# Introdução

# 1. Objetivo:

Observar como a importância do controle de finanças pessoais é importante, pois devidamente utilizado o usuário tem um controle fiel das suas finanças, assim podem se planejar melhor para o futuro, como por exemplo para fazer as prospecções o sistema acaba por administrar suas finanças auxiliando assim a comprar o produto que ele deseja.

#### 2. Publico:

Público alvo é para que os usuários que querem ter controle sobre sua condição financeira.

### 3. Uso Pretendido:

Uso para obter um controle de entrada e saídas de fluxo de caixa, também oferecendo um simulado de uma compra futura para saber se é possível comprar esse produto, o intuito é oferecer um software que atenda às necessidades do usuário de maneira fácil e de fácil adaptação, levando em consideração que a dificuldade de encontrar tais sistemas gratuitos e simplicidade no uso.

# Requisitos do Sistema

# **Requisitos Funcionais:**

# 1. Fazer Login

# a. Descrição:

i. Permitir usuário fazer login no sistema ou a inserção de um novo usuário caso não esteja cadastrado.

## b. Fluxo principal:

- i. Usuário informa dados: nome de usuário e senha.
- ii. Sistema verifica usuário se é válido ou não.
- iii. Sistema aceita e faz o login.
- iv. Sistema direciona para interface do sistema com usuário

#### c. Fluxo alternativo:

- i. Sistema alerta "Usuário incorreto".
- ii. Usuário informa que deseja se cadastrar

- iii. Sistema verifica que a senha é robusta
- iv. Sistema faz cadastro do usuário
- v. Sistema faz login na conta do usuário
- vi. Sistema informa "Usuário cadastrado com sucesso! Para receber avisos no seu correio eletrônico preferido, por favor informe seu e-mail."
- vii. Usuário informa e-mail
- viii. Sistema verifica e-mail
  - ix. Sistema informa "Cadastro completo:)"

### d. Fluxo de exceção: Senha muito fraca:

- i. Sistema informa "Senha muito fraca, use no mínimo 8 caracteres contendo no mínimo: 1 letra minúscula, 1 letra maiúscula e 1 caractere especial.".
- ii. Retorna ao início.

#### e. Fluxo alternativo: Desistir do cadastro de usuário

- i. Encerrar cadastro de usuário
- ii. Voltar a tela inicial de login

### 2. Lançar Contas

# a. Descrição:

i. Permitir usuário lançar contas a pagar.

### b. Fluxo principal:

- i. Usuário faz um pagamento e marca a data da conta.
- ii. Sistema verifica se é uma conta válida.
- iii. É agendado e lançado no sistema.
- iv. Sistema informa "Conta lançada com sucesso".

#### c. Fluxo de exceção: Conta invalida:

- i. Sistema informa "Conta a pagar não é válida, verifique e tente novamente".
- ii. Retorna ao início.

# d. Fluxo alternativo: Desistir do Lançamento:

- i. Encerrar o lance da conta
- ii. Voltar a tela inicial do usuário.

#### 3. Lançar Gastos

#### a. Descrição:

i. Permitir usuário lançar gasto.

### b. Fluxo principal:

i. Usuário registra um pagamento.

- ii. Sistema verifica se é um pagamento válido.
- iii. Sistema informa "Pagamento feito com sucesso".

# c. Fluxo de exceção: Pagamento invalida:

- i. Sistema informa "Pagamento negado, tente novamente".
- ii. Retorna ao início.

# d. Fluxo alternativo: Desistir de registrar Pagamento:

- i. Encerrar o registro do pagamento
- ii. Voltar a tela inicial do usuário.

### 4. Lançar Renda

### a. Descrição:

i. Permitir usuário lançar renda.

### b. Fluxo principal:

- i. Usuário registra renda.
- ii. Renda é lançada no sistema.
- iii. Sistema informa "Registro de renda foi um sucesso".

# C. Fluxo alternativo: Desistir do Registro:

- i. Encerrar o registro de renda.
- ii. Voltar a tela inicial do usuário.

# 5. Fazer Prospecção

# a. Descrição:

i. Pegar todas informações inseridas e calcular o resultado que dará final do mês com detalhes.

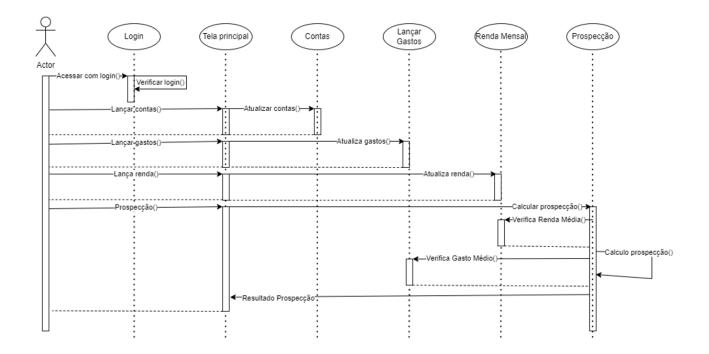
### b. Fluxo principal:

- i. Usuário faz o pedido(click) de prospecção.
- ii. Sistema pega todas informações.
- iii. Sistema abre gráfico detalhado.

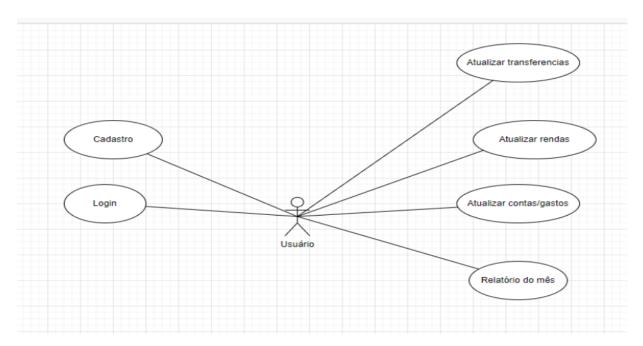
# c. Fluxo alternativo: Erro por falta de informações

- i. Sistema não inicia após o pedido do usuário.
- ii. Sistema informa mensagem de erro ao detalhar gráfico com a mensagem "Falta de informações mínimas".

# Diagrama de Sequência:



### Caso de uso:



# Diagrama de Classe:

#### usuario

- idUsuario : int
- nome : String
- senha: String
- confirmaSenha: String
- + CriarUsuario(): void
- + ValidarUsuario(): void

#### rendaMensal

- idRendaMensal: int
- valor : float
- data : date
- descricao : String
- + inserirValor(): void
- + inserirData(): void
- + inserirDescricao(): void
- + obterSaldo(): float
- + mostrarRelatorios(): void

#### prospeccao

- idProspeccao: int
- descricao : String
- valor : float
- parcelas : int
- + descricaoSimulado(): void
- + inserirValor(): void
- + inserirParcelas(): void
- + executarProspeccao(): void

#### conta

- idConta : int - nomeConta0 : String
- + inserirConta(): void
- + deletarConta(): void
- + AlterarConta(): void

#### **lacamentosGastos**

- idLancamentosGastos : int
- valorGasto : float
- dataGasto : date
- quantidade/consumo : String
- + inserirValor(): void
- + alterarValor(): void
- + deletarValor(): void
- + inserirData(): void
- + alterarData(): void
- + deletarData(): void
- + inserirParcelas(): void
- + obterSaldoTotal() : float
- + inserirQuantConsumo(): void
- + alterarQuantConsumo(): void
- + excluirQuantConsumo(): void
- + mostrarRelatorios(): void