* **1.1 Fundamentos da Engenharia de Software e Análise de Sistemas**

A engenharia de software é uma área crucial em TI que se dedica ao desenvolvimento e suporte de longo prazo de produtos de softwares. Softwares são programas de computador que vêm com documentação associada e podem ser feitos sob medida para um cliente específico ou para um mercado mais amplo. Um software é composto por três elementos principais:

* Introdução: quando executados, fornecem os atributos e funções de desempenho desejados pelos usuários.
* Estruturas de dados: permitem aos programadores manipular informações de forma adequada para a aplicação.
* Documentação: é toda a informação descritiva de software, detalhando a operação dos programas e diagramas de funcionalidades.

A união entre hardware e software é fundamental para a eficiência e agilidade das aplicações, e a demanda por softwares eficientes e inovadores impulsionam a necessidade de mão de obra especializada e de técnicas eficazes para a criação de sistemas.

* **1.2 Evolução do Software**

O software passou por uma evolução significativa, impulsionada pela popularização de novas tecnologias e pela crescente demanda da sociedade.

* 1940 - 1970:
  + Surgimento de programas executáveis com controle total do computador.
  + Aparecimento de sistemas operacionais e linguagens de programação.
  + Crise do Software: a falta de planejamento gerou atrasos e custos excessivos, porém, por conta da alta demanda de engenharia de software surgiu a programação orientada a objetos (POO).
* 1980 - 2000:
  + Ascensão dos computadores desktop.
  + Ampliação do uso da internet.
  + Nascimento da linguagem Java.
  + Softwares baseados em Web.
* 2010:
  + Computação em nuvem.
  + Apps mobile.
* 2020:
  + Utilização de IA.

Diferentemente do hardware, o software não se desgasta fisicamente, mas pode se deteriorar com as alterações e manutenções implantadas ao longo do tempo. A engenharia de software, segundo Pressman, possui quatro camadas:

1. Processo: base da engenharia de software, que conecta as demais camadas. É nessa camada que artefatos como modelos, documentos e relatórios são produzidos.
2. Métodos: fornecem as informações técnicas para o desenvolvimento abrangendo tarefas como comunicação, análise de requisitos, modelagem de projetos, codificação, testes e manutenção.
3. Ferramentas: oferecem suporte automatizado para o processo. Essa área engloba processos, métodos e ferramentas que possibilitam a construção de sistemas.

Existem sete grandes categorias de softwares:

1. Software de sistema: programas que auxiliam outros programas (compiladores, drivers).
2. Software de aplicação: programas independentes que resolvem problemas de negócios (planilhas, editores de textos, softwares personalizados).
3. Software de engenharia/científico: utilizados para cálculos complexos em áreas como meteorologia ou genética.
4. Software embarcado: programados para funcionar em produtos específicos (fornos-microondas, módulo de carros).
5. Software para linha de produtos: projetados para serem usados por diversos clientes, com ou sem adaptações.
6. Software de aplicações web/apps mobile: focados em dispositivos com acesso à internet (navegadores, celulares, tablets).
7. Softwares de IA: utilizam algoritmos sofisticados para solucionar problemas complexos (robótica, redes neurais, etc).

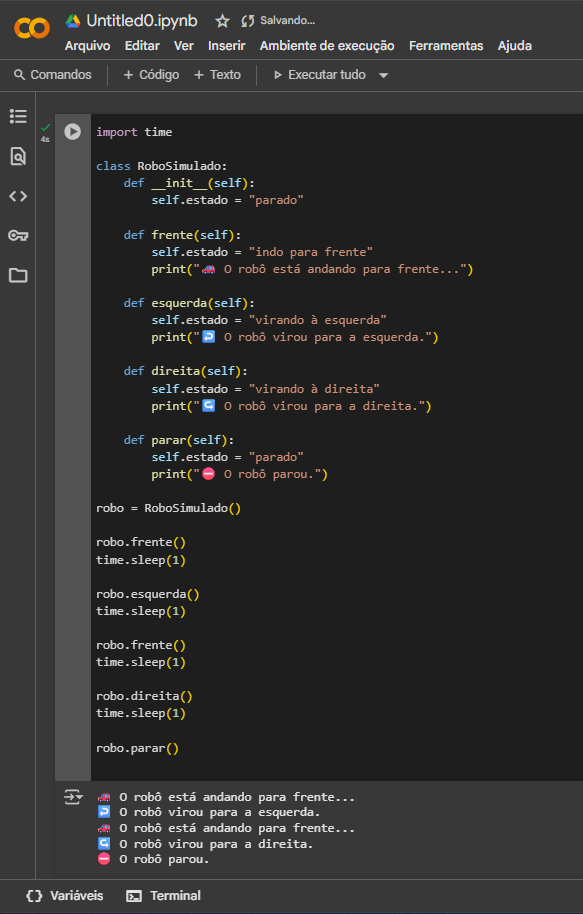
Um tipo comum encontrado em instituições são os softwares legados: sistemas antigos que continuam em uso, mas apresentam baixa qualidade, lentidão e funcionalidades desatualizadas ou não utilizadas. Frequentemente, apresentam documentação deficiente ou inexistente, tornando a manutenção cara e demorada, e a evolução geralmente implica a criação de um novo software com tecnologias mais recentes.

* **1.3 Análise de Sistemas**

Os princípios da análise de sistemas baseiam-se na necessidade de realizar estudos de processos para encontrar a melhor solução para a criação de um sistema. A análise de sistemas utiliza métodos e técnicas de investigação e especificação para resolver problemas computacionais ou necessidades de negócios, a partir das necessidades levantadas pelo analista.

Fases da Análise de um Sistema:

* Análise: estudo da viabilidade, definição de funcionalidades, escopo, alocação de recursos e orçamento.
* Projeto: definição lógica do software, elaboração de layouts de telas e relatórios, criação da estrutura do banco de dados e diagramas gráficos.
* Implementação: codificação do software usando a linguagem de programação definida na fase de análise.
* Testes: procedimento para buscar erros e verificar as funcionalidades codificadas.
* Documentação: registro de todos os processos.
* Manutenção: acompanhamento do software após implantação.
* **1.4 Código básico para robô seguidor de linha**

****

Na primeira linha do código tem a importação da biblioteca time que é usada para adicionar pausas no programa. Na linha 3 define uma classe para o robô e em seguida, das linhas 4 a 21, cada método representa uma ação que o robô pode fazer (frente, direita, esquerda e parado). Na linha 23 é criado um objeto “RoboSimulado” que pode usar os métodos frente(), esquerda(), etc. Já o time.sleep(1) faz o programa esperar 1 segundo, simulando o tempo de movimento.