

**DESENVOLVIMENTO DE UM APLICATIVO INTEGRADO BASEADO EM DATA  
SCRAPPING – WINDOWS 8 APP**

Julio Cesar DIAS

Rafael Miranda Brandão VIEIRA

Roque Redeford da SILVA

## RESUMO

Uma aplicação para *Windows 8* destaca-se pela forma como é integrada ao novo padrão de interface da mais nova versão do Sistema Operacional da *Microsoft*, permitindo uma interatividade mais dinâmica e prática. Este trabalho tem como objetivo explorar recursos e tecnologias emergentes para a criação de uma aplicação totalmente integrada que oferecerá a função de trazer notícias dos mais conhecidos portais de notícias da *Internet*, diretamente para uma interface do usuário, reunindo tudo em um único ambiente, não sendo necessária a utilização de um navegador *web*. Para tal, utilizou-se uma técnica denominada Raspagem de Dados, do inglês *Data Scrapping*, que permite extrair informações de páginas *web*, assim como uma ferramenta pronta para auxiliar na realização desse trabalho, o *HTML Agility Pack*. Como solução, foi desenvolvido um *Web Service*, recurso utilizado para integrar sistemas e aplicações que necessitam de dados que não se encontram em seu contexto.

**Palavra Chave:** *Windows 8*; aplicação integrada; raspagem de dados; *HTML Agility Pack*; *Web Service*.

# 1 INTRODUÇÃO

O *Windows 8*, a mais recente versão do sistema operacional da *Microsoft*, quando da produção deste trabalho, introduz uma nova experiência em aplicativos com recursos e integração exclusivos. Através da *Windows Store*, loja de aplicativos do *Windows*, esses *apps* (*applications*), que são programas específicos que podem ser baixados, instalados e executados em certos tipos de equipamentos eletrônicos, são publicados gratuitamente ou comercializados por desenvolvedores interessados.

A *Microsoft* disponibiliza as ferramentas e tecnologias necessárias para a criação dessas aplicações. Através do *Microsoft Virtual Academy*, é possível buscar auxílio passo a passo para a criação de *apps*, desde a instalação do sistema operacional *Windows 8*, passando pela configuração do ambiente de desenvolvimento, o próprio desenvolvimento até a publicação da *app* na *Windows Store*.

## 1.1 Objetivos

Construir um aplicativo *Windows Store* baseando-se no estudo das novidades do *Windows 8* e outras tecnologias já existentes.

Este aplicativo desempenhará a função de realizar amostragem de notícias, filtrando e selecionando as mais lidas ou em destaque de portais de notícias conhecidos na Internet e trazendo-as até o usuário de forma reunida e organizada, através de uma aplicação *desktop*, descartando o uso de um navegador e o acesso manual a cada um dos portais de notícias.

## 2 REFERENCIAL TEÓRICO

### 2.1 Microsoft Windows 8

Lançado mundialmente em 26 de outubro de 2012, o novo sistema operacional da *Microsoft*, trouxe inúmeras novidades e melhorias na sua oitava versão. Com uma nova arquitetura, o sistema possui uma versão que trabalha com os tradicionais *chips* da linha x86 para PCs (*Personal Computers*) e *notebooks* e uma nova versão para microprocessadores ARM, padrão em aparelhos portáteis, como *tablets* e *smarthphones* (GUGELMIN, 2012). De acordo com Honeycutt (2012), todos os recursos existentes na versão anterior do sistema, o *Windows 7*, estão presentes no *Windows 8*, mas de forma aperfeiçoada.

Entre as novidades apresentadas no *Windows 8*, Vasque (2012) destaca a compatibilidade com o *Kinect*<sup>1</sup>, a economia de memória RAM (*Random Access Memories*), devido a modificação de sistemas dentro da própria arquitetura do sistema operacional, maior velocidade de inicialização e desligamento, devido a uma nova ferramenta que simula um estado semelhante ao de “hibernação”, garantindo uma diminuição de até 70% nos tempos de inicialização e desligamento, maior segurança, através de um sistema de *boot* seguro, que promete deixar o sistema com um nível de proteção contra *malwares* semelhante ao do sistema operacional *Linux*, a implementação ao suporte ao *touchscreen* e a introdução a nova interface *Metro*, mais elegante e organizada, funcionando de maneira mais ligeira e dinâmica, dispensando o uso do antigo menu iniciar.

### 2.2 Desenvolvimento de Apps

Aplicações para *Windows 8* são desenvolvidas utilizando-se a ferramenta de desenvolvimento *Visual Studio* e são compartilhadas através da *Windows Store*, com o funcionamento semelhante a lojas de aplicativos já existentes em outras plataformas. Vale também ressaltar que qualquer versão do mesmo pode ser utilizada para a criação de *Apps* para *Windows*, sendo as versões *Professional*, *Premium* e *Ultimate* as mais usadas por desenvolvedores. Em relação a linguagens de programação, o *Windows 8 Apps* é bem flexível, podendo ser escritas em, por exemplo, *JavaScript*, *Visual Basic*, *Visual C#* e *Visual C++*.

A *Microsoft* exige que um *App* realize algo de valor, tenha mais funcionalidade que um simples *Website*, possua práticas de segurança e privacidade, seja facilmente identificado com

---

<sup>1</sup> Sensor de captura de movimentos desenvolvido pela Microsoft

um nome único, entre outras. Todas essas e outras exigências são encontradas no manual de requerimentos para certificação de um *Windows 8 App*, disponibilizado no site da MSDN (*Microsoft Developer Network*).

Uma vez aprovado, a *Microsoft* realiza a assinatura digital para prevenir adulterações do *App* e, posteriormente, lança o aplicativo na *Windows Store*.

### **2.3 Microsoft Windows Store**

A *Windows Store* é a loja oficial de aplicativos criada pela *Microsoft*. Liberada por volta de fevereiro de 2012, a loja foi criada com o intuito de fornecer aplicações, previamente aprovadas pela empresa, sejam gratuitas ou não, para o mais novo sistema operacional da *Microsoft*, o *Windows 8*. A loja está disponível totalmente em português (VELOSO, 2011).

## **2.4 Ambiente Windows 8 para desenvolvimento**

### **2.4.1 WPF**

O WPF apareceu em 2001 com o codinome “Avalon”, como a nova tecnologia de apresentação do *Windows Vista*. Dentre suas principais características estão: Flexibilidade da interface; Incorpora todas as funções do .NET 2.0, acrescentando às interfaces novos recursos como 3D, animações, gráficos vetoriais, reconhecimento de voz, *layouts* avançados, entre outros; Leva para o *desktop* o conceito já existente na *Web* de separação entre o *design* e o código; É independente de plataforma: o mesmo código-fonte funciona tanto na *Web* quanto para *desktop* e, no futuro, poderá ser distribuído até em sistemas como *Mac*, *Linux* e celulares, usando o WPF/E (*WPF Everywhere*, que está em desenvolvimento).

### **2.4.2 XAML**

A XAML ou *Extensible Application Markup Language* é uma linguagem baseada em XML (*eXtensible Markup Language*) criada pela *Microsoft* sendo fundamental para o desenvolvimento de aplicações WPF (*Windows Presentation Foundation*).

A XAML (pronúncia *zammel*) é a linguagem usada para criar interfaces com usuário e representar dados em aplicações WPF, sendo também usada para criar *workflows* em aplicações WF - *Windows Workflow Foundation*. (A XAML também é usada no *Silverlight*). A XAML pode ser compilada ou interpretada; Uma das maiores facilidades de se

utilizar o XAML é poder separar a interface de usuário e o código da aplicação. Este modelo de trabalho permite uma separação das tarefas de desenvolvimento de *software*.

### 2.4.3 Javascript

*Javascript* é uma linguagem “que liga uma página web a uma experiência interativa.” (Morrison, p.4, 2008) levando o cliente a ter uma maior interatividade com os mais variados recursos que uma página Web proporciona.

A tecnologia *Javascript* funciona juntamente com HTML (*HyperText Markup Language*) como um dos modernos componentes para a construção de uma página *web* (Morrison, p.4, 2008) e entra em ação sempre que solicitado.

O *Javascript* é atualmente a principal linguagem para programação *client-side* em navegadores *web*. Foi concebida para ser uma linguagem *script* com orientação a objetos baseada em protótipos, tipagem fraca e dinâmica e funções de primeira classe.

### 2.4.4 WinRT

Com o lançamento do *Windows 8*, o ambiente *WinRT* passa ser a plataforma para a criação de aplicações *Windows Store*.

O *WinRT* é uma API (*Application Programming Interface*) totalmente orientada a objeto, que permite o acesso as funcionalidades do sistema operacional para construção de aplicativos rápidos que usam operações assíncronas para as tarefas que demorem mais de 15 milissegundos à executar (Luís Abreu, p.19, 2013).

Com o *WinRT*, aplicações anônimas não serão mais permitidas. Inicialmente as aplicações poderão ser auto assinadas. Mas, para publicação na *App Store*, deverão ser assinadas com um certificado verdadeiro.

### 2.4.5 HTML 5

Desenvolvido por Tim Berners-Lee, o HTML é abreviação de *HyperText Markup Language*, ou seja, em português, linguagem de marcação de hipertexto. Pode-se marcar conteúdos nas páginas ou nos aplicativos *WinRT* como áudio, vídeos, imagens, etc. Atualmente, o HTML encontra-se na versão 5, que é uma evolução do HTML 4 com algumas novidades da W3C (*World Wide Web Consortium*) onde o objetivo é facilitar a manipulação dos elementos possibilitando que o desenvolvedor modifique as características dos objetos de forma não intrusiva e de maneira que seja transparente para o usuário final. Ao contrário das outras versões, a versão 5 oferece maior interatividade com o *Javascript* e o CSS tornando a página web mais leve e funcional.

### 3 MATERIAL E MÉTODOS

Para realizar a conclusão do trabalho em questão, cujo objetivo é desenvolver uma aplicação para o sistema operacional *Windows 8* – ou *Windows 8 App* – será necessário, primeiramente, uma pesquisa descritiva a respeito das novas características do sistema operacional, do ambiente para a criação de aplicações – *WinRT* –, assim como seu funcionamento e comportamento. Também será de suma importância, o estudo do maior número possível de tecnologias que poderão ser utilizadas na criação da aplicação, para a realização de um aplicativo bem mais completo, onde proporcionará ao usuário final, uma variedade muito maior de interações e funcionalidades e, conseqüentemente, estará ligado ao futuro sucesso do trabalho.

Na parte prática, o emprego de tais tecnologias será feito através da ferramenta de desenvolvimento *Visual Studio 2012* com o sistema de desenvolvimento (SDK) do *Windows 8*, o *Blend* para *Visual Studio* e modelos de projeto inclusos. A equipe será formada por três desenvolvedores, que farão uso de máquinas – *desktops* ou *notebooks* - que atendam as configurações mínimas de *hardware* necessário para a execução satisfatória do ambiente de trabalho.

Por fim, após a conclusão do projeto, o *software* será submetido a testes, visando avaliar a solidez e eficiência do mesmo. Para tal, serão executados testes de qualidade e funcionamento em ambiente de usuário, garantindo o total funcionamento da aplicação que, posteriormente, poderá ser enviada para avaliação e validação pela *Microsoft* e adicionada a *Windows Store*.

## **4 DESENVOLVIMENTO**

### **4.1 Tecnologias Utilizadas**

#### **4.1.1 HTML Agility Pack**

Segundo Harris (2009), o *HTML Agility Pack* é uma biblioteca de código aberto que garante, facilmente, a raspagem de dados de um HTML. Harris (2009) acrescenta que a raspagem de uma tela é o termo para quando se realiza o *download* de uma página *web* e, então, analisa-se seu HTML para extrair informações do mesmo.

#### **4.1.2 XPath – XML Path Language**

O *XPath* é um conjunto de regras com a finalidade de fornecer uma sintaxe e semântica comum para a funcionalidade entre *XSL Transformations* (XSLT) – uma linguagem para folhas de estilo de transformações – e *XPointer* que é, por sua vez, lida com aspectos estruturais do XML (CLARK; DEROSE, 1999).

Com isso, pode-se usar essa expressão *XPath* para realização de pesquisas sobre documentos XML, uma vez que ela provê a possibilidade de seleção e filtragem de dados contidos em um ou mais documentos, fornecendo uma sintaxe bem intuitiva, baseada em caminhos, como em um sistema de arquivos (VAZQUEZ, [200-?]).

#### **4.1.3 Entity Framework**

A modelagem Entidade-Relacionamento é um modelo abstrato que descreve, de forma conceitual, os dados a serem utilizados em um sistema, onde a principal ferramenta é a representação gráfica através do diagrama Entidade-Relacionamento. Com o ADO.NET *Entity Framework*, diz Lerman (2007), a *Microsoft* criou um modelo Entidade-Relacionamento executável, que permite o mapeamento de tabelas em classes (ORM - *Object Relational Mapping*) e são geradas a partir do esquema conceitual de um Banco de Dados. Assim, o *Framework* cuida de todas as traduções entre extrair dados do banco e enviá-los de volta para ele.



#### 4.1.4 JSON - JavaScript Object Notation

*JavaScript Object Notation* (JSON) é um formato de troca de dados baseado em texto que, como o XML, é legível, independente de qualquer plataforma e possibilita uma gama enorme de implementações.

Aziz e Scott (2007) reforçam que os dados formatados com o padrão JSON são mais leves e podem ser analisados, facilmente, por implementações em *JavaScript*, tornando seu uso ideal com AJAX, uma vez que o formato é mais rápido e fácil do que o XML. Porém, é importante destacar que JSON não é limitado somente a esse uso, uma vez que pode ser aplicado em qualquer situação onde haja a troca ou armazenamento de informações como texto.

#### 4.1.5 Web API – REST

De forma bem clara, Navarro ([200-?]), conceitua *Web API* como um *framework* utilizado para desenvolver serviços HTTP sobre o *.NET Framework*, onde esses serviços são expostos através do HTTP e acessados e consumidos de forma direta por uma diversidade de dispositivos, como *browsers* e equipamentos *mobile*.

REST vem de *REpresentational State Transfer* (ou Transferência de Estado Representativo) e define-se como um estilo de desenvolvimento de *Web Services*. Esse protocolo é guiado pelo uso adequado de métodos HTTP, de URLs (*Universal Resource Locator*), pelo uso de códigos de *status* para representação de sucesso ou falha, uso adequado de cabeçalhos HTTP e interligações entre vários recursos diferentes (SAUDATE, 2013).

#### 4.2 Introdução ao desenvolvimento

No início da pesquisa, a ideia era desenvolver um aplicativo do *Windows 8* sobre a nova plataforma de desenvolvimento da *Microsoft WinRT*. Mas surgiram as seguintes perguntas: Qual a finalidade desse aplicativo? O que ele se propõe a fazer? Então, definiu-se que o aplicativo faria uma amostragem de notícias dos principais portais da *Internet* brasileira. Com esse tema proposto, surgiram novas perguntas: Como fazer isso? As notícias serão armazenadas? Cada aplicativo baixado da *Windows Store* terá um Banco de Dados?

Com as devidas orientações, descobriu-se a técnica *Data Scrapping*, no português, Raspagem de Dados. Agora com as informações raspadas, foi preciso determinar o armazenamento dessas informações. Com a dúvida de que cada aplicativo iria possuir um

Banco de Dados, descobriu-se que cada um desses aplicativos teria que fazer a raspagem de dados para que as informações fossem armazenadas, porém, observou-se que essa modelagem de sistema possui um desempenho extremamente ruim, ou seja, quando o usuário começar a usar o aplicativo, o mesmo terá que se conectar em todos os portais de notícias propostos, fazer a raspagem das principais notícias e armazenar essas informações. Para a resolução desse problema de desempenho, foi pensado que o sistema teria que possui um Banco de Dados centralizado, ou seja, um Banco de Dados hospedado em um servidor onde todos os aplicativos baixados da *Windows Store* se conectassem no mesmo. Mas surgiram mais perguntas:

Como fazer esses aplicativos se conectarem nesse Banco de Dados centralizado? E mais, agora que o sistema só possui um Banco de Dados, como fazer a raspagem de dados se a técnica não está mais nos aplicativos?

Com base em estudos, descobriu a técnica de serviço exposto para resolver esse problema de centralização. Essa técnica consiste em um serviço que fica rodando em um servidor para que diversos dispositivos possam consumi-lo, ou seja, fazer a integração do aplicativo com o repositório de notícias. A princípio, optou-se, como técnica de serviço exposto, a *WebAPI*, que é um *framework* desenvolvido também pela empresa *Microsoft* baseado na arquitetura REST, ou seja, agora os aplicativos podem se conectar na *WebAPI* via protocolo HTTP realizando uma operação *GET* e receber como resposta um JSON representando a lista de notícias.

Em relação a como iria ser feita a raspagem de dados com essa modelagem diferente, foi proposto que fosse desenvolvido um aplicativo comum do *Windows* para que o mesmo ficasse de tempos em tempos no servidor da *API* raspando as informações necessárias. Para que o desenvolvimento fosse ágil, foi proposto que utilizasse um *framework ORM* para acesso aos dados tanto para o aplicativo de raspagem quanto para *API*. A tecnologia escolhida foi a *Entity Framework* que também faz parte do pacote .NET.

A FIG. 8 ilustra a modelagem final proposta para o sistema.

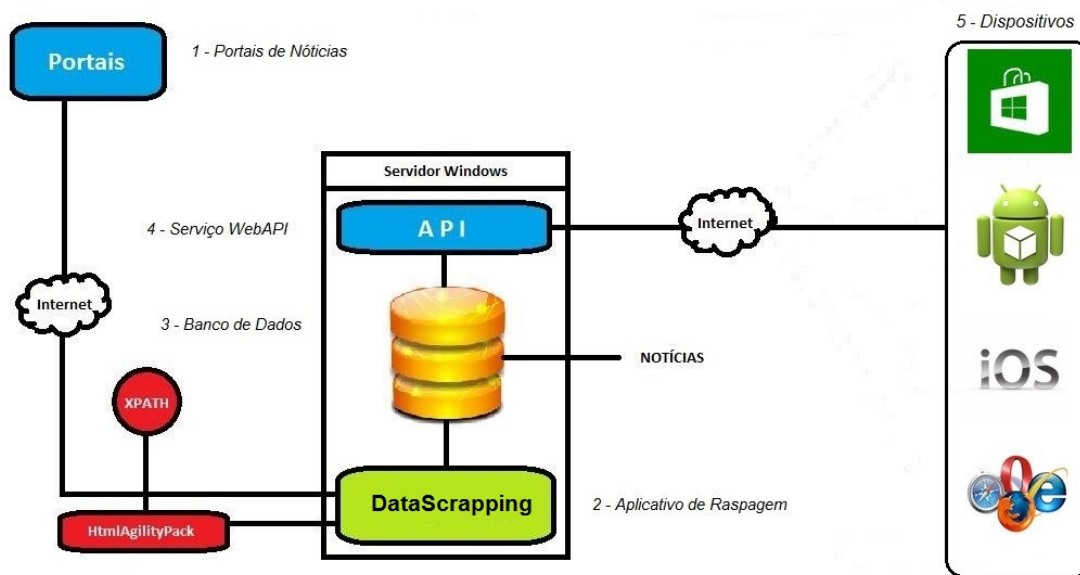


FIGURA 8 – Modelagem do funcionamento do sistema

Fonte: os próprios autores do trabalho

#### As legendas FIGURA 8 são detalhadas a seguir:

1. Portais de notícias: esse item especifica os portais de notícias que se encontram na *Internet* e de onde são extraídas as informações para o consumo dos dispositivos.
2. Aplicativo de Raspagem: esse item especifica o aplicativo ou programa que se encontra no servidor do repositório de notícias que se encarrega de se conectar aos portais de notícias e utilizar a técnica de *DataScraping* com auxílio do framework *HtmlAgilityPack* para extração das informações de tempos em tempos.
3. Banco de Dados: esse item representa o repositório onde as informações são gravadas.
4. Serviço *WebAPI*: esse item representa o serviço exposto que fica rodando no servidor para que diversos dispositivos de diferentes plataformas possam se conectar e consumir as informações.
5. Dispositivos: esse item representa os diversos dispositivos que podem consumir as informações através da *API*. A princípio, no trabalho desenvolveu um aplicativo *Windows Store* para o funcionamento do sistema.

## 5 RESULTADOS E DISCUSSÕES

Além do aplicativo de raspagem e da API de notícias, esse trabalho produziu outro produto final extremamente funcional para os usuários finais, uma aplicação para o *Windows 8* que consome a API de notícias. Como todo aplicativo do *Windows 8*, o aplicativo construído no trabalho, chamado de *scanews*, possui uma *splashscreen* como interface inicial. A *splashscreen* é a interface inicialização e o tema do aplicativo.

O *Scanews* traz uma página inicial interativa que lista os portais de notícias scaneados, a partir dos quais as notícias mais lidas, que está no ar naquele instante, podem ser lidas.

Depois que o usuário entrou na página inicial do aplicativo, ele pode clicar em qualquer portal e filtrar as categorias do mesmo na próxima página do aplicativo. As categorias do portal referenciado se encontram na lateral esquerda da página e as respectivas notícias da categoria referenciada se encontram na lateral direita da página.

Agora que o usuário conseguiu filtrar as categorias do portal de notícia escolhido, chegou o momento de visualizar a notícia que deseja. Quando o usuário clica em uma notícia, o mesmo é direcionado para outra página que contém um componente que simula um *browser*. O componente recebe a URL da notícia clicada para fazer a simulação do *browser*.

Pode-se observar que cada página do aplicativo possui um botão com a funcionalidade de voltar para a página anterior. O aplicativo ficou bem simples, porém bastante funcional centralizando os melhores conteúdos de todos os portais no ambiente do *Windows 8*.

## 6 CONCLUSÃO

Tendo em vista os aspectos observados e estudados a respeito do Sistema Operacional *Windows 8*, como sua arquitetura, suas novidades e plataforma de desenvolvimento, concluí-se que ele realmente traz uma experiência inovadora para seus usuários quando o assunto é aplicativos integrados. Essa foi a maior motivação para o desenvolvimento deste trabalho, a inovação em aplicativos.

Portanto, somos levados a acreditar que concluímos o objetivos do nosso trabalho, apresentando os principais pontos do Sistema Operacional *Windows 8*, principalmente em relação ao desenvolvimento para *Windows 8*, desenvolvendo uma aplicação funcional e com a possibilidade de distribuição, assim como era desejado, além de deixar um trabalho possível de ser continuado, através da integração de novos dispositivos de diferentes plataformas.

## REFERÊNCIAS

HONEYCUTT, JERRY. *Introducing Windows 8: An Overview for IT Professionals*. Redmond: Microsoft Press, 2012. 168p.

ABREU, LUÍS. **Desenvolvimento em Windows 8 - Curso Completo**. São Paulo: FCA, 2013. 660p.

MORRISON, MICHAEL. **Use a Cabeça Javascript**. São Paulo: Alta Books, 2008. 640p.

BATISTA, Thiago. **Windows 8: Suas versões e diferenças**. abr. 2013. Disponível em: <<http://gestaoemti.com.br/index.php/windows-8-suas-versoes-e-diferencas/>>. Acesso em: 15 mai. 2013

VASQUE, Verônica. **Windows 8: Confirma as novidades apresentadas pela Microsoft**. fev. 2012. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2012/02/windows-8-confira-todas-novidades-apresentadas-pela-microsoft.html>>. Acesso em: 20 mai. 2013

GUGELMIN, Felipe. **Microsoft explica a transição do Windows 8 para a arquitetura ARM**. fev. 2012. Disponível em: <<http://www.tecmundo.com.br/windows-8/19290-microsoft-explica-a-transicao-do-windows-8-para-a-arquitetura-arm.htm>>. Acesso em: 22 mai. 2013

APRIMORAMENTOS de **segurança do Windows 8**. fev. [200?]. Disponível em: <<http://technet.microsoft.com/pt-br/windows/jj983723.aspx>>. Acesso em: 15 abr. 2013

LEBLOND, Antoine. **A Windows Store para o Release Preview**. jun. 2012. Disponível em: <[http://blogs.msdn.com/b/windowsstore\\_br/archive/2012/06/12/a-windows-store-para-o-release-preview.aspx](http://blogs.msdn.com/b/windowsstore_br/archive/2012/06/12/a-windows-store-para-o-release-preview.aspx)>. Acesso em: 20 marc. 2013

VELOSO, Thássius. **Windows Store será mais completa do que você poderia imaginar**. dez. 2012. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2011/05/windows-store-sera-mais-completa-do-que-voce-poderia-imaginar.html>>. Acesso em: 5 abr. 2013

FERREIRA, Élcio; EIS, Diego. **Curso HTML5 - W3C Escritório Brasil**. nov. 2010. Disponível em: <<http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>>. Acesso em: 12 mar. 2013

MACORATTI, José Carlos. **NET - Introdução a XAML (*Extensible Application Markup Language*)**. set. 2003. Disponível em: <[http://www.macoratti.net/09/03/net\\_xaml.htm](http://www.macoratti.net/09/03/net_xaml.htm)>. Acesso em: 15 jun. 2013.

SOUZA, Vinícius. **Controles básicos do XAML**. jul. 2012. Disponível em: <http://channel9.msdn.com/posts/Controles-bsicos-do-XAML>. Acesso em: 15 jun. 2013

ARRIGONI, Ricardo. **Principais Frameworks de Javascript**. mai. 2013. Disponível em: <<http://www.linhadecodigo.com.br/artigo/3637/principais-frameworks-de-javascript.aspx>>. Acesso em: 21 jun. 2013

CHUMA, Euclides Lourenço. **Introdução a Javascript**. Disponível em: <<http://www.devmedia.com.br/introducao-curso-de-javascript-completo-aula-1/28287#ixzz2WHoN7bvl>>. Acesso em: 20 mar. 2013

ALLEN, Jonathan. **WinRT em detalhes: a nova API OO do Windows 8 que substituirá o Win32**. Set. 2011. Disponível em: <<http://www.infoq.com/br/news/2011/09/Windows8-WinRT>>. Acesso em: 20 jun. 2013

- SONNINO, Bruno, SONNINO, Roberto. **Introdução à WPF**. out. 2006. Disponível em: <<http://msdn.microsoft.com/pt-br/library/cc564903.aspx>>. Acesso em: 20 jun. 2013
- ALLEN, Jonathan. **WPF vs. Silverlight – Qual é a real diferença?** nov. 2009. Disponível em: <<http://www.infoq.com/br/news/2009/11/WPF-Silverlight>>. Acesso em: 15 jun. 2013
- HARRIS, Matthew. **Introdução ao HTML Agility Pack**. set, 2009. Disponível em: <<http://runtingsproper.blogspot.com.br/2009/09/introduction-to-htmlagilitypack-library.html>>. Acesso em: 21 out. 2013
- CLARK, James; DEROSE, Steve. **XML Path Language (XPath)**. nov. 1999. Disponível em: <<http://www.w3.org/TR/xpath/#section-Introduction>>. Acesso em: 21 out. 2013
- LERMAN, Julia. **Introducing ADO.NET Entity Framework**. nov. 2007. Disponível em: <<http://www.code-magazine.com/article.aspx?quickid=0711051>>. Acesso em: 21 out. 2013
- AZIZ, Atif; SCOTT, Mitchell. **An Introduction to JavaScript Object Notation (JSON) in JavaScript and .NET**. fev. 2007. Disponível em: <<http://msdn.microsoft.com/en-us/library/bb299886.aspx>>. Acesso em: 29 out. 2013
- NAVARRO, Fábio. **Introdução ao ASP.NET Web API**. Disponível em: <<http://www.devmedia.com.br/introducao-ao-asp-net-web-api/25180>>. Acesso em: 12 nov. 2013
- CIRIACO, Douglas. **O que é API?**. mar. 2009. Disponível em: <<http://www.tecmundo.com.br/programacao/1807-o-que-e-api-.htm>>. Acesso em: 12 nov. 2013
- TILKOV, Stefan. **Uma rápida Introdução ao REST**. out. 2008. Disponível em: <<http://www.infoq.com/br/articles/rest-introduction>>. Acesso em: 12 nov. 2013
- SAUDATE, Alexandre. **REST - Construa API's inteligentes de maneira simples**. São Paulo: Casa do Código, 2013. 101p.