

Documentação do Projeto

SISTEMA PORTAL DE CRIPTO ATIVOS

Discentes:

ALEXSANDRO NOGUEIRA COSTA FILHO

ANTONIO NETO DE MOURA MELO

CARLA SOFIA SANTOS RIBEIRO

EMERSON PAULO PINHEIRO MUNIZ

MARCOS DAVI TAVEIRA DE SOUSA

Docente Responsável:

THALES LEVI AZEVEDO VALENTE

São Luís - MA

2024



1. INTRODUÇÃO	3
2. OBJETIVOS	3
3. LEVANTAMENTO DE REQUISITOS	4
3.1. O que são requisitos?	4
3.2. Requisitos funcionais	4
3.3. Requisitos não funcionais	6
4. DIAGRAMAS DE CASOS DE USO	7
4.1. Cadastrar Usuário	8
4.2. Efetuar login	9
4.3. Gerenciar Perfil	11
4.4. Gerenciamento de Carteira	13
4.5. Visualizar notícias	15
4.6. Gerenciar criptomoedas	17
4.7. Gerenciador de idiomas	18
5. DIAGRAMA DE CLASSES	19
6. DIAGRAMAS DE SEQUÊNCIA	20
6.1. Login	21
6.2. Registro usuário	22
6.3. Registro alerta	23
6.4. Registro exchange	24
6.5. Registro criptomoeda	25
6.6. Atualizar usuário	26
6.7. Atualizar alerta	27
6.8. Atualizar quadro	27
6.9. Adicionar quadro	28
6.10. Apagar usuário	29
6.11. Apagar quadro	30
6.12. Apagar alerta	30
6.13. Apagar criptomoeda	31
6.14. Apagar exchange	32
6.15. Recuperar senha	32
6.16. Visualizar notícia	33
7. DIAGRAMAS DE ATIVIDADES	34
8. DIAGRAMAS DE ESTADO	41
9. TECNOLOGIAS	52
9. ESCOPO DA PROTOTIPAÇÃO	52
10. CONCLUSÃO	58



1. INTRODUÇÃO

As criptomoedas revolucionaram o mercado financeiro ao introduzirem um sistema descentralizado e digital de transações, que oferece agilidade, transparência e independência de intermediários tradicionais. Nos últimos anos, o aumento da popularidade das criptomoedas atraiu investidores, empresas e governos, consolidando esse mercado como um dos pilares da economia digital global. Contudo, a volatilidade dos preços, a complexidade das análises técnicas e a necessidade de informações confiáveis tornam essencial a criação de ferramentas que facilitem o gerenciamento de ativos digitais e a tomada de decisões.

Este documento apresenta o levantamento de requisitos e os diagramas de projeto para o desenvolvimento de um sistema inovador e intuitivo, projetado para atender às demandas de investidores em criptomoedas, desde iniciantes até profissionais experientes. O sistema busca integrar diversas funcionalidades que permitam o monitoramento de portfólios, análises de mercado e personalização da experiência do usuário, sempre com foco em segurança, usabilidade e eficiência.

2. OBJETIVOS

O objetivo principal deste sistema é oferecer uma plataforma completa e acessível para o gerenciamento e análise de ativos em criptomoedas, promovendo a inclusão e educação financeira no mercado de cripto ativos. A plataforma será projetada para:

- 2.1. Facilitar o acesso a dados relevantes:** Fornecendo informações atualizadas sobre preços, volumes de mercado, capitalização, transações e outros indicadores técnicos, de forma clara e interativa.



- 2.2. Promover a tomada de decisões informadas:** Integrando análises avançadas, comparações entre criptomoedas, e notificações personalizadas para ajudar o usuário a identificar oportunidades e riscos.
- 2.3. Garantir segurança e conformidade:** Implementando mecanismos robustos de proteção de dados, autenticação e criptografia, em alinhamento com as regulações internacionais como LGPD e GDPR.
- 2.4. Oferecer uma experiência personalizada:** Permitindo que cada usuário customize seu ambiente de navegação, dashboards e alertas, adequando a plataforma às suas preferências e objetivos.

Em um cenário onde as criptomoedas já representam um importante segmento da economia e estão em constante evolução, este sistema se propõe a ser uma ferramenta essencial para democratizar o acesso a informações e facilitar a interação de investidores com o universo dos ativos digitais.

3. LEVANTAMENTO DE REQUISITOS

3.1. O que são requisitos?

Requisitos são descrições detalhadas das funções e características que um sistema precisa ter para cumprir as necessidades dos usuários e atingir os objetivos do negócio. Existem dois tipos principais: requisitos funcionais, que especificam as ações que o sistema deve realizar, e requisitos não funcionais, que definem atributos como performance, segurança e usabilidade.

3.2. Requisitos funcionais

- **RF1. Sistema de login e cadastro de novos usuários**
 - o cliente deve ser capaz de acessar sua conta para verificar seus ativos personalizados
 - cadastrar um novo usuário
 - atualizar seus dados de cadastro



- apagar a conta
 - visualizar seus dados

 - **RF2. Gerenciamento de Carteira**
 - Visualizar preços atuais e históricos de investimentos
 - Volume de mercado e capitalização
 - Transações
 - Endereços ativos

 - **RF3. Gerenciador de cripto**
 - O sistema deve permitir selecionar a criptomoeda e visualizar os dados estatísticos da mesma em tempo real
 - Visualizar o histórico da criptomoeda
 - Incluir indicadores técnicos

 - **RF5. Visualizar notícias**
 - Deverá ser integrado feeds de notícias de fontes confiáveis (CoinDesk, CoinTelegraph)
 - Mostrar dicas de investimentos e tendências de mercado

 - **RF7. Comparar criptomoedas**
 - O sistema vai permitir selecionar duas criptomoedas e compará-las através dados técnicos

 - **RF8. Gerenciamento de Alertas**
 - Implementar alertas em uma página para preços e tendências
 - Permitir que o usuário configure alertas por criptomoeda
-



- **RF9. Gerenciamento de Quadros de informações de cripto**

- Permitir que usuários customizem seus dashboards, salvem preferências e histórico de atividade.
- Suporte para criação de múltiplos dashboards personalizados por usuário.

- **RF10. Alterar idioma**

- Implementar suporte para inglês e português no lançamento e um sistema que permita a fácil expansão para outros idiomas no futuro.
- Adaptar interface para diferentes moedas locais.

3.3. Requisitos não funcionais

- **RNF1. Segurança e proteção dos dados**

- Implementar criptografia SSL para todas as comunicações
- Implementar autenticação de dois fatores para todas as contas de usuário.
- Assegurar conformidade com LGPD e GDPR, incluindo políticas claras de privacidade e consentimento de usuários.
- Realizar auditorias de segurança trimestrais para identificar e corrigir vulnerabilidades.
- As senhas salvas dos usuários devem ser criptografadas antes de serem armazenadas no banco de dados.

- **RNF2. Desempenho e Escalabilidade**

- O sistema tem que garantir o suporte para pelo menos 10 mil usuários simultâneos com possibilidade de expansão para 100 mil usuários.



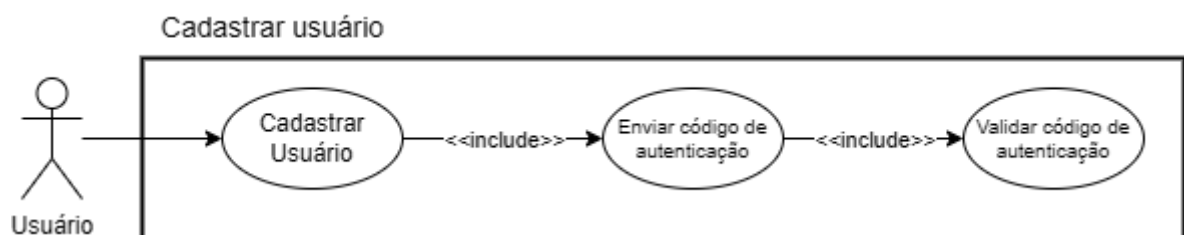
- Implementar cache para atualização de dados em tempo real com latência de até no máximo 1 segundo.
- Implementar monitoramento contínuo do sistema para detectar e responder rapidamente a falhas e problemas de desempenho.
- **RNF3. Usabilidade e experiência do usuário**
 - Implementar interfaces intuitivas para diferentes níveis de conhecimento.
- **RNF4. Confiabilidade**
 - Fonte de dados confiáveis e estáveis.
- **RNF5. Responsivo**
 - O sistema deve ser responsivo em diferentes dispositivos, como computadores, tablets e smartphones, garantindo uma experiência de usuário consistente
- **RNF6. Manutenibilidade**
 - Capacidade de realizar manutenções sem impactar a disponibilidade e o desempenho do sistema.
- **RNF7. Suporte a múltiplos idiomas**
 - Capacidade de mudar a página para mais de 2 idiomas

4. DIAGRAMAS DE CASOS DE USO

Casos de uso (ou use cases) são ferramentas utilizadas principalmente na engenharia de software para descrever como um sistema ou aplicação interage com seus usuários (também chamados de "atores") para atingir um objetivo específico. Eles representam as funcionalidades do sistema de uma forma simples e acessível, detalhando os passos necessários para alcançar um resultado desejado.

Essencialmente, um caso de uso é um cenário que descreve uma interação típica entre um usuário e o sistema, considerando tanto os passos bem-sucedidos quanto às exceções que podem ocorrer.

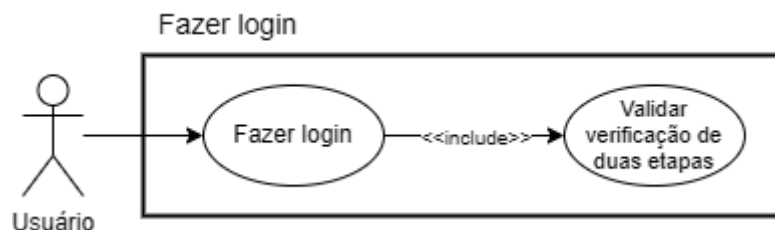
4.1. Cadastrar Usuário



- **Ator Principal** - Usuário
- **Objetivo** - Permitir que o usuário cadastre-se no sistema
- **Fluxo principal**
 - O usuário acessa a página de cadastramento de usuário
 - O sistema exibe os campos para preenchimento do nome de usuário, a senha, telefone e email
 - O usuário preenche os dados
 - O usuário solicita o cadastramento no sistema.
 - O sistema pede um código de autenticação enviado para o usuário
 - O usuário insere o código de autenticação

- O usuário solicita o cadastramento no sistema
- O sistema verifica se o nome de usuário já está registrado e se o código de autenticação é válido:
 - Se o nome de usuário não estiver registrado e o código de autenticação for válido, o sistema salva o novo usuário no banco de dados.
- O sistema direciona o usuário à tela de login.
- **Fluxo Alternativo**
 - Usuário Já Cadastrado ou código de autenticação inválido:
 - Caso o nome de usuário inserido já esteja cadastrado no sistema ou o código de autenticação seja inválido, o sistema exibe uma mensagem de erro e retorna para o segundo passo
- **Requisitos Especiais**
 - Os dados de acesso devem ser armazenados de forma segura em um banco de dados utilizando criptografia.
 - O nome de usuário deve ser único no sistema.
- **Pré-condições**
 - O nome de usuário a ser cadastrado não deve existir no sistema.
- **Pós-condições**
 - O novo usuário está registrado no sistema e pode realizar login.

4.2. Efetuar login





- **Ator Principal** - Usuário
- **Objetivo** - Permitir que o usuário acesse o sistema por meio de autenticação de credenciais.
- **Fluxo principal**
 - O usuário acessa a página de login.
 - O sistema exibe os campos para inserção do nome de usuário e da senha
 - O usuário insere o nome de usuário e a senha.
 - O sistema verifica as credenciais:
 - Se as credenciais estiverem corretas, o sistema permite solicitar código de autenticação no sistema.
 - O usuário solicita autenticação no sistema
 - O sistema solicita código enviado para o usuário
 - O usuário insere o código de autenticação
 - O usuário solicita a autenticação no sistema
 - O sistema direciona o usuário para a página principal.
- **Fluxo Alternativo**
 - Credenciais Inválidas:
 - Caso as credenciais inseridas sejam inválidas, o sistema exibe uma mensagem de erro e retorna ao segundo passo
 - Recuperar Senha:
 - Se o usuário esquecer a senha de acesso, o sistema oferece a opção de recuperação de senha na página de login.
 - O sistema manda um código de verificação para o usuário
 - O usuário insere o código de verificação
 - O sistema mostra uma tela para o usuário digitar a nova senha e confirmar a mesma

- O usuário digita a senha
- O usuário trocar a senha

- **Requisitos Especiais**

- O sistema deve operar sob conexão segura (HTTPS).
- As credenciais devem atender aos padrões de complexidade definidos (mínimo de 8 caracteres, incluindo letras, números e símbolos).

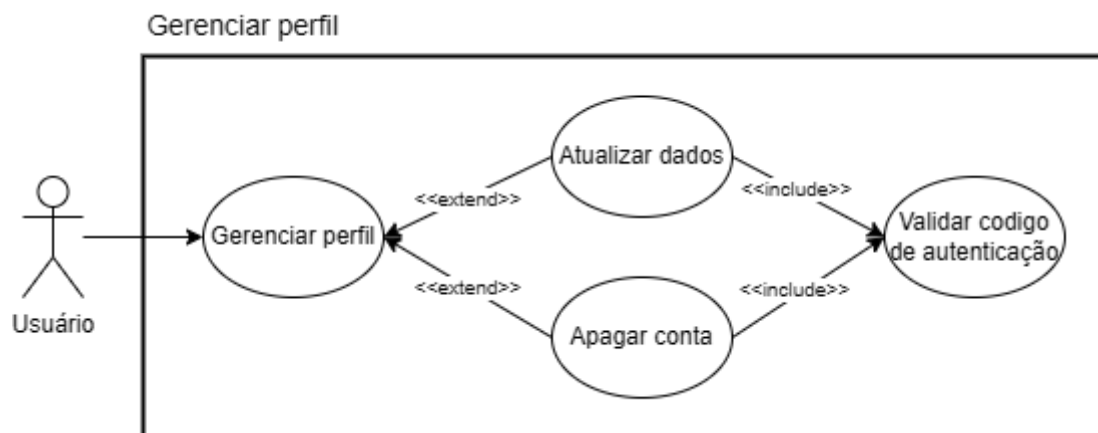
- **Pré-condições**

- O sistema deve estar operacional.
- O usuário deve possuir credenciais cadastradas no sistema.

- **Pós-condições**

- O usuário está autenticado no sistema.

4.3. Gerenciar Perfil



- **Ator Principal** - Usuário
- **Objetivo** - Permite que o usuário gerencie seu perfil , e lá ele consegue atualizar dados e apagar conta.
- **Fluxo principal**
 - O usuário acessa a funcionalidade de gerenciar seu perfil
 - O sistema exibe duas opções: uma de atualizar dados e outra de apagar conta

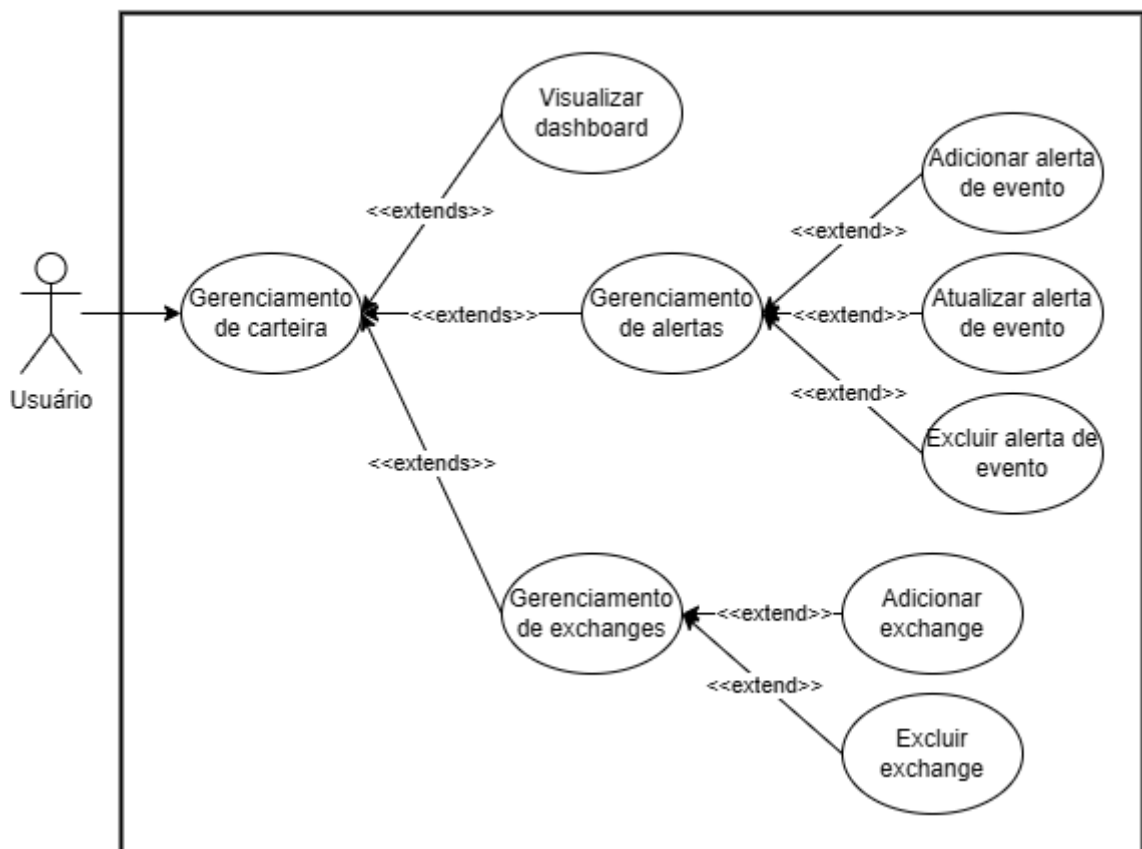


- O usuário seleciona de atualizar dados
 - O sistema exibe uma caixinha para cada opção a seguir - nome de usuário, senha, telefone e email
 - O usuário seleciona o que deseja editar e faz sua atualização
 - Se o que está sendo atualizado é a senha
 - O sistema manda um código de autenticação para o usuário
 - O usuário insere o código de autenticação
 - O usuário solicita a atualização da senha
 - Se o código de autenticação for correto
 - Segue para o próximo passo
 - O usuário seleciona a opção salvar e aparece a opção de encerrar a operação
 - **Fluxo Alternativo**
 - Código de autenticação incorreto
 - O sistema retorna para o passo 2
 - Apagar conta
 - No passo 3 o usuário seleciona apagar conta
 - O sistema exibe uma caixinha com a mensagem “você tem certeza que deseja apagar a sua conta ?” e uma opção “sim” e outra “não”
 - Se o usuário seleciona “sim” , é solicitado o código 2FA , se o código estiver correto, sua conta é desativada permanentemente e o software vai para a tela de cadastro
 - Se o usuário seleciona “nao” sua conta continua ativa e ele é mandado para a tela inicial do sistema
 - **Pré-condições**
 - O usuário está logado
 - **Pós-condições**
-

- O usuário consegue atualizar seu perfil ou ele apaga sua conta permanentemente.

4.4. Gerenciamento de Carteira

Gerenciamento de Carteira



- **Ator Principal** - Usuário
- **Objetivo** - Permitir que o usuário visualize seu dashboard, adicione, edite e exclua alertas, adicione criptomoedas e exchanges, e seja notificado dos alertas existentes
- **Fluxo principal**
 - O usuário acessa o gerenciamento da carteira digital

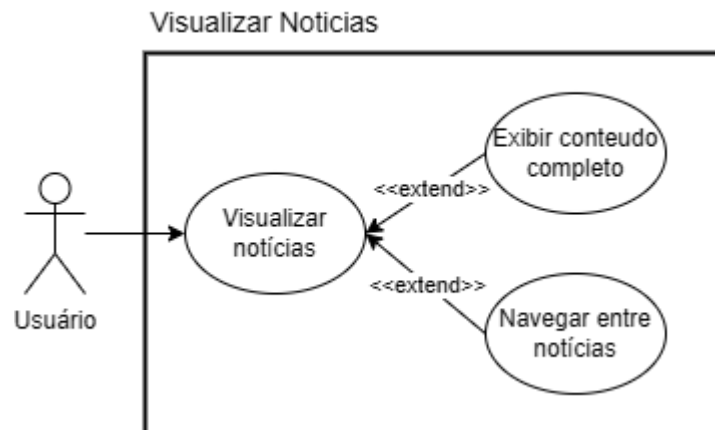


- O sistema exibe o dashboard pessoal, assim como as opções de adicionar criptomoedas e o gerenciamento de alertas
- Gerenciamento de alerta
 - O usuário seleciona o gerenciamento de alertas
 - O sistema exibe o menu de gerenciamento de alertas como todos os alertas adicionados (com opções de alterar alerta e excluir alerta) e o botão de adicionar alertas novos.
 - O usuário seleciona adicionar alerta
 - O sistema abre a interface de criação de alerta
 - O usuário seleciona uma moeda e quando quer receber o alerta
 - O sistema adiciona o alerta nos alertas do usuário
- Gerenciamento de Exchange
 - O usuário seleciona gerenciamento de exchanges
 - O sistema mostra o menu de exchange onde tem a opção de adicionar exchange e excluir exchange
 - Se tiver alguma exchange adicionada, mostra um caixa de seleção para selecionar a exchange a ser mostrada
 - O usuário seleciona a exchange adicionada
 - O sistema mostra um dashboard das atividades feitas na exchange adicionada.
 - adicionar exchange
 - O usuário seleciona adicionar exchange
 - O sistema mostra um popup pedindo um link fornecido pela exchange que quer adicionar
 - O usuário clica em adicionar
 - O sistema adicionar a exchange
 - O sistema mostra a dashboard da exchange
 - Excluir exchange



- O usuário seleciona a exchange que quer excluir
- O usuário seleciona Excluir
- O sistema mostra um popup perguntando se tem certeza que quer excluir
- Se o usuário confirma que quer excluir
- O sistema exclui os dados da exchange selecionada.
- **Fluxo Alternativo**
 - Se não tiver nenhuma exchange adicionada
 - O sistema mostra uma tela vazia com a imagem da marca do site de fundo.
 - Se o usuário não confirma que quer excluir a exchange
 - O sistema mostra o menu de Exchange novamente.
- **Exceção:**
 - O usuário pode tentar adicionar um alerta com a criptomoeda, e parâmetros iguais a outro alerta já existente
- **Pré-condições**
 - O usuário está logado
- **Pós-condições**
 - O usuário deve conseguir acessar a carteira digital
 - O usuário deve conseguir adicionar alertas sobre criptomoedas
 - O usuário deve conseguir editar alertas
 - O usuário deve conseguir excluir alertas
 - O usuário deve conseguir adicionar exchanges em seu perfil
 - O usuário deve conseguir excluir exchanges em seu perfil

4.5. Visualizar notícias



- **Ator Principal** - Usuário
- **Objetivo** - Permite que o usuário visualize notícias sobre o mundo das criptomoedas - tendências, “sentimento de mercado”, curiosidades e dicas de negócios com criptomoedas.
- **Fluxo principal**
 - O usuário acessa a funcionalidade de visualizar notícias
 - O sistema exibe várias caixinhas na vertical que possuem o título do conteúdo - sendo apenas uma prévia do conteúdo.
 - O usuário seleciona a caixinha desejada
 - O sistema apresenta a notícia na íntegra - texto, imagens ou vídeos - e apresenta a fonte do conteúdo no fim da página
 - O usuário pode sair da página da notícia, selecionando um “x” no canto esquerdo da página
 - O sistema volta para a página de visualizar notícias - exibindo todas as notícias atualizadas .
- **Fluxo Alternativo**
 - No passo 5 caso o usuário deseje apenas prosseguir para a próxima notícia, ele pode selecionar a seta abaixo do texto à direita da página,

e para ver a notícia anterior o usuário seleciona a seta abaixo do texto à esquerda da página

- **Pré-condições**

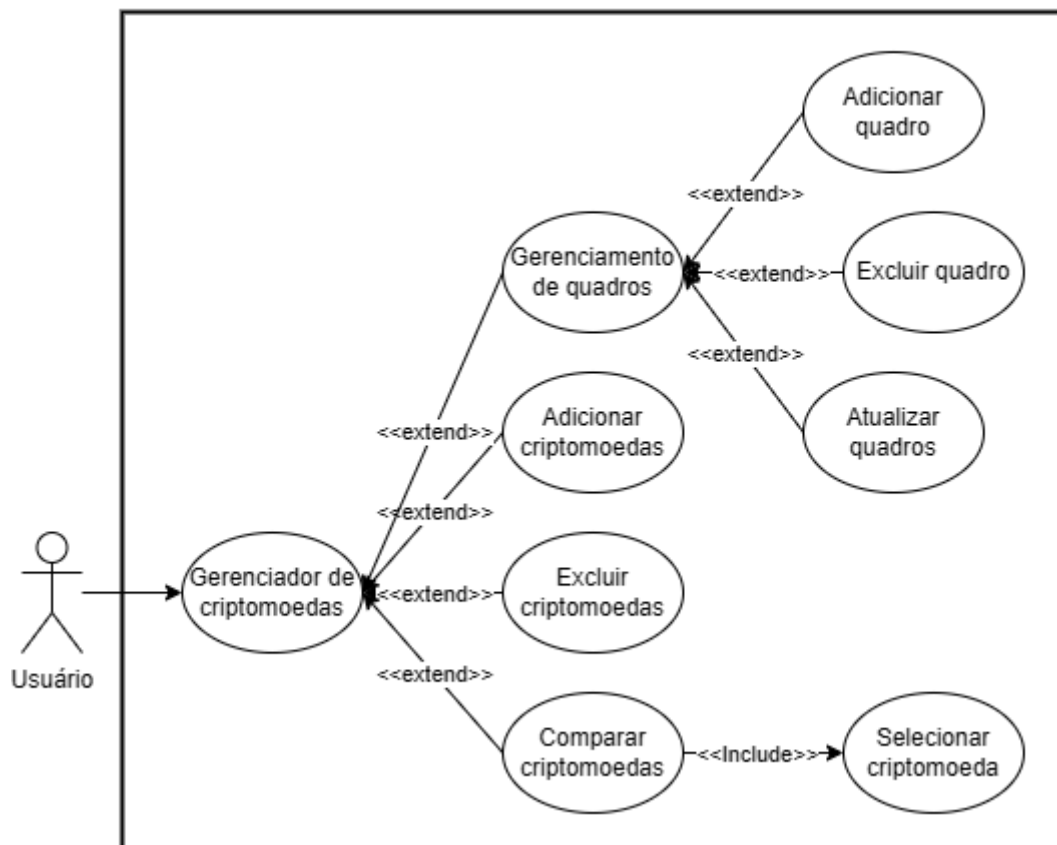
- O usuário está logado

- **Pós-condições**

- O usuário consegue visualizar todas as notícias atualizadas deste software.

4.6. Gerenciar criptomoedas

Gerenciador de criptomoedas

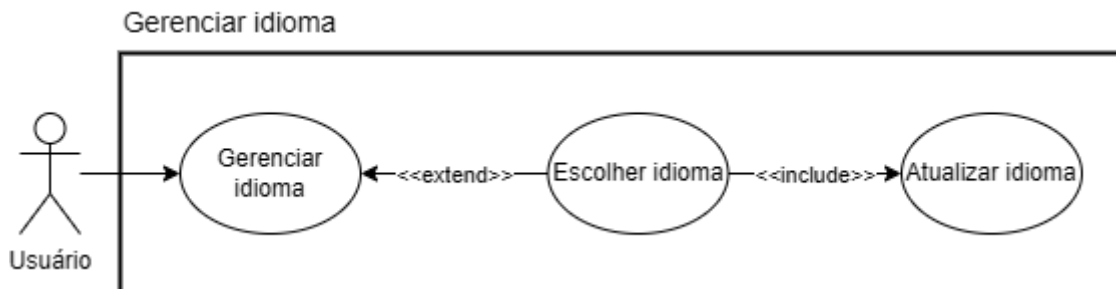


- **Ator Principal - Usuário**



- **Objetivo** - Permitir ao usuário visualizar, configurar e gerenciar os quadros de informação exibidos no dashboard de criptomoedas.
- **Fluxo principal**
 - O usuário acessa o dashboard de criptomoedas.
 - O sistema exibe os quadros de informações já adicionados (preços, gráficos, tendências, notícias).
 - O usuário seleciona uma ação para gerenciar os quadros:
 - Determinar o tempo em que um quadro cobre (1 dia, 1 semana, 1 mês, 1 trimestre ou 1 ano)
 - Ativar quadros específicos ao clicar no botão de adicionar quadros
 - Desativar quadros específicos em um botão no próprio quadro.
 - Atualizar as informações exibidas nos quadros.
 - O sistema aplica as alterações realizadas pelo usuário.
 - O dashboard reflete as mudanças imediatamente.
 - O caso de uso termina.
- **Pré-condições**
 - O usuário está logado
- **Pós-condições**
 - As configurações realizadas (ativação, exclusão, atualização ou ajustes no tempo de visualização) são salvas no sistema e refletidas no dashboard.

4.7. Gerenciador de idiomas



- **Ator Principal** - Usuário
- **Objetivo** - Permitir que o usuário possa mudar o idioma do sistema
- **Fluxo principal**
 - O usuário seleciona as configurações
 - O sistema exibe as configurações
 - O usuário seleciona a opção de alterar idioma
 - O sistema exibe os idiomas disponíveis
 - O usuário seleciona a opção de língua inglesa
 - O sistema muda o idioma para inglês
- **Fluxo Alternativo**
 - O usuário no passo 3, escolher manter o mesmo idioma
- **Pré-condições**
 - O usuário está logado no sistema
- **Pós-condições**
 - O usuário deve conseguir ver todos os idiomas disponíveis
 - O sistema deve mudar o idioma do site para o idioma selecionado

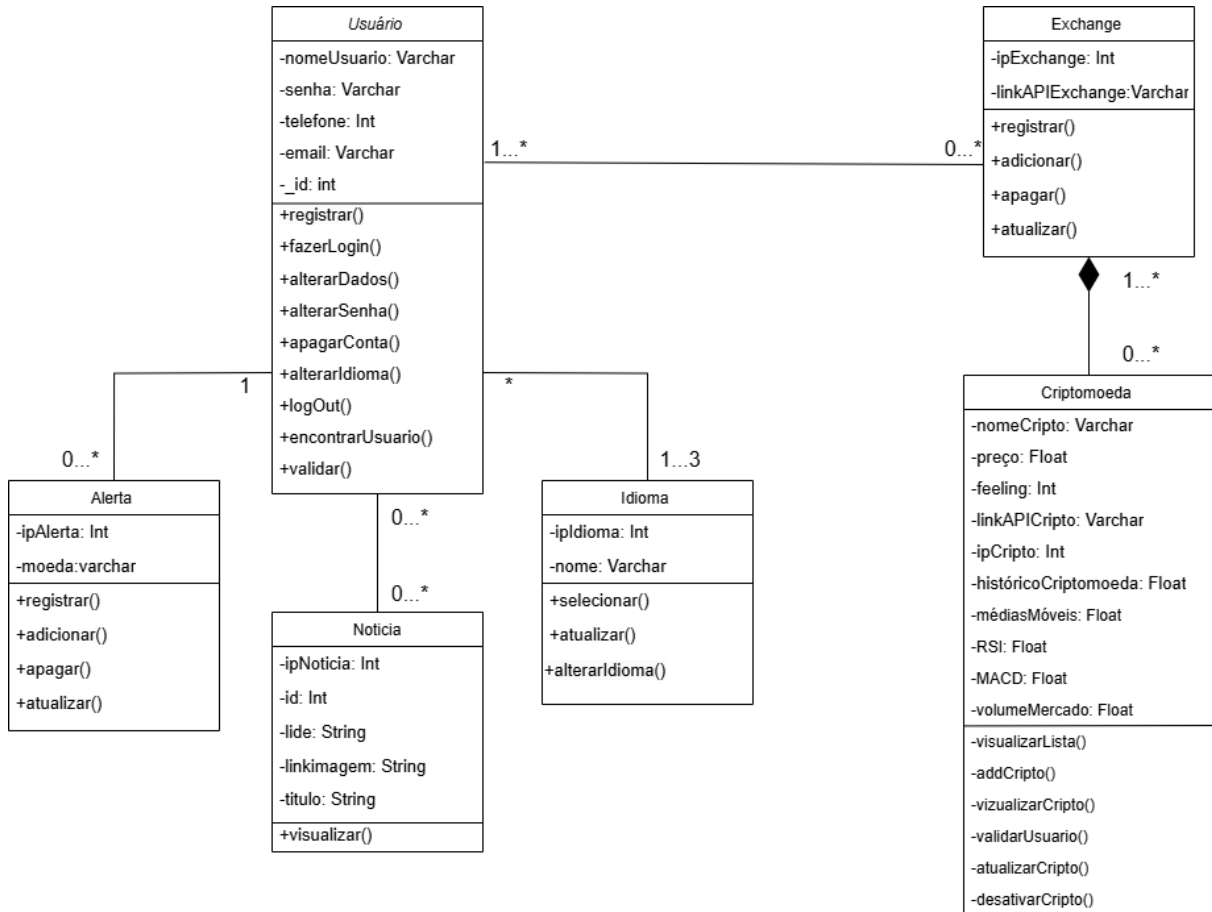


5. DIAGRAMA DE CLASSES

O diagrama de classes é um dos elementos mais importantes da UML (Unified Modeling Language) e desempenha um papel fundamental na modelagem de sistemas. Ele fornece uma representação visual da estrutura estática do software, organizando os principais componentes, como classes, atributos, métodos e as relações entre eles.

Por meio dessa representação, é possível identificar e estruturar os elementos essenciais do sistema, proporcionando uma visão clara de sua composição interna. Esse diagrama também serve como base para garantir que o desenvolvimento esteja alinhado aos objetivos do projeto, facilitando a comunicação entre os membros da equipe e promovendo a consistência durante as etapas de planejamento e implementação.

No contexto deste projeto, foram modeladas 6 classes essenciais para o sistema de portal de criptomoedas. A classe “Usuário” representa as pessoas que usarão o sistema para se informar sobre criptomoedas. A classe “Alerta” representa os alertas criados pelos usuários sobre a criptomoeda. A classe “Notícia” representa as notícias de criptomoedas que são mostradas para o usuário. A classe “Idioma” representa os diferentes tipos de linguagens que o sistema pode assumir. A classe “Exchange” representa as carteiras de criptomoedas disponíveis no mercado. Por último, a classe “Criptomoeda” representa as diferentes criptomoedas disponíveis para os usuários visualizarem os dados.



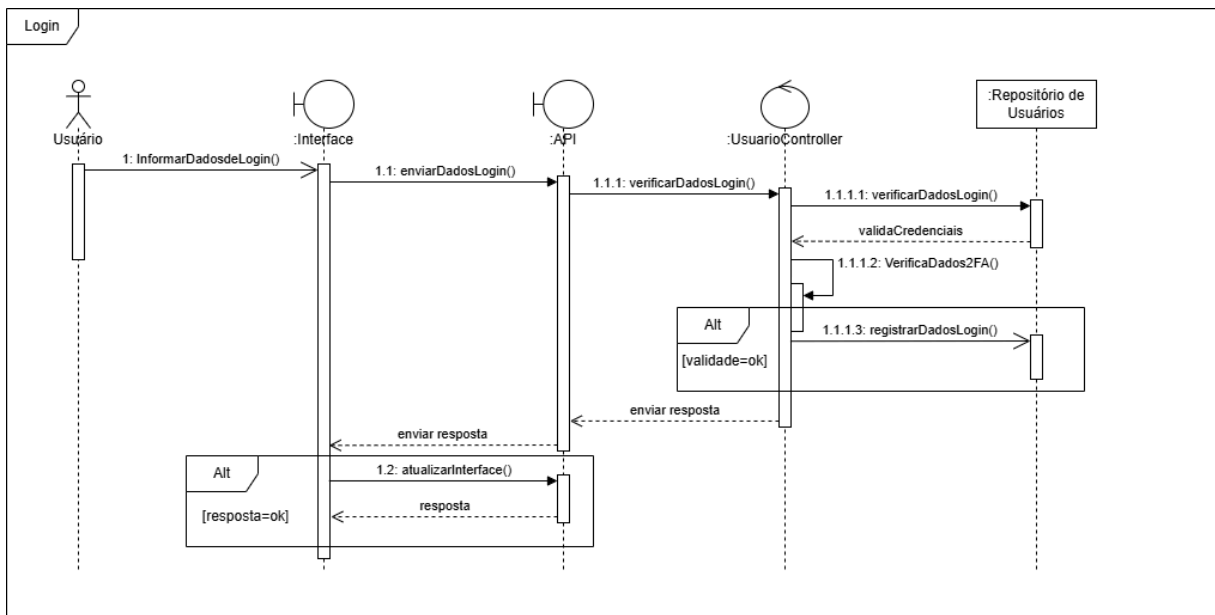
6. DIAGRAMAS DE SEQUÊNCIA

Um diagrama de sequência é uma ferramenta visual amplamente utilizada na engenharia de software, especialmente no contexto da UML (Unified Modeling Language). Ele é usado para modelar e descrever o comportamento de um sistema ao longo do tempo, evidenciando a interação entre objetos diferentes por meio de uma sequência de mensagens.

Esse tipo de diagrama destaca a ordem e o fluxo das interações, permitindo visualizar como os componentes do sistema colaboram para realizar uma funcionalidade ou um processo específico. É especialmente útil para compreender e

documentar a dinâmica de um sistema em diferentes cenários, auxiliando no alinhamento entre as partes interessadas e na validação dos requisitos.

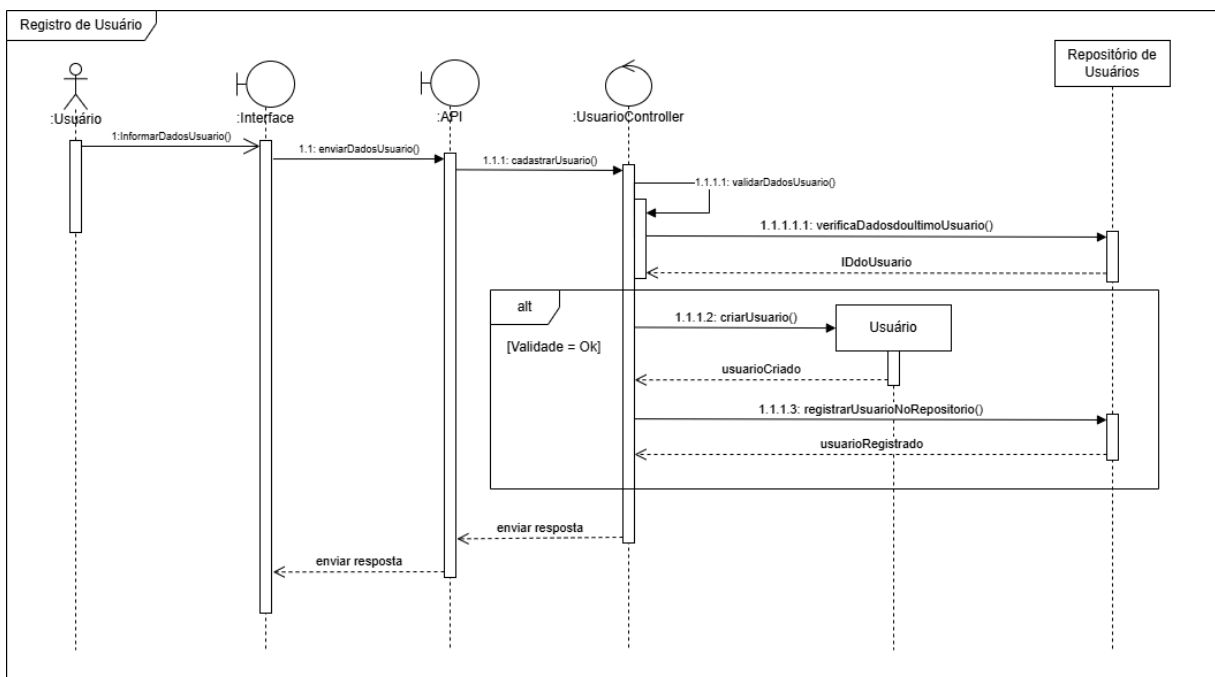
6.1. Login



O diagrama ilustra a interação entre os diferentes componentes envolvidos no login de um usuário, incluindo a interface, a API, o controlador de usuário e o repositório de usuários. O usuário insere suas credenciais na interface do sistema, a interface encaminha os dados do login para a API, o controlador de usuário recebe os dados e valida as credenciais fornecidas. Dentro deste processo, são realizadas as seguintes etapas: `verificarDadosLogin()`, onde o controlador solicita ao repositório de usuários a verificação das credenciais; `validaCredenciais`, momento em que as credenciais são comparadas com os dados armazenados no repositório; `VerificaDados2FA()`, caso seja necessária autenticação em dois fatores (2FA), ocorre uma verificação adicional; e `registrarDadosLogin()`, etapa em que os dados do login são registrados se a validação for bem-sucedida. O controlador retorna a resposta

para a API, que a encaminha para a interface. A interface recebe a resposta e pode seguir dois fluxos alternativos, representados pelo bloco "Alt": se a resposta for positiva (resposta=ok), a função atualizarInterface() é chamada para exibir a confirmação do login ao usuário, caso contrário, uma mensagem de erro é apresentada. O diagrama de sequência apresentado fornece uma visão clara do processo de login no sistema, evidenciando a interação entre o usuário e os componentes do sistema. O fluxo detalhado permite compreender as etapas de validação de credenciais, autenticação em dois fatores e registro dos dados de login, sendo fundamental para garantir um processo seguro e bem estruturado.

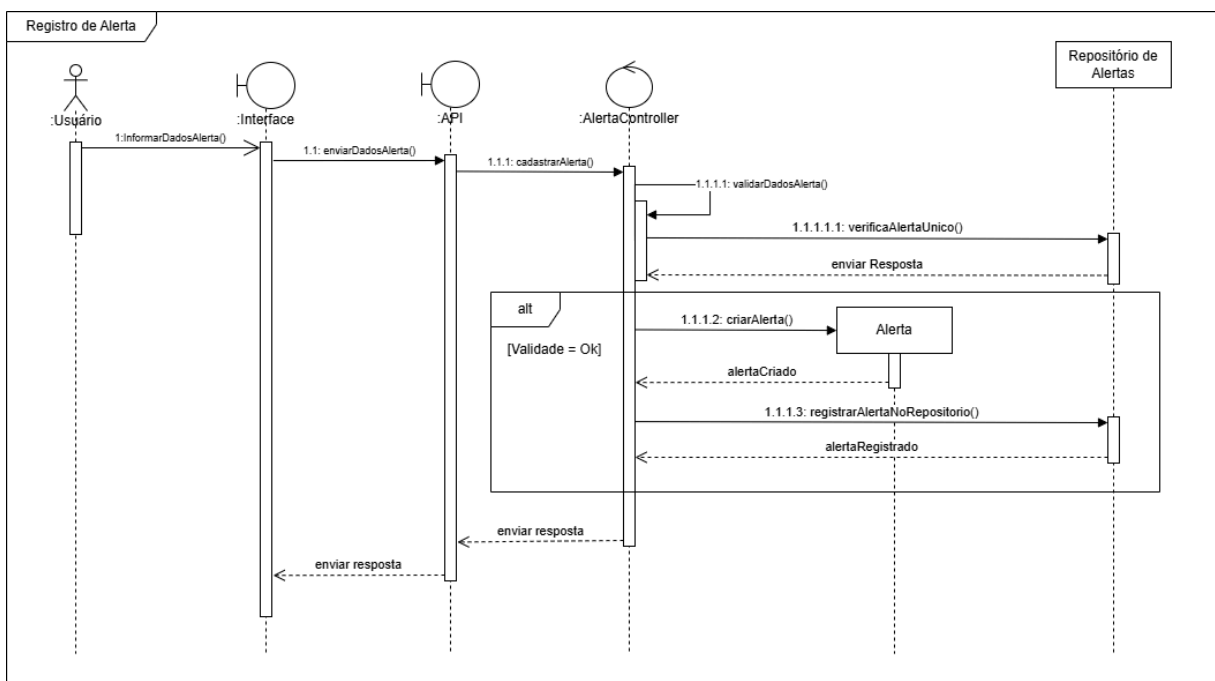
6.2. Registro usuário



O processo de registro de usuário começa quando o usuário insere seus dados de registro na interface, a interface envia esses dados para a API e em seguida, a API delega o processamento para o UsuarioController, que executa a

função `cadastrarUsuario()`. Dentro desse processo, ocorre a validação dos dados que verifica se as informações estão corretas e se o usuário é único no sistema. Caso os dados sejam válidos, cria-se a instância do novo usuário, e a função `registrarUsuarioNoRepositorio()` armazena o usuário no repositório. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface. Se a validação for bem-sucedida, o usuário é informado de que o registro foi concluído com sucesso.

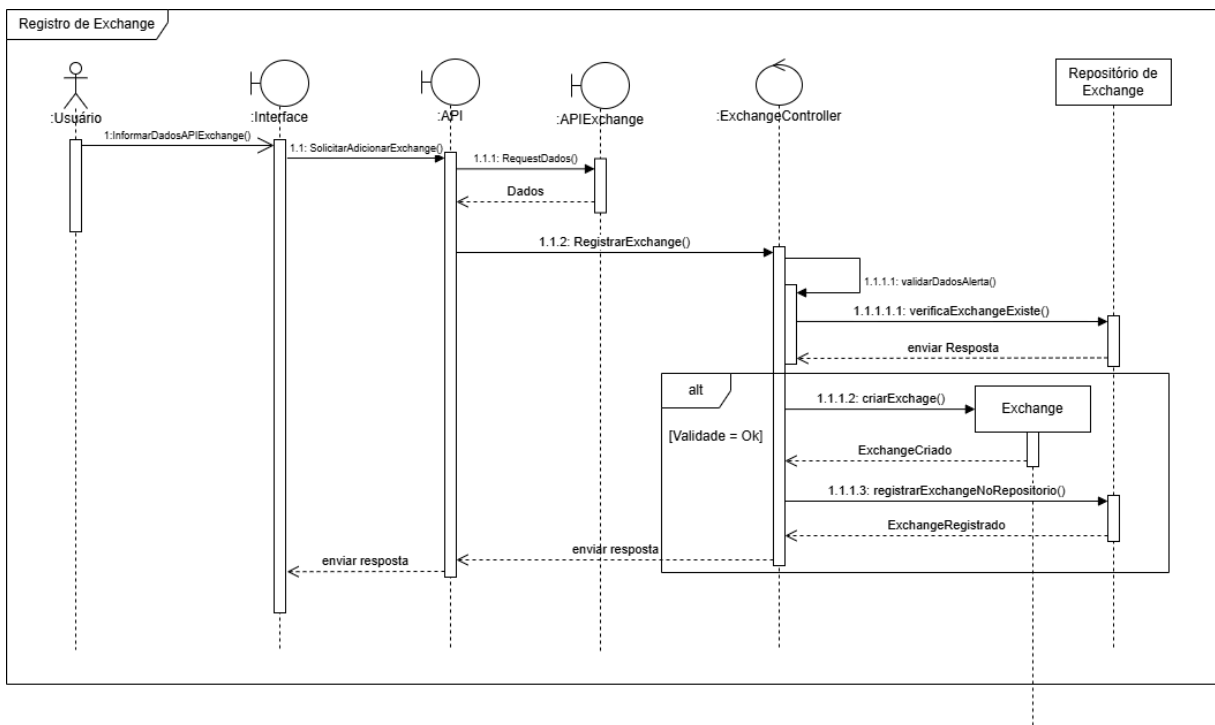
6.3. Registro alerta



O usuário fornece as informações para criar um alerta na interface, a interface envia os dados para a API da qual delega a solicitação ao `AlertaController`, que executa a função `cadastrarAlerta()`. Dentro desse processo, verifica-se a consistência dos dados e se o alerta é único. Caso os dados sejam válidos cria-se o novo alerta e o armazena no repositório correspondente. Assim como no registro de

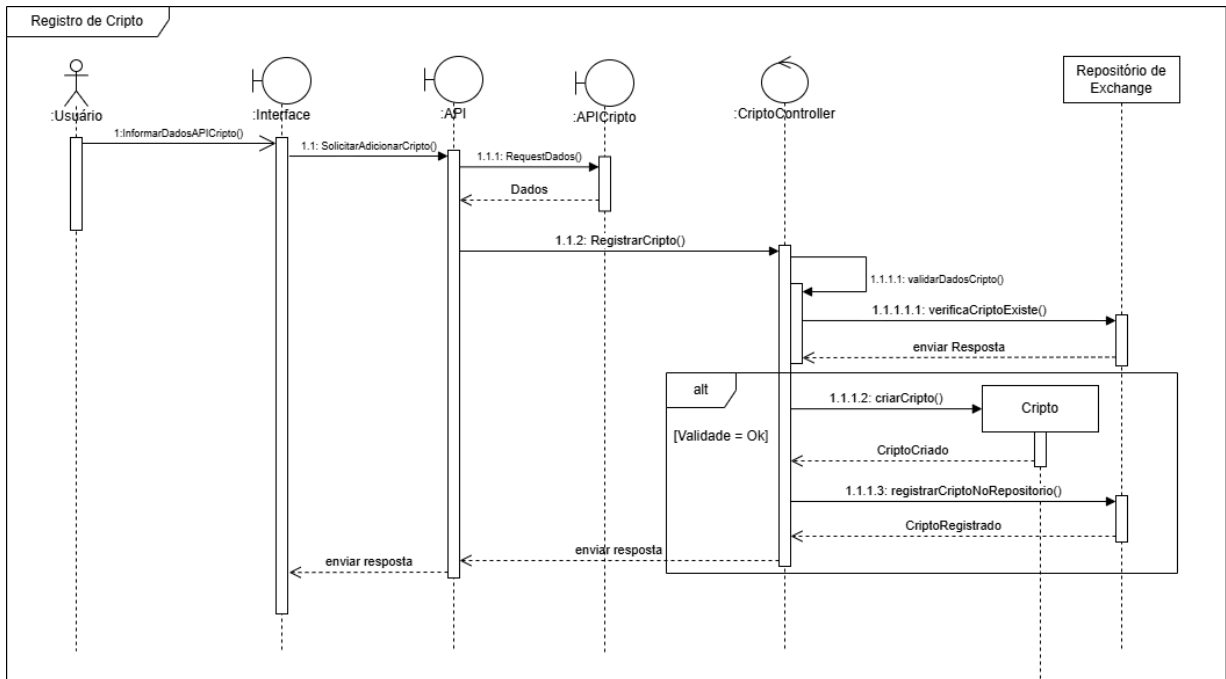
usuário, o controlador retorna a resposta para a API, que a envia para a interface. Se o registro for bem-sucedido, o usuário é notificado do sucesso da operação.

6.4. Registro exchange



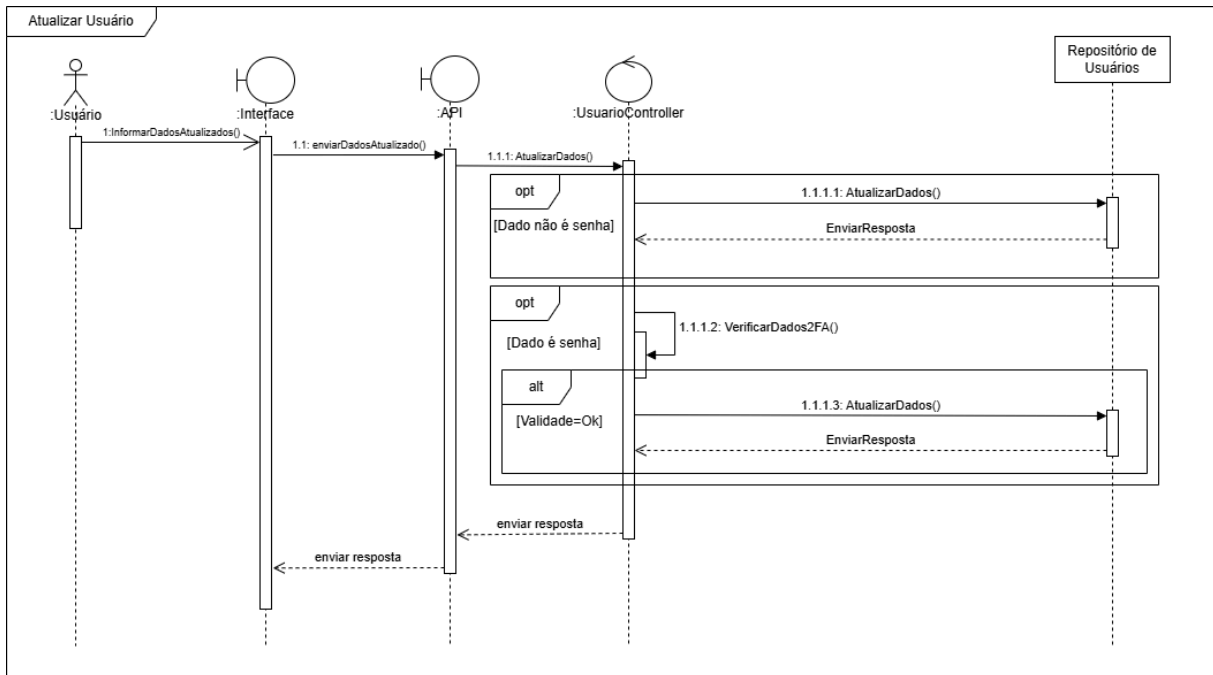
O processo de registro de exchange começa quando o usuário insere dados da API exchange na interface, a interface envia esses dados para a API e em seguida, a API faz a requisição desses dados para a API externa da exchange, e depois delega o processamento para o CriptoController, que executa a função registrarExchange(). Dentro desse processo, ocorre a validação dos dados que verifica se as informações estão corretas e se a exchange é única no sistema. Caso os dados sejam válidos, cria-se a instância da nova exchange, e a armazena no repositório. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface.

6.5. Registro criptomoeda



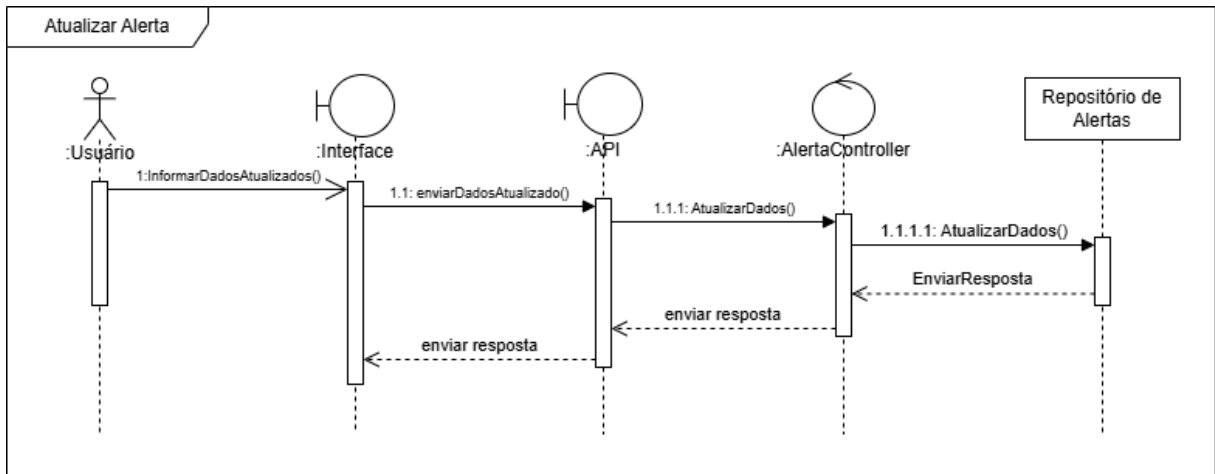
O processo de registro de cripto começa quando o usuário insere dados da API cripto na interface, a interface envia esses dados para a API e em seguida, a API faz a requisição desses dados para a API externa da cripto, e depois delega o processamento para o CriptoController, que executa a função registrarCripto(). Dentro desse processo, ocorre a validação dos dados que verifica se as informações estão corretas e se a cripto é única no sistema. Caso os dados sejam válidos, cria-se a instância da nova cripto, e a armazena no repositório. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface.

6.6. Atualizar usuário



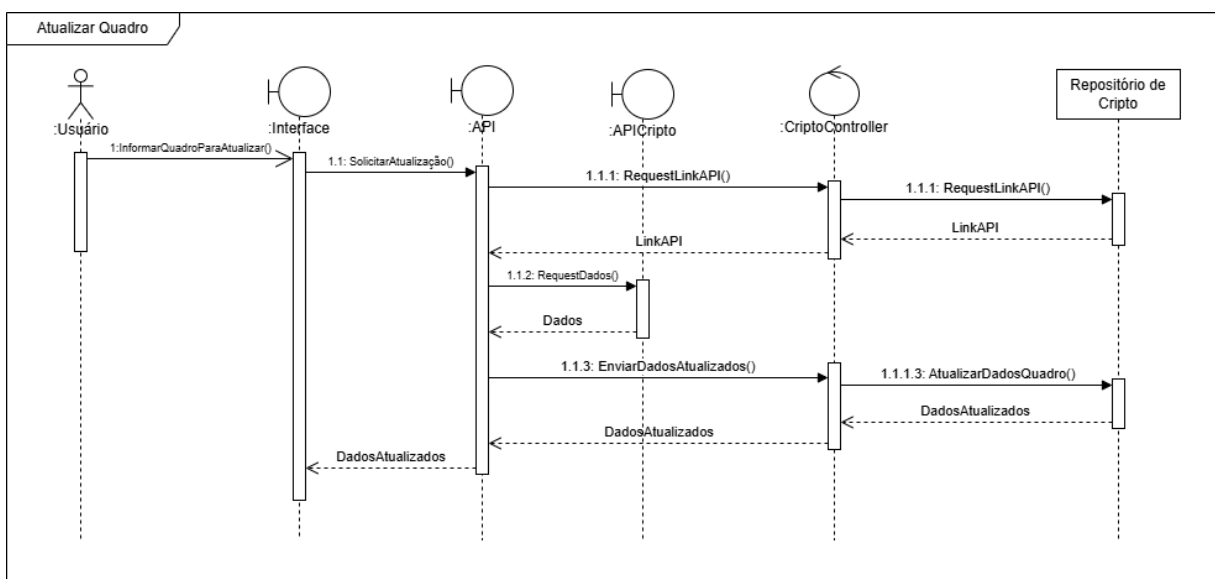
O processo de atualizar usuário começa quando o usuário insere dados atualizados na interface, a interface envia esses dados para a API e em seguida, a API delega o processamento para o UsuarioController, que atualiza os dados se o dado não for senha ou faz uma verificação de 2 fatores antes de atualizar os dados se for senha. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface.

6.7. Atualizar alerta



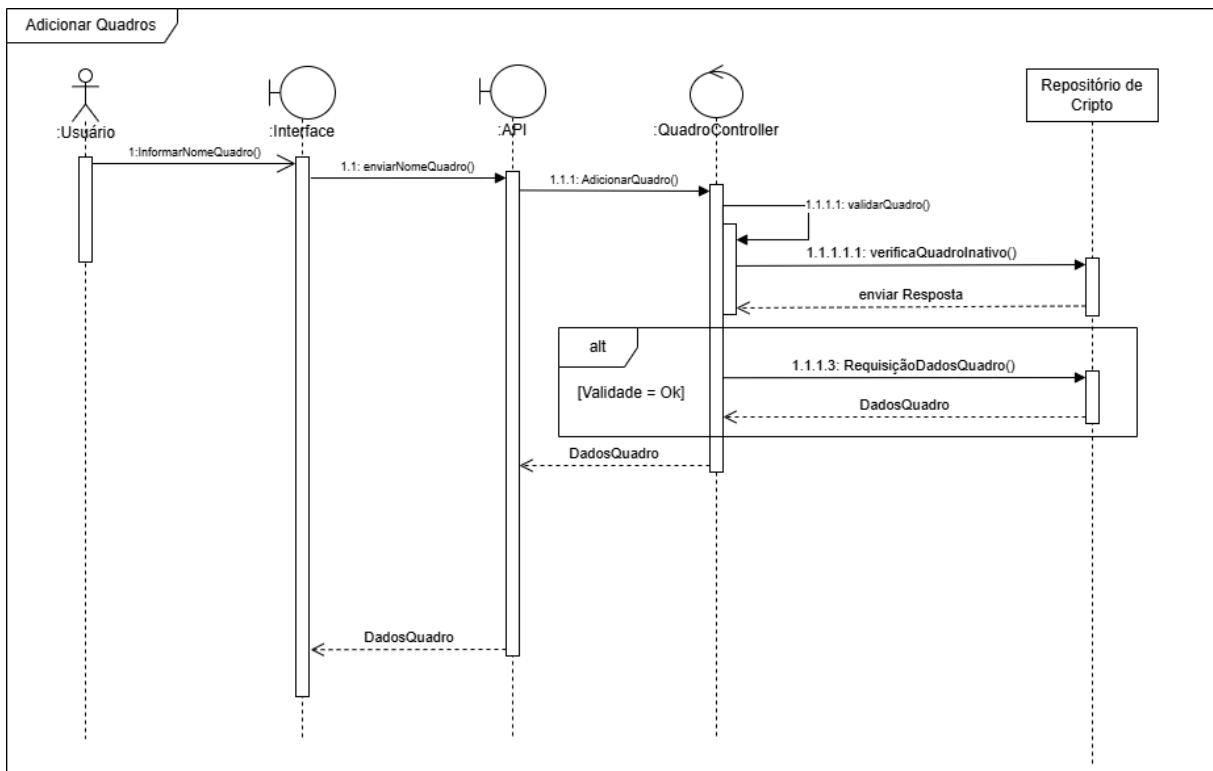
O processo de atualizar alerta começa quando o usuário insere dados atualizados do alerta na interface, a interface envia esses dados para a API e em seguida, a API delega o processamento para o AlertaController, que executa a Atualização. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface.

6.8. Atualizar quadro



O processo de atualizar quadro começa quando o usuário informa o quadro para atualizar na interface, a interface envia esses dados para a API e em seguida, a API busca no banco de dados o link da requisição para que depois a API faça a requisição dos dados atualizados para a API externa da cripto, e depois delega o processamento para o CriptoController, que atualiza o quadro. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface.

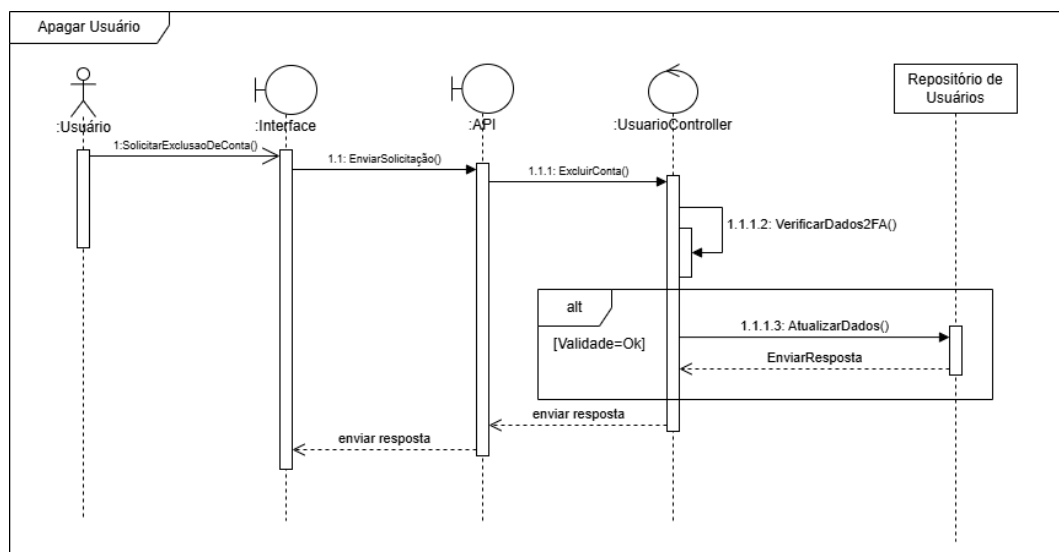
6.9. Adicionar quadro



O processo de adicionar quadro começa quando o usuário insere o nome do quadro na interface, a interface envia esses dados para a API e em seguida, a API delega o processamento para o QuadroController, que executa a função adicionarQuadro(). Dentro desse processo, ocorre a validação dos dados que verifica se o quadro realmente está inativo. Caso os dados sejam válidos, é feita a

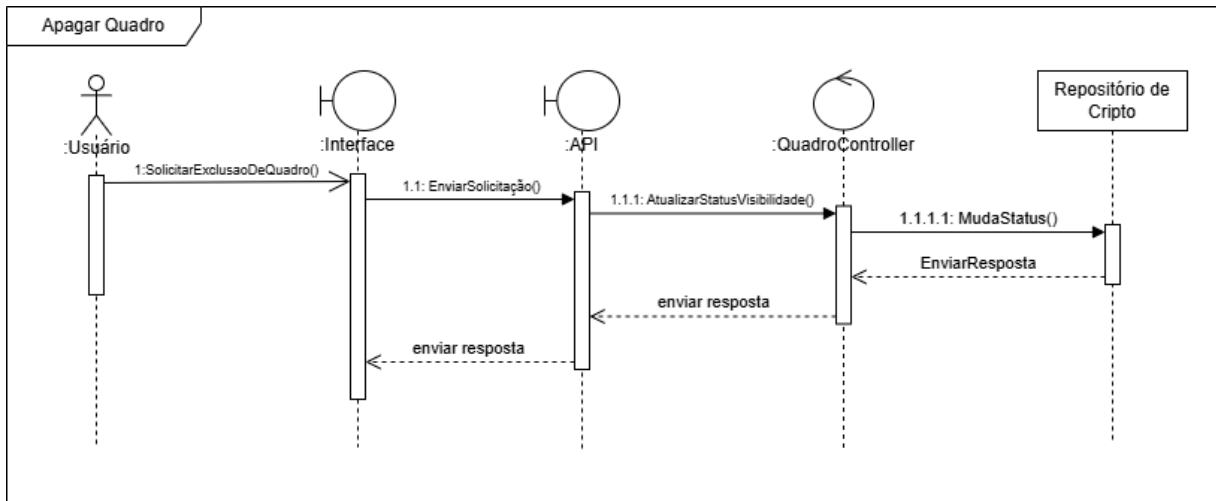
requisição dos dados do quadro. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface.

6.10. Apagar usuário



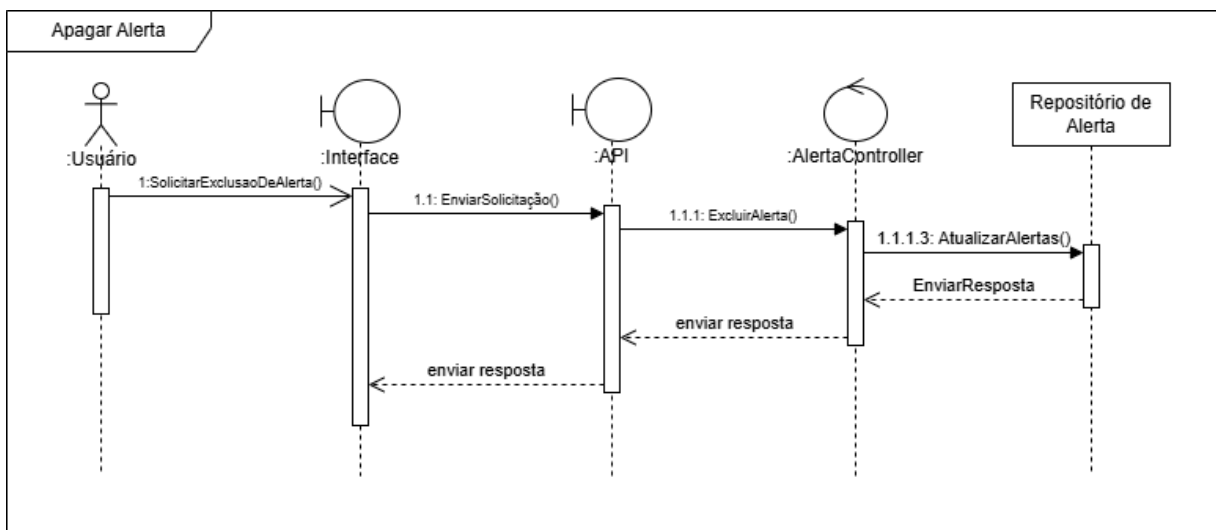
O processo de apagar usuário começa quando é solicitado a exclusão da conta na interface, a interface envia essa solicitação para a API e em seguida, a API delega o processamento para o UsuarioController, que executa a função `excluirConta()`. Dentro desse processo, ocorre a validação com a verificação de dois fatores. Caso os dados sejam válidos, exclui-se a conta e atualiza o repositório. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface.

6.11. Apagar quadro



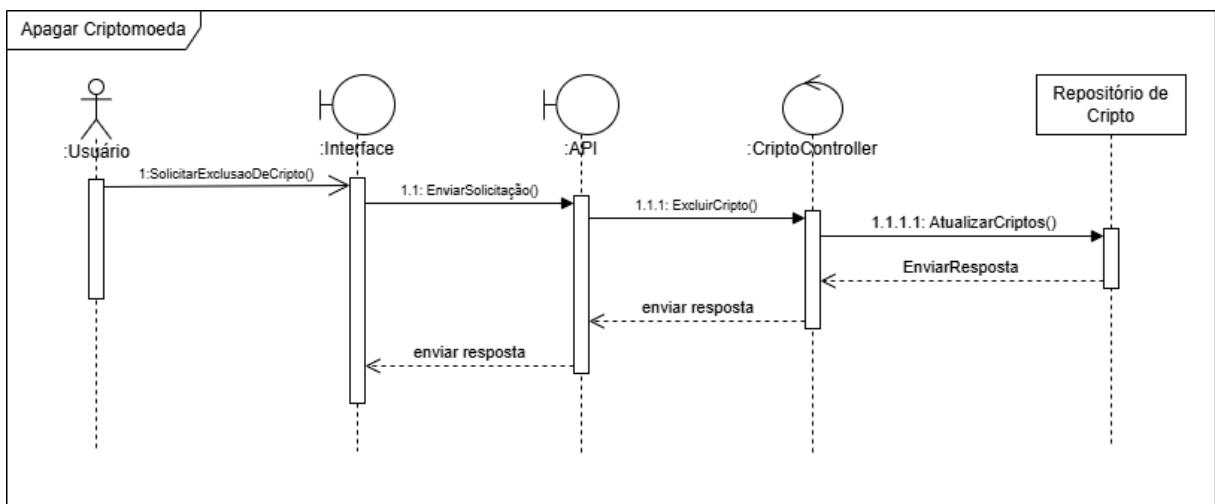
O processo de apagar quadro começa quando o usuário solicita exclusão de quadros na interface, a interface envia essa solicitação para a API e em seguida, a API delega o processamento para o QuadroController, que executa a Exclusão. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface.

6.12. Apagar alerta



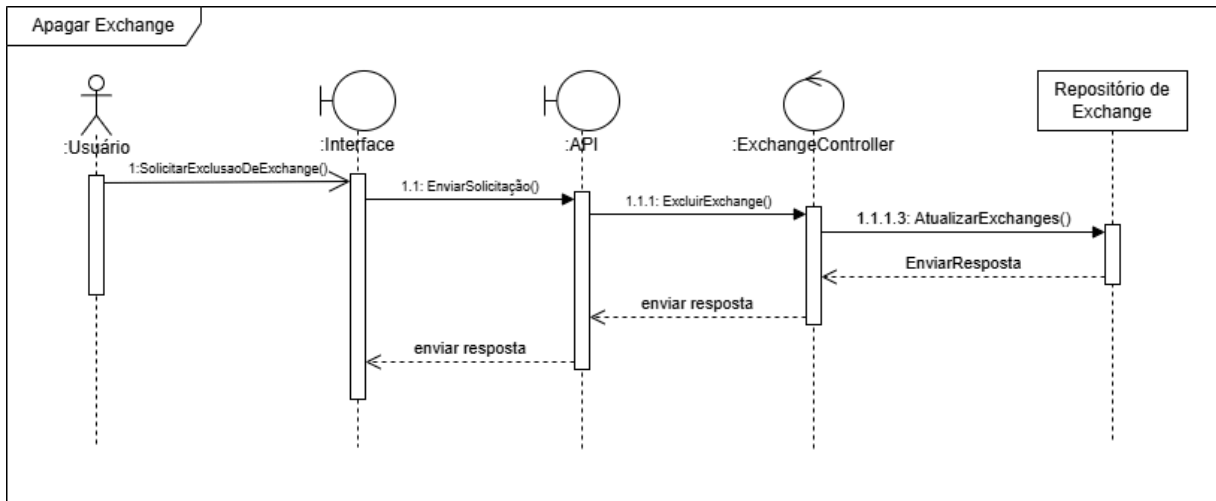
O processo de apagar alerta começa quando o usuário solicita exclusão do alerta na interface, a interface envia esses dados para a API e em seguida, a API delega o processamento para o AlertaController, que executa a Exclusão. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface.

6.13. Apagar criptomoeda



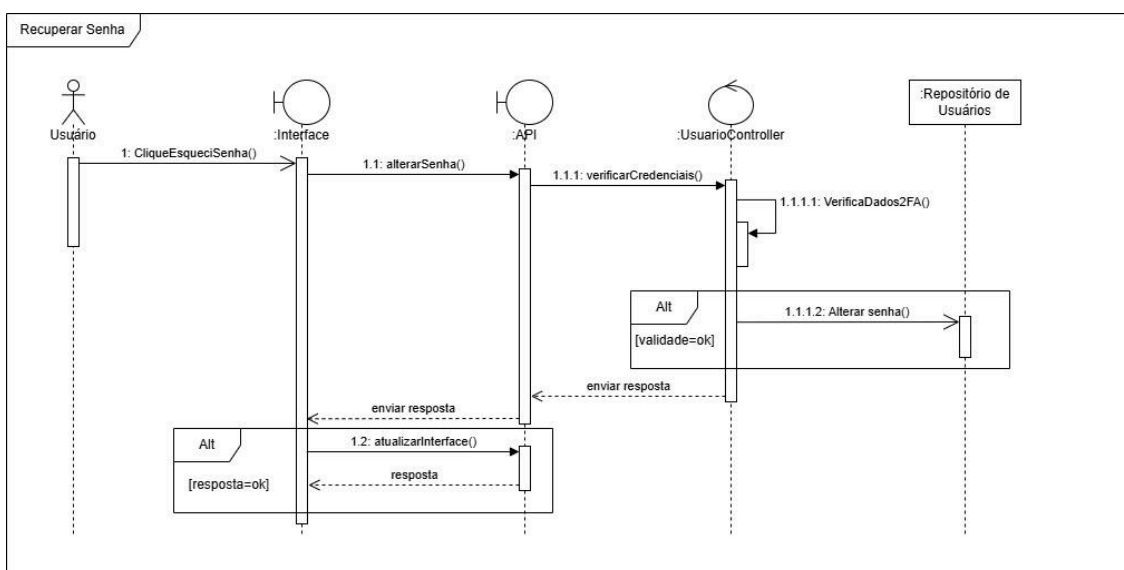
O processo de apagar criptomoeda começa quando o usuário solicita exclusão da criptomoeda na interface, a interface envia esses dados para a API e em seguida, a API delega o processamento para o CriptoController, que executa a Exclusão. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface.

6.14. Apagar exchange



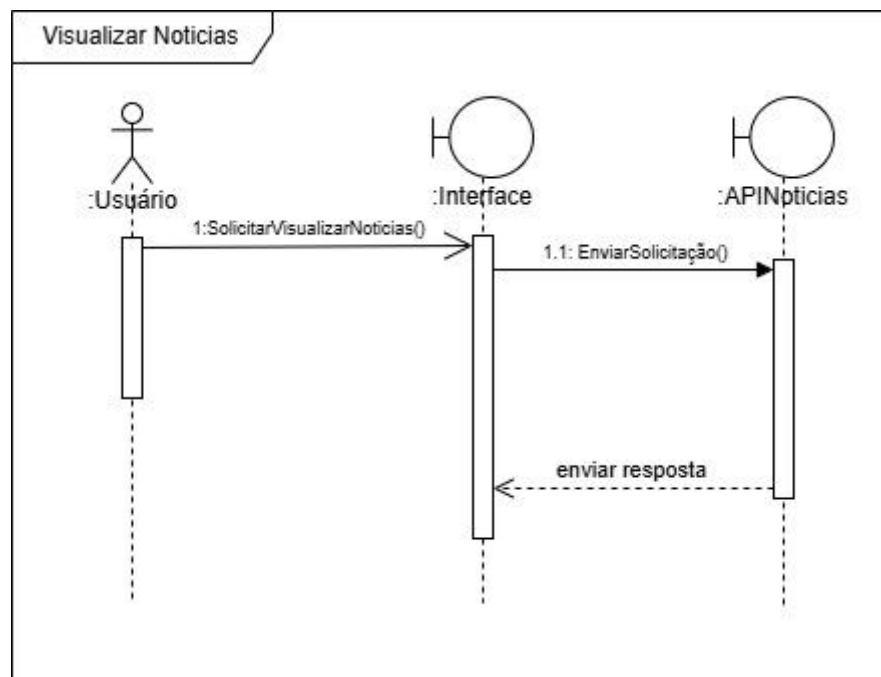
O processo de apagar exchange começa quando o usuário solicita exclusão do exchange na interface, a interface envia esses dados para a API e em seguida, a API delega o processamento para o ExchangeController, que executa a Exclusão. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface.

6.15. Recuperar senha



O processo de recuperação de senha e atualização da mesma começa quando o usuário avisa na interface que quer atualizar a senha interface, a interface envia esses dados para a API e em seguida, a API delega o processamento para o UsuarioController, que verifica as credenciais com a verificação de dois fatores. Caso os dados sejam válidos, altere a senha do repositório. Finalmente, o controlador retorna a resposta para a API, que a envia para a interface. Se a validação for bem-sucedida, o usuário é informado de que a atualização foi concluída com sucesso.

6.16. Visualizar notícia



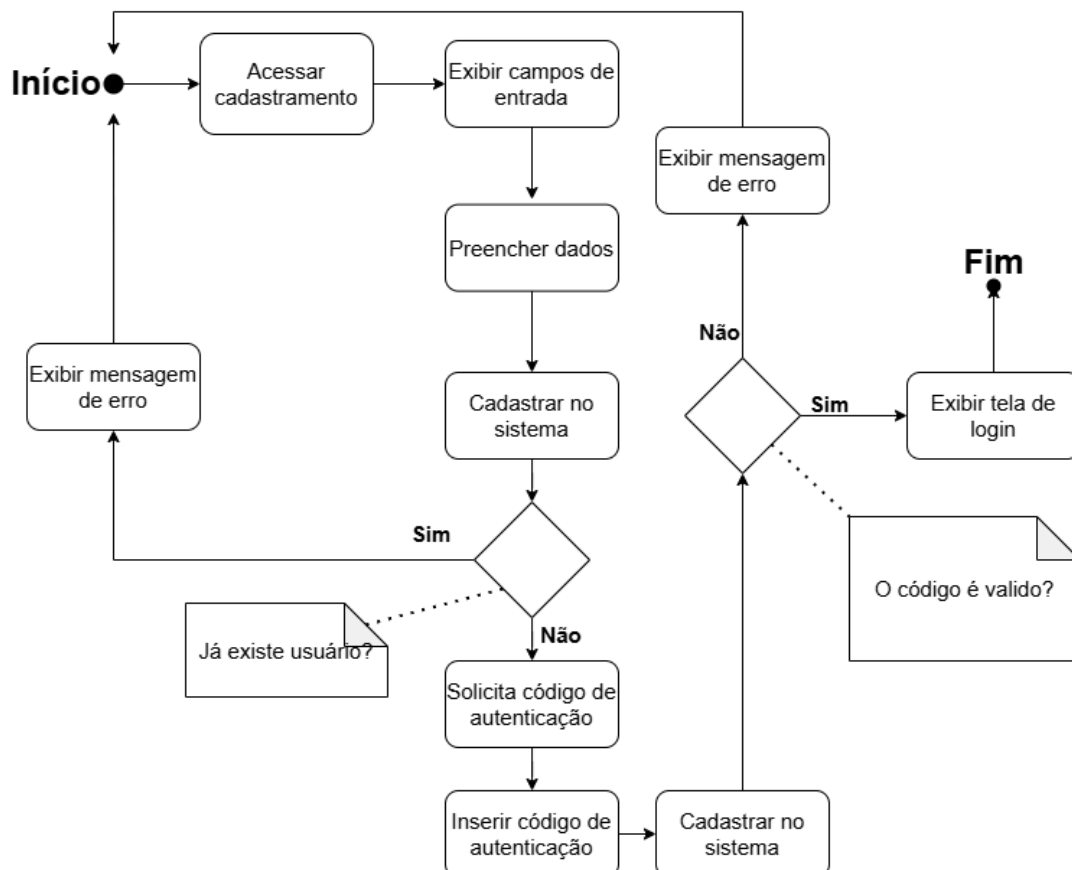
O processo de apagar alerta começa quando o usuário solicita visualizar notícias na interface, a interface envia esses dados para a API de Notícias externas. Finalmente, a API envia para a interface.

7. DIAGRAMAS DE ATIVIDADES

Um diagrama de atividades em UML é uma representação gráfica que descreve o fluxo de atividades em um sistema ou processo, destacando sequências, decisões e paralelismos. Utilizando elementos como estados de atividade e transições, ele mapeia o caminho de uma tarefa ou evento do início ao fim.

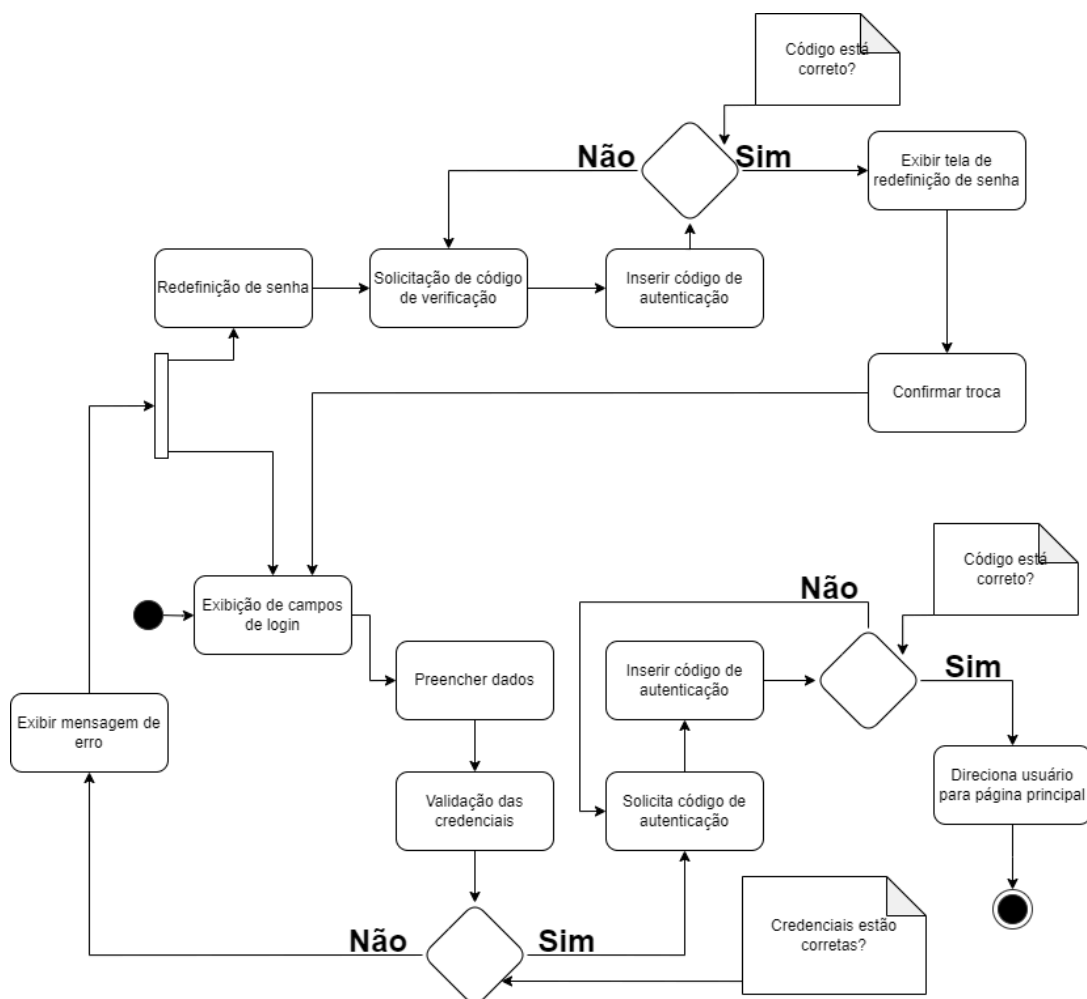
Sua utilidade está em simplificar processos complexos, promovendo entendimento entre equipes e partes interessadas. Amplamente aplicado em análise de requisitos, design de sistemas e documentação, ajuda a identificar gargalos, melhorar fluxos de trabalho e alinhar expectativas em projetos colaborativos.

7.1 Cadastrar Usuário



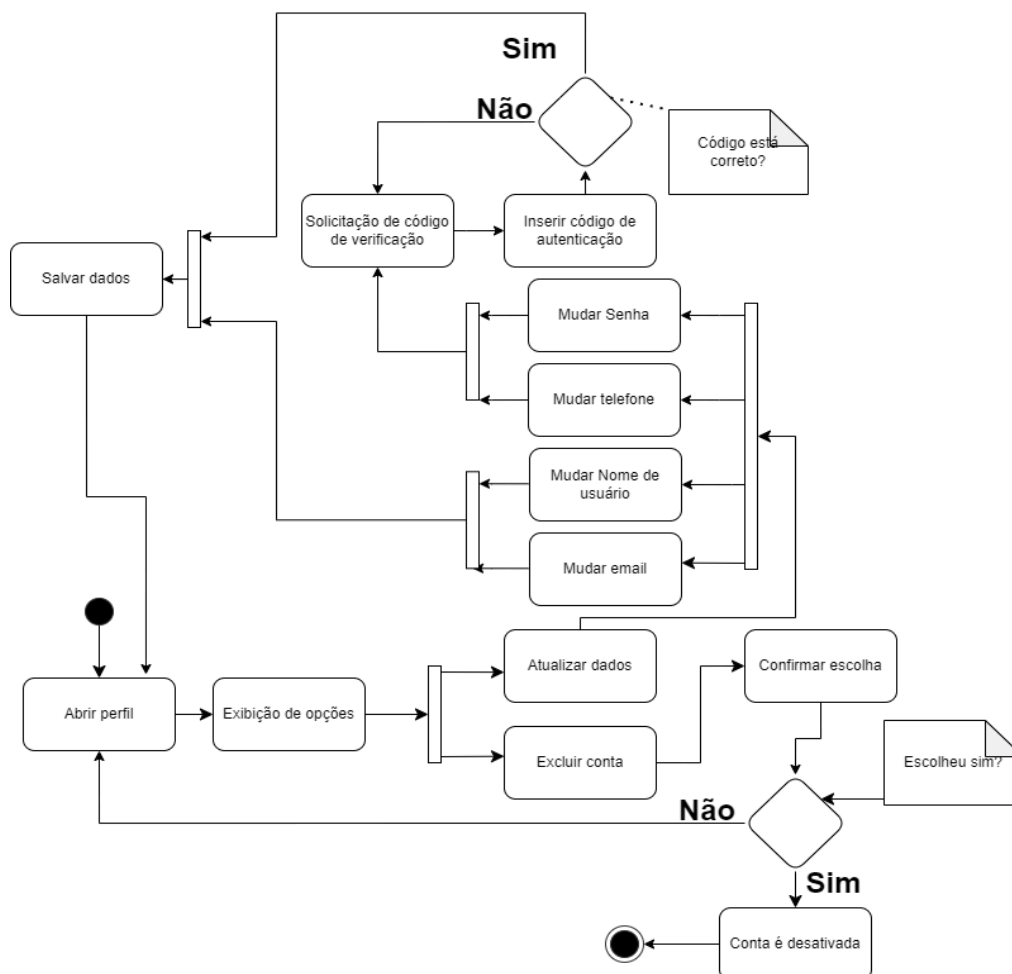
A atividade representa o fluxo de cadastro de um usuário em um sistema, começando pelo acesso ao cadastramento e a exibição dos campos de entrada. Após o preenchimento dos dados, o sistema verifica se o usuário já existe; se sim, exibe uma mensagem de erro e reinicia o processo. Caso contrário, solicita um código de autenticação. Se o código inserido for válido, o cadastro é concluído e a tela de login é exibida; caso contrário, uma mensagem de erro é apresentada. O diagrama inclui decisões condicionais para validação de dados e tratamento de erros, garantindo a segurança e a integridade do processo de cadastro.

7.2 Efetuar login



O diagrama de atividade acima apresenta o fluxo de login e redefinição de senha de um sistema, com validações de segurança e tratamento de erros. O processo começa com a exibição dos campos de login, onde o usuário insere suas credenciais. Se forem corretas, o sistema direciona para a página principal; se incorretas, há a opção de exibir uma mensagem de erro ou solicitar autenticação adicional por meio de um código. Caso necessário, o usuário pode optar por redefinir a senha, iniciando a solicitação de um código de verificação. Se o código estiver correto, a tela de redefinição é exibida, permitindo a troca da senha.

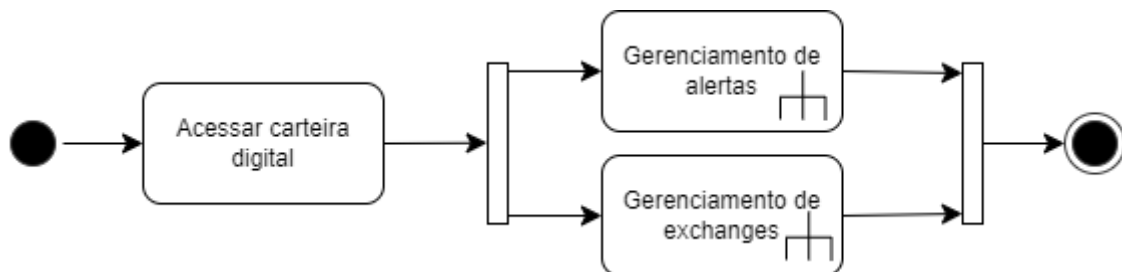
7.3 Gerenciar perfil



O fluxo de atividade se inicia ao usuário abrir o seu perfil e selecionar a exibição de opções, onde se o usuário escolher excluir sua conta, precisará confirmar a sua escolha. Se continuar com a decisão de excluir a conta, a sua conta vai ser desativada, se mudar de ideia, vai ser levado para a página de exibição de perfil.

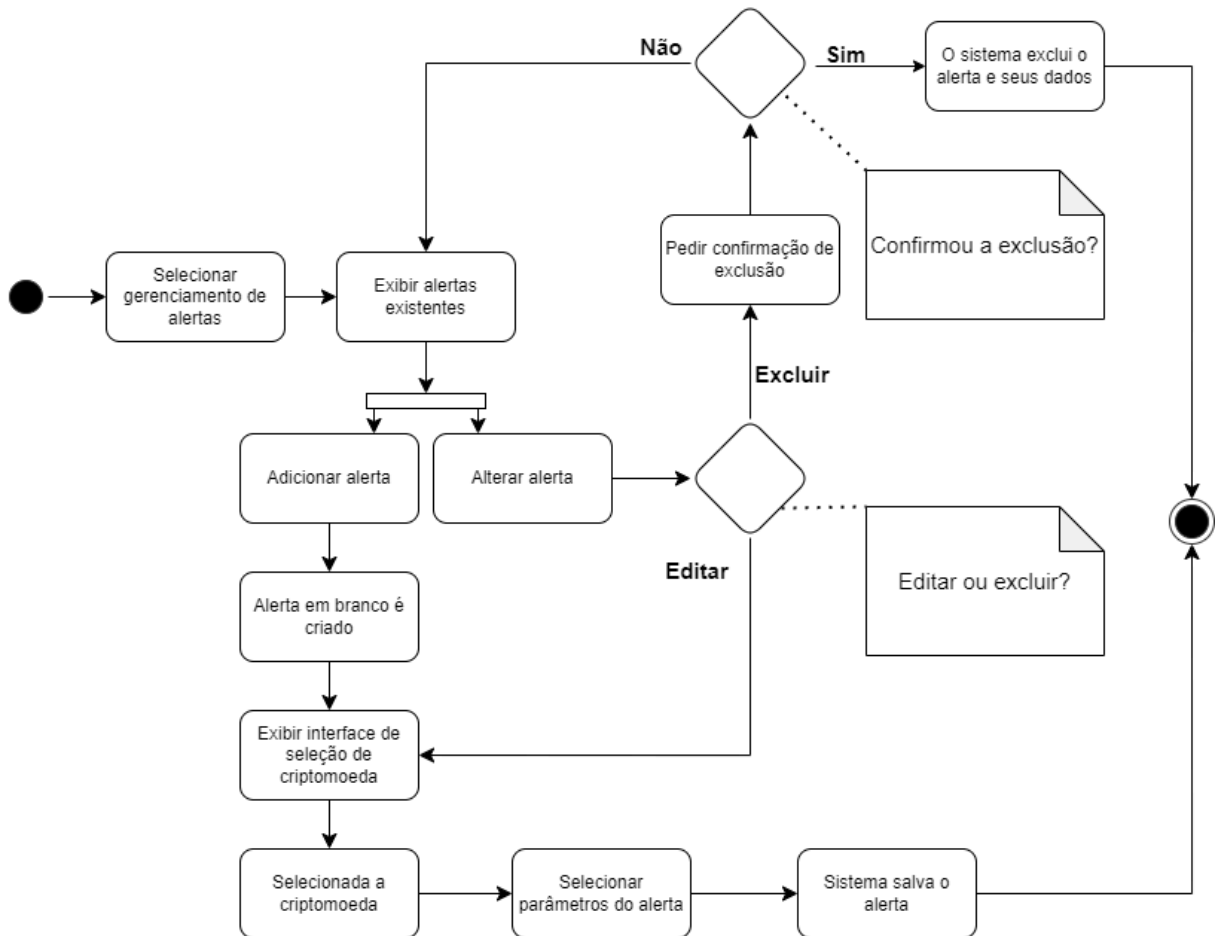
Se o usuário escolher atualizar os seus dados, quatro opções aparecerão (mudar senha, mudar telefone, mudar nome de usuário, e mudar email), se for escolhido alterar a senha ou o telefone, o sistema solicitará uma verificação com código de verificação antes de salvar as mudanças no sistema.

7.4 Gerenciamento de carteira



Nesse fluxo de trabalho o usuário acessa a carteira digital, onde pode acessar os sistemas de gerenciamento de alertas e de exchanges

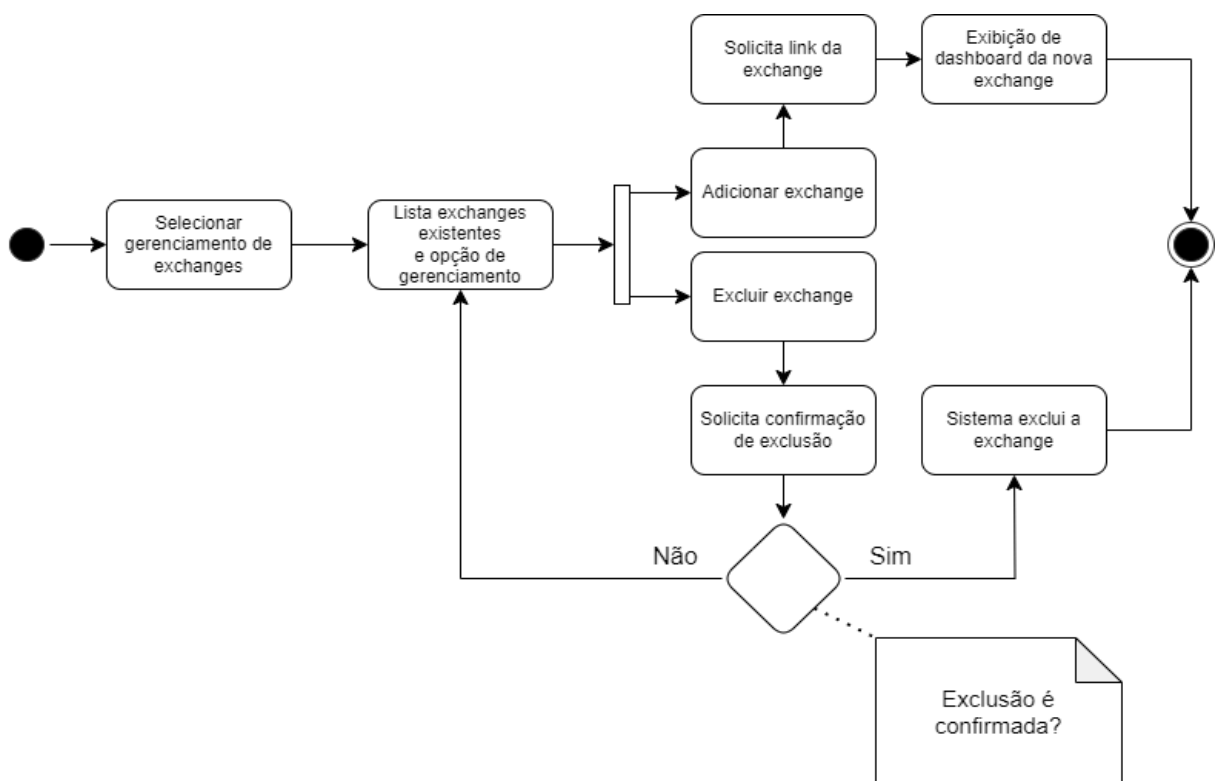
7.5 Gerenciar alertas



O fluxo de trabalho acima se inicia quando o usuário após acessar a carteira digital, seleciona o gerenciamento de alertas, o sistema exibe os alertas existentes e as opções disponíveis, se o usuário escolher adicionar alerta, um alerta em branco é criado, onde então vai ser exibida uma interface de seleção de criptomoedas. E quando é preenchido completamente o sistema salva o alerta.

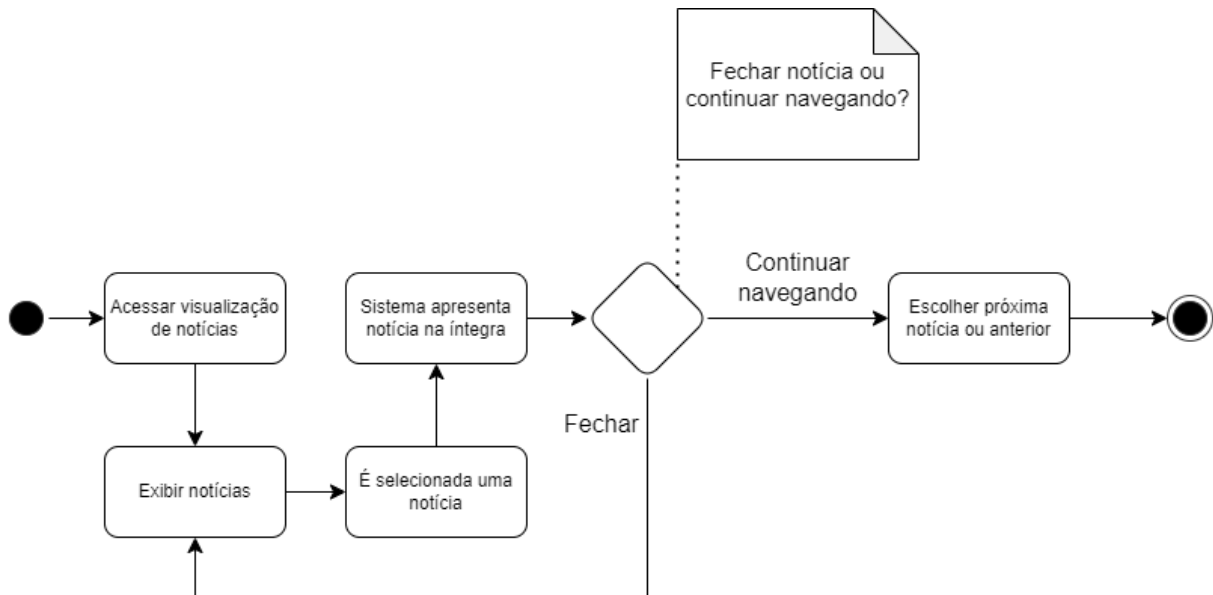
Se o usuário escolher alterar um alerta existente, o sistema pede uma confirmação para a alteração do alerta, assim que for confirmado, o fluxo continua para a interface de seleção de criptomoedas.

7.6 Gerenciar exchanges



O diagrama de atividades acima representa o fluxo de gerenciamento de exchanges dentro de um sistema. O processo inicia com a seleção da opção de gerenciamento de exchanges, levando à listagem das exchanges existentes e às opções de gerenciamento. O usuário pode optar por adicionar uma nova exchange, solicitando um link e exibindo o dashboard correspondente, ou excluir uma exchange. No caso de exclusão, o sistema solicita confirmação; se a exclusão for confirmada, a exchange é removida, caso contrário, retorna à listagem das exchanges. O fluxo encerra após a realização das ações escolhidas.

7.7 Visualizar notícias

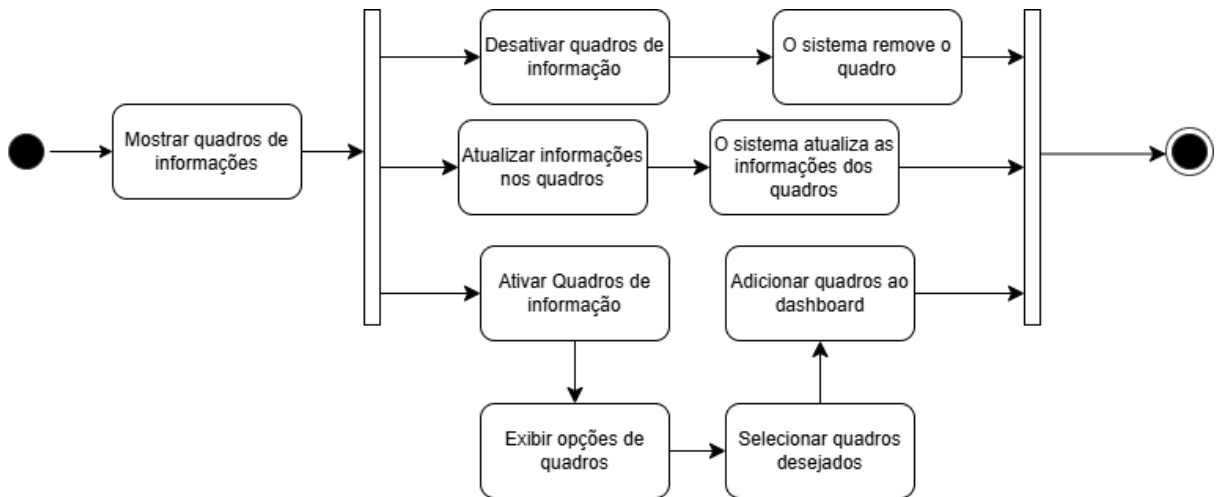


O diagrama de atividades representa o fluxo de navegação em um sistema de visualização de notícias. O processo inicia com o acesso à visualização de notícias, onde as notícias são exibidas. O usuário pode selecionar uma notícia específica, e o sistema apresenta seu conteúdo completo. Em seguida, há uma decisão: fechar a notícia ou continuar navegando. Se optar por continuar, o usuário pode escolher a próxima ou a notícia anterior. Caso contrário, retorna à exibição das notícias. O fluxo finaliza após a escolha da próxima ação pelo usuário.

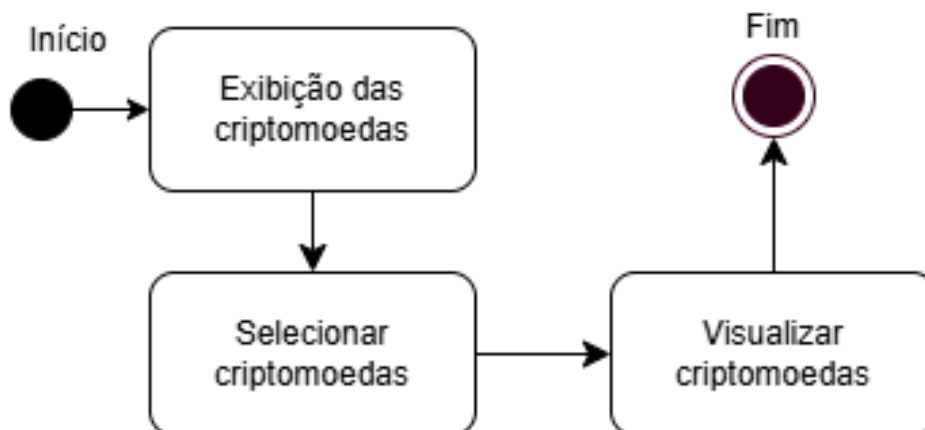
7.8 Visualizar dashboard de criptomoedas

O diagrama de atividades abaixo representa o processo de gerenciamento de quadros de informações em um sistema. O fluxo inicia com a exibição dos quadros de informações, permitindo ao usuário diversas ações. Ele pode desativar quadros, levando o sistema a removê-los; atualizar informações, fazendo com que o sistema atualize os dados nos quadros; ou ativar novos quadros, exibindo as opções

disponíveis. Caso o usuário ative quadros, ele pode selecionar os desejados e adicioná-los ao dashboard. Após essas ações, o fluxo é concluído.

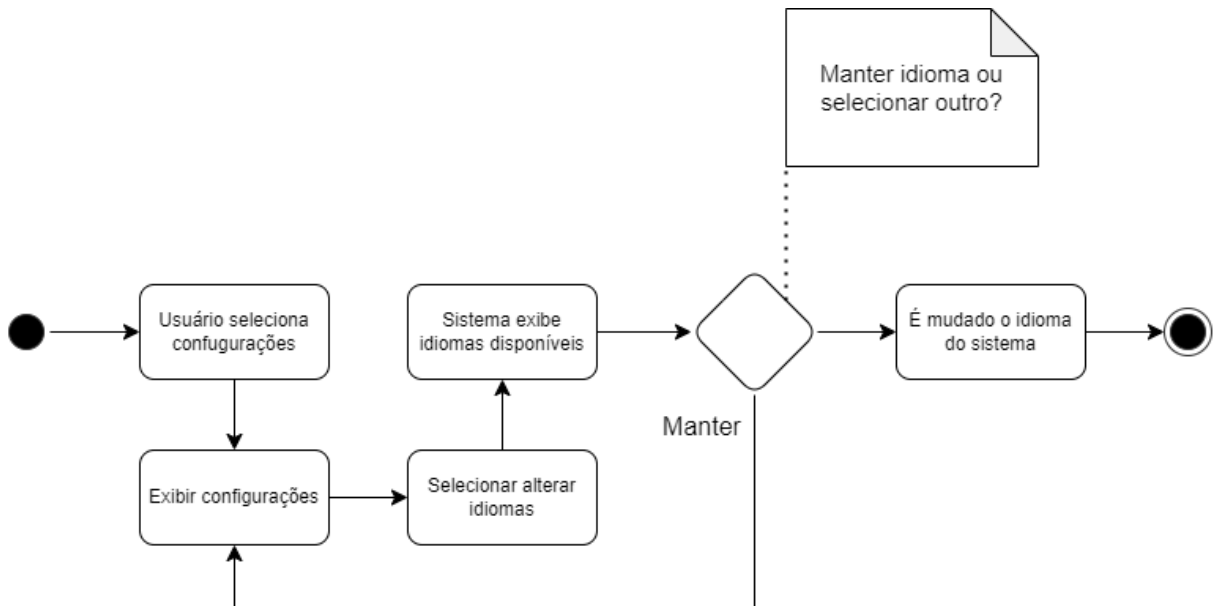


7.9 Comparador de criptomoedas



No diagrama de atividade acima, ao usuário selecionar a exibição das criptomoedas, ele pode selecionar as criptomoedas para a exibição dos dados lado a lado na visualização das criptos.

7.10 Gerenciador de idiomas



O diagrama de atividades representa o processo de alteração de idioma no sistema. O fluxo inicia com o usuário acessando as configurações, onde ele pode visualizar as opções disponíveis. Caso o usuário escolha a opção de alterar o idioma, o sistema exibe os idiomas disponíveis. Em seguida, há uma decisão: manter o idioma atual ou selecionar outro. Se o usuário optar por mudar, o idioma do sistema é atualizado; caso contrário, ele retorna às configurações sem alterações. O fluxo é finalizado após essa decisão.

8. DIAGRAMAS DE ESTADO

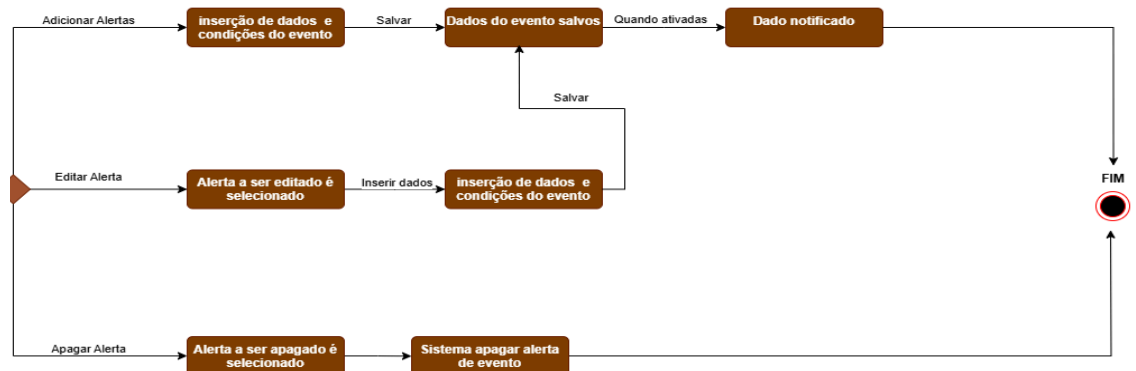
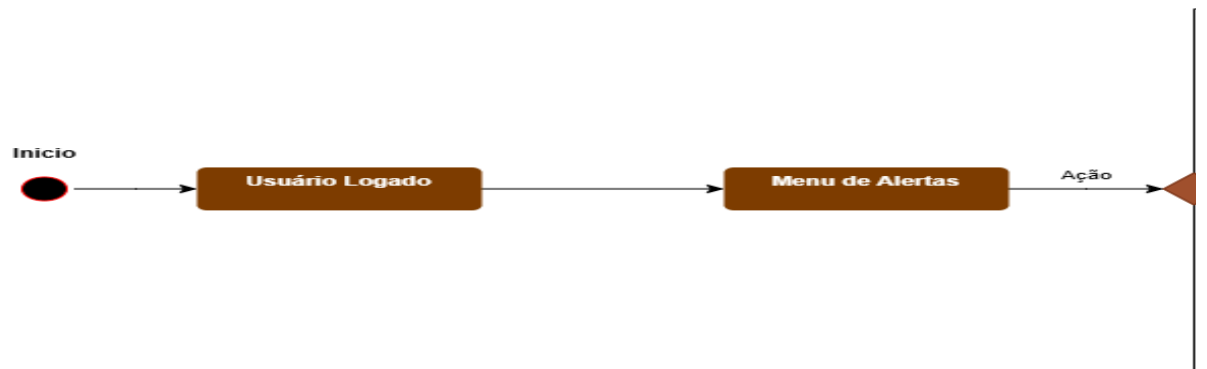
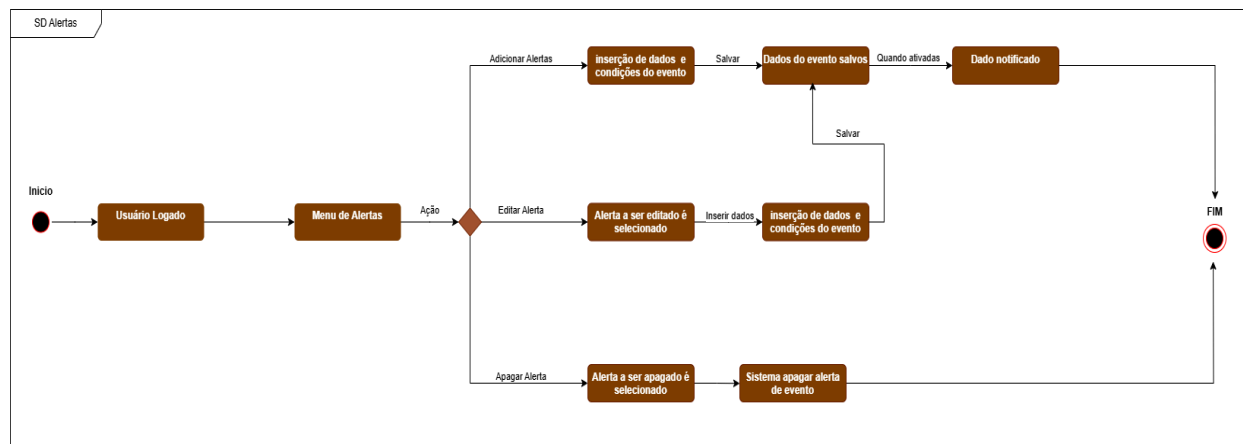
Os diagramas de estado são uma poderosa ferramenta na engenharia de software, utilizados para representar as diferentes etapas (ou estados) que um objeto ou sistema pode assumir ao longo de seu ciclo de vida. Eles descrevem como um sistema responde a eventos externos e internos, mostrando as transições entre os estados e os gatilhos que causam essas mudanças.

Esses diagramas são particularmente úteis para entender o comportamento dinâmico de componentes complexos e sistemas interativos. Eles ajudam a identificar todas as possíveis variações de estado, bem como os caminhos que podem ser seguidos dependendo das interações do usuário ou de eventos no ambiente.

Elementos de um Diagrama de Estado

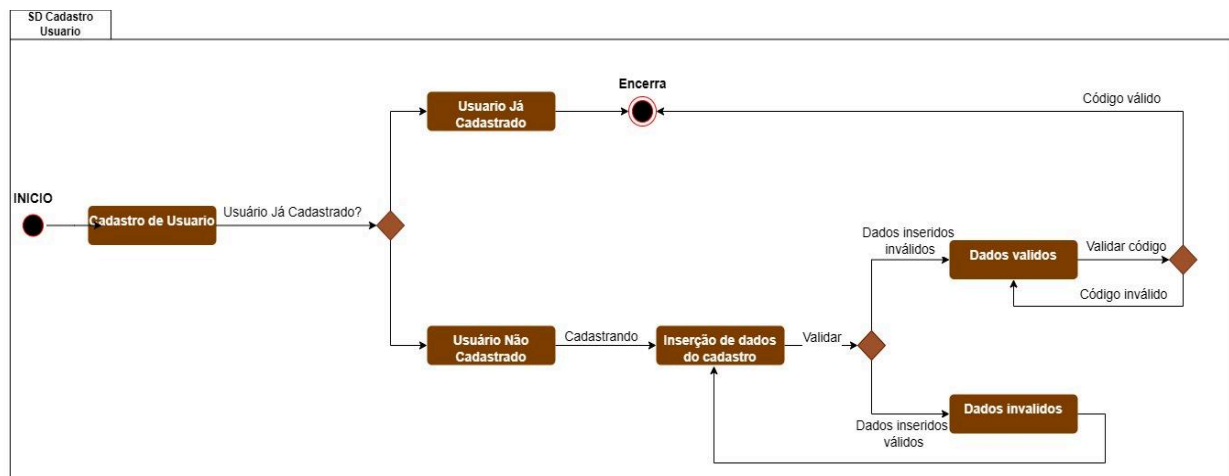
- **Estados:** Representam as condições em que o sistema ou objeto se encontra em um determinado momento. Cada estado reflete uma situação específica durante a execução.
- **Transições:** Conectam os estados e indicam as mudanças que ocorrem quando um evento específico é disparado.
- **Eventos:** Ações ou ocorrências que disparam a transição de um estado para outro.
- **Ações:** Atividades que ocorrem como resultado de uma transição ou dentro de um estado específico.
- **Estado Inicial:** O ponto de partida do ciclo de vida do sistema ou objeto.
- **Estado Final:** A condição de encerramento de um ciclo, se aplicável.

8.1. Alertas



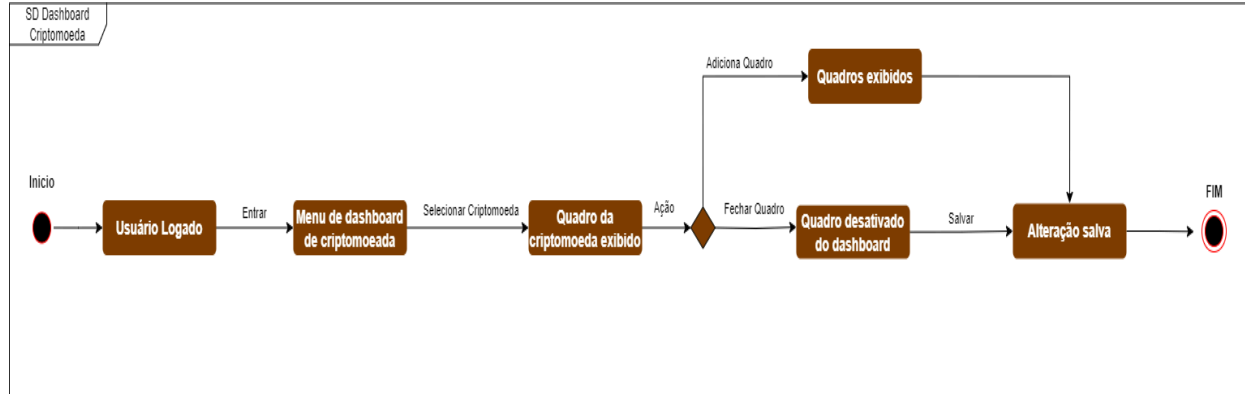
Com o usuário no estado logado podemos acessar o menu de alertar, podendo inserir, adicionar ou apagar uma alerta, em sequência o alerta é selecionado, por fim entra em seu estado final, que deve ser ou excluído ou editado ou cadastrado, encerrado o processo

8.2. Cadastrar usuário



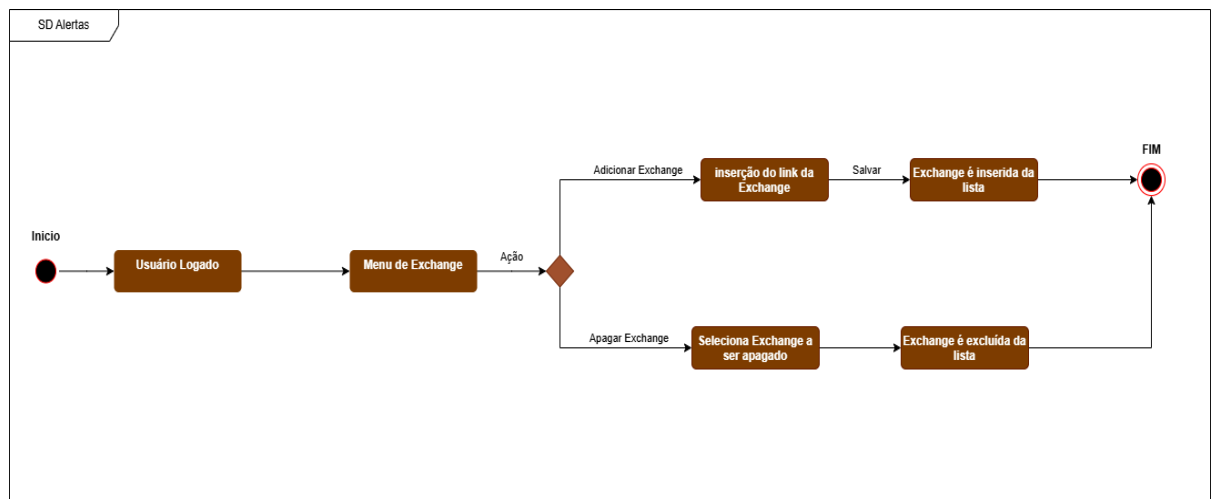
Como o usuário no estado deslogado, o sistema verifica se o cadastro já exista no sistema, se sim o sistema encerra, se não o usuário inicia o processo de cadastro, e vai para o estado não-cadastrado, em seguida o usuário insere seus dados, que são validados pelo sistema se válidos o usuário será submetido a uma verificação de duas etapas (visto diagrama 8.10), se inválido o usuário retorna para o passo anterior.

8.3. Dashboard



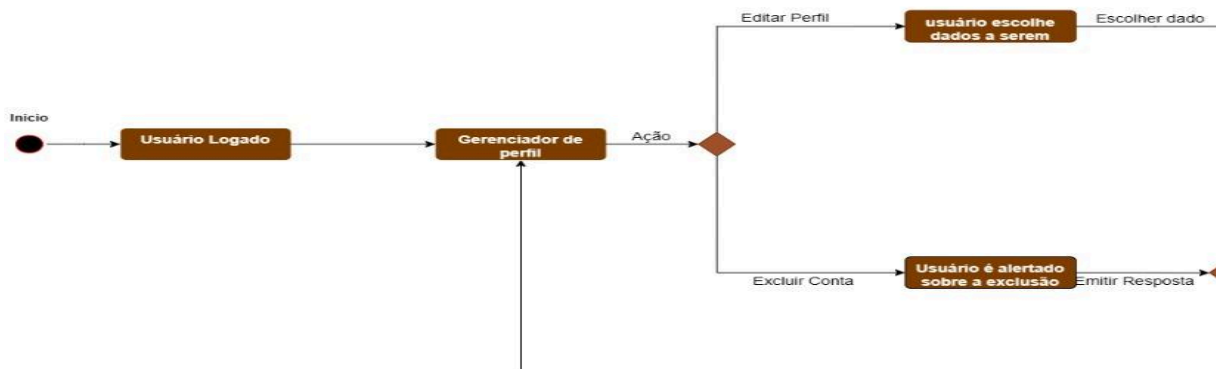
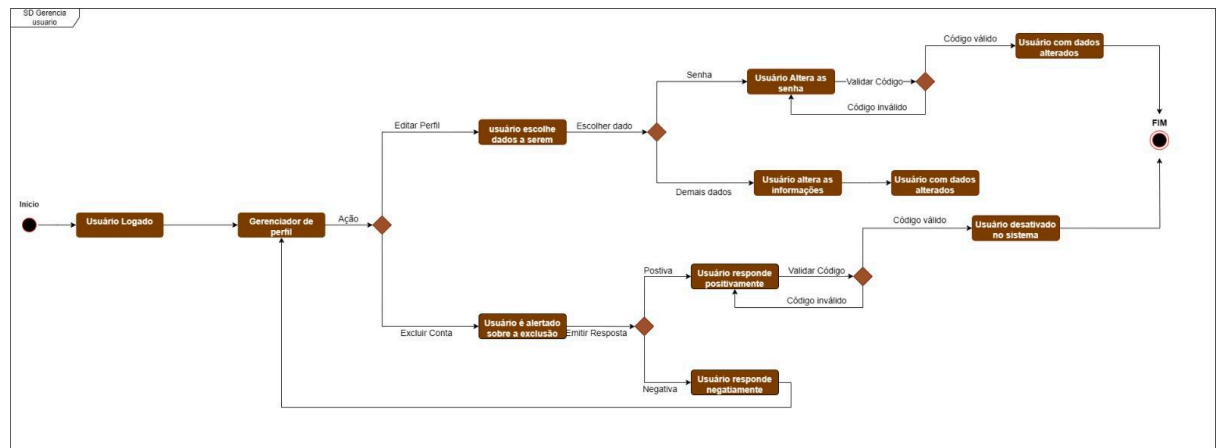
Com o usuário no estado logado, entra-se no menu de criptomoeda, após isso o quadro da criptomoeda selecionada é exibido, podendo adicionar ou desativar, encerrando o processo

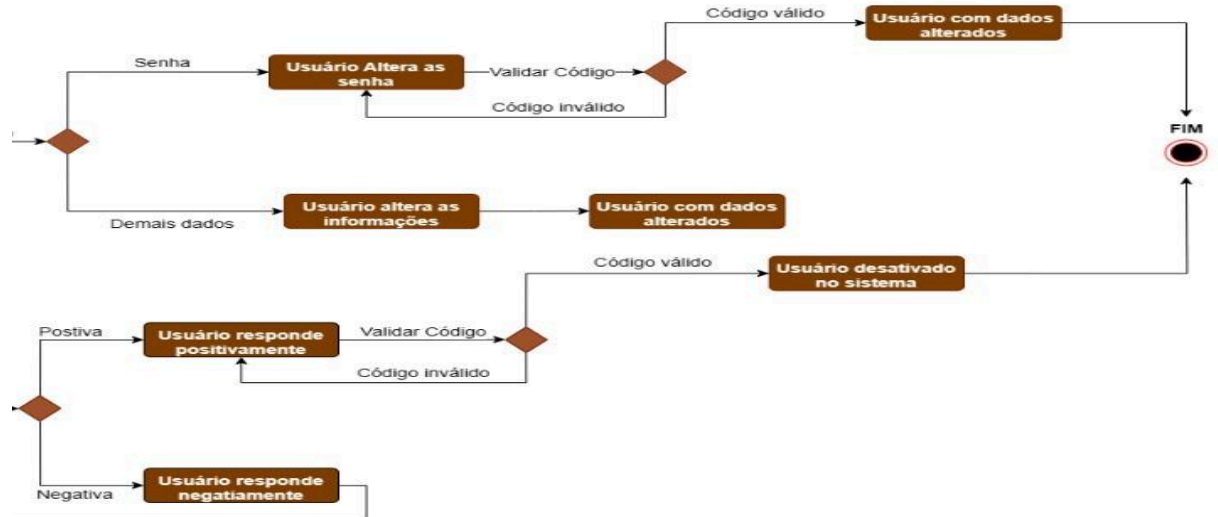
8.4. Exchange



Com o usuário logado, o usuário entra no menu exchange, podendo inserir ou apagar uma exchange, assim a mesma pode ser inserida ou apagada da lista finalizando o sistema

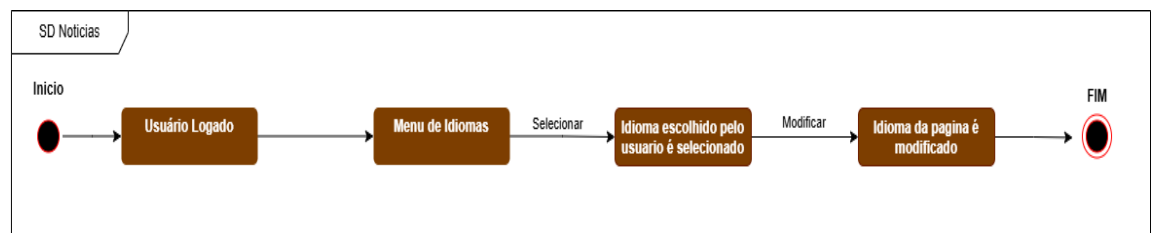
8.5. Gerenciar usuário





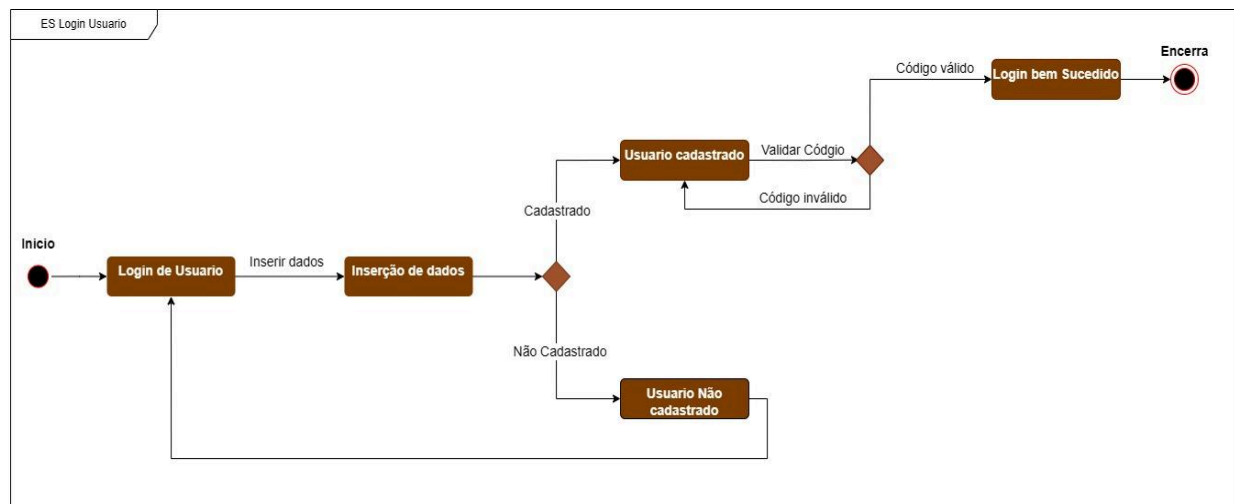
Com o usuário Logado, usuário entra o no gerenciador de perfil, podendo alterar a informações ou excluir a conta, se ele selecionar o dados a serem alterados ele mandará os dados ao sistema e o sistema vai alterá-los, se no caso este dado for um senha o sistema pedirá uma verificação de duas etapas, se caso escolher a exclusão da conta o sistema vai pedir outra verificação e assim desativar a conta.

8.6. Idioma



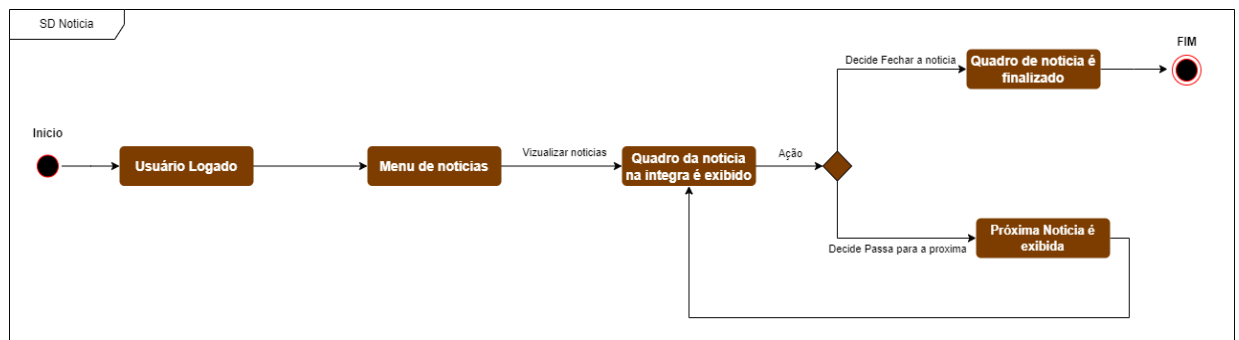
Com o usuário logado, o usuário vai abrir o menu de idiomas, selecionar o idioma escolhido, o idioma é modificado no site e encerra o sistema

8.7. Login



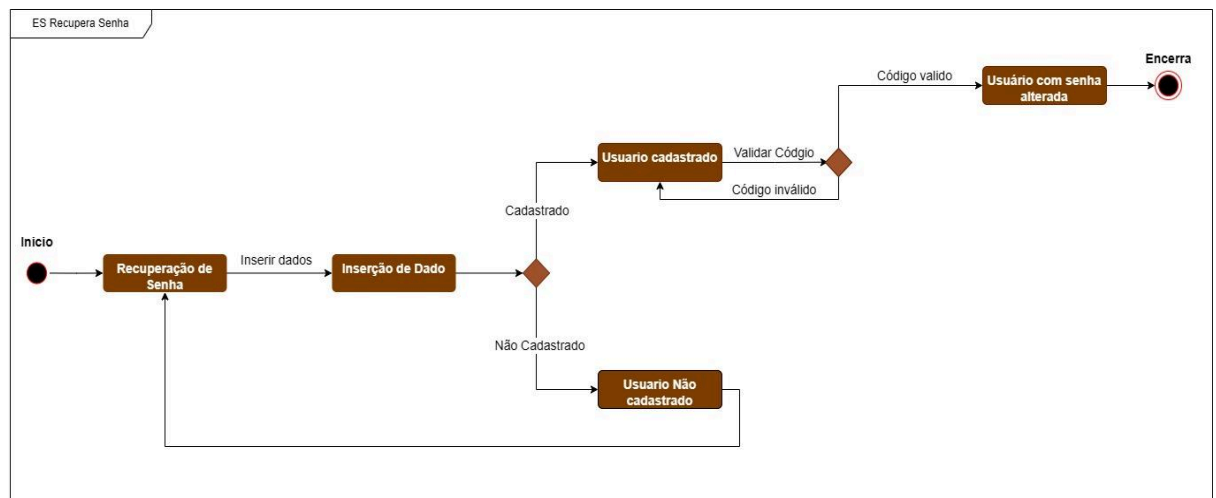
Com o Usuário deslogado, ele entra no login de usuário, insere os dados, se os dados existem no sistema inicia uma verificação de duas etapas, efetuando o login, caso o contrario retorno ao passo anterior, se o usuário não cadastrado retorna a usuário deslogado

8.8. Notícia



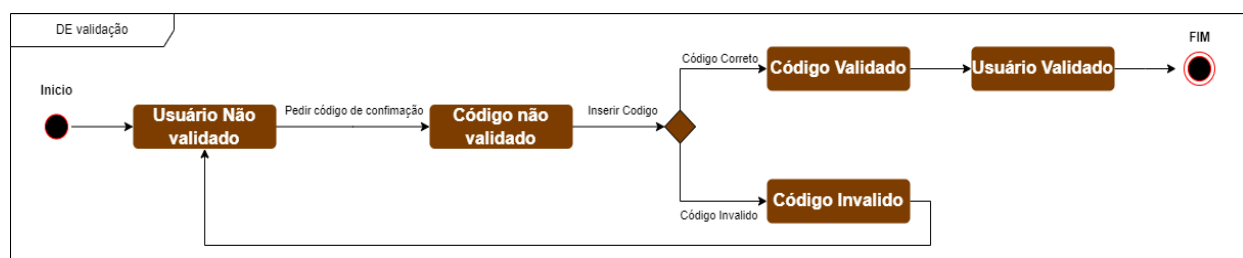
Com usuário logado , ele entra no menu de notícias, interage com os quadros de notícia, se quiser abrir outro quadro o quadro é aberto se desejar passar para a próxima notícia retorna o passo anterior

8.9. Recuperar senha



Usuário deslogado entra em recuperação de senha, insere os dados, se o usuário cadastrado, verifica as duas etapas, se não retorna ao passo anterior, se o código das duas etapas estiver certo usuário altera a senha e encerra o sistema

8.10. Validação de duas etapas





Usuário logado ou deslogado, onde o sistema pede um código para validação de duas etapas, o código é escrito, se certo o usuário é validado, se incorreto o retorna ao passo anterior

9. TECNOLOGIAS

No projeto de desenvolvimento de software, utilizamos diversas tecnologias e softwares para garantir eficiência e robustez. O framework Express e o REACT foram de suma importância para uma aplicação rápida e escalável. Para o ambiente de desenvolvimento, usamos o Visual Studio Code (VSCode), que proporcionou uma interface amigável e recursos avançados de edição de código. Entre as bibliotecas mais importantes usadas estão: bcrypt, usada para encriptar as senhas e garantir segurança, cookie-parser, usada para atribuir um token ao usuário, cors, usada para conexão da API com o frontend e o mongoose, que faz a conexão/interação do código com o banco de dados no MongoDB. Essas ferramentas e bibliotecas foram fundamentais para o sucesso do projeto, proporcionando uma base sólida e funcionalidades avançadas para o desenvolvimento da aplicação.

10. ESCOPO FINAL

a prototipação do sistema incluirá:

1. Tela de login: Autenticação de Usuários para acessos restritos a usuários logados



A screenshot of a login interface. At the top center is a logo consisting of a blue circle with a white geometric design inside. Below the logo is the word "Login" in a bold, blue, sans-serif font. Underneath, there are two input fields: "Email" with the text "emerson.muniz@discente.ufma.br" and "Senha" with three dots indicating a password. At the bottom is a dark blue button with the word "Entrar" in white.

2. Interface de cadastro: cadastrar novos usuários

A screenshot of a registration interface. At the top center is the same logo as the login page. Below it is the word "Cadastro" in a bold, blue, sans-serif font. Underneath are five input fields: "Nome", "Email" (with "emerson.muniz@discente.ufma.br"), "Senha" (with three dots), and "Telefone". At the bottom is a dark blue button with the word "Cadastrar" in white.

3. Interface de inicial: visualizar notícia



Principais Notícias



Bitcoin alcança novo recorde de preço em 2025

O Bitcoin atingiu um novo recorde histórico.



Ethereum 2.0 revoluciona o mercado de contratos inteligentes

Com o lançamento do Ethereum 2.0, a rede



Criptomoedas são cada vez mais aceitas por grandes empresas

Várias empresas de renome começaram a



China planeja criar sua própria moeda digital

O governo chinês revelou planos de lançar sua própria moeda digital, a 'digital yuan', que

 Sereno

[Cadastre-se](#) [Login](#)

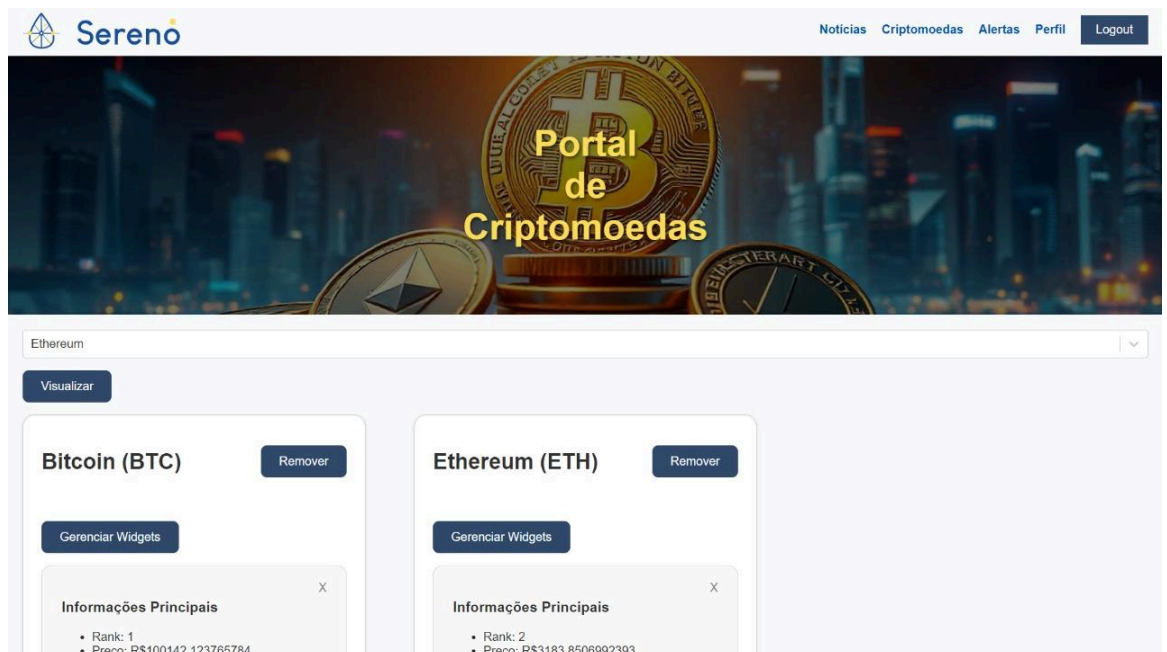
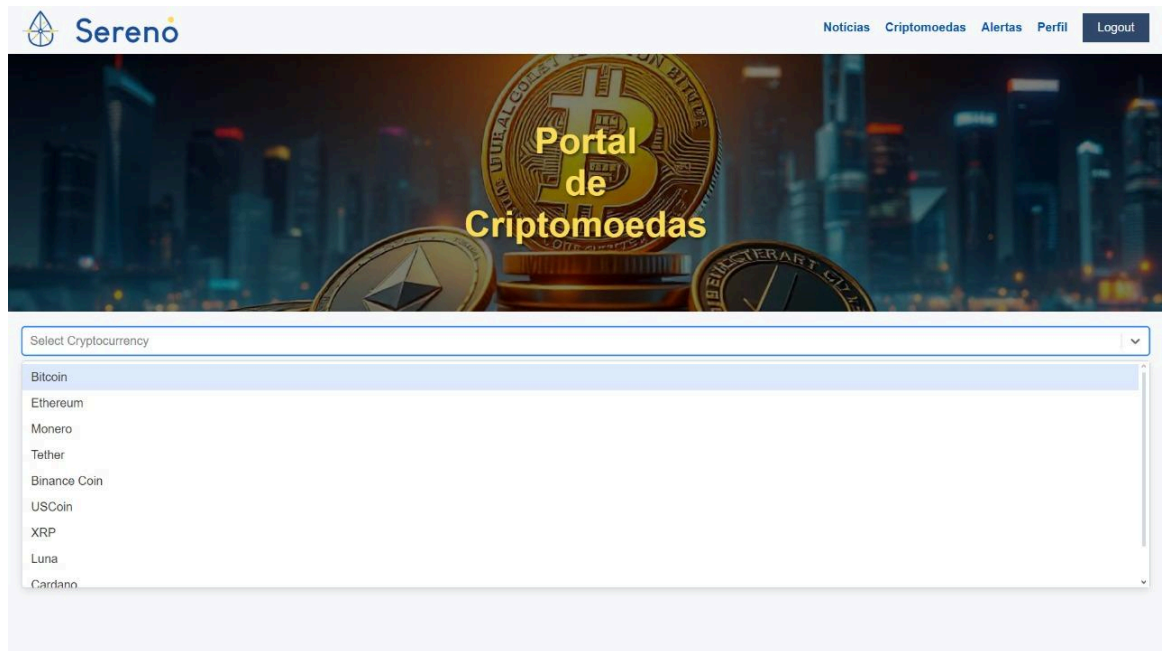
Bitcoin alcança novo recorde de preço em 2025

Publicado em: 21-01-2025



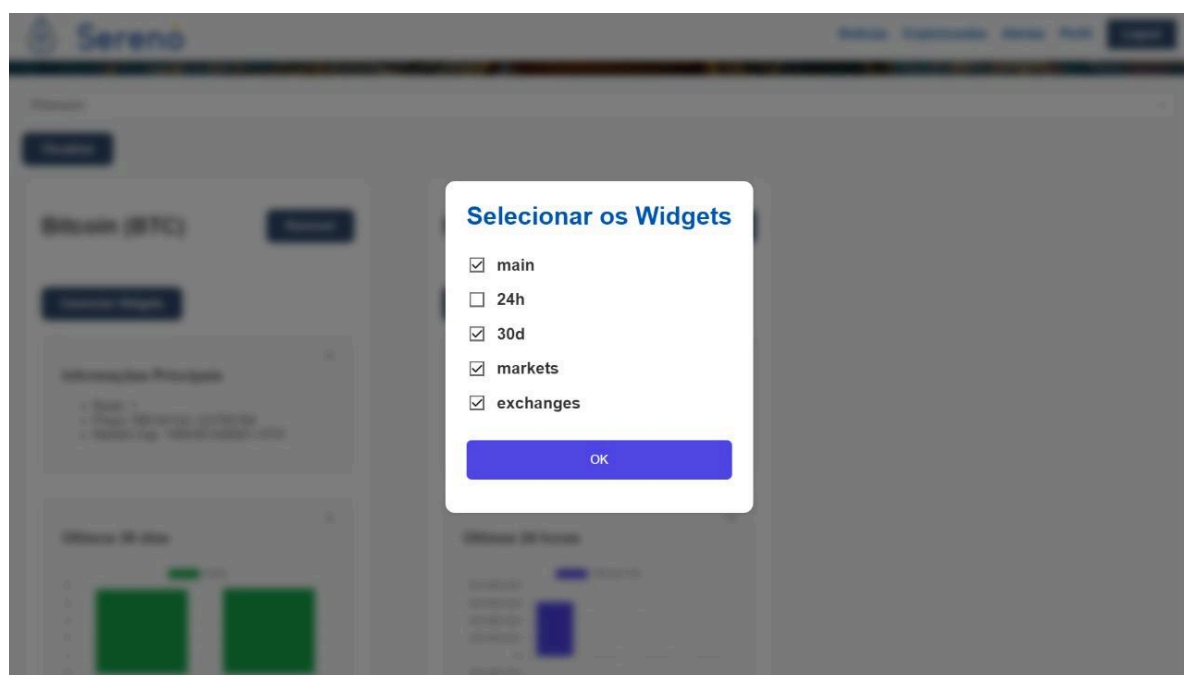
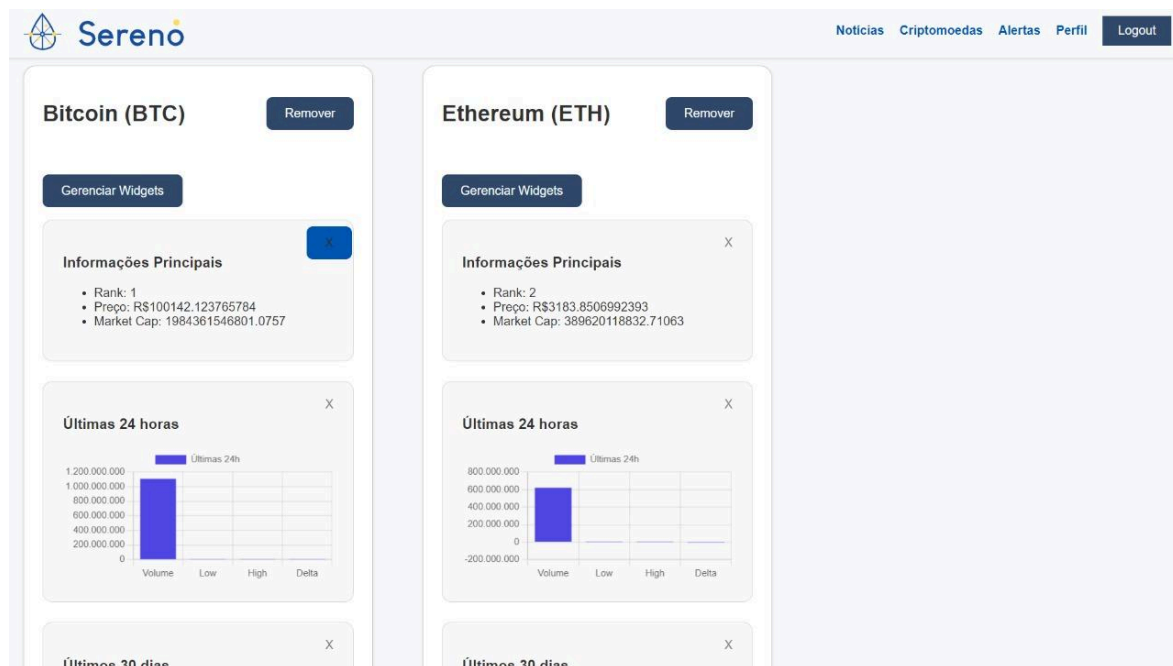
O Bitcoin tem mostrado uma recuperação impressionante após o colapso de 2023, com investidores voltando à criptomoeda. Especialistas apontam que o aumento de interesse por parte de investidores institucionais tem impulsionado o preço da moeda digital. As expectativas para 2025 são otimistas, com muitos prevendo que o Bitcoin pode continuar sua ascensão. No entanto, ainda há incertezas, como os reguladores globais e as flutuações do mercado. Algumas autoridades financeiras têm expressado preocupação com a volatilidade do Bitcoin e suas implicações no sistema financeiro tradicional. Apesar disso, muitos acreditam que o Bitcoin está se consolidando como um ativo seguro, especialmente em tempos de crise econômica, e que sua popularidade deve continuar crescendo.

4. Interface de criptomoedas: adicionar, excluir telas e criptomoedas.





UNIVERSIDADE FEDERAL DO MARANHÃO - UFMA
CENTRO DE CIÊNCIA EXATAS E TECNOLOGIA - CCET
COORDENAÇÃO DA ENGENHARIA DA COMPUTAÇÃO - EEC
PROJETO E DESENVOLVIMENTO DE SOFTWARE



5. Interface de alertas, adicionar ou retirar alertas



The screenshot shows a modal window titled "Adicionar Alerta" (Add Alert) overlaid on a blurred background of the Sereno application interface. The modal contains three input fields: "Criptomoeda:" (Cryptocurrency) with a dropdown menu showing "Bitcoin", "Valor de Referência:" (Reference Value) with a text input containing "123", and "Condição:" (Condition) with a dropdown menu showing "Menor que" (Less than). At the bottom of the modal are two buttons: "Adicionar" (Add) and "Fechar" (Close).

The screenshot shows a modal window titled "Editar Alerta" (Edit Alert) overlaid on a blurred background of the Sereno application interface. The modal contains three input fields: "Criptomoeda:" (Cryptocurrency) with a dropdown menu showing "Bitcoin", "Preço de Referência:" (Reference Price) with a text input containing "123", and "Condição:" (Condition) with a dropdown menu showing "Maior que" (Greater than). At the bottom of the modal are two buttons: "Salvar" (Save) and "Cancelar" (Cancel).

6. Interface de gerenciamento de usuário, pode-se deletar o perfil do usuário



Perfil do Usuário

name:

telephone:

email:

Apagar Perfil

Tem certeza que deseja excluir o perfil?



11. CONCLUSÃO

O presente projeto de desenvolvimento de um portal de criptomoedas foi concebido tendo em vista o cenário dinâmico e competitivo do mercado de ativos digitais. Com o objetivo de atender às principais demandas dos usuários, desde investidores iniciantes até profissionais experientes, o sistema foi projetado com base em um levantamento criterioso de requisitos funcionais e não-funcionais, assegurando a implementação de soluções que tratam problemas recorrentes enfrentados nesse ambiente.

Entre as funcionalidades mais relevantes, destacam-se a centralização do monitoramento e gerenciamento de portfólios, a disponibilização de análises de mercado em tempo real com insights técnicos claros e acessíveis, a possibilidade de comparar diferentes ativos e a garantia de segurança e privacidade de dados. Adicionalmente, o sistema foi projetado para integrar ferramentas de personalização da experiência do usuário, permitindo que cada investidor adapte o portal às suas necessidades e preferências.

Com o suporte de técnicas de modelagem, como diagramas de casos de uso, classes, atividades e sequência, foi possível estruturar e organizar o sistema de forma robusta e transparente, facilitando a comunicação entre as equipes de desenvolvimento, stakeholders e usuários finais. Essas ferramentas foram essenciais para detalhar os fluxos de interação, os relacionamentos entre as entidades e os processos temporais envolvidos, contribuindo para a clareza e consistência do projeto em todas as suas fases.

Em suma, este sistema visa não apenas solucionar as principais dificuldades dos usuários, mas também elevar o padrão de gerenciamento de ativos digitais, promovendo agilidade, segurança e eficiência. A combinação de técnicas de



modelagem, inovação e foco nas necessidades dos usuários posiciona este projeto como uma contribuição significativa para o mercado de criptomoedas, alinhando-se às demandas de um setor em constante evolução e crescimento.

Reconhecimentos e Direitos Autorais

@autor: Antonio Neto Moura Melo, Alexsandro Nogueira Costa Filho, Carla Sofia Santos Ribeiro, Emerson Paulo Pinheiro Muniz, Marcos Davi Taveira de Sousa

@contato: alexsandro.nogueira@discente.ufma.br, antonio.nmm@discente.ufma.br, carla.sofia@discente.ufma.br, emerson.muniz@discente.ufma.br, marcos.davi@discente.ufma.br

@data última versão: 28/01/2025

@versão: 1.0

@outros repositórios: https://github.com/ememuniz/portal_de_criptos.git

@Agradecimentos: Universidade Federal do Maranhão (UFMA), Professor Doutor Thales Levi Azevedo Valente, e colegas de curso.

Copyright/License

Este material é resultado de um trabalho acadêmico para a disciplina "PROJETO E DESENVOLVIMENTO DE SOFTWARE", sob a orientação do professor Dr. THALES LEVI AZEVEDO VALENTE, semestre letivo 2024.2, curso Engenharia da Computação, na Universidade Federal do Maranhão (UFMA). Todo o material sob esta licença é software livre: pode ser usado para fins acadêmicos e comerciais sem nenhum custo. Não há papelada, nem royalties, nem restrições de "copyleft" do tipo GNU. Ele é licenciado sob os termos da Licença MIT, conforme descrito abaixo, e, portanto, é compatível com a GPL e também se qualifica como software de código aberto. É de domínio público. Os detalhes legais estão abaixo. O espírito desta



licença é que você é livre para usar este material para qualquer finalidade, sem nenhum custo. O único requisito é que, se você usá-los, nos dê crédito.

Licenciado sob a Licença MIT. Permissão é concedida, gratuitamente, a qualquer pessoa que obtenha uma cópia deste software e dos arquivos de documentação associados (o "Software"), para lidar no Software sem restrição, incluindo sem limitação os direitos de usar, copiar, modificar, mesclar, publicar, distribuir, sublicenciar e/ou vender cópias do Software, e permitir pessoas a quem o Software é fornecido a fazê-lo, sujeito às seguintes condições:

Este aviso de direitos autorais e este aviso de permissão devem ser incluídos em todas as cópias ou partes substanciais do Software.

O SOFTWARE É FORNECIDO "COMO ESTÁ", SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO MAS NÃO SE LIMITANDO ÀS GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO FIM E NÃO INFRINGÊNCIA. EM NENHUM CASO OS AUTORES OU DETENTORES DE DIREITOS AUTORAIS SERÃO RESPONSÁVEIS POR QUALQUER RECLAMAÇÃO, DANOS OU OUTRA RESPONSABILIDADE, SEJA EM AÇÃO DE CONTRATO, TORT OU OUTRA FORMA, DECORRENTE DE, FORA DE OU EM CONEXÃO COM O SOFTWARE OU O USO OU OUTRAS NEGOCIAÇÕES NO SOFTWARE.

Para mais informações sobre a Licença MIT: <https://opensource.org/licenses/MIT>