Aula 9: Arquitetura MVC

# **Padrão Arquitetural**

padrão **DAO**

concentração das operações de acesso a banco de dados, que evita a multiplicação de comandos SQL ao longo de todo o código

**Pipes/Filters**

na qual um programa ou componente fornece uma saída que servirá de entrada para outro comando ou componente em uma sequência de processamento

# **Arquitetura MVC**

o Model-View-Control, ou **MVC**, acabou dominando o mercado de desenvolvimento no que se refere às aplicações cadastrais. Ele promove a divisão do sistema em três camadas: persistência de dados (**Model**), interação com o usuário (**View**) e processos de negócios (**Control**).

# **Padrão Front Control**

este padrão não pertence à camada Control, mas sim à camada **View**, pois não tem nenhuma relação com as regras de negócios, tendo como função primordial o simples controle de navegação do sistema.

# **Sessões e Controle de Acesso**

As sessões são de grande utilidade no ambiente Web, provendo uma forma de manutenção de estados na troca de páginas, pois ao contrário dos sistemas desktop, a cada nova página temos outro conjunto de variáveis na memória, desconsiderando-se todas aquelas existentes antes de a requisição ser efetuada.

|  |  |  |
| --- | --- | --- |
|  | | |
|  | |  |
|  | | |
| |  | | --- | | [**1.**](https://simulado.estacio.br/bdq_simulados_exercicio.asp#) | | |  | | --- | | Com relação aos conceitos de arquitetura de aplicações, é correto afirmar que o MVC é um padrão | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| Certo |  | de arquitetura de aplicações que visa separar a "lógica da aplicação", a "interface do usuário" e a "persistência de dados". |
|  |  | de objeto que executa uma operação quando ele recebe uma solicitação (mensagem) correspondente de um outro objeto. |
|  |  | de desenvolvimento cuja finalidade é definir uma interface na mesma camada da "lógica da aplicação". |
|  |  | de operação dirigida para uma classe e não para um objeto individual. |
|  |  | que garante que uma lógica de negócios possa ser acessada e visualizada por uma única interface. |

|  |  |
| --- | --- |
| Explicação:  A arquitetura MVC divide o aplicativo em 3 camadas: Model (Persistência), Control (Regras de Negócio) e View (Interfaces). Não pode ocorrer contato entre a View e a Model. | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | |  |
|  | | |
| |  | | --- | | [**2.**](https://simulado.estacio.br/bdq_simulados_exercicio.asp#) | | |  | | --- | | No ambiente JEE, a anotação PersistenceContext é utilizada para: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
|  |  | Definir uma interface remota para um EJB. |
| Certo |  | Instanciar um EntityManager a partir da unidade de persistência. |
|  |  | Definir um EJB do tipo Stateless. |
|  |  | Efetuar a conexão com as filas de uma mensageria a patir do MDB. |
|  |  | Acessar um EJB a partir de um Servlet. |

|  |  |
| --- | --- |
| Explicação:  O uso da anotação PersistenceContext permite instanciar um EntityManager a partir do nome da unidade de persistência, conforme as definições efetuadas no arquivo persistence.xml. | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | |  |
|  | | |
| |  | | --- | | [**3.**](https://simulado.estacio.br/bdq_simulados_exercicio.asp#) | | |  | | --- | | Sobre o controle de acesso é incorreto afirmar que: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
|  |  | Busca identificar os usuários legítimos das informações. |
|  |  | O controle de acesso permite manter a confidencialidade e integridade das informações. |
|  |  | Sessões são usadas |
| Certo |  | Por questões de segurança sessões não são usadas. |
|  |  | Logins são uma forma de controle de acesso. |

|  |  |
| --- | --- |
| Explicação:  A maioria das soluções de controle de acesso faz uso de sessões para aumentar o nível de segurança ao acesso à informação. Através das sessões, por exemplo, é possível ter o controle e o registro de acesso. Sessões ainda podem definir as operações que o usuário pode execer sobre um objeto. | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | |  |
|  | | |
| |  | | --- | | [**4.**](https://simulado.estacio.br/bdq_simulados_exercicio.asp#) | | |  | | --- | | O ferramental do NetBeans permite gerar de forma automática os EJBs responsáveis por invocar as diversas operações a serem efetuadas no banco a partir de entidades JPA. Ao fazer isto, ele segue um padrão de desenvolvimento denominado: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
|  |  | Intercept Filter |
|  |  | Front Control |
|  |  | Adapter |
|  |  | Singleton |
| Certo |  | Session Facade |

|  |  |
| --- | --- |
| Explicação:  É utilizado o padrão Session Facade, onde os Stateless Session Beans servem de fachada para as operações de banco intermediadas pelo JPA. | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | |  |
|  | | |
| |  | | --- | | [**5.**](https://simulado.estacio.br/bdq_simulados_exercicio.asp#) | | |  | | --- | | O protocolo HTTP não armazena estado, ou seja, os dados são perdidos na navegação entre as páginas. Para manter os dados referentes a uma conexão deve ser utilizado, no servidor, um objeto do tipo: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
|  |  | HttpRequest |
| Certo |  | HttpSession |
|  |  | HttpServletResponse |
|  |  | HttpServletRequest |
|  |  | HttpResponse |

|  |  |
| --- | --- |
| Explicação:  Os atributos de sessão, referentes a uma dada conexão, podem ser manipulados com o uso de um objeto do tipo HttpSession. No Servlet ele é obtido a partir da requisição com a chamada ao método getSession, enquanto no JSP ele é um objeto implícito denominado session. | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | |  |
|  | | |
| |  | | --- | | [**6.**](https://simulado.estacio.br/bdq_simulados_exercicio.asp#) | | |  | | --- | | Sobre os Servlets é incorreto afirmar que: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
|  |  | É uma classe Java usada para estender as funcionalidades de um servidor. |
|  |  | Não fazem uso de protocolos |
|  |  | São frequentemente usados para processar ou armazenar dados que foram submetidos de um formulário HTML. |
|  |  | Fornecem conteúdo dinâmico, como os resultados de uma consulta a um banco de dados. |
| Certo |  | Fazem uso exclusivo do protocolo HTTP. |

|  |  |
| --- | --- |
| Explicação:  Os servlets normalmente utilizam o protocolo HTTP, apesar de não serem restritos a ele. | |