e a ordem em que estariam dispostos no nome das variáveis. O primeiro nível que inicia o nome da variável é o *Sistema*, representa uma identificação de que esta variável pertence ao Design System, levando assim o nome do próprio. O segundo nível está relacionado ao *Conceito* identifica conceitos em volta de uma decisão de design, como cores de ação, feedback e afins. O terceiro nível é o de *Componente*, evidencia se o design token faz parte de algum componente específico.

O quarto nível, *Grupo*, mostra se a variável está vinculada à um grupo específico que compartilha a mesma decisão. O quinto nível, *Categoria*, está relacionado às decisões de design que podem vir a ser tomadas para se definir o visual de um componente, como cor, texto e espaçamento. O sexto nível, *Propriedade*, identifica a propriedade relacionada à categoria, como, tamanho do texto, cor do texto e assim por diante. Os três últimos níveis se referem às mudanças que podem ter em cada tipo de variável, no caso o nível sete, *Variante*, reflete uma variação do componente ou categoria, já o penúltimo nível, *Estado*, está relacionado a qual variação de estado de interação, como hover, pressionado,

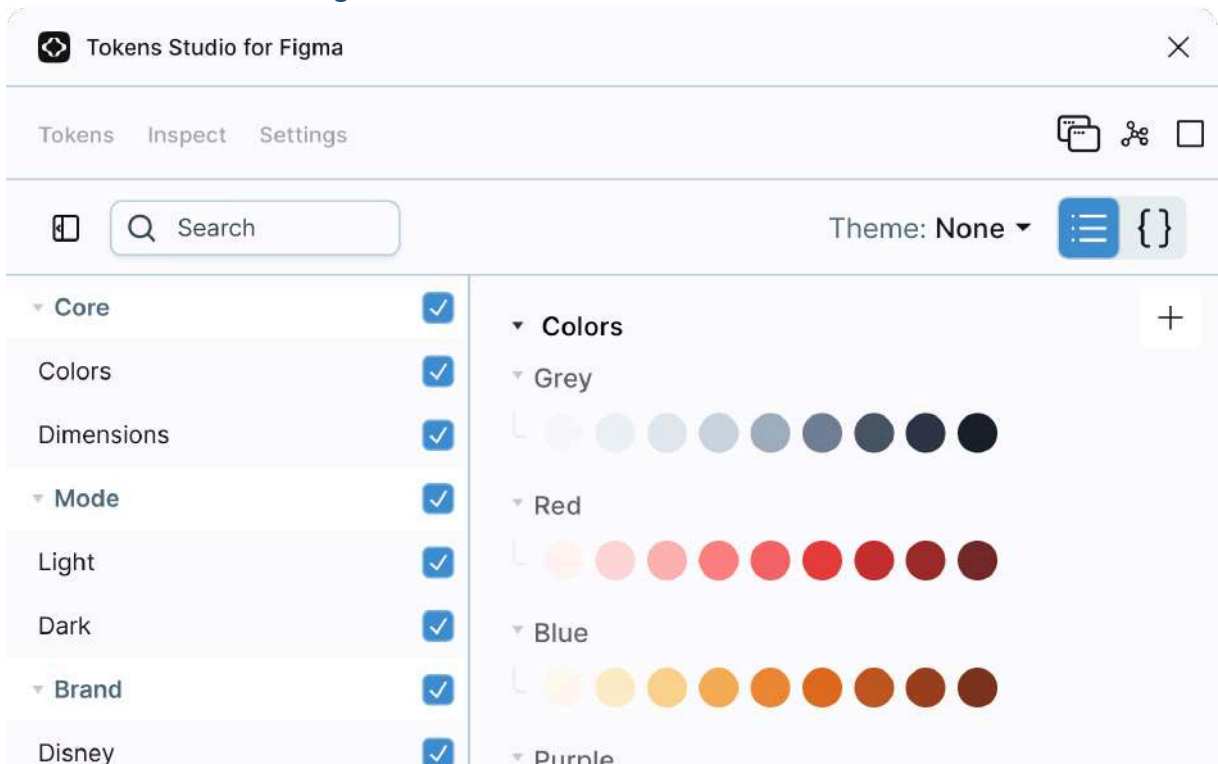
normal, e por último o nível de *Escala*, responsável pela identificação de qual escala a variável representa, usado em cenários de espaçamento e tamanhos.

Com os níveis de nomenclatura definidos e acordados entre Design e Engenharia, a próxima etapa de execução para a estruturação do Design System foi a definição de como os Design Tokens definidos para o sistema seria utilizado em engenharia e conectado com Design. Na próxima seção haverá os detalhes de como foi definida essa arquitetura

5.2.2 CONEXÃO DA LINGUAGEM ÚNICA

A estruturação do uso dos Design Tokens no contexto de Design foi pensada em torno da ferramenta Figma, utilizada pelas pessoas usuárias do Design System no projeto de construção dos protótipos de interface gráfica. Para que estas pessoas conseguissem utilizar os design tokens no dia-a-dia e os componentes da lista *Core* fossem construídos com essas mesmas variáveis, foi tomado a decisão de usar a ferramenta Tokens Studio, disponibilizada para uso interno no Figma, funcionando como um plugin. Veja abaixo na Figura 35, como é esse plugin.

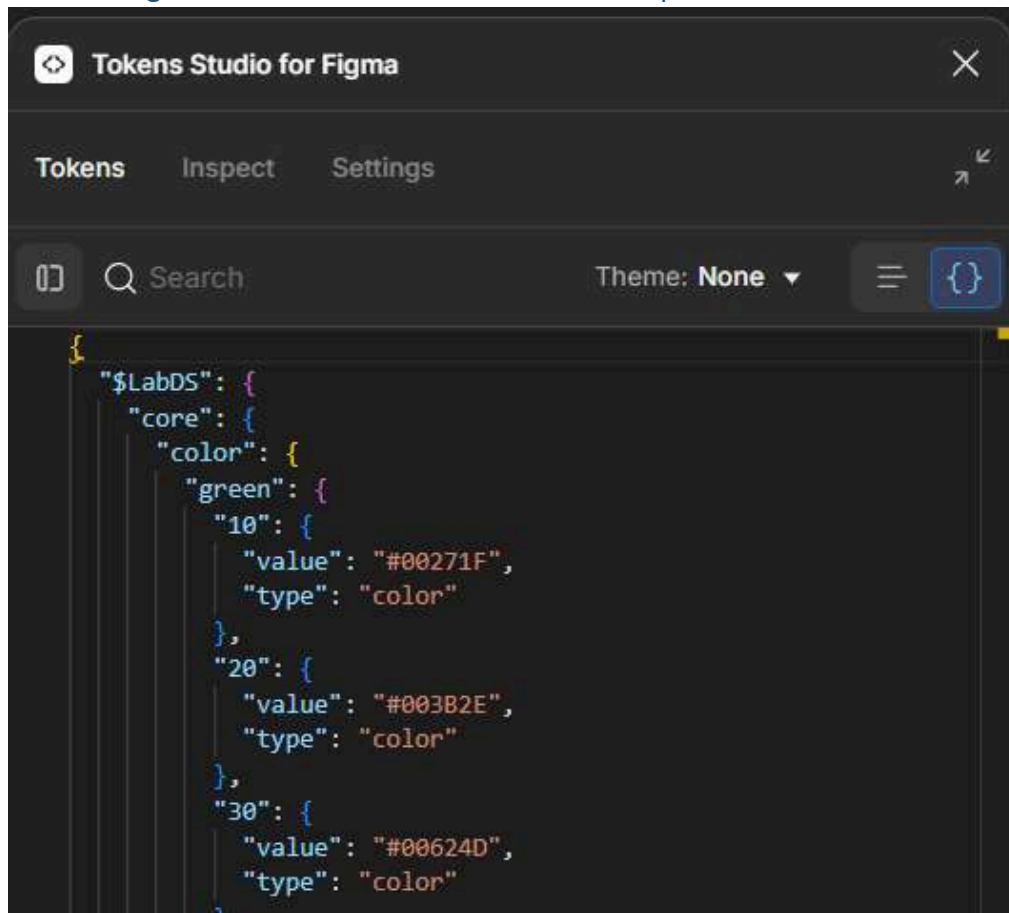
Figura 35 – Interface Gráfica do Tokens Studio



Fonte: tokens studio, (2025)

Dessa forma, será possível organizar todos os design tokens em uma estrutura de par chave-valor, onde a chave é o nome da variável e o valor é o conteúdo que essa variável armazena, podendo ser outra variável ou um valor bruto. Essa estrutura é chamada de JSON (*JavaScript Object Notation*) e possibilitará a conexão com a equipe de engenharia, pois se trata de um formato leve de troca de dados, sendo um formato de texto independente de linguagem de programação e que usa de convenções que são familiares a várias linguagens. (JSON.ORG, 2002). Abaixo na Figura 35 podemos observar como fica a estrutura dos design tokens dentro da própria ferramenta.

Figura 36 – Estrutura de JSON construída pelo Tokens Studio



```
Tokens Studio for Figma
Tokens Inspect Settings
Search Theme: None
{"$LabDS": {
  "core": {
    "color": {
      "green": {
        "10": {
          "value": "#00271F",
          "type": "color"
        },
        "20": {
          "value": "#003B2E",
          "type": "color"
        },
        "30": {
          "value": "#00624D",
          "type": "color"
        }
      }
    }
  }
}
```

Fonte: Autoria própria, (2025)

Tendo então a forma de uso desta ferramenta para centralizar o gerenciamento dos design tokens, coube então a estruturação de como essas variáveis seriam utilizadas pelas pessoas engenheiras usuárias do Design System, para que isso fosse possível é preciso entender que cada linguagem ou framework para desenvolvimento possui o seu formato próprio de declaração de variável, sendo assim para que os design tokens fossem consumidos independente da tecnologia, estruturamos uma forma de converter os design tokens para qualquer uma dessas linguagens através do Style Dictionary, este é um sistema de construção e estruturação dos transformadores de design tokens para cada tipo de plataforma, podendo ser executado em NodeJS ou nativamente em navegadores. Ao exportar os design tokens no formato JSON, o mesmo pôde ser adicionado na estrutura de construção do Style Dictionary e executando assim o comando para a transformação do arquivo em variáveis para cada tipo de plataforma.

No exemplo a seguir mostrará o passo a passo para a conversão de um arquivo JSON com design tokens para o formato de variável no css. Na Figura 36, podemos observar o conteúdo do arquivo de tokens.

Figura 37 – Conteúdo do arquivo JSON de exemplo

```
1  {
2    "colors": {
3      "red": {
4        "value": "#ff0000",
5        "type": "color"
6      }
7    }
8  }
```

Fonte: Style Dictionary, (2025)

Com o arquivo em mãos, podemos configurar o Style Dictionary para qual plataforma queremos transformar esse conteúdo, veja abaixo um exemplo.

Figura 38 – Definindo configuração de transformação

```
1  {
2    "platforms": {
3      "css": {
4        "transformGroup": "css",
5        "files": [
6          {
7            "destination": "vars.css",
8            "format": "css/variables"
9          }
10       ]
11     }
12   }
13 }
```

Fonte: Style Dictionary, (2025)

Para esse exemplo, foi escolhida a plataforma Web, utilizando css puro. Tendo a configuração necessária, podemos então importar o construtor e executar o script de transformação do conteúdo para css.

Figura 39 – Importando construtor

```
1 import StyleDictionary from 'style-dictionary'; 💡
```

Fonte: Style Dictionary, (2025)

Figura 40 – Arquivo com o conteúdo inicial convertido

```
1 /**
2  * Do not edit directly, this file was auto-generated.
3  */
4
5 :root {
6   --colors-red: ■ #ff0000;
7 }
8
```

Fonte: Style Dictionary, (2025)

Nas imagens abaixo é possível observar como os Design Tokens podem ser transformados para outras plataformas, como flutter, android, ios e reac app.

Figura 41 – Definindo configuração de transformação para flutter

```

"platforms": {
  "flutter": {
    "transformGroup": "flutter",
    "buildPath": "../lib/unique_file/",
    "files": [
      {
        "destination": "style_dictionary.dart",
        "format": "flutter/class.dart",
        "options": {
          "className": "StyleDictionary"
        }
      }
    ]
  }
},

```

Fonte: Style Dictionary, (2025)

Figura 42 – Definindo configuração de transformação para android e ios

```

android: {
  transformGroup: 'android',
  buildPath: `build/android/${brand}/`,
  files: [
    {
      destination: 'tokens.colors.xml',
      format: androidColors,
    },
    {
      destination: 'tokens.dimens.xml',
      format: androidDimens,
    },
    {
      destination: 'tokens.font_dimens.xml',
      format: androidFontDimens,
    },
  ],
},
ios: {
  transformGroup: 'ios',
  buildPath: `build/ios/${brand}/`,
  files: [
    {
      destination: 'tokens.h',
      format: iosMacros,
    },
  ],
},

```

Fonte: Style Dictionary, (2025)

Figura 43 – Definindo configuração de transformação para react app

```

"platforms": {
  "scss": {
    "transformGroup": "scss",
    "buildPath": "./src/style-dictionary-dist/",
    "prefix": "token",
    "files": [
      {
        "destination": "variables.scss",
        "format": "scss/variables"
      }
    ]
  }
},

```

Fonte: Style Dictionary, (2025)

Desta forma, com o arquivo JSON com todos os design tokens definidos na ferramenta tokens studio, conseguimos transformar para cada plataforma ou framework de código, ou seja, se o software interativo que está sendo construído, a interface gráfica é feita através do framework react, podemos converter as variáveis para este formato e disponibilizá-las às pessoas engenheiras, a Figura 43 mostra como é esse processo de transformação.

Figura 44 – Processo de transformação dos Design Tokens em variáveis específicas



Fonte: Autoria própria, (2025)

Com essa forma de distribuição dos design tokens definida, a próxima etapa mostrará como foi o processo de construção dos componentes Core, tanto para Design, quanto para Engenharia.

5.2.3 CONSTRUINDO COMPONENTES

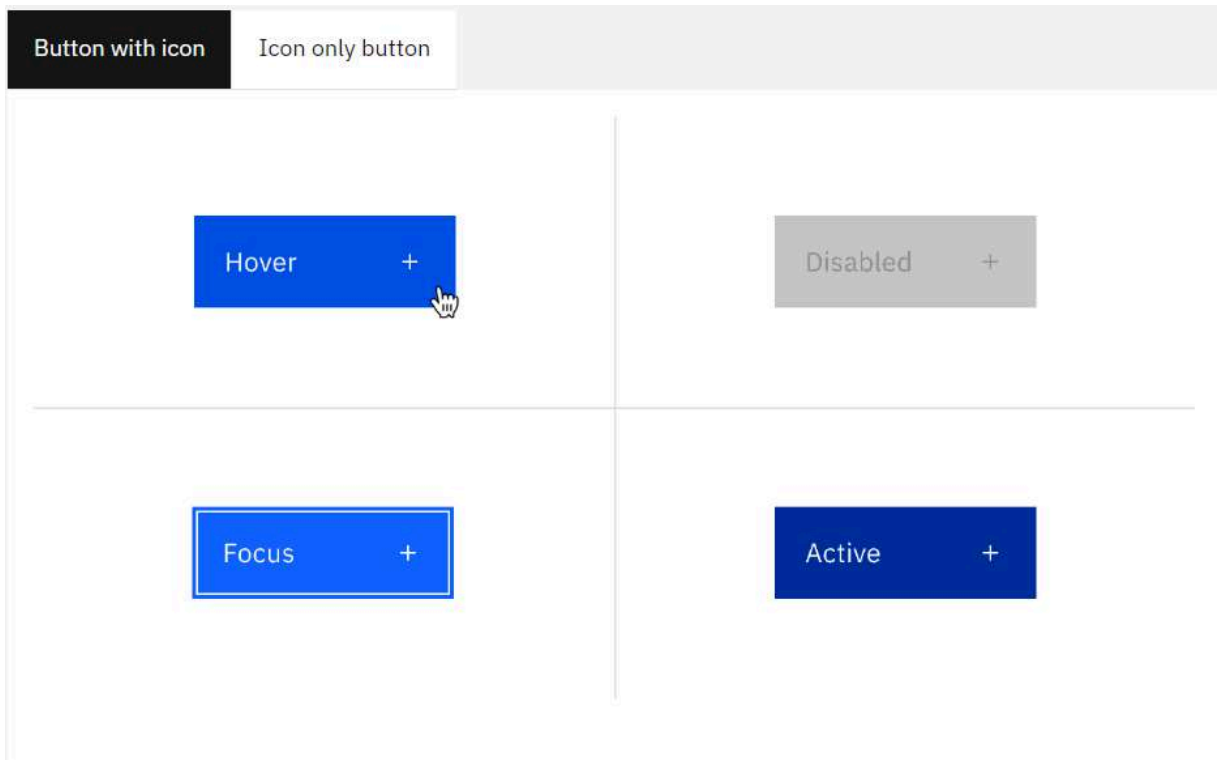
Ao definir todos os design tokens necessários para a estruturação do Design System e a forma como essas decisões estarão conectadas com engenharia, foi iniciada a etapa de construção dos componentes, em um primeiro momento feitos na ferramenta de prototipação, o Figma, e posteriormente em código através da estruturação de Web Components, a qual permite a criação de novas tags HTML customizadas e reutilizáveis, fazendo referência aos componentes do Design System, além de possibilitar o empacotamento para outros frameworks e assim ser utilizado em mais de um tipo de tecnologia e plataforma. O fato da construção acontecer primeiramente na ferramenta de prototipação possibilita que o componente seja testado e desenhado antes da sua construção em código, havendo assim uma otimização de tempo. Nos próximos dois tópicos será apresentado como ocorreu a construção em ambos os contextos.

1. CONTEXTO DE DESIGN

Para exemplificar o processo de construção dos componentes na ferramenta de prototipação, Figma, será mostrado as etapas até obter a versão final do componente, aplicando todas as decisões de design, considerando assim o exemplo do componente Button. Ao decorrer da exemplificação será mostrado também imagens que ilustram tal processo.

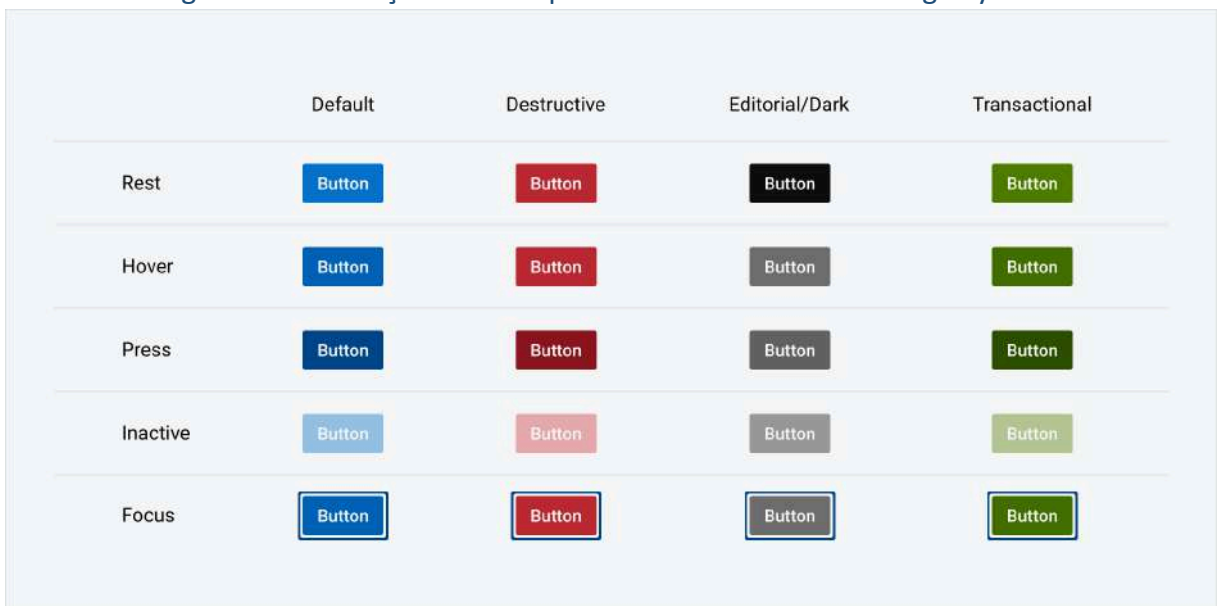
A primeira etapa deste processo é composto por analisar o que o mercado tem definido como necessário para que o componente Button tenha o aspecto de ser um elemento interativo e que guie o usuário durante a jornada, dessa forma foi utilizado como referência o Design System da empresa Dell e o Carbon Design System, mencionado anteriormente. Através da documentação pública disponibilizada por tais empresas, foi possível analisar esses pontos e comparar com as necessidades locais para o projeto em questão.

Figura 45 – Interações no componente Button em Carbon Design System



Fonte: Carbon Design System, (2025)

Figura 46 – Interações no componente Button em Dell Design System

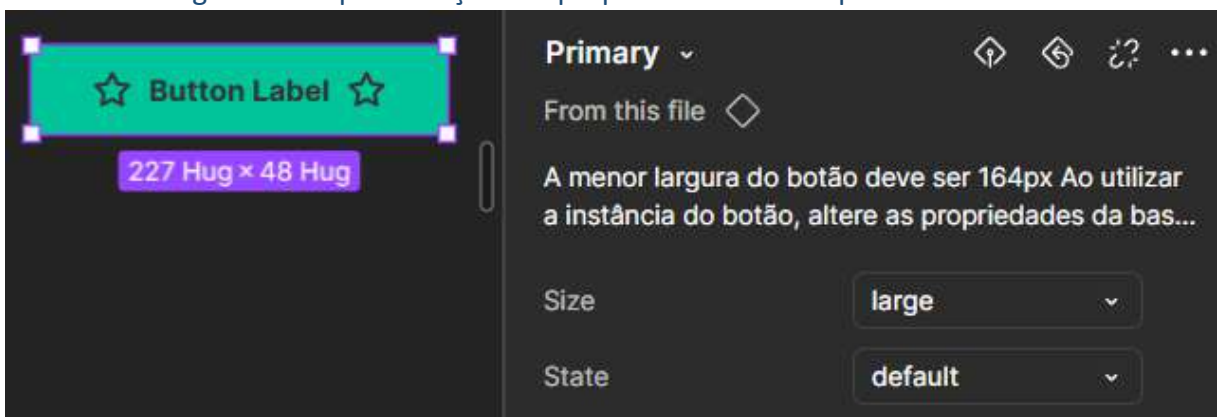


Fonte: Dell Design System, (2025)

Além das interações, é comum encontrar variações de tamanhos e tipos de contexto em que o componente pode ser aplicado, todas essas características

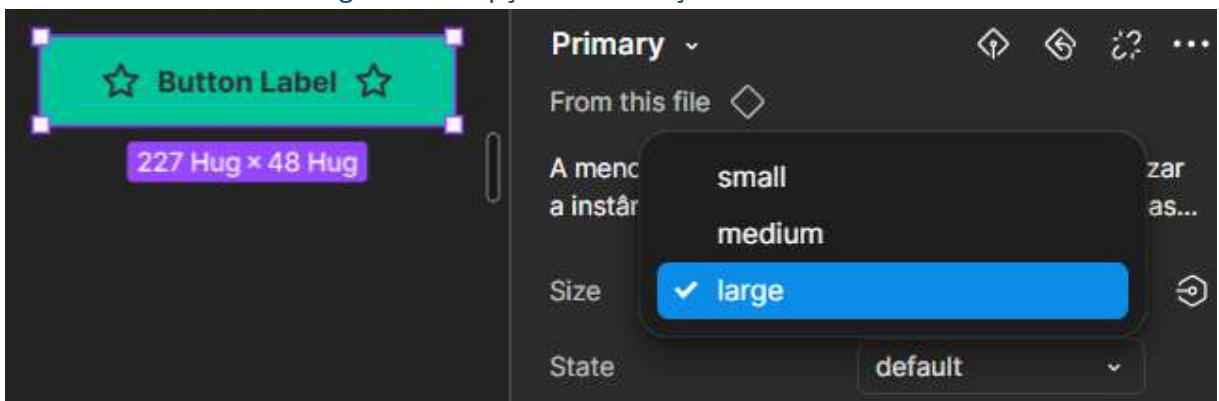
podem culminar em propriedades deste componente, além dos ícones dispostos à esquerda ou direita do texto. Desta forma, após a análise de mercado, foi definido quais seriam essas propriedades do componente Button, sendo elas: Tipo; Tamanho e Estado Interativo, essas propriedades guiam a forma que o componente é prototipado na ferramenta Figma, podendo definir as mesmas propriedades para dentro da plataforma. Veja nas próximas Figuras como é apresentado essas propriedades.

Figura 47 – Apresentação das propriedades do Componente Button



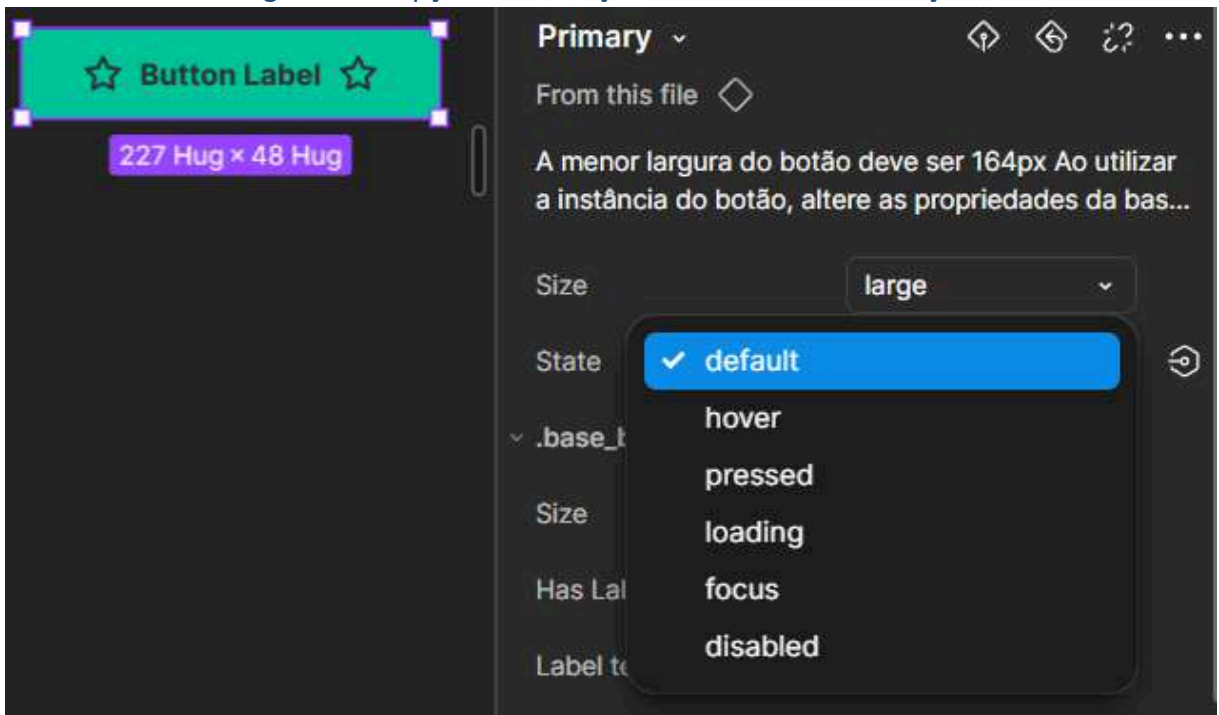
Fonte: Autoria própria, (2025)

Figura 48 – Opções de variações de tamanho



Fonte: Autoria própria, (2025)

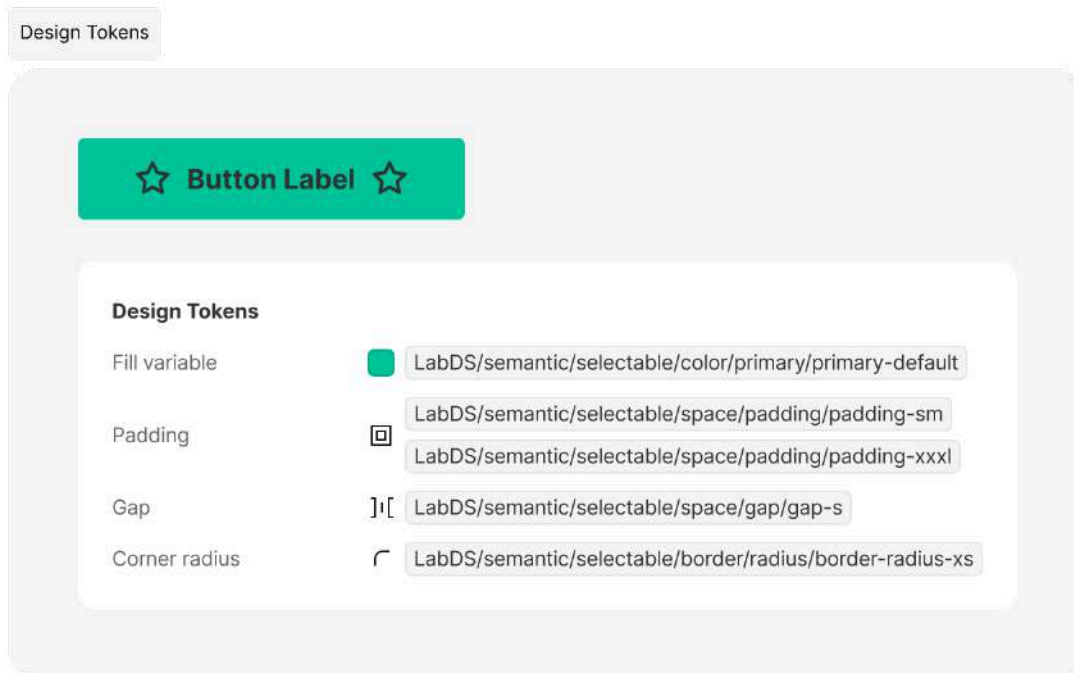
Figura 49 – Opções de variações de estados de interação



Fonte: Autoria própria, (2025)

Através do Figma é possível definir o nome da propriedade e as variações para cada, dessa forma ao utilizar o componente você pode selecionar a combinação específica para o seu contexto, além disso, ao construir os componentes é possível vincular os Design Tokens nas propriedades visuais do componente, como os espaçamentos, cores e arredondamento de bordas, propriedades usadas para definir o visual do componente button. Veja a figura abaixo com tais elementos.

Figura 50 – Representação dos Design Tokens utilizados no componente Button



Fonte: Autoria própria, (2025)

E é nesta etapa de construção dos componentes com os design tokens que avaliamos questões de acessibilidade, quanto ao contraste de elementos baseado no critério 1.4.3 e ao tamanho do alvo, baseado no critério 2.5.5 para que alvos clicáveis ou interagidos por toque devem ter 44x44 pixels de área, baseado na WCAG. Com todos esses processos analisados, temos o componente construído na ferramenta de design, na Figura abaixo é mostrado todas as variações do componente.

Figura 51 – Recorte das propriedades do componente Button



Fonte: Autoria própria, (2025)

Além da construção do componente, é necessário fazer a documentação sobre todas essas decisões tomadas para que o componente tenha essas características, essa documentação irá servir de base para o processo de

construção em código, para isso foi definido que seria documentado a parte anatômica do componente, como os seus espaçamentos, e elementos internos, como texto e ícones, além disso, estariam evidenciados nesta documentação todas as variações de tamanho e de interações previstas para tal componente, todas estas propriedades com os respectivos design tokens vinculados. Nas figuras abaixo há recortes que mostram os detalhes da anatomia e das variações de estado no componente.

Figura 52 – Recorte anatomia documentado para o componente Button

Anatomy

☆ Button Label ☆

Button / Small

- 1 content**

padding-top: `$LabDS.semantic.selectable.space.padding.s`

padding-bottom: `$LabDS.semantic.selectable.space.padding.s`

padding-left: `$LabDS.semantic.selectable.space.padding.x1`

padding-right: `$LabDS.semantic.selectable.space.padding.x1`

border-radius: `$LabDS.semantic.selectable.border.radius.xs`

gap: `$LabDS.semantic.selectable.space.gap.s`
- 2 Label text**

typography: `$LabDS.component.button.typography.none-decoraton.small`

font family: `$LabDS.component.button.font.family`

font size: `$LabDS.component.button.font.size.small`

font weight: `$LabDS.component.button.font.weight`

font line height: `$LabDS.component.button.font.line-height.small`

font letter case: `$LabDS.Core.Font.Letter-Case.02`

font-color: `$LabDS.semantic.color.bg.default`
- Component | Placeholder Icon / Not selectable

3 leading icon

size: medium

element: icon

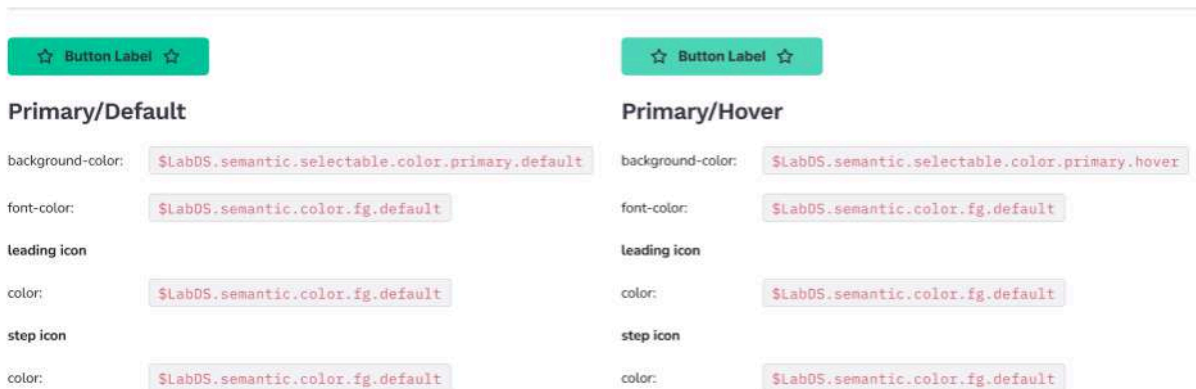
4 step icon

size: medium

element: icon

Fonte: Autoria própria, (2025)

Figura 53 – Recorte Estados de Interação documentados para o componente Button Primary



Fonte: Autoria própria, (2025)

Esse processo de construção se repetiu para todos os componentes da lista *Core*, havendo todos os componentes prontos com a documentação, a listagem já estava pronta para a construção na biblioteca de código em engenharia. Na próxima seção estarão detalhados os processos que envolvem tal contexto.

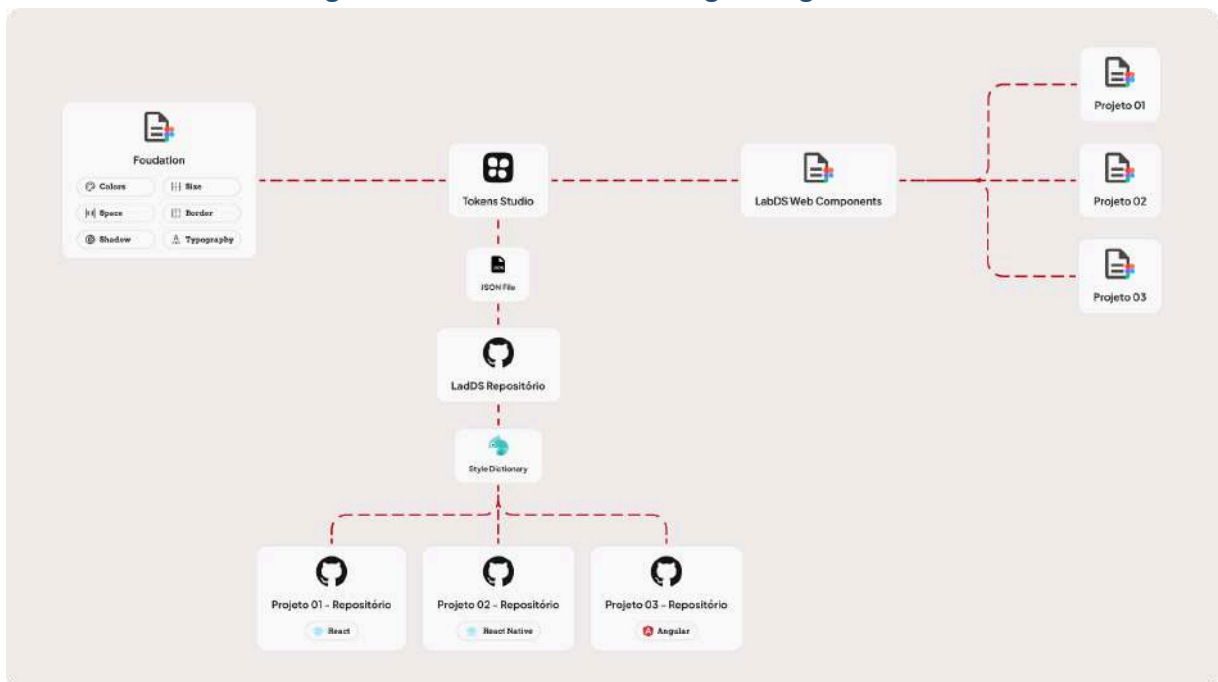
2. CONTEXTO DE ENGENHARIA

Com todos os componentes da lista *Core* prototipados na camada de design e documentados, a pessoa responsável pela construção do Design System na camada de engenharia pôde dar início ao processo de estruturação do DS em código, com a biblioteca de componentes, o repositório para distribuição e versionamento da biblioteca e os pacotes de instalação para uso nos projetos de desenvolvimento do software interativo. O primeiro passo coube conectar as mesmas decisões de design, ou seja, design tokens, para serem utilizados na construção da biblioteca em código, seguindo o processo já mostrado anteriormente na seção 5.2.2, onde através da ferramenta Tokens Studio é possível gerar um arquivo JSON com todos os design tokens e através desse arquivo, transformar os objetos em variáveis específicas para cada plataforma.

Para que essa conexão fosse feita entre os projetos de construção do sistema em desenvolvimento na empresa de marketing digital, foi necessário a criação de um repositório de versionamento e distribuição para a biblioteca de componentes, no

formato Web Components, e o arquivo com os Design Tokens no formato JSON. Dessa forma cada projeto que cuidava de uma parte do sistema, em seus respectivos repositórios, consumiam este repositório da biblioteca com a sua respectiva versão transformada para a plataforma específica utilizada no projeto, através do *Style Dictionary*, responsável pela a transformação dos Design Tokens, e o *transformer* de componentes, diferente para cada tipo de tecnologia específica. Sendo assim, garantimos que todas as equipes que construíam uma parte do software em desenvolvimento pudessem utilizar o Design System, a Figura 53 exemplifica como se deu essa conexão entre Design e Engenharia.

Figura 54 – Conexão entre Design e Engenharia



Fonte: Autoria própria, (2025)

Como é possível observar, na camada de Design, temos o Tokens Studio fazendo a distribuição do arquivo JSON, o qual é conectado ao repositório da biblioteca do Design System, fazendo assim o processo de distribuição para todos os projetos específicos na camada de engenharia.

Havendo assim definida essa conexão, seguimos com a construção de cada componente da lista *Core*, após todos os componentes estarem construídos em código e conectado com as decisões de design, a solução proposta para o problema

desta pesquisa estava pronta para ser utilizada e avaliada. As próximas seções irão abordar tal processo, mostrando como foi o processo de inserir a intervenção no formato de construção do sistema interativo e os impactos gerados.

5.3 USO E AVALIAÇÃO DO DESIGN SYSTEM

A construção da solução para o problema levantado nesta pesquisa aconteceu em períodos diferentes para cada área envolvida, design e engenharia. Da mesma forma aconteceu com o processo de introdução da intervenção para essas áreas, sendo que a equipe de design teve o primeiro contato com o Design System, onde houve uma etapa de treinamento e familiarização com o novo artefato sendo utilizado no processo de construção dos protótipos das interfaces gráficas projetadas para o sistema interativo, com as pessoas usuárias do Design System introduzidas ao novo contexto, a solução foi utilizada por 3 mês e após esse período houve a avaliação qualitativa na perspectiva destas pessoas.

Enquanto a equipe de Design avaliava a solução, a equipe de engenharia era exposta pela primeira vez ao Design System, seguindo a mesma orientação realizada com a área de Design, primeiramente um treinamento e depois a internalização do artefato no processo de construção das interfaces gráficas, seguida da avaliação referente ao uso no período dos 3 primeiros meses. As próximas seções irão mostrar como ocorreu cada contexto trazendo uma avaliação do Design System.

5.3.1 CONTEXTO DE DESIGN

1. INTRODUÇÃO À INTERVENÇÃO

Para que a intervenção fosse introduzida no contexto de construção dos protótipos de interface gráfica feita pela equipe de design, foi realizado um treinamento sobre como utilizar cada componente da lista *Core* do Design System em uma sessão de demonstração e resolução de dúvidas. Além disso, foram

também evidenciados novos processos para que tais pessoas pudessem buscar ajuda e solicitar novas necessidades relacionadas ao contexto do Design System.

Com o treinamento realizado, deu assim início a habilitação do uso da biblioteca de componentes *Core* para a construção dos protótipos para novas funções do sistema em desenvolvimento para a empresa de marketing. A partir desse momento o uso do Design System se daria constante por todo o tempo do projeto do sistema, dessa forma foi quebrado o foco do período de análise por trimestres, e assim para esta pesquisa o primeiro trimestre de uso da equipe de design foi o utilizado para realizar a avaliação da solução.

Durante esse período, cada pessoa utilizadora do DS tinha total liberdade de usar os componentes disponíveis conforme a sua necessidade e além disso, poderia solicitar ajudas e novos itens não mapeados anteriormente, os quais só seriam construídos após o primeiro trimestre. Ao final desse momento, foi realizada a avaliação da solução, o Design System, e quais os impactos trazidos por esse novo artefato. A próxima seção irá mostrar como foi feita essa avaliação e os resultados na visão da equipe de Design, perante ao novo contexto.

2. AVALIAÇÃO DA INTERVENÇÃO

A avaliação foi feita através da metodologia de entrevistas semiestruturada com três pessoas da equipe de Design e o objetivo de comparar se as dificuldades que existiam no contexto anterior foram sanadas, e se surgiram novas dificuldades, em relação ao resultado das entrevistas semiestruturadas feitas no início do projeto. Sendo assim, para que seguisse o mesmo rigor avaliativo dos resultados das entrevistas realizadas na etapa de diagnóstico, recorreremos novamente ao método de Bardin (2016) para analisar o conteúdo das entrevistas após o uso da intervenção pela equipe de Design. Chegamos então a seguinte análise:

a. Fase 1 – Pré-análise

Quadro 13 – Etapas do processo

| Etapa | Procedimentos | Aplicação ao estudo |
|---|--|--|
| Escolha dos documentos | Definir o corpus pertinente e preparar o material | Selecionou-se a seção Resultados (relatos dos designers) como corpus principal. O planejamento, hipóteses e perguntas serviram de referencial analítico para guiar a leitura. |
| 1.2 Leitura flutuante | Leitura exploratória para captar impressões gerais | Identificaram-se cinco grandes eixos tematizados espontaneamente pelos respondentes: Uso dos componentes, Documentação, Processos, Tokens e Aspectos gerais (pontos positivos / melhorias). |
| 1.3 Formulação das hipóteses e objetivos de codificação | Delimitar questões norteadoras | Questão-guia: “Em que medida a nova biblioteca atende às necessidades do time de design em termos de usabilidade, documentação e processos?” Hipóteses operacionais derivadas do planejamento (H1-H7) foram mantidas como categorias apriorísticas. |
| 1.4 Elaboração de indicadores | Definir unidades de registro e context | Unidade de registro: enunciado semântico (frase ou sintagma que exprima opinião, dificuldade, benefício etc.). |

| | |
|--|---|
| | Unidade de contexto: o bloco temático (positivo, dificuldade, necessidade etc.) onde o enunciado aparece. |
|--|---|

Fonte: Dados da pesquisa, (2025)

foram definidas as hipóteses para guiar as perguntas desta pesquisa, as quais estão expostas no quadro abaixo:

Quadro 14 – Hipóteses para validação

| | |
|-------------|---|
| Hipótese 01 | Acredito que o Design System facilitou o uso dos componentes, saberei se estou certo se fizer uma entrevista semiestruturada com o time de design e descobrir que cerca de mais da metade dos entrevistados evidenciaram pontos de facilidade ao usar os componentes. |
| Hipótese 02 | Acredito que a documentação disponibilizada, trouxe uma melhor compreensão de uso dos componentes, saberei se estou certo se fizer uma entrevista semi estruturada com o time de design e descobrir que cerca de mais da metade dos entrevistados ao estarem com dificuldade em usar certo componente, buscou auxílio na documentação mencionada. |
| Hipótese 03 | Acredito que a forma como está organizado as nomenclaturas dos componentes, facilitou a localização no momento de utilizá-los, saberei se estou certo se fizer uma entrevista semiestruturada com o time de design e descobrir que cerca de mais da metade dos entrevistados ao serem expostos no contexto de estarem procurando os componentes não tiveram dificuldades. |
| Hipótese 04 | Acredito que com a reconstrução e disposição das propriedades dos componentes logo na camada superior, deixou mais simples o uso dos componentes nos contextos devidos, saberei se estou certo se fizer uma entrevista semiestruturada com o time de design e descobrir que cerca de |

| | |
|-------------|--|
| | mais da metade dos entrevistados ao usarem os componentes, (Tabela, Dropdown e Textfield) não sentiram dificuldades em definir suas propriedades para o contexto em específico. |
| Hipótese 05 | Acredito que mesmo com a nova biblioteca, houveram muitos casos de desvinculação das instâncias dos componentes, saberei se estou certo se fizer uma entrevista semiestruturada com o time de design e descobrir que cerca de mais da metade dos entrevistados trouxeram mais de 3 cenários em que efetuou o desvinculação. |
| Hipótese 06 | Acredito que com a estratégia de design tokens utilizada, trouxemos um nível de complexidade que o time de design ainda possui dificuldades e ainda não conseguiram usar assertivamente durante o dia a dia saberei se estou certo se fizer uma entrevista semiestruturada com o time de design e descobrir que cerca de mais da metade dos entrevistados trouxeram momentos e seus pontos de dúvidas relacionado aos design tokens. |
| Hipótese 07 | Acredito que com os processos adotados de descrição das atualizações, problemas com componentes, necessidade de novos componentes, estão claros para o time, saberei se estou certo se fizer uma entrevista semiestruturada com o time de design e descobrir que todos os entrevistados não possuem dúvidas de como seguir os processos e o que é necessário ser feito. |

Fonte: Dados da pesquisa, (2025)

As hipóteses acima se conectam com pontos importantes levantados na primeira pesquisa realizada, relacionados a documentação e dificuldades no processo de construção das interfaces gráficas. Com isso, foi possível definir as perguntas para o roteiro das entrevistas semiestruturadas, o apêndice C possui todas as perguntas definidas. Indo então para a segunda fase do processo temos os seguintes pontos:

b. Fase 2 – Exploração do material (codificação e categorização)

i. **Codificação aberta**

1. Realizou-se um processo misto:

a. a priori – categorias ligadas às hipóteses (Usabilidade, Documentação, Localização, Propriedades, Desvinculação, Tokens, Processos).

b. a posteriori – subcategorias emergentes (ex.: “responsividade”, “gráficos”, “harmonia visual”).

2. Cada unidade de registro foi marcada com um rótulo inicial (ex.: “facilidade-tokens”, “dificuldade-ícones”, “detach-tabela”).

ii. **Categorização**

1. Estrutura final:

a. Usabilidade dos componentes

i. A1. Facilidade geral

ii. A2. Propriedades (tabela, dropdown, textfield)

iii. A3. Responsividade / tamanhos

iv. A4. Detach (desvinculação)

b. Documentação

i. B1. Consulta bem-sucedida

ii. B2. Dificuldade de localização

iii. B3. Padronização necessária

c. Processos & Comunicação

i. C1. Atualizações (clareza)

ii. C2. Fluxo de reporte / suporte

d. Design Tokens

i. D1. Compreensão conceitual

ii. D2. Uso prático (espaçamento, cor)

e. Aspectos Gerais

i. E1. Pontos positivos macro (rapidez, harmonia, colaboração)

- ii. E2. Pontos de melhoria macro (gráficos, paleta de cor que diferencia áreas da plataforma)
- iii. **Re-codificação e verificação**
 - 1. Para garantir exaustividade e exclusividade, cada enunciado foi revisto e realocado quando necessário, evitando sobreposição de categorias.
- iv. **Quantificação auxiliar (frequência simples)**

Quadro 15 – Frequência de ocorrências de cada tópico

| Categoria | Nº ocorrências (n = 43 unidades) |
|---------------------|----------------------------------|
| a – Usabilidade | 14 |
| b – Documentação | 7 |
| c – Processos | 5 |
| d – Design Tokens | 9 |
| e – Aspectos Gerais | 8 |

Fonte: Dados da pesquisa, (2025)

A frequência (número de ocorrências) serve apenas como indicativo; não há pretensão estatística com N = 3 entrevistadas.

c. Fase 3 – Tratamento dos resultados, inferência e interpretação

i. **Síntese interpretativa por categoria**

- 1. **Usabilidade (A)** – Predominância de relatos positivos: “A nova biblioteca está facilitando muito e ajudando na rapidez de construir as telas”. As dificuldades concentram-se em responsividade e necessidade de tamanhos *small/medium/large*. O *detach* de componentes ainda ocorre (*accordion*, tabela, botões) por requisitos

específicos de tipografia, corroborando a Hipótese 5 (Complexidade de desvinculo).

2. **Documentação (B)** – Reconhecimento do arquivo de documentação como útil, porém persiste “dificuldade em encontrar alguns ícones” e ausência de padronização no formato das páginas de help. A Hipótese 2 (documentação suficiente) foi invalidada, confirmando que melhorias na padronização são cruciais.
3. **Processos (C)** – Entrevistadas consideram as atualizações claras graças ao *Slack*, plataforma de comunicação geral da empresa, validando a Hipótese 7. Contudo, há lacunas no fluxo de solicitação de ajustes.
4. **Design Tokens (D)** – Tema mais sensível; embora tokens sejam vistos como facilitadores de consistência, surgem dúvidas de uso e limitação a “alguns tokens” (ex.: espaçamento). Confirma-se a Hipótese 6 de complexidade residual.
5. **Aspectos Gerais (E)** – Reconhecimento da harmonia visual e do maior leque de variações em comparação à biblioteca anterior. Por outro lado, os componentes de gráfico e a paleta de cores para distinguir áreas da plataforma em desenvolvimento requerem revisão para consistência cromática.

ii. Validação das hipóteses (triangulação com categorias)

Após a execução das entrevistas, apenas a segunda hipótese, relacionada a documentação, foi invalidada. As demais foram validadas, o quadro abaixo apresenta um resumo sobre os tópicos relacionados a cada hipótese, e como as pessoas entrevistadas avaliaram o Design System.

Quadro 16 – Validação das hipóteses

| Hipótese | Categoria(s) relacionada(s) | Status (self-report) | Evidência da análise |
|---------------------------------|-----------------------------|----------------------|---|
| H1 – Facilidade geral | A1 | Validada. | Frequência alta de menções positivas; nenhuma dificuldade grave de uso. |
| H2 – Documentação suficiente | B1-B3 | Invalidada | Ocorrências de “dificuldade em encontrar” e pedidos de padronização. |
| H3 – Localização/nomenclatura | A1, B2 | Validada | Não houve dificuldades significativas em buscar componentes, salvo ícones. |
| H4 – Propriedades simplificadas | A1, B2 | Validada | “Acesso fácil às propriedades”, “sem dificuldades em usar tabela, dropdown, textfield”. |
| H5 – Desvinculo excessivo | A4 | Validada | Relatos de detach em 3 tipos de componente. Ações corretivas propostas. |
| H6 – Complexidade dos tokens | D | Validada | Dúvidas conceituais e uso parcial, indicando barreiras de adoção. |
| H7 – Clareza de processos | C | Validada | Entendimento das atualizações; canais definidos. |

Fonte: Dados da pesquisa, (2025)

iii. Proposições e recomendações

1. Padronizar documentação (categoria B3) com Do's & Don'ts visuais e índice de ícones.

2. Treinamento focado em Design Tokens (categoria D) – três workshops práticos, conforme acionável.
3. Tabela de desvinculação (detach) pública (categoria A4) para rastrear causas e priorizar correções.
4. Versões responsivas dos componentes (categoria A3) priorizando mobile-first.
5. Revisão dos componentes de gráfico e paleta de cores que distinguem áreas da plataforma em desenvolvimento para coerência cromática (categoria E2).

d. Fase 4 - Considerações sobre a análise

Como é possível observar, os tópicos que haviam problemas antes da implementação da intervenção, foram mencionados na entrevista de forma positiva e além disso, houve declarações em que as pessoas validaram e enfatizaram que a intervenção melhorou o processo de construção de interfaces. Dessa forma, podemos validar a intervenção para o contexto de design e observar que os requisitos definidos anteriormente para a solução foram cumpridos, seguindo assim a base metodológica de PEFFERS *et al.* 2007. O fato de utilizar categorias mistas (a priori e emergente) permitiu respeitar o planejamento da pesquisa e capturar novas nuances, apesar do pequeno número de entrevistadas impedir generalização quantitativa a análise qualitativa forneceu evidências robustas para decisões de aprimoramento do Design System.

Outro ponto importante captado com a pesquisa de avaliação, foi quanto aos pontos negativos que houveram durante o período avaliado o que nos trás insumos para melhorarmos a solução e indo de encontro com os princípios da Design Science evidenciados por Simon 1996 ao enfatizar que o resultado ótimo não é o objetivo, mas sim um resultado satisfatório para o contexto do problema. (SIMON, 1996). E em adição, nos possibilita uma nova iteração do método para o contexto. (VAN AKEN, *et al.*, 2012).

Tendo assim o contexto de design avaliado positivamente, os pontos de melhoria foram mapeados para uma nova iteração da metodologia para que sejam

adicionados à intervenção e assim tornar cada vez melhor o processo de construção de interfaces. Na próxima seção será observada a avaliação no contexto de engenharia.

5.3.2 CONTEXTO DE ENGENHARIA

1. INTRODUÇÃO À INTERVENÇÃO

O Design System no contexto da equipe de engenharia foi disponibilizado após o período de 3 meses em que foi implementado para a equipe de design, pois dessa forma seria garantido que os protótipos feitos durante esse período estivessem com a solução em sua base, e assim após esse período, estar disponível para a equipe de engenharia construir. Dessa forma, a introdução da intervenção no processo de construção das interfaces gráficas em código aconteceu por meio de treinamentos realizados pelo responsável da construção do DS em código, e assim demonstrando para todos como deveria ser utilizada e incorporada tal solução.

Havendo assim a realização de um treinamento para toda a equipe de engenharia, a biblioteca em código com todos os componentes foram disponibilizados para o uso no processo. O momento de avaliação para a equipe de engenharia foi definido seguindo o período da equipe de design, ou seja, 3 meses, e durante esse período foram feitas algumas iniciativas para mapear quantos porcentos dos incrementos de software estavam utilizando o Design System, uma delas foi analisar no processo de solicitação para novos incrementos de software serem anexados ao software em produção. E durante esse período tivemos 61 incrementos solicitados e introduzidos no software em produção, desses 61, 56 estavam utilizando o Design System, tendo assim uma taxa de 91% de adoção pela equipe de desenvolvimento. Isso mostra também que o Design System chegou na camada dos usuários finais, que utilizam o software.

Além dessa análise feita durante o período de uso da intervenção, foram também realizados momentos de auxílio para as pessoas da equipe, conforme surgiam dúvidas. Tendo então passado o trimestre de uso da solução, realizamos uma entrevista semiestruturada com as pessoas da equipe, a fim de colher as percepções qualitativas quanto à solução. Na próxima seção será apresentado como foi conduzida a pesquisa e os seus resultados.

2. AVALIAÇÃO DA INTERVENÇÃO

Para entendermos o impacto gerado pela solução no processo de construção de interfaces gráficas em código, realizamos sete entrevistas semiestruturadas com as pessoas da equipe de engenharia que utilizaram a proposta durante o período citado na seção anterior, para isso, definimos que o objetivo da pesquisa era comparar se as dificuldades que existiam na biblioteca antiga foram sanadas, e se surgiram novas dificuldades, tendo como base o resultado da pesquisa inicial do projeto. Com esse objetivo como guia, definimos quais hipóteses serviriam de base para levantarmos os questionamentos para a pesquisa, a análise das entrevistas foi realizada utilizando também o método definido por Bardin (2016) a seguir estão descritos tais fases do método:

a. Pré-análise

- i. **Constituição do corpus** – O estudo partiu das sete entrevistas semiestruturadas realizadas com desenvolvedores front-end que consomem a biblioteca do DS em código. O planejamento da pesquisa delimita objetivo, perguntas e cinco hipóteses operacionais (H1-H5).
- ii. **Leitura flutuante** – Uma leitura exploratória das transcrições evidenciou recorrentes menções a usabilidade, problemas técnicos (carregamento, bugs), flexibilidade/personalização, componentes ausentes, processos & suporte e documentação.
- iii. **Objetivos de codificação** – Pergunta-guia: “Em que medida a nova versão da biblioteca satisfaz as necessidades do time de desenvolvimento em termos de uso, documentação e

processos?” Cada hipótese foi mantida como categoria a priori, sem impedir a emergência de temas novos.

- iv. **Unidades de análise** – Unidade de registro: enunciado semântico (frase ou sintagma de opinião/dificuldade). Unidade de contexto: bloco temático (p. ex., “Usabilidade – problemas técnicos”).

Abaixo segue as hipóteses definidas e que guiaram as perguntas da avaliação:

Quadro 17 – Hipóteses para validação

| | |
|-------------|---|
| Hipótese 01 | Acreditamos que a forma atual de uso dos componentes está melhor, comparando o momento anterior em que havia outra biblioteca. Saberei se estamos certos se fizermos uma entrevista semiestruturada com a equipe de desenvolvimento e descobrir que cerca de 70% dos entrevistados trouxeram pontos favoráveis para a nova biblioteca quanto a essa questão. |
| Hipótese 02 | Acreditamos que exista contextos em que o desenvolvedor se deparando com a não possibilidade de realizar alterações nos componentes acabam fazendo adaptações não recomendadas e não comunicando aos responsáveis. Saberemos se estamos certos se fizermos uma entrevista semiestruturada com o time de desenvolvimento e descobrir que cerca de 70% dos entrevistados trouxeram contextos evidenciando esse tipo de acontecimento e descrevendo a forma que lidou. |
| Hipótese 03 | Acreditamos que a documentação atual da biblioteca lab-ds consegue dar apoio para a equipe de engenharia, saberemos se estamos certos se fizermos uma entrevista semiestruturada com o time de desenvolvimento e descobrir que cerca de 70% dos entrevistados trouxeram pontos favoráveis quanto à documentação existente. |
| Hipótese 04 | Acreditamos que os processos que envolvem o uso da biblioteca estão claros para os usuários, saberemos se |

| | |
|-------------|---|
| | estamos certos se fizermos uma entrevista semi estruturada com o time de desenvolvimento e descobrir que cerca de 70% dos entrevistados evidenciaram que estão cientes dos processos existentes. |
| Hipótese 05 | Acreditamos que com a lista atual de componentes, abrangemos todos os componentes básicos mais necessários na visão da equipe de engenharia, saberemos se estamos certos se fizermos uma entrevista semiestruturada com o time de desenvolvimento e descobrir que cerca de 70% dos entrevistados evidenciaram que a lista atual está completa quanto aos componentes básicos. |

Fonte: Dados da pesquisa, (2025)

É possível observar tais perguntas nos apêndice D, onde estão todas as perguntas feitas. Com o roteiro definido, as entrevistas foram realizadas com as 7 pessoas que faziam parte da equipe.

b. Exploração do material (codificação e categorização)

Quadro 18 – Frequência de ocorrências de cada tópico

| Categoria (macro) | Subcategorias emergentes | n (43 unidades) |
|--------------------------|--|------------------------|
| A – Usabilidade | A1 Facilidades gerais; A2 Problemas técnicos; A3 Flexibilidade; A4 Componentes faltantes | 17 |
| B. Processos & Suporte | B1 Engajamento/comunidade; B2 Fluxo de contribuição; B3 Suporte dedicado | 8 |
| C. Documentação | C1 Organização intuitiva; C2 Exemplos de código; C3 Tokens pouco claros | 10 |
| D. Aspectos gerais | D1 Pontos positivos macro; D2 Pontos de melhoria macro | 8 |

Fonte: Dados da pesquisa, (2025)

A codificação híbrida (dedutiva + indutiva) garantiu exclusividade e exaustividade; todas as unidades foram recodificadas até não haver sobreposição.

c. Tratamento, inferência e interpretação

- i. **Usabilidade (A)** – O predomínio é positivo (“a biblioteca nova é bem melhor que a antiga”) , mas há entraves técnicos – lazy loading, ausência de sidebar account e persistência de focos em alguns componentes . A necessidade de personalização adicional no TextField e em outros elementos confirma a preocupação com flexibilidade.
- ii. **Processos & Suporte (B)** – Todos elogiam o suporte “muito bom e profissional”, mas pedem canal dedicado no Slack, plataforma de comunicação geral usada pela a empresa de marketing, automação de problemas e uma “anamnese” de componentes. O engajamento da comunidade ainda é baixo.
- iii. **Documentação (C)** – A estrutura atual é vista como intuitiva, porém faltam exemplos de código e explicações dos design tokens, considerados “difíceis de entender” .
- iv. **Aspectos gerais (D)** – Destacam-se ganhos de produtividade (“tudo que há no Figma tem na biblioteca em código”) e harmonização de propriedades, mas há alertas sobre sobrecarga de um único desenvolvedor e falta de padronização tipográfica.

d. Validação das hipóteses

Quadro 14 – Validação das hipóteses

| Hipótese | Status | Evidência |
|--|----------|-----------------------|
| H1 – Nova biblioteca melhor que antiga | Validada | 100 % relatam melhora |

| | | |
|------------------------------------|------------|--|
| H2 – Adaptações não comunicadas | Invalidada | Apenas parte das pessoas desenvolvedoras relataram adaptações silenciosas |
| H3 – Documentação dá apoio | Validada | Avaliação positiva da organização, apesar de lacunas |
| H4 – Processos claros | Validada | Entrevistados conhecem o fluxo, sugerem apenas refinamentos |
| H5 – Lista de componentes completa | Validada | Mais de 70% dos entrevistados julgaram a cobertura suficiente; pedem poucos itens (drag-n-drop, modal) |

Fonte: Dados da pesquisa, (2025)

E como resultado, apenas a hipótese 2 foi invalidada, mostrando que a solução teve em sua maioria pontos positivos em comparação ao estado anterior sem a solução proposta.

e. Recomendações imediatas

Como foi observado com os resultados das entrevistas, a maioria dos entrevistados mencionaram que a nova biblioteca de componentes trouxe melhorias ao processo, melhorando o estado do contexto em comparação quando não havia a solução proposta. Apesar dos pontos negativos evidenciados, no geral as pessoas estavam satisfeitas com o Design System, além disso, tais pontos são insumos para melhorias futuras, podendo assim realizar mais um processo de iteração da metodologia no contexto de engenharia e enfatizando assim o pensamento de Simon em 1996. Veja abaixo a lista das principais recomendações imediatas:

- i. **Criar canal no Slack**, plataforma de comunicação da empresa, exclusivo para assuntos que envolvam o DS e automatizar a abertura de problemas encontrados.
- ii. **Priorizar correções de lazy loading**, bugs de tabela e estados de foco.

- iii. **Produzir exemplos de código**, mais o guia de design tokens no Storybook.
- iv. **Definir critérios para flexibilização** (propriedades avançadas) de componentes críticos.
- v. **Avaliar alocação de pessoa engenharia** dedicada para acelerar manutenção e suporte.

Com a realização da avaliação e análise dos resultados para os dois principais contextos desta pesquisa, da perspectiva qualitativa, as pessoas utilizadoras do Design System em geral, preferiram o contexto atual com a intervenção aplicada do que o período anterior.

5.3.3 ANÁLISE DE AMBOS OS CONTEXTOS APÓS USO DA INTERVENÇÃO

Durante a etapa de diagnóstico do problema as entrevistas revelaram um cenário de **fragmentação estrutural**: ausência de documentação, nomenclaturas divergentes e processos inexistentes minavam a adoção da biblioteca anterior tanto por designers quanto por desenvolvedores. Esse quadro inicial justificou a implantação do Design System e ofereceu um mapa de prioridades que resultaram na intervenção utilizada pelas duas áreas.

Nos questionários para a equipe de design a primeira versão do DS demonstrou impacto positivo imediato: usabilidade percebida melhorou e processos de atualização tornaram-se visíveis. Todavia, lacunas de documentação – sobretudo busca de ícones e padronização de páginas – permaneceram, indicando que a intervenção inicial privilegiou a entrega de componentes, mas não consolidou completamente os artefatos de referência.

Já para a equipe de engenharia observa-se nova camada de maturidade: a documentação textual evoluiu, porém, emergem demandas típicas do ciclo de adoção em código – exemplos práticos, guia de tokens e solução de bugs de carregamento. A crítica desloca-se do *o que* para o *como usar*, sinais de que a biblioteca atingiu um estágio em que a qualidade da experiência de desenvolvimento determina sua eficácia.

A comparação mostra um **gradiente de melhoria consistente**: dores brutas do diagnóstico convertem-se em ajustes incrementais nas rodadas seguintes. No entanto, dois temas atravessam as três fases: **design tokens** e **governança**. Enquanto os tokens permanecem pouco claros (nomes, aplicação prática), a governança ainda carece de um ponto focal técnico que garanta manutenção ágil.

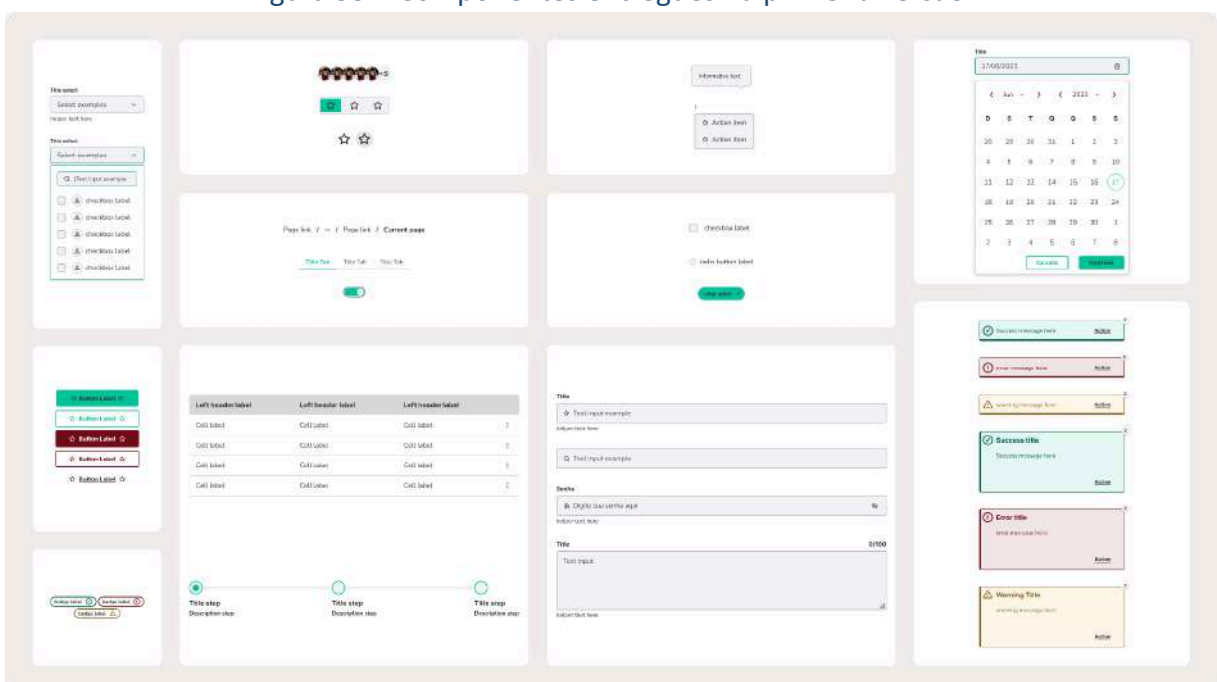
Com essa avaliação dos dois contextos após tal intervenção, recomenda-se que as próximas iterações concentrem-se em (i) finalizar o “manual do desenvolvedor” (tokens + exemplos), (ii) instituir um owner dedicado ao repositório e (iii) fechar o circuito de nomenclatura entre Figma e front-end. Esses passos tendem a consolidar o Design System como plataforma unificadora, reduzindo retrabalho, aumentando a consistência visual e promovendo velocidade de entrega em toda a organização. Na seção de conclusão será apresentado o direcionamento para esses pontos de melhoria e quais desses foram realizados.

6 CONCLUSÃO

Conforme pôde ser observado nesta pesquisa, o contexto de desenvolvimento de softwares interativos em empresas privadas podem acabar negligenciando critérios que fazem parte da natureza da interação humano-máquina, levando assim a inconsistências em quesitos de usabilidade, acessibilidade e *affordance*.

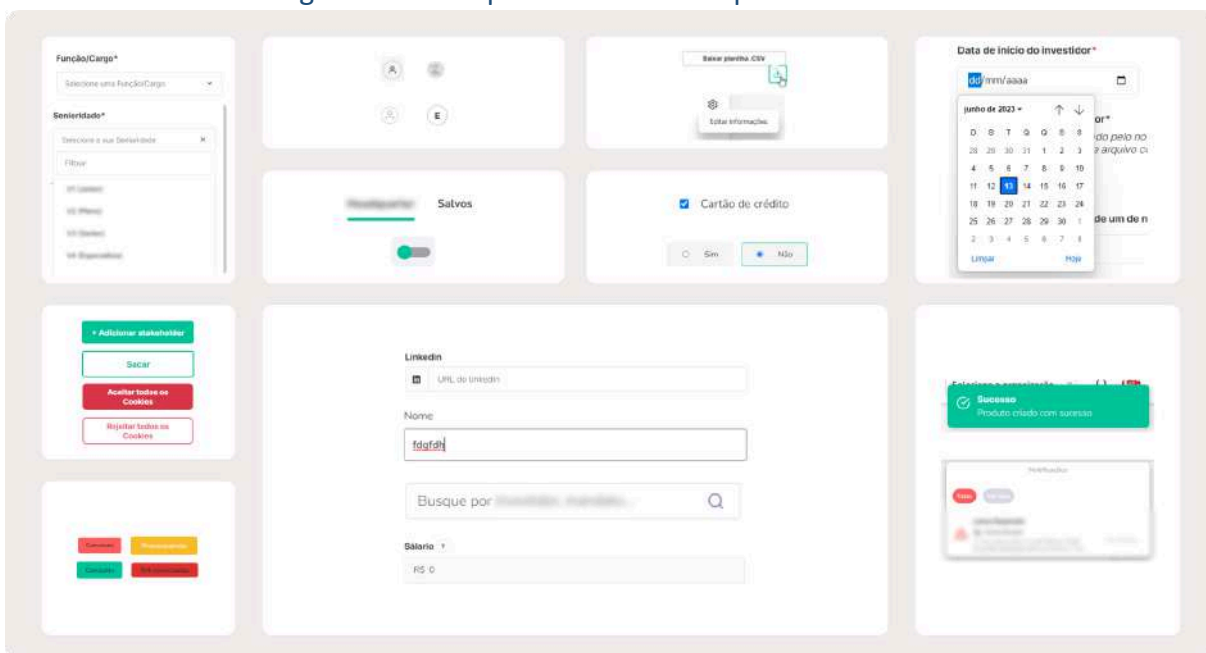
Dado o objetivo central desta pesquisa em descobrir uma forma de manter a consistência visual do sistema interativo nesse contexto de desenvolvimento de software, podemos concluir que a hipótese foi validada, pois com a aplicação da abordagem de design system conseguimos garantir em um contexto real, na empresa brasileira de marketing, a conexão entre design e engenharia, levando consistência para o sistema em desenvolvimento e melhorando o processo de construção de interfaces gráficas, mudando assim o contexto anterior para melhor. A imagem abaixo faz referência sobre o resultado final dos componentes da lista *core* do design system, visando os critérios de usabilidade, acessibilidade e *affordance* e na imagem seguinte um comparativo como era antes.

Figura 55 – Componentes entregues na primeira versão



Fonte: Autoria própria, (2025)

Figura 56 – Componentes antes da primeira versão



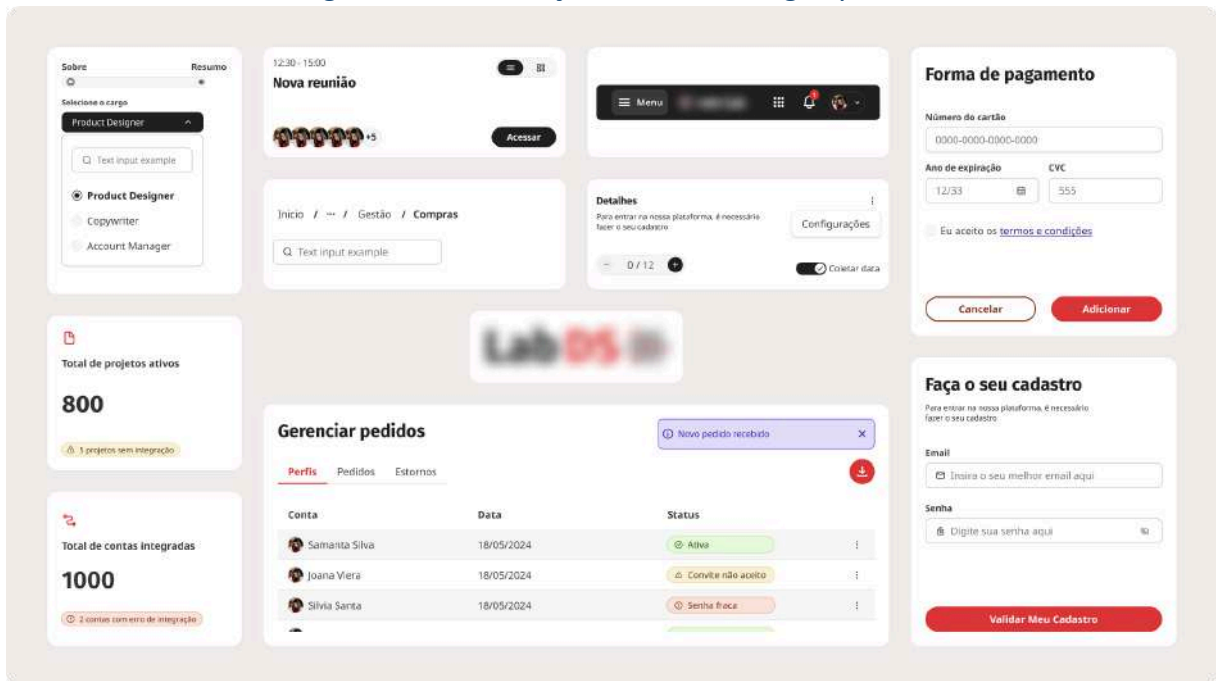
Fonte: Autoria própria, (2025)

E os resultados qualitativos reforçam a validação dessa hipótese, onde as pessoas responsáveis pela construção das interfaces gráficas enfatizaram que a solução tornou o contexto melhor. Além disso, o *Retorno Sobre Capital Investido*, após 9 meses com a solução sendo utilizada foi estimado em 0,99, faltando assim menos de um por cento para atingir o valor total investido pela empresa de marketing no projeto. Tal resultado foi obtido ao calcularmos o ganho de eficiência da equipe de design e desenvolvimento, comparado ao valor investido no projeto durante nove meses.

Outro ponto que valida também a hipótese está relacionado às melhorias feitas após a execução do primeiro ciclo da metodologia e também a forma como o design system facilitou a execução dessas melhorias. A principal melhoria feita posteriormente, foi a atualização visual dos componentes core e dos design tokens, esta foi uma melhoria potencializada pelo design system e executada em conjunto por toda a equipe de design, onde melhoramos a conexão visual dos componentes com a marca da empresa e resolvemos as dificuldades que haviam quanto ao

entendimento dos design tokens. As imagens abaixo mostram o resultado destas melhorias.

Figura 57 – Atualização visual do Design System



Fonte: Autoria própria, (2025)

Para pensarmos então em trabalhos futuros, indo além das recomendações tiradas através do método de Bardin (2016) seria necessário entender o impacto causado nas pessoas utilizadoras do software em desenvolvimento, através de uma pesquisa quantitativa sobre o Consumer Effort Score, metodologia utilizada para entendimento de quanto esforço é necessário para o usuário finalizar uma tarefa dentro do sistema, e também captaria percepções qualitativas que envolvam os critérios das heurísticas de Nielsen.

Quanto às recomendações que impactam as equipes de Design e Engenharia ((i) finalizar o “manual do desenvolvedor” (tokens mais exemplos), (ii) instituir um dono dedicado ao repositório e (iii) fechar o circuito de nomenclatura entre Figma e front-end) seriam efetuadas no próximo ciclo de interação do método seguido nesta pesquisa. Para os tópicos (i) e (iii) se faz necessário um melhor entendimento através de novas entrevistas para se definir quais seriam as informações necessárias para se compor o manual para a equipe de engenharia e também para a validação das nomenclaturas entre as duas frentes.

Dessa forma, o Design System continuará melhorando o contexto para os 3 principais atores do contexto de uso, a equipe de design, engenharia e as pessoas utilizadoras do sistema e assim fazendo com que o sistema desenvolvido pela empresa de marketing brasileira cumpra o seu objetivo principal de atender os usuários finais, fazendo com que o investimento da empresa esteja dando resultado.

REFERÊNCIAS

- BARBOSA, Simone Diniz Junqueira; Da Silva, Bruno Santana. **Interação Humano-Computador**: Elsevier Editora Ltda, 2010.
- NIELSEN NORMAN GROUP. **10 Usability Heuristics for User Interface Design**. Abr 24, 1994 · Atualizado Jan. 30, 2024. Disponível em: <https://www.nngroup.com/articles/ten-usability-heuristics/>.
- NIELSEN NORMAN GROUP. **Maintain Consistency and Adhere to Standards (Usability Heuristic #4)**. January 10, 2021. Disponível em: <https://www.nngroup.com/articles/consistency-and-standards/>.
- SOUZA, Sergio Adriano Menezes Dos Santos. **DESIGN SYSTEM EM UMA HOLDING DE ENSINO SUPERIOR**. Escola Superior de Desenho Industrial, Rio de Janeiro, 2023.
- DA SILVA, Matheus Procópio. **ATINGINDO INTERFACES DE USUÁRIOS CONSISTENTES POR COMPOSIÇÃO E SISTEMAS DE TOKENS**. Universidade Federal de Campina Grande, CAMPINA GRANDE/PB, 2019.
- PEREIRA, Marcus Vinicius Santos. **DESIGN SYSTEMS: UM ESTUDO EXPLORATÓRIO SOBRE A DOCUMENTAÇÃO DE COMPONENTES PARA INTERFACES DE USUÁRIO**. Universidade Federal do Ceará, Campus de Quixadá, Curso de Design Digital, Quixadá, 2024.
- VALENTE, Marco Tulio. **ENGENHARIA DE SOFTWARE MODERNA: PRINCÍPIOS E PRÁTICAS PARA DESENVOLVIMENTO DE SOFTWARE COM PRODUTIVIDADE**. Editora: Independente, 2020.
- KHOLMATOVA, Alla. **DESIGN SYSTEMS: A PRACTICAL GUIDE TO CREATING DESIGN LANGUAGES FOR DIGITAL PRODUCTS**. Publicado em 2017 por Smashing Media AG, Freiburg, Alemanha.
- BRAD FROST. **The Design System Ecosystem**. 21 Sep, 2023. Disponível em: <https://bradfrost.com/blog/post/the-design-system-ecosystem/>. Acesso em: 6 Jan. 2025.
- NATHAN CURTIS. **Defining Design Systems**. Oct 10, 2017. Disponível em: <https://medium.com/eightshapes-llc/defining-design-systems-6dd4b03e0ff6>. Acesso em: 6 Jan. 2025.
- DAN MALL. **What is a Design System?: 6 different types of design systems**. 03 Jan 2023. Disponível em: <https://danmall.com/posts/what-is-a-design-system/>. Acesso em: 6 Jan. 2025.
- BRAD FROST. **Design Systems are for user interfaces**. 15 Nov, 2021. Disponível em: <https://bradfrost.com/blog/post/design-systems-are-for-user-interfaces/>. Acesso em: 6 Jan. 2025.

BRAD FROST. **Managing Technology-Agnostic Design Systems**. 13 Nov, 2017. Disponível em:
<https://bradfrost.com/blog/post/managing-technology-agnostic-design-systems/>
Acesso em: 28 Mai. 2025.

NZONGO, Filipe. **Sistema de Design Generativo, Como meio Para Aumentar a Eficiência Operacional, Reduzir Custos e Explorar Novos Horizontes de Interface**. PUCRS ONLINE, 2022.

NZONGO, Filipe. **UI Design em foco: História e tipos**. UX Collective, Jul 15, 2024. Disponível em:
<https://brasil.uxdesign.cc/ui-design-em-foco-historia-e-conceitos-5b6b3371a097>.
Acesso em: 6 Jan. 2025.

SÖNKE ROHDE. **Living Design System: An approach for delivering a consistent design cross-platform**. Sep 11, 2014. Disponível em:
<https://medium.com/salesforce-ux/living-design-system-3ab1f2280ef7>. Acesso em: 6 Jan. 2025.

GALL, John. **Systemantics**. Editora: Times Books, 1975.

PEFFERS, K.; TUUNANEN, T.; ROTHENBERGER, M. A.; CHATTERJEE, S.
A Design Science Research Methodology for Information Systems Research.
Journal of Management Information Systems, v. 24, n. 3, p. 45-77, 1 dez 2007.

AKEN, JOAN ERNST VAN; BERENDS, H.; BIJ, H. VAN DER. **Problem Solving in Organizations**. 2. ed. United Kingdom, Cambridge: University Press Cambridge, 2012. p. 235

AKEN, J.E. VAN. Management Research as a Design Science: Articulating the Research Products of Mode 2 Knowledge Production in Management. **British Journal of Management**, v. 16, n. 1, p. 19-36, mar 2005

AKEN, J. E. VAN. Management Research Based on the Paradigm of the Design Sciences : The Quest for Field-Tested and Grounded Technological Rules. **Journal of Management Studies**, v. 41, n. 2, p. 219-246, 2004.

AKEN, JOAN ERNST VAN. **The Research Design for Design Science Research in Management**. . Eindhoven: [s.n]. , 2011

DRESCH, ALINE. **Design Science e Design Science Research como Artefatos Metodológicos para Engenharia de Produção**. Universidade do Vale do Rio dos Sinos, São Leopoldo, 2013.

NUNAMAKER, J. F.; CHEN, M.; PURDIN, T. D. M. Systems Development in Information Systems Research. **Journal of Management Information Systems**, v. 7, n. 3, p. 89-106, 1991.

EEKELS, J.; ROOZENBURG, N. F. M. A methodological comparison of the structures of scientific research and engineering design: their similarities and differences. **Design Studies**, v. 12, n. 4, p. 197-203, out 1991.

ROMME, A. G. L. Making a Difference : Organization as Design. **Organization Science**, v. 14, n. 5, p. 558-573, 2003.

SIMON, H. A. **The Sciences of the Artificial**. 3. ed. USA: MIT Press, 1996.

GIBBONS, M.; LIMOGES, C.; NOWOTNY, H. *et al.* **The new production of knowledge: The dynamics of science and research in contemporary societies**. Great Britain: Sage Publications Ltd, 1994.

WALLS, J. G.; WYIDMEYER, G. R.; SAWY, O. A. E. Building an Information System Design Theory for Vigilant EIS. **Information Systems Research**, v. March, p. 36-60, 1992.

BAYAZIT, N. Investigating Design : A Review of Forty Years of Design Research. **Massachusetts Institute of Technology: Design Issues**, v. 20, n. 1, p. 16-29, 2004.

HEVNER, A. R.; MARCH, S. T.; PARK, J.; RAM, S. Design Science in information systems research. **MIS Quaterly**, v. 28, n. 1, p. 75-105, 2004.

SEIN, M. K.; HENFRIDSSON, O.; PURAO, S.; ROSSI, M.; LINDGREEN, R. Action Design Research. **MIS Quaterly**, v. 35, n. 1, p. 37-56, 2011.

VAISHNAVI, V.; KUECHLER, W. **Design Research in Information Systems**. Disponível em: <<http://desrist.org/design-research-in-informationsystems>>. Acesso em: 20 dez. 2011.

TAKEDA, H.; VEERKAMP, P.; TOMIYAMA, T.; YOSHIKAWA, H. Modeling Design Processes. **AI Magazine**, v. 11, n. 4, p. 37-48, 1990.

COLE, R.; PURAO, S.; ROSSI, M.; SEIN, M. K. **Being Proactive : Where Action Research meets Design Research**. Proceedings of the Twenty- Sixth International Conference on Information Systems. **Anais...** Las Vegas: [s.n.] , 2005

TRIVIÑOS, Augusto N. S. **Introdução à pesquisa em ciências sociais: a pesquisa qualitativa em educação**. São Paulo: Atlas, 1987.

FREEMAN, R. Edward. **Stakeholder theory: The state of the art**. Cambridge University Press, 2010. Disponível em: <https://core.ac.uk/download/pdf/346447581.pdf> . Acesso em: 3 out. 2022.

ROSALA, Maria. **How to Analyze Qualitative Data from UX Research: Thematic Analysis**. August 17, 2022. Disponível em: <https://www.nngroup.com/articles/thematic-analysis/>

GIBBSON, Sarah. **UX Roadmaps: Definition and Components**. July 5, 2020. Disponível em: <https://www.nngroup.com/articles/ux-roadmaps/>

MORAN, Kate. **Benchmarking UX: Tracking Metrics**. May 3, 2020. Disponível em: <https://www.nngroup.com/articles/benchmarking-ux/>

IBM. **Carbon Design System**. Last updated 31 Mar, 2022. Disponível em: <https://carbondesignsystem.com>

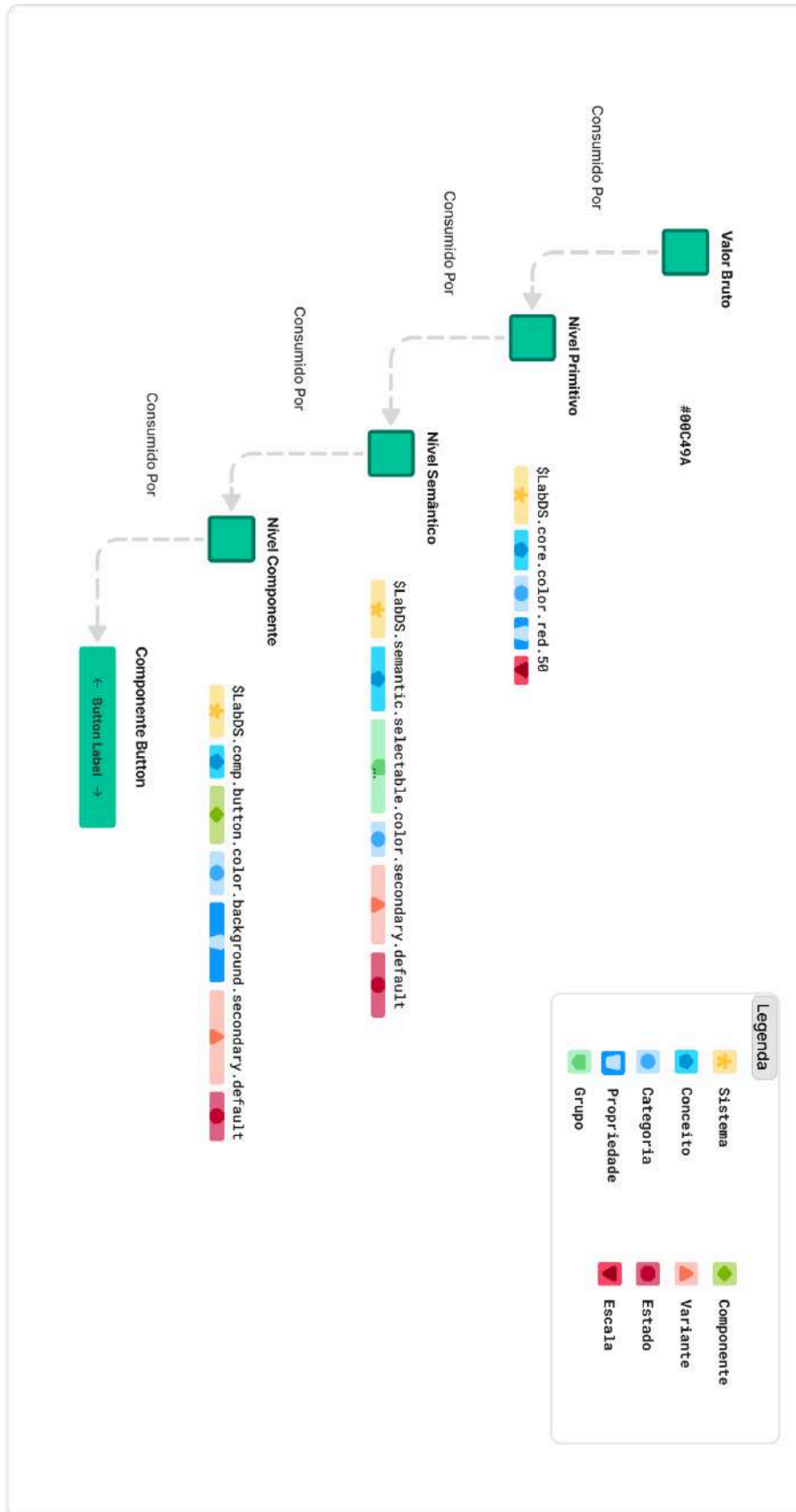
GOOGLE. **Material Design**. Last updated 10 September, 2024. Disponível em: <https://m3.material.io/blog>

APPLE INC. **Human Interface Guidelines**. Last updated October 17, 2024.

Disponível em: <https://developer.apple.com/design/human-interface-guidelines>

APÊNDICE A - EXEMPLO DE APLICAÇÃO DA NOMENCLATURA DEFINIDA

Exemplo - Aplicação Design Tokens



APÊNDICE B - ROTEIRO DE PERGUNTAS DA PESQUISA INICIAL

PESSOAS USUÁRIAS DO DESIGN SYSTEM NO CONTEXTO DE DESIGN

- Pergunta 1 - Quais são as suas maiores dores hoje ao construir uma tela?
- Pergunta 2 - Quais foram as experiências anteriores utilizando uma biblioteca no Figma?
- Pergunta 3 - Você sente falta de algo ao utilizar nossa biblioteca de componentes atual?
- Pergunta 4 - Quais são as suas maiores dores utilizando nossa biblioteca de componentes atual?
- Pergunta 5 - De 0 a 5, qual o seu nível de entendimento utilizando componentes e variantes no Figma? 0 - básico 5 - avançado

PESSOAS USUÁRIAS DO DESIGN SYSTEM NO CONTEXTO DE ENGENHARIA

- Pergunta 1 - Quais são as suas maiores dores hoje ao construir uma tela?
- Pergunta 2 - Quais foram as experiências anteriores utilizando frameworks para o front-end, já trabalhou com componentes pré-prontos?
- Pergunta 3 - Você sente falta de algo ao utilizar nossa biblioteca de componentes atual?
- Pergunta 4 - Quais são as suas maiores dores utilizando nossa biblioteca de componentes atual?
- Pergunta 5 - Algum ponto a mais que você prevê?
- Pergunta 6 - Se for necessário construir um componente hoje, como é esse processo?

PESSOAS CONSTRUTORAS DO DESIGN SYSTEM NO CONTEXTO DE DESIGN

- Pergunta 1 - O que tornaria o design system bem-sucedido?
- Pergunta 2 - Quais são as suas restrições e limitações atuais?
- Pergunta 3 - Quais são os objetivos como construtores?
- Pergunta 4 - Algum ponto a mais que você prevê?

PESSOAS CONSTRUTORAS DO DESIGN SYSTEM NO CONTEXTO DE ENGENHARIA

- Pergunta 1 - O que tornaria o design system bem-sucedido?
- Pergunta 2 - Quais são as suas restrições e limitações atuais?
- Pergunta 3 - Quais são os objetivos como construtores?
- Pergunta 4 - Algum ponto a mais que você prevê?
- Pergunta 5 - Quais foram as experiências anteriores utilizando frameworks para o front-end, já trabalhou com componentes pré-prontos?
- Pergunta 6 - Atualmente usamos algum framework ou linguagem no front-end?
- Pergunta 7 - Se for necessário construir um componente hoje, como é esse processo?

PESSOA REPRESENTANTE DA EQUIPE DE MARKETING DA EMPRESA

- Pergunta 1 - Onde podemos consultar sobre tudo relacionado à marca da empresa? como é atualizado essas materias? Tom de voz e Personas?
- Pergunta 2 - Vocês utilizam alguma plataforma para a construção das telas?
- Pergunta 3 - Vocês possuem um guia de estilo para a construção das telas?
- Pergunta 4 - Quem cuida da construção das telas?

APÊNDICE C - ROTEIRO DE PERGUNTAS DA AVALIAÇÃO NO CONTEXTO DE DESIGN

- Pergunta 1 - Quais foram os projetos em que você trabalhou nos últimos 3 meses?
- Pergunta 2 - Nesses projetos, você utilizou a bibliotecas antiga e a nova?
- Pergunta 3 - Usando o projeto x de exemplo, como foi usar a nova biblioteca para esse contexto?
- Pergunta 4 - Para você, quais foram os pontos de facilidade ao usar a nova biblioteca?
- Pergunta 5 - E quais foram as dificuldades?
- Pergunta 6 - No cenário de adequar o componente ao seu contexto, por exemplo uma tabela, como foi definir as suas propriedades?
 - Existiu algum ponto de empecilho?
 - Houve dúvidas em como usar as propriedades?
- Pergunta 7 - Como você compara a forma de definir propriedades entre a biblioteca antiga e a nova?
- Pergunta 8 - Houve momentos em que foi necessário dar detache em algum componente? em qual caso? o que te motivou?
- Pergunta 9 - Em momentos em que você esteve com dificuldade em usar tal componente, quais ações você fez para tentar resolver essa dificuldade?
- Pergunta 10 - Quanto ao contexto de encontrar um componente no Design System, houve dificuldade? quais foram?
- Pergunta 11 - Ao surgir algum problema com certo componente, o que você fazia em seguida?
- Pergunta 12 - Ao surgir uma necessidade de algo novo no Design System, o que você fazia em seguida?
- Pergunta 13 - Como foi utilizar os design tokens durante a construção das telas?
 - Quais dificuldades você teve ao usar os tokens?
- Pergunta 13 - Na última vez que tivemos uma atualização na biblioteca, houve algum momento de dúvida do que estava sendo atualizado?
- Pergunta 14 - Quando teve uma atualização, o que você fez em seguida?

- Pergunta 15 - Na sua visão o que precisa estar melhor no Design System daqui para frente?
- Pergunta 16 - Na sua visão o que precisa continuar e ser mais explorado no LabDS para o Q4 em relação ao Q3?

APÊNDICE D - ROTEIRO DE PERGUNTAS DA AVALIAÇÃO

- Pergunta 1 - Realizando um comparativo entre a biblioteca antiga e a nova, houve diferença no teu dia a dia de construção de telas? quais foram?
- Pergunta 2 - Seguindo o contexto de consumir os componentes da biblioteca atual e adaptar para o seu contexto, houve diferença para você, em relação a biblioteca antiga? quais foram?
- Pergunta 3 - Ainda nesse contexto de consumir os componentes e adaptar para o seu contexto de uso, como você está lidando com o fato dos componentes estarem fechados para atualizações?
- Pergunta 4 - Agora quanto a lista de componentes atuais que temos, você sente falta de algum que poderia ser construído?
- Pergunta 5 - Na última vez em que você se deparou com alguma dificuldade no contexto da LabDS, qual foi a sua ação?
- Pergunta 6 - Na última vez em que você precisou que houvesse um ajuste em algo relacionado ao Design System, como você resolveu essa questão?
- Pergunta 7 - Quando você precisa de um componente específico, qual a forma em que procura?
- Pergunta 8 - Na sua visão o que precisa estar melhor no Design System nos próximos meses?
- Pergunta 9 - Na sua visão o que fizemos bem e precisa continuar sendo mais explorado no Design System para os próximos meses?