

# Verificar a portabilidade de aplicações .NET quando executadas em ambientes diversos (Windows, Web e Mobile)

Alexandre Coutinho Evangelista<sup>1</sup>, Luiz Gustavo Galves Mählmann<sup>2</sup>

<sup>1</sup> Acadêmico do Curso de Sistemas de Informação da ULBRA Guaíba  
<alex.evangelista@gmail.com>

<sup>2</sup> Professor Orientador do Curso de Sistemas de Informação da ULBRA Guaíba  
<mahlmann@gmail.com>

*Resumo: Este artigo tem o objetivo de mostrar o andamento do projeto de verificação de portabilidade de aplicações .NET em mais de um ambiente, mais especificamente os ambientes Windows, Web e Mobile. O projeto consiste em verificar o nível de portabilidade que uma aplicação pode ter nestes três ambientes, para isso uma metodologia será desenvolvida demonstrando os aspectos existentes entre eles. A comprovação desta metodologia será feita através do desenvolvimento de uma solução multi-ambiente para o Departamento de Criminalística do Instituto-Geral de Perícias do Estado. As tecnologias que serão utilizadas para o desenvolvimento desta aplicação são: Servidor Web IIS, Linguagem de programação C# e Banco de dados Microsoft SQL Server 2005.*

*Abstract: This article proposes to show the development project of verification .NET applications portability in more than one environment, more specifically in Windows, Web and Mobile environments. This project consists in verifying the level of portability of an application on these three environments, and to achieve this goal, it will be developed a methodology by a multi-platform solution for the Departamento de Criminalística do Instituto-Geral de Perícias do Estado. The technologies that will be used for the development of this application are: IIS Web Server, C# programming language and Microsoft SQL Server 2005 database.*

## 1 - INTRODUÇÃO

Atualmente, desenvolver uma aplicação para o ambiente *Windows*, com diversas funcionalidades, e disponibilizar um módulo desta aplicação para que seja executada em outro ambiente, não é nenhuma novidade. Porém, essa situação deve ocasionar um retrabalho considerável para o desenvolvedor, porque a reutilização de código, por se tratar de ambientes distintos e até mesmo em exigir o emprego de linguagens de programação diferentes, é praticamente inviável.

Além destas justificativas já comentadas, dificilmente o profissional que desenvolveu a aplicação *Windows* será o mesmo que desenvolverá o módulo para *Web* ou para um dispositivo móvel, pois o grau de conhecimento deste profissional terá que ser muito amplo (variado) o que é difícil nos dias atuais haja vista o grau de especialização que um profissional de TI tem que ter.

Diante deste cenário a proposta deste artigo é apresentar o andamento de um trabalho que tem como meta o desenvolvimento de uma metodologia que propicie, utilizando-se uma única linguagem de programação, uma eficiente reutilização de código para o desenvolvimento de aplicações multi-ambientes (*Windows*, *Web* e *Mobile*) através do emprego da plataforma .NET (THAI, 2001).

Após o detalhamento desta metodologia será desenvolvida uma solução para um caso prático do Departamento de Criminalística do Instituto-Geral de Perícias a fim de comprovar a eficiência e a eficácia dos conceitos elaborados na metodologia.

Como o trabalho será feito utilizando-se a plataforma .NET faz-se necessário uma revisão das suas funcionalidades, dos seus recursos, assim como da maneira que a integração entre os ambientes poderá ser feita.

#### OBJETIVOS DO TRABALHO

O objetivo deste projeto é detalhar uma metodologia de projeto que priorize a portabilidade de uma aplicação, isto é para que a mesma possa ser executada em mais de um ambiente, facilitando a reutilização de código além de reduzir ao máximo o nível de retrabalho (do ponto de vista do desenvolvedor), pois a idéia é que a aplicação seja portátil, isto é, possa ser executada em outros ambientes.

Especificamente, a pesquisa irá focar a eficiência e eficácia do emprego de uma linguagem de programação em uma única aplicação, mas que essa seja executada independente do ambiente escolhido, podendo ser o próprio desktop, através de um navegador de internet ou até mesmo de um dispositivo móvel.

Os ambientes estudados para a execução desta aplicação são:

- Windows
  - Microsoft Windows XP e Microsoft Windows Vista
- Web
  - Navegador Internet Explorer e Mozilla Firefox
- Mobile
  - Smartphone (YANG, 2007)
    - Microsoft Windows Mobile 5 e 6 (YANG, 2007)
  - Pocket PC (YANG, 2007)
    - Microsoft Windows Mobile 5 e 6

## 2 - REFERENCIAL TEÓRICO

Diante do exposto faz-se necessário a realização de uma revisão sobre os recursos propiciados pela plataforma .NET e suas funcionalidades.

Para falarmos da plataforma .NET, não pode-se esquecer de outra poderosa plataforma –JAVA (DEITEL, 2001) – que oferece inúmeros recursos, JAVA como uma linguagem de programação, é orientada a objetos, a criação dela foi baseada no C e C++ (DEITEL, 2001), por ser uma linguagem orientada a objetos, ela prioriza a organização e a reutilização de classes e componentes, JAVA como uma plataforma, tem um aspecto importante que é a independência de plataforma.

Como o desenvolvimento de uma aplicação far-se-á necessário para a comprovação da metodologia criada, a adoção da UML (PIOLINE, 2005) para realização da modelagem do projeto torna-se imprescindível, assim como do Diagrama Entidade-Relacionamento (CHEN, 1990) para a planificação do Banco de Dados da aplicação.

### 2.1 PLATAFORMA JAVA

A plataforma JAVA oferece uma série de pacotes e definições para que os desenvolvedores possam criar soluções robustas, incluindo recursos de conectividade de banco de dados, rede, web e interfaces para aplicações desktop. Ela utiliza uma máquina virtual, chamada JVM (*Java Virtual Machine*) (DEITEL, 2001), com ela instalada no sistema operacional, é a garantia que uma aplicação desenvolvida em JAVA deve ser executada sem problemas.

Diante destas características podemos citar um problema nesta plataforma, pois ela não oferece integração com outras linguagens de programação, sendo que o princípio da plataforma é justamente que se utilize uma única linguagem para tudo. Dessa forma não existe a integração de aplicações desenvolvidas em outras linguagens de programação com as que foram desenvolvidas em JAVA.

### 2.2 PLATAFORMA .NET

A plataforma .NET foi desenvolvida pela Microsoft durante 4 anos e seu lançamento oficial ocorreu em junho de 2000, a estratégia desta plataforma era de

explorar ao máximo a integração de serviços através da *Web*, fazendo que tanto aplicações *Web* ou *Desktop* se comuniquem de uma forma simples, independente do sistema operacional e linguagem de programação.

Um diferencial significativo que a plataforma .NET oferece é o suporte a uma série de linguagens de programação, com isso uma mesma solução pode ser desenvolvida em mais de uma linguagem de programação ou até mesmo utilizar uma solução existente e integrar ela em um novo projeto, economizando em retrabalho.

Nos itens a seguir são apresentados os componentes da plataforma .NET.

- *.NET Framework*: Um *framework* (THAI, 2001) é uma estrutura pré-definida que contém um conjunto de componentes de software oferecendo uma estrutura padrão para o desenvolvimento de uma aplicação. O .NET Framework (THAI, 2001) oferece suporte a mais de 20 linguagens de programação, independentemente da linguagem escolhida, sendo ela suportada por este *framework*, o resultado da aplicação será o mesmo.
- *CLR (Common Language Runtime)*: O CLR (THAI, 2001) é o ambiente que faz a execução das aplicações desenvolvidas dentro do .NET Framework, é a máquina virtual similar ao JVM (*Java Virtual Machine*), ou seja, está entre a aplicação desenvolvida e o sistema operacional. Apesar de estas aplicações executarem no ambiente *Windows*, elas não são aplicações Win32, assim que uma aplicação é executada o Win32 chama o *runtime* .NET fazendo com que ele assuma o controle da aplicação executada podendo gerenciar a memória, diminuindo a chance de ocorrer erros durante sua execução. O CLR é o ambiente de execução da plataforma .NET, portanto independente da linguagem escolhida ele é o mesmo.
- *BCL (Base Class Library)*: A BCL (THAI, 2001) é a biblioteca de classes da plataforma .NET, nela estão as classes básicas para que uma aplicação possa ser desenvolvida. O uso dela facilita o processo de desenvolvimento, pois existe a possibilidade da reutilização de componentes de software, simplificando e agilizando toda a elaboração de uma aplicação. Nesta biblioteca é possível encontrar classes referentes a interfaces, manipulação de arquivos, gerenciamento de memória, funcionalidades de web, rede entre outras. As classes estão organizadas de forma hierárquica dentro de uma estrutura chamada *namespace* (THAI, 2001).

- *.NET Compact Framework*: (YANG, 2007) É a materialização do .NET Framework, específico, para dispositivos móveis. Devido às diferenças de *hardware* entre um computador e um dispositivo móvel, foi necessário criar um *framework* exclusivamente para este segmento de dispositivos. O funcionamento da CLR e BCL, que são os principais componentes do framework, seguem os mesmos princípios, porém na BCL existem classes que foram criadas especificamente para dispositivos móveis, assim como outras foram retiradas por fugirem do contexto destes dispositivos.
- *Web Services*: Um dos principais objetivos da plataforma .NET é a integração de aplicações *web* com aplicações *desktop*. Isso é possível graças a utilização de *Web Services* (SHORT, 2002), pois com a utilização deles é possível fazer com que aplicações de linguagens e ambientes diferentes se comuniquem. A troca de informações entre as aplicações envolvidas é feita através do formato XML (SHORT, 2002), que é um padrão aberto, portanto é independente de linguagem de programação ou plataforma. Diante da exigência de fazer com que aplicações *Windows*, *Web* e *Mobile* se comuniquem é inevitável que o desenvolvimento de *Web Services* seja necessário, com ele será possível reaproveitar o mesmo *Web Service* na integração dos três ambientes.

### 2.3 UML

É a linguagem de modelagem unificada (*Unified Modelling Language*). Através de seus diagramas ela facilita a especificação, visualização e documentação de modelos de sistemas orientados a objeto.

A fim de ter uma razoável modelagem para a solução proposta, para o Departamento de Criminalística, optou-se por utilizar alguns dos diagramas que compõem a UML. Serão utilizados os seguintes diagramas:

- *Diagrama de Caso de Uso* (PIOLINE, 2005): Capturam as exigências funcionais do sistema. Mostrando os atores que vão interagir com o sistema e os casos de uso com os respectivos relacionamentos.
- *Diagrama de Atividades* (PIOLINE, 2005): Captura o fluxo de um comportamento ou atividade, para o próximo. É semelhante a um fluxograma clássico, entretanto é mais expressivo.

- *Diagramas de Interação* (PIOLINE, 2005): Usados para mostrar a interação formada pelos objetos e seus relacionamentos, incluindo as mensagens trocadas entre os mesmos. Os tipos de diagramas de interação são os de Seqüência e os de Colaboração. Eles são equivalentes, podendo ser um convertido em outro, entretanto não é uma regra que eles visualizem exatamente as mesmas informações.
- *Diagrama de Classe* (PIOLINE, 2005): Usa classes e interfaces para capturar detalhes sobre as entidades que compoem seu sistema e as relações estáticas entre eles.

## 2.4 DIAGRAMA DE ENTIDADE-RELACIONAMENTO

Na solução proposta será utilizado um banco de dados relacional, portanto a modelagem será realizada através de um diagrama entidade-relacionamento (CHEN, 1990), pois ele é a representação gráfica dos relacionamentos entre as tabelas e as associações existentes entre seus respectivos atributos.

## 2.5 ANALISANDO AS VERSÕES DO .NET FRAMEWORK QUANTO A SUA PORTABILIDADE

Como o desenvolvimento da aplicação envolve o ambiente para dispositivos móveis, é necessária a utilização de um framework específico para este tipo de hardware, portanto aspectos de portabilidade, diferenças e similaridades entre os ambientes e seus respectivos frameworks serão vistos abaixo.

### 2.5.1 Portabilidade

Segundo (IEEE 90), portabilidade é a facilidade com que um sistema ou um componente pode ser transferido de um hardware ou software para outro ambiente.

Para concluirmos que uma aplicação é portátil, é preciso definir qual o seu grau de portabilidade, a medição é feita em relação a dois aspectos:

- As diferenças entre o ambiente original e o novo ambiente que a aplicação será desenvolvida.
- A capacidade do desenvolvedor em reutilizar classes e métodos comuns para dois ou mais ambientes, diminuindo o esforço exigido.

### 2.5.2 Diferenças entre os ambientes

A utilização da plataforma .NET facilitou o uso de uma única linguagem de programação para o desenvolvimento nos três ambientes, portanto a linguagem C# (TROELSEN, 2007) é portátil nos ambientes já mencionados, as diferenças que existem entre eles basicamente estão nas interfaces gráficas e nas classes que são específicas para cada tipo de plataforma.

Essas diferenças ocorrem porque estamos utilizando dois frameworks diferentes, o .NET Framework que é utilizado para os ambientes *Windows* e *Web*, já para os dispositivos móveis é utilizado o .NET Compact Framework, neste *framework* existem classes específicas para este tipo de hardware, além de classes específicas para os ambientes *Windows* e *Web* terem sido removidas, devido à limitação que estes dispositivos tem em relação aos computadores.

### 2.5.3 Similaridades entre os ambientes

Nos ambientes *Windows* e *Web*, como será utilizado o .NET Framework, com exceção da parte gráfica, praticamente todas as classes serão reutilizáveis, da mesma forma no ambiente *Mobile*, a aplicação para Smartphone e Pocket PC como utilizam o .NET Compact Framework, suas classes poderão ser reutilizadas.

Para que possa ser feita a integração de aplicações de um *framework* para o outro, é necessária a utilização de *Web Services*, pois neles são criadas classes que são comuns entre os ambientes, como o acesso ao banco de dados, por exemplo.

Com a elaboração de *Web Services* o ciclo de integração entre os ambientes é completo podendo assim avaliar o grau de portabilidade que a aplicação tem.

## 2.6 MVC – MODEL-VIEW-CONTROLLER

O padrão MVC (GAMMA, 1994) é utilizado há algum tempo, pois ele foi criado para a construção de interfaces no ambiente Smalltalk (BIBLIOGRAFIA). Utilizar este padrão no projeto consiste em dividir a aplicação em três camadas.

Com esta divisão a implementação e manutenção da aplicação são facilitadas, pois caso haja a necessidade de alterar alguma interface, as regras de negócios não é afetada, pois estão em outra camada. Podemos dizer que estas camadas são divididas em:

- *Modelo*: É a principal camada do modelo, pois ela contempla as regras de negócio da aplicação, que são as funcionalidades da aplicação, nela consiste a parte do armazenamento, manipulação e geração dos dados além de ser independente da camada de visualização.
- *Visualização*: É a camada responsável pela visualização das informações da aplicação, nela estão contidas as interfaces da aplicação, as interfaces são responsáveis pela entrada e saída das informações.
- *Controle*: É a camada intermediária entre o modelo e a visualização, pois ela é responsável pelo controle dos eventos da aplicação.

### **3 - SOLUÇÃO PROPOSTA**

Após a realização dos estudos na plataforma .NET o próximo passo no TCC I será definir uma metodologia de desenvolvimento para a plataforma .NET que contemple as diversidades existentes entre os ambientes (*Windows, Web e Mobile*), procurando identificar o que pode ser objeto de programação compartilhada ou não.

A partir da definição dessa metodologia, será desenvolvido um modelo para validar e verificar o funcionamento nestes três ambientes. Além de conhecer o nível de portabilidade existente, aspectos como reutilização de métodos, classes e interfaces serão consideradas, até mesmo a maneira de como os três ambientes se integrarão referentes ao acesso a dados.

O modelo desenvolvido no TCC I servirá de base para que no TCC II seja desenvolvida uma aplicação para o Departamento de Criminalística do Instituto-Geral de Perícias do Estado, com a finalidade de demonstrar os resultados obtidos ao longo da pesquisa.

A proposta é desenvolver uma única aplicação para o Departamento, sendo executada em ambientes *Windows, Web e Mobile*, com total integração entre eles. A aplicação consiste no controle das atividades dos peritos, podendo, as mesmas estarem em andamento, finalizadas e ainda a serem realizadas.

### **4 - SITUAÇÃO ATUAL DO PROJETO**

Com base no referencial teórico visto anteriormente, foi possível chegar nas seguintes constatações:

- Para proporcionar uma melhor estrutura e organização da aplicação foi adotado o padrão MVC, dividindo a aplicação em pelo menos três camadas.
- Diante das diferenças entre os frameworks estudados, existe a necessidade do desenvolvimento da camada de interfaces para cada um dos ambientes que o projeto atende.
- Foi concluído que a camada que envolve as regras de negócios é comum para os ambientes do .NET Framework, porém estas mesmas classes até o presente momento não puderam ser referenciadas no .NET Compact Framework.
- A integração entre as aplicações será feita através da utilização de Web Services.

## **5 - CONCLUSÃO**

Existem aplicações em diversas áreas de atuação muito bem consolidadas no mercado, os dispositivos móveis já não são mais uma novidade e sim uma realidade que vêm tomando uma dimensão cada vez maior. Sua infinidade de recursos faz com que aplicações possam ser usadas em qualquer local, sem que seja necessário o usuário estar em casa ou no seu trabalho.

Portanto, pode-se concluir que a união destes ambientes em uma única aplicação pode trazer diversos benefícios para todas as partes envolvidas. Desse modo, será possível desenvolver uma aplicação e adaptar apenas o que for necessário para outros ambientes, propiciando uma economia de esforço considerável para o desenvolvedor.

A aplicação não ficará dependente a um único ambiente. Com isto o usuário final terá maior mobilidade na execução de suas tarefas, além de um ganho de produtividade, agilidade, controle e confiabilidade.

Além disso, espero que este trabalho contribua para que novas soluções nesse âmbito surjam, fazendo com que diferentes ambientes e plataformas sejam

transparentes quando da execução de aplicativos desenvolvidos a partir da metodologia que está sendo elaborada.

## PROPOSTA PARA TCC-II

Para o TCC-II será realizado o desenvolvimento do sistema para o Departamento de Criminalística baseado na metodologia especificada em TCC I. A elaboração deste sistema será compreendida das seguintes atividades:

- Análise do sistema atual do Departamento de Criminalística.
- Requisitos da aplicação.
- Modelagem da aplicação.
- Desenvolvimento da aplicação com base no modelo desenvolvido no TCC-I.
- Testes.
- Entrega da aplicação.

## 6 - BIBLIOGRAFIA

CHEN, Peter. **Modelagem de Dados. A Abordagem Entidade-Relacionamento para Projeto Lógico**, 1990.

DEITEL, Harvey M. **C# How to Program**. [978-0130622211], 2001.

IEEE. **IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries**, 1990.

GAMMA, Erich. **Design Patterns: Elements of Reusable Object-Oriented Software**, 1994.

PILONE, Dan. PITMAN, Neil. **UML 2.0 in a Nutshell**. [0-596-00795-7], 2005.

THAI, Thuan. LAM, Hoang. **NET Framework Essentials**. [0-596-00165-7], 2001.

TROELSEN, Andrew. **Pro C# 2008 and the .NET 3.5 Platform, 4<sup>th</sup> Edition**. [978-1-59059-884-9], 2007.

SHORT, Scott. **Building XML Web Services for the .NET Platform**, [0-735-61406-7], 2002.

YANG, Baijian. ZHENG, Pei. NI, Lionel. **Professional Microsoft Smartphone Programming**. [0-471-76293-5], 2007.