 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUI</p>	<p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ</p> <p>Curso: ADS</p> <p>Disciplina: Projeto Integrador II e Programação para a Internet II</p> <p>Professores: Ely Miranda e Rogério Silva</p>
--	---

Relatório de Projeto: FinnApp

Equipe:

- Marcos Eduardo de Neiva Santos

Sumário

- **Introdução**
 - 1.1 Apresentação do Projeto
 - 1.2 Objetivo do Relatório
- **Descrição do Projeto**
 - 2.1 Apresentação
 - 2.2 Público-alvo
 - 2.3 Justificativa e Impactos
- **Papéis e Responsabilidades**
 - 3.1 Desenvolvedor Back-end
 - 3.2 Front-end
 - 3.3 Product Owner
 - 3.4 Dev Ops
- **Descrição Geral do Processo**
 - 4.1 Ciclo de Vida
 - 4.1.1 Fase Inception
 - 4.1.2 Projeto e Prototipação
 - 4.1.3 Implementação
 - 5.1.1 Pesquisa de Mercado
 - 5.1.2 Descrição de Personas e Mapas de Empatia
 - 5.1.3 Visão de Produto
 - 5.1.4 Caixa do Produto
 - 5.1.5 Mapa de Navegação, Rabisco Frame
 - 5.1.6 Flat Backlog ou Backlog Preliminar
 - 5.2 Projeto e Prototipação
 - 5.2.1 Documento de Arquitetura
 - 5.2.2 Sprint Backlog
 - 5.2.3 Modelo Entidade Relacionamento ou Equivalente
 - 5.3 Implementação
 - 5.3.1 Modelo de Banco de Dados Físico ou Esquema
 - 5.3.2 Relatório de Bugs
 - 5.3.3 Principais Telas do Sistema Funcionando
 - 5.3.4 Link para o Repositório
- **Artefatos Gerados**
 - 5.1 Inception

- **Detalhamento da Implementação**

- 6.1 A Web e Desenvolvimento Web Tradicional
- 6.2 API Rest e Restful, API Contracts
- 6.3 Arquiteturas Baseadas em Camadas / Separação de Responsabilidades
- 6.4 Segurança: Autenticação e Autorização
- 6.5 Persistência via ORM
- 6.6 API GraphQL
- 6.7 Serviços em Nuvem / Cloud

- 6.8 Clientes Web com Frameworks e Bibliotecas de Mercado
- 6.9 Deploy de Aplicação Web

- **Conclusão**

- 7.1 Feedback do PO e/ou da Equipe
- 7.2 Lições Aprendidas
- 7.3 Trabalhos Futuros

- **Referências Utilizadas**

1. Introdução

1.1 Apresentação do Projeto

O FinnApp emerge como uma solução inovadora no cenário de aplicativos financeiros, visando proporcionar aos usuários um controle preciso e intuitivo de suas finanças pessoais. Este aplicativo surge da crescente necessidade de indivíduos de todas as faixas etárias e perfis profissionais por uma ferramenta acessível, poderosa e adaptável, que vá além da simples categorização de despesas.

1.2 Objetivo do Relatório

O objetivo deste relatório é fornecer uma visão abrangente e detalhada do processo de desenvolvimento do projeto FinnApp, realizado durante a disciplina de Projeto Integrador II. Pretende-se apresentar de maneira clara e concisa todos os elementos-chave, desde a concepção da ideia até a implementação efetiva do aplicativo.

Além disso, o relatório busca destacar a estrutura organizacional do projeto, incluindo os papéis individuais dos membros da equipe, as fases do ciclo de vida, e os artefatos gerados em cada etapa. Esta documentação visa não apenas retratar o progresso do FinnApp, mas também servir como um recurso informativo para futuros desenvolvedores, acadêmicos e interessados no domínio de gerenciamento financeiro pessoal por meio de aplicativos.

Em última análise, o relatório pretende contribuir para a compreensão holística do projeto FinnApp, compartilhando aprendizados, práticas bem-sucedidas e desafios enfrentados ao longo do caminho, fornecendo uma base sólida para avaliações críticas e oportunidades de melhorias contínuas.

2. Descrição do Projeto

2.1 Apresentação

O FinnApp surge como resposta à crescente necessidade dos usuários de terem um controle mais efetivo sobre suas finanças pessoais. Este aplicativo oferece um sistema completo de gerenciamento de gastos e receitas, proporcionando aos usuários uma visão clara e detalhada de sua situação financeira. Com funcionalidades como controle de despesas, acompanhamento de receitas e estabelecimento de metas personalizadas, o FinnApp visa simplificar a vida financeira dos usuários de maneira inovadora.

2.2 Público-alvo

O público-alvo do FinnApp abrange principalmente jovens adultos e profissionais que buscam um meio eficiente e intuitivo de gerenciar suas finanças pessoais. Com uma interface amigável e recursos adaptáveis, o aplicativo atende às necessidades de quem busca desde uma solução básica de controle até usuários mais experientes em busca de ferramentas avançadas para atingir metas financeiras específicas.

2.3 Justificativa e Impactos

A escolha do FinnApp como projeto surge da observação de lacunas no mercado de aplicativos financeiros pessoais, aliada à crescente demanda por soluções que vão além do básico. Acreditamos que o impacto potencial do FinnApp vai além do gerenciamento financeiro, podendo transformar a relação das pessoas com seu dinheiro, capacitando-as a tomar decisões mais informadas e alcançar objetivos financeiros de longo prazo.

3. Papéis e Responsabilidades

3.1 Desenvolvedor Back-end

- Responsável: Marcos Eduardo de Neiva Santos
- Implementar a lógica de negócio.
- Integração e manutenção de banco de dados.
- Configurar e realizar o deploy do servidor (API).

3.2 Front-end

- Responsável: Marcos Eduardo de Neiva Santos
- Desenvolvimento de uma interface de usuário intuitiva e atraente.
- Em colaboração com o desenvolvimento do servidor, realizar o consumo da API na aplicação.
- Realizar o deploy da aplicação.

3.3 Product Owner

- Responsável: Marcos Eduardo de Neiva Santos
- Definição e priorização de funcionalidades.
- Tomada de decisões estratégicas para garantir a entrega do produto.

3.4 Dev Ops

- Responsável: Marcos Eduardo de Neiva Santos
- Implementação de práticas de integração contínua e entrega contínua.
- Gerenciamento e manutenção da infraestrutura do projeto.

4. Descrição Geral do Processo

A implementação do projeto FinnApp seguiu um processo estruturado e iterativo, abrangendo diversas fases, desde a concepção da ideia até a avaliação final do produto. A seguir, apresentamos uma visão geral das principais etapas, destacando os artefatos produzidos em cada fase do desenvolvimento:

4.1 Ciclo de Vida

4.1.1 Fase Inception

Esta fase marcou o início do projeto, envolvendo pesquisas de mercado detalhadas para identificar demandas e oportunidades. Os resultados dessas pesquisas fundamentaram a concepção do FinnApp.

4.1.2 Projeto e Prototipação

Durante esta fase, as ideias foram traduzidas do papel para algo mais próximo da implementação em si.

4.1.3 Implementação

Com base no projeto e prototipação, a equipe procedeu à fase de Implementação, colocando em prática o que foi estabelecido na fase anterior, melhorando o que for necessário.

5. Artefatos Gerados

5.1 Inception

5.1.1 Pesquisa de Mercado

Briefing

Desenvolver a aplicação “FinnApp” com o objetivo de oferecer aos usuários uma plataforma intuitiva e abrangente para monitorar suas despesas e definir metas financeiras alcançáveis. A aplicação utilizará gráficos e análises detalhadas para capacitar os usuários a tomar decisões financeiras informadas e alcançar uma gestão financeira sólida.

Levantamento Histórico

Principais destaques:

- 1983: Lançamento do Quicken, um dos primeiros softwares de gestão financeira.
- 1996: A Microsoft lança o Money como concorrente do Quicken.
- 2006: Mint, uma plataforma de gestão financeira online, é lançada.
- 2010: Personal Capital introduz agregação de contas e planejamento financeiro baseado na web.
- 2014: O You Need A Budget (YNAB) se destaca com sua abordagem de orçamentação baseada em regras.
- 2016: Surgimento de assistentes financeiros virtuais, como o Albert.
- 2017: A Microsoft descontinua o Microsoft Money.
- 2020: Quicken incorpora IA e aprendizado de máquina em suas análises financeiras.

- 2021: Fintechs como Nubank e Banco Inter oferecem serviços bancários integrados com ferramentas de gestão financeira.
- 2022: Expansão das plataformas para integração com investimentos em criptomoedas.

Concorrentes

Os critérios de seleção escolhidos foram:

- a usabilidade
- funcionalidade
- feedback de usuários

Diretos

Mobills

Pontos Fortes

- Planejamento financeiro com metas e orçamentos;
- Gráficos e relatórios avançados;
- Controle de cartões de crédito;

Pontos Fracos

- A sincronização com a Web constantemente dá divergência com o smartphone.
- A listagem de despesas e receita pelo smartphone não permite classificação de nenhum tipo (maior ou menor valor, data, etc).

Organizze

Pontos Fortes

- O aplicativo conta com um design elegante e simples;
- Gerenciamento de contas, cartões, despesas e receitas;
- Disponível em multiplataformas;
- Ótima usabilidade.

Pontos Fracos

- lançamentos de forma manual ou por meio importação do extrato bancário.

Fast Budget

Pontos Fortes

- Sincronização bancária;
- Definição de lembrete para transações próximas do vencimento;
- Sincronização de até 5 dispositivos;
- Backup automático na memória do dispositivo;
- Função de calendário, em que é possível ver lucros, gastos e pagamentos agendados; entre muitas outras.

Pontos Fracos

- não possui uma função de dividir as despesas.
- possui muitos anúncios.
- não constam atualizações desde abril de 2021.

Indiretos

Planilhas eletrônicas

Pontos Fortes

- Flexibilidade para criar um formato personalizado.
- Pode ser adaptado para atender a várias necessidades.
- Não requer aprendizado de uma nova plataforma.

Pontos Fracos

- Menos automatizado e mais propenso a erros humanos.

- Difícil de gerar análises visuais avançadas.

Aplicativos de banco

Pontos Fortes





- Integração direta com contas bancárias e transações.
- Acesso a informações financeiras centralizadas.

Pontos Fracos

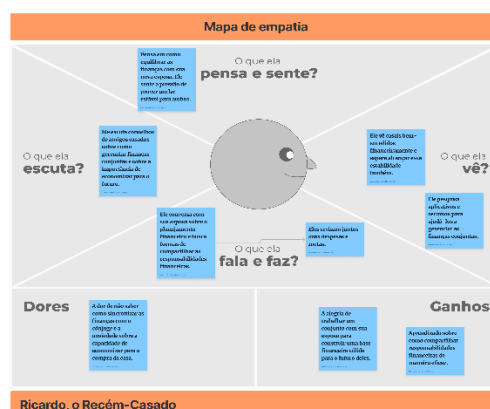
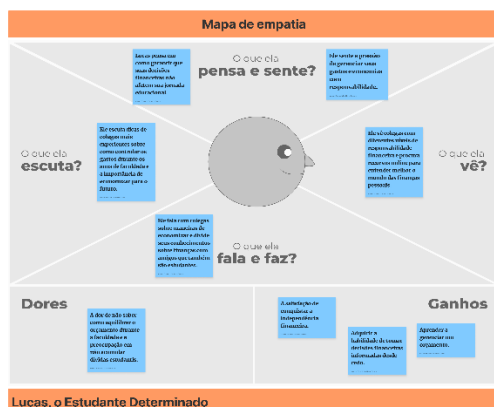
- Recursos limitados em comparação com aplicativos especializados.
- Pode não atender a necessidades complexas.

5.1.2 Descrição de Personas e Mapas de Empatia

Personas:

 <p>Lucas, o Estudante Determinado</p> <p>Idade: 21 anos Profissão: Estudante de Medicina</p> <p>"A disciplina me leva além"</p> <p>Objetivos</p> <ul style="list-style-type: none"> • Controlar os gastos durante a faculdade. • Economizar para viagens de estudo <p>Frustrações</p> <ul style="list-style-type: none"> • Dificuldade em gerenciar uma renda limitada. • Controlar impulsos de gastos. <p>Lucas é um estudante de medicina determinado a se formar sem acumular dívidas. Ele procura uma aplicação que o ajude a registrar seus gastos de maneira detalhada, criar um orçamento realista para suas despesas mensais e definir metas financeiras de curto e longo prazo. Gráficos claros o ajudarão a manter o foco em suas economias.</p>	 <p>Sofia, a Sonhadora Aventureira</p> <p>Idade: 30 anos Profissão: Fotógrafa Freelancer</p> <p>"A vida é uma aventura e eu sou a exploradora"</p> <p>Objetivos</p> <ul style="list-style-type: none"> • Gerenciar uma renda variável. • Economizar para viagens ao redor do mundo. <p>Frustrações</p> <ul style="list-style-type: none"> • Dificuldade em prever e planejar gastos devido à renda instável. <p>Sofia é uma fotógrafa freelancer apaixonada por viagens e exploração. Ela precisa de uma aplicação que lhe permita acompanhar sua renda variável, definir metas de economia flexíveis para diferentes destinos e ter uma visão geral de suas finanças ao longo do tempo. Gráficos intuitivos a ajudarão a visualizar seus avanços em direção aos destinos dos sonhos.</p>
 <p>Ricardo, o Recém-Casado</p> <p>Idade: 28 anos Profissão: Analista de TI</p> <p>"Construindo um lar e um futuro juntos."</p> <p>Objetivos</p> <ul style="list-style-type: none"> • Combinação de finanças após o casamento. • Economizar para comprar uma casa. <p>Frustrações</p> <ul style="list-style-type: none"> • Dificuldade em sincronizar finanças com o cônjuge. • Preocupações com a capacidade de economizar para grandes metas. <p>Ricardo é um analista de TI que recentemente se casou. Ele quer garantir uma base financeira sólida para seu novo lar e futuras metas. Ele procura uma aplicação que facilite o compartilhamento de finanças com seu cônjuge, ajude a economizar para a compra de uma casa e forneça uma visão clara das finanças conjuntas.</p>	 <p>Marina, a Jovem Empreendedora</p> <p>Idade: 26 anos Profissão: Proprietária de uma loja online de moda</p> <p>"Persistência é minha maior virtude."</p> <p>Objetivos</p> <ul style="list-style-type: none"> • Aumentar as vendas de sua loja. • Economizar para expandir o negócio. <p>Frustrações</p> <ul style="list-style-type: none"> • Dificuldade em equilibrar as finanças do negócio e as despesas pessoais. • Falta de tempo para planejamento detalhado. <p>Marina é uma empreendedora ambiciosa que abriu sua própria loja online de moda após anos de trabalho na indústria. Ela deseja uma aplicação que facilite o rastreamento das finanças do negócio, permitindo-lhe ver claramente o fluxo de caixa, definir orçamentos para despesas comerciais e pessoais, e analisar as tendências de vendas para tomar decisões estratégicas.</p>

Mapas de Empatia:



5.1.3 Visão de Produto

Para os indivíduos que desejam ter um controle adicional sobre suas finanças pessoais, analisar seus gastos e alcançar metas financeiras de maneira eficaz, o FinnApp é uma aplicação de Gestão Financeira Pessoal, que deve oferecer uma plataforma intuitiva e simples para monitorar despesas, criar orçamentos personalizados e definir metas financeiras ambiciosas, beneficiando-os de maneira distinta em comparação com as alternativas tradicionais. Alguns dos diferenciais do produto são a Visualização Abrangente e Intuitiva, a Orcamentação Personalizada e Definição de Metas.

5.1.4 Caixa do Produto

Benefícios

- Organização de ganhos e gastos.
- Estabelecimento e acompanhamento de metas

Funcionalidades

- **Monitoramento de Despesas:** A aplicação permite aos usuários registrar e categorizar todas as suas despesas de forma conveniente.
- **Criação de Orçamentos Personalizados:** Os usuários podem estabelecer orçamentos específicos para categorias de despesas, definindo limites de gastos mensais ou semanais para melhor controle financeiro.

- Definição de Metas Financeiras: A aplicação permite aos usuários definir metas de poupança, investimento ou pagamento de dívidas.
- Visualizações Gráficas Intuitivas: Gráficos e visualizações fáceis de entender apresentam os padrões de gastos, economias e progresso em direção às metas. Isso ajuda os usuários a ter uma visão clara das suas finanças.

5.1.5 Mapa de Navegação, Rabisco Frame

Mapa de Navegação:

Controlar gastos financeiros (O que sai).

Adicionar Gasto

Tela inicial --> Click no botão "Adicionar Gasto" --> Tela de adicionar gasto --> Escrever o valor e uma descrição(opcional) para o gasto --> Click no botão "Adicionar" --> Tela Inicial

Visualizar Gastos

Tela inicial --> Click no botão "Transações" --> Tela de Transações --> Click no botão "Filtrar" --> Selecione "Gastos" --> Tela de Transações

Controlar Receitas financeiras (O que entra).

Adicionar Transação (Entrada)

Tela inicial --> Click no botão "+" → Click no botão "Adicionar Transação" --> Tela de adicionar Transação --> Escrever o título, o valor, selecionar o tipo (Entrada ou Saída), selecionar a data da Transação e uma descrição(opcional) para o ganho --> Click no botão "Adicionar" --> Tela inicial

Visualizar Entradas

Tela inicial --> Click no botão "Transações" --> Tela de Transações --> Click no botão "Filtrar" --> Selecione "Entradas" --> Tela de Transações

Metas financeiras (Cofrinhos).

Adicionar Meta

Tela inicial --> Click no botão "+" → Click no botão "Adicionar Meta" --> Tela de adicionar meta --> Escrever o nome, o valor, a data limite e uma descrição(opcional) da meta --> Click no botão "Adicionar" --> Tela de metas

Visualizar Metas

Tela inicial --> Click no botão "Metas" --> Tela de metas

Rabiscos Frame:

Os Rabiscos Frames estão presentes em:

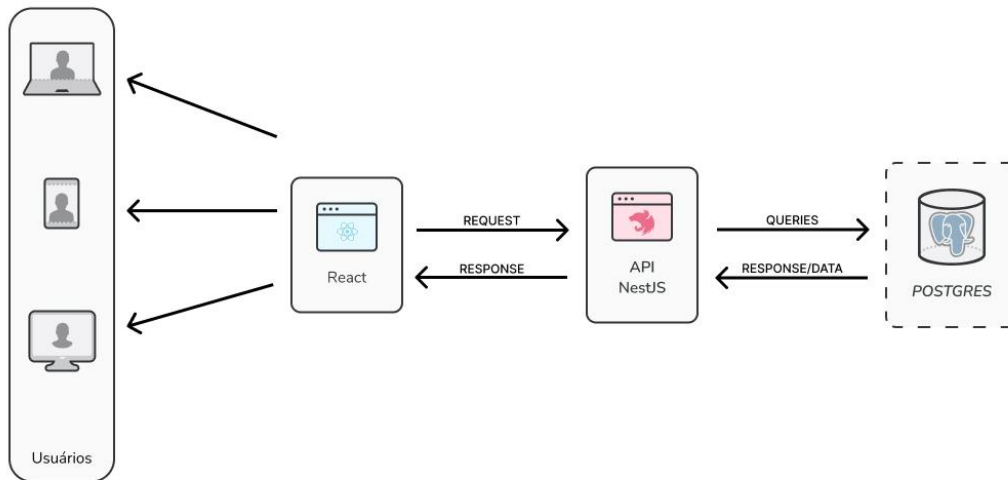
[RabiscosFrame](#)

5.1.6 Flat Backlog ou Backlog Preliminar

Pode ser acessado em: [Flat Backlog](#)

5.2 Projeto e Prototipação

5.2.1 Diagrama de Arquitetura



Informações adicionais aqui:

[Diagrama de arquitetura](#)

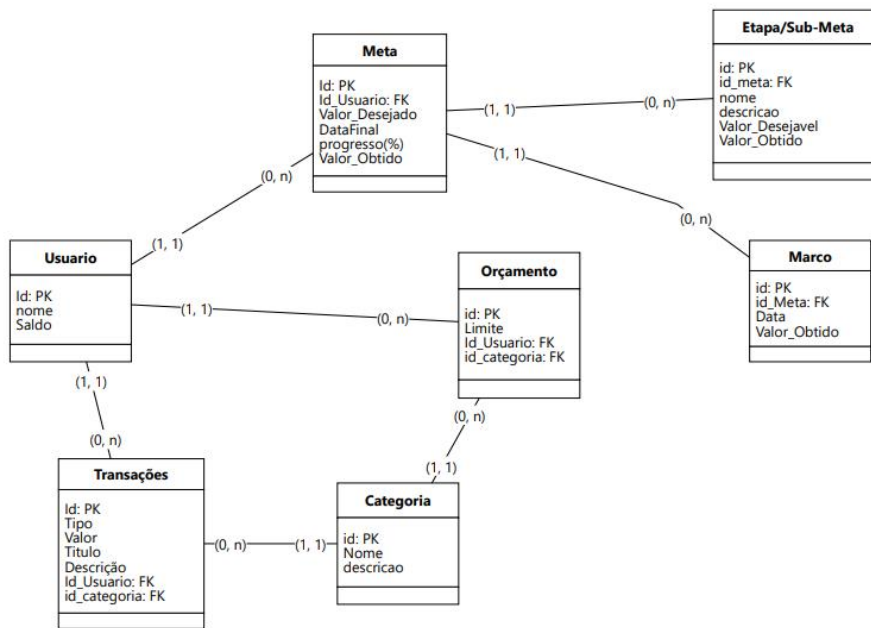
5.2.2 Sprint Backlog

O sprint Backlog pode ser acessado aqui:

[Sprint Backlog](#)

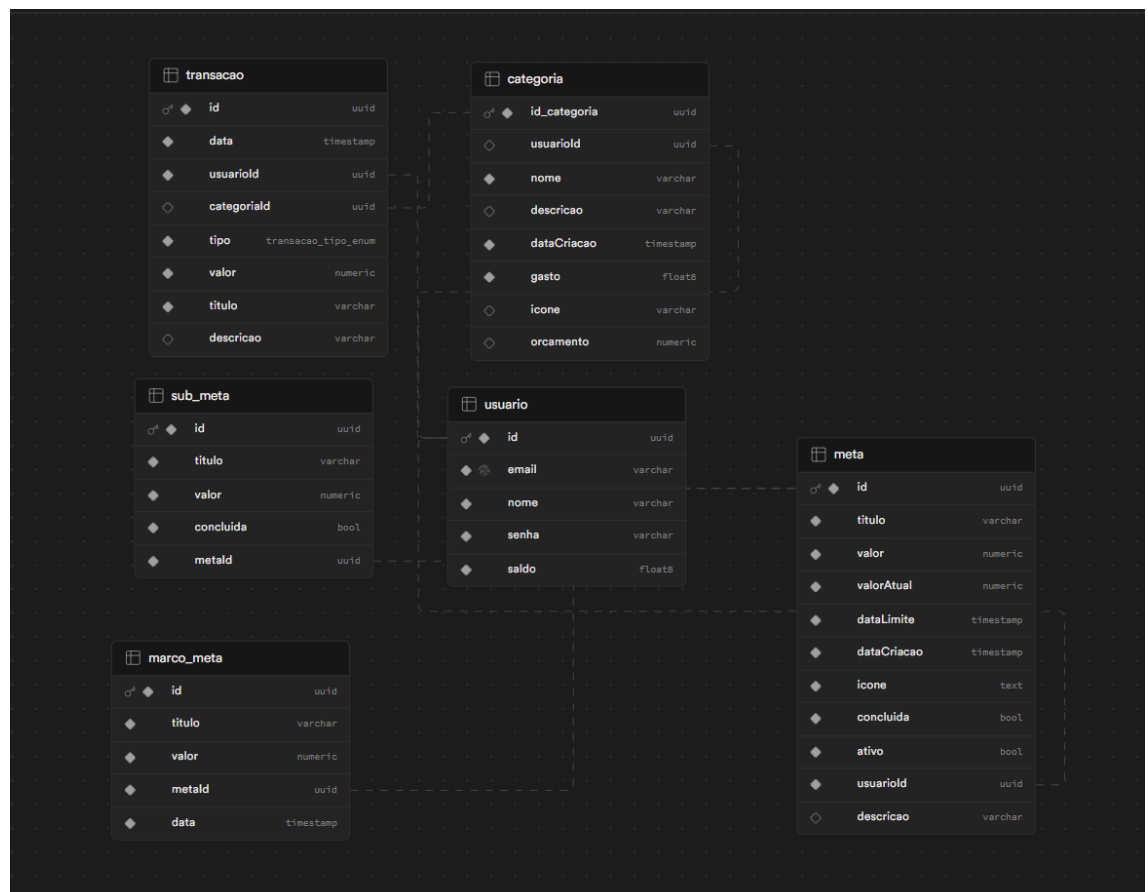
5.2.3 Modelo Entidade Relacionamento ou Equivalente

Diagrama de Dados base:



5.3 Implementação

5.3.1 Modelo de Banco de Dados Físico ou Esquema



5.3.2 Relatório de Bugs

Gráficos:

- Erros ao renderizar gráficos por conta de valores inválido e/ou incorretos recebidos da api → corrigido.

Transações:

- Erros ao adicionar e atualizar, por conta do envio de valores nulos ou indefinidos causados por equívocos durante o envio de dados -> corrigido.
- Erro ao atualizar saldo do usuário e gastos da respectiva categoria (caso seja gasto) -> corrigido.

TypeORM:

- Erros ao realizar consultas no banco de dados, retornando erros e travando a aplicação → corrigido
- Encerramento de conexão constante → resolvido parcialmente, pois em alguns casos específicos ainda ocorre (provavelmente deve ser por falta de alguma configuração...)

Api (De modo Geral)

- Excesso de consumo de memória, travando o sistema e apresentando falhas na api → Resolvido após uma refatoração nos métodos de consulta e de lógica de negócio.

5.3.3 Principais Telas do Sistema Funcionando







Categorias disponíveis



Ordenar por

Data de Criação



Pesquisar


Nome	Data de Criação	Gastos	Orçamento
 Teste	21 Nov 2023	R\$ 0	R\$ 1000
 Testes	21 Nov 2023	R\$ 0	R\$ 1000
 Teste	21 Nov 2023	R\$ 100	R\$ 500
 Lazer	12 Nov 2023	R\$ 15	-




1


...








Metas





Ordenar por


Data de Criação





Pesquisar


R\$ 1000.00
07 Dez 2023

Sei la


R\$ 10000.00
mil
28 Dez 2023

Testando 123

R\$ 10000.00
mil
28 Dez 2023

Testando 123

R\$ 1000.00
30 Nov 2023

Pagar as conta...

R\$ 1000.00
08 Dez 2023

Teste 123

R\$ 1000.00
29 Nov 2023

Teste 3





1


2

...


2








Transações










Ordenar por

Data da Transação



Pesquisar

Título	Data	Valor	tipo
 Teste	07 Dez 2023	R\$ 100	saida
 Teste	07 Dez 2023	R\$ 1.000	entrada
 Boteco ...	06 Dez 2023	R\$ 15	saida
 Depósito	05 Dez 2023	R\$ 1.200	entrada
 Teste	21 Nov 2023	R\$ 100	saida
 sla	21 Nov 2023	R\$ 100	entrada




1

2

3

...

4



Adicionar Transação

Título
Título da Transação

Valor **Data**
R\$ 0 08/12/2023

Tipo **Categoria**
Saída Seleccione uma categoria

Descrição
Descrição

Cancelar Adicionar

Adicionar Categoria

Nome
Nome da Categoria

Ícone
Barraquinha

☒ Definir orçamento?
Valor do orçamento
R\$ 0

Descrição
Descrição

Cancelar Adicionar Categoria

Adicionar Meta

Nome
Nome da Categoria

Ícone
Barraquinha

Valor da meta **Data limite**
R\$ 0 08/12/2023

Descrição
Descrição

Submetas Adicionar submeta

Cancelar Adicionar Categoria

Telas restantes:

- Detalhes → Transações, Categorias, Metas
- Perfil

5.3.4 Link para o Repositório

https://github.com/Marcos1701/Proj_integrador_II/tree/main

6. Detalhamento da Implementação

6.1 A Web e Desenvolvimento Web Tradicional

A interface web foi feita utilizando o framework React js.

6.2 API Rest e Restful, API Contracts

Desenvolvimento da API REST utilizando o framework NestJS para Node.js. Implementação de princípios RESTful para uma comunicação eficiente entre o frontend e o backend.

Contrato da api Rest: [Docs](#)

6.3 Arquiteturas Baseadas em Camadas / Separação de Responsabilidades

Implementação de uma arquitetura em camadas no backend para separar responsabilidades e facilitar a escalabilidade.

Utilização de Services para lógica de negócios, Controllers para gerenciar as requisições HTTP, services para implementar a lógica de negócio e Models para representar os dados.

6.4 Segurança: Autenticação e Autorização

Implementação de autenticação usando tokens JWT (JSON Web Token) para garantir segurança nas comunicações.

Os tokens gerados devem ser enviados via header (pelo cliente) e o servidor realiza a verificação do token, analisando se o token é válido, se pertence a um usuário existente e se o mesmo possui permissão para realizar o que está sendo solicitado na requisição.

6.5 Persistência via ORM

O typeORM do Nestjs foi utilizado para realizar o mapeamento, persistência e consultas das respectivas entidades do sistema. Além da utilização de alguns subscribers que executam ações e validações antes e/ou após alterações e consultas em determinadas entidades.

6.6 API GraphQL

Integração de uma API GraphQL para oferecer uma alternativa flexível às tradicionais APIs REST, foi utilizada em apenas um endpoint por efeito de teste.

Playground para visualização: <https://finnapp.onrender.com/graphql>

6.7 Serviços em Nuvem / Cloud

Hospedagem da aplicação em serviços especializados em cada necessidade:

Front-end: [Vercel](#) (Não está 100% funcional..)

Back-end: [Render](#)

6.8 Clientes Web com Frameworks e Bibliotecas de Mercado

Front-end:

- React js
 - Apache echarts → geração de gráficos
- Nest js
 - GraphQL

6.9 Deploy de Aplicação Web

Deploy realizado em:

Front-end: [Vercel](#) (Não está 100% funcional...)

Back-end: [Render](#)

7. Conclusão

7.1 Feedback do PO e/ou da Equipe

A aplicação abrange boa parte do que foi apresentado, apenas alguns detalhes como uma página específica para análise dos dados e a adição de marcos importantes às metas estão faltando, mas levando em conta os problemas ocorridos durante a evolução da aplicação, ela chegou em um patamar ótimo.

7.2 Lições Aprendidas

Organização a definição de bons requisitos são os principais diferenciais para a criação de um projeto duradouro e sustentável.

7.3 Trabalhos Futuros

Implementação das ferramentas restantes e a adição de análises que facilitem a análise e o controle das receitas e gastos do usuário, além de deixar a interface mais intuitiva.

8. Referências Utilizadas

[Artefatos criados durante a disciplina de Projeto Integrador II](#)