

DESENVOLVIMENTO DE UM APLICATIVO MULTIPLATAFORMA PARA ACOMPANHAMENTO DE PREÇOS DE ATIVOS DA BOLSA DE VALORES

Silas Henrique Gonçalves Júnior¹, Rodrigo de Oliveira Plotze¹

¹Faculdade de Tecnologia de FATEC Ribeirão Preto (FATEC)

Ribeirão Preto, SP – Brasil

silas.goncalves@fatec.sp.gov.br,
rodrigo.plotze@fatec.sp.gov.br

Resumo. *O mercado de capitais brasileiro, embora abrangente, permanece pouco explorado por pessoas físicas. Este artigo tem como objetivo disponibilizar informações sobre preços de ativos financeiros negociados em bolsa de valores, por meio do desenvolvimento de um aplicativo multiplataforma para dispositivos móveis. E, deste modo, incentivar pequenos investidores e demais interessados a participarem do mercado, facilitando o acompanhamento das movimentações e oportunidades. O sistema em questão foi projetado e desenvolvido conforme as boas práticas da Engenharia de Software.*

Abstract. *The Brazilian capital market, despite its size, remains underexplored by individual investors. This article aims to provide information on the prices of financial assets traded on stock exchanges through the development of a cross-platform mobile application. In doing so, it seeks to encourage small investors and other interested parties to engage in the market, making it easier to monitor market movements and opportunities. The system under discussion was designed and developed in accordance with best practices in Software Engineering.*

1. Introdução

No ano de 2020 foram estimados cerca de 234 milhões de smartphones no Brasil (FGV, 2020). Tal quantia é maior que a população brasileira registrada em 2022 pelo censo demográfico, de aproximadamente 203 milhões de pessoas (IBGE, 2023). Com este dado em mente, é natural enxergar a grande demanda pelo desenvolvimento de soluções de *software* para dispositivos móveis.

O mercado de capitais é dinâmico e possui um vasto potencial, pois milhões de brasileiros ainda não estão inseridos nele. Entre julho de 2021 e junho de 2022, a entrada de 1,25 milhão de novos investidores fez a quantidade de CPFs aumentar 40%, de 3,15 milhões para 4,40 milhões de investidores do tipo pessoa física na bolsa de valores brasileira (CNN BRASIL, 2022). Com os estímulos corretos, tais como educação financeira e ferramentas adequadas, esse potencial será aproveitado pelos empreendedores que se prepararem desde já.

Este cenário revela uma oportunidade: a da democratização do acesso à informação por meio dos aplicativos para celular. Este projeto almeja disponibilizar a cotação de diversos ativos financeiros, com menor atraso possível em relação aos

mercados e de forma gratuita. Deste modo, mesmo um pequeno investidor ou estudante/entusiasta poderá acompanhar a evolução do mercado e analisá-lo.

Sobre a estrutura deste artigo, seis seções foram elaboradas. A Introdução, que trata do contexto, do objetivo e da organização do texto. O Referencial Teórico, que descreve características fundamentais do mercado financeiro e seus ativos. Na sequência, há a parte de Materiais e Métodos com uma descrição da tecnologia e dos procedimentos utilizados no projeto e desenvolvimento do *software*. Logo após, são apresentados os Resultados. Enfim, estão presentes a Conclusão, que discute os resultados obtidos em comparação com o objetivo inicial, e as Referências.

2. Referencial Teórico

Atualmente, o sistema financeiro brasileiro pode ser dividido em quatro grandes mercados, segundo Riconnect (2021):

- Mercado de câmbio: no qual ocorre troca de moedas. Fundamental para empresas com negócios no exterior, importadoras e exportadoras e até para pessoas físicas que viajam para fora do país;
- Mercado monetário: modalidade em que ocorrem transações do curtíssimo prazo, majoritariamente atreladas ao CDI (Certificado de Depósito Interbancário). O CDI é a taxa que um banco cobra de outro por empréstimos com prazo de um dia e é muito próximo da taxa básica de juros, a Selic;
- Mercado de crédito: este segmento capta dinheiro de investidores para emprestar a empresas e pessoas físicas por uma taxa maior. Para pessoa jurídica, têm-se crédito para compra e manutenção maquinário, crédito para capital de giro, financiamento de projetos, entre outros. No âmbito de pessoa física se têm produtos como cartão de crédito, cheque especial e empréstimo pessoal;
- Mercado de capitais: foco deste projeto, desempenha um papel crucial no sistema financeiro do Brasil, permitindo a captação de recursos por parte das empresas e oferecendo oportunidades de investimento aos indivíduos. Nesse mercado, ocorre a negociação de diversos ativos financeiros, como ações, títulos públicos, debêntures, fundos de investimento, contratos futuros e derivativos entre outros.

O mercado de capitais possui dois intermediários: a B3 (Bolsa, Brasil, Balcão), que é a bolsa de valores brasileira e faz a gestão das negociações e garantias, e uma corretora de valores, que faz a mediação da compra e venda de ativos e presta assessoria aos investidores (RICONNECT, 2021).

As ações representam uma parcela do capital de uma empresa e são negociadas na bolsa de valores. Ao adquirir ações de uma empresa, os investidores se tornam seus acionistas (sócios) e têm direito a receber dividendos e participar das decisões corporativas em assembleias de acionistas (TORO INVESTIMENTOS, 2023).

Os títulos públicos, por sua vez, são emitidos pelo governo federal para financiar suas atividades. Os investidores que adquirem esses títulos estão emprestando dinheiro ao governo e recebem juros como remuneração. Estes instrumentos são considerados investimentos de baixo risco, pois têm o respaldo do governo (REIS, 2018).

Outra forma de captação de recursos pelas empresas são as debêntures, que são

títulos de suas dívidas. Debêntures podem ser emitidas por sociedades anônimas de capital fechado ou aberto. Ao adquirir tais ativos, os investidores emprestam dinheiro à empresa emissora e recebem juros de volta. Os contratos podem ter características diferentes, como prazo de vencimento, taxa de juros e forma de remuneração (B3, 2023).

De acordo com o Ministério da Fazenda (2023), os fundos de investimento são veículos de investimento coletivo, nos quais os recursos de diversos investidores são captados e gerenciados por um gestor profissional. Existem diversos tipos de fundos, como os de renda fixa, que investem em títulos de dívida, os de renda variável, que investem em ações, e os multimercados, que podem combinar diferentes tipos de ativos.

No mercado de capitais também são negociados contratos futuros e derivativos. Os contratos futuros são acordos de compra ou venda de um ativo financeiro para uma data futura, a uma cotação pré-fixada. Já o derivativo é um instrumento financeiro cujo valor é calculado a partir de um ativo subjacente, como ações ou *commodities*. Esse instrumento permite que investidores se protejam contra riscos de mercado, realizem operações de arbitragem ou especulem sobre movimentos futuros dos preços (INFOMONEY, 2022).

No Brasil, a Comissão de Valores Mobiliários (CVM) é a responsável pela regulamentação e fiscalização do mercado de capitais, visando à proteção dos investidores e ao bom funcionamento do mercado. A CVM estabelece regras e normas para a divulgação de informações pelas empresas, o registro de intermediários financeiros e a conduta dos participantes do mercado (BRASIL, 2020).

Outro aspecto de interesse do funcionamento dos mercados é o conceito de *ticker*. Consiste em um código único usado na bolsa de valores para identificar e negociar um determinado ativo. Em outras palavras, ele é uma abreviação do nome do ativo para facilitar tanto a identificação quanto a negociação dele. O *ticker* torna-se importante neste projeto pois é por meio dele que os ativos serão apresentados aos usuários. Outro papel fundamental que ele possui é o de identificador em diversas *APIs* voltadas para a precificação de ativos (SUNO RESEARCH, 2020).

3. Materiais e Métodos

A seguir estão apresentadas as ferramentas que serão utilizadas para o desenvolvimento da aplicação.

Flutter

Flutter é um *framework* que possibilita a criação de aplicativos nativos para dispositivos Android e iOS com a linguagem de programação Dart. O *framework* compila código *Advanced RISC Machine* (ARM) nativo para as duas plataformas, entregando um aplicativo com alto desempenho. A tecnologia utiliza o motor de renderização Skia 2D, uma ferramenta extremamente versátil que trabalha com diversas plataformas (*hardwares* e *softwares*). Além de ser usado nos produtos da empresa Google, é também usado no Mozilla Firefox, FirefoxOS entre outros. Ele também é capaz de compilar as telas em tempo real (*hot reload*), agilizando o desenvolvimento de aplicativos (NAPOLI, 2019).

Além da grande vantagem em criar aplicativos nativos para Android e iOS, o Flutter possui outros atrativos como tempo de compilação rápido comparado a outras

tecnologias, grande reutilização de código devido à ideia de *widgets* (componentização) e uma vasta biblioteca de testes para estes componentes.

Os *widgets* (componentes) representam um botão, uma lista, uma tela ou até mesmo uma animação. Portanto, uma parte da interface do usuário. Conceitualmente são uma classe em Dart, linguagem que faz uso do Paradigma de Programação Orientada a Objetos, e sabem como descrever sua própria exibição (WINDMILL, 2019). No Flutter há dois tipos de *widgets*: sem estado (*stateless*) e com estado (*stateful*). Um estado pode ser definido como uma mudança imperativa na interface do usuário.

A *Stateless Widget* não precisa armazenar informações ou mudar de estado durante seu ciclo de vida. Todos os dados são passados durante a instanciação do objeto da respectiva classe do *widget*. Seu ciclo de vida depende de outros componentes que o usam como parte integrante. Por exemplo, um botão com um texto estático é descartado pelo *garbage collector* do Dart quando a tela a qual ele faz parte é descartada.

A *Stateful Widget* possui estado mutável e ciclo de vida independente, logo, pode armazenar e manipular informações. Seu estado pode mudar por conta de uma entrada/*input* do usuário ou por causa de um gatilho programado (resposta de uma *API*, por exemplo).

É gerada uma árvore de *widgets* que possuem informações a serem renderizadas na tela. O Flutter percorre esta árvore da raiz para as folhas linearmente, coletando as restrições de posição e tamanho que cada elemento possui. Estas restrições são determinadas pelos *widgets* pais (*parents*). Após a definição das restrições, é feito o caminho inverso, para definir os valores reais da renderização (cores, tamanhos e posições de cada componente).

3.1. Dart

Dart é uma linguagem de programação de código aberto criada pelo Google. Segundo Walrath e Ladd (2012), a motivação para criar o Dart foram as necessidades de desenvolver aplicações Web complexas de forma mais simples e rápida, carregá-las rapidamente e executá-las de forma suave para oferecer uma boa experiência aos usuários. Por ser pensada para o desenvolvimento de aplicações Web, possui dentre suas principais bibliotecas recursos de programação assíncrona, codificação e decodificação de dados, manipulação de datas, protocolos de comunicação Web, programação concorrente entre outros.

Como já dito, é uma linguagem orientada a objetos e é possível executá-la nos navegadores e no *prompt* de comando, possibilitando seu uso tanto no lado do cliente como em servidores. Pode ser compilada para JavaScript e utilizar suas bibliotecas (BUCKETT, 2013).

3.2. Application Programming Interface (API)

Application Programming Interfaces (APIs) são conjuntos de regras e protocolos que permitem que diferentes softwares se comuniquem entre si. Elas definem os métodos e formatos de comunicação que os desenvolvedores devem seguir para acessar os recursos e funcionalidades de um determinado sistema ou serviço (HOSTGATOR, 2020).

Uma *API REST (Representational State Transfer)* é um estilo de arquitetura de

software para o desenvolvimento de *APIs*. Ela é baseada nos princípios fundamentais da web e utiliza os protocolos *HTTP* e *URLs* para expor e acessar recursos de forma padronizada (HOSTGATOR, 2020).

A principal diferença entre uma *API REST* dos outros tipos está no estilo de comunicação e no conjunto de princípios que guiam o seu *design*. Algumas das principais diferenças são (HOSTGATOR, 2020):

- Estado e comunicação: As *APIs REST* são baseadas no conceito de transferência de representação do estado de um recurso. Em vez de manter um estado de comunicação entre as solicitações, cada solicitação é autônoma e contém todas as informações necessárias para ser processada;
- Verbos *HTTP*: Esta arquitetura utiliza os verbos *HTTP* (*GET*, *POST*, *PUT*, *DELETE* etc.) para indicar a operação desejada em um recurso. Isso torna a aplicação mais intuitiva e alinhada com o protocolo web;
- Estrutura de *URLs*: São utilizadas *URLs* (*Uniform Resource Locators*) para identificar e acessar recursos. Estes endereços são projetados de forma hierárquica, seguindo um padrão que reflete a estrutura dos dados e ações disponíveis;
- Formato de dados: As *APIs REST* geralmente utilizam formatos de dados leves e universais, como *JSON* (*JavaScript Object Notation*) ou *XML* (*eXtensible Markup Language*), para representar os dados transmitidos entre o cliente e o servidor.

O padrão descrito anteriormente é amplamente adotado e popular na indústria. Existem algumas razões para isso (HOSTGATOR, 2020):

- Simplicidade: Uma *API REST* é relativamente fácil de entender e implementar, pois segue princípios e padrões bem estabelecidos. Isso facilita a adoção e a integração por parte dos desenvolvedores;
- Padronização: O uso do protocolo *HTTP*, verbos e *URLs* padronizados oferece consistência e interoperabilidade entre as diferentes aplicações. Isso permite que os desenvolvedores aproveitem seus conhecimentos e habilidades existentes ao trabalhar com vários projetos;
- Suporte a web: São compatíveis com a arquitetura da *Web*, o que significa que se encaixam bem em aplicativos *Web* e são facilmente acessíveis por meio de navegadores e outras ferramentas.

No entanto, é importante ressaltar que a escolha da melhor *API* depende do contexto e dos requisitos específicos do projeto. Existem outras arquiteturas, como *SOAP* (*Simple Object Access Protocol*) e GraphQL, que podem ser mais adequadas para determinados casos de uso, dependendo das necessidades de comunicação, segurança, desempenho, entre outros fatores.

No mercado financeiro, as *APIs* desempenham um papel crucial ao permitir a integração de sistemas e o acesso a dados e serviços relacionados às transações e atividades financeiras. Elas possibilitam que aplicativos e plataformas de terceiros se conectem a sistemas bancários, bolsas de valores, provedores de pagamentos e outros serviços financeiros.

O aplicativo objeto tema deste trabalho necessita de uma *API* de dados de mercado. Estes serviços fornecem acesso a dados em tempo real ou históricos sobre ações, moedas, *commodities* e outros ativos financeiros. Elas podem incluir cotações, gráficos,

notícias e informações fundamentais.

Existem várias APIs de dados de mercado disponíveis atualmente, oferecidas por provedores de dados financeiros renomados. Alguns exemplos populares são: Alpha Vantage (ALPHA VANTAGE, 2023), IEX Cloud (IEX CLOUD, 2023), Yahoo Finance (YAHOO FINANCE, 2023), Xignite (XIGNITE, 2030) e B3 API (B3, 2023).

A forma como essas aplicações funcionam pode variar entre os provedores, mas geralmente segue alguns passos:

1. Registro e autenticação: Os desenvolvedores precisam se registrar no site do provedor do serviço e obter uma chave de acesso (*API key*) para autenticação. Essa chave é usada para identificar o aplicativo ou plataforma que está fazendo a solicitação;
2. Construção da solicitação: O desenvolvedor constrói uma solicitação *HTTP* especificando os parâmetros desejados, como símbolo do ativo, intervalo de tempo e tipo de dado (cotações, gráficos etc.);
3. Envio da solicitação: A solicitação é enviada para o servidor do provedor da aplicação, normalmente usando os métodos *GET* ou *POST*;
4. Processamento da solicitação: O servidor do provedor processa a solicitação e retorna os dados solicitados no formato desejado, geralmente em *JSON* ou *XML*;
5. Manipulação dos dados: O desenvolvedor, a partir do lado do cliente (*front-end*), recebe a resposta do servidor e pode manipular os dados recebidos de acordo com as necessidades do aplicativo. Isso pode envolver análises, exibição de gráficos, cálculos ou qualquer outra funcionalidade específica.

É importante ler a documentação fornecida pelo provedor da *API* para entender os detalhes específicos de como construir as solicitações, os parâmetros disponíveis e as limitações de uso. Além disso, alguns provedores podem exigir uma assinatura paga para acesso a recursos adicionais ou para um volume maior de dados.

Para o desenvolvimento do aplicativo tema deste projeto, será utilizada a *API* Yahoo Finance. Se trata de um serviço gratuito, com um amplo acervo de dados e de rápido e fácil uso.

3.3. Firebase

O Firebase é uma plataforma de desenvolvimento de aplicativos criada pelo Google. Ele fornece infraestrutura de *back-end* pronta para os desenvolvedores, recurso conhecido como *Backend as a Service (BaaS)*. Isso significa que os desenvolvedores podem se concentrar na criação da aplicação em si e focar menos na parte de configuração e manutenção de servidores (ALURA, 2023).

A plataforma proporciona uma interação dinâmica dos usuários utilizando o Firebase Authentication, pode enviar notificações a várias plataformas com o Cloud Messaging, cria análises de resultados com o Analytics, entre outras funcionalidades (ROCKCONTENT, 2019).

Além disso, o Firebase usa a infraestrutura do Google Cloud, que é projetada para escalar automaticamente de acordo com a demanda. Isso significa que não é necessário preocupar-se com a gestão de servidores ou com o provisionamento de recursos adicionais à medida que o aplicativo cresce. O próprio serviço cuida disso (FIREBASE, 2023).

A solução de software proposta neste projeto é um aplicativo mobile, desenvolvido com o framework Flutter e integrado ao Firebase. O serviço Cloud Firestore será o banco de dados não-relacional (*NoSQL*) utilizado para o armazenamento dos dados dos usuários. A integração com o banco de dados será feita por meio da Firebase REST API. Ainda dentro da gama de serviços oferecidos, a autenticação de usuários será feita pela Firebase Authentication.

3.4. Procedimentos de Engenharia de Requisitos

Os requisitos de um sistema definem suas capacidades e limitações. A engenharia de requisitos é o processo que envolve a identificação, análise e documentação desse escopo. A obtenção e análise desses requisitos podem ser realizadas mediante pesquisas com os *stakeholders*, que são os usuários que interagem com o sistema ou qualquer grupo de pessoas que tenham interesse no *software*. As entrevistas são a técnica mais comumente empregada neste processo (SOMMERVILLE, 2011).

A modelagem é um componente integrante da Engenharia de Requisitos, frequentemente realizada utilizando a Linguagem de Modelagem Unificada (UML, na sigla em inglês). A UML serve como uma representação abstrata do sistema, facilitando a compreensão visual dos requisitos propostos. Dentre os diversos diagramas disponíveis na linguagem, alguns dos mais importantes incluem o Diagrama de Caso de Uso, que ilustra as interações entre o sistema e seu ambiente; o Diagrama de Classes, que exibe as classes de objetos do sistema e suas interações; e o Diagrama de Atividades, que demonstra as atividades envolvidas em um processo e seu fluxo de execução (SOMMERVILLE, 2011). Os diagramas aqui presentes foram criados no Astah (CHANGEVISION INC., 2023), uma ferramenta projetada para a modelagem UML, disponibilizada gratuitamente para estudantes.

3.5. Prototipação

Protótipos de alta fidelidade são necessários para uma comunicação assertiva entre os desenvolvedores e demais pessoas envolvidas no projeto. O Figma, uma ferramenta de desenvolvimento de protótipos, foi selecionada por oferecer uma versão gratuita que permite a criação de interfaces de alta fidelidade para representar os requisitos finais do aplicativo. Não é necessário instalar nenhum *software*, pois a plataforma funciona diretamente no navegador. Como todos os dados estão armazenados na nuvem, é mais fácil incluir pessoas que não estão diretamente envolvidas no *design*, tornando o processo mais simples, fluido, colaborativo e acessível (NIGRI, 2020).

4. Resultados

Nesta seção serão mostrados os diagramas UML feitos com base nos requisitos do sistema, os protótipos de alta fidelidade, assim como uma breve explicação do fluxo de uso do programa. O nome escolhido para o aplicativo foi Alpha Assets.

A seguir estão representados o Diagrama de Caso de Uso (Figura 1), o Diagrama de Atividades (Figura 2) e o Diagrama de Classes (Figura 3).

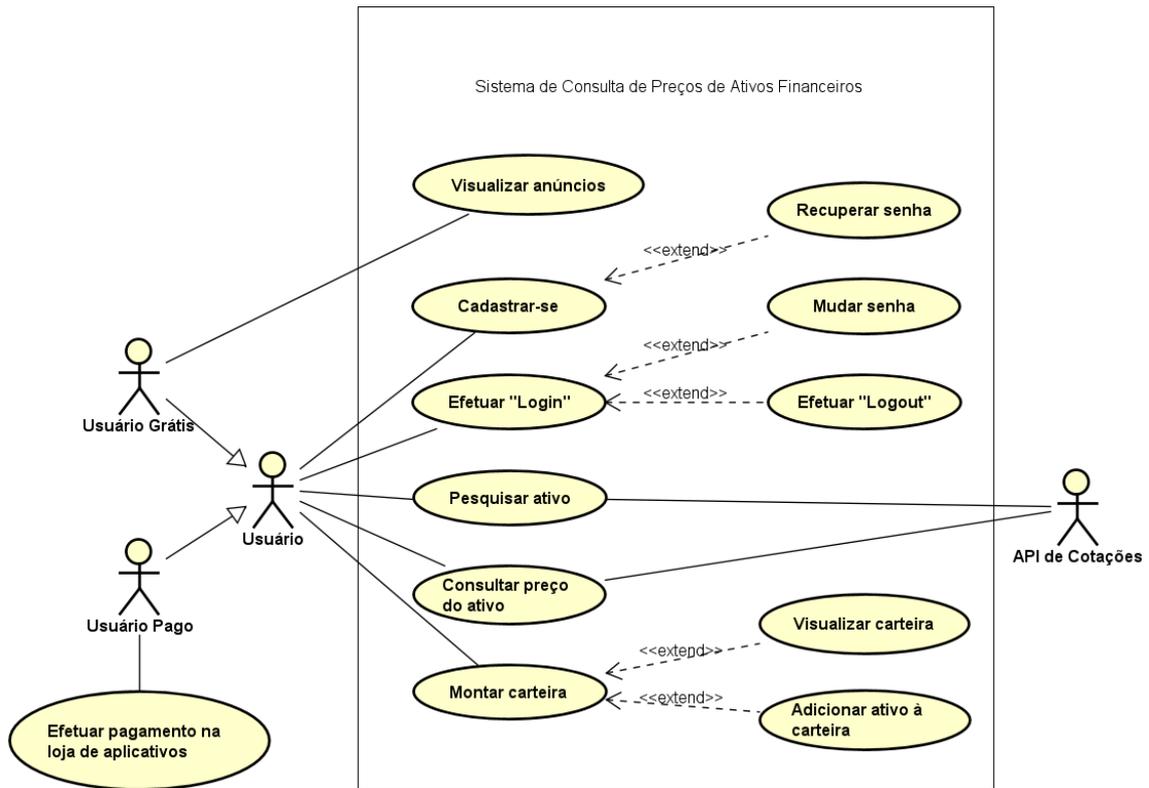


Figura 1: Diagrama de Caso de Uso do Sistema Alpha Assets.
Fonte: Autoria própria.

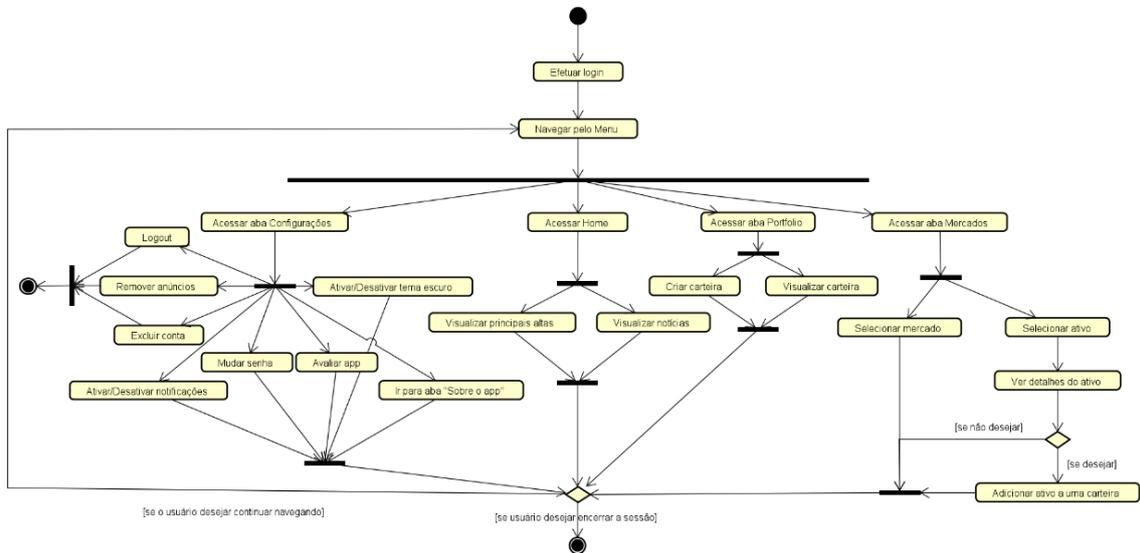


Figura 2: Diagrama de Atividades do Sistema Alpha Assets.
Fonte: Autoria própria.

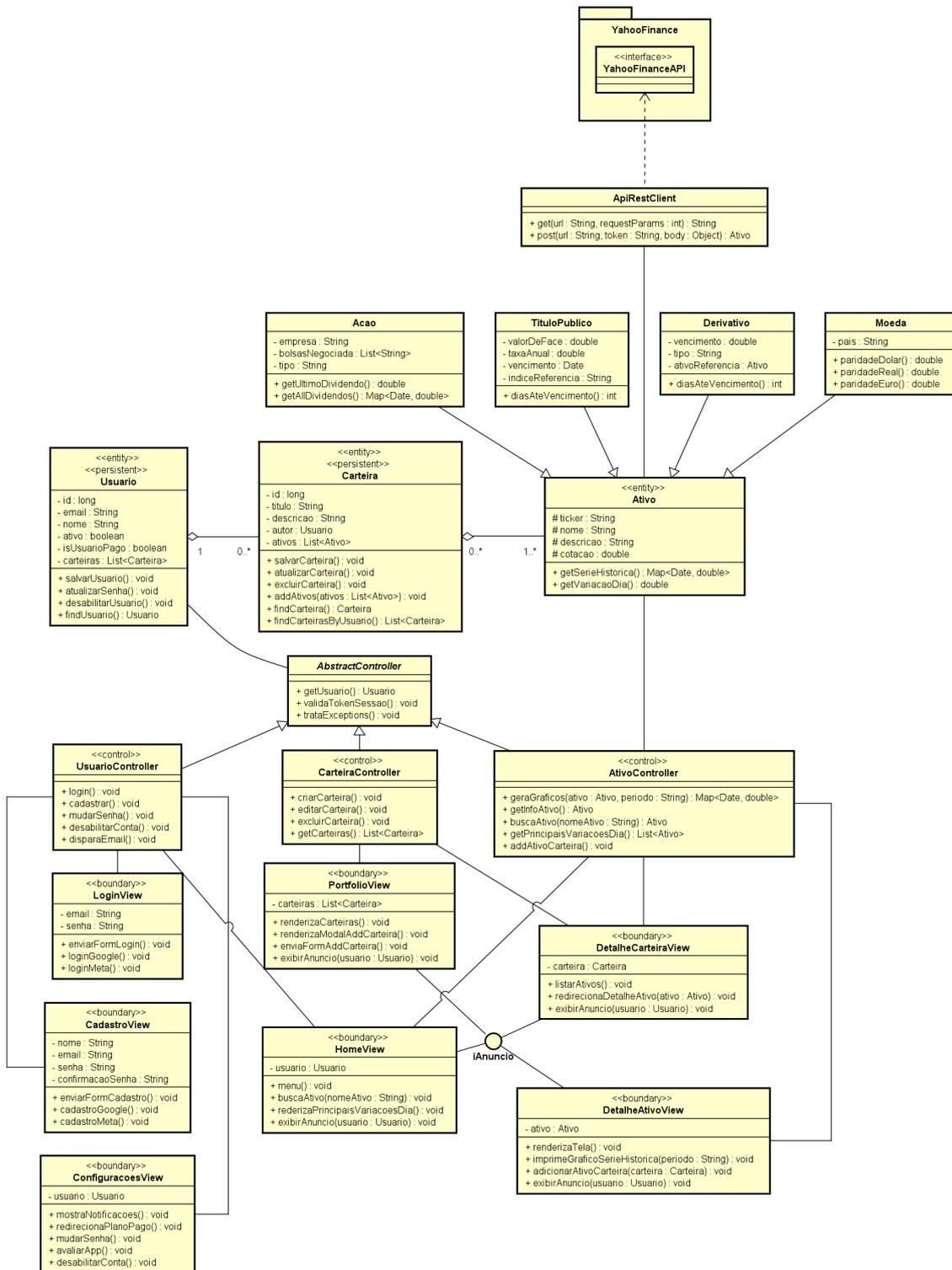


Figura 3: Diagrama de Classes do Sistema Alpha Assets.
Fonte: Autoria própria.

Durante a versão inicial foram prototipadas 10 telas. As principais serão discutidas a seguir para uma melhor compreensão dos casos de uso documentados. Com o objetivo de simplificar as interfaces e manter o foco nas funcionalidades mais importantes, a versão representada nos protótipos não contém anúncios. Esta será a visão do ator

“Usuário Pago” do Diagrama de Caso de Uso.

Na Figura 4a está ilustrada a Tela de Carregamento, também conhecida como *Splash Screen*, utilizada para a apresentação do *app*. Contém o logotipo e o nome do produto. Já na Figura 4b vê-se a Tela de *Login* com dois campos: *e-mail* e senha. É por meio deste formulário que o usuário irá se autenticar no sistema. Será possível também autenticar-se via serviços de terceiros (Meta e Google). Na Tela de Cadastro (Figura 4c), é apresentado um formulário contendo os campos “Nome”, “*E-mail*”, “Senha” e “Confirmar senha”. O *e-mail* cadastrado deve ser válido e acessível pelo usuário pois será enviada uma mensagem de confirmação com um *link* para validação da conta. Além disso, o cadastro poderá ser realizado via integração com contas das empresas Meta e Google.

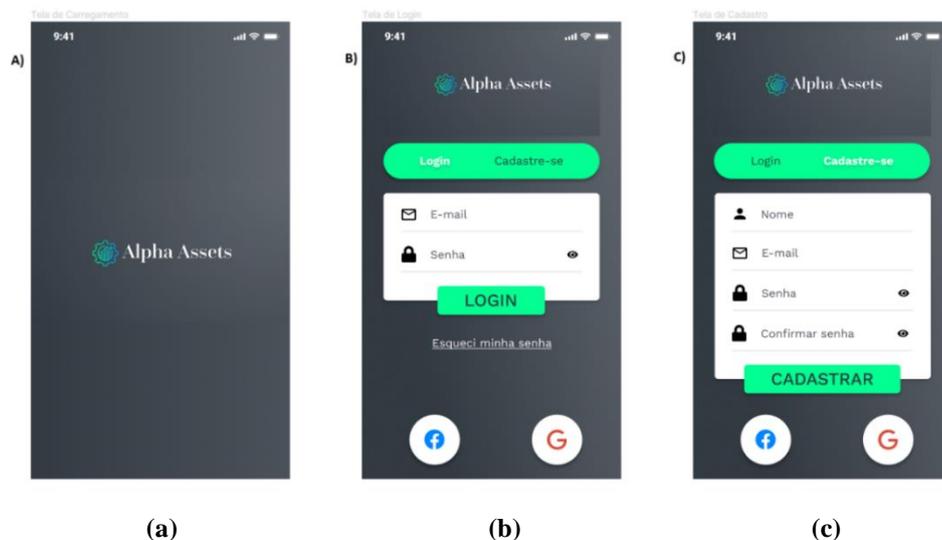


Figura 4: a) Tela de Carregamento; b) Tela de *Login*; c) Tela de Cadastro.
Fonte: Autoria própria.

Após se autenticar na Tela de *Login*, o usuário será redirecionado para a Tela *Home* (Figura 5a). Nesta interface estão presentes uma barra de pesquisa com o *placeholder* “Buscar ativo”, uma seção com o título “Principais Altas”, outra seção nominada “Notícias” e, por fim, um menu de navegação no formato de *Tab Bar*. Este *layout* do menu consiste em uma barra na horizontal, contendo os ícones que representam cada área da interface um ao lado do outro. A partir daqui todas as telas apresentam o menu.

Na Figura 5b está representada a Tela Mercados, segunda posicionada no menu da esquerda para a direita. Há o título “Mercados” em destaque no topo do protótipo e logo abaixo são apresentados os filtros “Todos”, “Criptomoedas” e “Ações”. Serão acessíveis via rolagem horizontal os filtros “Moedas”, “FIIs” e “Títulos Públicos”. Abaixo do grupo de filtragem há um *dropdown* onde é possível ordenar os ativos encontrados de acordo com os critérios “Mais negociados”, “Maiores altas” e “Maiores quedas”. Ao clicar em um ativo o usuário será levado à Tela de Detalhes do Ativo (Figura 5c).

Ao renderizar a próxima interface do fluxo, o *app* exibe um cabeçalho com o nome do ativo selecionado no centro, na esquerda uma seta representando a opção “Voltar” e à direita uma estrela, cujo clique abre um modal para a adição do ativo em uma carteira do usuário. A carteira precisa ser previamente criada para aparecer como opção disponível.

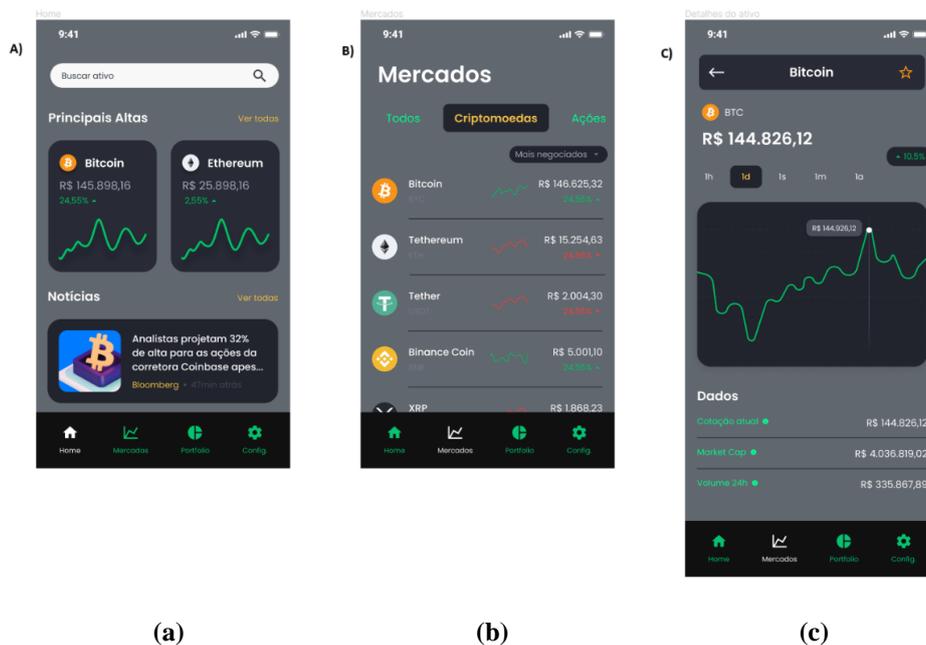


Figura 5: a) Tela *Home*; b) Tela *Mercados*; c) Tela de *Detalhes do Ativo*.
Fonte: *Autoria própria.*

Adiante no menu de navegação, encontra-se o ícone “Portfolio” que dá acesso à Tela Portfolio (Figura 6a). O título “Portfolio” é exibido alinhado à esquerda. Na mesma horizontal e alinhado à direita está presente um ícone com o símbolo “+”, que abre um modal com um formulário para cadastro de uma nova carteira. Os campos necessários são nome e descrição da carteira. Descendo, notam-se grupos que são as representações das carteiras do usuário. Cada grupo tem um nome e uma descrição. Ao selecionar um grupo, o sistema abre a Tela de Detalhe da Carteira (Figura 6b) com os dados pertinentes.

Ao se observar os detalhes de uma carteira, são dispostas informações agrupadas em uma formatação semelhante à outra telas da aplicação. Desta vez, cada grupo representa um ativo financeiro e é composto por título, descrição do ativo, cotação atual e variação diária (representada em verde para as altas e em vermelho para as quedas de preço). Ao tocar em um ativo de sua escolha, o usuário será redirecionado para a Tela de Detalhes do Ativo, Figura 5c, descrita anteriormente.

Por fim a opção mais à direita na *Tab Bar* é o acesso à Tela Configurações (Figura 6c), onde são dispostas em botões as funcionalidades “Desabilitar notificações”, “Tema”, “Remover anúncios”, “Mudar senha”, “Sobre o app”, “Avaliar app”, “Logout” e “Excluir conta”. As atividades “Remover anúncios”, “Logout” e “Excluir conta” irão encerrar a sessão do usuário, criando a necessidade de uma nova autenticação no próximo acesso, assim como ilustrado no Diagrama de Atividades (Figura 2).

A funcionalidade “Remover anúncios” irá redirecionar o usuário para a respectiva loja de aplicativos de seu sistema operacional (Play Store para Android ou App Store para iOS), na qual será oferecida a versão do *app* sem anúncios no formato de compra única.

O protótipo completo pode ser acessado pelo *link* <https://bit.ly/alpha-assets-figma>. O repositório público no GitHub com o código desenvolvido está no endereço <https://bit.ly/apha-assets-repo>.

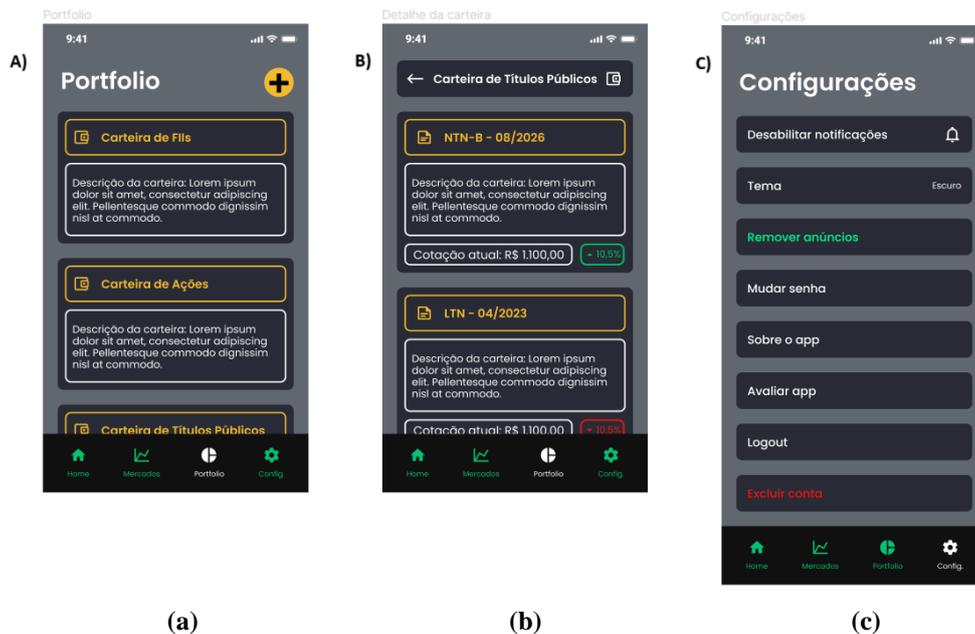


Figura 6: a) Tela Portfólio; b) Tela de Detalhe da Carteira; c) Tela Configurações.
Fonte: Autoria própria.

A primeira versão do aplicativo teve como propósito a implementação das interfaces gráficas e respectivas funcionalidades relativas a *Splash Screen* (Figura 4a), à Tela de *Login* (Figura 4b) e à Tela de Cadastro (Figura 4c). Nas telas de *Login* e Cadastro apenas as integrações de autenticação e criação de conta com os serviços do Google foram implementadas. Os mesmos serviços com a API da Meta serão disponibilizados nas próximas versões. As demais telas tiveram suas interfaces com o usuário parcialmente desenvolvidas, preparando o caminho para a expansão futura do aplicativo.

5. Conclusão

O projeto demonstrou a viabilidade e a importância de disponibilizar informações sobre o mercado financeiro de forma acessível e gratuita por meio de aplicativos móveis. Espera-se que essa iniciativa contribua para a democratização do acesso ao mercado de capitais e incentive a educação financeira da população.

Os devidos padrões da Engenharia de Software foram adotados, tais como a confecção dos diagramas da UML e dos protótipos de alta fidelidade apresentados. Estes processos aumentam a qualidade da aplicação e asseguram uma maior aderência aos requisitos estipulados ao possibilitarem uma comunicação assertiva entre os desenvolvedores e as partes interessadas no projeto.

Como expectativas futuras estão o desenvolvimento da segunda versão do sistema com todas as funcionalidades, entre elas: a disponibilização das Principais Altas e das Notícias na *Home*, a visualização dos dados e gráficos dos ativos e a criação das carteiras.

Estudos futuros incluem a implementação de funcionalidades adicionais, como a possibilidade de criar alertas personalizados para os ativos e a adição de recursos educacionais sobre investimentos, a codificação de testes unitários e testes automatizados, montagem de uma *pipeline* de desenvolvimento (*Continuous Integration/Continuous Delivery*) e a publicação do produto nas respectivas lojas de aplicativos.

Referências

- ALPHA VANTAGE (2023) Free Stock APIs in JSON & Excel, <https://www.alphavantage.co>, Setembro de 2023.
- ALURA (2023) O que é Firebase? Para que serve, principais características e um Guia dessa ferramenta Google, <https://www.alura.com.br/artigos/firebase>, Setembro de 2023.
- B3 (2023) B3 for Developers, https://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/b3-for-developers, Setembro de 2023.
- B3 (2023) Debêntures, https://www.b3.com.br/pt_br/produtos-e-servicos/negociacao/renda-fixa/debentures.htm, Setembro de 2023.
- BRASIL (2020) Comissão de Valores Mobiliários, Sobre a CVM, <https://www.gov.br/cvm/pt-br/aceso-a-informacao-cvm/institucional/sobre-a-cvm>, Setembro de 2023.
- BRASIL (2023) Ministério da Fazenda, Fundos de Investimentos, <https://www.gov.br/investidor/pt-br/investir/tipos-de-investimentos/fundos-de-investimentos>, Setembro de 2023.
- BUCKETT, C. (2013) Dart in Action, Manning Publications.
- CHANGEVISION INC. (2023), About Astah & Change Vision, Inc., <https://astah.net/about>, Outubro de 2023.
- CNN BRASIL (2022), Número de investidores na bolsa cresce 15% em 2022 apostando na diversificação, <https://www.cnnbrasil.com.br/economia/numero-de-investidores-na-bolsa-cresce-15-em-2022-apostando-na-diversificacao>, Novembro de 2022.
- HOSTGATOR (2020), API: Entenda a Interface de Programação de Aplicação, <https://www.hostgator.com.br/blog/api-entenda-a-interface-de-programacao-de-aplicacao>, Setembro de 2023.
- IBGE (2023), De 2010 a 2022, população brasileira cresce 6,5% e chega a 203,1 milhões. Censo 2022, 2023, <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/37237-de-2010-a-2022-populacao-brasileira-cresce-6-5-e-chega-a-203-1-milhoes>, Setembro de 2023.
- INFOMONEY (2022), O que são derivativos e como investir neles: guia completo para começar, <https://www.infomoney.com.br/guias/derivativos>, Setembro de 2023.
- IEX CLOUD (2023), Financial Data Platform | Stock API, <https://iexcloud.io>, Setembro de 2023.
- FGV (2020), Brasil tem 424 milhões de dispositivos digitais em uso, revela a 31ª Pesquisa Anual do FGVcia. <https://portal.fgv.br/noticias/brasil-tem-424-milhoes-dispositivos-digitais-uso-revela-31a-pesquisa-anual-fgvcia>, Maio de 2022.
- FIREBASE (2023), Documentação do Firebase, <https://firebase.google.com/docs?hl=pt-br>, Setembro de 2023.
- NAPOLI, M. (2019), Beginning Flutter: A Hands On Guide to App Development, Wiley.
- NIGRI, D. (2020), O Figma e a próxima geração de ferramentas de Design, UX

- Collective, <https://brasil.uxdesign.cc/o-figma-e-a-pr%C3%B3xima-gera%C3%A7%C3%A3o-de-ferramentas-de-design-ab1682d2a2dd>, Setembro de 2023.
- REIS, T. (2018), Títulos Públicos: o que são e como investir nesses ativos?, Suno Research, <https://www.suno.com.br/artigos/titulos-publicos>, Setembro de 2023.
- RICONNECT (2021), O Que é Mercado Financeiro e Como Ele Funciona, <https://riconnect.rico.com.vc/blog/mercado-financeiro>, Novembro de 2022.
- ROCKCONTENT (2019), Conheça Firebase: a ferramenta de desenvolvimento e análise de aplicativos mobile, <https://rockcontent.com/br/blog/firebase>, Setembro de 2023.
- SOMMERVILLE, I. (2011), Engenharia de Software, Pearson, edição 9, São Paulo.
- SUNO RESEARCH (2020), Ticker: entenda o que é e como interpretar cada um, <https://www.suno.com.br/artigos/ticker>, Novembro de 2022.
- TORO INVESTIMENTOS (2024), O que são ações? Entenda as ordinárias e preferenciais, <https://blog.toroinvestimentos.com.br/bolsa/o-que-sao-acoes>, Maio de 2024.
- WALRATH, K; LADD, S. (2012), Dart: Up and Running, Nutshell handbook, O'Reilly.
- WINDMILL (2019), E. Flutter in Action, Manning Publications.
- XIGNITE (2023), Xignite: Market Data Solutions, <https://www.xignite.com>, Setembro de 2023.
- YAHOO FINANCE (2023), Stock Market Live, Quotes, Business & Finance News, <https://finance.yahoo.com>, Setembro de 2023.