

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**JACSON SÁVIO DE SOUZA**

**DESENVOLVIMENTO DE UMA NOVA VERSÃO DO SISTEMA WEB PARA  
OFERTA E GERENCIAMENTO DE CURSOS PARA A EMPRESA  
MAXIAMBIENTAL TREINAMENTOS.**

**GUARAPUAVA**

**2025**

**JACSON SÁVIO DE SOUZA**

**DESENVOLVIMENTO DE UMA NOVA VERSÃO DO SISTEMA WEB PARA  
OFERTA E GERENCIAMENTO DE CURSOS PARA A EMPRESA  
MAXIAMBIENTAL TREINAMENTOS.**

**DEVELOPMENT OF A NEW VERSION OF THE WEB SYSTEM FOR COURSE  
OFFERING AND MANAGEMENT FOR THE COMPANY MAXIAMBIENTAL  
TREINAMENTOS.**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Dra. Renata Luiza Stange Carneiro  
Gomes

Coorientador: Dr. Diego Marczal

**GUARAPUAVA**

**2025**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**JACSON SÁVIO DE SOUZA**

**DESENVOLVIMENTO DE UMA NOVA VERSÃO DO SISTEMA WEB PARA  
OFERTA E GERENCIAMENTO DE CURSOS PARA A EMPRESA  
MAXIAMBIENTAL TREINAMENTOS.**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do  
título de Tecnólogo em Tecnologia em Sistemas  
para Internet do Curso Superior de Tecnologia  
em Sistemas para Internet da Universidade  
Tecnológica Federal do Paraná.

Data de aprovação: 30/junho/2025

---

Prof. Renata Luiza Stange Carneiro Gomes  
Doutora  
Universidade Tecnológica Federal do Paraná

---

Prof. Diego Marczal  
Doutor  
Universidade Tecnológica Federal do Paraná

---

Prof. Andres Jessé Porfirio  
Doutor  
Universidade Tecnológica Federal do Paraná

---

Prof. Emerson André Fedechen  
Doutor  
Universidade Tecnológica Federal do Paraná

**GUARAPUAVA**

**2025**

O que atenta para o ensino acha o bem, e o  
que confia no SENHOR, esse é feliz.  
Provérbios 16.20

## RESUMO

A crescente demanda por qualificação profissional e a expansão do ambiente digital impulsionam a necessidade de plataformas educacionais, justificando o desenvolvimento de soluções robustas para empresas do setor. O presente Trabalho de Conclusão de Curso descreve o desenvolvimento de uma nova versão do sistema web para a Maxiambiental Treinamentos, com o objetivo principal de melhorar a oferta e o gerenciamento de cursos, atualizando as tecnologias e buscando maior segurança e facilidade tanto para clientes quanto para administradores. Para isso, a metodologia empregada envolveu o levantamento das necessidades da empresa e a implementação de funcionalidades que abrangem desde o cadastro e acompanhamento de alunos e instrutores até a gestão de cursos e inscrições. Os resultados do desenvolvimento desta nova versão incluem a criação de um sistema web seguro e funcional, com uma arquitetura tecnológica atualizada que permite maior manutenibilidade. Assim, o sistema proporciona aos usuários acesso facilitado às informações e ferramentas de gestão para os administradores.

**Palavras-chave:** sistema web; gerenciamento de cursos; educação ambiental; maxiambiental;

## ABSTRACT

The growing demand for professional qualification and the expansion of the digital environment drive the need for educational platforms, justifying the development of robust solutions for companies in the sector. This Undergraduate Thesis describes the development of a new version of the web platform for Maxiambiental Treinamentos, with the main objective of improving course offerings and management, updating technologies, and seeking greater security and ease for both clients and administrators. To achieve this, the methodology employed involved surveying the company's needs and implementing functionalities ranging from the registration and monitoring of students and instructors to the management of courses and enrollments. The results of developing this new version include the creation of a secure and functional web system with an updated technological architecture that allows for greater maintainability. Thus, the platform provides users with facilitated access to information and management tools for administrators.

**Keywords:** web system; course management; environmental education; maxiambiental; .

## LISTA DE FIGURAS

Figura 1 – Página inicial da Maxiambiental Treinamentos . . . . .	15
Figura 2 – Responsividade Maxiambiental Treinamentos . . . . .	16
Figura 3 – Formulário Maxiambiental Treinamentos . . . . .	17
Figura 4 – História de usuário e cenários . . . . .	20
Figura 5 – GitLab Flow . . . . .	23
Figura 6 – Integração Contínua . . . . .	25
Figura 7 – Entrega Contínua . . . . .	25
Figura 8 – Prototipação - Início . . . . .	31
Figura 9 – Cursos Maxiambiental - Início . . . . .	32
Figura 10 – Cursos Maxiambiental . . . . .	32
Figura 11 – Curso Maxiambiental . . . . .	33
Figura 12 – Cadastro de usuário . . . . .	33
Figura 13 – Login . . . . .	34
Figura 14 – Formulário de inscrição . . . . .	35
Figura 15 – Formulário de inscrição - Dados do curso . . . . .	35
Figura 16 – Pagamento . . . . .	36
Figura 17 – Confirmação de Pagamento . . . . .	36
Figura 18 – Dashboard Administrador . . . . .	37
Figura 19 – Menu Administrador . . . . .	38
Figura 20 – Gerenciamento de Cursos . . . . .	38
Figura 21 – Formulário Novo Cursos . . . . .	39
Figura 22 – Gerenciamento de Turmas . . . . .	39
Figura 23 – Formulário Nova Turma . . . . .	40
Figura 24 – Gerenciamento de Instrutores . . . . .	40
Figura 25 – Formulário Novo Instrutor . . . . .	41
Figura 26 – Tabelas de usuários. . . . .	42
Figura 27 – Tabelas de administração. . . . .	42
Figura 28 – Tabelas de operações. . . . .	43
Figura 29 – Tabelas relacionadas ao institucional. . . . .	43
Figura 30 – Arquivo de CI. . . . .	47
Figura 31 – Comparação entre formulário nativo do Rails e <i>simple form</i> . . . . .	51
Figura 32 – Interface de login do administrador . . . . .	52
Figura 33 – Template de e-mail para recuperação de senha . . . . .	53
Figura 34 – Tela de solicitação de e-mail para recuperação de senha . . . . .	54
Figura 35 – Tela de redefinição de senha, com campos para nova senha e confirmação . . . . .	54
Figura 36 – Tela de gerenciamento do perfil do administrador. . . . .	55

Figura 37 – Código do editar perfil . . . . .	56
Figura 38 – Componente de <i>flash message</i> . . . . .	56
Figura 39 – Código do componente <i>flash message</i> . . . . .	57
Figura 40 – Modelo Client . . . . .	58
Figura 41 – Tela de cadastro do cliente . . . . .	58
Figura 42 – Credenciais Google . . . . .	59
Figura 43 – Implementação google no modelo Client . . . . .	59
Figura 44 – Tela de login do cliente com opção de autenticação Google. . . . .	60
Figura 45 – Tela de gerenciamento de instrutores. . . . .	61
Figura 46 – Tela de cadastro de novos instrutores. . . . .	61
Figura 47 – Tela de edição de instrutor. . . . .	63
Figura 48 – Tela de visualização do currículo do instrutor. . . . .	64
Figura 49 – Tela de listagem de cursos. . . . .	65
Figura 50 – Tela de cadastro de curso antes da seleção da imagem. . . . .	65
Figura 51 – <i>Input</i> customizado. . . . .	66
Figura 52 – <i>Input</i> customizado preechido. . . . .	66
Figura 53 – Tela de edição de curso. . . . .	67
Figura 54 – Cursos mostrados na tela inicial. . . . .	67
Figura 55 – Tela pública dos cursos. . . . .	68
Figura 56 – Tela pública de detalhes do curso. . . . .	68
Figura 57 – Tela de listagem de clientes para o administrador. . . . .	69
Figura 58 – Tela de cadastro simplificado de cliente pelo administrador. . . . .	70
Figura 59 – Tela de edição de cliente pelo administrador. . . . .	70
Figura 60 – Código: Componente Stats. . . . .	71
Figura 61 – Código: Componente Stats View. . . . .	71
Figura 62 – Código para o dashboard encapsulado em um service. . . . .	72
Figura 63 – Interface do Dashboard do Administrador. . . . .	73
Figura 64 – Tela de listagem de turmas no painel administrativo. . . . .	74
Figura 65 – Tela de cadastro de turma - Informações Básicas. . . . .	74
Figura 66 – Tela de cadastro de turma - Localização (Action Text). . . . .	75
Figura 67 – Tela de cadastro de turma - Informações do Curso (Datas, Horários, Sobre). . . . .	75
Figura 68 – Tela de cadastro de turma - Conteúdo (Programação - Action Text). . . . .	76
Figura 69 – Tela de cadastro de turma - Termos de Aceitação (Action Text). . . . .	76
Figura 70 – Tela de edição de uma turma. . . . .	77
Figura 71 – Tela de listagem de turmas na área pública. . . . .	77
Figura 72 – Tela de detalhes de uma turma na área pública. . . . .	78
Figura 73 – Passo 1 da Inscrição: Informações Pessoais. . . . .	79
Figura 74 – Passo 2 da Inscrição: Informações Adicionais e Termos. . . . .	79

Figura 75 – Passo 3 da Inscrição: Seleção da Forma de Pagamento (Simulação). . .	80
Figura 76 – Passo 4 da Inscrição: Confirmação Final. . . . .	80
Figura 77 – Tela de listagem de inscrições do cliente. . . . .	81
Figura 78 – Lógica do Form Object. . . . .	82
Figura 79 – Lógica do Form Object. . . . .	82
Figura 80 – Código: Controlador javascript do componente de busca. . . . .	85
Figura 81 – Código: Uso da <i>gem</i> Kaminari. . . . .	85
Figura 82 – Componente de busca integrado ao cabeçalho da tabela . . . . .	85
Figura 83 – Código para o Rake Populate. . . . .	86
Figura 84 – Código para a fábrica do instrutor. . . . .	87
Figura 85 – Código para o <i>populator</i> do cliente. . . . .	87
Figura 86 – Dados falsos criados via <i>populate</i> . . . . .	88
Figura 87 – Configurações PWA. . . . .	89
Figura 88 – Tags Open Graph. . . . .	90
Figura 89 – Exemplo de visualização de link compartilhado com Open Graph. . . .	91
Figura 90 – Teste do modelo <i>CLient</i> . . . . .	92
Figura 91 – Teste do componente de busca. . . . .	92
Figura 92 – Teste do controlador. . . . .	93
Figura 93 – Teste de sistema. . . . .	94
Figura 94 – Alcance Maxiambiental - Início . . . . .	103
Figura 95 – Fotos de turmas - Início . . . . .	104
Figura 96 – Depoimentos - Início . . . . .	104
Figura 97 – Cadastro de Email - Início . . . . .	105
Figura 98 – Perguntas frequentes . . . . .	105
Figura 99 – Contato Maxiambiental . . . . .	106
Figura 100 –Álbuns de fotos Maxiambiental . . . . .	106
Figura 101 –Compartilhamento de Fotos . . . . .	107
Figura 102 –Institucional Maxiambiental - Início . . . . .	107
Figura 103 –Histórico de Cursos . . . . .	108
Figura 104 –Contatos públicos . . . . .	109
Figura 105 –Gerenciamento de Álbuns . . . . .	109
Figura 106 –Formulario Novo Álbum . . . . .	110
Figura 107 –Gerenciamento de Mensagens . . . . .	110
Figura 108 –Gerenciamento do Institucional . . . . .	111

## LISTA DE ABREVIATURAS E SIGLAS

### Siglas

BDD	Desenvolvimento Orientado a Comportamento
CD	Entrega Contínua
CI	Integração Contínua
CSS	Cascading Style Sheets
ERB	Embedded Ruby
HTML	HyperText Markup Language
OG	Open Graph
PIB	Produto Interno Bruto
PWA	Progressive Web App
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience

## SUMÁRIO

<b>Lista de Figuras</b>	<b>5</b>	
<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>1.1</b>	<b>Considerações iniciais</b>	<b>12</b>
<b>1.2</b>	<b>Objetivos</b>	<b>13</b>
1.2.1	Objetivo geral	13
1.2.2	Objetivos específicos	13
<b>1.3</b>	<b>Justificativa</b>	<b>13</b>
<b>1.4</b>	<b>Estrutura do trabalho</b>	<b>14</b>
<b>2</b>	<b>MAXIAMBIENTAL TREINAMENTOS</b>	<b>15</b>
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>18</b>
<b>3.1</b>	<b>Materiais</b>	<b>18</b>
<b>3.2</b>	<b>Métodos</b>	<b>19</b>
3.2.1	Ciclo de vida ágil	19
3.2.2	Definição de histórias e cenários	20
3.2.3	Prototipação para levantar os requisitos de interface do usuário	21
3.2.4	Planejamento do projeto e monitoramento de progresso.	21
3.2.5	Organização do fluxo de desenvolvimento	22
3.2.6	Verificação e validação de software	23
3.2.7	Entrega e Implantação	24
<b>4</b>	<b>ANÁLISE E PROJETO DO SISTEMA</b>	<b>26</b>
<b>4.1</b>	<b>Histórias de usuário</b>	<b>26</b>
4.1.1	Área pública	26
4.1.2	Área do usuário	27
4.1.3	Área do administrador	29
<b>4.2</b>	<b>Prototipação de telas</b>	<b>31</b>
4.2.1	Área pública	31
4.2.2	Área do usuário	34
4.2.3	Área do Administrador	37
<b>4.3</b>	<b>Modelagem do banco de dados</b>	<b>41</b>
<b>5</b>	<b>DESENVOLVIMENTO DO SISTEMA</b>	<b>44</b>

<b>5.1</b>	<b>Escopo de desenvolvimento</b>	<b>44</b>
<b>5.2</b>	<b>Configuração do ambiente de desenvolvimento</b>	<b>45</b>
5.2.1	Criação e Configuração da Aplicação Web	45
5.2.2	Configuração de Integração Contínua (CI)	46
5.2.3	Configuração de Entrega Contínua (CD)	47
<b>5.3</b>	<b>Implementações das Funcionalidades</b>	<b>49</b>
5.3.1	Feature: Login de administrador	49
5.3.2	Feature: Recuperação de Senha do Administrador	52
5.3.3	Feature: Perfil do administrador	55
5.3.4	Feature: Autenticação dos clientes	57
5.3.5	Feature: Gerenciamento de instrutores	60
5.3.6	Feature: Gerenciamento de cursos	64
5.3.7	Feature: Gerenciamento de Clientes	69
5.3.8	Feature: Dashboard do Administrador	71
5.3.9	Feature: Gerenciamento de Turmas	73
5.3.10	Feature: Inscrição de Clientes em Turmas	78
<b>5.4</b>	<b>Atualizações e Melhorias</b>	<b>83</b>
5.4.1	Atualizações de Dependências	83
5.4.2	Implementação de Busca e Paginação	84
5.4.3	Populate	86
5.4.4	Favicon e Full Title	88
5.4.5	PWA	89
5.4.6	Open Graph	90
<b>5.5</b>	<b>Realização dos Testes</b>	<b>91</b>
5.5.1	Testes Unitários	91
5.5.2	Testes de controladores	92
5.5.3	Testes de Sistema	93
<b>5.6</b>	<b>Principais Problemas</b>	<b>94</b>
<b>6</b>	<b>CONCLUSÃO</b>	<b>96</b>
<b>6.1</b>	<b>Trabalhos futuros</b>	<b>96</b>
	<b>REFERÊNCIAS</b>	<b>97</b>
<b>A</b>	<b>HISTÓRIAS DE USUÁRIO COMPLEMENTARES</b>	<b>99</b>

<b>A.1</b>	<b>Área pública (Complementar)</b> . . . . .	<b>99</b>
<b>A.2</b>	<b>Área do usuário (Complementar)</b> . . . . .	<b>101</b>
<b>A.3</b>	<b>Área do administrador (Complementar)</b> . . . . .	<b>102</b>
<b>B</b>	<b>PROTOTIPAÇÕES ADICIONAIS DE TELAS</b> . . . . .	<b>103</b>
<b>B.1</b>	<b>Área pública</b> . . . . .	<b>103</b>
<b>B.1.1</b>	<b>Área do usuário</b> . . . . .	<b>108</b>

## 1 INTRODUÇÃO

Neste capítulo, será apresentada uma perspectiva geral do trabalho desenvolvido, demonstrando o contexto de aplicação e a relevância do desenvolvimento.

### 1.1 Considerações iniciais

Desde sua colonização, o Brasil sempre teve sua economia extremamente dependente do meio ambiente. Historicamente, a economia brasileira passou por diversos ciclos, por exemplo, o ciclo do Pau-brasil, o ciclo da cana-de-açúcar, o ciclo do ouro e o ciclo do café, todos estes sendo a base da economia e totalmente ligados à forma com que se maneja o meio ambiente. Atualmente, não muito diferente, o Brasil continua dependente do meio ambiente com a agropecuária sendo responsável pelo aumento de 2,9% do PIB brasileiro em 2023, um recorde de crescimento graças ao aumento na produção de milho e soja (GOV, 2024), demonstrando assim, a dificuldade em desvincular a economia de questões ambientais, pois sem condições ambientais propícias a agricultura não se desenvolve resultando na estagnação da economia.

O Brasil possui excelentes condições para atividades econômicas relacionadas ao ambiente, visto que possui um clima favorável, abundância de terra fértil e água potável (GOV, 2024). Entretanto, a falta de conhecimento sobre natureza pode ser um risco enorme para estas atividades, pois o mau uso dos recursos, como desperdícios ou depredação da natureza são os principais fatores para quedas de produtividade. Faz-se necessário, então, uma maior atenção para capacitação das pessoas envolvidas no trabalho com recursos naturais, como, por exemplo, o acesso a cursos na área (MARTINEZ, 2022).

Considerando este contexto, tem-se a Maxiambiental Treinamentos, uma empresa que iniciou seus trabalhos em 2009, ofertando e gerenciando cursos de curta duração na modalidade presencial voltados para a área ambiental. Em mais de 10 anos de atuação, a empresa capacitou aproximadamente 8.000 (oito mil) profissionais, enfatizando a relevância da oferta de cursos de capacitação relacionados ao meio ambiente (MAXIAMBIENTAL, 2019).

Com o passar do tempo, as tecnologias utilizadas no sistema web da empresa ficaram defasadas. Assim, o sistema da Maxiambiental precisa de uma nova versão, com atualizações que vão desde sua UI/UX até sua estrutura de códigos interna, para continuar sendo uma ferramenta robusta para captação de novos clientes que necessitam de capacitação na área de meio ambiente.

Logo, este trabalho propõe o redesenho da interface do usuário, bem como a atualização das tecnologias usadas no desenvolvimento do sistema, e com isso, espera-se aumentar o engajamento do sistema e assim atrair mais usuários que desejam se capacitar e trabalhar da melhor forma possível com o meio ambiente.

## 1.2 Objetivos

Nesta seção, serão apresentados os objetivos do trabalho de conclusão de curso.

### 1.2.1 Objetivo geral

Desenvolvimento de uma nova versão do sistema de gestão de cursos de curta duração da empresa Maxiambiental Treinamentos.

### 1.2.2 Objetivos específicos

- Analisar, projetar e desenvolver uma área administrativa para gerenciamento de cursos e inscrições;
- Analisar, projetar e desenvolver uma área para o usuário se inscrever em cursos, visualizar os cursos realizados e acessar seus certificados;
- Analisar, projetar e desenvolver a área pública do sistema de cursos de modo que seja possível visualizar os cursos com inscrições abertas e suas informações, assim como, outras informações relevantes da empresa.

## 1.3 Justificativa

Com o passar do tempo, o sistema de cursos da empresa Maxiambiental Treinamentos tornou-se desatualizada. Com a chegada da pandemia de COVID-19, os cursos presenciais foram paralisados e, desde então, não houve desenvolvimento de uma nova versão do sistema. Atualmente, identifica-se diversos problemas, como interface e tecnologias desatualizadas. Por exemplo, o uso de tecnologias desatualizadas pode resultar em falhas de segurança, pois problemas corrigidos nas versões mais recentes ainda persistem nas versões antigas. Além disso, versões desatualizadas de tecnologias não recebem suporte dos desenvolvedores, o que pode agravar os problemas e dificultar a resolução de falhas e vulnerabilidades.

Devido aos altos padrões de exigência dos usuários, são necessárias várias medidas e cuidados para oferecer experiências agradáveis. Uma interface consistente e intuitiva, um visual agradável e um bom desempenho são de extrema importância para agradar os usuários e incentivá-los a continuar utilizando o sistema (MEDIUM, 2023).

Assim, a retomada da oferta de cursos representa um momento oportuno para o desenvolvimento de uma nova versão do sistema. Pois quando comparada a sites como Udemy (UDEMY, 2024) e Coursera (COURSERA, 2024), que também oferecem cursos, o sistema precisa de uma interface mais responsiva e limpa, com foco nos cursos e nas conquistas da em-

presa, para atrair mais a atenção dos usuários. Portanto, o estudo e o desenvolvimento de uma versão atualizada do sistema que supra as necessidades dos usuários e dos proprietários justificam a concepção deste trabalho.

#### 1.4 Estrutura do trabalho

Dando seguimento a este trabalho, os próximos capítulos serão organizados da seguinte maneira: o Capítulo 2 será dedicado ao sistema de cursos Maxiambiental Treinamentos, explorando sua origem, histórico e características principais, incluindo as limitações tecnológicas da versão existente e as necessidades de atualização; o Capítulo 3 detalhará o processo de desenvolvimento, abrangendo a metodologia ágil BDD<sup>1</sup>, as ferramentas utilizadas (Figma<sup>2</sup>, Docker<sup>3</sup>, Ruby on Rails<sup>4</sup>) e os fluxos de trabalho implementados (GitLab Flow<sup>5</sup>, CI/CD<sup>6</sup>); o Capítulo 4 apresentará as histórias de usuário, protótipos de interface e a modelagem inicial do banco de dados que fundamentaram o escopo funcional do sistema; o Capítulo 5 descreverá a implementação técnica das funcionalidades, incluindo os sistemas de autenticação, gerenciamento de cursos e turmas, dashboard administrativo, além da configuração de ambientes e dos testes realizados; e finalmente, o Capítulo 6 resumirá os resultados alcançados, as limitações do projeto e as propostas para trabalhos futuros.

---

<sup>1</sup> Behavior-Driven Development (BDD) é uma abordagem de desenvolvimento ágil que promove a colaboração entre desenvolvedores, QA e stakeholders para definir o comportamento esperado do software por meio de exemplos concretos.

<sup>2</sup> Figma é uma ferramenta de design de interface e prototipagem colaborativa baseada na web.

<sup>3</sup> Docker é uma plataforma para desenvolvimento, envio e execução de aplicações em contêineres, que garantem ambientes padronizados e portáteis.

<sup>4</sup> Ruby on Rails é um framework web de código aberto escrito em Ruby, que segue o padrão MVC e facilita o desenvolvimento rápido de aplicações web.

<sup>5</sup> GitLab Flow é uma estratégia de controle de versão que integra práticas de integração contínua com um fluxo de trabalho baseado em branches.

<sup>6</sup> Continuous Integration/Continuous Deployment (CI/CD) refere-se a práticas de automação para integração e entrega contínua de código, visando melhorar a qualidade e a velocidade do desenvolvimento.

## 2 MAXIAMBIENTAL TREINAMENTOS

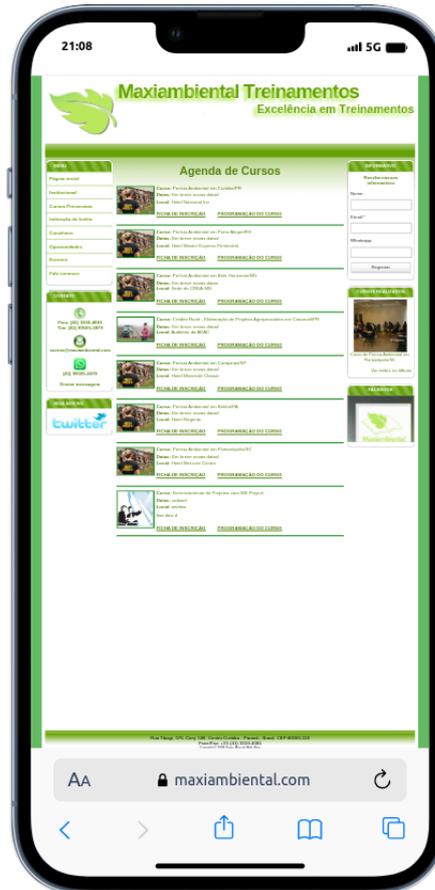
O sistema web de cursos da empresa Maxiambiental Treinamentos começou a ser desenvolvida em 2007 e a operar em 2009 com o objetivo de capacitar, por meio de treinamentos de curta duração, profissionais que atuam ou desejam atuar na área ambiental. Dentre estes profissionais estão agrônomos, engenheiros ambientais, biólogos, geólogos, além de economistas, administradores e muitos outros. Com professores altamente capacitados e qualificados, a Maxiambiental capacitou, nos 10 anos em que esteve ativa, mais de 8 mil profissionais em diversos Estados brasileiros se provando um sucesso (MAXIAMBIENTAL, 2019). Além disso, o sistema é um facilitador para quem gerencia os cursos, pois permite a gestão de inscrições, lista de espera, contatos e pagamentos.

Com mais de 10 anos de operação, o sistema está com uma interface não muito agradável, pois não aproveita os espaços nas laterais e ainda perde o foco dos cursos por ter muita informação nos menus laterais, como mostra a Figura 1. Além disso, o sistema não deixa claro como o usuário pode gerenciar seus cursos, pois não há uma opção de cadastro ou login, isso pode deixar novos usuários confusos.



**Figura 1 – Página inicial da Maxiambiental Treinamentos**

Atualmente, com os diversos tamanhos de tela é necessário que o site se adapte a cada uma delas, pois enquanto alguns usuários utilizam computadores, outros utilizam celulares o que torna indispensável uma boa responsividade. Por esta razão, o sistema da Maxiambiental precisa de uma melhoria na responsividade, visto que, ao mudar o tamanho da tela, o site permanece da mesma maneira e não se adapta, o que dificulta a navegação e a interação do usuário no sistema, como mostra a Figura 2.



**Figura 2 – Responsividade Maxiambiental Treinamentos**

Outras melhorias necessárias são nos formulários, pois quando ocorrem erros, estes são listados na parte superior da tela, como mostrado na Figura 3. Uma melhoria poderia ser apresentar os erros diretamente nos campos, isso daria mais contexto ao erro e facilitaria o entendimento por parte do usuário.

**MENU**

- Página inicial
- Institucional
- Cursos Presenciais
- Indicação de hotéis
- Convênios
- Oportunidades
- Eventos
- Fale conosco

**INFORMATIVO**

Receba nossos informativos

Nome:

Email:\*

Whatsapp:

Registrar

**CURSOS REALIZADOS**

Curso de Perícia Ambiental em Curitiba/PR

Ver todos os albums

**FACEBOOK**

Maxia...  
66 mil seguidores

**Ficha de inscrição para o curso Perícia Ambiental**

**Inscrição não pôde ser salvo: 13 erros.**

Por favor, cheque os seguintes campos:

- Email não é válido
- Número não pode ser vazio
- Forma de Pagamento não pode ser vazio
- Rua não pode ser vazio
- Telefone Celular não é válido
- Declaro que li e aceito as condições para realização do curso precisa ser aceita
- CEP não é válido
- Categoria não está incluso na lista
- Nome não pode ser vazio
- CPF não pode ser vazio
- Cidade não pode ser vazio
- Como Ficou Sabendo do Curso? não pode ser vazio
- Bairro não pode ser vazio

Nome:\*  CPF:\*

Rua:\*  Número:\*  Bairro:\*  Complemento:

CEP:\*  Estado:\*  Cidade:\*

Telefone Celular:\*  Telefone:  Telefone Comercial:

Email:\*  Email Secundário:

**CONTATO**

Fixo: (41) 3359-4081  
Tim: (41) 99505-3879

cursos@maxiambiental.com

(41) 99505-3879

Enviar mensagem

**SIGA-NOS NO**

MacBook Air

**Figura 3 – Formulário Maxiambiental Treinamentos**

Deve-se, entretanto, levar em consideração que o desenvolvimento do sistema iniciou-se em 2007, e que as tecnologias utilizadas na época ficaram defasadas. Ademais, o sistema conta com as seguintes tecnologias: Framework Ruby on Rails, para o *frontend* e *backend*, e PostgreSQL, para o gerenciamento dos dados. Todos, atualmente, em versões desatualizadas.

### 3 MATERIAIS E MÉTODOS

Para o desenvolvimento de uma plataforma que atenda às necessidades dos usuários, é fundamental a escolha de tecnologias e métodos adequados ao contexto da aplicação. Este capítulo apresenta os principais materiais e o método que irá nortear o desenvolvimento da nova versão da plataforma da Maxiambiental Treinamentos.

#### 3.1 Materiais

Para o desenvolvimento da plataforma, será utilizado uma combinação de ferramentas e tecnologias que abrangem desde o planejamento e design até a implementação e gerenciamento do código. A escolha destes materiais visa otimizar o processo de desenvolvimento e garantir a qualidade do produto final.

- **Pivotal Tracker:** Ferramenta ágil para gerenciar as histórias de usuário (descrições das funcionalidades da plataforma do ponto de vista do usuário), facilitando a organização e o planejamento do projeto (PIVOTAL, 2024). A ferramenta Pivotal Tracker será utilizada para organizar e gerenciar as histórias de usuário, promovendo a transparência no acompanhamento do desenvolvimento e facilitando a comunicação entre a equipe.
- **Figma:** Ferramenta para criar protótipos interativos das telas da interface, permitindo visualizar e testar o fluxo de navegação e a experiência do usuário antes do desenvolvimento (FIGMA, 2025).
- **Ruby on Rails:** Framework web completo e robusto que acelera o desenvolvimento de aplicações web, utilizando a linguagem Ruby. Facilita a criação de código estruturado, escalável e de fácil manutenção (RAILS, 2024).
- **PostgreSQL:** Sistema de gerenciamento de banco de dados relacional de código aberto, robusto e escalável. Oferece recursos avançados para garantir a integridade e a segurança dos dados da plataforma (POSTGRESQL, 2024).
- **Docker:** Plataforma para criar, implantar e executar aplicativos em contêineres. Garante a consistência do ambiente de desenvolvimento em diferentes máquinas e facilita a escalabilidade da aplicação (DOCKER, 2024).
- **Kamal** (anteriormente conhecido como MRSK): Ferramenta de orquestração de contêineres desenvolvida pela 37signals (criadores do Ruby on Rails e do Basecamp). Essa ferramenta simplifica o processo de implantação de aplicações web em servidores remotos, utilizando Docker e SSH (KAMAL, 2025).
- **RuboCop:** ferramenta de análise estática do código e formatador de código para Ruby, que aplica padrões de estilo e boas práticas de desenvolvimento (RUBOCOP, 2024).

- **Brakeman:** scanner de vulnerabilidades estática do código para aplicações Rails, que auxilia na identificação de potenciais problemas de segurança (BRAKEMAN, 2025).
- **GitHub:** Plataforma de hospedagem de código-fonte e colaboração para projetos de software. Permite o versionamento do código, facilitando o acompanhamento das alterações, a colaboração entre desenvolvedores e a resolução de conflitos (GITHUB, 2024).
- **Tailwind CSS:** Framework CSS utilitário que permite o desenvolvimento rápido de interfaces diretamente no HTML. Oferece um conjunto de classes pré-definidas que simplificam a criação de layouts responsivos, a estilização de componentes e a manutenção do código front-end, eliminando a necessidade de escrever CSS personalizado em muitos casos (TAILWIND, 2025)

## 3.2 Métodos

Para garantir um desenvolvimento organizado e focado na entrega de valor, será adotada uma abordagem estruturada e dividida em etapas. Este processo permitirá um melhor acompanhamento do progresso, a identificação e resolução de possíveis problemas e a entrega de um produto final que atenda às necessidades dos usuários e às expectativas dos *stakeholders*<sup>1</sup>.

### 3.2.1 Ciclo de vida ágil

Projetos de software falham por não fazerem aquilo que os clientes querem; ou por estarem atrasados; ou por serem difíceis de manter e evoluir; ou por todas as razões mencionadas (FOX; JOSEPH, 2017). O ciclo de vida ágil foi criado para remediar esses problemas presentes em muitos tipos de software. Neste projeto, é adotado o ciclo de vida ágil com Desenvolvimento Orientado a Comportamento (BDD). O BDD faz questionamentos sobre o comportamento de uma aplicação antes e durante o desenvolvimento. Dessa forma, as partes interessadas (*stakeholders*) estão menos propensas a falhas de comunicação. O BDD é uma metodologia ágil que foca no comportamento esperado do sistema por meio de uma linguagem clara para todos os participantes do projeto. O objetivo do BDD é garantir que o software satisfaça os requisitos dos *stakeholders* e que o código seja facilmente mantido e evoluído (FOX; JOSEPH, 2017). No desenvolvimento da nova plataforma da Maxiambiental Treinamentos, o BDD será aplicado da seguinte forma:

- **Definição de cenários:** Serão definidos cenários que descrevem o comportamento esperado do sistema em diferentes situações. Esses cenários serão elaborados em

---

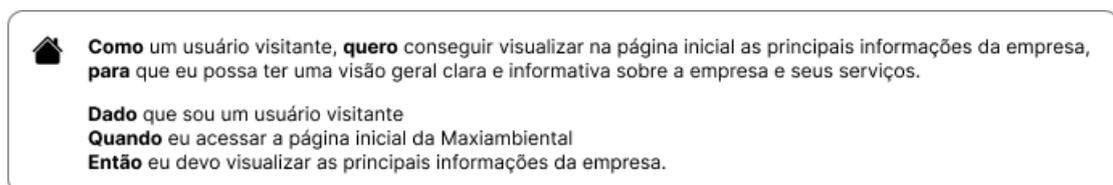
<sup>1</sup> Stakeholders são todas as partes interessadas em um projeto, incluindo clientes, usuários finais, patrocinadores, desenvolvedores e gestores, que têm interesse ou influência sobre os resultados.

linguagem natural, seguindo um formato estruturado que será detalhado ao longo deste capítulo;

- **Implementação dos testes:** Com cada cenário definido, a implementação de testes será facilitada, uma vez que os testes poderão ser diretamente derivados dos comportamentos descritos, permitindo a validação automatizada das funcionalidades do sistema;
- **Desenvolvimento do código:** Com os testes previamente definidos, o desenvolvimento do código será guiado pelos critérios de aceitação presentes nos cenários, assegurando que as funcionalidades implementadas atendam aos requisitos previamente estabelecidos.

### 3.2.2 Definição de histórias e cenários

A análise detalhada da plataforma atual é o ponto de partida para o desenvolvimento da nova versão. Ao investigar a estrutura do banco de dados, o fluxo de navegação, a interface e as tecnologias empregadas, é possível identificar os pontos fortes e fracos do sistema existente e também formas de otimizar o desenvolvimento. A partir da análise da plataforma atual serão levantados os requisitos para o desenvolvimento da nova versão. Esses requisitos serão elencados em forma de histórias de usuário <sup>2</sup>, com o intuito de capturar as necessidades e expectativas dos usuários e da Maxiambiental Treinamentos. A Figura 4 apresenta um modelo exemplo de história de usuário e um cenário. Os passos dos cenários usam a palavra-chave **Dado** para identificar o estado atual, **Quando** para identificar a ação e **Então** para identificar a consequência da ação. Os passos do cenário também usam a palavra-chave **E** e **Mas** para agir como uma conjunção para fazer mais descrições complexas de estado, ação e consequências.



**Figura 4 – História de usuário e cenários**

As histórias de usuário serão priorizadas considerando a urgência, a complexidade e o valor para o negócio da empresa. Para definir o escopo do projeto, será necessário priorizar as histórias de usuário. Para isso, cada membro da equipe atribui pontos para cada história, considerando o valor e o impacto que ela trará para a empresa. A sequência de Fibonacci (1, 1, 2, 3, 5, 8, 13...) será utilizada para essa pontuação, com valores menores para histórias mais simples e valores maiores para as mais complexas (FOX; JOSEPH, 2017). Essa priorização permitirá o

<sup>2</sup> O termo história de usuário é utilizado para se referir a uma única funcionalidade com um ou mais cenários que mostram diferentes maneiras de utilizá-las.

desenvolvimento de forma direcionada, focando nos itens de maior valor. Ao longo de aproximadamente três *Sprints*, é possível observar a velocidade do desenvolvedor, ou seja, a quantidade de histórias que consegue concluir em cada *Sprint*. Com essa informação, torna-se possível estimar quantas *Sprints* serão necessárias para concluir o projeto ou prever o progresso em um determinado período.

### 3.2.3 Prototipação para levantar os requisitos de interface do usuário

Serão criados protótipos interativos das novas telas da interface do usuário utilizando o Figma, incorporando os novos estilos e elementos visuais. Os protótipos serão revisados e validados pelos *stakeholders*, garantindo que o projeto atenda às expectativas e às necessidades dos usuários, visto que com o protótipo iterativo é possível visualizar e testar o fluxo de navegação e a experiência do usuário antes do desenvolvimento (FIGMA, 2025).

A prototipação interativa no Figma oferece diversas vantagens para o desenvolvimento da nova plataforma (AWARI, 2023):

- **Visualização antecipada:** Permite que os *stakeholders* visualizem a interface da plataforma antes de ser implementada, facilitando a comunicação e o alinhamento de expectativas;
- **Teste de usabilidade:** Possibilita simular a interação dos usuários com a plataforma, identificando problemas de usabilidade e validando as soluções propostas;
- **Iteração e aprimoramento:** Facilita a realização de ajustes e melhorias no design com base no *feedback* dos *stakeholders*, garantindo que a plataforma atenda às necessidades dos usuários;

### 3.2.4 Planejamento do projeto e monitoramento de progresso.

No desenvolvimento ágil de software, um *Backlog* é uma lista ordenada de tarefas ou funcionalidades a serem implementadas em um projeto. Neste projeto, o *Backlog* será composto pelas histórias de usuário. A priorização das histórias de usuário no *Backlog* será feita considerando a pontuação que cada história receberá, seguindo os pontos do Quadro 1:

Para facilitar a gestão do projeto e o acompanhamento do progresso do desenvolvimento, será utilizada a ferramenta Pivotal Tracker<sup>3</sup>. Essa ferramenta permite:

- Organizar as histórias de usuário, visualizando o fluxo de trabalho e o estado de cada tarefa;

<sup>3</sup> Disponível em: <https://www.pivotaltracker.com/>

**Quadro 1 – Pontuação das histórias de usuário**

Pontos	Dificuldades
1	Muito Fácil
2	Fácil
3	Médio
5	Difícil
8	Muito Difícil

**Fonte: Autoria própria (2025).**

- Estimar o esforço necessário para implementar cada história de usuário, facilitando o planejamento das *Sprints*;
- Atribuir tarefas aos membros da equipe, garantindo que todos saibam suas responsabilidades. Acompanhar o progresso do desenvolvimento em tempo real, permitindo identificar gargalos e ajustar o planejamento;
- Gerar relatórios e gráficos, facilitando a comunicação com os *stakeholders* e a tomada de decisões.

O uso do Pivotal Tracker contribuirá para um desenvolvimento mais eficiente e transparente, garantindo o foco nas tarefas mais importantes e que o projeto avance de forma organizada e previsível.

### 3.2.5 Organização do fluxo de desenvolvimento

No contexto de organização do fluxo de desenvolvimento utilizando o Git<sup>4</sup>, tem-se o GitLab Flow<sup>5</sup>, um modelo de fluxo de trabalho que define um conjunto de regras e procedimentos para organizar as ramificações (*branches*) do código e facilitar a colaboração entre desenvolvedores (ZUP, 2021). A organização do desenvolvimento será baseado no modelo GitLab Flow, com *branches* divididos em diferentes tipos:

- *Branch* principal:
  - **Main:** O *branch* principal do projeto, representa uma versão estável do código que está em produção.
  - **Staging:** *Branch* para testes finais antes da implantação em produção;

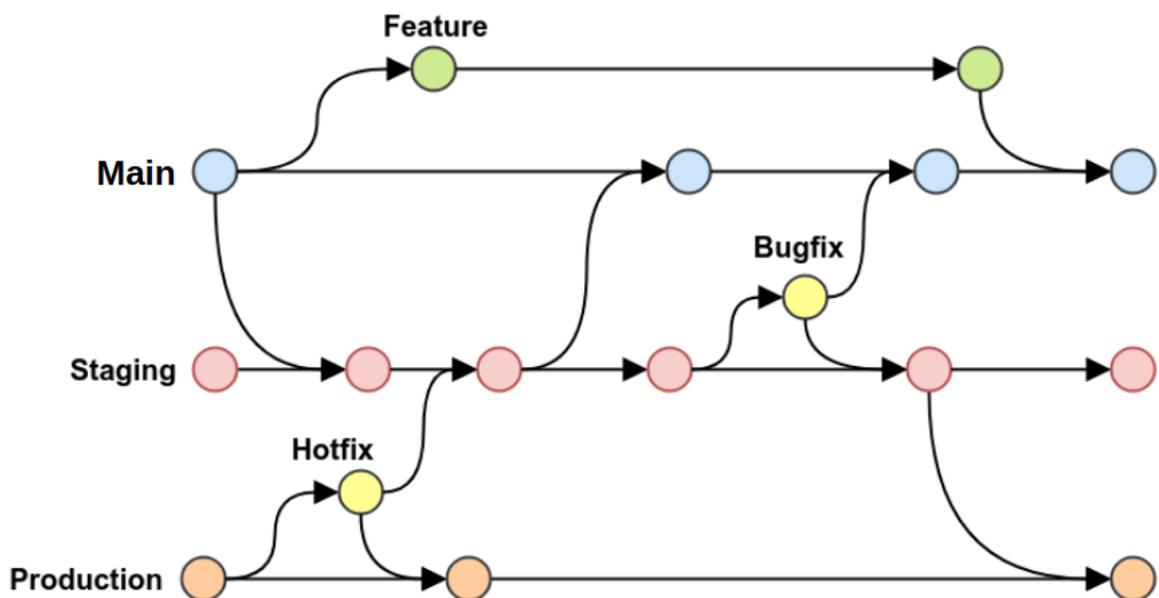
<sup>4</sup> Git é um sistema de controle de versão distribuído, amplamente utilizado no desenvolvimento de software para rastrear alterações no código-fonte e coordenar o trabalho entre múltiplos desenvolvedores.

<sup>5</sup> GitLab Flow é um modelo de fluxo de trabalho Git que combina o Git flow tradicional com práticas de integração contínua e entrega contínua, permitindo organizar melhor o processo de desenvolvimento e deploy.

- **Production:** *Branch* que espelha o ambiente de produção, usado apenas para implantação;
- *Branches* de suporte (Temporárias)
  - **Feature:** *Branches* temporários criados a partir de *Main* para desenvolver novas funcionalidades;
  - **Hotfix/BugFix:** *Branches* temporários criados a partir do *production* (*Hotfix*) ou do *staging* (*Bugfix*) para correção de *bugs*.

A Figura 5, mostra uma ilustração do fluxo de desenvolvimento do Gitlab Flow:

Figura 5 – GitLab Flow



Fonte: Extraída de ZUP (2021).

### 3.2.6 Verificação e validação de software

Os testes ajudam a verificar se o software atende à especificação e a validar se o design faz o que o cliente quer. O processo de verificação e validação pode ser dividido em (FOX; JOSEPH, 2017):

- **Teste de unidade:** assegura que um certo procedimento ou método faça o que dele é esperado.

- **Teste de módulo:** faz testes embarcando várias unidades de um mesmo módulo. Por exemplo, o teste de unidade trabalha com uma única classe enquanto o teste de módulo trabalha com relacionamentos entre as classes.
- **Teste de integração:** garante que as interfaces entre as unidades tenham pressupostos consistentes e se comuniquem corretamente. Esse nível não testa a funcionalidade das unidades.
- **Teste de sistema** ou **Teste de aceitação:** verificam se o software integrado atende às suas especificações.

Cada teste de nível superior delega os testes mais detalhados para os níveis inferiores.

Os detalhes de cada tipo de teste com aprofundamento em análises realizadas durante o desenvolvimento serão melhor explorados na escrita da monografia. Serão apresentadas as particularidades de cada nível de teste, desde a verificação de unidades individuais até a validação do sistema, demonstrando como as técnicas e ferramentas utilizadas contribuíram para uma melhor qualidade e confiabilidade do software.

Tal análise detalhada de cada etapa do processo de teste permitirá compreender a importância de uma abordagem abrangente e estruturada para a realização de testes em projetos de software.

### 3.2.7 Entrega e Implantação

Como no desenvolvimento serão utilizadas metodologias ágeis, que proporcionam ciclos de desenvolvimento mais curtos e *feedback* contínuo, será necessário utilizar práticas que deem mais velocidade<sup>6</sup> e otimização a esses ciclos. Logo, a abordagem escolhida para isso será o CI/CD (Integração Contínua e Entrega Contínua) pois, aliada à metodologia ágil, o CI/CD automatiza o processo de desenvolvimento, desde a integração do código-fonte em um repositório compartilhado até a entrega do software em produção. Isso permite o lançamento de novas funcionalidades e melhorias com mais frequência e segurança (JETBRAINS, 2023a).

---

<sup>6</sup> Velocidade é uma medida da taxa de funcionalidades adicionadas a uma aplicação.

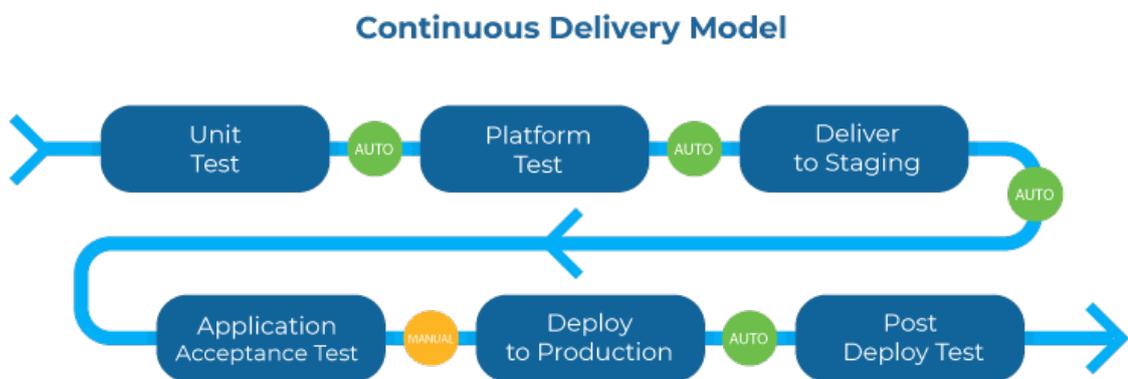
CI é a prática de automatizar a integração de alterações no código-fonte em um repositório centralizado, onde diversos testes automatizados são executados à medida que novas alterações são adicionadas (JETBRAINS, 2023b), como mostra a Figura 6.

### Continuous Integration:



**Figura 6 – Integração Contínua**  
**Fonte: Extraída de JetBrains (2023b).**

CD por sua vez, é a liberação automatizada do código já validado no CI para um ambiente de staging (teste) ou production (produção), sendo que o ambiente de staging é utilizado para testes finais antes da liberação para o usuário final, como mostra a Figura 7.



**Figura 7 – Entrega Contínua**  
**Fonte: Extraída de Silva (2023).**

## 4 ANÁLISE E PROJETO DO SISTEMA

Neste capítulo, serão apresentados os resultados parciais obtidos por meio da aplicação do Desenvolvimento Orientado a Comportamento (BDD), detalhando as histórias de usuário escritas e os protótipos de telas desenvolvidos, com o objetivo de demonstrar o progresso do projeto.

### 4.1 Histórias de usuário

Para o desenvolvimento da nova versão do sistema, as histórias de usuário coletadas e detalhadas a seguir representam os requisitos funcionais do sistema, que guiará o desenvolvimento e permitirá que o sistema atenda às necessidades dos usuários. Cada história de usuário segue o formato da Figura 4 mostrada na subseção 3.2.1.

Detalhes sobre funcionalidades além do escopo proposto estão nas histórias complementares, disponíveis em Apêndice A.

#### 4.1.1 Área pública

Área que deverá ser acessível e informativa para qualquer usuário visitante, esta seção do site terá como objetivo principal apresentar a empresa, seus valores e os serviços oferecidos, despertando o interesse de potenciais alunos e parceiros.

Feature: Informações da página inicial

**Como** um usuário visitante, **quero** conseguir visualizar na página inicial as principais informações da empresa, **para** que eu possa ter uma visão geral clara e informativa sobre a empresa e seus serviços. **Dado** que sou um usuário visitante **Quando** eu acessar a página inicial da Maxiambiental **Então** eu devo visualizar as principais informações da empresa.

#### Feature: Visualização dos Cursos Ofertados

**Como** um usuário visitante, **quero** visualizar os cursos ofertados pela empresa, **para** que eu possa saber quais cursos estão disponíveis para minha inscrição. **Dado** que sou um usuário visitante **Quando** eu estiver na página inicial ou na página de cursos **Então** eu devo visualizar a lista de cursos ofertados pela empresa, com informações básicas sobre cada curso.

#### Feature: Visualização de Detalhes do Curso

**Como** um usuário visitante, **quero** visualizar os detalhes de um curso oferecido pela empresa **para** que eu possa obter informações completas sobre o curso, incluindo conteúdo, duração, requisitos e certificações. **Dado** que sou um usuário visitante **Quando** eu clicar em um curso específico **Então** eu devo ser direcionado para a página de detalhes do curso, onde poderei ver informações completas sobre o curso, incluindo conteúdo, duração, requisitos e certificações.

### 4.1.2 Área do usuário

A área do usuário é o local onde será necessário credenciais para o acesso, será projetada para que alunos possam se inscrever em cursos e se conectarem a outros profissionais da área ambiental.

#### Feature: Cadastro de Usuário

**Como** um usuário não autenticado, **quero** poder me cadastrar no sistema, **para** que eu possa criar uma conta e acessar os recursos disponíveis. **Dado** que sou um usuário não autenticado **Quando** eu acessar a página de cadastro **Então** eu devo ver um formulário para preencher meus dados e criar uma conta.

## Feature: Login de Usuário

**Como** um usuário não autenticado, **quero** poder fazer login no sistema, **para** que eu possa acessar minha conta e utilizar as funcionalidades oferecidas. **Dado** que sou um usuário não autenticado **Quando** eu acessar a página de login **Então** eu devo ver um formulário para inserir meu email e senha e acessar minha conta.

## Feature: Redirecionamento para Login na Inscrição

**Como** um usuário não autenticado, **quero** ser redirecionado para a página de login ao tentar se inscrever em um curso, **para** que eu possa fazer login e completar a inscrição. **Dado** que sou um usuário não autenticado **Quando** eu tentar me inscrever em um curso **Então** eu devo ser redirecionado para a página de login para que eu possa fazer login e completar a inscrição.

## Feature: Completar Cadastro Antes da Inscrição

**Como** um usuário autenticado, **quero** completar meu cadastro, caso ainda não o tenha feito, **para** que eu possa prosseguir com a inscrição na Maxiambiental. **Dado** que sou um usuário autenticado, mas ainda não completei meu cadastro **Quando** eu clicar no botão de inscrição do curso **Então** eu devo ser direcionado para uma página de cadastro de usuário, onde posso preencher meus dados **E** após preencher meus dados corretamente, devo ser redirecionado para a página de inscrição do curso, com meus dados pré-preenchidos **E** devo receber uma confirmação de que meu cadastro foi realizado com sucesso.

## Feature: Confirmação de Inscrição em Curso

**Como** um usuário autenticado, **quero** acessar um formulário de confirmação de inscrição ao clicar em "Inscrever-se" em um curso, **para** que eu possa confirmar minha inscrição no curso desejado. **Dado** que sou um usuário autenticado **Quando** eu clicar em "Inscrever-se" em um curso **Então** eu devo ser direcionado para um formulário de confirmação de inscrição, onde poderei revisar os detalhes do curso e confirmar minha inscrição.

#### Feature: Pagamento da Inscrição

**Como** um usuário autenticado, **quero** ter a opção de realizar o pagamento após confirmar minha inscrição, **para** que eu possa completar o processo de inscrição no curso. **Dado** que sou um usuário autenticado e confirmei minha inscrição em um curso **Quando** eu acessar a página de pagamento **Então** eu devo ter a opção de realizar o pagamento da inscrição através dos métodos de pagamento disponíveis.

#### Feature: Confirmação de Pagamento e Inscrição

**Como** um usuário autenticado, **quero** receber uma confirmação após concluir o pagamento da inscrição, **para** que eu saiba que minha inscrição foi bem-sucedida. **Dado** que sou um usuário autenticado e realizei o pagamento da inscrição **Quando** o pagamento for processado com sucesso **Então** eu devo receber uma confirmação de que minha inscrição foi bem-sucedida e que meu acesso ao curso está liberado.

#### Feature: Visualização de Turmas Abertas

**Como** um usuário autenticado, **quero** visualizar as turmas abertas, **para** que eu possa ver quais cursos estão disponíveis e fazer minha inscrição. **Dado** que sou um usuário autenticado **Quando** eu acessar a página de cursos ou a área do aluno **Então** eu devo visualizar uma lista das turmas abertas, com informações sobre cada turma e a opção de me inscrever.

### 4.1.3 Área do administrador

A área do administrador será o centro de controle do sistema, onde administradores terão acesso a ferramentas e informações para gerenciar cursos, instrutores, alunos e todos os aspectos do sistema.

#### Feature: Gerenciamento de Cursos (Admin)

**Como** administrador, **quero** criar, editar e excluir cursos, **para** que eu possa gerenciar a oferta de cursos disponíveis. **Dado** que sou um administrador logado no sistema **Quando** eu acessar a página de gerenciamento de cursos **Então** eu devo ver uma lista de todos os cursos cadastrados, com opções para criar, editar e excluir cursos. **E** ao criar um novo curso, devo ser capaz de preencher os campos nome, ativo, informações, conteúdo programático, carga horária e imagem.

#### Feature: Gerenciamento de Turmas (Admin)

**Como** administrador, **quero** criar, editar e excluir turmas para cada curso, **para** que eu possa organizar as turmas e gerenciar suas inscrições. **Dado** que sou um administrador logado no sistema **Quando** eu acessar a página de gerenciamento de turmas **Então** eu devo ver uma lista de todas as turmas cadastradas, com opções para criar, editar e excluir turmas. **E** ao criar uma nova turma, devo ser capaz de associá-la a um curso, preencher os campos título, prazo para inscrições, ativo, status, instrutor, informações sobre o curso, formas de pagamento, regras para realização do curso, programação em PDF, folder do curso e imagem.

#### Feature: Gerenciamento de Instrutores (Admin)

**Como** administrador, **quero** adicionar, editar e remover instrutores, **para** que eu possa associar instrutores qualificados aos cursos e turmas. **Dado** que sou um administrador logado no sistema **Quando** eu acessar a página de gerenciamento de instrutores **Então** eu devo ver uma lista de todos os instrutores cadastrados, com opções para adicionar, editar e remover instrutores. **E** ao adicionar um novo instrutor, devo ser capaz de preencher os campos nome, email, email alternativo, telefone, WhatsApp, foto e currículo.

## 4.2 Prototipação de telas

A seguir, são apresentadas as interfaces de usuário resultantes do levantamento dos requisitos de interface do usuário.

### 4.2.1 Área pública

A área pública será a seção acessível a todos os visitantes, sem a necessidade de login ou cadastro. Apresentará informações básicas sobre a empresa com objetivo chamar a atenção do usuário visitante e despertar o interesse nos serviços oferecidos pela empresa.

A Figura 8 ilustra a página que será o primeiro contato de um novo usuário com o sistema, sendo um cartão de visitas, por essa razão foi escolhido chamar a atenção, destacando o nome da empresa com uma imagem ilustrativa de fundo que traz a ideia principal do sistema, o meio ambiente.

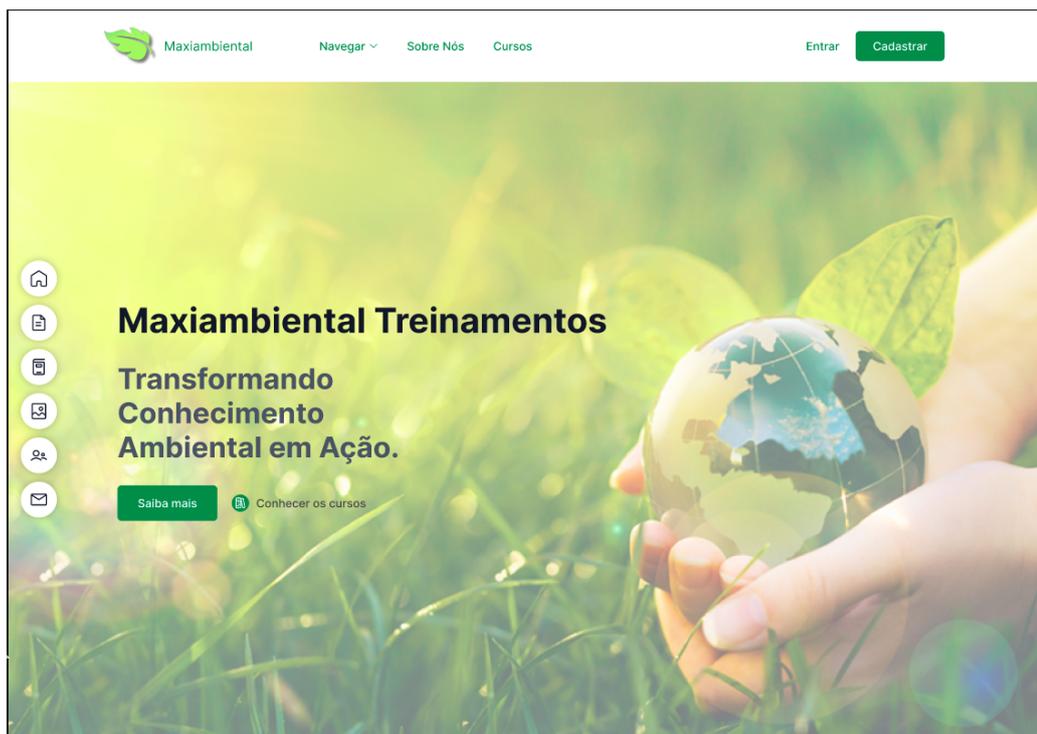
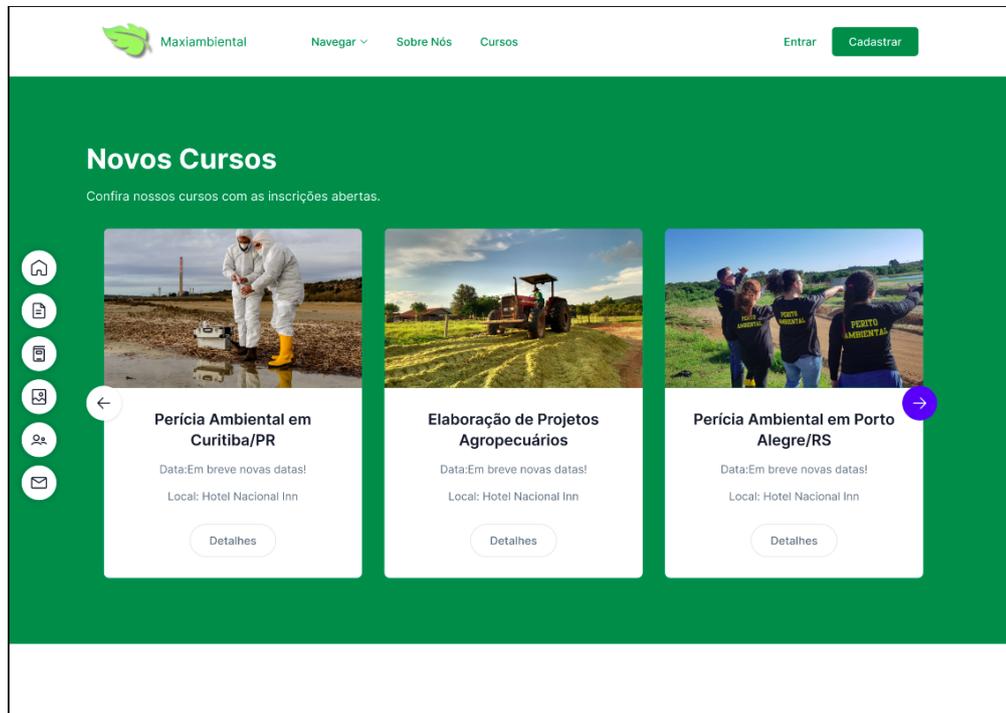


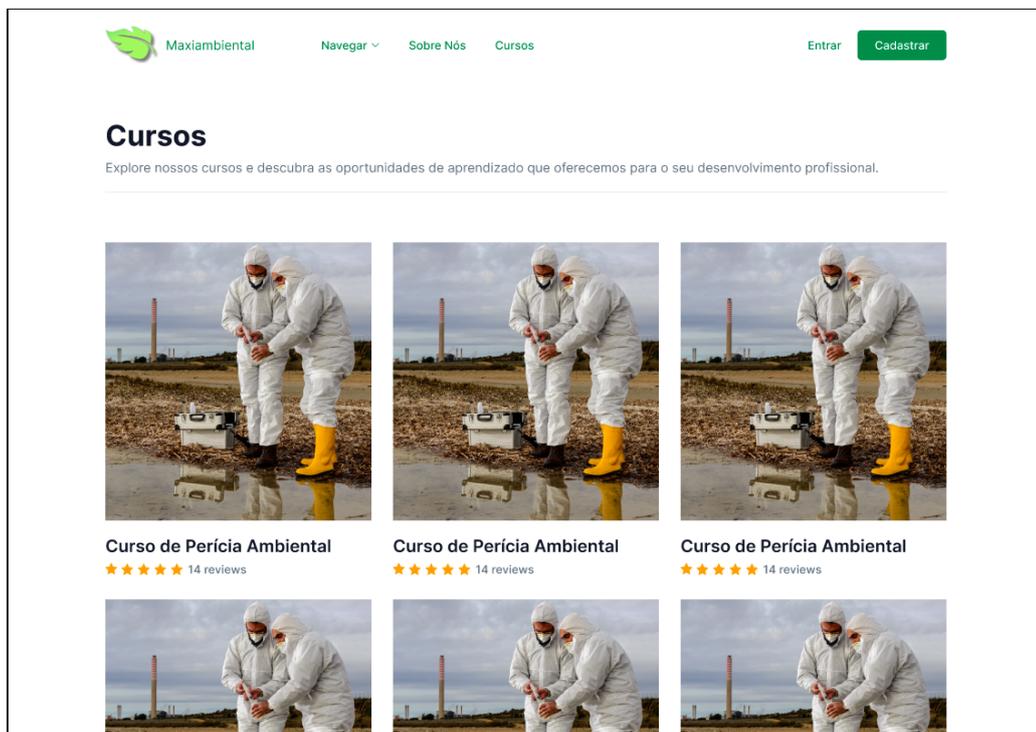
Figura 8 – Prototipação - Início

A Figura 9 ilustra a seção de cursos na página inicial, mostrando uma prévia dos cursos oferecidos pela Maxiambiental.



**Figura 9 – Cursos Maxiambiental - Início**

A página de cursos ilustrada na Figura 55 abaixo, irá mostrar todos os cursos ofertados pelo sistema, Feature: Visualização dos Cursos Ofertados.



**Figura 10 – Cursos Maxiambiental**

Na Figura 11 abaixo, detalha-se a página para visualização das informações de um curso com a possibilidade de inscrição, Feature: Visualização de Detalhes do Curso.

Maxiambiental Navegar Sobre Nós Cursos Entrar Cadastrar

**Curso de Perícia Ambiental**  
★★★★★ 5.00 Avaliação

A Perícia Judicial Ambiental é uma área que requer profissionais com formação na área ambiental e áreas afins (biologia, geologia, química, engenharias entre outras). A atuação do Perito Ambiental se dá de duas maneiras: ou como perito do Juiz ou como Assistente Técnico das Partes.

Ficha de inscrição

**Detalhes**

Datas: Em breve  
Horários do curso: 08:00 h às 12:00 h e 13:30 h às 18:00 h  
Cidade: Curitiba/PR  
Local: Hotel Nacional Inn - Rua Lourenço Pinto, 458, Centro

Sobre Programação Profissionais Pagamentos

**Objetivo**

Figura 11 – Curso Maxiambiental

A Figura 12 exibe a página de cadastro de usuário, onde os visitantes podem criar uma conta para acessar o sistema, Feature: Cadastro de Usuário.

Maxiambiental

**Cadastrar**

Nome

Email

Senha

Confirmar senha

Cadastrar

Continuar com o Google

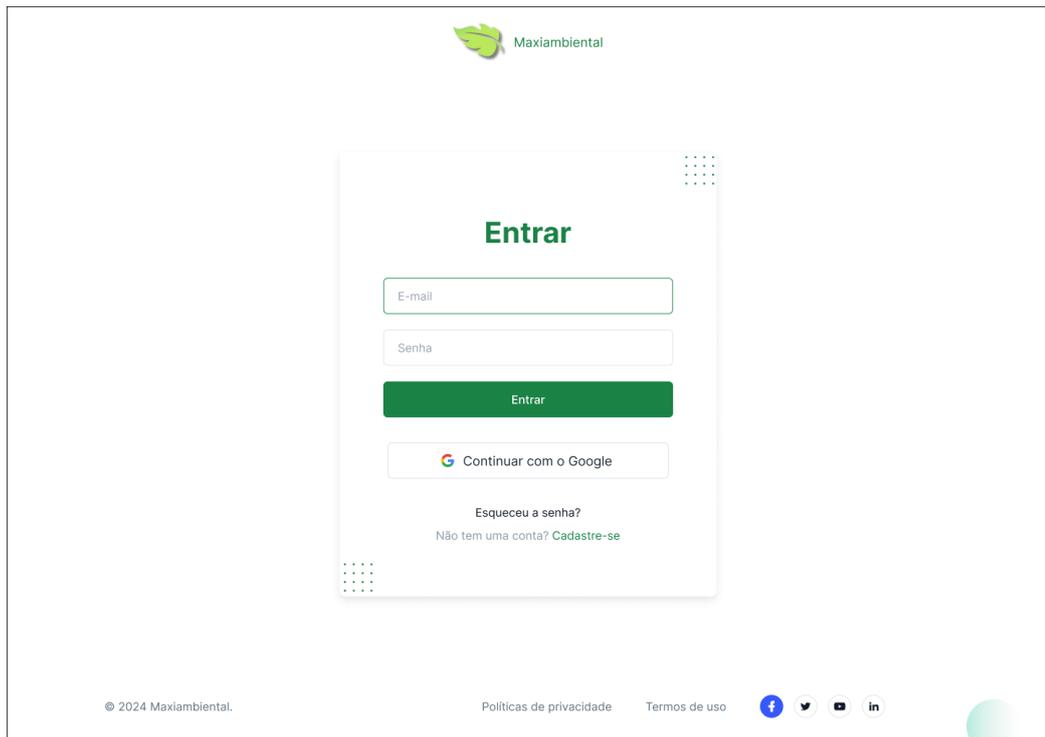
Já tem uma conta? [Entre](#)

Figura 12 – Cadastro de usuário

#### 4.2.2 Área do usuário

A área do usuário será uma seção exclusiva e restrita, acessível apenas após o usuário realizar login com suas credenciais (nome de usuário e senha). Essa área oferecerá funcionalidades e informações, que não estão disponíveis para visitantes não autenticados, tais como formulário de inscrição, acesso a certificados e perfis personalizados.

A Figura 13 exibe a página de login, onde os usuários podem inserir suas credenciais para acesso da conta, Feature: Login de Usuário.



**Figura 13 – Login**

A Figura 14 apresenta o formulário de inscrição, onde os usuários podem fornecer seus dados para se inscreverem em um curso, Feature: Completar Cadastro Antes da Inscrição.

The screenshot shows the 'Formulário de Inscrição' for 'Perícia Ambiental - Curitiba/PR'. The progress bar indicates the first step, 'Dados Pessoais', is active. The form fields include:

- Nome: Placeholder
- CPF: Placeholder
- Email: Placeholder
- Telefone: Placeholder
- CEP: Placeholder
- Estado: Dropdown menu (Estado)
- Cidade: Dropdown menu (Cidade)
- Endereço: Placeholder
- Formação: Dropdown menu (Estado)
- Empresa que trabalha: Placeholder

**Figura 14 – Formulário de inscrição**

A Figura 15 apresenta a etapa de confirmação dos dados do curso escolhido no formulário de inscrição, Feature: Confirmação de Inscrição em Curso.

The screenshot shows the 'Formulário de Inscrição' for 'Perícia Ambiental - Curitiba/PR'. The progress bar indicates the second step, 'Dados do Curso', is active. The form fields include:

- Como Ficou Sabendo do Curso?: Resposta (Placeholder)
- Você já participou de algum curso/palestra da Maxiambiental?:  Sim  Não
- Categoria: Dropdown menu (Categoria)
- Deseja fazer alguma observação?: Observação (Placeholder)
- Termos para realização do curso:
  - Investimento:
  - Profissionais: Taxa de matrícula R\$ 100,00 + R\$ 650,00
  - Estudantes: Taxa de matrícula R\$ 100,00 + R\$ 550,00

**Figura 15 – Formulário de inscrição - Dados do curso**

A Figura 16 ilustra a página de pagamento, onde o usuário pode escolher o método de pagamento para finalizar a inscrição, Feature: Pagamento da Inscrição.

Maxiambiental Navegar Sobre Nós Cursos Deivid Milinear

## Formulário de Inscrição

Perícia Ambiental - Curitiba/PR

1 2 3 4  
Dados Pessoais Dados do Curso Pagamento Confirmação

Formas de pagamento:

Cartão de crédito  Cartão de débito  Pix

VISA Mastercard Elo

Número do cartão

0000 0000 0000 0000

Nome do titular

Nome titular

Data de validade cvv

00/0000 000

**Figura 16 – Pagamento**

A Figura 17 exibe a página de confirmação de pagamento, onde o usuário recebe a confirmação de processamento do pagamento, Feature: Confirmação de Pagamento e Inscrição.

Maxiambiental Navegar Sobre Nós Cursos Deivid Milinear

## Formulário de Inscrição

Perícia Ambiental - Curitiba/PR

1 2 3 4  
Dados Pessoais Dados do Curso Pagamento Confirmação

Maxiambiental

**Seu pagamento está sendo processado!**

Agradecemos por escolher a Maxiambiental! Seu pagamento está sendo processado com segurança. Em breve, você receberá uma confirmação por e-mail com todos os detalhes da sua inscrição.

**Figura 17 – Confirmação de Pagamento**

### 4.2.3 Área do Administrador

A área de administração, mostrada na Figura 18, será uma seção restrita, acessível apenas a administradores do sistema. É onde todas as configurações, gerenciamento de conteúdo e funcionalidades são controlados.

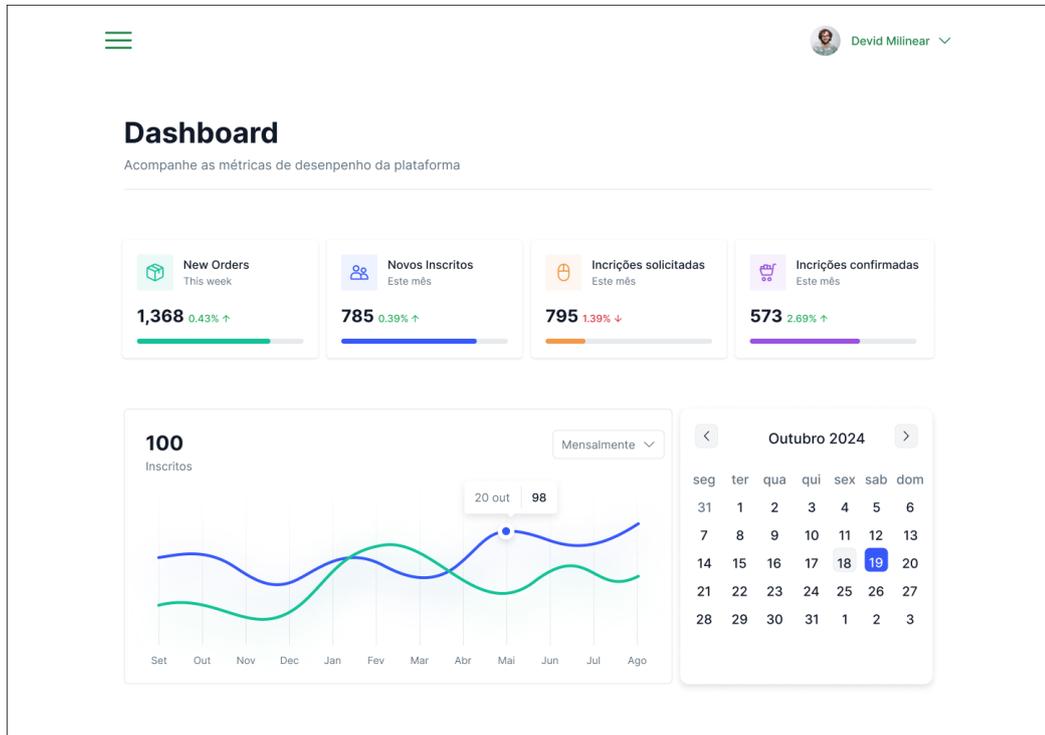
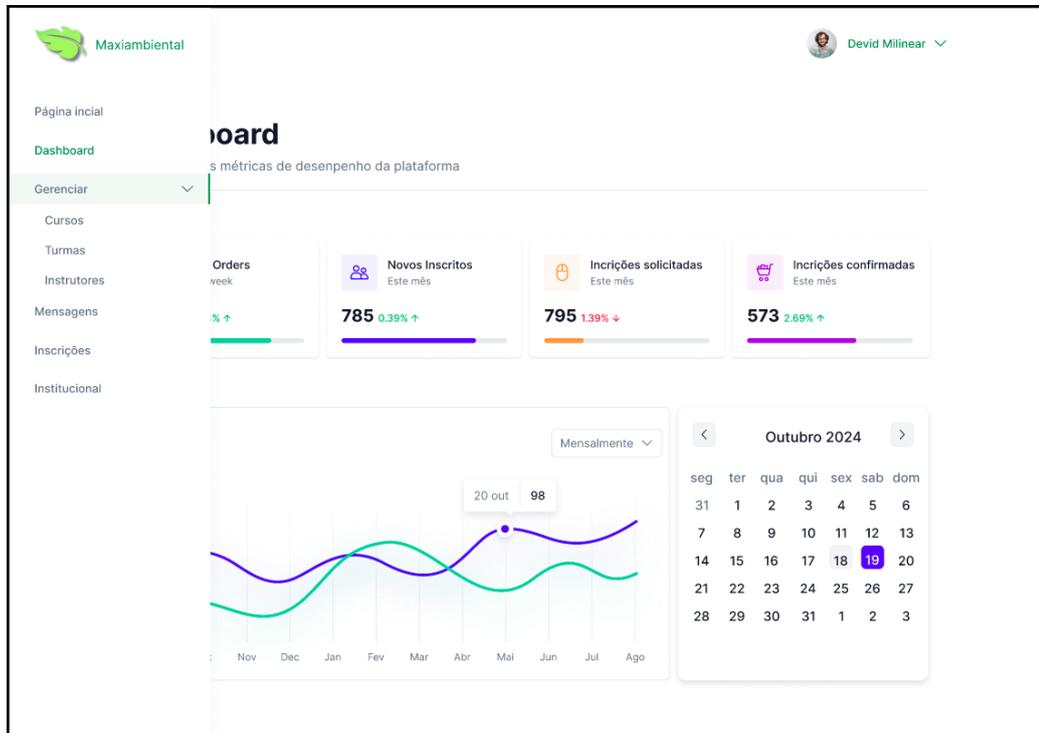


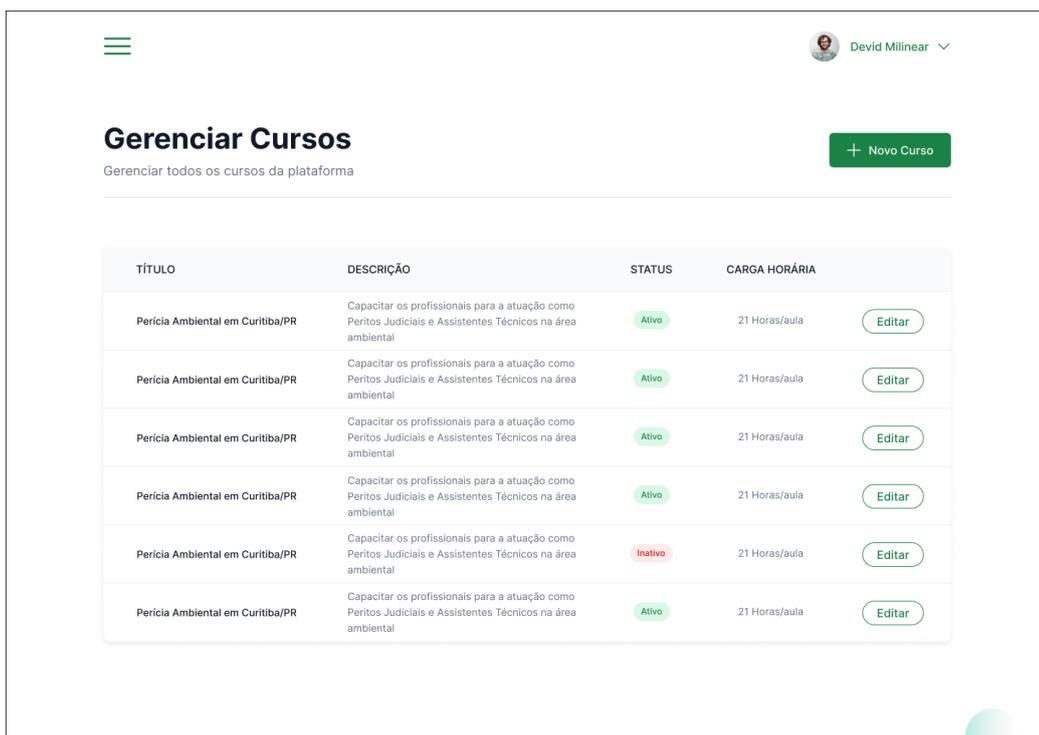
Figura 18 – Dashboard Administrador

A Figura 19 ilustra o menu com as opções de gerenciamento do administrador sendo elas gerenciar cursos, turmas, instrutores, informações da empresa e mensagens de usuários.



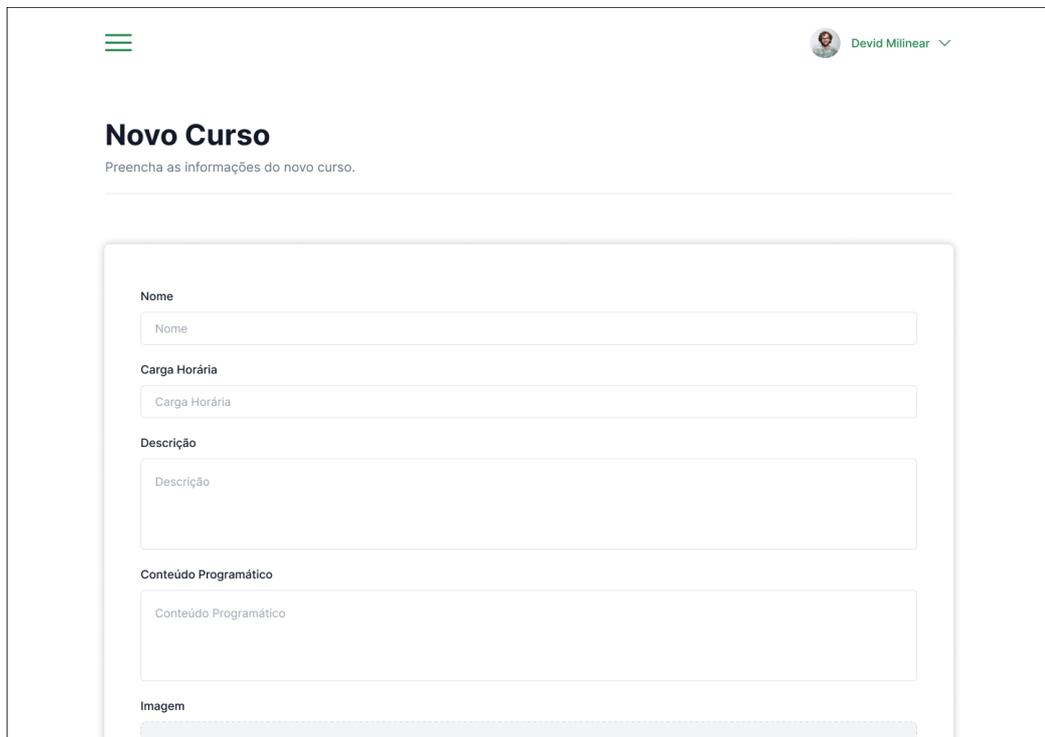
**Figura 19 – Menu Administrador**

A Figura 20 ilustra a interface de gerenciamento de cursos, permitindo que o administrador crie, edite e exclua cursos, Feature: Gerenciamento de Cursos (Admin).



**Figura 20 – Gerenciamento de Cursos**

A Figura 21 ilustra o formulário para criação de um novo curso, onde é necessário colocar todas as informações sobre o curso.



**Novo Curso**  
Preencha as informações do novo curso.

Nome  
Nome

Carga Horária  
Carga Horária

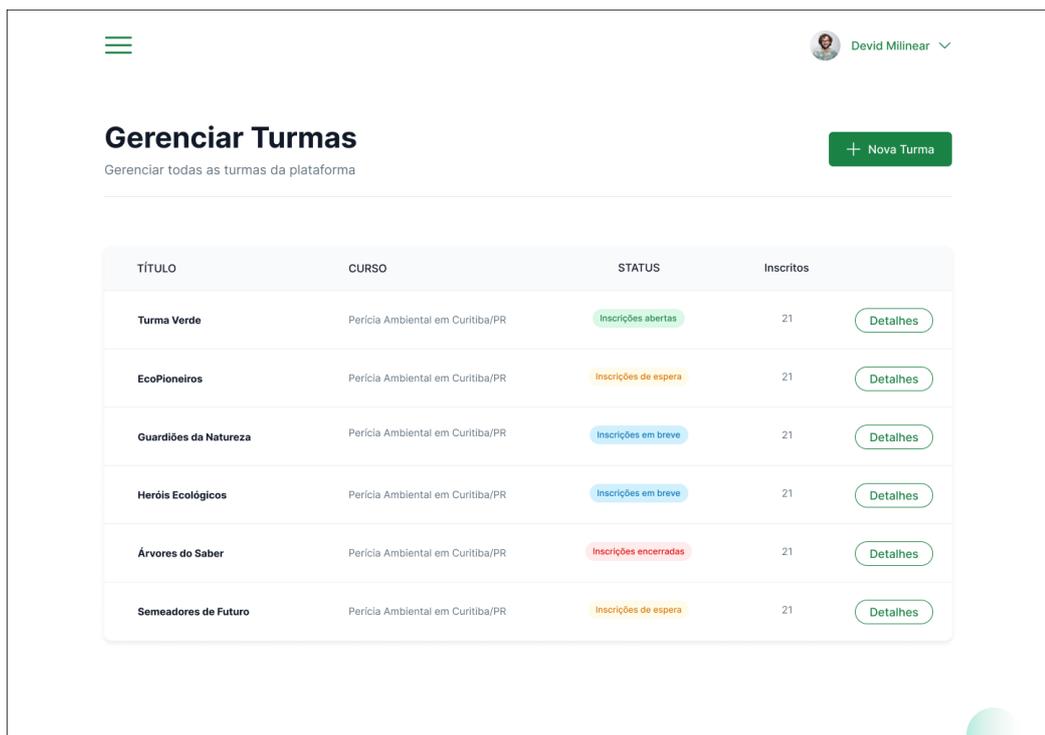
Descrição  
Descrição

Conteúdo Programático  
Conteúdo Programático

Imagem

**Figura 21 – Formulário Novo Cursos**

A Figura 22 exibe a interface de gerenciamento de turmas, permitindo que o administrador crie, edite e exclua turmas, Feature: Gerenciamento de Turmas (Admin).



**Gerenciar Turmas**  
Gerenciar todas as turmas da plataforma

+ Nova Turma

TÍTULO	CURSO	STATUS	Inscritos
Turma Verde	Perícia Ambiental em Curitiba/PR	Inscrições abertas	21
EcoPioneiros	Perícia Ambiental em Curitiba/PR	Inscrições de espera	21
Guardiões da Natureza	Perícia Ambiental em Curitiba/PR	Inscrições em breve	21
Heróis Ecológicos	Perícia Ambiental em Curitiba/PR	Inscrições em breve	21
Árvores do Saber	Perícia Ambiental em Curitiba/PR	Inscrições encerradas	21
Semeadores de Futuro	Perícia Ambiental em Curitiba/PR	Inscrições de espera	21

**Figura 22 – Gerenciamento de Turmas**

A Figura 23 ilustra o formulário para criação de uma nova turma, onde é necessário colocar todas as informações sobre a nova turma.

**Nova Turma**  
Preencha as informações da nova turma.

**Nome**  
Nome

**Curso**  
Associar Curso

**Instrutor**  
Associar Instrutor

**Início das Inscrições** **Fim das Inscrições**  
Data Data

**Formas de pagamento**  
Adicionar Formas de Pagamento

**Regras de Participação**  
Regras de participação

**Figura 23 – Formulário Nova Turma**

A Figura 24 apresenta a interface de gerenciamento de instrutores, permitindo que o administrador adicione, edite e remova instrutores, Feature: Gerenciamento de Instrutores.

**Gerenciar Instrutores**  
Gerenciar todos os intrutores

+ Novo Instrutor

Nome	Formação	STATUS	Telefone	
<b>Musharof Chowdhury</b> musharof@example.com	Mestre em Engenharia Ambiental	Ativo	(42) 984571236	Editar
<b>Nenifer Lofess</b> loffes.cooper@example.com	Doutor em Biologia com ênfase em Conservação	Ativo	(42) 984571236	Editar
<b>Jhon Smith</b> jhon.smith@example.com	Engenheiro Químico com foco em Gestão de Resíduos	Ativo	(42) 984571236	Editar
<b>Sulium Keilym</b> sulium.info@example.com	Advogado Ambiental	Ativo	(42) 984571236	Editar
<b>Alex Semuyel</b> alex.semuel@example.com	Geógrafo com foco em Geoprocessamento e Sensoriamento Remoto	Inativo	(42) 984571236	Editar
<b>Humil Limiton</b> limiton@example.com	Biólogo Marinho com experiência em Educação Ambiental	Ativo	(42) 984571236	Editar

**Figura 24 – Gerenciamento de Instrutores**

A Figura 25 ilustra o formulário para criação de um novo instrutor, onde é necessário colocar todas as informações sobre o novo instrutor.

**Figura 25 – Formulário Novo Instrutor**

### 4.3 Modelagem do banco de dados

A modelagem do banco de dados serve como um ponto de partida que será refinado ao longo do desenvolvimento, à medida que os requisitos do sistema evoluem e a compreensão dos dados se aprofunda. Assim, a modelagem foi construída levando em consideração três áreas funcionais do sistema, publica, cliente e administrativa. A estrutura organiza-se em quatro eixos principais:

- **Usuários e Acesso:** A tabela `users` atua como núcleo central, armazenando credenciais e perfis de todos os usuários. Especializações são feitas via `admins` (administradores com atributo `master` para superusuários) e `students` (alunos), mantendo integridade através de chaves estrangeiras, Figura 26.
- **Administração:** Entidades como `courses` (dados gerais dos cursos) e `classes` (turmas com prazos, `status` e regras) formam a estrutura principal da oferta de cursos. A relação `class_instructors` permite alocação de múltiplos instrutores por turma, enquanto `syllabus` e `workload` padronizam informações didáticas, Figura 27.
- **Operações:** O ciclo de vida do cliente é gerenciado por `enrollments`, que monitoram as inscrições, os pagamentos e a geração de certificados, Figura 28.

- **Institucional:** As tabelas `contact_messages` (solicitações de clientes), `settings` (conteúdo gerenciável), `hotels` e `partners` fornecem suporte operacional, Figura 29.

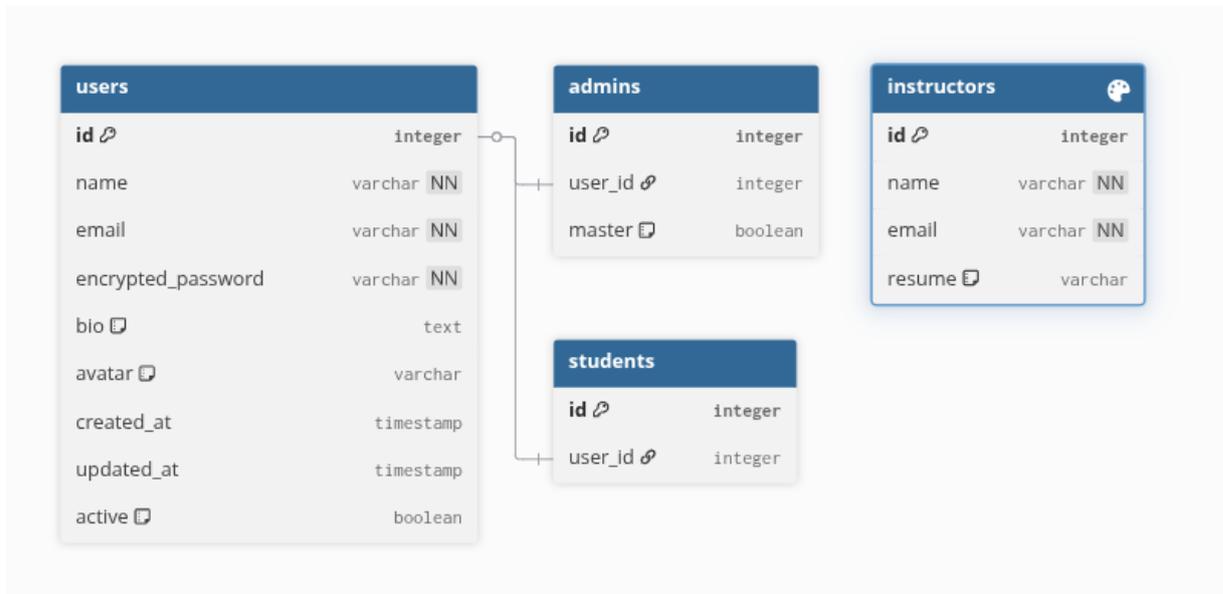


Figura 26 – Tabelas de usuários.

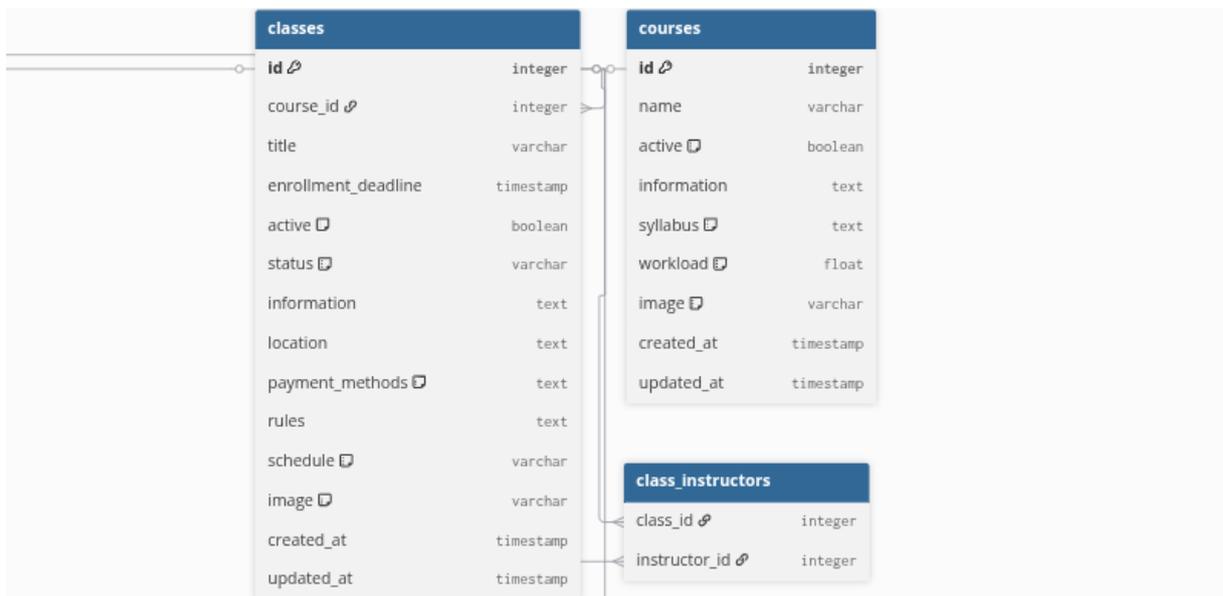


Figura 27 – Tabelas de administração.

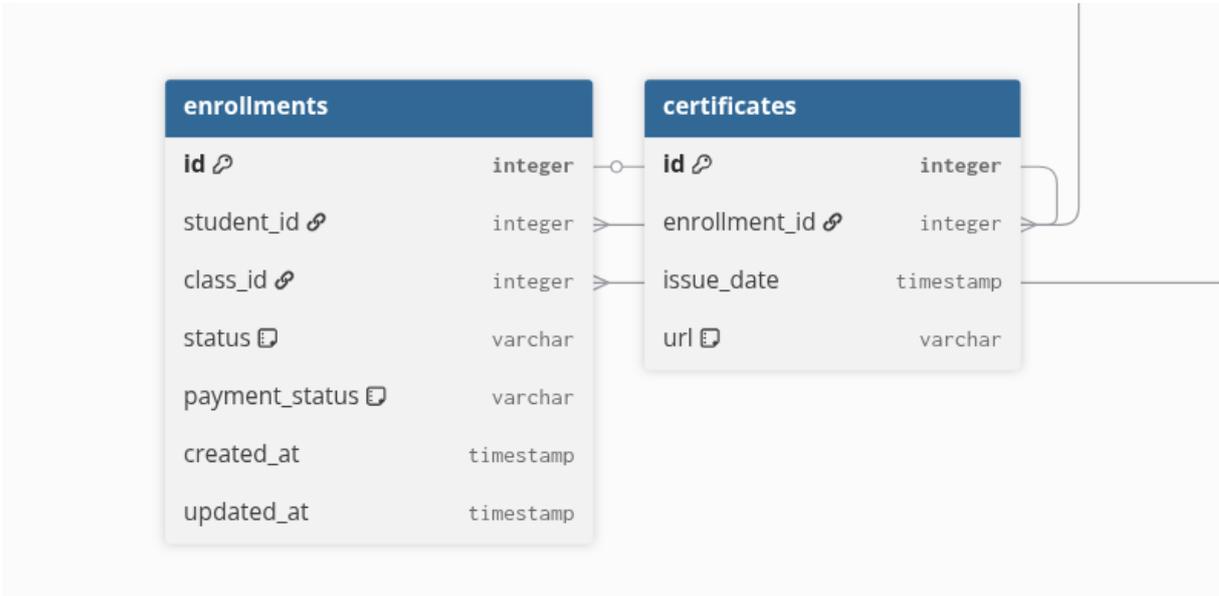


Figura 28 – Tabelas de operações.

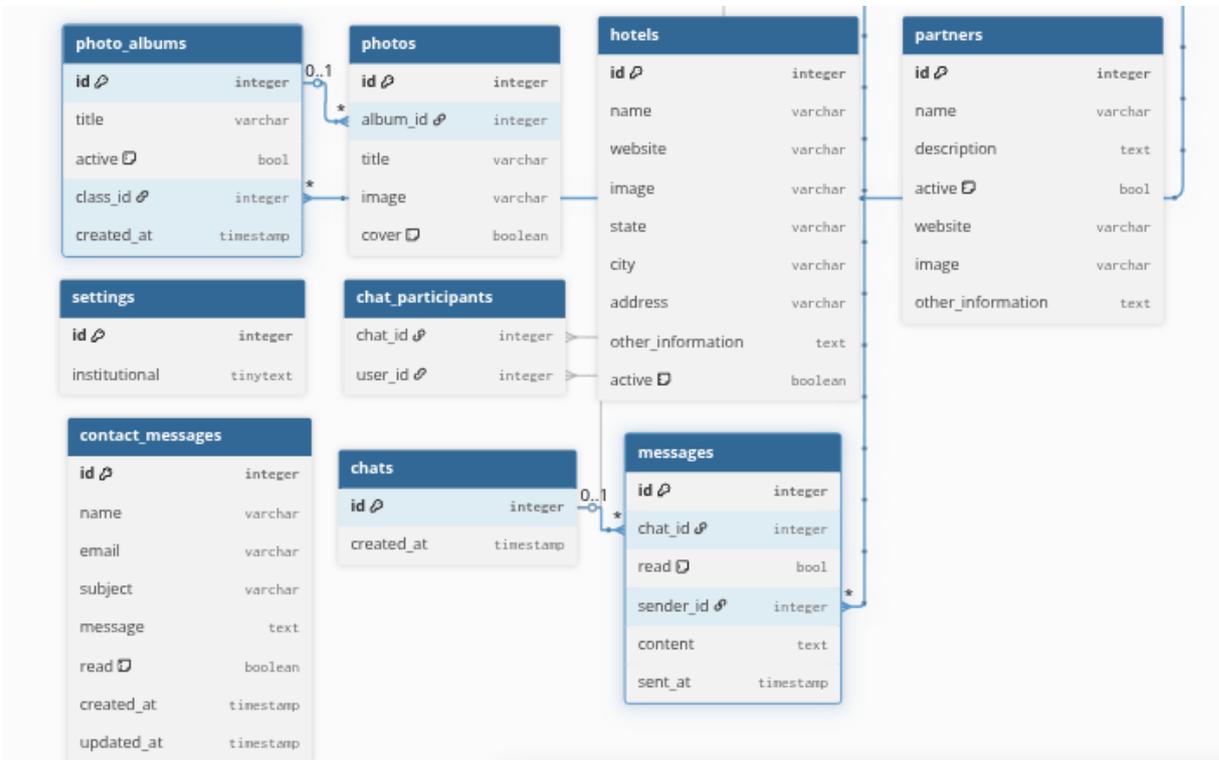


Figura 29 – Tabelas relacionadas ao institucional.

## 5 DESENVOLVIMENTO DO SISTEMA

Neste capítulo, serão apresentados os detalhes do processo de desenvolvimento, abrangendo as etapas de concepção, implementação e testes. Também serão discutidas as principais dificuldades encontradas e as alterações significativas que ocorreram no escopo ou nas funcionalidades anteriormente planejadas. Adicionalmente, serão detalhados os resultados alcançados ao fim de cada história desenvolvida.

### 5.1 Escopo de desenvolvimento

A partir da análise dos requisitos, histórias de usuário e prototipação de telas, definiu-se o escopo prioritário para a área administrativa do sistema. Realizou-se uma sessão colaborativa de pontuação com orientadores e orientando, onde cada funcionalidade foi avaliada considerando:

- Complexidade técnica (dependências, integrações)
- Impacto operacional (urgência para retomada das atividades)
- Esforço de implementação

Histórias com pontuação elevada (ex: Gerenciamento de Turmas e Gerenciamento de Fotos, ambas 8 pontos) foram decompostas em subtarefas para viabilizar *sprints* curtas e entregas incrementais.

Durante as discussões, estabeleceu-se que o valor das funcionalidades seria mensurado por:

- **Funcionalidades utilizáveis:** Partes operacionais imediatamente aproveitáveis pelo usuário final (ex: inscrição em turmas, emissão de certificados)
- **Melhorias incrementais:** Otimizações técnicas ou de experiência pós-implantação (ex: paginação de listas, validação de formulários)
- **Inovações estratégicas:** Recursos com potencial de ampliar o alcance do negócio (ex: integração com redes sociais na galeria)

As funcionalidades administrativas selecionadas para a primeira iteração, com respectivas pontuações consensuais:

**Tabela 1 – Priorização de funcionalidades para desenvolvimento inicial**

Funcionalidade	Pontuação
Cadastro de administradores	3
Gerenciamento de cursos	4
Gerenciamento de turmas	8
Controle de instrutores	3
Galeria de fotos	8
Mensagens institucionais	2
Edição de conteúdo institucional	1
Gerenciamento de certificados	5

**Nota:** Histórias com pontuação > 5 (Gerenciamento de Turmas e Galeria de Fotos) foram decompostas em subtarefas.

O processo de pontuação e priorização forneceu um roteiro inicial para o desenvolvimento, porém, a natureza iterativa do projeto permite ajustes dinâmicos ao longo da implementação. Funcionalidades inicialmente pontuadas podem ser repriorizadas ou não implementadas, assim como funcionalidades não pontuadas podem vir a serem implementadas.

## 5.2 Configuração do ambiente de desenvolvimento

Nesta seção, são descritas as etapas necessárias para a configuração inicial da aplicação, bem como a definição dos processos de automação que irão apoiar o ciclo de vida do desenvolvimento, desde a codificação até a entrega da aplicação em produção.

### 5.2.1 Criação e Configuração da Aplicação Web

Para estabelecer o ambiente de desenvolvimento da aplicação Rails, adotou-se uma abordagem containerizada utilizando Docker. Foi criada uma imagem personalizada, definida por meio de um `Dockerfile`<sup>1</sup>, que encapsula a versão específica do Ruby, as dependências essenciais e um usuário dedicado, a fim de evitar problemas de permissões.

A orquestração dos serviços foi realizada com o Docker Compose<sup>2</sup>, no qual foram definidos, no arquivo `docker-compose.yml`, os serviços necessários, como o servidor Rails, o banco de dados PostgreSQL e um servidor Chrome para execução de testes automatizados. Essa configuração proporciona um ambiente integrado e facilita a gestão isolada de cada componente.

Com o objetivo de garantir a segurança de informações sensíveis, como credenciais e cookies de sessão, foram utilizadas chaves secretas únicas para cada ambiente. Essas chaves

<sup>1</sup> `Dockerfile` é um script com instruções para construção de uma imagem Docker personalizada.

<sup>2</sup> Docker Compose é uma ferramenta para definir e gerenciar múltiplos contêineres Docker por meio de um único arquivo de configuração (`docker-compose.yml`).

são armazenadas de forma segura e referenciadas no código, protegendo o acesso a recursos críticos, como o banco de dados.

Para assegurar a qualidade do código, foram integradas ferramentas de análise estática. O RuboCop<sup>3</sup> garante a conformidade com as melhores práticas de estilo, enquanto a *gem* Bullet<sup>4</sup> auxilia na identificação de consultas ineficientes e oportunidades de otimização de desempenho. Além disso, o SimpleCov<sup>5</sup> é utilizado para gerar relatórios detalhados sobre a cobertura dos testes automatizados.

## 5.2.2 Configuração de Integração Contínua (CI)

Para garantir a consistência do código ao longo de todo o ciclo de desenvolvimento, foi configurada uma pipeline de Integração Contínua (CI) utilizando o `GitHub Actions`. A Integração Contínua é uma prática que consiste na integração frequente de alterações no código-fonte, permitindo a detecção precoce de erros e a validação automatizada de cada mudança realizada. Essa abordagem assegura que novas funcionalidades, ajustes ou correções não introduzam falhas no sistema, contribuindo para a estabilidade, qualidade e segurança contínuas do software durante o desenvolvimento.

O fluxo de CI do projeto é definido por meio de arquivos no formato `YAML (.yaml)`, localizados no diretório `.github/workflows`. Esses arquivos descrevem os processos automatizados que devem ser executados em eventos específicos, como `pull requests` ou `pushes` na `branch` principal.

Como pode ser observado na Figura 30, o arquivo `ci.yaml` atua como orquestrador principal, definindo os gatilhos para a execução da pipeline e as tarefas (*jobs*) que a compõem. O arquivo `ci.yaml` delega tarefas específicas para fluxos (*workflows*) separados, denominados `_tests.yaml` e `_linters.yaml`.

O fluxo `_tests.yaml` é responsável por executar todos os tipos de testes da aplicação (unitários, de componentes, de controladores e de sistema). A partir dele, um ambiente isolado é criado e a aplicação é simulada, permitindo a execução dos testes. Os *jobs* são executados em um contêiner Docker com a versão do Ruby utilizada no projeto.

O fluxo `_linters.yaml` é dedicado à análise estática de código, buscando por inconsistências de estilo, possíveis vulnerabilidades de segurança e outras violações às boas práticas de codificação. Os *jobs* também são executados em um contêiner Ruby isolado e incluem passos para o *checkout* do código e a instalação das *gems*.

Para garantir a qualidade do código e a segurança da aplicação, são utilizadas ferramentas de análise estática, que inspecionam o código-fonte sem executá-lo, buscando padrões,

<sup>3</sup> RuboCop é um analisador estático de código Ruby, voltado à verificação de estilo e boas práticas.

<sup>4</sup> Bullet é uma biblioteca Ruby que detecta consultas N+1 e outras ineficiências no acesso a banco de dados.

<sup>5</sup> SimpleCov é uma ferramenta de análise de cobertura de testes em Ruby, que gera relatórios sobre quais partes do código foram testadas.

```

1 name: Continuous Integration
2 on:
3   pull_request_target:
4     types: [opened, synchronize, reopened]
5   push:
6     branches:
7       - 'main'
8
9 jobs:
10  tests:
11    uses: ../github/workflows/_tests.yml
12    secrets:
13      RAILS_MASTER_KEY_TEST: ${ secrets.RAILS_MASTER_KEY_TEST }
14
15  code_quality:
16    uses: ../github/workflows/_linters.yml
17

```

Figura 30 – Arquivo de CI.

erros e vulnerabilidades. Foram utilizadas duas ferramentas estáticas para essa finalidade. A ferramenta RuboCop que realiza análise estática de código e funciona como formatador para Ruby, aplicando padrões de estilo e boas práticas de desenvolvimento. O Brakeman é um scanner de vulnerabilidades estáticas voltado para aplicações Rails, que auxilia na identificação de potenciais problemas de segurança.

### 5.2.3 Configuração de Entrega Contínua (CD)

Para automatizar o processo de implantação no ambiente *staging*, foi implementada uma pipeline de CD utilizando as ferramentas Kamal e GitHub Actions. A Entrega Contínua estende os processos de Integração Contínua, garantindo que, após a execução dos testes e das verificações de qualidade, a aplicação esteja sempre pronta para ser entregue.

A configuração do ambiente de *staging* é definida através de um arquivo YAML conforme o exemplo abaixo:

#### Listing 5.1 – Configuração do Kamal para Staging

```

1 # config/deploy.yml
2 service: my-app
3 image: user/my_app
4 servers:
5   web:
6     hosts:

```

```

7     - 192.168.0.100
8 registry:
9   username:
10    - KAMAL_REGISTRY_USERNAME
11  password:
12    - KAMAL_REGISTRY_PASSWORD
13 env:
14  ... # configuração do app

```

O arquivo `config/deploy.yml` é responsável por definir a configuração de implantação da aplicação utilizando a ferramenta Kamal. Nele, são especificados os parâmetros essenciais para a entrega no ambiente de produção. Ele especifica o nome do serviço (`service`), a imagem Docker utilizada (`image`), o servidor de destino (`servers`), as credenciais de acesso ao registro de imagens Docker (`registry`) e as variáveis de ambiente necessárias para a aplicação (`env`). Essa configuração permite que o Kamal execute o processo de deploy de forma automatizada no ambiente de staging.

A integração com GitHub Actions é realizada por meio de um *workflow* que é acionado automaticamente a cada atualização na branch staging:

#### Listing 5.2 – Pipeline simplificado de CD para Staging

```

1 jobs:
2   deploy:
3     runs-on: ubuntu-latest
4     steps: # exemplo com kamal apenas
5       - name: Install Kamal
6         run: gem install kamal
7       - name: Set up SSH key
8         env:
9           SSH_AUTH_SOCK: /tmp/ssh_agent.sock
10        run: # ... configurar o "agent"
11        run: | echo "... Credenciais ..." > .kamal/secrets
12       - name: Deploy
13         env: # env com credenciais da aplicação
14        run: |
15          git config --global --add safe.directory /__w/maxiambiental/maxiambiental
16          kamal deploy

```

Esse pipeline define um processo simplificado de entrega contínua (CD) para o ambiente de *staging*, utilizando o GitHub Actions. No *job deploy*, executado em uma máquina Ubuntu, são realizadas as etapas de instalação do Kamal, configuração das credenciais SSH e dos segredos necessários, além da execução do comando `kamal deploy`, que realiza a implantação da aplicação no servidor remoto. As credenciais e variáveis de ambiente são carregadas para garantir uma implantação segura e automatizada.

A facilidade de operação é um dos principais benefícios do Kamal, permitindo operações como:

#### Comandos úteis:

```
# Implantar versão específica
kamal deploy --version v1.2.3

# Rollback para versão anterior
kamal rollback v1.2.2

# Verificar status
kamal app status
```

### 5.3 Implementações das Funcionalidades

As seções a seguir apresentam a implementação das principais funcionalidades do sistema, organizadas em *features*, conforme a abordagem adotada no desenvolvimento com Ruby on Rails. Cada *feature* representa um conjunto coeso de requisitos relacionados a um fluxo específico de uso da aplicação. Inicialmente, são detalhadas as funcionalidades relacionadas à autenticação e gestão do administrador, abrangendo o login, a recuperação de senha e o gerenciamento do perfil, garantindo segurança e controle de acesso. Na sequência, são apresentadas as *features* de autenticação dos clientes e de gerenciamento de usuários, que incluem o cadastro e organização de clientes, instrutores, turmas e inscrições. Complementando o conjunto de requisitos funcionais, destaca-se o *dashboard* do administrador, que centraliza informações essenciais para uma melhor gestão do sistema.

#### 5.3.1 Feature: Login de administrador

Em aplicações web, garantir que apenas usuários autorizados acessem as áreas restritas é essencial. No projeto, isso significava criar um fluxo de autenticação capaz de lidar com administradores e clientes. Para implementar essa funcionalidade, adotou-se o **Devise**<sup>6</sup>, uma

<sup>6</sup> Devise. Total de Downloads: mais de 150 milhões; Estrelas no GitHub: aproximadamente 24 mil. Disponível em: <https://github.com/heartcombo/devise>. Acessado em: 29 de maio de 2025.

gem<sup>7</sup> consolidada em Ruby on Rails que abstrai boa parte da complexidade de criar autenticação do zero. Sua ampla base de usuários e manutenções frequentes garantiram confiabilidade e flexibilidade, possibilitando recursos nativos como recuperação de senha, bloqueio de conta e callbacks para auditoria.

Tendo adotado o Devise para a autenticação, era necessário organizar os diferentes perfis de usuário no banco de dados. A arquitetura inicial previa uma única tabela ‘users’, com um campo ‘role’ para distinguir administradores de clientes, conforme ilustrado na Figura 26. Entretanto, ao avançar para a fase de implementação, identificou-se três principais vantagens em separar as tabelas:

- **Segurança estrutural:** Dados críticos de administradores, como níveis de acesso, ficam fisicamente isolados, minimizando riscos de vazamentos acidentais.
- **Eficiência em escala:** Consultas massivas sobre clientes (por exemplo, análises de comportamento de compra) operam mais rapidamente em uma tabela dedicada, sem sofrer impacto do volume de registros administrativos.
- **Modelagem de domínio precisa:** Cada modelo pode ter comportamentos e validações próprias—por exemplo, políticas de senha ou regras de negócio específicas—sem exigir condicionais baseadas em ‘role’.

Dessa forma, adicionou-se duas tabelas distintas: `admins` e `clients`. Assim, o desenvolvimento da autenticação concentrou-se inicialmente no login do administrador.

Para este perfil, foi definido três atributos essenciais: `email`, `name` e `password`. Como só haverá poucos administradores, optou-se por não fornecer fluxo de cadastro via interface. Em vez disso, o primeiro usuário administrador é adicionado diretamente por um script de seed, garantindo controle centralizado. Exemplo simplificado do `db/seeds.rb`:

```
Admin.create_with(
  password: '123456',
  name: 'Maxiambiental ADM'
).find_or_create_by!(email: 'admin@maxiambiental.com')
```

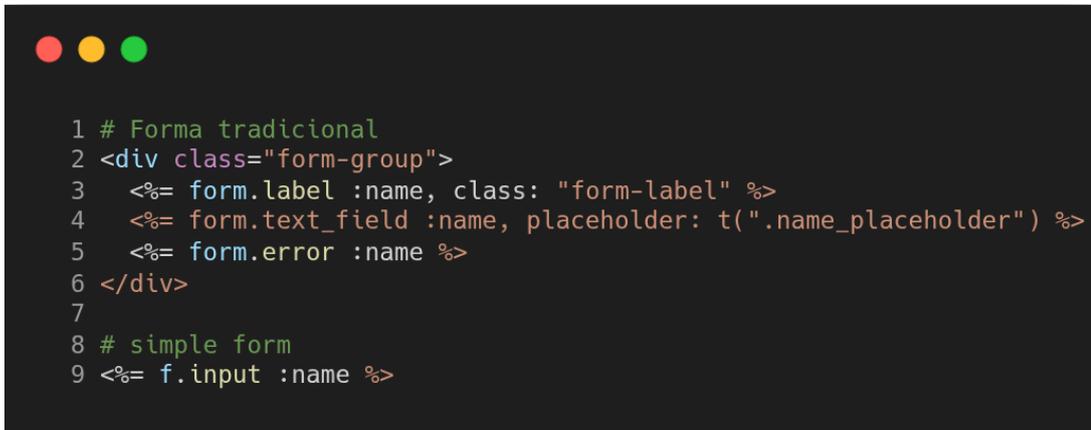
Para construir o formulário de login, foi utilizada a *gem simple form*<sup>8</sup>. Essa escolha foi motivada pela sua capacidade de gerar automaticamente campos HTML a partir das configurações do modelo, reduzindo significativamente o *boilerplate*<sup>9</sup> e melhorando a legibilidade das

<sup>7</sup> No contexto da linguagem Ruby, uma *gem* é um pacote reutilizável de código que pode ser facilmente instalado e integrado a uma aplicação. As gems são distribuídas via o sistema de gerenciamento de pacotes RubyGems.

<sup>8</sup> Simple Form é uma *gem* para Rails que gera formulários automaticamente. Documentação: [https://github.com/heartcombo/simple\\_form](https://github.com/heartcombo/simple_form) (acessado em 3 de junho de 2025).

<sup>9</sup> “Boilerplate” refere-se a trechos de código repetitivo e padronizado, que não trazem lógica de negócio específica e precisam ser escritos em várias partes da aplicação.

*views*<sup>10</sup>. Na Figura 31 é possível comparar um formulário criado com *helpers* nativos do Rails e outro montado com *simple form*. Nota-se que, com *simple form*, não é necessário declarar manualmente labels, divs ou classes CSS para cada campo: a *gem* infere o tipo de *input* com base na coluna do banco.



```

1 # Forma tradicional
2 <div class="form-group">
3   <%= form.label :name, class: "form-label" %>
4   <%= form.text_field :name, placeholder: t(".name_placeholder") %>
5   <%= form.error :name %>
6 </div>
7
8 # simple form
9 <%= f.input :name %>

```

Figura 31 – Comparação entre formulário nativo do Rails e *simple form*

A página de login do administrador, ilustrada na Figura 32, foi estilizada com *Tailwind CSS*<sup>11</sup>, garantindo consistência com o restante da aplicação e agilidade na definição de classes utilitárias. O layout exibe um título “Entrar”, seguido dos campos de e-mail e senha, cada um com *placeholder* que orienta o usuário (e.g., “Digite seu e-mail”). Ao submeter credenciais inválidas, o Devise dispara automaticamente uma mensagem de erro “E-mail ou senha inválidos.” exibida em forma de *flash message* abaixo do título, destacada em vermelho por meio das classes de *Tailwind*.

A seguir, as principais características dessa tela:

- **Estilização com *Tailwind*:** Uso de classes utilitárias para espaçamento, cores e responsividade, sem necessidade de CSS customizado.
- **Feedback de validação:** Mensagens de erro do Devise posicionadas abaixo de cada *input*, com destaque em tonalidade vermelha.
- **Componentização simples:** O *simple form* infere automaticamente o tipo de cada campo (‘email field’ e ‘password field’), reduzindo linhas de código na view.

Em suma, o *Devise* abstraiu a lógica de autenticação, sem a necessidade de criar as lógicas de controladores ou rotas, enquanto *Simple Form* e *Tailwind* reduziram em quase metade as linhas nas *views*, permitindo implementar o *login* em poucas linhas de ERB<sup>12</sup> e sem CSS adicional.

<sup>10</sup> No contexto de desenvolvimento web, *views* referem-se às interfaces gráficas que os usuários interagem, construídas com HTML, CSS e *frameworks* como *Tailwind CSS*.

<sup>11</sup> <https://tailwindcss.com>

<sup>12</sup> Sistema de templates Ruby que permite inserir código Ruby dentro de arquivos HTML.

**Figura 32 – Interface de login do administrador**

### 5.3.2 Feature: Recuperação de Senha do Administrador

Apesar de a autenticação ser fundamental para o controle de acesso ao sistema, é importante prever situações em que as credenciais possam ser esquecidas ou perdidas. Para garantir a continuidade do acesso e a segurança, foi implementada a funcionalidade de recuperação de senha. A *gem* Devise facilitou significativamente este processo, uma vez que oferece um fluxo pré-configurado para redefinição de senhas, abstraindo grande parte da complexidade.

A integração da funcionalidade de recuperação de senha foi simplificada com a adição da diretiva `:recoverable` no modelo `Admin`, conforme o trecho de código a seguir:

```
class Admin < ApplicationRecord
  devise :recoverable # Habilita tokens e fluxo seguro
end
```

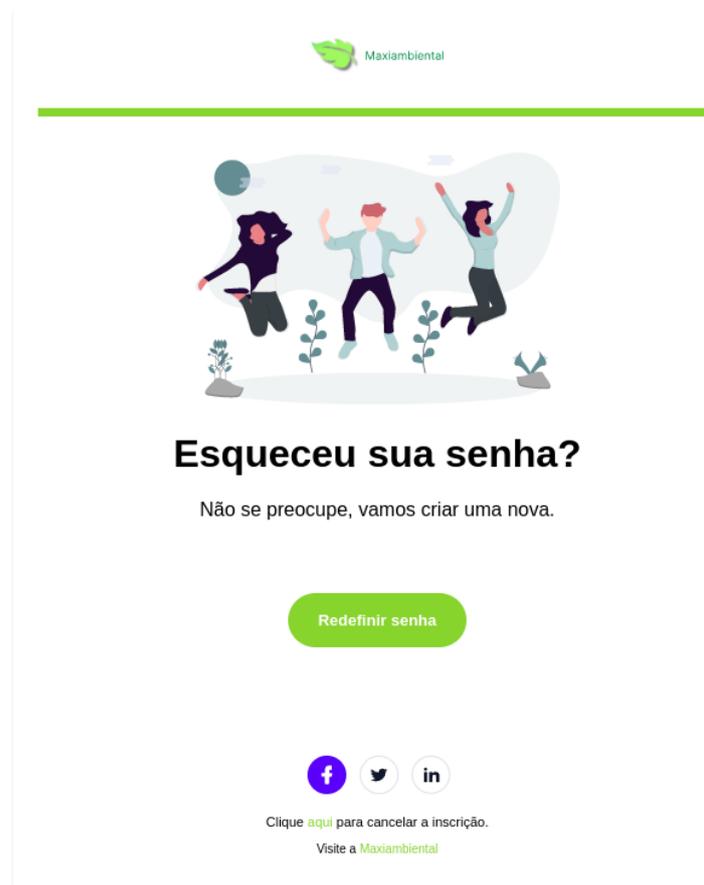
Essa configuração habilita automaticamente a geração de tokens criptografados de uso único, que são invalidados após a utilização, e registra todas as tentativas de redefinição para fins de auditoria.

Considerando que o fluxo de recuperação de senha depende do envio de e-mails, foi necessária uma plataforma para a gestão dessas comunicações. Para ambiente de desenvolvimento, adotou-se o *Mailtrap Email Sandbox*<sup>13</sup>, que intercepta as mensagens antes do envio

<sup>13</sup> Mailtrap. Plataforma de teste e envio de e-mails transacionais. Usada por mais de 800 mil desenvolvedores e 60 mil empresas. Disponível em: <https://mailtrap.io/>. Acessado em: 29 de maio de 2025.

real. Isso elimina o risco de disparos acidentais e permite uma validação segura do conteúdo do e-mail. Em produção, utilizou-se o *Mailtrap Email Sending* (API/SMTP), que oferece monitoramento detalhado de métricas de entrega, análise de spam score para otimização da taxa de inbox, e logs completos para rastreamento de transações e diagnóstico de falhas.

Com a lógica de recuperação de senha já incorporada pelo Devise, os esforços de implementação concentraram-se na personalização do template de e-mail e no desenvolvimento das *views* necessárias para o fluxo. Um template de e-mail customizado foi desenvolvido para a mensagem de recuperação de senha, sendo estilizado para garantir consistência com a identidade visual da aplicação. Priorizou-se a compatibilidade entre clientes de e-mail, como Gmail e Outlook, construindo o template com uma estrutura HTML simples e incorporando estilos diretamente na tag *style* para evitar inconsistências de renderização. A Figura 33 ilustra a interface visual do e-mail enviado ao administrador.



**Figura 33 – Template de e-mail para recuperação de senha**

Simultaneamente à criação do template, foram desenvolvidas as interfaces de usuário que compõem os dois passos principais do fluxo de recuperação. O processo é iniciado quando o administrador acessa uma view dedicada, onde informa o e-mail associado à sua conta. Esta tela inicial, apresentada na Figura 34, serve como ponto de entrada para a solicitação de redefinição. Após o envio, a solicitação é processada e, se o e-mail for válido, o link de recuperação é automaticamente enviado ao endereço fornecido.



**Esqueceu sua senha?**

**E-mail**

**Envie-me o link de redefinição**

Lembrou sua senha? [Entrar](#)

© 2024 Maxiam biental. [Política de Privacidade](#) [Termos de Uso](#) [f](#) [t](#) [i](#) [in](#)

**Figura 34 – Tela de solicitação de e-mail para recuperação de senha**

Ao receber o e-mail, conforme Figura 33, o administrador clica no link de recuperação, sendo redirecionado para uma nova view segura. Nesta tela, detalhada na Figura 35, é possível digitar a nova senha e sua confirmação. A correspondência entre os dois campos de senha é crucial para a precisão da redefinição e a segurança da nova credencial. Uma vez que a senha é redefinida com sucesso, o administrador é imediatamente capacitado a fazer login com suas novas credenciais.



**Digite sua nova senha**

**Senha**

**Confirmação da senha**

**Redefinir senha**

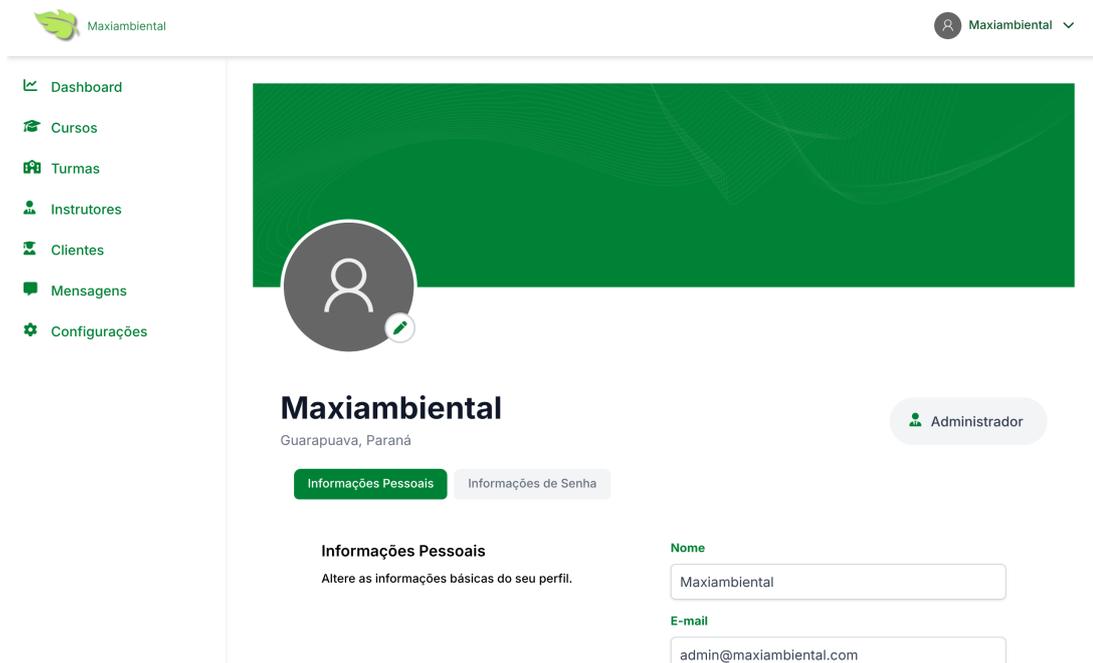
Lembrou sua senha? [Entrar](#)

© 2024 Maxiam biental. [Política de Privacidade](#) [Termos de Uso](#) [f](#) [t](#) [i](#) [in](#)

**Figura 35 – Tela de redefinição de senha, com campos para nova senha e confirmação**

### 5.3.3 Feature: Perfil do administrador

Com a garantia de acesso ao sistema pelo administrador, o desenvolvimento do sistema progrediu para oferecer a capacidade de gerenciamento das informações pessoais. Essa funcionalidade, importante para a manutenção de dados atualizados e para a autonomia do usuário, permite que o administrador visualize e edite seus dados, como nome, e-mail e senha. A Figura 36 ilustra a interface de gerenciamento do perfil, onde o administrador pode visualizar e modificar suas informações.



**Figura 36 – Tela de gerenciamento do perfil do administrador.**

Para manter a integridade e a segurança das alterações, qualquer atualização de e-mail ou senha exige a confirmação da senha atual. Essa medida impede modificações não autorizadas e reforça a segurança das credenciais, sendo gerenciada pelo Devise que verifica a senha atual antes de persistir as mudanças no banco de dados. O controle do processo de atualização é exemplificado no trecho do controlador, Figura 37, que utiliza o método `update_with_password` do Devise para lidar com a lógica de segurança e persistência dos dados.

```

1 def update
2   if @admin.update_with_password(admin_params)
3     bypass_sign_in(@admin)
4     redirect_to admin_edit_profile_path,
5               notice: t('flash_messages.profile_updated')
6   else
7     render :edit, status: :unprocessable_entity
8   end
9 end

```

Figura 37 – Código do editar perfil

Além das informações textuais, foi integrada ao perfil do administrador a possibilidade de adicionar e gerenciar uma foto de perfil. Essa funcionalidade personaliza a interface do administrador, e sua implementação foi realizada utilizando o Active Storage, a solução nativa do Ruby on Rails para upload de arquivos. A associação da foto ao modelo Admin é feita de forma simples, como ilustra o código abaixo:

```

class Admin < ApplicationRecord
  has_one_attached :avatar
end

```

Paralelamente ao desenvolvimento das funcionalidades de perfil, iniciou-se a integração de View Components no projeto, uma abordagem do Rails que promove a reutilização e organização do código de front-end. O primeiro componente implementado foi o de Flash Message, Figura 38, responsável por padronizar a exibição de notificações de sucesso, erro ou alerta em todo o sistema.

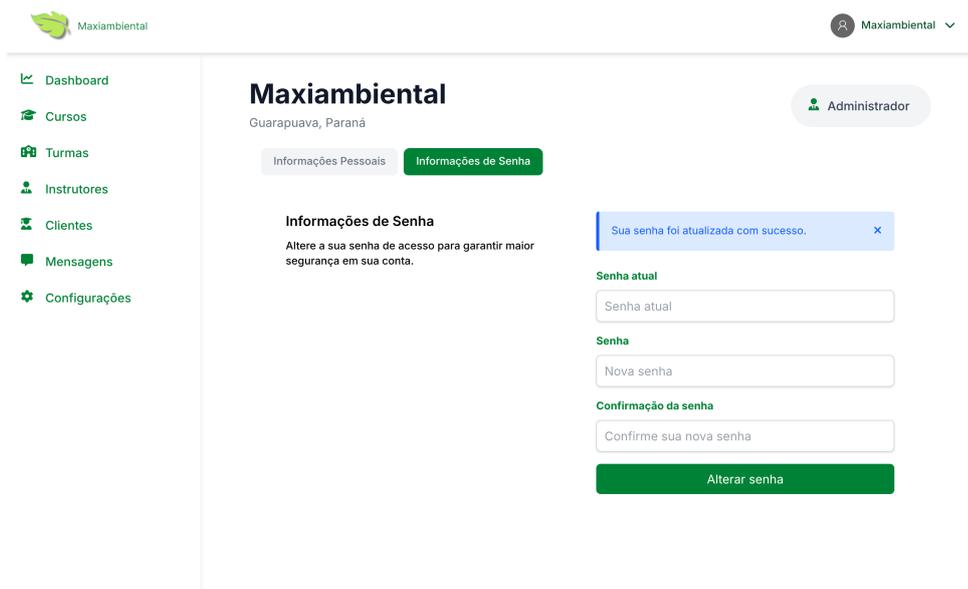


Figura 38 – Componente de *flash message*

Em vez de duplicar a lógica de renderização dessas mensagens em diversas *views*, o componente encapsula tanto a estrutura HTML quanto a lógica de apresentação. A forma como

o componente é chamado nas *views* é simplificada como mostra a Figura 39, melhorando a consistência da interface do usuário ao centralizar o controle das mensagens de feedback.

```

1 #component
2 <% messages.each do |type, message| %>
3   <div data-controller="alert">
4     <div data-alert-target="alert" role="alert" class="<%= alert_css_classes(type) %>">
5
6       <p class="mr-8"><%= message.html_safe %></p>
7
8       <button class="flex items-center justify-center absolute w-8 h-8 top-3 right-1.5"
9         type="button" data-action="click->alert#close">
10        <%= icon('fa-solid', 'xmark', class: "w-5 h-5") %>
11      </button>
12    </div>
13  </div>
14 <% end %>
15
16 # Utilização
17 <%= rc 'flash\message' %>

```

**Figura 39 – Código do componente *flash message***

#### 5.3.4 Feature: Autenticação dos clientes

Após estabelecer o acesso para o administrador, o próximo foco do desenvolvimento foi a implementação de um sistema de autenticação e gestão de contas para os clientes da Maxiambiental. Embora a base para o gerenciamento de sessões e credenciais assemelhe a mesma arquitetura utilizada para a autenticação do administrador, discutida na subseção 5.3.1, a funcionalidade para os clientes é mais abrangente e adaptada às suas necessidades.

A principal distinção está no ciclo de vida da conta do cliente, que inclui não apenas o login, mas também um processo de cadastro de usuário – uma etapa inexistente para o administrador. Além disso, para oferecer maior conveniência, foi incorporada a funcionalidade de autenticação via *Google*, permitindo que os clientes acessem o sistema utilizando suas credenciais de uma plataforma já conhecida e muito utilizada.

A implementação da autenticação tradicional (e-mail e senha) para o cliente segue os padrões do Devise, análoga à do administrador. O modelo Client foi configurado com as diretivas pertinentes, como mostra a Figura 40, refletindo as capacidades de registro e validação de dados. A Figura 41 apresenta a tela de cadastro de novos clientes, onde os usuários podem criar sua conta fornecendo as informações necessárias.

```

1 class Client < ApplicationRecord
2   devise :database_authenticatable, :registerable,
3         :recoverable, :rememberable, :validatable,
4         :confirmable, :lockable, :timeoutable, :trackable,
5         :omniauthable, omniauth_providers: [:google_oauth2]
6 end

```

Figura 40 – Modelo Client

Figura 41 – Tela de cadastro do cliente

A expansão para a autenticação social foi realizada por meio da gem `omniauth-googleoauth2`<sup>14</sup>, que facilitou a integração com o serviço de autenticação do Google. Essa funcionalidade exigiu configurações prévias no *Google Cloud Console*<sup>15</sup> para o registro da aplicação e a obtenção das credenciais de API (Client ID e Client Secret), essenciais para o *handshake* OAuth 2.0<sup>16</sup>. O arquivo de configuração (`config/initializers/devise.rb`) foi atualizado para incluir o provedor Google, como demonstrado no trecho de código a seguir, direcionando os parâmetros necessários para a conexão:

<sup>14</sup> `omniauth-googleoauth2` é uma gem Ruby que implementa o fluxo de autenticação via OAuth 2.0 para contas Google, facilitando a integração com o OmniAuth em aplicações Rails.

<sup>15</sup> Plataforma de gerenciamento de serviços e credenciais.

<sup>16</sup> OAuth 2.0 é um protocolo de autorização que permite a concessão de acesso limitado a recursos protegidos.

```

1 # config/initializers/devise.rb
2 Devise.setup do |config|
3
4 # ... outras configurações do Devise ...
5
6 google_oauth = Rails.application.credentials.google_oauth
7 config.omniauth :google_oauth2,
8                 google_oauth&.client_id,
9                 google_oauth&.client_secret
10 end

```

**Figura 42 – Credenciais Google**

A adição de *omniauthable* ao modelo *Client* é fundamental para ativar essa funcionalidade, permitindo que o Devise gerencie o fluxo de autenticação externo. Este trecho simples no modelo *Client* é o ponto de partida para a integração:

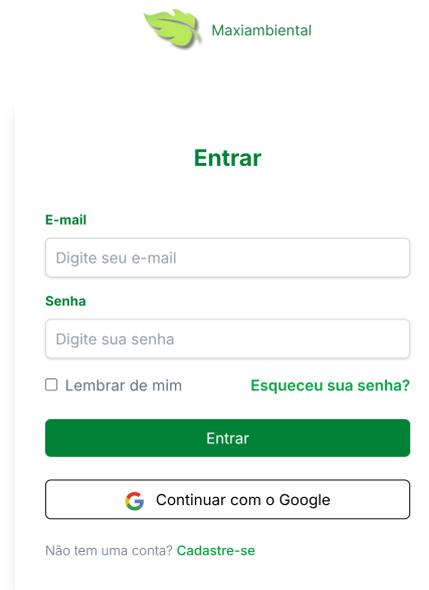
```

1 class Client < ApplicationRecord
2   include Searchable
3
4   devise :omniauthable, omniauth_providers: [:google_oauth2]
5
6   def self.from_google(user)
7     client = create_with(
8       uid: user[:uid],
9       provider: 'google',
10      password: Devise.friendly_token[0, 20],
11      name: user[:name]
12    ).find_or_create_by!(email: user[:email])
13
14    client.skip_confirmation! if client.respond_to?(:skip_confirmation!)
15    client
16  end
17 end

```

**Figura 43 – Implementação google no modelo Client**

Quando um cliente escolhe autenticar-se via Google, ele é redirecionado para a página de consentimento do Google, e após a autorização, a aplicação recebe os dados do usuário. O controlador *OmniauthCallbacksController* é o responsável por processar essa resposta, buscando um cliente existente com base no e-mail fornecido pelo Google ou criando um novo registro caso não seja encontrado. A Figura 44 ilustra a tela de login para clientes, destacando a opção de "Login com Google".



Maxiambiental

### Entrar

**E-mail**

**Senha**

Lembrar de mim [Esqueceu sua senha?](#)

**Entrar**

 Continuar com o Google

Não tem uma conta? [Cadastre-se](#)

**Figura 44 – Tela de login do cliente com opção de autenticação Google.**

### 5.3.5 Feature: Gerenciamento de instrutores

Com os sistemas de autenticação para administradores e clientes estabelecidos, o desenvolvimento avançou para o gerenciamento de instrutores. Esta área, acessível apenas por administradores, permite o acompanhamento completo desde o cadastro até a remoção de um profissional. É onde o administrador pode criar novos registros, atualizar dados existentes e até mesmo visualizar detalhadamente o perfil e o currículo de cada instrutor.

Para uma visão geral rápida, foi implementada uma tela índice que reúne todos os instrutores cadastrados. Essa interface, mostrada na Figura 45, oferece um panorama rápido com a foto do instrutor, nome, telefone e seu status no sistema (ativo ou inativo). Dali, o administrador tem acesso a links diretos para o currículo detalhado, a edição das informações ou a remoção do registro.

**Instrutores**  
Gerencie todos os instrutores cadastrados na Maxiambiental.

+ Novo Instrutor

Pesquisar...

Mostrando 1 - 10 de 20 instrutores

INSTRUTOR	CELULAR	STATUS	CURRÍCULO
<b>Carlos</b> victor.kub@keebler-bechtelar.test	(54) 92762-1457	Ativo	Visualizar Currículo
<b>Eduarda</b> margit_kessler@huels.test	(90) 94767-8250	Ativo	Visualizar Currículo
<b>Enzo Miguel</b> elwood.ernser@goyette.example	(31) 98817-7068	Ativo	Visualizar Currículo
<b>Henry</b> kurt.kuphal@ryan-mcglynn.example	(71) 96878-9959	Ativo	Visualizar Currículo
<b>Isabelly</b> matt@kuhic.test	(14) 98121-7655	Ativo	Visualizar Currículo
<b>João</b> octavio@schumm.example	(52) 95453-2981	Ativo	Visualizar Currículo

**Figura 45 – Tela de gerenciamento de instrutores.**

Quando um novo instrutor precisa ser adicionado, o administrador tem acesso a uma tela de cadastro específica. Os campos básicos como nome, e-mail e telefone são essenciais para identificação. No entanto, um diferencial dessa funcionalidade é a inclusão de um campo para o currículo do instrutor. Este não é um campo de texto comum; ele exige um editor de rich text para permitir formatação e a inserção de imagens. A Figura 46 mostra essa tela de cadastro em detalhes.

**Novo Instrutor**  
Preencha as informações do novo instrutor.

**Nome**  
Digite o nome do instrutor

**E-mail**  
Digite o e-mail do instrutor

**Celular**  
Digite o telefone do instrutor

**Currículo**

**B** **I** U Link **TT** **99** **<>** **≡** **≡** **≡** **≡** **≡** **≡** **↶** **↷**

**Figura 46 – Tela de cadastro de novos instrutores.**

Para a implementação do rich text, optou-se pela utilização do Action Text do Rails. Esta decisão foi tomada pelas seguintes facilidades:

- **Integração Nativa e Ecossistema Rails:** O Action Text<sup>17</sup> é uma funcionalidade built-in<sup>18</sup> do Rails (a partir da versão 6), o que garante uma integração com as demais ferramentas do framework, como Active Record<sup>19</sup>, Active Storage<sup>20</sup> e Asset Pipeline<sup>21</sup>.
- **Manutenção Simplificada:** Por ser desenvolvido e mantido pela equipe do Rails (Basecamp), o Action Text se beneficia das mesmas atualizações e padrões que o framework. Isso reduz a preocupação com incompatibilidades futuras entre versões do Rails e o editor, bem como a dependência de comunidades externas para suporte e resolução de problemas, o que é comum com gems de terceiros.
- **Segurança e Sanitização:** O Action Text já incorpora mecanismos robustos de sanitização de conteúdo HTML, prevenindo ataques XSS (Cross-Site Scripting) de forma nativa.
- **Simplicidade e Foco:** O Action Text, em conjunto com o Trix editor (que ele inclui), oferece um conjunto de funcionalidades de rich text essenciais e bem implementadas. Diferente de alguns editores de terceiros que podem ser excessivamente complexos e carregados de recursos não utilizados.
- **Gerenciamento de Anexos com Active Storage:** A integração nativa com o Active Storage é um ponto chave. O Action Text lida com o upload e o armazenamento de imagens e outros anexos diretamente no corpo do texto, utilizando a mesma infraestrutura de armazenamento já configurada na aplicação, como local ou Google Cloud Storage.

---

<sup>17</sup> Permite edição e exibição de conteúdo rico (rich text) usando o editor Trix, com suporte a formatação, links e anexos.

<sup>18</sup> Recurso nativo incorporado ao framework, sem necessidade de instalação externa.

<sup>19</sup> Camada de mapeamento objeto-relacional (ORM) do Rails, responsável pela comunicação com o banco de dados.

<sup>20</sup> Sistema do Rails para upload, armazenamento e associação de arquivos a modelos.

<sup>21</sup> Mecanismo de gerenciamento e empacotamento de arquivos estáticos (CSS, JS, imagens) no Rails.

Para associar o currículo ao modelo Instructor, basta adicionar uma diretiva simples no modelo:

```
#app/models/instructor.rb
class Instructor < ApplicationRecord
  has_rich_text :resume

  ... outras associações e validações ...
end
```

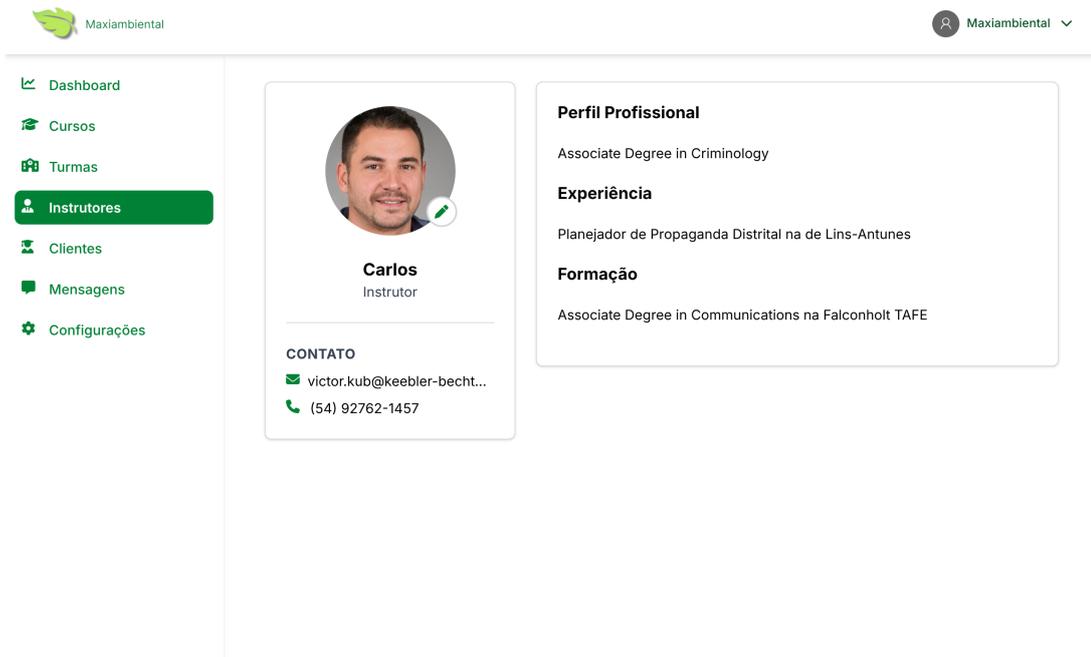
A atualização das informações de um instrutor é feita em uma tela, que espelha a tela de cadastro, mas com os dados pré-preenchidos. A Figura 47 mostra essa interface, onde o administrador pode ajustar nome, e-mail, telefone e o currículo.

The screenshot shows a web interface for editing an instructor's profile. On the left is a sidebar menu with options like Dashboard, Cursos, Turmas, Instrutores (highlighted), Clientes, Mensagens, and Configurações. The main content area is titled 'Editar Instrutor' and contains a form with the following fields:

- Nome:** Carlos
- E-mail:** victor.kub@keebler-bechtelar.test
- Celular:** (54) 92762-1457
- Currículo:** A rich text editor containing:
  - Perfil Profissional**
  - Associate Degree in Criminology
  - Experiência**
  - Planejador de Propaganda Distrital na de Lins-Antunes
  - Formação**

**Figura 47 – Tela de edição de instrutor.**

Finalmente, a visualização detalhada do currículo é o ponto onde o rich text faz a diferença. Esta página, apresentada na Figura 48, exibe o currículo formatado exatamente como foi digitado no editor, com todos os recursos visuais preservados. É também nesta tela que o administrador pode associar uma imagem ao instrutor, servindo como sua foto de perfil que aparecerá na listagem e no próprio currículo, utilizando o Active Storage de forma análoga ao perfil do administrador. A remoção de instrutores, por sua vez, é um processo simples, executado diretamente da tabela de instrutores.



Maxiambiental

Maxiambiental

- Dashboard
- Cursos
- Turmas
- Instrutores**
- Clientes
- Mensagens
- Configurações

**Carlos**  
Instructor

**CONTATO**

victor.kub@keebler-becht...  
(54) 92762-1457

**Perfil Profissional**

Associate Degree in Criminology

**Experiência**

Planejador de Propaganda Distrital na de Lins-Antunes

**Formação**

Associate Degree in Communications na Falconholt TAFE

**Figura 48 – Tela de visualização do currículo do instrutor.**

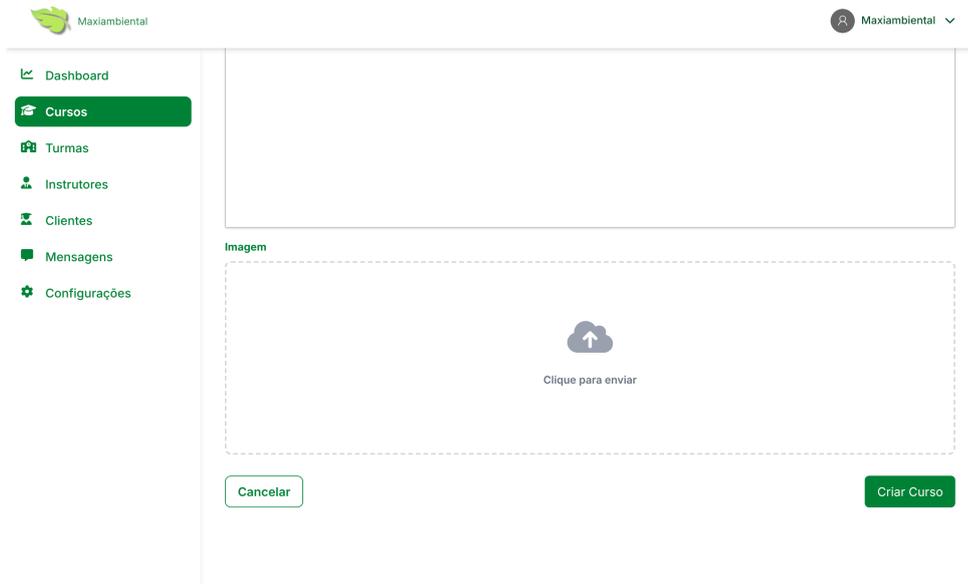
### 5.3.6 Feature: Gerenciamento de cursos

Prosseguindo com a expansão das funcionalidades de gerenciamento pelo administrador, a próxima etapa do desenvolvimento envolveu o CRUD para os cursos do sistema. Esta seção permite que o administrador tenha controle sobre o catálogo de cursos a serem ofertados, desde o cadastro inicial até a eventual remoção.

O ponto de partida para o gerenciamento de cursos é uma tela de listagem, que oferece uma visão geral de todos os cursos cadastrados no sistema. Como ilustrado na Figura 49, esta tela apresenta uma tabela contendo a imagem de cada curso, seu nome, o status do curso, uma opção para excluir o curso e um acesso direto à página de edição. Esta organização é para uma fácil identificação e execução das ações necessárias pelo administrador.

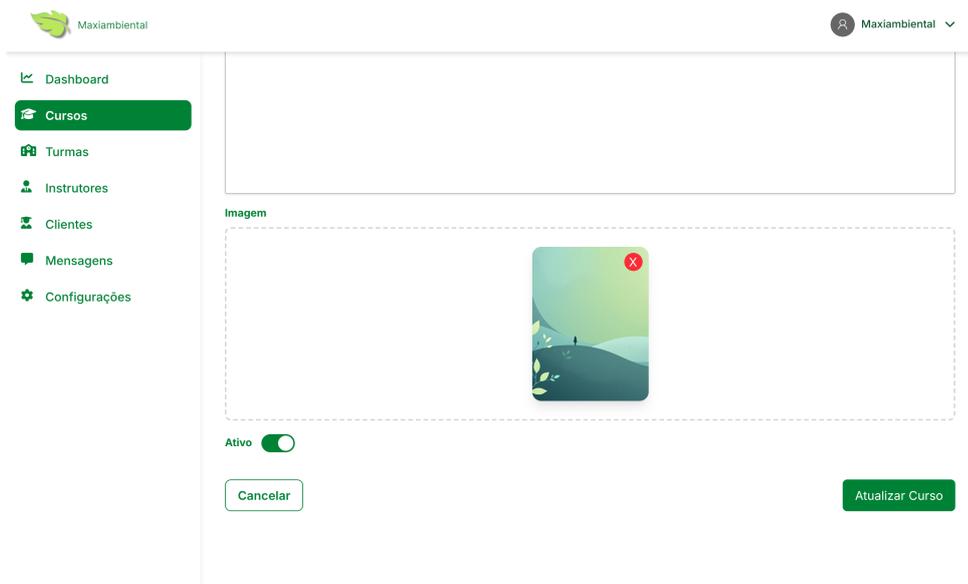


Um destaque na tela de cadastro, e também na de edição, é a customização do *input* para a imagem do curso. Em vez do *input* padrão do navegador, um campo estilizado foi implementado utilizando o *custom input* do *Simple Form* junto do JavaScript. Este campo apresenta bordas tracejadas, indicando visualmente a área para a seleção da imagem. A Figura 51 ilustra este design antes da seleção de qualquer arquivo.



**Figura 51 – Input customizado.**

Ao selecionar uma imagem, um *preview* dinâmico é gerado, permitindo ao administrador visualizar como a imagem do curso será exibida antes de salvar o registro. A Figura 52 demonstra este *preview*, oferecendo um *feedback* visual sobre a aparência do curso.



**Figura 52 – Input customizado preenchido.**

A tela de edição de um curso, apresentada na Figura 53, segue o padrão da tela de cadastro, com os campos já preenchidos com os dados existentes do curso. Assim como na

edição de instrutores, o foco aqui é a alteração do nome e da descrição, sendo a gestão da imagem realizada através do mesmo *input* customizado.

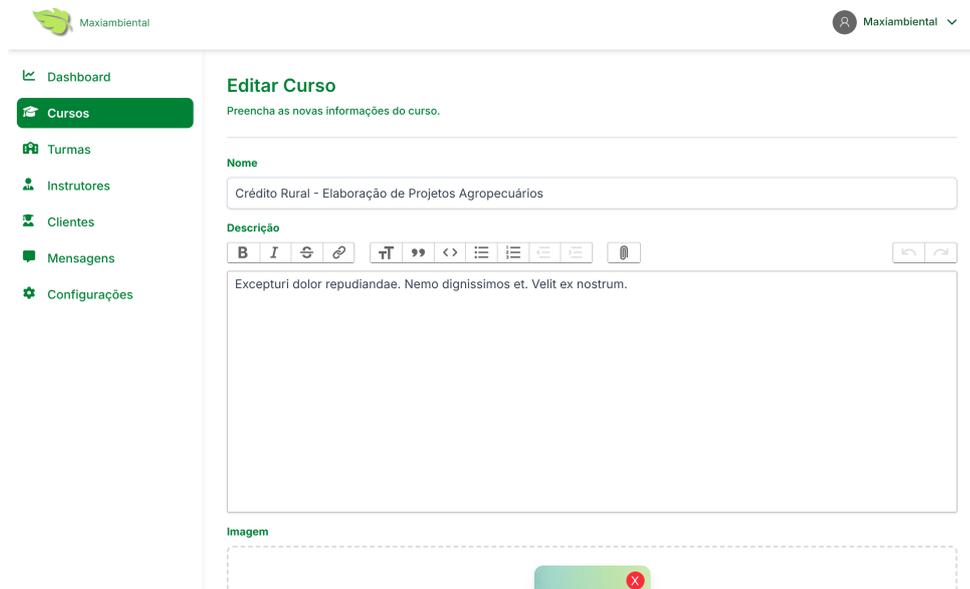


Figura 53 – Tela de edição de curso.

A organização e a apresentação dos cursos não se restringiram à área administrativa. Na *home* da aplicação, área pública, um carrossel foi implementado, Figura 54, para destacar os cursos disponíveis, exibindo até seis deles. Esta funcionalidade visa atrair a atenção dos visitantes e apresentar os cursos disponíveis.



Figura 54 – Cursos mostrados na tela inicial.

Além do carrossel, uma listagem completa dos cursos foi integrada à área pública, como mostra a Figura 55. Cada curso é apresentado em um card simples, contendo sua imagem e nome. Ao passar o cursor sobre o card, a imagem é sutilmente escurecida, e a descrição do curso ganha destaque, oferecendo uma breve visão do conteúdo sem a necessidade de clicar.

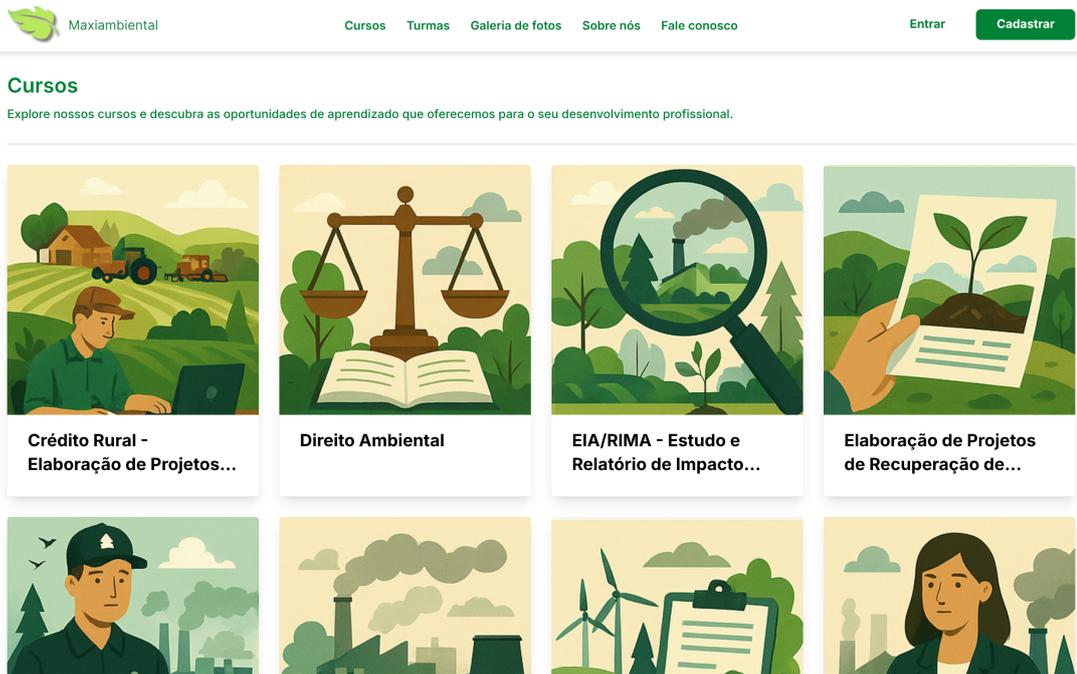


Figura 55 – Tela pública dos cursos.

Para informações mais detalhadas, cada curso possui uma página de detalhes específica, acessível a partir da listagem. Nesta página, ilustrada na Figura 56, são exibidas a imagem do curso em maior destaque, seu nome completo e a descrição detalhada. Adicionalmente, esta página fornece informações sobre as turmas abertas para aquele curso específico.

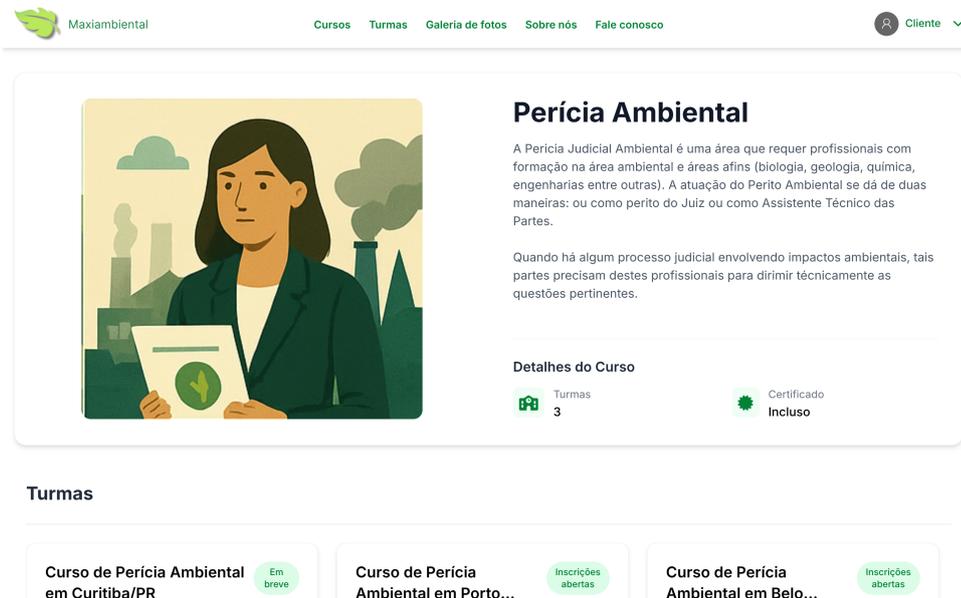


Figura 56 – Tela pública de detalhes do curso.

### 5.3.7 Feature: Gerenciamento de Clientes

Retornando às funcionalidades de gerenciamento do administrador, o sistema foi incluso um fluxo de gerenciamento de clientes. Essa funcionalidade permite o administrador supervisionar as contas dos usuários do sistema.

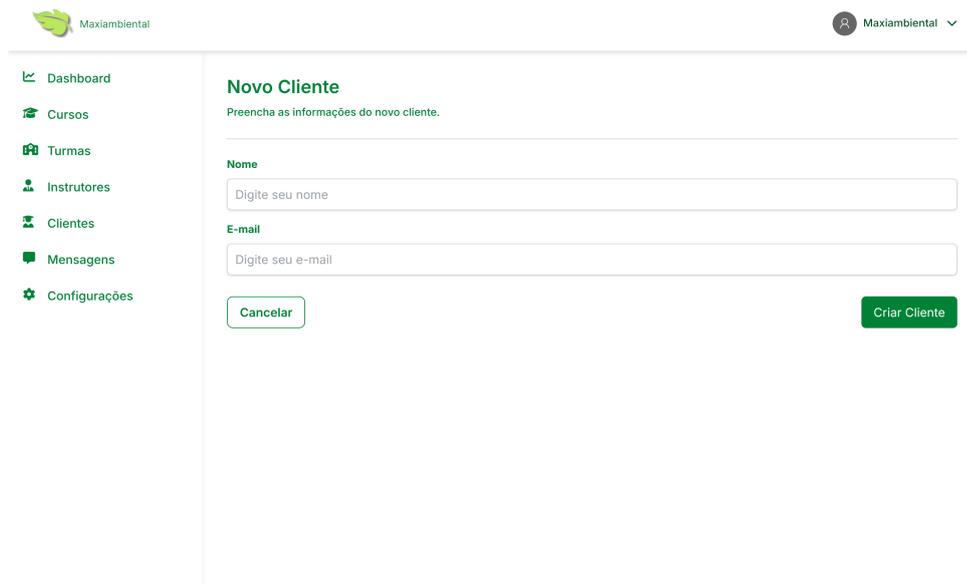
A interface principal para essa gestão é a tela de listagem que traz todos os clientes cadastrados. Como pode ser observado na Figura 57, essa tela oferece recursos de busca que permitem ao administrador localizar clientes específicos por nome ou e-mail. Além de apresentar o status (ativo/inativo) de cada cliente no sistema, a listagem oferece opções diretas para a exclusão de um registro, acesso à página de edição do cliente e um atalho para a tela de cadastro de novos clientes.

The screenshot displays the 'Clientes' management page. On the left is a sidebar with navigation icons for Dashboard, Cursos, Turmas, Instrutores, Clientes, Mensagens, and Configurações. The main area has a header with the 'Maxiambiental' logo and user profile. Below the header, there's a search bar labeled 'Pesquisar...' and a '+ Novo Cliente' button. A table lists clients with columns for 'CLIENTE' and 'STATUS'. The table shows six active clients with their names and email addresses, and icons for editing and deleting each record.

CLIENTE	STATUS
<b>Agatha Lopes</b> nyla.bauch@hoeger-cole.test	Ativo
<b>Alessandro Aroeira</b> bernard.bergstrom@reinger.test	Ativo
<b>Ana Beatriz Teixeira</b> boyce@grant.example	Ativo
<b>Carla Viveiros</b> kasey_kshlerin@veum.example	Ativo
<b>Carolina Silveira Neto</b> antoinette_ortiz@upton.test	Ativo
<b>Davi Luiz da Cruz Filho</b> raelene@bergstrom.test	Ativo

**Figura 57 – Tela de listagem de clientes para o administrador.**

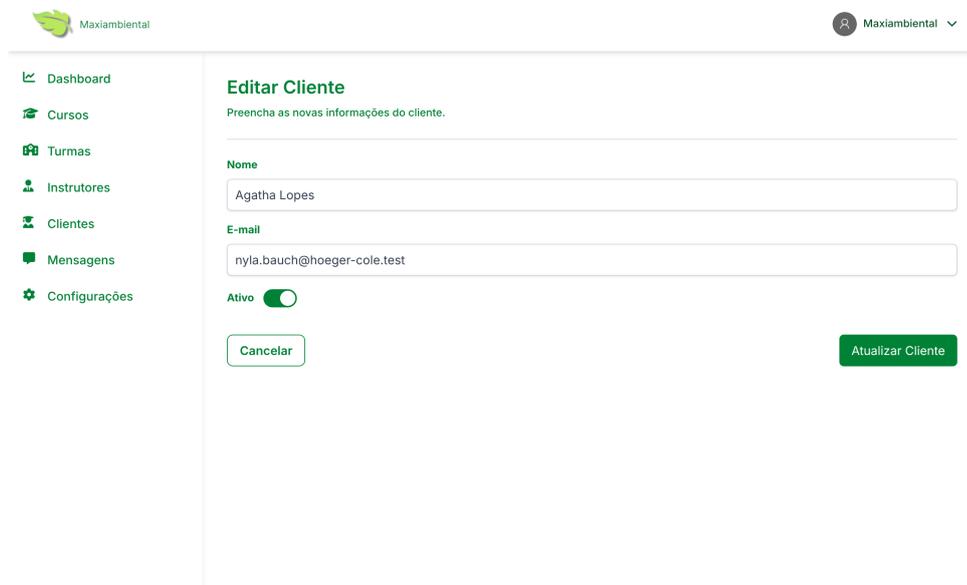
O cadastro de um novo cliente pelo administrador foi implementado para ser um processo simplificado, para apenas ser usado em caso de um novo cliente não conseguir se cadastrar. Nesta tela, exibida na Figura 58, o administrador precisa informar apenas o nome e o e-mail do cliente. Para a segurança da conta, uma senha aleatória é gerada pelo sistema no momento do cadastro. Após a conclusão do registro, um e-mail é enviado ao cliente recém-cadastrado, informando sobre a criação de sua conta. Caso o cliente necessite acessar o sistema, ele pode facilmente utilizar o fluxo de recuperação de senha, já detalhado na subseção 5.3.2, para definir sua própria credencial de acesso.



The screenshot shows the 'Novo Cliente' (New Client) registration form. The interface includes a sidebar with navigation options: Dashboard, Cursos, Turmas, Instrutores, Clientes, Mensagens, and Configurações. The main content area is titled 'Novo Cliente' and contains the instruction 'Preencha as informações do novo cliente.' Below this, there are two input fields: 'Nome' (Name) with the placeholder text 'Digite seu nome' and 'E-mail' with the placeholder text 'Digite seu e-mail'. At the bottom of the form, there are two buttons: 'Cancelar' (Cancel) and 'Criar Cliente' (Create Client).

**Figura 58 – Tela de cadastro simplificado de cliente pelo administrador.**

A edição dos dados de um cliente pelo administrador espelha o processo de cadastro. A tela de edição, mostrada na Figura 59, apresenta os campos de nome e e-mail pré-preenchidos com as informações atuais do cliente, seguindo o padrão já utilizado nas edições de instrutores e cursos. Nesta interface, o administrador tem a capacidade de atualizar apenas essas duas informações.



The screenshot shows the 'Editar Cliente' (Edit Client) form. The interface is similar to the registration form, with the same sidebar and main content area. The main content area is titled 'Editar Cliente' and contains the instruction 'Preencha as novas informações do cliente.' Below this, there are two input fields: 'Nome' (Name) with the pre-filled value 'Agatha Lopes' and 'E-mail' with the pre-filled value 'nyla.bauch@hoeger-cole.test'. Additionally, there is a toggle switch labeled 'Ativo' (Active) which is currently turned on. At the bottom of the form, there are two buttons: 'Cancelar' (Cancel) and 'Atualizar Cliente' (Update Client).

**Figura 59 – Tela de edição de cliente pelo administrador.**

### 5.3.8 Feature: Dashboard do Administrador

Para oferecer ao administrador uma melhor visão do sistema, foi implementada uma tela de dashboard, que serve como a página inicial para o administrador. Este dashboard foi montado pensando em quais dados da aplicação seriam úteis para o administrador.

O dashboard é composto por três cartões de estatísticas, que apresentam a quantidade de cursos finalizados, novas inscrições e novos clientes no período dos últimos 30 dias. Além da contagem absoluta, cada cartão exibe a porcentagem de crescimento ou decréscimo desses números em comparação com os 30 dias anteriores. Essa visualização comparativa permite ao administrador identificar e avaliar o desempenho do sistema.

Para a reusabilidade e a consistência visual desses cartões, um View Component específico, o `Dashboard::StatsComponent`, foi desenvolvido. Isso significa que, para exibir um novo conjunto de estatísticas, basta passar os dados e um texto descritivo para o componente, como mostrado na view:

```
1 # app/components/dashboard/stats_component.rb
2 class Dashboard::StatsComponent < ViewComponent::Base
3   def initialize(stats:, text: '', id: 'stats')
4     @stats = stats # Hash com :count e :growth
5     @text = text
6     @id = id
7   end
8   # Métodos como #count, #growth, #style_color, #style_icon, #formatted_growth
9   # são implementados para exibir os dados e estilizar o componente.
10 end
```

Figura 60 – Código: Componente Stats.

O template HTML do componente utiliza as propriedades calculadas para renderizar visualmente o cartão, como pode ser visto abaixo:

```
1 <%= # app/components/dashboard/stats_component.html.erb %>
2 <div id="<%= @id %>" class="flex items-center p-4 bg-white rounded shadow-xl">
3   <div class="flex flex-shrink-0 items-center justify-center bg-<%= style_color %>-200 h-
4     16 w-16 rounded">
5     <%= icon('fa-solid', style_icon, class: "fa-xl fill-current text-#{style_color}-700")
6     %>
7   </div>
8   <div class="flex-grow flex flex-col ml-4">
9     <span id="count" class="text-xl font-bold"><%= count %></span>
10    <div class="flex items-center justify-between">
11      <span id="text" class="text-gray-500"><%= @text %></span>
12      <span id="growth" class="text-<%= style_color %>-500 text-sm font-semibold ml-2">
13        <%= formatted_growth %></span>
14    </div>
15  </div>
16 </div>
```

Figura 61 – Código: Componente Stats View.

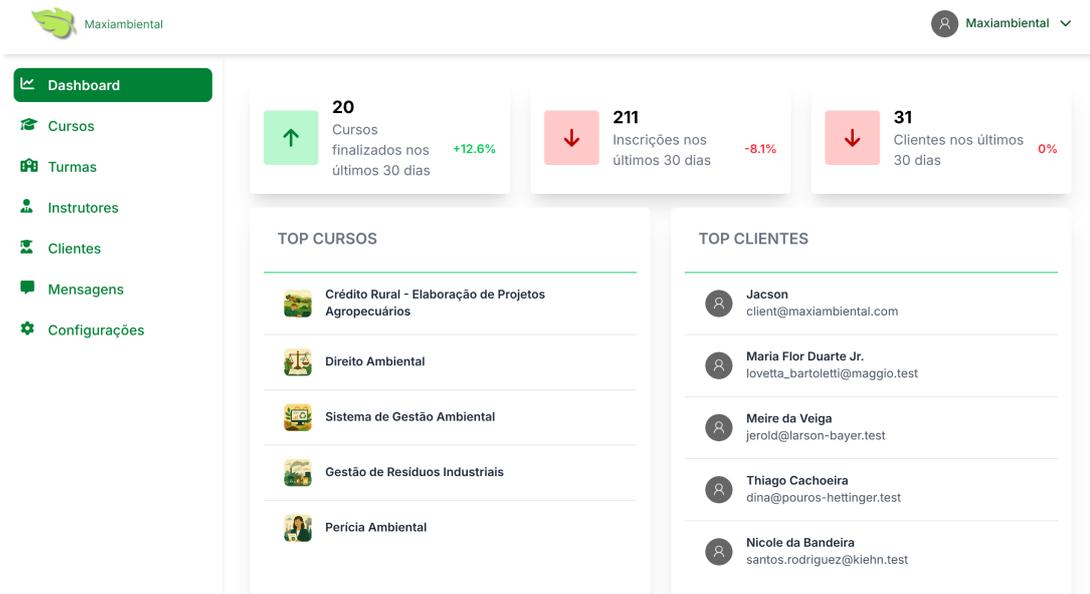
Além dos cartões de estatísticas, o dashboard também apresenta duas tabelas resumidas, exibindo o Top 5 Clientes (aqueles com mais inscrições em cursos) e o Top 5 Cursos (aqueles com o maior número de inscritos). Essas tabelas fornecem um panorama de popularidade dos cursos e da atividade dos clientes.

A lógica para o tratamento dos dados é encapsulada em um service. Essa abordagem de separação de responsabilidades faz com que o controlador se mantenha focado apenas na orquestração da view, enquanto o service lida com a complexidade da consulta e cálculo das métricas:

```
1 # app/services/dashboard/admin.rb
2 class Dashboard::Admin
3   def clients_metrics
4     {
5       count: clients_last_30_days_count,
6       growth: clients_percentage_increase
7     }
8   end
9
10  # ... Outros métodos
11
12  private
13
14  def clients_last_30_days_count
15    Client.where(created_at: 30.days.ago..).count
16  end
17
18  def clients_previous_30_days_count
19    Client.where(created_at: 60.days.ago..30.days.ago).count
20  end
21
22  def clients_percentage_increase
23    previous_count = clients_previous_30_days_count
24    return 0 if previous_count.zero?
25
26    increase = clients_last_30_days_count - previous_count
27    ((increase.to_f / previous_count) * 100).round(1)
28  end
29 end
```

Figura 62 – Código para o dashboard encapsulado em um service.

A Figura 63 ilustra a interface completa do dashboard do administrador, com os cartões de estatísticas e as tabelas de top 5, fornecendo uma visão abrangente e funcional do desempenho do sistema.



**Figura 63 – Interface do Dashboard do Administrador.**

### 5.3.9 Feature: Gerenciamento de Turmas

Com a estrutura de cursos e instrutores estabelecidas, foi possível adicionar um Gerenciamento de Turmas, uma funcionalidade que permite ao administrador organizar e controlar as ofertas de cursos. Cada curso pode possuir múltiplas turmas, cada uma com suas particularidades de horário, local e condições. Este fluxo segue o padrão já utilizado para instrutores, cursos e clientes, oferecendo listagem, cadastro, edição e remoção. No entanto, a complexidade das turmas é maior devido ao volume de atributos e relacionamentos.

A tela de listagem de turmas para o administrador mantém o padrão já presente em outras seções do sistema. Nela, o administrador encontra uma tabela que organiza todas as turmas cadastradas. Para a busca, é possível pesquisar turmas tanto pelo nome da turma quanto pelo nome do instrutor responsável. A paginação lida com grandes volumes de dados, e cada linha da tabela oferece botões de remoção e acesso direto à tela de edição de uma turma existente. A Figura 64 ilustra essa visão geral de gerenciamento.

**Turmas**  
Gerencie todas as turmas cadastradas na Maxiambiental.

+ Nova Turma

Pesquisar...

Total de 4 turmas

TURMA	CURSO	INSTRUTOR	STATUS DAS INSCRIÇÕES	ATIVO
Curso de Perícia...	Perícia Ambiental	Enzo Miguel	Inscrições abertas	Ativo
Sistema de Gestão...	Sistema de Gestão...	Salvador	Inscrições abertas	Ativo
Fiscalização Ambient...	Fiscalização Ambiental	Louise	Inscrições abertas	Ativo
Estudo e Relatório	EIA/RIMA - Estudo e...	Luana	Inscrições abertas	Ativo

**Figura 64 – Tela de listagem de turmas no painel administrativo.**

O cadastro de uma nova turma é um processo detalhado, dividindo as informações em seções para uma melhor organização, dada a grande quantidade de atributos. A primeira parte, focada nas Informações Básicas, é mostrada na Figura 65. Nela, o administrador define o nome da turma, o número de vagas disponíveis, e seleciona o instrutor e o curso associados através de campos select. Um campo importante nesta seção é o status de inscrição no curso, que define a fase atual das inscrições para aquela turma (como "Em breve", "Inscrições abertas", "Em andamento", "Inscrições encerradas" ou "Lista de espera"), controlando a visibilidade e a possibilidade de inscrição para os clientes.

**Nova Turma**  
Preencha as informações da nova turma.

**Informações Básicas**

**Nome**  
Digite o nome da turma

**Vagas disponíveis**  
Digite o número de vagas disponibilizadas

**Instrutor**  
Selecione o instrutor

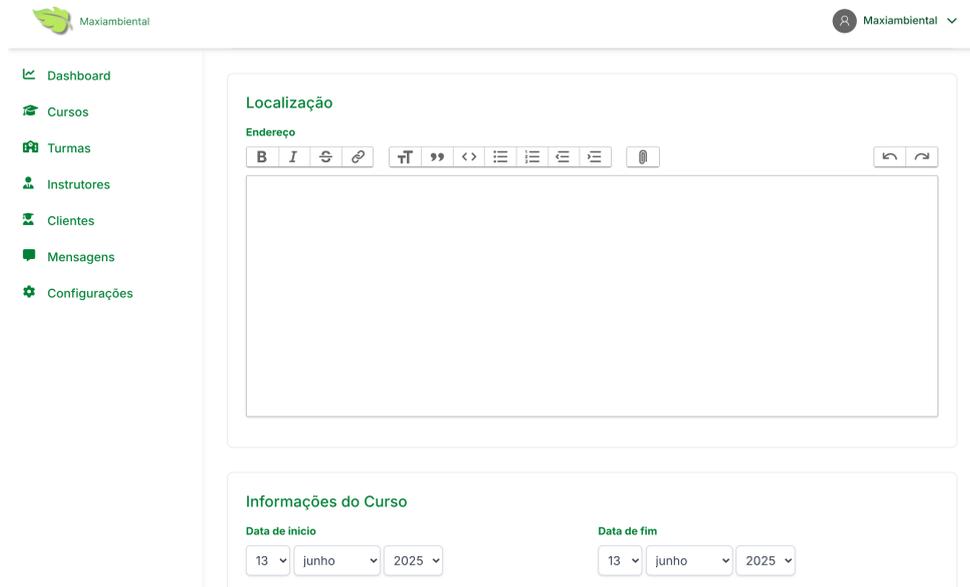
**Curso**  
Selecione o curso

**Status das inscrições**  
Em breve

**Ativo**

**Figura 65 – Tela de cadastro de turma - Informações Básicas.**

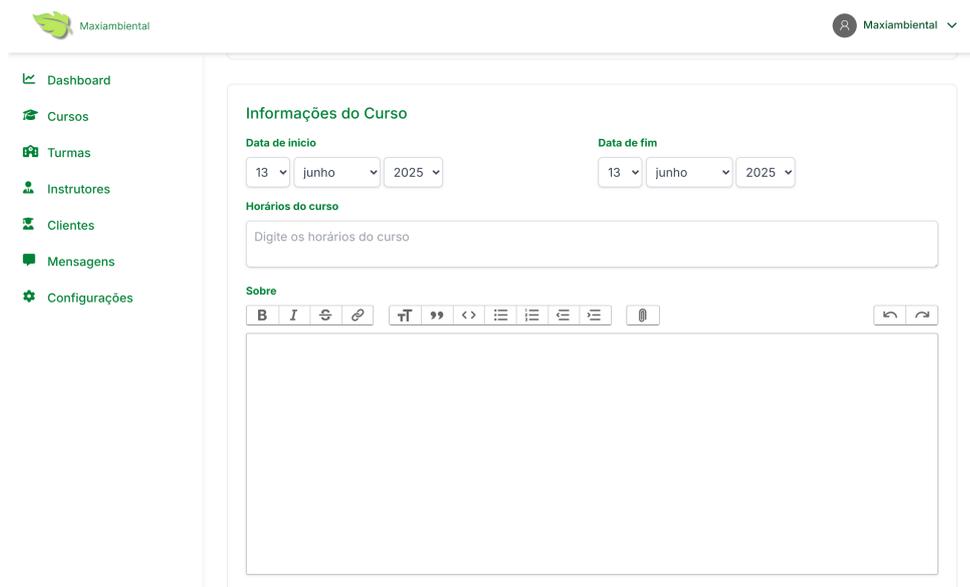
Na sequência, a seção de Localização, apresentada na Figura 66, permite ao administrador informar o endereço onde a turma será ministrada. Este campo utiliza o Action Text, oferecendo flexibilidade para formatar o endereço com detalhes relevantes, como pontos de referência ou instruções de acesso.



The screenshot shows a web interface for course registration. On the left is a sidebar menu with options: Dashboard, Cursos, Turmas, Instrutores, Clientes, Mensagens, and Configurações. The main content area is titled 'Localização' and contains a text editor labeled 'Endereço'. The text editor has a toolbar with icons for bold (B), italic (I), link, unlink, text color, background color, bulleted list, numbered list, indent, and outdent, along with undo and redo buttons. Below the text editor is a section titled 'Informações do Curso' with two date pickers: 'Data de início' (13, junho, 2025) and 'Data de fim' (13, junho, 2025).

**Figura 66 – Tela de cadastro de turma - Localização (Action Text).**

As Informações do Curso, detalhadas na Figura 67, abrangem as datas de início e fim da turma, os horários específicos das aulas e um campo "Sobre". Este último, também um Action Text, destina-se a fornecer detalhes adicionais sobre a turma, como objetivos específicos ou pré-requisitos, complementando a descrição geral do curso.



The screenshot shows the 'Informações do Curso' section of the course registration form. It features two date pickers for 'Data de início' (13, junho, 2025) and 'Data de fim' (13, junho, 2025). Below these is a text input field labeled 'Horários do curso' with the placeholder text 'Digite os horários do curso'. At the bottom is an 'Sobre' section with an Action Text editor, including a toolbar with icons for bold (B), italic (I), link, unlink, text color, background color, bulleted list, numbered list, indent, and outdent, along with undo and redo buttons.

**Figura 67 – Tela de cadastro de turma - Informações do Curso (Datas, Horários, Sobre).**

A próxima seção de Conteúdo, visível na Figura 68, apresenta um campo programação. Este é também um Action Text, necessário para descrever detalhadamente o cronograma das aulas, tópicos a serem cobertos, e outras informações pertinentes.

The screenshot shows a web interface for 'Maxiambiental'. On the left is a sidebar menu with items: Dashboard, Cursos, Turmas, Instrutores, Clientes, Mensagens, and Configurações. The main content area is titled 'Conteúdo' and contains a 'Programação' field. This field has a rich text editor toolbar with icons for bold (B), italic (I), link, unlink, text color, background color, bulleted list, numbered list, indent, and outdent. Below the toolbar is a large empty text area for entering the program details. Below the 'Programação' field is a 'Termos' section, which is currently empty.

**Figura 68 – Tela de cadastro de turma - Conteúdo (Programação - Action Text).**

Por fim, a seção de Termos, ilustrada na Figura 69, é dedicada aos Termos de Aceitação. Este é outro campo Action Text, onde o administrador pode inserir os termos e condições específicos que o cliente precisará aceitar no momento da inscrição na turma.

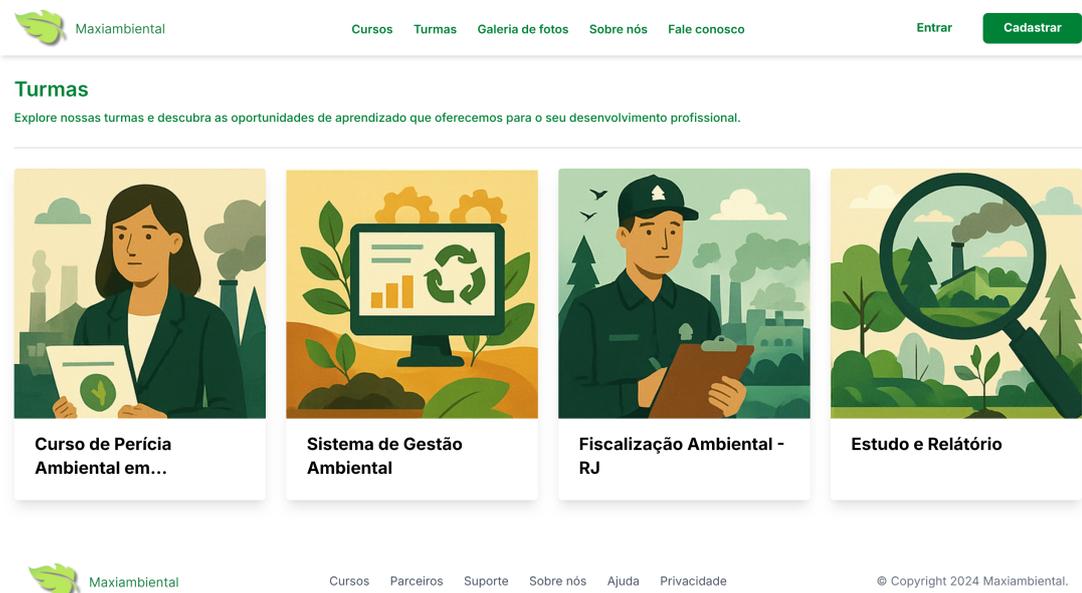
This screenshot shows the same 'Cadastro de turma' page, but with the 'Termos' section active. The 'Termos de aceitação' field is visible, featuring the same rich text editor toolbar as the 'Programação' field. Below the text area, there are two buttons: a light green 'Cancelar' button and a dark green 'Criar Turma' button.

**Figura 69 – Tela de cadastro de turma - Termos de Aceitação (Action Text).**

A tela de edição de turmas é um espelho fiel da tela de cadastro, com todos os campos pré-preenchidos, como mostra a Figura 70.

**Figura 70 – Tela de edição de uma turma.**

Além das funcionalidades exclusivas para o administrador, o gerenciamento de turmas se estende para a área pública da aplicação. A tela de listagem de turmas pública, apresentada na Figura 71, exibe as turmas cadastradas em um formato de cards simples.



**Figura 71 – Tela de listagem de turmas na área pública.**

Ao selecionar uma turma, o cliente é direcionado para uma tela de detalhes específica, mostrada na Figura 72. Nesta página, a imagem do curso associado é exibida, seguida pelo nome da turma e seus detalhes essenciais, como datas, horários e localização. Um elemento importante desta tela é o acesso à ficha de inscrição, que, por regra de negócio, só é habilitado se o status da turma estiver definido como "Inscrições abertas".



**Curso de Perícia Ambiental em Curitiba/PR**

**Detalhes**

Datas: 02/06/2025 - 02/08/2025

Horários: Das 08:00 h às 12:00 h e das 13:30 h às 18:00 h

Local: Hotel Nacional Inn - Rua Lourenço Pinto, 458, Centro em Curitiba/PR

**Ficha de inscrição**

**Sobre** | Programação | Profissionais | Termos e condições

**Objetivo:** Capacitar os profissionais para a atuação como Peritos Judiciais e Assistentes Técnicos na área ambiental através do conhecimento dos procedimentos, práticas e aspectos legais aplicáveis.

**Público Alvo:** Profissionais e estudantes com formação na área ambiental e afins como Químico; Biólogo; Geólogo; Geógrafo; Arquiteto; Arquiteto e Urbanista; Engenheiros de todas as modalidades: Ambiental, Agrônomo, Sanitarista, Agrícola, Florestal, Civil, Químico, de Minas; Médico Veterinário; Zootecnistas, Gestor Ambiental, Técnico em Meio Ambiente entre outras formações relacionadas à área ambiental e áreas que tenham interface com a área ambiental. O curso também é recomendado para advogados que acompanham ou pretendem acompanhar ações de natureza ambiental.

**Figura 72 – Tela de detalhes de uma turma na área pública.**

Abaixo dessas informações básicas, a tela de detalhes da turma aprofunda-se no conteúdo através de diferentes abas. Essas abas permitem ao cliente navegar por diferentes seções informativas do curso sem sair da página, como:

- **Sobre:** Exibe a descrição detalhada da turma.
- **Programação:** Apresenta a programação completa da turma.
- **Instrutor:** Fornece detalhes sobre o instrutor responsável pela turma.
- **Termos e Condições:** Mostra os termos de aceitação específicos para aquela turma.

#### 5.3.10 Feature: Inscrição de Clientes em Turmas

A funcionalidade de inscrição de clientes em turmas foi planejada para ser um formulário *multi-step* (passo a passo). Este design guia o cliente através do processo de matrícula, coletando informações em etapas. O cliente acessa a ficha de inscrição diretamente pela página de detalhes da turma de seu interesse, mas apenas se o status da turma estiver como "Inscrições abertas".

O processo de inscrição foi dividido em quatro etapas principais. O primeiro passo, ilustrado na Figura 73, foca na validação e complementação dos dados pessoais do cliente. Campos como nome, e-mail (desabilitado para edição, pois já está associado à conta do cliente), CPF, celular, CEP, cidade, estado e endereço são apresentados.

Maxiambiental Cursos Turmas Galeria de fotos Sobre nós Fale conosco Cliente

### Formulário de inscrição

Curso de Perícia Ambiental em Porto Alegre/RS

1 Dados pessoais >> 2 Dados do curso >> 3 Pagamento >> 4 Confirmação

**Nome**  **CPF**

**E-mail**  **Celular**

**CEP**  **Cidade**  **Estado**

**Endereço**

**Figura 73 – Passo 1 da Inscrição: Informações Pessoais.**

Após a confirmação dos dados pessoais, o cliente avança para a etapa de informações adicionais, conforme a Figura 74. Nela, são coletados dados como "como ficou sabendo da Maxiambiental", "se já participou de algum curso anterior" e "qual a categoria o cliente se encaixa" (estudante ou instrutor). Há também um campo opcional para "observações". O ponto mais importante desta etapa, e de toda inscrição, é a aceitação dos termos da turma, um requisito obrigatório para prosseguir com a inscrição.

Maxiambiental Cursos Turmas Galeria de fotos Sobre nós Fale conosco Cliente

1 Dados pessoais >> 2 Dados do curso >> 3 Pagamento >> 4 Confirmação

**Como ficou sabendo do curso?**

**Já participou de um curso/palestra da Maxiambiental?**  
 Sim  Não

**Categoria**

**Observações**

**Termos de aceitação**

**Informações Adicionais:**  
**Investimento:**  
**Profissionais:** Taxa de matrícula R\$ 100,00 + R\$ 650,00  
**Estudantes:** Taxa de matrícula R\$ 100,00 + R\$ 550,00

**Formas de pagamento:**  
 \*O pagamento da taxa de matrícula no valor de R\$100,00 será realizado através de

Li e aceito os termos e condições

**Figura 74 – Passo 2 da Inscrição: Informações Adicionais e Termos.**

Com todas as informações necessárias preenchidas e os termos aceitos, o cliente segue para a última etapa: a seleção da forma de pagamento. É importante ressaltar que, por não fazer

parte do escopo do projeto a integração com pagamento real, esta etapa é uma simulação, como mostra a Figura 75. O cliente simplesmente seleciona um método de pagamento e clica em "Confirmar".

The screenshot shows the 'Payment method' selection screen. A dropdown menu is set to 'Boleto'. Below it, the heading 'Pagamento via Boleto' is displayed, followed by the instruction 'Pague em qualquer agência bancária ou internet banking'. A summary box contains the following details:

Banco Fake:	341-7
Valor:	R\$ 150,00
Vencimento:	15/08/2023
Código de Barras:	34191.79001 01043.510047 91020.150008 7 87650000015000

A yellow banner below the summary box states: 'Este é um boleto simulado para testes.' At the bottom, there are two buttons: 'Voltar' (left) and 'Confirmar' (right).

**Figura 75 – Passo 3 da Inscrição: Seleção da Forma de Pagamento (Simulação).**

Após a confirmação do pagamento simulado, o cliente é redirecionado para o último passo, que é uma tela de confirmação, ilustrada na Figura 76. Esta tela finaliza o processo, indicando que a inscrição foi realizada com sucesso e que o cliente está agora inscrito na turma.

The screenshot shows the 'Confirmation' screen. At the top, a green banner with a white checkmark icon and the text 'Pagamento Aprovado!' is displayed. Below this, the text reads: 'Obrigado por sua compra! Um e-mail de confirmação foi enviado para você'. A 'Detalhes do Pagamento' box contains the following information:

Método:	PIX
Valor:	R\$ 149,90
Data:	15 de junho, 20:38
Código:	PIX - JEQF7MX

Below the details box, a blue banner with an information icon states: 'Esta é uma simulação para fins de desenvolvimento.' At the bottom, there is a green button labeled 'Ver Comprovante'.

**Figura 76 – Passo 4 da Inscrição: Confirmação Final.**

Além do formulário de inscrição, foi adicionada uma página para o cliente visualizar suas inscrições. Esta página, apresentada na Figura 77, exibe as inscrições em uma tabela, mostrando o nome da turma, o período (datas de início e fim) e os status de inscrição e pagamento.

The screenshot displays the 'Minhas Inscrições' (My Enrollments) page. At the top, there is a navigation bar with the Maxiambiental logo and links for 'Cursos', 'Turmas', 'Galeria de fotos', 'Sobre nós', and 'Fale conosco'. A user profile icon labeled 'Cliente' is visible in the top right. Below the navigation bar, the page title 'Minhas Inscrições' is followed by the subtitle 'Gerencie todas as inscrições cadastradas na Maxiambiental.' A search bar with the placeholder 'Pesquisar...' and a magnifying glass icon is located below the subtitle. To the right of the search bar, it indicates 'Total de 2 inscrições'. The main content is a table with the following data:

TURMA	PERÍODO	INSCRIÇÃO	STATUS	PAGAMENTO
 Curso de Perícia Ambiental em Porto Alegre/RS	15/06/2025 - 15/08/2025	15 de junho, 20:38	Confirmado	Confirmado
 Fiscalização Ambiental - RJ	15/06/2025 - 15/08/2025	15 de junho, 16:52	Confirmado	Confirmado

The footer contains the Maxiambiental logo, navigation links for 'Cursos', 'Parceiros', 'Suporte', 'Sobre nós', 'Ajuda', and 'Privacidade', and the copyright notice '© Copyright 2024 Maxiambiental.'

**Figura 77 – Tela de listagem de inscrições do cliente.**

Para a implementação da lógica do formulário multi-step foi utilizado o padrão Form Object. No Rails, um Form Object é uma classe Ruby simples que encapsula a lógica de um formulário que não se mapeia diretamente para um único modelo do ActiveRecord. Ele atua como uma camada intermediária entre o controlador e os modelos de domínio, centralizando validações, atributos do formulário e a lógica de persistência.

Neste projeto, um Form Object "pai", `Enrollments::CourseEnrollmentForm`, foi criado para gerenciar todo o fluxo. Este objeto principal coordena outros Form Objects menores (`ClientForm`, `EnrollmentForm`, `PaymentForm`), cada um responsável por uma parte específica dos dados do formulário multi-step.

```

1 # app/forms/enrollments/course_enrollment_form.rb
2 class Enrollments::CourseEnrollmentForm
3   include ActiveRecord::Model
4
5   STEPS = [:client, :enrollment, :payment, :confirmation].freeze
6
7   attr_reader :current_step
8   attr_accessor :client, :course_class
9
10  def initialize(course_class:, client: nil, attributes: {})
11    @current_step = (attributes[:current_step] || :client).to_sym
12    @client = client
13    @course_class = course_class
14    initialize_forms(attributes)
15  end
16
17  # Métodos como #valid?, #move_to_next_step, #save, #session_attributes
18  # coordenam a navegação entre os passos, validação e persistência dos dados,
19  # delegando a responsabilidade aos Form Objects filhos.
20  # ... (código resumido) ...
21
22  private
23
24  attr_reader :client_form, :enrollment_form, :payment_form
25
26  def initialize_forms(attributes)
27    @client_form = Enrollments::ClientForm.new(@client, attributes[:client_data] || {})
28    @enrollment_form = Enrollments::EnrollmentForm.new(attributes[:enrollment_data] || {})
29    @payment_form = Enrollments::PaymentForm.new(attributes[:payment_data] || {})
30  end
31
32  def persist_data
33    ActiveRecord::Base.transaction do
34      client_form.update
35      enrollment = enrollment_form.create(client, course_class)
36      payment_form.create(enrollment)
37    end
38  end
39 end

```

Figura 78 – Lógica do Form Object.

Este padrão permitiu manter a lógica de gerenciamento dos passos do formulário fora do controlador, que agora se concentra apenas na orquestração da sessão e na interação com o Form Object pai. Além disso, evitou a adição de lógica extra e atributos diretamente nos modelos, mantendo-os limpos e focados em suas responsabilidades primárias. O `Enrollments::EnrollmentForm`, por exemplo, cuida das validações e atributos específicos da inscrição:

```

1 # app/forms/enrollments/enrollment_form.rb
2 class Enrollments::EnrollmentForm < BaseForm
3   delegate :categories, :referral_sources, to: :Enrollment
4
5   attr_accessor :category, :referral_source, :notes, :terms_accepted, :previous_participation
6
7   validates :category, presence: true,
8             inclusion: { in: ->(form) { form.categories.keys.map(&:to_s) } }
9
10  validates :referral_source, presence: true,
11                        inclusion: { in: ->(form) { form.referral_sources.keys.map(&:to_s) } }
12
13  validates :terms_accepted, acceptance: true
14
15  # ... (outros métodos como #create, #attributes, #params para Strong Parameters, #options para
16  # selects) ...
17 end

```

Figura 79 – Lógica do Form Object.

## 5.4 Atualizações e Melhorias

Nesta seção, são abordadas as principais atualizações e aprimoramentos técnicos realizados na aplicação. Inicialmente, apresenta-se a atualização das dependências do projeto, garantindo maior segurança e compatibilidade. Em seguida, são detalhadas as funcionalidades implementadas para busca e paginação, que otimizam a navegação e acesso aos dados. Também são descritos processos de popular o banco de dados (`populate`) para facilitar testes e demonstrações. Complementando, são apresentadas melhorias visuais e funcionais, como a configuração do `favicon` e do título completo da aplicação, a implementação de Progressive Web App (PWA) para maior acessibilidade, e a configuração de metadados Open Graph (OG) para melhor integração com redes sociais.

### 5.4.1 Atualizações de Dependências

Esta etapa concentrou-se na atualização de dependências importantes para a aplicação: o *Rails* e o *118n* foram atualizados para a versão 8, enquanto o *Tailwind CSS* migrou para a versão 4. Essas atualizações garantem acesso a otimizações, reforços de segurança e novas funcionalidades.

A transição para o *Rails 8* introduziu melhorias no tratamento de parâmetros em controladores. Anteriormente, utilizava-se:

```
params.require(:model).permit(:name)
```

Na nova abordagem, emprega-se:

```
params.expect(model: [:name])
```

Esta mudança simplificou essa camada extra de segurança que exige que a estrutura dos dados corresponda exatamente ao esperado, gerando erros em caso de divergências e prevenindo o envio de parâmetros não autorizados.

Outra melhoria foi a inclusão no arquivo `config/application.rb`:

```
config.active_support.to_time_preserves_timezone = :zone
```

Essa configuração resolve inconsistências anteriores no manejo de fusos horários. Ao converter strings temporais para objetos `Time`, o sistema agora preserva o fuso horário original da string ou assume `config.time_zone` quando não especificado, garantindo integridade em cenários multiregionais.

Para a *gem* de internacionalização (*118n*), a atualização para a versão 8 manteve compatibilidade com o sistema existente, proporcionando apenas aprimoramentos internos de desempenho e correções.

Quanto ao *Tailwind CSS 4*, destacam-se mudanças estruturais:

- **Configuração simplificada:** O arquivo `config/tailwind.config.js` tornou-se opcional para configurações padrão
- **Reorganização de diretórios:** O ponto de entrada principal migrou de `assets/stylesheets/application.tailwind.css` para `assets/tailwind/application.css`
- **Isolamento de estilos:** Diretório dedicado (`assets/tailwind`) para arquivos processados pelo *framework*, separando-os de estilos convencionais (`assets/stylesheets`)

#### 5.4.2 Implementação de Busca e Paginação

Para melhorar a navegação em tabelas extensas (como listas de instrutores e cursos), foi necessário a implementação de uma solução que combina funcionalidades de busca e paginação. Essa abordagem permite que usuários localizem informações rapidamente e naveguem facilmente por grandes volumes de dados.

O ponto de partida foi o desenvolvimento de um componente de busca que permite filtragem por atributos específicos de determinado modelo em uma listagem. A implementação utiliza **ViewComponent** para encapsular a lógica do formulário:

```
# app/components/search/form_component.rb
class Search::FormComponent < ViewComponent::Base
  def initialize(base_url:)
    @base_url = base_url
  end
end
```

Com a interface renderizada através de um template simples:

```
<%# app/components/search/form_component.html.erb %>
<div class="w-full">
  ...
  <%= f.input :term,
    label: false,
    wrapper_html: { class: "col-span-11" },
    input_html: input_options %>
  ...
</div>
```

A interação é gerenciada por um controlador Stimulus que processa a submissão e redireciona para a URL filtrada:

```

1 // app/javascript/controllers/search_controller.js
2 export default class extends Controller {
3   search() {
4     const term = this.inputTarget.value.trim();
5     if (term) {
6       window.location.href =
7         `${this.baseUrlValue}/search/${encodeURIComponent(term)}`;
8     }
9   }
10 }

```

Figura 80 – Código: Controlador javascript do componente de busca.

Complementando a busca, foi implementado a paginação utilizando a *gem Kaminari*, que divide resultados em páginas gerenciáveis:

```

1 <%=# Visão de lista de instrutores %>
2 <%= paginate @instructors %>
3
4 <div class="mt-2 text-sm text-gray-600">
5   <%= page_entries_info @instructors %>
6 </div>

```

Figura 81 – Código: Uso da *gem Kaminari*.

Essa combinação oferece feedback visual através de mensagens como *"Mostrando 1 - 10 de 52 instrutores"*, informando o escopo dos dados visíveis. A navegação por páginas é mais rápida, pois carrega subconjuntos de dados, enquanto a busca direta por termos específicos elimina a necessidade de percorrer manualmente listas extensas, como mostra a Figura 82.

The screenshot shows a web application interface for 'Maxiambiental'. On the left is a sidebar with navigation links: Dashboard, Cursos, Turmas, Instrutores (highlighted), Clientes, Mensagens, and Configurações. The main content area is titled 'Instrutores' and includes a search bar with the text 'isadora' and a search icon. Below the search bar is a table with the following data:

INSTRUTOR	CELULAR	STATUS	CURRICULO
 Isadora desiree@zboncak-mckenzie.example	(51) 96174-5162	Ativo	Visualizar Currículo

At the top right of the table, it says 'Total de 1 instrutor'. There is also a '+ Novo Instrutor' button at the top right of the main content area.

Figura 82 – Componente de busca integrado ao cabeçalho da tabela

### 5.4.3 Populate

Para facilitar o desenvolvimento e os testes da aplicação em ambiente local, foi implementado um populate. Essa funcionalidade permite preencher o banco de dados com informações simuladas, mas com aparência de dados reais. Isso é útil para testar a interface do usuário, as listagens e as interações com grandes conjuntos de dados, sem a necessidade de inserção manual repetitiva.

O populate é executado via um Rake task (lib/tasks/populate.rake), que orquestra a criação de dados para diferentes modelos. Após a execução, o comando rails db:populate limpa o banco de dados e o preenche com as novas informações. No projeto, foram configurados três populators distintos: para Clientes, Instrutores e Cursos, criando 30 clientes, 20 instrutores e 9 cursos. O código, Figura 83, ilustra a estrutura do Rake task.

```

1 # lib/tasks/populate.rake
2
3 Rails.root.glob('lib/populators/*.rb').each { |file| require file }
4
5 namespace :db do
6   desc 'Erase and fill database'
7   task populate: :environment do
8     include Populators
9     ActiveSupport.logger = Logger.new($stdout)
10    ActiveSupport.logger.level = Logger::WARN
11
12    puts 'Running populate'
13
14    ClientPopulate.populate(30)
15    InstructorPopulate.populate(20)
16    CoursesPopulate.populate(9)
17
18    Rake::Task['db:seed'].invoke
19  end
20 end

```

**Figura 83 – Código para o Rake Populate.**

A criação dos dados simulados é construída utilizando as gems Factory Bot<sup>22</sup> e Faker<sup>23</sup>. O Factory Bot permite definir "fábricas" para cada modelo, especificando como os objetos devem ser criados, enquanto o Faker é utilizado para gerar valores realistas e aleatórios para os atributos (nomes, e-mails, telefones, etc.).

Um exemplo de definição de fábrica para o modelo Instructor, Figura 84, demonstra como esses dados são gerados:

<sup>22</sup> Factory Bot é uma biblioteca para Ruby que fornece uma maneira simples de criar objetos de teste. [https://github.com/thoughtbot/factory\_bot]

<sup>23</sup> Faker é uma biblioteca Ruby para gerar dados falsos (nomes, endereços, e-mails, etc.) para fins de desenvolvimento e teste. [https://github.com/faker-ruby/faker]

```

1 # tests/factories/instructors.rb
2
3 FactoryBot.define do
4   factory :instructor do
5     name { Faker::Name.name }
6     email { Faker::Internet.unique.email }
7     phone { Faker::PhoneNumber.numerify('(##) 9###-####') }
8
9     transient do
10      resume_sections { [:profile, :experience, :education] }
11    end
12
13    after(:build) do |instructor, evaluator|
14      instructor.resume.body = ResumeContentBuilder.generate(*evaluator.resume_sections)
15    end
16  end
17 end

```

**Figura 84 – Código para a fábrica do instrutor.**

Neste exemplo, `Faker::Name.name` e `Faker::Internet.unique.email` são utilizados para gerar nomes e e-mails aleatórios e únicos. A estrutura *transient* do e *after(:build)* é empregada para criar conteúdo de currículo, *rich text*, simulando dados mais complexos.

Cada populador específico, como o `ClientPopulate`, herda de uma classe base (`Populators::BasePopulate`) e implementa o método `create` para instanciar e persistir um objeto através do `Factory Bot`, como mostra a Figura 85.

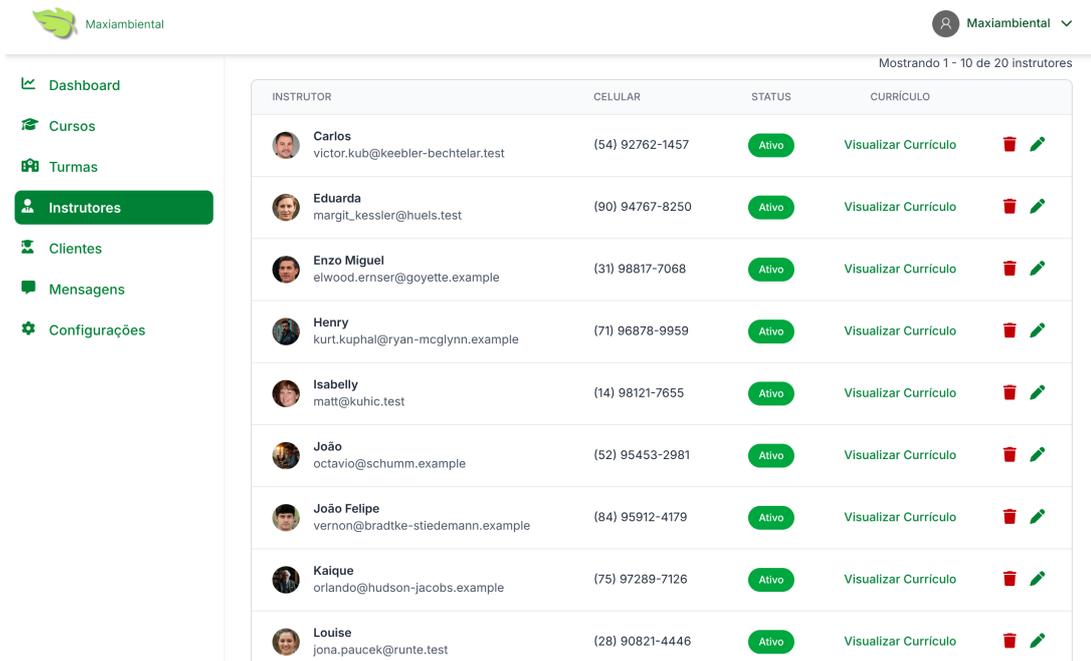
```

1 # lib/populators/client_populator.rb
2
3 require_relative 'base_populate'
4
5 class ClientPopulate < Populators::BasePopulate
6   def create
7     FactoryBot.create(:client)
8   end
9 end

```

**Figura 85 – Código para o populador do cliente.**

Com essa abordagem, tem-se um ambiente de desenvolvimento mais próximo à realidade, como mostra a Figura 86, facilitando os testes e verificações manuais.



INSTRUTOR	CELULAR	STATUS	CURRÍCULO
 <b>Carlos</b> victor.kub@keebler-bechtelar.test	(54) 92762-1457	Ativo	Visualizar Currículo
 <b>Eduarda</b> margit_kessler@huels.test	(90) 94767-8250	Ativo	Visualizar Currículo
 <b>Enzo Miguel</b> elwood.ernser@goyette.example	(31) 98817-7068	Ativo	Visualizar Currículo
 <b>Henry</b> kurt.kuphal@ryan-mcglynn.example	(71) 96878-9959	Ativo	Visualizar Currículo
 <b>Isabelly</b> matt@kuhic.test	(14) 98121-7655	Ativo	Visualizar Currículo
 <b>João</b> octavio@schumm.example	(52) 95453-2981	Ativo	Visualizar Currículo
 <b>João Felipe</b> vernon@bradtke-stiedemann.example	(84) 95912-4179	Ativo	Visualizar Currículo
 <b>Kaique</b> orlando@hudson-jacobs.example	(75) 97289-7126	Ativo	Visualizar Currículo
 <b>Louise</b> jona.paucek@runte.test	(28) 90821-4446	Ativo	Visualizar Currículo

**Figura 86 – Dados falsos criados via *populate*.**

#### 5.4.4 Favicon e Full Title

Com o objetivo de melhorar o reconhecimento da aplicação e facilitar a navegação, foi necessário adicionar um favicon com a logo da Maxiambiental e padronizar os títulos das páginas. A inclusão do favicon foi feita por meio do arquivo `favicon.ico`, inserido no diretório `app/assets/images`, e referenciado no layout principal da aplicação (`app/views/layouts/application.html.erb`) com a seguinte instrução:

```
<%= favicon_link_tag %>
```

Essa pequena adição contribui para a identificação visual da aplicação no navegador, dando destaque a logo sempre que uma aba estiver aberta.

Além disso, foi implementada uma padronização nos títulos das páginas, de forma que cada uma apresente um título coerente e informativo. Para isso, utilizou-se um helper denominado `full_title`, responsável por compor o título com base no conteúdo definido em cada view. A estrutura nas *views* é:

```
<% provide(:title, t('.title')) %>
<title><%= full_title(content_for(:title)) %></title>
```

O método `full_title` recebe um título de página e um título base (geralmente o nome da aplicação), retornando ambos combinados no formato *“Título da Página | Nome da Aplicação”*. Caso nenhum título seja fornecido pela view, apenas o nome base é exibido. A implementação do helper é apresentada a seguir:

```
def full_title(page_title = '', base_title = t('app.name'))
  page_title.blank? ? base_title : "#{page_title} | #{base_title}"
end
```

Essa abordagem garante uma navegação mais clara, com títulos em todas as abas do navegador, além de contribuir para uma experiência de uso mais organizada.

#### 5.4.5 PWA

Com o objetivo de oferecer uma experiência próxima à de aplicativos nativos, foi necessário configurar a aplicação como uma Progressive Web App (PWA). Essa adaptação permite que o sistema seja instalado diretamente na tela inicial do dispositivo e tenha uma aparência mais integrada ao sistema operacional do usuário, mesmo sendo executado em um navegador.

A configuração como PWA envolve algumas implementações-chave. A primeira delas é a criação do arquivo `manifest.json.erb`, que define metadados essenciais do aplicativo, como nome, ícones, cores e comportamento de exibição. Um exemplo simplificado desse arquivo é apresentado na Figura 87 a seguir:

```
1 {
2   "name": "<%= t('app.name') %>",
3   "short_name": "<%= t('app.short_name') %>",
4   "icons": [
5     {
6       "src": "<%= image_path('icons/logo-192x192.png') %>",
7       "sizes": "192x192"
8     },
9     {
10      "src": "<%= image_path('icons/logo-512x512.png') %>",
11      "sizes": "512x512"
12    }
13  ],
14  "start_url": "/",
15  "display": "standalone",
16  "theme_color": "#008236",
17  "background_color": "#ffffff"
18 }
```

**Figura 87 – Configurações PWA.**

Esse arquivo é incluído no layout principal da aplicação por meio de uma tag `link`, garantindo que os navegadores reconheçam a aplicação como instalável e ofereçam ao usuário a opção de adicioná-la à tela inicial.

Além disso, foram configuradas rotas específicas para disponibilizar o *manifest* e o *service worker*, como mostrado abaixo:

```
get "manifest" => "rails/pwa#manifest", as: :pwa_manifest
get "service-worker" => "rails/pwa#service_worker"
```

Essa configuração garante que os navegadores possam acessar corretamente os arquivos necessários para ativar os recursos do PWA, como a instalação.

#### 5.4.6 Open Graph

Para customizar a forma como o conteúdo da aplicação é exibido ao ser compartilhado em redes sociais, foi configurado o Open Graph (OG). O Open Graph melhora a experiência de compartilhamento, pois sem as tags, ao compartilhar um link, a rede social pode ter dificuldade em extrair as informações corretas ou exibir apenas um preview. Com o Open Graph, a aplicação controla como seu conteúdo é representado, garantindo que o título, a descrição e a imagem sejam previamente definidos.

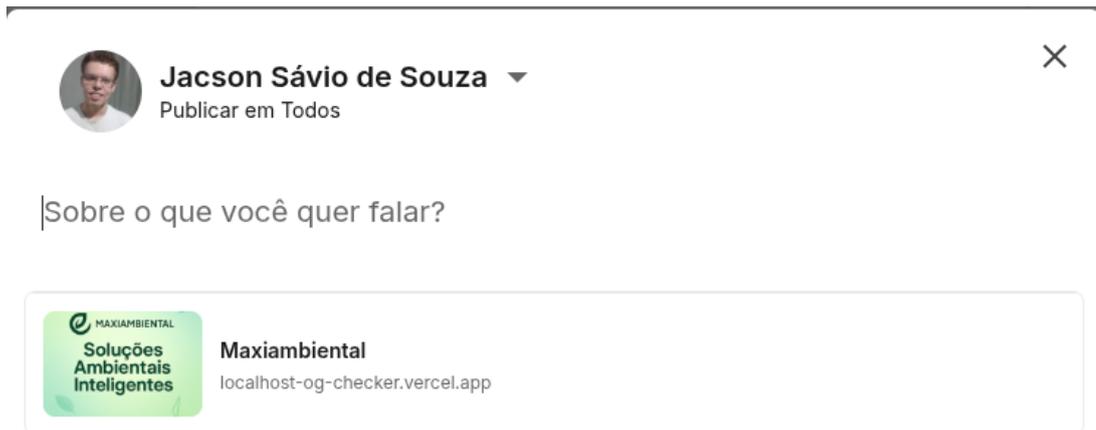
As meta tags implementadas abrangem os principais atributos do protocolo Open Graph, além de meta tags básicas para o título e descrição da página e tags específicas para o Twitter Card, que oferece uma experiência similar de compartilhamento no Twitter. O código da Figura 88 abaixo ilustra a configuração das meta tags do Open Graph:

```
1 <meta name="title" content="<%= t('app.name') %>">
2 <meta name="description" content="<%= t('app.description') %>">
3
4 <meta property="og:title" content="<%= t('app.name') %>">
5 <meta property="og:description" content="<%= t('app.description') %>">
6 <meta property="og:image" content="<%= image_url('maxiambiental_og.png') %>">
7 <meta property="og:url" content="<%= root_url %>">
8 <meta property="og:type" content="website">
9
10 <meta name="twitter:card" content="<%= image_url('maxiambiental_og.png') %>">
11 <meta name="twitter:title" content="<%= t('app.name') %>">
12 <meta name="twitter:description" content="<%= t('app.description') %>">
13 <meta name="twitter:image" content="<%= image_url('maxiambiental_og.png') %>">
```

**Figura 88 – Tags Open Graph.**

Neste contexto, `t('app.name')` e `t('app.description')` recuperam o nome e a descrição da aplicação. As funções `image_url` e `root_url` geram os caminhos absolutos para a imagem de pré-visualização (`maxiambiental_og.png`) e para a URL base da aplicação, respectivamente, conforme exigido pelo protocolo Open Graph para garantir que os recursos sejam acessíveis externamente.

A Figura 89 demonstra um exemplo de como a URL da aplicação é exibida quando compartilhada em uma plataforma social após a configuração do Open Graph.



**Figura 89 – Exemplo de visualização de link compartilhado com Open Graph.**

## 5.5 Realização dos Testes

Durante o desenvolvimento da aplicação, foi necessário testar cada funcionalidade implementada. Em cada *feature*, foram implementados 4 tipos principais de testes: unitários, de componentes (*ViewComponent*), de controladores e de sistema. Para essa abordagem utilizou-se o Minitest<sup>24</sup>, framework de testes padrão do Rails, e o Capybara<sup>25</sup> para testes de sistema.

### 5.5.1 Testes Unitários

Os testes unitários foram focados na validação de unidades isoladas do código, como modelos e lógicas de negócio específicas. O objetivo era que cada parte menor do sistema se comportasse exatamente como esperado, independentemente de outras. Isso incluiu a verificação de validações e métodos personalizados nos modelos e serviços.

A *gem* Shoulda Matchers<sup>26</sup> foi muito utilizada nos testes de validações. Ela oferece uma sintaxe concisa e legível para testar funcionalidades comuns do ActiveRecord, como validações de presença, unicidade e comprimento, reduzindo a verbosidade do código de teste.

Um exemplo de teste unitário para o modelo Client ilustra essa abordagem na Figura 90 abaixo:

<sup>24</sup> Minitest é uma estrutura de teste rápida e enxuta, padrão no Ruby e Rails. [<https://guides.rubyonrails.org/testing.html>]

<sup>25</sup> Capybara é uma ferramenta de automação para testes de aceitação em aplicações web, simulando a interação do usuário. [<https://github.com/teamcapybara/capybara>]

<sup>26</sup> Shoulda Matchers é uma coleção de matchers RSpec-inspired que podem ser usados com o Minitest ou RSpec para simplificar os testes de modelos e controladores no Rails. [<https://thoughtbot.com/open-source/shoulda-matchers>]

```

1 require 'test_helper'
2
3 class ClientTest < ActiveSupport::TestCase
4   should validate_presence_of(:email)
5   should validate_presence_of(:name)
6   should validate_presence_of(:password).on(:create)
7
8   should validate_uniqueness_of(:email).case_insensitive
9   should validate_length_of(:password).is_at_least(6)
10
11   test 'active attribute defaults to true' do
12     assert FactoryBot.build(:client).active
13   end
14 end

```

**Figura 90 – Teste do modelo *Client*.**

Esse teste valida as principais regras de negócio do modelo *Client*, garantindo que um cliente não possa ser criado sem nome ou e-mail, que o e-mail seja único e que a senha atenda aos requisitos mínimos.

Com a utilização de View Components na construção da interface do usuário, foi necessário testá-los de forma isolada. Assim, os testes de componentes focaram em verificar se cada ViewComponent renderizava o HTML correto e se a lógica de apresentação encapsulada funcionava conforme o esperado, sem depender do restante da aplicação.

O exemplo da Figura 91 abaixo demonstra um teste para o *Search::FormComponent*, verificando se o formulário gerado possui a ação e o método corretos:

```

1 # frozen_string_literal: true
2
3 require 'test_helper'
4
5 class Search::FormComponentTest < ViewComponent::TestCase
6   BASE_URL = '/admin/courses/search/test'
7
8   def test_renders_form_with_correct_action_and_method
9     render_inline(Search::FormComponent.new(base_url: BASE_URL))
10
11     assert_selector "form[action='#{BASE_URL}'][method='get']"
12   end
13 end

```

**Figura 91 – Teste do componente de busca.**

### 5.5.2 Testes de controladores

Os testes de controladores foram empregados para verificar a lógica dos controladores, as requisições e as respostas. Eles verificam se as ações do controlador respondem com o sta-

tus HTTP correto, que redirecionamentos ocorrem como esperado, que as variáveis de instância são atribuídas corretamente e que as mensagens flash são definidas apropriadamente.

```
1 require 'test_helper'
2
3 class Admin::InstructorsControllerTest < ActionDispatch::IntegrationTest
4   setup do
5     @admin = FactoryBot.create(:admin)
6
7     sign_in @admin
8   end
9
10  test 'should get new' do
11    get new_admin_instructor_path
12
13    assert_response :success
14  end
15
16  test 'should update an instructor' do
17    instructor = FactoryBot.create(:instructor)
18
19    patch admin_instructor_path(instructor),
20          params: { instructor: { name: 'Instructor 2', phone: '(42) 98425-2615' } }
21
22    assert_redirected_to admin_instructors_path
23
24    assert_equal I18n.t('flash_messages.updated', model: Instructor.model_name.human), flash[:notice]
25  end
26 end
```

**Figura 92 – Teste do controlador.**

### 5.5.3 Testes de Sistema

Por fim, os testes de sistema foram uma camada abrangente, simulando a interação do usuário com o navegador web. Utilizando o Capybara, esses testes navegam pela aplicação, clicam em botões, preenchem formulários e verificam o conteúdo da página, replicando fluxos de uso reais. Essa camada é importante para identificar problemas de integração entre diferentes partes da aplicação.

```

1 require 'application_system_test_case'
2
3 class Admin::CoursesTest < ApplicationSystemTestCase
4   setup do
5     @admin = create(:admin)
6     @course = create(:course)
7     login_as(@admin, scope: :admin)
8   end
9
10  test 'viewing courses index' do
11    visit admin_courses_path
12
13    assert_selector 'h1', text: I18n.t('admin.courses.index.title')
14    assert_selector 'table tbody tr', count: Course.count
15    assert_selector 'table tbody tr td', text: @course.name
16  end
17
18  test 'creating a course' do
19    visit new_admin_course_path
20
21    fill_in attribute(:name), with: 'Ambiental Web'
22    find('trix-editor').set('An awesome course about web development.')
23
24    click_on I18n.t('helpers.submit.create', model: model_name)
25
26    assert_current_path admin_courses_path
27    assert_alert I18n.t('flash.messages.created', model: model_name)
28    assert_text 'Ambiental Web'
29  end
30 end

```

Figura 93 – Teste de sistema.

## 5.6 Principais Problemas

O desenvolvimento deste projeto enfrentou alguns problemas na aplicação completa das metodologias ágeis planejadas inicialmente. Embora não houvesse complicações relacionadas ao desenvolvimento em si, a coleta de métricas de produtividade e evolução do projeto foi comprometida devido a restrições temporais durante a execução.

A natureza do sistema, combinada com o escopo abrangente definido nos objetivos, demandou um esforço de desenvolvimento concentrado nas funcionalidades essenciais relacionadas à área do administrador. Essa priorização, embora necessária para garantir a entrega do produto funcional, resultou na alocação limitada de tempo para:

- Documentação detalhada de métricas de produtividade;
- Implementação completa de todos os recursos de monitoramento planejados;
- Análise quantitativa aprofundada do processo de desenvolvimento.

Vale ressaltar que, apesar dessas limitações, a qualidade técnica foi mantida através de:

- Implementação de testes automatizados;
- Revisões contínuas do código;
- Validações periódicas com os requisitos originais;

O descontinuação do Pivotal Tracker durante o projeto representou um desafio adicional, impossibilitando a recuperação retrospectiva de alguns dados de acompanhamento. Esta experiência destacou a importância de estabelecer processos mais robustos para a preservação de métricas em trabalhos futuros.

Para projetos subsequentes, recomenda-se:

- Alocação específica de tempo para coleta de métricas;
- Uso de ferramentas consolidadas no mercado, com garantia de suporte continuado;

Estas medidas permitiriam uma análise mais quantitativa do processo de desenvolvimento sem comprometer o foco na implementação das funcionalidades principais.

## 6 CONCLUSÃO

Este trabalho propôs o desenvolvimento de uma nova versão do sistema web da Maxiambiental Treinamentos, visando atualizar um sistema com tecnologias defasadas e atender às demandas atuais da oferta de cursos. Com o escopo reduzido à área do administrador e a algumas partes da área dos clientes, os resultados obtidos foram satisfatórios. O novo sistema foi implementado com tecnologias atualizadas - Ruby on Rails, PostgreSQL, Docker, incorporou uma interface responsiva, e estruturou áreas distintas para visitantes, alunos e administradores. A análise de requisitos, baseada no sistema legado e no diálogo com os responsáveis pela empresa, direcionou funcionalidades essenciais, como a inscrição em turmas, o gerenciamento administrativo de cursos, turmas, instrutores e clientes e a análise de dados a partir de um *dashboard*.

Um ponto observado foi a limitação temporal para a coleta de métricas de produtividade durante o desenvolvimento. Embora metodologias ágeis e BDD tenham orientado o processo, a descontinuidade do *Pivotal Tracker* impediu uma análise quantitativa detalhada da evolução do projeto. Adicionalmente, a integração com *gateways* de pagamento reais não foi encaixada no escopo definido, permanecendo como uma simulação funcional.

Apesar dos desafios, o desenvolvimento da solução proporcionou um valioso aprendizado. A adoção de práticas como Integração Contínua (CI/CD) e a implementação de testes automatizados permitem uma manutenção mais simples do sistema. Além disso, o projeto estabelece uma base sólida para futuras expansões.

### 6.1 Trabalhos futuros

A continuidade do desenvolvimento do sistema web da Maxiambiental Treinamentos envolve a implementação de funcionalidades adicionais e a otimização dos recursos existentes.

Uma etapa importante consiste na integração com sistemas de pagamento. Esta ação substituirá a simulação atual por conexões com APIs de provedores como PagSeguro ou Stripe. Tal implementação permitirá transações reais e automatizará as confirmações de inscrição.

A expansão do suporte a dispositivos móveis constitui um próximo passo. Os recursos Progressive Web App (PWA) serão aperfeiçoados para incluir funcionamento *offline* e notificações *push*. Esta funcionalidade visa aumentar o engajamento dos usuários em cenários com conectividade limitada.

É relevante ressaltar que, devido ao escopo e cronograma deste trabalho, diversas histórias de usuário, criadas durante a fase de levantamento de requisitos, não foram implementadas. Tais histórias representam funcionalidades adicionais que podem ser consideradas em trabalhos futuros para expandir o sistema e atender a um conjunto mais amplo de necessidades da Maxiambiental Treinamentos.

## REFERÊNCIAS

- AWARI. **As Vantagens E Desvantagens Do Figma: Tudo O Que Você Precisa Saber Sobre Essa Ferramenta De Design**. 2023. Site Awari. Disponível em: <https://awari.com.br/as-vantagens-e-desvantagens-do-figma-tudo-o-que-voce-precisa-saber-sobre-essa-ferramenta-de-design>. Acesso em: 10 jun. 2024.
- BRAKEMAN. **Brakeman: Static analysis security scanner for Ruby on Rails**. 2025. GitHub. Scanner de segurança estática para aplicações Rails. Disponível em: <https://github.com/presidentbeef/brakeman>. Acesso em: 18 mar. 2025.
- COURSERA. **Dê o primeiro passo em direção à sua nova carreira**. 2024. Coursera. Disponível em: <https://www.coursera.org/>. Acesso em: 18 mar. 2024.
- DOCKER. **Docker Builds: Now Lightning Fast**. 2024. Docker. Disponível em: <https://www.docker.com/>. Acesso em: 18 mar. 2024.
- FIGMA. **How you design , align , and build matters. Do it together with Figma**. 2025. Figma. Disponível em: <https://www.figma.com/>. Acesso em: 18 mar. 2025.
- FOX, A.; JOSEPH, D. **Construindo Software como Serviço: Uma abordagem Ágil usando computação em nuvem**. Bookman Editora, 2017. ISBN 0984881271. Disponível em: <https://www.amazon.com.br/Construindo-Software-como-Servi%C3%A7o-SaaS/dp/0984881271>. Acesso em: 10 jun. 2024.
- GITHUB. **GitHub: Let's build from here**. 2024. Github. Disponível em: <https://github.com/>. Acesso em: 18 mar. 2024.
- GOV. **Biodiversidade**. 2024. Ministério do Meio Ambiente e Mudança do Clima. Disponível em: <https://www.gov.br/mma/pt-br/assuntos/biodiversidade-e-ecossistemas>. Acesso em: 18 mar. 2024.
- GOV. **Crescimento da economia brasileira é impulsionado pela alta de 15% da agropecuária em 2023**. 2024. Mesquita, Ministério da Agricultura e Pecuária. Disponível em: <https://www.gov.br/agricultura/pt-br/assuntos/noticias/crescimento-da-economia-brasileira-e-impulsionado-pela-alta-de-15-da-agropecuaria-em-2023>. Acesso em: 19 mar. 2024.
- JETBRAINS. **O que é CI/CD em DevOps?** 2023. Site Jetbrains. Disponível em: <https://www.jetbrains.com/pt-br/teamcity/ci-cd-guide/devops-ci-cd/>. Acesso em: 18 jun. 2024.
- JETBRAINS. **O que é Integração Contínua (CI)?** 2023. Site Jetbrains. Disponível em: <https://www.jetbrains.com/pt-br/teamcity/ci-cd-guide/continuous-integration/>. Acesso em: 18 jun. 2024.
- KAMAL. **Kamal: Deploy web apps anywhere**. 2025. GitHub. Ferramenta de deploy para aplicações Rails (anteriormente conhecida como MRSK). Disponível em: <https://github.com/basecamp/kamal>. Acesso em: 18 mar. 2025.
- MARTINEZ, P. H. **O Brasil e o meio ambiente: nossa trajetória, nossos desafios**. 2022. Jornal UNESP. Disponível em: <https://jornal.unesp.br/2022/05/05/o-brasil-e-o-meio-ambiente-nossa-trajetoria-nossos-desafios/>. Acesso em: 18 mar. 2024.

MAXIAMBIENTAL. **Institucional**. 2019. Maxiambiental, Institucional. Disponível em: [https://maxiambiental.com/about\\_maxiambiental](https://maxiambiental.com/about_maxiambiental). Acesso em: 18 mar. 2020.

MEDIUM. **UX/UI Design: A Importância da Retenção de Usuários com uma Interface de Usuário Intuitiva e Agradável**. 2023. Medium. Disponível em: <https://medium.com/@rf.sants001/ux-ui-design-a-import%C3%A2ncia-da-reten%C3%A7%C3%A3o-de-usu%C3%A1rios-com-uma-interface-de-usu%C3%A1rio-intuitiva-e-4fce39524a08>. Acesso em: 18 mar. 2024.

PIVOTAL. **Pivotal Tracker: Essentials**. 2024. Site Pivotal Tracker. Disponível em: <https://www.pivotaltracker.com/>. Acesso em: 10 jun. 2024.

POSTGRESQL. **PostgreSQL: The World's Most Advanced Open Source Relational Database**. 2024. PostgreSQL. Disponível em: <https://www.postgresql.org/>. Acesso em: 18 mar. 2024.

RAILS, R. on. **Ruby on Rails: Compress the complexity of modern web apps**. 2024. RubyOnRails. Disponível em: <https://rubyonrails.org/>. Acesso em: 18 mar. 2024.

RUBOCOP. **RuboCop: A Ruby static code analyzer and formatter**. 2024. GitHub. Ferramenta de análise estática e formatação de código Ruby. Disponível em: <https://github.com/rubocop/rubocop>. Acesso em: 18 mar. 2024.

SILVA, N. R. L. **Entendendo os benefícios do CI/CD**. 2023. <https://medium.com/@nataliaregina.lsilva/entendendo-os-benef%C3%ADcios-do-ci-cd-e76f3ba44d85>. Disponível em: <https://medium.com/@nataliaregina.lsilva/entendendo-os-benef%C3%ADcios-do-ci-cd-e76f3ba44d85>. Acesso em: 2024-06-18.

TAILWIND. **Tailwind CSS: Rapidly build modern websites without ever leaving your HTML**. 2025. Tailwind Labs. Framework CSS utilitário para desenvolvimento web moderno. Disponível em: <https://tailwindcss.com>. Acesso em: 18 mar. 2025.

UDEMY. **Aprendendo o que quiser, quando quiser**. 2024. Udemy. Disponível em: <https://www.udemy.com/pt/>. Acesso em: 18 mar. 2024.

ZUP. **Git Workflow: o que é e principais tipos**. 2021. Site zup. Disponível em: <https://www.zup.com.br/blog/git-workflow>. Acesso em: 10 jun. 2024.

## A HISTÓRIAS DE USUÁRIO COMPLEMENTARES

Neste apêndice, são detalhadas as histórias de usuário que representam requisitos de suporte e funcionalidades complementares do sistema, agrupadas por sua área de atuação.

### A.1 Área pública (Complementar)

#### Feature: Seção de Perguntas Frequentes

**Como** um usuário visitante, **quero** visualizar uma seção de perguntas frequentes **para** encontrar respostas rápidas e úteis para dúvidas comuns. **Dado** que sou um usuário visitante **Quando** eu acessar a seção de perguntas frequentes **Então** eu devo encontrar respostas rápidas e úteis para dúvidas comuns sobre os cursos, a empresa e o processo de inscrição.

#### Feature: Formulário de Contato

**Como** um usuário visitante, **quero** preencher um formulário de contato para enviar uma mensagem para o email da empresa **para** que eu possa fazer perguntas ou solicitar informações adicionais que não encontrei no site. **Dado** que sou um usuário visitante **Quando** eu acessar o formulário de contato **Então** eu devo poder preencher meus dados e enviar uma mensagem para o email da empresa.

#### Feature: Cadastro para Notificações

**Como** um usuário visitante, **quero** ter a opção de me cadastrar para receber notificações da empresa sobre cursos novos, vagas disponíveis e outras atualizações. **Dado** que sou um usuário visitante **Quando** eu preencher o formulário de cadastro para notificações **Então** eu devo receber atualizações da empresa sobre cursos novos, vagas disponíveis e outras notícias relevantes.

#### Feature: Galeria de Fotos por Álbum de Turma

**Como** um usuário visitante, **quero** visualizar a galeria de fotos agrupadas por álbum de turma **para** ver imagens das atividades e eventos da empresa. **Dado** que sou um usuário visitante **Quando** eu acessar a galeria de fotos **Então** eu devo ver as fotos agrupadas por álbum de turma.

#### Feature: Visualização de Álbum de Turma Específico

**Como** um usuário visitante, **quero** visualizar um álbum de uma turma específica **para** ver fotos relacionadas a essa turma. **Dado** que sou um usuário visitante **Quando** eu clicar em um álbum específico **Então** eu devo ver as fotos relacionadas àquela turma.

#### Feature: Compartilhamento de fotos

**Como** um usuário visitante, **quero** poder compartilhar uma foto no Instagram, Facebook e Twitter **para** mostrar aos meus amigos e seguidores e divulgar as atividades da turma. **Dado** que sou um usuário visitante **Quando** eu clicar em uma foto específica **Então** eu devo ter a opção de compartilhá-la no Instagram, Facebook e Twitter.

#### Feature: Informações Institucionais

**Como** um usuário visitante, **quero** visualizar as informações institucionais da empresa **para** entender melhor sua missão, visão e valores. **Dado** que sou um usuário visitante **Quando** eu acessar a página de informações institucionais **Então** eu devo visualizar informações sobre a missão, visão e valores da empresa.

#### Feature: Acesso ao Formulário de Inscrição

**Como** um usuário visitante, **quero** acessar o formulário de inscrição **para** me inscrever em um curso oferecido pela empresa. **Dado** que sou um usuário visitante **Quando** eu clicar em "Inscrever-se" em um curso **Então** eu devo ser direcionado para o formulário de inscrição do curso.

## A.2 Área do usuário (Complementar)

### Feature: Histórico de Cursos Realizados

**Como** um usuário autenticado, **quero** acessar uma lista dos cursos que já realizei, **para** que eu possa acompanhar meu histórico de cursos. **Dado** que sou um usuário autenticado **Quando** eu acessar a área do aluno **Então** eu devo visualizar uma lista dos cursos que já realizei, com informações sobre cada curso e meu progresso.

### Feature: Visualização e Download de Certificados

**Como** um usuário autenticado, **quero** visualizar os certificados que recebi, **para** que eu possa comprovar minha participação e conclusão dos cursos. **Dado** que sou um usuário autenticado **Quando** eu acessar a área do aluno ou a página de um curso concluído **Então** eu devo poder visualizar e baixar os certificados dos cursos que concluí.

### Feature: Visualização de Contatos Públicos

**Como** um usuário autenticado, **quero** visualizar os contatos públicos de outros usuários, **para** que eu possa me conectar e interagir com outros profissionais. **Dado** que sou um usuário autenticado **Quando** eu acessar o perfil de outro usuário **Então** eu devo poder visualizar seus contatos públicos, como email e LinkedIn, se ele tiver optado por compartilhá-los.

### Feature: Busca de Profissionais

**Como** um usuário autenticado, **quero** buscar profissionais cadastrados no sistema, **para** que eu possa encontrar e me conectar com profissionais relevantes para mim. **Dado** que sou um usuário autenticado **Quando** eu acessar a página de busca de profissionais **Então** eu devo poder buscar outros profissionais cadastrados no sistema por nome, área de atuação ou empresa.

### A.3 Área do administrador (Complementar)

#### Feature: Gerenciamento da Galeria de Fotos (Admin)

**Como** administrador, **quero** criar álbuns de fotos, adicionar imagens, definir títulos e escolher imagens de capa, **para** que eu possa organizar e exibir fotos relacionadas aos cursos e turmas. **Dado** que sou um administrador logado no sistema **Quando** eu acessar a página de gerenciamento da galeria de fotos **Então** eu devo ver uma lista de todos os álbuns de fotos, com opções para criar novos álbuns, adicionar imagens, definir títulos e escolher imagens de capa. **E** ao criar um novo álbum, devo ser capaz de associá-lo a uma turma ou deixá-lo sem associação.

#### Feature: Gerenciamento de Mensagens (Admin)

**Como** administrador, **quero** visualizar, filtrar e marcar mensagens como lidas ou não lidas, **para** que eu possa responder e gerenciar a comunicação recebida pelo site. **Dado** que sou um administrador logado no sistema **Quando** eu acessar a página de gerenciamento de mensagens **Então** eu devo ver uma lista de todas as mensagens recebidas pelo site, com opções para filtrar e marcar como lidas ou não lidas.

#### Feature: Gerenciamento Institucional (Admin)

**Como** administrador, **quero** editar as informações institucionais da empresa, **para** que eu possa manter as informações da empresa atualizadas para os usuários. **Dado** que sou um administrador logado no sistema **Quando** eu acessar a página de gerenciamento institucional da empresa **Então** eu devo ver todas as seções institucionais disponíveis para edição.

## B PROTOTIPAÇÕES ADICIONAIS DE TELAS

Esta seção apresenta prototipações adicionais de telas que complementam as interfaces de usuário descritas no corpo principal do documento.

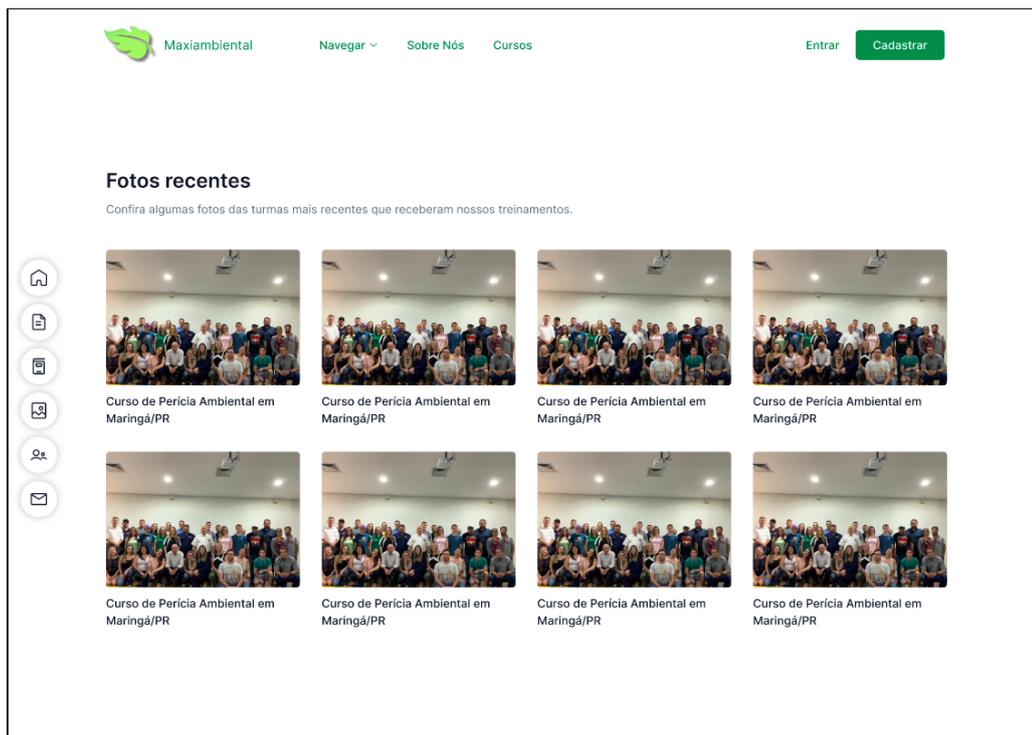
### B.1 Área pública

A Figura 94 destaca o alcance da Maxiambiental, apresentando suas conquistas e resultados de forma visual na página inicial.



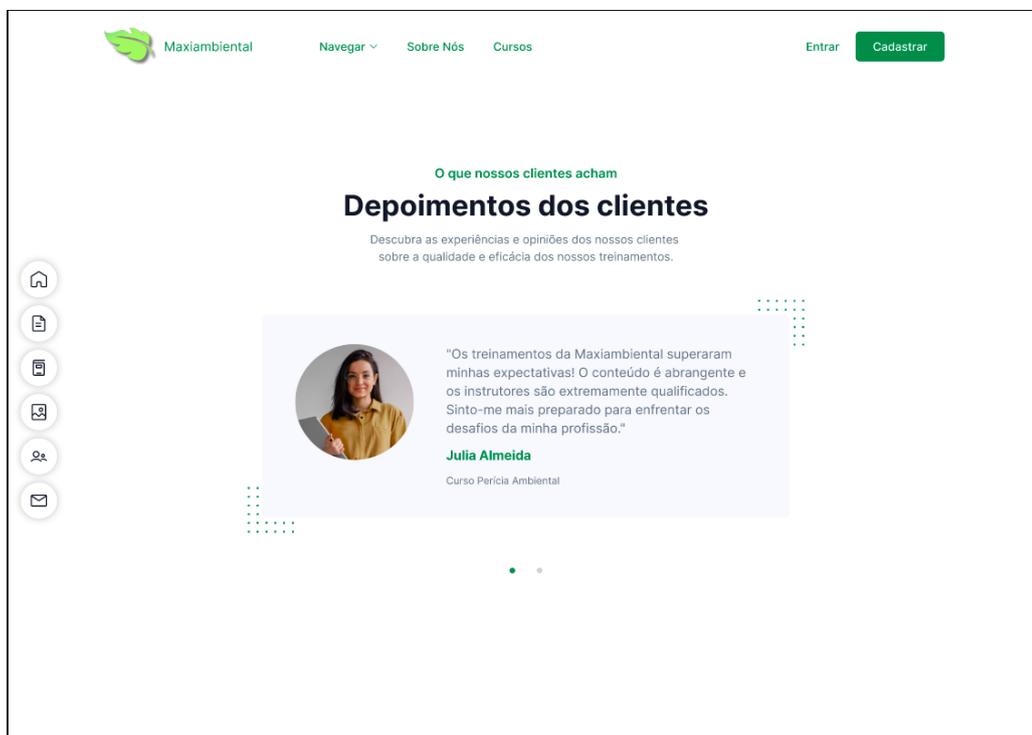
**Figura 94 – Alcance Maxiambiental - Início**

A Figura 95 exibe exemplos de fotos de turmas na página inicial, proporcionando uma visão do ambiente de aprendizado e da comunidade Maxiambiental.



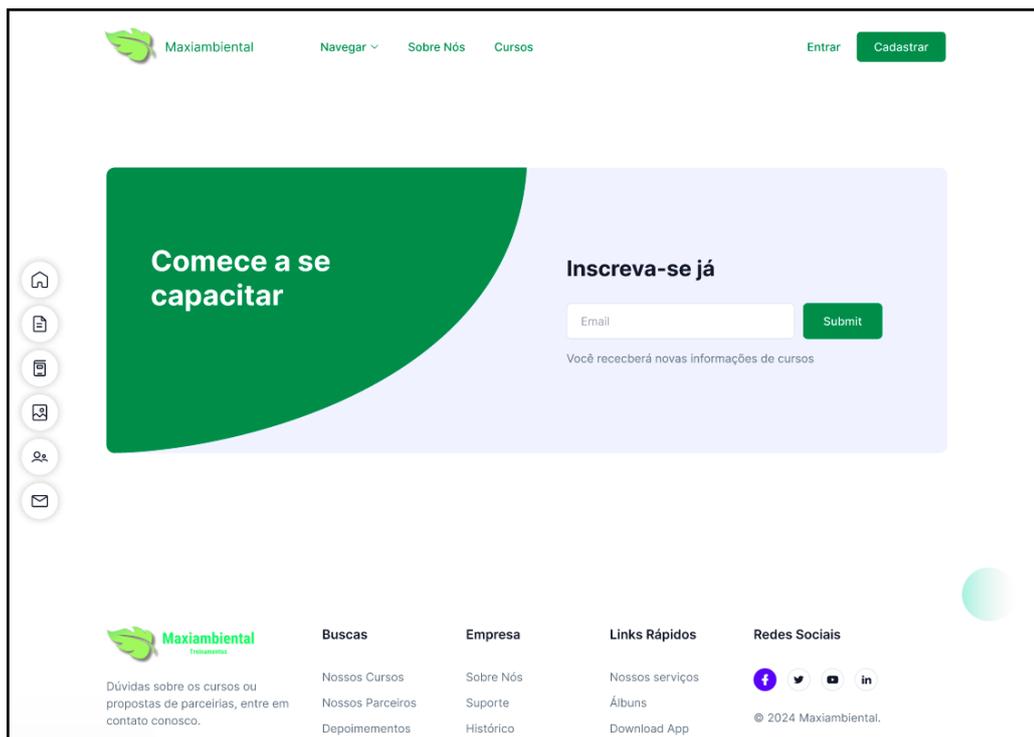
**Figura 95 – Fotos de turmas - Início**

A Figura 96 apresenta depoimentos de ex-alunos na página inicial, com o objetivo de fortalecer a credibilidade da Maxiambiental.



**Figura 96 – Depoimentos - Início**

A Figura 97 mostra a seção de cadastro de e-mail, incentivando os visitantes a se inscreverem para receber novidades e informações, Feature: Cadastro para Notificações.



**Figura 97 – Cadastro de Email - Início**

A Figura 98 apresenta as perguntas frequentes, onde os usuários podem encontrar respostas para suas dúvidas mais comuns, Feature: Seção de Perguntas Frequentes.

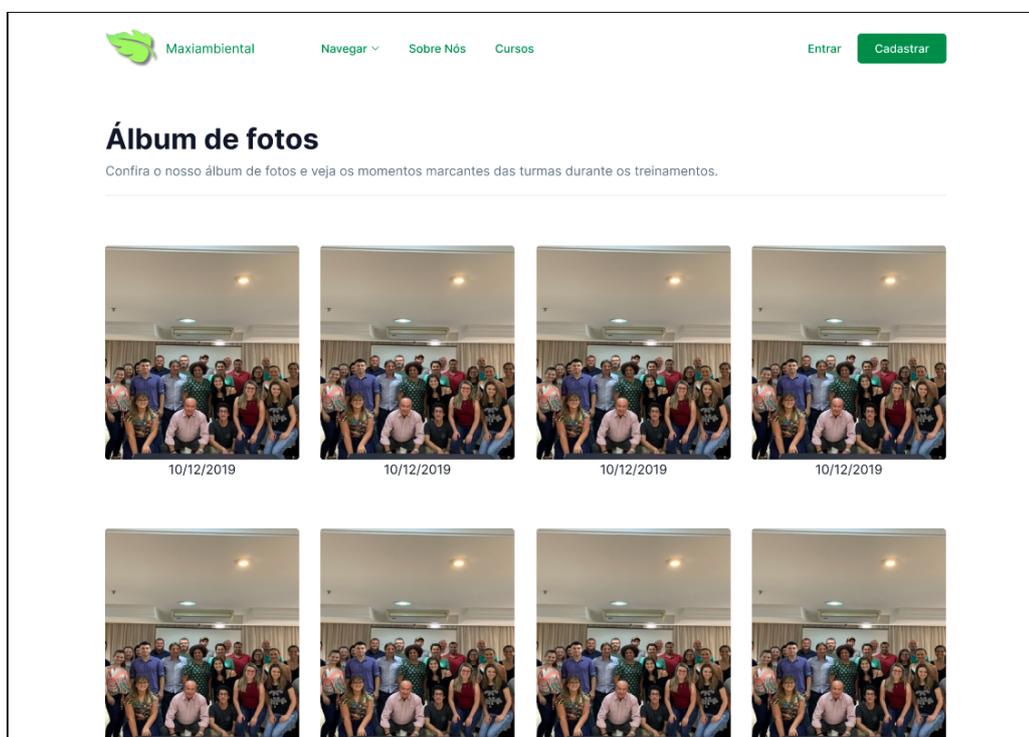


**Figura 98 – Perguntas frequentes**

A Figura 99 exibe a página de contato, onde os usuários podem encontrar informações de contato da empresa e um formulário para enviar mensagens, Feature: Formulário de Contato.

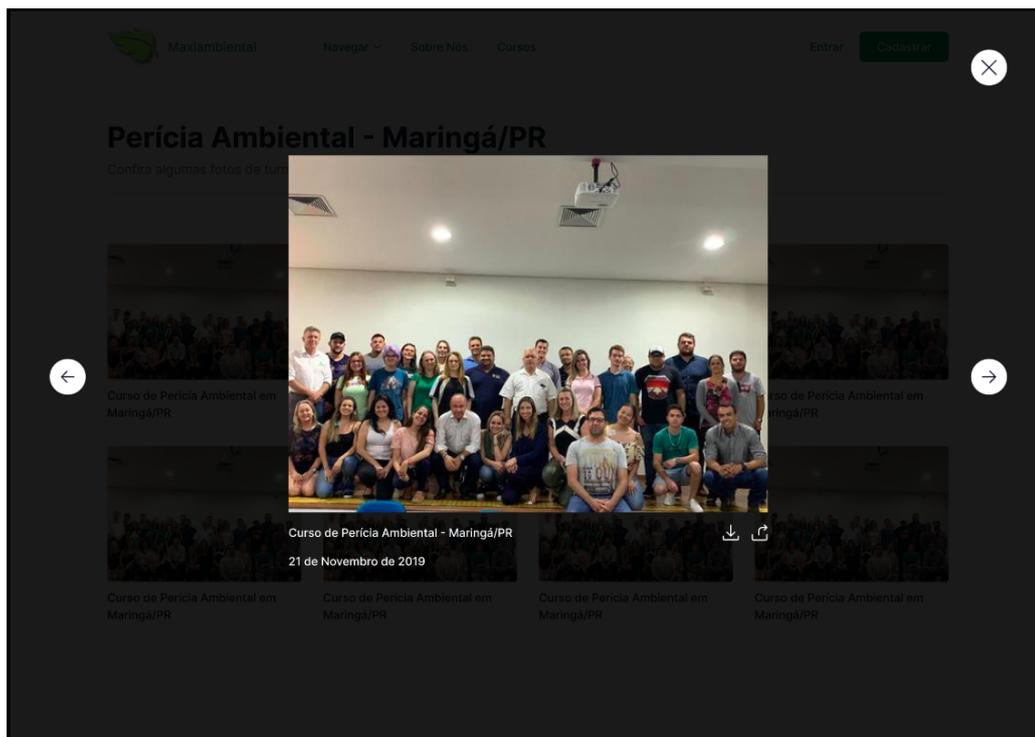
**Figura 99 – Contato Maxiambiental**

A Figura 100 mostra os álbuns dos cursos, com fotos dos alunos e professores interagindo nos cursos do sistema web, Feature: Galeria de Fotos por Álbum de Turma.



**Figura 100 – Álbuns de fotos Maxiambiental**

A Figura 101 ilustra a funcionalidade de compartilhamento de fotos, permitindo a divulgação das atividades nas redes sociais, Feature: Compartilhamento de fotos.



**Figura 101 – Compartilhamento de Fotos**

A Figura 102 ilustra sobre os valores da empresa e destaca os marcos alcançados durante as atividades, Feature: Informações Institucionais.

**Maxiambiental** Navegar Sobre Nós Cursos Entrar Cadastrar

**Sobre nós**

## Excelência em treinamentos ambientais.

Através da realização de treinamentos de curta duração o objetivo da empresa é proporcionar o aperfeiçoamento do conhecimento técnico-prático e científico de profissionais e estudantes que atuam ou pretendem atuar na área ambiental.

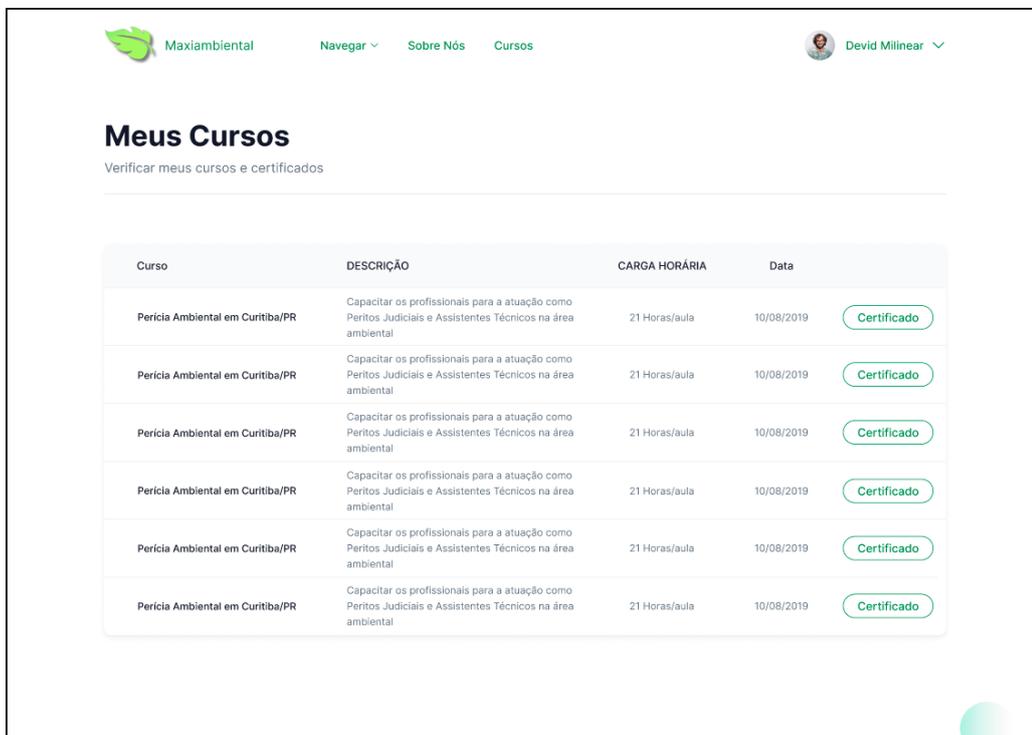
Os treinamentos são ministrados por profissionais altamente especializados e qualificados que possuem conhecimentos práticos e aplicados. Desta forma, os cursos são realizados de modo a proporcionar aos participantes o conhecimento que estes precisam para atuar no mercado, podendo assim ampliar as oportunidades de trabalho do profissional.

<b>+15</b>	<b>+150</b>	<b>+8K</b>
Estados brasileiros já receberam treinamentos	Turmas de cursos na área ambiental já foram capacitadas	Profissionais capacitados na modalidade presencial

**Figura 102 – Institucional Maxiambiental - Início**

### B.1.1 Área do usuário

A Figura 103 apresenta a área de histórico de cursos, onde os usuários podem acompanhar os cursos realizados, Feature: Histórico de Cursos Realizados.

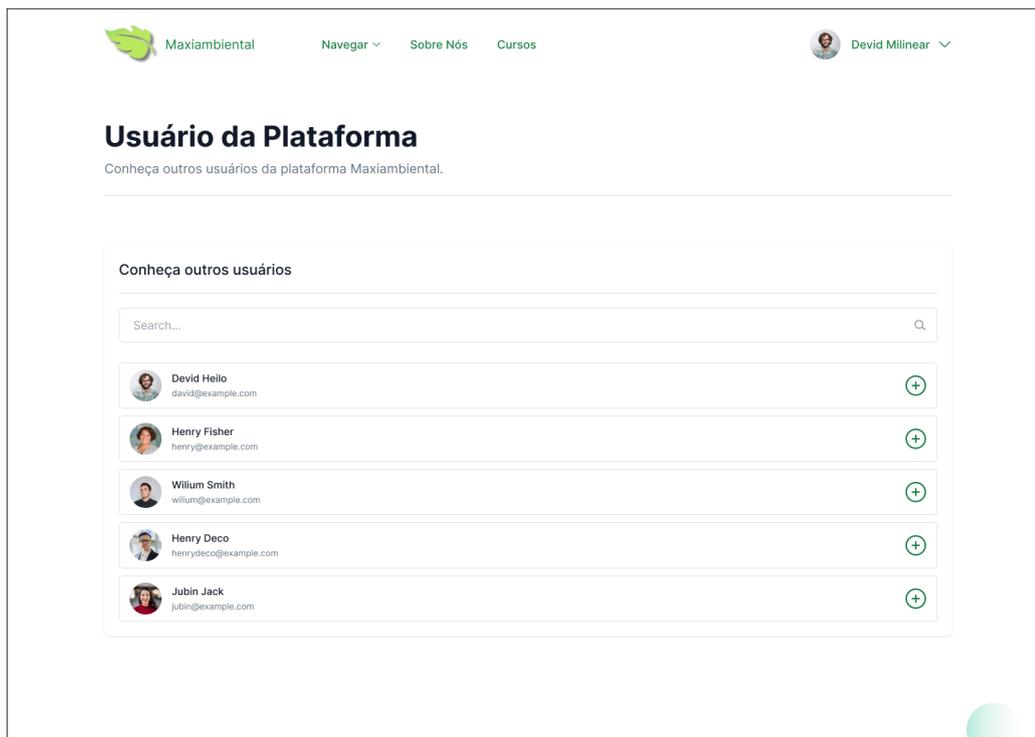


The screenshot displays the 'Meus Cursos' (My Courses) section of the Maxiambiental user interface. The page header includes the Maxiambiental logo, navigation links (Navegar, Sobre Nós, Cursos), and the user's name (Devid Milinear). Below the header, the title 'Meus Cursos' is followed by the subtitle 'Verificar meus cursos e certificados'. The main content is a table with the following data:

Curso	DESCRIÇÃO	CARGA HORÁRIA	Data	
Perícia Ambiental em Curitiba/PR	Capacitar os profissionais para a atuação como Peritos Judiciais e Assistentes Técnicos na área ambiental	21 Horas/aula	10/08/2019	<a href="#">Certificado</a>
Perícia Ambiental em Curitiba/PR	Capacitar os profissionais para a atuação como Peritos Judiciais e Assistentes Técnicos na área ambiental	21 Horas/aula	10/08/2019	<a href="#">Certificado</a>
Perícia Ambiental em Curitiba/PR	Capacitar os profissionais para a atuação como Peritos Judiciais e Assistentes Técnicos na área ambiental	21 Horas/aula	10/08/2019	<a href="#">Certificado</a>
Perícia Ambiental em Curitiba/PR	Capacitar os profissionais para a atuação como Peritos Judiciais e Assistentes Técnicos na área ambiental	21 Horas/aula	10/08/2019	<a href="#">Certificado</a>
Perícia Ambiental em Curitiba/PR	Capacitar os profissionais para a atuação como Peritos Judiciais e Assistentes Técnicos na área ambiental	21 Horas/aula	10/08/2019	<a href="#">Certificado</a>
Perícia Ambiental em Curitiba/PR	Capacitar os profissionais para a atuação como Peritos Judiciais e Assistentes Técnicos na área ambiental	21 Horas/aula	10/08/2019	<a href="#">Certificado</a>

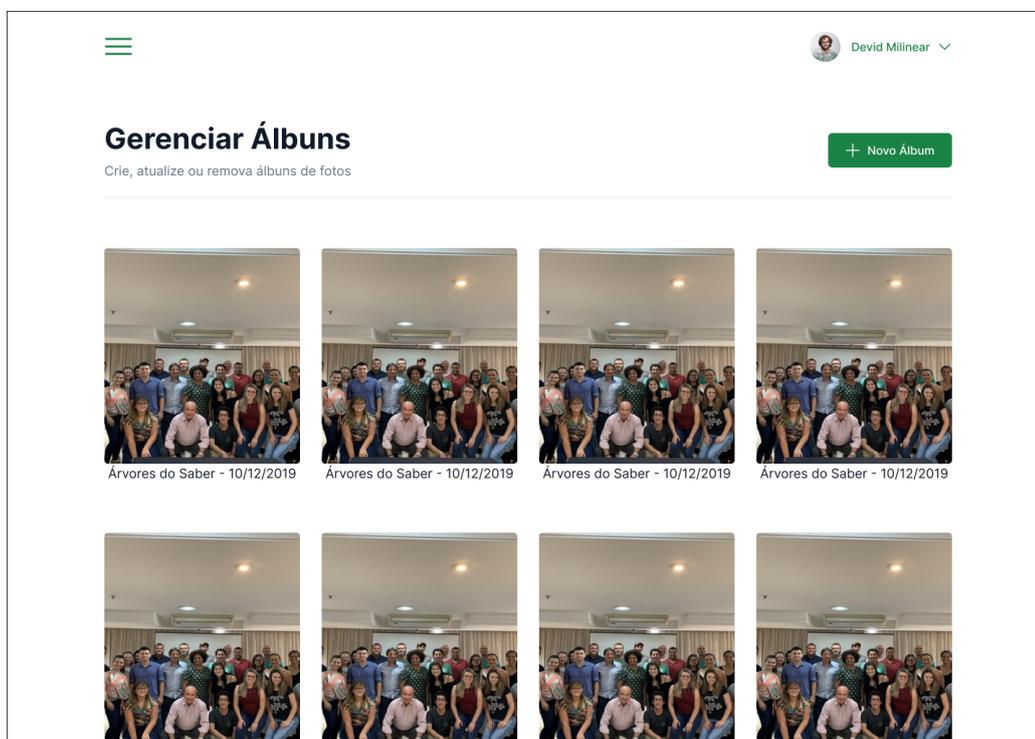
**Figura 103 – Histórico de Cursos**

A Figura 104 ilustra a seção onde os usuários podem visualizar e conectar-se com outros profissionais, Feature: Visualização de Contatos Públicos.



**Figura 104 – Contatos públicos**

A Figura 105 mostra a interface de gerenciamento de álbuns de fotos, permitindo ao administrador criar, editar e organizar álbuns, Feature: Gerenciamento de Fotos.



**Figura 105 – Gerenciamento de Álbuns**

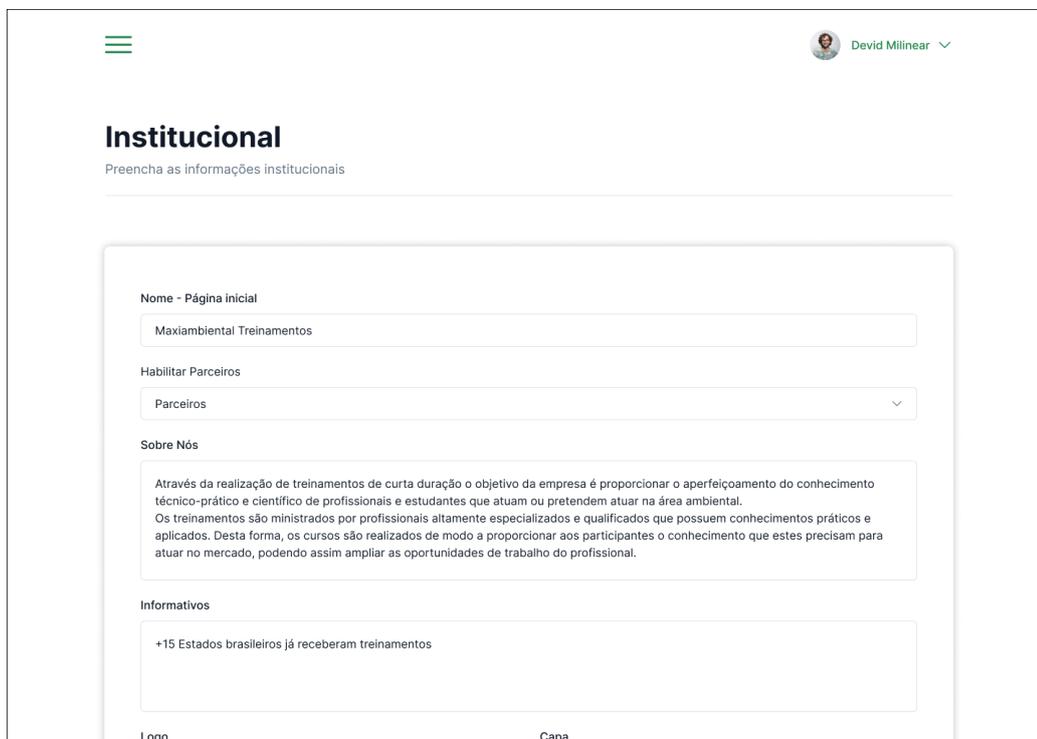
A Figura 106 ilustra o formulário para criação de um novo álbum, onde é necessário colocar todas as informações e fotos que compõem o álbum.

Figura 106 – Formulário Novo Álbum

A Figura 107 mostra a interface de gerenciamento de mensagens, permitindo ao administrador sanar as dúvidas dos usuários, Feature: Gerenciamento de Mensagens.

Figura 107 – Gerenciamento de Mensagens

A Figura 108 ilustra a interface de gerenciamento institucional, onde o administrador pode editar informações sobre a empresa Feature: Gerenciamento Institucional.



The screenshot displays a web interface for managing institutional information. At the top left is a hamburger menu icon, and at the top right is a user profile for 'David Miilinear'. The main heading is 'Institucional', followed by the instruction 'Preencha as informações institucionais'. The form contains several sections: 'Nome - Página inicial' with a text input field containing 'Maxiambiental Treinamentos'; 'Habilitar Parceiros' with a dropdown menu set to 'Parceiros'; 'Sobre Nós' with a text area containing a paragraph about training objectives; and 'Informativos' with a text area containing '+15 Estados brasileiros já receberam treinamentos'. At the bottom of the form are labels for 'Logo' and 'Capa'.

**Figura 108 – Gerenciamento do Institucional**