De acordo com Sommerville (2011), uma diferença importante entre o desenvolvimento de software profissional e amador é a complementação do programa em si com toda a documentação associada e dados de configurações necessários para fazer esse programa operar corretamente.

Apesar disso, alguns fundamentos se aplicam a todos os tipos de software, dentre eles está a especificação de requisitos, que também funciona como um gerenciamento de expectativas do cliente. Os requisitos de um sistema são as descrições do que um sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento (SOMMERVILLE, 2011).

Wazlawick (2014) expõe a necessidade de, além de organizar a estrutura de um sistema em camadas e módulos, organizar o código implementado dentro deles. Os programadores devem aprender as técnicas que podem levar à construção de código reusável e com baixo acoplamento.

Os conceitos básicos da UML (do inglês *Unified Modeling Language)* incluem diversos tipos de diagramas que representam diferentes aspectos do sistema em desenvolvimento. Esses diagramas podem ser agrupados em duas categorias principais: diagramas estruturais e diagramas comportamentais, como descrito por Booch *et al.* (2005).

Diagramas estruturais abordam diagrama de classes, diagrama de objetos, e diagrama de componentes; enquanto diagramas comportamentais seriam o diagrama de casos de uso, diagrama de sequência e diagrama de atividade, por exemplo (BOOCH et al., 2005).

Para Sommerville (2011), casos de uso são uma técnica de descoberta de requisitos, identificando os atores (que podem ser pessoas ou outro sistema) envolvidos em uma interação e nomeando-a. São documentados em seu diagrama com informações adicionais que descrevem a interação com o sistema.

Os casos de uso expandidos podem ser utilizados como comandos (métodos que são ativados por um evento de sistema, ou seja, como uma reação a uma ação do usuário) e consultas (verificação de informação já armazenada) de sistema. A sequência de eventos e respostas em um caso de uso são representados por um diagrama de sequência, com elementos como atores, interfaces do sistema e objetos (WAZLAWICK, 2014).

Já diagramas de classe serão definidos em diversos níveis de detalhamento. Em suma, mostra as classes (definição geral de um tipo de objeto do sistema) e as associações entre elas. Inicialmente os objetos representam algo do mundo real, entretanto enquanto uma aplicação é desenvolvida pode ser necessário definir objetos adicionais de implementação que são usados para fornecer a funcionalidade requerida do sistema. Por fim, os diagramas de implantação da UML mostram como os componentes de software são distribuídos em diferentes plataformas de hardware (SOMMERVILLE, 2005).

Para Wazlawick (2014), nem todos os diagramas devem ser usados durante o desenvolvimento de um sistema, apenas aqueles que representam informações úteis para o projeto são recomendados. Neste projeto foram desenvolvidos diagramas de caso de uso, sequência, classes e implementação.

Sommerville, I. **Engenharia de Software.** 9 ed. São Paulo: Pearson Prentice Hall, 2011.

WAZLAWICK, Raul S. **Análise e Design Orientados a Objetos para Sistemas de Informação: Modelagem com UML, OCL e IFML**. 3. ed. Rio de Janeiro: GEN LTC, 2014. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/9788595153653/>. Acesso em: 21 mai. 2025.

Booch, G., Jacobson, I., Rumbaugh, J. (2005). **UML: guia do usuário.** 2. ed. Rio de Janeiro: Campus.