

Gaste Menos - Aplicação de Gerenciamento Financeiro Pessoal

Alunos: Gilvo Rodrigues da Silva Junior, João Vitor do Nascimento, Lucas da Silva
Ferreira

HISTÓRICO DE ALTERAÇÕES

Versão	Alteração	Autor(es)
v1.0	Versão inicial do documento	Gilvo R. da S. Junior; J. Vitor do Nascimento; Lucas da S. Ferreira
v1.1	Detalhamento do diagrama de classes	João Vitor do Nascimento; Lucas da Silva Ferreira
v1.2	Adição de tópicos no sumário	Lucas da Silva Ferreira
v1.3	Adição do tópico 'Projeto de Banco de dados', 'Tecnologia' e 'Evolução do Sistema'	Gilvo Rodrigues da Silva Junior
v1.4	Revisor do documento	Lucas da Silva Ferreira

Sumário

1 - Descrição Geral do Projeto.....	03
2 - Análise.....	03
2.1 - Estudo de Viabilidade.....	03
2.2 - Elicitação de Requisitos.....	03
2.3 - Especificação de Requisitos.....	04
2.3.1 - Requisitos Funcionais.....	04
2.3.2 - Requisitos Não-Funcionais.....	05
2.3.3 - Validação de Requisitos.....	06
3 - Projeto.....	06
3.1 - Projeto de Arquitetura.....	06
3.2 - Projeto de Algoritmos.....	07
3.3 - Projeto de Interface com o Usuário.....	08
3.4 - Projeto de Banco de Dados.....	10
3.5 - Processo de Desenvolvimento.....	10
4 - Tecnologia.....	10
5 - Evolução do sistema.....	10
6 - Link do repositório no GitHub.....	10

1 - Descrição Geral do Projeto

Atualmente, a educação financeira é uma preocupação de quase todas as pessoas de diferentes classes sociais, porém nem todas as pessoas têm uma orientação mínima de como organizar seus gastos pessoais. O 'Gaste Menos' é uma aplicação de gerenciamento financeiro pessoal simples que fornece ao usuário a inserção dos seus gastos e visualizar os mesmos de forma mensal ou por categoria.

2 - Análise

Descrição dos detalhes de análise de requisitos.

2.1 - Estudo de Viabilidade

As funcionalidades realizadas pelo usuário são tecnicamente simples de serem implementadas. Java, a linguagem escolhida para a produção do software em questão, dá suporte para tudo que o sistema engloba, fazendo com que seja totalmente viável a produção desse software.

2.2 - Elicitação de Requisitos

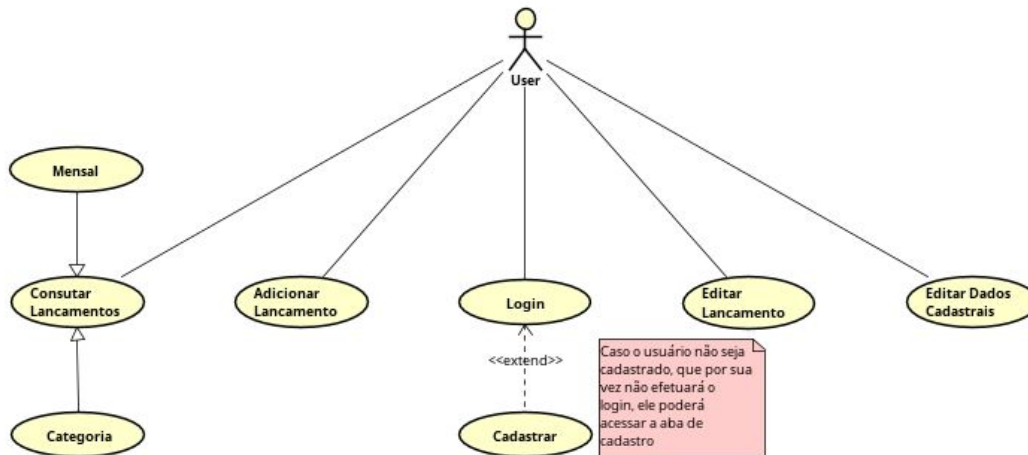
Foi adotado como método de elicitação, o brainstorm, captando várias idéias dos stakeholders, para que assim fossem filtradas as funcionalidades prioritárias para os usuários/comunidade.

Outros sistemas semelhantes foram utilizados como base para o levantamento de requisitos do sistema.

2.3 - Especificação de Requisitos

Descrição da atuação de cada requisito no sistema.

2.3.1 - Requisitos Funcionais



Nome: Login

Descrição: Executar a operação Login, pedindo ao usuário seus dados

Ator Primário: Usuário

Pré-Condições: Estar cadastrado

Fluxo principal: A tela inicial é exibida, o usuário clica na opção login, é exibido um formulário, com todos os campos necessários para que o usuário efetue seu login (cpf ou e-mail e senha), depois de preenchido esses campos, o usuário conclui a operação login e o sistema retorna o resultado

Fluxo de exceção: O cliente não adiciona os dados corretamente

Pós-condições: O usuário observa na tela, seus dados e as opções seguintes do sistema

Nome: Cadastrar

Descrição: Executar a operação cadastrar, pedindo ao usuário seus dados

Ator Primário: Usuário

Pré-Condições: -

Fluxo principal: A tela inicial é exibida, o usuário clica na opção cadastrar, é exibido um formulário, com todos os campos necessários para que o usuário efetue seu cadastro (nome, cpf, e-mail, data nascimento, senha), depois de preenchido esses campos, o usuário conclui a operação cadastrar e o sistema retorna o resultado

Fluxo de exceção: O cliente não adiciona os dados corretamente

Pós-condições: O usuário observa na tela, seus dados e as opções seguintes do sistema

Nome: Adicionar lançamento

Descrição: Executar a operação adicionar lançamento, pedindo ao usuário o tipo do produto e seu valor

Ator Primário: Usuário

Pré-Condições: Estar logado no sistema

Fluxo principal: A tela inicial é exibida, o usuário clica na opção login, é exibido um formulário, com todos os campos necessários para que o usuário efetue seu login (cpf ou e-mail e senha), depois de preenchido esses campos, o usuário conclui a operação login. Depois de logado, o usuário clica na opção adicionar lançamento, e abrirá outra tela com os campos da classe do produto e seus valores para serem adicionados.

Fluxo de exceção: -

Pós-condições: O usuário observa na tela, os dados de seus lançamentos adicionados

Nome: Visualizar lançamento

Descrição: Executar a operação visualizar lançamento, abrindo o campo com os dados já cadastrados

Ator Primário: Usuário

Pré-Condições: Estar logado no sistema e ter dados de lançamentos adicionados

Fluxo principal: A tela inicial é exibida, o usuário clica na opção login, é exibido um formulário, com todos os campos necessários para que o usuário efetue seu login (cpf ou e-mail e senha), depois de preenchido esses campos, o usuário conclui a operação login. Depois de logado, o usuário é direcionado para a tela principal, já sendo exibidos os lançamentos adicionados anteriormente.

Fluxo de exceção: -

Pós-condições: O usuário observa na tela, os dados de seus lançamentos

2.3.2 - Requisitos Não-Funcionais

Cada CPF efetua somente um cadastro:

Cada CPF deve ser único no sistema, fazendo com que só seja possível realizar um único cadastro, para que não exista pessoas diferentes com o mesmo CPF.

Segurança:

Deve ser mantido em sigilo, os dados dos usuários, principalmente senhas e CPF.

Efetuar cálculo dos valores dos lançamentos:

Quando um usuário adicionar vários dados, ele poderá realizar a soma de todos os lançamentos, para ter um balanço geral de quanto gastou durante um período específico.

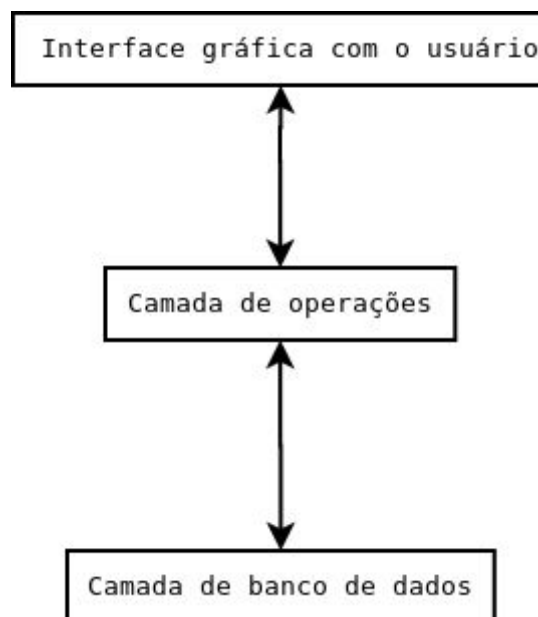
2.3.3 - Validação de Requisitos

Os requisitos foram levantados e listados em um documento, depois disso foram todos analisados um a um e consequentemente validados.

Todas as descrições detalhadas dos requisitos, podem ser facilmente encontradas no documento.

3 - Projeto

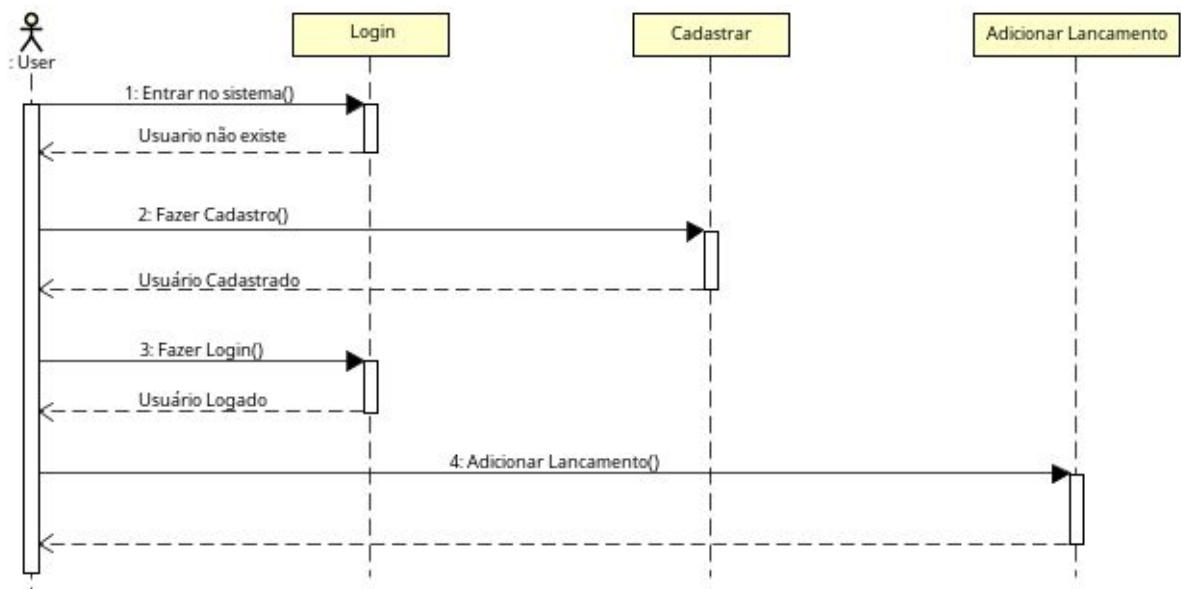
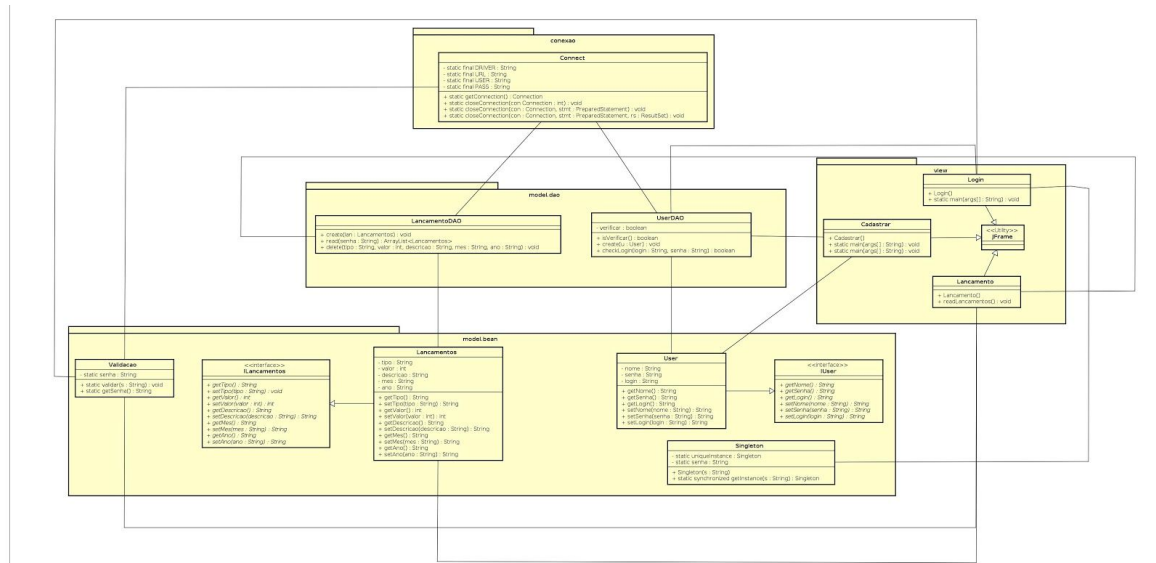
Descrição do projeto de software.

3.1 - Projeto de Arquitetura

Interface Gráfica com o usuário (UI) - subsistema responsável pela apresentação e captura dos dados.

Camada de operações - subsistema responsável pela execução das operações utilizando os dados capturados na UI.

Camada de banco de dados - subsistema responsável pela persistência dos dados.



3.2 - Projeto de Algoritmos

Algoritmo de cadastro

Algoritmo de edição de dados cadastrais

Algoritmo de login

Algoritmo de adição de dados (lançamentos)

Algoritmo de edição de dados (lançamentos)

Algoritmo de soma dos valores de todos os lançamentos

O usuário vai adicionar seus lançamentos. Com esse algoritmo, a soma dos valores dos lançamentos serão feitas, conforme o intervalo ou a categoria desejados.

Os outros algoritmos são simples, logo não precisamos colocar algum tipo de descrição adicional.

3.3 - Projeto de Interface com o Usuário

Usamos a biblioteca gráfica JavaFX para criar a interface com o usuário (as imagens abaixo são ilustrativas).



A screenshot of a JavaFX Login dialog box. The title bar at the top left says "▼ Gaste Menos". The main title "Login" is centered. Below it are two input fields: "Email/CPF" and "Senha". At the bottom are two buttons: "OK" and "Registrar-se".



A screenshot of a JavaFX Cadastro (Registration) dialog box. The title bar at the top left says "▼ Gaste Menos". The main title "Cadastro" is centered. Below it are five input fields: "Nome", "CPF", "Data de Nascimento", "Email", and "Senha". At the bottom are two buttons: "OK" and "Cancelar".



A screenshot of a software window titled "Adicionar lançamento" with a standard Mac OS window header showing a dropdown menu labeled "Gaste Menos". The window contains four text input fields stacked vertically, each with a label to its left: "Nome do lançamento", "Valor do lançamento", "Data do lançamento", and "Categoria". At the bottom right of the window are two buttons: "Salvar" and "Cancelar".

Um exemplo da interface com o usuário ilustrando algumas funcionalidades do sistema.

3.4 - Projeto de Banco de dados

O SGBD utilizado no sistema foi o MySQL, usando o gerenciador gráfico MySQL Workbench.

3.5 - Processo de Desenvolvimento

O processo adotado se baseia no modelo cascata com realimentação. Tendo em vista o tamanho do projeto, o processo cascata se apresenta como o mais adequado. Juntamente com a realimentação, é possível identificar e evitar a propagação de erros, tendo a possibilidade de ao final do processo, voltar a qualquer fase do mesmo.

O processo de entrega é efetuado uma única vez, entregando o software completo após a conclusão do mesmo.

4 - Tecnologia

A linguagem de programação utilizada no projeto foi, exclusivamente, Java 8.

5 - Evolução do Sistema

O Gaste Menos está apto a receber evoluções como adições de novas funcionalidades como somatório de lançamentos, listagem de lançamentos de acordo com um determinado parâmetro; e expansão para a plataforma Web e Mobile.

6 - Link do repositório no GitHub

<https://github.com/lucasferreiraa/gastemenos>