

Universidade Federal de Alagoas
Curso de Ciência da Computação

Projeto de Engenharia de Software 1 (1ª Etapa)
**Prego - Sistema para gerenciamento
de dívidas e devedores**

Antônio Andrade, Jefferson Farias, Erick Oliveira

Arapiraca, 12/setembro

Índice

1. Introdução.....	2
2. Requisitos.....	2
2.1. Elicitação dos requisitos.....	2
2.2. Requisitos funcionais.....	5
2.3. Requisitos não funcionais.....	7
2.4. Validação dos requisitos.....	8
3. Projeto.....	8
3.1. Projeto orientado a objetos.....	8
3.2. Projeto de interface com o usuário.....	12
3.3. Projeto de banco de dados.....	17
4. Planejamento de entregas.....	19

1. Introdução

Seu João possui uma loja de variedades, onde faz vendas a prazo confiando apenas na palavra de seus clientes (fiado). Seu João faz o controle das pessoas que o devem através de anotações feitas a mão em seu caderno. Em determinado momento, seu João se viu em meio a um monte de papéis desorganizados, e começou a perceber que estava perdendo o controle de seu negócio.

O sistema Prego foi criado com o objetivo de organizar, armazenar e catalogar devedores e dívidas, onde o usuário faz o cadastro de um devedor inserindo nome (obrigatório), endereço (não obrigatório) e telefone (não obrigatório), logo após é possível associar inúmeras dívidas a esse devedor, essas dívidas possuem um valor (obrigatório), descrição (não obrigatório) e data de fechamento (não obrigatório). Cada dívida cadastrada possui algum desses status: Aberta e no prazo (o devedor ainda não pagou mas a dívida ainda não venceu), Aberta e em atraso (a data de fechamento já passou e o devedor ainda não pagou), Aberta (o usuário não especificou uma data de fechamento no ato do cadastro da dívida), Paga (o devedor já fez o pagamento). Quando um devedor faz o pagamento, o usuário altera a dívida no sistema para paga.

Cada usuário está associado a seu devedor (o sistema permite mais de um usuário), é possível listar todos os devedores e ver nome, cidade (se houver), telefone (se houver) e quantas dívidas ele possui, também é possível listar todas as dívidas, onde além do nome, são mostrados valor da dívida, status e especificação (se houver). Tanto na listagem de dívidas quanto na de devedores, são mostrados somente devedores e dívidas associados ao usuário que está utilizando o sistema no momento.

O sistema é ideal para pessoas, empresários ou organizações que trabalham com vendas a prazo e precisam ter o controle de quanto tem a receber e das pessoas que tem a receber.

2. Requisitos

2.1. Elicitação dos requisitos

A primeira medida tomada para a elaboração dos requisitos, foi a observação in loco do negócio durante um período de 10 dias. Através disso foi possível observar particularidades difíceis de serem notadas por outros meios, por exemplo, o fluxo de clientes é moderado no período da manhã, porém a tarde, o fluxo é mais intenso, sendo necessário um atendimento mais ágil.

A segunda medida, pudemos ter acesso ao caderninho que seu João usava para anotar o nome e o valor que seus devedores o deviam, como pode ser visto a seguir:

D. TOTAL

05-05 - RESID.	715.00
" - " - " - "	180.00
12-05 - " - "	155.00
19-05 - " - "	190.00
26-05 - " - "	170.00
	<u>1410.00</u>

02-06 - R.	610.00	
01-11 - " - "	210.00	7
09-06 - " - "	130.00	7
16-06 - " - "	120.00	MARCA
23-06 - " - "	116.00	24-07-2000
23-06 - " - "	159.00	01-08-349.00
07-07 - " - "	177.00	" - " 230.00
	<u>1582.00</u>	11-08-125.00

14-07 - R.	782.00	15-08-89.00
" - " - " - "	148.00	25-08-145.00
21-07 - " - "	160.00	21-08-187.00
28-07 - " - "	160.00	<u>1321.00</u>
	<u>1249.00</u>	
	<u>109.00</u>	
	<u>1358.00</u>	

BE2 -

05-05- 35,00

11-11 35,00

13-05- 33,00

203,00

~~16-06- 35,00~~

~~23-06- 34,00~~

~~29-06- 35,00~~

~~06-07- 35,00~~

23,00

18-05- 35,00

26-05- 35,00

02-06- 35,00

09-06- 35,00

741,00

14-07- 32,00

21-07- 35,00

28-07- 33,00

04-08- 35,00

135,00

10-08- 46,00

181,00

11-11 30,00

18-08- 33,00

25-08- 40,00

01-09- 42,00

11-11 23,00

09-09- 35,00

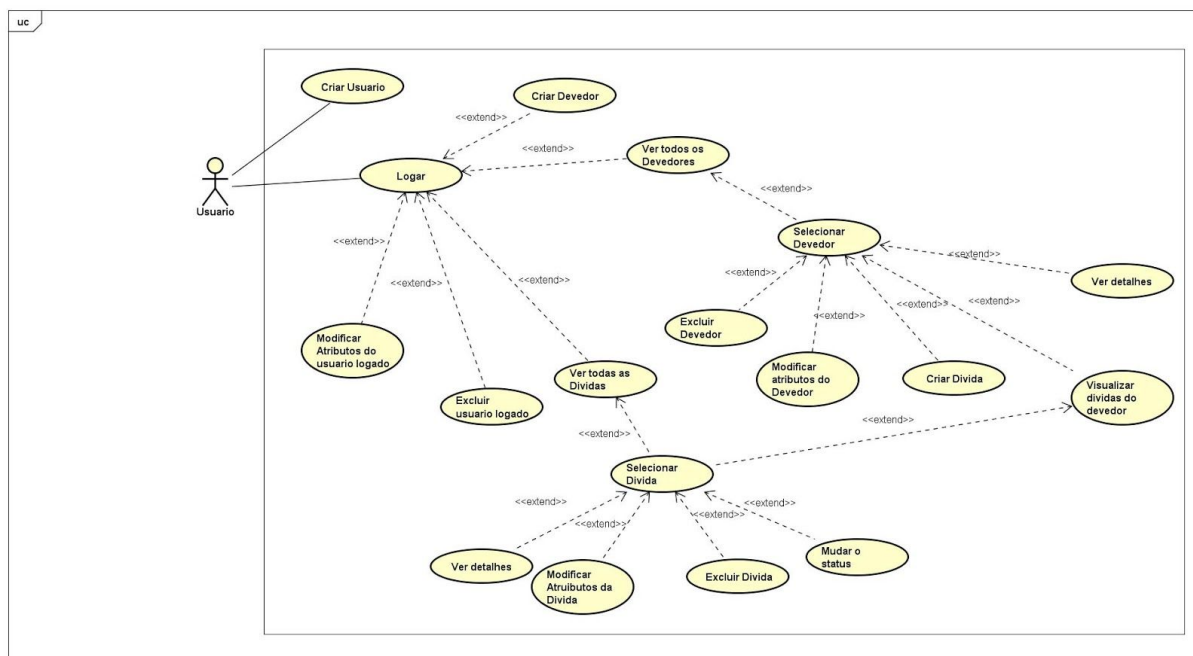
203,00

A terceira medida foram entrevistas com todos os potenciais usuários do sistema. O negócio do seu João é gerenciado por ele e sua família, através das entrevistas foi possível obter diferentes pontos de vista em relação aos recursos que desejavam no sistema. Um dos pontos importantes descobertos nessa etapa, é que seu João não é muito familiarizado com tecnologia.

Por fim, os pontos de vista descobertos durante a entrevista foram agrupados segundo uma hierarquia, que vai dos que foram consenso entre os entrevistados e foram descritos como mais essenciais, até os classificados como menos importantes.

2.2. Requisitos funcionais

Requisitos Funcionais	
1.	Criar usuário
2.	Cadastrar devedor
3.	Alterar dados de usuário logado
4.	Excluir usuário logado
5.	Ver lista de devedores associados ao usuário logado
6.	Excluir devedor
7.	Modificar atributos de um devedor
8.	Criar dívida
9.	Listar todas as dívidas
10.	Modificar atributos de uma dívida
11.	Excluir uma dívida



Caso de uso:	Modificar atributos do usuário logado
Ator(es):	Usuário
Pré-condições:	Estar logado como um usuário do sistema
Fluxo padrão:	<ol style="list-style-type: none"> 1. Usuário clica no botão “atualizar usuário” apresentado na tela principal. 2. Tela principal é substituída pela tela “Atualizar dados de usuário”. 3. Usuário modifica os dados que desejar e seleciona a opção “salvar”. 4. Uma tela de confirmação aparece e o usuário seleciona a opção “Sim” para confirmar a alteração dos dados.
Fluxo de exceção 1:	Ao modificar o nome de usuário, o campo é deixado em branco. Nesse caso uma mensagem “Campo nome de usuário não pode ser vazio” é mostrada.
Fluxo de exceção 2:	Ao tentar salvar as alterações, o sistema apresenta um erro de comunicação com o banco de dados. Nesse caso a mensagem “Erro ao salvar as alterações, por favor tente novamente!” é mostrada na tela.
Pós-condições:	<p>Ao confirmar a alteração dos dados a mensagem “Dados de usuário atualizados com sucesso” é mostrada.</p> <p>Tela principal é apresentada com os novos dados de usuário.</p>
Prioridade:	Alta

Caso de uso:	Ver todos os devedores
Ator(es):	Usuário
Pré-condições:	Estar logado como um usuário do sistema
Fluxo padrão:	<ol style="list-style-type: none"> 1. Usuário clica no botão “Listar devedores” apresentado na tela principal. 2. Na área de trabalho da tela principal a lista de devedores associados aquele usuário é mostrada, a lista contém o nome do devedor, cidade, telefone e número de dívidas que o mesmo possui.
Fluxo de exceção 1:	Ao tentar apresentar as informações, o sistema apresenta um erro de comunicação com o banco de dados. Nesse caso a mensagem “Erro ao mostrar informações, por favor tente novamente!” é mostrada na tela.
Fluxo de exceção 2:	Usuário não possui nenhum devedor cadastrado, nesse caso uma tela vazia é mostrada.
Pós-condições:	Usuário visualiza na tela a lista de devedores associados a ele.
Prioridade:	Alta

2.3. Requisitos não funcionais

Requisitos Não Funcionais	
RNF 1 - A linguagem utilizado nos menus deverá ser de fácil entendimento.	
Verificação: Ao utilizar o sistema pela primeira vez, o usuário deve ser capaz de identificar, intuitivamente, o comportamento de pelo menos 80% das opções apresentadas na tela.	
RNF 2 - O usuário deve ser capaz de acessar todas as funcionalidades do sistema passando por poucos menus.	
Verificação: Devem ser exibidas no máximo 4 (quatro) telas diferentes para se chegar em qualquer objetivo	
RNF 3 - O sistemas deve ter respostas rápidas, já que o fluxo de clientes muitas vezes é intenso.	
Verificação: Ao listar todos os clientes ou dívidas, os resultados devem ser apresentados em, no máximo, 4 segundos. Ao utilizar alguma opção de cadastramento ou alteração, os resultados devem ser apresentados em, no máximo, 2 segundos.	

2.4. Validação dos requisitos

Desde o início ficou claro que se trata de um sistema de grande valia porém simples em seu objetivo. Após colher os requisitos, apresentamos ao cliente todas as funcionalidades que o sistema final viria a ter, de acordo com o que pudemos observar e entender através do processo de análise. Para complementar a validação, criamos alguns protótipos de tela para simular o comportamento do sistema final e apresentamos aos potenciais usuários, após alguns ajustes chegamos em um acordo sobre como o sistema final deveria ser.

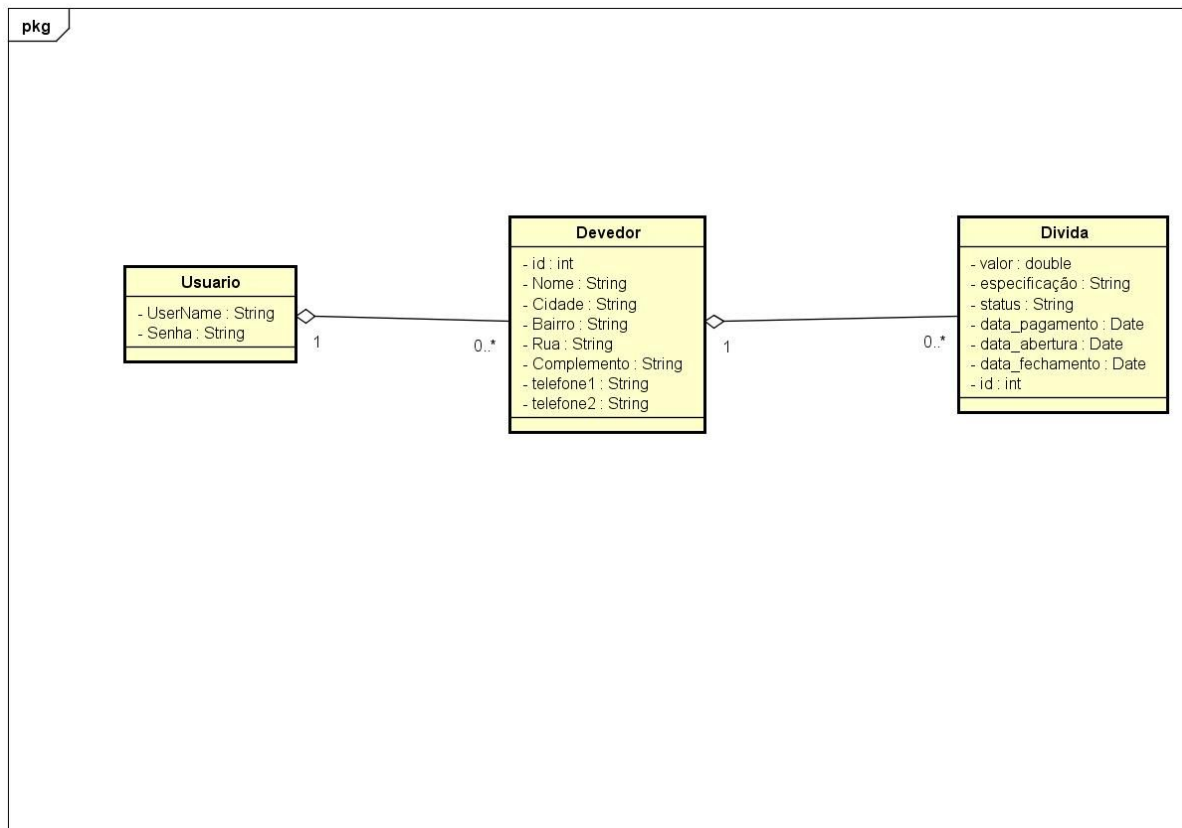
Ficou acordado também que teríamos um prazo de 4 meses para fazer a entrega final, o que está totalmente dentro dos parâmetros, já que o sistema não utiliza nenhuma tecnologia complexa ou de alto custo.

Apesar de simples, o sistema trará diversos benefícios ao negócio, já que o método utilizado anteriormente é muito ultrapassado e passível de vários problemas.

3. Projeto

3.1. Projeto orientado a objetos

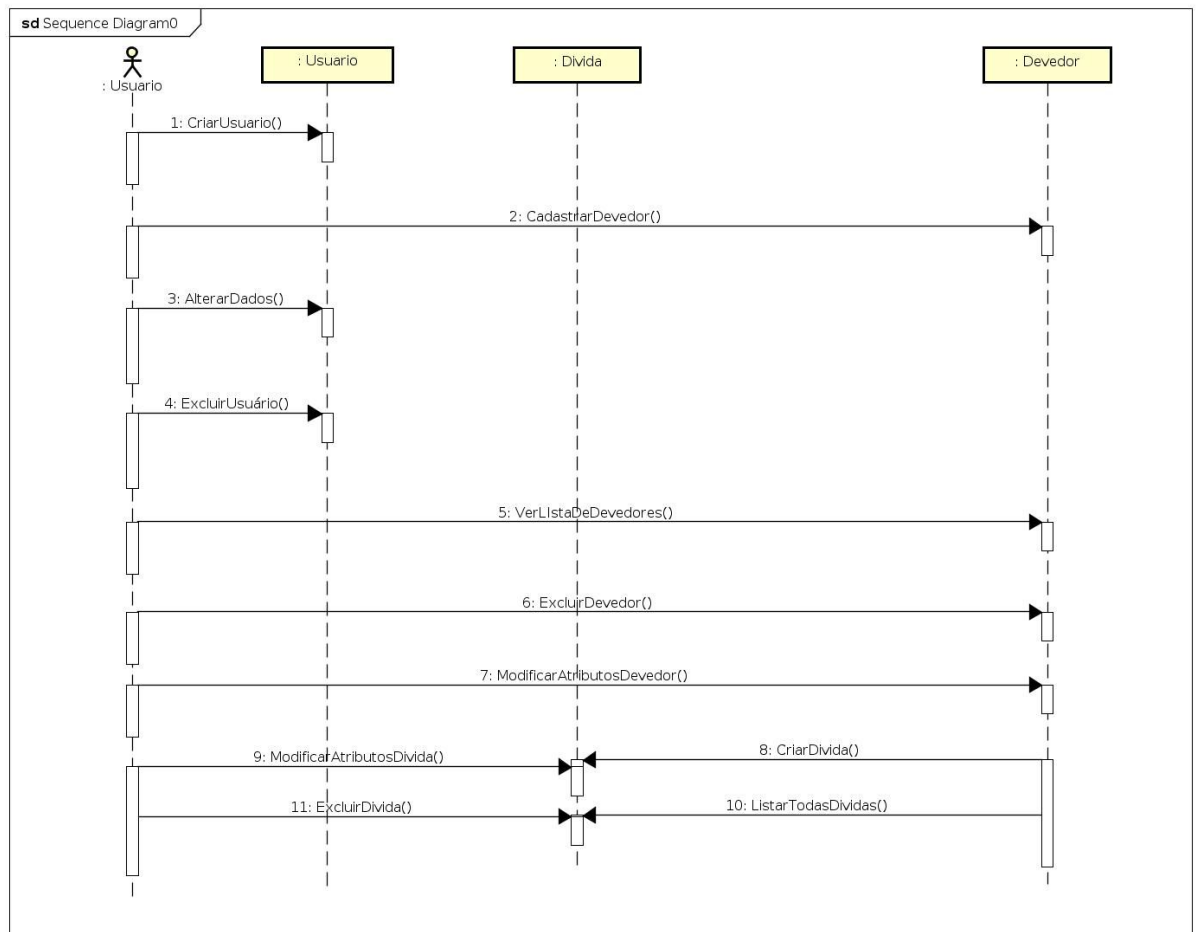
- Diagrama de Classes



O diagrama de classes em questão possui 3 classes, que são suficientes para expressar todas as funcionalidades do sistema, ao mesmo tempo que não comprometem a estrutura do código em relação aos princípios da orientação a objetos. Além disso é de fácil entendimento para o caso de alguma evolução futura.

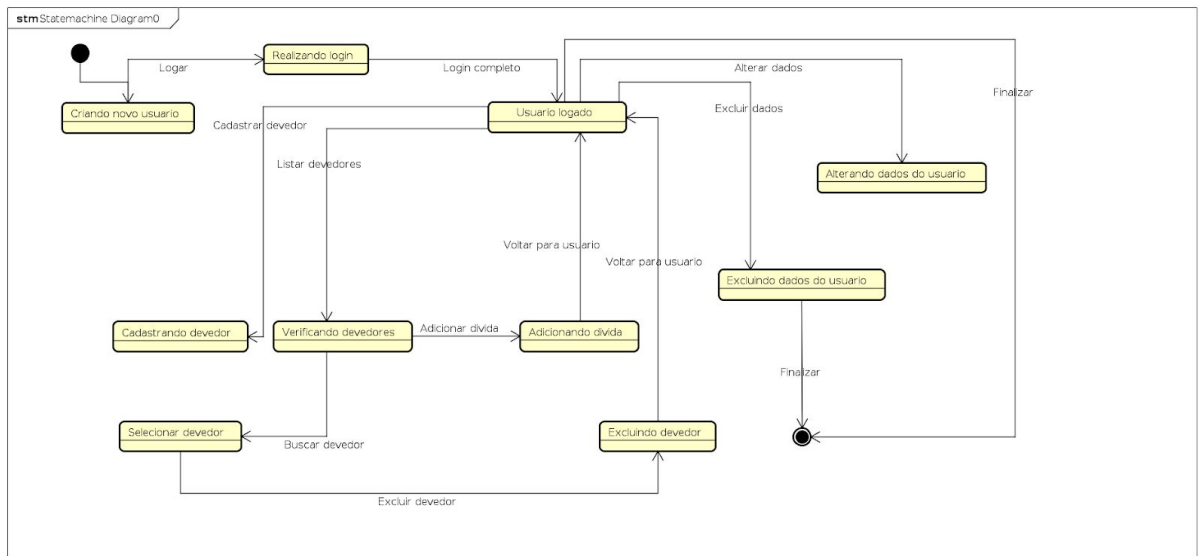
Por se tratar de um negócio pequeno, pelo menos inicialmente, não foram necessários muitos atributos para cada classe. Na classe “Usuario” existem apenas 2, já que o propósito do sistema é ser usado num escopo familiar, onde todos os usuários são conhecidos, não necessitando outros atributos de identificação. A classe “Devedor” possui “id”, que não é acessado pelo usuário, “Nome”, atributos de endereço (Cidade, Bairro, Rua, Complemento) e “telefone”. Por fim, a classe “Dívida” possui “id”, que não é acessado pelo usuário, “valor” da dívida, “especificação”, “status”, “data_pagamento”, “data_abertura” e “data_fechamento”, sendo a classe mais complexa, onde abrangemos todas as possibilidades sobre qual é a situação de uma dívida, armazenando a data que o devedor pagou, a data que a dívida foi criada e a data que o usuário estipulou para vencimento da dívida.

- Diagrama de sequência



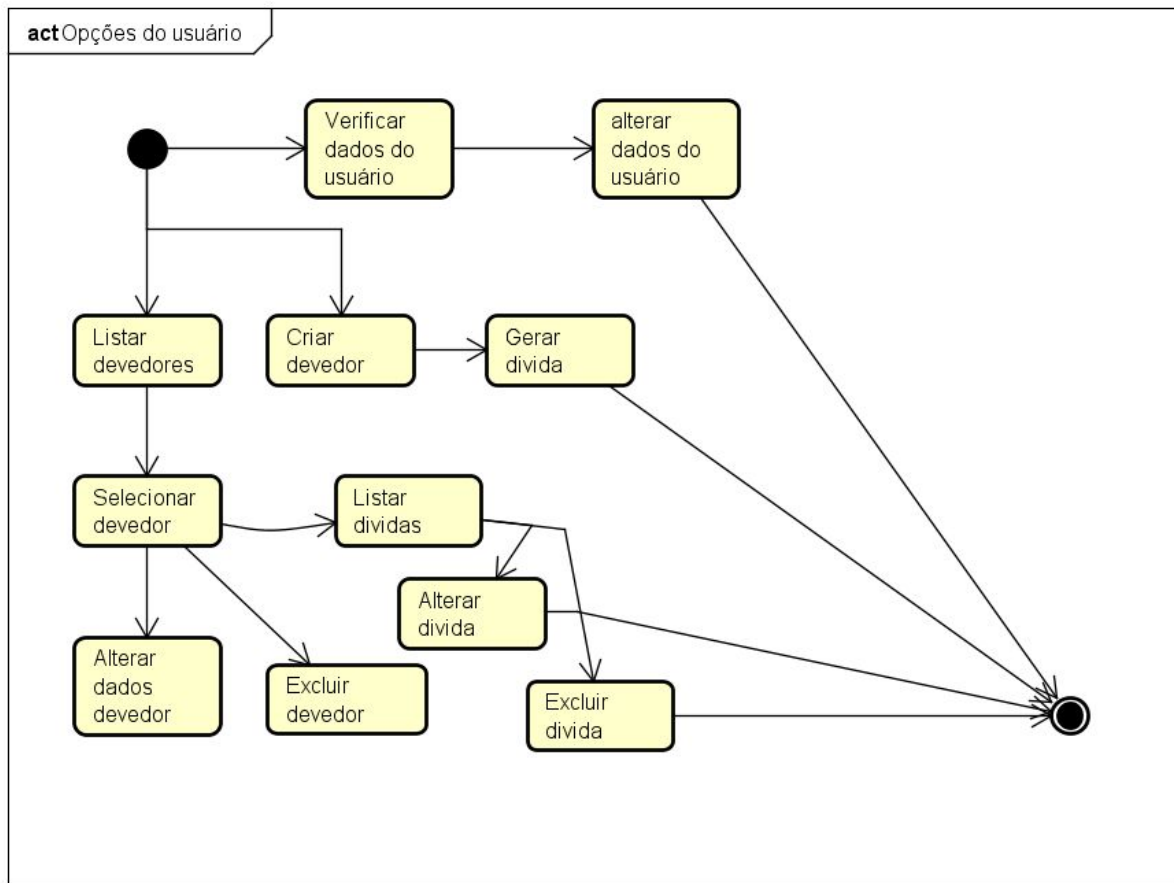
O diagrama de sequência do sistema possui o ator Usuário, e os Models (Classes), Usuario, Devedor e Dívida, onde o usuário dispara mensagens para as classes. A princípio o ator usuário dispara ações para usuário (classe) para que este faça as operações de cadastro, tanto de usuário quanto de dívidas e devedores, seguidos das operações de alterar e listar os dados criados pelo usuário.

- Diagrama de estados



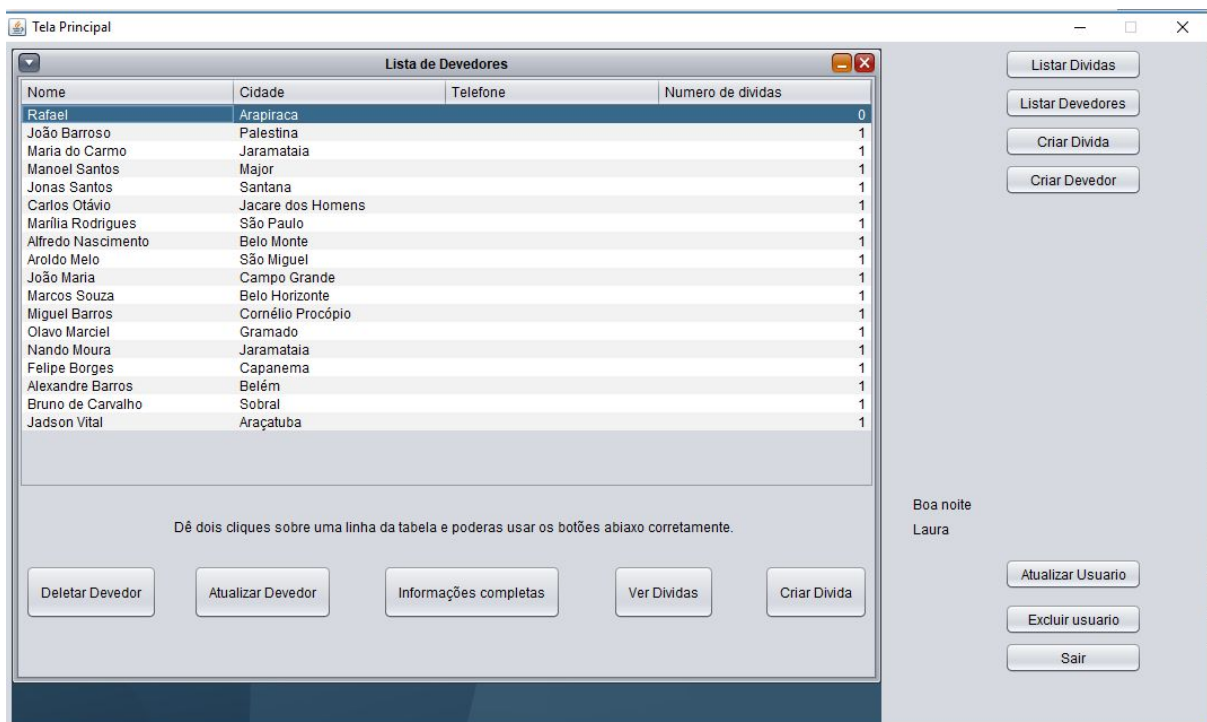
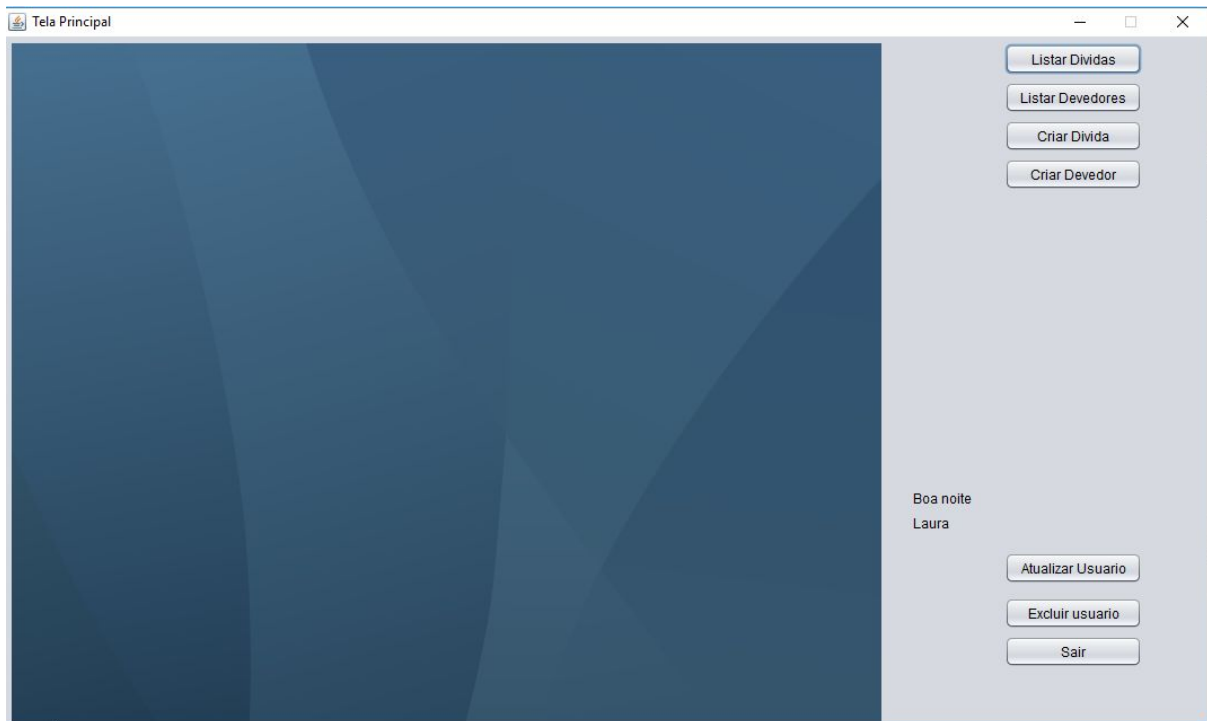
O diagrama de estados acima demonstra de forma prática os fluxos de entre os estados em tempo de execução, onde na primeira parte o usuário tem de se cadastrar para poder fazer um login, passando para o próximo estado que é quando ele estará logado e abrirá então um leque de opções para o próprio escolher, e manipular tanto os seus dados, quanto de seus clientes (no caso os devedores).

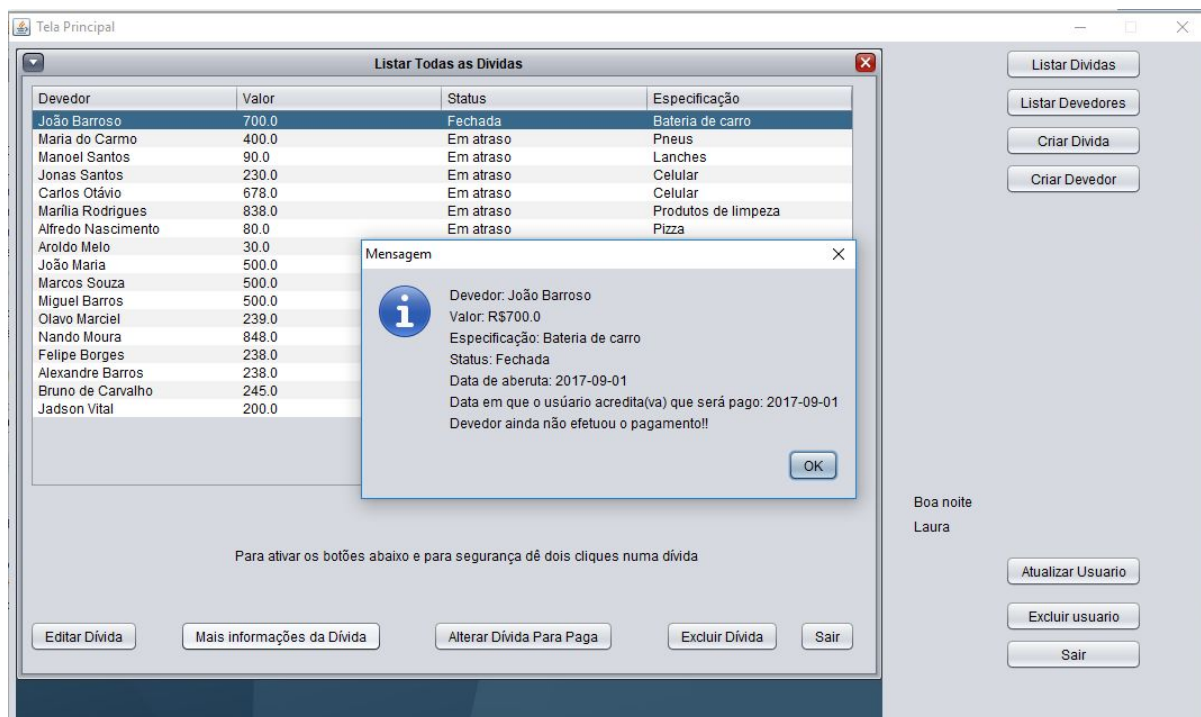
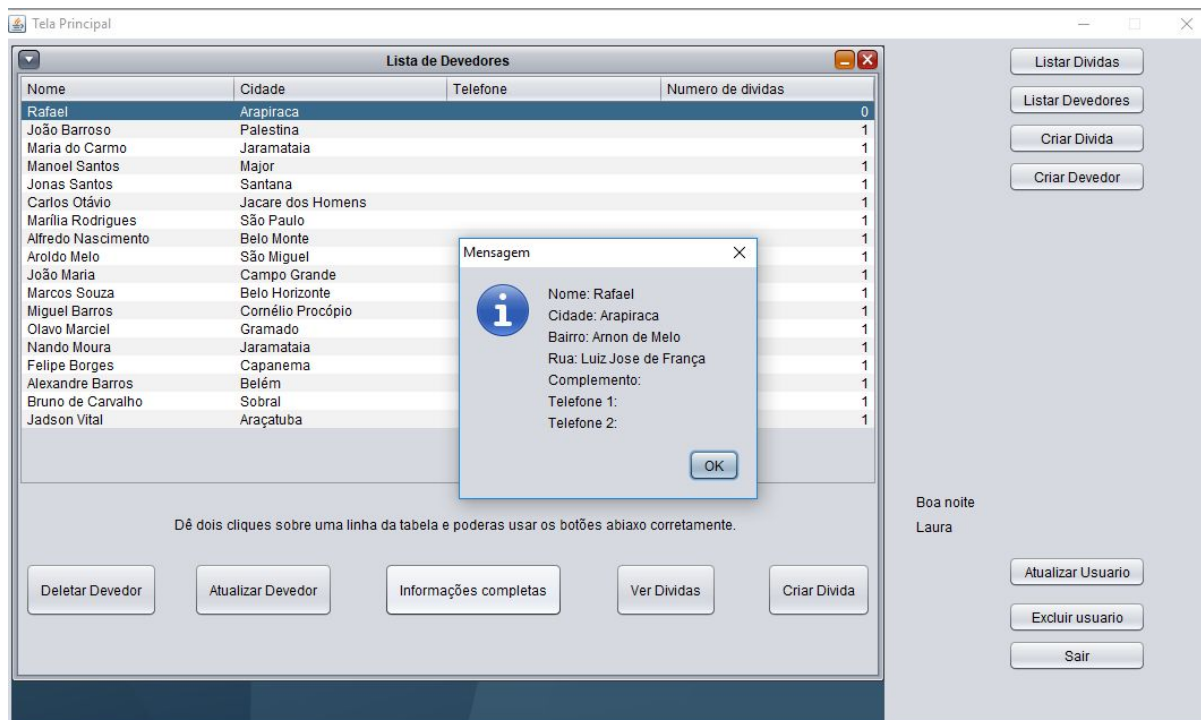
- Diagrama de atividades



O diagrama de atividades feito com a intenção de descrever os passos do usuário após o período de login, onde o mesmo tem várias opções, para criar devedores e dívidas, onde para excluir um devedor por exemplo ele tem de estar na lista de devedores, portanto passando pela operação de listar devedores, e também o usuário poderá alterar seus próprios dados, passando do estado de logado inicial para estados de alteração e criação de usuários e devedores(clientes).

3.2 Projeto de interface com o usuário





Tela Principal

Valor	Status	Especificação
Jonas Santos	230.0	Em atraso

Devedor: Jonas Santos

Para ativar os botões abaixo e para segurança dê dois cliques numa dívida

Editar Dívida

Mais informações da Dívida

Alterar Dívida Para Paga

Excluir Dívida

Sair

Listar Dividas

Listar Devedores

Criar Dívida

Criar Devedor

Boa noite
Laura

Atualizar Usuario

Excluir usuario

Sair

Tela Principal

Para atualizar os dados do devedor selecionado, edite os campos em questão abaixo com os novos dados

*Nome

Alfredo Nascimento

Cidade

Belo Monte

Bairro

Centro

Rua

Rua da Imprensa

Complemento

Telefone1

Telefone2

Atualizar Devedor Selecionado

Observações: As informações com * são essenciais
Devedores não devem ter nomes iguais

Cancelar

Atualizar

Listar Dividas

Listar Devedores

Criar Dívida

Criar Devedor

Boa noite
Laura

Atualizar Usuario

Excluir usuario

Sair

Tela Principal

Cadastrar dívida para um devedor

*Valor da Dívida:

Especificação:

Data de fechamento:

//

Devedor:

Miguel Barros
1

itens com * são obrigatorios

Cancelar

Salvar

Listar Dividas

Listar Devedores

Criar Divida

Criar Devedor

Boa noite
Laura

Atualizar Usuario

Excluir usuario

Sair

Tela Principal

Cadastrar Devedor

Nome do Devedor*

Cidade

Bairro

Rua

Complemento

Telefone1

Telefone2

Observações: As informações com * são essenciais
Devedores não devem ter nomes iguais

Cadastrar

Listar Dividas

Listar Devedores

Criar Divida

Criar Devedor

Boa noite
Laura

Atualizar Usuario

Excluir usuario

Sair

Atualizar dados de Usuario

Usuário:

Nova Senha:

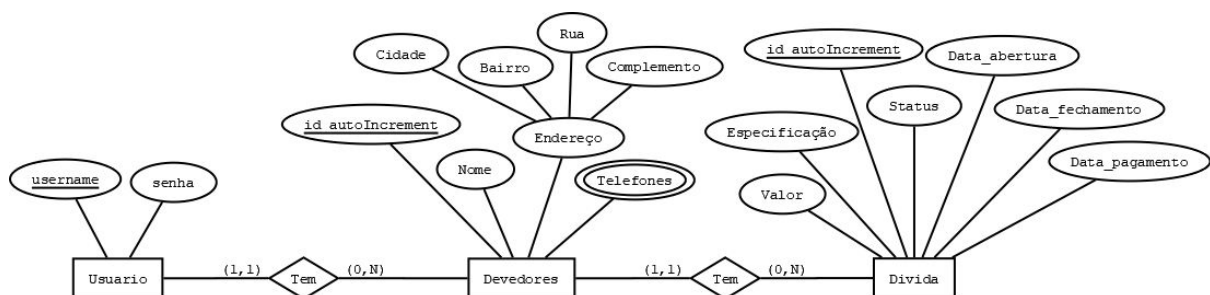
Novo Usuario

*Digite seu nome de usuário:

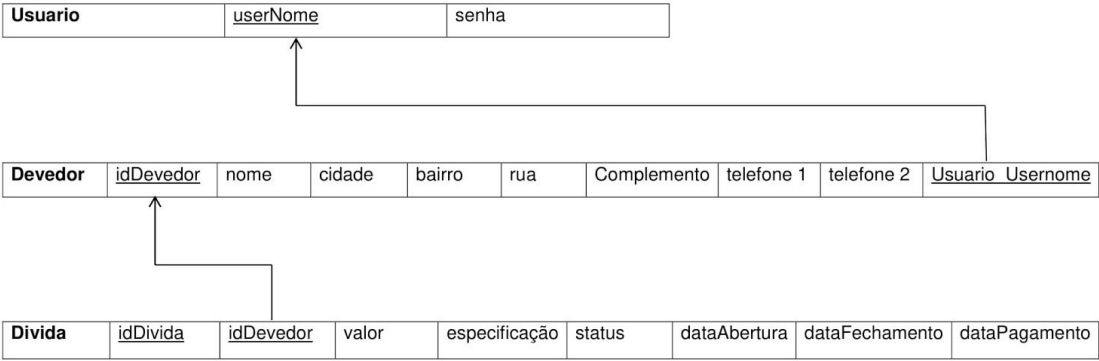
Senha:

3.3 Projeto de Banco de Dados

- Modelo Conceitual



- Modelo Relacional

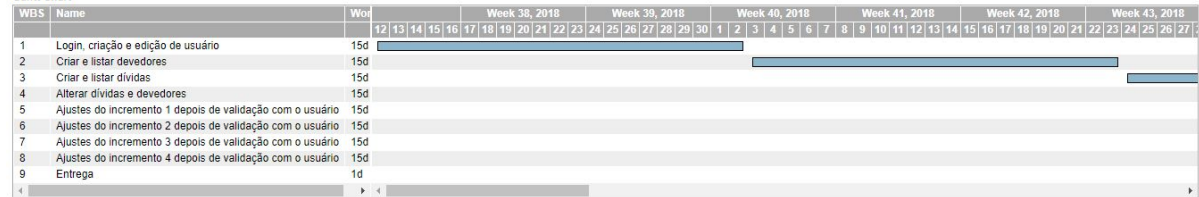


4. Planejamento de entregas

Prego - Sistema

Start: setembro 12, 2018
Finish: março 5, 2019
Report Date: setembro 12, 2018

Gantt Chart



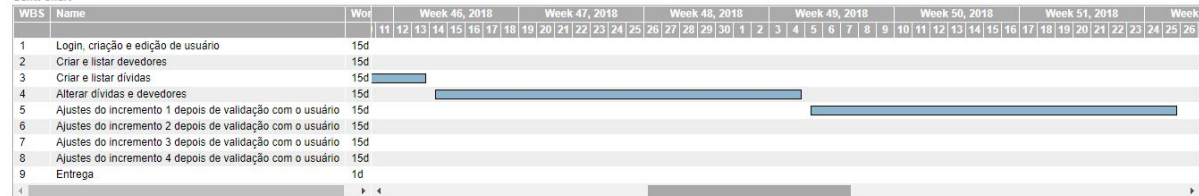
Tasks

WBS	Name	Start	Finish	Work	Priority	Complete	Cost	Notes
1	Login, criação e edição de usuário	Sep 12	Oct 2	15d	3	0%		
2	Criar e listar devedores	Oct 3	Oct 23	15d	3	0%		
3	Criar e listar dívidas	Oct 24	Nov 13	15d	3	0%		
4	Alterar dívidas e devedores	Nov 14	Dec 4	15d	2	0%		
5	Ajustes do incremento 1 depois de validação com o usuário	Dec 5	Dec 25	15d	3	0%		
6	Ajustes do incremento 2 depois de validação com o usuário	Jan 2	Jan 22	15d	3	0%		
7	Ajustes do incremento 3 depois de validação com o usuário	Jan 23	Feb 12	15d	3	0%		
8	Ajustes do incremento 4 depois de validação com o usuário	Feb 13	Mar 5	15d	2	0%		
9	Entrega	Mar 5	Mar 5	1d	3	0%		

Prego - Sistema

Start: setembro 12, 2018
Finish: março 5, 2019
Report Date: setembro 12, 2018

Gantt Chart



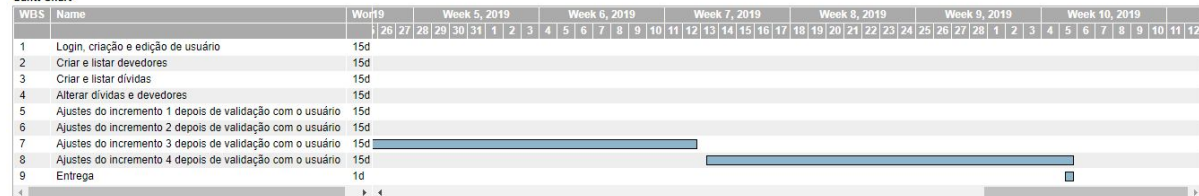
Tasks

WBS	Name	Start	Finish	Work	Priority	Complete	Cost	Notes
1	Login, criação e edição de usuário	Sep 12	Oct 2	15d	3	0%		
2	Criar e listar devedores	Oct 3	Oct 23	15d	3	0%		
3	Criar e listar dívidas	Oct 24	Nov 13	15d	3	0%		
4	Alterar dívidas e devedores	Nov 14	Dec 4	15d	2	0%		
5	Ajustes do incremento 1 depois de validação com o usuário	Dec 5	Dec 25	15d	3	0%		
6	Ajustes do incremento 2 depois de validação com o usuário	Jan 2	Jan 22	15d	3	0%		
7	Ajustes do incremento 3 depois de validação com o usuário	Jan 23	Feb 12	15d	3	0%		
8	Ajustes do incremento 4 depois de validação com o usuário	Feb 13	Mar 5	15d	2	0%		
9	Entrega	Mar 5	Mar 5	1d	3	0%		

Prego - Sistema

Start: setembro 12, 2018
Finish: março 5, 2019
Report Date: setembro 12, 2018

Gantt Chart



Tasks

WBS	Name	Start	Finish	Work	Priority	Complete	Cost	Notes
1	Login, criação e edição de usuário	Sep 12	Oct 2	15d	3	0%		
2	Criar e listar devedores	Oct 3	Oct 23	15d	3	0%		
3	Criar e listar dívidas	Oct 24	Nov 13	15d	3	0%		
4	Alterar dívidas e devedores	Nov 14	Dec 4	15d	2	0%		
5	Ajustes do incremento 1 depois de validação com o usuário	Dec 5	Dec 25	15d	3	0%		
6	Ajustes do incremento 2 depois de validação com o usuário	Jan 2	Jan 22	15d	3	0%		
7	Ajustes do incremento 3 depois de validação com o usuário	Jan 23	Feb 12	15d	3	0%		
8	Ajustes do incremento 4 depois de validação com o usuário	Feb 13	Mar 5	15d	2	0%		
9	Entrega	Mar 5	Mar 5	1d	3	0%		