



# Sistema para **Gestão** **de Locadora de** **Veículos**

**Equipe:** Alexandre Wesley dos Santos Silva, Alyson Clenardo Souza Pereira, Guilherme de Aquino Pacheco, João Manoel Torres Padilha



# Sumário

1. Introdução e Contextualização
2. Objetivo
3. Processo de Desenvolvimento
4. Requisitos: Funcionais e Não Funcionais
5. Diagrama de Casos de Uso
6. Diagrama de Classes
7. Diagramas de Sequência
8. Diagrama de Atividades
9. Conclusão

# 1. Introdução e Contextualização



# 1. Introdução e Contextualização

A **"MasterLocadora"** enfrenta desafios operacionais devido ao uso de um sistema de gestão manual com planilhas e documentos impressos, resultando em agendamentos duplicados, falta de transparência e atrasos na manutenção preventiva.

Para superar esses desafios, a empresa busca modernizar seus processos com um sistema de software integrado, visando maior eficiência operacional, transparência e aprimoramento da experiência do cliente, para se manter competitiva no mercado de locação de veículos.

## MasterLocadora



## 2. Objetivo





## 2. Objetivo

A solução proposta para a MasterLocadora consiste na implementação de um sistema de gestão de aluguel de veículos. Este sistema integrado proporcionará maior eficiência operacional, eliminando agendamentos duplicados, otimizando a gestão da frota, agendando manutenções preventivas, e oferecendo transparência através de um dashboard gerencial.

### **3. Processo de Desenvolvimento**



### 3. Processo de Desenvolvimento

#### Requisitos:

Identificação e definição das necessidades do cliente.

#### Análise:

Análise detalhada dos requisitos para entender desafios e oportunidades.

#### Projeto:

Criação da arquitetura e estrutura do software.

#### Codificação:

Codificação do software conforme o design.





### 3. Processo de Desenvolvimento

#### Testes:

Realização de testes unitários e de integração.

#### Implantação:

Lançamento do software no ambiente de produção.

#### Manutenção:

Suporte contínuo, correção de bugs e atualizações.

#### Avaliação:

Coleta de feedback dos usuários para melhorias contínuas.



### **3. Importância das Etapas**





## 3.1. Importância das Etapas

- Organização e Estruturação:
- Controle de Qualidade:
- Economia de Recursos:
- Comunicação Eficiente:
- Maior Rastreabilidade:
- Aprimoramento Contínuo:
- Gerenciamento de Riscos:
- Previsibilidade e Cumprimento de Prazos

## **4. Requisitos: Funcionais e Não Funcionais**



## 4. Requisitos

### Funcionais:

- Descrição de Funcionalidades
- Requisitos de Sistema
- Cenários de Uso
- Regras de Negócios
- Interações com Usuário



## 4. Requisitos

### Não Funcionais:

- Desempenho:
- Segurança:
- Usabilidade:
- Confiabilidade:
- Manutenibilidade:
- Compatibilidade:
- Escalabilidade:



## 4.1. Requisitos: Funcionais



## 4.1. Requisitos Funcionais (RF)

### RF1. Cadastro de Veículos:

O sistema deve permitir o cadastramento de veículos, incluindo informações como placa, modelo, ano, quilometragem atual, e dados de revisão como data da última revisão e quilometragem correspondente. Além disso, deve ser possível associar cada veículo a uma unidade específica da locadora e definir a taxa de locação diária por classe de veículo.







## 4.1. Requisitos Funcionais (RF)

### RF2. Cadastro de Clientes:

O sistema deve possibilitar o registro completo dos clientes, incluindo nome, telefone, endereço e CPF, facilitando o processo de atendimento e comunicação com os clientes.



## 4.1. Requisitos Funcionais (RF)

### RF3. Cadastro de Locação:

O sistema deve permitir o cadastro detalhado das locações, incluindo informações como funcionário responsável, unidade da locadora, quilometragem de saída e chegada, detalhes do carro alugado (incluindo a placa), datas de saída e chegada, bem como os valores da locação



## 4.1. Requisitos Funcionais (RF)

### RF4. Cadastro de Funcionários:

O sistema deve permitir o registro completo dos funcionários da locadora, incluindo nome, cargo e unidade de trabalho de cada funcionário, garantindo um controle eficaz sobre a equipe.



## 4.1. Requisitos Funcionais (RF)

### RF5. Reservas Online:

O sistema deve permitir que os clientes façam reservas de veículos online, escolhendo a unidade, datas de locação e devolução, classe de veículo e fornecendo informações de contato.



## 4.1. Requisitos Funcionais (RF)

### RF6. Avaliações e Feedback:

Após a devolução do veículo, os clientes devem ter a opção de fornecer feedback e avaliações sobre a experiência de locação.



## 4.1. Requisitos Funcionais (RF)

### RF7. Cadastro de Unidade da Locadora:

O sistema deve possibilitar o cadastro das unidades da locadora, incluindo informações de localização, nome e detalhes sobre os veículos disponíveis em cada unidade, bem como a pessoa responsável por cada unidade.





## 4.1. Requisitos Funcionais (RF)

### RF8.1 Relatório de Locação:

O sistema deve gerar relatórios de locações com a capacidade de aplicar filtros, permitindo segmentar as locações por unidade, datas de saída e chegada, cliente, funcionário e veículo, assim como por valor de locação.

### RF8.2 Relatório de Veículos:

O sistema deve gerar relatórios sobre veículos disponíveis e alugados, segmentados por unidade, com um filtro especial para identificar veículos com mais de 30 mil quilômetros rodados, indicando sua possibilidade de revenda.

## 4.1. Requisitos: Não Funcionais





## 4.1. Requisitos Não Funcionais (RNF)

### RNF1. Segurança e Autenticação:

O sistema deve oferecer segurança com um sistema de login e senha, garantindo acesso restrito e exclusivo para gestores, funcionários e clientes. As informações devem ser protegidas por criptografia.



## 4.1. Requisitos Não Funcionais (RNF)

### RNF2. Interface do Usuário:

O sistema deve fornecer uma interface com boa experiência do usuário e de uso intuitivo.



## 4.1. Requisitos Não Funcionais (RNF)

### RNF3. Desempenho:

O sistema deve ser capaz de lidar com um grande volume de transações simultâneas, garantindo resposta rápida mesmo durante períodos de alta demanda, como feriados ou estações turísticas.



## 4.1. Requisitos Não Funcionais (RNF)

### RNF4. Manutenibilidade:

O código-fonte do sistema deve ser bem estruturado e comentado para facilitar a manutenção contínua. Além disso, deve ser implementado um sistema eficiente de controle de versão para acompanhar as alterações no código.



## 5. Diagrama de Casos de Uso



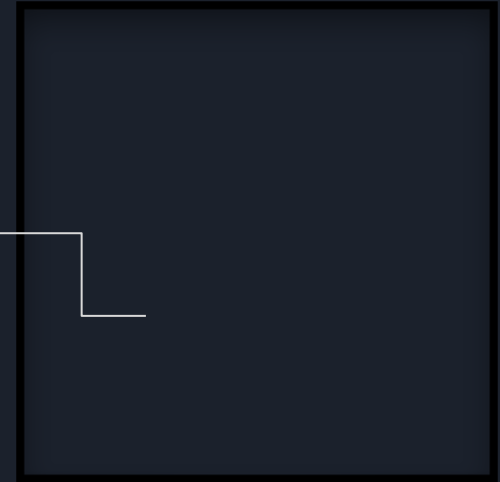
# 5. Diagrama de Casos de Uso

## Definição

Os diagramas de caso de uso servem para modelar o comportamento de um sistema e desempenham um papel crucial na captura dos requisitos desse sistema. Esses diagramas oferecem uma visão de alto nível ao descrever as funcionalidades principais e o escopo do sistema, focando nas interações entre o sistema em questão e os seus agentes externos.

## Propósito

Descrever o que o sistema faz e como os agentes o usam, mas não como o sistema opera internamente.



## 5.1. Elementos do Diagrama de Casos de Uso





## 5.1. Elementos do Diagrama de Casos de Uso

- Casos de Uso
- Agentes
- Subsistemas
- Relacionamentos em Diagrama de Casos de Uso



## 5.1. Elementos do Diagrama de Casos de Uso

- Casos de Uso

Descreve uma função que um sistema desempenha para alcançar a meta do usuário. Um caso de uso deve produzir um resultado observável que seja valioso para o usuário do sistema.





## 5.1. Elementos do Diagrama de Casos de Uso

- Agentes

Representam uma função de um usuário que interage com o sistema.

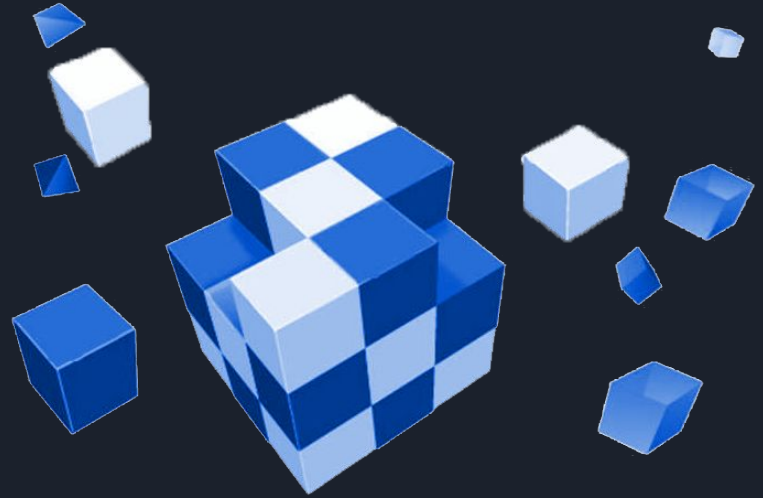




## 5.1. Elementos do Diagrama de Casos de Uso

- Subsistemas

Representam componentes de larga escala no sistema que está sendo modelado.

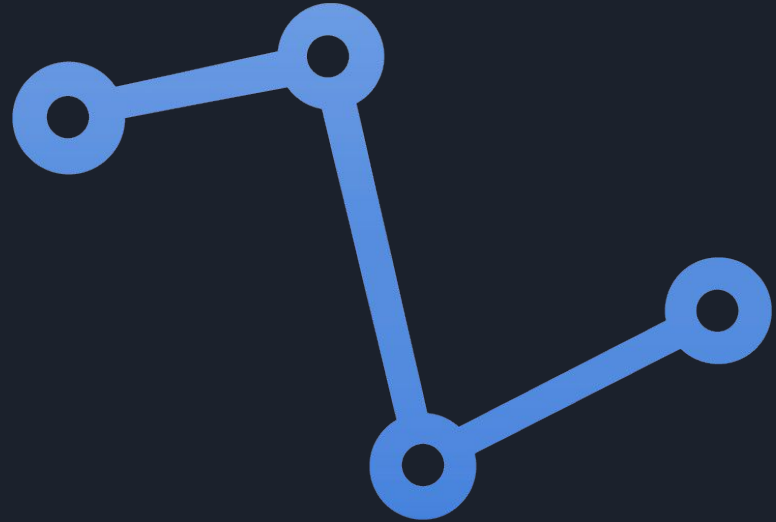




## 5.1. Elementos do Diagrama de Casos de Uso

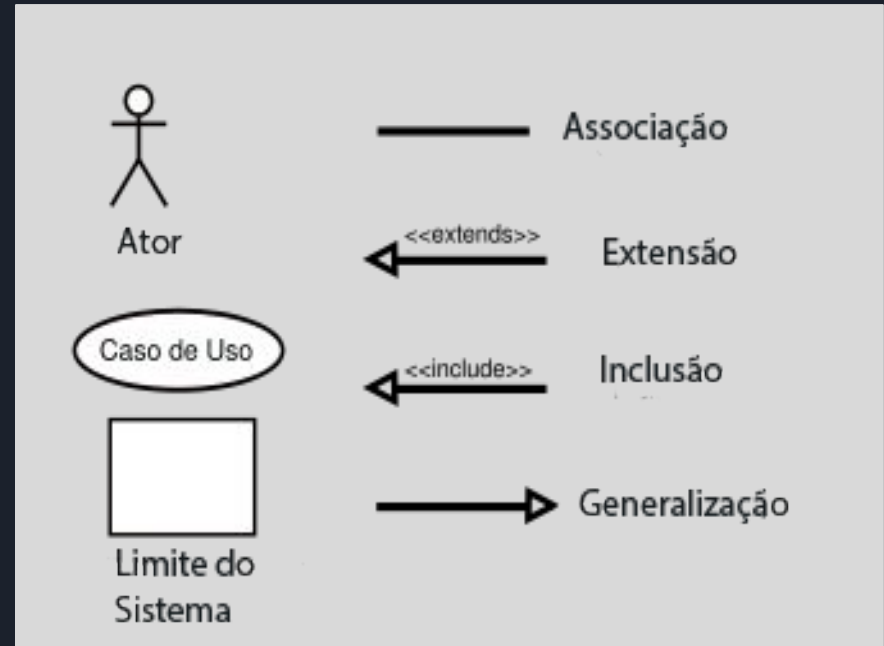
- Relacionamento em Diagrama de Casos de Uso

Na UML, um relacionamento é uma conexão entre elementos de modelo.

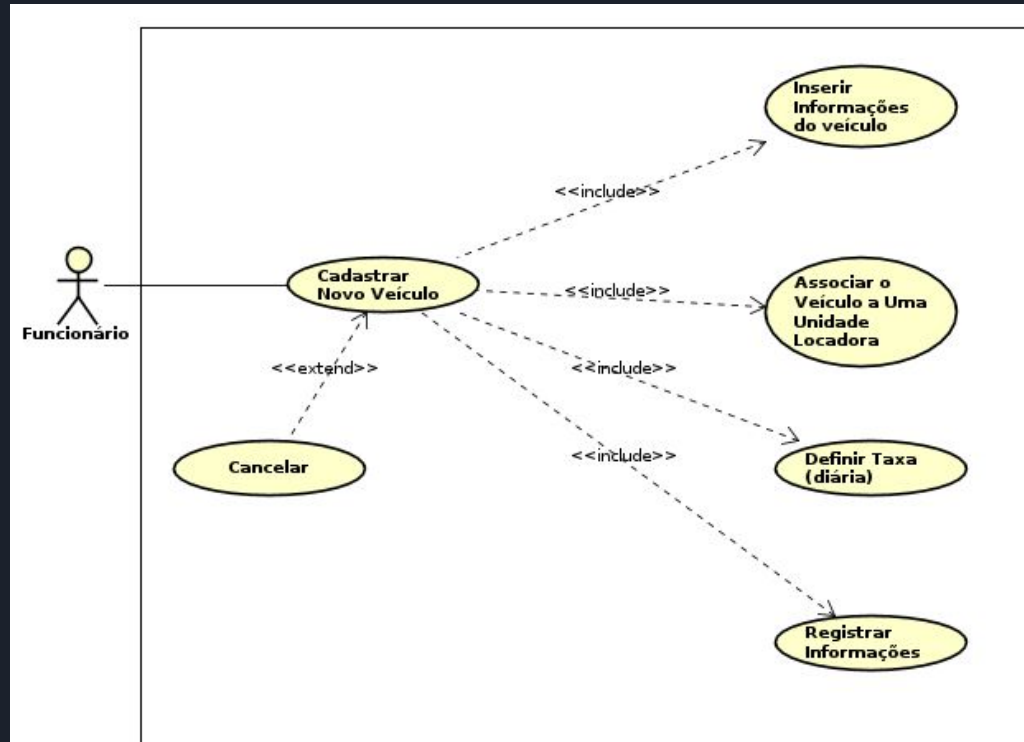


## 5.1. Elementos do Diagrama de Casos de Uso

- Notação



## 5.1. Elementos do Diagrama de Casos de Uso

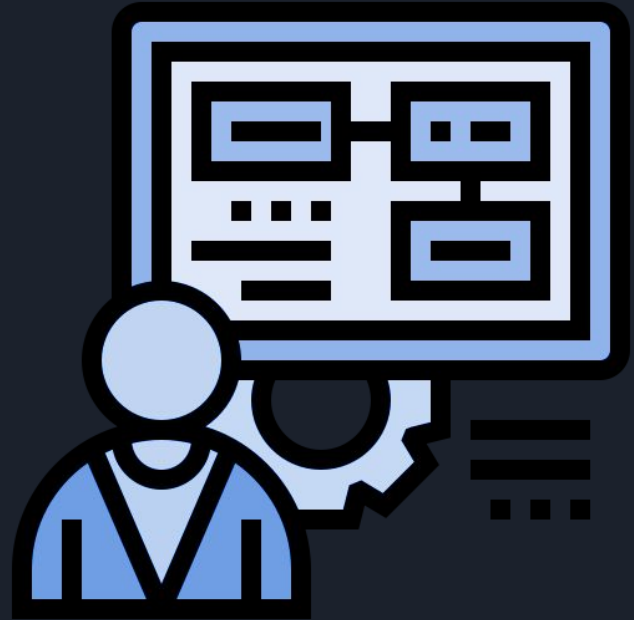


## 6. Diagrama de Classes



## 6. Diagrama de Classes

- O que é um Diagrama de Classe?
- Para que serve?





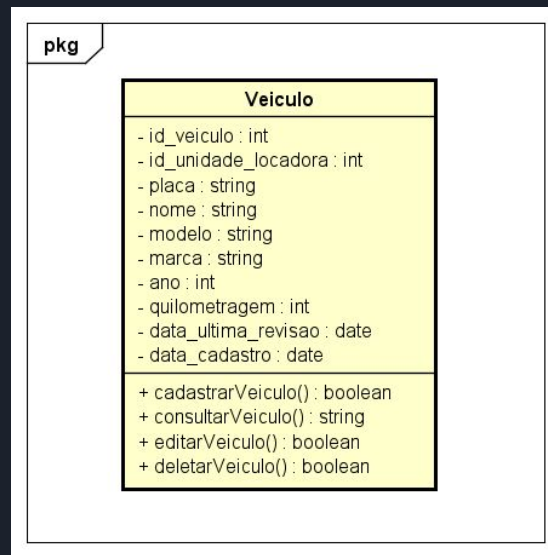
# 6. Diagrama de Classes

## Definição

O diagrama de classes é uma representação visual do design de um sistema orientado a objetos. Ele descreve a estrutura estática do sistema, mostrando as classes, seus atributos, métodos e as relações entre elas.

## Propósito

Explicar que o diagrama de classes ajuda a visualizar a estrutura do sistema, facilitando a compreensão das entidades principais, suas propriedades e como elas interagem.



## 6.1. Elementos do Diagrama de Classes

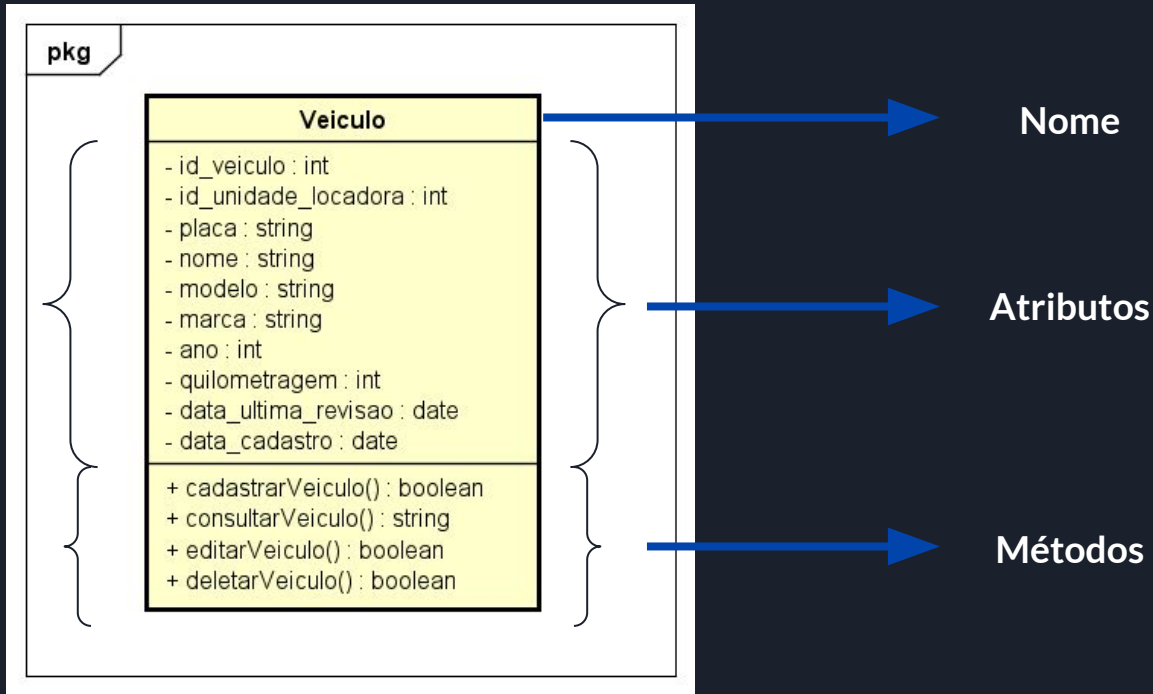




## 6.1. Elementos do Diagrama de Classes

- Classes
- Modificadores de Acesso
- Atributos
- Métodos
- Relacionamentos

## 6.1. Elementos do Diagrama de Classes

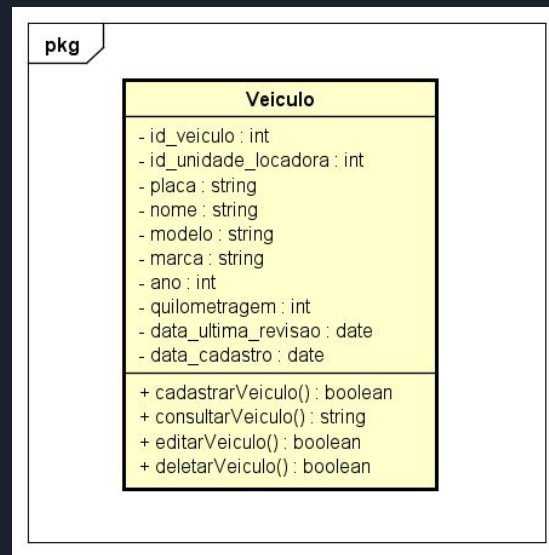


# 6.1. Elementos do Diagrama de Classes

## Modificadores de acesso de membro

Todas as classes têm diferentes níveis de acesso, dependendo do modificador de acesso (visibilidade). Veja os níveis de acesso com seus símbolos correspondentes:

- Público (+)
- Privado (-)
- Protegido (#)
- Pacote (~)
- Derivado (/)
- Estático (sublinhado)



## 6.1. Elementos do Diagrama de Classes



Associação Unidirecional



Associação Bidirecional



Agregação



Composição



Generalização/Herança

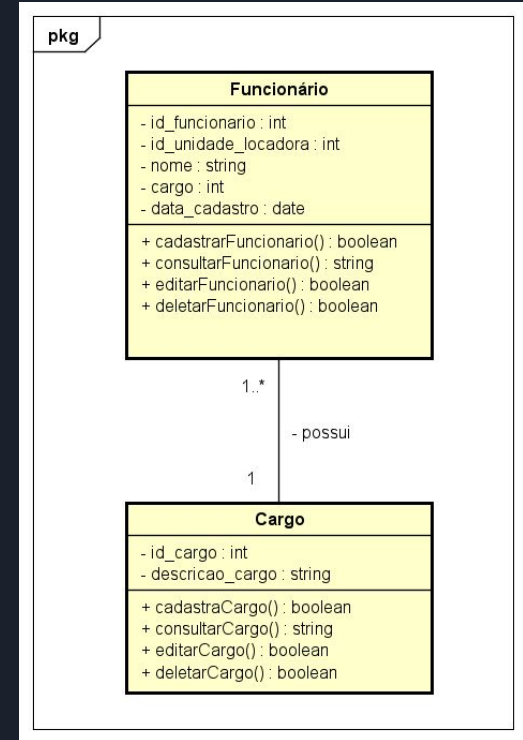
# 6.1. Elementos do Diagrama de Classes



Execução



Dependência



# 6.1. Elementos do Diagrama de Classes

## Multiplicidade

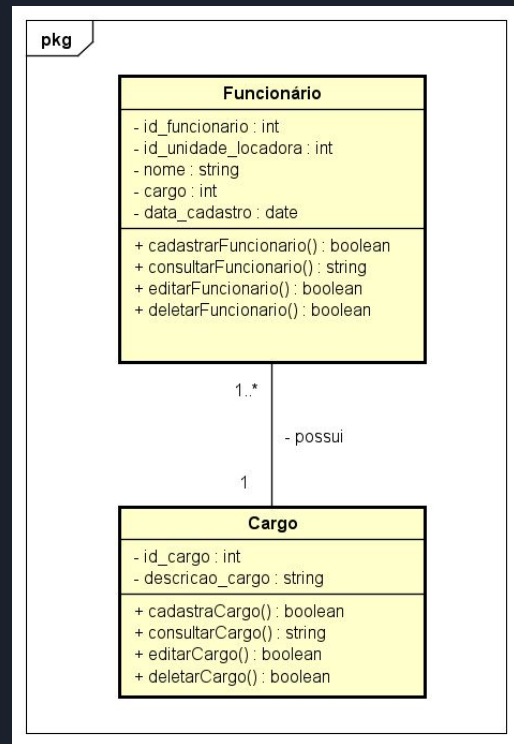
Representa o número de instâncias de uma classe que podem se associar a instâncias de outra classe em um relacionamento.

0..1: Zero ou uma instância.

1: Exatamente uma instância.

0..\* ou \*: Zero ou mais instâncias.

1..\*: Uma ou mais instâncias.





# 7. Diagrama de Sequência



# 7.1. Diagramas de Interação

- Modelos que descrevem como grupos de objetos colaboram para a realização de algum comportamento.
- Um diagrama de interação captura o comportamento de um único **caso de uso** por vez.
- O diagrama mostra vários **objetos** e as **mensagens** que são trocadas entre esses objetos.

Diagrama de Comunicação

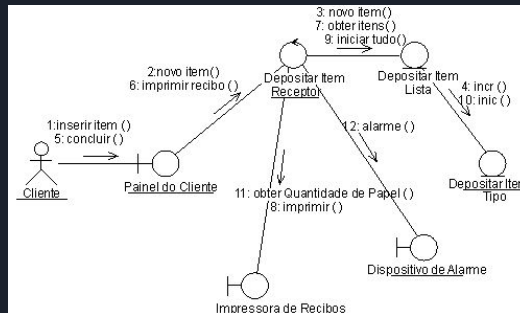
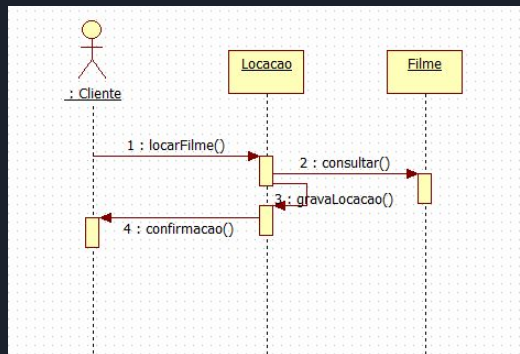


Diagrama de Sequência



# 7.1. Diagramas de Interação

- Representam como o sistema age internamente para que um ator atinja seu objetivo na realização de caso de uso.
- Documentam os **aspectos dinâmicos** do sistema.
- Mostra a interação entre os objetos tendo em vista as **mensagens trocadas** para a execução de **cenários dos casos de uso do sistema**.

Diagrama de Comunicação

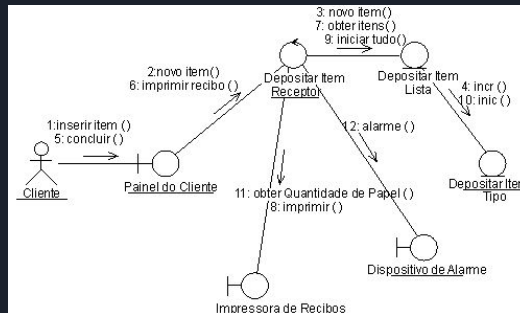
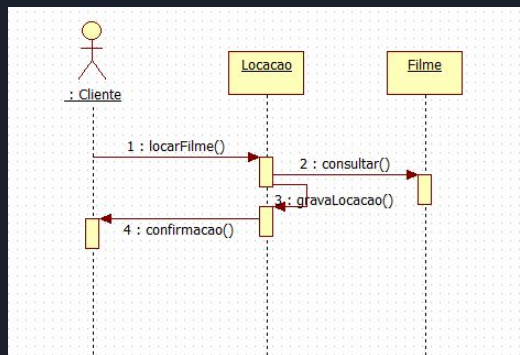


Diagrama de Sequência

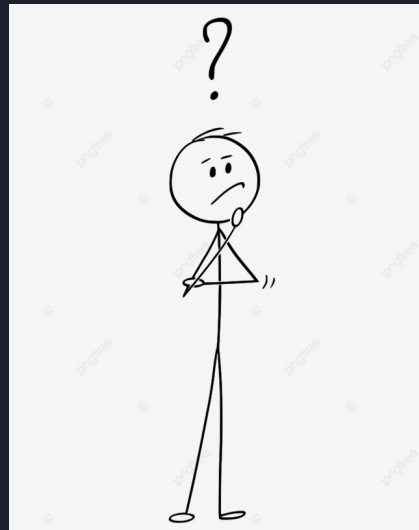


# 7.1. Diagramas de Interação

**Por que utilizar os diagramas de interação?**

**Para responder as seguintes perguntas:**

- Como as operações do sistema são executadas internamente?
- A que classes estas operações pertencem?
- Quais objetos participam da realização de um caso de uso?
- De que forma os objetos colaboram para que um determinado caso de uso seja realizado?
- Em que ordem as mensagens são enviadas durante esta realização?
- Que informações precisam ser enviadas de um objeto a outro?



# 7.1. Diagramas de Interação

Os principais tipos de diagramas de interação são:

## ❑ Diagrama de Sequência

- Ressalta a ordem cronológica das mensagens.
- Mensagens enviadas no decorrer do tempo.

## ❑ Diagrama de Comunicação

- Ressalta o relacionamento entre os objetos e a consequente troca de mensagem entre eles.

Diagrama de Sequência

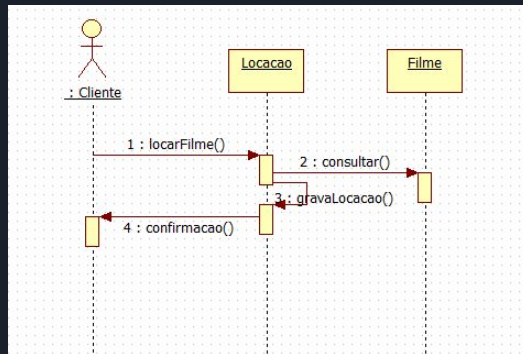
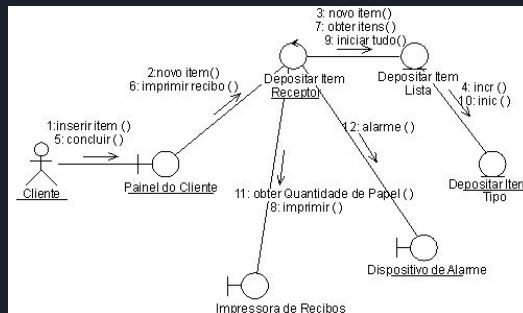


Diagrama de Comunicação





## 7.2. Conceitos Básicos

### Mensagens

- ▢ O princípio básico da interação entre objetivos é o conceito de **mensagem**.
- ▢ Um sistema OO pode ser visto como uma rede de objetos.
- ▢ Funcionalidades são realizadas pelos objetos, **que só podem interagir por meio do envio de mensagens**.



## 7.2. Conceitos Básicos

### Mensagens

- ▮ Uma mensagem representa a requisição de um **objeto remetente** a um **objeto receptor** para que este último **execute alguma operação** definida para a sua classe.
- ▮ Essa mensagem deve conter **informação** suficiente para que a operação do objeto receptor possa ser executada.



## 7.2. Conceitos Básicos

### Mensagens

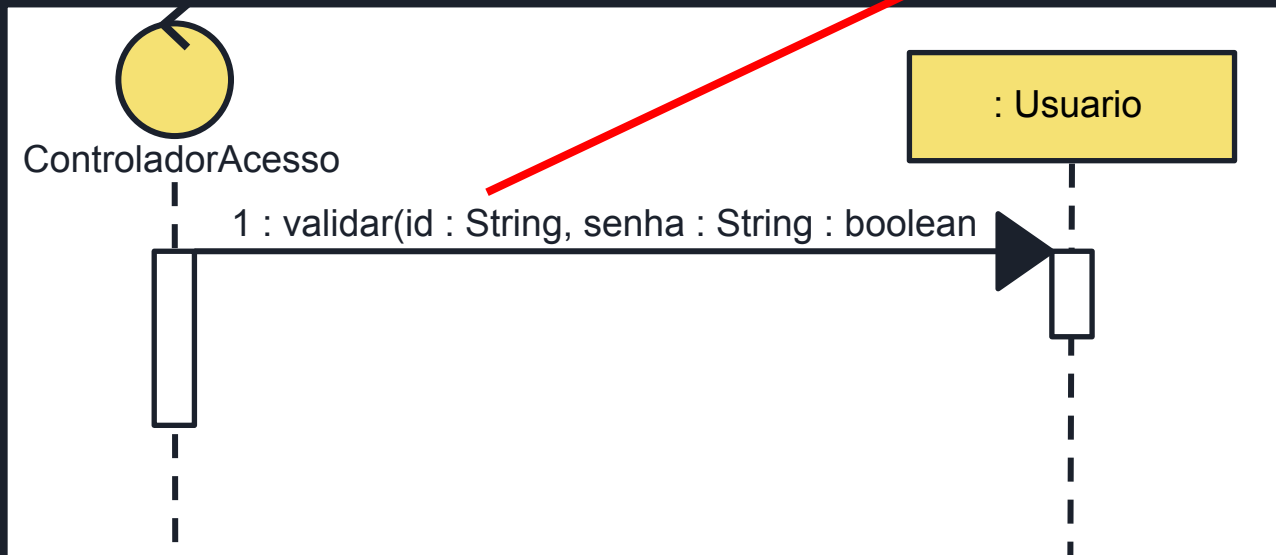
- ▣ O fato de um objeto “precisar de ajuda” indica sua necessidade de trocar mensagens.
- ▣ Na construção dos diagramas de interação, **mensagens de um objeto a outro implicam em operações que classes dos respectivos objetos devem possuir.**



## 7.2. Conceitos Básicos

### Mensagens

Usuario
- login : String - senha : String
+ validar(id : String, senha : String : boolean



## 7.2. Conceitos Básicos

### Elementos Básicos

#### ❑ **Objetos**

- ❑ São representados em um diagrama de interação utilizando-se a mesma notação gráfica do **Diagrama de Objetos**.
- ❑ Pode-se representar **objetos anônimos** ou **objetos nomeados**, dependendo da situação.



#### ❑ **Classes também podem ser representadas**

- ❑ Para o caso de mensagens enviadas para a classe em vez de ser enviada para um objeto.





## 7.2. Conceitos Básicos

### Elementos Básicos

**vendas : Departamento**

**Nome do Objeto e sua Classe  
(Objeto Nomeado)**

**: Departamento**

**Objeto Anônimo da  
Classe Departamento**

**Departamento**

**Classe Departamento**



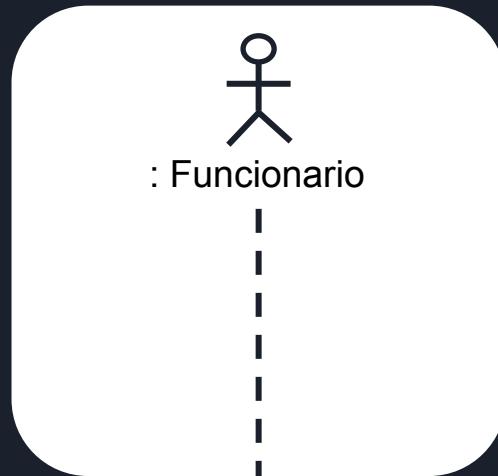
## 7.3. Diagrama de Sequência



## 7.3. Diagrama de Sequência

### Atores

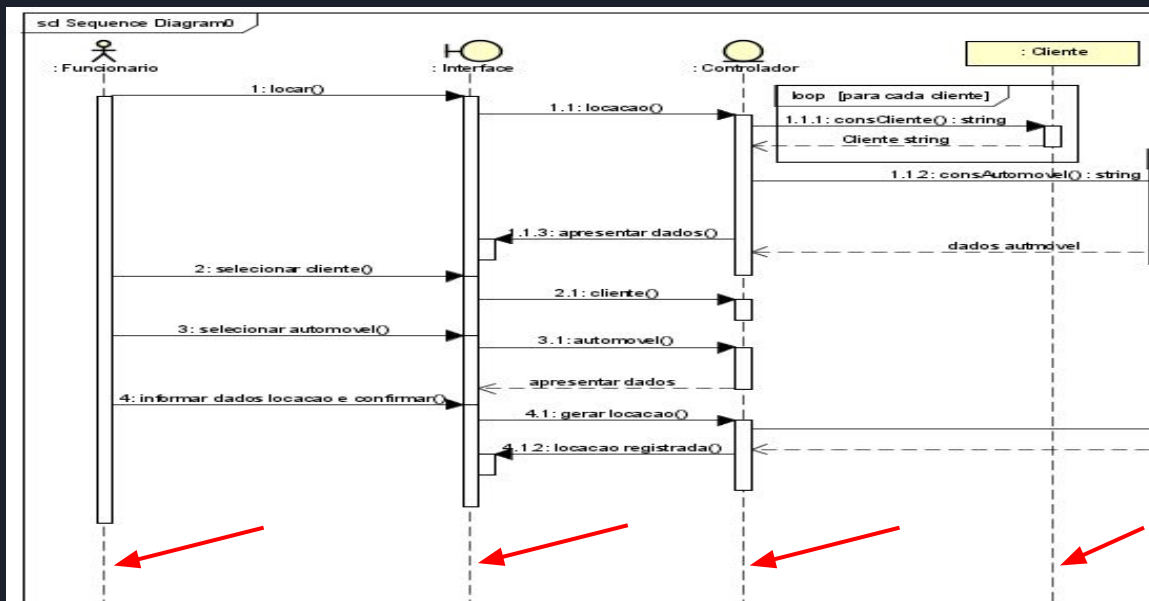
- Representam os atores declarados no Diagrama de Casos de Uso.
- Entidades externas que interagem com o sistema e que solicitam serviços.



## 7.3. Diagrama de Sequência

### Linha de vida

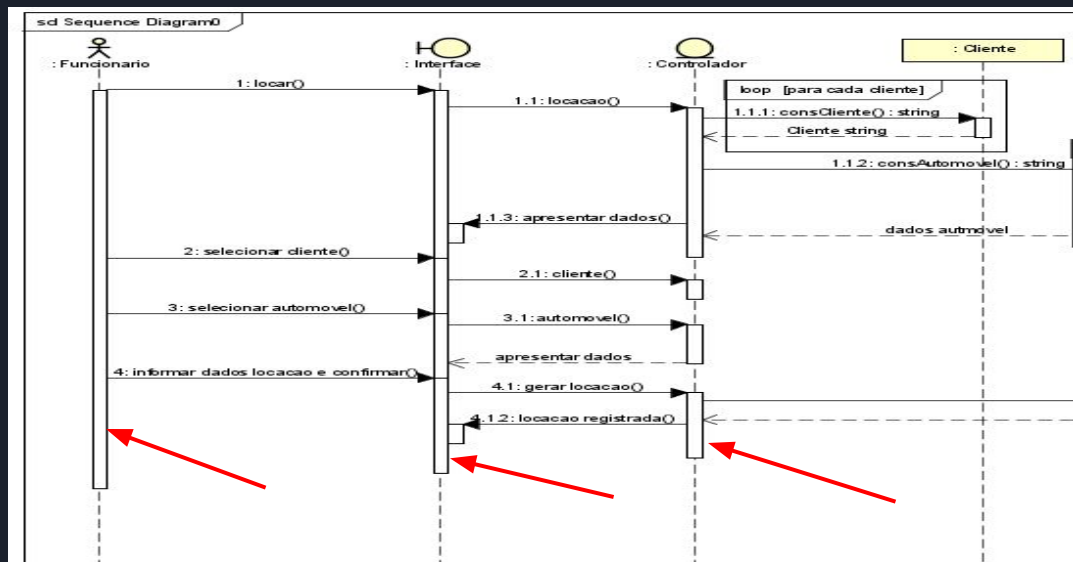
- Mostra o tempo de existência de um elemento na interação.



## 7.3. Diagrama de Sequência

### Foco de controle (ou ativação)

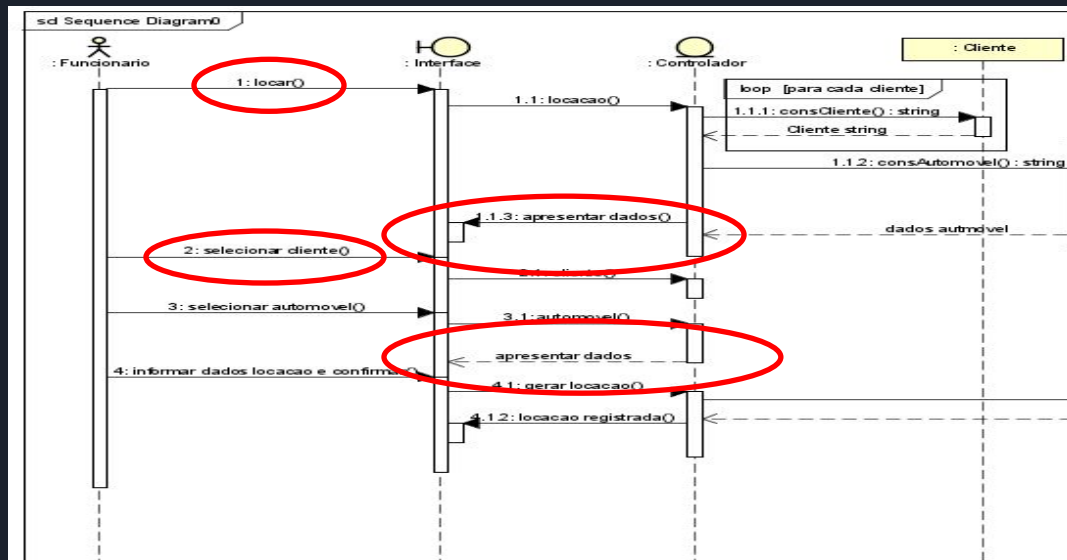
- É o tempo que um método leva para executar suas tarefas.
- Compreende todas as mensagens que se originam do método em execução



## 7.3. Diagrama de Sequência

### Mensagens

- Representadas por setas, indicam qual componente enviou a mensagem e qual recebeu



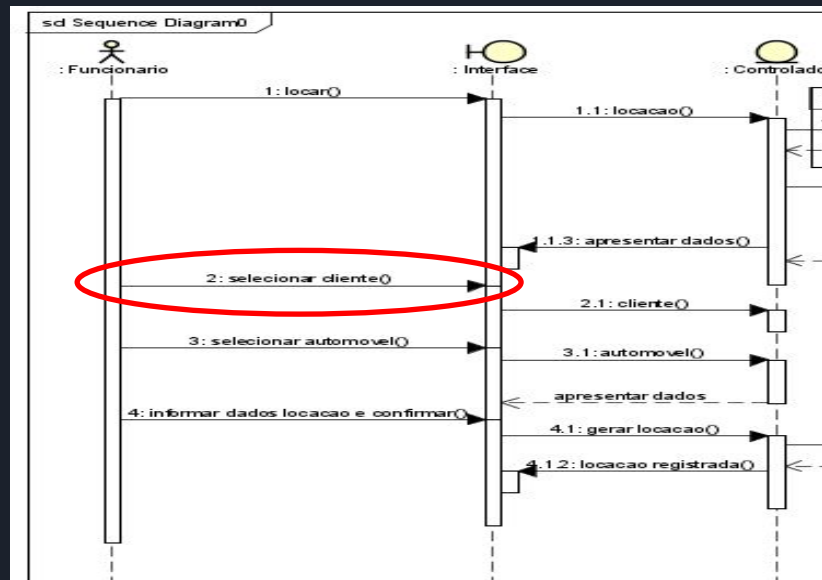


## 7.3. Diagrama de Sequência

### Tipos de Mensagens

#### 1 - Mensagem Síncrona

#### 2 - Mensagem Assíncrona

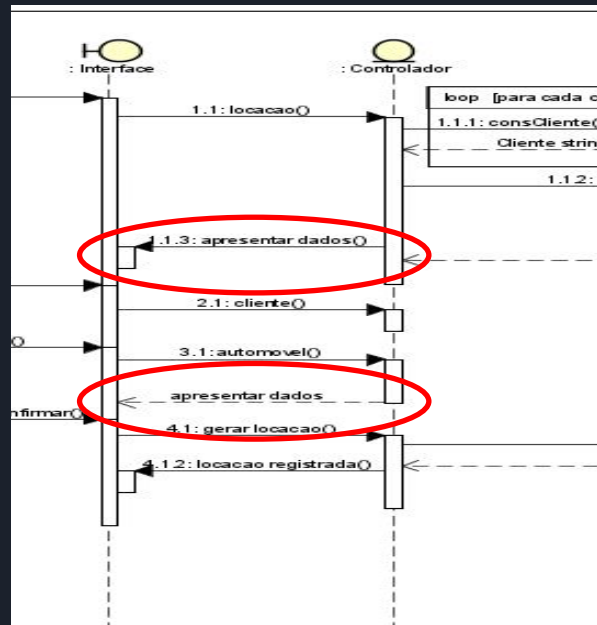


## 7.3. Diagrama de Sequência

### Tipos de Mensagens

3 - Mensagem Reflexiva

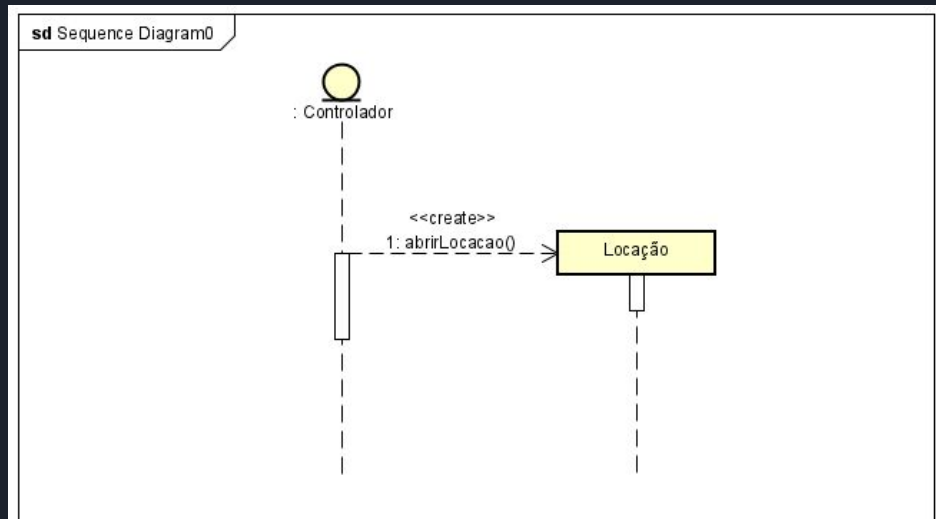
4 - Mensagem de Retorno



## 7.3. Diagrama de Sequência

### Tipos de Mensagens

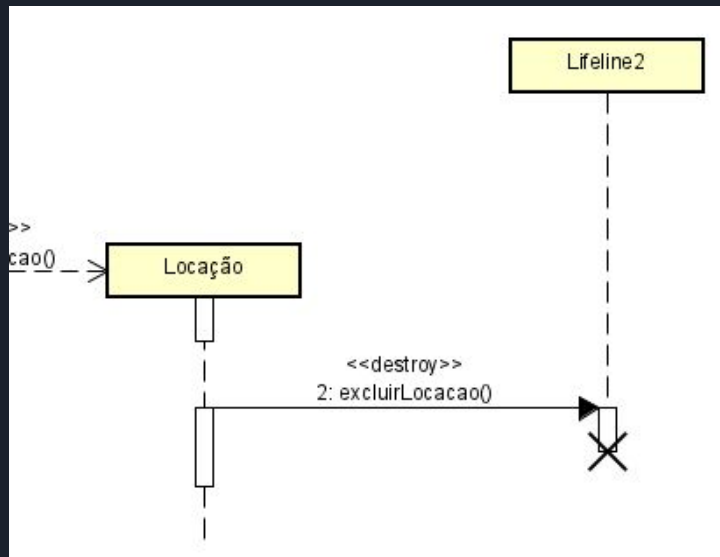
#### 5 - Mensagem de Criação



## 7.3. Diagrama de Sequência

### Tipos de Mensagens

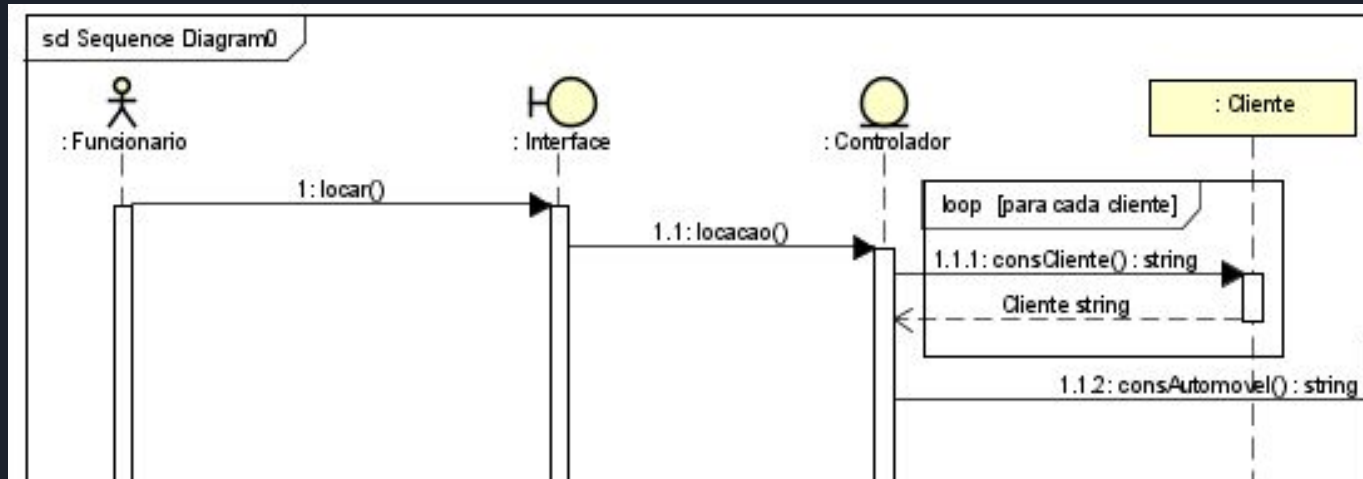
#### 6 - Mensagem de Destruição



## 7.3. Diagrama de Sequência

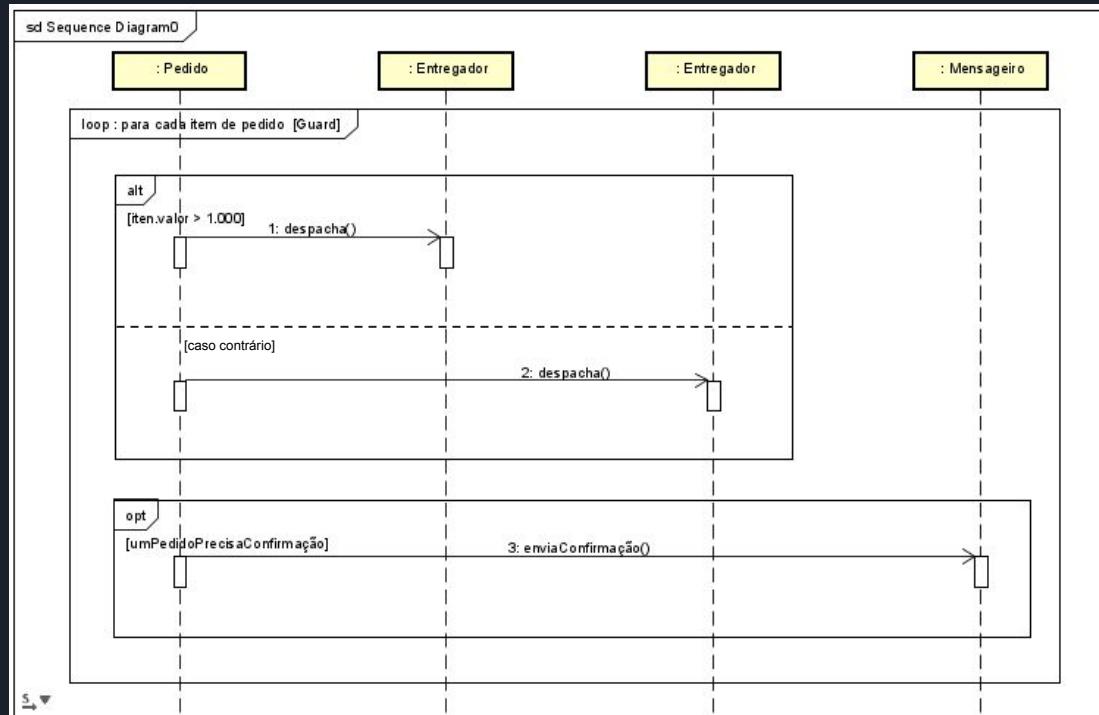
### Estereótipos

**<<Boundary>>, <<Control>>, <<Entity>>**



## 7.3. Diagrama de Sequência

### Frames de interação





## 7.3. Diagrama de Sequência

### Quando utilizar?

- Para observar o comportamento de vários objetos dentro de um caso de uso.
- Para ajudar a validar os métodos do diagrama de classes.

## 8. Diagrama de Atividades







## 8.1. O que é um Diagrama de Atividade?

- Diagrama comportamental que representa graficamente o fluxo de controle de uma atividade para outra, com descrição de ações passo-a-passo em um sistema.
- Especifica a transformação de entradas em saídas por meio de uma sequência controlada de ações.
- Semelhante a um fluxograma, porém com suporte ao paralelismo e sincronismo de atividades.
- Variação do diagrama de estados, que permite modelar comportamento baseado em fluxo.



## 8.2. Conceitos-chave

### Atividade

É um processo de negócio, como por exemplo, o cadastro de um veículo no sistema. Muitas vezes descreve a implementação de um caso de uso.

### Ação

É um passo individual dentro de uma atividade, como por exemplo a adição de uma informação em um campo de busca dentro do sistema.




## 8.3. Para que serve um Diagrama de Atividade?

- Mostrar interações entre objetos
- Expressar como as ações são executadas
- O que cada ação faz
- Sequência que as ações são executadas
- Onde as ações são executadas
- Quem realiza as ações
















## 8.4. Elementos de um Diagrama de Atividade

- **Nó inicial:** Ponto de início de uma atividade.
- **Fluxo / Aresta:** Descreve a sequência na qual as atividades se realizam. Conexões entre duas ações. Representada por uma seta.
- **Decisão:** Um único fluxo de entrada e vários fluxos de saída. Cada fluxo de saída possui uma condição booleana.
- **Intercalação:** Vários fluxos de entrada e uma única saída. Marca o final de um condicional iniciado por uma decisão.

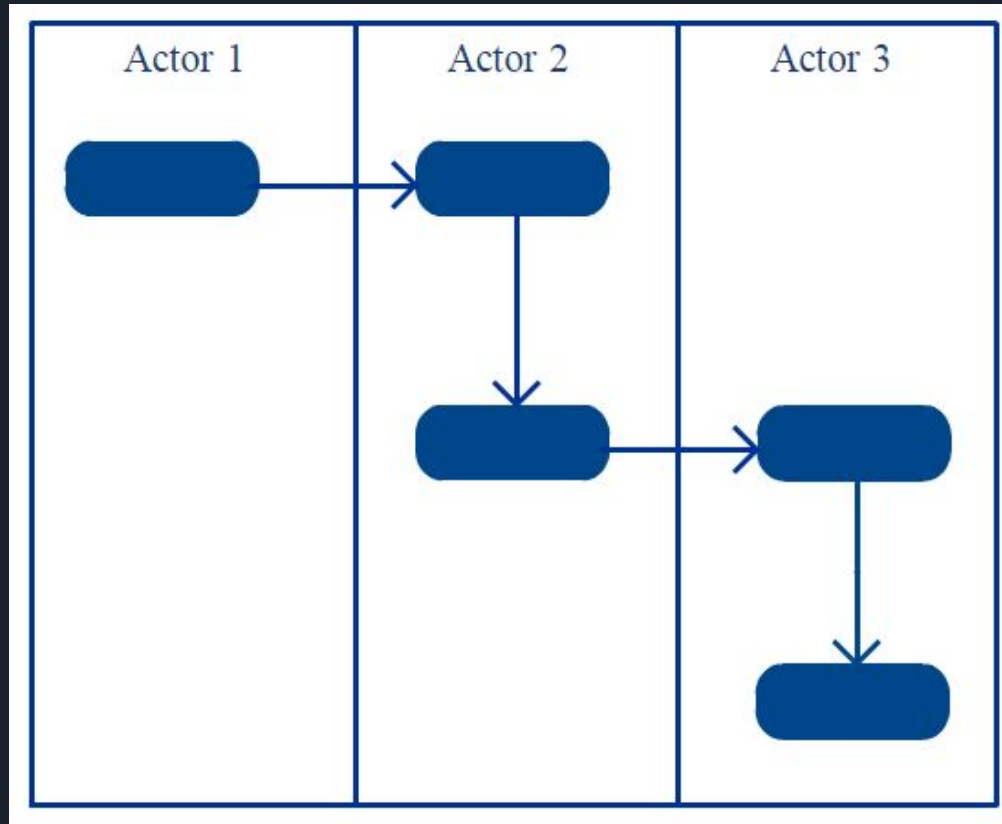
- 
- **Divergência / Fork:** Ponto no qual duas ou mais tarefas podem iniciar em paralelo
  - **Convergência / join:** Ponto no qual duas ou mais tarefas paralelas se unem para dar início a uma nova tarefa única
  - **Nó Final de Atividade:** Ponto onde termina a atividade
  - **Partições:** Mostra quem realiza cada ação ou conjunto de ações
  - **Sinais / Mensagens:** Envio ou recebimento de sinais ou mensagens por uma ação

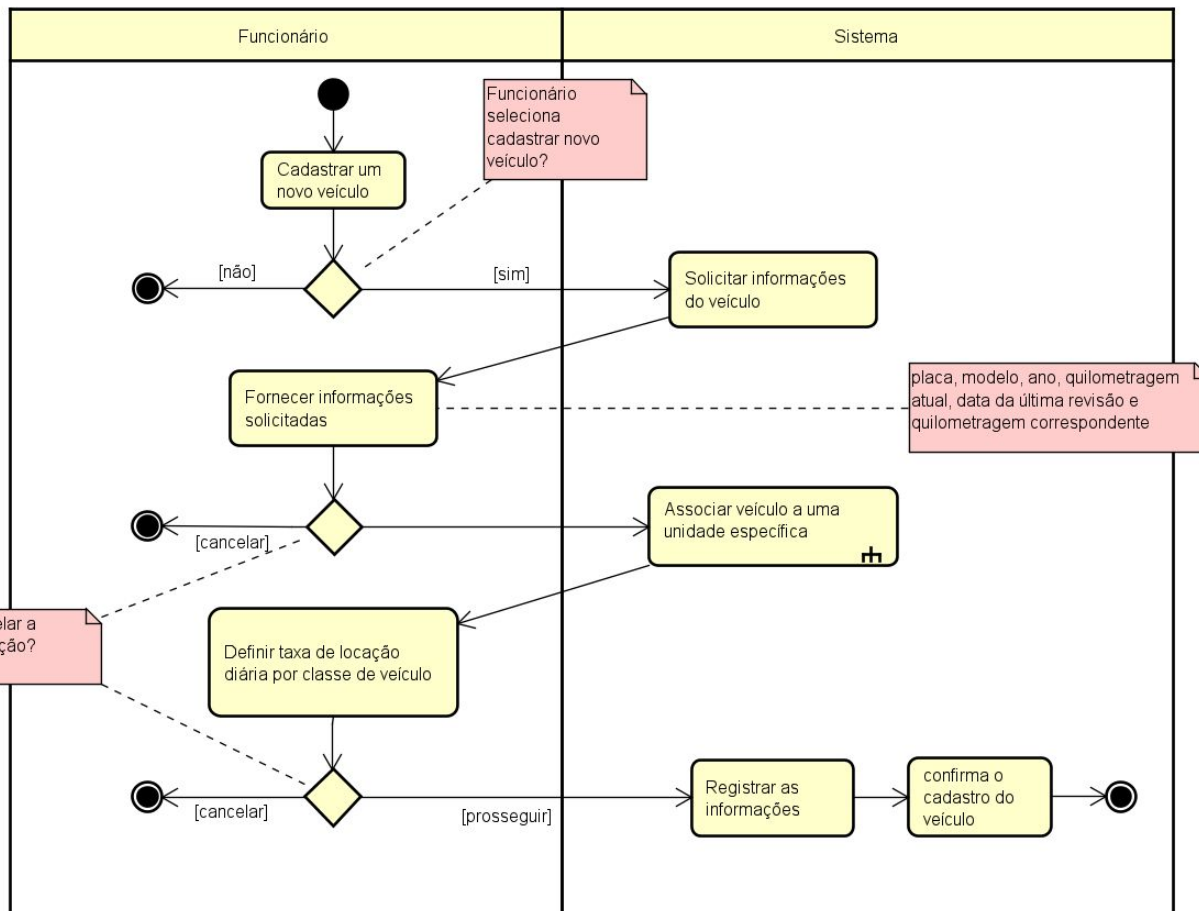
## 8.5. Símbolos comuns

	Start/ Initial Node
	Activity / Action State
	Action
	Control Flow / Edge
	Object Flow / Control Edge
	Activity Final Node
	Flow Final Node
	Decision Node

	Fork
	Merge
	Signal Sending
	Signal Receipt
	Note/ Comment

## 8.6. Partições







## 9. Conclusão



## 9. Conclusão

Em conclusão, os diagramas de classes, de sequência e de atividades desempenham papéis cruciais na modelagem e desenvolvimento de software. Ao proporcionar uma representação visual clara da estrutura, interações e fluxos do sistema, esses diagramas são fundamentais para a compreensão, comunicação eficaz entre equipes e orientação precisa no processo de desenvolvimento. A modelagem, facilitada por esses diagramas, não apenas aprimora a organização interna, mas também é essencial para assegurar a entrega de sistemas eficientes, alinhados com as expectativas do usuário e prontos para enfrentar os desafios do desenvolvimento de software.





# Referências

1. IBM. Diagramas de caso de uso. IBM Rational Software Modeler, Versão 7.5.0. Disponível em: <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case>. Acesso em: 15, novembro de 2023.
2. IBM. Diagramas de Classe. IBM Rational Software Modeler, Versão 7.5.0. Disponível em: <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams>. Acesso em: 15, novembro de 2023.
3. IBM. Diagramas de Atividades. IBM Rational Software Modeler, Versão 7.5.0. Disponível em: <https://www.ibm.com/docs/pt-br/rational-soft-arch/9.7.0?topic=diagrams-activity>. Acesso em: 15, novembro de 2023.
4. IBM. Diagramas de Sequência. IBM Rational Software Modeler, Versão 7.5.0. Disponível em: <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=uml-sequence-diagrams>. Acesso em: 15, novembro de 2023.
5. [https://conteudo.colaboraread.com.br/202002/INTERATIVAS\\_2\\_0/ANALISE\\_ORIENTADA\\_A\\_OBJETOS/LIVRO\\_DIGITAL/npf\\_u4s2.html](https://conteudo.colaboraread.com.br/202002/INTERATIVAS_2_0/ANALISE_ORIENTADA_A_OBJETOS/LIVRO_DIGITAL/npf_u4s2.html)

**Obrigado pela a  
atenção!**

