

WAD

WEB APPLICATION DOCUMENT

ATHENA

Autores: Davi Ferreira Arantes

Marcela Issa Avila Vieiralves Martins

Marcelo Miguel Pereira de Assis

Pedro Auler de Barros Martins

Raissa de Cássia Moraes Paula

Victor Gabriel Marques

Data de criação: 18 de abril de 2023

Versão: 0.3.0

Controle do Documento

Histórico de revisões

Quadro 1 - Histórico de revisões do documento

Data	Autor	Versão	Resumo da atividade
28/04/2023	Equipe Prometheus	0.1.0	Preenchimento das seções 1, 2 e 3 e do tópico 5.1.
12/05/2023	Equipe Prometheus	0.2.0	Preenchimento das seções 4 e 6, além do apêndice A.
26/05/2023	Equipe Prometheus	0.3.0	Preenchimento da seção 5.2, com as informações do Guia de Estilo.

Fonte: Os autores (2023)

Organização da Equipe

Quadro 2 - Organização da equipe de acordo com a versão do documento

Nome	Papel	Funções por Versão
Davi Ferreira Arantes	Design QA Programação Documentação	0.1.0 - Criação dos wireframes móveis da aplicação web e revisão do WAD. 0.2.0 - Elaboração do modelo conceitual do banco de dados, testes dos endpoints, documentação dos endpoints e apresentação. 0.3.0 - Estruturação do css principal da aplicação web, criação do header (mobile) e da versão inicial da página de login (desktop e mobile).
Marcela Issa Avila Vieiralves Martins	Design Programação Documentação	0.1.0 - Criação das personas e das <i>User Stories</i> , criação dos slides e das logos do grupo e da aplicação web.

		<p>0.2.0 - Criação dos slides, criação de endpoint de cadastro, escrita dos tópicos de tecnologia e criação do <i>wireframe</i> desktop.</p> <p>0.3.0 - Criação dos slides, criação da tela de cadastro (desktop e mobile), elaboração do <i>mockup</i> e do guia de estilos.</p>
Marcelo Miguel Pereira de Assis	QA Programação Documentação	<p>0.1.0 - Criação da matriz de risco e da matriz de oportunidades do projeto e da proposta de solução para cada risco.</p> <p>0.2.0 - Elaboração do modelo conceitual do banco de dados, criação dos endpoints, incluindo testes e documentação.</p> <p>0.3.0 - Revisão e melhoria da página de login, criação da página de perfil (desktop e mobile) e participação na apresentação do projeto.</p>
Pedro Auler de Barros Martins	Documentação	<p>0.1.0 - Criação da análise SWOT da empresa Nova Escola.</p> <p>0.3.0 - Criação da página de dados do perfil (mobile), pesquisa acerca da segurança cibernética e participação na apresentação do projeto.</p>
Raissa de Cássia Moraes Paula	Design Documentação	<p>0.1.0 - Escrita da Análise da Indústria baseada nas 5 forças de Porter, da descrição do problema e da descrição do parceiro e criação dos slides.</p> <p>0.2.0 - Escrita dos tópicos sobre tecnologias, criação dos slides e criação do <i>wireframe</i> para desktop.</p> <p>0.3.0 - Elaboração do <i>mockup</i> e do guia de estilos, implementação de validações nos endpoints de alunos, escrita do WAD.</p>

Victor Gabriel Marques	Design QA Programação Documentação	<p>0.1.0 - Escrita dos objetivos gerais e específicos do projeto, da descrição da solução, das partes interessadas, da proposta de valor e do texto dos wireframes. Revisão do WAD (SWOT, personas e matrizes de risco e de oportunidade). Criação dos wireframes e dos slides da apresentação.</p> <p>0.2.0 - Criação de endpoints, estruturação do padrão de projeto MVC, criação dos modelos conceitual, lógico e físico do banco de dados e testes dos endpoints.</p> <p>0.3.0 - Criação e implementação de validações dos endpoints de notas e blocos de questão, padronização das páginas de login, cadastro e perfil e criação da página inicial (mobile e desktop).</p>
------------------------	---	--

Fonte: Os autores (2023)

Sumário

1. Visão Geral do Projeto

- 1.1. Parceiro de Negócios
- 1.2. O Problema
- 1.3. Objetivos
 - 1.3.1. Objetivos gerais
 - 1.3.2. Objetivos específicos
- 1.4. Descritivo da Solução
- 1.5. Partes Interessadas

2. Análise do Problema

- 2.1. Análise da Indústria
- 2.2. Análise do cenário: Matriz SWOT
- 2.3. Proposta de Valor: Value Proposition Canvas
- 2.4. Matriz de Risco

3. Requisitos do Sistema

- 3.1. Persona
- 3.2. Histórias dos usuários (user stories)

4. Arquitetura do Sistema

- 4.1. Módulos do Sistema e Visão Geral (Big Picture)
- 4.2. Tecnologias Utilizadas

5. UX e UI Design

- 5.1. Wireframe
- 5.2. Design de Interface - Guia de Estilos

6. Projeto de Banco de Dados

- 6.1. Modelo Conceitual
- 6.2. Modelo Lógico
- 6.3. Modelo Físico

7. Testes de Software

- 7.1. Teste de Usabilidade

Referências

Apêndice A - Relatório de Criação e Documentação dos Endpoints

Anexos

1. Visão Geral do Projeto

Este tópico abordará a Visão Geral do Projeto, incluindo uma análise da Nova Escola, a problemática que será solucionada por meio do site, os objetivos gerais e específicos do projeto, a descrição da solução e quem são as partes interessadas.

1.1. Parceiro de Negócios

A Nova Escola, criada em 2015 com o apoio de sua mantenedora, a Fundação Lemann, e herdeira da revista de mesmo nome, nascida em 1986 na Fundação Victor Civita, é uma organização de impacto social sem fins lucrativos que possui o objetivo de fortalecer professores da Educação Básica em suas práticas, contribuindo para a melhoria da aprendizagem e do desenvolvimento dos estudantes de escola pública.

Além disso, a Nova Escola é uma plataforma digital que, de acordo com os dados encontrados em seu site, é acessada por 3,1 milhões de pessoas e a qual contém materiais produzidos pelos próprios professores, tais como reportagens, cursos auto instrucionais, formações, planos de aula e materiais educacionais.

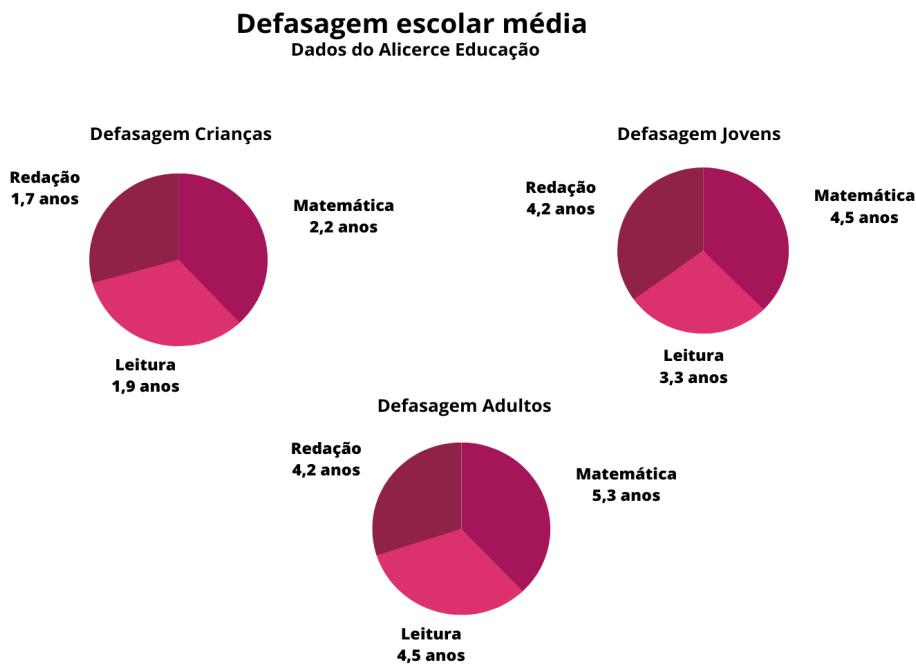
Ademais, a empresa é a marca mais reconhecida por professores de Educação Básica no Brasil e possui produtos, serviços e conteúdos que valorizam os professores, facilitam seu dia-a-dia e apoiam suas carreiras.

1.2. O Problema

Devido a grande quantidade de alunos nas escolas públicas, muitos professores não conseguem ter um contato direto com todos e acabam não os conhecendo direito, tendo dificuldade para reconhecer suas defasagens e não conseguindo ajudá-los.

Nesse cenário, de acordo com uma pesquisa realizada pela escola Alicerce com 2763 alunos de todo Brasil, sendo 2.265 crianças entre cinco e 13 anos, 237 jovens entre 14 e 25 anos, e 261 adultos com idade superior a 25 anos, foi identificado o índice de defasagem escolar médio, isto é, quando os níveis de conhecimento do estudante não correspondem à sua série escolar. Nesse viés, esses dados estão demonstrados na figura abaixo.

Figura 01 - Resultados da pesquisa do Alicerce Educação



Fonte: Os autores (2023) com base nos dados da pesquisa da escola Alicerce

Após adquirir os resultados, a preocupação aumentou, dado que o impacto pode ser irreversível na educação pública brasileira. Portanto, para a resolução dessa problemática abordada no projeto, é necessário que o professor consiga traçar estratégias de aulas, por meio de um diagnóstico que o capacite para identificar as defasagens que suas turmas apresentam, a fim de ter um bom plano de aula para auxiliar o aprendizado de seus alunos e reduzir a diferença no nível de conhecimento deles em relação a série em que se encontram.

1.3. Objetivos

Com o intuito de direcionar o projeto, além de informar o leitor sobre as metas deste trabalho, esta seção aborda os objetivos gerais e específicos do projeto, os quais representam, respectivamente, a ideia central do trabalho (construção da *Athena*) e os resultados que este trabalho visa atingir a partir da construção da *Athena*.

1.3.1. Objetivos gerais

Desenvolver uma aplicação web que contribua com o professor usuário no mapeamento das necessidades da sua turma, o que inclui a análise das habilidades

acadêmicas dos alunos que estão com maior defasagem e, por conseguinte, a sugestão de planos de ação para a melhoria contínua das aulas e para a diminuição dessa defasagem.

1.3.2. Objetivos específicos

- Construir a persona do usuário final, de modo a alinhar a aplicação web às necessidades dos usuários;
- Criar os wireframes da aplicação, para validar a ideia junto ao parceiro de projeto;
- Definir critérios para a segurança no desenvolvimento e na disponibilização da Athena;
- Estudar sobre acessibilidade na web, visando um desenvolvimento semântico da aplicação, de modo que pessoas com deficiências possam acessar, compreender, navegar e interagir na aplicação;
- Desenvolver o sistema para as bases de dados, as quais servirão como armazenamento das informações da aplicação web;
- Disponibilizar, entre as funcionalidades da aplicação web Athena, a possibilidade do professor cadastrar mais de uma turma;
- Providenciar um sistema para que o usuário final possa subir as avaliações feitas em sala de aula;
- Disponibilizar um sistema de cadastro de alunos, para que o professor consiga analisá-los individualmente;
- Exibir, nas páginas relacionadas aos alunos e às turmas, dados visuais e textuais para facilitar a análise do professor;
- Conceder um sistema de sugestão de planos de aula e materiais didáticos de acordo com o índice de defasagem dos alunos.

1.4. Descritivo da Solução

A *Athena* é uma aplicação web, criada a partir de tecnologias de design (como o Figma¹), de desenvolvimento front-end (HTML², CSS³ e JavaScript⁴), de desenvolvimento

¹ Editor gráfico para a prototipagem de projetos baseado no navegador: <https://www.figma.com/about/>

² Linguagem de marcação usada para construir a estrutura de um site:
<https://www.w3schools.com/html/>

³ Linguagem usada para utilizar componentes feitos em HTML: <https://www.w3schools.com/css/>

⁴ Linguagem de programação usada para adicionar elementos complexos num site:
<https://www.w3schools.com/js/>

back-end (Node.js⁵) e de bancos de dados (Sqlite⁶), com o intuito de colaborar com a resolução do problema apresentado pela empresa parceira, o qual trata da falta de métodos para avaliar o grau de defasagem dos alunos e para o planejamento eficiente das aulas. Sendo assim, tal solução é essencial para a apresentação acessível e detalhada desses aspectos (índice de defasagem dos alunos e sugestão de materiais acadêmicos) para os professores de toda a rede de ensino do Brasil.

Nesse viés, entre as suas funcionalidades destacam-se: sistema para o cadastro do professor na plataforma; sistema para o cadastro do aluno pelo professor; gráficos para análise da média e da defasagem dos alunos e da turma em geral; sistema para que o professor possa cadastrar avaliações que podem ser reaproveitadas entre turmas e resultem em análises detalhadas por áreas (habilidades); sugestão de planos de aulas e materiais didáticos para cada turma; entre outros.

1.5. Partes Interessadas

Os stakeholders do projeto são todas as pessoas, empresas ou organizações que possuem interesse na construção desta aplicação web e, por conseguinte, são aqueles que são impactados - direta ou indiretamente - pelos resultados do projeto. Nessa direção, abaixo estão listados os stakeholders deste trabalho e suas respectivas funções para tal projeto.

- **Instituto de Tecnologia e Liderança:** Instituição responsável pela concretização do projeto;
- **Fabiana Martins:** Orientadora do projeto;
- **Nova Escola:** Organização parceira do projeto, para qual destina-se a *Athena*;
- **Daniela Giugliano:** Ponto focal e responsável pela área executiva do projeto;
- **Laís Semis:** Ponto focal backup (auxiliar) do projeto;
- **Felipe Costa:** Responsável da área técnica do projeto;
- **Tatiana Martin:** Responsável da área de negócios do projeto.

⁵ Node é um ambiente de execução para códigos em JavaScript: <https://nodejs.org/en/docs>

⁶ Sqlite é um tipo de banco de dados que não precisa de servidor para atuar:
<https://www.sqlite.org/docs.html>

2. Análise do Problema

Este tópico aborda a análise do problema de uma maneira geral, incluindo a análise da indústria e do cenário da empresa parceira, as 5 forças de Porter, a Análise SWOT, o *Value Proposition Canvas* e a Matriz de Risco.

2.1. Análise da Indústria

As 5 forças de Porter, que correspondem aos concorrentes, concorrentes potenciais, clientes, fornecedores e produtos substitutos, é um framework utilizado para realizar uma análise da competitividade de empresas dentro de um mercado.

Sendo assim, após realizar uma pesquisa sobre a Nova Escola por meio das cinco forças de Porter, foi possível obter os seguintes resultados sobre as forças, as ameaças que elas representam e quais as reações possíveis que a empresa poderá ter para evitá-las:

Figura 02 - Análise das 5 forças de Porter sobre a Nova Escola



Fonte: Os autores (2023)

Concorrentes: Khan Academy, Educa Brasil, AVAMEC.

Ameaças: Perda de clientes e fornecedores

Reações Possíveis: Manter seu conteúdo atualizado, interessante e de fácil acesso para os professores, de modo que continuem querendo usufruir dos materiais e dos cursos da Nova Escola. Como também, continuar mostrando os bons resultados para os fornecedores a fim de que eles vejam o retorno de suas doações e queiram continuar contribuindo.

Concorrentes potenciais: Escolas conectadas, Só Educador, Sebrae, Apoio ao Professor.

Ameaças: Os concorrentes potenciais se desenvolverem e a nova escola acabar perdendo seus clientes e seus fornecedores.

Reações Possíveis: Manter o seu conteúdo sempre atualizado e interessante para os professores, a fim de que eles possam se manter fiéis à nova escola e queiram continuar utilizando dos seus serviços, gerando ótimos resultados para organização, o que fará com que ela possa conquistar cada vez mais clientes e fornecedores.

Clientes: Professores da rede pública

Ameaças: Os professores não terem tempo e infraestrutura para realizar os cursos, formações e planos de aula da nova escola e dificuldade de professores com idades mais avançadas de acessar o site.

Reações Possíveis: Buscar parceiros para ajudar os professores que tenham interesse em acessar os conteúdos com a infraestrutura ou recomendar lugares, como bibliotecas, que possuem acesso a computadores e internet. Além disso, mostrar maneiras de organização para os professores terem tempo, deixar sempre o conteúdo da maneira mais fácil e rápida de acessar e fazer tutoriais para que os usuários entendam como mexer no site.

Fornecedores: Gerdau, Instituto Natura, Itaú social, Fundação Lemann, Alana, Instituto CSHG, Instituto Cyrela, Facebook, Fundação Maria Cecilia Souto Vidigal, Fundação Roberto Marinho, Vivo, Futura, Verde Asset Management, Google.org, Gov.uk, Imaginable Futures, Instituto Palavra Aberta, Instituto Unibanco, Parceiros da educação RJ, Instituto Sonho Grande, Fundação Tide Setubal, UNDIME, Instituto Votorantim, YouTube, Instituto Chamex, B3 Social, Instituto XP, Tinker Foundation.

Ameaças: Desistirem de fornecer apoio para a nova escola continuar auxiliando os professores.

Reações Possíveis: Continuar demonstrando o ótimo trabalho e os resultados que vem sendo feito com a nova escola para que os fornecedores sejam motivados a continuar apoiando.

Produtos substitutos: cursos de maneira presencial em escolas, palestras, revistas; reformulação, por parte do MEC, da graduação dos professores.

Ameaças: Substituir os cursos, formações e planos de aula digitais da nova escola por outros formatos; diminuir a necessidade dos professores em acessar os conteúdos da Nova Escola, já que sairiam da graduação mais preparados.

Reações Possíveis: Uma divulgação maior da nova escola para que as instituições de ensino possam conhecê-la, indicá-la aos professores e instruí-los a fazerem as formações, cursos e utilizarem os seus planos de aula; inovação na entrega e na construção dos cursos e materiais, mostrando aos professores que o material da Nova Escola é essencial mesmo com a reformulação dos cursos.

Portanto, após essa análise, é possível concluir que a Nova Escola é uma empresa com tendência a evoluir cada vez mais, continuar conquistando novos clientes e fornecedores e seguir avançando em posição de destaque na área da educação.

2.2. Análise do cenário: Matriz SWOT

A priori, é válido ressaltar que a matriz SWOT trata-se de uma ferramenta gerencial de análise estratégica, a qual é capaz de explorar os potenciais, tanto internos quanto externos, da empresa ou organização analisada. Posto isso, essa matriz é composta de 4 principais conceitos, os quais estão representados em cada letra da palavra SWOT, sendo eles: Forças (*Strengths*), Fraquezas (*Weaknesses*), Oportunidades (*Opportunities*) e Ameaças (*Threats*). Dessa forma, a figura a seguir (figura 03) expõe a análise do cenário externo e interno da organização parceira deste projeto (a Nova Escola) com base nessa ferramenta.

Figura 03 - Matriz SWOT da Nova Escola

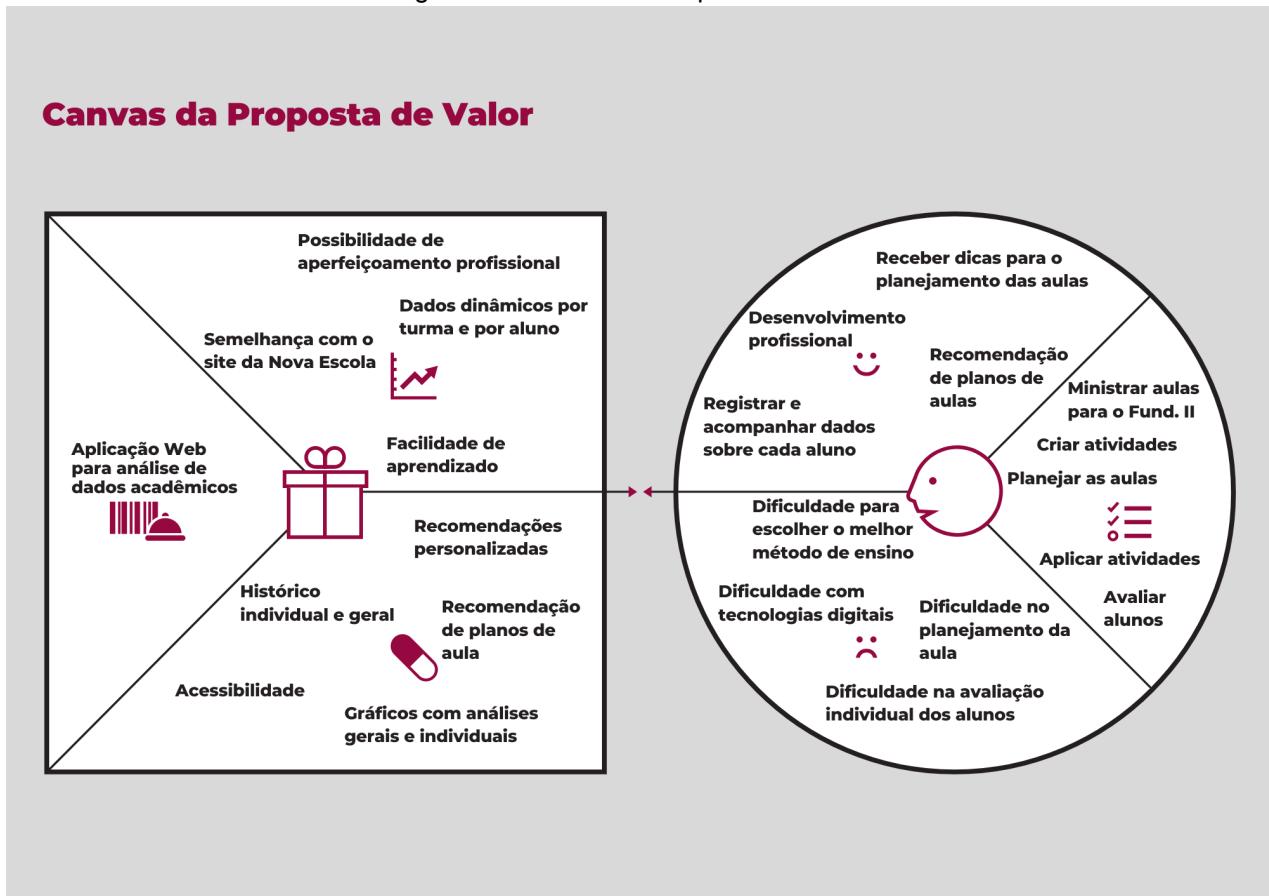


Fonte: Os autores (2023)

2.3. Proposta de Valor: Value Proposition Canvas

O *Value Proposition Canvas* (Canvas da Proposta de Valor) é uma ferramenta utilizada para ajudar empresas, organizações e até mesmo projetos a criar uma proposta de valor convincente para seus clientes ou usuários finais. Consiste em um modelo visual que permite a identificação dos elementos fundamentais que compõem a proposta de valor, tais como os benefícios oferecidos ao cliente, as dores e necessidades do cliente, as soluções oferecidas, dentre outros fatores. Por essa razão, para melhor apresentar a proposta de valor deste projeto, foi desenvolvido o canvas da proposta de valor da *Athena*, o qual está evidenciado na figura a seguir (figura 04).

Figura 04 - The Value Proposition Canvas



Fonte: Os autores (2023)

Como demonstrado na figura acima, os principais criadores de ganho deste projeto, especialmente da *Athena*, são: possibilidade de aperfeiçoamento profissional, dados dinâmicos por turma e por aluno, semelhança com o site da Nova Escola e facilidade de aprendizado. Esses temas carecem de um detalhamento, haja vista que são partes fundamentais deste trabalho.

Sob essa ótica, a possibilidade de aperfeiçoamento profissional diz respeito ao fato de que o professor terá acesso aos dados das suas turmas e, para além da análise que ele mesmo pode fazer, a aplicação dará sugestão de planos de aula e materiais didáticos para que ele possa melhorar as suas futuras aulas e, por conseguinte, há a melhoria contínua das aulas e a redução da defasagem dos alunos.

Os dados dinâmicos por turma e por aluno são fundamentais para que o usuário final consiga analisar tanto o aluno, enquanto indivíduo, quanto a turma, enquanto um conjunto de alunos. Nesse viés, os dados do aluno serão exclusivos dele e os dados da turma serão representados pela média entre todas as informações dos alunos dessa respectiva turma.

A semelhança com o site da Nova Escola é essencial para que o professor possa se sentir seguro ao navegar pela *Athena*, já que já navegou, pelo menos alguma vez, no site oficial da empresa. Por isso, além de conceder mais segurança ao professor, tal decisão implica na facilidade de aprendizado (outro criador de ganho) e de memorização da aplicação.

2.4. Matriz de Risco

Essa matriz tem como função principal indicar quais são os possíveis riscos para este projeto, assim como os impactos de cada risco, de modo que seja possível evitá-los ou amenizá-los. Todavia, caso eles venham a se concretizar, é possível contorná-los mais facilmente, devido ao plano de ação desenvolvido sobre a matriz. Além disso, a matriz de risco também pode servir como matriz de oportunidades e, para melhor representar ambas as ferramentas, as figuras a seguir (figura 05 e figura 06) representam, respectivamente, os riscos e as oportunidades deste projeto como um todo.

Figura 05 - Matriz de riscos do projeto

Probabilidade	Impacto			
		Alta	Média	Baixa
Alta	<ul style="list-style-type: none"> Aplicação incompatível com todas as versões de dispositivos [aplicação]. 	<ul style="list-style-type: none"> Equipe não segue o backlog [equipe]; Falta de tempo para maximizar a usabilidade do site [equipe]. 	<ul style="list-style-type: none"> Tarefas não realizadas por membros do grupo [equipe]; Pouco espaço no <i>market share</i> [aplicação]. 	
Média	<ul style="list-style-type: none"> Alterações no Design [aplicação]. 	<ul style="list-style-type: none"> Projeto sem padronização [aplicação]; Instabilidade na internet em algumas regiões do país [técnico]. 	<ul style="list-style-type: none"> Bugs não identificados até a implementação da aplicação. [aplicação]; Defeitos críticos encontrados na homologação da aplicação pelo cliente [aplicação]. 	
Baixa	<ul style="list-style-type: none"> Retirada de um membro do grupo [equipe]. 	<ul style="list-style-type: none"> Usuário não se sente confortável em disponibilizar seus dados [aplicação]. 		<ul style="list-style-type: none"> Solução não atende os requisitos/necessidades do projeto [aplicação]; Entregas fora dos prazos [equipe]; Usuário não entende como usar a aplicação [aplicação].
Insignificante		Moderado	Catastrófico	
Impacto				

Fonte: Os autores (2023)

Figura 06 - Matriz de oportunidades do projeto

Probabilidade	Alta	<ul style="list-style-type: none"> Feedbacks dos usuários e dos orientadores, após entrega do projeto [aplicação]. 		<ul style="list-style-type: none"> Equipe aprender sobre desenvolvimento web, design, negócios e metodologia ágil [equipe].
	Média	<ul style="list-style-type: none"> Estágio oferecido pela empresa parceira para os membros da equipe, relacionado ou não ao projeto [equipe]. 		<ul style="list-style-type: none"> Oportunidades profissionais e <i>networking</i> para os envolvidos no projeto [equipe]; Aporte financeiro da empresa parceira no projeto [aplicação].
	Baixa		<ul style="list-style-type: none"> Lançamento de uma nova versão da aplicação e adição de features [aplicação]; Retorno financeiro para empresa parceira e equipe [equipe]. 	<ul style="list-style-type: none"> Impacto positivo sobre outras empresas e públicos não planejados [aplicação].
	Baixo		Médio	Alto
	Impacto			

Fonte: Os autores (2023)

Plano de ação

A seguir serão indicados possíveis planos de ação para cada risco presente na matriz de risco.

- **Aplicação sem compatibilidade global:** É possível acessar o site pelo desktop, ou também é possível habilitar o modo desktop para smartphones. Além disso, a aplicação web está desenvolvida a partir do conceito de *mobile first*, isto é, é garantida a compatibilidade com dispositivos móveis;
- **Equipe não segue o backlog:** A melhor solução é a equipe tirar um momento para conversar, e assim colocar as ideias em ordem com o que foi proposto, tudo isso sendo conduzido pelo *Scrum Master*;
- **Falta de tempo para maximizar a usabilidade:** Nesse caso, é necessário refazer a divisão de tarefas entre o grupo, para que assim seja possível comprar tudo que foi proposto;
- **Tarefas não realizadas por membros:** Nesse contexto, o melhor a se fazer é conversar com quem não entregou o que foi proposto, para que isso não se repita, e dividir a tarefa não feita entre os membros do grupo;

- **Pouco espaço no market share:** A melhor solução é tentar inovar no nosso projeto, para que assim seja possível chegar no oceano azul;
- **Alterações no design:** Partindo da necessidade de alteração no design da aplicação, o melhor a se fazer é dividir essa tarefa com o grupo para conseguir terminar essa tarefa no prazo;
- **Projeto sem padronização:** Nesse contexto, o melhor a se fazer é dividir as tarefas entre os membros da equipe para que esse problema seja resolvido o quanto antes;
- **Instabilidade na internet em algumas regiões do país:** Partindo desse fato, não há muito o que possa ser feito, visto que nossa aplicação necessita de internet para funcionar. No mais, a aplicação pode estar otimizada para baixas conectividades;
- **Bugs não identificados:** A melhor estratégia para evitar que isso aconteça é fazer uma codificação bem modularizada e com testes contínuos;
- **Defeitos críticos encontrados na homologação:** Colocar a resolução desses defeitos em prioridade na sprint em sequência. Além disso, o desenvolvimento baseado em testes contínuos podem evitar tal risco;
- **Retirada de um membro do grupo:** Refazer a divisão de tarefas, para que ninguém fique sobrecarregado;
- **Usuário não se sente confortável em disponibilizar seus dados:** Divulgar na plataforma que nosso sistema de controle e segurança de dados é confiável;
- **A solução não atende os requisitos/necessidades do projeto:** Mudar o quanto antes os desvios de ideais que tivemos durante o projeto para que os professores possam ser atendidos;
- **Entregas fora dos prazos:** Conversar com os membros do grupo para que isso não ocorra mais vezes;
- **Usuário não entende como usar a aplicação:** Utilizar a próxima sprint para resolver problemas de usabilidade.

3. Requisitos do Sistema

Este tópico está direcionado ao detalhamento dos requisitos da aplicação web *Athena*, o que inclui a descrição das personas - as quais são fundamentais para tornar a solução mais acessível e, acima de tudo, significativa para o usuário final, já que, por intermédio da persona, é possível definir quais são as prioridades do projeto - e a descrição das histórias dos usuários, parte fundamental para a priorização das funcionalidades do projeto, já que, por meio das US (*User Stories*) é possível listar as principais tarefas que o usuário precisa realizar na aplicação final.

3.1. Persona

A persona é um perfil fictício que simula usuários típicos, objetivando auxiliar os criadores a mapear as necessidades, desejos e preocupações dos seus clientes. Para a criação de uma persona, é preciso levantar algumas informações básicas sobre a mesma, a fim de fomentar empatia pelos usuários.

Desse modo, foi analisado o censo escolar de 2022 disponibilizado pelo INEP, com o intuito de criar personas que sejam as mais semelhantes possíveis aos verdadeiros usuários finais da aplicação web. A partir dessas informações, o grupo compreendeu que a maior parte dos docentes dos anos finais do ensino fundamental são mulheres com idades entre 30 e 50 anos.

Nesse sentido, para mapear as necessidades de forma mais ampla, foram criadas duas personas baseadas em dados demográficos (do censo supracitado) e em dados informados pela empresa parceiro, sobretudo a seguinte informação: a maior parte dos(as) professores(as) que conhecem a nova escola já passaram dos 35/40 anos e, por essa razão, as pessoas mais novas não têm tanta familiaridade com a organização.

Portanto, para demonstrar visualmente essas informações, as personas (Silvia Souza e Júlia Silva) possuem um relacionamento com a empresa. Tal relacionamento é baseado nos dados fornecidos pela empresa. Sendo assim, as figuras a seguir (figura 07 e figura 08) expõem os detalhes sobre cada uma das personas deste projeto.

Figura 07 - Detalhes da persona Silvia Souza



Fonte: Os autores (2023)⁷

Figura 08 - Detalhes da persona Júlia da Silva



Fonte: Os autores (2023)⁸

⁷ Foto fictícia criada por uma Inteligência Artificial: <https://this-person-does-not-exist.com/pt>

⁸ Foto fictícia criada por uma Inteligência Artificial: <https://this-person-does-not-exist.com/pt>

3.2. Histórias dos usuários (user stories)

As histórias de usuários são frases curtas e diretas que visam descrever funcionalidades a partir da perspectiva do usuário final. Elas são fundamentais para que os desenvolvedores do projeto possam compreender quais são as necessidades e desejos de seus futuros clientes e decidir testes de validação.

Ademais, as *user stories* ajudam os membros da equipe de um projeto a definir os requisitos do projeto de forma eficiente e objetiva, permitindo que eles consigam reconhecer suas prioridades por meio do ponto de vista de seus consumidores.

Em resumo, as *user stories* são essenciais para a criação de um site orientado às necessidades do usuário e que alcance a satisfação dos seus clientes. Sendo assim, a equipe PROMETHEUS utilizou essa ferramenta, visando garantir que o site ATHENA seja o mais atrativo e eficiente possível.

Portanto, a equipe fez diversas histórias de usuários, visando conectar-se mais com seus futuros clientes. Além disso, foram definidas as principais metas do grupo para o site para a realização de análises de critérios, validações e testes de aceitação dessas histórias de usuários, objetivando garantir que o produto final atenda às necessidades dos usuários finais.

Quadro 3 - User Stories US01 relacionada à usabilidade na aplicação

Número	US01		
Título	Garantia de usabilidade na plataforma		
Persona	Professor com pouco contato com recursos tecnológicos		
História	Como professor usuário que não possui familiaridade com computadores, eu quero acessar uma plataforma simples e intuitiva, para facilitar a minha utilização.		
Critérios de Aceitação	CR01 - O site deve exibir somente informações essenciais na página inicial. Validação: O site deve ter uma homepage com links para as funcionalidades fundamentais da aplicação.	CR02 - O site deve possuir um design básico e de fácil compreensão. Validação: O site deve ter informações objetivas e com poucos detalhes, visando tornar a aplicação mais intuitiva e menos confusa.	CR03 - O site deve possibilitar que os seus usuários possam alcançar seus objetivos com poucas <i>user tasks</i> . Validação: O site deve possuir no máximo 2 passos para que o usuário consiga alcançar qualquer conteúdo, para torná-lo mais acessível a todos e, assim, garantir fluidez ao site.
Testes de Aceitação	CR01 - Acessar as principais funcionalidades da aplicação. Aceitou = O professor	CR02 - As informações textuais e visuais estão claras Aceitou = O professor	CR03 - As <i>user tasks</i> do site são fáceis e rápidas. Aceitou = O professor conseguiu realizar as ações

	conseguiu acessar todas as funcionalidades que precisava Recusou = O professor não conseguiu encontrar o que procurava	conseguiu entender os elementos da aplicação com facilidade Recusou = O professor não conseguiu entender os elementos da aplicação.	na plataforma com 2 ou menos cliques Recusou = O professor não conseguiu realizar ações na plataforma de maneira intuitiva e objetiva, realizando mais de 2 cliques por tarefa.
--	---	--	--

Fonte: Os autores (2023)

Quadro 4 - User Stories US02 relacionada ao cadastro na aplicação

Número	US02		
Título	Cadastro no site		
Persona	Professor usuário		
História	Como professor usuário, quero me cadastrar na plataforma, para ter acesso aos dados da(s) minha(s) turma(s).		
Critérios de Aceitação	CR01 - O site deve ter uma tela de cadastro Validação: O site precisa possuir uma tela, na qual o professor pode decidir suas credenciais para entrar futuramente	CR02 - O site deve ter uma tela de <i>login</i> . Validação: O site precisa possuir uma tela, na qual o usuário deve digitar seu <i>email</i> e senha, para que ele possa ter acesso à sua conta de modo pessoal e confidencial.	CR03 - O site deve possibilitar a recuperação das credenciais. Validação: O site deve ter uma tela para colocar o seu email em caso de o usuário esquecer a sua senha
Testes de Aceitação	CR01 - Cadastro de seu <i>email</i> e criação de senha: Aceitou = O professor conseguiu cadastrar seu email e senha Recusou = O professor não conseguiu cadastrar seu email e senha	CR02 - Acessar a sua conta por meio de suas credenciais: Aceitou = O professor conseguiu acessar sua conta; Recusou = O professor não conseguiu entrar em sua conta.	CR03 - Envio do email para a redefinição de senha para o usuário: Aceitou = O professor conseguiu redefinir a sua senha; Recusou = O professor não conseguiu redefinir a senha.

Fonte: Os autores (2023)

Quadro 5 - User Stories US03 relacionada aos planos de aula na aplicação

Número	US03		
Título	Acesso aos planos de aula da NOVA ESCOLA		
Persona	Professor usuário		
História	Como professor usuário, quero conhecer a defasagem dos meus alunos , para ter acesso a um bom plano de aula.		
Critérios de Aceitação	CR01 - O site deve possuir um repositório de dados dos alunos.	CR02 - o site deve ter gráficos para exibir a defasagem dos	CR03 - O site deve disponibilizar planos de aula. Validação: O site deve, por

	<p>Validação: É necessário que o site possa realizar a coleta dos dados dos estudantes, por meio de um questionário realizado pelos professores a partir dos resultados de seus alunos.</p>	<p>alunos.</p> <p>Validação: O site deve analisar os dados disponibilizados pelos professores e, então, gerar gráficos que auxiliem os usuários a entenderem as defasagens de sua turma.</p>	<p>meio das informações cadastradas pelos professores e dos resultados dos gráficos, apresentar aos usuários links da NOVA ESCOLA com os planos de aulas mais adequados para o perfil de sua turma.</p>
Testes de Aceitação	<p>CR01 - O site coleta dados dos alunos: Aceitou = O professor conseguiu acessar as informações e resultados de cada aluno; Recusou = O professor não consegue acessar as informações e resultados de cada aluno.</p>	<p>CR02 - Acessar a informação por meio de gráficos: Aceitou = O professor conseguiu entender a informação por meio dos gráficos; Recusou = O professor não conseguiu compreender as defasagens de sua(s) turma(s).</p>	<p>CR03 - Acessar os planos de aula adequados para sua turma: Aceitou = O professor conseguiu descobrir os planos de aula adequados para sua turma; Recusou = O professor não conseguiu descobrir os planos de aula adequados para sua turma.</p>

Fonte: Os autores (2023)

Sendo assim, os quadros acima (3, 4 e 5) demonstram as histórias de usuário US01, US02 e US03 de forma detalhada, considerando a persona, os critérios de aceitação e os testes de aceitação para cada uma, algo que ajuda na implementação, de fato, dessas histórias de usuário na *Athena*. Ademais, visando aprimorar o trabalho, foram criadas outras histórias que são fundamentais para a aplicação web, mas que não carecem de um detalhamento de aceitação, como as anteriores. Tais histórias estão listadas a seguir.

- Como professor usuário, quero criar uma conta na plataforma, para poder acessar todas as funcionalidades.
- Como professor usuário, quero me cadastrar na plataforma através da minha conta do Google, para facilitar o meu acesso.
- Como professor usuário, quero realizar o login na plataforma, para poder visualizar os dados da minha turma.
- Como professor usuário, quero ter a possibilidade de recuperar as minhas credenciais, para o caso de eu esquecê-las.

- Como professor usuário, quero ver as médias de cada classe para compreender que turmas precisam de mais atenção.

4. Arquitetura do Sistema

A arquitetura do sistema refere-se à estrutura, comportamento e organizações de um sistema de software ou hardware. Além disso, ela descreve a forma como um sistema pode trabalhar para executar as tarefas e alcançar um objetivo.

O projeto Athena possui uma Arquitetura de Software baseada no padrão MVC, o qual divide a aplicação em três camadas: interação do usuário (view), manipulação dos dados (model) e a camada de controle (controller). Sendo assim, cada camada possui uma função diferente, sendo elas:

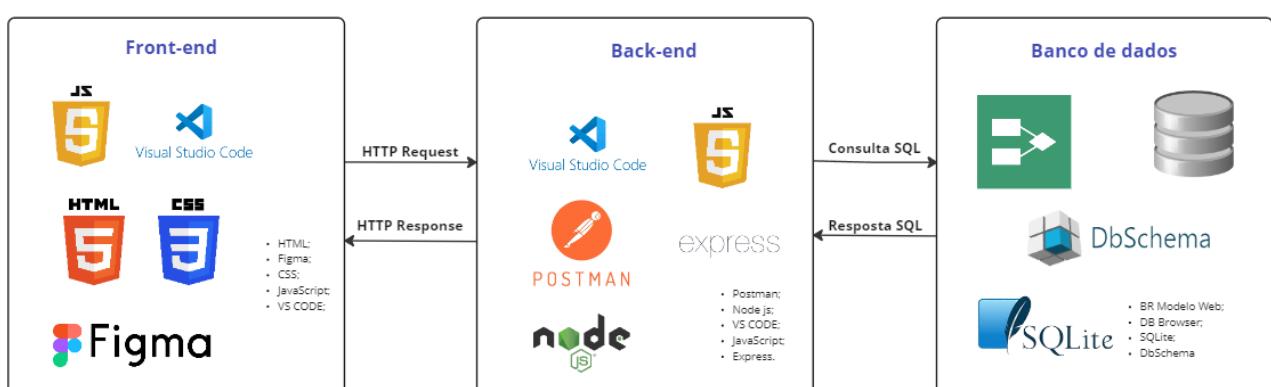
- **Model:** acesso e manipulação dos dados na aplicação.
- **View:** responsável pela interface que será mostrada, apresentando as informações do model para o usuário.
- **Controller:** responsável por conectar as camadas de model e view.

Ademais, esse modelo permite que o código seja mais fácil de ser mantido e escalável, já que as responsabilidades estão totalmente separadas, como também deixa o código mais comprehensível e de fácil reutilização.

4.1. Módulos do Sistema e Visão Geral (Big Picture)

Para o desenvolvimento do projeto foram utilizadas algumas tecnologias para a construção do front-end, back-end e do banco de dados, as quais estão representadas por meio dos ícones no fluxograma na figura 09.

Figura 09 - Fluxograma com as tecnologias utilizadas



Fonte: Os autores (2023)⁹

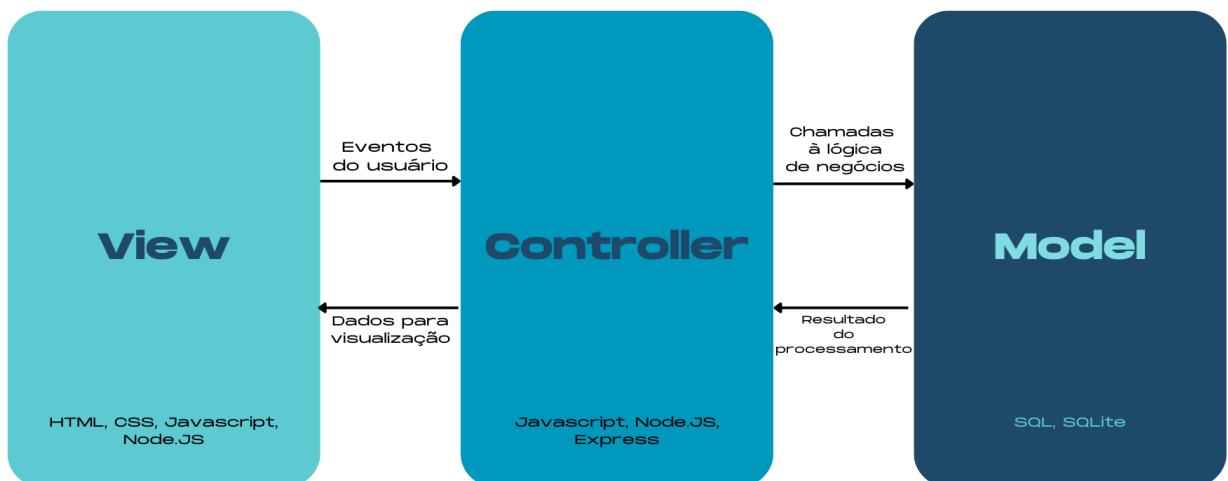
⁹ Foto criada pelo grupo por meio do aplicativo MIRO:

https://miro.com/welcomeonboard/YkgvT0J5UzVzVIFHdVVueGdzQ1hPVnlmNDh0T0NyamdDclB2SzUyWXp2Z0dxZXZGd2VwZDRGRXYxdXhZUW5XSnwzNDU4NzY0NTUxODk4NjAyNTExfDI=?share_link_id=607752422526

Ademais, é válido ressaltar que uma descrição mais detalhada sobre cada uma das tecnologias utilizadas é encontrada na próxima seção (4.2) e que HTTP Request e Response corresponde às requisições (pedidos) feitas pelo front-end (cliente) para adquirir informações e que se tudo estiver correto, o servidor responderá utilizando o mesmo protocolo. Como também, consulta e resposta SQL são consultas que o back-end realiza no banco de dados do projeto para obter os dados que ele precisar utilizar durante o funcionamento do site.

Além disso, visando exemplificar a implementação do padrão de projeto utilizado, o MVC, foi criada a seguinte representação visual sobre as três camadas deste padrão, considerando as tecnologias que atuam em cada uma delas. Para tal, a figura abaixo exemplifica essas camadas.

Figura 10 - Padrão MVC implementado no projeto



Fonte: Os autores (2023)

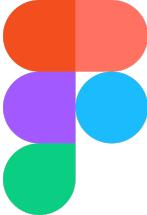
Portanto, é possível analisar que cada camada tem a sua respectiva funcionalidade, de modo que na view ficam todos os processos que possibilitam a visualização e a interação dos elementos gráficos, enquanto no controller fica toda a lógica que associa cada requisição ao respectivo modelo, o qual, por sua vez, retorna os dados para o usuário.

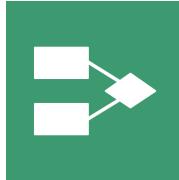
4.2. Tecnologias Utilizadas

A tabela a seguir apresenta as tecnologias utilizadas pelo grupo Prometheus durante o desenvolvimento do projeto Athena, o link para acessá-las, suas funções, o porquê de termos utilizado cada uma e as suas versões.

Quadro 06 - Informações sobre as tecnologias utilizadas

Nome	Link	Função	Por que usamos?	Versão
JavaScript 	https://developer.mozilla.org/en-US/docs/Web/JavaScript	JavaScript é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma.	O JavaScript é a principal linguagem de programação utilizada na criação do site.	ECMAScript 2022
HTML 	https://developer.mozilla.org/pt-BR/docs/Web/HTML	A Linguagem de Marcação de Hipertexto (HTML) é uma linguagem de marcação utilizada na construção de páginas na Web. Documentos HTML podem ser interpretados por navegadores.	O HTML será utilizado para a formatação dos textos.	HTML5
CSS 	https://developer.mozilla.org/en-US/docs/Web/CSS	Cascading Style Sheets (CSS) é um mecanismo para adicionar estilos a uma página web, aplicado diretamente nas tags HTML ou ficando contido dentro das tags <style>.	O CSS será utilizado para o design das páginas Web.	CSS3
VS Code 	https://code.visualstudio.com/brand	O Visual Studio Code é um editor de código-fonte, o qual inclui suporte para depuração, controle de versionamento Git incorporado, realce de sintaxe,	O VS Code foi utilizado pelo grupo como a principal ferramenta de criação e edição de códigos para o projeto.	1.78

		complementação inteligente de código, snippets e refatoração de código.		
Figma 	https://www.figma.com/	O Figma é uma ferramenta utilizada para editar gráficos vetoriais e criar protótipos de projetos de design. Ele é especialmente voltado para a utilização em navegadores web, mas também oferece ferramentas offline para aplicativos desktop compatíveis com GNU/Linux, macOS e Windows.	Utilizamos o Figma para a criação do wireframe versão mobile e do wireframe versão desktop.	88.1.0
Postman 	https://www.postman.com/downloads/	O Postman é uma plataforma de API (Application Programming Interface (Interface de Programação de Aplicação)), na qual possui a função de os desenvolvedores projetarem, construirão, testarem e iterarem suas APIs.	O Postman é usado pelo grupo para projetarmos e testarmos as APIs.	5.5.6
Node JS	https://nodejs.org/	O Node JS é um software de código aberto, multiplataforma,	O Node JS é utilizado para rodar o servidor fora de um navegador	18.16.0

		baseado no interpretador V8 do Google e que permite a execução de códigos JavaScript fora de um navegador web.	web.	
Express 	https://expressjs.com/pt-br/	O Express.js é um framework para Node.js que fornece recursos mínimos para construção de servidores web.	O Express foi usado para rodar o servidor.	4.18.2
BR Modelo Web 	https://www.brmodeloweb.com/	O BR Modelo Web é uma ferramenta criada para realizar a modelagem de banco de dados conceitual e lógico.	O BR Modelo Web é usado pelo grupo para a construção e manutenção do modelo conceitual de nosso banco de dados.	v3.31
DB Browser 	https://sqlitebrowser.org/	O DB Bowser é uma ferramenta visual para manipulação de base de dados SQLite	O DB Browser foi utilizado para a criação dos modelos físico e lógico do banco de dados.	3.12.2
DBSchema 	https://dbschema.com/	O DBSchema é uma ferramenta visual de gerenciamento e banco de dados que permite aos usuários projetar, documentar e gerenciar bancos em uma interface gráfica do usuário.	O DBSchema foi utilizado para construção e visualização do modelo lógico a partir do código.	9.3.0
SQLite 	https://sqlite.org/	O SQLite é uma biblioteca em C que oferece uma implementação	O SQLite foi usado para fazer o banco de dados.	2.3.0

		de banco de dados SQL embutida. Ao utilizar essa biblioteca em programas, é possível ter acesso a bancos de dados SQL sem a necessidade de executar um processo separado de Sistema Gerenciador de Banco de Dados (SGBD).		
Canva 	https://www.canva.com/pt_br/	O Canva é uma ferramenta de design gráfico utilizada para a criação de diversos conteúdos visuais.	O Canva foi a plataforma digital escolhida pelo grupo para a criação dos slides e das logos do projeto.	1.65.0
GitHub 	https://github.com/	O GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git.	O GitHub foi usado para compartilhar e receber as atualizações do projeto.	3.0
Slack 	https://slack.com/intl/pt-br	O Slack é uma plataforma de comunicação profissionais e organizacionais.	O Slack é utilizado como principal meio digital de comunicação do grupo.	4.32.122
Trello 	https://trello.com/pt-BR	O Trello é uma ferramenta de gerenciamento de projetos.	O Trello é utilizado para o planejamento e organização das tarefas do grupo.	2.12.3
Miro	https://www.miro.com	O Miro é uma plataforma de colaboração	O Miro é a plataforma utilizada para fazer	0.7.31

		digital para facilitar a comunicação das equipes a gerenciar seus projetos.	as atividades da <i>Sprint Review</i> , <i>Retrospective</i> e <i>Sprint Planning</i> , como também algumas atividades de entregas do artefato, como o fluxograma de tecnologias.	
Google Docs 	https://docs.google.com/	O Google Docs é um editor de texto que faz parte de um pacote de aplicativos Google Workspace.	O Google Docs é utilizado para realizar as documentações do trabalho da equipe, como o WAD.	0.10
Google Sheets 	https://docs.google.com/spreadsheets/create	O Google Sheets é um programa de planilhas do Google.	O Google Sheets é a ferramenta utilizada para fazer as planilhas que o grupo utiliza no projeto.	1.23.162.03 .90
Jamboard 	https://jamboard.google.com/	O Jamboard é um quadro branco digital, interativo e colaborativo.	O Jamboard é a ferramenta que o grupo utiliza para fazer as atividades da <i>Sprint Review</i> , <i>Retrospective</i> e <i>Sprint Planning</i> , tais como a matriz CSD.	0.2

Fonte: Os autores (2023)

5. UX e UI Design

Este tópico está direcionado ao detalhamento dos elementos de UX (*User Experience*) e UI (*User Interface*) da aplicação web *Athena*, o que inclui a exibição dos wireframes e do guia de estilos usados para a prototipação e implementação da solução.

5.1. Wireframe

A priori, é válido enfatizar que um *wireframe* é um esboço, um rascunho simples das telas de uma aplicação web, cujo objetivo é validar a ideia da estrutura das páginas, já que não apresenta cor, fontes, ícones nem imagens. Nesse viés, essa ferramenta serve como validação de um protótipo mínimo mostrado ao cliente, de modo que seja possível alinhar as expectativas do cliente com o que será desenvolvido e auxiliar o time de design na diagramação da identidade visual do site. Portanto, com esses objetivos, foram criados esboços para cada tela da aplicação *Athena*, cujas imagens - exceto as complexas - estão detalhadas abaixo.

Figura 11 - Tela de login (versão mobile)



Fonte: Os autores (2023)

Figura 12 - Tela de cadastro (versão mobile)



Fonte: Os autores (2023)

Figura 13 - Tela de perfil (versão mobile)



Fonte: Os autores (2023)

Figura 14 - Tela inicial - turmas (versão mobile)



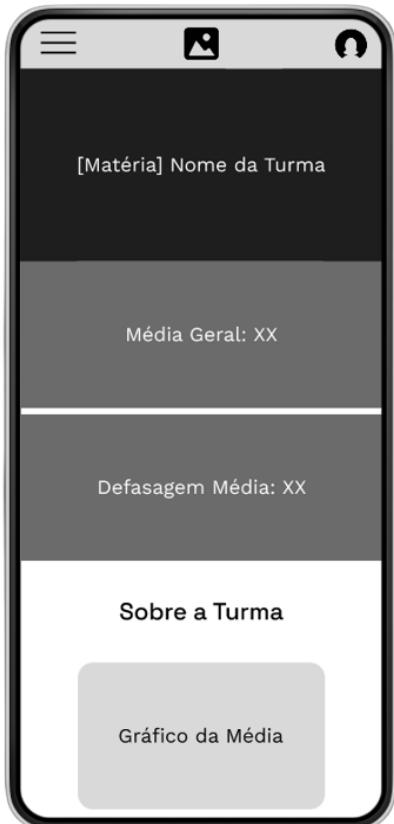
Fonte: Os autores (2023)

Figura 15 - Tela inicial - alunos (versão mobile)



Fonte: Os autores (2023)

Figura 16 - Tela da turma - parte 1 (versão mobile)



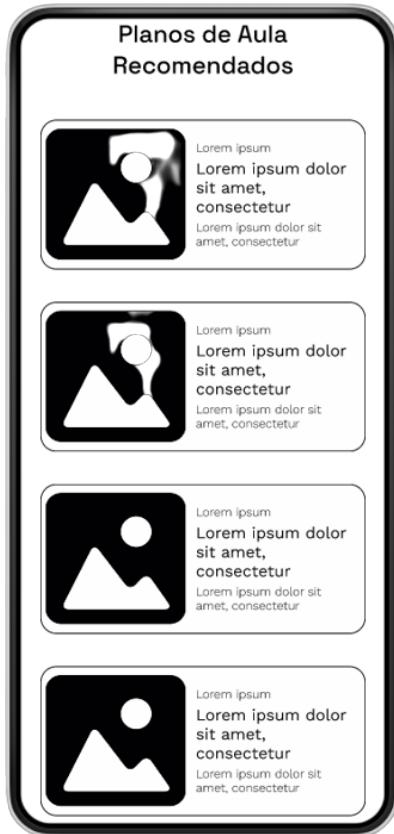
Fonte: Os autores (2023)

Figura 17 - Tela da turma - parte 2 (versão mobile)



Fonte: Os autores (2023)

Figura 18 - Tela da turma - parte 3 (versão mobile)



Fonte: Os autores (2023)

Figura 19 - Tela do cabeçalho - turmas (versão mobile)



Fonte: Os autores (2023)

Figura 20 - Tela do cabeçalho - avaliações (versão mobile)



Fonte: Os autores (2023)

Figura 21 - Tela do cabeçalho - alunos (versão mobile)



Fonte: Os autores (2023)

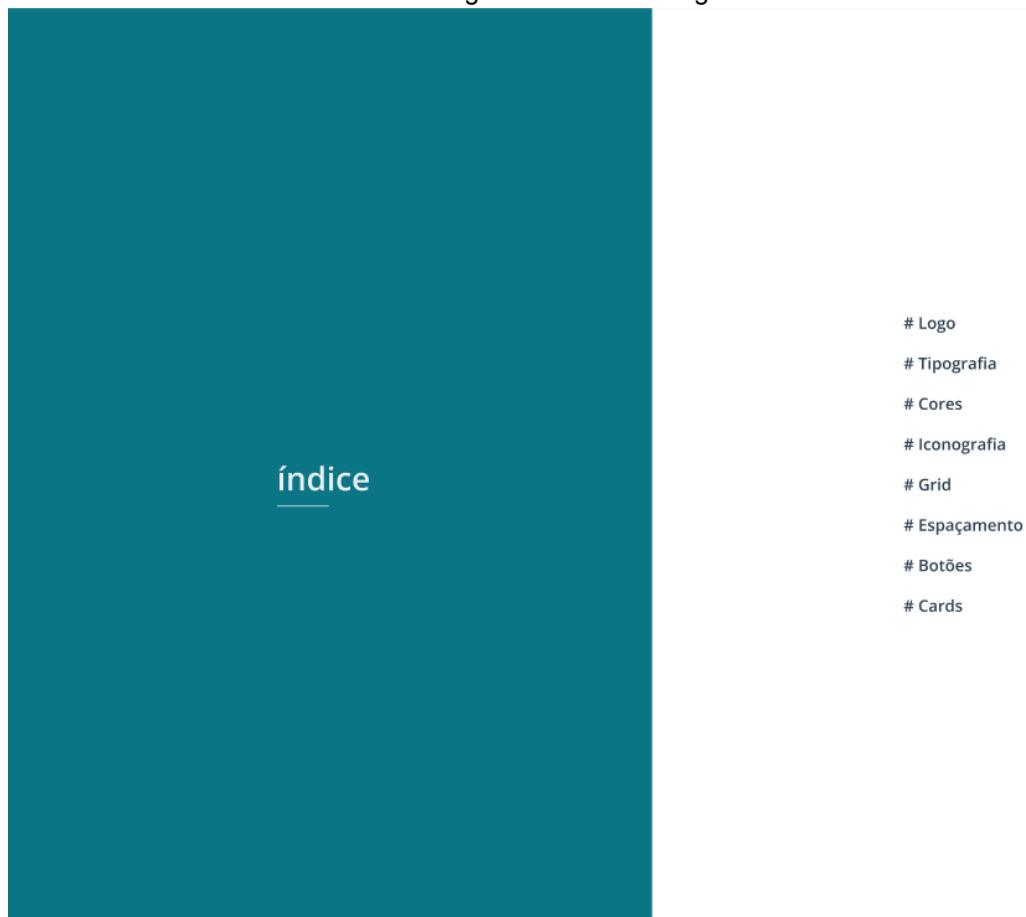
Outras muitas telas foram criadas, considerando todas as funcionalidades do projeto, como uma tela específica para detalhar os alunos, para cadastrar avaliações, entre outras. Para acessar o wireframe completo das telas da aplicação web *Athena* versões mobile e desktop, basta acessar este link: <https://www.figma.com/file/GvDKJ4P5CwUx9hepNtXcem/Aplica%C3%A7%C3%A3o-Web?node-id=16%3A266&t=iU5AJW7IKVWP7qdx-1>

5.2. Design de Interface - Guia de Estilos

O Guia de estilo consiste em um documento que possui todas as informações que são padrões do design do projeto, tais como logo, tipografia, cores, iconografia, grid, espaçamento, botões e cards. Ademais, ele possui a principal função de guiar a evolução do front-end da aplicação, visto que facilita o trabalho dos desenvolvedores, deixando mais prático para consultar cada elemento do design.

Sendo assim, o guia de estilo da aplicação web *Athena* conta com os tópicos demonstrados na figura a seguir (figura 22).

Figura 22 - Índice do guia de estilo

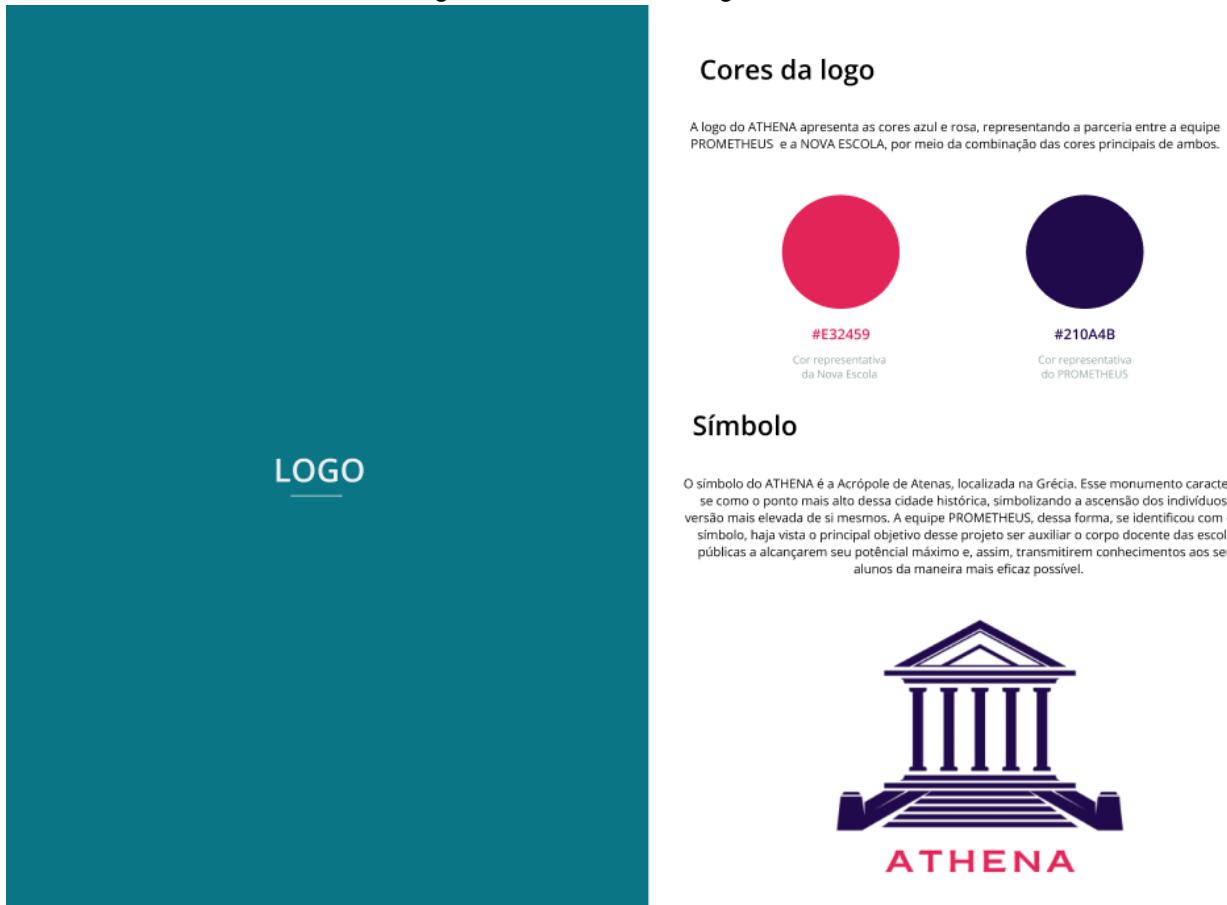


Fonte: Os autores (2023)

Nesse sentido, o guia de estilos da Athena apresenta, como demonstrado na figura supracitada, a explicação acerca da logo da aplicação, o detalhamento da tipografia utilizada na plataforma, a paleta de cores, os ícones, o sistema de grid, a demonstração visual dos espaçamentos utilizados na aplicação, os botões e os modelos de *cards* presentes no front-end da aplicação web.

Além disso, cada um dos tópicos estão dispostos conforme demonstra a imagem subsequente, apresentando o nome do elemento e as suas respectivas informações, e outras características extras, como detalhes ou outras informações que são relevantes para o leitor do guia de estilos. Sendo assim, a figura abaixo demonstra como encontra-se a página sobre a logo da Athena.

Figura 23 - Detalhes da logo do Athena



Fonte: Os autores (2023)

Por fim, o guia de estilos está disponível publicamente através deste link: [https://www.figma.com/proto/ZVYzFLbir0ykaKluJUVdTR/UI-Style-Guide-\(Community\)?type=design&node-id=1-108&scaling=contain&page-id=0%3A1](https://www.figma.com/proto/ZVYzFLbir0ykaKluJUVdTR/UI-Style-Guide-(Community)?type=design&node-id=1-108&scaling=contain&page-id=0%3A1). Ao acessar, é possível navegar por ele através das setas inferiores presentes na página.

6. Projeto de Banco de Dados

Para que o sistema web desenvolvido neste projeto atenda as necessidades da organização parceira de projeto, fez-se necessário modelar o banco de dados seguindo uma metodologia que assegura o desenvolvimento eficaz e coerente com os requisitos do projeto. Sendo assim, tal modelagem é baseada em três níveis: modelagem conceitual, modelagem lógica e modelagem física.

Nesse viés, o modelo conceitual elucida como os dados serão armazenados e como estarão relacionados, independentemente da implementação e da viabilidade tecnológica para tal realização, já que, nessa etapa do processo de criação da base de dados, não são considerados os limites ou as restrições impostas pela tecnologia final. Sob essa ótica, a técnica de modelagem conceitual amplamente utilizada é a entidade-relacionamento, a qual apresenta o modelo em forma de diagrama, com elementos gráficos para descrever as entidades, atributos e relacionamentos.

Ademais, diferentemente do modelo conceitual, o modelo lógico já considera as peculiaridades e limitações da tecnologia que será utilizada na implementação do banco de dados, como SQL Server, Oracle e MySQL, por exemplo. Sob essa perspectiva, esse modelo representa a estrutura lógica do armazenamento dos dados, considerando cada entidade como uma tabela, os atributos como colunas e as relações como chaves (primárias e estrangeiras). Essa modelagem está mais próxima da implementação, de fato, no SGBD, mas só é concebida a partir da modelagem conceitual, embora haja discordância nesse quesito.

Outrossim, o modelo físico é a parte da implementação da estrutura da base de dados (a qual foi criada a partir do modelo lógico), considerando os tipos dos dados que serão armazenados (textos, números, datas, entre outros), e outros componentes da estrutura física do banco (tabelas, campos, índices, etc). Esse modelo baseia-se em SQL, linguagem utilizada para gerenciar bancos relacionais, por meio de instruções que criarão, de fato, a base de dados. Por fim, o modelo físico também considera requisitos não funcionais, visando a otimização, a segurança e outros aspectos definidos em concordância com o cliente.

Sendo assim, as seções subsequentes (**6.1**, **6.2**, e **6.3**) tratam, respectivamente, sobre cada uma das três modelagens supracitadas, cuja temática é o projeto do banco de dados Athena DB.

6.1. Modelo Conceitual

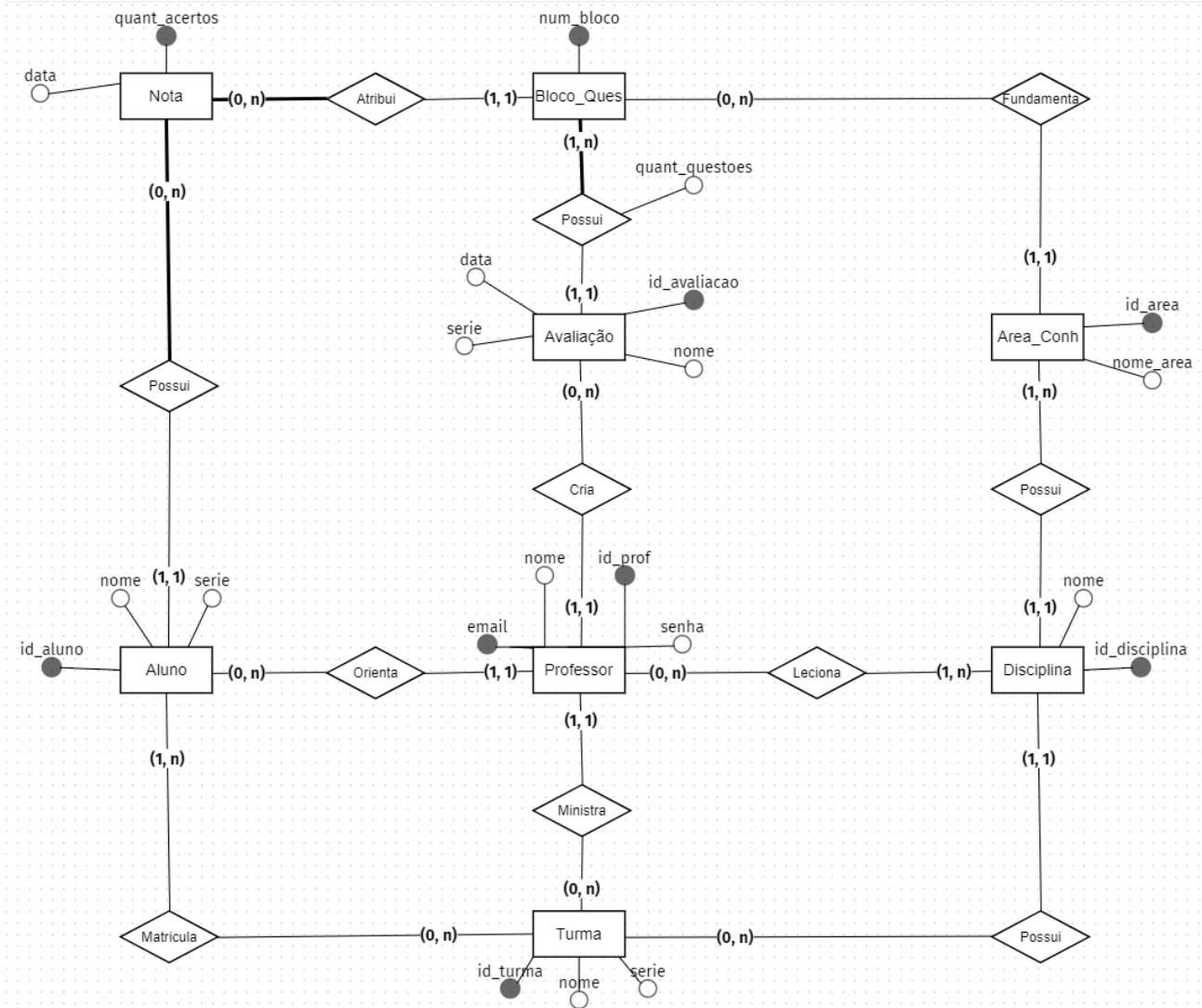
Para apresentar em formato de diagrama o modelo conceitual do banco de dados *Athena*, foi utilizada a plataforma [BR Modelo](#), disponível gratuitamente na web, a qual possibilitou a criação do Diagrama Entidade-Relacionamento (DER), que é utilizado para descrever os objetos do mundo real através de entidades de forma gráfica.

Nesse viés, as entidades são interpretações de objetos reais, podendo ser fortes - quando são independentes -, fracas - quando dependem da existência de outra entidade - ou associativas - quando existe a necessidade de associar uma entidade a um relacionamento.

Além disso, as entidades possuem atributos, que são, em linhas gerais, as características e propriedades de cada entidade. No caso do *Athena DB*, a equipe utilizou dois tipos de atributos: os simples, aqueles que não são divisíveis, e os fortes, aqueles atributos simples que identificam unicamente uma entidade.

Ademais, foi utilizado o conceito de relacionamento, que trata-se da associação entre entidades, os quais podem possuir atributos e cardinalidades. Nesse quesito, existem três principais tipos de relacionamentos: muitos para muitos, um para um e um para muitos. Todos, com exceção do segundo, estão presentes no banco de dados deste projeto.

Figura 22 - Modelo conceitual do banco de dados Athena



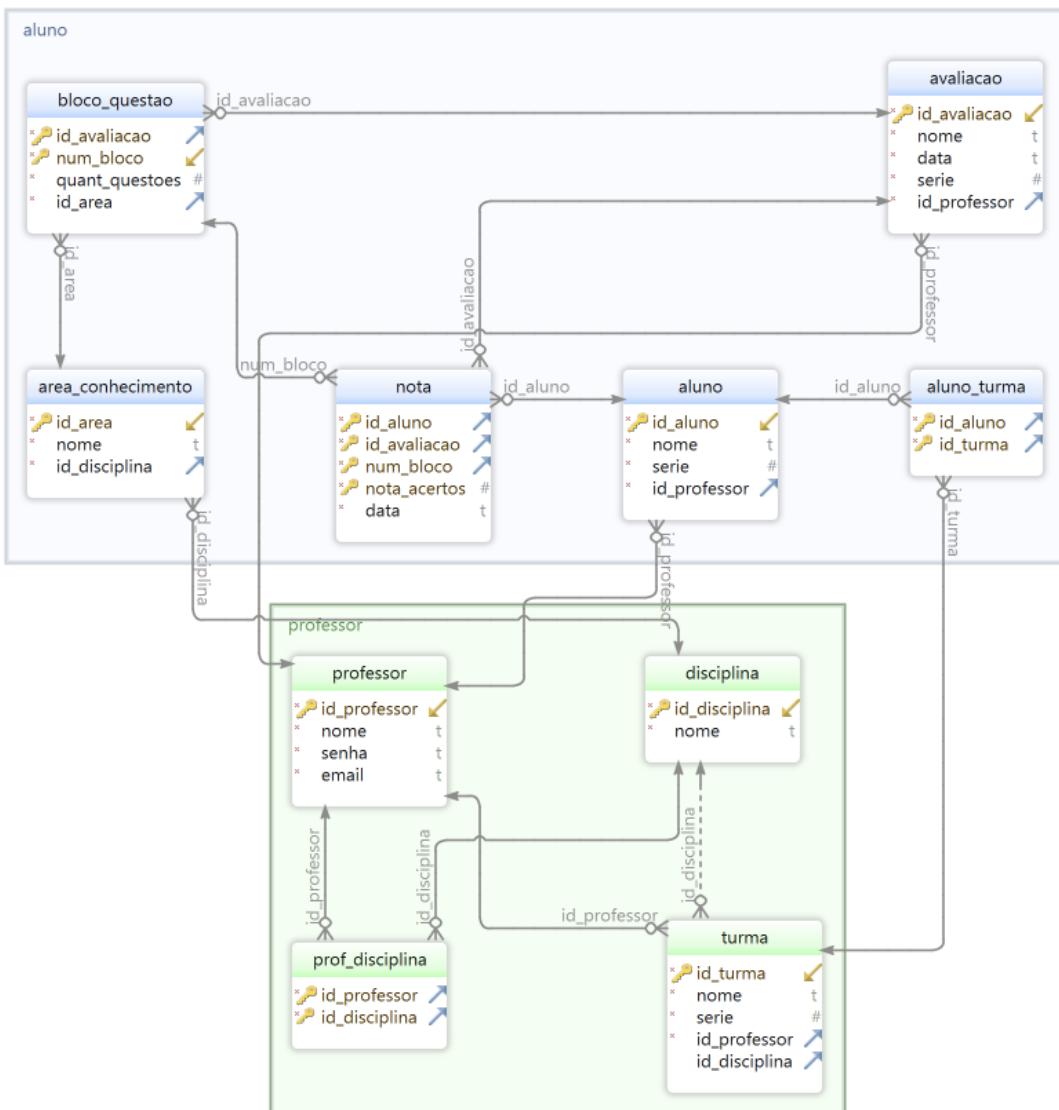
Fonte: Os autores (2023)

Conforme demonstrado na figura acima, o *Athena DB*, na versão V0.2.0, possui 8 entidades, sendo 2 fracas (*Bloco_Ques* e *Nota*) e 6 fortes (*Avaliação*, *Aluno*, *Area_Conh*, *Disciplina*, *Professor* e *Turma*). Além disso, todas as entidades estão com as suas respectivas cardinalidades explicadas, como o caso das entidades *Turma* e *Professor*, cujo relacionamento é um para muitos (1:n), já que um professor pode ter várias turmas (n), mas uma turma só está vinculada à um único professor (1).

6.2. Modelo Lógico

Logo após a construção do modelo conceitual, o modelo lógico do banco de dados da aplicação foi construído, visando demonstrar cada entidade como uma tabela, além das chaves primárias - provenientes dos atributos fortes do modelo conceitual - e das chaves estrangeiras - provenientes dos relacionamentos entre as entidades.

Figura 23 - Modelo lógico do banco de dados Athena



Fonte: Os autores (2023)

Sob essa perspectiva, a figura acima evidencia a estrutura lógica do banco de dados, cuja principal diferença entre o modelo conceitual é que, a partir dos relacionamentos n:n (muitos para muitos), sugeriram duas novas tabelas - aluno_turma e prof_disciplina, de modo que seja possível mesclar dados das quatro tabelas envolvidas nesses relacionamentos.

6.3. Modelo Físico

Considerando que os outros dois modelos foram criados, conforme supracitado nos tópicos anteriores, foi utilizado o software DB Browser, por meio da linguagem SQL, para criar modelar fisicamente o banco de dados da aplicação web.

Figura 24 - Modelo físico do banco de dados *Athena*

```
CREATE TABLE IF NOT EXISTS "professor" (
    "id_professor" INTEGER NOT NULL,
    "nome" TEXT NOT NULL,
    "senha" TEXT NOT NULL,
    "email" TEXT NOT NULL,
    PRIMARY KEY("id_professor" AUTOINCREMENT)
);
CREATE TABLE IF NOT EXISTS "disciplina" (
    "id_disciplina" INTEGER NOT NULL,
    "nome" TEXT NOT NULL,
    PRIMARY KEY("id_disciplina" AUTOINCREMENT)
);
CREATE TABLE IF NOT EXISTS "avaliacao" (
    "id_avaliacao" INTEGER NOT NULL,
    "nome" TEXT NOT NULL,
    "data" TEXT NOT NULL,
    "serie" INTEGER NOT NULL,
    "id_professor" INTEGER NOT NULL,
    PRIMARY KEY("id_avaliacao" AUTOINCREMENT),
    FOREIGN KEY("id_professor") REFERENCES "professor"( "id_professor" ) ON DELETE CASCADE ON UPDATE
    CASCADE
);
CREATE TABLE IF NOT EXISTS "area_conhecimento" (
    "id_area" INTEGER NOT NULL,
    "nome" TEXT NOT NULL,
    "id_disciplina" INTEGER NOT NULL,
    PRIMARY KEY("id_area" AUTOINCREMENT),
    FOREIGN KEY("id_disciplina") REFERENCES "disciplina"( "id_disciplina" ) ON DELETE CASCADE ON UPDATE
    CASCADE
);
CREATE TABLE IF NOT EXISTS "bloco_questao" (
    "id_avaliacao" INTEGER NOT NULL,
    "num_bloco" INTEGER NOT NULL,
    "quant_questoes" INTEGER NOT NULL,
    "id_area" INTEGER NOT NULL,
    PRIMARY KEY("id_avaliacao", "num_bloco"),
    FOREIGN KEY("id_avaliacao") REFERENCES "avaliacao"( "id_avaliacao" ),
    FOREIGN KEY("id_area") REFERENCES "area_conhecimento"( "id_area" ) ON DELETE CASCADE ON UPDATE
    CASCADE
);
CREATE TABLE IF NOT EXISTS "prof_disciplina" (
    "id_professor" INTEGER NOT NULL,
    "id_disciplina" INTEGER NOT NULL,
    PRIMARY KEY("id_professor", "id_disciplina"),
    FOREIGN KEY("id_disciplina") REFERENCES "disciplina"( "id_disciplina" ) ON DELETE CASCADE ON UPDATE
    CASCADE,
    FOREIGN KEY("id_professor") REFERENCES "professor"( "id_professor" ) ON DELETE CASCADE ON UPDATE
    CASCADE
);
CREATE TABLE IF NOT EXISTS "nota" (
    "id_aluno" INTEGER NOT NULL,
    "id_avaliacao" INTEGER NOT NULL,
    "num_bloco" INTEGER NOT NULL,
    "nota_acertos" INTEGER NOT NULL,
    "data" TEXT NOT NULL,
    PRIMARY KEY("id_aluno", "id_avaliacao", "num_bloco", "nota_acertos"),
    FOREIGN KEY("id_avaliacao") REFERENCES "avaliacao"( "id_avaliacao" ) ON DELETE CASCADE ON UPDATE
    CASCADE,
    FOREIGN KEY("num_bloco") REFERENCES "bloco_questao"( "num_bloco" ) ON DELETE CASCADE ON UPDATE
    CASCADE,
    FOREIGN KEY("id_aluno") REFERENCES "aluno"( "id_aluno" ) ON DELETE CASCADE ON UPDATE CASCADE
);
CREATE TABLE IF NOT EXISTS "aluno" (
    "id_aluno" INTEGER NOT NULL,
    "nome" TEXT NOT NULL,
    "serie" INTEGER NOT NULL,
    "id_professor" INTEGER NOT NULL,
    FOREIGN KEY("id_professor") REFERENCES "professor"( "id_professor" ) ON DELETE CASCADE ON UPDATE
    CASCADE,
    PRIMARY KEY("id_aluno" AUTOINCREMENT)
);
CREATE TABLE IF NOT EXISTS "aluno_turma" (
    "id_aluno" INTEGER NOT NULL,
    "id_turma" INTEGER NOT NULL,
    FOREIGN KEY("id_turma") REFERENCES "turma"( "id_turma" ) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY("id_aluno") REFERENCES "aluno"( "id_aluno" ) ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY("id_aluno", "id_turma")
);
CREATE TABLE IF NOT EXISTS "turma" (
    "id_turma" INTEGER NOT NULL,
    "nome" TEXT NOT NULL,
    "serie" INTEGER NOT NULL,
    "id_professor" INTEGER NOT NULL,
    "id_disciplina" INTEGER,
    FOREIGN KEY("id_professor") REFERENCES "professor"( "id_professor" ) ON DELETE CASCADE ON UPDATE
    CASCADE,
    FOREIGN KEY("id_disciplina") REFERENCES "disciplina"( "id_disciplina" ) ON DELETE CASCADE ON UPDATE
    CASCADE,
    PRIMARY KEY("id_turma" AUTOINCREMENT)
);
```

Fonte: Os autores (2023)

De acordo com a imagem acima, cada tabela é criada caso não exista e, logo em seguida, os atributos, incluindo as chaves primárias e as chaves estrangeiras, são atribuídos à respectiva tabela. A título de exemplificação, a primeira instrução do código é responsável por criar a tabela professor e, logo em seguida, criar os atributos id, nome, email e senha, de modo que o id seja a chave primária da tabela, já que é único de cada professor.

7. Testes de Software

7.1. Teste de Usabilidade

Link ou imagem da tabela com dados organizados dos testes realizados

Referências

ADOBE COLOR. **Color wheel.** Disponível em: <https://color.adobe.com/pt/create/color-wheel>. Acesso em: 20 abr 2023

EVA, NOVA ESCOLA. **Informações do design.** Disponível em: <https://design.novaescola.org.br/>. Acesso em: 23 abr 2023

PORTAL MEC, GOVERNO. **Estudo exploratório sobre o professor brasileiro.** Disponível em: <http://portal.mec.gov.br/dmddocuments/estudoprofessor.pdf> . Acesso em: 25 abr 2023

INEP, GOVERNO. **Censo escolar 2022.** Disponível em: https://download.inep.gov.br/publicacoes/institucionais/estatisticas_e_indicadores/resumo_tecnico_censo_escolar_2022.pdf . Acesso em: 25 abr 2023

VEJA, ABRIL. **Como saber se o professor dá certo?** Disponível em: <https://veja.abril.com.br/educacao/como-saber-se-o-professor-da-certo/>. Acesso em: 25 abr 2023

SOU MAMÃE. **O que define um bom professor.** Disponível em: <https://soumamae.com.br/o-que-define-um-bom-professor/>. Acesso em: 25 abr 2023

Exame. **Após pandemia, brasileiros apresentam até 4 anos de defasagem educacional.** Disponível em: <https://exame.com/bussola/apos-pandemia-brasileiros-apresentam-ate-4-anos-de-defasagem-educacional/> . Acesso em: 25 abr 2023

Nova Escola. **Quem Somos?** Disponível em: <https://novaescola.org.br/quem-somos> . Acesso em: 25 abr 2023

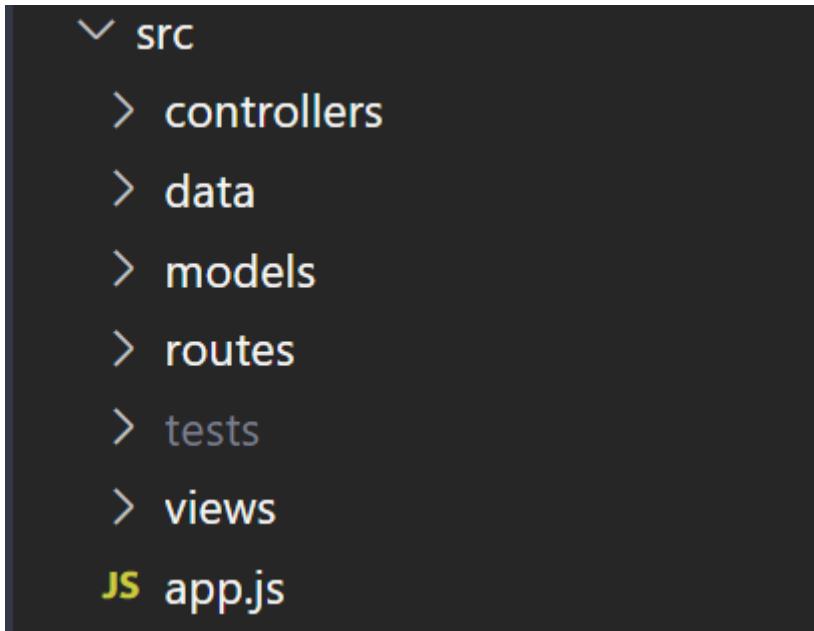
Treinaweb. **O que é MVC?** Disponível em: [https://www.treinaweb.com.br/blog/o-que-e-mvc#:~:text=O%20MVC%20sugere%20uma%20maneira.camada%20de%20controle%20\(controller\)](https://www.treinaweb.com.br/blog/o-que-e-mvc#:~:text=O%20MVC%20sugere%20uma%20maneira.camada%20de%20controle%20(controller)). Acesso em: 11 maio 2023

Apêndice A - Relatório de Criação e Documentação dos *Endpoints*

A priori, um *endpoint* é o local onde a requisição da API é atendida e, no caso específico deste projeto, o endpoint da API *Athena* baseia-se no modelo de projeto MVC, de modo que existe uma rota que, ao ser acionada, chama um modelo que, por sua vez, após executar uma série de consultas no banco de dados, passa os dados para a página HTML, ou para um arquivo JSON, de modo que os dados possam ser manipulados ou visualizados.

Sendo assim, para criar os endpoints para esta aplicação web, foi usada a seguinte estrutura de pastas: em “/controllers/” ficam todos os controladores - separados por tabelas do banco de dados -, em “/models/” ficam todos os modelos, em “/routes/” ficam as rotas de requisições, em “/data/” ficam os arquivos relacionados ao banco de dados e, por fim, em “/views/” ficam as páginas e os arquivos que são mostrados ao usuário final. Além disso, existe uma pasta “/tests/” onde ficam localizados os arquivos de testes locais. Quem gerencia a execução de todas essas pastas é o arquivo **app.js**, localizado logo na raiz da pasta dos códigos.

Apêndice A.1 - Estrutura de pastas dos *endpoints*



```
src
  > controllers
  > data
  > models
  > routes
  > tests
  > views
JS app.js
```

Nesse sentido, após estruturar as pastas, foram criados controllers, models e routes para cada tabela, de modo a simplificar a manutenção do código e a reutilização de métodos. Dessa forma, foi possível criar os endpoints da *Athena API* de maneira rápida,

limpa e organizada. Posto isso, o quadro a seguir mostra quais são os principais tipos dos endpoints criados.

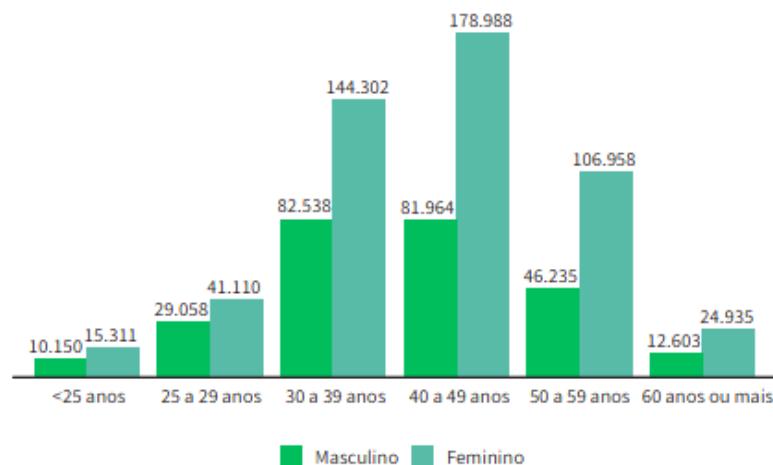
Apêndice A.2 - Quadro dos endpoints criados

Tipo	Método	Tabelas Envolvidas	Quantidade
Leitura	GET	turmas, professores, alunos, disciplinas, avaliações e áreas do conhecimento	6
Inserção	POST	turmas, professores, alunos, disciplinas, avaliações e áreas do conhecimento	6
Atualização	POST	turmas, professores, alunos, disciplinas, avaliações e áreas do conhecimento	6
Deleção	DELETE	turmas, professores, alunos, disciplinas, avaliações e áreas do conhecimento	6

Para melhor representar as funcionalidades, as características e os retornos de cada endpoint criado, foi desenvolvida a documentação de todos eles através da ferramenta *Postman*, cujo acesso pode ser realizado via este link:
<https://documenter.getpostman.com/view/26971033/2s93ebUWhW#dfeeff0e-da41-4c3f-9274-bfead53e0f8c>

Anexos

NÚMERO DE DOCENTES NOS ANOS FINAIS DO ENSINO FUNDAMENTAL, SEGUNDO A FAIXA ETÁRIA E O SEXO – BRASIL – 2022



Fonte: Censo escolar 2022, INEP