

PROGRAMAR

REVISTA PORTUGUESA DE PROGRAMAÇÃO • WWW.PORTUGAL-A-PROGRAMAR.ORG

EDIÇÃO #27 - FEVEREIRO 2011

ISSN 1647-0710

Microsoft

ASP.net

MVVC3



ANALISE DO LIVRO
SILVERLIGHT 4.0 CURSO COMPLETO



VISUAL (NOT) BASIC
WINDOWS PHONE 7

LUA 7ª PARTE
: LINGUAGEM DE PROGRAMAÇÃO

FLEX e BYACC 3ª PARTE

OPTIMIZAÇÃO DE SQL EM ORACLE

JQUERY 2ª PARTE



6 REGRAS PARA UTILIZAR **AJAX**

GCC - WALL MYAPP.C
-INSTANTANEOUS -O SUCCESS **CORE DUMP**

EQUIPA PROGRAMAR

Coordenadores

António Silva
Fernando Martins

Editor

António Silva

Design

Sérgio Alves (@scorpion_blood)

Redacção

Augusto Manzano, Caio Proiete, Carlos Gonçalves, CoolThingsPT, Fernando Martins, João Mares, Paulo Morgado, Ricardo Rodrigues, Ricardo Trindade, Rodrigo Pinto, Sara Barbosa, Sara Silva, Sérgio Ribeiro

Staff

António Santos, Fábio Domingos, Jorge Paulino, Marco Marques

Contacto

revistaprogramar@portugal-a-programar.info

Website

<http://www.revista-programar.info>

ISSN

1647-0710

Expressa-te!

Sem dúvida que nos últimos tempos têm-se falado bastante na liberdade de expressão, como os casos de divulgação de documentos por parte do Wikileaks, as “milenarios” censuras na China no uso da Internet (agora até estão censuradas pesquisas sobre o actual estado do Egipto), a tentativa de obter os dados de acesso às contas do Facebook dos tunisinos por parte do próprio governo, para limitar a troca e partilha de informações, e ainda mais recentemente o corte dos serviços de internet no Egipto aplicado a toda a população, para que esta não possa comunicar sobre a situação actual do país (entretanto, e pelo menos parcialmente o corte foi cancelado), o caso da Ensitel que queria obrigar uma cliente a retirar a informação sobre as reclamações que tinha feito para a empresa, entre muitos outros que poderia enumerar.

Na realidade ao longo dos séculos a denominada censura sempre existiu, mesmo em países designados como altamente evoluídos. Porque se me apetecer colocar online dados de outra pessoa sem a sua autorização não o posso fazer, e que é isso senão uma forma de censura? No entanto essa censura parece razoável porque o seu intuito é proteger os cidadãos. A grande questão é, até que ponto pode ir essa censura? No caso do Wikileaks essa é uma questão com muitas pontas. Afinal, grande parte da informação revelada lá é considerada sigilosa nos países onde foi obtida, nomeadamente os EUA. Na verdade a proibição de reproduzir informação sem autorização considerada secreta, nada mais é que censura. Mas aqui parece mais difícil avaliar se esta censura é a favor dos cidadãos ou contra eles. E aí as opiniões e os argumentos dividem-se.

Com a evolução, aquilo que muitos denominam de Web 2.0 fez muitas pessoas descobrirem que o que dizem ou fazem pode ter repercussões em todo mundo, gerando ondas de solidariedade ou aversão, mas abanando as estruturas da sociedade. O facto de cada um de nós poder ser também um meio de informação (ou por vezes desinformação) traz enormes responsabilidades. Quando foi lançada a suspeita que a licenciatura do Engº Sócrates tinha sido tirada ao Domingo, a página da Wikipédia referente à sua pessoa foi bloqueada por edições abusivas. A verdade é que com este poder muitos caem na tentação de apresentar as suas opiniões como se fossem factos consumados, e isso gera o famoso “diz que disse”.

António Silva

TEMA DE CAPA

- 6 Introdução ao ASP .NET MVC 3.0

A PROGRAMAR

- 13 Lua – Linguagem de Programação (Parte 7)
- 22 Flex e Byacc - Parte III
- 26 Optimização de SQL em Oracle - Índices
- 31 jQuery – A framework

COLUNAS

- 38 PROGRAMADOR EXCÊNTRICO - 6 regras para utilizar AJAX
- 40 VISUAL (NOT) BASIC - Windows Phone 7
- 47 CORE DUMP - gcc -Wall myApp.c -linstantaneous -o success

COMUNIDADES

- 51 Desenvolvimento em SharePoint 2010
- 59 GuiaTV CoolthingsPT
- 62 Habilitar external users no SharePoint online – Office365
- 66 O Futuro da Programação Assíncrona da Plataforma .NET

ANÁLISES

- 70 Silverlight 4.0 – Curso Completo

EVENTOS

- 18 Fev. **ESAF LanParty 2011**
- 18 Fev. **5º Portugal Girl Geek Dinner**
- 21 Fev. **Windows Azure BootCamp Lisboa 2011 - Lisboa**
- 23 Fev. **Visual Studio S2010 - Workshop Desenvolvimento Apicacional - Porto**
- 28 Fev. **Microsoft Business Integration RoadShow - Lisboa**
- 04 Mar. **XNA Pizza Night 2011 Spring Edition**
- 05 Mar. **3º Encontro WordPress Portugal em Lisboa**
- 26 Mar. **Sessão Presencial NetPonto - Lisboa**

ZX Spectrum: No futuro, vamos regressar ao passado

Que uma parte do mundo está em modo revivalista dos eighties já toda a gente percebeu. Da roupa aos acessórios, passando pela música e pelos carros, as tendências vão desaguar ao período da História que consagrou Madonna como ícone. Sem surpresas, os videojogos retro também voltaram à ribalta, a ponto de a empresa de programação Elite ter decidido reeditar o ZX Spectrum, que fez as delícias daqueles que agora andam na casa dos 30.

Os actuais endereços de Internet estão a acabar

No início do mês, a entidade americana responsável pelos endereços de Internet, chamada IANA, anunciou que tinha atribuído os últimos endereços disponíveis, o que deu azo a várias notícias sobre o esgotamento da Internet

Devido ao término dos blocos IPv4 disponíveis, os RIRs (Regional Internet Registries), entidades regionais administradoras mundiais desses endereços, terão que buscar métodos para administrar apenas os endereços que já existem. Isso pode virar um transtorno devido ao aumento do número de usuários e serviços que solicitam novos endereços. No continente asiático, essa diminuição nos endereços será sentida em larga escala, pois o número de acessos e serviços relacionados a grande rede cresce vertiginosamente.

"O IPv4 está, oficialmente, esgotado. Longa vida ao IPv6."

Portugal sobe de posição no ranking de inovação europeu

Nos últimos cinco anos Portugal subiu sete posições no ranking europeu de inovação, passando este ano à frente de Espanha e da Itália e liderando o grupo de inovadores moderados. Conquistando o 15º lugar do European Innovation Scoreboard de 2010, o país continua, ainda assim, abaixo da média na maioria dos indicadores.

O [estudo já está online](#) e vai ser apresentado hoje em Bruxelas mas os traços gerais já foram divulgados, mostrando que a Suécia, Dinamarca, Finlândia e Alemanha continuam no pelotão da frente. Portugal conquista porém o troféu de ser o país que mais subiu nos últimos cinco anos, sobretudo devido ao crescimento de empresas inovadoras e à melhoria do ensino de nível secundário.

PT recebe prémio europeu de fibra óptica

Portugal Telecom (PT) recebeu hoje um prémio que reconhece a rede de fibra óptica da operadora como «a melhor e mais inovadora na Europa», anunciou hoje a empresa em comunicado.

«Este prémio vem reconhecer o elevado nível de excelência do trabalho, empenho e dedicação das equipas da PT, e o sucesso da estratégia que adoptámos, que privilegia a inovação, a execução operacional e parcerias com líderes da indústria, de modo a podermos oferecer aos nossos clientes em Portugal os melhores e mais avançados serviços disponíveis no mercado», disse o presidente-executivo, Zeinal Bava.

Cerca de um milhão de casas em Portugal têm já acesso à fibra óptica da PT. O objectivo da operadora é reforçar a oferta em mais 600 mil casas até ao final do ano.

TEMA DE CAPA

Introdução ao ASP .NET MVC 3.0

Introdução ao ASP .NET MVC 3.0

Desde o lançamento da plataforma Microsoft .NET, passamos a contar com um framework de desenvolvimento de aplicações Web chamado ASP .NET, que oferece diversos componentes de base para a criação destas aplicações, como por exemplo, autenticação, autorização, controlo de sessões dos utilizadores, cookies, entre outras funcionalidades, juntamente com um modelo de programação baseado em páginas e eventos que permite desenvolver aplicações Web de uma forma muito semelhante ao desenvolvimento de aplicações Windows Forms.

Este modelo de programação chama-se ASP .NET WebForms, e era o único modelo existente no framework ASP .NET até o início de 2009, quando foi então lançado pela Microsoft a primeira versão do ASP .NET MVC, que é uma alternativa ao modelo WebForms existente, e oferece um modelo de programação muito mais simplificado e directo, e baseado num padrão (design pattern) bastante conhecido e utilizado em vários outros frameworks e plataformas, chamado Model-View-Controller (MVC).

“ ASP .NET MVC (...) oferece um modelo de programação muito mais simplificado e directo, e baseado num padrão bastante conhecido (...) ”

Assim, actualmente um developer que queira desenvolver uma aplicação Web na plataforma Microsoft .NET tem à sua disposição duas alternativas ou modelos de programação que pode escolher de acordo com a sua preferência: ASP .NET WebForms e ASP .NET MVC.

A versão mais recente do ASP .NET MVC no

momento em que escrevo este artigo é a 3.0 RTM, disponibilizada recentemente pela Microsoft, e que está disponível para download gratuitamente através do seguinte endereço:

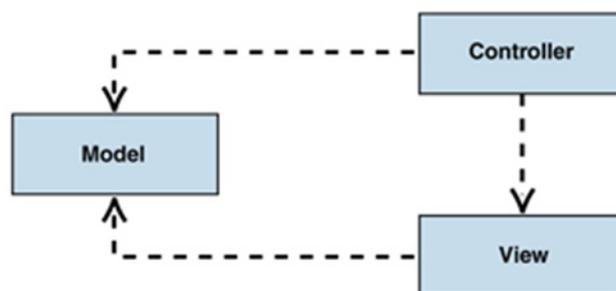
<http://tinyurl.com/downloadmvc3>

É importante destacar que o ASP .NET MVC 3.0 utiliza novos recursos que fazem parte do .NET Framework 4.0, e por isso necessita ter esta versão do .NET Framework e o Visual Studio 2010 para desenvolver aplicações (não é possível utilizar o Visual Studio 2008).

O Visual Studio 2010 não inclui o ASP .NET MVC 3.0, mas sim o ASP .NET MVC 2.0, portanto a versão 3.0 deve ser instalada manualmente.

Model-View-Controller (MVC)

O padrão Model-View-Controller ou MVC permite separar o código necessário para interagir com a interface com utilizador através de três conceitos: Model, View e Controller, onde cada um tem uma responsabilidade específica.



No ASP .NET MVC, um Controller corresponde a uma classe que pode ser responsável por receber as informações solicitadas pelo cliente (navegador), e enviar uma resposta num formato apropriado, de acordo com a solicitação que recebeu.

“ O padrão Model-View-Controller ou MVC permite separar o código necessário para interagir com a interface com utilizador através de três conceitos: Model, View e Controller ”

O recebimento das informações acontece através de métodos existentes nesta classe (Controller), chamados Action. Um Controller pode possuir uma ou mais acções (Actions) possíveis de serem executadas, e cada Action é então responsável por receber as informações enviadas através do cliente (navegador) efectuar qualquer processamento que seja necessário e então enviar a resposta apropriada.

Para enviar esta resposta, a Action do Controller interage com o modelo aplicacional (Model) geralmente composto de várias classes que implementam a lógica de negócio da aplicação, e desta forma, o Controller consegue preparar a resposta a ser enviada de volta para o cliente (navegador).

Esta resposta normalmente corresponde a uma View, cuja responsabilidade é construir a resposta ao pedido solicitado, no formato apropriado. As Views normalmente são compostas por código HTML e JavaScript, misturado com trechos de código C# ou VB .NET utilizando uma sintaxe chamada Razor.

Este modelo de programação traz uma série de benefícios aos developers, como:

- É possível ter diferentes representações (Views) de uma mesma informação (Model);
- Novas interfaces com o utilizador (Views) podem ser facilmente adicionadas, removidas ou alteradas, sem grande impacto ao restante da aplicação;
- A resposta a uma requisição pode ser facilmente alterada (pois está centralizada num Controller);

- Promove a reutilização de código (Por exemplo: Uma mesma View pode ser utilizada com diferentes Models);
- E provavelmente o mais importante: Ajuda os developers a manter o foco em um único aspecto da aplicação, de cada vez.

Funcionamento do ASP .NET MVC 3

Quando o framework inicia o processamento de uma requisição, obtém um determinado endereço (URL) que é então utilizado para determinar qual será o Controller que será responsável por responder a esta requisição, e qual será a Action deste Controller que será executada. Por exemplo:

<http://portugal-a-programar.org/revista/edicoes>

No exemplo acima, o framework ASP .NET MVC analisa o texto da URL que está após o endereço do site, e identifica “revista” como sendo o nome do Controller que deve ser responsável por responder a esta requisição, e “edicoes” como sendo o nome do método (Action) a ser executado neste Controller.

Para que esta requisição seja processada com sucesso, é preciso que um dos assemblies do projecto (ou em assemblies referenciados) possua uma classe com as características abaixo, caso contrário a resposta enviada será um erro do tipo “404 Not Found”:

- Precisa ser uma classe pública (Public);
- Precisa ser uma classe concreta (deve ser possível instanciar objectos desta classe... Não pode ser uma classe abstrata (abstract), ou uma interface);
- Precisa ser uma classe derivada da classe System.Web.Mvc.Controller;
- Precisa ser uma classe que tenha o mesmo nome identificado através da URL como sendo o nome da Controller, seguido da palavra “Controller”, e precisa ter um método com o mesmo nome identificado na URL como sendo a Action a ser executada, e que retorne um objecto do tipo

TEMA DE CAPA

Introdução ao ASP .NET MVC 3.0

System.Web.Mvc.ActionResult ou de uma classe derivada desta.

No exemplo acima, a classe responsável por processar este tipo de requisições (<http://portugal-a-programar.org/revista/edicoes>) deve chamar-se RevistaController, e deve possuir um método chamado Edicoes.

```
public static void
RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapRoute(
        "Default", // Route name
        "{controller}/{action}/{id}", // URL
        with parameters
        new { controller = "Home", action =
        "Index", id = UrlParameter.Optional } //
        Parameter defaults
    );
}
```

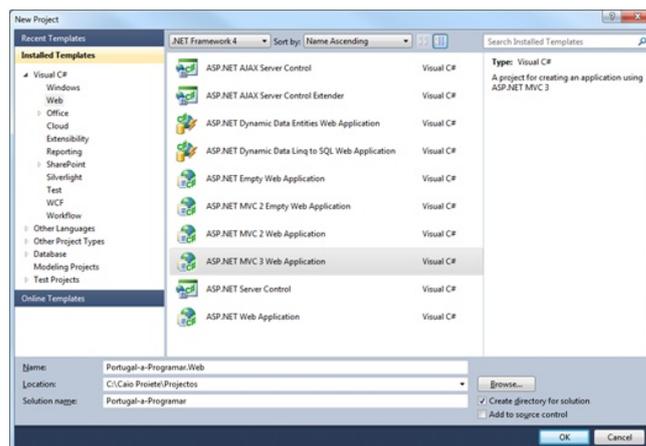
Caso o nome do Controller e/ou da Action não esteja presente no endereço (URL), a aplicação ASP .NET MVC assume os valores “Home” para Controller e “Index” para Action, ou seja, para uma requisição do tipo <http://portugal-a-programar.org/> onde não são informados os valores para Controller e Action, espera-se que exista uma classe chamada “HomeController” que por sua vez possua um método chamado “Index”.

Esta é uma convenção padrão que está definida no ficheiro Global.asax e que pode ser alterada se desejar:

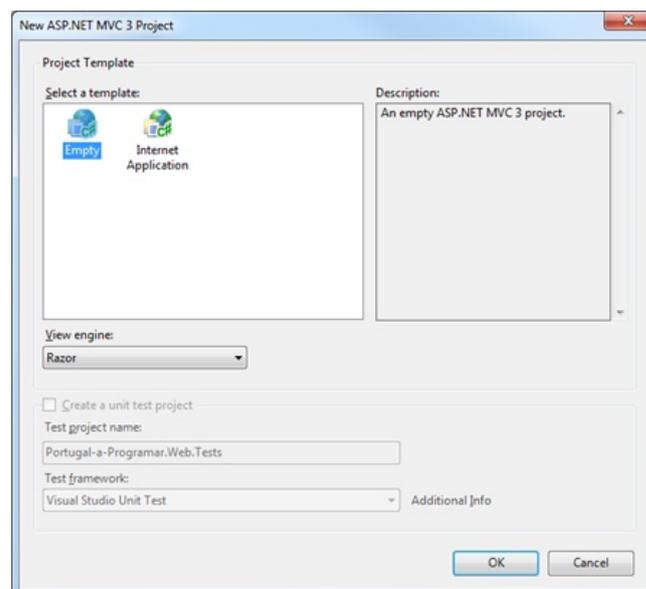
Criando o seu primeiro projecto ASP .NET MVC 3

Para iniciar um novo projecto ASP .NET MVC 3, a

partir do Visual Studio 2010 deve aceder ao menu File, seleccionar New, e então Project, onde poderá seleccionar o tipo de projecto “ASP.NET MVC 3 Web Application”, e definir o nome do projecto e o local onde será gravado.



A seguir será apresentado um ecrã onde onde pode escolher o template que deseja utilizar para o novo projecto, o View Engine (mecanismo que será utilizado para construir as Views), e tem a opção de criar um projecto de testes unitários para acompanhar este novo projecto que está a ser criado.



Os templates que pode escolher são: “Empty” e “Internet Application”.

A principal diferença entre os dois é que o “Internet Application” já inclui algumas classes, controllers e

views de exemplo que podem ser utilizadas como base para o desenvolvimento da aplicação, enquanto que o “Empty” apenas uma estrutura mínimo necessária para começar.

“ O Razor é a grande novidade do ASP.NET MVC 3 e possui uma sintaxe simplificada e (...) mais elegante ”

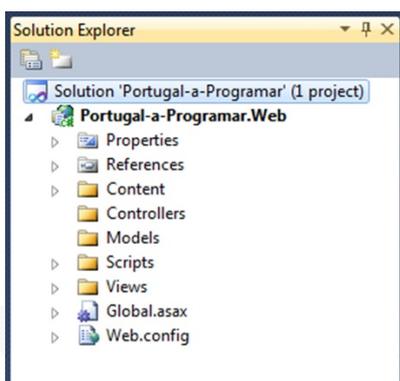
Os View Engines que pode escolher são “ASPX” e “Razor”. O View Engine define, entre outras coisas, a linguagem que será utilizada para a construção das Views, onde o mistura-se código HTML e JavaScript com código C# (ou VB .NET).

ASPX é o View Engine que foi introduzido na primeira versão do ASP .NET, juntamente com o modelo WebForms e possui uma sintaxe que utiliza tags do tipo “<% ... %>” para identificar os trechos de código.

Já o Razor é a grande novidade do ASP .NET MVC 3 e possui uma sintaxe simplificada e considerada mais elegante, pelo que é o View Engine padrão do ASP.NET MVC 3.

Para o exemplo explicado neste artigo, foi seleccionado o template “Empty” e o View Engine “Razor”.

Ao criar o novo projecto “ASP.NET MVC 3 Web Application” é criada uma estrutura como mostra a figura abaixo:



- Content: Pasta onde deve colocar as imagens e estilos (CSS) utilizados no projecto;
- Controllers: Pasta onde deve colocar todos os Controllers da aplicação;
- Models: Pasta onde deve colocar os serviços, repositórios e outras classes que implementam a lógica da aplicação;
- Scripts: Pasta onde deve colocar os ficheiros JavaScript utilizados no projecto;
- Views: Pasta onde deve colocar os ficheiros responsáveis por construir as Views;

Esta é uma estrutura sugerida e pode ser completamente alterada se desejar (necessita de alguns tweaks em alguns casos).

Criação da página inicial do site

Para criar a página inicial (home-page) da aplicação Web, o primeiro passo é criar o controller chamado “HomeController” (que é uma classe com este nome), que será responsável por receber/responder requisições feitas para a página principal do site.

Para isto, clique com o botão direito na pasta Controllers no Solution Explorer, seleccione “Add” e então “Controller...”. No nome do controller defina “HomeController” e seleccione “Add”.

Será criado um ficheiro com o nome “HomeController.cs” (ou .vb se utiliza Visual Basic .NET) na pasta “Controllers” com o seguinte conteúdo:

```
using System.Web.Mvc;

namespace
Portugal_a_Programar.Web.Controllers
{

    public class HomeController : Controller
    {

        //

        // GET: /Home/
        public ActionResult Index()
```

TEMA DE CAPA

Introdução ao ASP .NET MVC 3.0

```
{  
    return View();  
}  
}
```

A partir deste momento, ao executar esta aplicação, qualquer requisição feita para a home-page do site (ex: <http://portugal-a-programar.org/> ou <http://localhost:1234/>) fará com que um novo objecto da classe HomeController seja automaticamente instanciado pelo framework ASP .NET MVC, e o método Index será executado. A resposta deste método Index será então enviada de volta para o cliente (navegador) que fez a requisição.

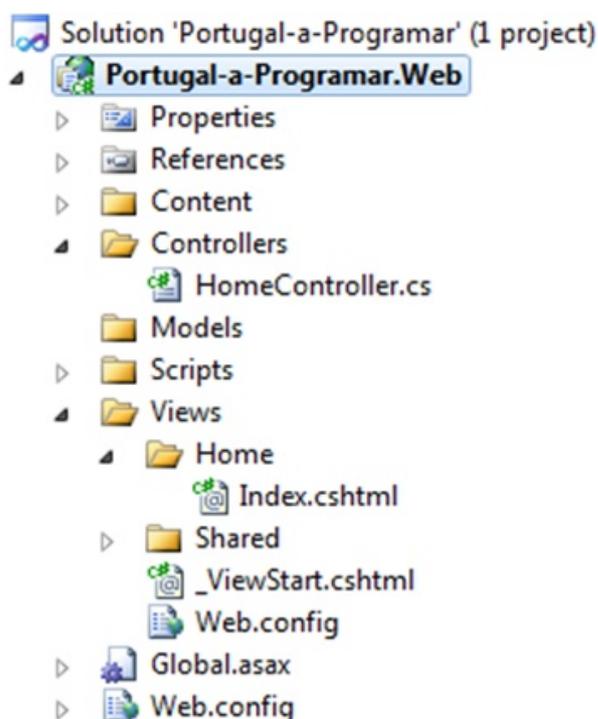
O método Index neste exemplo possui apenas uma chamada para um outro método da própria classe chamado View, um método herdado da classe Controller. O que este método faz é executar solicitar ao View Engine que seja construída uma determinada View, e como não foram passados parâmetros, este método assume que a View possui o mesmo nome do método (Index), e que está dentro de uma pasta com o mesmo nome do Controller que está a ser executado (Home).

O próximo passo é criar esta View chamada "Index", na estrutura de pastas do projecto. Para isso, vá ao Solution Explorer e dentro da pasta Views crie uma nova pasta chamada "Home" (botão direito na pasta "Views" => "Add" => "New Folder"), e então clique com o botão direito nesta nova pasta e seleccione "Add" e então "View...". No nome da View defina "Index" e seleccione "Add".

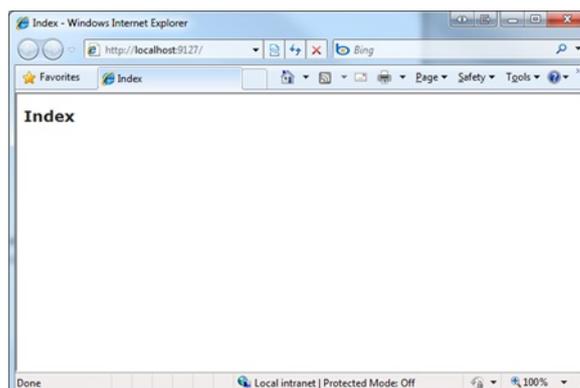
Será criado um ficheiro com o nome "Index.cshtml" (ou .vbhtml se utiliza Visual Basic .NET) na pasta "Views\Home" com o seguinte conteúdo:

```
@{  
    ViewBag.Title = "Index";  
}  
  
<h2>Index</h2>
```

Neste momento, o projecto possui uma estrutura similar a figura abaixo:



E já consegue executar a aplicação e visualizar a página inicial com a palavra "Index". Para executar a aplicação, basta seleccionar o botão "Play" no Visual Studio, ou utilizar as teclas de atalho (F5 ou CTRL + F5).



Conclusão

Como foi possível ver ao longo do artigo, que abordou de uma forma muito superficial o funcionamento e estrutura deste novo modelo de programação Web, o ASP .NET MVC 3 torna-se bastante simples de utilizar a partir do momento em que se percebe as convenções utilizadas para definir as diferentes partes de uma aplicação, e ao mesmo tempo permite-nos ter uma maior separação entre a lógica de negócio e a interface com o utilizador.

“ O ASP.NET MVC 3 (...) permite-nos ter uma maior separação entre a lógica de negócio e a interface com o utilizador ”



A comunidade NetPonto é uma iniciativa independente e sem fins lucrativos, que tem como simples e único objectivo a partilha de conhecimento na área de arquitectura e desenvolvimento de software na plataforma .NET, na tentativa de disseminar o conhecimento diferenciado de cada um de seus membros.

Cada um de nós tem diferentes talentos, e com as dezenas de tecnologias que são lançadas todos os dias, é muito difícil (para não dizer impossível) estar por dentro de tudo, e essa é uma das principais vantagens em fazer parte de uma comunidade de pessoas ligadas à nossa área. Podemos aprender mais, e mais rápido, com as experiências de cada um.

Visite-nos em <http://netponto.org>

AUTOR



Escrito por **Caio Proiete**

Exerce as funções de arquitecto de software e analista-programador na área da banca, e ministra cursos de formação em tecnologias Microsoft .NET como freelancer para empresas e centros de formação. É formador certificado pela Microsoft (MCT), Microsoft Certified Professional Developer (MCPD) nas áreas Windows, Web e Enterprise, é Microsoft Most Valuable Professional (MVP) em ASP .NET, e líder da Comunidade NetPonto (<http://netponto.org>), onde organiza reuniões presenciais todos os meses, e apresenta sessões de assuntos relacionados com desenvolvimento de software na plataforma Microsoft .NET. É autor do blog <http://caioproiete.net> - Twitter: [@CaioProiete](https://twitter.com/CaioProiete)

A PROGRAMAR

Lua – Linguagem de Programação (Parte 7)

Flex e Byacc - Parte III

Optimização de SQL em Oracle - Índices

jQuery – A framework

Lua – Linguagem de Programação (Parte 7)

No artigo anterior foi feita a explanação sobre o uso do ambiente interactivo de programação Lua, além de das operações de acesso a arquivo.

Neste artigo como prometido será tratado os assuntos relacionados a definição do escopo de variáveis, o uso de tabelas manipuladas por funções, descrição de dados e uso de programação orientada a objectos.

ESCOPO DE VARIÁVEL

A definição de variável em linguagem Lua ocorre sempre como escopo global. Caso haja a necessidade de fazer uso de uma variável com escopo local deve-se antes do nome da variável usar a instrução local. O escopo local é determinado pelo bloco no qual a variável foi declarada. Uma variável com escopo local que seja declarada fora de qualquer bloco é visível em todo o programa.

Por exemplo, o trecho de código seguinte é iniciado com a variável X global com valor 33. Dentro da decisão com a instrução if ... then / end a variável X é definida como local possuindo o valor 5.

```
X = 33
print(X)      -- mostra valor 33
if (X > 5) then
  local X = 5
  print(X)    -- mostra valor 5
end
print(X)     -- mostra valor 33
```

Ao ser executado o trecho de código anterior será apresentado os valores 33, 5 e 33, sendo o valor 33 para a variável X com escopo global e o valor 5 para a variável X com escopo local.

TABELAS MANIPULADAS POR FUNÇÕES

Na quarta parte desta série de fora abordado o uso de tabelas com listas de registos, unidimensionais e bidimensionais.

Agora será feita uma extensão a este estudo por meio do uso das funções de inserção e remoção de elementos em tabelas: insert() e remove() da biblioteca padrão table.

O programa seguinte efectua a acção de entrada e saída de elementos na tabela VETOR.

```
-- inicio do programa TAB01

local VETOR = {}

for I = 2, 10, 2 do
  table.insert(VETOR, I)
end

for I = 1, 5 do
  print("[.." .. I .."] = " .. VETOR[I])
end

-- fim do programa TAB01
```

Em seguida escreva o código de programa em um editor de texto, gravando-o com o nome tab01.lua e execute-o com a linha de comando lua 5.1 tab01.lua. Note a apresentação da saída do programa como:

```
[1] = 2
[2] = 4
[3] = 6
[4] = 8
[5] = 10
```

Perceba que foram definidos para a tabela VETOR a entrada dos valores pares de 2 até 10 de 2 em 2 por meio da função insert() da biblioteca table.

A função insert() da biblioteca table fez uso no

A PROGRAMAR

Lua - Linguagem de Programação (Parte 7)

programa de dois parâmetros, sendo o primeiro parâmetro a definição do nome da tabela e o segundo parâmetro o valor a ser inserido dentro da tabela. Na sua operação esta função pode trabalhar com até três parâmetros, sendo o segundo parâmetro de uso opcional. Assim sendo, foram usados no programa anterior, de fato, o primeiro e o terceiro parâmetros da função `insert()`.

Quando a função `table.insert()` está em uso ela efectua a entrada do valor na posição $n + 1$, onde "n" representa o tamanho da tabela. No entanto, esta função permite escolher a posição de inserção de um elemento junto a tabela, bastando fazer uso de um parâmetro a mais indicado na segunda posição.

O programa seguinte efectua a entrada na tabela de quatro valores pares e em seguida efectua a inserção do valor numérico impar 9 na primeira posição da tabela.

```
-- inicio do programa TAB02

local VETOR = {}

for I = 2, 10, 2 do
    table.insert(VETOR, I)
end

table.insert(VETOR, 1, 9)

N = table.maxn(VETOR)

for I = 1, N do
    print("[..I.."] = "..VETOR[I]")
end

-- fim do programa TAB02
```

Em seguida escreva o código de programa em um editor de texto, gravando-o com o nome `tab02.lua` e execute-o com a linha de comando `lua 5.1 tab02.lua`. Após a execução do programa ocorre a apresentação da saída:

```
[1] = 9
[2] = 2
[3] = 4
```

```
[4] = 6
[5] = 8
[6] = 10
```

Além da inserção do valor 9 na posição 1 da tabela `VETOR` está em uso a função `maxn()` que percorre a tabela contando quantos elementos existem. Desta forma, a instrução `N = table.maxn(VETOR)` atribui como valor para a variável `N` a quantidade de elementos na tabela `VETOR`.

Observe que após a inserção do valor 9 na primeira posição da tabela ocorreu o manejo de uma posição a mais para acomodar os elementos existentes.

Para fazer a remoção de elementos de uma tabela usa-se a função `table.remove()`.

O programa seguinte efectua a remoção de elementos de uma tabela definida de forma estática.

```
-- inicio do programa TAB03

local VETOR = {0, 2, 4, 6, 8}

table.remove(VETOR)

N = table.maxn(VETOR)

for I = 1, N do
    print("[..I.."] = "..VETOR[I]")
end

-- fim do programa TAB03
```

Em seguida escreva o código de programa em um editor de texto, gravando-o com o nome `tab03.lua` e execute-o com a linha de comando `lua 5.1 tab03.lua`. Após a execução do programa ocorre a apresentação da saída:

```
[1] = 0
[2] = 2
[3] = 4
[4] = 6
```

A PROGRAMAR

Lua - Linguagem de Programação (Parte 7)

Note que a função `remove()` com a definição do parâmetro `VETOR` efectuou a remoção do último elemento da tabela. No entanto, esta função permite a remoção de elementos a partir de uma posição definida, observe o código seguinte:

```
-- inicio do programa TAB04

local VETOR = {0, 2, 4, 6, 8}

table.remove(VETOR, 3)

N = table.maxn(VETOR)

for I = 1, N do
    print("[..I.."] = "..VETOR[I]")
end

-- fim do programa TAB04
```

Em seguida escreva o código de programa em um editor de texto, gravando-o com o nome `tab04.lua` e execute-o com a linha de comando `lua 5.1 tab04.lua`. Após a execução do programa ocorre a apresentação da saída:

```
[1] = 0
[2] = 2
[3] = 6
[4] = 8
```

Perceba que o valor 4 que estava na terceira posição foi removido. Com a remoção do valor 4 ocorreu o manejo dos outros valores.

No processo de manipulação de tabelas por funções há ainda a possibilidade de fazer uso da função `sort()` da biblioteca `table` para colocar os elementos em ordem ascendente se forem strings ou crescente se forem números.

O programa seguinte demonstra o uso da função `table.sort()`.

```
-- inicio do programa TAB05

local VETOR = {}
```

```
for I = 1, 5 do
    NOME = io.read()
    table.insert(VETOR, NOME)
end

for I = 1, 5 do
    print("[..I.."] = "..VETOR[I]")
end

table.sort(VETOR)
print()

for I = 1, 5 do
    print("[..I.."] = "..VETOR[I]")
end

-- fim do programa TAB05
```

Em seguida escreva o código de programa em um editor de texto, gravando-o com o nome `tab05.lua` e execute-o com a linha de comando `lua 5.1 tab05.lua`. Após a execução do programa forneça a entrada de cinco nomes. Por exemplo, se forem fornecidos os nomes: Joca, Augusto, Silvio, Paula e Sílvia, ocorrerá como saída:

```
[1] = Joca
[2] = Augusto
[3] = Silvio
[4] = Paula
[5] = Sílvia

[1] = Augusto
[2] = Joca
[3] = Paula
[4] = Sílvia
[5] = Silvio
```

Observe que na primeira parte da saída ocorrerá a apresentação dos nomes na ordem de entrada e após a execução da função `sort()` ocorre a apresentação dos nomes em ordem ascendente como indicado na segunda parte da saída.

A PROGRAMAR

Lua - Linguagem de Programação (Parte 7)

DESCRIÇÃO DE DADOS

Uma das características operacionais de Lua desde seu desenvolvimento é a capacidade de ser usada como uma linguagem de descrição de dados. No sentido de fazer uso desta característica crie um arquivo lua com o seguinte conteúdo:

```
peessoa
{
  nome = "Augusto",
  profissao = "Professor",
  idade = 45
}

peessoa
{
  nome = "Antonio",
  profissao = "Medico",
  idade = 55
}

peessoa
{
  nome = "Mariana",
  profissao = "Engenheiro",
  idade = 30
}

peessoa
{
  nome = "Marcelo",
  profissao = "Medico",
  idade = 45
}
```

Grave o arquivo com o nome popula.lua e observe que cada um dos registos está identificado com o nome pessoa. Esta é uma segunda maneira de se definir registos na linguagem Lua.

Na sequência será definido um programa que fará uso da função pessoa() que fará a apresentação dos nomes armazenados no arquivo de dados popula.lua.

```
-- inicio do programa TAB06

function pessoa(DADOS)
  print(DADOS.nome)
end

dofile("popula.lua")

-- fim do programa TAB06
```

Em seguida escreva o código de programa em um editor de texto, gravando-o com o nome tab06.lua e execute-o com a linha de comando lua 5.1 tab06.lua. Note que ao ser executado o programa os nomes existentes na descrição de dados do arquivo popula.lua são apresentados.

Observe que a função pessoa() efectua o recebimento de um parâmetro denominado DADOS que conterà os registos de todos os dados do arquivo popula.lua denominados pessoa e os insere na variável que se encontra definida como seu parâmetro, neste caso DADOS. O parâmetro DADOS está sendo tratado como se fosse uma tabela. Os nomes do registos existentes são apresentados por meio da função print().

Atente para o fato da função em uso possuir o mesmo nome de identificação de registo do arquivo popula.lua.

Para auxilio na obtenção dos dados do arquivo popula.lua faz-se uso da função dofile() que abre o arquivo indicado e faz a execução do seu conteúdo como se este conteúdo fosse um trecho de código escrito na linguagem Lua.

Outra possibilidade de uso do recurso de descrição de dados é a definição de uma tabela a partir dos dados existentes no arquivo popula.lua. A partir da definição da tabela é possível efectuar operações de controlo dos dados. Observe o código de programa seguinte:

```
-- inicio do programa TAB07

local DADOS = {}
local IND
local VLR
```

A PROGRAMAR

Lua – Linguagem de Programação (Parte 7)

```
function pessoa(CONTEUDO)
    table.insert(DADOS, CONTEUDO)
end

dofile("popula.lua")

for IND, VLR in pairs(DADOS) do
    if (VLR.idade >= 45) then
        print(VLR.nome)
    end
end

-- fim do programa TAB07
```

Em seguida escreva o código de programa em um editor de texto, gravando-o com o nome `tab07.lua` e execute-o com a linha de comando `lua 5.1 tab07.lua`. Ao ser executado o programa anterior serão apresentados os nomes Augusto, Antonio e Marcelo que são os nomes dos registos cujas idades são maiores ou iguais a 45 anos.

Para a execução do programa perceba o trecho de código da função `pessoa()` que por meio do parâmetro `CONTEUDO` faz a inserção deste dentro da tabela `DADOS`.

Observe que as variáveis de uso do programa estão definidas com escopo de visibilidade local.

Além deste efeito operacional o programa ainda faz uso de um laço `for` tipo genérico. Um laço genérico é usado para percorrer os valores de uma tabela, seus índices ou as linhas de um arquivo. Neste caso, o laço genérico está percorrendo as posições da tabela `DADOS` por meio da função iteradora `pairs()`. Caso queira percorrer os índices de uma tabela usa-se a função `ipairs()` e para as linhas de um arquivo usa-se a função `io.lines()`.

A função `pairs()` após percorrer os valores de uma tabela retorna três valores, sendo o valor da posição do próximo elemento na tabela, retorna a tabela e retorna o valor `nil` quando ultrapassar o fim da tabela. As variáveis `IND` e `VLR` do laço `for` genérico guardam respectivamente os valores do índice e do conteúdo da tabela `DADOS`.

META-TABELAS

Meta-tabela é um recurso que permite modificar o comportamento de uma tabela no sentido de definir qual ação pode ser tomada quando da realização de uma operação sobre esta tabela. Cada tabela em uso pode ter associada uma meta-tabela.

Para se definir uma meta-tabela é preciso utilizar a função `setmetatable()` que faz uso de dois parâmetros, sendo o primeiro parâmetro o nome da tabela e o segundo parâmetro o nome da meta-tabela. Assim sendo, observe a seguinte definição de meta-tabela:

```
local MT = {}
local TN = {}
setmetatable(TN, MT)
```

Onde `MT` é a meta-tabela e `TN` é a tabela normal. Isto posto, significa dizer que `MT` é uma meta-tabela de `TN`.

O próximo programa tentará efetuar a soma de dois valores associados cada um a uma tabela para criação de uma terceira tabela. Mas não conseguirá efetivar esta ação e no momento do processamento da adição ocorrerá um erro que interromperá a execução do programa. Observe o código seguinte:

```
-- inicio do programa TAB08

TABELA1 = {}
TABELA2 = {}
TABELA3 = {}

TABELA1.a = 2
TABELA1.b = 4
TABELA1.c = 6

TABELA2.a = 3
TABELA2.b = 5
TABELA2.c = 7

TABELA3 = TABELA1 + TABELA2

print(TABELA3.a)
```

A PROGRAMAR

Lua – Linguagem de Programação (Parte 7)

```
print(TABELA3.b)
print(TABELA3.c)

-- fim do programa TAB08
```

Em seguida escreva o código de programa em um editor de texto, gravando-o com o nome `tab08.lua` e tente executar o programa com a linha de comando `lua 5.1 tab08.lua`.

Note que assim que o programa for executado será apresentada a mensagem de erro `attempt to perform arithmetic on global 'TABELA1' (a table value)`.

O erro ocorre devido ao fato das tabelas `TABELA1` e `TABELA2` não possuírem meta-tabelas associadas. A linguagem Lua tenta efectuar a adição de forma tradicional, o que leva a uma falha na execução do programa.

Para sanar o problema é necessário definir para as tabelas normais uma operação de adição exclusiva para essas tabelas que serão associadas as suas meta-tabelas.

A operação de soma será efectuada pelo uso do campo `__add` (antes de `add` usa-se duas vezes o símbolo `under-line`). Assim sendo, observe o próximo código de programa:

```
-- inicio do programa TAB09

TABELA1 = {}
TABELA2 = {}
TABELA3 = {}
METATAB = {}

setmetatable(TABELA1, METATAB)

METATAB.__add = function(V1, V2)
    TABAUX = {}
    setmetatable(TABAUX, getmetatable(V1))
    TABAUX.a = V1.a + V2.a
    TABAUX.b = V1.b + V2.b
    TABAUX.c = V1.c + V2.c
    return TABAUX
end
```

```
TABELA1.a = 2
TABELA1.b = 4
TABELA1.c = 6

TABELA2.a = 3
TABELA2.b = 5
TABELA2.c = 7

TABELA3 = TABELA1 + TABELA2

print(TABELA3.a)
print(TABELA3.b)
print(TABELA3.c)

-- fim do programa TAB09
```

Em seguida escreva o código de programa em um editor de texto, gravando-o com o nome `tab09.lua` e execute-o com a linha de comando `lua 5.1 tab09.lua`. Ao executar o programa serão apresentados os resultados das adições dos valores das tabelas `TABELA1` e `TABELA2`, sendo 5, 9 e 13.

Além do uso do campo `__add` é possível efectuar operações com os campos:

- `__sub` (subtração);
- `__mul` (multiplicação);
- `__div` (divisão);
- `__mod` (resto de divisão de inteiros);
- `__pow` (exponenciação);
- `__concat` (concatenação);
- `__len` (tamanho);
- `__eq` (igualdade);
- `__lt` (menor que);
- `__le` (menor igual), entre outras.

No programa `tab09.lua` para execução da adição a linguagem Lua verifica a existência da meta-tabela e procura pelo campo `__add` que encontra-se definido para a meta-tabela `METATAB`. Encontrando executa acção de adição desejada.

ORIENTAÇÃO A OBJECTO

A linguagem Lua não é orientada a objectos, mas isto não impede que se faça uso deste paradigma de programação.

Para operar com classes, atributos, métodos e heranças, observe o que se apresenta em seguida..

Para fazer uso do paradigma da orientação a objectos em Lua é preciso criar protótipos de classes a partir de meta-tabelas dos objetos desta classe com o uso do campo `__index`.

Primeiramente faz-se a criação da estrutura básica de registo que será a base da classe. Assim sendo, considerando-se a criação de uma classe denominada TALUNO com os atributos NOME, NOTAS e MEDIA, faça:

```
TALUNO = {  
  NOME,  
  NOTAS = {},  
  MEDIA  
}
```

Observe que a definição da estrutura de uma classe é realizada sobre a mesma definição que se faz para a definição de um registo. A linguagem Lua não é orientada a objectos, pois se fosse permitiria a definição dentro da estrutura TALUNO da indicação de chamada de um método. Assim sendo, este requinte não pode ser usado em Lua, mas também não impede o uso da orientação a objectos.

Normalmente as linguagens de programação que operam com o paradigma de orientação a objectos conseguem definir uma variável a partir da estrutura de uma classe. A esta acção se dá o nome de instância e a variável usada é chamada de objecto.

Algumas das linguagens de programação com orientação a objectos fazem a instância de um objecto a partir do uso de um método interno chamado `new`, como ocorre em Java (`TALUNO ALUNO = new TALUNO();`). Será então necessário criar na linguagem Lua um método semelhante.

Após definir a estrutura da classe passa-se para a

etapa de desenvolvimento do método que fará a criação do objecto em memória a partir da classe TALUNO.

Note o código do método `new` seguinte:

```
function TALUNO.new(OBJECTO, ATRIB)  
  ATRIB = ATRIB or {}  
  setmetatable(ATRIB, OBJECTO)  
  OBJECTO.__index = OBJECTO  
  return ATRIB  
end
```

A função (método) `ALUNO.new()` faz uso de dois parâmetros, sendo o primeiro parâmetro a representação do objecto em si de nome `OBJECTO`, o segundo parâmetro representa os atributos da classe TALUNO. Esta função será usada para instanciar um objecto.

A linha `ATRIB = ATRIB or {}` faz com que ocorra a criação de um objecto vazio caso não seja passado nenhuma informação para este parâmetro.

A linha `setmetatable(ATRIB, OBJECTO)` estabelece `ATRIB` como sendo meta-tabela de `OBJECTO`.

Na sequência faz-se a atribuição `OBJECT` a `OBJECTO.__index` de modo que se houver um acesso a um atributo ausente do objecto a linguagem Lua fará sua busca em `ATRIB`.

A linha `return ATRIB` devolve para o método `ALUNO.new()` o atributo do objecto criado.

Após definir a classe e estabelecer o método que fará a criação do objecto, será definida a instância do objecto. Observe o seguinte:

```
ALUNO = TALUNO.new(TALUNO)
```

A partir deste ponto é possível escrever um programa orientado a objecto em linguagem Lua.

Na sequência será necessário preparar um método para o cálculo da média do aluno. O resultado da média será atribuído ao atributo `MEDIA` do objecto `ALUNO`. Desta forma, estará sendo simulado o uso de métodos do tipo off-line. Considere o código seguinte:

A PROGRAMAR

Lua - Linguagem de Programação (Parte 7)

```
function ALUNO:CMEDIA()
    MEDIA = 0
    for I = 1, 4 do
        MEDIA = MEDIA + ALUNO.NOTAS[I]
    end
    RESULTADO = MEDIA / 4
    return RESULTADO
end
```

A partir desta explanação considere o código de programa seguinte com a manipulação de uma classe, de atributos e de métodos.

```
-- inicio do programa TAB10

TALUNO = {
    NOME,
    NOTAS = {},
    MEDIA
}

function TALUNO.new(OBJECTO, ATRIB)
    ATRIB = ATRIB or {}
    setmetatable(ATRIB, OBJECTO)
    OBJECTO.__index = OBJECTO
    return ATRIB
end

ALUNO = TALUNO.new(TALUNO)

function ALUNO.CMEDIA()
    MEDIA = 0
    for I = 1, 4 do
        MEDIA = MEDIA + ALUNO.NOTAS[I]
    end
    RESULTADO = MEDIA / 4
    return RESULTADO
end

io.write("Nome .....: ")
ALUNO.NOME = io.read()
for I = 1, 4 do
    io.write(I.."a. nota .: ")
    NOTA = io.read("*number")
```

```
        ALUNO.NOTAS[I] = NOTA
    end
    ALUNO.MEDIA = ALUNO.CMEDIA()

    print()
    print("Nome ...: "..ALUNO.NOME)
    for I = 1, 4 do
        NOTA = ALUNO.NOTAS[I]
        print("Nota "..I.." .: "..NOTA)
    end
    print("Media ...: "..ALUNO.MEDIA)

-- fim do programa TAB10
```

Em seguida escreva o código de programa em um editor de texto, gravando-o com o nome tab10.lua e execute-o com a linha de comando lua 5.1 tab10.lua. Após exemplificar o uso de classes, atributos e métodos é oportuno exemplificar o uso de herança. Para demonstrar este aspecto será definida uma classe TSALA com atributo SALA que será herdado pela classe TALUNO. Observe o trecho:

```
TSALA = {
    SALA
}

TALUNO = {TSALA,
    NOME,
    NOTAS = {},
    MEDIA
}
```

Note que na classe TALUNO há a definição de uso da classe TSALA. O uso de TSALA como atributo de TALUNO é uma referência directa de herança sob a classe TSALA e também ao seu atributo SALA. Desta forma, está se simulando em Lua a definição de herança entre classe. Esta referência seria em C++ (class TALUNO: public TSALA).

O código seguinte demonstra na linguagem Lua o uso de herança.

A PROGRAMAR

Lua - Linguagem de Programação (Parte 7)

```
-- inicio do programa TAB11

TSALA = {
  SALA
}

TALUNO = {TSALA,
  NOME,
  NOTAS = {},
  MEDIA
}

function TALUNO.new(OBJECTO, ATRIB)
  ATRIB = ATRIB or {}
  setmetatable(ATRIB, OBJECTO)
  OBJECTO.__index = OBJECTO
  return ATRIB
end

ALUNO = TALUNO.new(TALUNO)

function ALUNO.CMEDIA()
  MEDIA = 0
  for I = 1, 4 do
    MEDIA = MEDIA + ALUNO.NOTAS[I]
  end
  RESULTADO = MEDIA / 4
  return RESULTADO
end

io.write("Nome .....: ")
ALUNO.NOME = io.read()
```

```
io.write("Sala .....: ")
ALUNO.SALA = io.read()
for I = 1, 4 do
  io.write(I.."a. nota ..: ")
  NOTA = io.read("*number")
  ALUNO.NOTAS[I] = NOTA
end
ALUNO.MEDIA = ALUNO.CMEDIA()

print()
print("Nome ....: "..ALUNO.NOME)
print("Sala ...: "..ALUNO.SALA)
for I = 1, 4 do
  NOTA = ALUNO.NOTAS[I]
  print("Nota "..I.." ..: "..NOTA)
end
print("Media ..: "..ALUNO.MEDIA)

-- fim do programa TAB11
```

Em seguida escreva o código de programa em um editor de texto, gravando-o com o nome `tab11.lua` e execute-o com a linha de comando `lua 5.1 tab11.lua`.

CONCLUSÃO

Neste artigo foi dada atenção as acções de processamento baseadas no uso de escopos de variáveis, funções de tabelas, descrição de dados e orientação a objectos.

No próximo artigo será tratado o assunto de uso de arquivos com a linguagem Lua..

AUTOR



Escrito por **Augusto Manzano**

Natural da Cidade de São Paulo, tem experiência em ensino e desenvolvimento de programação de software desde 1986. É professor da rede federal de ensino no Brasil, no Instituto Federal de Educação, Ciência e Tecnologia. É também autor, possuindo na sua carreira várias obras publicadas na área da computação.

A PROGRAMAR

Flex e Byacc - Parte III

Ao longo desta série de artigos, temos vindo a ver como utilizar as ferramentas flex e byacc para escrever uma calculadora simples. No primeiro artigo vimos como escrever o scanner da nossa calculadora usando flex e no segundo como escrever o parser usando byacc. Neste artigo veremos o que nos falta para conseguir criar o executável da nossa calculadora, e como lhe adicionar funcionalidade eficientemente.

O ficheiro do parser que vimos no artigo anterior estaria completo, mas como o código do scanner é C e o do parser é C++, precisamos de dar uma ajuda ao byacc.

```
#include <cstdlib>

extern "C"
{
    int yyparse(void);
    int yylex(void);
    int yywrap()
    {
        return 1;
    }
    void yyerror(char *);
}
```

Declaramos as funções do scanner que o parser usa como funções C no bloco dos #includes do parser e resolvemos o problema. Para concluir o exemplo vejamos como expandir a nossa calculadora, adicionando-lhe variáveis por exemplo. Introduzir variáveis implica alterar o parser e o scanner simultaneamente. Começando pelo parser, vejamos do que precisamos para ter variáveis. Sendo uma calculadora simples as nossas variáveis vão conter sempre doubles, e são identificadas por uma string, portanto podemos usar um map para as guardar.

Também vamos necessitar dum novo token que nos indique uma variável, e de outro token que sirva como operador de atribuição. Vejamos a nova primeira secção do parser:

```
%{
#include <cstdlib>
#include <cmath>
#include <iostream>
#include <map>
#include <string>

extern "C"
{
    int yyparse(void);
    int yylex(void);
    int yywrap()
    {
        return 1;
    }
    void yyerror(char *);
}

std::map<std::string, double> varMap;

%}

%union {
    double d;
    char * s;
};

%token<d> NUMBER
%token<s> VARIABLE
%token NL

%right '='
%left '+' '-'
%left '*' '/'
```

A PROGRAMAR

Flex e Byacc - Parte III

```
%right '^'  
%nonassoc UMINUS  
  
%type<d> expr
```

Temos o mapa onde guardamos as variáveis, varMap, um novo tipo na %union, char * para o novo token VARIABLE, e um operador de atribuição '=', associativo à direita e com a menor precedência de todos os operadores. A segunda secção do parser segue-se:

```
%%  
  
expr_list      : expr NL { std::cout << $1 <<  
std::endl; }  
               |expr_list expr NL { std::cout  
<< $2 << std::endl; }  
               ;  
  
expr : NUMBER          { $$ = $1; }  
     | VARIABLE  
     {  
         std::string key($1);  
         $$ = (varMap.find(key) !=  
varMap.end())? varMap[key] : 0;  
     }  
     | '-' expr %prec UMINUS { $$ =  
-$2; }  
     | expr '+' expr        { $$ =  
$1+$3; }  
     | expr '-' expr        { $$ =  
$1-$3; }  
     | expr '*' expr        { $$ =  
$1*$3; }  
     | expr '/' expr        { $$ =  
$1/$3; }  
     | expr '^' expr        { $$ =  
pow($1, $3); }  
     | VARIABLE '=' expr  
     {  
         std::string key($1);
```

```
varMap[key] = $3;  
         $$ = varMap[key];  
     }  
     | '(' expr ')'        { $$ = $2; }  
     ;
```

```
%%
```

A gramática mantém-se praticamente inalterada, só adicionámos duas novas produções, uma para indicar que uma expressão pode ser uma variável e a outra para indicar que uma expressão pode ser uma atribuição. Ambas têm acções associadas, a primeira verifica se a variável se encontra no mapa, se sim atribui à cabeça o valor da variável no mapa, se não atribui zero. A segunda atribui ao valor da variável no mapa o valor da expressão, e à cabeça o valor da variável no mapa, ou seja é possível escrever (f = 3) + 5 e obter 8 mas sendo atribuído 3 a f. A terceira secção mantém-se inalterada portanto com isto concluímos o parser.

O novo scanner necessita de ainda menos alterações que o parser, basta enviarmos os novos tokens VARIABLE e '='. Para o token VARIABLE adicionamos a seguinte expressão regular na primeira secção,

```
IDENTIFIER [A-Za-z_][A-Za-z0-9_]*
```

e a seguinte regra na segunda secção:

Para adicionar o token '=' só necessitamos de alterar

```
{IDENTIFIER} { yylval.s = strdup(yytext);  
return VARIABLE; }
```

a regra,

```
[-+*/^()] return *yytext;
```

para:

```
[-+*/^()] return *yytext;
```

A PROGRAMAR

Flex e Byacc - Parte III

Os ficheiros finais do scanner e do parser encontram-se abaixo.

```
%option outfile="CalculatorScanner.c"

%{
#include "CalculatorParser.tab.h"

void yyerror(char * err_msg)
{
    printf("%s\n", err_msg);
    exit(1);
}

%}

IDENTIFIER  [A-Za-z_][A-Za-z0-9_]*
WS          [ \t]
NL          "\n"|" \r"|" \r\n"
DIG         [0-9]

MANTISSA    ({DIG}*\.{DIG}+)|({DIG}+\.)
EXPONENT    [eE][+-]?{DIG}+

INT         (0|[1-9]{DIG}*)
DOUBLE      {MANTISSA}{EXPONENT}?

%%

{IDENTIFIER}    { yylval.s = strdup(yytext);
return VARIABLE; }
{INT}|{DOUBLE}  { yylval.d = strtod(yytext,
NULL); return NUMBER; }

[--+*/=^()]     return *yytext;

{NL}            return NL;
{WS}            ; /* ignore whitespace */

.               yyerror("Unknown character");

%%
```

```
{
#include <cstdlib>
#include <cmath>
#include <iostream>
#include <map>
#include <string>

extern "C"
{
    int yyparse(void);
    int yylex(void);
    int yywrap()
    {
        return 1;
    }
    void yyerror(char *);
}

std::map<std::string, double> varMap;

%}

%union {
    double d;
    char * s;
};

%token<d> NUMBER
%token<s> VARIABLE
%token NL

%right '='
%left '+' '-'
%left '*' '/'
%right '^'
%nonassoc UMINUS
%type<d> expr

%%

expr_list    : expr NL           { std::cout
<< $1 << std::endl; }
            | expr_list expr NL { std::cout
<< $2 << std::endl; }
            ;
```

```
expr      : NUMBER          { $$ = $1; }
          | VARIABLE
          {
            std::string key($1);
            $$ = (varMap.find(key) !=
varMap.end())? varMap[key] : 0;
          }
          | '-' expr %prec UMINUS  { $$ =
-$2; }
          | expr '+' expr          { $$ =
$1+$3; }
          | expr '-' expr          { $$ =
$1-$3; }
          | expr '*' expr          { $$ =
$1*$3; }
          | expr '/' expr          { $$ =
$1/$3; }
          | expr '^' expr          { $$ =
pow($1, $3); }
          | VARIABLE '=' expr
          {
            std::string key($1);
            varMap[key] = $3;
            $$ = varMap[key];
          }
          | '(' expr ')'           { $$ = $2; }
          ;

%%

int main()
{
    yyparse();
}
```

Tendo dado ao scanner o nome de `CalculatorScanner.l` e ao parser o nome de `CalculatorParser.y`, basta correr os seguintes comandos na shell para criar o executável `calculator`.

```
flex CalculatorScanner.l byacc -dtv -b
CalculatorParser CalculatorParser.y
gcc -c CalculatorScanner.c
g++ -c CalculatorParser.tab.c
g++ -o calculator CalculatorParser.tab.o
CalculatorScanner.o
```

Esta série de artigos pretende-se apenas como uma introdução a estas ferramentas e como tal não cobre muitos dos aspectos do flex e do byacc. Compreender o uso de estados num analisador lexical é fundamental em aplicações mais sofisticadas, e a introdução às gramáticas aqui feita é ténue e não aborda os conceitos com exactidão, tendendo a focar-se em explicações mais intuitivas. Para usar estas ferramentas com proficiência, é necessário mais estudo, para o qual pode consultar:

<http://dinosaur.compilertools.net/>

E ler os seguintes livros:

- *lex & yacc*, 2nd edition, John R. Levine, Tony Mason, Doug Brown, 1995, ISBN 10: 1-56592-000-7, ISBN 13: 9781565920002, O'Reilly & Associates
- *Compilers: Principles, Techniques, & Tools* (2nd Edition) ("livro do dragão"), Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, 2006, ISBN-10: 0321486811, ISBN-13: 978-0321486813, Addison-Wesley Professional

AUTOR



Escrito por **João Mares**

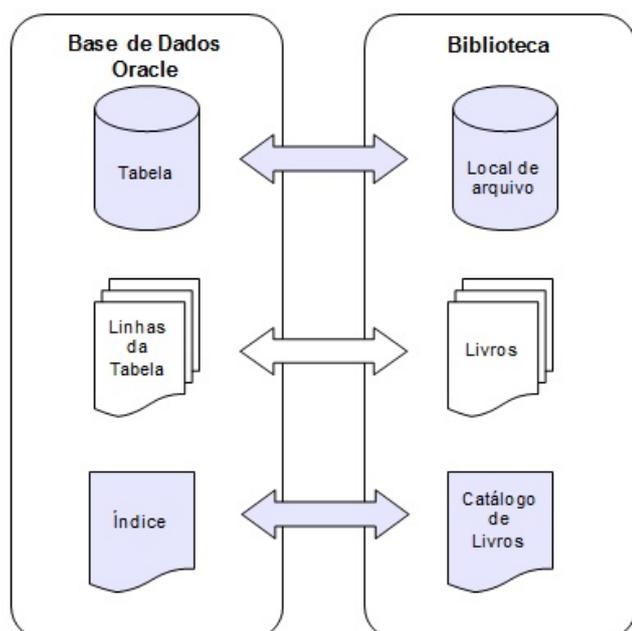
Estando actualmente no terceiro ano de LEIC no IST, interessa-se preferencialmente por arquitectura de computadores e computação gráfica.

Optimização de SQL em Oracle - Índices

Na última edição, iniciámos a temática da optimização de SQL recorrendo à utilização de bind variables. Verificámos que, em casos específicos, o aumento de desempenho do Oracle é significativo e fulcral para o sucesso de uma aplicação.

A optimização de SQL vai muito mais além das bind variables e da correcta gestão da shared pool. Nesta edição, vamos explorar os índices e perceber como a (in)correcta utilização dos mesmos pode afectar o desempenho das queries. A completa compreensão dos efeitos de um índice implica um conhecimento mais profundo das técnicas de diagnóstico de queries do Oracle. No entanto, essa não é a intenção deste artigo, pretendendo apenas mostrar o que é e como funciona um índice.

O índice é um dos objectos que está presente na generalidade das bases de dados. No Oracle, em particular, o índice é um objecto que está associado a uma ou várias colunas da tabela e é comum encontrar-se relacionado com qualquer tabela. Para se compreender a ideia do índice, vamos fazer a analogia de uma base de dados com uma biblioteca.



Podemos considerar que uma base de dados é uma biblioteca. Dependendo da normalização da base de dados, esta poderá ter n tabelas. Consideremos que a nossa base de dados tem apenas uma tabela e que a mesma se encontra na primeira forma normal. Cada entrada da tabela é um livro de um determinado autor, com um título, tema, editora, etc., e com dezenas de páginas de texto. Portanto, com vários campos.

Na biblioteca, se pretendemos procurar um livro, temos duas opções: ou percorremos toda a biblioteca ou consultamos o catálogo.

“ Na biblioteca, se pretendemos procurar um livro, temos duas opções: ou percorremos toda a biblioteca ou consultamos o catálogo. ”

Em Oracle, estas pesquisas traduzem-se num *full table scan* ou num *index scan*, respectivamente. Podemos deduzir o modo como o optimizador do Oracle efectua as pesquisas, se visitarmos a biblioteca. Vejamos como procederíamos para encontrar as seguintes entradas:
(ver tabela1)

É fácil identificarmos as situações em que o optimizador recorre ao *index scan* ou ao *full table scan*, nos casos extremos em que sabemos que o resultado da query será muito ou pouco selectivo, respectivamente. O nosso senso comum leva-nos a escolher a melhor opção e o mesmo deverá acontecer com o “senso comum” do optimizador. Mas, o que acontece nos restantes casos em que não é lógico e directo escolher uma opção, tal como no caso dos livros de 1980? Nestas situações, o

A PROGRAMAR

Optimização de SQL em Oracle - Índices

Entrada	Suposição	Pesquisa na Biblioteca	Pesquisa no Oracle
Livro "Os Maias" de Eça de Queirós	Livro específico com título e autor. Deverão existir poucos exemplares na biblioteca, logo, é uma pesquisa muito selectiva .	Pesquisa no índice do catálogo (com determinada ordenação) para procurar o livro.	<i>Index Scan</i>
Livros de língua portuguesa	Deverão ser a maior parte dos livros da biblioteca, logo, é uma pesquisa pouco selectiva .	Pesquisa directa pelos corredores da biblioteca.	<i>Full table scan</i>
Livros de 1980	Poderá existir uma vasta colecção de livros deste ano. Ou não?	Pesquisa no catálogo. Tomada de decisão, consoante o número de entradas no catálogo.	?

Oracle recorre às estatísticas.

Entre as várias informações que as estatísticas recolhem, destacamos aqui a informação da base de dados relativamente à dimensão das tabelas, dos índices e da sua selectividade (número de chaves diferentes), bem como da existência de histogramas.

A recolha automática das estatísticas poderá já estar activada através de um job mas, em certas circunstâncias e, em bases de dados de grande dimensão, é normal encontrar-se inactiva. Este é um procedimento muito pesado para a base de dados e que, se efectuado em horário de produção, poderá afectar significativamente o desempenho das aplicações. Daí que este procedimento só deva ser agendado para períodos mortos da base de dados.

Um método manual de recolha das estatísticas para um determinado objecto é executar o seguinte procedimento de PL/SQL que aplicaremos sobre a tabela **t_livros** do utilizador **usr_biblioteca**:

```
begin
  dbms_stats.gather_table_stats(
    ownname=>usr_biblioteca,
    tabname=>t_livros,
    estimate_percent=>30,
    method_opt=>'for all indexed
columns size auto',
    cascade=>true
  );
end;
```

Tabela1

Note-se o parâmetro *estimate_percent* que vai definir a percentagem de dados da base de dados a serem analisados para a obtenção da estimativa.

Após o cálculo das estatísticas, o optimizador conhece o número de chaves de um índice mas não a forma como essas chaves estão distribuídas.

Criemos um índice sobre a coluna do ano de edição do livro (ex.: data_edicao):

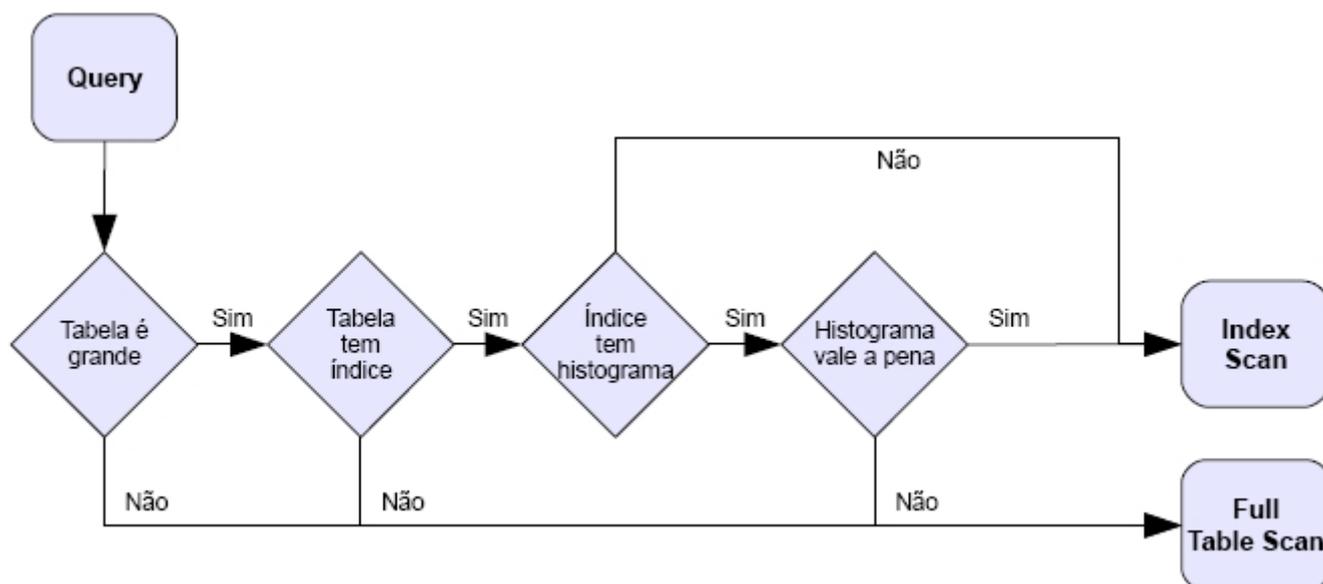
```
CREATE INDEX T_LIVROS$DATA_EDICAO ON
T_LIVROS(DATA_EDICAO);
```

O optimizador sabe que existem livros de 1970, 1971, ..., 1980, etc.. Os histogramas de frequência indicam quantos títulos existem referentes a cada ano. Só assim, o optimizador pode tomar a decisão correcta para encontrar todos os livros de 1980: ou recorrendo ao *index scan*, se forem poucos exemplares, comparativamente à dimensão da tabela; ou recorrendo ao *full table scan*, se existirem muitos exemplares.

A temática das estatísticas e histogramas requer uma análise mais detalhada mas que foge do âmbito deste artigo.

A PROGRAMAR

Optimização de SQL em Oracle - Índices



Podemos ver na figura a síntese do procedimento adoptado pelo optimizador ao receber uma query.

O que indexar?

Já conhecemos as vantagens de um índice e, à partida, apetece indexar tudo na base de dados. Mas, a indexação total poderá não ser uma boa opção.

O índice é um objecto que ocupa o seu espaço na base de dados. Um índice de uma coluna tem tantos valores quanto essa coluna. Se a coluna contiver mil registos, então o índice terá mil entradas, também. No entanto, o índice tem uma característica que o diferencia da tabela: é ordenado. Logo, uma pesquisa indexada será muito mais rápida do que uma pesquisa total à tabela. Mas, nem sempre é assim.

Na pesquisa:

```
SELECT TITULO
FROM T_LIVROS
WHERE TITULO LIKE 'OS MAIA%'
```

O índice salta directamente para a letra O, pois é a primeira letra do título. No entanto, nesta pesquisa:

```
SELECT TITULO
FROM T_LIVROS
WHERE TITULO LIKE '%MAIA%'
```

“(…) o índice tem uma característica que o diferencia da tabela: é ordenado (…)”

não sabemos qual a primeira letra do título. Todas as chaves deverão ser percorridas o que, em situações em que o tamanho do índice é semelhante à dimensão da tabela, não traz qualquer vantagem. Portanto, há também que ter em conta o modo como se efectuam as pesquisas.

Voltemos ao exemplo da biblioteca. Imaginemos que um dos campos do livro é o código ISBN. Como sabemos, o ISBN é um código único. Logo, o número de chaves do índice associado a este campo será igual ao total de obras, excluindo os exemplares duplicados. Uma pesquisa a um índice ordenado será muito mais rápida do que uma pesquisa total a uma tabela desordenada. Portanto, faz sentido indexar este campo.

Consideremos agora o campo idioma. Numa biblioteca portuguesa, de certo que a grande maioria dos livros serão de língua portuguesa. Valerá a pena construir um índice com milhares de entradas todas

iguais? Na prática, valerá a pena um índice bibliotecário ordenado simplesmente pelo idioma e que há-de conter quase todas as obras da biblioteca? A resposta é não.

Uma boa escolha do que se vai indexar é uma questão de bom senso e implica um conhecimento à priori do conteúdo da tabela. Podemos criar uma tabela sem índices e apenas os implementarmos à medida que a solução vai crescendo, à medida que temos dados que nos permitam identificar o que queremos pesquisar e quão organizada e selectiva está essa informação. Indexar uma coluna em que a maioria das entradas são idênticas é um desperdício de espaço e não traz qualquer benefício ao otimizador.

Indexação concatenada

Além da simples indexação de uma coluna, podemos também concatenar dois campos da tabela num só índice. Por exemplo, criando um índice para o conjunto de título e autor:

```
CREATE INDEX T_LIVROS$TIT_AUT ON  
T_LIVROS(TITULO, AUTOR)
```

Este índice vai permitir uma pesquisa mais rápida nas queries:

```
SELECT *  
FROM T_LIVROS  
WHERE TITULO LIKE 'OS MAI%' AND AUTOR LIKE  
'EÇA%';
```

ou

```
SELECT *  
FROM T_LIVROS  
WHERE TITULO LIKE 'OS MAI%' AND AUTOR LIKE  
'%QUEIR%';
```

mas não surtirá qualquer efeito em:

```
SELECT *  
FROM T_LIVROS  
WHERE AUTOR LIKE 'EÇA%' AND TITULO LIKE 'OS  
MAI%'
```

A justificação está no facto de que o índice concatenado só é utilizado se a pesquisa for feita pela mesma ordem da criação do índice. Isto é, se o índice recai sobre as colunas (a,b,c), a pesquisa deverá incidir sobre as colunas (a), (a,b) ou (a,b,c), mantendo a ordem do índice.

Índices de funções

Na pesquisa de campos de texto, o índice de coluna poderá nunca ser utilizado. Vejamos o caso em que os títulos dos livros estão inseridos na base de dados com letras maiúsculas e minúsculas. Se efectuarmos a seguinte query:

```
SELECT TITULO FROM T_LIVROS WHERE TITULO =  
'Os Maias'
```

o índice será usado e trará o resultado certo. Mas, se não soubermos como os dados foram introduzidos na tabela, ou se tivermos um utilizador da biblioteca a efectuar consultas, podemos ter que recorrer à pesquisa:

```
SELECT TITULO FROM T_LIVROS WHERE  
LOWER(TITULO) LIKE 'oS MaiAs'
```

Neste caso, o índice não é utilizado, pois os valores guardados no índice são sensíveis a maiúsculas e minúsculas. A transformação da cadeia de caracteres para minúsculas não vai produzir resultados na pesquisa do índice. A solução passa pela criação de uma function index:

```
CREATE INDEX T_LIVROS$LOWER_TITULO ON  
T_LIVROS(LOWER(TITULO))
```

A PROGRAMAR

Optimização de SQL em Oracle - Índices

Valores NULL

O Oracle não guarda valores NULL. Podemos verificar, tentando introduzir uma linha totalmente NULL numa tabela. Como tal, também os índices não guardam os valores NULL dos campos de uma tabela. Portanto, uma pesquisa:

```
SELECT * FROM T_LIVROS WHERE TEMA IS NULL
```

vai obrigar a um *full table scan*, mesmo que exista um índice altamente selectivo para esta coluna.

Conclusão

A utilização e parametrização de índices é um tema bem mais complexo. O acompanhamento e a monitorização dos índices e das queries através de planos de execução permite conhecer a fundo o porquê das ilações acima descritas.

“ O acompanhamento e a monitorização dos índices e das queries através de planos de execução permite conhecer a fundo o porquê das ilações (...) ”

O que aqui tentámos expor foi apenas uma pequena abordagem desmistificando a verdadeira natureza deste objecto tão falado no mundo das bases de dados mas cuja utilização é, muitas vezes, desconhecida e mal aplicada.

Nota: Agradeço ao Eng.º José Aser, formador do Citeforma, a sua colaboração na ajuda da revisão dos meus artigos de Oracle, bem como no conhecimento que me transmitiu no curso de SQL Avançado.

AUTOR



Escrito por **Ricardo Trindade**

É actualmente o responsável pela área informática dos Serviços Sociais da Câmara Municipal de Lisboa onde efectua, desde administração de sistemas, até programação em .NET. Está também ligado ao projecto N-Ideias na área da consultoria, webdesign e software, trabalhando essencialmente com BD's Oracle, PostgreSQL e SQL Server.

jQuery – A framework

Neste segundo artigo vou entrar mais em detalhe nas funcionalidades do jQuery, como o motor de selectores, que é uma das funcionalidades core, as funcionalidades de traversing e de manipulation.

Uma das funcionalidades que torna o jQuery muito compacto é o chaining (encadeamento). Este consiste no facto de qualquer função executada sobre um objecto jQuery, retorna o objecto (com as devidas alterações efectuadas na função), permitindo um executar funções encadeadas, num estilo de código altamente compacto e mesmo assim extremamente perceptível.

Relativamente ao motor de selectores, ele é o que nos permite, através de uma sintaxe mista entre XPath, CSS e algumas extensões nativas ao jQuery, obter um conjunto de elementos do DOM.

Existe vários tipos de selectores, por atributo, hierárquico, filtros pelo tipo de elemento, pelo estado do elemento, etc:

Por elemento:

- Filtro universal:

`$("*")` -> selecciona todos os elementos;

- Selecção por classe:

`$(".classe")` -> selecciona todos os elementos que contenham determinada classe;

- Selecção por nome de elemento:

`$("elemento")` -> selecciona todos os elementos com este nome

- Selecção por id de elemento:

`$("#id")` -> selecciona o elemento com este id

- Selecção múltipla:

`$("selector1, selector2, selectorN")` -> selecciona a união de todos os selectores

Genérico:

- Transformar um objecto num objecto jQuery

`$(object)` -> ao dar como input ao motor de selectores um objecto, o jQuery encarrega-se de transformá-lo num objecto jQuery automaticamente

- Contextualizar selecção:

`$("algumselector", contexto)` -> deste modo podemos restringir o âmbito do nosso selector a um contexto, que é uma nossa selecção prévia.

Por atributo:

- Elemento contém atributo:

`$("[atributo]")` -> selecciona todos os elementos que tenham o atributo especificado

- Valor atributo igual a:

`$("[atributo='valor']")` -> selecciona todos os elementos cujo atributo especificado tenha exactamente o valor especificado

- Valor atributo diferente de:

`$("[atributo!='valor']")` -> selecciona todos os elementos cujo atributo especificado tenha um valor diferente do valor especificado

- Valor atributo contém:

`$("[atributo*='texto']")` -> seleccionar todos os elementos cujo atributo especificado contém o texto especificado

- Valor atributo termina com:

`$("[atributo$='texto']")` -> seleccionar todos os elementos cujo atributo especificado termina com o texto especificado

- Valor atributo começa com:

`$("[atributo^='texto']")` -> seleccionar todos os elementos cujo atributo especificado começa com o texto especificado

- Múltiplos filtros por atributo:

`$("[atributo='valor'][atributo^='texto']")` -> selecciona todos os elementos que correspondam à intersecção de todos os filtros

Por conteúdo:

- Elemento contém texto:

A PROGRAMAR

jQuery - A framework

`$(":contains('texto'))` -> selecciona todos os elementos que contenham o texto especificado (case sensitive)

- Elemento contém texto:

`$(":has(elemento))` -> selecciona todos os elementos que contenham o elemento especificado

Depois de ter dado uma visão sobre o poder dos selectores jQuery, vamos ao que interessa:

```
<html>
  <head>
    <script type="text/javascript"
src="jquery-1.4.4.min.js"></script>
  </head>
  <body>
    <a href="http://www.portugal-a-
programar.org/">PaP</a>
  </body>
</html>
```

Existem vários modos de obter uma referência à âncora segundo o que foi dito anteriormente:

1. `$("#a")` visto que é a única âncora do documento
2. `$("#a:contains('PaP')")` através do seu conteúdo
3. `$("#a[href*='portugal-a-programar']")` através do atributo href
4. `$("#a[href^='http://www.portugal']")` através do início do atributo href
5. `$(document).find("a")` deste modo estamos a encapsular o objecto document num objecto jQuery, e a utilizar a função de traversing find que possibilita pesquisar dentro de um objecto jQuery pelos seus descendentes, neste caso pelos descendentes do document, mas um melhor exemplo poderia ser:

```
<ul class="main">
  <li class="first-li"></li>
  <li class="last-li"></li>

  <li class="parent">
    <ul>
```

```
    <li class="inner"></li>
    <li class="inner"></li>
  </ul>
</li>
</ul>

$("#ul.main").find('li.inner').size()
```

Neste caso, o objecto jQuery iria conter a duas li com a class inner, e a função size invocada no final permite ao utilizar saber exactamente quantos elementos do DOM contém o objecto jQuery.

A nível de traversing existem algumas funções chave como a já falada find, entre outras:

- `add()`: com esta função podemos adicionar ao objecto jQuery mais elementos do DOM, através de selector ou objecto que referencie um elemento, ou até fornecer html que adiciona o elemento ao objecto jQuery, mas não o mostra na página, visto que temos de utilizar uma das funções de adição de elementos ao DOM de seguida, para tornar o pedaço de HTML parte do DOM;
- `children()`: esta função é similar à find com a diferença que só permite encontrar elementos no nível imediatamente abaixo ao dos elementos no objecto jQuery;
- `contents()`: função similar à anterior, mas que inclui além dos elementos, também o texto;
- `each()`: esta função permite iterar pelos items de um objecto jQuery:

```
$('.li').each(function(index) {
  alert([' + index + ' ] : ' +
$(this).text());
});
```

- `parent()`: obter o parent de cada item de um objecto jQuery apenas um nível acima;
- `parents()`: obter o parent de cada item de um objecto jQuery até ao topo da árvore DOM;

- `is()`: retorna um bool que nos diz se os items de um objecto jQuery correspondem ao selector indicado como parâmetro;

- `filter()`: esta função é também similar à `find` mas ao contrário deste, remove os items filtrados do objecto jQuery

Quanto a manipulation, existem muitas funções utilizadas frequentemente:

- `addClass()`, `hasClass()`, `removeClass()`, `toggleClass()`: estas funções, como o próprio nome indica, têm como função adicionar classes, retornar um bool caso tenha ou não determinada classe, remover classes, ou adicionar caso não tenha ou remover caso tenha determinadas classes, respectivamente;

- `append()`: com o `append`, como o próprio nome indica, adiciona-se HTML a todos os elementos de um objecto jQuery;

- `attr()`, `removeAttr()`: com a função `attr`, podemos ler atributos, escrever em atributo, ou ou vários de cada vez, e até utilizar uma função como callback, para computar o valor de um atributo. Com a `removeAttr`, podemos remover atributos;

- `clone()`: esta função cria uma cópia de todos os elementos num objecto jQuery, incluindo os seus descendentes e é a indicada para copiar elementos numa página. Opcionalmente também pode copiar os eventos, caso passemos o parâmetro com valor true;

- `css()`: através da função `css` podemos manipular o style de um elemento, sem preocupações com diferentes nomenclaturas destas propriedades entre cada browser, funciona de modo similar à função `attr`;

- `empty()`: com esta função podemos limpar o conteúdo de um ou vários elementos, tanto elementos descendentes como texto;

- `html()`: retorna o conteúdo html de um elemento utilizando o `innerHTML`;

- `text()`: retorna o texto de todos os elementos do objecto jQuery, à excepção de elementos do tipo `input`, para os quais é usada a função `val()` descrita abaixo;

- `wrap()`, `unwrap()`: a primeira envolve todos os elementos do objecto jQuery num html passado por parâmetro, e a segunda remove os parents de todos os elementos do objecto jQuery;

- `wrapAll()`, `wrapInner()`: a primeira envolve todos os elementos do objecto jQuery num único parent, passado por parâmetro, enquanto que a segunda envolve os descendentes de todos os elementos do objecto jQuery no parent passado por parâmetro;

Já tendo uma visão de como podemos obter, refinar e manipular a nossa lista de elementos num objecto jQuery, passemos a casos de uso.

Neste caso queremos gerar dinamicamente uma lista de li com base num dado JSON (JavaScript Object Notation), dentro de uma ul:

```
<html>
  <head>
    <script type="text/javascript"
src="jquery-1.4.4.min.js"></script>
  </head>
  <body>
    <ul class="main">
    </ul>
  </body>
</html>
```

```
var items = {"news":[{"title":"PaP has a new
magazine","lead":"some text to fill
this..."}, {"title":"This is the second
article","lead":"some other text to fill
this..."}]};
```

```
var $ul = $("ul.main"); /*aqui, uma das boas
práticas, sempre que criamos um objecto de
jQuery com base num dado selector, devemos
sempre guardar o resultado numa var, deste
modo estamos a colocar o resultado de
execução do selector em cache, para não se
repetir a mesma operação N vezes. Quanto ao
prefixo de $ é algo que se costuma utilizar
para tornar o código mais legível, de modo
```

A PROGRAMAR

jQuery – A framework

que sempre que uma var tem este prefixo, depreende-se que é um objecto jQuery.*/

```
/*quanto à função $.each, é uma das utilities do jQuery, que permite iterar sobre arrays e objects e como callback para cada iteração, define-se uma function que recebe 2 parâmetro, como primeiro o index da iteração, e como segundo o valor actual da iteração, neste caso o valor é o objecto que contém as propriedades title e lead, definidas no JSON acima.*/
```

```
$.each(items.news, function(index, value) {  
    $ul.append("<li>"+value.title+"</li>");  
});
```

/*gerando o html abaixo, imaginemos que queremos, colocar uma div como parent da ul, envolver o texto de cada li numa span, adicionar classes às li, clonar as li e adicionar as clonadas à mesma ul, adicionar algum texto extra com o número da linha a cada li e alguns estilos directamente no atributo style. Finalmente obter o texto de todas as li, sem HTML.*/

```
<html>  
  <head>  
    <script type="text/javascript"  
src="jquery-1.4.4.min.js"></script>  
  </head>  
  <body>  
    <ul class="main">  
      <li>PaP has a new magazine</li>  
      <li>This is the second article</li>  
    </ul>  
  </body>  
</html>
```

/*gerando o html abaixo, imaginemos que queremos, colocar uma div como parent da ul, envolver o texto de cada li numa span, adicionar classes às li, clonar as li e

adicionar as clonadas à mesma ul, adicionar algum texto extra com o número da linha a cada span da li e alguns estilos directamente no atributo style. Finalmente obter o texto de todas as li, sem HTML.*/

```
//a função end() é utilizada para voltar ao context anterior, ou seja, neste caso, antes do find()  
$ul.wrap("<div/>").children().wrapInner("<span/>").find("span").addClass("testing").end().clone().appendTo($ul);  
var text =  
$ul.find("span").append(function(index,html)  
{  
    return " : " + (index+1);  
}).css({"font-weight":"bold",  
"color":"blue"}).parent().text(); /*aqui utiliza-se o método css() mas ao invés de passar cada propriedade style em diferentes invocações à função, invoca-se apenas uma vez, usando um objecto com as várias propriedade definidas*/  
alert(text); //mostrar o texto das 4 li
```

//o html final depois de todas as transformações é:

```
<html>  
  <head>  
    <script type="text/javascript"  
src="jquery-1.4.4.min.js"></script>  
  </head>  
  <body>  
    <div>  
      <ul class="main">  
        <li><span class="testing"  
style="font-weight: bold; color: blue; ">PaP  
has a new magazine : 1</span></li>  
        <li><span class="testing"  
style="font-weight: bold; color: blue; ">This  
is the second article : 2</span></li>  
        <li><span class="testing"
```




ppp
portugal-a-programar



facebook.com/portugal.programar



twitter.com/pt_programar

COLUMNAS

PROGRAMADOR EXCÊNTRICO - 6 regras para utilizar AJAX

VISUAL (NOT) BASIC - Windows Phone 7

CORE DUMP - gcc -Wall myApp.c -linstantaneous -o success

PROGRAMADOR EXCÊNTRICO

6 regras para utilizar AJAX

AJAX, ou Asynchronous JavaScript And XML, tem ganho muita força nos últimos tempos. Os Programadores Web têm feito coisas magníficas com o AJAX, mas também já vi grandes erros e uma má aplicação do AJAX.

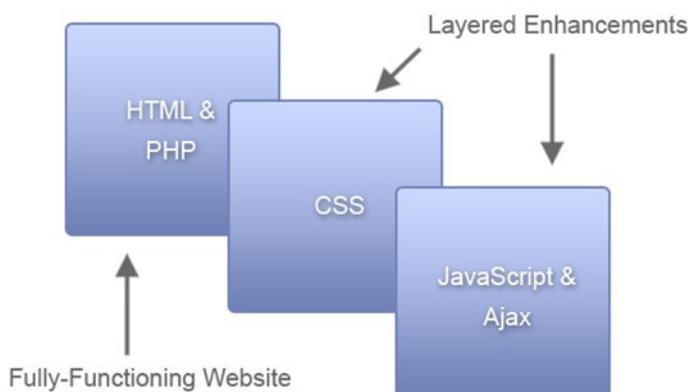
Queria partilhar convosco, através da minha experiência como Programador Web mas também como utilizador frequente da Internet, seis regras para utilizar o AJAX.

AJAX para melhorar, não para funcionar

Em primeiro lugar, a parte mais importante na criação de um site de qualidade é fazer com que ele seja funcional. Como? Não importa, contanto que o utilizador saiba que funcione. Um utilizador genérico até é capaz de ignorar a velocidade de load do site se ele fizer o que quer que seja para o qual foi programado.

Dito isto, queres que um utilizador ao desligar o JavaScript deixe de conseguir utilizar o site, que ele deixe de ser funcional? Sendo que para esse utilizador o site não funciona, ele certamente não irá voltar.

Nada melhor que a imagem abaixo para explicar como deve ser o processo:



O princípio da melhoria progressiva do Ajax é bem conhecido, porque comumente é utilizado de forma discreta no JavaScript, como mostra a imagem acima.

Então, o que deves fazer é construir o teu site para funcionar sem JavaScript, e só depois adicionar JavaScript como sendo um acessório, da mesma forma que utilizas o HTML como linguagem de marcação e depois o CSS para melhorar o visual da página.

Informa sempre o utilizador do que está a acontecer

Não há nada pior que clicar em algo e não ver nada a acontecer por 2 segundos. Os utilizadores estão habituados a clicar e avançar para a acção seleccionada, ou pelo menos, ver o progresso da acção solicitada.

Lembra-te que a tecnologia AJAX é relativamente recente, se os utilizadores não souberem o que está a acontecer vão pensar que o site não está a funcionar. Sugiro que utilizem uma mensagem não intrusiva, que mostre o estado da acção solicitada.

Usaste AJAX? Boa! A mim não me interessa

Encara a verdade, na maioria dos sites, 90% ou mais dos utilizadores não sabem o que é AJAX ou o porquê de ser uma boa tecnologia. Eu, pessoalmente, aprecio um bom script em AJAX, mas será que todos os demais também apreciam? Obviamente que não. A não ser que tenhas um site focado apenas em Profissionais Web, faz um favor aos utilizadores do teu site, e não mostres a mensagem "Feito com AJAX".

PROGRAMADOR EXCÊNTRICO

6 regras para utilizar AJAX

Eu como utilizador não me interessa saber que tipo de tecnologias e/ou magias obscuras utilizaste, desde que o site funcione para mim é o que interessa.

AJAX só no Fim

Entregar o Projecto deverá ser o objectivo número um, então adiciona o AJAX no fim do projecto ou depois que o site esteja feito. Claro que, o AJAX pode poupar que uma página seja actualizada, mas os utilizadores estão habituados à maneira antiga. Não consigo imaginar um cliente teu satisfeito com "Não está feito, mas olhe como esta caixa é actualizada sem actualizar a página!".

Eu opto sempre pela maneira antiga (na verdade, standard é provavelmente a palavra correcta), faço o site, e só depois no fim é que acrescento melhorias na sua funcionalidade

As Regras de Segurança também se aplicam

O URL do AJAX pode até estar escondido no teu código, e assim a maioria dos utilizadores não o vê, mas aposto que consigo encontra-lo. Se é possível de ser encontrado é possível de ser explorado. Não assumem que só porque fizeram o código do vosso formulário e/ou página à prova de bala que um

utilizador não pode manipular o vosso script.

Certifiquem-se que validam todas as variáveis GET e POST antes de executarem qualquer acção do vosso script.

AJAX poupa tempo no Load... Mas o teu Framework JavaScript não

O teu script em AJAX consegue poupar uma actualização da página ao utilizador, sem carregar o header, footer e sidebar? Ótimo. O teu Framework JavaScript tem 80kb? Trabalho em vão.

Certifica-te que apenas carregas o teu Framework quando for realmente necessário, e não adiciones mais código do que o absolutamente necessário para a página. Poupar tempo de load, neste caso, não é eficiente ou user-friendly.

“ Eu opto sempre pela maneira antiga ... faço o site, e só depois no fim é que acrescento melhorias na sua funcionalidade ”

AUTOR



Escrito por Carlos Gonçalves

Finalista em Engenharia Informática de Gestão na Escola Superior de Tecnologia de Setúbal do Instituto Politécnico de Setúbal. Actualmente trabalho como Freelancer em Web Development, sou uma pessoa tendencialmente optimista e adapto-me bem a novas circunstâncias. Sempre disposto a aprender, desde o início identifiquei-me com o meu trabalho e gosto de fazer as coisas de maneira simples e objectiva.

VISUAL (NOT) BASIC

Windows Phone 7

Que melhor maneira poderia existir para demonstrar o poder da .NET Framework do que uma verdadeira demonstração de união?

O resultado é a entrada da linguagem Visual Basic .NET na corrida móvel de última geração: o Windows Phone 7.

O desenvolvimento é feito utilizando o poderoso Visual Studio, com todas as comodidades a ele associadas.

Acesso ao ambiente de desenvolvimento

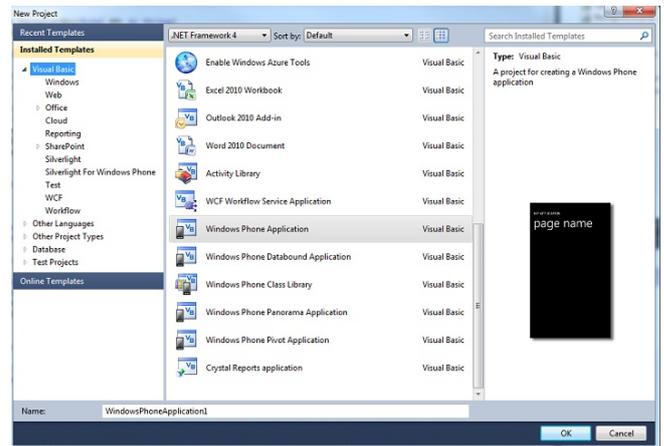
À semelhança das versões Mobile, precisamos de instalar kits de desenvolvimento que se vão integrar com o nosso Visual Studio. Para os interessados que não tenham mais do que a versão Express do Visual Basic, lamento informar que o kit não se lhe associa. Utilizadores de Windows XP: também nada podem fazer.

Para desenvolver para Windows Phone 7 (WP7) sem qualquer custo, podemos usar o Visual Studio Express for Windows Phone, mas condicionados a C#.

O que existe agora é uma versão do kit de desenvolvimento que se associa ao Visual Studio 2010 Professional e só aí é possível começar a escrever aplicações em VB.

Nesta solução, não podemos usar XNA nem Expression Blend pois ainda não são suportados.

Depois de ter um Visual Studio suportado, precisamos de instalar as Windows Phone Developer Tools, que preparam o ambiente necessário para emular um "telefone", e de seguida a CTP que vai acrescentar as templates e suporte para os projectos "Mobile" no nosso Visual Studio para o Visual Basic. No final ficarão disponíveis algumas templates dos projectos mobile, como podem observar na seguinte imagem.



O ambiente de desenvolvimento

Para quem já estava familiarizado com aplicações Silverlight no VB, o ambiente já é confortável.

O designer do lado esquerdo, onde podemos não só visualizar a interpretação do XAML (eXtensible Application Markup Language) mas também provocar alterações visuais que se reflectem no XAML.

O manifesto XAML propriamente dito a meio e os já conhecidos "Project Explorer" e "Properties" à direita. Tudo isto é obviamente configurável das formas a que já estamos habituados nas versões anteriores do Visual Studio.

Tipos de página

Existem vários 3 tipos de páginas base, onde todos os outros controlos são dispostos: Portrait/Landscape, Panorama e Pivot, cada um deles com características distintas.

Portrait/Landscape

É este o tipo de página base que é empregue ao criar um novo projecto para uma aplicação.

Trata-se de um modo de visualização regular, onde os conteúdos são dispostos no espaço visível disponível, sem qualquer particularidade.

As duas denominações, portrait e landscape, derivam

VISUAL (NOT) BASIC

Windows Phone 7

apenas da forma como estamos a segurar o telefone, e consequentemente da forma como estamos a olhar para o ecrã.

Existem apenas duas formas possíveis: ou temos o telefone orientado em "retrato", vertical, ou em "paisagem", horizontal. Às duas orientações possíveis acrescentam-se as variações inversas, onde a orientação de 0° é a inversa da orientação 180° mas no entanto ambas verticais, e 90° é inversa da orientação 270° mas no entanto ambas horizontais.

A orientação é automaticamente adoptada sempre que os recursos físicos do aparelho detectem variação na orientação da força gravitacional exercida sobre o mesmo.

Panorama

Panorama é um tipo de página base que nos permite dispor os itens num plano horizontal, virtualmente infinito.

Apesar de existirem separações distintas entre as várias páginas do panorama, a navegação é feita de forma contínua, como que se esteja a arrastar uma folha que se observa através de uma abertura.

A navegação pode ser feita a qualquer altura, em ambas as direcções.

É um modo de visualização ideal para apresentar grandes quantidades de informação relacionada que de outra forma se teriam de distribuir por vários ecrãs.

Pivot

Pivot é um tipo de página base bastante semelhante



com o Panorama.

A grande diferença reside numa separação mais distinta dos vários grupos: não existe tanto a ideia da continuidade.

Finalmente um exemplo

A melhor forma de expor todos os aspectos básicos para vos iniciar no mundo WP é, e será sempre, a execução de uma pequena aplicação.

O que se vai construir é um leitor de RSS 2.0 (Real Simple Syndication), ainda que não se aproveitem todas as capacidades do padrão 2.0.

1- Criar um projecto com o template "Windows Phone Application"

O ambiente vai carregar com todos os elementos familiares, e anteriormente descritos. A página que têm no "designer" é uma página Portrait/Landscape.

2- O leitor de RSS que vamos criar lê dois RSS simultaneamente. Poderia ser apenas um, ou poderiam ser virtualmente infinitos, mas vamos criar apenas 2 "hardcoded" para demonstrar mais tarde as páginas "pivot" e não confundir muito quem está a começar. Criem duas caixas de texto com nomes "txRSS1" e "txRSS2" e de seguida um botão de nome "btLer". Usem também os pontos de controlo nos centros de das arestas da esquerda e direita para ancorar o controlo aos limites horizontais do container. Isto fará com que o controlo "estique" horizontalmente caso o container também estique.

3- Trocar os textos dos títulos na parte superior para identificarem a página e a aplicação.

4- Testem a aplicação. Ainda não faz nada a este ponto, mas é importante que tentem escrever algo nas caixas de texto para ver como é que o WP7 gere a introdução de escrita.

VISUAL (NOT) BASIC

Windows Phone 7



Métodos de Introdução

Notaram certamente que ao tocar na textbox surgiu um teclado QWERTY virtual.

No contexto das aplicações para aparelhos de comunicação móveis, as letras não são suficientes para cobrir todas as necessidades.

Existem 5 principais métodos de introdução:

URL

Disponibiliza introdução para o âmbito web

Number

Disponibiliza introdução para o âmbito de escrita numérica

Text

Disponibiliza introdução para o âmbito de escrita

TelephoneNumber

Disponibiliza introdução para o âmbito telefónico

EmailNameOrAddress

Disponibiliza introdução para o âmbito de escrita alfanumérica

Estes tipos de introdução devem ser alterados no XAML, na propriedade `InputScope` de cada caixa de texto, utilizando as denominações descritas.

Os diferentes métodos de introdução disparam diferentes teclados virtuais, para que a velocidade e comodidade de introdução seja óptima.

Em desenvolvimento, o emulador não permite introdução através do nosso teclado físico e existe um atalho para mudar essa situação. Se carregarem em `Pause/Break` passa a ser possível introduzir escrita com o teclado do computador.



5- A leitura dos “feeds” vai ser feita numa outra página, uma página pivot onde vamos organizar cada um dos dois “feeds” em itens diferentes do pivot.

Através do menu de contexto que se obtém com o clique direito do rato sobre o nome do projecto no

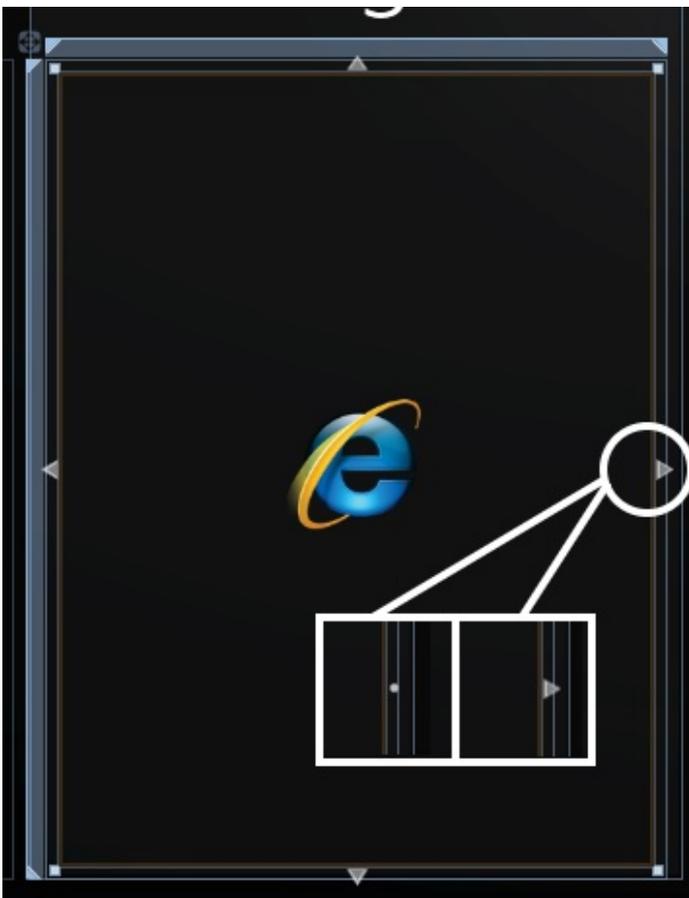
VISUAL (NOT) BASIC

Windows Phone 7

Solution Explorer, escolham "Add" e depois "New item". De entre os templates escolham "Windows Phone Pivot Page" e apliquem o nome "Leitor.xaml". A nova página surge no editor.

6- Depois de trocar os títulos superiores, adicionem um componente "WebBrowser". Disponham-no como na imagem e usem os pontos de controlo nos centros de cada aresta para ancorar o controlo aos limites do container. Isto fará com que o controlo "estique" caso o container também estique.

7- Por defeito uma pivot page já vem com



dois itens de do pivot, que neste caso são exactamente quantos precisamos.

Depois de colocar o "WebBrowser" num item, para não complicar, teremos que o duplicar para o segundo item.

Isto pode ser alcançado através do designer mas torna-se muito mais fácil se identificarem a linha que representa o "WebBrowser" no XAML e a copiarem para o outro item do pivot.

```
<!--Pivot item one-->
<controls:PivotItem Header="A carregar..." Name="RSS1">
  <Grid>
    <phone:WebBrowser Margin="10,10,0,9" Name="wbVista1" UseLayoutRounding="True" />
  </Grid>
</controls:PivotItem>
<!--Pivot item two-->
<controls:PivotItem Header="A carregar..." Name="RSS2">
  <Grid>
    <phone:WebBrowser Margin="10,10,9,9" Name="wbVista2" UseLayoutRounding="True" />
  </Grid>
</controls:PivotItem>
```

8- Depois de colocar os dois "WebBrowser", e de lhes colocar os nomes "wbVista1" e "wbVista2", precisamos de adicionar uma referência necessária à forma como vamos ler XML, que é o formato dos "RSS feeds".

Temos de escolher "Add reference" através do menu de contexto resultante do clique direito do rato sobre o nome do projecto, no Solution Explorer. Vamos assim, adicionar referência a "System.Xml.Linq"

9- Depois de todos os componentes se encontrarem nas suas posições está na altura de começar a escrever código. Voltemos à primeira página. É necessário fazer com que o botão valide as introduções nas caixas de texto e que envie os seus valores para a segunda página.

Navegação entre páginas

Para quem sempre programou Visual Basic, a navegação entre páginas vai parecer no mínimo estranha.

Podemos imaginar o WP7 como um browser que nos apresenta páginas baseadas numa linguagem de marcação localizada algures, apontada por um URL: o ficheiro XAML existe, construímos um URI que nos aponte a ele, neste caso através de um caminho relativo, e utilizamos essa instância do novo URI no método Navigate do NavigationService.

Depois de carregar no botão, a página actual "perde-

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.Windows.RoutedEventArgs) Handles Button1.Click
    NavigationService.Navigate(New Uri("/Leitor.xaml"), UriKind.Relative)
End Sub
```

VISUAL (NOT) BASIC

Windows Phone 7

se" e a nova é carregada e apresentada.

Passagem de valores entre duas páginas

Com o sistema de navegação equiparado a um browser, sem poder criar instâncias de objectos Form, como normalmente fazemos no Visual Basic Windows Forms, transportar valores como um simples número poderiam tornar-se em tarefas complicadas.

Já que para a navegação entre páginas temos de construir um URI, é perfeitamente possível passar parâmetros através do URL, o que já reduz significativamente a complexidade para transportar valores.

Como pretendemos passar dois endereços distintos, podemos adicionar os dois parâmetros ao URL e transportar com eles os valores de texto das caixas de texto.

Depois de carregar no botão, ocorre exactamente o

```
NavigationService.Navigate(New
Uri(String.Format("/Leitor.xaml?rss1={0}&rss2
={1}", txRSS1.Text, txRSS2.Text),
UriKind.Relative))
```

mesmo que numa navegação normal, pois estamos a apontar à mesma para uma página, mas o do cabeçalho já constam os valores passados no URL. Juntando à passagem de valores a validação simples das caixas, ficamos com:

10- Depois do transporte de valores estar

```
Private Sub btLer_Click(ByVal sender As
System.Object, ByVal e As
System.Windows.RoutedEventArgs) Handles
btLer.Click
    If txRSS1.Text = String.Empty Or
txRSS1.Text = String.Empty Then
MessageBox.Show("URLs vazios!") : Exit Sub
    NavigationService.Navigate(New
Uri(String.Format("/Leitor.xaml?rss1={0}&rss2
={1}", txRSS1.Text, txRSS2.Text),
UriKind.Relative))
End Sub
```

garantido, temos também de garantir que conseguimos resgatar esses valores na nova página.

Resgatar valores do cabeçalho da página

Depois de ter a página carregada (e só depois de estar completamente carregada, isto é importante) podemos resgatar os valores do cabeçalho da página, utilizando o NavigationContext, e servirmo-nos deles da forma que entendermos.

11- Ao conseguirmos agregar toda a

```
Private Sub Leitor_Loaded(ByVal sender As
Object, ByVal e As
System.Windows.RoutedEventArgs) Handles
MyBase.Loaded
    Dim RSS1URL As String =
NavigationContext.QueryString("rss1")
    Dim RSS2URL As String =
NavigationContext.QueryString("rss2")
End Sub
```

informação na segunda página, e da segunda página estiver completamente carregada, é altura de fazer as chamadas, procedimentos e handlers necessários para fazer os pedidos dos itens dos "feeds".

```
Private Sub Leitor_Loaded(ByVal sender As
Object, ByVal e As
System.Windows.RoutedEventArgs) Handles
MyBase.Loaded
    Dim RSS1URL As String =
NavigationContext.QueryString("rss1")
    Dim RSS2URL As String =
NavigationContext.QueryString("rss2")

    Dim CRSS1 As New CarregarRSS(RSS1URL)
    Dim CRSS2 As New CarregarRSS(RSS2URL)

    AddHandler CRSS1.ChegouTitulo, AddressOf
ChegouTitulo1
    AddHandler CRSS2.ChegouTitulo, AddressOf
```

VISUAL (NOT) BASIC

Windows Phone 7

```
ChegouTitulo2
    AddHandler CRSS1.HTMLPronto, AddressOf
HTMLPronto1
    AddHandler CRSS2.HTMLPronto, AddressOf
HTMLPronto2
End Sub

Private Sub ChegouTitulo1(ByVal Titulo As
String)
    RSS1.Header = Titulo
End Sub

Private Sub ChegouTitulo2(ByVal Titulo As
String)
    RSS2.Header = Titulo
End Sub

Private Sub HTMLPronto1(ByVal HTML As String)
    wbVista1.NavigateToString(HTML)
End Sub

Private Sub HTMLPronto2(ByVal HTML As String)
    wbVista2.NavigateToString(HTML)
End Sub
```

12- Adicionem ao projecto o ficheiro CarregarRSS.vb, cujo link podem encontrar no final do artigo. Trata-se da classe criada para ler os feeds, que não é relevante para o desenvolvimento em WP7.

13- A aplicação está completa.

Corram agora a aplicação. O primeiro ecrã apresenta as duas caixas.

Preenham as caixas com dois feeds diferentes, por exemplo:

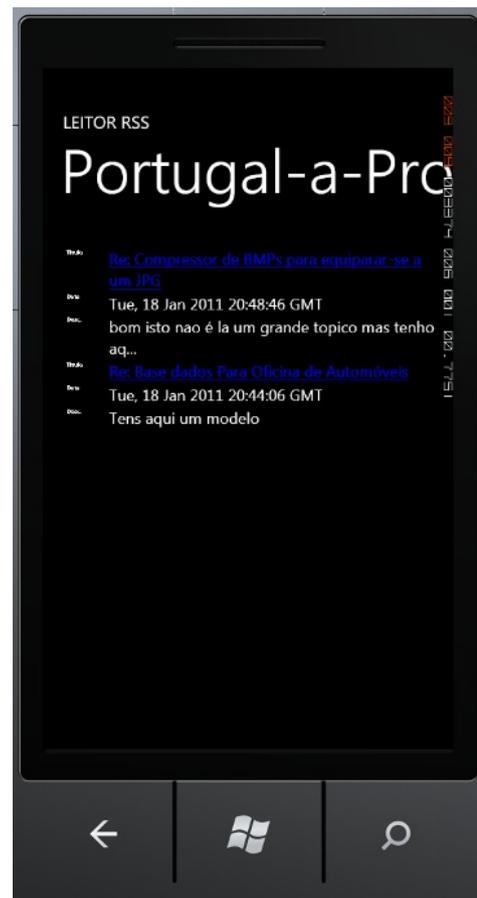
O “nosso” feed:

<http://www.portugal-a-programar.org/forum/index.php?type=rss;action=.xml>

O feed de notícias do JN:

<http://feeds.jn.pt/JN-ULTIMAS>

Depois de carregar no botão, a segunda página é carregada e é apresentado o feed da primeira caixa de texto. Podemos passar para o segundo feed arrastando o título do feed para a esquerda ou para a direita.



Orientação do aparelho

Se têm a oportunidade de estar a fazer debug directamente no telefone, já devem ter reparado com certeza no comportamento da aplicação quando inclinamos o telefone nas posições que normalmente disparam mudanças de apresentação noutras aplicações nativas: não acontece rigorosamente nada.

No emulador, podem inclinar o aparelho em 90° através dos botões para o efeito, disponíveis no menu do lado direito.

Por defeito, as páginas estão configuradas para

VISUAL (NOT) BASIC

Windows Phone 7

ignorar qualquer outra orientação que não seja Portrait.

A propriedade `SupportedOrientations` dita as orientações suportadas por aquela página e na maioria dos casos, passá-la para `PortraitOrLandscape` resolve o layout para a orientação pretendida, mas também é obviamente possível controlarmos os comportamentos.

Troquem a propriedade `SupportedOrientations` no XAML de cada uma das duas páginas para: `PortraitOrLandscape`



Conclusão

Ainda que diferentes, nota-se perfeitamente o esforço que a Microsoft está a fazer para substituir Windows Forms por Windows Presentation Framework.

O desenvolvimento em WP7 assenta em Windows Presentation Framework com todas as vantagens conhecidas da .NET Framework.

Os IDE para os quais o desenvolvimento WP7 se aplica estão muito bem conseguidos e bastante polidos, o que torna a experiência bastante agradável e produtiva, desviando o foco do programador para pormenores que não interessam tanto como a aplicação em si.

Não tive oportunidade de fazer debug no aparelho físico, mas se estiver tão boa como o antigo Windows Mobile já está com nota positiva.

Links

VB CTP para WP7:

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=a808a69d-6119-47b7-b858-262be5c49915>

WP Developer Tools RTW:

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=04704acf-a63a-4f97-952c-8b51b34b00ce>

AUTOR



Escrito por **Sérgio Ribeiro**

Curioso e auto-didacta com uma enorme paixão por tecnologias de informação e uma saudável relação com a .NET framework.

Moderador do quadro de Visual Basic.NET na comunidade Portugal@Programar desde Setembro de 2009.

Alguns frutos do seu trabalho podem ser encontrados em <http://www.sergioribeiro.com>

gcc -Wall myApp.c -linstantaneous -o success

Desde a década de 1970 que a área de informática tem sido prolifera na geração de empresas, empreendedores e aplicações de sucesso. Estes sucessos, muitas vezes bombásticos e sempre mediáticos, parecem indicar que é muito fácil criar um negócio multi-milionário do dia para a noite. Quando um bom programador olha para uma aplicação de sucesso identifica imediatamente um conjunto de variáveis que o levam, invariavelmente, a concluir “Eu sou capaz de fazer isto!” e costuma rematar com “E até sou capaz de o fazer melhor!” E tipicamente um bom programador diz a verdade numa situação destas.[1]

Se juntarmos à capacidade técnica o baixo custo actual de criar uma aplicação e de a disponibilizar para todo o mundo ficamos a pensar porque razão apenas uma ínfima a quantidade de programadores cria e gere o seu próprio negócio e porque não são todos multi-milionários.

A resposta é bastante simples, e ao contrário de 99% das respostas dadas por quantos lêem esta coluna, a razão não é dinheiro para arrancar. A questão também não é técnica nem tecnológica mas sim de personalidade. Apenas uma ínfima parte dos programadores é realmente empreendedor e desses nem todos têm a coragem de arriscar a voar por si. E todos quantos arriscam mudar de vida sofrem na pele todas as torturas e maus tratos que advém da gestão de um negócio próprio, a começar pelo facto de que não existem sucessos instantâneos, muito menos para negócios multi-milionários. Sem dúvida que há excepções à regra, mas mesmo essas excepções que tiveram uma capitalização exponencial sofreram as dores de cabeça de um negócio pequeno onde, cada dia passado era uma vitória e sem haver certezas quanto ao mês seguinte.

Uma lição muito importante para quem pretende ser empreendedor é compreender que a tecnologia é

totalmente irrelevante para o sucesso, excepto, obviamente, se o negócio se baseia precisamente na tecnologia em si. Aprender esta lição é de extrema importância e muitas vezes não é compreendida por quem faz o desenvolvimento. Ninguém compra um produto por este ser desenvolvido em determinada tecnologia.

Outra lição de extrema importância é que um programador não é um designer e também não é um utilizador comum, o que tipicamente o leva a desenhar aplicações funcionalmente más. E como se não bastasse ainda fica ofendido com os utilizadores por estes não compreenderem que para activar a assinatura digital apenas têm de correr determinado comando numa shell com permissões de administração![2]

“ (...) um programador não é um designer e também não é um utilizador comum (...) ”

Uma terceira lição de extrema importância é que uma aplicação não é um produto. Um produto é constituído por muito mais do que uma aplicação de software. Por exemplo, um produto tem um manual de instruções que efectivamente explica o produto e como operar com o mesmo.

Por esta altura muitos estão a pensar “mas eu compreendo tudo isto, acho que tenho veia de empreendedor”. A todos vocês que pensaram isso vou lançar um desafio. Respondam às seguintes questões:

1. Quantos projectos vossos já idealizaram?
2. Desses, quantos começaram?
3. Dos projectos que começaram, quantos terminaram?

gcc -Wall myApp.c -linstantaneous -o success

4. Dos projectos que terminaram, quantos lançaram para o mercado?

Se a resposta foi “nenhum” à questão 1, então vocês não são empreendedores. Se a resposta foi “nenhum” à questão 2, então vocês não são empreendedores. Se a resposta foi “nenhum” à questão 3, então vocês não são empreendedores. Se a resposta foi “nenhum” à questão 4, então vocês não são empreendedores. A minha aposta é que a larga maioria de vocês respondeu “imensos” às duas primeiras questões, mas muito poucos responderam “um” ou “dois” e muitíssimo poucos terão respondido “um” à quarta pergunta.

Acima de tudo um empreendedor “ships it”. Um empreendedor tem, pelo menos, um projecto que já terminou. Só há produto quando este se encontra concluído. No entanto um empreendedor tem também de saber quando desistir de um projecto e saber adaptar o seu produto ao longo do caminho. Tipicamente a ideia inicial é apenas isso mesmo, uma ideia que se tem no início. É apenas a semente que, bem cuidada, germinará algo de que poderemos oferecer e alguém quererá comprar. Resumindo, e para os que ainda continuam animados, desenvolver um produto é a parte fácil.

O empreendedorismo, de forma geral, não se esgota na construção do produto. Na verdade isso é apenas um meio para atingir um fim. E o fim é a criação de um negócio, inicialmente sustentável e mais tarde de sucesso.

E também aqui é necessário explicar que o sucesso não existe apenas se conseguirmos entrar para a Fortune 500. A maioria dos empreendedores considera o seu negócio um sucesso quando este lhes dá aquilo que o empreendedor procura na vida. Mas para lá chegar há um longo caminho a percorrer. Tipicamente esse caminho começa com uma ideia, mas antes da sua concretização é necessário passar por vários estados, sendo um dos mais importantes o da validação da ideia.

Muitas vezes o programador aspirante a empreendedor valida a ideia com amigos e familiares. Aqui acontecem duas coisas, quem tem conhecimentos técnicos compreende a coisa e apaixonar-se pela ideia e diz “espectacular” e quem não tem conhecimentos técnicos acredita no que lhe é dito e diz que “é boa ideia”. Há vários erros aqui, um deles é que os amigos técnicos compram a parte sexy do desenvolvimento da aplicação, outro é que o aspirante a empreendedor não sabe explicar a uma qualquer pessoa o que realmente vai ser essa sua ideia. Mas o erro mais grosseiro é não saber responder à pergunta “que problema é que isso resolve?” Se não houver um problema, então não é necessária nenhuma solução, pelo que a ideia cairá por terra. Havendo um problema, e quanto mais difícil de resolver melhor, então tal terá de ser expresso numa frase curta e clara, finda a qual a nossa tia-avó compreende perfeitamente que produto será construído.[3]

“ (...) que problema é que isso resolve? ”

Para todos os que ainda não abandonaram a leitura desta coluna furiosos comigo por lhes estar a deitar por terra ser o próximo multi-milionário da web, peço apenas mais um minuto.

É que mesmo com uma ideia validada e uma vontade inabalável de desenvolver um software que se vai vender como bolos quentes porque resolve um problema complexo a um número infindável de pessoas, há imensas variáveis que o programador aspirante não pensou ou, pior, pensou de forma leviana. É necessário ter a consciência de que não basta construir um bom produto de software, é necessário vendê-lo. Isto implica uma estratégia de marketing e de vendas. E não, gastar uns trocos, ou mesmo uns milhares, em Google Ads não é suficiente. Da mesma forma, não é suficiente colocar um link para download da aplicação para que alguém a use. Da mesma forma, não é suficiente ter um

gcc -Wall myApp.c -linstantaneous -o success

manual de instruções, é necessário estar preparado para dar suporte à aplicação e, em alguns casos, até formação.

E enquanto tudo isto se desenrola é necessário tratar da burocracia inerente ao negócio, recolher feedback dos utilizadores, corrigir bugs em produção, ajustar o road map para a próxima release, melhorar a

“ **Dos projectos que começaram, terminaram? quantos** ”

funcionalidade, escolher um novo plano de alojamento, dar o produto a conhecer a mais pessoas, pagar as contas do mês e, no meio de tudo isto, ter umas horas para dormir por dia e tentar fazer com que o termo “fim-de-semana” passe a ter o mesmo significado que os nossos amigos não-empresendedores lhe dão...

Ainda sobrou alguém a ler? Então estás à espera de quê? Valida a tua ideia e mão à obra. Garanto-te que é uma experiência profissional única da tua vida, e quem sabe, talvez venhas mesmo a ser o próximo multi-milionário! Boa sorte! ;)

[1] *A prová-lo está o facto de determinada aplicação de sucesso ser clonada por várias empresas e por vezes até melhor do que original. Mas só ser bom não basta, e por isso os clones raramente destronam quem chegou primeiro.*

[2] *Todos quantos não compreenderam o que acabei de dizer nesta linha têm duas hipóteses: (i) esquecer o empreendedorismo, ou (ii) contactar-me para vos explicar o que isto tem de mal.*

[3] *Para os mais cépticos deixo um desafio. Olhem para o vosso desktop e vejam se têm alguma aplicação instalada que não vos seja útil. Todas elas vos resolvem um problema, seja apenas ouvir um MP3 ou efectuar o controlo de versões em ambientes distribuídos.*



AUTOR



Escrito por **Fernando Martins**

Faz parte da geração que se iniciou nos ZX Spectrum 48K. Tem um Mestrado em Informática e mais de uma década de experiência profissional nas áreas de Tecnologias e Sistemas de Informação. Criou a sua própria consultora sendo a sua especialidade a migração de dados.

COMUNIDADES

Desenvolvimento em SharePoint 2010

GuiaTV CoolthingsPT

Habilitar external users no SharePoint online – Office365

O Futuro da Programação Assíncrona da Plataforma .NET

Desenvolvimento em SharePoint 2010

Developer em Sharepoint?

Desde muito cedo que esta mania\lara\obsessão por debitar linhas de código pelo puro prazer de mover um pixel de um lado para o outro, tem-me atormentado como se de um vírus se tratasse. Não sei se é mania ou não mas sempre gostei de assuntos que me fizessem pensar e criar algo novo. Sim , eu confesso na minha essência sou um Developer , e presumo que vocês que lêem estas linhas também o são, senão não teriam chegado até aqui.

Um developer, e isto na minha opinião terá de ter sempre uma percepção global das coisas e focus no desenvolvimento, um pouco de bom senso, e muita paciência. Estas eram as métricas gerais seguidas por mim até meados de 2004. SharePoint... Pareceu-me interessante a aposta, e aqui estou.

Não desfazendo os outros tão prestigiados "cargos" de IT, Project Manager, and whatever, penso que a ocupação de Developer é a mais cool de todas. Um developer de SharePoint obrigatoriamente tem mais valências de que só a de desenvolvimento: somos Administradores de Sistemas, Helpdesks, Arquitectos , Testers, DBA , Designer e utilizadores finais da plataforma. Em SharePoint é extremamente rigorosa a interacção com a plataforma mas torna-se gratificante as tecnologias que abrange e a potencialidade das áreas que toca é surpreendente.

Microsoft SharePoint Server, o que é?

Embora muitas pessoas já tenham uma ideia clara do que o produto SharePoint representa, existem ainda algumas dúvidas relativamente ao que este produto pretende atingir.

De uma forma muito sucinta SharePoint é uma plataforma corporativa Microsoft assente sobre áreas funcionais.

Nesta última versão, 2010 e existem 6 cores

aplicacionais:



Sites: Criação e Gestão de sites Internos e Externos. Infraestrutura única para todos os sites empresariais Web : Intranet, Extranets e Internet

Communities: Ambiente de colaboração social, Ferramentas com capacidades sociais, como blogs, wikis, rssfeeds, tagging, perfis pessoais recios em informação

Content:Gestão documental , políticas de retenção e classificação automática de conteúdos,

Search: Pesquisas com grande relevância, contexto social e refinamento

Insights: Business Intelligence com acesso a informações em bases de dados, relatórios e aplicações empresariais.

Composites: Ferramentas e componentes para criação de soluções comerciais

Sendo uma plataforma consolidada nestas diferentes áreas, com tantas funcionalidades OutOfTheBox, é clara a necessidade de customização e criação de novas, específicas a cada tipo de negócio e cliente. "Developer's , Developer's, Developer's" , sim somos

COMUNIDADE SHAREPOINTPT

Desenvolvimento em SharePoint 2010

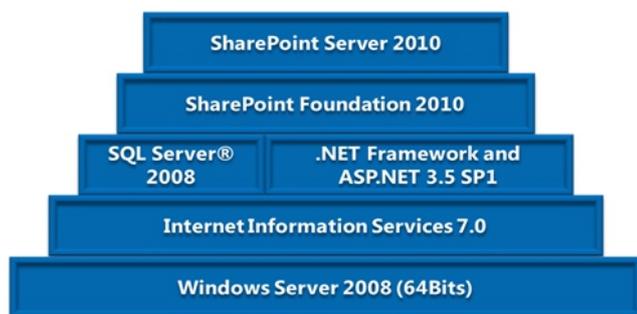
nós e estamos aqui.

SharePoint 2010 e tecnologias inerentes

Antes de tudo, é importante que um developer perceba e entenda minimamente a arquitectura da plataforma e alguns dos seus mecanismos inerentes: SharePoint é baseada na framework .NET, subdividindo-se em 2 tecnologias principais:

SharePoint Foundation: Tecnologia subjacente a todos os sites SharePoint, encontra-se disponível gratuitamente. É o core aplicacional da plataforma te. Podemos utilizar o SharePoint Foundation para criar sites tipificados de colaboração, blogs, wikis

SharePoint Server: Tecnologia servidor. Inclui todas as funcionalidades do SharePoint Foundation além de funcionalidades e capacidades adicionais como Gestão de Conteúdos, B.I., Pesquisa empresarial.



A figura acima ilustra as diferentes tecnologias inerentes à plataforma. Essencialmente temos o sistema operativo, Windows Server 2008, o IIS, SqlServer, a .Net Framework e ASP.Net, o SharePoint Foundation e o SharePoint Server em si.

Embora o SharePoint 2010 funcione em Windows 7 ou Windows Vista, não é suportado pela Microsoft em ambientes de produção e para desenvolvimento recomendo avidamente a utilização de tecnologia server. O ambiente em que desenvolvemos deverá ser semelhante ao que desenvolvemos, always!

Arquitectura

A infraestrutura do SharePoint 2010 foi pensada de forma a respeitar uma arquitectura lógica de 3 camadas: Camada dos Dados(Database Layer), à

Camada Aplicaional(Application Layer)à Camada de Apresentação(Presentation Layer). Estes diferentes layers podem estar localizados numa só máquina (Standalone Server)

Stand-Alone



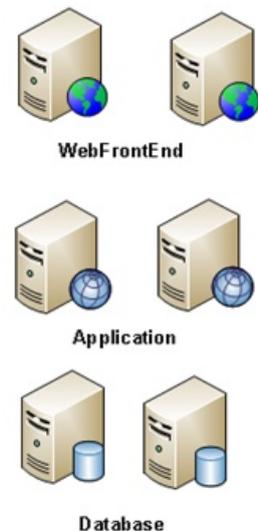
Database, Application, WebFrontEnd

ou espalhados em diversas máquinas (Small, Medium, Large Farm)

Small-Farm



Medium Farm



Fisicamente esse conceito também é real, temos servidores físicos que espelham o mesmo conceito.

O esquema físico da farm SharePoint foi pensado de forma a ser escalado horizontalmente e verticalmente, caso exista necessidade mais capacidade a nível do User Interface, podem ser adicionados mais Servidores de Frontend (Web front-end), se existir necessidade de aumento da capacidade dos Serviços, podem ser adicionados mais Servidores Aplicaionais (Application Servers). E finalmente se existir necessidade de mais capacidade a nível de base de dados, podem ser adicionados mais Servidores de Base de Dados (Database Server).

Quaisquer que sejam as necessidades, é permitido desenhar uma topologia que as complemente com uma flexibilidade extrema.

E como plataforma de desenvolvimento?

“Estás a desenvolver em coisas já feitas” comentava uma pessoa amiga um dia destes.

É verdade, a framework do SharePoint é realmente muito completa mas não é suficiente para muitos dos casos e pode ser estendida. Da mesma forma que hoje, é vulgar falarmos em desenvolvimento em .NET (que by the way também é uma coisa já feita) também é natural cada vez mais pensarmos além do que o core aplicacional e virarmos a nossa atenção para implementações corporativas de alto nível. E garanto-vos que o desafio é superior.

Com uma framework e Object model extremamente ricos em métodos e funcionalidades é possível estender a plataforma a um nível quase ilimitado.

Em termos de desenvolvimentos foca-se 2 pontos cruciais: Soluções e Features.

Uma Solução SharePoint, não é mais do que um pacote comprimido com extensão WSP, contendo componentes que caracterizam funcionalidades(Features) dentro da plataforma. Um único pacote pode ter mais da uma funcionalidade. De um forma mais simples, acaba por ser o ficheiro de instalação (vulgo Setup.exe) de aplicações em SharePoint .

Uma farm de Sharepoint pode englobar inúmeros servidores, pelo que com esta tecnologia, podemos instalar uma solução numa máquina que pertença à farm e esta é propagada e instalada em todos os servidores que pertencem a essa Farm. Cool, não?

Features, têm por objectivo reduzir a complexidade envolvida na personalização de sites simples, inclusão e manutenção de novas funcionalidades. Estão intrinsecamente ligadas ao core aplicacional e podem ser activadas via interface gráfico ou command-line.

SharePoint e o Visual Studio 2010

Aquando a saída da versão 2007 da plataforma, a integração com a ferramenta de desenvolvimento Visual Studio era muito limitada. Em meados de 2007, as soluções e features eram criadas manualmente, com edição de ficheiros XML , não existindo qualquer ferramenta out of the box ou add-in no Visual Studio que pudesse facilitar esta tarefa. Inúmeras equipas de desenvolvimento em SharePoint em todo o mundo, tiveram que criar os seus próprios automatismos e mecanismos de criação destas funcionalidades. No decorrer deste processo, foram lançadas algumas ferramentas e add-ins para o Visual Studio dos quais se destaca o WSPBuilder(<http://wspbuilder.codeplex.com/>) Com esta nova versão do SharePoint 2010 e do Visual Studio 2010, a música é outra.

Os Projectos para SharePoint 2010 estão completamente integrados com o Visual Studio 2010, existindo diversos templates que podemos utilizar:

Nome	Icon	Descrição
Empty SharePoint Project<< >>		Projecto Vazio onde se pode adicionar novos itens
Visual Web Part		Cria uma nova webpart utilizando um visual designer
Sequential WorkFlow		Cria um novo workflow sequencial
State Machine WorkFlow		Cria um novo workflows de estado
Business Data Connectivity Model		Cria um novo "SharePoint Business Data Connectivity model"
Event Receiver		Cria um event receiver, que trata os eventos disparados pelo Sharepoint
List Definition		Cria uma definição de uma lista
Content Type		Cria um Content Type Customizado
Module		Cria um módulo de SharePoint que pode conter vários ficheiros
Site Definition		Cria um template de definição de um site
Import Reusable Wrokflow		Importa um workflow criado com o SharePoint Designer
Import SharePoint Solution Package		Cria uma nova solução baseada num pacote de solução existente

Cada um destes projectos permite a inclusão de vários itens sendo que alguns dos templates são soluções vazias com pelo menos um item predefinido.

Passemos então a um caso prático: o objectivo deste que vos apresento passa por criar uma funcionalidade de registar os contactos dos actuais

COMUNIDADE SHAREPOINTPT

Desenvolvimento em SharePoint 2010

membros da Comunidade Portuguesa de SharePoint . Para tal vamos criar um tipo de conteúdo (Content Type) com alguns campos e criar uma lista baseada nesse tipo de conteúdo.

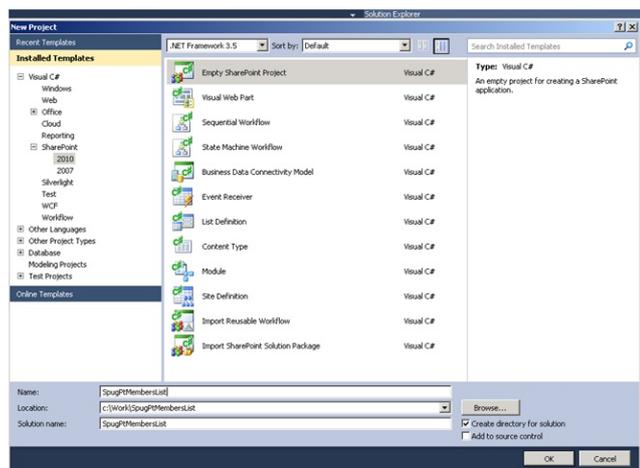
Apesar de ser um exemplo simples, acaba por demonstrar algumas funcionalidades que a plataforma disponibiliza

1. Abrir o Visual Studio, no menu principal, selecciona a opção File>New Project

•Em Installed Templates, seleccionar a opção SharePoint>2010 e Empty SharePoint Project

•Digitar o nome do projecto, e a sua localização : neste caso o nome escolhido foi SPugPtMembersList e a localização do projecto irá ser c:\work\SPugPtMembersList

•Seleccionar o botão OK



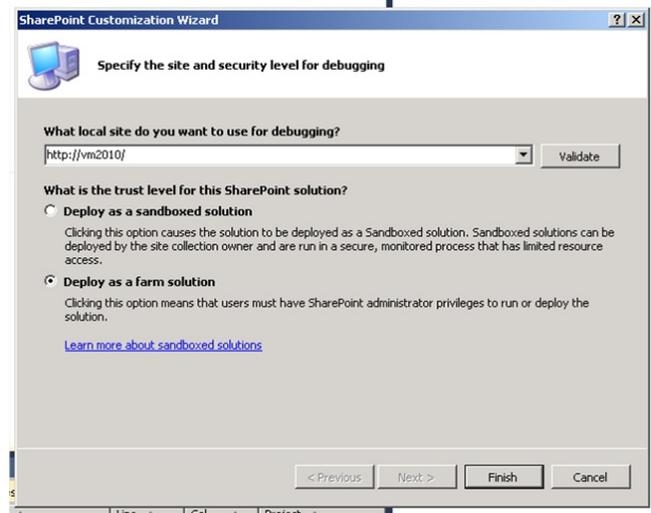
2. De seguida, surge o ecrã abaixo, com informação com o url do site SharePoint em que vamos trabalhar, neste caso é http://vm2010/ pois é o site por omissão criado aquando a instalação do SharePoint.

Nesta versão do SharePoint existem 2 tipos de soluções que podem ser criadas: Sandbox e Farm solutions.

Sandbox solutions são soluções em que os processos de execução, correm em ambientes muito controlados com restrições variadas tendo em vista um enclausuramento aplicacional em toda a Farm de SharePoint.

Farm solutions , são soluções mais abrangentes, contemplando desde o inicio uma aplicação de

funcionalidades por toda a Farm



Neste exemplo vamos criar uma Farm Solution.

•Seleccionar a opção “Deploy as a farm solution”

•Seleccionar o botão "OK”

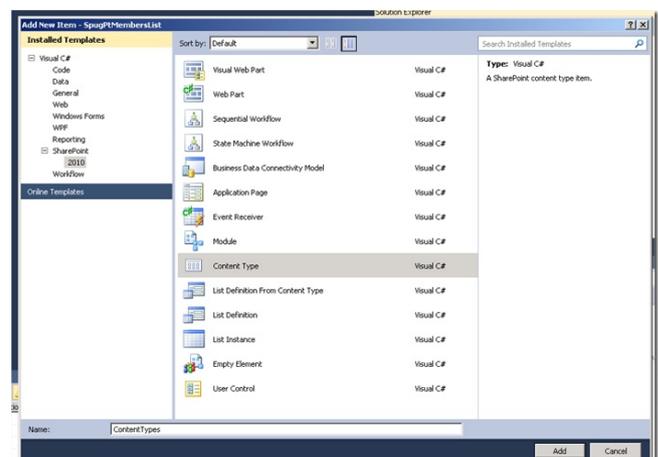
3. Temos então o projecto Visual Studio.

Começamos por adicionar alguns itens ao projecto, nomeadamente uma Feature para provisionar o nosso Content Type

•Seleccionar o projecto com o botão direito do rato e seleccionar a opção Add>NewItem

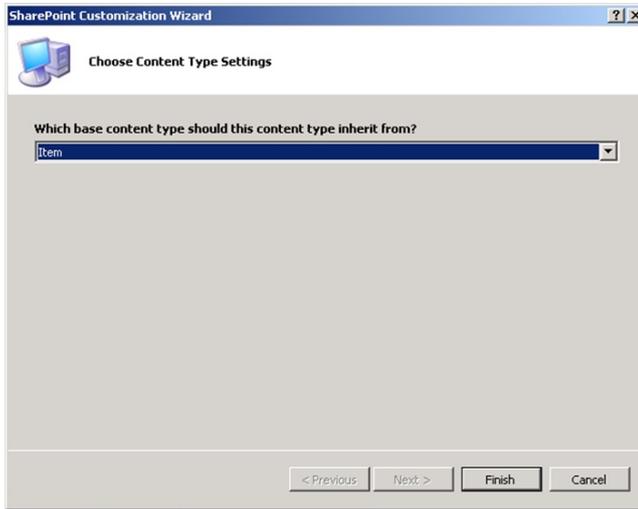
•Seleccionar o item “Content Type” e o digitar o nome “ContentTypes” para este item.

•Seleccionar o botão “Add”



4. Os content types são elementos com herança pelo que vamos seleccionar o content type mais simples de todos: Item

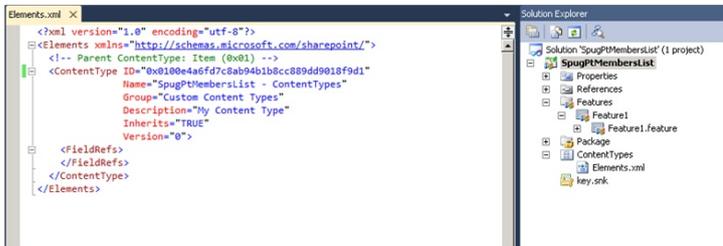
- Selecção a opção "Item"
- Selecção o botão "Finish"



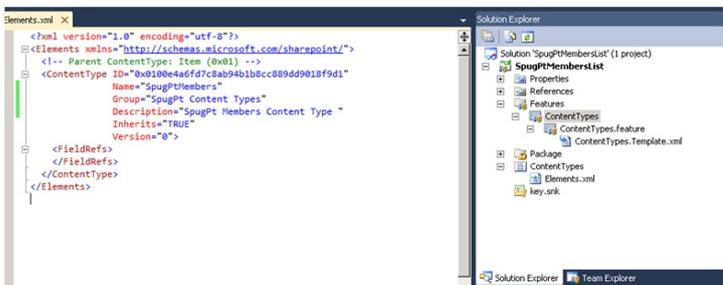
5. Neste momento, já temos a nossa Feature criada. Como podem ter verificado por omissão o Visual Studio cria estes itens com nomes e conteúdos com denotações automáticas.

Desta forma:

- Editar na window "Solution Explorer" o nome da "Feature1" para "ContentTypes"



- Editar o ficheiro Elements.xml com conteúdo mais contextualizado e lógico



- Após edição do campos Name, Group, Description e o nome da Feature temos: Resta acrescentar alguns campos, de forma a caracterizar o nosso Content Type:

```
<Field ID="{FBAFB32B-D2FA-4BE6-BA7C-CAA3E9534545}" Name="MemberName"
  DisplayName="Name" Group="SpugPtMembers
  Columns" Type="Text" Required="TRUE"
  ShowInViewForms="TRUE"
  ShowInDisplayForm="TRUE" ShowInNewForm="TRUE"
></Field >
<Field ID="{4DBBB65D-A401-41E8-BA71-C200EEFBB7F3}" Name="MemberCompany"
  DisplayName="Company" Group="SpugPtMembers
  Columns" Type="Text" ShowInViewForms="TRUE"
  ShowInDisplayForm="TRUE" ShowInNewForm="TRUE"
></Field >
<Field ID="{706647BC-77EA-46E0-A4F3-CCD9BD291A85}" Name="MemberEmail"
  DisplayName="Email" Group="SpugPtMembers
  Columns" Type="Text" Required="TRUE"
  ShowInViewForms="TRUE"
  ShowInDisplayForm="TRUE" ShowInNewForm="TRUE"
></Field >
<Field ID="{B7F1842E-4862-43F9-8D4A-9D6C0B5A2BC4}" Name="MemberPhone"
  DisplayName="Phone" Group="SpugPtMembers
  Columns" Type="Text" ShowInViewForms="TRUE"
  ShowInDisplayForm="TRUE" ShowInNewForm="TRUE"
></Field >
```

- Inserir os seguintes campos , como 1ºs elementos do ficheiro Elements.xml

- E inserir as seguintes referências aos mesmos dentro do elemento <FieldsRefs>:

```
<FieldRef ID="{FBAFB32B-D2FA-4BE6-BA7C-CAA3E9534545}" Name="MemberName" />
<FieldRef ID="{4DBBB65D-A401-41E8-BA71-C200EEFBB7F3}" Name="MemberCompany" />
<FieldRef ID="{706647BC-77EA-46E0-A4F3-CCD9BD291A85}" Name="MemberEmail" />
<FieldRef ID="{B7F1842E-4862-43F9-8D4A-9D6C0B5A2BC4}" Name="MemberPhone" />
```

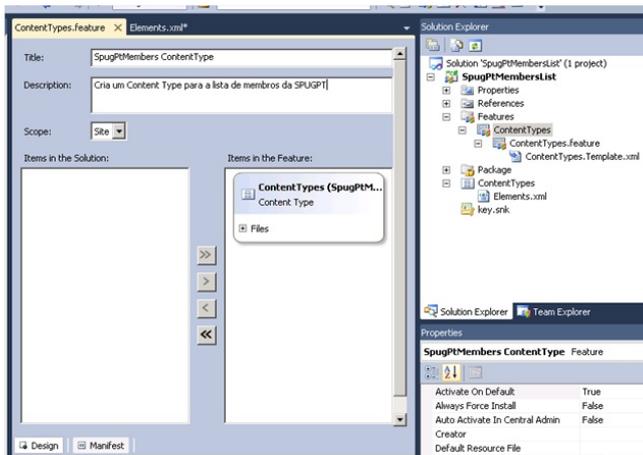
Ficamos então com algo como :

COMUNIDADE SHAREPOINTPT

Desenvolvimento em SharePoint 2010

```
<?xml version="1.0" encoding="utf-8" ?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <Field ID="{FBAFB32B-D2FA-4BE6-BA7C-CAA3E9534545}" Name="MemberName" DisplayName="Name" Group="" />
  <Field ID="{4088B65D-A401-41E8-BA71-C200EEFB7F3}" Name="MemberCompany" DisplayName="Company" Group="" />
  <Field ID="{706647BC-77EA-46E0-A4F3-CCD98D291A85}" Name="MemberEmail" DisplayName="Email" Group="" />
  <Field ID="{B7F1842E-4862-43F9-8D4A-9D6C0B5A2BC4}" Name="MemberPhone" DisplayName="Phone" Group="" />
  <!-- Parent ContentType: Item (0x01) -->
  <ContentType ID="0x0100e46fd7c8ab94b1b8cc889dd9018f9d1"
    Name="SpugPtMembers"
    Group="SpugPt Content Types"
    Description="SpugPt Members Content Type"
    Inherits="TRUE"
    Version="0">
    <FieldRefs>
      <FieldRef ID="{FBAFB32B-D2FA-4BE6-BA7C-CAA3E9534545}" Name="MemberName" />
      <FieldRef ID="{4088B65D-A401-41E8-BA71-C200EEFB7F3}" Name="MemberCompany" />
      <FieldRef ID="{706647BC-77EA-46E0-A4F3-CCD98D291A85}" Name="MemberEmail" />
      <FieldRef ID="{B7F1842E-4862-43F9-8D4A-9D6C0B5A2BC4}" Name="MemberPhone" />
    </FieldRefs>
  </ContentType>
</Elements>
```

6. Resta-nos ainda caracterizar a nossa feature .
Desta forma seleccionar a feature “ContentTypes” , e com o botão do lado direito, seleccionar a opção “View Designer”
Editar os campos Title e Description como mostra a figura abaixo.
Não esquecer de seleccionar o Scope da Feature para Site

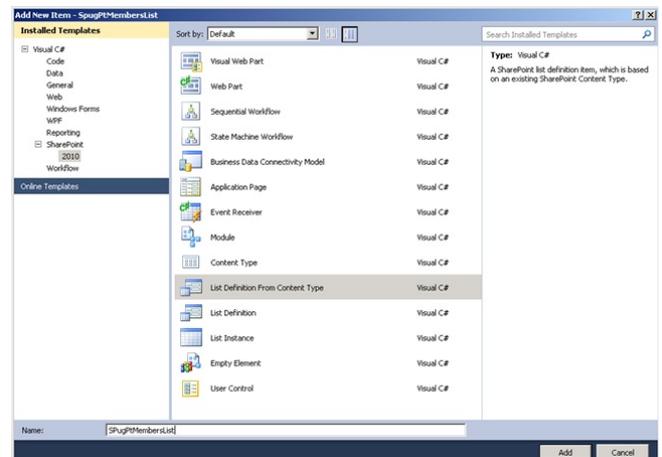


Neste momento temos a nossa Feature de Content Type , claramente definida, com conotações lógicas e campos inerentes.

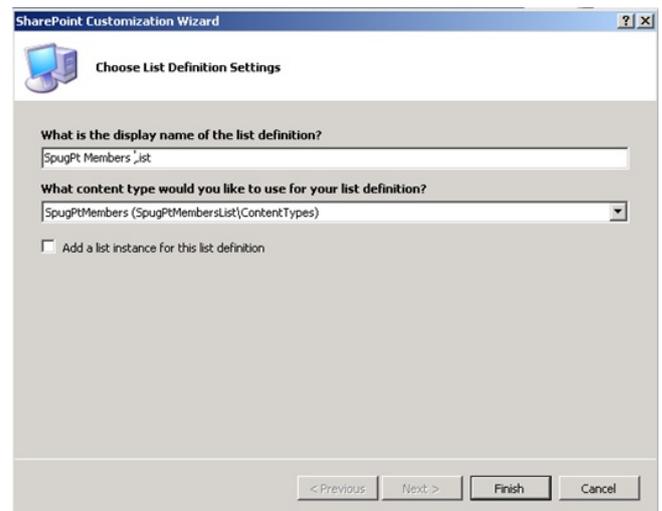
7. Vamos de seguida criar uma list definition de forma a provisionar uma definição de uma lista SharePoint baseada neste Content Type criado

Assim:

- Seleccionar o projecto com o botão direito do rato e seleccionar a opção Add>NewItem
- Seleccionar o item “List Definition from Content Type” e o digitar o nome “SpugPtMembersList” para este item.
- Seleccionar o botão “Add”



- No ecrã seguinte , digitar em “What is the display name...” o seguinte texto “SpugPt Members List”
- Deseleccionar a opção “add a list instance”
- Seleccionar o botão “Finish”



Mais uma vez, teremos que editar o conteúdo deste novo ficheiro Elements.xml criado , mas desta vez só precisamos alterar o campo Description.

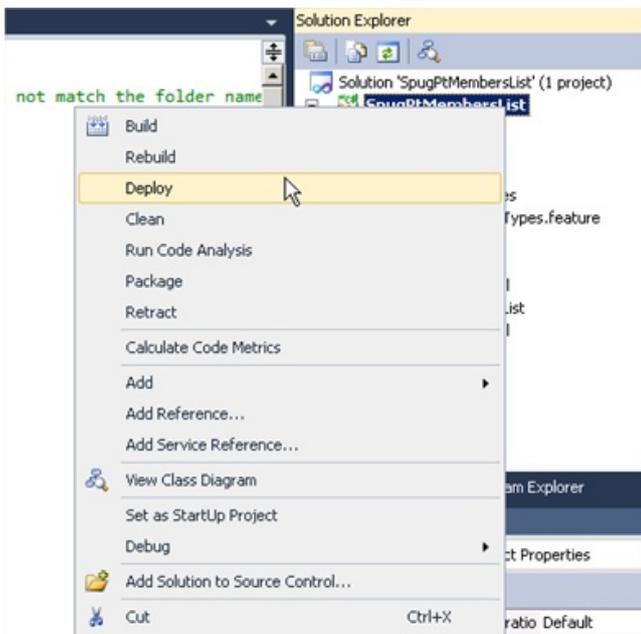
```
<?xml version="1.0" encoding="utf-8" ?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <!-- Do not change the value of the Name attribute below. If it does not matc -->
  <ListTemplate
    Name="SpugPtMembersList"
    Type="10000"
    BaseType="0"
    OnQuickLaunch="TRUE"
    SecurityBits="11"
    Sequence="410"
    DisplayName="SpugPt Members List"
    Description="SpugPt Members List"
    Image="/_layouts/images/itgen.png"/>
</Elements>
```

7.Finalmente, temos a solução completa pelo que só

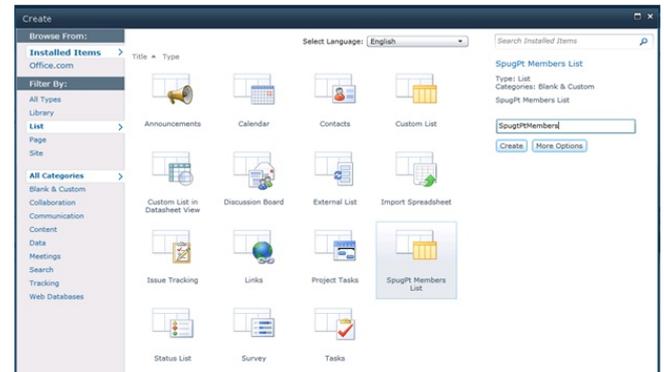
COMUNIDADE SHAREPOINTPT

Desenvolvimento em SharePoint 2010

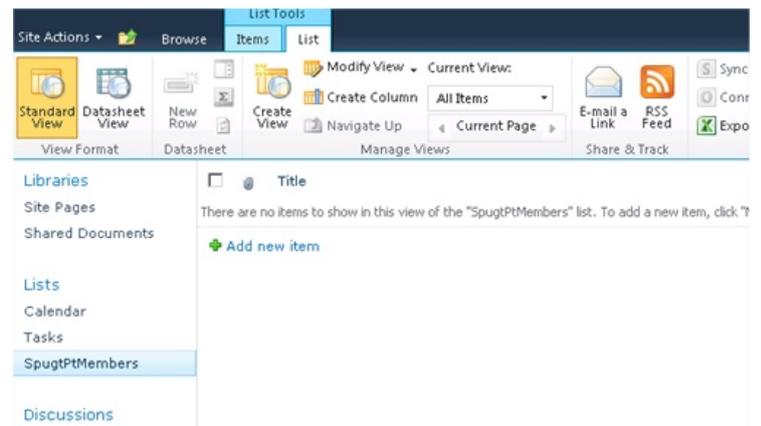
nos falta fazer a instalação no nosso SharePoint Como? Fácil.Primeiro executamos o Build da solução e de seguida o Deploy e a nossa solução fica instalada no SharePoint.



- Seleccionar a opção Lists> SpugPt Members List
- Digitar SpugPtMembers na caixa de texto
- Seleccionar o botão “Create”



Temos a nossa lista criada, resta-nos adicionar itens



Como teste final, vamos criar a nossa lista e criar alguns itens.

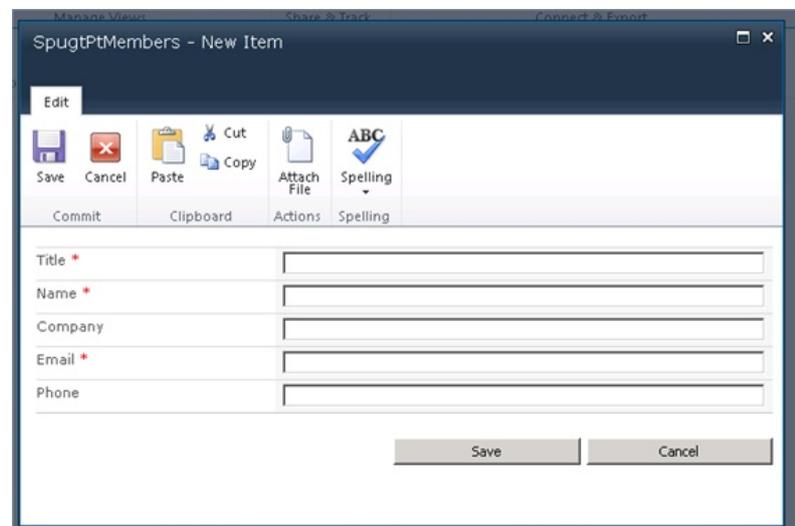
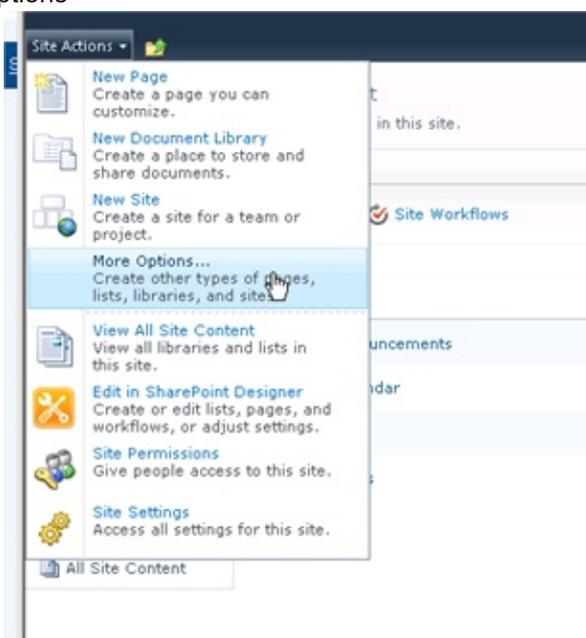
Para tal:

•Aceder ao site SharePoint, via browser , neste exemplo “http://vm2010”

•Seleccionar a opção Site Settings > More Options

•Seleccionar a opção “Add new item”

et Voilà!



COMUNIDADE SHAREPOINTPT

Desenvolvimento em SharePoint 2010

Conclusão

Embora tenha sido um exemplo muito simples, espero que tenha atingido o objectivo de como o desenvolvimento em SharePoint encontra-se extremamente facilitado comparativamente com as versões anteriores. É uma plataforma extremamente poderosa como ferramenta corporativa e como plataforma de desenvolvimento. Dentro deste tema existem claramente imensos subtópicos, integrações e tipo de implementação e neste momento é claro que este é um produto da Microsoft que irá evoluir e consolidar-se cada vez mais.

Resta-me fazer-vos um convite: todos os meses aos 2ª Sábados, a Comunidade Portuguesa de SharePoint apresenta 1 sessão e chalk talks com membros activos na tecnologia onde podemos trocar ideias, e esclarecer dúvidas e acima de tudo trocar experiências.

Apareçam, e espero que tenha sido útil.

Aquele abraço com muitos Dispostos à mistura

AUTOR



Escrito por **Rodrigo Pinto**

SharePoint Specialist, Evangelist, SharePoint Portuguese User Group (SPUGPT) Founder

Com 12 anos de experiência em Software Engineering & Architecture, é SharePoint Specialist, Evangelist na Indra. Tem experiência nas diversas áreas do SharePoint, destacando-se as áreas de object model, implementação de soluções, e Web Content Management com conteúdos media. Evangelista acérrimo de um roadmap de best practices na plataforma SharePoint assente em rigor e criatividade, procura disponibilizar estas provas de conceito pela comunidade, clientes e parceiros.

É o fundador da Comunidade Portuguesa de SharePoint.(www.sharepointpt.org)

GuiaTV CoolthingsPT

A aplicação GuiaTv é uma aplicação para dispositivos móveis com o sistema operativo Android que permite aos utilizadores visualizarem a programação dos canais de televisão de uma forma simples e ao mesmo tempo apelativa. Foi esta aplicação que garantiu à equipa CoolthingsPT o primeiro prémio do Concurso de Programação para Android (CPA) da comunidade Androidpt (<http://www.androidpt.com/>).



Esta aplicação foi a primeira aposta da startup nacional CoolthingsPT. Os objectivos da CoolthingsPT a curto prazo passam por fazer da CoolthingsPT uma marca com qualidade a nível nacional e internacional, tanto no desenvolvimento de aplicações Android como em serviços de consultoria nessa área.

O desenvolvimento da aplicação começou algum tempo antes do anúncio do 1º CPA (<http://concurso.androidpt.com/>).

Mas assim que soubemos da existência do concurso definimos logo essa data como a deadline para apresentar a versão beta da aplicação. Para conseguirmos cumprir os prazos por nós definidos, foi necessário cortar algumas funcionalidades mais complexas e diminuir a prioridade a algumas tarefas que poderiam ser mais motivantes. Tudo isto, para garantirmos que tínhamos tempo suficiente para criar todas as funcionalidades básicas da aplicação.



Ecrã Zapping

Método de Desenvolvimento

No desenvolvimento da aplicação, a equipa empregou métodos ágeis de desenvolvimento de software. Desses métodos destaca-se o scrum, onde se encaixam as reuniões em que cada membro basicamente comenta aquilo que fez na última sessão de desenvolvimento, o que pretende desenvolver nessa sessão de desenvolvimento e os entraves que prevê para essa mesma sessão. Esta

COMUNIDADE ANDROIDPT

GuiaTV CoolthingsPT

prática é muito interessante porque permite a todos os membros da equipa estarem actualizados sobre o desenvolvimento de todas as frentes de desenvolvimento da aplicação e terem em todas as sessões de desenvolvimento uma visão geral do estado de desenvolvimento da aplicação.

Para além destas reuniões, também é importante salientar uma prática que nós achamos importante na equipa CoolthingsPT que é o facto de programarmos em pares e discutirmos a melhor forma de desenvolver uma funcionalidade entre toda a equipa.



Ecrã Destaques

Estrutura da Aplicação

Uma das primeiras e principais dificuldades na implementação do GuiaTV foi definir claramente qual a estrutura da interface a utilizar. Depois de experimentar algumas aplicações já existentes, decidimos optar por utilizar uma interface com tabs recorrendo ao uso de "Tabhost"s. A TabHost é um tipo de ViewGroup disponível na API do Android. Fazendo uso destas TabHosts, é possível agrupar o conteúdo da aplicação em tabs dentro de uma TabActivity. Com este tipo de interface, conseguimos apresentar todas

a funcionalidades que pretendíamos no mesmo ecrã permitindo assim uma navegação mais fluida para o utilizador.

Todas as TabHosts do GuiaTV sofreram uma modificação visual na implementação da TabHost, pois esta não era suficientemente atractiva visualmente e não se adequava ao tema da aplicação. Estas alterações foram possíveis porque a plataforma android oferece total liberdade ao developer para alterar do modo pretendido os componentes existentes na API.

A visualização de todos os canais disponíveis da MEO e da ZON são apresentados ao utilizador dentro de uma "HorizontalScrollView". Esta view foi escolhida para apresentar os mais de 200 canais porque permite uma interacção rápida com os mesmos, o que permite uma alteração rápida de canal e é ao mesmo tempo visualmente atractiva e cativante.

Relativamente ao modo de obtenção da informação dos canais e respectiva programação, esta é efectuada por requisições "post", sendo a resposta obtida no formato "xml". Relativamente à obtenção da informação da MEO, foi utilizado o webservice disponibilizado pela sapo disponível no seguinte endereço:

<http://services.sapo.pt/Metadata/Service/EPG>

No que diz respeito aos destaques, é utilizado um webservice que recolhe os destaques diariamente, colocando a informação posteriormente numa "listView".

Para proceder ao parsing da informação obtida em xml é utilizada a API SAX(Simple API for XML) pois é um "parser" rápido e gasta relativamente menos memória quando comparado com DOM(Document Object Model). Isto faz com que seja adequado para um dispositivo móvel dada a memória ser mais limitada.

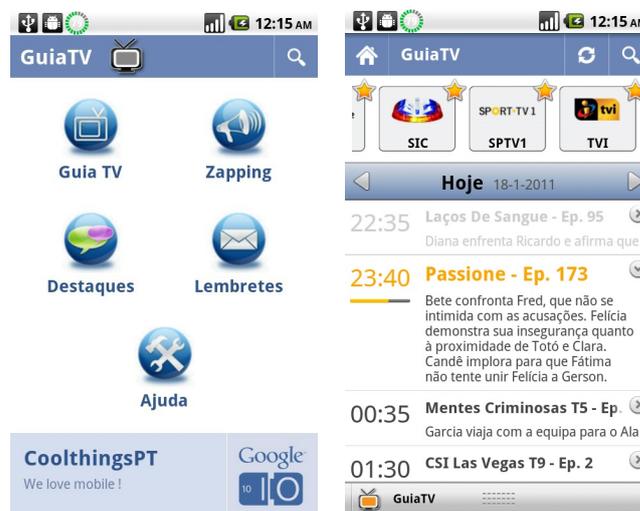
Um dos objectivos iniciais era permitir a utilização da aplicação no modo offline, para tal foi criado um serviço para obtenção da informação dos canais favoritos, guardando a mesma numa base de dados interna (sqlite). Sendo que os canais favoritos são definidos pelo utilizador, foi necessário encontrar uma forma simples, intuitiva e cativante para proceder a essa selecção. A funcionalidade de Zapping está directamente ligada aos favoritos, expondo a informação dos programas que estão a decorrer no momento nos canais favoritos, bem como a percentagem de programa já decorrida.

Tanto a lista que mostra a programação de cada canal como a lista de canais favoritos no Zapping é utilizado um "ListAdapter", que estende do "BaseAdapter" e que é modificada sempre que houver alterações, ou seja, quando se muda de canal ou quando um canal é adicionado ou removido dos favoritos. Esta é a forma mais funcional e correcta de apresentar informação numa aplicação android.

Sneak Peek

Actualmente estamos a preparar a próxima versão do GuiaTV CoolthingsPT que incluiu, por exemplo, pesquisa automática da box Meo para mudança de canais, "zapping" automático, menu de configurações, ordenação de canais, reestruturação da interface visual, entre outras.

De seguida apresentamos duas imagens da próxima versão da aplicação GuiaTV CoolthingsPT.



Ecrã Principal

Ecrã Guia TV

* Os icones apresentados são somente exemplos e não representam os finais.

Como se vê nas imagens acima apresentadas, a interface foi remodelada de acordo com as "guidelines" de construção de aplicações proposta pela Google. Mais informações sobre este assunto em:

<http://android-developers.blogspot.com/2010/05/twitter-for-android-closer-look-at.html>

AUTOR



Escrito por **CoolThingsPT**

Equipa constituída por: Bernardo Pina, Celso Cardoso, Claudia Peixoto, Diogo Júnior, Juarez Braga, Manuel Silva, Renato Oliveira. Este é o primeiro projecto conjunto desta equipa.

Como habilitar external users no SharePoint online – Office365

Office 365 será lançado em 2011, produto online com um poder de produtividade e maturidade já visto em relação ao BPOS (<http://www.microsoft.com/online/>), dentre os recursos que foram anunciados e já testados por alguns no beta, vou apresentar o recurso de external users no SharePoint Online. No SharePoint online disponível no BPOS (versão 2007 do SharePoint) não era possível acesso de usuários externos.

Uma característica interessante e que é importante para garantir uma colaboração rica e completa com todos os colaboradores, parceiros e até clientes, é a possibilidade de compartilhar um site de documentos através do SharePoint Online. Agora no SharePoint Online que será lançado no Office365, que já vem com a versão 2010 do SharePoint isso será possível.

Detalhes do Office365 visite:

<http://www.microsoft.com/pt-br/cloud/>

<http://office365.microsoft.com/en-US/>

Nesse artigo pretendo em primeira mão mostrar como é fácil habilitar e configurar para acesso de usuários externos usando o Windows Live ID para acessar informações do seu site SharePoint.

Primeiro passo é registrar na secção de administração do Office365:

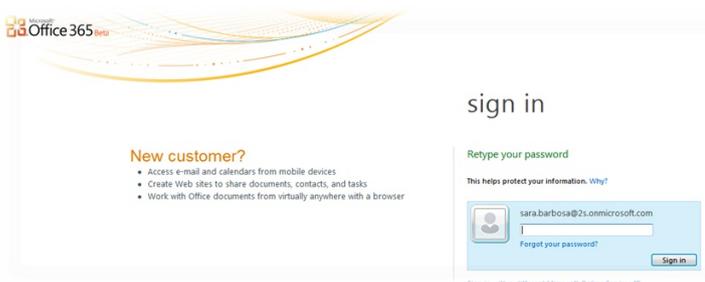


Figura 1 - Página de login no Office365

Em admin, na lista de serviços disponíveis, seleccione a opção “manage” no SharePoint Online:

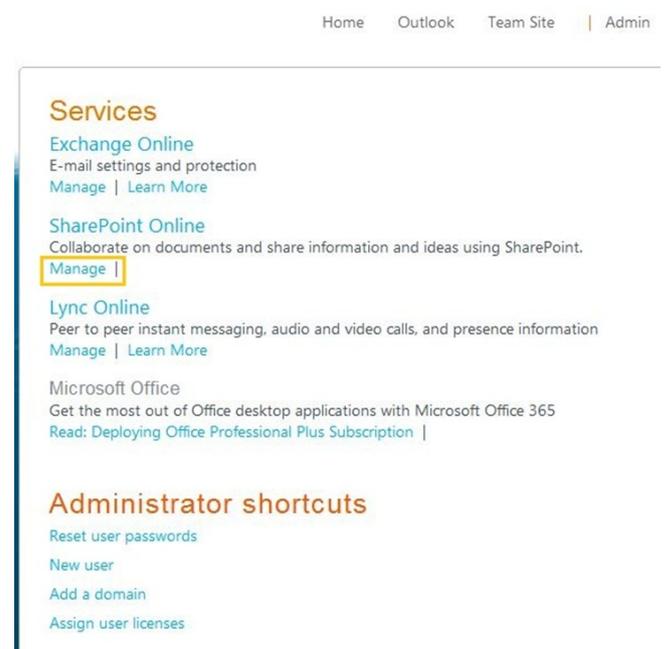


Figura 2 - Serviços disponíveis

Em manage, você será direccionado ao Administration Center do SharePoint Online, dentre as opções de configurações, vá em “manage site collections”:



Figura 3 - Administration Center SharePoint

Vá em “Site Collections”, veja que nos sites abaixo a opção “External Users” está deny. Isso significa que nem dos sites está habilitado para acesso para

COMUNIDADE CODIFICANDO.NET

Como habilitar external users no SharePoint online - Office365

acesso de usuários externos.

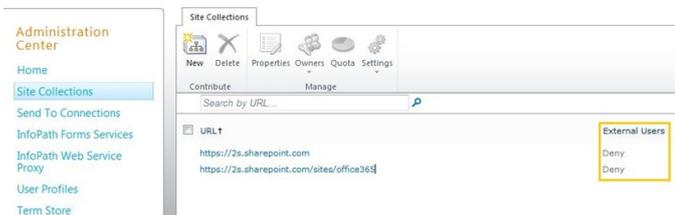


Figura 4 - Sites Collections

Escolha o site que deseja habilitar o acesso externo. Selecciono o site, clique em 'settings', 'manage external users', escolha opção 'allow' e confirme o procedimento.

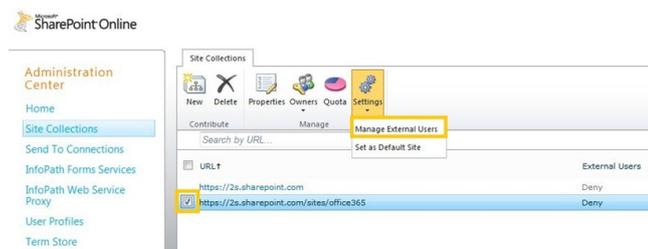


Figura 5 - Habilitando external users

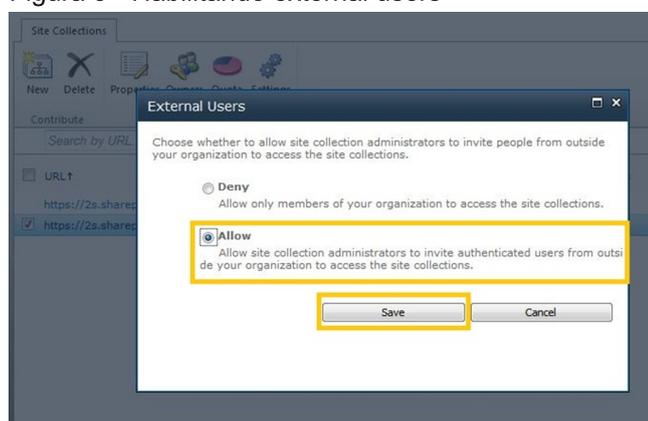


Figura 6 - Salvando alteração

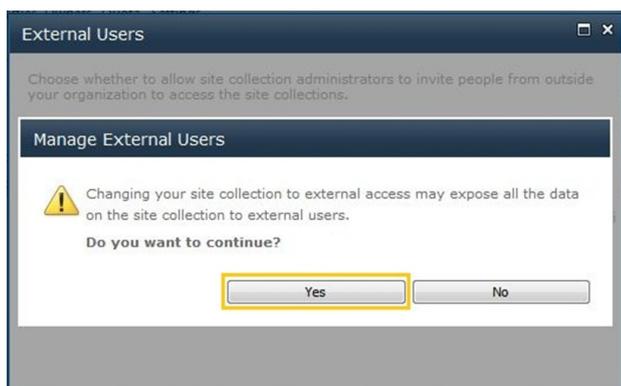


Figura 7 - Confirmando liberação para external users

Repare que imediatamente já foi alterado de deny para allow na opção de external users.

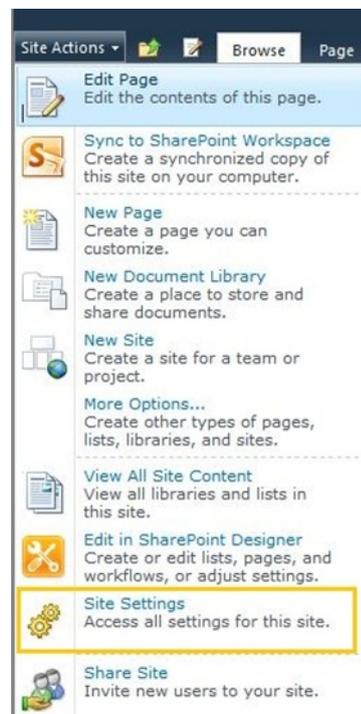


Figura 8 - Alteração concluída

Depois de habilitado você vai criar o utilizador no seu site. A criação desse utilizador funciona da seguinte forma: você adiciona o utilizador através do e-mail que será usado para acessar o site, escolhe o grupo na qual o utilizador deve pertencer e confirma. Quando confirmar, o Office365 envia um e-mail para o utilizador informando que foi convidado para acessar o site e deve aceitar o invite.

Para adicionar o utilizador, entre no site, exemplo no meu caso: '<https://2s.sharepoint.com/sites/office365>', clique em 'sites actions', 'site settings':

Figura 9 - Site Actions



Clique em 'external users permissions':

COMUNIDADE CODIFICANDO.NET

Como habilitar external users no SharePoint online – Office365

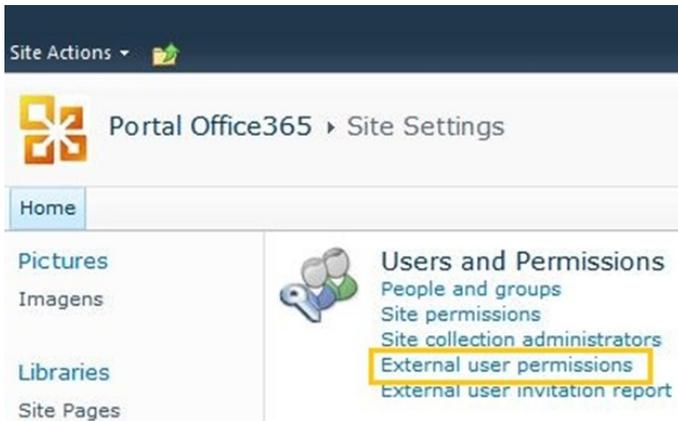


Figura 10 - Users and Permissions

Adicione o e-mail, permissão (é possível atribuir o utilizador a um grupo específico, ficando mais fácil controlar as permissões dos utilizadores dentro do seu site e garantido a segurança das informações) e mensagem personalizada que vai para o convidado externo:

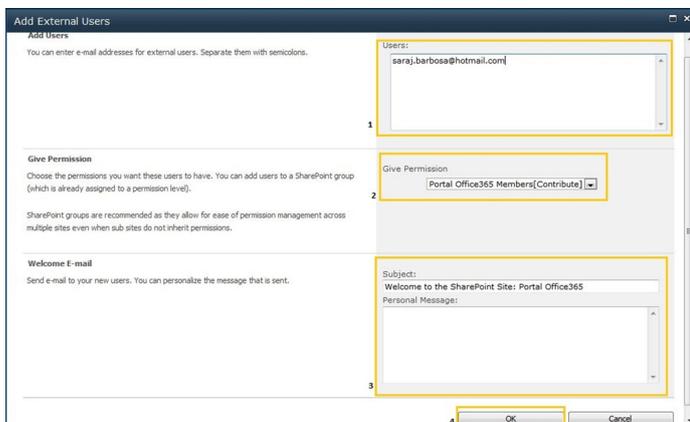


Figura 11 - Add External Users

Nota: Caso o usuário não aceite o convite que recebeu por e-mail, ele não será adicionado como membro do site no SharePoint e também terá acesso negado no site em questão.

Processo de convite foi concluído, agora falta o utilizador aceitar o convite para liberação no site. Processo simples, basta o utilizador verificar o e-mail que recebeu e aceitar o convite clicando no campo 'accept your invitation'. Neste artigo o e-mail foi

enviado para um endereço do hotmail, então fiz o acesso ao hotmail e já consegui visualizar o e-mail recebido.



Figura 12 - Exemplo de e-mail

Após aceitar o convite o utilizador já será direcionado a pagina de autenticação do Online Services. No momento que o utilizador digitar o e-mail (no caso hotmail – Live ID) e clicar no campo senha o campo fica isolado para digitar a senha e automaticamente, aparece uma mensagem informando para fazer o sign in directo da página do hotmail.

Basta clicar em 'sign in at hotmail' e seguir em frente com o login.

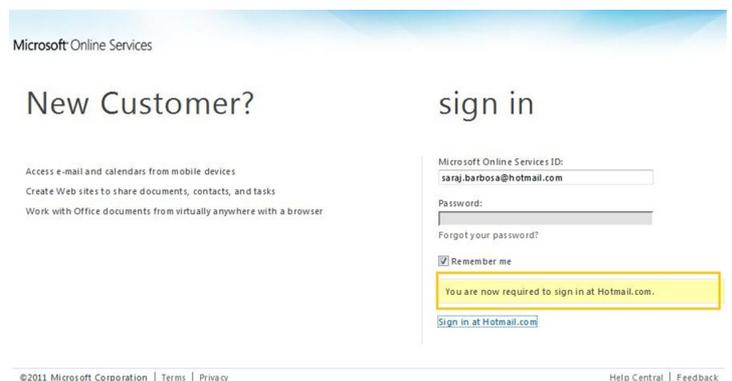


Figura 13 - Sign in

Depois que ocorrer autenticação e se tudo estiver OK, você será direcionado para home do site SharePoint. Repare o utilizador hotmail registado no site:

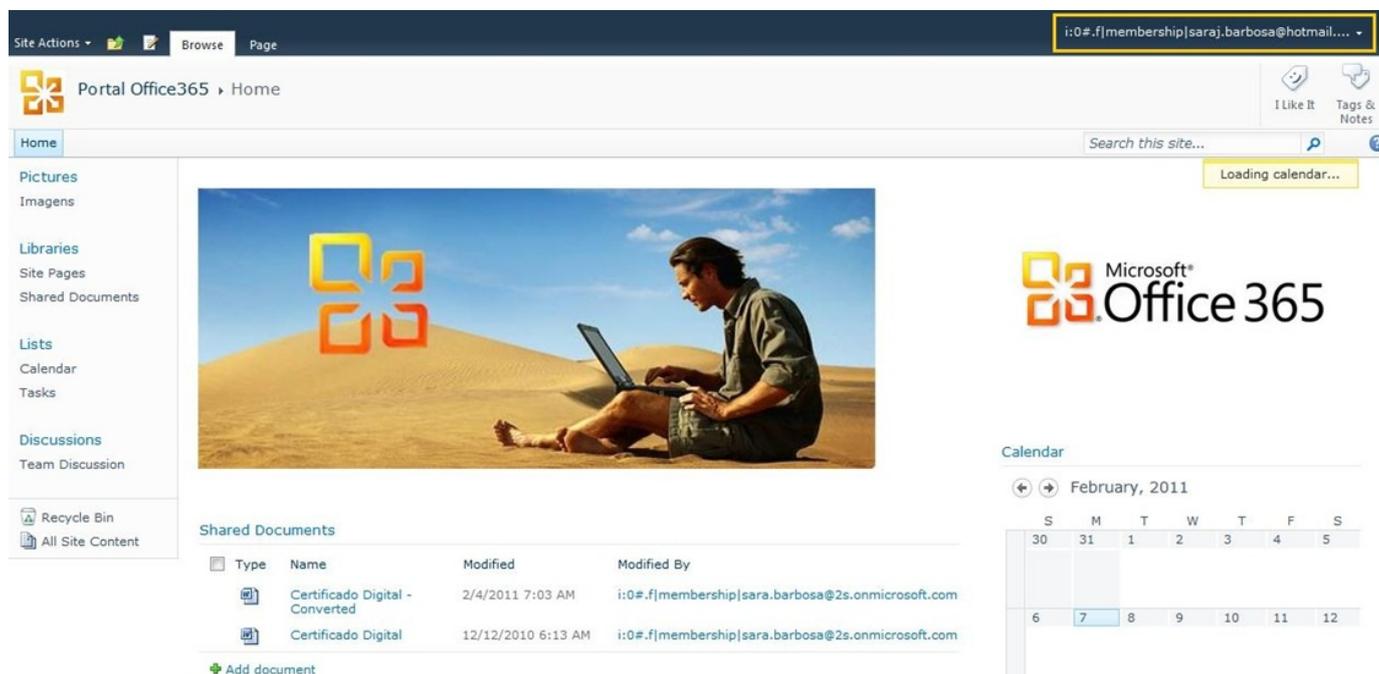


Figura 14 - Home do site SharePoint

Agora sim, você já pode usufruir do external users e facilitar a partilha de documentos e aumentar a produtividade da sua empresa.

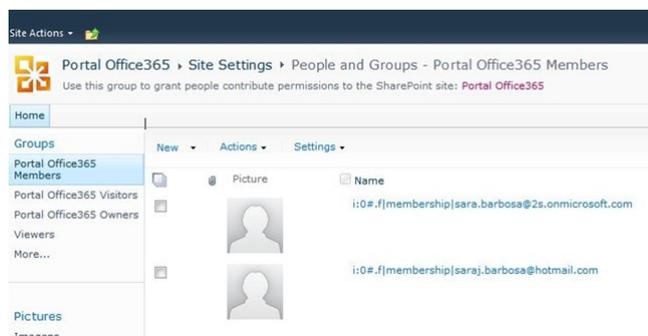


Figura 15 - Visualizar pessoas e grupos

Um detalhe, caso queria excluir, editar e visualizar os utilizadores, basta ir no SharePoint > site actions > site settings e clicar em people and groups, você terá lista de utilizadores do seu site.

Neste passo a passo foi possível visualizar a maneira fácil e rápida de adicionar utilizadores para usufruir de conteúdos que hoje talvez você estaria enviando por e-mail para compartilhar com seus clientes, parceiros e que no SharePoint Online do Office365 será simplificado, poupando tempo e agilizando os processos.

Agora é aguardar que a versão fique disponível para todos testarem e usufruírem dos benefícios das versões 2010 dos serviços online.

AUTOR



Escrito por **Sara Barbosa**

MSP Microsoft. Faz parte da comunidade WININFO. Co-Leader da comunidade AzureServicesBr. Revê artigos para revista online Codificando.NET e-Magazine. Entusiasta Microsoft Online Services [BPOS] e [Office365] e actualmente faz parte da equipa Pensou.NET empresa Gold Certified Partner em Goiás. Mantém o seu blog: <http://sarajbarbosa.wordpress.com> e twitter: [@sarabarbosa](https://twitter.com/sarabarbosa).

COMUNIDADE PONTONETPT

O Futuro da Programação Assíncrona da Plataforma .NET

Introdução

Existe alguma confusão entre execução paralela, concorrente e assíncrona.

Execução paralela significa que vários troços de código são executados ao mesmo tempo (em paralelo).

Execução concorrente significa que vários troços de código podem aceder aos mesmos dados, não necessariamente ao mesmo tempo, mas durante uma operação (que pode ser interrompida).

Execução assíncrona significa que o código é executado fora da linha lógica do código actual.

Os três modos de execução acima referidos podem ocorrer isoladamente ou em combinação. Vários troços de código podem estar em execução simultaneamente sem que isso signifique concorrência se não acederem aos mesmos recursos. Uma operação assíncrona pode ser executada sincronamente.

Assynchronous Programming Model (APM)

Até à versão 4.0 da plataforma .NET a norma para operações assíncronas era o Assynchronous Programming Model (APM) a) que consiste em dois métodos que chamados BeginOperation e EndOperation.

Este modelo é seguido pela quase totalidade das classes da Base Class Library (BCL), como, por exemplo, as streams. b)

Os métodos anónimos introduzidos no C# 2.0 ou as expressões Lambda introduzidas no C# 3.0 podem

tornar o código mais legível, mas não deixa de ser difícil perceber a intenção do código:

```
var wr = WebRequest.Create(url);
wr.BeginGetResponse(
    ar =>
    {
        var rs = wr.EndGetResponse(ar)
            .GetResponseStream();
        var sr = new StreamReader(rs);
        Console.WriteLine(
            sr.ReadToEnd());
    },
    null);
```

Event-based Asynchronous Pattern (EAP)

A partir da versão 2.0 começaram a surgir algumas implementações da Event-based Asynchronous Pattern (EAP) c) que consiste num método OperationAsync e um evento OperationCompleted.

Um exemplo da utilização deste padrão é a classe WebClient.

Mais uma vez, usando métodos anónimos ou expressões Lambda talvez até se perceba a intenção do código:

```
var wc = new WebClient();
wc.DownloadProgressChanged +=
    (dps, dpe) =>
    {
        Console.Write(
            "\r{0}%. {1} of {2}",
            dpe.ProgressPercentage,
            dpe.BytesReceived,
            dpe.TotalBytesToReceive);
    };
```

```
wc.DownloadStringCompleted +=  
    (dss, dse) =>  
    {  
        Console.Write('\r');  
        Console.WriteLine(dse.Result);  
    };  
  
wc.DownloadStringAsync(uri);
```

Task-based Asynchronous Pattern (TAP)

A plataforma 4.0 introduziu uma nova forma de programação assíncrona baseada em tarefas que faz uso da Task Parallel Library (TPL). d)

Esta biblioteca permite a execução de tarefas assíncronas mesmo que as classes usadas não tenham uma API assíncrona, assim como definir a tarefa a executar após a conclusão de uma tarefa (continuação) e permite o cancelamento da execução a tarefa.

Esta API por ter surgido após o C# 3.0 foi desenhada já com vista à utilização de expressões Lambda:

```
Task.Factory.StartNew(  
    () =>  
    {  
        return new WebClient()  
            .DownloadString(uri);  
    }).ContinueWith(  
    task =>  
    {  
        Console.WriteLine(task.Result);  
    }  
);
```

Aqui torna-se mais claro que a intenção do código é descarregar um texto e, no final, enviá-lo para a consola.

Caso fizesse sentido cancelar a tarefa em execução,

poderia ser passado à factory um token de cancelamento (Cancellation token). e)

O Futuro Da Programação Assíncrona Na Plataforma .NET

A próxima versão da plataforma .NET vai trazer, não um novo padrão de execução ou uma nova biblioteca, mas uma nova forma de escrever código assíncrono.

Assentando na TPL, a nova forma de escrever código introduz duas novas palavras chave: `async` (Async em Visual Basic) e `await` (Await em Visual Basic).

A palavra chave `async` é aplicada aos métodos e indica ao compilador que deve reescrever o método de forma assíncrona.

A palavra chave `await` identifica, dentro de um método, a execução assíncrona de um método.

Com este novo modelo de programação assíncrona vai ser possível escrever código como este:

```
public static async Task<byte[]>  
DownloadDataAsync(  
    string url,  
    CancellationToken cancellationToken,  
    IProgress<long> progress)  
{  
    using(var request =  
        WebRequest.Create(url))  
        using(var response = await  
            request.GetResponseAsync())  
            using(var responseStream =  
                response.GetResponseStream())  
                using(var result = new MemoryStream())  
                {  
                    await  
                        responseStream.CopyToAsync(result,  
                            cancellationToken, progress);  
                    return result.ToArray();  
                }  
}
```

COMUNIDADE PONTONETPT

O Futuro da Programação Assíncrona da Plataforma .NET

Que será usado do seguinte modo:

```
byte[] result = await DownloadDataAsync(  
    url,  
    cancellationToken,  
    progress);
```

As equipas dos respectivos compiladores ainda têm alguns problemas para resolver, mas não fugirá muito disto e existe já uma CTP de uma primeira implementação.

Recursos

- Asynchronous Programming Design Patterns
(<http://msdn.microsoft.com/library/ms228969.aspx>)

- Asynchronous Programming Overview
(<http://msdn.microsoft.com/library/ms228963.aspx>)

- Event-based Asynchronous Pattern Overview
(<http://msdn.microsoft.com/library/wewwczdw.aspx>)

- Visual Studio Async CTP
(<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=18712f38-fcd2-4e9f-9028-8373dc5732b2>)
Extensão ao Visual Studio 2010 (não compatível com o Visual Studio 2010 SP1) que providencia as novas funcionalidades de programação assíncrona.

- Parallel Programming with .NET

(<http://blogs.msdn.com/b/pfxteam/archive/tags/async/>)
Blogue da equipa de Parallel Programming.

- Lucian's VBlog
(<http://blogs.msdn.com/b/lucian/archive/tags/async/>)
Blogue de Lucian Wischik – Program Manager da linguagem Visual Basic.

- Fabulous Adventures In Coding
(<http://blogs.msdn.com/b/ericlippert/>)
Blogue de Eric Lippert – Programador na equipa do compilador de C#.

- Task-based Asynchronous Pattern
(<http://www.microsoft.com/downloads/details.aspx?FamilyID=4969771d-9855-4421-baa0-c91848db4ab0>)
Artigo do Lucian Wischik sobre a Task-based Asynchronous Pattern e a sua utilização nas novas funcionalidades das linguagens C# e Visual Basic.

Hiperligações:

a) <http://msdn.microsoft.com/library/ms228963.aspx>
b)

<http://msdn.microsoft.com/library/System.IO.Stream.aspx>

c) <http://msdn.microsoft.com/library/wewwczdw.aspx>

d) <http://msdn.microsoft.com/library/dd460717.aspx>

e)

<http://msdn.microsoft.com/library/system.threading.cancellationtoken.aspx>

AUTOR



Escrito por **Paulo Morgado**

É licenciado em Engenharia Electrónica e Telecomunicações (Sistemas Digitais) pelo Instituto Superior de Engenharia de Lisboa e Licenciado em Engenharia Informática pela Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa. Pelo seu contributo para a comunidade de desenvolvimento em .NET em língua Portuguesa, a Microsoft premeia-o com o prémio MVP (C#) desde 2003. É ainda co-autor do livro "LINQ Com C#" da FCA.

ANÁLISES

Silverlight 4.0 – Curso Completo

Silverlight 4.0 – Curso Completo

O livro Silverlight 4.0 – Curso Completo, de autoria de Luís Abreu e colaboração de João Carreiro, é um livro lançado pela FCA – Editora de Informática, com 400 páginas e dividido em 15 capítulos. É apresentado com o objectivo de “ensinar o programador que se está a iniciar na plataforma Silverlight, sendo também uma ferramenta indispensável para o programador conhecedor da plataforma Silverlight que pretende fazer a transição para a nova versão”.

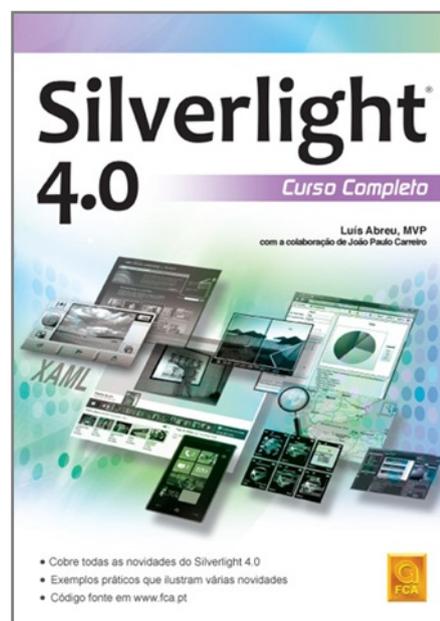
O livro começa por apresentar os requisitos, o público-alvo, convenções, código fonte e organização do mesmo, fazendo um pequeno resumo do que será abordado em cada capítulo.

De forma breve é apresentado no primeiro capítulo as principais funcionalidades do Silverlight 4.0 e é feito um enquadramento da tecnologia a nível histórico, a nível de arquitectura e a nível de desenvolvimento, mostrando interacção com o Visual Studio 2010. Não ficando de fora uma abordagem sobre Assemblies e Pacotes XAP e a integração com páginas Web.

“ É apresentado com o objectivo de “ensinar o programador que se está a iniciar na plataforma Silverlight” ”

O segundo capítulo aborda os conceitos base, que servem de suporte ao desenvolvimento de aplicações Silverlight, de destacar o conceito XAML, dependency properties e attached properties, routed events e comandos, incluindo implementação prática destes conceitos.

Nos dois capítulos seguintes, é feita abordagem sobre o Layout e Controlos. Em Layout, a abordagem é mais direccionada para o processo de renderização



e apresentação dos diversos controlos layout (Canvas, InkPresenter, StackPanel e Grid), não ficando de parte o conceito de painéis virtuais e painéis personalizados. O capítulo de Controlos inicia com apresentação da hierarquia de controlos e em seguida é abordado os conceitos content control, items control, selectores, range controls, controlos de texto, sendo apresentado exemplos concretos para cada conceito. Por fim é apresentado outros controlos generalistas, é o caso da DataGrid e Chart.

No quarto capítulo, Recursos, é feita uma abordagem sobre recursos binários, onde é exposta a forma como pode ser feito o acesso aos recursos, localização de aplicações e por fim é feita a apresentação de recursos XAML, sendo abordado os conceitos ResourceDictionary, MergedDictionary e StaticResource.

Personalização de Controlos, é o quinto capítulo, onde é abordado os conceitos relacionados com estilos, partilha de estilos, herança de estilos, propagação de estilos, temas, templates e estados. Em cada um dos conceitos é apresentado exemplos

que clarificam os conceitos teóricos.

Networking apresenta as classes introduzidas pela plataforma que permitem a obtenção remota de dados. Este capítulo abrange muitos conceitos da plataforma .Net e seria uma mais-valia ter sido mencionadas referências de leitura opcional.

DataBinding, é o sexto capítulo, a introdução do conceito data binding e a sua ilustração é uma mais-valia deste capítulo. É apresentada a forma como é feita a propagação automática de alterações, usando a interface INotifyPropertyChanged, os vários modos de binding, conversores e utilização de bindings em XAML. A validação de dados, faz parte deste capítulo, sendo feito comparação entre a implementação da interface IDataErrorInfo e INotifyDataErrorInfo. O capítulo termina com a exemplificação de binding em colecções, cenários de master-detail e paginação, ordenação, agrupamento e filtragem.

No capítulo Gráficos e outros elementos visuais, é feito abordagem de dos vários objectos que permitem definir o aspecto visual dos user controls de Silverlight, começando por apresentar geometrias, shapes, brushes, efeitos (zooming e panning), imagens, fontes e perspectivas 3D.

O décimo capítulo, Animações, apresenta as várias animações que estão disponíveis em Silverlight, sendo apresentado em seguida alguns exemplos práticos. Faz referência a animações com Keyframes e Frame-a-frame e faz ainda algumas considerações sobre performance. No final faz uma pequena referência a ferramentas gráficas, nomeadamente o Expression Blend, no entanto esta ferramenta está fora de âmbito do livro.

Média é o capítulo que apresenta os conceitos relacionadas com ficheiros de multimédia, sendo os mais relevantes MediaElement e MediaPlayer. É feita uma abordagem dos eventos mais importante, produção de vídeo, dowloand progressivo vs streaming, utilização de playlist e marcadores,

construção de um player e apresentação do elemento VideoBrush. Por fim, como seria de esperar, é apresentada a interação com webcam e microfone.

Silverlight e o Browser, é o décimo segundo capítulo, que apresenta a forma como é feita a integração do Silverlight com uma página de HTML, como é controlada a interação entre os restantes componentes existentes numa página e o código de um user control de Silverlight.

“ De salientar que ao longo da leitura, deparamo-nos com o recurso de diagramas para mostrar a relação entre classes; ”

O décimo terceiro capítulo, Outros Serviços, é um capítulo onde foi colocado todos os conceitos que não se enquadravam nos outros capítulos, são eles: impressão (PrintDocument, eventos associado, personalização, paginação e feedback de impressão), menu de contexto personalizado, Isolated Storage (gestão de ficheiros, escrita e leitura de ficheiro, aumento de quota), acesso ao clipboard, drag-n-drop, navegação entre páginas e o controlo pivotviewer.

Aplicação Out of Browser, é um dos capítulos muito relevantes deste livro e nele apresentado algumas considerações importantes, configurações e desinstalação de aplicações, detecção do host de alojamento, actualização de aplicações instaladas, verificação do estado de ligação à rede, modo full trust entre outras funcionalidades específicas. Por fim termina com Debugging de aplicações out of the browser sendo feita uma abordagem com o Visual Studio.

No último capítulo, Segurança aborda o modelo de segurança aplicado ao código de Silverlight, explicando o porquê de muitas restrições impostas. Terminando com apresentação de alguns cuidados e

ANÁLISE

Silverlight 4.0 – Curso Completo

considerações a ter em conta no desenvolvimento.

De salientar que ao longo da leitura, deparamo-nos com o recurso de diagramas para mostrar a relação entre classes; uso de imagens de apoio à compreensão de conceitos; destaque e notas de aspectos mais ou menos relevantes, orientando o leitor a poder complementar a sua leitura com outros recursos; apresentação de exemplos práticos e em alguns casos é apresentada a implementação de conceitos.

Um aspecto que pode ser melhorado é na apresentação dos exemplos, especificar o respectivo

projecto no conjunto de projectos do código fonte do livro.

Em conclusão, este livro aborda os conceitos de Silverlight 4.0, está bem organizado em termos de conteúdo, é de fácil leitura e é um bom guia para quem vai iniciar ou para quem já tem alguns conhecimentos nesta tecnologia.

AUTOR



Escrito por **Sara Silva**

É licenciada em Matemática – Especialidade em Computação, pela Universidade de Coimbra, actualmente é Software Developer no Porto.

O entusiasmo pela área resultou na obtenção dos títulos de Microsoft Certified Professional Developer – Windows 3.5, Microsoft Certified Technology Specialist – WPF 3.5, WPF 4 e Windows Forms. Faz parte de várias comunidades, tendo uma participação activa na Comunidade NetPonto e no P@P.

DUVIDAS?

IDEIAS?

AJUDAS?

PROJECTOS?



portugal-a-programar
•org

