



Quem se prepara, não para.

Análise de Sistemas

3º período

Professora: Michelle Hanne

UML (*Unified Modeling Language*)

- UML é uma linguagem de notação padrão da OMG (*Object Management Group*)¹ para:

Visualização

- Diagramas que facilitam a comunicação entre membros da equipe.

Especificação

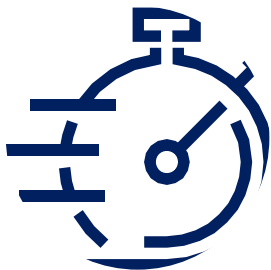
- Apresenta diferentes aspectos de arquitetura e comportamento.

Documentação

- Construção e documentação de software orientado a objetos.

1- Consórcio internacional de empresas que define padrões na área de orientação a objetos.

UML (*Unified Modeling Language*)



Modelagem de Negócios e
Processos



Análise, design e
implementação de
Softwares

UML – Linha do Tempo

1994

- James Rumbaugh (OMT) se juntou a Grady Booch (Booch) na Rational Software com objetivo de unificar os métodos

1995

- Lançado um esboço da versão 0.8 **Unified Process** - Processo Unificado
- Ivair Jacobson se associou à Rational e o método OOSE é incorporado

1996

- Surge a versão 0.9 da UML

1997

- Lançada a versão 1.1 adotada pela OMG (**Object Management Group**)

2000

- Versão 1.2 com várias alterações no metamodelo, na semântica e na notação

2005

- versão 2.0 adiciona novos diagramas como objetos, pacotes e estruturas

2015

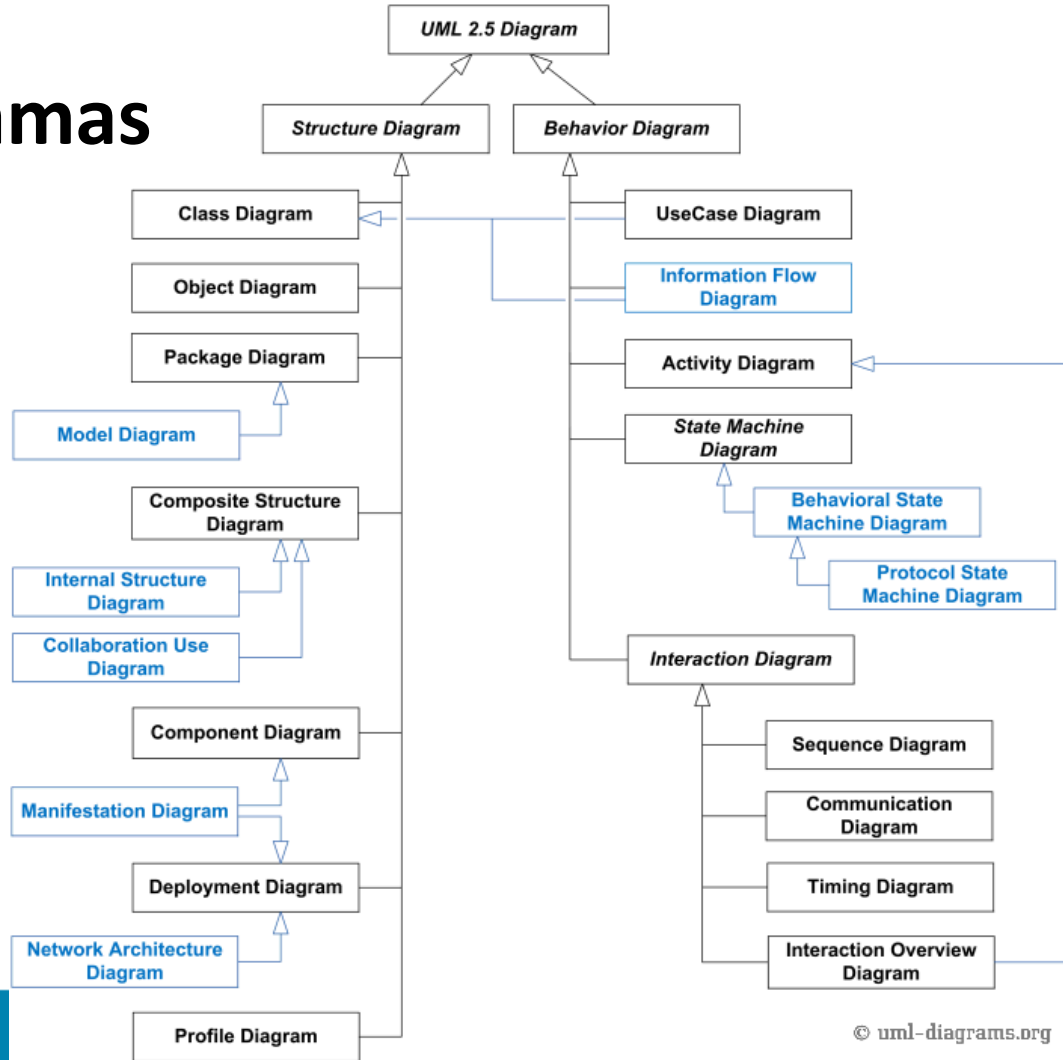
- Versão 2.5 com poucas alterações, o documento de especificação da UML foi reescrito

2017

- Versão atual 2.5.1



Diagramas UML



Visão geral dos diagramas da UML 2.5. **Fonte:**
<https://www.uml-diagrams.org/uml-25-diagrams.html>

*Os itens em azul não fazem parte da taxonomia oficial dos diagramas da UML 2.5.

Tipos de Diagramas UML

Os diagramas de Estrutura mostram a estrutura estática do sistema e suas partes em diferentes níveis de abstração e implementação e como essas partes se relacionam.

Os diagramas de Comportamento mostram o comportamento dinâmico dos objetos em um sistema, que pode ser descrito como uma série de alterações no sistema ao longo do tempo .

Principais Diagramas UML

Principais Diagramas UML, segundo Pressman e Maxim (2016):

- **Diagrama de caso de uso:** exibe as funcionalidades do sistema pela visão do usuário e a interação do usuário no sistema.
- **Diagrama de classes:** mostra a visão estática ou estrutural do sistema. Composto pelas classes, seus atributos, operações e relacionamentos.
- **Diagrama de objetos:** representa classes com visão estática do sistema, assim os objetos mostram os valores armazenados pelos objetos das classes em um dado momento de execução do sistema.
- **Diagrama de atividades:** mostra o comportamento dinâmico do sistema por meio de fluxos de controle, podendo mostrar também fluxos concorrentes.
- **Diagrama de sequência:** mostrada a comunicação dinâmica dos objetos no momento de execução, quando um determinado objeto é executado em ordem temporal.
- **Diagrama de estado:** O diagrama de estado procura exibir como essas variáveis de controle trabalham ao longo do processo do sistema.

Ferramentas Case

Permitem a criação de diagramas de forma simples em um ambiente de fácil utilização.

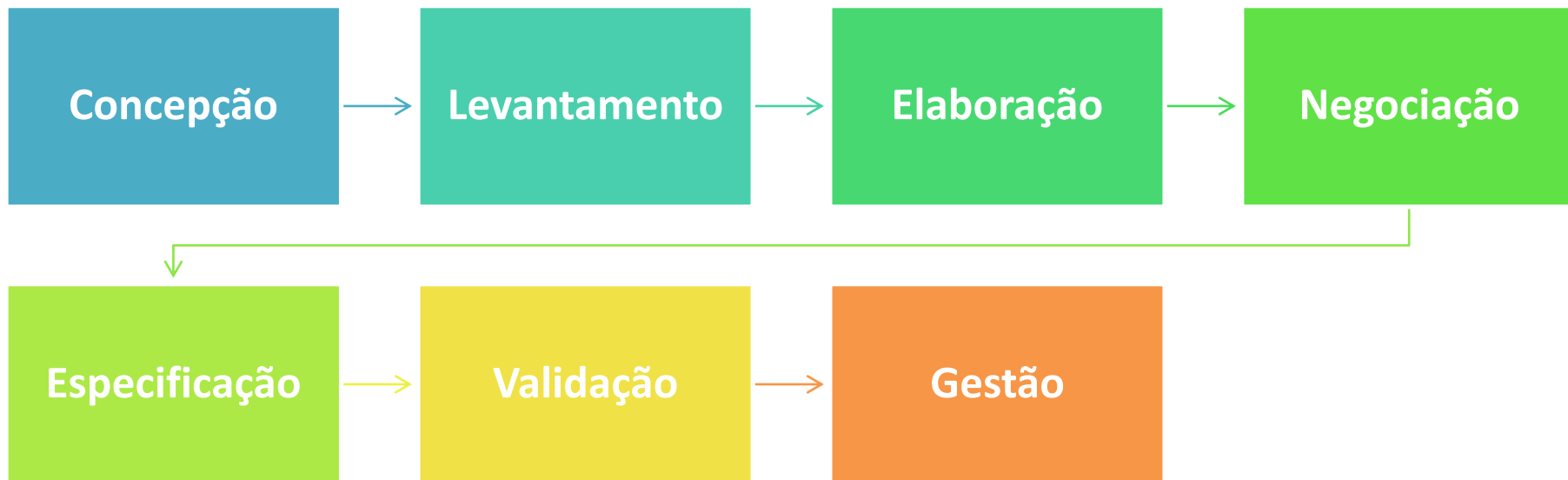
Exemplos: [Star UML](#), [Astah](#), [ERwin Data Modeler](#), [Draw.io](#), [Argo UML](#), [Lucidchart](#).

Documento de Visão Geral de Requisitos

O documento de visão define o escopo de alto nível e o propósito de um programa, produto ou projeto. Uma instrução clara do problema, solução proposta e os recursos de alto nível do produto ajudam a estabelecer expectativas e a reduzir riscos.

Estrutura de Tópicos recomendo pela IBM -
https://www.ibm.com/support/knowledgecenter/pt-br/SSYMRC_6.0.3/com.ibm.rational.rrm.help.doc/topics/r_vision_doc.html

Atividades da Engenharia de Requisitos



As Atividades não precisam ser realizadas exatamente de forma sequencial.

Etapas de Levantamento de Requisitos

Obtenção de requisitos: ocorre a interação com as partes interessadas (*stakeholders*) para coletar informações sobre os requisitos.

Classificação e organização de requisitos: se agrupam os requisitos relacionados, e esses são organizados de forma coerente.

Priorização e negociação de requisitos: é usada no momento em que se têm requisitos no sistema que começam a ser conflitantes entre si. Nesse momento, o analista deve procurar negociar os requisitos para evitar conflitos e também para priorizá-los.

Documentação de requisitos: os requisitos levantados são documentados, podendo ser documentos formais ou informais, passando por um ciclo iterativo.

The Scrum Guide

Scrum Team:

- Product Owner
- Development Team
- Scrum Master

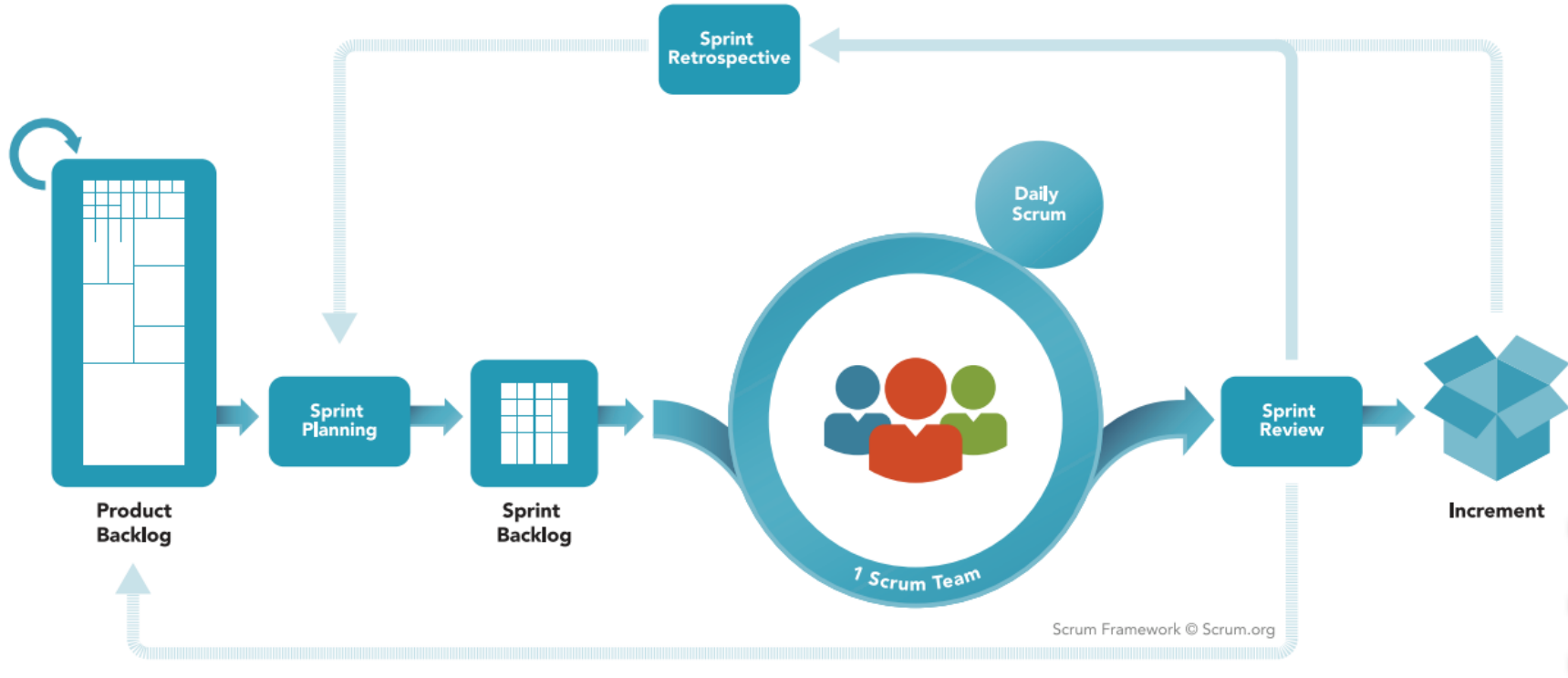
Artefatos:

- Product Backlog
- Sprint Backlog
- Increment

Eventos:

- Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Scrum Framework



Como Levantar os Requisitos

Pontos de vista: a captura de requisitos por diversos pontos de vista, através de workshops por exemplo.

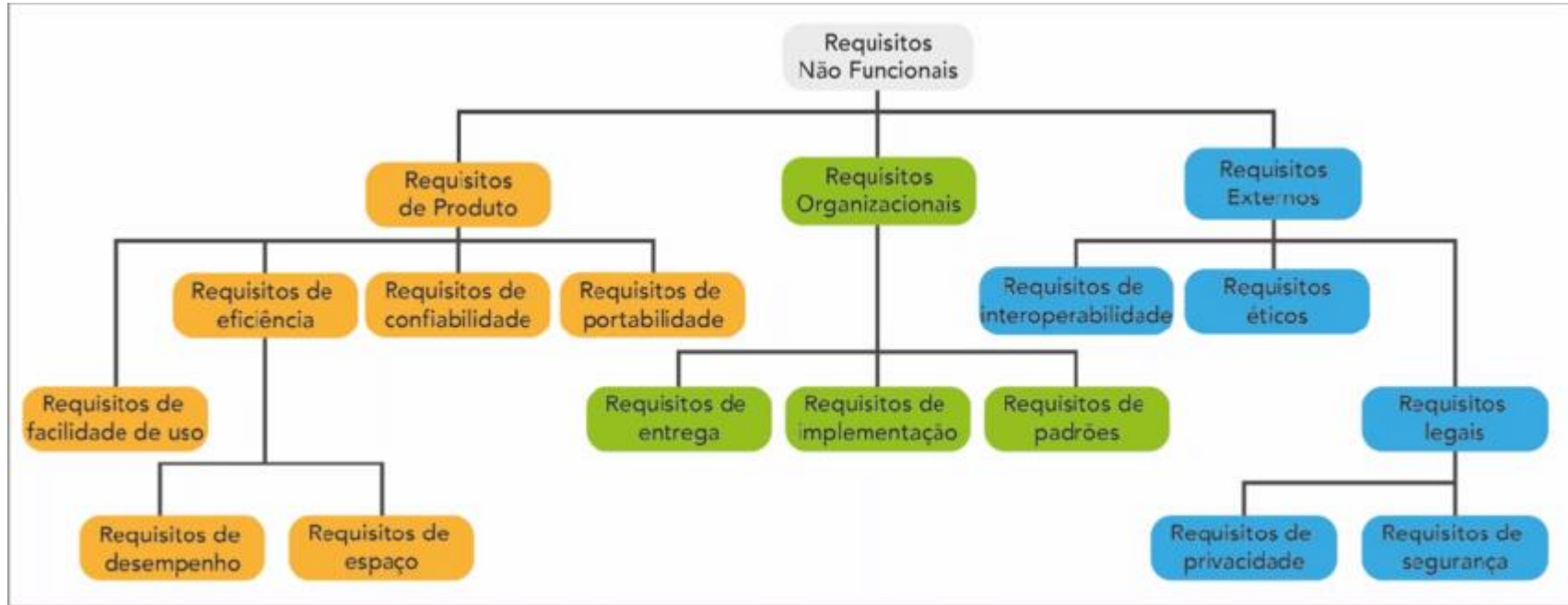
Entrevistas: podem ser feitas com um roteiro predefinido ou de uma forma exploratória. Podem ocorrer em grupo ou individuais.

Cenário: exemplos reais que representam a interação com o sistema. Desenho de Fluxo ou Processo.

Caso de uso: identifica ações e atores que irão interagir com o sistema.

Etnografia: baseada na observação com o usuário, “o sombra”.

Requisitos Não Funcionais



Tipos de Requisitos Não Funcionais. Fonte: Sommerville (2011) - https://www.researchgate.net/figure/Figura-1-Tipos-de-Requisitos-Nao-Funcionais-Fonte-Sommerville-2011-Os-RNFs-que-vem_fig1_311459854

Requisitos Não Funcionais

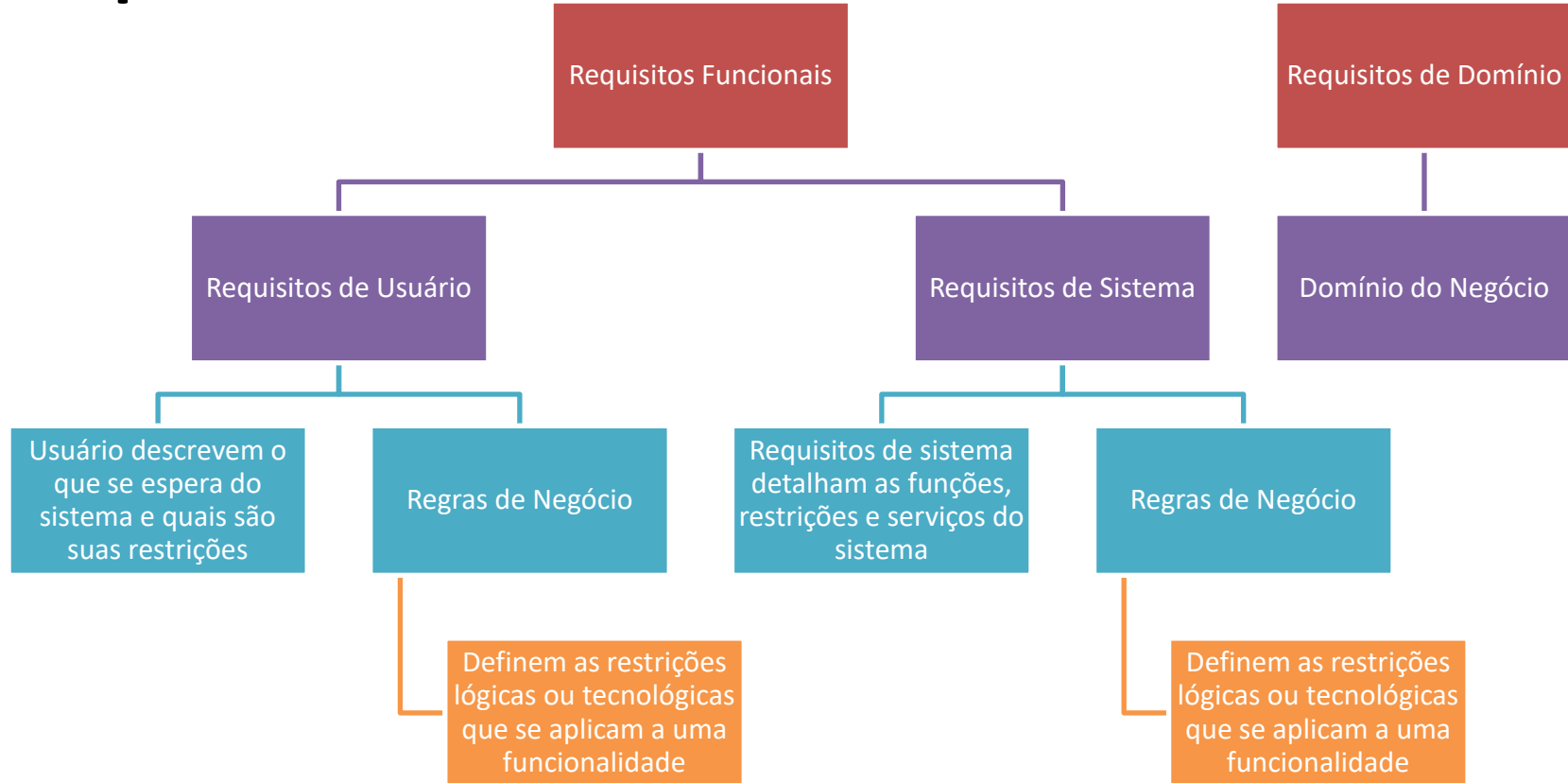
Requisitos de produto: definem o comportamento do produto, como, por exemplo, requisitos de velocidade, quanto de memória ele requer, quantidade de falhas aceitáveis, portabilidade e usabilidade.

Requisitos organizacionais: são definições das políticas do cliente ou do desenvolvedor, como padrões de processos e linguagem de programação.

Requisitos externos: como o próprio nome sinaliza, esses requisitos são definidos por outros, como sistemas no qual o produto irá interagir (interoperabilidade), conformidade com a lei e questões éticas.

- Desempenho
- Usabilidade
- Confiabilidade
- Segurança
- Disponibilidade
- Manutenção
- Tecnologias

Requisitos Funcionais



Caso de Uso Descritivo

Os casos de uso são definidos sob o ponto de vista de um ator. Ator é um papel que as pessoas (usuários) ou dispositivos desempenham ao interagir com o software. (Pressman, 2016.)

Existem diferentes templates de casos de uso descritivos.

Caso de Uso Descritivo

Caso de Uso	Título de Caso de Uso
Ator primário	Quem o ou que inicia a interação com o sistema
Ator(es) Secundário(s)	Outros atores envolvidos no sistema
Visão Geral	Descrever resumidamente qual objetivo/meta do caso de uso
Atributos	Lista dos atributos necessários para este caso de uso
Precondição	Condições iniciais que acionam o caso de uso
Fluxo Principal	Passo a passo do Cenário Principal
Fluxo Alternativo	Passo a passo do Cenário Alternativo
Fluxo de Exceção	Descrição detalhada da exceção referente ao Cenário Principal
Pós Condições	Condições finais que finalizam o caso de uso

Caso de Uso Descritivo - Exemplo

Caso de Uso	Emprestar Livro
Ator primário	Atendente (bibliotecário)
Ator(es) Secundário(s)	Leitor (aluno)
Visão Geral	A Atendente da biblioteca realiza o empréstimo de um ou mais livros a um leitor apto a emprestar livros. O empréstimo é válido por um determinado período de tempo, de acordo com o tipo de leitor. Os livros são levados pelo leitor, depois de devidamente desmagnetizados e marcados como emprestados.
Atributos	Código do Leitor, Situação (Apto ou Pendente), Id do Livro, Data de Entrega
Precondição	O Atendente está identificado e autenticado.

Caso de Uso Descritivo - Exemplo

Fluxo Principal

1. O Leitor chega ao balcão de atendimento da biblioteca e diz ao Atendente que deseja emprestar um ou mais livros da biblioteca.
2. O Atendente seleciona a opção para realizar um novo empréstimo.
3. O Atendente solicita ao leitor sua carteira de identificação, seja de estudante ou professor.
4. O Leitor fornece sua carteira de identificação.
5. O Atendente informa ao sistema a identificação do leitor.
6. O Sistema exibe o nome do leitor e sua situação.
7. O Atendente solicita os livros a serem emprestados.
8. O Leitor entrega os livros para a Atendente.
9. A Atendente informa ao sistema o código de identificação dos livros.
10. O Sistema informa a data de devolução de cada livro.
11. O Atendente desbloqueia os livros para que possam sair da biblioteca.
12. O Leitor sai com os livros.

Caso de Uso Descritivo - Exemplo

Fluxo Alternativo

4. O Leitor informa ao Atendente que esqueceu a carteira de identificação.
5. O Atendente solicita o número de identificação ao Leitor.
6. O Leitor fornece o número de identificação ao Atendente
7. O Atendente busca o Leitor no sistema pelo número de identificação.
8. O Sistema exibe informações sobre o Leitor.
9. A Atendente confirma com o Leitor algumas informações pessoais para garantir que ele é mesmo quem diz ser. Continua no 7 do Fluxo Principal.
10. O Sistema informa que o Livro não pode ser emprestado, pois está reservado.
11. O Atendente informa ao Leitor que não poderá emprestar um dos livros. Cancelar toda a operação (se for o único livro), senão continua no 11 do Cenário de Sucesso Principal.

Caso de Uso Descritivo - Exemplo

Fluxo de Exceção	(2-11). A qualquer momento, o Leitor informa ao Atendente que desistiu do empréstimo. Cancelar toda a operação 6. O Sistema informa que o Leitor está impedido de fazer empréstimo, por ter uma situação irregular. Cancelar toda a operação.
Pós Condições	Os dados do novo empréstimo são armazenados no sistema. Os livros emprestados possuem status “emprestado”.

Baseado em: https://edisciplinas.usp.br/pluginfile.php/1906994/mod_resource/content/1/Aula03_DescricaoCasosDeUso.pdf

Referências

PRESSMAN, Roger S. MAXIM, Bruce R. **Engenharia de Software - Uma Abordagem Profissional**. 8.ed. Porto Alegre: Amgh Editora, 2016. 968p. ISBN 9788580555332.

SOMMERVILLE, Ian. **Engenharia de software**. 8.ed. São Paulo: A. Wesley publishing company, 2010.