

# Projecto Gerenciamento Finanças Pessoal

## POCKET+

Nosso Projeto visa atender às nossas necessidades de controle financeiro de modo fácil e dinâmico focado em débitos e créditos diários onde podemos especificar os gastos do dia ou simplesmente colocar o total gasto no dia. Faremos também o controle de contas recorrentes com um alerta lembrando que essa conta precisa ser paga e um alerta para adicionar créditos recorrentes como Salários e outras fontes de renda fixas.

### Objetivos

- Fornecer de maneira simples e organizada todos os gastos e ganhos de um usuário diário e mensal levando em conta diferentes fontes de renda e diferentes classes de gastos.
- Produzir alertas e lembretes para o usuário de contas prestes a vencer cadastradas no programa dando a opção de fazer o débito de maneira simples dentro do sistema.
- Proporcionar metas para os usuários para manter o interesse no sistema e ajudar a alcançar um objetivo.
- Produzir relatórios mensais de gastos com diferentes configurações de análise baseando nas classes de de gastos, análise de débito e crédito e metas pessoais.

Poucas pessoas conseguem ter um total controle financeiro no seu dia a dia e por mais que já existam muitas soluções para esse problema nosso sistema se foca em resolver as questões financeiras de nosso grupo e com isso fornecer uma nova solução pensada para o dia a dia corrido te dando a opção de fazer um lançamento genérico e posteriormente poder especificá-lo ou não e o sistema ainda poderá fornecer dados relevantes que ajudam na tomada de decisão do nosso cotidiano.

### Metodologias

#### 1. Definição e Planejamento:

- Definição de Objetivos: Classificamos os objetivos do projeto, incluindo os recursos principais do sistema e as metas de entrega.
- Análise de Mercado: Pesquisaremos outros aplicativos de gerenciamento financeiro para identificar lacunas e oportunidades no que possam agregar ao nosso projeto ou que devemos corrigir .
- Levantamento de Requisitos: Refine as histórias de usuário e defina requisitos detalhados para cada funcionalidade do aplicativo. Cada um de nós como usuário e pessoas próximas responderam um questionário para fazer o levantamento de dados necessário para a análise e elaboração do projeto.
- Escopo do Projeto: Definimos o escopo inicial do projeto, definindo o que estará incluído na versão inicial e o que poderá ser adicionado em versões futuras.

#### 2. Design e Prototipagem:

- Design da Interface do Usuário (UI): Desenvolvemos um design simples e funcional, levando em consideração a experiência do usuário.

#### 3. Desenvolvimento:

- Escolha Tecnológica: Nosso sistema será desenvolvido para web. Usaremos para Figma para prototipação do projeto, MongoDB para modelagem de banco de dados, Node e React para desenvolvimento do front e back end.
- Desenvolvimento de Funcionalidades: Implementar as funcionalidades do aplicativo com base nos requisitos funcionais e não funcionais.
- Testes Unitários e de Integração: Realizar testes regulares para identificar e corrigir problemas de funcionamento e integração.

#### 4. Testes :

- Testes de Usabilidade: Conduzir testes com usuários reais para avaliar a facilidade de uso e a experiência geral do aplicativo.

#### 5. Lançamento:

- Preparação para Lançamento: Preparar a versão final do aplicativo para lançamento, incluindo a revisão final de funcionalidades, design e desempenho.

Fase 1: Definição e Planejamento (2 semanas) - Fase de documentação, pesquisa e revisão do projeto.  
Fase 2: Design e Prototipagem (3 semanas) - Fase de prototipação e escolha de layout.  
Fase 3: Desenvolvimento (9 semanas) - desenvolvimento do sistema e correções de bugs.  
Fase 4: Testes e Entrega do Projeto(2 semanas) - Últimos ajustes e testes do sistema.

17 semanas

Bruno Vinicius de Laia Silva (Front end)

Desenvolver uma interface baseada no protótipo de alta fidelidade, renderizado no Figma, com o objetivo de trazer a melhor experiência ao usuário. Contribuir também na apresentação do projeto.

Felipe Bacagini Couto (Banco de Dados)

Modelar e desenvolver o banco de dados não relacional que será integrado à aplicação.

Felipe Carvalho Tardivo (Designer e UI)

Utilizar o figma para desenvolvimento da prototipagem com objetivo de buscar uma experiência de uso fluida e interface harmônica.

Gabriel Araujo de Padua (Documentação)

Elaboração da documentação do sistema

Pesquisa

Testes

Marlon Vinicius de Souza (Back end)

Garantir conexão entre front-end e banco de dados, teste de todos os endpoints da aplicação, tratamentos relacionados aos status de resposta HTTP, testes unitários e de integração.

## **REQUISITOS**

Requisito	Categoria	Prioridade	Descrição
RF001 - MANTER CADASTRO DE USUÁRIO	<ul style="list-style-type: none"><li>• <u>Oculto</u></li><li>• <b><u>Evidente</u></b></li></ul>	<ul style="list-style-type: none"><li>• <b><u>Altíssima</u></b></li><li>• <u>Alta</u></li><li>• <u>Média</u></li><li>• <u>Baixa</u></li></ul>	O sistema deve fazer o cadastro do cliente com as informações básicas: Nome/senha/e-mail.
RF002 - MANTER DESPESAS	<ul style="list-style-type: none"><li>• <u>Oculto</u></li><li>• <b><u>Evidente</u></b></li></ul>	<ul style="list-style-type: none"><li>• <b><u>Altíssima</u></b></li><li>• <u>Alta</u></li><li>• <u>Média</u></li><li>• <u>Baixa</u></li></ul>	O sistema deve manter as despesas e rendimentos fornecidos pelo usuário de maneira simples e dinâmica.
RF003 - MANTER ORÇAMENTOS PRÉ-DEFINIDOS	<ul style="list-style-type: none"><li>• <u>Oculto</u></li><li>• <b><u>Evidente</u></b></li></ul>	<ul style="list-style-type: none"><li>• <u>Altíssima</u></li><li>• <b><u>Alta</u></b></li><li>• <u>Média</u></li><li>• <u>Baixa</u></li></ul>	O sistema deve fornecer a opção do usuário adicionar ou selecionar categorias para diferenciar suas despesas e rendimentos.

RF004 - MANTER CONSULTA DE RENDIMENTOS	<ul style="list-style-type: none"> <li>• <u>Oculto</u></li> <li>• <b>Evidente</b></li> </ul>	<ul style="list-style-type: none"> <li>• <u>Altíssima</u></li> <li>• <b>Alta</b></li> <li>• <u>Média</u></li> <li>• <u>Baixa</u></li> </ul>	O sistema deve fornecer de maneira simples as informações de débito e crédito do dia/mês/ano de acordo com o selecionado pelo usuário
RF005 - CADASTRO DE NOTIFICAÇÕES	<ul style="list-style-type: none"> <li>• <u>Oculto</u></li> <li>• <b>Evidente</b></li> </ul>	<ul style="list-style-type: none"> <li>• <u>Altíssima</u></li> <li>• <b>Alta</b></li> <li>• <u>Média</u></li> <li>• <u>Baixa</u></li> </ul>	O sistema deve fazer o cadastro de contas recorrentes e valores recebidos (como Salário) e mostrar uma notificação baseada no dia cadastrado perguntando se o valor foi recebido ou se a conta foi paga.
RF006 - ATUALIZAÇÕES AUTOMÁTICAS	<ul style="list-style-type: none"> <li>• <b>Oculto</b></li> <li>• <u>Evidente</u></li> </ul>	<ul style="list-style-type: none"> <li>• <u>Altíssima</u></li> <li>• <u>Alta</u></li> <li>• <b>Média</b></li> <li>• <u>Baixa</u></li> </ul>	O sistema deve verificar periodicamente por atualizações e baixá-las automaticamente para manter o aplicativo atualizado
RF007 - BACKUP AUTOMÁTICO DE DADOS	<ul style="list-style-type: none"> <li>• <u>Oculto</u></li> <li>• <u>Evidente</u></li> </ul>	<ul style="list-style-type: none"> <li>• <u>Altíssima</u></li> <li>• <u>Alta</u></li> <li>• <b>Média</b></li> <li>• <u>Baixa</u></li> </ul>	O sistema deve realizar backups automáticos dos dados financeiros dos usuários para garantir a recuperação em caso de falhas.
RF007 - TEMAS DE INTERFACE DE USUÁRIO	<ul style="list-style-type: none"> <li>• <u>Oculto</u></li> <li>• <b>Evidente</b></li> </ul>	<ul style="list-style-type: none"> <li>• <u>Altíssima</u></li> <li>• <u>Alta</u></li> <li>• <u>Média</u></li> <li>• <b>Baixa</b></li> </ul>	Os usuários podem escolher entre diferentes temas de interface para personalizar a aparência do aplicativo de acordo com suas preferências.
Requisitos	Categoria	Prioridade	Descrição
RNF001 - Tela login	<ul style="list-style-type: none"> <li>• <u>Desejável</u></li> <li>• <b>Obrigatório</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>Permanente</b></li> <li>• <u>Transitório</u></li> </ul>	O sistema deve ter uma tela de login para acesso de diferentes atores
RNF002 - Segurança	<ul style="list-style-type: none"> <li>• <u>Desejável</u></li> <li>• <b>Obrigatório</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>Permanente</b></li> <li>• <u>Transitório</u></li> </ul>	O sistema deve garantir a segurança dos dados financeiros dos usuários por meio de criptografia e autenticação robusta.
RNF003 - Desempenho	<ul style="list-style-type: none"> <li>• <u>Desejável</u></li> <li>• <b>Obrigatório</b></li> </ul>	<ul style="list-style-type: none"> <li>• <u>Permanente</u></li> <li>• <b>Transitório</b></li> </ul>	O sistema deve responder a consultas de saldo e relatórios em até 2 segundos, garantindo uma experiência de usuário rápida.

RNF04 - Escalabilidade	<ul style="list-style-type: none"> <li>• <u>Desejável</u></li> <li>• <u>Obrigatório</u></li> </ul>	<ul style="list-style-type: none"> <li>• <u>Permanente</u></li> <li>• <u>Transitório</u></li> </ul>	O sistema deve ser projetado para lidar com um aumento futuro de usuários e transações sem comprometer o desempenho.
RNF005 - Usabilidade	<ul style="list-style-type: none"> <li>• <u>Desejável</u></li> <li>• <u>Obrigatório</u></li> </ul>	<ul style="list-style-type: none"> <li>• <u>Permanente</u></li> <li>• <u>Transitório</u></li> </ul>	A interface do usuário deve ser intuitiva e de fácil navegação, permitindo que os usuários gerenciem suas finanças sem dificuldades.
RNF006 - Conformidade	<ul style="list-style-type: none"> <li>• <u>Desejável</u></li> <li>• <u>Obrigatório</u></li> </ul>	<ul style="list-style-type: none"> <li>• <u>Permanente</u></li> <li>• <u>Transitório</u></li> </ul>	O sistema deve estar em conformidade com regulamentações financeiras relevantes, como proteção de dados e privacidade.
RNF007 - Manutenção	<ul style="list-style-type: none"> <li>• <u>Desejável</u></li> <li>• <u>Obrigatório</u></li> </ul>	<ul style="list-style-type: none"> <li>• <u>Permanente</u></li> <li>• <u>Transitório</u></li> </ul>	O sistema deve ser projetado de forma modular para facilitar futuras atualizações e adições de recursos

### Casos de uso

Caso de Uso - Cadastro de Usuário	
ID	UC 01
Descrição	O usuário se cadastra no aplicativo para criar uma conta.
Ator Primário	Usuário
Pré-condição	O usuário não possui uma conta no aplicativo.
Cenário Principal	O usuário fornece informações de registro (nome, e-mail, senha) e confirma o cadastro.
Pós-Condição	O usuário é registrado no sistema e pode fazer login.

Cenário Alternativo	
---------------------	--

Caso de Uso - Inserir despesas	
ID	UC 02
Descrição	O usuário insere uma nova despesa no aplicativo podendo ou não atrelar a uma categoria.
Ator Primário	Usuário
Pré-condição	O usuário está autenticado no aplicativo.
Cenário Principal	O usuário seleciona a opção de adicionar uma nova despesa, insere o valor e se a transação é uma receita ou despesa e pode se quiser adicionar uma categoria, detalhes e confirmar.
Pós-Condição	
Cenário Alternativo	

Caso de Uso - Definir Orçamento Mensal	
ID	UC 03
Descrição	O usuário define um orçamento mensal para uma categoria específica.
Ator Primário	Usuário
Pré-condição	O usuário está autenticado e tem despesas registradas.

Cenário Principal	O usuário seleciona a categoria desejada, insere o valor do orçamento e salva.
Pós-Condição	O orçamento é definido para a categoria escolhida.
Cenário Alternativo	

Caso de Uso - Visualizar Relatório Mensal	
ID	UC 04
Descrição	O usuário visualiza um relatório mensal de gastos.
Ator Primário	Usuário
Pré-condição	O usuário está autenticado e tem despesas registradas
Cenário Principal	O usuário seleciona o mês desejado e visualiza um relatório com gastos por categorias.
Pós-Condição	O relatório mensal é exibido ao usuário.
Cenário Alternativo	

Caso de Uso - Receber Alerta de Orçamento Excedido	
ID	UC 05
Descrição	O usuário recebe um alerta quando seu gasto em uma categoria excede o orçamento definido.
Ator Primário	Sistema

Pré-condição	O usuário tem um orçamento definido e registra despesas.
Cenário Principal	O sistema detecta que o gasto excedeu o orçamento e envia um alerta por notificação ou e-mail.
Pós-Condição	O usuário é notificado sobre o orçamento excedido.
Cenário Alternativo	

Caso de Uso - Acompanhar Metas Financeiras	
ID	UC 06
Descrição	O usuário acompanha o progresso em relação a metas financeiras estabelecidas.
Ator Primário	Usuário
Pré-condição	O usuário está autenticado e definiu metas financeiras.
Cenário Principal	O usuário visualiza um painel com o progresso em relação às metas, incluindo gráficos e valores
Pós-Condição	O usuário obtém informações sobre suas metas financeiras.
Cenário Alternativo	

Caso de Uso - Realizar Autenticação Segura	
ID	UC 07

Descrição	O usuário realiza a autenticação segura para acessar o aplicativo.
Ator Primário	Usuário
Pré-condição	O usuário fornece credenciais de login.
Cenário Principal	O sistema verifica as credenciais e concede acesso se forem válidas.
Pós-Condição	O usuário é autenticado e pode acessar o sistema.
Cenário Alternativo	

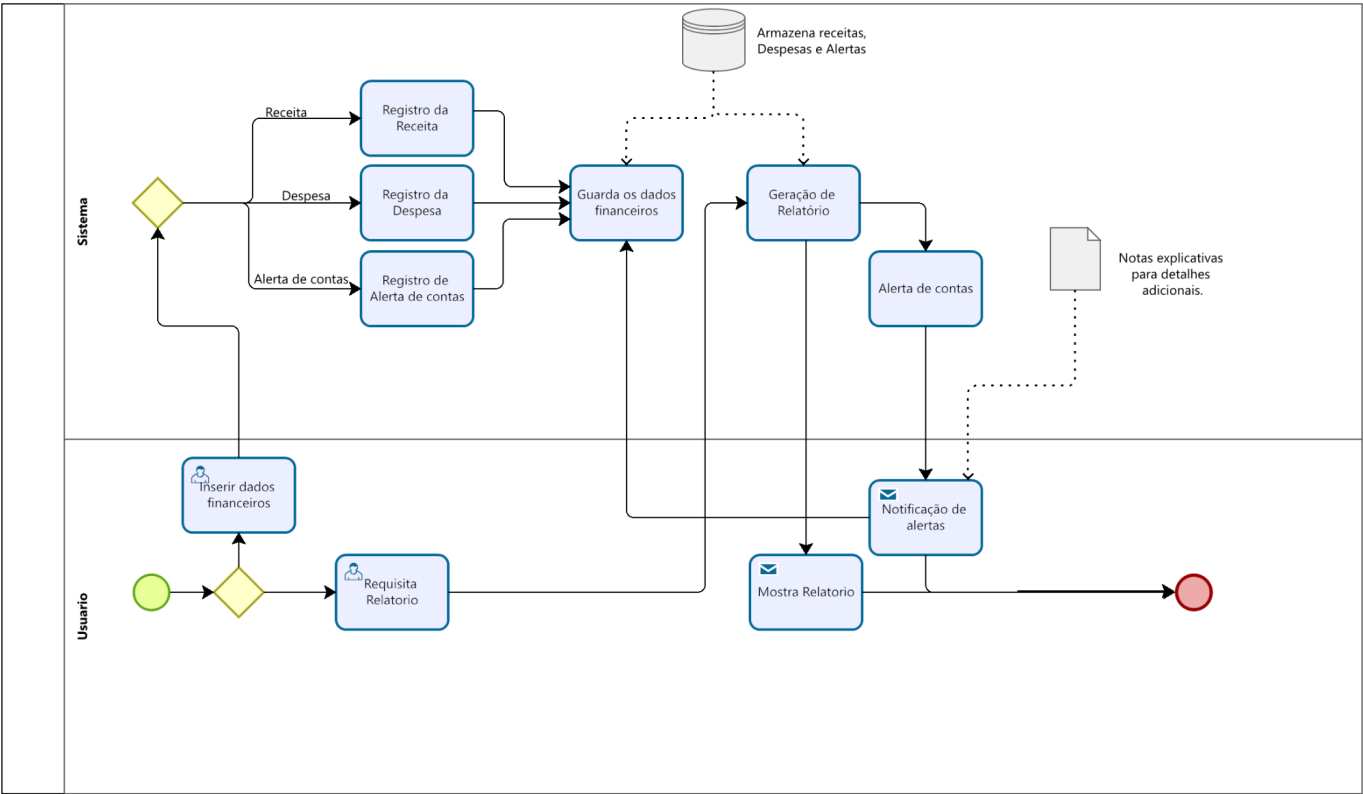
Caso de Uso - Alterar Orçamento Mensal	
ID	UC 08
Descrição	O usuário modifica um orçamento mensal existente para uma categoria.
Ator Primário	Usuário
Pré-condição	O usuário está autenticado e tem orçamentos definidos.
Cenário Principal	O usuário seleciona a categoria desejada, modifica o valor do orçamento e salva as alterações.
Pós-Condição	O orçamento é atualizado com o novo valor.
Cenário Alternativo	



Caso de Uso - Definir Lembrete Recorrente

ID	UC 09
Descrição	O usuário configura um lembrete recorrente para pagamentos regulares
Ator Primário	Usuário
Pré-condição	O usuário está autenticado e tem contas recorrentes a pagar.
Cenário Principal	O usuário seleciona a conta, define a frequência (semanal, mensal, etc.) e agenda o lembrete.
Pós-Condição	Um lembrete recorrente é configurado para a conta especificada.
Cenário Alternativo	

BPMN



No decorrer do projeto tivemos o desligamento de um colega e devido a isso tivemos que replanejar nossa estrutura de desenvolvimento da seguinte forma:

Bruno Vinicius de Laia Silva (Front end)

Desenvolver uma interface baseada no protótipo de baixa fidelidade, renderizado no Figma, com o objetivo de trazer a melhor experiência ao usuário. Contribuir também na apresentação do projeto.

Felipe Carvalho Tardivo (Designer e UI)

Utilizar o Figma para desenvolvimento da prototipagem com objetivo de buscar uma experiência de uso fluida e interface harmônica.

Desenvolver o banco de dados não relacional que será integrado à aplicação.

Gabriel Araujo de Padua (Documentação)

Transcrever os Objetivos a serem atingidos, definição e planejamento do projeto, análise de mercado, levantamento de requisitos, escopo inicial e escolha de tecnologias, divisão do projeto em fases, com estimativas de tempo para cada fase, totalizando 17 semanas. Identificação de cinco melhores práticas a serem seguidas na aplicação, incluindo segurança de dados, usabilidade e educação financeira. Elaboração da documentação usando a notação BPMN, considerando várias interações do usuário e a lógica do processo também definindo os Requisitos, Casos de uso e História de Usuário.

Marlon Vinicius de Souza (Back end)

Garantir conexão entre front-end e banco de dados, teste de todos os endpoints da aplicação, tratamentos relacionados aos status de resposta HTTP, testes unitários e de integração.

Modelagem do Banco de dados e desenvolver o banco de dados não relacional que será integrado à aplicação.

Como referencial teórico temos alguns aplicativos para celular que oferecem as mesmas funcionalidades que estamos desenvolvendo. Nosso diferencial é que não é preciso preencher nada além do valor e tipo de conta para fazer um lançamento, o que por sua vez facilita muito o uso do aplicativo no dia a dia e dando a opção do usuário fornecer essas informações posteriormente.

Nosso projeto está com a documentação inicial finalizada feita pelo colaborador Gabriel Araújo, Plano de identidade visual e prototipação no Figma em baixa fidelidade finalizados pelo colaborador Felipe Carvalho. Estudo de ferramentas para o desenvolvimento preciso da aplicação como também modelos funcionais em Javascript feitos pelo colaborador Bruno Vinicius. Modelagem de banco de dados e estudo sobre tecnologias para desenvolver o Backend e também as ligações com o Frontend e Banco de dados feita pelo colaborador Marlon Vinicius. Todo o nosso projeto está dentro do cronograma tirando a parte do protótipo em alta fidelidade que decidimos deixar de lado e começar a desenvolver o código do projeto.

### **Referencial teórico**

Aqui traremos algumas matérias e exemplos de aplicações com o mesmo propósito e também incluindo literatura para trazer embasamento ao projeto.

#### **Artigo acadêmico**

[https://admpg2018.com.br/anais/2018/arquivos/06032018\\_180644\\_5b14614c290b8.pdf](https://admpg2018.com.br/anais/2018/arquivos/06032018_180644_5b14614c290b8.pdf)

#### **Links e pesquisas**

Matéria 99

<https://99app.com/blog/99pay/controle-financeiro-pessoal/>

BV financeira

<https://www.bv.com.br/bv-inspira/credito-consignado-privado/controle-de-financas-pessoais>

Revista Exame

<https://exame.com/invest/minhas-financas/5-aplicativos-controlar-gastos-2023/>

ONZE

<https://www.onze.com.br/blog/app-de-controle-financeiro/#:~:text=Para%20que%20serve%20um%20app,elas%20sejam%20alcan%C3%A7adas%20mais%20facilmente.>

Nielsen Norman Group (NN/g)

<https://www.nngroup.com/>

Livros

**Não me faça pensar:** atualizado Edição Português por Steve Krug (Autor)

**A Transformação Total Do Seu Dinheiro:** Um Plano Eficaz Para Alcançar O Bem-estar Financeiro  
Edição Português | por Dave Ramsey, Ulisses Trevisan Palhavan, e outros. | 17 mai. 2021