

< VOLTAR



Desenvolvimento web

Capacitar para o desenvolvimento de aplicações dinâmicas para internet ou intranet, em ambiente empresarial, utilizando uma linguagem de Programação Orientada a Objetos (POO).

NESTE TÓPICO

- > Introdução
- > Sites dinâmicos e tecnologias de desenvolvimento
- > A web 2.0
- > Linguagens para web
- > Linguagens interpretadas e compiladas
- > Execução no lado do cliente e no lado do servidor
- > Referências



Introdução

Para que esse objetivo seja alcançado, serão apresentados conceitos imprescindíveis referentes ao funcionamento da **Web** com Java, conceitos de uma aplicação **Web** ou de geração de conteúdo dinâmico, **Web Servlet**, **Browser**, ciclo de vida do **servlet**, métodos de serviço, solicitações, respostas e alguns outros recursos **Web**. Acesso a contexto e manutenção do estado do cliente, arquivos de configuração **Web Application Archives** – WAR, ciclo de vida de página **Java ServerPage** – JSP, conteúdo estático e dinâmico e a utilização do **Java Database Connectivity** – JDBC com JSP e **servlet**.



Sites dinâmicos e tecnologias de desenvolvimento

Há quem pense que sites dinâmicos são sites que exibem algum texto ou figuras em movimento na tela do browser. No entanto, essa é uma ideia muito "infantil" sobre o assunto.

Com o avanço da tecnologia que ocorre com uma velocidade cada vez maior, as aplicações corporativas passam a ter a necessidade de mudanças na arquitetura para que seja possível compartilhar as informações armazenadas em diferentes locais, diferentes meios de armazenamento e diferentes dispositivos de acesso a essas informações (SANTOS, 2007, p.3).

Até não muito tempo, as aplicações corporativas eram, em sua maioria, desenvolvidas em ambientes internos das empresas, sendo executadas em redes locais, com a necessidade de um servidor para aplicação, para o banco de dados e equipamentos espalhados pelos departamentos para os usuários acessarem o servidor, suprindo assim as necessidades gerenciais e administrativas das empresas.

Vejamos algumas das tecnologias da plataforma Java, desenvolvidas para suprir essa necessidade, como também as necessidades de implementação de soluções para internet (que é o nosso principal foco), para dispositivos móveis, entre outros.

Tecnologias:

- **JSE** (*Java Standard Edition*): Edição padrão de desenvolvimento de aplicações desktop e servidores (ORACLE, 2012a).
- **JEE** (*Java Enterprise Edition*): Edição que agrega o poder do desenvolvimento de aplicações para internet na plataforma Java. Essa plataforma utiliza o núcleo central da *Application Programming Interface* (API) da plataforma JSE (ORACLE, 2012b).
- **JME** (*Java Micro Edition*): Fornece um ambiente robusto e flexível para aplicações rodando dispositivos embarcados, tais como telefones celulares, *Personal Digital Assistant* (PDAs), TV, entre outros (ORACLE, 2012c).
- **JAVA TV**: É uma tecnologia de desenvolvimento de soluções de aplicativos Java executados em dispositivos de TV (ORACLE, 2012d).
- **JavaFX**: É uma evolução do Java como uma plataforma *richclient*. Projetado para fornecer uma interface rica e leve, aproveitando os recursos das plataformas empresariais existentes ou plataformas intermediárias (ORACLE, 2012e).



Dentre as tecnologias acima apresentadas, nosso foco será a tecnologia JEE e nosso objetivo é demonstrar, de forma conceitual e prática, como essa tecnologia pode nos ajudar na construção de sites dinâmicos com o auxílio dessa poderosa linguagem de programação, que é o Java.

A web 2.0

Com o surgimento da web na década de 1990, aos poucos foi sendo mudado o paradigma sobre o desenvolvimento de sistemas, que antes eram realizados localmente. Aos poucos as empresas foram vendo a internet como um meio de divulgação de seus produtos e serviços e, mais tarde, a comercialização deles por meio de portais de e-commerce. Porém, se antes os sistemas eram desenvolvidos em redes locais, por que não utilizar a rede mundial para melhorar a troca de informações, não só interna, mas o relacionamento entre clientes, fornecedores, parceiros, órgãos públicos etc.?

Inicialmente, a web (1.0), ainda restrita a poucos e com pouca tecnologia disponível, não gerava muitos atrativos. No entanto, com o avanço tecnológico constante, infinitas possibilidades apareceram e continuam aparecendo, mudando conceitos e estabelecendo novos. A web 2.0 é apenas um termo criado em 2004 para colocar um marco divisor entre uma web inicial simples, para uma web com mais serviços e infinitas possibilidades.

Os sistemas da Web 2.0 usam a web como uma plataforma, como uma enorme gama de dispositivos interconectados que podem fornecer um novo nível de experiência rica e imersiva para o usuário, um modelo de programação leve e fácil de usar para o desenvolvedor e um mecanismo de implantação rápido e flexível para o fornecedor. A Web 2.0 recia a Internet pelas perspectivas dos usuários, desenvolvedor e fornecedor; cada uma delas permitem novos e criativos usos da Internet.[...]As empresas com bancos de dados de grande porte de funcionários, parceiros e clientes há muito conhecem o valor do conhecimento que reside nas mentes dos funcionários, nos bancos de dados e nos documentos não estruturados encontrados em toda a organização. No passado, tentativas de coletar essas informações nos sistemas de gestão de conhecimento foram feitas, com vários níveis de sucesso, mas as tecnologias da Web 2.0, como blogs, wikis e o empenho da empresa em descobrir pessoas e dados poderão facilitar a gestão de conhecimento e fornecer uma nova plataforma de colaboração para tarefas complexas e criativas.

((PLATT, 2007))

Linguagens para web

Existem os mais diferentes tipos de linguagens, para as mais variadas necessidades encontradas no mercado, considerando-se também que algumas linguagens foram criadas com o surgimento do computador, e outras foram evoluindo conforme a tecnologia, tanto de software como de hardware.

Vamos comentar rapidamente algumas das mais famosas linguagens de desenvolvimento para web e suas principais características:

- **Ruby:** Linguagem interpretada, fortemente ¹tipada, totalmente orientada a objetos, quase não se utiliza dos recursos de colchetes e parênteses em sua sintaxe, é multiplataforma, não proprietária e executa no servidor. Veja mais: www.ruby-lang.org/pt (<http://www.ruby-lang.org/pt>)
- **ColdFusion:** Linguagem interpretada, trabalha com base em TAGs, fácil de aprender, é multiplataforma, suporta programação orientada a objetos, não proprietária e executa no servidor.
- **Pearl:** Linguagem interpretada, semelhante à linguagem C, é multiplataforma, com facilidades para manipulação de Strings e textos, suporta programação orientada a objetos, não proprietária e executa no servidor. Veja mais: www.perl.org (<http://www.perl.org>)
- **Microsoft.NET:** Linguagem interpretada e compilada, Necessita da plataforma .NET framework para ser executada, é proprietária e executa no cliente e no servidor e permite desenvolver aplicações em diversas linguagens de programação, tais como C#, Visual Basic, entre outras. Veja mais: [www.msdn.microsoft.com/pt-br](http://msdn.microsoft.com/pt-br) (<http://msdn.microsoft.com/pt-br>)
- **ASP:** Linguagem *procedural* interpretada, plataforma Windows, tipagem fraca, utiliza normalmente o servidor IIS ou outros servidores Windows, utiliza algumas linguagens de script, tais como VBScript e JScript, é



proprietária e executa no servidor. A evolução do ASP é o ASP.NET, incorporado no Microsoft .NET.

- **PHP:** Linguagem procedural interpretada, multiplataforma, tipagem fraca, suporta orientação a objetos, sintaxe semelhante à linguagem C/C++, não é proprietária e executa no servidor. Veja mais: www.php.net.br (<http://www.php.net.br>)
- **JavaScript:** Linguagem interpretada, multiplataforma, geralmente procedural, mas suporta orientação a objetos, sintaxe semelhante à linguagem C/C++, não é proprietária e executa no cliente. Veja mais: developer.mozilla.org/en-US/docs/JavaScript (<https://developer.mozilla.org/en-US/docs/JavaScript>)
- **JSP (JavaServerPages):** Linguagem de marcação de texto HTML com *Java* embutido, multiplataforma, tipagem forte, suporta orientação a objetos, necessita de um aplicativo servidor web ou de conteúdo como o GlassFish ou Tomcat, não é proprietária e executa a parte Java no servidor e envia HTML para o cliente. Veja mais: www.oracle.com/technetwork/java/javasee/jsp/index.html (<http://www.oracle.com/technetwork/java/javasee/jsp/index.html>)
- **Java Servlet:** Linguagem compilada (Java "puro"), multiplataforma, orientada a objetos, tipagem forte, necessita de um aplicativo servidor *web* ou de conteúdo como o GlassFish ou Tomcat, não é proprietária e executa no servidor. Veja mais: java.sun.com/j2ee/tutorial/1_3-fcs/doc/Servlets.html (http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Servlets.html)

Linguagens interpretadas e compiladas



Como vimos nos exemplos de linguagens acima, algumas são compiladas e outras são simplesmente interpretadas. Há também as linguagens que são executadas no lado cliente ou no lado servidor. No tópico seguinte, veremos a questão da execução acontecer no lado cliente ou servidor. Agora, passaremos a discutir a questão da linguagem interpretada ou compilada.

JavaScript, por exemplo, é uma linguagem de programação executada no lado cliente e interpretada pelo browser que estiver instalado e configurado no equipamento local. Portanto, é uma linguagem interpretada e executa no lado cliente.

Linguagens como o ASP e o PHP também são linguagens interpretadas. Porém, são interpretadas no lado servidor, por um serviço web instalado no servidor, como o *Internet Information Service* – IIS da *Microsoft* (2012) ou Apache (2012).

No entanto, há as linguagens que são compiladas, por exemplo: linguagens como C, C++, C#, Java etc., que executam inclusive em equipamentos desktop. Porém, linguagens como as da plataforma Dot.Net ou Java também podem ser compiladas e executadas no lado servidor.

Coloquemos, então, nossa atenção à linguagem *Java*, com as tecnologias *JSP* e *servlet*, que terão nosso principal foco no desenvolvimento de aplicações dinâmicas para *web*, acessando a banco de dados e fazendo a interação necessária entre as aplicações e plataformas, colocando nas mãos do desenvolvedor uma poderosa ferramenta para um desenvolvimento robusto e profissional na *web*.

Execução no lado do cliente e no lado do servidor

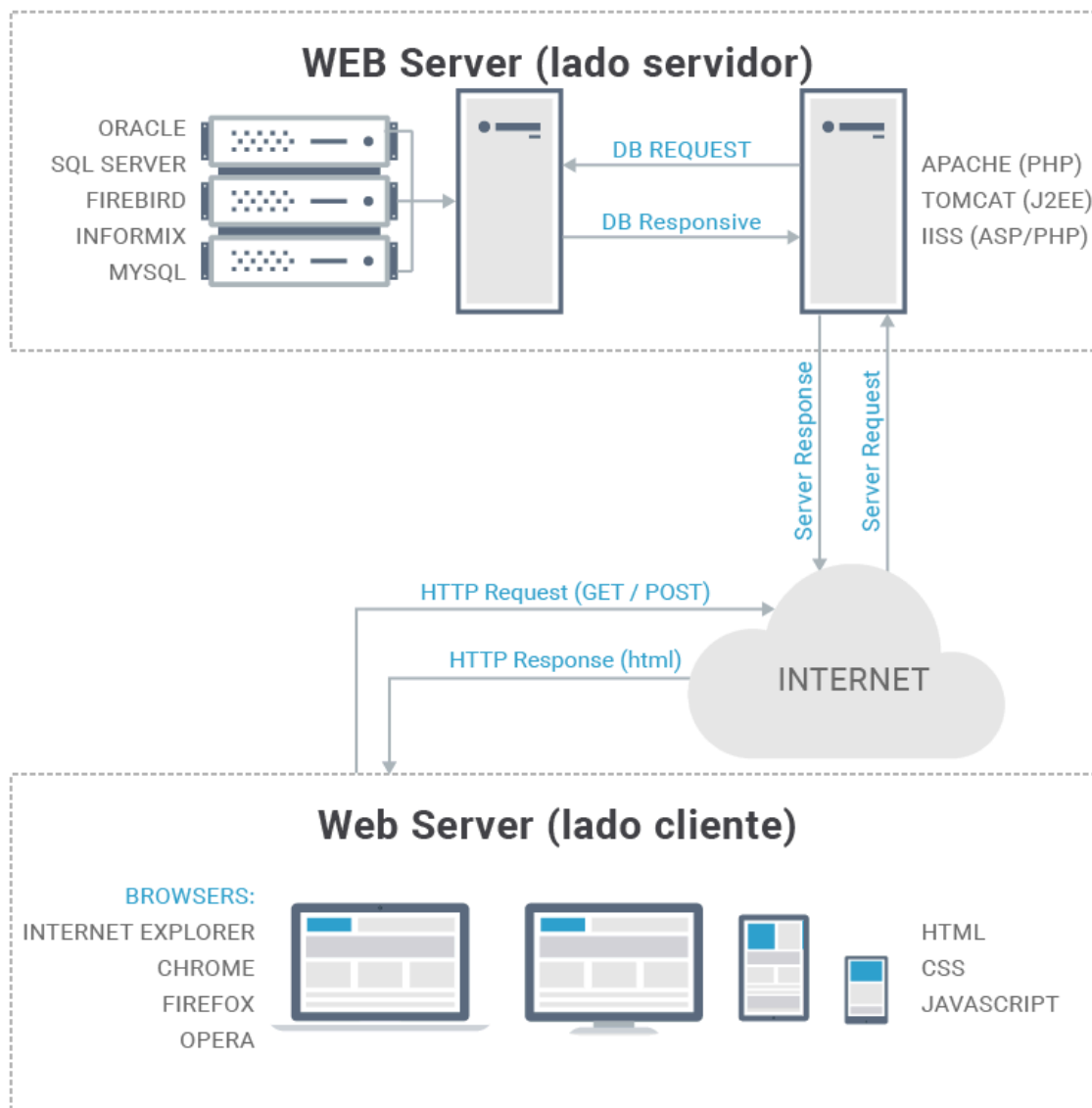
Tecnologias como o *HyperText Markup Language* – HTML, *Cascading Style Sheets* – CSS e JavaScript, embora sejam hospedadas no servidor, são enviadas ao cliente a cada solicitação e são interpretadas pelo *browser* no lado cliente.

No entanto, se desejamos acessar um banco de dados para obter e armazenar informações, precisamos de tecnologias que executam no lado servidor. Para o desenvolvimento no lado servidor, utilizaremos as tecnologias já mencionadas, como: JSP, servlet e banco de dados.

Parece pouco, diante de um emaranhado de siglas de tecnologias que temos para o desenvolvimento de aplicações dinâmicas na web. É bom enfatizar, porém, que para trabalharmos com essas tecnologias, diversos conceitos utilizados no mercado no dia a dia serão discutidos ao longo da disciplina. Alguns desses conceitos terão validade também para outras linguagens ou tecnologias.

Para melhor exemplificar o conceito de aplicação Client/Server, ou melhor, entender a execução lado cliente ou lado servidor, analisemos a figura a seguir:





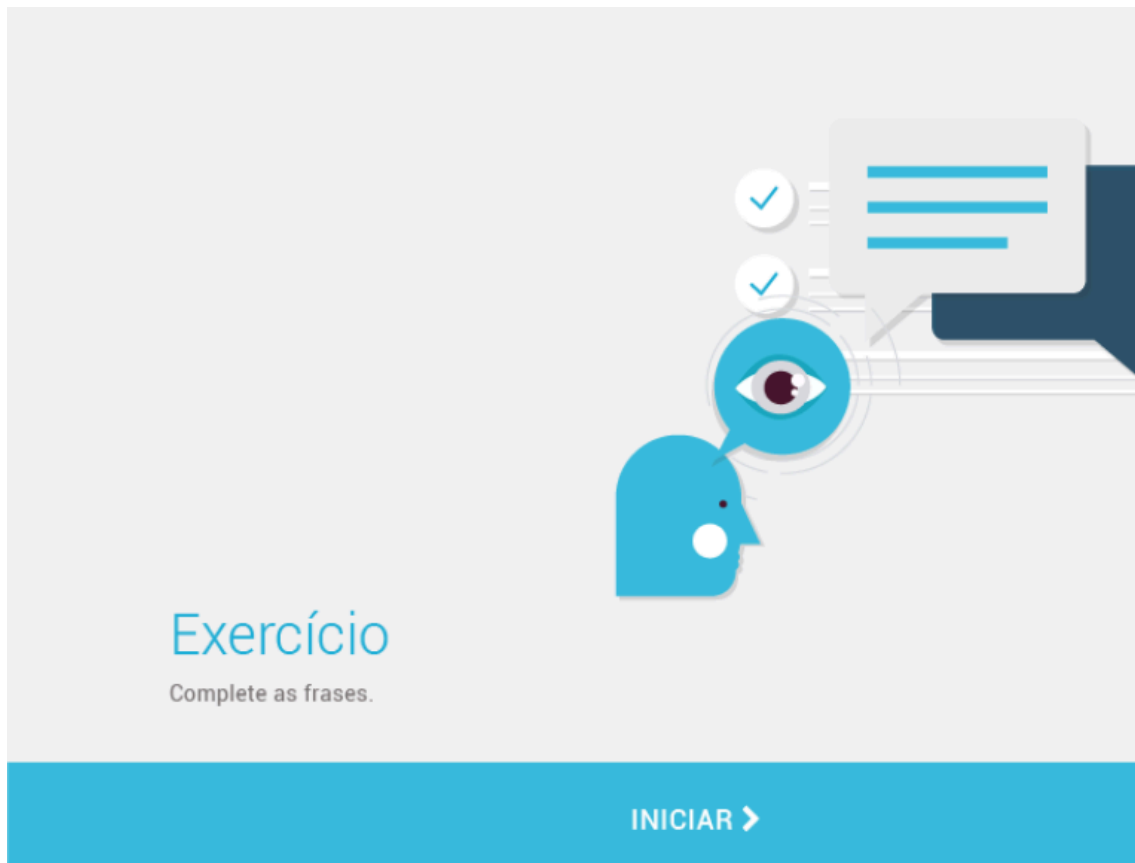
Na próxima aula, vamos entender um pouco mais sobre os servidores de conteúdo na internet, softwares e toda a infraestrutura necessária para a construção e execução de sites dinâmicos.

Como desenvolvedor apaixonado pelo poder da criação de aplicativos ou sites que possam enchê-lo de orgulho (e por que não dizer um retorno financeiro satisfatório?), precisamos primeiro entender o emaranhado de tecnologias que nos cercam, para depois colocar a "mão na massa" e começar a construir códigos.

SAIBA MAIS

¹**Tipada:** Uma linguagem que oferece uma grande variedade de tipos de dados.

A seguir, preencha a(s) lacuna(s) com a(s) palavra(s) adequada(s) às afirmações.



Referências

APACHE. *The Apache Software Foundation*. Disponível em: www.apache.org/ (<http://www.apache.org/>). Acesso em: 1º set. 2012.

MICROSOFT. *TechNet: Servidor Web (IIS)*. Disponível em: [technet.microsoft.com](http://technet.microsoft.com/pt-br/library/cc753433(v=ws.10)) ([http://technet.microsoft.com/pt-br/library/cc753433\(v=ws.10\)](http://technet.microsoft.com/pt-br/library/cc753433(v=ws.10))). Acesso em: 02 set. 2012.

ORACLE. *Java SE at a Glance*. Disponível em: [www.oracle.com](http://www.oracle.com/technetwork/java/javase/overview/index.html) (<http://www.oracle.com/technetwork/java/javase/overview/index.html>). Acesso em: 1º set. 2012.

_____. *Java EE at a Glance*. Disponível em: [www.oracle.com](http://www.oracle.com/technetwork/java/javaee/overview/index.html) (<http://www.oracle.com/technetwork/java/javaee/overview/index.html>). Acesso em: 1º set. 2012.

_____. *Java ME and Java Card Technology*. Disponível em: [www.oracle.com](http://www.oracle.com/technetwork/java/javame/index.html) (<http://www.oracle.com/technetwork/java/javame/index.html>). Acesso em: 1º set. 2012.

_____. *Java TV*. Disponível em: www.oracle.com (<https://www.oracle.com>). Acesso em: 1º set. 2012.

_____. *JavaFX*. Disponível em: [www.oracle.com](http://www.oracle.com/technetwork/java/javafx/overview/index.html) (<http://www.oracle.com/technetwork/java/javafx/overview/index.html>). Acesso em: 1º set. 2012.

PLATT, Michael. Biblioteca MSDN. *Web 2.0 na empresa* (29 de outubro de 2007). Disponível em: [msdn.microsoft.com](http://msdn.microsoft.com/pt-br/library/bb735306) (<http://msdn.microsoft.com/pt-br/library/bb735306>). Acesso em: 02 set. 2012.

SANTOS, Rui Rossi dos. *Java na web: programando sites dinâmicos*. Rio de Janeiro: Axcel Books, 2007.



Avalie este tópico



Biblioteca

([https://www.uninove.br/conheca-](https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Ajuda?
PRODUTO
(<https://ava.uninove.br/cursos/>)

Servidores de conteúdo e navegação

© Todos os direitos reservados

