

Atividade 02 - 03/06

Victor Hugo Vieira Cruz - 20231011090296

1 - Explique como o uso do padrão Model-View-Controller (MVC) facilita a manutenção e evolução de sistemas que possuem múltiplas formas de visualização e interação com os dados.

O padrão MVC facilita bastante a manutenção de sistemas mais complexos porque separa o que cada parte do sistema precisa fazer. O Modelo cuida dos dados e da lógica, a Visão é o que o usuário vê e interage, e o Controlador faz a ponte entre os dois. Essa divisão deixa o sistema mais organizado e permite que, por exemplo, a gente mude a interface sem mexer na parte de regras de negócio. Isso ajuda muito quando o sistema precisa crescer, ganhar novas funcionalidades ou ser adaptado para outros tipos de interface.

2 - Em um sistema com alto volume de dados e múltiplas ferramentas acessando informações compartilhadas, por que o padrão Repostro pode ser mais adequado que o padrão Cliente-Servidor?

O padrão Repositório acaba sendo uma escolha melhor nesses casos porque centraliza o acesso aos dados. Isso ajuda a manter a consistência e facilita a gestão, principalmente quando várias ferramentas ou serviços diferentes estão acessando as mesmas informações. Com ele, também é mais fácil aplicar cache ou otimizações de consulta. Já o Cliente-Servidor pode sobrecarregar o servidor com muitas requisições diretas, o que pode impactar o desempenho em sistemas maiores.

3 - Cite dois exemplos de sistemas reais nos quais a Arquitetura em Camadas seria a escolha ideal. Justifique com base nas vantagens apresentadas.

1. Sistemas de Gestão Empresarial (ERP): A separação por camadas permite que cada parte do sistema (interface, regras de negócio, dados) seja desenvolvida e mantida de forma independente. Isso é ideal em ERPs, que são sistemas grandes e complexos.
2. Plataformas de E-commerce: Um e-commerce precisa de uma camada de apresentação para o site, uma de lógica para tratar pedidos e pagamentos, e uma de dados para gerenciar os produtos e usuários. Com essa arquitetura, fica mais fácil escalar e adicionar funcionalidades como recomendação de produtos ou integração com meios de pagamento.

4 - Um sistema de processamento de pedidos precisa aplicar validações, calcular impostos, gerar relatórios e emitir notas iscais. Como o padrão Duto e Filtro pode ser usado para estruturar esse sistema?

O padrão Duto e Filtro pode organizar essas etapas como uma sequência de filtros, onde cada um faz uma tarefa específica: um filtro valida os dados, o outro calcula impostos, outro gera relatórios, e assim por diante. Isso deixa o sistema modular e fácil de manter, já que cada parte faz só o que precisa e pode ser alterada ou substituída sem afetar o resto.

5 - Explique por que o padrão Cliente-Servidor é considerado altamente escalável. Quais são os principais desafios na implementação dessa

escalabilidade?

Esse padrão é escalável porque permite que muitos clientes acessem o sistema ao mesmo tempo, usando um servidor central. Para acompanhar o crescimento da demanda, é possível aumentar a capacidade do servidor (verticalmente) ou adicionar mais servidores (horizontalmente). O desafio está em gerenciar esse crescimento: garantir que o sistema continue rápido, lidar com a sincronização entre servidores e manter a segurança e integridade dos dados.

6 - Em que cenários a combinação de dois ou mais padrões arquiteturais é recomendada? De um exemplo prático baseado na apresentação.

Combinar padrões faz sentido quando o sistema é complexo e tem necessidades diferentes. Um bom exemplo é um CMS (sistema de gerenciamento de conteúdo), que pode usar MVC para organizar a interface e a lógica de negócio, e Repositório para lidar com os dados. Assim, o sistema fica mais flexível, organizado e preparado para crescer.

7 - Compare o padrão MVC com o padrão Arquitetura em Camadas em termos de separação de responsabilidades. Em que situações um seria mais vantajoso que o outro?

Os dois padrões ajudam a organizar o sistema, mas com focos diferentes. O MVC separa as responsabilidades de forma mais voltada à interação do usuário: modelo, visualização e controle. É ideal para aplicações com interface dinâmica. Já a Arquitetura em Camadas organiza o sistema em níveis (interface, lógica, dados), o que é ótimo para sistemas grandes com regras de negócio complexas. Em geral, MVC é mais vantajoso para apps interativos, e Camadas para sistemas corporativos mais robustos.

8 - Quais critérios devem ser considerados na escolha de um padrão arquitetural? Relacione esses critérios com os cinco padrões estudados.

Alguns critérios importantes são:

- Complexidade: Sistemas complexos se beneficiam do MVC ou da Arquitetura em Camadas pela separação de responsabilidades.
- Escalabilidade: Cliente-Servidor e Repositório funcionam bem em sistemas que precisam crescer e suportar muitos usuários.
- Facilidade de manutenção: MVC e Camadas ajudam a isolar mudanças e manter o código organizado.
- Desempenho: Duto e Filtro é útil em sistemas que precisam processar dados em várias etapas, como validações ou transformações.
- Experiência do usuário: MVC é ideal para interfaces dinâmicas que precisam responder rapidamente às ações do usuário.
- Integração com outros sistemas: Repositório facilita a comunicação com outras ferramentas ou bancos de dados.

9 - Quais são as principais vantagens e desvantagens do padrão Repositório em sistemas colaborativos?

- Vantagens:
 - Centraliza o acesso aos dados, o que ajuda a manter tudo sincronizado.
 - Facilita a integração com outras ferramentas e serviços.

- Permite aplicar otimizações como cache.
- Torna a manutenção mais simples, já que as mudanças ficam concentradas no repositório.
- Deixa o sistema mais flexível, porque novos serviços podem usar os dados sem precisar mudar como tudo é acessado.
- Garante mais consistência, pois todos usam a mesma fonte de dados.
- Desvantagens:
 - Pode virar um gargalo se não for bem dimensionado, já que todos os acessos passam por ele.
 - Exige atenção extra com concorrência e sincronização quando muitos acessos acontecem ao mesmo tempo.

10 - Um sistema de biblioteca digital que oferece interface web, autenticação de usuários, acesso a catálogos e serviços de empréstimo pode se beneficiar de quais padrões arquiteturais? Justifique.

Esse tipo de sistema pode aproveitar vários padrões combinados:

- MVC: Para separar a interface da lógica e dos dados, o que ajuda na manutenção da interface web.
- Arquitetura em Camadas: Para organizar o sistema em níveis, facilitando a adição de novas funcionalidades como autenticação e serviços de empréstimo.
- Repositório: Para gerenciar os dados dos usuários e dos livros de forma centralizada, garantindo consistência.
- Cliente-Servidor: Para permitir que muitos usuários acessem o sistema ao mesmo tempo, com o servidor respondendo às requisições.
- Duto e Filtro: Para organizar etapas como login, busca no catálogo, validação de empréstimos, etc. Usando esses padrões juntos, o sistema se torna mais organizado, escalável e fácil de manter.