**Unidade 1**

**Seção 1**

**Análise e modelagem de sistemas**

A utilização de **TÉCNICAS PARA MODELAGEM E ANÁLISE DE SISTEMA** traz segurança aos desenvolvedores, pois é uma garantia de que se está desenvol­vendo o solicitado pelo cliente, além de ser uma forma de estabelecer padrões de qualidade para o desenvolvimento.

Na Unidade 1 deste livro, você verá como o **DESENVOLVIMENTO DE SOFTWARE EVOLUIU** ao longo dos anos, a **IMPORTÂNCIA DA ENGENHARIA DE SOFTWARE** para a evolução de processos de software, bem como o **DESENVOLVIMENTO AGIL NO PROCESSO DE SOFTWARE.**

Além disso, você conhecerá **O PAPEL DO ANALISTA DE SISTEMAS** e os **PROCESSOS E MODELOS DE SOFTWARE**  aplicados no desenvolvimento deste com o objetivo de reconhecer e identificar os **PROCESSOS DE SOFTWARE** e suas metodologias.

**Introdução à engenharia de software e à análise de sistemas;**

**Desafios**: **Analisar o software** de um possível cliente:

1. **Investigar o processo de software** da empresa de desenvolvimento em que você trabalha e propor melhorias nos processos de desenvolvimento de software.
2. **Avaliar e propor** um **modelo de processo de software** a partir da observação das características do software solicitado por um novo cliente.

**Sua missão será descobrir:**

\* E mais viável atualizar o sistema existente ou criar um?

\* Quais as novas tecnologias que podem ser utilizadas em um novo sistema para uma empresa de exames laboratoriais?

\* Quais processos deverão ser adotados para manter o software sempre atualizado?

\* Qual o perfil dos profissionais envolvidos no processo da análise do sistema do cliente?

Primeiro passo : **Fundamentos da engenharia de software:**

**A natureza e a evolução do software:**

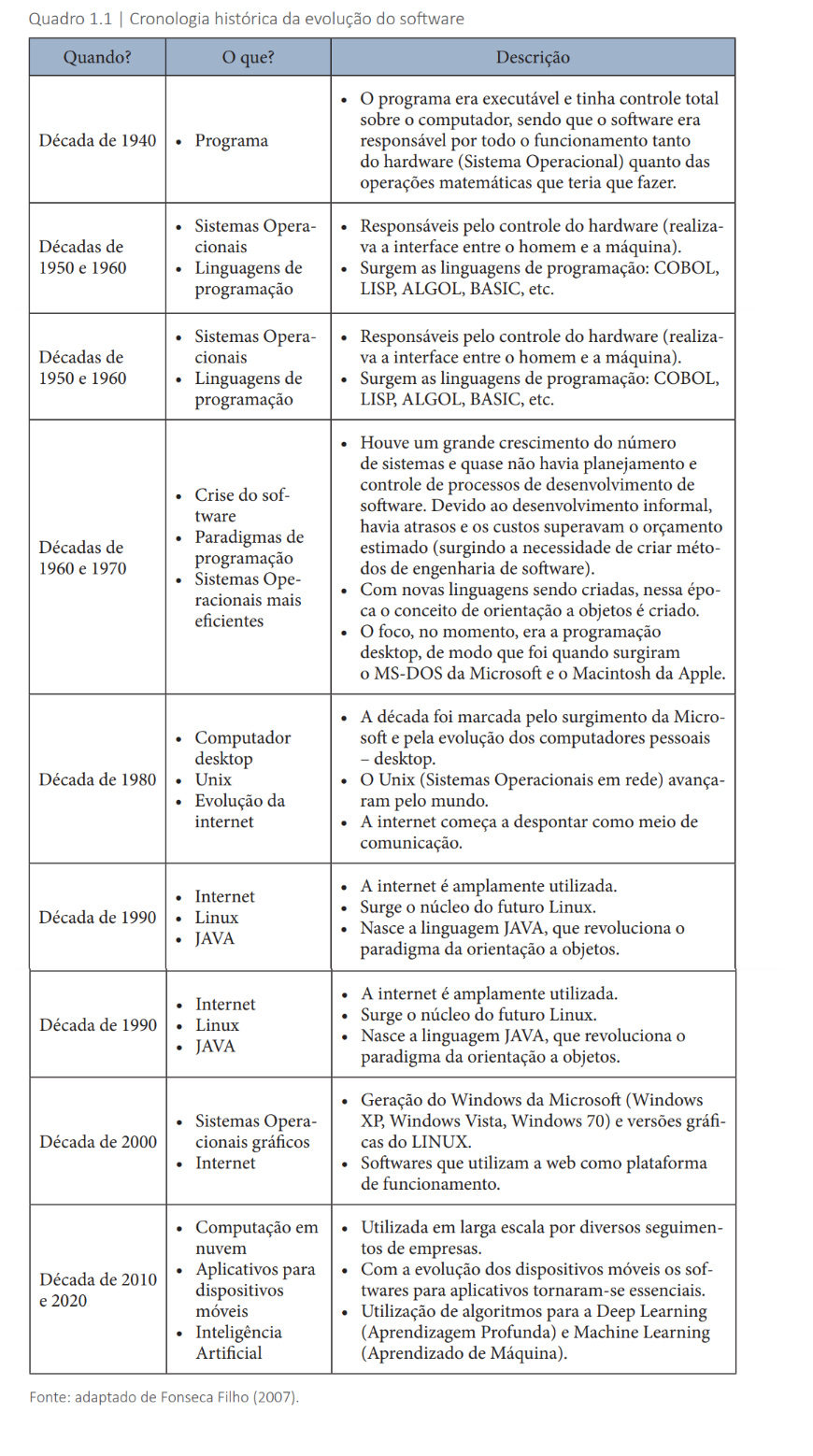
**Sommerville** (2011, p. 4) **estabelece** que os **softwares** são “**programas de computadores** com uma documentação associada e [que] os **produtos de software** podem ser desenvolvidos para um determinado cliente ou para um mercado mais generalizado”.

**Pressman** (2016) **afirma** que um **software de computador** é **um produto** que profissionais da área da Tecnologia da Informação (TI) desenvolvem e para o qual darão suporte a longo prazo; Além disso, abrange qualquer tipo de mídia eletrônica, consistindo na união de três elementos:

• **Instruções**: Quando executadas, fornecem os atributos e funções de desempenhos desejados pelos usuários.

• **Estruturas de dados**: Possibilitam aos programadores manipular as informações de forma mais adequada conforme a necessidade da aplicação.

• **Documentação**: É toda a informação descritiva do software, a qual detalha a operação de uso dos programas, diagramas de funcionalidades etc.



Assimile:

Qual a diferença entre **Software e Sistema?**

**Software** é definido como **um programa de computador**, uma **sequência lógica** de instruções a serem seguidas e/ou **executadas** na manipulação de um determinado conjunto de informações.

**Sistema** é um conjunto de compo­nentes inter-relacionados que funcionam de forma unificada para atingir um certo objetivo.

Essa definição de sistema é conhecida em algumas outras áreas, como a de engenharia de sistemas, sendo alguns exemplos: Sistema Operacional, sistema acadêmico e sistema bancário.

**É um conjunto de software, hardware e recursos humanos.**

As mudanças nos softwares ocorrem constantemente e essas mudanças podem ocasionar a necessidade de criar outro software;

A vida útil de um software é de acordo com a sua funcionalidade, pois pode parar de ser útil com o tempo;

**Leis de Lehman** podem ser utilizadas para **verificar a dinâmica da evolução dos softwares**

Podem ser utilizadas em diferentes softwares que possuam os seguintes processos:

Tomada de decisão, planejamento, desenvolvimento e manutenção.

**Mudança contínua:** Deve ser sempre adaptado para não ficar menos eficaz;

**Aumento da complexidade :** Com as mudanças ele se tornam complexo, deve ser simplificado;

**Evolução (autorregulação):** Auto ajustável, os atributos quase não mudam;

**Estabilidade organizacional:** Evolução tende a ser constante independe dos recursos;

**Conservação da familiaridade:** A evolução é proporcional ao domínio que a equipe adquire;

**Crescimento contínuo:** A funcionalidade tende a aumentar conforme novas versões surge;

**Declínio da qualidade:** Se não evoluir perde a qualidade**;**

**Sistema de *feedback*:** São agregados para obter melhorias;

**Prática da Engenharia de Software:**

Se faz necessária por dois fatores:

1. A sociedade precisa cada vez mais de software modernos;
2. A longo prazo é mais barato pois pode usar tecnologia já existentes ou reutilizar;

Conforme Pressman (2016), a **Engenharia de Software** é uma tecnologia em quatro camadas que objetiva **a Disciplina, a Adaptabilidade e a Agilidade**:

**As quatro camadas são**:



**Foco na qualidade:** É o objetivo final de toda engenharia de software; A gestão de todas camadas devem focar na qualidade;

**Processo:** É a base da engenharia de software; é a ligação entre as camadas é a base para o controle;

**Métodos**: Fornecem informações técnicas para o desenvolvimento do software;

e gerenciamento de projetos;

**Ferramentas:** É um alicerce para o processo e o método;(automatizado ou semi -cases);

Possibilitam a construção de sistemas, incorporando cinco atividades especí­ficas nesses processos: **Comunicação, Planejamentos, Modelagem, Construção e Entrega.**

**A Engenharia de Software,** segundo Sommerville (2011), preocupa-se com **todos os aspectos** **de produção do software.**

**Sete grandes categorias de Softwares**

**Software de Sistema:** Programas para auxiliar programas :Compiladores, Drivers;

**Software de Aplicação:** Resolver problema específicos: Planilhas, Editor texto;

**Software de Engenharia/Científico:** Grande volumes informações: Astronomia;

**Software Embarcado:** Funciona em determinado produto: Micro-ondas, tv;

**Software para Linha de Produtos:** Usados por diversos clientes: RH, Financeiro;

**Software de Aplicações Web:** Aplicativos móveis Foco em dispositivo com internet;

**Software de Inteligência Artificial:** Algoritmos sofisticados: Reconhecimento Voz Facial;

**Softwares legados:** Sistemas antigos;

**Os princípios da análise de sistemas:**

Os princípios da análise de sistemas fundamentam-se na necessidade de realizar **estudos de processos** para encontrar a melhor solução para a criação de um sistema.

Conforme Roth, Dennis e Wixom (2014), **A ANÁLISE DE SISTEMAS** baseia-se em **Métodos e Técnicas de Investigação e em Especifi­cação** para encontrar a melhor solução para algum problema ou necessidade computacional de determinada área de negócio, a partir das funcionalidades levantadas pelo analista de sistemas;

**Fases** que envolvem os **Processos da Análise de Sistema,** destacam-se, conforme Pressman:

**Análise:** Nesta fase são realizados estudos que objetivam a **Especifi­cação do Software,** de modo a **Verificar a Viabilidade** (custo-benefício), definir as **Funcionalidades** que o software deverá possuir e realizar o **Escopo,** alocando recursos e realizando o orçamento do software.

O resultado desta fase será utilizado nas próximas etapas.

**Projeto:** Nesta etapa há uma preocupação com a **Definição Lógica do Software,** são elaborados os *layouts* de telas e relatórios e são criados a estrutura de banco de dados e os diagramas gráficos para o desenvol­vimento do software.

**Implementação:** Nesta fase é realizada a codificação do software por meio de uma linguagem de programação (definida na fase de análise);

**Testes:** Objetivando a procura de erros, nesta fase, são realizados procedimentos de testes que verificam as funcionalidades dos itens codificados.

**Documentação:** Trata-se de documentar todos os processos (de todas as fases) e diagramas produzidos; são utilizados documentos padroni­zados (e personalizados por cada empresa de desenvolvimento), que servem como ferramenta de comunicação entre as pessoas envolvidas no desenvolvimento também como parte de contrato entre as partes interessadas na produção do software.

**Manutenção:** Esta fase consiste em fazer o acompanhamento do software após ser implantado e entrar em funcionamento (durante um período), visando a registrar e corrigir falhas, propor melhorias ou incluir novas funcionalidades.

**Problemas** ao não adotarem as **Técnicas da Engenharia de Software na Análise de Sistemas:**

1. Softwares com manutenção muito onerosa e demorada;
2. Falta de padronização na documentação;
3. Falta de controle em determinar a sequência do que deve ser realizado;
4. Previsões imprecisas a respeito do prazo de entrega e do preço final do software desenvolvido;

**Princípios da Análise de Sistemas**:

**- O Domínio** da informação de um determinado problema deve ser **representado e entendido** por todas as partes envolvidas;

1. **- As funcionalidades** de um software precisam ser **definidas e descritas** de forma genérica (inicialmente) até a forma mais genérica;
2. **- O Comportamento** do software deve ser representado através das interações com o ambiente externo, ou seja, com **usuários** e/ou **outros sistemas.**
3. **- Os Diagramas** que demonstram as funções e comportamentos do software **devem ser divididos** para revelar detalhes em **forma de camadas**, de modo que se decompõe um problema complexo em partes menores para facilitar sua compreensão;
4. **- A tarefa de Análise** deve ir **da Informação essencial** até os detalhes **de Implementação,** sem a preocupação de como será realizada a codifi­cação da solução; os detalhes sobre a implementação determinam como a solução será realizada.

**Papel do analista de sistemas:**

O A**nalista de Sistemas** possui um papel fundamental nos **Processos da Engenharia de Software** conforme afirma Sommerville (2011). Esse profis­sional é o responsável por realizar Atividades da Análise de Sistema como: **Pesquisas, Planejamentos, Coordenação de Equipes de Desenvolvimento e Recomendação de Alternativas de Software** de acordo com as necessidades de desenvolvimento ou de solução para problemas de negócios.

**O Analista de Sistemas** possui como tarefas **a** **Criação, a Implementação e a Implantação de um Software,** de maneira que deve, **primeiro:** Descobrir o que um sistema deverá fazer; **Depois** entender e avaliar as necessidades e expectativas de cada usuário do software, a fim de que estas sejam **Organizadas, Especificadas e Documentas.**

Para Elmasri e Navathe (2011), o Analista de Sistema **desenvolve as especi­ficações, as funcionalidades e as transações** customizando as soluções para atender as solicitações dos usuários.

O **Analista de Sistemas** é a **** entre os **Programadores** e os **Usuários Finais** do software, sendo ele o responsável por **interpretar** os anseios dos usuários e por saber o que é ou não viável para ser desenvolvido.

Durante o desenvolvimento de um software, o **Analista de Sistemas** possui as seguintes **atribuições:**

• **Interagir** com clientes e usuários finais do software.

• **Analisar** **custos** e **verificar a viabilidade** do projeto.

• Fazer o **levantamento das informações** através de **entrevistas** com usuários do software.

• **Levantar os dados e os requisitos** do software para analisar e propor soluções.

• **Criar a modelagem** do software.

• **Orientar os Programadores**, acompanhando todo o desenvolvimento do software (tanto na parte lógica quanto na parte de interface gráfica).

• **Acompanhar e executar** testes.

• **Preparar e acompanhar** toda a documentação do software.

• **Gerenciar** eventuais **mudanças** no projeto.

• **Determinar padrões** de desenvolvimento.

• **Garantir a qualidade** final do software e que este esteja de acordo com o solicitado pelo cliente.

• **Realizar o monitoramento** e fazer auditorias, procurando eventuais falhas.

• **Planejar e aplicar** treinamentos para a utilização do software desenvolvido.

• **Implantar** o software desenvolvido, acompanhando o processo de adaptação e integração dos sistemas do cliente.

• **Proporcionar consultoria** técnica para identificar as necessidades dos clientes nas mais diversas áreas de negócio.

• **Pesquisar** novas tecnologias, fornecedores e, se for o caso, buscar por especializações para si e

para a equipe de desenvolvimento.