

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

EDUARDO AMADOR LUCAS

**Sistema de Gestão e Acompanhamento
Educativo**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Prof. Dr. Leandro Krug Wives
Co-orientador: Prof. Ms. Francisco Dutra dos
Santos Jr.

Porto Alegre
2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Wladimir Pinheiro do Nascimento

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Raul Fernando Weber

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“Ensina à criança o caminho que ela deve seguir
mesmo quando envelhecer, dele não há de se afastar.”*
— PROVÉRBIOS, 22, 6 - BÍBLIA SAGRADA VERSÃO AVE MARIA

AGRADECIMENTOS

Agradeço a todos que conviveram comigo neste último ano e principalmente neste último mês, pelo apoio, compreensão e/ou a paciência, pois, não bastasse esta monografia, também havia meu casamento para organizar e para participar como noivo :o. Mas preciso citar de forma explícita meus orientadores, professores Leandro e Chico pela compreensão pela situação e pelos ensinamentos que distribuíram fortemente a mim.

Cito também uma família de origem, meus pais e minha 'irmã' Cris, principalmente pelo amor e apoio que dispõe sempre a mim.

Também a minha, antes noiva, agora esposa Fran por não desistir de mim nunca, mesmo quando tenho poucas esperanças. Obrigado também pelo amor que me dispõe todos os dias em sua fala e suas atitudes, incluindo o lanche que me entregou para poder concluir esse trabalho. :)

E por último a Deus, princípio e fim de todas as coisas, causa última de tudo.

RESUMO

Este estudo tem por objetivo descrever o projeto e o desenvolvimento de um sistema para auxílio dos professores de atendimento educacional especializado em salas de recursos, que permita um acompanhamento de alunos com necessidades educacionais especiais ao longo do processo de escolarização, que seja acessível e fundamentado teoricamente. Esse sistema foi denominado de Sistema de Gestão e Acompanhamento Educacional. Foi usada a metodologia ágil com alguns elementos da tradicional. O desenvolvimento do sistema resultou em um protótipo utilizável que pôde ser apresentado aos usuários para avaliação.

Palavras-chave: Sistema web. Javascript. MongoDB. Gestão educacional especializado. NEE.

Educational Management and Monitoring System

ABSTRACT

This study aims to describe a system web of help for teachers of educational services specialized, which can be an accompaniment of children with special educational needs throughout the schooling process, which is accessible and theoretically grounded. This system was called “Educational Management and Monitoring system”. An agile methodology was used with some elements of the traditional one. The development of the system resulted in a usable prototype that can be presented to the user for evaluation.

Keywords: Web APP, Javascript, MongoDB, Specialized education, Special Necessities.

LISTA DE FIGURAS

| | | |
|-------------|--------------------------------------------------------------------------------------|----|
| Figura 3.1 | Estrutura e fluxo de dados do Flux numa aplicação | 23 |
| Figura 4.1 | Página Inicial | 38 |
| Figura 4.2 | Login do sistema..... | 38 |
| Figura 4.3 | Cadastro de Usuário do sistema..... | 39 |
| Figura 4.4 | Lista de alunos do professor | 39 |
| Figura 4.5 | Cadastramento de Estudantes - aba 1 | 41 |
| Figura 4.6 | Cadastramento de Estudantes - aba 2 | 42 |
| Figura 4.7 | Cadastramento de Estudantes - aba 3 | 43 |
| Figura 4.8 | Lista de Escolas | 44 |
| Figura 4.9 | Cadastro de Escolas | 44 |
| Figura 4.10 | Menu do aluno - Todas as opções de documento e acompanhamentos do aluno..... | 45 |
| Figura 4.11 | Acompanhamento por foto ou video | 46 |
| Figura 4.12 | Acompanhamento de texto somente | 47 |
| Figura 4.13 | Cadastro de Parecer | 48 |
| Figura 4.14 | Cadastro do PDI - Dados de identificação e histórico do estudante | 49 |
| Figura 4.15 | Cadastro do PDI - situação das habilidades..... | 50 |
| Figura 4.16 | Cadastro do PDI - Proposições | 51 |
| Figura 4.17 | Adequação curricular - Aba 1 - Dados da matéria ou área do conhecimento | 53 |
| Figura 4.18 | Adequação curricular - Aba 2 - Matriz Histórica | 54 |
| Figura 4.19 | Adequação curricular - Aba 3 - Matriz Necessidade..... | 55 |
| Figura 4.20 | Adequação curricular - Aba 4 - Matriz Programação Curricular | 56 |
| Figura 4.21 | Adequação curricular - Aba 4 - Matriz Sugestões para Adequação Cur- ricular | 57 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|-------------------------------------------|
| NEE | Necessidades Educacionais Especiais |
| AEE | Atendimento Educacional Especializado |
| PDI | Plano de desenvolvimento individual |
| CRUD | acrónimo de Create, Read, Update e Delete |
| API | Application Programming Interface |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| Sass | Syntactically Awesome StyleSheets |
| JSON | JavaScript Object Notation |
| SPA | Single Page Application |
| BSON | Binary JSON |
| AJAX | Asynchronous Javascript and XML |
| XML | Extensible Markup Language |
| JSX | JavaScript eXtension |
| DOM | Document Object Model |
| HTTP | HyperText Transfer Protocol |

SUMÁRIO

| | |
|--------------------------------------------------------------------------------------|-----------|
| 1 INTRODUÇÃO | 11 |
| 1.1 A proposta..... | 11 |
| 1.2 Objetivo..... | 12 |
| 1.3 Estrutura do Texto | 12 |
| 2 DEFINIÇÃO DO SISTEMA | 14 |
| 2.1 Requisitos e restrições..... | 14 |
| 2.2 Funcionalidade 1 - Professores/Usuários | 16 |
| 2.3 Funcionalidade 2 - Estudante | 16 |
| 2.4 Funcionalidade 3 - Escola..... | 17 |
| 2.5 Funcionalidade 4 - Registro de Acompanhamento | 17 |
| 2.6 Funcionalidade 5 - PDI..... | 17 |
| 2.7 Funcionalidade 7 - Parecer: | 18 |
| 2.8 Funcionalidade 8 - Linha do Tempo do Estudante | 19 |
| 3 ARQUITETURA DO SISTEMA E TECNOLOGIAS UTILIZADAS | 20 |
| 3.1 Arquitetura do Front-End | 20 |
| 3.1.1 HTML | 20 |
| 3.1.2 CSS | 20 |
| 3.1.3 JavaScript | 21 |
| 3.1.4 JSON..... | 21 |
| 3.1.5 SPA..... | 21 |
| 3.1.6 React | 22 |
| 3.1.7 Redux e a Arquitetura Flux..... | 22 |
| 3.2 Back-end | 23 |
| 3.2.1 NodeJs..... | 23 |
| 3.2.2 Express..... | 23 |
| 3.2.3 MongoDB | 23 |
| 3.2.4 Interface da API REST | 24 |
| 3.2.4.1 Entidade API User: para controle de usuários | 25 |
| 3.2.4.2 Entidade API Student: para controle de estudante | 26 |
| 3.2.4.3 Entidade API School: para controle de escola..... | 27 |
| 3.2.4.4 Entidade API Files: para controle de acompanhamento com fotos ou vídeo. | 28 |
| 3.2.4.5 Entidade API Adaptation: para controle de adequações curriculares..... | 29 |
| 3.2.4.6 Entidade API Plan: para controle de PDIs..... | 30 |
| 3.2.4.7 Entidade API Judgement: para controle de Parece..... | 31 |
| 3.2.5 Modelo de dados de armazenamento persistente..... | 32 |
| 3.2.5.1 User:..... | 32 |
| 3.2.5.2 Student | 32 |
| 3.2.5.3 School | 34 |
| 3.2.5.4 File | 35 |
| 3.2.5.5 Document..... | 35 |
| 4 TELAS DO SISTEMA E NAVEGAÇÃO | 37 |
| 4.1 Páginas Iniciais e Login..... | 37 |
| 4.2 Cadastro de Usuário | 37 |
| 4.3 Listagem de alunos..... | 37 |
| 4.4 Cadastramento de Estudantes. | 40 |
| 4.5 Listagem e cadastro de Escolas..... | 40 |
| 4.6 Menu do aluno..... | 45 |

| | |
|-------------------------------------------|-----------|
| 4.7 Acompanhamento do aluno. | 46 |
| 4.7.1 Foto/Vídeo | 46 |
| 4.7.2 Texto..... | 47 |
| 4.8 Parecer | 47 |
| 4.9 PDI | 48 |
| 4.10 Adequação curricular | 58 |
| 5 CONCLUSÃO | 59 |
| REFERÊNCIAS | 60 |

1 INTRODUÇÃO

A educação de alunos com necessidades educacionais especiais (daqui em diante, NEE) tem mudado por entendimento dos especialistas em educação para que esses alunos frequentem preferencialmente escolas e salas de aulas regulares, sendo um entendimento divergente ao anterior que privilegiava alunos com NEE de maneira a estudarem em escolas especializadas. Com a introdução dos alunos nas salas de aulas regulares, tornou-se necessário a implementação das chamadas *Salas de Recursos*. Essas salas têm um professor especializado com o objetivo de apoiar na escolarização desses alunos a partir de uma perspectiva individualizada às necessidades de cada um deles, normalmente no turno inverso ao da sala de aula regular. Nessas salas de recursos são desenvolvidas atividades de compreensão dos processos mentais e do funcionamento da aprendizagem do aluno, propostas por meio de estratégias lúdicas, sistemas ativos de ensino e recursos diferenciados.

Os professores especializados da sala de recursos devem analisar constantemente as especificidades dos alunos e seu desenvolvimento ao longo da escolarização, acompanhando-os em vários momentos e etapas do ensino. No decorrer desse processo é analisado o desempenho, bem como a geração de documentos e realização de planejamentos e adequações específicas para cada aluno e para cada disciplina ou conteúdo. Esses documentos e informações são elaborados em diferentes formatos, alguns em formulários de papel, enquanto outros em forma digital, tais como fotos e filmes capturados em *smartphones*, *tablets* ou computadores, além disso ficam armazenados em lugares diferentes, como na secretaria da escola, na própria sala de recurso ou ainda em computadores ou *smartphones* particulares, isso quando não ficam somente nas memórias dos professores envolvidos. Tudo isso dificulta a análise longitudinal desses alunos, sendo que os professores valem-se muito de sua memória para fazer uma análise da evolução e aprendizado dos alunos, já que alunos com NEE não têm obrigatoriedade de aprovação em avaliações como provas e testes.

1.1 A proposta

Com base na situação descrita, torna-se necessário organizar o trabalho e principalmente as informações obtidas nos atendimentos da sala de recursos, bem como os documentos que o professor gera a partir desses atendimentos. A partir disso torna-se

necessário o desenvolvimento de uma solução que contemple esses problemas e ainda facilidade o trabalho do professor na confecção dos documentos como pareceres, adequações curriculares ou Planos de Desenvolvimento Individualizado (PDI). Para isso, este trabalho consiste em planejar e desenvolver um sistema educacional de gerenciamento e de acompanhamento de alunos da sala de recursos.

Cabe salientar que já há dois sistemas desenvolvidos para este fim, sendo o primeiro, *SIR-EDU*, chegou ao teste com usuários reais, mas não houve muita aceitação do mesmo, pois, segundo relatos deles, o sistema apresenta problemas de usabilidade e não apresenta versão *mobile* e nem responsiva. Além disso, (FERREIRA, 2017) propôs uma nova arquitetura e uma API para o *SIR-EDU*, mas o mesmo não chegou a ser colocado em uso.

O sistema aqui proposto é uma extensão do trabalho de (FERREIRA, 2017), os quais estão baseados, tanto conceitualmente quanto teoricamente, no trabalho de pesquisa do coorientador deste trabalho, Prof. Ms. Francisco Dutra dos Santos Jr., o qual desenvolve tese de doutorado no Programa de Pós-Graduação em Informática na Educação (PGIE) da UFRGS, e usará esse sistema para gerar os dados para suportar sua pesquisa (SANTOS Jr; LUCAS; WIVES, 2018). Algumas decisões do projeto deste sistema foram apoiadas pelas informações descritas no trabalho de (SANTOS Jr; LUCAS; WIVES, 2018), principalmente os cadastros de *adequação curricular* e o *PDI*, visto que não há padrão amplamente aceito pelos professores da sala de recurso para esses documentos.

1.2 Objetivo

Desenvolver um sistema para servir para auxílio dos professores de atendimento escolar da *salas de recursos*, a fim de armazenar, organizar e facilitar o trabalho. Espera-se ao final que possa-se apresentar a usuários potenciais do sistema para que eles possam dar seus *feedbacks*.

1.3 Estrutura do Texto

De maneira a atingir o objetivo proposto, o projeto e o desenvolvimento do sistema foi realizado sendo descrito nos próximos capítulos. O texto está, portanto, organizado da seguinte forma: O Capítulo 2, descreve o sistema a ser desenvolvido, seus requisitos

e funcionalidades. O Capítulo 3 descreve a arquitetura do sistema e as tecnologias utilizadas na construção do sistema. O Capítulo 4 descreve as telas do sistema com suas funcionalidade e navegação. Finalmente, o último capítulo apresenta as conclusões do trabalho, incluindo suas limitações e trabalhos futuros.

2 DEFINIÇÃO DO SISTEMA

Conforme já mencionado, já há dois sistemas desenvolvidos para o gerenciamento de salas de recursos, o SIR-EDU¹, porém o mesmo apresenta alguns problemas de usabilidade e não possui versão específica para plataformas móveis, o que é considerado fundamental pelos profissionais da sala de recursos. Sendo que não passou da fase de testes.

Também há a versão desenvolvida por (FERREIRA, 2017), O seu foco foi na API de acesso ao banco e de gestão dos dados relacionados com alunos, professores e arquivos multimídia, em especial fotos, som e vídeos. Por isso, sua interface apresenta problemas de navegação por não ter sido foco do trabalho não estando pronta para usuários finais, além de não apresentar todas as funcionalidades necessárias como as escolas e os documentos do estudante. Esses sistemas foram usados como base para a identificação de requisitos e campos para o novo sistema.

Diante disso, o presente trabalho estende e amplia o trabalho de (FERREIRA, 2017), refazendo a interface web do sistema, mas respeitando a interface da API proposta naquele trabalho, o qual já apresentava um protótipo de aplicativo móvel e que deve continuar funcionando com pequenos ajustes, como, por exemplo, a troca do endereço de servidor.

Para a elaboração do sistema, primeiramente foram levantados seus requisitos a partir da análise dos sistemas anteriores e também do *feedback* dos seus usuários, além de entrevistas e encontros com o coordenador, pesquisador que idealizou o sistema.

2.1 Requisitos e restrições

O sistema tem por usuário principal os professores da sala de recursos e as seguintes características e restrições foram consideradas:

- **Escolas com computadores não muito potentes.** Principalmente em escolas públicas, que é um possível lugar onde o sistema pode ser útil, não há computadores muito potentes. Por isso, é necessário escolher tecnologias evitem tornar o sistema lento, por exemplo, diminuindo o processamento na máquina do usuário.
- **Internet.** Foi levado em consideração que há acesso a Internet difundido, porém a

¹<<http://sir-edu.net/>>

Internet não é rápida.

- **Resolução baixa de monitores.** Também não há garantia de monitores grandes, nem com boa resolução. Então foi considerada uma resolução de 1024x768.

O sistema foi organizado em um sistema web para acesso pelo computador ou *tablet*, um banco de dados e uma API para comunicação entre o sistema e o banco. O fluxo básico é: o professor cadastra-se, depois cadastra seus alunos, cadastra os acompanhamentos e os documentos para cada um deles e visualiza/analisa os dados.

Para descrever, documentar e validar as funcionalidades a serem desenvolvidas, foram usadas *user stories*, abaixo é citado aquelas que foram consideradas. Os principais pontos do sistema que necessitavam ser desenvolvido eram a interface web (que fosse funcional) e usando as entidades presentes na API e banco de dados já desenvolvidos anteriormente, e o desenvolvimento de todos do documentos e acompanhamentos dos estudantes (interface, API e banco de dados).

Embora, no levantamento inicial, foi elencados diversos tipos de usuários com permissões restritas ao sistema, neste trabalho somente foram consideradas as histórias do papel de *Professor* da sala de recursos. Para a versão básica proposta, os outros papéis não seriam essenciais, e os mesmos podem ficar para futuras versões. Os outros usuários (papéis) seriam:

- Outros professores que não sejam da sala de recursos.
- Outros profissionais como médicos, psicólogos.
- Pais ou responsáveis de alunos.
- Diretores, supervisores de rede de escolas e pesquisadores da universidade.

As atividades do sistema na versão proposta ficaram definidas como as seguintes:

1. Funcionalidade 1 - Professores/Usuários: Como professor da sala de recurso quero cadastrar-me no sistema e, eventualmente, alterar meu cadastro.
2. Funcionalidade 2 - Estudante: Como professor da sala de recurso quero cadastrar, alterar e excluir estudante.
3. Funcionalidade 3 - Escola: Como professor da sala de recurso quero cadastrar, alterar e excluir informações sobre a escola.
4. Funcionalidade 4 - Registro de acompanhamento: Como professor da sala de recurso quero cadastrar um registro de acompanhamento do estudante através de imagem, vídeo ou texto.

5. Funcionalidade 5 - PDI: Como professor da sala de recurso quero cadastrar um PDI (plano de desenvolvimento individual), que é um documento do estudante.
6. Funcionalidade 6 - Adequação Curricular: Como professor da sala de recurso quero cadastrar uma adequação curricular, que é um documento do aluno.
7. Funcionalidade 7 - Parecer: Como professor da sala de recurso quero cadastrar um parecer, que é um documento do aluno.
8. Funcionalidade 8 - Linha do Tempo do Estudante: Como professor da sala de recurso quero ver os acompanhamentos, os documentos (PDIs, adequações curriculares e pareceres) que são de um estudante, em uma *timeline*, e usá-la para referenciar outros documentos.

Segue o detalhamento de cada atividade.

2.2 Funcionalidade 1 - Professores/Usuários

Os professores são os usuários do sistema. A eles é permitido cadastrar-se e efetuar o login. A tela de cadastro dos usuários possui os seguintes dados:

- Nome;
- Endereço;
- Telefone;
- E-mail para login;
- Senha para login;
- Escola.

2.3 Funcionalidade 2 - Estudante

Os estudantes que podem ser cadastrados pelo professor (usuário) responsável da sala de recursos devem ter autorização digitalizada (Termo de Aprovação) dos pais ou responsáveis. O professor deve informar os seguintes dados do estudante:

1. Informações pessoais;
2. Foto;
3. Endereço;

4. Dados de contatos dos pais e do responsável;
5. Dados da escola, matrícula, turma e turno;
6. Dados referentes à tipologia de deficiências ou Necessidades Educacionais (Específicas) Especiais (NEE) do aluno;
7. Histórico do aluno e encaminhamento ao AEE (Atendimento Educacional Especializado);
8. Termo de aprovação dos pais.

2.4 Funcionalidade 3 - Escola

Informações da escola que o professor deve cadastrar:

1. Nome da Escola;
2. Endereço;
3. Contato.

2.5 Funcionalidade 4 - Registro de Acompanhamento

O professor deve cadastrar os registros do acompanhamento ou atendimento de cada estudante. Essas informações servem para armazenar as situações ocorridas ou desenvolvidas com os estudante. Esses acompanhamentos podem ter o formato de foto, vídeo ou somente texto, e os seguintes campos:

1. Arquivo de media (foto ou vídeo);
2. Data e Hora;
3. Texto com comentário.

2.6 Funcionalidade 5 - PDI

Periodicamente o professor deve fazer um PDI (Plano de Desenvolvimento Individualizado) por estudante, o período depende da instituição o qual ele esteja vinculado. O PDI é um instrumento utilizado para planejar o currículo escolar às necessidades dos alunos NEE. Os dados para o preenchimento abrangem os seguintes campos:

1. Identificação do aluno, de suas atividades, período turno de aula na sala de recursos;
2. Histórico do aluno, familiar e de atendimento no AEE;
3. Situação do desenvolvimento do aluno em cada habilidade;
4. Ação que estão sendo realizadas;
5. Ações propositivas.

Para cada disciplina, matéria ou área do conhecimento, dependendo da instituição, o professor deve fazer uma adequação curricular para o estudante. Essa adequação leva em conta as necessidades específicas e por isso deve ser individualizada. Para realizar a adequação o professor deve preencher 4 matrizes pedagógicas, além de outros dados como segue, conforme (SANTOS Jr; LUCAS; WIVES, 2018).

1. Dados da matéria ou área do conhecimento;
2. Matriz 1, histórico do aluno;
3. Matriz 2, necessidade especiais que o aluno possui;
4. Matriz 3, Programação Curricular;
5. Matriz 4, Sugestões para Adequação Curricular.

2.7 Funcionalidade 7 - Parecer:

O professor pode realizar pareceres periodicamente ou devido a alguma demanda específica. Cada parecer tem um dos seguintes tipos:

1. Avaliação inicial: ocorre quando um estudante é encaminhado para avaliação diagnóstica pela Sala de Recursos;
2. Avaliação processual: pareceres que ocorrem no meio do processo de AEE, conforme necessidade ou funcionamento da instituição de ensino;
3. Avaliação Final: ocorre quando completa um ciclo letivo, na troca de professor do AEE ou quando um aluno é encaminhado de ou para outras instituições de ensino.
4. Encaminhamento para pais ou responsáveis: parecer para ser enviado aos pais ou responsáveis;
5. Encaminhamento para outros profissionais: parecer para ser encaminhado a outros profissionais, como médicos, fisioterapeutas, psicólogos, ou outros conforme a necessidade.

Além do tipo, os campos do parecer são:

1. Data;
2. Título;
3. Texto-livre.

2.8 Funcionalidade 8 - Linha do Tempo do Estudante

O sistema deve exibir de forma cronológica e simples para consulta rápida as informações dos registros de acompanhamento e dos documentos do estudante (PDI, Adequação curricular e Parecer). Além disso quando o professor for compor algum documento do estudante (parecer, PDI, etc.), deve poder referenciar ou incluir outros acompanhamentos ou documentos dele através da linha cronológica (*timeline*). Por exemplo, quando um professor estiver escrevendo um parecer sobre a melhora de uma criança na sua coordenação motora fina, ele pode referenciar uma foto que mostre um desenho ou um texto construído pelo referido estudante.

3 ARQUITETURA DO SISTEMA E TECNOLOGIAS UTILIZADAS

Este capítulo descreve os componentes do Sistema, sua arquitetura, e as tecnologias utilizadas no seu desenvolvimento.

Sistema foi desenvolvido para Web e será acessado pela Internet. A estrutura do sistema baseia-se na arquitetura cliente-servidor para dividir suas responsabilidades, sendo o “Servidor” responsável pelo armazenamento das informações, a autenticação do usuário, etc. O “cliente” é responsável pela exibição do conteúdo, comunicação com o servidor, validação básica dos campos, roteamento da aplicação.

Abaixo segue o detalhamento dos elementos da arquitetura e as tecnologias utilizadas em cada um deles.

3.1 Arquitetura do Front-End

Como em quase toda aplicação Web, o sistema utiliza HTML, CSS e Javascript. Mas bibliotecas auxiliares foram utilizadas. Elas são descritas nas subseções seguintes.

3.1.1 HTML

O HTML (HyperText Markup Language) é a linguagem de marcação básica das páginas Web. Foi utilizada sua versão mais recente, o HTML 5 (MOZILLA, 2018).

3.1.2 CSS

O CSS (Cascading Style Sheets) é a linguagem que os navegadores padronizada pela W3C (W3C, 2018) para a descrição de layout e estilo de páginas Web. Porém, neste projeto foi usada a tecnologia SASS (Syntactically Awesome StyleSheets) que é um pré-processador de CSS para escrever estilos com um super-conjunto do CSS e usar um processador para gerar CSS (SASS, 2018).

3.1.3 JavaScript

Em termos de linguagem, JavaScript foi escolhido para ser utilizado tanto no *front-end* quanto no *back-end* da aplicação. No *front-end* foi utilizada a versão Javascript 6 (ECMAScript 6 (ECMA INTERNATIONAL, 2015)) e no *back-end* a versão 5.1 (ECMAScript 5.1 (ECMA INTERNATIONAL, 2011)). Embora a versão 6 já seja suportada por navegadores mais modernos, como Chrome a partir da versão 58, Firefox da versão 54 e Safari versão 10 Refsnes Data (2018a), foi utilizado o Babel (ver a seguir) para deixá-lo compatível com os navegadores mais antigos, visto que a tecnologia utilizada nas escolas, principalmente públicas, não é a mais atual.

Babel (BABEL, 2018) é um “compilador” de Javascript, i.e., uma ferramenta para converter ECMAScript 2015+ em código compatível com versões anteriores de Javascript para rodar em navegadores antigos. O Babel também é compatível com a sintaxe JSX que é usada para biblioteca React (a qual será abordada mais adiante no documento).

3.1.4 JSON

JSON (JavaScript Object Notation) é um formato de intercâmbio de informações entre sistema. Ele é fácil para humanos lerem e escreverem, além de serem compreendidos e gerados por máquinas. Ele é baseado na versão 3 do Javascript de 99 e por isso é muito fácil de trabalhar com esse formato no Javascript atual (JSON, 2018). Ele foi usado no sistema para comunicação entre a API do servidor e a aplicação do front-end, e também foi usada uma variação chamada BJON no banco de dados Mongo.

3.1.5 SPA

A aplicação roda no formato de *Single Page Application* (SPA), ou seja, uma aplicação que roda numa única página e "imita" o comportamento de aplicações que são instaladas no sistema operacional, não recarregando toda a página a cada clique do usuário, oferecendo rapidez de resposta ao usuário, etc (WIKIMEDIA FOUNDATION, INC., 2018). Para isso, no entanto, a aplicação SPA precisa lidar com arquivos e dados dinamicamente carregados de forma assíncrona com AJAX, roteamento e componente reutilizáveis, entre outras funcionalidades.

3.1.6 React

Atualmente existem vários *frameworks* ou bibliotecas Javascript. Para criar e gerenciar a interface do sistema proposto foi utilizado o ReactJS, uma dos frameworks mais usados, e que permite gerar aplicações SPA.

React é uma biblioteca criada e mantida pelo Facebook. Das características principais, destacam-se duas, sendo a primeira o uso do JSX que é uma sintaxe parecida com o XML ou HTML para escrever o Javascript que o React precisa para gerar os elementos que aparecem na tela, tornando essa parte parecida com o HTML normal (FACEBOOK INC., 2018b).

A outra é o Virtual DOM (virtual Document Object Model), que é uma estrutura de cache na memória usada para pelo programador para trabalho podendo alterá-la e fazer processamentos necessários, para quando alterar a página que é exibida para o usuário isso seja feita de modo eficiente. React também trabalha com o conceito de Web Component que é capacidade de criar tags HTML personalizadas para a página (REFS-NES DATA, 2018b) que podem ser reutilizados conforme necessidade (GITHUB, INC., 2018).

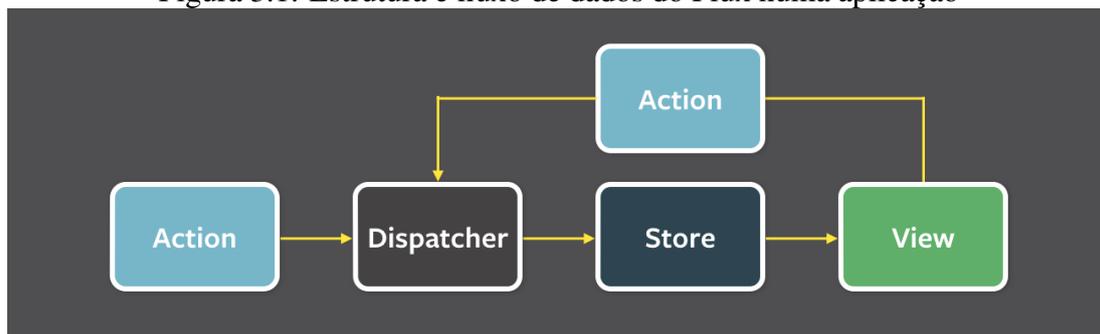
O React também gerencia o sistema de rotas de URLs internas para a aplicação com o React Router. A cada rota é associado um componente controlador de uma tela.

3.1.7 Redux e a Arquitetura Flux

O Redux foi usado em complemento ao React. Ele é uma biblioteca que implementa a arquitetura FLUX foi proposta e é usada para o desenvolvimento de aplicações cliente-servidor pelo Facebook também. O Flux é usado para construir aplicações que trabalhem de forma reativa. Basicamente uma forma de fluxo unidirecional de dados entre eventos e ouvintes (FACEBOOK INC., 2018a) (Figura 3.1).

O Redux divide a aplicação em 3 partes: a *view* que o React cuida, o *dispatcher* e o *store*. O *dispatcher* é responsável por receber ações e o único a alterar o *store*. A *view* somente pode ler o *store*, assim definindo um fluxo unidimensional na aplicação 3.1.

Figura 3.1: Estrutura e fluxo de dados do Flux numa aplicação



Fonte: Facebook Inc. <<https://facebook.github.io/flux/docs/in-depth-overview.html>>

3.2 Back-end

O servidor servidor fica responsável pelo armazenamento das informações da aplicação. E se comunica através da API definida em (FERREIRA, 2017). Seguem detalhes sobre as tecnologias utilizadas no servidor de *back-end*.

3.2.1 NodeJs

O servidor da aplicação é feito em NodeJS, que é um servidor que interpreta Javascript (NODE.JS FOUNDATION, 2018b). Ele aceita as requisições através do Express.

3.2.2 Express

O Express (NODE.JS FOUNDATION, 2018a) gerencia o sistema de roteamento para processar requisições do *Front-End* da aplicação. Esse sistema de roteamento foi criado para processar e responder requisições correspondentes ao seu método do protocolo HTTP.

3.2.3 MongoDB

Foi escolhido do banco dados MongoDB para armazenamento. MongoDB é um banco de dados NoSQL, i.e., não usa modelo relacional. Ele é orientado a documentos e tem alta performance, alta disponibilidade e escala automaticamente (MONGODB INC., 2018d).

Um registro no MongoDB é um “documento” o qual é uma estrutura composta por chave e valor. Os documentos são parecidos com objetos JSON. Os valores podem ter múltiplos valores, ou seja podem ser *arrays*, outros documentos, objetos, etc (MONGODB INC., 2018c). O MongoDB armazena documentos em coleções, que são como tabelas nos banco de dados relacionais (MONGODB INC., 2018b). Além disso, as coleções têm esquema flexível, ou seja, podem acrescentar campos sem muito esforço (MONGODB INC., 2018a).

O MongoDB foi escolhido principalmente devido ao *Schema flexível* que permitiu que, por exemplo, fosse possível com uma única entidade chamada *Document* pudesse representar os acompanhamentos e documentos, pois há somente um relacionamento que os estudantes em todos eles (como mostrado em 3.2.5.5).

3.2.4 Interface da API REST

Única forma de comunicação da aplicação com o servidor para receber dados é através de requisições a uma API que faz a interface com o banco de dados.

Essa API fornece isolamento entre a aplicação e banco de dados a fim de que esta não precise conhecer detalhes desnecessários tais como qual banco foi utilizada, qual as tabelas/coleções existentes, etc..

Ela também fornece integração com o aplicativo mobile desenvolvido anteriormente por (FERREIRA, 2017). Sendo um dos requisitos do desenvolvimento do sistema a alteração desta interface mantendo a sua compatibilidade.

O acesso a API dá-se por envio e recebimento de requisições HTTP, essas precisam enviar o token de autenticação válido para poder interagir com o sistema, exceto a requisição para login.

As entidades da API são:

- User
- Student
- School
- Files
- Adaptation
- Plan
- Judgement

Abaixo o detalhamento de cada uma delas:

3.2.4.1 Entidade API User: para controle de usuários

Login no sistema

- URI: /user/authenticate
- Parâmetro: objeto JSON com email e senha.
- Método POST
- Resposta esperada: token de autenticação.

Incluir novo Usuário

- URI: /user/register
- Parâmetro: objeto JSON com os dados do usuário.
- Método POST
- Resposta esperada: JSON com o usuário cadastrado.

Obter Usuário do parâmetro

- URI: /user/[id]
- Parâmetro: id do usuário
- Método GET
- Resposta esperada: JSON com o usuário encontrado ou vazio.

Obter lista de Usuários

- URI: /user/
- Parâmetro: inexistente.
- Método GET
- Resposta esperada: listagem com todos usuários.

Alterar usuário do parâmetro

- URI: /user/[id]
- Parâmetros: [id] do usuário e campo a serem atualizado.
- Método PUT
- Resposta esperada: JSON com o usuário alterado.
- Altera usuário do parâmetro.

Exclui usuário do parâmetro

- URI: /user/[id]
- Parâmetros: [id] do usuário
- Método DELETE
- Resposta esperada: JSON com a usuário.
- Exclui usuário do parâmetro.

3.2.4.2 Entidade API Student: para controle de estudante

Incluir novo Estudante

- URI: /students/register
- Parâmetros: objeto JSON com os dados do estudante.
- Método POST
- Resposta esperada: JSON com o estudante cadastrado.

Obter Usuário do parâmetros

- URI: /students/[id]
- Parâmetros: [id] do estudante
- Método GET
- Resposta esperada: JSON com o estudante encontrado ou vazio.

Obter lista de Usuários

- URI: /students/
- Parâmetros: inexistente.
- Método GET
- Resposta esperada: listagem com todos estudante.

Alterar usuário do parâmetro

- URI: /students/[id]
- Parâmetros: [id] do estudante e campos a serem atualizado.
- Método PUT
- Resposta esperada: JSON com o estudante alterado.
- Altera usuário do parâmetro.

Exclui usuário do parâmetro

- URI: /students/[id]
- Parâmetros: [id] do estudante
- Método DELETE
- Resposta esperada: JSON com a estudante.
- Exclui estudante do parâmetro.

3.2.4.3 Entidade API School: para controle de escola

Incluir novo escola

- URI: /schools/register
- Parâmetros: objeto JSON com os dados da escola.
- Método POST
- Resposta esperada: JSON com a escola cadastrado.

Obter escola do parâmetros

- URI: /schools/[id]
- Parâmetros: [id] da escola
- Método GET
- Resposta esperada: JSON com o escola encontrado ou vazio.

Obter lista de escolas

- URI: /schools/
- Parâmetros: inexistente.
- Método GET
- Resposta esperada: listagem com todos escola.

Alterar escola do parâmetro

- URI: /schools/[id]
- Parâmetros: [id] da escola e campos a serem atualizado.
- Método PUT
- Resposta esperada: JSON com a escola alterada.
- Altera usuário do parâmetro.

Exclui escola do parâmetro

- URI: /schools/[id]
- Parâmetros: [id] da escola
- Método DELETE
- Resposta esperada: JSON com a escola.
- Exclui escola do parâmetro.

3.2.4.4 Entidade API Files: para controle de acompanhamento com fotos ou vídeo.

Incluir novo acompanhamento

- URI: /files/register
- Parâmetros: objeto FormData com dos dados do acompanhamento.
- Método POST
- Resposta esperada: JSON com o escola cadastrado.

Obter acompanhamento do parâmetros

- URI: /files/[id]
- Parâmetros: [id] do acompanhamento
- Método GET
- Resposta esperada: download via streaming com o acompanhamento ou vazio.

Obter lista de acompanhamentos

- URI: /schools/[id_estudante]
- Parâmetros: [Id] do estudante.
- Método GET
- Resposta esperada: listagem com todos acompanhamentos do estudante.

Exclui acompanhamento parâmetro

- URI: /files/[id]
- Parâmetros: [id] do acompanhamento
- Método DELETE
- Resposta esperada: JSON com a acompanhamento.
- Exclui acompanhamento do parâmetro.

3.2.4.5 Entidade API Adaptation: para controle de adequações curriculares

Incluir nova adequação curricular

- URI: /adptation/register
- Parâmetros: objeto JSON com os dados da adequação.
- Método POST
- Resposta esperada: JSON com a adequação cadastrado.

Obter adequação curricular do parâmetros

- URI: /adptation/[id]
- Parâmetros: [id] da adequação
- Método GET
- Resposta esperada: JSON com a adequação encontrado ou vazio.

Obter lista de adequações curriculares

- URI: /adptation/id_student
- Parâmetros: [Id] do estudante.
- Método GET
- Resposta esperada: listagem com todos adequação do estudante.

Alterar adequação curricular do parâmetro

- URI: /adptation/[id]
- Parâmetros: [id] da escola e campos a serem atualizado.
- Método PUT
- Resposta esperada: JSON com a adequação alterada.

Exclui adequação curricular do parâmetro

- URI: /schools/[id]
- Parâmetros: [id] da adequação
- Método DELETE
- Resposta esperada: JSON com a adequação .
- Exclui adequação do parâmetro.

3.2.4.6 Entidade API Plan: para controle de PDIs

Incluir novo PDI

- URI: /plan/register
- Parâmetros: objeto JSON com os dados do PDI.
- Método POST
- Resposta esperada: JSON com o PDI.

Obter PDI do parâmetros

- URI: /plan/[id]
- Parâmetros: [id] do PDI
- Método GET
- Resposta esperada: JSON com do PDI encontrado ou vazio.

Obter lista de PDIs

- URI: /plan/id_student
- Parâmetros: [Id] do estudante.
- Método GET
- Resposta esperada: listagem com todos PDI do estudante.

Alterar PDI do parâmetro

- URI: /plan/[id]
- Parâmetros: [id] do PDI e campos a serem atualizado.
- Método PUT
- Resposta esperada: JSON com o PDI alterada.
- Altera PDI do parâmetro.

Exclui PDI do parâmetro

- URI: /plan/[id]
- Parâmetros: [id] do PDI
- Método DELETE
- Resposta esperada: JSON com o PDI.
- Exclui PDI do parâmetro.

3.2.4.7 Entidade API Judgement: para controle de Parece

Incluir novo Parecer

- URI: /judgement/register
- Parâmetros: objeto JSON com os dados do parecer.
- Método POST
- Resposta esperada: JSON com o parecer.

Obter Parecer do parâmetros

- URI: /judgement/[id]
- Parâmetros: [id] do parecer
- Método GET
- Resposta esperada: JSON com o parecer encontrado ou vazio.

Obter lista de Pareceres

- URI: /judgement/id_student
- Parâmetros: [Id] do estudante.
- Método GET
- Resposta esperada: listagem com todos pareceres do estudante.

Alterar Parecer do parâmetro

- URI: /judgement/[id_student]/[id]
- Parâmetros: [id] do estudante, [id] do parecer e campos a serem atualizados.
- Método PUT
- Resposta esperada: JSON com o parecer alterado.
- Resultado: altera o parecer do parâmetro.

Excluir Parecer do parâmetro

- URI: /judgement/[id]
- Parâmetros: [id] do parecer
- Método DELETE
- Resposta esperada: JSON com o parecer.
- Resultado: exclui parecer do parâmetro.

3.2.5 Modelo de dados de armazenamento persistente

O armazenamento permanente é feito em MongoDB, que se dá por meio de coleção, que são análogas as tabelas nos bancos de dados relacionais (MONGODB INC., 2018b). Mas diferentemente destes, o MongoDB tem um Schema flexível, sendo possível incluir com pouco esforço novos campos, ou mesmo não utilizar mais alguns campos já existentes, sem que isso ocupe mais espaço de armazenamento.

As coleções do banco de dados, e o que estas representam são as seguintes:

- User - armazena as informações dos usuários do sistema.
- Student - armazena as informações dos estudantes.
- School - armazena as informações das escolas.
- File - armazena os arquivos de imagens e vídeos dos acompanhamentos do estudante.
- Document - armazena as informações dos acompanhamentos e documentos do estudante.

Abaixo o detalhe de cada coleção do banco de dados:

3.2.5.1 User:

Seguem os campos da entidade User.

- `_id`: String. Gerado automaticamente pelo MongoDB na inclusão do registro.
- `email`: String. Armazena o e-mail do usuário.
- `password`: String. Armazena a senha criptografada.
- `role`: String. Armazena o tipo do usuário (professor ou outro).
- `name`: String. Armazena nome.
- `lastName`: String. Armazena sobrenome.
- `phone`: Number. Armazena número do telefone.
- `adress`: String. Armazena endereço.
- `_school`: ObjectId. Armazena referência para uma escola.

3.2.5.2 Student

Seguem os campos da entidade Student.

- `_id`: String. Gerado automaticamente pelo MongoDB na inclusão do registro.
- `name`: String. Armazena o nome.
- `avatar`: Binary. Armazena a foto do estudante.
- `birthDate`: Date. Armazena aniversário.
- `motherName`: String. Armazena nome da mãe.
- `motherPhone`: String. Armazena telefone da mãe
- `motherEmail`: String. Armazena e-mail da mãe.
- `fatherName`: String. Armazena nome do pai.
- `fatherPhone`: String. Armazena telefone do pai.
- `fatherEmail`: String. Armazena e-mail do pai.
- `responsible`: String. Armazena nome do responsável.
- `responsiblePhone`: String. Armazena telefone do responsável.
- `responsibleEmail`: String. Armazena e-mail do responsável.
- `relationship`: String. Armazena parentesco do responsável.
- `address`: String. Armazena endereço do estudante.
- `address2`: String. Armazena complemento do estudante.
- `address3`: String. Armazena ultima parte do endereço se necessário.
- `district`: String. Armazena o bairro.
- `zipcode`: String. Armazena CEP.
- `city`: String. Armazena a cidade.
- `state`: String. Armazena a UF.
- `country`: String. Armazena o país.
- `school`: String. Armazena a escola do estudante.
- `registration`: String. Armazena a matrícula do estudante.
- `classNumber`: String. Armazena a turma do estudante.
- `series`: String. Armazena o série do estudante.
- `shift`: String. Armazena o turno do estudante.
- `professorNEE`: String. Armazena o professor NEE.
- `orientation`: String. Armazena o orientador.
- `coordination`: String. Armazena o coordenador.
- `routingDate`: String. Armazena a data de encaminhamento para a sala de recurso.

- `routingReason`: String. Armazena a motivo do encaminhamento para a sala de recurso.
- `specialNeeds`: List. Armazena a uma lista de necessidades especiais do estudante.
- `anotherSpecialNeeds`: String. Armazena outras necessidades especiais.
- `cid`: String. Armazena os CIDs (Classificação Internacional de Doenças) do estudante.
- `historical`: String. Armazena o histórico.
- `docParentsApproval`: Binary. Armazena o documento de aprovação dos responsáveis.
- `termOfUse`: Boolean. Armazena a informação se foi assinado ou não o termo de uso.
- `_dateCreate`: Date. Armazena a data de criação do estudante.
- `_dateModifi`: Date. Armazena a última data de modificação do estudante.
- `_createdBy`: String, Armazena o usuário que criou o estudante. Referência a entidade 'User'.
- `_schoolId`: String, Armazena a escola. Referência a entidade 'School'.

3.2.5.3 *School*

Seguem os campos da entidade *School*.

- `_id`: String. Gerado automaticamente pelo MongoDB na inclusão do registro.
- `name`: String. Armazena nome da escola.
- `zipcode`: Number. Armazena o CEP da Escola.
- `adress`: String. Armazena o endereço.
- `adress2`: String. Armazena o complemento do endereço.
- `district`: String. Armazena o bairro.
- `city`: String. Armazena a cidade.
- `state`: String. Armazena o estado.
- `country`: String. Armazena o país.
- `phone`: String. Armazena o telefone.
- `email`: String. Armazena o email.
- `_dateCreate`: Date. Armazena a data de criação.

- `_dateModifi`: Date. Armazena a data de modificação.
- `_createdBy`: String. Armazena o usuário que criou. Referência a entidade 'User'.

3.2.5.4 File

Seguem os campos da entidade File.

- `_id`: String. Gerado automaticamente pelo MongoDB na inclusão do registro.
- `media`: Binary. Armazena a foto ou vídeo.
- `date`: Date. Armazena a data.
- `comment`: String. Armazena o texto de comentário.
- `_createdBy`: String. Armazena usuário que criou. Referência a entidade 'User'.

3.2.5.5 Document

Seguem os campos da entidade *Document*.

3.2.5.5.1 Campos compartilhados pelos documentos e acompanhamento:

- `_id`: String. Gerado automaticamente pelo MongoDB na inclusão do registro.
- `_createdBy`: String. Armazena usuário que criou. Referência a entidade 'User'
- `_studentId`: String. Armazena estudante. Referência a entidade 'Student'.
- `date`: Date. Armazena data.
- `title`: String. Armazena o título.
- `type`: String. Armazena o tipo para distinguir entre 'PDI', 'Adequação curricular', 'parecer' ou 'acompanhamento'.

3.2.5.5.2 Campo relacionado ao parecer:

- `text`: String. Armazena o texto.

3.2.5.5.3 Campo relacionados a Adequação curricular:

- `grade`: String. Armazena a série do aluno.
- `class`: String. Armazena o turno.
- `period`: String. Armazena o trimestre ou semestre.
- `matters`: String. Armazena a matéria ou área do conhecimento.

- teacher: String. Armazena o professor.
- adaptationHistoric: String. Armazena histórico.
- adaptationNeed: String. Armazena necessidades.
- programGolas: String. Armazena programação de metas.
- suggetionGols: String. Armazena sugestões de metas.
- programConceptual: String. Armazena a programação dos conceitos.
- suggetionConceptual: String. Armazena sugestão dos conceitos.
- programContents: String. Armazena a programação do conteúdo.
- suggestionContents: String. Armazena sugestão do conteúdo.
- programEvaluation: String. Armazena a programação da avaliação.
- suggestionEvaluation: String. Armazena a sugestão de avaliação.

3.2.5.5.4 Campo relacionados ao PDI:

- days: String. Armazena os dias da semana que estudante tem aula.
- extraClassActivity: String. Armazena as atividade extraclasse do estudante.
- otherInformation: String. Armazena mais informações relevantes.
- forwardingDate: String. Armazena a data de encaminhamento ao AEE.
- forwardingReason: String. Armazena a razão de encaminhamento do AEE.
- familyDinamics: String. Armazena as informações da dinâmica familiar.
- historical: String. Armazena as informações do histórico.
- attendance: String. Armazena as informações de atendimentos anteriores.
- character: Array que define as habilidade do estudante.
 - name: String. Armazena o nome da habilidade.
 - value: String. Armazena o texto da habilidade.
- actionExiting: String. Armazena as ações já existentes.
- actionFuture: String. Armazena as ações futuras.

4 TELAS DO SISTEMA E NAVEGAÇÃO

Este capítulo mostra as telas que o usuário tem acesso no sistema e explica as ações que ele pode executar.

4.1 Páginas Iniciais e Login

A tela inicial (Figura 4.1) do sistema exibe o nome do sistema e permite a navegação através no menu superior em tela de computadores ou *tablet*, e um menu ‘sanduíche’ em dispositivos com telas pequenas.

Para o usuário realizar qualquer ação no sistema, o mesmo precisa identificar-se clicando no botão *Entrar* (Figura 4.2). Seja através do login com usuário e senha que foram previamente cadastrados ou através do cadastro de usuários (Figura 4.3).

Após o login, o usuário não visualiza mais a o botão *Entrar* no topo, sendo este substituído pela saudação “Olá” seguida de seu nome para indicar que encontra-se autenticado no sistema.

4.2 Cadastro de Usuário

O cadastro de usuário (Figura 4.3) é utilizado para cadastrar professores a fim destes acessarem ao sistema e poderem realizar as ações no sistema. Após o cadastro o professor terá permissão para cadastrar os seus alunos.

O formato da URL segue o padrão:

/aluno/[id_usuario]/[modo]

4.3 Listagem de alunos

Após o login o usuário é encaminhado para a tela de gerenciamento de alunos (Figura 4.4). A primeira vez que o professor acessar o sistema não haverá estudantes cadastrados. Para realizar tal função o professor deve clicar em ‘Cadastrar Aluno’.

Os estudantes cadastrados aparecem um por linha. As informações das colunas são:

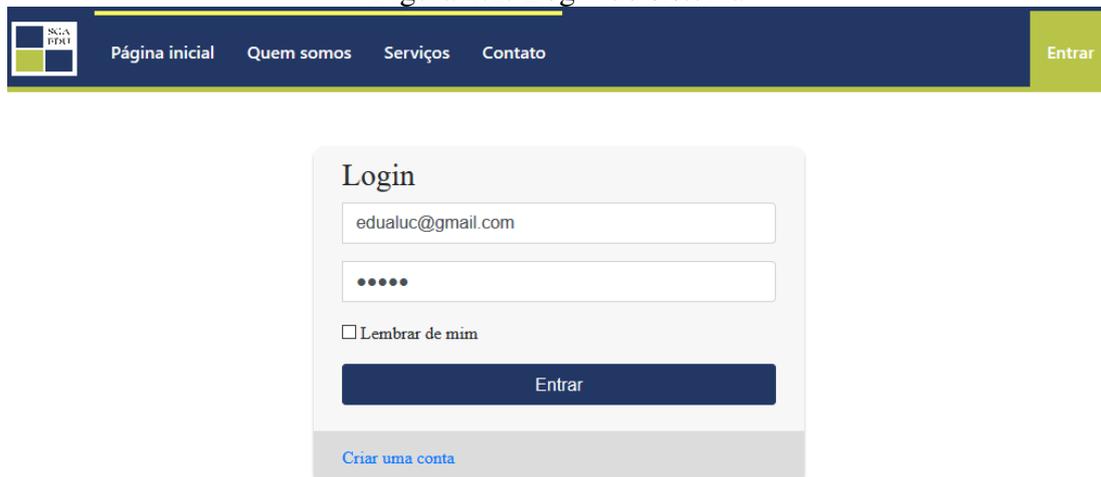
Figura 4.1: Página Inicial



 Sistema de Gestão e Acompanhamento Educacional - 2018

Fonte: O Autor

Figura 4.2: Login do sistema



 Sistema de Gestão e Acompanhamento Educacional - 2018

Fonte: O Autor

Figura 4.3: Cadastro de Usuário do sistema.

 Sistema de Gestão e Acompanhamento Educacional - 2018

Fonte: O Autor

Figura 4.4: Lista de alunos do professor

| Foto | Nome | Escola | Turma | Editar | Excluir |
|-------------------------------------------------------------------------------------|--------------------|----------------|-------|----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
|  | Franciele Ramos | Getulio Vargas | 122 |  Editar |  Excluir |
|  | Ana Josefa Cardoso | Getúlio Vargas | 120 |  Editar |  Excluir |
|  | Estela Micala | Getúlio Vargas | 22 |  Editar |  Excluir |

 Sistema de Gestão e Acompanhamento Educacional - 2018

Fonte: O Autor

- Foto do estudante,
- Nome do Estudante
- Escola.
- Turma.

Há também três ações que podem ser tomadas. Clicando no nome do estudante o professor será encaminhado para o *Menu do Estudante*. Clicando em *editar* e *excluir* o professor será encaminhado para o cadastro de estudante onde poderá revisar as informações cadastradas e posteriormente alterá-las ou excluí-las.

O formato da URL segue o padrão:

`/alunos/`

4.4 Cadastramento de Estudantes.

O *Cadastro de Estudante* (Figuras 4.5, 4.6 e 4.7) recebe dois parâmetro na URL. Sendo eles a identificação do estudante (`id_estudante`) e o modo de acesso (`modo`). Formando a URL no seguinte padrão:

`/aluno/[id_estudante]/[modo]`

Quanto ao parâmetro *modo* há quatro possibilidades de valores, sendo eles: de *inserção*, *consulta*, *edição* e *exclusão*. Cada modo é dado pela ação que o professor quer executar do CRUD. O modo *inserção* abre a tela para cadastro do novo estudante. O modo *consulta* carrega os dados do estudante, mas desabilita a alteração dos mesmos. O modo *edição* carrega os dados e permite a alteração, salvando no banco após confirmação. E o modo *exclusão* permite a consulta dos dados antes da confirmação da exclusão.

Já o *id de estudante* é uma sequência de caracteres que referencia o registro do estudante salvo no banco de dados. Em modo de *inserção* esse valor é ignorado visto que o estudante será cadastrado como novo. Para os outros modos o valor é utilizado para carregar as informações salvas do estudante no banco de dados.

4.5 Listagem e cadastro de Escolas

O cadastro de escola funciona da mesma forma que o cadastro de estudante.

O formato da URL para a listagem de escola (Figura 4.8) segue o padrão:

`/escolas/`

Figura 4.5: Cadastramento de Estudantes - aba 1


Página inicial Alunos Escolas Grupos
Bem-vindo, Eduardo Sair

Cadastro de Aluno

1 Dados do aluno 2 Dados da Escola 3 NEE

Imagem do aluno:



Arraste e solte uma imagem ou clique no botão para selecionar um arquivo

[Selecionar arquivo](#)

Nome:

Nome da Mãe:
 Telefone na Mãe:
 E-mail da Mãe:

Nome do Pai:
 Telefone do Pai:
 E-mail do Pai:

Nome do Responsável:
 Parentesco:
 Telefone do Responsável:
 E-mail do Responsável:

Endereço:
 Complemento:

Bairro:
 Cidade:
 Estado:

[VOLTAR](#) [PRÓXIMO](#)

Figura 4.6: Cadastramento de Estudantes - aba 2

Cadastro de Aluno

1 Dados do aluno — 2 Dados da Escola — 3 NEE

Escola:

Matrícula:

Turma:

Turno:

Ano/Série:

Professor NEE:

Orientação Educacional:

Coordenação Pedagógica:

Data de encaminhamento a AEE/SR: _____

Motivo do encaminhamento a AEE/SR:

VOLTAR

Figura 4.7: Cadastramento de Estudantes - aba 3

Cadastro de Aluno

1 Dados do aluno — 2 Dados da Escola — 3 NEE

CID:

- Altas Habilidades/Superdotado
- Deficiência Visual/Cegueira
- Deficiência Física
- Dislexia
- Deficiência Múltipla
- Síndrome de Ritt
- Transtorno do Espectro Autista
- Baixa Visão
- Deficiência Auditiva/Surdez
- Deficiência Intelectual
- Transtorno de Oposição e desafio(TOD)
- Síndrome de Down
- Surdocegueira
- Síndrome do X frágil
- Transtornos psicóticos agudos e transitórios
- Transtorno de conduta

Outras necessidades especiais:

Cadastrar

VOLTAR

Figura 4.8: Lista de Escolas

SIGA FDU

Página inicial Alunos **Escolas** Grupos Bem-vindo, Eduardo Sair

Página Inicial / Alunos

Minhas Escolas

Procurar escola

Nome da Escola Bairro Cidade Alunos Cadastrados

 Sistema de Gestão e Acompanhamento Educacional - 2018

Fonte: O Autor

Figura 4.9: Cadastro de Escolas

SIGA FDU

Página inicial Alunos Escolas Grupos Bem-vindo, Eduardo Sair

Cadastro de Escola

1 Dados da Escola

Nome:

Telefone:

E-mail:

Endereço:

Número:

Bairro:

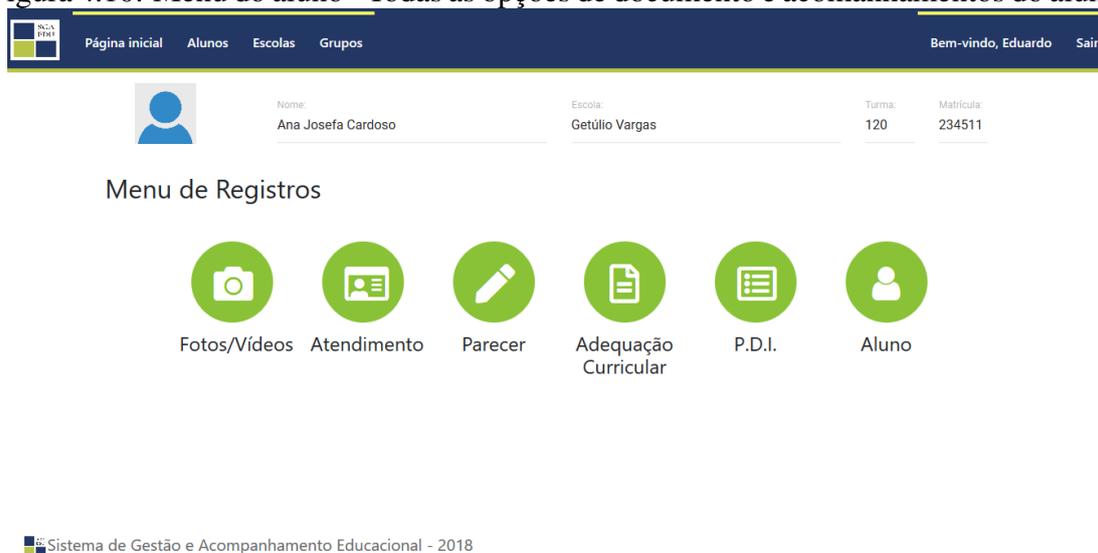
Cidade:

Estado:

 Sistema de Gestão e Acompanhamento Educacional - 2018

Fonte: O Autor

Figura 4.10: Menu do aluno - Todas as opções de documento e acompanhamentos do aluno.



Fonte: O Autor

O formato da URL para o cadastro da escola (Figura 4.9) segue o padrão:

`/escola/[id_escola]/`

As escolas cadastradas aparecem uma por linha. As informações que são mostradas da escola nas colunas são:

- Nome da Escola
- Cidade.
- Bairro.

O professor deve clicar em *Cadastrar Escola* para cadastrar nova escola, assim ele será encaminhado para o *Cadastro de Escola*.

Há também duas ações que podem ser tomadas que são *editar* ou *excluir*. O professor será encaminhado para o cadastro da escola onde ele pode revisar as informações cadastradas e posteriormente alterar ou excluir conforme clicado anteriormente.

4.6 Menu do aluno

O *menu do aluno* (Figura 4.10) é formado pelas opções de inclusão de acompanhamento do estudante (seja foto, vídeos e texto) e de documento (parecer, adequação curricular, PDI). Além de também ter um botão para de edição de estudante.

Desta tela do menu aparece duas regiões, uma acima e outra abaixo das opções principais. A superior é a *capa do estudante* e nela ficam os dados do estudante em que

Figura 4.11: Acompanhamento por foto ou vídeo

The screenshot shows a web interface for uploading photos or videos. At the top, there is a navigation menu with 'Página inicial', 'Alunos', 'Escolas', and 'Grupos'. On the right, it says 'Bem-vindo, Eduardo' and 'Sair'. Below the menu, there is a user profile section with a blue circular icon and the following information:

| Nome: | Escola: | Turma: | Matrícula: |
|--------------------|----------------|--------|------------|
| Ana Josefa Cardoso | Getúlio Vargas | 120 | 234511 |

The main section is titled 'Cadastrar Fotos ou Vídeos'. It features a dashed box containing a cloud icon with an upward arrow and the text: 'Arraste e solte uma imagem/vídeo ou clique no botão para selecionar um arquivo'. Below this is a 'Selecionar arquivo' button. Underneath the dashed box, there is a 'Data:' label followed by a text input field with the placeholder 'dd/mm/aaaa'. Below that is a 'Comentário sobre a foto/vídeo:' label followed by a text input field. At the bottom right, there are two buttons: 'CANCELAR' and 'SALVAR'. In the bottom left corner, there is a small logo and the text 'Sistema de Gestão e Acompanhamento Educacional - 2018'.

Fonte: O Autor

se está trabalhando com as informações de nome, escola, turma, matrícula e a foto do estudante. Ela aparece em todas as telas de *acompanhamento* e *documento*.

A região inferior do menu é a *timeline do estudante* que ficam todos os documentos e acompanhamentos do mesmo. Ela fica na horizontal em ordem cronológica.

O formato da URL segue o padrão:

`/aluno/[id_estudante]/`

4.7 Acompanhamento do aluno.

O acompanhamento do estudante deve ser uma das ações mais usadas do sistema, pois o professor deve cadastrar todas as atividades relevantes ao desenvolvimento dele a cada dia. Tirar fotos, gravar vídeos ou escrever um comentário, para que com o passar do tempo possa ser analisada a evolução do estudante.

4.7.1 Foto/Vídeo

Para o acompanhamento de foto e vídeo (Figura 4.11) basta o professor realizar o *upload* clicando em *Selecionar Arquivo* ou arrastando para a área determinada (Cadastrar

Figura 4.12: Acompanhamento de texto somente

Sistema de Gestão e Acompanhamento Educacional - 2018

Fonte: O Autor

Fotos ou Vídeos). Eles devem ter sido gravadas anteriormente e salvos no dispositivo que está acessando o aplicativo, seja computador ou celular, já que o design é responsivo.

O formato da URL para o acompanhamento com foto ou vídeo segue o padrão:

`/aluno/[id_estudante]/arquivos`

4.7.2 Texto

Para o acompanhamento de texto (Figura 4.12), a data será preenchida automaticamente com a data do dia do cadastro.

O formato da URL para o acompanhamento de texto segue o padrão:

`/aluno/[id_estudante]/acompanhamento`

4.8 Parecer

Os pareceres (Figura 4.13) são documento usados conforme a necessidade. Eles são em um formato mais aberto para serem mais versáteis.

Figura 4.13: Cadastro de Parecer

Fonte: O Autor

O formato da URL segue o padrão:

`/aluno/[id_estudante]/parecer`

Eles são composto por:

- Título;
- Data;
- Tipo: Avaliação inicial, Avaliação processual, Avaliação Final, Encaminhamento para pais ou responsáveis ou Encaminhamento para outros profissionais.
- Texto longo para escrever o parecer.

4.9 PDI

Para o fazer um PDI o professor deve analisar o histórico do aluno e a situação atual de desenvolvimento do estudante, ponderando as ações que estão sendo tomadas com o mesmo.

O formato da URL segue o padrão:

`/aluno/[id_estudante]/pdi`

O PDI foi dividido em algumas abas para o cadastro. A primeira *Identificação e Histórico*: refere-se aos dados gerais de identificação e histórico do estudante. Sendo eles:

Figura 4.14: Cadastro do PDI - Dados de identificação e histórico do estudante


[Página inicial](#)
[Alunos](#)
[Escolas](#)
[Grupos](#)
Bem-vindo, Eduardo [Sair](#)



Nome: **Ana Josefa Cardoso**

Escola: **Getúlio Vargas**

Turma: **120**

Matrícula: **234511**

P.D.I.: Plano de Desenvolvimento Individualizado

1 Geral
2 Habilidades
3 Proposições

Data
14/12/2018

Atividades Extra-Classe:

Atividades Extra-Classe

Outras informação:

Outras informação

Dias de atendimento:

Turno:

AEE:

AEE

Histórico

Código no Encaminhamento AEE:

Código no Encaminhamento AEE

Motivo do encaminhamento:

Motivo do encaminhamento

Histórico Escolar:

Histórico Escolar

Data do encaminhamento:

Data do encaminhamento

Dinâmica Familiar:

Dinâmica Familiar

Atendimento (especializado)*:

Atendimento (especializado)*

VOLTAR
PRÓXIMO

Figura 4.15: Cadastro do PDI - situação das habilidades


Página inicial Alunos Escolas Grupos
Bem-vindo, Eduardo Sair



Nome: Ana Josefa Cardoso

Escola: Getúlio Vargas

Turma: 120

Matrícula: 234511

P.D.I.: Plano de Desenvolvimento Individualizado

1 Geral
2 Habilidades
3 Proposições

| | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Memória (Curto/Longo Prazo):</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Memória (Curto/Longo Prazo)</div> <p>Raciocínio Lógico:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Raciocínio Lógico</div> <p>Linguagem:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Linguagem</div> <p>Atenção:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Atenção</div> <p>Percepção:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Percepção</div> <p>Funções Executivas:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Funções Executivas</div> | <p>Relações Interpessoais:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Relações Interpessoais</div> <p>Relações Comportamentais:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Relações Comportamentais</div> <p>Autonomia:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Autonomia</div> <p>Relações Culturais:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Relações Culturais</div> <p>Auto-imagem:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Auto-imagem</div> <p>Resolução de conflitos:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Resolução de conflitos</div> | <p>Dinâmica Geral - Controle das ações motoras:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Dinâmica Geral - Controle das ações motoras</div> <p>Execução e intencionalidade:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Execução e intencionalidade</div> <p>Flexibilidade, tonicidade, movimentos involuntários:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Flexibilidade, tonicidade, movimentos involuntários</div> <p>Lateralidade e domínio de esquema corporal:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Lateralidade e domínio de esquema corporal</div> <p>Equilíbrio estático e dinâmico:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Equilíbrio estático e dinâmico</div> <p>Compensações motoras:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Compensações motoras</div> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

VOLTAR
PRÓXIMO

Figura 4.16: Cadastro do PDI - Proposições

The screenshot displays a web interface for PDI registration. At the top, a dark blue navigation bar contains the logo on the left, menu items 'Página inicial', 'Alunos', 'Escolas', and 'Grupos' in the center, and 'Bem-vindo, Eduardo' and 'Sair' on the right. Below the navigation bar, a user profile section shows a blue circular avatar icon, the name 'Ana Josefa Cardoso', the school 'Getúlio Vargas', the class '120', and the enrollment number '234511'. The main heading is 'P.D.I.: Plano de Desenvolvimento Individualizado'. A progress indicator below the heading shows three steps: '1 Geral', '2 Habilidades', and '3 Proposições', with the third step being active. The title 'Plano de Intervenção Pedagógica' is followed by two text input fields: 'Ações já existentes:' and 'Ações a serem desenvolvidas:'. A 'Cadastrar' button is positioned below the second field. At the bottom, there are 'VOLTAR' and 'PRÓXIMO' buttons.

Página inicial Alunos Escolas Grupos Bem-vindo, Eduardo Sair

Nome: Ana Josefa Cardoso Escola: Getúlio Vargas Turma: 120 Matrícula: 234511

P.D.I.: Plano de Desenvolvimento Individualizado

1 Geral 2 Habilidades 3 Proposições

Plano de Intervenção Pedagógica

Ações já existentes:

Ações já existentes

Ações a serem desenvolvidas:

Ações a serem desenvolvidas

Cadastrar

VOLTAR PRÓXIMO

- Data.
- Dias de atendimento na sala de recursos.
- Turno de atendimento,
- Atividades extra-classe que realiza.
- Dias da atividade extraclasse.
- Outras informações.
- Data de encaminhamento ao AEE.
- Motivo do encaminhamento.
- Dinâmica Família.
- Histórico escolar.
- Atendimento AEE.
- NEES.

A próxima aba é referente a situação de desenvolvimento do estudante. Ela é dividida em três colunas cada uma com seis campos de texto grande com habilidades que o estudante adquiriu ou que está desenvolvendo.

- Coluna 1 - Funções Cognitivas. São elas:
 - Memória (Curto/Longo Prazo);
 - Raciocínio Lógico;
 - Linguagem;
 - Atenção;
 - Percepção;
 - Funções Executivas.
- Coluna 2 - Funções Interativas Sociais. São elas:
 - Relações Interpessoais;
 - Relações Comportamentais;
 - Autonomia;
 - Relações Culturais;
 - Auto-imagem, auto-estima;
 - Resolução de conflitos.
- Coluna 3 - Funções Motoras. São elas:

Figura 4.17: Adequação curricular - Aba 1 - Dados da matéria ou área do conhecimento

Página inicial Alunos Escolas Grupos Bem-vindo, Eduardo Sair

Nome: Ana Josefa Cardoso Escola: Getúlio Vargas Turma: 120 Matrícula: 234511

Adequação Curricular

1 Dados gerais — 2 Matriz Histórica — 3 Matriz Necessidade — 4 Matriz Programação — 5 Matriz Sugestões

Data: Série/Etapa: Turma: Trimestre:

Matéria/Área: Professor:

VOLTAR

Fonte: O Autor

- Dinâmica Geral - Controle das ações motoras;
- Execução e intencionalidade;
- Flexibilidade, tonicidade, movimentos involuntários;
- Lateralidade e domínio de esquema corporal;
- Equilíbrio estático e dinâmico;
- Compensações motoras.

E a última aba existente *Proposições* há dois campos de texto grandes:

- Ações já existentes.
- Ações a serem desenvolvidas.

Figura 4.18: Adequação curricular - Aba 2 - Matriz Histórica

The screenshot displays a web application interface for curriculum adjustment. At the top, there is a dark blue navigation bar with the logo of SCSA EPDU on the left and the text 'Página inicial', 'Alunos', 'Escolas', and 'Grupos' in the center. On the right side of the bar, it says 'Bem-vindo, Eduardo' and 'Sair'. Below the navigation bar, there is a user profile section with a blue circular icon on the left. To the right of the icon, the following information is displayed: 'Nome: Ana Josefa Cardoso', 'Escola: Getúlio Vargas', 'Turma: 120', and 'Matrícula: 234511'. Below the profile information, the main heading is 'Adequação Curricular'. Underneath this heading is a horizontal progress bar with five steps: '1 Dados gerais', '2 Matriz Histórica' (which is highlighted with a blue circle), '3 Matriz Necessidade', '4 Matriz Programação', and '5 Matriz Sugestões'. Below the progress bar, the section is titled 'Histórico'. Under 'Histórico', there is a text input field with the placeholder text 'Tras o texto do Histórico Anterior aqui...'. At the bottom of the page, there are two buttons: 'VOLTAR' and 'PRÓXIMO'.

Figura 4.19: Adequação curricular - Aba 3 - Matriz Necessidade

The screenshot shows a web application interface for 'Adequação Curricular'. At the top, there is a dark blue navigation bar with the SCSA EPEU logo on the left and menu items: 'Página inicial', 'Alunos', 'Escolas', and 'Grupos'. On the right side of the bar, it says 'Bem-vindo, Eduardo' and 'Sair'. Below the navigation bar, there is a user profile section with a blue circular icon and the following information: Nome: Ana Josefa Cardoso, Escola: Getúlio Vargas, Turma: 120, and Matrícula: 234511. The main heading is 'Adequação Curricular', followed by a progress indicator with five steps: 1 Dados gerais, 2 Matriz Histórica, 3 Matriz Necessidade (highlighted in blue), 4 Matriz Programação, and 5 Matriz Sugestões. Below this, the section 'Necessidade' contains a text input field with the placeholder text 'Tras o texto do Necessidades aqui...'. At the bottom of the page, there are two buttons: 'VOLTAR' and 'PRÓXIMO'.

Figura 4.20: Adequação curricular - Aba 4 - Matriz Programação Curricular

Objetivo

Conteúdos Conceituais

Conteúdos Procedimentais

Avaliações

VOLTAR PRÓXIMO

Figura 4.21: Adequação curricular - Aba 4 - Matriz Sugestões para Adequação Curricular

Sistema de Gestão e Acompanhamento Educacional

Página inicial Alunos Escolas Grupos Bem-vindo, Eduardo Sair

Nome: Ana Josefa Cardoso Escola: Getúlio Vargas Turma: 120 Matrícula: 234511

Adequação Curricular

1 Dados gerais 2 Matriz Histórica 3 Matriz Necessidade 4 Matriz Programação 5 Matriz Sugestões

Sugestão

Objetivos:

Objetivos

Conteúdos Conceituais:

Conteúdos/Conceitos

Conteúdos Procedimentais:

Conteúdos Procedimentais

Avaliações:

Avaliações

Cadastrar

VOLTAR PRÓXIMO

4.10 Adequação curricular

A programação curricular normal que cada professor de sala regular faz para sua turma normalmente não deve ser aplicada para estudantes do AEE (atendimento educacional especializado), pois eles necessitam de uma programação curricular mais personalizada e específica. A adequação curricular é o documento com a proposta de adaptação do conteúdo, metodologia e avaliação para aquele estudante específico.

O formato da URL segue o padrão:

/aluno/[id_estudante]/adequacao

O sistema usa uma adequação curricular baseada em quatro eixos chamada de *Matrizes*, conforme (SANTOS Jr, 2018) que são: *Histórico*, *Necessidade*, *Programação* e *Sugestão*.

O professor da sala de recurso deve fazer uma adequação para cada matéria/área do conhecimento.

Os campos do cadastro são:

1. Aba dados da matéria ou área do conhecimento que são: série, turma, trimestre matéria/área e o professor desta matéria/área.
2. Aba *Matriz Histórica*: um campo de texto grande para descrever o histórico do aluno, trazendo como texto inicial o histórico cadastrado no *Cadastro de Estudante*.
3. Aba *Matriz Necessidade*: um campo de texto grande para descrever a necessidade trazendo como texto inicial a necessidade cadastrada no *Cadastro de Estudante*.
4. Aba *Programação Curricular*: quatro campos de textos grandes: *Objetivo*, *Conteúdos Conceituais*, *Procedimentos* e *Avaliações*.
5. Aba *Sugestões para Adequação Curricular*: quatro campos de textos *Objetivo*, *Conteúdos Conceituais*, *Procedimentos* e *Avaliações*.

5 CONCLUSÃO

O sistema que foi a motivação para o desenvolvimento deste trabalho tem por proposta resolver os problemas mais frequentes relacionados aos professores da sala de recurso, mas além disso é apoio prático para tese de doutorado que está sendo desenvolvida pelo professor coorientador Prof. Ms. Francisco Santos Jr.. Sendo assim o sistema precisa auxiliar o desenvolvimento do estudante de forma longitudinal e permitir observar as evoluções do estudante durante o período.

Sendo o objetivo do trabalho alcançado com desenvolvimento do sistema para acompanhar as atividades desenvolvidas, as interações sociais ou ambientais realizadas, e permitir a análise da evolução no aprendizado do estudante, gerando registros de qualquer fato que possa ser útil posteriormente. Como forma de avaliação do sistema desenvolvido foram realizadas reuniões com professores da sala de recursos do Colégio Marista Rosário. Estes demonstraram-se animados com a sistema e dispostos a utilização futura. Eles também foram apresentados ao sistema antigo de maneira a comparar as duas versões quanto ao design e facilidade de uso, e mostraram-se favoráveis a nova versão desenvolvida.

De forma a adequar a metodologia de trabalho da instituição surgiu a demanda de aprimoramento do sistema em alguns pontos como no PDI, as habilidades utilizadas pela instituição são diferentes, assim como a divisão em três grupo (que no sistema corresponde são as colunas *Funções cognitivas*, *Funções Interativas Sociais*, *Funções Motoras*) não necessariamente correspondem a metodologia utilizada pela instituição. Além disto, tornar-se necessário a adaptação do sistema a fim de torná-lo útil para a instituição que venha a usá-lo, visto que a proposta deste trabalho era o desenvolvimento de um protótipo funcional não focado em uma instituição específica.

Também é importante para o futuro uma preocupação maior com segurança das informações e do sistema, principalmente da camada da API que implementa a interface com o banco dados, pois o armazenamento das informações pessoais dos estudante e professores devem ter sua confiabilidade mantidas. Também não foram desenvolvidos acessos a outros tipos de usuário, além do professor da sala de recurso, como outros professores, coordenadores, outros profissionais da área da saúde e pais. Esses teriam acesso limitado as informações do sistema, como pareceres ou cadastro do estudante.

REFERÊNCIAS

BABEL. **What is Babel? - Babel**. 2018. <<https://babeljs.io/docs/en/>>. Accessed in: 22-11-2018.

ECMA INTERNATIONAL. **ECMAScript Language Specification - ECMA-262 Edition 5.1**. 2011. <<https://www.ecma-international.org/ecma-262/5.1/index.html>>. Accessed in: 15-12-2018.

ECMA INTERNATIONAL. **ECMAScript® 2015 Language Specification**. 2015. <<https://www.ecma-international.org/ecma-262/6.0/index.html>>. Accessed in: 15-12-2018.

FACEBOOK INC. **Flux | Application Architecture for Building User Interfaces**. 2018. <<https://facebook.github.io/flux/docs/in-depth-overview.html>>. Accessed in: 22-11-2018.

FACEBOOK INC. **JSX In Depth - React**. 2018. <<https://reactjs.org/docs/jsx-in-depth.html>>. Accessed in: 22-11-2018.

FERREIRA, G. M. **SIR-EDU : sistema integrado de recursos educacionais para a gestão do acompanhamento de alunos com necessidades especiais**. 71 f. Monografia (Graduação) — Instituto de Informática, UFRGS - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

GITHUB, INC. **react2**. 2018. <<https://github.com/facebook/react>>. Accessed in: 22-11-2018.

JSON. 2018. <<https://www.json.org/>>. Accessed in: 19-11-2018.

MONGODB INC. **Data Modeling Introduction — MongoDB Manual**. 2018. <<https://docs.mongodb.com/manual/core/data-modeling-introduction/>>. Accessed in: 22-11-2018.

MONGODB INC. **Databases and Collections — MongoDB Manual**. 2018. <<https://docs.mongodb.com/manual/core/databases-and-collections/>>. Accessed in: 22-11-2018.

MONGODB INC. **Documents — MongoDB Manual**. 2018. <<https://docs.mongodb.com/manual/core/document/>>. Accessed in: 22-11-2018.

MONGODB INC. **Introduction to MongoDB — MongoDB Manual**. 2018. <<https://docs.mongodb.com/manual/introduction/>>. Accessed in: 22-11-2018.

MOZILLA. **HTML5 | MDN**. 2018. <<https://developer.mozilla.org/pt-BR/docs/Web/HTML/HTML5>>. Accessed in: 10-11-2018.

NODE.JS FOUNDATION. **Express framework de aplicativo da web Nodejs**. 2018. <<https://expressjs.com/pt-br/>>. Accessed in: 14-12-2018.

NODE.JS FOUNDATION. **Node.js**. 2018. <<https://nodejs.org/pt-br/>>. Accessed in: 14-12-2018.

REFSNES DATA. **ECMAScript 6**. 2018. <https://www.w3schools.com/js/js_es6.asp>. Accessed in: 16-11-2018.

REFSNES DATA. **HTML5 Browser Support**. 2018. <https://www.w3schools.com/html/html5_browsers.asp>. Accessed in: 22-11-2018.

SANTOS Jr, F. D. dos. Adequações curriculares em quatro matrizes. In: _____. **Reflexões sobre o Currículo Inclusivo**. Porto Alegre, Brasil: [s.n.], 2018.

SANTOS Jr, F. D. dos; LUCAS, E. A.; WIVES, L. K. Uma proposta metodológica para apoiar e acompanhar alunos com necessidades educacionais especiais e realizar a prospecção e o desenvolvimento longitudinal de sua aprendizagem. In: **XXVI Ciclo de Palestras sobre novas tecnologias na educação**. Porto Alegre, Brasil: [s.n.], 2018.

SASS. **SASS_REFERENCE — Documentation by YARD 0.9.12**. 2018. <https://sass-lang.com/documentation/file.SASS_REFERENCE.html>. Accessed in: 22-11-2018.

W3C. **Cascading Style Sheets**. 2018. <<https://www.w3.org/Style/CSS/#specs>>. Accessed in: 10-11-2018.

WIKIMEDIA FOUNDATION, INC. **Single-page application - Wikipedia**. 2018. <https://en.wikipedia.org/wiki/Single-page_application>. Accessed in: 22-11-2018.