

# **AutismGuide: Sistema de avaliação de usabilidade e acessibilidade em produtos digitais para pessoas no espectro autista**

Augusto Henrique Oliveira dos Anjos



CENTRO DE INFORMÁTICA  
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2024

Augusto Henrique Oliveira dos Anjos

# AutismGuide: Sistema de avaliação de usabilidade e acessibilidade em produtos digitais para pessoas no espectro autista

Relatório Técnico apresentado ao Centro de  
Informática da Universidade Federal da  
Paraíba, como requisito obrigatório para  
obtenção de título de Bacharel em Ciência da  
Computação, com linha específica em  
Desenvolvimento Web.

**Orientadora:** Danielle Rousy Dias Ricarte.

João Pessoa, 2024

**Catálogo na publicação**  
**Seção de Catalogação e Classificação**

A599a Anjos, Augusto Henrique Oliveira Dos.

AutismGuide: sistema de avaliação de usabilidade e acessibilidade em produtos digitais para pessoas no espectro autista / Augusto Henrique Oliveira Dos Anjos. - João Pessoa, 2024.

51 f. : il.

Orientação: Danielle Rousy Dias Ricarte Ricarte.

Coorientação: Yuska Paola Costa Aguiar Aguiar.

TCC (Graduação) - UFPB/CI.

1. Acessibilidade. 2. Usabilidade. 3. Espectro Autista. I. Ricarte, Danielle Rousy Dias Ricarte. II. Aguiar, Yuska Paola Costa Aguiar. III. Título.

UFPB/CI

CDU 004.9:616.896



UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE INFORMÁTICA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO  
COORDENAÇÃO DO CURSO



1  
2  
3  
4  
5  
6

Ata da Sessão Pública de Defesa de Trabalho de Conclusão de Curso de Augusto Henrique Oliveira dos Anjos, realizada em **19 de Junho de 2024**.

1 Aos **19** dias do mês de **junho**, do ano de **2024**, às **10** horas, no Centro de Informática,  
2 reuniram-se os membros da Banca Examinadora constituída para julgar o Trabalho de  
3 Conclusão de Curso do Sr. **Augusto Henrique Oliveira dos Anjos**, matrícula nº  
4 **20170009919**, aluno do Curso de Bacharelado em Ciência da Computação da Universidade  
5 Federal da Paraíba. A comissão examinadora foi composta pela professora **Danielle**  
6 **Rousy Dias Ricarte** (UFPB), orientador e presidente da banca, e pelos professores  
7 **Yuska Paola Costa Aguiar** (UFPB), examinador interno, **Derzu Omaia** (UFPB),  
8 examinador interno. Iniciando os trabalhos, a presidente da banca cumprimentou os  
9 presentes, comunicou-os da finalidade da reunião e passou a palavra ao candidato para que  
10 fizesse a exposição oral da monografia intitulada **“AutismGuide: Sistema de avaliação**  
11 **de usabilidade e acessibilidade em produtos digitais para pessoas no espectro**  
12 **autista”**. Concluída a exposição, o candidato foi arguido pela Banca Examinadora que, em  
13 seguida, emitiu o seguinte parecer: **“aprovado”**, com conceito 10,0 (0,0 a 10,0). Do ocorrido,  
14 eu, Giorgia de Oliveira Mattos, Coordenador do Curso de Bacharelado em Ciência da  
15 Computação, lavrei a presente ata que vai assinada por mim e pelos membros da banca  
16 examinadora. João Pessoa, 19 de junho de 2024.

17  
18

Profa. Giorgia de Oliveira Mattos  
Coordenadora do Curso de Ciência da Computação

Profa. Danielle Rousy Dias Ricarte  
Orientadora (UFPB)

Documento assinado digitalmente  
 DANIELLE ROUSY DIAS RICARTE  
Data: 21/06/2024 12:44:31-0300  
Verifique em <https://validar.iti.gov.br>

Profa. Yuska Paola Costa Aguiar  
Examinador interno(UFPB)

Documento assinado digitalmente  
 YUSKA PAOLA COSTA AGUIAR  
Data: 20/06/2024 09:36:03-0300  
Verifique em <https://validar.iti.gov.br>

Prof. Derzu Omaia  
Examinador interno (UFPB)

Documento assinado digitalmente  
 DERZU OMAIA  
Data: 19/06/2024 22:55:51-0300  
Verifique em <https://validar.iti.gov.br>

*“Cada fase da vida tem os seus encantos e desencantos, e todas passam. Crescer é inevitável, superar é uma boa opção.”, (Rosi Coelho).*

## **AGRADECIMENTOS**

Aos meus pais, Adauto Henrique e Ana Cláudia, que foram a base de todo o meu percurso educacional. Seu apoio incondicional e os sacrifícios feitos desde meus primeiros anos escolares foram essenciais para minhas conquistas até aqui. Vocês foram, sem dúvida, meus primeiros e mais importantes professores.

À professora Danielle Rousy, minha orientadora, cuja orientação e sabedoria foram cruciais para a conclusão deste trabalho. Sou imensamente grato por seu apoio acadêmico, paciência e valiosos ensinamentos.

À professora Yuska, que não só me ofereceu a oportunidade de trabalhar em um sistema com tanto potencial de impacto positivo, mas também acreditou na importância de apoiar pesquisas voltadas para melhorar a vida das pessoas no espectro autista. Sua visão e incentivo foram fundamentais para o meu envolvimento neste projeto.

À minha avó, Noemi Falcão, por seu amor e dedicação desde o meu nascimento. Sua influência em minha vida ultrapassa o apoio acadêmico, abrangendo minha formação como ser humano. Sua perseverança e foco sempre foram inspirações para mim.

Aos meus colegas de classe, pelo companheirismo e pelos momentos de amizade e apoio mútuo. Vocês tornaram a jornada acadêmica mais leve e gratificante!

## RESUMO

Este trabalho aborda o projeto e desenvolvimento do AutismGuide, uma aplicação projetada para auxiliar na avaliação da usabilidade e acessibilidade de produtos digitais destinados a usuários no espectro autista. A plataforma inclui funcionalidades como registro de avaliações, cadastro de produtos e análise de conformidade com diretrizes específicas no contexto do autismo. O sistema foi desenvolvido utilizando tecnologias web: a biblioteca ReactJS para a interface do usuário, GraphQL com Apollo Client para gerenciamento de dados e Node.js para o servidor backend. Na sua versão atual, o AutismGuide é totalmente funcional, incorporando todas as funcionalidades essenciais necessárias para a realização de avaliações e geração de relatórios. Após o preenchimento completo dos critérios, que consistem em conjuntos detalhados de recomendações, o usuário obtém o resultado da avaliação em que é possível visualizar claramente os aspectos nos quais o produto digital apresentou deficiências e domínio em termos de usabilidade e acessibilidade para usuários no espectro autista.

**Palavras-Chave:** Aplicação Web, Conformidade, Espectro Autista, Acessibilidade, Usabilidade.

## **ABSTRACT**

This work discusses the design and development of AutismGuide, an application designed to assist in the evaluation of usability and accessibility of digital products intended for users on the autism spectrum. The platform includes functionalities such as evaluation registration, product registration, and compliance analysis with specific guidelines in the context of autism. The system was developed using web technologies: ReactJS library for the user interface, GraphQL with Apollo Client for data management, and Node.js for the backend server. In its current version, the AutismGuide is fully functional, incorporating all the essential features needed for conducting evaluations and generating reports. After the complete fulfillment of the criteria, which consist of detailed sets of recommendations, the user obtains the evaluation result where it is possible to clearly see the aspects in which digital product showed deficiencies and proficiency in terms of usability and accessibility for users on the autism spectrum.

**Keywords:** Web Application, Compliance, Autism Spectrum, Accessibility, Usability.

## LISTA DE FIGURAS

Figura 1: Timeline AutismGuide.....	2
Figura 2: Arquitetura de Aplicação Web.....	6
Figura 3: Diferenças entre Backend e Frontend.....	8
Figura 4: Transferência de dados em aplicações SPA.....	9
Figura 5: Fluxo MVC.....	10
Figura 6: Diagrama de casos de uso.....	13
Figura 7: ERD do banco de dados do AutismGuide.....	20
Figura 8: Tela Home.....	22
Figura 9: Iniciar ou continuar avaliação.....	22
Figura 10: Preencher id da avaliação.....	23
Figura 11: Tela dados do produto.....	24
Figura 12: Tela dados do avaliador.....	25
Figura 13: Tela de Listagem de Categorias.....	26
Figura 14: Tela de Listagem de Recomendações.....	27
Figura 15: Preencher Recomendação.....	27
Figura 16: Ícone de acesso ao Resultado da Avaliação.....	28
Figura 17: Mensagem de acesso ao Resultado da Avaliação.....	29
Figura 18: Avaliação Geral.....	30
Figura 19: Avaliação Detalhada por Categoria.....	31
Figura 20: Avaliação de Inconformidades.....	32
Figura 21: Tabela de Inconformidade.....	32

## **LISTA DE ABREVIATURAS E SIGLAS**

AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
CSS	Cascading Style Sheets
DOM	Document Object Model
ERD	Entity Relationship Diagram
HTML	Hypertext Markup Language
JS	JavaScript
JSON	JavaScript Object Notation
MVC	Model View Controller
MVP	Minimum viable product
TEA	Transtorno do Espectro Autista

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>1</b>
1.1 Objetivos.....	2
1.1.1 Objetivo Geral.....	2
1.1.2 Objetivos Específicos.....	2
1.2 Estrutura do relatório técnico.....	2
<b>2 CONCEITOS GERAIS.....</b>	<b>2</b>
2.1 Aplicativo Web.....	3
2.2 Frontend.....	6
2.3 Backend.....	6
2.4 Single Page Application.....	7
2.5 Model View Controller.....	8
<b>3 PROCESSO DE DESENVOLVIMENTO.....</b>	<b>9</b>
3.1 Visão Geral do AutismGuide como uma Web App.....	10
3.2 Principais Features.....	10
3.3 Tecnologias adotadas.....	11
3.3.1 JavaScript.....	12
3.3.2 ReactJS.....	12
3.3.3 Material-UI.....	13
3.3.4 Node.js e Express.....	14
3.3.5 GraphQL e Apollo Client.....	15
3.3.6 PostgreSQL.....	15
3.4 Arquitetura.....	16
3.4.1 Estrutura e Fluxo do Sistema.....	16
3.5 Banco de dados.....	17
<b>4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS.....</b>	<b>19</b>
4.1 Interface da aplicação.....	19
4.1.1 Tela Home.....	19
4.1.2 Tela Dados do Produto.....	21
4.1.3 Tela Dados do Avaliador.....	22
4.1.4 Tela de Listagem de Categorias.....	23
4.1.5 Tela de Listagem de Recomendações.....	24
4.1.6 Tela de Resultados da Avaliação.....	26
4.1.6.1 Avaliação Geral.....	27
4.1.6.2 Avaliação Detalhada por Categoria.....	28
4.1.6.3 Avaliação de Inconformidades.....	29
<b>5 CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>30</b>
<b>REFERÊNCIAS.....</b>	<b>30</b>
<b>APÊNDICES.....</b>	<b>32</b>
Apêndice A - Tabelas do banco de dados do AutismGuide.....	32

# 1 INTRODUÇÃO

A crescente integração das tecnologias digitais no cotidiano das pessoas exige uma atenção especial às necessidades de usuários com características neurodivergentes. Indivíduos com autismo podem enfrentar desafios únicos ao interagir com interfaces digitais convencionais, que muitas vezes não contemplam suas necessidades específicas em termos de usabilidade e acessibilidade.

O Transtorno do Espectro Autista (TEA) é um distúrbio do neurodesenvolvimento que afeta as habilidades de comunicação, socialização, interesse e imaginação [13]. Alguns indivíduos apresentam graus mais severos enquanto outros apresentam graus mais leves, com cada pessoa exibindo características próprias.

Indivíduos com TEA podem enfrentar diversas barreiras ao interagir com recursos digitais, especialmente no que diz respeito ao uso de cores. A percepção e a interpretação de cores podem variar significativamente entre indivíduos com autismo, tornando essencial o desenvolvimento de interfaces inclusivas que atendam às suas necessidades. O GAIA (Guia de Acessibilidade e Inclusão Autista) [12] fornece diretrizes específicas para o design de interfaces digitais voltadas para usuários no espectro autista. Segundo a diretriz G01 do GAIA [12]: "As cores não devem ser a única forma de transmitir um conteúdo e o contraste entre as cores de fundo e objetos de primeiro plano deve ser adequado para distinguir os itens e diferenciar conteúdos ou relacionar informações similares". Isso significa que, além de garantir um contraste adequado, as interfaces devem usar cores como um complemento, e não como a única maneira de transmitir informações. O conteúdo deve ser compreensível sem a necessidade de imagens ou estilos, garantindo que todos os usuários, independentemente de suas habilidades visuais ou cognitivas, possam acessar e entender a informação de maneira eficaz.

Esta lacuna na concepção de produtos digitais não apenas limita a autonomia desses usuários, mas também dificulta sua plena participação na sociedade digital. Nesse contexto, o AutismGuide surge como uma opção dentro de um conjunto de diretrizes claras e efetivas para a avaliação da usabilidade e acessibilidade dessas tecnologias para indivíduos com TEA. Embora existam outros guias importantes, como o GAIA (Guia de Acessibilidade e Inclusão Autista) [12] e o COGA (Cognitive and Learning Disabilities Accessibility) [15], que fornecem recomendações sobre design digital inclusivo, o AutismGuide complementa essas iniciativas com uma abordagem específica e detalhada.

Um estudo comparativo realizado por Douglas Cardins, graduado em sistemas de informação pela Universidade Federal da Paraíba, analisou as recomendações do AutismGuide em relação a esses dois guias (GAIA e COGA), destacando suas particularidades e complementaridades [14].

**Figura 1: Timeline AutismGuide**



Figura 1: Elaboração própria

O AutismGuide foi desenvolvido em 2019 como resultado de um projeto de pós-doutorado conduzido pela professora Yuska Paola, em parceria com um grupo de professores de psicologia. O projeto focava em questões cognitivas relacionadas ao autismo, incluindo o uso de tecnologia como suporte para intervenções pedagógicas [17]. A partir desse projeto, foi criado um conjunto de recomendações destinado a auxiliar a avaliação da usabilidade e acessibilidade de produtos digitais para usuários no espectro autista [16].

Seguindo a linha do tempo apresentada na Figura 1, em abril de 2021, uma tese de doutorado utilizou o AutismGuide, que até então era apenas um conjunto de recomendações, para verificar a eficácia do suporte tecnológico em comparação com o suporte tradicional em intervenções pedagógicas para promover a autonomia e a aprendizagem de crianças com autismo. Ao longo de um ano, 42 crianças com TEA foram acompanhadas neste estudo [18]. Durante o processo contínuo de aplicação do AutismGuide, surgiu a necessidade de evoluir as recomendações, transformando-as em um instrumento de avaliação mais completo [19]. Isso

envolveu a atribuição de valores aos critérios com base em sua importância, bem como a definição de um score de avaliação e os respectivos níveis de qualidade. Dessa forma, torna-se possível mensurar em valores absolutos o desempenho de um produto digital em termos de usabilidade e acessibilidade para usuários com TEA.

Em novembro de 2021, duas alunas da Universidade Federal da Paraíba, Layslene Silva e Mírian Batista, desenvolveram um projeto de design de interação e criaram um protótipo para uma plataforma web destinada a auxiliar na avaliação da qualidade de software voltado para pessoas com autismo, baseado no AutismGuide [24]. Com o objetivo de tornar o processo de avaliação mais eficiente e integrado, propôs-se o desenvolvimento de uma plataforma web que centralize todas as diretrizes em um único sistema, substituindo o uso de planilhas por uma aplicação. Este é o foco principal deste trabalho, que não apenas busca facilitar a avaliação de software para pessoas com autismo, mas também inspirar a criação de novos projetos e pesquisas que contribuam para uma maior inclusão digital. Todas essas etapas foram cruciais para o desenvolvimento do sistema, culminando na criação de uma versão mínima viável do produto (MVP) em maio de 2024.

## **1.1 Objetivos**

### **1.1.1 Objetivo Geral**

O trabalho visa desenvolver uma aplicação web responsiva, denominada AutismGuide, voltada para fornecer um sistema de suporte para a avaliação de usabilidade e acessibilidade para produtos digitais desenhados para atender às necessidades de usuários no espectro autista.

### **1.1.2 Objetivos Específicos**

- Analisar os requisitos para o AutismGuide em termos de usuário e sistema;
- Definir um modelo arquitetural para o produto;
- Estabelecer um conjunto de tecnologias para apoiar o desenvolvimento;
- Desenvolver uma versão viável mínima do produto (MVP).

## **1.2 Estrutura do Relatório Técnico**

Além desta primeira seção, que engloba os aspectos motivacionais e objetivos do trabalho, este documento apresenta mais quatro seções. A segunda seção dedica-se a

descrever os conceitos fundamentais relacionados ao trabalho. Na terceira seção, a aplicação AutismGuide é explicada em detalhes, incluindo as tecnologias utilizadas no seu desenvolvimento e a modelagem de sua arquitetura. A quarta seção detalha visualmente a aplicação desenvolvida, assim como seu fluxo de telas e realiza uma validação funcional utilizando dados reais gerados por um avaliador. Por fim, a quinta seção conclui o trabalho, apresentando as conclusões, sugerindo melhorias e indicando direções para futuras pesquisas.

## 2 CONCEITOS GERAIS

Nesta seção, serão discutidos os fundamentos que embasam o desenvolvimento de uma aplicação web. Inicialmente, será apresentada uma definição de aplicação web, estabelecendo a base para o entendimento das demais terminologias técnicas. Em seguida, serão detalhados os conceitos de Backend e Frontend, elementos cruciais no universo das aplicações web. Além disso, discutiremos aspectos da arquitetura do sistema, incluindo a abordagem Single Page Application, que define o tipo da nossa aplicação, e o modelo MVC (Model-View-Controller), que delinea a arquitetura de nosso sistema.

### 2.1 Aplicativo Web

Uma aplicação web é um tipo de software projetado para ser acessado e utilizado através de um navegador da web, como Google Chrome, Mozilla Firefox ou Safari, sem a necessidade de instalação prévia no dispositivo do usuário. Essencialmente, uma aplicação web é executada em um servidor remoto e é acessada pelo usuário através da internet. Isso permite que aplicações web sejam atualizadas em tempo real, eliminando a necessidade de o usuário instalar atualizações de software.

As aplicações Web podem assumir os seguintes tipos:

1. **Aplicações Web Estáticas:** Compostas por conteúdo estático, como texto e imagens, essas páginas não requerem interação do usuário e servem principalmente para fornecer informações.
2. **Aplicações Web Dinâmicas:** Interativas e capazes de realizar diversas ações, como preencher formulários ou realizar compras online. Elas são construídas com tecnologias como HTML, CSS, JavaScript, além de frameworks e

linguagens de programação para backend, como PHP, Python ou JavaScript (via Node.js).

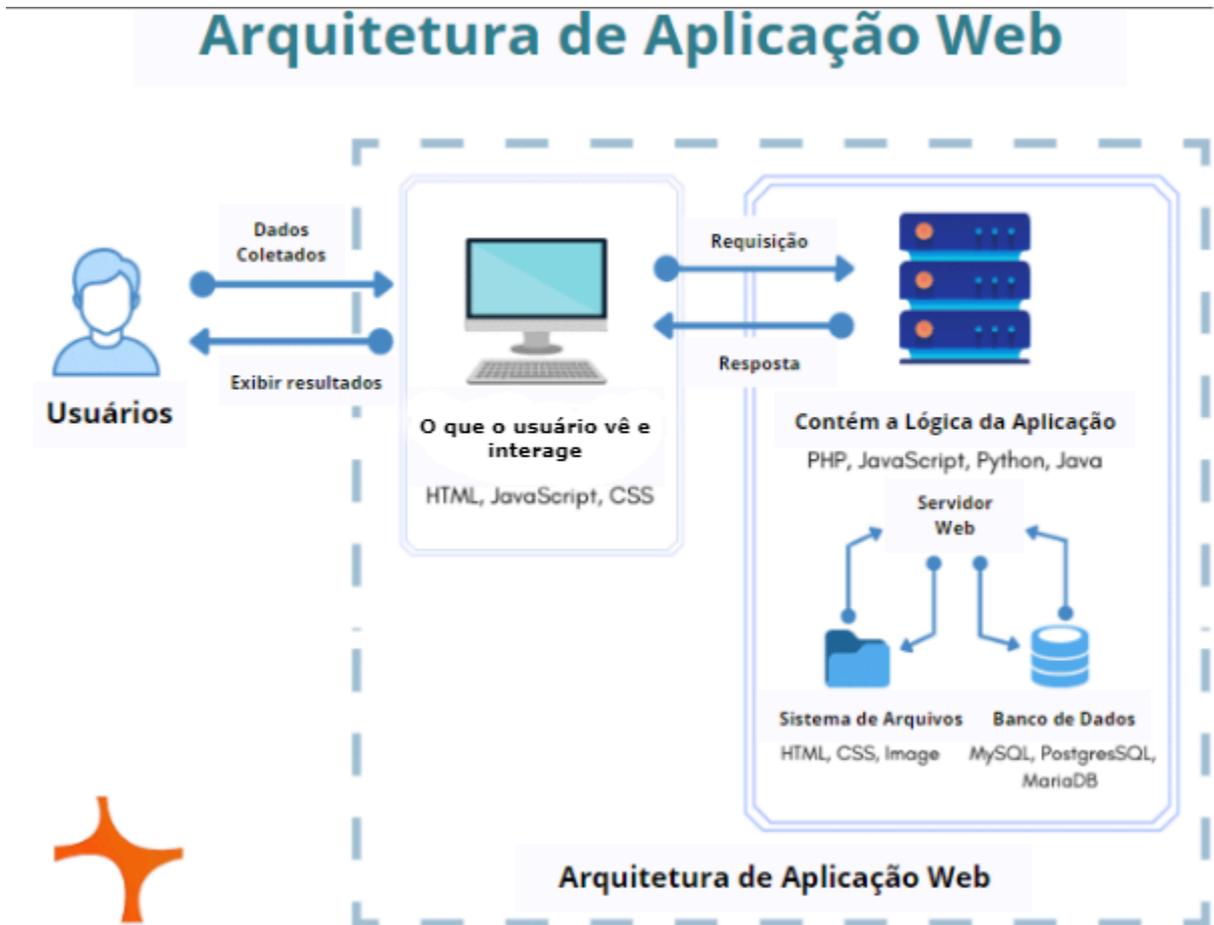
3. **Aplicações Web Progressivas (PWAs):** Oferecem uma experiência semelhante a aplicativos nativos, podendo ser acessadas offline e integrar funcionalidades como notificações push e acesso ao hardware do dispositivo.

Dentre as principais vantagens em desenvolver um software como uma aplicação web, destacam-se:

- Acessibilidade em qualquer dispositivo com um navegador da web.
- Atualizações centralizadas no servidor, refletidas instantaneamente para todos os usuários.
- Eliminação da necessidade de instalação e manutenção de software em dispositivos individuais, simplificando a gestão e o suporte.

A Figura 2 ilustra a arquitetura típica de uma aplicação web, destacando os componentes essenciais e o fluxo de interação entre eles. O diagrama apresenta o frontend, onde o usuário interage diretamente com a aplicação através de tecnologias como HTML, JavaScript e CSS. Este frontend comunica-se com o backend através de requisições que são processadas por um servidor web. No backend, a lógica da aplicação é gerenciada por linguagens de programação como PHP, JavaScript (Node.js) ou Python, e os dados são armazenados e gerenciados em sistemas de banco de dados como MySQL, PostgreSQL ou MariaDB. O servidor também interage com o sistema de arquivos para acessar recursos estáticos como HTML, CSS e imagens. Este modelo permite uma atualização eficiente e um gerenciamento centralizado do software, facilitando a manutenção e melhorando a experiência do usuário.

Figura 2: Arquitetura de Aplicação Web



Fonte: [11]

Em resumo, na Figura 2 tem-se o seguinte fluxo:

1. O usuário interage com a interface do aplicativo através de um navegador e envia um comando ao servidor via Internet.
2. O servidor web recebe este comando e o redireciona para o servidor apropriado.
3. Este servidor processa o comando e busca os resultados necessários.
4. Os dados processados são então enviados de volta para a aplicação web, que os encaminha ao servidor web.
5. Por fim, o servidor web entrega os dados processados de volta ao usuário.

Este diagrama simplificado, ilustrado na Figura 1, ajuda a entender claramente o fluxo de operações dentro de uma arquitetura de aplicativo web.

Para o projeto AutismGuide, optou-se por desenvolver uma Aplicação Web Dinâmica, que integra uma interface intuitiva e funcionalidades interativas. O objetivo com isso é um software que permita uma avaliação eficiente e sistemática das diretrizes de acessibilidade de softwares destinados a usuários com Transtorno do Espectro Autista (TEA).

## **2.2 Frontend**

O Frontend de uma aplicação web representa a interface com a qual o usuário interage diretamente. É construído principalmente utilizando HTML e CSS, responsáveis pela estruturação e estilização do conteúdo da web, respectivamente [2].

O JavaScript é uma linguagem de programação utilizada no desenvolvimento web, essencialmente presente no núcleo do desenvolvimento Frontend trazendo interatividade à interface, permitindo que os usuários executem ações como clicar em botões, enviar formulários e navegar entre páginas sem a necessidade de recarregar a página inteira. Essas tecnologias trabalham em conjunto para traduzir o design e a experiência do usuário em uma interface funcional e dinâmica que se comunica com o servidor para processar as solicitações dos usuários.

## **2.3 Backend**

O Backend, ou lado servidor de uma aplicação web, é onde ocorre o processamento central dos dados e a lógica da aplicação [3]. Quando um usuário realiza uma ação no Frontend, essa solicitação é enviada ao servidor, que opera no Backend. Utilizando linguagens de programação como Python, Java, Ruby, Node.js, entre outras, o servidor processa as requisições, interage com bancos de dados para armazenar ou recuperar informações, executa a lógica de negócios necessária e, em seguida, responde ao cliente [3].

A resposta do servidor é frequentemente formatada em pacotes JSON, um formato leve de troca de dados, que são então interpretados pelo Frontend para exibir o resultado ao usuário. A Figura 3 pode servir como uma referência visual para compreender as principais diferenças entre o Frontend e o Backend.

**Figura 3: Diferenças entre Backend e Frontend.**

FRONTEND	BACKEND
- Envolve a construção da interface visual que os usuários veem	- Lida com a lógica de negócios, gerenciamento de banco de dados e a aplicação que executa no servidor, processando as requisições do usuário
- Tem como linguagens principais: HTML, CSS e JavaScript	- Tem como linguagens principais PHP, Java, Python e JavaScript (para Node.js)
- Um desenvolvedor Frontend é responsável por implementar a interface gráfica do usuário, garantir a acessibilidade e usabilidade do site	- Um desenvolvedor Backend é responsável por gerenciar e proteger os dados, garantir a comunicação entre banco de dados e aplicações e manter a lógica de negócios da aplicação

Fonte: Elaboração própria a partir das referências [2] e [3].

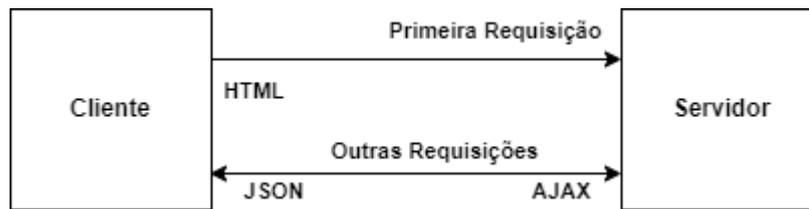
## 2.4 Single Page Application

Como apresentado na seção 2.1, um aplicativo web, ou web app, é um software que é acessado e operado através de um navegador da web, eliminando a necessidade de instalação local no dispositivo do usuário. Esses aplicativos podem ser implementados de diversas maneiras, sendo uma das abordagens mais populares atualmente a utilização de *Single Page Applications* (SPAs).

Em uma SPA, ao contrário de aplicações web tradicionais de múltiplas páginas, a interação com o usuário é centralizada em uma única página web. Esta página é carregada inicialmente no navegador do usuário, onde um documento HTML é enviado juntamente com CSS para controlar a estilização e o JavaScript para funcionalidade. Este JavaScript é essencial porque, após o carregamento inicial, ele gerencia todas as interações subsequentes dentro da aplicação. Ao invés de recarregar páginas completas a cada ação do usuário, a SPA atualiza dinamicamente apenas partes da página, melhorando significativamente a experiência do usuário por ser mais rápida e fluida.

A Figura 4 apresenta o funcionamento das SPA de uma forma mais simples.

**Figura 4: Transferência de dados em aplicações SPA.**



Fonte: Adaptado [1].

Para o AutismGuide decidiu-se desenvolvê-lo como uma web app do tipo SPA. Dessa forma, quando um usuário acessa o AutismGuide pela primeira vez, uma página pré-gerada contendo HTML e CSS é enviada ao navegador. Esta página também inclui JavaScript, que é executado imediatamente ao ser recebido. Conforme o usuário interage com a aplicação, este código JavaScript modifica o DOM (Document Object Model) da página HTML de forma dinâmica, sem a necessidade de fazer novas requisições ao servidor para carregar novas páginas. Isso permite uma interação contínua e sem interrupções, característica fundamental das SPAs.

## 2.5 Model View Controller

O *Model-View-Controller* (MVC) é um padrão de arquitetura que divide uma aplicação em três componentes principais: o modelo (*Model*), a visualização (*View*) e o controlador (*Controller*). Esse padrão é amplamente utilizado no desenvolvimento de aplicações web e permite uma separação clara entre a lógica de negócios, a interface do usuário e a intermediação entre os dois [4].

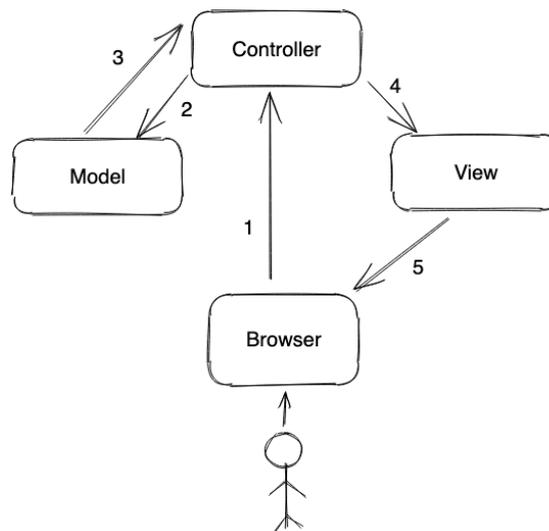
No MVC:

- O **Modelo** representa a lógica de dados da aplicação e interage diretamente com o banco de dados ou o armazenamento de dados. Ele gerencia os dados, a lógica de negócios e as regras da aplicação [4].
- A **Visualização** é a representação da interface do usuário, a qual compõe tudo o que o usuário pode ver e interagir, como caixas de texto e botões. Esta parte é responsável por exibir os dados recebidos do modelo de uma maneira que seja entendida e interativa para o usuário [4].

- O **Controlador** atua como um intermediário entre o modelo e a visualização. Ele controla o fluxo de dados entre o modelo e a visualização e responde a entradas do usuário, convertendo-as em ações a serem realizadas pelo modelo [4].

A Figura 5 ilustra de forma simplificada o funcionamento do padrão MVC.

**Figura 5: Fluxo MVC**



Fonte: [4]

O MVC oferece várias vantagens significativas:

- **Separação de preocupações:** Cada componente do MVC opera de forma independente. Isso significa que os desenvolvedores podem trabalhar em áreas específicas do projeto sem afetar outras partes do código.
- **Facilidade de atualização e manutenção:** Alterações na lógica de negócios ou na interface do usuário podem ser realizadas com mínimo impacto nas outras partes da aplicação.
- **Suporte à modularidade:** Ao seguir o padrão MVC, o AutismGuide pode facilmente incorporar novas funcionalidades ou mudanças nos requisitos sem extensas reformulações do sistema.

Foi o estilo arquitetural MVC, o adotado para o desenvolvimento da web app AutismGuide.

### **3 PROCESSO DE DESENVOLVIMENTO**

Esta seção descreve o processo de desenvolvimento do AutismGuide, incluindo a fundamentação para a escolha das tecnologias empregadas, a estruturação da arquitetura do sistema, a modelagem do banco de dados e a definição dos casos de uso.

#### **3.1 Visão Geral do AutismGuide como uma Web App**

O AutismGuide é uma aplicação projetada para avaliar a usabilidade e acessibilidade de produtos digitais voltados para usuários no espectro autista. O objetivo principal deste sistema é criar um ambiente interativo e eficaz para avaliadores e desenvolvedores aprimorarem seus produtos, considerando as necessidades específicas desse público. A plataforma permite que os avaliadores registrem e analisem variados aspectos dos produtos, como a facilidade de uso, a consistência de interface e a acessibilidade das funcionalidades.

Para garantir uma avaliação eficaz, o AutismGuide foi concebido com requisitos específicos. A aplicação precisa ser altamente responsiva para operar eficientemente em diferentes dispositivos e tamanhos de tela, facilitando o acesso dos avaliadores e usuários em diversos contextos, incluindo computadores e dispositivos móveis. Além disso, a interatividade e a adaptabilidade das interfaces são essenciais para atender às variadas necessidades dos usuários finais, tornando a ferramenta acessível e intuitiva para todos os envolvidos.

#### **3.2 Processo de Trabalho**

O desenvolvimento da aplicação web AutismGuide foi conduzido por meio de reuniões semanais com as professoras Yuska Paola e Danielle Rousy Dias Ricarte, realizadas via Google Meet. Durante essas reuniões, foram discutidas melhorias, ajustes e novos itens a serem adicionados ao backlog do produto. Esses pontos eram registrados em uma ferramenta de gerenciamento chamada Trello, que ajudava a controlar os itens já implementados, aqueles que ainda precisavam ser desenvolvidos e os que seriam adicionados ao backlog para versões futuras.

Além da minha participação, o processo de desenvolvimento dessa versão contou com a colaboração de outro desenvolvedor, Antoni Vinicius, graduando em Ciência da Computação na UFPB. Para versionamento do código, utilizou-se o GitHub, assegurando um controle

eficiente das modificações e permitindo uma colaboração contínua e organizada entre os desenvolvedores.

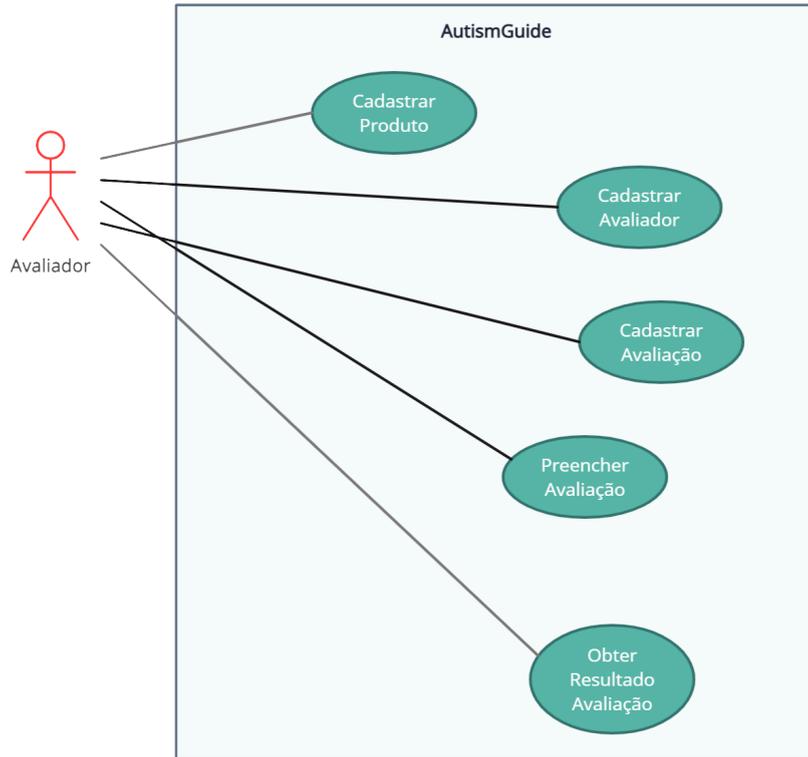
### 3.3 Principais Funcionalidades

Como principais funcionalidades para o AutismGuide, pode-se elencar:

- **Cadastro de Produto Digitais:** Este caso de uso permite ao avaliador inserir informações sobre novos produtos no sistema. É um processo fundamental que envolve o preenchimento de dados específicos relacionados a produtos que serão avaliados ou referenciados dentro da plataforma.
- **Cadastro do Avaliador:** Aqui, o foco está em registrar novos avaliadores no sistema. Isso pode envolver a criação de um perfil, a definição de informações pessoais como nome, profissão e o fornecimento de informações que habilitam o avaliador a participar ativamente das avaliações na plataforma.
- **Cadastro da Avaliação:** Este caso de uso refere-se à criação de uma nova avaliação. Uma avaliação é vinculada a um avaliador e possui uma série de recomendações pré-definidas que são vinculadas a diferentes categorias.
- **Preenchimento da Avaliação:** Depois de uma avaliação ser criada, o avaliador irá efetivamente preencher essa avaliação. Cada recomendação, vinculada a categorias pré-determinadas, é avaliada pelo avaliador com o intuito de refletir o nível de conformidade e a importância do produto digital em análise.
- **Resultado da Avaliação:** Este caso de uso gera uma tabela analítica que exibe não apenas o "Score Máximo" teórico, refletindo o potencial máximo do produto, mas também o "Score Obtido", calculado a partir dos dados inseridos pelo Avaliador.

Na Figura 6, apresenta-se essas funcionalidades usando o diagrama de casos de uso do sistema. Nesse diagrama estão as várias interações que o usuário, identificado como "Avaliador", pode ter com o sistema. Cada caso de uso é representado por uma elipse e está diretamente ligado ao Avaliador, evidenciando as ações que este pode realizar dentro da plataforma:

**Figura 6: Diagrama de casos de uso**



Fonte: Elaboração própria

### 3.4 Tecnologias adotadas

O AutismGuide foi desenvolvido utilizando a biblioteca ReactJS para a construção das interfaces do usuário, permitindo uma experiência rica e responsiva, complementada pelo uso do Material-UI (MUI)<sup>1</sup> como biblioteca de componentes no frontend, o que facilita a criação de interfaces consistentes e visualmente atraentes. Para o armazenamento de dados e gerenciamento de Backend, foi utilizada a tecnologia GraphQL com Apollo Client<sup>2</sup>, escolhida por sua flexibilidade e eficiência na gestão de dados em tempo real. O servidor foi implementado em Node.js, utilizando Express.js<sup>3</sup> para uma gestão eficaz das requisições e rotas, proporcionando uma solução robusta para o processamento de consultas e a integração com o banco de dados PostgreSQL, conhecido por sua confiabilidade e capacidade de lidar com grandes volumes de dados. Ao longo desta seção, será discutido de forma detalhada cada

<sup>1</sup> Material-UI. Biblioteca de componentes. Disponível em: <https://mui.com/material-ui/all-components/>. Acesso em 04/06/2024.

<sup>2</sup> GraphQL com Apollo Client. Documentação. Disponível em: <https://www.apollographql.com/docs/react/>. Acesso em 04/06/2024

<sup>3</sup> Express-JS. Documentação. Disponível em: <https://expressjs.com/>. Acesso em 04/06/2024

uma dessas tecnologias, explorando como elas contribuem para a funcionalidade e eficácia do AutismGuide.

### 3.4.1 JavaScript

O JavaScript (JS) é uma linguagem de programação fundamental para o desenvolvimento web, que funciona junto com as linguagens *Hypertext Markup Language* (HTML) e *Cascading Style Sheets* (CSS) para construir o núcleo do desenvolvimento Frontend [7]. JavaScript é uma linguagem interpretada que suporta vários paradigmas, como funcional, imperativo e orientado a objetos. Além disso, tem tipagem fraca, isto é, significa que permite a conversão implícita de tipos quando uma operação envolve tipos incompatíveis, em vez de gerar erros de tipo. O JavaScript, originalmente foi desenvolvido para funcionar no lado do cliente, agora pode funcionar no lado do servidor usando tecnologias como Node.js, aumentando sua versatilidade e aplicabilidade em aplicações atuais

Muitos frameworks e bibliotecas para desenvolvimento Frontend e Backend usam JavaScript. O gerenciador de pacotes Node Package Manager (NPM) é uma plataforma online onde os desenvolvedores podem publicar e compartilhar pacotes de código reutilizável, e é o maior repositório de pacotes de software no mundo. Sua vasta coleção de módulos disponíveis sublinha o sucesso e a importância crescente do JavaScript ao longo dos anos [7].

### 3.4.2 ReactJS

ReactJS é uma biblioteca JavaScript robusta e amplamente utilizada para construir interfaces de usuário, especialmente para aplicações web de página única (*Single Page Applications* - SPAs). Foi desenvolvida e mantida pelo Facebook, agora Meta Platforms Inc., e lançada em 2013. A principal vantagem do ReactJS reside na sua eficiência na atualização seletiva de partes da interface do usuário (UI) em resposta a mudanças de dados, o que proporciona uma experiência de usuário mais rápida e fluida [6].

A arquitetura do ReactJS é centrada em componentes, permitindo que desenvolvedores construam aplicativos complexos através da composição de componentes reutilizáveis e isolados que gerenciam seu próprio estado [6]. Esses componentes podem ser aninhados, manipulados e reutilizados, facilitando a manutenção e a escalabilidade dos projetos.

Uma das características mais distintas do ReactJS é o Virtual DOM, uma representação abstrata e leve do DOM (*Document Object Model*) que permite ao React fazer atualizações eficientes. O Virtual DOM trabalha reconciliando as mudanças no estado de um componente

com o DOM real de maneira otimizada, aplicando apenas as diferenças detectadas, o que minimiza o custo de performance das atualizações e oferece uma melhoria significativa na experiência do usuário.

O ReactJS também suporta padrões modernos de desenvolvimento como hooks, que permitem o uso de estado e outros recursos do React sem escrever uma classe. Isso simplifica o código e melhora a legibilidade e a organização dos componentes [5]. Com uma comunidade extensa e ativa, ReactJS dispõe de uma vasta gama de bibliotecas e ferramentas complementares que continuam a expandir suas capacidades e facilitar o desenvolvimento de aplicações ricas e interativas. Tudo isso faz do ReactJS uma das ferramentas mais populares e influentes no desenvolvimento Frontend moderno, reforçando a razão pela qual optamos por utilizar essa biblioteca no desenvolvimento do AutismGuide.

### **3.4.3 Material-UI**

O Material-UI é uma biblioteca de componentes React amplamente utilizada, projetada para implementar a linguagem de design do Material Design do Google. Lançado em 2014, o Material-UI tem se tornado uma escolha popular entre os desenvolvedores para criar interfaces de usuário consistentes e atraentes, com um foco significativo na usabilidade e acessibilidade [6].

O Material-UI oferece uma variedade extensa de componentes pré-fabricados que seguem as diretrizes de Material Design, garantindo uma aparência coesa e familiar aos usuários finais. Isso inclui botões, diálogos, ícones, cartões e muito mais, o que facilita o desenvolvimento de uma interface de usuário consistente sem a necessidade de estilização personalizada extensa.

A escolha de Material-UI para o AutismGuide foi motivada por sua rica coleção de componentes acessíveis e personalizáveis, a consistência visual que ela oferece e a eficiência que ela traz para o desenvolvimento do projeto. Essas características tornam Material-UI uma ferramenta valiosa para criar uma aplicação que não só é esteticamente agradável, mas também inclusiva e fácil de usar para todos os usuários.

#### **3.4.4 Node.js e Express**

O Node.js é uma plataforma de código aberto para executar código JavaScript fora de um navegador. Tradicionalmente, o JavaScript era usado principalmente em navegadores web para criar interatividade nas páginas da internet. No entanto, o Node.js permite utilizar JavaScript para desenvolver aplicações no lado do servidor, como serviços web, APIs e até mesmo sistemas operacionais em tempo real.

O Node.js se destaca por sua maneira especial de lidar com operações que podem levar tempo, como ler arquivos ou buscar dados de um banco de dados [8]. Ao invés de esperar que cada tarefa seja concluída antes de iniciar a próxima, o Node.js pode processar muitas solicitações ao mesmo tempo. Ele faz isso através de um modelo chamado "não bloqueante e orientado a eventos" (I/O), o que significa basicamente que ele configura um alerta para cada tarefa e segue adiante [8]. Quando a tarefa termina (por exemplo, os dados são recuperados do banco de dados), o Node.js é avisado e pode então lidar com os resultados. Isso permite que o Node.js gerencie muitas ações simultaneamente, sem perder tempo esperando que cada uma termine, o que o torna muito rápido e eficaz para construir o Backend de uma aplicação como o AutismGuide, onde várias operações podem ocorrer ao mesmo tempo.

Express é um framework de aplicações web para Node.js, que proporciona uma camada fina de recursos fundamentais para a criação de servidores web e APIs[8]. Juntamente com o Node.js, o Express é utilizado por ser um framework web rápido, opinativo e minimalista que oferece ferramentas robustas para criar servidores web e APIs de maneira rápida e fácil. Este par de tecnologias é bem documentado e amplamente suportado pela comunidade, fazendo com que sua adoção no projeto não apenas acelere o desenvolvimento, mas também assegure a estabilidade e a manutenção a longo prazo [6].

#### **3.4.5 GraphQL e Apollo Client**

A escolha de GraphQL e Apollo Client para o desenvolvimento do Backend do AutismGuide foi baseada em sua capacidade de otimizar a interação entre cliente e servidor. O GraphQL é uma linguagem de consulta para APIs que oferece uma abordagem mais eficiente e flexível para o desenvolvimento de interfaces entre cliente e servidor [9]. Isso

permite uma consulta precisa dos dados necessários sem extrair informações excessivas, o que reduz a sobrecarga na rede e melhora o desempenho geral do aplicativo.

Além disso, o Apollo Client, uma biblioteca de gerenciamento de estado avançada para JavaScript, complementa o GraphQL ao gerenciar dados tanto locais quanto remotos de forma eficiente. Ele oferece funcionalidades como cache inteligente e uma integração reativa com a interface do usuário, facilitando a criação de interfaces dinâmicas e responsivas [10]. Isso simplifica o desenvolvimento e a manutenção do sistema, permitindo aos desenvolvedores concentrarem-se mais na lógica do negócio e menos na gestão dos dados.

A integração dessas tecnologias não apenas impulsiona a performance, como também proporciona uma arquitetura mais limpa e fácil de manter, crucial para a sustentabilidade do projeto a longo prazo.

### **3.4.6 PostgreSQL**

O PostgreSQL é um sistema de gerenciamento de banco de dados relacional e objeto-relacional avançado, conhecido por sua robustez, confiabilidade e conformidade. Ele suporta uma ampla gama de tipos de dados e operações avançadas como transações, procedimentos armazenados e replicação, tornando-o ideal para aplicações que exigem alta performance e complexidade operacional. A escolha do PostgreSQL para o AutismGuide foi motivada pela necessidade de um sistema de banco de dados que possa lidar com grandes volumes de dados e complexidade de operações sem comprometer a performance.

### **3.5 Arquitetura**

O AutismGuide adotou o padrão Model-View-Controller (MVC) para promover uma clara separação entre a representação da informação e a interação do usuário. Esta arquitetura estabelece uma estrutura onde o usuário interage com a interface, ou View, que comunica-se com o modelo de dados, ou Model. Todas as operações são intermediadas pelo Controller, que é responsável pela lógica de negócios. A escolha do MVC foi impulsionada pela necessidade de organizar o código de forma eficiente, o que não apenas facilita a manutenção, mas também possibilita a escalabilidade do sistema.

Na prática, no AutismGuide, a View é implementada usando a biblioteca ReactJS, proporcionando uma interface dinâmica que responde às interações do usuário e atualizações de estado. O Model é gerenciado pelo Apollo Client que, junto ao GraphQL, manipula o armazenamento e recuperação de dados, garantindo que as informações estejam sempre atualizadas e consistentes. O Controller, implementado no servidor Node.js com o uso do Express.js, gerencia o fluxo de dados entre a View e o Model, controlando a lógica de negócios e as regras de validação. Isso permite que modificações em uma parte do sistema, como uma atualização da interface do usuário ou uma alteração na lógica de processamento de dados, sejam realizadas independentemente, sem afetar as outras partes do sistema.

### 3.5.1 Estrutura e Fluxo do Sistema

A arquitetura do AutismGuide foi estrategicamente projetada para otimizar a interação entre o Frontend e o Backend, garantindo um fluxo de dados eficiente e uma manutenção facilitada. Utilizando ReactJS no Frontend e Node.js com Apollo Server no Backend, o sistema é estruturado de acordo com o modelo GraphQL, que oferece flexibilidade e eficiência na gestão das requisições e respostas entre o cliente e o servidor.

O sistema é estruturado em torno de arquivos chave que gerenciam diferentes aspectos da interação entre o cliente e o servidor, bem como a manipulação dos dados. Esses arquivos, fundamentais para a arquitetura do sistema, são `schema.js`, `resolvers.js`, e `server.js`, cada um desempenhando papéis específicos dentro da *stack* tecnológica adotada.

- **schema.js:** Define a estrutura dos dados que o sistema pode consumir e fornecer, estabelecendo um contrato claro entre o Frontend e o Backend.
- **resolvers.js:** Responsável pela lógica de obtenção e manipulação dos dados, atuando como intermediário entre as requisições GraphQL e o banco de dados PostgreSQL.
- **server.js:** Configura e inicializa o servidor, integrando o Apollo Server com o framework Express para tratar as requisições e respostas HTTP.

Esses componentes são projetados para trabalhar em conjunto, formando um pipeline de processamento de dados que começa com a recepção de uma requisição, passa pela lógica de negócios nos *resolvers*, e termina com o envio de uma resposta ao cliente. A segregação

clara de responsabilidades facilita a manutenção e a expansão do sistema, permitindo que ajustes ou adições sejam feitos com impacto mínimo nas outras partes do sistema.

O fluxo de dados no AutismGuide é gerenciado da seguinte forma:

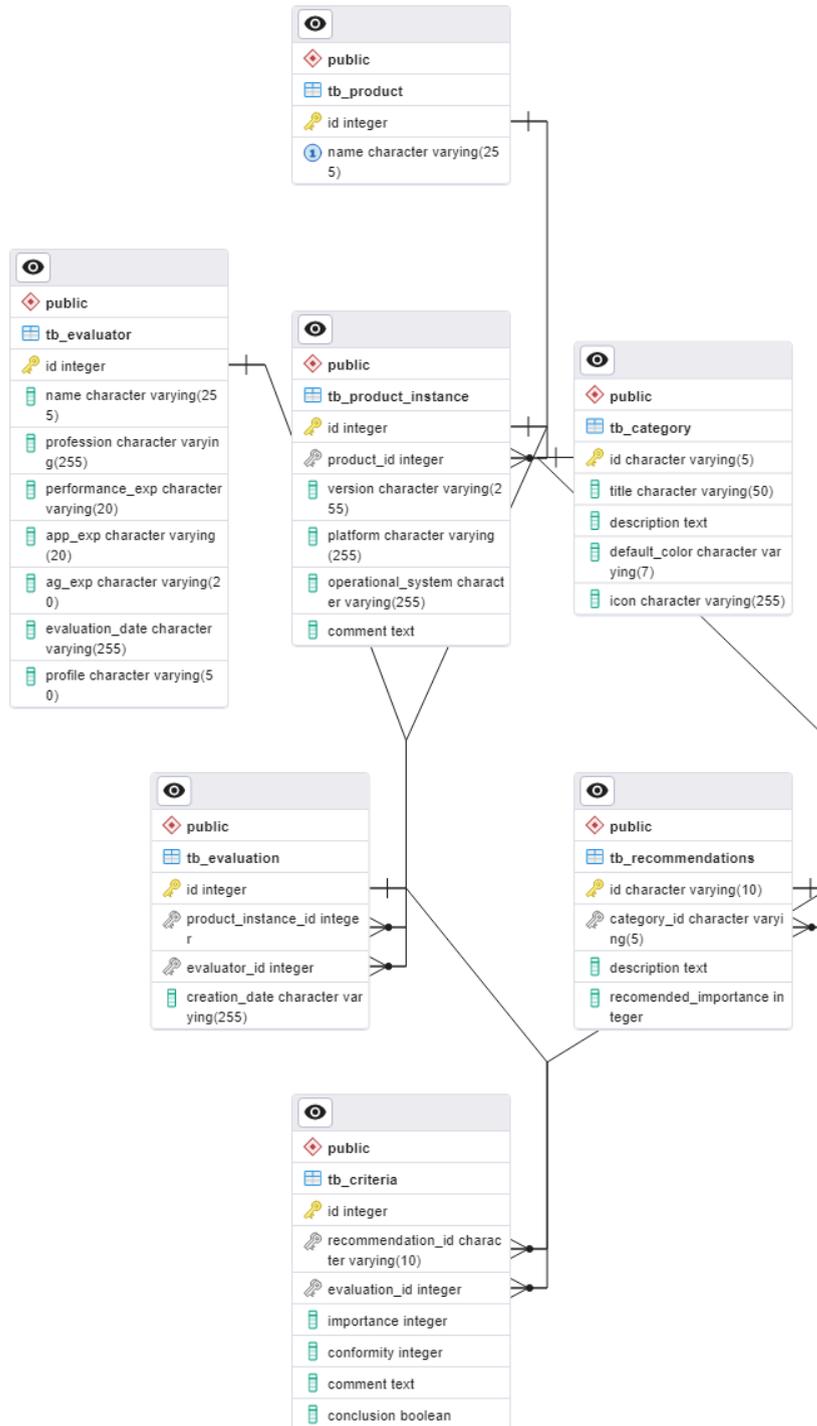
1. **Requisição:** O cliente (FrontEnd) faz uma requisição GraphQL.
2. **Processamento de Requisições:** O Apollo Server recebe a requisição e usa os resolvers apropriados para processar os dados.
3. **Interação com o Banco de Dados:** Os resolvers interagem com o banco de dados PostgreSQL para buscar, registrar ou modificar dados.
4. **Resposta:** Os dados processados são enviados de volta ao cliente como uma resposta GraphQL.

Esta organização assegura que o AutismGuide seja não apenas eficiente em termos de desempenho, mas também escalável e fácil de manter, suportando assim o desenvolvimento contínuo e a implementação de novas funcionalidades conforme necessário.

### 3.6 Banco de dados

O AutismGuide utiliza um banco de dados relacional estruturado conforme o modelo da Figura 7. Ele consiste de várias tabelas inter-relacionadas que armazenam informações sobre produtos, avaliadores, instâncias de produtos, categorias, recomendações, critérios e avaliações. Cada tabela é projetada para armazenar dados específicos e interagir com outras tabelas através de chaves estrangeiras. Os detalhes dessas tabelas e suas inter-relações estão disponíveis nos apêndices do documento, especificamente no Apêndice A.

**Figura 7: ERD do banco de dados do AutismGuide**



Fonte: Elaboração própria

#### 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Nesta seção, são analisados os resultados obtidos através das avaliações realizadas com o uso do AutismGuide. São apresentadas as tabelas que exibem os resultados das avaliações, proporcionando uma visão clara do desempenho e da eficácia dos produtos digitais avaliados. Adicionalmente, apresentaremos as telas da aplicação e descreveremos o

fluxo de processo recomendado, desde o início de uma avaliação até a análise final dos resultados. Esta abordagem não apenas ilustra a funcionalidade do sistema, mas também orienta os usuários sobre como navegar eficientemente pela aplicação.

#### **4.1 Interface da aplicação**

A interface do AutismGuide foi projetada para oferecer uma experiência de usuário intuitiva e acessível, facilitando a navegação através das diversas etapas do processo de avaliação de produtos digitais. Este tópico apresenta um detalhamento do fluxo de telas do sistema, descrevendo o caminho que um usuário percorre desde a criação de uma nova avaliação até a visualização dos resultados. Um vídeo demonstrativo foi publicado na plataforma do YouTube para fins de exibição fluida dos resultados: <https://youtu.be/09XRDmVhENg>

##### **4.1.1 Tela Home**

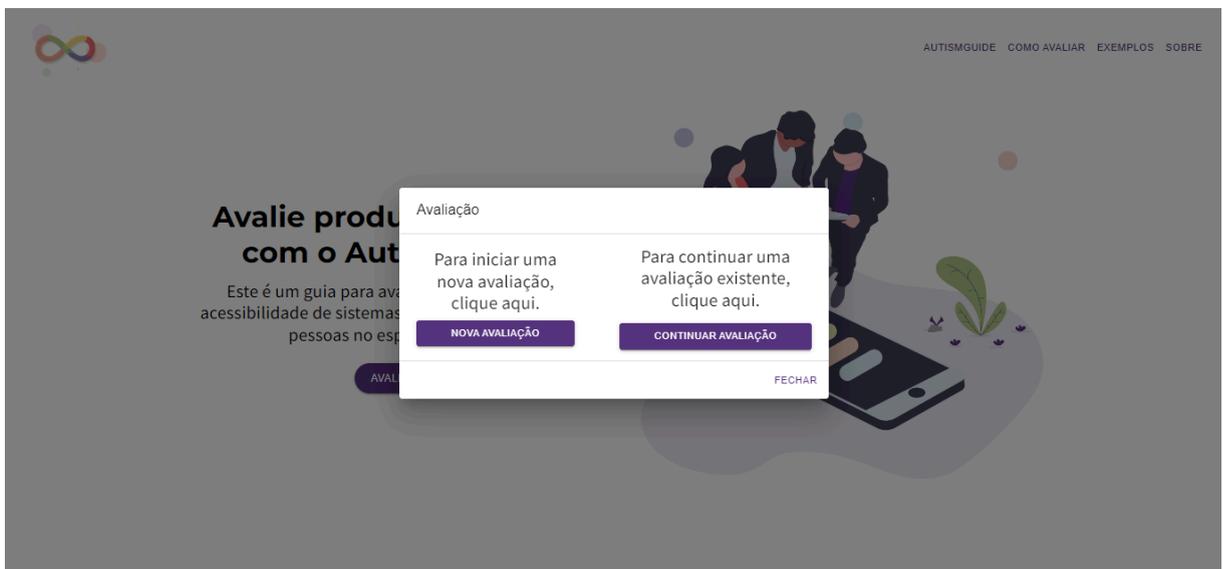
A primeira tela em que temos acesso ao entrar no sistema é a Tela Home (Figura 8). Esta tela apresenta um texto explicativo que destaca a missão do AutismGuide de avaliar a usabilidade e a acessibilidade de produtos digitais destinados a pessoas no espectro autista. Na mesma tela temos um botão "Avaliar" que inicia o processo de avaliação. Ao clicar neste botão, são apresentadas duas opções aos usuários. Como primeira opção, o usuário pode iniciar uma nova avaliação, o que os levará a uma interface para selecionar o produto a ser avaliado e iniciar o preenchimento dos critérios de avaliação. Como segunda opção, para avaliações que já foram iniciadas, há a opção de "Continuar Avaliação". Neste caso, o usuário pode inserir o ID da avaliação e continuar o processo. Esse fluxo pode ser identificado nas figuras 7, 8 e 9, estando relacionado com a feature Cadastro de Avaliação.

**Figura 8: Tela Home**



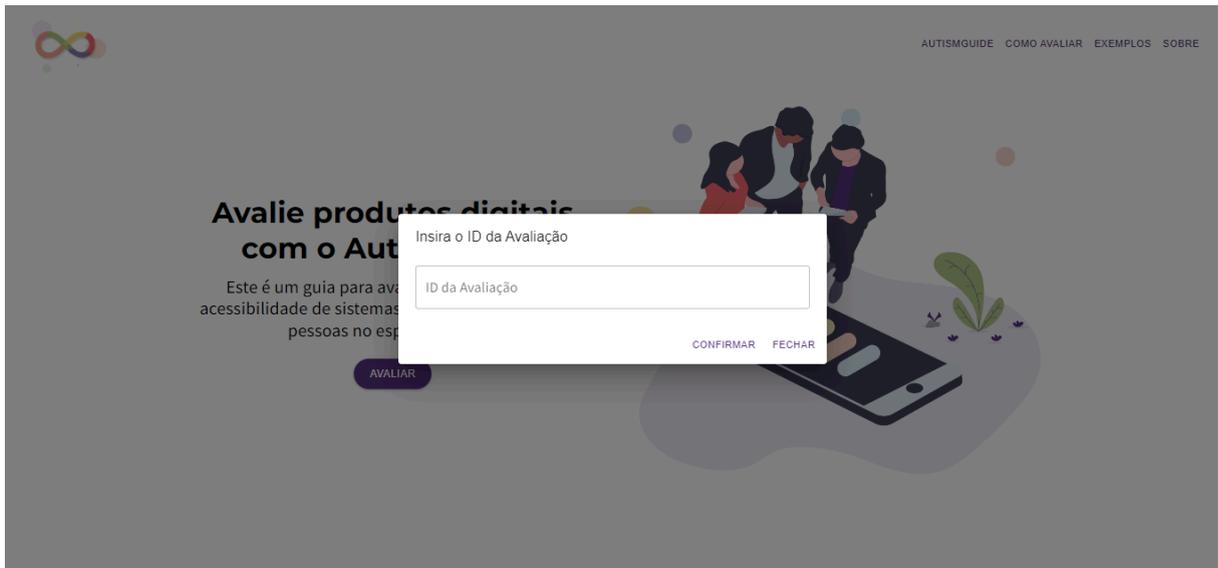
Fonte: Elaboração própria

**Figura 9: Iniciar ou continuar avaliação**



Fonte: Elaboração própria

**Figura 10: Preencher id da avaliação**



Fonte: Elaboração própria

Como o AutismGuide é uma SPA, a interface inclui um cabeçalho com um menu de navegação que facilita o acesso a diferentes seções do sistema. O menu oferece links diretos para "Exemplos de Avaliações", que mostra casos de uso práticos; "Como Avaliar", que fornece orientações detalhadas sobre o processo de avaliação; e "Sobre", onde os usuários podem aprender mais sobre os objetivos e a estrutura do AutismGuide. Adicionalmente, o logo "AutismGuide" no cabeçalho serve como um botão que, quando clicado, redireciona o usuário de volta para a tela inicial, garantindo uma navegação rápida e intuitiva entre as seções.

#### **4.1.2 Tela Dados do Produto**

Nesta tela, o usuário preenche detalhes sobre o produto que está sendo avaliado, incluindo informações fundamentais como o nome e a versão do produto (Figura 11). Além disso, são solicitadas especificações sobre o ambiente de teste do produto, como a plataforma e o sistema operacional utilizado. Há também um campo de observações, onde o usuário pode registrar quaisquer notas adicionais relevantes à avaliação, como particularidades do ambiente de teste, mencionando por exemplo que "A versão do Windows era a 11". Esta organização facilita um registro detalhado das condições sob as quais a avaliação foi realizada, garantindo que o contexto de cada avaliação seja claramente documentado e fácil de compreender.

**Figura 11: Tela dados do produto**

A captura de tela mostra a interface de usuário para a edição de dados de um produto. No topo, há um logotipo colorido e o texto "Home > Produto". No canto superior direito, há links para "AUTISMGUIDE", "COMO AVALIAR", "EXEMPLOS" e "SOBRE". O formulário principal, intitulado "Dados do Produto", contém os seguintes elementos:

- Um campo de texto "Nome" com um ícone de lupa e uma lista de sugestões de produtos.
- Um campo de texto "Versão" com um ícone de engrenagem.
- Dois menus suspensos: "Plataforma" com um ícone de monitor e "Sistema Operacional" com um ícone de engrenagem.
- Um campo de texto "Observações" com um ícone de mensagem.
- Um botão "AVANÇAR" no canto inferior direito.

Abaixo do campo de observações, há um texto pequeno: "Adicione comentários adicionais sobre o produto ou sistema."

Fonte: Elaboração própria

Na Figura 11, o campo de nome de produto é um autocomplete que lista produtos que já foram avaliados no sistema. Ao escolher um produto que já foi avaliado conseguimos manter uma rastreabilidade das avaliações cadastradas para aquele produto.

#### **4.1.3 Tela Dados do Avaliador**

Nesta tela, o usuário insere informações sobre o avaliador responsável pela avaliação (Figura 12). O campo "Nome do Avaliador" apresenta uma lista de avaliadores previamente registrados no sistema. Ao selecionar um avaliador já cadastrado, os demais campos são automaticamente preenchidos e desabilitados para edição, garantindo que os dados do avaliador sejam consistentes com registros anteriores. Esta funcionalidade facilita o processo de entrada de dados e mantém a integridade das informações associadas a cada avaliador.

**Figura 12: Tela dados do avaliador**

The screenshot shows the 'Dados do Avaliador' (Evaluator Data) form. At the top left is the AutismGuide logo and the breadcrumb 'Home > Produto > Avaliador'. At the top right are navigation links: 'AUTISMGUIDE', 'COMO AVALIAR', 'EXEMPLOS', and 'SOBRE'. The form itself is titled 'Dados do Avaliador' and contains the following fields:

- Nome**: A text input field with a person icon.
- Profissão**: A text input field with a briefcase icon.
- Perfil**: A dropdown menu with a question mark icon.
- Experiência**: A section containing three dropdown menus:
  - TEA**: A dropdown menu with a star icon.
  - Recurso Digital**: A dropdown menu with a grid icon.
  - AutismGuide**: A dropdown menu with a graduation cap icon.
- AVANÇAR**: A purple button at the bottom right of the form.

Fonte: Elaboração própria

#### 4.1.4 Tela de Listagem de Categorias

Nesta tela são listadas as categorias vinculadas a avaliação que está sendo registrada. Cada categoria contém um conjunto específico de recomendações associadas (Figura 13). Ao selecionar uma categoria da lista, o usuário é direcionado para uma tela subsequente que detalha as recomendações. Na tela de listagem de recomendações o usuário pode avaliar cada recomendação, atribuindo-lhe um nível de importância e conformidade, além de poder adicionar comentários relevantes.

**Figura 13: Tela de Listagem de Categorias**

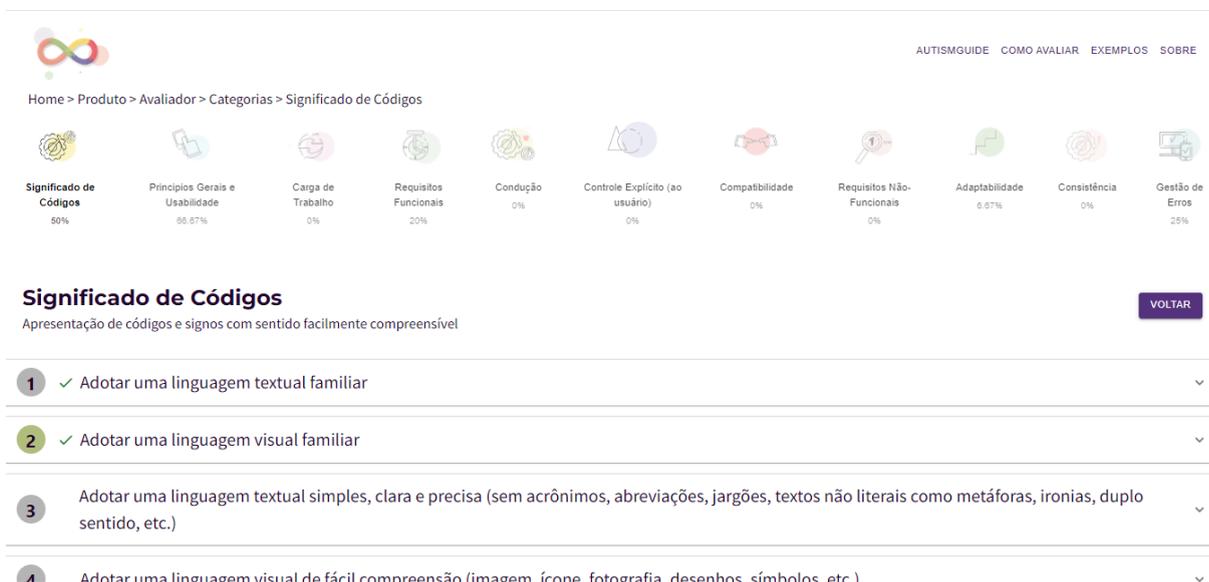


Fonte: Elaboração própria

#### 4.1.5 Tela de Listagem de Recomendações

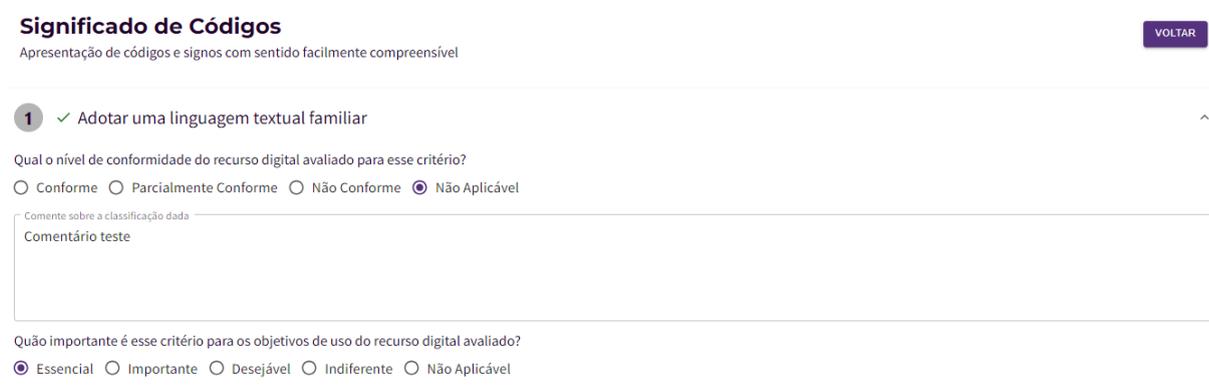
Na tela de listagem de recomendações, o usuário visualiza todas as recomendações associadas à categoria selecionada (Figura 14). Ao clicar em uma recomendação específica, é apresentado um formulário detalhado (Figura 15). Neste formulário, o usuário pode definir o nível de conformidade e importância de cada recomendação, além de inserir quaisquer comentários adicionais que considerar pertinentes. Essa funcionalidade permite uma análise mais detalhada personalizada de cada recomendação, assegurando que todos os aspectos relevantes sejam avaliados e documentados.

**Figura 14: Tela de Listagem de Recomendações**



Fonte: Elaboração própria

**Figura 15: Preencher Recomendação**



Fonte: Elaboração própria

A tela exibida na Figura 15 apresenta um formulário destinado ao preenchimento de uma avaliação específica dentro do AutismGuide. Nesta interface, os usuários têm a oportunidade de avaliar uma recomendação relacionada ao recurso digital em análise. Eles podem determinar o nível de conformidade do recurso com os critérios estabelecidos, avaliar a importância desses critérios para o uso do recurso e adicionar comentários pertinentes. Essas informações coletadas são cruciais, pois contribuem diretamente para o cálculo do

resultado final da avaliação, oferecendo uma base quantitativa e qualitativa para a análise da acessibilidade e usabilidade dos produtos digitais destinados a usuários no espectro autista.

#### 4.1.6 Tela de Resultados da Avaliação

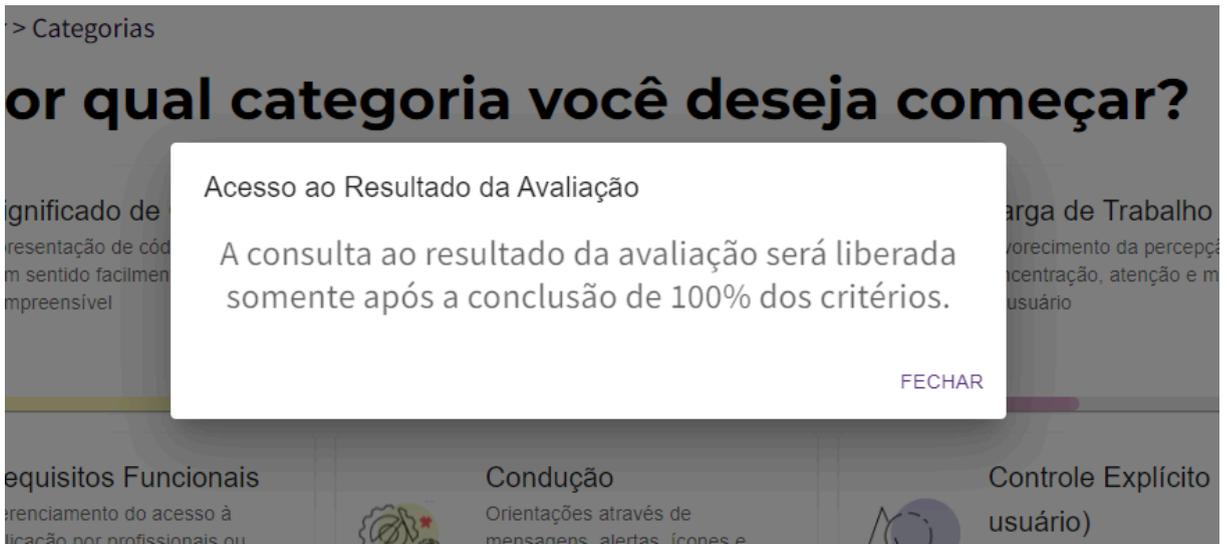
Para acessar a tela de resultados da avaliação, o usuário deve clicar em um ícone global, que é exibido de forma flutuante no canto inferior direito da aplicação, conforme ilustrado na Figura 16. Este ícone só se torna visível após uma avaliação ter sido iniciada ou quando o usuário retorna para continuar o processo de avaliação. Caso o ícone seja selecionado antes da finalização do processo, e ainda não tenham sido atendidos 100% dos critérios de avaliação, será exibida uma mensagem de aviso: “A consulta ao resultado da avaliação será liberada somente após a conclusão de 100% dos critérios”, como mostrado na Figura 17.

**Figura 16: Ícone de acesso ao Resultado da Avaliação**



Fonte: Elaboração própria

**Figura 17: Mensagem de acesso ao Resultado da Avaliação**



Fonte: Elaboração própria

Após o preenchimento completo de 100% dos critérios de avaliação, o ícone para acesso à tela de resultados se torna habilitado. Esta tela é dividida em três seções principais, cada uma detalhando diferentes aspectos dos resultados obtidos. Para facilitar a análise e compreensão dos resultados, as três seções serão discutidas em detalhes nos respectivos subtópicos deste capítulo intitulado “Tela de Resultados da Avaliação”.

#### **4.1.6.1 Avaliação Geral**

A tela de avaliação geral, ilustrada na Figura 18, apresenta uma tabela que inclui uma régua de porcentagem na primeira linha, variando de 0 a 100% em incrementos de 5%. A segunda linha exibe o resultado geral da avaliação do recurso digital, onde 100% representa o score máximo possível e a marcação na linha indica o resultado efetivamente obtido.

No cabeçalho da tabela, são exibidas duas escalas de avaliação. A primeira escala classifica o nível de aceitação do score obtido em três categorias:

- **Inaceitável** - 0% a 45%
- **Intermediário** - 50% a 65%
- **Aceitável** - 65% a 100%

A segunda escala avalia a qualidade com as seguintes classificações:

- **Pior** - 0% a 25%

- **Ruim** - 30% a 35%
- **Razoável** - 40% a 50%
- **Bom** - 55% a 70%
- **Excelente** - 75% a 90%
- **Melhor** - 95% a 100%

Após a linha que exibe o resultado geral da avaliação, a tabela na Figura 18 detalha os resultados por categoria em cada linha subsequente. Esta disposição oferece uma análise específica do desempenho em cada categoria avaliada, permitindo uma visão mais granular dos resultados. Além disso, na Figura 18 é exibido um resultado hipotético da avaliação de um produto digital. O Spotify não foi efetivamente avaliado neste relatório técnico; seu nome foi incluído apenas para fins de demonstração.

**Figura 18: Avaliação Geral**

VOLTAR

### Resultado da Avaliação - Spotify

#### Avaliação Geral

Pct%	Inaceitável (0-45%)										Intermediário (50-65%)				Aceitável (acima de 65%)						
	Pior					Ruim		Razoável			Bom		Excelente			Melhor					
	0%	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
Geral	0	121	242	362	483	604	725	846	966	1087	1208	1329	1450	1570	1691	1812	1933	2058	2174	2295	2416
ERRM	0	6	11	17	22	28	34	39	45	50	56	62	67	73	78	84	90	95	104	106	112
ADAP	0	21	42	64	85	106	127	148	170	191	212	233	254	276	297	318	356	360	382	403	424
FRQ	0	16	31	47	62	78	94	109	125	140	156	172	187	203	218	234	250	265	292	296	312
SGCD	0	6	11	17	22	28	34	39	45	50	56	62	67	73	78	84	90	95	104	106	112
CNST	0	2	3	5	6	8	10	11	13	14	16	18	19	21	22	24	26	27	29	30	32
GUP	0	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114	120
CMPT	0	9	18	28	37	46	55	64	74	83	92	101	110	120	130	138	147	156	166	175	184
GUID	0	19	38	56	75	94	113	132	150	169	188	207	226	244	263	282	301	320	338	357	376
ECTL	0	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160
WKLD	0	12	24	36	48	60	72	84	96	108	120	132	144	156	168	180	192	204	216	228	240
NFRQ	0	17	34	52	69	86	103	120	138	155	172	189	206	224	262	258	275	292	310	327	344

Fonte: Elaboração própria

**Tabela 1: Identificador e Título de Categorias**

IDENTIFICADOR	TÍTULO
ADAP	Adaptabilidade

CMPT	Compatibilidade
CNST	Consistência
ECTL	Controle Explícito (ao usuário)
ERRM	Gestão de Erros
FRQ	Requisitos Funcionais
GUID	Condução
GUP	Princípios Gerais e Usabilidade
NFRQ	Requisitos Não-Funcionais
SGCD	Significado de Códigos
WKLD	Carga de Trabalho

Fonte: Elaboração Própria

A Tabela 1 apresenta uma categoria por linha e contém duas colunas. A primeira coluna exibe o identificador da categoria, enquanto a segunda mostra o título correspondente. Na tabela de resultados da avaliação geral (mostrada na Figura 18), os resultados são listados por categoria após a linha de resultado geral, com cada categoria sendo identificada pelo seu respectivo identificador.

#### 4.1.6.2 Avaliação Detalhada por Categoria

A seção "Avaliação Detalhada por Categoria", ilustrada na Figura 19, exibe uma análise específica dos resultados da avaliação, organizada por categorias. A primeira coluna identifica o nome da categoria. As três colunas subsequentes, rotuladas como "Não Conforme", "Parcialmente Conforme" e "Conforme", exibem o somatório das recomendações que se enquadram em cada nível de conformidade para a categoria em análise. Após essas colunas, a tabela mostra o score obtido em cada categoria, seguido pelo score percentual, que indica a proporção do score obtido em relação ao score máximo possível. Finalmente, a tabela apresenta a classificação de qualidade para cada categoria e o valor do score máximo possível para cada uma delas.

**Figura 19: Avaliação Detalhada por Categoria**

**Avaliação Detalhada por Categoria**

Categoria	Não Conforme	Parcialmente Conforme	Conforme	Score	Score Percentual	Classificação	Score Máximo
 Gestão de Erros	0	24	80	104	92.86%	Excelente	112
 Adaptabilidade	-12	96	272	356	83.96%	Excelente	424
 Requisitos Funcionais	0	60	232	292	93.59%	Excelente	312
 Significado de Códigos	0	24	80	104	92.86%	Excelente	112
 Consistência	-8	0	16	8	25.00%	Pior	32
 Princípios Gerais e Usabilidade	0	0	120	120	100.00%	Melhor	120
 Compatibilidade	-16	18	128	130	70.65%	Bom	184
 Condução	0	114	224	338	89.89%	Excelente	376

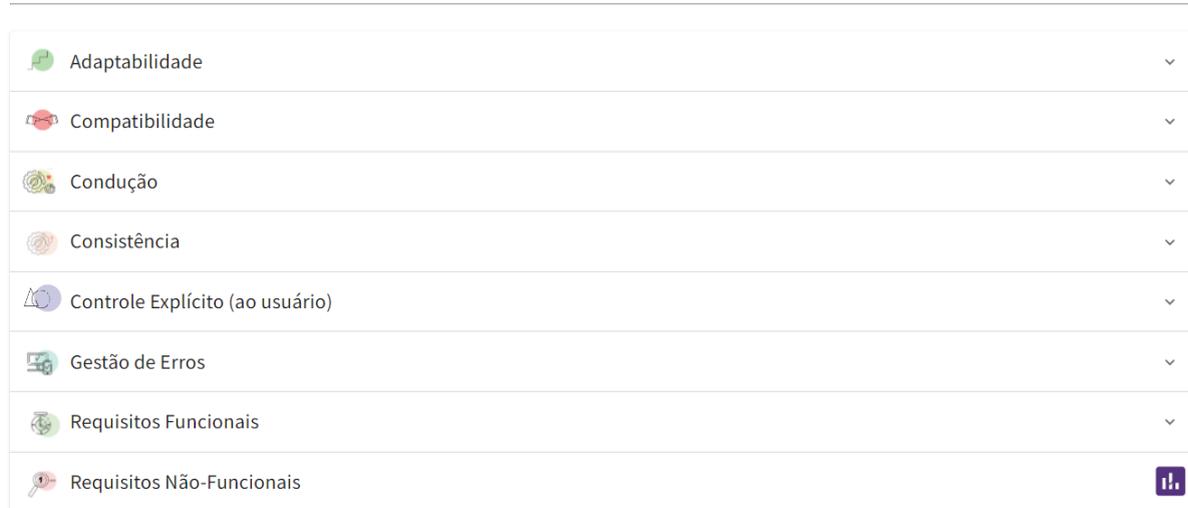
Fonte: Elaboração própria

#### 4.1.6.3 Avaliação de Inconformidades

Nesta seção, a "Avaliação de Inconformidades" fornece uma visão detalhada de cada recomendação categorizada como "Parcialmente Conforme" ou "Não Conforme" (Figura 20). Cada item na tabela inclui o nível de importância atribuído à recomendação e qualquer comentário associado à avaliação. Os dados são organizados por categoria, e os resultados detalhados podem ser acessados clicando nos acordeões correspondentes, conforme ilustrado nas Figuras 20 e 21. Esta organização facilita a revisão das áreas que necessitam de melhorias.

**Figura 20: Avaliação de Inconformidades**

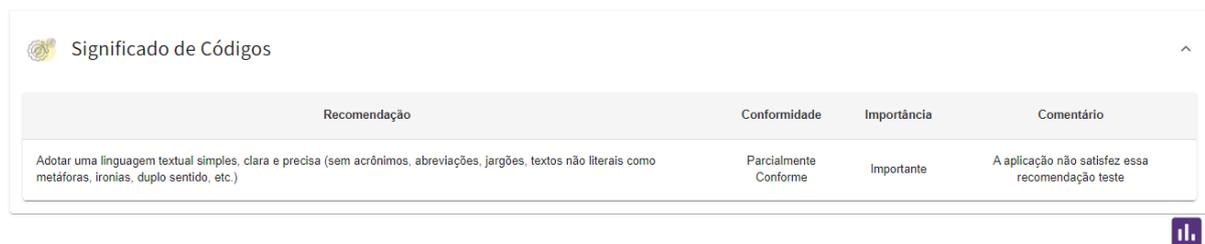
**Avaliação de Inconformidades**



 Adaptabilidade	▼
 Compatibilidade	▼
 Condução	▼
 Consistência	▼
 Controle Explícito (ao usuário)	▼
 Gestão de Erros	▼
 Requisitos Funcionais	▼
 Requisitos Não-Funcionais	▼

Fonte: Elaboração própria

**Figura 21: Tabela de Inconformidade**



Recomendação	Conformidade	Importância	Comentário
Adotar uma linguagem textual simples, clara e precisa (sem acrônimos, abreviações, jargões, textos não literais como metáforas, ironias, duplo sentido, etc.)	Parcialmente Conforme	Importante	A aplicação não satisfaz essa recomendação teste

Fonte: Elaboração própria

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou o desenvolvimento do AutismGuide, uma aplicação web destinada a melhorar a avaliação de usabilidade e acessibilidade de produtos digitais para usuários no espectro autista. O estudo conduzido revelou a necessidade crítica de ferramentas que auxiliem desenvolvedores e avaliadores a entender e implementar requisitos de acessibilidade mais efetivamente. A aplicação foi projetada para oferecer uma interface intuitiva que facilita o registro e a análise de avaliações, fornecendo pontos valiosos sobre a conformidade dos produtos com as diretrizes de acessibilidade.

A arquitetura e as funcionalidades do AutismGuide foram descritas, com demonstrações através das interfaces de usuário do sistema.

Apesar dos resultados positivos, o AutismGuide ainda possui espaço para melhorias e evolução. A aplicação provou ser funcional e foi disponibilizada para os professores responsáveis pela idéia da aplicação, com a expectativa de que se torne uma ferramenta essencial para os avaliadores de acessibilidade digital. Para continuar a atender às necessidades dos usuários e expandir sua funcionalidade, os seguintes itens foram identificados para inclusão no backlog do produto para futuras entregas:

- **Autenticação de Usuário no Sistema:** Implementação de um sistema de login para aumentar a segurança e personalizar a experiência do usuário.
- **Nível de Importância da Recomendação Pré-Definida:** Permitir que um perfil administrador defina um valor recomendado à importância das recomendações, personalizando a avaliação conforme a necessidade.
- **Internacionalização da Aplicação:** Analisar a tradução do sistema para Inglês e Francês para tornar o AutismGuide acessível a um público mais amplo.
- **Filtragem de Sistema Operacional por Plataforma:** Desenvolver uma funcionalidade que permita, no cadastro do produto, filtrar os sistemas operacionais listados por plataforma selecionada.
- **Refinamento da Estrutura de Dados de Avaliadores:** Quebrar a tabela de avaliadores em três partes: `tb_evaluator`, `tb_exp` e `tb_evaluator_instance`, para possibilitar a atualização dos níveis de experiência dos avaliadores.

Estas melhorias visam não apenas otimizar o desempenho e a eficácia do AutismGuide, mas também expandir sua aplicabilidade e relevância no campo da acessibilidade digital. Trabalhos futuros também poderão explorar a integração com outras plataformas e ferramentas, aumentando ainda mais o impacto do AutismGuide no desenvolvimento de produtos acessíveis para usuários com TEA.

## REFERÊNCIAS

- [1] BAJAJ, Pankaj. **Overview of Single Page Application (SPA)**. C# Corner [Internet]. Outubro, 2017. Disponível em: <http://www.c-sharpcorner.com/blogs/overview-of-single-page-applicationspa1>
- [2] FERGUSON, Nicole. **What's the Difference Between Frontend vs Backend Web Development?** CAREERFOUNDRY [Internet]. Fevereiro, 2024. Disponível em: <https://careerfoundry.com/en/blog/web-development/whats-the-difference-between-frontend-and-backend/>
- [3] LEMONAKI, Dionysia. **Frontend VS Backend – What's the Difference?** FreeCodeCamp [Internet]. Março, 2022. Disponível em: <https://www.freecodecamp.org/news/frontend-vs-backend-whats-the-difference/>
- [4] KHOT, Akshay. **MVC Pattern, explained**. DEV Community [Internet]. Janeiro, 2021. Disponível em: [https://dev.to/software\\_writer/introduction-to-the-mvc-pattern-in-asp-net-58c8](https://dev.to/software_writer/introduction-to-the-mvc-pattern-in-asp-net-58c8)
- [5] KUMAR, Akshay. **A Guide to React Hooks With Examples**. Built In [Internet]. Novembro, 2023. Disponível em: <https://builtin.com/software-engineering-perspectives/react-hooks>
- [6] OLAWANLE, Joel. **How to Get Started With React – A Beginner's Guide**. FreeCodeCamp [Internet]. Abril, 2022. Disponível em: <https://www.freecodecamp.org/news/get-started-with-react-for-beginners/>
- [7] FLANAGAN, David. **JavaScript: The Definitive Guide, 7th Edition**. O'Reilly Media. [Livro]. Janeiro, 2020.
- [8] STEVEN, Perry. **Using Express.js for Node applications**. IBM Developer [Internet]. Novembro, 2021. Disponível em: [https://developer.ibm.com/tutorials/learn-nodejs-expressjs/?mhsrc=ibmsearch\\_a&mhq=Node%26period%3Bjs%20and%20express](https://developer.ibm.com/tutorials/learn-nodejs-expressjs/?mhsrc=ibmsearch_a&mhq=Node%26period%3Bjs%20and%20express)

- [9] ATANDA, David. **GraphQL On The Front-End (React And Apollo)**. Smashing Magazine [Internet]. Novembro, 2021. Disponível em: <https://www.smashingmagazine.com/2021/11/graphql-frontend-react-apollo/>
- [10] LEE, David. **Integrating third-party APIs into GraphQL with Apollo Client**. ALGOLIA [Internet]. Março, 2022. Disponível em: <https://www.algolia.com/blog/engineering/integrating-third-party-apis-into-graphql-with-apollo-client/>
- [11] Sharma, Pooja. **7 razões para tratar seriamente a arquitetura de aplicativos da web**. CYNOTECK [Internet]. Dezembro, 2021. Disponível em: <https://cynoteck.com/pt/blog-post/reasons-to-treat-web-application-architecture-seriously/>
- [12] PAGANI, Talita. **GAIA: Sites inclusivos a pessoas com autismo**. GAIA [Internet]. Acesso: Junho, 2024. Disponível em: <https://gaia.wiki.br/>
- [13] American Psychiatric Association. **Diagnostic and Statistical Manual of Mental Disorders, 5th Edition**. ARTMED [Editora]. Maio, 2013. Versão traduzida disponível em: <https://www.institutopebioetica.com.br/documentos/manual-diagnostico-e-estatistico-de-trans-tornos-mentais-dsm-5.pdf>
- [14] CARDINS, Douglas. **Towards accessibility for users with autism: a comparative analysis of guidelines**. SBCOPENLIB [Internet]. Outubro, 2022. Disponível em: <https://sol.sbc.org.br/index.php/ihc/article/view/22274>
- [15] SEEMAN, Lisa; COOPER, Michael. **COGA: Cognitive and Learning Disabilities Accessibility**. W3C Editor's. Setembro, 2021. Disponível em: <https://w3c.github.io/coga/issue-papers/#abstract>
- [16] PAOLA, Yuska. **AutismGuide: a usability guidelines to design software solutions for users with autism spectrum disorder**. Taylor & Francis [Internet]. Novembro, 2020. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/0144929X.2020.1856927>

- [17] PAOLA, Yuska. **A survey on accessibility guidelines for users with autism: a broad understanding of the relevance and completeness.** SBCOPENLIB [Internet]. Outubro, 2022. Disponível em: <https://sol.sbc.org.br/index.php/ihc/article/view/22280>
- [18] TREMAUD, Maela. **What do digital tools add to classical tools for sociocommunicative and adaptive skills in children with Autism Spectrum Disorder?** CAIRN.INFO [Internet]. Abril, 2021. Disponível em: <https://www.cairn.info/revue-l-annee-psychologique-2021-4-page-361.htm>
- [19] CHAVES, João; PAOLA, Yuska; MARITAN, Tiago. **Digital Resources suiting people on the Autism Spectrum: usability criteria prioritization through crowdsourcing.** SBCOPENLIB [Internet]. Outubro, 2023. Disponível em: <https://sol.sbc.org.br/index.php/ihc/article/view/27563>
- [20] CHAVES, João; PAOLA, Yuska; MARITAN, Tiago. **Crowdsourcing in the prioritization of accessibility and usability criteria for autism.** ResearchGate [Internet]. Janeiro, 2024. Disponível em: [https://www.researchgate.net/publication/377216607\\_Crowdsourcing\\_in\\_the\\_prioritization\\_of\\_accessibility\\_and\\_usability\\_criteria\\_for\\_autism](https://www.researchgate.net/publication/377216607_Crowdsourcing_in_the_prioritization_of_accessibility_and_usability_criteria_for_autism)
- [21] BATISTA, Mirian. **Relatório de Estágio Curricular Supervisionado Obrigatório Interno.** Universidade Federal da Paraíba. 2021.
- [22] SILVA, Layslene. **Relatório de Estágio Curricular Supervisionado Obrigatório Interno.** Universidade Federal da Paraíba. 2021.
- [23] NUNES, Glênio. **Catálogo de exemplos para as recomendações do AutismGuide com base na avaliação do Spirit TEA.** Universidade Federal da Paraíba. Junho, 2022.
- [24] SILVA, Layslene; BATISTA, Mirian. **Protótipo de interface: AutismGuide.** FIGMA [Internet]. Novembro, 2021. Disponível em: <https://www.figma.com/design/20okXpNtgx1SiYSeM9R80A/AutismGuide?node-id=0-1>

## APÊNDICES

### Apêndice A - Tabelas do banco de dados do AutismGuide

#### Tb\_product

Esta tabela armazena informações básicas sobre os produtos que são avaliados no AutismGuide. Cada produto é identificado unicamente, facilitando referências em outras partes do sistema.

Campos:

- **id (Integer):** Identificador único para cada produto
- **name (character varying(25)):** Nome do produto

#### Tb\_evaluator

Esta tabela contém dados sobre os avaliadores, que realizam as avaliações dos produtos digitais.

Campos:

- **id (Integer):** Identificador único para cada avaliador
- **name (character varying(25)):** Nome do avaliador
- **profession (character varying(25)):** Profissão do avaliador
- **performance\_exp (character varying(25)):** Nível de experiência do avaliador com relação ao conhecimento acerca do espectro autista (Muito Experiente, Experiente, Inexperiente)
- **app\_exp (character varying(25)):** Nível de experiência do avaliador com relação ao recurso digital avaliado (Muito Experiente, Experiente, Inexperiente)

- **ag\_exp (character varying(25))**: Nível de experiência do avaliador com relação a utilização AutismGuide (Muito Experiente, Experiente, Inexperiente)
- **evaluation\_date (character varying(25))**: Data de registro da avaliação
- **profile (character varying(25))**: Perfil do Avaliador

### **Tb\_product\_instance**

Esta tabela gerencia as instâncias específicas dos produtos, permitindo múltiplas avaliações de diferentes versões ou configurações do mesmo produto.

Campos:

- **id (Integer)**: Identificador único para cada instância de produto
- **product\_id (Integer)**
- **version (character varying)**: Versão do produto
- **platform (character varying)**: Plataforma utilizada na avaliação
- **operational\_system (character varying)**: Sistema operacional utilizado na avaliação
- **comment (text)**: Comentários adicionais sobre a instância do produto e processo de avaliação.

### **Tb\_evaluation**

Armazena registros de cada avaliação realizada, vinculando-a ao avaliador e à instância do produto avaliado.

Campos:

- **id (Integer)**: Identificador único para cada avaliação
- **product\_instance\_id (Integer)**: Chave estrangeira que referencia uma instância do produto

- **evaluator\_id (Integer):** Chave estrangeira que referencia um avaliador
- **creation\_date (character varying):** Data de criação da avaliação

### **Tb\_category**

Categorias são usadas para agrupar critérios de avaliação, facilitando a organização e a visualização das métricas de avaliação.

Campos:

- **id (Integer):** Identificador único para cada categoria
- **title (character varying):** Título da categoria
- **description (character varying):** Descrição da categoria
- **default\_color (character varying):** Cor da categoria
- **icon (character varying):** Ícone da categoria

### **Tb\_recommendations**

Recomendações são sugestões ou diretrizes específicas dentro de cada categoria, destinadas a orientar os avaliadores durante o processo de avaliação.

Campos:

- **id (Integer):** Identificador único para cada recomendação
- **category\_id (character varying):** Chave estrangeira que referencia uma categoria
- **description (text):** Descrição da recomendação
- **recommended\_importance (Integer):** Importância recomendada como default para recomendação

## Tb\_criteria

Esta tabela armazena os critérios específicos avaliados em cada recomendação, associados a uma avaliação particular, permitindo uma análise detalhada e sistemática das características do produto.

Campos:

- **id (Integer):** Identificador único para cada critério
- **recommendation\_id (character varying):** Chave estrangeira que referencia uma recomendação
- **evaluation\_id (character varying):** Chave estrangeira que referencia uma avaliação
- **importance (Integer):** Nível de importância da recomendação
- **conformity (Integer):** Nível de conformidade da recomendação
- **comment (text):** Comentário sobre o recomendação
- **conclusion (boolean):** Conclusão de se a recomendação está com campos obrigatórios respondidos ou não