

## Unidade 1: Introdução à Programação de Servidores com Java

Prof. Daniel Caetano

**Objetivo:** Apresentar os conceitos fundamentais da programação de servidores e da linguagem Java, bem como sua história e nomenclatura.

**Bibliografia:** DEITEL, 2005; CAELUM, 2010; HOFF, 1996.

### INTRODUÇÃO

O objetivo desta aula é proporcionar uma uniformização do conhecimento básico sobre o que é a programação para servidores, o que é a linguagem Java, para apoio de todas as aulas subsequentes. Esta aula está dividida nas seguintes seções:

- 1) O que é Programação para Servidores?
- 2) Histórico da Linguagem Java
- 3) Ambientes Java
- 4) A Linguagem Java e as Máquinas Virtuais

### 1. PROGRAMAÇÃO PARA SERVIDORES

A maioria dos alunos da área de TI já está familiarizada com o conceito de software: praticamente qualquer pessoa do mundo moderno já tomou contato com um programa de computador, em especial aquelas que escolhem um curso voltado ao desenvolvimento deste tipo de produto.

A maioria dos programas aos quais as pessoas estão acostumadas - como o Word, o Excel, os jogos e tantos outros - são aqueles programas que são executados no próprio computador do usuário. Entretanto, desde o surgimento da Internet, os usuários se acostumaram com um outro tipo de recurso: aqueles disponíveis através da rede. Alguns usuários se perguntam: onde fica a internet? Onde fica uma página de internet? Onde fica o Google? Onde fica o Facebook?

Antes de uma análise mais apurada sobre o local no qual estes sistemas estão disponíveis, convém responder primeiramente **o que** são esses sistemas. O que é o Google? O que é o Facebook? O que é o gMail?

A resposta é muito simples: eles são **serviços**.

É notório que estes serviços não estão na máquina do usuário: quando a rede cai, o usuário perde acesso a eles. Esses serviços ficam em algum outro lugar, que não sabemos

exatamente qual é. A resposta mais comum para a localização destes serviços nos tempos atuais é... "**na nuvem**". Mas você sabe o que é "a nuvem"?

A "nuvem" é um termo que foi criado para designar a rede formada pela internet, incluindo todos os computadores do mundo - e isso vale para o seu também. Quando se diz que um documento, um site ou um serviço está "na nuvem", na verdade significa que ele está em um computador idêntico ao seu, embora muitas vezes façamos a menor idéia do local em que esse computador se encontra!

Os computadores que executam serviços como o Google, o Facebook e o gMail, exatamente por fornecerem serviços, são chamados de **servidores**. Há diversos tipos de serviços que um servidor pode fornecer, como e-mail, msn, páginas web, aplicações...

Os computadores que, por outro lado, usam esses serviços, são chamados de computadores **clientes**. Quando você navega na internet, o seu computador está atuando como **cliente** de diversos **servidores web**.

Para que você veja uma página qualquer, o seu computador (cliente) deve enviar uma **requisição** para o servidor, que enviará a resposta solicitada. Nosso computador, na realidade, não sabe como encontrar o servidor. Ele sabe apenas o **nome** do servidor (frequentemente indicado como uma URL ou um número IP).

Assim, para solicitar alguma coisa ao servidor, o computador cliente constrói um documento de requisição chamado **REQUEST**, que inclui, no mínimo, o **nome do servidor**, **nome da origem** e o **texto da requisição**. Este arquivo é enviado pela rede (nuvem), que se responsabiliza por entregar a requisição ao servidor cujo nome foi informado.

Ao chegar neste servidor, a requisição é processada e, então, o servidor cria um documento de resposta, chamado **RESPONSE**, que irá conter todo o texto da resposta daquela requisição. A resposta também contém um endereço de destino (nome do cliente) e o nome da origem (o servidor). Assim que a resposta está pronta, o servidor a envia através da rede que, mais uma vez, se responsabiliza pelo encaminhamento da mesma de volta até o computador do cliente.

No caso de uma página web, essa resposta é, usualmente, um arquivo HTML, que será usado pelo navegador para desenhar a página web para o usuário. Genericamente, todo aplicativo que aguarda requisições e envia respostas às requisições recebidas é chamado de **aplicativo servidor**. Um equipamento que executa um aplicativo servidor é, assim, um **equipamento servidor**.

Neste curso iremos desenvolver aplicativos servidores. Como a linguagem C/C++ não é exatamente prática para este tipo de desenvolvimento, iremos utilizar, neste curso, a Linguagem Java. A Linguagem Java tem algumas facilidades e vantagens, como uma grande infraestrutura pronta para o desenvolvimento de aplicativos servidores, incluindo uma vasta biblioteca com esta finalidade, além de uma enorme semelhança com a linguagem C/C++, o que facilita o aprendizado da mesma por programadores que já conhecem a linguagem C.

## **2. A LINGUAGEM JAVA**

Existem diversas linguagens de programação disponíveis no mercado. Praticamente todas elas possuem características semelhantes, como possibilitarem a descrição precisa dos passos que uma tarefa precisa para ser concluída e disponibilizarem uma biblioteca com tarefas complexas pré-programadas.

Os símbolos e regras são bastante similares na maioria destas linguagens, estando a maior diferença entre elas justamente nas bibliotecas. Dependendo do tipo de uso que foi imaginado para uma linguagem - isto é, se ela é para a web, para banco de dados, para cálculos matemáticos etc., sua biblioteca englobará tarefas diferentes.

Assim, antes de nos aprofundarmos no estudo das linguagens de programação com o uso da linguagem Java, vamos conhecer um pouco do histórico da linguagem Java.

### **2.1. Histórico do Java**

A linguagem Java foi concebida como uma linguagem para desenvolvimento de produtos eletrônicos de consumo (eletrodomésticos e eletro-eletrônicos), com software embarcado. Entretanto, ela acabou se popularizando apenas com o advento da World Wide Web e apenas recentemente vem voltando à sua vocação inicial.

#### **Origens**

No início da década de 1990 estavam se popularizando os equipamentos eletro-eletrônicos programáveis/programados, indo desde televisores até fornos de microondas e geladeiras. Embora muitas empresas tivessem notado que as linguagens existentes traziam problemas para o desenvolvimento destes equipamentos, foi a Sun Microsystems quem primeiro propôs uma solução.

Antes de entendermos qualquer tipo de solução, é importante entendermos qual era o problema, que talvez não seja óbvio para aqueles que nunca trabalharam com projeto de equipamentos eletro-eletrônicos.

Sempre que um projeto é realizado, uma decisão importante que deve ser feita é a definição de quais serão os componentes do equipamento que está sendo projetado. No caso de um equipamento eletrônico, componentes importantes são os eletrônicos, em especial os circuitos integrados e, no caso dos eletro-eletrônicos programáveis (ou programados), os microprocessadores.

Via de regra, o processador selecionado é aquele que tiver o menor custo, dado que atende às características básicas do projeto. Entretanto, um eletro-eletrônico pode continuar sendo produzido e vendido por vários anos; por outro lado, o preço dos processadores não é estático ao longo destes mesmos anos, fazendo com que o "processador mais barato que

atenda às necessidades" possa mudar com o tempo. Nestas situações, em geral os equipamentos voltam para a prancheta e são redesenhados para acomodar um novo processador, por exemplo.

É importante ressaltar que uma economia de alguns reais em cada unidade pode levar a grandes lucros para a empresa, visto que dezenas de milhares de unidades daquele eletro-eletrônico são produzidas ao longo de um ano: um aumento de lucro que as empresas em geral não desprezam. Exemplos, em casos de video-games (SMS1/2/3/Compact, MD1/2/3, PS/PSOne, PS2/PS2Slim, PS3/PS3Slim, XBox/XBox360, GameCube/NintendoWii...)

Entretanto, a troca de um processador muitas vezes implica em troca de todo o software, já que usualmente processadores distintos têm linguagens de máquina distintas. O problema então surge: a necessidade de se re-compilar e, muitas vezes, reescrever um software para o novo processador... acaba com grande parte do lucro obtido com a troca do processador. E, mesmo quando isso não ocorria, muitas vezes significava novos "bugs" e problemas, algo bastante indesejável. De olho nisso, em 1990, James Gosling começou a trabalhar em uma linguagem que funcionasse de tal forma que os programas raramente precisassem ser reescritos quando a plataforma onde são executados fosse substituída, desde que ambas oferecessem recursos similares. Essa linguagem acabou por ficar conhecida como Linguagem Java.

### Projetos Iniciais

Raramente uma linguagem baseada apenas em teoria e sem experimentação prática consegue ter sucesso. Por esta razão, os técnicos da Sun Microsystems, durante o desenvolvimento do Java desenvolveram projetos em Java, para testar suas funcionalidades. O primeiro destes projetos foi o Projeto Green, que visava a criação de uma nova interface com o usuário para o equipamento "\*7" (Star Seven), que tinha o objetivo de controlar os eletrodomésticos de uma casa através de ícones animados e uma touch screen. Um outro projeto foi o de VoD (Video On Demand), com uma função similar ao que hoje se chama de TV Interativa.

Entretanto, foi com o surgimento da Web que a nova linguagem realmente apareceu a público: os navegadores web estavam em franca evolução quando a Sun apresentou o WebRunner, mais tarde renomeado para HotJava. A principal característica destes browsers não era exatamente a renderização HTML (o que eles faziam de forma similar aos já existentes Mosaic e Netscape), mas sim o fato de terem capacidade de executar applets java, pequenos programas que rodavam no computador do usuário, fosse esse computador IBM PC ou Apple MacIntosh.

A inovação fez tanto sucesso que em poucas semanas a Netscape lançava sua primeira versão capaz de executar a Java Virtual Machine da Sun como plugin e, com isso, executar também applets java. Mais tarde foi incorporado no browser da Netscape também o JavaScript e, rapidamente, ambos se tornaram padrões tão importantes que é quase impossível navegar hoje sem os mesmos instalados, juntamente com o Macromedia Flash.

## O Java Hoje

O tempo foi passando e mostrou que a Sun Microsystems, de alguma forma, estava adiante de seu tempo. Com o surgimento dos PDAs (Personal Data Assistants, os "PALMs") e telefones celulares capazes de executar aplicativos, tornou-se bastante atrativa uma tecnologia que permitisse que um programa pudesse ser executado em máquinas diferentes: afinal de contas, não só os recursos disponíveis nestes equipamentos, como também seus processadores e arquiteturas podem ser bastante diferentes até mesmo de um modelo para outro!

Assim, hoje o Java voltou a ter sua vocação inicial: desenvolvimento de software embarcado em eletro-eletrônicos. Ainda não é muito comum, mas vem crescendo o número de equipamentos como Set-Top-Boxes (HDTV), modems ADSL, computadores portáteis, DVD players, TVs e outros equipamentos que se utilizam de programas escritos na linguagem Java para permitir que o usuário se comunique com o equipamento.

## 3. AMBIENTES JAVA

Como dito anteriormente, como as funcionalidades exigidas por uma aplicação depende de seu tipo e finalidade, a linguagem Java foi dividida em três grandes pacotes, que englobam as principais áreas de utilização da linguagem Java: Java SE, Java ME e Java EE.

**Java SE: Java Standard Edition** - O Java SE é, por assim dizer, um pacote básico do Java, voltado à construção de aplicações tradicionais, isto é, que são executadas em um computador com boa capacidade de processamento e memória, e executam integralmente (ou quase) na máquina do usuário.

**Java ME: Java Micro Edition** - O Java ME é uma versão bastante reduzida do Java, com bibliotecas relativamente simplificadas - não existem tipos float e double, por exemplo -, voltada para a construção de aplicações pequenas, isto é, executadas usualmente em dispositivos móveis, como celulares e palmtops, com pouca capacidade de processamento e memória, sendo normalmente executadas integralmente no equipamento do usuário.

**Java EE: Java Enterprise Edition** - O Java EE é uma versão voltada ao desenvolvimento de aplicações que são executadas em ambiente servidor, incluindo todos os recursos necessários para seu uso em ambiente de rede e, em especial, a Web, incluindo recursos de persistência de dados, gerenciamento de transações e uma série de outros recursos que facilitam o desenvolvimento deste tipo de aplicação.

### 3.1. Versões do Java

A nomenclatura do Java traz alguma confusão para os iniciantes. A função desta seção é elucidar algumas destas questões.

Primeiramente, **Java Runtime Environment (JRE)** é um pacote que inclui tudo que se precisa para rodar um programa Java tradicional. Este pacote inclui a **Java Virtual Machine (JVM)** e todo o conjunto de bibliotecas e pacotes da linguagem Java.

O outro pacote disponível, chamado **Java Development Kit (JDK)**, é um pacote mais completo, que inclui o suporte básico ao desenvolvimento Java. Este pacote **inclui** tudo que o JRE inclui, **adicionando** os componentes necessários para gerar programas Java.

Ambos os pacotes existem em sabores SE, EE e ME, referindo-se aos pacotes com componentes Java SE, Java EE e Java ME, respectivamente.

Com relação ao **número** de versão, é preciso entender que, até hoje, o Java não saiu da versão 1.x, isto é, a primeira versão. Por questões comerciais, a Sun/Oracle adotaram nomes que sugerem versões mais avançadas, mas isso só traz confusão aos desenvolvedores. Abaixo segue uma lista com as principais versões de Java:

<b>Versão Real</b>	<b>Nome</b>	<b>Descrição</b>
1.0 a 1.1	Java	Versões iniciais do Java
1.2 a 1.4	Java 2	Adição de um conjunto enorme de componentes básicos
1.5	Java 5	Mais pacotes básicos acrescentados
1.6	Java 6	Otimização para melhoria de desempenho do Java 5

#### **4. COMO FUNCIONA O JAVA**

Já foi discutida a capacidade de um programa Java poder ser executado em qualquer lugar, mas como isso ocorre? Como um código feito para um "computador que não existe" consegue rodar em qualquer lugar?

Na verdade, o funcionamento é muito similar ao dos populares emuladores de videogames, que permitem a execução de jogos de PlayStation, GameCube, DreamCast e outros no seu PC. É como se Java fosse a linguagem de um computador antigo e existisse um "emulador" para executar os programas desse computador no PC. Esse "emulador" chama-se **Interpretador Java** ou **Java Virtual Machine (JVM)** e, uma vez reescrito para um novo equipamento, todos os programas Java passam a executar neste equipamento.

A JVM exerce o papel de um "tradutor simultâneo". É ela quem lê o programa Java e diz para um computador específico o que deve ser feito para realizar aquela tarefa. Ela funciona como um intermediário. É como um intérprete de um técnico de futebol que não fala a língua dos jogadores:

<i>Nome do Técnico</i>	<i>Língua do Técnico</i>	<i>Conversão</i>	<i>Língua dos Jogadores</i>
Luis Felipe	Português	Intérprete P/A	Árabe
Luis Felipe	Português	Intérprete P/I	Inglês
Luis Felipe	Português	Intérprete P/J	Japonês

<i>Nome do Programa</i>	<i>Linguagem do Programa</i>	<i>Conversão</i>	<i>Linguagem do Processador</i>
MeuPrograma	Java	JVM J/P4	Pentium IV ASM
MeuPrograma	Java	JVM J/PPC	PowerPC ASM
MeuPrograma	Java	JVM J/A7	ARM7 ASM

Perceba que ao trocar a língua do time, não é preciso trocar o técnico nem a língua que ele fala, pois existe um intérprete que faz as traduções. Se trocar o time e mantiver o técnico, basta trocar o intérprete. No caso do programa em Java, ocorre o mesmo: não é preciso trocar o programa nem a linguagem dele quando se troca de processador: basta trocar a JVM.

Como existe um passo a mais de tradução, isso tem influência direta no desempenho das aplicações Java. Apesar de aplicações Java possuírem um desempenho bastante superior ao de linguagens script normais, seu desempenho pode ser bastante mais lento que uma linguagem compilada como C. Entretanto, os fabricantes não têm se mostrado muito preocupados com esse "problema", dado que os equipamentos têm poder de processamento cada vez maior a custos cada vez menores: preservar o investimento em software desenvolvido acaba sendo muito mais importante quando se visa lucro em alguns mercados (como o dos celulares).

Nas versões mais recentes, a Sun se empenhou em resolver o problema "desempenho", sempre associado à linguagem Java. Para isso criaram um sistema chamado de "hotspots", com o uso da tecnologia JIT (Just-in-Time), que compilam o código à medida em que ele é executado, com grande otimização, permitindo que, em muitos casos, programas em Java de versão recente sejam executados em velocidade similar a programas em C ou C++.

### **13. BIBLIOGRAFIA**

CAELUM, FJ-11: Java e Orientação a Objetos. Acessado em: 10/01/2010. Disponível em: <  
<http://www.caelum.com.br/curso/fj-11-java-orientacao-objetos/> >

DEITEL, H.M; DEITEL, P.J. **Java: como programar** - Sexta edição. São Paulo: Pearson-Prentice Hall, 2005.

HOFF, A; SHAIQ, S; STARBUCK, O. **Ligado em Java**. São Paulo: Makron Books, 1996.