


Documento de Análise de Requisitos	 UFOP
Projeto Money	

Controle de Versões			
Versão	Data	Autor	Notas da Revisão
1.0	09/11/15	Dayana C. Oliveira	Primeira Versão
1.2	14/11/15	Marina Rocha Maia	Segunda Versão
1.3	18/11/15	Dayana C. Oliveira	Adição dos Cronogramas
1.4	19/11/15	Paulo Henrique F. e Silva	Adição dos Casos de Uso
2.0	20/11/15	Dayana C. Oliveira	Versão para entrega
3.0	16/03/16	Dayana C. Oliveira	Versão Final

Introdução

Quando entramos para uma Universidade e mudamos de cidade para morar longe dos pais a liberdade financeira se mostra ao mesmo tempo um sonho e um pesadelo.

Acostumados a não se preocupar com o preço de nada principalmente de coisas essenciais como aluguel, energia elétrica e água nos vemos perdidos no emaranhado de contas mensais.

O sistema Money vem auxiliar os estudantes a resolver este problema ajudando-os no controle de gastos e no gerenciamento dos rendimentos que estes têm durante sua permanência na Universidade. É uma forma simples e intuitiva de gerenciar seu dinheiro através de tabelas e gráficos, permitindo que análises de como seu dinheiro está sendo gasto sejam feitas.

Descrição do Problema

Quando um estudante entra em uma universidade e não mora mais com os pais, ele se depara com o problema de não saber como gerenciar os seus possíveis gastos de acordo com o que ele tem de rendimento. Esses gastos podem ser essenciais como, por exemplo, água, energia elétrica, internet, aluguel ou outros gastos mais supérfluos como festas e roupas.


Objetivos

Objetivo Geral

- Construir um software que auxilie aos universitários a realizar seu controle de gastos.

Objetivos Específicos

- Criar um software em PHP que auxilie os universitários a realizar seu controle financeiro;

Documento de Análise de Requisitos	 UFOP
Projeto Money	

- Permitir que o usuário adicione suas contas referentes a cada mês;
- Criar a interface do software o mais simples e intuitiva possível;
- Mandar e-mail para os usuários lembrando que suas contas estão para vencer;
- Mostrar ao usuário gráficos de seus gastos;
- Permitir que usuário consulte e edite o histórico dos meses anteriores;
- Realizar Link com o Banco de Dados salvando todas as informações dinamicamente.

Escopo da Aplicação

O software Money será uma aplicação em PHP voltada para estudantes universitários. Para atender tal público e fazer com que de fato um grande número de pessoas utilize-o de forma rotineira, o software contará com as seguintes funcionalidades:

- Controles dos gastos;
- Adicionar contas do mês;
- Relatórios com gráficos dos gastos do usuário;
- Avisos por e-mail antes do vencimento de cada conta;
- Consulta do histórico dos meses anteriores com as respectivas contas relacionadas;

Dentre as limitações do projeto estão: não fornecer o serviço de pagamento online de contas, não ter acesso às transações bancárias; e não permitir gerar boletos de contas.

Descrição do Produto

O Money é um website criado para facilitar a vida dos universitários, ele possui uma interface simples e intuitiva que permite ao usuário controlar suas finanças com facilidade. Nele é possível adicionar as contas do mês como aluguel, água, etc.. Pode também inserir gastos imprevistos e saber quanto foi o total de gastos do mês.

É possível ainda consultar o histórico e assim rever todos os meses já fechados por aquele usuário e caso tenha se esquecido de adicionar algum gasto ou rendimento é possível alterar um mês já fechado.

Por fim o Money permite gerar relatórios com gráficos referentes aos gastos do usuário permitindo uma imagem real de como e com o que seu dinheiro é gasto.


Casos de Uso

Cenário A: Logar no sistema

1. Usuário acessa sistema Money;
2. O sistema apresenta a tela de login;
3. Usuário insere o login e sua senha;
4. Usuário aperta o botão de Entrar;
5. O sistema valida o login.

Cenário alternativo: Usuário não tem cadastro ainda, volta ao passo 2

2. Usuário insere login, nome, e-mail e senha no campo de cadastro;
3. Usuário aperta o botão de Cadastrar;

Documento de Análise de Requisitos	 UFOP
Projeto Money	

4. O sistema cadastra o novo usuário.

Cenário B: Adicionar contas ao mês

1. Usuário acessa a página principal do sistema Money;
2. Usuário acessa a página “Adicionar Conta”;
3. Usuário insere os valores e o nome da conta e seleciona o mês;
4. Usuário clica no botão Salvar;

Cenário C: Fechar mês

1. Usuário acessa a página principal do sistema Money;
2. Usuário acessa a página “Fechar Mês”;
3. O sistema mostra ao usuário a tabela com todos os meses adicionados;
4. Usuário seleciona o mês desejado;
5. O valor referente aos gastos do mês selecionado é atualizado;
6. O sistema altera no banco de dados com os novos valores referentes ao mês.

Cenário D: Verificar histórico de mês

1. Usuário acessa a página “Histórico” do sistema Money;
2. O sistema apresenta ao usuário uma lista contendo todos os meses adicionados com seus respectivos gastos
3. Usuário seleciona um mês.
4. O sistema apresenta todas as contas relacionadas ao mês, bem como seu valor e sua data de vencimento.

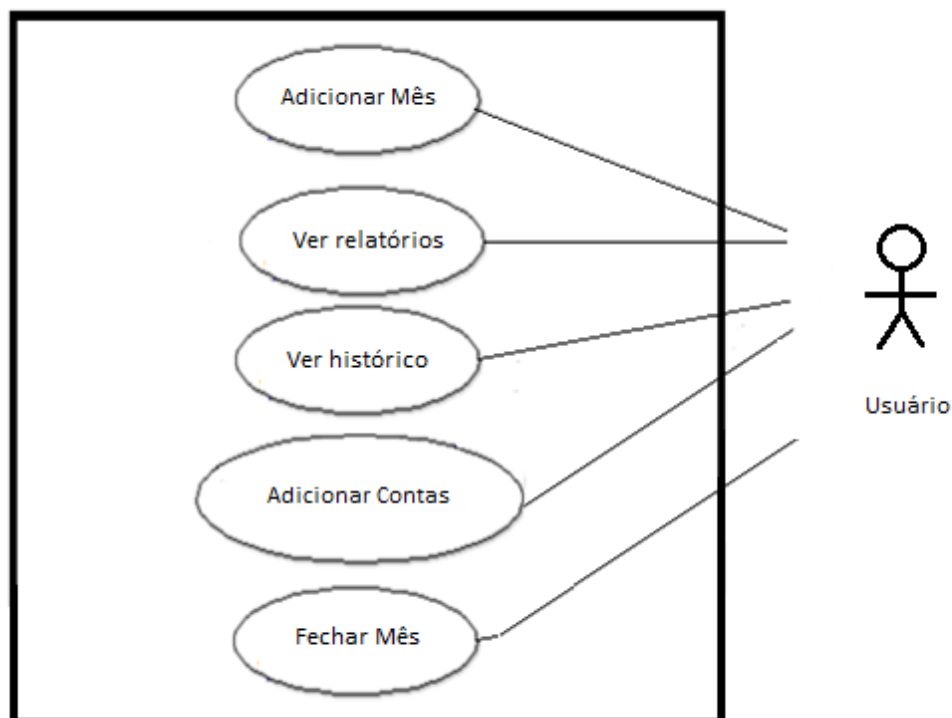

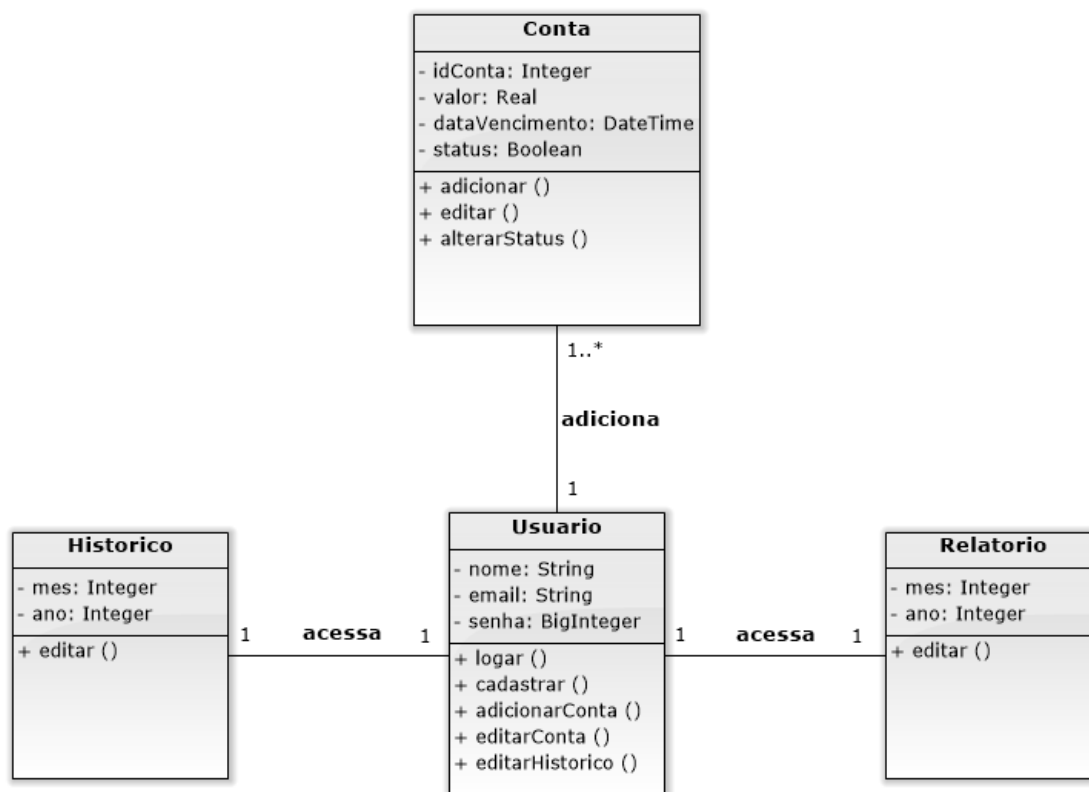


Figura 1 - Diagrama de caso de uso

Documento de Análise de Requisitos	 UFOP
Projeto Money	

Classes



A classe **Conta** possui os atributos `idConta` que é um inteiro, `valor` que é real, `dataVencimento` que é `dateTime` e o `status` que é booleano. As suas operações `adicionar`, `editar` e `alterarStatus`. A classe **Conta** se relaciona com a classe **Usuário**, 1 (um) usuário adiciona n (várias) contas.

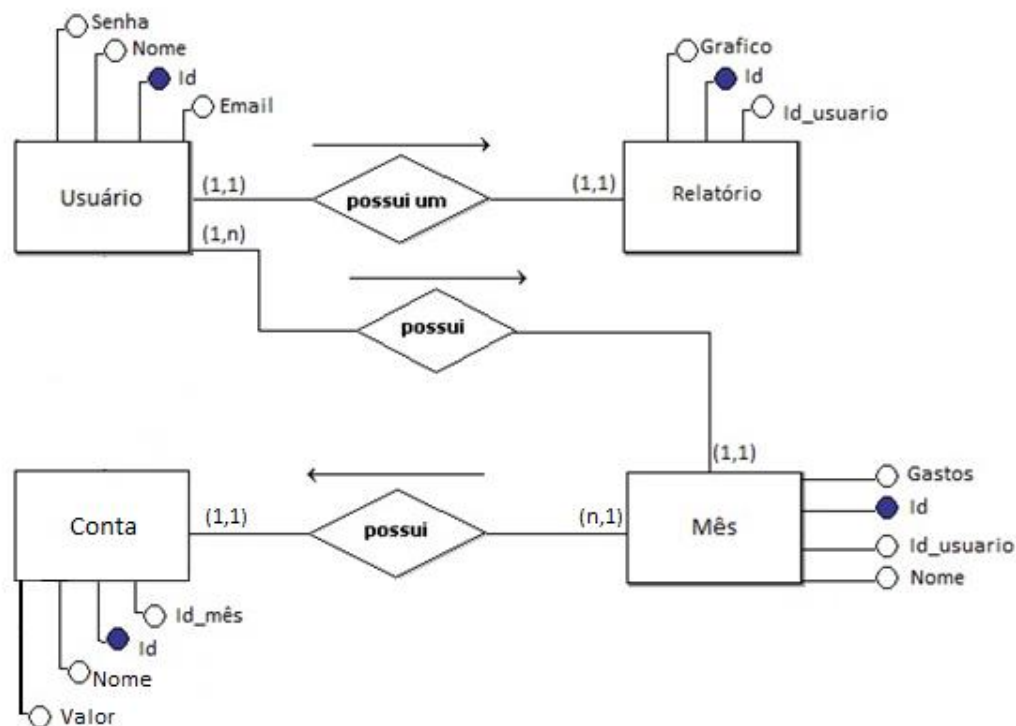
A classe **Usuário** possui os atributos `nome` que é uma `String`, `email` que também é uma `String` e `senha` que é um `BigInteger`. Suas operações são `login`, `cadastrar`, `adicionarConta`, `editarConta` e `editarHistorico`. A classe **Usuário** se relaciona com a classe **Relatório**, 1 (um) usuário acessa 1 (um) relatório, com a classe **Histórico**, 1(um) usuário acessa 1(um) histórico e com **Conta**, 1 (um) usuário acessa n (várias) contas.

A classe **Histórico** possui os atributos `mês` que é um inteiro e `ano` que também é um inteiro. Sua operação é `editar`. A classe **Histórico** se relaciona com a classe **Usuário**, 1 (um) usuário acessa 1 (um) histórico.

A classe **Relatório** possui os atributos `mês` que é um inteiro e `ano` que também é um inteiro. Sua operação é `editar`. A classe **Relatório** se relaciona com a classe **Usuário**, 1 (um) usuário acessa 1 (um) relatório.





Banco de Dados



Cronograma

Tarefa	Mês			
	12	1	2	3
Documento de Requisitos	X			
Desenvolvimento das Interfaces			X	X
Implementação do Software em PHP			X	X
Feedback do Cliente				X
Possíveis Reparos				X

Documento de Análise de Requisitos	 
Projeto Money	

Funções não implementadas

Infelizmente algumas funções não foram implementadas devido ao pouco conhecimento referente às mesmas por parte da equipe do projeto.

As funções não implementadas foram:

- Relatórios com gráficos dos gastos do usuário;
- Avisos por e-mail antes do vencimento de cada conta;

Por se tratar de um projeto desenvolvido para uma disciplina acadêmica, funções que necessitam de conhecimento superior ao conteúdo exposto na disciplina, tornaram-se difíceis e mesmo com tentativas da equipe de implementá-las, não foi possível incluir as mesmas neste projeto.