



Etec
Adolpho Berezin
Mongaguá



SÃO PAULO
GOVERNO DO ESTADO

ADRIANE DA COSTA SANTOS
JULIANE DE ALMEIDA AMORIM
LEONARDO GONSALEZ

FOOCO - FÓRUM DE ESTUDOS
Equipe Animalia

Orientador Prof. Graciete Henriques dos Santos

MONGAGUÁ

07/2023

**ADRIANE DA COSTA SANTOS
JULIANE DE ALMEIDA AMORIM
LEONARDO GONSALEZ**

**FOOCO - FÓRUM DE ESTUDOS
Equipe Animalia**

Trabalho de Conclusão de Curso
apresentado à Escola Técnica Adolpho
Berezin, como parte dos requisitos para a
obtenção do título de Técnico em
Informática.

Orientadora Profa.: Graciete Henriques
dos Santos

MONGAGUÁ

07/2023

*Foi pensando nas pessoas que se executou este projeto,
por isso dedicamos este trabalho a todos aqueles a quem
esta aplicação possa ajudar de alguma forma.*

AGRADECIMENTOS

Agradecemos todos os amigos e familiares cujo apoiaram e deram forças para que a realização do trabalho fosse possível, o forte apoio dos professores, pelas correções e ensinamentos que nos permitiram apresentar um melhor desempenho no nosso processo de formação profissional ao longo do curso, agradecimentos especiais aos orientadores Kleyton Sartori e Graciete Henriques pela paciência e sabedoria. Em suma, a equipe Animalia agradece todos aqueles que contribuíram, de alguma forma, para a realização deste trabalho.

Juntos somos mais fortes, assim se muda o mundo.

A Equipe

RESUMO

O presente projeto "Fooco" é um fórum acadêmico que tem como principal objetivo promover a cooperação dos alunos em instituições de ensino, desenvolvido principalmente pensando no Centro Paula Souza e os cursos que são oferecidos pela instituição, de nível médio, técnico e superior.

Visando ser uma solução para a evasão escolar por falta de integração, e/ou dificuldade de associar os conteúdos passados, o aplicativo estimula a socialização entre os estudantes, fazendo com que cada um possa ajudar com o conhecimento que possui e receber ajuda quando preciso, é um sistema de perguntas e respostas classificadas por cursos e temas, que conta com um método de gamificação como forma de incentivar o usuário a interagir no fórum.

Se objetiva, contudo, promover a socialização, cooperação e a troca de conhecimento em ambientes de ensino, de forma que traga cidadania e empatia com o problema do próximo, também se visa auxiliar as modalidades EAD, possibilitando uma maior integração entre os estudantes a distância. "Aprendemos mais juntos"

PALAVRAS-CHAVES: Cooperação, solução, fórum.

ABSTRACT

The present project "Fooco" is a forum whose main objective is to promote the cooperation of students in secondary and technical education institutions.

Aiming to be a solution for school dropout due to lack of integration, and/or difficulty in associating past contents, the application stimulates socialization among students, making each one of them able to help with the knowledge they have and receive help when needed, is a system of questions and answers classified by rooms and themes, which has a gamification method to encourage the user to interact in the forum.

If objectified, however, it aims to promote socialization, cooperation, and the exchange of knowledge in educational environments, in a way that brings citizenship and empathy towards the issues of others. It also seeks to support online learning modalities, enabling greater integration among remote students. "We learn more together."

Key Words: Cooperation, solution, forum.

LISTA DE FIGURAS

Figura 1 - Logo da Equipe	21
Figura 2 - Logo do Projeto	21
Figura 3 - Modelagem do Banco de Dados	23
Figura 4 – Wireframe da Tela Inicial	31
Figura 5 - Wireframe do Dashboard	32
Figura 6 - Wireframe da Tela Não encontrado.....	33
Figura 7 - Tela Inicial do Foco	34
Figura 8 - Tela de Login.....	35
Figura 9 - Tela para criar Conta.....	35
Figura 10 - Tela do Dashboard.....	36
Figura 11 - Tela de Publicação e Comentário.....	37
Figura 12 - Tela de Denunciar Publicação.....	38
Figura 13 - Tela para criar Publicação.....	38
Figura 14 - Tela para criar Publicação.....	39
Figura 15 - Tela de Perfil do Usuário	39
Figura 16 - Tela de Publicações Salvas.....	40
Figura 17 - Tela de Pesquisa.....	40
Figura 18 - Tela de Configurações	41
Figura 19 - Trecho do código fonte do Endpoint da API	42
Figura 20 - Trecho do código fonte com integração de API.....	43
Figura 21 - Trecho do código fonte do Ranking.....	44

LISTA DE GRÁFICOS

Gráfico 1 - Você tem dificuldade em compreender os conteúdos passados em sala?.....	13
---	----

SUMÁRIO

LISTA DE FIGURAS	7
LISTA DE GRÁFICOS.....	7
INTRODUÇÃO	11
1. NICHU DE MERCADO DO PROJETO	12
1.1. Cliente para Estudo de Aplicação – Etec Adolpho Berezin	13
1.2. Problema do Cliente	13
1.3. Solução Proposta	14
2. TECNOLOGIAS UTILIZADAS	15
2.1. Front-End	15
2.1.1. <i>Figma</i>	15
2.1.2. <i>HTML</i>	15
2.1.3. <i>Styled Components</i>	16
2.1.4 <i>Typescript</i>	16
2.1.5 <i>React</i>	17
2.1.6 <i>React Router</i>	17
2.1.7 <i>Radix Primitives</i>	18
2.2. Back-End.....	18
2.2.1. <i>Typescript</i>	18
2.2.2. <i>Express</i>	18
2.2.3. <i>Postgres</i>	19
2.2.4 <i>TypeORM</i>	19
3. EMPRESA	20
3.1 Missão	20
3.2 Visão	20
3.3 Valores	20
3.4 Logo21	
3.4.1 <i>Logo do Projeto</i>	21
3.5 Slogan	21
4. ANÁLISE	22

4.1. Descrição das Funcionalidades.....	22
4.1.1 <i>Publicação</i>	22
4.1.2 <i>Salvar para mais tarde</i>	22
4.1.3 <i>Ranking</i>	22
4.1.4 <i>Comentários</i>	23
4.2. MER – Modelo do Banco de Dados	23
4.3. Create do Banco de Dados	23
4.3.1 <i>Migration do banco de dados (PostgreSQL)</i>	24
4.3.1 <i>Entidade do usuário</i>	25
4.3.3 <i>Entidade da publicação</i>	27
4.3.4 <i>Entidade do comentário</i>	28
4.4 Principais Selects do Banco de Dados.....	29
4.4.1 <i>Users</i>	29
4.4.2 <i>Posts</i>	29
4.4.3 <i>Comments</i>	30
4.5. Wireframe das Telas	31
4.5.1 <i>Tela inicial</i>	31
4.5.2 <i>Dashboard</i>	32
4.5.3 <i>Não encontrado (404)</i>	33
4.6 Prints das Telas.....	34
4.6.1 <i>Tela inicial</i>	34
4.6.2 <i>Entrar</i>	35
4.6.3 <i>Criar conta</i>	35
4.6.4 <i>Dashboard</i>	36
4.6.6 <i>Publicação e comentário</i>	37
4.6.10 <i>Denunciar publicação</i>	38
4.6.9 <i>Criar publicação</i>	38
4.6.12 <i>Editar publicação</i>	39
4.6.7 <i>Perfil do usuário</i>	39
4.6.11 <i>Publicações salvas</i>	40
4.6.8 <i>Pesquisa</i>	40
4.6.5 <i>Configurações</i>	41
4.7 Trecho do Código Fonte.....	42
4.7.1 <i>Endpoints da API</i>	42

4.7.2 Integração com a API para criar publicação	43
5.7.3 Componente do Ranking	44
5. MANUAL DO USUÁRIO	45
5.1 Como acessar o Fooco?	45
5.2 Como criar uma conta?	45
5.3 Conta criada, como interagir no fórum?	47
5.4 Como fazer uma publicação?	51
5.5 Como modificar meu perfil e configurações	52
5.6 Pesquisa por tags.....	55
CONCLUSÃO.....	56
REFERÊNCIA BIBLIOGRÁFICA	57

INTRODUÇÃO

Assim como no ambiente profissional, uma boa integração social dentro da escola é um ponto de muita importância para aumentar seu rendimento escolar, de acordo com a equipe de especialistas Faz Educação & Tecnologia:

Problemas de socialização também podem diminuir a frequência escolar. As dificuldades de socialização são, em geral, sintomas de um problema maior. Os alunos se tornaram mais inseguros e isolados durante a pandemia. Agora, precisam passar por desafios para voltar a fazer e manter amizades. (Faz Educação & Tecnologia, 2022),

pois desse modo, além do estudante não se sentir inseguro por estar em um lugar totalmente desconhecido, ele não se sentirá deslocado, impedindo que ele se distancie mais ainda da escola. De acordo com Cabral (2011), é muito comum que alunos deslocados dentro do ambiente escolar sofram com um pior rendimento, além de prejudicar sua socialização.

O projeto consiste em um fórum de estudos, com o intuito de gerar proatividade nos estudantes, além de integrar socialmente os alunos da rede escolar. O aplicativo contará com recurso de perguntas e repostas em que os estudantes poderão se comunicar, auxiliando-se em tarefas, dúvidas gerais ou sobre a escola. O fórum será um excelente lugar para que o estudante se sinta pertencente e confortável na escola, no qual ele poderá entender os conteúdos escolares e criar laços.

1. NICHOS DE MERCADO DO PROJETO

O aplicativo será direcionado para estudantes que procuram socialização ou melhorar seu rendimento escolar em cooperação, além daqueles que querem expandir ou criar uma rede de contatos (*networking*). Busca-se alcançar todos os estudantes de escolas brasileiras do Ensino Médio e Técnico com o auxílio de um eficiente programa de marketing e com a recomendação do aplicativo pelos usuários.

Segundo estatísticas do Inep/MEC,

Existem no Brasil 7.770.557 matriculados no ensino médio. Já no médio técnico temos 767.435 alunos em todo o país. Juntando os dois níveis (médio + médio técnico), temos 8.537.992 estudantes, desses, 248.066 realizam os cursos concomitantemente. (ABRES, 2021)

É importante ressaltar que público-alvo em questão possui um grande acesso à internet móvel. Sendo um dos grandes fatores responsáveis pelo aumento do uso da internet o isolamento social decorrente da COVID-19, que resultou numa maior procura das pessoas por fóruns na internet, a fim de desenvolver seus assuntos de interesse. De acordo com Isadora da Cunha Prado (2021):

A vida de todas as pessoas se transformou drasticamente e, junto ao isolamento social, houve um crescente uso das redes sociais. Segundo um estudo realizado pela Kantar (empresa especializada em pesquisas de mercado), no Brasil em 2020, as redes sociais alcançaram uma taxa de aumento de 40% no uso em plataformas como Facebook, WhatsApp e Instagram na pandemia. (PRADO, 2021)

Considerando que mais de 43 milhões de brasileiros acima de 12 anos, navegam na internet através de dispositivos móveis (DATAFOLHA, 2014), o acesso a esses recursos, não será um empecilho para a aplicação do projeto no Brasil.

O fórum será desenvolvido com o intuito de facilitar e incentivar a cooperação entre os estudantes das escolas brasileiras. Tem-se como objetivo desenvolver o aplicativo para ser acessível à maior quantidade de pessoas, utilizando estratégias como criar sistemas de acessibilidade para o auxílio de PCDs (Pessoas com Deficiência) e tornar o aplicativo totalmente gratuito.

É comum que pessoas busquem por sites ou aplicativos para se integrar socialmente e aprender mais sobre determinado assunto, seja para um hobby, fins profissionais ou para fins educacionais.

1.1. Cliente para Estudo de Aplicação – Etec Adolpho Berezin

A escola técnica “Adolpho Berezin” é uma instituição que iniciou seu funcionamento em 1995, oferecendo nos dias de hoje ensino médio de qualidade e cursos técnicos sem custo para aproximadamente 900 alunos de acordo com a diretora da escola, os quais passaram pelo processo de entrada chamado de “Vestibulinho”, uma prova de conhecimentos básicos com conteúdo de sexto a nono ano do ensino fundamental.

Realizando uma breve pesquisa de campo, feita entre os dias 23/02/2023 e 20/06/2023 que conta com o total de 76 respostas de estudantes dos variados cursos presentes na instituição, pode-se perceber que a maioria encontra sim dificuldade em entender os assuntos que foram citados nas aulas.

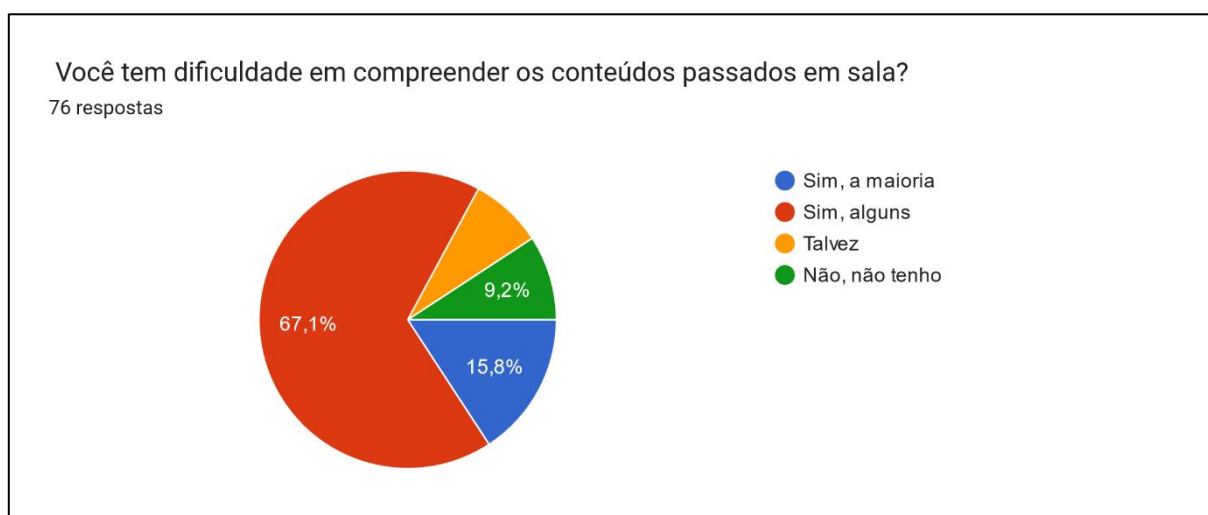


Gráfico 1 - Você tem dificuldade em compreender os conteúdos passados em sala?

Fonte: Aatoria da Equipe

1.2. Problema do Cliente

Dados do IBGE apontam que “em 2019, das 50 milhões de pessoas com idades entre 14 e 29 anos, 20% delas não tinham terminado alguma das etapas da educação básica”, e um dos principais motivos disso é a dificuldade de entender os conteúdos passados em aula (JORNAL NACIONAL, 2020).

Atualmente, muitos estudantes possuem enorme dificuldade em socializar-se, resultando na falta do desenvolvimento de habilidades extremamente importantes nos dias de hoje, como o trabalho em equipe, comunicar-se com eloquência e entender o momento certo de buscar e prover ajuda de forma eficiente.

Segundo o Hospital Santa Monica, a mudança repentina de rotina na quarentena trouxe à algumas pessoas diversas dificuldades de convívio social, afetou as emoções, trouxe insegurança, medo, frustração e angústia. Devemos manter as interações sociais, mesmo à distância, para driblarmos esses efeitos negativos. (HOSPITAL SANTA MÔNICA, 2020)

Portanto, entende-se que os benefícios do convívio social e da cooperação é de suma importância para o desenvolvimento escolar e pessoal, sendo a falta destes, um dos grandes agravantes para o regresso do aprendizado escolar.

1.3. Solução Proposta

Pretende-se desenvolver um fórum que traz a possibilidade de os estudantes cooperarem para solucionar suas dificuldades em comum, oferecendo a troca de conhecimentos entre si através do sistema de pergunta e resposta, comumente usados em fóruns da internet. Além disso, o aplicativo contará com um sistema denominado de gamificação (ou ludificação), no qual incentivará o usuário a participar ativamente da comunidade escolhida, lhe dando recompensas lúdicas ao responder dúvidas ou receber “likes”, usando um sistema de pontos, por exemplo. A gamificação é a aplicação dos princípios dos jogos em situações educativas, como sistemas de evolução ou de pontos, atraindo o jovem e os levando a querer jogar mais, e conseqüentemente, aprendendo mais. (EDUCADOR360, 2021)

De acordo com o site Educador360, os benefícios que o aluno pode ganhar com a gamificação são:

Os benefícios para os estudantes são imensos. Os jogos despertam o prazer de aprender, o protagonismo na aprendizagem, a autonomia, o desenvolvimento da memória e da assimilação de conteúdos, e ajudam a desenvolver as habilidades socioemocionais tão desejadas pelo mercado de trabalho do século 21. (EDUCADOR360, 2021)

Tendo em vista o problema gerado com a dificuldade do entendimento dos estudantes, o aplicativo visa, de maneira organizada, facilitar a busca e o compartilhamento de conhecimento em um único lugar. O fórum será dividido em grupos específicos para cada conhecimento, dessa forma, os usuários poderão se juntar a comunidade na qual estiverem mais interessados.

2. TECNOLOGIAS UTILIZADAS

A plataforma Foooco será desenvolvida utilizando as tecnologias de interfaces mais recentes e populares do mercado, com o intuito de obter um design moderno, intuitivo e simples, sendo compatível com o público jovem e estudante, e assim, criando uma experiência confortável para o usuário.

Para o desenvolvimento e o funcionamento “por trás dos panos” do aplicativo, será utilizado o Typescript juntamente com o React, sendo esses grandes aliados para a facilitação da criação do aplicativo, pois torna possível a obtenção de resultados mais concisos, seguros e com maior qualidade.

Segundo Kriger (2022), “Um *framework* é uma estrutura que serve de base para a construção de aplicações web de finalidade específica cujo desenvolvimento pode ser muito custoso e/ou problemático”. Kriger também diz que *frameworks* podem ser utilizados para construir aplicativos a partir de esqueletos pré-definidos, alterando apenas as informações desejadas.

2.1. Front-End

2.1.1. Figma

A principal ferramenta utilizada na prototipação e design da plataforma será o Figma, onde serão tomadas as decisões de *layout*, cores, tamanhos e tipos de fonte, além de ser o responsável por guiar o desenvolvimento das interfaces.

De acordo com Villain (2022), o Figma carrega diversos benefícios para a prototipação de designs, sendo possível explorar ao máximo da criação de interfaces e dos recursos existentes dentro da plataforma, além de ser uma ferramenta ágil, prática, gratuita e possuir uma excelente integração com ferramentas de desenvolvimento web, como o CSS ou React.

2.1.2. HTML

O HTML será utilizado como ferramenta de marcação de texto e da criação de acessibilidade para usuários com deficiências físicas ou visuais, sendo conhecido como “HTML semântico”.

Um bom HTML semântico, permite que leitores de tela, através das *tags*, auxiliem de forma eficiente os usuários que possuem deficiências visuais ou físicas, o

que será de suma importância, tendo em vista que a plataforma Fooco tem como objetivo ser utilizada pelo maior público possível.

De acordo com Amauri Corrêa (2022), esses são os motivos do porquê devemos nos preocupar com a acessibilidade:

Há vários motivos para deixar o HTML acessível, primeiramente, e mais importante, serve para que todos consigam entender e navegar por nossa página. Além do que foi dito, um dos critérios para ranquear páginas é a sua acessibilidade. Eles reconhecem a adaptação do HTML através de atributos que adicionamos juntamente das *tags*, ou, até mesmo, de um HTML semântico. Vale ressaltar, que, existe um decreto federal de 2004 nº 5.296 que obriga sites no Brasil a serem acessíveis. Esse decreto foi sancionado no dia 6 de julho de 2015 a Lei Brasileira de Inclusão da Pessoa com Deficiência. (CORRÊA, 2022)

Corrêa também explica que pensar em acessibilidade no início de um projeto, pode impactar em apenas 5% no custo previsto, por outro lado, implementar esse tipo de funcionalidade mais adiante ou após o projeto ter sido finalizado, esse custo pode chegar em até 40% mais.

2.1.3. *Styled Components*

O *Styled Components* é uma biblioteca que junta *Javascript* e CSS em um só lugar. Esta biblioteca utiliza o SCSS (ou SASS) para realizar a marcação de estilos, sendo este nada mais do que CSS com superpoderes, como os criadores mesmos dizem, adicionado diversas formas de dinamicidade ao código CSS. Através do *Styled Components* será alinhado *layouts*, definido estilos de texto, temas de cores e elementos do aplicativo, tudo isso de forma totalmente compatível e interativa com o Typescript.

Com *Styled Components*, também pode-se estilizar componentes em arquivos separados do código Typescript React (ou TSX), garantindo maior organização, além de deixar o aplicativo mais facilmente suscetível a mudanças posteriores. A biblioteca também facilita a reutilização de códigos de estilo em outros componentes, reduzindo assim, a necessidade de códigos repetidos durante o desenvolvimento.

2.1.4 *Typescript*

Visando a produtividade e a manutenibilidade do código, será utilizado o recente Typescript, sendo este muito popular em empresas de tecnologia, na qual os códigos desenvolvidos devem ser sempre fáceis de entender e de continuar seu desenvolvimento. O Typescript nada mais é do que uma versão “turbinada” do

Javascript, ou como os programadores que utilizam dizem, um *superset* da linguagem. É importante ressaltar que o Typescript não é uma linguagem de programação, mas sim uma versão melhorada do Javascript, como foi dito.

No mundo do desenvolvimento Web, se torna quase imprescindível utilizar o Typescript em qualquer projeto, desde o Front-end com React, até no Back-end com o Express e Node, por exemplo. O Typescript, em sua essência, visa acabar com os famosos erros de *runtime*, ou seja, os erros que ocorrem quando seu código está em processamento, ao invés disso, ele é capaz de apontar erros com precisão durante o desenvolvimento do projeto, ressaltando-os na interface do editor de código apontando exatamente o local do problema, o que acaba aumentando a produtividade de qualquer desenvolvedor.

No Front-end o Typescript atuará juntamente com o React, sendo esses dois responsáveis por gerenciar toda a estrutura de dados na tela do usuário. Sua maior contribuição ao código será a requisição de dados do Back-end, com ele será possível “tipar” os dados que virão do servidor, facilitando o entendimento do desenvolvedor e sua produtividade.

2.1.5 React

O React é uma biblioteca do Javascript que cria diversas ferramentas para a criação de interfaces interativas pelo usuário, além de funcionalidades que aumentam a produtividade e facilitam a vida do desenvolvedor, sendo ela uma biblioteca flexível e adaptável a qualquer tipo de projeto. Com o React, é possível “componentizar” regiões específicas de um código e assim, reutilizar esse componente estilizado pelo Styled Components, em qualquer lugar do site.

A plataforma Fooco contará com diversos *inputs* espalhados por várias regiões da interface, como a sessão de criação do usuário, editar o perfil ou fazer um post. Tendo isso em vista, o React será o responsável por controlar esses *inputs* garantindo que os dados deles cheguem de maneira eficiente à requisição ao servidor, além de realizar a verificação e a validação dos dados.

2.1.6 React Router

Ao utilizar uma página da web, é comum haver links de navegação interna ao decorrer da página, que é responsável por levar de uma ponta a outra de um site. O React Router é uma biblioteca feita com esse mesmo propósito, com a diferença que

ele funciona com a implementação *single page application* (Aplicação de Página Única), que consiste em uma única página web responsável por exibir e gerenciar todas as rotas e conteúdo do site de forma dinâmica e interativa. Com o React Router é possível navegar através do site de forma simples e sem telas de carregamentos, visto que não é necessário recarregar a página inteira para renderizar o novo conteúdo.

2.1.7 Radix Primitives

O Radix é uma biblioteca de interfaces *open-source* e foi desenvolvida com o propósito de criar componentes de design focado em acessibilidade, alta qualidade e customização. A biblioteca possui diversos componentes primitivos de *layout*, ou seja, componentes “crus” nos quais é necessário apenas a sua estilização através do *Styled Componentes* ou qualquer outra ferramenta de estilos, como botões, listas *dropdown*, telas de diálogo, *inputs* como *select* e *checkbox*, *popups* e muitos outros componentes. Outra grande característica do Radix é o seu suporte à acessibilidade, sendo este um recurso fundamental para a Web atualmente, permitindo que PCD's tenham uma melhor experiência de usuário.

2.2. Back-End

2.2.1. Typescript

No lado do servidor, o Typescript também será um dos grandes agentes durante o desenvolvimento, ao passo que o código será produzido com base nele. Assim como no Front-end, o superset de Javascript ajudará na certificação das “tipagens” do projeto, garantindo um código sem erros e com maior legibilidade.

2.2.2. Express

O Express é uma ferramenta imprescindível para qualquer pessoa que deseja criar um servidor usando o Javascript. Esse framework atua juntamente com o Node, que nada mais é do que uma versão do Javascript criada com o único propósito de rodar o código da linguagem através do servidor, o que não era impossível antes de seu desenvolvimento. Tendo isso posto, o Express pode expandir grandes horizontes para um desenvolvedor Back-end, já que ele é capaz de introduzir diversos métodos para aumentar a produtividade no desenvolvimento de uma aplicação voltada para um

servidor. Seguindo o padrão REST, o Express pode criar rotas de *endpoint* usadas para manipular os dados de um servidor. Pode-se criar, obter, editar, deletar entre outras funções que o Express pode prover.

2.2.3. Postgres

O Postgres auxilia no gerenciamento dos dados de forma organizada e eficaz. O usuário pode fazer consultas de forma mais rápida, sem acesso direto com o banco de dados, fazendo que apenas o servidor faça essa consulta direta ao conteúdo. De modo geral o Postgres é um organizador de todas as informações, funcionando também como uma plataforma de consultas e configurações.

De acordo com o THE POSTGRES SQL GLOBAL DEVELOPMENT GROUP (2019) “O PostgreSQL é um sistema de banco de dados objeto-relacional de código aberto”, simplificando na relação de gerenciamento do banco de dados, proporcionando vantagens principalmente nas extensões.

2.2.4 TypeORM

O TypeORM é mais uma das ferramentas ORM's disponíveis para gerenciar *queries* SQL de maneira simples e com maior legibilidade, além de também ser o agente de conexão entre o Back-end e o banco de dados. No site oficial, os criadores dizem que diferente de todos os outros Javascript ORM's, o TypeORM é capaz ser escrito com mais qualidade de leitura, de forma mais escalável, manutenível e da maneira mais produtível possível. (TYPEORM, 2023, tradução nossa)

Essa ferramenta será fundamental para a criação das tabelas SQL, de forma que será possível o gerenciamento dos tipos de dados, valores padrões, colunas de ID geradas automaticamente utilizando UUID, colunas com datas de criação ou de atualização, tudo isso utilizando a notação do Typescript.

3. EMPRESA

A *Animalia* foi criada com o intuito de gerar cooperação e desenvolver empatia ao passo da troca de conhecimento entre indivíduos inseridos no mesmo ambiente ou comunidade. O foco da empresa é estar presente em ambientes no qual a cooperação seja imprescindível, sendo uma ferramenta de auxílio simples e intuitiva, acessível para qualquer tipo de público.

3.1 Missão

A empresa *Animalia* foi fundada com o propósito de desenvolver ferramentas que sirvam como auxílio para comunidades que buscam maneiras alternativas de melhorar a cooperação, de forma divertida e interativa, sem quaisquer barreiras entre as pessoas.

Focado em ensino médio e cursos técnicos, *Fooco* é o primogênito da *Animalia*. Com ele, objetifica-se melhorar o senso de comunidade, cidadania e empatia dos estudantes, além de aplicar os valores da *Animalia* neste projeto.

3.2 Visão

Pretende-se ao curto prazo, lançar o projeto primogênito da empresa, *Fooco*, do qual se espera a implementação na rede de ensino de estudo, a Etec Adolpho Berezin. Realizando atualizações de melhoria do aplicativo com base nos *feedbacks* recebidos, buscando moldar e atingir uma versão satisfatória, coletando informações do usuário afim de entender o público-alvo estudado.

Ao longo prazo, a *Animalia* almeja expandir sua rede de aplicativos, criando outras ferramentas que disseminem a sua missão de promover a cooperação e a troca de conhecimentos. Visando ser presente no cotidiano geral das pessoas.

3.3 Valores

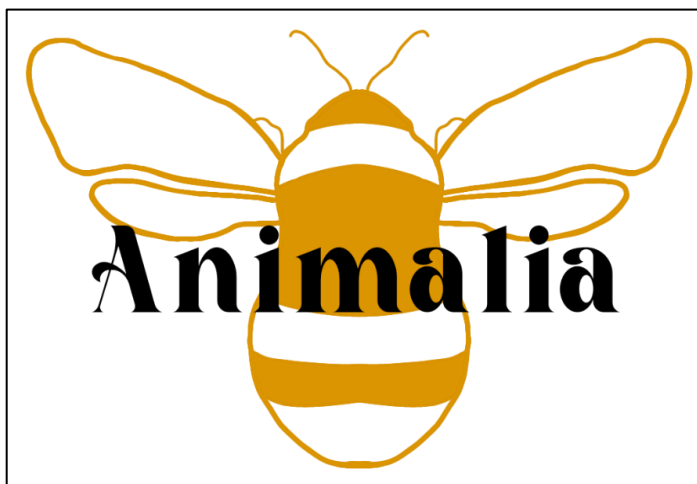
A empresa busca, principalmente, a coletividade dentro do ambiente interno e externo, sendo o ambiente interno composto pelos colaboradores e funcionários da empresa e o ambiente externo os usuários que utilizam a plataforma.

Tendo em vista que o principal produto da *Animalia* é o aplicativo *Fooco* que, por consequência, é centrado exclusivamente em torno de comunidades, é essencial

que exista o senso de ética, respeito e honestidade presente dentro da empresa e que isso seja um dos pilares a move.

3.4 Logo

Figura 1 - Logo da Equipe



Fonte: Criado pela equipe

3.4.1 Logo do Projeto

Figura 2 - Logo do Projeto



Fonte: Criado pela equipe

3.5 Slogan

“Juntos aprendemos mais”

4. ANÁLISE

4.1. Descrição das Funcionalidades

4.1.1 Publicação

A principal funcionalidade da plataforma Foco é seu sistema de publicação, na qual o usuário pode compartilhar seu conhecimento com outros usuários do mesmo nicho, gerando interações e riqueza de conhecimentos para a comunidade.

Para a criação da publicação, utiliza-se a linguagem de marcação *Markdown*, linguagem essa que é extremamente popular entre comunidades de fórum mais consolidadas na internet, como Stack OverFlow e TabNews. O *Markdown* possui uma notação simplificada para usuários que não são habituados com ela. Mas ainda assim, o usuário pode optar por não a utilizar durante a criação da publicação, o que pode comprometer a visibilidade e interesse dos outros usuários em seu conteúdo.

A janela de criação da publicação, é acompanhada por um sinal de interrogação no topo, que ao clicar, o usuário verá um mini manual de ajuda e dicas com a sintaxe do Markdown.

4.1.2 Salvar para mais tarde

Caso o usuário se interesse pela publicação, mas não tem tempo de vê-la agora, ele pode optar por salvar a publicação para mais tarde, removendo as chances de perder a publicação de vista. Todos os conteúdos salvos são exibidos na guia “Salvos” do perfil do usuário.

4.1.3 Ranking

O sistema de gamificação da plataforma Foco, traz consigo o ranking de usuários com mais pontos dentro da comunidade, indo de forma decrescente até o número de 10 usuários. Sendo assim, os usuários que mais geraram engajamento e interações dentro da plataforma são expostos na página inicial do site, atraindo o interesse de outras pessoas para fazer parte do ranking.

Os primeiros 3 usuários são destacados de cima para baixo com a cor dourada, prata e cobre, respectivamente. Caso você seja o proprietário da conta, seu usuário ficará destacado com a cor verde, como forma de identificação.

4.1.4 Comentários

Para incentivar o engajamento e a troca de conhecimento, foi desenvolvido no site o sistema de comentários, no qual outras pessoas podem responder o dono da publicação de forma a acrescentar mais conhecimento para a comunidade. Os comentários também podem ser feitos utilizando toda a marcação *Markdown*, incluído a estilização de citações para referenciar trechos da publicação original utilizando o sinal de maior (>) e respondê-las separadamente.

4.2. MER – Modelo do Banco de Dados

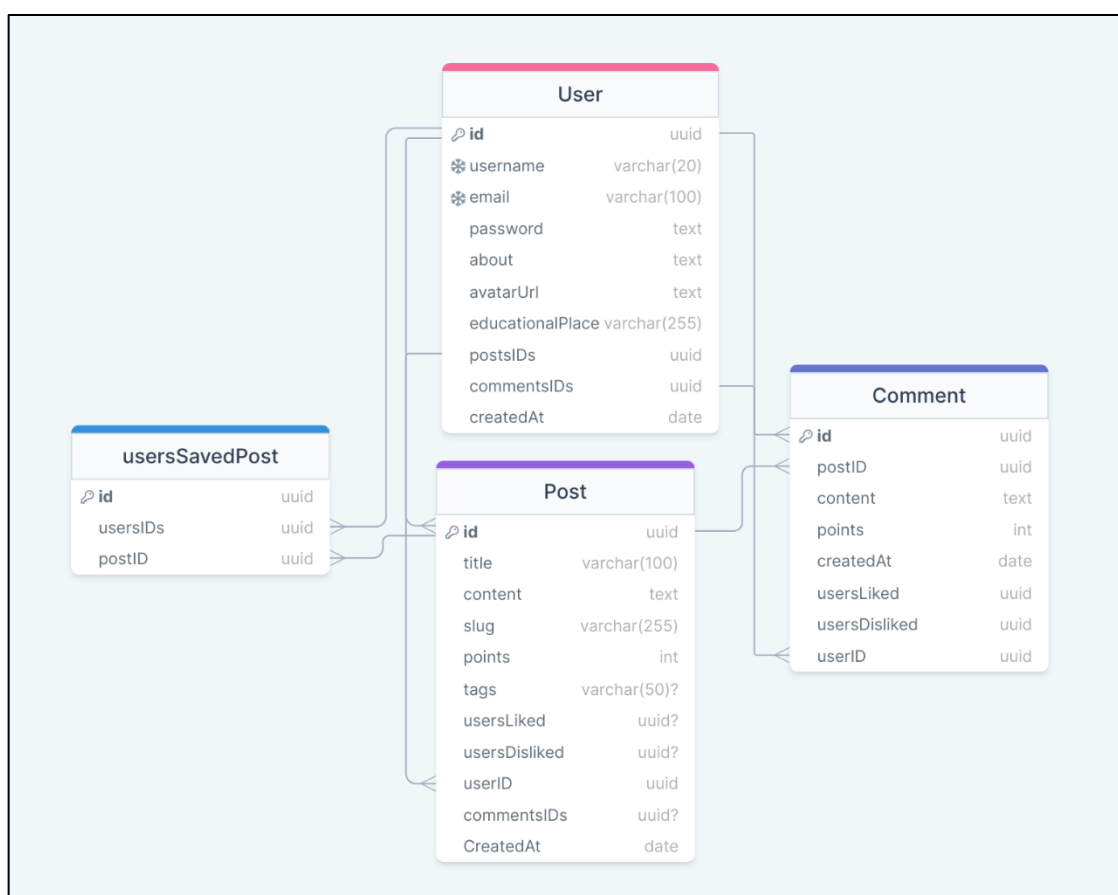


Figura 3 - Modelagem do Banco de Dados

4.3. Create do Banco de Dados

Para a criação, manipulação e alteração do banco de dados, foi utilizado o serviço *TypeORM*, que funciona através do mapeamento de objeto-relacional (ORM) utilizando a sintaxe do Javascript. Com ele, pode-se criar, consultar, atualizar e excluir registros do banco de dados, além de também oferecer recursos para o relacionamento entre tabelas ou entidades.

Para a criação do banco de dados, o *TypeORM* converte toda notação escrita em Javascript, através das entidades, para a notação do banco de dados selecionado utilizando o sistema de “migrações” (*migrations*).

4.3.1 Migration do banco de dados (PostgresSQL)

```

public async up(queryRunner: QueryRunner): Promise<void> {
  await queryRunner.query(`CREATE TABLE "comments" ("id" uuid NOT NULL DEFAULT uuid_generate_v4(), "post_id" uuid NOT NULL, "content" text NOT NULL,
"points" integer NOT NULL DEFAULT '0', "created_at" TIMESTAMP NOT NULL DEFAULT ('now'::text)::timestamp without time zone, "users_liked" character
varying array NOT NULL DEFAULT '{}', "users_disliked" character varying array NOT NULL DEFAULT '{}', "postIdId" uuid, "userId" uuid, CONSTRAINT
"PK_8bf68bc968f2b69e818bdb90dcb" PRIMARY KEY ("id"))`);
  await queryRunner.query(`CREATE TABLE "posts" ("id" uuid NOT NULL DEFAULT uuid_generate_v4(), "title" character varying NOT NULL, "content" text NOT
NULL, "points" integer NOT NULL DEFAULT '0', "created_at" TIMESTAMP NOT NULL DEFAULT ('now'::text)::timestamp without time zone, "tags" character
varying array NOT NULL DEFAULT '{}', "users_liked" character varying array NOT NULL DEFAULT '{}', "users_disliked" character varying array NOT NULL
DEFAULT '{}', "slug" character varying NOT NULL DEFAULT '', "user_id" uuid, CONSTRAINT "PK_2829ac61efff68fcec60d7274b9e" PRIMARY KEY ("id"))`);
  await queryRunner.query(`CREATE TABLE "users" ("id" uuid NOT NULL DEFAULT uuid_generate_v4(), "username" character varying NOT NULL, "slug"
character varying NOT NULL DEFAULT '', "email" character varying NOT NULL, "password" text NOT NULL, "educational_place" character varying NOT NULL
DEFAULT '', "educational_place_url" character varying NOT NULL DEFAULT '', "about" character varying(500) NOT NULL DEFAULT '', "avatar_url"
character varying NOT NULL DEFAULT '', "banner_url" character varying NOT NULL DEFAULT '', "created_at" TIMESTAMP NOT NULL DEFAULT now(), CONSTRAINT
"UQ_fe0bb3f6520ee0469504521e710" UNIQUE ("username"), CONSTRAINT "UQ_97672ac88f789774dd47f7c8be3" UNIQUE ("email"), CONSTRAINT
"PK_a3fffb1c0c8416b9fc6f907b7433" PRIMARY KEY ("id"))`);
  await queryRunner.query(`CREATE TABLE "users_saved_posts_posts" ("usersId" uuid NOT NULL, "postsId" uuid NOT NULL, CONSTRAINT
"PK_ac8f0bad46eb4f52ec93b6ec307" PRIMARY KEY ("usersId", "postsId"))`);
  await queryRunner.query(`CREATE INDEX "IDX_bd5ab113a3b6d29246f1a04fb4" ON "users_saved_posts_posts" ("usersId")`);
  await queryRunner.query(`CREATE INDEX "IDX_f9df0e1ae84cd0d5227ca12dd3" ON "users_saved_posts_posts" ("postsId")`);
  await queryRunner.query(`ALTER TABLE "comments" ADD CONSTRAINT "FK_30e6a2e07ef940670d493b759ed" FOREIGN KEY ("postIdId") REFERENCES "posts"("id") ON
DELETE CASCADE ON UPDATE NO ACTION`);
  await queryRunner.query(`ALTER TABLE "comments" ADD CONSTRAINT "FK_7e8d7c49f218ebb14314fdb3749" FOREIGN KEY ("userId") REFERENCES "users"("id") ON
DELETE CASCADE ON UPDATE NO ACTION`);
  await queryRunner.query(`ALTER TABLE "posts" ADD CONSTRAINT "FK_c4f9a7bd77b489e711277ee5986" FOREIGN KEY ("user_id") REFERENCES "users"("id") ON
DELETE NO ACTION ON UPDATE NO ACTION`);
  await queryRunner.query(`ALTER TABLE "users_saved_posts_posts" ADD CONSTRAINT "FK_bd5ab113a3b6d29246f1a04fb42" FOREIGN KEY ("usersId") REFERENCES
"users"("id") ON DELETE CASCADE ON UPDATE CASCADE`);
  await queryRunner.query(`ALTER TABLE "users_saved_posts_posts" ADD CONSTRAINT "FK_f9df0e1ae84cd0d5227ca12dd34" FOREIGN KEY ("postsId") REFERENCES
"posts"("id") ON DELETE CASCADE ON UPDATE NO ACTION`);
}

```

4.3.1 Entidade do usuário

```
1 @Entity("users")
2 class User {
3     @PrimaryGeneratedColumn("uuid")
4     id: string
5
6     @Column({ type: "varchar", unique: true })
7     username: string
8
9     @Column({ type: "varchar", default: "" })
10    slug: string
11
12    @Column({
13        type: "varchar",
14        unique: true,
15        transformer: {
16            to(value: string) {
17                return value.toLowerCase()
18            },
19            from(value: string) {
20                return value.toLowerCase()
21            }
22        }
23    })
24    email: string
25
26    @Column({ type: "text", select: false })
27    password: string
28
29    @OneToMany(() => Post, (post) => post.user, { eager: true })
30    @JoinTable({
31        name: "post_user",
32        joinColumn: {
33            name: "user_id",
34            referencedColumnName: "id"
35        },
36        inverseJoinColumn: {
37            name: "post_id",
38            referencedColumnName: "id"
39        }
40    })
41    posts: Post[]
42
43    @OneToMany(() => Comment, (comment) => comment.user, { cascade: ["remove"] })
44    @JoinTable({
45        name: "user_comment",
46        joinColumn: {
47            name: "user_id",
48            referencedColumnName: "id"
49        },
50        inverseJoinColumn: {
51            name: "comment_id",
52            referencedColumnName: "id"
53        }
54    })
55    reply: Comment[]
```

```
1 @Column({ type: "varchar", default: "" })
2   educational_place: string
3
4 @Column({ type: "varchar", default: "" })
5   educational_place_url: string
6
7 @Column({ type: "varchar", length: 500, default: "" })
8   about: string
9
10 @Column({ type: "varchar", default: "" })
11   avatar_url: string
12
13 @Column({ type: "varchar", default: "" })
14   banner_url: string
15
16 @ManyToMany(() => Post, (post) => post.usersSaved, {
17   onDelete: "CASCADE",
18   orphanedRowAction: "delete",
19   cascade: true
20 })
21 @JoinTable()
22   savedPosts: Post[]
23
24 @CreateDateColumn({
25   transformer: {
26     from: (date: Date) => {
27       return date.toLocaleString("pt-BR", {
28         localeMatcher: "best fit"
29       })
30     },
31     to: (date: Date) => {
32       return new Date()
33     }
34   }
35 })
36   created_at: Date
37 }
```

4.3.3 Entidade da publicação

```
1 @Entity("posts")
2 class Post {
3     @PrimaryGeneratedColumn("uuid")
4     id: string
5
6     @Column({ type: "varchar" })
7     title: string
8
9     @Column({ type: "text" })
10    content: string
11
12    @Column({ type: "int", default: 0 })
13    points: number
14
15    @Column({
16        default: () => "LOCALTIMESTAMP"
17    })
18    created_at: Date
19
20    @Column({ default: [], array: true, type: "varchar" })
21    tags: string[]
22
23    @Column({ default: [], array: true, type: "varchar" })
24    users_liked: string[]
25
26    @Column({ default: [], array: true, type: "varchar" })
27    users_disliked: string[]
28
29    @Column({ type: "varchar", default: "" })
30    slug: string
31
32    @OneToMany(() => Comment, (comment) => comment.post_id, {
33        nullable: true,
34        eager: true,
35        cascade: ["remove"]
36    })
37    @JoinTable({
38        name: "reply_post",
39        joinColumn: {
40            name: "reply_id",
41            referencedColumnName: "id"
42        },
43        inverseJoinColumn: {
44            name: "post_id",
45            referencedColumnName: "id"
46        }
47    })
48    reply: Comment[]
49
50    @ManyToMany(() => User, (user) => user.savedPosts, {
51        onDelete: "CASCADE",
52        orphanedRowAction: "delete"
53    })
54    usersSaved: User[]
55
56    @ManyToOne(() => User, (user) => user.posts)
57    @JoinColumn({ name: "user_id" })
58    user: User
59 }
```

4.3.4 Entidade do comentário

```
1  @Entity("comments")
2  class Comment {
3      @PrimaryGeneratedColumn("uuid")
4      id: string
5
6      @ManyToOne(() => Post, (post) => post.id, {
7          onDelete: "CASCADE",
8          orphanedRowAction: "delete"
9      })
10     @Column({ type: "uuid" })
11     post_id: string
12
13     @Column({ type: "text" })
14     content: string
15
16     @Column({ type: "int", default: 0 })
17     points: number
18
19     @Column({
20         default: () => "LOCALTIMESTAMP"
21     })
22     created_at: Date
23
24     @Column({ default: [], array: true, type: "varchar" })
25     users_liked: string[]
26
27     @Column({ default: [], array: true, type: "varchar" })
28     users_disliked: string[]
29
30     @ManyToOne(() => User, (user) => user.reply, {
31         onDelete: "CASCADE",
32         orphanedRowAction: "delete"
33     })
34     @JoinTable({
35         name: "comment_user",
36         joinColumn: {
37             name: "comment_id",
38             referencedColumnName: "id"
39         },
40         inverseJoinColumn: {
41             name: "user_id",
42             referencedColumnName: "id"
43         }
44     })
45     user: User
46 }
```


4.4.3 Comments



4 rows returned

	id uuid	post_id uuid	content text	points integer	created_at timestamp without time zone	users_liked character varying[]	users_disliked character varying[]	postIdId uuid	userId uuid
1	e4025935-2519-48e8-bff6-72066521ea95	fa697c4f-da02-4280-98ac-48338301db65	## subtítulo legal!!	0	Thu Jun 01 2023 17:51:11 GMT-0300 (Brasilia Standard Time)			fa697c4f-da02-4280-98ac-48338301db65	782eb935-d8c0-40e3-b4d4-00bcf9ade2dd
2	2d959448-4c86-4f99-abd1-7d8337fe7ed6	fd95b37e-a258-48c3-939f-55275024e76f	muito bom !!!!	0	Mon Jun 05 2023 22:34:03 GMT-0300 (Brasilia Standard Time)			fd95b37e-a258-48c3-939f-55275024e76f	34627679-44cf-4e30-8f71-c60bb2f67741
3	e782356e-7feb-4ad0-bd55-bdf7cd51d672	4a3052e6-9b81-449e-a573-329bc217ae88	## Ótimo! **Que guia incrível!** 🍌🍌 Aprender a fazer uma requisição HTTP em Rust nunca foi tão fácil. Este passo a p...	0	Mon Jun 12 2023 02:29:48 GMT-0300 (Brasilia Standard Time)			4a3052e6-9b81-449e-a573-329bc217ae88	34627679-44cf-4e30-8f71-c60bb2f67741
4	24ddc199-d278-42c4-8d34-b2bdd2a74f1c	5a42ce46-dfc4-4f95-b18e-9a08744a1203	## Ótimo **A computação quântica é uma fascinante fronteira da ciência e da tecnologia!** 🍌🍌 Este artigo fornece um...	0	Mon Jun 12 2023 02:33:13 GMT-0300 (Brasilia Standard Time)			5a42ce46-dfc4-4f95-b18e-9a08744a1203	34627679-44cf-4e30-8f71-c60bb2f67741

4.5. Wireframe das Telas

4.5.1 Tela inicial



Figura 4 – Wireframe da Tela Inicial

4.5.2 Dashboard

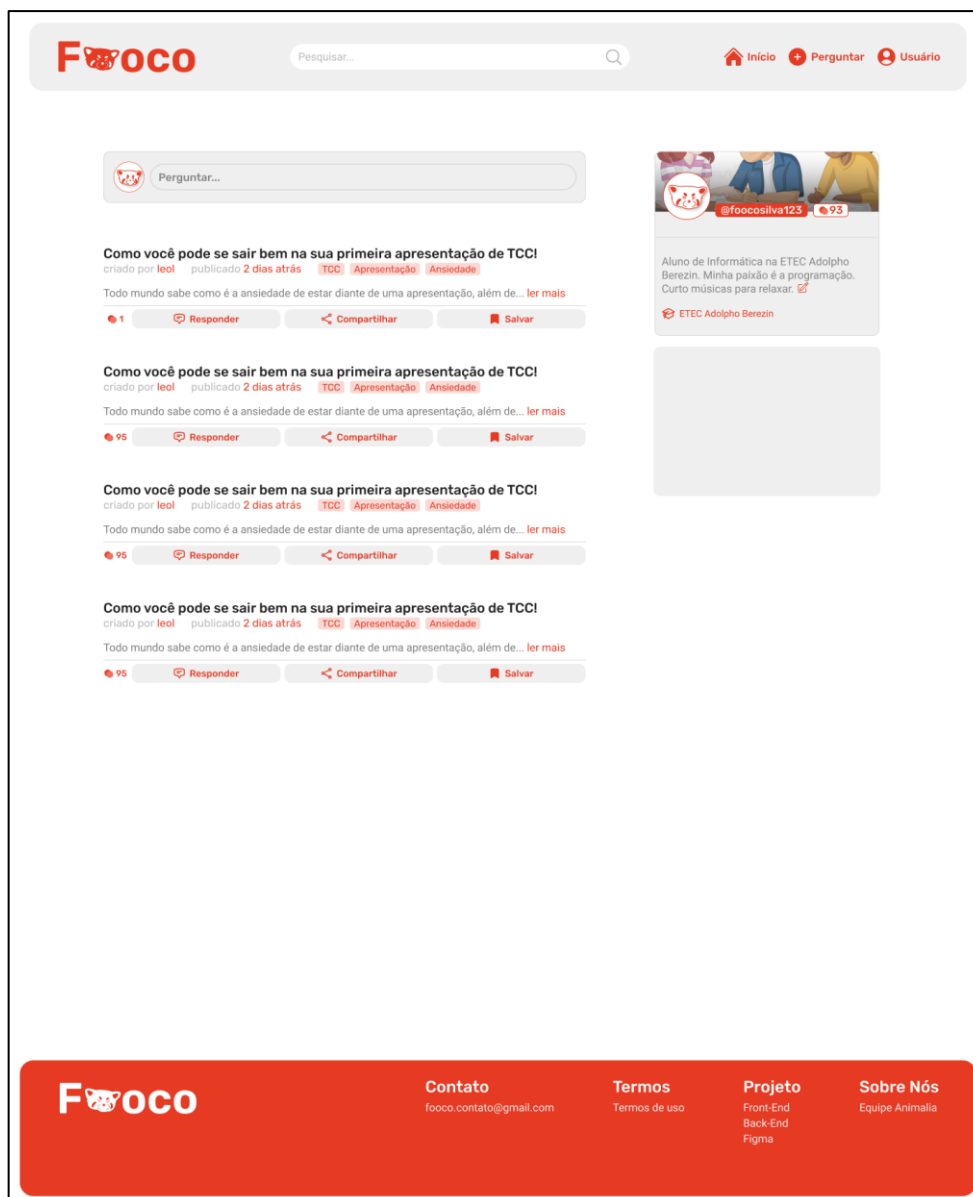


Figura 5 - Wireframe do Dashboard

4.5.3 Não encontrado (404)



Figura 6 - Wireframe da Tela Não encontrado

4.6 Prints das Telas

4.6.1 Tela inicial



Figura 7 - Tela Inicial do Fooco

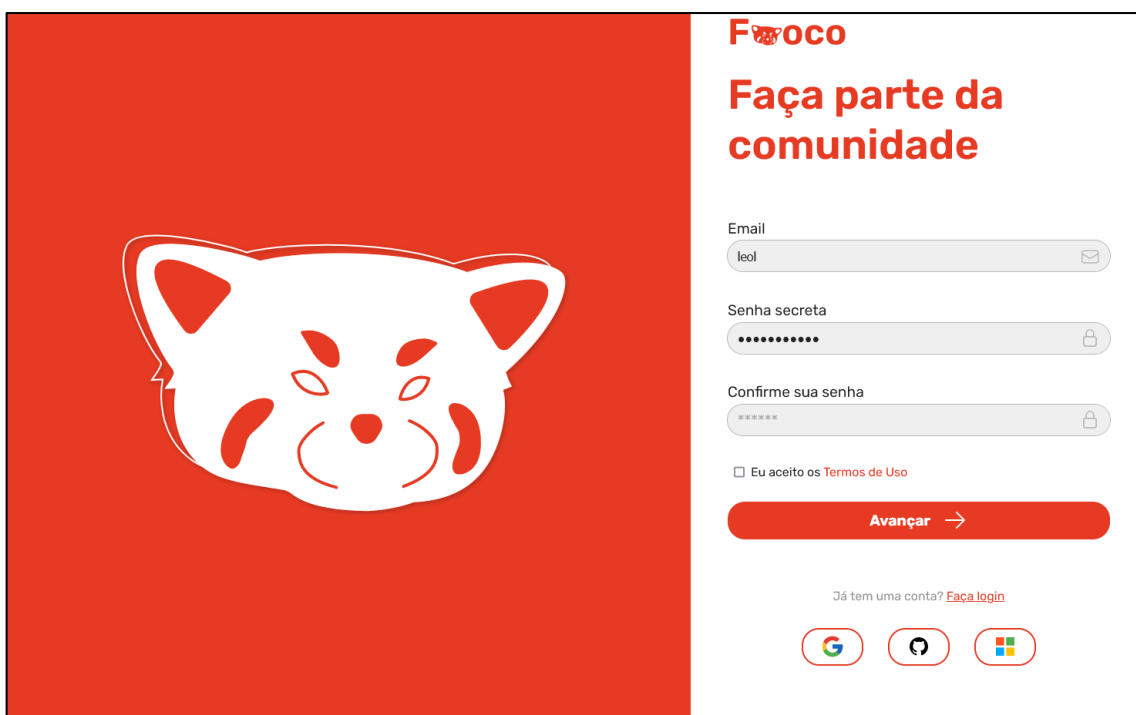
4.6.2 Entrar



The login screen features a large red background on the left with a white outline of a cat's face. On the right, the Fwoco logo is at the top. Below it, the text "Bem-vindo de volta!" is displayed. The form includes fields for "Email ou Apelido" (containing "leol") and "Senha secreta" (masked with dots). A red "Entrar" button with a right arrow is positioned below the fields. At the bottom, there is a link "Ainda não tem uma conta? [Crie agora](#)" and three social login icons for Google, Facebook, and Microsoft.

Figura 8 - Tela de Login

4.6.3 Criar conta



The registration screen features a large red background on the left with a white outline of a cat's face. On the right, the Fwoco logo is at the top. Below it, the text "Faça parte da comunidade" is displayed. The form includes fields for "Email" (containing "leol"), "Senha secreta" (masked with dots), and "Confirme sua senha" (containing "*****"). A checkbox "Eu aceito os [Termos de Uso](#)" is present. A red "Avançar" button with a right arrow is positioned below the fields. At the bottom, there is a link "Já tem uma conta? [Faça login](#)" and three social login icons for Google, Facebook, and Microsoft.

Figura 9 - Tela para criar Conta

4.6.4 Dashboard

The screenshot displays the Fooco dashboard interface. At the top, there is a navigation bar with the Fooco logo, a search bar, and links for 'Início', 'Publicar', and 'Perfil'. Below the navigation bar, the main content area is divided into two columns.

Left Column (Post Creation and Feed):

- A post creation box at the top allows publishing as 'leol...'.
- The first post is titled 'EXEMPLO MARKDOWN' by @leonardo_g141, published 6 days ago with 0 responses. It includes tags 'Markdown', 'Fooco', and 'Exemplo'. Below the post are buttons for '1' (comment count), 'Responder', 'Enviar', and 'Salvar'.
- The second post is titled 'Os Melhores Aplicativos de Banco de Dados da atualidade' by @Drih_drih, published 11 days ago with 1 response. It includes tags 'BD', 'Banco', 'Dados', and 'Aplicativo'. Below the post are buttons for '4' (comment count), 'Responder', 'Enviar', and 'Remover'.
- The third post is titled 'Diferença entre Banco de Dados Informatizado e Formatizado' by @Drih_drih, published 11 days ago with 0 responses. It includes tags 'BD', 'Informatizado', 'Formatizado', and 'Diferenças'.

Right Column (User Profile and Ranking):

- The top section shows the profile of user '@leol', identified as 'Desenvolvedor do Fooco, 1° usuário registrado' from 'ETEC Adolpho Berezin'.
- The 'Ranking' section, titled 'Acompanhe o ranking global de pontos', lists the top four users:

Rank	Letter	Username	Points
1°	A	julya	12
2°	A	Drih_drih	8
3°		leol	2
4°		leonardo_g141	1

Figura 10 - Tela do Dashboard

4.6.6 Publicação e comentário

The screenshot displays the Foco website interface. At the top, there is a navigation bar with the Foco logo, a search bar containing the word 'administração', and links for 'Início', 'Publicar', and 'Perfil'. The main content area features a post by user @julya, titled 'O que é COMPUTAÇÃO QUÂNTICA?'. The post includes an introduction to quantum computing, explaining that it uses physical principles to process information differently than classical computers. It discusses superposition, which allows qubits to exist in multiple states simultaneously, and entanglement, where qubits are directly related to each other. The post concludes by noting that quantum computing is still in its early stages and faces technical challenges, but has the potential to revolutionize fields like cryptography and system simulation.

Below the post, there is a 'Responder' button and two tags: 'Computação' and 'Quântica'. A comment from user @leonardo_g141 follows, praising the article as a 'fascinating frontier of science and technology' and providing a detailed introduction to the complex topic. The comment mentions concepts like superposition, entanglement, and qubits, and expresses hope for the future of quantum computing. It includes social media sharing icons and a '#ComputaçãoQuântica #InovaçãoTecnológica' hashtag.

The footer of the page is a red bar containing the Foco logo, contact information (fooco.contato@hotmail.com), a help section (Ajuda - Termos de Uso), a projects section (Projetos - Front-End, Back-End, Figma), and an 'About Us' section (Sobre Nós - Equipe Animalia).

Figura 11 - Tela de Publicação e Comentário

4.6.10 Denunciar publicação

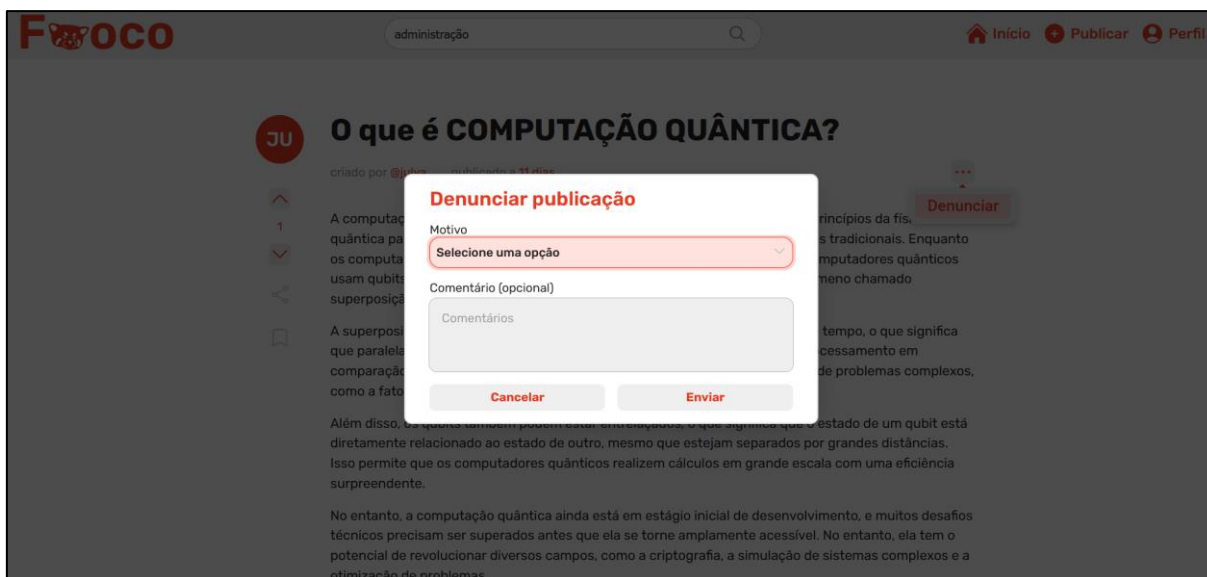


Figura 12 - Tela de Denunciar Publicação

4.6.9 Criar publicação

The image shows a form titled "Publicar" with the following fields and elements:

- Título da publicação:** A text input field with the placeholder "Descreva sua publicação".
- Rich Text Editor:** A large text area with a toolbar containing icons for bold (H), italic (I), quote ("), link (🔗), undo (↶), code ({}), list (☰), and indent (☰). On the right side of the toolbar are icons for image, video, link, unlink, and refresh.
- Tags:** A text input field with the placeholder "Palavras-chave da sua publicação".
- Buttons:** Two buttons at the bottom right, "Cancelar" and "Publicar", both in red.

Figura 13 - Tela para criar Publicação

4.6.12 Editar publicação

Fooco administração

Início Publicar Perfil

Título da publicação

Fazendo uma Requisição HTTP em Rust: Um Guia Passo a Passo

H B I

Fazer requisições HTTP é uma tarefa comum ao desenvolver aplicativos que precisam interagir com APIs e serviços da web. Neste post, vamos explorar como fazer uma requisição HTTP em Rust usando a biblioteca `request`. O Rust é uma linguagem de programação moderna e segura, conhecida por sua performance e pela ausência de erros de memória e segurança. Vamos ver como utilizar essa poderosa biblioteca para realizar requisições HTTP de forma fácil e eficiente.

```

```

Passo 1: Configurando o ambiente

Tags

Rust Http Tokio Requisições Request Backend Programação

Cancelar Editar

Figura 14 - Tela para criar Publicação

4.6.7 Perfil do usuário

@leol

Desenvolvedor do Fooco. 1º usuário registrado.

ETEC Adolpho Berezin

Publicações Salvos

Fazendo uma Requisição HTTP em Rust: Um Guia Passo a Passo

por @leol publicado a 12 dias 0 respostas

Rust Http Tokio Requisições Request Backend Programação

Responder Enviar Remover

Figura 15 - Tela de Perfil do Usuário

4.6.11 Publicações salvas



Figura 16 - Tela de Publicações Salvas

4.6.8 Pesquisa

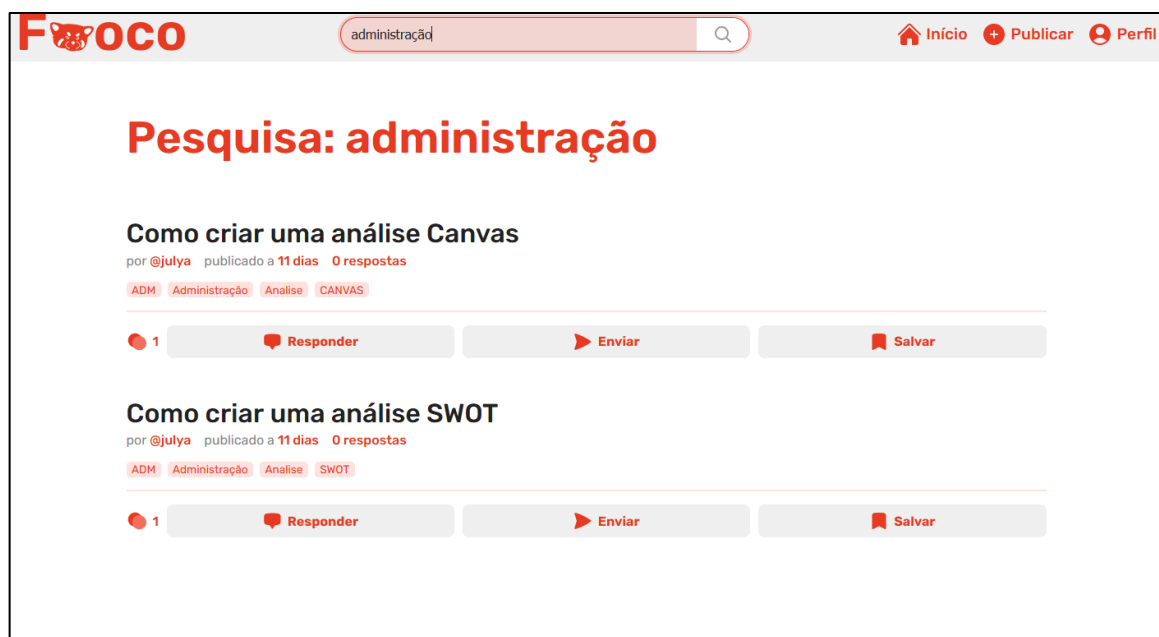


Figura 17 - Tela de Pesquisa

4.6.5 Configurações

The screenshot displays the user configuration interface. At the top left is the 'FOCO' logo. A search bar contains the text 'administração'. On the top right are navigation links: 'Início' (Home), 'Publicar' (Publish), and 'Perfil' (Profile). The left sidebar is titled 'Configurações' and contains two menu items: 'Conta' (Account) and 'Tema' (Theme). The main content area is titled 'Alterar informações da conta'. It features two input fields: 'Novo e-mail' with the value 'novoemail@exemplo.com' and 'Nova senha' with masked characters. To the right is a circular profile picture of a cat. Below the profile picture are two buttons: 'Cancelar' (Cancel) and 'Salvar' (Save).

Figura 18 - Tela de Configurações

4.7 Trecho do Código Fonte

4.7.1 Endpoints da API

```
1 // POSTS
2 router.get("/post", postController.index)
3 router.get("/post/:username/:post_slug", postController.show)
4 router.get("/post/:username", postController.show)
5 router.post("/post", schemaParse(schemaCreatePost), auth, postController.store)
6 router.put(
7   "/post/:post_id",
8   schemaParse(schemaUpdatePost),
9   auth,
10  postController.update
11 )
12 router.delete("/post/:post_id", auth, postController.delete)
13 router.post("/post/feedback/:post_id", auth, postController.feedback)
14
15 // POST CHILDREN
16
17 router.get("/reply", commentController.index)
18 router.get("/reply/:reply_id", commentController.show)
19 router.post("/reply/:post_id", auth, commentController.store)
20
21 // USERS
22 router.get("/user", userController.index)
23 router.get("/user/:username", schemaParse(schemaGetUser), userController.show)
24 router.post("/user", schemaParse(schemaCreateUser), userController.store)
25 router.post("/user/login", userController.login)
26 router.put(
27   "/user/:user_id",
28   schemaParse(schemaUpdateUser),
29   auth,
30   userController.update
31 )
32 router.delete(
33   "/user/:user_id",
34   auth,
35   schemaParse(schemaDeleteUser),
36   userController.delete
37 )
38
39 // TOKEN
40 router.post("/token", tokenController.validadeToken)
```

Figura 19 - Trecho do código fonte do Endpoint da API

4.7.2 Integração com a API para criar publicação

```
1  const { mutateAsync, isLoading } = useMutation({
2    mutationFn: async (data: Partial<UserPosts> & { user_id: string }) => {
3      const { options, url } = POST_POST(data)
4      return instance(url, options).then((res) => res.data)
5    },
6    mutationKey: "createPost",
7  })
8  const { setToastMessage } = UseToastStore()
9
10 const handleCreatePost = handleSubmit(async ({ tags, title }) => {
11   const arrTags = tags?.split(/[;,\\s]/g).filter((value) => Boolean(value))
12
13   if (!isLoggedIn || !userData) return nav("/entrar")
14
15   await mutateAsync(
16     {
17       title,
18       content: value,
19       user_id: userData.id,
20       tags: arrTags || [],
21     },
22     {
23       onSuccess(data) {
24         console.log(data)
25         nav(`/app/${userData.username}/${data.slug}`)
26         setToastMessage("Sucesso", "Publicação criada com sucesso!")
27       },
28       onError(error) {
29         if (isError(error) || error) {
30           window.scrollTo({ top: 0, behavior: "smooth" })
31           setToastMessage("Erro", "Não foi possível publicar")
32           return
33         }
34       },
35     }
36   )
37 })
```

Figura 20 - Trecho do código fonte com integração de API

5.7.3 Componente do Ranking

```

1  const RankingCard = () => {
2    const { isLoading, data } = useQuery<UserData[]>("points", {
3      queryFn: () => instance("/user").then((res) => res.data),
4    })
5    const { userData } = useUserStore()
6
7    if (isLoading)
8      return (
9        <Container>
10         <Title size="md">Ranking</Title>
11         <Paragraph size="lg">Acompanhe o ranking global de pontos</Paragraph>
12         <Separator />
13         <SkeletonLoad>
14           <Skeleton width="100%" height="10rem" borderRadius="0.4rem" />
15         </SkeletonLoad>
16       </Container>
17     )
18   return (
19     <Container>
20       <Header>
21         <Title size="md">Ranking</Title>
22         <Paragraph size="lg">Acompanhe o ranking global de pontos</Paragraph>
23         <Separator />
24       </Header>
25
26       <Rank>
27         {data
28           ?.filter((user) => user.posts.length > 0)
29           .sort(
30             (a, b) =>
31               b.posts.reduce((acc, { points }) => acc + points, 0) -
32               a.posts.reduce((acc, { points }) => acc + points, 0)
33           )
34           .slice(0, 10)
35           .map((user, index) => (
36             <Link
37               to={`/${user.username}`}
38               key={user.id}
39               className={userData?.id === user.id ? "owner" : ""}
40             >
41               <Item>
42                 <Position>{index + 1}</Position>
43                 <Avatar
44                   src={user.avatar_url}
45                   fallback="a"
46                   delayMs={0}
47                   size={2.5}
48                 />
49                 <Username>{user.username}</Username>
50                 <Points>
51                   <Point />
52                   {getUserPoints(user)}
53                 </Points>
54               </Item>
55             </Link>
56           )})
57       </Rank>
58     </Container>
59   )
60 }

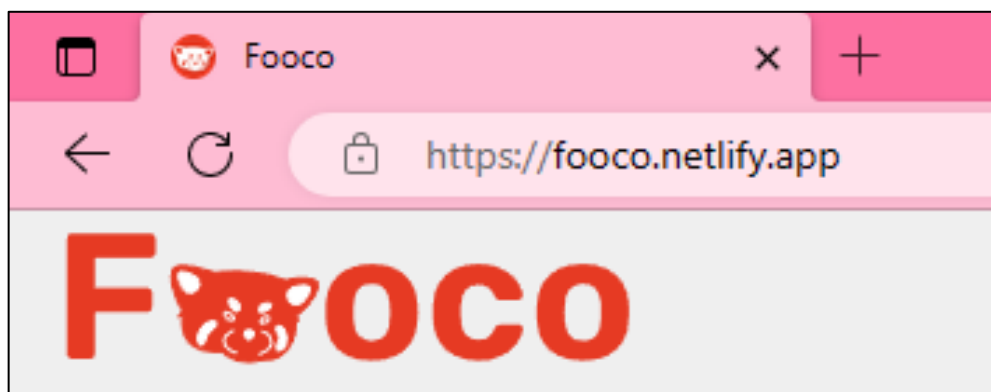
```

Figura 21 - Trecho do código fonte do Ranking

5. MANUAL DO USUÁRIO

5.1 Como acessar o Fooco?

A plataforma pode ser acessada em: <https://fooco.netlify.app>.

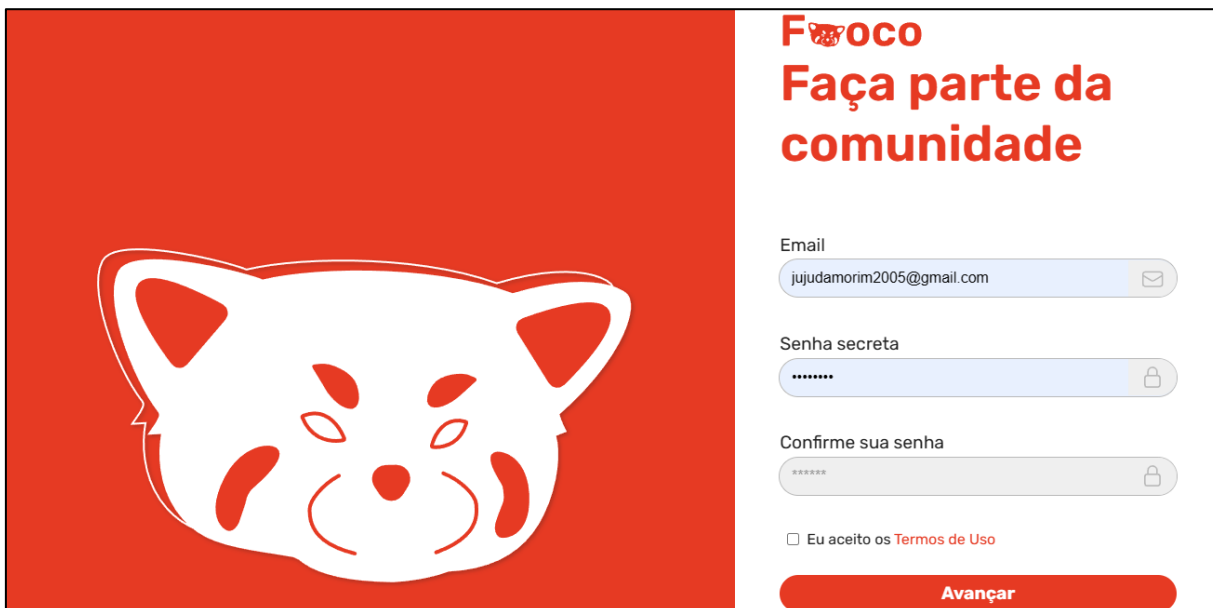


5.2 Como criar uma conta?

Após entrar no fórum, um convite será feito ao usuário, ao colocar o e-mail em “fazer parte”, sublinhado de amarelo, será redirecionado para a página de cadastro. Outra opção se localiza acima em conjunto com a logo do projeto.



Já na página de cadastro, é preciso criar uma senha, confirmá-la e ler os termos confirmando estar ciente deles.



The registration form features a red background on the left with a white cat face icon. On the right, the title 'Foco' is followed by 'Faça parte da comunidade'. The form includes fields for 'Email' (filled with 'jujudamorim2005@gmail.com'), 'Senha secreta' (masked with dots), and 'Confirme sua senha' (also masked). A checkbox for 'Eu aceito os Termos de Uso' is present, and a red 'Avançar' button is at the bottom.

Foco
Faça parte da comunidade

Email
jujudamorim2005@gmail.com

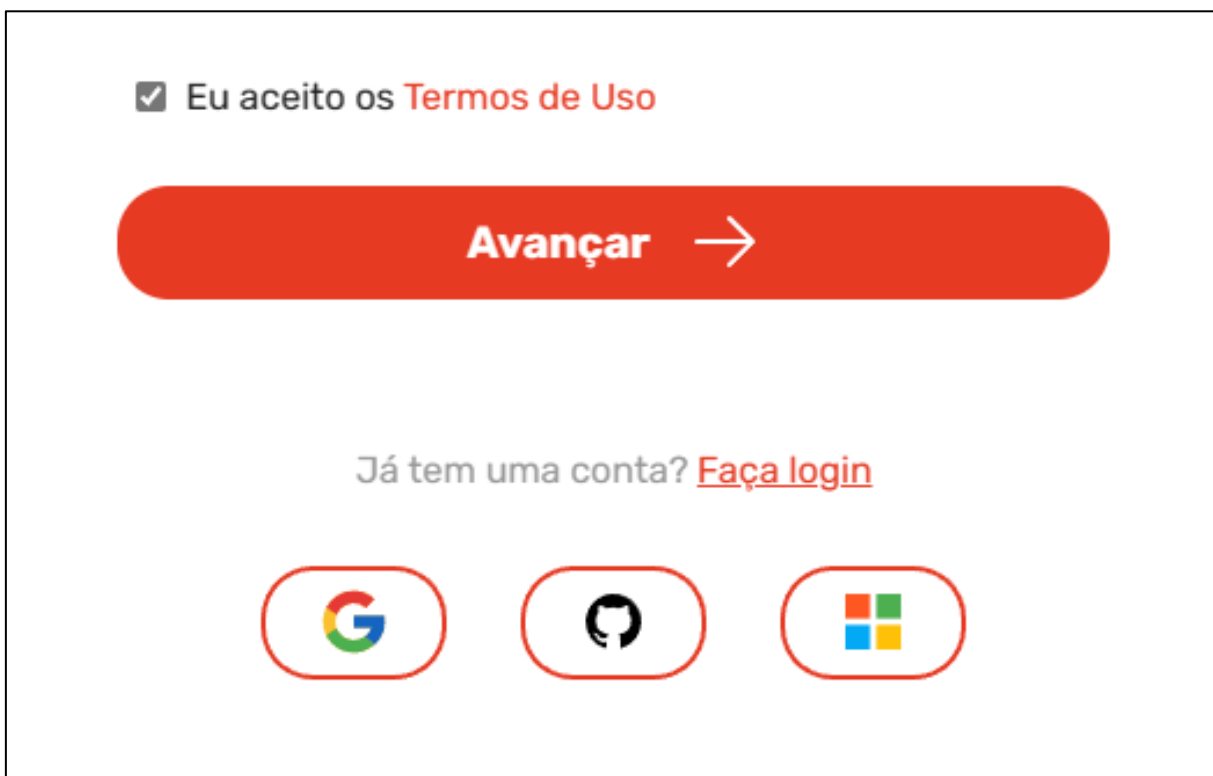
Senha secreta

Confirme sua senha

Eu aceito os [Termos de Uso](#)

Avançar

É ofertado a possibilidade de criar conta com a conta que o usuário usa do Google, GitHub e Microsoft.






This section shows the 'Eu aceito os Termos de Uso' checkbox checked, a large red 'Avançar' button with a right-pointing arrow, and the text 'Já tem uma conta? [Faça login](#)'. Below are three social login buttons: Google, GitHub, and Microsoft.

Eu aceito os [Termos de Uso](#)

Avançar →

Já tem uma conta? [Faça login](#)

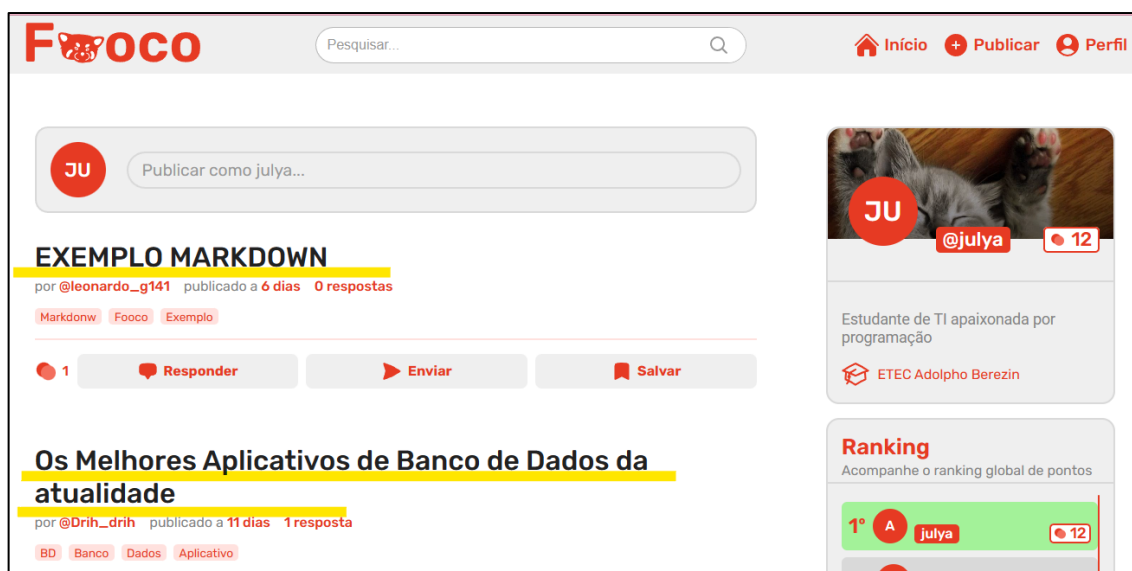
  

5.3 Conta criada, como interagir no fórum?

Conta criada com sucesso, há duas principais formas de interação no Foco, ler os conteúdos postados por outros usuários e interagir com ele, ou compartilhar alguma dúvida ou ensinamento.

As publicações são separadas por títulos e botões para interação, em cada uma delas são exibidas informações importantes como o total de pontos recebidos, as tags (posteriormente utilizadas nas pesquisas), o nome do usuário criador do conteúdo e a quantidade de respostas que a publicação recebeu.

Para poder ver uma publicação completa, é preciso clicar no título que ela recebe, dois estão sublinhados na imagem abaixo como demonstração:



Ao clicar em uma publicação, se tem acesso ao conteúdo total postado e opções de interação aparecem, como dar ou tirar um ponto do usuário dono da publicação, circulado em amarelo, compartilhar sublinhado em azul, e salvar para mais tarde sublinhado em roxo. Além disso, caso o assunto do post não seja adequado para a comunidade, é possível denunciá-lo clicando nos três pontos localizados no lado direito da publicação, sublinhado em cinza:



No campo de denúncia, após escolher o motivo mais adequado para aquela denúncia, é possível deixar um comentário justificando a denúncia, e após isso enviar ela para análise.

As categorias para denuncia são: Conteúdo ofensivo, Assédio, Spam, Desinformação, Fraude, Golpe e Outros, sendo esse para caso o usuário não encontre o motivo já listado.

Denunciar publicação

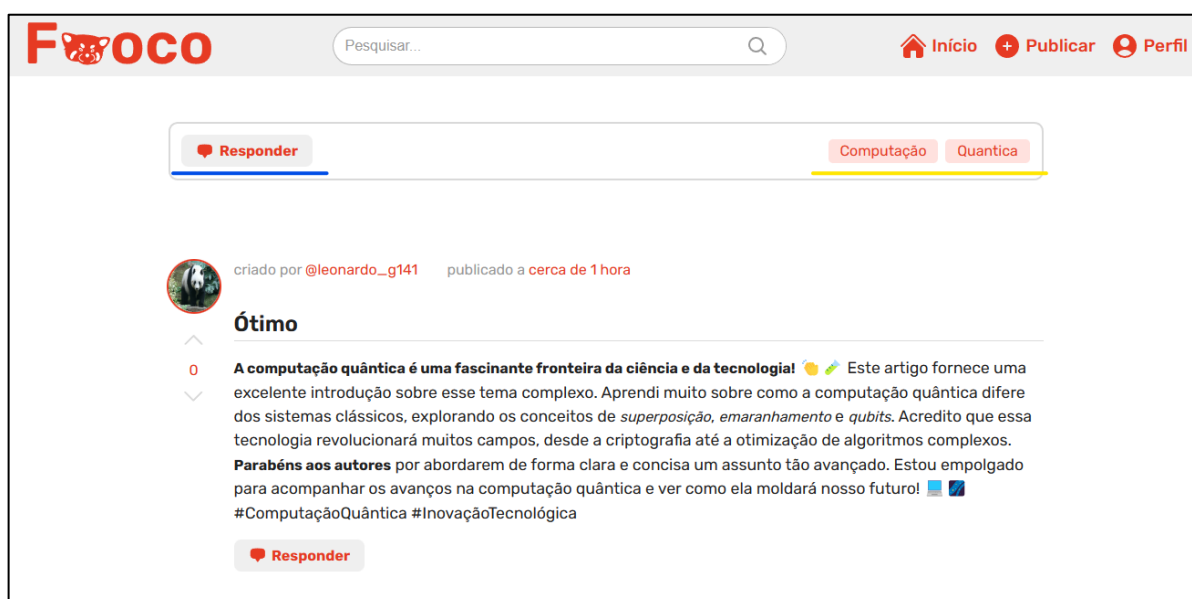
Motivo
Selecione uma opção

Comentário (opcional)
Comentários

Cancelar Enviar

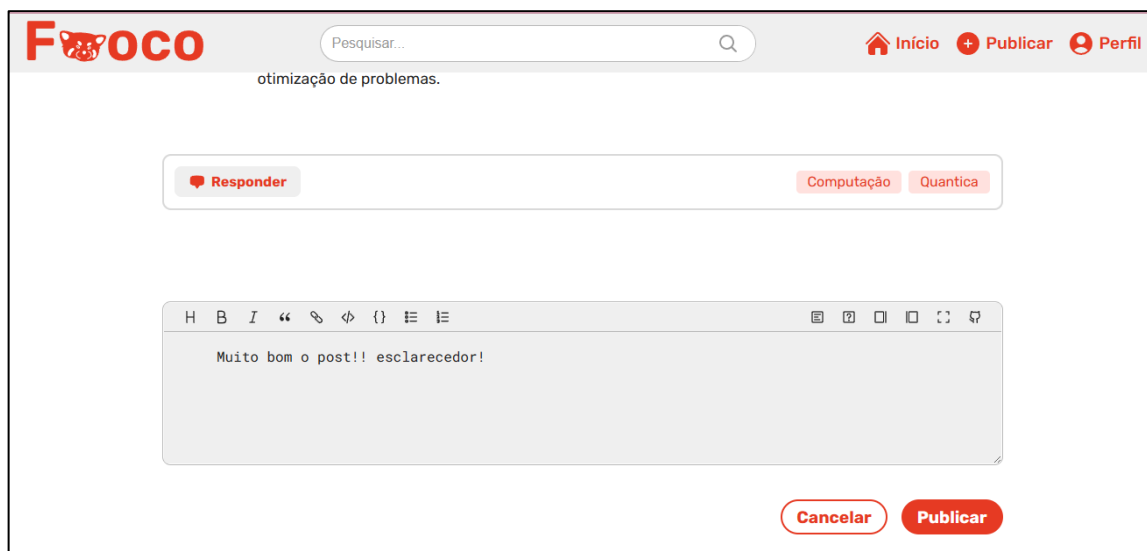


Ao final da publicação, são encontradas as *tags*, sublinhadas em amarelo, que resumem em poucas palavras o conteúdo lido e a opção de responder, sublinhada em azul, também é possível ler as respostas que outros usuários fizeram para aquela postagem:

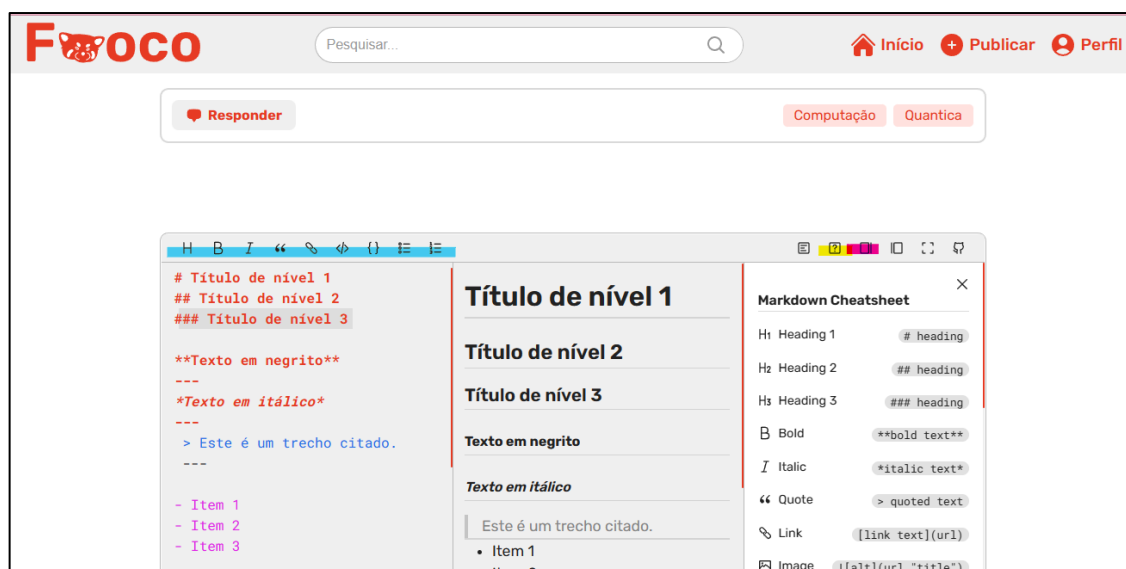


Ao escolher responder, seja para acrescentar algo ao conteúdo em questão, responder a dúvida do criador ou apenas agradecê-lo, aparecerá um campo de texto onde o usuário poderá escrever seu comentário. O campo de texto utiliza a tecnologia Markdown para oferecer mais funções de indentação e estilização, ela é utilizada na hora de criação de uma postagem.

Exemplo de comentário sem uso do *Markdown*:



Exemplo de uso do *Markdown* para indentação e estilização:

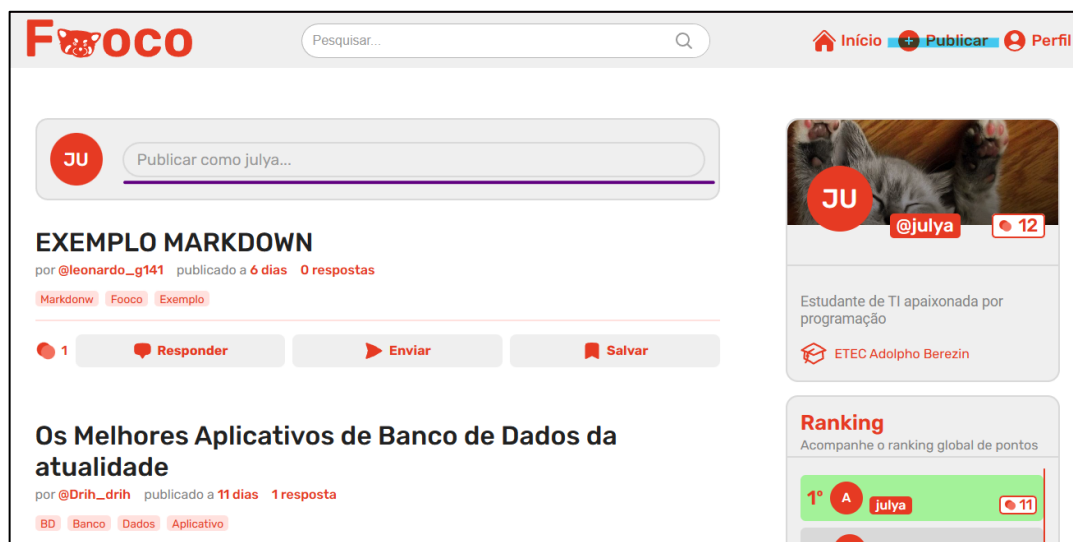


Na imagem acima é possível ver exemplos de como estilizar o texto com essa ferramenta, grifado de azul estão as principais opções de estilização como o negrito, itálico, adicionar link, organizar em lista etc. Ao lado direito está a opção de ajuda, grifada de amarelo, que mostra como fazer as estilizações disponíveis, e a opção de ver em tempo real como sua resposta está ficando após os estilos colocados, grifada de rosa.

Após revisar a resposta escrita, é possível deixá-la pública clicando em “publicar” abaixo da caixa de texto.

5.4 Como fazer uma publicação?

No começo da página inicial estão localizadas duas opções onde o usuário pode clicar e ser redirecionado para a sessão de criação de uma publicação, “publicar” grifado de azul na parte superior direita e “publicar como [nome/apelido do usuário]”.



Indo para a página de publicação, o usuário se deparará com 3 campos para a entrada de texto, o título, o conteúdo e as tags da publicação.

The screenshot shows the 'Publicar' page. At the top, the word 'Publicar' is displayed in large red font. Below it, there is a section for 'Título da publicação' with a text input field containing the placeholder 'Descreva sua publicação'. Underneath is a rich text editor with a toolbar containing icons for bold (B), italic (I), quote (”), link (🔗), unlink (🔗), code ({}), list (☰), and indent (≡). Below the editor is a 'Tags' section with a text input field containing the placeholder 'Palavras-chave da sua publicação'. At the bottom right, there are two buttons: 'Cancelar' and 'Publicar'.

Como já explicado anteriormente, para a criação e edição de uma publicação, é utilizada a linguagem de marcação de texto *Markdown*, que pode ser utilizada como forma de organizar sua publicação através de estilizações como títulos, citações, blocos de código e listas. O usuário pode optar por não utilizar as marcações *Markdown*, escrevendo diretamente o conteúdo, porém, isso pode acabar comprometendo o interesse dos outros usuários, que gostam de publicações organizadas e estilizadas.

Para as *tags*, que nada mais são do que palavras-chaves que resumem o conteúdo, o usuário deve sempre lembrar que são elas que farão a indexação da sua publicação nas pesquisas, então ele deve ser cauteloso e ao mesmo tempo criativo para criar *tags* eficazes que gerem engajamento e chamem a atenção de outros usuários.

Ao clicar em “publicar”, o usuário será redirecionado para a sua publicação que já está visível a toda comunidade.

Caso seja do interesse do usuário, ele poderá editar ou excluir o conteúdo da plataforma através do botão à direita da publicação. As opções de edição e exclusão são visíveis apenas para o dono da publicação.



**Fazendo uma Requisição HTTP em Rust:
Um Guia Passo a Passo**

criado por @leol publicado a 13 dias

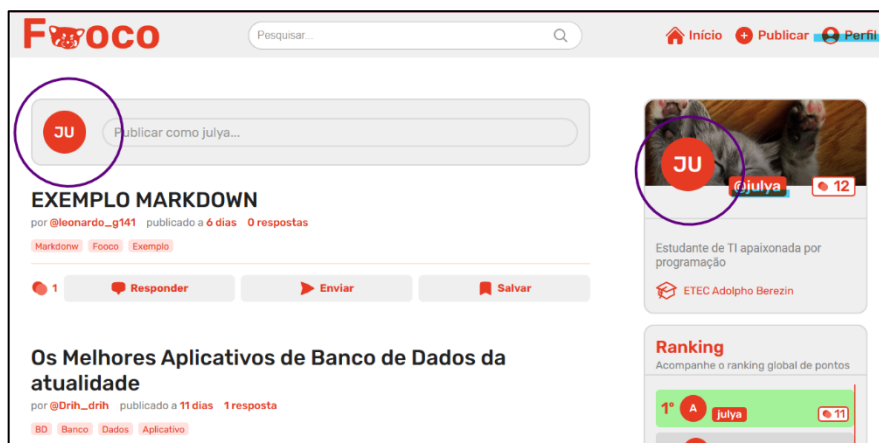
2

Fazer requisições HTTP é uma tarefa comum ao desenvolver aplicativos que precisam interagir com APIs e serviços da web. Neste post, vamos explorar como fazer uma requisição HTTP em Rust usando a biblioteca `reqwest`. O Rust é uma linguagem de programação moderna e segura, conhecida por sua performance e pela ausência de erros de memória e segurança. Vamos ver como utilizar essa poderosa biblioteca para realizar requisições HTTP de forma fácil e eficiente.

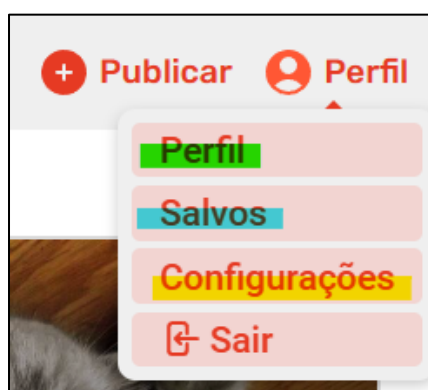
Editar
Deletar

5.5 Como modificar meu perfil e configurações

Ao criar uma conta no Fooco, há a possibilidade de estilizar o perfil com um novo apelido, nova foto, colocar uma pequena descrição e a instituição onde estuda, assim como é mostrado na imagem abaixo no canto direito:

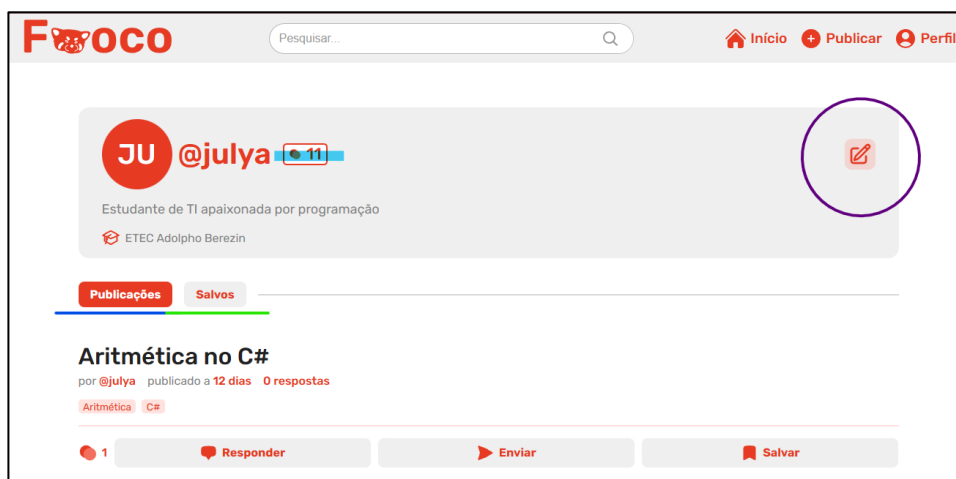


Para modificar estes dados citados, é preciso clicar em algum dos ícones circulos de roxo e grifados de azul disponíveis na tela de início principal, após isso, será redirecionado para a página de seu perfil. Clicando no “perfil” localizado no canto direito da tela, três opções são ofertadas:



Dessa forma, além da possibilidade de trocar as informações de perfil clicando no campo grifado em verde, é possível ver as publicações salvas no campo grifado em azul e acessar as configurações da conta, onde é possível trocar e-mail, senha e foto de perfil, no campo grifado em amarelo.

Na página perfil:



Os pontos obtidos com as interações pelo fórum, grifado em azul, ficam ao lado do nome do usuário. Ao canto direito e circulado em roxo é possível ver o ícone que, caso clicado, mostra as opções de estilização de perfil:



Editar perfil

Apelido
julya

Sobre
Estudante de TI apaixonada por programação

Instituição de ensino
ETEC Adolpho Berezin

Cancelar Salvar

Após realizar a estilização, é preciso clicar no campo “salvar” para que as alterações sejam salvas.

Já na página de configurações:



Fooco Pesquisar... Início Publicar Perfil

Configurações

- Conta
- Tema

Alterar informações da conta

Novo e-mail
juliane.amorim@etec.sp.gov.br

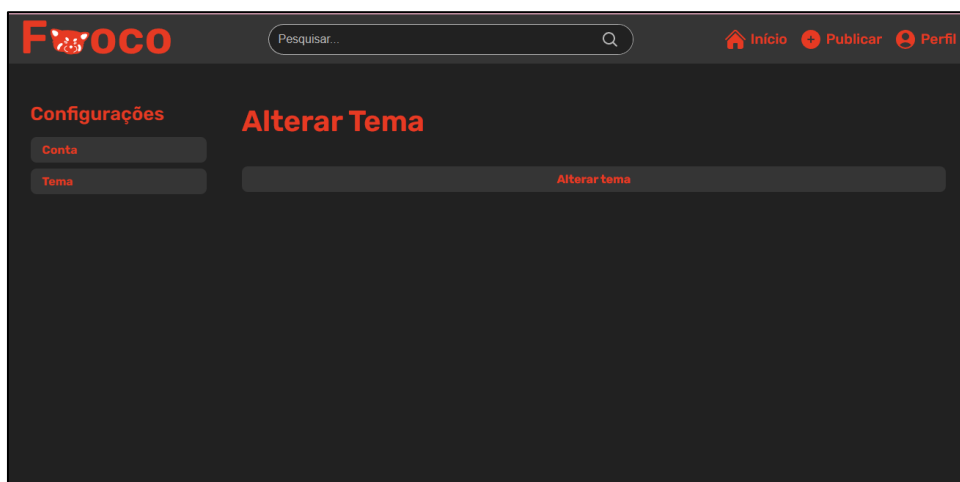
Nova senha
.....

Alterar Avatar
Nenhum arquivo escolhido

Cancelar Salvar

Nessa página, é possibilitado alterar o e-mail, a senha e colocar uma nova foto de perfil, foto essa que tem que estar armazenada no dispositivo do usuário.

Na aba “tema” localizada no canto esquerdo é oferecido ao usuário a troca de tema do claro ao escuro, caso seja de sua preferência:



5.6 Pesquisa por tags

Já sabendo como criar uma conta, interagir no post de outros estudantes e como criar o próprio e estilizar a conta como preferir, por fim, para ajudar na busca sobre assuntos específicos, existe uma pesquisa por *tag*, assim, facilitando a busca por determinadas publicações:



Sublinhado em roxo está a barra de pesquisa.

CONCLUSÃO

Em suma Foco é um fórum estudantil, onde dúvidas, problemas e ensinamentos do mundo acadêmico são publicados com o objetivo de obter e passar aprendizado, de aluno para aluno.

Se visa desenvolver a cooperatividade dos estudantes, desenvolvimento de senso de comunidade e a conexão da comunidade escolar, cujos foram prejudicados pela pandemia. Colocamos o Foco por alguns dias no ar e a adesão dos alunos foi grande, desta forma acreditamos no projeto como solução para a interação e cooperação entre alunos. Além disso, este projeto tornou-se uma grande experiência para cada integrante da equipe Animalia, no convívio profissional e na prática do desenvolvimento de sistemas.

REFERÊNCIA BIBLIOGRÁFICA

ABRES. **Estatísticas.** [S. l.], 2021. Disponível em: <https://abres.org.br/estatisticas/>. Acesso em: 6 out. 2022.

CABRAL, Lorena Machado Vicente. **O Professor Frente ao Aluno Desperiodizado Buscando Valores e Novos Desafios Presentes na Sociedade.** Publicado em: Produção Didático-Pedagógica (2010), O Professor Pde E Os Desafios Da Escola Pública Paranaense, Volume II, Secretaria De Estado Da Educação Do Paraná – Seed, Superintendência Da Educação, Coordenação Estadual Do Pde, Curitiba (2011), disponível em: http://www.diaadiaeducacao.pr.gov.br/portals/cadernospde/pdebusca/producoes_pde/2010/2010_ufpr_ped_pdp_lorena_machado_vicente_cabral.pdf Acesso em: 05 do mês 09 de 2022.

DATA FOLHA. 43 milhões de brasileiros acessam internet por dispositivos móveis. São Paulo: Folha de São Paulo, 2014. Disponível em: <https://datafolha.folha.uol.com.br/mercado/2014/01/1400618-43-milhoes-de-brasileiros-acessam-internet-por-dispositivos-moveis.shtml>. Acesso em: 9 set. 2022.

EDUCADOR360. **Os benefícios da Gamificação na educação.** [S. l.], 24 ago. 2021. Disponível em: <https://educador360.com/matricula/os-beneficios-da-gamificacao-na-educacao/>. Acesso em: 6 out. 2022.

FAZ EDUCAÇÃO & TECNOLOGIA. **Dificuldades de socialização pós-pandemia.** Disponível em: Dificuldades Acesso em: 25 maio. 2023.

G, Ariane. **O que é CSS? Guia Básico para Iniciantes.** 7 out. 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>. Acesso em: 31 out. 2022.

HOSPITAL SANTA MÔNICA. **Por que a falta do convívio social impacta na nossa saúde mental?** Hospital Santa Mônica. Disponível em:

<https://hospitalsantamonica.com.br/por-que-a-falta-do-convivio-social-impacta-na-nossa-saude-mental/>. Acesso em: 9 nov. 2022.

JORNAL NACIONAL. **IBGE mede o problema nacional da evasão escolar.** G1, 2020. Disponível em: <https://g1.globo.com/jornal-nacional/noticia/2020/07/15/ibge-mede-o-problema-nacional-da-evasao-escolar.ghtml>. Acesso em: 14 set. 2022.

KRIGER, Bruno. **O QUE É HTML, QUAL A SUA FUNÇÃO E COMO APRENDER HTML?** 2 dez. 2020. Disponível em: <https://kenzie.com.br/blog/html/>. Acesso em: 31 out. 2022

KRIGER, Daniel. **O QUE SÃO FRAMEWORKS E POR QUE SÃO IMPORTANTES PARA OS DEVS.** 27 maio 2022. Disponível em: <https://kenzie.com.br/blog/framework/>. Acesso em: 31 out. 2022.

MOBILE, Use. **React Native:** saiba o que é, vantagens e funcionalidades. 15 ago. 2022. Disponível em: <https://usemobile.com.br/react-native/>. Acesso em: 31 out. 2022

MYSQL. **MySQL: MySQL Workbench.** Disponível em: <https://www.mysql.com/products/workbench/>. Acesso em: 1 nov. 2022.

NOLETO, Cairo. **Javascript: o que é, aplicação e como aprender a linguagem JS.** blog.betrybe.com. Disponível em: <https://blog.betrybe.com/javascript/>. Acesso em: 9 nov. 2022.

PRADO, Isadora da Cunha. **O USO DAS MÍDIAS SOCIAIS DURANTE A PANDEMIA DO COVID-19.** Orientador: Jussara Goulart da Silva. 2021. 23 f. Trabalho de conclusão de curso (Graduação em Administração) - Universidade Federal de Uberlândia, Ituiutaba, 2021. Disponível em: <https://repositorio.ufu.br/handle/123456789/33332>. Acesso em: 6 out. 2022.

SOARES, João Paulo. **Acessibilidade em projetos web:** como pensar desde o início. 20 dez. 2022. Disponível em:

<https://www.treinaweb.com.br/blog/acessibilidade-em-projetos-web-como-pensar-desde-o-inicio>. Acesso em: 31 out. 2022

STACK OVERFLOW. **Stack Overflow Developer Survey 2020**. Stack Overflow. Disponível em: <https://insights.stackoverflow.com/survey/2020>. Acesso em: 31 out. 2022.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL: the world's Most Advanced Open Source Database**. Disponível em: <<https://www.postgresql.org/>>. Acesso em: 3 mar. 2023.

TYPEORM. **TypeORM - Amazing ORM for TypeScript and JavaScript**. Disponível em: <<https://typeorm.io/>>. Acesso em: 3 mar. 2023.

VILLAIN, Mateus. **Figma: o que é a ferramenta, Design e uso**. 20 out. 2022. Disponível em: <https://www.alura.com.br/artigos/figma>. Acesso em: 31 out. 2022