

## Questão 2: Plataforma de Pagamentos (Factory Method)

Uma plataforma de pagamentos deve permitir a integração com diferentes meios. A escolha do meio de pagamento deve ser configurável e o sistema precisa instanciar dinamicamente o processador correspondente sem que o código principal da aplicação precise conhecer detalhes da criação de cada objeto.

### Justificativa da Escolha do Padrão: Factory Method

O padrão **Factory Method** (neste caso, implementado como uma *Simple Factory*, que é uma variação comum e direta). Ele provê uma interface para criar objetos em uma superclasse, mas permite que as subclasses (ou um método estático, na abordagem *Simple Factory*) alterem o tipo de objetos que serão criados.

- **Encapsulamento da Criação:** A lógica de qual classe concreta instanciar (`new CartaoCreditoProcessador()`, `new PixProcessador()`, etc.) fica centralizada em um único lugar (a Factory), escondida do código cliente.
- **Redução do Acoplamento:** O código cliente depende apenas da interface `ProcessadorPagamento` e da Factory. Ele não precisa saber o nome de todas as classes concretas de processadores, facilitando a adição de novos meios de pagamento no futuro sem alterar o cliente.
- **Instanciação Dinâmica:** O cliente pode decidir qual processador criar em tempo de execução, simplesmente passando um parâmetro (como uma string) para a Factory.