DESENVOLVIMENTO DE APLICATIVO GERENCIADOR DE CRIPTOMOEDAS

Donizete Crisostomo Barbosa[1], Debora Pelicano Diniz[1]

[1]Faculdade de Tecnologia de FATEC Ribeirão Preto (FATEC)

Ribeirão Preto, SP – Brasil

donizete.barbosa01@fatec.sp.gov.br,
debora.diniz2@fatec.sp.gov.br

Resumo. *Este artigo descreve o projeto que teve como objetivo a construção de um aplicativo de gerenciamento de criptomoedas para auxiliar pessoas interessadas em monitorar seus criptoativos, sem a necessidade de estar constantemente verificando sua carteira de moedas. O aplicativo foi criado de acordo com os padrões mais eficazes da Engenharia de Software, e se encontra atualmente na fase inicial de testes beta, sem que tenha havido tempo hábil para analisar os feedbacks dos usuários envolvidos na etapa de testes.*

Abstract. *This article describes the project that aimed to build a cryptocurrency management application to help people interested in monitoring their crypto assets without the need to constantly check their coin wallet. The application was created in accordance with the most effective Software Engineering standards, and is currently in the initial beta testing phase, without having had time to analyze feedback from users involved in the testing stage.*

1. Introdução

O atual momento em que vivemos na procura por Bitcoin e a crescente valorização das moedas digitais, em que a pessoa é a única detentora do seu dinheiro digital, traz riscos ao usuário de criptoativos de serem hackeados ou, acidentalmente, perderem suas criptomoedas, caso não sejam cautelosos.

Segundo Ulrich (2014) Bitcoin é uma moeda digital *peer-to-peer* (par a par ou, simplesmente, de ponto a ponto), de código aberto, que não depende de uma autoridade central. Entre muitas outras coisas, o que faz o Bitcoin ser único é o fato de ele ser o primeiro sistema de pagamentos global totalmente descentralizado.

Diversamente das moedas tradicionais, as criptomoedas, não são emitidas ou controladas por um governo centralizado, são totalmente digitais e não necessitam de um terceiro intermediário para a sua emissão e para realização de suas transações *on-line*. Atualmente, essas moedas se tornaram populares, por trazerem dois aspectos importantes: 1) elas não dependem de empresas, bancos ou governos centrais para a sua emissão, distribuição e transferência; 2) o segundo aspecto diz respeito à segurança, tanto da identidade de quem as possuem, quanto das transações (ULRICH, 2014).

O presente trabalho tem como objetivo a construção de um aplicativo de gerenciamento de criptomoedas para auxiliar pessoas interessadas em monitorar seus

criptoativos, sem a necessidade de estar, constantemente, verificando sua carteira de moedas. Sua finalidade é registrar e monitorar as atividades do portfólio do usuário, eliminando a necessidade de verificação constante da carteira.

Assim, o trabalho foi organizado da seguinte forma: na Seção 2 são descritos o referencial teórico, explicando o que é Bitcoin, uma carteira de criptomoeda, e Exchange; na Seção 3 é apresentada a metodologia e as ferramentas utilizadas; na Seção 4 são descritos a documentação e o desenvolvimento do projeto; e, por fim, na Seção 5 está a conclusão e os próximos passos para melhorias.

2. Criptomoedas

Criptomoedas são sistemas de dinheiro eletrônico ponto a ponto, que é algo maior que uma moeda ou meio de pagamento, pois além de compreender a emissão de novas unidades, como faz a Casa da Moeda, também é um sistema de controle e distribuição. Tudo isso ocorre de forma independente e sem uma autoridade central, baseado somente em códigos matemáticos (PELLINI, 2020).

O autor também destaca que o Bitcoin foi a primeira criptomoeda do mundo, criada em 2008, mas só entrou em funcionamento em 3 de janeiro de 2009. Para funcionar, usa um sistema criptografado (por isso o uso do prefixo "cripto") em uma rede distribuída, que é a *blockchain* (que, na verdade, foi criada por causa do Bitcoin).

O lugar mais comum para comprar Bitcoin são as corretoras de criptomoedas, também conhecidas por ***Exchanges***. Existe uma série de corretoras de criptomoedas nas quais é possível comprar e vender Bitcoins, e a melhor maneira de descobrir uma confiável é por meio de pesquisas na internet. Também é possível comprar diretamente de pessoas que possuem Bitcoins. Porém, como a tecnologia ainda é muito nova, realizar essa transação ainda não é algo amigável para o consumidor comum e é difícil encontrar pessoas de confiança (PELLINI, 2020).

Apesar dos benefícios que ele apresenta, o Bitcoin tem algumas desvantagens que usuários em potencial devem levar em consideração. Houve significativa volatilidade no preço ao longo de sua existência. Novos usuários correm o risco de não proteger devidamente suas carteiras ou de, até mesmo, acidentalmente apagar seus bitcoins, caso não sejam cautelosos. Além disso, há preocupações sobre se *hackers* podem, de alguma forma, comprometer a economia Bitcoin (ULRICH, 2014).

De acordo com o Infomoney (2022) quando uma pessoa transfere criptomoedas para outra, esses ativos ficam guardados na blockchain. A blockchain é o famoso banco de dados descentralizado que nasceu com o Bitcoin (BTC) no final de 2008. As carteiras digitais são os softwares e os dispositivos físicos que dão aos usuários acesso a esses ativos digitais armazenados nesse sistema.

Além disso, elas também permitem o envio de moedas digitais sem a necessidade de intermediários. Na prática, as carteiras são semelhantes a contas bancárias, mas com uma grande diferença: é o dono da carteira o responsável pela posse e segurança de seus ativos, não o banco. Depois que uma carteira é criada, imediatamente é gerada uma *seed* (semente, em português), que é uma sequência de 12 a 24 palavras (em inglês), que funciona como uma senha de recuperação de sistema. A s*eed* deve ser bem guardada e não se deve mostrar para ninguém, pois, caso seja perdida, os seus fundos também serão.

Logo após a criação da *seed*, a carteira libera uma chave privada, uma chave pública e um endereço. A chave privada é como a senha de sua conta bancária. Não se deve passá-la para ninguém, pois é por meio dela que se tem acesso às criptomoedas. A chave pública, por sua vez, é como sua conta bancária. Suas criptomoedas ficam guardadas nela e só podem ser liberadas com a chave privada. Como é pública, qualquer um pode ver, mas ninguém pode movimentar os fundos dela. Por fim, o endereço é como o número da sua conta no banco – ou pode ser comparado também com seu PIX. É esse endereço, derivado da chave pública, que deve ser informado em transferências de criptomoedas. Ele é basicamente uma sequência alfanumérica como, por exemplo, 1G2bAat5x1HUXrCPQbtMDqcw7o5MNn5owX.

3. Metodologia e ferramentas

Para o desenvolvimento do trabalho aqui apresentado, inicialmente foi realizada a etapa de Engenharia de Requisitos para coletar os requisitos do sistema, que, de acordo com Sommerville (2011), são as descrições do que o sistema deve fazer, o que os serviços oferecem e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada e são frequentemente classificados como requisitos funcionais (declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações; em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer) e requisitos não funcionais (que são restrições aos serviços ou funções oferecidos pelo sistema, incluindo restrições de tempo, restrições no processo de desenvolvimento e restrições impostas pelas normas, e, muitas vezes, aplicam-se ao sistema como um todo). Essa atividade foi realizada por meio de entrevistas com usuários atuantes no mercado de criptomoedas e pelo estudo de aplicativos similares.

Para fazer a modelagem funcional do aplicativo utilizou-se a UML (*Unified Modeling Language*, em português Linguagem de Modelagem Unificada) que de acordo com Sommerville (2011), possui vários diagramas que auxiliam no entendimento de um sistema, mas, nessa aplicação foram utilizados apenas: (i) Diagrama de Casos de Uso, que mostra as interações entre um sistema e seu ambiente; (ii) Diagrama de Classe, que mostra as classes do sistema e as associações entre elas; e (iii) Diagramas de Sequência, que mostram as interações entre os atores e o sistema, e entre os componentes do sistema. Os digramas foram criados com a ferramenta Astah, ferramenta específica para modelagem de UML (ASTAH, 2023).

Para elaboração do protótipo foi escolhido o **Figma**, que é um editor gráfico de vetor e prototipagem de projetos de design baseado, principalmente, no navegador web, com ferramentas *offline* adicionais para aplicações desktop para GNU/Linux, macOS e Windows (FIGMA, 2023). O Figma proporcionou uma colaboração intuitiva e em tempo real das telas, facilitando o processo de prototipagem.

Para a etapa da codificação foi escolhida a linguagem de programação **Dart**, que é uma linguagem otimizada para cliente para desenvolver aplicativos rápidos em qualquer plataforma, cujo objetivo é oferecer a linguagem de programação mais produtiva para desenvolvimento multiplataforma, combinada com uma plataforma de execução flexível para estruturas de aplicativos (DART, 2023), e o framework de desenvolvimento **Flutter**, que foi desenvolvido pela Google para criar aplicativos multiplataforma bonitos e

compilados nativamente a partir de uma única base de código, e utiliza a linguagem Dart para programação, pode ser executado para aplicações, desktop, web e dispositivos móveis Android e IOS, sua estrutura e baseado em Widgets (FLUTTER, 2023). Foi utilizado o **Visual Studio Code**, que é um editor de código-fonte leve, executado em sua área de trabalho e disponível para Windows, macOS e Linux e possui extensões para outras linguagens e tempos de execução (como C++, C#, Java, Python, PHP, Go, .NET) (VISUAL STUDIO CODE, 2023).

O Banco de Dados escolhido foi o **Firebase**, que fornece uma variedade de recursos, que auxiliam no desenvolvimento e serviços como: Authentication, o Firestore e a integração com o Google Analytics (FIREBASE, 2023). Nessa aplicação será utilizado o **Firebase Cloud Firestore** como banco de dados e o **Authentication** para autenticar o login.

Foi utilizada a API do **CoinMarketCap** que é um conjunto de endpoints RESTful JSON de alto desempenho projetado especificamente para atender às demandas de missão crítica de desenvolvedores de aplicativos, cientistas de dados e plataformas de negócios empresariais (COINMARKETCAP, 2023). Atualmente a api conta com um uso pessoal básico, chamado de Free que disponibiliza 10 mil créditos de chamadas/mês que atende muito bem a expectativa do desenvolvimento, para gerar a cotação atual.

No momento da escrita deste artigo o projeto encontra-se na fase de testes beta, que estão sendo realizados para avaliar o desempenho e a funcionalidade do software. Durante esta etapa, o produto está sendo disponibilizado para um grupo selecionado de usuários de criptomoedas. Esse grupo está participando ativamente no processo, explorando o aplicativo em condições do mundo real e fornecendo *feedback*.

Toda a documentação está disponível no **GitHub**, que é uma plataforma para hospedar código-fonte com controle de versão usando Git, na qual programadores e usuários cadastrados podem contribuir em projetos privados ou de código aberto (GITHUB, 2023), e pode ser acessada via link: https://github.com/Doni-zete/projeto-flutter-tcc.

4. Desenvolvimento da aplicação

Nessa seção serão apresentadas as documentações geradas nas etapas do desenvolvimento do projeto.

a) Engenharia de Requisitos

A etapa de Engenharia de Requisitos foi conduzida por meio de entrevistas com usuários ativos no mercado de criptomoedas e pelo estudo de aplicativos semelhantes. O principal objetivo do aplicativo é gerenciar ativos digitais, proporcionando maior controle aos demais usuários atuantes no mercado de criptomoedas que tenham interesse em cibersegurança e criptoativos, com a ajuda de gerenciadores. Ao selecionar a cotação, campos serão apresentados para inserção, como a sigla da criptomoeda e o valor. Em todos os campos em qualquer tela que requerem valor, apenas números válidos serão aceitos, evitando entradas inválidas que possam comprometer os cálculos. Ao escolher a lista de moedas, todas as moedas adicionadas devem ser exibidas de maneira clara e organizada, permitindo aos usuários visualizar facilmente informações sobre as diferentes moedas cadastradas, como sigla e valor atualizado. Por fim, a tela do saldo total apresentará o valor consolidado de todas as criptomoedas. Além disso, o botão de sair

proporcionará uma maneira fácil e segura para os usuários encerrarem suas sessões, garantindo a privacidade e segurança das informações contidas no aplicativo. Após a análise das entrevistas e das informações coletadas por meio do estudo dos softwares similares, foi definido o Documento de Requisitos, que pode ser visto na Figura 1.

Requisitos Funcionais (RF)

Requisitos com as principais funções do aplicativo.

- Função para cadastrar novo usuário com e-mail e senha;
- Função para cadastrar criptomoeda, editar ou excluir;
- Função para exibir o saldo total;
- Função exibir uma tela com a porcentagem de cada moeda que o usuário possui.

Requisitos Não-Funcionais (RN)

Requisitos para otimizar o uso do aplicativo.

- Limite de 30s para carregar o saldo;
- Interface gráfica simples facilitando o uso do aplicativo, realizando cadastro de criptomoedas.

Requisitos de Restrições (RR)

Requisitos para utilizar os recursos do aplicativo.

- Utilizar o banco de dados Firebase Cloud Firestore para salvar de maneira persistente os dados do usuário na nuvem, e-mail, senha, moedas.

Figura 1. Documento de Requisitos do aplicativo
Fonte: Autoria pessoal

b) Modelagem funcional do sistema

Modelagem do sistema é o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema. A modelagem de sistema geralmente representa o sistema com algum tipo de notação gráfica, que, atualmente, quase sempre é baseada em notações de UML (SOMMERVILLE, 2011).

Foram criados o Diagrama de Casos de uso (apresentado na Figura 2), com o objetivo de especificar as atividades realizadas no sistema e os atores envolvidos; o Diagrama de Classes, com os atributos e métodos e os relacionamentos entre as classes (Figura 3); e dois Diagramas de Sequências – um para representar a funcionalidade do **Login** (Figura 4) e outro para representar a funcionalidade **Cadastrar** (Figura 5).

c) Codificação

Na Figura 6 está exemplificada a utilização da API. Nas linhas 4 e 5, encontram-se a chave de API e a URL base para as solicitações à API do CoinMarketCap. Na linha 8, há uma função assíncrona para obter dados da criptomoeda usando seu símbolo (coinId). Na linha 9, é construída a URL da solicitação com o símbolo da moeda e a chave de API. Na

linha 12, é feita uma solicitação do tipo GET para a API do CoinMarketCap. Na linha 14, verifica-se se a resposta da API é bem-sucedida (código 200) e se contém os dados esperados. Na linha 27, se os dados estiverem presentes, retorna os dados.

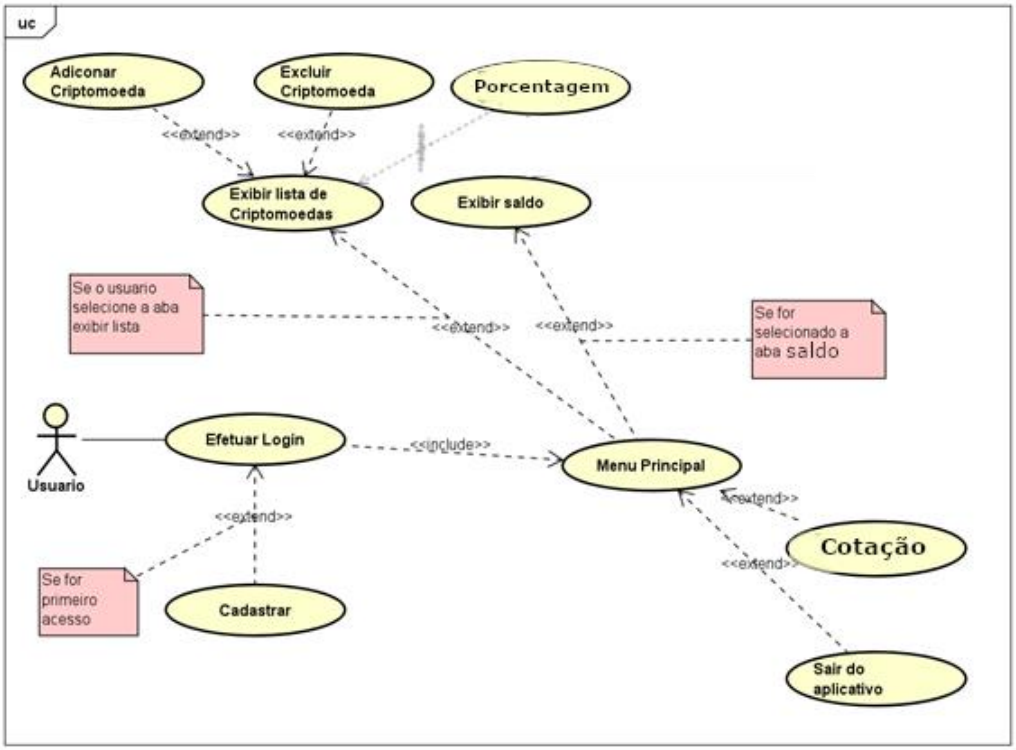


Figura 2. Diagrama de Caso de Uso do aplicativo
Fonte: Autoria pessoal

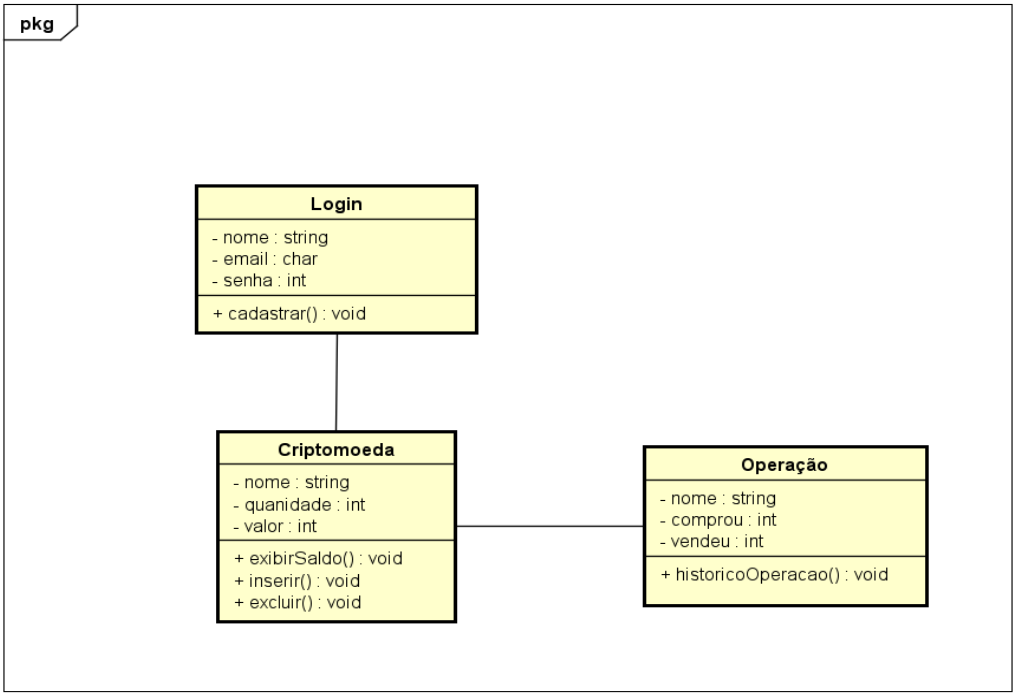


Figura 3. Diagrama de classe do aplicativo
Fonte: Autoria pessoal.

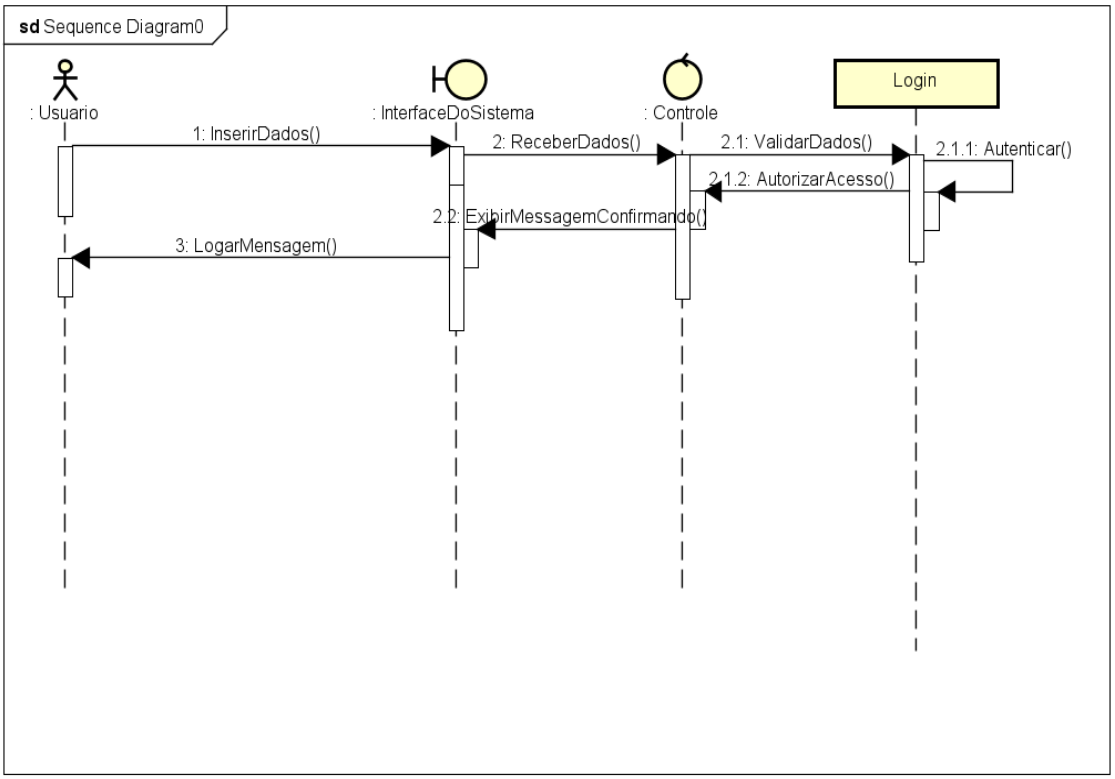


Figura 4. Diagrama de Sequência.
Fonte: Autoria pessoal.

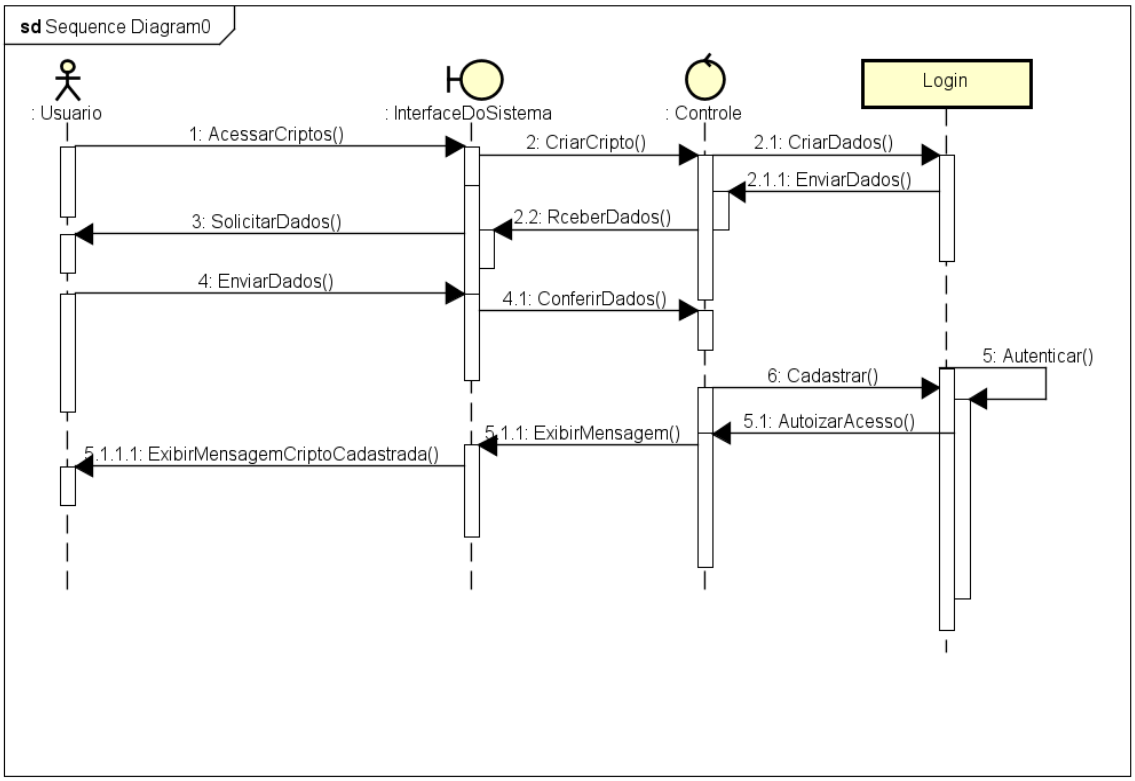


Figura 5. Diagrama de Sequência.
Fonte: Autoria pessoal.

```dart
const String apiKey = 'sua-key';
const String requestBaseUrl =
    "https://pro-api.coinmarketcap.com/v1/cryptocurrency/quotes/latest";

Future<Map<String, dynamic>?> getData(String coinId) async {
  String request =
      "$requestBaseUrl?symbol=$coinId&convert=USD&CMC_PRO_API_KEY=$apiKey";

  http.Response response = await http.get(Uri.parse(request));

  if (response.statusCode == 200) {

    Map<String, dynamic> data = json.decode(response.body);


    if (data.containsKey('data') &&
        data['data'] is Map<String, dynamic> &&
        data['data'].containsKey(coinId) &&
        data['data'][coinId] is Map<String, dynamic> &&
        data['data'][coinId].containsKey('quote') &&
        data['data'][coinId]['quote'] is Map<String, dynamic> &&
        data['data'][coinId]['quote']['USD'] is Map<String, dynamic> &&
        data['data'][coinId]['quote']['USD'].containsKey('price')) {
      return data;
    }
  }
```

Figura 6. Utilização da API CoinMarketCap
Fonte: Autoria pessoal

Na Figura 7 está representada uma função para calcular o saldo total com base nas moedas na lista. Ela itera sobre a lista de moedas e retorna o saldo total calculado.

```dart
  double calcularSaldo() {
    double saldoTotal = 0.0;
    for (Moeda moeda in listaMoeda) {
      saldoTotal += moeda.valor;
    }
    saldo = saldoTotal;
    return saldoTotal;
  }
```

Figura 7. Calcula o valor total
Fonte: Autoria pessoal

Na Figura 8 está representada uma função assíncrona para obter o valor da criptomoeda e sua imagem correspondente, e na linha 23 verifica se os dados da API foram obtidos com sucesso. Caso os dados não sejam encontrados, redefine os valores e exibe uma mensagem de erro.

```
18  void _getCoinValue() async {
19    coinId = coinController.text.toUpperCase();
20    Map<String, dynamic>? data = await getData(coinId);
21    String? coinImage = await getCoinImage(coinId);
22
23    if (data != null) {
24      setState(() {
25        coinValorInUSD = data['data'][coinId]['quote']['USD']['price'] ?? 0.0;
26        quantidade = double.parse(quantidadeController.text);
27        coinImageUrl = coinImage;
28        errorMessage = '';
29      });
30    } else {
31      setState(() {
32        coinValorInUSD = 0;
33        quantidade = 0;
34        coinImageUrl = null;
35        errorMessage = coinId.isNotEmpty
36            ? 'Valor não encontrado ou inválido.'
```

Figura 8. Obter o a cotação da moeda
Fonte: Autoria pessoal

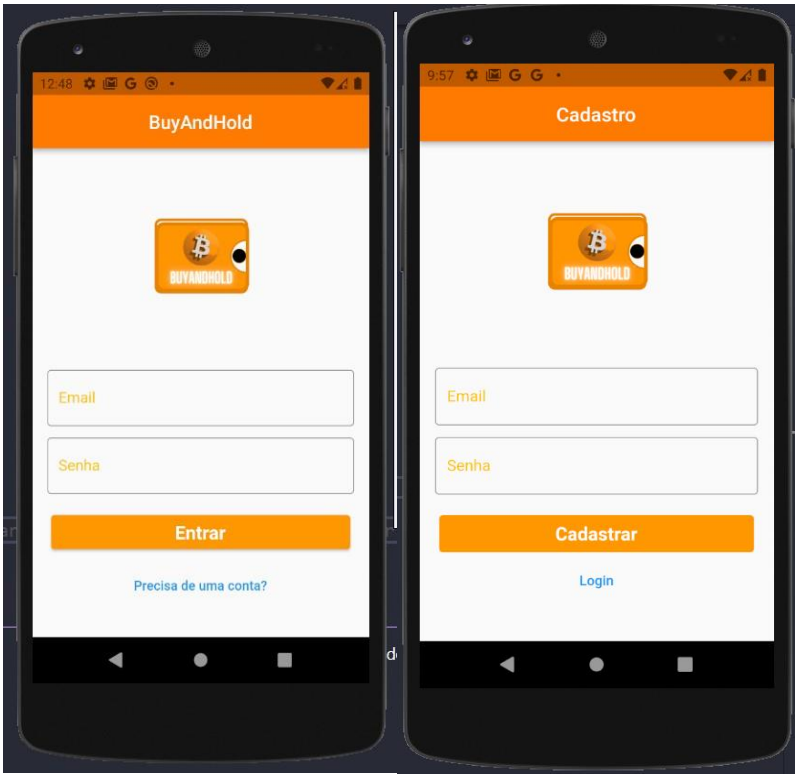
d) Protótipo

O aplicativo gerenciador de criptomoedas possui algumas funções iniciais básicas, como listar as criptomoedas, adicionar, editar, deletar, verificar a cotação da criptomoeda e a aba de saldo total.

A tela de Login (apresentada na Figura 9a) oferece a opção de fazer login ou realizar um possível Cadastro (apresentado na Figura 9b). Em caso de primeiro acesso, os dados do usuário são salvos diretamente no Firebase do Google. Após o cadastro com e-mail e senha, uma mensagem de sucesso é exibida na tela, e o usuário é redirecionado para a tela de login.

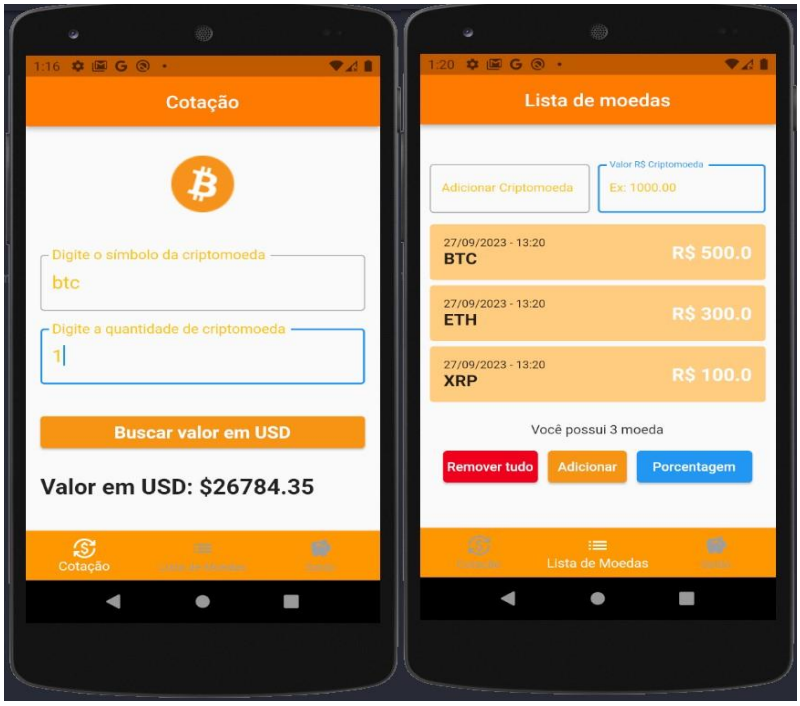
A tela de Cotação (apresentada na Figura 10a), existem dois campos: um para inserir a sigla da moeda, geralmente com 3 caracteres, e outro para inserir a quantidade de criptomoeda. É feito o cálculo, do valor atual de uma única moeda retirado da API do CoinMarketCap, multiplicado pela quantidade total de moedas digitadas. Após digitar nos campos e selecionar '**buscar valor**', a cotação atual da criptomoeda digitada é apresentada em dólar.

Na tela de Lista de moedas (apresentada na Figura 10b), possui dois campos para adicionar a criptomoeda e o valor. Assim que o botão '**adicionar**' é selecionado, a moeda adicionada é listada juntamente com a data e hora em que foi efetuado o registro. Além disso, existem dois botões: um para remover todas as criptomoedas e outro para mostrar a Porcentagem (apresentada na Figura 11a) que lista a porcentagem total e a porcentagem de cada moeda registrada no aplicativo. E, por fim, na tela de Saldo (apresentada na Figura 11b), é exibida a soma de todos os valores das criptomoedas adicionadas no aplicativo, juntamente com a opção de sair do aplicativo.



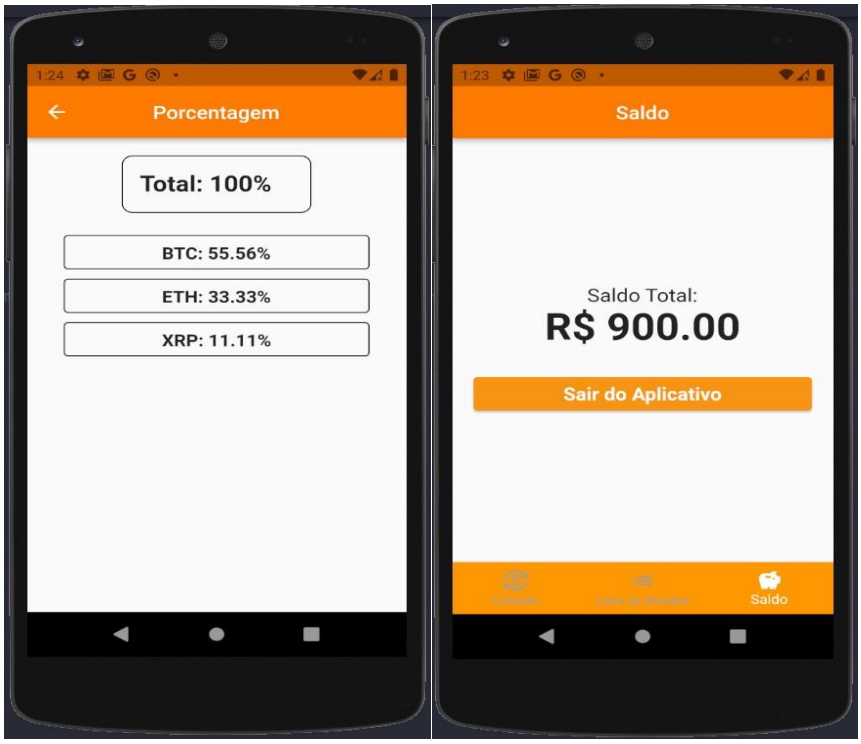
a) b)

Figura 9. a) Tela de Login; b) Tela de Cadastro
Fonte: Autoria pessoal



a) b)

Figura 10 a) Tela de Cotação; b) Tela de Lista de Moedas
Fonte: Autoria pessoal



a) b)

Figura 11 a) Tela de Porcentagem; b) Tela de Saldo
Fonte: Autoria pessoal

5. Conclusão

Neste trabalho foi apresentada a construção de um aplicativo de gerenciamento de criptomoedas para auxiliar pessoas interessadas em monitorar seus criptoativos, sem a necessidade de estar constantemente verificando sua carteira de moedas. Os resultados do desenvolvimento da aplicação foram satisfatórios, pois o aplicativo possui as funções necessárias para gerenciar as criptomoedas.

O aplicativo proporciona acesso rápido às informações sobre as criptomoedas, permitindo que o investidor monitore seu portfólio a qualquer momento e em qualquer lugar conectado à internet, utilizando apenas o e-mail e senha para recuperação de conta, caso seja necessário recuperar a senha ou em situações semelhantes. Uma das vantagens é que ele facilita a diversificação do portfólio, permitindo que os investidores gerenciem várias criptomoedas de maneira centralizada. No entanto, uma das desvantagens é que ele requer conexão com a internet para acesso, o que pode limitar a disponibilidade de informações críticas em momentos importantes. Além disso, o aplicativo ainda não conta com recursos avançados de análise técnica e gráficos detalhados que poderiam auxiliar os investidores a tomar decisões baseadas em dados. O aplicativo encontra-se atualmente na versão beta, em fase de testes de aplicações de uso, sendo utilizado por alguns usuários de criptomoedas, os quais posteriormente fornecerão suas observações sobre o desempenho do aplicativo.

Como parte do planejamento futuro, está prevista uma análise cuidadosa das respostas obtidas durante os testes beta. Quaisquer problemas ou questões identificadas serão devidamente abordados e corrigidos.

Referências

ASTAH. (2023). Disponível em: <https://astah.net/products/compare-editions/> . Acesso em: 20 Mar 2023.

COINMARKETCAP. (2023). Disponível em:<https://coinmarketcap.com/api/documentation/v1/>Acesso em: 15 set. 2023

FLUTTER. (2023). Disponível em: < https://flutter.dev/> Acesso em: 25 set. 2023.

FIGMA. (2023). Disponível em: < https://www.figma.com/> Acesso em: 15 set. 2023.

FIREBASE. (2023). Disponível em.< https://firebase.google.com/?hl=pt-br> Acesso em: 27 set. 2023.

INFOMONEY. (2022). Carteiras de criptomoedas: conheça os diferentes tipos e saiba por que são importantes. Disponível em:<https://www.infomoney.com.br/guias/carteira-de-criptomoedas/>Acesso em: 20 Out. 2023.

PELLINI, R. (2020). O futuro do dinheiro. Disponivel em <https://doceru.com/doc/n05secvv> Acesso em: 20 Out. 2023.

SOMMERVILLE, I. (2011). Engenharia de software.9 ed.

ULRICH, F. (2014). Bitcoin a moeda na era digital. Disponível em: < https://rothbardbrasil.com/wp-content/uploads/2020/01/Bitcoin-A-Moeda-na-Era-Digital.pdf >. Acesso em: 20 Out. 2023

VISUAL STUDIO CODE. (2023). Visual Studio Code Documentation. Disponível em: https://code.visualstudio.com/docs. Acesso em: 01 set. 2023