# Análise de Sistemas e Engenharia de Software

## Capítulo 1 - Introdução ao Software e à Engenharia de Software

1. Defina o que é software e cite os três principais tipos.
2. Quais são as principais diferenças entre software genérico e software personalizado?
3. Por que a Engenharia de Software é importante no desenvolvimento de sistemas?
4. Explique por que "Engenharia de Software ≠ Programar".

## Capítulo 2 - Processos de Desenvolvimento de Software

1. O que é um processo de software e quais são as fases comuns independentemente do modelo adotado?
2. Descreva o Modelo em Cascata (Waterfall), suas vantagens e desvantagens.
3. Explique como funciona o Modelo de Prototipagem e em que situações ele é mais útil.
4. Descreva o Modelo Espiral e suas características principais.
5. O que é o Unified Process (UP) e quais são suas fases principais?
6. Quais são as atividades fundamentais de um processo de software?

## Capítulo 3 - Metodologias Ágeis de Desenvolvimento

1. Quais são os quatro valores fundamentais do Manifesto Ágil?
2. Liste e explique três princípios ágeis fundamentais.
3. Quais são as práticas principais do Extreme Programming (XP) e como ele contribui para a qualidade do software?
4. Descreva a estrutura do Scrum, seus papéis principais e reuniões.
5. Como as metodologias ágeis devem ser adaptadas ao contexto organizacional?

## Capítulo 4 - Análise e Elicitação de Requisitos

1. O que é Engenharia de Requisitos e quais são suas fases principais?
2. Quais são algumas técnicas eficazes para elicitação de requisitos?
3. Explique o que são personas e como elas ajudam no desenvolvimento de software.
4. Como são estruturadas as histórias de utilizador e qual o seu papel no desenvolvimento ágil?
5. Por que a validação de requisitos é importante e que técnicas podem ser usadas?

## Capítulo 5 - Modelação de Sistemas

1. Por que a modelação é importante no desenvolvimento de sistemas?
2. Quais são os principais tipos de modelos na UML e sua finalidade?
3. Explique os conceitos-chave da modelação orientada a objetos.
4. Descreva o ciclo prático da modelação orientada a objetos com um exemplo simples.

## Capítulo 6 - Arquitetura de Software

1. O que é arquitetura de software e por que é importante?
2. Quais são as características desejáveis numa boa arquitetura de software?
3. Explique o que são arquiteturas distribuídas e seus principais desafios.
4. Descreva o padrão MVC (Model-View-Controller) e seu funcionamento.
5. O que são trade-offs em decisões arquiteturais e dê exemplos.

## Capítulo 7 - Computação em Nuvem

1. O que é computação em nuvem e quais são suas características principais?
2. Compare os três principais modelos de serviço na cloud: IaaS, PaaS e SaaS.
3. Explique a diferença entre arquiteturas multi-tenant e multi-instance na nuvem.

## Capítulo 8 - Arquitetura de Microserviços

1. O que são microserviços e quais suas características principais?
2. Como é feita a comunicação entre microserviços e quais os padrões comuns?
3. Explique como a arquitetura de microserviços facilita a implementação contínua.

## Capítulo 9 - Segurança e Privacidade em Software

1. Quais são os três princípios básicos da segurança da informação (triângulo CIA)?
2. Liste e explique três ameaças comuns à segurança de software.
3. Diferencie autenticação e autorização em sistemas de software.
4. Explique os tipos de encriptação e suas aplicações em software.
5. Quais são os princípios fundamentais do RGPD (Regulamento Geral sobre a Proteção de Dados)?

## Capítulo 10 - Programação Confiável

1. Quais são algumas boas práticas essenciais para programação confiável?
2. Por que a validação de entradas é crucial e que tipos de validação devem ser feitos?
3. Explique a importância da gestão de erros e as boas práticas associadas.
4. O que são padrões de desenho (design patterns) e dê exemplos.
5. O que é refatoração e qual a sua importância no desenvolvimento de software?

## Capítulo 11 - Testes de Software

1. Quais são os principais tipos de testes de software e seus objetivos?
2. Explique o ciclo do TDD (Test-Driven Development) e suas vantagens.
3. Qual a importância das revisões de código (code reviews) e como devem ser conduzidas?
4. O que são testes de segurança e que tipos existem?

## Capítulo 12 - DevOps e Gestão de Código

1. O que é DevOps e quais são os conceitos-chave de CI/CD?
2. Explique o funcionamento prático de um pipeline de CI/CD e seus benefícios.
3. Quais são os conceitos principais de controlo de versões e como funciona um fluxo de trabalho comum?
4. Quais são as métricas importantes para monitorização de sistemas em produção?
5. Quais são os tipos de manutenção em sistemas de software e como funciona o ciclo de melhoria contínua?

## Capítulo 13 - Desenho de Interfaces Humanas

1. Quais são os princípios fundamentais de usabilidade?
2. Descreva as boas práticas para desenho de formulários e relatórios.
3. O que são diagramas de diálogo e qual sua importância no desenho de interfaces?

## Capítulo 14 - Implementação e Operação de Sistemas

1. Quais são as boas práticas de codificação para desenvolvimento de software confiável?
2. Quais são os tipos de testes que devem ser integrados à codificação?
3. Quais são os tipos essenciais de documentação no desenvolvimento de software?
4. Como deve ser estruturado o suporte ao utilizador após a implementação de um sistema?

## Capítulo 15 - Gestão de Projetos de Software

1. Explique os desafios da estimativa em projetos de software e as técnicas utilizadas.
2. Como se deve abordar a gestão de riscos em projetos de software?
3. Explique os princípios W5HH de gestão eficaz segundo Barry Boehm.

## Capítulo 16 - Qualidade de Software

1. O que define a qualidade de software segundo o modelo ISO/IEC 25010?
2. O que é Garantia da Qualidade (QA) e quais são as técnicas mais comuns?
3. Explique os diferentes tipos de custos associados à qualidade de software.

## Capítulo 17 - Melhoria de Processos e Tendências Futuras

1. O que é Melhoria de Processos de Software (SPI) e como funciona o ciclo de melhoria contínua?
2. Descreva os níveis de maturidade do CMMI (Capability Maturity Model Integration).
3. Quais são as tendências emergentes em desenvolvimento de software e por que a adaptabilidade é crucial?