

Análise de Sistemas

Prof. Romário Nzenguele da Silva



Licenciatura em Informática

Sobre o Documento



Material de apoio **gratuito**, desenvolvido exclusivamente para os estudantes do **3º Ano da Licenciatura em Informática**;



O conteúdo, sempre que necessário, será atualizado para refletir as mudanças e melhorias no decorrer do curso. Cada atualização será devidamente versionada, permitindo o fácil acompanhamento das revisões;



Este recurso é um **resumo das aulas**, essencial para apoiar no aprendizado da disciplina;



Sugestões de melhoria podem ser apresentadas pelo email que se encontra no rodapé.

Conteúdo

1. Fundamentos e Conceitos Básicos

- 1.1. Conceito de Sistema
- 1.2. Sistema de informação vs. sistema informático
- 1.3. Processo de Desenvolvimento de Software
- 1.4. Introdução à análise de sistemas orientada a objetos

Objectivos

- ✓ Compreender o Conceito de Sistema
- ✓ Diferenciar Sistema de Informação e Sistema Informático
- ✓ Conhecer as etapas do Processo de Desenvolvimento de Software (SDLC)
- ✓ Introduzir a Análise de Sistemas Orientada a Objetos

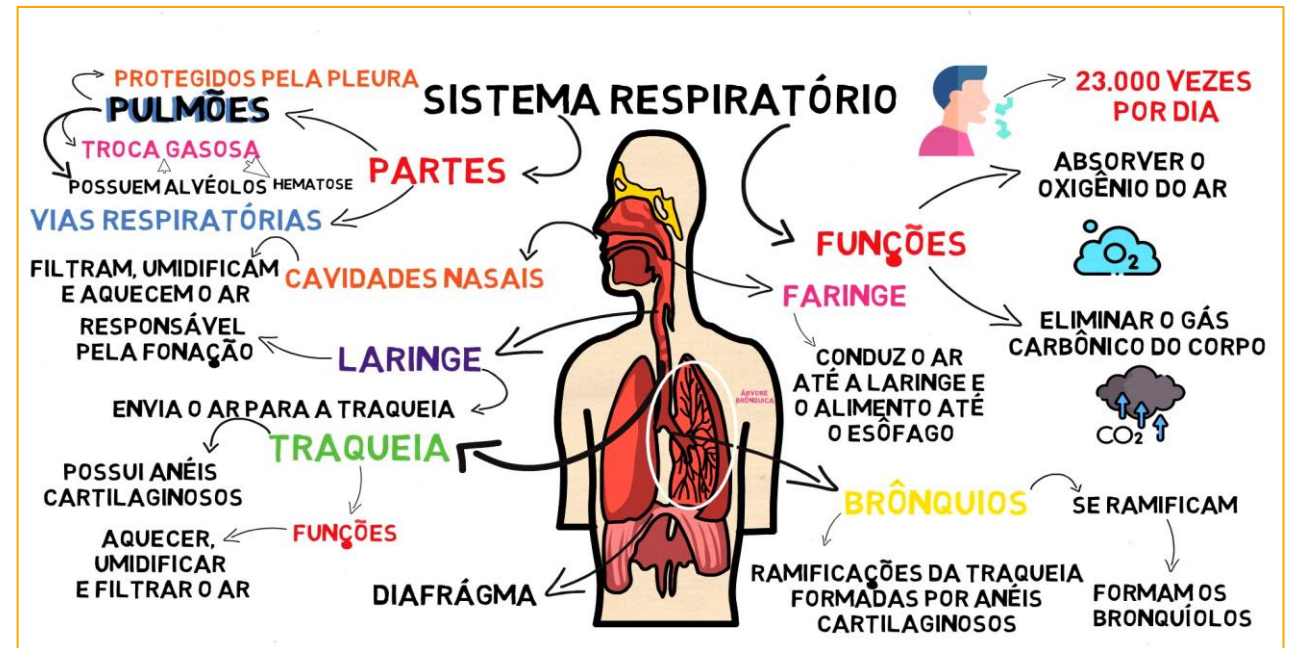


1. Fundamentos e Conceitos Básicos

1.1. Conceito de Sistema

❑ Elementos inter-relacionados que trabalham em conjunto para alcançar um objetivo comum.

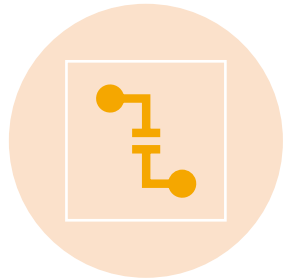
- Esses elementos podem ser pessoas, equipamentos, software, dados ou outros recursos que interagem entre si para gerar resultados específicos.
- O conceito de sistema é aplicável em várias áreas, desde sistemas biológicos até **sistemas computacionais**.



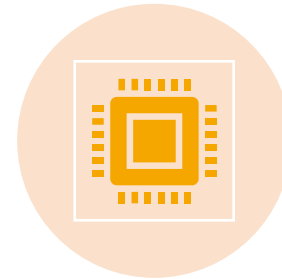
Fonte: [amazonaws](https://www.amazonaws.com)

1.1. Conceito de Sistema

❑ Características dos Sistemas



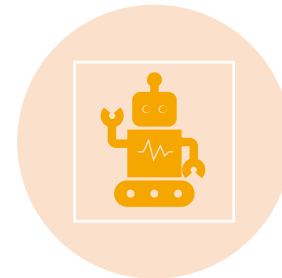
Entrada: dados, materiais, energia,... inputs necessários para iniciar o seu funcionamento.



Processamento: transformação das entradas para gerar um resultado.



Saída: resultado, que pode ser uma informação, produto, serviço...



Feedback: retroalimentação do sistema, usada para ajustar suas operações e melhorar a eficiência. Ele permite que o sistema se adapte a mudanças.

1.2. Sistema de Informação vs. Sistema Informático

❑ Sistema de informação

- Elementos inter-relacionados que trabalham em conjuntos para coletar, processar, armazenar e distribuir informações.
- Engloba **pessoas, dados, processos e tecnologia**.
- **Ex:** ERP (SAP, Primaveira), CRM, Sistemas De Gestão Hospitalar....



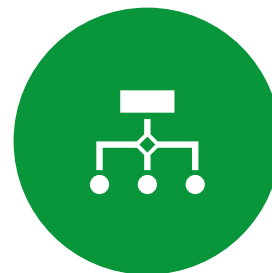
Fonte: genyo.com.br

1.2. Sistema de Informação vs. Sistema Informático

❑ Componentes de um Sistema de Informação



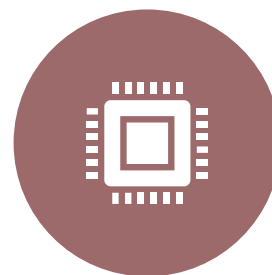
Pessoas: usuários ou operadores que interagem com o sistema.



Processos: definem como as operações e transações devem ser realizadas.



Dados: elementos brutos que são processados pelo sistema e transformados em informação útil.

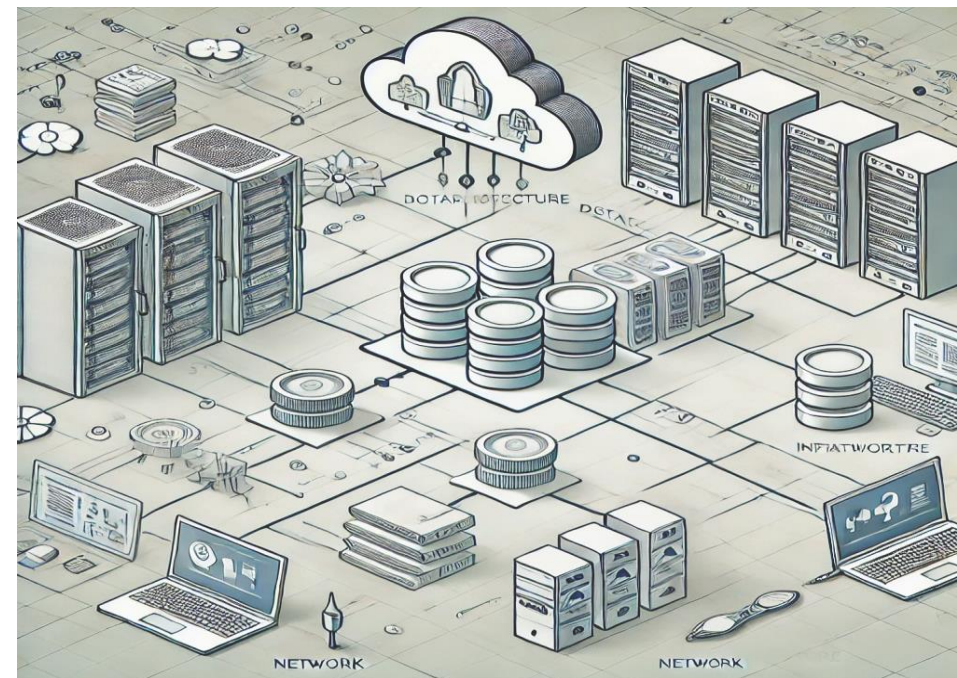


Tecnologia: componentes tecnológicos, que possibilitam o processamento e armazenamento dos dados.

1.2. Sistema de Informação vs. Sistema Informático

❑ Sistema informático

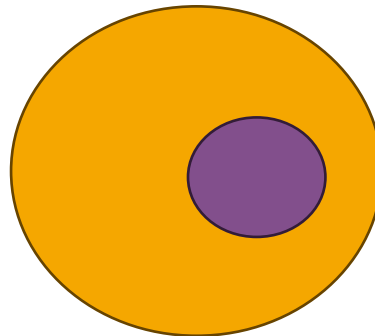
- Refere-se aos componentes tecnológicos de **hardware e software** utilizados para realizar processamento de dados.
- **Foca** no ambiente técnico, na **infraestrutura** de tecnologia que **suporta os processos de um sistema de informação**.
- É parte integral de um **sistema de informação**.
- Ex: Data Centers, Sistemas Operativos, Sistemas de Redes (hosts, switchs, hubs, routers, ...), ...



Fonte: Imagem gerada por I.A

1.2. Sistema de Informação vs. Sistema Informático

- Essencialmente, **sistemas informáticos** fornecem a **base tecnológica**, enquanto **sistemas de informação** utilizam essa base para o processamento e manipulação de dados e comunicação.



1.3. Processo de Desenvolvimento de Software

❑ Desenvolvimento de software

- Processo de criação de programas que atendem a requisitos específicos de usuários ou organizações.
- Segue um conjunto de fases que ajudam a garantir que o software seja projetado corretamente e mantenha sua qualidade.



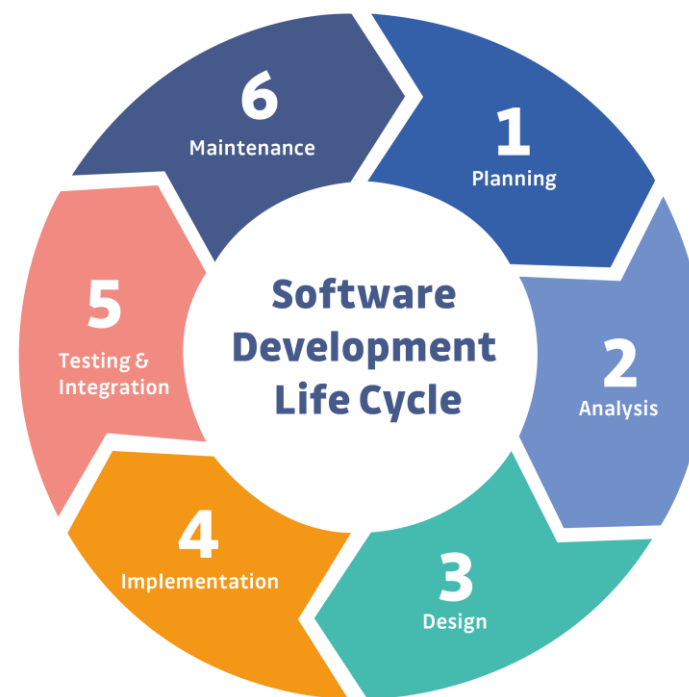
Fonte: [freepik.com](https://www.freepik.com)

1.3. Processo de Desenvolvimento de Software

❑ Fases do Desenvolvimento de software (ou SDLC)

1. **Planificação:** definição do escopo do projecto
2. **Análise:** entendimento dos requisitos do sistema e documentação
3. **Projeto:** definição da arquitetura do sistema, desenhando como os componentes interagirão e como os dados serão organizados.
4. **Implementação:** codificação e integração do software conforme especificado no projeto.
5. **Testes:** verificação das funcionalidades do software, assegurando que atenda aos requisitos especificados.
6. **Manutenção:** Atualizações, correção de *bugs* (erros) e melhorias contínuas no software após sua entrega.

Nota: A **análise** é uma fase crítica, pois define o que o sistema deve fazer e estabelece as bases para o restante do processo de desenvolvimento. Uma boa análise minimiza erros e garante que o software final atenda às necessidades dos usuários.



Fonte: blog.gitguardian.com

1.3. Processo de Desenvolvimento de Software

❑ Fases do Desenvolvimento de software (ou SDLC)

- De modo geral, o desenvolvimento de software obedece a uma **estrutura** baseada nas fases do **SDLC**. Essa **estrutura** varia de acordo ao **modelo de desenvolvimento utilizado**.
- Dentre os modelos de desenvolvimento, podemos citar: Cascata, Ágil, Espiral, Evolutivo, Incremental, ...

Sugestão de leitura sobre **modelos de desenvolvimento de software**:

- ✓ Ruparelia, N. B. (2010). Software development lifecycle models. ACM SIGSOFT Software Engineering Notes, 35(3), 8-13.

1.4. Introdução à Análise de Sistemas Orientados a Objetos

- ❑ **Baseia-se nos princípios da programação orientada a objetos, que ajudam a modelar sistemas mais próximos da realidade, utilizando conceitos que facilitam a manutenção e a evolução dos softwares.**

Princípios da Orientação a Objetos

- **Abstração:** Foco nos aspectos essenciais de uma entidade, ignorando detalhes irrelevantes.
- **Encapsulamento:** Proteção dos dados internos de um objeto, expondo apenas o necessário.
- **Herança:** Capacidade de criar novas classes baseadas em classes existentes, reutilizando código.
- **Polimorfismo:** Habilidade de tratar objetos de diferentes classes de forma unificada, através de interfaces comuns.

Nota: teremos um espaço sobre os princípios da O.O

1.4. Introdução à Análise de Sistemas Orientados a Objetos

Algumas vantagens da abordagem Orientada a Objetos

- Facilita a reutilização de código através;
- Melhora a manutenção ao permitir que os sistemas sejam facilmente modificados e estendidos;
- Proporciona maior flexibilidade e escalabilidade em comparação com abordagens estruturadas;
- **Modularidade;**
- ...

...

Sequência -> **M02**
