

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**QUEBRACÓDIGOS.COM: SITE DE APRENDIZAGEM DE PROGRAMAÇÃO
ORIENTADA A OBJETOS**

ÁLVARO RENAN COSTA FRADE

Cajazeiras

2025

ALVARO RENAN COSTA FRADE

**QUEBRACÓDIGOS.COM: SITE DE APRENDIZAGEM DE PROGRAMAÇÃO
ORIENTADA A OBJETOS**

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto

Cajazeiras

2025

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

F799q Frade, Álvaro Renan Costa.

Quebracódigos.com : site de aprendizagem de programação orientada a objetos / Álvaro Renan Costa Frade. – Cajazeiras, 2025.
56f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2025.

Orientador: Prof. Me. Francisco Paulo de Freitas Neto.

1. Desenvolvimento de sistemas. 2. Programação Orientada a Objetos. 3. Java. 4. Plataforma web. I. Instituto Federal da Paraíba. II. Título.

IFPB/CZ

CDU: 004.4(043.2)



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

ALVARO RENAN COSTA FRADE

**SITE DE APRENDIZAGEM DE PROGRAMAÇÃO ORIENTADA A
OBJETOS**

Trabalho de Conclusão de Curso apresentado junto ao
Curso Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas do Instituto Federal de
Educação, Ciência e Tecnologia da Paraíba - Campus
Cajazeiras, como requisito à obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto

Aprovada em: **21 de Março de 2025.**

Prof. Me. Francisco Paulo de Freitas NEto - Orientador

Prof. Me. Ricardo de Sousa Job - Avaliador

IFPB - Campus Cajazeiras

Prof. Esp. João Igor Barros Rocha - Avaliador

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Francisco Paulo de Freitas Neto**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 22/03/2025 10:13:21.
- **Ricardo de Sousa Job**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 22/03/2025 19:30:30.
- **Joao Igor Barros Rocha**, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO, em 24/03/2025 08:28:20.

Este documento foi emitido pelo SUAP em 22/03/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 686529
Verificador: d43e3bb87a
Código de Autenticação:



Rua José Antônio da Silva, 300, Jardim Oásis, CAJAZEIRAS / PB, CEP 58.900-000
<http://ifpb.edu.br> - (83) 3532-4100

Dedico este trabalho a meus tios João Robson (In memoriam) e Fátima que sempre me auxiliaram e cuidaram de mim, mesmo morando longe.

AGRADECIMENTOS

A Deus, por me dar não somente a vida, mas também sabedoria e forças para continuar a minha jornada. “E tudo quanto fizerdes, fazei-o de todo o coração, como ao Senhor, e não aos homens” Colossenses 3:23.

Aos meus pais Sterfani e Máxima, por sempre terem me educado e me apoiado, inclusive na escolha do curso, e me dado os melhores conselhos. Sem eles nunca teria chegado onde estou agora.

Ao meu irmão mais novo, Théo, por sempre me alegrar enquanto passamos momentos divertidos juntos. Às vezes brigamos, mas tudo isso faz parte deste grande laço que chamamos de fraternidade.

Aos meus familiares, por sempre terem cuidado de mim e me dado todo o amor do mundo. Amo todos vocês de coração.

Aos meus amigos Caio e Débora, que estiveram comigo desde antes da faculdade e iniciaram junto a mim essa jornada. Obrigado por terem continuado comigo mesmo após a formatura do colégio.

Aos meus amigos Alana e Franks, que conheci durante o curso, mas que me ajudaram muito enquanto trocamos conhecimentos e nos divertimos. Vocês iluminaram meu caminho e também me encorajaram nos momentos difíceis.

A todos os meus amigos, que não posso citar aqui, pois são vários. Obrigado por toda a ajuda que me deram e também pela experiência que me forneceram ajudando vocês.

Aos professores Cristiano Alves Fontes e Francisco Paulo de Freitas Neto, professores e orientadores, e ao professor Diogo Dantas Moreira, que me auxiliaram bastante no desenvolvimento deste projeto.

À empresa Loopis Jr. por ter me acolhido e ajudado a aperfeiçoar habilidades que irão alavancar minha carreira profissional no futuro. Todo o tempo que passei lá foi precioso e não me arrependo de ter feito parte da empresa.

À empresa RedeNet Comércio e Soluções em TI, onde eu aprendi muitas das habilidades que eu possuo hoje, inclusive para concluir este projeto. As pessoas da

empresa não foram apenas colegas de trabalho e chefes, como também amigos que me ajudaram sempre que eu precisei.

A todos do IFPB Campus Cajazeiras, por terem me ensinado todo o conhecimento que possuo desde o básico até conceitos mais avançados.

“A única maneira de fazer algo excelente é amar o que você faz. Se você ainda não a encontrou, continue procurando. Não se acomode.”

Steve Jobs

RESUMO

A importância dos softwares aumenta a cada ano, e com isso surge a necessidade de uma maneira mais eficiente e versátil de se desenvolver aplicações. Para isso, foi criado o paradigma da Programação Orientada a Objetos (POO). Nesse contexto, várias linguagens de programação orientadas a objetos começaram a ser desenvolvidas, sendo a linguagem de programação Java, criada pela empresa Sun Microsystems, uma das mais importantes. Diante disso, este trabalho visa implementar uma plataforma web feita em Typescript utilizando a biblioteca React para a construção da interface, além de uma API feita com o framework NestJS e a base de dados MongoDB. A plataforma permitirá que os usuários aprendam sobre programação orientada a objetos, linguagem de programação Java e outros temas relacionados, em uma única plataforma de acesso gratuito escrita em língua portuguesa, além de uma área onde poderão trocar conhecimento enviando dúvidas e respondendo perguntas de outros usuários, facilitando a jornada daqueles que buscam se inserir nessa área.

Palavras-chave: Programação Orientada a Objetos; Java; Plataforma Web;

ABSTRACT

As the importance of software has grown over the years, so has the need for a more efficient and versatile way to develop applications. To this end, the Object-Oriented Programming (OOP) paradigm was created. In this context, several object-oriented programming languages have been developed, of which the Java programming language, created by Sun Microsystems, is one of the most important. In this context, this work aims to implement a web platform written in Typescript using the React library to build the interface, as well as an API built with the NestJS framework and a MongoDB database. The platform will allow users to learn about object-oriented programming, the Java programming language and other related topics on a single, freely accessible platform written in Portuguese, as well as an area where they can exchange knowledge by asking questions and answering questions from other users, making the journey easier for those who want to enter the field.

Keywords: Object Oriented Programming, Java, Web Platform.

LISTA DE FIGURAS

Figura 1 – Página de tutorial do W3Schools	17
Figura 2 – Página principal do Stack Overflow	18
Figura 3 – Exemplo de componente React	27
Figura 4 – Diagrama de casos de uso	30
Figura 5 – Representação do modelo conceitual do sistema	31
Figura 6 – Diagrama de modelo lógico de dados	33
Figura 7 – Primeira parte da página principal	37
Figura 8 – Segunda parte da página principal	38
Figura 9 – Página de login	38
Figura 10 – Página de cadastro	39
Figura 11 – Página de curso na aba “sobre”	40
Figura 12 – Página de curso na aba “aulas”	41
Figura 13 – Página de aula	42

LISTA DE QUADROS

Quadro 1 – Comparativo entre as o "Quebracódigos.com" e as outras plataformas apresentadas	19
Quadro 2 – Atividades do projeto	20
Quadro 3 – Requisitos funcionais da aplicação "Quebracódigos.com"	28
Quadro 4 – Requisitos não funcionais da aplicação "Quebracódigos.com"	29
Quadro 5 – Caso de Uso: Ver cursos	47
Quadro 6 – Caso de Uso: Ver aulas	47
Quadro 7 – Caso de Uso: Ver postagens	48
Quadro 8 – Caso de Uso: Ver comentários	48
Quadro 9 – Caso de Uso: Fazer Login	49
Quadro 10 – Caso de Uso: Criar Conta	49
Quadro 11 – Caso de Uso: Criar pergunta	50
Quadro 12 – Caso de Uso: Editar pergunta	50
Quadro 13 – Caso de Uso: Excluir pergunta	51
Quadro 14 – Caso de Uso: Criar resposta	51
Quadro 15 – Caso de Uso: Editar resposta	52
Quadro 16 – Caso de Uso: Excluir resposta	52
Quadro 17 – Caso de Uso: Criar comentário	53
Quadro 18 – Caso de Uso: Editar comentário	53
Quadro 19 – Caso de Uso: Excluir comentário	54
Quadro 20 – Caso de Uso: Criar curso	54
Quadro 21 – Caso de Uso: Editar curso	55
Quadro 22 – Caso de Uso: Excluir curso	55
Quadro 23 – Caso de Uso: Criar aula	56
Quadro 24 – Caso de Uso: Editar aula	56
Quadro 25 – Caso de Uso: Excluir aula	57

ABREVIATURAS E SIGLAS

ACID	<i>Atomicity, Consistency, Isolation and Durability</i>
API	<i>Application Programming Interface</i>
BASE	<i>Basically Available, Soft State, Eventual Consistency</i>
BSON	<i>Binary JSON</i>
CSS	<i>Cascade Style Sheets</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTML	<i>HyperText Markup Language</i>
JSON	<i>JavaScript Object Notation</i>
NoSQL	<i>Not only SQL</i>
POO	Programação Orientada a Objetos
RDBMS	<i>Relational Database Management System</i>
REST	<i>Representational State Transfer</i>
RF	Requisito Funcional
RNF	Requisito Não Funcional
SQL	<i>Sequenced Query Language</i>
TCC	Trabalho de Conclusão de Curso
UC	<i>Use Case</i>
UI	<i>User Interface</i>

1 INTRODUÇÃO

A criação dos *softwares* deu início à atual era da informação digital, onde programas são utilizados em todos os lugares do dia-a-dia das pessoas. Esses *softwares* são criados a partir de linhas de código que os computadores interpretam como comandos para que determinadas ações dentro do programa sejam executadas. No entanto, a estrutura do código também influencia na maneira como essas tarefas são executadas, visto que existem vários jeitos de se escrever comandos que significam a mesma ação. Para isso, existem os *paradigmas de programação*, que são as diferentes maneiras com que um código pode ser estruturado de forma que haja a saída desejada.

De acordo com Jungthor e Goulart (2009), cada paradigma determina uma forma particular de abordar os problemas e de formular respectivas soluções. Além disso, uma linguagem de programação pode combinar dois ou mais paradigmas para potencializar as análises e soluções. Deste modo, cabe ao programador escolher o paradigma mais adequado para analisar e resolver cada problema. Dentre os paradigmas mais utilizados, está a *Programação Orientada a Objetos (POO ou OOP da nomenclatura em inglês)*.

Nesse paradigma, o código é dividido em “blocos” denominados *objetos*. Esses objetos são responsáveis por realizar essas funções individualmente, também sendo possível a troca de dados entre os objetos. Isso permite a reusabilidade de trechos de código, uma das características diferenciais do paradigma. Segundo P. Deitel e H. Deitel (2017), objetos ou, mais precisamente, as classes de onde os objetos vêm são essencialmente componentes reutilizáveis de software. Há objetos data, objetos data/hora, objetos áudio, objetos vídeo, objetos automóvel, objetos pessoas etc. Quase qualquer substantivo pode ser razoavelmente representado como um objeto de software em termos dos atributos (por exemplo, nome, cor e tamanho) e comportamentos (por exemplo, calcular, mover e comunicar).

Hoje, como a demanda por software novo e mais poderoso está aumentando, construir softwares de maneira rápida, correta e econômica continua a ser um

objetivo indefinido. Existem muitas linguagens de programação que possuem orientação a objetos. Porém, uma das principais e mais utilizadas atualmente para este fim é a linguagem de programação Java¹, que está presente em inúmeros dispositivos e é usada frequentemente por dezenas de milhões de desenvolvedores. Sua flexibilidade a permite ser usada para desenvolvimento de aplicações para empresas, jogos eletrônicos, programação para eletrodomésticos, entre outros meios.

No entanto, dada a complexidade da linguagem de programação, muitos estudantes costumam ter mais dificuldade em aprender seus conceitos, principalmente quando esses já possuem uma certa experiência com alguma outra linguagem mais simples. Diante desse problema, este trabalho visa o desenvolvimento de uma aplicação web onde é possível aprender sobre tais conceitos de maneira fácil, além de poderem também revisar o que já haviam aprendido. Além disso, os usuários também terão acesso a um fórum dentro da plataforma, onde poderão postar dúvidas em relação ao conteúdo abordado e/ou responder às perguntas de outros usuários.

1.1 Problemática

Devido à grande quantidade de programas que utilizam a linguagem Java, muitas pessoas que estão iniciando sua jornada de aprendizagem de programação, bem como aqueles que já estão na área há algum tempo, buscam aprendê-la para eventualmente usá-la no trabalho ou apenas para agregar ao seu conhecimento no caso de uma eventual necessidade. Com isso, muitos procuram utilizar sites da Internet como plataforma para auxiliar em sua aprendizagem.

A partir de observações e pesquisas na Internet, foi visto que, até o momento da construção deste projeto, não há uma plataforma que engloba tanto Orientação a Objetos e conceitos relacionados, quanto a linguagem Java e que tenha acesso gratuito. Com isso, os usuários precisam acessar várias plataformas que muitas vezes não são muito claras em seus ensinamentos e que podem contradizer umas às outras, já que todo desenvolvedor costuma possuir uma maneira diferente de escrever seus códigos.

¹ <https://docs.oracle.com/javase/8/docs/technotes/guides/language/index.html>

Além disso, muitos dos *websites* para ensino de programação estão localizados em língua Inglesa, dificultando o aprendizado daqueles que possuem pouca ou nenhuma experiência com o idioma. Por esses e outros motivos, alguns desistem de tentar aprender a linguagem pela dificuldade de entendimento sobre o que está sendo lecionado na plataforma, perdendo assim uma grande oportunidade de ampliar suas carreiras.

1.2 Descrição da solução

A partir do momento em que o paradigma de orientação a objetos foi desenvolvido, várias mudanças benéficas ocorreram com relação ao desenvolvimento de sistemas em diversos contextos. E tendo em vista que muitos sistemas hoje em dia utilizam programação orientada a objetos em sua estrutura, o conhecimento deste assunto tornou-se amplamente requisitado em empresas ao redor do mundo, o que levou várias a pessoas despertarem o interesse nessa área, buscando qualificação para melhor inserir-se no mercado de trabalho.

Este projeto visa auxiliar essas pessoas dando-lhes a oportunidade de aprender com mais facilidade os conceitos necessários para a introdução no tema bem como conceitos mais avançados para aqueles que já estão imersos na área e procuram aperfeiçoar suas habilidades. Sendo uma aplicação de acesso gratuito e que engloba os principais temas relativos à área, o eventual esforço daqueles que procuram pelo conhecimento do tema será diminuído, já que o site compreende em um único lugar o conteúdo que antes necessitaria de visita a várias plataformas diferentes.

1.3 Ferramentas Relacionadas

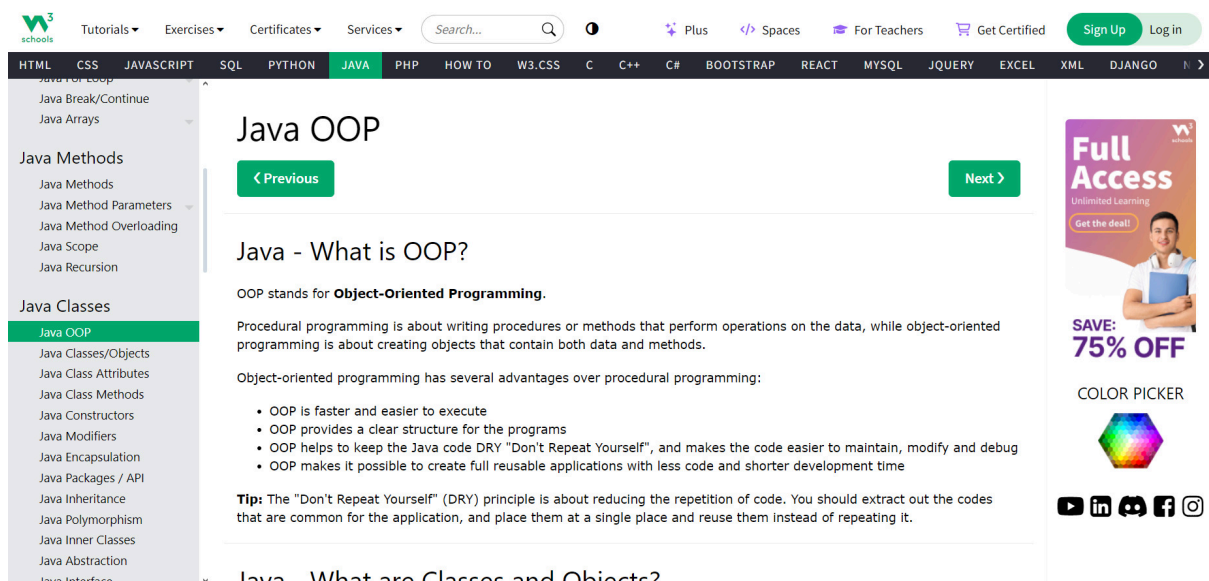
Nesta seção, serão mostradas ferramentas que possuem um objetivo similar à plataforma proposta neste trabalho. Por meio de pesquisas na internet e experiências do autor deste projeto, foram escolhidas as ferramentas *W3Schools* e *Stack Overflow*.

1.3.1 W3Schools

W3Schools é um site criado com a finalidade de aprendizagem de tecnologias voltadas para o desenvolvimento web, como HTML, CSS e Javascript,

oferecendo vários tutoriais e também uma ferramenta chamada *TryIt Editor*, onde é possível executar e editar os exemplos de código integrados nos tutoriais dentro do próprio site. A plataforma foi lançada em 1998 pela Refsnes Data, uma empresa norueguesa de desenvolvimento e consultoria de software e seu nome é inspirado na *World Wide Web*, apesar de não ter nenhuma relação com o W3C (*World Wide Web Consortium*). A Figura 1 apresenta uma captura de tela de uma das páginas de tutorial do W3Schools.

Figura 1 - Página de tutorial do W3Schools



Fonte: W3Schools, 2025

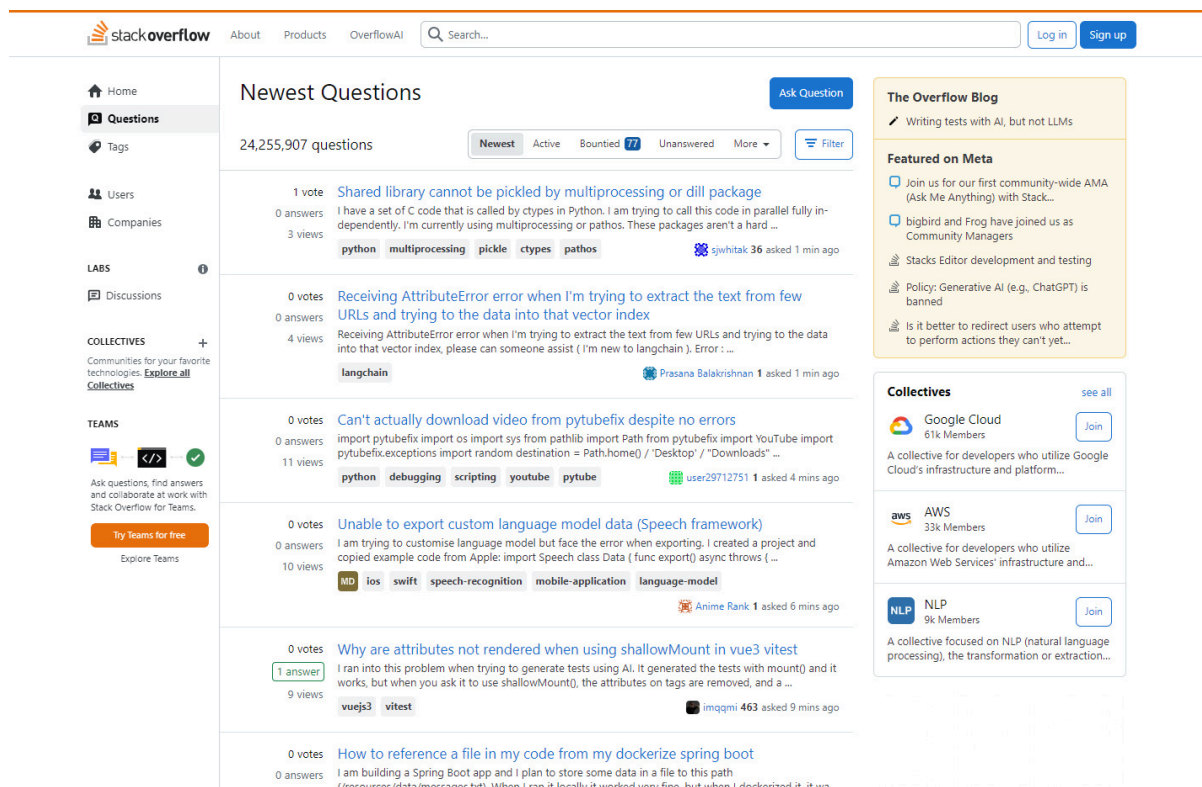
No entanto, como descrito acima, os tutoriais desta plataforma são voltados apenas para o desenvolvimento web. Apesar de existir um módulo para ensino da linguagem Java e, consequentemente, orientação a objetos, as explicações presentes são bastante resumidas, e os conceitos não são apresentados de maneira geral, mas sim de acordo com cada linguagem.

1.3.2 Stack Overflow

Stack Overflow é um site lançado em 2008 por Jeff Attwood e Joel Spolsky e administrado pela *Stack Exchange Inc.* Trata-se de um fórum onde usuários podem enviar perguntas sobre determinado assunto referente a tecnologia, e outros usuários podem respondê-las. Essa estratégia permite uma maior interação entre a comunidade de desenvolvedores e auxilia no aprendizado de vários temas, obtendo

respostas de usuários mais experientes. Além disso, a plataforma oferece outros tipos de serviço como ferramentas para empresas. A Figura 2 exibe a página principal do Stack Overflow.

Figura 2 - Página principal do Stack Overflow



Fonte: Stack Overflow, 2025

O site também conta com uma versão em português² criada em 2013 que possui o mesmo objetivo, porém centralizada no público falante da língua portuguesa, especialmente brasileiros. A plataforma proposta neste trabalho contém uma seção, denominada “Fórum”, onde os usuários poderão interagir entre si por meio de perguntas e respostas, assim como no *Stack Overflow*. Contudo, como mencionado anteriormente, não se limita a essa proposta, tendo como principal objetivo a disponibilização de aulas sobre orientação a objetos e temas relacionados.

Da mesma forma que o *Stack Overflow*, postagens feitas na seção “fórum” do “Quebracódigos.com” podem ser marcadas como “duplicada” caso exista outra postagem com conteúdo e finalidade similares.

² <https://pt.stackoverflow.com>

1.3.3 Google Classroom

Lançado em 2014, o *Google Classroom* – ou Google Sala de Aula, em português – é uma plataforma criada pela Google para compartilhamento de materiais e tarefas em uma turma, podendo ser acessada via *web* ou pelo seu aplicativo em dispositivos móveis. Nele, professores podem criar turmas onde podem enviar materiais sobre os assuntos discutidos em aulas, criar trabalhos para os alunos enviarem e atribuir notas a esses. Além disso, os alunos podem mandar comentários com perguntas de forma geral, onde todos podem ver, ou diretamente para o professor.

A principal diferença entre o “Quebracódigos.com” e o *Google Classroom* é que, os usuários podem acessar as aulas sem a necessidade de serem cadastrados em uma turma ou sequer ter uma conta cadastrada no site, já que elas são de acesso completamente público. No entanto, o usuário ainda precisa estar logado no site para poder enviar perguntas, respostas e comentários.

1.3.4 Comparação Entre as Plataformas Apresentadas

O Quadro 1 mostra a comparação entre as principais funcionalidades da aplicação proposta neste trabalho e as ferramentas apresentadas nesta seção.

Quadro 1 - Comparativo entre o “Quebracódigos.com” e as outras plataformas apresentadas

	W3Schools	Stack Overflow	Google Classroom	Quebracódigos.com
Tutoriais completos sobre orientação a objetos	Sim	Não	Não ³	Sim
Acesso gratuito	Sim	Sim	Sim	Sim
Acesso público	Sim	Sim	Não	Sim
Conteúdo em português	Não	Sim	Sim	Sim
Fórum online para compartilhamento de dúvidas	Não	Sim	Sim	Sim

Fonte: Elaborado pelo autor

³ Os tutoriais estão disponíveis apenas caso o usuário esteja em uma turma com aulas sobre Orientação a Objetos.

1.4 Objetivos

1.4.1 Objetivo Geral

O objetivo deste projeto é facilitar o aprendizado do paradigma de Orientação a Objetos e linguagem de programação Java por meio de um *website* de acesso gratuito contendo cursos sobre o assunto.

1.4.2 Objetivos Específicos

- Desenvolver um site com cursos relacionados a Orientação a Objetos;
- Tornar público o acesso às aulas de tais cursos;
- Possibilitar a postagem de dúvidas relacionadas ao conteúdo proposto.

1.5 Atividades

No processo de criação e construção deste TCC, foram realizadas as seguintes atividades, descritas no Quadro 2.

Quadro 2 - Atividades do projeto

	ATIVIDADE	DESCRIÇÃO
A1	Elaboração e pesquisas	Escolha do tema do projeto e pesquisas para compor a fundamentação teórica do mesmo.
A2	Elicitação de requisitos	Levantamento dos requisitos funcionais e não-funcionais que o sistema deve cumprir.
A3	Seleção das tecnologias	Seleção das tecnologias que foram utilizadas no desenvolvimento do projeto.
A4	Elaboração do protótipo	Construção de um protótipo para guiar o desenvolvimento da interface do sistema.
A5	Elaboração do documento	Escrita do documento do TCC.
A6	Construção do sistema	Início do desenvolvimento da interface e da <i>API</i> .
A7	Refinamento do protótipo	Melhorias feitas no protótipo para a interface ao longo do desenvolvimento da aplicação.
A8	Atualização do documento	Inclusão de informações no documento do projeto conforme a construção do sistema.

Fonte: Elaborado pelo autor

1.6 Organização do Documento

As próximas seções deste documento contêm a fundamentação teórica, onde será abordado o embasamento teórico utilizado para a idealização deste projeto e construção do sistema proposto. Logo após, será explicado como ocorreu a implementação da plataforma, desde o levantamento de requisitos e a elaboração do protótipo, até as tecnologias utilizadas no desenvolvimento do sistema e as telas presentes no mesmo. Finalmente, serão exibidas as considerações finais de acordo com o conteúdo mostrado ao longo do documento.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão apresentados os principais conceitos e tecnologias usados para o desenvolvimento deste trabalho. Os temas abordados serão: Orientação a Objetos, *RESTful API*, *NoSQL* e *React*.

2.1 Orientação a objeto

De acordo com Farinelli (2007), ao longo de várias pesquisas e testes, verificou-se que a reutilização era a peça chave para o aumento da produtividade e melhoria da qualidade dos softwares. A reutilização significa usar novamente algo que foi feito. Seu objetivo é permitir uma ampla utilização de todos os tipos de informações encontradas na situação de desenvolvimento. Dessa forma, a programação orientada a objetos foi desenvolvida com base nos objetos, que contêm funções chamadas de métodos que podem ser utilizadas e reutilizadas em qualquer parte do código.

Um dos grandes diferenciais da programação orientada a objetos em relação a outros paradigmas de programação que também permitem a definição de estruturas e operações sobre essas estruturas está no conceito de herança, mecanismo através do qual definições existentes podem ser facilmente estendidas. Juntamente com a herança deve ser enfatizada a importância do polimorfismo, que permite selecionar funcionalidades que um programa irá utilizar de forma dinâmica, durante sua execução. (RICARTE, 2001)

2.1.1 Linguagem Java

Dentro do universo das linguagens de programação orientadas a objeto, a linguagem Java é a mais popular e mais utilizada pelos desenvolvedores no mercado. A linguagem de programação Java foi desenvolvida em 1995 por um grupo de programadores liderado por *James Gosling* da empresa *Sun Microsystems*, e foi adquirida pela empresa *Oracle Corporation* em 2008.

Java foi introduzida como uma linguagem de programação orientada a objetos simples e segura. É uma linguagem única nisso, sendo uma linguagem nova nesse período, foi capaz de atrair bastante interesse da comunidade de computação

(POO; KIONG; ASHOK, 2007). Atualmente, a maioria dos dispositivos possui integração com a linguagem, possuindo nativamente um método de executá-la. Além disso, alguns eletrodomésticos inteligentes possuem sistema programado em Java.

Ademais, a linguagem de programação Java também é bastante utilizada para o desenvolvimento de aplicações, tanto para *desktop*, com várias ferramentas disponíveis para auxílio ao desenvolvedor, quanto para dispositivos móveis, sendo uma das linguagens mais utilizadas para esse meio e possuindo uma comunidade mais estabelecida em relação às outras linguagens.

2.1.2 Padrões de Projeto

Um dos temas mais importantes na utilização de linguagens orientadas a objetos são os Padrões de Projeto de Software. Estes são convenções criadas pelos desenvolvedores para que os códigos-fonte dos sistemas tenham uma organização considerável e características semelhantes uns aos outros, bem como facilitar a leitura e previsibilidade. “Em uma equipe ágil, o quadro geral evolui junto com o software. Com cada adição, a equipe melhora o design do sistema de modo que ele seja tão bom quanto pode ser para o sistema como ele é no momento.” (MARTIN, 2003)

2.2 RESTful API

Uma API (*Application Programming Interface* ou Interface de Programação de Aplicação, em tradução literal) é um mecanismo que permite a interação entre softwares e seus componentes. Em sua obra, Gough, Bryant e Auburn (2023) consideram uma API como uma abstração da implementação subjacente, representada por uma especificação que introduz tipos. Desenvolvedores podem entender as especificações e usar as ferramentas para gerar código em múltiplas linguagens e implementar um consumidor de API (o software que faz uso de uma API).

O padrão *REST* (*Representational State Transfer* ou Transferência de Estado Representacional, em tradução literal) consiste em um modelo de arquitetura de sistemas proposto por Roy Thomas Fielding no ano 2000. O modelo tem como

objetivo definir um conjunto de regras para o desenvolvimento de sistemas distribuídos cuja comunicação se estabelece via rede. Conforme Fielding (2000), o estilo REST é uma abstração dos elementos arquiteturais em um sistema distribuído. O REST ignora os detalhes da implementação dos componentes e a sintaxe de protocolos para focar nas funções dos componentes, as restrições das interações entre os componentes e sua interpretação de elementos significativos de dados.

Com isso, uma API é considerada *RESTful* quando ela aplica o modelo REST em sua arquitetura. Por meio de requisições HTTP (*HyperText Transfer Protocol* ou Protocolo de Transferência de Hipertexto, em tradução literal), o cliente se comunica com o servidor e, no processo, dados são enviados entre um e outro para que uma aplicação funcione. Com isso, é possível que o usuário interaja com o sistema de acordo com suas permissões e com as funcionalidades da aplicação.

2.3 NoSQL

O modelo NoSQL (*Not only SQL* ou Não apenas SQL, em tradução direta) consiste em uma abordagem de gerenciamento de bancos de dados de forma não-relacional cuja estrutura permite uma maior flexibilidade, escalabilidade e desempenho do que os bancos de dados relacionais.

Os RDBMS podem “aumentar a escala”, mas não podem “diminuir a escala”, portanto, são adequados para uma forte consistência e disponibilidade. A maioria dos dados NoSQL é particionada e replicada em vários nós. Inerentemente, a maioria usa o MapReduce do Google, o Hadoop Distributed File System ou o Hadoop MapReduce para agregação de dados (CHANDRA, 2015).

Segundo Nayak, Poriya e Poojary (2013), diferentemente dos RDBMS, bancos de dados NoSQL são projetados para facilmente escalonados à medida que crescem. A maioria dos sistemas NoSQL removeram o suporte à multi-plataforma e mais algumas funcionalidades desnecessárias dos RDBMS, deixando-os mais leves e eficientes do que suas contrapartes RDMS. O modelo de dados NoSQL não garante as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade), mas sim as propriedades BASE (Basicamente disponível, Estado leve e Consistência eventual).

2.3.1 Bancos de Dados Orientados a Documentos

Bancos de dados orientados a documentos são baseados no armazenamento de pares de chave-valor, tendo um esquema altamente flexível. Esta característica torna os bancos de dados orientados a documentos ótimas opções para dados semi-estruturados, como os utilizados em ferramentas web colaborativas (OLIVEIRA, 2014).

No modelo orientado a documentos, os dados ficam armazenados em coleções de documentos, os quais possuem um esquema de chave-valor, ou seja, campos (chaves) que armazenam os dados (valor). Uma das vantagens de se utilizar um banco de dados orientado a documentos é a sua flexibilidade. Por não se basear em esquemas rígidos, como ocorre nos bancos de dados relacionais, os documentos de uma coleção podem conter campos distintos. Além disso, é possível atualizar um documento excluindo determinados campos ou inserindo novos, o que não é permitido em um banco relacional.

2.3.1.1 MongoDB

Sendo um banco de dados orientado a documentos, o *MongoDB* armazena os dados em documentos, estes escritos em formato BSON (abreviação de *Binary JSON*), um formato similar ao JSON e que também suporta os mesmos tipos de dados, criado pela própria equipe do MongoDB. Ademais, a ferramenta foi criada na linguagem C++, o que facilita a portabilidade e execução da aplicação em diversos ambientes.

De acordo com Hows, Membrey e Plugge (2019), o BSON torna o MongoDB mais rápido ainda ao fazer com que seja muito mais fácil para um computador processar e efetuar pesquisas nos documentos. O BSON também acrescenta alguns recursos que não estão disponíveis no JSON padrão, incluindo a capacidade de adicionar tipos para tratamento de dados binários.

2.4 React

ReactJS é uma biblioteca JavaScript implantada para desenvolver componentes interfaces de usuário (UI) reutilizáveis. De acordo com sua documentação oficial, React é uma biblioteca para construção de interfaces de

usuário modulares. Basicamente proporciona o desenvolvimento de grandes e complexas aplicações baseadas em web, as quais podem alterar seus dados sem que as páginas sejam recarregadas (AGGARWAL, 2018).

A biblioteca React, desenvolvida pelo Facebook (atual Meta), é uma biblioteca feita em JavaScript para a construção de interfaces de usuário em sistemas web e de código aberto, permitindo com que a comunidade de desenvolvedores possa criar novos recursos e contribuir com o projeto. Com o React, é possível programar uma interface de usuário através da sintaxe JSX, uma extensão da sintaxe comum do JavaScript que permite com que funções retornem elementos HTML. Isso garante a possibilidade de introduzir valores de variáveis dentro de um documento HTML, fazendo com que a interface se torne mais dinâmica e facilitando a interação do usuário com a aplicação. Atualmente, muitas empresas, como Netflix, Airbnb e Twitter, utilizam React em suas aplicações web.

Segundo Gackenhimer (2015), olhando para a motivação por trás do React, é possível observar que foi criado para suprir uma necessidade específica de um conjunto específico de desafios tecnológicos enfrentados pelo Facebook. Esses desafios não são e nunca foram únicos à empresa, porém, o que eles fizeram foi enfrentá-los diretamente com uma abordagem para resolver o problema.

A estruturação de uma aplicação web em React é feita através de componentes, os quais não necessariamente possuem uma ligação entre si. Na Figura 3 é mostrado um exemplo de componente React.

Figura 3 - Exemplo de componente React

```
1  import React from 'react';
2
3  export default function Example() {
4    const title = 'Hello World!';
5
6    return (
7      <div>
8        <h1>{title}</h1>
9      </div>
10   );
11 }
```

Fonte: Elaborado pelo autor, 2025

Neste exemplo, é possível ver os elementos HTML sendo implementados dentro de uma função JavaScript, com o auxílio de uma variável para construção do componente.

Os componentes criados são, então, unidos em um único componente que representa uma página da aplicação. E, utilizando bibliotecas e frameworks de roteamento de páginas, é possível criar uma aplicação web com várias páginas e um sistema de navegação entre as mesmas.

3 PROJETO

Neste capítulo, serão exibidas as etapas que foram necessárias para a construção do sistema, como o levantamento de requisitos, a moldagem do protótipo, a arquitetura do *software* e as tecnologias utilizadas no desenvolvimento.

3.1 Requisitos

Nesta seção, serão mostrados os requisitos e os casos de uso *software*. Como afirma Mendonça (2014), o levantamento de requisitos desempenha um papel importante na construção de um sistema de informação, pois é o início para toda a atividade de desenvolvimento de software. É onde o analista faz as primeiras reuniões com os clientes e/ou usuários para conhecer as funcionalidades do sistema que será desenvolvido. O Quadro 2 apresenta os requisitos funcionais levantados para o desenvolvimento da plataforma descrita neste documento.

Quadro 3 - Requisitos funcionais da aplicação Quebracódigos.com

ID	REQUISITO	DESCRIÇÃO
RF_01	Ver cursos	Permite que o usuário tenha acesso aos cursos presentes na plataforma.
RF_02	Ver aulas	Permite que o usuário tenha acesso às aulas dos cursos presentes na plataforma.
RF_03	Ver postagens	Permite que o usuário veja as postagens no fórum.
RF_04	Ver comentários	Permite que o usuário veja os comentários das postagens no fórum.
RF_05	Criar conta	Permite que o usuário crie uma conta na plataforma.
RF_06	Fazer login	Permite que o usuário faça login em sua conta na plataforma.
RF_07	Gerenciar perguntas	Permite que um usuário logado na plataforma crie, edite, e exclua suas perguntas no fórum.
RF_08	Gerenciar respostas	Permite que um usuário logado na plataforma crie, edite e exclua suas respostas em uma pergunta no fórum.
RF_09	Gerenciar comentários	Permite que um usuário logado na plataforma envie, edite e exclua seus comentários em uma pergunta no fórum.
RF_10	Gerenciar cursos	Permite que um usuário administrador cadastre, edite e exclua seus cursos na plataforma.
RF_11	Gerenciar aulas	Permite que um usuário administrador crie, edite e exclua aulas em um curso cadastrado pelo mesmo na plataforma.

Fonte: Elaborado pelo autor, 2025

Para que o sistema funcione da maneira desejável, faz-se necessário apontar também os requisitos não funcionais que a aplicação deve cumprir. De acordo com Vazquez e Simões (2016), enquanto os requisitos funcionais referem-se especificamente a uma tarefa do usuário, os requisitos não funcionais indicam restrições de ordem geral que abordam aspectos relativos, por exemplo, ao ambiente, à organização, à implementação e à qualidade. O Quadro 3 indica os requisitos não funcionais identificados com base no levantamento de requisitos funcionais realizado.

Quadro 4 - Requisitos não funcionais da aplicação Quebracódigos.com

ID	REQUISITO	DESCRIÇÃO
RNF_01	Restringir funcionalidades	Garantir que o acesso a certas funcionalidades da aplicação seja restringido ao tipo do usuário.

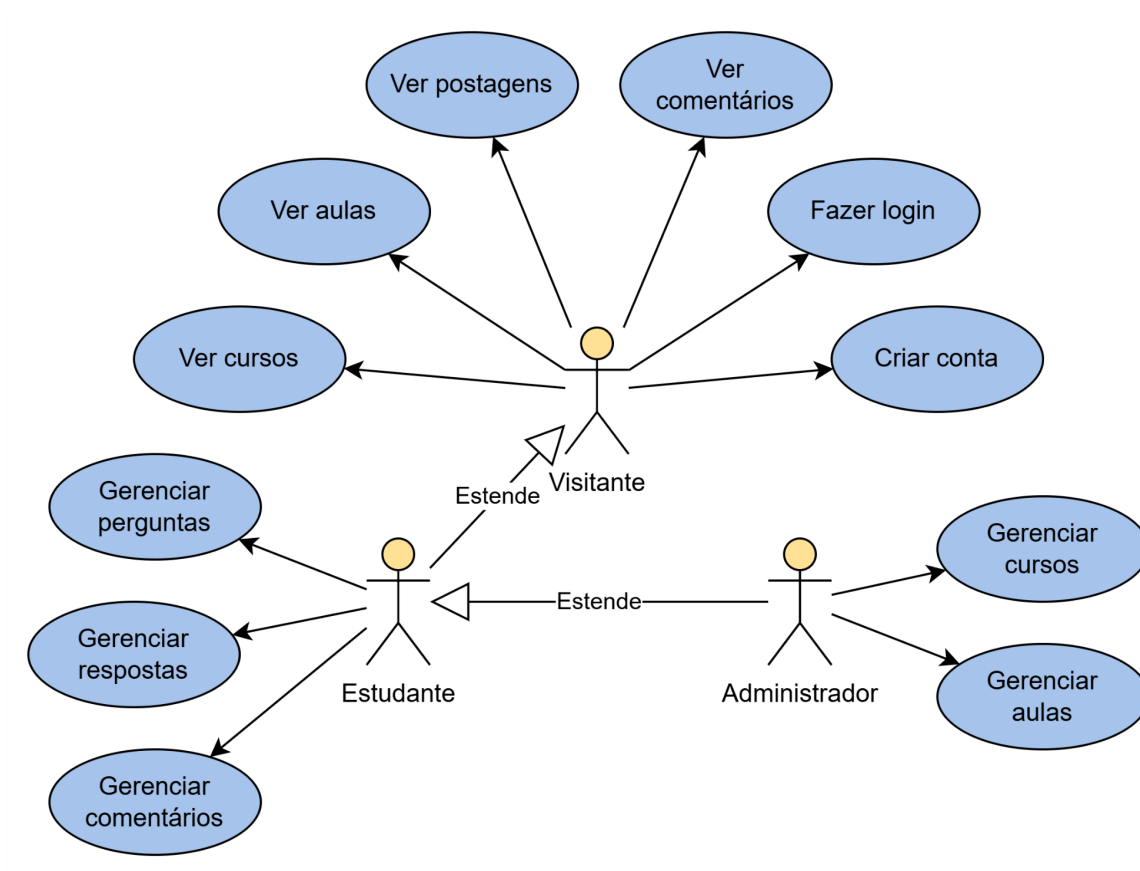
Fonte: Elaborado pelo autor, 2025

3.1.1 Casos de Uso

Um caso de uso captura um contrato entre os *stakeholders* de um sistema sobre o seu comportamento. O caso de uso descreve o comportamento de um sistema sob diversas condições conforme o sistema responde a uma requisição de um dos *stakeholders*, chamado *ator primário* (COCKBURN, 2005).

Com base nos requisitos funcionais apresentados no quadro 2, foi construído o diagrama de casos de uso da aplicação, apresentado na Figura 4.

Figura 4 - Diagrama de casos de uso



Fonte: Elaborado pelo autor, 2025

No diagrama acima, são apresentados três atores que utilizarão o sistema, sendo eles:

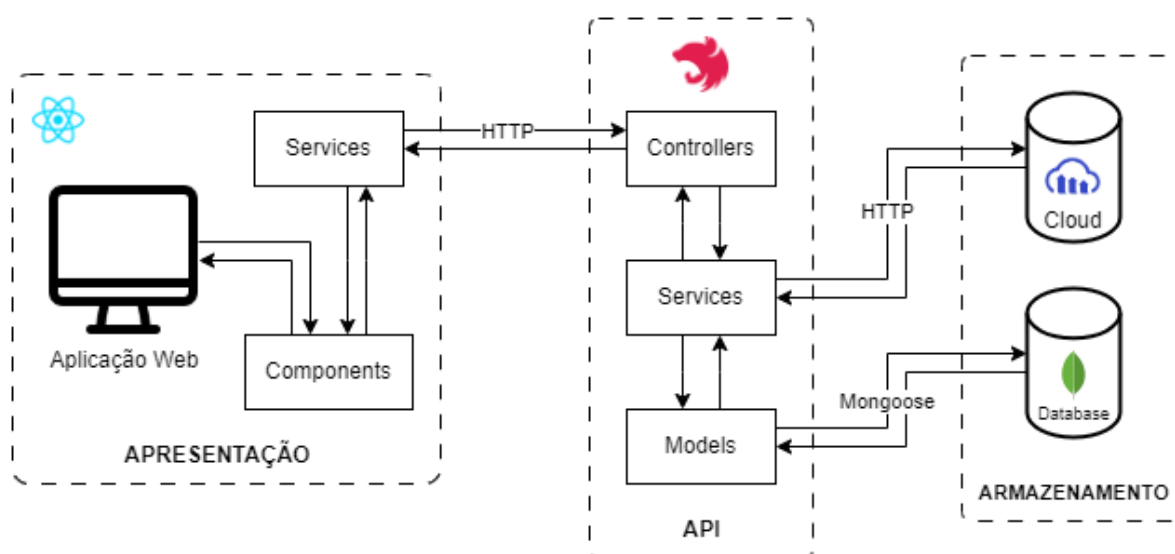
- **Visitante:** usuário anônimo. Este possui acesso a ver os cursos, as aulas e as postagens e comentários no fórum, porém, não pode criar postagens ou comentários;
- **Estudante:** usuário registrado na plataforma. Este possui todas as permissões de um usuário comum, além de poder criar e gerenciar suas próprias postagens e comentários no fórum;
- **Administrador:** usuário responsável pela administração da plataforma. Este possui todas as permissões de um estudante, além de poder gerenciar cursos e aulas na plataforma e ter a permissão de excluir e marcar postagens como “duplicadas” e excluir comentários de outros usuários no fórum.

Para facilitar a leitura do diagrama apresentado na Figura 4, os casos de uso que seriam apresentados com as funcionalidades de criar, editar e excluir os dados referenciados nas ações foram resumidos utilizando o verbo “gerenciar”.

3.2 Arquitetura

Esta seção contém a arquitetura de *software* utilizada na construção da aplicação proposta neste trabalho. A aplicação foi construída com base no padrão *REST (Representational State Transfer)* para fluxo de dados e consiste em três camadas: Apresentação, API e Armazenamento, sendo todas igualmente importantes para o funcionamento da plataforma. A Figura 5 contém uma representação conceitual das camadas descritas acima.

Figura 5 - Representação do modelo conceitual do sistema



Fonte: Elaborado pelo autor, 2025

A camada de *apresentação* consiste na interface responsável por garantir a interação do usuário com o sistema através de uma aplicação *web* construída inteiramente utilizando a biblioteca *React*. Nela, os componentes são unidos e transformados em um documento *HTML* para ser renderizado na tela do navegador. Os componentes, por sua vez, por meio das funções disponíveis no módulo *Services* (serviços), fazem requisições à *API* para que sejam recebidos os dados a serem apresentados ao usuário ou para que os dados inseridos pelo usuário sejam enviados ao sistema.

Na camada da *API* estão as funções responsáveis por tratar os dados enviados pela camada de *apresentação* e enviar requisições à camada de *armazenamento*. Construída utilizando o *framework NestJS*, a camada utiliza a biblioteca *Express* para construir as rotas da *API*, as bibliotecas *JWT (JSON Web Token)*⁴ e *Bcrypt*⁵ para segurança e autenticação dos dados dos usuários, e a biblioteca *Mongoose*⁶ para fazer consultas ao banco de dados presente na camada de *armazenamento*.

O módulo *Controllers* (controladores) consiste nas rotas da aplicação. Nele as requisições *HTTP* enviadas pela camada de *apresentação* são computadas e seus dados são tratados para que sejam enviadas as respectivas respostas. Essas requisições, caso não apresentem algum erro, são enviadas para para o módulo *Services* (serviços) para que sejam tratados, caso necessário, aplicadas as regras de negócio referentes à *API* e então enviados para a camada de *armazenamento* por meio do módulo *Models* (modelos). Arquivos de imagens enviados pela camada de *apresentação* são repassados pelo módulo *Services* diretamente para a camada de *armazenamento* por meio de requisições *HTTP* e também recebidas pelo mesmo meio.

Finalmente, a camada de *armazenamento* é responsável por armazenar os dados enviados pelos usuários e tratados na camada da *API* além de prover os dados solicitados nas consultas. A camada utiliza a base de dados *MongoDB* para armazenar dados e a plataforma *Cloudinary*⁷ para armazenar imagens em nuvem.

3.2.1 Modelagem do Banco de Dados

Esta subseção contém a descrição da modelagem do banco de dados da aplicação. O banco de dados escolhido para a aplicação foi o *MongoDB*, por sua versatilidade e capacidade de armazenamento de grande quantidade de dados, assim como descrito no capítulo referente à fundamentação teórica do projeto. Na Figura 6, é possível observar o diagrama lógico utilizado para a modelagem dos dados da aplicação, contendo as coleções nas quais os dados estão armazenados e suas respectivas chaves.

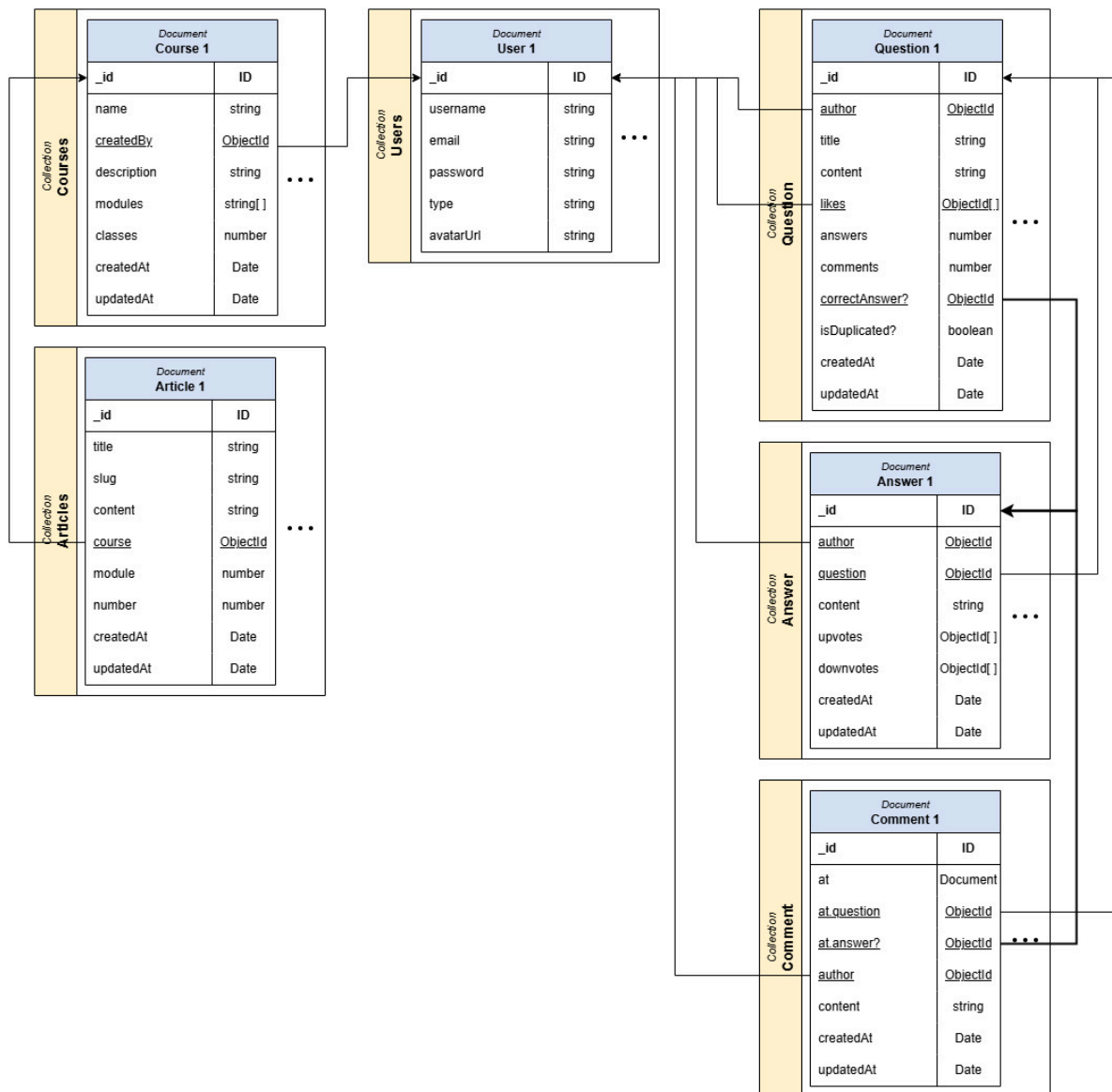
⁴ <https://jwt.io>

⁵ <https://www.npmjs.com/package/bcrypt>

⁶ <https://mongoosejs.com>

⁷ <https://cloudinary.com>

Figura 6 - Diagrama de modelo lógico de dados



Fonte: Elaborado pelo autor, 2025

A coleção *Users* armazena os dados dos usuários cadastrados na aplicação e possui as seguintes chaves: *username* (nome do usuário), *email*, *password* (senha), *type* – que indica se o usuário é um estudante ou administrador – e *avatarUrl* – que armazena a *URL* da foto de perfil do usuário. Por motivos de segurança, as senhas dos usuários são criptografadas em *hash* antes de serem enviadas ao banco de dados.

A coleção *Courses* armazena os dados relativos aos cursos da plataforma. Os documentos desta coleção apresentam as chaves *name* (nome), *createdBy*

(criado por) – que armazena o ID do usuário que cadastrou o curso na plataforma –, *description* (descrição), *modules* (módulos) – lista com os nomes dos módulos das aulas – e *classes* (aulas) – que indica o número de aulas presentes no curso. Já a coleção *Article* representa as aulas presentes nos cursos, que estão identificadas como “artigos”. Os documentos presentes nesta coleção contêm as chaves *title* (título), *slug* – que armazena uma versão editada do título da aula, composta apenas por caracteres alfanuméricos, sem letras maiúsculas ou acentuadas e com hífens no substituindo os espaços vazios –, *content* (conteúdo), *course* (curso) – que indica o ID do curso ao qual a aula pertence –, *module* (módulo) e *number* (número) – indicando a posição que a aula ocupa na ordem das aulas do curso.

Os documentos da coleção *Question* armazenam as postagens feitas no fórum e contêm as seguintes chaves: *author* (autor) – que guarda o ID do usuário que enviou a pergunta –, *title* (título), *content* (conteúdo), *likes* – que armazena uma lista contendo os IDs dos usuários que marcaram a postagem como “gostei” –, *answers* (respostas) – que indica o número de respostas enviadas para a pergunta –, *comments* (comentários) – que indica o número de comentários presentes –, *correctAnswer* (resposta correta) – chave opcional que armazena o ID da resposta que foi marcada como a mais correta para a pergunta em questão – e *isDuplicated* (está duplicada) – que indica se há uma pergunta parecida já postada anteriormente.

A coleção *Answers* armazena os dados das respostas feitas às perguntas no fórum e apresenta as chaves *author* (autor), *question* (pergunta) – que armazena o ID da pergunta à qual essa resposta foi vinculada –, *content* (conteúdo) e *upvotes* e *downvotes* – listas que armazenamos IDs dos usuários que marcaram a resposta como, respectivamente, “relevante” e “não relevante”. E, por fim, a coleção *Comments* contém os comentários feitos tanto às perguntas postadas no fórum quanto às respostas. Os documentos desta coleção têm as seguintes chaves: *at*, que guarda um documento contendo as chaves *question* (pergunta) e *answer* (resposta) – sendo esta última uma chave opcional que contém o ID da resposta caso o comentário tenha sido feito à uma resposta –, *author* (autor) e *content* (conteúdo).

Além das chaves apresentadas acima, todos os documentos nestas coleções contêm a chave “*_id*”, que é usada como identificador do documento e, com exceção da coleção *Users*, as chaves *createdAt* (criado em) e *updatedAt* (atualizado em), que guardam, respectivamente, a data de criação e a data da última edição do documento.

3.3 Tecnologias Utilizadas

Nesta seção, serão apresentadas as tecnologias utilizadas no desenvolvimento da aplicação descrita neste documento. Tanto a aplicação web quanto a *API* foram construídas utilizando a linguagem de programação *Typescript*⁸ e o ambiente de desenvolvimento *Node.js*⁹. Para a construção da interface da aplicação web, foram utilizadas as bibliotecas *React* e *Styled Components*¹⁰, esta sendo usada para estilização dos componentes da aplicação. Os dados do usuário persistem no navegador por meio do *Local Storage* (armazenamento local) e as requisições à *API* são feitas utilizando a biblioteca *Axios*¹¹. A fim de possibilitar uma escrita mais dinâmica das aulas, perguntas e respostas, foi decidido que essas seriam escritas utilizando *markdown*¹², uma linguagem voltada para formatação de textos. O *front-end* usa as bibliotecas *uiw/react-md-editor*¹³ para que o usuário escreva o texto e *react-markdown*¹⁴ para exibir o conteúdo *markdown* na tela.

Já para o desenvolvimento da *API* foi utilizado o *NestJS*. Segundo a página de introdução da sua documentação, *Nest* (*NestJS*, 2025) é um *framework* para construção de aplicações *Node.js* do lado do servidor eficientes e escaláveis, utiliza *JavaScript* progressivo e é construída em *Typescript*, também tendo completo suporte à linguagem. O *Nest* combina elementos de POO (Programação Orientada a Objetos), Programação Funcional e PRF (Programação Reativa Funcional) e faz uso de *frameworks* de servidor *HTTP* robustos como o *Express*.

Com o intuito de garantir a segurança dos dados tanto da aplicação quanto dos usuários, foram utilizadas as bibliotecas *JWT* (*JSON Web Token*), para validar o

⁸ <https://www.typescriptlang.org>

⁹ <https://nodejs.org/pt>

¹⁰ <https://styled-components.com>

¹¹ <https://axios-http.com/ptbr/docs/intro>

¹² <https://tecnoblog.net/responde/o-que-e-markdown/>

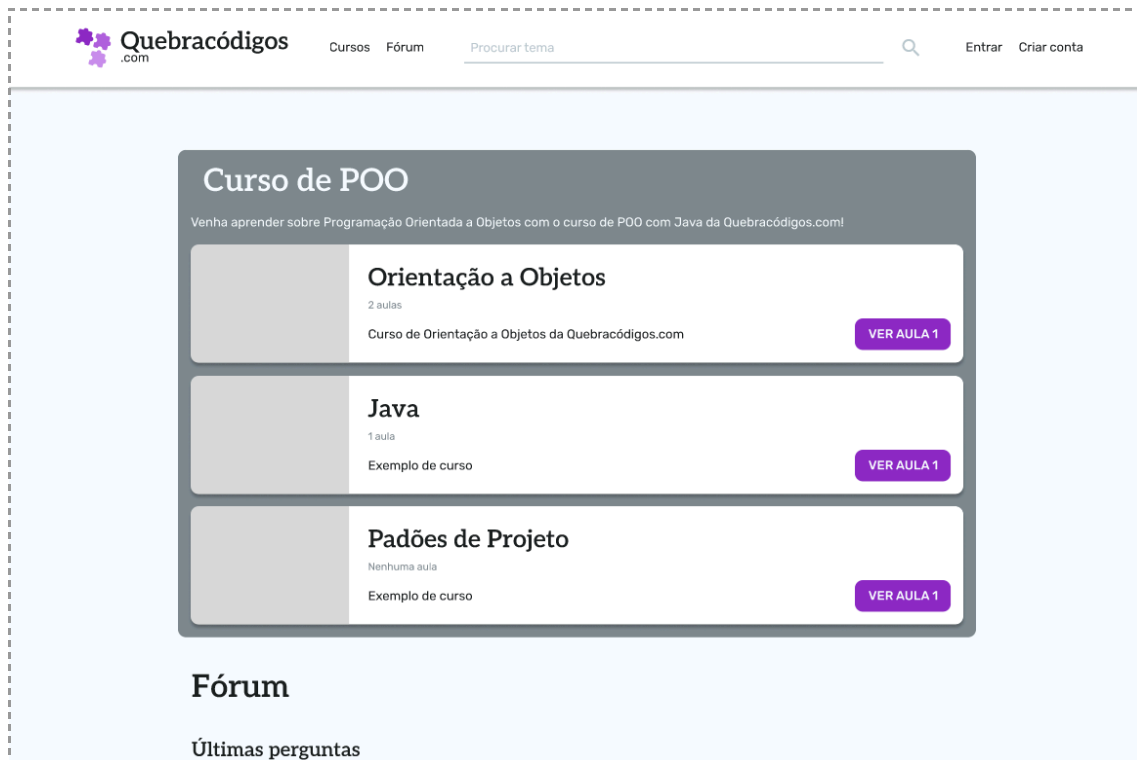
¹³ <https://www.npmjs.com/package/@uiw/react-md-editor>

¹⁴ <https://www.npmjs.com/package/react-markdown>

login, e *BCrypt*, para encriptar as senhas dos usuários antes de enviá-las para o banco de dados e comparar a senha enviada no momento do *login* e a senha armazenada para aquele usuário. Para conectar a *API* à base de dados, foi escolhida a biblioteca *Mongoose*, que possui integração com o *NestJS* e contém as funções necessárias para as consultas no banco de dados *MongoDB*. Em relação ao envio de imagens, foi escolhida a plataforma *Cloudinary* para armazenamento dos arquivos e usada a biblioteca de mesmo nome para gerenciar o envio e a obtenção dos mesmos.

3.4 Páginas da Aplicação

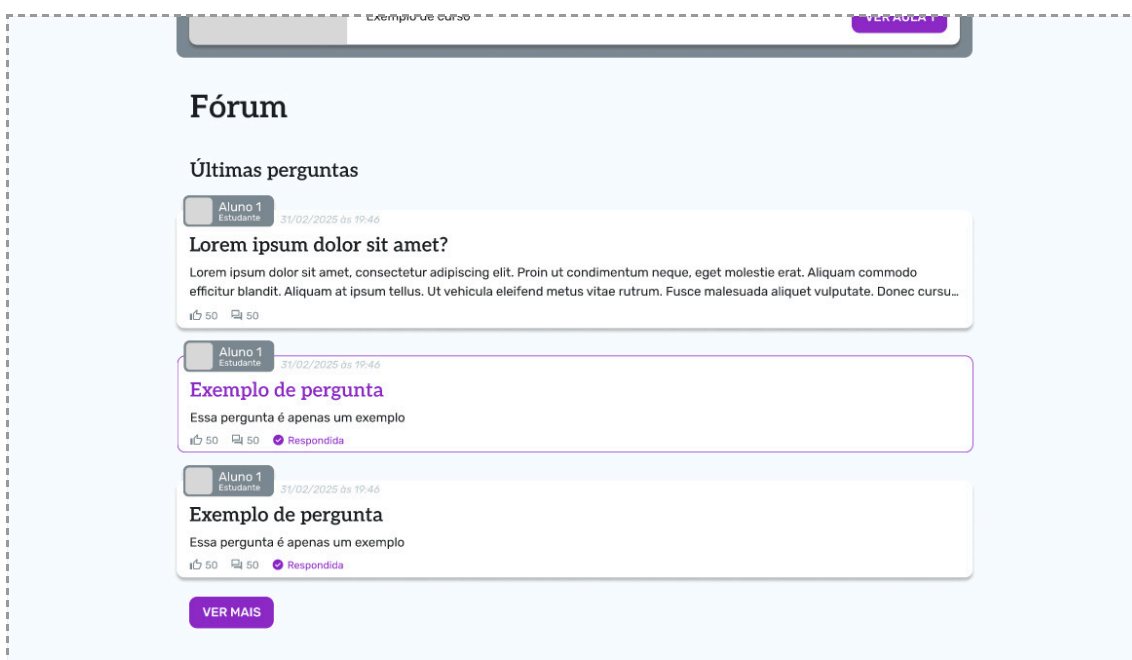
Nesta seção serão mostradas as páginas presentes na aplicação proposta neste trabalho, juntamente com a descrição das funcionalidades de cada página. Assim que o usuário entra na plataforma, é exibida a página principal. Nela é possível visualizar a seção de cursos, onde é possível clicar em algum dos cursos disponíveis, levando para a página do curso, ou no botão “Ver aula 1”, levando para a página da primeira aula do curso. Além disso, é possível ver, na parte superior de cada página, o cabeçalho da plataforma, onde o usuário tem acesso à página principal – clicando na logo do site – e às páginas de cursos, fórum, login, e criação de conta, além de um campo onde é possível pesquisar temas pelo título. A figura 7 mostra uma parte da página principal.

Figura 7 - Primeira parte da página principal

Fonte: Elaborado pelo autor, 2025

A Figura 8 mostra a segunda parte da página principal, onde o usuário pode ver as últimas perguntas postadas no fórum. Ao clicar em alguma pergunta, o usuário é redirecionado para a página daquela pergunta, e, ao clicar no botão “Ver mais”, é levado para a página do fórum.

Figura 8 - Segunda parte da página principal



Fonte: Elaborado pelo autor, 2025

Na figura 9, é possível ver a página de login, que pode ser acessada clicando no botão “Entrar” disponível no cabeçalho do site. É nela que o usuário pode inserir os dados da sua conta para entrar na plataforma ou se dirigir para a página de criação de conta.

Figura 9 - Página de login



Fonte: Elaborado pelo autor, 2025

Na Figura 10, é possível visualizar a página de cadastro, onde o usuário irá inserir os dados para criar sua conta ou clicar em “Clique aqui” para ir para a página de login caso já possua uma conta cadastrada na plataforma.

Figura 10 - Página de cadastro

Criar conta

Nome de usuário *

E-mail *

Senha *

Confirmar senha *

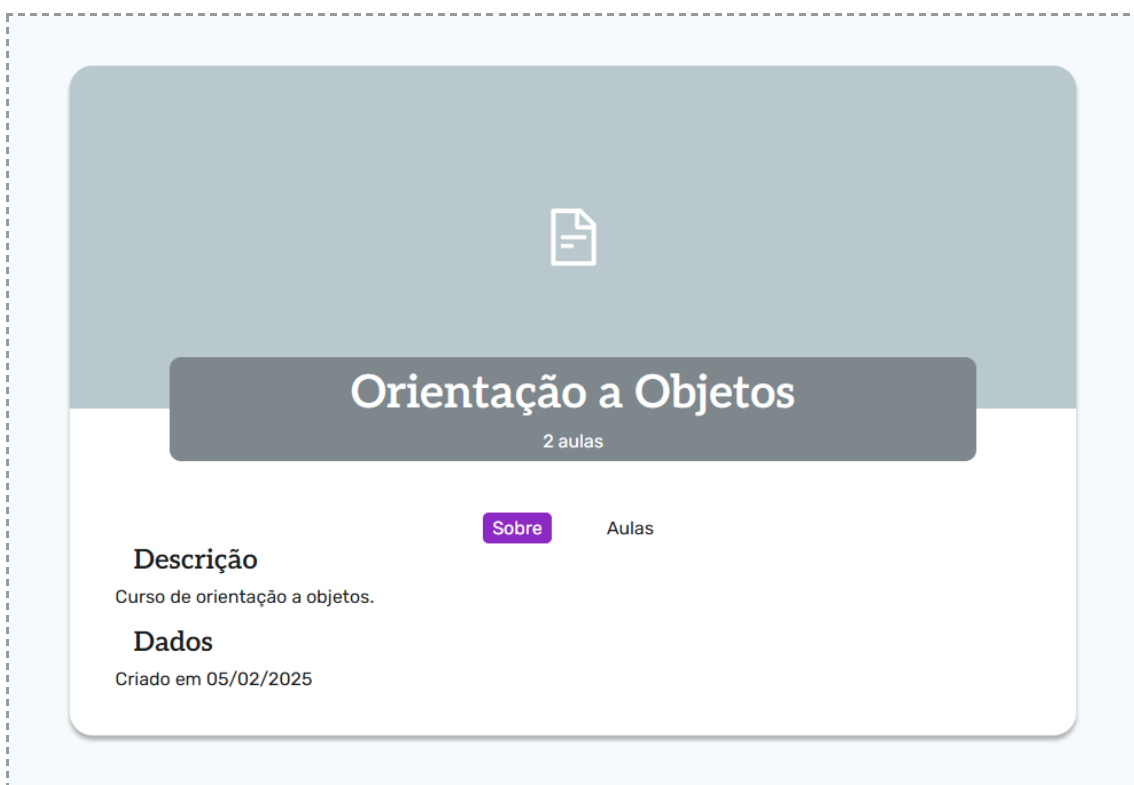
CRIAR CONTA

Já tem uma conta? [Clique aqui](#) para entrar!

Fonte: Elaborado pelo autor, 2025

A Figura 11 mostra a página de um curso, com a aba “Sobre” aberta. Nela o usuário pode ver algumas informações sobre curso ou clicar no botão “Aulas” para abrir a aba de aulas. Caso o usuário seja o criador daquele curso, ele terá acesso ao botão “Editar informações”, que abrirá uma página onde o usuário poderá alterar as informações do curso.

Figura 11 - Página de curso na aba “sobre”



Fonte: Elaborado pelo autor, 2025

A Figura 12 mostra a página de um curso, com a aba “Aulas” aberta. Nela o usuário pode ver algumas informações sobre curso ou clicar no botão “Sobre” para retornar à aba das informações do curso. Caso o usuário seja o criador daquele curso, o botão “Adicionar aula” será visível, permitindo com que o usuário crie uma aula naquele curso.

Figura 12 - Página de curso na aba “aulas”

Fonte: Elaborado pelo autor, 2025

A figura 13 exibe a página de uma aula. Nela, o usuário pode ver o conteúdo mostrado naquela aula. Caso a aula não seja a primeira do curso, é possível ver, no final, um botão que leva à aula anterior, e, caso não seja a última, um botão para seguir para a próxima aula.

Figura 13 - Página de aula

Introdução à POO

Olá! Seja bem-vindo ao nosso curso de Programação Orientada a Objetos.

Nesse módulo, você irá aprender os conceitos que envolvem a Programação Orientada a Objetos (POO), um paradigma da programação criado para facilitar a organização e a legibilidade do código, bem como. No entanto, como dizem as más línguas, vamos começar pelo começo:

O que é um "paradigma da programação"?

Um **paradigma da programação** é um conjunto de regras e convenções criadas com o intuito de guiar o desenvolvedor na resolução de problemas e na escrita do código. Cada linguagem de programação é criada com base nessas regras, e cada uma possui seu conjunto de paradigmas que os desenvolvedores seguirão ao escrever o código. Os paradigmas mais utilizados são a Programação Funcional e a Programação Orientada a Objetos. Vamos entender sobre cada uma delas.

Programação Funcional x Orientação a Objetos

A **Programação Funcional** se baseia no uso de funções como as peças fundamentais do código. Ela permite que as funções sejam tratadas de maneira flexível, como blocos de construção que podem ser usados, passados e armazenados em diferentes partes do programa. Sua principal característica é a imutabilidade dos dados, ou seja, os dados, uma vez que são criados, não poderão ser alterados. Isso ajuda a evitar erros e efeitos colaterais nos códigos. Além disso, a programação funcional utiliza a recursividade para abordar problemas de maneira direta e concisa. Esse paradigma, derivado da programação estruturada, possui uma estrutura linear, onde o código segue instruções especificadas dentro das funções.

Já a **Programação Orientada a Objetos**, assim como o nome sugere, é baseada no conceito de "objetos". Esses objetos podem ser abstrações de objetos existentes na vida real ou apenas ferramentas com funcionalidades que ajudam na execução do código. O intuito da sua criação, além de ser uma alternativa aos conceitos da programação estruturada, foi de aproximar a execução do programa ao manuseio das coisas na vida real. Nesse curso, focaremos nos conceitos do paradigma de Orientação a Objetos, bem como os assuntos que envolvem o tema.

Referências:

<https://tripleten.com.br/blog/paradigmas-de-programacao-o-que-sao-e-quais-os-principais/>
<https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos>
<https://www.davemorris.com.br/programacao-orientada-a-objetos-a-programacao-estruturada/72917>

Fonte: Elaborado pelo autor, 2025

4 CONSIDERAÇÕES FINAIS

Diante da relativa complexidade dos temas relacionados à Orientação a Objetos e à linguagem de programação Java, o “Quebracódigos.com” é uma plataforma que visa facilitar o aprendizado nessa área apresentando aulas especializadas e também possibilitando que os usuários enviem suas dúvidas e troquem conhecimentos. Além disso, é possível ver as aulas a qualquer momento, já que não é necessário o login para que as mesmas possam ser acessadas. Desse modo, a plataforma possui um ambiente onde o usuário pode aprender sobre o assunto, também, auxiliar outros usuários compartilhando o que aprendeu.

As principais dificuldades encontradas na elaboração deste projeto foram a pouca familiaridade do autor com certas funcionalidades importantes para o sistema e a elaboração das aulas presentes na plataforma. Contudo, com o tempo, tais problemas foram sendo superados e conseguiu-se fazer a plataforma descrita neste documento funcionar da forma desejada.

Devido à complexidade e ao tempo necessário, algumas ideias propostas no início da elaboração do projeto tiveram que ser descartadas. Uma delas era a possibilidade de usuários selecionados enviarem artigos separados das aulas sobre temas relacionados ao conteúdo proposto na plataforma, como notícias, tecnologias, entre outros. Essa ideia pode ser considerada para trabalhos futuros, bem como, por exemplo, a introdução de cursos sobre outras linguagens de programação orientadas a objetos e possíveis melhorias na interface do site.

REFERÊNCIAS

AGGARWAL, Sanchit et al. Modern web-development using reactjs. **International Journal of Recent Research Aspects**, v. 5, n. 1, p. 133-137, 2018.

AWS. O que é uma API (interface de programação de aplicações)? [s.d.]. Disponível em <<https://aws.amazon.com/pt/what-is/api>>. Acesso em 26 fev. 2025.

AWS. O que é uma API RESTful? [s.d.]. Disponível em <<https://aws.amazon.com/pt/what-is/restful-api>>. Acesso em 1 mar. 2025.

CHANDRA, Deka Ganesh. BASE analysis of NoSQL database. **Future Generation Computer Systems**, v. 52, p. 13-21, 2015.

COCKBURN, Alistair. Escrevendo Casos de Usos Eficazes: Um guia prático para desenvolvedores de software. **Bookman Editora**, 2005.

DEITEL, Paul; DEITEL, Harvey. Java: como programar. 10. ed. **Pearson Education**, 2017. p. 43.

FARINELLI, Fernanda. Conceitos básicos de programação orientada a objetos. **Instituto Federal Sudeste de Minas Gerais**, 2007.

FIELDING, Roy Thomas. Architectural styles and the design of network-based software architectures. **University of California, Irvine**, 2000.

GACKENHEIMER, Cory. Introduction to React. **Apress**, 2015.

GOUGH, James; BRYANT, Daniel; AUBURN, Matthew. Mastering API Architecture: Design, Operate, and Evolve API-Based Systems. “**O’Reilly Media, Inc.**”, 2021.

HENRIQUE, João. POO: o que é programação orientada a objetos? **Alura**, 2023. Disponível em: <<https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos>>. Acesso em 29 nov. 2023.

HOWS, David; MEMBREY, Peter; PLUGGE, Eelco. Introdução ao MongoDB. **Novatec Editora**, 2019.

JUNGTHON, Gustavo; GOULART, Cristian Machado. Paradigmas de programação. Monografia (Monografia) — **Faculdade de Informática de Taquara**, Rio Grande do Sul, v. 57, 2009.

LOUZADA, Vinícius; CARVALHO, Caroline; LARANJA, Emerson. API: o que é, para quê serve e qual é a sua importância. **Alura**, 2024. Disponível em <<https://www.alura.com.br/artigos/api>>. Acesso em 26 fev. 2025.

MARTIN, Robert C.. Agile Software Development: Principles, Patterns, and Practices. Reino Unido, **Pearson Education**, 2003.

MENDONÇA, Ricardo Augusto Ribeiro de. Levantamento de requisitos no desenvolvimento ágil de software. **Semana da Ciência e Tecnologia da PUC Goiás**, v. 12, 2014.

MONGODB. O que é NoSQL? Bancos De Dados NoSql Explicados. [s.d.]. Disponível em <<https://www.mongodb.com/pt-br/resources/basics/databases/nosql-explained>>. Acesso em 05 mar. 2025.

NAYAK, Ameya; PORIYA, Anil; POOJARY, Dikshay. Type of NOSQL databases and its comparison with relational databases. **International Journal of Applied Information Systems**, v. 5, n. 4, p. 16-19, 2013.

NESTJS. NestJs. [s.d.] Disponível em <<https://docs.nestjs.com>>. Acesso em 10 mar. 2025.

NEVES, Vinícius. React: o que é, como funciona e um guia dessa popular ferramenta JS. **Alura**, 2023. Disponível em <<https://www.alura.com.br/artigos/react-js>>. Acesso em 06 mar. 2025.

OLIVEIRA, Samuel Silva de. Bancos de dados Não-Relacionais: um novo paradigma para armazenamento de dados em sistemas de ensino colaborativo. **Revista da Escola de Administração Pública do Amapá**, v. 2, n. 1, p. 184-194, 2014.

POO, Danny; KLONG, Derek; ASHOK, Swarnalatha. Object-oriented programming and Java. **Springer Science & Business Media**, 2007.

RICARTE, Ivan Luiz Marques. Programação Orientada a Objetos: uma abordagem com Java. Disponível em <<http://www.dca.fee.unicamp.br/cursos/Poo.Java/Aulas/poojava.pdf>>. Acesso em 08 mar. 2025, v. 29, n. 10, p. 2014, 2001.

STACK OVERFLOW. Stack Overflow, c2025. Página inicial. Disponível em <<https://stackoverflow.com/questions>>. Acesso em 26 fev. 2025.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira. Engenharia de Requisitos: software orientado ao negócio. **Brasport**, 2016.

W3SCHOOLS. W3Schools, c2025. Disponível em <<https://w3schools.com>>. Acesso em 25 fev. 2025.

APÊNDICE A - CASOS DE USO

Quadro 5 - Caso de Uso: Ver cursos

UC-01: Ver cursos	
Ator principal	Usuário (todos)
Atores secundários	Nenhum
Resumo	O usuário visualiza os cursos da plataforma
Pré-condições	Nenhuma
Pós-condições	Nenhuma
Ações	
1. O sistema direciona o usuário para a página principal; 2. O usuário clica em um dos cursos presentes; 3. O sistema redireciona o usuário para a página contendo as informações do curso.	

Fonte: Elaborado pelo autor (2025)

Quadro 6 - Caso de Uso: Ver aulas

UC-02: Ver aulas	
Ator principal	Usuário (todos)
Atores secundários	Nenhum
Resumo	O usuário visualiza as aulas de um curso da plataforma
Pré-condições	Estar na página de um curso
Pós-condições	Nenhuma
Ações	
1. O usuário clica no botão “Aulas”; 2. O sistema abre a aba de aulas do curso; 3. O usuário clica em uma das aulas; 4. O sistema redireciona o usuário para a página da aula.	
Ações secundárias	
5. a) A aula está em um módulo fechado do curso: 1. O usuário clica no módulo ao qual a aula pertence; 2. O sistema abre o módulo revelando as aulas presentes; 3. Avança ao passo 6.	

Fonte: Elaborado pelo autor (2025)

Quadro 7 - Caso de Uso: Ver postagens

UC-03: Ver postagens	
Ator principal	Usuário (todos)
Atores secundários	Nenhum
Resumo	O usuário visualiza as perguntas no fórum e suas respostas
Pré-condições	Nenhuma
Pós-condições	Nenhuma
Ações	
1. O sistema direciona o usuário para a página principal; 2. O usuário clica em uma das perguntas presentes; 3. O sistema redireciona o usuário para a página da pergunta.	
Ações Secundárias	
1. a) Acesso à página do fórum: 1. O usuário clica no botão “Ver mais”; 2. O sistema redireciona o usuário para a página do fórum; 3. Avança ao passo 2;	

Fonte: Elaborado pelo autor (2025)

Quadro 8 - Caso de Uso: Ver comentários

UC-04: Ver comentários	
Ator principal	Usuário (todos)
Atores secundários	Nenhum
Resumo	O usuário visualiza os comentários presentes na pergunta do fórum ou em uma de suas respostas
Pré-condições	Estar na página de uma postagem no fórum
Pós-condições	Nenhuma
Ações	
1. O usuário clica no ícone de comentários na postagem ou em uma resposta; 2. O sistema abre os comentários presentes.	
Ações Secundárias	
1. a) Nenhum comentário: 1. O sistema informa: “Faça login em sua conta para enviar um comentário”;	

Fonte: Elaborado pelo autor (2025)

Quadro 9 - Caso de Uso: Fazer login

UC-05: Fazer login	
Ator principal	Visitante
Atores secundários	Nenhum
Resumo	O usuário entra em sua conta na plataforma
Pré-condições	Ter uma conta cadastrada no sistema
Pós-condições	Usuário está logado no sistema
Ações	
1. O sistema direciona o usuário para a página de login; 2. O usuário insere e-mail e senha; 3. O usuário clica em “Entrar”; 4. O sistema redireciona o usuário para a página principal.	
Ações Secundárias	
3. a) E-mail ou senha incorretos: 1. O sistema informa: “E-mail ou senha incorretos”; 2. Retorna ao passo 2.	

Fonte: Elaborado pelo autor (2025)

Quadro 10 - Caso de Uso: Criar conta

UC-06: Criar conta	
Atores principais	Visitante
Atores secundários	Nenhum
Resumo	O usuário cadastra sua conta no sistema
Pré-condições	Nenhuma
Pós-condições	Nenhuma
Ações	
1. O sistema direciona o usuário para a página de criar conta; 2. O usuário insere nome, e-mail e senha; 3. O usuário clica em “Salvar”; 4. O sistema informa: “Dados cadastrados com sucesso.”; 5. O usuário clica em “Ok”; 6. O sistema redireciona o usuário para a página principal.	
Ações Secundárias	
3. a) E-mail inválido: 1. O sistema informa: “E-mail inválido”; 2. Retorna ao passo 2. b) Campo “Repetir senha” diferente do campo “Senha”: 1. O sistema informa: “As senhas não coincidem”; 2. Retorna ao passo 2.	

Fonte: Elaborado pelo autor (2025)

Quadro 11 - Caso de Uso: Criar pergunta

UC-07: Criar pergunta	
Ator principal	Estudante, Administrador
Atores secundários	Nenhum
Resumo	O usuário cria uma pergunta no fórum
Pré-condições	Estar logado no sistema, estar na página do fórum
Pós-condições	Nenhuma
Ações	
1. O usuário clica em “Enviar pergunta”; 2. O sistema abre o modal de criação de pergunta; 3. O usuário insere título e conteúdo da pergunta; 4. O usuário clica em “Enviar”; 5. O sistema informa: “Pergunta postada com sucesso”.	
Ações Secundárias	
4. a) Há uma postagem na plataforma de título exatamente igual ao inserido pelo usuário: 1. O sistema informa “Já existe uma pergunta com esse título”; 2. Retorna ao passo 3;	

Fonte: Elaborado pelo autor (2025)

Quadro 12 - Caso de Uso: Editar pergunta

UC-08: Editar postagem	
Ator principal	Estudante, Administrador
Atores secundários	Nenhum
Resumo	O usuário edita uma postagem feita por ele mesmo na plataforma.
Pré-condições	Estar logado no sistema, estar na página de uma postagem feita pelo usuário.
Pós-condições	Nenhuma
Ações	
1. O usuário clica no ícone de editar pergunta; 2. O sistema abre o modal de edição de pergunta; 3. O usuário edita as informações presentes; 4. O usuário clica em “Salvar”; 5. O sistema informa: “Pergunta editada com sucesso”.	

Fonte: Elaborado pelo autor (2025)

Quadro 13 - Caso de Uso: Excluir pergunta

UC-09: Excluir postagem	
Ator principal	Estudante, Administrador
Atores secundários	Nenhum
Resumo	O usuário exclui uma pergunta feita por ele mesmo no fórum. Caso o usuário seja um administrador, é possível excluir qualquer pergunta.
Pré-condições	Estar logado no sistema, estar na página de uma postagem feita pelo usuário ou ser administrador
Pós-condições	Nenhuma
Ações	
1. O usuário clica no ícone de excluir pergunta; 2. O sistema informa: “Deseja mesmo excluir a pergunta?”; 3. O usuário clica em “Excluir”; 4. O sistema informa: “Pergunta excluída com sucesso”.	

Fonte: Elaborado pelo autor (2025)

Quadro 14 - Caso de Uso: Criar resposta

UC-10: Criar resposta	
Ator principal	Estudante, Administrador
Atores secundários	Nenhum
Resumo	O usuário cria uma resposta para uma pergunta no fórum
Pré-condições	Estar logado no sistema, estar na página de uma pergunta no fórum
Pós-condições	Nenhuma
Ações	
1. O usuário clica no botão “Enviar resposta”; 2. O sistema abre o modal de envio de resposta; 3. O usuário insere a resposta; 4. O usuário clica em “Enviar”; 5. O sistema informa: “Resposta enviada com sucesso”.	

Fonte: Elaborado pelo autor (2025)

Quadro 15 - Caso de Uso: Editar resposta

UC-11: Editar resposta	
Ator principal	Estudante, Administrador
Atores secundários	Nenhum
Resumo	O usuário edita uma resposta feita por ele mesmo a uma pergunta no fórum
Pré-condições	Estar logado no sistema, estar na página de uma pergunta com uma resposta do usuário no fórum
Pós-condições	Nenhuma
Ações	
1. O usuário clica no ícone de edição de resposta; 2. O sistema abre o modal de edição de resposta; 3. O usuário edita a resposta; 4. O usuário clica em “Salvar”; 5. O sistema informa: “Resposta editada com sucesso”.	

Fonte: Elaborado pelo autor (2025)

Quadro 16 - Caso de Uso: Excluir resposta

UC-12: Excluir resposta	
Ator principal	Estudante, Administrador
Atores secundários	Nenhum
Resumo	O usuário exclui uma resposta feita por ele mesmo a uma pergunta no fórum. Caso o usuário seja administrador, é possível excluir qualquer resposta.
Pré-condições	Estar logado no sistema, estar na página de uma pergunta com uma resposta do usuário no fórum ou ser administrador
Pós-condições	Nenhuma
Ações	
1. O usuário clica no ícone de excluir resposta; 2. O sistema informa “Deseja mesmo excluir a resposta?”; 3. O usuário clica em “Excluir”; 4. O sistema informa: “Resposta excluída com sucesso”.	

Fonte: Elaborado pelo autor (2025)

Quadro 17 - Caso de Uso: Criar comentário

UC-13: Criar comentário	
Ator principal	Estudante, Administrador
Atores secundários	Nenhum
Resumo	O usuário cria um comentário em uma pergunta ou resposta no fórum
Pré-condições	Estar logado no sistema, estar na página de uma pergunta no fórum
Pós-condições	Nenhuma
Ações	
1. O usuário clica no ícone de comentário na pergunta ou em uma resposta; 2. O sistema abre os comentários presentes na pergunta ou resposta; 3. O usuário clica no campo de enviar comentário; 4. O usuário escreve o comentário; 5. O usuário pressiona a tecla “Enter” ou clica no botão de enviar; 6. O sistema recarrega a página.	

Fonte: Elaborado pelo autor (2025)

Quadro 18 - Caso de Uso: Editar comentário

UC-14: Editar comentário	
Ator principal	Estudante, Administrador
Atores secundários	Nenhum
Resumo	O usuário edita um comentário feito por ele mesmo em uma pergunta ou resposta no fórum
Pré-condições	Estar logado no sistema, estar na página de uma pergunta contendo um comentário feito pelo usuário no fórum
Pós-condições	Nenhuma
Ações	
1. O usuário clica no ícone de editar comentário; 2. O usuário edita o comentário; 3. O usuário pressiona a tecla “Enter” ou clica no botão de enviar; 4. O sistema recarrega a página.	

Fonte: Elaborado pelo autor (2025)

Quadro 19 - Caso de Uso: Excluir comentário

UC-15: Excluir comentário	
Ator principal	Estudante, Administrador
Atores secundários	Nenhum
Resumo	O usuário exclui um comentário feito por ele mesmo em uma pergunta ou resposta no fórum. Caso o usuário seja administrador, é possível excluir qualquer comentário.
Pré-condições	Estar logado no sistema, estar na página de uma pergunta contendo comentário feito pelo usuário no fórum
Pós-condições	Nenhuma
Ações	
1. O usuário clica no ícone de excluir comentário; 2. O sistema informa: “Deseja mesmo excluir o comentário?”; 3. O usuário clica em “Excluir”; 4. O sistema informa: “Comentário excluído com sucesso”.	

Fonte: Elaborado pelo autor (2025)

Quadro 20 - Caso de Uso: Criar curso

UC-16: Criar curso	
Ator principal	Administrador
Atores secundários	Nenhum
Resumo	O usuário cria um curso na plataforma
Pré-condições	Estar logado no sistema
Pós-condições	Nenhuma
Ações	
1. O usuário clica em “Criar curso”; 2. O sistema redireciona o usuário para a página de criação de curso; 3. O usuário insere as informações do curso; 4. O usuário clica em “Criar”; 5. O sistema informa: “Curso criado com sucesso”; 6. O usuário clica em “Ok”; 7. O sistema redireciona o usuário para a página do curso criado.	
Ações Secundárias	
4. a) Há um curso de mesmo nome na plataforma: 1. O sistema informa “Já existe um curso com esse nome”; 2. Retorna ao passo 3;	

Fonte: Elaborado pelo autor (2025)

Quadro 21 - Caso de Uso: Editar curso

UC-17: Editar curso	
Ator principal	Administrador
Atores secundários	Nenhum
Resumo	O usuário edita as informações de um curso criado por ele mesmo na plataforma
Pré-condições	Estar logado no sistema, estar na página de um curso criado pelo usuário
Pós-condições	Nenhuma
Ações	
1. O usuário clica em “Editar informações”; 2. O sistema redireciona o usuário para a página de edição de curso; 3. O usuário edita as informações do curso; 4. O usuário clica em “Salvar”; 5. O sistema informa: “Dados editados com sucesso”; 6. O usuário clica em “Ok”; 7. O sistema redireciona o usuário de volta para a página do curso.	

Fonte: Elaborado pelo autor (2025)

Quadro 22 - Caso de Uso: Excluir curso

UC-18: Excluir curso	
Ator principal	Administrador
Atores secundários	Nenhum
Resumo	O usuário exclui um curso criado por ele na plataforma
Pré-condições	Estar logado no sistema, estar na página de um curso criado pelo usuário
Pós-condições	Nenhuma
Ações	
1. O usuário clica em “Excluir curso”; 2. O sistema informa: “Deseja mesmo excluir o curso?”; 3. O usuário clica em “Excluir”; 4. O sistema informa: “Curso excluído com sucesso”.	

Fonte: Elaborado pelo autor (2025)

Quadro 23 - Caso de Uso: Criar aula

UC-19: Criar aula	
Ator principal	Administrador
Atores secundários	Nenhum
Resumo	O usuário cria uma aula em um curso criado por ele na plataforma
Pré-condições	Estar logado no sistema, estar na página de um curso criado pelo usuário
Pós-condições	Nenhuma
Ações	
1. O usuário clica na aba “Aulas”; 2. O sistema abre a aba de aulas do curso; 3. O usuário clica no botão “Criar aula”; 4. O sistema redireciona o usuário para a página de criação de aula; 5. O usuário insere o título e o conteúdo da aula; 6. O usuário clica em “Criar aula”; 7. O sistema informa: “Aula criada com sucesso”; 8. O usuário clica em “Ok”; 9. O sistema redireciona o usuário de volta para a página do curso;	
Ações Secundárias	
6. a) Há uma aula de mesmo título na plataforma: 1. O sistema informa “Já existe uma aula com esse título”; 2. Retorna ao passo 5;	

Fonte: Elaborado pelo autor (2025)

Quadro 24 - Caso de Uso: Editar aula

UC-20: Editar curso	
Ator principal	Administrador
Atores secundários	Nenhum
Resumo	O usuário edita uma aula em um curso criado por ele na plataforma
Pré-condições	Estar logado no sistema, estar na página de um curso criado pelo usuário
Pós-condições	Nenhuma
Ações	
1. O usuário clica na aba “Aulas”; 2. O sistema abre a aba de aulas do curso; 3. O usuário clica no ícone de editar aula; 4. O sistema redireciona o usuário para a página de editar aula; 5. O usuário edita os dados da aula; 6. O usuário clica em “Salvar”;	

7. O sistema informa: “Aula editada com sucesso”; 8. O usuário clica em “Ok”; 9. O sistema redireciona o usuário para a página do curso criado.
Ações Secundárias
10. a) A aula está em um módulo fechado do curso: 1. O usuário clica no módulo ao qual a aula pertence; 2. O sistema abre o módulo revelando a aula desejada; 3. Avança ao passo 3.

Fonte: Elaborado pelo autor (2025)

Quadro 25 - Caso de Uso: Excluir aula

UC-21: Excluir aula	
Ator principal	Administrador
Atores secundários	Nenhum
Resumo	O usuário exclui uma aula em um curso criado por ele na plataforma
Pré-condições	Estar logado no sistema, estar na página de um curso criado pelo usuário
Pós-condições	Nenhuma
Ações	
1. O usuário clica na aba “Aulas”; 2. O sistema abre a aba de aulas do curso; 3. O usuário clica no ícone de excluir aula; 4. O sistema informa: “Deseja mesmo excluir a aula”; 5. O usuário clica em “Excluir”; 6. O sistema informa: “Aula excluída com sucesso”; 7. O usuário clica em “Ok”; 8. O sistema recarrega a página do curso.	
Ações Secundárias	
2. a) A aula está em um módulo fechado do curso: 1. O usuário clica no módulo ao qual a aula pertence; 2. O sistema abre o módulo revelando a aula desejada; 3. Avança ao passo 3.	

Fonte: Elaborado pelo autor (2025)

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Cajazeiras - Código INEP: 25008978
	Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
	CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Ostensivo (Público)

Trabalho de Conclusão de Curso

Assunto:	Trabalho de Conclusão de Curso
Assinado por:	Alvaro Frade
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Alvaro Renan Costa Frade, DISCENTE (202112010013) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS, em 31/03/2025 18:17:59.

Este documento foi armazenado no SUAP em 31/03/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1441938
Código de Autenticação: fba09efafd

