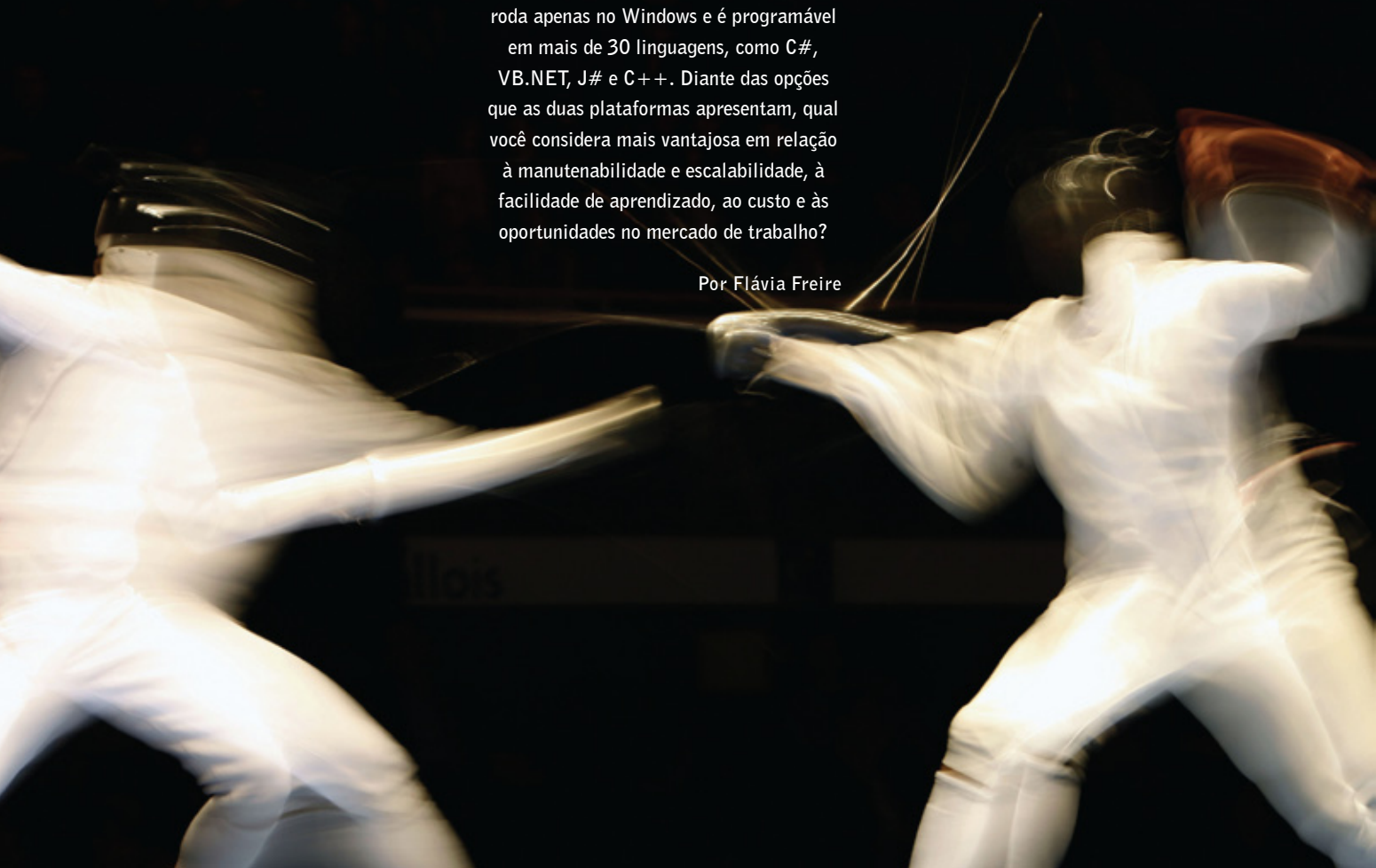


JAVA X .NET

Java versus .Net: uma discussão que permanece viva desde o surgimento do .Net Framework, da Microsoft, em 2002, quando a Sun já dominava o mercado com o Java. Duas fortes plataformas de desenvolvimento, capazes de oferecer soluções para dispositivos móveis, desktops, web, entre outros, mas que possuem uma grande diferença: Java é multiplataforma e programado em uma linguagem; .Net roda apenas no Windows e é programável em mais de 30 linguagens, como C#, VB.NET, J# e C++. Diante das opções que as duas plataformas apresentam, qual você considera mais vantajosa em relação à manutenibilidade e escalabilidade, à facilidade de aprendizado, ao custo e às oportunidades no mercado de trabalho?

Por Flávia Freire





Fabio Kung

Instrutor, desenvolvedor e consultor na Caelum. Certificado SCJP, SCBCD 5, SCEA 5 da Sun e Ruby Certified Programmer Silver. Já palestrou em eventos como Conexão Java, JustJava e Falando em Java. Trabalhou com Java para a web na Alemanha e participa de projetos open source como VRaptor, Waffle, G.U.J., Caelum Stella e JettyRails.

Como o nosso papel é sempre considerar vantagens e desvantagens a cada caso, a resposta certa para esse tipo de pergunta é “depende”. Defender algo como “a bala de prata” pode fazer com que deixemos de evoluir e de considerar novas possibilidades e abordagens melhores. Ao invés de apontar um vencedor, prefiro discutir vantagens e desvantagens para alguns dos pontos levantados. O melhor sempre dependerá de cada caso.

A plataforma não é decisiva para a escalabilidade, pois não é a tecnologia que diz se uma aplicação escala ou não. O importante é o conhecimento da equipe sobre a plataforma, para fazê-la escalar. O ponto positivo para o Java aqui é a imensidão de bibliotecas e frameworks existentes, que fornecem mais opções quando a escalabilidade não está boa. Além disso, podemos rodar aplicações Java nos mais diversos hardwares e sistemas operacionais, aproveitando o que têm de melhor para escalabilidade: bons escalonadores de threads e processos, gerenciamento de memória, rede e sistemas de arquivos; distribuídos ou não. Mesmo assim, .Net não fica tão atrás, já que o projeto Mono tem evoluído a passos largos e é uma ótima opção para aplicações .Net em plataformas não Windows.

A manutenibilidade também tem muito mais a ver com a forma com que a aplicação foi construída, se usa boas práticas de arquitetura e design orientado a objetos, principalmente se tem testes automatizados, e se o acoplamento entre componentes está baixo. Dessa forma, uma pequena vantagem ao mundo Java pela cultura, que desde o início busca boas práticas e procura discutir design orientado a objetos. O mérito aqui não é da plataforma, mas sim da comunidade. Claro que não é regra, mas **os desenvolvedores Java tradicionalmente procuram entender mais o que acontece “por baixo dos panos” e como as peças se encaixam na construção de uma aplicação, sem depender de ferramentas que escondem tudo isso e prometem produtividade. Existe até mesmo um movimento na comunidade .Net, conhecido como Alt.Net, que promove boas práticas e é contra a cultura de “wizards com arrasta-e-solta”, por muito tempo difundida como vantagem, mas que muitas vezes acaba alienando e banalizando o desenvolvedor.**

As duas plataformas têm especificações abertas, mas Java é gerida por um comitê - JCP - enquanto as decisões relativas a .Net são tomadas apenas pela Microsoft. O comitê traz segurança às corporações, que não ficam na mão de apenas uma empresa e evitam o “vendedor lock-in”. Em contrapartida, a burocracia do comitê inibe a inovação e faz com que as especificações sempre estejam alguns passos atrás do mercado.

Novidades chegam com rapidez ao mundo .Net, bastando a vontade da Microsoft. No Java, as inovações vêm da comunidade, geralmente na forma de projetos open source por fora do comitê; o que gera a imensidão de alternativas. Ter várias opções é por um lado vantajoso, pois alguma é melhor para o seu caso; porém, não há tempo para avaliar todas e a decisão acaba não levando em conta vantagens e desvantagens técnicas, pois ficamos perdidos no meio de tantas opções. No .Net, a Microsoft já toma boa parte das decisões, como quais bibliotecas e frameworks usar.

Programação poliglota (uso da melhor linguagem para cada tarefa) é uma tendência muito forte. A plataforma .Net saiu na frente: desde o começo foi feita para suportar várias linguagens. Mais ainda, com a chegada do projeto DLR (Dynamic Language Runtime), melhorando o suporte às linguagens dinâmicas como Ruby e Python. Mesmo um pouco atrasada, Java também segue a mesma linha. Hoje é possível rodar programas escritos em mais de 200 linguagens (procure por “JVM Languages”). A linguagem Java (que não evolui há mais de 5 anos) não é mais o foco da plataforma, que se volta para a importância que Java tem como plataforma de execução de código: uma excelente máquina virtual, compilador JIT adaptativo, algoritmos eficientes de coleta de lixo e bibliotecas consagradas como o Spring, Hibernate e outros.

Muitos acreditam em um futuro sem polarização de linguagens (Java vs. C#), mas com polarização de plataformas (Java vs. .Net), em que escolheremos a linguagem mais adequada a cada caso. A briga ficará por qual plataforma escolher para rodar este código. Veremos.





Denis Bittencourt Muniz

Desenvolvedor de software na Synchron Informática. Estudante em Análise e desenvolvimento de sistemas pelo IFSP - São Carlos. Experiente em .Net, se dedica também às boas práticas de modelagem de arquitetura de software (design patterns, SOA, etc.) e frameworks.

Conheço a plataforma .Net desde a versão 2.0 beta, utilizando C#, Visual Basic .Net e, mais como um hobby, C++/CLI. Na mesma época que conheci a plataforma .Net, estava também estudando a plataforma Java. **Estudando e comparando ambas, me identifiquei mais com a plataforma .Net. Fiquei impressionado com a riqueza de suas bibliotecas. De fácil aprendizado, alto grau de utilidade e flexibilidade, e bem documentada, aumentam a produtividade do desenvolvedor e a facilidade em criar-se softwares robustos e de fácil manutenção.** Um exemplo é a arquitetura elegante do ADO.NET: arquitetar a camada de dados (conhecida também como camada de persistência) do software com independência de banco de dados ficou facilitado. Um outro exemplo, seria o ASP.NET, um framework web (popularmente, conhecido como uma linguagem de programação) na qual o desenvolvedor utiliza juntamente com uma linguagem da plataforma .Net, que revolucionou o modo de desenvolver para a web, utilizando-se de componentes (controles de interface) para construir sites ou aplicações web com maior produtividade, além de fácil configuração, fácil manutenção e flexibilidade (desde a construção do aplicativo em si a criação de componentes de interface, Web Services, módulos HTTP, por exemplo). Outro fator importante é a grande evolução (arquitetura, bibliotecas, expansão diante ao mercado) da plataforma em pouco espaço de tempo, se comparado ao Java. Um outro aspecto importante, desconhecido para muitos, é que a plataforma .Net, assim como Java, segue uma especificação padronizada. A plataforma .Net (assim como o Mono e o DotGNU) descende da arquitetura CLI (Common Language Infrastructure), padronizada pelas organizações ECMA e ISO.

A plataforma .Net também oferece ótimas oportunidades no mercado de trabalho. Grande parte das propostas de emprego que me candidatei ou, principalmente, recebi, envolvia a plataforma .Net, tanto Visual Basic .Net, quanto C#. Sinto-me satisfeito trabalhando com a plataforma no dia a dia.

Hoje, para desenvolvimento de aplicações, temos muitas opções de linguagens e plataformas. Focando em plataforma, as principais são o J2EE e .Net. O que muitos não sabem é que as especificações da plataforma .Net (CLI) e C# são padrão ISO/ECMA (<http://migre.me/3nkm>), quem quiser portá-la para outra plataforma pode fazê-lo, e já fazem, como por exemplo: o projeto Mono (<http://www.mono-project.com>).

A curva de aprendizado em .Net é mais suave do que em J2EE, pois tudo que o desenvolvedor precisa está disponível no Framework .Net. Seja qual for a linguagem escolhida, o resultado final do desenvolvimento será o mesmo, pois todas as linguagens .Net devem gerar o mesmo código IL. Na J2EE, temos o Java como linguagem, que por si só não apresenta uma alta curva de aprendizado. O problema reside no fato de muitas coisas que precisamos fazer termos que usar frameworks, ou seja, não é nativo da plataforma, como por exemplo o JSF e o Hibernate, o que obriga o desenvolvedor a aprendê-los.

As aplicações .Net podem se apoiar em um ecossistema de servidores/serviços como o IIS, Sharepoint, Windows Azure etc., tornando simples as tarefas como manutenção, escalabilidade, segurança, entre outras. A plataforma J2EE também tem à sua disposição vários servidores/serviços como WAS, JBoss etc.

Quanto à IDE, para .Net temos o Visual Studio .Net, um ambiente onde se pode desenvolver aplicações para dispositivos móveis, web, desktop, serviços etc. Para desenvolver em Java, temos diversas IDEs como Eclipse, JCreator e NetBeans.

Ambas as plataformas atendem às necessidades de desenvolvimento de aplicações simples até aplicações corporativas que exigem alta performance, escalabilidade, interoperabilidade etc. Caso o desenvolvimento seja voltado para a plataforma Windows, a performance do .Net em relação ao Java é visível. Em termos gerais, a plataforma .Net apresenta melhor performance e produtividade, seja no ambiente Windows, desktop ou web.

O mercado de trabalho é bom para quem desenvolve para ambas as plataformas, com a adoção variando de acordo com a região.



Leonardo Bruno Lima

Microsoft Most Valuable Professional desde 2004, arquiteto de software na ATTPS Informática, certificado MCP, MCTS e MCPD. Amplo conhecimento em soluções distribuídas, metodologia de desenvolvimento ágil e tecnologias para desenvolvimento de sistemas na plataforma .Net.

Para a comparação entre tecnologias faz-se necessário a escolha dos fatores a serem comparados, pois cada plataforma apresenta suas vantagens e desvantagens. Assim, primeiramente deve-se determinar os objetivos que direcionarão a escolha da plataforma.

No quesito manutenibilidade, Java leva vantagem em função de sua forte estrutura de linguagem e frameworks disponíveis. O .Net permite a utilização de diversas linguagens, o que aumenta o risco da entropia do código-fonte do sistema.

No item escalabilidade, as implementações das duas plataformas proveem mecanismos de balanceamento de carga, permitindo habilitar um cluster de máquinas para servir às cargas dos usuários que forem aumentando ao longo do tempo.

No item facilidade de aprendizado, o Java, por ter mais tempo de mercado, possui muito mais material disponível, embora em relação à dificuldade, acredito que no .Net se consiga desenvolver uma aplicação mais rapidamente para quem começou sem saber nada da linguagem.

Apesar da Sun deter a marca Java, a mesma tornou a linguagem open source, permitindo aos usuários efetuar alterações convenientes. Para a plataforma .Net, existem algumas iniciativas free, como os projetos Mono e DotGNU. Porém, para aproveitar todo o potencial da plataforma, é praticamente necessário adquirir o servidor Microsoft juntamente com o ambiente de desenvolvimento Visual Studio. Sem estas ferramentas, além de não aproveitar tudo o que se tem, a velocidade de desenvolvimento fica fortemente prejudicada. Assim, o TCO (Total Cost Ownership) do .Net costuma ser maior do que o do Java.

No último quesito, oportunidades no mercado de trabalho, minha leitura diz que há, levemente, mais vagas para programadores Java, embora os salários sejam equivalentes.



Paulo Krieser

Diretor Executivo da Krieser IT Solutions (www.krieser.com.br), empresa especializada em desenvolvimento de aplicações corporativas e e-commerce. Possui ampla experiência em gerência de projetos e na utilização de diversas tecnologias.

Ainda que muito intrigante e quase sempre calorosa, essa é uma discussão que cada vez menos é tida como uma guerra entre Xiitas e Sunitas. É muito importante ressaltar que o Java em si não é somente uma linguagem de programação, mas também uma plataforma, portanto, essas discussões necessitam estarem de acordo com o contexto a ser discutido, caso contrário, ela não terá sentido, pois não podemos comparar uma plataforma com uma linguagem e vice-versa.

Ao considerarmos uma discussão entre a linguagem Java e .NET, estamos cometendo um grande equívoco, pois .NET, assim como o Java, é uma plataforma. O correto seria uma comparação entre a linguagem Java e C# ou Java e Visual Basic para .NET.

Outro item de grande valia é a versão a ser comparada. Precisamos estar cientes de que existem grandes diferenças entre versões de plataformas e linguagens. Ao longo destes anos que venho atuando em diversas empresas de segmentos variados, **venho acompanhando o crescimento e amadurecimento das duas plataformas: o .NET, em muitos casos, sendo considerado imbatível em aplicações desktop; já o Java, o grande alicerce de sistemas que necessitem de grande disponibilidade.**

As tecnologias se equivalem em grandes e decisivos quesitos, vai muito a gosto do cliente optar por qual será a escolhida para sua empresa. A grande dificuldade da Microsoft é fazer com que a maioria dos desenvolvedores da plataforma .NET venha a trabalhar e focar nos padrões de projeto com a tecnologia, ponto este que é, de longe, o grande diferencial dos desenvolvedores Java, que para tudo e qualquer sistema, focam em bons padrões.

Através do meu vasto relacionamento com empresas de RH devido ao meu site (www.empregonaweb.com), venho acompanhando uma tendência cada vez maior de procura por profissionais .NET, eles estão cada vez mais escassos e o mercado vem aumentando seu valor hora na medida que a procura fica cada vez mais difícil. O profissionais Java continuam, e continuarão, sendo os grandes diamantes do mercado, muitos sistemas de grande valia para empresas foram desenvolvidos nesta plataforma, e novos virão.

É muito importante citar que as empresas que contratam meus serviços são, na maioria, grandes defensoras do Open Source, portanto, a minha visão tende a ser pró Java. ■



Dalton Camargo

Criador do JavaFree.org, Infoblogs.com.br e EmpregoNaWeb.com, é também arquiteto de soluções e idealizador do JavaBB.org, atualmente adotado pelo Universo Online para ser a base de fóruns de discussões da companhia. daltondecamargo.wordpress.com