



CENTRO UNIVERSITÁRIO DA GRANDE DOURADOS

CAIO ALVES CAMPOS

CRYPTOQUEST

Dourados MS
2025



CENTRO UNIVERSITÁRIO DA GRANDE DOURADOS

CAIO ALVES CAMPOS

CRYPTOQUEST

Trabalho de Conclusão de Curso
apresentado ao Curso de Engenharia de
Software da Faculdade de Ciências Exatas e
Agrárias como pré-requisito para obtenção
do título de Bacharel em Engenharia de
Software.

Orientador: Prof. MSc. Jose Gonçalves Dias
Neto

Dourados MS
2025

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SUMÁRIO

1	ESCOPO DO SISTEMA	1
1.1	DADOS INICIAIS	1
1.2	MOTIVAÇÃO E PROBLEMÁTICA ABORDADA PELO SOFTWARE	2
1.2.1	Definição e importância	2
1.2.2	Contextualização	2
1.2.3	O Público-alvo	2
1.3	JUSTIFICATIVA DO PROJETO.....	3
1.4	ENTREGAS DO PROJETO	3
1.5	OBJETIVOS DO SISTEMA.....	3
1.6	CRITÉRIOS DE ACEITAÇÃO DO SISTEMA	4
1.7	CONSULTOR DO SISTEMA	4
1.8	ENTREVISTA COM O CONSULTOR DO SISTEMA.....	4
2	REQUISITOS DO SISTEMA	7
2.1	METODOLOGIA DE LEVANTAMENTO DE REQUISITOS	7
2.2	REQUISITOS	7
2.3	MATERIAIS E MÉTODOS (LINGUAGEM E FERRAMENTAS UTILIZADAS) 9	
2.3.1	Casos de Usos Gerais.....	10
2.3.2	Atores envolvidos.....	14
2.4	CASOS DE USO ESPECÍFICOS.....	15
2.4.1	Cadastrar Usuários	15
2.4.2	Responder Questionário Inicial	17
2.4.3	Responder Quizz	19
2.4.4	Acessar Conquistas e Recompensas.....	22

2.4.5	Personalizar Perfil.....	24
2.5	ARQUITETURA DO SISTEMA	26
2.6	DIAGRAMAS	29
2.6.1	Modelo de Classes (ou DER se não usar Orientação a Objetos)	29
3	DESENVOLVIMENTO DE SOFTWARE	30
3.1	METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE	30
3.1.1	Ambientes de desenvolvimento/produção	30
3.1.2	Bibliotecas principais.....	30
3.2	MÓDULOS DO CRONOGRAMA DE DESENVOLVIMENTO	31
3.3	MOCKUPS.....	31
4	CONCLUSÃO	38
5	REFERÊNCIAS	39

HISTÓRICO DE ALTERAÇÕES

Data	Versão	Descrição	Autor

O versionamento do documento será feito utilizando os parâmetros baseados na metodologia semver. O documento só será considerado na versão 1.0 quando completar os capítulos 1, 2 e 3. Toda alteração no documento deve constar na tabela acima.

		Versionamento numeração x.y.z
X	MAJOR	Alterações drásticas (Inclusão/Alteração Caso de Uso Geral) Adição de novos capítulos (4 e 5)
Y	MINOR	Adição/Remoção de Funcionalidades
z	PATCH	Correções ortográficas e/ou tipográficas

CONVENÇÕES, TERMOS E ABREVIACÕES

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

Identificação dos requisitos

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos, seguidos do identificador do requisito, de acordo com a especificação a seguir:

[nome da subseção. identificador do requisito]

Por exemplo, o requisito funcional [Recuperação de dados. RF016] deve estar descrito em uma subseção chamada “Recuperação de dados” (que indica um subsistema), em um bloco identificado pelo número [RF016]. Já o requisito não-funcional [Confiabilidade.NF008] deve estar descrito na seção de requisitos não-funcionais de Confiabilidade, em um bloco identificado por [NF008].

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF001] ou [NF001] e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

Prioridades dos requisitos

Para estabelecer a prioridade dos requisitos, nos capítulos 3 e 4, foram adotadas as denominações “essencial”, “importante” e “desejável”.

Essencial é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.

Importante é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.

Desejável é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

1 ESCOPO DO SISTEMA

1.1 DADOS INICIAIS

Nome do software:

CryptoQuest

Patrocinador

Caio Alves Campos

Público-alvo

- **Investidores iniciantes**, que desejam ingressar no mercado de criptomoedas, mas enfrentam dificuldades devido à falta de conhecimento técnico e insegurança na tomada de decisões financeiras.
- **Estudantes e profissionais de tecnologia**, interessados em entender os fundamentos da blockchain e como essa tecnologia impacta diversas áreas.
- **Entusiastas do setor financeiro**, que querem expandir seus conhecimentos sobre ativos digitais e economia descentralizada.
- **Empresas e instituições**, que precisam capacitar seus funcionários sobre o funcionamento das criptomoedas e suas aplicações no mercado.

Stakeholders

Investidores e entusiastas de Criptomoedas e BlockChain

Equipe Básica

Analistas/Desenvolvedores:

Caio Alves Campos

Orientadores:

Jose Gonçalves Dias Neto
André Martins Do Nascimento

Consultor:

Alex Viana Pereira

1.2 MOTIVAÇÃO E PROBLEMÁTICA ABORDADA PELO SOFTWARE

1.2.1 Definição e importância

O CryptoQuest se insere no ramo de educação Digital, com foco no ensino de criptomoedas, blockchain e investimentos financeiros. Esse setor tem crescido significativamente nos últimos anos, impulsionado pela transformação digital e pela necessidade crescente de aprendizado acessível, interativo e eficiente.

1.2.2 Contextualização

Com o crescimento das criptomoedas e da tecnologia blockchain, a falta de conhecimento técnico se tornou um grande obstáculo para investidores e entusiastas. Muitas pessoas entram nesse mercado sem saber dos riscos e fundamentos, o que pode levar a decisões financeiras ruins. A gamificação tem sido uma estratégia eficaz para tornar o aprendizado mais envolvente e acessível, permitindo que usuários adquiram conhecimento de forma prática e interativa. Assim, uma plataforma educacional gamificada pode ajudar a reduzir essa barreira, tornando o ensino de criptomoedas mais dinâmico e eficiente.

1.2.3 O Público-alvo

O **CryptoQuest** é uma plataforma educacional gamificada que tem como público-alvo pessoas interessadas em aprender sobre criptomoedas, blockchain e economia digital. O sistema visa atender diferentes perfis de usuários, incluindo:

- **Investidores iniciantes**, que desejam ingressar no mercado de criptomoedas, mas enfrentam dificuldades devido à falta de conhecimento técnico e insegurança na tomada de decisões financeiras.
- **Estudantes e profissionais de tecnologia**, interessados em entender os fundamentos da blockchain e como essa tecnologia impacta diversas áreas.
- **Entusiastas do setor financeiro**, que querem expandir seus conhecimentos sobre ativos digitais e economia descentralizada.
- **Empresas e instituições**, que precisam capacitar seus funcionários sobre o funcionamento das criptomoedas e suas aplicações no mercado.

Atualmente, esses usuários enfrentam desafios significativos para adquirir conhecimento sobre o tema. A falta de materiais didáticos acessíveis e interativos

torna o aprendizado pouco atrativo. Além disso, muitos cursos disponíveis são teóricos e extensos, o que dificulta o engajamento e a retenção de conhecimento. O desconhecimento técnico também torna os iniciantes vulneráveis a golpes e investimentos inseguros.

Para solucionar esses problemas, o **CryptoQuest** utilizará gamificação e inteligência artificial para personalizar o aprendizado e tornar o ensino mais dinâmico e eficaz.

1.3 JUSTIFICATIVA DO PROJETO

A falta de conhecimento sobre criptomoedas resulta em investimentos inseguros, vulnerabilidade a golpes e resistência à adoção do blockchain. A proposta desse projeto é criar um ambiente interativo que facilite o aprendizado, utilizando inteligência artificial para personalizar o ensino e data mining para identificar dificuldades em comum entre usuários. A gamificação mantém o engajamento e incentiva a participação ativa tornando a experiência mais atrativa. Dessa forma a plataforma contribuirá para maior compreensão desse mercado.

1.4 ENTREGAS DO PROJETO

- Documento de Requisitos
- Sistema codificado com os requisitos implementados

1.5 OBJETIVOS DO SISTEMA

O sistema tem como objetivos:

- **Fornecer um ambiente educacional interativo** para o ensino de conceitos sobre criptomoedas e blockchain, utilizando desafios e quizzes gamificados.
- **Personalizar o aprendizado por meio de inteligência artificial**, ajustando a dificuldade dos desafios conforme o nível de conhecimento do usuário.
- **Criar um sistema de progressão**, permitindo que os usuários avancem de nível ao completarem desafios e missões.
- **Oferecer recompensas e rankings** para aumentar o engajamento dos usuários na plataforma.

- **Coletar e analisar dados sobre o desempenho dos usuários**, identificando dificuldades comuns e aprimorando a experiência de aprendizado.
- **Facilitar o acesso à educação financeira**, tornando o aprendizado sobre criptomoedas acessível para qualquer pessoa, independentemente do seu nível de conhecimento.

1.6 CRITÉRIOS DE ACEITAÇÃO DO SISTEMA

Todas as funcionalidades do website devem ser testadas através do emprego de:

- Testes de Usabilidade;
- Testes de Software;
- Teste nos Navegadores;

1.7 CONSULTOR DO SISTEMA

Alex Viana Pereira, possui graduação em Graduação em Matemática pela Faculdade Cidade Verde (2024) e graduação em Física pela Universidade Estadual de Mato Grosso do Sul (2012). Atualmente é professor da Escola Nossa Senhora Aparecida - Maracaju e professor da Escola Estadual Padre Constantino de Monte.

1.8 ENTREVISTA COM O CONSULTOR DO SISTEMA

01) Como medir o progresso e o engajamento do usuário em uma plataforma de aprendizado?

R: Medir o progresso e o engajamento do usuário é essencial para garantir que a plataforma está realmente sendo eficaz. Para isso, podemos utilizar algumas estratégias:

- **Sistema de progressão:** Implementar níveis, conquistas e recompensas virtuais para incentivar os usuários a continuarem aprendendo.
- **Monitoramento de métricas:** Avaliar dados como tempo médio de uso, frequência de acesso, taxa de conclusão das atividades e quantidade de tentativas em quizzes.
- **Relatórios visuais:** Painéis interativos e gráficos ajudam tanto os usuários quanto os administradores a acompanharem o progresso de maneira clara.

02) Você vê alguma desvantagem no uso de quizzes e desafios para aprendizado?

R: Os quizzes são uma ferramenta poderosa para aprendizado, mas devem ser bem estruturados para evitar algumas armadilhas, como:

- **Memorização mecânica:** Se os quizzes forem apenas questões de múltipla escolha sem contexto real, os usuários podem decorar as respostas sem realmente aprender.
- **Falta de contextualização:** Quizzes sem conexão com o dia a dia do usuário podem parecer irrelevantes e menos eficazes no aprendizado.
- **Frustração:** Questões mal formuladas ou excessivamente difíceis podem levar à desmotivação.

Para evitar esses problemas, é importante criar quizzes desafiadores, mas equilibrados, variando os formatos (simulações, perguntas abertas, estudos de caso) e oferecendo feedback explicativo em cada resposta.

03) Quais são as melhores práticas para criar quizzes educativos que realmente testem e reforcem o aprendizado do usuário?

R: Para que um quiz seja realmente eficaz, é fundamental seguir algumas boas práticas:

- **Adaptação da dificuldade:** Os quizzes devem se ajustar ao nível de conhecimento do usuário, para evitar frustração ou tédio.
- **Conexão com a realidade:** As perguntas devem estar ligadas a situações práticas do cotidiano.
- **Variedade de formatos:** Misturar múltipla escolha, arrastar e soltar, perguntas abertas e estudos de caso para tornar o aprendizado mais dinâmico.
- **Feedback detalhado:** Após cada resposta, o usuário deve receber explicações sobre o motivo da resposta correta ou incorreta.
- **Gamificação:** O uso de recompensas e desafios progressivos pode tornar os quizzes mais envolventes.

04) Como podemos tornar o conteúdo acessível para pessoas que não têm familiaridade com tecnologia e finanças?

R: Para tornar o conteúdo acessível a iniciantes, algumas estratégias são fundamentais:

- **Evitar jargões técnicos:** Introduzir os conceitos de forma simples, sem excesso de termos técnicos no início.
- **Uso de metáforas e analogias:** Explicar conceitos complexos com comparações fáceis de entender (exemplo: "O blockchain é como um livro contábil gigante, onde cada página representa um bloco").

- **Histórias e narrativas:** Criar uma jornada interativa, onde o usuário se veja como parte do aprendizado, pode facilitar a assimilação do conteúdo.
 - **Interatividade:** Vídeos curtos, animações e simulações podem ser mais eficazes do que apenas texto.
-

05) Existe alguma recomendação sobre o tempo ideal de uma atividade educativa dentro do aplicativo para evitar fadiga ou perda de interesse?

R: Estudos sobre aprendizado digital mostram que sessões curtas são mais eficazes para manter o engajamento. Algumas recomendações incluem:

- **Duração ideal:** Atividades entre **5 a 15 minutos** evitam a fadiga mental e aumentam a retenção do conhecimento.
 - **Microlearning:** Dividir o conteúdo em pequenas partes facilita o aprendizado e mantém o usuário motivado.
 - **Sistema de pausa e retomada:** Permitir que o usuário pause a atividade e continue de onde parou ajuda na adaptação ao seu ritmo de aprendizado.
 - **Desafios rápidos:** Pequenos testes ao longo do aprendizado ajudam a reforçar o conteúdo sem tornar a experiência cansativa.
-

06) Você conhece alguma referência ou estudo sobre aprendizado digital que possa ser útil para o desenvolvimento desse sistema?

R: Existem várias referências importantes sobre aprendizado digital e gamificação. Algumas das principais são:

- **Plataformas de referência:** Duolingo e Khan Academy são exemplos práticos de como aprendizado digital pode ser interativo e gamificado.

Esses materiais podem fornecer visões valiosas para o desenvolvimento da plataforma.

07) Como podemos garantir que o aprendizado seja contínuo e não apenas baseado em desafios pontuais (exemplo: usuário faz um quiz e depois nunca mais volta ao app)?

R: Para garantir que o usuário continue utilizando a plataforma de forma constante, algumas estratégias podem ser aplicadas:

- **Sistema de progressão contínua:** Criar desafios e conteúdos desbloqueáveis para manter o interesse do usuário.

- **Notificações inteligentes:** Mensagens estratégicas para lembrar o usuário de continuar aprendendo, mas sem exageros para evitar spam.
- **Conteúdo sempre atualizado:** Adicionar novidades e tópicos relevantes mantém o usuário engajado.
- **Funcionalidades sociais:** Fóruns, desafios entre amigos e ranking de pontuações podem incentivar a participação contínua.

2 REQUISITOS DO SISTEMA

Neste capítulo, serão apresentados os aspectos técnicos do projeto a ser desenvolvido.

2.1 METODOLOGIA DE LEVANTAMENTO DE REQUISITOS

O projeto terá início em marco de 2025, com as etapas de planejamento e desenvolvimento do sistema. O projeto será dividido em algumas etapas:

A primeira etapa consiste na realização de uma entrevista com um especialista em educação. Esse profissional ajudará a definir as melhores estratégias para tornar o aprendizado sobre criptomoedas, blockchain mais eficaz. Após a entrevista, serão gerados um relatório detalhado com as respostas do consultor e uma análise de recomendação para inclusão do projeto.

A segunda etapa será a análise etnográfica que consiste na observação do comportamento do público alvo em relação ao aprendizado sobre criptomoedas. Essa análise pode ser feita por meio de pesquisas de mercado, análise de concorrentes e semelhantes, buscas em fóruns etc. Será gerado um relatório de análise de comportamento do usuário.

A terceira etapa será a prototipação inicial, após a análise das entrevistas e pesquisa etnográfica, será criada um protótipo interativo do sistema. Esse sistema será testado para validar se as funcionalidades atendem a expectativa. Ao final dessa etapa, serão gerados relatórios com feedback dos primeiros testes de usabilidade e um protótipo navegável.

2.2 REQUISITOS

O sistema deverá prover os seguintes requisitos:

Requisitos Funcionais

- **RF001** - O sistema deve cadastrar e autenticar usuários.
- **RF002** - O sistema deve permitir que os usuários personalizem seus perfis.
- **RF003** - O sistema deve permitir que o usuário avance conforme seu desempenho.
- **RF004** - O sistema deverá aplicar o uso de IA para recomendar conteúdos de acordo com o nível de experiência do usuário.
- **RF005** - O sistema deve fornecer pontos e medalhas para usuários que concluírem desafios e quizzes.
- **RF006** - O sistema deve possuir um ranking de usuários.
- **RF007** - O sistema deve fornecer um sistema de missões diárias com recompensas para engajar usuários.
- **RF008** - A IA deve adaptar a dificuldade dos quizzes conforme a evolução do usuário.
- **RF009** - O Sistema deve permitir que o usuário realize simulações de compra e venda em tempo real.
- **RF010** - O sistema deve coletar, armazenar e analisar os dados de desempenho dos usuários para identificar padrões de aprendizado e dificuldades.

Requisitos Não-Funcionais

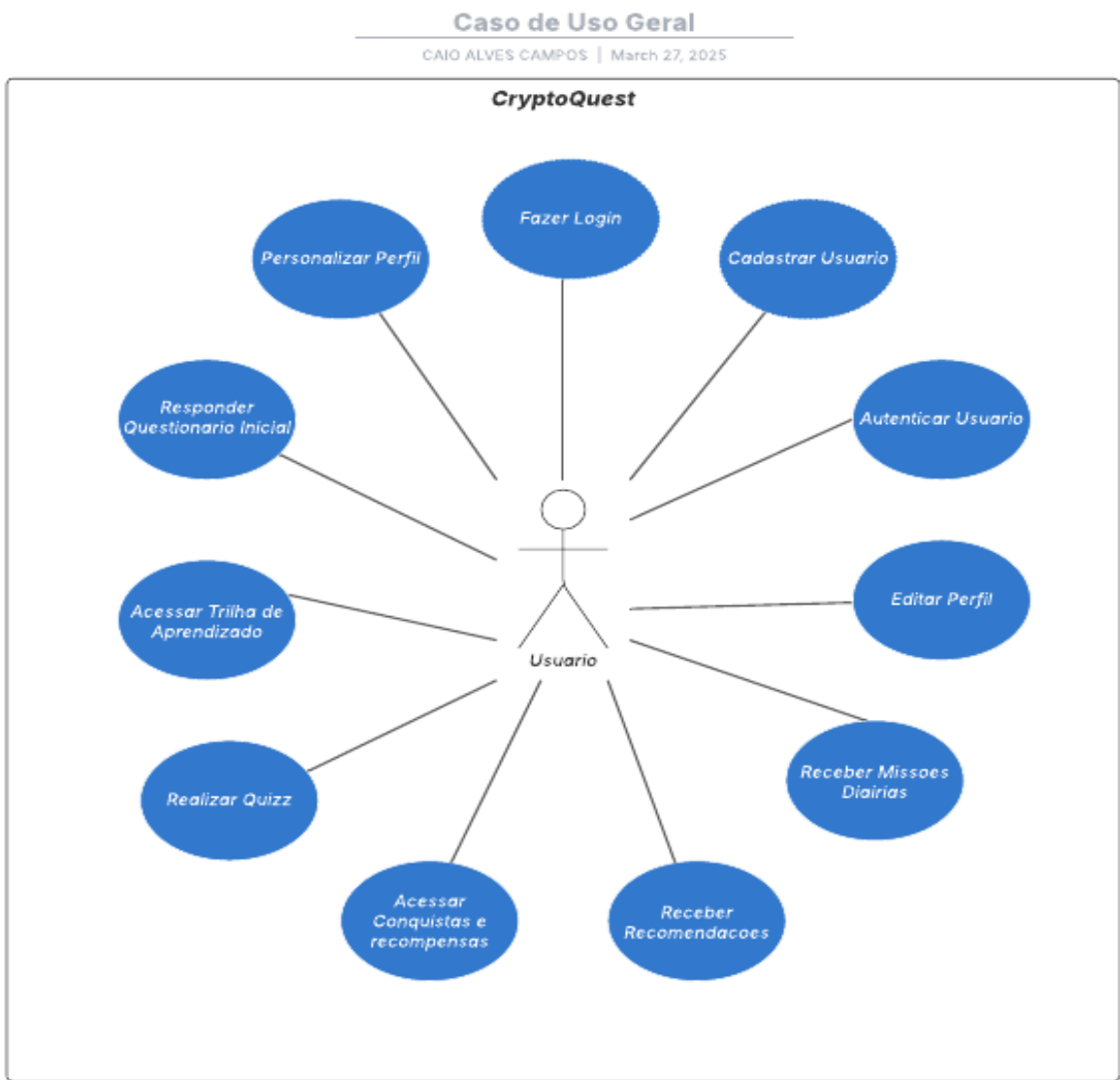
- **NF001** - O servidor deve suportar pelo menos 2.000 usuários simultâneos.
- **NF002** - O banco de dados Firebase Firestore deve ser otimizado para consultas rápidas.
- **NF003** - Os dados dos usuários devem ser armazenados de forma criptografada (AES-256 para dados sensíveis e SHA-256 para senhas).
- **NF004** - A autenticação do usuário deve ser segura, utilizando Firebase Auth com suporte a OAuth (Google, Facebook, etc.).
- **NF005** - A IA e o Data Mining devem respeitar a privacidade dos usuários, garantindo que dados sensíveis não sejam compartilhados sem consentimento.
- **NF006** - O sistema deve estar em conformidade com a LGPD (Lei Geral de Proteção de Dados) e GDPR (General Data Protection Regulation).
- **NF007** - A interface deve ser intuitiva e acessível, garantindo que qualquer usuário consiga navegar sem dificuldades.
- **NF008** - A curva de aprendizado deve ser progressiva, iniciando com conceitos básicos e aumentando a dificuldade de forma gradual.
- **NF009** - As notificações devem ser personalizáveis, permitindo que o usuário escolha a frequência e os tipos de alertas recebidos.
- **NF010** - O sistema deve ser compatível com Android (a partir do Android 8.0) e iOS (a partir do iOS 12).
- **NF011** - O aplicativo deve ser responsivo, garantindo boa experiência tanto em smartphones quanto em tablets.

- **NF012** -O backend deve ser implementado com APIs RESTful, permitindo futura integração com plataformas web e outras aplicações.
- **NF013** - A IA deve ser treinada para funcionar tanto online quanto offline, garantindo aprendizado contínuo
- **NF014** - O código-fonte deve seguir boas práticas de desenvolvimento, utilizando padrões como MVC ou Clean Architecture.
- **NF015** - O código deve ser versionado no GitHub, utilizando GitFlow para organização das branches.

2.3 MATERIAIS E MÉTODOS (LINGUAGEM E FERRAMENTAS UTILIZADAS)

O CryptoQuest será desenvolvido utilizando Flutter para o front-end, Python com FastAPI para o backend, Firebase Firestore para o banco de dados e Firebase Auth para autenticação. O uso de Inteligência Artificial e Data Mining permitirá personalizar a experiência de aprendizado, enquanto a infraestrutura em Firebase Hosting e Google Cloud garantirá escalabilidade e confiabilidade. Essa abordagem permitirá criar um aplicativo interativo, acessível e inovador para ensinar criptomoedas de maneira gamificada.

2.3.1 Casos de Usos Gerais



RF 1 - Fazer Login

Descrição: Este caso de uso permite que o usuário faça login com suas credenciais para acessar a plataforma. Caso os dados estejam corretos, o usuário é autenticado e redirecionado para a página inicial.

Prioridade: Essencial

Pré-condições: O usuário já deve ser cadastrado no sistema e deve fornecer credenciais validas.

Entrada: Nome de usuário ou Email e senha.

Saída e pós-condições: Se as credenciais forem validas, o usuário é autenticado e redirecionado a tela inicial. Caso contrário, o sistema exibe uma mensagem de erro.

RF 2 - Cadastrar Usuário

Descrição: Este caso de uso permite que novos usuários se cadastrem no sistema, fornecendo informações básicas como nome, email e senha.

Prioridade: Essencial

Pré-condições: O usuário não deve estar cadastrado previamente. O sistema deve validar senhas e emails conforme regras estabelecidas.

Entrada: Nome completo, Email valido e senha forte.

Saída e pós-condições: O usuário recebe uma confirmação de cadastro. Caso as credenciais não sejam validadas ou já estejam em uso, o sistema exibe uma imagem de erro. Os dados do usuário são armazenados no banco de dados de acordo com a LGPD.

RF 3 - Autenticar Usuário

Descrição: Valida as credenciais do usuário para garantir que ele tenha permissão de acessar a plataforma

Prioridade: Essencial

Pré-condições: O usuário deve ter sido cadastrado previamente.

Entrada: Nome de usuário ou email e senha.

Saída e pós-condições: Caso seja autenticado, o usuário pode acessar o sistema, caso contrário uma mensagem de erro será exibida.

RF 4 – Editar Perfil

Descrição: Este caso de uso permite que o usuário consiga alterar informações da sua conta, como nome, email, senha.

Prioridade: Média

Pré-condições: O usuário deve ser autenticado pelo sistema.

Entrada: Dados atualizados do perfil.

Saída e pós-condições: Os dados do usuário são alterados no banco de dados. O sistema confirma e salva as atualizações.

RF 5 – Personalizar Perfil

Descrição: Este caso de uso permite que o usuário consiga personalizar elementos visuais do seu perfil, como avatar, conquistas, preferencias de exibição etc.

Prioridade: Baixa

Pré-condições: O usuário deve estar autenticado.

Entrada: Novas preferencias do perfil.

Saída e pós-condições: O perfil do usuário é atualizado conforme as escolhas feitas.

RF 6 – Responder Questionário Inicial

Descrição: Apresenta um questionário para entender o nível de conhecimento do usuário e personalizar sua trilha de aprendizado.

Prioridade: Alta

Pré-condições: O usuário deve estar autenticado.

Entrada: Respostas do questionário.

Saída e pós-condições: O sistema ajusta a trilha de aprendizado do usuário com base nas respostas.

RF 7 – Acessar trilha de Aprendizado

Descrição: Permite que o usuário visualize seu progresso na trilha de aprendizado.

Prioridade: Essencial

Pré-condições: O usuário deve estar autenticado.

Entrada: Nenhuma.

Saída e pós-condições: O usuário visualiza sua trilha de aprendizado e pode acessar conteúdos recomendados.

RF 8 – Realizar Quizz

Descrição: Permite que o usuário responda as perguntas para testar seus conhecimentos.

Prioridade: Essencial

Pré-condições: O usuário deve estar autenticado.

Entrada: Resposta do usuário para as perguntas do quizz.

Saída e pós-condições: O sistema exibe o resultado do quizz e atualiza a pontuação do usuário, assim como retorna as respostas do questionário.

RF 9 – Acessar Conquistas e Recompensas

Descrição: Permite que o usuário visualize suas conquistas, resgate recompensas, e visualize futuras conquistas.

Prioridade: Média

Pré-condições: O usuário deve estar autenticado.

Entrada: Nenhuma.

Saída e pós-condições: O sistema mostra as conquistas do usuário, permite o resgate de recompensas disponíveis.

RF 10 – Receber Missões Diárias

Descrição: O sistema deve apresentar desafios diários para incentivar o aprendizado contínuo do usuário.

Prioridade: Média

Pré-condições: O usuário deve estar autenticado.

Entrada: Nenhuma.

Saída e pós-condições: O sistema deve gerar desafios diários com recompensas para o usuário ao completá-lo.

RF 11 – Receber recomendações

Descrição: O sistema sugere conteúdos, quizzes, e desafios com base no conhecimento e desempenho do usuário.

Prioridade: Alta

Pré-condições: O usuário deve estar autenticado.

Entrada: Histórico de interações do usuário no sistema.

Saída e pós-condições: O sistema exibe sugestões personalizados para o usuário.

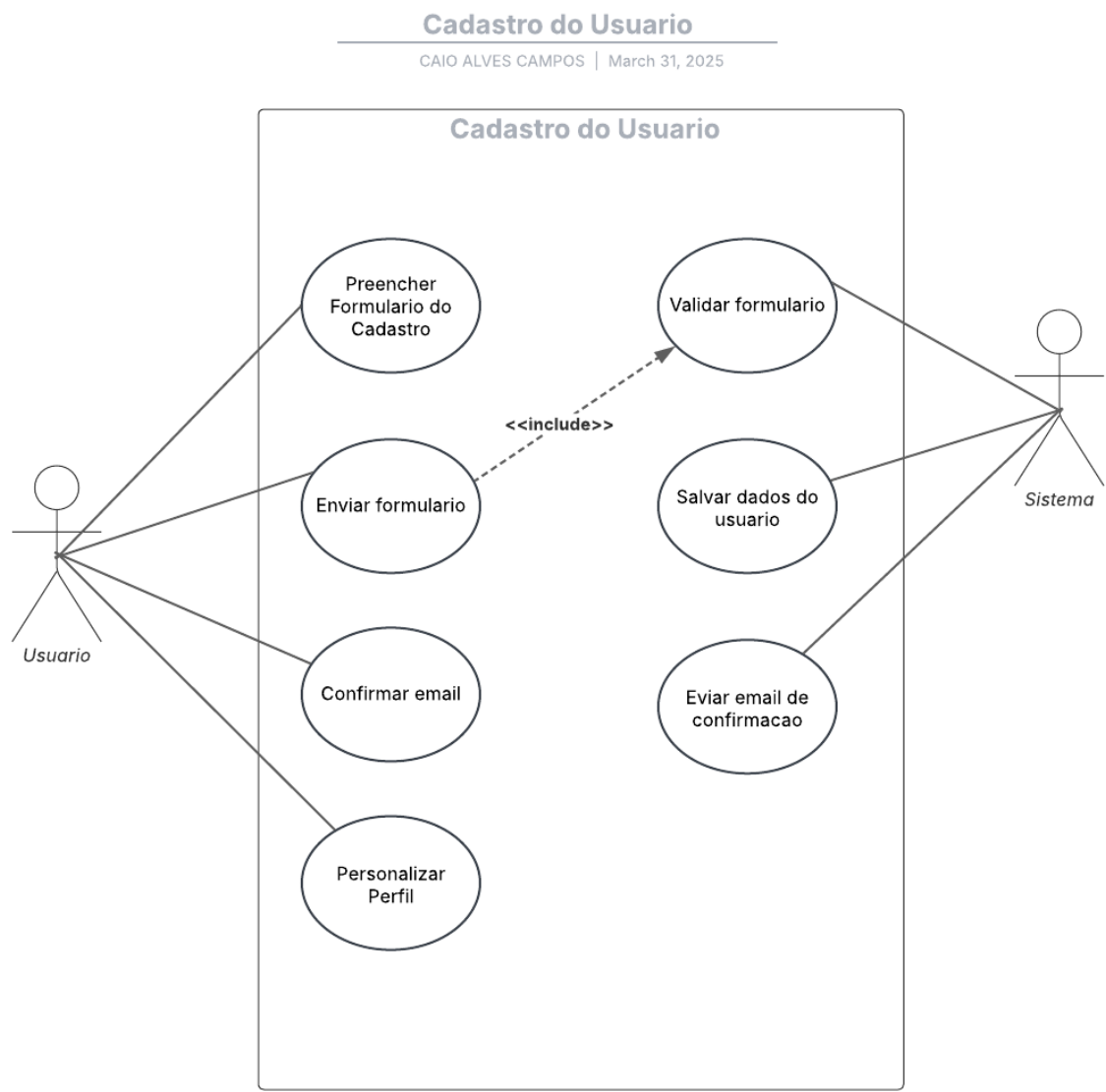
2.3.2 Atores envolvidos

Neste sistema, há apenas um ator que irá usar diretamente este sistema proposto:

- Usuário: Seria um usuário comum que deseja utilizar a plataforma.

2.4 CASOS DE USO ESPECÍFICOS

2.4.1 Cadastrar Usuários



UC 1 - Preencher Formulário do Cadastro

Descrição: Permite que o usuário preencha o formulário de cadastro com seus dados pessoais.

Prioridade: Essencial

Pré-condições: O sistema deve mostrar o formulário ao usuário.

Entrada: Recebe o formulário para cadastro.

Saída e pós-condições: O usuário envia o formulário ao sistema e recebe uma mensagem de conclusão.

UC 2 – Personalizar perfil

Descrição: Permite que o usuário personalize seu perfil de acordo com suas preferências.

Prioridade: Baixa

Pré-condições: O usuário deve estar devidamente cadastrado na plataforma.

Entrada: Recebe as personalizações feitas pelo usuário.

Saída e pós-condições: O sistema recebe as personalizações, salva no banco de dados e altera visualmente o perfil do usuário.

UC 3 – Validar Formulário

Descrição: O sistema valida os dados pessoais do usuário recebidos pelo envio do formulário.

Prioridade: Essencial

Pré-condições: O sistema deve receber os dados pessoais do usuário.

Entrada: Recebe como entrada os dados pessoais do usuário.

Saída e pós-condições: O sistema valida os dados, envia o email para confirmação e salva o cadastro no banco de dados.

UC 4 – Enviar email de confirmação

Descrição: Permite que o usuário receba um email de confirmação para melhor segurança.

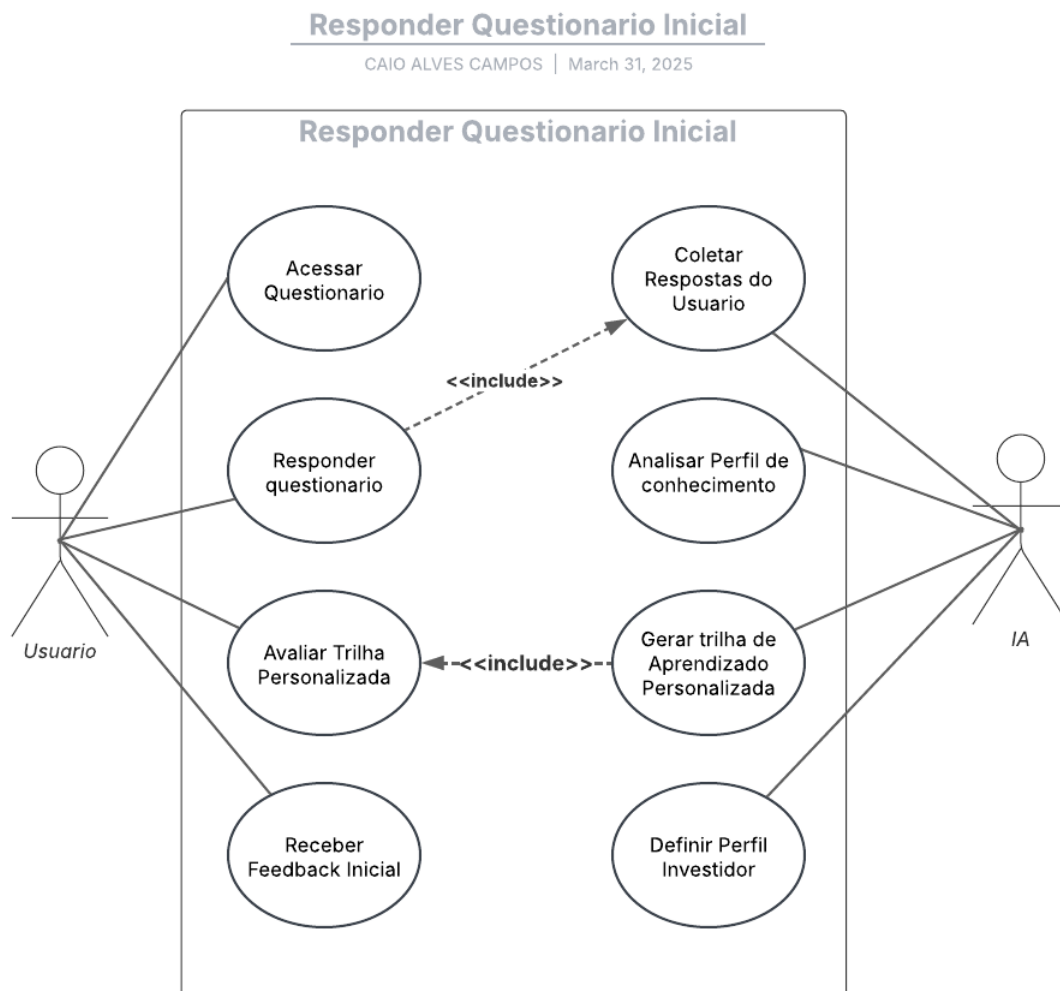
Prioridade: Média

Pré-condições: O usuário deve estar cadastrado no sistema.

Entrada: Recebe como entrada o email para confirmação.

Saída e pós-condições: O sistema recebe o email de confirmação e valida o cadastro do usuário.

2.4.2 Responder Questionário Inicial



UC 1 – Avaliar Trilha Personalizada

Descrição: Permite que o usuário receba uma trilha de aprendizado personalizada e avalie se o resultado foi bom ou não.

Prioridade: Média

Pré-condições: O usuário deve ter respondido o questionário inicial.

Entrada: Recebe a trilha personalizada como entrada.

Saída e pós-condições: Após a avaliação da trilha o usuário decide se quer utilizá-la ou não.

UC 2 – Analisar perfil de conhecimento

Descrição: Permite que a IA utilize os dados coletados para analisar o perfil de conhecimento do usuário.

Prioridade: Alta

Pré-condições: O usuário deve estar logado no sistema e deve responder ao questionário inicial.

Entrada: Recebe como entrada os dados do questionário.

Saída e pós-condições: A IA define qual o perfil de conhecimento do usuário e retorna ao sistema.

UC 3 – Gerar Trilha de Aprendizado Personalizada

Descrição: Permite que a IA gere uma trilha de aprendizado para cada usuário, baseado na análise do perfil de conhecimento.

Prioridade: Alta

Pré-condições: O usuário deve estar cadastrado no sistema e o perfil de conhecimento deve ter sido gerado.

Entrada: Recebe os parâmetros necessários para a geração da trilha.

Saída e pós-condições: A IA recebe os parâmetros necessários e gera uma trilha personalizada para a aprovação do usuário, caso seja recusada, a trilha de aprendizado padrão é utilizada.

UC 4 – Definir Perfil Investidor

Descrição: Permite que a IA defina qual o perfil investidor do usuário baseado nas respostas coletadas do questionário inicial.

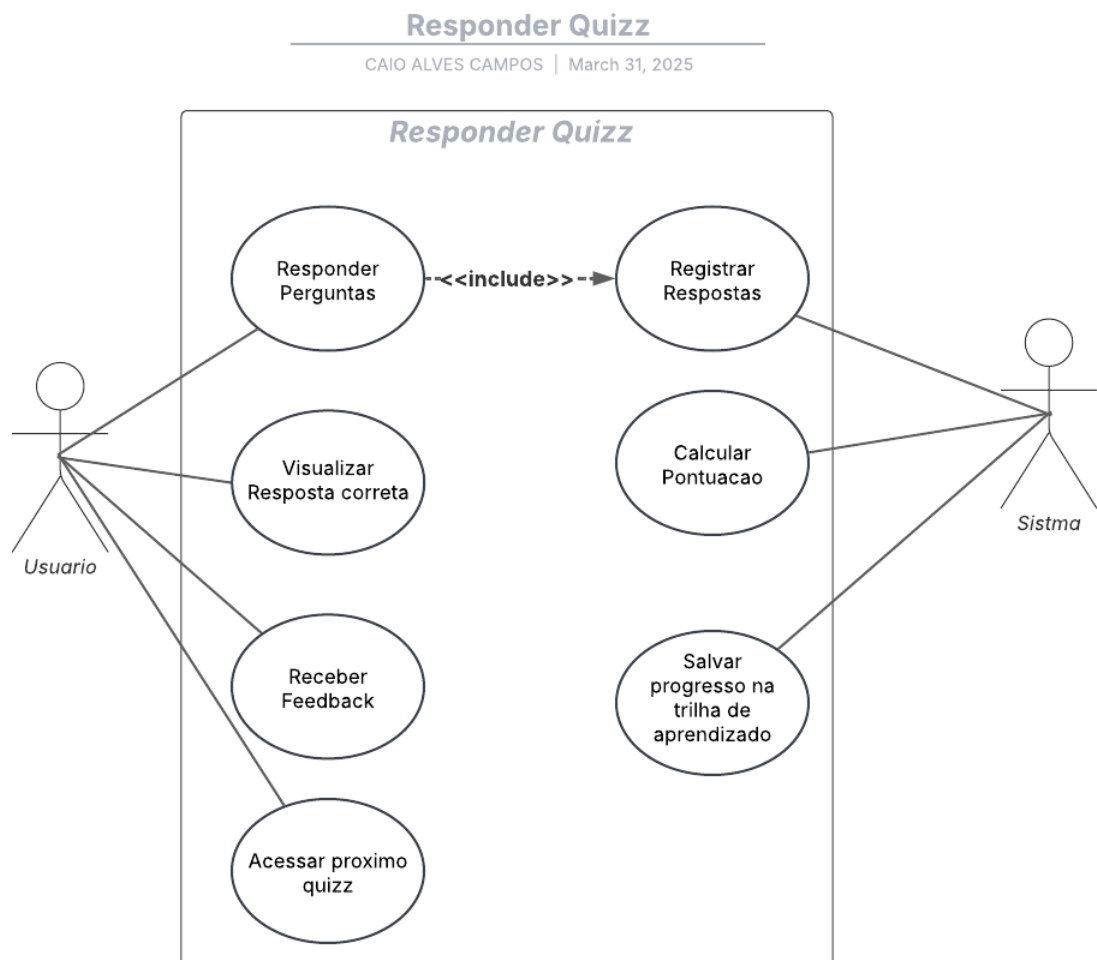
Prioridade: Alta

Pré-condições: O usuário deve estar cadastrado no sistema e deve ter completado o questionário inicial.

Entrada: Recebe como entrada os dados coletados do questionário inicial.

Saída e pós-condições: Após a análise dos dados coletados, da trilha de aprendizado gerada e do perfil de conhecimento do usuário, a IA define qual o perfil investidor e retorna ao sistema.

2.4.3 Responder Quizz



UC 1 – Responder perguntas

Descrição: Permite que o usuário responda as perguntas geradas pelo sistema.

Prioridade: Alta

Pré-condições: O usuário deve estar cadastrado no sistema e deve ter completado o questionário inicial.

Entrada: Nenhuma.

Saída e pós-condições: Após a resposta, o sistema coleta os registros das respostas.

UC 2 – Visualizar resposta correta

Descrição: Após responder à pergunta o usuário consegue acessar a resposta correta para um feedback.

Prioridade: Alta

Pré-condições: O usuário deve estar logado no sistema e deve responder à pergunta.

Entrada: Nenhuma.

Saída e pós-condições: Após o usuário responder a pergunta, o sistema retorna a resposta correta.

UC 3 – Receber feedback

Descrição: O usuário consegue acessar um feedback detalhado sobre a pontuação recebida, resposta correta, explicação da questão etc.

Prioridade: Média

Pré-condições: O usuário deve estar logado no sistema e deve ter respondido a pergunta correspondente.

Entrada: Nenhuma.

Saída e pós-condições: Após receber a requisição do usuário, o sistema retorna o feedback detalhado sobre a questão.

UC 5 – Definir Perfil Investidor

Descrição: Permite que o usuário acesse o próximo quizz da trilha de aprendizado.

Prioridade: Alta

Pré-condições: O usuário deve estar logado no sistema.

Entrada: Nenhuma.

Saída e pós-condições: O sistema verifica qual o ultimo quizz salvo na trilha de aprendizado do usuário e disponibiliza o próximo.

UC 6 – Registrar respostas

Descrição: O sistema coleta a resposta do usuário e registra no banco dados.

Prioridade: Alta

Pré-condições: O usuário deve estar logado no sistema e deve ter respondido a pergunta.

Entrada: Recebe como entrada as respostas do usuário.

Saída e pós-condições: O sistema salva as respostas do usuário para que a IA consiga analisar esses dados.

UC 7 – Calcular Pontuação

Descrição: Permite que o sistema calcule a pontuação gerada no quizz ou desafio.

Prioridade: Baixa

Pré-condições: O usuário deve estar logado e deve ter respondido à pergunta.

Entrada: Recebe a resposta do usuário e alguns outros dados para cálculo de pontuação.

Saída e pós-condições: Após calcular a pontuação, o sistema salva no banco de dados e contabiliza os pontos para os desafios ou conquistas selecionadas pelo usuário.

UC 8 – Salvar progresso na trilha de aprendizado

Descrição: Permite que o sistema salve o progresso completo do usuário após as responder quizzes no banco de dados.

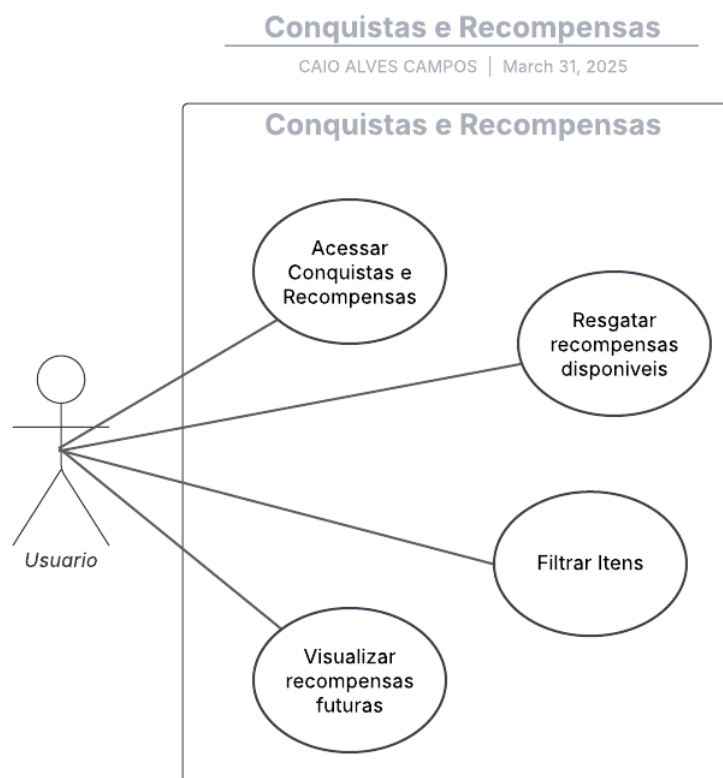
Prioridade: Alta

Pré-condições: O usuário deve estar logado e deve ter respondido o quizz.

Entrada: Recebe pontuação, resposta, feedback.

Saída e pós-condições: O sistema salva todas as informações de progresso no banco de dados.

2.4.4 Acessar Conquistas e Recompensas



UC 1 – Acessar conquistas e recompensas

Descrição: Permite que o usuário consiga acessar suas conquistas e recompensas.

Prioridade: Baixa

Pré-condições: O usuário deve estar logado no sistema.

Entrada: Nenhuma.

Saída e pós-condições: Após validar se o usuário está logado no sistema, ele busca os dados salvos de conquistas e recompensas.

UC 2– Resgatar recompensas disponíveis

Descrição: Permite que o usuário resgate as recompensas disponíveis.

Prioridade: Baixa

Pré-condições: O usuário deve estar logado no sistema e deve ter cumprido os requisitos da recompensa.

Entrada: Recebe as recompensas disponíveis.

Saída e pós-condições: Após o usuário selecionar quais recompensas deseja resgatar, o sistema salva no banco de dados.

UC 3 – Filtrar itens

Descrição: Permite que o usuário utilize filtros para melhorar a experiência de itens específicos.

Prioridade: Baixa

Pré-condições: O usuário deve estar logado no sistema.

Entrada: Recebe como entrada os filtros selecionados pelo usuário.

Saída e pós-condições: Após receber os filtros selecionados, o sistema busca os itens que passam no filtro e mostra ao usuário.

UC 4 – Visualizar recompensas futuras

Descrição: Permite que o usuário veja quais as recompensas ele pode conquistar.

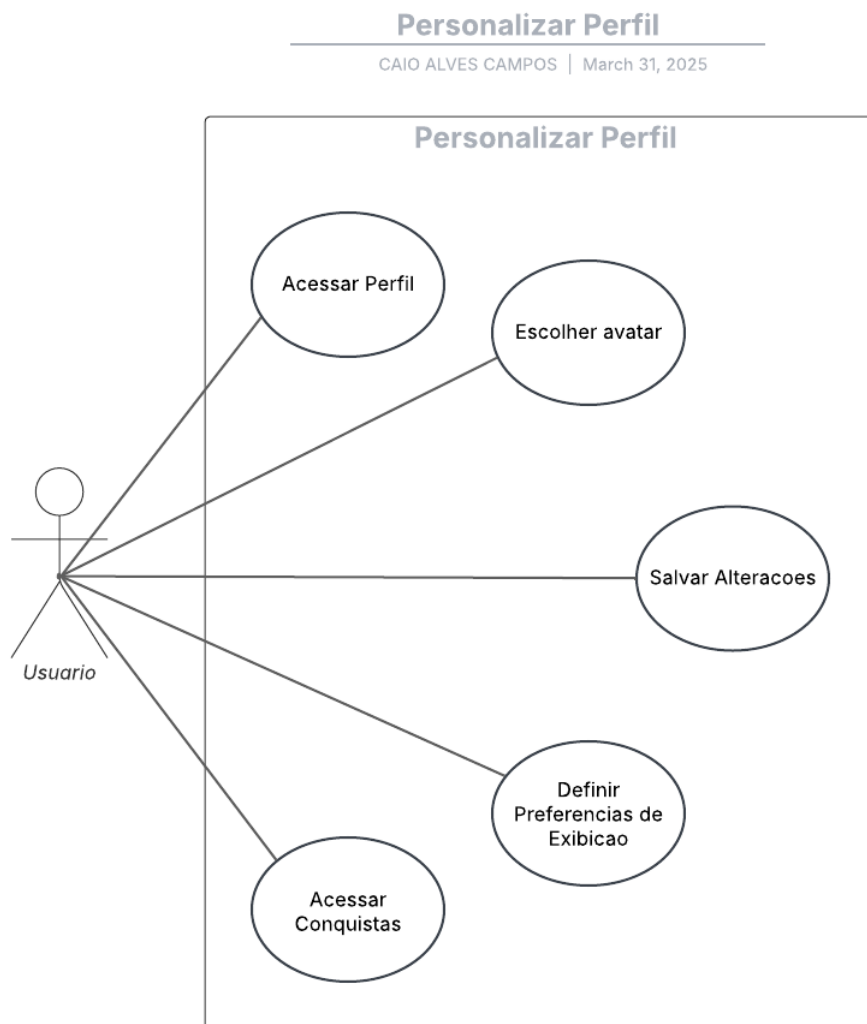
Prioridade: Baixa

Pré-condições: O usuário deve estar logado no sistema.

Entrada: Recebe como todos as recompensas que o usuário irá receber.

Saída e pós-condições: Após o usuário acessar a página de recompensas, ele consegue visualizar todas as recompensas que já possui progresso.

2.4.5 Personalizar Perfil



UC 1 – Acessar Perfil

Descrição: Permite que o usuário acesse seu perfil.

Prioridade: Baixa

Pré-condições: O usuário deve estar logado no sistema.

Entrada: Nenhuma.

Saída e pós-condições: O usuário consegue acessar seu perfil para fazer alterações ou somente visualizar.

UC 2 – Escolher avatar

Descrição: Permite que o usuário consiga escolher um avatar de sua preferência.

Prioridade: Baixa

Pré-condições: O usuário deve estar logado no sistema.

Entrada: Nenhuma.

Saída e pós-condições: Após a escolha do avatar pelo usuário, o sistema salva o avatar escolhido e ativa sua visualização.

UC 3 – Salvar as alterações

Descrição: Permite que o usuário salve as alterações que foram feitas no seu perfil.

Prioridade: Baixa

Pré-condições: O usuário deve estar logado no sistema.

Entrada: Alterações no perfil.

Saída e pós-condições: Após o usuário escolher as alterações que deseja, o sistema salva no banco de dados.

UC 4 – Definir preferencias de exibição

Descrição: Permite que o usuário selecione quais as preferencias de exibição.

Prioridade: Baixa

Pré-condições: O usuário deve estar logado no sistema.

Entrada: Recebe as preferencias de exibição.

Saída e pós-condições: O sistema salva as preferencias de exibição do usuário.

UC 5 – Acessar conquistas

Descrição: Permite que o usuário consiga acessar suas conquistas e recompensas.

Prioridade: Baixa

Pré-condições: O usuário deve estar logado no sistema.

Entrada: Nenhuma.

Saída e pós-condições: Após validar se o usuário está logado no sistema, ele busca os dados salvos de conquistas e recompensas.

2.5 ARQUITETURA DO SISTEMA

O sistema CryptoQuest foi projetado com base nos princípios da Clean Architecture, com o objetivo de garantir uma separação clara de responsabilidades entre as camadas da aplicação, facilitar a manutenção do código, promover a escalabilidade e permitir a reutilização de componentes ao longo do desenvolvimento.

A arquitetura está dividida em três principais camadas:

a) Camada de Apresentação (Frontend)

Desenvolvida com a tecnologia Flutter, essa camada é responsável por toda a interface gráfica do sistema e pela interação com o usuário final. Inclui as telas de login, cadastro, quizzes, trilha de aprendizado, conquistas, entre outras.

As principais responsabilidades da camada de apresentação são:

- Exibir as interfaces do sistema de forma responsiva, compatível com dispositivos Android e iOS;
- Realizar a navegação entre as telas;
- Validar dados localmente antes do envio ao backend;
- Realizar chamadas à API REST do sistema para comunicação com o servidor;
- Persistir dados básicos localmente (preferências e tokens de autenticação).

b) Camada de Domínio (Regras de Negócio)

A camada de domínio representa a lógica central do sistema. Essa camada está parcialmente implementada tanto no frontend quanto no backend, e engloba regras relacionadas à pontuação, progressão de trilhas, liberação de conquistas e classificação do nível de conhecimento do usuário.

As principais responsabilidades dessa camada incluem:

- Processar regras de negócios aplicadas à evolução do usuário;
- Controlar o sistema de pontuação, rankings e recompensas;
- Orquestrar a comunicação entre os dados recebidos do frontend e os modelos de IA utilizados para personalização do conteúdo.

c) Camada de Dados (Backend + IA)

O backend foi desenvolvido utilizando o framework FastAPI, que possibilita a construção de uma API RESTful leve, segura e de alto desempenho. Essa camada centraliza toda a lógica de persistência, autenticação e integração com os módulos de inteligência artificial.

O armazenamento de dados é realizado no Firebase Firestore, um banco de dados NoSQL em tempo real, otimizado para aplicações móveis. A autenticação é feita por meio do Firebase Auth, com suporte a OAuth, permitindo login via Google, Facebook, entre outros.

Para a camada de inteligência artificial, foi utilizada a biblioteca Scikit-learn, em conjunto com Pandas, para analisar o desempenho do usuário, classificá-lo por nível de conhecimento e recomendar conteúdo personalizado de acordo com suas respostas e histórico.

As principais responsabilidades da camada de dados são:

- Gerenciar os dados dos usuários, progresso, quizzes e trilhas;
- Realizar autenticação e controle de sessões com tokens JWT;
- Integrar a IA ao backend para análise e recomendação de conteúdo;
- Armazenar e recuperar informações no Firestore com segurança;
- Aplicar criptografia a dados sensíveis (AES-256 e SHA-256).

d) Comunicação entre os componentes

A comunicação entre o aplicativo (frontend) e o backend é realizada por meio de requisições HTTP (REST), utilizando endpoints definidos no FastAPI. As respostas incluem dados processados ou resultados da IA.

A autenticação é gerenciada diretamente pelo Firebase Auth, e os tokens de sessão são utilizados para proteger as requisições à API. O backend, por sua vez, manipula os dados no Firestore e invoca a IA sempre que necessário.

e) Considerações sobre segurança

A segurança dos dados do usuário é uma prioridade na arquitetura do sistema. As principais medidas adotadas incluem:

- Utilização de HTTPS em todas as comunicações entre o cliente e o servidor;
- Criptografia dos dados sensíveis utilizando AES-256 (para informações pessoais) e SHA-256 (para senhas);
- Autenticação segura via Firebase Auth, com tokens JWT válidos;
- Conformidade com legislações de proteção de dados como a LGPD (Lei Geral de Proteção de Dados) e o GDPR (General Data Protection Regulation).

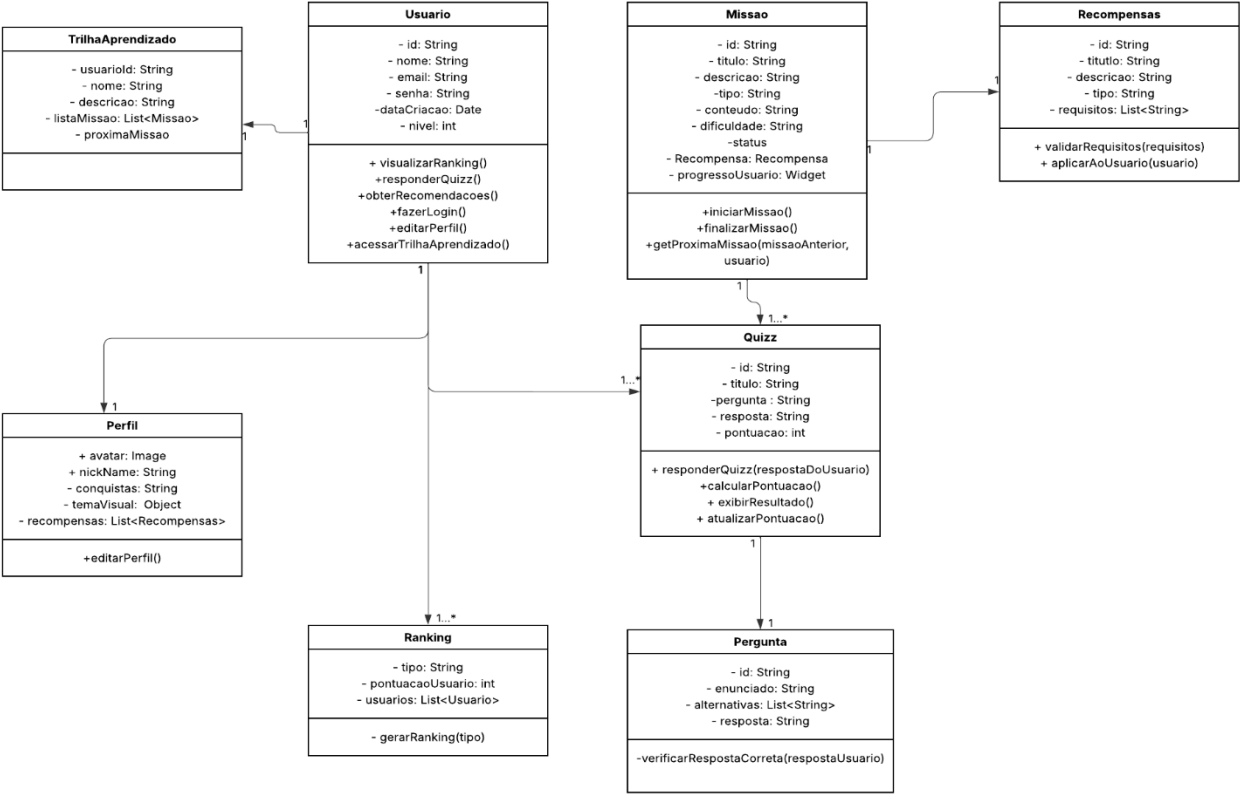
f) Justificativa da arquitetura

A adoção dessa arquitetura se justifica pelos seguintes fatores:

- Modularidade e separação clara de responsabilidades;
- Facilidade de manutenção e evolução do sistema;
- Escalabilidade da aplicação com uso de serviços em nuvem (Firebase, Google Cloud);
- Segurança robusta na autenticação e armazenamento de dados;
- Flexibilidade para testes automatizados e integração contínua;
- Suporte à personalização da experiência do usuário com uso de IA.

2.6 DIAGRAMAS

2.6.1 Modelo de Classes



3 DESENVOLVIMENTO DE SOFTWARE

3.1 METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE

Com os requisitos levantados e validados pelo Consultor do Sistema e com os Orientadores, o projeto entra na fase de desenvolvimento, utilizando como base a metodologia ágil Scrum, visando entregas contínuas e evolutivas do sistema. O desenvolvimento será dividido em sprints de duração pré-definida cada um com suas devidas metas e entrega de funcionalidade. A cada ciclo um conjunto de requisitos será selecionado e desenvolvido e avaliado.

A escolha do Scrum se deu pela abordagem iterável e flexível, que favorece projetos com mudanças e permite que o desenvolvedor exerça múltiplos papéis. O controle de versão do projeto será feito utilizando GitHub e o gerenciamento de tarefas será realizado pelo Trello para facilitar a visualização do progresso.

3.1.1 Ambientes de desenvolvimento/produção

No ambiente de desenvolvimento, as ferramentas serão configuradas para permitir testes locais e emulados. Será utilizado o Visual Studio Code com extensões Flutter, Python e Firebase. Para o Backend, serão utilizados ambientes virtuais, execução local de Endpoints e logs e mensagens de debug serão habilitados neste ambiente para facilitar a detecção e resolução de erros.

Já o ambiente de produção será voltado para a disponibilidade do sistema aos usuários finais. O frontend estará hospedado em Firebase Hosting com uma build otimizada e o Backend estará hospedado na Google Cloud para maior flexibilidade. As configurações de segurança serão forçadas nesse ambiente, com autenticação real via Firebase Auth. Essas escolhas foram feitas pela facilidade, baixo custo inicial e infraestrutura baseada em nuvem para que não seja necessário a configurar e manter servidores físicos

3.1.2 Bibliotecas principais

Frontend(Flutter):

- Material
- Provider
- Firebase_auth
- Cloud_firestore
- Flutter_secure_storage
- http

Backend(Python)

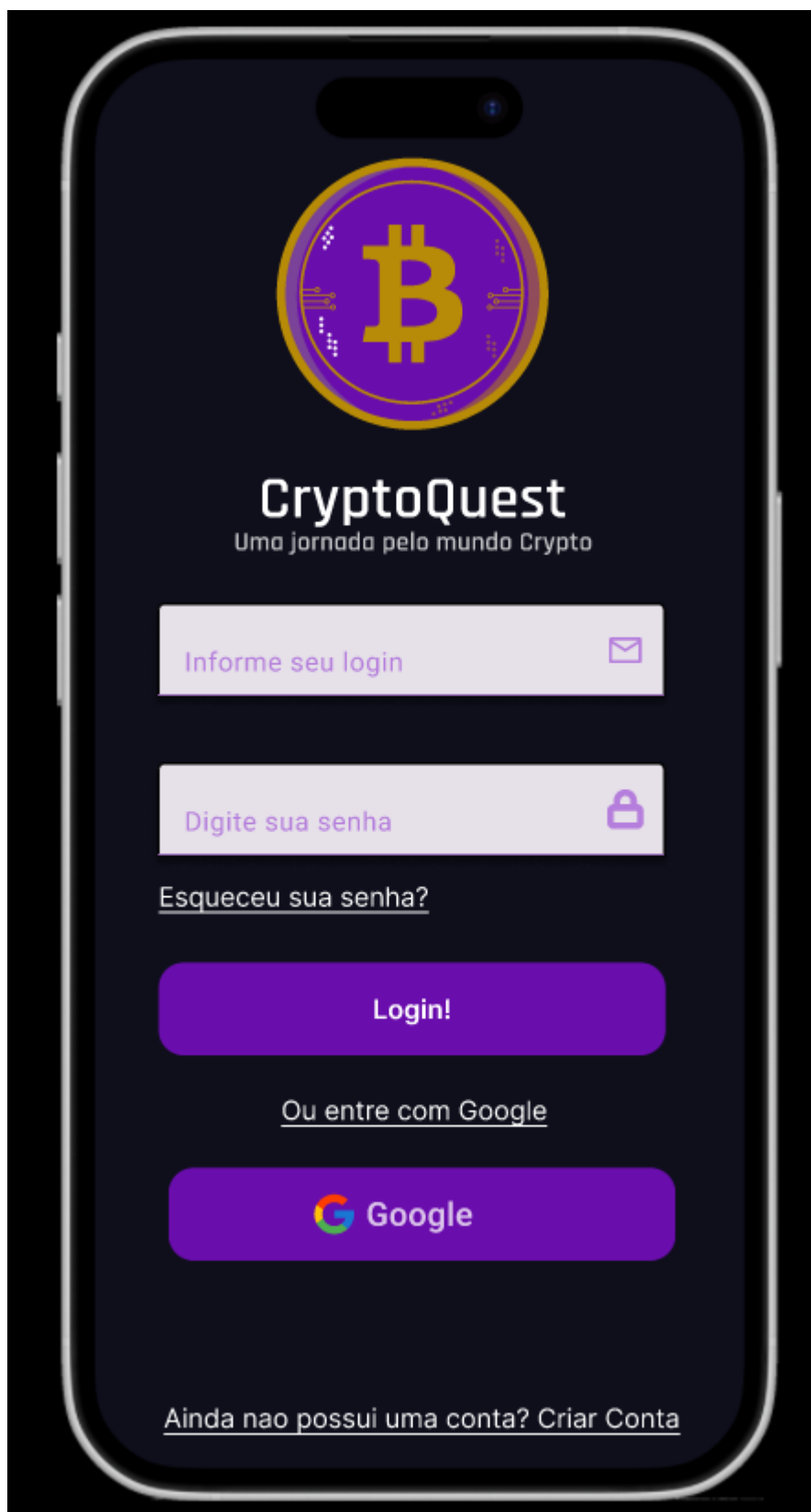
- FastApi
- Firebase_admin
- Pandas
- Scikit
- pytest

3.2 MÓDULOS DO CRONOGRAMA DE DESENVOLVIMENTO

- 1 – Cadastro e autenticação
 - Cadastro / Login de usuário
 - Criação de perfil
- 2 – Trilha e Missões
 - Acesso a trilha
 - Execução de missão
- 3 – Quis e IA
 - Resolução de Quis
 - Recomendação da IA
- 4 – Recompensas e Ranking
 - Sistema de Recompensas
 - Ranking de Usuários
- 5 – Integração final e testes
 - Integração dos módulos
 - Testes finais e ajustes

3.3 MOCKUPS

3.3.1 – Tela de Login



3.3.2 – Tela de Cadastro de Usuário

A mobile application registration screen with a dark blue background. At the top, there is a white user icon inside a circle. Below it, the word "Cadastro" is written in large white font. Underneath, the text "Informe seus dados para fazer Login" is displayed in a smaller white font. The screen features three white input fields with rounded corners. The first field is labeled "Informe seu login" and has a white envelope icon on the right. The second field is labeled "Digite sua senha" and has a white padlock icon on the right. The third field is labeled "Confirme sua senha" and also has a white padlock icon on the right. Below these fields is a large, rounded blue button with the text "Cadastrar!" in white. Underneath the button, the text "Ou cadastre-se com Google" is displayed in white. Below this text is another rounded blue button featuring the Google logo and the word "Google" in white. At the bottom of the screen, the text "Ja possui uma conta? Fazer login" is displayed in white, with "Fazer login" underlined.



Cadastro

Informe seus dados para fazer Login

Informe seu login 

Digite sua senha 

Confirme sua senha 

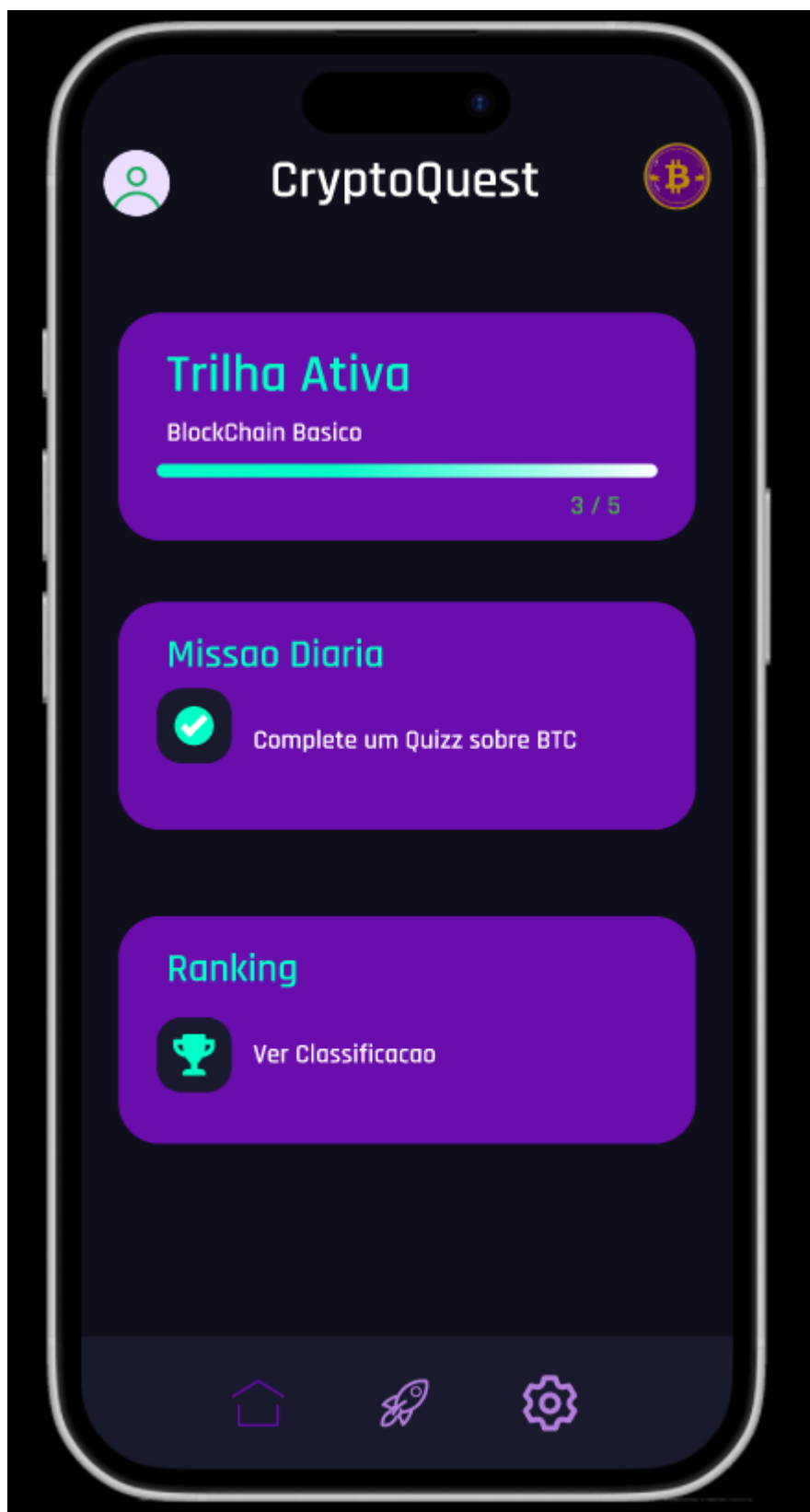
Cadastrar!

[Ou cadastre-se com Google](#)

 **Google**

[Ja possui uma conta? Fazer login](#)

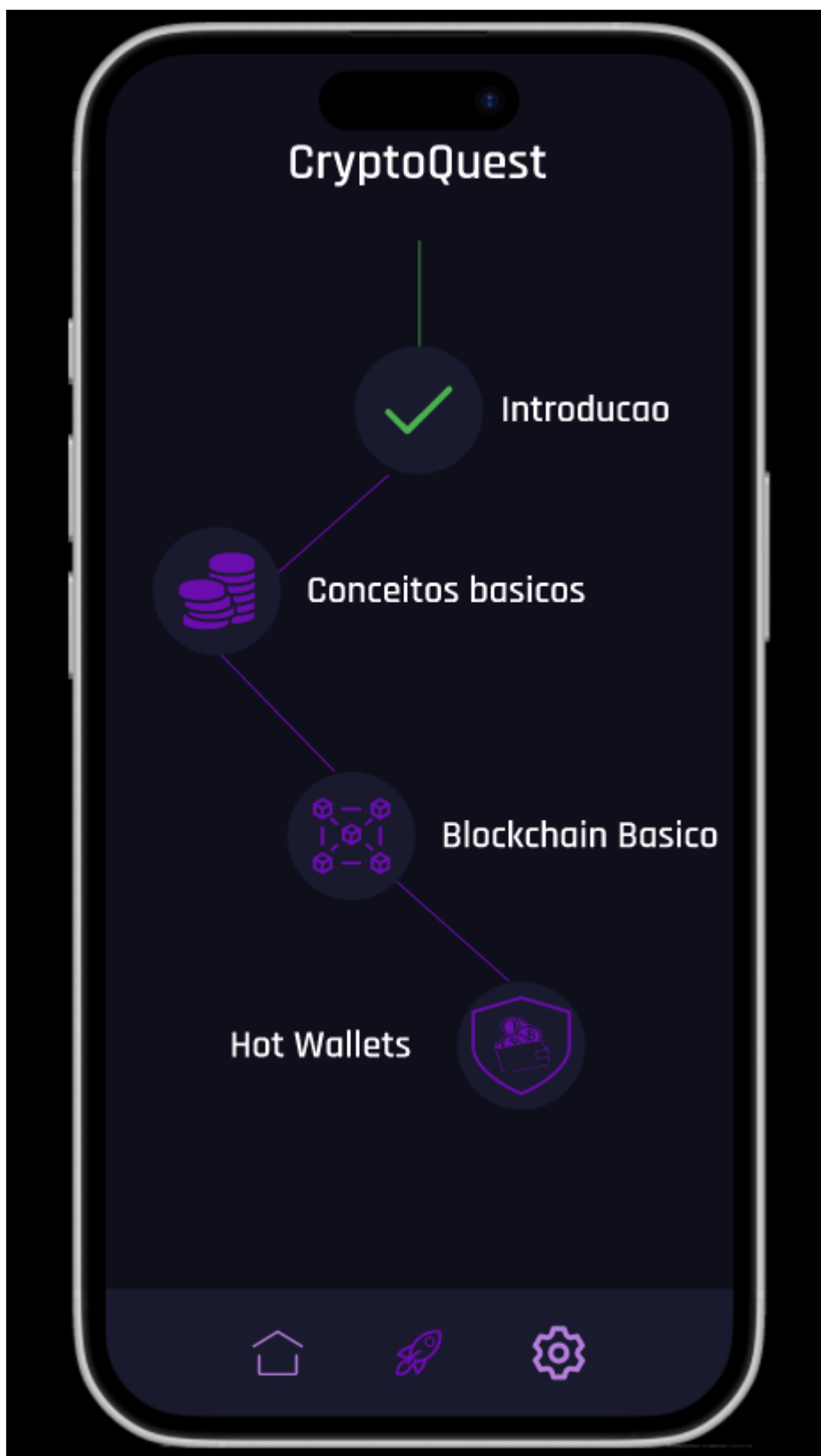
3.3.3 – Tela Principal



3.3.4 – Tela para responder quizz



3.3.5 – Tela da Trilha de Aprendizado



4 CONCLUSÃO

Nessa conclusão, deve-se conter:

- Uma breve descrição a respeito de todo o projeto e desenvolvimento.
- Dificuldades encontradas
- Trabalhos Futuros

5 REFERÊNCIAS

TECHTUDO. **Como funciona um site de Sorteios.** Disponível em: <<https://www.techtudo.com.br/noticias/2020/05/como-funciona-o-sorteiogram-conheca-site-para-sorteios-no-instagram.ghml/>> Acesso em: 10 jun. 2021.

WEBMUSEUM. **A história do primeiro site publicado.** Disponível em: <<https://museuweg.net/blog/conheca-a-historia-do-primeiro-site-publicado/>> Acesso em: 10 jun. 2021.



Colocar em ordem alfabética

ANEXO

MANUAL DO USUÁRIO

(NÃO PRECISA COLOCAR CAPA E FOLHA DE ROSTO DA UNIGRAN)