**FUNDAMENTOS DA ENGENHARIA DE SOFTWARE E ANÁLISE DE SISTEMAS E SUA ILUSTRAÇÃO EM ROBÓTICA**

**Palavras-chave:** Engenharia de Software; Análise de Sistemas; Evolução do Software; Robótica; Software de IA.

**1. INTRODUÇÃO**

A engenharia de software é uma área crucial da Tecnologia da Informação (TI), dedicada ao desenvolvimento e suporte de longo prazo de produtos de softwares. O software, composto por programas executáveis, estruturas de dados e documentação, é fundamental para a eficiência e agilidade das aplicações, sendo impulsionado pela crescente demanda por soluções inovadoras. Paralelamente, a análise de sistemas emerge como disciplina essencial para investigar e especificar a melhor solução para problemas computacionais e necessidades de negócios. Ao longo das décadas, o software passou por uma evolução significativa, desde o surgimento de programas executáveis nas décadas de 1940-1970 até a ascensão da Inteligência Artificial (IA) em 2020. Este artigo tem como objetivo sintetizar os fundamentos da engenharia de software e da análise de sistemas, contextualizando sua evolução e ilustrando sua aplicação prática por meio de um exemplo de código para robótica, alinhado aos "Softwares de IA".

**2. OBJETIVO**

O presente trabalho busca explorar os conceitos basilares da engenharia de software e da análise de sistemas, detalhando suas camadas, categorias e fases. Adicionalmente, visa demonstrar, através de um exemplo prático de código para um robô seguidor de linha, como esses princípios teóricos se materializam no desenvolvimento de soluções de software, especialmente no contexto de sistemas embarcados e de inteligência artificial.

**3. MATERIAL E MÉTODOS**

Este resumo expandido foi elaborado a partir de uma **análise conceitual e descritiva** do material textual fornecido. Os dados foram extraídos do documento "Análise e Modelagem de Sistemas - Atividade.pdf", focando nas seções "1.1 Fundamentos da Engenharia de Software e Análise de Sistemas", "1.2 Evolução do Software", "1.3 Análise de Sistemas" e "1.4 Código básico para robô seguidor de linha". A metodologia consistiu na identificação dos conceitos chave, na organização lógica das informações para apresentar a inter-relação entre engenharia de software e análise de sistemas, e na contextualização da evolução do software. O código básico para robô seguidor de linha foi utilizado como um estudo de caso ilustrativo para exemplificar a aplicação prática dos conceitos discutidos, especialmente em se tratando de softwares embarcados e de IA. A estrutura e formatação seguiram as regras de "Resumo Expandido" do regulamento do 28° EAC.

**4. RESULTADOS E DISCUSSÃO**

A **Engenharia de Software** é definida pela dedicação ao desenvolvimento e suporte de softwares, que são compostos por programas executáveis, estruturas de dados e documentação. Embora o software não se desgaste fisicamente, ele pode se deteriorar com alterações e manutenções ao longo do tempo. Segundo Pressman, a engenharia de software é estruturada em quatro camadas:

1. **Processo:** a base que conecta as demais camadas, onde artefatos como modelos e relatórios são produzidos.

2. **Métodos:** fornecem as informações técnicas para o desenvolvimento, abrangendo tarefas como comunicação, análise de requisitos, modelagem, codificação, testes e manutenção.

3. **Ferramentas:** oferecem suporte automatizado ao processo, integrando processos, métodos e ferramentas para a construção de sistemas.

A **evolução do software** é marcada por etapas significativas:

• **1940 - 1970:** Surgimento de programas executáveis, sistemas operacionais e linguagens de programação, seguido pela Crise do Software que impulsionou a Programação Orientada a Objetos (POO).

• **1980 - 2000:** Ascensão dos computadores desktop, ampliação da internet, nascimento da linguagem Java e softwares baseados em Web.

• **2010:** Computação em nuvem e aplicativos mobile.

• **2020:** Intensificação da utilização da Inteligência Artificial (IA).

O software pode ser categorizado em sete grandes tipos:

1. **Software de sistema:** programas que auxiliam outros programas (ex: compiladores, drivers).

2. **Software de aplicação:** programas independentes que resolvem problemas de negócios (ex: planilhas, editores de textos).

3. **Software de engenharia/científico:** para cálculos complexos em áreas como meteorologia.

4. **Software embarcado:** programados para funcionar em produtos específicos (ex: fornos micro-ondas, módulos de carros).

5. **Software para linha de produtos:** projetados para diversos clientes com ou sem adaptações.

6. **Software de aplicações web/apps mobile:** focados em dispositivos com acesso à internet.

7. **Softwares de IA:** utilizam algoritmos sofisticados para solucionar problemas complexos, incluindo robótica e redes neurais. Os softwares legados, sistemas antigos em uso, representam um desafio devido à baixa qualidade e manutenção cara.

A **Análise de Sistemas** baseia-se no estudo de processos para encontrar a melhor solução na criação de um sistema, utilizando métodos e técnicas de investigação e especificação para resolver problemas computacionais ou necessidades de negócios. As fases da análise de um sistema incluem:

• **Análise:** estudo de viabilidade, definição de funcionalidades, escopo, alocação de recursos e orçamento.

• **Projeto:** definição lógica do software, elaboração de layouts, criação da estrutura do banco de dados e diagramas.

• **Implementação:** codificação do software na linguagem definida.

• **Testes:** busca de erros e verificação de funcionalidades.

• **Documentação:** registro de todos os processos.

• **Manutenção:** acompanhamento do software após implantação.

Como exemplo prático da aplicação desses conceitos, o **"Código básico para robô seguidor de linha"** ilustra um software embarcado que utiliza lógica de programação para resolver um problema específico. O código fornecido demonstra a importação de uma biblioteca (time), a definição de uma classe para o robô (RoboSimulado) e métodos que representam ações como frente(), direita(), esquerda() e parado(). A criação de um objeto RoboSimulado e a utilização de time.sleep(1) para simular o tempo de movimento são exemplos diretos da **implementação** e **testes** de um software no contexto da robótica, que pode ser considerado um tipo de software de IA. Este exemplo, embora simples, encapsula as atividades de construção e parte da funcionalidade de um sistema de software, desde a definição de suas capacidades até a simulação de sua execução.

**5. CONCLUSÃO**

A engenharia de software e a análise de sistemas são disciplinas interdependentes e essenciais para o desenvolvimento de soluções tecnológicas eficientes. A contínua evolução do software, culminando na ascensão da Inteligência Artificial, ressalta a importância de metodologias robustas e mão de obra especializada. O exemplo do robô seguidor de linha demonstra como os princípios e fases da engenharia e análise de sistemas se traduzem em aplicações práticas, desde a concepção de suas funcionalidades até a sua implementação e validação, ilustrando a capacidade do software de utilizar algoritmos para resolver problemas complexos. Compreender esses fundamentos é crucial para projetar e construir sistemas que atendam às demandas de um mundo cada vez mais digitalizado e complexo.

**6. REFERÊNCIAS**

Excerpt from "Análise e Modelagem de Sistemas - Atividade.pdf", Seção 1.1, p. 1. Excerpt from "Análise e Modelagem de Sistemas - Atividade.pdf", Seção 1.1, p. 1. Excerpt from "Análise e Modelagem de Sistemas - Atividade.pdf", Seção 1.2, p. 1. Excerpt from "Análise e Modelagem de Sistemas - Atividade.pdf", Seção 1.2, p. 1-2. Excerpt from "Análise e Modelagem de Sistemas - Atividade.pdf", Seção 1.2, p. 2. Excerpt from "Análise e Modelagem de Sistemas - Atividade.pdf", Seção 1.2, p. 2. Excerpt from "Análise e Modelagem de Sistemas - Atividade.pdf", Seção 1.3, p. 3. Excerpt from "Análise e Modelagem de Sistemas - Atividade.pdf", Seção 1.3, p. 3.