

# **Engenharia de Software e Análise de Sistemas: Fundamentos e Processos para o Desenvolvimento de Soluções Tecnológicas**

## **Autores:**

- 1 - Eduardo Furlan
- 2 - João Vitor Magri
- 2 - Alessandro Silva Costa Coimbra
- 2 - Nelson Gabriel Vilas Boas Florentino
- 2 - Guilherme de Lima Brussolo

- 1 - Docente da Universidade Anhanguera
- 2 - Acadêmicos do curso de Análise e Desenvolvimento de Sistemas (ADS) da Universidade Anhanguera em SP.

Palavras-chave: Engenharia de Software, Análise de Sistemas, Processo de Software, Desenvolvimento de Software, Analista de Sistemas.

## **Introdução**

A tecnologia, em sua constante evolução, está cada vez mais integrada ao nosso estilo de vida, com inovações como computação em nuvem, realidades virtual e aumentada, e reconhecimento facial tornando-se comuns no dia a dia. Contudo, uma inovação de suma importância, e muitas vezes subestimada, é o software. A combinação eficiente de hardware e software é fundamental para diversas aplicações, sendo que um dispositivo de última geração sem funcionalidades ou aplicativos adequados perde sua utilidade. Essa demanda crescente por softwares eficientes e inovadores impulsiona a necessidade de mão de obra especializada e de técnicas eficazes para a criação de novos sistemas.

Nesse cenário, a Engenharia de Software e a Análise de Sistemas emergem como disciplinas cruciais, fornecendo os conhecimentos e as ferramentas necessárias para desenvolver soluções tecnológicas de forma eficaz e organizada. A complexidade inerente ao desenvolvimento de software, aliada às expectativas dos usuários e às mudanças tecnológicas, exige uma abordagem estruturada para garantir a qualidade, a funcionalidade e a longevidade dos sistemas.

Este artigo tem como objetivo explorar os fundamentos da engenharia de software, os princípios da análise de sistemas e a estrutura dos processos de desenvolvimento de software, conforme delineado nas Seções 1 e 2 da Unidade 1 do livro "Introdução à Engenharia de Software e à Análise de Sistemas". A compreensão aprofundada desses conceitos é essencial para profissionais da área de Tecnologia da Informação (TI), permitindo-lhes enfrentar os desafios do desenvolvimento de sistemas e propor melhorias contínuas.

## Objetivo

O objetivo principal deste trabalho é identificar e analisar os fundamentos da engenharia de software e da análise de sistemas, bem como os processos e metodologias envolvidos no desenvolvimento de software.

Os objetivos específicos a serem atingidos incluem:

- Descrever a natureza e a evolução do software, destacando a importância da prática da engenharia de software.
- Apresentar os princípios da análise de sistemas e o papel fundamental do analista de sistemas no ciclo de vida do software.
- Explicar a estrutura e as atividades de um processo genérico de software, bem como as abordagens de avaliação e melhoria de processos.

## Material e Métodos

O presente trabalho caracteriza-se como uma revisão bibliográfica, de natureza qualitativa e descritiva, baseada na análise e síntese de informações contidas na Unidade 1, Seção 1 ("Fundamentos da engenharia de software") e Seção 2 ("O processo de software") do livro "Introdução à Engenharia de Software e à Análise de Sistemas", de autoria de Claudia Werlich (2020).

A metodologia empregada envolveu a leitura aprofundada das seções mencionadas, a extração de conceitos chave, definições e discussões teóricas. Para a estruturação e formatação do artigo, foram seguidas as diretrizes estabelecidas no "Regulamento Geral" do 28º Encontro de Atividades Científicas (EAC). As citações e referências foram realizadas de forma a atribuir corretamente as informações às fontes originais, conforme o padrão [i] exigido pelo evento.

## Resultados e Discussão

### 1. Fundamentos da Engenharia de Software e Análise de Sistemas

A engenharia de software é uma disciplina que se preocupa com todos os aspectos da produção de software. Segundo Sommerville (2011), softwares são "programas de computadores com uma documentação associada". Pressman (2016) complementa, afirmando que o software abrange instruções, estruturas de dados e documentação descritiva.

A evolução do software tem sido contínua e impactante. Desde a década de 1940, com programas executáveis e controle total sobre o computador, até as décadas de 2010 e 2020, com a popularização da computação em nuvem, aplicativos móveis e Inteligência Artificial. Essa evolução é um reflexo das necessidades dos clientes, que se alteram conforme a tecnologia avança.

É crucial diferenciar software de sistema: enquanto o software é um programa de computador, um sistema é um conjunto de componentes inter-relacionados (software, hardware e recursos humanos) que trabalham para um objetivo comum. A complexidade e a necessidade de manutenção do software são constantes, pois ele sofre alterações ao longo de sua vida útil, o que pode gerar novos erros e aumentar a taxa de defeitos. As Leis de Lehman (década de

1980) ainda são relevantes, destacando a mudança contínua, o aumento da complexidade, a evolução autorregulada, a estabilidade organizacional, a conservação da familiaridade, o crescimento contínuo, o declínio da qualidade e a importância dos sistemas de feedback no processo evolutivo do software.

A prática da engenharia de software é essencial para a produção de softwares confiáveis e econômicos, além de permitir a reutilização de componentes em outros sistemas. Ela é estruturada em quatro camadas: foco na qualidade, processo, métodos e ferramentas. Engloba atividades como comunicação, planejamento, modelagem, construção e entrega. Existem diversas categorias de softwares, incluindo software de sistema, de aplicação, de engenharia/científico, embarcado, para linha de produtos, aplicações web/móveis e de Inteligência Artificial. Um desafio comum são os softwares legados, que são sistemas antigos com baixa qualidade, defasados e com documentação deficitária, tornando sua manutenção cara e complexa.

A análise de sistemas baseia-se em métodos e técnicas de investigação e especificação para encontrar a melhor solução computacional para problemas de negócio. As fases da análise de sistema incluem: Análise (viabilidade, funcionalidades, escopo), Projeto (definição lógica, layouts, estrutura de banco de dados), Implementação (codificação), Testes (busca de erros), Documentação (registro de processos e diagramas) e Manutenção (acompanhamento e correção de falhas). Os princípios da análise de sistemas enfatizam a compreensão do domínio da informação, a definição genérica das funcionalidades, a representação do comportamento do software, a divisão de diagramas em camadas para facilitar a compreensão e a separação da informação essencial dos detalhes de implementação.

O papel do analista de sistemas é fundamental, atuando na pesquisa, planejamento, coordenação de equipes e recomendação de alternativas de software. Ele serve como uma "ponte" entre os programadores e os usuários finais, interpretando anseios e viabilidades técnicas. Suas atribuições incluem interagir com clientes, analisar custos, levantar informações e requisitos, criar modelagens, orientar programadores, acompanhar testes, preparar documentação, gerenciar mudanças, determinar padrões de desenvolvimento, garantir a qualidade, monitorar, auditar, planejar treinamentos e proporcionar consultoria técnica. Habilidades como conhecimento tecnológico atualizado, organização, visão gerencial e bom relacionamento interpessoal são essenciais para este profissional.

## **2. O Processo de Software**

Um Processo de Software é um conjunto de atividades e resultados relacionados que levam à produção de um software. Pressman (2016) destaca que não é uma determinação rigorosa, mas uma abordagem adaptável que permite à equipe escolher os processos mais adequados, com foco na qualidade, prazo e redução de custos. Sommerville (2011) o descreve como a abordagem sistemática usada pela Engenharia de Software para produzir software.

A importância de um processo de software bem definido reside na sua capacidade de criar padronização, permitir a repetição e reutilização, reter conhecimento na empresa, definir e guiar as atividades do projeto, determinar tarefas, reduzir riscos, proporcionar visões comuns e servir como template para novos projetos. Na definição dos processos, são estabelecidos parâmetros como o evento de início, matriz de responsabilidades, atividades e sequencialidades, entradas e saídas, regras, infraestrutura e resultados.

A estrutura de um Processo Genérico de Software pode ser diferente em cada empresa, mas geralmente inclui quatro atividades fundamentais: especificação, projeto e implementação, validação e evolução. Pressman (2016) detalha cinco atividades metodológicas genéricas: Comunicação (entendimento dos objetivos e requisitos), Planejamento (criação de um mapa do projeto, riscos, recursos, cronograma), Modelagem (criação de diagramas para melhor compreensão), Construção (codificação e testes) e Entrega (apresentação do software para feedback do cliente).

A organização dessas atividades pode variar de acordo com o Fluxo de Processo, que descreve como as atividades metodológicas são organizadas. Os fluxos podem ser: Linear (sequencial), Iterativo (repetição de atividades), Evolucionário (execução circular, gerando versões mais completas) ou Paralelo (atividades simultâneas). Projetos distintos exigem conjuntos de tarefas e modelagem de atividades diferenciados. Um Modelo de Processo de Software é uma descrição simplificada que especifica atividades, produtos, papéis e oferece um roteiro para a Engenharia de Software.

A qualidade do software produzido é diretamente influenciada pelos padrões de qualidade impostos durante os processos. Para garantir a integração e validação entre as atividades, são utilizadas diversas abordagens de avaliação e aperfeiçoamento de processos de software, como: SCAMPI (Método Padrão CMMI de Avaliação para Aperfeiçoamento de Processos), que fornece um modelo de avaliação em cinco etapas; CBA IPI (Avaliação para o Aperfeiçoamento do Processo Interno baseado na CMM), uma técnica de diagnóstico para avaliar a maturidade organizacional; SPICE (ISO/IEC15504), um padrão que define requisitos para avaliação de processos de software; e ISO 9001:2000 para software, um padrão genérico de qualidade global aplicável a produtos, sistemas ou serviços. O CMMI (Modelo de Capacidade e Maturidade Integrado) é um conjunto de práticas para alcançar metas pré-estabelecidas e aumentar o amadurecimento organizacional, com modelos específicos para Desenvolvimento, Aquisição e Serviços.

## **Conclusão**

A Engenharia de Software e a Análise de Sistemas são pilares essenciais no desenvolvimento de soluções tecnológicas que permeiam a sociedade contemporânea. A análise aprofundada dos fundamentos da engenharia de software revela a complexidade da criação e evolução de sistemas, destacando a necessidade de abordagens estruturadas e adaptáveis. A compreensão da natureza do software, sua evolução e as leis que regem sua dinâmica são cruciais para a construção de produtos robustos e eficazes.

Adicionalmente, os princípios da análise de sistemas fornecem a base para investigar, especificar e projetar soluções computacionais, garantindo que as necessidades dos usuários sejam compreendidas e traduzidas em funcionalidades tangíveis. Nesse contexto, o analista de sistemas assume um papel central, atuando como um elo entre as partes interessadas e garantindo a qualidade e a conformidade do software ao longo de todo o seu ciclo de vida.

Por fim, a definição e a aplicação de um processo de software bem estruturado são indispensáveis para gerenciar a complexidade do desenvolvimento, padronizar atividades, reter conhecimento e garantir a entrega de produtos de qualidade dentro dos prazos e orçamentos estabelecidos. As diversas abordagens de avaliação e melhoria de processos, como SCAMPI, CBA IPI, SPICE e ISO 9001:2000, reforçam a importância da melhoria contínua para o sucesso dos projetos de software.

Em suma, a interseção entre a engenharia de software e a análise de sistemas, mediada por processos bem definidos e profissionais qualificados, é o caminho para o desenvolvimento de soluções tecnológicas inovadoras e eficientes, capazes de atender às demandas em constante transformação da sociedade digital.

## **Referências**

PEREIRA, E. F.; FERNANDES, C. P. A Aprendizagem da Criança e Adolescente com Dificuldade de Aprendizagem. TCC, Anhanguera - Faculdade Anhanguera de Bauru, 2022.  
WERLICH, C. Introdução à Engenharia de Software e à Análise de Sistemas. Londrina: Editora e Distribuidora Educacional S.A., 2020. ENCONTRO DE ATIVIDADES CIENTÍFICAS (EAC). Regulamento Geral. [S.l.], [s.d.].