



**UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DO TOCANTINS/CAMETÁ
FACULDADE DE SISTEMAS DE INFORMAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

CARLA THAIS GONÇALVES CALDAS

**Projeto e Desenvolvimento de Um Sistema de Gestão Escolar: Da Matrícula
de Alunos até a Gestão Escolar Municipal**

**Cametá
2026**



**UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DO TOCANTINS/CAMETÁ
FACULDADE DE SISTEMAS DE INFORMAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

CARLA THAIS GONÇALVES CALDAS

**Projeto e Desenvolvimento de Um Sistema de Gestão Escolar: Da Matrícula
de Alunos até a Gestão Escolar Municipal**

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Sistemas de Informa-
ção, Faculdade de Sistemas de Informação, Cam-
pus Universitário do Tocantins/Cametá, Univer-
sidade Federal do Pará, como requisito parcial à
obtenção do título de Bacharel em Sistemas de
Informação.

Orientador: Prof. Dr. Fabricio de Souza Farias.

**Cametá
2026**

Caldas, Carla Thais Gonçalves

Projeto e Desenvolvimento de Um Sistema de Gestão Escolar: Da Matrícula de Alunos até a Gestão Escolar Municipal/ CARLA THAIS GONÇALVES CALDAS. – Cametá, 2026.

87 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Fabricio de Souza Farias.

Monografia – UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DO TOCANTINS/CAMETÁ
BACHARELADO EM SISTEMAS DE INFORMAÇÃO, 2026.

1. Gestão escolar municipal. 2. Sistema web. 3. Informatização. I. Título.

CARLA THAIS GONÇALVES CALDAS

**Projeto e Desenvolvimento de Um Sistema de Gestão Escolar:
Da Matrícula de Alunos até a Gestão Escolar Municipal**

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Sistemas de Informa-
ção, Faculdade de Sistemas de Informação, Cam-
pus Universitário do Tocantins/Cametá, Univer-
sidade Federal do Pará, como requisito parcial à
obtenção do título de Bacharel em Sistemas de
Informação.

Data da Defesa: 16 de Julho de 2026
Conceito: Excelente

Banca Examinadora

Prof. Dr. Fabricio de Souza Farias.

Faculdade de Sistemas de Informação - UFPA
Orientador

Prof. Dr. Allan Barbosa Costa

Faculdade de Sistemas de Informação - UFPA
Membro da Banca

Prof. Me. Leonardo Nunes Gonçalves

Faculdade de Sistemas de Informação - UFPA
Membro da Banca

Prof. Keventon Rian Guimarães Gonçalves

Faculdade de Sistemas de Informação - UFPA
Membro da Banca

Cametá
2026

Dedico esta conquista à minha família, em especial às mulheres extraordinárias que a compõem, exemplos de força, resiliência e coragem. Obrigada por me ensinarem a persistir firmemente, dia após dia, em direção aos meus objetivos.

AGRADECIMENTOS

Agradeço primeiramente a Deus por ser o meu sustento em vários momentos da minha vida, sempre me abençoando, protegendo e guiando para os caminhos e decisões certas.

Aos meus pais, Rosangela Pereira Gonçalves e Marinaldo Lima Caldas, e aos meus irmãos, Wendell Gonçalves Caldas e Paulo Victor Gonçalves Mória, agradeço por serem o meu alicerce durante toda a minha jornada acadêmica, sempre me apoiando e me dando condições para que eu finalizasse o curso. Obrigada pelo amor incondicional, pela compreensão nos momentos de ausência e por partilharem de cada uma das minhas preocupações e sonhos. Esta conquista é tão minha quanto de vocês.

Ao meu amor, Roberto Rodrigues, agradeço profundamente pela paciência, pelo carinho e por todo o suporte emocional durante esta jornada. Sua presença foi fundamental para manter meu equilíbrio e minha motivação.

Ao meu orientador, Prof. Dr. Fabrício Farias, meu muito obrigada por ter orientado este trabalho com tanta dedicação e comprometimento. Sua paciência, incentivo constante e ensinamentos tanto na elaboração deste trabalho e nas aulas ministradas ao longo do curso, quanto na oportunidade de atuar como voluntária em seus projetos de extensão, foram fundamentais para o meu crescimento profissional e acadêmico.

Aos meus colegas do Laboratório de Programação Extrema (LABEX), deixo meus sinceros agradecimentos pelo companheirismo, pelas valiosas trocas de experiências e conhecimentos, e pelo suporte constante durante a minha rotina de estudos.

A todos os professores da Faculdade de Sistemas de Informação por todos os ensinamentos, conselhos e oportunidades compartilhados ao longo destes anos.

A todos os colegas e amigos da turma de Sistemas de Informação 2020 pela parceria e pelo apoio no decorrer do curso, em especial às minhas amigas Vitória Peres, Anna Maria Moraes e Mayara Martins, por todo o apoio, companhia e incentivo no decorrer dessa caminhada que não foi fácil.

Meu muito obrigada a todos!

“A tecnologia não é nada. O importante é que você tenha fé nas pessoas, que elas sejam basicamente boas e inteligentes, e se você lhes der ferramentas, elas farão coisas maravilhosas com elas”
(Steve Jobs)

RESUMO

O avanço das Tecnologias da Informação e Comunicação tem impulsionado a modernização dos processos administrativos em diferentes setores, incluindo a educação. Entretanto, muitas redes municipais de ensino ainda enfrentam dificuldades relacionadas à informatização da gestão escolar, mantendo processos manuais que comprometem a eficiência administrativa, a segurança das informações e a tomada de decisões. Diante desse cenário, propõe-se desenvolver um Sistema de Gestão Escolar Municipal em ambiente web capaz de integrar processos administrativos e acadêmicos, abrangendo desde a matrícula de alunos até a gestão escolar municipal. O desenvolvimento do sistema foi conduzido com pesquisas sobre gestão escolar, levantamento de requisitos, definição das regras de negócio, modelagem do banco de dados e elaboração da arquitetura do sistema. Como resultado, foi desenvolvido um sistema web com diferentes níveis de acesso para administrador, diretor, secretário, professor e aluno, permitindo o gerenciamento de matrículas, turmas, disciplinas, frequência, notas, histórico escolar, calendário acadêmico e relatórios gerenciais. Contudo, este trabalho apresenta um software de Gestão Escolar que fornece um ambiente de integração completa e essa solução contribui para a centralização das informações, redução de processos manuais, melhoria da organização administrativa e apoio à tomada de decisões na gestão educacional municipal.

Palavras-chave: Gestão escolar municipal, sistema web, informatização.

ABSTRACT

The advancement of Information and Communication Technologies has driven the modernization of administrative processes in different sectors, including education. However, many municipal school networks still face difficulties related to the computerization of school management, maintaining manual processes that compromise administrative efficiency, information security, and decision-making. Given this scenario, it is proposed to develop a web-based Municipal School Management System capable of integrating administrative and academic processes, spanning from student enrollment to municipal school management. The development of the system was conducted through research on school management, requirements gathering, business rules definition, database modeling, and system architecture design. As a result, a web system was developed with different access levels for administrator, principal, secretary, teacher, and student, enabling the management of enrollments, classes, subjects, attendance, grades, academic records, academic calendar, and management reports. Ultimately, this work presents a School Management software that provides a complete integration environment, and this solution contributes to the centralization of information, reduction of manual processes, improvement of administrative organization, and support for decision-making in municipal educational management.

Keywords: Municipal school management, web system, computerization.

LISTA DE ILUSTRAÇÕES

Figura 1 – Etapas da metodologia do desenvolvimento da solução proposta	19
Figura 2 – Página Inicial do Sistema Escola	38
Figura 3 – Catálogo de Unidades Escolares	39
Figura 4 – Formulário de Registro de Novo Usuário	39
Figura 5 – Tela de Login do Sistema	40
Figura 6 – Central de Seleção de Portais de Perfil	40
Figura 7 – Visão Inicial do Painel Administrativo	41
Figura 8 – Interface de Cadastro de Escola e Vínculo de Liderança	42
Figura 9 – Formulário de Registro de Usuário pelo ADM	42
Figura 10 – Tela de Consulta Geral de Alunos e Servidores	43
Figura 11 – Formulário de Vínculo de Servidores	43
Figura 12 – Configuração de Prazos e Calendário Letivo	44
Figura 13 – Painel de Aprovação de Matrículas	45
Figura 14 – Emissão de Relatórios de Servidores	45
Figura 15 – Novo Registro de Usuário	46
Figura 16 – Calendário de Aulas Montado	47
Figura 17 – Adicionar Novo Horário ao Calendário	47
Figura 18 – Painel de Planejamento de Vagas	48
Figura 19 – Gerenciamento de Solicitações de Vaga	48
Figura 20 – Mensagem de matrícula aceita	49
Figura 21 – Etapa 1: Dados do Aluno	49
Figura 22 – Etapa 2: Documentação Civil	49
Figura 23 – Etapa 3: Responsáveis	50
Figura 24 – Etapa 4: Localização e Vínculo	50
Figura 25 – Etapa 5: <i>Upload</i> de Documentos	50
Figura 26 – Painel de Enturmação Pendente	51
Figura 27 – Seleção de Turma	51
Figura 28 – Vínculo de Servidor à Unidade Escolar	52
Figura 29 – Cadastro e Parâmetros de Turma	52
Figura 30 – Tela de Gestão de Turmas	53
Figura 31 – Detalhamento da Lotação da Turma	53
Figura 32 – Listagem de Alunos Matriculados por Turma	54
Figura 33 – Modal de Situação Acadêmica	54
Figura 34 – Gerenciamento de Vínculos da Turma	55
Figura 35 – Controle de Frequência de Professores	55
Figura 36 – Tela de Transferências Pendentes	56
Figura 37 – Relatório Consolidado de Servidores	56

Figura 38 – Tela Minhas Turmas	57
Figura 39 – Tela da Funcionalidade Lançar Notas	58
Figura 40 – Tela de Registro de Frequência	58
Figura 41 – Calendário Escolar e Prazos de Lançamento	59
Figura 42 – Portal do Aluno	60
Figura 43 – Fluxo de Solicitação de Nova Matrícula	60
Figura 44 – Tela de Solicitação de Transferência	61
Figura 45 – Tela do Boletim Escolar Digital	61
Figura 46 – Histórico Escolar Gerado pelo Sistema	62
Figura 47 – Representação estrutural de banco de dados.	69

LISTA DE TABELAS

Tabela 1 – Lista de trabalhos e portais visitados e analisados	20
Tabela 2 – Requisitos Funcionais	67
Tabela 3 – Requisitos Não Funcionais	68

LISTA DE ABREVIATURAS E SIGLAS

ADM	Administrador
AJAX	<i>Asynchronous JavaScript and XML</i>
API	<i>Application Programming Interface</i>
CRUD	<i>Create, Read, Update e Delete</i>
CSS	<i>Cascading Style Sheets</i>
DOM	<i>Document Object Model</i>
ENAP	Escola Nacional de Administração Pública
HTML	<i>HyperText Markup Language</i>
IDEB	Índice de Desenvolvimento da Educação Básica
MVC	<i>Model-View-Controller</i>
MySQL	<i>My Structured Query Language</i>
PHP	<i>Hypertext Preprocessor</i>
SGBDR	Sistema Gerenciador de Banco de Dados Relacional
SQL	<i>Structured Query Language</i>
TICs	Tecnologias da Informação e Comunicação
VSCode	Visual Studio Code

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Contexto	16
1.2	Justificativa	16
1.3	Objetivos	18
1.3.1	Objetivo Geral	18
1.3.2	Objetivos Específicos	18
1.4	Metodologia	19
1.4.1	Análise de Cenário e Definição de Requisitos.	19
1.4.2	Modelagem do Sistema e Banco de Dados	21
1.4.3	Desenvolvimento do <i>Software</i>	22
1.5	Estrutura do Trabalho	23
2	REFERENCIAL TEÓRICO	25
2.1	Tecnologias da Informação na Educação	25
2.2	Sistemas de Gestão Escolar e Informatização	25
2.3	Engenharia de <i>Software</i> Aplicada	26
2.3.1	Modelo Cascata (<i>Waterfall</i>)	27
2.3.2	Arquitetura <i>Model View Controller</i>	27
2.4	Tecnologias <i>Back-end</i>	28
2.4.1	PHP	29
2.4.2	<i>Composer</i>	30
2.4.3	Bancos de Dados Relacionais e <i>MySQL</i>	30
2.5	Tecnologias <i>Front-end</i>	31
2.5.1	<i>HTML5</i>	32
2.5.2	<i>CSS3</i>	32
2.5.3	<i>JavaScript</i>	33
2.5.4	<i>Bootstrap</i>	34
2.6	Ferramenta de Desenvolvimento	34
2.6.1	<i>Visual Studio Code</i>	35
2.6.2	<i>Git</i>	36
2.6.3	<i>GitHub</i>	36
3	RESULTADOS	38
3.1	Portais de Acesso por Perfil	40
3.2	Área do Administrador	41
3.2.1	Visão Geral e Navegação	41
3.2.2	Gestão de Unidades Escolares e Liderança	41
3.2.3	Cadastro e Gestão de Usuários	42
3.2.4	Vínculo Profissional de Servidores	43

3.2.5	Gestão de Calendário Escolar Municipal	44
3.2.6	Gestão de Matrículas e Relatórios	44
3.3	Área do Diretor e Secretário	45
3.3.1	Gestão de Usuários	46
3.3.2	Montagem e Gerenciamento do Calendário de Aulas	46
3.3.3	Planejamento de Vagas e Solicitações de Matrícula	47
3.3.4	Vinculação de Aluno a Turma	51
3.3.5	Gestão Administrativa: Servidores e Turmas	52
3.3.6	Lotação e Monitoramento Acadêmico	53
3.3.7	Recursos Humanos e Fluxo de Transferência	55
3.4	Área do Professor	56
3.4.1	Painel de Gestão de Turmas	57
3.4.2	Lançamento de Notas e Médias	57
3.4.3	Controle de Frequência Diária	58
3.4.4	Calendário Escolar e Prazos Administrativos	58
3.5	Área do Aluno	59
3.5.1	Painel de Controle e Acompanhamento Diário	59
3.5.2	Gestão de Matrículas e Séries	60
3.5.3	Processo de Transferência Entre Unidades	60
3.5.4	Visualização de Boletim	61
3.5.5	Emissão de Documentos Oficiais de Histórico	62
4	CONSIDERAÇÕES FINAIS	63
	REFERÊNCIAS	64
	 APÊNDICES	 66
	APÊNDICE A – DOCUMENTAÇÃO DE REQUISITOS	67
A.1	Requisitos Funcionais	67
A.2	Requisitos Não Funcionais	67
	APÊNDICE B – PROJETO DE BANCO DE DADOS	69
B.1	Modelo Lógico	69
B.2	Tabelas de Banco de Dados	70
B.2.1	Tabela usuario	70
B.2.2	Tabela usuario_has_tipo	71
B.2.3	Tabela tipo	71
B.2.4	Tabela matricula	71
B.2.5	Tabela escola	74
B.2.6	Tabela servidor_vinculo	75
B.2.7	Tabela turma	76
B.2.8	Tabela matricula_has_turma	77

B.2.9	Tabela notas_alunos	77
B.2.10	Tabela frequencia_alunos	78
B.2.11	Tabela transferencia	79
B.2.12	Tabela disciplinas	79
B.2.13	Tabela disciplinas_has_turma	80
B.2.14	Tabela disciplinas_has_matricula	80
B.2.15	Tabela historico	81
B.2.16	Tabela calendario_gestao	81
B.2.17	Tabela calendario_aula	83
B.2.18	Tabela horario_aula	83
B.2.19	Tabela cargo	84
B.2.20	Tabela falta_professor	84
B.2.21	Tabela responsavel	85
B.2.22	Tabela responsavel_has_usuario	85
B.2.23	Tabela nivel	86
B.2.24	Tabela <i>password_reset_tokens</i>	86
B.2.25	Tabela estado	86
B.2.26	Tabela municipios	87

1 INTRODUÇÃO

1.1 Contexto

O avanço das Tecnologias de Informação e Comunicação (TICs) tem transformado significativamente a gestão pública e educacional. A crescente adoção de soluções digitais tem permitido a modernização de processos administrativos, promovendo maior eficiência, transparência e qualidade na prestação de serviços públicos (MORAN, 2015). Nesse cenário, os sistemas de gerenciamento escolar tornaram-se ferramentas essenciais para a administração educacional, possibilitando a substituição de procedimentos manuais por processos informatizados e integrados (SOUZA et al., 2023).

Conforme destaca a Escola Nacional de Administração Pública (ENAP), a inovação e a transformação digital no setor público contribuem para o aprimoramento da gestão, otimização de recursos e melhoria dos serviços oferecidos à sociedade (ENAP, 2022). Na mesma direção, a UNESCO (2025) ressalta que o uso de tecnologias na gestão educacional fortalece os processos de tomada de decisão, favorece o monitoramento das informações acadêmicas e amplia a eficiência administrativa das instituições de ensino. Entretanto, essa realidade de informatização ainda enfrenta desafios em diversos municípios brasileiros, especialmente naqueles de pequeno e médio porte.

Em muitas localidades, a gestão escolar ainda depende de procedimentos manuais para controle de matrículas, frequência e registros acadêmicos, o que pode comprometer a eficiência dos processos administrativos e dificultar a centralização e o gerenciamento das informações educacionais (GIOTTO; CARDOSO, 2012).

De acordo com o Comitê Gestor da Internet no Brasil, embora a transformação digital tenha avançado no setor público brasileiro, persistem desigualdades relacionadas à infraestrutura tecnológica, conectividade e capacidade de utilização de ferramentas digitais, principalmente em municípios com menor disponibilidade de recursos tecnológicos (CGIB, 2024). Como consequência, problemas como duplicidade de informações, dificuldade na geração de relatórios gerenciais e fragmentação dos dados continuam presentes em diversas redes municipais de ensino. Somado a isso, a carência de integração de dados entre as unidades escolares de um mesmo município dificulta o planejamento estratégico das secretarias, que muitas vezes operam com informações fragmentadas. Diante desse cenário, surge a necessidade da criação de ferramentas que democratizem o acesso às tecnologias educacionais na esfera municipal.

1.2 Justificativa

Atualmente, a informatização é essencial para a administração pública e privada. O avanço das TICs tem promovido transformações profundas na gestão pública e educacional,

estabelecendo novos paradigmas de eficiência e transparência (SCHWONKE, 2025). No cenário brasileiro, a adoção de sistemas de gerenciamento escolar tornou-se uma realidade consolidada em instituições privadas e em diversas redes estaduais de ensino, permitindo a automação de processos administrativos e o armazenamento mais seguro dos dados escolares (CANDINHO et al., 2025).

Sobre essa mudança de comportamento, Souza et al. (2023) argumentam que a introdução da computação e de novas tecnologias na rede educacional transformou significativamente o dia a dia e os procedimentos pedagógicos das instituições de ensino. Essa transição para o meio digital não apenas otimizou o tempo dos gestores, mas também qualificou a tomada de decisão baseada em dados precisos e acessíveis em tempo real, por exemplo, passou a ser informatizado a contabilização dos resultados das provas nacionais de desempenho estudantil, como o Índice de Desenvolvimento da Educação Básica (IDEB). Com essa centralização e informatização dos indicadores educacionais, tornou-se consideravelmente mais simples compreender com clareza em qual nível cada unidade escolar se encontra e mapear com precisão quais aspectos pedagógicos ou estruturais precisam de melhorias, permitindo que a secretaria de educação e os gestores sejam substancialmente mais assertivos nas tomadas de decisões estratégicas.

Em termos de soluções baseadas em TICs, Giotto e Cardoso (2012) propuseram um sistema de gestão escolar voltado para a administração da rede municipal de ensino, buscando informatizar processos administrativos e os dados escolares. O trabalho demonstrou a importância da centralização das informações escolares para apoiar a gestão educacional. Entretanto, a solução proposta possui foco predominantemente administrativo, não contemplando de forma integrada funcionalidades como acompanhamento escolar completo do aluno, gerenciamento de múltiplos perfis de usuários e recursos modernos de acesso *web*.

Por sua vez, oliveira (2023) analisou o Sistema Mineiro de Administração Escolar (SIMADE), na oportunidade, foi identificado contribuições importantes da informatização para a gestão educacional. Contudo, o estudo teve caráter avaliativo, concentrando-se na utilização de uma ferramenta já existente e apontando limitações relacionadas à usabilidade, inconsistências de dados e dificuldades operacionais encontradas pelos usuários. Dessa forma, o trabalho não propôs uma nova solução tecnológica capaz de mitigar esses problemas.

De forma semelhante, Abreu et al. (2025), investigaram o desenvolvimento de sistemas *web* para gestão escolar e destacaram os benefícios da utilização de plataformas digitais para o gerenciamento de informações acadêmicas. Entretanto, o estudo concentrou-se na análise conceitual e bibliográfica da temática, sem apresentar a implementação completa de uma solução integrada capaz de atender às demandas administrativas e pedagógicas de uma rede municipal de ensino.

Mais recentemente, Mayrer (2025) propôs um *software web* voltado para o gerenciamento de dados e documentos de estudantes com o objetivo de modernizar o armazenamento e a organização das informações acadêmicas, assim como, de facilitar o acesso aos registros escolares

em ambiente digital. O estudo evidenciou a importância da digitalização documental para a redução do uso de documentos físicos, aumentando a organização das informações e melhorando a eficiência administrativa das instituições de ensino. Entretanto, a solução proposta possui foco específico na gestão de dados e documentos estudantis, não contemplando de forma integrada funcionalidades como matrícula escolar, controle de frequência, lançamento de notas, acompanhamento acadêmico completo e gerenciamento de múltiplos perfis de usuários em uma rede municipal de ensino.

Considerando os trabalhos citados, percebe-se que existem diversas pesquisas relacionadas à informatização da gestão escolar. No entanto, observa-se uma lacuna referente ao desenvolvimento de uma solução *web* integrada que contemple simultaneamente os processos de matrícula, gerenciamento acadêmico, controle de frequência, lançamento de notas, emissão de informações escolares e administração da rede municipal de ensino brasileira em uma única plataforma.

1.3 Objetivos

1.3.1 Objetivo Geral

Este trabalho tem como objetivo desenvolver e implementar um sistema de gestão escolar *web* capaz de integrar diferentes processos administrativos e acadêmicos em um ambiente centralizado.

1.3.2 Objetivos Específicos

- Propor uma solução que atenda múltiplos níveis de acesso, incluindo administrador, diretor, secretário, professor e aluno, permitindo a gestão completa do ciclo acadêmico, desde a matrícula até o acompanhamento escolar.
- Modelar a arquitetura da aplicação utilizando o padrão MVC;
- Criar um banco de dados em modelagem de entidade e relacionamento capaz de atender os requisitos do sistema de gestão escolar;
- Desenvolver módulos para cadastro de usuários e gerenciamento de matrículas, turmas e disciplinas;
- Implementar mecanismos de autenticação e controle de acesso para diferentes perfis de usuários;

- Disponibilizar o código-fonte da aplicação em um repositório público no *GitHub* para possibilitar sua utilização, manutenção e evolução por outros desenvolvedores e instituições.

1.4 Metodologia

Esta pesquisa classifica-se como aplicada, uma vez que não busca apenas estabelecer teorias sobre o problema, mas sim gerar conhecimentos para aplicação prática e solução de um problema da falta de informatização eficiente em escolas municipais brasileiras.

A abordagem do trabalho é qualitativa e tecnológica, pois fundamentou-se no entendimento do cenário e dos desafios das escolas municipais para definir os requisitos do sistema, tendo como foco central a criação de um *software web* como resposta a um problema identificado. A Figura 1 ilustra as etapas do desenvolvimento do sistema, a qual está dividida em análise de cenário e definição de requisitos, modelagem do sistema e banco de dados, e finalizando com o desenvolvimento do software (SARAVANOS; CURINGA, 2023).



Figura 1 – Etapas da metodologia do desenvolvimento da solução proposta

1.4.1 Análise de Cenário e Definição de Requisitos.

A fase inicial deste trabalho consistiu na análise do cenário educacional em vários municípios do Brasil. Esta análise não foi realizada através de visitas presenciais, mas fundamentou-se em uma revisão bibliográfica e documental de artigos científicos e portais governamentais que discutem o estágio atual da informatização nas escolas públicas brasileiras. A Tabela 1 apresenta a lista de artigos e portais visitados.

Tabela 1 – Lista de trabalhos e portais visitados e analisados

ID	Descrição	Tipo de Trabalho	Referência
1	Desenvolvimento de sistemas <i>web</i> para gestão escolar: uma análise sobre funcionalidade, acessibilidade e impacto social	Artigo	ABREU, F. M et al. (2025)
2	Desenvolvimento e aplicabilidade de um sistema de gestão escolar municipal	Artigo	GIOTTO, E.; CARDOSO, C. D. V. (2012)
3	Informática na Educação: a Gestão Escolar e as Fases de Implementação de um Projeto de Tecnologia Educacional	Artigo	SOUZA, E. N. et al. (2023)
4	Desenvolvimento de <i>software web</i> para gerenciamento de dados e documentos de estudantes	Monografia / Trabalho de Conclusão de Curso	MAYRER, M. (2025)
5	Sistema Mineiro de Administração Escolar (SIMADE): avaliação do impacto e da percepção dos usuários operacionais sobre os desafios na gestão das escolas estaduais de Minas Gerais	Monografia / Trabalho de Conclusão de Curso	OLIVEIRA, D. G. de (2023)
6	Pesquisa sobre o uso das tecnologias de informação e comunicação no setor público brasileiro: TIC Governo Eletrônico 2023	Portal Governamental	CGIB (2024)
7	Caminhos da inovação no setor público	Portal Governamental	ENAP (2022)
8	Gestão escolar e tecnologia na educação do Brasil	Portal Internacional	UNESCO (2026)
9	Uso de Ferramentas Digitais na Gestão Escolar	Artigo	CANDINHO, A. A. M. et al (2026)

A partir desse estudo, identificou-se que a ausência de ferramentas digitais adequadas gera uma dependência de processos manuais, o que compromete a segurança dos dados e a agilidade no atendimento à comunidade escolar (SOUZA et al., 2023). Esse diagnóstico foi o ponto de partida para a definição das funcionalidades necessárias para o sistema proposto, visando garantir que a solução fosse aderente à realidade de secretarias municipais que buscam modernizar seus processos.

Com o cenário mapeado, procedeu-se à definição dos requisitos do *software*, que foi desenvolvido para ser uma plataforma *web* unificada para gerenciar o ciclo de vida do aluno na rede municipal, desde a solicitação da matrícula até a gestão do histórico escolar. Segundo Pressman e Maxim (2016), o levantamento e a análise de requisitos são fundamentais para que o *software* atenda às necessidades de seus usuários e garanta a qualidade do produto. Essa etapa permite converter a arquitetura planejada em código funcional, assegurando que as ferramentas de automação atendam à eficiência administrativa esperada para o contexto educacional.

A estrutura do *software* foi dividida em módulos que atendem aos perfis de Administrador, Diretor, Secretária, Professor e Aluno, garantindo que cada usuário tenha acesso apenas às funcionalidades pertinentes ao seu nível de permissão. Conforme destaca a Engenharia de *Software*, essa organização sistemática e sequencial é essencial quando os requisitos sejam bem compreendidos (SOMMERVILLE, 2019), o que facilita o controle técnico de usuários e servidores dentro da rede municipal. Nessa etapa, foi determinada uma segmentação em cinco eixos fundamentais, representada por uma arquitetura modular planejada para assegurar a integridade e a segurança das informações escolares. Os eixos definidos foram o Administrador, o Diretor, o Secretário, o Professor e o Aluno, sendo cada um pertencente a um nível de acesso e permissões específicos. Deste modo, esse modelo de gestão centralizada permite que uma única conta gerencie múltiplos níveis de permissão, transformando processos antes manuais em fluxos digitais integrados, garantindo a integridade dos dados e o suporte à tomada de decisão pedagógica (PRESSMAN; MAXIM, 2016).

Por fim, os requisitos foram classificados em requisitos funcionais, que descrevem as ações que o sistema deve realizar como, por exemplo, o cadastro de usuários, gerenciamento de turmas e o registro de matrículas (SOMMERVILLE, 2019), e em requisitos não funcionais, que se referem às qualidades do sistema, como a responsividade da interface e a segurança no armazenamento das informações (PRESSMAN; MAXIM, 2016). A listagem completa e detalhada de todos os requisitos levantados para o desenvolvimento desta ferramenta encontra-se disponível para consulta no Apêndice A.

Por fim, vale destacar que a documentação de requisitos serviu como guia para a fase de modelagem e posterior implementação do código-fonte, assegurando que o sistema final atendesse aos objetivos de eficiência e transparência propostos para a gestão escolar municipal.

1.4.2 Modelagem do Sistema e Banco de Dados

Nesta fase, foi desenvolvido o projeto lógico e estrutural da solução, com foco central na organização dos dados e na definição da arquitetura do *software* para garantir o controle eficiente das informações e a redução de erros manuais. A modelagem relacional foi desenvolvida utilizando o sistema gerenciador de banco de dados *MySQL* (ELMASRI; NAVATHE, 2018), onde se buscou a criação de um esquema robusto fundamentado na integridade referencial. Conforme destaca Silberschatz et al. (2020), a utilização de modelos relacionais permite que os

dados sejam estruturados de forma a garantir a consistência e a facilidade de recuperação.

Para garantir que as informações de alunos, turmas e matrículas estivessem conectadas corretamente, foram utilizadas chaves primárias e chaves estrangeiras. Essa estrutura evita a repetição de dados e garante que qualquer atualização seja replicada de forma automática e organizada por todo o sistema (SILBERSCHATZ et al., 2020). Além disso, o banco de dados foi estruturado seguindo as regras de normalização, o que torna as buscas mais rápidas e elimina dados desnecessários. Os detalhes técnicos e o dicionário de dados desta estrutura estão disponíveis para consulta no Apêndice B.

Complementarmente, a organização do código foi pautada no padrão de arquitetura *Model-View-Controller (MVC)* (FOWLER, 2006), no qual o *Model* gerencia a lógica de manipulação e interação direta com o banco de dados, a *View* provê uma interface intuitiva e responsiva para o usuário, e o *Controller* atua como intermediário no processamento das requisições. Essa separação de responsabilidades garantiu que a lógica de negócio permanecesse isolada da tela, resultando em um sistema de gestão escolar mais seguro, escalável e eficiente para a realidade municipal.

1.4.3 Desenvolvimento do *Software*

Nesta etapa, a arquitetura previamente planejada foi convertida em código funcional, focando na construção do núcleo do sistema e na criação de ferramentas de automação para garantir a eficiência administrativa. O desenvolvimento foi norteado pelo Modelo Cascata (do inglês, *Waterfall*), uma abordagem clássica da Engenharia de *Software* que prevê a execução de etapas sequenciais e sistemáticas para assegurar a qualidade do produto (SOMMERVILLE, 2019). Conforme destaca Pressman e Maxim (2016), esse modelo é ideal quando os requisitos são bem compreendidos e as mudanças são limitadas durante o processo, o que se aplica à natureza desta ferramenta de gestão escolar.

O ciclo de vida do projeto compreendeu desde a análise detalhada dos requisitos até a implementação das interfaces e funcionalidades. Para a escrita do código-fonte, utilizou-se o ambiente de desenvolvimento integrado *Visual Studio Code (VSCode)*. A construção do *back-end* fundamentou-se na linguagem *Hypertext Preprocessor (PHP)*, que gerenciou a lógica de negócio e o processamento das requisições. O *PHP* foi essencial para realizar a validação rigorosa dos dados antes da persistência no banco de dados, além de permitir o desenvolvimento de *scripts* para as operações completas de *CRUD (Create, Read, Update e Delete)* de usuários com diferentes níveis de acesso, bem como o gerenciamento seguro e controlado de todo o fluxo do ciclo de vida das matrículas.

Complementando o ambiente de desenvolvimento, utilizou-se o *Composer* como gerenciador de dependências do projeto. Essa ferramenta possibilitou a instalação e o gerenciamento automatizado de bibliotecas externas, bem como o carregamento automático de classes (*auto-*

load), contribuindo para a organização estrutural do código e para a adoção de boas práticas no desenvolvimento da aplicação.

A integração do *PHP* com o *MySQL* possibilitou a execução de consultas rápidas, fundamentais para a geração de relatórios e acompanhamento pedagógico em tempo real. Para o dinamismo da interface, utilizou-se a linguagem *JavaScript*, responsável pelas interações em tempo real e pela melhoria da experiência do usuário no preenchimento de formulários.

Para o desenvolvimento do *front-end*, empregou-se o *HyperText Markup Language (HTML5)* para a estruturação semântica e o *Cascading Style Sheets (CSS3)* para a personalização estética, aliados ao *framework Bootstrap*. Essa combinação assegurou uma interface moderna por meio de componentes como formulários estruturados e tabelas dinâmicas, facilitando a automação de processos. O uso do *Bootstrap* garantiu o *design* responsivo, tornando o sistema adaptável a diversos dispositivos, como computadores e *smartphones*, pois é crucial para a democratização do acesso tecnológico. Por fim, a padronização visual do *framework* permitiu uma navegação fluida, visando reduzir a curva de aprendizado para os usuários em transição dos processos manuais para o ambiente digital.

1.5 Estrutura do Trabalho

Este trabalho de conclusão de curso está organizado em quatro capítulos principais, estruturados de modo a apresentar desde o embasamento teórico até a concretização da solução tecnológica proposta. No primeiro capítulo, a Introdução, são apresentadas a contextualização do tema, a justificativa para o desenvolvimento do sistema, os objetivos, bem como a metodologia utilizada para o desenvolvimento do sistema. Também são descritas as etapas adotadas durante o processo de construção da aplicação, incluindo a análise de requisitos, modelagem do sistema e implementação da solução.

O segundo capítulo dedica-se ao Referencial Teórico, onde são explorados os conceitos fundamentais sobre sistemas de gestão escolar e são discutidos conceitos relacionados às Tecnologias da Informação na Educação, sistemas de gestão escolar, Engenharia de *Software*, Modelo Cascata, arquitetura *MVC* e as tecnologias utilizadas no desenvolvimento da aplicação, como *PHP*, *Composer*, *MySQL*, *HTML5*, *CSS3*, *JavaScript* e *Bootstrap*.

O terceiro capítulo apresenta os resultados e discussões, contemplando a modelagem do banco de dados, a arquitetura da aplicação e as funcionalidades implementadas para os diferentes perfis de usuários do sistema. Também são apresentadas as telas desenvolvidas e o funcionamento dos principais módulos responsáveis pelo gerenciamento das atividades acadêmicas e administrativas da rede municipal de ensino.

Por fim, o quarto capítulo apresenta as conclusões do trabalho, destacando os resultados alcançados, as contribuições da solução desenvolvida para a gestão escolar municipal e as

possibilidades de expansão do sistema por meio de trabalhos futuros. Complementam este trabalho os apêndices, contendo a documentação dos requisitos do sistema e o projeto do banco de dados desenvolvido para a aplicação.

2 REFERENCIAL TEÓRICO

2.1 Tecnologias da Informação na Educação

A evolução das TICs no setor público estabeleceu novos paradigmas para a administração educacional, fundamentados nos princípios da eficiência e da transparência de dados (MORAN, 2015). No contexto brasileiro, observa-se um desenvolvimento heterogêneo, onde enquanto sistemas digitais de gestão já são padrões em redes estaduais e no setor privado, em diversos municípios as escolas ainda operam sob lacunas severas de infraestrutura e conectividade (SOUZA et al., 2023). Essa disparidade tecnológica resulta na manutenção de métodos analógicos de registro, o que compromete a agilidade no atendimento à comunidade e a precisão na geração de indicadores estatísticos essenciais para o planejamento governamental.

Deste modo, a implementação de recursos computacionais nas instituições de ensino não se limita ao suporte técnico, mas reconfigura drasticamente o cotidiano e os procedimentos das práticas pedagógicas e administrativas. Souza et al. (2023) argumentam que a modernização desses processos por meio da tecnologia é um fator determinante para otimizar tanto o aprendizado quanto a didática educacional dos professores, criando um ambiente escolar mais dinâmico e organizado. Sob essa ótica, as TICs funcionam como o alicerce para a transição do modelo de registro manual para o digital, permitindo que as limitações logísticas e informacionais da gestão municipal sejam superadas por meio de plataformas centralizadas que integram os dados de forma segura.

Para que a modernização tecnológica seja efetiva, sua integração deve seguir uma abordagem estratégica que promova a eficiência operacional e o suporte à decisão. Conforme reforça Liu et al. (2025), o uso de sistemas automatizados permite que administradores foquem em planejamentos estratégicos, substituindo tarefas rotineiras por processos orientados a dados que otimizam a alocação de recursos e a transparência institucional. Dessa forma, as TICs deixam de ser vistas apenas como ferramentas acessórias e passam a ocupar o papel de eixo central em uma administração pública moderna, capaz de democratizar o acesso à informação, fortalecer a comunicação entre os atores escolares e oferecer respostas rápidas e precisas às demandas da sociedade civil.

2.2 Sistemas de Gestão Escolar e Informatização

Um sistema de gestão escolar atua como uma ferramenta de conexão vital entre a administração pública e a comunidade escolar, integrando as necessidades de cada parte em um ambiente digital unificado (CANDINHO et al., 2025). A informatização de processos como matrícula, registro de frequência e geração de relatórios estatísticos visa simplificar a vida dos usuários e transformar fluxos manuais, como a consulta de vagas, em fluxos automatizados e

intuitivos. De acordo com Laudon e Laudon (2021), a implementação de sistemas de informação nas organizações não é apenas uma escolha técnica, mas uma necessidade estratégica para melhorar a eficiência operacional e transformar a maneira como os serviços são entregues. Essa modernização garante não apenas a agilidade administrativa, mas também a integridade e a segurança dos dados, fundamentais para o acompanhamento da trajetória escolar dos alunos.

Segundo Shah (2014), o uso de sistemas de informação de gestão reduz significativamente a carga de trabalho e melhora a gestão do tempo, permitindo que os agentes escolares acessem ferramentas que qualificam sua atuação e aumentam a utilização dos recursos disponíveis. Assim, sistemas integrados proporcionam uma melhor acessibilidade à informação e elevam a qualidade dos relatórios gerados, facilitando o planejamento de políticas públicas baseadas em evidências sólidas (SHAH, 2014). Por fim, Laudon e Laudon (2021) destacam que a visibilidade ampla do cenário organizacional é essencial para que os gestores tomem decisões precisas e melhorem o desempenho global da instituição.

2.3 Engenharia de *Software* Aplicada

De acordo com Pressman e Maxim (2016), o desenvolvimento de sistemas computacionais envolve um conjunto de práticas, metodologias e técnicas voltadas para a construção de soluções de *software* seguras, organizadas, eficientes e de fácil manutenção. Nesse contexto, a Engenharia de *Software* desempenha papel fundamental na definição de processos estruturados capazes de garantir qualidade durante todas as etapas do ciclo de vida do *software*, desde o levantamento de requisitos até a manutenção da aplicação após sua implantação (SOMMERVILLE, 2019).

Ainda segundo Sommerville (2019), a Engenharia de *Software* consiste na aplicação de abordagens sistemáticas, disciplinadas e mensuráveis para o desenvolvimento, operação e manutenção de *software*. Essa área busca estabelecer métodos capazes de reduzir falhas, aumentar a produtividade e assegurar que os sistemas atendam corretamente às necessidades dos usuários e das organizações.

Além disso, a Engenharia de *Software* permite que aplicações computacionais sejam desenvolvidas de forma planejada e padronizada, utilizando metodologias que auxiliam no controle das etapas do projeto, na documentação dos processos e na organização estrutural do código-fonte (PRESSMAN; MAXIM, 2016). A utilização dessas práticas torna-se essencial em sistemas de gestão escolar, nos quais há grande volume de dados acadêmicos e administrativos que precisam ser processados com confiabilidade, integridade e segurança.

Outro aspecto importante da Engenharia de *Software* está relacionado à modularização e organização da aplicação, permitindo maior facilidade na manutenção, atualização e escalabilidade do sistema (SOMMERVILLE, 2019). Dessa forma, a adoção de metodologias de desenvolvimento e padrões arquiteturais contribui diretamente para a construção de aplicações

mais robustas e reutilizáveis.

Dentre as abordagens utilizadas neste trabalho, destacam-se o Modelo Cascata (*Waterfall*), empregado como metodologia de desenvolvimento para organização sequencial das etapas do projeto, e a arquitetura *MVC (Model-View-Controller)*, utilizada para separação das responsabilidades do sistema em diferentes camadas estruturais. Essas abordagens são detalhadas nas subseções a seguir.

2.3.1 Modelo Cascata (*Waterfall*)

O Modelo Cascata, também conhecido como *Waterfall*, é uma metodologia clássica da Engenharia de *Software* caracterizada pela execução sequencial e linear das etapas de desenvolvimento (SOMMERVILLE, 2019). Nesse modelo, cada fase deve ser concluída e validada antes do início da próxima, garantindo maior controle sobre o processo de construção do *software*. De acordo com Pressman e Maxim (2016), o Modelo Cascata é indicado para projetos cujos requisitos são bem definidos e apresentam poucas alterações durante o desenvolvimento, permitindo maior previsibilidade no planejamento e execução das atividades.

A estrutura do modelo é composta por etapas organizadas de forma hierárquica, abrangendo levantamento de requisitos, análise do sistema, modelagem, implementação, testes e manutenção. Cada uma dessas fases possui documentação própria, permitindo rastreabilidade das informações e maior controle sobre possíveis falhas no desenvolvimento. Uma das principais características do Modelo Cascata é a forte documentação das etapas do projeto, possibilitando melhor acompanhamento do desenvolvimento e facilitando futuras manutenções no sistema.

Outro aspecto relevante desse modelo está relacionado à validação progressiva das etapas, isto é, antes que uma nova fase seja iniciada, a etapa anterior deve estar devidamente concluída e revisada, garantindo maior estabilidade estrutural ao *software*. Apesar de possuir menor flexibilidade em comparação com metodologias ágeis, o Modelo Cascata apresenta vantagens em projetos com escopo bem definido e requisitos estáveis, como sistemas administrativos e acadêmicos, nos quais as funcionalidades seguem processos previamente estabelecidos.

2.3.2 Arquitetura *Model View Controller*

O padrão de arquitetura *MVC* é uma abordagem amplamente utilizada no desenvolvimento de aplicações *web* para promover a separação das responsabilidades do sistema em diferentes camadas estruturais. Essa arquitetura contribui para maior organização do código-fonte, reutilização de componentes e facilidade de manutenção da aplicação. O *MVC* estabelece a separação entre a lógica de negócio, a *interface* gráfica e o processamento das requisições, reduzindo o acoplamento entre os componentes do sistema e tornando o *software* mais modular e escalável (FOWLER, 2006).

A arquitetura é composta por três componentes principais, sendo eles: *Model*, *View* e

Controller. O *Model* é responsável pela manipulação dos dados e pelas regras de negócio da aplicação, realizando operações relacionadas ao banco de dados e ao processamento das informações. A *View* corresponde à camada de interface, sendo responsável pela apresentação visual dos dados ao usuário. Já o *Controller* atua como intermediador entre as camadas, recebendo as requisições do usuário, processando as ações necessárias e encaminhando as respostas adequadas para exibição na interface.

Uma das principais características do *MVC* é a separação entre lógica de negócio e apresentação visual, permitindo que alterações na *interface* não comprometam diretamente o funcionamento das regras do sistema. Essa divisão estrutural também facilita a reutilização de componentes e o trabalho colaborativo durante o desenvolvimento da aplicação. O padrão *MVC* contribui para maior organização do projeto, manutenção simplificada do código-fonte e melhor escalabilidade do sistema, assim, mesmo que o sistema tenha sido desenvolvido sob um escopo de requisitos estáveis e bem definidos, essa divisão em camadas garante que a plataforma esteja pronta para receber atualizações e novas funcionalidades no futuro, caso surjam novas necessidades.

Outra vantagem da arquitetura está relacionada à facilidade de manutenção corretiva e evolutiva, uma vez que cada camada possui responsabilidades específicas e independentes (FOWLER, 2006). Isso reduz a complexidade do sistema e melhora a legibilidade do código.

2.4 Tecnologias *Back-end*

No desenvolvimento de aplicações *web*, a arquitetura cliente-servidor estabelece uma divisão de responsabilidades entre a camada de apresentação, executada no navegador do usuário, e a camada de processamento, executada no servidor. Nesse modelo, o cliente é responsável pela interação com o usuário e envio das requisições, enquanto o servidor executa as regras de negócio, processa os dados e retorna as respostas solicitadas (DEITEL; DEITEL, 2010). Nesse contexto, as tecnologias de *back-end* desempenham papel fundamental na implementação das regras de negócio, processamento das requisições, validação dos dados e gerenciamento da comunicação com os bancos de dados.

Segundo Laudon e Laudon (2021), a camada de *back-end* é responsável pelo processamento das informações e pela execução dos serviços que garantem o funcionamento adequado das aplicações corporativas, atuando como elemento central na integração entre usuários, sistemas e bases de dados. As tecnologias de *back-end* são responsáveis pelo controle de autenticação e autorização de usuários, gerenciamento de sessões, tratamento de exceções, aplicação de mecanismos de segurança e persistência das informações. Dessa forma, constituem o núcleo funcional da aplicação, assegurando que os dados sejam processados de maneira consistente, confiável e segura.

Para o desenvolvimento do sistema proposto, foram utilizadas tecnologias amplamente

consolidadas no ecossistema *PHP*, destacando-se a linguagem *PHP* para implementação da lógica de negócio, o *Composer* para gerenciamento de dependências e organização estrutural do projeto e o *MySQL* como Sistema Gerenciador de Banco de Dados Relacional (SGBDR), responsável pelo armazenamento, gerenciamento e recuperação das informações da aplicação.

2.4.1 PHP

O *PHP* é uma linguagem de programação interpretada amplamente utilizada no desenvolvimento de aplicações *web* dinâmicas. A linguagem opera no modelo cliente-servidor, no qual o navegador do usuário atua como cliente responsável por enviar requisições ao servidor, enquanto o servidor processa os *scripts PHP* e retorna ao cliente o conteúdo gerado em formato *HTML* (PHP, 2025).

Nesse modelo de arquitetura, o código *PHP* é executado no lado do servidor (*server-side*), permitindo que operações sensíveis da aplicação sejam processadas de forma segura antes da resposta ser enviada ao navegador. Diferentemente das linguagens executadas diretamente no cliente, o *PHP* processa dados internamente no servidor, possibilitando maior controle sobre autenticação, sessões, validações e manipulação de banco de dados.

Segundo Welling e Thomson (2017), o *PHP* destaca-se pela facilidade de integração com tecnologias *web* e pela eficiência no processamento de aplicações dinâmicas, sendo amplamente utilizado em sistemas *web* devido a sua flexibilidade e compatibilidade com diferentes servidores e sistemas operacionais. Uma das principais características do *PHP* é sua integração nativa com o *HTML*, permitindo que *scripts* sejam incorporados diretamente às páginas *web* por meio das tags `<?php e ?>`, que por sua vez são responsáveis pela delimitação dos blocos de código *PHP* dentro dos documentos *HTML*. Essa característica possibilita a geração dinâmica de conteúdos, tornando as páginas mais interativas e adaptáveis às ações do usuário.

O *PHP* oferece suporte à programação orientada a objetos, permitindo a utilização de classes, objetos, herança, encapsulamento e reutilização de código, fatores importantes para organização estrutural e manutenção de sistemas de grande porte. Adicionalmente relevante da linguagem, está relacionada ao gerenciamento de sessões e autenticação de usuários. O *PHP* permite controlar sessões por meio da variável global `$_SESSION`, possibilitando armazenamento temporário de informações do usuário durante a navegação no sistema. A linguagem também disponibiliza variáveis globais como `$_POST`, `$_GET` e `$_REQUEST`, utilizadas para recebimento e processamento de dados enviados por formulários e requisições *HTTP*.

O *PHP* também possui ampla integração com SGBDR, especialmente o *MySQL*, permitindo a execução de consultas *SQL* (*do inglês, Structured Query Language*), manipulação de registros e gerenciamento das informações armazenadas na aplicação. Entre suas principais características destacam-se a execução *server-side*, integração com bancos de dados, suporte à programação orientada a objetos, manipulação de formulários, gerenciamento de sessões e

compatibilidade com diferentes ambientes *web*.

2.4.2 *Composer*

O *Composer* é um gerenciador de dependências para a linguagem *PHP*, desenvolvido com o objetivo de automatizar a instalação, atualização e gerenciamento de bibliotecas utilizadas em aplicações *web*. A ferramenta tornou-se um dos principais componentes do ecossistema *PHP* moderno, permitindo que desenvolvedores utilizem pacotes externos de forma organizada e padronizada (NASCIMENTO, 2021). De acordo com Nascimento (2021), o *Composer* simplifica o gerenciamento de dependências ao permitir que bibliotecas e *frameworks* sejam instalados automaticamente a partir de repositórios remotos, reduzindo o trabalho manual e minimizando problemas de compatibilidade entre versões.

O funcionamento do *Composer* baseia-se principalmente no arquivo *composer.json*, responsável por definir as dependências necessárias para o projeto, suas versões e configurações. A partir desse arquivo, a ferramenta realiza o *download* automático dos pacotes requeridos e faz a gerência das atualizações de forma centralizada. Uma das principais características do *Composer* é o gerenciamento automático de dependências, isto é, quando uma biblioteca necessita de outros pacotes para funcionar, o *Composer* identifica e instala todas as dependências necessárias, garantindo compatibilidade entre os componentes da aplicação.

Outra característica importante é a utilização do recurso *autoload*, responsável pelo carregamento automático de classes *PHP*. Esse mecanismo elimina a necessidade de múltiplas instruções como *include* e *require*, tornando o código mais organizado, modular e de fácil manutenção. Além disso, o *Composer* possui integração com o *Packagist*, considerado o principal repositório de pacotes *PHP* da atualidade. Por meio dessa integração, os desenvolvedores podem acessar milhares de bibliotecas voltadas para autenticação, validação de dados, envio de *emails*, geração de relatórios, segurança, testes automatizados e diversas outras funcionalidades (NASCIMENTO, 2021).

2.4.3 Bancos de Dados Relacionais e *MySQL*

Os bancos de dados relacionais constituem uma das principais tecnologias utilizadas no armazenamento, organização e gerenciamento de informações em sistemas computacionais. Esse modelo é fundamentado na organização dos dados em tabelas relacionais compostas por linhas e colunas, permitindo o estabelecimento de vínculos entre diferentes conjuntos de informações por meio de relacionamentos estruturados. Segundo Silberschatz et al. (2020), os bancos de dados relacionais utilizam tabelas, chaves primárias e chaves estrangeiras para garantir consistência, integridade referencial e organização lógica dos dados armazenados. Nesse modelo, cada tabela representa uma entidade específica do sistema, enquanto os relacionamentos estabelecem conexões entre diferentes entidades da aplicação.

Uma das principais características do modelo relacional é a utilização de chaves primárias denominada de *Primary Keys*, responsáveis pela identificação única de cada registro armazenado em uma tabela. Também são utilizadas, as chaves estrangeiras denominada de *Foreign Keys* usada para estabelecer relacionamentos entre tabelas distintas, permitindo a associação consistente entre os dados e reduzindo redundâncias no banco de dados. Outro aspecto importante dos bancos de dados relacionais está relacionado ao processo de normalização, técnica utilizada para organizar as estruturas das tabelas de forma eficiente, reduzindo duplicidade de informações e evitando inconsistências nos registros armazenados. A normalização também contribui para maior integridade dos dados e melhor desempenho das operações realizadas pelo sistema.

Além disso, os sistemas relacionais oferecem mecanismos de controle de integridade, segurança e confiabilidade das informações, garantindo que os dados armazenados permaneçam consistentes mesmo em operações simultâneas realizadas por múltiplos usuários. O gerenciamento dessas informações é realizado por meio de SGBDR, responsáveis pelo controle de acesso, execução de consultas, armazenamento físico dos dados e gerenciamento das transações da aplicação.

Dentre os SGBDRs mais utilizados no desenvolvimento *web*, destaca-se o *MySQL*, amplamente empregado em aplicações devido à sua performance, estabilidade e integração com diversas linguagens de programação, especialmente o *PHP*. O *MySQL* é um SGBDR de código aberto que utiliza a linguagem *SQL* para manipulação e gerenciamento das informações armazenadas. A *SQL* possibilita a execução de comandos responsáveis por consultas, inserções, atualizações e exclusões de registros, além da criação e gerenciamento das estruturas do banco de dados.

Por fim, entre as principais características do *MySQL* destacam-se o suporte a múltiplos usuários, controle de transações, gerenciamento de permissões, mecanismos de segurança, alta capacidade de processamento e compatibilidade com aplicações *web* de diferentes portes. O *MySQL* também oferece suporte à integridade referencial, permitindo o controle automático dos relacionamentos entre tabelas e garantindo maior confiabilidade das informações armazenadas no sistema.

2.5 Tecnologias *Front-end*

Em aplicações *web*, a camada de *front-end* corresponde ao conjunto de tecnologias responsáveis pela apresentação das informações e pela interação entre o usuário e o sistema. Essa camada é executada diretamente no navegador e tem como objetivo transformar os dados processados pelo servidor em interfaces visuais acessíveis, intuitivas e funcionais (FREEMAN, 2019).

Segundo Deitel e Deitel (2010), o *front-end* representa a camada de apresentação visual das aplicações *web*, sendo responsável pela construção da interface gráfica, exibição dos conteú-

dos e tratamento das interações realizadas pelos usuários. Dessa forma, essa camada atua como intermediária entre o usuário e as funcionalidades disponibilizadas pelo sistema.

As tecnologias de *front-end* desempenham papel fundamental na usabilidade, acessibilidade e experiência do usuário, permitindo a criação de interfaces responsivas capazes de se adaptar a diferentes dispositivos e resoluções de tela. Essas tecnologias também possibilitam a validação de informações, manipulação dinâmica de elementos visuais e aprimoramento da navegação dentro da aplicação.

2.5.1 HTML5

O *HyperText Markup Language (HTML5)* é a linguagem de marcação responsável pela estruturação lógica e semântica do conteúdo das páginas *web*. Sua principal função consiste em organizar os elementos que compõem a interface de uma aplicação, definindo a estrutura que será posteriormente estilizada pelo *CSS* e manipulada pelo *JavaScript*. De acordo com Freeman (2019), o *HTML5* representa uma evolução significativa das versões anteriores da linguagem ao introduzir recursos voltados para semântica, acessibilidade, multimídia e interoperabilidade entre diferentes dispositivos e navegadores.

A estrutura de um documento *HTML* é composta por elementos delimitados por tags, responsáveis por identificar e organizar os diversos componentes da página. Entre as principais tags utilizadas destacam-se ‘<html>’, que define o documento; ‘<head>’, responsável pelas configurações e metadados; ‘<body>’, que contém os elementos visíveis ao usuário; além de componentes estruturais como ‘<header>’, ‘<nav>’, ‘<section>’, ‘<article>’, ‘<aside>’ e ‘<footer>’, que contribuem para a organização semântica do conteúdo.

Uma das principais características do *HTML5* é a introdução de elementos semânticos que permitem aos navegadores, mecanismos de busca e tecnologias assistivas compreender melhor a função de cada parte da página. Essa abordagem melhora a acessibilidade, a indexação por motores de busca e a manutenção do código.

Dessa forma, o *HTML5* oferece suporte nativo a recursos multimídia por meio das tags ‘<audio>’ e ‘<video>’, reduzindo a necessidade de tecnologias externas para reprodução de conteúdo multimídia. A linguagem também aprimorou os formulários *web* através da inclusão de novos tipos de campos, como *email*, *data*, *telefone* e *URL*, permitindo validações básicas diretamente no navegador, além, de sua compatibilidade com aplicações responsivas e integração com *APIs* modernas.

2.5.2 CSS3

O *Cascading Style Sheets (CSS3)* é uma linguagem utilizada para definir a apresentação visual de documentos *HTML*. Sua principal função consiste em controlar aspectos relacionados à aparência da interface, como cores, fontes, espaçamentos, alinhamentos, posicionamentos

e comportamento responsivo dos elementos da página. De acordo com Duckett (2016), o CSS permite separar a camada de apresentação da estrutura do conteúdo, promovendo maior organização, reutilização de código e facilidade de manutenção das aplicações *web*.

O funcionamento do CSS baseia-se na aplicação de regras compostas por seletores e propriedades. Os seletores identificam os elementos que receberão determinada formatação, enquanto as propriedades definem as características visuais que serão aplicadas. A ampliação dos recursos de estilização avançada é considerado uma evolução do CSS, incluindo sombras, gradientes, bordas arredondadas, transformações, transições e animações. Esses recursos permitem a criação de interfaces modernas sem a necessidade de bibliotecas externas ou imagens adicionais.

Outra característica relevante é o suporte aos sistemas *Flexbox* e *CSS Grid Layout*, mecanismos que facilitam o posicionamento e alinhamento dos elementos na página, permitindo a construção de *layouts* complexos de forma mais eficiente. O CSS3 também desempenha papel fundamental no desenvolvimento responsivo por meio das *Media Queries*, recurso que possibilita adaptar automaticamente a interface para diferentes resoluções e dispositivos, como computadores, *tablets* e *smartphones*.

2.5.3 JavaScript

O *JavaScript* é uma linguagem de programação de alto nível amplamente utilizada no desenvolvimento de aplicações *web* para implementação de funcionalidades dinâmicas e interativas. Diferentemente do *PHP*, que executa suas instruções no lado do servidor, o *JavaScript* é tradicionalmente executado no lado do cliente (*client-side*), diretamente pelo navegador do usuário, permitindo respostas imediatas às ações realizadas na interface.

O *JavaScript* é uma das principais tecnologias da *web* moderna, sendo responsável por adicionar comportamento dinâmico às páginas, possibilitando a manipulação de elementos visuais, validação de dados, tratamento de eventos e atualização de conteúdos em tempo real (FLANAGAN, 2020). Uma das principais características da linguagem é a capacidade de manipular o *Document Object Model (DOM)*, estrutura que representa os elementos de uma página *web* como objetos programáveis. Por meio dessa manipulação, o *JavaScript* pode alterar textos, imagens, estilos, formulários e demais componentes da interface sem a necessidade de recarregamento da página.

Além do mais, a linguagem possui suporte ao tratamento de eventos, permitindo a execução de ações específicas em resposta às interações do usuário, como cliques em botões, preenchimento de formulários, movimentação do mouse e utilização do teclado. Esse mecanismo possibilita a criação de telas mais responsivas e interativas. Outra característica importante do *JavaScript* é a realização de validações no lado do cliente, permitindo verificar informações inseridas pelo usuário antes do envio ao servidor. Essa abordagem contribui para a melhoria da

experiência do usuário e reduz o processamento desnecessário no servidor, uma vez que erros simples podem ser identificados imediatamente na interface.

O *JavaScript* também oferece suporte à programação assíncrona, permitindo a troca de informações com o servidor sem a necessidade de recarregar completamente a página. Esse recurso é amplamente utilizado por meio de tecnologias como (*Asynchronous JavaScript and XML (AJAX)*) e *APIs* modernas, possibilitando atualizações dinâmicas de conteúdo e maior desempenho das aplicações *web*.

Outros aspectos relevantes do *JavaScript* são: a execução no lado do cliente, manipulação do DOM, tratamento de eventos, validação de formulários, programação assíncrona e integração nativa com *HTML* e *CSS*. Essas funcionalidades tornam o *JavaScript* uma das linguagens fundamentais para o desenvolvimento *front-end* moderno.

2.5.4 *Bootstrap*

O *Bootstrap* é um *framework front-end* de código aberto desenvolvido para acelerar a criação de *interfaces web* modernas, responsivas e padronizadas. O *framework* é composto por um conjunto de arquivos *CSS* e *JavaScript* que disponibilizam componentes prontos e reutilizáveis para o desenvolvimento de aplicações *web*.

O *Bootstrap* foi concebido com o objetivo de padronizar telas e reduzir o tempo necessário para o desenvolvimento *front-end*, fornecendo uma biblioteca consistente de estilos e componentes (*BOOTSTRAP*, 2026). Uma das principais características do *framework* é seu sistema de *grid* responsivo baseado em colunas, que permite organizar o conteúdo da página de forma flexível e adaptável a diferentes tamanhos de tela. Esse mecanismo facilita a construção de *layouts* compatíveis com dispositivos móveis e *desktops*.

O *Bootstrap* disponibiliza diversos componentes pré-construídos, incluindo formulários, tabelas, botões, barras de navegação, modais, alertas, *cards* e menus responsivos. Esses elementos seguem padrões visuais consistentes e podem ser personalizados conforme as necessidades da aplicação. Para mais, o *framework* possui compatibilidade com os principais navegadores modernos e oferece recursos voltados para acessibilidade e usabilidade, contribuindo para uma experiência de navegação mais intuitiva.

Outro aspecto importante está relacionado à integração com *JavaScript*, permitindo a implementação de funcionalidades interativas como menus expansíveis, janelas modais, carrosséis de imagens e validações visuais.

2.6 Ferramenta de Desenvolvimento

O desenvolvimento de *software* moderno não depende apenas das linguagens de programação e tecnologias utilizadas na implementação da aplicação, mas também de ferramentas que

auxiliam os desenvolvedores durante todo o ciclo de desenvolvimento. Essas ferramentas fornecem recursos voltados para edição de código, depuração, gerenciamento de projetos, controle de versões e automação de tarefas, contribuindo para a qualidade e produtividade do processo de desenvolvimento (PRESSMAN; MAXIM, 2016).

De acordo com Sommerville (2019), o uso de ferramentas de apoio ao desenvolvimento é fundamental para aumentar a eficiência das atividades de programação, reduzir a ocorrência de erros e facilitar a manutenção dos sistemas ao longo de seu ciclo de vida. Além disso, essas ferramentas permitem maior organização do código-fonte, rastreabilidade das alterações e suporte às boas práticas da Engenharia de *Software*.

Nesse contexto, as ferramentas de desenvolvimento desempenham papel estratégico na construção de aplicações computacionais, auxiliando na implementação, depuração, manutenção e controle de versões do *software*. Para o desenvolvimento do sistema de gestão escolar, foram utilizados o *Visual Studio Code* como ambiente de desenvolvimento, o *Git* para controle de versão do código-fonte e o *GitHub* para armazenamento remoto e gerenciamento do projeto.

2.6.1 *Visual Studio Code*

O *Visual Studio Code (VSCode)* é um ambiente de desenvolvimento integrado (IDE) desenvolvido pela *Microsoft* para edição, gerenciamento e depuração de código-fonte. A ferramenta é amplamente utilizada no desenvolvimento de aplicações *web* devido à sua flexibilidade, desempenho e suporte a múltiplas linguagens de programação.

Segundo a *Microsoft* (2026), o *VSCode* foi projetado para oferecer uma experiência de desenvolvimento moderna, combinando recursos avançados de edição com uma arquitetura leve e altamente extensível. Dentre suas principais características evidencia-se o destaque de sintaxe (*syntax highlighting*), que facilita a identificação visual dos elementos do código, o *IntelliSense* que é responsável pela sugestão automática de comandos, variáveis e funções, e o sistema de depuração (*debugging*) que auxilia na identificação e correção de erros durante o desenvolvimento.

A ferramenta também possui integração nativa com sistemas de controle de versão, especialmente o *Git*, permitindo o gerenciamento de alterações no código-fonte e o trabalho colaborativo entre desenvolvedores. Outra característica importante é o suporte a extensões, que possibilita adicionar funcionalidades específicas para diferentes tecnologias como *PHP*, *JavaScript*, *HTML*, *CSS*, bancos de dados e *frameworks* modernos. Essa capacidade torna o *VSCode* altamente adaptável a diferentes ambientes de desenvolvimento.

Ademais, o editor oferece terminal integrado, navegação inteligente entre arquivos, refatoração de código e recursos de produtividade que contribuem para o desenvolvimento de aplicações de médio e grande porte.

2.6.2 *Git*

O *Git* é um sistema de controle de versão distribuído amplamente e utilizado no desenvolvimento de *software* para gerenciar alterações realizadas no código-fonte ao longo do ciclo de vida de um projeto. Criado por *Linus Torvalds* em 2005, o *Git* permite registrar, rastrear e controlar modificações em arquivos, proporcionando maior segurança, organização e colaboração entre desenvolvedores. Segundo Chacon e Straub (2014), o *Git* utiliza uma arquitetura distribuída na qual cada desenvolvedor possui uma cópia completa do repositório, incluindo todo o histórico de alterações. Essa característica garante maior confiabilidade, desempenho e flexibilidade no gerenciamento dos projetos, além de reduzir a dependência de um servidor central.

Entre as principais funcionalidades do *Git* destacam-se o versionamento de código, controle de histórico de alterações, criação de ramificações chamadas de *branches*, integração de modificações por meio de operações de *merge* e recuperação de versões anteriores do sistema. Esses recursos permitem que diferentes funcionalidades sejam desenvolvidas simultaneamente sem comprometer a estabilidade da aplicação principal.

Outro aspecto importante do *Git* é a utilização de *commits*, que representam registros das alterações realizadas no projeto. Cada *commit* armazena informações sobre as modificações efetuadas, permitindo rastrear a evolução do sistema, identificar mudanças específicas e facilitar processos de manutenção corretiva e evolutiva. O *Git* contribui para a adoção de boas práticas de desenvolvimento de *software*, promovendo maior controle sobre as versões do sistema e facilitando o trabalho colaborativo entre desenvolvedores.

2.6.3 *GitHub*

O *GitHub* é uma plataforma de hospedagem de repositórios baseada em nuvem que utiliza o *Git* como mecanismo de controle de versão. A plataforma foi desenvolvida para facilitar o armazenamento, compartilhamento e gerenciamento de projetos de *software*, permitindo a colaboração entre desenvolvedores de forma organizada e segura.

De acordo com a documentação oficial do *GitHub*, a plataforma disponibiliza recursos que auxiliam no desenvolvimento colaborativo, incluindo hospedagem de repositórios, gerenciamento de versões, revisão de código, rastreamento de problemas e integração com ferramentas de automação (GITHUB, 2026). Entre as principais funcionalidades do *GitHub* destacam-se o armazenamento remoto de repositórios, gerenciamento de *branches*, realização de *pull requests*, controle de *issues*, documentação por meio de arquivos *README* e integração com pipelines de Integração Contínua e Entrega Contínua (*Continuous Integration/Continuous Delivery – CI/CD*).

Outro recurso relevante da plataforma é a possibilidade de realizar revisões de código antes da integração das alterações ao projeto principal, contribuindo para a qualidade, segurança e padronização do desenvolvimento. O *GitHub* funciona como uma camada adicional de segurança para os projetos, permitindo o armazenamento remoto do código-fonte e a recuperação de versões

anteriores sempre que necessário.

3 RESULTADOS

Este capítulo apresenta todas as telas do *software* de gestão escolar que foi desenvolvido com o objetivo de facilitar e auxiliar a gestão educacional municipal, integrando as necessidades de alunos, pais, professores e administração em um ambiente digital unificado.

A solução foi concebida para atuar como uma ferramenta de conexão entre a administração pública e a comunidade escolar. Estruturalmente, o *software* segue o modelo de sistemas integrados de gestão, visando simplificar a vida dos usuários através da tecnologia e transformar processos manuais, como a consulta de vagas e matrículas, em fluxos automatizados e intuitivos.

Desse modo, serão apresentadas a descrição e imagem da tela de cada uma das funcionalidades. O *software* de gestão escolar implementa um rigoroso controle de acessos, segmentando as funcionalidades em áreas distintas para garantir a segurança e a integridade das informações escolares do usuário. Cada perfil de usuário interage com uma *interface* gráfica própria, adequada para cada nível de usuário e suas responsabilidades, sendo elas: Área Administrativa e de Direção, Área de Secretaria, Área do Professor e do Aluno.

O processo de ingresso na plataforma inicia-se pela página inicial do sistema que apresenta uma proposta de Educação Digital, destacando-se pela simplicidade na navegação. Conforme ilustrado na Figura 2, o topo da página contém o logotipo do sistema, tela de boas-vindas com as opções de “Ver Unidades” e “Criar Conta”, uma área de Consulta Pública e o botão de *Login* para usuários cadastrados.

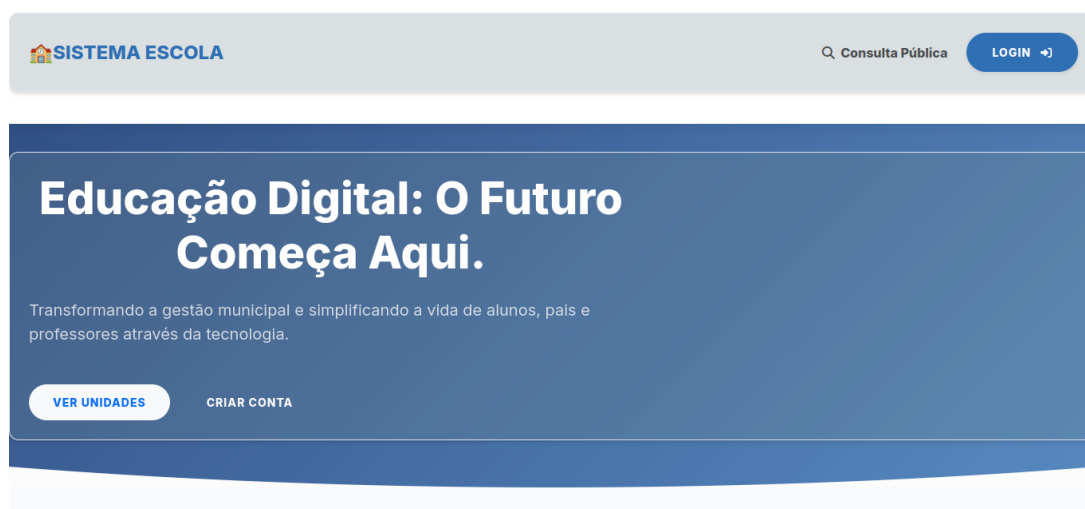


Figura 2 – Página Inicial do Sistema Escola

A funcionalidade de ver unidades permite que a comunidade visualize as instituições de ensino cadastradas. Na Figura 3, observa-se o catálogo de escolas, onde cada cartão apresenta informações cruciais como: Nome da Unidade, Endereço e Contato telefônico, Localização, e *Status* de Inscrição. Quando a unidade apresenta *status* “Inscrições Abertas” o usuário pode iniciar o processo de matrícula por meio da opção “Solicitar Vaga”.

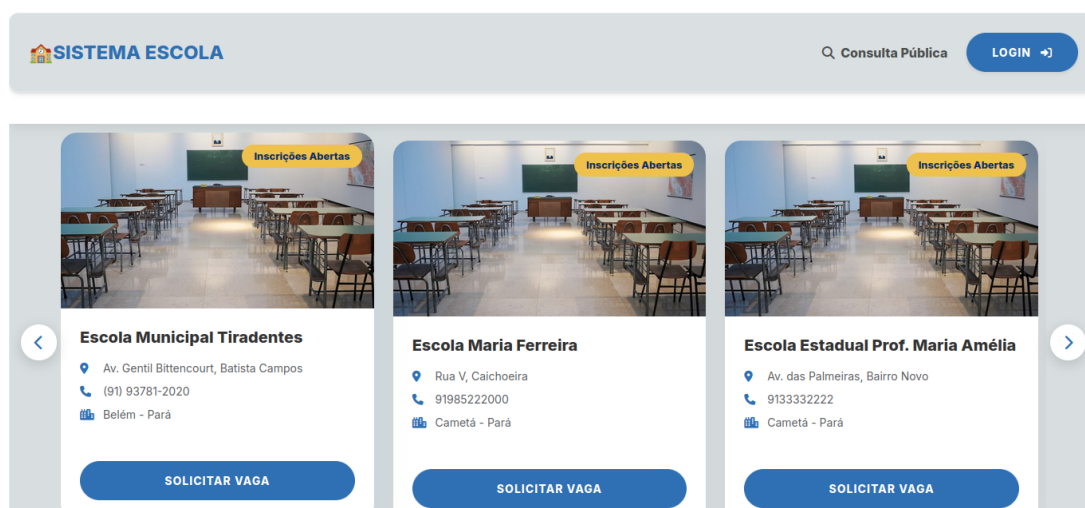


Figura 3 – Catálogo de Unidades Escolares

Para garantir a integridade dos dados escolares do usuário, o sistema implementa um controle rigoroso de acesso. Na tela de Registro, ilustrada na Figura 4, o usuário é convidado a preencher os dados completos como nome, *email*, telefone, endereço de residência, *login* e senha.

Figura 4 – Formulário de Registro de Novo Usuário

A Figura 5 apresenta a tela de Login, responsável por fornecer um ambiente seguro para autenticação dos usuários. Nessa tela, o usuário deve informar suas credenciais de *login* e senha. Além disso, o sistema ainda oferece a opção de recuperar a senha através do *link* “Esqueceu senha?”. Por fim, a tela também ilustra a opção via *link* para a criação de novos cadastros.

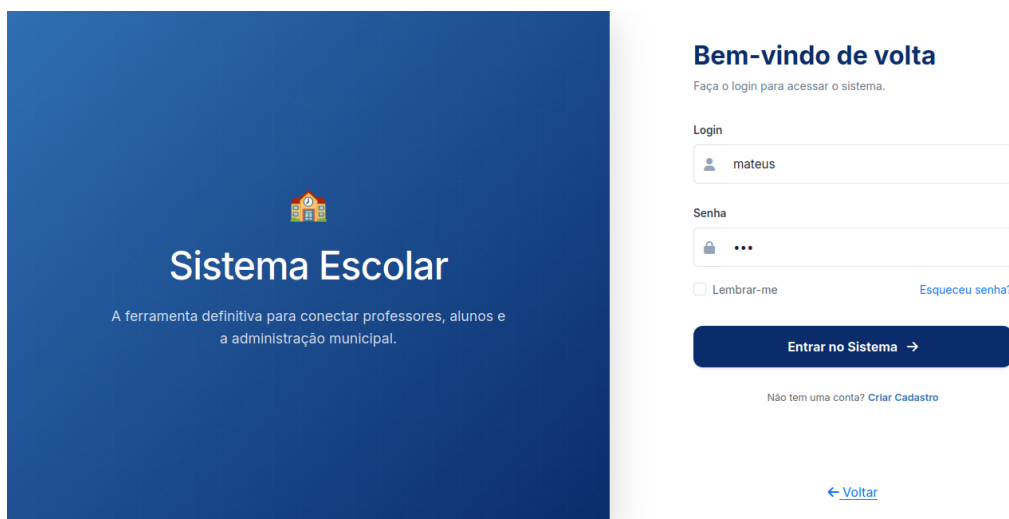


Figura 5 – Tela de Login do Sistema

3.1 Portais de Acesso por Perfil

Após a autenticação bem-sucedida, o sistema direciona o usuário para a tela de seleção de perfis, evidenciando sua arquitetura modular que permite ao usuário ter múltiplos perfis, como mostra a Figura 6.

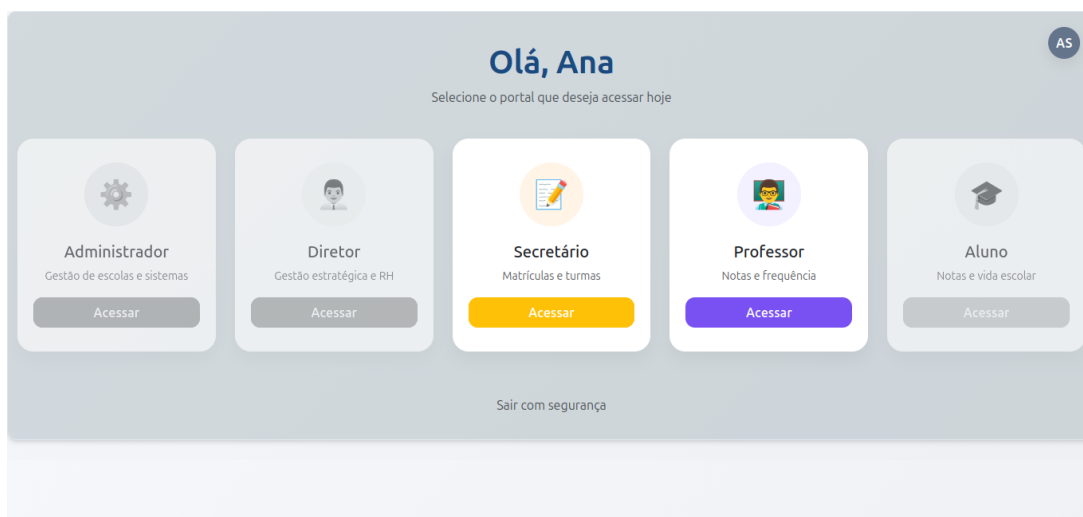


Figura 6 – Central de Seleção de Portais de Perfil

Essa estrutura centralizada permite que uma única conta gerencie múltiplos níveis de permissão de acesso, segmentados em cinco eixos fundamentais: o Administrador, responsável pela gestão global de escolas e configurações; o Diretor e o Secretário, focados na visão estratégica, recursos humanos, operação de processos de matrículas e turmas; o Professor, habilitado para lançamentos de notas e frequência; e o Aluno, restrito à visualização de desempenho e acompanhamento da vida escolar.

3.2 Área do Administrador

A área do administrador é o núcleo operacional da plataforma, projetada para fornecer uma visão macro da rede de ensino municipal, assim como para prover o acesso à ferramentas facilitadoras da gestão técnica de escolas, de usuários e de servidores. O painel utiliza um *layout* com menu lateral fixo para navegação rápida e uma área de conteúdo dinâmico que responde às seleções do usuário.

3.2.1 Visão Geral e Navegação

Ao realizar a autenticação com perfil administrativo, o usuário é direcionado ao Painel ADM. Conforme ilustrado na Figura 7, composta por uma mensagem de boas-vindas do administrador e um menu de navegação lateral que reúne as principais funcionalidades do sistema, como Cadastrar Escola, Cadastro Usuário, Cadastrar Servidor, Consulta, Calendário Escolar, Matrículas Pendentes e Relatório de Servidores. No topo à direita, são exibidos o nome e o perfil do usuário logado.

Essa mesma *interface* de tela inicial de recepção ao usuário se repete na área do diretor, secretário e professor.

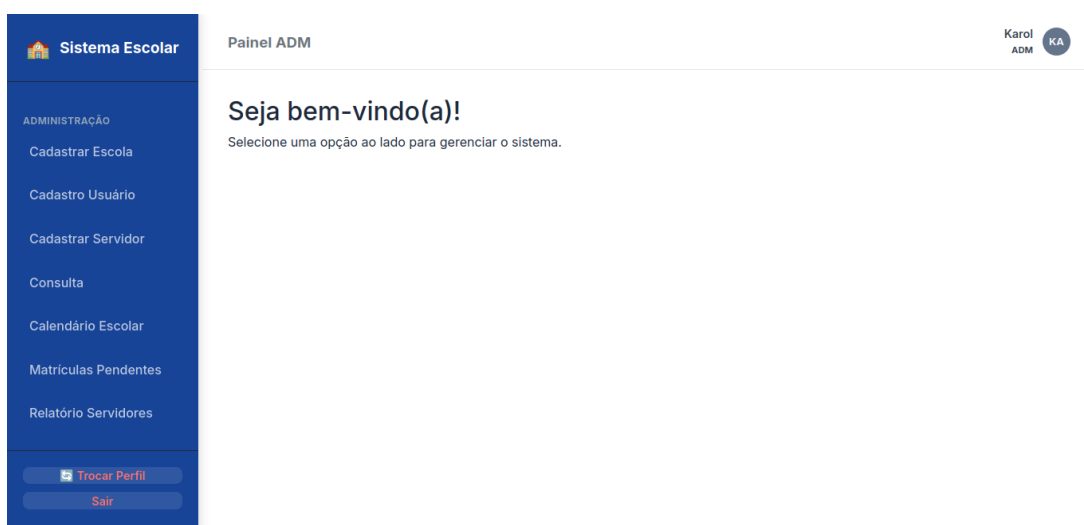


Figura 7 – Visão Inicial do Painel Administrativo

3.2.2 Gestão de Unidades Escolares e Liderança

O módulo de Gestão de Unidades Escolares permite a inserção de novas instituições no banco de dados. Como observado na Figura 8, a tela de cadastro de novas unidades e associação estratégica de diretores, possui o formulário de cadastro que exige dados estruturais como nome da instituição, CNPJ e informações de localização logradouro, número, bairro, complemento e município. Adicionalmente, o sistema implementa a funcionalidade de “Gestão de Liderança”,

permitindo associar um servidor previamente cadastrado para responder como diretor da unidade escolhida.

Figura 8 – Interface de Cadastro de Escola e Vínculo de Liderança

3.2.3 Cadastro e Gestão de Usuários

Garantindo o controle e a integridade dos dados escolares, o sistema estrutura o controle de acesso por meio de interfaces especializadas para registro e monitoramento. A Figura 9 ilustra a tela para a criação do registro de um novo usuário cadastrado por um usuário com perfil administrador.

Figura 9 – Formulário de Registro de Usuário pelo ADM

A coleta de informações é segmentada em dois blocos lógicos, sendo eles: de cadastro de Dados Pessoais, que compreende a coleta do nome completo, telefone e endereço residencial do novo usuário, e o bloco de Acesso ao Sistema, que por sua vez exige o preenchimento de *email*, *login* e definição de senha com campo de confirmação. Complementarmente, para o

acompanhamento dos registros concluídos, o sistema possui a funcionalidade de consulta geral, como ilustra a Figura 10.

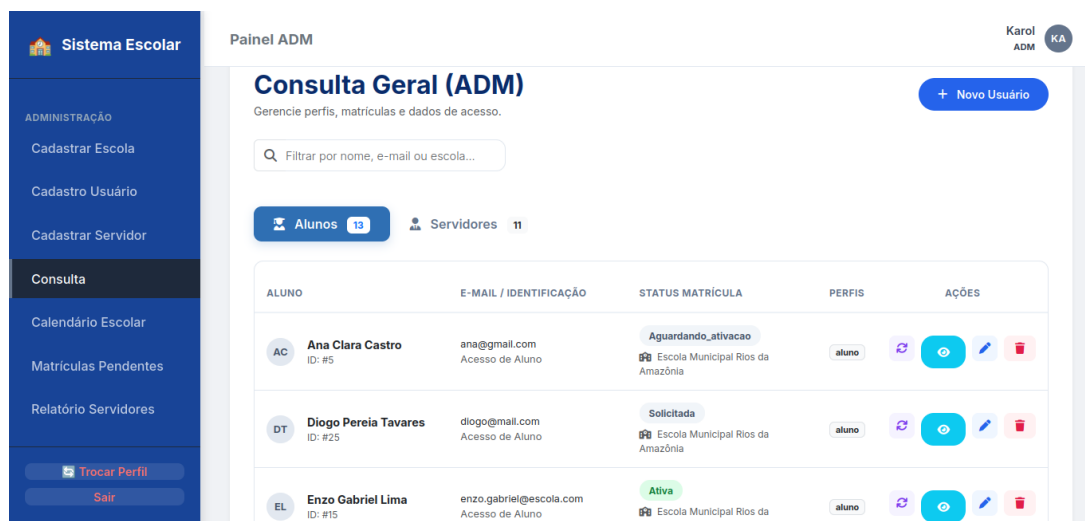


Figura 10 – Tela de Consulta Geral de Alunos e Servidores

A tela consulta geral oferece uma visão tabular dos registros existentes, permitindo a filtragem dinâmica por nome, *email* ou unidade escolar. Esta visualização exibe o *status* da matrícula classificada como ativa, solicitada ou aguardando ativação, e disponibiliza ferramentas de gestão direta como a visualização de perfis, edição de dados ou remoção de registros.

3.2.4 Vínculo Profissional de Servidores

A plataforma proposta serve para automatizar a alocação de servidores por meio da funcionalidade de Vínculo de Servidor à Unidade Escolar. Este módulo permite a associação de usuários com perfil de servidor à cargos e unidades específicas da rede municipal, como mostra a Figura 11.

A imagem mostra o formulário de 'Vínculo de Servidor à Unidade Escolar'. No topo, há uma barra de navegação com o nome do sistema e o perfil do usuário. À esquerda, um menu lateral contém opções como 'ADMINISTRAÇÃO', 'Cadastrar Escola', 'Cadastrar Usuário', 'Cadastrar Servidor' (destacado), 'Consulta', 'Calendário Escolar', 'Matrículas Pendentes' e 'Relatório Servidores'. O conteúdo principal é o 'Painel ADM' com o título 'Vínculo de Servidor à Unidade Escolar'. Abaixo do título, há uma barra de busca para filtrar por nome, e-mail ou escola. Abaixo disso, há dois botões para alternar entre 'Alunos' (13) e 'Servidores' (11). O formulário contém seções para 'IDENTIFICAÇÃO E LOCALIZAÇÃO' (com campos para Servidor e Unidade Escolar), 'DETALHES PROFISSIONAIS' (com campos para Cargo / Função, Regime de Trabalho e Data de Admissão) e 'VIGÊNCIA E OBSERVAÇÕES' (com campos para Fim da Validade e Observações Internas).

Figura 11 – Formulário de Vínculo de Servidores

O formulário de vínculo de servidores, possui um fluxo de preenchimento que se organiza nos pilares de Identificação, Detalhes Profissionais e Vigência. A Identificação consiste na seleção do servidor e da escola correspondente. Os Detalhes Profissionais abrangem a definição de cargo, função, regime de trabalho e data de admissão. Já a Vigência estabelece o prazo de validade do vínculo e disponibiliza um campo para observações internas.

3.2.5 Gestão de Calendário Escolar Municipal

Visando garantir a padronização dos prazos em toda a rede de ensino, o sistema disponibiliza ao administrador o módulo de Gestão de Calendário Escolar, que podemos ver na Figura 12.

A imagem mostra a interface de um sistema web. No topo, há uma barra azul com o ícone de uma casa e o texto "Sistema Escolar". Abaixo, um menu lateral azul contém as opções: ADMINISTRAÇÃO, Cadastrar Escola, Cadastro Usuário, Cadastrar Servidor, Consulta, Calendário Escolar (destacado), Matrículas Pendentes e Relatório Servidores. No topo da área principal, há "Painel ADM" e o nome de usuário "Karol ADM" com um ícone de perfil. O título principal é "Gestão de Calendário Escolar (Municipal)" com um subtítulo "Defina as datas e períodos de lançamento para o Ano Letivo **2026**". O formulário contém três seções: 1. "Período de Matrícula" com campos para "Início da Matrícula" (05/01/2026) e "Fim da Matrícula" (30/01/2026). 2. "Período de Aulas" com campos para "Início das Aulas" (17/02/2026) e "Fim das Aulas" (21/12/2026). 3. "Períodos de Lançamento de Notas (Professores)" com campos para "Início Lançamento AV1" (12/04/2026) e "Fim Lançamento AV1" (27/04/2026). Cada campo de data possui um ícone de calendário para seleção.

Figura 12 – Configuração de Prazos e Calendário Letivo

Nesta área, são estabelecidos os prazos obrigatórios a serem respeitados para o pleno cumprimento do ano letivo vigente. Para isso, o preenchimento é dividido em Período de Matrícula, Período de Aulas e Lançamento de Notas. No Período de Matrícula são delimitadas as datas de início e término para novas matrículas. No Período de Aulas é definido o período que compreende o cronograma letivo anual. Por fim, no Lançamento de Notas são estipulados os prazos específicos para que o professor realize o registro de avaliações e processos de recuperação.

3.2.6 Gestão de Matrículas e Relatórios

O sistema atua como um facilitador da análise de solicitações de vagas. Como é mostrado na tela de Gestão de Matrículas na Figura 13, o administrador pode filtrar pendências por unidade escolar e executar ações de “ativar” ou “cancelar” solicitações de alunos, exibindo o código do processo e o CPF do solicitante.

Sistema Escolar Painel ADM Karol ADM KA

Gestão de Matrículas
Análise e aprovação de solicitações pendentes da rede escolar.

Filtrar por Unidade Escolar
Todas as Escolas

CÓDIGO	ALUNO / CPF	ESCOLA SOLICITADA	ANO LETIVO	AÇÕES
#2025-000001	Ana Clara Castro 02333322211	Escola Municipal Rios da Amazônia	2026	Ativar Cancelar
#2026-000006	Tamyris Baia Silva 217.954.584-16	Escola Municipal Rios da Amazônia	2026	Ativar Cancelar
#2026-000009	Lucio Costa de Oliveira 023.789.456-24	Escola Estadual Prof. Maria Amélia	2026	Ativar Cancelar

Figura 13 – Painel de Aprovação de Matrículas

A tela de relatório dos servidores apresenta a relação dos servidores em formato de lista. A tela apresenta informações sobre o cargo, dados de contato e *status* funcional de cada funcionário da rede, conforme ilustra a Figura 14.

Sistema Escolar Painel ADM Karol ADM KA

Relatório de Servidores
Consulte a listagem de funcionários e seus vínculos ativos.

Selecione a Unidade Escolar: Escola Municipal Rios da Amazônia Status do Vínculo: Todos os status Aplicar Filtros

NOME DO SERVIDOR	CARGO / FUNÇÃO	CONTATOS	STATUS	DATA ADMISSÃO
Maria da Silva ID: #0002	Diretor	maria@gmail.com 91988887777	ATIVO	11/09/2025 Vínculo Escolar
Ana Beatriz Silva ID: #0014	Professor	ana.beatriz@email.com 91964567890	ATIVO	20/01/2026 Vínculo Escolar
Bernardo Carvalho Amorim ID: #0027	Professor	bernardo@gmail.com 91983794562	ATIVO	05/02/2026 Vínculo Escolar
Pedro Andrade ID: #0004	Professor	pedro@gmail.com 91966665555	ATIVO	20/01/2026 Vínculo Escolar

Figura 14 – Emissão de Relatórios de Servidores

3.3 Área do Diretor e Secretário

A área do diretor foi projetado para oferecer uma gestão centralizada e estratégica de uma unidade de ensino específica. Desse modo, o diretor possui uma visão focada no corpo de servidores e alunos da sua escola, permitindo o acompanhamento de metas pedagógicas e operacionais. A área do secretário tem as mesmas funções que a área do diretor tem, nesse caso, tudo que o diretor faz, o secretário pode fazer também.

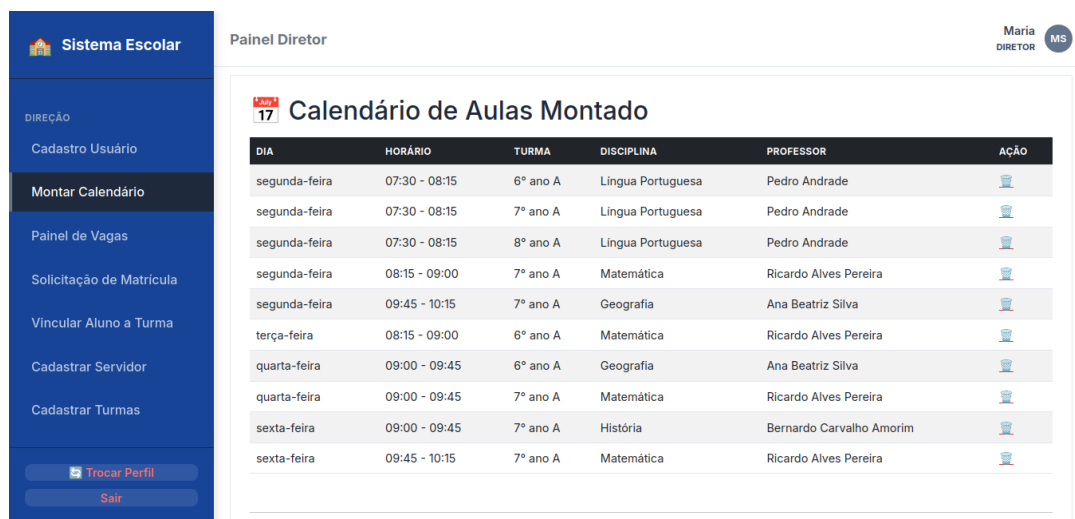
3.3.1 Gestão de Usuários

Esta funcionalidade permite fazer novos registros de usuários no sistema, coletando dados essenciais para identificação e controle de acesso. Como podemos ver na Figura 15, o processo é dividido em seções de dados pessoais e credenciais de acesso, a tela de cadastro solicita informações como nome completo, telefone e endereço nos campos de Dados Pessoais, além de *email*, *login* e senha na seção de Acesso ao Sistema. Existe também um botão Ver Lista para consulta de registros existentes.

Figura 15 – Novo Registro de Usuário

3.3.2 Montagem e Gerenciamento do Calendário de Aulas

O sistema oferece uma ferramenta dinâmica para a organização do cronograma escolar. A tela permite visualizar o quadro de horários completo e adicionar novos períodos de aula de forma vinculada a turmas, disciplinas e professores, como ilustra na Figura 16. A tela apresenta o calendário de aulas montado, representado por uma tabela com as colunas contendo o dia, horário, turma, disciplina, professor e ação, tornando possível visualizar a distribuição detalhada das aulas ao longo da semana.



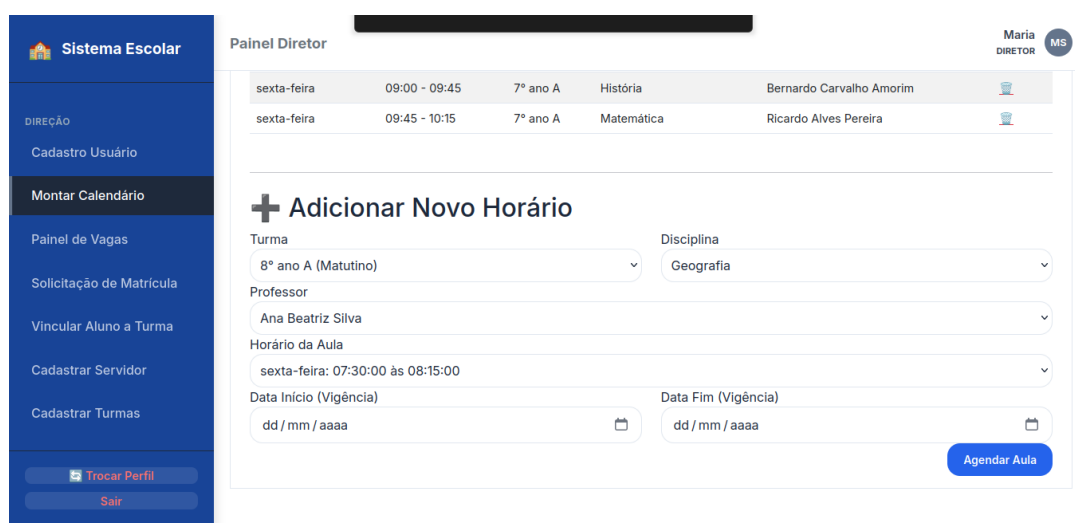
Sistema Escolar Painel Diretor Maria DIRETOR MS

Calendário de Aulas Montado

DIA	HORÁRIO	TURMA	DISCIPLINA	PROFESSOR	AÇÃO
segunda-feira	07:30 - 08:15	6º ano A	Língua Portuguesa	Pedro Andrade	
segunda-feira	07:30 - 08:15	7º ano A	Língua Portuguesa	Pedro Andrade	
segunda-feira	07:30 - 08:15	8º ano A	Língua Portuguesa	Pedro Andrade	
segunda-feira	08:15 - 09:00	7º ano A	Matemática	Ricardo Alves Pereira	
segunda-feira	09:45 - 10:15	7º ano A	Geografia	Ana Beatriz Silva	
terça-feira	08:15 - 09:00	6º ano A	Matemática	Ricardo Alves Pereira	
quarta-feira	09:00 - 09:45	6º ano A	Geografia	Ana Beatriz Silva	
quarta-feira	09:00 - 09:45	7º ano A	Matemática	Ricardo Alves Pereira	
sexta-feira	09:00 - 09:45	7º ano A	História	Bernardo Carvalho Amorim	
sexta-feira	09:45 - 10:15	7º ano A	Matemática	Ricardo Alves Pereira	

Figura 16 – Calendário de Aulas Montado

Como uma continuação direta da figura anterior, a Figura 17 ilustra a parte do formulário para adicionar novos horários de aula, permitindo que o diretor selecione a turma, a disciplina, o professor e defina o horário do bloco, datas de início e fim de vigência daquele período letivo.



Sistema Escolar Painel Diretor Maria DIRETOR MS

+ Adicionar Novo Horário

Turma: 8º ano A (Matutino) Disciplina: Geografia

Professor: Ana Beatriz Silva

Horário da Aula: sexta-feira: 07:30:00 às 08:15:00

Data Início (Vigência): dd / mm / aaaa Data Fim (Vigência): dd / mm / aaaa

Agendar Aula

Figura 17 – Adicionar Novo Horário ao Calendário

3.3.3 Planejamento de Vagas e Solicitações de Matrícula

A Figura 18 apresenta o painel de planejamento de vagas, funcionalidade desenvolvida para apoiar o diretor no acompanhamento e gerenciamento das solicitações de matrícula da unidade escolar. Nesta tela, o sistema exibe informações sobre a demanda de matrículas, permitindo o diretor analisar a quantidade de novos alunos e de rematrículas por série antes de tomar decisões relacionadas à ocupação das vagas disponíveis. Além disso, o painel disponibiliza um resumo das solicitações recebidas e o botão “Ver Lista”, que direciona para a tela de gerenciamento das solicitações.

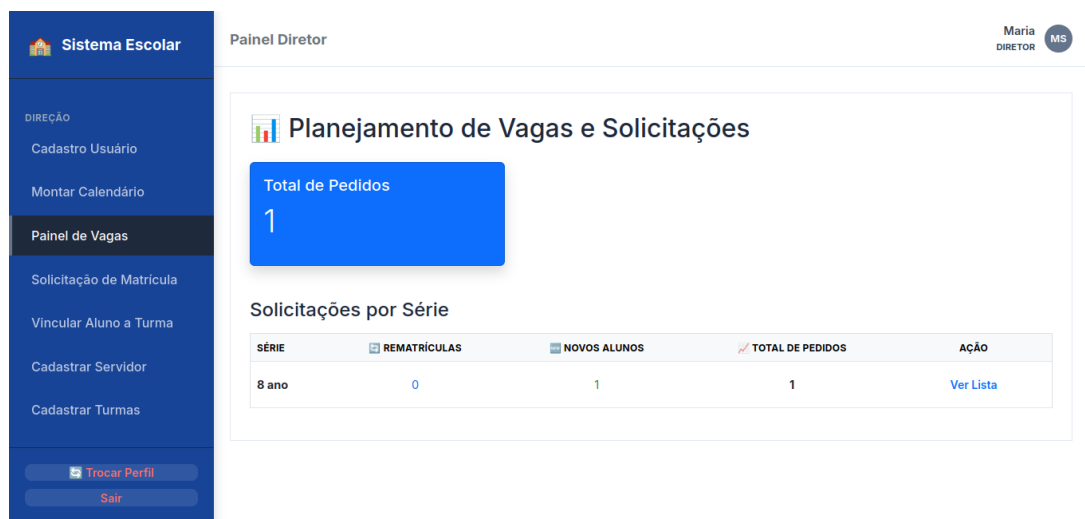


Figura 18 – Painel de Planejamento de Vagas

A partir desse painel, o diretor pode acessar a lista completa das solicitações por meio do botão “Ver Lista”, sendo direcionado para a tela apresentada na Figura 19. Nessa tela são exibidos todos os candidatos que solicitaram matrícula, juntamente com informações como endereço de *email*, situação da solicitação e as ações administrativas disponíveis. Dessa forma, o diretor pode analisar cada pedido individualmente e decidir entre aceitar a solicitação, notificar o responsável, atualizar os dados cadastrados ou finalizar o processo de matrícula.



Figura 19 – Gerenciamento de Solicitações de Vaga

Quando a solicitação é aprovada por meio da opção “Aceitar” e “Notificar”, o sistema envia automaticamente uma mensagem eletrônica ao aluno ou ao seu responsável, conforme ilustrado na Figura 20. Esse *email* informa que a solicitação foi aceita e orienta o comparecimento à unidade escolar na data estabelecida, acompanhado da documentação necessária para a efetivação da matrícula. Essa etapa integra o fluxo de atendimento do sistema, permitindo que

a comunicação entre a escola e a família seja realizada de forma automatizada, agilizando o processo de matrícula.

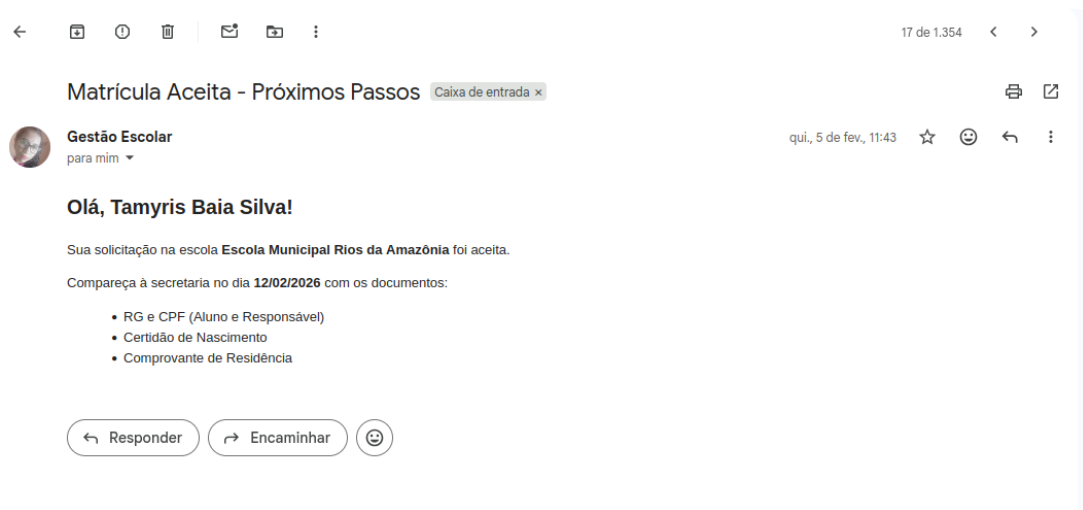


Figura 20 – Mensagem de matrícula aceita

Após o recebimento da mensagem, os responsáveis devem comparecer à escola para apresentar a documentação exigida e concluir o cadastro do estudante. A efetivação da matrícula é realizada pela secretaria da unidade escolar em cinco etapas sequenciais, apresentadas nas Figuras 21, 22, 23, 24 e 25.

A imagem mostra a primeira etapa de um formulário web. No topo, há uma barra de progresso com cinco etapas numeradas. A primeira etapa, "Dados do Aluno", está selecionada. O formulário contém campos para: Nome Completo (Guilherme Sampaio Alves), Data de Nascimento (22/05/2015), Telefone (91985794613), E-mail (guilherme@gmail.com), Gênero (Masculino), Nacionalidade (Brasil) e Tipo Sanguíneo (O+). Um botão "Próximo" está no canto inferior direito.

Figura 21 – Etapa 1: Dados do Aluno

A imagem mostra a segunda etapa do formulário web. A segunda etapa, "Documentação Civil", está selecionada. O formulário contém campos para: CPF (023.789.456-24), RG (12.345.67), Órgão Emissor (SSP), UF Emissor (PA), Data Emissão (24/10/2016), NIS (Bolsa Família) (58841145566) e Número e Série da Certidão de Nascimento (4477741). Botões "Voltar" e "Próximo" estão no canto inferior.

Figura 22 – Etapa 2: Documentação Civil

Figura 23 – Etapa 3: Responsáveis

Figura 24 – Etapa 4: Localização e Vínculo

Figura 25 – Etapa 5: Upload de Documentos

O Cadastro de Matrícula está dividido em cinco etapas. A etapa 1, coletam-se as informações básicas do estudante, incluindo nome completo, data de nascimento, telefone, *email*, gênero, nacionalidade, naturalidade e tipo sanguíneo. Prosseguindo para a etapa 2, o sistema exige o preenchimento de documentos obrigatórios como cpf, rg, órgão emissor, uf, data de emissão, nis e o número da certidão de nascimento. Na etapa 3, realiza-se o registro do responsável principal, solicitando-se o nome, cpf, *email* e telefone para comunicações oficiais. A etapa 4, é dedicada aos dados de residência, como endereço completo, cep, uf, município, bairro e zona; além disso, nesta fase, registra-se a escola de origem e os detalhes da vaga pretendida, definindo a unidade escolar, a série e o turno desejados. Por fim, a etapa 5, encerra o cadastro com o envio das cópias digitais obrigatórias, organizadas em seis categorias específicas: rg ou certidão do aluno, cpf do aluno, comprovante de residência, histórico escolar ou declaração de transferência, documento de identificação do responsável e uma foto 3x4 para identificação.

3.3.4 Vinculação de Aluno a Turma

A Figura 26 apresenta a tela de alunos com *status* de “Enturmação Pendente”. Após a conclusão do processo de matrícula, o estudante é automaticamente inserido nessa lista, indicando que seu cadastro foi efetivado, mas que ainda aguarda a definição da turma pela direção da escola. Nessa tela são exibidas informações como o nome do aluno, número de matrícula, série pretendida e turno. Para iniciar o processo de alocação, o diretor deve selecionar a opção “Alocar na Turma”.

Sistema Escolar Painel Diretor Maria DIRETOR MS

Enturmação Pendente
Alunos com matrícula ativa aguardando alocação em sala de aula. TOTAL PENDENTE 1

IDENTIFICAÇÃO DO ALUNO	SÉRIE PRETENDIDA	PERÍODO	AÇÕES
T Tamyris Baia Silva Matricula: 2026-000006	7º ano	Manhã	Alocar na Turma

Trocar Perfil Sair

Figura 26 – Painel de Enturmação Pendente

Após essa seleção, o sistema exibe a tela de escolha da turma, ilustrada na Figura 27. Nessa tela, são apresentadas as turmas disponíveis ao ano letivo vigente, informa a quantidade de vagas ocupadas, capacidade total de cada turma e o botão “Vincular Turma”.

Sistema Escolar Painel Diretor Maria DIRETOR MS

Finalizar Matrícula: Seleção de Turma
Aluno: Tamyris Baia Silva
Escola: Escola Municipal Rios da Amazônia
Matrícula ID: 36 (Cód.: 2026-000006)

Turmas Disponíveis (Ano Letivo: 2026)

TURMA	SÉRIE	TURNO	CAPACIDADE (OCUPAÇÃO)	AÇÃO
6º ano A	6º ano	Matutino	1 / 35 (34 vagas)	Vincular Turma
7º ano A	7º ano	Matutino	3 / 35 (32 vagas)	Vincular Turma
8º ano A	8º ano	Matutino	1 / 40 (39 vagas)	Vincular Turma

Voltar

Trocar Perfil Sair

Figura 27 – Seleção de Turma

3.3.5 Gestão Administrativa: Servidores e Turmas

A Figura 28 apresenta a tela de vínculo de servidor, responsável por vincular os profissionais cadastrados às respectivas unidades escolares. Nessa tela, o diretor pode associar cada servidor ao seu cargo ou função, informando dados como unidade escolar, regime de trabalho, data de admissão e período de vigência do vínculo. Essa funcionalidade permite manter atualizado o quadro de servidores da instituição, facilitando a organização administrativa e o gerenciamento dos profissionais.

Figura 28 – Vínculo de Servidor à Unidade Escolar

A Figura 29 apresenta a tela de cadastro de turmas. Para criar uma nova turma, o diretor deve preencher um formulário contendo informações como nome da turma, ano letivo, capacidade máxima de alunos, etapa de ensino, turno, unidade escolar, sala de aula, disciplinas e professores responsáveis. Após a confirmação do cadastro, a turma passa a integrar a estrutura acadêmica da escola.

Figura 29 – Cadastro e Parâmetros de Turma

Após o cadastro, a turma pode ser visualizada na tela de gestão de turmas, apresentada na Figura 30. Nessa tela, o sistema centraliza todas as turmas cadastradas em uma tabela contendo informações como nome da turma, série e turno. Além disso, cada registro disponibiliza ações para gerenciamento da lotação, matrículas, vínculos entre disciplinas e professores, bem como a exclusão da turma quando necessário.

NOME DA TURMA	TURNO	SÉRIE	AÇÕES
8º ano A	Matutino	8 ano	Lotação Matrículas Vínculos Excluir
7º ano A	Matutino	7 ano	Lotação Matrículas Vínculos Excluir
6º ano A	Matutino	6 ano	Lotação Matrículas Vínculos Excluir
6º ano B	Vespertino	6 ano	Lotação Matrículas Vínculos Excluir
6º ano A	Matutino	6 ano	Lotação Matrículas Vínculos Excluir

Figura 30 – Tela de Gestão de Turmas

3.3.6 Lotação e Monitoramento Acadêmico

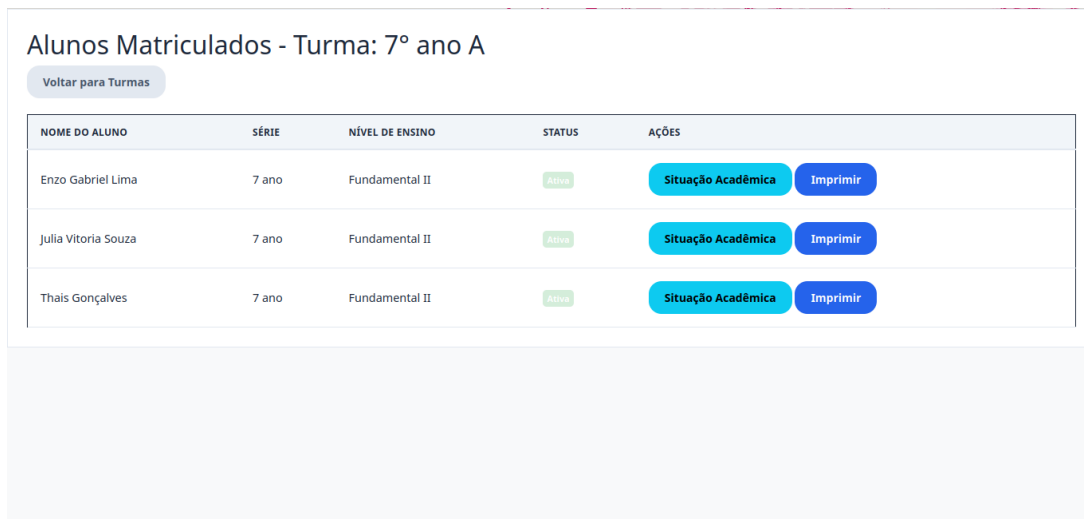
A Figura 31 apresenta a tela de lotação da turma. Nessa tela, o diretor pode acompanhar a distribuição das disciplinas vinculadas à turma, visualizando a carga horária de cada componente curricular e o respectivo professor responsável. Essa funcionalidade auxilia no planejamento da grade curricular e na organização das atividades pedagógicas da turma.

DISCIPLINA	CARGA HORÁRIA	PROFESSOR
Língua Portuguesa	200 horas	Pedro Andrade
Língua Portuguesa	200 horas	Pedro Andrade
Matemática	200 horas	Ricardo Alves Pereira
História	200 horas	Bernardo Carvalho Amorim
Geografia	150 horas	Ana Beatriz Silva

Figura 31 – Detalhamento da Lotação da Turma

A Figura 32 apresenta a tela de alunos matriculados por turma. Nessa tela, é exibida uma

tabela contendo a relação dos estudantes vinculados à turma, juntamente com informações como nível de ensino e situação da matrícula. Além da consulta dos dados, o sistema permite acessar a situação acadêmica individual de cada aluno ou emitir sua declaração de matrícula.



NOME DO ALUNO	SÉRIE	NÍVEL DE ENSINO	STATUS	AÇÕES
Enzo Gabriel Lima	7 ano	Fundamental II	Ativa	Situação Acadêmica Imprimir
Julia Vitoria Souza	7 ano	Fundamental II	Ativa	Situação Acadêmica Imprimir
Thais Gonçalves	7 ano	Fundamental II	Ativa	Situação Acadêmica Imprimir

Figura 32 – Listagem de Alunos Matriculados por Turma

A Figura 33 apresenta o modal de situação acadêmica do aluno. Essa janela exibe informações detalhadas da matrícula, incluindo data de ingresso, série e turno, além de permitir ao diretor alterar o status acadêmico do estudante. As alterações realizadas são registradas imediatamente no banco de dados, garantindo que as informações permaneçam sempre atualizadas.



Alunos Matriculados - Turma: 8º ano A

[Voltar para Turmas](#)

NOME DO ALUNO	SÉRIE
Mateus Oliveira Santos	8 ano

Situação Acadêmica

Nome: Mateus Oliveira Santos

Turma: 8º ano A

Série: 8 ano

Nível: Fundamental II

Turno: Manhã

Data Matrícula: 2026-01-31

Status Atual: ativa

Alterar Status:

Ativa

[Salvar Alteração](#) [Fechar](#)

Figura 33 – Modal de Situação Acadêmica

A Figura 34 apresenta a tela de gerenciamento de vínculos da turma. Essa funcionalidade permite associar disciplinas e professores às turmas já cadastradas, possibilitando a inclusão de novos vínculos sem alterar aqueles previamente existentes. Dessa forma, a composição da grade curricular pode ser ajustada sempre que necessário, preservando a integridade das informações já registradas.

Figura 34 – Gerenciamento de Vínculos da Turma

3.3.7 Recursos Humanos e Fluxo de Transferência

As funcionalidades finais do módulo do Diretor abrangem o controle da frequência dos professores e o gerenciamento das solicitações de transferência entre unidades escolares. A Figura 35 apresenta a tela de controle de frequência dos professores, dividida entre as seções “Novo Registro” e “Histórico de Faltas”. Nessa tela, é possível registrar ausências, informar justificativas, anexar atestados médicos em formato PDF e identificar o responsável pelo lançamento da ocorrência, proporcionando maior controle sobre a assiduidade dos profissionais.

Figura 35 – Controle de Frequência de Professores

A Figura 36 apresenta o painel de transferências pendentes. Nessa tela, o diretor pode analisar as solicitações de transferência de alunos provenientes de outras escolas, visualizando informações como unidade de origem e unidade de destino. Após a análise, o gestor pode aprovar ou recusar a solicitação, permitindo que o processo seja conduzido de forma organizada e registrada no sistema.

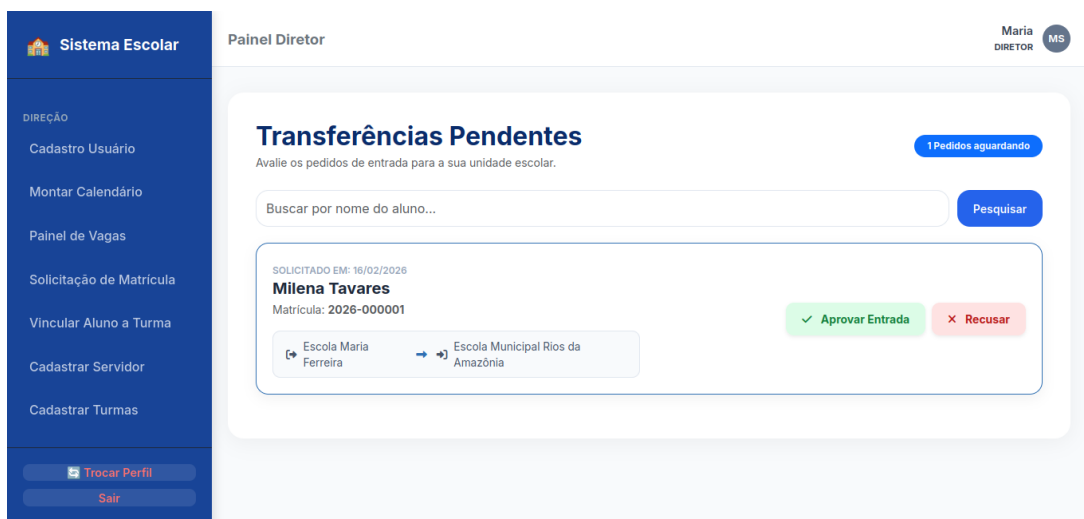


Figura 36 – Tela de Transferências Pendentes

A Figura 37 apresenta a tela de relatório consolidado de servidores. Nessa tela, o sistema reúne todos os profissionais vinculados à unidade escolar em uma única tabela, contendo informações como cargo, contatos, situação do vínculo e data de admissão. Além da consulta dos dados, o relatório pode ser exportado para impressão, servindo como ferramenta de apoio às atividades de gestão e auditoria administrativa.



Figura 37 – Relatório Consolidado de Servidores

3.4 Área do Professor

A área do professor foi desenvolvida com o objetivo de centralizar a gestão pedagógica, permitindo o controle de turmas, registros de notas, frequência e acompanhamento do calendário letivo. A interface é intuitiva e organizada para otimizar a rotina docente.

3.4.1 Paineis de Gestão de Turmas

A primeira tela de interação do professor é a aba “Minhas Turmas”. Esta área funciona como um agregador de informações, exibindo as turmas vinculadas ao perfil do professor para o ano letivo ativo. A Figura 38 apresenta a tela de seleção de turmas do professor, o sistema utiliza um mecanismo de cards para cada turma, exibindo a disciplina ministrada e o *status* da gestão de assiduidade. Através desta tela, o professor tem acesso rápido aos botões de ação para os módulos de notas e frequência.

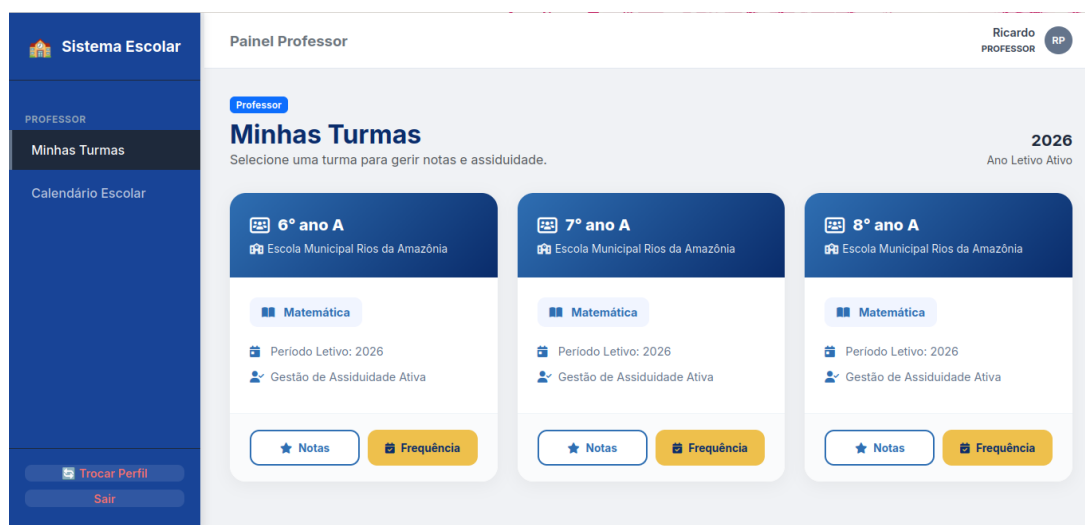


Figura 38 – Tela Minhas Turmas

3.4.2 Lançamento de Notas e Médias

O sistema oferece uma tela automatizada para o registro de avaliações, capaz de processar cálculos de médias em tempo real e sinalizar o *status* de aprovação de cada discente. Na Figura 39, é ilustrada a funcionalidade de lançamento de notas de uma disciplina. A tela apresenta uma tabela com a matrícula e o nome do aluno, permitindo a inserção de quatro avaliações. O sistema utiliza um mecanismo de validação que calcula automaticamente a média e exibe na tela. Há também um campo específico para notas de Recuperação, garantindo o controle completo do histórico avaliativo.

Sistema Escolar

Painel Professor

LANÇAR NOTAS

Professor: Ricardo Alves Pereira
 Ano Letivo: 2026
 Escola: Escola Municipal Rios da Amazônia
 Turma: 7º ano A
 Disciplina: Matemática - Horário da Aula: 08:15:00 às 09:00:00

MATRICULA	ALUNO	1ª AVALIAÇÃO	2ª AVALIAÇÃO	3ª AVALIAÇÃO	4ª AVALIAÇÃO	MÉDIA	RECUPERAÇÃO
000022	Enzo Gabriel Lima	10	9,5	8	7,5	8,75 Aprovado	
000026	Julia Vitoria Souza	5	4,5	4,5	5	4,75 Aprovado (Rec.)	10
000001	Thais Gonçalves	10	10	9,5	8,5	9,50 Aprovado	

* A média será calculada automaticamente e será registrada no campo 'Média' ao clicar em **Confirmar**.

Concluir Lançamento

Figura 39 – Tela da Funcionalidade Lançar Notas

3.4.3 Controle de Frequência Diária

A gestão da assiduidade é realizada através de um diário de classe digital, que permite o registro pontual e a visualização estatística da presença dos alunos. A Figura 40 apresenta o mecanismo de controle de frequência, utilizado pelo professor. O sistema disponibiliza um seletor de datas e uma listagem da semana de segunda-feira a sexta-feira, onde o docente pode marcar a presença através de *checkboxes*. Um diferencial desta funcionalidade é a exibição da porcentagem de presença acumulada e o campo para inserção de justificativas da semana, facilitando o acompanhamento pedagógico individualizado.

Sistema Escolar

Painel Professor

FREQUÊNCIA

Professor: Pedro Andrade
 Ano Letivo: 2026
 Escola: Escola Municipal Rios da Amazônia
 Turma: 7º ano A
 Disciplina: Língua Portuguesa - Horário: 07:30:00 às 08:15:00

Buscar Período (Selecionar Dia)
 16 / 02 / 2026

MATRICULA	ALUNO	SEG (16/02)	TER (17/02)	QUA (18/02)	QUI (19/02)	SEX (20/02)	JUSTIFICATIVAS DA SEMANA
000022	Enzo Gabriel Lima (Presença: 100%)	☒	☐	☐	☐	☐	16/02: Justificativa...
000026	Julia Vitoria Souza (Presença: 100%)	☒	☐	☐	☐	☐	16/02: Justificativa...
000001	Thais Gonçalves (Presença: 100%)	☒	☐	☐	☐	☐	16/02: Justificativa...

Registrar Frequência

Figura 40 – Tela de Registro de Frequência

3.4.4 Calendário Escolar e Prazos Administrativos

Para garantir o cumprimento dos prazos definidos pela Secretaria de Educação, a plataforma disponibiliza um cronograma detalhado com os períodos de início e término das atividades.

A Figura 41 apresenta o Calendário Escolar para o ano letivo. A tela é dividida entre o período de aulas e a tabela de prazos de lançamento de notas. Este componente é essencial para a organização professor, pois delimita as janelas temporais permitidas para o envio das notas de cada avaliação e do período de recuperação final.

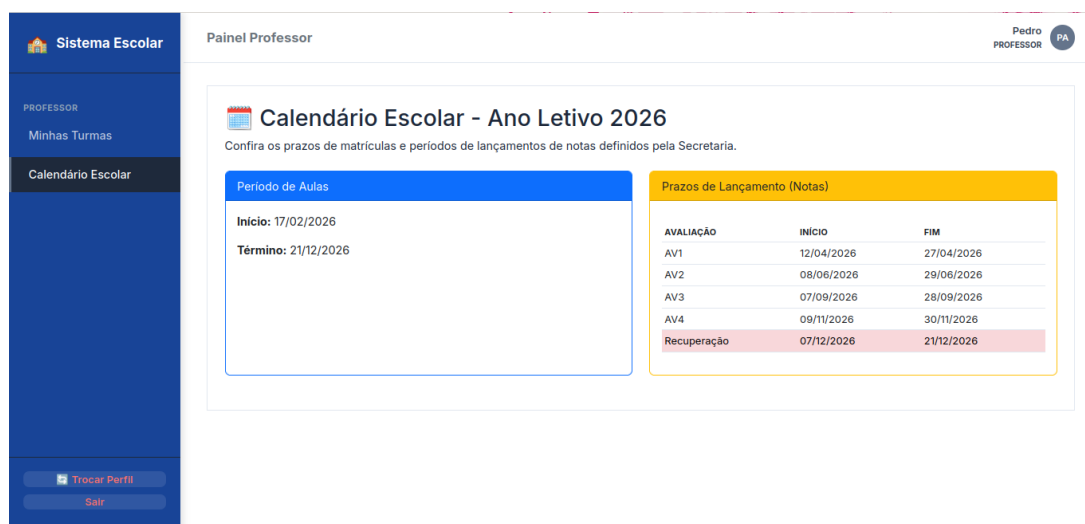


Figura 41 – Calendário Escolar e Prazos de Lançamento

3.5 Área do Aluno

A área do aluno foi concebida para fornecer ao discente total autonomia e transparência sobre sua vida escolar. Através de uma *interface* responsiva, o sistema centraliza o acompanhamento de frequência, consulta de desempenho, gestão de solicitações matrícula e transferência e geração de documentos oficiais.

3.5.1 Painel de Controle e Acompanhamento Diário

A Figura 42 apresenta a tela inicial do Portal do Aluno, que funciona como uma central de informações acadêmicas. Ao acessar o sistema, o estudante é recebido por uma tela de boas-vindas que reúne, em um único ambiente, as principais informações relacionadas à sua vida escolar. Nessa tela, é possível consultar a situação da matrícula, a frequência por disciplina, o desempenho nas avaliações e os dados cadastrais. O sistema também disponibiliza o calendário escolar, com as datas do ano letivo e dos lançamentos de notas, além de oferecer acesso rápido ao histórico de boletins.

Sistema Escolar

Painel Aluno

Thais ALUNO TG

Olá, Thais!
Bem-vindo ao seu portal escolar.

Dados Atuais

Escola
Escola Municipal Rios da Amazônia

Turma / Série (2026)
7º ano - 7º ano A

ATIVA

Frequência por Disciplina

DISCIPLINA	AULAS	FALTAS	% PRESENÇA
Geografia	1	0	100.0%
História	2	0	100.0%
Língua Portuguesa	1	0	100.0%
Matemática	3	0	100.0%

Calendário Escolar 2026

Aulas: 17/02/2026 a 21/12/2026

Avaliação 1: 12/04/2026 - 27/04/2026

Avaliação 2: 08/06/2026 - 29/06/2026

Avaliação 3: 07/09/2026 - 28/09/2026

Documentos Escolares

BOLETINS ANTERIORES

2026

HISTÓRICO ESCOLAR PDF

Ano ▼ Gerar

Figura 42 – Portal do Aluno

3.5.2 Gestão de Matrículas e Séries

Para automatizar processos de matrículas que antes eram manuais, o sistema implementa um módulo de autoatendimento para solicitações de matrícula. Na Figura 43, é apresentado o formulário para solicitar nova matrícula, permitindo que o aluno selecione o ano letivo de destino. A tela utiliza componentes de seleção dinâmica para a escola e campos de texto para a série desejada, garantindo que os dados sejam validados e enviados diretamente para a análise da secretaria escolar.

Sistema Escolar

Painel Aluno

Thais ALUNO TG

Solicitar Nova Matrícula

Ano Letivo de Destino: 2026

Selecionar a Escola:

Clique para selecionar...

Série Desejada:

Ex: 1ª Série

Voltar

Confirmar Solicitação

Figura 43 – Fluxo de Solicitação de Nova Matrícula

3.5.3 Processo de Transferência Entre Unidades

Visando a mobilidade estudantil, o sistema oferece um mecanismo seguro para a migração de dados escolares entre diferentes instituições da rede. A Figura 44 ilustra a funcionalidade de transferência, que exige que o aluno selecione sua matrícula ativa e a unidade de destino.

Conforme descrito na seção de “Informação Importante” da tela, o sistema implementa uma funcionalidade onde os dados acadêmicos são migrados automaticamente apenas após a aprovação da escola de destino, assegurando a integridade do histórico do aluno.

Figura 44 – Tela de Solicitação de Transferência

3.5.4 Visualização de Boletim

O módulo de notas e boletins permite o acompanhamento detalhado de cada avaliação realizada ao longo dos bimestres. Na Figura 45, apresenta o boletim com as notas do aluno. O sistema organiza as disciplinas em uma tabela que exibe as notas das quatro avaliações, a média parcial e a situação final. Caso as notas ainda não tenham sido lançadas pelo professor, o sistema exibe o status “Aguardando”, mantendo o aluno informado sobre o andamento pedagógico.

DISCIPLINA	AV1	AV2	AV3	AV4	MÉDIA PARCIAL	RECUPERAÇÃO	MÉDIA FINAL	SITUAÇÃO
Língua Portuguesa	10	9,5	10	9,5	9,8	-	9,8	Aprovado
Matemática	10	10	9,5	8,5	10,0	-	10,0	Aprovado
História	-	-	-	-	0,0	-	0,0	Aguardando
Geografia	-	-	-	-	0,0	-	0,0	Aguardando

Figura 45 – Tela do Boletim Escolar Digital

3.5.5 Emissão de Documentos Oficiais de Histórico

Como resultado da consolidação de todos os dados inseridos pelos professores e pela secretaria, o sistema é capaz de gerar documentos em formato pdf para uso legal. A Figura 46 apresenta o Histórico Escolar gerado pelo sistema. O documento consolida a trajetória do aluno, listando as médias e a carga horária por área de conhecimento. No rodapé, o sistema insere automaticamente o certificado de regularidade de matrícula para o ano vigente, validando as informações conforme as normas vigentes.

 PREFEITURA MUNICIPAL SECRETARIA MUNICIPAL DE EDUCAÇÃO DIRETORIA DE ENSINO HISTÓRICO ESCOLAR 																	
ENDEREÇO Rua E. 50						CNPJ 45600000						TELEFONE 91955554444					
NOME DO ALUNO (A) Thais Gonçalves												DATA DE NASCIMENTO 04/08/2014					
NACIONALIDADE Brasil						RG 1110223						CPF 02311122203					
NOTAS E / OU CONCEITOS																	
ÁREA DE CONHECIMENTO	1º ANO		2º ANO		3º ANO		4º ANO		5º ANO		6º ANO		7º ANO		8º ANO		
	MEDIA	CH	MEDIA	CH	MEDIA	CH	MEDIA	CH	MEDIA	CH	MEDIA	CH	MEDIA	CH	MEDIA	CH	
Língua Portuguesa											3.375	200	9.75	200			
Arte																	
Educação Física																	
História																	
Geografia																	
Ensino Religioso																	
Ciências Naturais																	
Matemática													10	200			
Ling. Estrangeira																	
Faltas																	
Resultado Final																	
ESTUDOS REALIZADOS																	
ANO	ANO LETIVO		ESTABELECIMENTO DE ENSINO										MUNICÍPIO		UF		
1º																	
2º																	
3º																	
4º																	
5º																	
6º	2025		Escola Municipal Rios da Amazônia										Belém		Pará		
7º	2026		Escola Municipal Rios da Amazônia										Belém		Pará		
8º																	

Figura 46 – Histórico Escolar Gerado pelo Sistema

4 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o projeto, desenvolvimento e resultado da construção de um *software* para gestão escolar municipal. Para isso, foi adotado o modelo em cascata e o padrão de projeto *MVC*, garantindo o cumprimento de boas práticas de programação.

A partir do desenvolvimento deste trabalho, é possível concluir que é inegável a necessidade de uma ferramenta de *software* para prover suporte à gestão escolar. Deste modo, garantindo maior organização e melhor acompanhamento das atividades educacionais. Nesse contexto, a proposição da adoção da tecnologia da informação tornou-se de suma importância para a adequação das instituições de ensino no contexto do ano de 2026.

É possível concluir que a implementação da solução proposta poderá modernizar os processos administrativos escolares, além de proporcionar maior segurança, controle e acessibilidade às informações.

O estudo realizado mostrou que os desafios relacionados à administração escolar vão além do simples armazenamento de dados, exigindo mecanismos que possibilitem organização, controle e facilidade no acesso às informações. Assim, por meio do levantamento de requisitos, da modelagem do banco de dados e da implementação do sistema, foi possível compreender as necessidades envolvidas nesse processo e desenvolver uma ferramenta robusta e alinhada às demandas da gestão educacional.

O processo de desenvolvimento do *software* contemplou todas as etapas propostas, desde a análise do problema até a implementação da solução. Podemos destacar que o sistema desenvolvido não apenas auxilia as atividades administrativas da escola, mas também contribui para a melhoria da comunicação, do acompanhamento acadêmico e da organização das informações, tornando os processos mais eficientes e reduzindo a dependência de controles manuais.

Como trabalhos futuros, propõe-se a realização de testes em ambiente real de utilização, bem como a implementação de novas funcionalidades, como integração com sistemas governamentais da área da educação, emissão automática de documentos escolares e envio de notificações aos usuários. A partir dos resultados obtidos nessas futuras implementações, o sistema poderá ser ampliado e adaptado para atender um número ainda maior de instituições de ensino, contribuindo para a modernização da gestão educacional.

REFERÊNCIAS

- ABREU, F. M. de; QUARESMA, E. R.; OLIVEIRA, J. J. R. de; CABRAL, Y. R.; JESUS, P. C. A. de; NASCIMENTO, R. da C.; CABRAL, D. F.; SANTOS, G. N. de L.; SILVA, S. D.; CARVALHEDO, A. dos R. Desenvolvimento de sistemas web para gestão escolar: uma análise sobre funcionalidade, acessibilidade e impacto social. **Revista Ibero-Americana de Humanidades, Ciências e Educação**, v. 11, n. 7, p. 2997–3005, 2025.
- BOOTSTRAP. **Bootstrap Documentation**. 2026. Disponível em: <<https://getbootstrap.com/docs/5.3/about/overview/>>. Acesso em: 03 jun. 2026.
- CANDINHO, A. A. M.; SCHMITZ, H. U. B.; MEDEIROS, M. de; SILVA, P. P.; GABRIEL, S. G. Uso de ferramentas digitais na gestão escolar. **Ibero-Americana De Humanidades, Ciências E Educação**, v. 11, n. 1, p. 592–608, 2025.
- CGIB. **Pesquisa sobre o uso das tecnologias de informação e comunicação no setor público brasileiro: TIC Governo Eletrônico 2023**. 2024. Disponível em: <https://www.cgi.br/media/docs/publicacoes/2/20241104103247/tic_governo_eletronico_2023_livro_eletronico.pdf>. Acesso em: 18 mai. 2026.
- CHACON, S.; STRAUB, B. **Pro Git**. 2. ed. New York: Apress, 2014.
- DEITEL, P.; DEITEL, H. **Internet e World Wide Web: como programar**. 5. ed. São Paulo: Pearson Education do Brasil, 2010.
- DUCKETT, J. **HTML e CSS: projete e construa websites**. Rio de Janeiro: Alta Books, 2016.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 7. ed. São Paulo: Pearson Education do Brasil, 2018.
- ENAP. **Caminhos da inovação no setor público**. Brasília: Escola Nacional de Administração Pública, 2022. Disponível em: <<http://repositorio.enap.gov.br/handle/1/7420>>. Acesso em: 11 mai. 2026.
- FLANAGAN, D. **JavaScript: o guia definitivo**. 7. ed. Porto Alegre: Bookman, 2020.
- FOWLER, M. **Padrões de Arquitetura de Aplicações Corporativas**. Porto Alegre: Bookman, 2006.
- FREEMAN, E. **Use a Cabeça! HTML e CSS**. 2. ed. Rio de Janeiro: Alta Books, 2019.
- GIOTTO, E.; CARDOSO, C. D. V. Desenvolvimento e aplicabilidade de um sistema de gestão escolar municipal. **Revista de Ciências Humanas**, v. 2, n. 2, p. 113–134, 2012.
- GITHUB. **GitHub Docs**. 2026. Disponível em: <<https://docs.github.com>>. Acesso em: 08 jun. 2026.
- LAUDON, K. C.; LAUDON, J. P. **Sistemas de Informação Gerenciais**. 17. ed. São Paulo: Pearson, 2021.
- LIU, T.; LUO, Y. T.; PANG, P. C.-I.; KAN, H. Y. Exploring the impact of information and communication technology on educational administration: A systematic scoping review. **Education Sciences**, v. 15, n. 9, p. 1114, 2025.

MAYRER, M. **Desenvolvimento de software web para gerenciamento de dados e documentos de estudantes**. Trabalho de Conclusão de Curso (Graduação) — Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Rio Grande do Sul, 2025.

Microsoft. **Visual Studio Code Documentation**. 2026. Disponível em: <<https://learn.microsoft.com/en-us/visualstudio/windows/?view=visualstudio>>. Acesso em: 06 jun. 2026.

MORAN, J. M. **A educação que desejamos: novos desafios e como chegar lá**. Campinas: Papirus, 2015.

NASCIMENTO, L. **PHP moderno: boas práticas e desenvolvimento profissional**. São Paulo: Casa do Código, 2021.

OLIVEIRA, D. G. de. **Sistema Mineiro de Administração Escolar (SIMADE): avaliação do impacto e da percepção dos usuários operacionais sobre os desafios na gestão das escolas estaduais de Minas Gerais**. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, 2023.

PHP. **PHP Manual**. 2025. Disponível em: <<https://www.php.net/docs.php>>. Acesso em: 22 jun. 2026.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software: uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2016.

SARAVANOS, A.; CURINGA, M. X. Simulating the software development lifecycle: The waterfall model. **Applied System Innovation**, v. 6, n. 6, p. 108, 2023.

SCHWONKE, C. H. E. **Análise do Modelo de Gestão Escolar e dos Benefícios da Utilização do Sistema de Informação ISE (Informatização da Secretaria de Educação)**. Trabalho de Conclusão de Curso (Graduação) — Universidade Federal de Pelotas, Pelotas, 2025.

SHAH, M. Impact of management information systems (mis) on school administration: What the literature says. **Procedia - Social and Behavioral Sciences**, v. 116, p. 2799–2804, 2014.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 7. ed. Rio de Janeiro: Elsevier, 2020.

SOMMERVILLE, I. **Engenharia de Software**. 10. ed. São Paulo: Pearson, 2019.

SOUZA, E. N. de; ROCHA, A. B. A.; BARBOSA, L. S. de O.; OLIVEIRA, R. B. de. Informática na educação: a gestão escolar e as fases de implementação de um projeto de tecnologia educacional. **RECIMA21 - Revista Científica Multidisciplinar**, v. 4, n. 4, p. 1–12, 2023. Disponível em: <<https://doi.org/10.47820/recima21.v4i4.3017>>.

UNESCO. **Gestão escolar e tecnologia na educação do Brasil**. 2025. Disponível em: <<https://www.unesco.org/pt/node/99503>>. Acesso em: 18 mai. 2026.

WELLING, L.; THOMSON, L. **PHP e MySQL: desenvolvimento web**. 5. ed. Rio de Janeiro: Elsevier, 2017.

Apêndices

APÊNDICE A – DOCUMENTAÇÃO DE REQUISITOS

A.1 Requisitos Funcionais

Apresentação dos requisitos funcionais que descrevem as ações que o sistema deve ser capaz de executar. A Tabela 1 apresenta os requisitos essenciais implementados no sistema.

Tabela 2 – Requisitos Funcionais

ID	DESCRIÇÃO
1	O sistema deve permitir o primeiro acesso e cadastro inicial do Secretário de Educação, com perfil de administrador.
2	O Secretário pode cadastrar outros usuários com privilégios administrativos equivalentes.
3	Super usuários podem cadastrar escolas e associar um diretor a cada instituição.
4	Diretores devem poder cadastrar secretários escolares responsáveis pelo gerenciamento local.
5	Secretários escolares devem cadastrar professores e configurar turmas com: nome, ano letivo, nível de ensino, série, turno, e limites de alunos.
6	O sistema deve permitir o cadastro de alunos com: nome, <i>email</i> , endereço, responsáveis, documentos pessoais. Deve gerar automaticamente um número único de matrícula por ano e associá-lo a uma ou mais turmas.
7	O sistema deve impedir a matrícula duplicada de um aluno em escolas diferentes simultaneamente, oferecendo opção de solicitação de transferência.
8	A escola de origem pode aceitar a transferência, migrando o histórico escolar do aluno.
9	Alunos ou responsáveis devem acessar boletins e histórico escolar com uso de CPF e <i>email</i> previamente cadastrados.
10	O sistema deve permitir consulta do histórico escolar por ano e deve estar disponível para exportação em formato PDF por responsáveis ou pela secretaria.
11	O sistema deve gerar relatórios filtráveis por escola, turma, professor ou município.
12	Cada aluno e turma deve estar vinculado corretamente ao nível de ensino (fundamental menor, maior ou médio).
13	O sistema deve permitir a atualização de endereço e contato de alunos e responsáveis.
14	O sistema deve permitir o cadastro de usuários com nome, <i>email</i> , senha, categoria, dados pessoais e também deve permitir que o usuário tenha mais de uma categoria.
15	O sistema deve permitir que o usuário efetue login na plataforma e exibir mensagens de erro em caso de falha no login, assim como, possa recuperar senha via <i>email</i> .
16	exibir mensagens de erro em caso de falha no login, assim como, possa recuperar senha via <i>email</i> .
17	O sistema deve permitir diferenciar os perfis de acesso (aluno, professor, diretor, secretaria).

A.2 Requisitos Não Funcionais

Apresentação dos requisitos funcionais que descrevem as propriedades do sistema e restrições. A Tabela 2 detalha os critérios de qualidade estabelecidos.

Tabela 3 – Requisitos Não Funcionais

ID	DESCRIÇÃO
1	O sistema deve ter a <i>interface</i> leve, simples e intuitiva, com foco na usabilidade.
2	O sistema deve ser desenvolvido em arquitetura MVC utilizando PHP, HTML, CSS e MySQL.
3	O sistema deve permitir a autenticação segura com criptografia de senhas.
4	O sistema deve garantir que apenas usuários autorizados possam acessar determinadas funções.
5	O sistema deve permitir a validação dos dados antes de persistência no banco.
6	O sistema deve garantir acesso adequado por dispositivos móveis e <i>desktop</i> .
7	O sistema deve permitir navegação simples e clara para todos os tipos de usuário.
8	O sistema deve mandar mensagens de erro e confirmação devem ser claras e instrutivas.
9	O sistema deve suportar múltiplos acessos simultâneos.

APÊNDICE B – PROJETO DE BANCO DE DADOS

Este tópico detalha a arquitetura de dados do Sistema de Gestão Escolar, fundamentada em metodologias de modelagem que definem como as informações são estruturadas e armazenadas. A elaboração deste projeto concentrou-se na definição do Modelo Lógico, que reflete a implementação prática das entidades e seus relacionamentos no sistema. Para a persistência das informações, utilizou-se o banco de dados relacional, tendo como sistema de gerenciamento o *MySQL*.

O esquema visual dessa organização pode ser observado na Figura 47, que ilustra o modelo lógico de banco de dados da plataforma. Ao todo, a estrutura relacional é composta pelas seguintes tabelas: *calendario_aula*, *calendario_gestao*, *cargo*, *disciplinas*, *disciplinas_has_matricula*, *disciplinas_has_turma*, *escola*, *estado*, *falta_professor*, *frequencia_alunos*, *historico*, *horario_aula*, *matricula*, *matricula_has_turma*, *municipios*, *nivel*, *notas_alunos*, *password_reset_tokens*, *responsavel*, *responsavel_has_usuario*, *servidor_vinculo*, *tipo*, *transferencia*, *turma*, *usuario* e *usuario_has_tipo*. O detalhamento técnico de cada tabela e seus respectivos atributos será apresentado nos tópicos seguintes.

B.1 Modelo Lógico

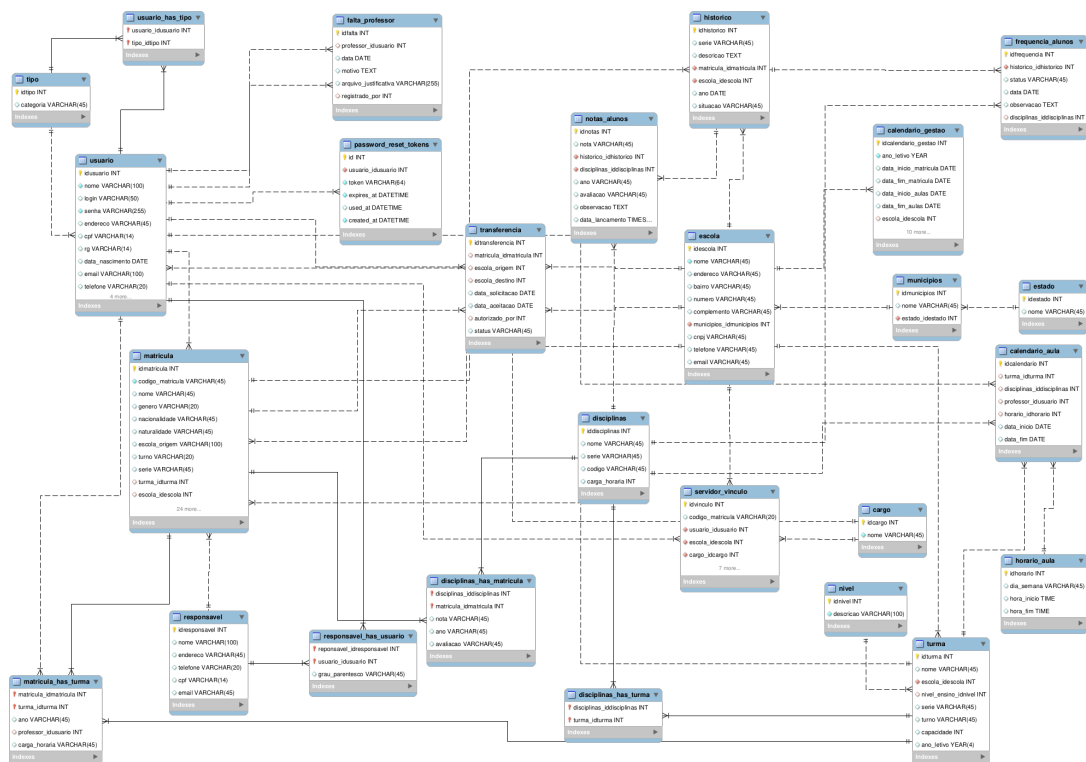


Figura 47 – Representação estrutural de banco de dados.

B.2 Tabelas de Banco de Dados

B.2.1 Tabela usuario

A tabela usuario é responsável por armazenar os dados de cada usuário cadastrado no sistema. Possui os seguintes campos:

- **idusuario:** Chave primária, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro na tabela.
- **nome:** Campo do tipo *varchar*, aceita até 100 caracteres como entrada e não pode ser nulo. Armazena o nome completo do usuário.
- **login:** Campo do tipo *varchar*, aceita até 50 caracteres como entrada e não pode ser nulo. Armazena o identificador único para acesso ao sistema.
- **senha:** Campo do tipo *varchar*, aceita até 255 caracteres como entrada e não pode ser nulo. Armazena a senha criptografada do usuário.
- **endereco:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o endereço residencial do usuário.
- **cpf:** Campo do tipo *varchar*, aceita até 14 caracteres como entrada e pode ser nulo. Armazena o Cadastro de Pessoa Física.
- **rg:** Campo do tipo *varchar*, aceita até 14 caracteres como entrada e pode ser nulo. Armazena o Registro Geral (Identidade).
- **data_nascimento:** Campo do tipo *date* e pode ser nulo. Armazena a data de nascimento do usuário.
- **email:** Campo do tipo *varchar*, aceita até 100 caracteres como entrada e pode ser nulo. Armazena o *email* do usuário.
- **telefone:** Campo do tipo *varchar*, aceita até 20 caracteres como entrada e pode ser nulo. Armazena o número de contato telefônico.
- **tipo_idtipo:** Chave estrangeira da tabela tipo, do tipo Inteiro e pode ser nula. Identifica o perfil de acesso do usuário (ex: aluno, adm, professor).

- **nis:** Campo do tipo *varchar*, aceita até 20 caracteres como entrada e pode ser nulo. Armazena o Número de Identificação Social.
- **tipo_sanguineo:** Campo do tipo *varchar*, aceita até 5 caracteres como entrada e pode ser nulo. Armazena a tipagem sanguínea para registros de saúde.
- **cargo_idcargo:** Chave estrangeira da tabela cargo, do tipo Inteiro e pode ser nula. Identifica o cargo ocupado pelo usuário na instituição.

B.2.2 Tabela usuario_has_tipo

A tabela usuario_has_tipo é responsável por associar os usuários aos seus respectivos tipos de perfis de permissão. Possui os seguintes campos:

- **usuario_idusuario:** Chave estrangeira da tabela usuario, do tipo Inteiro e não nula. Armazena a identificação do usuario referente a tabela usuario.
- **tipo_idtipo:** Chave estrangeira da tabela tipo, do tipo Inteiro e não nula. Armazena a identificação do tipo de usuário referente a tabela tipo.

B.2.3 Tabela tipo

A tabela tipo é responsável por definir os tipos de acesso de perfis de usuários. Possui os seguintes campos:

- **idtipo:** Chave primária, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro na tabela.
- **nome:** Campo do tipo *varchar*, aceita até 45 caracteres e não pode ser nulo. Armazena o nome do tipo de perfil de acesso.

B.2.4 Tabela matricula

A tabela matricula é a tabela central para o controle de alunos, armazenando desde os dados pessoais e escolar até o gerenciamento de documentos e o status da solicitação de vaga. Possui os seguintes campos:

- **idmatricula:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro.
- **codigo_matricula:** Campo do tipo *varchar*, aceita até 20 caracteres e possui restrição de valor único. Armazena o código identificador gerado pelo sistema para o aluno.

- **nome:** Campo do tipo *varchar*, aceita até 45 caracteres e pode ser nulo. Armazena o nome completo do aluno.
- **genero:** Campo do tipo *varchar*, aceita até 20 caracteres e pode ser nulo. Armazena o gênero do aluno.
- **nacionalidade:** Campo do tipo *varchar*, aceita até 45 caracteres e pode ser nulo. Indica o país de origem do aluno.
- **naturalidade:** Campo do tipo *varchar*, aceita até 45 caracteres e pode ser nulo. Indica a cidade e estado de nascimento.
- **escola_origem:** Campo do tipo *varchar*, aceita até 100 caracteres e pode ser nulo. Registra a última instituição de ensino onde o aluno estudou.
- **turno:** Campo do tipo *varchar*, aceita até 20 caracteres e pode ser nulo. Define o período de estudo.
- **serie:** Campo do tipo *varchar*, aceita até 45 caracteres e pode ser nulo. Indica o ano escolar para o qual o aluno solicita a vaga.
- **turma_idturma:** Chave estrangeira da tabela turma, do tipo Inteiro e pode ser nula. Vincula o aluno a uma turma específica após a ativação da matrícula.
- **escola_idescola:** Chave estrangeira da tabela escola, do tipo Inteiro e pode ser nula. Identifica a unidade escolar onde o aluno está vinculado.
- **descricao:** Campo do tipo *texto* pode ser nulo. Espaço para observações gerais sobre a matrícula.
- **usuario_idusuario:** Chave estrangeira da tabela usuario, do tipo Inteiro e pode ser nula. Associa a matrícula ao perfil de usuário do aluno no sistema.
- **responsavel_idresponsavel:** Chave estrangeira da tabela responsavel, do tipo Inteiro e pode ser nula. Vincula o aluno ao seu responsável legal.
- **status:** Campo do tipo *varchar*, aceita até 45 caracteres, não nulo e com valor padrão 'pendente'. Indica a situação atual da solicitação de matrícula.

- **ano_letivo:** Campo do tipo *year* e pode ser nulo. Define o ano letivo de vigência da matrícula.
- **data_matricula:** Campo do tipo *date* e pode ser nulo. Registra o dia em que a solicitação foi realizada.
- **data_aceitacao_diretor:** Campo do tipo *date* e pode ser nulo. Registra quando a matrícula foi aceita pela direção.
- **certidao_numero_serie:** Campo do tipo *varchar*, aceita até 50 caracteres e pode ser nulo. Armazena dados da certidão de nascimento do aluno.
- **rg_orgao_emissor:** Campo do tipo *varchar*, aceita até 10 caracteres e pode ser nulo. Armazena o órgão emissor do documento de identificação do aluno.
- **rg_uf_emissor:** Campo do tipo *varchar*, aceita até 2 caracteres e pode ser nulo. Armazena a UF do estado que o documento de identificação foi tirado.
- **rg_data_emissao:** Campo do tipo *date* e pode ser nulo. Indica a data de emissão que o documento de identificação.
- **bairro:** Campo do tipo *varchar*, aceita até 255 caracteres e pode ser nulo. Armazena o bairro do endereço residencial do aluno.
- **cep:** Campo do tipo *varchar*, aceita até 10 caracteres e pode ser nulo. Armazena o cep do endereço residencial do aluno.
- **uf_endereco:** Campo do tipo *varchar*, aceita até 2 caracteres e pode ser nulo. Armazena a UF do estado do endereço residencial do aluno.
- **municipio_endereco:** Campo do tipo *varchar*, aceita até 255 caracteres e pode ser nulo. Armazena o município do endereço residencial do aluno.
- **zona_endereco:** Campo do tipo *varchar*, aceita até 50 caracteres e pode ser nulo. Armazena a zona do endereço residencial do aluno.
- **doc_aluno_rg:** Campos do tipo *varchar*, aceita até 255 caracteres e pode ser nulo. Armazena o caminho (*path*) do arquivo rg digital enviado no servidor.

- **doc_aluno_cpf:** Campos do tipo *varchar*, aceita até 255 caracteres e pode ser nulo. Armazena o caminho (*path*) do arquivo cpf digital enviado no servidor.
- **doc_comprovante_residencia:** Campos do tipo *varchar*, aceita até 255 caracteres e pode ser nulo. Armazena o caminho (*path*) do arquivo comprovante de residência digital enviado no servidor.
- **doc_historico:** Campos do tipo *varchar*, aceita até 255 caracteres e pode ser nulo. Armazena o caminho (*path*) do arquivo histórico digital enviado no servidor.
- **doc_foto_3x4:** Campos do tipo *varchar*, aceita até 255 caracteres e pode ser nulo. Armazena o caminho (*path*) do arquivo foto 3x4 digital enviado no servidor.
- **doc_responsavel_rg_cpf:** Campos do tipo *varchar*, aceita até 255 caracteres e pode ser nulo. Armazena o caminho (*path*) do arquivo rg e cpf do responsável digital enviado no servidor.
- **tipo_solicitacao:** Campo do tipo *enum*('rematricula', 'nova') e pode ser nulo. Indica o tipo de solicitação de matrícula.
- **validade:** Campo do tipo *date* e pode ser nulo. Armazena a data de validade da matrícula.

B.2.5 Tabela escola

A tabela escola é responsável por armazenar os dados cadastrais de cada unidade de ensino no banco de dados e possui os seguintes campos:

- **idescola:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro na tabela.
- **nome:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e não pode ser nulo. Armazena o nome oficial da instituição de ensino.
- **endereço:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o logradouro onde a escola está localizada.
- **bairro:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o bairro da unidade escolar.
- **numero:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o número do imóvel da escola.

- **complemento:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena informações adicionais de localização.
- **municipios_idmunicipios:** Chave estrangeira da tabela *municipios*, do tipo Inteiro e não nula. Realiza o vínculo da escola com a sua respectiva cidade.
- **cnpj:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada, possui restrição de valor único e pode ser nulo. Armazena o Cadastro Nacional da Pessoa Jurídica da escola.
- **telefone:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o número de contato telefônico da instituição.
- **email:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o *email* da institucional.

B.2.6 Tabela *servidor_vinculo*

A tabela *servidor_vinculo* é responsável por gerenciar o vínculo empregatício e funcional dos servidores com as unidades de ensino. Ela armazena informações contratuais e de admissão no banco de dados e possui os seguintes campos:

- **idvinculo:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro de vínculo.
- **codigo_matricula:** Campo do tipo *varchar*, aceita até 20 caracteres como entrada, possui restrição de valor único e pode ser nulo. Armazena o número de matrícula funcional do servidor no município.
- **usuario_idusuario:** Chave estrangeira da tabela *usuario*, do tipo Inteiro e não nula. Identifica o usuário proprietário do vínculo.
- **escola_idescola:** Chave estrangeira da tabela *escola*, do tipo Inteiro e não nula. Realiza o vínculo do servidor com a instituição de ensino onde está lotado.
- **cargo_idcargo:** Chave estrangeira da tabela *cargo*, do tipo Inteiro e não nula. Identifica a função exercida pelo servidor.
- **regime:** Campo do tipo *varchar*, aceita até 30 caracteres como entrada e pode ser nulo. Define o tipo de contratação.

- **data_admissao:** Campo do tipo *date* e pode ser nulo. Armazena a data oficial de entrada do servidor no serviço público ou na unidade.
- **status:** Campo do tipo *varchar*, aceita até 20 caracteres como entrada, não pode ser nulo e possui o valor padrão 'ativo'. Indica a situação atual do vínculo do servidor.
- **observacao:** Campo do tipo *text* e pode ser nulo. Espaço destinado a anotações administrativas complementares.
- **validade_inicio:** Campo do tipo *date* e pode ser nulo. Registra a data inicial de vigência do contrato ou portaria.
- **validade_fim:** Campo do tipo *date* e pode ser nulo. Registra a data final de vigência, caso o vínculo seja por tempo determinado.
- **criado_em:** Campo do tipo *timestamp*, pode ser nulo e possui valor padrão `CURRENT_TIMESTAMP`. Registra automaticamente o momento em que o vínculo foi inserido no sistema.

B.2.7 Tabela turma

A tabela turma é responsável por armazenar a configuração e as características das turmas escolares em cada ano letivo e unidade de ensino. Possui os seguintes campos:

- **idturma:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro na tabela.
- **nome:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena a nomenclatura de identificação da turma
- **escola_idescola:** Chave estrangeira da tabela escola, do tipo Inteiro e pode ser nula. Realiza o vínculo da turma com a instituição de ensino onde ela funciona.
- **nivel_ensino_idnivel:** Chave estrangeira da tabela nivel, do tipo Inteiro e pode ser nula. Identifica a qual nível educacional a turma pertence
- **serie:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Indica o ano ou série escolar correspondente à turma.
- **turno:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o período em que as aulas ocorrem

- **capacidade:** Campo do tipo Inteiro e pode ser nulo. Define o limite máximo de alunos que podem ser matriculados na referida turma.
- **ano_letivo:** Campo do tipo *year* e pode ser nulo. Armazena o ano de vigência da turma no calendário escolar.

B.2.8 Tabela *matricula_has_turma*

Esta tabela é responsável por realizar o relacionamento entre os alunos matriculados e suas respectivas turmas, permitindo o controle de enturmação por ano letivo. Possui os seguintes campos:

- **matricula_idmatricula:** Chave estrangeira da tabela *matricula*, do tipo Inteiro e não nula. Identifica o registro da matrícula do aluno associado à turma.
- **turma_idturma:** Chave estrangeira da tabela *turma*, do tipo Inteiro e não nula. Identifica a turma na qual o aluno está enturmado.
- **ano:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o ano referente ao vínculo de enturmação.
- **professor_idusuario:** Chave estrangeira da tabela *usuario*, do tipo Inteiro e pode ser nula. Identifica o professor responsável ou conselheiro vinculado àquela turma/aluno.
- **carga_horaria:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena a carga horária total vinculada a este registro de enturmação.

B.2.9 Tabela *notas_alunos*

A tabela *notas_alunos* é responsável por armazenar os lançamentos individuais das avaliações dos estudantes ao longo do ano letivo. Possui os seguintes campos:

- **idnotas:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro.
- **nota:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o valor numérico ou conceito atribuído ao aluno.
- **historico_idhistorico:** Chave estrangeira da tabela *historico*, do tipo Inteiro e pode ser nula. Vincula a nota ao registro acadêmico específico do aluno naquela série e ano.

- **disciplinas_iddisciplinas:** Chave estrangeira da tabela disciplinas, do tipo Inteiro e pode ser nula. Identifica a qual matéria a nota pertence.
- **ano:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Registra o ano letivo do lançamento.
- **avaliacao:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Define o período ou tipo de avaliação
- **observacao:** Campo do tipo *text* e pode ser nulo. Espaço para anotações complementares do professor sobre o desempenho do aluno.
- **data_lancamento:** Campo do tipo *timestamp*, pode ser nulo e possui valor padrão CURRENT_TIMESTAMP. Registra automaticamente a data e hora em que a nota foi inserida no sistema.

B.2.10 Tabela frequencia_alunos

A tabela frequencia_alunos é responsável por armazenar o registro diário de presença e ausência dos estudantes em cada disciplina, permitindo o controle da assiduidade escolar. Possui os seguintes campos:

- **idfrequencia:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro de frequência.
- **historico_idhistorico:** Chave estrangeira da tabela historico, do tipo Inteiro e pode ser nula. Vincula a frequência ao registro acadêmico anual do aluno.
- **status:** Campo do tipo *varchar*, aceita até 20 caracteres como entrada e pode ser nulo. Armazena a situação do aluno no dia.
- **data:** Campo do tipo *date* e não pode ser nulo. Registra o dia específico em que a chamada foi realizada.
- **observacao:** Campo do tipo *text* e pode ser nulo. Espaço destinado a justificativas ou anotações complementares sobre a frequência do aluno.
- **disciplinas_iddisciplinas:** Chave estrangeira da tabela disciplinas, do tipo Inteiro e pode ser nula. Identifica a qual matéria o registro de presença se refere.

B.2.11 Tabela transferencia

A tabela transferencia é responsável por registrar o histórico e o gerenciamento das solicitações de movimentação de alunos entre diferentes unidades de ensino. Ela armazena os dados do processo de saída e entrada no banco de dados e possui os seguintes campos:

- **idtransferencia:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro de transferência.
- **matricula_idmatricula:** Chave estrangeira da tabela matricula, do tipo Inteiro e pode ser nula. Identifica o aluno que está solicitando a transferência.
- **escola_origem:** Chave estrangeira da tabela escola, do tipo Inteiro e pode ser nula. Identifica a instituição de ensino de onde o aluno está saindo.
- **escola_destino:** Chave estrangeira da tabela escola, do tipo Inteiro e pode ser nula. Identifica a instituição de ensino para onde o aluno deseja ser transferido.
- **data_solicitacao:** Campo do tipo *date* e pode ser nulo. Registra a data em que o pedido de transferência foi inserido no sistema.
- **data_aceitacao:** Campo do tipo *date* e pode ser nulo. Registra a data em que a escola de destino ou a secretaria confirmou o recebimento do aluno.
- **autorizado_por:** Chave estrangeira da tabela usuario, do tipo Inteiro e pode ser nula. Identifica o servidor ou diretor responsável pela homologação do processo.
- **status:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Indica a situação atual do processo.

B.2.12 Tabela disciplinas

A tabela disciplinas é responsável por armazenar as informações das matérias que compõem a grade curricular das unidades de ensino no banco de dados. Possui os seguintes campos:

- **iddisciplinas:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação da disciplina no sistema.
- **nome:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o nome oficial da disciplina.

- **serie:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Indica a série ou ano escolar ao qual a disciplina está vinculada.
- **codigo:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena um código interno de identificação para controle acadêmico.
- **carga_horaria:** Campo do tipo Inteiro e pode ser nulo. Define a quantidade total de horas-aula previstas para a disciplina durante o ano letivo.

B.2.13 Tabela disciplinas_has_turma

Esta tabela é responsável por realizar o relacionamento entre as disciplinas e as turmas, definindo quais matérias compõem a grade curricular de cada turma específica. Possui os seguintes campos:

- **disciplinas_iddisciplinas:** Chave estrangeira da tabela disciplinas, do tipo Inteiro e não nula. Identifica o registro da disciplina que está sendo vinculada à turma.
- **turma_idturma:** Chave estrangeira da tabela turma, do tipo Inteiro e não nula. Identifica a turma específica à qual a disciplina pertence.

B.2.14 Tabela disciplinas_has_matricula

Esta tabela é responsável por realizar o vínculo entre as disciplinas e a matrícula ativa do aluno, permitindo o armazenamento do histórico de notas e médias finais por disciplina no banco de dados. Possui os seguintes campos:

- **disciplinas_iddisciplinas:** Chave estrangeira da tabela disciplinas, do tipo Inteiro e não nula. Identifica a matéria vinculada à matrícula do aluno.
- **matricula_idmatricula:** Chave estrangeira da tabela matricula, do tipo Inteiro e não nula. Identifica o registro de matrícula ao qual a disciplina pertence.
- **nota:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o valor da nota final ou média obtida pelo estudante.
- **ano:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Registra o ano letivo referente ao aproveitamento daquela disciplina.
- **avaliacao:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Define a categoria da nota registrada, como "media", "final" ou "recuperacao".

B.2.15 Tabela historico

A tabela historico é responsável por registrar o desempenho consolidado, a série cursada e a situação acadêmica final do aluno em cada ano letivo. Ela armazena os dados do percurso escolar no banco de dados e possui os seguintes campos:

- **idhistorico:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro acadêmico.
- **serie:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Indica o ano ou série escolar correspondente ao registro.
- **descricao:** Campo do tipo *text* e pode ser nulo. Espaço destinado a observações detalhadas sobre o desempenho ou ocorrências do aluno naquele período.
- **matricula_idmatricula:** Chave estrangeira da tabela matricula, do tipo Inteiro e pode ser nula. Realiza o vínculo do histórico com o registro de matrícula do aluno.
- **escola_idescola:** Chave estrangeira da tabela escola, do tipo Inteiro e pode ser nula. Identifica a unidade escolar onde o aluno cursou a referida série.
- **ano:** Campo do tipo *date* e pode ser nulo. Registra a data de referência para o período letivo em questão.
- **situacao:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena a situação final do ano escolar do aluno.

B.2.16 Tabela calendario_gestao

A tabela calendario_gestao é responsável por gerenciar os prazos administrativos e registros de avaliações da escola, estabelecendo os períodos para matrículas, aulas e lançamentos de notas no banco de dados. Possui os seguintes campos:

- **idcalendario_gestao:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro.
- **ano_letivo:** Campo do tipo *year* e não pode ser nulo. Define o ano de vigência do calendário escolar.
- **data_inicio_matricula:** Campo do tipo *date* e pode ser nulo. Registra o primeiro dia para a realização de matrículas.

- **data_fim_matricula:** Campo do tipo *date* e pode ser nulo. Registra o prazo final para o processo de matrícula.
- **data_inicio_aulas:** Campo do tipo *date* e pode ser nulo. Indica a data de início do primeiro semestre letivo.
- **data_fim_aulas:** Campo do tipo *date* e pode ser nulo. Indica o encerramento das atividades escolares do ano.
- **escola_idescola:** Chave estrangeira da tabela escola, do tipo Inteiro e pode ser nula. Vincula o cronograma de prazos a uma unidade escolar específica.
- **inicio_lancamento_av1:** Campos do tipo *date* e pode ser nulo. Delimita o prazo de início para os professores inserirem as notas da primeira avaliação no sistema.
- **fim_lancamento_av1:** Campos do tipo *date* e pode ser nulo. Delimita o prazo que finaliza para os professores inserirem as notas da primeira avaliação no sistema.
- **inicio_lancamento_av2:** Campos do tipo *date* e pode ser nulo. Delimita o prazo de início para os professores inserirem as notas da segunda avaliação no sistema.
- **fim_lancamento_av2:** Campos do tipo *date* e pode ser nulo. Delimita o prazo que finaliza para os professores inserirem as notas da segunda avaliação no sistema.
- **inicio_lancamento_av3:** Campos do tipo *date* e pode ser nulo. Delimita o prazo de início para os professores inserirem as notas da terceira avaliação no sistema.
- **fim_lancamento_av3:** Campos do tipo *date* e pode ser nulo. Delimita o prazo que finaliza para os professores inserirem as notas da terceira avaliação no sistema.
- **inicio_lancamento_av4:** Campos do tipo *date* e pode ser nulo. Delimita o prazo de início para os professores inserirem as notas da quarta avaliação no sistema.
- **fim_lancamento_av4:** Campos do tipo *date* e pode ser nulo. Delimita o prazo que finaliza para os professores inserirem as notas da quarta avaliação no sistema.
- **inicio_lancamento_recuperacao:** Campos do tipo *date* e pode ser nulo. Delimita o prazo de início para os professores inserirem as notas de recuperação no sistema.
- **fim_lancamento_recuperacao:** Campos do tipo *date* e pode ser nulo. Delimita o prazo que finaliza para os professores inserirem as notas de recuperação no sistema.

B.2.17 Tabela `calendario_aula`

A tabela `calendario_aula` é responsável por armazenar o cronograma de aulas, realizando o vínculo entre turmas, disciplinas, professores e horários para a organização do ano letivo no banco de dados. Possui os seguintes campos:

- **idcalendario:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro do calendário.
- **turma_idturma:** Chave estrangeira da tabela `turma`, do tipo Inteiro e pode ser nula. Realiza o vínculo do horário com uma turma específica.
- **disciplinas_iddisciplinas:** Chave estrangeira da tabela `disciplinas`, do tipo Inteiro e pode ser nula. Identifica a qual matéria o horário de aula se refere.
- **professor_idusuario:** Chave estrangeira da tabela `usuario`, do tipo Inteiro e pode ser nula. Identifica o professor responsável por ministrar a aula naquele período.
- **horario_idhorario:** Chave estrangeira da tabela `horario_aula`, do tipo Inteiro e pode ser nula. Vincula o registro ao tempo de aula (dia da semana e hora) definido na grade horária.
- **data_inicio:** Campo do tipo `date` e pode ser nulo. Registra a data inicial de vigência deste cronograma de aula.
- **data_fim:** Campo do tipo `date` e pode ser nulo. Registra a data final de validade do horário estabelecido.

B.2.18 Tabela `horario_aula`

A tabela `horario_aula` é responsável por definir os intervalos de tempo e os dias da semana em que as atividades letivas ocorrem, servindo de base para a montagem do quadro de horários das turmas. Possui os seguintes campos:

- **idhorario:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação de cada horário cadastrado no sistema.
- **dia_semana:** Campo do tipo `varchar`, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o nome do dia da semana referente ao horário.
- **hora_inicio:** Campo do tipo `time` e pode ser nulo. Registra o horário exato em que a aula inicia.

- **hora_fim:** Campo do tipo *time* e pode ser nulo. Registra o horário exato em que a aula encerra.

B.2.19 Tabela cargo

A tabela cargo é responsável por definir as funções e ocupações dos servidores dentro da instituição de ensino no banco de dados. Possui os seguintes campos:

- **idcargo:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro na tabela.
- **nome:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e não pode ser nulo. Armazena a nomenclatura do cargo ocupado.

B.2.20 Tabela falta_professor

A tabela falta_professor é responsável por registrar as ausências dos docentes, permitindo o controle administrativo das faltas e o armazenamento de documento de justificativas no banco de dados. Possui os seguintes campos:

- **idfalta:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro de falta.
- **professor_idusuario:** Chave estrangeira da tabela usuario, do tipo Inteiro e pode ser nula. Identifica o professor que faltou à atividade letiva.
- **data:** Campo do tipo *date* e pode ser nulo. Registra o dia específico em que ocorreu a ausência.
- **motivo:** Campo do tipo *text* e pode ser nulo. Descrição detalhada sobre a razão da falta.
- **arquivo_justificativa:** Campo do tipo *varchar*, aceita até 255 caracteres como entrada e pode ser nulo. Armazena o caminho (*path*) do arquivo de comprovante ou atestado médico enviado ao servidor.
- **registrado_por:** Chave estrangeira da tabela usuario, do tipo Inteiro e pode ser nula. Identifica o servidor ou secretário responsável por inserir o registro de falta no sistema.

B.2.21 Tabela responsavel

A tabela responsavel é utilizada para armazenar as informações cadastrais e de contacto dos pais ou tutores legais dos alunos no banco de dados. Possui os seguintes campos:

- **idresponsavel:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro na tabela.
- **nome:** Campo do tipo *varchar*, aceita até 100 caracteres como entrada e pode ser nulo. Armazena o nome completo do responsável legal.
- **endereço:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o endereço residencial do responsável.
- **telefone:** Campo do tipo *varchar*, aceita até 20 caracteres como entrada e pode ser nulo. Armazena o número de contato do responsável.
- **cpf:** Campo do tipo *varchar*, aceita até 14 caracteres como entrada, possui restrição de valor único e pode ser nulo. Armazena o Cadastro de Pessoa Física do responsável.
- **email:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena o endereço de *email* para comunicações oficiais.

B.2.22 Tabela responsavel_has_usuario

Esta tabela é responsável por realizar o relacionamento entre os responsáveis cadastrados e os usuários aluno do sistema, permitindo definir o vínculo familiar ou legal entre as partes no banco de dados. Possui os seguintes campos:

- **reponsavel_idresponsavel:** Chave estrangeira da tabela responsavel, do tipo Inteiro e não nula. Identifica o registro do responsável na associação.
- **usuario_idusuario:** Chave estrangeira da tabela usuario, do tipo Inteiro e não nula. Identifica o usuário aluno que está vinculado ao referido responsável.

grau_parentesco: Campo do tipo *varchar*, aceita até 45 caracteres como entrada e pode ser nulo. Armazena a descrição da relação de parentesco entre o responsável e o aluno.

B.2.23 Tabela nivel

A tabela nivel é responsável por definir as diferentes etapas ou níveis de ensino oferecidos pela instituição no banco de dados. Possui os seguintes campos:

- **idnivel:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação de cada nível de ensino cadastrado.
- **descricao:** Campo do tipo *varchar*, aceita até 100 caracteres como entrada e pode ser nulo. Armazena a nomenclatura descritiva do nível educacional.

B.2.24 Tabela *password_reset_tokens*

Esta tabela é responsável por gerir os tokens de segurança gerados para o processo de recuperação de palavra-passe dos utilizadores, garantindo a validade e o controlo do uso destes pedidos no sistema. Possui os seguintes campos:

- **id:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação do registro de recuperação.
- **usuario_idusuario:** Chave estrangeira da tabela usuario, do tipo Inteiro e não nula. Identifica o utilizador que solicitou a alteração da palavra-passe.
- **token:** Campo do tipo *varchar*, aceita até 64 caracteres como entrada, possui restrição de valor único e não pode ser nulo. Armazena a chave encriptada necessária para validar o pedido de recuperação.
- **expires_at:** Campo do tipo *datetime* e não pode ser nulo. Armazena a data e a hora limite para a utilização do *token*.
- **used_at:** Campo do tipo *datetime* e pode ser nulo. Registra o momento exato em que o *token* foi utilizado para concluir a alteração da palavra-passe.
- **created_at:** Campo do tipo *datetime*, não nulo e com valor padrão *CURRENT_TIMESTAMP*. Registra automaticamente a data e a hora em que o pedido de recuperação foi gerado pelo sistema.

B.2.25 Tabela estado

A tabela estado é responsável por armazenar o cadastro do nome dos estados para a organização geográfica dos endereços no sistema. Possui os seguintes campos:

- **idestado:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação de cada estado cadastrado no banco de dados.
- **nome:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e não pode ser nulo. Armazena o nome completo ou a sigla do Estado

B.2.26 Tabela municipios

A tabela municipios é responsável por armazenar o cadastro das cidades vinculadas aos seus respectivos estados, sendo utilizada para a padronização de endereços de alunos, usuários e escolas no banco de dados. Possui os seguintes campos:

- **idmunicipios:** Chave primária da tabela, do tipo Inteiro, auto incrementável e não nula. Possui o código único de identificação de cada município cadastrado.
- **nome:** Campo do tipo *varchar*, aceita até 45 caracteres como entrada e não pode ser nulo. Armazena o nome oficial da cidade.
- **estado_idestado:** Chave estrangeira da tabela estado, do tipo Inteiro e pode ser nula. Realiza o vínculo do município com a seu estado correspondente.