



UNIVERSIDADE FEDERAL DO CEARÁ - CAMPUS SOBRAL

CURSO DE ENGENHARIA DE COMPUTAÇÃO

DISCIPLINA: TECNOLOGIAS WEB II

PROFESSOR: THIAGO IACHILEY ARAUJO DE SOUZA

RELATÓRIO TÉCNICO: TRABALHO 01 DE TECNOLOGIAS WEB 2

ALUNO - MATRÍCULA

GABRIEL VASCONCELOS SANTOS - 497688

IZAIAS MACHADO PESSOA NETO - 497372

MARCOS VINÍCIUS ANDRADE DE SOUSA - 496788

WILLIAM BRUNO SALES DE PAULA LIMA - 497345

YANN LUCCA MIRANDA MARTINS BARROS - 497746

SOBRAL

2024

SUMÁRIO

1	INTRODUÇÃO	3
2	ARQUITETURA DA APLICAÇÃO	4
3	IMPLEMENTAÇÃO	5
4	USABILIDADE	10
4.1	Funcionalidades públicas	10
4.2	Login e cadastro de usuários	12
4.3	Funcionalidades protegidas	13
5	CONCLUSÃO	16

1 INTRODUÇÃO

Com o avanço contínuo da tecnologia e a crescente demanda por soluções web interativas e eficientes, o desenvolvimento de aplicações que facilitam a comunicação e o compartilhamento de conhecimento tornou-se essencial. Este trabalho, encontra-se em um repositório do Github¹ e foca na criação de uma aplicação web que consiste em um fórum projetado para permitir que os usuários publiquem perguntas e respostas, promovendo um ambiente colaborativo de troca de informações.

O objetivo principal deste projeto é demonstrar a viabilidade de construir um site responsivo e dinâmico utilizando um conjunto robusto de tecnologias modernas. Entre as ferramentas e linguagens escolhidas estão Node.js, MySQL, Docker, EJS, Prisma, HTML, CSS e JavaScript. Cada uma destas tecnologias desempenha um papel crucial na estruturação e no funcionamento da aplicação, garantindo não apenas uma experiência de usuário fluida e intuitiva, mas também uma infraestrutura sólida e escalável.

Node.js oferece uma plataforma eficiente para o desenvolvimento do *back-end*, permitindo a criação de um servidor rápido e capaz de lidar com múltiplas requisições simultâneas. MySQL, um dos sistemas de gerenciamento de banco de dados mais populares, fornece uma base de dados relacional confiável para armazenar e gerenciar as informações dos usuários e suas respectivas postagens. O Docker foi utilizado para inicializar o banco de dados, facilitando a implantação e o gerenciamento de ambientes de desenvolvimento e produção consistentes.

Além disso, o EJS (Embedded JavaScript) foi empregado para a renderização de páginas dinâmicas no lado do servidor, permitindo a criação de interfaces interativas e atualizáveis. O Prisma, um ORM (Object-Relational Mapping), foi integrado para simplificar o gerenciamento das operações de banco de dados, proporcionando uma camada de abstração eficiente e segura.

Além disso, a aplicação² proposta visa ilustrar a importância da colaboração entre desenvolvedores e a comunidade de usuários, oferecendo uma plataforma onde perguntas podem ser respondidas de maneira eficiente, contribuindo para o crescimento coletivo do conhecimento.

¹ <https://github.com/izaiasmachado/guia-de-perguntas>

² <https://queueunderflow-test.izaias.me>

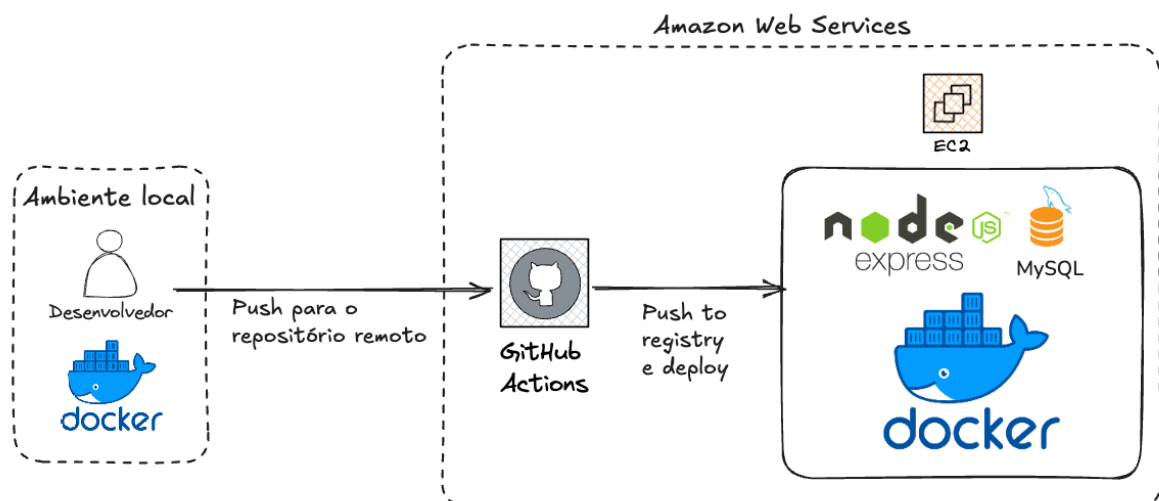
2 ARQUITETURA DA APLICAÇÃO

Como ilustrado na Figura 1, a arquitetura da solução consiste em uma aplicação *back-end* que é hospedada em um servidor, utilizando Node.js, em conjunto com o *framework web* Express.js. Além disso, a aplicação conversa com um banco de dados MySQL, persistindo e consumindo dados do com base nas ações realizadas pelo usuário. A aplicação consiste em um servidor construído com as tecnologias citadas anteriormente para comportar toda a lógica do fórum, e utiliza a tecnologia EJS para renderizar todos os componentes no navegador do usuário. Além disso, a aplicação foi hospedada em uma máquina virtual EC2 hospedada na AWS.

Para realizar o desenvolvimento, teste e publicação da aplicação na máquina virtual da AWS, o desenvolvedor deve hospedar o banco de dados da aplicação em seu ambiente local utilizando o Docker. O desenvolvedor deve então construir a nova funcionalidade que deseja criar dentro da aplicação, testar localmente e verificar se toda a aplicação está funcionando. Após testar a nova funcionalidade com sucesso, o desenvolvedor deve enviar suas alterações feitas para revisão para outros desenvolvedores do time por meio do GitHub.

Por fim, após a revisão das mudanças ter sido finalizada com sucesso no GitHub, a plataforma aciona um gatilho para publicar em uma máquina virtual EC2 da AWS utilizando o Docker para construir, implantar e iniciar um *container* dentro do servidor.

Figura 1 – Arquitetura da solução

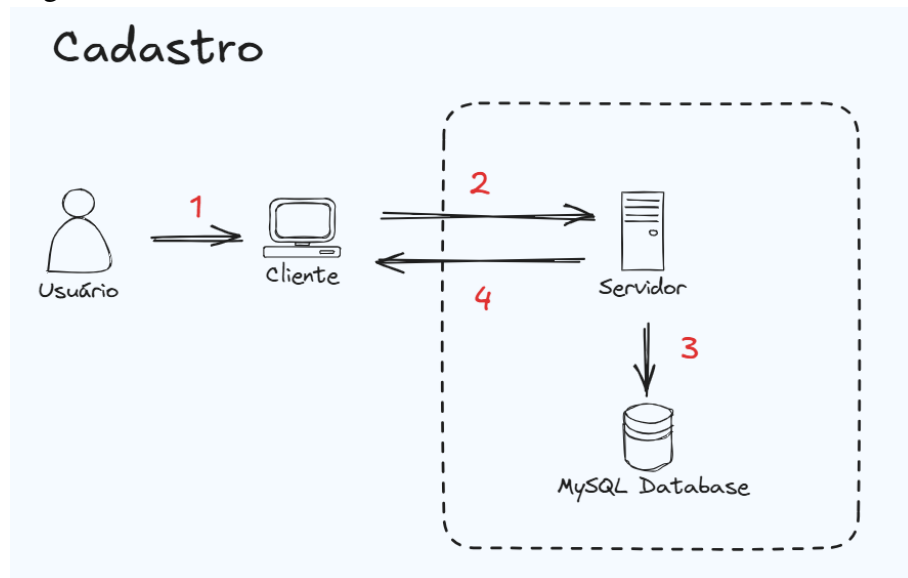


3 IMPLEMENTAÇÃO

A ação de realizar o cadastro do usuário se inicia no momento em que o mesmo acessa a aplicação e ainda não possui o *login* efetuado na plataforma. Por conta disto, o usuário não conseguirá realizar, visualizar e nem responder perguntas. Para tanto, o indivíduo deve clicar no botão de cadastro e fornecer seu nome, senha e e-mail para cadastro no fórum.

Em seguida, os dados fornecidos pelo usuário são enviados do navegador para o servidor onde a aplicação está hospedada. Os dados são salvos dentro do banco de dados utilizando a tecnologia *Prisma*. Após isto, o servidor redireciona o usuário para a tela de *Login*. Todo o fluxo de cadastro pode ser visualizado na Figura 2.

Figura 2 – Funcionalidade de cadastro de usuário

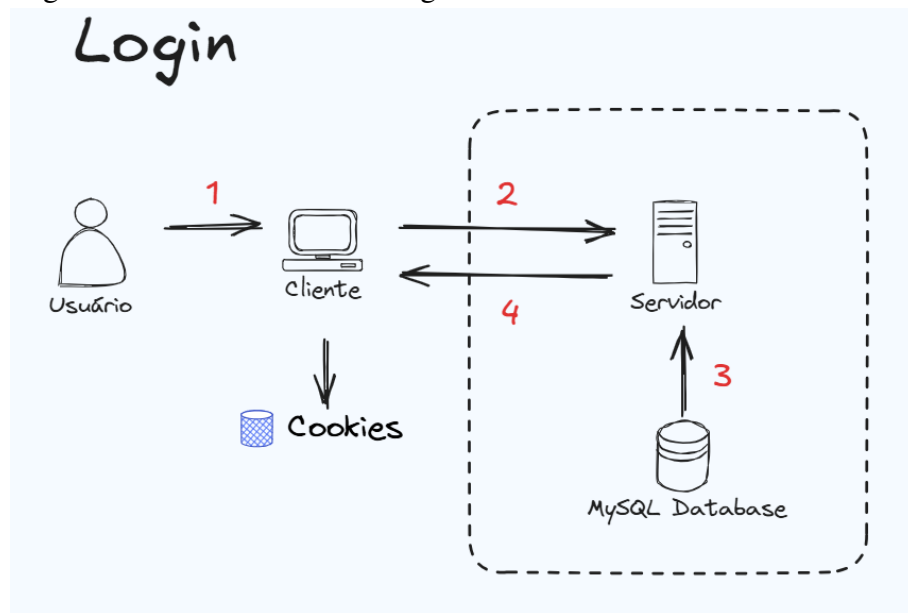


A Figura 3 ilustra o fluxo de atividades para que o usuário realiza o *login* na aplicação. Após ter realizado o cadastro, o usuário é redirecionado para a página de *Login*. Nesta página, o usuário insere no navegador o mesmo e-mail e senha que ele utilizou no momento de se cadastrar. Por conseguinte, os dados fornecidos pelo usuário são enviados ao servidor, que realiza uma consulta no banco de dados para verificar se o e-mail e senha fornecidos condizem corretamente com as informações fornecidas no momento do cadastro.

Por fim, caso os dados fornecidos na etapa de login condizam com as informações salvas no banco de dados de cadastro, o servidor redireciona o usuário para a tela inicial do fórum, onde são ilustradas todas as perguntas feitas no mesmo. Vale ressaltar que as senhas salvas no banco de dados e processadas no lado do servidor estão criptografadas utilizando a

tecnologia *Bcrypt*, a fim de garantir a segurança das informações transacionadas na aplicação.

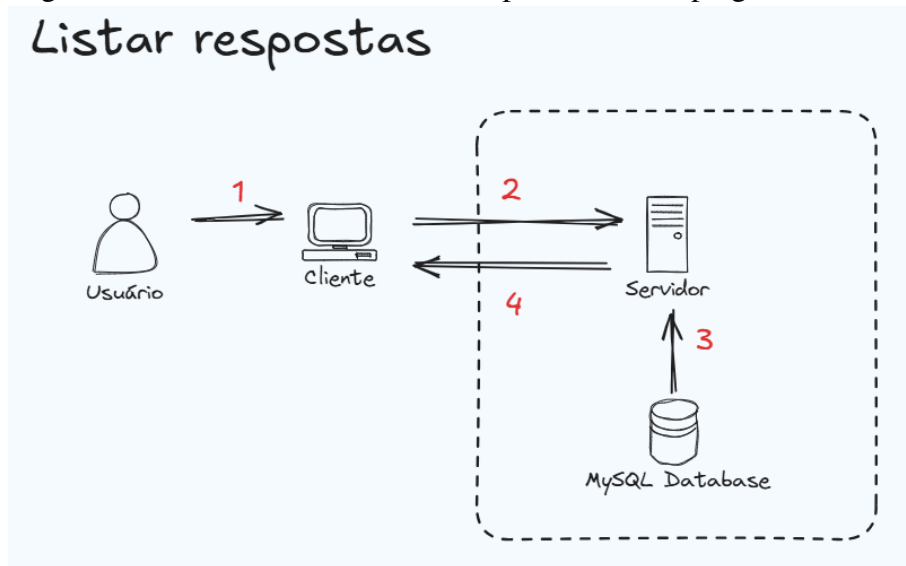
Figura 3 – Funcionalidade de login



A Figura 4 ilustra a funcionalidade de listar respostas. Dentro da funcionalidade de listas respostas, o usuário pode selecionar qualquer pergunta que ele esteja visualizando na página inicial do fórum. Assim, é enviada uma requisição enviada pelo navegador para o servidor para que o mesmo renderize a página contendo a pergunta selecionada e as suas respectivas respostas já postadas na aplicação.

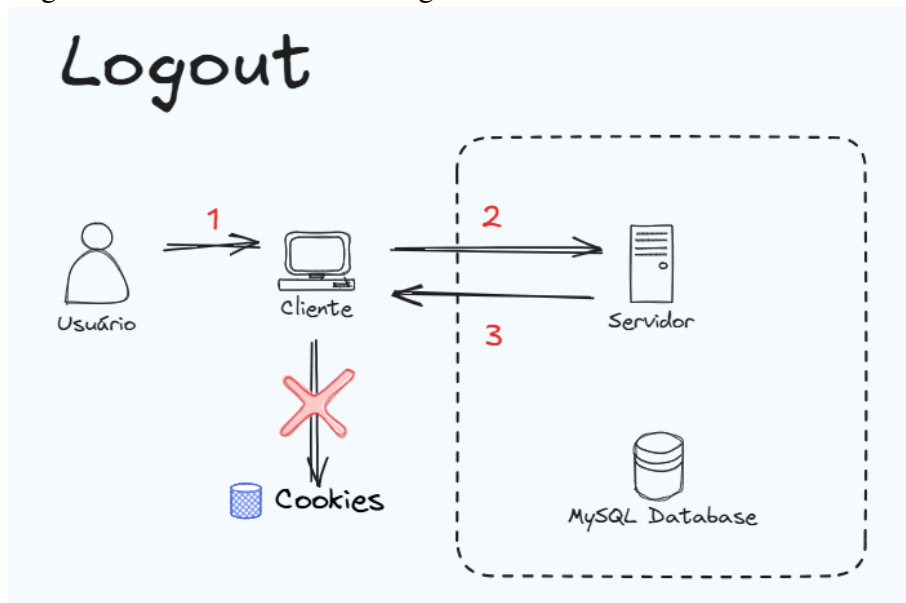
O servidor então consulta o banco de dados da maneira já citada previamente, de forma a carregar todas as informações das respostas já postadas para a pergunta requisitada. O servidor renderiza a página da pergunta selecionada com todas suas respostas e o usuário as visualiza no seu navegador.

Figura 4 – Funcionalidade de listar respostas de uma pergunta



A Figura 5 apresenta a funcionalidade que permite que o usuário consiga sair da aplicação. Inicialmente, o usuário clica no botão para sair da aplicação. Com isso, uma requisição é feita para o servidor solicitando que essa ação seja realizada. Por sua vez, o servidor retorna um *header* solicitando que o *cookie* de nome “authorization” seja apagado. Com isso, os *cookies* são apagados e o servidor redireciona o usuário para a tela de *login*.

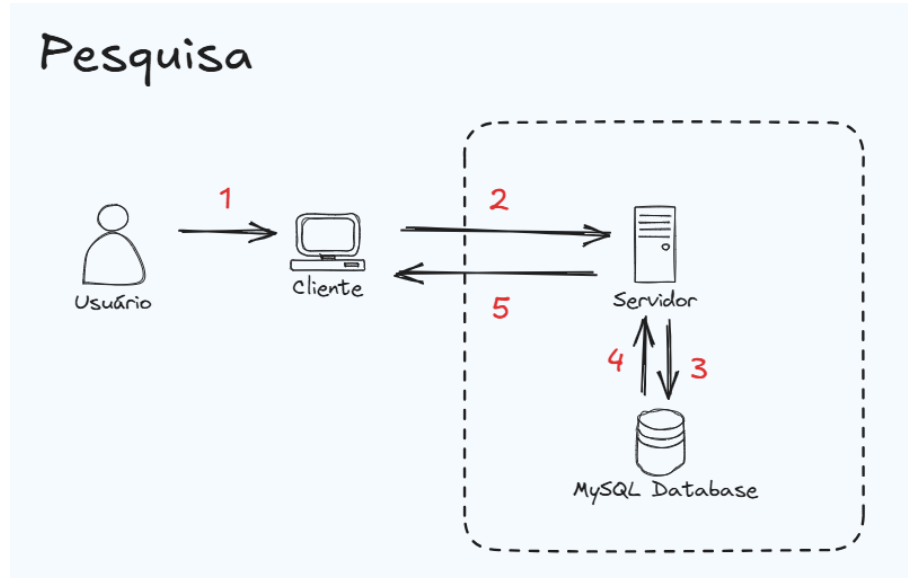
Figura 5 – Funcionalidade de logout



Já na Figura 13, é apresentado o fluxo de dados da funcionalidade de pesquisa. Inicialmente, o usuário digita uma frase de busca em seu navegador. Essa frase de busca é enviada para o servidor que por conseguinte faz uma por meio dessa frase no banco de dados de

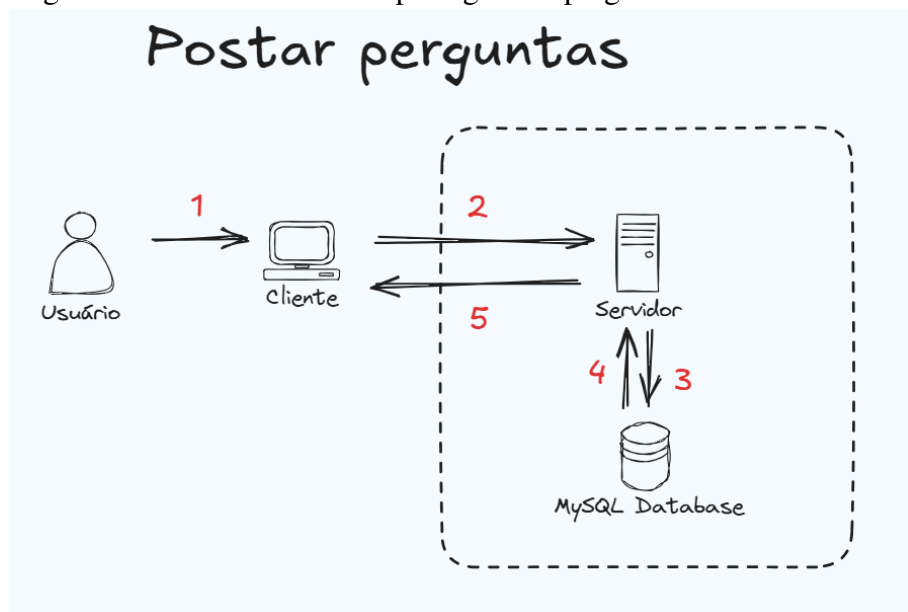
perguntas. Após a busca, caso existam resultados, eles são retornados para o servidor que logo em seguida devolve a página para o cliente.

Figura 6 – Funcionalidade de pesquisa



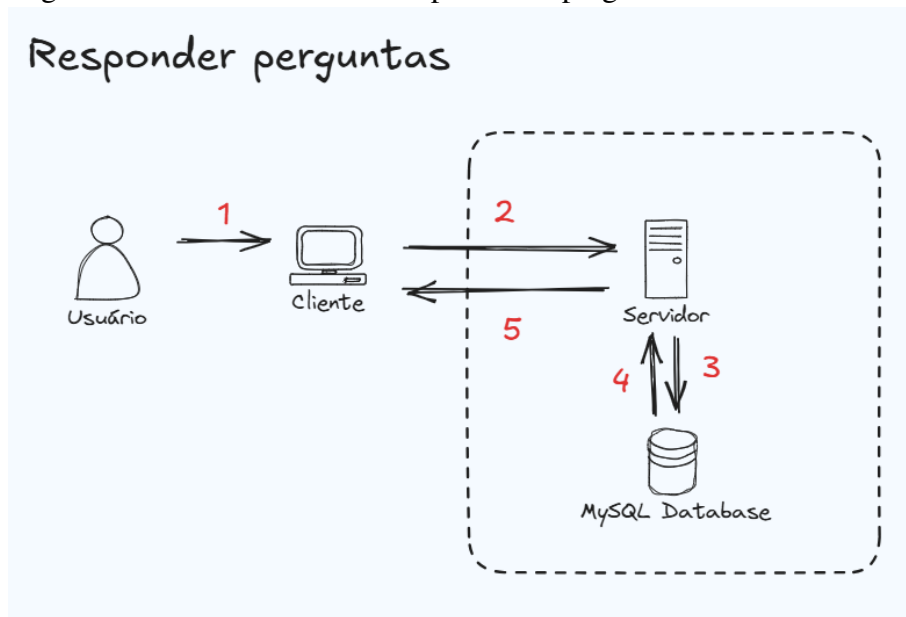
Na Figura 7 é apresentada a funcionalidade de postagem de perguntas. O fluxo de dados dessa implementação consiste no usuário preencher o dados de uma pergunta, que são o título e o corpo da pergunta. Uma vez clicado no botão de postar, a nova pergunta é enviada ao servidor que faz as devidas validações dos dados inseridos. Passadas as validações, o servidor persiste os dados no banco de dados e retorna a nova pergunta com seu identificador único para o servidor. O servidor, por sua vez, redireciona o usuário para a página da pergunta recém criada.

Figura 7 – Funcionalidade de postagem de perguntas



Por fim, a Figura 8 mostra a ilustração do fluxo de dados para a resposta de uma pergunta. Inicialmente o usuário se encontra na página da pergunta e preenche os dados da resposta que deseja postar. Ao clicar no botão de postar os dados preenchidos no navegador do usuário são enviados ao servidor. Os dados da resposta são persistidos no servidor e na resposta é retornado o identificador único da resposta e também dados de hora de criação e atualização. A página da pergunta é renderizada novamente, mostrando a resposta recém postada.

Figura 8 – Funcionalidade de responder as perguntas



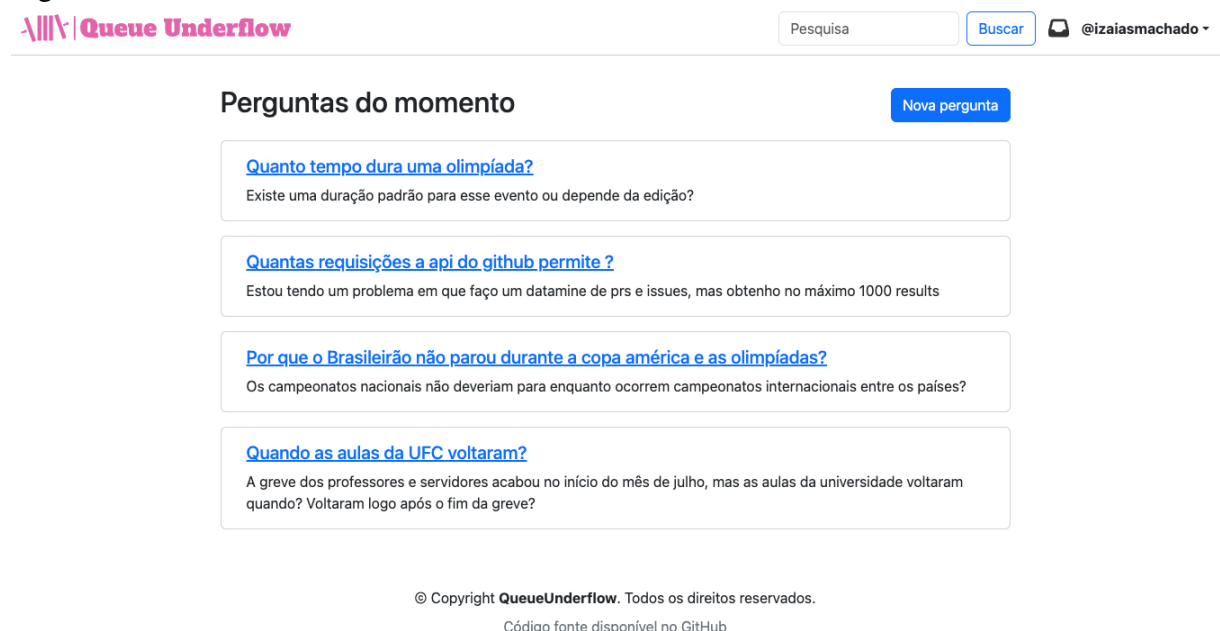
4 USABILIDADE

Este capítulo apresenta uma visão geral das interfaces das implementações feitas, o que abrangem as funcionalidades públicas (Seção 4.1), login e cadastro de usuários (Seção 4.2) e funcionalidades protegidas (Seção 4.3).

4.1 Funcionalidades públicas

A página inicial do Queue Underflow é apresentada na Figura 9, que mostra uma barra superior de navegação, uma lista de perguntas e um rodapé inferior.

Figura 9 – Tela inicial



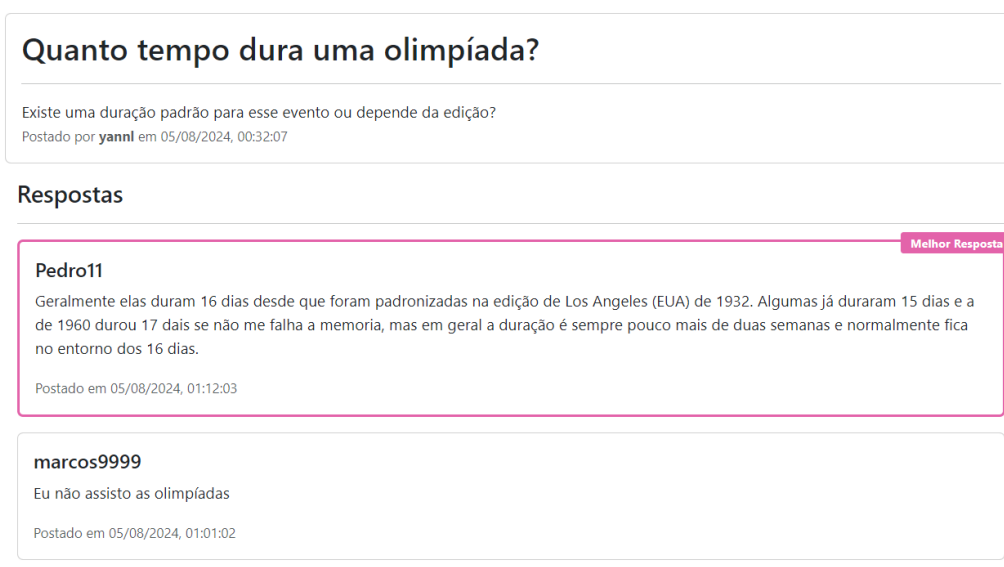
Para criar uma nova pergunta é possível clicar no botão azul que permite destacado por **(a)** na Figura 10. Já para visualizar as respostas de uma pergunta postada no QueueUnderflow é necessário clicar em uma das pergunta listadas na página inicial, conforme indicado por **(b)** na Figura 10.

Figura 10 – Selecionando uma pergunta



A Figura 11 apresenta a tela da pergunta, que mostra todas as respostas para aquela pergunta. Ainda na Figura 11, é destacado em rosa a melhor resposta para a pergunta. Essa escolha pode ser feita pelo autor da pergunta.

Figura 11 – Visualizando uma pergunta e suas respostas



A Figura 12 mostra um recorte da parte inferior da página da pergunta. Nessa imagem é apresentado que o usuário, contanto que esteja logado, pode responder a uma pergunta.

Figura 12 – Caixa de resposta

Adicionar uma resposta

Sua resposta

Enviar

Na Figura 13 foi destacado em vermelho a barra de pesquisa. Ao digitar uma frase de busca e clicar no botão buscar são apresentadas perguntas que tem a frase de busca como uma *substring*. Dessa forma, podem ser encontradas perguntas de acordo com o seu conteúdo.

Figura 13 – Realizar pesquisa

 **Queue Underflow**

UFC

Buscar



@izaiasmachado ▾

Perguntas do momento

Nova pergunta

[Quando as aulas da UFC voltaram?](#)

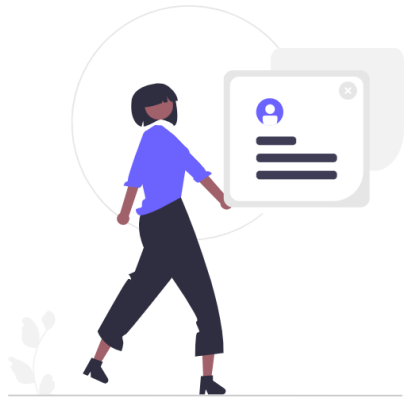
A greve dos professores e servidores acabou no início do mês de julho, mas as aulas da universidade voltaram quando? Voltaram logo após o fim da greve?

4.2 Login e cadastro de usuários

Ao tentar realizar uma ação protegida sem estar logado, o usuário é redirecionado para a página apresentada na Figura 14 e é informado por meio de um alerta que é necessário estar logado para realizar essa ação. Isso acontece ao tentar postar uma pergunta e responder a uma pergunta sem estar logado. É importante ressaltar que caso o usuário não tenha uma conta, é possível clicar no botão “Criar conta” para ser redirecionado para a tela de cadastro.

Já na Figura 15 é apresentada a tela de cadastro de usuários, por meio da qual o usuário consegue criar uma nova conta. Após o preenchimento dos campos com dados válidos e realizado o cadastro, o usuário é redirecionado para a tela de login.

Figura 14 – Tela de login



Faça login para continuar sua navegação

E-mail

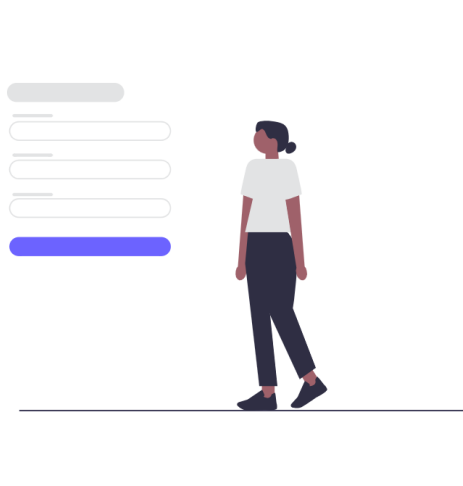
Senha

Entrar

OU

[Criar conta](#)

Figura 15 – Tela de cadastro de usuários



Nome

Nome de usuário

E-mail

Senha

Criar conta

OU

[Já tenho uma conta](#)

4.3 Funcionalidades protegidas

Com o usuário logado, é possível criar suas próprias perguntas, conforme apresentado na Figura 16. Nessa imagem é possível preencher um título e descrição de uma pergunta. Após isso, a nova pergunta aparece na tela inicial. Para acessar a página de envios de perguntas basta acessar a página inicial e seguir a explicação feita na Seção 4.1.

Figura 16 – Tela de enviar perguntas
Realizar Pergunta

Título

Quando as aulas da UFC voltaram?

Descrição

A greve dos professores e servidores acabou no início do mês de julho, mas as aulas da universidade voltaram quando? Voltaram logo após o fim da greve?

Enviar pergunta

Após receber respostas em uma pergunta publicada pelo usuário, é possível escolher uma resposta para ser mostrada como a melhor resposta. A Figura 17 apresenta a visualização de quem criou uma pergunta, sendo possível marcar ou desmarcar uma resposta como a melhor.

Figura 17 – Marcar e desmarcar como melhor resposta

Quando as aulas da UFC voltaram?

A greve dos professores e servidores acabou no início do mês de julho, mas as aulas da universidade voltaram quando? Voltaram logo após o fim da greve?

Postado por izaiaismachado em 05/08/2024 às 02:43

Respostas

yannl Melhor Resposta

Na verdade, as aulas da UFC voltaram no dia 15 e não logo após o fim da greve. Houve um pequeno período de tempo para os alunos se prepararem para a volta.

Postado em 04/08/2024 às 22:22

Desmarcar como melhor resposta (a)

Pedro11

Acho que foi lá pro dia 20, sei lá kkkkkkkkk

Postado em 05/08/2024 às 07:10

Marcar como melhor resposta (b)

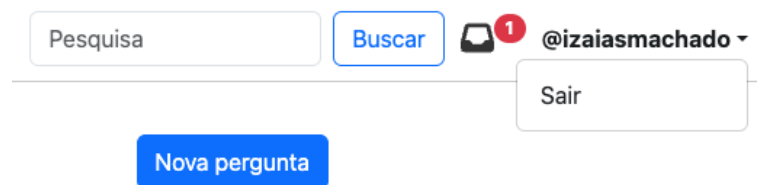
Quando um usuário recebe respostas em uma pergunta, é possível acessar sua caixa de entrada na barra de superior. A Figura 18 mostra a caixa de entrada do usuário aberta. Além disso, ao clicar na notificação, o usuário é redirecionado para a página da pergunta que respondeu a dada resposta.

Figura 18 – Caixa de entrada



Ao realizar todas as ações desejadas, o usuário pode, por fim, sair da aplicação. Para fazer isso, basta clicar no nome de usuário e na opção “Sair” do menu apresentado na Figura 19. Após o sair da aplicação o usuário é redirecionado novamente para a tela de *login*.

Figura 19 – Menu para sair



5 CONCLUSÃO

Neste trabalho, foi desenvolvida uma aplicação web para postagem de perguntas e respostas de diferentes usuários, utilizando tecnologias modernas e eficientes. O objetivo principal foi demonstrar a viabilidade de construir um site responsivo empregando Node.js, MySQL, Docker, HTML, CSS e JavaScript. Adicionalmente, foram utilizados EJS para renderização do lado do servidor e Prisma como ORM para interagir com o banco de dados.

O servidor web foi desenvolvido para gerenciar requisições HTTP, servir arquivos estáticos e dinâmicos, além de garantir respostas adequadas, como erros 404 para arquivos não encontrados. A interface de usuário foi projetada e estilizada para oferecer uma experiência agradável e intuitiva aos usuários, com funcionalidades como cadastro, login, postagem de perguntas e respostas, pesquisa e logout.

Para garantir a segurança das senhas dos usuários, foi utilizada a criptografia com Bcrypt. A integração com o banco de dados feita pelo uso do Prisma, proporcionando uma camada de abstração eficiente para operações de CRUD. Docker foi empregado para inicializar e gerenciar o ambiente do banco de dados, garantindo consistência e facilidade de configuração.

Além das tecnologias mencionadas, o desenvolvimento foi conduzido utilizando boas práticas de gerenciamento de projetos e controle de versão. Git foi a ferramenta de controle de versão adotada, permitindo um desenvolvimento colaborativo e organizado através de pull requests (PRs). O Kanban do GitHub Projects foi utilizado para a gestão das tarefas, facilitando a visualização do progresso e a organização do fluxo de trabalho.

A aplicação resultante é um sistema robusto e seguro, que atende aos requisitos funcionais e oferece uma plataforma eficiente para interação dos usuários. O desenvolvimento seguiu padrões de qualidade, incorporando práticas modernas de engenharia de software para assegurar a manutenção e escalabilidade futura do projeto.

Por fim, a aplicação foi implantada em uma máquina virtual EC2 da AWS e está disponível em dois ambientes, um de testes¹ e outro de produção².

¹ <https://queueunderflow-test.izaias.me>

² <https://queueunderflow.izaias.me>