

Universidade Aberta
Departamento de Ciências e Tecnologia

Aplicações Web Para Qualquer Dispositivo

-

(Web Applications For Any Device)

Licenciatura em Informática

Xiquinho F Silva – 905444

Orientador do Projecto:

**Prof. Vitor José Crêspo Cardoso,
Departamento de Ciências e Tecnologia (DCeT),
Universidade Aberta**

ÍNDICE

1	Objectivo do documento	7
2	Objectivo do projecto	8
3	Desenvolvimento	9
3.1	Configuração.....	9
3.1.1	Microsoft Azure	10
3.1.2	Windows Server 2012	10
3.1.3	IIS 8.0	11
3.1.4	Visual Studio Online	12
3.2	Framework, Tecnologias e Ferramentas	13
3.2.1	Visual Studio 2012	14
3.2.2	.NET Framework 4.5	14
3.2.3	Compact.NET Framework 4.5.....	14
3.2.4	ASP.NET	14
3.2.5	Síntaxe Razor	16
3.2.6	Ajax	17
3.2.7	JavaScript.....	17
3.2.8	HTML5.....	17
3.2.9	Less	18
3.2.10	JQuery.....	18
3.2.11	Bootstrap.....	18
3.2.12	MS SQL Server 2012	19
3.2.13	SQLyte.....	19
3.2.14	Lucene	20
3.3	Considerações de Design.....	21
3.3.1	Tipo de projecto	22
3.3.2	Navegação ASP.NET.....	23

3.3.3	Acesso a dados	25
3.3.4	Aparência e <i>design</i> consistente.....	27
3.3.5	Funcionalidades AJAX.....	29
3.3.6	Funcionalidades de gerenciamento do estado da aplicação.....	31
3.3.7	Armazenamento em memória (cache)	32
3.3.8	Segurança	34
3.3.9	Acessibilidade	35
3.3.10	Globalização e Localização.....	37
3.3.11	Compilação.....	38
3.4	Web Design Responsivo	40
3.4.1	Layout flexível baseado em grelha fluída.....	40
3.4.2	Imagens e Media flexíveis	42
3.4.3	Media queries, um módulo da especificação CCS3	44
3.4.4	Tempo de espera do carregamento da página	45
3.4.5	Redução de pedidos	46
3.4.6	Redução do tamanho da página	47
3.4.7	Design em função da velocidade	51
3.4.8	Versão Desktop.....	56
3.4.9	Versão Tablet.....	57
3.4.10	Versão Mobile	58
3.5	Módulo my books	59
3.5.1	Classe mybooks	66
3.6	Módulo my movies	69
3.6.1	Classe mymovies	73
3.7	Módulo my pics.....	76
3.7.1	Classe mypics.....	85
3.8	Módulo my travels.....	87

3.8.1	Classe mytravels	97
3.9	Módulo my shop	99
3.9.1	Sistemas de Recomendação Web	102
3.9.2	Sistema de Recomendação da Amazon.com	103
3.9.3	Classe myshop	104
3.10	Classes Comuns	110
4	Conclusão	119
5	Referências Bibliográficas	120

1 OBJECTIVO DO DOCUMENTO

Este documento descreve detalhadamente o trabalho desenvolvido para a disciplina de Projecto Final do curso de Licenciatura em Informática na Universidade Aberta. Assimilados os conceitos e a filosofia, tornou-se a dada altura imperativo pensar num caso de estudo que permitisse pôr em prática conhecimentos adquiridos. Este documento condensa um conjunto desses conhecimentos e justifica as abordagens seguidas na sua implementação.

2 OBJECTIVO DO PROJECTO

Ao longo do percurso académico pretende-se que o aluno adquira diversas competências que se reflectem no desenvolvimento de um projecto final à sua escolha. Escolha esta que foi previamente construída e desenvolvida em colaboração com o Professor orientador, culminando no objectivo de construir uma aplicação web baseada em modelos como *HTML5*, *CSS3* e outras tecnologias *web* modernas. Sem quaisquer conversões especiais ou reprogramação, uma aplicação *web* para qualquer dispositivo pode ser executada em praticamente qualquer plataforma munida de um *browser* moderno, compatível com as normas. Uma vez que a aplicação web é invocada, os utilizadores utilizando *desktops*, *tablets*, *iPads*, *iPhones*, *WindowsPhone*, *Android*, *Kindle*, etc. podem correr a mesma aplicação e executá-la como em qualquer outra plataforma.

O objectivo principal deste projeto é a implementação de uma aplicação web (e dos seus diversos módulos) visualizável em qualquer dispositivo moderno.

É igualmente importante mencionar outros objectivos secundários que estiveram também presentes e que são:

- Implementação e configuração do servidor *Web* em *Windows Azure*;
- Aprofundamento de competências de programação em *.Net Framework*, especialmente em *ASP.NET Web Pages*;
- Realização de testes aos módulos desenvolvidos;
- Aprendizagem da capacidade de redacção de relatórios.

Desenvolver um *design* que pretende integrar diversos tipos de dispositivos conciliando diversas formas de utilização não é algo trivial nem pode ser descurado. A aplicação deve adaptar-se ao utilizador, nunca o contrário.

A aplicação *web* é constituída por diversos módulos que os utilizadores podem activar facultativamente. Neste momento encontram-se já desenvolvidos alguns módulos que reflectem as preferências do aluno e os seus *hobbies* favoritos: livros, filmes, fotografias, viagens, compras. Posteriormente poderão ser desenvolvidos outros módulos destinados a inventariar outros *hobbies*, por exemplo: receitas culinárias, filatelia, numismática, etc. assim como um módulo destinado a administração/configuração da aplicação.

3 DESENVOLVIMENTO

3.1 CONFIGURAÇÃO

A aplicação *web* foi desenvolvida com recurso a tecnologias modernas Web (Asp.Net/Ajax/Java/HTML5/JQuery, etc.), a *software* livre (ex., *Calibre - e-book manager*, *My Movies – movie manager*), proprietário (ex, *SQL Server*) e também desenvolvido pelo estudante. (VB, C# e Java).

A aplicação encontra-se numa *virtual machine*, alojada em nuvem na *Microsoft Azure*, sistema operativo *Windows Server 2012 R2*, correndo num servidor HTTP IIS 8.5. (Url para demo: <http://myweb.simple-url.com/>)

EXCERTO DO CÓDIGO XML UTILIZADO PARA ESTABELECEER AS DEFINIÇÕES GERAIS:

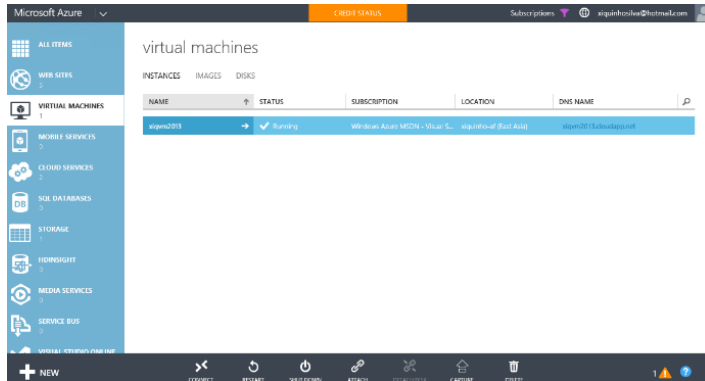
```
<?xml version="1.0" encoding="utf-8"?>
<settings>
  <enablehttpcompression>True</enablehttpcompression>
  <enablepasswordreset>True</enablepasswordreset>
  <enableerrorlogging>True</enableerrorlogging>
  <culture>Auto</culture>
  <smtpserverport>25</smtpserverport>
  <timezone>0</timezone>
  <enableoptimization>True</enableoptimization>
```

EXCERTO DO FICHEIRO WEB.CONFIG UTILIZADO PARA CONFIGURAR A APLICAÇÃO WEB:

```
<system.web>
  <compilation debug="true" targetFramework="4.5">
    <assemblies>
      <add assembly="System.Net.Http, Version=4.0.0.0, Culture=neutral, PublicKeyToken=B03F5F7F11D50A3A"/>
      <add assembly="System.Net.Http.WebRequest, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B03F5F7F11D50A3A"/>
    </assemblies>
    <codeSubDirectories>
      <add directoryName="CS"/>
    </codeSubDirectories>
  </compilation>
  <httpRuntime targetFramework="4.5"/>
  <customErrors defaultRedirect="/error/" mode="Off"/>
</system.web>
```

3.1.1 Microsoft Azure

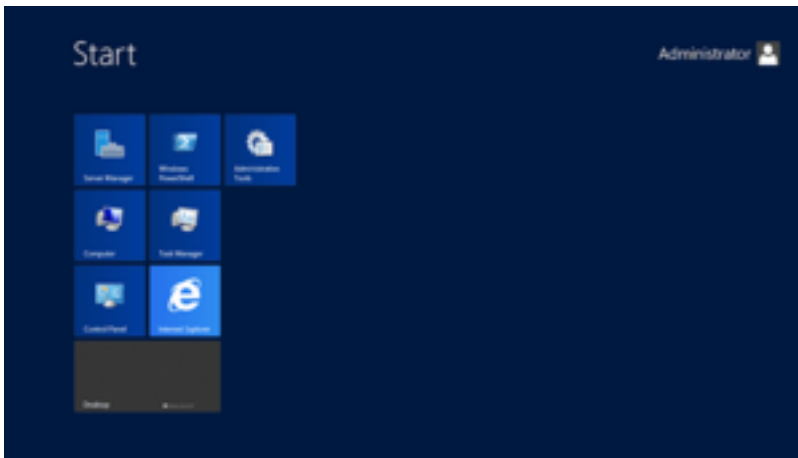
A Microsoft Azure (antes de 25 de março de 2014, também chamada Windows Azure) é uma



plataforma de computação em nuvem e infraestrutura, criada pela Microsoft, para a construção, implementação e gerenciamento de aplicações e serviços através de uma rede global de *datacenters* supervisionados pela Microsoft. Ela fornece plataformas e infraestruturas como serviços e suporta várias linguagens de programação, ferramentas e

estruturas, incluindo *software* específico da Microsoft assim como *software* e sistemas de terceiros. (<http://azure.microsoft.com/en-us/>)

3.1.2 Windows Server 2012



O *Windows Server 2012* é a sexta versão do *Windows Server*. É a versão de servidor do *Windows 8* lançada em Setembro de 2012, que sucede ao *Windows Server 2008 R2*.

Ao contrário de seu antecessor, o *Windows Server 2012* não tem suporte para computadores baseados em Itanium, e tem quatro edições. Vários

recursos foram adicionados ou melhorados em relação ao *Windows Servidor 2008 R2* (com muitos colocando ênfase na computação em nuvem), como uma versão actualizada do *Hyper-V*, uma função de gerenciamento de endereços *IP*, uma nova versão do *Windows Task Manager*, e um novo sistema de arquivos. O *Windows Server 2012* recebeu na generalidade boas críticas, apesar de ter incluído a mesma GUI controversa baseada em *Metro* utilizada no *Windows 8* versão cliente. (<http://www.microsoft.com/en-us/server-cloud/products/windows-server-2012-r2>)

3.1.3 IIS 8.0



O *Internet Information Services (IIS)* é um servidor *Hypertext Transfer Protocol (HTTP)*, *HyperText Transfer Protocol Secure (HTTPS)*, *File Transfer Protocol (FTP)*, *Simple Mail Transfer Protocol (SMTP)* e *Negotiable Mail Transfer Protocol (NMTP)* que corre em *Windows*. Foi utilizado no projecto como servidor de *HTTP* de modo a disponibilizar a aplicação criada *online*.

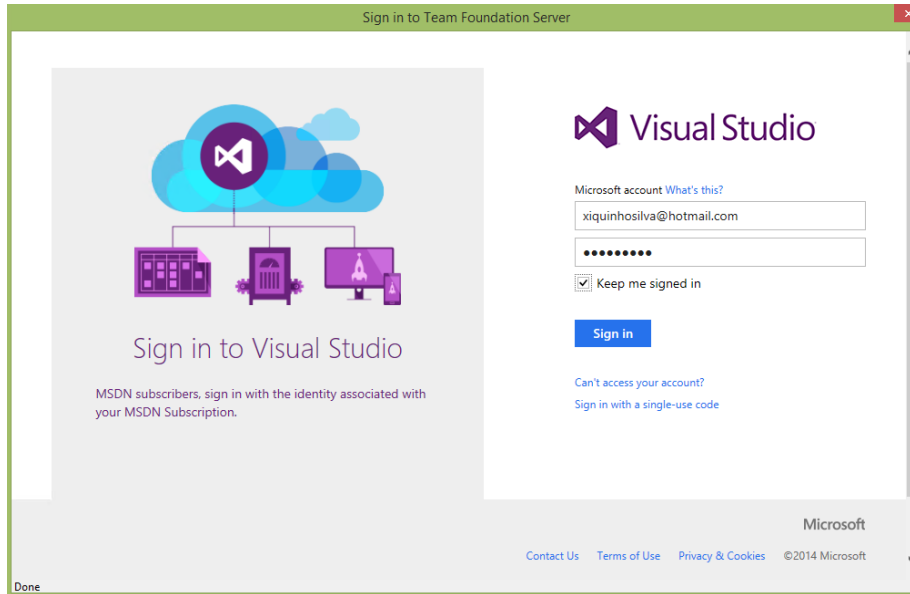
O *IIS* tem sido uma parte integrante da família *Windows NT* desde o *Windows NT 4.0*, embora tenha estado ausente de algumas edições (por exemplo, *Windows XP Home Edition*). O *IIS* não é activado por defeito quando o *Windows* é instalado. O gerenciador do *IIS* é acedido através da *Microsoft Management Console* ou em *Ferramentas Administrativas* no *Painel de Controlo*.

O *IIS 8.0* só está disponível no *Windows Server 2012* e *Windows 8*. O *IIS 8.0* inclui aplicativos de inicialização, suporte centralizado de certificados *SSL* e dimensionamento *multicore* em *hardware NUMA*, entre outras novidades.

IIS 8.5 está incluído no *Windows Server 2012 R2* e *Windows 8.1*. Esta versão inclui novas funcionalidades tais como, *Idle worker-Process page-out*, *Dynamic Site Activation*, *Enhanced Logging*, *ETW logging*, and *Automatic Certificate Rebind*. (<http://www.iis.net>)

3.1.4 Visual Studio Online

O *Visual Studio Online*, é um serviço online baseado no *Team Foundation Server* com serviços adicionais em nuvem, dedicado ao desenvolvimento de projectos. Aproveitando a infraestrutura da *Microsoft* permite conectar o *Visual Studio* a uma variedade de plataformas colaborativas de centralização de *source control*, tais como, Xcode, Eclipse e outros clientes do GitHub. (<http://www.visualstudio.com/>)



3.2 FRAMEWORK, TECNOLOGIAS E FERRAMENTAS

A programação web, dada a sua natureza, permanece em desenvolvimento contínuo. Existem nos meios académicos e comerciais algumas frameworks desenvolvidas e outras em desenvolvimento. Todavia a abordagem adoptada no desenvolvimento da aplicação foi orientada usando, sempre que possível, ferramentas da *Microsoft*.

A programação de aplicações web apresenta desafios que normalmente não surgem ao programar aplicações baseadas em clientes tradicionais. Entre alguns dos desafios temos:

- **Implementação de uma interface atractiva** - Pode ser difícil e trabalhoso projectar e implementar uma interface para utilizador usando recursos HTML básicos, especialmente se a página tiver um *layout* complexo, uma grande quantidade de conteúdo dinâmico e objectos interactivos.
- **Separação de cliente e servidor** - Numa aplicação web, o cliente (*browser*) e o servidor são programas diferentes geralmente executados em diferentes computadores (e até mesmo em diferentes sistemas operativos). Consequentemente, as duas metades da aplicação compartilham muito pouca informação; podem comunicar entre si, mas normalmente trocam apenas pequenos pedaços de informação simples.
- **Execução *Stateless*** - Quando um servidor Web recebe uma solicitação para uma página, encontra a página, processa-a, envia-a para o *browser* e, de seguida, descarta todas as informações dessa página. Se o utilizador solicita a mesma página novamente, o servidor repete toda a sequência, reprocessando a página a partir do zero. Dito de outra forma, um servidor não tem memória de páginas que processou. As páginas processadas são *stateless*. Ou seja, se uma aplicação web precisa manter informações sobre uma página, a sua natureza *stateless* pode constituir um problema.
- **Recursos de cliente desconhecidos** - Em muitos casos, as aplicações Web são acessíveis a muitos utilizadores que utilizam diferentes *browsers*. *Browsers* diferentes que possuem capacidades diferentes, o que torna difícil criar uma aplicação web que seja executada de forma perfeita em todos eles.
- **Dificuldades com acesso aos dados** – Operações de leitura e gravação em aplicações web tradicionais podem ser complexas e obrigar à utilização intensiva de recursos.
- **Escalabilidade** - Em muitos casos, as aplicações web projectadas com os métodos existentes não conseguem atingir as metas de escalabilidade devido à falta de

compatibilidade entre os vários componentes da aplicação, constituindo um ponto de falha para aplicações sujeitas a um ciclo de crescimento forte.

A aplicação web utiliza múltiplas técnicas e tecnologias mas nem todas estão aqui descritas. De seguida são apresentadas muito brevemente algumas das tecnologias que estão de uma ou outra forma integradas na aplicação. (MSDN, 2014)

3.2.1 Visual Studio 2012

O *Visual Studio 2012 (VS2012)* é um *Integrated Development Environment (IDE)* da *Microsoft* que permite desenvolver aplicações usando um leque interessante de linguagens para diversas situações. O *VS2012* permite criar bibliotecas, aplicações para consola, web ou gráficas para *Windows* e *Windows Phone*. Através deste *IDE* foi possível utilizar e integrar linguagens como o *VB*, *C#* e *Java* no desenvolvimento da aplicação. (<http://www.visualstudio.com/>)

3.2.2 .NET Framework 4.5

A *.NET Framework* é uma componente integrante dos sistemas operativos da *Microsoft*. Esta é talvez um dos maiores pontos favoráveis ao desenvolvimento de aplicações em *Windows*, uma vez que aglomera um conjunto enorme de bibliotecas. Para além disso, a *framework* é expansível e pode ser usada em múltiplas linguagens flexibilizando a programação.

(<http://www.microsoft.com/net>)

3.2.3 Compact.NET Framework 4.5

A *Compact .NET Framework* é a versão *.NET Framework* da *Microsoft* para dispositivos móveis e embebidos. Apesar de serem muito semelhantes é preciso ter em atenção que nem todas as funcionalidades da *.NET Framework* existem nesta versão.

3.2.4 ASP.NET

ASP.NET é a plataforma da *Microsoft* para o desenvolvimento de aplicações web. É a sucessora da tecnologia ASP, um componente do *IIS* que permite através de uma linguagem de programação integrada na *.NET Framework* criar páginas dinâmicas. É baseado na *.NET Framework* herdando todas as suas características, por isso, como qualquer aplicação *.NET*, as aplicações para essa *framework* podem ser escritas em várias linguagens, tais como *C#* e *Visual Basic*.

Uma aplicação para web desenvolvida em ASP.NET pode reutilizar código de qualquer outro projecto escrito para a .NET Framework, mesmo que em linguagem diferente. Uma página ASP.NET escrita em VB.NET pode chamar componentes escritos em C# ou Web Services escritos em C++, por exemplo. Ao contrário da tecnologia ASP, as aplicações ASP.NET são compiladas antes da execução, trazendo sensível ganho de desempenho.

(<http://www.asp.net>)

Nos *websites* projectados em ASP.NET o código reside num servidor Web e as páginas são compiladas automaticamente a primeira vez que um utilizador navega para qualquer página do site. Opcionalmente, o *website* também pode ser pré-compilado para que não haja nenhum atraso na compilação na primeira vez que um utilizador acede a uma página. Para projectos de aplicações Web ASP.NET as páginas são compiladas antes de serem enviadas ao servidor.

Enfrentar os desafios da programação de aplicações web já referidos na introdução pode exigir tempo e esforço substancial. Vejamos agora como as Web Pages ASP.NET e *.NET framework* nos ajudam a superar esses desafios:

- **Modelo consistente e intuitivo** - A framework das páginas ASP.NET apresenta um modelo de objecto que permite pensar na sua estrutura como uma unidade, não como peças separadas cliente/servidor. Neste modelo, a programação pode ser feita de forma mais intuitiva do que em aplicações Web tradicionais, incluindo a capacidade de definir propriedades para os elementos da página e de responder a eventos.
- **Modelo de programação orientado a eventos** - A framework traz para as aplicações Web o modelo familiar de programar *handlers* de eventos que ocorrem no cliente ou no servidor. A framework abstrai-se deste modelo de tal forma que o mecanismo subjacente de capturar um evento no cliente é automático e invisível, transmitindo-o para o servidor e chamando o método apropriado. O resultado é uma estrutura de código claro e facilmente escrito que suporta a programação orientada a eventos.
- **Gerenciamento de estado intuitivo** - A framework encarrega-se da tarefa de manter o estado da página e dos seus componentes automaticamente, e fornece formas explícitas para manter o estado de informações específicas da aplicação. Tal é atingido sem uso intensivo de recursos do servidor e pode ser implementado recorrendo ou não à utilização de *cookies*.
- **Aplicações independentes do browser** - A framework permite criar toda a lógica da aplicação no servidor, eliminando a necessidade de programar explicitamente para as

diferenças dos diversos browsers. No entanto, é possível tirar proveito de recursos específicos de um determinado *browser* programando o lado do cliente para fornecer um melhor desempenho e uma melhor experiência.

- **Common Language Runtime** - Em virtude de ter sido construída no topo da *.NET Framework*, permite que todas as funcionalidades da framework estejam disponíveis para as aplicações web. Estas podem ser escritas em qualquer linguagem que seja compatível com a CLR. Além disso, o acesso aos dados é simplificado, incluindo a utilização da infra-estrutura de acesso a dados fornecidos pela *.NET Framework*, tais como o *ADO.NET*.
- **Escalabilidade** - A *framework* das páginas ASP.NET permite escalar a aplicação web de um computador com um único processador para uma *WebFarm* de forma clara e sem alterações complexas à lógica da aplicação. (MSDN, 2014)

3.2.5 Síntaxe Razor

As ASP.NET *Web Pages* funcionam como uma interface programável entre o utilizador e a aplicação web, apresentando as informações no *browser* segundo a lógica implementada no servidor. A programação do interface é dividida em duas partes a componente visual e a componente lógica.

O elemento visual consiste num ficheiro que contém HTML ou *ASP.NET server controls* ou ambos. A ASP.NET *Web Page* funciona como um recipiente para o texto estático e os *ASP.NET server controls* a exibir.

A lógica das ASP.NET *Web Pages* consiste num código criado para interagir com a página. O código pode residir num bloco de script na página ou numa classe separada. Se o código está num arquivo de classe separado, este arquivo é conhecido como arquivo *code-behind*. O código no arquivo *code-behind* pode ser escrito em *Visual Basic*, *C#*, ou qualquer outra linguagem suportada pela *.NET Framework*.

Na presente aplicação web foi utilizada a sintaxe *Razor*. *Razor* é uma sintaxe de programação ASP.NET usado para criar páginas web dinâmicas utilizando linguagens de programação tais como o *C#* ou *Visual Basic*. A sintaxe *Razor* é um mecanismo de exibição simples, muito recente e foi lançada para o *Visual Studio 2010*, em 2011 como parte do ASP.NET *MVC3* e do conjunto de ferramentas do *WebMatrix*. (<http://www.asp.net/web-pages>)

3.2.6 Ajax

AJAX (*Asynchronous Javascript and XML*) é o uso metodológico de tecnologias como Javascript e XML em *browsers*, para tornar páginas web mais interactivas com o utilizador, através de solicitações de informações assíncronas. Apesar do nome, a utilização de XML não é obrigatória (JSON é frequentemente utilizado) e as solicitações não necessitam ser assíncronas. Ajax não é uma tecnologia única, mas um grupo de tecnologias. HTML e CSS podem ser utilizados em combinação para formatar e estilizar informações. O acesso ao DOM é feito através de JavaScript para exibir dinamicamente - e permitir ao utilizador interagir com - as informações apresentadas. JavaScript e o objecto XMLHttpRequest fornecem um método para troca de dados de forma assíncrona entre o *browser* e o servidor para evitar o *refresh* da página na sua totalidade.

3.2.7 JavaScript

JavaScript é uma linguagem de programação interpretada. Foi originalmente implementada como parte dos *browsers* web para que scripts pudessem ser executados do lado do cliente e interagissem com o utilizador sem a necessidade do script passar pelo servidor, controlando o browser, realizando comunicação assíncrona e alterando o conteúdo do documento exibido. É actualmente a principal linguagem para programação *client-side* em *browsers* web. Foi concebida para ser uma linguagem script com orientação a objectos baseada em protótipos, tipagem fraca e funções de primeira classe. Possui suporte à programação funcional e apresenta recursos como fechamentos e funções de alta ordem normalmente indisponíveis em linguagens populares como Java e C++. (Mozilla Developer Network, 2014)

3.2.8 HTML5

HTML5 (*Hypertext Markup Language*, versão 5) é uma linguagem para estruturação e apresentação de conteúdo para a Web. É a quinta versão da linguagem HTML. Esta nova versão traz consigo importantes mudanças quanto ao papel do HTML no mundo da Web, através de novas funcionalidades como semântica e acessibilidade. Com novos recursos, antes só possíveis por meio de outras tecnologias. A sua essência tem sido melhorar a linguagem com o suporte para as mais recentes funcionalidades de multimédia, enquanto a mantém facilmente legível por seres humanos e consistentemente compreendida por computadores e outros dispositivos (*browsers*, *parsers*, etc). O HTML5 será o novo padrão para HTML, XHTML,

e HTML DOM. Actualmente, ainda se encontra parcialmente em fase de esboço, porém diversos *browsers* já implementam algumas de suas funcionalidades. (Hunt, 2010)

3.2.9 Less

LESS (CSS mais leve) é uma linguagem dinâmica de *stylesheet* criada por Alexis Sellier. *LESS* é *open-source*. A sua primeira versão foi escrita em *Ruby*, no entanto, em versões posteriores, o uso do *Ruby* foi substituído por *JavaScript*. A indentação do *LESS* é uma metalinguagem aninhada, onde o CSS válido é o *LESS* válido com a mesma semântica. *LESS* oferece os seguintes mecanismos: variáveis, aninhamento, mixins, operadores e funções; *LESS* permite a compilação em tempo real via *LESS.js* através do *browser*. *LESS* pode ser executado no lado do cliente ou do lado do servidor, ou pode ser compilado em CSS simples. (Wikipedia contributors, s.d.)

3.2.10 JQuery

jQuery é uma biblioteca *JavaScript cross-browser* desenvolvida para simplificar os *scripts client side* que interagem com o *HTML*. É usada por cerca de 80% dos *websites* mais visitados do mundo, sendo a mais popular das bibliotecas *JavaScript* usada nos dias de hoje. A sintaxe do *jQuery* foi desenvolvida para tornar mais simples a navegação dos documentos *HTML*, a selecção de elementos *DOM*, criação de animações, manipulação de eventos e desenvolvimento de aplicações *AJAX*. A biblioteca também oferece a possibilidade de criação de *plugins*. Tirando partido de tais facilidades, os programadores podem criar camadas de abstracção para interacções de mais baixo nível, simplificando o desenvolvimento de aplicações web dinâmicas de grande complexidade. (<http://jquery.com/>)

3.2.11 Bootstrap

Bootstrap é uma colecção de ferramentas para otimizar a criação de *websites* e aplicações *web*, desenvolvido pela *Twitter* como uma *framework* para encorajar a consistência entre os projectos internos. Utiliza uma combinação de *HTML* e modelos de *design* baseados em *CSS* e *LESS* para tipografia, formulários, botões, navegação e outros componentes de interface, bem como extensões *JavaScript* opcionais. É o projecto mais popular no *GitHub* com milhares de utilizadores, entre os quais a *MSNBC* e a *NASA*. (<http://getbootstrap.com>)

3.2.12 MS SQL Server 2012

O *SMS SQL Server* é um servidor de base de dados para *Windows* com uma interface de gestão completa. Uma vez que a aplicação necessita de armazenar alguns dados no servidor a fim de aumentar a sua eficiência e velocidade recorrendo à cache, a solução adoptada foi recorrer às bases de dados da Microsoft. Em virtude do objectivo deste projecto ser educativo, foram utilizadas diversas bases de dados para melhorar a aprendizagem e estudar as capacidades e funcionalidades de cada uma: *SQL Server Express*, *Local DB*, *SQL Server Compact 3.5* e também *SQLite*. (<http://www.microsoft.com/en-us/server-cloud/products/sql-server>)

3.2.13 SQLite

SQLite é um sistema de gestão de base de dados relacionais contido numa biblioteca escrita em linguagem C. Em contraste com outros SGBD, SQLite não é um processo separado, que é acedido a partir da aplicação cliente, mas uma parte integrante do mesmo. A biblioteca SQLite lê e escreve directamente para e do ficheiro da base de dados no disco. É auto-suficiente, não necessita de servidor nem requer configuração. Este SGBD é utilizada pelo *Calibre-e-book Manager* para armazenar os diversos dados relativos ao módulo mybooks. (<http://www.sqlite.org/>)

3.2.14 Lucene

O *Apache Lucene*, ou simplesmente *Lucene*, é um *software* de pesquisa e uma *API* de indexação de documentos, escrito em *Java*. É um *software open source* da *Apache Foundation* que foi utilizada para motor de busca em diversos módulos. Para o *Lucene* não importa a origem dos dados, formato ou a linguagem em que foram escritos, desde que esses dados possam ser convertidos para texto. Isto significa que o *Lucene* pode ser utilizado para indexar e pesquisar dados em ficheiros de textos, *Word*, *HTML*, bases de dados ou qualquer outro formato do qual possa ser extraída informação textual. (<http://lucenenet.apache.org>)

EXCERTO DO CÓDIGO VB UTILIZADO PARA CRIAÇÃO DE UM INDEX EM LUCENE:

```
Dim writerlucene As New IndexWriter(MapPath("~/App_Data/MyPicsIndex/"), New StandardAnalyzer(), True)
Dim sql_str As String = "SELECT * FROM FlickrCollections"
Dim dt As DataTable = GetDataTableSqlCE(sql_str, "MyPics")
Dim dv As New DataView(dt)

For Each rowlucene As DataRowView In dv
Dim doclucene As New Document()
doclucene.Add(New Field("ID", rowlucene("ID"), Field.Store.YES, Field.Index.UN_TOKENIZED))
doclucene.Add(New Field("Title", rowlucene("Title"), Field.Store.YES, Field.Index.TOKENIZED))
doclucene.Add(New Field("CatType", rowlucene("CatType"), Field.Store.YES, Field.Index.UN_TOKENIZED))
doclucene.Add(New Field("Description", rowlucene("Description"), Field.Store.YES, Field.Index.TOKENIZED))
doclucene.Add(New Field("IconLarge", rowlucene("IconLarge"), Field.Store.YES, Field.Index.UN_TOKENIZED))
writerlucene.AddDocument(doclucene)
Next

writerlucene.Optimize()
writer.Close()
```

3.3 CONSIDERAÇÕES DE DESIGN

Antes de criar um *website*, é sempre útil planear o que se pretende fazer antes de começar a criar páginas e escrever código. Planear um *website* com antecedência torna mais fácil a elaboração do design e a sua navegação.

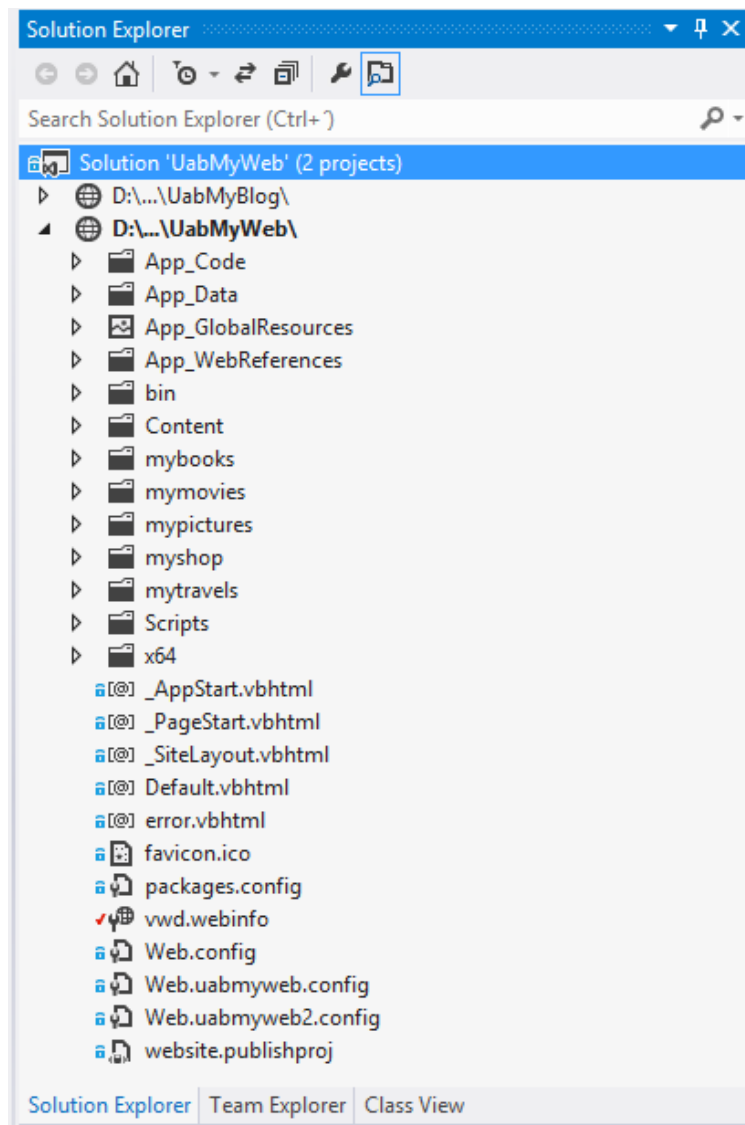
Muitas vezes o tamanho de um *website* pode determinar que planeamento deve ser feito. Um *website* pequeno, tipo brochura que fornece apenas informações estáticas pode ser um *website* relativamente simples que requer pouco planeamento. Um *website* que acede a bases de dados, autentica utilizadores, localizado e obedecendo às normas de acessibilidade pode exigir um planeamento mais longo. (MSDN, 2014)

As seguintes considerações devem ser tomadas em conta:

- O tipo de *website* que se pretende.
- O tipo de *navegação* que se vai utilizar.
- Como criar um *design* consistente para todas as páginas do *website*.
- Como aceder às bases de dados

3.3.1 Tipo de projecto

Existem dois tipos de projectos para Web possíveis de criar no Visual Studio. O modelo de projecto *Website* padrão usa uma estrutura de directório de ficheiros para definir o conteúdo de um projecto. Neste modelo, não há arquivo de projecto e todos os ficheiros no directório fazem parte do projecto.



Modelo de projecto Website padrão

Em contraste, num projecto de aplicação Web apenas os ficheiros que são explicitamente mencionados no arquivo do projecto da solução fazem parte do projecto. Esses ficheiros são exibidos no *Solution Explorer*, e eles são os únicos ficheiros que são compilados durante a compilação.

3.3.2 Navegação ASP.NET

Com o crescimento de um website, rapidamente pode tornar-se difícil gerir todos os seus links. O ASP.NET dispõe de *server controls* e classes que permitem fornecer uma forma consistente de os utilizadores navegarem no website, armazenando os links para todas as páginas num único ficheiro (geralmente um arquivo XML ou uma tabela de uma base de dados). Uma parte fundamental da navegação do website ASP.NET é o *site map provider*. Esta é uma classe que é usada com um *site map data source* que expõe as informações de navegação. Por exemplo, o *site map provider* padrão do ASP.NET obtém dados de navegação de um arquivo XML que é chamado *Web.sitemap*, e comunica os dados para o *server control SiteMapPath* directamente.

EXCERTO DO CÓDIGO XML UTILIZADO PARA DEFINIR AS CATEGORIAS E SUBCATEGORIAS DOS DIVERSOS MÓDULOS:

```
<?xml version="1.0"?>
<categories>
  <mybooks name="my books" url="mybooks" root="True" active="True">
    <cat ID="0" Title="titles" Index="0" Icon="titles" Bgc="green" />
    <cat ID="1" Title="authors" Index="1" Icon="authors" Bgc="orange" />
    <cat ID="3" Title="formats" Index="0" Icon="formats" Bgc="blueDark" />
    <cat ID="4" Title="identifiers" Index="0" Icon="identifiers" Bgc="greenDark" />
    <cat ID="5" Title="languages" Index="0" Icon="languages" Bgc="purple" />
    <cat ID="6" Title="publisher" Index="1" Icon="publisher" Bgc="red" />
    <cat ID="7" Title="rating" Index="0" Icon="rating" Bgc="yellow" />
    <cat ID="8" Title="series" Index="1" Icon="series" Bgc="pink" />
    <cat ID="9" Title="tags" Index="1" Icon="tags" Bgc="gray" />
  </mybooks>
  <mymovies name="my movies" url="mymovies" root="True" active="True">
    <cat ID="0" Title="movies" Index="0" Icon="movies" Bgc="purple" />
    <cat ID="1" Title="people" Index="0" Icon="people" Bgc="orange" />
    <cat ID="2" Title="genres" Index="1" Icon="genres" Bgc="blueDark" />
    <cat ID="3" Title="year" Index="0" Icon="year" Bgc="red" />
    <cat ID="4" Title="subtitles" Index="0" Icon="subtitles" Bgc="blue" />
    <cat ID="5" Title="studios" Index="2" Icon="studios" Bgc="pink" />
    <cat ID="6" Title="rating" Index="0" Icon="rating" Bgc="yellow" />
  </mymovies>
  <myshop name="my shop" url="myshop" root="True" active="True">
    <cat ID="1000" Title="books" SearchGroup="Books" Keywords="asp.net" Index="4" Icon="books" Bgc="orange" />
    <cat ID="265011" Title="movies" SearchGroup="DVD" Keywords="tarantino" Index="1" Icon="movie" Bgc="purple" />
    <cat ID="301668" Title="music" SearchGroup="Music" Keywords="pink floyd" Index="0" Icon="music" Bgc="red" />
  </myshop>
</categories>
```

Ficheiro categories.xml, que funciona também como custom site map provider

Na aplicação web foram desenhados diversos *custom site map providers*. Os dados da navegação geral são obtidos de um arquivo XML, *categories.xml*, que além de funcionar como site map provider também fornece os dados da navegação de cada módulo.

ID	Title	CatType	Url	Parent
0	Home	rootcat	/mypictures/	
7105944-72157640520062184	Morocco	countrycat	morocco	7105944-72157625110456705
7105944-72157616904847692	Egypt	countrycat	egypt	7105944-72157625110456705
7105944-72157625629454678	Mexico	countrycat	mexico	7105944-72157616814207671
7105944-72157616904851032	Brazil	countrycat	brazil	7105944-72157616814207671
7105944-72157616904849532	Usa	countrycat	usa	7105944-72157616814207671
7105944-72157616904885208	Brunei	countrycat	brunei	7105944-72157616814179227
7105944-72157616904883576	Cambodia	countrycat	cambodia	7105944-72157616814179227
7105944-72157616763517747	China	countrycat	china	7105944-72157616814179227
7105944-72157616814568787	Hong Kong Sar	countrycat	hong-kong-sar	7105944-72157616814179227
7105944-72157616814565837	Indonesia	countrycat	indonesia	7105944-72157616814179227
7105944-72157616814567151	India	countrycat	india	7105944-72157616814179227
7105944-72157625235171328	Japan	countrycat	japan	7105944-72157616814179227
7105944-72157616814562349	Malaysia	countrycat	malaysia	7105944-72157616814179227
7105944-72157616763402713	Macau Sar	countrycat	macau-sar	7105944-72157616814179227
7105944-72157616814555099	Nepal	countrycat	nepal	7105944-72157616814179227
7105944-72157616904871198	Philippines	countrycat	philippines	7105944-72157616814179227
7105944-72157616904874584	Myanmar	countrycat	myanmar	7105944-72157616814179227
7105944-72157616904869404	Republic Of Singapore	countrycat	republic-of-singapore	7105944-72157616814179227
7105944-72157616904868094	Sri Lanka	countrycat	sri-lanka	7105944-72157616814179227
7105944-72157616904865804	Thailand	countrycat	thailand	7105944-72157616814179227
7105944-72157631538717181	Taiwan	countrycat	taiwan	7105944-72157616814179227
7105944-72157616814546817	Vietnam	countrycat	vietnam	7105944-72157616814179227
7105944-72157631648717385	Andorra	countrycat	andorra	7105944-72157616813773871
7105944-72157631596624329	Albania	countrycat	albania	7105944-72157616813773871
7105944-72157616763362527	Austria	countrycat	austria	7105944-72157616813773871
7105944-72157616813778987	Belgium	countrycat	belgium	7105944-72157616813773871
7105944-72157631960923167	Bosnia And Herzegovina	countrycat	bosnia-and-herzegovina	7105944-72157616813773871
7105944-72157631565691419	Bulgaria	countrycat	bulgaria	7105944-72157616813773871
7105944-72157616814138561	Croatia	countrycat	croatia	7105944-72157616813773871
7105944-72157635617505214	Czech Republic	countrycat	czech-republic	7105944-72157616813773871
7105944-72157625235222698	Denmark	countrycat	denmark	7105944-72157616813773871

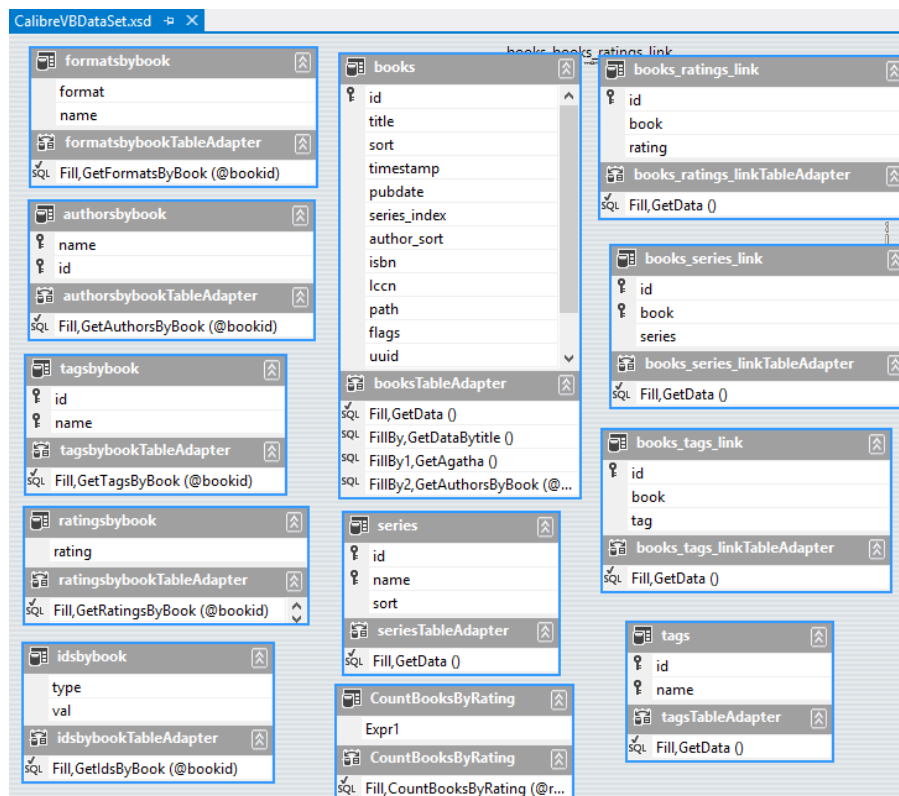
Datatable representando as informações de navegação do módulo mypictures. Funciona como custom site map provider que é obtido através da leitura da API do Flickr onde está guardada a informação relativa aos álbuns (anteriormente chamados photosets) do utilizador.

3.3.3 Acesso a dados

Utilizando o *data binding* do ASP.NET, podemos conectar qualquer controle apropriado a propriedades simples, coleções, expressões e/ou métodos. Esta possibilidade proporciona uma grande flexibilidade quando se usa uma base de dados.

Os controlos *Data Source* são parte de um padrão de dados comum. Este padrão de dados separa o código de *data-access* e o código de *business logic* a partir das páginas que formam a camada de apresentação do website. A camada de *access code* consiste em métodos que são usados para aceder a um repositório de dados. A camada *business logic* estabelece as regras para a camada de *access code*, tais como restringir o acesso de quem pode visualizar ou alterar os dados. A camada de apresentação é composta de páginas que o utilizador acede para visualizar e modificar os dados.

Tirando partido dos controlos *DataSource* como o *LinQDataSource*, *ObjectDataSource* ou *SqlDataSource* podemos criar uma camada de *data access* e uma camada de *business logic* separadas da camada de apresentação.



Data Access Layer (DAL) utilizado no módulo mybooks para aceder à base de dados do Calibre (SQLite)

Também deve ser considerado o uso de dados na memória (cache). Se uma parte significativa dos dados não se altera com frequência e é comum a diversos utilizadores, podemos manter os dados na memória do servidor. Assim reduzimos o número de consultas à base de dados e aumentamos a velocidade. O ASP.NET dispõe da classe *DataSet* que permite criar uma base de dados em memória. Outro aspecto útil do objecto *DataSet* é que ele permite uma aplicação criar subconjuntos de dados de uma ou mais fontes para o espaço da aplicação podendo manipular os dados na memória, mantendo a sua forma relacional. (MSDN, 2014)

EXCERTO DO CÓDIGO VB UTILIZADO PARA GUARDAR EM MEMÓRIA (CACHE) UMA TABELA COM AS INFORMAÇÕES DE NAVEGAÇÃO DO MÓDULO MYTRAVELS :

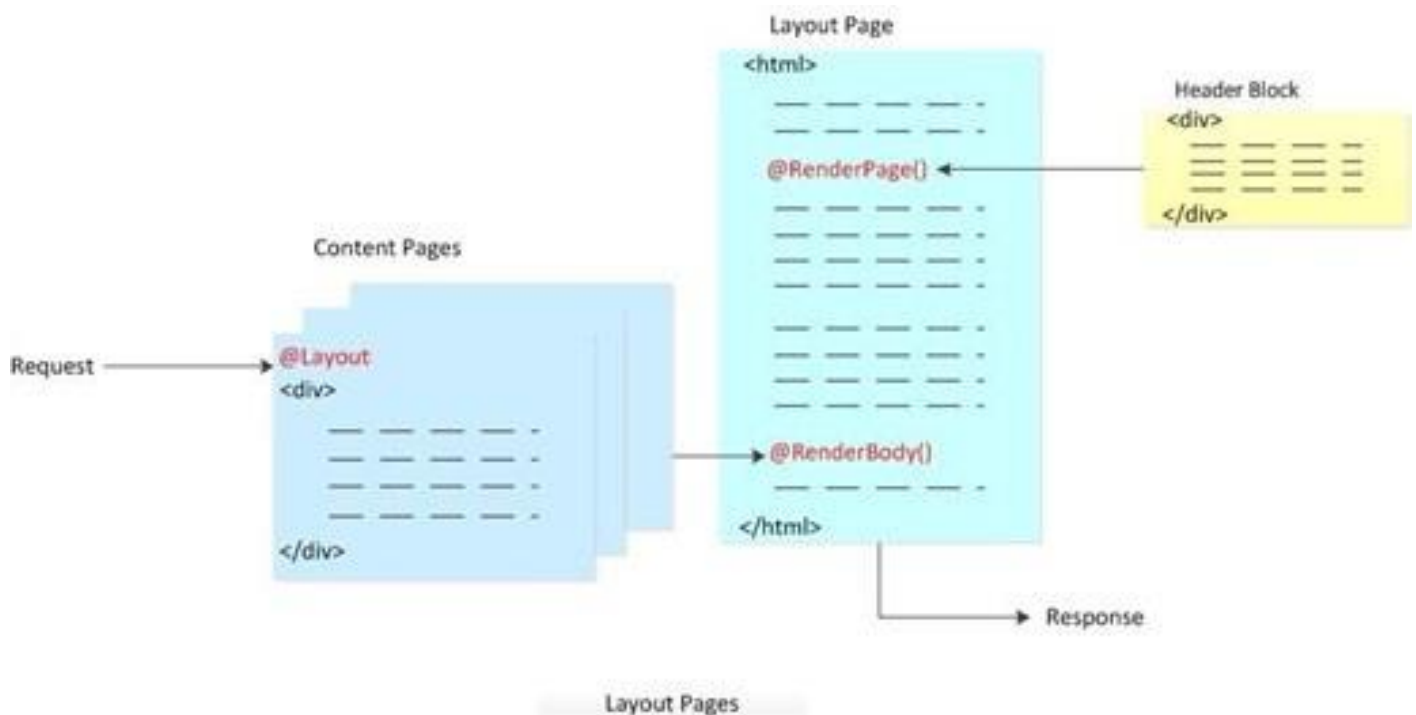
```
Public Shared Function GetListForSiteMapMyTravels(_name As String, lang As String, Optional useCache As Boolean = True) As DataTable
```

```
    Dim cachekey = _name & lang
    Dim dataSet As DataSet
    Dim table As DataTable = Nothing
    table = New DataTable(cachekey & "Table")
    If Not useCache Then
        System.Web.HttpContext.Current.Cache.Remove(cachekey)
    End If
    If System.Web.HttpContext.Current.Cache(cachekey) Is Nothing Then
        Dim column As DataColumn
        Dim row As DataRow
        (...code removed for brevity...)
        Dim i As Integer
        Dim _CatRoot() As String = XMLReturnArray("mytravels", "Title")
        For Each _cat As String In _CatRoot
            row = table.NewRow()
            i = i + 1
            row("ID") = i
            row("Title") = StrConv(_cat, VbStrConv.ProperCase)
            row("CatType") = "region"
            row("Url") = "/mytravels/browse/" & LCase(StripSlug(_cat))
            row("Parent") = "Home"
            table.Rows.Add(row)
        Next
        System.Web.HttpContext.Current.Cache.Insert(cachekey, table, Nothing,
System.Web.Caching.Cache.NoAbsoluteExpiration, TimeSpan.FromMinutes(CacheTimeSpan))
    Else
        table = DirectCast(System.Web.HttpContext.Current.Cache(cachekey), DataTable)
    End If
    Return table
End Function
```

3.3.4 Aparência e *design* consistente

Vários recursos estão disponíveis em ASP.NET para ajudar a criar e manter uma aparência e design consistente para todo o website, tais como *themes* e *master pages*. Estes recursos podem ser usados o início do processo de desenvolvimento para criar um aspecto consistente.

Normalmente, os websites têm conteúdo comum que é exibido em todas as páginas, tais como o cabeçalho e rodapé, ou uma *textbox* seguida da informação que indica se o utilizador está autenticado ou não. O ASP.NET permite que se crie um ficheiro separado contendo um fragmento do conteúdo com texto, imagens, código, etc, tal como uma página web normal. Podemos inserir este fragmento em qualquer outra página onde pretendemos que esta informação apareça. Desta forma, evitamos copiar e colar o mesmo conteúdo em todas as páginas e simplificamos qualquer actualização que seja necessária fazer. Basta actualizar o fragmento e todas as actualizações serão espelhadas onde quer que o fragmento esteja inserido.



Anatomia de uma página Layout

A utilização de páginas de *layout* é uma técnica mais estruturada para criarmos uma aplicação web com aparência e design consistente é. Uma página de *layout* define a estrutura de uma página web, mas não contém obrigatoriamente o conteúdo. Após a criação da página de *layout*, criamos os fragmentos (*Content Pages*) para o conteúdo e, de seguida, unimos os fragmentos à página de layout. Quando estas páginas são visualizadas, eles serão formatados de acordo com a página de *layout*. Neste sentido, uma página de *layout* funciona como uma espécie de *template* para o conteúdo que é definido em outras páginas. (FitzMacken, 2014)

EXCERTO DO CÓDIGO HTML UTILIZADO PARA A PÁGINA LAYOUT (SITE_LAYOUT.VBHTML):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>@PageData("Title")</title>
  <meta name="description" content="my website" />
  <meta name="keywords" content="my website, books, movies, ptotography, travels, shopping" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  @Styles.Render("~/bundles/css")
  @Scripts.Render("~/bundles/modernizr")
  @Scripts.Render("~/bundles/jquery", "~/bundles/jqueryui", "~/bundles/others")
  @RenderSection("Scripts", required:=False)
</head>
<body>
  <header id="nav-bar">
    @***code removed for brevity***@
  </header>
  <div id="body">
    @RenderBody()
  </div>
  <footer>
    @***code removed for brevity***@
  </footer>
  @RenderSection("BingMaps", required:=False)
</div>
</body>
</html>
```

A página layout na aplicação web é semelhante a qualquer outra página HTML, excepto que contém uma chamada para o método *RenderBody*. A posição do método *RenderBody* na página de layout determina onde a página de conteúdo será incluída.

3.3.5 Funcionalidades AJAX

As funcionalidades *AJAX* do ASP.NET permitem criar rapidamente aplicações web responsivas que melhoraram a experiência dos utilizadores através da utilização de elementos visuais atractivos e familiares para os utilizadores.

Permitem também actualizar fragmentos da página sem ser necessário actualizar a página inteira evitando assim viagens desnecessárias ao servidor, melhorando assim a eficiência da aplicação.

Os recursos *AJAX* integram-se com a plataforma de desenvolvimento baseada em servidores ASP.NET e incluem scripts que incorporam tecnologias *cross-browser ECMAScript (JavaScript)* e *HTML* dinâmico (*DHTML*).

* *Lazy Load - jQuery plugin for lazy loading images*

```
(function($, window, document, undefined) {
  var $window = $(window);

  $.fn.lazyload = function(options) {
    var elements = this;
    var $container;
    var settings = {
      threshold      : 0,
      failure_limit  : 0,
      event          : "scroll",
      effect         : "show",
      container      : window,
      data_attribute : "original",
      skip_invisible : true,
      appear        : null,
      load          : null,
      placeholder   :
        "data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAEAAAABCAyAAAFcSJAAAAAAXNSR0IArs4c6
    };
```

A utilização do plug-in lazy load para JQuery permite o carregamento de imagens e conteúdo somente quando estão visíveis evitando assim o desperdício de recursos: se uma imagem não está visível não precisa de ser carregada.

Nas versões mais recentes do Visual Studio, é incluída uma biblioteca *jQuery* quando se cria uma aplicação web utilizando os *templates* disponíveis. (MSDN, 2014)

EXCERTO DO CÓDIGO JAVASCRIPT UTILIZANDO FUNCIONALIDADES AJAX PARA ACEDER À API DE ROTTENTOMATOES QUE FORNECE AS CRÍTICAS DOS FILMES PARA O MÓDULO MYMOVIES:

```

<script>
  var apikey = '@rtapikey';
  var baseUrl = "http://api.rottentomatoes.com/api/public/v1.0";
  // construct the uri with our apikey
  var moviesAliasUrl = baseUrl + '/movie_alias.json?apikey=' + apikey;
  var imdbid = '@movie.imdb';
  var query = imdbid.substring(2);
  //var query = '0091605';
  $(document).ready(function () {
    // send off the query
    $.ajax({
      url: moviesAliasUrl + '&id=' + query + '&type=imdb',
      dataType: "json",
      success: searchCallbackAlias,
      error: rterror
    });
  });
  function rterror(data) {
    document.getElementById("rterror").innerHTML = 'error on rt api: not available ' + data;
    $('#rt').hide();
  }
  // callback for when we get back the results
  function searchCallbackAlias(data) {
    //code removed for brevity//
  }
</script>
<noscript>Your browser does not support JavaScript!</noscript>

```

critics rating:






tomatometer:



audience rating:



Found 52 reviews for Gravity

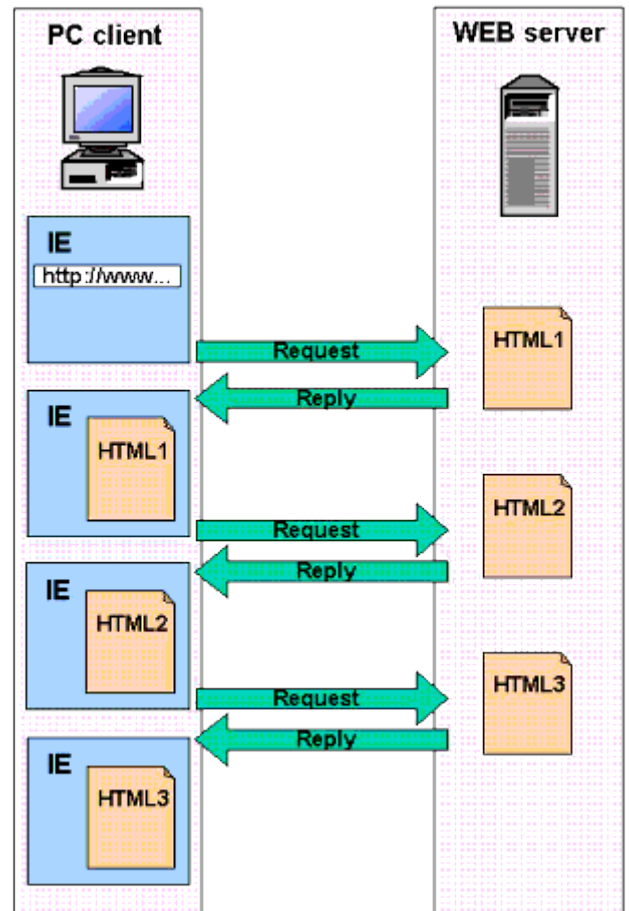
- 
ChristyLemire.com
 - Believe the hype: Gravity is as jaw-droppingly spectacular as you've heard - magnificent from a technical perspective but also a marvel of controlled acting and precise tone. -- By: Christy Lemire, 2013-10-05 - Full Review
- 
The Atlantic
 - Unfolding as a series of terrifying object lessons in Newtonian physics, the movie lends new meaning to the phrase "spatial geometry." -- By: Christopher Orr, 2013-10-04 - Full Review
- 
New Yorker
 - Gravity is not a film of ideas, like Kubrick's techno-mystical 2001, but it's an overwhelming physical experience -- a challenge to the senses

3.3.6 Funcionalidades de gerenciamento do estado da aplicação

O HTTP é um protocolo *stateless*. Cada pedido (*request*) é atendido tal como é recebido; após o pedido ser processado, todos os dados serão descartados. Nenhum estado é mantido mesmo entre pedidos do mesmo cliente. No entanto, para a maioria das aplicações web, é útil manter o estado entre os diversos pedidos.

A ASP.NET dispõe de funcionalidades de gerenciamento de estado (*state management*) que permitem armazenar informações entre diversos pedidos, (*requests*), tais como informações dos clientes ou o conteúdo de um carrinho de compras.

Podemos guardar e salvar informações específicas da aplicação, da sessão, da página, do utilizador, que incluem o uso de *cookies*, *view state*, *session state*, *application state*, *hidden fields*, *query strings*, *profile properties*, etc. Esta informação pode ser independente de quaisquer outros controles da página.



EXCERTO DO CÓDIGO VB QUE VERIFICA SE O UTILIZADOR ESTÁ AUTENTICADO ATRAVÉS DA SUA CONTA FLICKR UTILIZANDO FUNCIONALIDADES DO GERENCIAMENTO DE ESTADO:

```

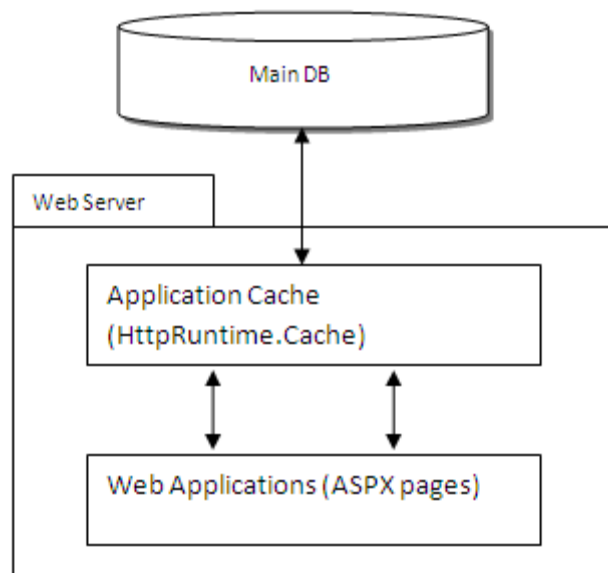
If Request.QueryString("oauth_verifier") IsNot Nothing AndAlso Session("RequestToken") IsNot Nothing Then
    Dim f As Flickr = FlickrManager.GetInstance()
    Dim requestToken As OAuthRequestToken = TryCast(Session("RequestToken"), OAuthRequestToken)
    Dim accessToken As OAuthAccessToken = f.OAuthGetAccessToken(requestToken,
Request.QueryString("oauth_verifier"))
    FlickrManager.OAuthToken = accessToken.Token
    FlickrManager.OAuthTokenSecret = accessToken.TokenSecret
    ResultsLabel.Text = "You successfully authenticated as " + accessToken.FullName
End If

```

3.3.7 Armazenamento em memória (cache)

O desempenho de uma aplicação web pode ser aumentado se armazenarmos em memória os dados que são acessados frequentemente e requerem um tempo de processamento significativo para serem criados. Por exemplo, se a aplicação processa largas quantidades de dados usando lógica complexa e retorna a informação em forma de relatório, é ineficiente recriar o relatório cada vez que um utilizador o solicita. Da mesma forma, se a aplicação inclui uma página que processa dados complexos, mas esses dados são actualizados com pouca frequência, é ineficiente o servidor recriar essa página em cada pedido.

As novas capacidades do *HTML5* permitem a cache persistente de dados e ficheiros que sobrevivem a sessões do *browser* e ciclos de energia. Esses recursos podem ser utilizados para reduzir a latência associada com o *download* de ficheiros em redes móveis, bem como para tirar partido de capacidades como o *localStorage* para armazenar em cache ficheiros *JavaScript* e *CSS*.



A *cache* da ASP.NET é um recurso específico de armazenamento em memória para aplicações web. Para ajudar a aumentar o desempenho das aplicações podemos usar dois tipos de cache:

- **Cache da aplicação**, que permite armazenar em memória dados gerados pela aplicação, como por exemplo, um Dataset.
- **Cache de saída da página**, que guarda em memória o resultado do processamento da página, evitando assim o seu reprocessamento cada vez que um utilizador a solicita.

3.3.7.1 Cache da aplicação

O exemplo seguinte, utilizado no módulo myshop, demonstra como se pode guardar em cache a informação relativa a uma determinada colecção de produtos do catálogo da Amazon durante um determinado período de tempo. Na próxima vez que um utilizador desejar consultar estes dados, estes irão ser carregados a partir da memória do servidor evitando assim uma viagem ao servidor da Amazon.

```

If useCache Then
    Dim cacheKey As String = "myshop_browse_" + nodeId
    ' Here verify if this request is in the cache
    If System.Web.HttpContext.Current.Cache(cacheKey) Is Nothing Then
        ' Cache key null. We need to send the request and then insert the response in the cache.
        Try
            browseNodeLookupResponse = awClient.BrowseNodeLookup(browseNodeLookup)
            System.Web.HttpContext.Current.Cache.Insert(cacheKey, browseNodeLookupResponse, Nothing,
DateTime.Now.AddDays(30), TimeSpan.Zero)
        Catch e As Exception
            Throw New Exception(e.ToString())
        End Try
    Else
        ' Cache key found. Load results from cache.
        browseNodeLookupResponse = DirectCast(System.Web.HttpContext.Current.Cache(cacheKey),
BrowseNodeLookupResponse)
    End If

```

3.3.7.2 Cache de saída da página

O exemplo seguinte demonstra como se pode definir a duração (em milissegundos) da cache de saída da página. Durante a próxima hora a página não será processada e os seus valores serão obtidos a partir da memória.

```
<%@ OutputCache Duration="3600" VaryByParam="none" %>
```

ou

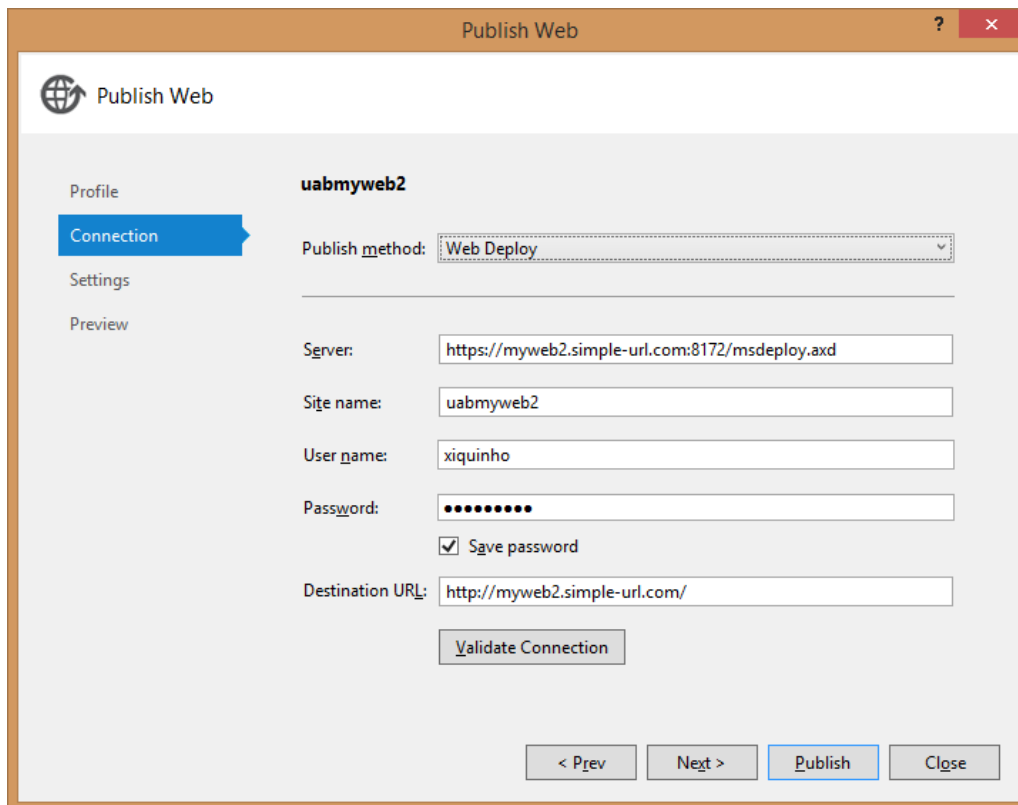
```
Response.OutputCache(3600)
```

Fazendo uso destas funcionalidades podemos aumentar o desempenho e velocidade da aplicação. (MSDN, 2014)

3.3.8 Segurança

Além das funcionalidades de segurança da *.NET Framework*, a ASP.NET fornece uma infraestrutura de segurança para autenticação e autorização de acesso do utilizador, bem como para executar outras tarefas relacionadas com a segurança. Os utilizadores podem ser autenticados usando a autenticação do *Windows* fornecida pelo *IIS*. Alternativamente, podemos usar a autenticação de formulários ASP.NET.

ASP.NET sempre é executado com uma identidade específica do *Windows* para que a aplicação web possa ser protegida usando os recursos do *Windows*, tais como *Access Control Lists (ACLs)* de sistema de ficheiros *NTFS* e permissões de bases de dados.

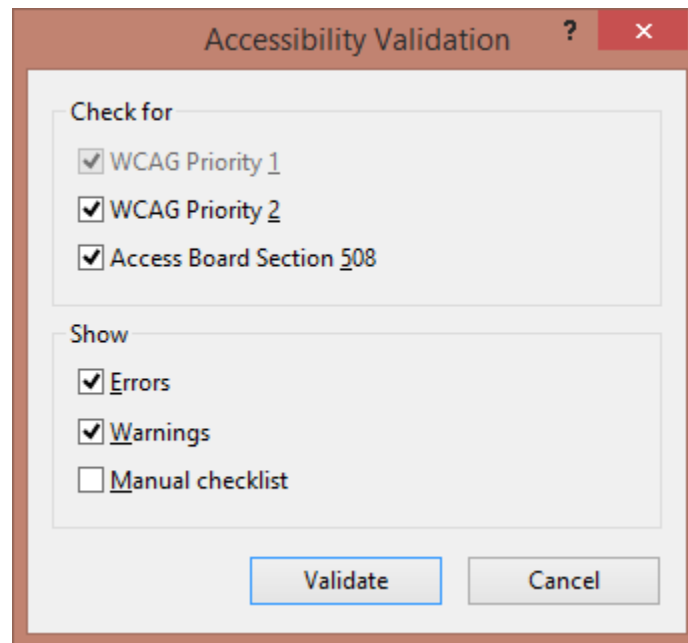


WebDeploy, o método utilizado para publicação da aplicação web.

Microsoft Web Deploy - é uma tecnologia utilizada para enviar de forma abrangente a aplicação web ao servidor *IIS*. *Web Deploy* permite não só publicar os ficheiros, mas também o esquema e conexões para a base de dados, correr scripts para fazer alterações, definir *ACLs* de segurança, para sincronizar servidores *IIS* em *web farms* ou a migrar para versões mais recentes do *IIS*. Todas estas etapas podem ser automatizadas utilizando o *Visual Studio*. (Guthrie, 2010)

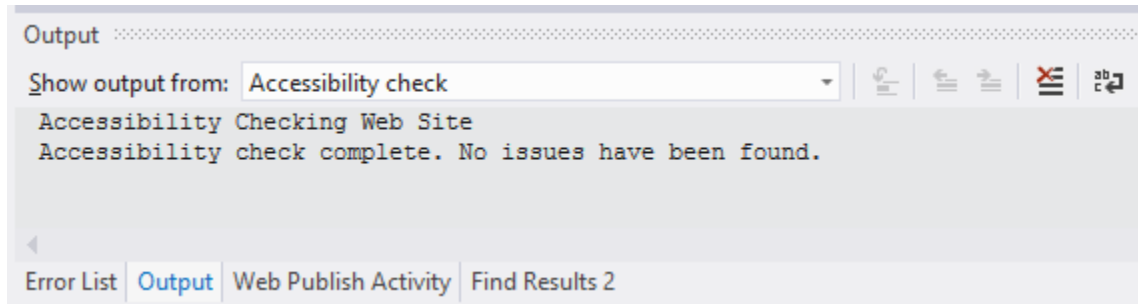
3.3.9 Acessibilidade

Em informática, programas com recursos de acessibilidade são ferramentas ou conjuntos de ferramentas que permitem que portadores de deficiências (as mais variadas) utilizem os recursos que o computador oferece. Essas ferramentas podem constituir leitores de ecrã para deficientes visuais visualizarem páginas web, teclados virtuais para portadores de deficiência motora ou com dificuldades de coordenação motora, sintetizadores de voz para pessoas com problemas de fala.



Em aplicações web o termo acessibilidade refere-se também a recomendações do W3C, (W3C, 2014) que visam permitir que todos possam ter acesso aos *websites*, independente de terem alguma deficiência ou não. As recomendações abordam desde o tipo de fonte a ser usado, bem como seu tamanho e côr de acordo com as necessidades do utilizador, até a recomendações relativas ao código (*HTML* e *CSS*, por exemplo). Tal ajuda os utilizadores uma melhoria na visualização, podendo assim garantir que a informação chegue ao maior número de pessoas de forma simples e precisa. As páginas web acessíveis têm as seguintes vantagens:

- São utilizáveis pelo maior número de pessoas possível.
- Envolvem princípios de *design* que frequentemente beneficiam todos os utilizadores, não apenas as pessoas com deficiência.
- Atendem aos requisitos de muitas instituições que exigem que todas as aplicações Web sejam acessíveis.



Ferramentas do Visual Studio para verificar se uma aplicação web obedece aos padrões de acessibilidade.

Na maioria das vezes, os controles ASP.NET obedecem aos padrões de acessibilidade expondo as propriedades que podem ser definidas para tornar as páginas acessíveis. (MSDN, 2014)

3.3.10 Globalização e Localização

Globalização é o processo de desenvolver aplicações que funcionam para várias culturas. Localização é o processo de personalizar a aplicação para uma determinada cultura e local.

Se a aplicação for visualizada por utilizadores de línguas diferentes, torna-se necessário que possam visualizar as páginas na sua própria língua. A ASP.NET permite criar páginas que vão obter o seu conteúdo a ficheiros de recursos ou outras fontes, com base no idioma padrão da configuração do *browser* ou outro idioma escolhido pelo utilizador. As páginas ASP.NET permitem configurar os seus elementos para obter os seus valores a partir dos ficheiros de recurso e uma vez executada a página, as expressões de recurso são substituídas por expressões do ficheiro de recurso apropriado. (MSDN, 2014)

Name	Value	Comment
Home	Home	
not_yet_rated	(not yet rated)	
var_codlang	en	
var_last_visited	last visited:	
var_you_might_be	you might be also interested in:	
*		

Exemplo de ficheiro de recurso para idioma padrão

Name	Value	Comment
Home	Home	
not_yet_rated	(ainda não avaliado)	
var_codlang	pt	
var_last_visited	última visita:	
var_you_might_be	poderá estar também interessado em:	
*		

Ficheiro de recurso equivalente para a língua portuguesa

EXCERTO DO CÓDIGO HTML UTILIZADO PARA INVOCAR A EXPRESSÃO DE RECURSO:

```

If showHeading Then
  @<h3 class="shopamazon">@Resources.MainResources.var_you_might_be</h3>
  @<br />
End If

```

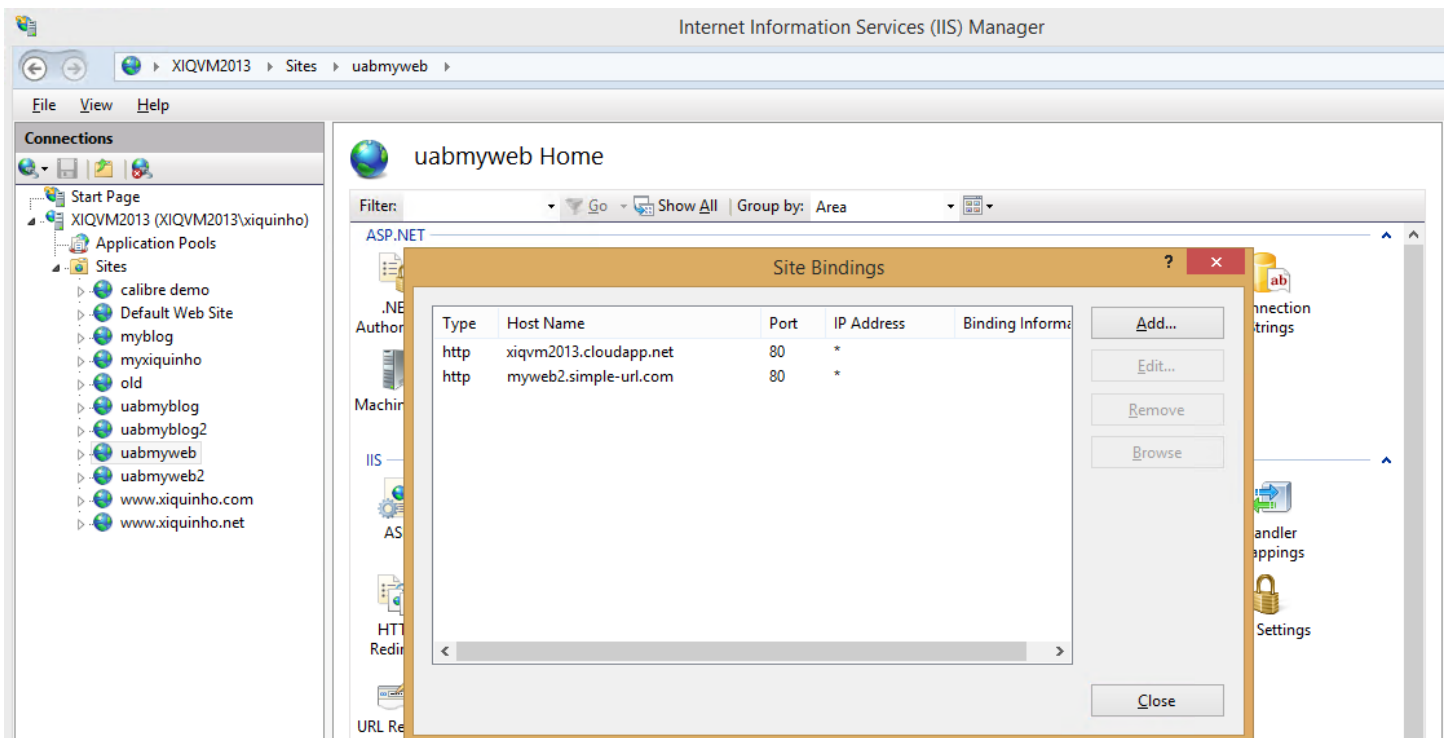
Uma vez a página executada, a expressão de recurso "var_you_might_be" é substituída pela expressão do ficheiro de recurso apropriado.

3.3.11 Compilação

Compilar uma página ou um *website* é parte do desenvolvimento de uma aplicação web, e destina-se a ajudar a encontrar erros que podem ocorrer quando da execução. No entanto um *website* também pode ser publicado sem efectuar a compilação, copiando todos os ficheiros para o servidor. Neste caso, quando uma página é solicitada pela primeira vez, a ASP.NET compila dinamicamente a aplicação executando efectivamente os mesmos passos que o processo executaria no Visual Studio, guardando a saída resultante para que as páginas não precisem de ser recompiladas a cada solicitação. (MSDN, 2014)

Boas práticas em ambiente de produção

Para diminuir o risco de erros de compilação, deve criar-se dois *websites* no IIS para cada aplicação web. Por exemplo, seja o *website* registado no IIS como *myweb*, apontando para *www.myweb.com*, o *website* que está neste momento em produção. Para actualizar esta aplicação web, deve publicar-se as alterações com ajuda do *WebDeploy* para o *website* registado no IIS como *myweb2*, apontando para *www2.myweb.com*. De seguida, testamos a aplicação no endereço *www2.myweb.com*, certificando assim que não existem erros no ambiente mais próximo possível da produção.



Alteração manual dos apontadores (site bindings) no IIS

Se depois de efectuados os testes não foi encontrado nenhum erro, apontamos o *website* registado no *IIS* como *myweb2* para *www.myweb.com* e o *website* registado no *IIS* como *myweb* para *www2.myweb.com*. Da próxima vez que necessitarmos de fazer actualizações, publicamos de novo as alterações para *www2.myweb.com* e depois de testado invertemos o processo: apontamos de novo *myweb* para *www.myweb.com* e *myweb2* para *www2.myweb.com*. Seguindo esta metodologia, qualquer erro de última hora que possa surgir nunca é visível no *website* em produção, podendo ser detectado e corrigido no *website* que está em teste.

```
appcmd set site /site.name: myweb /-bindings.[protocol='http',bindingInformation='*:80:www.myweb.com']
```

```
appcmd set site /site.name: myweb2 /+bindings.[protocol='http',bindingInformation='*:80:www.myweb.com']
```

```
appcmd set site /site.name: myweb2 /-bindings.[protocol='http',bindingInformation='*:80:www2.myweb.com']
```

```
appcmd set site /site.name: myweb /+bindings.[protocol='http',bindingInformation='*:80:www2.myweb.com']
```

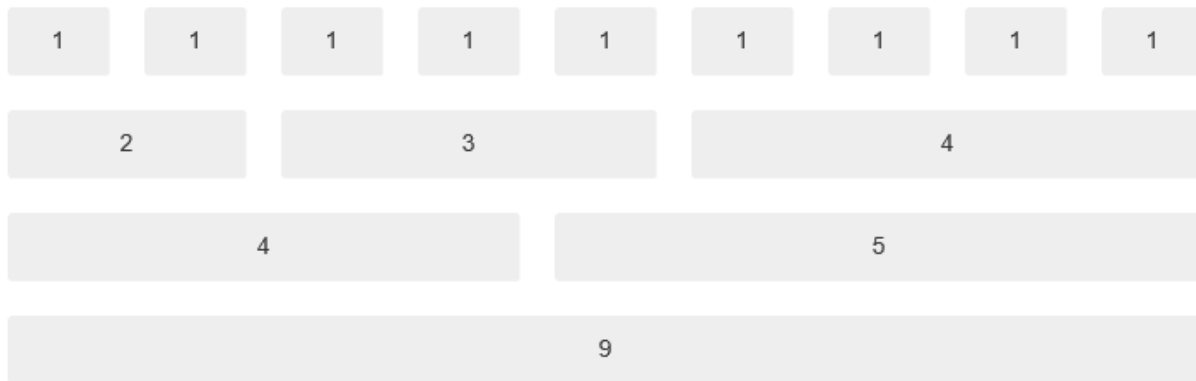
O processo pode ser automatizado com recurso a *comandlets* do *IIS*. Os comandos acima permitem trocar os apontadores (*bindings*) entre os dois *websites*.

3.4 WEB DESIGN RESPONSIVO

Web Design Responsivo é uma abordagem de *web design* destinada a construir aplicações *web* que fornecem uma ótima experiência de visualização, fácil leitura e navegação com um mínimo de redimensionamento e visionamento, para uma ampla gama de dispositivos (de monitores de computador a telefones móveis).

3.4.1 Layout flexível baseado em grelha fluída

O conceito de grelha fluída requer que as dimensões dos elementos sejam definidas por unidade relativas como percentagem, em vez de unidades absolutas como *pixels* ou pontos. (Marcotte, Fluid Grids, 2009)



O sistema padrão da grelha do *Bootstrap* utiliza 12 colunas, para um *container* largo de 940 *pixels*. Com a utilização de recursos responsivos, a grelha adapta-se a larguras entre os 724 *pixels* e 1170 *pixels*, dependendo do *viewport*. Abaixo de 767px, as colunas fluem e empilham-se verticalmente.

EXCERTO DO CÓDIGO HTML UTILIZADO NUMA GRELHA BOOTSTRAP :

```
<div class="row">
  <div class="row span9">
    @CommonHelpers.Getbreadcrumb(LCase(book.Title), "mybooks", book.Title, "", "", "", "")
    @CommonHelpers.FormatDataView("formats", formatsdv, "mybooks/file/" & id)
```

(code removed for brevity)

```
</div>
  <div class="span3">
    @CommonHelpers.FormatCover("mybooks", book.title, book.has_cover, id, "", "")
  </div>
<div class="row span12">
  @CommonHelpers.FormatString("Summary", book.comments)
</div>
</div>
```

Para um layout simples de duas colunas, cria-se uma linha e adiciona-se o número apropriado de .span colunas que deve ser sempre somar 12 em cada linha (ou o número de span da linha parent). Neste exemplo, a primeira linha tem uma 9 + 3 span e a segunda linha tem 12 span.*

3.4.2 Imagens e Media flexíveis

Imagens flexíveis são as que também têm suas dimensões em unidades relativas, de modo a se ajustem dentro do seu elemento HTML. (Marcotte, Fluid Images, 2011)

```
img {
  width: auto;
  height: auto;
  max-width: 100%;
  vertical-align: middle;
  border: 0;
  -ms-interpolation-mode: bicubic;
}
```

[my web](#) [my books](#) [my movies](#) [my shop](#) [my pictures](#) [my travels](#)

the butler

🔍

[home](#) / [my movies](#) / [the butler](#)

genres: [Drama]

studios: [Anchor Bay Entertainment] [The Weinstein Company] [Laura Ziskin Productions] [Windy Hill Pictures] [Follow Through Productions] [Salamander Pictures] [Pam William Productions] [A Lee Daniels Films] [Starz Media]



year: 2013

updated: 2014 Jan 24

dvd release:

rating: 3.5

id's: [imdb] [rotten tomatoes]

[▶ watch trailer](#)

[Like](#) Be the first of your friends to like this.

O estilo `max-width: 100%;` fornece uma restrição incrivelmente útil para todas as imagens no documento. A imagem poderá ser visualizada a qualquer tamanho, desde que seja mais estreita do que o elemento que a contém. Mas se acontecer ser mais larga, a directiva `max-width: 100%` obriga a largura da imagem a ficar com a mesma largura do seu recipiente. (Marcotte, Responsive Web Design, 2011)

[my web](#) [my books](#) [my movies](#) [my shop](#) [my pictures](#) [my travels](#)

the butler



[home](#) / [my movies](#) / [the butler](#)

genres: [Drama]

studios: [Anchor Bay Entertainment] [The Weinstein Company] [Laura Ziskin Productions] [Windy Hill Pictures] [Follow Through Productions] [Salamander Pictures] [Pam William Productions] [A Lee Daniels Films] [Starz Media]

year: 2013

updated: 2014 Jan 24

dvd release:

rating: 3.5

id's: [imdb] [rotten tomatoes]

[▶ watch trailer](#)



3.4.3 Media queries, um módulo da especificação CCS3

Media queries permitem que a página use diferentes regras de estilo CSS com base nas características do dispositivo onde o *website* está sendo exibido, sendo mais comum a largura de visualização do browser.

```
/*@media (min-width: 768px) and (max-width: 979px)*/
```

```
@media (min-width: 768px) and (max-width: 979px) {
  .hidden-desktop {
    display: inherit !important;
  }
  .visible-desktop {
    display: none !important ;
  }
  .visible-tablet {
    display: inherit !important;
  }
  .hidden-tablet {
    display: none !important;
  }
}
```

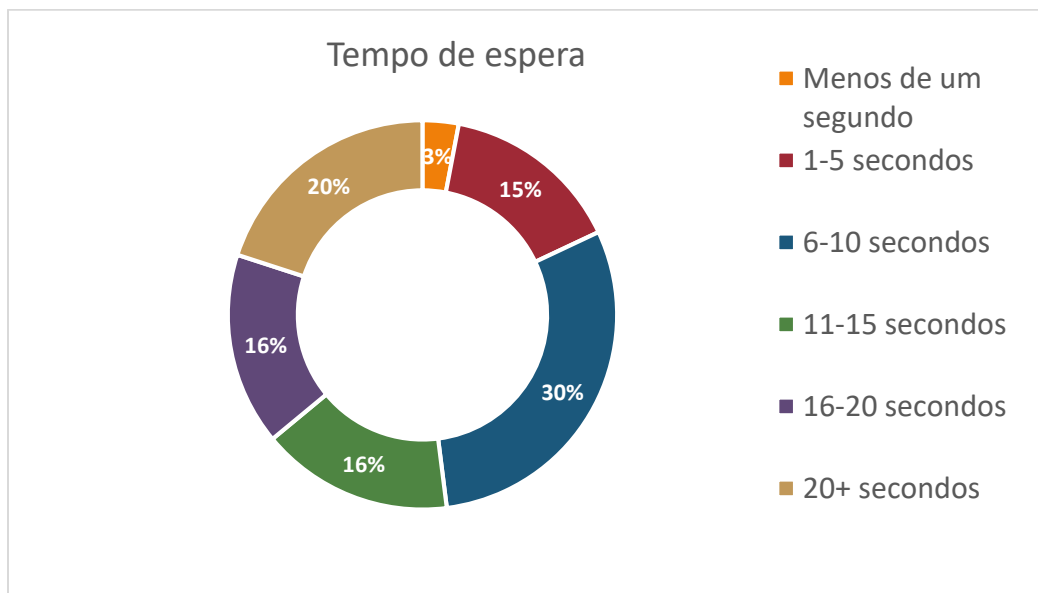
```
@media (min-width: 1200px) {
  .row {
    margin-left: -30px;
    *zoom: 1;
  }
  .row:before,
  .row:after {
    display: table;
    line-height: 0;
    content: "";
  }
  .row:after {
    clear: both;
  }
  [class*="span"] {
    float: left;
    min-height: 1px;
    margin-left: 30px;
  }
  .container,
  .navbar-static-top .container,
  .navbar-fixed-top .container,
  .navbar-fixed-bottom .container {
    width: 1170px;
  }
}
```

```
@media (min-width: 768px) and (max-width: 979px)
{
  .row {
    margin-left: -20px;
    *zoom: 1;
  }
  .row:before,
  .row:after {
    display: table;
    line-height: 0;
    content: "";
  }
  .row:after {
    clear: both;
  }
  [class*="span"] {
    float: left;
    min-height: 1px;
    margin-left: 20px;
  }
  .container,
  .navbar-static-top .container,
  .navbar-fixed-top .container,
  .navbar-fixed-bottom .container {
    width: 724px;
  }
}
```

3.4.4 Tempo de espera do carregamento da página

O tempo de espera do carregamento de uma página faz parte da experiência do utilizador e diversos estudos sugerem uma correlação directa com o tempo que os utilizadores se detêm num *website* e o seu grau de satisfação bem como a sua influência no *ranking* dos motores de busca. (Gardner, 2011)

A figura seguinte mostra o tempo que os usuários estão na disposição de esperar para uma página web para carregar. De acordo com pesquisas realizadas a maioria dos utilizadores abandonam um *website* se o tempo de espera para carregamento da página for superior a três segundos. E mesmo um segundo de atraso na resposta da página pode afectar o número de utilizadores. Para um *website e-commerce* que vende vários milhares de euros diariamente um atraso de um segundo, por exemplo, poderá custar vários milhões de euros perdidos em vendas anuais. (Work, 2011)



*Relação entre o tempo de carregamento da página e a paciência dos utilizadores.
A maioria dos utilizadores acha aceitável um tempo de resposta entre os 6 – 10 segundos.*

Garantir um óptimo desempenho para os utilizadores de dispositivos móveis é hoje fundamental, uma vez que estes já constituem mais de metade de todas as novas conexões de Internet. Para estes dispositivos surge a necessidade de tomar em conta a largura de banda reduzida, o aumento da latência nas redes móveis e o seu poder de processamento reduzido. (Weinstein, 2014)

Embora não resolvam totalmente este problema, algumas técnicas podem ser utilizados para melhorar a velocidade de uma aplicação web.

3.4.5 Redução de pedidos

Cada pedido (*request*) que uma página faz ao servidor para carregar uma imagem, script, ou um ficheiro CSS tem um custo de tempo associado. Este custo aumenta com dispositivos móveis em virtude da latência inerente às conexões móveis. Surpreendentemente, apesar de a velocidade de *download* poder ser até 10 vezes mais rápida nas redes 4G, a latência de conexão é idêntica entre as redes 3G e 4G.

Para reduzir os pedidos de uma página podemos:

- combinar ficheiros *CSS* e *JavaScript*,
- incorporar *binary data* para imagens em ficheiros *CSS* via *data URIs* e
- carregar imagens e conteúdo somente quando estão visíveis.

EXCERTO DO CÓDIGO UTILIZADO EM APPSTART PARA COMBINAR DIVERSOS FICHEIROS NUM ÚNICO (BUNDLING):

```
Dim bundles = BundleTable.Bundles
bundles.UseCdn = True
Dim jqueryCdnPath = "http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.10.0.min.js"

bundles.Add(New ScriptBundle("~/bundles/jquery", jqueryCdnPath).Include(
    "~/Scripts/jquery-{version}.js"))

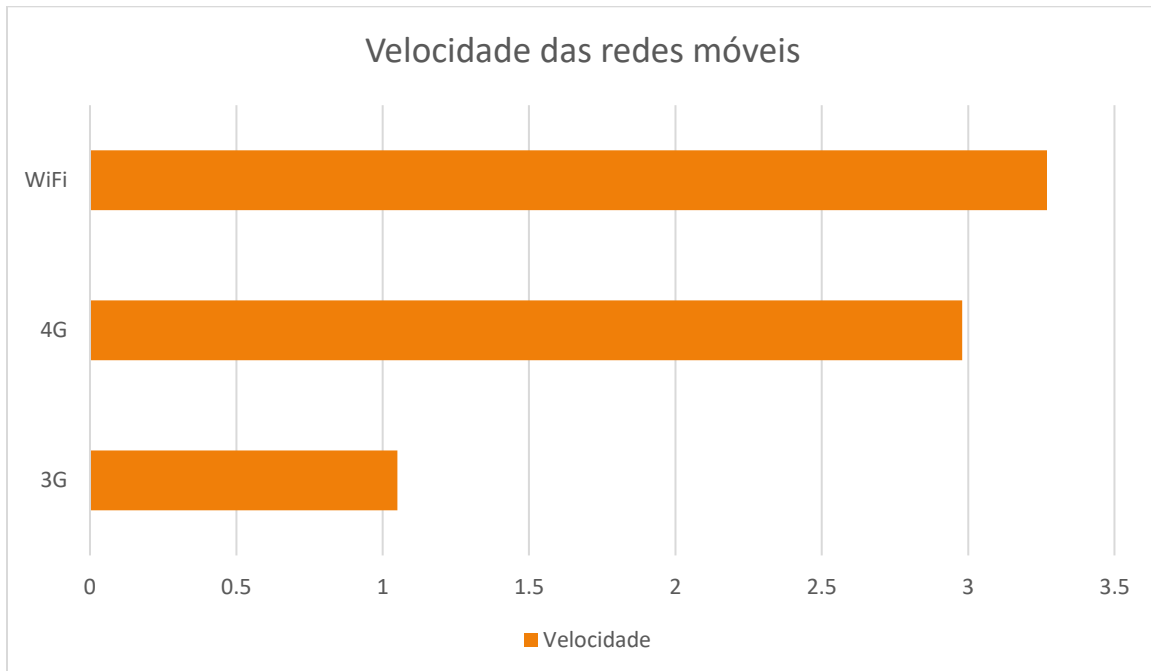
bundles.Add(New ScriptBundle("~/bundles/jqueryui").Include(
    "~/Scripts/jquery-ui-{version}.js"))

bundles.Add(New StyleBundle("~/bundles/css").Include(
    "~/Content/css/bootmetro.css",
    "~/Content/css/bootmetro-responsive.css",
    "~/Content/css/bootmetro-ui-light.css",
    "~/Content/css/datepicker.css",
    "~/Content/css/bootstrap-extras.css"))
```

Os *browsers* modernos limitam em seis o número de conexões simultâneas. *Bundling* é uma técnica utilizada em ASP.NET para combinar diversos ficheiros, reduzindo assim o número de conexões. Menos ficheiros significam menos pedidos de HTTP e conseqüentemente um melhoramento da performance de carregamento da página.

3.4.6 Redução do tamanho da página

Se menos dados forem transferidos para uma página, mais rápido será o *download* e a sua visualização. Atendendo que as redes 3G são cerca de 3 vezes mais lentas que as redes 4G e *wi-fi*, reduzir o tamanho da página pode ser crucial para reduzir o tempo de espera.



Algumas das técnicas que podemos utilizar são:

- usar *CSS3* em vez de imagens para elementos estilísticos, como os cantos arredondados, gradientes e sombras;
- redimensionar as imagens, em função do tamanho do ecrã;
- reduzir o tamanho da imagem, ajustando a compressão e qualidade;
- combinar ou imagens "*sprite*" para servir através de *CSS*; e
- utilizar técnicas de compressão para o código

EXEMPLO DE CÓDIGO VB UTILIZADO PARA REDIMENSIONAMENTO DE IMAGENS AJUSTANDO A COMPRESSÃO E QUALIDADE:

@Code

```
Dim db = Database.Open("MyMovies")
Dim photo = db.QuerySingle(strsql, id)
If photo Is Nothing Then
    Response.StatusCode = HttpStatusCode.NotFound
    Return
End If
Dim MyMoviesStorage As String = ConfigurationManager.AppSettings("MyMoviesStorage").ToString
Dim strpath = MyMoviesStorage & folder & photo.image & ".jpg"
strpath = Replace(strpath, "/", "\")
Dim cover() As Byte = File.ReadAllBytes(strpath)
Dim img = New WebImage(New MemoryStream(CType(cover, System.Byte())))
Dim h As Integer = img.Height
Dim w As Integer = img.Width
Dim MaxWidthThumb As Integer = 256
Dim MaxHeightThumb As Integer = 256
Dim MaxWidthNormal As Integer = 512
Dim MaxHeightNormal As Integer = 512
Dim NewWidth As Integer
Dim NewHeight As Integer
Dim size = If(Request("size"), "")
Select Case size.ToUpperInvariant()
    Case "SLIDER"
        NewWidth = 150
        NewHeight = 150
    Case "CAST"
        NewWidth = 100
        NewHeight = 100
    Case "SMALL"
        NewWidth = 200
        NewHeight = 200
    Case "MEDIUM"
        NewWidth = 500
        NewHeight = 400
    Case "LARGE"
        NewWidth = 625
        NewHeight = 625
    Case Else
End Select
img.Resize(NewWidth, NewHeight).Write()
End Code
```


EXEMPLO DE TÉCNICA DE COMPRESSÃO UTILIZADA EM ASP.NET (MINIFICATION):

Minification executa uma variedade de diferentes técnicas de código para *scripts* ou *CSS*, removendo espaços desnecessários, os comentários e encurtando os nomes de variáveis para um caracter. (Anderson, 2012)

Consideremos uma função *JavaScript*.

```
AddAltToImg = function (imageTagAndImageID, imageContext) {
  ///

```

Depois da *minification* a mesma função é reescrita como:

```
AddAltToImg = function (n, t) { var i = $(n, t); i.attr("alt", i.attr("id").replace(/ID/, "")) }
```

Além de remover os comentários e espaços desnecessários, os nomes das variáveis foram reescritos (reduzidos) como:

Original	Reduzido
imageTagAndImageID	n
imageContext	t
imageElement	i

Impacto do *Bundling e Minification*

A seguinte tabela mostra a diferença numa aplicação web entre o carregamento dos ficheiros individualmente e com a utilização de *Bundling e Minification* (B/M):

	Sem B/M	Com B/M	Diferença
Quantidade de ficheiros	34	9	256%
KB enviados	11.92	3.26	266%
KB recebidos	530	388.51	36%
Tempo de carregamento	780 ms	510 ms	53%

EXEMPLO DE SPRITE PARA SERVIR ATRAVÉS DE CSS



```

.icon {
  display: inline-block;
  width: 32px;
  height: 32px;
  fill: currentColor;
}
.icon-whatsnew {
  /*fill: #000000;*/
}
.icon-star {
  /*fill: #000000;*/
}
.icon-star2 {
  /*fill: #000000;*/
}
.icon-star3 {
  /*fill: #000000;*/
}
.icon-windows8 {
  /*fill: #000000;*/
}
.icon-windows {
  /*fill: #000000;*/
}

```

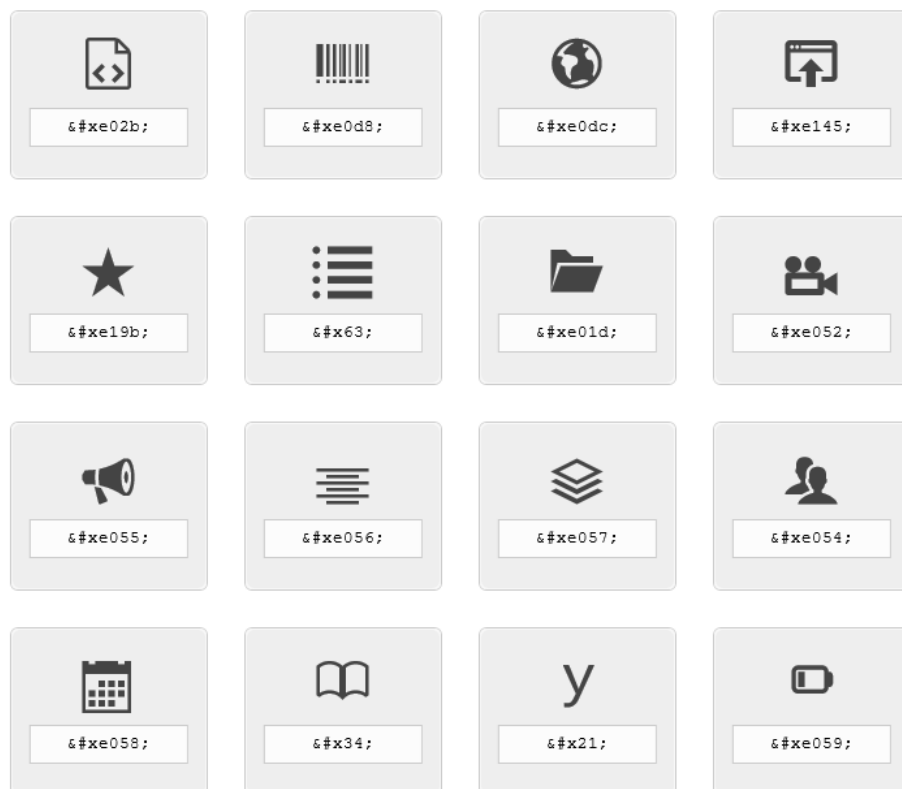
3.4.7 Design em função da velocidade

Quando a velocidade é uma característica fundamental da aplicação *web*, os *designers* são mais propensos a tomar decisões que optimizem o desempenho. Duas estratégias úteis que se podem seguir para aumentar a velocidade são:

- Considerar um **design minimalista**, evitando o uso de imagens sendo estas substituídas por elementos tipográficos e
- Desenvolver **fragmentos de páginas com requisitos mínimos** especificamente para o uso móvel, permitindo a remoção do código utilizado em interações com desktops

Design minimalista

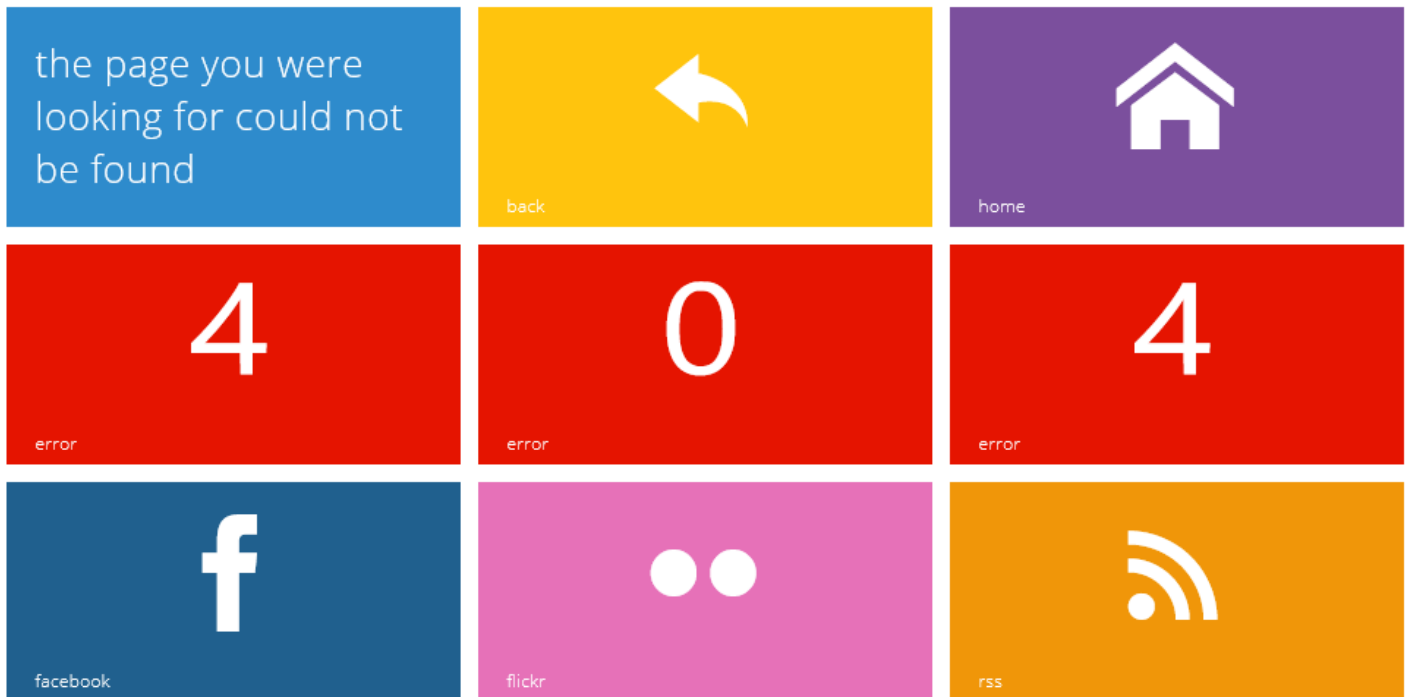
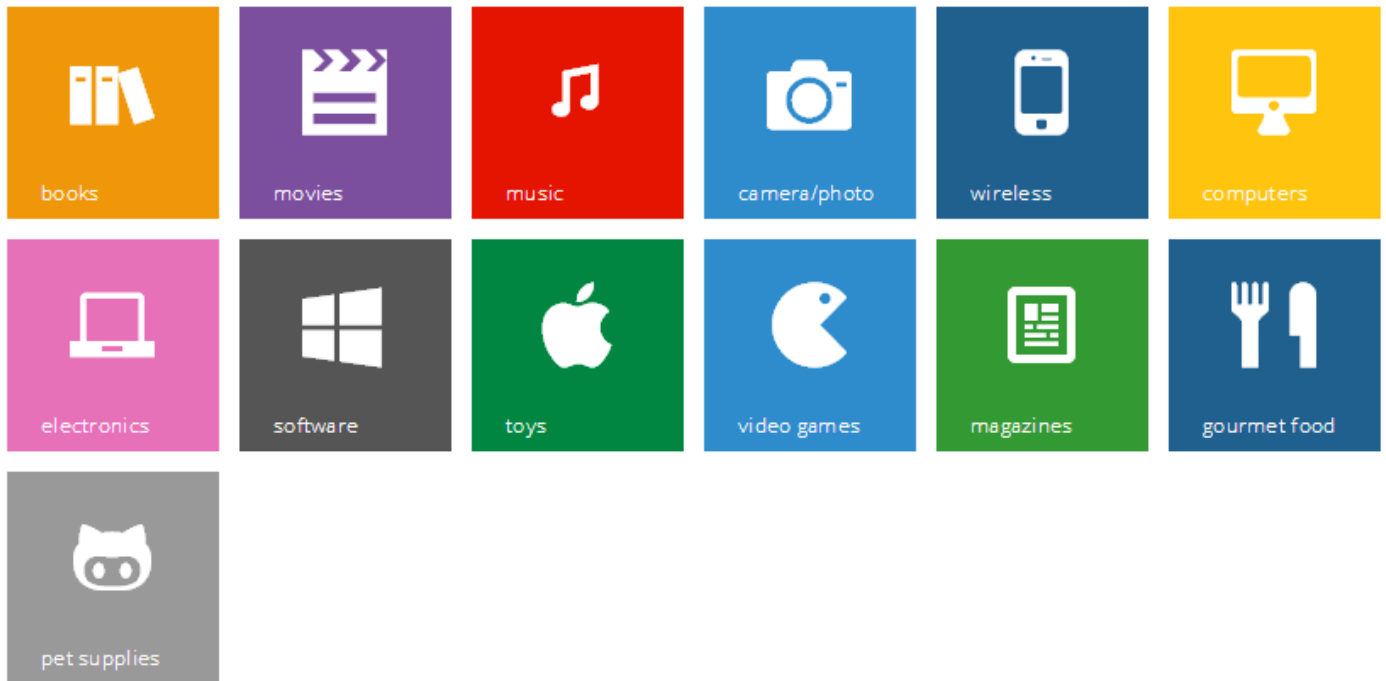
Recorrendo a elementos tipográficos como *fonts* de icons, ideais para utilização em aplicações móveis mas também adequados para qualquer outra aplicação, foi possível evitar o uso de imagens, reduzindo assim a transferência de dados, aumentando a velocidade e mantendo um visual atractivo.



A font de icons na figura utilizada na aplicação web foi criada com recurso a IcoMoon: <http://icomoon.io/>

my shop

all 



Páginas construídas recorrendo unicamente a elementos tipográficos.

Fragmentos de páginas com requisitos mínimos

Ao criar um fragmento da página principal especificamente para o uso móvel, podemos otimizar a visualização nestes dispositivos.

```
<div id="homedesktab">
  <div id="wrap">
    <div class="metro panorama">
      <div class="panorama-sections">
        <div class="panorama-section">
          <h2>my calibre library</h2>
          <div class="tile-column-span-8">
            @MyBooksHelpers.FormatHome(SiteMapTableMyBooks, mybooks_nodelist)
          </div>
          ' code removed for brevity.....
        </div>
      </div>
    </div>
  </div>
</div>
```

Fragmento utilizado para dispositivos desktop e tablets

```
<div id="homemobile">
  <div class="container">
    <div class="metro span12">
      <div class="row">
        <h2>my calibre library</h2>
        @MyBooksHelpers.FormatHome(SiteMapTableMyBooks, mybooks_nodelist)
      </div>
      ' code removed for brevity.....
    </div>
  </div>
</div>
```

Fragmento simplificado utilizado para dispositivos móveis

Teste e otimização

Várias ferramentas estão à disposição dos *designers* para medir o desempenho da aplicação web e observar o impacto das otimizações feitas para melhorar a velocidade. Estudando atentamente o impacto de todos os elementos da página, incluindo o número de pedidos, o tamanho dos ficheiros e o tempo de resposta podemos a melhorar as áreas que terão o maior impacto sobre a velocidade.

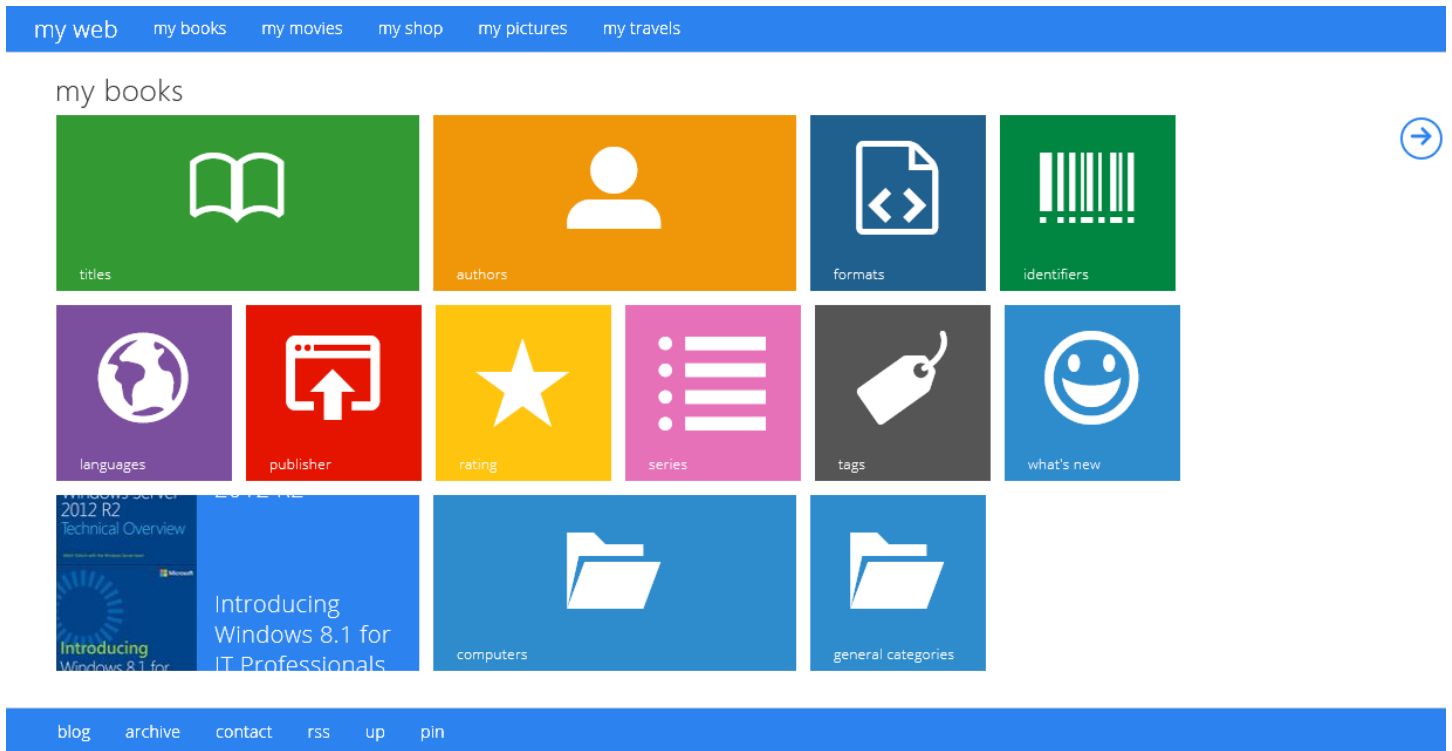
The screenshot shows the Google Developers PageSpeed Insights interface. At the top, the Google Developers logo is visible. Below it, the breadcrumb 'Products > PageSpeed Insights' is shown. The main heading is 'PageSpeed Insights' with a '+1' rating and '18k' reviews. A search bar contains the URL 'myweb2.simple-url.com' and an 'ANALYZE' button. Below the search bar, there are two tabs: 'Mobile' (selected) and 'Desktop'. The main content area displays a green box with '100 / 100 User Experience'. Below this, a green checkmark icon is followed by the text 'Congratulations! No issues found.' The page lists several optimization suggestions, all of which are satisfied: 'Avoid plugins', 'Configure the viewport', 'Size content to viewport', 'Size tap targets appropriately', and 'Use legible font sizes'. To the right of the text, a smartphone mockup displays a mobile application interface with a blue header 'home', a search bar, and a list of items: 'my books', 'titles', 'authors', 'formats', and 'identifiers'.

Algumas ferramentas que podemos utilizar são:

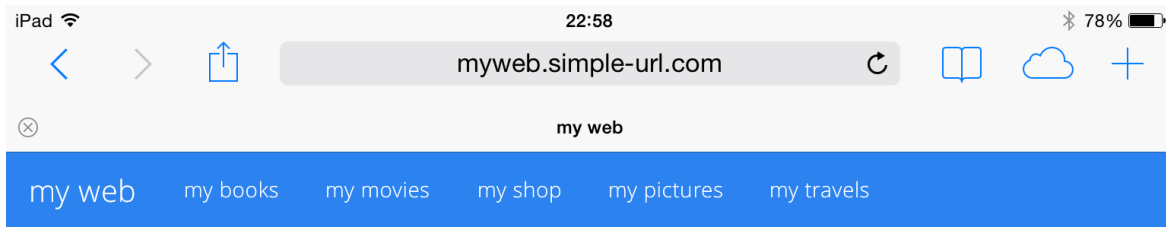
- **PageSpeed** (<http://code.google.com/speed/page-speed/>),
- **yslow** (<http://developer.yahoo.com/yslow/>).

(Gardner, 2011)

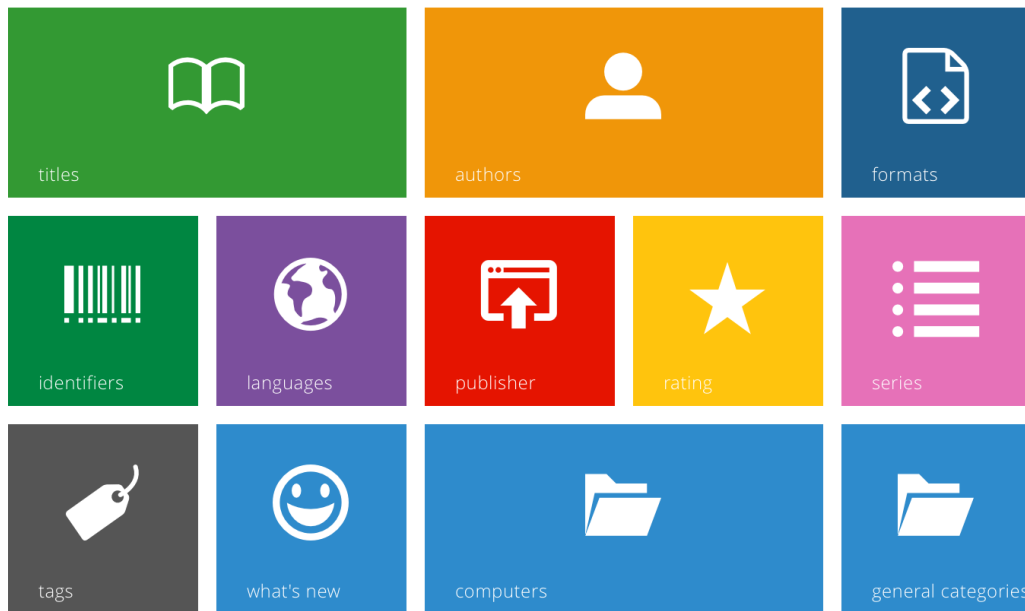
3.4.8 Versão Desktop



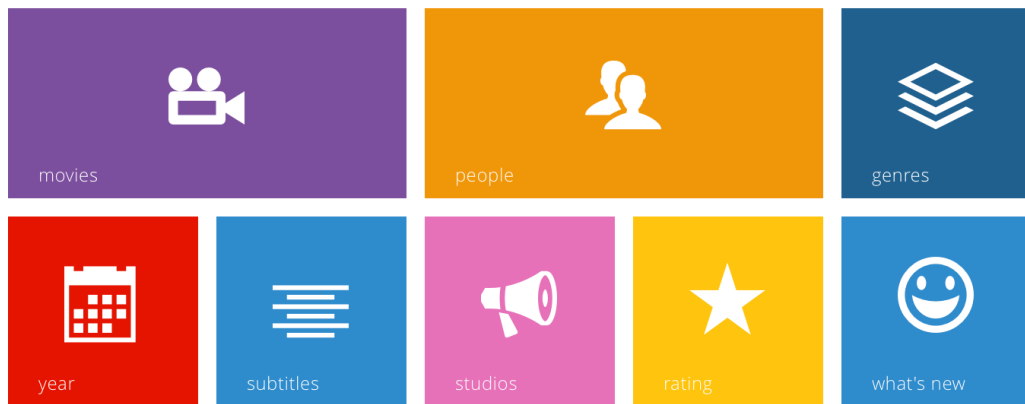
3.4.9 Versão Tablet



my books



my movies

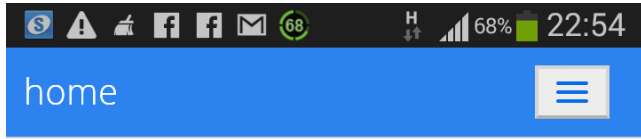


my shop

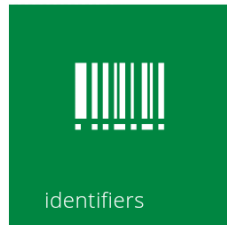


iOS 7 (Ipad)

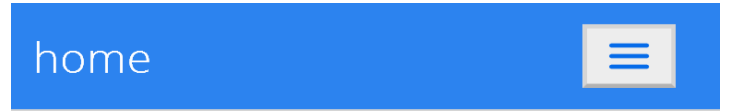
3.4.10 Versão Mobile



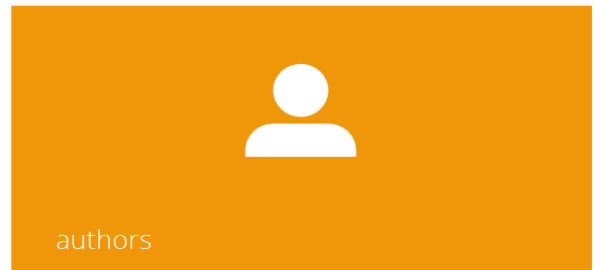
my books



Android (Galaxy S3)



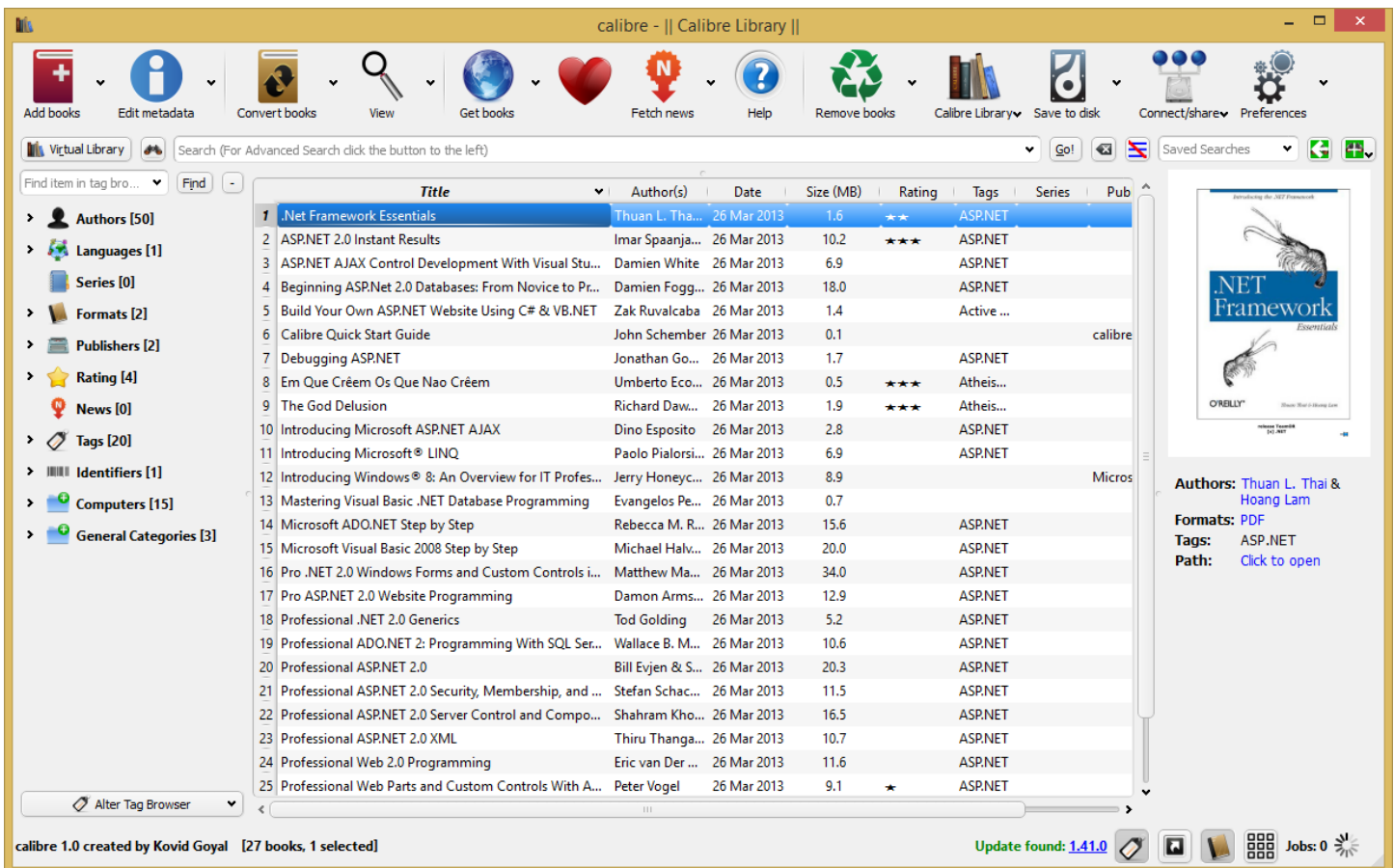
my books



WiondowsPhone (Lumia 920)

3.5 MÓDULO MY BOOKS

Módulo destinado a inventariar a colecção de livros do utilizador, com armazenamento de dados em *SQLite* e *Calibre e-book Manager*.



O Calibre é um gestor de bibliotecas virtuais, *open source*, desenvolvido principalmente em *Python*, com uma infinidade de recursos relativos a *e-books*. (<http://calibre-ebook.com/>)

Quando o módulo *mybooks* corre pela primeira vez os seus dados são carregados a partir da base de dados *SQLite* do *Calibre*, invocando o método **GetListForSiteMapMyBooks** que cria uma *datatable* contendo todas as informações das categorias e sub-categorias necessárias à construção da navegação do módulo.

Além das categorias definidas em *App_Data\categories.xml*, é necessário fazer a leitura das categorias e sub-categorias definidas pelo utilizador que constam na base de dados do *Calibre*. Estes dados estão guardados numa string JSON, localizada na tabela *preferences*, na coluna *val* correspondendo ao campo *user_categories* da coluna *key*.

The screenshot shows the SQLite Manager interface. The left sidebar displays the database structure for 'metadata.db', with the 'preferences' table selected. The main window shows the SQL query 'SELECT * FROM preferences' and the resulting table data. The 'user_categories' row is highlighted in blue.

id	key	val
1	saved_searches	{}
2	book_display_fields	[["title", false], ["authors", true], ["formats", true ...
3	namespaced:ViewManagerPlugin:settings	{ "viewToApply": "Last View Used", "autoApplyView": false, "Sch...
7	column_color_rules	{}
8	grouped_search_terms	{}
12	grouped_search_make_user_categories	{}
13	namespaced:QualityCheckPlugin:settings	{ "SchemaVersion": 1.9, "exclusionsByCheck": {} }
14	bools_are_tristate	true
180	virtual_libraries	{ "Shelf": "series\ "= Shelf"" }
181	user_categories	{ "Computers": [["Adobe", "tags", 10630], ["...
189	namespaced:FindDuplicatesPlugin:settings	{ "bookExemptions": [], "SchemaVersion": 1.5, "lastLibraryCompar...
355	news_to_be_synced	{}
363	tag_browser_hidden_categories	{}
365	gui_view_history	[[4535, "Pro HTML5 With Visual Studio 2012"], [5381, "S...
366	library_view books view state	{ "column_alignment": { "timestamp": "center", "pubdate": "cen...
367	field_metadata	{ "rating": { "is_category": true, "is_csp": false, "search_terms": ...

O *SQLite Manager* é um plugin para o *Firefox* que permite gerenciar qualquer base de dados *SQLite*. (<https://addons.mozilla.org/en-US/firefox/addon/sqlite-manager/>)

Após efectuarmos a leitura da string *JSON* com ajuda da biblioteca *Newtonsoft (Json.NET)*, <http://james.newtonking.com/json>) convertemos a mesma num objecto.

```
Dim commandText As String = "SELECT val FROM preferences Where key = 'user_categories'"
Dim json As String = ExecuteScalarSQLite(commandText,ConnectionStrings("MyBooks"))
Dim jsonObj As JObject = JObject.Parse(json)
json = jsonObj.ToString(Newtonsoft.Json.Formatting.None)
json = Replace(json, " ", ".")
```

Este objecto é em seguida convertido num ficheiro XML com ajuda do método **ReadJSON**:

```
Dim doc As XmlDocument = New XmlDocument()
doc = DirectCast(JsonConvert.DeserializeXmlNode(json, "rootcat"), XmlDocument)
Dim writer As XmlTextWriter = New
XmlTextWriter(HttpContext.Current.Server.MapPath("~/App_Data/UserCategories-input.xml"), Nothing)
writer.Formatting = System.Xml.Formatting.Indented
doc.Save(writer)
writer.Close()
```

E o resultado é o ficheiro *UserCategories.xml* que guardamos dentro de *App_Data*:

```
<rootcat>
  <Computers.Web..Design>
    <Computers.Web..Design>Web..Servers</Computers.Web..Design>
    <Computers.Web..Design>tags</Computers.Web..Design>
    <Computers.Web..Design>62</Computers.Web..Design>
  </Computers.Web..Design>
  <Computers.Web..Design>
    <Computers.Web..Design>Web..Site..Development</Computers.Web..Design>
    <Computers.Web..Design>tags</Computers.Web..Design>
    <Computers.Web..Design>61</Computers.Web..Design>
  </Computers.Web..Design>
  <Computers.Web..Design>
    <Computers.Web..Design>Web..Sites</Computers.Web..Design>
    <Computers.Web..Design>tags</Computers.Web..Design>
    <Computers.Web..Design>55</Computers.Web..Design>
  </Computers.Web..Design>
  <Computers>
    <Computers>Active..Server..Pages</Computers>
    <Computers>tags</Computers>
    <Computers>52</Computers>
  </Computers>
  <Computers>
    <Computers>ASP.NET</Computers>
    <Computers>tags</Computers>
    <Computers>58</Computers>
  </Computers>
  <Computers>
    <Computers>C#</Computers>
    <Computers>tags</Computers>
    <Computers>72</Computers>
```

É a partir da leitura deste ficheiro *XML* que obtemos a *DataTable* que serve de informação ao *custom site map provider* utilizada na navegação ASP.NET do módulo my books:

ID	Title	CatType	Url	Parentstr	Parent
0	Home	rootcat	/mybooks/browse/		
1	Adobe	tags	/mybooks/list/tags/302	Computers	50
2	AI	tags	/mybooks/list/tags/1480	Computers	50
3	ASP.NET	tags	/mybooks/list/tags/1	Computers	50
4	Assembly	tags	/mybooks/list/tags/869	Computers	50
5	BSD	tags	/mybooks/list/tags/260	Computers	50
6	C	tags	/mybooks/list/tags/1582	Computers	50
7	C#	tags	/mybooks/list/tags/5	Computers	50
8	Google	tags	/mybooks/list/tags/1556	Computers	50
9	Hacking	tags	/mybooks/list/tags/3096	Computers	50
10	Java	tags	/mybooks/list/tags/1145	Computers	50
..... data removed for brevity					
41	Linguistics	tags	/mybooks/list/tags/223	Categories	51
42	Mathematics	tags	/mybooks/list/tags/265	Categories	51
43	Philosophy	tags	/mybooks/list/tags/89	Categories	51
44	Photography	tags	/mybooks/list/tags/19	Categories	51
45	Physics	tags	/mybooks/list/tags/231	Categories	51
46	Politics	tags	/mybooks/list/tags/91	Categories	51
47	Romance	tags	/mybooks/list/tags/272	Categories	51
48	Spies	tags	/mybooks/list/tags/189	Categories	51
49	Vampires	tags	/mybooks/list/tags/252	Categories	51
50	Computers	usercat	/mybooks/browse/usercat/50	Home	0
51	Categories	usercat	/mybooks/browse/usercat/51	Home	0
52	Titles	calcat	/mybooks/browse/titles	Home	0
53	Authors	calcat	/mybooks/browse/authors	Home	0
54	Formats	calcat	/mybooks/browse/formats	Home	0
55	Identifiers	calcat	/mybooks/browse/identifiers	Home	0
56	Languages	calcat	/mybooks/browse/languages	Home	0
57	Publisher	calcat	/mybooks/browse/publisher	Home	0
58	Rating	calcat	/mybooks/browse/rating	Home	0
59	Series	calcat	/mybooks/browse/series	Home	0
60	Tags	calcat	/mybooks/browse/tags	Home	0

Esta *DataTable* é obtida invocando o método **GetListForSiteMapMyBooks**

```
Dim doc As New XmlDocument()
doc.Load(HttpContext.Current.Server.MapPath("~/App_Data/UserCategories.xml"))
' Declare variables for DataColumn and DataRow objects.
Dim column As DataColumn
Dim row As DataRow
' Add the new DataTable to the DataSet.
dataSet.Tables.Add(table)
row = table.NewRow()
row("ID") = 0
row("Title") = "Home"
row("CatType") = "rootcat"
row("Url") = "/mybooks/browse/"
row("Parentstr") = ""
table.Rows.Add(row)
```

Primeiro adicionamos à *DataTable* as categorias nativas do *Calibre*, tais como *authors*, *publishers*, *series*, *tags*:

```
Dim root As XmlNode = doc.FirstChild
Dim i As Integer
Dim _CatType As String = ""
Dim str As String = "authors,publisher,series,tags"
Dim _CatTypes() As String = str.Split(",")
'Display the contents of the child nodes.
If root.HasChildNodes Then
    For i = 0 To root.ChildNodes.Count - 1
        row = table.NewRow()
        row("ID") = i + 1
        For Each c As String In _CatTypes
            If InStr(root.ChildNodes(i).InnerText, c) > 0 Then
                _CatType = c
            End If
        Next
        Dim CatArray() As String = Split(root.ChildNodes(i).InnerText, _CatType)
        row("Title") = Replace(CatArray(0).ToString(), "..", " ")
        row("CatType") = _CatType
        Dim tablename As String = _CatType
        Dim Id As String = ExecuteScalarSQLite("Select Id FROM " & tablename & " WHERE name = " & ...
        row("Url") = "/mybooks/list/" & _CatType & "/" & Id
        row("Parentstr") = Replace(root.ChildNodes(i).LocalName.ToString(), "..", " ")
        table.Rows.Add(row)
    Next i
End If
```

De seguida, adicionamos as sub-categorias definidas pelo utilizador:

```

Dim view As New DataView(table)
Dim distinctValues As DataTable = view.ToTable(True, "Parentstr")
Dim foundRows() As DataRow
' Use the Select method to find all rows.
foundRows = distinctValues.Select()
Dim y As Integer
Dim position As Integer
Dim cat As String
Dim catparent As String
For y = 1 To foundRows.GetUpperBound(0)
    cat = foundRows(y)(0)
    position = InStrRev(cat, ".", -1, vbTextCompare)
    If position > 0 Then
        catparent = Left(cat, position - 1)
    Else
        catparent = "Home"
    End If
    row = table.NewRow()
    i = i + 1
    row("ID") = i
    row("Title") = cat
    row("CatType") = "usercat"
    row("Url") = "/mybooks/browse/usercat/" & i
    row("Parentstr") = catparent
    table.Rows.Add(row)
Next y

```

E por último as categorias de raiz do Calibre:

```

Dim _CatRoot() As String = XMLReturnArray("mybooks", "Title")
For Each _cat As String In _CatRoot
    row = table.NewRow()
    i = i + 1
    row("ID") = i
    row("Title") = StrConv(_cat, VbStrConv.ProperCase)
    row("CatType") = "calcat"
    row("Url") = "/mybooks/browse/" & _cat
    row("Parentstr") = "Home"
    table.Rows.Add(row)
Next

```

Finalmente guardamos a *DataTable* em *cache*:

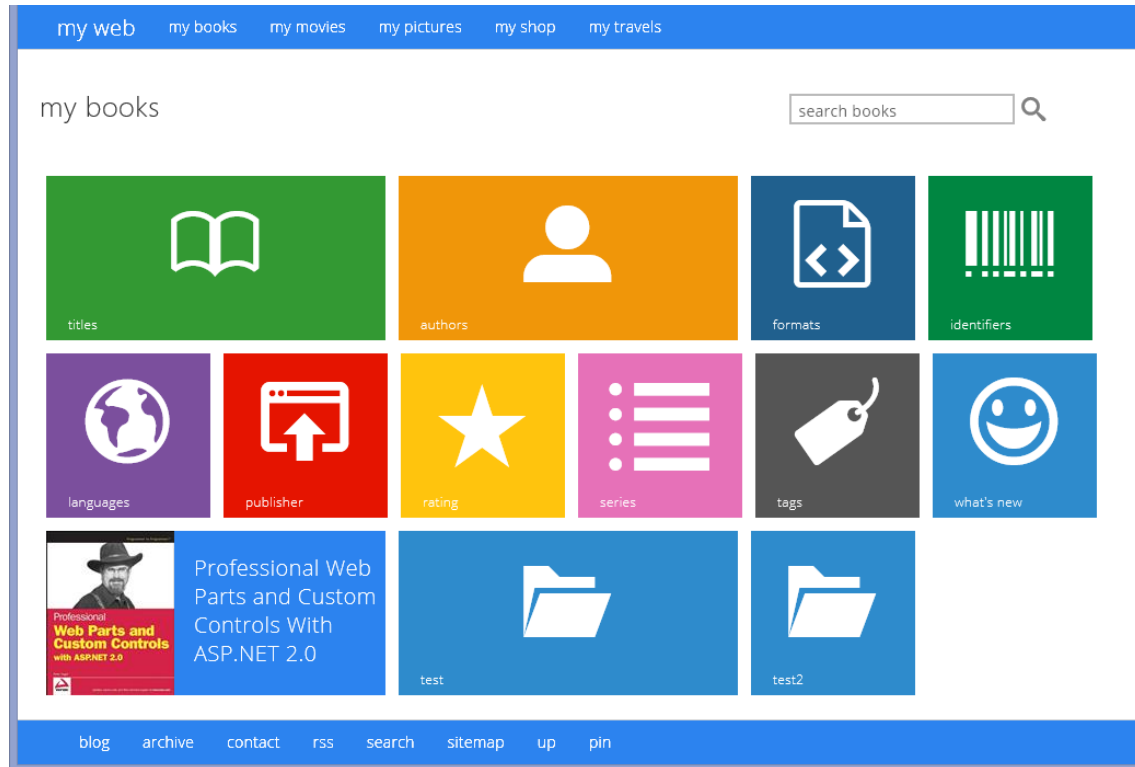
```

System.Web.HttpContext.Current.Cache.Insert(cachekey, table, Nothing,
System.Web.Caching.Cache.NoAbsoluteExpiration, TimeSpan.FromMinutes(CacheTimeSpan))

```


E está pronta a ser utilizado na construção dos menus, como vemos na página principal do módulo my books:

Menu my books



Lista de livros

my web my books my movies my pictures my shop my travels

publisher :: microsoft press

home / my books / publisher / microsoft press

Programming Windows® 8 Apps With HTML, CSS, and JavaScript
 Apply your existing skills with HTML, CSS, and JavaScript—and start building your own Windows 8 apps now. As a member of the Windows Ecosystem team, the author has trained hundreds of Microsoft engineers and has been on the front lines of bringing the first Windows 8 apps to the Windows Store. Through this book, you'll get a thorough grounding in platform features and considerations, and delve into development essentials across the whole scope of the Windows 8 platform. And you'll gain insights and best p... [Read More](#)

Introducing Windows® 8: An Overview for IT Professionals
 Get a head start evaluating Window 8. Based on the Release Preview of the software, this guide introduces new features and capabilities with

blog archive contact rss search sitemap up pin

Detalhes do livro

my web my books my movies my pictures my shop my travels

programming windows® 8 apps with html, css, and javascript

home / my books / programming windows® 8 apps with html, css, and javascript

formats: EPUB
 author sort: Brockschmidt, Craig
 authors: [Craig Brockschmidt]
 date: 2013 Mar 26
 IDs: isbn
 languages: eng
 modified: 2013 Mar 26
 published: 2012 Nov
 publisher: Microsoft Press
 rating: 5

blog archive contact rss search sitemap up pin

3.5.1 Classe mybooks

```
Imports Microsoft.VisualBasic
Imports System
Imports System.Web
Imports System.Drawing
```

```
Imports System.Drawing.Imaging
Imports System.Drawing.Drawing2D
Imports System.Data
Imports System.IO
Imports System.Web.UI.WebControls
Imports System.Data.SQLite
Imports System.Text.RegularExpressions
Imports CalibreVBDataSetTableAdapters
Imports System.Net
Imports LibCommon
```

```
Public Class LibMyBooks
```

```
'Returns a datatable with the book formats
```

```
Public Shared Function GetFormats(ByVal id As Integer) As DataTable
    Dim formatsbybookAdapter As New formatsbybookTableAdapter
    Dim formatsbybookTable As New CalibreVBDataSet.formatsbybookDataTable
    Try
        formatsbybookTable = formatsbybookAdapter.GetFormatsByBook(id)
    Catch ex As Exception
    End Try
    Return formatsbybookTable
End Function
```

```
'Returns a datatable with the book authors
```

```
Public Shared Function GetAuthors(ByVal id As Integer) As DataTable
    Dim authorsbybookAdapter As New authorsbybookTableAdapter
    Dim authorsbybookTable As New CalibreVBDataSet.authorsbybookDataTable
    Try
        authorsbybookTable = authorsbybookAdapter.GetAuthorsByBook(id)
    Catch ex As Exception
    End Try
    Return authorsbybookTable
End Function
```

```
'Returns a datatable with the book identifiers
```

```
Public Shared Function GetIds(ByVal id As Integer) As DataTable
    Dim idsbybookAdapter As New idsbybookTableAdapter
    Dim idsbybookTable As New CalibreVBDataSet.idsbybookDataTable
    Try
        idsbybookTable = idsbybookAdapter.GetIdsByBook(id)
    Catch ex As Exception
    End Try
    Return idsbybookTable
End Function
```

```
'Returns a datatable with the book tags
```

```
Public Shared Function GetTags(ByVal id As Integer) As DataTable
```

```

Dim tagsbybookAdapter As New tagsbybookTableAdapter
Dim tagsbybookTable As New CalibreVBDataSet.tagsbybookDataTable
Try
    tagsbybookTable = tagsbybookAdapter.GetTagsByBook(id)
Catch ex As Exception
End Try
Return tagsbybookTable
End Function

```

'Returns a datatable with the book series

```

Public Shared Function GetSeries(ByVal id As Integer) As DataTable
    Dim seriesbybookAdapter As New seriesbybookTableAdapter
    Dim seriesbybookTable As New CalibreVBDataSet.seriesbybookDataTable
    Try
        seriesbybookTable = seriesbybookAdapter.GetSeriesByBook(id)
    Catch ex As Exception
    End Try
    Return seriesbybookTable
End Function

```

'Returns a string with the book description

```

Public Shared Function GetComments(ByVal id As Integer) As String
    Dim commentsTableAdapter As New CalibreVBDataSetTableAdapters.commentsbybookTableAdapter()
    Dim comments As String = CType(commentsTableAdapter.ScalarGetCommentsByBook(id), String)
    If Not comments = String.Empty Then
        comments = StripAllTags(comments)
        Dim GetTitlebyBook As New CalibreVBDataSetTableAdapters.GetTitlebyBook()
        Dim title As String = StripSlug(CType(GetTitlebyBook.ScalarGetTitlebyBook(id), String))
        comments = GetReadMore(comments.ToString, 512, "mybooks", id, title)
    End If
    Return comments
End Function

```

'Returns a string with the book rating

```

Public Shared Function GetRating(ByVal id As Integer) As String
    Dim ratingsTableAdapter As New CalibreVBDataSetTableAdapters.ratingsbybookTableAdapter()
    Dim rating As Object = CType(ratingsTableAdapter.ScalarGetRatingsByBook(id), Object)
    Dim ratingValue As Integer
    If Not (rating Is Nothing) Then
        If Not rating.ToString = String.Empty Then
            ratingValue = CType(rating / 2, String)
        End If
    End If
    Return ratingValue.ToString
End Function
End Class

```

3.6 MÓDULO MY MOVIES

Módulo destinado a inventariar a colecção de filmes do utilizador, desenvolvido com recurso a código do aluno, armazenamento de dados em *SQL Server*, e *My Movies Movie Manager*.

(<http://www.mymovies.dk/>)

À semelhança do módulo anterior, quando o módulo my movies corre pela primeira vez os seus dados são carregados a partir do ficheiro *categories.xml*, ao invocarmos o método **GetListForSiteMapMyMovies** que também cria uma *datatable* contendo todas as informações das categorias necessárias à construção da navegação do módulo.

ID	Title	CatType	Url	Parentstr	Parent
0	Home	rootcat	/mymovies/browse/		
1	Movies	calcat	/mymovies/browse/movies	Home	0
2	People	calcat	/mymovies/browse/people	Home	0
3	Genres	calcat	/mymovies/browse/genres	Home	0
4	Year	calcat	/mymovies/browse/year	Home	0
5	Subtitles	calcat	/mymovies/browse/subtitles	Home	0
6	Studios	calcat	/mymovies/browse/studios	Home	0
7	Rating	calcat	/mymovies/browse/rating	Home	0

No entanto, este método é muito mais simples pois a únicas categorias utilizadas são as de raí do My Movies Manager, uma vez que este não permite a criação de outras categorias pelo utilizador.

```

Dim view As New DataView(table)
Dim distinctValues As DataTable = view.ToTable(True, "Parentstr")
Dim foundRows() As DataRow
' Use the Select method to find all rows.
foundRows = distinctValues.Select()
Dim _CatRoot() As String = XMLReturnArray("mymovies", "Title")
For Each _cat As String In _CatRoot
    row = table.NewRow()
    i = i + 1
    row("ID") = i
    row("Title") = StrConv(_cat, VbStrConv.ProperCase)
    row("CatType") = "calcat"
    row("Url") = "/mymovies/browse/" & _cat
    row("Parentstr") = "Home"
    table.Rows.Add(row)
Next

```

E de seguida guardamos a *DataTable* em *cache*:

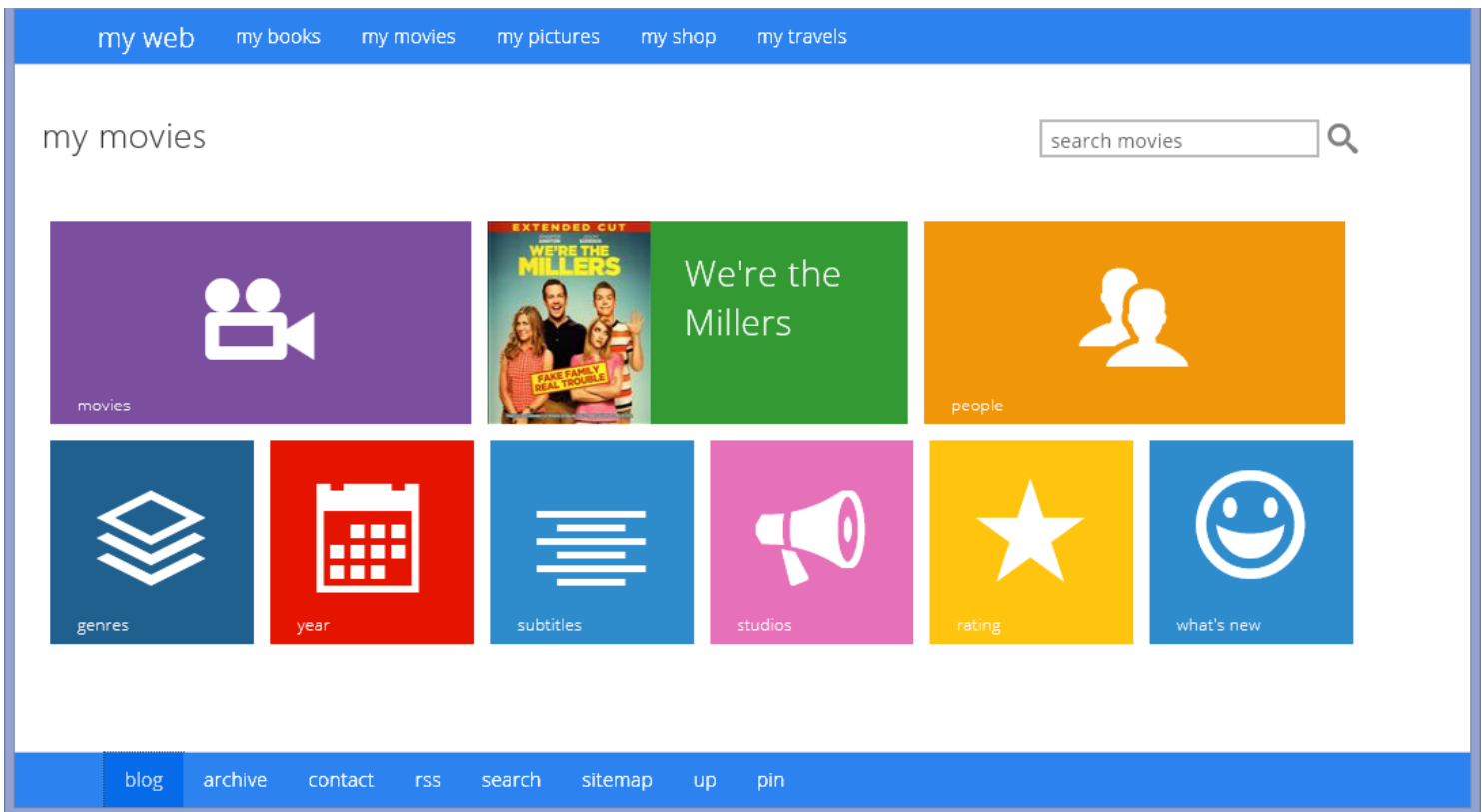
```

System.Web.HttpContext.Current.Cache.Insert(cachekey, table, Nothing,
System.Web.Caching.Cache.NoAbsoluteExpiration, TimeSpan.FromMinutes(CacheTimeSpan))

```

Pronta a ser utilizada na construção do menu da página principal do módulo my movies:

Menu my movies



Lista de filmes

my web my books my movies my pictures my shop my travels

movies :: a

home / my movies / movies / a

1 2 3 4

BLU-RAY + DIGITAL HD
ROBERT REDFORD
ALL IS LOST
"A MASTERPIECE!"
"A TRIUMPH."
"AMAZING."
FROM THE OSCAR-NOMINATED WRITER OF MARGHERITA

All Is Lost [2013]

Action

Robert Redford

Lionsgate Films

4

BLU-RAY + DVD + DIGITAL HD
FROM THE DIRECTOR OF LOVE ACTUALLY
ABOUT TIME

About Time [2013]

The night after another unsatisfactory New Year party, Tim's (Domhnall Gleeson) father (Bill Nighy) tells his son that the men in his family have always had the ability to travel through time. Tim can't change history, but he can change what happens and has happened in his own life - so he decides to make his world a better place...by getting a girlfriend (Rachel Watson). But as his unusual life progresses, Tim finds out that

Comedy

Graham Curry

blog archive contact rss search sitemap up pin

Detalhes do filme

my web my books my movies my pictures my shop my travels

about time

home / my movies / about time

genres: [Science-Fiction] [Comedy]
studios: [Universal Pictures] [Relativity Media] [Working title production] [Universal Studios Home Entertainment]
year: 2013
updated: 2014 Jan 31
dvd release: 2014 Apr
rating: 4
id's: [imdb] [rotten tomatoes]

[watch trailer](#)

BLU-RAY + DVD + DIGITAL HD
FROM THE DIRECTOR OF LOVE ACTUALLY
ABOUT TIME
"SWIFT, SENTIMENTAL, FUNNY AND TOTALLY CHARMING."

HD PICTURE AND THEATER QUALITY SOUND
"PRESENTS ONE TO MANDY PATTON AND OTHER ACTORS"
DIGITAL HD
UNIVERSAL PICTURES PRESENTS
A RELATIVITY MEDIA PRODUCTION
"ABOUT TIME"
STARRING DOMHNALL GLEESON, BILL NIHY, RACHEL WATSON
DIRECTED BY ROBERT SWICK

blog archive contact rss search sitemap up pin

3.6.1 Classe mymovies

```
Imports Microsoft.VisualBasic
Imports System
Imports System.Web
Imports System.Drawing
Imports System.Drawing.Imaging
Imports System.Drawing.Drawing2D
Imports System.Data
Imports System.Data.SqlClient
Imports System.IO
Imports System.Web.UI.WebControls
Imports System.Text.RegularExpressions
Imports MyMoviesTableAdapters
Imports LibCommon
```

```
Public Class LibMyMovies
```

```
    'Returns a datatable with the movie studios
```

```
    Public Shared Function GetStudios(ByVal id As Integer) As DataTable
        Dim StudiosbytitleAdapter As New StudiosbytitleTableAdapter
        Dim StudiosbytitleTable As New MyMovies.StudiosbytitleDataTable
        Try
            StudiosbytitleTable = StudiosbytitleAdapter.GetStudiosbytitle(id)
        Catch ex As Exception
        End Try
        Return StudiosbytitleTable
    End Function
```

```
    'Returns a datatable with the movie genres
```

```
    Public Shared Function GetGenres(ByVal id As Integer) As DataTable
        Dim GenresbytitleAdapter As New GenresbytitleTableAdapter
        Dim GenresbytitleTable As New MyMovies.GenresbytitleDataTable
        Try
            GenresbytitleTable = GenresbytitleAdapter.GetGenresbytitle(id)
        Catch ex As Exception
        End Try
        Return GenresbytitleTable
    End Function
```

```
    'Returns a datatable with the first five cast
```

```
    Public Shared Function Get5Persons(ByVal id As Integer) As DataTable
        Dim PersonsbytitleAdapter As New PersonsbytitleTableAdapter
        Dim PersonsbytitleTable As New MyMovies.PersonsbytitleDataTable
        Try
            PersonsbytitleTable = PersonsbytitleAdapter.Get5Personsbytitle(id)
        Catch ex As Exception
```

```
End Try
Return PersonsbytitleTable
End Function
```

'Returns a datatable with the movie cast

```
Public Shared Function GetCast(ByVal id As Integer) As DataTable
    Dim CastCrewAdapter As New CastCrewTableAdapter
    Dim CastCrewTable As New MyMovies.CastCrewDataTable
    Try
        CastCrewTable = CastCrewAdapter.GetCast(id)
    Catch ex As Exception
    End Try
    Return CastCrewTable
End Function
```

'Returns a datatable with the movie crew

```
Public Shared Function GetCrew(ByVal id As Integer) As DataTable
    Dim CastCrewAdapter As New CastCrewTableAdapter
    Dim CastCrewTable As New MyMovies.CastCrewDataTable
    Try
        CastCrewTable = CastCrewAdapter.GetCrew(id)
    Catch ex As Exception
    End Try
    Return CastCrewTable
End Function
```

'Returns a datatable with the movie crew data

```
Public Shared Function GetCrewData(ByVal id As Integer) As DataTable
    Dim CrewAdapter As New CrewTableAdapter
    Dim CrewTable As New MyMovies.CrewDataTable
    Try
        CrewTable = CrewAdapter.GetCrewData(id)
    Catch ex As Exception
    End Try
    Return CrewTable
End Function
```

'Returns a string with the movie description

```
Public Shared Function GetDescription(ByVal id As Integer) As String
    Dim DescriptionTableAdapter As New MyMoviesTableAdapters.ScalarGetValueByTitleIDTableAdapter()
    Dim Description As String = CType(DescriptionTableAdapter.ScalarGetDescriptionByTitleID(id), String)
    If Not Description = String.Empty Then
        Description = StripAllTags(Description)
        Dim title As String = StripSlug(CType(DescriptionTableAdapter.ScalarGetTitleByTitleID(id), String))
        Description = GetReadMore(Description.ToString, 756, "mymovies", id, title)
    End If
    Return Description
End Function
```

```
'Returns a string with the movie crew role description
Public Shared Function CrewFunction(ByVal str As String) As String
    Dim fname = ""
    Select Case str
        Case "2"
            fname = "Director"
        Case "3"
            fname = "Writer"
        Case "4"
            fname = "Producer"
        Case "5"
            fname = "Cinematography"
        Case "6"
            fname = "Editor"
        Case "7"
            fname = "Music"
        Case Else
            fname = "Crew"
    End Select
    Return fname
End Function
```

```
'Returns a string with the movie crew roles
Public Shared Function CrewParse(ByVal str As String) As String
    Dim strArray() As String
    Dim strRoles As String = ""
    Dim i As Integer
    strArray = Split(str, ",")
    If UBound(strArray) > 0 Then
        For i = LBound(strArray) To UBound(strArray)
            strRoles = strRoles & CrewFunction((strArray(i))) & ", "
        Next
        strRoles = Left(strRoles, Len(strRoles) - 2)
    Else
        strRoles = CrewFunction(str)
    End If
    Return strRoles
End Function
```

```
End Class
```

3.7 MÓDULO MY PICS

Módulo destinado a inventariar a colecção de fotografias do utilizador, armazenamento de dados em XML (em integração com Flickr, <https://www.flickr.com/services/api/>)

Sincronização

Para aumentar a velocidade e eficiência, o módulo my pics não se conecta directamente à API do Flickr.

Antes do módulo my pics ser invocado é necessário replicar em ficheiros locais o conteúdo da API do Flickr onde estão armazenadas todas as informações que nos vão permitir visualizar as fotos. Tal é feito invocando a página sync.aspx:

Os métodos `Synchronize Collections` e `Synchronize PhotoSets` são públicos, não necessitando de autenticação. No entanto, para invocar o método `Synchronize Pictures`, tal torna-se necessário:

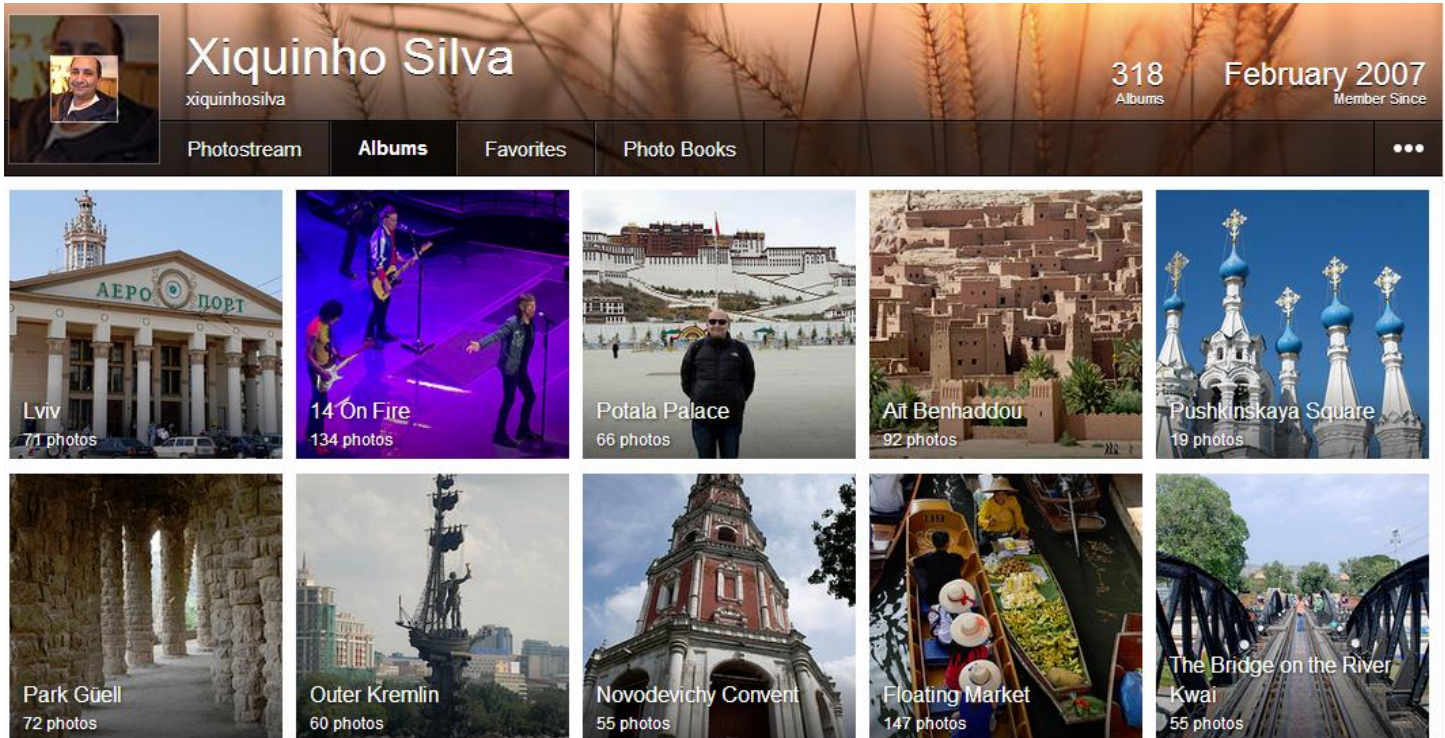
```
Dim f As Flickr = FlickrManager.GetInstance()
Dim token As OAuthRequestToken = f.OAuthGetRequestToken(Request.Url.AbsoluteUri)
Session("RequestToken") = token
Dim url As String = f.OAuthCalculateAuthorizationUrl(token.Token, AuthLevel.Read)
Response.Redirect(url)
```

Daqui somos redireccionados para a página de autenticação do Flickr. Após autenticarmos, estamos aptos a invocar o método `Synchronize Pictures`

The screenshot shows the Flickr interface with the following elements:

- Navigation bar: flickr, You, People, Groups, Explore, Upload, search bar, and user profile picture.
- User profile: Hi xiquinhosilva.
- Notification: MyPictures wants to link to your Flickr account. This is a third-party service. If you don't trust it with access to your account, then you should not authorize it.
- Permissions section:
 - By authorizing this link, you'll allow MyPictures to:
 - ✓ Access your Flickr account (including private content)
 - MyPictures will not have permission to:
 - ✗ Upload, Edit, or Replace photos and videos in your account
 - ✗ Interact with other members' photos and videos (comment, add notes, favorite)
 - ✗ Delete photos and videos from your account
- Buttons: OK, I'LL AUTHORIZE IT (highlighted in blue), NO THANKS (greyed out).
- Disclaimer: Your Yahoo! and Flickr passwords will always remain private, but MyPictures will have the permissions listed above until you revoke its link to your account. (You can revoke such links at any time in the Extending Flickr section of your account prefs.)

Estes três métodos vão gerar os três ficheiros xml, a partir dos quais vamos buscar todas as informações necessárias a visualizar os álbuns do Flickr e suas fotografias, FlickrCollections.xml, FlickrPhotoSets.xml, FlickrPictures.xml



Invocamos o método `Synchronize Collections` para sincronizar as colecções (collections). As colecções funcionam como conjuntos de álbuns (photosets) e os álbuns são conjuntos de fotos.

```
Dim setpath = HttpContext.Current.Server.MapPath("~/App_Data/settings.xml")
Dim setdoc As New XmlDocument()
setdoc.Load(setpath)
Dim api_key = setdoc.SelectNodes("/settings/FlickrApiKey").Item(0).InnerText
Dim user_id = setdoc.SelectNodes("/settings/flickrUserId").Item(0).InnerText
Dim apiUrl As String = "https://api.flickr.com/services/rest/?method=flickr.collections.getTree&api_key=" &
api_key & "&user_id=" & user_id & "&format=rest"
Dim doc = XDocument.Load(apiurl)
```

Após a leitura da API do Flickr, fazemos a conversão para um ficheiro XML

```
Dim writer As XmlTextWriter = New
XmlTextWriter(HttpContext.Current.Server.MapPath("~/App_Data/FlickrCollections.xml"), Nothing)
writer.Formatting = System.Xml.Formatting.Indented
doc.Save(writer)
writer.Close()
```

Onde ficam guardadas as todas colecções e albuns segundo uma estrutura *Continente -> País -> Cidade*, cujo resultado é:

```
<?xml version="1.0"?>
<rsp stat="ok">
  <collections>
    <collection id="7105944-72157625110456705" title="Africa" description="Africa is the world's ..."
    iconlarge="http://farm2.staticflickr.com/1067/cols/72157625110456705_6c3e76c02a_l.jpg"
    iconsmall="http://farm2.staticflickr.com/1067/cols/72157625110456705_6c3e76c02a_s.jpg">
      <collection id="7105944-72157640520062184" title="Morocco" description="Morocco, officially ..."
      iconlarge="http://farm8.staticflickr.com/7303/cols/72157640520062184_d5f5e95423_l.jpg"
      iconsmall="http://farm8.staticflickr.com/7303/cols/72157640520062184_d5f5e95423_s.jpg">
        <set id="72157640468295915" title="Ait Benhaddou" description="Ait Benhaddou is a ..." />
      </collection>
    <collection id="7105944-72157616904847692" title="Egypt" description="Egypt (Arabic: مصر Miṣr..."
    iconlarge="http://farm2.staticflickr.com/1368/cols/72157616904847692_f29ab92f8f_l.jpg"
    iconsmall="http://farm2.staticflickr.com/1368/cols/72157616904847692_f29ab92f8f_s.jpg">
      <set id="72157616624317758" title="Alexandria" description="Alexandria ..." />
      <set id="72157616533223656" title="Cairo" description="Cairo (Arabic: القاهرة .. " />
      <set id="72157616740733656" title="Giza" description="The Giza Necropolis stands .. " />
      <set id="72157616898777424" title="Memphis" description="Memphis (Arabic: منف) was .. " />
      <set id="72157617376856870" title="Saqqara" description="Saqqara or Sakkara, .. " />
    </collection>
  </collection>
</collections>
```

De seguida, após a leitura recursiva de todos os nódulos do ficheiro XML,

```
For Each result In doc.<rsp>.<collections>.<collection>.<collection>.<set>
  Dim item = result
  row = table.NewRow()
  row("ID") = item.Attribute("id").Value.ToString
  row("Title") = StrConv(item.Attribute("title").Value.ToString, VbStrConv.ProperCase)
  row("CatType") = "set"
  row("Description") = item.Attribute("description").Value.ToString
  row("IconLarge") = ""
  table.Rows.Add(row)
Next
```

Fazemos a sua conversão para uma *datatable*:

File renamed: FlickrCollections.xml - FlickrCollections-2014-06-21-10-39-23.xml

Authenticate

You successfully create MyPics Lucene Index

Synchronize Collections

You successfully synchronized collections!

Synchronize PhotoSets

Synchronize Pictures (Authentication required)

ID	Title	CatType	Description
7105944-72157625110456705	Africa	collection	Africa is the world's second-largest and second most-populous continent, after Asia. islands, it covers 6% of the Earth's total surface area and 20.4% of the total land area about 14.72% of the world's human population.
7105944-72157616814207671	America	collection	The Americas, or America, are lands in the Western hemisphere, also known as the 1 America with their associated islands and regions. The plural form the Americas is o more commonly used to refer to the United States of America. The Americas cover 8 contain about 13.5% of the human population (about 900 million people).
7105944-72157616814179227	Asia	collection	Asia is the world's largest and most populous continent, located primarily in the east area (or 29.9% of its land area) and with approximately 4 billion people, it hosts 60% Asia's population nearly quadrupled.
7105944-72157616813773871	Europe	collection	Europe is one of the world's seven continents. Comprising the westernmost peninsul water divide of the Ural Mountains, the Ural River, the Caspian Sea, the Caucasus re Arctic Ocean and other bodies of water to the north, the Atlantic Ocean to the west, t waterways to the southeast. Yet the borders for Europe—a concept dating back to cl refer to a cultural and political distinction or a physiographic one.
7105944-72157616904511742	Middle East	collection	The Middle East (from a European perspective) is a region that encompasses southw expanded in usage to sometimes include Afghanistan and Pakistan, the Caucasus and East, in opposition to Far East. The corresponding adjective is Middle-Eastern and th

Esta *datatable* é guardada na tabela `FlickrCollections` da base de dados *SQL Server Compact*, `MyPics.sdf`:

```
Dim keepNulls = True
Dim reader As IDataReader = table.CreateDataReader()
Dim options As New SqlCeBulkCopyOptions()
If keepNulls Then
    options = options = options Or SqlCeBulkCopyOptions.KeepNulls
End If
Using bc As New SqlCeBulkCopy(MyPicsconnectionString, options)
    bc.DestinationTableName = "FlickrCollections"
    bc.WriteToServer(reader)
End Using
```

FlickrCollections:...pp_Data\MyPics.sdf

ID	Title	CatType	Description	IconLarge
2157625110456705	Africa	collection	Africa is the world's sec...	http://farm2.static...
7105944-72157616814207671	America	collection	The Americas, or Ameri...	http://farm4.static...
7105944-72157616814179227	Asia	collection	Asia is the world's large...	http://farm4.static...
7105944-72157616813773871	Europe	collection	Europe is one of the wo...	http://farm3.static...
7105944-72157616904511742	Middle East	collection	The Middle East (from a...	http://farm3.static...
7105944-72157625235212818	Oceania	collection	Oceania is a geographic...	http://farm5.static...
7105944-72157640520062184	Morocco	collection	Morocco, officially the ...	http://farm8.static...
7105944-72157616904847692	Egypt	collection	Egypt (Arabic: مصر Miṣr...	http://farm2.static...
7105944-72157625629454678	Mexico	collection	Mexico, officially know...	http://farm6.static...
7105944-72157616904851032	Brazil	collection	Brazil, officially the Fed...	http://farm4.static...
7105944-72157616904849532	Usa	collection	The United States of A...	http://farm6.static...
7105944-72157616904885208	Brunei	collection	Brunei, officially the Na...	http://farm4.static...
7105944-72157616904883576	Cambodia	collection	Cambodia (Khmer: កម្ពុជា...	http://farm2.static...
7105944-72157616763517747	China	collection	China, officially the Peo...	http://farm4.static...
7105944-72157616814568787	Hong Kong ...	collection	Hong Kong (香港) is on...	http://farm8.static...
7105944-72157616814565837	Indonesia	collection	Indonesia , officially the...	http://farm3.static...
7105944-72157616814567151	India	collection	India, officially the Rep...	http://farm2.static...
7105944-72157625235171328	Japan	collection	Japan (日本 Nihon or N...	http://farm2.static...
7105944-72157616814562349	Malaysia	collection	Malaysia is a federal co...	http://farm2.static...
7105944-72157616763402713	Macau Sar	collection	Macau (Chinese: 澳門),...	http://farm4.static...
7105944-72157616814555099	Nepal	collection	Nepal (Nepali: नेपाल), o...	http://farm2.static...
7105944-72157616904871198	Philippines	collection	The Philippines, official...	http://farm4.static...
7105944-72157616904874584	Myanmar	collection	Burma, officially the Re...	http://farm2.static...
7105944-72157616904869404	Republic Of ...	collection	Singapore, officially kn...	http://farm4.static...

1 of 379

E de seguida, criamos o índice *Lucene* que vai funcionar como motor de busca do módulo *my pics*:

```
Dim writerlucene As New IndexWriter(MapPath("~/App_Data/MyPicsIndex/"), New StandardAnalyzer(), True)
Dim sql_str As String = "SELECT * FROM FlickrCollections"
Dim dt As DataTable = GetDataTableSqlCE(sql_str, "MyPics")
Dim dv As New DataView(dt)
For Each rowlucene As DataRowView In dv
    Dim doclucene As New Document()
    doclucene.Add(New Field("ID", rowlucene("ID"), Field.Store.YES, Field.Index.UN_TOKENIZED))
    doclucene.Add(New Field("Title", rowlucene("Title"), Field.Store.YES, Field.Index.TOKENIZED))
    doclucene.Add(New Field("CatType", rowlucene("CatType"), Field.Store.YES, Field.Index.UN_TOKENIZED))
    doclucene.Add(New Field("Description", rowlucene("Description"), Field.Store.YES, Field.Index.TOKENIZED))
    doclucene.Add(New Field("IconLarge", rowlucene("IconLarge"), Field.Store.YES, Field.Index.UN_TOKENIZED))
    writerlucene.AddDocument(doclucene)
Next
writerlucene.Optimize()
writer.Close()
```

Os outros dois métodos, *Synchronize PhotoSets* e *Synchronize Pictures* são análogos a este pelo que não irão ser aqui objecto de exposição detalhada.

Uma vez, criada a base de dados local para o módulo *my pics*, podemos invocar a sua página principal. A primeira vez que o módulo *my pics* corre os seus dados são carregados a partir da base de dados *SQL Server Compact*, invocando o método **GetListForSiteMapMyPictures** que, à semelhança dos módulos anteriores, cria uma *datatable* contendo todas as informações das categorias necessárias à construção da navegação do módulo.

ID	Title	CatType	Url	Parent
0	Home	rootcat	/mypictures/	
7105944-72157625110456705	Africa	contincat	africa	0
7105944-72157616814207671	America	contincat	america	0
7105944-72157616814179227	Asia	contincat	asia	0
7105944-72157616813773871	Europe	contincat	europa	0
7105944-72157616904511742	Middle East	contincat	middle-east	0
7105944-72157625235212818	Oceania	contincat	oceania	0
7105944-72157608569202434	Myschool	contincat	myschool	0
7105944-72157640520062184	Morocco	countrycat	morocco	7105944-72157625110456705
7105944-72157616904847692	Egypt	countrycat	egypt	7105944-72157625110456705
7105944-72157625629454678	Mexico	countrycat	mexico	7105944-72157616814207671
7105944-72157616904851032	Brazil	countrycat	brazil	7105944-72157616814207671
7105944-72157616904849532	Usa	countrycat	usa	7105944-72157616814207671
7105944-72157616904885208	Brunei	countrycat	brunei	7105944-72157616814179227
7105944-72157616904883576	Cambodia	countrycat	cambodia	7105944-72157616814179227
7105944-72157616763517747	China	countrycat	china	7105944-72157616814179227
7105944-72157616814568787	Hong Kong Sar	countrycat	hong-kong-sar	7105944-72157616814179227
7105944-72157616814565837	Indonesia	countrycat	indonesia	7105944-72157616814179227

As categorias utilizadas no módulo my pics baseiam-se na localização geográfica das fotos. Para criarmos esta *datatable*, primeiro lemos o ficheiro

```
' we're using cached version
apiurl = HttpContext.Current.Server.MapPath("~/App_Data/FlickrCollections.xml")
Dim doc = XDocument.Load(apiurl)
```

E de seguida, adicionamos os continentes, os países e as cidades:

```
For Each result In doc.<rsp>.<collections>.<collection>.<collection>.<set>
    Dim item = result
    Dim p = item.Parent
    row = table.NewRow()
    row("ID") = item.Attribute("id").Value.ToString
    row("Title") = StrConv(item.Attribute("title").Value.ToString, VbStrConv.ProperCase)
    row("CatType") = "setcat"
    row("Url") = LCase(StripSlug(item.Attribute("title").Value.ToString))
    row("Parent") = p.Attribute("id").Value.ToString
    row("icon") = ""
    table.Rows.Add(row)
Next
```

E por último guardamos a *datatable* em *cache*:

```
System.Web.HttpContext.Current.Cache.Insert(cachekey, table, Nothing,
System.Web.Caching.Cache.NoAbsoluteExpiration, TimeSpan.FromMinutes(CacheTimeSpan))
```

Esta *datatable* irá ser utilizada para a construção da navegação do módulo my pics:

Menu my pics

The screenshot displays a web application interface for 'my pictures'. At the top, a blue navigation bar contains links: 'my web', 'my books', 'my movies', 'my pictures', 'my shop', and 'my travels'. Below this, the page title 'my pictures' is on the left, and a search bar labeled 'search pictures' with a magnifying glass icon is on the right. The main content area is a grid of 18 items, each with a thumbnail image, a title, and a count:

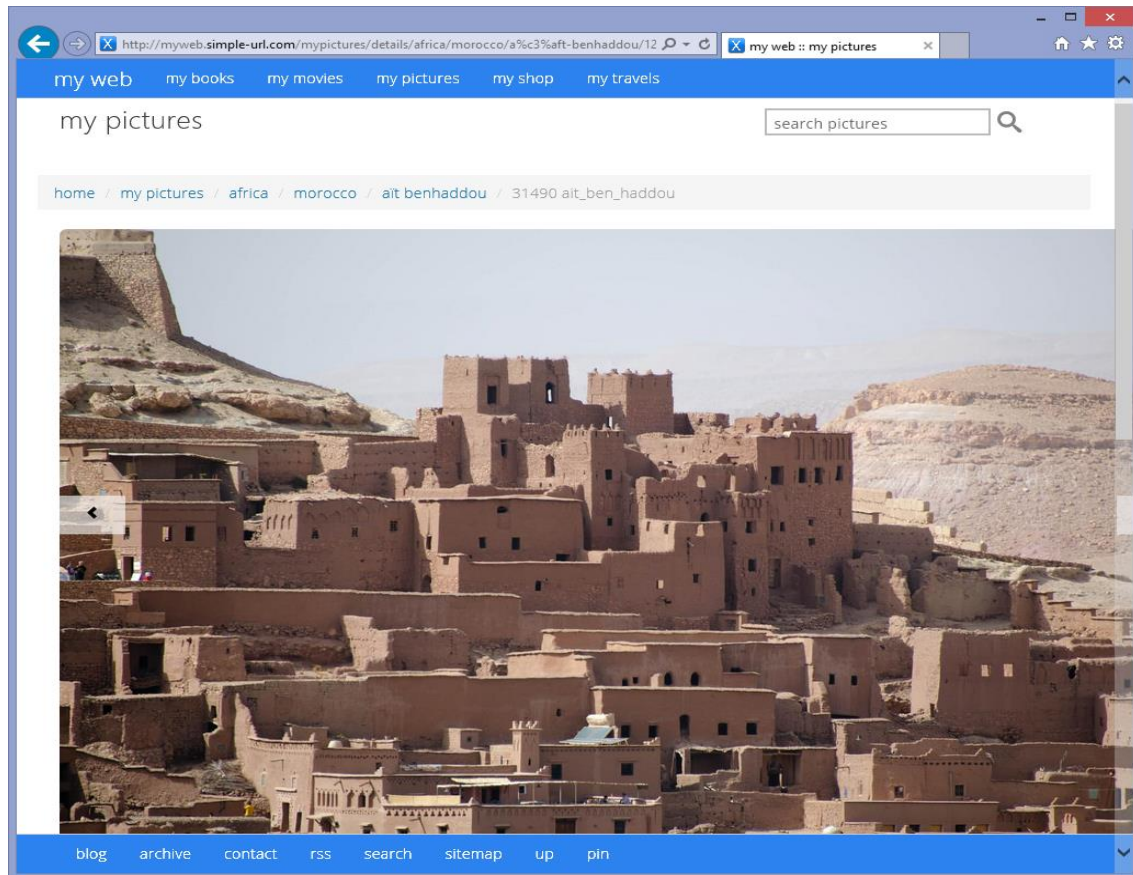
Ait Benhaddou 92	Pushkinskaya Square 19	Park Güell 72	Outer Kremlin 59	Novodevichy Convent 55	Floating Market 147
The Bridge on the River Kwai 55	Salisbury 23	The Kremlin 80	Moscow Metro 54	Kolomenskoye 52	GUM 38
Christ The Saviour 31	Memorial Museum of 110	All-Russia Exhibition 50	all sets 313	africa 2	america 3
asia 17	europe 26	middle east 6	oceania 2		

At the bottom, a blue navigation bar contains links: 'blog', 'archive', 'contact', 'rss', 'search', 'sitemap', 'up', and 'pin'.

Álbum de fotos

The screenshot shows a web browser window displaying a gallery page for 'aït benhaddou'. The browser's address bar shows the URL: `http://myweb.simple-url.com/mypictures/browse/africa/morocco/a%3c%aft-benhaddou`. The page has a blue header with navigation links: [my web](#), [my books](#), [my movies](#), [my pictures](#), [my shop](#), and [my travels](#). Below the header, the title 'aït benhaddou' is displayed on the left, and a search box labeled 'search pictures' is on the right. A breadcrumb trail reads: [home](#) / [my pictures](#) / [africa](#) / [morocco](#) / [ait benhaddou](#). A paragraph of text describes the location: 'Ait Benhaddou is a fortified city, or ksar, along the former caravan route between the Sahara and Marrakech in present-day Morocco. It is situated in Souss-Massa-Drâa on a hill along the Ounila River and is known for its kasbahs, although they take damage with each rainstorm. Most of the town's inhabitants now live in a more modern village at the other side of the river; however, eight families still live within the ksar. Ait Benhaddou has been a UNESCO World Heritage Site since 1987 and several films have been shot there.' Below the text is a grid of 24 small images showing various views of the ksar, including its mud-brick buildings, towers, and surrounding landscape. At the bottom of the page, a blue footer contains navigation links: [blog](#), [archive](#), [contact](#), [rss](#), [search](#), [sitemap](#), [up](#), and [pin](#).

Detalhes da foto



3.7.1 Classe mypics

'Returns a datatable with Mypics Album parent data

```
Public Shared Function GetMyPicsParentUrlIDV(SiteMapTable As DataTable, url As String) As DataView
```

```
url = Replace(url, "'", "'")
```

```
Dim result As DataRow() = SiteMapTable.[Select]("Url = '" & url & "'")
```

```
Dim nodeTable As New DataTable()
```

```
If result.Count > 0 Then
```

```
Dim parentnodeid As String = result(0)("Parent") ' parent's id
```

```
nodeTable.Columns.Add(New DataColumn("Name", GetType(String)))
```

```
nodeTable.Columns.Add(New DataColumn("ID", GetType(String)))
```

```
nodeTable.Columns.Add(New DataColumn("sort", GetType(String)))
```

```
Dim c = 0
```

```
Dim pathDone As Boolean = False
```

```
While Not pathDone
```

```
Dim resultparentnode As DataRow()
```

```
resultparentnode = SiteMapTable.[Select]("ID = '" & parentnodeid & "'")
```

```
If resultparentnode.Count > 0 Then
```

```
If resultparentnode(0)("Parent").ToString = String.Empty Then
```

```
pathDone = True
```

```

Else
    Dim rowparent As DataRow = nodeTable.NewRow()
    rowparent("Name") = resultparentnode(0)("Title")
    rowparent("ID") = resultparentnode(0)("Url")
    rowparent("sort") = c
    nodeTable.Rows.InsertAt(rowparent, 0)
    parentnodeid = resultparentnode(0)("Parent")
    c = c + 1
End If
End If
End While
Dim dvparents As New DataView(nodeTable)
Return dvparents
End If
Return Nothing
End Function

'Returns a string with Mypics Album parent url data
Public Shared Function GetMyPicsParentUrl(SiteMapTable As DataTable, url As String) As String
    url = Replace(url, "", "")
    Dim result As DataRow() = SiteMapTable.[Select]("Url = '" & url & "'")
    For Each row As DataRow In result
        Dim parentrow = row(4) 'row4 = parent
        Dim resultparent As DataRow() = SiteMapTable.[Select]("ID = '" & parentrow & "'")
        For Each rowparent As DataRow In resultparent
            Return rowparent(3) 'row3 = url
        Next
    Next
    Return Nothing
End Function

'Returns a datatable with Mypics Album child data
Public Shared Function GetMyPicsChildUrl(SiteMapTable As DataTable, url As String) As DataRow()
    url = Replace(url, "", "")
    Dim result As DataRow() = SiteMapTable.[Select]("Url = '" & url & "'")
    For Each row As DataRow In result
        Dim id = row(0) 'row0 = id
        Dim resultchildren As DataRow() = SiteMapTable.[Select]("Parent = '" & id & "'")
        Return resultchildren
        For Each rowchild As DataRow In resultchildren
            Return rowchild(0) 'row0 = id
        Next
    Next
    Return Nothing
End Function

```

3.8 MÓDULO MY TRAVELS

Módulo destinado a inventariar a colecção de viagens do utilizador, armazenamento de dados em XML (em integração com a base de dados da Unesco WH Database.) Posteriormente este módulo irá permitir a introdução de viagens pessoais do utilizador, sendo neste momento utilizada a base de dados da UWH para efeitos de demonstração.

(<http://whc.unesco.org/en/syndication/>)

Sincronização

À semelhança do módulo *my pics*, o módulo *my travels* também não se conecta directamente à API da Unesco WH.

Antes do módulo *mytravels* ser invocado é necessário replicar em ficheiros locais a Lista de Património Mundial onde estão armazenadas todas as informações que nos vão permitir visualizar as viagens. A Unesco disponibiliza a Lista em vários formatos, entre os quais XML, pelo que não necessitamos de efectuar qualquer conversão. Assim, gravamos localmente o ficheiro *UnescoHeritage.xml* em *App_data*.

```
?xml version="1.0"?>
<query columns="22" rows="981">
  <row>
    <category>Cultural</category>
    <criteria_txt>(iv)</criteria_txt>
    <danger>
    </danger>
    <date_inscribed>1983</date_inscribed>
    <historical_description>
    </historical_description>
    <http_url>http://whc.unesco.org/en/list/275</http_url>
    <id_number>275</id_number>
    <image_url>http://whc.unesco.org/uploads/sites/site_275.jpg</image_url>
    <iso_code>ar,br</iso_code>
    <justification>
    </justification>
    <latitude>-28.5433333300</latitude>
    <location>State of Rio Grande do Sul, Brazil; Province of Misiones, Argentina</location>
    <longitude>-54.2658333300</longitude>
    <long_description>&lt;p&gt;The remains of these Jesuit missions are outstanding ...</long_description>
    <region>Latin America and the Caribbean</region>
    <revision_extension>bis</revision_extension>
    <secondary_dates>1984</secondary_dates>
```

```

<short_description>&lt;p&gt;The ruins of S&amp;atilde;o Miguel das Miss&amp;otilde;es in
Brazil...</short_description>
<site>Jesuit Missions of the Guaranis: San Ignacio Mini, Santa Ana, Nuestra Señora de Loreto and Santa Maria Mayor
(Argentina), Ruins of Sao Miguel das Missoes (Brazil)</site>
<states>Argentina,Brazil</states>
<transboundary>1</transboundary>
<unique_number>326</unique_number>
</row>

```

A partir deste ficheiro, vamos criar duas datatables invocando o método `Synchronize MyTravels-UnescoDB`: a lista de viagens e a lista de viagens por país. Tal é feito invocando a página `sync.aspx` onde também criamos a *datatable* para lista de álbuns de fotos por viagem:

Após criarmos duas datatables vazias destinadas à lista de viagens e a lista de viagens por país, vamos ler recursivamente todos os nódulos do ficheiro XML, para preencher a tabela `Heritage` destinada à lista de viagens. Simultaneamente, aproveitamos esta iteração, para criamos também o índice *Lucene* que vai funcionar como motor de busca do módulo *my travels*.

```

For Each result In doc.<query>.<row>
  Dim doclucene As New Document()
  Dim item = result
  row = table.NewRow()
  row("ID") = item.Elements("id_number").Value.ToString
  doclucene.Add(New Field("ID", item.Elements("id_number").Value.ToString, Field.Store.YES)
  row("Title") = item.Elements("site").Value.ToString
  doclucene.Add(New Field("Title", item.Elements("site").Value.ToString, Field.Store.YES)
  row("CatType") = Trim(item.Elements("category").Value.ToString)
  doclucene.Add(New Field("CatType", Trim(item.Elements("category").Value.ToString)
  row("short_description") = item.Elements("short_description").Value.ToString
  row("long_description") = item.Elements("long_description").Value.ToString
  row("IconLarge") = item.Elements("image_url").Value.ToString
  doclucene.Add(New Field("IconLarge", item.Elements("image_url").Value.ToString)
  row("date_inscribed") = item.Elements("date_inscribed").Value.ToString
  doclucene.Add(New Field("date_inscribed", item.Elements("date_inscribed").Value.ToString)
  doclucene.Add(New Field("active", 1, Field.Store.YES, Field.Index.UN_TOKENIZED))
  row("http_url") = item.Elements("http_url").Value.ToString
  row("iso_code") = item.Elements("iso_code").Value.ToString
  row("latitude") = item.Elements("latitude").Value.ToString
  row("longitude") = item.Elements("longitude").Value.ToString
  row("location") = item.Elements("location").Value.ToString
  row("region") = item.Elements("region").Value.ToString
  doclucene.Add(New Field("region", item.Elements("region").Value.ToString, Field.Store.YES)
  row("states") = item.Elements("states").Value.ToString
  row("transboundary") = item.Elements("transboundary").Value.ToString
  row("unique_number") = item.Elements("unique_number").Value.ToString
  table.Rows.Add(row)

```

E de seguida, criamos a *datatable* `HeritageCountries` que tal como o seu nome indica destina-se à lista de países por viagem:

File renamed: UnescoHeritage.xml - UnescoHeritage-2014-06-29-12-19-37.xml

You successfully create Heritage table

Synchronize MyTravels-UnescoDB

You successfully synchronized heritage!

Synchronize MyTravels-MyPics

heritageID	country	countrycode	region
275	Argentina	ar	Latin America and the Caribbean
275	Brazil	br	Latin America and the Caribbean
1363	Austria	at	Europe and North America
1363	France	fr	Europe and North America
1363	Germany	de	Europe and North America
1363	Italy	it	Europe and North America
1363	Slovenia	si	Europe and North America
1363	Switzerland	ch	Europe and North America
772	Austria	at	Europe and North America
772	Hungary	hu	Europe and North America
1187	Belarus	by	Europe and North America
1187	Estonia	ee	Europe and North America
1187	Finland	fi	Europe and North America
1187	Latvia	lv	Europe and North America
1187	Lithuania	lt	Europe and North America
1187	Norway	no	Europe and North America
1187	Moldova	md	Europe and North America
1187	Russia	ru	Europe and North America
1187	Sweden	se	Europe and North America
1187	Ukraine	ua	Europe and North America
33	Belarus	by	Europe and North America
33	Poland	pl	Europe and North America
943	Belgium	be	Europe and North America
943	France	fr	Europe and North America

' Turn states input to a list of countries.

```
Dim countries As String() = item.Elements("states").Value.ToString.Split(New String() {"", ""},
StringSplitOptions.RemoveEmptyEntries)
```

```
Dim codes As String() = item.Elements("iso_code").Value.ToString.Split(New String() {"", ""},
StringSplitOptions.RemoveEmptyEntries)
```

```
Me.countries = countries.ToList()
```

```
For i As Integer = 0 To countries.Count - 1
```

```
crow = ctable.NewRow()
```

```
'Response.Write(String.Format("keywords[{0}] = '{1}';", i, keywords(i)))
```

```

crow("heritageID") = item.Elements("id_number").Value.ToString
crow("country") = ChangeCountryName(countries(i))
crow("countrycode") = ChangeCountryName(codes(i))
crow("region") = item.Elements("region").Value.ToString
ctable.Rows.Add(crow)
str_countries = str_countries & crow("country") & " "
Next

```

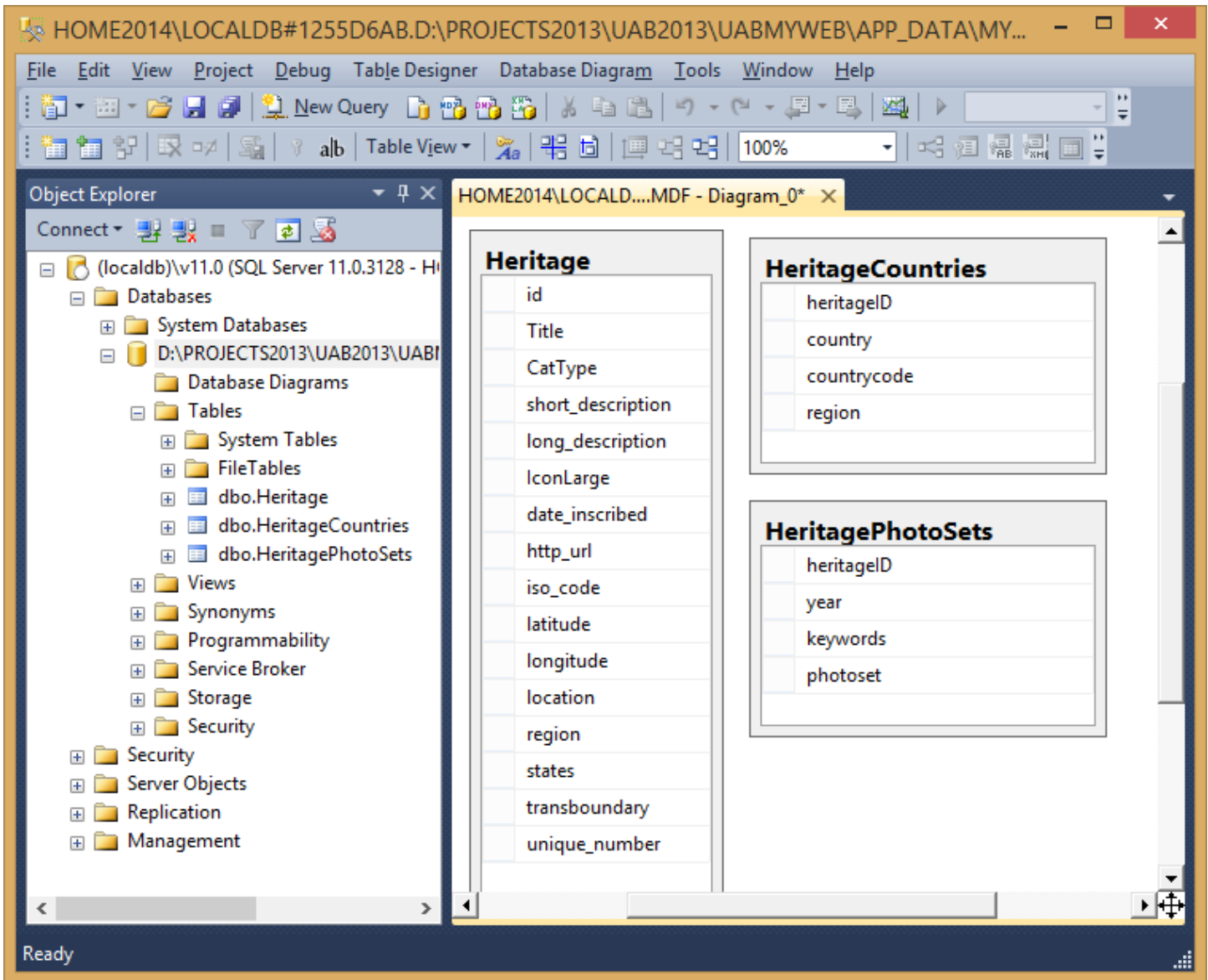
Estas *datatables* são copiadas para as tabelas `Heritage` e `HeritageCountries` da base de dados *SQL Server LocalDB*, `MyTravels.mdf`:

```

Using connection As SqlConnection = New SqlConnection(MyTravelsLocalDB)
connection.Open()
Using bulkCopy As SqlBulkCopy = New SqlBulkCopy(connection)
    bulkCopy.DestinationTableName = "dbo.Heritage"
    Try
        ' Write from the source to the destination.
        bulkCopy.WriteToServer(table)
    Catch ex As Exception
        Response.Write(ex.Message)
    End Try
End Using
Using bulkCopy As SqlBulkCopy = New SqlBulkCopy(connection)
    bulkCopy.DestinationTableName = "dbo.HeritageCountries"
    Try
        ' Write from the source to the destination.
        bulkCopy.WriteToServer(ctable)
    Catch ex As Exception
        Response.Write(ex.Message)
    End Try
End Using

```

End Using



Uma vez, criada a base de dados para o módulo my travels, podemos invocar a sua página principal. A primeira vez que o módulo my travels corre os seus dados são carregados a partir da base de dados *SQL Server LocalDB*, invocando o método **GetListForSiteMapMyTravels** que, à semelhança dos módulos anteriores, cria uma *datatable* contendo todas as informações das categorias necessárias à construção da navegação do módulo.

ID	Title	CatType	Url	Parent
0	Home	rootcat	/mytravels/	
1	Africa	region	/mytravels/browse/africa	Home
2	Arab States	region	/mytravels/browse/arab-states	Home
3	Asia And The Pacific	region	/mytravels/browse/asia-and-the-pacific	Home
4	Europe And North America	region	/mytravels/browse/europe-and-north-america	Home
5	Latin America And The Caribbean	region	/mytravels/browse/latin-america-and-the-caribbean	Home
6	Year	region	/mytravels/browse/year	Home
7	Category	region	/mytravels/browse/category	Home

As categorias utilizadas no módulo my travels baseiam-se na distribuição geográfica das regiões segundo a Lista de Património Mundial da UNESCO:

```

Dim i As Integer
Dim _CatRoot() As String = XMLReturnArray("mytravels", "Title")
For Each _cat As String In _CatRoot
    row = table.NewRow()
    i = i + 1
    row("ID") = i
    row("Title") = StrConv(_cat, VbStrConv.ProperCase)
    row("CatType") = "region"
    row("Url") = "/mytravels/browse/" & LCase(StripSlug(_cat))
    row("Parent") = "Home"
    table.Rows.Add(row)
Next

```

E por último guardamos a *DataTable* em *cache*:

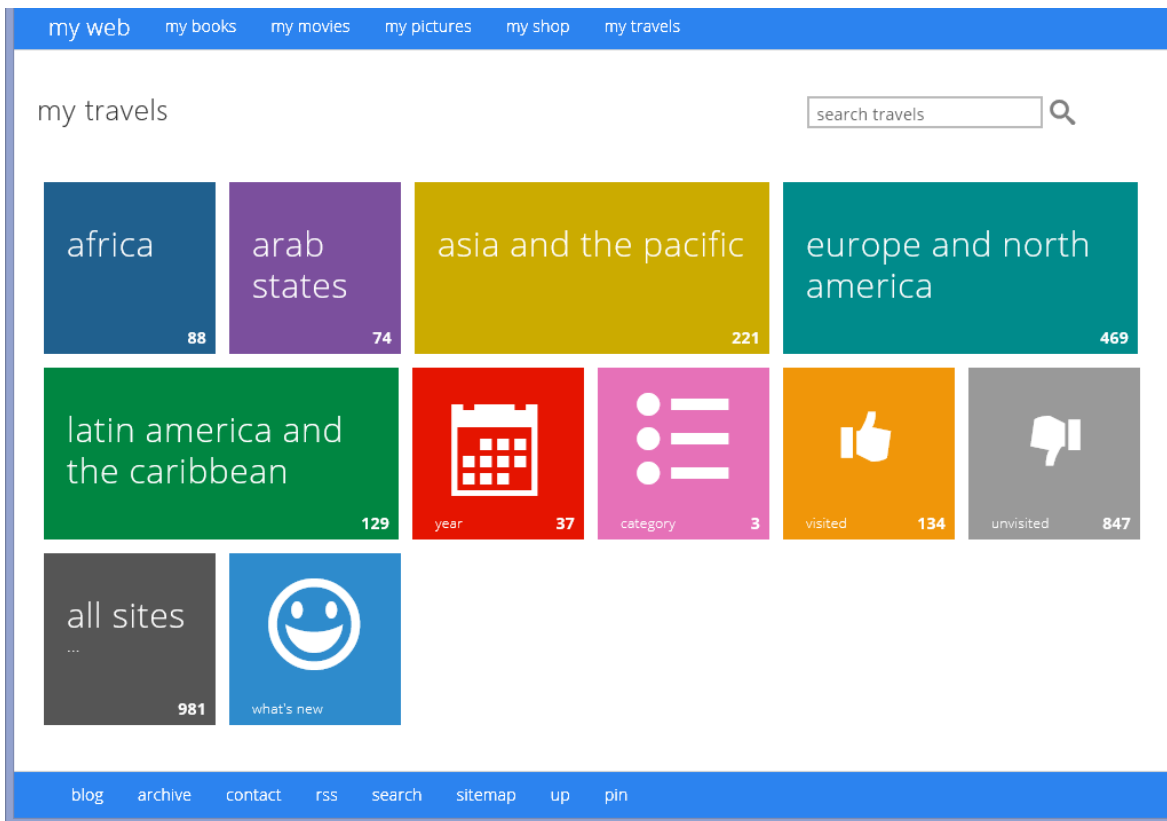
```

System.Web.HttpContext.Current.Cache.Insert(cachekey, table, Nothing,
System.Web.Caching.Cache.NoAbsoluteExpiration, TimeSpan.FromMinutes(CacheTimeSpan))

```

É esta datatable que irá ser utilizada para a construção da navegação do módulo my pics:

Menu my travels



Ainda na página sync.aspx podemos invocar o método `Synchronize MyTravels-MyPics` relacionar a lista de viagens com os álbuns de fotografias.

```

For Each result In pdoc.<DocumentElement>.<site>
    Dim pitem = result
    If pitem.Attribute("Year").Value.ToString <> String.Empty Then
        Dim str_photoset = pitem.Attribute("photoset").Value.ToString
        If str_photoset = String.Empty Then
            pro = ptable.NewRow()
            pro("HeritageID") = pitem.Attribute("ID").Value.ToString
            pro("Year") = pitem.Attribute("Year").Value.ToString
            pro("Keywords") = pitem.Attribute("Keywords").Value.ToString
            'pro("photoset") = pitem.Attribute("photoset").Value.ToString
            ptable.Rows.Add(pro)
        Else
            Dim photosets As String() = str_photoset.Split(New String() {","}, StringSplitOptions.RemoveEmptyEntries)
            Dim photosetlist As List(Of String) = New List(Of String)
            photosetlist = photosets.ToList()
            For Each photoset In photosetlist
                Try

```

```

    prow = ptable.NewRow()
    prow("HeritageID") = pitem.Attribute("ID").Value.ToString
    prow("Year") = pitem.Attribute("Year").Value.ToString
    prow("Keywords") = pitem.Attribute("Keywords").Value.ToString
    prow("photoset") = Trim(photoset).ToString
    ptable.Rows.Add(prow)
Catch ex As Exception
    Response.Write(pitem.Attribute("ID").Value.ToString)
End Try
Next
End If
End If
Next

```

Esta informação está guardada no ficheiro `UnescoHeritageVisited.xml`, criado pelo utilizador, onde guardamos também a lista dos sítios já visitados no campo `Year` e no campo `Keywords` a lista de palavras-chave que queremos utilizar no sistema de recomendação da Amazon.com do módulo My shop. O campo `photosets` é destinado aos álbuns de fotos, do módulo `my pics`:

```

<DocumentElement>
  <site ID="29" Year="2010" Keywords="Kraków" photoset="" />
  <site ID="30" Year="2010" Keywords="Warsaw" photoset="" />
  <site ID="31" Year="2010" Keywords="Auschwitz Birkenau" photoset="" />
  <site ID="32" Year="2010" Keywords="Wieliczka and Bochnia Royal Salt Mines" photoset="" />
  <site ID="42" Year="2010" Keywords="Boyana Church" photoset="" />
  <site ID="75" Year="2009" Keywords="Grand Canyon" photoset="Grand Caynon" />
  <site ID="83" Year="2004" Keywords="Versailles" photoset="" />
  <site ID="86" Year="2007" Keywords="Giza Pyramids" photoset="Giza,Memphis,Saqqara" />
  <site ID="89" Year="2007" Keywords="Cairo Travel" photoset="cairo" />
  <site ID="91" Year="2007" Keywords="historic centre of rome" photoset="Rome,Aventine &
  Trastevere,Capitoline Hill,Colosseum,Pantheon,Roman Forum,Santa Maria Maggiore,San Giovanni In Laterano" />

```

Finalmente copiamos a datatable para a tabela `HeritagePhotosets` da base de dados `SQL Server LocalDB`, `MyTravels.mdf`.

```

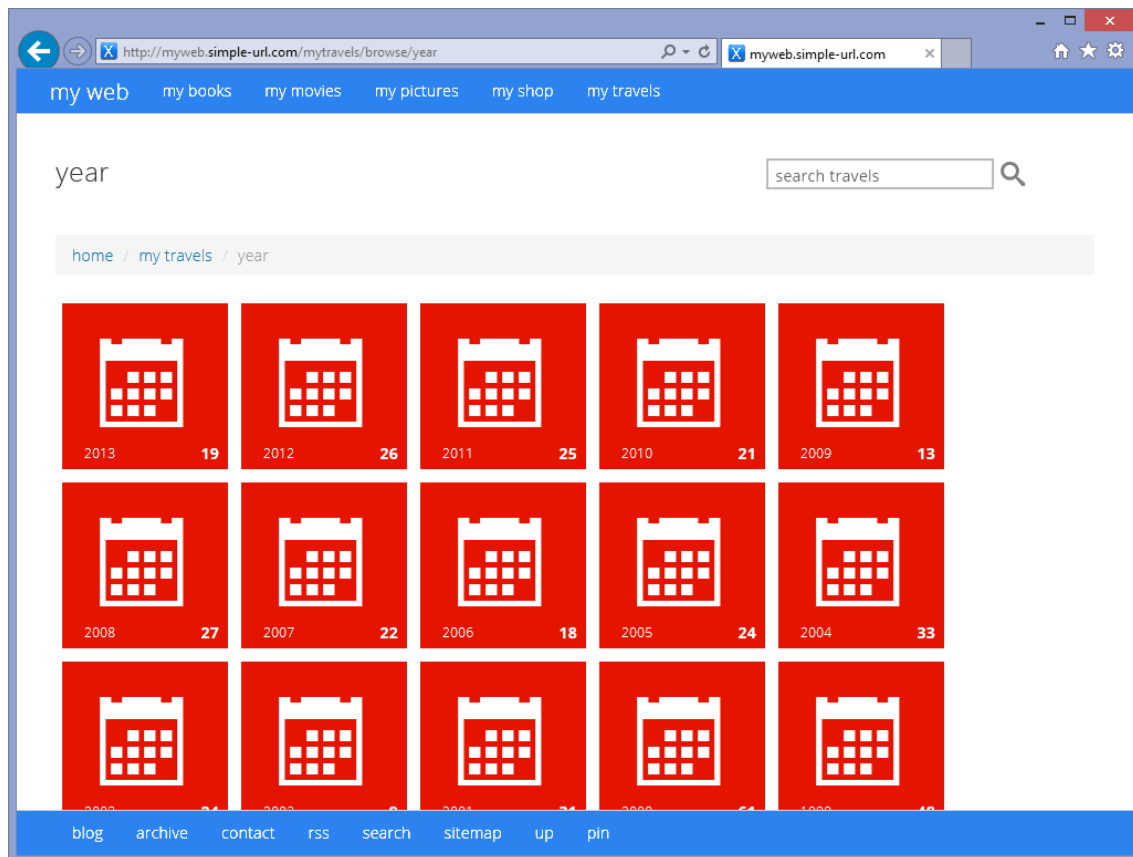
Using connection As SqlConnection = New SqlConnection(MyTravelsLocalDB)
connection.Open()

Using bulkCopy As SqlBulkCopy = New SqlBulkCopy(connection)
    bulkCopy.DestinationTableName = "dbo.HeritagePhotosets"
    Try
        ' Write from the source to the destination.
        bulkCopy.WriteToServer(ptable)
    Catch ex As Exception
        Response.Write(ex.Message)
    End Try
End Using
End Using

```

Ficando completa a base de dados do módulo *my travels*.

Lista por ano



Lista de viagens

year :: 2013

home / my travels / year / 2013

Wooden Tserkvas of the Carpathian Region in Poland and Ukraine

Situated in the eastern fringe of eastern Europe, the transnational property numbers a selection of 16 tserkvas, churches, built of horizontal wooden logs between the 16th and 19th centuries by communities of the Eastern Orthodox and Greek Catholic faiths. They represent the cultural expression of four ethnographic groups and the formal, decorative and technical characteristics they developed over time. The tserkvas bear testimony to a distinct building tradition rooted in Orthodox ecclesiastic design interwoven with elements of local tradition, and symbolic references to their communities' cosmogony. The tserkvas are built on a tri-partite plan surmounted by open quadrilateral or octagonal domes and cupolas. They feature wooden bell towers, iconostasis screens, and interior polychrome decorations as well as churchyards, gatehouses and graveyards.

Red Bay Basque Whaling Station

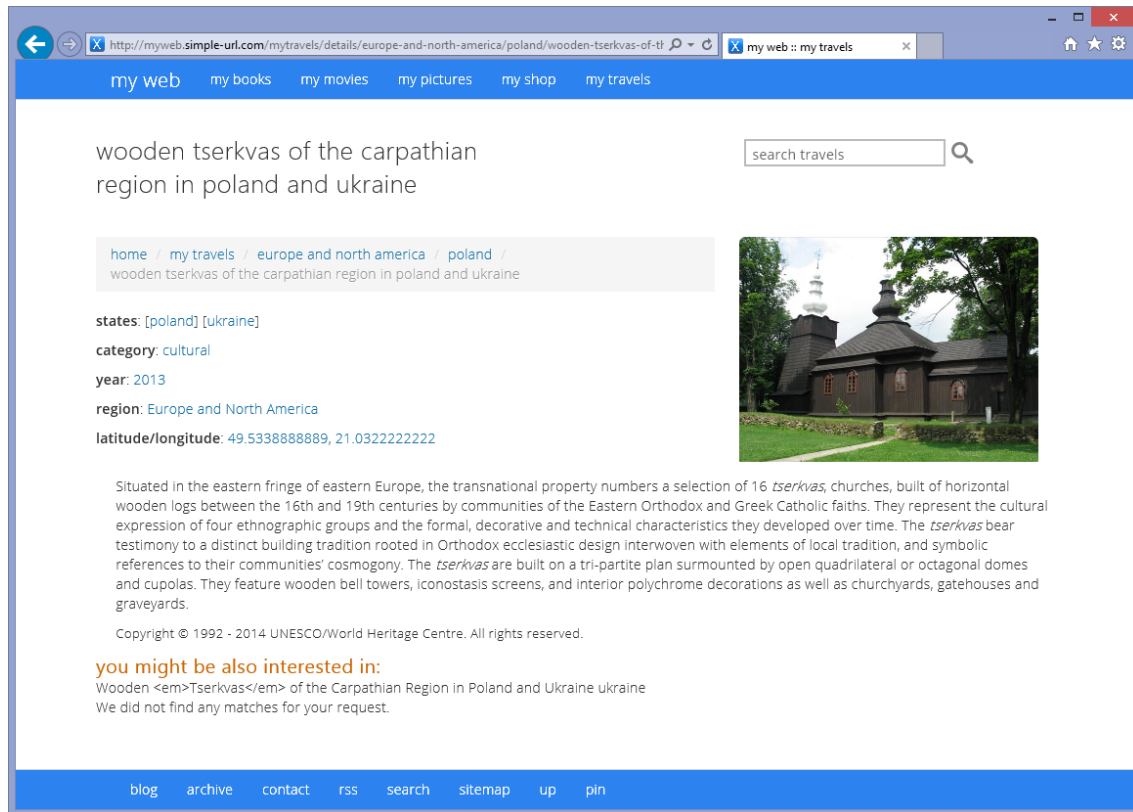
Red Bay, established by Basque mariners in the 16th century at the north-eastern tip of Canada on the shore of the Strait of Belle Isle is an archaeological site that provides the earliest, most complete and best preserved testimony of the European whaling tradition. Gran Baya, as it was called by those who founded the station in 1530s, was used as a base for coastal hunting, butchering, rendering of whale fat by heading to

ukraine cultural 2013 europe and north america

canada cultural

blog archive contact rss search sitemap up pin

Detalhes da viagem



3.8.1 Classe mytravels

' Returns an array of Photo Albums by Heritage Site

Public Shared Function GetSitePhotosets(siteid As String) As String()

Dim myvisitedpath = HttpContext.Current.Server.MapPath("~/App_Data/UnescoHeritageVisited.xml")

Dim myvisiteddoc As New XmlDocument()

myvisiteddoc.Load(myvisitedpath)

Dim myvisited_nodelist As XmlNodeList = myvisiteddoc.SelectNodes("/DocumentElement/site[@ID='" & siteid & "']")

If myvisited_nodelist.Count > 0 Then

If myvisited_nodelist.Item(0).Attributes.GetNamedItem("photoset").Value.ToString <> String.Empty Then

Dim photosets As String() =

myvisited_nodelist.Item(0).Attributes.GetNamedItem("photoset").Value.ToString.Split(New String() {"", ""}, StringSplitOptions.RemoveEmptyEntries)

Return photosets

End If

End If

Return Nothing

End Function

' Returns a string of Heritage Site Keywords

Public Shared Function GetSiteKeywords(siteid As String) As String

```

Dim myvisitedpath = HttpContext.Current.Server.MapPath("~/App_Data/UnescoHeritageVisited.xml")
Dim myvisiteddoc As New XmlDocument()
myvisiteddoc.Load(myvisitedpath)
Dim myvisited_nodelist As XmlNodeList = myvisiteddoc.SelectNodes("/DocumentElement/site[@ID="" & siteid &
"""])
If myvisited_nodelist.Count > 0 Then
    If myvisited_nodelist.Item(0).Attributes.GetNamedItem("Keywords").Value.ToString <> String.Empty Then
        Dim Keywords As String = myvisited_nodelist.Item(0).Attributes.GetNamedItem("Keywords").Value.ToString
        Return Keywords
    End If
End If
Return Nothing
End Function

```

' Returns a string with the year of a Heritage Site visit

```

Public Shared Function DisplayYearVisited(id As String) As String
    Dim yearvisited = ""
    Dim myvisitedpath = HttpContext.Current.Server.MapPath("~/App_Data/UnescoHeritageVisited.xml")
    Dim myvisiteddoc As New XmlDocument()
    myvisiteddoc.Load(myvisitedpath)
    Dim myvisited_nodelist As XmlNodeList = myvisiteddoc.SelectNodes("/DocumentElement/site[@ID="" & id & """])
    If myvisited_nodelist.Count > 0 Then
        yearvisited = myvisited_nodelist.Item(0).Attributes.GetNamedItem("Year").Value.ToString
    End If
    Return yearvisited
End Function

```

'Returns a datatable of countries belonging to Heritage Site

```

Public Shared Function GetCountries(ByVal siteID As String) As DataView
    Dim HeritageTableAdapter As New MyTravelsTableAdapters.HeritageCountriesTRTableAdapter()
    Dim Countriestable As DataTable = HeritageTableAdapter.GetCountriesBySiteID(siteID)
    Dim dataview As DataView = Countriestable.DefaultView
    Return dataview
End Function

```

3.9 MÓDULO MY SHOP

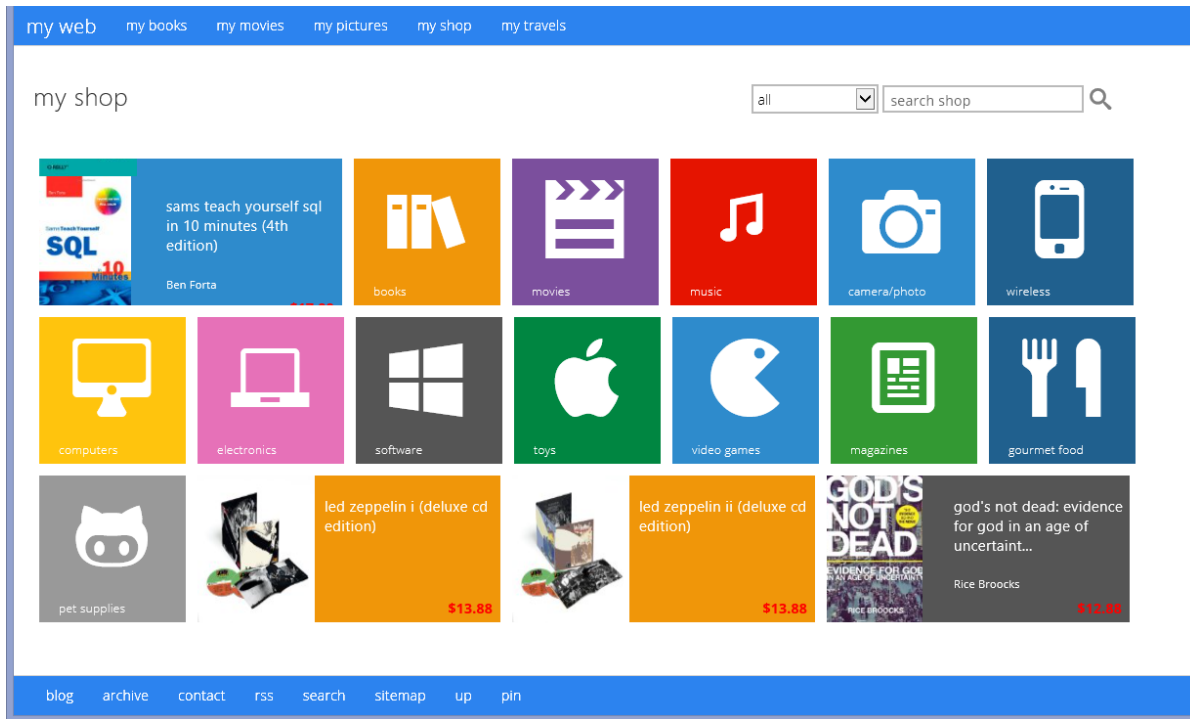
O módulo my shop, constitui o módulo base do Sistema de Recomendação Web, comum a todos os outros módulos, não necessita armazenamento de dados (em integração com amazon.com) – (http://en.wikipedia.org/wiki/Amazon_Product_Advertising_API)

Ao contrário dos módulos anteriores a navegação do módulo my shop é construída directamente a partir da API da Amazon. Apenas as categorias raiz, que são definidas pelo utilizador, são carregadas a partir do ficheiro categories.xml:

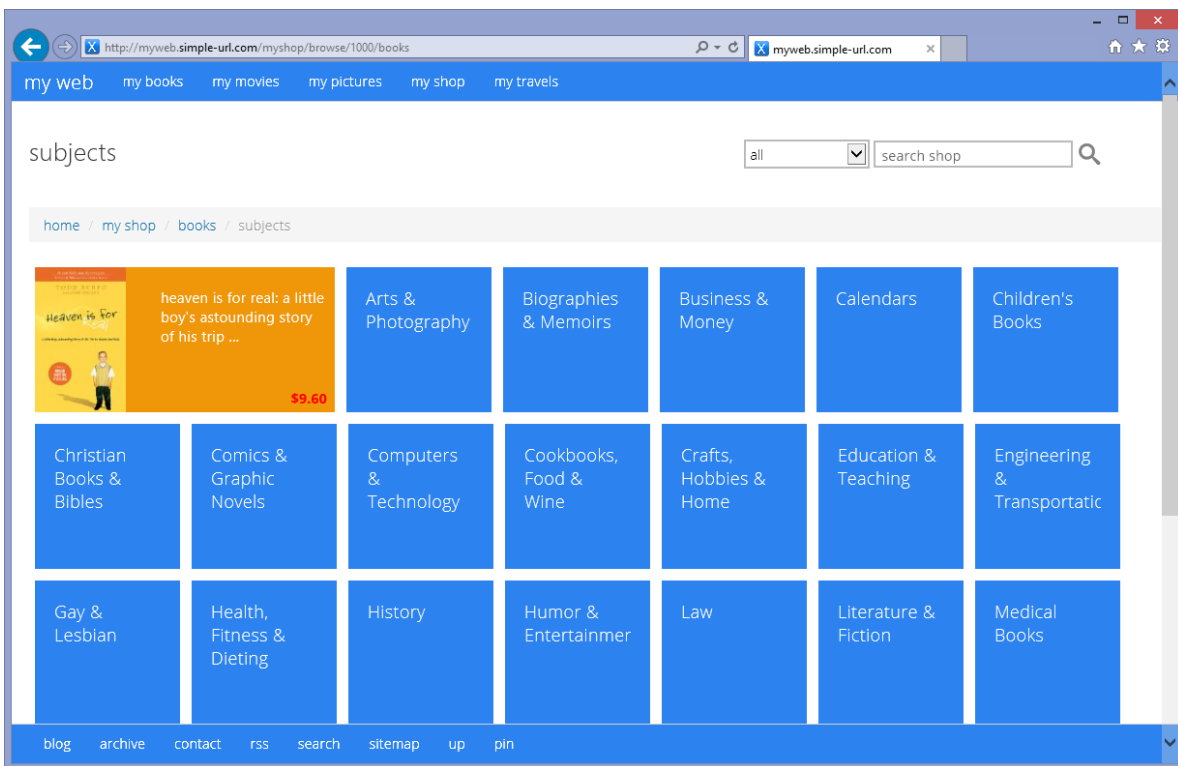
```
<myshop name="my shop" url="myshop" root="True" active="True">
<cat ID="1000" Title="books" SearchGroup="Books" Keywords="asp.net" Index="4" Icon="books" Bgc="orange" />
<cat ID="2650362011" Title="movies" SearchGroup="DVD" Keywords="tarantino" Index="1" Icon="movie" Bgc="purple" />
<cat ID="301668" Title="music" SearchGroup="Music" Keywords="pink floyd" Index="0" Icon="music" Bgc="red" />
<cat ID="3017941" Title="camera/photo" SearchGroup="Photo" Keywords="nikon" Index="8" Icon="camera-3" Bgc="blue" />
<cat ID="2407749011" Title="wireless" SearchGroup="Wireless" Keywords="lumia" Index="33" Icon="mobile-6" Bgc="blueDark" />
<cat ID="565098" Title="computers" SearchGroup="PCHardware" Keywords="microsoft surface" Index="21" Icon="monitor" Bgc="yellow" />
<cat ID="493964" Title="electronics" SearchGroup="Electronics" Keywords="sony" Index="8" Icon="laptop" Bgc="pink" />
<cat ID="491286" Title="software" SearchGroup="Software" Keywords="windows" Index="25" Icon="windows8-3" Bgc="gray" />
<cat ID="165795011" Title="toys" SearchGroup="Toys" Index="28" Keywords="lego" Icon="apple" Bgc="greenDark" />
<cat ID="11846801" Title="video games" SearchGroup="VideoGames" Keywords="halo" Index="31" Icon="pacman" Bgc="blue" />
<cat ID="599872" Title="magazines" SearchGroup="Magazines" Keywords="time" Index="13" Icon="newspaper" Bgc="green" />
<cat ID="2255571011" Title="gourmet food" SearchGroup="GourmetFood" Keywords="cheese" Index="9" Icon="food" Bgc="blueDark" />
<cat ID="2619534011" Title="pet supplies" SearchGroup="PetSupplies" Keywords="pedigree" Index="8" Icon="github" Bgc="grayLight" />
</myshop>
```

A partir do qual obtemos a informação para o menu da página principal:

Menu my shop



Lista de categorias (livros)



Lista de mais vendidos

history

all search shop

home / my shop / books / subjects / arts & photography / architecture / history

top sellers

A Field Guide to American Houses
The Definitive Guide to Identifying and Understanding America's Domestic Architecture
Virginia Savage McAlester
\$33.56

HOW ARCHITECTURE WORKS: A Humanist's Toolkit
WITOLD HYCZYNSKI
\$17.08

SAVING ITALY
THE RACE TO RESCUE A NATION'S TREASURES FROM THE NAZIS
ROBERT M. EDEL
Kindle eBook

Brunelleschi's Dome
How a Renaissance Genius Reinvented Architecture
ROSS KING
\$10.52

SAVING ITALY
THE RACE TO RESCUE A NATION'S TREASURES FROM THE NAZIS
ROBERT M. EDEL
\$13.05

blog archive contact rss search sitemap up pin

Detalhes do produto

lost boston

all search shop

home / my shop / lost boston

List Price: ~~\$18.95~~
Our Price: **\$14.44**
You Save: 23.8%

Availability: Usually ships in 24 hours
Author: Anthony Sammarco
Company: Pavilion

blog archive contact rss search sitemap up pin

3.9.1 Sistemas de Recomendação Web

Sistemas de Recomendação podem ser vistos ao realizar buscas em *websites* de pesquisa da internet, em compras online, ou até mesmo ao visualizamos nossos emails. São o mecanismo por trás da publicidade personalizada que recebemos na web, com indicações de *websites* para visitarmos ou produtos para compramos.

Com o advento do consumo em dispositivos móveis e a propagação *no e-commerce*, sistemas de recomendação tornaram-se um tema extremamente atractivo. Através de algoritmos simples e facilmente integráveis a aplicações web, os sistemas de recomendação agregam valor ao negócio *online*, promovendo itens de consumo direccionados a um público-alvo.

Por trás da publicidade, estes sistemas utilizam abstrações matemáticas de dados. Eles consistem basicamente em algoritmos de filtragem e inferência de dados, que recomendam produtos de acordo com os interesses dos utilizadores. (Souza, 2014)

Os sistemas de recomendação são uma subárea de aprendizagem de máquina (*machine learning*) e tem por objectivo sugerir itens a um utilizador, com base nas suas preferências. Podem ser recomendados itens diversos como livros, filmes ou viagens. É amplamente utilizado como uma estratégia de marketing, já que ao recomendar produtos que estejam alinhados ao interesse do utilizador, é mais provável que ele venha adquirir tal produto.

É possível fazer recomendações comparando as preferências de um utilizador com um grupo de outros utilizador. Também é possível fazer recomendações procurando produtos com características similares aos que o utilizador já demonstrou interesse. Na presente aplicação web as recomendações são feitas com base nos interesses demonstrados pelo utilizador. Por exemplo, se o utilizador demonstrou interesse em visitar o centro histórico do Porto, poderão ser recomendadas diversas publicações referentes a essa cidade.

you might be also interested in:

Postales desde Oporto. Edición en blanco y negro. (Spanis...



Arturo Villarrubia

"En "Postales desde Oporto", Arturo Villarrubia pone de manifiesto su prodigiosa habilidad para capturar la magia de Oporto y pa...

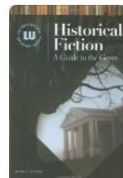
Postales desde Oporto (Spanish Edition)



Arturo Villarrubia

"En "Postales desde Oporto", Arturo Villarrubia pone de manifiesto su prodigiosa habilidad para capturar la magia de Oporto y pa...

Historical Fiction: A Guide to the Genre (Genreflecting ...



Sarah L. Johnson

Unlike guides that take a simple geographical-chronological approach to listing historical fiction, this book takes a genre approa...

New: \$71.25

Strolling Around Porto



Irene Reid

Explore the colourful city of Porto, famous for its Port and its eye-catching six bridges which criss-cross the River Douro. Th...

3.9.2 Sistema de Recomendação da Amazon.com

Na presente aplicação web, em vez de fazer corresponder o utilizador com outros utilizadores, o sistema de recomendação produto-produtos faz corresponder o produto consultado pelo utilizador com produtos similares e combina esses produtos numa lista de recomendação.

Para determinar a maior similaridade de um determinado produto, o algoritmo constrói uma tabela de produtos semelhantes encontrando os produtos que os utilizadores tendem a comprar em conjunto. Podemos construir uma matriz produto-a-produto por iteração todos os pares de produto e calcular uma métrica de similaridade para cada par. No entanto, muitos pares de produtos não têm clientes comuns e, portanto, a abordagem é ineficiente em termos de tempo de processamento e uso de memória. O seguinte algoritmo iterativo fornece uma melhor abordagem por meio do cálculo da similaridade entre um único produto e todos os produtos relacionados: (Linden, Smith, & York, 2003)

```

For each item in product catalog, I1
  For each customer C who purchased I1
    For each item I2 purchased by customer C
      Record that a customer purchased I1 and I2
  For each item I2
    Compute the similarity between I1 and I2

```

É possível calcular a semelhança entre dois produtos de diversas formas, mas um método comum é a utilização da medida de co-seno, em que cada vector corresponde a um produto, e o vector Mdimensions corresponde aos clientes que tenham comprado esse produto.

$$\text{similarity}(\vec{A}, \vec{B}) = \cos(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| * \|\vec{B}\|}$$

O algoritmo gera recomendações com base em outros produtos que são mais similares ao produto seleccionado. Pode medir-se a similaridade entre dois produtos, A e B, de diversas formas; um comum método é medir o co-seno do ângulo entre os dois vectores. (Sarwarm, 2000)

O cálculo da tabela de produtos similares é bastante demorado, com $O(N^2M)$ no pior caso. Na prática, porém, é mais perto de $O(NM)$, porque a maioria dos clientes têm muito poucas compras. Fazendo uma amostragem com clientes que compram os produtos mais vendidos reduz-se bastante o tempo de execução, com pouca redução na qualidade.

Dada uma tabela de itens semelhantes, o algoritmo encontra produtos semelhantes ao que o utilizador está a consultar, agrega esses itens e, em seguida, recomenda os itens mais populares ou relacionados. Este cálculo é muito rápido, dependendo apenas do número de produtos que o utilizador está a consultar, que no caso desta aplicação web é sempre igual a um.

3.9.3 Classe myshop

```
Imports System
```

```
Imports System.Data
```

```
Imports System.ServiceModel
```

```
Imports AmazonECS2011
```

```
Namespace MyShopAmazon
```

```
Public Class MyShopRequest
```

```
Private browseNodeLookup As New BrowseNodeLookup()
```

```
Private browseNodeLookupRequest As New BrowseNodeLookupRequest()
```

```
Private browseNodeLookupResponse As BrowseNodeLookupResponse
```

```
Public Function GetNodeData(nodeId As String, useCache As Boolean) As DataTable
```

```
Dim accessKeyId As String = (System.Configuration.ConfigurationManager.AppSettings("SubscriptionId"))
```

```
Dim sSecretKey As String = (System.Configuration.ConfigurationManager.AppSettings("AmazonSecretKey"))
```

```
Dim AssociateTag As String = (System.Configuration.ConfigurationManager.AppSettings("AssociateTag"))
```

```
Dim awBinding As New BasicHttpBinding(BasicHttpSecurityMode.Transport)
```

```
awBinding.MaxReceivedMessageSize = Int32.MaxValue
```

```
Dim awClient As New AWSECommerceServicePortTypeClient(awBinding, New
```

```
EndpointAddress("https://webservices.amazon.com/onca/soap?Service=AWSECommerceService"))
```

```
'Add authentication to the ECS client
```

```
awClient.ChannelFactory.Endpoint.Behaviors.Add(New AmazonSigningEndpointBehavior(accessKeyId,
sSecretKey))
```

```
' Setting the basic Request parameters
```

```
browseNodeLookup.AWSAccessKeyId = accessKeyId
```

```
browseNodeLookup.AssociateTag = AssociateTag
```

```
' Request Parameters
```

```
browseNodeLookupRequest.BrowseNodeId = New String() {nodeId}
```

```
browseNodeLookupRequest.ResponseGroup = New String() {"TopSellers", "BrowseNodeInfo", "NewReleases"}
```

```
browseNodeLookup.Request = New BrowseNodeLookupRequest(0) {browseNodeLookupRequest}
```

```
If useCache Then
```

```
Dim cacheKey As String = "myshop_browse_" + nodeId
```

```
' Here verify if this request is in the cache
```



```

If System.Web.HttpContext.Current.Cache(cacheKey) Is Nothing Then
    ' Cache key null. We need to send the request and then insert the response in the cache.
    Try
        browseNodeLookupResponse = awClient.BrowseNodeLookup(browseNodeLookup)
        System.Web.HttpContext.Current.Cache.Insert(cacheKey, browseNodeLookupResponse, Nothing,
DateTime.Now.AddDays(30), TimeSpan.Zero)
    Catch e As Exception
        Throw New Exception(e.ToString())
    End Try
Else
    ' Cache key found. Load results from cache.
    browseNodeLookupResponse = DirectCast(System.Web.HttpContext.Current.Cache(cacheKey),
BrowseNodeLookupResponse)
End If
Else
    Try
        browseNodeLookupResponse = awClient.BrowseNodeLookup(browseNodeLookup)
    Catch e As Exception
        Throw New Exception(e.ToString())
    End Try
End If
Dim browseNodesResponse As BrowseNodes() = browseNodeLookupResponse.BrowseNodes
' Verify the response
If browseNodesResponse Is Nothing Then
    Throw New Exception("Server Error - no response received")
End If
If browseNodesResponse(0).Request.Errors IsNot Nothing Then
    Throw New Exception(browseNodesResponse(0).Request.Errors(0).Message)
End If
If browseNodesResponse IsNot Nothing Then
    Dim browseNodes As BrowseNodes = browseNodesResponse(0)
    If browseNodes IsNot Nothing Then
        Dim itemArray As BrowseNode() = browseNodes.BrowseNode
        If itemArray IsNot Nothing Then
            If itemArray(0) IsNot Nothing Then
                Dim NodeDataTable As New DataTable()
                NodeDataTable.Columns.Add(New DataColumn("Name", GetType(String)))
                NodeDataTable.Columns.Add(New DataColumn("ID", GetType(String)))
                NodeDataTable.Columns.Add(New DataColumn("ImageURL", GetType(String)))
                NodeDataTable.Columns.Add(New DataColumn("Price", GetType(String)))
                NodeDataTable.Columns.Add(New DataColumn("Format", GetType(String)))
                NodeDataTable.Columns.Add(New DataColumn("Cat", GetType(String)))
                Try
                    Dim topSellerArray As TopSellersTopSeller() = itemArray(0).TopSellers
                    If topSellerArray IsNot Nothing Then
                        For Each topSeller As TopSellersTopSeller In topSellerArray
                            If topSeller IsNot Nothing Then
                                Dim row As DataRow = NodeDataTable.NewRow()

```

```

If topSeller.Title IsNot Nothing Then
    row("Name") = topSeller.Title
End If
If topSeller.ASIN IsNot Nothing Then
    row("ID") = topSeller.ASIN
    Dim req2 As New ItemDetailsRequest()
    Dim itemsim As Item = req2.GetItemDetails(topSeller.ASIN, True)
    If itemsim.MediumImage IsNot Nothing Then
        row("ImageURL") = itemsim.MediumImage.URL
    Else
        row("ImageURL") = "/content/img/noimage.gif"
    End If
    If itemsim.ItemAttributes.ListPrice IsNot Nothing Then
        row("Price") = itemsim.ItemAttributes.ListPrice.FormattedPrice
    End If
    If itemsim.ItemAttributes.Format IsNot Nothing Then
        row("Format") = itemsim.ItemAttributes.Format(0)
    End If
    If itemsim.Offers IsNot Nothing Then
        Dim offerArray() As Offer = itemsim.Offers.Offer
        If offerArray IsNot Nothing Then
            If offerArray(0).OfferListing IsNot Nothing Then
                If offerArray(0).OfferListing(0).Price IsNot Nothing Then
                    row("Price") = offerArray(0).OfferListing(0).Price.FormattedPrice
                    If row("Price") = "Too low to display" Then
                        row("Price") = "Price available at checkout"
                    End If
                End If
            End If
        End If
    Else
        Dim _amazonURL As String = itemsim.DetailPageURL.ToString()
        row("Price") = "Click <a href='" & _amazonURL & "'>here</a> for price."
    End If
End If
row("Cat") = "TopSellers"
NodeDataTable.Rows.Add(row)
End If
Next

End If
Catch ex As Exception
End Try
Try
    Dim newReleaseArray As NewReleasesNewRelease() = itemArray(0).NewReleases
    If newReleaseArray IsNot Nothing Then
        For Each newRelease As NewReleasesNewRelease In newReleaseArray
            If newRelease IsNot Nothing Then

```

```

Dim row As DataRow = NodeDataTable.NewRow()
If newRelease.Title IsNot Nothing Then
    row("Name") = newRelease.Title
End If
If newRelease.ASIN IsNot Nothing Then
    row("ID") = newRelease.ASIN
    Dim req2 As New ItemDetailsRequest()
    Dim itemsim As Item = req2.GetItemDetails(newRelease.ASIN, True)
    If itemsim.MediumImage IsNot Nothing Then
        row("ImageURL") = itemsim.MediumImage.URL
    Else
        row("ImageURL") = "/content/img/noimage.gif"
    End If
    If itemsim.ItemAttributes.ListPrice IsNot Nothing Then
        row("Price") = itemsim.ItemAttributes.ListPrice.FormattedPrice
    End If
    If itemsim.ItemAttributes.Format IsNot Nothing Then
        row("Format") = itemsim.ItemAttributes.Format(0)
    End If
    If itemsim.Offers IsNot Nothing Then
        Dim offerArray() As Offer = itemsim.Offers.Offer
        If offerArray IsNot Nothing Then
            If offerArray(0).OfferListing IsNot Nothing Then
                If offerArray(0).OfferListing(0).Price IsNot Nothing Then
                    row("Price") = offerArray(0).OfferListing(0).Price.FormattedPrice
                    If row("Price") = "Too low to display" Then
                        row("Price") = "Price available at checkout"
                    End If
                End If
            End If
        End If
    Else
        Dim _amazonURL As String = itemsim.DetailPageURL.ToString()
        row("Price") = "Click <a href='" & _amazonURL & "'>here</a> for price."
    End If
End If
row("Cat") = "NewReleases"
NodeDataTable.Rows.Add(row)
End If
Next
End If
Catch ex As Exception
End Try
Try
    Dim childArray As BrowseNode() = itemArray(0).Children
    If childArray IsNot Nothing Then
        For Each childNode As BrowseNode In childArray
            If childNode IsNot Nothing Then

```

```

'create a datarow, populate it and add it to the table
Dim row As DataRow = NodeDataTable.NewRow()
If childNode.Name IsNot Nothing Then
    row("Name") = childNode.Name
End If
If childNode.BrowseNodeId IsNot Nothing Then
    row("ID") = childNode.BrowseNodeId
End If
row("Cat") = "children"
NodeDataTable.Rows.Add(row)
End If
Next
End If
Catch ex As Exception
End Try
Try
Dim ancestorArray As BrowseNode() = itemArray(0).Ancestors
If ancestorArray IsNot Nothing Then
    ' We use the first node in the array to build the path
    Dim ancestorNode As BrowseNode = ancestorArray(0)
    Dim pathDone As Boolean = False
    While Not pathDone
        'create a datarow, populate it and add it to the table
        Dim row As DataRow = NodeDataTable.NewRow()
        If ancestorNode.Name IsNot Nothing Then
            row("Name") = ancestorNode.Name
        End If
        If ancestorNode.BrowseNodeId IsNot Nothing Then
            row("ID") = ancestorNode.BrowseNodeId
        End If
        row("Cat") = "parents"
        NodeDataTable.Rows.InsertAt(row, 0)
        'NodeDataTable.Rows.Add(row)
        If ancestorNode.Ancestors IsNot Nothing Then
            ancestorNode = ancestorNode.Ancestors(0)
        Else
            pathDone = True
        End If
    End While
    ' We add the current node at the end of the path
    Dim lastRow As DataRow = NodeDataTable.NewRow()
    If itemArray(0).Name IsNot Nothing Then
        lastRow("Name") = itemArray(0).Name
    End If
    If itemArray(0).BrowseNodeId IsNot Nothing Then
        lastRow("ID") = itemArray(0).BrowseNodeId
    End If
    lastRow("Cat") = "current"

```

```
        NodeDataTable.Rows.Add(lastRow)
    End If
    Catch ex As Exception
    End Try
    Return NodeDataTable
End If
End If
End If
End If
Return Nothing
End Function
End Class
End Namespace
```

3.10 CLASSES COMUNS

[Imports](#) Microsoft.VisualBasic
[Imports](#) System
[Imports](#) System.Web
[Imports](#) System.Drawing
[Imports](#) System.Drawing.Imaging
[Imports](#) System.Drawing.Drawing2D
[Imports](#) System.Data
[Imports](#) System.Data.SqlClient
[Imports](#) System.Data.SqlServerCe
[Imports](#) System.IO
[Imports](#) System.Data.SQLite
[Imports](#) System.Text.RegularExpressions
[Imports](#) System.Net
[Imports](#) System.Security.Cryptography
[Imports](#) System.Xml
[Imports](#) System.Xml.Linq
[Imports](#) System.ServiceModel.Syndication

[Public Class](#) LibCommon

' Returns an array of data from the categories

[Public Shared Function](#) XMLReturnArray([ByVal](#) dir [As String](#), [ByRef](#) attribute [As String](#)) [As String](#)()

[Dim](#) Names(0) [As String](#)

[Dim](#) docpath = [HttpContext](#).Current.Server.MapPath("~/App_Data/categories.xml")

[Dim](#) m_xmlId [As XmlDocument](#)

[Dim](#) m_nodelist [As XmlNodeList](#)

[Dim](#) m_node [As XmlNode](#)

'Create the XML Document

m_xmlId = [New XmlDocument](#)()

'Load the Xml file

m_xmlId.Load(docpath)

'Get the list of name nodes

m_nodelist = m_xmlId.SelectNodes("/categories/" & dir & "/cat")

[Dim](#) TempCount [As Integer](#) = 0

'Loop through the nodes

[For Each](#) m_node [In](#) m_nodelist

[ReDim Preserve](#) Names(TempCount)

[Dim](#) NameValue = m_node.Attributes.GetNamedItem(attribute).Value.ToString

Names(TempCount) = NameValue

TempCount += 1

[Next](#)

[Return](#) Names

[End Function](#)

' Returns a string of data from the settings

```
Public Shared Function XMLReturnString(ByVal setting As String) As String
    Dim setpath = HttpContext.Current.Server.MapPath("~/App_Data/settings.xml")
    'Create the XML Document
    Dim setdoc As New XmlDocument()
    'Load the Xml file
    setdoc.Load(setpath)
    'Get the setting value
    Dim strsetting = setdoc.SelectNodes("/settings/" & setting).Item(0).InnerText
    Return strsetting
End Function
```

' Returns a XmlNodeList of data from the categories

```
Public Shared Function XMLReturnNodeList(ByVal dir As String) As XmlNodeList
    Dim mycatpath = HttpContext.Current.Server.MapPath("~/App_Data/categories.xml")
    'Create the XML Document
    Dim mycatdoc As New XmlDocument()
    'Load the Xml file
    mycatdoc.Load(mycatpath)
    'Get the NodeList
    Dim nodelist As XmlNodeList = mycatdoc.SelectNodes("/categories/" & dir & "/cat")
    Return nodelist
End Function
```

'Verify if an item has cover image

```
Public Shared Function HasCover(ByVal cover As String) As Boolean
    If cover = String.Empty Then
        Return False
    Else
        Return True
    End If
End Function
```

'Verify if a file has backup

```
'source http://msdn.microsoft.com/en-us/library/system.io.file.move.aspx
'source http://msdn.microsoft.com/en-us/library/system.io.file.delete.aspx
Public Shared Function BackupFile(ByVal dirname As String, ByRef filename As String, ByRef extension As String) As Boolean
    Dim myPath As String = HttpContext.Current.Server.MapPath("~/ " & dirname & "/")
    Dim myFilePath As String = HttpContext.Current.Server.MapPath("~/ " & dirname & "/" & filename & "." & extension)
    If File.Exists(myFilePath) Then
        'do my work here'
        Dim nowstr As String = DateTime.Now.ToString("-yyyy-MM-dd-hh-mm-ss")
        Dim oldfile As String = filename & "." & extension
        Dim newFile As String = filename & nowstr & "." & extension
    End If
End Function
```

```

System.IO.File.Move(System.IO.Path.Combine(myPath, oldfile), System.IO.Path.Combine(myPath & "bak/",
newFile))
HttpContext.Current.Response.Write("File renamed: " & oldfile & " - " & newFile)
Return True
End If
Return False
End Function

```

'Returns a string to construct a html5 style dropdown-menu

```
Public Shared Function BuildStartMenu(SiteMapTable As DataTable, dir As String) As String
```

```

Dim startmenu As New HtmlGenericControl("ul")
startmenu.Attributes.Add("class", "dropdown-menu")
Dim expression As String = "[Parent] = 0"
Dim foundrows() As DataRow
foundrows = SiteMapTable.Select(expression)
Dim li As New HtmlGenericControl("li")
' Get column 1 & 3 of each returned row.
For i = 0 To foundrows.GetUpperBound(0)
startmenu.Controls.Add(li)
Dim anchor As New HtmlGenericControl("a")
Select Case foundrows(i)(1)
Case "Titles"
anchor.Attributes.Add("href", dir & "/list/titles")
Case "Movies"
anchor.Attributes.Add("href", dir & "/list/movies")
Case Else
anchor.Attributes.Add("href", dir & "/" & foundrows(i)(3))
End Select
anchor.InnerText = foundrows(i)(1)
li.Controls.Add(anchor)
Next i
Dim lidiv As New HtmlGenericControl("li")
startmenu.Controls.Add(lidiv)
lidiv.Attributes.Add("class", "divider")
Dim lihome As New HtmlGenericControl("li")
startmenu.Controls.Add(lihome)
Dim ahome As New HtmlGenericControl("a")
ahome.Attributes.Add("href", "/")
ahome.InnerText = "Home"
lihome.Controls.Add(ahome)
Dim stringwriter As New System.IO.StringWriter()
Dim writer As New HtmlTextWriter(stringwriter)
startmenu.RenderControl(writer)
Dim videov = New HtmlString(writer.ToString())
Return stringwriter.ToString()
End Function

```

'Returns a string of data from a SQL database


```

Public Shared Function ExecuteScalarSqlClient(sql As String, connString As String) As String
    Try
        Using conn As New SqlConnection(connString)
            conn.Open()
            Using cmd As New SqlCommand(sql, conn)
                cmd.CommandText = sql
                cmd.CommandType = CommandType.Text
                Dim value As Object = cmd.ExecuteScalar()
                If value IsNot Nothing Then
                    Return value.ToString()
                Else
                    Return ""
                End If
            End Using
        End Using
    Catch ex As Exception

    End Try
    Return Nothing
End Function

```

'Returns a datatable from a SQL database

```

Public Shared Function GetDataTableSqlClient(sql_str As String, connString As String) As DataTable
    Dim dataTable As New DataTable()
    Using connection As New SqlConnection(connString)
        Using cmd As New SqlCommand(sql_str, connection)
            cmd.CommandType = CommandType.Text
            Using adapter As New SqlDataAdapter(cmd)
                adapter.Fill(dataTable)
            End Using
        End Using
    End Using
    Return dataTable
End Function

```

'Returns a datatable from a SQL Compact database

```

Public Shared Function GetDataTableSqlCE(sql_str As String, connString As String) As DataTable
    Dim connStringlocal As String = ConfigurationManager.ConnectionStrings("MyPics").ConnectionString
    Dim dataTable As New DataTable()
    Using connection As New SqlCeConnection(connStringlocal)
        Using cmd As New SqlCeCommand(sql_str, connection)
            cmd.CommandType = CommandType.Text
            Using adapter As New SqlCeDataAdapter(cmd)
                adapter.Fill(dataTable)
            End Using
        End Using
    End Using
    Return dataTable

```

End Function

'Returns a datatable from a SQL LocalDB database

```
Public Shared Function GetDataTableSqlCEMT(sql_str As String, connString As String) As DataTable
    Dim connStringlocal As String = ConfigurationManager.ConnectionStrings("MyTravels").ConnectionString
    Dim dataTable As New DataTable()
    Using connection As New SqlConnection(connStringlocal)
        Using cmd As New SqlCommand(sql_str, connection)
            cmd.CommandType = CommandType.Text
            Using adapter As New SqlDataAdapter(cmd)
                adapter.Fill(dataTable)
            End Using
        End Using
    End Using
    Return dataTable
End Function
```

'Returns a string of data from a SQLite database

```
Public Shared Function ExecuteScalarSQLite(sql As String, connString As String) As String
    Try
        Using conn As New SQLiteConnection(connString)
            conn.Open()
            Using cmd As New SQLiteCommand(conn)
                cmd.CommandText = sql
                cmd.CommandType = CommandType.Text
                Dim value As Object = cmd.ExecuteScalar()
                If value IsNot Nothing Then
                    Return value.ToString()
                Else
                    Return ""
                End If
            End Using
        End Using
    Catch ex As Exception
        HttpContext.Current.Response.Write(sql)
        HttpContext.Current.Response.Write(ex)
    End Try
    Return Nothing
End Function
```

'Returns a datatable from a SQLite database

```
Public Shared Function GetDataTableSQLite(sql_str As String, connString As String) As DataTable
    Dim dataTable As New DataTable()
    Using connection As New SQLiteConnection(connString)
        Using cmd As New SQLiteCommand(sql_str, connection)
            cmd.CommandType = CommandType.Text
            Using adapter As New SQLiteDataAdapter(cmd)
                adapter.Fill(dataTable)
            End Using
        End Using
    End Using
End Function
```

```
End Using
End Using
End Using
Return dataTable
End Function
```

'Returns a string with the last parameter from a strURI

```
Public Shared Function GetLastURLdata(ByVal strURI As String) As String
    Dim lastcharright = Right(strURI, 1)
    If lastcharright = "/" Then
        strURI = strURI.Substring(0, strURI.Length - 1)
    End If
    Try
        Dim urladdress() = Split(strURI, "/")
        Dim lasturl = "none"
        For i As Integer = 0 To urladdress.Length - 1
            lasturl = urladdress(i)
        Next
        Return lasturl
    Catch ex As Exception
        Return ex.Message
    End Try
End Function
```

'Returns a string with the contents of a webpage for Amazon Reviews

```
Public Shared Function GetWebPageAsString(ByVal strURI As String) As String
    Dim sStream As Stream
    Dim URLReq As HttpWebRequest
    Dim URLRes As HttpWebResponse
    Try
        URLReq = WebRequest.Create(strURI)
        URLRes = URLReq.GetResponse()
        sStream = URLRes.GetResponseStream()
        Return New StreamReader(sStream).ReadToEnd()
    Catch ex As Exception
        Return ex.Message
    End Try
End Function
```

'Returns a condensed string

```
Public Shared Function GetCondensed(ByVal str As String, ByVal size As Integer) As String
    Dim returnValue As String = ""
    If str <> "UC" Then
        If Len(str) < size Then
            returnValue = str
        Else
            returnValue = Left(str, size) + "..."
        End If
    End If
```

```

End If
Return returnValue
End Function

```

'Returns a condensed string with a link to read the full content

```

Public Shared Function GetReadMore(ByVal Desc As String, ByVal size As Integer, ByVal dir As String, ByVal id As
String, ByVal title As String) As String
Dim returnValue As String = ""
If Desc <> "UC" Then 'UC = Under Construction
If Len(Desc) < size Then
returnValue = Desc
Else
returnValue = Left(Desc, size) + "... <a href='/' & dir & "/details/" & id.ToString & "/" & title & ">Read
More</a>"
End If
End If
Return returnValue
End Function

```

'Returns a string without some html tags

```

Public Shared Function StripAllTags(ByVal html As String) As String
' Remove HTML tags.
Return Regex.Replace(html, "<.*?>", "")
End Function

```

'Returns a string without any html tags

```

Public Shared Function RemoveHTML(ByVal html As String) As String
" remove comments
html = Regex.Replace(input, "<!--(.|\s)*?-->", String.Empty)
" remove HTML
Return Regex.Replace(input, "<(.\s)*?>", String.Empty)
End Function

```

'Returns a string without any illegal characters

```

Public Shared Function StripAllIllegal(ByVal html As String) As String
'to keep
Regex.Replace(html, "[^0-9a-zA-Z ]+?", "-")
'to remove
Regex.Replace(html, "[-\#!\£\$\%^&*()\|\\ ]+?", "-")
Return html
End Function

```

'Returns a string without some formatting html tags

```

Public Shared Function StripTags(ByVal html As String) As String
If Not html = String.Empty Then
html = Regex.Replace(html, "</?[hH]3[^\>]*>", "")
html = Regex.Replace(html, "</?[div[^\>]*>", "")
html = Regex.Replace(html, "</?[DIV[^\>]*>", "")

```

```

html = Regex.Replace(html, "</?[i[^>]*>", "")
html = Regex.Replace(html, "</?[p[^>]*>", "")
html = Regex.Replace(html, "</?[b[^>]*>", "")
html = Regex.Replace(html, "</?[P[^>]*>", "")
html = Regex.Replace(html, "</?[B[^>]*>", "")
html = Regex.Replace(html, "</?[em[^>]*>", "")
html = Replace(html, """"", "")

```

```
End If
```

```
Return html
```

```
End Function
```

'Returns a string without some special characteres

```
Public Shared Function StripSlug(ByVal str As String) As String
```

```

str = Replace(str, " ", "-")
str = Replace(str, ":", "")
str = Replace(str, "+", "")
str = Replace(str, "& ", "")
str = Replace(str, "&", "")
str = Replace(str, ".", "")
str = Replace(str, "/", "")
str = Replace(str, "--", "-")

```

```
Return str
```

```
End Function
```

'Returns a string with spaces replaced by hiffens for url

```
Public Shared Function UnStripSlug(ByVal str As String) As String
```

```
str = Replace(str, "-", " ")
```

```
Return str
```

```
End Function
```

'Returns the number of pages for a search

```
Public Shared Function CalculateNumberOfPages(totalRows As Integer, pageSize As Integer) As Integer
```

```
Dim result = totalRows Mod pageSize
```

```
If result = 0 Then
```

```
Return totalRows / pageSize
```

```
Else
```

```
Return Int(totalRows / pageSize + 1)
```

```
End If
```

```
End Function
```

'Returns an image resized

```
Public Shared Function ResizeImageFile(ByVal imageFile() As Byte, ByVal targetSize As Integer) As Object
```

```
Using oldImage As System.Drawing.Image = System.Drawing.Image.FromStream(New MemoryStream(imageFile))
```

```
Dim newSize As Size = CalculateDimensions(oldImage.Size, targetSize)
```

```
Using newImage As Bitmap = New Bitmap(newSize.Width, newSize.Height, PixelFormat.Format24bppRgb)
```

```
Using canvas As Graphics = Graphics.FromImage(newImage)
```

```
canvas.SmoothingMode = SmoothingMode.AntiAlias
```

```
canvas.InterpolationMode = InterpolationMode.HighQualityBicubic
```

```
canvas.PixelOffsetMode = PixelOffsetMode.HighQuality
canvas.DrawImage(oldImage, New Rectangle(New Point(0, 0), newSize))
Dim m As New MemoryStream
newImage.Save(m, ImageFormat.Jpeg)
Return m.GetBuffer
End Using
End Using
End Using
End Function
```

'Returns the size for an image

```
Public Shared Function CalculateDimensions(ByVal oldSize As Size, ByVal targetSize As Integer) As Size
    Dim newSize As Size
    If (oldSize.Height > oldSize.Width) Then
        newSize.Width = CType((oldSize.Width * CType((targetSize / CType(oldSize.Height, Single)), Single)), Integer)
        newSize.Height = targetSize
    Else
        newSize.Width = targetSize
        newSize.Height = CType((oldSize.Height * CType((targetSize / CType(oldSize.Width, Single)), Single)), Integer)
    End If
    Return newSize
End Function
```

4 CONCLUSÃO E DESENVOLVIMENTOS FUTUROS

Tal como está referido na proposta do projecto, este pretende implementar uma aplicação web que permitisse melhorar a experiência de visualização e navegação dos utilizadores em qualquer dispositivo dispondo de um *browser* moderno. O objectivo principal de implementar uma aplicação web (e dos seus diversos módulos) visualizável em qualquer dispositivo moderno foi cumprido, bem como os restantes objectivos tais como, a implementação e configuração do servidor *Web* em *Windows Azure*, aprofundamento de competências de programação em *.Net Framework*, realização de testes aos módulos desenvolvidos e desenvolvimento de competências de redacção de relatórios.

A parte do projecto que constava realizar de forma independente consistiu em desenhar e implementar uma aplicação web para qualquer dispositivo. No plano de trabalho constavam cinco módulos temáticos que foram todos desenvolvidos.

Este projecto representou um desafio muito aliciante devido à necessidade de aplicar um conjunto bastante largo de conhecimentos adquiridos ao longo do curso e pela possibilidade de desenvolver outros. Sendo este um curso de Licenciatura Informática, um projecto como este é o ideal para o aluno realizar. Para além do projecto ter tido uma componente de instalação e configuração de sistemas, permitiu apurar conhecimentos sobre programação para *.NET*, *VB*, *C#*, *p.Net/Ajax/Java/HTML5/JQuery*, *SQL*, *Web Services* e dispositivos móveis. Contemplando parte de matérias de disciplinas como *Sistemas e Serviços Web*, *Bases de Dados*, *Programação por Objectos*, *Estruturas e algoritmos*, *Linguagens de Programação*, *Sistemas Operativos*, *Redes*, *Sistemas Distribuídos*, *Compiladores*, *Gestão de Projectos Informáticos*.

Como em qualquer projecto existiram obviamente algumas dificuldades na implementação da aplicação idealizada. É difícil enumerá-las todas, no entanto destaca-se o facto de o projecto ter um período de desenvolvimento curto (cerca de quatro meses com o relatório). Algumas opções tiveram de ser tomadas rapidamente e nem sempre foi feita a devida ponderação. O importante é que todas as dificuldades foram superadas.

DESENVOLVIMENTOS FUTUROS

Nesta fase, a configuração da aplicação necessita de ser feita manualmente através da edição dos ficheiros XML. Está prevista a implementação de um módulo opcional para configuração da aplicação. O manual de utilizador será objecto de elaboração posterior, uma vez que a sua execução depende das funcionalidades a incluir no módulo de configuração.

5 REFERÊNCIAS BIBLIOGRÁFICAS

- Anderson, R. (2012). *Bundling and Minification*. Retrieved 2014, from ASP.NET:
<http://www.asp.net/mvc/tutorials/mvc-4/bundling-and-minification>
- FitzMacken, T. (2014). *Creating a Consistent Layout in ASP.NET Web Pages (Razor) Sites*. Retrieved 2014, from ASP.NET: <http://www.asp.net/web-pages/tutorials/working-with-pages/3-creating-a-consistent-look>
- Gardner, B. S. (2011). Responsive Web Design: Enriching the User Experience. *Sigma Dig Eco*, //, pp. 13-19.
- Guthrie, S. (2010). *Automating Deployment with Microsoft Web Deploy*. Retrieved 2014, from ASP.NET: <http://weblogs.asp.net/scottgu/automating-deployment-with-microsoft-web-deploy>
- Hunt, L. (Ed.). (2010). *HTML5 Reference*. Retrieved 2014, from dev.w3.org:
<http://dev.w3.org/html5/html-author/>
- Linden, G., Smith, B., & York, J. (2003). Amazon.com Recommendations - Item-to-Item Collaborative Filtering. *IEEE Internet Computing*, 76-79.
- Marcotte, E. (2009). *Fluid Grids*. Retrieved 2014, from alistapart.com:
<http://alistapart.com/article/fluidgrids>
- Marcotte, E. (2011). *Fluid Images*. Retrieved 2014, from alistapart.com:
<http://alistapart.com/article/fluid-images>
- Marcotte, E. (2011). *Responsive Web Design*. New York: A Book Apart.
- Mozilla Developer Network. (2014). *JavaScript*. Retrieved 2014, from Mozilla Developer Network: <http://en.wikipedia.org/w/index.php?title=JavaScript>
- MSDN. (2014). *Accessibility in Visual Studio and ASP.NET*. Retrieved 2014, from MSDN:
[http://msdn.microsoft.com/en-us/library/vstudio/ms228004\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/ms228004(v=vs.100).aspx)
- MSDN. (2014). *ASP.NET Caching Overview*. Retrieved 2014, from MSDN:
[http://msdn.microsoft.com/en-us/library/vstudio/ms178597\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/ms178597(v=vs.100).aspx)
- MSDN. (2014). *ASP.NET data binding overview*. Retrieved 2014, from MSDN:
<http://support.microsoft.com/kb/307860>

- MSDN. (2014). *ASP.NET Globalization and Localization*. Retrieved 2014, from MSDN:
[http://msdn.microsoft.com/en-us/library/vstudio/c6zzy3s9\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/c6zzy3s9(v=vs.100).aspx)
- MSDN. (2014). *ASP.NET Web Pages Overview*. Retrieved 2014, from MSDN:
[http://msdn.microsoft.com/en-us/library/vstudio/428509ah\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/428509ah(v=vs.100).aspx)
- MSDN. (2014). *ASP.NET Web Pages Overview*. Retrieved 2014, from MSDN:
[http://technet.microsoft.com/en-ca/428509ah\(v=vs.95\).aspx](http://technet.microsoft.com/en-ca/428509ah(v=vs.95).aspx)
- MSDN. (2014). *Building (Compiling) Web Site Projects*. Retrieved 2014, from MSDN:
[http://msdn.microsoft.com/en-us/library/vstudio/hwxa5aha\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/hwxa5aha(v=vs.100).aspx)
- MSDN. (2014). *Microsoft Ajax Overview*. Retrieved 2014, from MSDN:
[http://msdn.microsoft.com/en-us/library/vstudio/bb398874\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/bb398874(v=vs.100).aspx)
- MSDN. (2014). *Planning an ASP.NET Web Site*. Retrieved 2014, from MSDN:
[http://msdn.microsoft.com/en-us/library/bb470439\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/bb470439(v=vs.90).aspx)
- Sarwarm, B. (2000). Analysis of Recommendation Algorithms for E-Commerce. *ACM Conf. Electronic Commerce* (pp. 158-167). ACM Press.
- Souza, R. G. (2014). *Sistemas de Recomendação - Aplicando Sistemas de Recomendação em Situações Práticas*. Retrieved 2014, from IBM Developer Works:
http://www.ibm.com/developerworks/br/local/data/sistemas_recomendacao/
- W3C. (2014). *Web Content Accessibility Guidelines Working Group*. Retrieved 2014, from W3C.org: Web Content Accessibility Guidelines Working Group
- Weinstein, J. (2014). *Let's Make the Web Faster*. Retrieved 2014, from Google Code:
<https://developers.google.com/speed/articles/mobile?csw=1>
- Wikipedia contributors. (n.d.). *LESS*. Retrieved 2014, from Wikipedia, The Free Encyclopedia:
<http://en.wikipedia.org/w/index.php?title=LESS>
- Work, S. (2011). *How Loading Time Affects Your Bottom Line*. Retrieved 2014, from KISSmetrics Marketing Blog: <http://blog.kissmetrics.com/loading-time>