

Plano de Gerência de Software

Sistema de Gerenciamento Financeiro

Versão: Preliminary

Histórico de Revisões

Versão	Autor	Descrição	Data
Inicial	Paulino Alexandre	Elaboração da primeira versão do plano de gerência	20/09/2024
Review	Paulo Victor	Revisão da versão Inicial	25/10/2024
Preliminary	Sávio Sáron	Implementação de Riscos e Mecanismos de Controle	15/11/2024
Final			

1. Introdução

Projeto de software a ser desenvolvido por alunos do 4º ano do curso de Informática para Internet do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – IFRN, como requisito da disciplina “Projeto de Desenvolvimento de Sistemas para Internet”.

1.1. Resumo do projeto

Este projeto propõe um software para controle de despesas financeiras, onde cada indivíduo pode registrar os gastos que teve ao longo do mês. O sistema gera relatórios que permitem o monitoramento dos gastos.

Cada usuário terá um perfil próprio para adicionar, editar e excluir suas despesas, categorizando-as conforme necessário. O software também permite que

membros de uma mesma casa, por exemplo, compartilhem e visualizem as categorias com maiores despesas e identifiquem quem está gastando mais.

O objetivo é facilitar o planejamento financeiro e melhorar a transparência na gestão de gastos.

1.2. Entrega de tarefas

Data	Marco	Tarefas desenvolvidas	Gerente
11/09/2024	I	Definição de Tema e Escopo	Paulino
20/09/2024	I	Elaboração do Plano de Gerência	Paulino
11/10/2024	II	Criação do repositório GitHub	Paulo Victor
25/10/2024	II	Primeira versão do API (Iteração 1)	Paulo Victor

1.3. Evolução do plano de gerência de software

Espera-se que as tarefas planejadas sejam cumpridas e entregues dentro do prazo destinado de desenvolvimento.

Versão	Autor	Descrição	Data
Inicial	Paulino Alexandre	Primeira versão do plano de gerência	20/09/2024
Review	Paulo Victor	Versão revisada do plano de gerência	25/10/2024
Preliminary	Sávio Sáron	Versão revisada do plano de gerência	15/11/2024

1.4. Documentos

Os documentos e artefatos criados estarão no seguinte repositório GitHub:

https://github.com/PauloV82/gestao_financeira.git

2. Organização do Projeto

2.1. Processo

O processo utilizado para o desenvolvimento do projeto será o RAD (Desenvolvimento Rápido de Aplicações), que prioriza a criação rápida de protótipos funcionais e o feedback contínuo. A equipe é composta por Paulino Alexandre, Paulo Victor e Sávio Saron. O projeto será dividido em ciclos curtos, onde as funcionalidades serão rapidamente desenvolvidas e avaliadas. No início de cada ciclo, haverá uma reunião com o cliente (indefinido) para priorizar as funcionalidades a serem implementadas.

A equipe seleciona as tarefas que podem ser desenvolvidas rapidamente e, ao concluir uma tarefa ou encontrar impedimentos, os membros farão um breve relatório e informarão os demais através do Whatsapp. Essa abordagem garantirá que o produto final atenda às necessidades dos usuários, com ajustes feitos de acordo com o feedback recebido em cada iteração.

2.2. Histórico de tarefas

Tarefa	Data	Poderá haver mudanças?	Entregue ao cliente?	Autor
Definição de Tema e Escopo	11/09/2024	Não	Não	Sávio Saron
Plano de Gerência	20/09/2024	Sim	Não	Paulino Alexandre
Primeira versão do API (Iteração 1)	25/10/2024	Não	Não	Paulino Alexandre

Projeto de Baixa Fidelidade	14/11/2024	Sim	Não	Paulo Victor

2.3. Estrutura Organizacional

Marco I		
Integrante	Função	Responsabilidades
Paulino Alexandre	Gerente	Elaboração do plano de gerência
Sávio Sáron	Analista	Definir o escopo
Paulo Victor	Analista	Pesquisar referências
Marco II		

Integrante	Função	Responsabilidades
Paulo Victor	Gerente	Revisar plano de gerência, criar repositório GitHub e coordenar o processo de especificações
Paulino Alexandre	Desenvolvedor	Iniciar o desenvolvimento back-end
Sávio Sáron	Analista de negócios	Definir especificações, objetivos e público alvo
Marco III		
Integrante	Função	Responsabilidades
Sávio Sáron	Gerente	Coordenar o processo de definição de identidade visual e a implementação de requisitos
Paulino Alexandre	Desenvolvedor front-end	Projetar a interface (front-end) com base no projeto de baixa fidelidade
Paulo Victor	Desenvolvedor back-end	Implementar banco de dados e fazer testes de funcionalidades
Marco IV		
Integrante	Função	Responsabilidades
Paulo Victor	Gerente	Coordenar o desenvolvimento de funcionalidades
Sávio Sáron	Desenvolvedor full-stack	Implementar e testar funcionalidades conectando-as com o front-end
Paulino	Desenvolvedor back-end	Implementar e testar as funcionalidades

Apresentação Banca		
Integrante	Função	Responsabilidades
Sávio Sáron	Gerente	Apresentar e justificar o projeto desenvolvido
Paulo Victor	Responsável front-end	Complementar a apresentação, mostrando as estratégias usadas na identidade visual
Paulino	Responsável back-end	Complementar a apresentação, mostrando as ferramentas usadas nas funcionalidades

3. Processo Gerencial

3.1. Planejamento

Marco	Data Inicial	Data Final
I	05/09/2024	20/09/2024
Elaboração do plano de gerência.		
II	07/10/2024	21/10/2024
Identificação do público-alvo; Discriminação das funcionalidades e objetivos do projeto; Iniciação no processo de desenvolvimento.		
III	22/10/2024	06/11/2024
Criação do banco de dados; Criação do front-end; Implementação de funcionalidades e testes.		

IV	07/11/2024	21/11/2024
Aperfeiçoamento no processo das funcionalidades; Teste das funcionalidades; Aperfeiçoamento do front-end.		
Encerramento	21/11/2024	22/11/2024
Entrega dos documentos requeridos pelos coordenadores;		
Banca	23/11/2024	24/11/2024
Apresentação do projeto desenvolvido para a banca avaliadora;		

3.2. Objetivos e prioridades

- Garantir o bom funcionamento do sistema e sua comunicação com os documentos e diagramas.
- Garantir a entrega dos processos sempre dentro do prazo.

3.3. Riscos

3.3.1. Alto

- Inconsistências de dados (Requisitos e Tecnologia): com efeito sério, pois erros nos cálculos e relatórios financeiros prejudicam a confiança no sistema.
- Falhas de segurança (Tecnologia): com efeito catastrófico, já que um vazamento de dados financeiros pode resultar em perda de confiança, sanções legais e danos à reputação.

3.3.2. Médio

- Falta de adesão dos usuários (Organizacional e Pessoaç): com efeito sério, podendo prejudicar o crescimento do projeto e sua

viabilidade a longo prazo.

- Problemas de desempenho (Tecnologia e Estimativas): têm efeito tolerável, afetando a experiência do usuário, porém sem comprometer a segurança ou a integridade dos dados.

3.4. Mecanismos de controle

- Segurança: Implementar criptografia e autenticação para proteger dados sensíveis e assegurar a privacidade do usuário.
- Validação de dados: Usar verificações constantes para garantir a precisão dos registros financeiros e prevenir possíveis erros de cálculo.
- Monitoramento de desempenho: Implementar alertas, acionados ao perceber oscilações do sistema e visíveis somente aos desenvolvedores.
- Feedback de usuários: Promover testes constantes de usabilidade e coletar feedback para melhorias contínuas.

4. Processo Técnico

4.1. Métodos, Ferramentas e Técnicas

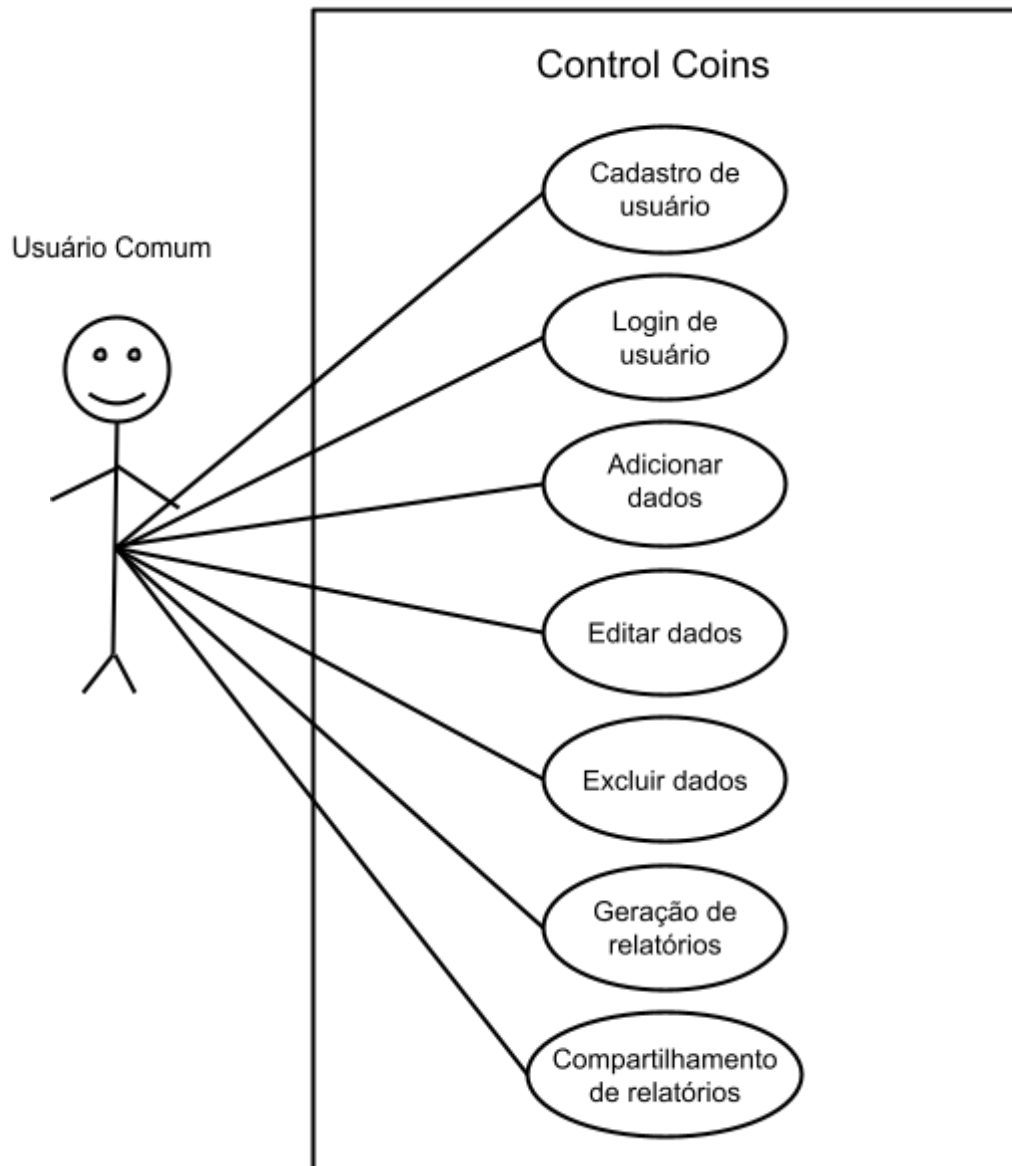
Neste projeto web, utilizaremos uma combinação de tecnologias para garantir um desenvolvimento eficiente e integrado. O backend será estruturado com o framework Laravel. Para o front-end, utilizaremos o React, permitindo a criação das interfaces responsivas.

O armazenamento do código-fonte ocorrerá no GitHub, enquanto a comunicação será realizada através de e-mail, discord e WhatsApp, assegurando a troca de informações de maneira eficaz.

4.2. Artefatos

- Plano de gerência.

- Diagrama de casos de uso.



- Código-fonte.
- Interfaces.

5. Requisitos

Os Requisitos Funcionais estão relacionados à maneira de como o sistema deverá operar e o que os clientes precisam que ofereça, ou seja, são requisitos que estão diretamente ligados à funcionalidade do sistema.

Os Requisitos Não-Funcionais são padrões que não envolvem diretamente a funcionalidade do sistema, mas garantem seu desempenho, segurança e disponibilidade. Eles definem as qualidades do sistema, tais como tempo de resposta, disponibilidade,

compatibilidade e nível de segurança.

5.1 Requisitos funcionais (RFs)

ID	Requisitos	Descrição
[RF01]	Cadastro de Usuário	Permitir que usuários façam o cadastro em um banco de dados
[RF02]	Registro de Despesas	Permitir que usuários registrem despesas, incluindo detalhes como valor, categoria, data e descrição
[RF03]	Edição de Despesas	Usuários podem editar os registros de suas despesas.
[RF04]	Exclusão de Despesas	Usuários podem excluir despesas do sistema.
[RF05]	Listagem de Despesas	Listar todas as despesas do usuário, com possibilidade de filtros por categoria, data e valor.
[RF06]	Geração de Relatórios	O sistema gera relatórios mensais que destacam os gastos por categoria e totais.
[RF07]	Login de Usuário	Verificação de credenciais de login, com suporte para autenticação por e-mail e senha.
[RF08]	Perfis Compartilhados	Usuários podem compartilhar suas despesas com outros perfis.
[RF09]	Visualização de Relatórios Compartilhadas	Permitir que usuários autorizados visualizem relatórios e detalhes de despesas compartilhadas.
[RF10]	Cadastro e Gestão de Categorias (tags)	Adicionar, editar e remover categorias de despesas.
[RF11]	Login com Conta Suap	Possibilidade de login usando conta do Suap para

		facilitar o acesso.
[RF12]	Notificações de Alerta de Gastos	Alertar usuários por e-mail ou notificações push quando os gastos excederem um limite definido.
[RF13]	Referências de Informações Relacionais	Mensagens mostrando relações do mês atual com os meses anteriores.

5.2 Requisitos Não-Funcionais (RNFs)

ID	Requisitos	Descrição
[RNF01]	Suporte a Usuários Simultâneos	Capacidade de atender a múltiplos usuários simultaneamente sem queda de desempenho.
[RNF02]	Facilidade de Uso	O sistema deve ter uma interface intuitiva que permita aos novos usuários aprender as funcionalidades principais em menos de 5 minutos.
[RNF03]	Armazenamento Criptografado de Senhas	As senhas dos usuários devem ser criptografadas para maior segurança.
[RNF04]	Documentação do Código Fonte	O código deve ser documentado de forma adequada para facilitar a manutenção futura.