

- 1) O padrão *Factory* é um padrão de criação que fornece uma interface para criar objetos em uma superclasse, mas permite que as subclasses decidam qual classe específica será instanciada. Ele oculta os detalhes de instanciamento e centraliza a criação de objetos, tornando o código mais flexível e facilitando a manutenção, especialmente quando novas classes são adicionadas.
- 2) O padrão *DAO* (Data Access Object) é utilizado para separar a lógica de POO do acesso aos dados, criando uma camada que interage diretamente com o banco de dados. Isso permite que operações como salvar, atualizar, excluir e buscar sejam feitas de forma centralizada e independente da lógica de aplicação, tornando o código mais organizado e fácil de manter.
- 3) O padrão *MVC* (Model-View-Controller) divide a aplicação em três componentes: *Model*, que representa os dados e a lógica de negócios; *View*, que é a interface com o usuário e exibe os dados; e *Controller*, que gerencia a interação entre o Model e a View. Esse padrão facilita a organização do código e melhora a separação de responsabilidades, tornando a aplicação mais modular e fácil de manter.