

# ATIVIDADES DA ENGENHARIA DE REQUISITOS DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

## ENGENHARIA DE SOFTWARE



Vamos iniciar nosso estudo sobre o processo de desenvolvimento de software, no contexto da Engenharia de Software, que tem como principal produto o **software**. Aproveito para destacar que a bibliografia Pressman (2016) é uma referência mundial nesta área.

Imaginando um software embarcado em uma aeronave com centenas de pessoas ou controlando o tráfego aéreo de uma grande cidade, podemos destacar uma característica comum: a **complexidade**. A melhor tratativa para a complexidade é a aplicação de metodologia que permita a decomposição do problema em problemas menores de forma sistemática, cabendo à **Engenharia de Software** essa sistematização.

Entretanto, diferentemente de produtos de outras engenharias em produção, o software possui alta volatilidade, em função de constantes evoluções na tecnologia e nos requisitos, agregando a ele uma complexidade adicional.

A Engenharia de Software é uma tecnologia em camadas. Vejamos as descrições das referidas camadas na tabela a seguir:

Camada	Descrição
Camada qualidade	Garante que os requisitos atendam às expectativas dos usuários.
Camada de processo	Define as etapas de desenvolvimento do software.
Camada de métodos	Determina as técnicas e os artefatos de software.
Camada ferramentas	Estimula a utilização de ferramentas CASE.

Importante destacar que a base da Engenharia de Software é a camada de processo que trata das etapas de desenvolvimento.

## PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Você sabe por que se aplica de forma intensa o conceito de abstração no desenvolvimento de software?

Porque esse processo é iniciado com especificações e modelos com alto nível de abstração e, à medida que o desenvolvimento de software se aproxima da codificação, o nível de abstração diminui, de modo que o código representa o nível mais baixo da abstração ou de maior detalhamento na especificação do software.

Os diferentes modelos de processos de desenvolvimento de software possuem as seguintes atividades típicas:

- **Levantamento de requisitos**
- **Análise**
- **Projeto**
- **Implementação**
- **Testes**
- **Implantação**

Vamos, agora, descrever cada uma das atividades comumente previstas em um processo de desenvolvimento de software.

## ENGENHARIA DE REQUISITOS

As etapas de levantamento de requisitos e análise, no processo de desenvolvimento de software, compõem a Engenharia de Requisitos, de modo que essa engenharia está no contexto da Engenharia de Software.

Neste momento, precisamos apresentar a conceituação de requisito:

**“Os requisitos de um sistema são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais. Esses requisitos refletem as necessidades dos clientes de um sistema que ajuda a resolver algum problema”.**

SOMMERVILLE, 2007

A Engenharia de Requisitos inclui as atividades de descobrir, analisar, documentar e verificar os serviços fornecidos pelo sistema e suas restrições operacionais, possuindo um processo próprio, tal qual ilustrado na Figura 1. Destacamos a existência de outras propostas de processos.

**Concepção**

**Levantamento**

**Elaboração**

**Negociação**

**Especificação**

**Validação**

**Gestão**

Figura 1 – Tarefas da Engenharia de Requisitos.

Você poderia perguntar: “Outro processo?”. Sim! Lembre-se de que a engenharia tem como base a camada de processos. Vamos entender cada etapa desse processo ilustrado na Figura 1

### **CONCEPÇÃO**

Essa etapa exige do engenheiro de software o estabelecimento do entendimento inicial do problema, a identificação das partes interessadas que serão atendidas pelo software, a natureza da solução desejada e a eficácia da comunicação e colaboração preliminares entre clientes e usuários com a equipe de projeto.

Cabe destacar que um software costuma ter vários tipos de usuários, como, por exemplo, as partes interessadas em diferentes níveis gerenciais de uma empresa.

### **LEVANTAMENTO**

Atividade que permite definir o escopo do projeto, ou “tamanho do problema”, além de possibilitar que usuários e desenvolvedores tenham a mesma visão do problema a ser resolvido. Isso é um desafio! Concorda?

Nesta etapa, é gerada uma especificação de requisitos que serve como um contrato entre clientes e equipe de projeto, esclarecendo aos clientes o que será entregue como produto do trabalho da equipe de desenvolvimento.

Ainda sobre a etapa de levantamento, esses clientes devem ser capazes de compreender os requisitos e fornecer *feedback* sobre eventuais falhas na especificação, para que estas sejam corrigidas de imediato, antes que o trabalho errado se propague pelo projeto.

A referida especificação comumente tipifica os requisitos em três categorias: requisitos funcionais, não funcionais e de domínio.

Os requisitos funcionais estão relacionados aos serviços fornecidos pelo sistema, ou seja, as funcionalidades que estarão disponíveis no software, tal como a geração de um histórico escolar em um sistema de gestão acadêmica.

Os requisitos não funcionais incluem as restrições operacionais impostas ao software, tais como o sistema gerenciador de banco de dados, a linguagem de programação, legislação pertinente a *compliance*, entre outros, bem como os requisitos de qualidade, e.g., confiabilidade, manutenibilidade, usabilidade etc.

Os requisitos de domínio também são conhecidos como “regras de negócio”, que normalmente apresentam-se como restrições ao requisito funcional. Como exemplo, temos o cálculo da média para aprovação em determinada disciplina, a contagem de pontuação de multas para cálculo da perda de uma carteira de motorista ou o cálculo dos impostos quando da geração de uma nota fiscal. O não cumprimento de um requisito de domínio pode comprometer o uso do sistema.

## Agora, vejamos as técnicas mais utilizadas para levantar requisitos.

A observação, ou etnografia, permite ao engenheiro de software imergir no ambiente de trabalho onde a solução será usada, observando o trabalho rotineiro e tomando notas das tarefas em execução nas quais as partes interessadas estão envolvidas.

A entrevista é uma forma de diálogo, formal ou informal, onde o entrevistador busca respostas para um conjunto de questões previamente definidas e os entrevistados se apresentam como fontes de informação.

A pesquisa consiste na aplicação de um questionário às partes interessadas e posterior análise das respostas. Essa técnica permite a rápida obtenção de informações quantitativas e qualitativas de um público-alvo numeroso, particularmente quando não estão em um único local físico.

O JAD, *Joint Application Design*, é um método de projeto iterativo que substitui as entrevistas individuais por reuniões de grupo, onde participam representantes dos usuários e dos desenvolvedores.

A técnica *brainstorming* inclui reuniões na qual participam todos os envolvidos na idealização do produto, como os engenheiros de software, clientes e usuários finais. Todos os envolvidos devem expor suas ideias e necessidades em relação ao produto.

## Você poderia imaginar uma entrega nesta etapa?

Temos uma entrega denominada “documento de requisitos”, cujo objetivo é documentar de forma fiel e completa todas as necessidades dos clientes e obter um aceite sobre o que se está propondo entregar em termos de produto.

A partir desse documento, inicia-se a rastreabilidade dos requisitos, garantindo que as especificações geradas até a codificação estejam de acordo com a documentação de requisitos.

### ELABORAÇÃO

Nesta etapa, os engenheiros de software realizam um estudo detalhado dos requisitos levantados e, a partir desse estudo, são construídos modelos para representar o sistema a ser construído.

Na etapa de elaboração, a modelagem é guiada pela criação e pelo refinamento de cenários, identificados a partir dos requisitos funcionais, que descrevem como os usuários interagem com o sistema. A modelagem de casos de uso da UML (*Unified Modeling Language*) representa os referidos cenários, incluindo diagramas de casos de uso, artefatos gráficos, e descrições de casos de uso, artefatos textuais. A Figura 2 ilustra um diagrama de caso de uso.

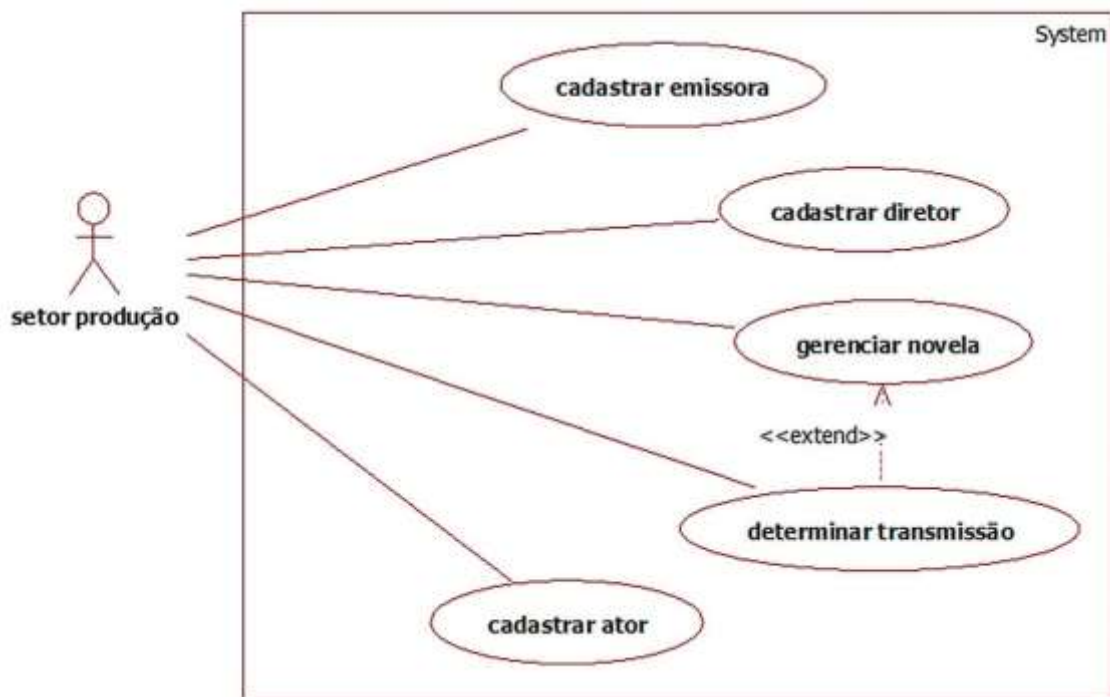


Figura 2 – Exemplo de diagrama de casos de uso.

A partir dos casos de uso, podemos identificar as classes de análise que representam os objetos do negócio ou do domínio do problema. A Figura 3 ilustra um diagrama de classes da UML.

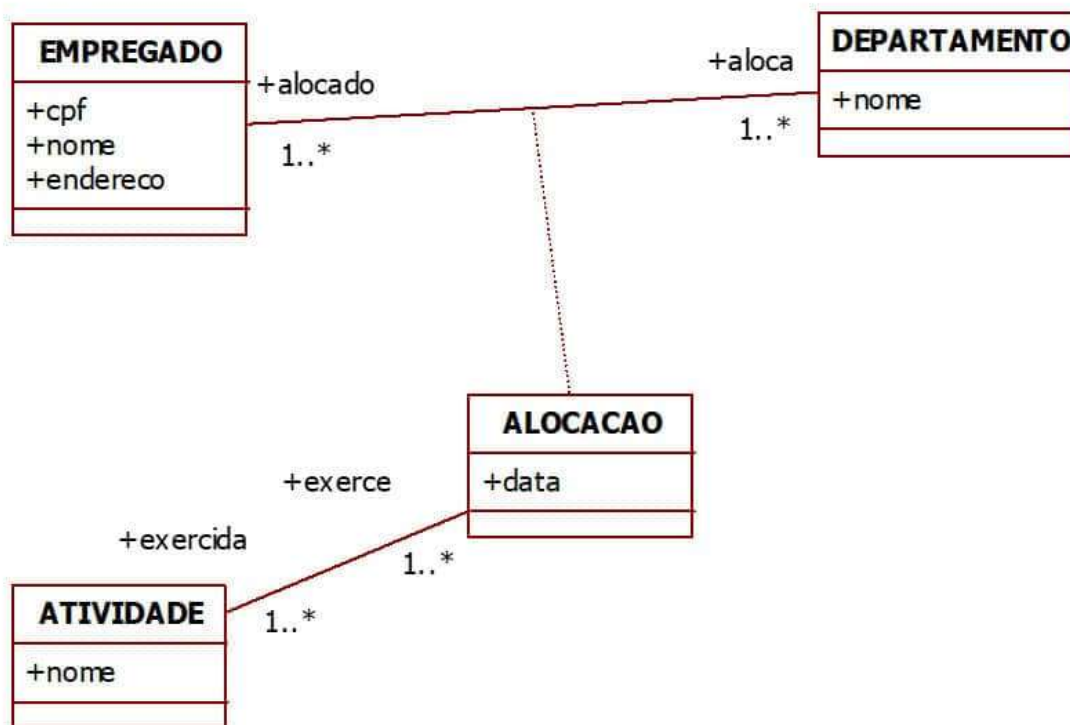


Figura 3 – Exemplo de diagrama de classes.

Na modelagem de análise, podemos considerar a construção do modelo de atividades e do modelo de estados. O modelo de atividades, ilustrado na Figura 4, possui diagramas que podem ser utilizados no mapeamento de processos de negócios, na descrição gráfica de um caso de uso ou na definição do algoritmo de um método. O modelo de estados permite representar mudanças de estados significativas e respectivos eventos que causam a referida mudança.

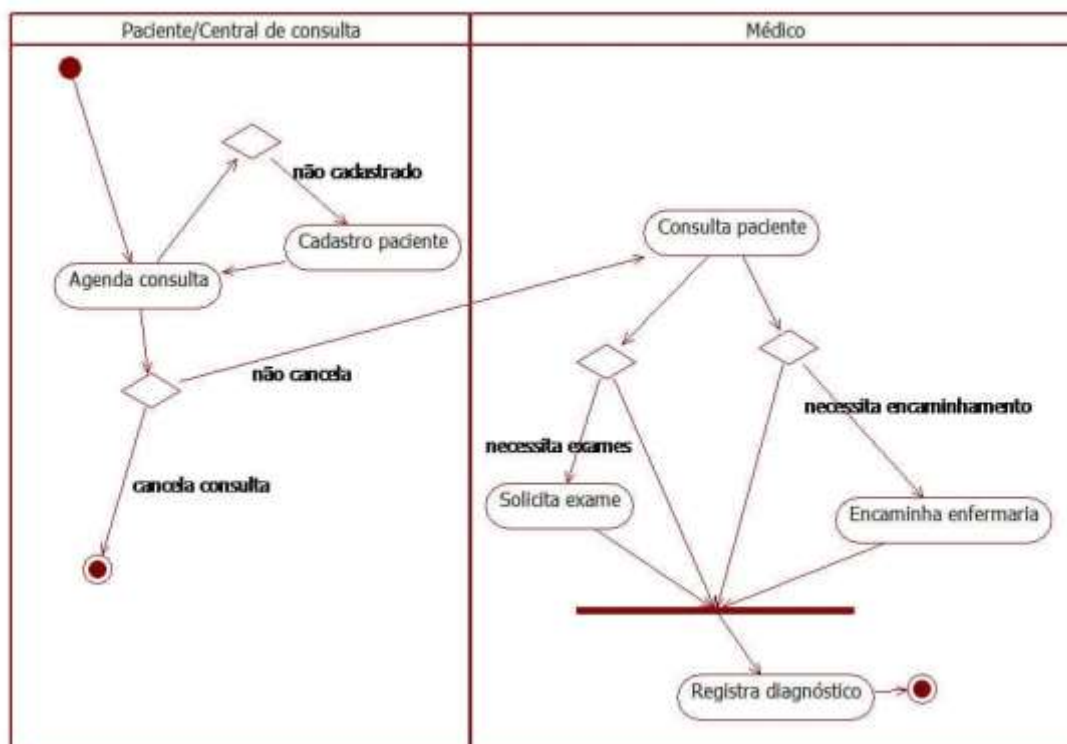


Figura 4 – Exemplo de diagrama de atividades.

## Negociação

Nesta etapa, ocorre a priorização e a resolução de conflitos entre os requisitos definidos nas etapas anteriores. Todos os envolvidos, equipe de projeto e usuários, participam da avaliação de custos, riscos e conflitos, a fim de eliminar, combinar e/ou modificar os referidos requisitos e, logicamente, priorizá-los.

## Especificação

O engenheiro de software deverá gerar um documento de especificação incluindo todos os requisitos e modelos gerados nas etapas anteriores.

## Validação

A validação permite ao engenheiro de software evidenciar que os modelos refletem as necessidades dos usuários com relação ao sistema a ser desenvolvido. Um defeito não considerado gera a construção de um sistema que não corresponderá às expectativas do usuário.

## Gestão

Finalizando o processo da engenharia de requisitos, a etapa de gestão permite controlar as mudanças dos requisitos à medida que o projeto evolui.

Podemos destacar que o engenheiro de software deverá considerar que o gerenciamento do escopo do projeto inclui um documento denominado matriz de rastreabilidade dos requisitos, sendo essa uma tabela que liga os requisitos dos

produtos desde as suas origens até as entregas que lhes satisfazem. A manutenção desse documento permite monitorar a estabilidade dos requisitos.

## **Resumindo**

Neste módulo, podemos destacar a relevância da Engenharia de Requisitos no contexto do processo de desenvolvimento de software. Importante lembrar que a Engenharia de Requisitos inclui as etapas de levantamento de requisitos e análise do referido processo.

Os requisitos são comumente categorizados em requisitos funcionais, não funcionais e de domínio. Os requisitos funcionais estão associados aos serviços ou às funcionalidades disponibilizadas pelo sistema. Os requisitos não funcionais estão relacionados com as restrições operacionais, tais como linguagem de programação, padrão de arquitetura etc., além de requisitos de qualidade, e.g., manutenibilidade, usabilidade e outros. Os requisitos de domínio detalham as regras de negócio identificadas no domínio do problema.

A Engenharia de Requisitos inclui um processo com tarefas que, de forma simplificada, permitem a identificação dos requisitos, a geração de modelos de análise, a validação dos requisitos por parte dos usuários e a gestão dos requisitos, possibilitando a rastreabilidade durante as etapas seguintes do projeto de software.