

Victor de Sousa Lamarca

Plataforma digital com autenticação para discussões sobre programação

São Paulo

2020

Victor de Sousa Lamarca

Plataforma digital com autenticação para discussões sobre programação

Trabalho apresentado à Escola
Politécnica da Universidade de São
Paulo para obtenção do Título de
Engenheiro Eletricista com ênfase
em Computação

São Paulo

2020

Victor de Sousa Lamarca

Plataforma digital com autenticação para discussões sobre programação

Trabalho apresentado à Escola
Politécnica da Universidade de São
Paulo para obtenção do Título de
Engenheiro Eletricista com ênfase
em Computação

Área de Concentração:
Plataformas digitais

Orientador:
Prof. Dr. Reginaldo Arakaki

São Paulo
2020

AGRADECIMENTOS

Gostaria de agradecer meus familiares, amigos de universidade e meu orientador Reginaldo Arakaki por me oferecer o apoio necessário na jornada de se tornar um engenheiro.

Agradecimentos especiais aos colegas da USP do grupo de extensão MaratonUSP e demais colegas praticantes de programação competitiva que me auxiliaram no desenvolvimento deste projeto.

RESUMO

Com o crescente uso de redes sociais, fóruns e demais interações entre pessoas desconhecidas no meio virtual, é natural o questionamento do que é dito e da autenticidade das pessoas com as quais se discute. A autenticação dos usuários é uma forma de agregar credibilidade às discussões, mas esta solução nem sempre é escalável. Neste projeto é desenvolvida uma plataforma de discussão para programadores na qual os usuários são autenticados pela pontuação em um site de programação competitiva - o Codeforces. Assim, a plataforma visa promover enquetes sobre vários temas relativos ao domínio da programação nas quais os votos advêm de usuários autenticados.

Palavras-chave - Plataformas digitais. Rede social. Programação competitiva. Autenticação.

Abstract

With the increasing use of social networks, forums and other forms of interactions among unknown people in a virtual environment, it is natural to question what is said and the authenticity of the people with whom it is discussed. User authentication is a way of adding credibility to discussions, but this solution is not always scalable. In this project, a digital platform for programmers is developed in which users are authenticated by scoring on a competitive programming website - Codeforces. Thus, the platform aims to promote polls on various topics related to the domain of programming in which votes come from authenticated users.

Keywords - Digital platform. Social network. Competitive programming. Authentication.

Lista de figuras

- Figura 1 - Proposta de funcionalidade “Problem Likeability”
- Figura 2 - Desenho arquitetural alto nível do Twitter
- Figura 3 - Página inicial do Codeforces
- Figura 4 - Rank de participantes da USP no Codeforces
- Figura 5 - Codeforces Visualizer
- Figura 6 - Página inicial da plataforma Code-drills
- Figura 7 - Conta de Neymar Jr. no Instagram
- Figura 8 - Conta do autor no Codeforces
- Figura 9 - Placar final da Primeira Fase da maratona de Programação 2020
- Figura 10 - Classificação Final do IEEEExtreme Programming Competition 2020
- Figura 11 - Placar brasileiro da Google Code Jam 2020
- Figura 12 - Enquete plataforma
- Figura 13 - Tela inicial plataforma
- Figura 14 - Tela inicial com filtro expandido
- Figura 15 - Tela de visualização da autenticação
- Figura 16 - Tela de Autenticação - Input válido
- Figura 17 - Tela de Autenticação - Input válido Zoom
- Figura 18 - Tela de Autenticação - Input inválido Zoom
- Figura 19 - Tela submissão Codeforces
- Figura 20 - Página inicial logada
- Figura 21 - Tela validação com tempo expirado
- Figura 22 - Tela votação enquete
- Figura 23 - Desenho arquitetura
- Figura 25 - Tempo de resposta por requisição ao longo de 1 minuto

Lista de tabelas

Tabela 1 - Distribuição de títulos no Codeforces

Tabela 2 - Tempos de resposta por requisições de leitura

Sumário

1. Introdução	10
1.1 Motivação	10
1.2 Justificativa	10
1.3 Objetivo	11
2. Fundamentos conceituais	11
2.1 Princípios de arquitetura de software	12
2.2 Microserviços	12
2.3 Arquitetura típica de rede social	13
2.4 Javascript e Node.js	15
2.5 Codeforces e programação competitiva	15
2.6 Trabalhos relacionados	17
2.6.1 Plataformas derivadas do Codeforces	17
2.6.2 Plataformas de discussão com autenticação	18
2.7 Participações do autor em programação competitiva	20
3. Especificação do projeto	24
3.1 Público alvo e demanda	24
3.2 Seções da plataforma	24
3.3 Casos de uso	25
3.4 Formato de enquetes	26
3.5 Sistema de votação e autenticação	27
4. Implementação do sistema	27
4.1 Fluxo de telas	27
4.1.1 Visualização enquete e filtro	27
4.1.2 Fluxo de autenticação	30
4.1.3 Votação	36
4.2 Arquitetura	37
4.2.1 Desenho arquitetural	37
4.2.2 Módulo de autenticação	38
4.2.3 Módulo de leitura	38
4.2.4 Módulo de votação	39
4.2.5 Banco de dados	39
5. Resultados e próximos passos	40
5.1 Resultados	40
5.2 Próximos passos	41
6. Conclusão	42
Referências	43

1. Introdução

1.1 Motivação

Há uma demanda natural por autenticação de usuários existente em redes sociais como facebook ou aplicativos de encontro como Tinder, em que a autenticação dos usuários daria mais confiança de que os participantes da rede não são contas fakes ou mal intencionados. No projeto em questão, em que o público da rede seriam programadores no primeiro momento, a autenticação passaria maior credibilidade das informações discutidas.

1.2 Justificativa

Na maioria dos casos, o processo de autenticação pode não parecer escalável se depender de verificação de documentos de cada um dos usuários, mas a proposta desse projeto, no contexto de programação, é fazer a validação a partir de um website de competições de programação, o Codeforces [\[4\]](#). O Codeforces é uma plataforma de programação competitiva onde usuários competem em rounds, em geral de 2 horas, numa disputa de resolução de problemas relacionados a algoritmos e estrutura de dados. A cada round, os usuários recebem uma atualização de seu rating, que é uma pontuação que afere a habilidade do programador em comparação com os demais. O website vem ganhando popularidade visto que as competições são semelhantes a entrevistas técnicas usadas em processos seletivos de grandes empresas de tecnologia como Google e Facebook, onde eu, o autor, estagiei. A quantidade de contas registradas no Codeforces excede 100 mil, e o número de participantes dos rounds semanais excede 20 mil

Vale ressaltar que a plataforma na qual este projeto se baseia, o Codeforces, dispõe de uma completa API que permite projetos derivados, como este. O Codeforces já possui, também, de um sistema de fóruns embutida, mas pode-se afirmar que este é obsoleto devido a alguns motivos: o sistema de recomendação de blogs recomenda apenas baseado no tempo de publicação, apresentando os mais recentes; não há ferramenta de pesquisa interna de blogs; não há uma boa classificação quanto aos tipos de blog.

A imagem abaixo ilustra um dos muitos pedidos por funcionalidade nova ao site - um pedido por se incluir “problem likeability” ou seja, uma maneira de se permitir que usuários exponham suas opiniões sobre a qualidade dos problemas. A proposta recebeu mais de 500 upvotes, mas não foi implementada.

Figura 1 - Proposta de funcionalidade “Problem Likeability”

Noam527's blog

Problem Likeability Feature

By **Noam527**, [history](#), 2 months ago, 

Proposal

The feature to “star” or “favorite” a problem already exists; if you go to the problemset archive, you can choose to mark any problem as

 +506  

 [Noam527](#)

 2 months ago

 [19](#)

Fonte: [\[5\]](#)

1.3 Objetivo

A plataforma a ser desenvolvida pretende autenticar seus usuários a partir de suas contas no Codeforces, o que permitiria filtrar opiniões de programadores mais bem ranqueados pela pontuação do site. O formato de discussão principal que a plataforma visa promover é o de enquetes, no qual é possível avaliar numericamente as opções escolhidas por usuários autenticados. Exemplos de enquetes incluem: melhor editor de código; melhores problemas para praticar para entrevista técnica do Google; melhores livros para aprender certo tópico. Os votos podem então ser filtrados de acordo com o desejo do usuário para só considerar os programadores acima de um certo nível do Codeforces.

2. Fundamentos conceituais

2.1 Princípios de arquitetura de software

Há alguns princípios que guiam o desenvolvimento arquitetural de sistemas de software que buscam alta escalabilidade mesmo em meio a sua alta complexidade e dinamismo. Alguns princípios são descritos no livro Search Microsoft Application Architecture Guide, 2nd Edition [\[1\]](#):

- Separation of Concerns (SoC): separar o sistema em funcionalidades com o mínimo de interseção possível.
- Princípio da responsabilidade singular: cada módulo deve ter em geral apenas uma responsabilidade bem definida
- Princípio do conhecimento mínimo: cada módulo não deve conhecer detalhes internos de outros módulos
- Não se repita: as funcionalidades não devem ser repetidas e ou implementadas repetidas vezes mesmo entre diferentes módulos
- Minimização do design inicial: não realizar um design excessivamente detalhista no começo devido a dinamicidade do sistema, para assim evitar o desenvolvimento de funcionalidades que não serão usadas.

2.2 Microsserviços

Os princípios supramencionados a respeito do desenvolvimento de software se assemelham ao que é corroborado pela arquitetura de microsserviços conforme descrito por Martin Fowler [\[2\]](#) e [\[3\]](#). Ele descreve a arquitetura como sendo uma derivação da arquitetura orientada a serviços, em que módulos são criados com baixo acoplamento, interfaces simples e de granularidade fina. Isto permite que o desenvolvimento do software seja escalável no sentido que códigos podem ser implantados (deployed) de maneira independente, de maneira que uma mudança numa configuração implique, na maioria das vezes, numa mudança de um só módulo. Isto também permite que módulos sejam implementados com tecnologias diferentes, podendo se usar dar aquela mais adequada à funcionalidade específica do módulo.

Críticas comumente feitas ao modelo de microsserviços incluem, contudo, alguns fatores. Um deles é o *overhead* tipicamente demandado pelos módulos para

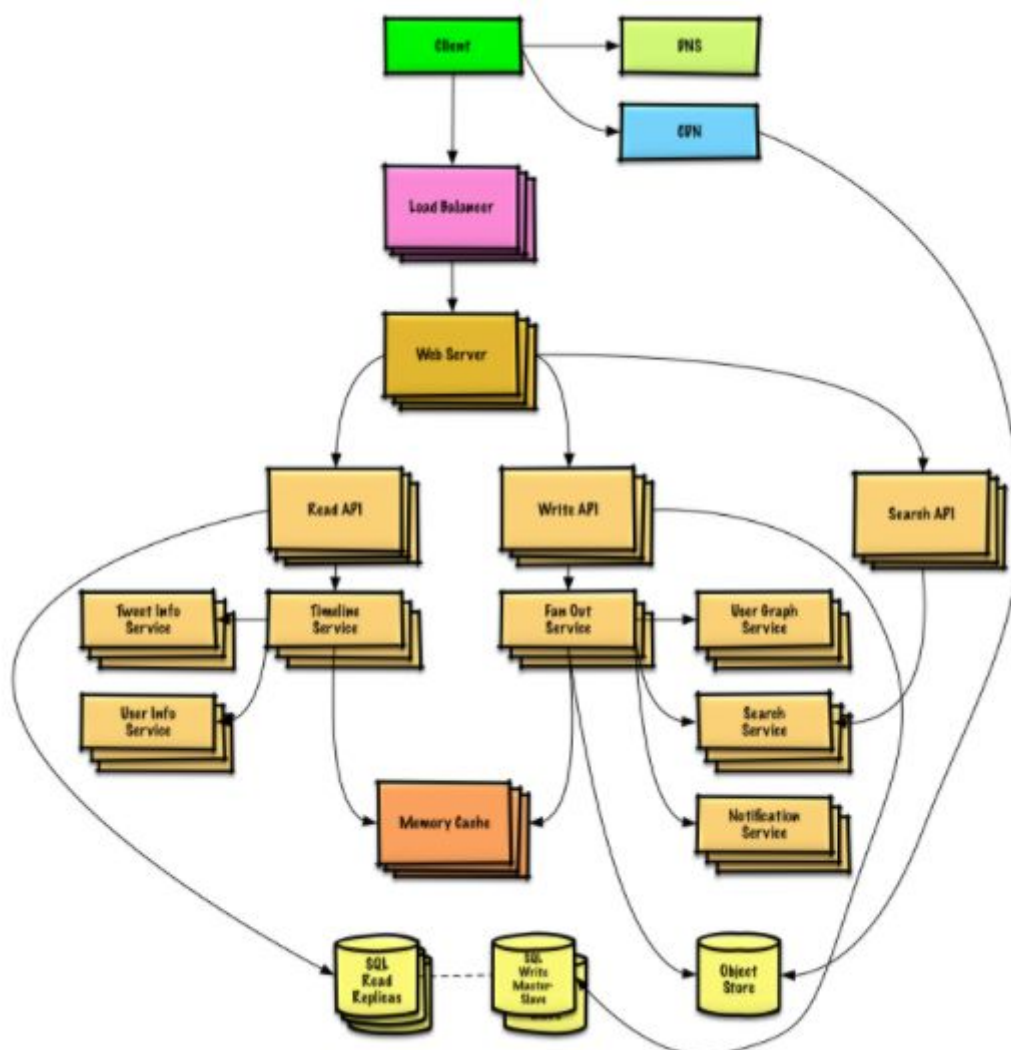
serem capazes de restaurar uma informação dependente de mais de um deles. Outro defeito neste sentido é descrito como sendo que, conforme o sistema evolui, as responsabilidades dos módulos podem ser movidas entre eles, o que exige a implementação repetida destas responsabilidades. Além disso, é criticado a falta de um parâmetro objetivo para se caracterizar um módulo como sendo do tipo microsserviços.

Os pontos positivos juntamente das críticas elencadas ao modelo de microsserviços constituem os *tradeoffs* considerados pelo arquiteto do sistema para de implementar este modelo. Contudo, visto ubiquidade desta implementação em sistemas complexos, é possível considerar o modelo como sendo empiricamente testado e, em geral, bem sucedido.

2.3 Arquitetura típica de rede social

A plataforma busca permitir que usuários conseguiram se comunicar em grande número, o que justifica uma arquitetura de software típica de rede social ou fóruns. Abaixo há um desenho arquitetural típico do twitter, que é uma plataforma que busca compartilhar opiniões e textos de curto formato.

Figura 2 - Desenho arquitetural alto nível do Twitter



Fonte: [6]

A arquitetura busca ilustrar alguns requisitos típicos que a arquitetura de um sistema desse tipo tem. Em primeiro lugar pode-se notar o load balancer, que faz o gerenciamento da demanda do sistema. Em seguida pode-se notar a caixa triplicada que corresponde a maioria dos módulos do sistema, ilustrando o paralelismo do processamento. Outra característica típica de sistemas do tipo twitter é que a demanda por leitura é muito maior que a de escrita na plataforma, ou seja, o uso de caso de se ler um twitte é muito mais comum que o de se escrever um twitte. Assim, a arquitetura do banco de dados é do tipo mestre escravo no sentido de que a escrita é realizada por um módulo singular, que evita duplicação, ao passo que a leitura dispõe de múltiplas cópias, numa espécie de memória cache, que aumenta o

custo ou chance de disponibilizar informações não atualizadas, mas permite a rápida leitura em grande quantidade.

2.4 Javascript e Node.js

O Javascript, tipicamente abreviado de JS, é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma. Juntamente do HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web [7]. Como uma linguagem multiparadigma, o JavaScript suporta estilos de programação orientados a eventos, funcionais e imperativos (incluindo orientado a objetos e prototype-based), apresentando recursos como fechamentos (closures) e funções de alta ordem

A linguagem surgiu para atender a demanda de processamento client-side, isto é, sem precisar passar pelo servidor. Em outras palavras, o servidor apenas fornece os dados necessários para que então, parte do processamento ocorra localmente na máquina do usuário e possa ser apresentado pelo navegador. Isto evidencia a natureza tipicamente de *front-end*, isto é, referente a desenvolvimento de interface, atribuída a linguagem Javascript.

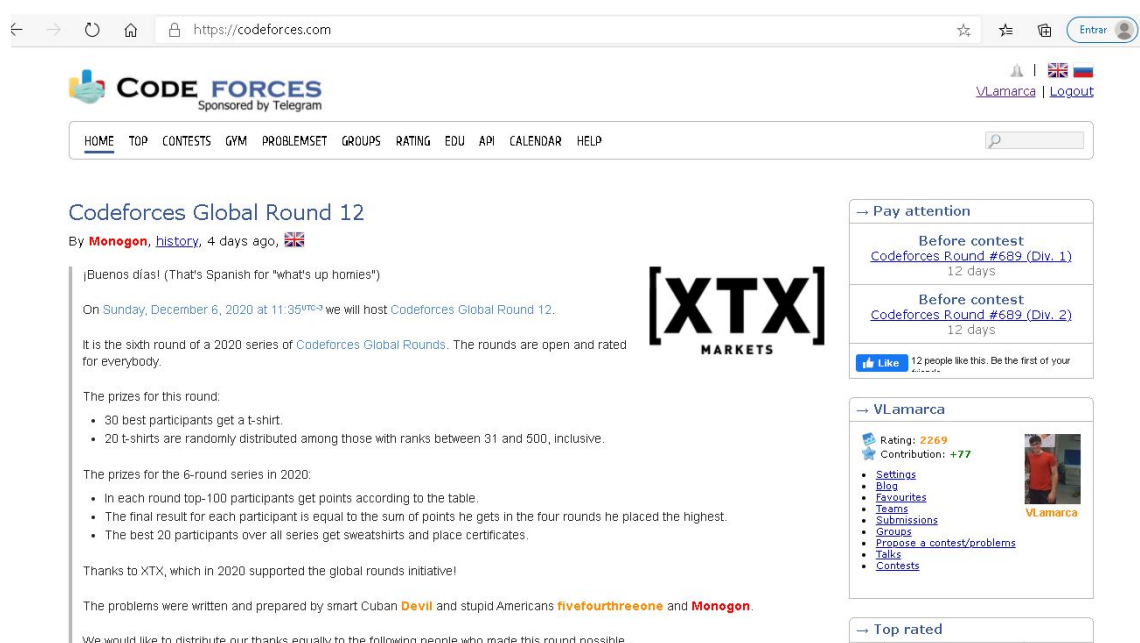
Contudo, frameworks vem sendo desenvolvidos em cima do Javascript de maneira a permitir seu trabalho tanto no *front-end* quanto no *back-end*. Exemplo mais notório de framework com esta função é o Node.js, um software open-source, cross-platform, capaz de executar o código de JavaScript a nível backend e frontend. Este framework unifica, portanto, o desenvolvimento de aplicações web ao redor de uma só linguagem [8].

Características notórias do Node.js incluem sua arquitetura movida a eventos (típica de tecnologias web), capacidade de comunicação I/O assíncrona, backwards compatibility (minimização da depreciação de código conforme o framework evolui). Essas e outras características tornaram o framework bastante popular não só apenas com desenvolvedores independentes ou entusiastas da comunidade open-source. O uso corporativo desta tecnologia reúne nomes como Microsoft, Netflix, IBM, SAP, Walmart, Yahoo entre outros [9].

2.5 Codeforces e programação competitiva

O Codeforces, plataforma sobre a qual a autenticação do projeto desta monografia deriva, é um popular website de programação competitiva, que vem ganhando popularidade juntamente da programação competitiva em si, que vem se mostrando um método efetivo de aprendizado de programação. O praticante desta atividade aprimora habilidades de implementação de código de maneira rápida, correta e eficiente, aliado, também, ao conhecimento de algoritmos e estrutura de dados.

Figura 3 - Página inicial do Codeforces



Fonte: [4]

Figura 4 - Rank de participantes da USP no Codeforces

RATING RATING (ALL) FRIENDS RATING Country: any country City: any city Organization: University of São Paulo, 111

Rating

	Who	#	=
1 (413)	 arthur.nascimento	66	2446
2 (517)	 VLamarca	90	2404
3 (764)	 queria.ser.arthur	6	2318
4 (1707)	 gafeol	70	2189
5 (3615)	 thilio	33	2064
6 (5182)	 tubone	210	1953
7 (5600)	 fakhoury	41	1932
8 (6742)	 Leonardo_Blanger	64	1884
9 (7229)	 noronha	28	1858
10 (7522)	 rmedeiros23	53	1843

Fonte: [\[4\]](#)

Um outro importante fator responsável pela popularidade da plataforma é também o fato de que os problemas tipicamente propostos são similares ao usados em entrevistas de grandes empresas de tecnologias, as chamadas big techs, que incluem Google, Facebook, Microsoft, etc. A combinação destes fatores resultam no expressivo número de usuários do Codeforces, que conta com 100 mil contas registradas e rounds (provas) semanais com cerca de 20 mil participantes [\[4\]](#).

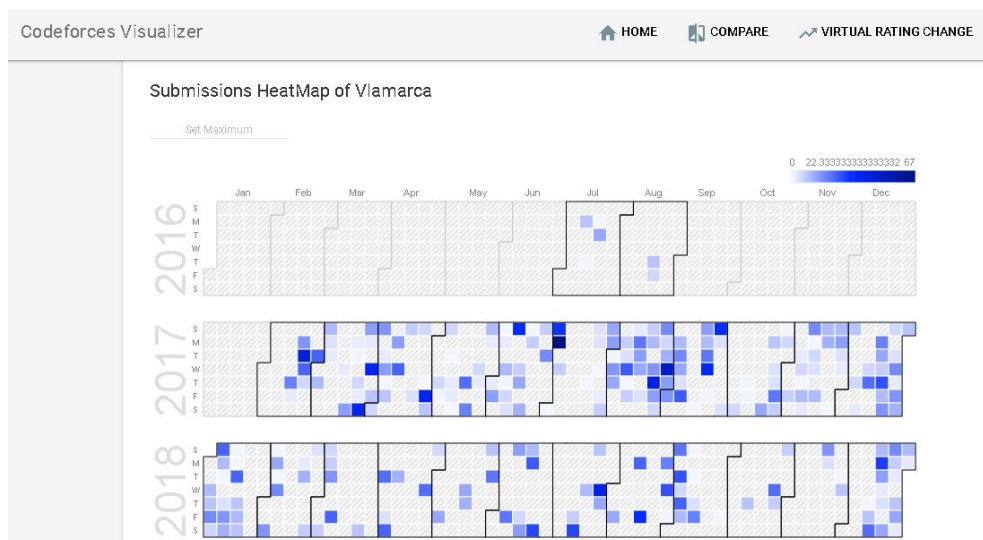
2.6 Trabalhos relacionados

2.6.1 Plataformas derivadas do Codeforces

O projeto desenvolvido nesta monografia é, como mencionado, derivado da plataforma de programação competitiva Codeforces. Outros projetos são derivados do do website também, visto que este disponibiliza uma API que possibilita coletar informações publicamente disponíveis no Codeforces [\[10\]](#).

Uma das plataformas derivadas que faz uso da API do Codeforces é uma intitulada de Codeforces Visualizer [\[11\]](#), que permite a visualização das submissões de um usuário num formato de heatmap, além de outras estatísticas como número de problemas resolvidos na plataforma, etc. Uma imagem exemplo da plataforma pode ser vista abaixo.

Figura 5 - Codeforces Visualizer



Fonte [\[11\]](#)

Outra plataforma também derivada do Codeforces que faz uso de sua API é denominada code-drills [\[12\]](#). O propósito desta é recomendar ao praticante de programação competitiva, baseado no seu histórico de participação no Codeforces, quais problemas o praticante deveria fazer por estarem num nível adequado de acordo com suas habilidades. Uma imagem da plataforma pode ser vista abaixo.

Figura 6 - Página inicial da plataforma Code-drills

The image shows the 'code-drills' website interface. At the top, there are navigation links: Tools, Resources, and FAQ. A 'Recommend 105' button is visible. A green banner at the top reads: 'We have launched! Hey there! We have launched our brand new codedrills site and have our own online judge now. You can visit and start solving at [codedrills.io](#). This site will remain at this new address [recommender.codedrills.io](#) and will continue to work as before. The old address will redirect here. Write to [hello@codedrills.io](#) for feedback and concerns'. Below the banner, the main heading is 'Practice problems recommender'. A subheading reads: 'Analyze your profile on various competitive programming sites and get practice problem recommendations based on your past submissions'. There is a text input field labeled 'Enter handle(s) with site prefix' containing the text 'Viamarca'. To the right of the input field is an 'Analyze' button. Below the input field, there is a section titled 'Input format' with two bullet points: 'Each handle should be prefixed with the site prefix (in lowercase) of that handle' and 'Sites supported'. A table follows, listing supported sites and their prefixes.

Site	Prefix	Example
Codeforces	cf/ no prefix	cf/tourist petr
Codechef	cc/	cc/mugurelionut
Spoj	sp/	sp/xilinx

Fonte [\[12\]](#)

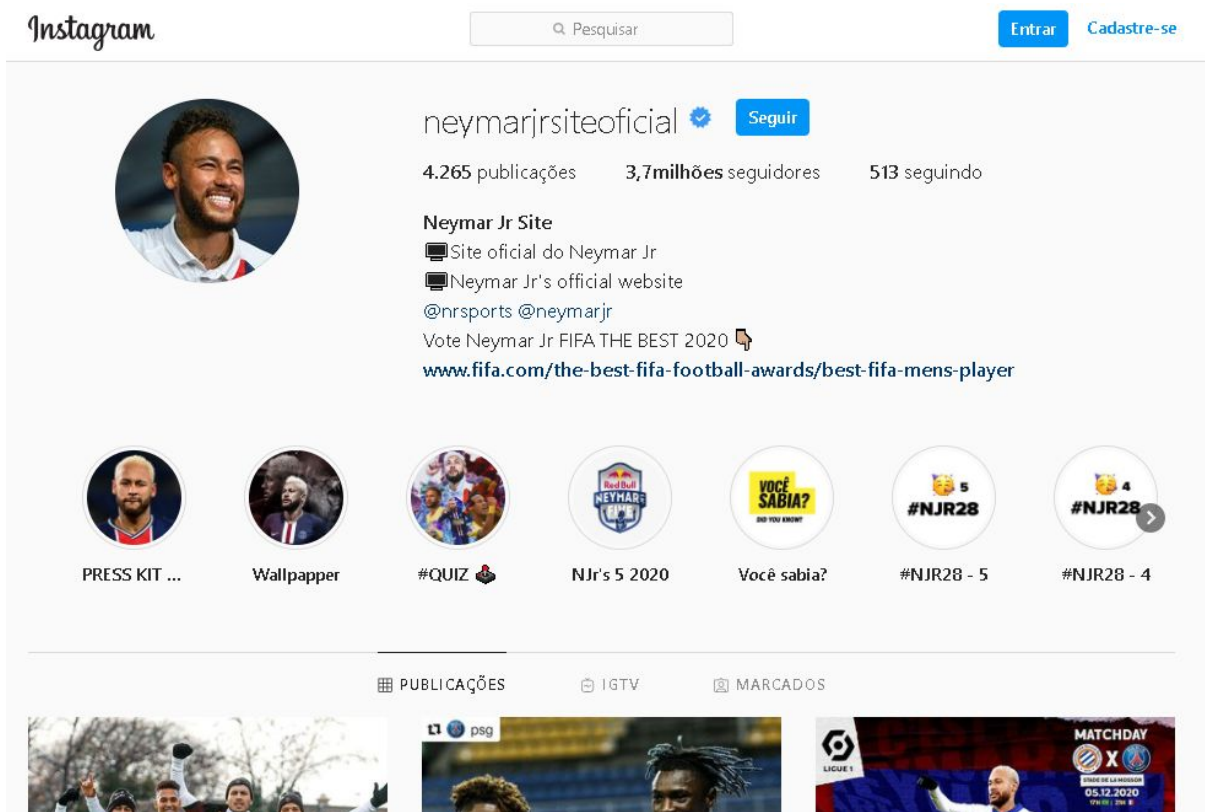
2.6.2 Plataformas de discussão com autenticação

Talvez uma das plataformas mais semelhantes com o propósito da desenvolvida neste projeto é a popular plataforma de perguntas e respostas relacionadas a programação no geral denominada Stackoverflow [\[13\]](#). O propósito desta plataforma é responder perguntas essencialmente técnicas e o autor da pergunta escolhe a melhor resposta. Demais usuários também podem assinalar se gostaram ou não da resposta com os botões upvote e downvote respectivamente. A credibilidade de uma resposta advém do próprio histórico de respostas de um usuário e do histórico que suas perguntas no Stackoverflow têm. A diferença dos Stackoverflow para a plataforma desenvolvida nesta monografia inclui aspectos como: o tópico das perguntas, que são técnicas e relacionadas a programação, mas não programação competitiva; autenticidade e credibilidade dos usuários é baseado no histórico da própria plataforma e em critérios subjetivos como a opinião dos próprios usuários da plataforma.

Já uma plataforma que inclui uma verificação da autenticidade de seus usuários é a plataforma de encontros românticos Inner Circle [\[14\]](#). Nela usuários são verificados antes de poderem participar da plataforma, garantindo que eles são quem dizem ser, com verificações que incluem exigência de documentos e verificação de histórico criminal. Tal autenticação não é, contudo, escalável e depende de trabalho humano para validação.

As redes sociais mais populares como Facebook, Instagram e Twitter também disponibilizam de verificação de autenticidades de seus usuários famosos através do tique azul. Contudo esta operação de validação também não é escalável e exige conferência humana de documentos, vídeo-conferências, entre outros. Assim, este recurso só está disponível aos usuários que são em geral mais alvo de contas maliciosas de usuários que tentam roubar a identidade de outros. Abaixo há uma imagem referente a conta do Instagram do futebolista Neymar Jr. em que é possível ver o tique azul que indica que a conta é de fato dele.

Figura 7 - Conta de Neymar Jr. no Instagram



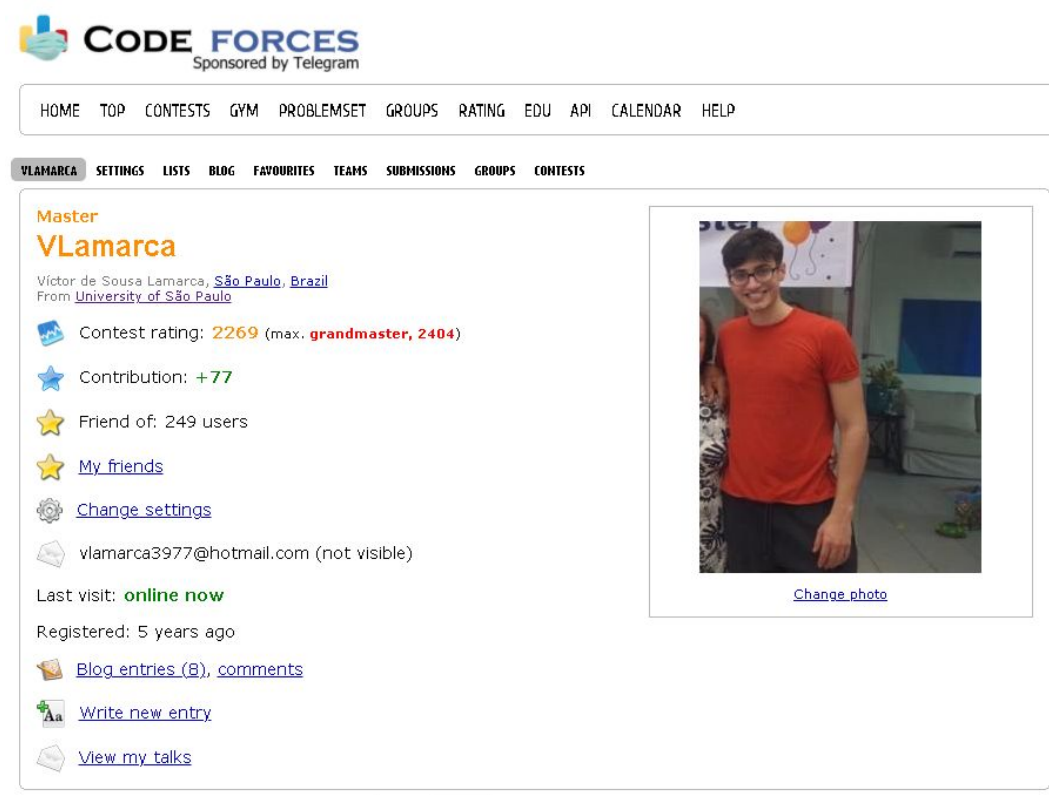
Fonte: [15]

2.7 Participações do autor em programação competitiva

Nesta seção são descritos alguns dos resultados do autor desta monografia na área de programação competitiva. São consideradas competições que ocorreram no meu período de graduação e demonstram a minha experiência acerca do tema. Outras conquistas minhas que são resultado direto da minha experiência em programação competitiva foram minha aprovação nos processos seletivos de empresas de tecnologia como Google e Facebook, onde estagiei. Mais especificamente estagiei no Google em Nova Iorque de julho a outubro de 2019 e no Facebook de janeiro a fevereiro de 2020. Abaixo uma lista com algumas de minhas conquistas na área de programação competitiva.

- Classificação de Grandmaster no Codeforces, que representa os top 0,5% mais bem ranqueados na plataforma

Figura 8 - Conta do autor no Codeforces



The image shows the Codeforces profile of a user named VLamarca. The profile is for a Master-level user with a contest rating of 2269. The user's name is Victor de Sousa Lamarca, from São Paulo, Brazil, and he is associated with the University of São Paulo. The profile includes a list of statistics: Contest rating (2269), Contribution (+77), and Friend of (249 users). There are links for 'My friends', 'Change settings', and a visible email address (vlamarca3977@hotmail.com). The user's last visit was 'online now' and they were registered 5 years ago. The profile also shows links for 'Blog entries (8), comments', 'Write new entry', and 'View my talks'. A photo of the user is displayed on the right side of the profile.

Fonte [16]

- Rank 7/799 na primeira fase da maratona de programação 2020 no time [USP] KondZhiLa. Classificado para representar a Universidade de São Paulo na final brasileira da ICPC em março de 2021 em Gramado.

Figura 9 -Placar final da Primeira Fase da maratona de Programação 2020

#	User/Site	Name
1	 teambrsp113/BR	[UNICAMP] namae wa??
2	 teambrsp066/BR	[USP - São Carlos] deitando no gramado
3	 teambrmg111/BR	[UFMG] Xerebêlerebêbis
4	 teambrba027/BR	[UFBA] União Fiasco
5	 teambrmg142/BR	[UNIFEI] The droids you are looking for
6	 teambrpb025/BR	[UFCG] vae victis
7	 teambrsp167/BR	[USP] KondZhiLa
8	 teambrmg107/BR	[UFMG] GTA: Grover's Street Algorithms
9	 teambrsp116/BR	[UNICAMP] \$wa
10	 teambrpb022/BR	[UFCG] O amarelo é impostor

Fonte [\[17\]](#)

- Rank 3/4557 na competição de programação do IEEE - IEEEExtreme Programming Competition 2020, representando a Universidade de São Paulo no time WhereIsOurIEEEtshirt.

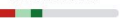
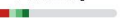
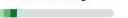
Figura 10 - Classificação Final do IEEEExtreme Programming Competition 2020

Rank	Team	Score (Penalty)	University
1	 Aurora	2056.69 (2931)	Princess Sumaya University for Technology
2	 KnimAndNafrohim	2054.21 (5412)	Multiple Schools
3	 WhereIsOurEEETshirt	1950.48 (3085)	University of Sao Paulo
4	 unswcoffeeoncampus	1935.34 (3453)	UNSW Sydney
5	 OxdeadCode	1923.64 (5638)	Unknown
6	 TURandomizedAlgos	1906.6 (4919)	University of Tartu
7	 RatingMiSeRable	1882.94 (3644)	Universidad Nacional de Ingenieria Lima
8	 SamuraisDelUwu	1817.56 (5817)	Universidad de Ingenieria y Tecnologia - UTEC
9	 SherbinilsBluffing	1794.35 (6829)	Mansoura Univ
10	 MaxFlowWhiteBeard	1775.93 (5698)	Princess Sumaya University for Technology

Fonte [\[18\]](#)

- Primeiro lugar do Brasil na Google Code Jam 2020

Figura 11 - Placar brasileiro da Google Code Jam 2020

code jam									
Round 3 2020									
Show round overview ☐									
The top 25 contestants in this round will advance to the World Finals.									
		Naming Compromise		Thermometers		Pen Testing		Recalculating	
									
		Open problem		Open problem		Open problem		Open problem	
192	 Vlamarca	0:47:04 +0	✓ ✓	2:16:28 +0	✓	—		17	2:16:28
Brazil									
		4	8	5	19	6	11	15	
192	 Vlamarca	★	0:47:04 +0	✓ ✓	2:16:28 +0	✓	—	17	2:16:28
303	 tfgg	☆	0:30:21 +3	✓ ✓				12	0:42:21
321	 Nson	☆	0:46:47 +0	✓ ✓	7 attempts	—	—	12	0:46:47
343	 dilsonguim	☆	0:51:00 +0	✓ ✓				12	0:51:00

Fonte [\[19\]](#)

3. Especificação do projeto

Aqui são melhor definidas as especificações do projeto e como construiu-se uma plataforma que permite autenticação de usuários e o compartilhamento de suas opiniões em enquetes.

3.1 Público alvo e demanda

A plataforma é desenvolvida para programadores que estão praticando suas habilidades em algoritmos e estrutura de dados através de programação competitiva, com o intuito serem bem sucedidos em entrevistas técnicas de empresas como Google, Facebook e Microsoft. Em especial o público alvo desta plataforma são aqueles que já tem conta na plataforma Codeforces, e podem perceber valor numa plataforma que permite discussões com usuários autenticados do Codeforces. Vale ressaltar que o Codeforces têm 300 mil contas criadas e cerca de 30 mil usuários semanais, sendo este um limite para o público da plataforma desenvolvida.

Considera-se a demanda da plataforma como sendo semelhante a de plataformas como o Twitter. A demanda do Twitter pode ser ilustrada como tendo 100 milhões de usuários mensais, 15 bilhões de tweets por mês e 250 bilhões de requisições de leitura por mês, o que equivale a 5 milhões de requisições de leitura por minuto [\[6\]](#) e [\[20\]](#). Extrapolando para a plataforma a ser desenvolvida, tem-se 2 importantes conclusões a respeito da demanda que deve guiar a arquitetura do sistema:

- Haverá cerca de 10 vez mais requisições de leitura do que de escrita
- Assumindo 10 mil usuários ativos, espera-se um pico de 1000 requisições de leitura por minuto.

3.2 Seções da plataforma

Visto que a plataforma tem um nicho específico de usuários, já há predefinidos seções de discussão de temas relacionados ao domínio da programação competitiva. Em outras palavras, diferente de redes sociais como o

Facebook, há uma limitação no tipo de conteúdo publicado, com seções pré-definidas. As seções buscam abordar os principais temas de programação competitiva e são elas:

- **Mais populares:** Esta seção, que é também a home page do site, busca listar as melhores publicações das outras seções.
- **Melhores tutoriais:** Esta seção busca unir os melhores tutoriais acerca do tema competitive programming. A demanda desta seção advém da extensa quantidade de tutoriais disponíveis na internet, de maneira que nesta seção se busca filtrar e indicar os melhores tutoriais de forma orgânica, ou seja, através da votação dos usuários autenticados.
- **Melhores problemas:** Assim como a seção melhores tutoriais, esta seção busca reunir de forma orgânica os melhores problemas ou exercícios da área, com indicações do tema. Ou seja, busca-se reunir os problemas com solução mais bonita, ou então os problemas que melhor ilustram certo algoritmo de forma didática.
- **Entrevista técnica:** Nesta seção busca-se discutir a respeito de entrevistas técnicas cobradas em grandes empresa de tecnologia. Exemplo de enquetes nessa seção são: qual material seguir para melhor se preparar para entrevista de algoritmos do Google.
- **Off-topic:** Esta seção é dedicada a assuntos não necessariamente diretamente relacionadas ao domínio da programação competitiva, onde pode-se discutir a respeito de outros hobbies por exemplo.

3.3 Casos de uso

Neste item são elencadas quais as operações que serão possíveis de serem realizadas na plataforma. Abaixo há uma lista dos casos de uso da plataforma, que compreende os requisitos funcionais.

- **Cadastro/autenticação de conta:** É possível criar uma conta na plataforma com email, nome do usuário e senha. É possível associar a conta na plataforma com uma conta no codeforces (gerando a autenticação).
- **Pesquisa e leitura de tópicos:** Mesmo sem estar logado em uma conta, é possível pesquisar sobre os tópicos publicados na plataforma,

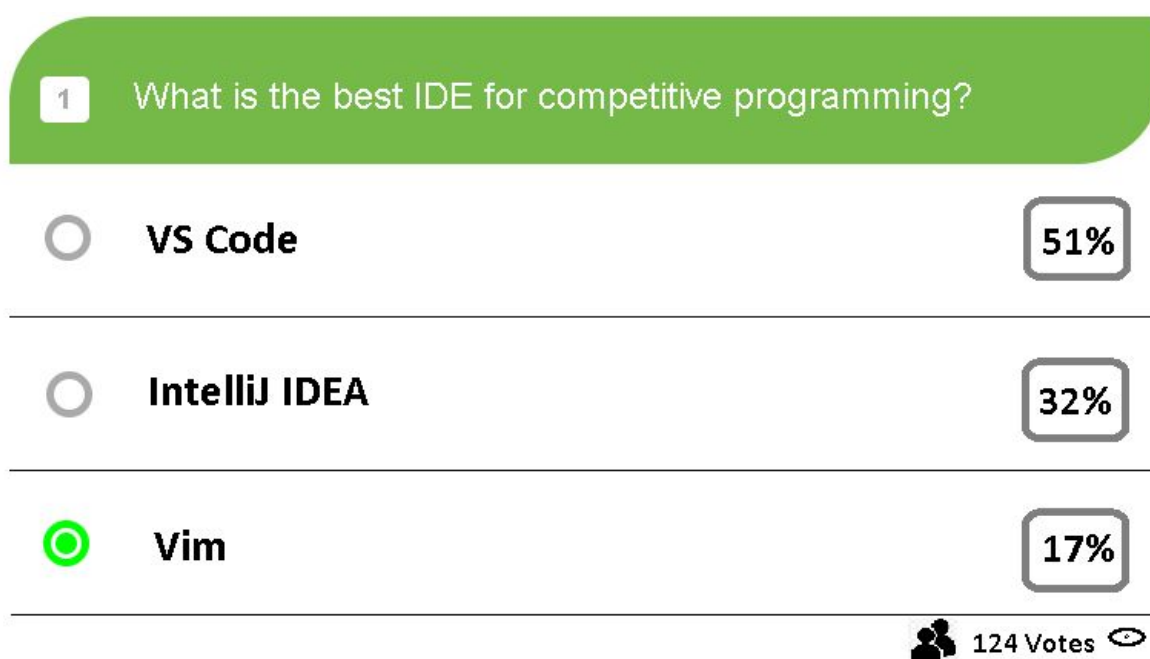
que consiste em ler a enquete, que inclui a pergunta e as opções de resposta com a respectiva porcentagem de votos recebidos de acordo com o nível.

- **Criação e postagem de tópicos:** Criação das enquetes a serem publicadas. Apenas usuários autenticados e de nível referente a 1ª divisão do Codeforces têm este recurso
- **Votação em tópico:** Votar em uma das opções de uma enquete, apenas usuários autenticados têm este recurso.

3.4 Formato de enquetes

As enquetes são parte central da plataforma e representam uma forma de quantificar opções de resposta através de um número de votos de usuários autenticados. Abaixo há uma imagem do tipo de enquete que é promovida, pode-se ver uma pergunta, as opções resposta acompanhadas da respectiva porcentagem de votos recebidos, bem como o total de votos naquela enquete específica. Vale ressaltar que tem-se como requisito permitir a visualização de enquetes mesmo que sem uma conta autenticada na plataforma. Deve ser possível também filtrar o nível dos usuários que tem seus votos considerados na enquete, de maneira que apenas os votos de usuários do mais alto nível, por exemplo, sejam computados.

Figura 12 - Enquete plataforma



3.5 Sistema de votação e autenticação

O sistema de votação e autenticação são atrelados, já que para ser permitido votar é necessário ter uma conta autenticada. O usuário é autenticado através da plataforma Codeforces, que atribui, portanto o rating de sua conta, parâmetro que é usado para mensurar a habilidade do programador e pode ser filtrada na plataforma de acordo com o desejo de quem visualiza as enquetes. Para efeitos de simplificação a votação permite apenas a escolha de uma das opções contida na enquete. Vale também ressaltar que, até o momento, também para efeitos de simplificação, não são permitidas criação de enquetes pelos usuários, apenas pelo administrador da plataforma.

4. Implementação do sistema

4.1 Fluxo de telas

4.1.1 Visualização enquete e filtro

Na home page da plataforma pode-se verificar, primeiramente, o nome da plataforma: “The Correct Opinion”. Quanto ao conteúdo, a primeira seção que aparece ao se entrar na plataforma é a “Mais populares” que une as publicações e enquetes mais populares, para já no primeiro momento evidenciar ao usuário os tipos de discussão presentes na plataforma. Abaixo há um exemplo da home page ilustrando um exemplo de enquete que pode estar presente nela.

Figura 13 - Tela inicial plataforma

[The Correct Opinion](#) [Most Popular](#) [Best Tutorials](#) [Best problems](#) [Technical Interview](#) [Off Topic](#) [About](#) [Sign Up/Login](#)

The Correct Opinion

We share the best opinions on the web!

Choose the considered rating of voters here

Candidate Master ▼

1

What is the best IDE for competitive programming?

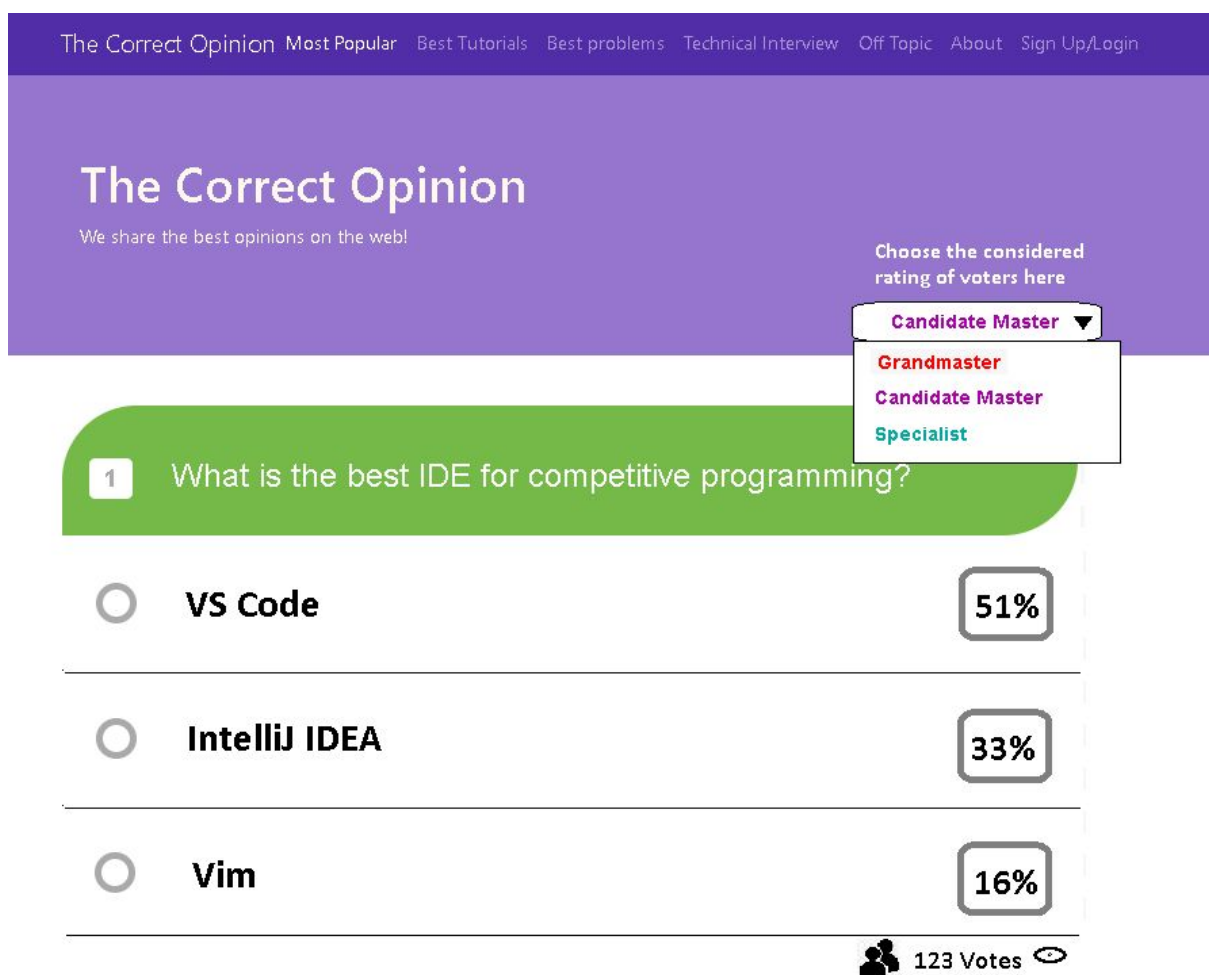
☐	VS Code	51%
☐	IntelliJ IDEA	33%
☐	Vim	16%

 123 Votes 

Na imagem em particular pode-se visualizar a enquete em si, um dos componentes mais importantes da plataforma. O exemplo apresentado mostra uma pergunta: “What is the best IDE for competitive programming?” que traduzido significa “Qual o melhor IDE (Ambiente Integral de Desenvolvimento) para programação competitiva?”. Há 3 opções presentes como resposta nessa enquete VS Code, IntelliJ IDEA. Na parte a direita da enquete, acompanham as opções de resposta as respectivas porcentagens referente ao total de votos que cada uma das opções recebeu. No canto inferior direito é mostrado o total de votos que a enquete em particular recebeu, acompanhado do botão de olho que representa a visualização dos usuários que votaram na enquete. No canto superior direito da página é possível ver a opção de filtro, que determina o rating dos participantes que são considerados na votação.

Abaixo é possível ver a opção do filtro expandida. As opções de escolha são Grandmaster, Candidate Master e Specialist.

Figura 14 - Tela inicial com filtro expandido



Na tabela a seguir é possível ver o que os ratings representam. Das opções disponíveis para escolha, Grandmaster é o nível mais alto (e portanto menos usuários possuem esse título) enquanto que Specialist é o nível mais baixo. Na tabela abaixo é mostrado o título acompanhado do número de contas que possuem este nível ou superior. É também mostrado o título newbie, título mais baixo, que mostra o número total de contas registradas no Codeforces.

Tabela 1 - Distribuição de títulos no Codeforces

Título do Codeforces	Número aproximado de contas que têm título igual	Porcentagem do total de contas
----------------------	--	--------------------------------

	ou superior	
Grandmaster	700	0,2%
Candidate Master	6 500	2,2%
Specialist	24 000	8%
Newbie	300 000	100%

Fonte : [\[4\]](#)

4.1.2 Fluxo de autenticação

Para o fluxo de autenticação, parte central da plataforma, tem-se a imagem posta a seguir. A tela é alcançada da página inicial, escolhendo no menu superior a opção “Sign Up/Login”. Há dois campos de preenchimento, o Codeforces handle (nome da conta do usuário na plataforma codeforces) e a senha. Na tela é possível ler a descrição sobre como a autenticação funciona, indicando que, ao se clicar no botão “Validate”, um link será fornecido para um problema do Codeforces, do qual será necessário enviar uma submissão em até 2 minutos. O sistema valida que tal ação foi tomada e portanto aceita a conta como sendo de fato do usuário.

Figura 15 - Tela de visualização da autenticação

The Correct Opinion: [Most Popular](#) [Best Tutorials](#) [Best problems](#) [Technical Interview](#) [Off Topic](#) [About](#) [Sign Up/Login](#)

The Correct Opinion

We share the best opinions on the web!

[Sign Up](#) / [Login](#)

Enter your codeforces handle along with a password, different from your codeforces password. You will receive a link to a Codeforces problem, to validate your account you need to submit a code that gets **compilation error verdict** in the given problem within 2 minutes.

Codeforces handle

Password

Na imagem abaixo é mostrado o caso principal, em que a conta fornecida é válida. Neste caso um timer começa a contar no canto direito da tela e o link para submissão é fornecido logo acima

Figura 16 - Tela de Autenticação - Input válido

[The Correct Opinion](#): [Most Popular](#) [Best Tutorials](#) [Best problems](#) [Technical Interview](#) [Off Topic](#) [About](#) [Sign Up/Login](#)

The Correct Opinion

We share the best opinions on the web!

[Sign Up](#) / [Login](#)

Enter your codeforces handle along with a password, different from your codeforces password. You will receive a link to a Codeforces problem, to validate your account you need to submit a code that gets **compilation error verdict** in the given problem within 2 minutes.

Codeforces handle

✓
Valid handle

Password

Validate

Please submit a code in this problem: [467A](#)

0m 38s

Na tela abaixo é possível ver a mesma imagem com um zoom na parte mais relevante, referente a confirmação que de fato a conta fornecida é existente na plataforma do Codeforces.

Figura 17 - Tela de Autenticação - Input válido Zoom

[Sign Up](#) / [Login](#)

Enter your codeforces handle along with a password, different from your codeforces password. You will receive a link to a Codeforces problem, to validate your account you need to submit a code that gets **compilation error verdict** in the given problem within 2 minutes.

Codeforces handle

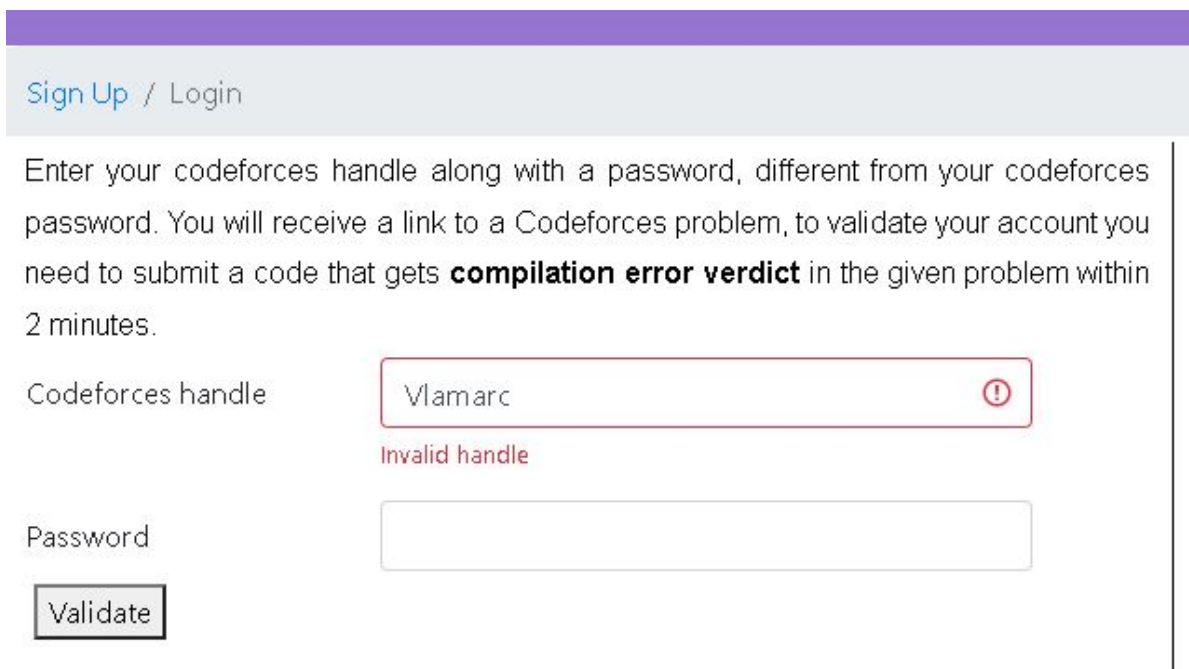
✓
Valid handle

Password

Validate

Já na imagem abaixo é possível visualizar um dos fluxos alternativos no qual o nome de usuário (handle) fornecido não consta na base de dados do Codeforces

Figura 18 - Tela de Autenticação - Input inválido Zoom



The screenshot shows the Codeforces authentication interface. At the top, there is a purple header bar. Below it, a light blue bar contains the text "Sign Up / Login". The main content area has a light gray background. It starts with a paragraph: "Enter your codeforces handle along with a password, different from your codeforces password. You will receive a link to a Codeforces problem, to validate your account you need to submit a code that gets **compilation error verdict** in the given problem within 2 minutes." Below this, there are two input fields. The first is labeled "Codeforces handle" and contains the text "Vlamarc". To the right of the input field is a red circle with a white exclamation mark. Below the input field, the text "Invalid handle" is displayed in red. The second input field is labeled "Password" and is empty. Below the password field is a button labeled "Validate".

No caso principal a ação tomada, ao clicar no link fornecido, deve redirecionar o usuário a plataforma do Codeforces, no problema em questão. Na imagem abaixo é possível ver como a tela se parece. A ação tomada seria enviar um código qualquer no problema, sendo escolhido pelo botão “browse” no canto direito.

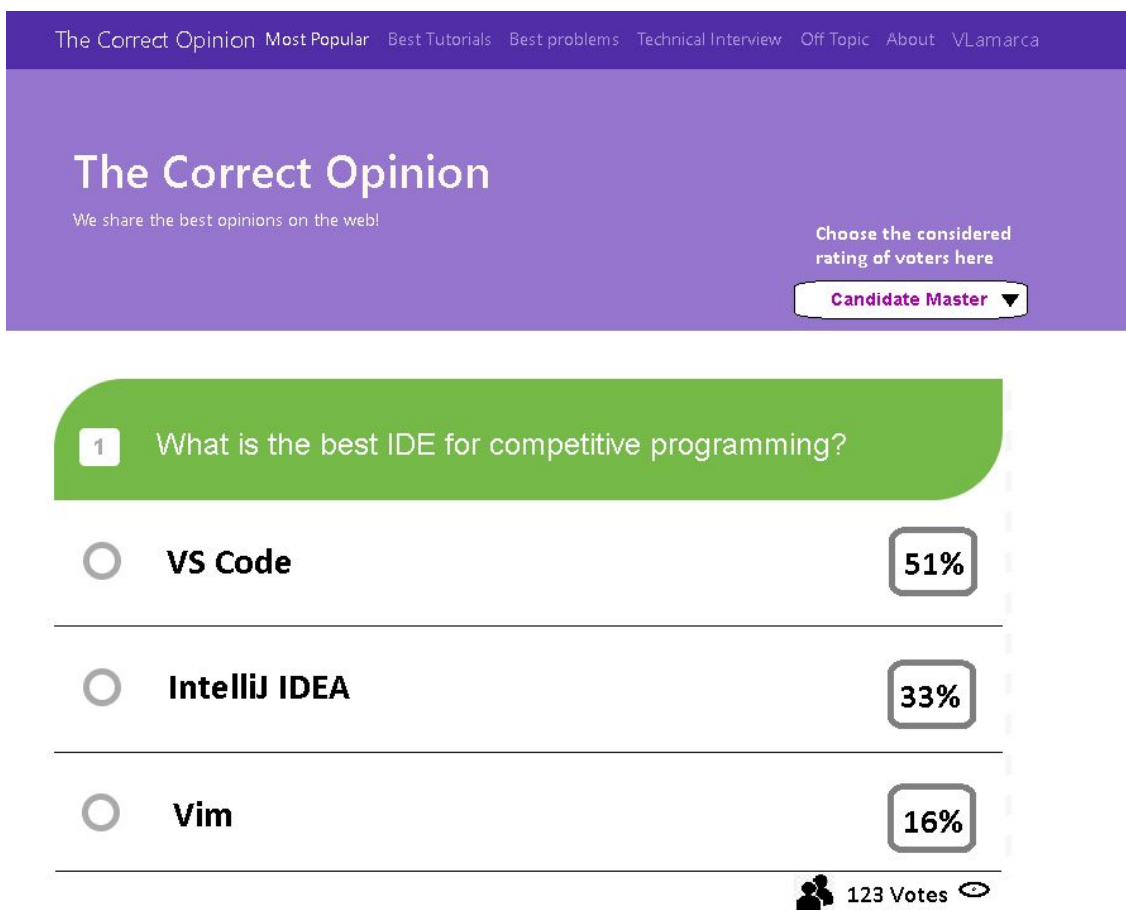
Figura 19 - Tela submissão Codeforces

The screenshot shows the Codeforces website interface for a problem titled "A. George and Accommodation". The URL in the browser is <https://codeforces.com/problemset/problem/467/A>. The page includes navigation tabs: PROBLEMS, SUBMIT, STATUS, STANDINGS, and CUSTOM TEST. The problem details specify a time limit of 1 second and a memory limit of 256 megabytes, with standard input and output. The problem description states that George and Alex are moving into a dormitory with n rooms. Each room i has p_i people already living in it and a capacity of q_i . The task is to determine the number of rooms where George and Alex can move in, given that each room can accommodate at most one person. The input format is: the first line contains n ($1 \leq n \leq 100$), and the next n lines contain pairs of integers p_i and q_i ($0 \leq p_i \leq q_i \leq 100$). The output is a single integer representing the number of suitable rooms. Examples are provided: for input $n=3$ with rooms $(1,1)$, $(2,2)$, and $(3,3)$, the output is 0 . On the right side, there are buttons for "Finished", "Practice", and "Start virtual contest". There is also a section for "Virtual participation" and a "Clone Contest to Mashup" button. At the bottom right, there is a "Submit?" section with a language dropdown set to "GNU G++17 9.2.0 (64 bit, m)" and a "browse" button for file selection.

Fonte: [4]

Caso a operação seja bem sucedida, o usuário é redirecionado a tela inicial, mas agora logado em sua conta, conforme imagem abaixo. Note que agora, no canto superior direito, consta o nome do usuário que agora está logado.

Figura 20 - Página inicial logada



Já outro fluxo alternativo pode ocorrer quando a ação não é tomada dentro do intervalo estipulado de 2 minutos. Neste caso, a plataforma indica que o tempo expirou, e que o usuário deve requisitar outro problema para tentar revalidar sua conta novamente. A imagem abaixo mostra a tela quando o tempo expira.

Figura 21 - Tela validação com tempo expirado

The Correct Opinion:
Most Popular
Best Tutorials
Best problems
Technical Interview
Off Topic
About
Sign Up/Login

The Correct Opinion

We share the best opinions on the web!

[Sign Up](#) / [Login](#)

Enter your codeforces handle along with a password, different from your codeforces password. You will receive a link to a Codeforces problem, to validate your account you need to submit a code that gets **compilation error verdict** in the given problem within 2 minutes.

Codeforces handle

VLamarca
✓

Valid handle

Password

.....

Validate

Please submit a code in this problem: [467A](#)

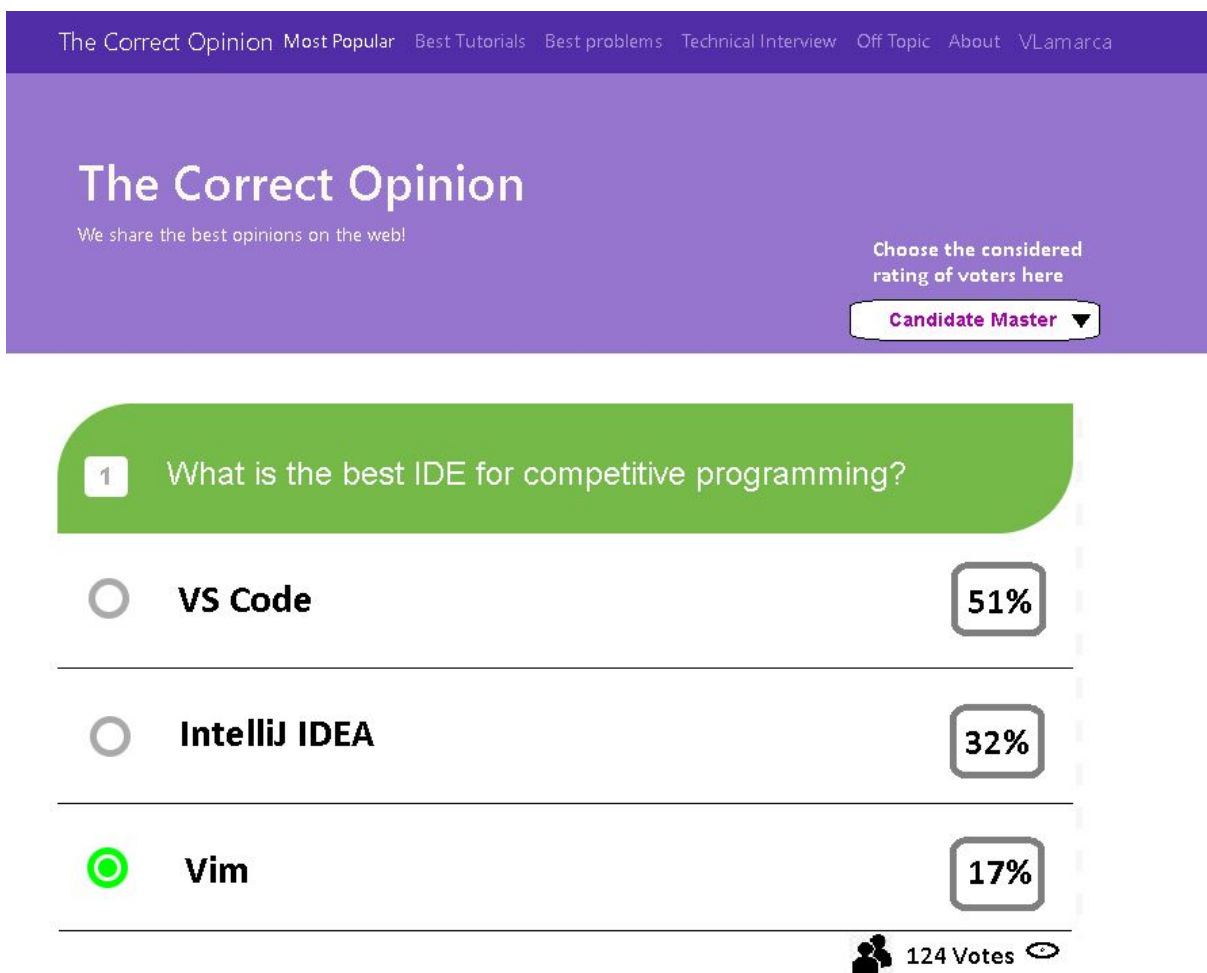
EXPIRED

Validate again for a new problem

4.1.3 Votação

O fluxo de votação é bastante simples, e pode já ser visto a partir da tela inicial. Contudo, a operação de votar apenas funcionará caso o usuário esteja logado. Na imagem abaixo é possível ver uma das opções da enquete selecionada, que indica que a votação ocorreu. Para executar esta ação basta apertar no círculo à esquerda da opção correspondente.

Figura 22 - Tela votação enquete

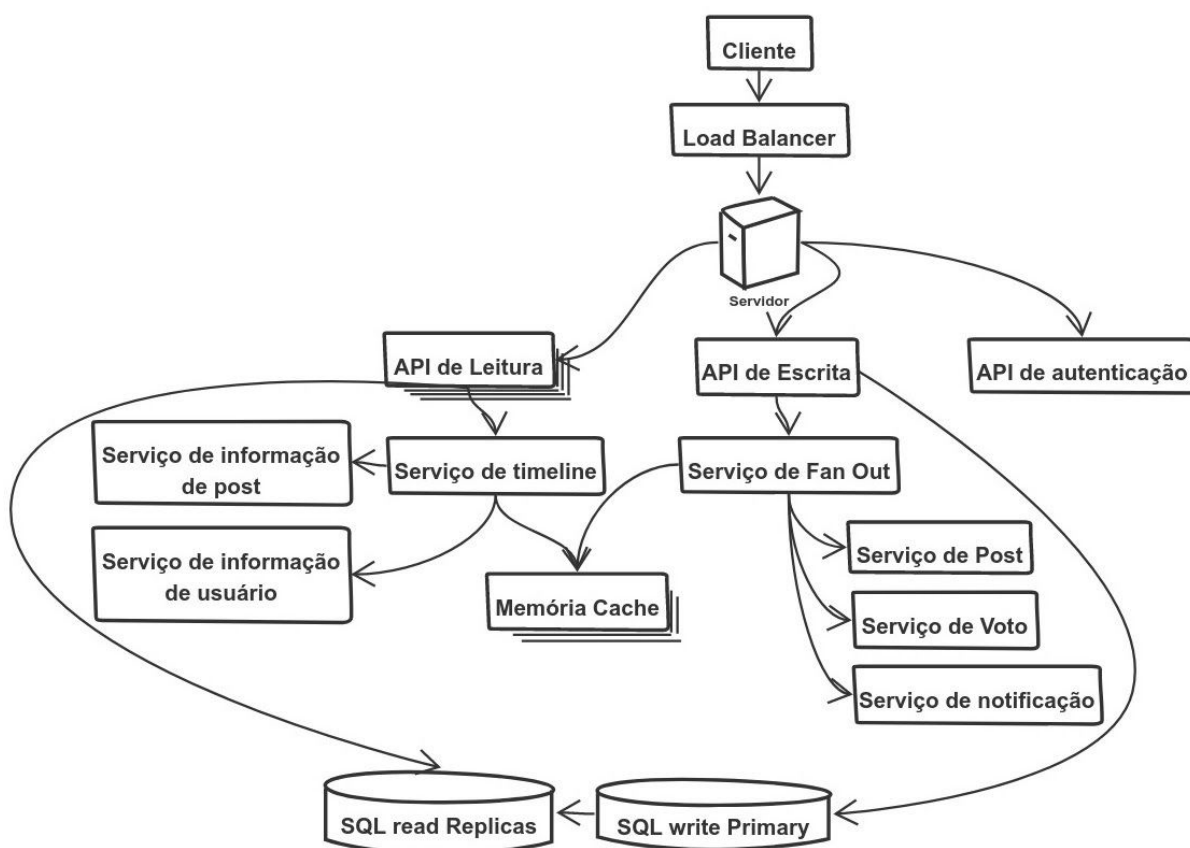


4.2 Arquitetura

4.2.1 Desenho arquitetural

Abaixo consta o desenho arquitetural da plataforma, que ilustra alguns requisitos que o sistema deve atender bem como os trade offs de design feitos com base na provável proporção de uso dos diferentes módulos. Alguns dos módulos são descrito com mais profundidade nos itens seguintes.

Figura 23 - Desenho arquitetural



4.2.2 Módulo de autenticação

A autenticação é implementada a partir da API do Codeforces. Primeiramente o usuário indica na plataforma “The Correct Opinion” a conta da qual é dono no Codeforces e então tem 2 minutos para tomar determinada ação no Codeforces para comprovar a posse da conta de fato. Caso a ação seja cumprida, a autenticação é feita. Mais especificamente, há um conjunto de 100 problemas pré definidos dos quais um é escolhido ao se fazer uma requisição de autenticação. Primeiro é verificado que o usuário de fato existe no Codeforces, em seguida o problema sorteado é informado ao usuário. Então o usuário deve submetê-lo no codeforces dentro de 2 minutos. O servidor faz requisições ao Codeforces a cada 2 segundos para verificar que o usuário de fato tomou a ação pedida. Este módulo foi implementado inteiramente em Javascript.

4.2.3 Módulo de leitura

O módulo de leitura é responsável por determinar quais enquetes aparecem na seção (serviço de timeline) e disponibilizar as enquetes com seus respectivos

valores. O serviço de timeline simplesmente alterna enquetes mostrando primeiro a com maior quantidade de votos, seguido da enquete mais recente feita na plataforma, intercalando assim as melhores enquetes baseadas ora em um parâmetro ora noutro. Espera-se desse módulo maior dinamismo com a evolução da plataforma, de maneira que pode-se mudá-lo muitas vezes, seja para mudar o design da enquete, seja para mudar os parâmetros de ordenação do serviço de timeline.

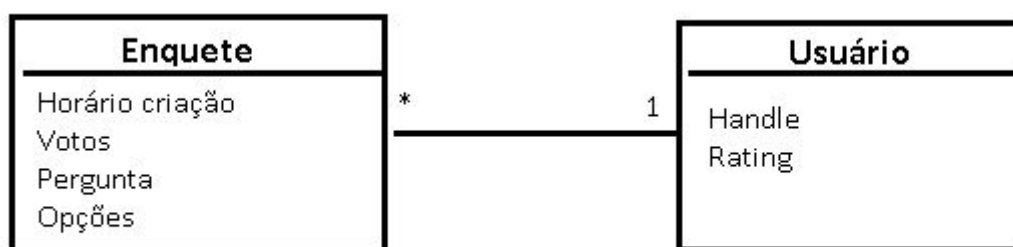
4.2.4 Módulo de votação

O módulo de votação processa os votos e atualiza os valores da enquete de maneira síncrona, ou seja, apenas quando a persistência do voto é feita no banco de dados que o usuário vê a confirmação visual que seu voto foi processado. Este é o módulo de escrita que é mais demandado de maneira que a lógica do servidor foi implementado em C++ ao invés de Javascript como a maioria dos demais componentes. Esta decisão foi feita visando a eficiência deste. [\[3\]](#)

4.2.5 Banco de dados

O banco de dados implementado segue princípios explicados no item 3.1 de que o número de requisições de leitura em redes sociais segue uma razão de 10 vezes mais requisições de leitura do que escrita. Para tanto o sistema segue o princípio de computação distribuída de banco de dados primário-réplica (antigo mestre-escravo que vêm caindo em desuso). Há apenas um servidor primário e várias réplicas, o primário lê e escreve enquanto que réplicas apenas leem, o que mantém a confiabilidade do conteúdo escrito, enquanto que permite um número grande de leituras (não necessariamente atualizadas). A implementação deste sistema seguiu Postgresql como primário e MySQL como réplicas. [\[21\]](#). A imagem do diagrama relacional mantido segue abaixo.

Figura 24 - Diagrama Relacional



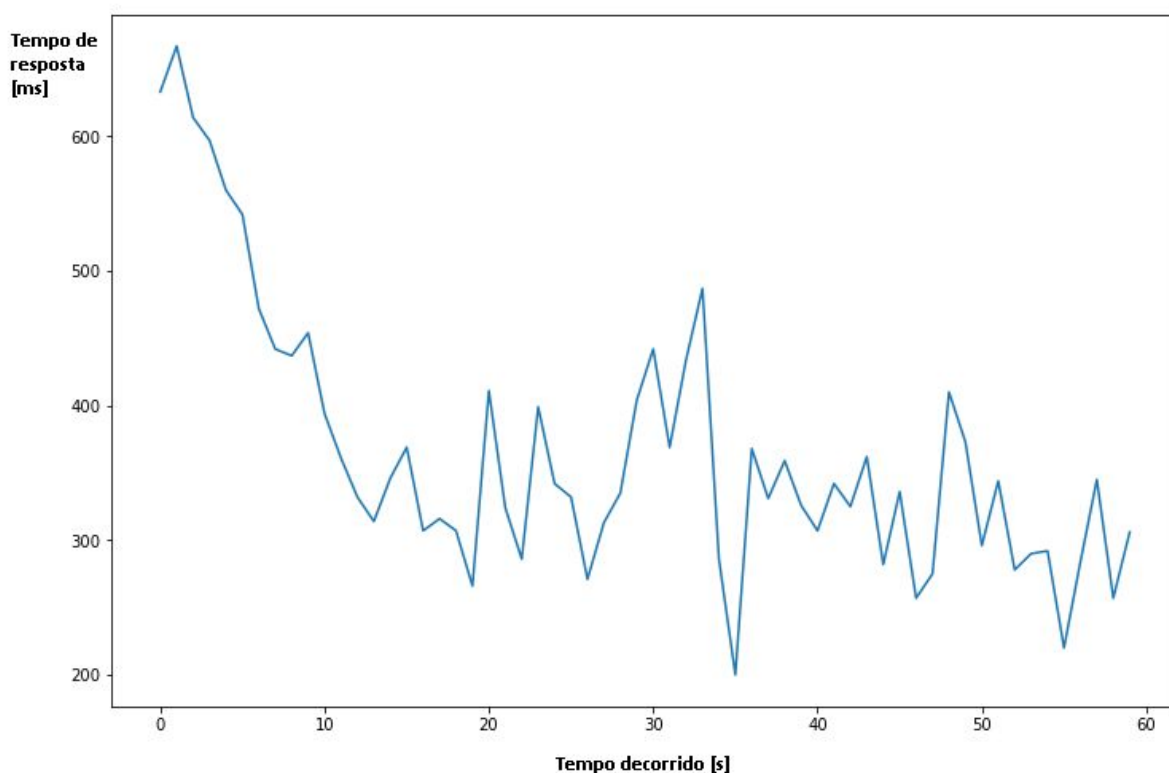
5. Resultados e próximos passos

5.1 Resultados

O sistema desenvolvido atende os requisitos apresentados no item 3 desta monografia, permitindo operações de autenticação e voto. O sistema foi testado com cerca de 15 colegas que funcionaram como o grupo de alpha-testers, dos quais foram discutidos a demanda e credibilidade que a plataforma passa, bem como eventuais bugs que foram encontrados.

Efetuuou-se testes de carga de acordo com a demanda para a qual o sistema foi projetado conforme especificado no item 3.1, ou seja, para uma demanda de 1000 requisições de leitura por minuto. Apenas a requisição de leitura foi testada, visto que costuma ser o gargalo nestes tipos de plataforma. Com o site hospedado em um computador pessoal, requisitou-se, de outra rede, uma demanda referente a 1000 leituras de enquetes equidistribuídas ao longo de 1 minuto, tendo-se o gráfico referente ao tempo de resposta conforme abaixo (as requisições sem resposta foram descartadas).

Figura 25 - Tempo de resposta por requisição ao longo de 1 minuto



Também se obteve os seguintes parâmetros de tempo de resposta conforme tabela abaixo.

Tabela 2 - Tempos de resposta por requisições de leitura

Tempo de resposta pico (máximo)	720 ms
Tempo de resposta médio	350 ms
Porcentagem de requisições perdidas (Error Rate)	0,8%

Conclui-se que os tempos de resposta são satisfatórios e permitem o funcionamento quase que imperceptível para os usuários [\[22\]](#).

5.2 Próximos passos

A plataforma ainda não foi desenvolvida por completo e ainda necessita de algumas funcionalidades antes de sua divulgação para o público para qual é destinada. Em específico é uma pendência a funcionalidade de criação de enquetes por parte dos usuários bem como a visualização dos usuários que votaram nas enquetes. Ainda faltam mais testes para verificar a funcionalidade correta da

plataforma internacionalmente, com configurações corretas de load balancers para tal, etc.

6. Conclusão

Este trabalho de conclusão de curso permitiu relacionar conhecimentos acumulados na Poli-USP, na indústria de software onde estagiei e das competições que participei. Foi uma oportunidade para demonstrar espírito empreendedor e conhecimento técnico alinhados para atender uma demanda com claro potencial. Foram relacionados conceitos de eficiência, sistemas distribuídos, engenharia de software, banco de dados, microserviços, entre outros

Referências

- [1] Rob Conery, et. al. **Search Microsoft Application Architecture Guide, 2nd Edition**. Estados Unidos: 2009, Chapter 2: Key Principles of Software Architecture.
- [2] Martin Fowler. **Microservices**. Disponível em <https://martinfowler.com/articles/microservices.html>> Acesso em: 2 dez. 2020.
- [3] Martin Fowler. **Microservice Trade-Offs**. Disponível em <https://www.martinfowler.com/articles/microservice-trade-offs.html#ops>>. Acesso em 2 dez. 2020.
- [4] **Codeforces**. Disponível em <https://codeforces.com/>>. Acesso em 2 dez. 2020.
- [5] Problem Likeability Feature. Disponível em [Problem Likeability Feature - Codeforces](#)>. Acesso em 2 dez. 2020.
- [6] Design the Twitter timeline and search. Disponível em [system-design-primer/README.md at master · donnemartin/system-design-primer · GitHub](#)>. Acesso em 2 dez. 2020.
- [7] Flanagan, David. **JavaScript - The definitive guide (6 ed.)**. p. 1
- [8] IBM. **JavaScript Everywhere and the Three Amigos (Into the wild BLUE yonder!)**. 24 October 2013.
- [9] Wikipedia. **Node.js**. Disponível em [Node.js - Wikipedia](#)>. Acesso em 2 dez. 2020.
- [10] **Codeforces API introduction**. Disponível em <https://codeforces.com/apiHelp>>. Acesso em 2 dez. 2020.
- [11] **Codeforces Visualizer**. Disponível em <https://cfviz.netlify.app/>>. Acesso em 2 dez. 2020.

- [12] **Code-drills**. Disponível em <[code-drills - Competitive programming resources and problem recommender](#)> . Acesso em 2 dez. 2020.
- [13] **Stackoverflow**. Disponível em <<https://stackoverflow.com/>> . Acesso em 2 dez. 2020.
- [14] **Inner circle**. Disponível em <<https://theinnercircle.co/>> . Acesso em 2 dez. 2020.
- [15] **Conta no Instagram de Neymar Jr.** Disponível em <<https://www.instagram.com/neymarjrsiteoficial/>> . Acesso em 2 dez. 2020.
- [16] **Conta do Victor Lamarca no Codeforces**. Disponível em <<https://codeforces.com/profile/VLamarca>> . Acesso em 2 dez. 2020.
- [17] **Placar final da primeira fase da Maratona de Programação 2020**. Disponível em <<http://maratona.sbc.org.br/primfase20/reports/ScoreBrasil.html>> . Acesso em 2 dez. 2020.
- [18] **Placar final do IEEEExtreme Programming Competition 2020**. Disponível em <<https://www.facebook.com/IEEEExtreme/photos/a.170045896342493/5182209081792791/>> . Acesso em 2 dez. 2020.
- [19] **Placar brasileiro da Google Code Jam 2020**. Disponível em <https://codingcompetitions.withgoogle.com/codejam/round/000000000019ff7e?scoreboard_type=Brazil> . Acesso em 2 dez. 2020.
- [20] **60 Incredible and Interesting Twitter Stats and Statistics**. Disponível em <<https://www.brandwatch.com/blog/twitter-stats-and-statistics>> . Acesso em 2 dez. 2020.
- [21] **The Master-Slave Database Concept For Beginners**. Disponível em <[The Master-Slave Database Concept For Beginners | Data Driven Investor](#)> . Acesso em 2 dez. 2020.

[22] **Response Time Testing**. Disponível em <[What is Response Time Testing? \(guru99.com\)](https://www.guru99.com/response-time-testing.html)> . Acesso em 2 dez. 2020.