



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

MARCELO CARDOSO JOAQUIM

SISTEMA DE CONTROLE PARA EMPRESAS DE PUBLICIDADE

Assis

2015

MARCELO CARDOSO JOAQUIM

SISTEMA DE CONTROLE PARA EMPRESAS DE PUBLICIDADE

Trabalho de Conclusão de Curso
apresentado ao Instituto Municipal de
Ensino Superior de Assis, como
requisito do Curso de Graduação

Orientador: Luiz Ricardo Begosso

Área de Concentração: Desenvolvimento de Software.

Assis

2015

FICHA CATALOGRÁFICA

JOAQUIM, Marcelo

SISTEMA DE CONTROLE PARA EMPRESAS DE PUBLICIDADE / Marcelo Cardoso
Joaquim. Fundação Educacional do Município de Assis – FEMA – Assis, 2015.

65 páginas.

Orientador: Luiz Ricardo Begosso

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de
Assis - IMESA.

1. Controle

2. Prioridade

CDD: 001.61
Biblioteca da FEMA

SISTEMA DE CONTROLE PARA EMPRESAS DE PUBLICIDADE

MARCELO CARDOSO JOAQUIM

Trabalho de Conclusão de curso
apresentado ao Instituto Municipal de Ensino
Superior de Assis, como requisito do Curso
de Tecnólogo em Análise e
Desenvolvimento de Sistemas, analisado
pela seguinte comissão examinadora:

Orientador: _____

Analisador: _____

Assis

2015

DEDICATÓRIA

Dedico este trabalho a todos que me apoiaram nos meus estudos, minhas dificuldades, minhas alegrias, etc. Amigos e Família.

AGRADECIMENTOS

Ao professor, Luiz Ricardo Begosso, pela orientação e pelo constante estímulo transmitido durante o trabalho.

A todos os amigos que me apoiaram, colaborando direta ou indiretamente para realização deste trabalho.

Aos meus familiares que sempre me apoiaram e deram prioridade aos estudos, Mãe, Pai e Irmão.

RESUMO

Este trabalho descreve o desenvolvimento de um sistema feito para controlar a gerencia de contratos empresarias, visando a validade dos contratos, valores cobrados por cada propaganda, pagamentos realizados pelos clientes, controlar os gastos que a empresa tem, para que seja tudo registrado e no final de um mês ou de um ano, possa-se gerar um relatório com todas as despesas que a empresa teve, com todos os clientes que estão cadastrados no sistema, os clientes que estão bloqueados ou não. O sistema também possui um controle de usuário de acesso, controlando quem é o administrador do sistema, no caso que possui o acesso total a todas as funcionalidades. Este também avisa ao usuário quando ele entra no sistema se a algum contrato vencendo ou preste a vencer, para que se possível possa renovar o contrato já existente. Por fim, para a emissão de notas fiscais, o sistema verifica no cadastro do cliente se o CPF e o CNPJ são validos, para que não ocorra erros na geração das notas fiscais e emissão de boletos.

Palavras-chaves: Desenvolvimento, controle.

ABSTRACT

This paper describes the development of a system designed to control manages entrepreneurial contracts to the validity of contracts, amounts charged by each advertising, payments made by customers, control spending that the company has to be all recorded and at the end a month or a year, can be generated a report with all the expenses that the company had, to all customers who are registered in the system, customers that are locked or not. The system also features a user access control, controlling who is the system administrator in the event that has full access to all features. It also warns the user when he logs in to a winning contract or about to win, so that is possible to renew the existing contract. Finally, for the issuance of notes you stand, the system checks the customer's registration if the CPF and CNPJ are valid, so that there is errors in the generation of you stand notes and issuance of bills.

Keywords: Development, control.

LISTAS DE ILUSTRAÇÕES

Figura 1- Mapa Mental.....	Erro! Indicador não definido.
Figura 2 - Diagrama de UC Geral.....	Erro! Indicador não definido.
Figura 3 - UC Cadastrar Cliente.....	Erro! Indicador não definido.
Figura 4- UC Listar Cliente.....	Erro! Indicador não definido.
Figura 5 - UC Gerar Relatório de Clientes com Débitos	Erro! Indicador não definido.
Figura 6 - UC Gerar Relatório de Clientes Ativos	Erro! Indicador não definido.
Figura 7 - UC Gerar Relatório de Clientes Inativos.....	Erro! Indicador não definido.
Figura 8 - UC Gerar Relatório de todos os Clientes	Erro! Indicador não definido.
Figura 9 - UC Cadastrar Contrato.....	Erro! Indicador não definido.
Figura 10 - Gerar Relatório de todos os Contratos	Erro! Indicador não definido.
Figura 11- UC Listar Contrato	Erro! Indicador não definido.
Figura 12 - UC Gerar Relatório de Contratos Ativos	Erro! Indicador não definido.
Figura 13 - UC Gerar Relatório de Contratos Finalizados	Erro! Indicador não definido.
Figura 14 - UC Gerar Relatório de Clientes com Débitos Quitados.....	22
Figura 15 - UC Gerar Relatório de Clientes com Débitos Parcelados.....	23
Figura 16 - UC Lançar Despesa	24
Figura 17- UC Listar Despesa.....	25
Figura 18 - Gerar Relatório de Todas as Despesas	26
Figura 19 - UC Gerar Relatório de Despesa Mensal.....	27
Figura 20- UC Listar Contabilidade	28
Figura 21 - Gerar Relatórios de Propagandas Ativas.....	30
Figura 22 - Manter Usuário	31
Figura 23 - UC Logar Usuário	33
Figura 24 - Deslogar Usuário.....	34
Figura 25 - Diagrama de Classe	35
Figura 26 - Sequence Diagram - Cadastrar Cliente.....	36
Figura 27 - Sequence Diagram - Excluir Cliente	37
Figura 28 - Sequence Diagram - Listar Cliente	37
Figura 29 - Sequence Diagram - Criar Contrato.....	38
Figura 30 - Sequence Diagram - Excluir Contrato	38
Figura 31 - Sequence Diagram - Listar Contrato	39
Figura 32 - Sequence Diagram - Lançar Despesa.....	39
Figura 33 - Sequence Diagram - Excluir Despesa.....	40
Figura 34 - Sequence Diagram - Listar Despesas	40
Figura 35 - Sequence Diagram - Listar Contabilidade	41
Figura 36 - Sequence Diagram - Gerar Relatório	41
Figura 38 - Sequence Diagram - Excluir Usuário.....	Erro! Indicador não definido.
Figura 40 - Sequence Diagram - Realizar Log Out.....	Erro! Indicador não definido.
Figura 41 - Entidade Relacionamento.....	Erro! Indicador não definido.
Figura 44 - Sequenciamento de Atividade	46
Figura 45 - Camadas	47
Figura 46 – Model.....	48
Figura 47 - Classe Conexão	49
Figura 48 – 1º SELECT	49

Figura 49 - 2º SELECT	50
Figura 50 - INCLUDE	51
Figura 51 - UPDATE	51
Figura 52 – DELETE	52
Figura 53 – BLL	53
Figura 54 - Cassetes Imagine	54
Figura 55 - Tela Login	55
Figura 56 - Tela Inicial	55
Figura 57 - Tela Cadastro Cliente	56
Figura 58 - Tela Cadastro Usuário	56
Figura 59 - Tela Lista de Clientes	57
Figura 60 - Tela Lista de Contratos	57
Figura 61 - Tela Lançamento de Despesas	58
Figura 62 - Tela Lançamento de Contratos	58
Figura 63 - Tela Suporte	59
Figura 64 – Orçamento	61
Figura 65 - Cronograma	62

LISTAS DE ABREVIATURAS E SIGLAS

SGBD	Sistema Gerenciador de Banco de Dados
OO	Orientação a Objeto
C#	C Sharp
BD	Banco de Dados
UC	Caso de Uso
ER	Entidade Relacionamento

Sumário

1- INTRODUÇÃO	1
1.1- OBJETIVOS	1
1.2- JUSTIFICATIVA.....	1
1.3- MOTIVAÇÕES	1
1.4- ESTRUTURA DO TRABALHO.....	1
2- METODO DE DESENVOLVIMENTO.....	3
2.1- TECNOLOGIAS UTILIZADAS.....	3
2.1.1- C#	3
2.1.2- SQL Express.....	3
2.1.3- Astah	4
2.1.4- FreeMind.....	4
2.1.5- DBDesignerFork.....	4
2.1.6- VISIO.....	4
3- DESCRIÇÃO GERAL DO SISTEMA	5
3.1- ANÁLISE DE REQUISITOS.....	5
3.2- EVENTOS DO SISTEMA	5
3.3- DIAGRAMAS.....	6
3.3.1- Mapa Mental.	6
3.3.2- Diagrama de UC (Caso de Uso).....	7
3.3.2.2 – UC Listar Cliente	10
3.3.2.3 - UC Gerar Relatório de Clientes com Débitos.....	12
3.3.2.4 - UC Gerar Relatório de Clientes Ativos	13
3.3.2.5 - UC Gerar Relatório de Clientes Inativos	14
3.3.2.6 – UC Gerar Relatório de Todos os Clientes	15
3.3.2.7 – UC Manter Contrato	16
3.3.2.8 – UC Gerar Relatório de todos os Contratos	17
3.3.2.9 – UC Listar Contrato	18
3.3.2.10 – UC Gerar Relatório de Contratos Ativos	20
3.3.2.11 – UC Gerar Relatório de Contratos Finalizados.....	21
3.3.2.12 – UC Gerar Relatório de Cliente com Débitos Quitados	22
3.3.2.13 – UC Gerar Relatório de Clientes com Débitos Parcelados	23
3.3.2.14 – UC Manter Despesa.....	24
3.3.2.15 – UC Listar Despesas	25

3.3.2.16 – UC Gerar Relatório de todas as Despesas	26
3.3.2.17 – UC Gerar Relatório de Despesas Mensal	27
3.3.2.18 – UC Listar Contabilidade	28
3.3.2.19 – UC Gerar Relatórios de Propagandas Ativas	30
3.3.2.20 – UC Manter Usuários.....	31
3.3.2.21 – UC Logar Usuários	33
3.3.2.22 – UC Deslogar Usuários	34
3.3.3- Diagrama de Classe	35
3.3.4- Diagrama de Sequência.....	36
3.3.5- Entidade Relacionamento.....	44
3.3.6- Estrutura Analítica de Projeto.	44
3.3.7- Sequenciamento de Atividade.	45
4- IMPLEMENTAÇÃO	46
4.1- ORGANIZAÇÃO DO PROJETO	47
4.2 – Interfaces do Projeto	54
5 – ORÇAMENTO	60
5.1 – ESTIMATIVA DE CUSTOS.....	60
5.2 – CUSTOS DE MATERIAIS.....	60
5.3 – ORÇAMENTO DO PROJETO.....	61
6 – CRONOGRAMA.....	62
7 – CONCLUSÃO	63
8 – REFERENCIAS.....	64

1- INTRODUÇÃO

Como a publicidade é o maior meio de uma empresa se mostrar no mercado surgiu-se IMAGINE PAINES, que faz publicidade das empresas que a contratam. Com o contrato feito, cria-se uma mídia de apresentação que ficará em execução em um painel de LED, com alcance para mais de 20 mil pessoas diariamente.

Com isso surgiu a necessidade do sistema de controle, para que haja um melhor aproveitamento do tempo um mais preciso meio de gerencia de contrato.

1.1- OBJETIVOS

Este projeto tem o objetivo de desenvolver um software desktop para o gerenciamento de contratos de publicidade, no qual cada empresa contrata um determinado pacote de tempo de exibição publicitária, seja por mês, bimestre ou semestre. O software deverá realizar toda a parte de contabilidade e relatórios da empresa.

1.2- JUSTIFICATIVA

Com a realização deste projeto, será muito mais fácil e eficiente o gerenciamento dos contratos de publicidade, as datas de vencimentos, a quantidade de propagandas que foram montadas para cada empresa, uma listagem de todas as propagandas que estão ativas no painel ou serão ativas.

1.3- MOTIVAÇÕES

Com o surgimento da empresa IMAGINE PAINES, houve a necessidade de um sistema para gerenciamento dos contratos de publicidade ativos.

1.4- ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte forma:

O Capítulo 1 irá descrever a introdução e o detalhamento do projeto a ser desenvolvido e implantado.

O Capítulo 2 descreverá a metodologia de análise de sistemas a ser adotada para a modelagem do problema, a Análise Orientada a Objetos.

O Capítulo 3 descreverá o ambiente de desenvolvimento do software, bem como o modelo de banco de dados a ser utilizado.

O Capítulo 4 apresentará a modelagem do software, de acordo com a metodologia adotada.

O Capítulo 5 apresentará os layouts de telas e relatórios e o trabalho será encerrado pelas Conclusões e Referências Bibliográficas.

2- METODO DE DESENVOLVIMENTO

Para análise e desenvolvimento desta aplicação, será utilizado a metodologia de Orientação a Objetos, que é utilizada para um melhor entendimento dos requisitos e uma melhor organização na estrutura de dados, tendo um código mais limpo e eficiente na manipulação dos dados ali processados.

2.1- TECNOLOGIAS UTILIZADAS

Para o desenvolvimento desta aplicação desktop foi utilizado a linguagem de programação C#, usando o Visual Studio 2013 como editor e o BD foi o SQL Express, poderosas ferramentas da Microsoft. Para a modelagem do sistema foi utilizado Astah, DBDesignerFork e o FreeMind, montando todos os diagramas.

2.1.1- C#

C# combina os melhores elementos de múltiplas linguagens de ampla difusão como C++, Java, Visual Basic ou Delphi. De fato, seu criador Anders Heljsberg foi também o criador de muitas outras linguagens e meios como Turbo Pascal, Delphi ou Visual J++. A ideia principal atrás da linguagem é combinar a potência de linguagens como C++ com a simplicidade de linguagens como Visual Basic, e que ademais a migração a esta linguagem pelos programadores de C/C++/Java seja o mais imediato possível (Miguel Angel Alvarez, 2001).

2.1.2- SQL Express

O Microsoft SQL Server 2008 Express é uma plataforma de banco de dados que se baseia no Microsoft SQL Server. O SQL Server Express facilita o desenvolvimento de aplicativos ricos em recursos controlados por dados, oferece segurança de armazenamento aprimorada e é rápido de implantar (Microsoft, 2008).

2.1.3- Astah

O Astah é um programa que permite o desenvolvimento de projetos UML, o desenvolvimento de diversos diagramas, sendo a ferramenta de fácil entendimento e manipulação (Astah, 2006).

2.1.4- FreeMind

O FreeMind é um programa que permite o desenvolvimento de projetos, como ele pode-se montar o mapa mental do projeto, com todos as funcionalidades que o sistema irá possuir (Source Forge, 2013).

2.1.5- DBDesignerFork

O DBDesignerFork é um programa que permite o desenvolvimento de modelos de dados, com PK e FK, facilitando para o desenvolvimento (DB Designer, 2009).

2.1.6- VISIO

O Visio Professional 2013 torna mais fácil do que nunca que pessoas e equipes criem e compartilhem diagramas profissionais e versáteis que simplifiquem informações complexas (Microsoft, 2013).

3- DESCRIÇÃO GERAL DO SISTEMA

O sistema faz todo o controle de contratos publicitários e cadastrados, ele avisará com um mês de antecedência as atividades que estão previstas para acontecer, término de um contrato, aniversário de um cliente ou representante da empresa.

O software também tem a função de controle de gastos da empresa, todas as entradas e saídas, vendas, despesas, manutenção e impostos, tudo calculado.

Este também só pode ser acessado por um usuário cadastrado, sendo que somente o administrador do sistema pode realizar o cadastro de um usuário, por medida de segurança.

3.1- ANALISE DE REQUISITOS

Para o desenvolvimento foi feita a entrevista padrão com o administrador da empresa IMAGINE Painéis de LED, Maurilio Tiago Cardoso Joaquim, sendo que os seguintes requisitos foram identificados.

- Desenvolver um cadastro, para que possa ser feito o controle de todos os clientes, no cadastro deverá conter todos os dados do cliente e da empresa.
- O sistema deverá avisar com um mês de antecedência o fim de um contrato e mandar um lembrete no dia do aniversário do cliente.
- Controlar os gastos da empresa, entrada/saída, sendo que os gastos de saída, deverá serem todos especificados.
- Cadastro de Usuário, para que haja uma melhor segurança de acesso ao sistema.
- Relatório de Clientes Ativos, Inativos, Gastos.
- Gerar um arquivo com o e-mail de todos os clientes cadastrados, para que possam ser encaminhados e-mails publicitários ou avisos.

3.2- EVENTOS DO SISTEMA

1. Manter Cliente.
2. Manter Contrato.
3. Manter Despesa.
4. Listar Contabilidade.
5. Gerar Relatório de Clientes com Débitos.
6. Gerar Relatório de Clientes Ativos.
7. Gerar Relatório de Clientes Inativos.
8. Gerar Relatório de Todos os Clientes.
9. Gerar Relatório de Todos os Contratos.
10. Gerar Relatório de Contratos Ativos.
11. Gerar Relatório de Contratos Finalizados
12. Gerar Relatório de Clientes com Débitos Quitados.
13. Gerar Relatório de Clientes com Débitos Parcelados.
14. Gerar Relatório de Todas as Despesas.
15. Gerar Relatório de Despesa Mensal.
16. Gerar Relatório de Propagandas Ativas.
17. Manter Usuários.
18. Logar Usuário.
19. Deslogar Usuário.

3.3- DIAGRAMAS

A seguir, os diagramas que demonstram como é o funcionamento do sistema, com ou sem a interação do usuário.

3.3.1- Mapa Mental.



Figura 1 - Mapa Mental

3.3.2- Diagrama de UC (Caso de Uso).

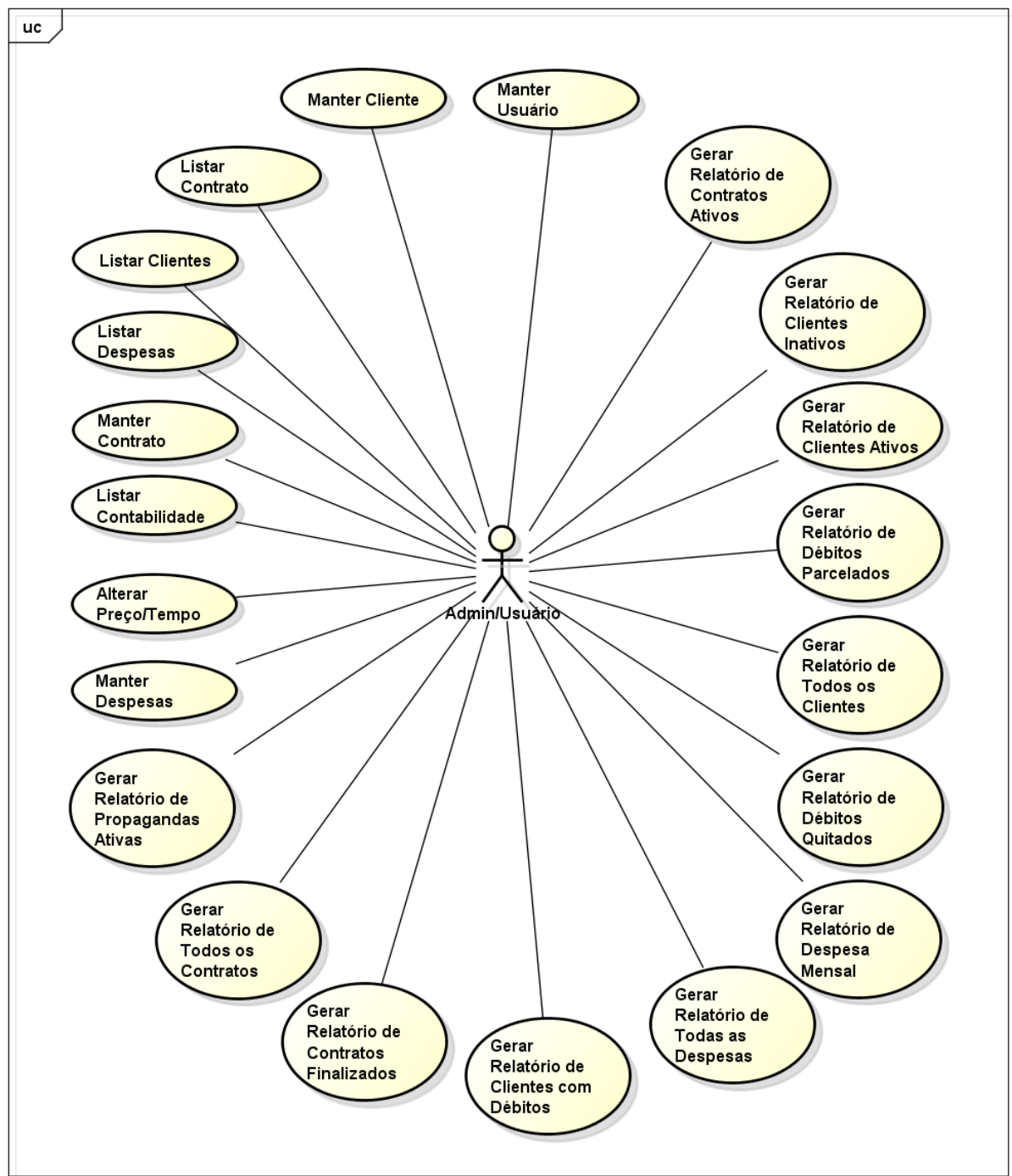
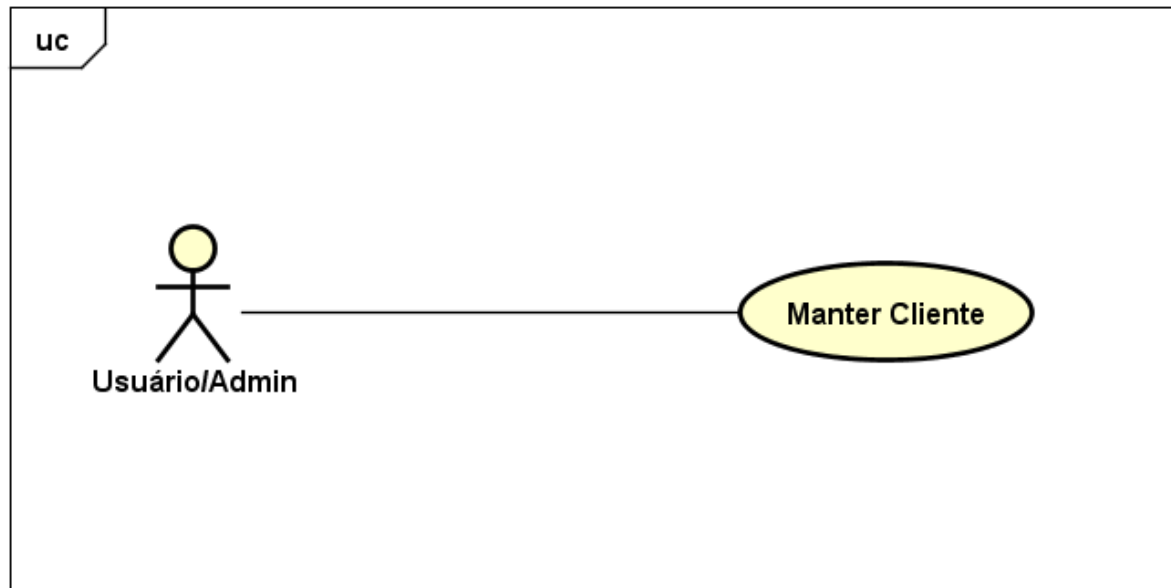


Figura 2 - Diagrama de UC Geral

3.3.2.1 – UC Manter Cliente



powered by Astah

Figura 3 - UC Cadastrar Cliente

1- Finalidade/Objetivo

- Permite que o Ator do sistema possa cadastrar, editar e excluir os clientes, sem exceções.

2- Ator

- Usuário
- Administrador

3- Precondições

- O Ator necessita fazer o login no sistema, para serem identificados.

4- Fluxo Principal

- O ator do sistema seleciona a opção cadastro no menu principal e em seguida seleciona o item Cliente.
- O sistema pede para informar os dados do cliente e da empresa a quem ele pertence.
- O operador do sistema informa os dados e finaliza a operação.
- O sistema verifica se todos os dados informados estão corretos e em seguida mostra uma mensagem de conclusão.

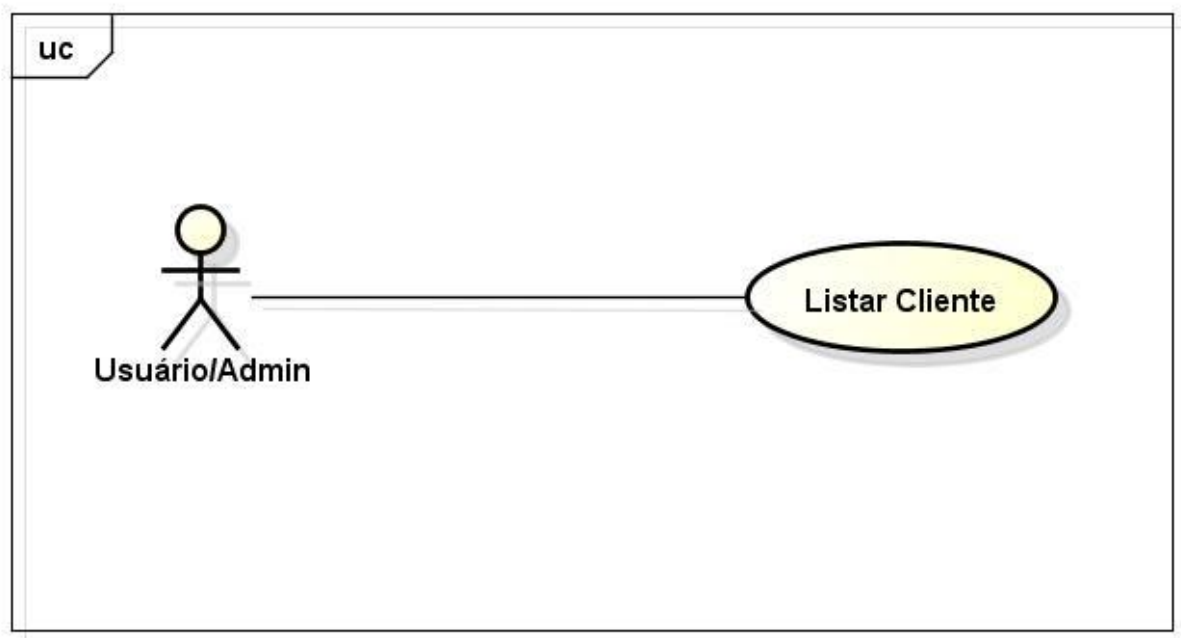
5- Fluxo Alternativo

- O ator do sistema seleciona a opção cadastro no menu principal e em seguida seleciona o item Editar Cliente.
- O sistema mostra os dados do cliente, para que possam ser alterados.
- O ator seleciona a opção cancelar.
- O sistema volta para a tela inicial.
- O ator seleciona o item excluir cliente.
- O sistema confirma a ação e a exclui.

6- Casos de Teste

- Verifica se todos os campos obrigatórios estão preenchidos.
- Verifica se RG, CPF e CNPJ são verdadeiros.

3.3.2.2 – UC Listar Cliente



powered by Astah

Figura 4 - UC Listar Cliente

1- Finalidade/Objetivo

- Mostrar todos os clientes cadastrados no sistema.

2- Ator

- Usuário
- Administrador

3- Precondições

- O ator necessita fazer o login no sistema e ter um ou mais clientes cadastrados.

4- Fluxo Principal

- O ator do sistema seleciona a opção cadastro no menu principal e em seguida seleciona o item Listar Cliente.
- O sistema pergunta se a regra de pesquisa e mostra os clientes cadastrados, conforme o solicitado.

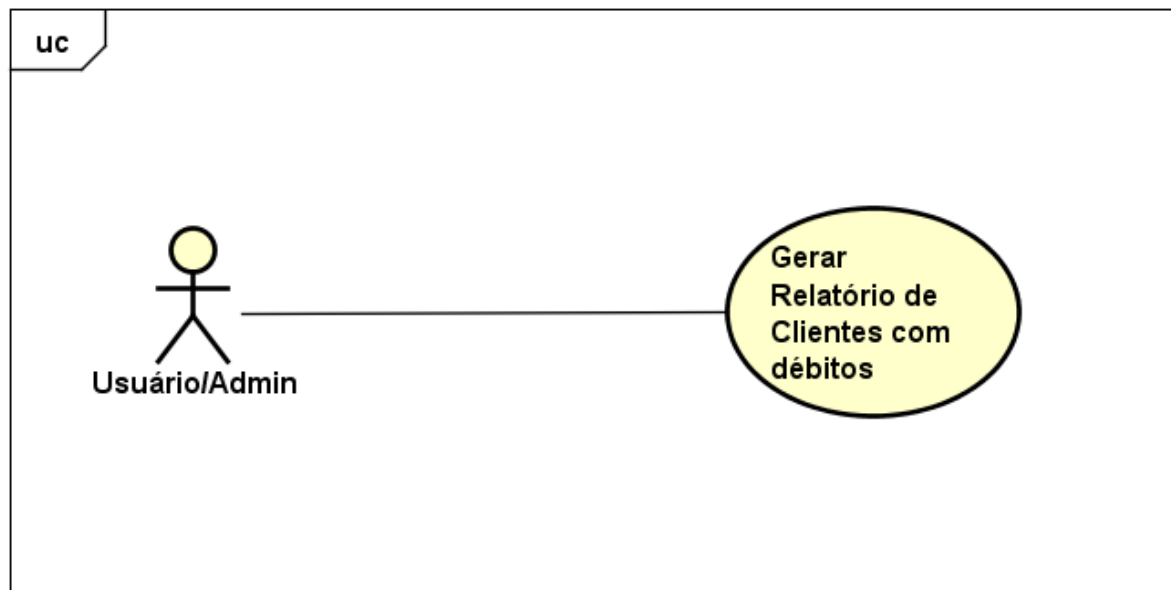
5- Fluxo Alternativo

- O ator seleciona o item cancelar.
- O sistema volta para a tela inicial.

6- Casos de Teste

- O sistema verifica quais os clientes com propaganda ativa.
- O sistema verifica quais os clientes com propaganda inativa.
- O sistema verifica quais os clientes com débitos.

3.3.2.3 - UC Gerar Relatório de Clientes com Débitos



powered by Astah

Figura 5 - UC Gerar Relatório de Clientes com Débitos

1- Finalidade/Objetivo

- Permite que o Ator do sistema possa gerar um relatório com todos os clientes com débitos.

2- Ator

- Usuário
- Administrador

3- Precondições

- O Ator necessita fazer o login no sistema.

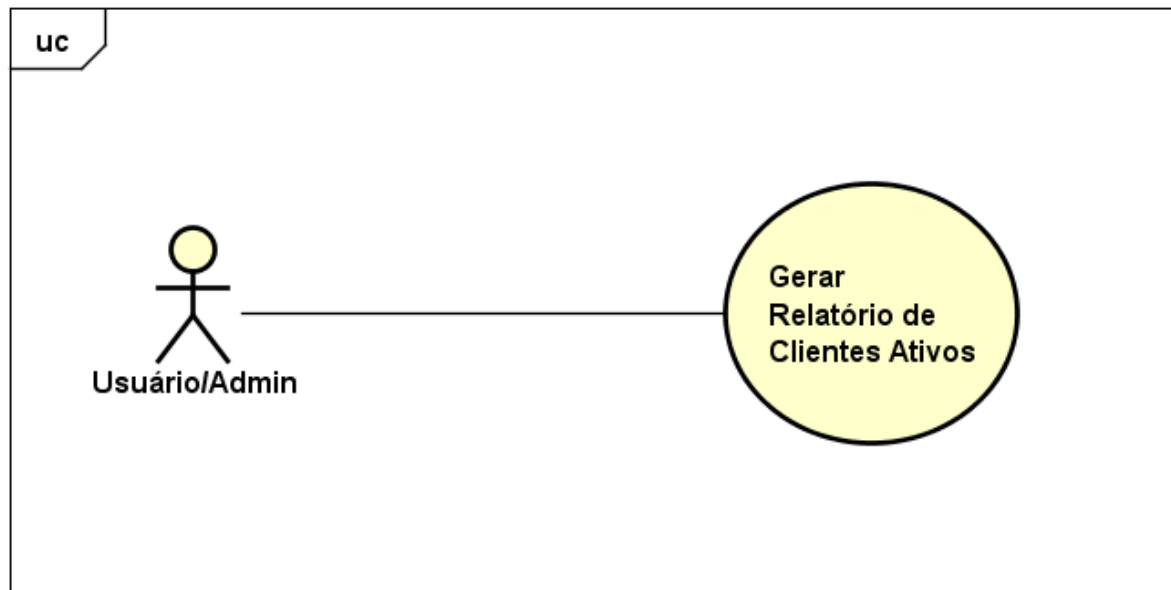
4- Fluxo Principal

- O ator do sistema seleciona o item Relatório, em seguida Clientes com Débitos.
- O sistema pede para informa os mesmos.

5- Casos de Teste

- Verifica se existem clientes com débitos.

3.3.2.4 - UC Gerar Relatório de Clientes Ativos



powered by Astah

Figura 6 - UC Gerar Relatório de Clientes Ativos

1- Finalidade/Objetivo

- Permite que o Ator do sistema possa gerar um relatório com Ativos (Que não estão bloqueados).

2- Ator

- Usuário
- Administrador

3- Precondições

- O Ator necessita fazer o login no sistema.

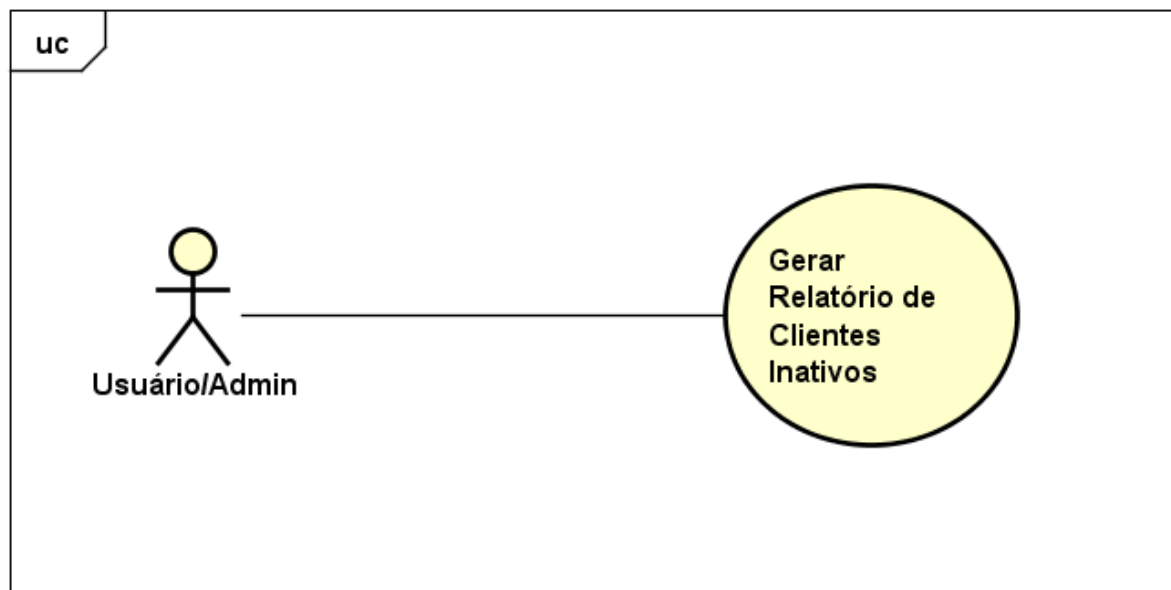
4- Fluxo Principal

- O ator do sistema seleciona o item Relatório, em seguida Clientes Ativos.
- O sistema pede para informa os mesmos.

5- Casos de Teste

- Verifica se existem clientes ativos.

3.3.2.5 - UC Gerar Relatório de Clientes Inativos



powered by Astah

Figura 7 - UC Gerar Relatório de Clientes Inativos

1- Finalidade/Objetivo

- Permite que o Ator do sistema possa gerar um relatório com todos os clientes Inativos (Que estão bloqueados).

2- Ator

- Usuário
- Administrador

3- Precondições

- O Ator necessita fazer o login no sistema.

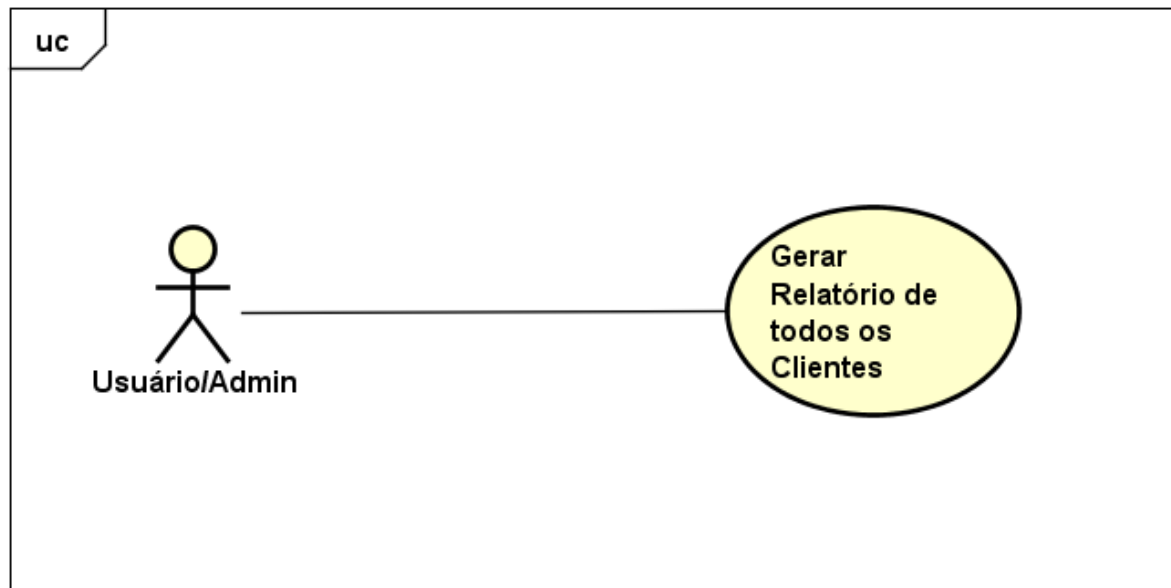
4- Fluxo Principal

- O ator do sistema seleciona o item Relatório, em seguida Clientes com Inativos.
- O sistema pede para informa os mesmos.

5- Casos de Teste

- Verifica se existem clientes inativos.

3.3.2.6 – UC Gerar Relatório de Todos os Clientes



powered by Astah

Figura 8 - UC Gerar Relatório de todos os Clientes

1- Finalidade/Objetivo

- Permite que o Ator do sistema possa gerar um relatório com todos os clientes cadastrados.

2- Ator

- Usuário
- Administrador

3- Precondições

- O Ator necessita fazer o login no sistema.

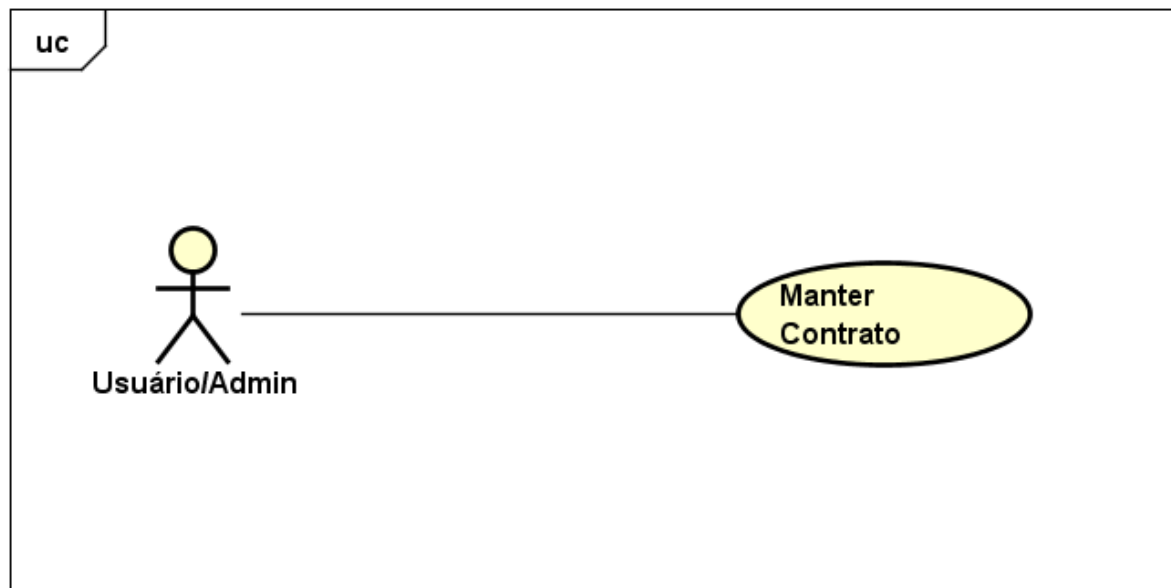
4- Fluxo Principal

- O ator do sistema seleciona o item Relatório, em seguida Todos os Clientes.
- O sistema pede para informa os mesmos.

5- Casos de Teste

- Verifica se existem clientes cadastrados.

3.3.2.7 – UC Manter Contrato



powered by Astah

Figura 9 - UC Cadastrar Contrato

1- Finalidade/Objetivo

- Permite que o ator cadastre, edite ou exclua um contrato.

2- Ator

- Usuário
- Administrador

3- Precondições

- O ator necessita fazer o login no sistema e ter um ou mais clientes cadastrados.

4- Fluxo Principal

- O ator do sistema seleciona a opção cadastro no menu principal e em seguida seleciona o item Criar Contrato.
- O sistema pede para informar os dados e verifica os mesmos se estão corretos.
- O sistema emite um documento com os dados do contrato.

5- Fluxo Alternativo

- O ator digita o código de um contrato já cadastrado.
- O sistema mostra os dados do contrato, deixando-o altera-los.
- O ator seleciona o item cancelar.
- O sistema volta para a tela inicial.

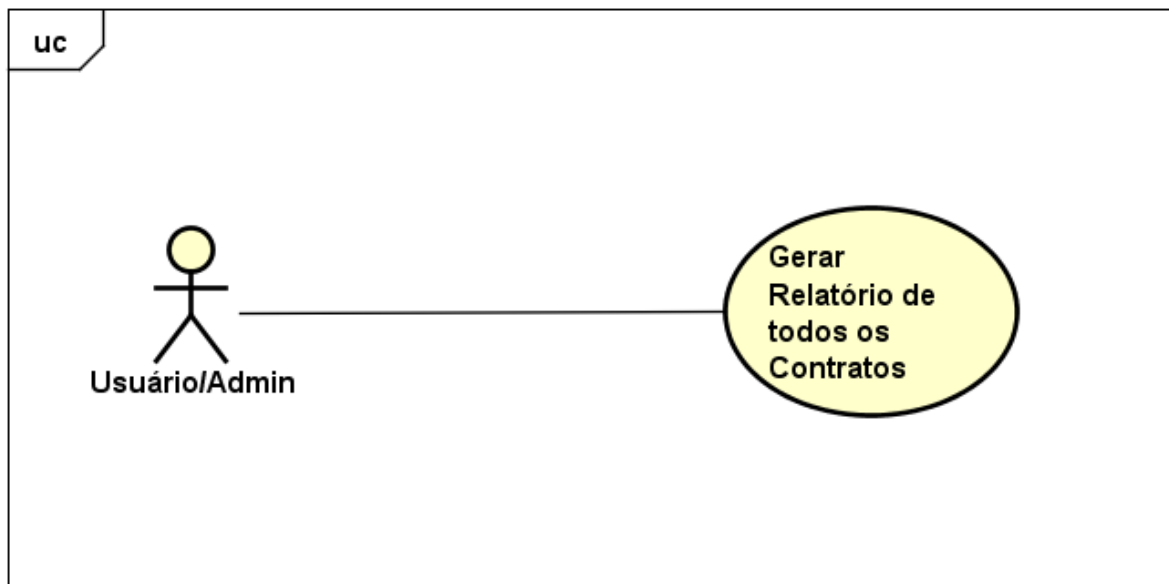
- O ator seleciona o item editar.
- O ator seleciona o item salvar.
- O sistema verifica os dados e salva.

- O ator seleciona o item excluir.
- O sistema confirma e exclui.

6- Casos de Teste

- O sistema verifica se o cliente não possui débitos.
- O sistema verifica se o cliente não possui bônus.

3.3.2.8 – UC Gerar Relatório de todos os Contratos



powered by Astah

Figura 10 - Gerar Relatório de todos os Contratos

1- Finalidade/Objetivo

- Permite que o Ator do sistema possa gerar um relatório com todos os contratos cadastrados.

2- Ator

- Usuário
- Administrador

3- Precondições

- O Ator necessita fazer o login no sistema.

4- Fluxo Principal

- O ator do sistema seleciona o item Relatório, em seguida Todos os Contratos.
- O sistema pede para informa os mesmos.

5- Casos de Teste

- Verifica se existem contratos cadastrados.

3.3.2.9 – UC Listar Contrato

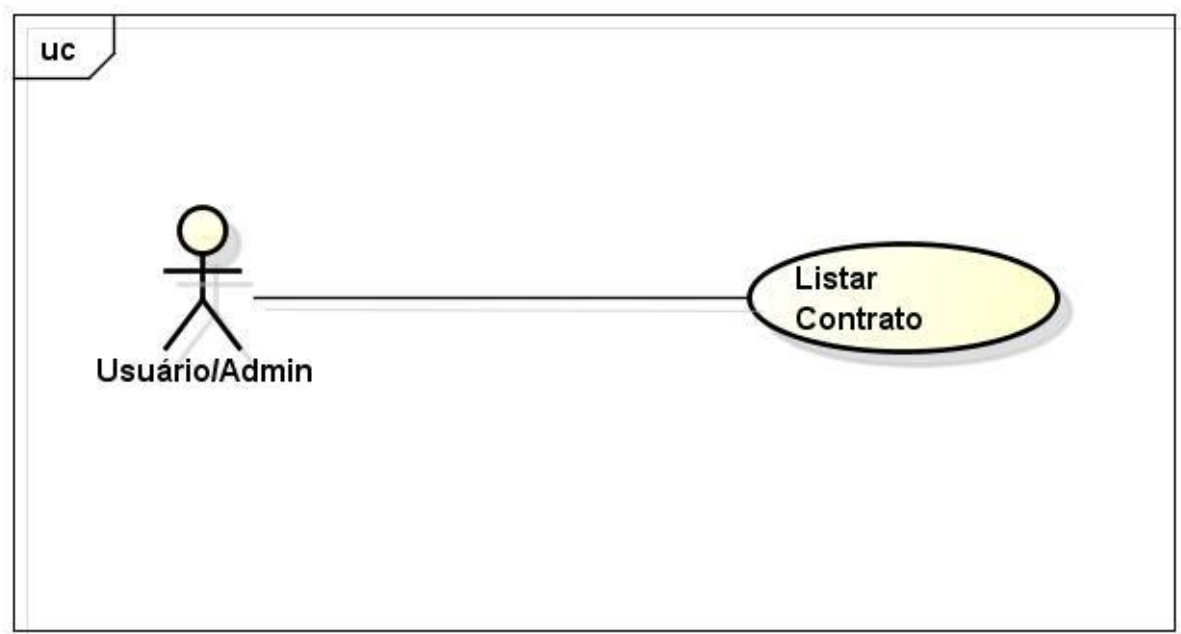


Figura 11 - UC Listar Contrato

1- Finalidade/Objetivo

- Permite o ator listar todos os contratos cadastrados, sejam eles ativos ou inativos.

2- Ator

- Usuário
- Administrador

3- Precondições

- O ator necessita fazer o login no sistema e ter um ou mais clientes cadastrados no sistema.

4- Fluxo Principal

- O ator seleciona o item clientes e em seguida seleciona o filtro de clientes conforme o desejado.

5- Fluxo Alternativo

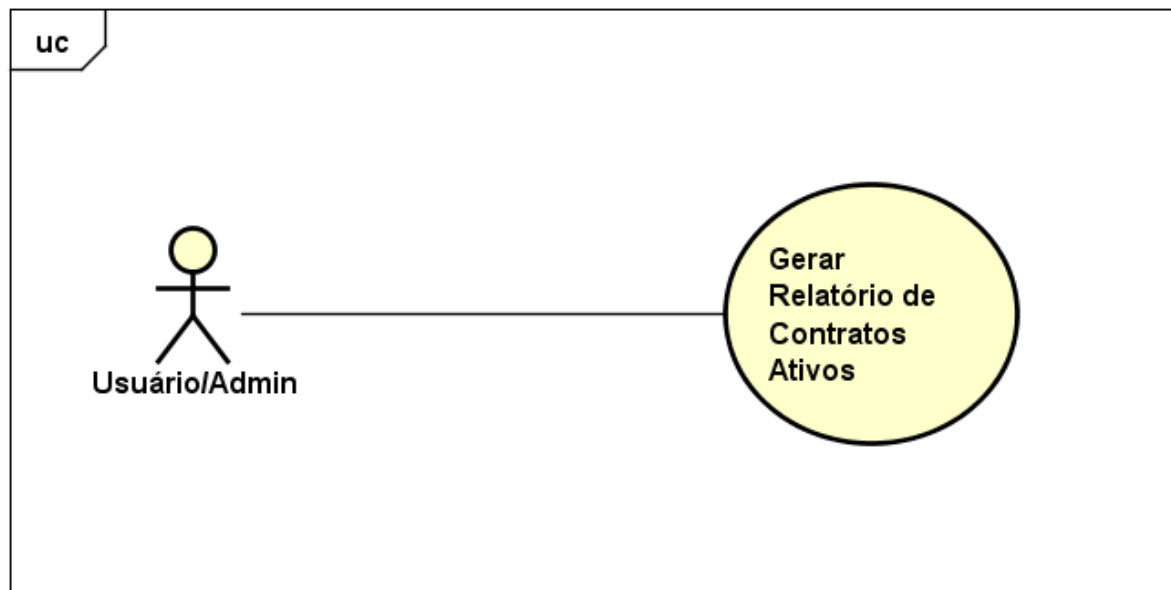
- O ator seleciona o item novo.
- O sistema o endereça para a tela de cadastro de cliente.

- O ator seleciona o item cancelar.
- O sistema volta para a tela inicial.

6- Casos de Teste

- O sistema verifica de acordo com as especificações do ator, se o ator deseja, clientes ativos, clientes inativos, clientes com débitos e por fim todos os clientes cadastrados.

3.3.2.10 – UC Gerar Relatório de Contratos Ativos



powered by Astah 

Figura 12 - UC Gerar Relatório de Contratos Ativos

1- Finalidade/Objetivo

- Permite que o Ator do sistema possa gerar um relatório com todos os contratos ativos.

2- Ator

- Usuário
- Administrador

3- Precondições

- O Ator necessita fazer o login no sistema.

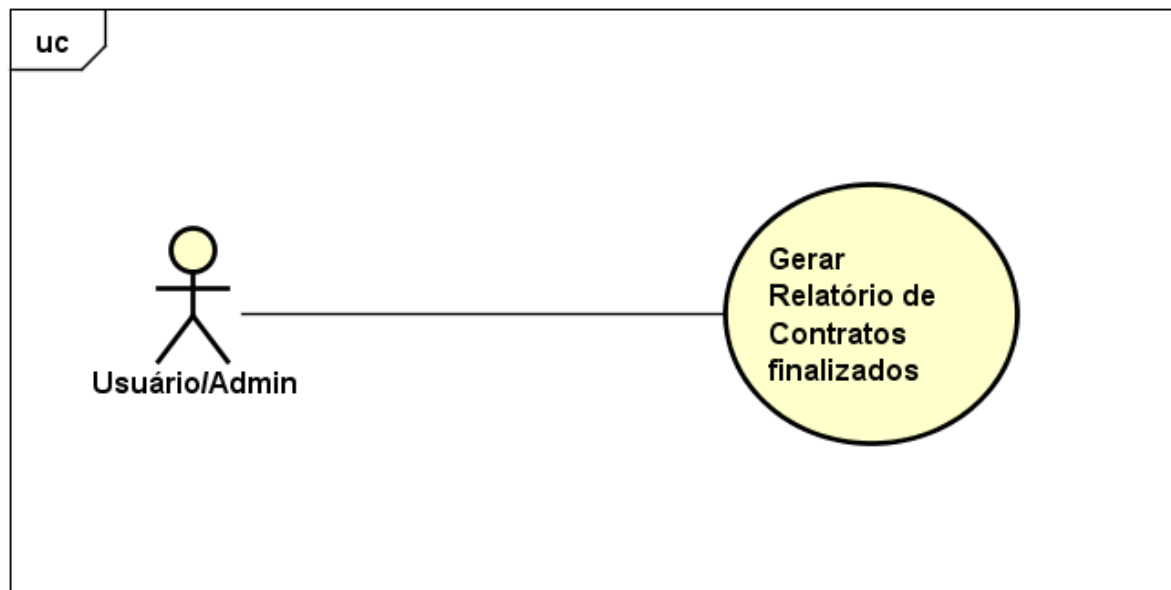
4- Fluxo Principal

- O ator do sistema seleciona o item Relatório, em seguida Contratos Ativos.
- O sistema pede para informa os mesmos.

5- Casos de Teste

- Verifica se existem contratos ativos.

3.3.2.11 – UC Gerar Relatório de Contratos Finalizados



powered by Astah

Figura 13 - UC Gerar Relatório de Contratos Finalizados

1- Finalidade/Objetivo

- Permite que o Ator do sistema possa gerar um relatório com todos os contratos finalizados.

2- Ator

- Usuário
- Administrador

3- Precondições

- O Ator necessita fazer o login no sistema.

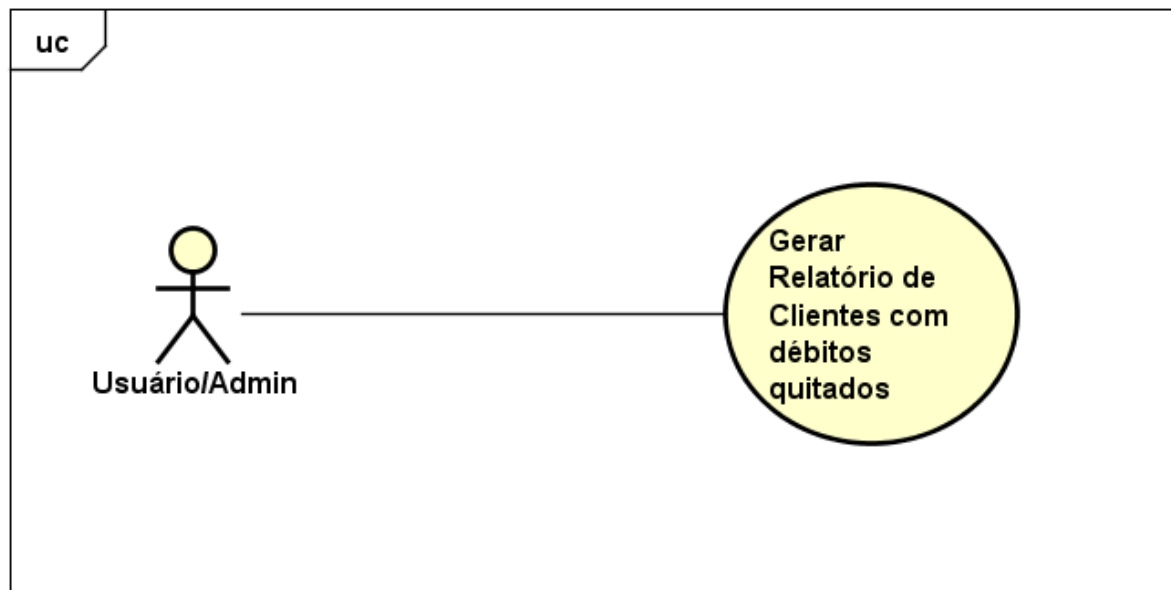
4- Fluxo Principal

- O ator do sistema seleciona o item Relatório, em seguida Contratos Finalizados.
- O sistema pede para informa os mesmos.

5- Casos de Teste

- Verifica se existem contratos finalizados.

3.3.2.12 – UC Gerar Relatório de Cliente com Débitos Quitados



powered by Astah

Figura 14 - UC Gerar Relatório de Clientes com Débitos Quitados

1- Finalidade/Objetivo

- Permite que o Ator do sistema possa gerar um relatório com todos os clientes com débitos quitados.

2- Ator

- Usuário
- Administrador

3- Precondições

- O Ator necessita fazer o login no sistema.

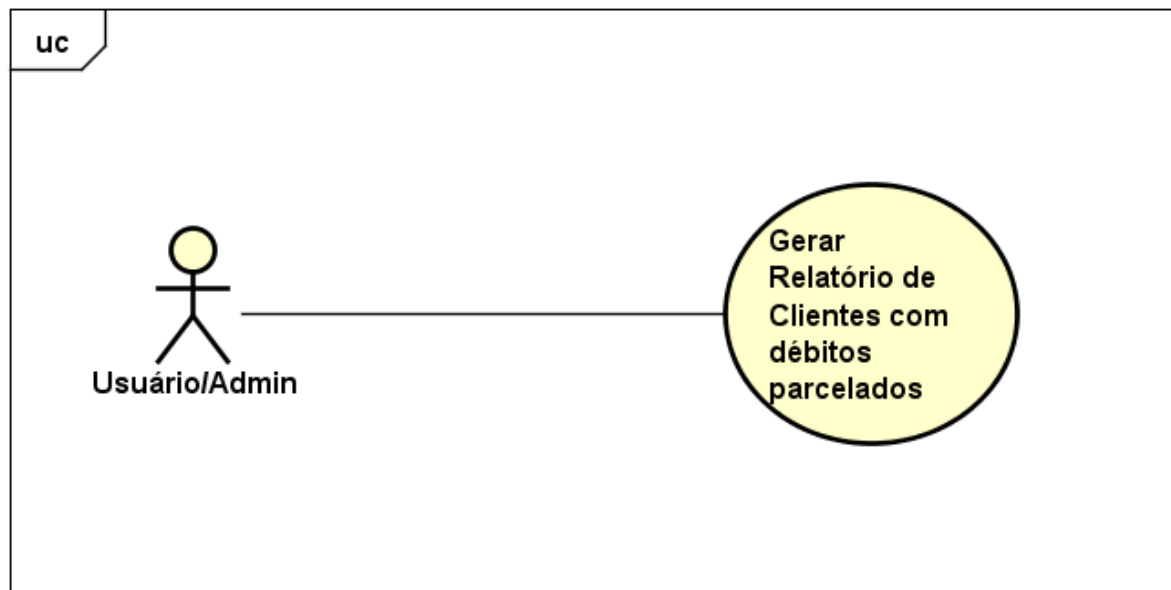
4- Fluxo Principal

- O ator do sistema seleciona o item Relatório, em seguida Clientes sem Débitos.
- O sistema pede para informa os mesmos.

5- Casos de Teste

- Verifica se existem clientes sem débitos.

3.3.2.13 – UC Gerar Relatório de Clientes com Débitos Parcelados



powered by Astah

Figura 15 - UC Gerar Relatório de Clientes com Débitos Parcelados

1- Finalidade/Objetivo

- Permite que o Ator do sistema possa gerar um relatório com todos os clientes com débitos parcelados.

2- Ator

- Usuário
- Administrador

3- Precondições

- O Ator necessita fazer o login no sistema.

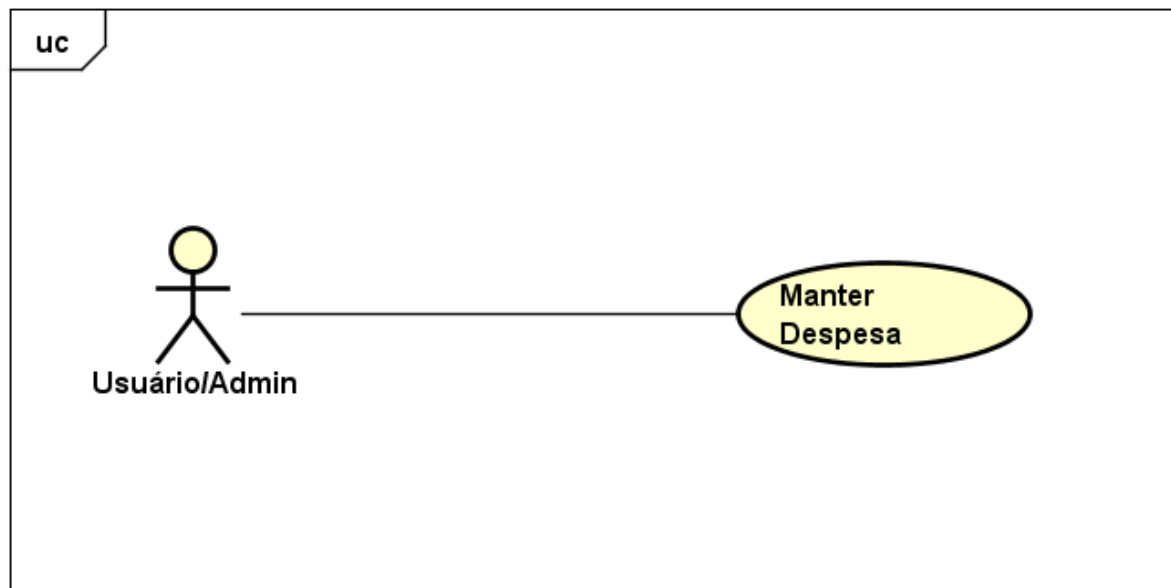
4- Fluxo Principal

- O ator do sistema seleciona o item Relatório, em seguida Clientes com Débitos Parcelados.
- O sistema pede para informa os mesmos.

5- Casos de Teste

- Verifica se existem clientes com débitos parcelados.

3.3.2.14 – UC Manter Despesa



powered by Astah

Figura 16 - UC Lançar Despesa

1- Finalidade/Objetivo

- Permite ao ator cadastrar, edite e exclua todos os gastos com materiais, manutenção e outros que a empresa teve.

2- Ator

- Usuário
- Administrador

3- Precondições

- O ator necessita fazer o login no sistema.

4- Fluxo Principal

- O ator seleciona o Item despesa e em seguida o item lançar despesas.

5- Fluxo Alternativo

- O ator seleciona o item cancelar.
- O sistema volta para a tela inicial.

- O ator seleciona o Item editar.

- O ator seleciona o item salvar.
- O ator seleciona o item excluir.
- O sistema confirma e exclui.

6- Casos de Teste

- O sistema verifica se o caixa da empresa é compatível com a compra.

3.3.2.15 – UC Listar Despesas

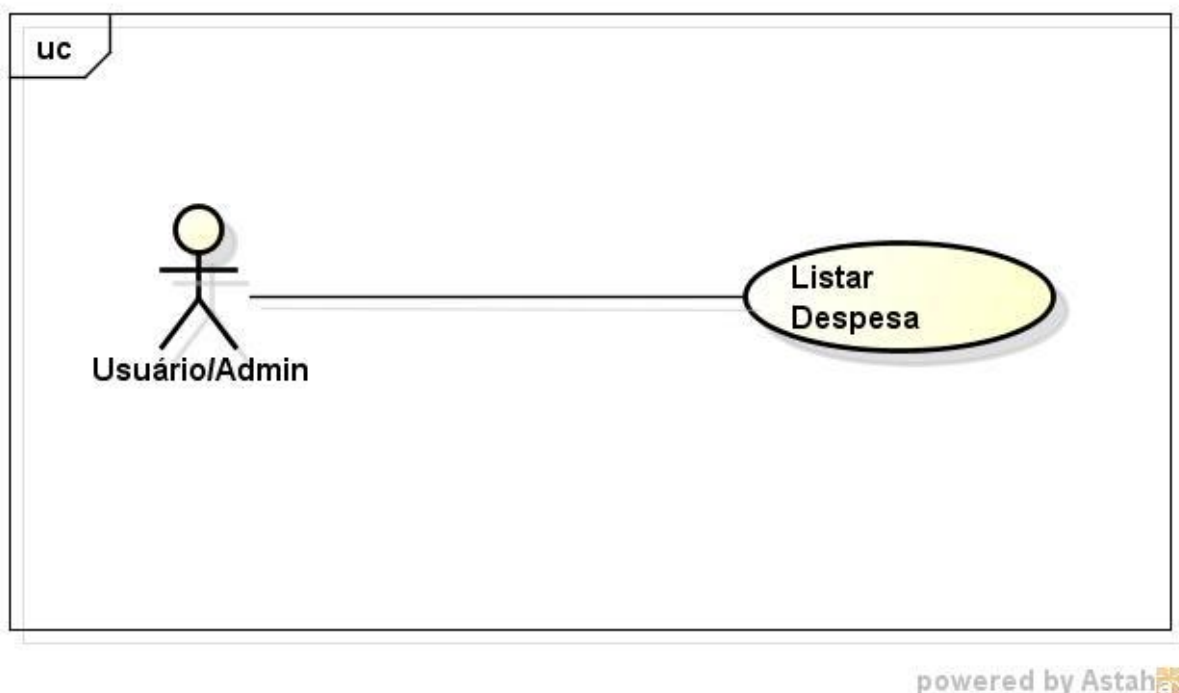


Figura 17- UC Listar Despesa

1- Finalidade/Objetivo

- Permite ao ator ver todas as despesas já lançadas no sistema.

2- Ator

- Usuário
- Administrador

3- Precondições

- O ator necessita fazer login no sistema e ter um ou mais despesas lançadas no sistema.

4- Fluxo Principal

- O ator seleciona o item despesas e em seguida o item todos os lançamentos.

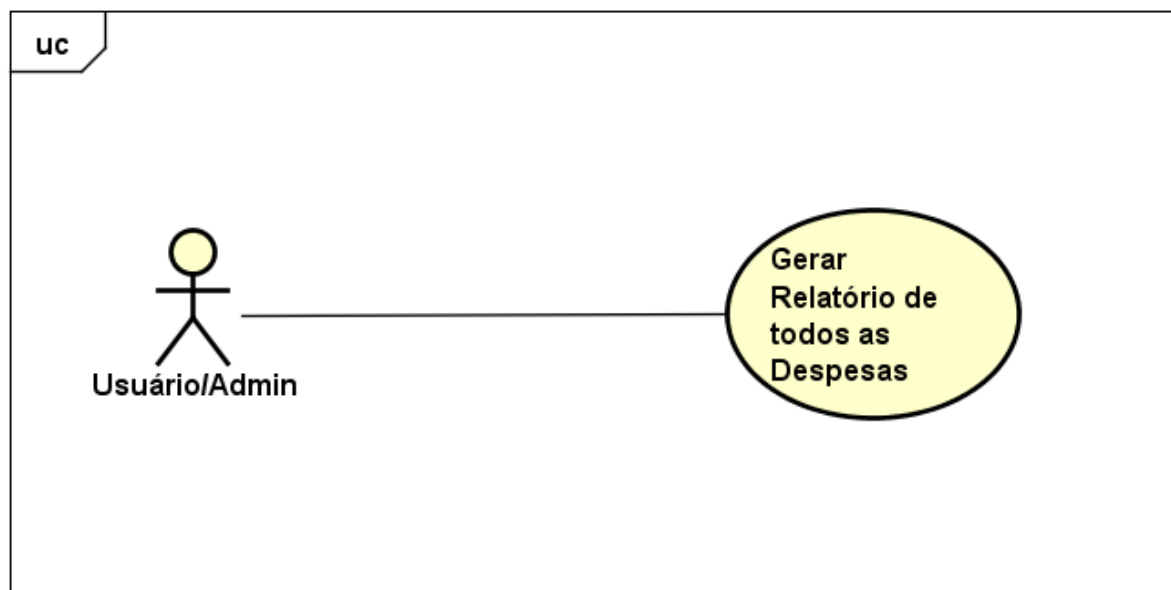
5- Fluxo Alternativo

- O sistema filtra os lançamentos por mês e ano.
- O ator seleciona o item cancelar.
- O sistema volta para a tela inicial.

6- Casos de Teste

- O sistema verifica a o mês e o ano, para informar.

3.3.2.16 – UC Gerar Relatório de todas as Despesas



powered by Astah

Figura 18 - Gerar Relatório de Todas as Despesas

1- Finalidade/Objetivo

- Permite que o Ator do sistema possa gerar um relatório com todos as despesas.

2- Ator

- Usuário
- Administrador

3- Precondições

- O Ator necessita fazer o login no sistema.

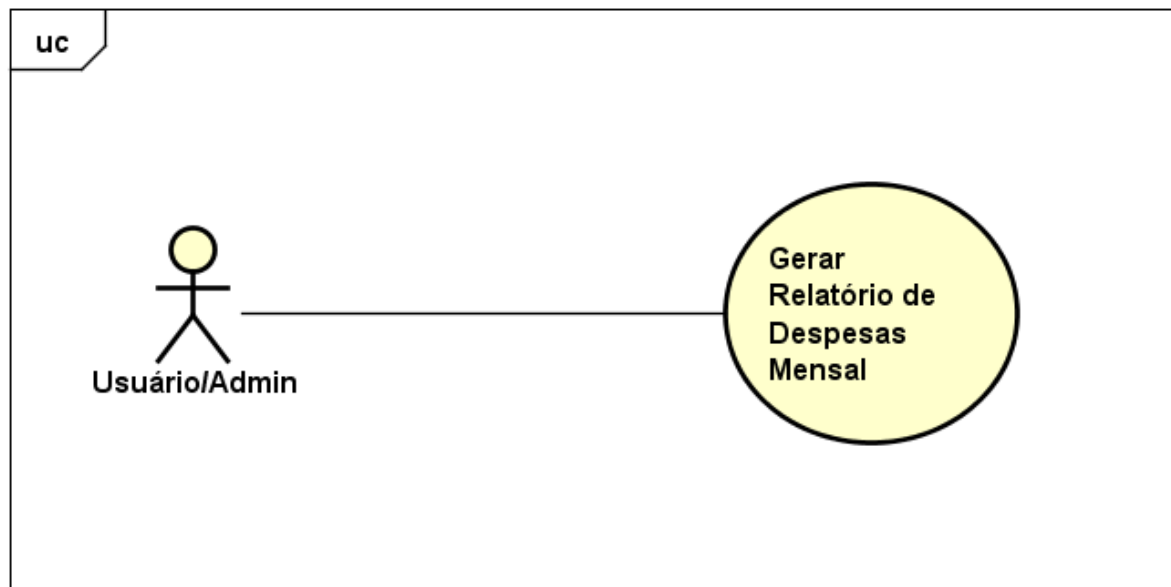
4- Fluxo Principal

- O ator do sistema seleciona o item Relatório, em seguida Todas as Despesas.
- O sistema pede para informa os mesmos.

5- Casos de Teste

- Verifica se existem despesas.

3.3.2.17 – UC Gerar Relatório de Despesas Mensal



powered by Astah

Figura 19 - UC Gerar Relatório de Despesa Mensal

1- Finalidade/Objetivo

- Permite que o Ator do sistema possa gerar um relatório com todos as despesas do mês.

2- Ator

- Usuário
- Administrador

3- Precondições

- O Ator necessita fazer o login no sistema.

4- Fluxo Principal

- O ator do sistema seleciona o item Relatório, em seguida Despesas Mensais.
- O sistema pede para informa os mesmos.

5- Casos de Teste

- Verifica se existem despesas mensal.

3.3.2.18 – UC Listar Contabilidade

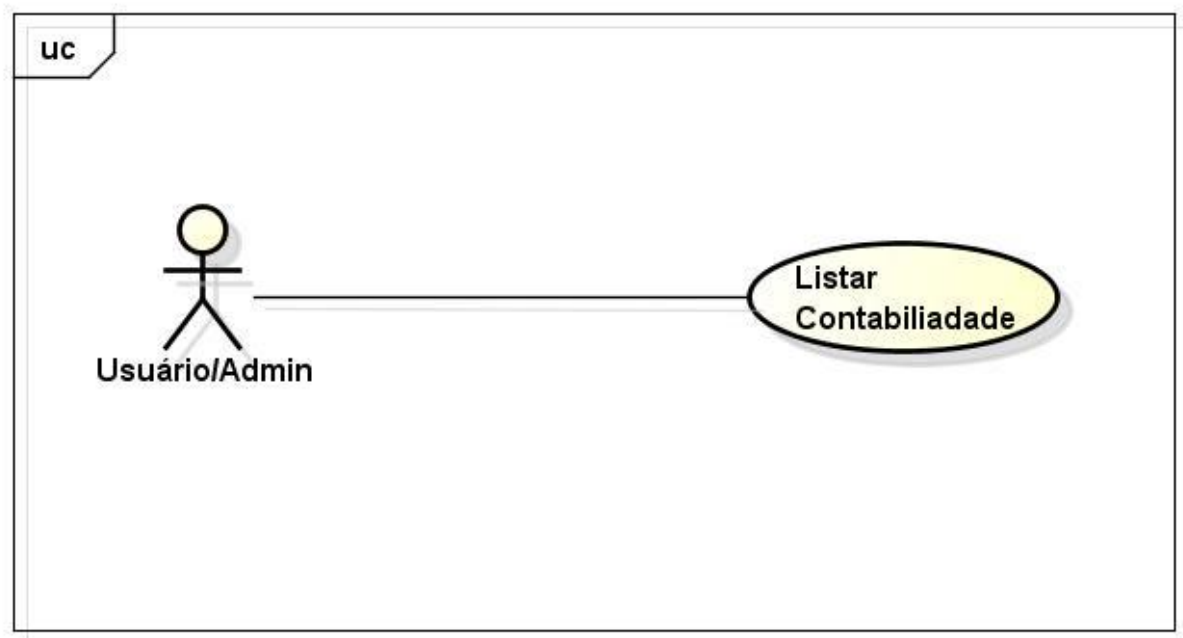


Figura 20- UC Listar Contabilidade

1- Finalidade/Objetivo

- Permite ao ator ver todos os movimentos de entrada e saída da empresa.

2- Ator

- Usuário
- Administrador

3- Precondições

- O ator necessita fazer o login no sistema e ter feito vendas (contratos ativos/Inativos), e ter lançado despesas.

4- Fluxo Principal

- O ator seleciona o item contabilidade, em seguida o item entrada/saída.
- O sistema mostra todos lucros e gastos.

5- Fluxo Alternativo

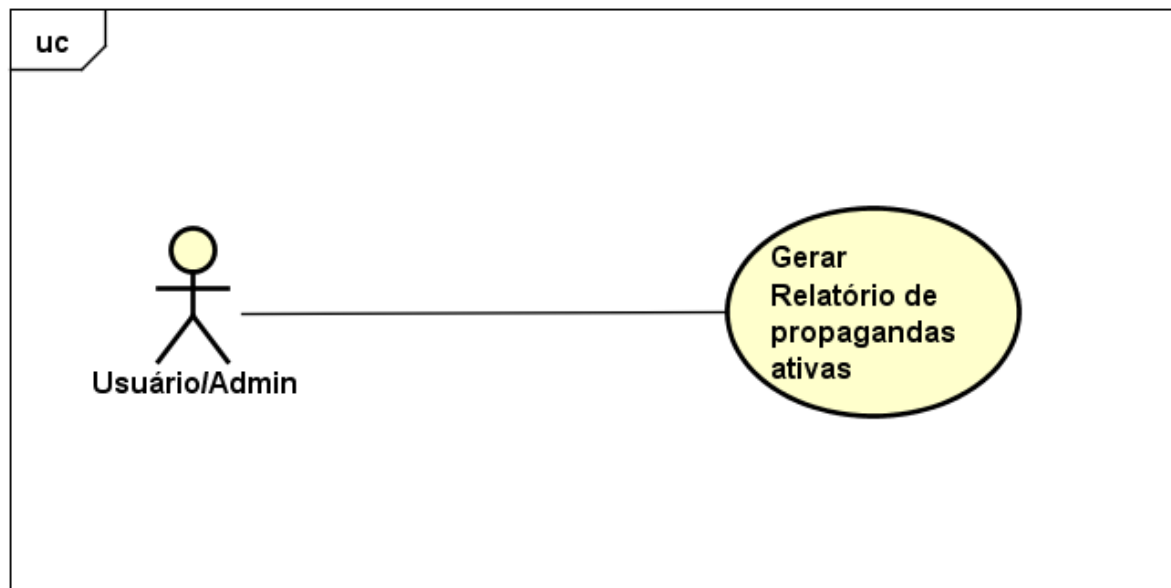
- O ator seleciona o item gerar pdf.
- O sistema gera um arquivo pdf.

- O ator seleciona o item cancelar.
- O sistema volta para a tela inicial.

6- Casos de Teste

- Não a casos de teste.

3.3.2.19 – UC Gerar Relatórios de Propagandas Ativas



powered by Astah

Figura 21 - Gerar Relatórios de Propagandas Ativas

1- Finalidade/Objetivo

- Permite ao ator gerar relatório das propagandas ativas.

2- Ator

- Usuário
- Administrador

3- Precondições

- O ator necessita fazer o login no sistema e ter clientes cadastrados, contratos cadastrados, lançamentos realizados.

4- Fluxo Principal

- O ator seleciona o item relatórios.
- O sistema gera o relatório em pdf e mostra ao ator.

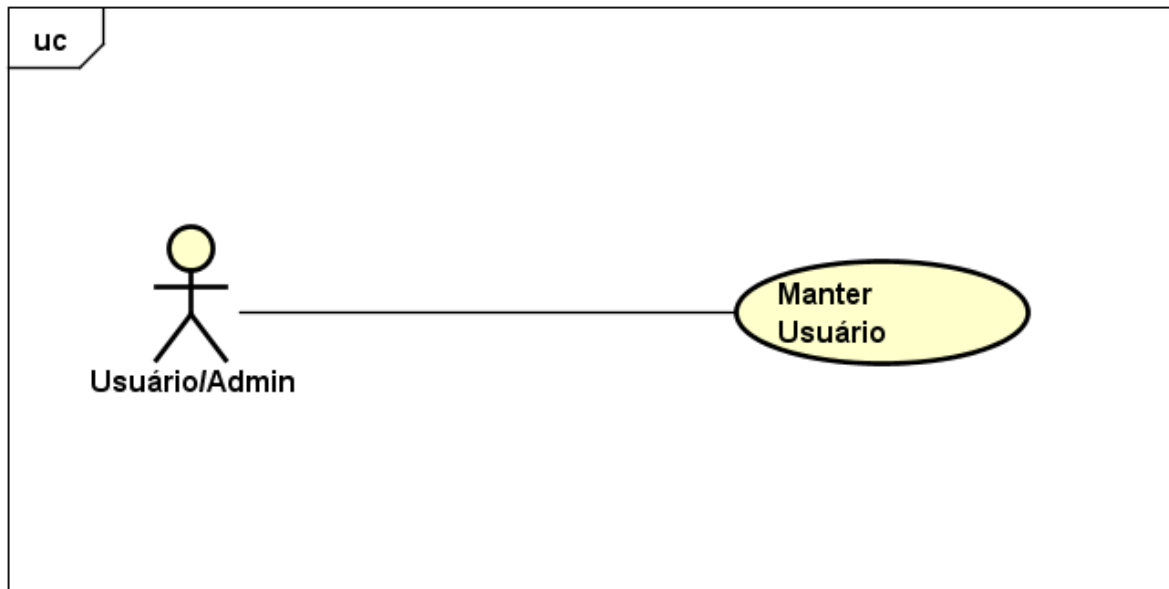
5- Fluxo Alternativo

- O ator seleciona o item cancelar.
- O sistema volta para a tela inicial.

6- Casos de Teste

- Não a casos de teste.

3.3.2.20 – UC Manter Usuários



powered by Astah

Figura 22 - Manter Usuário

1- Finalidade/Objetivo

- Permite que o Administrador cadastre, edite e exclua um usuário do sistema.

2- Ator

- Administrador

3- Precondições

- O administrador necessita fazer o login no sistema e ter um ou mais usuários cadastrados.

4- Fluxo Principal

- O administrador seleciona o item cadastro, em seguida o item usuário.
- O sistema mostra os usuários já cadastrados.
- O administrador seleciona o usuário que deseja excluir e seleciona o botão excluir.

- O sistema confirma com o administrador.

5- Fluxo Alternativo

- O administrador cadastra usuário.
- O administrador seleciona o item cancelar.
- O sistema volta para a tela inicial.

6- Casos de Teste

- O sistema verifica se o administrador não está tentando excluir o seu próprio cadastro.

3.3.2.21 – UC Logar Usuários

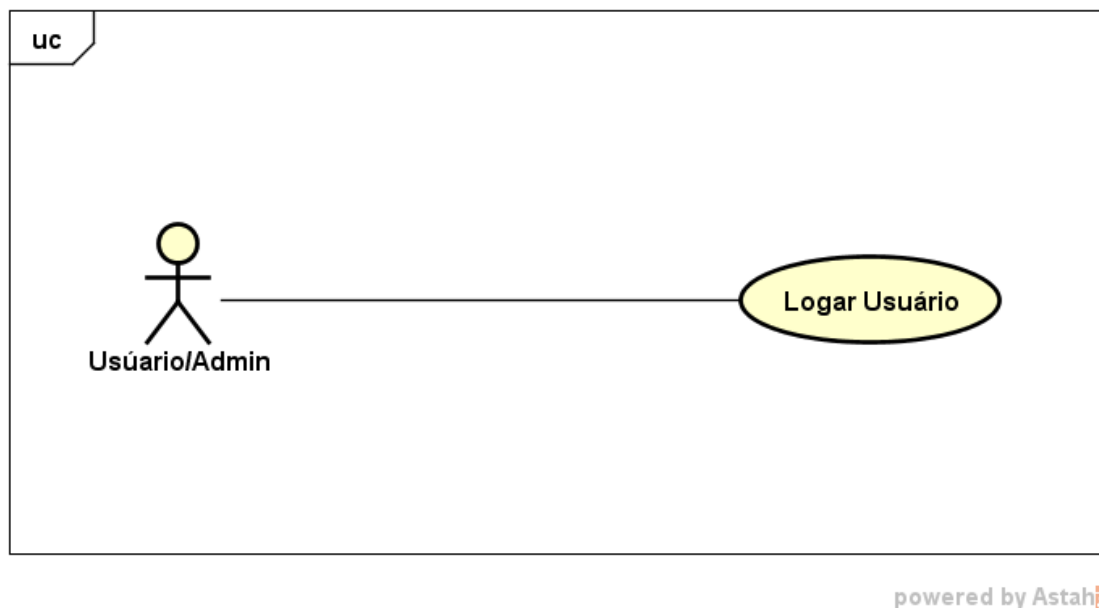


Figura 23 - UC Logar Usuário

1- Finalidade/Objetivo

- Permite que somente pessoas cadastradas no sistema possam utiliza-lo.

2- Ator

- Administrador e usuários.

3- Precondições

- Basta estar cadastrado como usuário no sistema.

4- Fluxo Principal

- O usuário/administrador abre o sistema.
- O sistema pede que preencha os campos obrigatório.
- O usuário/administrador fornece os dados corretos e seleciona o botão entrar.
- O sistema verifica de os dados fornecidos estão corretos e cadastrados.

5- Fluxo Alternativo

- O usuário/administrador pode cancelar e finalizar o sistema.

6- Casos de Teste

- O sistema verifica se os dados fornecidos são os que estão cadastrados.

- O sistema verifica se os campos não estão em branco.

3.3.2.22 – UC Deslogar Usuários

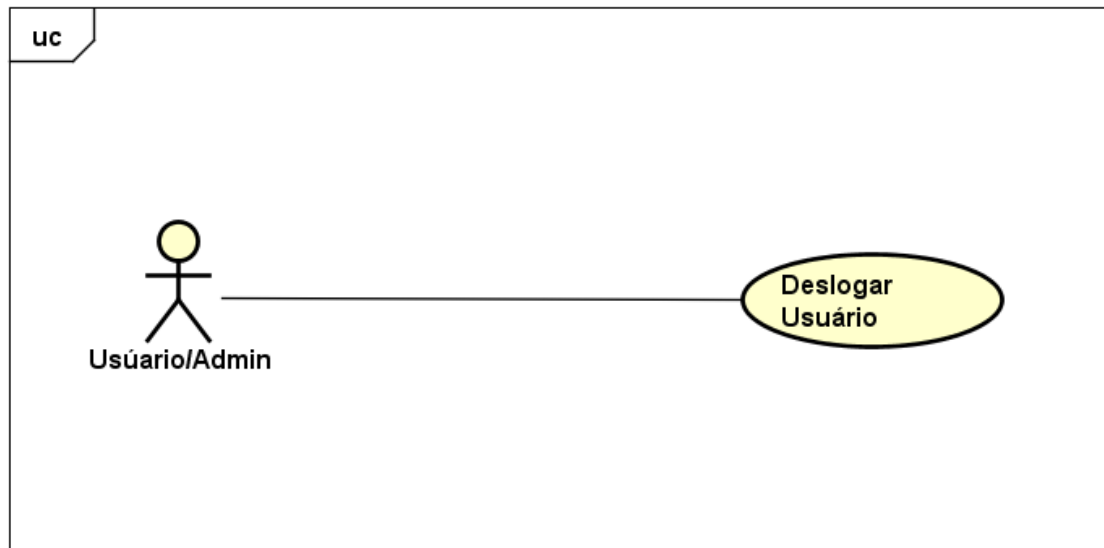


Figura 24 - Deslogar Usuário

1- Finalidade/Objetivo

- Permite que os usuários/administrador logado no sistema possa sair ou trocar de usuário.

2- Ator

- Administrador e usuários.

3- Precondições

- Basta estar logado no sistema.

4- Fluxo Principal

- O usuário/administrador seleciona o botão sair.
- O sistema volta para a tela de login.

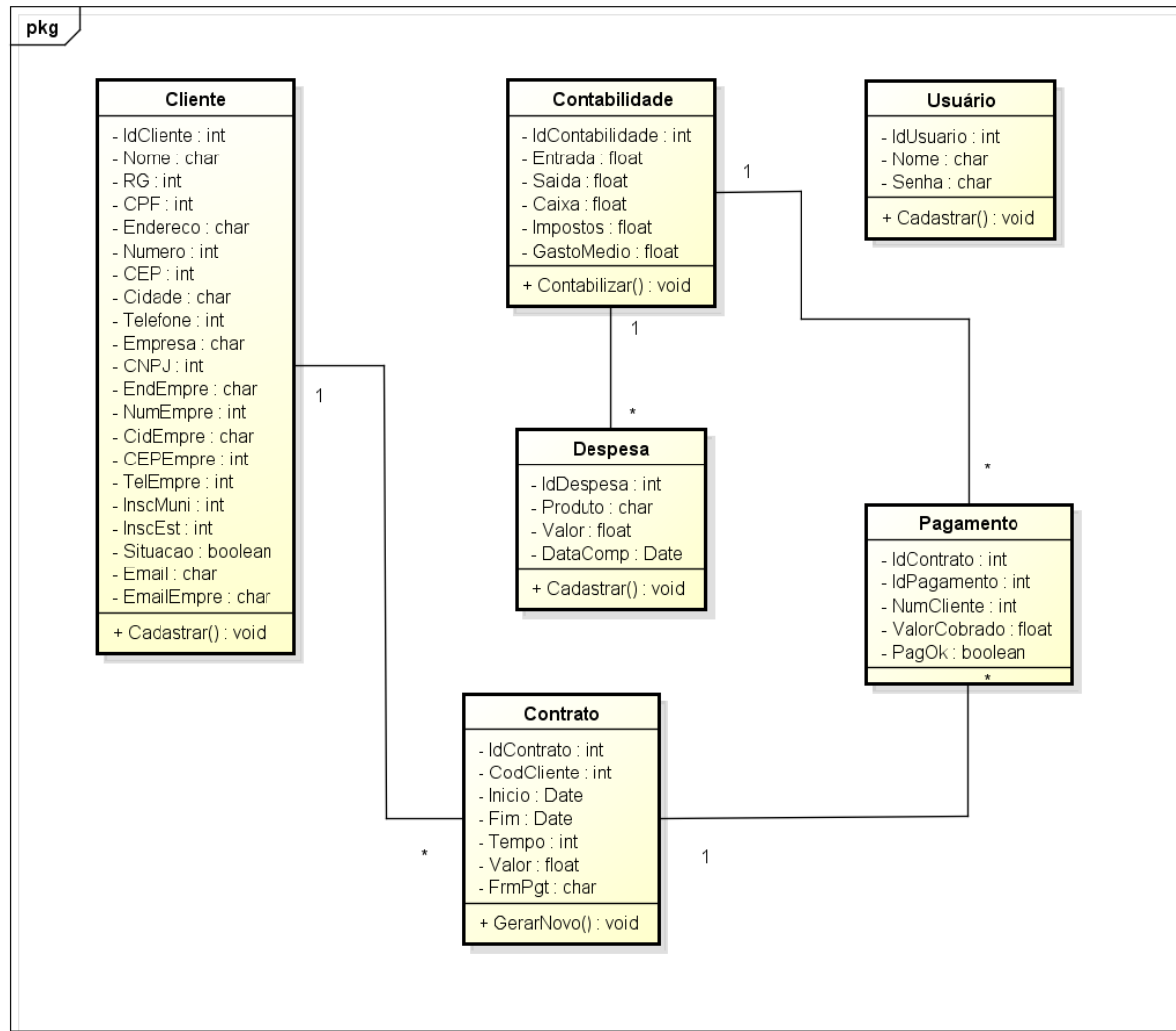
5- Fluxo Alternativo

- Não há fluxo alternativo.

6- Casos de Teste

- Não há testes.

3.3.3- Diagrama de Classe



powered by Astah

Figura 25 - Diagrama de Classe

3.3.4- Diagrama de Sequência.

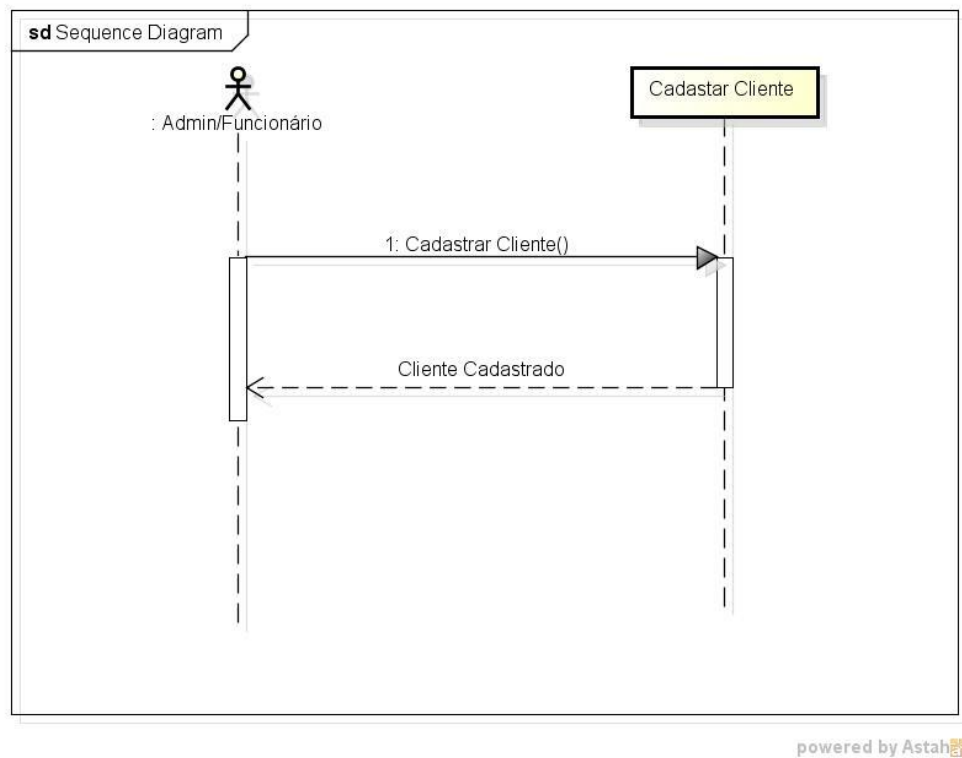


Figura 26 - Sequence Diagram - Cadastrar Cliente

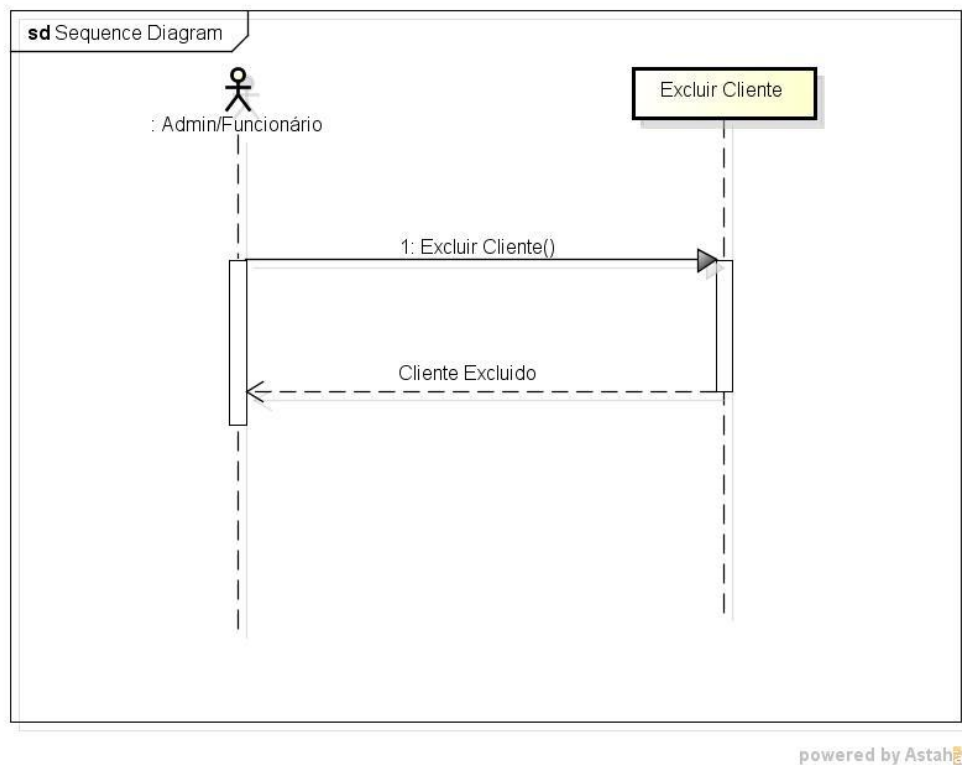


Figura 27 - Sequence Diagram - Excluir Cliente

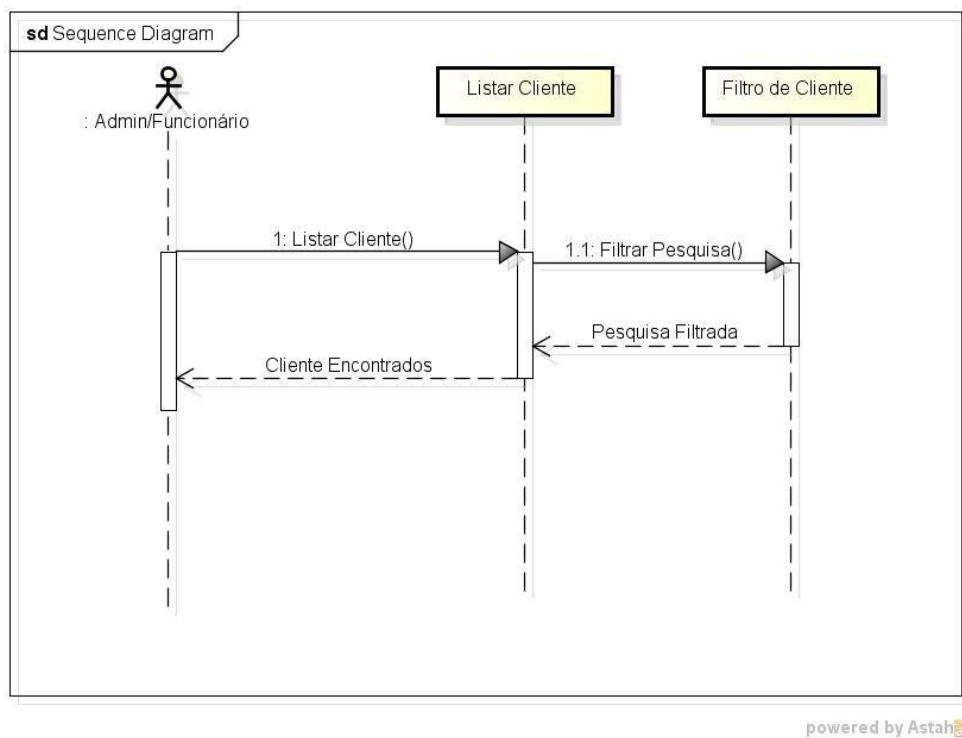


Figura 28 - Sequence Diagram - Listar Cliente

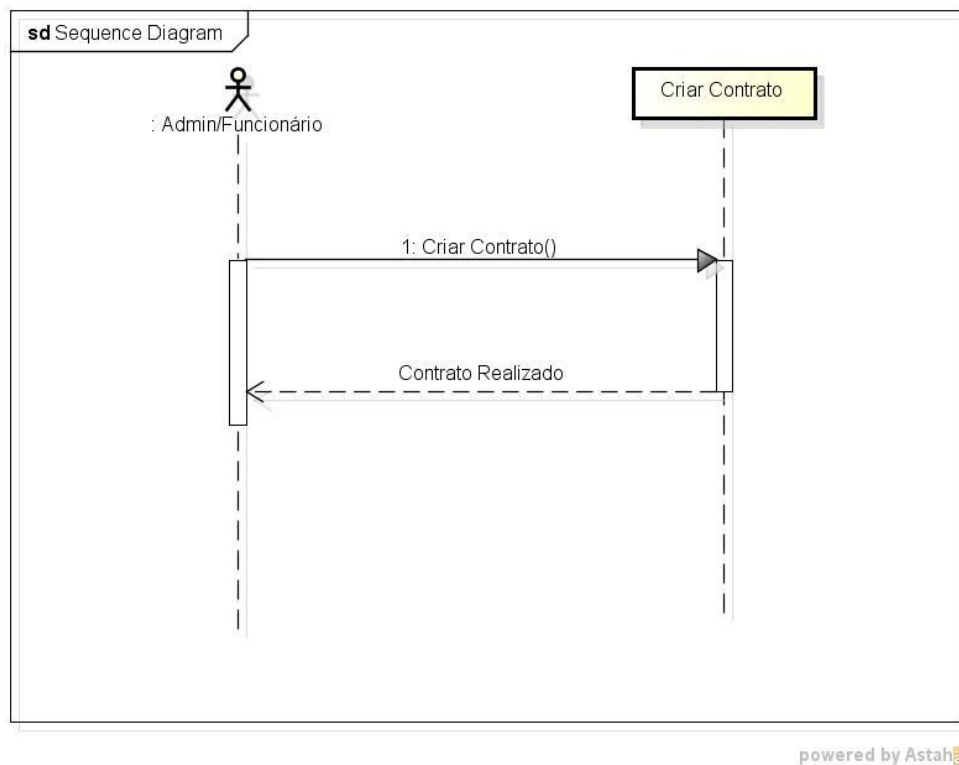


Figura 29 - Sequence Diagram - Criar Contrato

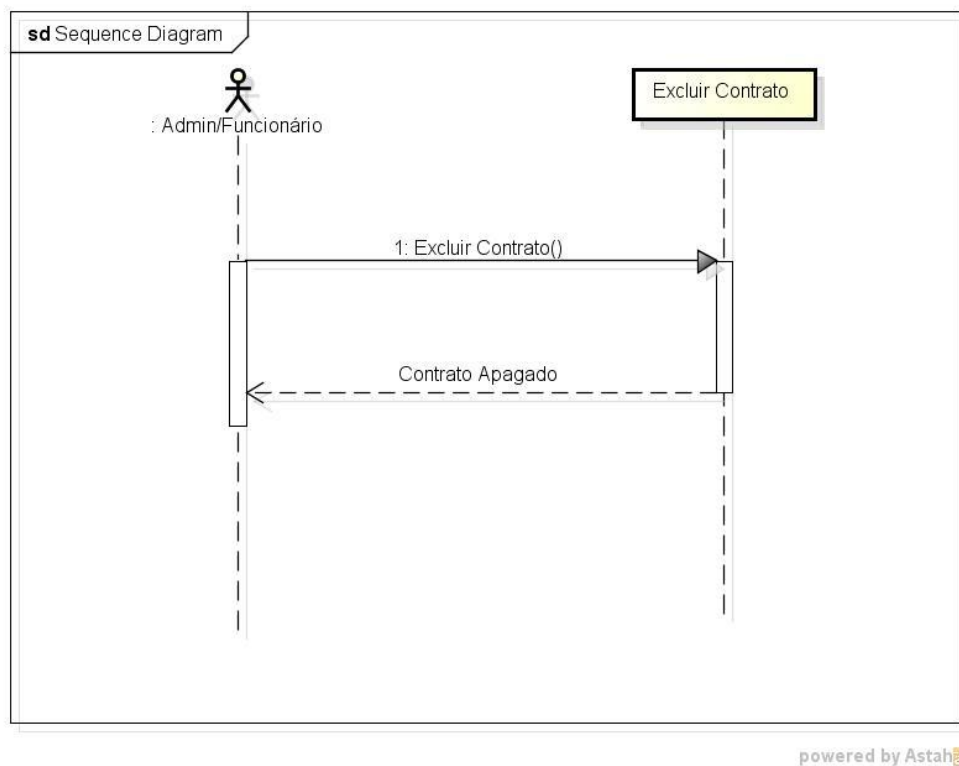


Figura 30 - Sequence Diagram - Excluir Contrato

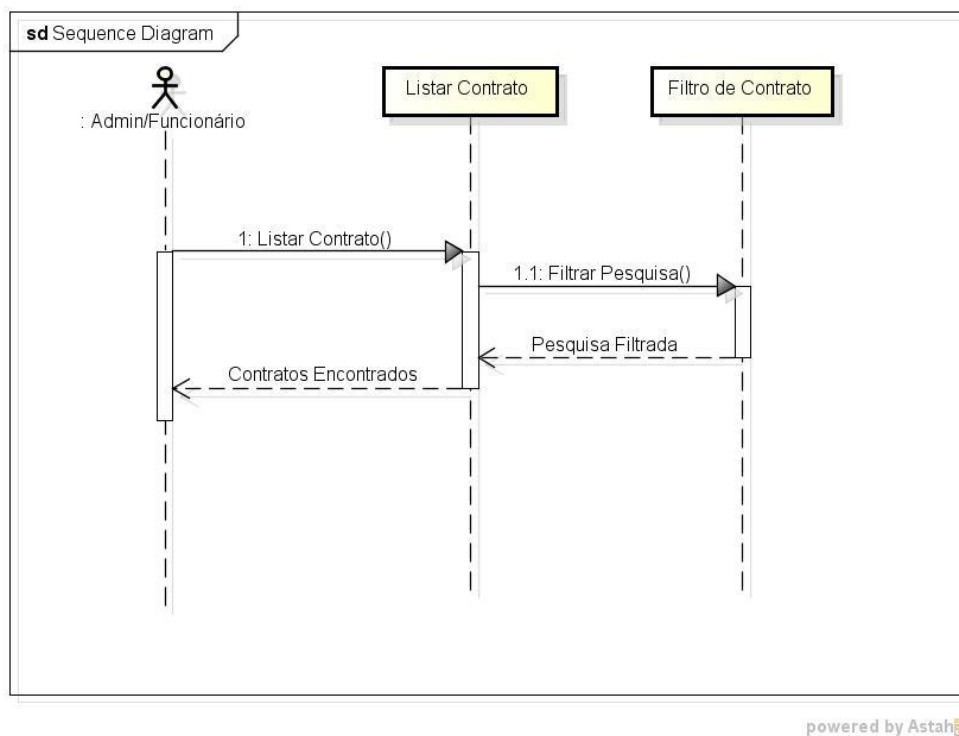


Figura 31 - Sequence Diagram - Listar Contrato

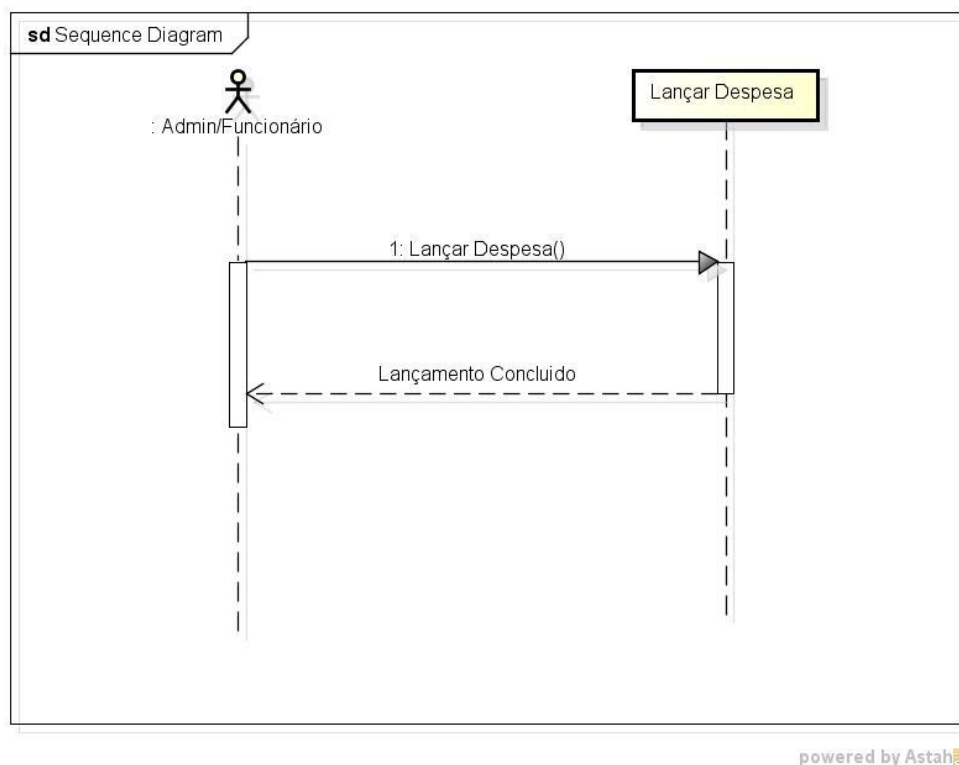


Figura 32 - Sequence Diagram - Lançar Despesa

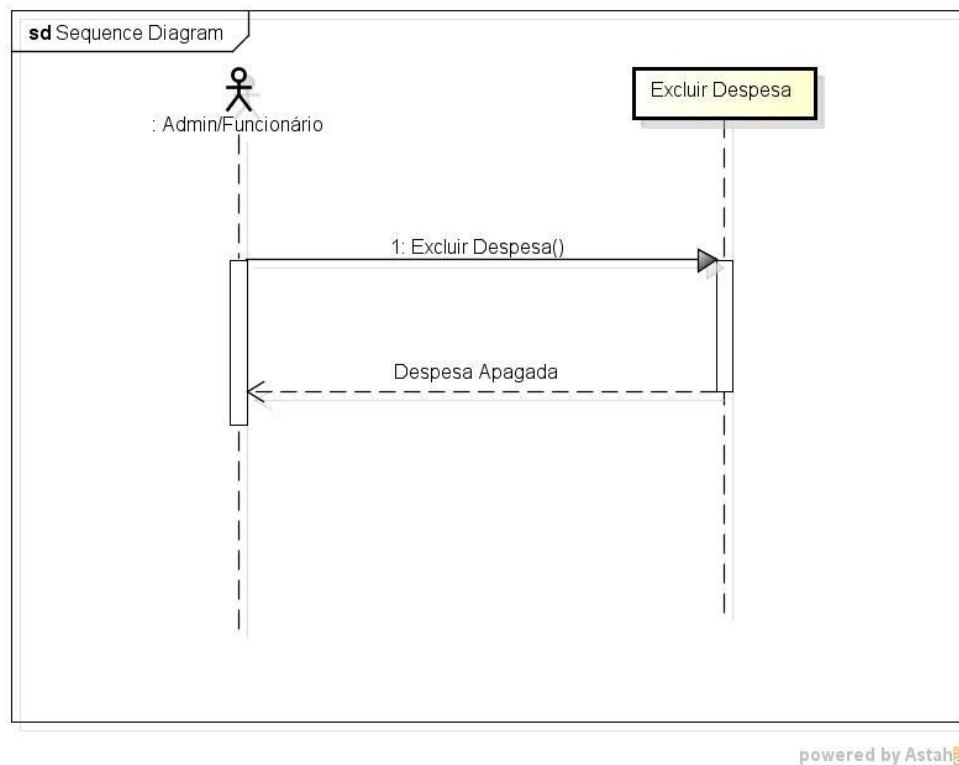


Figura 33 - Sequence Diagram - Excluir Despesa

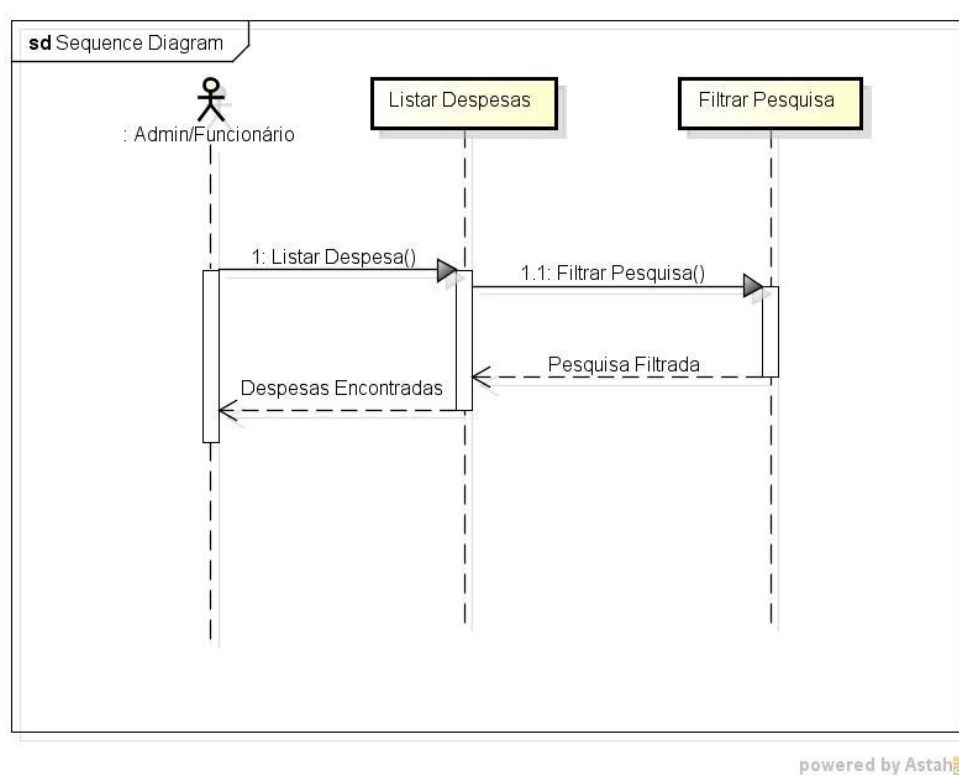


Figura 34 - Sequence Diagram - Listar Despesas

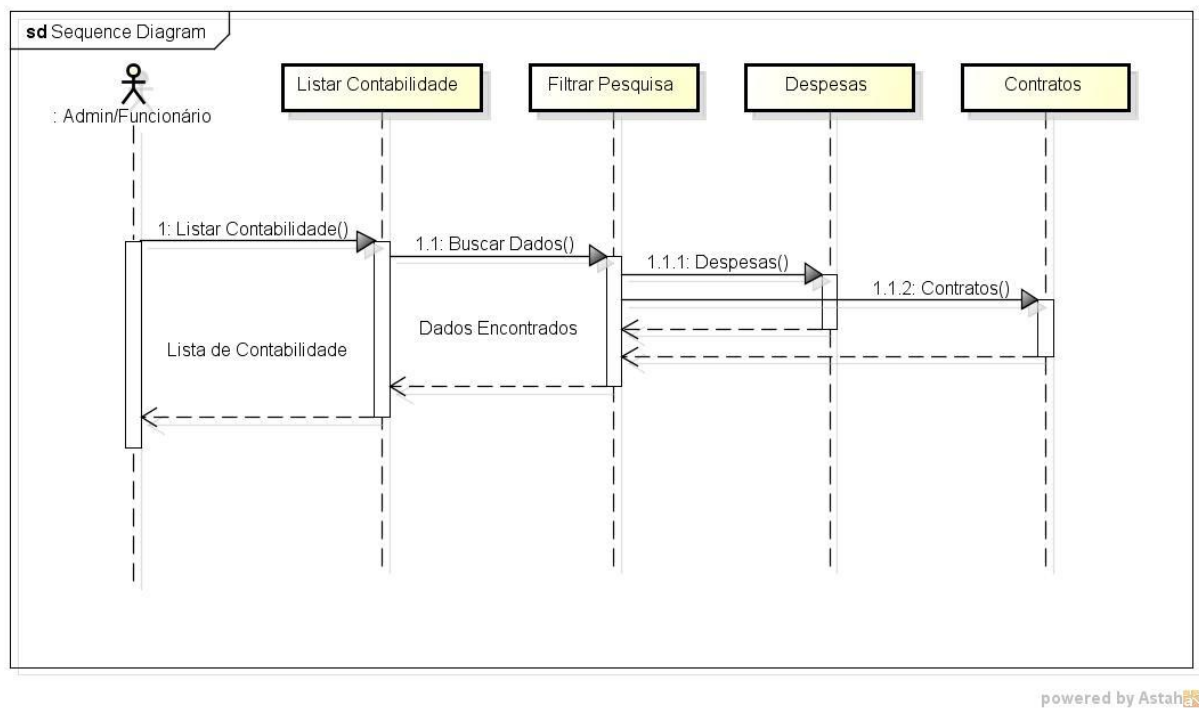


Figura 35 - Sequence Diagram - Listar Contabilidade

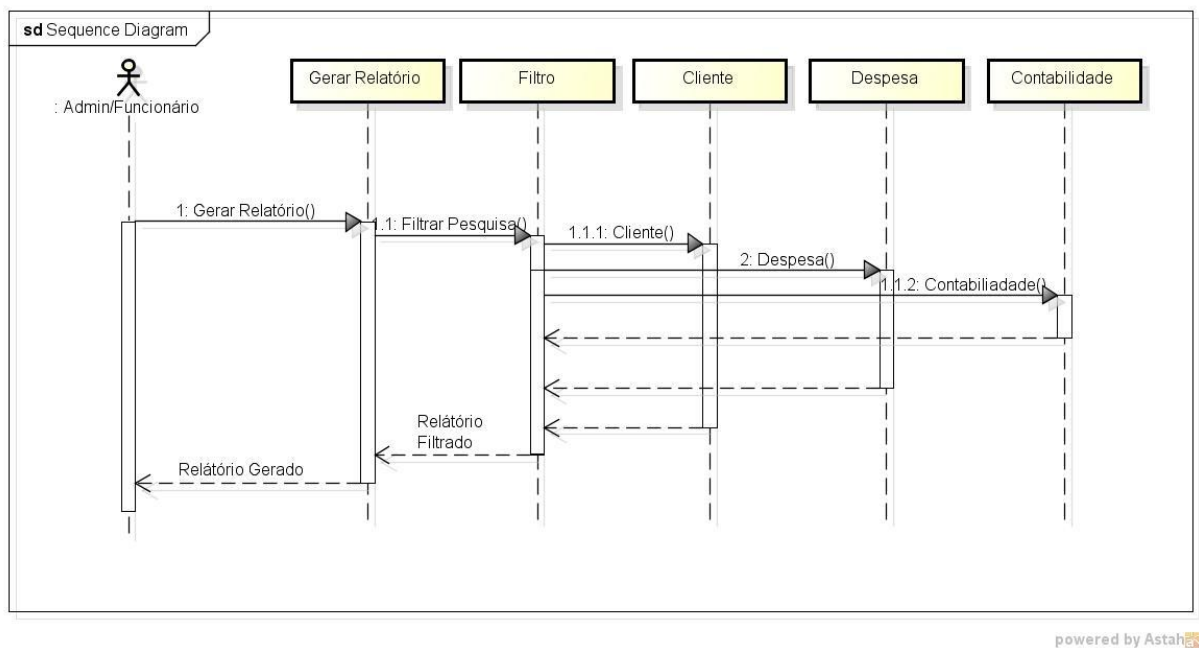


Figura 36 - Sequence Diagram - Gerar Relatório

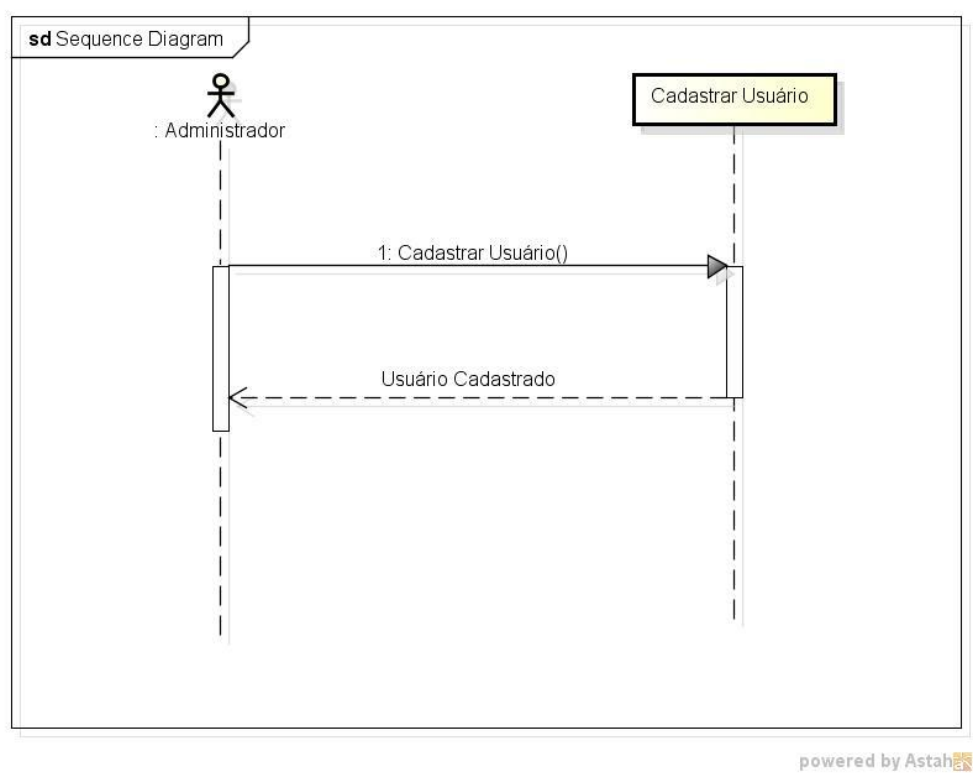


Figura 37 - Sequence Diagram - Cadastrar Usuário

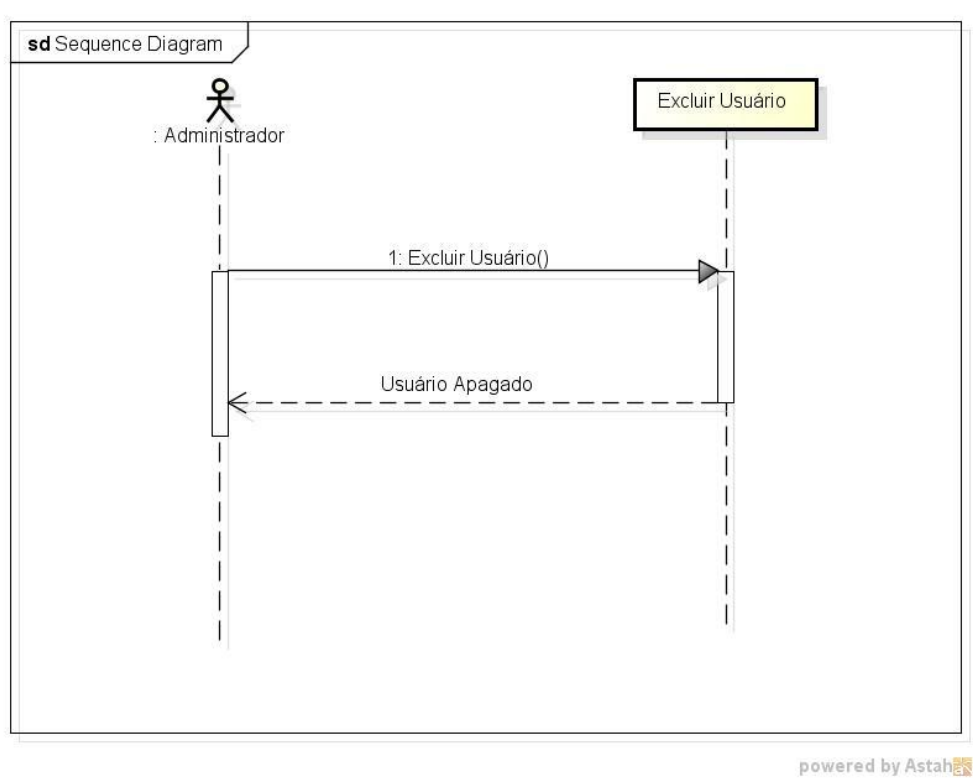


Figura 38 - Sequence Diagram - Excluir Usuário

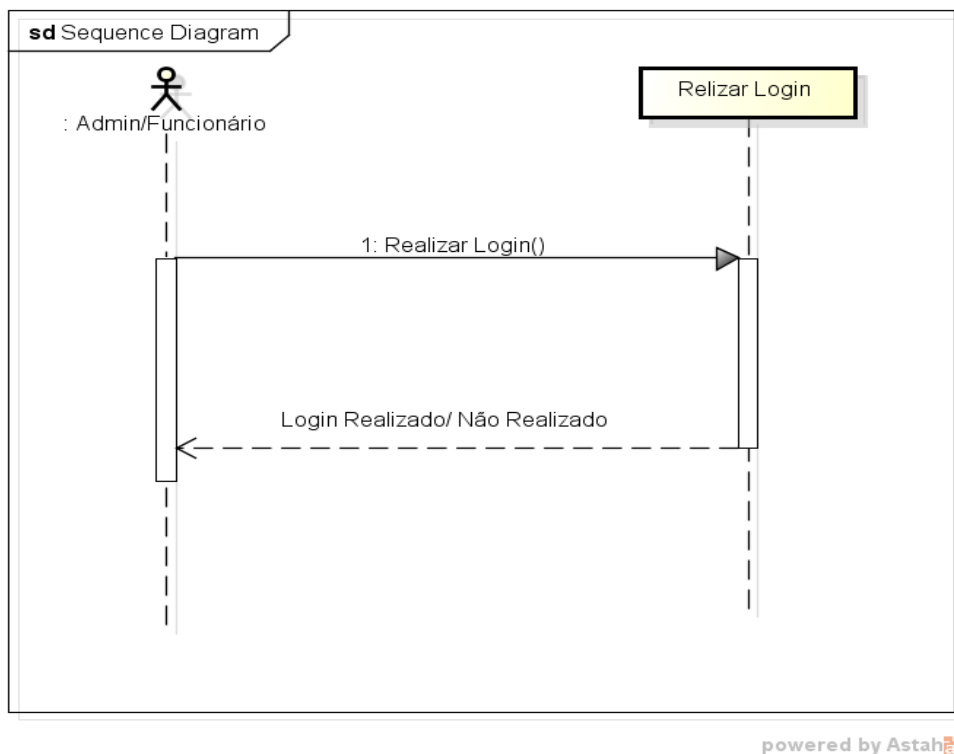


Figura 39 - Sequence Diagram - Realizar Login

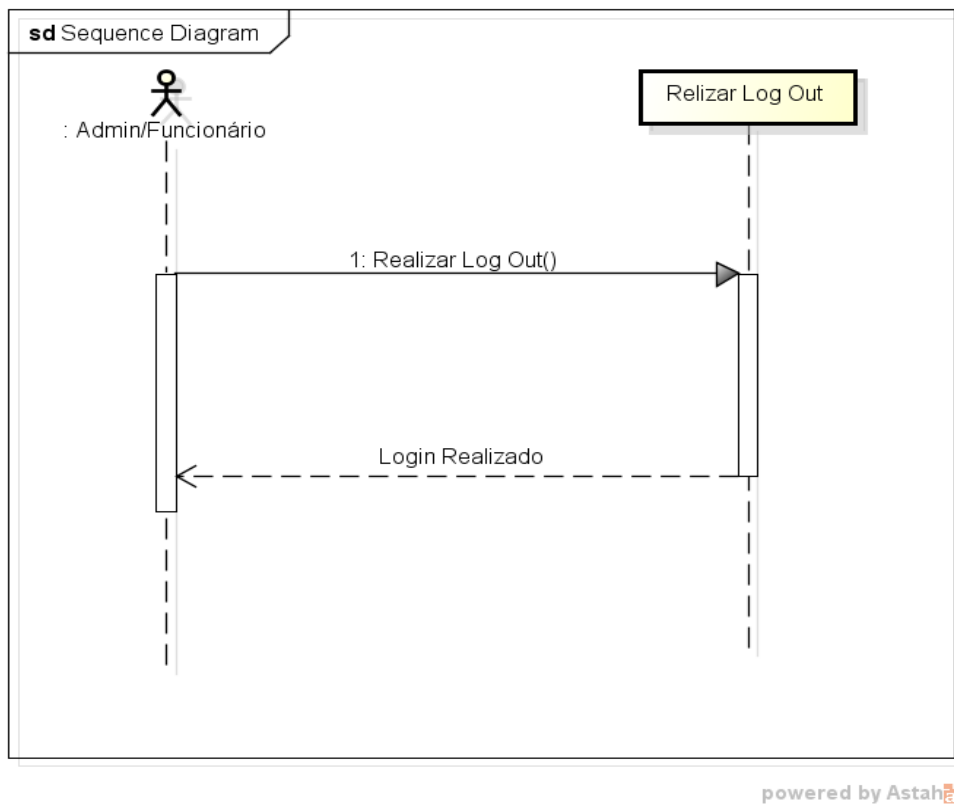


Figura 40 - Sequence Diagram - Realizar Log Out

3.3.5- Entidade Relacionamento.

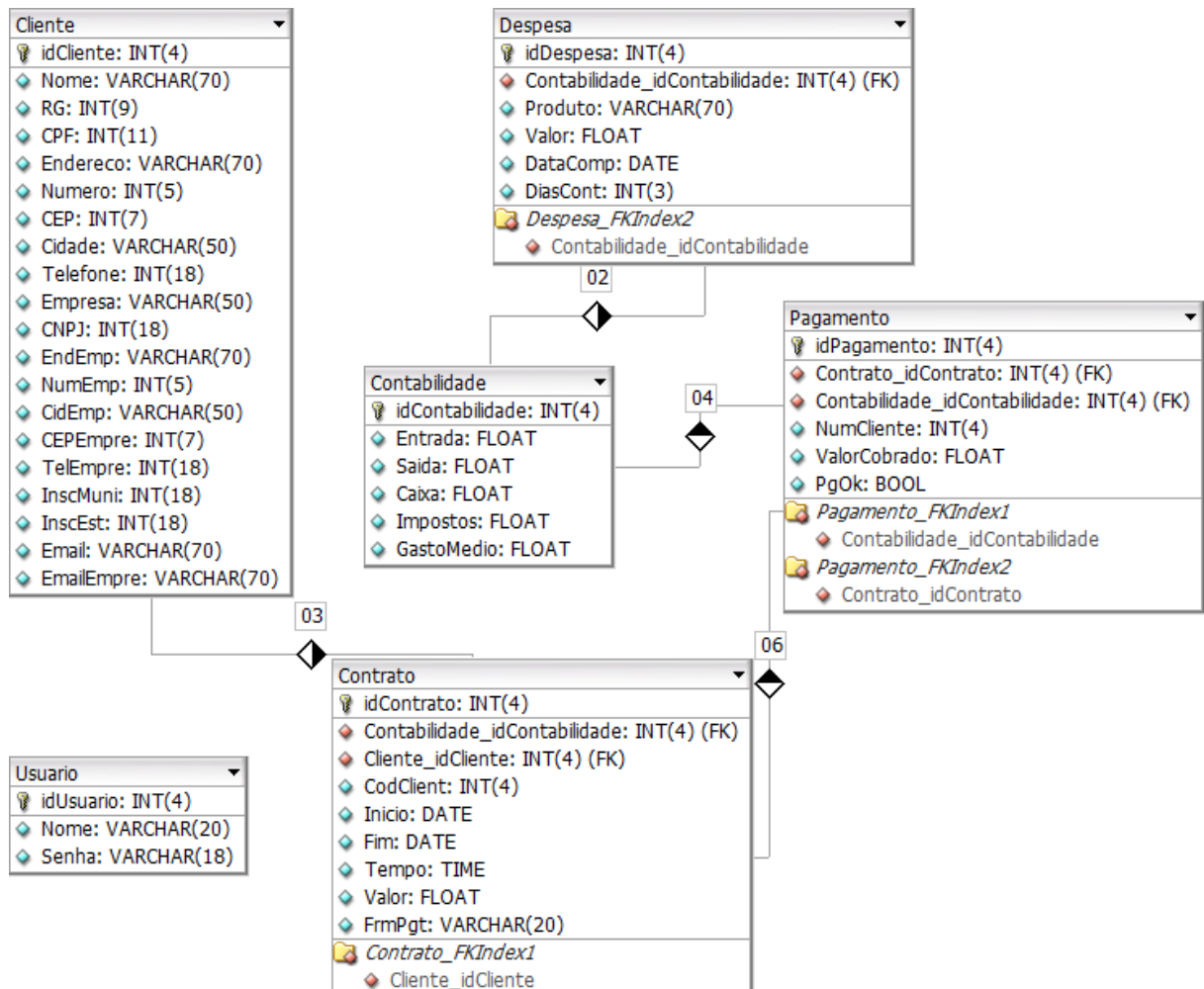


Figura 41 - Entidade Relacionamento

3.3.6- Estrutura Analítica de Projeto.

Em Gerência de projetos, uma Estrutura Analítica de Projetos (EAP), do Inglês, Work breakdown structure (WBS) é uma ferramenta de decomposição do trabalho do projeto em partes manejáveis. É estruturada em árvore exaustiva, hierárquica (de mais geral para mais específica) orientada às entregas (*deliverables*) que precisam ser feitas para completar um projeto.

Desconhecido.

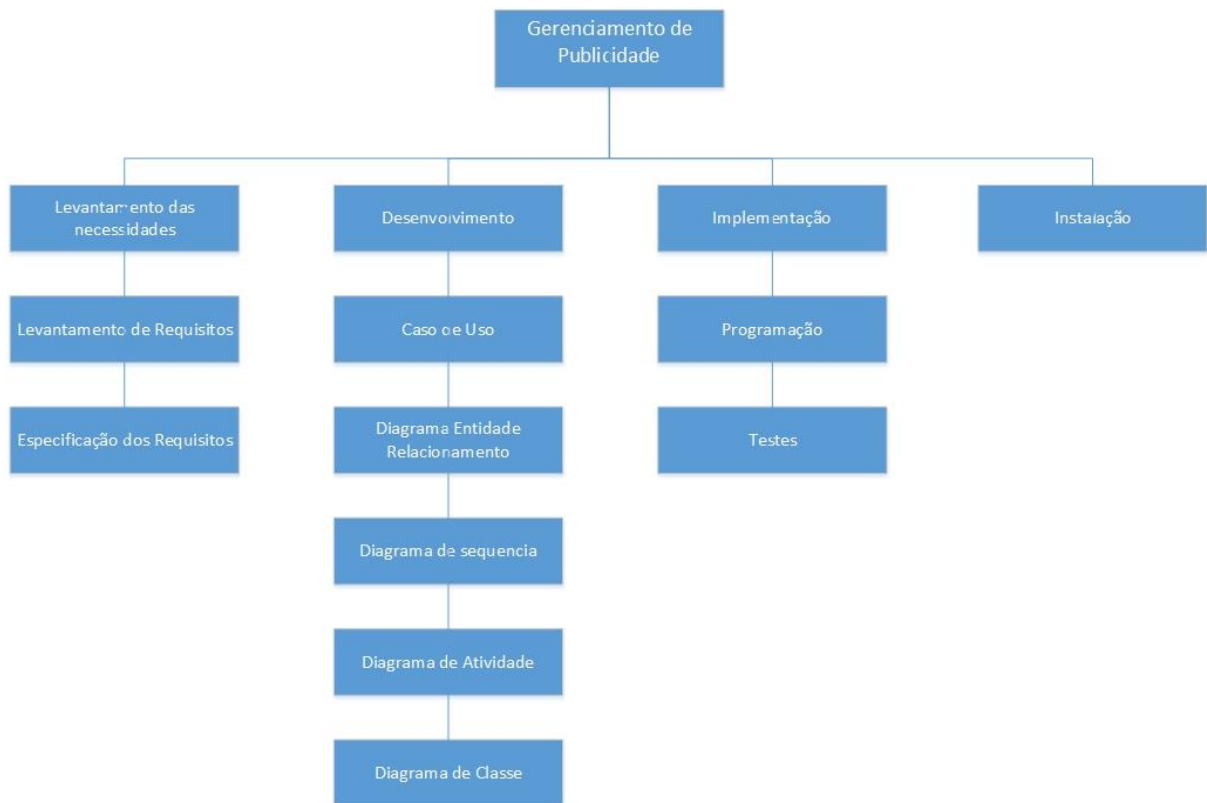


Figura 42 - EAP

3.3.7- Sequenciamento de Atividade.

De acordo com PMBOK (2000). Capítulo 06 - Gerencia de Projeto

O sequenciamento da atividade envolve identificar e documentar os relacionamentos lógicos entre as atividades. As atividades devem ser sequenciadas corretamente para suportar o desenvolvimento de um cronograma realístico e alcançável. O sequenciamento pode ser feito com o auxílio de um computador (por exemplo, utilizando softwares de gerência de projeto) ou com técnicas manuais. As técnicas manuais são, geralmente, mais efetivas em projetos menores e em fases iniciais de projetos maiores quando poucos detalhes estão disponíveis. As técnicas manuais e automatizadas podem, também, ser utilizadas em conjunto.

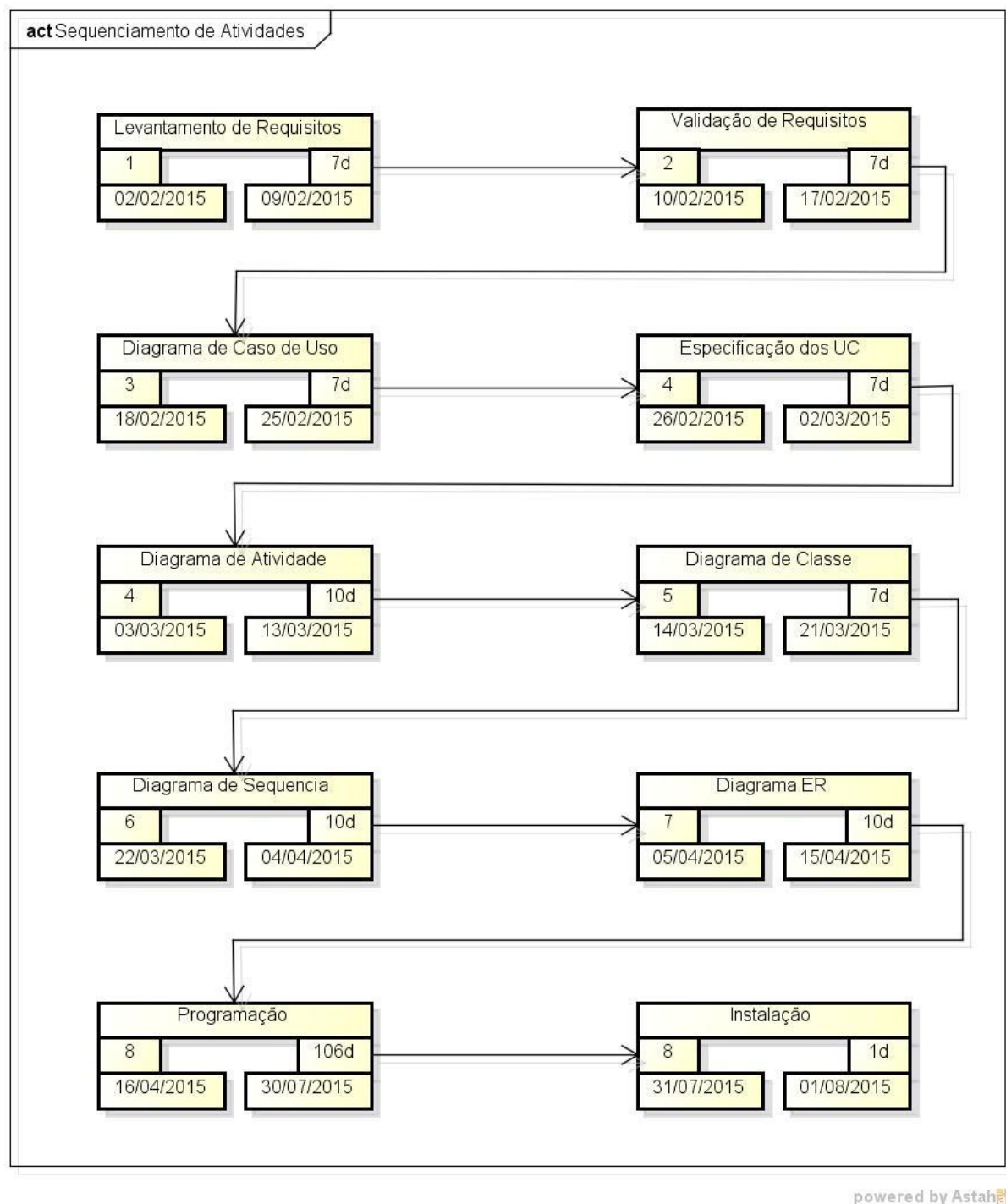


Figura 43 - Sequenciamento de Atividade

4- IMPLEMENTAÇÃO

Na implementação foi utilizado a linguagem de programação C#, no Visual Studio 2013. Também, foi utilizado o SQL Server 2012 para o banco de dados.

4.1- ORGANIZAÇÃO DO PROJETO

Para a melhor organização do projeto e desenvolvimento do sistema, foram desenvolvidas três camadas, como mostra a figura a seguir (Figura 35). Model, DAL, BLL.

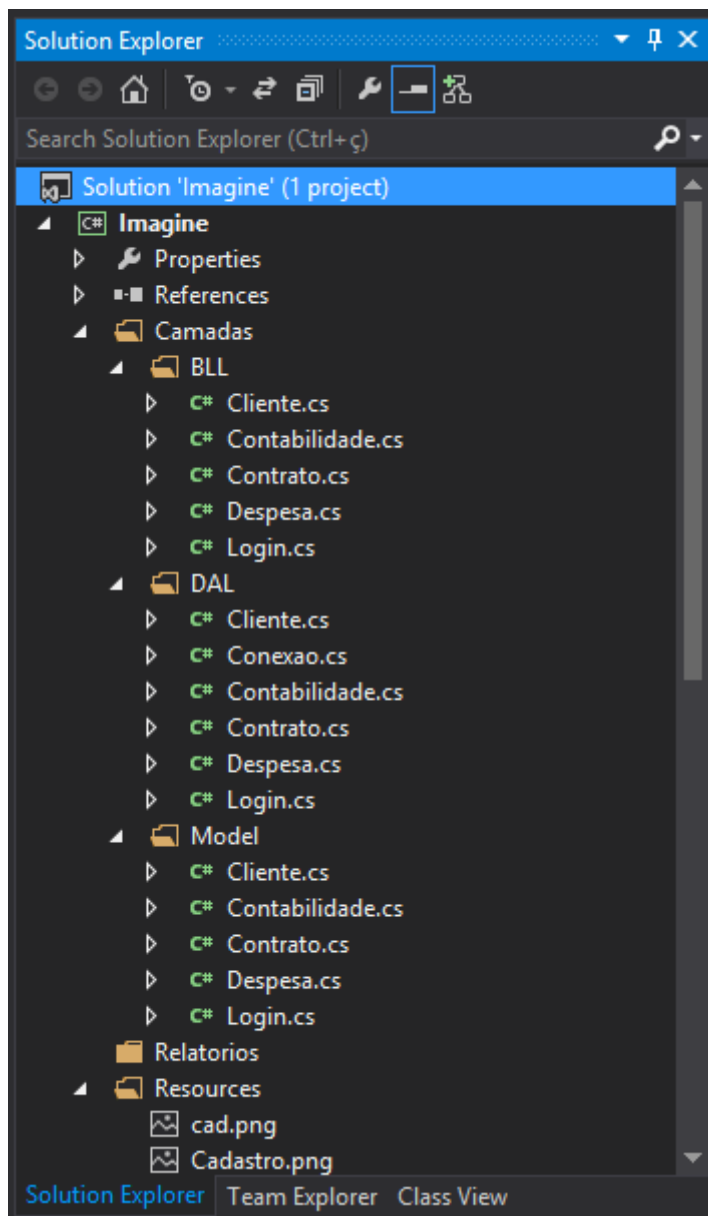


Figura 44 - Camadas

A Camada Model é onde estão criadas as entidades que modelam o sistema de dados, como mostra a figura a seguir (Figura 36).

```

namespace Imagine.Camadas.Model
{
    20 references
    class Login
    {
        private int _IdUsuario;
        private string _Nome;
        private string _Senha;

        2 references
        public int IdUsuario
        {
            get { return _IdUsuario; }
            set { _IdUsuario = value; }
        }

        6 references
        public string Nome
        {
            get { return _Nome; }
            set { _Nome = value; }
        }

        4 references
        public string Senha
        {
            get { return _Senha; }
            set { _Senha = value; }
        }
    }
}

```

Figura 45 – Model

A Camada DAL é onde estão criados os métodos que fazem relacionamentos com o banco de dados. Na camada DAL foi criado a classe conexão, a qual é responsável por mostrar o caminho, o endereço para o BD (Figura 37). Nas demais classes foram implementados os métodos de SELECT, UPDATE, INSERT, DELETE, que são os métodos de Incluir, Excluir, Editar, Consultar, com os parâmetros passados pela String de conexão, que está na classe de conexão (Figura 38, 39, 40, 41 e 42).

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Imagine.Camadas.DAL
{
    5 references
    class Conexao
    {
        5 references
        public static string getConexao()
        {
            return @"Data Source=.\sqlexpress;Initial Catalog=IMAGINE;Integrated Security=True;Pooling=False";
        }
    }
}

```

Figura 46 - Classe Conexão

```

class Login
{
    private string strCon = Conexao.getConexao();

    1 reference
    public Model.Login SelectById(int Id)
    {
        Model.Login login = new Model.Login();
        SqlConnection conexao = new SqlConnection(strCon);
        string sql = "Select * from Usuario where IdUsuario=@id";
        SqlCommand cmd = new SqlCommand(sql, conexao);
        cmd.Parameters.AddWithValue("@Id", Id);
        conexao.Open();
        SqlDataReader reader = cmd.ExecuteReader(CommandBehavior.CloseConnection);
        try
        {
            if (reader.Read())
            {
                login.IdUsuario = Convert.ToInt32(reader[0].ToString());
                login.Nome = reader["Nome"].ToString();
                login.Senha = reader["Senha"].ToString();
            }
        }
        catch
        {
            Console.WriteLine("Deu erro na execucao do select...");
        }
        finally
        {
            conexao.Close();
        }

        return login;
    }
}

```

Figura 47 – 1º SELECT

```
public List<Model.Login> Select()
{
    List<Model.Login> lstLogin = new List<Model.Login>();
    SqlConnection conexao = new SqlConnection(strCon);
    string sql = "Select * from Usuario;";
    SqlCommand cmd = new SqlCommand(sql, conexao);
    conexao.Open();
    SqlDataReader reader = cmd.ExecuteReader(CommandBehavior.CloseConnection);
    try
    {
        while (reader.Read())
        {
            Model.Login login = new Model.Login();
            login.Nome = reader["Nome"].ToString();
            // login.Senha = reader["Senha"].ToString();
            lstLogin.Add(login);
        }
    }
    catch
    {
        Console.WriteLine("Deu erro na execucao do select...");
    }
    finally
    {
        conexao.Close();
    }

    return lstLogin;
}
```

Figura 48 - 2º SELECT

```

public void Insert(Model.Login login)
{
    SqlConnection conexao = new SqlConnection(strCon);
    string sql = "Insert into Usuario values (@Nome, @Senha);";
    SqlCommand cmd = new SqlCommand(sql, conexao);
    cmd.Parameters.AddWithValue("@Nome", login.Nome);
    cmd.Parameters.AddWithValue("@Senha", login.Senha);
    conexao.Open();
    try
    {
        cmd.ExecuteNonQuery();
    }
    catch
    {
        Console.WriteLine("Deu erro insercao de usuario....");
    }
    finally
    {
        conexao.Close();
    }
}

```

Figura 49 - INCLUDE

```

public void Update(Model.Login login)
{
    SqlConnection conexao = new SqlConnection(strCon);
    string sql = "Update Usuario set nome=@Nome where Senha=@Senha, ";
    sql += "IdUsuario=@IdUsuario";
    SqlCommand cmd = new SqlCommand(sql, conexao);
    //cmd.Parameters.AddWithValue("@IdUsuario", login.IdUsuario);
    cmd.Parameters.AddWithValue("@Nome", login.Nome);
    cmd.Parameters.AddWithValue("@Senha", login.Senha);
    conexao.Open();
    try
    {
        cmd.ExecuteNonQuery();
    }
    catch
    {
        Console.WriteLine("Deu erro na atualizaçao de Usuario...");
    }
    finally
    {
        conexao.Close();
    }
}

```

Figura 50 - UPDATE

```
public void Delete(Model.Login login)
{
    SqlConnection conexao = new SqlConnection(strCon);
    string sql = "Delete from Usuario where IdUsuario=@IdUsuario;";
    SqlCommand cmd = new SqlCommand(sql, conexao);
    cmd.Parameters.AddWithValue("@IdUsuario", login.IdUsuario);
    conexao.Open();
    try
    {
        cmd.ExecuteNonQuery();
    }
    catch
    {
        Console.WriteLine("Deu erro na remoção....");
    }
    finally
    {
        conexao.Close();
    }
}
```

Figura 51 – DELETE

A Camada BLL é onde está criado a interface da classe, a qual se chama a camada DAL (Camada de Apresentação), para que o que foi desenvolvido nas outras classes da camada sejam acessados. Isso é feito com a orientação a objetos, que nada mais são do que ponteiro que aponta para a classe em questão, como mostra na imagem (Figura 43). As classes referenciadas estão demonstradas na imagem (Figura 44).

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Imagine.Camadas.BLL
{
    4 references
    class Login
    {
        0 references
        public Model.Login SelectById(int id)
        {
            DAL.Login dalLogin = new DAL.Login();
            return dalLogin.SelectById(id);
        }

        2 references
        public List<Model.Login> Select()
        {
            DAL.Login dalLogin = new DAL.Login();
            return dalLogin.Select();
        }

        1 reference
        public void Insert(Model.Login login)
        {
            DAL.Login dalLogin = new DAL.Login();
            dalLogin.Insert(login);
        }

        1 reference
        public void Update(Model.Login login)
        {
            DAL.Login dalLogin = new DAL.Login();
            dalLogin.Update(login);
        }

        1 reference
        public void Delete(Model.Login login)
        {
            DAL.Login dalLogin = new DAL.Login();
            dalLogin.Delete(login);
        }
    }
}
```

Figura 52 – BLL

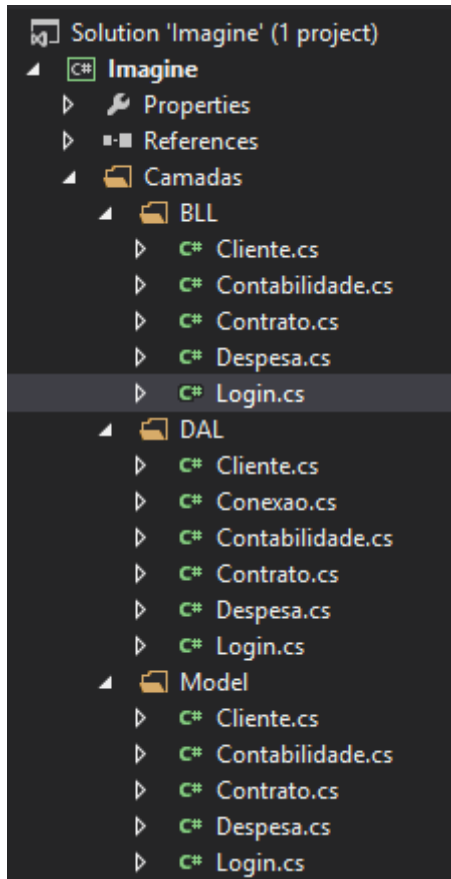


Figura 53 - Casses Imagine

4.2 – Interfaces do Projeto

Para a utilização do sistema necessita-se fazer o login, como demonstra a imagem, com os campos de Login e Senha, sendo possível somente entrar no sistema ou cancelar a execução (Figura 46).



Figura 54 - Tela Login

A segunda e principal tela é a de menu, ao clicar em entrar e conferir os dados abre-se a tela menu (Figura 46).

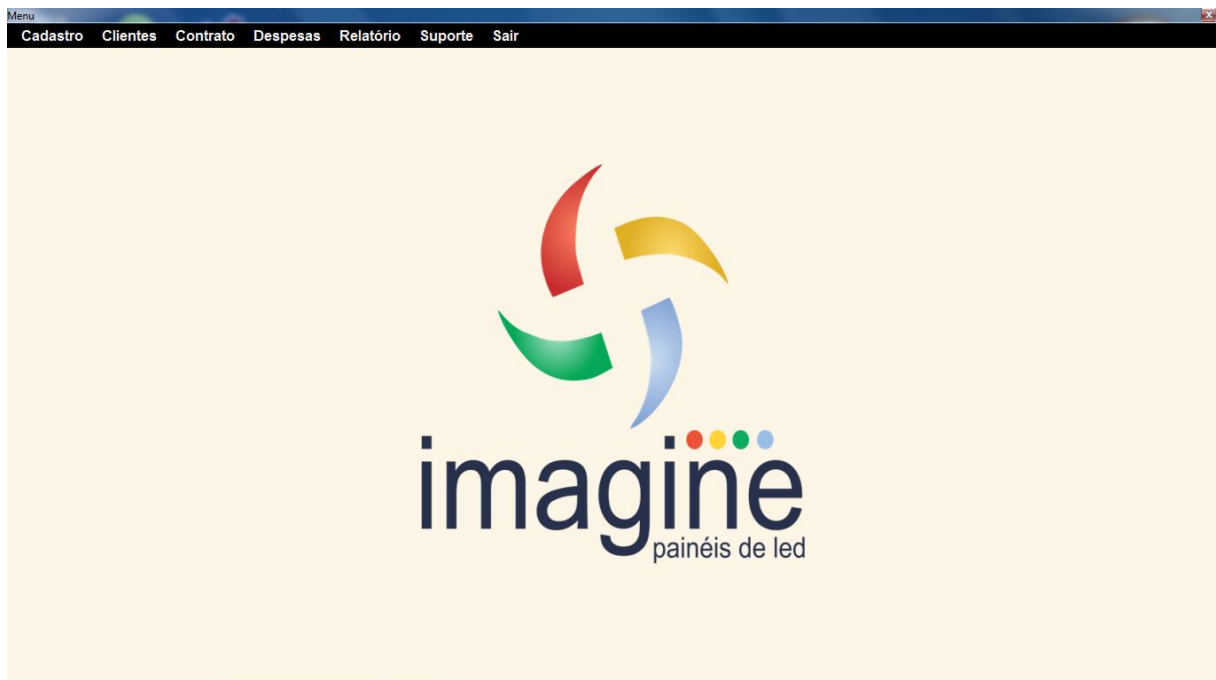


Figura 55 - Tela Inicial

Tela de Cadastro de Clientes, onde se coloca todos os dados do cliente e da empresa (Figura 48).

Figura 56 - Tela Cadastro Cliente

Tela Cadastro de usuários, onde o administrador inclui, visualiza, edita, exclui o usuário. (Figura 49)

Figura 57 - Tela Cadastro Usuário

Tela de Clientes, Lista todos os clientes, de forma que se pode pesquisar de acordo com a necessidade. (Figura 50)

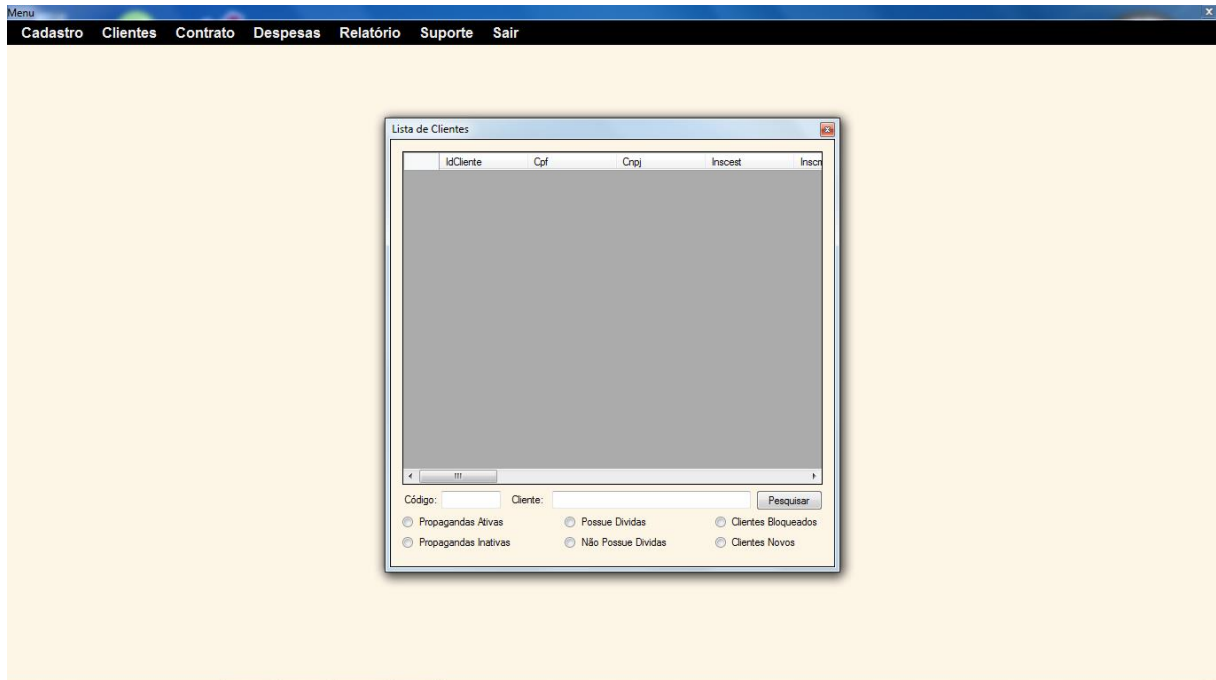


Figura 58 - Tela Lista de Clientes

Tela de Contratos, Lista todos os contratos de clientes, de forma que se pode pesquisar de acordo com a necessidade que possa surgir. (Figura 51)

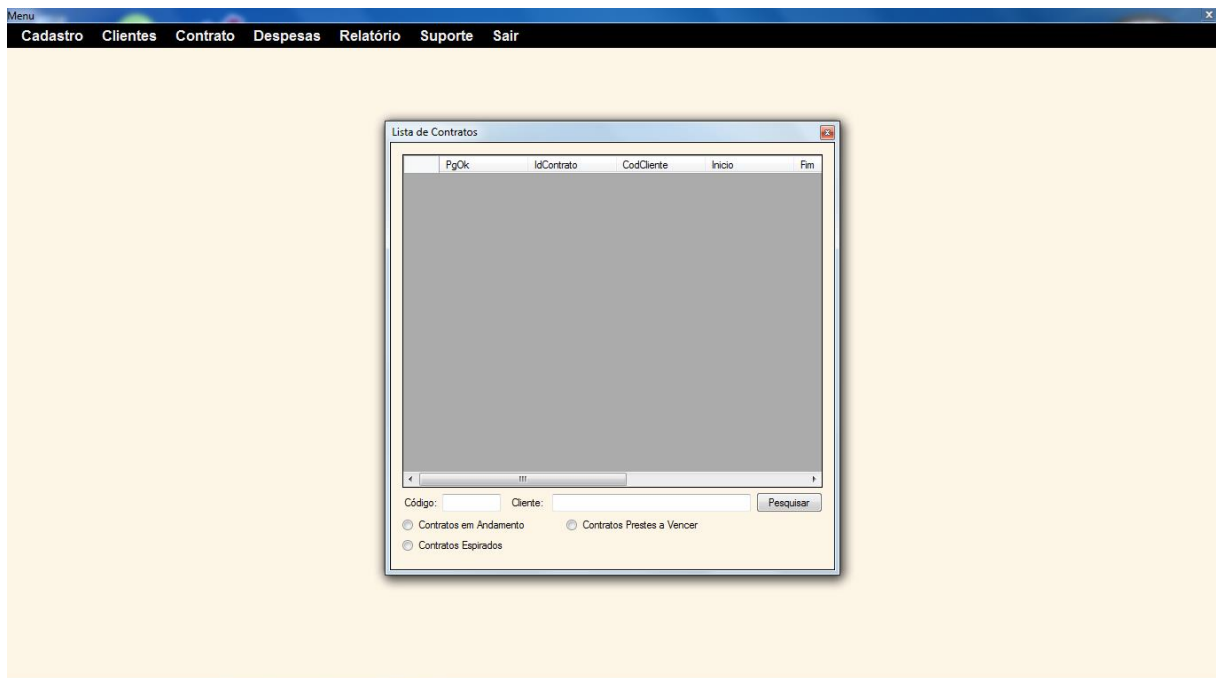


Figura 59 - Tela Lista de Contratos

Tela de Lançamentos de Despesas, Realiza os lançamentos e mostra-os aos usuários os mesmos, facilitando os futuros lançamentos. (Figura 52)

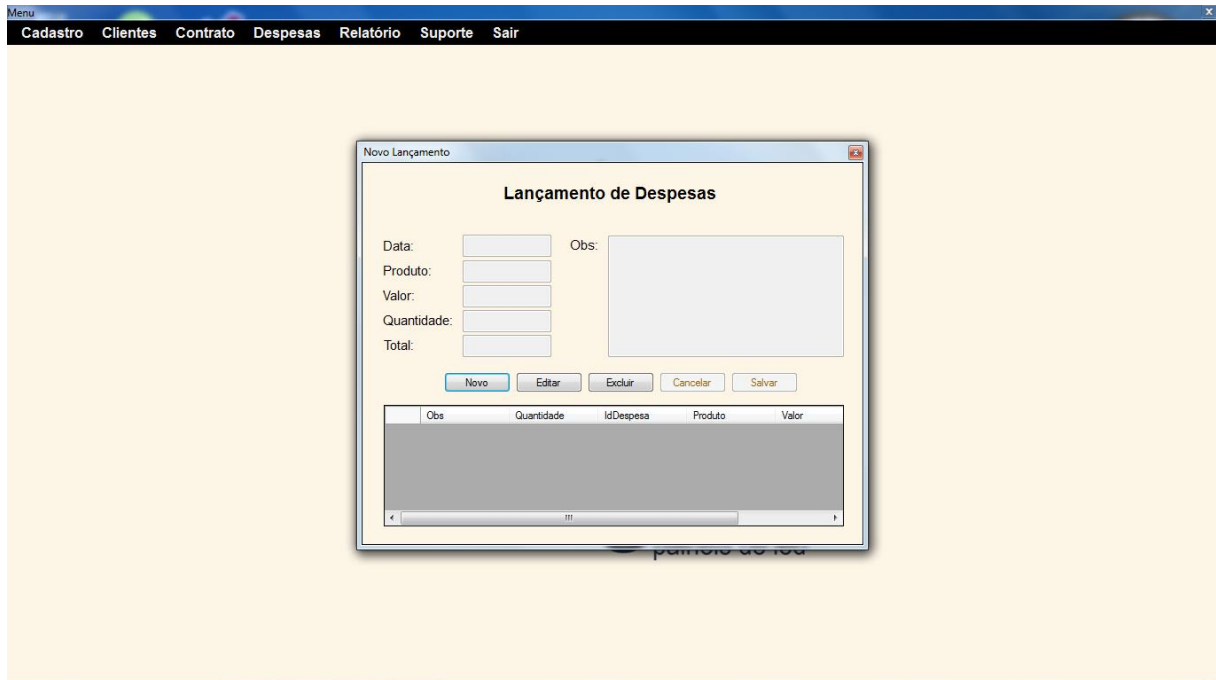


Figura 60 - Tela Lançamento de Despesas

Tela de Contratos, Realiza o cadastro dos novos contratos e mostra-os aos usuários os mesmos. (Figura 53)

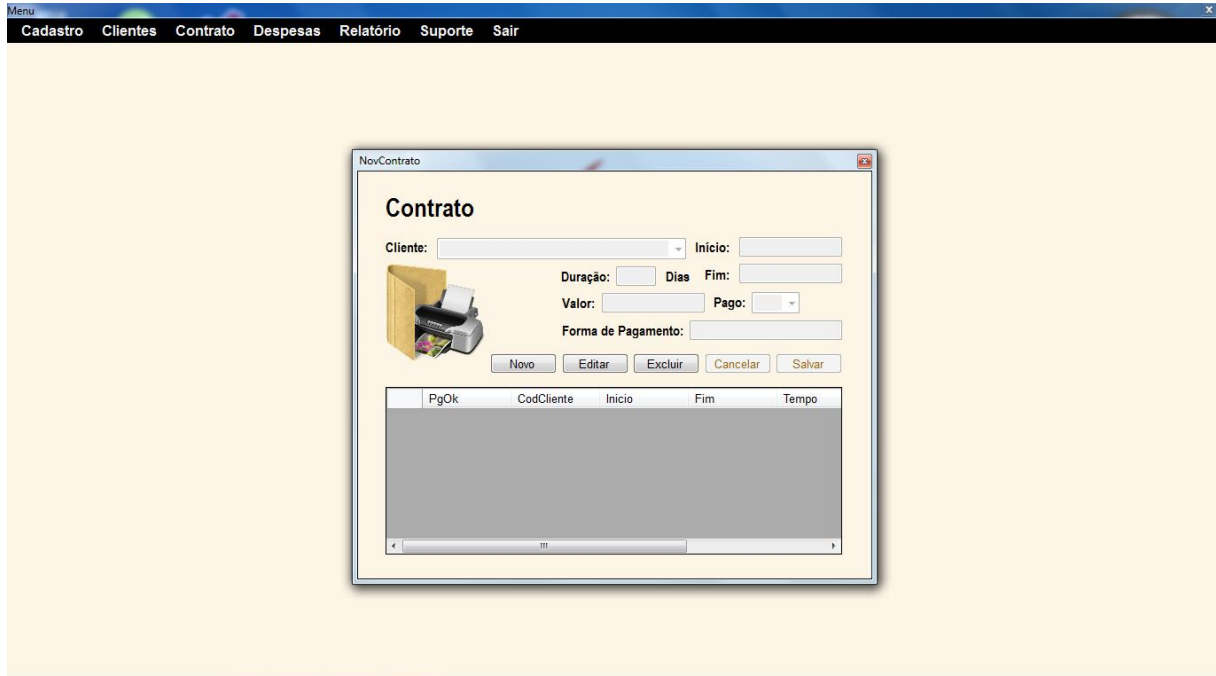


Figura 61 - Tela Lançamento de Contratos

Tela de suporte, mostra ao usuário do sistema as forma de contato ao administrador do software. (Figura 54)

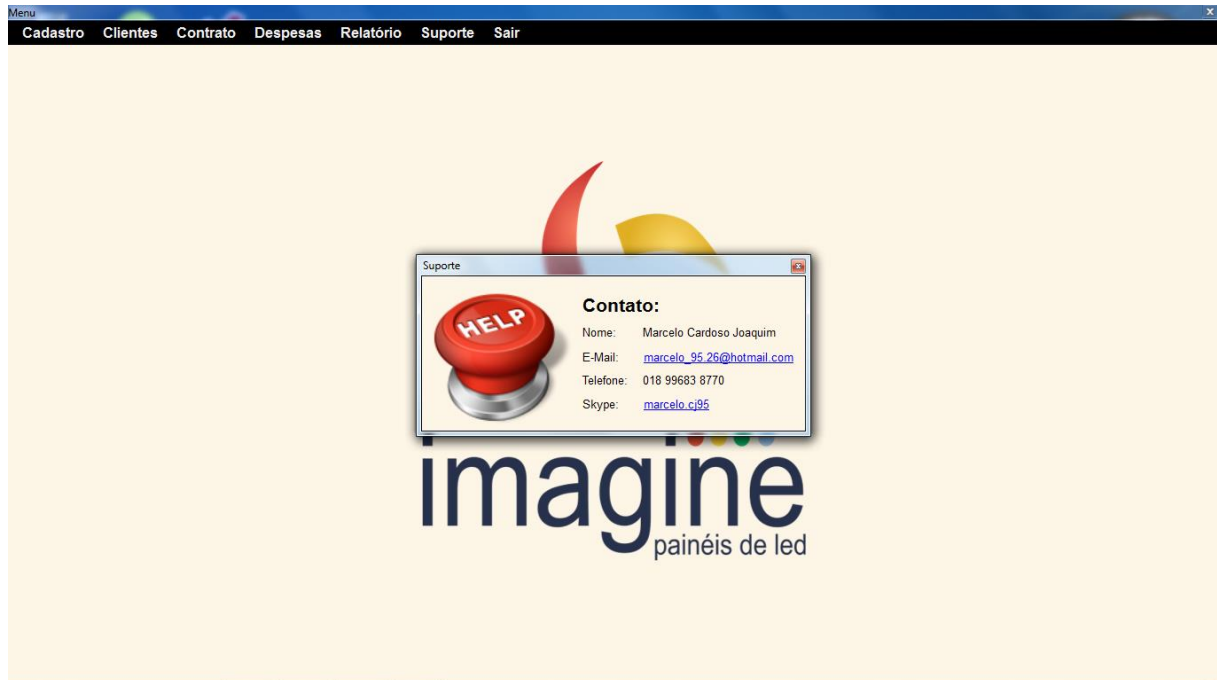


Figura 62 - Tela Suporte

5 – ORÇAMENTO

- 01 - Analista Programador.
- 01 - Notebook ASUS, Core i5, 8g de RAM, HD de 500g.
- 01 - Notebook PHILCO, Core 2 Duos, 4g de RAM, HD de 500g.
- Visual Studio Professional.
- Microsoft Windows Server 2012.

5.1 – ESTIMATIVA DE CUSTOS

Custo do Analista programador.

Custo Diário: R\$ 35,00 (Quarenta Reais).

Total de Dias: 152.

Custo Total ($152 * 40,00$) = R\$ 5.320,00 (Cinco mil e trezentos e vinte reais).

5.2 – CUSTOS DE MATERIAIS

Notebook Asus: R\$ 2.200,00.

Depreciação 1 ano: $R\$ 2.200,00 / 12$ (meses) = R\$ 183,33 (Sento e oitenta e três e trinta e três centavos) por mês.

Custo de um dia: $R\$ 183,33 / 30$ (dias) = R\$ 6,11 (Seis reais e onze centavos) por dia.

Custo de 152 dias: $R\$ 6,11 * 152$ = R\$ 928,72 (novecentos e vinte e oito e setenta e dois centavos).

Notebook Philco: R\$ 500,00.

Depreciação 1 ano: $R\$ 500,00 / 12$ (meses) = R\$ 41,67 (Quarenta e um reais e sessenta e sete centavos) por mês.

Custo de um dia: $R\$ 41,67 / 30$ (dias) = R\$ 1,39 (Um real e trinta e nove centavos) por dia.

Custo de 152 dias: $R\$ 1,39 * 152 = R\$ 211,11$ (Duzentos e onze reais e onze centavos).

Visual Studio Professional 2013: R\$ 795,00.

Depreciação 1 ano: $R\$ 795,00 / 12$ (meses) = R\$ 66,25 (Sessenta e seis reais e vinte e cinco centavos) por mês.

Custo de um dia: $R\$ 66,25 / 30$ (dias) = R\$ 2,21 (Dois Reais e vinte e um centavo) por dia.

Custo de 152 dias: $R\$ 2,21 * 152 = R\$ 335,67$ (Trezentos e trinta e cinco reais e sessenta e sete centavos).

Microsoft Windows Server 2012: R\$ 1.947,00.

Depreciação 1 ano: $R\$ 1.947,00 / 12$ (meses) = R\$ 162,25 (Sento e sessenta e dois reais e vinte e cinco centavos) por mês.

Custo de um dia: $R\$ 162,25 / 30$ (dias) = R\$ 5,41 (Cinco reais e quarenta e um centavo) por dia.

Custo de 152 dias: $R\$ 5,41 * 152 = R\$ 822,32$ (Oitocentos e vinte e dois reais e vinte e dois centavos).

5.3 – ORÇAMENTO DO PROJETO

Recursos	Valor
Analista Programador	R\$ 5.320,00
Notebook Asus	R\$ 2.200,00
Notebook Philco	R\$ 500,00
Visual Studio Professional	R\$ 795,00
Microsoft Windows Server 2012	R\$ 1.947,00
Total:	R\$ 10.762,00

Figura 63 – Orçamento

6 – CRONOGRAMA

O cronograma é um instrumento de planejamento e controle de tarefas, semelhante a um diagrama, em que são definidas e detalhadas minuciosamente as atividades a serem executadas durante um período estimado.

Atividades/Período	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO
Pré-Projeto								
Levantamento de Requisitos								
Validação de Requisitos								
Estrutura Analítica								
Caso de Uso								
Diagrama de Sequencia								
Diagrama de Classe								
Diagrama ER								
Sequenciamento de Atividades								
Orçamento								
Escrita da Qualificação								
Implementação								
Documentação Final do TCC								
Testes do Sistemas								
Apresentação Final								

Figura 64 - Cronograma

7 – CONCLUSÃO

As funcionalidades do sistema são especificamente o que empresa necessita, existem outros softwares no mercado, mas estes possuem funções que nunca serão usadas e ainda podem tornar mais difícil a vida do usuário do sistema.

A etapa de análise facilitou o entendimento do sistema. Por meio da lista de eventos e os casos de uso foi possível identificar com mais facilidade as funcionalidades do sistema. O diagrama de sequência facilitou a compreensão do funcionamento do software, enquanto que o diagrama de classes e o diagrama entidade relacionamento orientou no desenvolvimento do projeto.

No desenvolvimento deste projeto, existiram algumas dificuldades pelo fato de imprevistos técnicos e dificuldades no conhecimento da linguagem.

8 – REFERENCIAS

Alvarez, Miguel Angel Alvarez. Oque é cada tecnologia. WEB. Disponível em: <http://www.doislajeadosrs.com.br/programas/programas%20para%20manipulacao%20pagina%20internet/Curso%20HTML/O%20que%20%E9%20cada%20tecnologia%20-%20Manual%20completo/O%20que%20%E9%20cada%20tecnologia%20-%20Manual%20completo.htm>

Acesso em: 04/02/2015.

Costa, Luciano Costa. A estrutura analítica do projeto - EAP. WEB. Disponível em: <http://www.oficinadanet.com.br/artigo/gerencia/eap-a-estrutura-analitica-do-projeto>

Acesso em: 11/02/2015.

DB Designer, DB Designer Fork. WEB. Disponível em: <http://db-designer-fork.soft112.com/>

Acesso em 16/03/2015.

Astah. WEB. Disponível em: <http://astah.net/download>

Acesso em 16/03/2015.

Microsoft, Office. Visio Professional 2013. WEB. Disponível em: http://www.microsoftstore.com/store/msbr/pt_BR/pdp/Visio-Professional-2013/productID.260737200

Acesso em 11/02/2015.

Microsoft, Microsoft SQL Server 2008 Express. WEB. Disponível em: <https://www.microsoft.com/pt-br/download/details.aspx?id=1695>

Acesso em 20/03/2015.

Microsoft, SQL. Visão geral do SQL Server Express. WEB. Disponível em:

<https://technet.microsoft.com/pt-br/library/ms165588%28v=sql.105%29.aspx>

Acesso em: 04/02/2015.

Source Forge. FreeMind. WEB. Disponível em:

http://freemind.sourceforge.net/wiki/index.php/Main_Page

Acesso em 16/03/2015.

Square, Newtown. PMBOK. WEB. Disponível em:

http://www.cin.ufpe.br/~if717/Pmbok2000/pmbok_v2p/wsp_6.2.html

Acesso em 12/02/2015.