

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE
DO NORTE

Radmila Gama da Silva

**Avalista: Concepção e Desenvolvimento do frontend de uma ferramenta para Avaliação
de Interfaces por Inspeção**

Natal/RN

2025

Radmila Gama da Silva

Avalista: Concepção e Desenvolvimento do frontend de uma ferramenta para Avaliação de Interfaces por Inspeção

Trabalho de Conclusão de Curso apresentado ao Curso Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Analista de Sistemas.

Orientadora: Dra. Marília Aranha Freire

Coorientadora: Dra. Silvia Aparecida de Oliveira de Alencar Matos

Natal/RN

2025

Silva, Radmila Gama da.
S586a Avalista : concepção e desenvolvimento do frontend de uma ferramenta
para avaliação de interfaces por inspeção / Radmila Gama da Silva. – 2025.
56 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Instituto Federal de
Educação, Ciência e Tecnologia do Rio Grande do Norte, Natal, 2025.
Orientadora: Dra. Marília Aranha Freire.
Coorientadora: Dra. Sílvia Aparecida de Oliveira de Alencar Matos.

1. Software - Usabilidade. 2. Interação homem-máquina. 3. Avaliação
heurística. I. Título.

SIBi/IFRN

CDU: 004.4

Radmila Gama da Silva

Avalista: Concepção e Desenvolvimento do frontend de uma ferramenta para Avaliação de Interfaces por Inspeção

Trabalho de Conclusão de Curso apresentado ao Curso Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Analista de Sistemas.

Trabalho de Conclusão de Curso, Dissertação ou Tese aprovado (a) em 28/07/2025 pela seguinte Banca Examinadora:

Documento assinado digitalmente
 **MARILIA ARANHA FREIRE**
Data: 12/08/2025 15:03:56-0300
Verifique em <https://validar.iti.gov.br>

Marília Aranha Freire, Dra. – Orientadora

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Documento assinado digitalmente
 **SILVIA APARECIDA DE OLIVEIRA DE ALENCAR M/**
Data: 11/08/2025 17:48:50-0300
Verifique em <https://validar.iti.gov.br>

Silvia Aparecida de Oliveira de Alencar Matos, Dra – Coorientadora

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Documento assinado digitalmente
 **ALESSANDRO JOSE DE SOUZA**
Data: 12/08/2025 14:44:30-0300
Verifique em <https://validar.iti.gov.br>

Alessandro José de Souza, Dr.

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

À minha madrinha, Maria de Fátima Cabral Barbosa, que esteve ao meu lado nos momentos mais difíceis, me acolheu com generosidade durante a pandemia e me deu as condições para continuar estudando.

Essa conquista também é sua. Obrigada por acreditar em mim quando eu mais precisei.

À minha avó, Avani Felipe (em memória), que partiu em 2019, mas cuja torcida eu sinto até hoje. Eu lembro até hoje da nossa última conversa e sei que, onde você estiver, está sorrindo.

Esta conquista também tem o seu brilho.

AGRADECIMENTOS

Agradeço às minhas orientadoras, Marília Freire e Silvia Matos, pela excelente orientação, pelo apoio constante e pelo conhecimento compartilhado ao longo da minha trajetória na faculdade. Suas contribuições foram fundamentais para a realização deste trabalho e para a minha formação.

Ao meu colega de TCC, Ramon Vieira, por ter topado entrar comigo nessa jornada com o Avalista, dividindo os desafios e aprendizados do processo.

Um agradecimento especial à Ana Maria, por ter ajudado a escolher o nome, por ter feito pesquisas de bibliotecas em *React* para me auxiliar na implementação.

Agradeço a professora Danielle Freitas pelas aulas de IHC, de quando eu ainda estava cursando info web. Elas foram essenciais para a minha escolha de profissão e para a minha permanência na área de tecnologia.

Agradeço também a cada pessoa que respondeu à minha pesquisa e participou, de alguma forma, da concepção e construção do Avalista. A colaboração de vocês foi essencial para que este projeto se concretizasse.

Deixo aqui meu reconhecimento e gratidão aos nomes abaixo:

Felipe Alves de Vasconcelos,
Arthur de Melo Galvão,
Felipe Xavier de Carvalho,
Cibele Barros Diniz,
Pedro Maure Frutuoso de Andrade,
Rômulo,
Marcus Vinicius,
Renan do Nascimento Lisboa,
Luna,
Manoel,
Ana Maria,
Luan da Costa,
Ramon,
Maycon Alexsander,
Leonardo Viana Brito,
Alanis Isabelle de Oliveira Silva,
Ítalo Almeida,
Dina,
Lucas Melo,
Adna Oliveira Silva,
Maria Giovanna,
Dassaev Lima,
Samuel Lúcio,
Yuri Oliveira,
Ítalo Gabriel da Silva Monteiro,

Rebeca Noemi,
Ester,
Pedro,
Anna Carolinne Albuquerque,
Anaíse,
Mauro,
Giovanna,
Ariane Silveira Correa,
Gabrielly Stéphaney,
Gustavo Maia Militão,
Antônio Irineu Filho,
Ícaro Gabriel,
Tâmara Thais,
Robson Fernandes,
Deividson Pereira Oliveira da Silva,
Ítalo Mageste,
Rafael P.,
Joana Darc Fernandes Silva,
Danilo Ladeira de Oliveira Cosme,
Daniel Braga,
Alice Fonseca,
Davi Alessandro Canuto da Silva Gregório,
Erick Carlos,
Igor Gabriel,
José Vinicius Targino Muniz,
Victor Paiva Passos,
Ryan Paulista,
Paulo Henrique Soares Lima e Silva,
Jesrriel Moura Lopes,
Carolina Azevedo,
Emanuelly Karine Fernandes dos Santos,
Victor Matheus Lopes de Sousa,
Josephy Cruz Araújo e
Mário Luiz Silva Guimarães.

What if I told you none of it was accidental... I laid the groundwork and then, just like clockwork, The dominoes cascaded in a line. What if I told you I'm a mastermind?

Taylor Swift, *Mastermind*

RESUMO

A avaliação de interfaces é uma prática fundamental para garantir a qualidade e a usabilidade de sistemas interativos. No contexto do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do IFRN, essa prática é central e apoiada por um projeto de pesquisa iniciado em 2019, que resultou na criação da Lista Eureka 2024, um consolidado instrumento de avaliação por inspeção. Este trabalho se insere como a próxima etapa evolutiva dessa pesquisa, abordando o desafio de otimizar a aplicação manual da lista. O objetivo foi conceber e desenvolver a primeira versão do protótipo *frontend*, uma ferramenta web para automatizar, padronizar e otimizar o processo de avaliação. A metodologia partiu do Laboratório de Problemas para a investigação das necessidades e culminou no desenvolvimento do *frontend* com *React*, guiado por Histórias de Usuário. O principal resultado é um protótipo que implementa o fluxo essencial de uma avaliação individual, desde a gestão de projetos até o registro de problemas, e que foi desenvolvido para se integrar a uma API de *backend*, desenvolvida em paralelo por outro estudante do curso. Conclui-se que o protótipo valida a viabilidade da proposta, estabelecendo uma base para trabalhos futuros, como a implementação de módulos colaborativos e estratégias de engajamento.

Palavras-chave: usabilidade; avaliação heurística; interface humano-computador; ferramenta web; automação de processos.

ABSTRACT

Interface evaluation is a fundamental practice for ensuring the quality and usability of interactive systems. In the context of the Technology in Systems Analysis and Development course at IFRN, this practice is central and supported by a research project initiated in 2019, which resulted in the creation of the Eureka List 2024, a consolidated inspection-based evaluation instrument. This work presents itself as the next evolutionary step of this research, addressing the challenge of optimizing the manual application of the list. The objective was to conceive and develop the first version of the frontend prototype, a web tool to automate, standardize, and optimize the evaluation process. The methodology was based on the Problem Laboratory for the investigation of needs and culminated in the development of the frontend with React, guided by User Stories. The main result is a prototype that implements the essential workflow of an individual evaluation, from project management to the registration of issues, and which was developed to integrate with a backend API, developed in parallel by another student of the course. It is concluded that the prototype validates the proposal's feasibility, establishing a foundation for future work, such as the implementation of collaborative modules and engagement strategies.

Keywords: usability; heuristic evaluation; human-computer interaction; web tool; process automation.

SUMÁRIO

1 Introdução.....	25
1.1 Objetivos.....	26
1.2 Justificativa.....	26
1.3 Metodologia.....	27
1.4 Estrutura do Trabalho.....	28
2 Fundamentação Teórica.....	29
2.1 Avaliação de Interfaces.....	29
2.1.1 Interação e Interface.....	29
2.1.2 Dimensões da Qualidade de Uso.....	30
2.2 Avaliação Por Inspeção.....	30
2.3 O Laboratório de Problemas como Metodologia de Projeto.....	31
2.4 Engenharia de Requisitos.....	32
2.4.1 Histórias de Usuário.....	33
2.5 Tecnologias utilizadas no trabalho.....	34
2.5.1 React: A Biblioteca para Interfaces de Usuário Declarativas.....	34
2.5.2 TypeScript: Robustez com Tipagem Estática.....	35
2.5.3 Vite: O Ambiente de Desenvolvimento Moderno.....	35
2.5.4 Ecossistema e Bibliotecas de Suporte.....	35
2.5.5 Arquitetura Baseada em Funcionalidades (Feature-Based).....	36
2.6 Trabalhos Correlatos.....	37
3 Desenvolvimento.....	38
3.1 Fundamentação e Pesquisa.....	38
3.1.1 Apresentação e Análise da Pesquisa com Estudantes.....	38
3.1.2 Problema Identificado.....	38
3.1.3 Requisitos da Plataforma.....	39
3.2 Concepção e Design.....	43
3.2.1 Laboratório de Problemas.....	43
3.2.1.1 Quadro de Atributos e Decisões Projetuais.....	43
3.2.2 Criação da Identidade Visual.....	45
3.2.2.1 Símbolo.....	45
3.2.2.2 Logotipo e Tipografia.....	46
3.2.2.3 Grade Construtiva e Proporções.....	46
3.2.2.4 A Paleta de Cores Mutante.....	47
3.2.2.5 Sistema de Assinaturas Visuais.....	48
3.3 Execução e Entrega.....	52
3.3.1 Planejamento das Histórias de Usuário.....	52
3.3.2 Sistema de Design e Estilização.....	52
3.3.3 Decisões Arquiteturais.....	53
3.3.4 Integração com o Backend.....	54
4 Resultados.....	55
4.1 Investigação e Concepção.....	55

4.2 Solução Implementada.....	55
4.2.1 Apresentação das Interfaces Desenvolvidas.....	56
4.2.2 Funcionalidades e Contribuições.....	61
5 Conclusão e Trabalhos Futuros.....	63
5.1 Conclusão.....	63
5.2 Trabalhos Futuros.....	63
6 Referências.....	65

1 Introdução

A qualidade da interface é um fator fundamental para que sistemas interativos sejam utilizados com sucesso. Nesse contexto, a avaliação de interfaces desempenha um papel crucial, pois permite a identificação e o ajuste de problemas de interação durante todo o ciclo de desenvolvimento. A área de Interação Humano-Computador (IHC) dedica-se exatamente ao projeto (design), avaliação e implementação de sistemas computacionais interativos para uso humano, juntamente com os fenômenos associados a este uso (HEWETT et al., 1992).

Existem diferentes métodos para avaliar interfaces, geralmente classificadas em dois grandes grupos: avaliações com usuários e avaliações por especialistas. Segundo Fernandez *et al.* (2011), os métodos de avaliação de usabilidade (Usability Evaluation Methods - UEMs) podem ser divididos entre métodos empíricos, que envolvem coleta e análise de dados a partir da interação de usuários reais com o sistema, e métodos de inspeção, que consistem na análise crítica de interfaces realizada por avaliadores treinados. Embora os métodos empíricos sejam bastante eficazes, eles exigem maior investimento de tempo e recursos, e são geralmente aplicados em fases mais avançadas do desenvolvimento. (Preece, Roger e Sharp, 2023)

Como destacam Prates e Barbosa (2003), é essencial que as interfaces sejam avaliadas continuamente durante o processo de design (avaliação formativa), pois quanto mais cedo os problemas são encontrados, menor é o custo para corrigi-los. Além disso, os autores alertam que um projetista não deve presumir que os usuários são como ele próprio e que sua avaliação individual seria suficiente para atestar a qualidade de um sistema.

Desde o final dos anos 80, as práticas ensinadas são amplamente fundamentadas em obras de referência de autores como Norman (1988), Nielsen (1994) e Preece, Rogers e Sharp (2013), que estabeleceram conceitos e diretrizes essenciais para a criação de interfaces de alta qualidade. No curso de Tecnologia em Análise e Desenvolvimento de Sistemas (TADS) do Instituto Federal do Rio Grande do Norte (IFRN), as disciplinas de Interface Humano-Computador e Testes de Software abordam esses conteúdos, formando profissionais capazes de projetar e avaliar soluções digitais.

Para apoiar o ensino e a aplicação prática dessas diretrizes, um projeto de pesquisa foi iniciado em 2019 no curso de TADS, resultando em dois artefatos centrais. O primeiro é a Lista Eureka (MATOS et al., 2024), um instrumento que compila 72 recomendações de boas práticas para interfaces contemporâneas. Para além da ferramenta, foi desenvolvida também a Técnica Eureka, um método que guia a aplicação da lista, estruturando o processo de avaliação em três fases claras: planejamento, execução (individual e em grupo) e entrega do relatório. Com a consolidação de um instrumento (a Lista) e de um processo (a Técnica), o próximo passo natural da pesquisa tornou-se a automação, cenário que deu origem à proposta da plataforma Avalista.

Uma pesquisa com uma amostra de 59 estudantes em relação à realização de avaliações por inspeção, apontou para uma certa dificuldade dos mesmos em utilizar a técnica de forma manual. Apesar de perceberem que a qualidade das interfaces evolui com a avaliação, o processo foi visto como repetitivo e por vezes cansativo, além da dificuldade em identificar críticas sobre o próprio trabalho.

Diante desse contexto, propõe-se a Avalista: uma plataforma web que visa organizar e padronizar o processo de avaliação de interfaces, utilizando como base a Lista Eureka (2024). A materialização desta proposta foi concebida como um projeto colaborativo, dividido entre dois trabalhos de conclusão de curso: um focado na concepção e desenvolvimento do *frontend* (o presente trabalho) e outro focado no desenvolvimento do *backend*, realizado pelo discente Ramon Vieira. A proposta busca, portanto, automatizar a aplicação da técnica de avaliação por meio da construção de uma ferramenta integrada.

1.1 Objetivos

O objetivo geral deste trabalho é a concepção e o desenvolvimento do *frontend* de uma plataforma web que organize e otimize a avaliação de interfaces por inspeção, utilizando a Lista Eureka como base para estudantes de tecnologia do IFRN.

Objetivos Específicos:

- Investigar os principais fatores que dificultam a realização da avaliação de interfaces por parte dos estudantes;
- Levantar e documentar os requisitos da aplicação;
- Conceber a solução de interface, incluindo a identidade visual, a arquitetura da informação e o fluxo de interação do usuário;
- Implementar o módulo de autenticação e gestão de projetos da plataforma;
- Desenvolver a interface para o registro de problemas de usabilidade, utilizando a Lista Eureka de forma interativa.

1.2 Justificativa

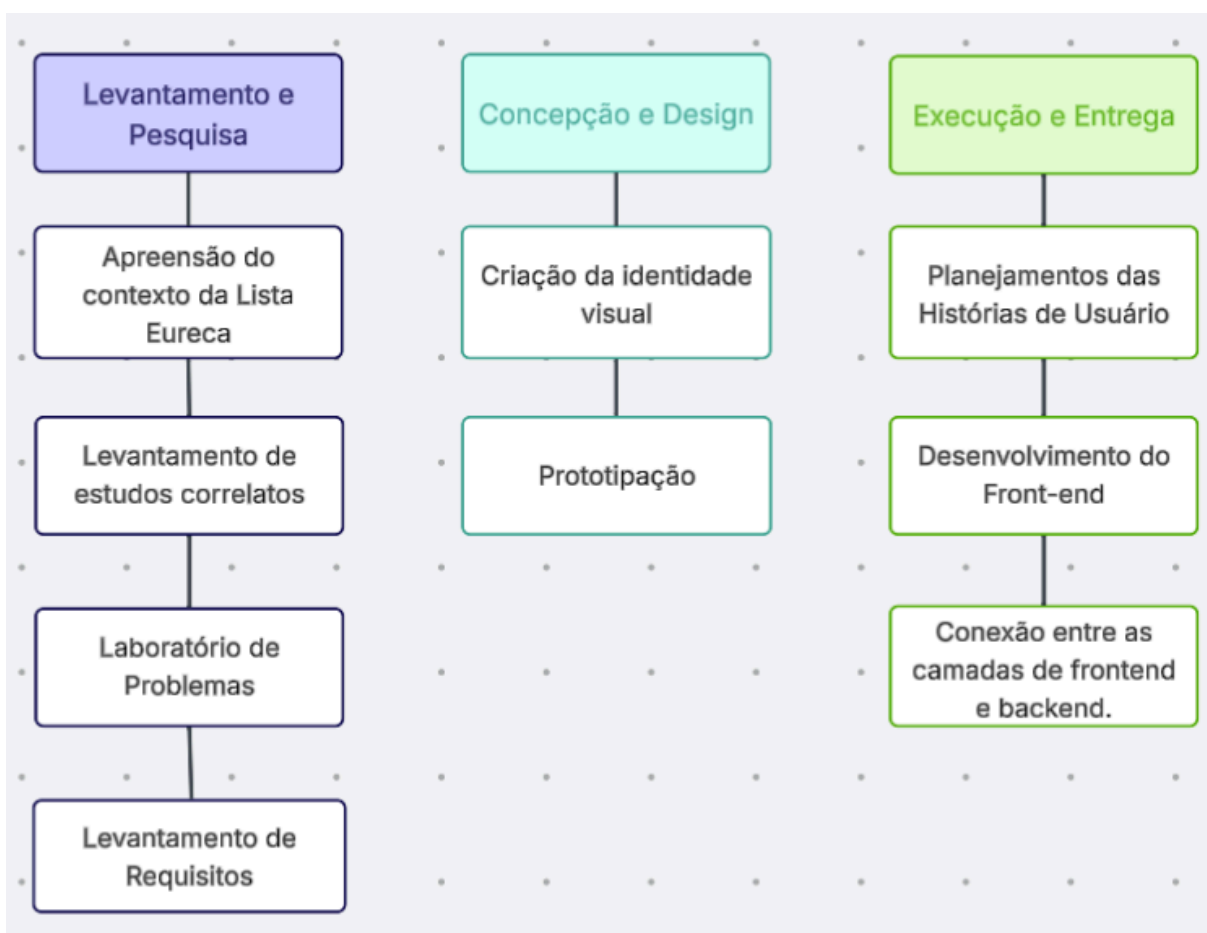
Desde 2019 houve um esforço de sistematização de listas de diretrizes a partir do trabalho pregresso de vários autores da área de design de interfaces. Esse projeto de pesquisa culminou, no ano de 2024, na disponibilização de uma ferramenta para auxiliar a avaliação por inspeção. Essa ferramenta, chamada de Lista Eureka 2024, é uma lista de diretrizes atualizada para as interfaces contemporâneas que combina recomendações de formatação, funcionalidade, usabilidade, acessibilidade, segurança de dados, entre outros.

Para além da ferramenta foi ainda desenvolvida uma técnica para guiar a aplicação da lista, intitulada Técnica Eureka. O próximo passo previsto pela pesquisa seria exatamente a automação desse processo, utilizando a lista Eureka como ferramenta com o sentido de apoiar o processo de avaliação manual utilizado atualmente.

1.3 Metodologia

A metodologia adotada para a concepção e o desenvolvimento da plataforma Avalista foi fundamentada em dois princípios essenciais do Design Centrado no Usuário e da Interação Humano-Computador. O design centrado no usuário, conforme preconizado por autores como Lowdermilk (2014), é uma abordagem que coloca as necessidades, os desejos e as limitações dos usuários finais no centro de cada etapa do processo. Complementarmente, a abordagem iterativa, descrita por Preece, Rogers e Sharp (2013), reconhece que o design é um processo cíclico de prototipação, avaliação e refinamento. A união desses dois princípios guiou a estrutura do projeto, que foi dividida em três grandes fases, conforme ilustrado na Figura 1.

Figura 1 – Desenho da Metodologia



Fonte: Autoria Própria (2025)

A primeira fase, de Fundamentação e Pesquisa, consistiu em três sub-fases sequenciais. Primeiramente, foi realizada a apreensão do contexto do projeto de pesquisa da Lista Eureka (MATOS et al., 2024). Em seguida, foi feito um levantamento de estudos correlatos, analisando ferramentas existentes para identificar lacunas e oportunidades. Por fim, a terceira sub-fase consistiu em uma pesquisa com 59 estudantes do curso de TADS para entender as dificuldades do processo manual de avaliação, servindo de base para a definição dos requisitos iniciais da plataforma.

A fase de Concepção e Design seguiu os princípios do Laboratório de Problemas, focando no desenvolvimento das decisões projetuais. Nesta etapa, foi definida a criação da identidade visual da marca, incluindo o símbolo e a paleta de cores. Para traduzir os requisitos em tarefas de desenvolvimento, foi realizado o planejamento com Histórias de Usuário (HUs), etapa conduzida em conjunto com colaborador responsável pelo *backend* do projeto.

Por fim, a fase de Execução e Entrega compreendeu o desenvolvimento do *frontend* da aplicação em *React*, guiado pelas Histórias de Usuário definidas anteriormente. Este trabalho ocorreu em paralelo ao desenvolvimento do *backend* por um colaborador de outro TCC. O processo culminou na entrega da interface interativa da plataforma Avalista, que integra as camadas de *frontend* e *backend* e serve como prova de conceito da solução proposta.

1.4 Estrutura do Trabalho

O trabalho está estruturado em seis capítulos principais. Além da Introdução, que compõe o Capítulo 1, o documento segue com o Capítulo 2, que apresenta a Fundamentação Teórica, abordando os conceitos que embasam a proposta da plataforma Avalista, como usabilidade, avaliação heurística e ensino de design de interfaces.

No Capítulo 3, é descrito o Desenvolvimento do Projeto, desde a pesquisa inicial até as etapas de implementação, detalhando a construção da plataforma, incluindo o projeto da interface, as funcionalidades desenvolvidas e a arquitetura técnica do *frontend*.

O Capítulo 4 trata dos Resultados, apresentando a primeira versão do Avalista, sua análise funcional e os resultados alcançados com base nos objetivos definidos.

Já o Capítulo 5, Conclusão e Trabalhos Futuros, reúne as principais contribuições do projeto, seus impactos no ensino da avaliação de interfaces por inspeção, bem como sugestões para aprimoramento e expansão da plataforma.

O Capítulo 6, Referências, apresenta a bibliografia utilizada como base teórica e metodológica para a realização do trabalho.

2 Fundamentação Teórica

Este capítulo reúne os pilares teóricos que sustentam a concepção e o desenvolvimento da aplicação Avalista. São explorados os conceitos essenciais de Interação Humano-Computador (IHC), com foco em avaliação por inspeção. Além disso, são detalhadas as metodologias e ferramentas que guiaram o projeto, como o Laboratório de Problemas e as Histórias de Usuário, contextualizando o trabalho frente a outras soluções existentes.

2.1 Avaliação de Interfaces

A usabilidade é um atributo de qualidade fundamental que mede a facilidade de uso de uma interface. Segundo Jakob Nielsen (1994), um dos pioneiros da área, a usabilidade é composta por cinco dimensões principais: facilidade de aprendizado, eficiência de uso, memorização, baixa ocorrência de erros e satisfação do usuário.

Para criar uma interface com qualidade na interação são empregados Métodos de Avaliação de Interface. Conforme apontado por Fernandez et al. (2011), esses métodos se dividem em duas grandes categorias:

- **Métodos Empíricos:** Envolvem a participação direta de usuários finais, como nos testes de usabilidade, onde se observa a interação real das pessoas com o sistema.
- **Métodos de Inspeção:** Realizados por especialistas em usabilidade ou design, que analisam a interface com base em diretrizes e princípios predefinidos. A avaliação heurística é o método de inspeção mais conhecido.

A qualidade da interface é um fator determinante para o sucesso de qualquer sistema interativo. No campo da Interação Humano-Computador (IHC), a avaliação de interfaces surge como um processo essencial para analisar e alcançar a qualidade de uso de um software, permitindo a identificação e correção de problemas de interação desde as fases iniciais do desenvolvimento (PRATES; BARBOSA, 2003). Para fundamentar essa prática, é crucial distinguir os conceitos de interação e interface, bem como as diferentes dimensões que compõem a qualidade de uso.

2.1.1 Interação e Interface

A interação pode ser compreendida como o processo de comunicação entre uma pessoa e um sistema computacional. Já a interface é a porção do sistema com a qual o usuário entra em contato para que essa comunicação aconteça. Uma definição clássica de Moran (1981) descreve a interface como a parte do sistema com a qual uma pessoa entra em contato de forma física, perceptiva ou conceitual. Isso significa que a interface não é apenas o que o usuário vê e toca (dimensões perceptiva e física), mas também o modelo mental que ele constrói para entender o sistema (dimensão conceitual).

2.1.2 Dimensões da Qualidade de Uso

A qualidade de uso de um sistema é um conceito multidimensional. Prates e Barbosa (2003) destacam que, embora a usabilidade seja a dimensão mais conhecida, outros conceitos como comunicabilidade e aplicabilidade são igualmente importantes para uma análise completa.

- Usabilidade: Focada na eficiência e satisfação, a usabilidade, segundo Nielsen (1994), pode ser decomposta em fatores como facilidade de aprendizado (esforço para atingir proficiência), eficiência de uso (rapidez para realizar tarefas), memorização (facilidade de lembrar como usar o sistema após um tempo), baixa ocorrência de erros e a satisfação subjetiva do usuário.
- Comunicabilidade: Este conceito, proposto por de Souza (1999), avalia a capacidade da interface de comunicar ao usuário a intenção do designer. Uma interface com alta comunicabilidade permite que o usuário entenda claramente para que o sistema serve, como ele funciona e quais são seus princípios de interação, evitando que o uso se torne um exercício de tentativa e erro.
- Aplicabilidade: Relacionado à utilidade e flexibilidade de um sistema em diversas situações, este conceito, discutido por Fischer (1998), critica interfaces excessivamente restritivas (às vezes chamadas de "à prova de idiotas"). Em vez disso, defende sistemas que ampliem as capacidades dos usuários, tratando-os como especialistas em suas próprias tarefas e permitindo que eles ajam de forma mais inteligente e criativa.

2.2 Avaliação Por Inspeção

A avaliação é uma atividade central no ciclo de vida do design de sistemas interativos, cujo objetivo fundamental é analisar a qualidade de uso de um software e verificar se ele apoia adequadamente seus usuários em suas tarefas. Autores como Preece et al. (1994) e Hartson (1998) distinguem duas finalidades principais para esta atividade: a avaliação formativa, que ocorre durante o processo de desenvolvimento para identificar problemas e guiar melhorias; e a avaliação somativa, realizada ao final para julgar a qualidade do produto. O presente trabalho foca nos métodos de inspeção, que são amplamente utilizados por serem mais rápidos, de menor custo e aplicáveis desde as fases iniciais do projeto.

A categoria de métodos de inspeção, também chamada de avaliação analítica, inclui diversas técnicas, como o Percurso Cognitivo (Cognitive Walkthrough), a inspeção de conformidade com diretrizes e a Avaliação Heurística (PRATES; BARBOSA, 2003). Dentre estas, a Avaliação Heurística, popularizada por Jakob Nielsen (1994), é a mais difundida na indústria e na academia. Sua premissa é que um pequeno grupo de especialistas, ao inspecionar uma interface com base em um conjunto de princípios reconhecidos (as "heurísticas"), pode identificar uma parcela significativa dos problemas de usabilidade de um sistema.

Embora as 10 heurísticas de Nielsen (1994) sejam a referência pioneira, a crescente complexidade das interfaces contemporâneas tornou evidente a necessidade de diretrizes mais abrangentes. Com o passar dos anos, diversos autores propuseram novas listas para endereçar aspectos não cobertos originalmente, como interações em dispositivos móveis e novas tecnologias web. Autores como Dan Saffer (2010), com seus princípios para design de interação, e David Benyon (2011), com suas diretrizes para sistemas interativos, são exemplos desse esforço contínuo para adaptar os fundamentos da avaliação.

É dentro desse contexto de evolução que se insere a Lista Eureka (2024). Desenvolvida no IFRN desde 2019, ela representa um esforço de sistematização que compila e adapta as recomendações de mais de 18 autores da área, incluindo os já citados. O resultado é um conjunto de 72 recomendações de boas práticas, organizadas em oito categorias, que serve como um recurso didático centralizado e atualizado para a realidade dos projetos desenvolvidos no curso de TADS, sendo a espinha dorsal da plataforma Avalista.

2.3 O Laboratório de Problemas como Metodologia de Projeto

A abordagem metodológica utilizada neste projeto é uma adaptação do método Roda de Fuxico (MATOS, 2023), que foi originalmente concebido para apoiar processos de criação de identidades visuais. Para o contexto dos projetos de software do IFRN, o método foi reestruturado para funcionar como um roteiro para o desenvolvimento de interfaces gráficas.

Como ponto de partida deste roteiro, foi desenvolvido um seminário intensivo, com oito horas de trabalho, intitulado Laboratório de Problemas. O seu objetivo principal é guiar as equipes na definição das decisões projetuais estratégicas que servirão como alicerce para a criação de seus projetos de interface.

O princípio fundamental que norteia o Laboratório de Problemas é a mudança de uma mentalidade focada no produto para uma mentalidade centrada nos atores e fatores que interessam ao projeto. Desse modo, a metodologia instrumentaliza os estudantes a saírem do campo das suposições e a basearem suas decisões de design em evidências coletadas no "mundo real".

O processo do Laboratório de Problemas é organizado em um fluxo de sete etapas principais, que conduzem o estudante de forma lógica desde a exploração inicial até a materialização de uma proposta de solução. As fases são:

1. **Busca de Problemas:** A etapa inicial consiste na imersão no tema de interesse para identificar não apenas problemas evidentes, mas também oportunidades e necessidades latentes. Trata-se de um momento de prospecção e observação do cotidiano para a identificação de problemas reais.
2. **Coleta de Dados:** Uma vez definido o problema, aprofunda-se o conhecimento por meio de ferramentas de pesquisa como entrevistas, questionários e observação direta. A coleta de dados é realizada com os diversos atores do projeto — como clientes e os diferentes tipos de sujeitos (apoiadores, oponentes, etc.) — e também com os próprios

designers, que expõem suas ideias e reconhecem seus vieses. Por fim, a coleta abrange fatores de contexto e a análise de aplicativos similares.

3. **Análise:** Esta fase é o momento de organizar e transformar as informações coletadas em textos-síntese e palavras-chave. Também é a etapa de comparar atributos dos aplicativos similares, destacando as características a serem "imitadas", como inspiração, e "evitadas", como más práticas.
4. **Interpretação:** Esta fase é dedicada à tradução dos dados analisados em *insights* acionáveis e à tomada de decisões estratégicas. É o momento em que os padrões identificados são interpretados para fazer escolhas e tomar partido. O resultado dessa "costura" de informações é a criação de um Quadro de Atributos, que por sua vez dá origem ao conceito do projeto e às decisões projetuais que guiarão o desenvolvimento.
5. **Desenho:** Com as estratégias projetuais definidas, a fase de Desenho foca na materialização da identidade visual da interface. Nesta etapa são definidos os elementos gráficos fundamentais, como a marca, a tipografia, a paleta de cores, a hierarquia visual da informação e a grade de acomodação do conteúdo.
6. **Prototipação:** As decisões de design são transformadas em protótipos tangíveis, que são versões simplificadas da solução. Com os elementos da interface definidos, realiza-se a prototipação, que pode variar da baixa à alta fidelidade, para testar a composição visual, validar hipóteses e simular a interação com baixo custo.
7. **Avaliação e Resultado:** Após os ciclos de prototipação, ocorre a fase de Avaliação, na qual o protótipo é submetido a uma análise formal utilizando a Técnica Eureka e a Lista Eureka. O objetivo é identificar problemas e oportunidades de melhoria. O Resultado de todo o processo é a entrega do artefato de software, juntamente com o relatório de avaliação que guia os refinamentos finais.

A aplicação desta metodologia foi um pilar para o projeto Avalista, assegurando que a plataforma fosse integralmente desenhada a partir das necessidades e desafios reais vivenciados por estudantes e professoras no processo de avaliação de interfaces.

2.4 Engenharia de Requisitos

A Engenharia de Requisitos (ER) é a abordagem sistemática e disciplinada para a especificação e o gerenciamento de requisitos, com o objetivo de compreender os desejos e as necessidades dos stakeholders e minimizar o risco de entregar um sistema que não atenda a essas expectativas (GLINZ et al., 2021). No centro deste processo está o conceito de requisito, que, segundo os mesmos autores, pode ser entendido de três formas: 1) uma necessidade percebida por um stakeholder; 2) uma capacidade ou propriedade que um sistema deve ter; ou 3) uma representação documentada dessa necessidade ou capacidade.

Os requisitos são classicamente divididos em três tipos principais:

- **Requisitos Funcionais (RF):** Dizem respeito a um resultado ou comportamento que deve ser fornecido por uma função do sistema, incluindo requisitos de dados ou de interação com seu ambiente. Essencialmente, definem "o que" o sistema deve fazer.
- **Requisitos de Qualidade (ou Não Funcionais - RNF):** Dizem respeito a questões de qualidade que não são cobertas pelos requisitos funcionais, como desempenho, disponibilidade, segurança e, de forma crucial para este trabalho, a usabilidade. A norma ABNT NBR ISO/IEC 25010 (2011) detalha a usabilidade em subcaracterísticas como apreensibilidade, operacionalidade e estética da interface.
- **Restrições:** São requisitos que limitam o espaço da solução, indo além do que é necessário para atender aos requisitos funcionais e de qualidade, como a obrigatoriedade de usar uma determinada tecnologia.

Além da classificação, a qualidade da especificação de um requisito é fundamental. Glinz et al. (2021) destacam que um requisito bem formulado deve ser inequívoco (ter uma única interpretação), completo (conter toda a informação necessária), consistente (não contradizer outros requisitos) e verificável (ser possível testar se foi implementado). A aplicação desses critérios durante a documentação dos requisitos do Avalista foi um passo crucial para evitar ambiguidades e garantir que as Histórias de Usuário refletissem de forma clara as necessidades do projeto.

2.4.1 Histórias de Usuário

Após a definição do problema e da solução, é necessário traduzir os requisitos do projeto em itens de desenvolvimento gerenciáveis. No contexto das metodologias ágeis, uma das técnicas para esse fim são as **Histórias de Usuário** (em inglês, *User Stories*).

Segundo Mike Cohn (2004), um dos principais expoentes do tema, uma história de usuário é uma descrição curta e simples de uma funcionalidade, contada a partir da perspectiva da pessoa que deseja a nova capacidade, geralmente um usuário ou cliente do sistema. Elas não são especificações técnicas detalhadas, mas sim um lembrete para uma conversa futura entre a equipe de desenvolvimento e o cliente, garantindo que o foco permaneça no valor entregue ao usuário final.

Para complementar a história e torná-la testável, são definidos os Critérios de Aceitação (*Acceptance Criteria*). Eles descrevem as condições específicas que a funcionalidade deve satisfazer para ser considerada "concluída". Uma forma comum de escrever esses critérios é o formato Given-When-Then (Dado-Quando-Então), que descreve um cenário de uso e seu resultado esperado, uma prática derivada do *Behavior-Driven Development* (BDD) e defendida como essencial para a clareza dos requisitos (COHN, 2004).

A seguir, um exemplo prático de uma história de usuário completa, que ilustra o formato utilizado no planejamento do projeto Avalista:

HU000 - Criar projeto

Como um usuário autenticado, **Eu quero** criar um novo projeto **Para que** eu possa organizar e visualizar as avaliações.

Critérios de Aceitação:

- **Dado que** estou logado no sistema, **quando** eu clicar no botão "Novo Projeto" na dashboard, **então** o sistema deve exibir o formulário de criação de projeto.
- **Dado que** preenchi os campos "Nome do Projeto" e "Descrição", **quando** eu clicar em "Salvar", **então** o projeto deve ser criado, salvo no banco de dados e exibido na lista de projetos.
- **Dado que** o nome do projeto já exista, **quando** eu tentar salvar, **então** o sistema deve impedir a criação e mostrar a mensagem de erro "Nome já em uso".
- **Dado que** eu não preencher o campo obrigatório "Nome do Projeto", **quando** eu clicar em "Salvar", **então** deve aparecer a validação *inline* "Este campo é obrigatório".

No projeto Avalista, a utilização de histórias de usuário com seus respectivos critérios de aceitação foi fundamental para documentar os requisitos e dividir o escopo completo da plataforma em entregas menores e incrementais, organizadas em *sprints*.

2.5 Tecnologias utilizadas no trabalho

O desenvolvimento da interface do Avalista foi sustentado por um ecossistema de tecnologias escolhidas para promover produtividade, manutenibilidade e agilidade no desenvolvimento. A arquitetura de software se apoia em uma base consolidada de linguagens web, apoiada por uma biblioteca de interface declarativa e um conjunto de ferramentas que compõem um conjunto de ferramentas de desenvolvimento atuais.

2.5.1 React: A Biblioteca para Interfaces de Usuário Declarativas

Para a construção da interface do Avalista, a tecnologia central escolhida foi o *React*, uma biblioteca JavaScript de código aberto para a criação de interfaces de usuário, mantida pela Meta (META PLATFORMS, INC., 2024). A hegemonia do *React* no desenvolvimento *frontend* moderno se deve a paradigmas que foram cruciais para este projeto:

- **Arquitetura Baseada em Componentes:** O *React* encoraja a construção de interfaces através da composição de peças de UI isoladas e reutilizáveis, chamadas "componentes". Cada componente gerencia seu próprio estado e lógica, e eles podem ser combinados para construir UIs complexas. Esta abordagem, delineada na própria documentação oficial da biblioteca como "Pensar em *React*", é um dos pilares da tecnologia (META PLATFORMS, INC., 2024b).
- **Virtual DOM:** Para otimizar o desempenho, o *React* utiliza um "DOM Virtual", uma representação do DOM real do navegador em memória. Quando o estado de um componente muda, o *React* primeiro atualiza essa representação virtual, calcula de

forma otimizada a diferença entre a versão antiga e a nova (processo conhecido como "diffing"), e só então atualiza o mínimo necessário no DOM real (GACKENHEIMER, 2019).

2.5.2 TypeScript: Robustez com Tipagem Estática

Para trazer maior robustez, escalabilidade e manutenibilidade ao código, o projeto utiliza *TypeScript*, um superconjunto de JavaScript desenvolvido pela Microsoft. A principal vantagem do *TypeScript* é a adição de um sistema de tipagem estática opcional (MICROSOFT, 2024). Na prática, dentro do Avalista, isso significa que:

- **Prevenção de Erros:** Muitos erros comuns de programação são detectados durante o desenvolvimento, antes mesmo de o código ser executado. Por exemplo, o *TypeScript* impede que um componente `<ProjectCard />` seja renderizado sem receber as propriedades obrigatórias que ele espera, como `id` e `name`.
- **Clareza e Documentação:** A definição de interfaces e `types` serve como um contrato claro e uma documentação viva dos formatos de dados que transitam entre o *frontend* e o *backend*.
- **Melhora na Experiência do Desenvolvedor:** O sistema de tipos fornece um autocompletar mais inteligente e uma refatoração de código mais segura dentro do editor, tornando o desenvolvimento de funcionalidades complexas mais ágil e menos suscetível a bugs.

2.5.3 Vite: O Ambiente de Desenvolvimento Moderno

Todo o projeto é servido e construído utilizando o *Vite*, uma ferramenta de *build* de última geração. Diferente de ferramentas mais antigas, o *Vite* utiliza módulos ES nativos do navegador durante o desenvolvimento, o que resulta em um tempo de inicialização do servidor e de atualizações (*Hot Module Replacement* - HMR) ordens de magnitude mais rápido. Essa velocidade proporciona um ciclo de feedback imediato para o desenvolvedor, aumentando a produtividade. Para a *build* de produção, o *Vite* utiliza o *Rollup*, um empacotador, para gerar arquivos estáticos pequenos e eficientes (VITE, 2024).

2.5.4 Ecossistema e Bibliotecas de Suporte

Além das tecnologias base, o Avalista se apoia em um ecossistema de bibliotecas especializadas para resolver problemas comuns:

- **React Router DOM:** É a biblioteca padrão para gerenciamento de rotas em aplicações de página única (SPA) construídas com *React*. Ela permite a navegação entre diferentes visualizações da aplicação de forma declarativa e sem a necessidade de recarregar a página inteira (REMIX SOFTWARE, 2024).
- **Axios:** Um cliente HTTP baseado em Promises, utilizado para realizar requisições à API do projeto a partir do navegador. Facilita o envio e recebimento de dados, o

tratamento de erros e a configuração de cabeçalhos de requisição (THE AXIOS PROJECT, 2024).

- **React Flow:** Uma biblioteca de alta performance para a construção de interfaces e editores baseados em nós e grafos. Foi a tecnologia escolhida para criar o organograma interativo da Lista Eureka, oferecendo controle total sobre a renderização e o comportamento dos elementos (XYFLOW, 2024).
- **Recharts:** Uma biblioteca de gráficos "composável", construída sobre componentes *React*. Foi utilizada no Avalista para criar visualizações de dados complexas, como o gráfico de radar na página de detalhes do projeto, que permite comparar o desempenho de diferentes avaliadores (RECHARTS, 2024).
- **Lucide React:** Uma biblioteca de ícones SVG de código aberto, leve e de fácil customização. Oferece um conjunto de ícones limpos e consistentes que foram integrados em todo o projeto para melhorar a comunicação visual da interface (THE LUCIDE TEAM, 2024).

2.5.5 Arquitetura Baseada em Funcionalidades (Feature-Based)

Além das tecnologias, a forma como o código-fonte é organizado é crucial para o sucesso de um projeto de software. Uma abordagem moderna para estruturar projetos de *frontend* é a Arquitetura Baseada em Funcionalidades (*Feature-Based Architecture*). Este padrão, derivado de metodologias ágeis como o *Feature-Driven Development* (FDD), propõe que a estrutura do código reflita as funcionalidades visíveis do sistema, em vez de agrupá-lo por tipos técnicos de arquivos (PALMER; FELSING, 2002).

Diferente de modelos tradicionais que agrupam arquivos por tipo (por exemplo, uma pasta para todos os componentes, outra para todos os serviços), este padrão organiza o código em torno das funcionalidades do sistema. Cada funcionalidade (como "autenticação", "gestão de projetos" ou "avaliação") possui sua própria pasta, que contém todos os arquivos necessários para seu funcionamento: componentes, lógica, estilos e testes.

Os principais benefícios dessa abordagem são a escalabilidade e a manutenibilidade. Torna-se mais fácil adicionar, remover ou dar manutenção a uma funcionalidade sem causar impacto em outras partes do sistema, pois todo o seu código está encapsulado. Essa organização também torna o projeto mais intuitivo, pois a estrutura de pastas reflete diretamente as capacidades visíveis da plataforma.

2.6 Trabalhos Correlatos

A iniciativa de criar ferramentas para apoiar a avaliação heurística não é inédita, e a análise de soluções existentes é fundamental para posicionar um novo projeto. Uma solução de referência na área é o Heurix, uma aplicação que guia o avaliador através das heurísticas de Nielsen utilizando formulários, permitindo adicionar comentários, capturar telas e gerar relatórios automatizados. O Heurix serve como uma importante validação da viabilidade técnica de se sistematizar o processo avaliativo.

Contudo, uma análise crítica revela que o Heurix, embora funcional, possui um escopo que não atende a desafios centrais do contexto educacional e de engajamento que motivaram a criação do Avalista. As principais limitações identificadas são:

- **Rigidez das Heurísticas:** A ferramenta é estruturada em torno das 10 heurísticas de Nielsen, apresentando pouca ou nenhuma flexibilidade para a adoção de listas personalizadas ou em evolução, como a Lista Eureka (2024). Isso a torna inadequada para contextos que utilizam diretrizes próprias e adaptadas.
- **Ausência de Foco Pedagógico:** O Heurix pressupõe que o usuário já é um especialista. Não há um componente educacional que ensine ou reforce os conceitos por trás das diretrizes, um dos pilares da proposta do Avalista, que visa também ser uma ferramenta de aprendizagem.

Essas lacunas identificadas na principal ferramenta de referência reforçam a relevância e o caráter inovador do Avalista, que foi concebido justamente para unir a funcionalidade de um sistema de avaliação com as camadas de engajamento, flexibilidade e pedagogia.

3 Desenvolvimento

Este capítulo detalha a jornada prática de construção da plataforma Avalista. Sua estrutura espelha as três grandes fases da metodologia adotada: Fundamentação e Pesquisa, Concepção e Design e Execução e Entrega. Primeiramente, são apresentados os resultados da fase de investigação, que culminaram no diagnóstico do problema e na definição dos requisitos da plataforma. Em seguida, descreve-se o processo de concepção e design, que traduziu a pesquisa em uma identidade visual e um plano de funcionalidades. Por fim, a fase de execução detalha a arquitetura técnica, o sistema de design e o escopo do *frontend* implementado.

3.1 Fundamentação e Pesquisa

3.1.1 Apresentação e Análise da Pesquisa com Estudantes

A primeira etapa do diagnóstico do problema consistiu na aplicação de um formulário com 59 estudantes de Tecnologia em Análise e Desenvolvimento de Sistemas do IFRN. O objetivo era de entender a percepção e a experiência de uma parcela de alunos com o processo de avaliação de interfaces por inspeção realizado de forma manual. A análise das respostas revelou os seguintes pontos:

- **Alto Valor Percebido:** A grande maioria dos estudantes (cerca de 85% da amostra) que tiveram experiência com a avaliação consideraram a prática "importante" ou "extremamente importante" para a melhoria de um sistema, atribuindo notas entre 8 e 10. Isso confirma que o valor pedagógico da atividade é reconhecido.
- **Dificuldade:** Apesar de reconhecerem a importância, os participantes relataram um grau de dificuldade. A nota média de dificuldade para realizar a avaliação foi de 6 em 10, e uma parcela expressiva de 70% dos estudantes afirmou que precisou de auxílio para conseguir completar a tarefa.
- **Processo Cansativo e Demorado:** O tempo de avaliação variou significativamente, de 1 a até 16 horas, e termos como "cansativo", "trabalhoso" e "repetitivo" foram frequentemente utilizados nas respostas abertas para descrever o processo manual.
- **Impacto Positivo na Formação:** Mesmo com as dificuldades, os relatos qualitativos reforçaram o impacto positivo da experiência na formação do olhar crítico dos estudantes. Como relatado por um dos participantes:

"Foi a partir desta avaliação que eu compreendi o verdadeiro significado e importância do design para o desenvolvimento em geral [...] consigo ainda hoje detectar falhas em produtos que tornaram meu olhar mais crítico."

Outro estudante complementou a visão sobre a dificuldade do processo:

"Foi uma experiência boa, mas é consideravelmente difícil avaliar pois tem que ter muita atenção aos detalhes."

3.1.2 Problema Identificado

A fase de diagnóstico, embasada na pesquisa com 59 estudantes, confirmou o valor da avaliação heurística como ferramenta pedagógica, mas também revelou uma série de desafios operacionais e culturais que dificultam sua aplicação eficaz no contexto acadêmico. O problema não reside no método em si, mas nas barreiras impostas pelo seu processo manual.

Os principais pontos de atrito identificados, que o projeto Avalista busca solucionar, foram:

- **Falta de Sistematização:** A ausência de uma ferramenta padronizada leva a avaliações desorganizadas, dificultando o registro consistente dos problemas, a colaboração entre avaliadores e a posterior consulta dos dados.
- **Dificuldade do Processo Manual:** A execução manual da avaliação exige uma alta carga cognitiva do estudante. O avaliador precisa alternar constantemente entre a interface que está sendo analisada, o documento com a lista de diretrizes (Lista Eureka) e uma terceira ferramenta para registrar os problemas, como um editor de texto ou uma planilha. Essa fragmentação do fluxo de trabalho não só torna o processo repetitivo e cansativo, mas também abre margem para inconsistências no registro dos dados e perda de foco na análise crítica, que é o verdadeiro objetivo da atividade.
- **Dificuldade em Transformar Dados em Ação:** Talvez o maior obstáculo seja a dificuldade de converter os relatórios de avaliação em ações de melhoria práticas e priorizadas. A falta de métricas claras sobre a gravidade e o esforço de correção impede que as equipes de desenvolvimento saibam por onde começar.

3.1.3 Requisitos da Plataforma

Com base na análise do problema, os requisitos da plataforma foram levantados e priorizados utilizando a metodologia MoSCoW (Must Have, Should Have, Could Have, Won't Have), que os classifica em Essenciais, Importantes, Desejáveis e Opcionais. A seguir, são detalhados os Requisitos Funcionais (RF) que descrevem o que o sistema deve fazer.

Tabela 1 – Requisitos Funcionais do Avalista

Código	Nome	Descrição	Prioridade
RF01	Autenticação	O sistema deve permitir efetuar login e cadastro de usuários com diferentes permissões (Administrador, Avaliador).	Essencial
RF02	Adição de Projetos	O sistema deve permitir criar, editar e excluir projetos, contendo nome, descrição e marca gráfica.	Essencial

RF03	Gerenciar Telas do Projeto	O sistema deve permitir que o usuário adicione, edite e exclua as telas (imagens) a serem avaliadas dentro de um projeto.	Essencial
RF04	Upload de Imagens	O sistema deve permitir o upload das imagens que representam as telas do projeto a ser avaliado.	Essencial
RF05	Categorizar Violações	O sistema deve permitir que o avaliador associe violações encontradas às diretrizes da Lista Eureka.	Essencial
RF06	Estimar o Tempo de Resolução	O sistema deve permitir que o avaliador estime, em horas e minutos, o tempo necessário para corrigir cada problema.	Essencial
RF07	Avaliação de Esforço	O sistema deve permitir que o avaliador estime o esforço necessário para corrigir cada problema.	Essencial
RF08	Proposta de Melhoria	O sistema deve permitir que o avaliador descreva sugestões de melhorias para resolver os problemas identificados.	Essencial
RF09	Avaliação por Múltiplos Avaliadores	O sistema deve permitir que de 3 a 5 avaliadores avaliem o mesmo projeto de forma independente.	Essencial
RF10	Consolidação de Avaliações	O sistema deve permitir que os avaliadores, após suas avaliações individuais, consolidem os resultados em uma única avaliação final.	Essencial
RF11	Identificação de Erros Comuns	O sistema deve permitir a identificação dos erros que foram encontrados por mais de um avaliador.	Essencial

RF12	Resumo da Avaliação Consolidada	O sistema deve gerar um resumo da avaliação final, indicando os erros consensuais, erros únicos e soluções propostas.	Essencial
RF13	Controle de Status da Avaliação	O sistema deve permitir que um projeto tenha status como: "Em Avaliação Individual", "Em Consolidação" e "Avaliação Consolidada".	Essencial
RF14	Coleta de Métricas do Projeto	O sistema deveria coletar métricas detalhadas por projeto (ex: diretrizes mais violadas, tempo/esforço total).	Importante
RF15	Coleta de Métricas do Avaliador	O sistema deveria coletar métricas detalhadas do avaliador (ex: projetos com maior tempo de resolução, categoria mais comum).	Importante
RF16	Visualização de Métricas	O sistema deveria permitir que o avaliador visualize as métricas coletadas em gráficos e classificações.	Importante
RF17	Resolver a Diretriz	O sistema poderia permitir que o avaliador marque uma diretriz como resolvida e faça upload de uma foto mostrando a resolução.	Desejável
RF18	Seção Educacional	O sistema deveria oferecer uma sessão educativa com materiais sobre princípios de design de interfaces.	Opcional

Fonte - Autoria Própria (2025)

Além das funcionalidades, foram definidos os Requisitos Não Funcionais (RNF), que descrevem os atributos de qualidade e as restrições operacionais do sistema, buscando uma boa experiência de uso.

Tabela 2 – Requisitos Não Funcionais do Avalista

Código	Nome	Descrição	Categoria	Classificação
--------	------	-----------	-----------	---------------

RNF01	Segurança dos Dados	As informações armazenadas devem ser protegidas com autenticação e criptografia.	Segurança	Obrigatório
RNF02	Tempo de Resposta	O sistema deve carregar os projetos em menos de 5 segundos.	Eficiência e Performance	Desejável
RNF03	Ambiente de funcionamento	O sistema deve funcionar nos navegadores mais usados, como Chrome, Firefox e Edge.	Compatibilidade	Essencial
RNF04	Qualidade da Interação	A interface deve proporcionar uma experiência de uso positiva, sendo não apenas eficiente e eficaz, mas também clara em sua comunicação e propósito, buscando a satisfação do avaliador.	Usabilidade	Essencial

Fonte - Autoria Própria (2025)

3.2 Concepção e Design

Com os requisitos definidos, a fase de ideação focou em traduzi-los em estratégias de design, uma identidade forte e uma interface funcional e que veio do Laboratório de Problemas (MATOS, 2023).

3.2.1 Laboratório de Problemas

A construção da plataforma Avalista teve início com a apuração de requisitos junto às professoras Silvia Aparecida de Oliveira de Alencar Matos, Marília Aranha Freire, Isis Duarte e Ramon Vieira, coautores da Lista Eureka 2024 (MATOS et al., 2024). Ambas as professoras atuaram como clientes e especialistas no processo.

Para guiar o projeto, foi utilizado o método Laboratório de Problemas, adaptado da tese "Roda de Fuxico" (MATOS, 2023), que se organiza no seguinte fluxo de sete etapas:

1. **Busca de Problemas:** Imersão no contexto do projeto de pesquisa Eureka para identificar as dores e oportunidades no processo de avaliação manual de interfaces.
2. **Coleta de Dados:** Aplicação de um formulário com 59 estudantes e levantamento de trabalhos correlatos para aprofundar o conhecimento sobre o problema.
3. **Análise:** Organização dos dados coletados em textos-síntese, palavras-chave e uma análise comparativa de ferramentas similares.
4. **Interpretação:** Construção do Quadro de Atributos e tradução dos *insights* nas Decisões Projetuais que nortearam a solução.
5. **Desenho:** Criação da identidade visual completa da plataforma, incluindo o símbolo, a marca gráfica, a tipografia e a paleta de cores.
6. **Prototipação:** Realização de sessões colaborativas com as orientadoras para a criação de esboços e protótipos de baixa fidelidade em quadro branco, validando o fluxo de uso e a estrutura das telas.
7. **Avaliação e Resultado:** Validação contínua dos artefatos de design com as especialistas e, por fim, o desenvolvimento do *frontend* da plataforma como resultado prático do processo.

3.2.1.1 Quadro de Atributos e Decisões Projetuais

A fase de Análise e Interpretação culminou na construção de um quadro de atributos para mapear o ecossistema do projeto (cliente, usuários, contexto), que serviu como base para as decisões estratégicas.

Tabela 3 - Quadro de Atributos

Atributos	Similaridade	Diferenciação	Problemas	Desejos	Memória Afetiva	Críticas
Cliente	Organização das Avaliações	Gameificação	Queda na procura do conhecimento	Linguagem local ou regional	Evolução das interfaces	A ferramenta não pode ser utilizada para

			o		na disciplina	limitar o aprendizado dos estudantes
Sujeitos	Divisão de tarefas	Unificar o conteúdo	Avaliação manual e maçante, sem padronização	Praticidade	Melhora contínua das interfaces	Avaliações cansativas, repetitivas
Oponente	Gamificação	Informação clara e objetiva	Dificuldade desconforto em expor críticas	Processo automatizado	Cansaço e indiferença em realizar avaliações	Não vê sentido em realizar avaliações e possui resistência a elas
Designer	Lista de Nielsen	Atualização com os sistemas atuais	Dificuldade em analisar os problemas	Unificação das ferramentas	Encontrar vários pontos de melhoria	Trabalho repetitivo e longo
Contexto	Ferramentas Anteriores	Mudança na cultura de avaliação	Resistência dos desenvolvedores pesquisados	Integração com plataformas já utilizadas	Experiências frustradas com sistemas antigos	Avaliações subjetivas e inconsistentes; desvalorização do processo

Fonte - Adaptado de Matos (2023)

Com a conclusão da fase de Análise e Interpretação do Laboratório de Problemas, os dados e percepções sintetizados no Quadro de Atributos serviram de base para a etapa de Ideação. O desafio seguinte foi traduzir esse diagnóstico aprofundado em diretrizes concretas que pudessem nortear o design da plataforma Avalista.

Cada estratégia foi concebida como uma resposta direta a um ou mais problemas e desejos identificados no quadro. Por exemplo, a constatação de que o processo era uma "avaliação manual e maçante, sem padronização" fundamentou a estratégia de Simplificação, que visa criar um fluxo de trabalho padronizado e mais eficiente. A concepção do projeto também considerou o desejo por uma "mudança na cultura de avaliação", que futuramente será potencializada pela gamificação para aumentar o engajamento.

As decisões resultantes, que formaram o alicerce para a concepção da marca, dos protótipos e das funcionalidades, estão detalhadas na tabela a seguir:

Tabela 4 - Decisões Projetuais

Estratégia	Decisões	Justificativa
------------	----------	---------------

Gamificação	Sistema de recompensas para estimular a prática de avaliações heurísticas	Recompensas como selos ou pontuação motivam os usuários a realizar mais avaliações
Colaboração	Espaço para reportar erros que não pertencem a nenhuma diretriz da lista	Estimula o feedback contínuo dos usuários, contribuindo para a evolução colaborativa da lista
Simplificação	Facilitar a avaliação, padronizando e organizando a formatação das avaliações	Facilita a leitura e preenchimento das avaliações, tornando o processo mais eficiente
Aprendizagem Ativa	Espaço de aprendizagem, contendo conceitos, dicas e conhecimento	Ajuda o usuário a entender melhor as diretrizes e aplicar corretamente as heurísticas
Transparência	Tornar visível o impacto das avaliações realizadas pelos usuários	Avaliação visíveis e quantificáveis A visibilidade das métricas permite que os usuários quantifiquem o impacto de suas avaliações e compreendam como suas contribuições ajudam a melhorar as diretrizes

Fonte - Adaptado de Matos (2023)

3.2.2 Criação da Identidade Visual

A criação da identidade visual da plataforma foi uma etapa estratégica do design, resultando em um conjunto de elementos gráficos que representam o propósito do projeto: o símbolo, o logotipo, a tipografia e a paleta de cores.

3.2.2.1 Símbolo

Para quebrar a seriedade do processo de avaliação, foi criado o Avalista, um símbolo que personifica a ferramenta. Trata-se de um papagaio nordestino, falante e perspicaz, que observa as interfaces com um olhar crítico, mas de forma bem-humorada. O ícone do papagaio foi adaptado a partir de um vetor de licença livre, cujos traços e cores foram modificados para se alinharem à identidade do projeto. A narrativa do personagem, descrita abaixo, reforça a ideia de que a avaliação pode ser feita com leveza e engajamento:

"Sou o avalista: um papagaio resistente. Resisto à falação porque sou falante. Resisto ao desconforto das críticas, porque sou forte, assim como o meu bico, que é forte e curvo. Com ele eu quebro sementes e frutos que chegam no meio do caminho. Aprendi que só com um olho na lista e outro na interface que conseguimos evoluir. Interface pronta é interface morrida. Sou brasileiro, nordestino. Sou bonito, alegre, verde, amarelo, laranja e azul. Falo o que penso, doa a quem doer. Sou amostradinho e não ligo para mufinagem. Aprendi com as pesquisas EURECAS - lista e técnica - a encontrar problemas nas interfaces. aprendi a imitar

a lição das diretrizes direitinho. Faço avaliações em forma de jogo. Como dizem por aí: sou gueimer. E sou bem mais ligeiro. Não é agilidade que o povo quer?"

3.2.2.2 Logotipo e Tipografia.

A logo da plataforma é formado pela combinação do símbolo do papagaio com o nome "Avalista". Para a construção do logotipo, foi utilizada a fonte *Baumans*, selecionada do catálogo do Google Fonts por suas formas arredondadas e estilo distinto, que complementam a estética do símbolo. A tipográfica do projeto foi dividida em duas frentes:

- Tipografia Principal: A fonte *Baumans* é utilizada exclusivamente no logotipo e em materiais de comunicação da marca.
- Tipografia Acessório: Para garantir legibilidade e uma estética moderna na plataforma, foram selecionadas as fontes *Manrope* para títulos e *DM Mono* para o corpo de texto, ambas também do catálogo do Google Fonts.

3.2.2.3 Grade Construtiva e Proporções

Para garantir a consistência, o alinhamento e o equilíbrio visual da marca em diferentes aplicações, foi definida uma grade construtiva modular. As proporções entre o símbolo (o papagaio) e o logotipo (o nome "Avalista") foram estabelecidas com base em uma unidade de medida "X", derivada da dimensão do próprio símbolo.

Na versão vertical da marca gráfica, o símbolo ocupa uma altura proporcional a X, enquanto o logotipo abaixo dele ocupa uma altura de 2X, resultando em uma composição total com altura de 3X, conforme ilustrado na Figura 2. Essa proporção foi escolhida para dar o devido peso a ambos os elementos quando empilhados.

Já na versão horizontal, a proporção foi baseada na largura dos elementos. O símbolo do papagaio ocupa uma largura de X, e o logotipo ao seu lado ocupa uma largura equivalente, também de X. A marca gráfica completa, portanto, possui uma largura total de 2X, sem contar o espaçamento de respiro entre os elementos, como demonstrado na Figura 2.

Figura 2 – Grade Construtiva e Proporções



Fonte: Autoria Própria (2025)

3.2.2.4 A Paleta de Cores Mutante

Uma identidade visual não precisa ser estática. Conforme aponta Kreutz (2007), a identidade visual é uma estratégia que reflete o contexto sócio-histórico, e a contemporaneidade é marcada pela mutação e pela velocidade. Nesse cenário, surge o conceito de Identidade Visual Mutante, caracterizada por ser aberta, inovadora e dinâmica, em oposição às identidades convencionais, que são mais rígidas. A autora distingue dois tipos principais de identidade mutante:

- Mutante Programada: Variações que ocorrem de forma controlada, por um tempo ou contexto determinado.
- Mutante Poética: Variações que ocorrem de forma espontânea, sem regras pré-determinadas, seguindo um intuito mais criativo.

Seguindo esse princípio, a identidade visual do Avalista foi concebida como uma Identidade Visual Mutante Programada. A "programação" da mutação está diretamente ligada à funcionalidade principal da plataforma: a cor do símbolo muda de acordo com a categoria da Lista Eureka que está sendo avaliada. Essa abordagem não é meramente estética, mas uma decisão estratégica de design. Ela permite que a interface se adapte visualmente ao contexto da avaliação, reforçando para o usuário em qual categoria de diretrizes ele está trabalhando e

melhorando a organização do processo. O quadro a seguir detalha a paleta de cores e sua associação programada com as categorias.

Tabela 5 – Paleta de Cores Associada às Categorias da Lista Eureka

Categoria (código)	Cor Principal (Hex)	Cor Secundária (Hex)
Aspectos Funcionais (AF)	#070248	#CECDDF
Comunicação (CO)	#36A08E	#D5FFF9
Formatação (FM)	#4DAA00	#E1FCCF
Navegação (NA)	#D57C19	#FFE6C5
Particularidades do Usuário (PU)	#C61819	#FFD4D4
Particularidades do Dispositivo (PD)	#782A53	#FBDAEC
Acessibilidade (AC)	#4C85C7	#C7E2FF
Privacidade e Segurança de Dados (LGPD)	#9C1DFF	#D9D2E9

Fonte: Autoria Própria (2025)

3.2.2.5 Sistema de Assinaturas Visuais

Uma identidade visual eficaz não se resume a um único logotipo estático, mas a um sistema flexível capaz de se adaptar a diferentes contextos sem perder sua essência. Conforme defende a designer e educadora Ellen Lupton (2016), uma identidade moderna funciona como um sistema, no qual os elementos (logotipo, cor, tipografia) são projetados para serem combinados de várias maneiras, criando um conjunto de ferramentas visuais em vez de uma regra rígida.

Seguindo esse princípio, foi desenvolvido para o Avalista um sistema completo de assinaturas visuais, garantindo sua correta e versátil aplicação. O sistema foi estruturado com base em três eixos de variação:

- **Variações de Layout:** Foram criadas uma versão horizontal (assinatura primária) e uma vertical do logotipo. A versão horizontal é a principal, ideal para aplicações em layouts largos, como cabeçalhos de sites. A versão vertical funciona como uma alternativa para contextos mais compactos ou quadrados, como posts em redes sociais ou materiais impressos específicos, garantindo flexibilidade sem comprometer a integridade da marca.
- **Variações de Cor:** O sistema cromático foi projetado para ser tanto temático quanto versátil. Além das oito variações de cor baseadas na paleta das categorias da Lista Eureka foram criadas três versões neutras do logotipo (monocromática positiva/branca, negativa/preta e em tons de cinza). Essas versões garantem a

aplicação legível da marca sobre fundos fotográficos ou coloridos onde a paleta principal não seria adequada. Ao todo, a marca possui 11 variações cromáticas oficiais, assegurando sua correta aplicação em praticamente qualquer cenário.

Figura 3 – Variações da marca gráfica nas versões vertical



Fonte: Autoria Própria (2025)

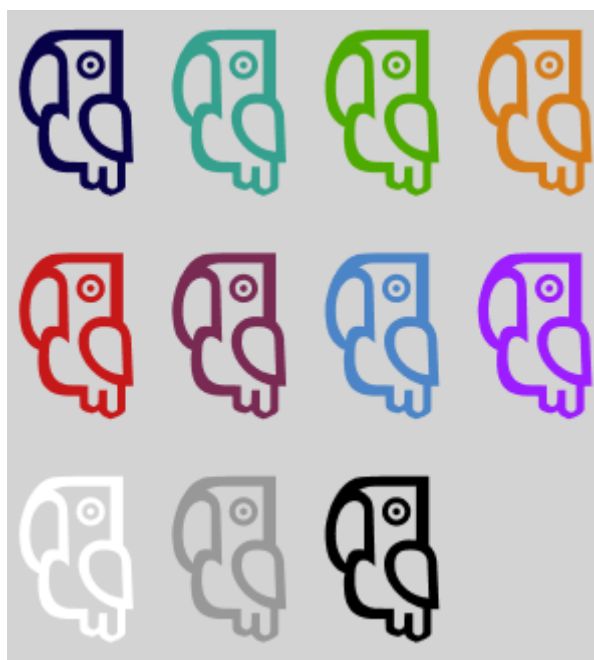
Figura 4 – Variações da marca gráfica nas versões horizontal



Fonte: Autoria Própria (2025)

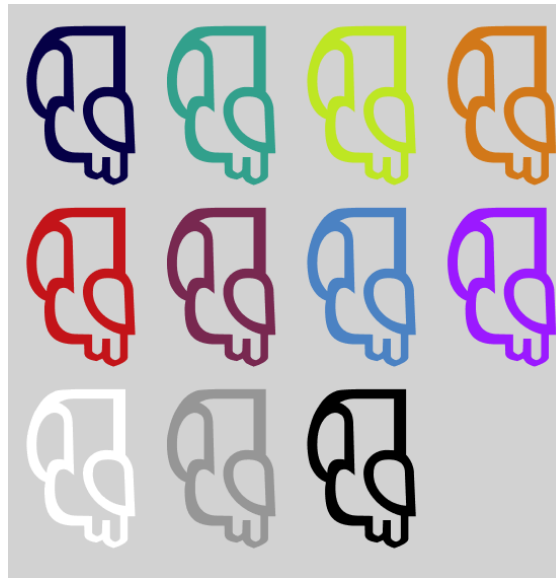
- **Variações de Complexidade:** Compreendendo que a marca seria aplicada em tamanhos muito distintos, o sistema foi projetado para ser escalável. Ele conta com o símbolo completo (o papagaio com o olho e todos os detalhes) para uso geral e um símbolo simplificado (o papagaio em silhueta, sem o olho) projetado especificamente para aplicações em tamanhos reduzidos, como favicons de navegador ou ícones de perfil em redes sociais.

Figura 5 – Variações do símbolo completo



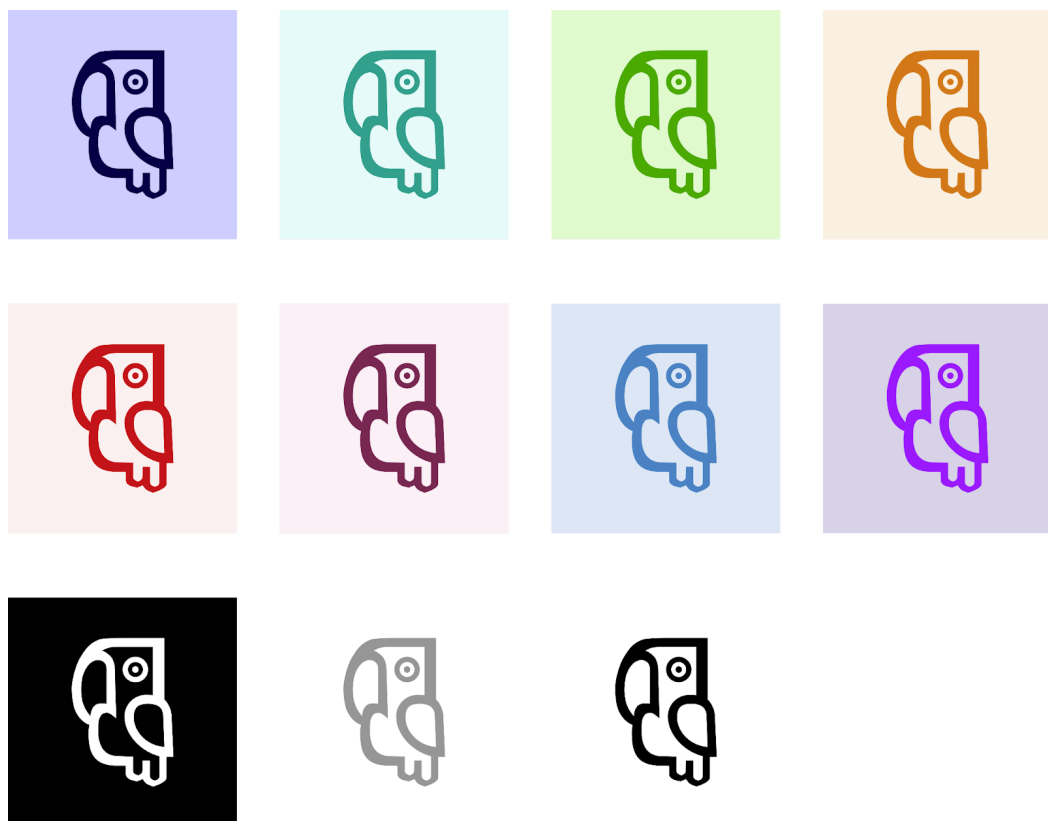
Fonte: Autoria Própria (2025)

Figura 6 – Variações do símbolo simplificado



Fonte: Autoria Própria (2025)

Figura 7 – Variações do símbolo como ícone para redes sociais



Fonte: Autoria Própria (2025)

3.3 Execução e Entrega

3.3.1 Planejamento das Histórias de Usuário

O escopo de desenvolvimento deste trabalho concentrou-se na implementação das funcionalidades essenciais que constituem a jornada principal do avaliador dentro da plataforma Avalista. As seguintes funcionalidades foram implementadas no *frontend*:

- **Módulo de Autenticação e Gestão de Projetos:** Foi desenvolvido o fluxo completo que permite ao usuário criar uma conta ([HU001](#)) e fazer login ([HU002](#)). Uma vez autenticado, o usuário é direcionado para um Dashboard que serve como painel central. A partir dali, ele pode criar um novo projeto de avaliação através de um modal ([HU004](#)), visualizar seus projetos em uma página dedicada com filtros ([HU005](#)) e acessar os detalhes de um projeto específico ([HU006](#)). Também foi implementada a interface para edição de projetos ([HU007](#)), adição de telas ([HU009](#)) e a gestão de membros da equipe ([HU008](#)), utilizando modais para manter o usuário no contexto da página.
- **Módulo de Avaliação Individual:** Foi implementado o coração da plataforma. Ao iniciar uma avaliação ([HU012](#)), o usuário é levado a uma nova tela onde a interface exibe a imagem da tela a ser analisada e uma ferramenta de organograma interativo. Este organograma, construído com a biblioteca *React Flow*, renderiza dinamicamente toda a hierarquia da Lista Eureka. O sistema permite que o usuário clique em uma diretriz para abrir um modal e adicionar um problema ([HU013](#)), com campos para descrição, sugestão de melhoria, classificação de gravidade e esforço, e o upload de uma imagem como evidência.

Este conjunto de funcionalidades garante a base operacional completa para uma avaliação individual, desde a configuração até o registro de problemas, estabelecendo o alicerce fundamental do sistema para futuras expansões.

Este conjunto de funcionalidades garante a base operacional para uma avaliação individual, desde a configuração até o registro e consulta de problemas, estabelecendo o alicerce fundamental do sistema para futuras expansões.

3.3.2 Sistema de Design e Estilização

Para materializar a identidade visual e garantir a consistência em toda a aplicação, foi desenvolvido um sistema de design fundamental, codificado no arquivo `design-system.css`. A criação de um sistema de estilos centralizado evita inconsistências. Para isso, foram criadas classes genéricas e reutilizáveis, como `.card` para painéis de conteúdo, um sistema de botões com modificadores como `.btn` e `.btn-primary`, e um conjunto de classes para formulários como `.form-group` e `.form-input`. Essa abordagem, combinada com o uso de variáveis CSS, permitiu a rápida prototipação e a manutenção da consistência visual em toda a aplicação.

Este sistema padroniza a aparência e o comportamento dos elementos mais recorrentes da interface. Foram estilizados cards (`.card`) para servirem como contêineres de conteúdo

padronizados; um sistema de botões (*.btn*) com variações de hierarquia visual (*.btn-primary*, *.btn-secondary*) e estados de interação (*:hover*, *:disabled*); um conjunto completo de estilos para formulários, incluindo rótulos (*.form-label*) e campos de texto (*.form-input*); e um sistema de mensagens de erro para comunicar validações ao usuário de forma consistente.

Tecnicamente, a implementação se apoia fortemente no uso de variáveis CSS, que herdam a paleta de cores "mutante" do projeto e permitem a manutenção centralizada de todo o estilo. A criação de classes reutilizáveis complementa a arquitetura de componentes do *React*, assegurando que um mesmo componente tenha uma aparência e comportamento previsíveis em qualquer parte da aplicação. A consistência tipográfica da interface foi garantida pela definição das fontes '*Manrope*' para títulos e '*DM Mono*' para o corpo de texto.

3.3.3 Decisões Arquiteturais

O *frontend* da plataforma Avalista foi desenvolvido utilizando a biblioteca *React*, escolhida por sua eficiência na criação de interfaces de usuário componentizadas e interativas. Para a organização do código, o projeto adotou o padrão *Feature-Based Architecture* (Arquitetura Baseada em Funcionalidades), cujos conceitos e benefícios foram detalhados na Fundamentação Teórica (Seção 2.5.5).

Para materializar esse padrão, o código-fonte do *frontend* foi organizado em uma estrutura de diretórios clara e coesa, conforme ilustrado na Figura 8.

Figura 8 – Estrutura de Diretórios do *frontend* do Avalista

```
/src
|
|-- /api
|   |-- apiClient.ts
|   |-- authApi.ts
|
|-- /assets
|
|-- /components
|   |-- /ui
|
|-- /features
|   |-- /auth
|   |-- /dashboard
|
|-- /hooks
|   |-- useDropdown.ts
|
|-- /mocks
|
|-- /pages
```

```
|  
|-- /routes  
|  
|-- /styles  
|   |-- index.css  
|   |-- design-system.css  
|  
|-- /types  
|-- /utils  
|  
--- App.tsx  
--- main.tsx
```

Fonte: Autoria Própria (2025)

Nesta estrutura, cada diretório possui uma responsabilidade bem definida: a pasta */features* é o coração da arquitetura, encapsulando as funcionalidades completas do sistema (como autenticação e o *dashboard*); a */components/ui* armazena componentes de interface puros e reutilizáveis (botões, *inputs*); a */pages* é responsável por montar as telas da aplicação, compondo os componentes das *features*; e diretórios como */api*, */routes* e */types* gerenciam, respectivamente, a comunicação com o *backend*, a navegação da aplicação e as tipagens do TypeScript.

3.3.4 Integração com o Backend

Uma etapa fundamental do desenvolvimento foi a integração contínua com a API do *backend*, desenvolvida em paralelo pelo colaborador Ramon Vieira. Para garantir a comunicação eficaz entre as camadas, foi utilizada a biblioteca *Axios* para gerenciar as requisições HTTP. Foram implementadas as chamadas para os endpoints de autenticação (login e cadastro), bem como para as operações de CRUD (Criar, Ler, Atualizar e Deletar) de projetos e telas. A utilização do *TypeScript* para definir as interfaces dos dados que transitam entre *frontend* e *backend* foi crucial para garantir um contrato claro entre as duas partes e prevenir erros de integração.

Este conjunto de funcionalidades e integrações garante a base operacional completa para uma avaliação individual, desde a configuração inicial até o registro de problemas, servindo como uma robusta prova de conceito da solução completa.

4 Resultados

Este capítulo apresenta os resultados consolidados do projeto, estruturados de acordo com as fases da metodologia adotada. O percurso se inicia com os achados da investigação e os artefatos de design resultantes, e culmina na análise aprofundada da interface interativa do *frontend* da plataforma Avalista. As principais funcionalidades da plataforma são detalhadas, conectando o software desenvolvido aos objetivos do trabalho e à solução para o problema investigado.

4.1 Investigação e Concepção

A fase inicial da metodologia gerou como principal resultado a validação da necessidade de uma ferramenta de apoio automatizado, a partir da análise do contexto e da pesquisa com 59 estudantes. Conforme detalhado na Seção 3.1.1, a investigação confirmou que o processo manual de avaliação heurística, embora reconhecido como valioso, enfrenta desafios operacionais e culturais significativos. Para recordar, os principais pontos de atrito identificados foram:

- **Falta de Sistematização:** A ausência de uma ferramenta padronizada resulta em avaliações desorganizadas e dificulta o registro consistente dos problemas.
- **Resistência à Prática:** O caráter repetitivo e manual do processo gera uma resistência motivacional nos estudantes, que percebem a atividade como cansativa.
- **Dificuldade em Transformar Dados em Ação:** A falta de métricas claras sobre a gravidade e o impacto dos problemas impede que os achados da avaliação sejam facilmente transformados em um plano de melhorias priorizado.

Esses achados, portanto, não apenas justificaram a necessidade do Avalista, mas também serviram como o alicerce direto para a definição de seus requisitos e para a fase de concepção, cujos resultados são descritos a seguir.

4.2 Solução Implementada

O resultado final e principal artefato deste trabalho é o protótipo funcional do *frontend* da plataforma Avalista. Desenvolvido com as tecnologias e a arquitetura descritas na Fundamentação Teórica, ele materializa as etapas de pesquisa e design em uma interface interativa. A utilização da plataforma segue um fluxo de trabalho claro e linear, projetado para guiar o avaliador desde a configuração inicial até o registro de problemas de usabilidade.

Primeiramente, o usuário realiza sua autenticação na plataforma, criando uma conta ou efetuando login. Uma vez autenticado, ele é direcionado ao painel principal (dashboard), que serve como uma central para visualizar todos os seus projetos de avaliação. A partir deste painel, o avaliador pode iniciar a criação de um novo projeto, fornecendo um nome e uma descrição. Em seguida, na tela de detalhamento do projeto, ele realiza o cadastro das interfaces que serão objeto da análise, fazendo o upload das imagens de cada tela. Com o

projeto devidamente configurado, o usuário pode, finalmente, iniciar o processo de avaliação. Nesta etapa, ele é levado à interface principal de avaliação, onde visualiza a imagem da tela selecionada e utiliza o organograma interativo da Lista Eureka para registrar os problemas de usabilidade encontrados.

A seguir, são apresentadas as interfaces que compõem este fluxo e uma análise de como suas funcionalidades contribuem para os objetivos do projeto.

4.2.1 Apresentação das Interfaces Desenvolvidas

A interface do Avalista foi projetada com foco em clareza e usabilidade, buscando criar um fluxo de trabalho intuitivo desde o primeiro acesso. As figuras a seguir ilustram os principais módulos desenvolvidos, que representam a prova de conceito visual e funcional da plataforma.

A Figura 9 exibe a tela de autenticação, porta de entrada do sistema. O design prioriza a simplicidade, com um formulário limpo que solicita apenas as credenciais essenciais. É neste ponto que o usuário tem o primeiro contato com a identidade visual do Avalista. A Figura 10 detalha o formulário de cadastro, que segue a mesma linha visual, solicitando as informações mínimas para a criação de uma nova conta de avaliador.

Figura 9 – Tela de Autenticação

Bem-vindo(a) de volta!

Insira seus dados para acessar a plataforma Avalista.

Seu e-mail

Sua senha

Entrar

Não tem uma conta? [Cadastre-se](#)

Fonte: Autoria Própria (2025)

Figura 10 – Tela de Cadastro



A tela de cadastro do sistema Avalista é dividida em duas seções principais. À esquerda, sobre um fundo branco, há o formulário de criação de conta. À direita, sobre um fundo roxo claro, está o logotipo da marca.

Crie sua Conta
Junte-se à comunidade e comece a transformar suas avaliações.

Seu nome

Seu e-mail

Crie uma senha

Confirme sua senha

Cadastrar

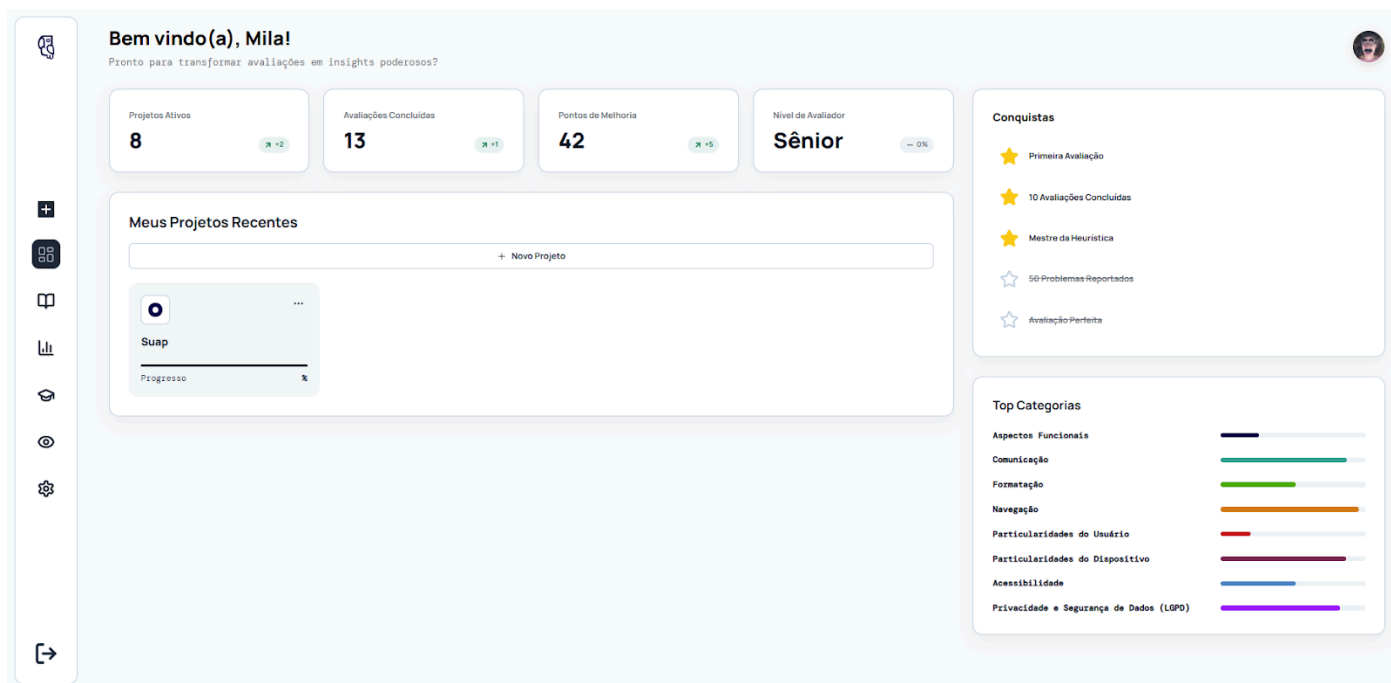
Já tem uma conta? [Faça login](#)

Avalista

Fonte: Autoria Própria (2025)

Após a autenticação, o usuário é direcionado ao painel principal, ou dashboard, exibido na Figura 11. Esta tela funciona como a central de controle do avaliador, apresentando uma listagem dos projetos nos quais ele está envolvido. Cada projeto é representado por um card que resume suas informações e permite o acesso rápido aos detalhes.

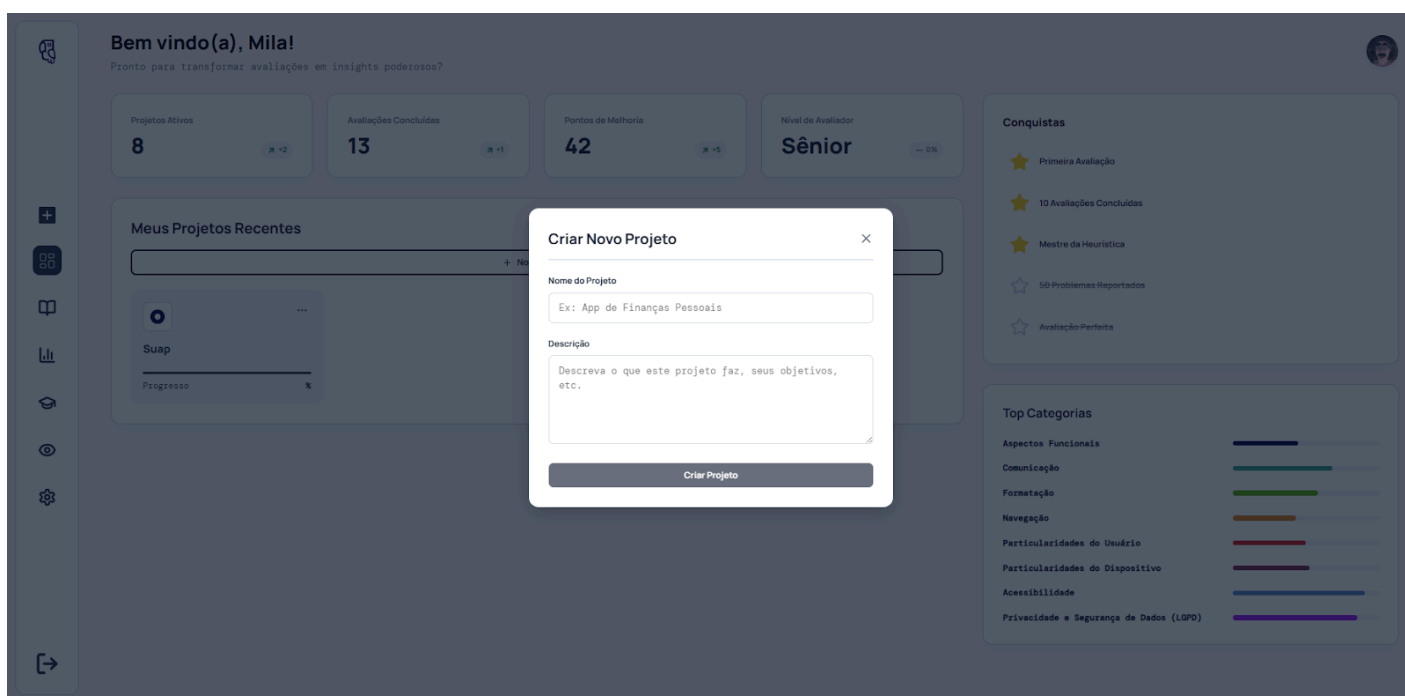
Figura 11 – Painel Principal - Dashboard



Fonte: Autoria Própria (2025)

A Figura 12 demonstra a interface para a criação de um novo projeto, apresentada ao usuário através de um modal para não interromper seu contexto de navegação. O formulário solicita informações essenciais como o nome do projeto e sua descrição, permitindo que o avaliador comece a organizar seu trabalho de forma rápida.

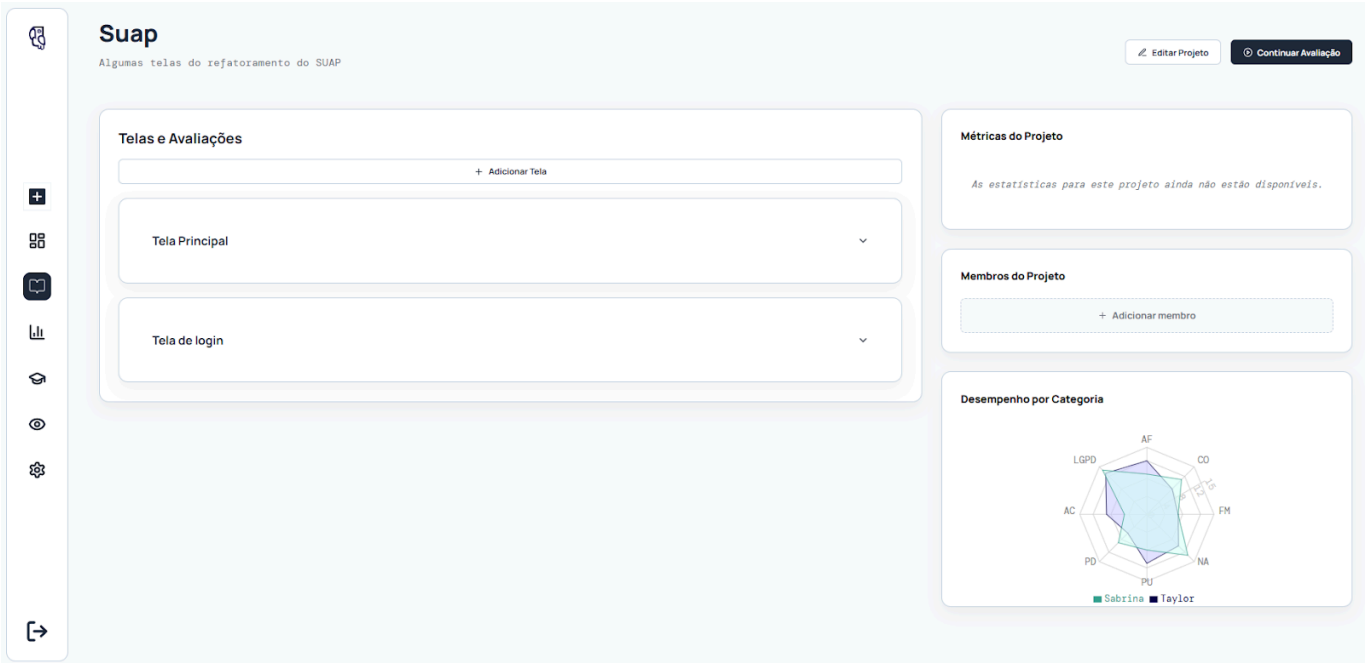
Figura 12 – Criar novo projeto



Fonte: Autoria Própria (2025)

Ao selecionar um projeto, o usuário navega para a tela de detalhamento, ilustrada na Figura 13. Esta página centraliza todas as informações e ações relativas a um projeto específico, como a gestão da equipe de avaliadores e, principalmente, a área para o cadastro das telas que serão analisadas.

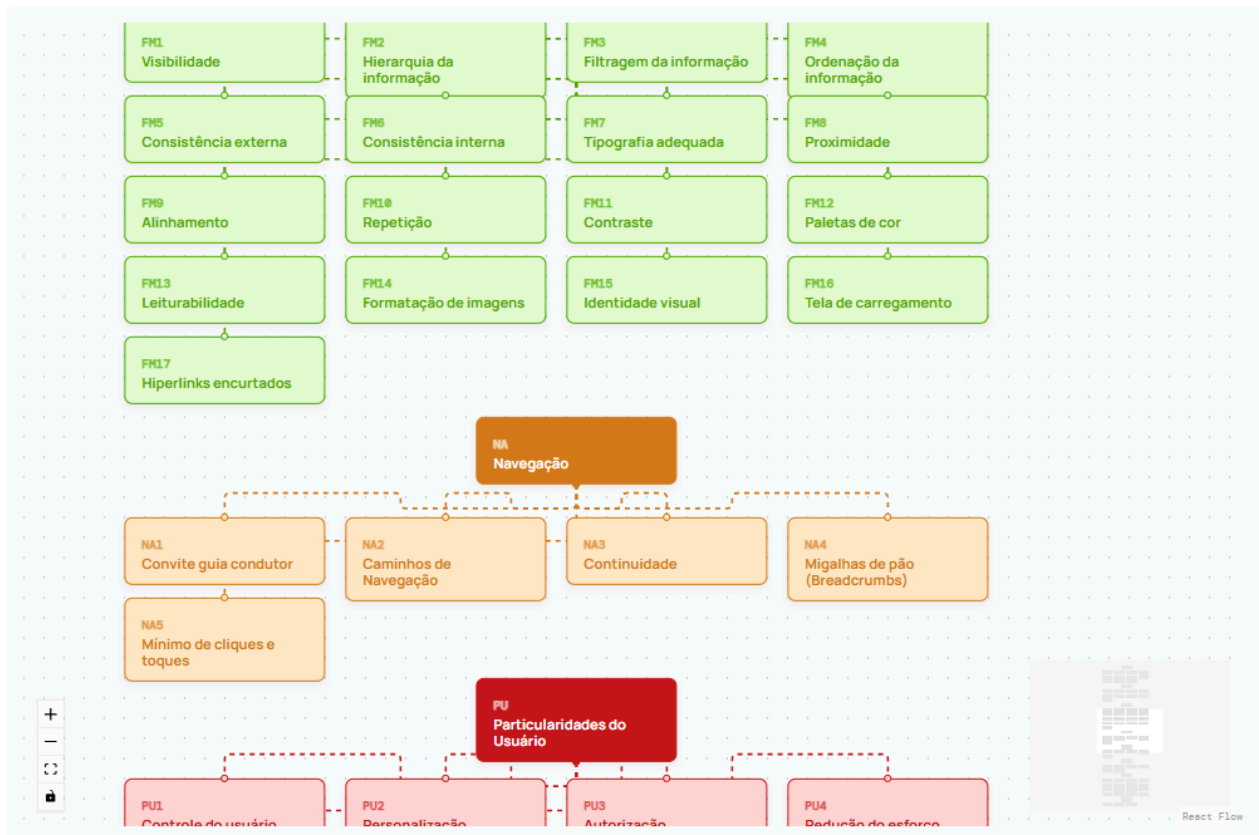
Figura 13 – Detalhamento do Projeto



Fonte: Autoria Própria (2025)

A Figura 14 exibe a tela de avaliação, o coração da plataforma. Esta interface foi projetada para otimizar o fluxo de trabalho, apresentando lado a lado a imagem da tela em análise e a ferramenta de avaliação. O destaque é o organograma interativo, construído com a biblioteca *React Flow*, que renderiza dinamicamente toda a hierarquia da Lista Eureka, desde suas categorias até as diretrizes.

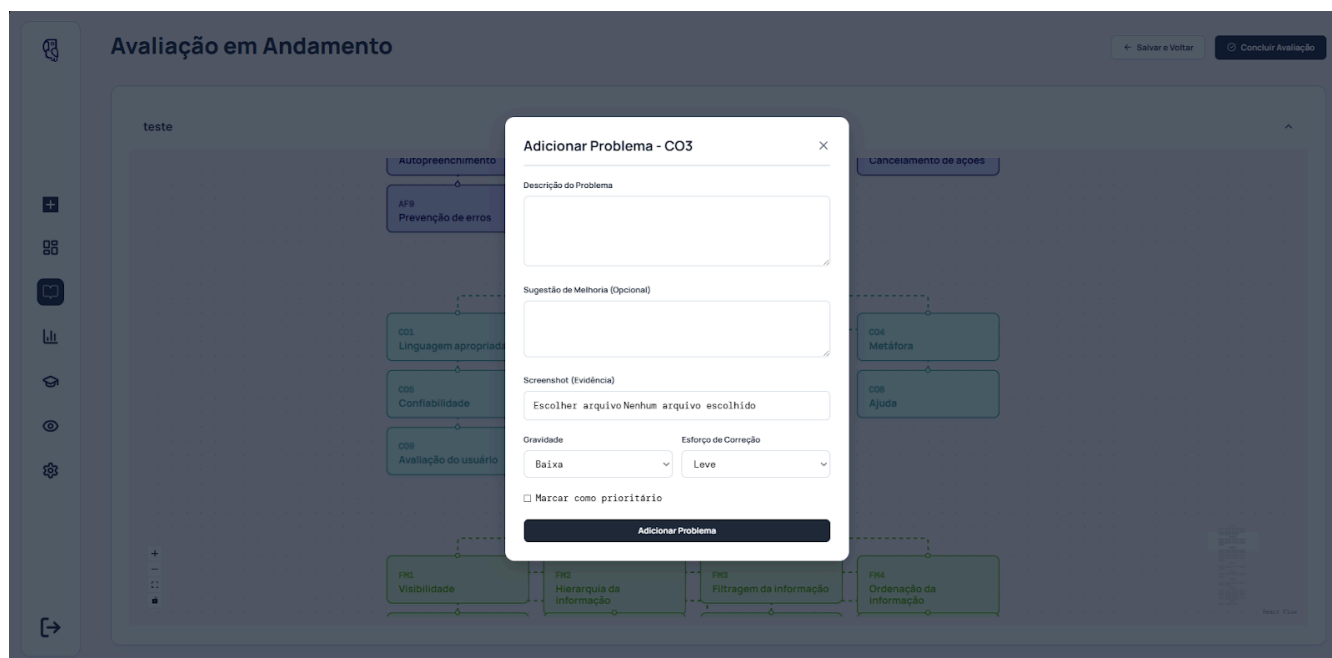
Figura 14 – Interface de Avaliação com Organograma da Lista Eureka



Fonte: Autoria Própria (2025)

Finalmente, a Figura 15 detalha o modal para adição de um problema. Ao clicar em uma diretriz no organograma, este formulário é aberto, permitindo ao avaliador descrever a violação encontrada, sugerir melhorias, classificar a gravidade e o esforço de correção, e anexar uma imagem como evidência, centralizando todo o registro da avaliação em um único fluxo interativo.

Figura 15 – Adicionar problema na avaliação



Fonte: Autoria Própria (2025)

4.2.2 Funcionalidades e Contribuições

O protótipo implementa a jornada essencial do avaliador, e cada módulo foi projetado como uma resposta direta aos objetivos do projeto e aos problemas identificados.

- **Módulo de Autenticação e Gestão de Projetos:** Compreende o fluxo completo de criação de conta, login e o painel central onde o usuário pode criar, visualizar e gerenciar seus projetos (HUs [001](#), [002](#), [004](#), [005](#), [006](#)). Este módulo ataca diretamente o problema da falta de sistematização, oferecendo um ambiente centralizado que substitui a variabilidade dos relatórios manuais. Ao padronizar a criação de projetos e o cadastro de telas, a plataforma tem o propósito de reduzir o esforço operacional do avaliador.
- **Módulo de Avaliação Individual:** Corresponde ao coração da plataforma, implementando o fluxo que permite ao usuário iniciar uma avaliação ([HU012](#)) e adicionar problemas ([HU013](#)). A utilização da biblioteca *React Flow* para exibir a hierarquia da Lista Eureka 2024 de forma interativa foi uma decisão de design que busca apoiar o processo de avaliação de forma automatizada. Ao eliminar a necessidade de consultar um documento externo e ao digitalizar o registro dos achados, a ferramenta visa tornar o processo mais fluido, menos repetitivo e mais convidativo do que o uso de planilhas ou documentos genéricos.
- **Viabilidade Técnica e Conceitual:** Finalmente, a própria existência do protótipo funcional serve como uma validação da viabilidade do projeto. Ele demonstra que é possível, com as tecnologias escolhidas, apoiar um processo de avaliação por inspeção, com uso da Lista Eureka 2024 neste trabalho, em uma ferramenta interativa, estabelecendo uma base de código sobre a qual os futuros módulos poderão ser construídos.

5 Conclusão e Trabalhos Futuros

Este capítulo final consolida a jornada do trabalho. Primeiramente, são apresentadas as conclusões correspondentes aos objetivos propostos e às contribuições do projeto. Em seguida, são reconhecidas as limitações do escopo atual e, por fim, é delineado um caminho claro e detalhado para os trabalhos futuros, que visam a evolução da plataforma Avalista.

5.1 Conclusão

Este trabalho de conclusão de curso partiu de um projeto de pesquisa sobre avaliação de interfaces gráficas, uma competência essencial na formação em tecnologia,. O objetivo central foi, portanto, desenvolver um protótipo funcional, que apoiasse a aplicabilidade da Lista Eureca 2024 e minimizasse as barreiras do processo manual.

Para alcançar tal objetivo, foi realizada a investigação das necessidades dos stakeholders, clientes e também a comunidade acadêmica. O principal resultado entregue é o protótipo funcional do *frontend* da plataforma Avalista, desenvolvido em *React*, que implementa o fluxo essencial de um avaliador individual: desde a autenticação e gestão de projetos até o registro de problemas de usabilidade em uma interface interativa. Este esforço, realizado em paralelo ao desenvolvimento do *backend* por um colaborador, validou a viabilidade técnica da proposta.

As contribuições deste trabalho são: (i) um protótipo funcional de software, que serve como uma prova de conceito e uma base para a evolução do projeto; (ii) uma proposta de automatização de processo, atacando diretamente os problemas de falta de sistematização e do trabalho repetitivo identificados na pesquisa; e (iii) uma solução contextualizada, projetada especificamente para as necessidades da comunidade do IFRN e integrada à pesquisa do projeto Lista Eureca.

Reconhece-se como limitação deste trabalho o escopo focado na implementação do fluxo de avaliação individual, não abrangendo, nesta fase, as funcionalidades colaborativas e de gamificação. Além disso, a não realização de um ciclo formal de avaliação empírica com estudantes representa uma etapa futura crucial para a validação das decisões de design. Contudo, mesmo com essas limitações, o projeto estabelece uma base e aponta um caminho para sua continuidade.

5.2 Trabalhos Futuros

Com a base operacional estabelecida, o cronograma de desenvolvimento revela uma visão integral para a plataforma Avalista. Os próximos passos, planejados para as futuras versões da plataforma, visam transformar o protótipo atual em uma ferramenta robusta e completa, abordando as seguintes frentes estratégicas:

- **Finalização e Aprimoramento do Módulo de Avaliação:** A prioridade máxima é completar o ciclo de avaliação para refletir a experiência concebida. A evolução prevê que, ao criar um projeto, o avaliador possa selecionar as categorias específicas da Lista

Eureca que deseja analisar, adaptando o escopo ao seu contexto. O fluxo de avaliação será aprimorado para fornecer feedback visual de progresso ([HU014](#), [HU015](#)), onde as diretrizes e categorias mudarão de cor conforme forem concluídas. Por fim, serão implementadas as funcionalidades para editar problemas já registrados ([HU020](#)) e finalizar formalmente a avaliação ([HU017](#)), consolidando os dados para a geração de relatórios.

- **Implementação das Funcionalidades Colaborativas:** Para refletir a realidade do desenvolvimento de software em equipe, a próxima grande evolução será a camada colaborativa. Serão desenvolvidos os recursos que permitem adicionar múltiplos avaliadores a um projeto ([HU008](#)) e, o mais importante, iniciar uma avaliação final consolidada ([HU021](#)), permitindo adicionar ([HU047](#)) e editar ([HU022](#)) problemas nesta fase de consenso. Isso transformará o Avalista de uma ferramenta individual para uma plataforma de avaliação em grupo.
- **Construção da Área Administrativa:** Para garantir a autonomia e a evolução da plataforma a longo prazo, será desenvolvida a área de administração. A prioridade é a implementação do *frontend* para o CRUD completo de categorias ([HU027](#), [HU028](#), [HU029](#), [HU030](#)), cujo *backend* já se encontra preparado. Em seguida, será implementado o gerenciamento completo das diretrizes da Lista Eureca ([HU031](#)-[HU035](#)) e um sistema para que os administradores possam analisar e aprovar as sugestões de melhoria enviadas pela comunidade ([HU036](#)-[HU041](#)).
- **Introdução do Sistema de Engajamento e Gamificação:** Para atacar diretamente a barreira da "resistência à prática" identificada na pesquisa, será implementado o sistema de gamificação. Isso incluirá a criação de uma página de perfil do usuário, onde o avaliador poderá visualizar seus emblemas e conquistas ([HU025](#)), que serão atribuídos automaticamente pelo sistema com base em seu desempenho e engajamento ([HU024](#)).
- **Desenvolvimento dos Painéis de Métricas e Relatórios:** Para resolver o desafio de "transformar dados em ação", serão criados painéis para a visualização das estatísticas pessoais do avaliador ([HU044](#)) e, principalmente, das métricas consolidadas de um projeto ([HU045](#)). A capacidade de analisar visualmente as heurísticas mais violadas, o tempo e o esforço estimados para correção, e o progresso geral, transformará os dados brutos da avaliação em insights acionáveis para as equipes de desenvolvimento.

6 Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 25010: Engenharia de sistemas e de software — Requisitos de qualidade e avaliação de sistema e de software (REsqUARE) — Modelos de qualidade de sistema e de software. Rio de Janeiro, 2011.

BENYON, David. Designing interactive systems: a comprehensive guide to HCI, UX and interaction design. 2. ed. Harlow: Pearson, 2011.

COHN, Mike. User stories applied: for agile software development. Boston: Addison-Wesley Professional, 2004.

FERNANDEZ, Adrian; INSFRAN, Emilio; ABRAHÃO, Silvia. Usability evaluation methods for the web: A systematic mapping study. Information and Software Technology, v. 53, n. 8, p. 789–817, 2011.

FISCHER, Gerhard. Beyond ‘couch potatoes’: from consumers to designers. In: ASIA PACIFIC COMPUTER-HUMAN INTERACTION, 5., 1998, Hayama. Proceedings [...]. Los Alamitos: IEEE Computer Society, 1998. p. 2-9.

GLINZ, Martin et al. Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam - foundation level. 2. ed. Santa Barbara: Rocky Nook, 2021.

GACKENHEIMER, Cory. Introduction to React: a-z guide to modern state-driven development. Berkeley: Apress, 2019.

HARTSON, H. Rex. Human-computer interaction: Interdisciplinary roots and trends. The Journal of Systems and Software, v. 43, n. 2, p. 103-118, 1998.

HEURIX. Heuristic evaluation tool for UX practitioners. 2024. Disponível em: <https://heurix.io>. Acesso em: 12 dez. 2024.

HEWETT, Thomas. et al. ACM SIGCHI Curricula for Human-Computer Interaction. New York: ACM, 1992. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/2594128>. Acesso em: 30 jun. 2025.

KOPP, Rudinei. Design gráfico cambiante. Santa Cruz do Sul: Edunisc, 2002.

KREUTZ, Elizete de Azevedo. Identidade Visual Corporativa Mutante: uma estratégia comunicacional contemporânea. In: CONGRESSO BRASILEIRO DE CIÊNCIAS DA COMUNICAÇÃO, 30., 2007, Santos. Anais [...]. São Paulo: Intercom, 2007.

LOWDERMILK, Travis. Design Centrado no Usuário: um guia para o desenvolvimento de aplicativos amigáveis. São Paulo: Novatec Editora, 2014.

LUCIDE TEAM, The. Lucide: Beautiful & consistent icons. 2024. Disponível em: <https://lucide.dev>. Acesso em: 14 jul. 2025.

LUPTON, Ellen; PHILLIPS, Jennifer Cole. Novos fundamentos do design. 2. ed. São Paulo: Gustavo Gili, 2016.

MACK, Robert L.; NIELSEN, Jakob. Usability inspection methods. New York: John Wiley & Sons, 1994.

MATOS, Sílvia Aparecida de Oliveira de Alencar. Roda de Fuxico: um método de ensino brasileiro como proposta de apoio a professores na orientação da criação de sistemas de identidades visuais participativos. 2023. Tese (Doutorado em Design) – Universidade Federal de Pernambuco, Recife, 2023. Disponível em: <https://repositorio.ufpe.br/handle/123456789/52520>. Acesso em: 25 jun. 2025.

MATOS, Sílvia; FREIRE, Marília. Lista Eureka: diretrizes de design para criação e avaliação de interfaces. Natal: IFRN, 2019. (Material técnico).

MATOS, Sílvia; FREIRE, Marília; DUARTE, Isis; VIEIRA, Ramon. Lista Eureka: diretrizes de design para criação e avaliação de interfaces. Natal: IFRN, 2024. (Material técnico).

MATOS, S. A. A. O.; FREIRE, M. A.; MARTINS, G. S. S. M.; MAIA, L. B. L. Fluxograma da Técnica Eureka 2024. (Material técnico). Disponível em: https://drive.google.com/file/d/1_Eg9PSwDZJjbnvvsXZZt0YnHZN1S498D/view.

META PLATFORMS, INC. React. 2024. Disponível em: <https://react.dev/>. Acesso em: 14 jul. 2025.

META PLATFORMS, INC. Thinking in React. 2024b. Disponível em: <https://react.dev/learn/thinking-in-react>. Acesso em: 14 jul. 2025.

MICROSOFT. TypeScript: JavaScript With Syntax For Types. 2024. Disponível em: <https://www.typescriptlang.org/>. Acesso em: 14 jul. 2025.

MORAN, Thomas P. The Command Language Grammar: a representation for the user interface of interactive computer systems. International Journal of Man-Machine Studies, v. 15, n. 1, p. 3-50, 1981.

NIELSEN, Jakob. Heuristic evaluation. In: NIELSEN, Jakob; MACK, Robert L. (ed.). Usability inspection methods. New York: John Wiley & Sons, 1994. p. 25-62.

NIELSEN, Jakob. Usability Engineering. San Francisco: Morgan Kaufmann, 1994.

NIELSEN, Jakob; LORANGER, Hoa. Prioritizing web usability. Berkeley: New Riders, 2005.

NORMAN, Donald A. O design do dia a dia. Rio de Janeiro: Rocco, 2006.

PALMER, Stephen R.; FELSING, John M. A Practical Guide to Feature-Driven Development. Upper Saddle River: Prentice Hall, 2002.

PRATES, Raquel de Oliveira; BARBOSA, Simone Diniz Junqueira. Avaliação de Interfaces de Usuário – Conceitos e Métodos. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA, 22., 2003, Campinas. Anais [...]. Campinas: Sociedade Brasileira de Computação, 2003. cap. 6.

PREECE, Jenny; ROGERS, Yvonne; SHARP, Helen. Design de Interação: Além da Interação Humano-Computador. Porto Alegre: Bookman, 2013.

PREECE, Jenny et al. Human-Computer Interaction. Harlow: Addison-Wesley, 1994.

RECHARTS. Recharts: A composable charting library built on React components. 2024. Disponível em: <https://recharts.org>. Acesso em: 14 jul. 2025.

REMIX SOFTWARE. React Router: Declarative routing for React.js. 2024. Disponível em: <https://reactrouter.com>. Acesso em: 14 jul. 2025.

SAFFER, Dan. Designing for interaction: creating innovative applications and devices. 2. ed. Berkeley, CA: New Riders, 2010.

SOUZA, Clarisse S. de; PRATES, Raquel O.; BARBOSA, Simone D. J. A method for evaluating software communicability. In: WORKSHOP SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, 2., 1999, Campinas. Anais [...]. Campinas, 1999. Artigo 28.

THE AXIOS PROJECT. Axios: Promise based HTTP client for the browser and node.js. 2024. Disponível em: <https://axios-http.com>. Acesso em: 14 jul. 2025.

VITE. Vite: Next Generation Frontend Tooling. 2024. Disponível em: <https://vitejs.dev/>. Acesso em: 14 jul. 2025

XYFLOW. React Flow: A powerful React library for building node-based UIs. 2024. Disponível em: <https://reactflow.dev>. Acesso em: 14 jul. 2025.