

Plataforma web para gestão de investimentos

Eric Pereira Posser

Curso de Sistemas de Informação

Universidade Franciscana

CEP 97010-032 – Santa Maria, RS, Brasil

eric.posser@ufn.edu.br

Sylvio André Garcia Vieira

Curso de Sistemas de Informação

Universidade Franciscana

CEP 97010-032 – Santa Maria, RS, Brasil

sylvio@ufn.edu.br

Resumo—Investir dinheiro se tornou uma situação mais corriqueira e abrangente, fazendo com que a poupança não seja a única alternativa para as pessoas que tem pouco dinheiro para guardar. Assim este artigo apresenta uma plataforma web voltada à gestão destes investimentos, auxiliando investidores iniciantes e intermediários. O contexto da plataforma é contribuir com esses investidores no controle do patrimônio financeiro e na tomada de decisões mais assertivas, com o apoio de uma interface simplificada, livre de informações complicadas e muitas vezes desnecessárias. Para o desenvolvimento, foi adotada a metodologia FDD *Feature-Driven Development*. O sistema foi implementado utilizando o framework ASP.NET Core, integrado ao sistema gerenciador de banco de dados MySQL. O sistema desenvolvido pode atuar como uma ferramenta de gestão de investimentos, garantindo a qualidade das informações necessárias para seus usuários tanto no controle quanto na tomada de decisão sobre novos investimentos.

I. INTRODUÇÃO

As plataformas de investimentos são locais online que dão acesso a vários produtos para os investidores. Tanto de renda fixa, quanto renda variável. Assim, se pode ter acesso a informações como negociações de aplicações, ativos e outros serviços úteis ao investidor. Desta forma, é possível enviar ordens de compra e venda de ativos, analisar as tendências e acompanhar as movimentações diárias do mercado.

As plataformas de investimentos podem atuar de várias maneiras. Desde permitir acesso aos investidores em diversos tipos de produtos de renda fixa e variável; até agir como uma consultoria, dando várias informações úteis sobre produtos e serviços. Por meio de uma plataforma de investimentos, os investidores podem analisar as melhores ofertas de produtos e fazer suas escolhas de acordo com o seu perfil de investidor.

A falta de gestão financeira é uma realidade para muitos investidores. Muitos enfrentam dificuldades para monitorar seus investimentos [1] e tomar decisões mais assertivas de acordo com o mercado atual [2]. As plataformas existentes possuem muitas informações, interfaces complexas ou exigem conhecimento avançado no assunto, o que acaba afastando investidores iniciantes ou intermediários do mercado financeiro.

O objetivo deste trabalho é proporcionar um gerenciamento de investimentos, por meio de uma interface atrativa e intuitiva para os usuários, acompanhando tendências de design atuais.

Assim, pode-se elencar os seguintes objetivos específicos:

- Ser planejada para facilitar a compreensão do usuário;

- Ser orientada a investidores iniciantes ou intermediários;
- Ter foco na simplicidade e objetividade;
- Proporcionar ao usuário uma visualização gráfica da evolução dos seus investimentos;
- Ter informações divididas por categorias de investimentos evitando a poluição visual;
- Auxiliar o usuário na tomada de decisões mais assertivas na hora de investir seu dinheiro;

II. REFERENCIAL TEÓRICO

Quando se fala em mercado de ações, é necessário primeiro, entender o que são, portanto, ações são títulos que representam uma fração do valor das empresas ou sociedades anônimas. Ou seja, uma ação é como se fosse um pedaço de uma empresa. [3]

Quando uma instituição decide expandir seu negócio, muitas vezes necessita buscar mais dinheiro para isso. Portanto, muitas delas se tornam empresas de capital aberto e ofertam suas ações, chamadas também de papéis, no mercado para obter recursos.

Dessa forma, qualquer pessoa devidamente registrada na Bolsa de Valores pode adquirir esses títulos, passando a integrar o grupo de acionistas da companhia.

Os Fundos de Investimento Imobiliário (FII) são compostos por grupos de investidores que possuem o objetivo de aplicar recursos em diversos tipos de investimentos imobiliários, seja no desenvolvimento de empreendimentos, em imóveis já prontos como edifícios comerciais, shopping centers e hospitalares, ou até mesmo em títulos de dívida imobiliária. [4]

Criptomoeda é o nome genérico para moedas digitais descentralizadas, criadas em uma rede blockchain a partir de sistemas avançados de criptografia que protegem as transações, suas informações e os dados de quem transaciona. [5]

Em outras palavras, criptomoedas são moedas digitais porque, diferentemente do real, do dólar e de outras moedas que podem ser tocadas, elas só existem na internet.

Renda fixa é um tipo de investimento que geralmente é mais seguro e previsível. Na prática, quando alguém aplica em renda fixa, está emprestando dinheiro para uma instituição financeira, uma empresa ou diretamente para o país. Em troca, quem toma o empréstimo se compromete a devolver o seu dinheiro com uma porcentagem de ganho, que pode estar relacionada à variação da inflação, à taxa básica de juros ou a uma taxa já determinada no momento da aplicação. [6]

C Sharp, é a principal linguagem de programação em .NET, que é uma plataforma de programação da empresa Microsoft. Com C Sharp é possível desenvolver aplicações de praticamente todos os tipos: desde aplicações embarcadas até aplicativos de área de trabalho, mobile e sistemas web. Como é orientada a objetos, é possível criar classes e estruturas de programação que são reaproveitadas por diversas aplicações. [7]

Framework é um conjunto de bibliotecas, que abordam funcionalidades, e estruturas, para o desenvolvimento de aplicações, a fim de fornecer soluções para um mesmo domínio de problema, permitindo a reutilização do seu código. Assim, por meio das diversas linguagens de programação. [8]

Com o Entity Framework Core, o acesso a dados é executado usando um modelo. Um modelo é feito de classes de entidade e um objeto de contexto que representa uma sessão com o banco de dados. O objeto de contexto permite consultar e salvar dados. [9]

.NET é uma plataforma de desenvolvimento gratuita multiplataforma e de código aberto para criar muitos tipos de aplicativos. Ele pode executar programas escritos em vários idiomas, com C Sharp sendo os mais populares. [10]

ASP.NET é uma estrutura popular de desenvolvimento na Web para a criação de aplicativos Web em Plataforma .NET. [11]

O MVC é um padrão de arquitetura de software que sugere uma maneira para se pensar na divisão de responsabilidades, principalmente dentro de um software web. O princípio básico do MVC é a divisão da aplicação em três camadas: a camada de interação do usuário (view), a camada de manipulação dos dados (model) e a camada de controle (controller). [12]

HTML, CSS e Javascript são um conjunto de estrutura, regras e funcionalidades que podem ser utilizadas para realizar a construção de uma página web ou até mesmo de uma aplicação desktop. [13]

O HTML, que possui a sigla original *Hyper Text Markup Language*, ou em português Linguagem de Marcação de Hipertexto, trata-se de blocos que são utilizados para construção básica de layouts voltados para web com o uso de regras e sintaxes que são reservadas para essa linguagem de marcação. [13]

CSS significa *Cascading Style Sheet* (Folha de Estilo estruturada em Cascata), utilizado para realizar a estilização de elementos que são escritos por alguns frameworks ou até mesmo pela linguagem de marcação HTML. Ele é responsável por toda a parte visual do site. [13]

O Javascript é uma linguagem de programação de alto nível, voltada para o desenvolvimento web dando a possibilidade da criação de conteúdos dinâmicos. Com a utilização do HTML, apenas conteúdos estáticos eram possíveis de serem criados. Já com o Javascript, é permitida a criação de textos e imagens animadas, além de informações dinâmicas para as pessoas usuárias. [13]

Bootstrap é um framework front-end que fornece estruturas de CSS para a criação de sites e aplicações responsivas de forma rápida e simples. [14]

jQuery é uma biblioteca JavaScript. Sua principal finalidade é associar-se aos elementos JavaScript em HTML para conferir mais dinamismo e usabilidade às páginas na internet. [15]

Git é um sistema de controle de versão open-source, ou seja, gratuito. Ele é utilizado para a criação de um histórico de alterações em código-fonte de projetos de desenvolvimento de software. [16]

O Github é um repositório remoto, criado como um serviço online de hospedagem de repositórios do Git. Funciona como um servidor que agrupa todas as modificações realizadas por cada uma das pessoas envolvidas em um projeto, unificando as diferentes versões de código e seus históricos, permitindo compartilhamento entre as equipes. [16]

MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto. Assim como em outros bancos de dados relacionais, o MySQL armazena dados em tabelas compostas de linhas e colunas. Os usuários podem definir, manipular, controlar e consultar dados usando a linguagem de consulta estruturada, mais conhecida como SQL. [17]

A sigla API significa *Application Programming Interface* (ou Interface de Programação de Aplicações, em uma tradução livre). O termo designa todas as interfaces computacionais utilizadas para a coleta, o envio e a troca de informações entre diferentes programas e equipamentos. [18]

A Brapi é uma plataforma open source que oferece uma poderosa API para desenvolvedores, empresas e entusiastas do mercado financeiro e tem como objetivo simplificar o acesso a dados financeiros e promover a democratização das informações no mercado de ações e moedas.[19]

III. TRABALHOS CORRELATOS

1) Desenvolvimento de um software para gerenciamento de investimentos

O trabalho de Rodrigo Pacheco [20] teve como objetivo a criação de um software para o gerenciamento de carteiras de investimento, voltado especialmente para investidores iniciantes no mercado financeiro. A proposta busca auxiliar esses usuários a terem uma visualização clara e simples de cada ativo presente em seu portfólio, destacando a parcela que cada investimento representa dentro da carteira.

Os autores apresentaram que no decorrer do trabalho foram levantadas diversas questões sobre a real necessidade do sistema, o público-alvo e as diretrizes para o seu desenvolvimento. O planejamento focou na escolha das ferramentas e na definição da arquitetura mais adequada, visando à construção de um sistema funcional e eficiente. Os autores optaram pela arquitetura Model View Controller (MVC), amplamente reconhecida por sua organização e escalabilidade. Na modelagem do sistema, foi utilizada a *Unified Modeling Language* (UML) para a criação de diagramas que ilustram o funcionamento do software. A codificação foi realizada em linguagem Java, por meio da plataforma de desenvolvimento NetBeans IDE 12.5.

O resultado final foi satisfatório, destacando-se principalmente pela interface simples e intuitiva do aplicativo, que facilita a usabilidade para os usuários. As metodologias e ferramentas adotadas mostraram-se extremamente eficazes quando

utilizadas em conjunto, contribuindo para um desenvolvimento harmonioso do projeto e possibilitando o cumprimento de todos os requisitos propostos.

2) Desenvolvimento de uma aplicação web para gestão de carteira de investimentos em bolsa de valores

O trabalho de Vitor Martins Basso [21] tem como objetivo o desenvolvimento de um sistema web voltado para o gerenciamento de carteiras de investimentos em ações na bolsa de valores. Além de permitir o acompanhamento detalhado dos ativos, a aplicação também realiza o cálculo do imposto de renda resultante do portfólio do usuário — uma funcionalidade de grande importância que, em muitas ferramentas similares, costuma ser negligenciada.

O planejamento do sistema foi estruturado em etapas bem definidas. A primeira consistiu no levantamento dos requisitos funcionais e não funcionais, que determinaram o escopo e o comportamento esperado da aplicação. Em seguida, foi realizada a modelagem de dados, com o desenvolvimento de um modelo para organizar o armazenamento das informações de forma eficiente. Por fim, houve a criação de protótipos de baixo nível, que permitiram uma rápida iteração e coleta de feedbacks para melhorias antes da implementação final.

Como resultado, foi idealizada e implementada uma aplicação para a gestão de investimentos na B3 (bolsa de valores oficial do Brasil que reúne negociações de ações, títulos públicos, commodities, entre outros ativos), com o intuito de auxiliar investidores a enfrentarem os desafios inerentes a esse tipo de operação. O sistema permite o cadastro de transações, o acompanhamento do desempenho da carteira e a geração de relatórios detalhados sobre o imposto de renda. Além disso, sua interface responsiva garante usabilidade em diferentes tamanhos de tela, tornando a experiência do usuário mais prática e acessível.

3) Desenvolvimento de um aplicativo para controle financeiro pessoal

O trabalho de Bruno Souza Ramos [22] tem como objetivo o desenvolvimento de uma aplicação que proporcione um controle mais transparente das receitas e despesas dos usuários, facilitando a fiscalização do seu estado financeiro atual. Além disso, a proposta busca, de forma implícita, promover a educação financeira do usuário por meio do uso contínuo da ferramenta.

Para compreender melhor o perfil financeiro do brasileiro, foi realizada uma pesquisa do tipo survey, com abordagem quantitativa e caráter qualitativo. A coleta de dados foi feita através de um formulário distribuído pelas redes sociais. Vizando à segurança e ao anonimato dos participantes, nenhuma informação de identificação pessoal foi solicitada. A partir da análise dos resultados obtidos nessa pesquisa, foram definidos os requisitos fundamentais para o desenvolvimento do projeto.

Com base nesses requisitos, foi realizado uma análise criteriosa das tecnologias e ferramentas disponíveis, o autor optou por aquelas que melhor atendem aos critérios de eficiência e segurança. Como resultado, acredita-se que o trabalho cumpriu com êxito seu objetivo principal, oferecendo um aplicativo móvel que contribui para inverter o cenário financeiro pre-

ocupante vivido por muitos brasileiros, ao favorecer tanto o controle quanto o aprendizado financeiro do usuário.

4) Inferência dos Trabalhos Correlatos

O trabalho de Rodrigo Pacheco [20] contribuiu significativamente para esta pesquisa, especialmente pela forma como os autores abordaram o desenvolvimento de um software voltado para investidores iniciantes. A proposta de oferecer uma interface intuitiva e simples serviu como referência importante. Além disso, o trabalho se destacou pela escolha da arquitetura Model-View-Controller (MVC), aplicada ao desenvolvimento da aplicação.

O trabalho de Vitor Martins Basso [21] colaborou com a estruturação da etapa de planejamento, destacando o levantamento de requisitos e a modelagem prévia ao desenvolvimento. Sua contribuição também se estendeu à abordagem da gestão de investimentos na B3 (bolsa de valores oficial do Brasil, responsável pelas negociações de ações, títulos públicos, commodities, entre outros ativos), além da funcionalidade de geração de relatórios detalhados sobre os investimentos realizados.

O trabalho de Bruno Souza Ramos [22] contribuiu com a promoção da educação financeira, incentivando mais pessoas a ingressarem no mercado financeiro brasileiro. Além disso, trouxe como foco o atendimento à comunidade de usuários mobile, o que reforçou a necessidade de desenvolver uma aplicação responsiva, compatível tanto com plataformas web quanto mobile. Sua pesquisa também se destacou pela preocupação em escolher tecnologias adequadas aos objetivos do projeto.

IV. METODOLOGIA

O modelo de metodologia ágil escolhida foi a *Feature-Driven Development* (FDD) [23], por ser adequada a projetos com funcionalidades bem definidas e por se mostrar eficaz tanto para pequenas equipes quanto para desenvolvedores individuais. Esse modelo destaca-se por listar funcionalidades de forma clara e prática, minimizando a burocracia e mantendo o foco no desenvolvimento efetivo do software.

O FDD [23] é uma das metodologias ágeis originais que foca na entrega de funcionalidades específicas de um software, segmentando o projeto em pequenas “features”. Esse método segue um processo interativo dividido em cinco etapas subsequentes:

- Desenvolvimento de um modelo global;
- Elaboração de listas de funcionalidades;
- Planejamento por funcionalidade;
- Projeto por funcionalidade;
- Construção por funcionalidade;

A. Desenvolvimento de um modelo global

É a fase inicial do FDD, nesta etapa, a equipe avalia o sistema, e os analistas identificam os objetos essenciais do sistema, suas propriedades e comportamentos. Na Figura 1, é possível visualizar um diagrama de domínio do sistema, a

fim de possibilitar uma visão geral do que foi desenvolvido.

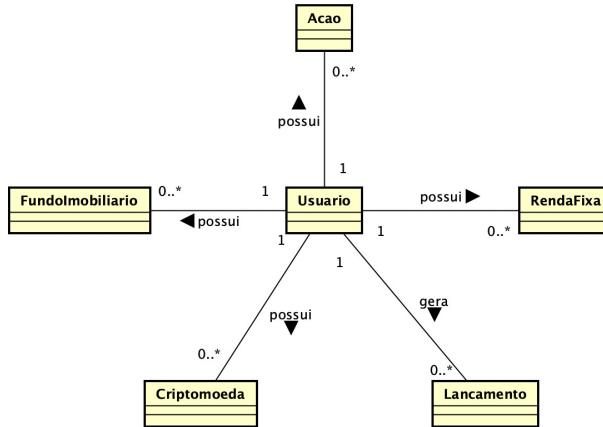


Figura 1: Diagrama de Domínio do Sistema, elaborado pelos autores.

B. Elaboração de listas de funcionalidades

Na segunda etapa do FDD, é necessário desenvolver uma lista de funcionalidades, descrevendo os requisitos funcionais e não funcionais do sistema. Os requisitos funcionais são os serviços que o sistema deve fornecer e funcionalidades específicas que ele deve executar. Por outro lado, os requisitos não funcionais são os que não possuem relação direta com os serviços e normalmente especificam as características do sistema.

Tabela I: Tabela de Requisitos Funcionais, elaborada pelos autores.

REQUISITOS FUNCIONAIS	
RF01	O sistema deve permitir que novos usuários se cadastrem informando login e senha e deve ter o controle individual de investimentos por usuário.
RF02	O sistema deve permitir que o usuário se autentique com login e senha criptografada.
RF03	O sistema deve permitir a adição e exclusão de Ações.
RF04	O sistema deve permitir a adição e exclusão de Fundos Imobiliários.
RF05	O sistema deve permitir a adição e exclusão de Criptomoedas.
RF06	O sistema deve permitir a adição e exclusão de investimentos de Renda fixa.
RF07	O sistema deve consultar informações de ativos selecionados por meio da API da Brapi.
RF08	O sistema deve permitir que o usuário ordene os ativos por nome, código, valor e quantidade clicando no cabeçalho da tabela.
RF09	O sistema deve gerar lançamentos de compra ou venda para cada investimento cadastrado do usuário.

Tabela II: Tabela de Requisitos Não Funcionais, elaborada pelos autores.

REQUISITOS NÃO FUNCIONAIS	
RNF01	A aplicação deve ser escalável para suportar múltiplos usuários simultâneos.
RNF02	O sistema deve utilizar boas práticas de desenvolvimento como separação de responsabilidades e uso de serviços.
RNF03	O sistema deve ser compatível com os navegadores modernos.
RNF04	O sistema deve ter tempo de resposta inferior a 3 segundos por ação para melhor experiência do usuário.

C. Planejamento por funcionalidade

A terceira etapa da metodologia consiste em planejar o desenvolvimento. Baseado na lista de funcionalidades construída, é descrita uma ordem de implementação de funcionalidades. Na Figura 2, pode-se analisar o diagrama de atividades do projeto, utilizado para demonstrar o processo de negócio em que o software é empregado, nele, é possível visualizar o principal fluxo de atividades do sistema, que consiste em o usuário realizando seu login, acessando o dashboard, selecionando a seção correspondente ao tipo de investimento desejado (ações, fundos imobiliários, criptomoedas ou renda fixa), preenchendo o formulário de cadastro do ativo com os dados necessários e, por fim, visualizando a listagem de seus investimentos.

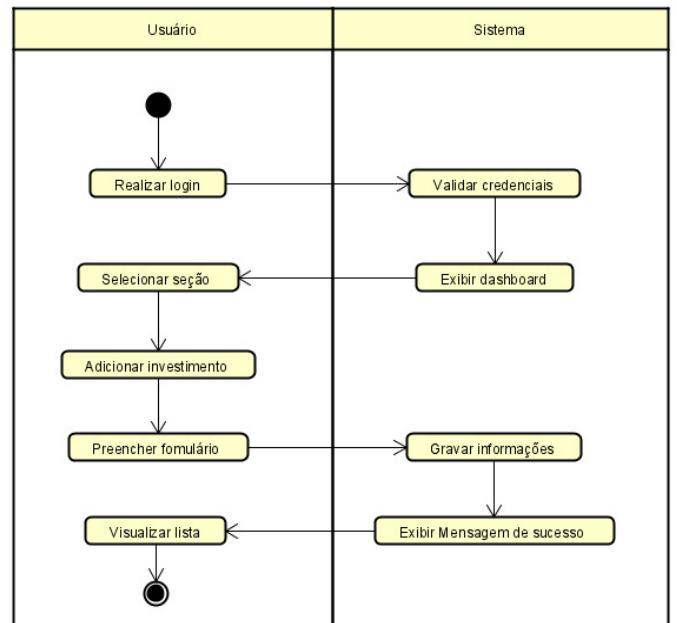


Figura 2: Diagrama de Atividade do Sistema, elaborado pelos autores.

D. Projetar por funcionalidade

Nesta etapa da metodologia, o objetivo é detalhar a implementação do conjunto de atividades. Na Figura 3, apresenta-se o Diagrama de Caso de Uso do sistema, que ilustra as funcionalidades disponíveis aos usuários.

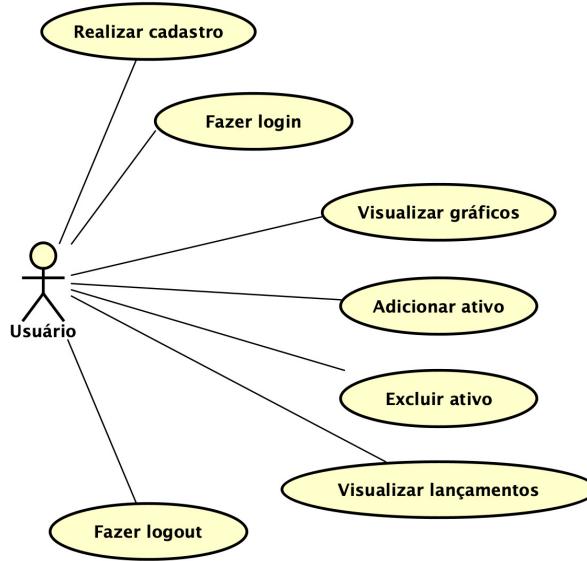


Figura 3: Diagrama de Caso de Uso do Sistema, elaborado pelos autores.

Na Figura 4 apresenta o diagrama de classe do sistema, focado na modelagem das entidades que refletem a estrutura do banco de dados.

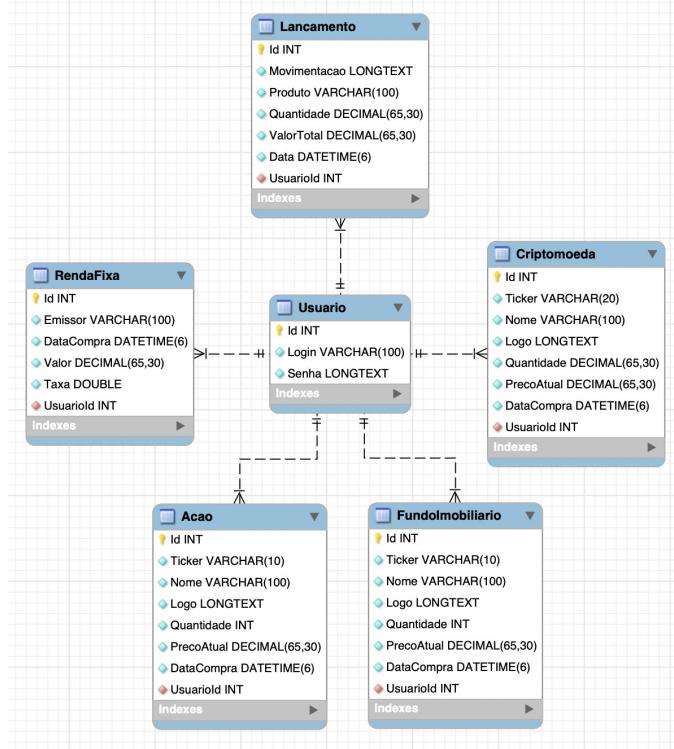


Figura 4: Diagrama de classe, elaborado pelos autores.

E. Construção por funcionalidade

Na última seção do FDD [23], ocorre a implementação do software, e consiste no desenvolvimento de cada funcionalidade identificada nas etapas iniciais. O sistema elabo-

rado é composto por uma integração via API a plataforma "Brapi" para acesso aos dados do mercado financeiro [19] e uma interface web, ambos integrados para garantir uma experiência fluida e eficiente ao usuário final.

1) Estrutura do projeto: A plataforma foi desenvolvida em ASP.NET Core, organizando o projeto em camadas de Models, Views, Controllers, Services e Interfaces. As classes contidas na pasta Models representam as entidades do banco de dados, sendo definidas com propriedades que refletem as colunas e os relacionamentos das tabelas. As Controllers atuam como intermediárias, recebendo as requisições HTTP, acionando os serviços da camada Service e mapeando cada funcionalidade necessária para realizar operações de criação e exclusão nas entidades, além de gerenciar a autenticação dos usuários. As Views são responsáveis por armazenar o código da interface que interage com o usuário, sendo projetadas para proporcionar uma experiência intuitiva e responsiva.

Para conectar a API ao banco de dados, foi criada uma classe de contexto do Entity Framework [9], que serve como ponto de acesso principal para as operações com o banco. Essa classe gerencia a comunicação com o banco e permite configurar regras de mapeamento, como chaves primárias e relacionamentos entre tabelas, promovendo uma camada de abstração entre o código e o banco de dados.

Na Figura 5, é possível visualizar um dos endpoints presentes no AcaoController, em que é feita uma requisição GET para buscar na API a ação que o usuário deseja adicionar retornando o nome e o preço atual. Além disso, na Figura 6 é exibida o método que faz esta chamada da API para buscar o ativo selecionado e logo na Figura 7 é exibida a organização de classes e pastas do projeto.

```
[HttpGet]
public async Task<IActionResult> BuscarAtivo(string acao, [FromServices] IBrapi brapi)
{
    var ativo:Brapi? = await brapi.ObterAtivo(acao);

    if (ativo is null)
    {
        return NotFound(new { mensagem = "Ativo não encontrado." });
    }

    return Json( new
    {
        nome = ativo.Nome,
        precoAtual = ativo.PrecoAtual
    });
}
```

Figura 5: Exemplo de endpoint da API.

```

public async Task<Brapi> ObterAtivo(string ticker)
{
    var token:string? = _config["Brapi:Token"];

    var url = $"https://brapi.dev/api/quote/{ticker}?token={token}";

    var response = await _httpClient.GetAsync(url);
    if (!response.IsSuccessStatusCode)
        return null;

    var jsonParaString = await response.Content.ReadAsStringAsync();
    using var documentoJson = JsonDocument.Parse(jsonParaString);
    var ativoJson = documentoJson.RootElement.GetProperty("results")[0];

    return new Brapi
    {
        Ticker = ativoJson.GetProperty("symbol").GetString() ?? string.Empty,
        Nome = ativoJson.GetProperty("longName").GetString() ?? string.Empty,
        Logo = ativoJson.GetProperty("logourl").GetString() ?? string.Empty,
        PrecoAtual = ativoJson.GetProperty("regularMarketPrice").GetDecimal()
    };
}

```

Figura 6: Método para buscar ativo da API.

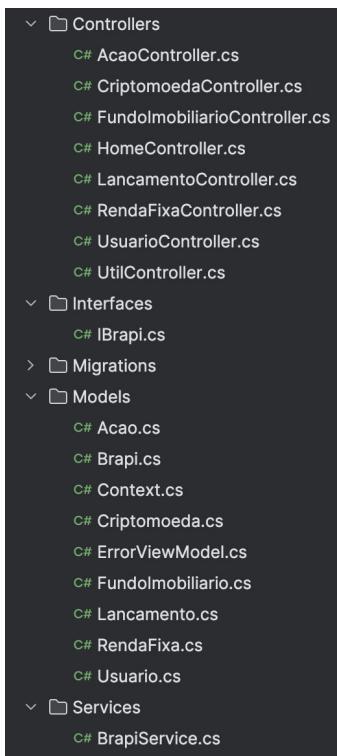


Figura 7: Estrutura do projeto.

2) *Interface Web*: A interface web do sistema foi desenvolvida com o uso do ASP.NET Core MVC, possibilitando uma separação clara entre a camada de visualização e as demais camadas do sistema. As páginas que compõem a aplicação encontram-se na pasta Views, subdivididas por entidade, como Ação, Fundo Imobiliário, Criptomoeda, Renda Fixa e Lançamento. Cada subpasta contém arquivos .cshtml correspondentes às ações de criação e de listagem de ativos,

já a exclusão de um ativo, foi feita por meio de um modal. A Figura 8 apresenta o método para carregar as ações do usuário com a funcionalidade de filtro por código, nome, preço ou quantidade.

```

[HttpGet]
♂ 8 usages
public async Task<IActionResult> Index(string sortOrder)
{
    var usuarioId int = ObterUsuarioId();

    var acoes IQueryable<Acao> = _context.Acao.Where(a.Acao => a.UsuarioId == usuarioId);

    ViewBag.TickerSort = string.IsNullOrEmpty(sortOrder) ? "ticker_desc" : "";
    ViewBag.NomeSort = sortOrder == "Nome" ? "nome_desc" : "Nome";
    ViewBag.PrecoSort = sortOrder == "Preco" ? "preco_desc" : "Preco";
    ViewBag.QuantidadeSort = sortOrder == "Quantidade" ? "quantidade_desc" : "Quantidade";

    acoes = sortOrder switch
    {
        "ticker_desc" => acoes.OrderByDescending(a.Acao => a.Ticker),
        "Nome" => acoes.OrderBy(a.Acao => a.Nome),
        "nome_desc" => acoes.OrderByDescending(a.Acao => a.Nome),
        "Preco" => acoes.OrderBy(a.Acao => a.PrecoAtual),
        "preco_desc" => acoes.OrderByDescending(a.Acao => a.PrecoAtual),
        "Quantidade" => acoes.OrderBy(a.Acao => a.Quantidade),
        "quantidade_desc" => acoes.OrderByDescending(a.Acao => a.Quantidade),
        _ => acoes.OrderBy(a.Acao => a.Ticker),
    };

    return View(await acoes.ToListAsync());
}

```

Figura 8: Método para listar ações com filtros.

A Figura 9 apresenta a estrutura de pastas das interfaces do sistema.

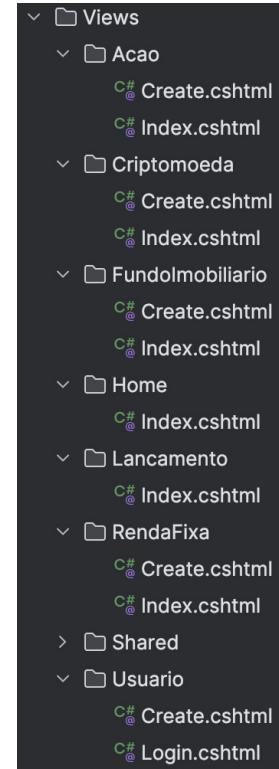


Figura 9: Estrutura de pastas da interface.

V. RESULTADOS

Como resultado deste trabalho, foi desenvolvida uma plataforma web que permitiu a gestão de investimentos para investidores iniciantes e intermediários, disponível em <http://ericposser-001-site2.jtempurl.com>.

A Figura 10 apresenta a página inicial da plataforma, onde está localizado o dashboard com as informações mais relevantes sobre os investimentos do usuário. Nela, há um gráfico com a porcentagem de cada categoria de ativo, outro gráfico com os valores totais investidos pelo usuário, o total investido por categoria e o valor total de sua carteira completa. Além disso, há um gráfico que mostra a evolução patrimonial no ano atual.



Figura 10: Tela de dashboard do usuário.

A Figura 11 apresenta a interface com as ações cadastradas. Nela, o usuário pode adicionar um ativo com uma busca automatizada por meio da integração com a API da "Brapi"[19], que retorna dados como o nome completo do ativo e sua cotação atual. A interface também permite excluir ações cadastradas, além de aplicar filtros por nome, valor total ou quantidade, facilitando a organização e análise da carteira.

Ativo	Nome	\$ Valor total	# Quantidade	Ações
ABEV3	Ambel S.A.	R\$ 4.371,00	300	<button>Excluir</button>
BBAS3	Banco do Brasil S.A.	R\$ 2.025,10	70	<button>Excluir</button>
BBDC4	Banco Bradesco S.A.	R\$ 10.984,00	800	<button>Excluir</button>
BPAN4	Banco Pan S.A.	R\$ 684,90	90	<button>Excluir</button>
LREN3	Lojas Renner S.A.	R\$ 730,50	60	<button>Excluir</button>
MGLU3	Magazine Luiza S.A.	R\$ 1.118,40	120	<button>Excluir</button>
OIBR3	Qt S.A.	R\$ 2,85	5	<button>Excluir</button>
PETR4	Petróleo Brasileiro S.A. - Petrobras	R\$ 91,74	3	<button>Excluir</button>
TIM3	Tim S.A.	R\$ 379,00	20	<button>Excluir</button>

Figura 11: Tela de gestão de ações.

A Figura 12 apresenta a tela de extrato de operações, na qual são exibidas todas as movimentações de compra e venda realizadas pelo usuário e podem ser refinados por meio de filtros no cabeçalho da tabela, como tipo de operação, código, quantidade, valor total ou data. Essa funcionalidade auxilia o investidor a visualizar seu histórico com clareza e precisão.

Movimentação	Produto	Quantidade	Preço Unitário	Valor Total	Data
VENDA	PETR4	3	R\$ 30,08	R\$ 91,74	01/05/2025
VENDA	BBDC4	800	R\$ 13,73	R\$ 10.984,00	01/05/2025
VENDA	LREN3	50	R\$ 14,61	R\$ 730,50	01/05/2025
VENDA	TIM3	20	R\$ 18,95	R\$ 379,00	01/05/2025
VENDA	HAPV3	4	R\$ 2,40	R\$ 9,60	01/05/2025
COMPRA	BPAN4	90	R\$ 7,61	R\$ 684,90	01/05/2025
COMPRA	OIBR3	5	R\$ 0,57	R\$ 2,85	01/05/2025
COMPRA	TIM3	20	R\$ 18,95	R\$ 379,00	01/05/2025
COMPRA	BBAS3	70	R\$ 28,93	R\$ 2.025,10	01/05/2025
COMPRA	LREN3	50	R\$ 14,61	R\$ 730,50	01/05/2025
COMPRA	MGLU3	120	R\$ 9,32	R\$ 1.118,40	01/05/2025
COMPRA	ABEV3	300	R\$ 14,57	R\$ 4.371,00	01/05/2025
COMPRA	BBDC4	800	R\$ 13,73	R\$ 10.984,00	01/05/2025
VENDA	KLB11	4	R\$ 18,57	R\$ 74,28	24/04/2025
VENDA	AZUL4	2	R\$ 2,56	R\$ 5,12	24/04/2025
VENDA	USM5	4	R\$ 5,94	R\$ 23,76	24/04/2025

Figura 12: Tela de lançamentos.

O sistema garante que cada usuário acesse exclusivamente os próprios dados.

Além das telas citadas, a plataforma conta com uma tela inicial de login e de cadastro que protege o acesso aos dados, e um painel de controle que é responsável por exibir um resumo geral da carteira do usuário. Esse painel inclui informações sobre ações, fundos imobiliários, criptomoedas e investimentos em renda fixa, oferecendo uma visão completa e organizada da carteira do investidor.

Por fim, destaca-se a integração com a API pública da "Brapi"[19], responsável por fornecer dados financeiros instantaneamente, como a cotação atual de ativos financeiros. Essa funcionalidade não só automatiza parte do preenchimento dos ativos como também garante a confiabilidade e a atualização constante das informações exibidas ao usuário.

VI. CONSIDERAÇÕES FINAIS

Com o desenvolvimento desta plataforma torna-se possível auxiliar investidores iniciantes ou intermediários na gestão de seus investimentos pessoais de forma centralizada, moderna e intuitiva, contribuindo para a tomada de decisões na hora de investir.

Também contribui para que seus usuários tenham uma visualização gráfica da evolução de seus investimentos, divididos por categorias, o sistema evita a poluição visual e mantém a organização.

O projeto foi baseado na simplicidade, utilizando boas práticas de desenvolvimento, separação em camadas e integração com serviços externos, como a API "Brapi" [19], que fornece dados do mercado financeiro de forma online. A plataforma conta, ainda, com funcionalidades como geração automática de lançamentos de compra ou venda de ativos e controle individual por usuário, promovendo maior eficiência e organização na gestão da carteira.

A metodologia aplicada, *Feature-Driven Development* (FDD) [23], possibilitou uma análise detalhada dos requisitos e a criação de diagramas e protótipos que orientaram o desenvolvimento da solução.

Dessa forma, o trabalho entregou uma solução eficiente para a gestão de investimentos, proporcionando maior praticidade,

organização e apoio à tomada de decisão para investidores iniciantes e intermediários.

REFERÊNCIAS

- [1] Redação Brasilprev. *Decisão de investimento: como fazer a escolha certa?* <https://blog.brasilprev.com.br/heuristicas-e-vieses-decisoes>. 2022.
- [2] Sandra Capomaccio. *Falta de planejamento financeiro é o principal motivo de endividamento das novas gerações.* <https://jornal.usp.br/atualidades/falta-de-planejamento-financeiro-e-o-principal-motivo-de-endividamento-das-novas-geracoes/>. 2024.
- [3] Equipe Toro Investimentos. *O que são ações? Conheça mais e descubra os principais tipos.* <https://blog.toroinvestimentos.com.br/bolsa/o-que-sao-acoes/>. 2025.
- [4] XP Investimentos. *O que são Fundos Imobiliários?* <https://www.xpi.com.br/produtos/fundos-imobiliarios/>. 2024.
- [5] Mariana Neves. *O que é criptomoeda e para que ela serve?* <https://blog.nubank.com.br/o-que-e-criptomoeda/>. 2024.
- [6] Itaú. *Renda fixa: o que é e como funciona?* <https://blog.itaú.com.br/artigos/renda-fixa>.
- [7] Alura. *Aprendendo a programar em C com Orientação a Objetos.* <https://www.alura.com.br/formacao-c-sharp-orientacao-objetos>.
- [8] Cristiano B. Noronha. *O que é um framework.* <https://balta.io/blog/o-que-e-um-framework>.
- [9] Microsoft. *Entity Framework Core.* <https://learn.microsoft.com/pt-br/ef/core/>. 2021.
- [10] Microsoft. *Introdução ao .NET.* <https://learn.microsoft.com/pt-br/dotnet/core/introduction>. 2024.
- [11] Microsoft. *O que é ASP.NET Core?* <https://dotnet.microsoft.com/pt-br/learn/aspnet/what-is-aspnet-core>.
- [12] Marylene Guedes. *O que é MVC?* <https://treinaweb.com.br/blog/o-que-e-mvc>. 2020.
- [13] Betrybe. *HTML, CSS e Javascript: quais as diferenças e como aprender?* <https://blog.betrybe.com/desenvolvimento-web/html-css-e-javascript>. 2024.
- [14] Guilherme Lima. *Bootstrap: O que é, Documentação, como e quando usar.* <https://www.alura.com.br/artigos/bootstrap>. 2023.
- [15] Larissa Gaspar. *jQuery: o que é e como usar.* <https://www.hostgator.com.br/blog/aprenda-o-que-e-jquery>. 2023.
- [16] Betrybe. *Git e GitHub: o que são, quais as diferenças e como usar na prática?* <https://blog.betrybe.com/tecnologia/git-e-github>. 2021.
- [17] Google Cloud. *O que é o MySQL?* <https://cloud.google.com/mysql?hl=pt-BR>.
- [18] Betrybe. *API: o que é, como funciona e exemplos práticos!* <https://blog.betrybe.com/desenvolvimento-web/api>. 2020.
- [19] Brapi. *Bem-vindo(a) à brapi! Nossa Missão.* <https://brapi.dev/about>.
- [20] Rodrigo Moraes dos Santos Pacheco. *Desenvolvimento de um software para gerenciamento de investimentos.* <https://docs.spring.io/spring-boot/docs/current/reference/html/>. Bom Sucesso MG. 2022.
- [21] Vitor Martins Basso. *Desenvolvimento de uma aplicação web para gestão de carteira de investimentos em bolsa de valores.* <https://repositorio.ufu.br/bitstream/123456789/33880/1/DesenvolvimentoAplicacaoWeb.pdf>. Uberlândia. 2021.
- [22] Bruno Souza Ramos. *Desenvolvimento de um aplicativo para controle financeiro pessoal.* https://www.monografias.ufop.br/bitstream/35400000/3134/6/Monografia_DesenvolvimentoAplicativoControle.pdf. João Monlevade MG. 2021.
- [23] Time Pontotel. *Conheça a metodologia FDD (Feature Driven Development) e entenda qual o seu papel no dia a dia das empresas.* <https://www.pontotel.com.br/fdd-feature-driven-development/>. 2023.