

# Engenharia de Software: Arquitetura de Software e Design de Software

Discente: Gustavo Santos Teixeira

Matrícula: 202200149

Com os fundamentos apresentados ao longo dos capítulos, é possível compreender que os requisitos de software funcionam como a base conceitual que orienta o comportamento esperado do sistema, descrevendo tanto as funcionalidades essenciais quanto as restrições que determinam a forma como essas funcionalidades devem ocorrer. Nesse contexto, os requisitos podem assumir naturezas distintas, como requisitos funcionais, que especificam as ações observáveis que o sistema precisa executar, e requisitos não funcionais, que impõem condições de qualidade, desempenho, confiabilidade ou segurança, além das restrições tecnológicas que moldam as possibilidades de implementação. Também se destacam os requisitos derivados, que surgem das decisões internas de projeto e passam a influenciar etapas subsequentes do desenvolvimento.

Ao longo do processo de engenharia de requisitos, observa-se que a elicitação depende de técnicas diversas, entrevistas, *workshops*, observação, prototipagem que procuram captar as necessidades reais das partes interessadas, enquanto a análise busca garantir que esses requisitos sejam inequívocos, testáveis, completos e consistentes. A validação, por sua vez, assegura que tais informações representem corretamente os objetivos dos *stakeholders*, utilizando revisões, inspeções, simulações e protótipos para evitar desalinhamentos. Como nem todos os requisitos possuem a mesma relevância ou urgência, a priorização torna-se necessária para orientar decisões de escopo,

*trade-offs* e direcionamento de esforços ao longo do projeto. Assim, o gerenciamento contínuo desses requisitos, incluindo rastreabilidade, controle de mudanças e análise de impacto, sustenta a estabilidade do projeto diante da natural evolução das demandas.

No campo da arquitetura e do design, observa-se que sistemas complexos requerem documentações estruturadas em múltiplas visões, permitindo que diferentes *stakeholders* compreendam aspectos físicos, lógicos e organizacionais da solução. A arquitetura estabelece as decisões estruturais que moldam todo o sistema, ao passo que o design aprofunda os detalhes internos de cada componente, definindo algoritmos, interfaces e estratégias de implementação. Nesse processo, requisitos arquitetonicamente significativos influenciam diretamente a estrutura geral, exigindo a avaliação constante de *trade-offs* entre atributos como desempenho, segurança ou escalabilidade. Além disso, princípios como coesão, acoplamento e encapsulamento orientam a construção de módulos mais robustos e reutilizáveis, enquanto padrões de design, como o *Observer* para notificação desacoplada, fornecem soluções consolidadas para problemas recorrentes.

Desse modo, a integração entre requisitos, arquitetura e design forma um ciclo contínuo em que as necessidades do usuário moldam o sistema e, simultaneamente, as decisões estruturais realimentam a forma como essas necessidades são interpretadas e implementadas, garantindo coerência, qualidade e alinhamento entre o que se pretende desenvolver e o que, de fato, será entregue.