

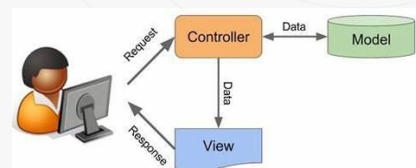


Programação Back End
Helio Rangel

A arquitetura MVC

A arquitetura **MVC** tem sido largamente utilizada e consagrada como forma mais adequada, segura e produtiva de se desenvolver aplicações e sistemas;

- **View** - Páginas em formato com resposta HTML (JSP,ASP,PHP).
- **Controller** - Controle da regra de negócio.
- **Model** - Representação dos elementos do seu domínio e interação com as ferramentas de persistência.



As linguagens mais usadas no Back End



PHP (Hypertext Preprocessor)

- Como vimos, é uma linguagem de script *open source* de uso geral, muito utilizada, e especialmente pensada para o desenvolvimento WEB e que pode ser editada dentro da página HTML.
- É uma das linguagens WEB mais utilizadas e produtivas. Existem *frameworks* disponíveis que permitem desenvolver em PHP utilizando conceitos de Orientação a Objetos e que viabilizam uma separação mais clara entre os projetos de Front End e Back End.
- O que distingue o PHP do Javascript no lado do cliente é que o código é executado no servidor, gerando a página HTML enviadas para o navegador. O navegador recebe os resultados da execução desse script, mas não conhece o código fonte que deu origem a página.
- Um ponto forte do PHP é que ele não é considerado complexo, e mesmo assim, oferece recursos de desenvolvimento bastante avançados.



As linguagens mais usadas no Back End



Java

- Uma das primeiras linguagens a ser utilizada na WEB através dos Applets e Servlets.
- Aplicações web, são sistemas ou sites onde grande parte da programação fica hospedada em servidores na internet e o usuário normalmente não precisa ter nada instalado em sua máquina para utilizá-las além de um navegador.
- Na plataforma Java, esse modelo foi implementado através da Interface de Programação de Aplicação (API) de *Servlets*. Um *Servlet* estende a funcionalidade de um servidor WEB para servir páginas dinâmicas aos navegadores, utilizando o protocolo HTTP.
- Os servidores web são chamados de *Servlet Container*, pois implementam a especificação de *Servlet*.
- O servidor converte a requisição em um objeto do tipo `HttpServletRequest` que é então passado aos componentes web, que podem executar qualquer código Java para que possa ser gerado um conteúdo dinâmico. Em seguida, o componente web devolve um objeto `HttpServletResponse`, que representa a resposta ao cliente.



As linguagens mais usadas no Back End



C# - ASP.NET

- É uma solução integrada proprietária Microsoft que permite o desenvolvimento de aplicações WEB utilizando linguagens como C#, VB.NET, F# entre outras. A linguagem atualmente mais em uso na plataforma .NET é o C# (*C Sharp*).
- O C# é uma linguagem 100% orientada a objetos com algumas semelhanças com C++ e Java. Por ser um “pacote” de desenvolvimento, oferece uma série de possibilidades e “*templates*” para sistemas WEB com as mais diversas aplicações.
- O desenvolvimento de sistemas em ASP.NET é uma experiência intuitiva e com uma excelente curva de aprendizagem, veloz e produtiva. Um ponto considerado fraco na plataforma ASP.NET é o editor de páginas para uso dos desenvolvedores de **Front End**. Contudo, os desenvolvedores **Back End** não terão do que se queixar.



A persistência



- Naturalmente precisamos armazenar de forma prática e segura os dados gerados e utilizados pelas operações do sistema. Chamamos esta operação de persistência;
- Normalmente a persistência dos dados fica a cargo do **Back End**. O mais comum é utilizar um banco de dados relacional de mercado. Bancos como MySql, Oracle, SqlServer, PostgreSQL entre outros são os mais utilizados e sua escolha está associada a razões econômicas e/ou técnicas.
- Bancos especiais do tipo NoSQL ou serialização de objetos também não são incomuns. O volume de dados, a necessidade de consultas e atualizações rápidas, a segurança, confiabilidade, praticidade, custo, escalabilidade e disponibilidade são alguns dos fatores levados em conta no momento da decisão de qual mecanismo de persistência será adotado.

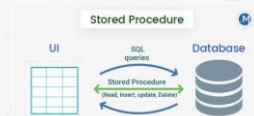


A persistência e o MVC

- A arquitetura MVC recomenda uma separação clara entre a persistência e as demais camadas do sistema. Esta separação ajuda bastante o desenvolvedor Back End a pensar separadamente nos processos de persistência, como comandos SQL, conexões etc. e as regras de negócio.
- Em linguagens OO, recomenda-se que cada tabela identidade possua sua própria classe de controle onde podemos codificar tudo que for necessário para persistência dos dados de forma que as classes Back End dos formulários fiquem livres de código SQL.



A persistência via *Stored Procedures*



- É um conjunto de comandos em **SQL** que podem ser executados de uma só vez, como em uma função. Ele armazena tarefas repetitivas e aceita parâmetros de entrada para que a tarefa seja efetuada de acordo com a necessidade individual.
- Um Stored Procedure pode reduzir o tráfego na rede, melhorar a performance de um banco de dados, criar tarefas agendadas, diminuir riscos, criar rotinas de processamento etc.



Stored Procedures - Exemplo

```
1 USE BancoDados
2 GO
3 CREATE PROCEDURE Busca --- Declarando o nome da procedure
4 @CampoBusca VARCHAR (20) --- Declarando variável (note que utilizamos o @ antes do nome da variável)
5 AS
6 SELECT Codigo, Descrição --- Consulta
7 FROM NomeTabela
8 WHERE Descricao = @CampoBusca --- Utilizando variável como filtro para a consulta
```

Para executar uma procedure basta utilizar a cláusula EXECUTE seguido pelo nome da procedure e na frente o valor do argumento caso haja.

```
1 EXECUTE Busca 'DEVMEDIA'
```



Stored Procedures

Quando se recomenda

- Quando temos várias aplicações escritas em diferentes linguagens, ou rodam em plataformas diferentes, porém executam a mesma função;
- Quando damos prioridade à consistência e segurança;
- Desempenho é item crítico.

Desvantagens

- Nas *Stored Procedures* acabamos implementando regras de negócio. Desta forma ficamos mais dependentes dos bancos de dados e diminuimos a independência das camadas MVC.
- Em caso de necessidade, a migração de uma tecnologia de persistência para outra pode ficar mais trabalhosa, cara e complexa.
- Em algumas organizações o time de back end não pode desenvolver e manter as *stored procedures* pois são de responsabilidade dos DBA's.

