

Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Engenharia de Software

OJTracker: Evolução da aplicação com foco no suporte a técnicos

Autor: João Matheus de Sousa Rodrigues
Orientador: Dr. Edson Alves da Costa Júnior

Brasília, DF
2024



João Matheus de Sousa Rodrigues

OJTracker: Evolução da aplicação com foco no suporte a técnicos

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Dr. Edson Alves da Costa Júnior

Brasília, DF

2024

João Matheus de Sousa Rodrigues

OJTracker: Evolução da aplicação com foco no suporte a técnicos/ João Matheus de Sousa Rodrigues. – Brasília, DF, 2024-

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

Orientador: Dr. Edson Alves da Costa Júnior

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2024.

1. Criação de treinos integrados com juiz online. 2. Recomendação de problemas. I. Dr. Edson Alves da Costa Júnior. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. OJTracker: Evolução da aplicação com foco no suporte a técnicos

CDU 02:141:005.6

João Matheus de Sousa Rodrigues

OJTracker: Evolução da aplicação com foco no suporte a técnicos

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 09 de setembro de 2024:

Dr. Edson Alves da Costa Júnior
Orientador

Prof. Dr. Bruno Cesar Ribas
Convidado 1

**Prof. Dr. Vinícius Ruela Pereira
Borges**
Convidado 2

Brasília, DF
2024

Resumo

A programação competitiva é uma atividade crescente no Brasil, com diversas competições nacionais e internacionais. Para se preparar para essas competições, os participantes costumam treinar em juízes *online*, que fornecem uma variedade de problemas de diferentes níveis de dificuldade e tópicos. No entanto, a grande quantidade de problemas disponíveis pode dificultar a escolha dos mais adequados para o aprendizado. Por isso, uma ferramenta que possua métodos manuais e automatizados de recomendação de problemas nos principais juízes disponíveis é relevante na preparação desses competidores. Este trabalho tem como objetivo evoluir a aplicação OJTracker. A nova versão incluirá suporte aos técnicos de programação competitiva, permitindo que eles elaborem treinamentos e listas de problemas personalizados para seus alunos. Além disso, serão coletadas automaticamente algumas métricas de software para averiguar a funcionalidade e a usabilidade do que foi implementado. A diferenciação entre atletas e treinadores na plataforma foi estabelecida por meio da implementação de um serviço de autenticação.

Palavras-chave: juiz online, recomendação de problemas, programação competitiva.

Abstract

Competitive programming is a growing activity in Brazil, with various national and international competitions. To prepare for these competitions, participants often train on online judges that provide a variety of problems at different difficulty levels and topics. However, the abundance of available problems can make it challenging to choose the most suitable ones for learning. Therefore, a tool that incorporates both manual and automated methods for recommending problems on major online judges is relevant for the preparation of these competitors. This work aims to enhance the OJTracker application. The new version will include support for competitive programming coaches, enabling them to create custom training sessions and problem sets for their students. Additionally, some software metrics will be automatically collected to evaluate the functionality and usability of what has been implemented. The differentiation between contestants and coaches on the platform has been established through the implementation of an authentication service.

Key-words: online judges, problems recommendation, competitive programming.

Lista de ilustrações

Figura 1 – Visualização de Listas no C2Ladder.	24
Figura 2 – Informações de Usuário no C2Ladder.	24
Figura 3 – Visualização de Status no C2Ladder.	25
Figura 4 – Página Inicial do CP Academy.	26
Figura 5 – Seleção de Tópico no CP Academy.	26
Figura 6 – Listas de Problemas no CP Academy.	27
Figura 7 – Listas de Problemas no Virtual Judge.	28
Figura 8 – Seleção de OJ no Virtual Judge.	29
Figura 9 – Criação de Competição no Virtual Judge.	29
Figura 10 – Adição de Problema em Competição no Virtual Judge.	30
Figura 11 – Página Inicial do AtCoder.	34
Figura 12 – Página de Perfil de Usuário no AtCoder.	34
Figura 13 – Página de Ranking do AtCoder.	35
Figura 14 – Página de Contest do AtCoder.	36
Figura 15 – Página Inicial do Judge.	36
Figura 16 – Página de Problemas do Judge.	37
Figura 17 – Exemplos de Gráficos de Progresso do Judge.	38
Figura 18 – Nova Arquitetura de Software do OJTracker.	45
Figura 19 – Respostas dos Usuários sobre a Usabilidade da Primeira Versão.	48
Figura 20 – Tela Principal em Resolução Pequena na Primeira Versão.	48
Figura 21 – Menu e Drawer Temporário na Nova Versão.	49
Figura 22 – Modal de Informações de Usuário Responsiva em Desktop.	49
Figura 23 – Modal de Informações de Usuário Responsiva em Mobile.	50
Figura 24 – Tela de Login do OJTracker.	51
Figura 25 – Tela Principal com Alterações da Autenticação.	53
Figura 26 – Página de Listas Curadas.	53
Figura 27 – Adição de Nova Lista Curada.	54
Figura 28 – Página de Visualização de Lista Curada.	55
Figura 29 – Edição de Lista Curada.	55
Figura 30 – Adição de Problema na Lista Curada.	55
Figura 31 – Comparação de Tempos das Etapas de Adição de Problema.	56
Figura 32 – Visualização de Problemas.	56
Figura 33 – Página de Cadastro.	58
Figura 34 – Tela de Gerenciamento de Usuários.	58
Figura 35 – Tela de Treinamento de USER.	59
Figura 36 – Tela de Criação de Treinamento.	59

Figura 37 – Tela de Treinadores.	60
Figura 38 – Tela de Problemas do Treinamento.	60
Figura 39 – Dispositivos Utilizados	61
Figura 40 – Sistemas Operacionais Utilizados	62
Figura 41 – Ferramentas Utilizadas	62
Figura 42 – Páginas Acessadas	63

Lista de tabelas

Tabela 1 – Quantidade de Problemas de OJs em 26/11/2023.	32
------------------------------------------------------------------	----

Lista de Quadros

1	Retornos da Análise de uma Solução por um Juiz Eletrônico.	31
2	Funcionalidades do Judge.	37
3	Características Essenciais das Métricas de <i>Software</i>	40
4	Requisitos elicitados através do questionário.	42
5	Requisitos elicitados através do <i>brainstorm</i>	42
6	Detalhes da Pesquisa Bibliográfica.	46

Lista de códigos

Código 1	–	Retorno de Login de Sucesso.	50
Código 2	–	Formato do JWT Decodificado.	51
Código 3	–	Visualização de ADMIN no MongoDB.	52
Código 4	–	Visualização de Competidor no MongoDB.	52
Código 5	–	Resposta da Requisição de Listas.	54
Código 6	–	Resposta da Requisição de Lista Curada.	56
Código 7	–	Início da Configuração do Apache.	71
Código 8	–	Habilitação de Módulos do Apache.	71
Código 9	–	Execução Docker MongoDB.	71
Código 10	–	Execução Docker <i>mongodump</i>	72
Código 11	–	Copiando para o servidor.	72
Código 12	–	Importando dados para Docker MongoDB.	72
Código 13	–	Conteúdo <i>.bashrc</i>	72
Código 14	–	Copiando pacote para o servidor.	73
Código 15	–	Configuração de Aplicação no Apache.	73
Código 16	–	Execução de Aplicação em Segundo Plano.	73
Código 17	–	Aprovar e Definir Usuário como ADMIN.	74
Código 18	–	Variáveis de Ambiente do Frontend.	74
Código 19	–	Configuração do Frontend no Apache.	74

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
IC	Instituto de Computação
ICPC	<i>International Collegiate Programming Contest</i>
IOI	Olimpíada Internacional de Informática
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JSON Web Token</i>
OBI	Olimpíada Brasileira de Informática
OJ	<i>Online Judge</i>
PO	<i>Product Owner</i>
SBC	Sociedade Brasileira de Computação
SPOJ	<i>Sphere Online Judge</i>
UNICAMP	Universidade Estadual de Campinas

Sumário

1	INTRODUÇÃO	21
1.1	Motivação	21
1.2	Objetivos	22
1.3	Estrutura do Trabalho	22
2	FUNDAMENTAÇÃO TEÓRICA	23
2.1	Trabalhos Correlatos	23
2.1.1	C2Ladder	23
2.1.2	CP Academy	25
2.1.3	Virtual Judge	27
2.2	Juiz Eletrônico	30
2.3	Juiz Online	32
2.3.1	AtCoder	33
2.3.2	Judge	35
2.4	Recomendação	38
2.5	Métricas	39
3	METODOLOGIA	41
3.1	Elicitação de Requisitos	41
3.2	Desenvolvimento do Software	42
3.2.1	Método de Desenvolvimento de Software	43
3.2.2	Desenvolvimento	43
3.3	Desenvolvimento do Texto	45
3.4	Métricas	46
4	RESULTADOS	47
4.1	Interface	47
4.2	Autenticação	50
4.3	Listas Curadas	53
4.4	Gerenciamento de Usuários	57
4.5	Treinamento	59
4.6	Métricas	61
5	CONSIDERAÇÕES FINAIS	65
5.1	Trabalhos Futuros	65

	REFERÊNCIAS	67
A	DEPLOY	71
A.1	APIs	73
A.1.1	Primeiro ADMIN	73
A.2	FRONTEND	74

1 Introdução

A programação competitiva é uma atividade relativamente recente, tendo surgido na década de 1970 (VIANNA et al., 2020). Atualmente, no Brasil, existem diversas competições de programação, sendo as mais notáveis a Maratona de Programação¹ e a Olimpíada Brasileira de Informática² (OBI), realizada pela Sociedade Brasileira de Computação (SBC) em conjunto com o Instituto de Computação da Universidade Estadual de Campinas (IC - UNICAMP) (FLORES, 2023).

Originada, em 1996, de competições regionais classificatórias, a Maratona de Programação também é organizada pela SBC e compõe a regional latino americana do *International Collegiate Programming Contest*³ (ICPC), o concurso internacional de programação universitária (SBC, 2023).

Geralmente, durante uma competição, os competidores, que podem ser indivíduos ou equipes, resolvem um conjunto de problemas. Tais problemas são geralmente abertos, o que significa que não existe uma única e definitiva solução correta. O objetivo do competidor é encontrar uma solução eficiente que atenda aos requisitos de uso de memória, tempo de execução e outros critérios definidos, e que seja bem-sucedida em todos os casos de teste, públicos e secretos.

A ferramenta responsável por analisar a corretude de uma solução é o juiz eletrônico. Um tipo de juiz eletrônico é o juiz *online*, ou *online judge* (OJ), que, além de verificar a solução, disponibiliza uma variedade de problemas de programação competitiva de diferentes níveis de dificuldade e em diferentes tópicos para treinamento, além de manter registro das estatísticas de seus usuários. Essas características o transforma em um meio fundamental para preparação daqueles que desejam competir.

1.1 Motivação

Para que essa preparação seja eficaz, é necessário que o futuro competidor selecione problemas que realmente contribuam para o seu aprendizado, evitando resolver problemas de conceitos que já domina ou aqueles que ainda não estudou. Dada a grande quantidade de problemas disponíveis, receber recomendações é considerável, não apenas para aprimorar o desempenho, mas também, segundo Ferreira (2023), para manter a motivação do atleta.

¹ <<https://maratona.sbc.org.br/>>

² <<https://olimpiada.ic.unicamp.br/>>

³ <<https://icpc.global/>>

A primeira versão do OJTracker, desenvolvida por Ferreira (2023), implementa métodos automatizados de recomendação de problemas para as plataformas Codeforces⁴, AtCoder⁵, Online Judge⁶, Sphere Online Judge (SPOJ)⁷ e Codechef⁸. No entanto, os técnicos (*coaches*) ainda não tinham sido incorporados à aplicação, o que impedia que eles utilizassem sua experiência e conhecimento para indicar problemas com base em estratégias de aprendizado incremental específicas. Consequentemente, um técnico também não conseguia se beneficiar diretamente do uso da plataforma durante os treinamentos dos competidores.

1.2 Objetivos

Visando atender essas demandas, o objetivo geral deste trabalho é adicionar, ao OJTracker, o suporte aos técnicos de programação competitiva e ampliar as opções dos usuários na aplicação, com coleta de métricas de *software* para apoiar futuras tomadas de decisão. A fim de alcançar esse objetivo, foram definidos os seguintes objetivos específicos:

1. permitir que um treinador elabore treinamentos e listas de problemas, utilizando o material disponível nas plataformas já integradas na aplicação;
2. desenvolver novos métodos automatizados de recomendação de problemas;
3. realizar as melhorias necessárias para os novos recursos, refinar a interface, implementar autenticação com diferentes níveis de acesso de usuário, entre outras;
4. implantar os serviços da aplicação na infraestrutura própria da universidade, reduzindo custos e ampliando o acesso por parte dos competidores e técnicos.

1.3 Estrutura do Trabalho

O trabalho em questão está dividido em cinco capítulos. O primeiro capítulo, Introdução, estabelece o contexto, a motivação e os objetivos deste trabalho, além de detalhar sua estrutura. O segundo, Fundamentação Teórica, aborda os conceitos essenciais para compreensão do trabalho. A Metodologia descreve o procedimento seguido na execução. Resultados, o quarto capítulo, apresenta e discute criticamente os resultados obtidos. E, por fim, Considerações Finais avalia o desenvolvimento do trabalho e propõe futuras melhorias.

⁴ <<https://codeforces.com/>>

⁵ <<https://atcoder.jp/>>

⁶ <<https://onlinejudge.org/>>

⁷ <<https://br.spoj.com/>>

⁸ <<https://www.codechef.com/>>

2 Fundamentação Teórica

O propósito deste capítulo é apresentar trabalhos correlacionados e elucidar os conceitos indispensáveis para a compreensão deste trabalho. A Seção 2.1 aborda projetos que apresentam afinidades com o que será desenvolvido. A Seção 2.2 inicia a explanação dos conceitos, definindo juiz eletrônico. A conceituação dos juízes *online*, além da apresentação de exemplos desses juízes, é realizada na Seção 2.3. Na Seção 2.4, são conceitualizados os sistemas de recomendação. Ao final, na Seção 2.5, métricas e conceitos relacionados, como usabilidade, são definidos.

2.1 Trabalhos Correlatos

Nesta seção examinaremos três projetos correlatos: o C2Ladder, uma aplicação *web* com listas curadas de problemas do Codeforces; o CP Academy, que recomenda listas de problemas por tópicos e níveis de dificuldade; e o Virtual Judge, uma plataforma abrangente para prática de programação competitiva. Cada projeto oferece abordagens distintas, como critérios de seleção baseados em *rating*, organização por tópicos e simulação de submissões em vários juízes online. Essa análise crítica dos trabalhos correlatos é essencial para identificar lacunas, compreender abordagens e reconhecer contribuições, fornecendo base para o presente trabalho.

2.1.1 C2Ladder

O C2Ladder¹ é um projeto *Open Source* desenvolvido em repositório hospedado no GitHub, [c2ladders](https://github.com/prince776/c2ladders)², em que os principais contribuidores são Prince Gupta e Abhijeet Kumar. Esse projeto é uma aplicação *web* que indica listas de problemas da plataforma Codeforces de acordo com um *rating* (índice de ranqueamentos dos usuários no Codeforces) selecionado, como podemos observar na Figura 1.

As listas são estáticas e o critério de seleção dos problemas foi a quantidade de pessoas que resolveram e tiveram um aumento constante de classificação (C2LADDER, 2023), que no Codeforces também é chamada de *rating*. Além disso, é possível inserir um identificador de usuário da plataforma Codeforces. Com esse identificador, conforme podemos observar na Figura 2, o C2Ladder consegue coletar algumas informações do usuário, o título de classificação, o *rating* atual e o *rating* máximo, e calcular quantos problemas da lista selecionada ele resolveu.

¹ <<https://c2-ladders-juol.onrender.com/>>

² <<https://github.com/prince776/c2ladders>>

Figura 1 – Visualização de Listas no C2Ladder.

Index	Problem	Frequency	Rating	Status
1	Same Differences	4246	1200	-
2	Challenging Cliffs	4026	1200	-
3	Pleasant Pairs	3982	1200	-
4	Palindrome Game (easy version)	3632	1200	-
5	Stable Groups	3595	1200	-
6	Mocha and Hiking	3520	1200	-

Fonte: [C2Ladder \(2023\)](#)

Figura 2 – Informações de Usuário no C2Ladder.



tourist

tourist (legendary grandmaster)
 Rating: 3585 (Max: 3979)
 Solved: 19 Unsolved: 57

Fonte: [C2Ladder \(2023\)](#)

Figura 3 – Visualização de Status no C2Ladder.

42	Modest Substrings	6	3500	-
43	Spaceship Crisis Management	5	3500	AC
44	Rin and The Unknown Flower	5	3500	AC
45	Spiderweb Trees	5	3500	WA
46	Two Sequences	5	3500	-
47	Boboniu and Banknote Collection	5	3500	-
48	Finding the Vertex	5	3500	-
49	Prison Break	5	3500	WA
50	Epic Convolution	5	3500	AC
51	Ruler Of The Zoo	5	3500	TLE
52	Xor on Figures	5	3500	-

Fonte: C2Ladder (2023)

A Figura 3 mostra que, com os dados coletados, também é possível preencher as colunas *Status* (última coluna) da lista. Essa coluna indica se o usuário já realizou alguma submissão no problema e qual foi o resultado da última submissão. Por fim, os autores do C2Ladder o definem como a reconstrução do descontinuado projeto A2OJ Ladders com um conjunto de problemas atualizado (C2LADDER, 2023).

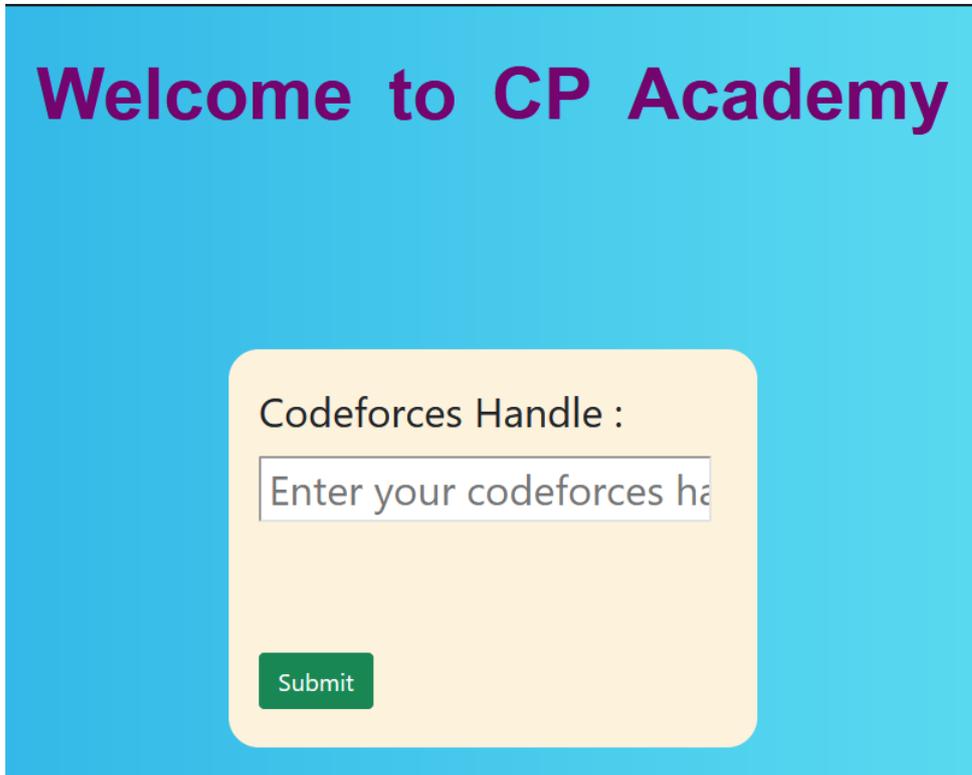
2.1.2 CP Academy

O CP Academy³ é uma aplicação *web* criada para prática de problemas de programação competitiva da plataforma *Codeforces*. Ao acessar a aplicação, como podemos observar na Figura 4, o usuário precisa informar seu identificador de usuário do *Codeforces*. Após informar um identificador válido, ele é redirecionado para a página onde as listas de problemas serão apresentadas.

Na página das listas de problemas o usuário deve indicar um tópico, conforme podemos observar na Figura 5. Atualmente, existem 36 tópicos possíveis, como *chinese remainder theorem*, *divide and conquer* e *string suffix structures*. Selecionando um tópico, as listas de problemas, que são organizadas por dificuldade, serão preenchidas, como é apresentado na Figura 6.

³ <<https://cp-academy.netlify.app/>>

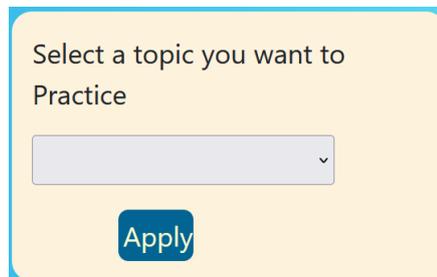
Figura 4 – Página Inicial do CP Academy.



The image shows the homepage of CP Academy. At the top, there is a large blue banner with the text "Welcome to CP Academy" in a bold, purple font. Below the banner is a light yellow rounded rectangle containing a form. The form has the label "Codeforces Handle :" followed by a text input field with the placeholder text "Enter your codeforces handle". Below the input field is a green button labeled "Submit".

Fonte: CP-Academy (2023)

Figura 5 – Seleção de Tópico no CP Academy.



The image shows a form for selecting a topic to practice. It features a light yellow rounded rectangle with a blue border. Inside, the text "Select a topic you want to Practice" is displayed above a dropdown menu. Below the dropdown menu is a blue button labeled "Apply".

Fonte: CP-Academy (2023)

Figura 6 – Listas de Problemas no CP Academy.

Basics	
S.No.	Questions
1	Smilo and Monsters
2	Fear of the Dark
3	Haunted House
4	Dances (Easy version)
5	Iva & Pav
6	Money Trees
7	Building an Aquarium
8	Data Structures Fan
9	Colorful Table
10	Ice Cream Balls

Fonte: [CP-Academy](#) (2023)

Ao todo, são quatro listas de problemas: Basics, Easy, Medium e Hard. O nível de dificuldade dos problemas é definido conforme a classificação do usuário na plataforma Codeforces e o conjunto de problemas utilizados na construção das listas de recomendação são constantemente atualizados, ou seja, novos problemas do Codeforces também são considerados ([CP-ACADEMY, 2023](#)).

Para concluir, o CP Academy também é um projeto *open source* hospedado no GitHub, no repositório [CP-Academy](#)⁴. No momento presente, o projeto conta com quatro contribuidores, sendo um deles Rashmi Singh.

2.1.3 Virtual Judge

O Virtual Judge⁵, também conhecido como vjudge, é uma plataforma online que fornece um ambiente para prática e aprimoramento de habilidades de programação competitiva. A plataforma é robusta e oferece uma ampla gama de recursos, como biblioteca de problemas, hospedagem de competições (*contests*), cadastro de usuário, fórum de discussão, entre outros.

O vjudge é uma aplicação gratuita e aberta a todos. Como o nome sugere, essa aplicação não é um juiz online real. Os problemas são extraídos de juizes online regulares

⁴ <<https://github.com/Code-Aditya-14/CP-Academy>>

⁵ <<https://vjudge.net/>>

Figura 7 – Listas de Problemas no Virtual Judge.

OJ	Prob	Title	Source	Solved	Update Time
☆ POJ	3904	Sky Code	Southeastern European Regional Programming Contest 2008	1380	9 min ago
☆ Gym	101193A	Street magic	2015-2016 6th BSUIR Open Pro	224	9 min ago
☆ Gym	101484I	Matrix Sum	2017 USP-ICMC	54	9 min ago
☆ 计蒜客	T1794	学习效率		946	9 min ago
☆ UVA	247	Calling Circles		5577	9 min ago
☆ AtCoder	tenka1_2012_qual	算盤の書	天下一プログラマーコンテスト2011	10	9 min ago
☆ LibreOJ	3156	回家路线	NOI2019	800	10 min ago
☆ Gym	340533A	Командир Ciel	difbmsklbf	18	10 min ago
☆ HDU	7202	Maex	2022"杭电杯"中国大学生算法设计赛	381	10 min ago
☆ LibreOJ	3176	景点划分	IOI2019	316	10 min ago
☆ SPOJ	FINDPRM	Finding Primes	own problem used for Codechef	1534	10 min ago
☆ SPOJ	CRYPTON	The Embarrassed Cryptographer	NCPC 2005	375	10 min ago
☆ CSES	1686	Coin Collector	Graph Algorithms	3555	10 min ago
☆ SPOJ	AMP10C	Source-Free Factorization	own problem, ICPC Asia region	1679	10 min ago

Fonte: [vjudge \(2023\)](#)

e as submissões dos usuários são simuladas (VJUDGE, 2023), realizadas no juiz de origem indiretamente. Atualmente, suporta 49 juízes online, entre eles, o HackerRank⁶, o SPOJ, o Codeforces e o AtCoder, e, conseqüentemente, consegue reunir e oferecer uma ampla variedade de problemas para programadores de todos os níveis de habilidade.

Na Figura 7 é possível observar como a plataforma vjudge apresenta sua lista de problemas. A lista pode ser filtrada por problemas resolvidos (Solved), problemas tentados (Attempted) e problemas favoritos (Favorites). Os problemas podem ser buscados, utilizando campos de texto, por código do problema (Prob), título do problema (Title) e fonte do problema (Source), que é um *contest* (concurso) na maioria dos casos. E também são exibidos a quantidade de pessoas que solucionaram o problema (coluna Solved) e a última vez que ele foi atualizado (Update Time).

Na lista de problemas, também é possível selecionar um juiz online em OJ. Como podemos observar na Figura 8, a seleção, provavelmente pelo grande número de opções, ocorre através de um *modal*. Os juízes online que são apresentados com um alto nível de opacidade são os juízes em que a maioria dos problemas não são atualizados há bastante tempo.

Ademais, o Virtual Judge facilita a criação de competições. Para criar uma competição, não é preciso desenvolver novos problemas e dados de teste (VJUDGE, 2023). Conforme apresentado na Figura 9, a criação da competição também é realizada em um *modal* e é possível definir diversas configurações, como título, horário de início, regra de ranqueamento, entre outras.

Os problemas da competição são problemas dos juízes online suportados pela aplicação. Para adicioná-los na competição, no final do *modal*, o usuário deve selecionar o

⁶ <<https://www.hackerrank.com/>>

Figura 8 – Seleção de OJ no Virtual Judge.

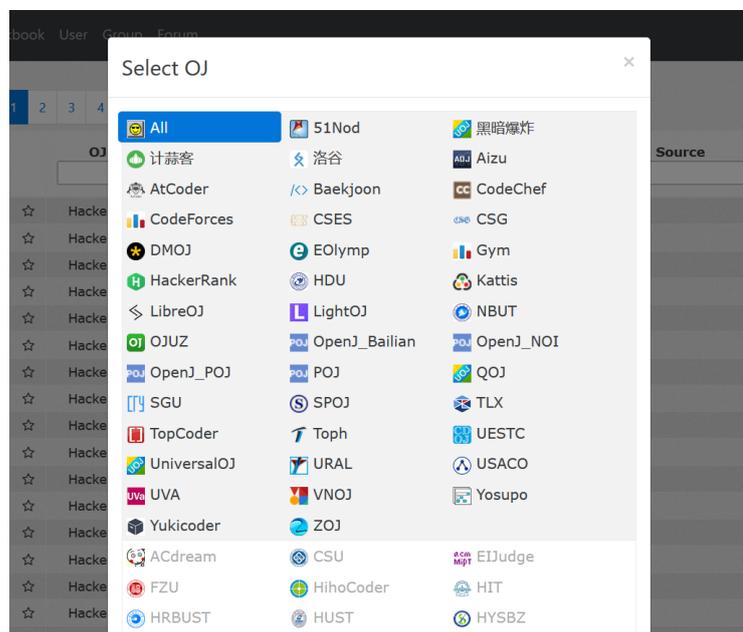
Fonte: [vjudge \(2023\)](#)

Figura 9 – Criação de Competição no Virtual Judge.

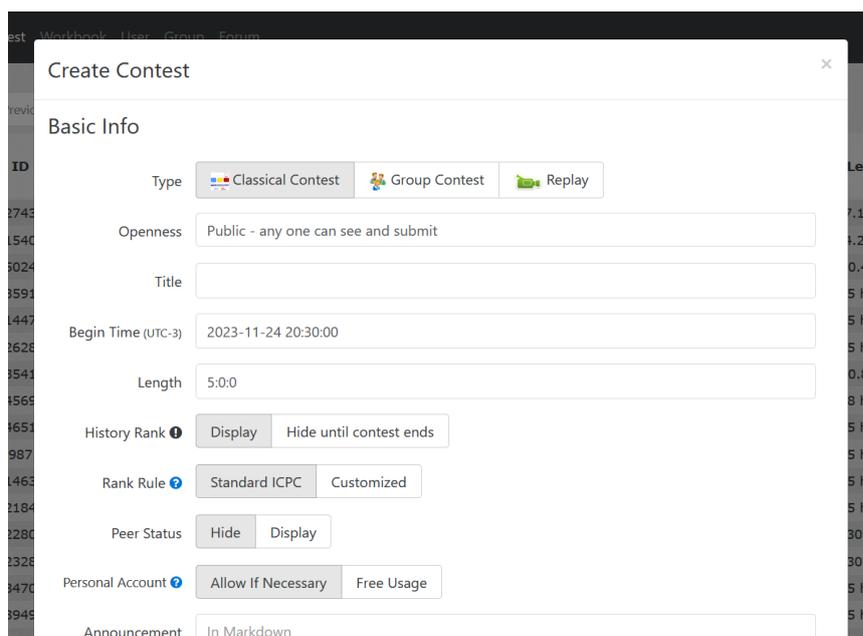
Fonte: [vjudge \(2023\)](#)

Figura 10 – Adição de Problema em Competição no Virtual Judge.

OJ	ProbNum	Wgt.	Alias	Title
CodeForces	123A	1		A ✓ Prime Permutation

+ Add a problem

Cancel Confirm

Fonte: [vjudge \(2023\)](#)

juiz online e inserir o código de identificação de cada problema desejado. Se esse código de identificação for válido, a aplicação vjudge identificará automaticamente o problema, preenchendo o seu título. Na Figura 10, podemos visualizar essas características e observar que o usuário também pode atribuir um pseudônimo (Alias) e um peso (Wgt.) customizado ao problema.

2.2 Juiz Eletrônico

O juiz eletrônico é o sistema encarregado de executar as etapas do procedimento de avaliação de soluções para problemas de programação competitiva (RAHMAN et al., 2023). Dessa forma, representa um componente essencial e altamente sensível dentro do ecossistema de uma competição de programação, sendo responsável por verificar a correção e outros aspectos, como a eficiência, das soluções submetidas pelos participantes.

A primeira etapa do procedimento de avaliação é a submissão. Durante essa fase, é realizada uma verificação e compilação do código submetido, para confirmar a viabilidade de execução do código no ambiente homogêneo de avaliação (WASIK et al., 2017). Se não for possível compilar corretamente a submissão, o juiz retornará *Compilation Error* (CE), erro de compilação em português (WATANOBE et al., 2022).

A análise é a segunda etapa do procedimento de avaliação. Nessa etapa, a submissão é avaliada com base no conjunto de casos de teste específicos do problema (WASIK et al., 2017). Os resultados possíveis da aplicação dos casos de teste são apresentados no Quadro 1. Podemos observar que para cada execução de caso de teste, é verificado se o processo ocorreu sem erros, se quaisquer limitações de recursos específicas do pro-

blema não foram excedidas e se a saída obtida obedece às regras descritas na definição do problema.

Quadro 1: Retornos da Análise de uma Solução por um Juiz Eletrônico.

Código	Retorno	Descrição
RE	<i>Runtime Error</i> (Erro de Tempo de Execução)	Indica que ocorreu um erro durante a execução da solução. Exemplo: divisão por zero, estouro de pilha, etc.
TLE	<i>Time Limit Exceeded</i> (Limite de Tempo Excedido)	Indica que o uso da CPU excedeu o tempo limite especificado.
MLE	<i>Memory Limit Exceeded</i> (Limite de Memória Excedido)	Indica que o uso de memória excedeu o limite especificado.
OLE	<i>Output Limit Exceeded</i> (Limite de Saída Excedido)	Indica que o tamanho da saída gerada pela solução excedeu o limite especificado.
WA	<i>Wrong Answer</i> (Resposta errada)	Indica que a saída da solução não coincide com a saída esperada.
PE	<i>Presentation Error</i> (Erro de Apresentação)	Indica que a saída da solução está formatada incorretamente.
AC	<i>Accepted</i> (Solução Aceita)	Indica que a solução atendeu todas as condições e não ocorreram falhas em sua execução.

Fonte: Elaborado pelo autor com base em [Watanobe et al. \(2022\)](#).

Finalmente, considerando os resultados de todos os casos de teste da solução de um problema, a pontuação da solução é calculada. Essa é terceira e última etapa, pontuação, do procedimento de avaliação ([WASIK et al., 2017](#)).

A construção de um juiz eletrônico demanda a implementação de requisitos especiais e específicos. A precisão na medição do tempo de execução, frequentemente avaliado em milissegundos, dos casos de teste é vital para garantir a igualdade de avaliação de soluções de um problema e, em uma esfera maior, garantir uma competição equitativa. Nesse contexto, é essencial que o sistema possua métodos de análise de desempenho suficientemente sensíveis e determinísticos para distinguir pequenas frações de tempo, assegurando medições reproduzíveis em execuções consecutivas do mesmo código para um caso de teste específico ([WASIK et al., 2017](#)).

Em competições, o juiz também deve estar preparado para receber um grande número de submissões em um curto período de tempo. Para cumprir esse requisito, o sistema deve ser eficiente, e a estratégia indicada para maioria dos casos é ter uma arquitetura que utilize execução concorrente e processamento paralelo ([WASIK et al., 2017](#)).

Por fim, é importante ressaltar que nem todos os juizes eletrônicos necessariamente seguem todas as características e requisitos descritos nessa seção ([WASIK et al., 2017](#)). A variedade de aplicações pode resultar em juizes eletrônicos com diferentes funcionalidades.

dades e prioridades, adaptados às necessidades específicas de cada ambiente. Em alguns casos, não há necessidade de calcular pontuação ou essa é calculada considerando outros aspectos, como tempo para submissão da solução.

2.3 Juiz Online

Um sistema de juiz online é um tipo de juiz eletrônico. Segundo [Wasik et al. \(2017\)](#), juiz online é um juiz eletrônico que realiza qualquer uma das etapas do procedimento de avaliação online. A viabilidade da utilização desses juízes pela internet permitiu que evoluíssem rapidamente, agrupando novas funcionalidades, que, conseqüentemente, possibilitaram o seu uso em diferentes contextos.

Inicialmente, os juízes online eram poucos e as suas funcionalidades se restringiam a organização de concursos competitivos de programação, como a International Olympiad in Informatics (IOI), Olimpíada Internacional de Informática em português, e arquivamento de problemas utilizados durante tais competições ([WASIK et al., 2018](#)). Com o passar dos anos, novos juízes surgiram e, por serem de fácil acesso, se popularizaram.

A natureza remota e de alta disponibilidade qualificam esses juízes como ferramentas de fácil acesso. Os seus usuários podem submeter códigos a qualquer momento e de qualquer lugar, recebendo *feedback* instantâneo. Essa acessibilidade elimina a necessidade de configuração local complexa, permitindo que programadores aprimorem suas habilidades e participem de competições de programação de forma conveniente. Portanto, os juízes online desempenham um papel crucial na facilitação do acesso à prática de programação.

As pessoas interessadas solucionam problemas frequentemente, o que cria uma demanda por um aumento contínuo na quantidade de problemas disponíveis. Atualmente, os juízes online de destaque possuem bases de dados com milhares de problemas, como podemos observar na Tabela 1, de diferentes níveis de dificuldade, tópicos, estratégias de solução, entre outras características.

Tabela 1 – Quantidade de Problemas de OJs em 26/11/2023.

Juiz Online	Quantidade
Codeforces	9064
OnlineJudge	4975
AtCoder	6380
SPOJ	3964

Fonte: o Autor.

Os juízes online também estão sendo empregados em processos educativos, não somente em treinamentos para programação competitiva mas também em processos de

recrutamento de funcionários (LIU et al., 2023). O URI Online Judge, atual Beecrowd⁷, o CD-MOJ⁸ (Contest-Driven Meta Online Judge) e o Jutge⁹ são exemplos de sistemas que, para auxiliar tanto os alunos quanto os professores, integram as funcionalidades comuns de um juiz online com as de um sistema de gerenciamento de aprendizado (MANI et al., 2014).

Grandes empresas utilizam os sistemas de juízes online para buscar e avaliar desenvolvedores (LIU et al., 2023). A plataforma TopCoder¹⁰ é uma aplicação de recrutamento que inclui programação competitiva e, conseqüentemente, um juiz online. Nessa plataforma, empresas buscam e podem fazer propostas para desenvolvedores altamente qualificados (MANI et al., 2014). Outras plataformas, como Codewars¹¹, InterviewBit¹², e LeetCode¹³, são focadas em auxiliar aspirantes a programadores profissionais a treinarem para entrevistas técnicas de seleção, oferecem cursos, mentorias, um ambiente para resolução de problemas típicos, entre outros recursos (LIU et al., 2023).

2.3.1 AtCoder

O AtCoder é uma plataforma *web* japonesa de competições de programação. Semanalmente, concursos de programação são realizados de forma online pela plataforma AtCoder. O público alvo desses concursos é bastante diverso, desde iniciantes até especialistas (ATCODER, 2023). Na Figura 11 podemos visualizar a sua página inicial.

Na página inicial, a aplicação AtCoder apresenta as competições permanentes, as próximas competições, um *feed* de competições e uma pequena seção de informações. A aplicação pode ser utilizada nos idiomas inglês e japonês, além de suportar o cadastro de usuários. Conforme podemos observar na Figura 12, cada usuário da plataforma possui um perfil.

No perfil de usuário são apresentados a foto, informações, como nome, país/região, entre outros dados, histórico de competição e recente condição competitiva, como posição do *ranking* do usuário. Os *rankings* gerais do AtCoder são acessados na página de *ranking*, que pode ser visualizada na Figura 13. Essa página informa a última atualização dos *rankings* e permite que o usuário aplique filtros na listagem, como seleção de tipo de usuário e intervalo de *rating*, que é o índice de ranqueamento.

Para acessar todas as competições da plataforma, incluindo as arquivadas, o usuário deve abrir a página de *contest*. Nessa página, como podemos visualizar na Figura 14,

⁷ <<https://www.beecrowd.com.br/>>

⁸ <<https://moj.naquadah.com.br/>>

⁹ <<https://jutge.org/>>

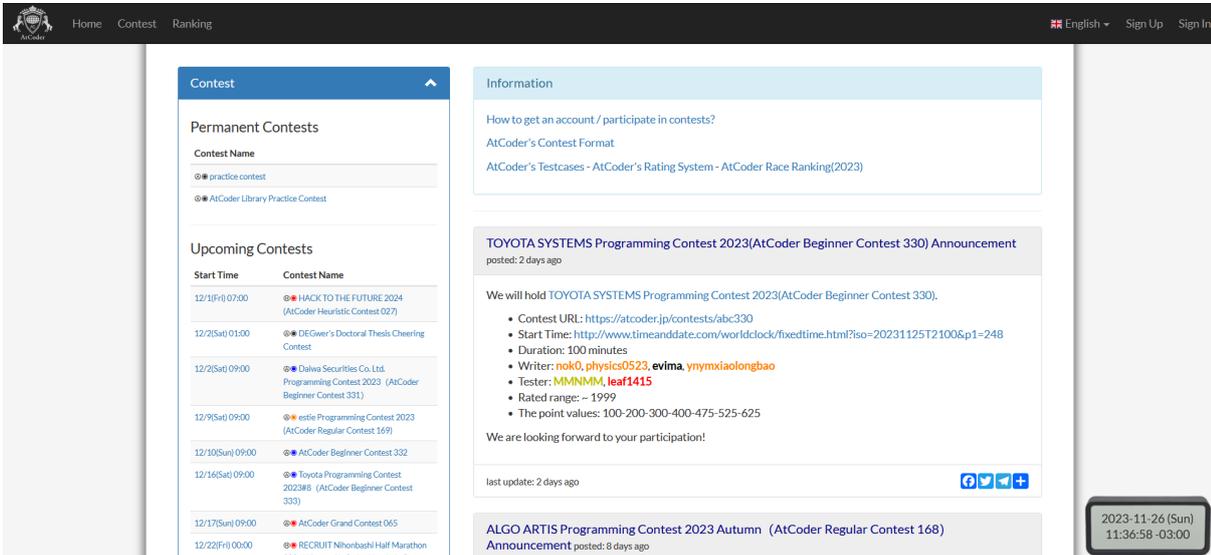
¹⁰ <<https://www.topcoder.com/>>

¹¹ <<https://www.codewars.com/>>

¹² <<https://www.interviewbit.com/>>

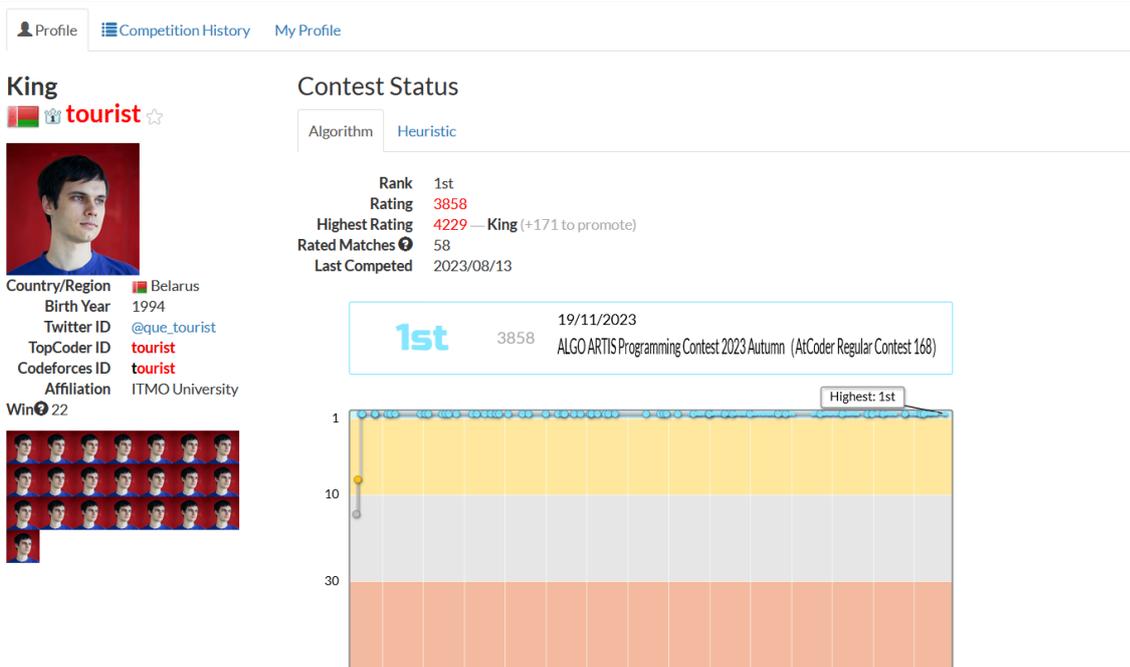
¹³ <<https://leetcode.com/>>

Figura 11 – Página Inicial do AtCoder.



Fonte: AtCoder (2023)

Figura 12 – Página de Perfil de Usuário no AtCoder.



Fonte: AtCoder (2023)

Figura 13 – Página de Ranking do AtCoder.

Ranking Active Users ? Last Updated: 2023-11-25(Sat) 18:00

Algorithm Heuristic

Active Users All Users

1 2 4 8 16 32 64 128 256 512 1024 1045

Filter						
Rank	User	Birth	Rating	Highest	Match	Win
1	 tourist <small>ITMO University</small>	1994	3858	4229	58	22
2	 ksun48 <small>MIT</small>	1998	3679	3802	57	5
3	 Um_nik	1996	3648	3948	59	7
4	 apiad	1997	3638	3852	50	6
5	 Stonefeang <small>University of Warsaw</small>	1997	3630	3658	36	2
6	 ecnerwala	1997	3613	3814	35	2
7	 mnbvmar <small>University of Warsaw</small>	1996	3555	3736	22	1
8	 newbiedmy <small>MIT</small>	2002	3516	3714	25	1
9	 semiexp <small>Preferred Networks, Inc.</small>		3481	3594	24	3
10	 jiangly <small>Peking University</small>	2003	3446	3488	26	2

Fonte: [AtCoder \(2023\)](#)

é possível buscar competições, filtrando por intervalo de *rating* e categoria ou procurando por nome.

Por fim, segundo [AtCoder \(2023\)](#), a aplicação é desenvolvida e administrada pela empresa AtCoder Inc., estabelecida em 20 de junho de 2012 e sediada no Shinjuku SKY Bldg, 4-1-23, Shinjuku, Shinjuku-ku, Tokyo, 160-0022, Japão. Atualmente, essa empresa está sob a liderança do CEO Naohiro Takahashi e possui um capital de 100 milhões de ienes jponês ([ATCODER, 2023](#)).

Também compõem os negócios da empresa o planejamento e a administração de concursos de programação ([ATCODER, 2023](#)). Essa empresa tem desempenhado um papel significativo no cenário da programação, fornecendo uma plataforma para competições e contribuindo para o desenvolvimento e avaliação de talentos na área de engenharia de software e afins.

2.3.2 Jutge

O Jutge é um juiz de programação educacional online e gratuito, um ambiente virtual de aprendizagem para programação de computadores que oferece cursos bem organizados com problemas classificados e outras funcionalidades para estudantes e professores

Figura 14 – Página de Contest do AtCoder.

Present Contests Past Contests

Search in Archive

Rated Range

- ABC Class (Rated for ~1999)
- ARC Class (Rated for ~2799)
- AGC Class (Rated for ~9999)
- AHC Class

Category

- All
- Unofficial(unrated)
- Sponsored Tournament
- Sponsored Parallel(rated)
- Sponsored Parallel(unrated)
- Sponsored ABC
- Sponsored ARC
- Heuristic Contest
- Sponsored Heuristic Contest

Search

Contest Name

Keyword

Contest Archive

1 2 4 8 12

Start Time (local time)	Contest Name	Duration	Rated Range
2023-11-25(Sat) 09:00	TOYOTA SYSTEMS Programming Contest 2023(AtCoder Beginner Contest 330)	01:40	-1999
2023-11-19(Sun) 09:00	ALGO ARTIS Programming Contest 2023 Autumn (AtCoder Regular Contest 168)	02:00	-2799
2023-11-18(Sat) 09:00	S k y Inc, Programming Contest 2023 (AtCoder Beginner Contest 329)	01:40	-1999
2023-11-11(Sat) 09:00	Toyota Programming Contest 2023#7 (AtCoder Beginner Contest 328)	01:40	-1999
2023-11-05(Sun) 03:00	Toyota Programming Contest 2023#6 (AtCoder Heuristic Contest 026)	04:00	All
2023-11-04(Sat) 09:00	HHKB Programming Contest 2023(AtCoder Beginner Contest 327)	01:40	-1999
2023-10-28(Sat) 09:00	Panasonic Programming Contest 2023 (AtCoder Beginner Contest 326)	01:40	-1999
2023-10-21(Sat) 09:00	KEYENCE Programming Contest 2023 Autumn (AtCoder Beginner Contest 325)	01:40	-1999
2023-10-15(Sun) 09:00	AtCoder Regular Contest 167	02:00	-2799
2023-10-14(Sat) 09:00	Japan Registry Services (JPRS) Programming Contest 2023 (AtCoder Beginner Contest 324)	01:40	-1999
2023-10-14(Sat) 00:00	AtCoder Heuristic Contest 025	199:00	All
2023-10-08(Sun) 09:00	AtCoder Regular Contest 166	02:00	-2799
2023-10-07(Sat) 09:00	UNIQUE VISION Programming Contest 2023 Autumn(AtCoder Beginner Contest 323)	01:40	-1999
2023-09-30(Sat) 09:00	AtCoder Beginner Contest 322	01:40	-1999
2023-09-24(Sun) 03:00	Marubeni Programming Contest 2023 (AtCoder Heuristic Contest 024)	04:00	All
2023-09-23(Sat) 09:00	Suntory Programming Contest 2023 (AtCoder Beginner Contest 321)	01:40	-1999

Fonte: [AtCoder](#) (2023)

Figura 15 – Página Inicial do Jutge.

Jutge.org

Jutge.org

The Virtual Learning Environment for Computer Programming

user@example.com password Sign in

Registration Reset password Telegram channel

Fonte: [Jutge](#) (2023)

(JUTGE, 2023). A página inicial da plataforma web Jutge é apresentada na Figura 15 e as funcionalidades dessa plataforma divididas por público alvo são listadas no Quadro 2.

A plataforma Jutge também proporciona a realização de competições de programação (JUTGE, 2023). Atualmente, já foram realizados 274 competições, 2630 exames, 5.2 milhões de submissões e estão cadastrados na plataforma 5518 problemas, segundo

Quadro 2: Funcionalidades do Jutge.

Para Estudantes	Para Professores
Acesso a cursos públicos	Automatização do processo de correção
Inscrição nos cursos do professor	Criação de cursos e listas
Biblioteca de problemas com diferentes níveis de dificuldade	Gerência de lista de tarefas
<i>Feedback</i> instantâneo sobre soluções	Estatísticas instantâneas sobre alunos
Prática de linguagens de programação	Compartilhamento de cursos, listas e documentos
	Reutilização de problemas
	Criação de problemas

Fonte: Elaborada pelo autor com base em [Jutge \(2023\)](#)

Figura 16 – Página de Problemas do Jutge.

Problems

Català **English** Español Français Deutsch

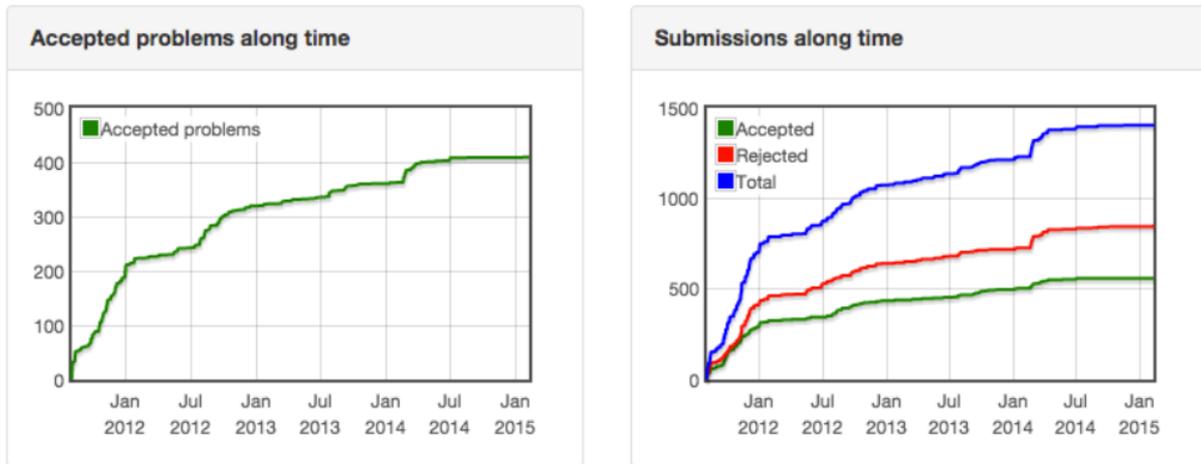
G00001	BattleRoyale	
G00002	Tron3D	
G00007	StarWar	
G00009	Poquémon	
G16109	Tron3D 2017	
G57610	Dominator	
P00000	Hello world!	
P10051	Worst path	
P10121	Number of fiscal identification	
P10387	Minimum cost of a correct parenthesization (2)	
P10432	ApocalypseNow2018	
P10622	Three points	
P10694	Bars (3)	
P11141	Students	

Fonte: [Jutge \(2023\)](#)

[Jutge \(2023\)](#). Os problemas são separados pelo idioma utilizado na sua descrição, catalão, inglês, espanhol, francês e alemão, e podem ser visualizados na página de problemas, apresentada na Figura 16.

Esse juiz online suporta 20 linguagens de programação diferentes, linguagens de multiparadigma, como C++, Java, Python, linguagens funcionais, como Haskell, entre outras, além de verificar soluções considerando também restrições de segurança e não

Figura 17 – Exemplos de Gráficos de Progresso do Jutge.



Fonte: [Jutge \(2023\)](#)

somente de tempo e memória ([JUTGE, 2023](#)). Podemos observar na Figura 17, que o juiz também gera gráficos para acompanhamento de progresso.

A aplicação Jutge foi criada por Jordi Petit e Salvador Roura em 2006 ([DÍEZ, 2018](#)). Segundo [Petit et al. \(2017\)](#), nos últimos anos, a aplicação foi extensivamente utilizada na Universitat Politècnica de Catalunya para fortalecer a abordagem de aprendizado prático em diversos cursos. Ademais, também é empregada na Olimpíada Catalã de Informática ([JUTGE, 2023](#)).

O estudo realizado por [Petit et al. \(2017\)](#) indica uma série de impactos positivos advindos da utilização do Jutge em sala de aula, incluindo que a maioria dos alunos considera a plataforma uma grande ajuda e expressa preferência por seu uso.

2.4 Recomendação

A recomendação desempenha um papel crucial na internet contemporânea, e é uma ferramenta essencial para oferecer sugestões relevantes aos usuários em meio à vastidão disponível, otimizando a sua experiência. As estruturas responsáveis por identificar e prover recomendações são chamadas de sistemas de recomendação.

Os sistemas de recomendação são sistemas de software que aliviam o processo de pesquisa em um ambiente sobrecarregado, oferecendo uma coleção de resultados com elementos pertinentes ([JULCA-MEJIA; PAUCAR-CURASMA, 2023](#)). Esses sistemas, dependendo da forma como estão sendo aplicados, utilizam os dados disponíveis para prever recomendações generalizadas, iguais para todos os usuários, ou personalizadas ([SIDDIQUI; AGARWAL, 2018](#)).

Atualmente, existem quatro abordagens de destaque para construção de sistemas

recomendativos: as filtragens baseada em conteúdo, colaborativa, demográfica e híbrida (THORAT; GOUDAR; BARVE, 2015). A filtragem baseada em conteúdo fundamenta-se na busca e recuperação de informações de itens utilizados por um usuário para identificar itens similares (VIEIRA; NUNES, 2012). Caso o usuário avalie o item utilizado anteriormente, essa avaliação será considerada na construção da recomendação (SIDDIQUI; AGARWAL, 2018).

Na filtragem colaborativa, o sistema identifica usuários com características afins ao usuário que solicitou a recomendação e sugere elementos com base nas escolhas desses perfis semelhantes, otimizando a personalização das recomendações (VIEIRA; NUNES, 2012). Já a filtragem demográfica formula a recomendação com base no perfil demográfico do usuário, levando em conta informações fornecidas, como nacionalidade, idade, gênero, etnia, estado civil, entre outras (THORAT; GOUDAR; BARVE, 2015).

A filtragem híbrida, como o próprio nome indica, consiste na combinação de duas ou mais técnicas. A fusão mais prevalente é entre a filtragem baseada em conteúdo e a filtragem colaborativa, explorando a semelhança entre usuários e entre os elementos utilizados (VIEIRA; NUNES, 2012). Essa abordagem foi introduzida para superar alguns problemas, como a especialização excessiva, buscando simultaneamente melhorar a precisão e eficiência do processo (THORAT; GOUDAR; BARVE, 2015).

Finalmente, para alguns casos de uso, como é possível observar na implementação da primeira versão do OJTracker, realizada por Ferreira (2023), a construção de sistemas de recomendação necessita de abordagens específicas e personalizadas.

2.5 Métricas

Métricas são medidas quantitativas utilizadas para avaliar, comparar e monitorar diferentes aspectos, como qualidade, eficiência, entre outros, em diversos contextos, processo, produto ou serviço. Sua coleta possibilita a tomada de decisão informada, por isso a definição e o uso adequado de métricas são cruciais para garantir que os objetivos sejam atingidos de maneira eficaz (DONEGAN et al., 2005).

No contexto de um produto de *software*, as métricas são chamadas de métricas de *software*. As métricas de *software* também desempenham um papel fundamental na medição e avaliação tanto de produtos de *software* quanto dos processos pelos quais eles são desenvolvidos. Como definido por Mills (1988), boas métricas vão além de meramente descrever o que está acontecendo, elas permitem a criação de modelos preditivos que podem antecipar parâmetros do processo ou do produto.

Para que as métricas sejam verdadeiramente eficazes, elas precisam possuir algumas características essenciais (MILLS, 1988), conforme podemos visualizar no Quadro 3.

Quadro 3: Características Essenciais das Métricas de *Software*.

Simplicidade	Definição precisa.
Objetividade	Resultados consistentes e confiáveis.
Fácil Obtenção	Custos não elevados.
Validade	Deve medir o que se pretende.
Robustez	Relativamente insensível a mudanças insignificantes.

Fonte: o Autor.

A simplicidade garante que as métricas serão avaliadas claramente e a objetividade trata da inferência do que foi avaliado, garantindo a minimização da interpretação subjetiva.

3 Metodologia

Este capítulo tem o objetivo de descrever os processos realizados na execução deste trabalho. A Seção 3.1 apresenta o processo de elicitação de novos requisitos para o OJTracker. A Seção 3.2 aborda o desenvolvimento do *software*, fornecendo detalhes sobre o método de desenvolvimento de *software* utilizado e as evoluções realizadas. Na Seção 3.3 a produção do texto é detalhada, explicitando o processo de pesquisa bibliográfica e a ferramenta de escrita utilizada. Por fim, na Seção 3.4, as métricas utilizadas, a forma de coleta e seus objetivos são explicitados.

3.1 Elicitação de Requisitos

Uma das etapas essenciais do ciclo de vida do desenvolvimento de *software* é a elicitação de requisitos (TIWARI; RATHORE, 2017). Essa etapa consiste na identificação e compreensão dos anseios dos *stakeholders*, partes interessadas, de um sistema de *software* (SHARMA; PANDEY, 2013). Através desse processo, garante-se que o *software* desenvolvido atenda às reais necessidades dos usuários e esteja alinhado aos objetivos de negócio.

Este trabalho é a continuação do trabalho desenvolvido por Ferreira (2023). A fim de realizar uma contribuição significativa ao projeto OJTracker, foram utilizados métodos de elicitação de requisitos para identificar novas funcionalidades e correções necessárias ao aprimoramento da aplicação.

Primeiramente, foi utilizada a técnica de questionário. O questionário é um valioso método de elicitação, pois permite a obtenção de evidências estatísticas, a fim de apoiar suposições, e possibilita a coleta de opiniões e sugestões diretamente dos usuários ou outras partes interessadas (IQBAL, 2014). Analisando os resultados do questionário conduzido por Ferreira (2023) após a conclusão da primeira versão do OJTracker, foram identificados os requisitos apresentados no Quadro 4.

Também foi utilizada a técnica do *brainstorm*, que consiste em reunir duas ou mais partes interessadas com o objetivo de identificar e gerar novos requisitos por meio de uma discussão informal e espontânea (OKESOLA et al., 2019). Aplicando esse método, foram elicidados mais três requisitos, que podem ser observados no Quadro 5.

Ao analisar os Quadros 4 e 5, percebe-se que os requisitos são identificados por um código específico, composto por duas letras seguidas de um número de dois dígitos. Para maior clareza, foi adotada a convenção de prefixar os requisitos provenientes do questionário com **RQ** e os do *brainstorm* com **RB**. Dessa forma, o código não apenas

Quadro 4: Requisitos elicitados através do questionário.

Código	Épico	Descrição
RQ01	Interface	Eu, como usuário , desejo que o módulo de inserção de <i>handles</i> seja responsivo.
RQ02	Interface	Eu, como usuário , desejo que o acesso às listas de submissões seja mais intuitivo.
RQ03	Interface	Eu, como usuário , desejo que seja possível filtrar as listas de submissões.
RQ04	Recomendação	Eu, como usuário , desejo que exista um método de recomendação que indique problemas de <i>tags</i> que resolvo com frequência.
RQ05	Recomendação	Eu, como usuário , desejo que exista um método de recomendação que indique problemas de <i>tags</i> que resolvo raramente.
RQ06	Recomendação	Eu, como usuário , desejo que exista um método de recomendação que permita a seleção de múltiplas <i>tags</i> de interesse.
RQ07	Recomendação	Eu, como usuário , desejo que exista um método de recomendação que indique problemas que tentei resolver e não obtive êxito.
RQ08	Recomendação	Eu, como usuário , desejo que exista um método de recomendação que indique problemas em mais de uma plataforma.

Fonte: o Autor

Quadro 5: Requisitos elicitados através do *brainstorm*.

Código	Épico	Descrição
RB01	Listas Curadas	Eu, como treinador , desejo adicionar listas curadas.
RB02	Treinamentos	Eu, como treinador , desejo criar treinamentos para os meus competidores, com problemas que nenhum deles tenha resolvido ainda.
RB03	Autenticação	Eu, como treinador , desejo que existam diferentes níveis de acesso de usuário (<i>user</i> , <i>coach</i> e <i>admin</i>).

Fonte: o Autor

proporciona uma identificação única, mas também indica a origem do requisito.

3.2 Desenvolvimento do Software

Esta seção retrata o desenvolvimento do software do OJTracker. Ela apresenta o método de desenvolvimento de software empregado, detalha as ferramentas utilizadas e discute a nova arquitetura da aplicação, entre outros progressos realizados.

3.2.1 Método de Desenvolvimento de Software

A execução prática desse trabalho de conclusão de curso possui algumas características específicas, como uma equipe de desenvolvimento composta por somente uma pessoa. Por essa razão, o método Scrum com adaptações foi o método de desenvolvimento de software aplicado. O Scrum é um método ágil de desenvolvimento, com característica iterativa e incremental (SACHDEVA, 2016).

Nessa versão adaptada, além do time de desenvolvimento, foi mantido o papel do *Product Owner* (PO). O PO é a pessoa responsável por representar os interesses do cliente na definição de requisitos, na priorização de tarefas e na aceitação das entregas (LIMA; AMARAL, 2022).

As características principais do método também foram preservadas nessa versão. A adaptação mantém uma natureza iterativa, com *sprints* semanais, que começam e terminam às quartas-feiras, continuando incremental, porque, ao fim de cada *sprint*, é realizada uma reunião, presencial ou remota, através da plataforma de comunicação Teams¹, para entrega do que foi produzido.

3.2.2 Desenvolvimento

Com o objetivo de preservar a rastreabilidade ao longo de todo o desenvolvimento do *software* OJTracker, os códigos continuaram sendo hospedados no GitHub². Dessa forma, a ferramenta Git³ foi utilizada para o versionamento, proporcionando uma gestão eficiente e transparente do histórico de alterações. Essa abordagem não apenas facilitou o acompanhamento das evoluções do sistema, mas também permitirá futuras colaborações.

Nesta fase do projeto OJTracker, a plataforma do ambiente de desenvolvimento foi o sistema operacional da Microsoft Windows 11 Home Single Language versão 22H2. Porém, com o Docker Desktop⁴, versão 4.22.1, o docker, versão 24.0.5, e o docker compose, versão v2.20.2-desktop.1, a virtualização também foi utilizada, pois era essencial na execução do banco de dados e no desenvolvimento do *frontend*.

O desenvolvimento do código da aplicação foi realizado utilizando o Visual Studio Code⁵, um editor de código gratuito e de código aberto. Esse editor, além de ser um editor de texto, oferece uma interface intuitiva e recursos poderosos para a escrita, depuração e execução de código, como terminais embutidos.

Outras ferramentas de destaque também foram empregadas, o Postman⁶ e o Fi-

¹ <<https://www.microsoft.com/pt-br/microsoft-teams/group-chat-software>>

² <<https://github.com/>>

³ <<https://git-scm.com/>>

⁴ <<https://www.docker.com/products/docker-desktop/>>

⁵ <<https://code.visualstudio.com/>>

⁶ <<https://www.postman.com/>>

refox⁷, versão *120.0.1 (64-bits)*. O Postman foi usado para acelerar o processo de teste de alterações nas APIs (*Application Programming Interface* ou Interface de Programação de Aplicação) do OJTracker, visto que, com essa ferramenta, não era necessário acessar a API pelo *frontend*.

O Firefox, com o seu conjunto de funcionalidades chamado Ferramentas de Desenvolvimento, contribuiu significativamente para o cumprimento dos requisitos de Interface, que podem ser visualizados no Quadro 4. Esse conjunto inclui Console, para leitura de *logs*, e Redimensionador, para visualização de interface em diferentes resoluções de tela, entre outras utilidades.

Com a finalidade de possibilitar a implementação dos requisitos **RB01** e **RB02**, além de satisfazer o requisito **RB03**, descritos no Quadro 5, como pode-se observar na Figura 18, a arquitetura de software da aplicação OJTracker foi remodelada. A nova arquitetura possui um novo serviço chamado de *Auth* API, que foi a única alteração estrutural necessária.

O novo serviço é responsável por autenticar os usuários. Além disso, possui a função de administrar os conjuntos de dados que envolvem operações restritas, as listas curadas e os treinamentos. Para adicionar nessas coleções de dados, é necessário ser um usuário com nível de acesso *coach* ou *admin* e, para editar ou excluir, o usuário deve ser o *coach* autor do objeto ou um *admin*. Esse serviço persiste dados no banco de dados utilizado na arquitetura anterior, não aumentando a complexidade da solução.

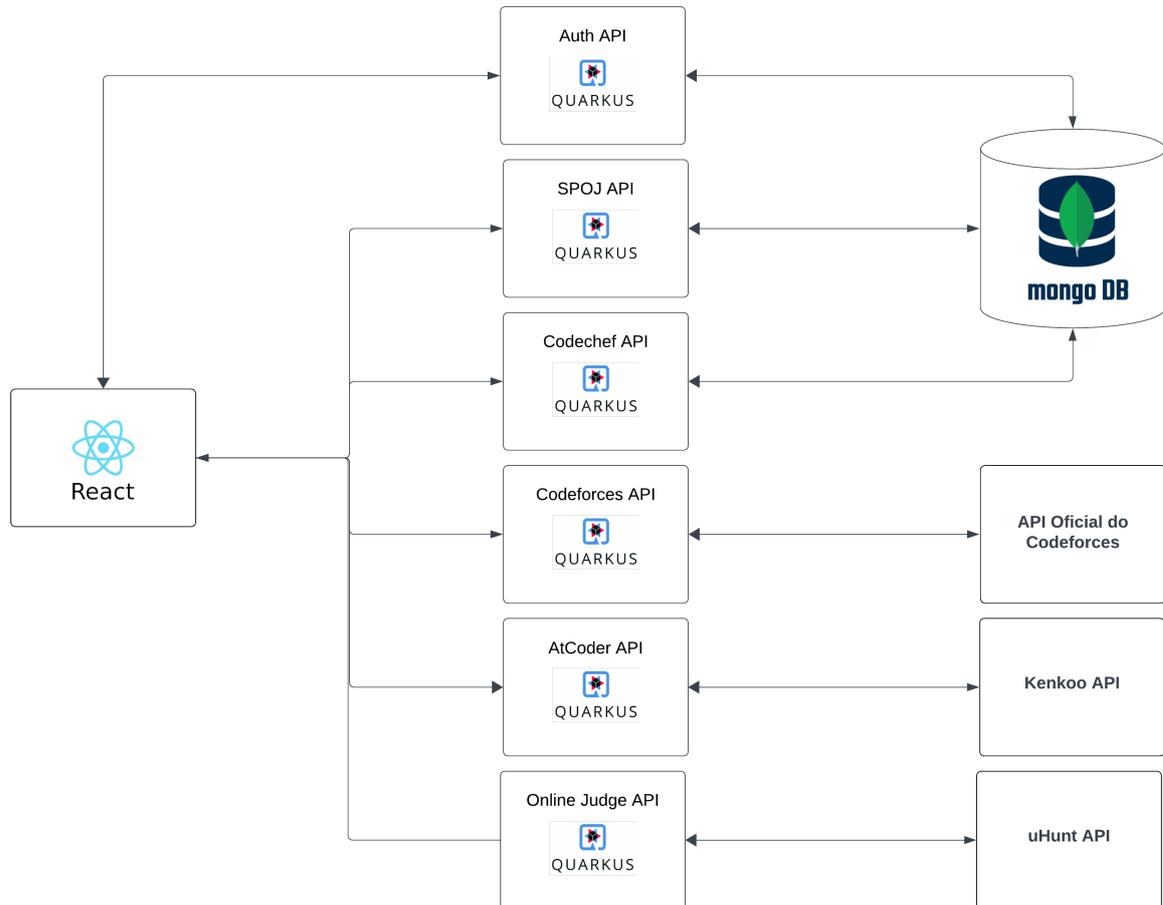
Durante o processo de *login*, a autenticação é realizada mediante o fornecimento de um nome de usuário e senha. No *backend*, a senha recebida na criação de um usuário é, usando a função `bcryptHash` de `io.quarkus.elytron.security.common.BcryptUtil`, transformada em *hash*, garantindo assim uma manipulação e persistência seguras. Por esse motivo, a comparação da senha informada para um usuário com a senha real desse usuário é efetuada por meio da função `BcryptUtil.matches`.

Um *login* bem sucedido indica que o *frontend* recebeu um *JSON Web Token* (JWT). O *token* JWT é uma maneira compacta e autocontida de transmitir informações via JSON com segurança, pois, por ser assinado digitalmente, suas informações podem ser verificadas (OKTA, 2023). No projeto OJTracker, as informações presentes no JWT são o *timestamp* de expiração e o nível de acesso e o nome do usuário autenticado no *login*.

Por conseguinte, o JWT foi utilizado para autenticar os usuários e verificar o seu nível de acesso em *endpoints* da *Auth* API, como o *endpoint* de criação de lista curada. O *frontend*, quando necessário, envia o *token* no cabeçalho da requisição e o *backend* recebe e faz a verificação automática. Se ele não for válido, tiver expirado ou o usuário não possuir o nível de acesso demandado, uma resposta de erro será gerada.

⁷ <<https://www.mozilla.org/pt-BR/firefox/>>

Figura 18 – Nova Arquitetura de Software do OJTracker.



Fonte: o Autor

Por fim, a *Auth* API emprega um método de assinatura de JWTs que utiliza um par de chaves, composto por uma chave pública e uma chave privada. Esse processo gera uma assinatura que, além de verificar a autenticidade do JWT, certifica que o responsável por ela é, de fato, o detentor da chave privada correspondente (OKTA, 2023).

3.3 Desenvolvimento do Texto

O Overleaf⁸ foi a principal ferramenta utilizada no desenvolvimento do texto deste trabalho. Essa ferramenta é um editor LaTeX intuitivo, online e colaborativo. As entregas de texto foram incluídas nas iterações incrementais do método de desenvolvimento de software utilizado no projeto, descrito na Subseção 3.2.1.

A pesquisa bibliográfica foi conduzida por meio de uma abordagem abrangente,

⁸ <<https://pt.overleaf.com/>>

utilizando as plataformas renomadas do Google Scholar⁹ e ScienceDirect¹⁰. A utilização dessas plataformas proporcionou um acesso eficiente a uma diversidade de informações relevantes, contribuindo significativamente para a fundamentação teórica e a abordagem crítica adotada neste trabalho. O Quadro 6 apresenta outros detalhes sobre a pesquisa realizada.

Quadro 6: Detalhes da Pesquisa Bibliográfica.

Etapa	Detalhes
Termos de Busca	Termos predominantemente em inglês, com filtros específicos quando possível (ex: artigos da SBC)
Resultados Encontrados	37 resultados relevantes encontrados
Itens Consultados	28 itens de fato consultados
Critério de Escolha	Relevância, atualidade, credibilidade dos autores, conexão com os objetivos do estudo

Fonte: o Autor

3.4 Métricas

Neste trabalho, as métricas de *software* serão utilizadas para aferir a funcionalidade, identificar funções sem demanda e a usabilidade do produto desenvolvido. A usabilidade refere-se à capacidade de um *software* ser intuitivo (LIMA, 2007), garantindo que o usuário consiga alcançar seus objetivos de maneira eficiente e sem dificuldades significativas.

Segundo Toscano (2006), lapsos de tempo e tarefas sucedidas são parâmetros quantitativos de usabilidade. Para coletar dados desses aspectos, foi utilizada a ferramenta *Umami*¹¹, que é uma plataforma de análise *web* de código aberto que permite monitorar o tráfego e o desempenho de *websites* de maneira simples e eficiente. Ela também é focada em privacidade, não rastreando os dados dos usuários e sendo compatível com as regulamentações de proteção de dados (UMAMI, 2024).

As métricas coletadas serão plataforma utilizada no acesso, que inclui tipo de dispositivo e sistema operacional, ferramenta utilizada, como navegador Chrome, Firefox, entre outros, taxa de rejeição, duração de visita, páginas acessadas e eventos, redirecionamento para problema no serviço de origem. Segundo Umami (2024), a taxa de rejeição representa a porcentagem de visitantes que entram em uma página do *site* e saem sem interagir com qualquer outra página ou ação adicional.

⁹ <<https://scholar.google.com/>>

¹⁰ <<https://www.sciencedirect.com/search>>

¹¹ <<https://eu.umami.is/>>

4 Resultados

Este capítulo aborda os resultados obtidos durante o desenvolvimento de uma nova versão da plataforma OJTracker, destacando as principais alterações e funcionalidades implementadas.

A Seção 4.1 discute as melhorias realizadas na interface da aplicação, abordando questões de usabilidade. Em seguida, a Seção 4.2 apresenta detalhes sobre a implementação da autenticação, incluindo a utilização de tokens JWT e a persistência de dados de usuários. A Seção 4.3 destaca a funcionalidade de listas curadas, demonstrando como alguns usuários podem criar, editar e visualizar listas de problemas e a Seção 4.4 discorre sobre as funcionalidades implementadas para gerenciamento de usuários. Por fim, a Seção 4.5 demonstra a funcionalidades de treinamento, como a vinculação de treinadores e estudantes, entre outras.

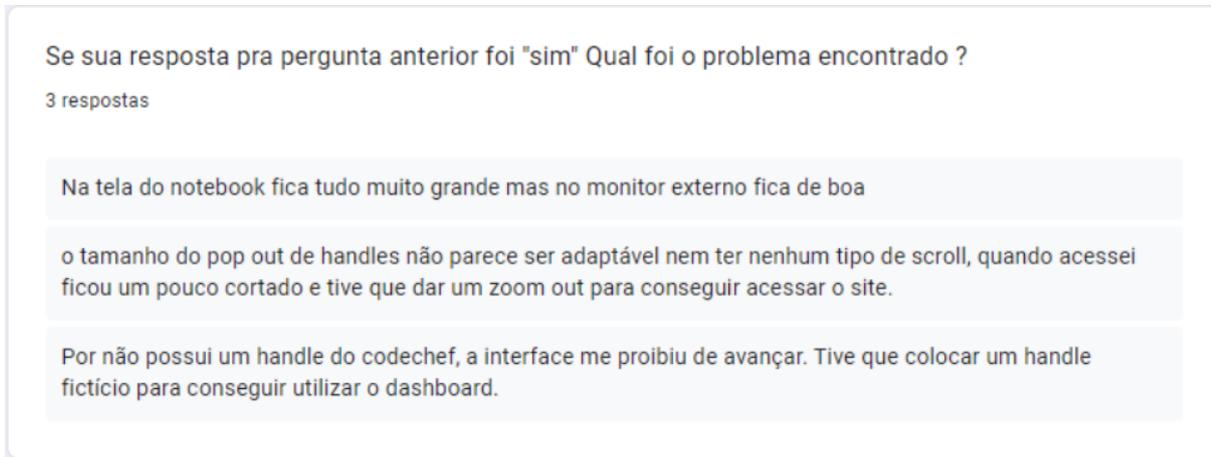
4.1 Interface

A análise da validação da primeira versão da plataforma OJTracker demonstra que alguns usuários enfrentaram problemas de usabilidade. Conforme podemos observar na Figura 19, um usuário mencionou que a exibição na tela de um *notebook* apresentava elementos muito grandes, outro usuário informou que os campos não obrigatórios do *modal* de informações de usuário bloqueavam a adição de dados se um deles não estivesse preenchido e um outro usuário completou adicionando que esse *modal* não era exibido completamente e a rolagem não era ativada na tela. Essas observações ressaltam a importância de uma abordagem mais flexível e responsiva na interface, visando proporcionar uma experiência consistente e eficaz independentemente do dispositivo utilizado para acessar à aplicação.

A Figura 20 demonstra a primeira observação. Para diminuir o tamanho dos componentes da tela e para iniciar a adição de responsividade na aplicação OJTracker, foi implementado um menu na barra do topo e o *drawer* lateral foi transformado em um *drawer* temporário em telas menores, como podemos observar na Figura 21. Nessa figura também podemos observar que o conteúdo das páginas ainda não foi refatorado e que os usuários estão conseguindo cadastrar somente um juiz *online* nas informações de usuário.

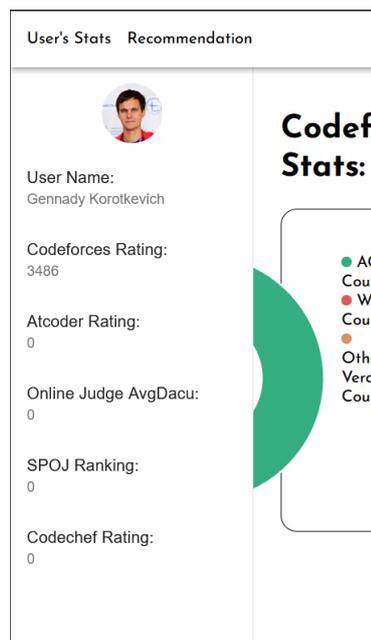
Também foi possível reproduzir a terceira observação. A Figura 22 apresenta a *modal* de informações de usuário responsiva no ambiente *desktop* e a Figura 23 no ambiente *mobile*.

Figura 19 – Respostas dos Usuários sobre a Usabilidade da Primeira Versão.



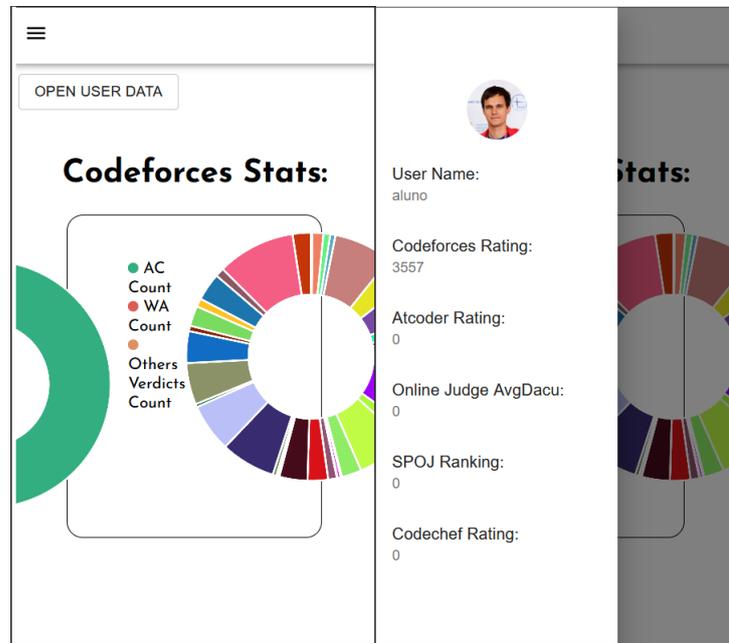
Fonte: Elaborada pelo autor com base em [Ferreira \(2023\)](#)

Figura 20 – Tela Principal em Resolução Pequena na Primeira Versão.



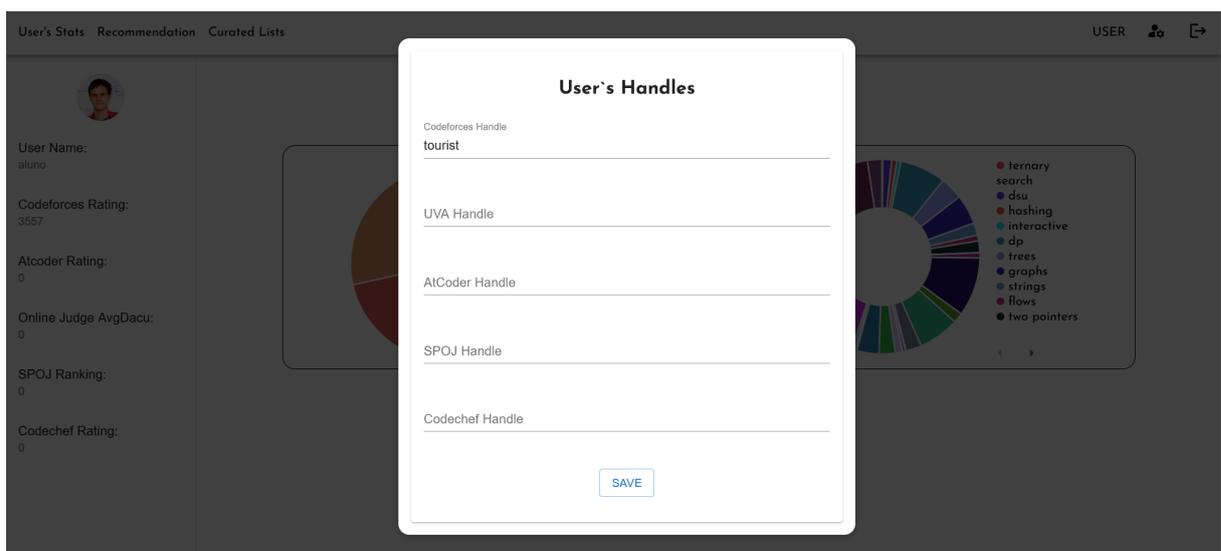
Fonte: o Autor

Figura 21 – Menu e Drawer Temporário na Nova Versão.



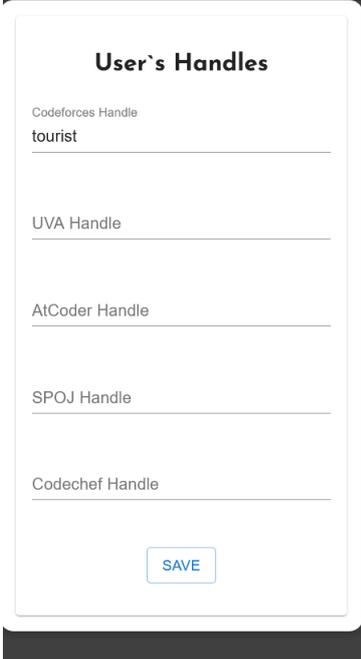
Fonte: o Autor

Figura 22 – Modal de Informações de Usuário Responsiva em Desktop.



Fonte: o Autor

Figura 23 – Modal de Informações de Usuário Responsiva em Mobile.



User's Handles

Codeforces Handle
tourist

UVA Handle

AtCoder Handle

SPOJ Handle

Codechef Handle

SAVE

Fonte: o Autor

4.2 Autenticação

A fim de proporcionar funcionalidades específicas para os técnicos, foi implementada autenticação na plataforma OJTracker. Nessa autenticação são utilizados dois tipos de concessão de autorização, a autorização por credenciais e a autorização implícita.

A autorização por credenciais ocorre na página de *login* da plataforma (veja a Figura 24). O usuário informa suas credenciais e realiza a submissão, a qual gera uma requisição para o novo serviço, Auth API. O novo serviço verifica e retorna o *status* 401 (*Unauthorized*) ou, em caso de sucesso, o 200 (*OK*) e um token JWT, conforme apresentado no Código 1.

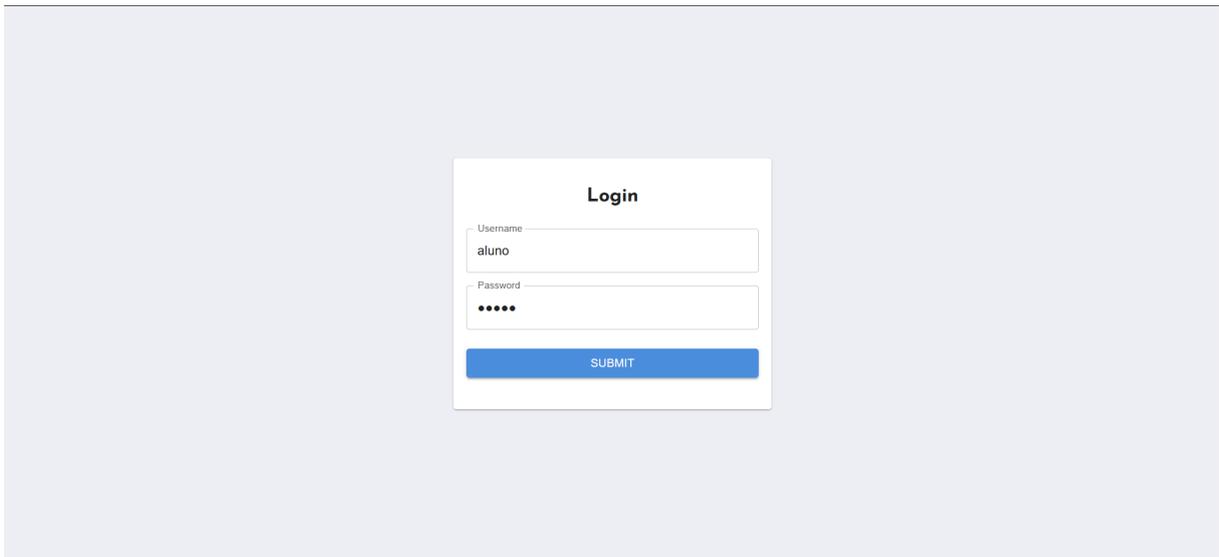
Código 1 – Retorno de Login de Sucesso.

```
1 {  
2   "token": "<token_jwt>",  
3   "profilePicURI": "",  
4   "codeforcesHandle": null,  
5   "codeforcesRanking": null,  
6 }
```

Fonte: o Autor

O JWT proporciona a autorização implícita. O formato utilizado na aplicação OJTracker do produto da decodificação desse token é apresentado no Código 2. O atributo

Figura 24 – Tela de Login do OJTracker.



Fonte: o Autor

upn, *User Principal Name*, representa o nome do usuário e o *jti*, *JWT ID*, é o identificador único do token JWT.

Código 2 – Formato do JWT Decodificado.

```
1 {
2   "typ": "JWT",
3   "alg": "RS256"
4 },
5
6 {
7   "iss": "quarkus",
8   "upn": "admin",
9   "sub": "66011985fd17587f67c97513",
10
11  "groups": [
12    "ADMIN"
13  ],
14
15  "iat": 1725245213,
16  "exp": 1725263213,
17  "jti": "ff612043-b8c6-41a7-8d92-15de9e143be6"
18 }
```

Fonte: o Autor

Foi considerado que as exigências de segurança da aplicação OJTracker eram supridas com a utilização de um banco de dados não relacional. No Código 3 podemos observar como os dados de um usuário com perfil ADMIN são persistidos no banco de dados da aplicação, o MongoDB. O atributo `_id` é automaticamente atribuído a cada documento e serve como identificador exclusivo, porém o nome de usuário também deve ser único.

Código 3 – Visualização de ADMIN no MongoDB.

```
1 oj-tracker> db.users.find({_id:ObjectId("66011985fd17587f67c97513")})
2 [
3   {
4     _id: ObjectId("66011985fd17587f67c97513"),
5     password: '$2a$10$SArIyq1Mxk.myYH2SJBfX.YD019AG4v4SMLpG5i4Ls/u9dhaZnU3u',
6     role: 'ADMIN',
7     username: 'admin',
8     author: 'admin'
9   }
10 ]
```

Fonte: o Autor

Para acomodar essa nova funcionalidade, também foi necessário fazer dois pequenos ajustes na tela principal da aplicação. Conforme podemos observar na Figura 25, no canto superior esquerdo, na barra do topo, foi adicionado um descritor para informar o tipo do usuário logado e um botão para sair da conta logada.

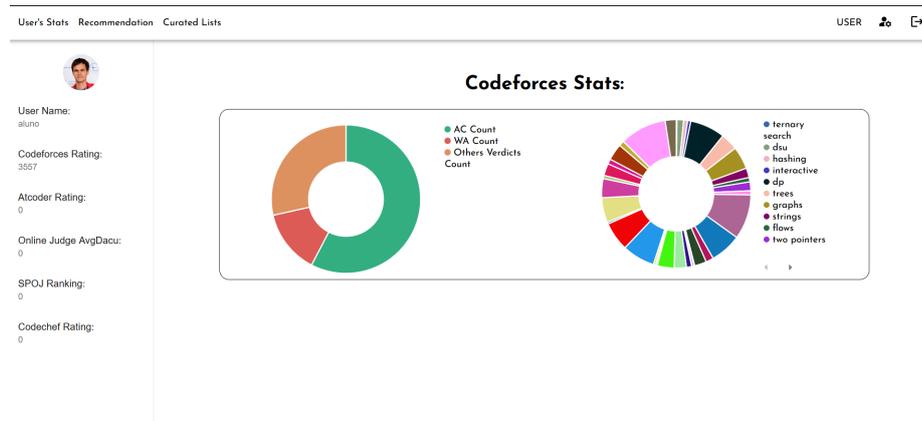
Código 4 – Visualização de Competidor no MongoDB.

```
1 oj-tracker> db.users.find({_id:ObjectId("6648254485650958474f0a19")})
2 [
3   {
4     _id: ObjectId("6648254485650958474f0a19"),
5     atcoderHandle: '',
6     atcoderProfilePicURI: '',
7     atcoderRanking: '0',
8     codechefHandle: '',
9     codechefRanking: '0',
10    codeforcesHandle: 'edsomjr',
11    codeforcesProfilePicURI: 'https://userpic.codeforces.org/no-avatar.jpg',
12    codeforcesRanking: '1907',
13    password: '$2a$10$MtOZZKbMGJTXvPwx69KfZuwAJWTtUpwyjaxZgeOG.VHY0Bh6KhT/y',
14    role: 'USER',
15    spojHandle: '',
16    spojProfilePicURI: '',
17    spojRanking: '0',
18    username: 'aluno1',
19    uvaHandle: '',
20    uvaRanking: '0'
21  }
22 ]
```

Fonte: o Autor

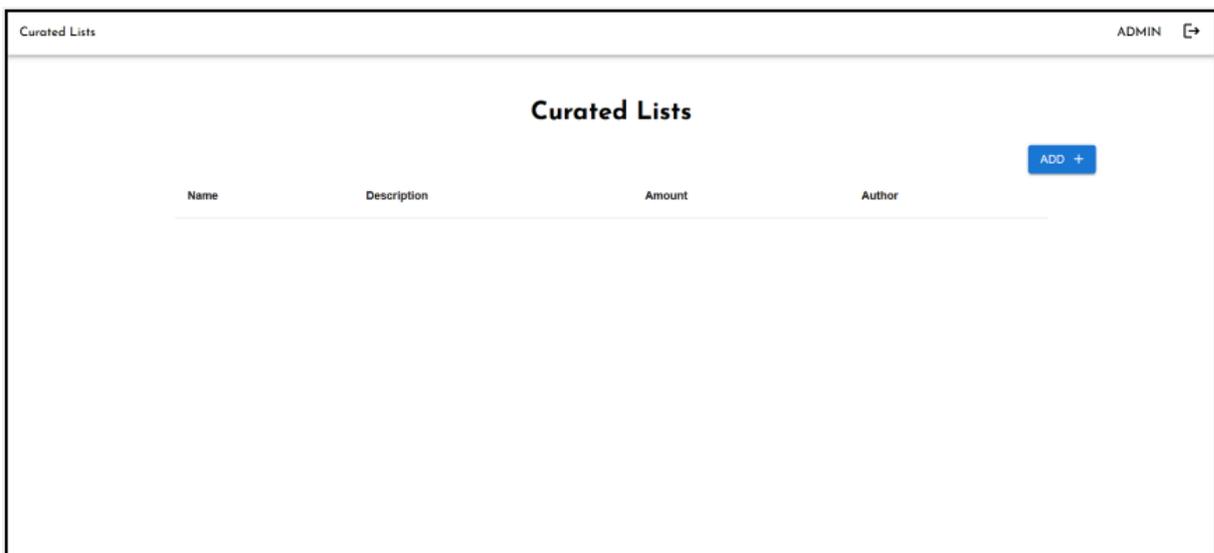
Para terminar, a autenticação também proporcionou a persistência dos dados dos usuários de perfil competidor, não somente dos dados informados como também dos dados coletados dos juízes online. O Código 4 apresenta um competidor salvo no banco de dados, é possível observar que também foram persistidos o seu *handle* e *ranking* na plataforma Codeforces, além do URL da sua foto de perfil nessa plataforma.

Figura 25 – Tela Principal com Alterações da Autenticação.



Fonte: o Autor

Figura 26 – Página de Listas Curadas.



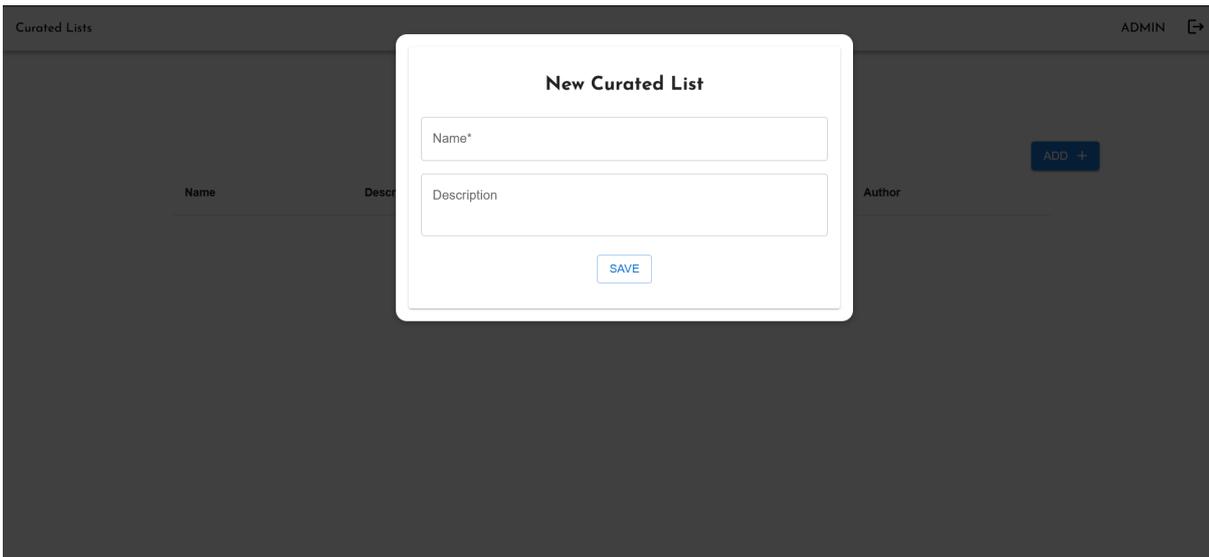
Fonte: o Autor

4.3 Listas Curadas

Ao acessarem a página de Listas Curadas, os usuários visualizaram o que está representado na Figura 26. Destaca-se que, conforme evidenciado na figura mencionada, usuários com os privilégios de perfil ADMIN e COACH têm a capacidade exclusiva de incluir novas listas na plataforma OJTracker. Quando esses usuários clicam no botão, o *modal* de nova lista curada é apresentado, como podemos visualizar na Figura 27. O Código 5 apresenta o formato da resposta da requisição de listas curadas.

Após adicionar corretamente uma lista curada, o usuário é redirecionado para a página de visualização da nova lista, conforme podemos observar na Figura 28. Na plataforma, as ações de edição, remoção e adição de problema em uma lista curada somente

Figura 27 – Adição de Nova Lista Curada.



The screenshot displays a web interface for managing curated lists. A modal window titled "New Curated List" is centered on the screen. It contains two text input fields: "Name*" and "Description". Below these fields is a "SAVE" button. The background shows a table with columns labeled "Name", "Description", and "Author". To the right of the table, there is an "ADD +" button. The top right corner of the interface shows "ADMIN" and a user icon.

Fonte: o Autor

podem ser realizadas por um usuário de perfil ADMIN ou pelo COACH autor da lista. O *modal* de edição de lista curada é apresentado na Figura 29.

Código 5 – Resposta da Requisição de Listas.

```
1 [
2   0: {
3     "id": "656f6ce3797421231d1d5b7e",
4     "name": "Lista Curada Teste",
5     "description": "Descrição Lista Curada Teste",
6     "amount": 0,
7     "author": "admin"
8   }
9 ]
```

Fonte: o Autor

Para adicionar um problema na lista, o usuário deve informar o identificador e o juiz online do problema, como podemos observar na Figura 30. Atualmente, a adição de problema ocorre em duas etapas, pesquisa de problema no juiz online e persistência do problema na lista. Essa estratégia de implementação é semelhante com a estratégia utilizada pelo Virtual Judge.

A Figura 31 apresenta um gráfico de comparação dos tempos de cada etapa para a adição de alguns problemas. Esses problemas ainda não tinham sido cadastrados em nenhuma outra lista curada, mas podemos observar que mesmo assim há espaço para otimização do seu tempo de pesquisa.

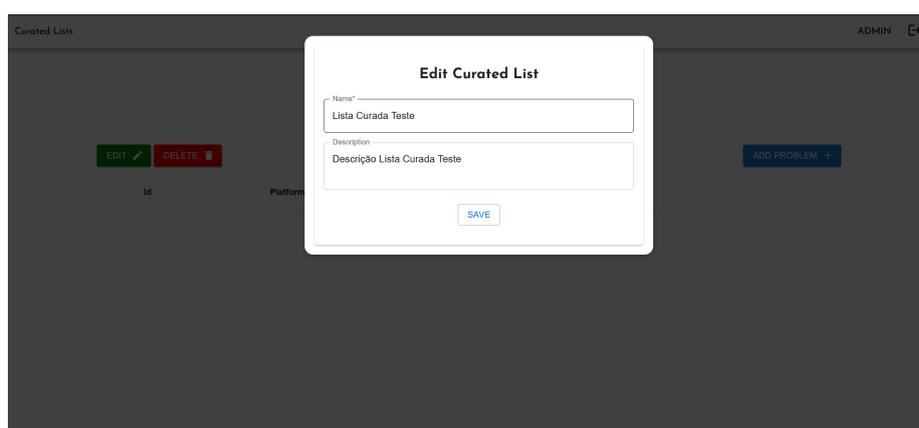
A tabela preenchida com os problemas adicionados pode ser observada na Figura

Figura 28 – Página de Visualização de Lista Curada.



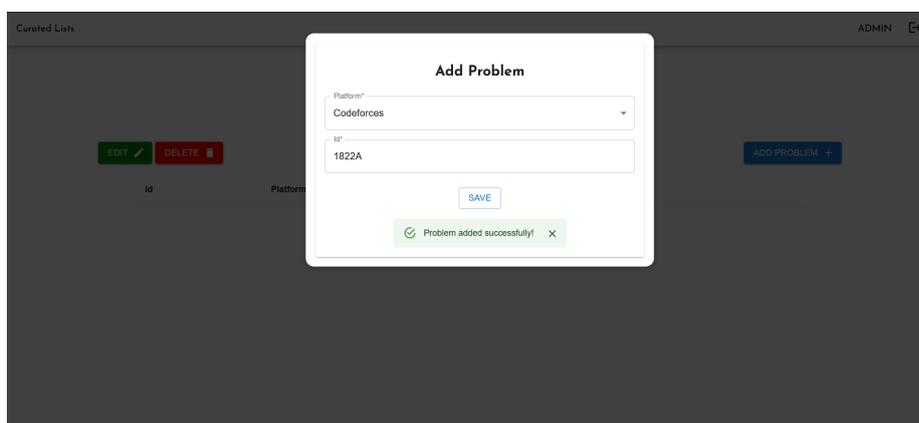
Fonte: o Autor

Figura 29 – Edição de Lista Curada.



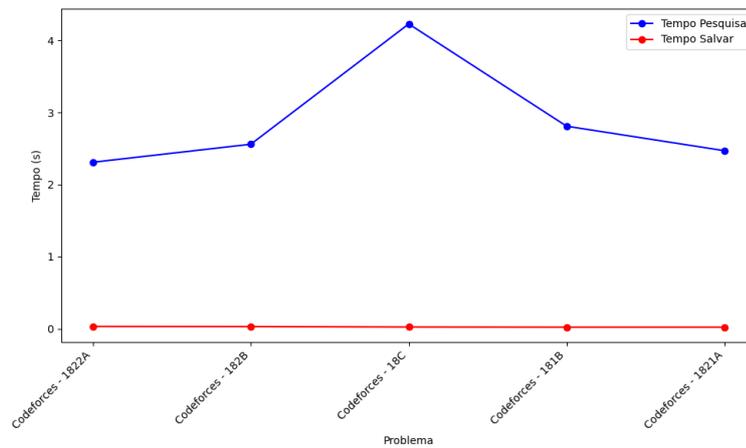
Fonte: o Autor

Figura 30 – Adição de Problema na Lista Curada.



Fonte: o Autor

Figura 31 – Comparação de Tempos das Etapas de Adição de Problema.



Fonte: o Autor

Figura 32 – Visualização de Problemas.

Curated Lists ADMIN ↗

Lista Curada Teste

Descrição Lista Curada Teste

EDIT ✎
DELETE 🗑️
ADD PROBLEM +

Id	Platform	Name	
1822A	CODEFORCES	TubeTube Feed	🗑️
182B	CODEFORCES	Vasya's Calendar	🗑️
18C	CODEFORCES	Stripe	🗑️
181B	CODEFORCES	Number of Triplets	🗑️
1812A	CODEFORCES	Are You a Robot?	🗑️

Fonte: o Autor

32, e o formato da resposta da requisição de uma lista curada na aplicação OJTracker é apresentado no Código 6. A regra aplicada à remoção de um problema é a mesma aplicada aos botões de ação da lista.

Código 6 – Resposta da Requisição de Lista Curada.

```

1 {
2   "id": "660e24280712d217d026f343",
3   "name": "Title",
4   "description": "Descrição",
5   "amount": "5",
6   "author": "coach",
7   "problems": [

```

```
8     {
9         "problemId": "660efea3958acd55ef7c7397",
10        "externalId": "FOODCOST",
11        "platform": "CODECHEF",
12        "name": "Food Costs",
13        "link": "https://www.codechef.com/problems/FOODCOST"
14    },
15    {
16        "problemId": "660f019e958acd55ef7c7399",
17        "externalId": "caddi2018_c",
18        "platform": "ATCODER",
19        "name": "Negative Doubling",
20        "link": "https://atcoder.jp/contests/caddi2018/taskscaddi2018_c"
21    },
22    {
23        "problemId": "660f6640958acd55ef7c739b",
24        "externalId": "124B",
25        "platform": "CODEFORCES",
26        "name": "Permutations",
27        "link": "https://codeforces.com/problemset/problem/124/B"
28    },
29    {
30        "problemId": "6659606e8754e1259bb6e37c",
31        "externalId": "103",
32        "platform": "UVA",
33        "name": "Stacking Boxes",
34        "link": "https://onlinejudge.org/external/1/103.pdf"
35    },
36    {
37        "problemId": "665be4448dc7e53fa205289c",
38        "externalId": "TEST",
39        "platform": "SPOJ",
40        "name": "Life, the Universe, and Everything",
41        "link": "https://www.spoj.com/problems/TEST"
42    }
43 ]
44 }
```

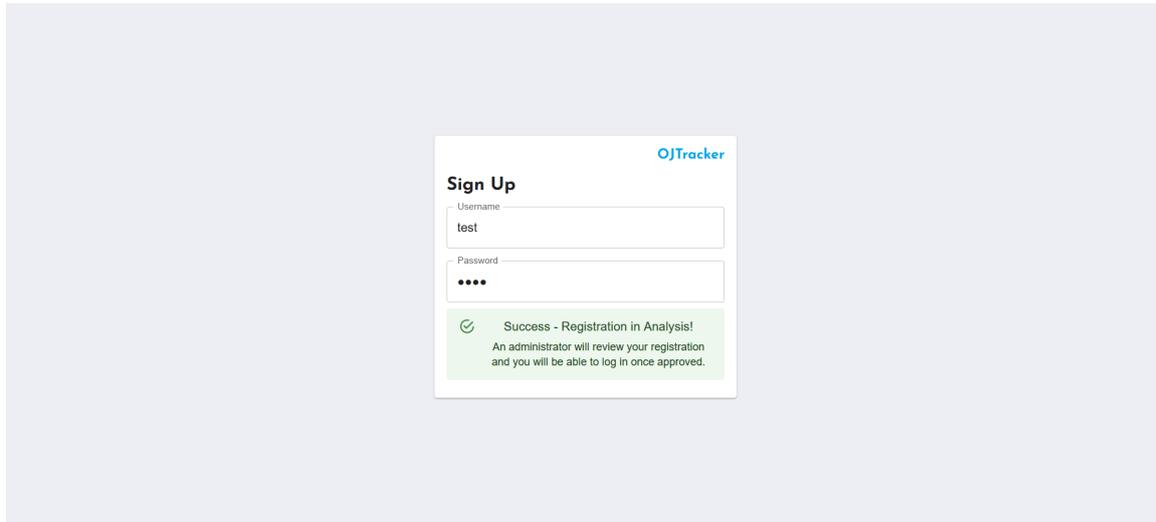
Fonte: o Autor

Por fim, no banco de dados MongoDB, a lista curada e seus problemas são persistidos no formato $n \times n$, permitindo uma associação muitos para muitos entre eles. Essa característica possibilita que um mesmo problema seja atribuído a mais de uma lista, eliminando a redundância de dados e promovendo a normalização.

4.4 Gerenciamento de Usuários

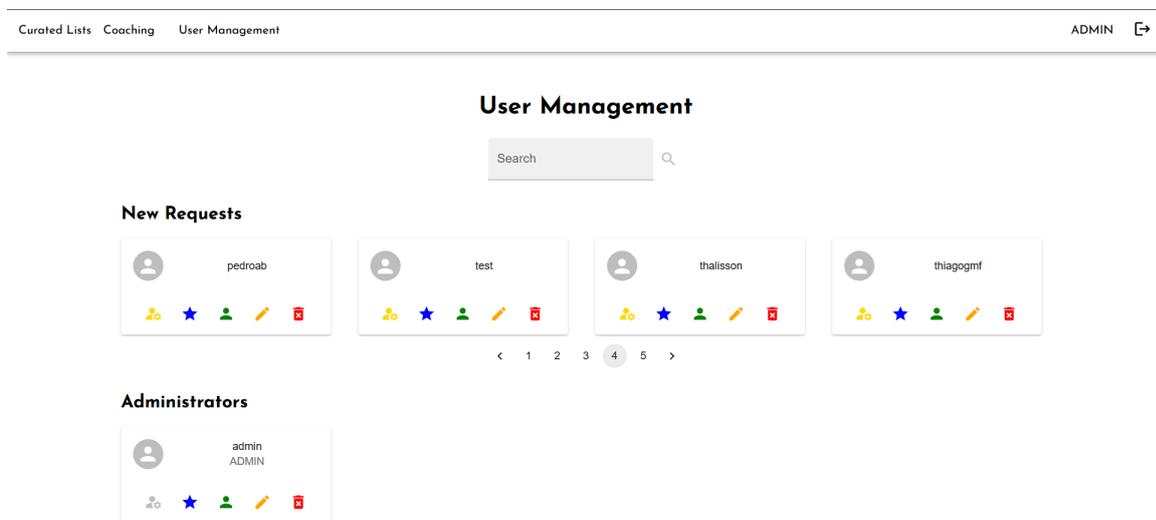
O cadastro de um novo usuário é realizado em uma tela auxiliar da tela de *login*. Como podemos visualizar na Figura 33, quando um usuário preenche um nome de usuário e senha válidos, é pré-cadastrado na plataforma e recebe um aviso de que seu cadastro, para ser efetivado, será analisado por um administrador.

Figura 33 – Página de Cadastro.



Fonte: o Autor

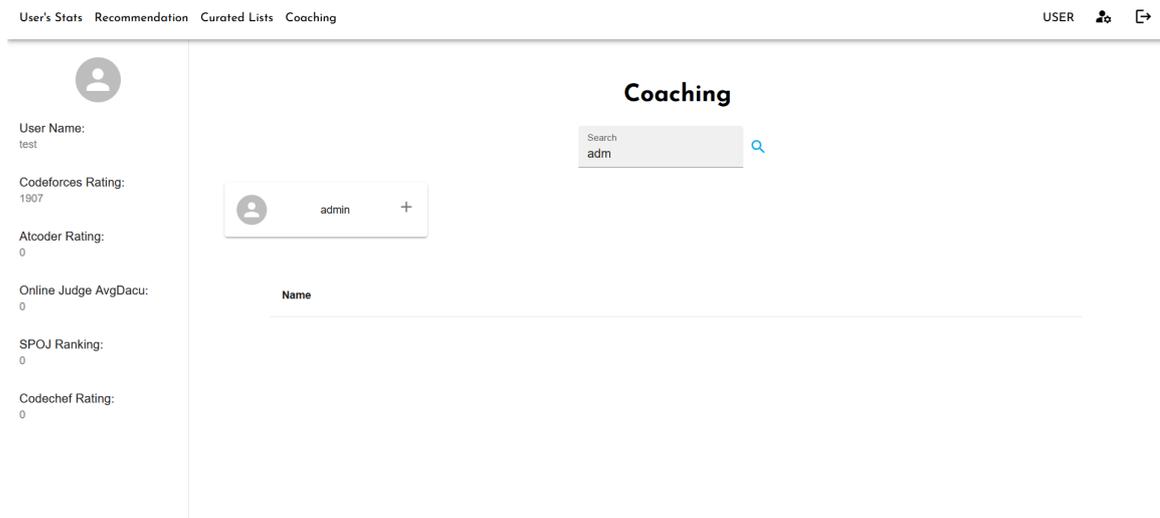
Figura 34 – Tela de Gerenciamento de Usuários.



Fonte: o Autor

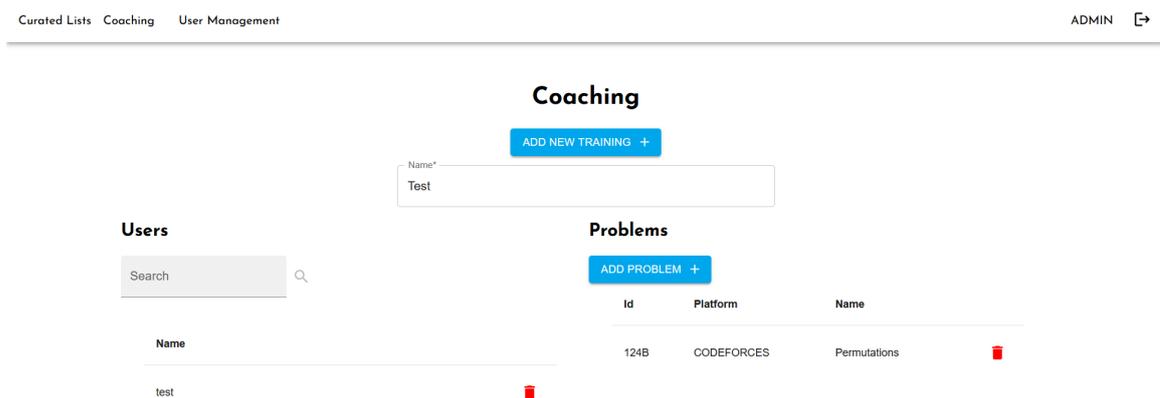
Após o pré-cadastro, o usuário será listado na aba de *New Requests* da página de gerenciamento de usuários. Nessa página, conforme podemos observar na Figura 34, para efetivar o usuário, um administrador deve selecionar sua *role*, ou, para negar seu cadastro, excluir sua solicitação. No gerenciamento de usuários também é possível alterar a *role* e excluir a conta de usuários efetivados, além de proporcionar a edição de nome de usuário e senha.

Figura 35 – Tela de Treinamento de USER.



Fonte: o Autor

Figura 36 – Tela de Criação de Treinamento.



Fonte: o Autor

4.5 Treinamento

O treinamento ocorre entre usuários de perfil USER e usuários de perfil ADMIN ou COACH. Essa relação é concretizada na tela de *coaching* dos usuários comuns, apresentada na Figura 35, onde o usuário pode pesquisar por novos treinadores e remover treinadores adicionados.

A criação de um treinamento ocorre em uma tela auxiliar da página de *coaching* na visão de um usuário especial. Como podemos visualizar na Figura 36, um usuário de perfil COACH ou ADMIN pode definir um nome para o novo treinamento, selecionar usuários, do grupo de usuários que o escolheram como treinador, e selecionar problemas,

Figura 37 – Tela de Treinadores.

The screenshot shows the 'Coaching' page for user 'test'. The sidebar on the left displays the following statistics:

- User Name: test
- Codeforces Rating: 1907
- Atcoder Rating: 0
- Online Judge AvgDacu: 0
- SPOJ Ranking: 0
- Codechef Rating: 0

The main content area is titled 'Coaching' and shows the name 'admin'. Below this is a table with the following data:

Index	Disclosure Link	Problems Number
#1	Test	1

Fonte: o Autor

Figura 38 – Tela de Problemas do Treinamento.

The screenshot shows the 'Coaching' page for user 'test'. The sidebar on the left displays the same statistics as in Figure 37. The main content area is titled 'Coaching' and shows the name '#1 - Test - admin'. Below this is a table with the following data:

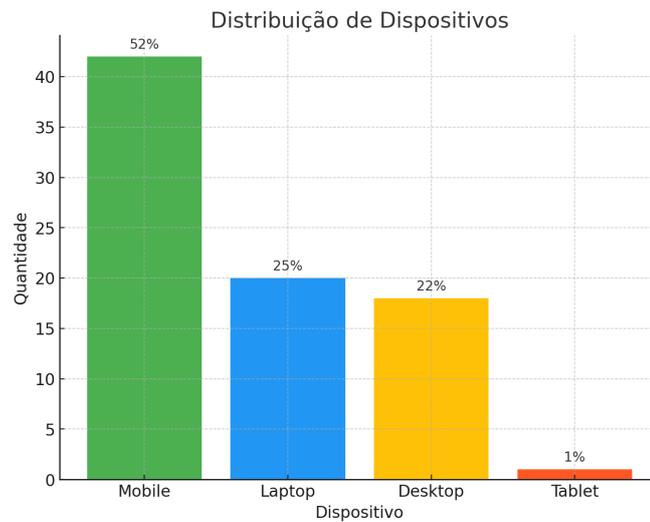
Id	Platform	Name
124B	CODEFORCES	Permutations

Fonte: o Autor

como são selecionados nas listas curadas, vide Figura 27.

Após criado, quando o usuário clicar no treinador autor na sua lista de treinadores, esse treinamento será listado, conforme é apresentado na Figura 37. Na Figura 38, é demonstrada a visualização dos problemas de um treinamento. Por fim, as ações de um treinamento, edição e deleção, seguem as regras aplicadas às listas curadas, estão disponíveis para os usuários ADMIN e o usuário COACH autor.

Figura 39 – Dispositivos Utilizados



Fonte: o Autor

4.6 Métricas

A coleta das métricas de software da nova versão da aplicação OJTracker ocorreu de 31 de Julho de 2024 a 31 de Agosto de 2024. Durante esse período, a infraestrutura energética da Universidade enfrentou alguns problemas, com recorrentes e longevas quedas de energia. Essa situação impediu que a aplicação estivesse disponível na *internet* durante a maior parte do período de coleta, conseqüentemente foram coletadas poucas medidas e não foi possível coletar eventos.

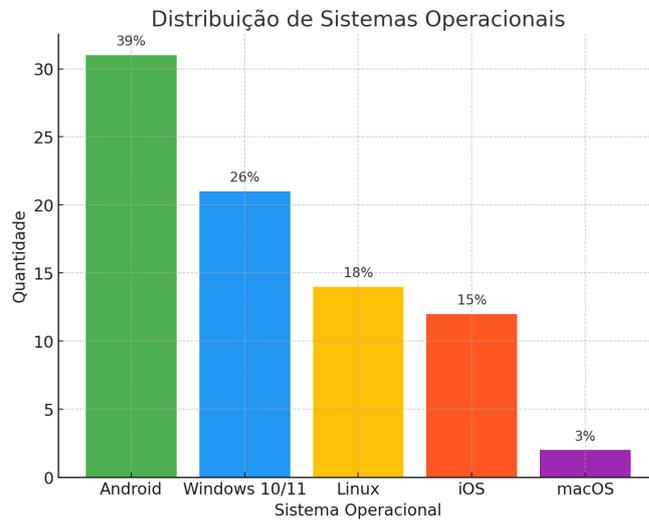
No decorrer da coleta de métricas, a plataforma foi divulgada em grupos de mensagens compostos por interessados em programação competitiva, como estudantes, professores, entre outros. Ao total, foram realizados 18 cadastros de usuários comuns na aplicação. Essa amostra é composta principalmente por competidores iniciantes.

Na Figura 39, podemos visualizar os dispositivos utilizados no acesso à plataforma OJTracker. Observando também a Figura 40, que apresenta os sistemas operacionais utilizados nesse acesso, percebemos que os acessos iniciais dos usuários são preferencialmente realizados a partir de equipamentos móveis. Isso indica a necessidade da finalização do refino da interface, garantindo total responsividade na plataforma.

A taxa de rejeição foi de 54% e a duração de visita foi de dois minutos e quatro segundos. Para uma plataforma interativa, esses valores são considerados negativos, porém eles foram prejudicados pela baixa disponibilidade da plataforma, que impediu que os usuários evoluíssem no seu uso. Além disso, como podemos observar na Figura 41, a ferramenta mais utilizada no acesso foi o navegador Chrome.

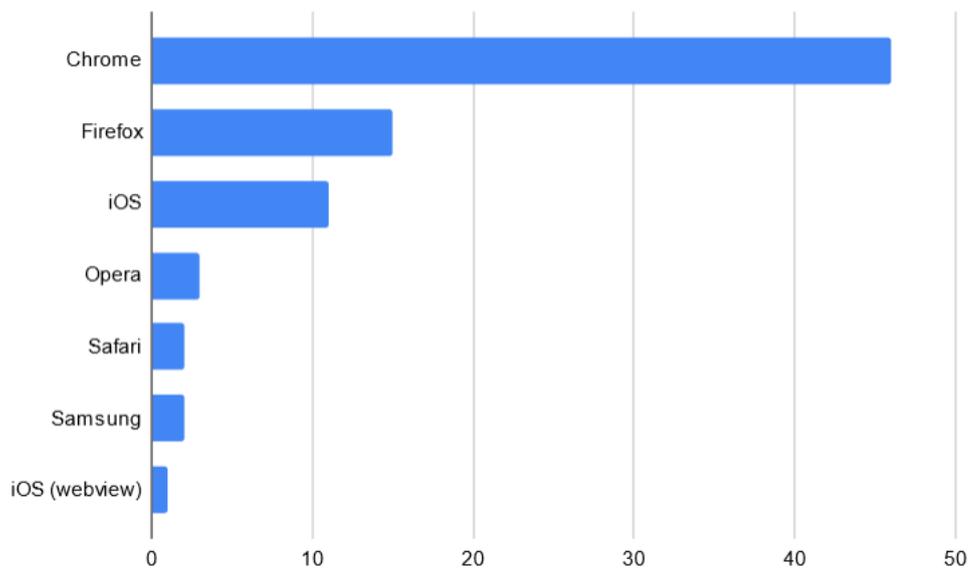
Por fim, a Figura 42 apresenta as páginas relevantes mais acessadas. A página de

Figura 40 – Sistemas Operacionais Utilizados



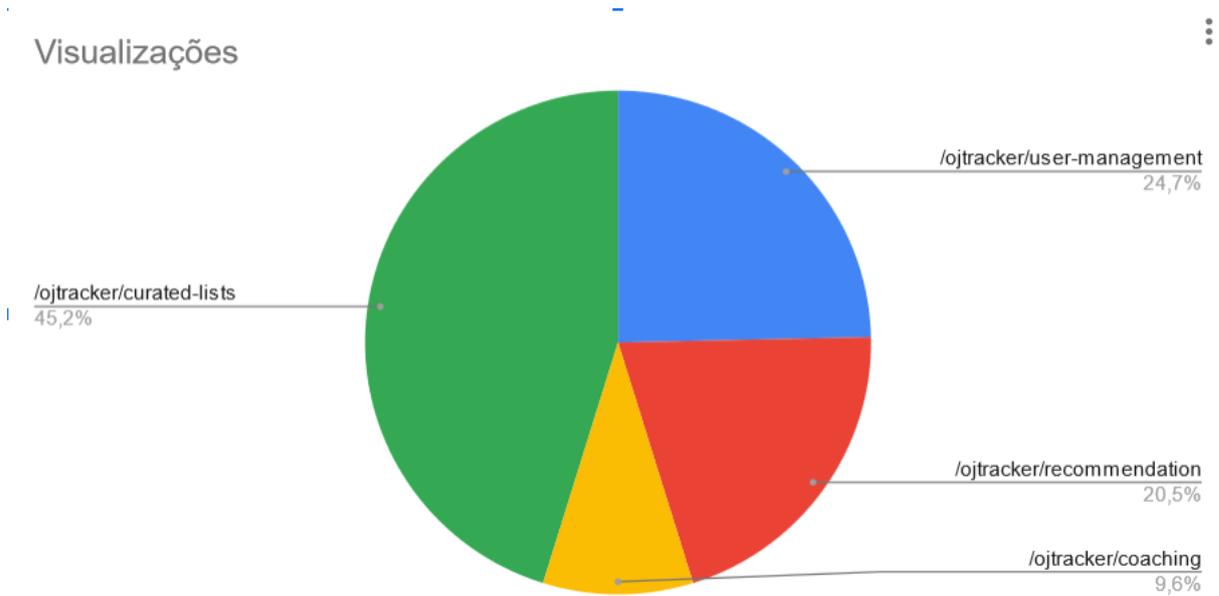
Fonte: o Autor

Figura 41 – Ferramentas Utilizadas



Fonte: o Autor

Figura 42 – Páginas Acessadas



Fonte: o Autor

gerenciamento de usuários (`/ojtracker/user-management`) possui uma duração de visita de dois minutos e nove segundos, como essa página foi utilizada consistentemente na ativação de novos usuários, essa medida indica que a sua interface é de fácil manejo. Outro destaque foi o conjunto de páginas de recomendação, que não foram as mais acessadas, mas possuem os maiores valores de duração de visita.

5 Considerações Finais

O presente trabalho pretendia evoluir a aplicação OJTracker, focando em soluções para técnicos. Seu objetivo geral era adicionar suporte aos técnicos de programação competitiva, além de ampliar as opções dos usuários na plataforma. Para avaliar o quanto desse objetivo foi satisfeito ao término do desenvolvimento, foram elencados quatro objetivos específicos.

O primeiro objetivo específico foi cumprido integralmente. Na nova versão da plataforma OJTracker, usuários credenciados possuem a possibilidade de elaborar treinamentos de programação competitiva para usuários específicos e listas curadas de problemas, utilizando-se de materiais disponíveis nas plataformas já integradas na aplicação, como AtCoder e Codeforces.

O desenvolvimento de novos métodos automatizados de recomendação de problemas, o segundo objetivo específico, não foi realizado. Como foi dito anteriormente, o foco do trabalho eram as funcionalidades relacionadas aos técnicos, que de certa forma também ampliam as opções dos usuários na aplicação, por isso, na priorização do desenvolvimento, esse objetivo foi preterido. Dessa forma, os requisitos elicitados na Seção 3.1 do épico de Recomendação são indicados para futuras evoluções.

A nova versão do OJTracker também implementa funcionalidades de autenticação, *login*, cadastro, gerenciamento de usuários, entre outras. O refino da interface também foi iniciado, mas não concluído, devido a falta de tempo hábil para todas as alterações necessárias. Sendo assim, o terceiro objetivo específico, realizar as melhorias necessárias para os novos recursos, foi concluído parcialmente.

Por fim, a nova versão da plataforma foi implantada com sucesso na infraestrutura própria da universidade, concluindo o quarto objetivo específico. Durante o processo de implantação foi identificado que futuramente pode-se adicionar ao servidor a possibilidade de realizar *scrapping* que necessita de um *driver* de navegador, permitindo a automatização da atualização dos materiais e dados do Codechef.

5.1 Trabalhos Futuros

Este trabalho e o trabalho desenvolvido por [Ferreira \(2023\)](#) identificam algumas possíveis evoluções futuras. Destacam-se: otimização da consulta de problemas que ainda não foram adicionados à aplicação, novos métodos automatizados de recomendação de problemas, completo refinamento da interface e implementação própria de coleta de métricas de *software*.

Referências

- ATCODER. Atcoder. 2023. Acesso em: 26 nov. 2023. Disponível em: <<https://atcoder.jp/>>. Citado 4 vezes nas páginas 33, 34, 35 e 36.
- C2LADDER. C2 ladders. 2023. Acesso em: 21 nov. 2023. Disponível em: <<https://c2-ladders-juol.onrender.com/>>. Citado 3 vezes nas páginas 23, 24 e 25.
- CP-ACADEMY. Cp academy. 2023. Acesso em: 23 nov. 2023. Disponível em: <<https://cp-academy.netlify.app/main.html>>. Citado 2 vezes nas páginas 26 e 27.
- DONEGAN, P. et al. Métricas de software: Um mapeamento entre six sigma e cmmi. *SIMPROS*, 2005. Citado na página 39.
- DÍEZ, B. S. M. *Detecció de plagi a Judge.org*. Trabalho de Conclusão de Curso — Universitat Politècnica de Catalunya Barcelona Tech - UPC Facultat d'Informàtica de Barcelona – FIB, 2018. Citado na página 38.
- FERREIRA, T. G. M. *OJTracker: Uma aplicação para acompanhamento pessoal em juízes online*. Trabalho de Conclusão de Curso — Universidade de Brasília – UnB Faculdade UnB Gama – FGA, 2023. Citado 6 vezes nas páginas 21, 22, 39, 41, 48 e 65.
- FLORES, G. M. A olimpíada brasileira de informática: Uma análise comparativa entre as escolas que tiveram alunas premiadas e aquelas nas quais apenas alunos foram premiados entre 2007 e 2013. 2023. Disponível em: <<https://www.prp.unicamp.br/inscricao-congresso/resumos/2023P22386A39103O6209.pdf>>. Citado na página 21.
- IQBAL, T. Requirement elicitation technique: - a review paper. 2014. Disponível em: <https://www.researchgate.net/profile/Mohammad-Suaib/publication/272944882_Requirement_Elicitation_Technique_-_A_Review_Paper/links/54f412140cf2f9e34f08d5b2/Requirement-Elicitation-Technique-A-Review-Paper.pdf>. Citado na página 41.
- JULCA-MEJIA, W.; PAUCAR-CURASMA, H. A cloud based recommender system for competitive programming platforms with machine and deep learning. 2023. Disponível em: <<https://sol.sbc.org.br/index.php/ctrl/article/view/25779/25595>>. Citado na página 38.
- JUTGE. Judge. 2023. Acesso em: 26 nov. 2023. Disponível em: <<https://judge.org/>>. Citado 3 vezes nas páginas 36, 37 e 38.
- LIMA, A. P. dos R.; AMARAL, D. C. O que fazer quando não há product owner? um estudo sobre a ausência desse papel. *Revista de Gestão e Projetos*, 2022. ISSN 2236-0972. Disponível em: <<https://periodicos.uninove.br/gep/article/view/22209/9745>>. Citado na página 43.
- LIMA, S. T. de. *AVALIAÇÃO DA ACESSIBILIDADE DE SÍTIOS WEB POR MEIO DE MÉTRICAS DE SOFTWARE*. Tese (Doutorado) — Universidade Católica de Brasília, 2007. Disponível em: <<https://bdtd.ucb.br:8443/jspui/handle/123456789/1507>>. Citado na página 46.

- LIU, K. et al. Who judges the judge: An empirical study on online judge tests. 2023. Disponível em: <https://chenzhenpeng18.github.io/papers/ISSTA23_OJ.pdf>. Citado na página 33.
- MANI, A. et al. Better feedback for educational online judges. 2014. Disponível em: <<https://upcommons.upc.edu/bitstream/handle/2117/28174/csedu-2014.pdf?sequence=1&isAllowed=y>>. Citado na página 33.
- MILLS, E. E. Software metrics. *International Journal of Computer Applications*, December 1988. Disponível em: <<https://apps.dtic.mil/sti/pdfs/ADA236140.pdf>>. Citado na página 39.
- OKESOLA, O. J. et al. Qualitative comparisons of elicitation techniques in requirement engineering. 2019. Disponível em: <<https://ir.tech-u.edu.ng/842/1/QUALITATIVE%20COMPARISONS%20OF%20ELICITATION%20TECh.pdf>>. Citado na página 41.
- OKTA, A. by. Introduction to json web tokens. 2023. Acesso em: 17 nov. 2023. Disponível em: <<https://jwt.io/introduction>>. Citado 2 vezes nas páginas 44 e 45.
- PETIT, J. et al. Judge.org: Characteristics and experiences. 2017. Disponível em: <<https://upcommons.upc.edu/bitstream/handle/2117/110183/Judge.org.pdf>>. Citado na página 38.
- RAHMAN, M. M. et al. Exploring automated code evaluation systems and resources for code analysis: A comprehensive survey. 2023. Disponível em: <<https://arxiv.org/pdf/2307.08705.pdf>>. Citado na página 30.
- SACHDEVA, S. Scrum methodology. *International Journal of Engineering and Computer Science*, v. 5, n. 6, p. 16792–16799, 2016. ISSN 2319-7242. Citado na página 43.
- SBC. Maratona sbc de programação. 2023. Acesso em: 10 nov. 2023. Disponível em: <<https://maratona.sbc.org.br/sobre/index.html>>. Citado na página 21.
- SHARMA, S.; PANDEY, S. K. Revisiting requirements elicitation techniques. 2013. Disponível em: <<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=692b30d03ab7f1fd797327d26d2392008b8e3a93>>. Citado na página 41.
- SIDDIQUI, M. A.; AGARWAL, H. Recommendation engine for competitive coding questions using restricted boltzmann machines, a hybrid approach. *International Journal of Advanced Research in Computer Science*, IJARCS, v. 9, n. 1, January-February 2018. ISSN 0976-5697. Technical Note. Citado 2 vezes nas páginas 38 e 39.
- THORAT, P. B.; GOUDAR, R. M.; BARVE, S. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, v. 110, n. 4, January 2015. ISSN 0975–8887. Citado na página 39.
- TIWARI, S.; RATHORE, S. S. A methodology for the selection of requirement elicitation technique. 2017. Disponível em: <<https://arxiv.org/pdf/1709.08481.pdf>>. Citado na página 41.
- TOSCANO, W. Métricas de usabilidade. *Exacta*, v. 4, n. Esp, p. 79–80, November 2006. ISSN 1678-5428. Disponível em: <<https://www.redalyc.org/pdf/810/81009912.pdf>>. Citado na página 46.

UMAMI. Umami. 2024. Acesso em: 14 agos. 2024. Disponível em: <<https://umami.is/>>. Citado na página 46.

VIANNA, A. C. G. et al. Pogramação competitiva - a união entre a matemática e computação. 2020. Disponível em: <<https://prp.usp.br/wp-content/uploads/sites/653/2020/06/Programa%C3%A7%C3%A3o-competitiva-a-uni%C3%A3o-entre-a-Matem%C3%A1tica-e-Computa%C3%A7%C3%A3o.pdf>>. Citado na página 21.

VIEIRA, F. J. R.; NUNES, M. A. S. N. Dica: Sistema de recomendação de objetos de aprendizagem baseado em conteúdo. *SCIENTIA PLENA*, v. 8, n. 5, 2012. Disponível em: <<https://scientiaplena.org.br/sp/article/view/464/567>>. Citado na página 39.

VJUDGE. Virtual judge. 2023. Acesso em: 24 nov. 2023. Disponível em: <<https://vjudge.net/>>. Citado 3 vezes nas páginas 28, 29 e 30.

WASIK, S. et al. A survey on online judge systems and their applications. 2017. Disponível em: <<https://arxiv.org/pdf/1710.05913.pdf>>. Citado 3 vezes nas páginas 30, 31 e 32.

WASIK, S. et al. Evaluation as a service architecture and crowdsourced problems solving implemented in optil.io platform. 2018. Disponível em: <<https://arxiv.org/pdf/1807.06002.pdf>>. Citado na página 32.

WATANOBE, Y. et al. Online judge system: Requirements, architecture, and experiences. 2022. Disponível em: <<https://www.worldscientific.com/doi/epdf/10.1142/S0218194022500346>>. Citado 2 vezes nas páginas 30 e 31.

A Deploy

Este apêndice compreende os passos seguidos para realização do *deploy* da plataforma OJTracker na infraestrutura da universidade.

Inicie adicionando o arquivo `000-ojtracker.conf`, arquivo de configuração da aplicação *web* no Apache¹, no diretório `/etc/apache2/sites-available/` com o seguinte conteúdo, verifique o valor de `ServerName`:

Código 7 – Início da Configuração do Apache.

```
1 <VirtualHost *:80>
2     ServerAdmin webmaster@localhost
3     ServerName unboca.naquadah.com.br
4 </VirtualHost>
```

Fonte: o Autor

Após, habilite módulos necessários e habilite a nova aplicação no Apache executando os seguintes comandos no terminal:

Código 8 – Habilitação de Módulos do Apache.

```
1 a2enmod proxy
2 a2enmod proxy_fcgi
3 a2enmod proxy_http
4 systemctl restart apache2
5 a2ensite 000-ojtracker
6 systemctl reload apache2
```

Fonte: o Autor

Ainda no servidor, instale o Docker², `snap install docker`, e adicione e execute o MongoDB, altere os parâmetros se necessário:

Código 9 – Execução Docker MongoDB.

```
1 docker run -d \
2     --name mongodb --restart always -p 27017:27017 \
3     -v ./data:/data/db mongo
```

Fonte: o Autor

Para popular o banco criado com problemas do CodeChef, no banco de origem dos dados, faça um *mongodump*, observe que os valores entre `<>` precisam ser substituídos:

¹ <https://httpd.apache.org/>

² <https://www.docker.com/>

Código 10 – Execução Docker *mongodump*.

```
1 docker exec -it <container> \  
2     mongodump --db oj-tracker --collection 'codechef-problems' --out /backup
```

Fonte: o Autor

Nesse caso, o comando *mongodump* irá criar um arquivo dentro do *container*. Copie o arquivo gerado para o ambiente externo, `docker cp <container>:/backup .`, e envie para o servidor:

Código 11 – Copiando para o servidor.

```
1 scp -J <user>@<address>:<port> -P <port> -r backup/ \  
2     <user>@<address>:~/ojtracker/codechef-api-aux/cp_backup
```

Fonte: o Autor

Em `~/ojtracker/codechef-api-aux/`, importe os dados para o banco destino:

Código 12 – Importando dados para Docker MongoDB.

```
1 docker exec -it <container> \  
2     mongorestore --db oj-tracker --collection 'codechef-problems' \  
3     cp_backup/oj-tracker/codechef-problems.bson
```

Fonte: o Autor

Atualize o arquivo `~/.bashrc`, com o seguinte conteúdo, e execute o comando `source ~/.bashrc` para solidificar as configurações e gere as chaves:

Código 13 – Conteúdo `.bashrc`.

```
1 // CodeChef  
2 export MONGODB_CONNECTION_STRING=mongodb://localhost:27017  
3 export OJTRACKER_DATABASE=oj-tracker  
4 export CHROMEDRIVER_PATH=path  
5  
6 // Auth  
7 export OJTRACKER_PRIVATE_KEY=~/ojtracker/auth-api-aux/priv.pem  
8 export OJTRACKER_PUBLIC_KEY=~/ojtracker/auth-api-aux/pub.pem  
9 export OJTRACKER_ORIGINS=*
```

Fonte: o Autor

A.1 APIs

No diretório do repositório do projeto de uma das APIs, gere o pacote do projeto com o comando `./mvnw package`, e realize a cópia da pasta gerada `target/quarkus-app/` para o servidor:

Código 14 – Copiando pacote para o servidor.

```
1 scp -J <user>@<address>:<port> -P <port> -r target/quarkus-app/ \  
2 <user>@<address>:~/ojtracker/<codeforces>-api/
```

Fonte: o Autor

No servidor, atualize o arquivo de configuração da aplicação no Apache, executando após o comando `systemctl reload apache2`:

Código 15 – Configuração de Aplicação no Apache.

```
1 ProxyPreserveHost On  
2 ProxyPass /api/<codeforces> http://localhost:<8080>/  
3 ProxyPassReverse /api/<codeforces> http://localhost:<8080>/
```

Fonte: o Autor

Por fim, execute a aplicação em segundo plano:

Código 16 – Execução de Aplicação em Segundo Plano.

```
1 nohup java -jar <codeforces>-api/quarkus-run.jar &
```

Fonte: o Autor

A.1.1 Primeiro ADMIN

Depois de realizar o *deploy* da API Auth, faça um *POST*, informando os campos *username* e *password*, no *endpoint* `auth/api/users`. Tendo sucesso, acesse o *mongosh*, o *bash* do *container* do MongoDB, `docker exec -it mongodb mongosh`, entre no banco da aplicação, use `oj-tracker`, e visualize o usuário e sua requisição, `db.users.find()` e `db['userRequests'].find()`, respectivamente.

Visualizando o usuário e sua requisição, realize sua aprovação e o transforme em um usuário ADMIN com os comandos apresentados no Código 17, que devem ser utilizados em um *shell* MongoDB, como o do Docker MongoDB. O primeiro comando altera o *status* da requisição do usuário, aprovando-o, e o segundo atribui 'ADMIN' como valor do seu atributo *role*, uma *flag* que indica que o usuário compõe o grupo de administradores da plataforma.

Código 17 – Aprovar e Definir Usuário como ADMIN.

```
1 db['userRequests'].updateOne(  
2   {userId: ObjectId('<id>')}, {$set: { status: 'APPROVED'}}  
3 )  
4 db.users.updateOne(  
5   {_id: ObjectId('<id>')}, {$set: { role: 'ADMIN'}}  
6 )
```

Fonte: o Autor

A.2 FRONTEND

Atualize o arquivo `.env` do projeto do *frontend*, adicionando os valores adequados para o contexto e execute `docker exec -it <container> yarn build`:

Código 18 – Variáveis de Ambiente do Frontend.

```
1 REACT_APP_ATCODER_API_URL=https://<address>/api/atcoder  
2 REACT_APP_CODEFORCES_API_URL=https://<address>/api/codeforces  
3 REACT_APP_UVA_API_URL=https://<address>/api/uva  
4 REACT_APP_CODECHEF_API_URL=https://<address>/api/codechef  
5 REACT_APP_SPOJ_API_URL=https://<address>/api/spoj  
6 REACT_APP_AUTH_API_URL=https://<address>/api/auth
```

Fonte: o Autor

Envie o que foi construído, diretório `build`, para a pasta `/var/www/ojtracker` do servidor, e atualize o arquivo de configuração da aplicação, adicionando:

Código 19 – Configuração do Frontend no Apache.

```
1 <Directory /var/www/ojtracker>  
2   Options Indexes FollowSymLinks  
3   AllowOverride All  
4   Require all granted  
5  
6   RewriteEngine On  
7  
8   RewriteCond %{REQUEST_FILENAME} !-f  
9   RewriteCond %{REQUEST_FILENAME} !-d  
10  RewriteRule ^ /ojtracker/index.html [L]  
11 </Directory>  
12  
13 Alias /ojtracker /var/www/ojtracker
```

Fonte: o Autor

Para finalizar, habilite o módulo de reescrita, `a2enmod rewrite`, e confirme as alterações reiniciando o servidor Apache, `systemctl restart apache2`.