

Unidade Jundiaí

Ciência da Computação



MONO.NET e SPRING.NET

Disciplina: Paradigmas de Linguagem de Programação

Professor: Ângelo Amaral

Bruno de Oliveira Teles // RA: 8062779752

Cecília Junqueira Sartini // RA: 8483182138

Douglas Cristiano // RA: 8483182181

Fabiano de Almeida // RA: 7278601753

Raone Thiago Cavalcante // RA: 1299263894

Jundiaí

2014

Sumário

| | |
|----------------------|----|
| Introdução..... | 3 |
| Desenvolvimento..... | 4 |
| SPRING.NET..... | 4 |
| EJB ou SPRING?..... | 6 |
| MONO.NET..... | 6 |
| JAVA X MONO.NET..... | 7 |
| Curiosidade:..... | 8 |
| Conclusão..... | 9 |
| Referências..... | 10 |

Introdução

O paradigma de programação orientada ao objeto trouxe consigo diversos conceitos e padrões que hoje são fundamentais na programação, diversas linguagens usam esses conceitos, como C#, C++ e o Java, eles são alguns dos diversos exemplos existentes no mercado, a programação orientada ao objeto veio para melhorar a forma como programamos, que antigamente era de uma forma procedural. (**Wikipédia: Framework, 2014**)

Dentre as melhorias que a OO trouxe ainda existia alguns problemas que eram difíceis de serem identificados e solucionados, quando se tem um sistema muito grande, é natural que diversos métodos executem as mesmas tarefas, que em muitos casos acabam repetindo o mesmo comando, percebendo que isso não era eficiente, criava um grande acoplamento entre as classes e não se tinha uma boa abstração, além de dificultar na manutenção do código, surgiram diversos frameworks distintos, dentre elas a Spring.Net.

Essa framework foi desenvolvida com base em Java, portanto roda em sistemas com a JVM, e por consequência utiliza os conceitos de OO, e utiliza os conceitos de Programação Orientada a Aspectos (POA), dentre outras particularidades da linguagem Java, como Interfaces e POJO's. (**Wikipédia: Orientação a Objetos, 2014**)

O Spring.Net surgiu de necessidades muito simples, fornece meios para diminuir o acoplamento entre classes, garantir maior abstração, integração com outras tecnologias, garantir um sistema estável, flexível e de fácil manutenção. (**Wikipédia: Framework, 2014**)

Com a evolução da tecnologia de programação, o nível de transformação em conceitos à programação viabilizou o compartilhamento de recursos em desenvolvimento de códigos e aplicações, que por ventura, possibilitou o funcionamento e comportamento de programas de software. Essa linguagem de programação orientada a objetos, utilizamos como padrões, e se destacam a C#, C++ e o Java, que são ainda uma das multiplataformas existente de maior proporção de programação até os dias atuais. Porém, nessa programação orientada a objetos havia problemas com o excesso de tarefas repetidas "comandos". Com a possibilidade de apresentar uma solução a esse problema surgiu distintos framework, o "Spring.net", como a grande solução para POA, Serviços e IOC. Esse framework criado por Rod Johnson e descrito em seu livro "Expert One-on-One: JEE Design e Development". Trata-se de um Framework não instrutivo, baseado nos padrões de projetos inversão de controle (IoC) e injeção de dependência. Com essa tecnologia oferecida se deu o compartilhamento e integração com outras tecnologias, tornando incrível a transformação de trabalho que os programadores e desenvolvedores passaram a ter através de estarem integrados a complexidade de acordo com o acoplamento entre seus objetivos.

Já a Mono.net, é uma tecnologia de plataforma pronta para desenvolver e criar aplicações que rodarão em vários sistemas operacionais distintos. Mono é um projeto de códigos que possibilita a expansão em multiplataformas, suas multiplataformas suportam a estrutura do Windows, Linux e MAC OS X. É óbvio que a possibilidade dessas tecnologias permanecerem com a responsabilidade de tornar a estrutura de aplicações e códigos dos programadores uma garantia estável de referência e adaptação ao mercado

O grupo não conhecia nenhum dos temas, então discutimos e escolhemos estes.

Desenvolvimento

SPRING.NET

O Spring.Net é um framework desenvolvido tendo base a plataforma Java, portanto ele é implementado em sistemas que possuem as JVM (Java Virtual Machine), para entendermos o Spring.Net é necessário entender alguns conceitos importantes, como o que é um framework e da onde surgiu sua necessidade, e entendermos um pouco sobre programação orientada ao objeto (POO) e Programação Orientada ao Aspecto (POA).

Spring.Net é um conjunto de instruções comuns que tem como objetivo fornecer soluções para diversos problemas comuns em aplicações, ou seja, quando programamos em alguma linguagem, acabamos criando classes, métodos, entre outros que partilham de um problema, que muitas vezes demandam de muito tempo do desenvolvedor e criam um acoplamento muito grande entre si, o Spring.Net cria soluções inteligentes para esse problema, o framework engloba as instruções para esse problema comum que facilitam a vida do desenvolvedor. (**Wikipédia: Framework, 2014**)

O Spring.Net deve ser flexível e poder ser extensível, permitindo que novos recursos sejam implementados a ele, garantindo uma maior quantidade de instruções para prover as soluções necessárias para as mais diversas aplicações. (**Universidade Federal de Campina Grande: O que é Frameworks**)

O conceito de paradigma de orientação ao objeto parte do princípio de fazermos uma análise cognitiva de como uma classe se comportaria caso existisse, ou seja, quais seus atributos, seu comportamento, com quais outros objetos ele pode/deve interagir, esse conceito é amplamente utilizado em diversas linguagens de programação, tais como: Java, C++, C#, VB.Net, entre outros. (**Wikipédia: Orientação a Objetos, 2014**)

Apesar da programação OO ter trago muitos conceitos que facilitam o desenvolvedor, entender e modelar o comportamento de cada classe ainda é algo difícil de ser feita, por isso surgiu diversas frameworks, como o Spring.Net, fazendo uso da OO, ele traz consigo o conceito de POA (Programação orientada ao Aspecto), esse conceito tem como objetivo fazer uma análise dos aspectos de uma classe.

Quando estamos desenvolvendo alguma aplicação é comum criarmos classes que possuem tarefas comuns entre si, por exemplo, podemos ter diversas classes que precisem acessar as informações de um determinado arquivo, então teríamos que criar linhas de código para executar essa tarefa, porém o Spring.Net é utilizado justamente para analisar esse aspecto da nossa aplicação e encapsular esse comando nele mesmo, ou seja, em vez de termos diversas classes fazendo a mesma coisa, podemos fazer com que as classes façam referência ao framework, que por sua vez fica encarregado de executar essa rotina para todas as classes. (**Wikipédia: Programação Orientada a Aspecto, 2013**)

É interessante analisar que o Spring.Net faz uso dos padrões da Inversão de Controle (IoF) e Injeção de Dependência.

Inversão de controle é onde a chamada de métodos é delegado a um container, ou seja, ele determina qual método será chamado dependente do que está sendo solicitado, na programação tradicional quem determina essa chamada é o próprio programador no momento do desenvolvimento. (**Wikipédia: Inversão de Controle, 2013**)

Injeção de dependência é o nome dado para quando o container fica responsável por declarar o nível de acoplamento entre os diversos módulos da aplicação, na programação padrão o próprio desenvolvedor determina qual módulo depende de qual e qual o grau do acoplamento, a injeção de dependência é uma técnica muito utilizado quando não se quer ter um acoplamento muito grande e desnecessário entre os diferentes módulos da aplicação.(**Wikipédia: Injeção de dependência**, 2013)

O Spring.Net tem como base de sua arquitetura POJO's e Interfaces:

POJO são classes simples em Java, ela possui basicamente seus getters e setters e não dependem de interfaces e de outras frameworks.(**GUJ: O que é POJO, afinal**, 2002)(**Wikipédia: Plain Old Java Objects**, 2013)

O conceito de interface parte do princípio que diferentes classes poderem executar uma mesma operação de formas diferentes, o framework obriga a classe a executar determinada tarefa, porém a forma como será feito varia para cada classe.(**GUJ: Interfaces**,2002)(**Caelun: Interfaces**, 20-.)(**Wikipédia: Interface (Ciência da Computação)**, 2013)

A framework possui diversos módulos que facilitam o desenvolvimento da aplicação, alguns deles são:

Spring.Core: É utilizada na gerência do tipo de objeto, através desse módulo a configuração da injeção de dependência fica mais fácil de ser feito.(**Spring Framework: Transaction management**, 20-?)

Spring.AOP: Esse módulo permite dizer o grau de acoplamento entre as classes, esse módulo tem como objetivo centralizar os métodos em comum da aplicação.(**Spring Framework: Aspect Oriented Programming with Spring.NET**, 20-.)

Spring.Data: Facilita a persistência de dados, graças a esse módulo o Spring.Net consegue trabalhar de forma confiável com diversas tecnologias, além de dar suporte a interfaces DAO (Data Access Object), que consistem em fazer com que partes diferentes da aplicação não enxerguem a outra, não interferindo no acesso ao banco de dados, diferentes partes da aplicação podem acessar o banco de dados, uma pode estar atualizando, outra visualizando, ou deletando, por exemplo.(**Wikipédia: Injeção de dependência**) (**Spring Framework: DAO support**, 20-.)(**Wikipédia: Objeto de Acesso a Dados**)

Spring.Data.Nhibernate: Este módulo tem como objetivo a integração como o Nhibernate através do gerenciamento feito usando o módulo Spring.Data, graças a ele a integração é mais fácil de ser feita.(**Spring Framework: DAO support**, 20-?.)

Spring.Testing.Nunit: Módulo utilizado para rodar testes de integridade utilizando a ferramenta Nunit.

Spring.Testing.Microsoft: Faz teste na integração assim como o Spring.Testing.Nunit, porém ele é utilizado na ferramenta MSTEST, ferramenta de testa da Microsoft.(**Spring Framework: Testing**)

Spring.Scheduling.Quartz: Integra o sistema com a ferramenta agendador de tarefas do Quartz.Net, ele ajuda na criação de rotinas pré-agendadas, que são executadas mediante uma determinada condição, estabelecidas pelo desenvolvedor.

(**Spring Framework: Scheduling and Thread Pooling**, 20-?.)

EJB ou SPRING?

EJB (Enterprise JavaBeans) – tem como objetivo simplificar o desenvolvimento de aplicações Java. Tendo assim, três tipos de EJB's: ENTITY BEANS, SESSION BEANS e o MESSAGE DRIVEN BEANS. O EJB trabalha com algumas empresas em destaque, como: IBM, Oracle e HP, acompanhado de uma grande comunidade de programadores. (**Wikipédia: Enterprise JavaBeans**)

Com a sua mudança atualmente para versão 3.1, tem se aproximado mais do Spring. Mas pelo Spring ser um modelo bem mais leve, e não requer um servidor de aplicação para se executar o código. Deixando o Spring assim com uma vantagem em questão de tempo de desenvolvimento, e a execução de testes. Em testes o Spring se mostra mais fácil para se trabalhar, até mesmo por conta do framework ser leve.

As críticas que rodeia o Spring é o de ter muita configuração XML. O que já foi descartado por usuários do mesmo! E ainda sim a vantagem com relação ao EJB é de não precisar alterar as classes incluindo anotações para deixá-las gerenciáveis pelo container.

As mudanças que houve nas versões do EJB, das versões 2.1 para 3.0 se diz respeito as anotações Java. Aonde vem a facilitar o desenvolvimento reduzindo a quantidade de códigos e o uso dos arquivos de configuração XML. Para ter acesso EJB, é preciso definir sua interface, que venha ser local ou remota. Sendo a local acessada somente pelo servidor de aplicação onde sendo executado. Quanto à remota permitindo acesso por elementos externos.

Quem sabe daqui alguns anos o EJB venha superar o SPRING. Mas levando em consideração o legado de usuários SPRING talvez isso não ocorra, mas a EJB, se mostra um ótimo candidato. (**Devmedia: EJB ou Spring, 2012**)

MONO.NET

Criado pela Xamarin (originado da Ximian), o Mono.net é um compilador multiplataforma de software livre criada para dar aos desenvolvedores facilidade em criar aplicações de plataformas cruzadas. É uma implementação do Framework .NET da Microsoft. (**Wikipedia: Mono Software**)(**Mono Project: What is mono**)(**Wikipedia: Xamarin**)

Suas multiplataformas suportadas são: Linux, Windows, BSD, Solares e Mac OS X. Sua edição de texto avançado suportam completamente modelos de códigos e código fonte. Sua bancada configurável consiste em janelas de layout completamente customizáveis, permissão ao usuário para definir atalhos do teclado, e ferramentas externas. Suporta múltipla linguagem de C#, Visual Basic.Net, C e C++. O Mono realiza uma limpeza nos aplicativos nativos, dessa forma tornando-os mais eficientes. (**MonoDevelop: Feature Highlights**)(**Slideshare: Mono.NET além do Windows, 20-?**)

Em Fevereiro de 2014, após cinco meses de desenvolvimento foi liberada sua nova versão, o Mono 3.2.7, trazendo com ele inúmeros benefícios, como por exemplo se podendo remover muito mais os limites para verificar as arquiteturas de 64bits. Foi adicionado também "Alias Analysis" e Código de Moção Invariável Loop que permitiu ainda mais melhorar códigos a serem gerados. Houve a melhoria de performance sob alguns pontos de referência, e também melhorou o trabalho do compilador,

identificando melhor um maior número de bugs. Agora tem uma porta inicial do mono para ARM (uma estrutura de processador com 32 bits usada especialmente em sistemas embarcados), hardfp (porta flutuante) e ABI (interface binária de aplicação), que são usados por vários distribuidores do Linux. Por fim como último benefício, passou a ter tempo de execução usado para 64bits, e comparar hardware quando de 32bits. (**Mono Project News: Mono 3.2.7 is out!**)(**Wikipédia: Arquitetura ARM**) (**Wikipédia: Interface binária de aplicação**)

Sua vantagem é que já existem vários desenvolvedores utilizando o framework. E com o Mono, podemos rodar nossas aplicações em várias plataformas citadas. Além disto, ainda temos acesso ao Class Library do .Net, e podemos desenvolver gerenciamento da memória usando reflections, generics e threading. Dentre diversos outros benefícios.

Como desvantagem, há que não se pode fazer chamadas à APIs (Interface de Programação de Aplicativos) do Windows, pois elas não funcionaram rodar sua aplicação com o Mono em outra plataforma. (**Tec Mundo: O que é API**, 2009)

É independente de linguagem. Aliás o correto seria chamá-lo de plataforma de desenvolvimento. Embora ainda não sejam todas, a tendência é de no futuro muitas linguagens serem suportadas. Como Python e Object Pascal, que até então são terceirizadas. O bom de tudo isso é que programadores podem criar classes em Java (por exemplo) e tais classes serem reaproveitadas em outros programas. Ou seja, Mono permite a portabilidade entre programas ou códigos escritos em diferentes linguagens. (**Viva o Linux: Mono é independente de linguagem**, 20-.)

JAVA X MONO.NET

O Java foi desenvolvido pela Sun Microsystems, e atualmente é de propriedade da Oracle com sete anos de vida, desenvolvida para grandes corporações e largamente empregada pelos líderes de mercado como IBM, trata-se de uma das linguagens mais usadas no meio comercial, sobretudo na programação de sistemas. Sua VM (Java Virtual Machine) está disponível sobre diversas plataformas e um dos seus princípios básicos é a execução interpretada do bytecode, o conjunto de instruções da JVM distribuídos tem maior fase e tempo de teste.(**COMPUTERWORLD: Java X. .Net: disputa acirrada no mercado nacional**, 2003)(**Stackoverflow: Is .NET/Mono or Java the better choice for cross-platform development?**, 2008?.)

A disputa é grande entre eles. A .NET é da Microsoft, e criada para empresas de pequeno e médio porte, se vale de algumas das linguagens de programação como Visual Basic (VB), C++ e C#, também tem a vantagem de ser criada após ver os erros do rival. Sua VM, (Common Language Infrastructure - CLI) é um padrão aberto desenvolvido junto com alguns parceiros industriais. A partir de sua especificação é possível que terceiros criem linguagens compatíveis, as quais, teoricamente, são capazes de serem executadas em diferentes plataformas computacionais.(**Universidade Federal do Ceará: COMPARAÇÃO ENTRE MÁQUINAS VIRTUAIS JAVA E CLI EM ALGORITMOS DE COMPUTAÇÃO NUMÉRICA INTENSIVA**, 20-?)

Curiosidade:

Por volta de março de 2013) nasceu a MonoGame, software livre criado para rodar jogos de plataformas específicas em suas concorrentes. No mesmo ano, em junho, a empresa SONY anunciou dois jogos desenvolvidos por ela. O MONO do PlayStation 4 para uso público ainda não foi divulgado. (**Facebook/ Monobrasil: Acompanhando a E3**)(**Wikipédia: MonoGame**)

Conclusão

Fica claro que o Spring.Net trouxe diversas melhorias em como as aplicações são desenvolvidas, fazendo o uso de diversos conceitos e recursos já existentes na linguagem Java.

Para as empresas isso significa um corte de custos considerável, já que é reduzida a quantidade de desenvolvedores necessários devido à complexidade de aplicação a ser desenvolvida.

A linguagem java vem crescendo, e hoje diversas aplicações fazem uso da linguagem, portanto o Spring.Net traz toda a robustez da linguagem que ainda cresce, mais as suas próprias particularidades que fazem esse framework ser extremamente recomendado para aplicações que usam a JVM.

O framework traz consigo uma série de módulos que garantem boa integração com diversas tecnologias de persistência, como o Hibernate por exemplo, que é muito usado entre os desenvolvedores, e ainda traz ferramentas para debug, como suporte ao NUnit, muito utilizado para fazer testes unitários, e ainda traz suporte para o MSTEST, ferramenta de testes da própria Microsoft. (**Spring Framework: Testing, 20-?**) (**Spring Framework: The IoC Container, 20-.**)

A cada dia novas tecnologias nascem para suprir as mais diversas necessidades, e o Spring.Net vem com uma proposta de diminuir o acoplamento, garantir maior abstração, e ainda ter um sistema robusto, confiável, flexível e que permita uma manutenção mais fácil para os desenvolvedores.

O projeto Mono é uma plataforma que permite aos desenvolvedores criarem facilmente aplicações que poderão rodar em vários sistemas operacionais diferentes. Ele é uma implementação OpenSource do Microsoft Framework .Net e podemos rodar desde aplicações web, como desktops. (**Código Fonte.net: O que é o Mono e como usá-lo?, 2008**)

O mono foi criado para pegar um código base e expandir em multiplataforma. Tornando o nível de trabalho de linguagem de programação um padrão confiável.

Java é realmente mais portátil. Mono não está implementado em todos os lugares, e isso fica para trás a implementação da Microsoft de forma significativa. O Java parece ficar em melhor sincronização entre plataformas e tem mais disponibilidade de ferramentas em todas elas, embora existam muitas ferramentas disponíveis para .NET em plataformas Windows.

Referências

Caelun: Interfaces, 20-. Disponível em: <<http://www.caelum.com.br/apostila-java-orientacao-objetos/interfaces/#10-2-interfaces>> Último acesso em: 24 de Março de 2014

Código Fonte.net: O que é o Mono e como usá-lo, 2008. Disponível em: <http://www.codigofonte.net/dicas/dotnet/321_o-que-e-o-mono-e-como-usa-lo> Último acesso: 22 de Março de 2014

COMPUTERWORLD: Java X .Net: disputa acirrada no mercado nacional, 2003. Disponível em <<http://computerworld.com.br/tecnologia/2003/06/17/idgnoticia.2006-05-15.5204280289/>> Último acesso em: 24 de Março de 2014

Devmedia: EJB ou Spring, 2012. Disponível em: <<http://www.devmedia.com.br/forum/ejb-ou-spring/430646>> Último acesso em: 24 de Março de 2014

Devmedia: Spring.NET - .Net Magazine 78, 20-?. Disponível em: <<http://www.devmedia.com.br/spring-net-net-magazine-78/18220>> Último acesso em: 24 de Março de 2014

Devmedia: Spring.NET, 20-?. Disponível em: <http://www.devmedia.com.br/websys.5/webreader.asp?cat=1&artigo=2948&revista=netmagazine_78#a-2948> Último acesso em: 24 de Março de 2014

Facebook: Acompanhando a E3, 2013. Disponível em: <<https://www.facebook.com/MonoBrasil>> Último acesso: 22 de Março de 2014

GUJ: Interfaces, 2002. Disponível em: < <http://www.guj.com.br/articles/123>>. Último acesso: 24 de Março de 2014

GUJ: O que é POJO, afinal?, 2002. Disponível em: < <http://www.guj.com.br/java/206651-o-que-e-pojo-afinal>>. Último acesso: 22 de Março de 2014

GUJ: EJB ou Spring, 2012. Disponível em <<http://www.guj.com.br/java/289146-ejb-ou-spring>> Último acesso: 24 de Março de 2014

GUJ: É bom o Spring ou o EJB é melhor?, 2011. Disponível em <<http://www.guj.com.br/java/229266-e-bom-o-spring-ou-o-ejb-e-melhor>> Último acesso: 24 de Março de 2014

JUNIOR, Francisco Heron Carvalho; MAGALHÃES, Francisco José Lins; SILVA, Jefferson de Carvalho. **Universidade Federal do Ceará: COMPARAÇÃO ENTRE MÁQUINAS VIRTUAIS JAVA E CLI EM ALGORITMOS DE COMPUTAÇÃO NUMÉRICA INTENSIVA**, 20-?. Disponível em: <<http://lia.ufc.br/~franzejr/ApresentacaoERCEMAPI.pdf>> Último acesso: 24 de Março de 2014

MonoDevelop: Feature Highlights, 20-. Disponível em: <www.monodevelop.com>Último acesso: 22 de Março de 2014

Mono Project: Cross Plataform, opens source .NET development framework,20-?. Disponível em:

<http://www.mono-project.com/Main_Page>Último acesso: 22 de Março de 2014

Mono Project News: Mono 3.2.7 is out!, 2014. Disponível em: <<http://news.mono-project.com/>>Último acesso: 22 de Março de 2014

Mono Project: What is mono, 20-?. Disponível em: <http://www.mono-project.com/What_is_Mono>Último acesso: 22 de Março de 2014

Slideshare: Elaboração de Referencias segundo ABNT/NBR 60:23: 2002, 2012. Disponível em: <<http://pt.slideshare.net/UnespRC/elaborao-de-referencias-abntnbr-60232002>>Último acesso: 24 de Março de 2014

Slideshare: Mono.NET além do Windows, 20-?. Disponível em: <<http://pt.slideshare.net/viniciusquaiato/mono-net-alm-do-windows>>Último acesso: 24 de Março de 2014

Spring Framework: Aspect Oriented Programming with Spring.NET, 20-?. Disponível em:

<<http://www.springframework.net/doc-latest/reference/html/aop.html>>Último acesso: 24 de Março de 2014

Spring Framework: DAO support, 20-?. Disponível em:

<<http://www.springframework.net/doc-latest/reference/html/dao.html>>Último acesso: 24 de Março de 2014

Spring Framework: Scheduling and Thread Pooling, 20-?. Disponível em: <<http://www.springframework.net/doc-latest/reference/html/scheduling.html#scheduling-quartz>>Último acesso: 24 de Março de 2014

Spring Framework: Testing, Disponível em: <
<http://www.springframework.net/doc-latest/reference/html/testing.html>>Último
acesso: 24 de Março de 2014

Spring Framework: The IoC Container, 20-?. Disponível em: <
<http://www.springframework.net/doc-latest/reference/html/objects.html>>Último
acesso: 24 de Março de 2014

Spring Framework: Transaction management, 20-?. Disponível em:
<<http://www.springframework.net/doc-latest/reference/html/transaction.html>>Último
acesso: 24 de Março de 2014

Stackoverflow: Is .NET/Mono or Java the better choice for cross-platform development?, 2008?. Disponível em: <<http://stackoverflow.com/questions/61559/is-net-mono-or-java-the-better-choice-for-cross-platform-development>> Último acesso: 24 de Março de 2014

Tec Mundo: O que é API, 2009. Disponível em:
<<http://www.tecmundo.com.br/programacao/1807-o-que-e-api-.htm>> Último acesso: 22 de Março de 2014

Universidade Federal de Campina Grande: O que é Frameworks, 20-?. Disponível
em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/oque.htm>>Último
acesso: 24 de Março de 2014

Viva o Linux: Mono é independente de linguagem?, 20-?. Disponível em:
<[http://www.vivaolinux.com.br/artigo/Mono-e-C-\(parte-2\)?pagina=1](http://www.vivaolinux.com.br/artigo/Mono-e-C-(parte-2)?pagina=1)> Último acesso: 22 de Março de 2014

Wikipédia: Arquitetura ARM, 2013. Disponível em
<http://pt.wikipedia.org/wiki/Arquitetura_ARM> Último acesso: 22 de Março de 2014

Wikipédia: Enterprise JavaBeans, 2014. Disponível em
<http://pt.wikipedia.org/wiki/Enterprise_JavaBeans> Último acesso: 24 de Março de 2014

Wikipédia: Interface binária de aplicação, 2013. Disponível em: <
http://pt.wikipedia.org/wiki/Interface_bin%C3%A1ria_de_aplica%C3%A7%C3%A3o>
Último acesso: 22 de Março de 2014

Wikipédia: Interface (Ciência da Computação), 2013. Disponível em:
<[http://pt.wikipedia.org/wiki/Interface_\(ci%C3%Aancia_da_computa%C3%A7%C3%A3o\)](http://pt.wikipedia.org/wiki/Interface_(ci%C3%Aancia_da_computa%C3%A7%C3%A3o))>Último acesso: 22 de Março de 2014

Wikipédia: Injeção de dependência, 2013. Disponível em: <http://pt.wikipedia.org/wiki/Inje%C3%A7%C3%A3o_de_depend%C3%Aancia> Último acesso: 22 de Março de 2014

Wikipédia: Inversão de Controle, 2013. Disponível em: <http://pt.wikipedia.org/wiki/Invers%C3%A3o_de_controle> Último acesso: 22 de Março de 2014

Wikipédia: Framework, 2014. Disponível em: <<http://pt.wikipedia.org/wiki/Framework>> Último acesso em 24 de Março de 2014

Wikipédia: MonoGame, 20-?. Disponível em: <<http://en.wikipedia.org/wiki/MonoGame>> Último acesso: 22 de Março de 2014

Wikipédia: Mono Software, 2014. Disponível em: <[http://en.wikipedia.org/wiki/Mono_\(software\)](http://en.wikipedia.org/wiki/Mono_(software))> Último acesso: 22 de Março de 2014

Wikipédia: Objeto de Acesso a Dados, 2013. Disponível em: <http://pt.wikipedia.org/wiki/Objeto_de_acesso_a_dados> Último acesso: 22 de Março de 2014

Wikipédia: Orientação a Objetos, 2014. Disponível em: <http://pt.wikipedia.org/wiki/Orienta%C3%A7%C3%A3o_a_objetos> Último acesso: 22 de Março de 2014

Wikipedia: Plain Old Java Objects, 2013. Disponível em: <http://pt.wikipedia.org/wiki/Plain_Old_Java_Objects>. Último acesso em: 22. Setembro. 2014.

Wikipédia: Programação Orientada a Aspecto, 2013. Disponível em: <http://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_orientada_a_aspecto> Último acesso em 24 de Março de 2014

Wikipédia: Xamarin, 2011. Disponível em: <<http://en.wikipedia.org/wiki/Xamarin>> Último acesso: 22 de Março de 2014