Design Patterns

(Padrões de Projeto)

Arquitetura de Software

O que é Arquitetura de Software?

É a estrutura fundamental de um sistema de software, que compreende seus componentes, as relações entre eles, e os princípios e diretrizes que governam seu design e evolução ao longo do tempo. É uma representação de alto nível de um sistema, focando mais na estrutura e organização do que nos detalhes de implementação.

Arquitetura de Software

Principais elementos

Componentes: São as partes funcionais do sistema, como módulos, classes, funções ou serviços. Cada componente é responsável por uma funcionalidade específica.

Conexões: Definem como os componentes se comunicam e interagem entre si. Isso pode incluir chamadas de métodos, troca de mensagens, acesso a bancos de dados, etc.

Princípios e Diretrizes: Regras e boas práticas que orientam as decisões de design e garantem que o sistema atenda aos requisitos funcionais e não funcionais.

Arquitetura de Software

Padrões Arquiteturais

Monolítica: Toda a funcionalidade do sistema é implementada em um único bloco de código. Simples de desenvolver, mas difícil de escalar e manter à medida que cresce.

Três Camadas: Organiza o sistema em três camadas: apresentação, lógica de negócios e dados. Promove a escalabilidade e a separação de responsabilidades.

Cliente-Servidor: Divide o sistema em clientes (interface do usuário) e servidores (lógica de negócios e dados). Facilita a distribuição de tarefas e a segurança.

Microsserviços: Divide o sistema em pequenos serviços independentes, cada um responsável por uma funcionalidade específica. Oferece escalabilidade granular e flexibilidade tecnológica.

Arquitetura de Software

Padrões Arquiteturais

Orientada a Serviços (SOA): Organiza a funcionalidade do software como serviços distintos e reutilizáveis. Foca na reusabilidade e interoperabilidade.

Arquitetura de Software

Resumindo...

- É a base sobre a qual um sistema é construído.
- Define os componentes principais do sistema, suas interações e os princípios e diretrizes para seu design e evolução.
- Foca na estrutura e organização e não nos detalhes de implementação.

O que são Padrões de Projeto?

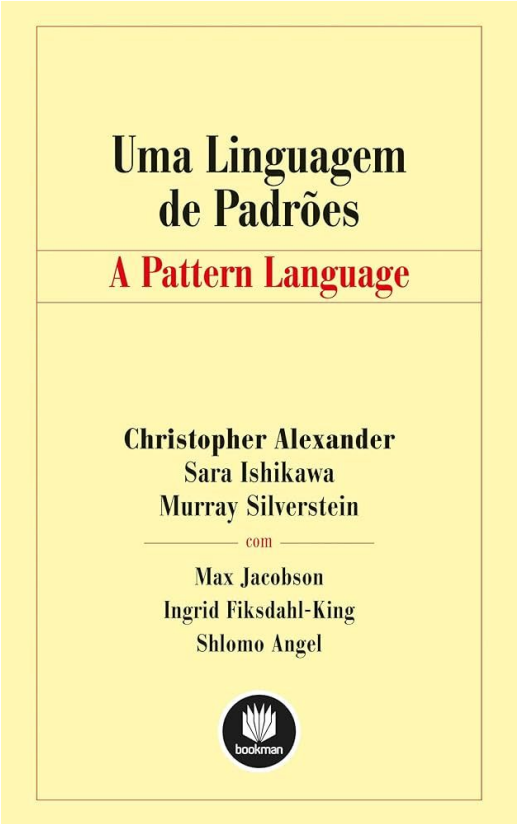
São soluções típicas para **problemas comuns** em projetos de software.

1. Quando você precisa garantir que apenas uma instância de uma classe seja criada e que ela seja acessível globalmente;
2. Quando você precisa notificar objetos sobre mudanças em outro objeto, sem que eles estejam fortemente acoplados;
3. Quando você precisa adicionar funcionalidades a um objeto sem modificar sua classe.
4. Etc...

O que são Padrões de Projeto?

Descrito por Christopher Alexander no livro Uma Linguagem de Padrões.

“Cada padrão descreve um problema no nosso ambiente e o cerne da sua solução, de tal forma que você possa usar essa solução mais de um milhão de vezes, sem nunca fazê-lo da mesma maneira.”

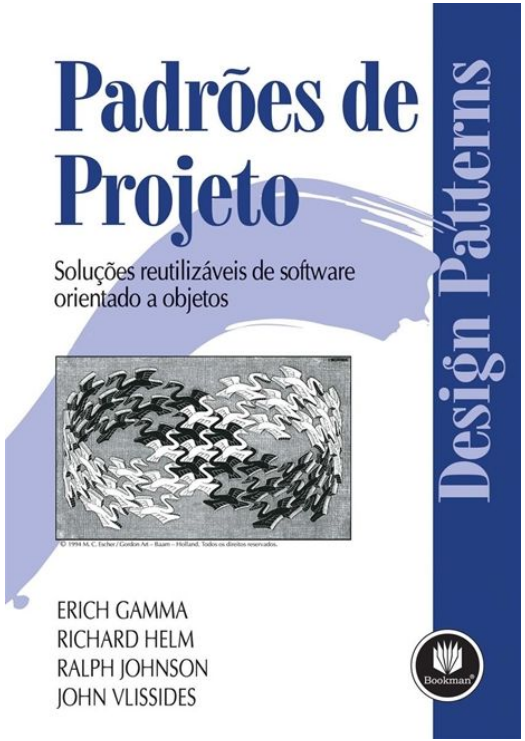


1977

Arquitetura e Urbanismo

O que são Padrões de Projeto?

Seguindo a mesma ideia de Christopher, Erich Gamma, John Vlissides, Ralph Johnson e Richard Helm escreveram o livro Padrões de Projeto - Soluções reutilizáveis de software orientado a objetos.



1994

Elementos essenciais de um padrão

O que geralmente está presente na descrição de um padrão:

- O **Propósito** do padrão descreve brevemente o problema e a solução.
- A **Motivação** explica a fundo o problema e a solução que o padrão torna possível.
- As **Estruturas** de classes mostram cada parte do padrão e como se relacionam.
- **Exemplos de código** em uma das linguagens de programação populares tornam mais fácil compreender a ideia por trás do padrão.

Classificação dos padrões

Padrões podem ser categorizados por seu **propósito** ou **intenção**.

Criacionais,
fornecem mecanismos de criação de objetos que aumentam a flexibilidade e a reutilização de código.

Factory Method, Abstract Factory e Builder

Comportamentais,
cuidam da comunicação eficiente e da assinalação de responsabilidades entre objetos.

Chain of Responsibility, Command e Interpreter

Estruturais,
explicam como montar objetos e classes em estruturas maiores, enquanto ainda mantém as estruturas flexíveis e eficientes.

Adapter, Bridge e Composite