**DES - Documento de Engenharia de Software**

Versão: 0.2

12 de Setembro de 2016

**Projeto Batman**

Matheus de Oliveira Campos

Nathália Souza Campos

William Pinheiro Castro

EC205 - Documento Engenharia de Software.docx

Tabela de Revisões

| Versão | Principais Autores | Descrição da Versão | Data de Término | Aprovação e data | |
| --- | --- | --- | --- | --- | --- |
| V0.1 | Matheus  Nathália  William | Introdução, Escopo e Descrição de Funcionamento. | 26/08/16 | **Daniel Izario** | 27/08/2016 |
| V0.2 | Matheus  Nathália  William | Especificação de Requisitos Funcionais e Não -Funcionais. | 12/09/16 | **Daniel Izario** |  |

Índice

Tabela de Revisões 2

Índice 3

1. Lista de Figuras 5

2. Lista de Tabelas 6

3. Introdução 7

3.1 Definições, Acrônimos e Abreviaturas 7

4. Visão geral 8

4.1 Introdução 8

4.2 Escopo 8

4.3 Descrição de funcionamento 10

5. Especificação de Requisitos 12

5.1 Requisitos Funcionais 12

5.1.1 Req.1 – Efetuar o login no sistema. 12

5.1.2 Req.2 – Efetuar o cadastro dos vilões. 12

5.1.3 Req.3 – Editar os dados dos vilões. 12

5.1.4 Req.4 – Buscar e listar os vilões cadastrados. 13

5.1.5 Req.5 – Excluir os vilões cadastrados. 13

5.2 Diagrama de Casos de Uso 14

5.2.1 Descrição dos Atores 14

5.2.2 Descrição dos Casos de Uso 14

5.3 Fluxos de Eventos de Casos de Uso 15

5.4 Requisitos Não-Funcionais 17

5.4.1 Req.6 – Criação do Software com IDE Eclipse. 17

5.4.2 Req.7 – Linguagem de programação JAVA. 17

5.4.3 Req.8 – Banco de dados SQLite. 17

5.4.4 Req.9 – Rodar em qualquer Sistema Operacional. 17

5.4.5 Req.10 – Código comentado e identado. 17

5.4.6 Req.11 – Uso de programação orientada a objetos. 17

5.4.7 Req.12 – O sistema deverá ter disponibilidade de serviço acima de 98%. 17

5.4.8 Req.13 – Tempo de resposta do software alta. 17

5.4.9 Req.14 –Confiabilidade dos dados privados dos usuários. 17

6. Projeto Lógico 18

6.1 Diagrama de Classes 18

6.2 Diagrama de Sequência 18

6.3 Diagrama de Pacotes 19

6.4 Diagrama de Atividade 19

7. Anexos 20

7.1 Storyboarding 20

7.2 Estrutura Analítica do Projeto - EAP 20

7.3 Cronograma de Atividades 21

# Lista de Figuras

**Figura 1** - Ilustração da história. 8

**Figura 2** - Exemplo da implementação. 9

**Figura 3** - Fluxograma Geral do MOCEGO.BAT. 10

**Figura 4** - Fluxograma das funções do MOCEGO.BAT. 11

Figura 5 - Diagrama de casos de uso 14

Figura 6 - Fluxo de eventos (descrição) 15

Figura 7 - Fluxo de eventos (exemplo: caso de uso) 16

Figura 8 - Fluxo de eventos (exemplo: fluxo principal) 16

Figura 9 - Diagrama de Classes 18

Figura 10 - Diagrama de Pacotes 19

Figura 11 - Diagrama de Atividades 19

Figura 12 - Telas do Software 20

*Figura 13 - EAP* 20

*Figura 14 - Cronograma* 21

# Lista de Tabelas

Tabela 1 - Requisito Req.1 12

Tabela 2 - Requisito Req.2 12

Tabela 3 - Requisito Req.3 13

Tabela 4 - Requisito Req.4 13

Tabela 5 - Requisito Req.5 13

# Introdução

## Definições, Acrônimos e Abreviaturas

**Backup -** Nomenclatura que significa a cópia de dados para um meio seguro onde possa ser restaurado.

**UML -** Sigla que representa a padronização utilizada para a linguagem de modelagem.

**Banco de dados**- Servidor para armazenamento de dados.

# Visão geral

## Introdução

Batman, o Cavaleiro das Trevas e justiceiro de Gotham City, combate à criminalidade de sua cidade natal, onde seus pais foram brutalmente assassinados quando ele tinha apenas oito anos de idade.

Criado pelo seu mordomo Alfred Pennyworth e com sede de vingança, viajou ao redor do mundo aprendendo diversas artes de lutas marciais para que um dia possa tornar Gotham City pacifica e harmoniosa para viver.

Para combater o crime, Batman conta com um amigo secreto dentro da GCPD (Gotham City Police Department), o detetive James Gordon, juntos prendem e puni todos que quebram a lei de ordem e paz da cidade.

Chegou um dia que, Batman e o detetive Gordon, já prenderam mais de três mil criminosos e precisam de um software de controle para registrar todos os vilões de Gotham City, afim de saberem quais estão presos e quais estão soltos para praticarem maldades e também salvar algumas características de todos os vilões. Então secretamente contataram nossa empresa de tecnologia para desenvolver um software que faça o controle de todos os vilões e pediram total sigilo sobre o projeto.

A **Figura 1** abaixo, ilustra um pouco da história do desenho:



**Figura 1** - Ilustração da história.

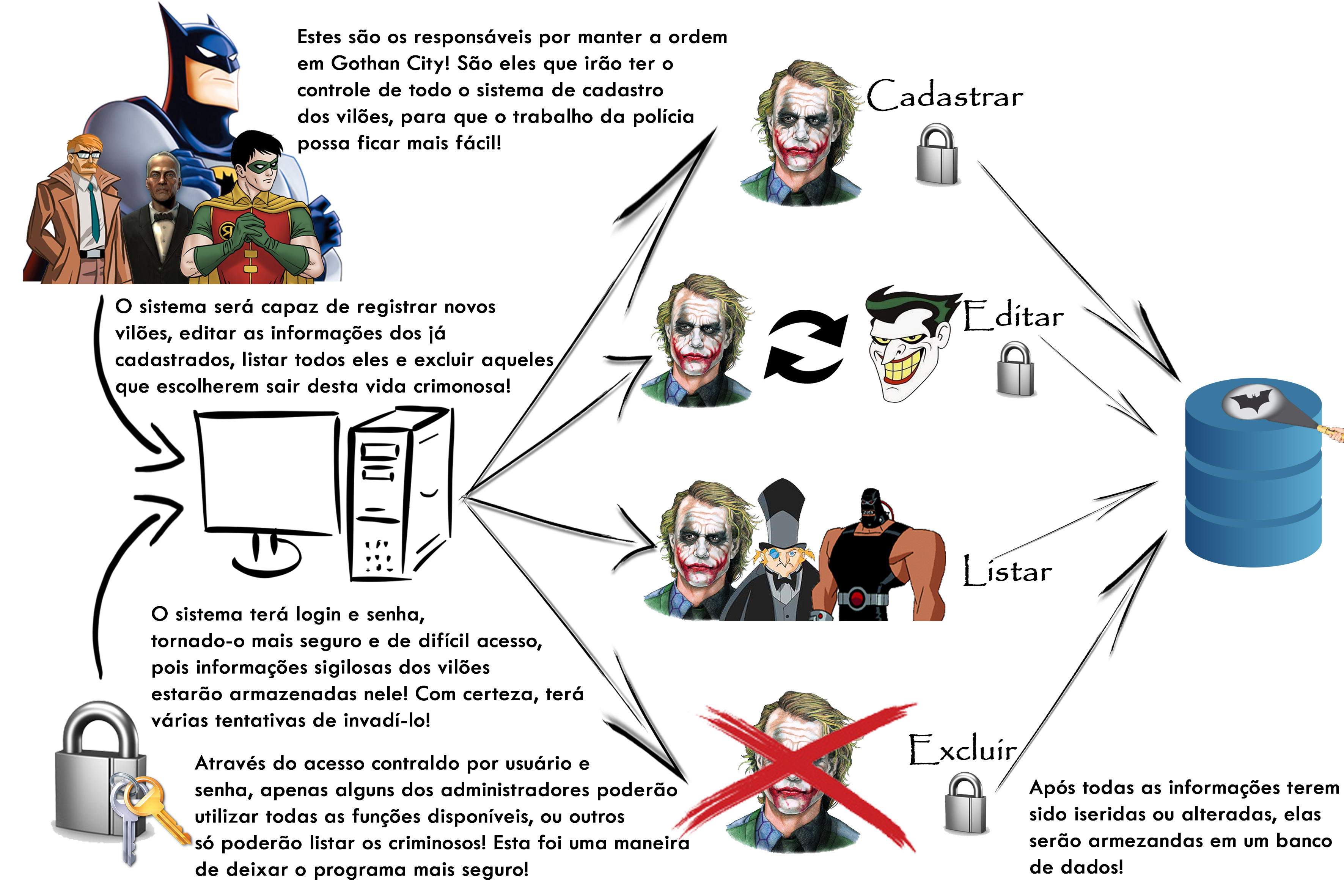
## Escopo

O Sistema terá acesso restrito, ou seja, algumas pessoas terão acesso as informações e funções do sistema, isso será controlado por Login e Senha. Serão 4 pessoas que terão o acesso (Batman, Detetive Gordon, Robin e Alfred), mas dessas 4 pessoas, 2 (Batman e Detetive Gordon) terão acesso privilegiado, ou seja, a todas as funções do sistema (Cadastrar, Editar, Listar e Excluir) e as outras 2 (Robin e Alfred) terão acesso a apenas uma função (Listar).

As funções disponíveis no sistema são:

* **Cadastrar:** principal função do sistema, nela serão inseridos os dados dos vilões, como o nome, apelido ou nome pelo qual é conhecido por todos, características físicas de vilão, modo de agir em seus crimes, status do vilão, categoria em que ele se encaixa, local de maior atuação e algumas observações importantes para defini-lo melhor e indicar seu histórico.
* **Editar:** caso alguns dos dados inseridos na hora de cadastrar necessitem de alterações, nesta função será possível modificá-los e ou atualiza-los. Se por acaso alguma informação durante o cadastro for esquecida ela poderá ser alterada e atualizada nesta função.
* **Listar:** todos os vilões cadastrados poderão ser listados através dessa função. Haverá algumas formas de lista-los, que podem deixar o processo de busca mais rápido e eficiente, ou seja, terão alguns filtros de busca que ficará muito fácil de encontrar o vilão caso a pessoa que queira busca-lo lembre de apenas algumas informações do vilão.
* **Excluir:** caso algum criminoso tem sido cadastrado por engano ou ele mudou de lado para auxiliar aqueles que combatem o crime, ele poderá ser retirado do sistema.

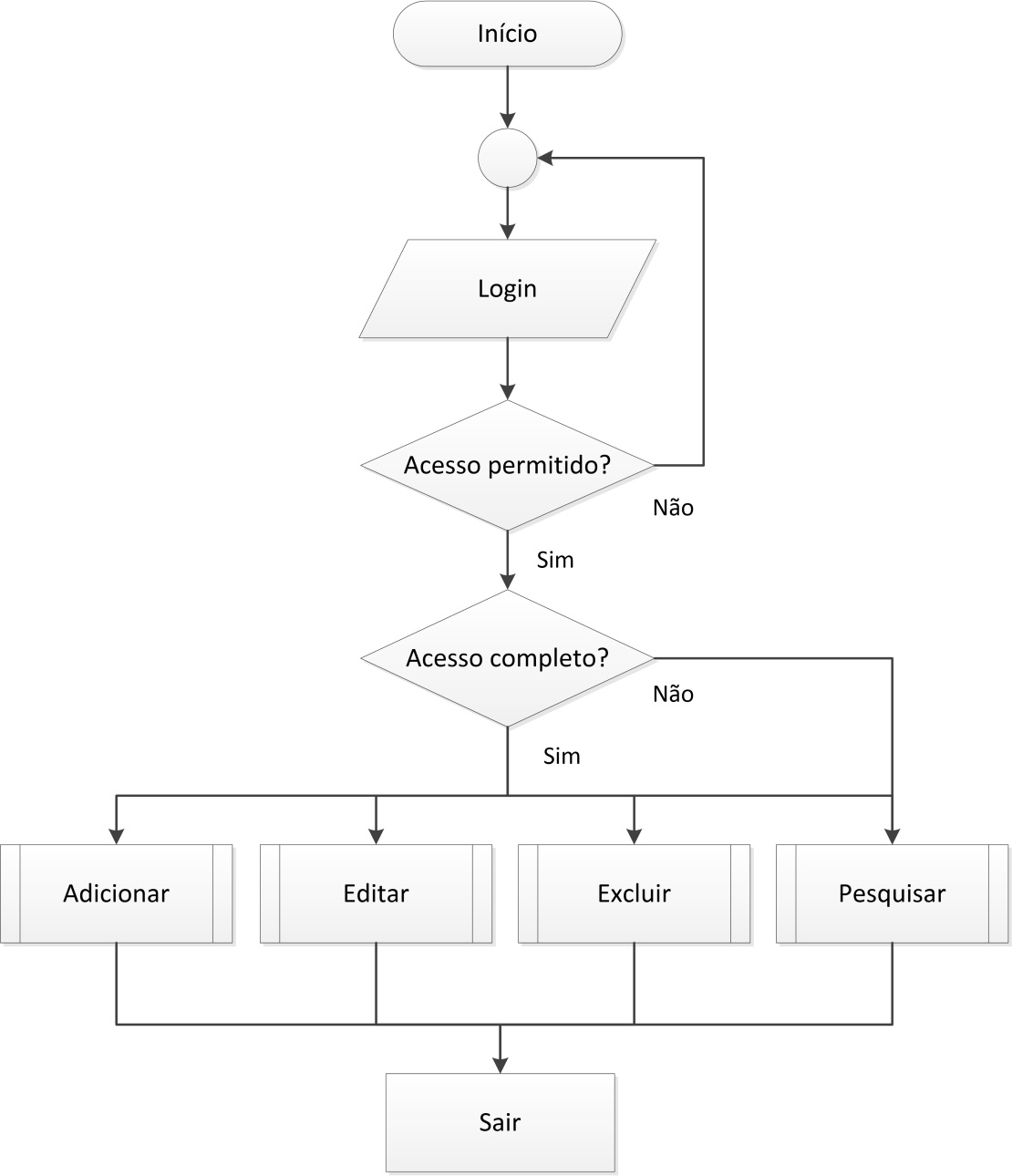
Segue abaixo a **Figura 2**, exemplificando o funcionamento do software:



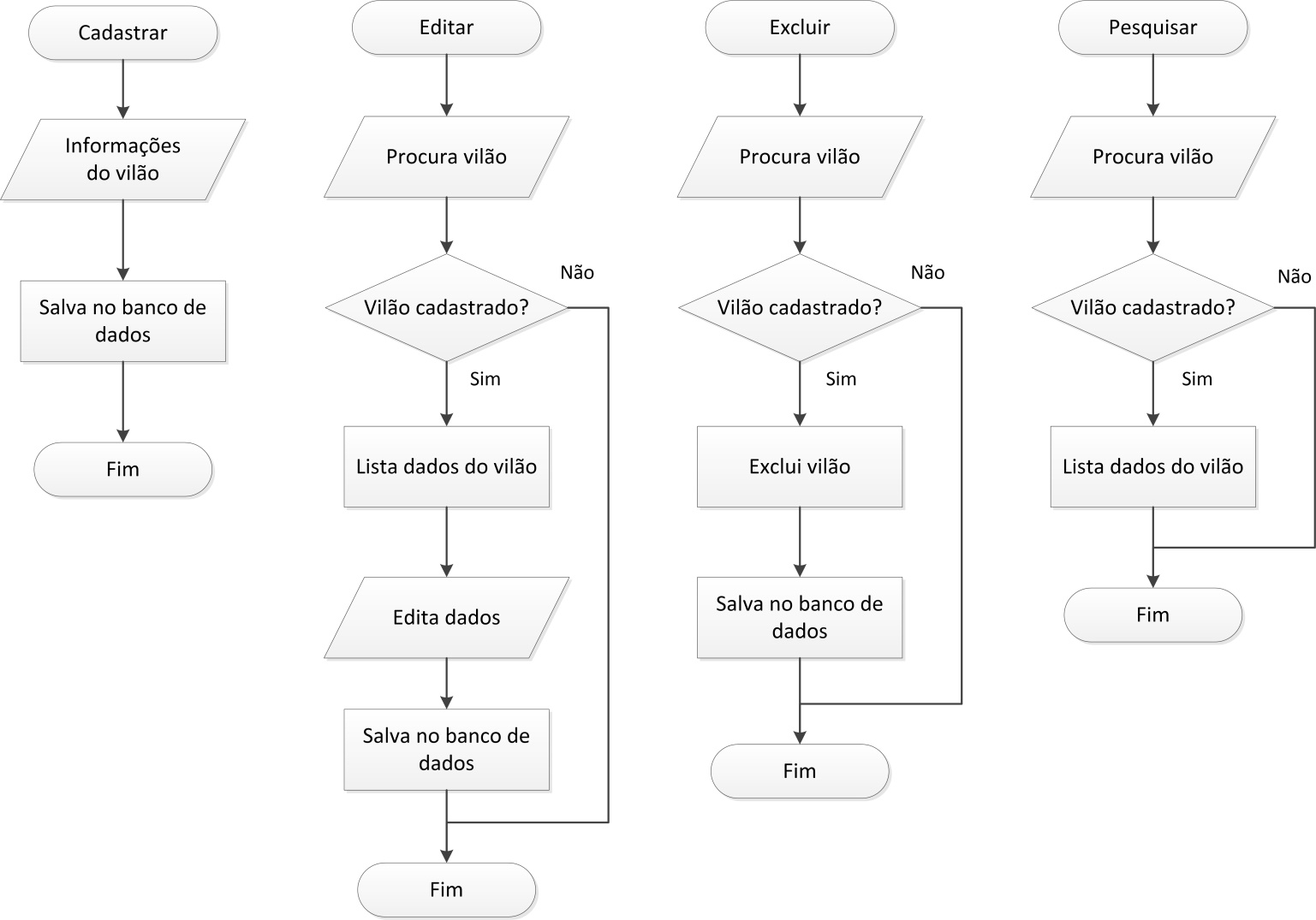
**Figura 2** - Exemplo da implementação.

## Descrição de funcionamento

O software será desenvolvido através da plataforma Eclipse Mars, na linguagem de programação Java. Na **Figura 3**, é ilustrado o Fluxograma Geral do sistema e na **Figura 4**, o Fluxograma das funções contidas no programa.



**Figura 3** - Fluxograma Geral do MOCEGO.BAT.

****

**Figura 4** - Fluxograma das funções do MOCEGO.BAT.

# Especificação de Requisitos

## Requisitos Funcionais

### Req. – Efetuar o login no sistema.

|  |  |
| --- | --- |
| **Detalhamento** | Cada usuário deverá entrar com os seguintes dados:  - Usuário;  - Senha. |
| **Observação** | O usuário já é previamente cadastrado pelos desenvolvedores do sistema, necessitando apenas de suas credenciais para acessa-lo. |
| **Prioridade** | Alta |

Tabela 1 - Requisito Req.1

### Req. – Efetuar o cadastro dos vilões.

|  |  |
| --- | --- |
| **Detalhamento** | Cada vilão irá ter os seguintes dados informados pelo usuário:  - Nome;  - Apelido;  - Características físicas do vilão;  - Como agir;  - Status;  - Categoria criminal;  - Locais de ação;  - Observações. |
| **Observação** | O cadastro só poderá ser feito pelos usuários que tem acesso total do sistema. |
| **Prioridade** | Alta |

Tabela 2 - Requisito Req.2

### Req.3 – Editar os dados dos vilões.

|  |  |
| --- | --- |
| **Detalhamento** | Qualquer vilão poderá ter os seguintes dados alterados pelo usuário:  - Nome;  - Apelido;  - Características físicas do vilão;  - Como agir;  - Status;  - Categoria criminal;  - Locais de ação;  - Observações. |
| **Observação** | A edição só poderá ser feita pelos usuários que tem acesso total do sistema. |
| **Prioridade** | Alta |

Tabela 3 - Requisito Req.3

### Req.4 – Buscar e listar os vilões cadastrados.

|  |  |
| --- | --- |
| **Detalhamento** | Qualquer vilão poderá ser buscado e as seguintes informações serão exibidas:  - Nome;  - Apelido;  - Características físicas do vilão;  - Como agir;  - Status;  - Categoria criminal;  - Locais de ação;  - Observações. |
| **Observação** | A busca poderá ser feita por qualquer usuário do sistema. A busca será realizada através do “nome”, “apelido”, “Status” ou “Locais de ação” informados pelo usuário. |
| **Prioridade** | Alta |

Tabela 4 - Requisito Req.4

### Req.5 – Excluir os vilões cadastrados.

|  |  |
| --- | --- |
| **Detalhamento** | Qualquer vilão poderá ser totalmente excluindo pelo usuário:  - Nome;  - Apelido;  - Características físicas do vilão;  - Como agir;  - Status;  - Categoria criminal;  - Locais de ação;  - Observações. |
| **Observação** | Toda as informações serão deletadas, não será possível excluir apenas algumas informações do vilão, ele por completo será excluído do banco de dados. Só poderá realizar esta operação o usuário com o controle total do sistema |
| **Prioridade** | Média |

Tabela 5 - Requisito Req.5

## Diagrama de Casos de Uso

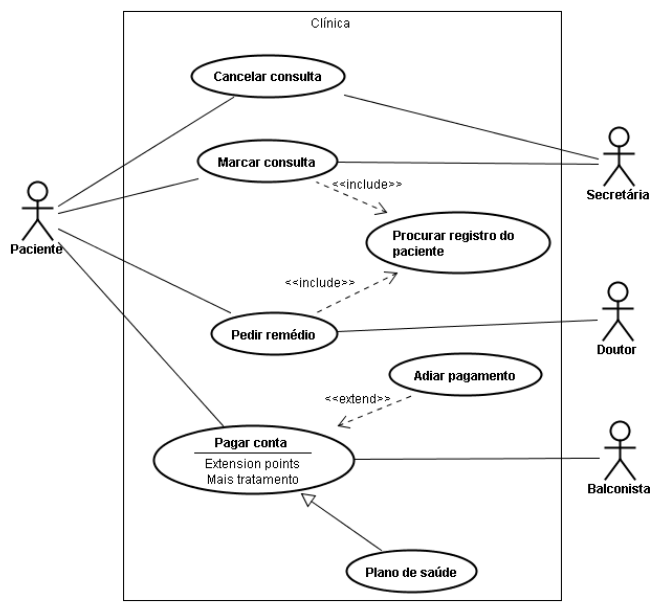


Figura - Diagrama de casos de uso

### Descrição dos Atores

**A1 – Administrador**

O Administrador tem acesso à todas funcionalidades de Manter Viagens, Reservar Viagem, Manter Clientes, Manter Funcionários.

### Descrição dos Casos de Uso

**CaU1 – Manter Clientes**

Este caso de uso tem como objetivo manipular os dados dos clientes no banco de dados. Ela é composta pelas funcionalidades de cadastrar, listar, editar e excluir clientes. Somente o Administrador tem acesso à este caso de uso

## Fluxos de Eventos de Casos de Uso

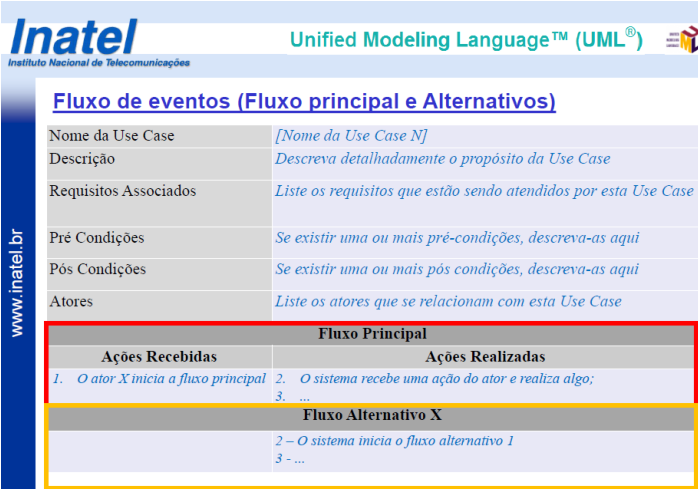


Figura - Fluxo de eventos (descrição)

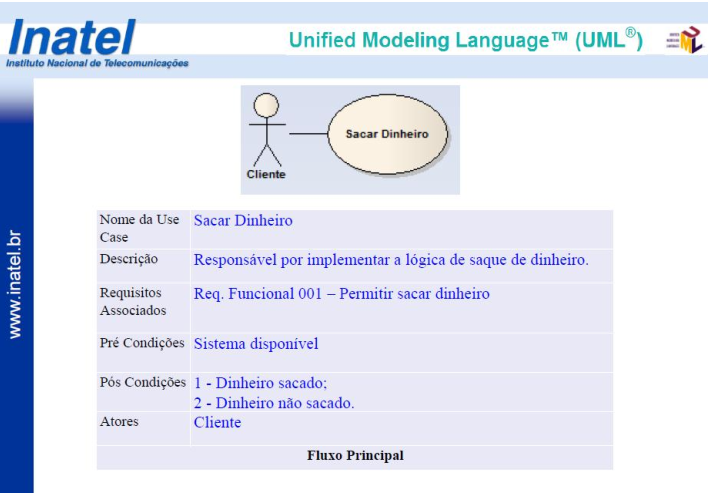


Figura - Fluxo de eventos (exemplo: caso de uso)

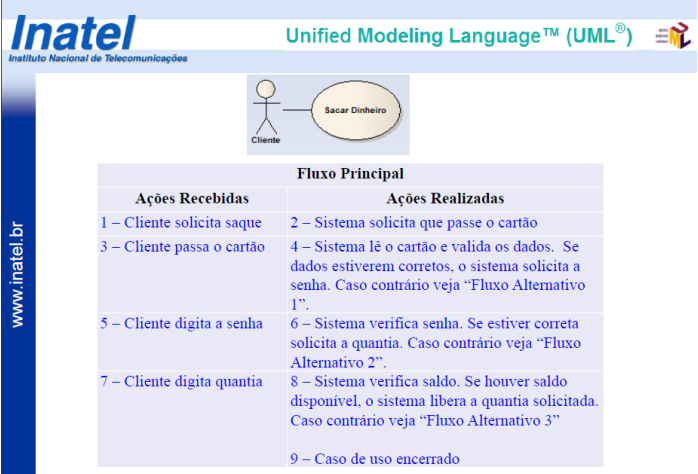


Figura - Fluxo de eventos (exemplo: fluxo principal)

## Requisitos Não-Funcionais

### Req.6 – Criação do Software com IDE Eclipse.

IDE escolhida pela sua facilidade e suporte dos desenvolvedores, sendo cotada como umas das melhores plataformas para desenvolvimento MVC (Model, View, Controller).

### Req.7 – Linguagem de programação JAVA.

Java é uma linguagem usada no mundo todo, com mais 3 bilhões de dispositivos que utilizam a tecnologia.

### Req.8 – Banco de dados SQLite.

É um banco dados muito utilizado para aplicações simples, que não demandam alto índice de requisições, é de fácil implementação e usabilidade.

### Req.9 – Rodar em qualquer Sistema Operacional.

Como Java é uma linguagem interpretada, o programa deve funcionar nos principais sistemas operacionais que suportam Java, Windows, Linux, OSX.

### Req.10 – Código comentado e identado.

É importante que o código seja devidamente comentado e identado para que futuramente outras pessoas possam trabalhar no código e também promover a facilidade no desenvolvimento do projeto para a equipe, para que todos entendam o código criado.

### Req.11 – Uso de programação orientada a objetos.

O software deverá ser feito orientado a objetos para tratar as características dos criminosos.

### Req.12 – O sistema deverá ter disponibilidade de serviço acima de 98%.

O software deverá ter disponibilidade acima de 98% do dia, segundo determinações do cliente.

### Req.13 – Tempo de resposta do software alta.

O tempo de resposta do software deve ser alta, porque trabalha com informações críticas para o cliente.

### Req.14 –Confiabilidade dos dados privados dos usuários.

O sistema não deverá fornecer aos usuários nenhum dado de cunho privativo do cliente.

# Projeto Lógico

## Diagrama de Classes

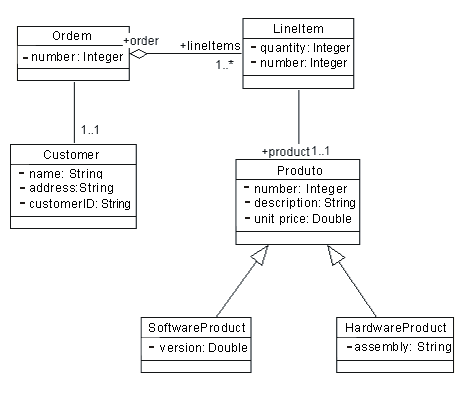


Figura - Diagrama de Classes

## Diagrama de Sequência

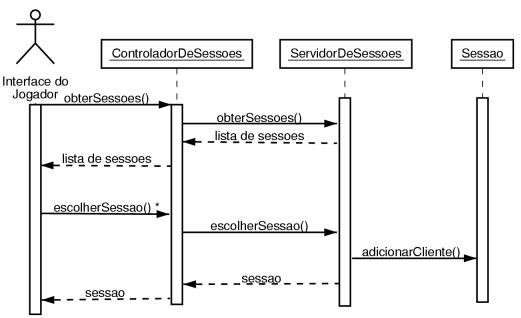


Figura 9 - Diagrama de Sequência

## Diagrama de Pacotes



Figura - Diagrama de Pacotes

## Diagrama de Atividade

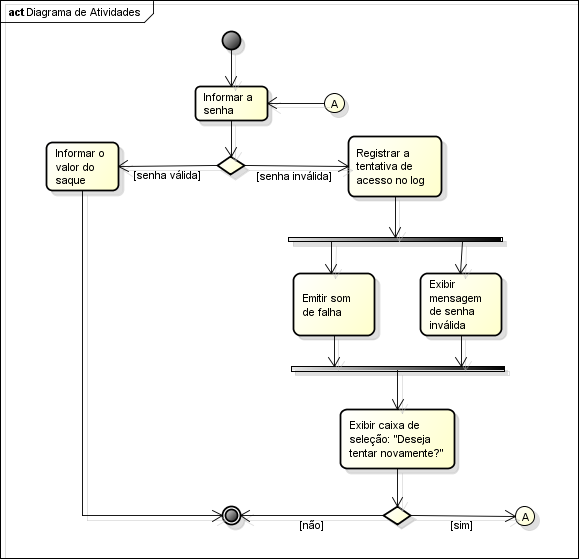


Figura - Diagrama de Atividades

# Anexos

## Storyboarding

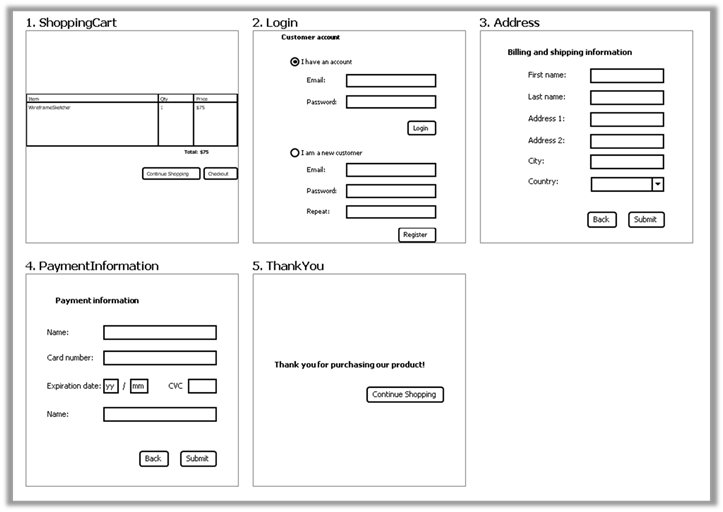
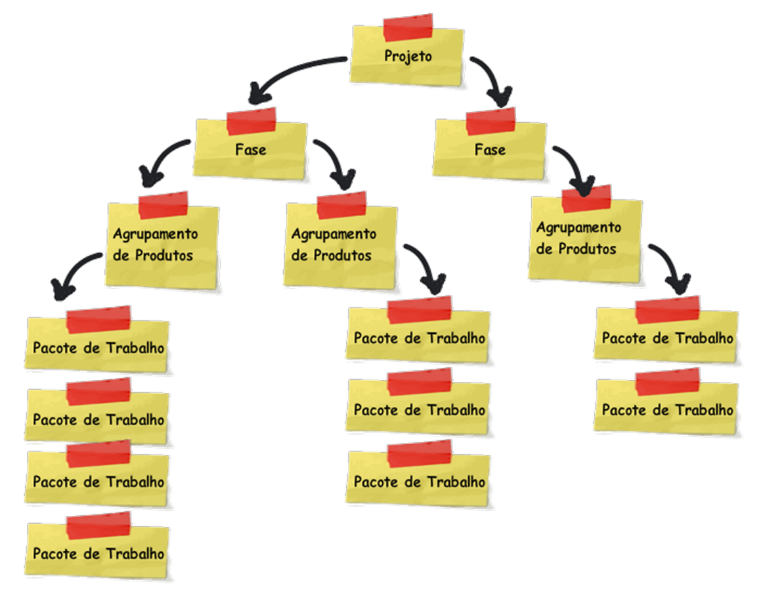


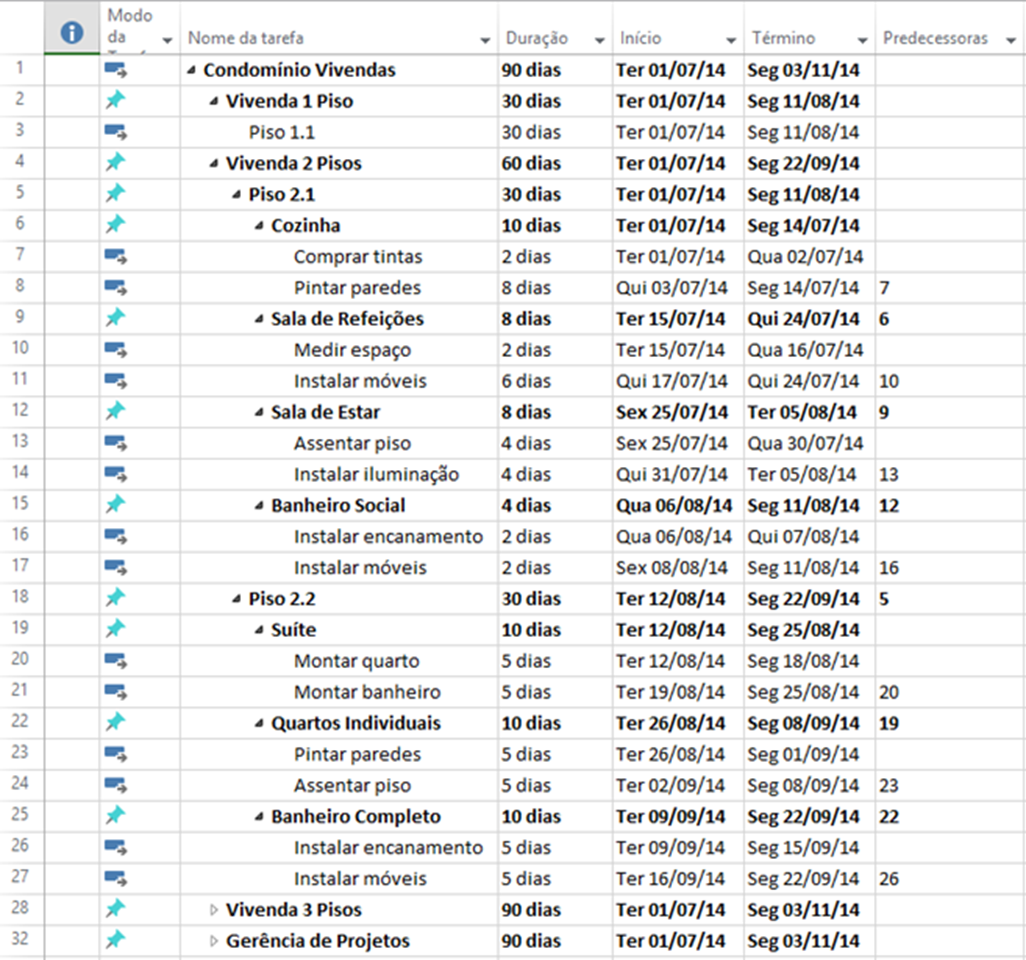
Figura - Telas do Software

## Estrutura Analítica do Projeto - EAP



*Figura 13 - EAP*

## Cronograma de Atividades



*Figura 14 - Cronograma*