

PROGRAMAR

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

EDIÇÃO #36 - AGOSTO 2012

ISSN 1647-0710



NOVIDADES DO VISUAL STUDIO 2012 VISUAL BASIC

REVIEWS DE LIVROS

DO MAC OS X **FUNDAMENTAL**

A PROGRAMAR

HTML5 FORMULÁRIOS
HORAS

ARDUINO MEDIÇÃO DE
TEMPERATURA

PASCAL CONSTRUÇÃO DE UMA
CALCULADORA GRÁFICA

SEO PARTE
II

WINDOWS AZURE IMPLEMENTAÇÃO DA CIFRA ARSD EM
AMBIENTE EMULADO WINDOWS AZURE

COMUNIDADES

PTCORESEC SEGURANÇA NA WEB
PARTE III

INTEGRATION SERVICES:
AN EXPERT COOKBOOK

MICROSOFT SQL SERVER 2012

NO CODE

DA EDUCAÇÃO
SEIS ANOS DEPOIS **PLANO TECNOLÓGICO**

NA COMUNIDADE P@P
NOTIFICATION AGENDA **PROJECTO EM DESTAQUE**

COMO UMA OPÇÃO DE
AUTO-EMPREGO E INCONFORMISMO **EMPREENDEDORISMO**

COLUNAS

'AVALIAÇÃO DA LISTA
DE ARGUMENTOS' **ENIGMAS DE C#**

CONTINUAR A ACREDITAR
NAS TIC? **CORE DUMP**

EQUIPA PROGRAMAR

Coordenador
António Santos

Editor
António Santos

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

Redacção
André Lourenço
António Santos
Fernando Martins
Igor Nunes
Miguel Lobato
Miguel Oliveira
Nelson Lima
Nuno Santos
Paulo Morgado
Ricardo Amaral
Ricardo Castro
Rita Peres
Sérgio Ribeiro
Tiago Henriques

Staff
António Santos
Fábio Domingos
Fernando Martins
Gil Sousa
Jorge Paulino
Sara Santos

Contacto
revistaprogramar@portugal-a-programar.org

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

ISSN
1 647-071 0

Em movimento

Vivemos num mundo em movimento, onde tudo está em constante “evolução”, onde o que “hoje é um facto, amanhã é um mito”, onde a tecnologia muda e transforma-se constantemente!

Ainda há pouco tempo me ri um pedaço ao ler o que escreviam críticos sobre uma apresentação de um sistema operativo que por incidente, crashou. Diziam os críticos “é mais um sistema sem qualidade”, outros questionavam em tom de ironia “será isto mais uma manobra de marketing?”, outros sem qualquer conhecimento apenas diziam, “já é habitual”.

Ora pensando bem, há bastantes anos atrás, os sistemas operativos crashavam, e pouco se dizia. O Disk Operating System, nem sequer funcionava em “protected mode”, mas não se dava a importância que agora se dá a essa funcionalidade. Há 10 anos atrás os smartphones não tinham um decimo das funcionalidades dos modelos de entrada de gama actuais, mas na altura maravilhavam muita gente e eram alvo de criticas severas de quem dizia que eles nunca iriam “singrar” no mercado das Tecnologias de Informação. Parece que o tempo provou que todas as criticas de então estavam erradas!

Num mundo em constante mudança e evolução, a única certeza que havia é que os programadores continuariam a evoluir, a tecnologia também, algumas tecnologias iriam vencer, outras não! A evolução traçaria a “linha” que separaria aquilo que iria continuar e aquilo que iria “ficar pelo caminho”.

Há 10 anos atrás, muita gente nem sequer sabia que existiam processadores ARM, hoje toda a gente os usa nos tablets, smartphones, etc... A mudança e a evolução assim traçaram o rumo!

Há 30 anos atrás, programava-se em Basic, os PC's tinham 1MB de Ram, os Mac's eram um pequeno nicho, e não se liam grandes criticas aos produtos tecnológicos. Há 30 anos atrás o PC foi considerado a “pessoa do ano” pela revista Times e nos dias que correm, nós vivemos todo o dia agarrados a gadgets várias vezes mais capazes que o PC de então, igualmente programados por Humanos, feitos e desenhados por Humanos, e com uma mesma coisa em comum! Os que hoje são de topo amanhã estarão obsoletos.

Uma grande parte dos programadores formados agora não sabe sequer que existiu uma era em que não existiam GUI's, em que as IDE's não eram tão “bonitas e apelativas”, eram mais um editor de texto em modo de texto, e se programava com as preocupações de manter o código “limpo”, organizado, a economizar recursos de sistema. Nos dias de hoje, mais do que nunca existe tendência a escrever “spaghetti-code” sem grande qualidade, consumidor de enormes recursos de hardware, muitas vezes “usando e abusando” de código gerado automaticamente pelas IDE's, etc...

Parece que todas as lições aprendidas nos anos anteriores, foram esquecidas, como se fosse esquecida uma parte da história e da evolução, da tecnologia. Felizmente ainda existe muita gente interessada em fazer um bom trabalho e não apenas em “produzir trabalho”, a tecnologia continuará a evoluir e o mundo a girar!

António Santos
<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

[6](#) **As Novidades do Visual Studio 2012 RC (Sérgio Ribeiro)**

Conheça as novidades da última versão deste IDE.

A PROGRAMAR

[14](#) **Pascal - Construção de uma Calculadora Gráfica (Igor Nunes)**

Veja como construir uma calculadora Gráfica em Object-Pascal

[28](#) **Formulários em HTML5 (Nelson Belo Lima)**

Aprenda a criar formulários em HTML5 com validação em Javascript

[34](#) **SEO: Search Engine Optimization - Introdução Parte II (Miguel Lobato)**

A segunda parte do intrigante artigo sobre Search Engine Optimization, do mesmo autor

[39](#) **Implementação da Cifra ARSD em Ambiente Emulado Windows Azure (Rita Peres e André Lourenço)**

Veja como implementar uma cifra em ambiente Windows Azure

[50](#) **Estação de Controlo de Temperaturas Usando Arduino e Texas Instruments LM335A (Nuno Santos)**

Aprenda a construir uma estação de controlo de temperaturas com a famosa board, Arduino

COLUNAS

[56](#) **Enigmas de C#: Avaliação da Lista de Argumentos (Paulo Morgado)**

Mais um enigma da linguagem de programação C#, desta vez envolvendo Lista de Argumentos.

[57](#) **Core Dump: Podemos Continuar a Acreditar nas TIC? (Fernando Martins)**

Será que os podemos continuar a ter acreditar nas Tecnologias de Informação hoje em dia? Conheça a opinião deste autor.

ANÁLISES

[60](#) **Fundamental do MAC OS X (Miguel Oliveira)**

[61](#) **Microsoft SQL Server 2012 Integration Services: an Expert Cookbook (António Santos)**

COMUNIDADES

[63](#) **PtCoreSec - Segurança na WEB (Parte 3) (Tiago Henriques)**

A terceira parte do artigo sobre segurança na WEB, do mesmo autor da parte II.

NO CODE

[69](#) **Plano Tecnológico da Educação: Seis Anos Depois (Ricardo Castro)**

[78](#) **O Empreendedorismo Como Uma Opção de Auto-emprego e Inconformismo (Marisa Almeida)**

[81](#) **Projeto em Destaque na Comunidade Portugal-a-Programar: Notification Agenda**

EVENTOS

25 Ago 2012	32ª Reunião Presencial da Comunidade NetPonto em Lisboa
13 Out 2012	UploadLisboa 2012
15-17 Nov 2012	Sapo CodeBits VI

Para mais informações/eventos: http://bit.ly/PAP_Eventos

Investigadores portugueses ganham prémio da Google

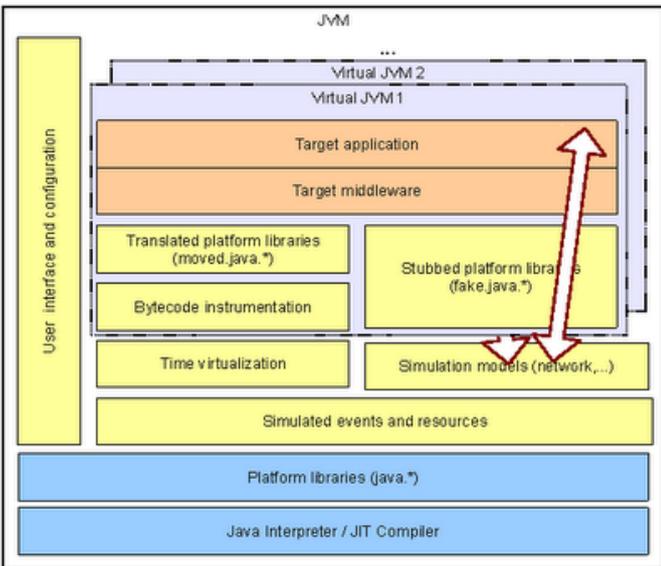
Investigadores da Universidade do Minho e do Inesc Tec ganharam o Google Research Award pelo trabalho que têm levado a cabo na área do cloud computing. O prémio da Google pretende distinguir o contributo dos investigadores do HASLab - Laboratório de Software Confiável (que tem investigadores da Universidade do Minho e do Inesc Tec) para o desenvolvimento de ferramentas que facilitam a produção de software para ambientes de Cloud Computing.

De acordo com um comunicado da Universidade do Minho, o prémio foi atribuído ao projeto Minha, por ter criado plataformas que permitem testar o desempenho das pesquisas do Google e do portal de vídeo YouTube a partir de um único computador.

Através da tecnologia Java, os investigadores nortenhos desenvolveram, no âmbito do projeto Minha, uma plataforma que simula a atividade de vários computadores e permite apurar em que condições funcionam algumas das principais redes e serviços que operam na Internet.

A solução permite fazer testes e observar os resultados na própria hora, e está também preparada para ensaios em sistemas que ainda não estão completos, permitindo analisar eventuais lacunas antes do lançamento comercial de um serviço.

No site www.minha.pt, pode descobrir mais detalhes sobre este projeto que mereceu o apoio da Fundação para a Ciência e Tecnologia e fundos comunitários do Compete/QREN.



Escrito ao abrigo do novo Acordo Ortográfico

Fonte: exameinformatica.sapo.pt

Jelly Bean disponível para todos os programadores

O anúncio foi feito num fórum Google, pelo programador Jean Baptiste Queru: a versão 4.1 está AOSP (de Android Open Source Project). O Jelly Bean traz algumas novidades, nomeadamente no que diz respeito às interfaces táteis e a nível do desempenho, noticia o *Ars Technica*.

As novas versões do sistema operativo são desenvolvidas pela Google e depois libertadas para a comunidade poder programar as suas aplicações. Com esta abertura é possível ainda desenhar imagens ROM de cada versão para as instalar em diversos dispositivos.

O novo tablet Nexus 7 já utiliza esta versão.

Escrito ao abrigo do novo Acordo Ortográfico

Fonte: exameinformatica.sapo.pt



TMN aposta em plataforma de empreendedorismo

Chama-se Projector a nova aposta da TMN que quer dar continuidade ao mote lançado há 100 dias, "Vamos lá". A operadora decidiu dar um empurrão aos empreendedores e lançou um portal interativo onde pretende que os portugueses divulguem as suas ideias e projetos, angariando o financiamento e os meios necessários para as pôr no terreno.

No site já há ideias e projetos de negócio, mas a TMN quer que este número cresça. E há espaço para a oferta e procura, podendo ser colocadas iniciativas de quem procura meios de as concretizar mas também de quem tem recursos para aplicar e está a tentar encontrar os parceiros certos. O site está alojado no portal "Vamos Lá", que a operadora lançou quando mudou a assinatura da marca, já teve mais de 100 mil visitantes e recolheu mais de 100 histórias de portugueses que "fizeram acontecer" as suas ideias.

A tecnologia é apenas um meio para esta iniciativa de proximidade, mais relevante na atual conjuntura de crise que impede muitos jovens e empreendedores de porem em prática o conhecimento adquirido e encontrarem um emprego ou uma forma de implementarem as suas ideias.

Escrito ao abrigo do novo Acordo Ortográfico

<http://projektor.vamosla.pt/>

Fonte: TEK Sapo

Microsoft mostra "segredos" da velocidade do Windows 8

Enquanto o novo Windows 8 não chega às lojas, a Microsoft vai revelando novas razões para apostar no sistema operativo, deixando para trás a performance do Windows 7.

No blog Building Windows 8 - da autoria da equipa de engenheiros envolvidos no desenvolvimento deste sistema operativo - a Microsoft explica a importância do processamento gráfico no novo Windows, bem como das novas ferramentas de aceleração por hardware que o Windows 8 inclui.

O *post* começa por explicar como o Windows 8 se baseia no "sucesso" da utilização dos gráficos DirectX no Windows 7, referindo que foram adicionadas novas capacidades de visualização durante a interação do utilizador, não só com o próprio Windows, mas também com as Apps específicas desenvolvidas.

De acordo com o post, as novidades introduzidas pelo Windows 8 a nível gráfico centram-se sobretudo em quatro áreas:

- ⇒ Otimização da utilização do DirectX no IE 9, pegando nos exemplos da sua utilização com o Windows Live Mail e Messenger para aplicar no desenvolvimento de novas Apps;
- ⇒ Aceleração do processo de "rendering" do texto em páginas Web, mensagens instantâneas e programas de email;
- ⇒ Melhorias no "rendering geométrico" 2D, usado para a criação de tabelas, gráficos, quadros, diagramas e em outro tipo de interfaces de utilizador. Neste caso, as alterações esperam quer a utilização mais fluída e compatível de apps online, quer no próprio desktop do PC;
- ⇒ "Grande quantidade" de melhorias na "renderização" de imagens vídeo e fotos nos formatos JPEG, GIF e PNG, de modo a aproveitar a arquitetura dos CPUs para decodificar mais rapidamente as fotos, bem como melhorar a gestão do DirectX na reprodução de vídeos online.

Para conhecer em pormenor as justificações da Microsoft poderá ler o post oficial no blogue Building Windows 8, através do endereço http://blogs.msdn.com/b/b8_br/.

Escrito ao abrigo do novo Acordo Ortográfico

Fonte: TEK Sapo



Novo sistema operativo para Mac disponível para download

Ficou hoje disponível para download a partir da loja de aplicações para Mac a mais recente versão do sistema operativo que equipa os computadores da Apple. Designada Mountain Lion, fica acessível a título de upgrade, (apenas) para utilizadores do OS X Lion e do Snow Leopard, por 15,99 euros.

Apresentada a 11 de junho tinha estreia agendada para julho e o calendário cumpriu-se com o lançamento mundial na Mac App Store da nova versão - que segundo a Apple traz mais de duas centenas de novidades, entre novas funcionalidades e melhorias a já existentes.



Da lista de alterações que chegam à boleia daquela nona versão principal do software (que o TeK detalhou aqui), a fabricante destaca pontos como a integração com a iCloud e o Facebook, a aplicação Mensagens, o Notification Center, Sharing para todo o sistema, acesso ao Game Center, Dictation e AirPlay Mirroring.

"As pessoas vão adorar as novas funcionalidades no Mountain Lion, bem como a facilidade com que este pode ser descarregado e instalado a partir da Mac App Store", afirma o vice presidente de marketing da Apple, Philip Schiller, citado num comunicado enviado aos meios de comunicação.

A nova forma de comercialização do sistema operativo, através da loja online, foi introduzida com a versão anterior do software, o OS X Lion, e ao que tudo indica tem estado a ser bem recebida. Quando este foi lançado, em julho de 2011, foram registados 1 milhão de downloads logo nas primeiras 24 horas. A novidade permite também disponibilizar a solução a um preço mais baixo, o que é apontado como um fator a seu favor.

De acordo com os últimos números fornecidos pela empresa de Cupertino, há 66 milhões de clientes Mac, um número três vezes maior que o registado há cinco anos. A última versão do OS X foi descarregada 26 milhões de vezes, fazendo com que 40% dos utilizadores de computadores da marca tenham a versão mais recente do sistema operativo instalada.

Escrito ao abrigo do novo Acordo Ortográfico

Joana M. Fernandes

Fonte: TEK Sapo

TEMA DA CAPA

As novidades do Visual Studio 2012 RC (Visual Basic)

As novidades do Visual Studio 2012 RC

É mais que oficial. Foi já desde dia 31 de Maio que ficaram disponíveis para download as versões RC (Release Candidate) do novíssimo Visual Studio 2012 (também conhecido por Visual Studio 11.0) e da .NET Framework 4.5. Como sempre, uma nova versão, implica um reforço de funcionalidades e ferramentas com o intuito de nos facilitar o trabalho e melhorar a produtividade. O novo Visual Studio 2012 (que vou abreviar para VS11 daqui em diante) não é excepção. Aliás, o VS11 é diferente o suficiente para dar que falar, talvez até muito mais do que alguma versão que já tenha sido libertada desde o Visual Studio 7 (também conhecido simplesmente por Visual Studio .NET). A versão 4.5 da .NET Framework conta com a versão 11 da linguagem Visual Basic, versão 5 de C# e versão 3 de F#.

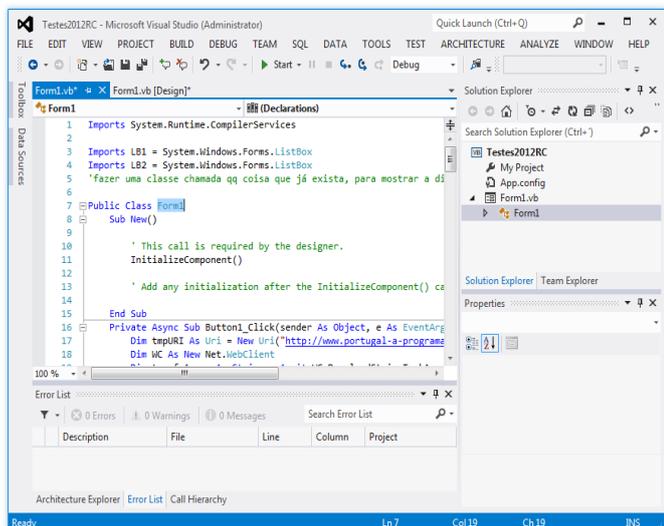
Já todos ouvimos falar do interface Metro, e já todos percebemos o esforço imenso que a Microsoft tem vindo a empregar para uniformizar a experiência de todos os seus produtos para o novo interface. A maior mudança do VS11 passa precisamente por aí. Quer se goste, quer não, o Metro veio para ficar.

As influências Metro começam-se a notar logo desde o novo logótipo do IDE.



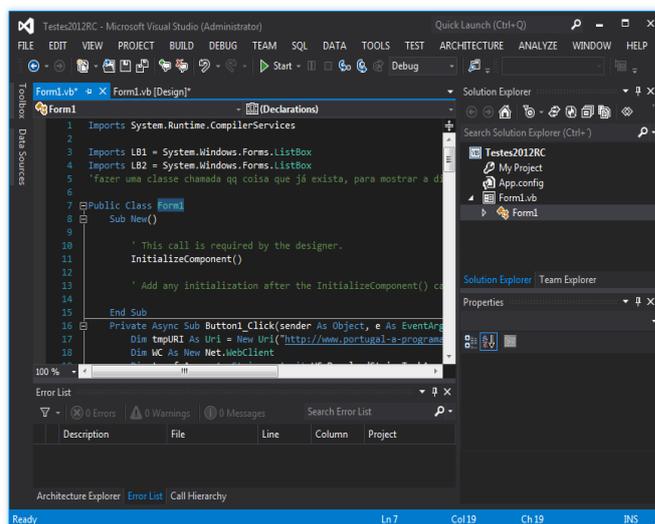
Os complexos gradientes e transparências, bem como as curvas perfeitamente calculadas dão lugar à cor sólida e à forma simples e limpa.

Na nova interface predominam os contrastes.



Desaparecem todos os indícios de curvas e até os rótulos dos itens de menu foram colocados em maiúsculas, à semelhança da experiência Metro nos dispositivos que já a implementam.

Todos os ícones foram também eles redesenhados para a nova realidade. Os mesmos ícones são igualmente usados no novo tema nativo: dark, tal e qual como a experiência Metro do Windows Phone 7 prevê: light e dark. É possível observar que os ícones são rigorosamente os mesmos, quando deveriam ter invertido a cor. É possível que seja temporário.



Este interface já foi retocado desde o release Beta, onde existiam indícios mais fortes do Metro, mas que foram fortemente criticados, por a negativa, por a maioria dos utilizadores. Existiam, por exemplo, muito mais sítios onde se usavam as letras maiúsculas.

Mas, felizmente, a versão 11 não é só uma cara lavada. Existem algumas melhorias que vale a pena destacar, não só directamente do ambiente de desenvolvimento, mas relacionadas também com a nova versão da Framework.

Ao nível do IDE

Vou começar por focar alguns dos pontos mais relevantes a nível do IDE.

A leveza que se nota

Não é novidade se disser que o VS10 já rendia algumas coisas por via de aceleração gráfica por hardware, e no

TEMA DA CAPA

AS NOVIDADES DO VISUAL STUDIO 2012 RC

VS11 essa presença é reforçada. Não só o interface se tornou mais limpo e mais fluído, como o desempenho geral do IDE está significativamente melhorado. A resposta do interface foi muito favorecida e os recursos utilizados muito melhorados, mesmo ainda se tratando de um candidato a release.

O lento assassinio do rato

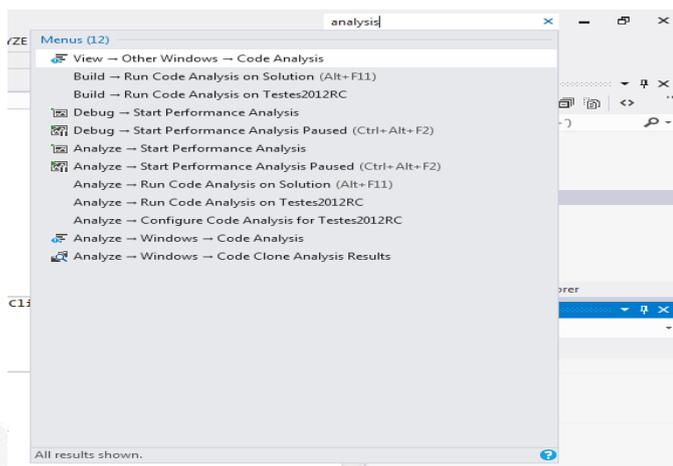
Um dos pontos que mais me agradou na nova versão, e notei praticamente de imediato, é um pormenor, do qual já tive oportunidade de discutir com outros colegas de ofício. Pode parecer irrelevante, mas é algo que pode poupar muito tempo quando estamos a desenhar o nosso interface, (nos tipos de projectos que implicam interface). Falo do separador da caixa de ferramentas, ou dos separadores da consola inferior.



Nas versões anteriores, só existiam duas formas de aceder à caixa de ferramentas, ou à consola inferior: ou se afixava, sacrificando espaço, ou se passava o cursor do rato por cima, o que as fazia surgir. Para poupar espaço útil, afixar sempre esteve fora de questão, e o surgir torna-se verdadeiramente irritante quando não temos intenção de as abrir. A aproximação à experiência Metro obriga quase inevitavelmente a ter de pensar nos interfaces como sendo exclusivamente desenhados para o toque, em detrimento do rato. Assim, a única forma de fazer surgir estas janelas é através de um toque, quando realmente as queremos abrir.

O sentido de acessibilidade permanente

À semelhança de vários produtos Microsoft, o VS11 possui uma vasta selecção de opções e menus que se podem tornar verdadeiros labirintos onde acabamos sempre por perder mais do que tínhamos pensado perder só para encontrar determinada funcionalidade. O VS11 conta com uma barra de pesquisa rápida, como a que foi introduzida no menu Iniciar do Windows Vista.



A partir deste "Quick Launch" convenientemente colocado em espaço antes inútil, podemos aceder a qualquer item que exista na árvore de menus e também à lista de documentos abertos.

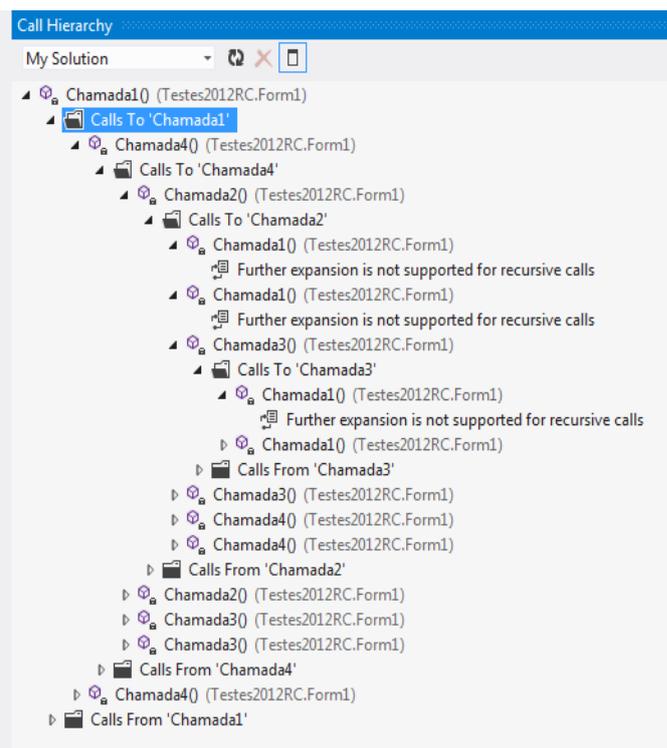
Hierarquia de chamadas

Mais do que apenas encontrar referências de uso de determinado membro, o VS11 disponibiliza uma versão reforçada da "Call Hierarchy" que nos permite não só encontrar referências de determinado membro, mas também perceber de imediato a partir de que posição hierárquica é chamado e que chamadas subordinadas pode despoletar.

Para efeitos de teste, vamos considerar o seguinte bloco de código:

```
Private Sub Chamada1()  
    Chamada2()  
    Chamada3()  
End Sub  
Private Sub Chamada2()  
    Chamada4()  
End Sub  
Private Sub Chamada3()  
    Chamada2()  
    Chamada4()  
End Sub  
Private Sub Chamada4()  
    Chamada1()  
    Chamada2()  
End Sub
```

Poderá ser fácil de rastrear, porque é pequeno e não contém nada, mas até aqui se consegue demonstrar a utilidade da hierarquia de chamadas.



AS NOVIDADES DO VISUAL STUDIO 2012 RC

Como se observa, conseguimos de imediato uma representação visual das chamadas aos membros que são feitas, até um passo antes da recursão.

Os projectos orientados para a Web também receberam vários brinquedos novos no interface, sendo o mais significativo a opção de poder escolher o browser a testar naquela execução, poupando minutos preciosos, mas não é na Web que me quero focar.

Existem, naturalmente, muito mais pormenores e diferenças que se vão notar com o uso. Não posso enumerar tudo.

Ao nível da linguagem

Ao nível da linguagem também encontramos melhorias significativas, que provêm da novíssima .NET 4.5.

Programação Assíncrona

A .NET 4.5 disponibiliza um novo modificador **Async** que está intrinsecamente ligado com a nova palavra-chave **Await**.

Este novo conjunto de recursos permitem-nos realizar operações assíncronas, em diferentes threads, de uma forma tão natural e fluida como se tratasse de fluxo síncrono.

Para dar o devido valor a esta nova funcionalidade, há que entender qual a vantagem em executar operações assincronamente ou esperar que uma termine para completar outra. Vamos pensar na execução da thread do interface como um carro em marcha, numa linha recta. Vamos também considerar que o condutor é algo limitado em termos de execução de tarefas.

Durante o percurso, queremos executar duas operações muito básicas: abrir um vidro e mais à frente ligar o rádio e só voltar à condução quando o vidro terminar de abrir ou quando o CD do carro carregar as pistas. Ao ocupar o condutor de forma a que seja necessário aguardar que o vidro desça ou que o leitor de CDs carregue as pistas, teríamos obrigatoriamente de parar o carro, esperar por o fim da operação e só depois arrancar. Isto é moroso, e afasta o condutor do que realmente deveria estar a fazer: a conduzir.

Quando executamos determinadas operações de forma assíncrona, é como se o nosso condutor agora viajasse também com a sua mulher. Enquanto ele se focava única e exclusivamente à condução, e com o carro em marcha permanente, basta indicar ao passageiro que abra o vidro, ou que ligue o rádio. Independentemente do tempo que este demorar, foi ordenado algures no tempo, e vai surtir um efeito, quando o tiver de surtir. Não interessa quando. E o carro continua a andar.

Ainda que vaga, esta analogia pode sensibilizar, não só para a importância da thread do interface (a responsável por desenhar o que vemos no ecrã), mas também no que isso significa para a experiência do utilizador.

A maioria das pequenas operações podem ser executadas logo na thread do interface (UIT a partir daqui [User Interface Thread, ou Thread da interface do utilizador]), e muitas fazem todo o sentido que lá sejam executadas. Todas as outras, que exigem poder de processamento ou que são demoradas, devem ser executadas por todos, excepto a UIT, que tem de se mostrar sempre "responsive".

Peguemos num exemplo básico onde a execução assíncrona faz todo o sentido: download da resposta a um pedido a um webserver. Se feito sem ser assincronamente, o UIT teria de esperar por a resposta completar, o que poderia demorar vários segundos, dependendo da capacidade de resposta do servidor. Em versões anteriores da .NET, poderia fazer-se da seguinte forma:

```
Private Sub Button1_Click(sender As Object,
    e As EventArgs) Handles Button1.Click
    Dim tmpURI As Uri = New Uri("http://
www.portugal-a-programar.pt")
    Dim WC As New Net.WebClient
    AddHandler WC.DownloadStringCompleted,
        Sub(sender As Object,
            e As Net.DownloadStringCompletedEventArgs)
            Trace.Write(e.Result)
        End Sub
    WC.DownloadStringAsync(tmpURI)
End Sub
```

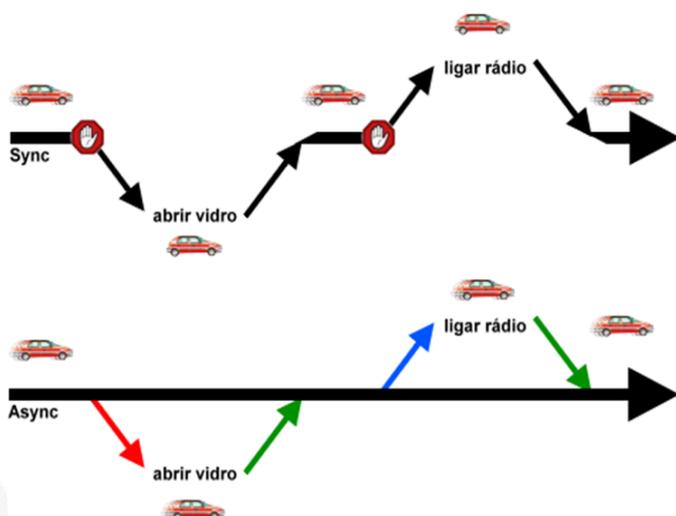
O handler de um evento tinha sempre de ser registado, o que se traduz em mais um membro, neste caso anónimo, mas sempre mais um membro. Para além disso, o fluxo do código torna-se algo confuso e difícil de acompanhar.

Agora é possível escrever o mesmo da seguinte forma:

```
Private Async Sub Button1_Click(sender
    As Object, e As EventArgs) Handles Button1.Click
    Dim tmpURI As Uri = New Uri("http://
www.portugal-a-programar.pt")

    Dim WC As New Net.WebClient

    Dim tarefaAsync As String = Await
```



TEMA DA CAPA

AS NOVIDADES DO VISUAL STUDIO 2012 RC

```
WC.DownloadStringTaskAsync(tmpURI)
    Trace.Write(tarefaAsync)
End Sub
```

O modificador **Async** na definição do membro indica que aquele método poderá fazer chamadas assíncronas. Quando essas chamadas forem necessárias, devem ser acompanhadas por a keyword **Await** que indica que aquele é um ponto de execução onde o fluxo deve resumir dentro daquele método, assim que a chamada assíncrona terminar.

Este é um exemplo com um método conhecido, onde a tarefa assíncrona já está implementada, mas é perfeitamente possível implementar as nossas próprias tarefas. O seguinte bloco de código representa um método “pesado” que devolve uma Task. Qualquer devolução de Task é elegível para processamento assíncrono.

```
Private Function MetodoPesado() As Task(Of Integer)
Dim RT = New Task(Of Integer)(Function()
    Dim c As Integer = 0
    For i As Integer = 0 To Integer.MaxValue / 2
        c += 1
    Next
    Return c
End Function)
RT.Start()
Return RT
End Function
```

O nosso método “pesado” implica uma contagem desde o zero até metade do limite máximo que um integer pode representar.

Uma tarefa tão simples como esta, poderia pendurar o UIT por alguns segundos, sem necessidade nenhuma.

Para executar este método assincronamente bastaria:

```
Private Async Sub Button2_Click(sender As Object,
    e As EventArgs)
Handles Button2.Click
    Trace.Write("A começar o método pesado..." &
        vbCrLf)
    Dim num As Integer = Await MetodoPesado()
    Trace.Write("Resultado do método pesado: " &
        num.ToString)
End Sub
```

Onde se podem identificar novamente o **Async** e o **Await**

Caller information

À semelhança do que já acontece em outras linguagens, é agora possível extrair informações do membro que foi chamado.

Isto é perfeito se pensarmos em mais-valias de depuração, relatório de erros e bugs vindos do outro mundo.

Para receber essas informações, basta acrescentar à assinatura do método a monitorar as propriedades que se pretendem recolher identificadas por o seu atributo.

Existem 3 possíveis:

Atributos	Descrição	Tipo de dados
CallerFilePathAttribute	Caminho completo do ficheiro fonte onde o membro pertence	String
CallerLineNumberAttribute	Número da linha no código do ficheiro fonte onde o membro pertence	Integer
CallerMemberNameAttribute	Nome do membro em questão	String

```
Private Sub btCallerInfo_Click(sender As Object,
    e As EventArgs) Handles btCallerInfo.Click
    CallerInfo()
End Sub

Public Sub CallerInfo(
    <CallerMemberName> Optional nomeMembro As String =
    Nothing, <CallerFilePath> Optional ficheiroFonte
    As String = Nothing, <CallerLineNumber> Optional
    numeroLinha As Integer = 0)
    Trace.Write("Chamada disparada de " & nomeMembro
        & vbCrLf)
    Trace.Write("Do ficheiro " & ficheiroFonte &
        vbCrLf)
    Trace.Write("Na linha " & numeroLinha & vbCrLf)
End Sub
```

O que produz algo assim:

Chamada disparada de btCallerInfo_Click

Do ficheiro C:\Users\Sergio Ribeiro\Documents\Visual Studio 2012\Projects\Testes2012RC\Testes2012RC\Form1.vb

Na linha 51

Esta passagem de informação é assegurada por o namespace “Compiler Services”, onde pertencem os 3 atributos usados.

Para que a informação seja correctamente atribuída, os parâmetros precisam de ser opcionais, e consequentemente de possuir um valor por defeito. Caso contrário, faziam parte da assinatura obrigatória.

Iteradores

Os iteradores, apesar de relativa novidade, já foram abordados na edição 33 da revista, pelo que não vou aprofundar.

São basicamente a disponibilização de novas keywords que nos permitem criar os nossos próprios tipos iteradores.

Sempre que utilizamos a keyword **Yield** dentro de um método assinalado com o modificador **Iterator** estamos a instruir uma nova iteração.

Com esta lógica, podemos construir colecções iteráveis, com qualquer regra de iteração que precisemos de empregar. Dois exemplos simples, iterar os números 1,5,8,11,57,12, que não seguem uma lógica de incremento, e iterar todos os números pares dentro de um dado alcance, como exemplo de implementação de regra de iteração:

```
Private Sub btIteradores_Click(sender As Object,
    e As EventArgs) Handles btIteradores.Click
    For Each number As Integer In ParesEntre(5, 18)
        Trace.Write(number & ",")
    Next
    For Each number As Integer In Simples()
        Trace.Write(number & ",")
    Next
End Sub
Private Iterator Function Simples() As IEnumerable
(Of Integer)
    Yield 1
    Yield 5
    Yield 8
    Yield 11
    Yield 57
    Yield 12
End Function
Private Iterator Function ParesEntre(ByVal primeiro
As Integer, ByVal ultimo As Integer)
As IEnumerable(Of Integer)
    For numero As Integer = primeiro To ultimo
        If numero Mod 2 = 0 Then Yield numero
    Next
End Function
```

O código acima produzia o seguinte output:

6,8,10,12,14,16,18,1,5,8,11,57,12,

Namespaces ambíguos

Também os namespaces receberam algum foco nesta nova versão, nomeadamente ao que toca a ambiguidade dos nomes.

Um dos grandes focos, que não são novidade para o VS11, são as referências a nomes “fully qualified”. É possível fazer objectos referência a determinado namespace, e usar a referência para definir os tipos durante o projecto.

Vamos imaginar que temos um projecto onde precisamos de chamar a determinado objecto nosso, “TextBox”, que vamos definir da seguinte forma:

```
Namespace MeusObjectos
    Public Class TextBox
        Sub New(Linhas As Integer)
            End Sub
    End Class
End Namespace
```

Num projecto Windows Forms, criamos uma suposta instância desse objecto:

```
Dim T As New TextBox
```

Mas o compilador não dá sinal de erro. Deveria dar porque a assinatura do meu objecto obriga a um parâmetro. O problema, é que o nosso objecto tem um nome que já existe num namespace preferencial. O objecto que estamos na verdade a instanciar é o **System.Windows.Form.TextBox**

Para contornar isto, teríamos de fazer a instância com o nome completamente qualificado:

```
Dim T As New Testes2012RC.MeusObjectos.TextBox(1)
```

Assim já temos a certeza a que objecto nos estamos a referir, e já precisamos de especificar o parâmetro do construtor. Estamos de facto a instanciar o nosso objecto em detrimento do outro.

Isto faz sentido, mas se imaginarmos que sempre que nos quisermos referir ao nosso objecto, temos de fornecer o nome completamente qualificado, o código vai-se tornar desnecessariamente extenso e cada vez menos legível.

Como também não podemos fazer “Imports” de ambos, por terem o mesmo nome, é nos possível criar um Import especial, em referência:

```
Imports TextBoxNativa =
    System.Windows.Forms.TextBox
Imports MinhaTextBox =
    Testes2012RC.MeusObjectos.TextBox
```

Com estes Imports já nos é permitido usar a referência da seguinte forma:

```
Dim T1 As New TextBoxNativa
Dim T2 As New MinhaTextBox(1)
```

Este tipo de referência é um excelente atalho e uma mais-valia para a legibilidade do código.

Existem outros casos de ambiguidade mais graves, que geram quebras em namespaces importantes. Por exemplo, vamos alterar a nossa TextBox para a seguinte:

```
Namespace System.MeusObjectos
    Public Class TextBox
        Sub New(Linhas As Integer)
            End Sub
    End Class
End Namespace
```

Ao usarmos o namespace System, estamos a impedir que nos possamos referir ao namespace System nativo. Por exemplo, se eu quiser instanciar uma TextBox nativa, vou ter problemas porque os namespaces de System passam a ser apenas o “MeusObjectos”.

Para contornar a situação, existe um namespace Global. É a partir dele que recuperamos novamente a referência ao

TEMA DA CAPA

AS NOVIDADES DO VISUAL STUDIO 2012 RC

System que queremos.

Isto levanta outra questão: e se eu quiser chamar Global ao meu namespace?

Não é possível. Podemos sim incluir o nosso namespace no namespace Global:

```
Namespace Global.MeusObjectos
    Public Class TextBox
        Sub New(Linhas As Integer)
            End Sub
        End Class
    End Namespace
```

Mas não é possível “aninhar” namespaces com o nome Global, por motivos óbvios.

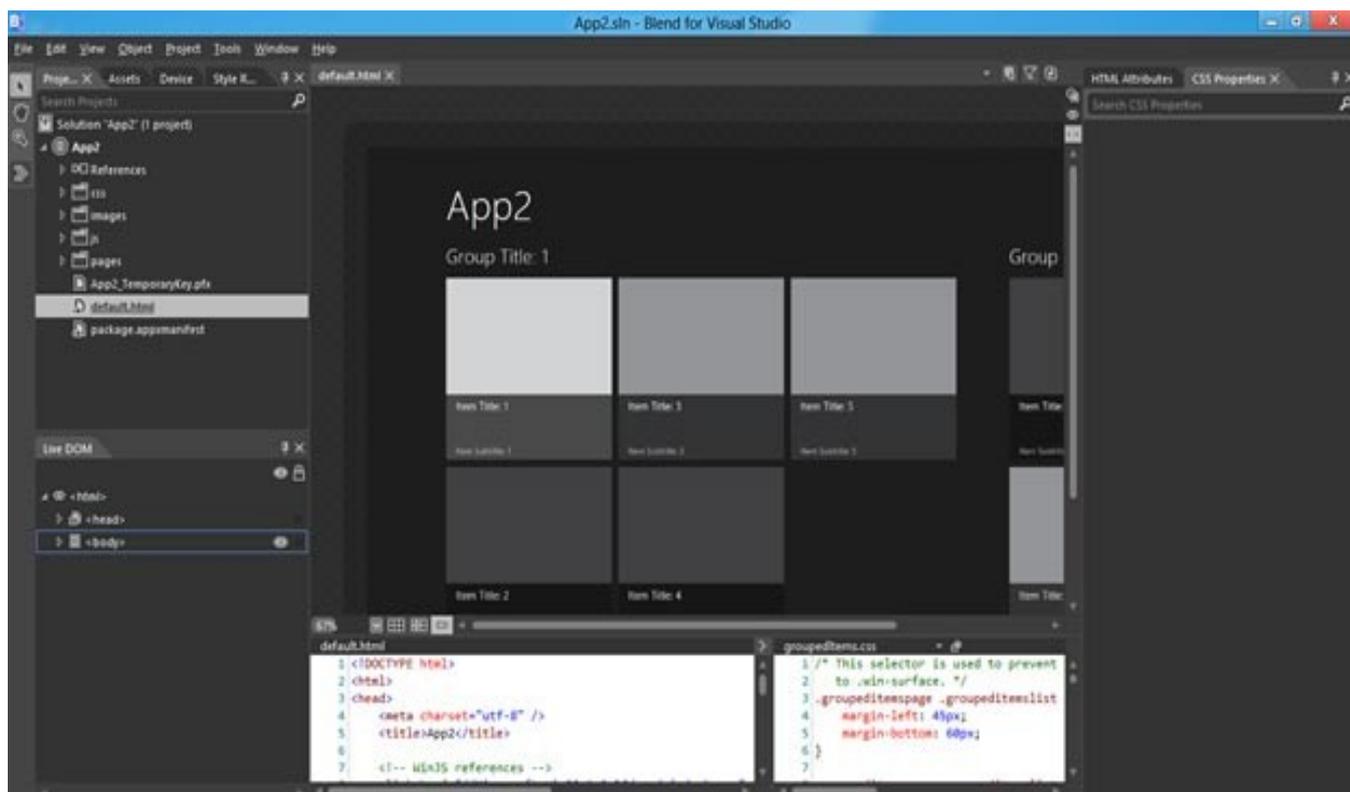
Em suma

No universo Microsoft (e a chave aqui é “universo Microsoft”), o Visual Studio sempre foi um companheiro de todos os programadores. Um excelente ambiente de desenvolvi-

mento que tem vindo de versão para versão mostrar a quão séria é a atitude da empresa com os programadores. Cada nova versão faz passar a noção de que o programador é sempre colocado no topo de todas as novas tecnologias Microsoft e de que tem mesmo de experimentar um bocadinho de todas para descobrir as necessidades dos seus projectos.

A Microsoft consegue assim, andar mais um passo em direcção da completa convergência para um excelente ecossistema, onde existe uma única ferramenta, um único ambiente, para todas as mainstream platforms no mercado.

Julgo que as versões Express não estejam oficialmente confirmadas, mas são tidas quase como certas. As iniciativas Express são a forma perfeita de viver a experiência Microsoft, gratuitamente, uma vez que não existe limitação a nível da Framework.



AUTOR



Escrito por Sérgio Ribeiro

Curioso e autodidacta 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>

A PROGRAMAR

Pascal – construção de uma calculadora gráfica

Formulários em HTML5

SEO: Search Engine Optimization – Introdução parte II

Implementação da Cifra ARSD em Ambiente Emulado Windows Azure

Estação de Controlo de Temperaturas Usando Arduino e Texas Instruments LM335A

Pascal – construção de uma calculadora gráfica

Muitos acharão que Pascal e as suas variantes, com a excepção para o Delphi, mais não fazem se não programas para consola, ou seja, *Console Applications*. Isso seria verdade não fosse o facto do Turbo Pascal ter dado há vários anos, ainda na década de 80, um grande passo para a criação de ambientes gráficos mais apelativos.

Tinha surgido, então, a **unidade Graph**. Esta nova unidade traz consigo uma panóplia de métodos que nos permitem criar desenhos pixel a pixel, ou seja, criam ambientes para além da tradicional e “velhinha” consola.

Na sua essência mais simples, esta unidade detecta as características gráficas do computador onde corre o programa e cria uma área gráfica onde se podem criar os mais variados desenhos e textos.

Para quem está familiarizado com *Visual Basic .NET*, a classe **Graphics** bem como as *namespaces Drawing* e *Drawing2D* são os exemplos com termo de comparação mais próximo.

Para auxiliar o programador, alguns novos tipos de dados foram incluídos na unidade **types**, sendo de enfatizar as duas mais simples e usuais, **TPoint** e **TSize**.

O objectivo deste artigo não será, contudo, dar noções introdutórias desta unidade de um ponto de vista puramente teórico. Esta unidade é tão única e tão diferente de qualquer outra que o Free Pascal oferece que merece uma abordagem totalmente diferente.

Para dar a conhecer devidamente o poder desta unidade, o objectivo do artigo será a construção daquilo que é a base de qualquer calculadora gráfica: uma **Plotter**.

Para quem não está familiarizado com o termo, uma *Plotter* não é mais do que uma calculadora gráfica. As suas capacidades são diversas, consoante as funcionalidades que o seu programador lhe coloque. Contudo, qualquer *Plotter* terá de receber uma função, calcular os seus valores, que serão pontos no gráfico, e unir esses mesmos pontos com linhas de forma a dar uma visualização contínua e não “picotada”.

Assim sendo, neste artigo será criada uma pequena unidade que terá um novo tipo de dado que conterá as propriedades e os métodos básicos de uma *Plotter* – uma classe, portanto, bem como um pequeno programa de teste que irá usar essa unidade para representar graficamente a seguinte função:

$$f(x) = x^2 - 8$$

Ou seja, esperamos que o resultado final da utilização da unidade por nós criada para este feito seja semelhante ao da **Figura 1**.

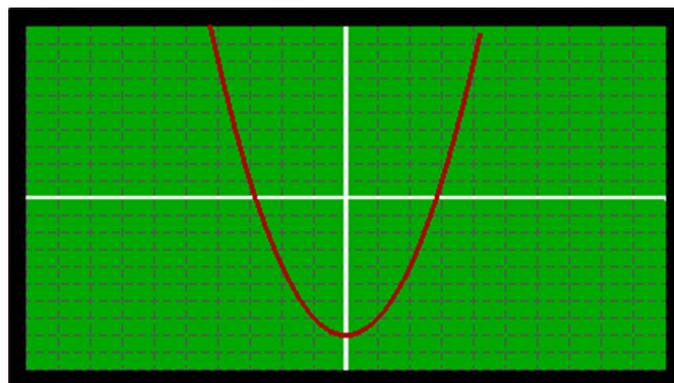


Figura 1 – Exemplo de *output* esperado com a construção da *Plotter*.

Logo com esta imagem denotamos as diferenças entre a área gráfica criada pela unidade **Graph** e a consola a que estamos tão acostumados: a personalização colorida pixel a pixel.

Portanto, em suma, os objectivos para este artigo são os seguintes:

Dar a conhecer a *unit Graph* no seu conceito geral;

- Criar uma classe que permita gerar e personalizar uma *Plotter* básica;
- Conceber uma *unit* que tenha esta classe como um novo tipo de dados, bem como inclua os tipos de dados auxiliares que sejam necessários formar;
- Criar um pequeno programa de teste desta nova *unit* que permita experimentar a *Plotter*;

Mais uma vez, a par dos meus artigos anteriores, identificarei sempre a linguagem de programação por Pascal e não por Object Pascal por uma questão de simplificação. Será usada, doravante, as notações *unit* ou unidade, e não módulo.

Todos os métodos e propriedades, tipos de dados e constantes serão nomeados em inglês.

Por fim, o código aqui presente foi todo feito para ser compilado no **Free Pascal 2.6.0** em *Windows*. Algumas adaptações poderão ter de ser feitas para *Mac OS* e para *GNU/Linux*.

A *unit Graph* – noções e conceitos

É óbvio que não faria muito sentido começar a construir a *Plotter* sem antes ter algumas noções introdutórias sobre a *unit* na qual nos vamos basear.

A *unit Graph* é uma unidade que contém uma série de métodos e constantes que permitem criar um ambiente gráfico numa janela à parte da consola e que permite o desenho de formas avançadas.

Para se ter uma noção geral da imensidão desta *unit*, esta contém:

- Mais de 70 métodos de desenho e personalização;
- Mais de 200 constantes;
- Mais de 20 tipos de dados.

A criação do ambiente gráfico (**Figura 2**) é feita à custa de um procedimento, **InitGraph**, que irá receber por parâmetro a resolução e a *driver*. Contudo, como estes valores fazem parte das mais de 200 constantes da unidade, um outro procedimento, **DetectGraph**, oferece a possibilidade de detectar automaticamente estes parâmetros.

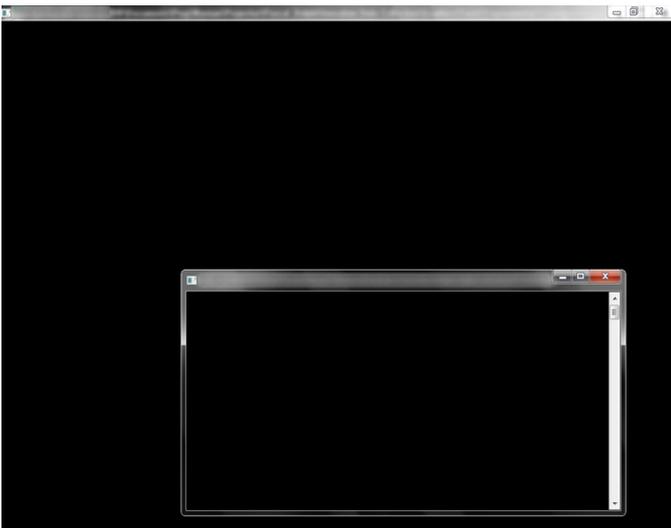


Figura 2 – A área gráfica criada pelo **InitGraph** com a tradicional consola sobre ela.

Assim sendo, e sabendo que os valores obtidos são do tipo **SmallInt**, é necessária a declaração de duas variáveis que irão receber através do **DetectGraph** os valores a utilizar no procedimento **InitGraph**:

```
var Driver, Modus : SmallInt;  
// ...  
DetectGraph(Driver, Modus);  
InitGraph(Driver, Modus, '');
```

Como poderão ter reparado, o procedimento **InitGraph** recebe um terceiro parâmetro, do tipo **String**, e que, neste caso, recebeu uma *String* vazia. Este procedimento é para ser, regra geral, ignorado, a não ser que utilize o Borland Pascal – nesse caso, terá de passar a *String* "C:\PP\BGI", isto su-

pondo que tem o Borland Pascal instalado na raiz da unidade C.

Por fim, antes do programa ser terminado, ou quando a área gráfica já não for necessária, esta deve ser encerrada através do seguinte procedimento:

```
CloseGraph;
```

Em princípio não deverão ocorrer erros no decorrer destes processos já que, segundo a documentação do Free Pascal, estes não devolvem quaisquer erros na sua execução. Contudo, outros métodos da *unit Graph* poderão gerar erros, e, por este motivo, a parte do programa que estiver a trabalhar com esta unidade deverá estar enquadrada numa estrutura de tentativa. Como em Pascal esta não inclui os blocos **Except** e **Finally** numa só estrutura e devem estar sim encadeadas, recomendo a seguinte:

```
try  
  try  
    DetectGraph(Driver, Modus);  
    InitGraph(Driver, Modus, '');  
    // código...  
  except  
    ON ex:exception do begin  
      // em caso de erro  
    end  
  end  
finally  
  CloseGraph;  
end;
```

Ou seja, qualquer excepção que ocorra deverá ser detectada e, caso o programador assim o deseje, informada. Por fim, aquando do final de todas as acções, a área gráfica é encerrada (**CloseGraph**), e como este processo não sofre erros, ficará no bloco **Finally**.

Para a possibilidade de utilizar esta estrutura, dever-se-á incluir, no Free Pascal, a directiva de compilação **{\$mode objfpc}** na primeira linha do código-fonte. A *unit sysutils* deverá ser igualmente incluída caso se queira "apanhar" e tratar as excepções ocorridas no decorrer do programa.

A *unit Graph* permite, então, desenhar e preencher objectos variados (**Figura 3**), tais como:

- Pontos;
- Linhas;
- Rectângulos;
- Polígonos;
- Elipses;
- Círculos;

A PROGRAMAR

PASCAL – CONSTRUÇÃO DE UMA CALCULADORA GRÁFICA

- “Queijos” (*pies*);



Figura 3 – Exemplo muito simples da utilização das capacidades da unidade *Graph*.

A *unit Graph* permite também usar preenchimentos que não sejam sólidos mas que tenham no seu lugar um padrão (*pattern*), sendo mesmo possível criar padrões personalizados, entre muitas outras capacidades.

Dada esta muito breve introdução genérica à unidade sobre a qual vamos trabalhar, é chegado o momento de iniciar a construção da *Plotter*, mas sem antes definir o que esta fará.

A *Plotter* – funcionalidades e personalizações a incluir

- A nossa *Plotter*, como calculadora gráfica que é, deverá incluir:
- Uma área de trabalho, área esta que tem um tamanho (**Size**);
- Eixos das abcissas (XX) e das ordenadas (YY) (**Axes**);
- Uma grelha que marca os pontos relevantes do *zoom* actual (**Grid**);
- Um **Zoom**;

Para a personalizar, poderemos incluir as seguintes opções:

- Cor de fundo;
- Estilo de linha dos eixos;
- Estilo de linha da grelha;
- Estilo de linha da função.

Para esclarecer bem o que são a grelha, os eixos, a função e a área de trabalho segue-se uma imagem elucidativa (Figura 4).

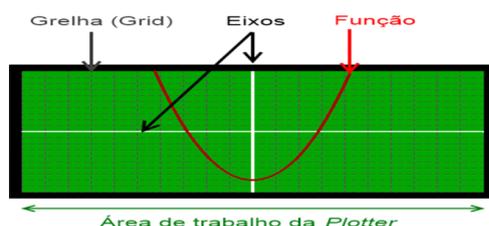


Figura 4 – Estrutura básica da *Plotter* a ser criada.

Para facilitar o programador que utilizará a classe, podemos incluir valores padrão, isto é, **constantes**, na unidade que irá conter a nova classe:

- Zoom padrão;
- Estilos de linha padrão para os eixos, a grelha e a função;
- Cor de fundo padrão.

Deveremos, então, começar a definir os métodos e propriedades da classe. Por convenção, novos tipos de dados iniciam-se com a letra “T”. Para a nossa classe, os valores privados terão a letra inicial “v”, de *value* (valor), e os parâmetros das funções e procedimentos terão a letra inicial “c”, de *constant* (constante), de forma a poder distinguir devidamente os valores dos parâmetros e das propriedades sem ter de diferenciar em demasia os nomes.

A nossa classe será, então, a **TPlotter** e esta irá herdar daquela que é classe “principal” do Object Pascal: a **TObject**.

```
TPlotter = class(TObject)
```

Como propriedades e respectivos tipos de dados, a classe **TPlotter** incluirá:

- **Grelha**, do tipo **Estilo de Linha**, a definir mais à frente;
- **Eixos e Estilo da Função**, do mesmo tipo de dado;
- **Posição**, do tipo **TPoint**;
- **Dimensão**, do tipo **TSize**;
- **Zoom**, do tipo **Zoom**, igualmente a definir mais à frente;
- **Cor de fundo**, do tipo **Word**;

Outras propriedades poderão ser necessárias e definidas ao longo da construção da *Plotter*, mas estas são as essenciais. Por fim, os métodos da classe serão, essencialmente:

- **Criar** (construtor);
- **Libertar** (destrutor);
- **Mostrar**;
- **Apagar**;
- **Relocalizar**;
- **Redimensionar**;
- **Redefinir Zoom**;
- **Carregar função**;
- **Desenhar função**;
- **Apagar função**;
- **Libertar função**;

Iremos definir os novos tipos de dados quanto a classe for construída, já que, desta forma, se tornará mais óbvia a necessidade não só da sua existência mas também da sua estrutura final.

Devido às dificuldades que se colocam em explicar a construção de uma classe destas, iremos construí-la por passos segundo a ordem lógica de “*construir primeiro os alicerces e só depois o telhado*” – todos estes passos serão, doravante, numerados para uma melhor interligação entre estes.

1. Declaração da nova unidade e definição da classe

Como estamos a criar uma pequena calculadora gráfica, vamos denominar a nossa unidade de **GraphCalc**, que ao mesmo tempo aluz ao facto de se utilizar a *unit* **Graph**.

Como vamos necessitar dos tipos de dados **TPoint** e **TSize**, necessitaremos igualmente da *unit* **types**.

```
UNIT GraphCalc;
INTERFACE
uses graph, types;
// declaração
IMPLEMENTATION
// implementação
END.
```

A nossa classe **TPlotter** herda da classe genérica **TObject**, e terá valores privados e métodos e propriedades públicas:

```
TYPE
TPlotter = CLASS(TObject)
PRIVATE
// privado
PUBLIC
// público
END;
```

2. Propriedades de localização, tamanho e cor de fundo

A **cor de fundo** não necessita de nenhum método especial de escrita nem de leitura – basta-lhe atribuição directa –, nem necessita de ser só de leitura ou só de escrita (ou seja, *ReadOnly* ou *WriteOnly*). Dado isto, podemos declarar a cor de fundo directamente como uma variável na secção pública da classe:

```
PUBLIC
BackColor : Word;
```

Por sua vez, a **posição** e a **dimensão** podem ser lidos, mas a sua alteração implica que o desenho da *Plotter* mude – a escrita destas duas propriedades deverá ser controlada. Para relocalizar e redimensionar a *Plotter* iremos usar dois procedimentos separados. Ou seja, teremos estas propriedades como *ReadOnly* e a sua alteração dar-se-á com métodos independentes destas propriedades e que serão implementadas mais à frente.

Em código, teremos o seguinte:

```
PRIVATE
vPosition : TPoint;
vSize : TSize;
PUBLIC
property Position : TPoint read vPosition;
property Size : TSize read vSize;
```

Caso utilize outro compilador e necessite de declarar estes dois tipos de dados devido à inexistência da *unit* **types**, segue-se a sua definição segundo a documentação do Free Pascal, pronto a implementar:

```
TYPE
TSize = packed record
cx: LongInt; // largura em pixéis
cy: LongInt; // altura em pixéis
end;

TPoint = packed record
X: LongInt; // posição horizontal
Y: LongInt; // posição vertical
end;
```

3. Estilos das linhas da grelha, dos eixos e da função – definição do tipo de dados **TLineStyle**

Os estilos de linha da grelha, dos eixos e da função podem ser alterados e lidos a qualquer momento. Para que as alterações sejam aplicadas, o mais fácil será criar um procedimento genérico de actualização (*Refresh*) que irá limpar a área da *Plotter* e redesenhá-la. Por este motivo, os estilos de linhas serão de acesso geral, sem qualquer restrição na leitura ou na escrita:

```
PUBLIC
Grid : TLineStyle;
Axes : TLineStyle;
FunctionLine : TLineStyle;
```

Como é constatável, este tipo de dados não existe, pelo que será criado por nós. Seguindo a mesma estrutura dos tipos **TPoint** e **TSize**, iremos criar o **TLineStyle** como um **Packed Record**, ou seja, este ocupará em memória única e exclusivamente o espaço necessário.

- Este tipo de dados incluirá as seguintes definições de linha:
- Cor;
- Espessura;
- Estilo (contínua, tracejada...);
- Visibilidade.

A PROGRAMAR

PASCAL – CONSTRUÇÃO DE UMA CALCULADORA GRÁFICA

```
TYPE
  TLineStyle = PACKED RECORD
    Color : word;
    Thickness : word;
    Style : word;
    Visible : boolean;
  END;
```

A visibilidade fica definida pois tornar-se-á fácil controlar quais as linhas que aparecem fazendo um controlo com uma estrutura de controlo **if then**.

Desta forma, se definirmos **Grid.Visible** a **False**, a grelha não é desenhada e o *output* já será do género da **Figura 5**.

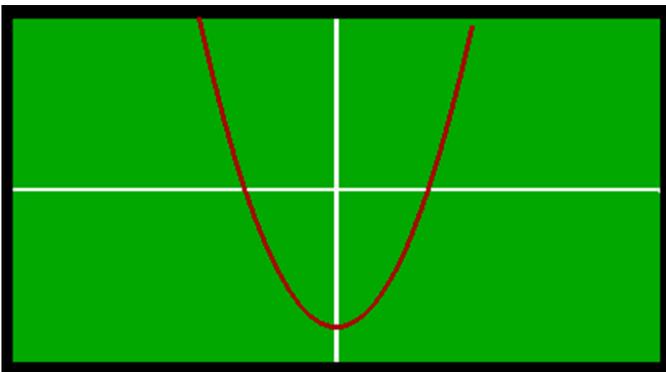


Figura 5 – Output da Plotter com a visibilidade da grelha definida a False.

4. Zoom do gráfico – definição do tipo de dados TZoom

Algo muito importante numa calculadora gráfica, e possivelmente aquilo que é mais complicado de aplicar, é o Zoom. O Zoom vai depender não só das definições dos eixos mas também das dimensões da área de trabalho. Caso a área de trabalho seja fixa, esta questão ganha contornos mais simples.

Contudo, como já foi feito no **Passo 2**, a nossa *Plotter* tem dimensão variável, definida pelo programador que utilizar a classe, pelo que o nosso trabalho não é simplificado.

Contudo, para já, vamos apenas definir a propriedade Zoom da classe e, para tal, criar um novo tipo de dados que contenha os dados relativos a esta característica da calculadora gráfica.

Visto que a redefinição do Zoom implica uma alteração do gráfico, um procedimento à parte será criado para esta redefinição mais adiante.

```
PRIVATE
  vZoom : TZoom;
PUBLIC
  property Zoom : TZoom read vZoom;
```

Ora, temos de definir o tipo de dados **TZoom**. A **Figura 6** representa os dados que devem estar presentes neste.

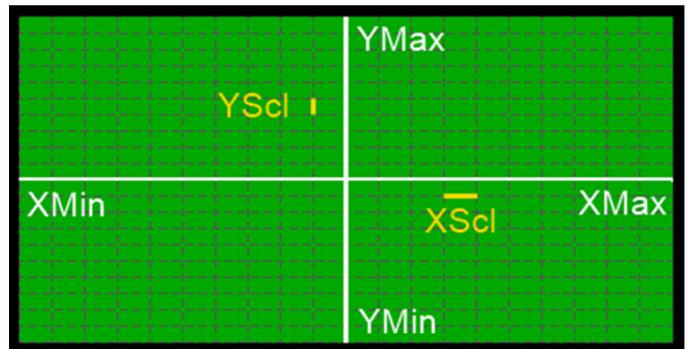


Figura 6 – Definições do Zoom.

Com base nesta definição de Zoom, podemos definir o tipo **TZoom** do seguinte modo:

```
TYPE
  TZoom = PACKED RECORD
    XMin, XMax, XScI : real;
    YMin, YMax, YScI : real;
  END;
```

5. Cálculo do centro do gráfico e da distância em pixéis entre cada linha da grelha

Eis algo que o leitor possivelmente não estava à espera. Contudo, antes de podermos avançar mais, é de extrema importância definir e calcular qual é o centro do gráfico e qual é a distância, em pixéis, entre cada linha da grelha em ambos os eixos.

Como se verá mais adiante, isto será importante para o desenhar da *Plotter* e para o cálculo dos pontos da função nesta.

Para tornar o procedimento de desenho da *Plotter* e do gráfico em si mais fácil, vamos calcular tudo em função da posição e do tamanho da área de trabalho – ou seja, determinamos em que pixéis da área gráfica produzida pelo **InitGraph** deverá aparecer cada ponto e cada linha.

Começemos pelo cálculo da distância entre as linhas da grelha. Este cálculo é independente da posição da *Plotter*, mas depende directamente da dimensão e do zoom:

- Mantendo o zoom constante, à medida que a dimensão aumenta, a distância entre as linhas da grelha também aumenta.

Para cada um dos eixos, a distância entre linhas da grelha é dada por:

$$\text{Distância} = \text{Zoom.Escala} \times \frac{\text{Dimensão}}{\text{Zoom.Máximo} - \text{Zoom.Mínimo}}$$

Por exemplo, para o eixo XX a fórmula será:

$$XScI \times \frac{\text{Size.cx}}{XMax - XMin}$$

Como este cálculo é interno e será necessário para alguns métodos (bem como o cálculo do centro), vamos declará-lo como um procedimento privado. E como há uma distância para cada um dos eixos, vamos aproveitar a estrutura do **TPoint** para o gravar:

```
PRIVATE
  DistBetweenGrid : TPoint;
  procedure CalcDistGridAndCenter;
```

Implementando este procedimento segundo a fórmula infra-apresentada:

```
PROCEDURE TPlotter.CalcDistGridAndCenter;
begin
  // eixo XX:
  self.DistBetweenGrid.X :=
    Round(self.Zoom.XSc1 *
      (self.Size.cx / (self.Zoom.XMax -
        self.Zoom.XMin)));
  // eixo YY:
  self.DistBetweenGrid.Y :=
    Round(self.Zoom.YSc1 *
      (self.Size.cy / (self.Zoom.YMax -
        self.Zoom.YMin)));
end;
```

Falta acrescentar a este procedimento o cálculo do centro. O centro depende do Zoom, e poderá não ser necessário calculá-lo:

- Se o Zoom estiver definido tal que a janela esteja toda na área positiva do eixo XX (por exemplo, **XMin** vale 2 e **XMax** vale 8), então o centro estará fora da área e pode-se definir a coordenada X do centro como sendo a posição X da *Plotter*.

Tendo em conta esta condição, e sabendo que o centro pode ser calculado segundo a fórmula que se segue, resta-nos implementá-lo em Pascal.

$$\text{Centro} = \text{Posição} + |\text{Zoom.Mínimo}| \times \frac{\text{Distância entre linha grelha}}{\text{Zoom.Escala}}$$

```
// centro em X:
if (self.Zoom.XMin < 0) and
  (self.Zoom.XMax > 0) then begin
  self.vCenter.X :=
    round(self.Position.X +
      abs(self.Zoom.XMin) *
      (self.DistBetweenGrid.X /
        self.Zoom.XSc1));
end else begin
  if (self.Zoom.XMin < 0) and
    (self.Zoom.XMax < 0) then
    self.vCenter.X :=
      self.Position.X +
      self.Size.cx
  else self.vCenter.X :=
```

```
self.Position.X;
end;

// centro em Y:
if (self.Zoom.YMin < 0) and
  (self.Zoom.YMax > 0) then begin
  self.vCenter.Y :=
    round(self.Position.Y +
      abs(self.Zoom.YMax) *
      (self.DistBetweenGrid.Y /
        self.Zoom.YSc1));
end else begin
  if (self.Zoom.YMin < 0) and
    (self.Zoom.YMax < 0) then
    self.vCenter.Y :=
      self.Position.Y +
      self.Size.cy
  else self.vCenter.Y :=
    self.Position.Y;
end;
```

Como se pode verificar, o cálculo do centro deve ser feito *a posteriori* já que este depende da distância em pixels entre as linhas da grelha previamente calculada.

6. Constantes necessárias

Agora que a base em termos de propriedades e valores necessários está feita, e antes de se avançar para a criação dos construtores, podemos definir as constantes discutidas aquando a definição teórica da classe.

Recordando, as constantes serão as seguintes:

```
CONST
  // Zoom:
  ZoomStandard : TZoom =
    (XMin:-10; XMax:10; XSc1:1;
     YMin:-10; YMax:10; YSc1:1);
  // Grelha:
  DefaultGrid : TLineStyle =
    (Color:DarkGray; Thickness:NormWidth;
     Style:DashedLn; Visible:True);
  // Eixos:
  DefaultAxes : TLineStyle =
    (Color:White; Thickness:ThickWidth;
     Style:SolidLn; Visible:True);
  // Linha de função:
  DefaultFunctionLine : TLineStyle =
    (Color:Red; Thickness:ThickWidth;
     Style:SolidLn; Visible:True);
  // Cor de fundo:
  DefaultPlotterBackColor : word = Green;
```

Estes valores podem ser alterados ao gosto do leitor:

- **Zoom padrão:**
- **XMin = -10**
- **XMax = 10**

A PROGRAMAR

PASCAL – CONSTRUÇÃO DE UMA CALCULADORA GRÁFICA

- **XScI** = 1
- **YMin** = -10
- **YMax** = 10
- **YScI** = -1

Estilo padrão da grelha:

- **Tipo de linha** – tracejada (*dashed*)
- **Espessura** – normal (1 pixel)
- **Cor** – cinzento escuro
- **Visível** – sim

Estilo padrão dos eixos:

- **Tipo de linha** – contínua (*solid*)
- **Espessura** – espessa (2 pixéis)
- **Cor** – branco
- **Visível** – sim

Estilo padrão da linha da função:

- **Tipo de linha** – contínua (*solid*)
- **Espessura** – espessa (1 pixéis)
- **Cor** – vermelho
- **Visível** – sim

Cor de fundo padrão – verde.

7. Construtores

O meu objectivo não vai ser criar uma panóplia de construtores com todos os *overloads* possíveis. Criei o construtor completo e, de seguida, criei uma versão muito mais simples. A partir destes será fácil criar mais construtores – basta recorrer directamente ao construtor completo.

O meu construtor completo irá receber 7 parâmetros, permitindo uma total personalização da instância da *Plotter a priori*. O construtor mais simples irá receber apenas os primeiros 2 parâmetros, que serão a posição e a dimensão.

```
PUBLIC
constructor Create
(cPosition : TPoint; cSize : TSize);
constructor Create
(cPosition : TPoint; cSize : TSize;
cZoom : TZoom;
cGrid, cAxes, cFnLine : TLineStyle;
cBkColor : word); overload;
```

A implementação do construtor em si é simples: simplesmente atribuem-se os parâmetros às respectivas propriedades, e fazem-se os cálculos do **Passo 5**:

```
CONSTRUCTOR TPlotter.Create
(cPosition : TPoint; cSize : TSize;
cZoom : TZoom;
cGrid, cAxes, cFnLine : TLineStyle;
cBkColor : word); OVERLOAD;
begin
self.vPosition := cPosition;
self.vSize := cSize;
self.vZoom := cZoom;
self.Grid := cGrid;
self.Axes := cAxes;
self.FunctionLine := cFnLine;
self.vFunction := nil;
self.BackColor := cBkColor;
self.CalcDistGridAndCenter;
end;
```

O construtor mais simples ir-se-á servir do construtor completo, e irá atribuir aos restantes 5 parâmetros os valores padrão definidos nas constantes:

```
CONSTRUCTOR TPlotter.Create
(cPosition : TPoint; cSize : TSize);
begin
self.Create(cPosition, cSize,
ZoomStandard, DefaultGrid,
DefaultAxes, DefaultFunctionLine,
DefaultPlotterBackColor);
end;
```

Não serão criados destrutores já que, para esta classe, bastará utilizar o **Free**, visto que esta herda da classe genérica **TObject**.

8. Método Show – desenho da Plotter

Após a criação dos construtores começa a fase de desenho da *Plotter* na área gráfica e a criação dos métodos que redefinem as propriedades *ReadOnly*, já previamente criadas. E é a partir desta fase que começa realmente o uso da *unit Graph*.

Para mostrar a *Plotter* vamos utilizar um procedimento denominado **Show**:

```
PUBLIC
procedure Show;
```

Primeiramente temos de definir o modo de preenchimento e, de seguida, devemos “pintar” a área de trabalho da *Plotter* com a cor de fundo definida aquando a criação da instância da classe:

```
PROCEDURE TPlotter.Show;
begin
SetFillStyle(SolidFill, self.BackColor);
Bar(self.Position.X, self.Position.Y,
self.Position.X + self.Size.cx,
self.Position.Y + self.Size.cy);
```

Constatamos dois novos métodos que pertencem à *unit Graph*:

Método & sintaxe	Descrição
SetFillStyle (estilo, cor);	Define o estilo de preenchimento de todos os métodos da família. Estilo – do tipo <i>Word</i> , define o tipo de preenchimento, ou seja, qual o padrão de preenchimento; Cor – do tipo <i>Word</i> , define qual é a cor de fundo.
Bar (X1, Y1, X2, Y2);	Preenche um rectângulo com o estilo definido pelo SetFillStyle . (X1,Y1) – canto superior esquerdo do rectângulo; (X2,Y2) – canto inferior direito do rectângulo.

No método **SetFillStyle** recorreu-se a uma das 200 constantes da *unit Graph* – **SolidFill** – que define um preenchimento sólido e simples.

De seguida deve-se desenhar a grelha e só depois os eixos, isto porque, se a grelha for desenhada após os eixos, esta ir-se-á sobrepor aos eixos, e a intenção dos eixos é estarem sobrepostos à grelha.

Para desenhar a grelha necessitamos de uma estrutura de repetição, ou seja, um ciclo. Para tal, iremos utilizar um ciclo **Repeat Until**.

```
if self.Grid.Visible then begin
// linhas verticais:
SetColor(self.Grid.Color);
SetLineStyle(self.Grid.Style, 0,
self.Grid.Thickness);

counter := round(self.Zoom.XMin);
repeat
Line(self.Center.X + counter *
self.DistanceBetweenGrid.X,
self.Position.Y, self.Center.X +
counter * self.DistanceBetweenGrid.X,
self.Position.Y + Self.Size.cy);

Inc(counter, 1);
until counter >= self.Zoom.XMax;

// linhas horizontais:
counter := round(self.Zoom.YMin);
repeat
Line(self.Position.X,
self.Center.Y - counter *
self.DistanceBetweenGrid.Y,
self.Position.X + self.Size.cx,
self.Center.Y - counter *
self.DistanceBetweenGrid.Y);

Inc(counter, 1);
until counter >= self.Zoom.YMax;
end;
```

Mais uma série de métodos da *unit Graph* foram utilizados:

Método & sintaxe	Descrição
SetColor (cor);	Define a cor das linhas de todos os métodos da família. Cor – do tipo <i>Word</i> , define qual é a cor.
SetLineStyle (estilo, padrão, espessura);	Define o estilo das linhas de todos os métodos da família. Estilo – do tipo <i>Word</i> , define o estilo da linha; Padrão – regra geral é ignorado; Espessura – do tipo <i>Word</i> , define a espessura da linha.
Line (X1, Y1, X2, Y2);	Desenha uma linha entre dois pontos com o estilo definido pelo SetLineStyle . (X1,Y1) – primeiro ponto; (X2,Y2) – segundo ponto.

De referir que o contador utilizado, **counter**, é do tipo **Integer**.

Como se pode verificar, a distância entre as linhas da grelha é fundamental para se saber onde a linha da grelha a cada passo deve ser colocada – como a distância em pixels entre cada linha depende não só do *Zoom* mas também da dimensão da *Plotter*, coloca-se na prática a importância desta distância.

Por fim, deve-se desenhar os eixos que são colocados em função do centro do gráfico, também previamente determinado:

```
if self.Axes.Visible then begin
SetColor(self.Axes.Color);
SetLineStyle(self.Axes.Style, 0,
self.Axes.Thickness);
Line(self.Center.X,
self.Position.Y, self.Center.X,
self.Position.Y + self.Size.cy);
Line(self.Position.X,
self.Center.Y, self.Position.X +
self.Size.cx, self.Center.Y);
end;
```

De notar o facto da grelha e dos eixos só serem desenhados caso tenham a propriedade **Visible** definida a **True** – caso contrário, o seu desenho é ignorado.

9. Método **Clear** – apagar a *Plotter*

Agora que se sabe preencher um rectângulo com o método **Bar**, o princípio do método **Clear** é simples: apagar a área de trabalho com a cor de fundo da área gráfica.

```
PUBLIC
procedure Clear;
```

A PROGRAMAR

PASCAL – CONSTRUÇÃO DE UMA CALCULADORA GRÁFICA

```
PROCEDURE TPlotter.Clear;  
begin  
  SetFillStyle(EmptyFill, Black);  
  Bar(self.Position.X,  
    self.Position.Y,  
    self.Position.X + self.Size.cx,  
    self.Position.Y + self.Size.cy);  
end;
```

De referir que a constante **EmptyFill** utiliza a cor de fundo da área gráfica, pelo que, qualquer que esta seja, o resultado final será sempre uma área gráfica “limpa”. Por defeito, o segundo parâmetro deve ser recebido, e por convenção utiliza-se a constante **Black** para este caso.

10. Método *Refresh* – actualização

Este método, público, faz uma actualização da *Plotter*. Este método é útil na medida em que há propriedades que podem ser alteradas sem passar por métodos controlados, pelo que, para estas fazerem efeito, é necessário actualizar a *Plotter*.

```
PUBLIC  
  procedure Refresh;
```

Este método simplesmente limpa a actual *Plotter* e desenha-a toda do zero:

```
PROCEDURE TPlotter.Refresh;  
begin  
  self.Clear;  
  self.Show;  
end;
```

11. Método *Relocate* – relocalização

A partir deste momento, com os métodos *Show*, *Clear* e *Refresh* à disposição e prontos a usar, torna-se deveras fácil modificar as propriedades da *Plotter* e, por conseguinte, modificar o desenho desta que está na área gráfica:

- 1) Limpa-se o desenho da *Plotter* actual;
- 2) Muda-se a propriedade desejada;
- 3) Volta-se a desenhar a *Plotter*, já com as propriedades desejadas.

Para algumas propriedades a limpeza pode ser efectuada mesmo após a mudança da propriedade, pelo que o método *Refresh* pode ser utilizado para otimizar o código. Contudo, outras propriedades, como a **posição**, **não** podem ter este tratamento óptimo: se mudarmos a posição antes de limpar a área gráfica, o método *Clear* vai efectuar a sua acção já com a nova propriedade de posição, pelo que parte ou a totalidade do desenho da *Plotter* na antiga posição não é limpa. Desta forma, primeiro deve ser executado o *Clear* e só alterada a propriedade para que aquele realize a sua acção ainda na posição inicial.

Para começar vamos criar o método ***Relocate***, público, que irá relocalizar a *Plotter* numa nova posição. Como foi

agora mesmo discutido, o *Refresh* não pode ser utilizado:

```
PUBLIC  
  procedure Relocate(cNewPosition:TPoint);  
  procedure Relocate(cX,cY:LongInt);  
    overload;
```

Para facilitar o trabalho do programador, criamos um *overload* do método principal em que se passam por parâmetro as coordenadas da nova posição ao invés de se passar as coordenadas com o tipo **TPoint**.

Implementando um destes métodos, o outro pode-se basear nesse:

```
PROCEDURE TPlotter.Relocate  
  (cNewPosition : TPoint);  
begin  
  self.Clear;  
  self.vPosition := cNewPosition;  
  self.Show;  
end;
```

```
PROCEDURE TPlotter.Relocate  
  (cX, cY : LongInt); OVERLOAD;  
var NewPos : TPoint;  
begin  
  NewPos.X := cX;  
  NewPos.Y := cY;  
  self.Relocate(NewPos);  
end;
```

12. Métodos para a função

Se modificarmos o *Zoom* ou a dimensão da *Plotter*, as coordenadas dos pontos da função vão mudar, pelo que, ao alterar estas propriedades, temos de recalculer estes pontos. Por isso, antes de implementar os métodos *Resize* e *ReZoom*, devemos implementar os métodos que irão manipular a função:

Receber (carregar) a função: ***LoadFunction***;

Desenhar a função: ***DrawFunction***;

Limpar a função da *Plotter*: ***ClearFunction***;

Libertar os recursos da função: ***DisposeFunction***.

A nossa intenção é que a *Plotter* receba uma função real e de variável real por referência (ou seja, por **apontador**), pelo que a nossa classe deverá ter uma variável, de carácter privado, que guarde qual é a função com que vai trabalhar:

```
PRIVATE  
  vFunction : TRealFunction;  
  vValuesOfFunction :  
    array [0..2000] of real;
```

Denota-se de imediato que os valores da função vão ser guardados num *Array*. Mais à frente veremos como este será

A PROGRAMAR

PASCAL – CONSTRUÇÃO DE UMA CALCULADORA GRÁFICA

utilizado.

Um novo tipo de dados foi aqui introduzido, pelo que o vamos definir. Tratando-se esta uma *Plotter* que trabalha com funções reais de variável real, então estas devem ter um parâmetro real e devolver um valor real:

```
TYPE
  TRealFunction=FUNCTION(x:real):real;
```

Primeiramente necessitamos de carregar a função (**LoadFunction**) e, neste processo, calcular os pontos na *Plotter*. Por uma questão de optimização, só vamos calcular os pontos estritamente necessários e não todos os 2001 pontos que o *Array* coloca ao nossos dispor.

```
PUBLIC
  procedure LoadFunction
    (fn:TRealFunction);
```

Este método deve:

- Gravar a função;
- Calcular os pontos da função. Para tal:
 - Calcular o valor real da função através da abcissa;
 - Transformar este valor numa coordenada Y a ser gravada em **vValuesOfFunction**;

Para este processo, o cálculo deve entrar com o Zoom, a dimensão e o centro do gráfico, para além do valor real da função a cada X. Implementando estes princípios, deveremos chegar ao seguinte código:

```
PROCEDURE TPlotter.LoadFunction
  (fn : TRealFunction);
var abcissa : real;
    counter : integer;
    step : real;
begin
  // grava função:
  self.vFunction := fn;
  // inicia contadores:
  counter := self.Position.X;
  abcissa := self.Zoom.XMin;
  // define passo de incrementação:
  step := self.Zoom.XSc1 /
    self.DistanceBetweenGrid.X;

  repeat
    // cálculo do ponto:
    self.vValuesOfFunction[counter] :=
      self.vCenter.Y -
      self.vFunction(abcissa) *
      (self.DistanceBetweenGrid.Y /
        self.Zoom.YSc1);
    // incrementa contadores:
    abcissa := abcissa + step;
```

```
Inc(counter, 1);
until counter >= self.Position.X +
  self.Size.cx;
end;
```

Os dois contadores são os seguintes:

- counter – define em que índice do Array **vValuesOfFunction** deve ser gravado o valor calculado;
- abcissa – define o valor da abcissa a cada pixel da área de trabalho da *Plotter*.

Para determinar o ponto da função na *Plotter*, sigamos o seguinte raciocínio:

- Tendo em conta o centro do gráfico, vamos andar para trás ou para a frente deste consoante o valor real da função: isto é definido através de uma subtração;

```
self.vCenter.Y - self.vFunction(abcissa)
```

- Contudo, o valor real da função deve ser multiplicado por um factor que nos indique a verdadeira posição, em pixels, do ponto. Assim sendo, se a distância entre linhas da grelha está para a escala, então 1 pixel está para o factor que procuramos:

$$\begin{array}{l} \text{Distância} \rightarrow \text{Escala} \\ 1 \rightarrow \text{Factor} \end{array}$$

Este factor é, então:

```
self.DistanceBetweenGrid.Y / self.Zoom.YSc1
```

E, assim, temos o cálculo do ponto da função na *Plotter*.

Depois de carregar a função, obviamente queremos que esta seja desenhada, ou representada (**DrawFunction**).

```
PUBLIC
  procedure DrawFunction;
```

A função só será desenhada se a propriedade **FunctionLine.Visible** estiver definida a **True**, e só será desenhada a parte do gráfico que estiver dentro da área da *Plotter*, pelo que este controlo será feito dentro do próprio ciclo que irá desenhar a função, e esta análise terá de ser, obviamente, ponto a ponto. A vantagem de ter calculado os pontos aquando o **LoadFunction** reside no facto de se evitarem mais cálculos no desenho da função, bastando fazer o output das linhas que unem os pontos.

```
PROCEDURE TPlotter.DrawFunction;
var counter : integer;
begin
  if self.FunctionLine.Visible
  and not(self.vFunction=nil)
  then begin
    // define estilo da linha
```

A PROGRAMAR

PASCAL – CONSTRUÇÃO DE UMA CALCULADORA GRÁFICA

```
SetLineStyle(self.FunctionLine.Style,
  0, self.FunctionLine.Thickness);
// define cor
SetColor(self.FunctionLine.Color);
// desenha a função:
for counter := self.Position.X to
(self.Position.X + self.Size.cx - 2)
do begin
  {evita que o gráfico seja desenhado
  fora da área da Plotter}
  if not ((round(
self.vValuesOfFunction[counter+1]) <
self.Position.Y) or (round(
self.vValuesOfFunction[counter+1]) >
self.Position.Y + self.Size.cy))
  then begin
    //usa directamente vValuesOfFunction
    Line(counter, round(
      self.vValuesOfFunction[counter]),
      counter+1, round(
      self.vValuesOfFunction[counter+1]));
  end;
end;
end;
end;
```

Como será óbvio, grande parte deste código é o conjunto de todas as condições de controlo para o desenho da função dentro da área. O verdadeiro acto de desenhar a função está quase no fim, aquando a utilização do método **Line**.

De seguida, poderemos querer “limpar” a função (**ClearFunction**), ou seja, voltar a ter a *Plotter* sem qualquer função desenhada.

```
PUBLIC
procedure ClearFunction;
```

Para isto, definimos a visibilidade a **False**, fazemos o **Refresh**, e voltamos a definir a visibilidade a **True**. Voltamos a definir a **True** porque, caso o programador queira fazer **DrawFunction** mais tarde, teria de redefinir a visibilidade a **True** – a ordem é para limpar a função, e não mudar a sua visibilidade. Apenas a mudamos para tirar proveito dos métodos que já criámos até ao momento. Desta forma:

```
PROCEDURE TPlotter.ClearFunction;
begin
  self.FunctionLine.Visible:=False;
  self.Refresh;
  self.FunctionLine.Visible:=True;
end;
```

Por fim, vamos libertar os recursos ocupados pela função (**DisposeFunction**), deixando a *Plotter* sem qualquer função com que trabalhar.

```
PUBLIC
procedure DisposeFunction
  (cClearPlotter : boolean);
```

Visto que o tipo de dados de **vFunction** não é mais do que um apontador, definimos que esta vale **NIL**. Para libertar os recursos podemos dar a opção de o gráfico ser também apagado da área de trabalho ou não:

```
PROCEDURE TPlotter.DisposeFunction
  (cClearPlotter : boolean);
var counter : integer;
begin
  self.vFunction := nil;
  for counter:=0 to 2000 do
    self.vValuesOfFunction[counter]:=0;
  if cClearPlotter then
    self.ClearFunction;
end;
```

Desta forma, a *Plotter* fica pronta a receber uma nova função, tendo passado previamente por uma limpeza devida de todos os recursos que, de outra forma, o método **LoadFunction** não faria.

Temos então os principais métodos de manipulação da função criados, pelo que podemos continuar a construir os métodos que redefinem as propriedades gráficas da *Plotter*.

13. Método **Resize** – redimensionar a *Plotter*

Redimensionar a *Plotter* implica que a função seja redeseenhada, daí já termos tratado de todos os métodos relativos à manipulação da função.

O método **Resize** terá dois cabeçalhos:

- Recebe um parâmetro do tipo **TSize**;
- Recebe dois parâmetros – largura e altura – do tipo **LongInt**.

```
PUBLIC
procedure Resize(cNewSize:TSize);
procedure Resize(ccx, ccy:LongInt);
  overload;
```

Como a alteração da dimensão implica uma nova distância entre as linhas das grelhas, bem como a mudança do centro do gráfico, estes devem ser recalculados. Do mesmo modo, a função terá de se adaptar, pelo que a solução é recarregar a função fazendo um **LoadFunction** da função que a *Plotter* já tiver.

```
PROCEDURE TPlotter.Resize
  (cNewSize : TSize);
begin
  self.vSize := cNewSize;
  self.CalcDistGridAndCenter;
  if not(self.vFunction = nil) then
    self.LoadFunction(self.vFunction);
```

```
self.Refresh;  
end;
```

```
PROCEDURE TPlotter.Resize  
(ccx, ccy : LongInt); OVERLOAD;  
var NewSize : TSize;  
begin  
  NewSize.cx := ccx;
```

14. Método ReZoom – redefinir o zoom

Como o zoom necessita de 6 parâmetros, vamos criar apenas um método geral que recebe um parâmetro do tipo TZoom. Se se pretender criar um método que receba os 6 argumentos totais relativos aos 6 dados do zoom, bastará fazer o mesmo que os métodos Resize e Relocate.

```
PUBLIC  
procedure Rezoom(cNewZoom : TZoom);
```

O zoom, tal como a dimensão, obriga a um recálculo da distância entre as linhas da grelha e do centro do gráfico, bem como ao recálculo dos pontos da função.

```
PROCEDURE TPlotter.ReZoom  
(cNewZoom : TZoom);  
begin  
  self.vZoom := cNewZoom;  
  self.CalcDistGridAndCenter;  
  if not(self.vFunction = nil) then  
    self.LoadFunction(self.vFunction);  
  self.Refresh;  
end;
```

Criação de um pequeno programa de teste

De facto já terminámos a construção da nossa classe! Reunindo toda a informação e todo o código na unidade CalcGraph, ficamos com uma unit pronta-a-usar.

A classe criada pode ser melhorada, bem como a unidade. O intuito deste artigo não era criar uma classe “altamente profissional”, mas sim criar uma Plotter básica e com os métodos mais básicos. Esta classe está passível a sofrer bugs no seu uso pois não fizemos, por exemplo, o controlo dos valores do zoom (XMax deve ser maior do que XMin, por exemplo) nem de outros parâmetros. De modo a tornar a classe mais segura, o leitor poderá, e deverá, acrescentar-lhe condições e métodos de segurança e de controlo. Contudo, não se esqueça de deixar um certo nível de liberdade para o programador final.

Agora vamos testar a unidade criada. Não vou criar um programa que teste a fundo a unidade. Farei apenas um pequeno programa que cria e personaliza uma Plotter, carrega uma função, desenha-a e, depois, encerra o programa.

A função será, relembrando o início do artigo:

$$f(x) = x^2 - 8$$

Vamos alterar algumas propriedades da Plotter para termos um resultado diferente do das imagens iniciais.

Segue, portanto, o código do programa seguido de uma breve explicação.

```
PROGRAM Teste;  
uses Graph,  
  CalcGraph,  
  SysUtils,  
  Windows;  
  
FUNCTION f(x:real):real;  
begin  
  f := sqr(x)-8;  
end;  
  
CONST  
  InitPos:TPoint=(X:10; Y:10);  
  InitSize:TSize=(cx:400; cy:200);  
  PosZoom:TZoom =  
    (XMin:-4; XMax:8; XSc1:2;  
     YMin:-9; YMax:2; YSc1:1);  
VAR Driver, Modus : SmallInt;  
    Plot : TPlotter;  
  
BEGIN  
  TRY  
    TRY  
      // inicia área gráfica  
      DetectGraph(Driver, Modus);  
      InitGraph(Driver, Modus, '');  
      // cria Plotter  
      Plot:=  
        TPlotter.Create(InitPos,InitSize);  
      // altera propriedades  
      Plot.BackColor:=Blue;  
      Plot.Grid.Color:=Black;  
      Plot.Axes.Color:=Yellow;  
      Plot.Axes.Thickness:=NormWidth;  
      // desenha a Plotter  
      Plot.Show;  
      // carrega a função  
      Plot.LoadFunction(@f);  
      // altera cor da função  
      Plot.FunctionLine.Color:=LightRed;  
      // desenha função  
      Plot.DrawFunction;  
  
      readln; // pausa  
      Plot.Resize(700,400);  
      Plot.ReZoom(PosZoom);  
      Plot.DrawFunction;  
    EXCEPT  
      ON ex:exception do begin  
        MessageBox(0,@ex.message[1], 'Erro!',  
          MB_ICONASTERISK);  
      end;  
    END  
  FINALLY  
    write('FIM.');
```

A PROGRAMAR

PASCAL – CONSTRUÇÃO DE UMA CALCULADORA GRÁFICA

```
Plot.Free;  
CloseGraph;  
END;  
END.
```

Neste programa criámos uma *Plotter* através da variável *Plot*, ou seja, uma instância da classe *TPlotter*. Os seus valores padrão para as várias propriedades foram alterados:

- A cor de fundo foi definida a azul, em vez da predefinição (verde);
- A cor da grelha foi definida a preto e não a cinzento-escuro;
- A cor dos eixos passou a ser amarela, e a sua espessura passou a ser de 1 pixel;
- A cor da função passou para vermelho-vivo (LightRed) em vez de vermelho-escuro (Red).

Após a alteração destas propriedades, o resultado final (Figuras 7 e 8) é diferente do inicialmente representado na Figura 1, que seria o output gerado caso se mantivesse a predefinição.

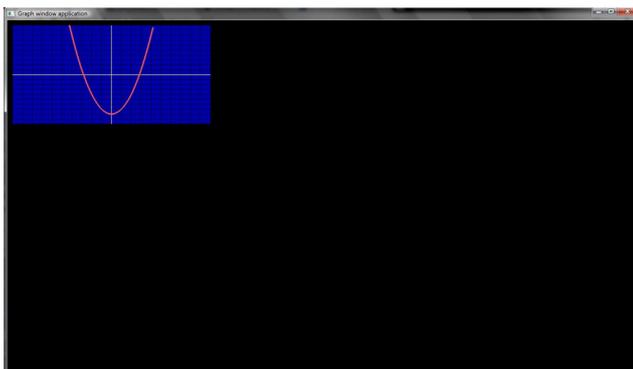


Figura 7 – Output gerado pelo programa de testes antes da pausa.

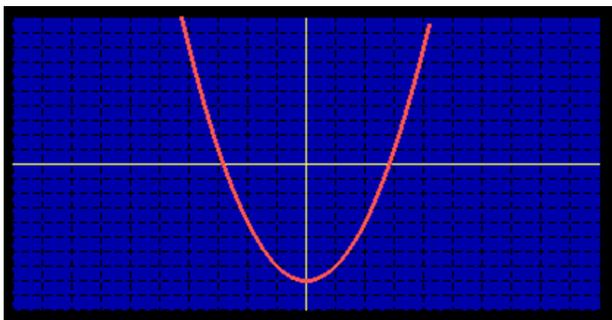


Figura 8 – Pormenor da *Plotter* criada no programa antes da pausa – de realçar as propriedades alteradas.

A Figura 7 mostra a importância da posição e do tamanho da *Plotter*: todo o fundo negro é a área gráfica gerada pelo *InitGraph* que, no caso desta figura, foi de 1280x800.

Contudo foi feita uma pausa no programa e depois foram aplicados os métodos *Resize* e *ReZoom*. Como estes métodos não redesenham a função, fazemos de novo o output desta, e obtemos o resultado representado pela Figura 9.

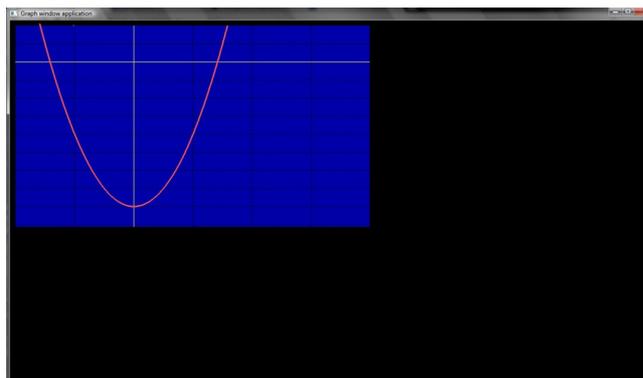


Figura 9 – Output gerado pelo programa de testes após a pausa.

Notamos a adaptação realizada por todos os cálculos auxiliares, privados, na alteração das propriedades de zoom e dimensão.

Resumo

Com este artigo recorremos à *unit Graph* para conceber uma *Plotter*, tendo a sua implementação sido feita com o uso do **OOP**, ou seja, o paradigma da Programação Orientada aos Objectos.

Desta forma será possível ter, numa só área gráfica, várias *Plotters* com características diferentes e a trabalhar com funções diferentes, mas totalmente independentes umas das outras – para tal, bastará criar diferentes instâncias da classe **TPlotter**.

Apesar de não se ter aprofundado a unidade na qual nos baseamos, ficou claro que com pouco se pode fazer muito, e com meia dúzia de métodos da **Graph** uma *Plotter* totalmente funcional e personalizável foi construída.

Destacam-se, então, os métodos **Bar**, **Line**, **SetFillStyle** e **SetLineStyle**, assim como as constantes **NormWidth**, **ThickWidth**, **EmptyFill**, **SolidFill**, **SolidLn** e **DashedLn**.

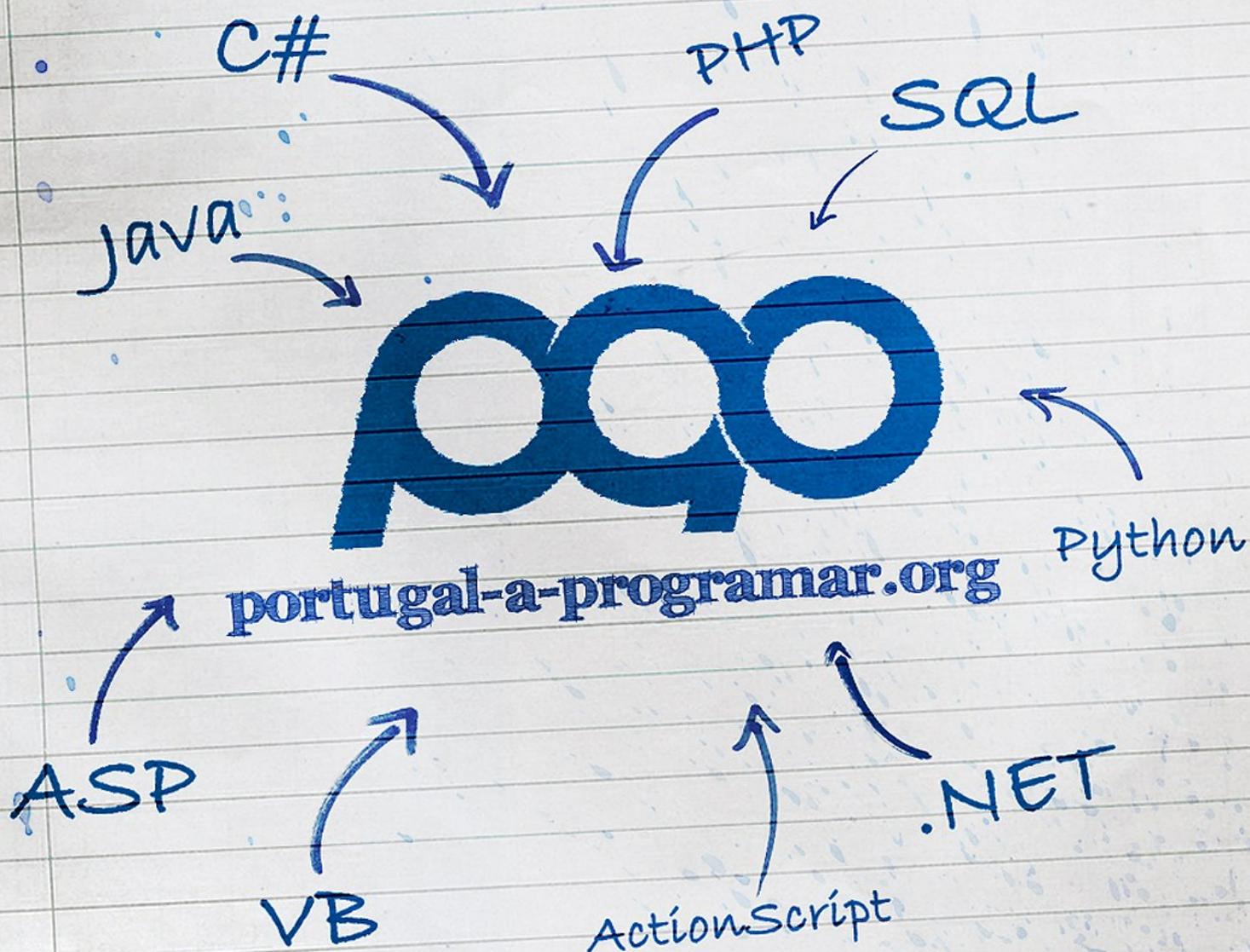
AUTOR



Escrito por Igor Nunes

Curioso na área das tecnologias e em especial da programação, tem um grande interesse pelas tecnologias de ensino da Texas Instruments™ e pelo TI-Basic, além de ter uma saudável relação com o Pascal. É conhecedor de outras linguagens de programação, como VB.NET. No P@P, é membro da Wiki Team e Moderador Local dos quadros de Pascal e Delphi/Lazarus. Escreveu o novo Tutorial de Pascal, [recentemente disponibilizado em PDF](#).

A maior comunidade portuguesa de programação!



Visite-nos!

Formulários em HTML5

HTML



HTML5 é a mais recente linguagem utilizada para a criação de páginas web. Infelizmente ainda não é suportada por todos os navegadores disponíveis no mercado. De momento, o Opera, é o único que consegue dar suporte a todos os novos atributos e elementos dos formulários que

o HTML5 nos trouxe.

Sem mais demoras, estão enumerados a baixo os novos elementos:

- **color** - mostra uma paleta de cores
- **date** - mostra no campo um calendário que permite navegar entre meses e anos
- **datetime** - configura data e hora
- **datetime-local** - data e hora local
- **email** - endereço de e-mail
- **month** - mostra no campo um calendário que permite navegar entre meses e anos
- **number** - são setas para aumentar e diminuir números, limitando número máximo e mínimo
- **range** - botão deslizante
- **search** - para pesquisas
- **tel** - números de telefones
- **time** - para mostrar a hora
- **url** - endereço de url
- **week** - mostra no campo um calendário que permite navegar entre meses e anos

A sua implementação é feita da seguinte forma:

```
<input type="elemento" name="whatever"/>
```

No elemento **range** e **number** já temos de adicionar dois campos: min e max para atribuir o valor mínimo e máximo do botão deslizante no caso do **range** e setas para aumentar e diminuir números, limitando o seu máximo e mínimo no caso

do **number**:

```
<input type="range" name="point" min="1" max="10"/>
```

Após abordar os elementos, chegou a altura de falar nos novos atributos:

- **Autocomplete** - se colocado como "on", ele habilita o "autocompletar" nos campos.
- **Novalidate** - é um atributo booleano. Quando se encontra ativo, especifica que o formulário não deve ser validado quando submetidos.
- **Autofocus** - Foca o campo no carregamento da página.
- **Formaction** - Substitui o atributo forma de ação. Deve ser uma URL válida que define para onde enviar os dados quando o formulário é submetido.
- **Formmethod** - Substitui atributo do formulário método. Define o método HTTP para o envio de dados para o URL de ação. Aceita "get", "post", "put" e "delete".
- **Formnovalidate** - Substitui atributo "novalidate" no formulário. Se for colocado como "true", esse campo não será validado quando for submetido.
- **Formtarget** - Aceita "blank", "_self", "_parent" e "_top".
- **Width** e **Height** - Definem largura e altura do campo do formulário, respetivamente. Podem-se utilizar medidas em percentagem e pixels.
- **List** - Referencia um elemento DataList. Caso tenha um elemento com determinado ID, referencie esse ID.
- **Multiple** - Permite que o utilizador selecione mais de um item.
- **Pattern** - Permite criar restrições por expressões regulares.
- **Placeholder** - Define ajuda para o usuário preencher o campo com os dados corretos.
- **Step** - Utilizado juntamente com os tipos "datetime", "time", "date", "datetime-local", "month", "week", "number", "range", definindo um intervalo numérico

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<link rel="stylesheet" href="css/style.css" />
<title>Formulário HTML5</title>
</head>
<body>
<form id=payment onsubmit=
"return validateFormOnSubmit(this)" action="">
<fieldset id="1"><legend>Dados</legend>
<ol><li>
<label for=name>Name</label>
<input id=name name=name type=text
placeholder="Primeiro e Ultimo Nome" required
autofocus>
</li><li>
<label for=email>Email</label>
<input id=email name=email type=email
placeholder="example@gmail.com" required>
</li><li>
<label for=phone>Telefone</label>
<input id=phone name=phone type=tel
placeholder="+351987654321" required>
</li></ol>
</fieldset>
<fieldset id="2"><legend>
Morada de Entrega</legend>
<ol><li>
<label for=address>Morada</label>
<textarea id=address name=address rows=5
required placeholder="Insira a Sua Morada">
</textarea>
<p></p></li><li>
<label for=postcode>Código Postal</label>
<input id=postcode name=postcode type=text
required placeholder="0000-000 Localidade">
</li><li>
<label for=country>País</label>
<input id=country name=country type=text required>
</li></ol>
</fieldset>
```

Nesta parte deixo o código de exemplo dum pequeno formulário em HTML5, incorporando alguns atributos e elementos abordados na pagina anterior. Esta página contem o código HTML.

No head definimos o charset que é a codificação dos caracteres, indicamos qual o titulo para a nossa página e carregamos a folha de estilo CSS.

A quão do envio do formulário, este será validado por JavaScript, na função **validateFormOnSubmit**.

Fieldset é a tag utilizada para agrupar elementos relacionados dentro duma form.

O **placeholder** faz com que apareça um texto "predefinido" antes de se preencher o campo. **Autofocus** faz com que o campo seja focado com o cursor, pronto para escrever. O **required**, a quão da validação do formulário, faz com que só deixe validar caso o campo esteja preenchido.

Legend especifica qual o subtítulo do fieldset.

O **label for** indica qual o elemento a que a label está ligada.

```
<fieldset id="3"><legend>Detalhes Cartão</legend>
<ol><li>
<fieldset><legend>Tipo Cartão</legend>
<ol><li>
<input id=visa name=cardtype type=radio>
<label for=visa>VISA</label>
</li><li>
<input id=paypal name=cardtype type=radio>
<label for=paypal>PayPal</label>
</li><li>
<input id=mastercard name=cardtype type=radio>
<label for=mastercard>Mastercard</label>
</li></ol>
</fieldset>
</li><li>
<label for=cardnumber>Numero Cartão</label>
<input id=cardnumber name=cardnumber type=
number required>
</li><li>
<label for=secure>Código Pessoal</label>
<input id=secure name=secure type=number
required>
</li><li>
<label for=namecard>Nome do Cartão</label>
<input id=namecard name=namecard type=text
placeholder="Nome igual ao do Cartão" required>
</li></ol>
</fieldset>
<fieldset>
<button type=submit>Enviar</button>
</fieldset>
</form>
</body>
</html>
```

Ao definir **type=radio** em cada input da list, fazemos com que só seja possível selecionar um, neste caso, um tipo de cartão para o pagamento.

A PROGRAMAR

FORMULÁRIOS HTML5

Numero Cartão	<input type="text" value="12345678"/>
Código Pessoal	<input type="text" value="1234"/>

Com a utilização do elemento **number** obtemos setas laterais para o incremento do valor preenchido na label.

The last but not the least, temos o nosso submit **button**. Antigamente declarado como *input type=button*.

Chegou agora a parte de tratar-mos da CSS para termos o nosso formulário mais bonitinho. 2 paginas a seguir esta demonstrada a diferença.

```
@charset "utf-8";
/* CSS Document - style.css*/
html, body, h1, form, fieldset, legend, ol, li
{margin: 0;padding: 0;}
body {margin-left: auto;margin-right:auto; width:
440px;background: #ffffff;color: #111111;font-
family: Georgia, "Times New Roman", Times, se-
rif;padding: 20px;}
form#payment {background: #9cbc2c;-moz-border-
radius: 5px;-webkit-border-radius: 5px;border-
radius: 5px;padding: 20px;width: 400px;}
form#payment fieldset {border: none;margin-
bottom: 10px;}
form#payment fieldset:last-of-type {margin-
bottom: 0;}
form#payment legend {color: #384313;font-size:
16px;font-weight: bold;padding-bottom: 10px;text-
shadow: 0 1px 1px #c0d576;}
form#payment fieldset fieldset legend {color:
#111111;font-size: 13px;font-weight: nor-
mal;padding-bottom: 0;}
form#payment ol li {background:
#b9cf6a;background: rgba(255,255,255,.3);border-
color: #e3ebc3;border-color: rgba
(255,255,255,.6);border-style: solid;border-
width: 2px;-moz-border-radius: 5px;-webkit-border-
-radius: 5px;border-radius: 5px;line-height:
30px;list-style: none;padding: 5px 10px;margin-
bottom: 2px;}
form#payment ol ol li {background: none;border:
none;float: left;}
form#payment label {float: left;font-size:
13px;width: 110px;}
form#payment fieldset fieldset label {background:
none no-repeat left 50%;line-height:
20px;padding: 0 0 30px;width: auto;}
form#payment label[for=visa] {background-image:
url(..img/visa-icon.png);}
form#payment label[for=paypal] {background-image:
url(..img/paypal-icon.png);}
form#payment label[for=mastercard] {background-
image: url(..img/mastercard-icon.png);}
form#payment fieldset fieldset label:hover
{cursor: pointer;}
```

Começamos por declarar o **charset**, já bordado anteriormen- te o seu significado na pagina HTML.

Declaramos que não existe **margin**, que é o espaço a volta do elemento e o **padding** que é o espaço dentro da margin a volta do conteúdo.

Dentro do **body** damos margem a esquerda e à direita auto- máticas, definimos a largura, fonte do texto e damos um pad- ding de 20px.

O nosso formulário tem uma **border-radius** que são os can- tos redondos em torno dele, neste caso de 5px.

Dados

Aqui é declarado o **texto-shadow** que aplica uma sombra por baixo do texto que é afetado, uma propriedade do CSS3. O primeiro valor corresponde a posição para a direita, o se- gundo a posição para baixo, o terceiro é a "flocosidade" ou *fuzziness* e por ultimo temos a cor da sombra.

```
list-style-type: none • list-style-type: disc
```

List-style especifica o tipo de marcador utilizado antes de cada elemento da lista.

O **float** serve para alinhar horizontalmente um elemento, neste caso, a esquerda (left). Também pode ser centrado (center), alinhado a direita (right), herdar (inherit) ou não ser alinhado (none).

Line-height é utilizado para especificar a altura da linha.

Background-image define uma ou mais imagens para o elemento.

Quando o rato para por cima dum certo elemento, e quere- mos mudar o cursor, utilizamos **cursor:pointer** para mudar o tipo de cursor a ser mostrado, neste caso o



Quando o rato para por cima dum certo elemento, e queremos mudar o cursor, utilizamos **cursor:pointer** para mudar o tipo de cursor a ser mostrado, neste caso o apontador.

```
form#payment input:not([type=radio]), form#payment
textarea {background: #ffffff;border: none;-moz-
border-radius: 3px;-webkit-border-radius: 3px;-
khtml-border-radius: 3px;border-radius: 3px;font:
13px Georgia, "Times New Roman", Times, se-
```

A PROGRAMAR

FORMULÁRIOS HTML5

```
rif;outline: none;padding: 10px;width: 200px;}
form#payment input:not([type=submit]):focus,
form#payment textarea:focus {background:
#eaeaea;}
form#payment input[type=radio] {float:
left;margin-right: 5px;}
form#payment button {background: #384313;border:
none;-moz-border-radius: 20px; -webkit-border-
radius: 20px;khtml-border-radius: 20px;border-
radius: 20px;color: #ffffff;display: block;font:
18px Georgia, "Times New Roman", Times, se-
rif;letter-spacing: 1px;padding: 7px 25px;text-
shadow: 0 1px 1px #000000;text-transform: upper-
case;width: 126px;margin-left: auto;margin-right:
auto;}
form#payment button:hover {background:
#1e2506;cursor: pointer;}
```

Pseudo-classe **:not** engloba todos os outros elementos fora daquele input, neste caso *radio*, lhes sejam atribuídos os valores declarados.

Focus é utilizado quando um elemento esta a ser alvo duma inserção por parte do utilizador.

Display define como um elemento de HTML deve ser mostrado, neste caso em bloco para ter espaço em cima e por baixo dele.

Text-transform transforma todos os caracteres introduzidos, neste caso em maiúsculas (uppercase).

Validação JavaScript

```
function validateFormOnSubmit(theForm) {
    var reason = "";
    reason += validateEmail(theForm.email);
    reason += validatePhone(theForm.phone);
    reason += validateEmpty(theForm.from);
    if (reason != "") {
        alert("São necessárias correções:\n" +
            reason);
        return false;
    }
    return true;
}
function validateEmail(fld) {
    var error = "";
    var tfld = trim(fld.value);
    var emailFilter = /^[^@]+@[^@.]+\.[^@]
        *w\w$/;
    var illegalChars = /[\(\)
        \<\>\,;\:\\\\"\\\]/;
    if (fld.value == "") {
        fld.style.background = 'Yellow';
    }
}
```

```
error = "Não introduziu um email.\n";
} else if (!emailFilter.test(tfld)) {
    fld.style.background = 'Yellow';
    error = "Introduza um email válido.\n";
} else if (fld.value.match(illegalChars)) {
    fld.style.background = 'Yellow';
    error = "O endereço de email contem caracte-
res inválidos.\n";
} else {
    fld.style.background = 'White';
}
return error;
}
```

Posto tudo o que é visível para o utilizador, temos também de fazer umas validações por trás dos dados inseridos. Neste caso vou dar um pequeno exemplo para validação de email, numero de telefone e se existem campos por preencher. Nestas 2 paginas está demonstrado o código com uma breve explicação das funcionalidades.

Declarada a variável **reason** que guardará todos os erros das funções de validação do formulário.

“Reason += validate...” acrescenta os erros relatados.

Caso o **reason** tenha algum conteúdo, mostra os erros e não deixa a validação prosseguir.

Caso não tenha conteúdo, a validação está terminada e o formulário pode prosseguir.

Validação do email

Declarada a variável **error** que guarda os erros encontrados.

trim(fld.value) retira os espaços em branco presentes.

emailFilter declara quais os caracteres possíveis para um email.

illegalChars declara quais os caracteres inválidos do email.

Neste exemplo, o background do fiel é tornado amarelo para se perceber melhor onde está o erro

Se estiver a branco, guarda o erro que não introduziu um email.

Testa se contem caracteres para além dos possíveis declarados.

Caso encontre um caracter ilegal, emite a mensagem que é guardada em **error**.

A PROGRAMAR

FORMULÁRIOS HTML5

Se não encontrar erros, deixa o fundo da field em branco.

Se encontrar erros, retorna-os onde serão visíveis através do **reason**.

```
function validatePhone(fld) {
    var error = "";
    var stripped = fld.value.replace(/[\\(\\)\\.\\-\\ ]/g, '');

    if (fld.value == "") {
        error = "Não introduziu um numero de
                telefone.\\n";
        fld.style.background = 'Yellow';
    } else if (isNaN(parseInt(stripped))) {
        error = "O telefone contem caracteres
                ilegais.\\n";
        fld.style.background = 'Yellow';
    } else if (!(stripped.length == 6
        || stripped.length == 7)) {
        error = "O telefone tem o tamanho errado.
                Verifique se introduziu o indicativo.\\n";
        fld.style.background = 'Yellow';
    }
    return error;
}

function validateEmpty(fld) {
    var error = "";
    if (fld.value.length == 0) {
        fld.style.background = 'Yellow';
        error = "O espaço requerido não tem
                conteúdo.\\n"
    } else {
        fld.style.background = 'White';
    }
    return error;
}
```

Validação do numero de telefone

Variável **stripped** - substitui quaisquer caracteres ilegais por um espaço em branco.

Se vazio, reporta o erro de não inserção de numero de telefone.

Colorização do background no field com erro.

Lugar onde os caracteres inválidos são validados. Caso contenha algum caracter inválido é emitido o erro.

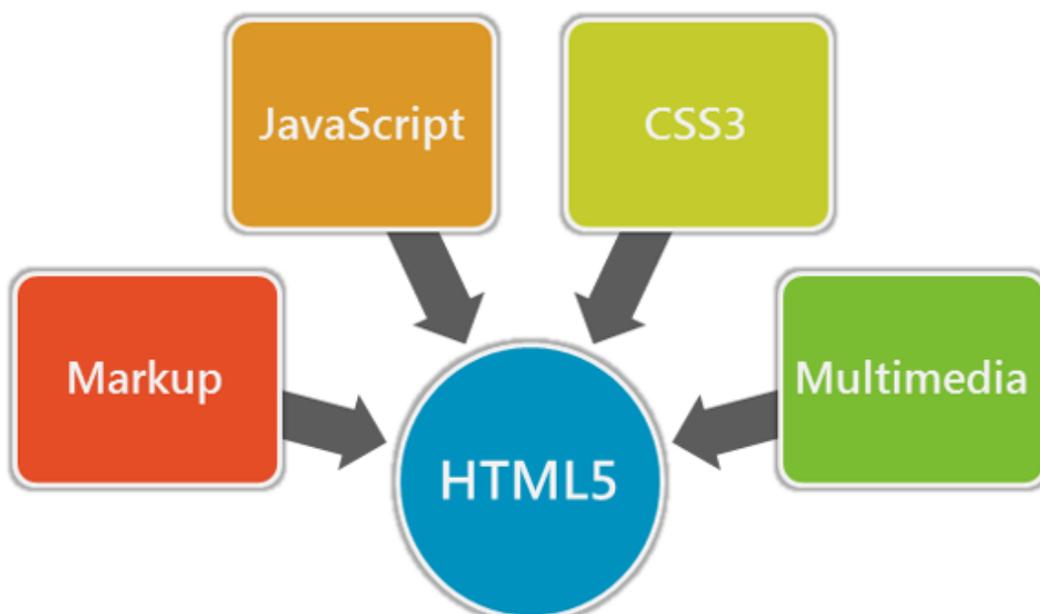
Caso o tamanho do numero de telefone seja de 6 ou 7 caracteres, é mencionado o possível esquecimento do indicativo da zona.

Quaisquer erros existentes no formulário são retornados ou a variável vai em branco caso não contenha erros.

Validação do preenchimento das fields

Verifica se o valor dentro da field tem um tamanho de 0 (zero) caracteres. Caso tenha, é reportado o erro que é necessário preencher esse campo .

Mais uma vez é devolvido o erro, caso existente. É necessário fazer o return no final de cada função de validação.



Dados

1. Name
2. Email
3. Telefone

Morada de Entrega

1. Morada
2. Código Postal
3. País

Detalhes Cartão

1. Tipo Cartão
 - 1. VISA
 - 2. PayPal
 - 3. Mastercard
2. Numero Cartão
3. Código Pessoal
4. Nome do Cartão

Dados

Name

Email

Telefone

Morada de Entrega

Morada

Código Postal

País

Detalhes Cartão

Tipo Cartão

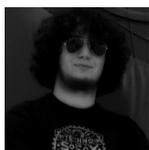
 VISA  PayPal  Mastercard

Numero Cartão

Código Pessoal

Nome do Cartão

AUTOR



Escrito por Nelson Belo Lima

Frequenta o CET Tecnologias de Programação de Sistemas de Informação na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Viana do Castelo. Antigo Técnico de Mecatrónica Automóvel e Mecânico Perito na Citroen, nunca perdeu a paixão pela informática e foi ao longo do tempo acompanhando de perto o seu desenvolvimento. Ultimamente mais dentro desta área tecnológica devido ao percurso escolar e extraescolar.

<http://www.portugal-a-programar.pt/user/25616-nelsonbelolima/>

SEO – Search Engine Optimization – Introdução parte II

Na edição n.º 34 da **Revista Programar**, que foi disponibilizada em Abril, mais concretamente na primeira parte desta Introdução ao SEO, fiz uma breve abordagem ao conceito de **Search Engine Optimization** com o intuito de desmistificar um assunto que para muitos ainda parece confuso, mas que aos poucos, e espero poder contribuir para esse efeito, vai sendo um tema cada vez mais claro e importante para todos os Webmasters/Webdesigners.

Na primeira parte já percebemos que os motores de busca na internet são grandes fontes de tráfego para os nossos sites e é tendo em conta essa importância que o SEO ganha força nos dias que correm. Para os leitores que ainda não tiveram a oportunidade de ler esse artigo, sugiro que o façam em primeiro lugar e só depois deverão ler este, tornando-se assim mais fácil de

entender o que irei falar hoje.

Esta segunda parte pretende encerrar a parte introdutória sobre o SEO e apresentar algumas das ferramentas e restantes conceitos que iremos aprofundar nos artigos seguintes desta série.



ENIGMAS DO C#: AVALIAÇÃO DA LISTA DE ARGUMENTOS

por Paulo Morgado

Enigma

Dado o seguinte método:

E os seguintes métodos:

```
private static void M(  
    int x = 10,  
    int y = 20,  
    int z = 30)  
{  
    Console.WriteLine(  
        "x={0}, y={1}, z={2}", x, y, z);  
}
```

```
int a = 0;  
  
M(++a, ++a, ++a);  
  
M(++a, z: ++a);
```

Qual é o resultado do seguinte código?

Veja a resposta e explicação na página 56

SEO – SEARCH ENGINE OPTIMIZATION – INTRODUÇÃO PARTE II

O que são os Bots, Spiders ou Crawlers

Não podemos falar em otimizar os nossos portais para os motores de busca sem primeiro conhecer os “bichinhos” que os visitam e por sua vez os indexam, atribuindo-lhes um ranking. Eles são dos mais importantes e críticos visitantes que os nossos sites recebem, pelo que lhes devemos dar o máximo de “atenção” se queremos ter sucesso nesta área. Estes Spiders, Bots e Crawlers (significam todos a mesma coisa) são mecanismos automatizados de software, que varrem a Internet (Web – teia – daí o termo spider: aranha) e catalogam os sites e páginas por onde vão passando. Nem todos os spiders são bons, pois alguns dos que respeitam menos as normas e protocolos, podem catalogar e indexar alguns conteúdos que nós não pretendíamos. Os bons crawlers são os que respeitam a norma REP – Robots Exclusion Protocol (veremos mais à frente nesta série o que é este protocolo e como tirar partido dele num site). De uma forma geral podemos perceber as passagens destes agentes pelos nossos sites através dos logs (relatórios) dos nossos Servidores Web.



Existem vários tipos de Web-spiders, que podemos conhecer mais em detalhe no endereço: <http://www.user-agents.org/>. Estes visitam os nossos sites de tempos a tempos, dependendo do algoritmo do motor de busca que os envia ao nosso site, assim como da relevância do nosso conteúdo. Eles não visitam um site apenas uma vez, mas vão voltando à medida que detectam alterações nele, ou novos conteúdos. Todo esse mecanismo pode ser definido por nós, como iremos aprender, ou ser deixado em modo “automático” e gerido de forma independente pelos motores de busca.

Uma das “aranhas” mais famosa da grande teia que é a internet é a “**Googlebot**”, a *Web-spider* da Google. A sua assí-

natura pode ser encontrada nos *logs* dos Servidores Web, através do *header* do protocolo HTTP:

```
GET / HTTP/1.1
Host: www.XXXXXXXXXX.pt
Connection: Keep-alive
Accept: */*
From: googlebot(at)googlebot.com
User-Agent: Mozilla/5.0 (compatible; Googlebot/2.1;
+HTTP://www.google.com/bot.html)
Accept-Encoding: gzip,deflate
```

Neste exemplo extraído de um *log file*, podemos visualizar que o **Googlebot** fez um pedido para indexar a raiz do nosso site e que ele suporta a compressão gzip, uma forma de otimizar o tráfego web, comprimindo os dados transferidos entre o Servidor Web e o cliente.

Do lado da concorrência ao Google, o *crawler* da **Yahoo** chama-se **Slurp** e o do motor de busca da Microsoft, o Bing, o seu *spider* apelida-se de **MSNBot**, tendo ambos comportamentos semelhantes, se bem que os algoritmos de indexação são diferentes.

Contudo a tecnologia dos *Web-spiders* não é perfeita. Algumas vezes estes não conseguem indexar todos os conteúdos de uma página ou site (chama-se a isso os **Search Engine Traps**, ou armadilhas), devido à tecnologia com que estes estão inseridos no site, como por exemplo ficheiros em Flash, JavaScript, texto em imagens etc. Iremos também alertar e aconselhar os leitores para estes problemas, que em muito podem afectar a optimização de um *website*, ao longo dos próximos artigos.

Contudo o campo de trabalho destes *bots* não é um campo infinitamente aberto. Pelo menos para os que respeitam o protocolo **REP**. Existe também a possibilidade de nós proibirmos o acesso dos *crawlers* ao nosso site, ou a determinadas zonas deste, como forma de impedirmos que determinados conteúdos que desejamos serem privados, surjam nos resultados dos motores de busca, algo muito comum nos dias de hoje e que infelizmente gera algumas surpresas desagradáveis, como por exemplo quando pesquisamos na Web e damos de caras com informação a nosso respeito, num resultado de um motor de busca, por este estar disponível num *website* e os seus *webmasters* não terem tido o cuidado de restringir o acesso dos motores de busca a esse conteúdo. Lembrem-se que até o conteúdo escrito de um ficheiro PDF, como por exemplo a pauta das notas de uma determinada Cadeira da Faculdade, pode ser indexado por um motor de busca, caso este possua permissões para tal, e posteriormente disponibilizado para toda a internet através dos **SERP's** (*search engine result pages*).

Como já deu para perceber, os *spiders* não indexam apenas ficheiros HTML, PHP, ASP etc., mas é já vasta a quantidade de tipos diferentes de ficheiros que os motores de busca

A PROGRAMAR

SEO – SEARCH ENGINE OPTIMIZATION – INTRODUÇÃO PARTE II

conseguem ler e indexar: PDF, DOC, XLS, PPT, e muitos outros tipos de ficheiros, são já facilmente indexáveis e os seus conteúdos percorridos pelos *bots* dos motores de busca.

Como os spiders lêem os nossos sites

Os Webspiders não são humanos confinados numa enorme sala cheia de computadores ligados à Internet e que catalogam manualmente os sites que vão visitando. Estes *bots* são mecanismos de *software* automatizados que varrem a internet, saltando de link em link, sendo a “visão” que eles têm de um site um pouco diferente da nossa.

De acordo com a ferramenta **Search Engine Spider Simulator**, um simulador disponibilizado gratuitamente no site www.webconfs.com, esta é por exemplo a visão (parcial – apenas uma parte do texto gerado) que um *spider* tem do site www.portugal-a-programar.org:

Spidered Text :

Portugal-a-Programar Ir para conteúdo Twitter Facebook Google + Entrar Registe-se Pesquisa Avançado Pesquisar secção: Fórum Membros Help Files Blog Downloads Calendário Ver Novo Conteúdo Blog Fórum Downloads Calendário Membros Revista PROGRAMAR Wiki Planeta IRC Mais Javascript Desactivado Tem actualmente o javascript desactivado. Diversas funcionalidades podem estar indisponíveis. Por favor active o javascript para ter acesso a todas as funcionalidades. Em Destaque Revista PROGRAMAR – Edição Nº 35 Junho 2012 Publicado por jpaolino em Revista PROGRAMAR, 08 Junho 2012 · 1401 visualizações Nesta edição continuaremos também a premiar os autores dos três melhores artigos, dado o sucesso nas edições anteriores. E os leitores devem dar a sua opinião para que possamos premiar correctamente. Para isso vote em http://tiny.cc/ProgramarED35_V Assim nesta edição trazemos até si, como... Ler o resto desta entrada → 1 Comentários Alternar Bem-vindos ao Portugal-a-Programar Bem-vindos ao Portugal-a-Programar...

Como percebemos facilmente, é apenas texto o que os spiders extraem dos sites. Todo o CSS, imagens e outras técnicas de embelezamento estético de um site são cortadas desta interpretação do nosso conteúdo. Onde nós encontramos caracteres especiais, isso não se torna um problema para o crawler, pois possuem algoritmos que os convertem em texto correcto, na medida em que as páginas HTML possuem no cabeçalho as indicações relativas à codificação do texto, que podem variar de país para país.

Rankings dos motores de busca

É aqui que o SEO entra um pouco no domínio do “esotérico”. Sabemos e lemos em tudo o que é livro sobre o tema, que os *rankings* são definidos através de algoritmos “ultra-secretos” e sofisticados das empresas proprietárias dos mo-

tores de busca. No entanto nunca conseguimos saber como é que essas fórmulas realmente funcionam, tornando-se difícil afinar o nosso site para atingir rankings mais elevados.

No entanto são disponibilizadas por estes as linhas mestras de orientação sobre as formas de melhorar um ranking de um site e que nos vão elucidando de qual o caminho a seguir. Outra forma de controlar todo esse processo de optimização, que é contínuo, é ir seguindo o desempenho dos nossos sites, através de ferramentas que iremos apresentar e utilizar, como o **Google Analytics** (www.google.com/analytics), ou o **Statcounter** (www.statcounter.com), sendo estas as mais populares das inúmeras à nossa disposição, umas pagas e outras gratuitas. Se a performance se degradar, é sinal que não estamos a utilizar a estratégia mais correcta. Nos próximos artigos irei explicar como utilizar estas ferramentas e outras ferramentas e tirar o máximo partido delas, no processo de optimização dos nossos sites.

StatCounter.com



Google Analytics

Outro factor que está constantemente a “baralhar” o universo de fiéis seguidores da religião SEO, é que os tão badalados algoritmos estão também em constante mudança e aperfeiçoamento, pelo que convém estar atento e seguir as últimas modas. Um site optimizado para as regras de hoje, pode estar menos optimizado no ano seguinte. Além disso é aconselhável que verifiquemos o *ranking* do nosso site regularmente, de forma a detectarmos alterações menos positivas no nosso ranking.

Qual é o interesse de se atribuírem *rankings* às páginas web? A resposta é simples, porque as pessoas quando pesquisam algo num motor de busca, esperam encontrar o que procuram! Se não existisse um processo de selecção em que os melhores sites ficam mais bem posicionados, a experiência do utilizador num motor de pesquisa seria péssima.

O algoritmo do Google para atribuição de *ranking* a uma página Web chama-se PageRank.

De acordo com uma própria definição antiga da Google

SEO – SEARCH ENGINE OPTIMIZATION – INTRODUÇÃO PARTE II

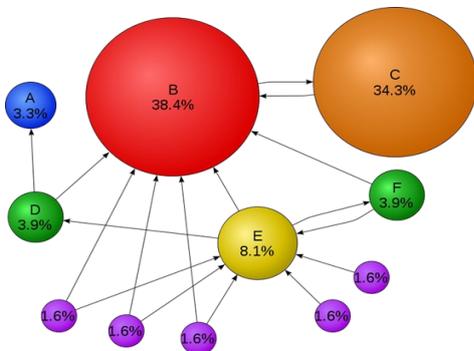
(2005), PageRank é:

“PageRank is a link analysis algorithm that assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of “measuring” its relative importance within the set. The algorithm may be applied to any collection of entities with reciprocal quotations and references. The numerical weight that it assigns to any given element E is also called the PageRank of E and denoted by PR(E).”

Que em Português significa que o PageRank é um algoritmo de análise de links que atribui um peso numérico a cada elemento de um conjunto de documentos interligados por links. Este algoritmo pode ser aplicado a uma colecção de entidades com referências reciprocas. O peso numérico que é atribuído ao elemento E, é também chamado de PageRank de E ou PR(E).

Em termos mais abstractos, o PageRank (PR) é uma aferição muito importante da popularidade que determinada página da Internet possui, numa escala de 0 a 10 em que 10 significa “o mais importante” ou de “alta popularidade”.

Podemos também pensar em PR como um conjunto de votos agregados de todas as páginas que “linkam” (hiperligam) para determinada página, como podemos observar pela imagem seguinte (fonte: Wikipedia):



Um dos sites sobejamente conhecidos por quem se interessa pelo tema, é o www.prchecker.info/, que permite visualizar qual o rank do nosso site, introduzindo o seu endereço num campo de pesquisa, que posteriormente nos devolve o nosso ranking, que pode ir de uma escala de 0 a 10, sendo que

10 é o santo Gral do SEO, onde poucos sites conseguem chegar. Como podem ver pela imagem seguinte, nem o portal www.google.pt consegue atingir essa meta, embora seja uma das páginas mais utilizadas no nosso país:

Check PAGE RANK of Web site pages Instantly

In order to check pagerank of a single web site, web page or domain name, please submit the URL of that web site, web page or domain name to the form below and click "Check PR" button.

Web Page URL: <http://google.pt>

The Page Rank: 8/10

(the page rank value is 8 from 10 possible points)

Actualmente, e depois de muitas “mexidas” por parte da Google, muitos são os factores que levam ao cálculo deste número. Não é certo nem sabido como é que a Google atribui esta pontuação, mas este algoritmo é o resultado de uma equação composta por múltiplas variáveis, cada qual com o seu peso, como por exemplo o número de *links* de qualidade que existem para a nossa página, riqueza de *keywords*, rapidez e desempenho do servidor, etc. etc. Fiquem descansados que entraremos em detalhe em cada um deles, em próximos artigos. Outra mensagem que gostaria que retivessem é que este número não é estático e que varia temporalmente. Umhas vezes a Google deixa passar um ano sem lhe mexer, sendo que em outras o altera mensalmente. Não sejamos “fundamentalistas” do PageRank! Este número deve ser encarado como um barómetro, servindo apenas para irmos monitorizando o desempenho SEO do nosso site.

Lembrem-se sempre que a variável com maior peso deve ser sempre o nosso “conteúdo” e a sua qualidade. Os sites são construídos e actualizados a pensar nas pessoas e não para motores de busca! Ao despendermos demasiados recursos com o SEO podemos correr o risco de descuidar no conteúdo e isso sim faz com que as nossas páginas percam visitas! Os principais destinatários dos nossos sites são as pessoas, os *bots* vêm em segundo lugar!

AUTOR



Escrito por Miguel Lobato

Licenciado em Tecnologias da Informação e Comunicação (TIC) e é Consultor de Search Engine Optimization – SEO e Administração de Sistemas. É também Webdesigner, Webdeveloper e um apaixonado por tudo o que é relacionado com as novas tecnologias.

<https://www.facebook.com/MiguelLobatoSEO> - [@MiguelLobatoSEO](https://twitter.com/MiguelLobatoSEO)

Elege o melhor artigo desta edição

Revista PROGRAMAR

http://tiny.cc/ProgramarED36_V

Implementação da Cifra ARSD em Ambiente Emulado Windows Azure

Base Teórica

Antes de iniciarmos a parte prática que sustentou este artigo, tornou-se necessário para nós compreender melhor a teoria por trás de um novo modelo de computação.

Windows Azure

O Windows Azure (WA), foi apresentado pela primeira vez ao público em Outubro de 2008, durante a Conferência de Desenvolvedores Profissionais da Microsoft, e é considerado um sistema operativo para o modelo de computação em nuvem (Cloud). Por este motivo, o WA não é vendido para computadores pessoais ou servidores empresariais, a sua licença de uso é paga mensalmente, consoante os recursos necessários pelo utilizador.

Visão Geral do Windows Azure

A execução de aplicações e o armazenamento de dados em máquinas de um centro de dados acessível via Internet vão oferecer uma série de vantagens. Onde quer que sejam executados, esses aplicativos são criados numa plataforma computacional. Para aplicações locais, como aqueles executados dentro da base de dados de uma organização, essa plataforma geralmente inclui um sistema operativo e uma forma de armazenar esses mesmos dados. Os aplicativos em execução na nuvem também necessitam de uma base semelhante. O objetivo do Windows Azure é oferecer essa base a todos os seus utilizadores. A sua plataforma funciona como uma base para a execução de aplicações Windows e armazenamento de dados na nuvem, sendo um novo modelo de computação como pode ser observado na ilustração 1.



Ilustração 1 - Esquema Representativo da Plataforma WA

Como já foi referido anteriormente, ao contrário do que a Microsoft nos tem habituado, o WA não é um novo sistema operativo, mas sim um serviço. Ou seja, em vez de fornecer software que os clientes da Microsoft possam instalar e executar por conta própria nos seus computadores, estes fazem uso dele para executar aplicativos e armazenar dados em máquinas acessíveis pela Internet pertencentes à Microsoft. Alguns exemplos dos tipos de aplicativos que podem ser criados com o Windows Azure são:

- Um fornecedor de software independente, ISV, pode criar uma aplicação para empresas, numa abordagem denominada por SaaS, ou Software As A Service. O Windows Azure foi desenhado para dar apoio a este tipo de aplicações, sendo estas as próprias aplicações SaaS da Microsoft. Assim, os ISV podem usá-lo como base para uma grande variedade de softwares e serviços numa nuvem empresarial. Um programador pode criar uma aplicação SaaS voltada para os consumidores particulares, em vez de consumidores empresariais.

O Windows Azure foi feito para suportar software bastante escalonável e, com isso, uma empresa que pretenda alargar o seu mercado de possíveis consumidores pode escolher o WA como plataforma para uma nova aplicação. Por outro lado, as empresas podem utilizar esta plataforma para criar e executar aplicações usadas pelos próprios funcionários. Ainda que esta situação, dificilmente exija uma escalabilidade maior do que um software voltado para os consumidores, a confiabilidade e a capacidade de gerenciamento oferecidas pelo Windows Azure continuam a apresentá-lo, como uma oferta a ter em conta. Para suportar dados e aplicações na nuvem, o Windows Azure possui cinco componentes, como é visível na ilustração 2.

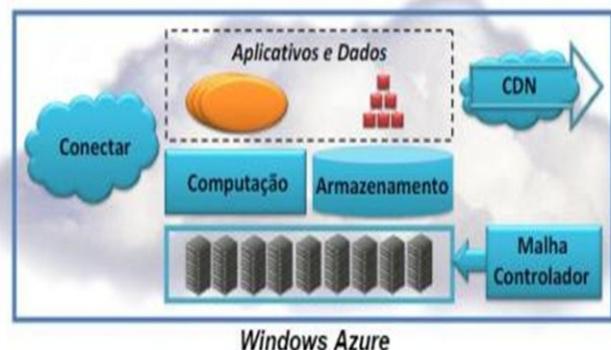


Ilustração 2 - Esquema Representativo dos Componentes da Plataforma Windows Azure

A PROGRAMAR

IMPLEMENTAÇÃO DA CIFRA ARSD EM AMBIENTE EMULADO WINDOWS AZURE

Ou seja, os cinco componentes principais do WA são:

- **Computação:** Executam aplicações na nuvem.
- **Armazenamento:** Armazena dados binários e estruturados na nuvem.
- **Controlador da Malha:** Implementa, gere e monitoriza aplicações. O controlador da malha, também controla as atualizações no software do sistema.
- **Content Delivery Network (CND):** Acelera o acesso global a dados binários no armazenamento do Windows Azure, mantendo cópias em cache daqueles dados no mundo inteiro.
- **Conexão:** Permite a criação de conexões, no nível de IP entre computadores locais e aplicações do Windows Azure.

Nos próximos itens, de forma a que o leitor possa compreender melhor o funcionamento deste modelo, vamos tentar, de uma forma simples, explicar um pouco melhor o que cada uma destas componentes nos permite efetuar.

Computação

O serviço de computação do Windows Azure, pode executar muitos tipos diferentes de aplicações. Independente do objetivo da aplicação, esta deve ser implementada com uma ou mais instâncias. Assim, o Windows Azure executa, geralmente, múltiplas instâncias de cada função, usando um balanceamento embutido para distribuir as solicitações entre elas. A ilustração 3 ajuda a entender essa mesma realidade.

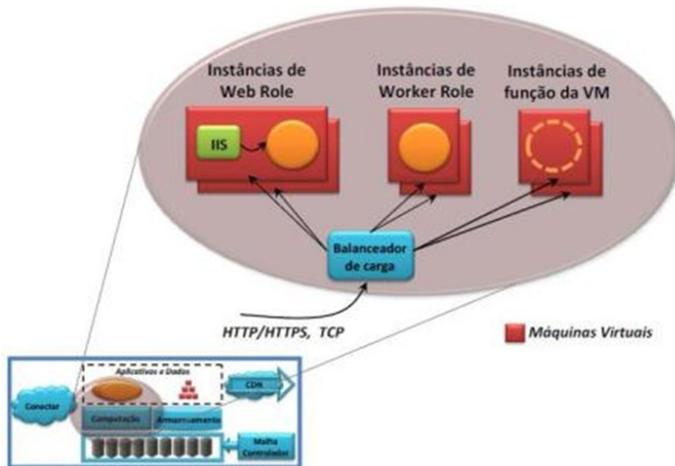


Ilustração 3 - Esquema Representativo dos Componentes da Plataforma Windows Azure

Na versão atual do Windows Azure, os programadores podem escolher entre três tipos de instâncias:

1. **Web Roles:** A sua principal finalidade é facilitar a criação de aplicações baseadas na Web. Cada ins-

tância da Web Role traz o Internet Information Services (IIS) pré-configurado, o que simplifica a criação de aplicativos usando ASP.NET ou Windows Communication Foundation (WCF). Contudo, caso os programadores o desejem, também podem criar aplicativos noutras linguagens de programação, ou seja, o uso do .NET Framework mas não é obrigatório. Isso significa que estes também podem instalar e executar outras tecnologias, não pertencentes à Microsoft, incluindo PHP ou o Java, como é o nosso caso, neste artigo.

2. **Worker Roles:** Desenhadas para executar uma série de códigos baseados em Windows, a sua principal diferença em comparação com as anteriores é que esta não traz o IIS configurado, então o código que esta executa não é hospedado pelo IIS. Uma Worker Role pode, por exemplo, executar uma simulação ou manipular um processamento de vídeo. É comum um aplicativo interagir com os utilizadores por meio de uma Web Role e então entregar as tarefas para que sejam processadas por uma Worker Role. Mais uma vez, o desenvolvedor é livre para usar a linguagem que mais desejar.
3. **Funções VM:** Cada uma delas executa uma imagem do Windows Server fornecida pelo utilizador. Entre outras coisas, uma função deste tipo pode ser útil na migração de um aplicativo local do Windows Server para o Windows Azure.

Para enviar um aplicativo para o Windows Azure, o programador pode usar o portal da plataforma. Juntamente com o aplicativo, são enviadas informações de configuração que informam à plataforma quantas instâncias de cada função devem ser executadas. O controlador da malha do Windows Azure cria nessa altura uma máquina virtual (VM) para cada instância, executando o código para a função apropriada naquela VM. Como exemplifica a ilustração 3, as solicitações dos utilizadores do aplicativo podem ser feitas usando protocolos como HTTP, HTTPS e TCP, dependendo da necessidade de cada caso em concreto. Independentemente do protocolo usado, esses pedidos são balanceados entre todas as instâncias de uma função. O balanceador de carga não permite a criação de uma afinidade com uma instância específica, ou seja, mesmo que o utilizador seja o mesmo, os seus pedidos não serão enviados para a mesma instância do serviço. Ou seja, qualquer estado específico de um cliente deve ser gravado no armazenamento do Windows Azure.

Para permitir a monitorização e a depuração de aplicativos do Windows Azure, cada instância pode chamar uma API de registro em que escreve informações que o programador ache importantes. O programador pode ainda configurar o sistema para obter contadores de desempenho de um determinado serviço. Por exemplo, se uma Worker Role parar de

responder três vezes numa hora, um código personalizado, previamente definido pelo programador, pode ser enviado por email para o administrador desse serviço.

Armazenamento

A maneira mais simples de armazenar dados no armazenamento do Windows Azure é com a utilização de blobs. Um blob contém dados binários e, como mostra a ilustração 4, há uma hierarquia simples: cada contentor pode ter um ou mais blobs, podendo estes ser grandes (até 1 terabyte) e ter metadados associados. Fornecem armazenamento subjacente para drives do Windows Azure, um mecanismo que permite que uma instância de uma função do Windows Azure interaja com o armazenamento persistente, como se fosse um sistema local de arquivos NTFS. Quando os blobs não são suficientes e é necessário tratar dos dados de forma mais apurada, o armazenamento do Windows Azure oferece tabelas. Contudo, não se tratam de tabelas relacionais. Os dados que cada função traz são armazenados num grupo de entidades que contém propriedades. E, em vez de usar SQL, um aplicativo pode consultar os dados de uma tabela usando as convenções definidas nos serviços de dados OData. Esta abordagem permite um armazenamento de escalabilidade horizontal. Seja como for a maneira de armazenar os dados, quer seja em blobs, tabelas ou filas, todas as informações mantidas no armazenamento do Windows Azure são replicadas três vezes. Esta replicação permite tolerância a falhas, já que a perda de uma cópia não é fatal. No entanto, o sistema oferece uma forte consistência, e um aplicativo que leia imediatamente dados que acabou de gravar tem a garantia de obter de volta tudo aquilo que acabou de gravar. O WA mantém uma cópia de backup de todos os dados em outro data center na mesma parte do mundo. Assim, se o data center que armazena a cópia principal não estiver disponível ou for destruído, o backup continua sempre acessível. O armazenamento do Windows Azure pode ser acessado por um aplicativo dele próprio, por um aplicativo local ou por um aplicativo executado noutra plataforma na nuvem. Em todos esses casos, os três estilos de armazenamento do Windows Azure usam as convenções do REST (Transferência de Estado Representacional) para identificar e expor os dados.

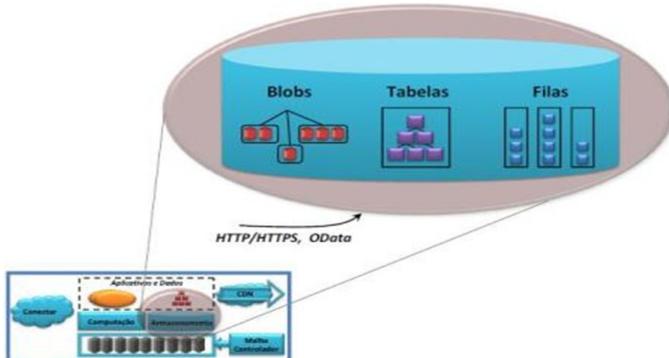


Ilustração 4 - Esquema Representativo dos Componentes da Plataforma Windows Azure

Controlador da Malha

Todos os aplicativos do Windows Azure e todos os dados no armazenamento do Windows Azure são armazenados num data center da Microsoft. No data center, o conjunto de máquinas dedicadas ao Windows Azure e o software nelas executado são gerenciados pelo controlador da malha. A ilustração 5, realça essa realidade. O controlador da malha é um aplicativo distribuído que é replicado entre um grupo de máquinas. Este controlador detém todos os recursos de seu ambiente: computadores, comutadores, balanceadores de carga, etc. Pode ainda comunicar com um agente de

malha em cada computador, estando assim ciente de todos os aplicativos do Windows Azure nessa mesma malha. Esse conhecimento total do sistema permite que o controlador de malha faça várias coisas, como monitorizar todos os aplicativos em execução, oferecendo, assim uma visão atualizada de tudo o que está acontecendo. Ele também decide onde os novos aplicativos devem ser executados, escolhendo servidores físicos para otimizar a utilização de hardware. Para fazer isso, o controlador de malha depende das informações de configuração que são carregadas com cada aplicativo do Windows Azure. Esse arquivo fornece uma descrição, baseada em XML, do que o aplicativo precisa de saber, ou seja, quantas instâncias de Web Roles e quantas instâncias de Worker Roles.

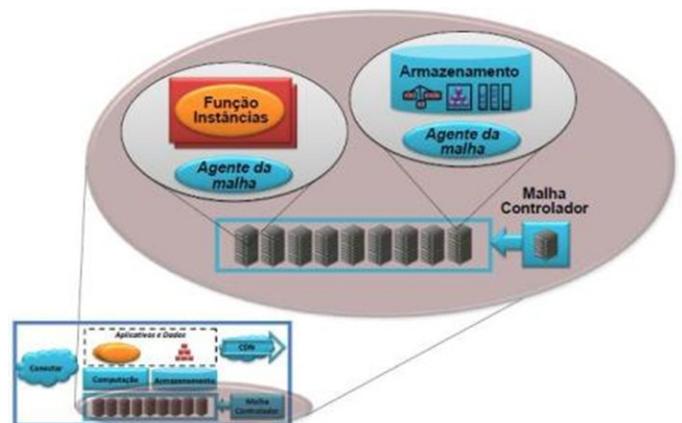


Ilustração 5 - Esquema Representativo dos Componentes da Plataforma Windows Azure

Quando o controlador da malha implementa uma nova aplicação, usa este arquivo de configuração para determinar quantas VM devem ser criadas. Depois de criar as referidas VM, o controlador de malha monitoriza cada um deles. Se um aplicativo exigir cinco instâncias de uma Web Role e, por exemplo, uma delas for encerrada, o controlador de malha iniciará automaticamente uma nova.

Tendo em conta que o Controlador de Malha é o responsável pelo bom funcionamento do serviço, seria inconveniente colocar todas as instâncias a serem servidas pelo mesmo computador de rede. A ilustração 6, permite-nos visualizar melhor a disposição das instâncias do serviço.

A PROGRAMAR

IMPLEMENTAÇÃO DA CIFRA ARSD EM AMBIENTE EMULADO WINDOWS AZURE

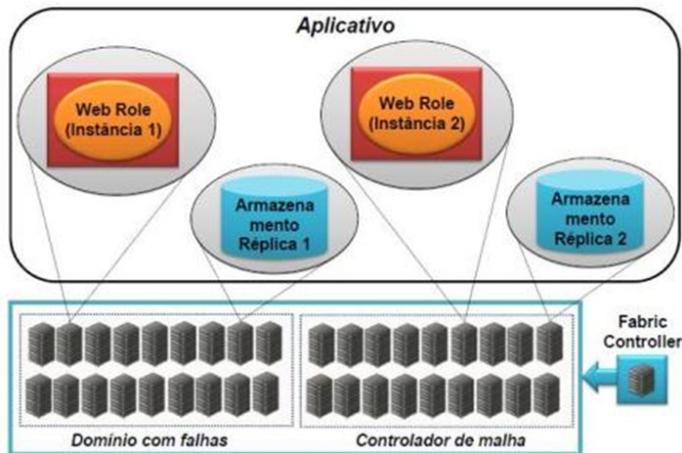


Ilustração 6 - Esquema Representativo dos Componentes da Plataforma Windows Azure

Content Delivery Network

Um uso comum dos blobs é armazenar informações que são acessadas de muitos lugares diferentes. De uma forma simples, se considerarmos um aplicativo que fornece vídeos, por exemplo, a clientes Flash, HTML 5 ou Silverlight em todo o mundo. Para melhorar o desempenho do sistema, o Windows Azure oferece uma rede de fornecimento de conteúdo (CDN). A CDN armazena cópias de um blob em locais mais próximos dos clientes que as utilizam, de forma a que a cópia esteja disponível no menor tempo possível. A ilustração 7, realça essa ideia.

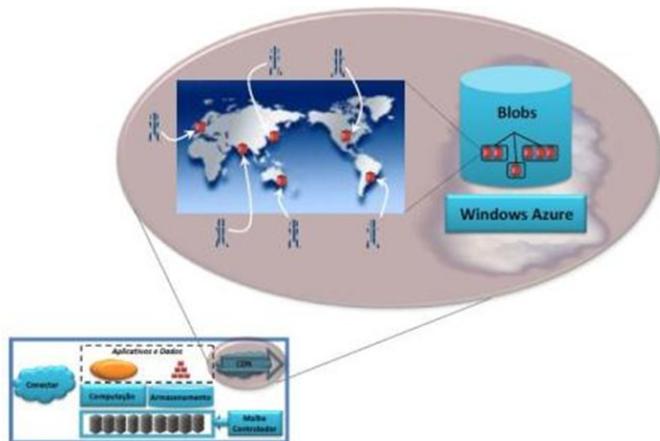


Ilustração 7 - Esquema Representativo dos Componentes da Plataforma Windows Azure

Conexão

Executar aplicações na nuvem Microsoft pode ser útil. Contudo os serviços e os dados que usamos, por exemplo, na nossa empresa, possivelmente vão ter uma duração mais longa. Neste caso, é importante que a conexão de ambientes locais com o Windows Azure seja eficiente. A componente de Conexão do Windows Azure foi criada para isso. Esta componente pode facilitar essa combinação ao oferecer co-

nectividade no nível de IP entre uma aplicação do Windows Azure e as máquinas em execução fora da nuvem Microsoft, como pode ser visível na ilustração 8. Como podemos verificar por esta figura, o uso da componente de Conexão do Windows Azure requer a instalação de um agente de Endpoint em todos os computadores locais que estiverem conectados a uma aplicação da plataforma. Como a tecnologia usa o IP v6, o agente de Endpoint está disponível atualmente apenas para o Windows Server 2008, Windows Server 2008 R2, Windows Vista e Windows 7. O Windows Azure também precisa ser configurado, para trabalhar com a componente de Conexão dele próprio. Assim que isso for feito, o agente pode usar o IPsec para interagir com uma função específica da aplicação. É importante o programador ajustar a nuvem ao ambiente local, de modo a permitir conectividade direta no nível de IP.

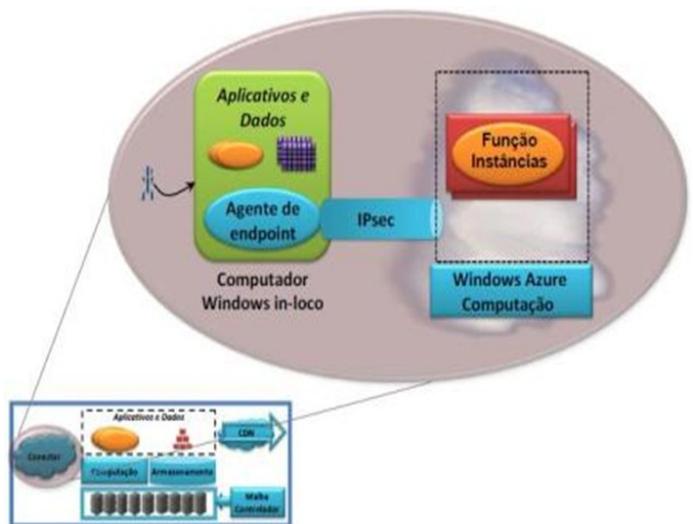


Ilustração 8 - Esquema Representativo dos Componentes da Plataforma Windows Azure

Desenvolvimento e Administração de Aplicações

O Windows Azure fornece uma porta de entrada para o ambiente de nuvem, sendo que é possível hospedar as soluções para a plataforma Azure, após estas serem desenvolvidas e implementadas. O primeiro passo necessário, além de uma conta no ambiente, é um alocar o serviço que irá alocar os recursos para o nosso aplicativo. De forma a proporcionar a maior disponibilidade possível aos seus clientes, a Microsoft estabeleceu centros de dados regionais que permitem aos clientes selecionar serviços geograficamente dispersos.

Desta forma, quando criamos os nossos servidores no Windows Azure, devemos sempre especificar qual a localização geográfica dos mesmos. Este recurso é denominado por Windows Azure Geo-Location. Apesar de inicialmente ser tentador para escolher a localização geográfica da nossa empresa para um melhor desempenho, se precisamos por exemplo, que a disponibilidade da aplicação seja mais impor-

A PROGRAMAR

IMPLEMENTAÇÃO DA CIFRA ARSD EM AMBIENTE EMULADO WINDOWS AZURE

tante do que o tempo de resposta, podemos precisar de escolher outro local. Ao selecionar uma localização geográfica, devemos considerar os seguintes pontos:

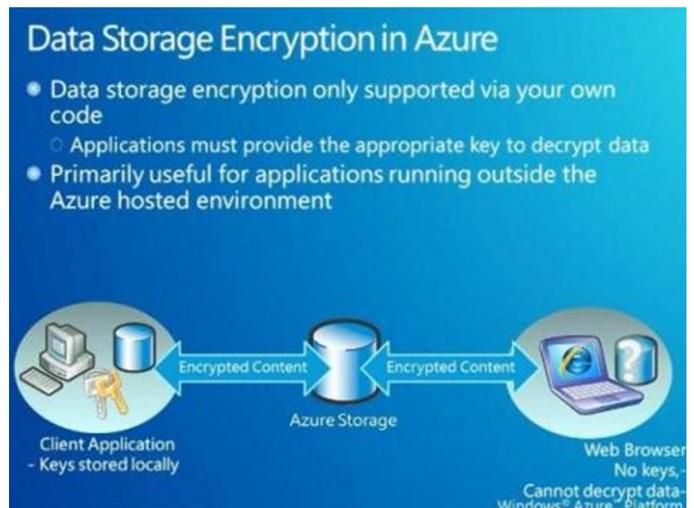
- **Desempenho:** Quando os dados estão mais perto dos clientes da empresa, a latência da rede pode ser visivelmente inferior, melhorando a experiência do cliente.
- **Recuperação de desastres:** Se garantir a disponibilidade da nossa aplicação é importante, podemos querer dispersar os seus serviços e dados em várias regiões.
- **Fatores legais:** Devemos considerar o tipo de informação armazenada na nuvem, e garantir que não são necessários regulamentos específicos e mandatos que podem impedir-nos de selecionar localizações geográficas mais remotas.

É sempre importante alertar de novo para o facto de que cada vez mais, cada caso é um caso, isto é, se tivermos uma empresa de software é importante recordar sempre que cada cliente procura uma solução única à sua medida, um bom profissional de informática é aquele que após estudar o caso do seu cliente, consegue desenhar uma solução à medida do mesmo e, sempre que possível facilitar e melhorar ainda o dia-a-dia do cliente. Portanto nunca é demais referir que neste modelo de computação, cada caso é um caso.

Segurança do Windows Azure

Apesar de termos tido um curto espaço de tempo, para estudar este novo modelo de computação e tendo em conta que hoje em dia, cada vez se dá mais valor aos dados privados e pessoais de uma entidade ou de um indivíduo, é importante referir algo acerca da segurança desta nova tecnologia. O facto de cada pedaço de informação ser replicado três vezes, pode levar a que os programadores mais desconfiados não vejam com bons olhos a evolução deste sistema. Se por um lado é verdade que a nuvem, ajuda a que haja uma poupança a nível de custos de hardware, porque não estamos preocupados em quantas máquina precisamos para manter disponível o nosso serviço 24 horas por dia, a verdade é que todos os dados que estão em suporte digital podem ser um alvo fácil. É importante referir que, como seria esperado a Microsoft não descurou a segurança, pelo contrário, a segurança do WA foi reforçada. Os dados são armazenados em recursos de hardware separados, organizados em contas de armazenamento, e o acesso a esses mesmos dados é específico de cada conta, sendo que cada uma dessas contas tem uma password de acesso. O programador pode ainda definir a sua própria política de segurança implementando SAS (Shared Access Signatures) aos blobs de dados, os dados podem assim ser encriptados, consoante seja o desejo do programador, como mostra a ilustração 9.

Ilustração 9 - Esquema Representativo da Segurança do Windows Azure



Algumas das boas práticas de segurança que podemos implementar na Nuvem, enquanto programadores são:

- Validação do input;
- Autenticação;
- Validação;
- Definir o acesso a dados sensíveis;
- Criptografia;
- Administração de configuração;
- Administração de Sessão.

As ilustrações 10 e 11, exemplificam diferentes tipos de acesso aos dados da aplicação.

Mais uma vez, chamamos a atenção de que qualquer que seja a aplicação e o serviço presente neste modelo de computação, o principal responsável pelo bom uso dos dados da mesma, é o próprio programador.

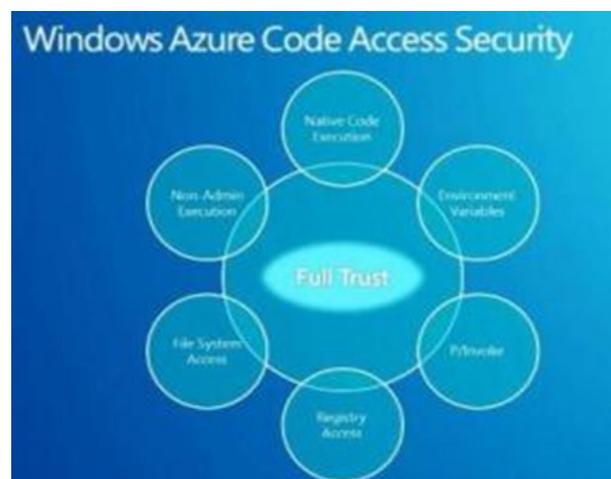


Ilustração 10 - Esquema Representativo da Segurança do Windows Azure

A PROGRAMAR

IMPLEMENTAÇÃO DA CIFRA ARSD EM AMBIENTE EMULADO WINDOWS AZURE



Ilustração 11 - Esquema Representativo da Segurança do Windows Azure

Cifra ARSD

Para este artigo, implementamos uma cifra simples de forma a tentar perceber melhor qual a “máquina operacional” em torno do WA, por esse motivo decidimos construir o nosso próprio algoritmo de encriptação e desencriptação, ao qual demos o nome de ARSD (André Rita Sistemas Distribuídos).

O algoritmo ARSD é um algoritmo simples que:

- **Na encriptação do ficheiro:** O sistema vai ler o que está no ficheiro para o buffer. Depois de tudo colocado no buffer, vai ler carácter a carácter e converter cada carácter para o código ASCII do carácter seguinte na tabela desse mesmo código e guardar esse mesmo valor. Posteriormente, vai fazer a conversão desse mesmo valor ASCII para o seu equivalente em carácter e vai escrevê-lo no ficheiro encriptado.
- **Na desencriptação do ficheiro:** O sistema vai fazer exatamente o contrário do que é feito na encriptação, ou seja, vai ler o que está no ficheiro para o buffer. Depois de tudo colocado no buffer, vai ler carácter a carácter e converter cada um para o código ASCII do carácter anterior na tabela e guardar esse mesmo valor. Posteriormente, vai fazer a conversão desse mesmo valor ASCII para o seu equivalente em carácter e vai escrevê-lo no ficheiro desencriptado.

Como o leitor pode observar, o algoritmo realizado não tem um nível de dificuldade muito elevado, mas tendo em conta que o principal objetivo deste artigo é o entendimento do funcionamento de um novo modelo de computação, esta

pequena cifra permite-nos visualizar todas as mais-valias do WA.

Desenvolvimento

Ferramentas Utilizadas

Ambiente de Desenvolvimento

Este projeto foi desenvolvido em ambiente Windows, sendo usado a versão Windows Vista Home Edition. Foi usado ainda o software Eclipse com o plugin do Windows Azure como IDE de desenvolvimento aplicação.

Linguagem de Implementação – JSP

A linguagem utilizada na escrita deste projeto foi o JSP. O java server page é uma linguagem/tecnologia utilizada para o desenvolvimento de aplicações Web, e é semelhante ao ASP.NET da Microsoft. Como é baseada na Linguagem Java, tem como grande vantagem a sua portabilidade, o que permite a sua execução e utilização em diversos sistemas operativos e em diferentes tipos de hardware. Uma página criada em JSP, após ser instalada num servidor compatível com a tecnologia Java, pode então ser transformada em Servlet (componente semelhante a um servidor que gera dados HTML e XML). Pode ser encarado como um módulo que estende a funcionalidade de um servidor Web, através de módulos implementados em Java.

Servidor Auxiliar Usado – GlassFish

O servidor auxiliar que usamos no nosso trabalho foi o GlassFish versão 3. Criado em 2005, pela Sun Microsystems para a plataforma Java EE, este é um servidor open-source. Passados uns anos, a comunidade GlassFish lançou a versão 2 de GlassFish (também conhecida como Sun Java System Application Server 9.1), com capacidade completa para o clustering corporativo e web services compatíveis com tecnologia Microsoft. Contudo em Fevereiro de 2011, a Oracle lançou ao mundo, o GlassFish v3.1. Esta versão introduziu assim suporte ao clustering e ao balanceamento de carga, além de suportar a nova especificação Java EE 6 Web Profile.

As opções tomadas

Com já referimos anteriormente, umas das opções tomadas ao longo do decorrer do nosso projeto foi a implementação do Algoritmo ARSD, de modo a que pudéssemos atingir um dos objetivos principais deste trabalho que era a implementação de um algoritmo de encriptação e de desencriptação. Foram

A PROGRAMAR

IMPLEMENTAÇÃO DA CIFRA ARSD EM AMBIENTE EMULADO WINDOWS AZURE

desenvolvidos seis ficheiros jsp ao longo da implementação deste projeto, de modo a facilitar a interação do utilizador com a nossa aplicação para a nuvem, no final de cada encriptação ou desencriptação o utilizador tem ainda a possibilidade de imprimir o ficheiro gerado. Como estamos a trabalhar em localhost, os ficheiros gerados encontram-se na pasta C://Azure, de modo a que caso o utilizador deseje, os ficheiros estão acessíveis. Ou seja, se o nosso sistema estivesse implementado na nuvem, o utilizador A poderia encriptar um ficheiro e envia-lo ao utilizador B que poderia desencriptar esse ficheiro através da nossa aplicação, e desta forma, poderia então ter acesso ao conteúdo original do ficheiro desencriptado.

Para que este seja um artigo que ajuda o leitor a colocar em prática todos os passos da implementação do ambiente WA, elaboramos também um pequeno tutorial de como instalar todos os componentes necessários. Assim no anexo deste artigo é possível verificar como instalar o software Eclipse, o plugin do WA no Eclipse, foi ainda explicado como iniciar um projeto em Windows Azure assim como a compilação do mesmo.

É importante referir que o tutorial apresentado, é baseado no tutorial oficial que a própria Microsoft disponibiliza em <http://bit.ly/sKLMeM>

Contudo, foi tomada a opção de alterar o servidor auxiliar, sendo utilizado neste projeto o GlassFish, no anexo "Iniciar um projeto em Windows Azure", foi ainda explicada a forma como o ficheiro auxiliar que inicia o processo foi alterado.

Ficheiros

index.jsp

Como pode ser visível no código abaixo, o ficheiro apresenta a hora do sistema, e pede ao utilizador que introduza o ficheiro que deseja encriptar ou desencriptar.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" http://www.w3.org/TR/html4/loose.dtd">
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>SD - Cifra por Blocos</title>
</head>
<body>
<b>Bem-Vindo à Cifra Por Blocos - Cifra ARSD</b>
<br>
<b>Versao 3.0.3</b>
<br><br>
<%@ page import="java.net.*" %>
<%
    out.println( "Data: " );
```

```
java.util.Date date = new java.util.Date();
out.println(" " + date);
%>
<br>
<FORM ENCTYPE="multipart/form-data" ACTION="upload.jsp" METHOD=POST><br>
<b>Escolha o ficheiro a analisar:</b><INPUT NAME="F1" TYPE="file"> <br>
<INPUT TYPE="submit" VALUE="Enviar Ficheiro" >
</FORM> </body> </html>
```



Ilustração 12 - Página Inicial da Aplicação

A ilustração 12, mostra a página de apresentação gerada por este ficheiro de código.

Upload.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Pagina de Upload</title>
</head>
<body>
<%@ page import="java.io.*" %>
<% String contentType = request.getContentType();
if ((contentType != null) && (contentType.indexOf("multipart/form-data") >= 0)) {
    DataInputStream in = new DataInputStream(request.getInputStream());
    int formDataLength = request.getContentTypeLength();
    byte dataBytes[] = new byte[formDataLength];
    int byteRead = 0;
    int totalBytesRead = 0;
    while (totalBytesRead < formDataLength)
    {
        byteRead = in.read(dataBytes, totalBytesRead, formDataLength);
        totalBytesRead += byteRead;
    }
    String file = new String(dataBytes);
```

A PROGRAMAR

IMPLEMENTAÇÃO DA CIFRA ARSD EM AMBIENTE EMULADO WINDOWS AZURE

```
String saveFile = file.substring(file.indexOf(
    ("filename=\"") + 10);
saveFile = saveFile.substring(0, saveFile.indexOf(
    ("\n"));
saveFile = saveFile.substring(saveFile.lastIndexOf(
    ("\\")) + 1,saveFile.indexOf("\\"));
int lastIndex = contentType.lastIndexOf("=");
String boundary = contentType.substring(lastIndex
    + 1,contentType.length());

int pos;
pos = file.indexOf("filename=\"");
pos = file.indexOf("\n", pos) + 1;
pos = file.indexOf("\n", pos) + 1;
pos = file.indexOf("\n", pos) + 1;
int boundaryLocation = file.indexOf(boundary, pos)
    - 4;

int startPos = ((file.substring(0, pos)).getBytes(
    ())).length;

int endPos = ((file.substring(0,
    boundaryLocation)).getBytes()).length;
File f = new File("C:\\Azure\\"+saveFile);
FileOutputStream fileOut = new FileOutputStream(f);
fileOut.write(dataBytes, startPos,
    (endPos - startPos));
fileOut.flush();
fileOut.close();
%>
<br>
<FORM ENCTYPE="multipart/form-data"
    ACTION="encrypta.jsp" METHOD=POST>
    <INPUT TYPE="submit" VALUE="Encriptar Ficheiro" >
</FORM>
<FORM ENCTYPE="multipart/form-data"
    ACTION="decrypta.jsp" METHOD=POST>
    <br> <INPUT TYPE="submit" VALUE="Desencriptar
Ficheiro" >
</FORM>
</body></html>
```

Este ficheiro coloca o ficheiro escolhido pelo utilizador no “suposto servidor WA”, recordamos ao leitor que o nosso sistema emula o WA, e que o ficheiro inserido é colocado na localização C://Azure, caso esse upload seja bem-sucedido, é apresentada uma mensagem ao utilizador dizendo que o ficheiro foi corretamente enviado para o servidor.

São ainda apresentados dois botões que permitem ao utilizador escolher se quer encriptar ou desencriptar o ficheiro, como mostra a ilustração 13.



Ilustração 13 - Página de escolha da Aplicação

Encripta.jsp

```
<%@ page language="java" contentType="text/html;
    charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
    Transitional//EN" "http://www.w3.org/TR/html4/
    loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
    charset=ISO-8859-1">

<title>Encriptar</title>
</head><body>
<br>
<b>Versao 3.0.3</b>
<br>
<%@ page import="java.io.*" %>
<%@ page import="java.math.*" %>
<%@ page import="java.net.*" %>
<%
char[] buffer = new char[8] ;
char[] enc = new char[8];
int len, cod;
FileReader r = new FileReader
("C:\\Azure\\encryptar.txt");
FileWriter w = new FileWriter
("C:\\Azure\\encryptado.txt") ;
while( (len = r.read(buffer)) != -1 ){
    for(int j=0; j<len; j++){
        cod = (int) buffer[j] + 1;
        enc[j] = (char) cod;
    }
    w.write(enc, 0, len) ;
}
r.close() ;
w.close() ;
%>

<FORM ENCTYPE="multipart/form-data" ACTI-
ON="mostraEncryptado.jsp" METHOD=POST> <br>
    <INPUT TYPE="submit" VALUE="Mostrar Encriptação"
    >
</FORM>
</body></html>
```

Como pode ser observado, é lido da pasta emulada do WA, o ficheiro encryptar.txt, ao qual é aplicado a cifra ARSD, sendo esse mesmo resultado arquivado no artigo encryptado.txt.

Decrypta.jsp

```
<%@ page language="java" contentType="text/html;
    charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
    Transitional//EN" "http://www.w3.org/TR/html4/
    loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
```

IMPLEMENTAÇÃO DA CIFRA ARSD EM AMBIENTE EMULADO WINDOWS AZURE

```
charset=ISO-8859-1">
<title>Desencriptar</title>
</head>
<body>
<br>
<b>Versao 3.0.3</b>
<br>

<%@ page import="java.io.*" %>
<%@ page import="java.math.*" %>
<%@ page import="java.net.*" %>

<%

char[] buffer = new char[8] ;
char[] desenc = new char[8];
int cod2, len2;

FileReader dr = new FileReader
("C:\\Azure\\encryptado.txt");
FileWriter dw = new FileWriter
("C:\\Azure\\desencriptado.txt");

    while( (len2 = dr.read(buffer)) != -1 )
    {
        for(int j=0; j<len2; j++)
        {
            cod2 = 0;
            cod2 = (int) buffer[j] - 1;
            desenc[j] = (char) cod2;
        }
        dw.write(desenc, 0, len2) ;
    }
    dr.close() ;
    dw.close() ;
%>

<FORM ENCTYPE="multipart/form-data"
ACTION="mostraDesencriptado.jsp" METHOD=POST>
<br>
<INPUT TYPE="submit" VALUE=
"Mostrar Desencriptação" >
</FORM>

</body>

</html>
```

À semelhança do ficheiro apresentado anteriormente, este ficheiro apenas lê o encryptado.txt e aplica a cifra ARSD de desencriptação e guarda o resultado no ficheiro desencriptado.txt.

Em ambos os ficheiros anteriores é dada a opção de mostrar o ficheiro gerado, (recordamos que o ficheiro gerado é o ficheiro encryptado.txt ou o ficheiro desencriptado.txt, dependendo do algoritmo escolhido pelo utilizador da aplicação).

Os restantes ficheiros da aplicação o Mostraencryptado.txt e o mostradesencriptado.txt são análogos, sendo que cada um desses ficheiros dá ao utilizador a possibilidade de ver a cifra gerada pela aplicação e possibilita ainda a impressão da mesma ou então que o utilizador arquive esse numa localização à sua escolha.

mostraEncryptado.jsp

```
<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/
loose.dtd">

<html>

<head>
<meta http-equiv="Content-Type" content="text/html;
charset=ISO-8859-1">
<title>Mostrar ficheiro Encryptado</title>

</head>

<body>
vou mostrar o ficheiro Encryptado ....
<br>
<b>Versao 3.0.3</b>
<br>
<input type="button" name="print" value="Imprimir"
onClick="javascript:window.print();">
<FORM ENCTYPE="multipart/form-data"
ACTION="index.jsp" METHOD=POST>

<br>
<INPUT TYPE="submit" VALUE="Voltar" >
</FORM>
<br>
<%
String filename = "encryptado.txt";
String filepath = "C:\\Azure\\";
java.io.FileInputStream fileInputStream = new ja-
va.io.FileInputStream(filepath + filename);
int i;
while ((i=fileInputStream.read()) != -1)
{
    out.write(i);
}
fileInputStream.close();
out.close();
%>

</body>

</html>
```

Além da possibilidade de impressão ou de guardar o ficheiro, foi ainda incluída a opção de “Voltar” para que a aplicação volte ao ponto inicial de modo a que caso o desejo, o utilizador possa inserir um novo ficheiro e ponha em prática uma

A PROGRAMAR

IMPLEMENTAÇÃO DA CIFRA ARSD EM AMBIENTE EMULADO WINDOWS AZURE

nova execução do mesmo, a ilustração 14 é demonstrativa dessa fase da aplicação.



Ilustração 14 - Página de Visualização do Ficheiro Gerado

Tendo o emulador WA instalado no computador em que desenvolvemos este projeto, todas as funcionalidades que desenvolvemos podem ser utilizadas. Assim o utilizador apenas precisa de iniciar um browser e digitar o seguinte endereço:

localhost:8080/HelloWorld

Será mostrada a página inicial do nosso programa, seguidamente, o utilizador apenas tem que escolher o ficheiro que deseja encriptar e carregar no botão Enviar Ficheiro, para que o ficheiro seja carregado para o nosso servidor, de notar que como estamos a trabalhar em localhost, que na raiz do computador foi criada uma pasta chamada Azure que serve como “armazenamento” de dados.

Reflexão Crítica/Conclusão

Um dos maiores desafios no desenvolver deste projeto foi o facto de que o WA ser uma ferramenta que demora algum tempo a executar. Apesar do computador que usamos para realizar este projeto ter 4GB de memória RAM, e de termos dedicado quase todos os nossos recursos disponíveis para esta aplicação, quando corremos o emulador do WA, o tempo médio de espera é de cerca de 25 minutos, o que muito contribuiu para o lento andamento do nosso desenvolvimento.

Por este motivo, um dos nossos maiores desafios desta implementação, foi o facto de termos que pensar sempre no próximo passo, e caso a versão que estávamos a tentar verificar tivesse algum erro, como poderíamos resolver esse erro antes da próxima compilação. Este foi de facto um dos maio-

res desafios da nossa implementação.

Um outro desafio, foi a compreensão do nosso desenvolvimento, e o facto de que ao desenvolver aplicações para o WA, podemos de facto, usar mais linguagens do que a linguagem .NET. Um outro desafio foi que apesar de todos os tutoriais que pesquisamos e de toda a informação que recolhemos, a verdade é que na prática, a computação em nuvem ainda é um “Adamastor” para os informáticos em geral, uma vez que ainda há pouca informação disponível. Na fase de instalação do IDE, as dificuldades que tivemos relacionaram-se com o facto de que ao instalar o SDK do Windows Azure, automaticamente foi instalado o IIS, por este motivo, o nosso localhost assumia automaticamente este serviço e não o nosso servidor base que tínhamos previamente escolhido. Ultrapassado este contratempo, no passo seguinte, ao seguir o tutorial oficial da Microsoft, chegámos à conclusão de que apesar do tutorial recomendar o TomCat como servidor auxiliar, que no nosso caso, deveríamos utilizar o GlassFish como auxiliar.

Optámos por este último devido ao facto de que nos permitia ter uma maior eficiência na nossa aplicação. Apesar de termos também consciência de que o algoritmo implementado não tem grande dificuldade, estamos satisfeitos com o facto de termos conseguido desenvolver e implementar uma aplicação de encriptação e desencriptação de ficheiros para a nuvem, pois conseguimos compreender melhor um novo modelo de computação que cada vez mais se adivinha o futuro.

Este artigo é o culminar de uma nova descoberta que queremos agora partilhar com o nosso leitor.

Por esse mesmo motivo, todos os ficheiros descritos neste trabalho, assim como a jdk utilizada e a versão usada do Glassfish e os anexos deste artigo que poderão ajudar o leitor a instalar e experimentar o funcionamento do WA, estão disponíveis para o download em <http://bit.ly/NYNEBT>

Relembramos ao nosso leitor que este artigo foi desenvolvido com base em documentação oficial da Microsoft e que caso o desejem poderão aprofundar os vossos conhecimentos nesta área.

Agradecimentos

Gostaríamos ainda de agradecer ao professor Nuno Garcia, docente da Universidade da Beira Interior, toda a ajuda prestada.

A PROGRAMAR

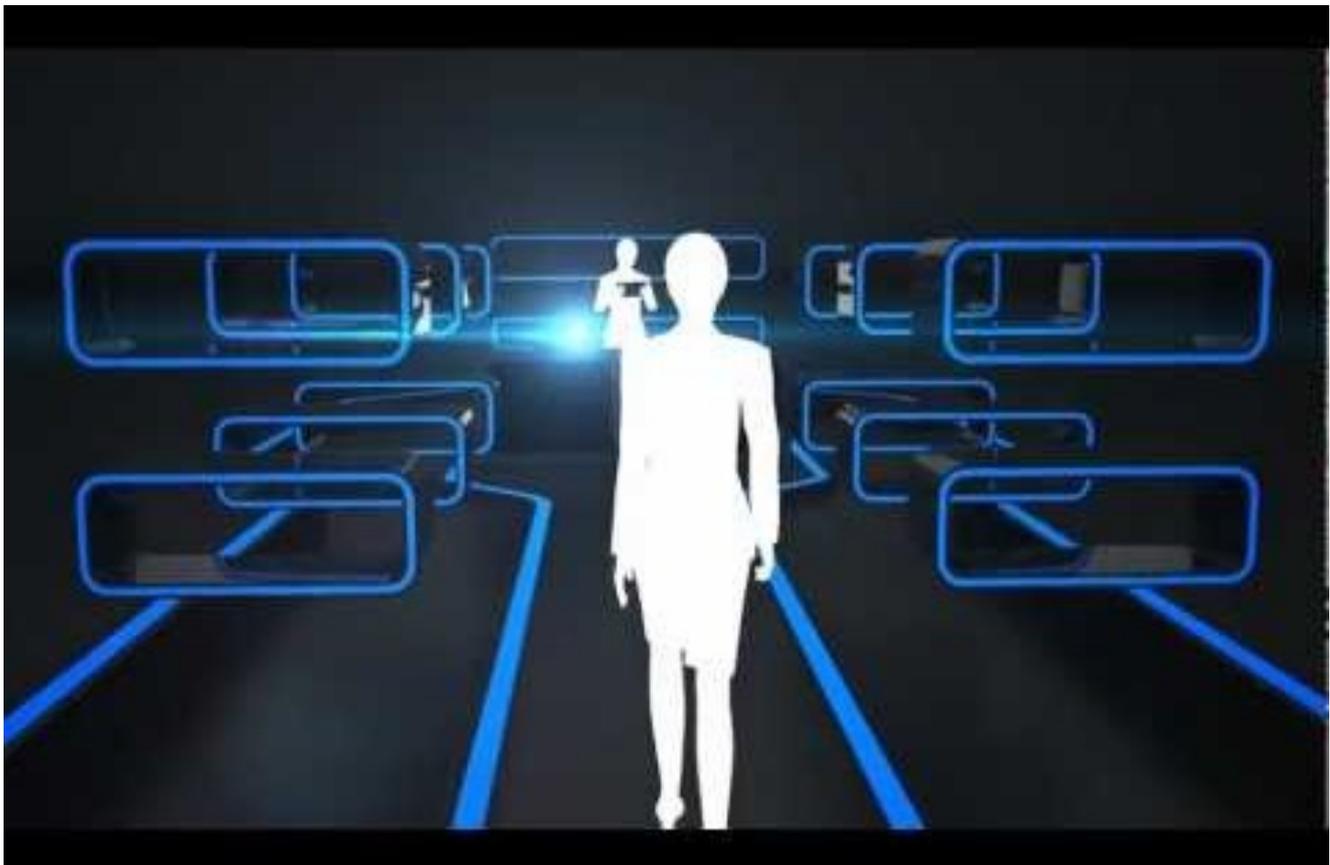
IMPLEMENTAÇÃO DA CIFRA ARSD EM AMBIENTE EMULADO WINDOWS AZURE

Webgrafia

Microsoft Virtual Academy, Apontamentos teóricos do Curso "*Introduction To Windows Azure*", disponibilizados online em <http://bit.ly/fH1Wsp>

Java SE Downloads - <http://bit.ly/bMkbpo>

GlassFish Community - <http://bit.ly/eEgNbg>



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>)



Escrito por André Lourenço

Mais conhecido por KK31, natural de Pinhel, é estudante de Engenharia Informática da Universidade da Beira Interior (<https://www.facebook.com/kk31mdb>)

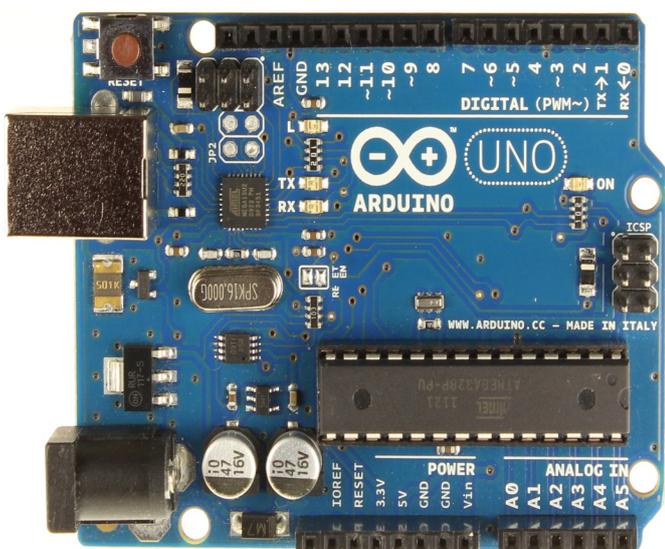
A PROGRAMAR

Estação de Controlo de Temperaturas usando Arduino e Texas Instruments LM335A

Introdução:

Este artigo tem como objectivo demonstrar algumas das capacidades do microcontrolador Arduino quando conjugado com o sensor de temperatura Texas Instruments LM335A. Usando estes dois componentes é relativamente fácil criar uma pequena estação de controlo de temperaturas e posteriormente, mediante uma pequena aplicação programada numa linguagem a gosto (neste caso foi utilizada a linguagem C) poderá ser feito o tratamento estatístico das informações recolhidas pelo Arduino em conjunto com o sensor de temperatura Texas Instruments LM335A.

Arduino Uno:



(Figura 1 – Arduino Uno)

O Arduino Uno é um pequeno microcontrolador baseado no microprocessador ATmega328 a 16MHz. Possui 14 pinos I/O digitais, sendo que 6 deles podem ser usados como PWM, 6 pinos de entrada analógicos e conexão USB para se ligar facilmente ao computador. A linguagem de programação mais usada na programação do microcontrolador é o C/C++.

Microcontrolador	ATmega328
Voltagem de Trabalho	5V
Voltagem Recomendada no Input	7V – 12V
Limites de Voltagens	6V – 20V
Pinos Digitais I/O	14 Pinos(6 deles PWM)
Pinos Analógicos de Entrada	6 Pinos
Corrente DC por Pino I/O	40mA
Corrente DC no Pino 3.3V	50mA
Memória Flash	32Kb (0.5Kb Usados no Bootloader)
SRAM	2Kb
EEPROM	1Kb
Velocidade de Relógio	16MHz

Especificações:

Poderá encontrar todas as informações detalhadas acerca do microcontrolador Arduino Uno no seu datasheet Aqui: <http://bit.ly/At0G9L>

Texas Instruments LM335A:



(Figura 2 – LM335A)

O sensor Texas Instruments LM335A é um pequeno sensor de temperatura capaz de operar em temperaturas que vão dos -40°C até aos 100°C. Quando devidamente calibrado o LM335A tem um erro de leitura de cerca de 1°C, no entanto estes valores podem ser superiores caso não seja feita a devida calibragem do sensor. O Texas Instruments LM335A

funciona numa faixa de corrente que vai dos 5mA até 400mA.

Especificações:

Tensão de Alimentação Mínima	3V
Tensão de Alimentação Máxima	5V
Iq	1000 uA
Aumento na Leitura	10 mV/°C
Precisão Máxima do Sensor	+/- 3°C
Impedância de Saída	0.6 Ohm
Faixa de Temperatura Operacional	-40°C até 100°C

Poderá encontrar todas as informações detalhadas acerca do sensor de temperatura Texas Instruments LM335A no seu datasheet Aqui: <http://bit.ly/ORLAhN>

Material Necessário:

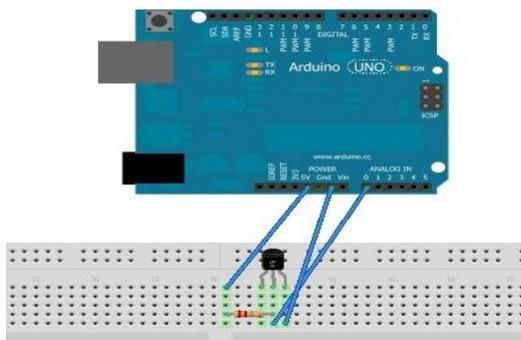
Para a realização deste pequeno projecto foi necessário o seguinte material:

- Breadboard
- Arduino Uno
- Texas Instruments LM335A
- 1 Resistência de 2KΩ
- 4 fios de conceção

Esquema Eléctrico:

A montagem deste pequeno projecto é relativamente simples de ser feita uma vez que são usados poucos componentes e ligações.

A montagem deve ser feita seguindo o seguinte esquema eléctrico:



(Figura3 – Esquema Eléctrico)

Deve ser tomada atenção na ligação dos componentes evitando sempre efectuar qualquer alteração com o Arduino conectado à fonte de alimentação prevenindo qualquer possível curto-circuito.

Programação do Arduino Uno + Texas Instruments LM335A

Após efetuarmos toda a parte física do projecto é necessário programar o Arduino para que consiga ler e interpretar os dados obtidos pelo sensor de temperatura. Obter e interpretar os dados obtidos é relativamente simples, bastando recorrer ao seguinte código-fonte:

```
float temp;
int tempPin = 0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  temp = analogRead(tempPin);
  temp = (((temp/1023)*5)*100)-273.15;
  Serial.print("A Temperatura é de: ");
  Serial.println(temp);
  delay(1000);
}
```

Para quem já está habituado a programar ou pelo menos tem algumas noções este pequeno pedaço de código escrito em C é de fácil interpretação, com exceção da linha 12 que é mais específica para a obtenção da temperatura correcta, o que vou explicar em pormenor já de seguida.

Por defeito o sensor de temperatura Texas Instruments LM335A lê a temperatura em Kelvin, sistema de medição que não é o mais usado por nós, comuns Portugueses.

É importante referir que os valores de output do sensor de temperatura LM335A são expressos em mV, sendo posteriormente necessário converter este valor para Volts de modo a que o Arduino saiba como lidar e interpretar a informação recebida.

Os valores lidos irão tomar valores de 0 até 1023, então nesse caso iremos ter $(temp*5volts)/1023$, mas como o output do LM335A é em mV temos de multiplicar ainda o valor por 100 para ficar em Volts.

Após isto obtemos o valor da temperatura em Kelvin, mas como o nosso objectivo é obter temperaturas em °C temos de subtrair 273.15, obtendo assim por fim a seguinte linha de código:

```
temp = (((temp/1023)*5)*100)-273.15;
```

A PROGRAMAR

ESTAÇÃO DE CONTROLO DE TEMPERATURAS

Pequena tabela de conversão de temperaturas:

0°k = -273.15°C

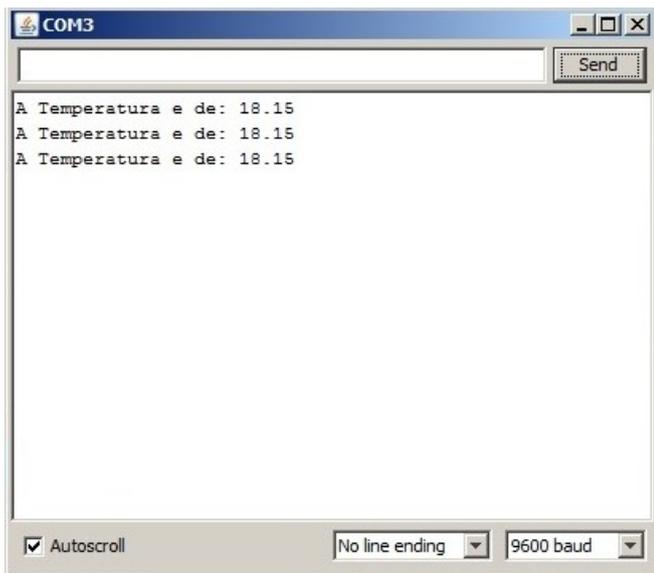
0°C = 271.15°K.

Arduino IDE + SerialMonitor

A programação do Arduino deve ser feita preferencialmente recorrendo ao seu IDE, o que facilita bastante a tarefa de compilar e enviar o código para o microcontrolador.

Após enviarmos o código acima indicado para o microcontrolador vamos verificar se o Arduino em conjunto com o sensor de temperatura estão a produzir os resultados pretendidos. Para isso podemos usar a ferramenta incorporada no Arduino IDE, o SerialMonitor.

O SerialMonitor não faz mais do que apresentar o output dos dados recebidos e interpretados pelo Arduino, e fá-lo da seguinte forma:



(Figura4 – SerialMonitor)

Como podem ver na figura acima apresentada o SerialMonitor é apenas um auxílio na verificação do Output do Arduino.

Automatizando a leitura e armazenamento dos dados num ficheiro txt

O processo de recolha e armazenamento dos dados pode ser feito de várias formas, mas para o projecto em causa tornou-se necessário encontrar uma forma que possibilitasse a automatização deste processo guardando todos os dados recolhidos pelo Arduino e armazenando-os num ficheiro de txt de modo a serem posteriormente processados.

Para efectuar esta tarefa utilizei um script escrito em VBScript que permite recolher os dados lidos pela porta COM do Arduino e efetua o seu armazenamento num ficheiro txt pre-

viamente escolhido.

```
Const ForReading = 1 'declara uma constante com o
'nome ForReading e atribui-lhe o valor 1
Const ForWriting = 2 'declara uma constante com o
'nome ForWriting e atribui-lhe o valor 2
Set fso = CreateObject
("Scripting.FileSystemObject") 'cria um objeto do
tipo "Scripting.FileSystemObjcet com o nome fso
Set com = fso.OpenTextFile("COM3:9600,N,8,1",
ForReading) ' cria um objeto do tipo
'Scriping.FileSystemObject, utiliza a proprieda-
de OpenTextFile do objeto com os parâmetros
"COM3:9600,n,8,1, 2 em que COM3 é a porta serie 3,
9600 é a velocidade de comunicação em bps, n é a
paridade, 8 é o numero de bits de dados, 1 é o nu-
mero de bits de paragem" e 2 que é o valor da cons-
tante ForReading que indica que é para leitura
Set objFSO = CreateObject
("Scripting.FileSystemObject") 'cria um objeto do
'tipo "Scripting.FileSystemObjcet com o nome fso
Set objFile = objFSO.OpenTextFile
("C:\Utilizadores\Nuno\TempLogger.txt", ForWriting,
True) ' cria um objeto do tipo
'Scripting.FileSystemObject, utiliza a propriedade
'OpenTextFile do objeto com os parâmetros
"C:\Utilizadores\Nuno\TempLogger.txt" , 2 (que é o
'valor da constante ForWriting e True que indica
'que define a propriedade de escrita para
"escrever"
MsgBox("Iniciado TempLogger") ' apresenta uma caixa
'de mensagem
Do While com.AtEndOfStream <> True ' inicia um
'ciclo do tipo Do While em que enquanto
'com.atEndOfStrem for diferente de true, repete as
'instruções dentro do loop (enquanto não chegar ao
'fim da stream de dados)
temperatura = com.ReadLine 'temperatura toma o
'valor de com.ReadLine
objFile.Write(now()) 'escreve a data e hora no
'preciso momento
objFile.Write(" ") ' escreve um espaço no ficheiro
'objFile.WriteLine(temperatura) 'escreve o valor de
'da variável temperatura
WScript.Sleep(100) 'aguarda 100ms
Loop 'repete o ciclo
objFile.Close 'fecha o ficheiro
com.Close() 'fecha a comunicação com a porta
'pelo que entendo este script emula um ficheiro em
'memória para receber dados de uma porta serie e
'cria um ficheiro na filesystem do windows, para os
'armazenar
```

(O script está acima descrito e comentado)

Com o uso deste pequeno script torna-se muito mais fácil e rápido de recolher os dados uma vez que por defeito o Arduino não tem a capacidade de guardar os seus próprios dados, sendo necessário recorrer a uma aplicação externa, neste caso o script VBScript, podendo ainda ser usada uma aplicação do tipo Termite 2.8 ou semelhante.

Criar uma pequena Aplicação em C para tratamento estatístico dos dados

O objectivo da informática é gerar informação automática e é seguindo esse pressuposto que sentimos necessidade de programar uma pequena aplicação, neste caso recorrendo à linguagem de programação C para que possa ser feito o tratamento estatístico dos dados recolhidos e armazenados de uma forma mais simples e rápida, tirando todo o proveito da capacidade computacional do computador.

Aqui está o código-fonte da pequena aplicação em linguagem C que desenvolvi:

```
#include <stdio.h>
#include <stdlib.h>
#define N 1000
int ler_ficheiro(int *dia, int *mes, int *ano, int
    *hora, int *min, int *seg, float *temperatura)
{
    FILE *entrada;
    int i = 0;
    int num_linhas = 0;
    entrada = fopen("TempLogger.txt", "r");
    if (entrada == NULL)
    {
        printf("Erro na Abertura do Ficheiro de Entra
            da:\n\n");
        system("PAUSE");
    }
    else
    {
        while (!feof(entrada))
        {
            fscanf(entrada, "%d-%d-%d %d:%d:%d A
                Temperatura é de: %f\n", &dia[i], &mes[i],
                &ano[i], &hora[i], &min[i], &seg[i],
                &temperatura[i]);
            printf("%d-%d-%d %d:%d:%d A Temperatura é de:
                %f\n", dia[i], mes[i], ano[i], hora
                [i], min[i], seg[i], temperatura[i]);
            i++;
        }
        return i;
    }
}
int media_temp(int *dia, int *mes, int *ano, int
    *hora, int *min, int *seg, float *temperatura)
{
    float soma_temp = 0, temp_media = 0;
    int j, num_linhas = 0;
    num_linhas = ler_ficheiro(dia, mes, ano, hora,
        min, seg, temperatura);
    for(j = 0; j < num_linhas; j++)
    {
        soma_temp = soma_temp + temperatura[j];
        temp_media = soma_temp / num_linhas;
    }
    printf("Media = %f", temp_media);
}
```

```
}
float temp_max(int *dia, int *mes, int *ano, int
    *hora, int *min, int *seg, float *temperatura)
{
    float temp_max = 0;
    int j, num_linhas = 0;
    num_linhas = ler_ficheiro(dia, mes, ano, hora,
        min, seg, temperatura);
    for(j = 0; j < num_linhas; j++)
    {
        if (temperatura[j] > temp_max)
        {
            temp_max = temperatura[j];
        }
    }
    printf("Temperatura Maxima = %f", temp_max);
}
float temp_min(int *dia, int *mes, int *ano, int
    *hora, int *min, int *seg, float *temperatura)
{
    float temp_min;
    int j, num_linhas = 0;
    num_linhas = ler_ficheiro(dia, mes, ano, hora,
        min, seg, temperatura);
    for(j = 0; j < num_linhas; j++)
    {
        temp_min = temperatura[0];
        if (temperatura[j] < temp_min)
        {
            temp_min = temperatura[j];
        }
    }
    printf("Temperatura Maxima = %f", temp_min);
}
int main()
{
    int dia[N], mes[N], ano[N], hora[N], min[N],
        Seg[N];
    float temperatura[N];
    int num_linhas;
    int escolha;
do
{
    printf("\t\t\t\t\t***MENU***");
    printf("\n\n");
    printf("1 - Ler Ficheiro de Dados:\n");
    printf("2 - Calcular a Média das
        Temperaturas:\n");
    printf("3 - Verificar Temperatura Maxima:\n");
    printf("4 - Verificar Temperatura Minima:\n");
    printf("5 - SAIR\n");
    printf("\n");
    printf("Opcao:");
    scanf("%d", &escolha);
    switch(escolha)
    {
        case 1:
        {
            printf("\n\n1- Escolheu Ler o Ficheiro de Da
                dos:\n\n");
            ler_ficheiro(dia, mes, ano, hora, min, seg,
                temperatura);
        }
    }
}
```

A PROGRAMAR

ESTAÇÃO DE CONTROLO DE TEMPERATURAS

```
num_linhas = ler_ficheiro(dia, mes, ano, hora,
                        min, seg, temperatura);
printf("Num Linhas = %d", num_linhas);
printf("\n\n");
break;
}
case 2:
{
printf("\n\n2 - Escolheu Calcular a Media das
      Temperaturas:\n\n");
media_temp(dia, mes, ano, hora, min, seg,
temperatura);
break;
}
case 3:
{
printf("\n\n3 - Verificar Temperatura
      Maxima:\n\n");
temp_max(dia, mes, ano, hora, min, seg,
temperatura);

break;
}
case 4:
{
printf("\n\n3 - Verificar Temperatura
      Minima:\n\n");
temp_min(dia, mes, ano, hora, min, seg,
temperatura);

break;
}
case 5:
{
printf("Escolheu SAIR:");
exit(0);
}
default:
{
printf("\n\n Opção Errada:");
return main();
}
}
}
while(escolha != 5);
}
```

A aplicação desenvolvida para este pequeno projecto é bastante simples e rudimentar, mas é o suficiente para que possa fazer o tratamento estatístico de um conjunto de dados.

A pequena aplicação será capaz de verificar qual a temperatura máxima e mínima do conjunto de dados assim como a

sua média das temperaturas de todo o conjunto de dados recolhidos.

A aplicação irá ler dados armazenados num ficheiro de dados .txt que será o output do script VBS com a seguinte estrutura:

```
18-07-2012 17:18:26 A Temperatura é de: 26.46
18-07-2012 17:18:26 A Temperatura é de: 25.97
```

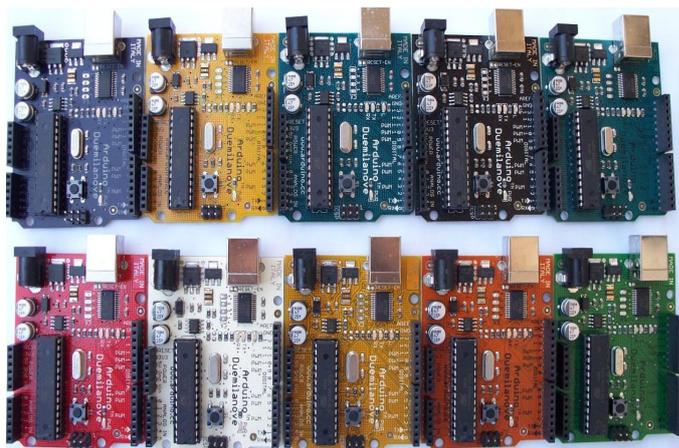
A aplicação desenvolvida para este pequeno projecto é bastante simples e rudimentar, mas é o suficiente para que possa fazer o tratamento estatístico de um conjunto de dados.

A pequena aplicação será capaz de verificar qual a temperatura máxima e mínima do conjunto de dados assim como a sua média das temperaturas de todo o conjunto de dados recolhidos.

Conclusão:

O micro-controlador Arduino é um dispositivo muito versátil que permite criar um sem número de projectos, podendo ser aliado a um enorme conjunto de outros dispositivos. No caso apresentado acima é fácil perceber que o Arduino é uma plataforma muito simples mesmo para quem não tem conhecimentos muito aprofundados e ainda está a dar os primeiros passos, quer na área da electrónica quer no campo da programação.

A mini-aplicação escrita em linguagem C é muito simples, mas demonstra as potencialidades do projecto, podendo qualquer pessoa adicionar novas funcionalidades, completando a aplicação mediante as necessidades.



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/>

COLUNAS

[Enigmas de C# - Avaliação da Lista de Argumentos](#)

[CoreDump - Podemos continuar a acreditar nas TIC?](#)

ENIGMAS C#

ENIGMAS DO C# - AVALIAÇÃO DA LISTA DE ARGUMENTOS

Resultado (continuação da página 34)

O resultado vai depender da versão de C# que se está a usar.

Usando C# 4.0, o resultado será:

```
x=1, y=2, z=3  
x=5, y=20, z=4
```

Usando C# 5.0, o resultado será:

```
x=1, y=2, z=3  
x=4, y=20, z=5
```

Explicação

O conjunto de argumentos de invocação de um método designa-se por **lista de argumentos** (§7.5.1 Argument lists).

A avaliação dos valores dos argumentos (§7.5.1.2 Run-time evaluation of argument lists) durante a invocação de um método ocorre pela ordem por que são especificados (da esquerda para a direita).

Todos os efeitos colaterais (*side effects*) ocorrem antes da invocação do método do lado do chamador.

As diferenças entre o C# 4.0 e C# 5.0 resultam do facto de existir uma falha no compilador 4.0 que avalia primeiro os argumentos com nome e só depois os argumentos posicionais.

Conclusão

Em face ao demonstrado, a forma mais segura (para evitar enganar) de escrever código é efetuar os cálculos dos valores dos argumentos fora da especificação da lista de argumentos:

Ligações

[C# Reference](#)

[Visual C# Breaking Changes in Visual Studio 2012](#)

[Argumentos com Nome e Opcionais](#)



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.

Podemos continuar a acreditar nas TIC?

Tive o prazer de ser orador na versão portuguesa do evento europeu [eSkills 2012](#) como representante da comunidade P@P, organizado pela [ANPRI](#).

Sob o tema “Podemos continuar a acreditar nas TIC?”, ficou patente de que **as TIC são, e devem continuar a ser, um ponto de partida importante para criar bons profissionais.**

O evento juntou professores, alunos, antigos alunos e profissionais e mostrou que as TIC têm um papel que se pode tornar mais relevante na formação de profissionais para a área das TI.

Alguns pontos onde a indústria está claramente de acordo foram focados e debatidos na altura das questões. Esses pontos prendem-se com a necessidade dos profissionais terem no seu curriculum um fator diferenciador, com a inevitável constante atualização tecnológica necessária e com a imprescindível pro-atividade intrínseca à profissão.

Na comunidade P@P alguns destes temas são recorrentes, sendo do interesse particular por quem está a concluir o secundário e se questiona “e agora, que opções tenho”. Durante vários anos tenho defendido, inclusive no fórum da comunidade, que é importante que os alunos tenham a consciência de que devem sempre escolher a melhor universidade, caso pretendam continuar o seu percurso académico, e que devem ter sempre um fator diferenciador que os distinga da

generalidade dos seus pares.

Estes pontos tornam-se ainda mais relevantes quando o mercado de trabalho atinge níveis de desemprego e baixa procura como é, infelizmente, o panorama atual.

Já lá vão mais de duas décadas desde que se começou a ensinar informática no secundário. Evoluímos do ensino do COBOL e do GW BASIC para PHP, SQL e Photoshop, criando assim nos alunos competências importantes para os dias de hoje para quem pretende fazer carreira em TI.

Sem dúvida que as TIC têm um papel importante neste ponto, uma vez que é da sua competência ajustar os conteúdos do curriculum às necessidades reais do mercado de trabalho nacional e, atualmente, parecem-me alinhados. Quanto ao futuro, as notícias recentes indicam que as TIC vão começar no 7º ano de escolaridade, mostrando que existe consciência sobre a importância das TIC na preparação dos alunos.

Assim, a minha resposta ao título do artigo é **sim, podemos continuar a acreditar nas TIC.**

Bastando para tal que as TIC, tal como os profissionais, sejam pro-ativas e se atualizem de forma a poderem continuar a proporcionar aos seus formandos um bom ensino atualizado.



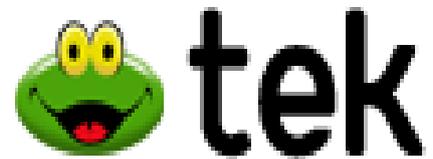
AUTOR



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

Fundamental do MAC OS X

Microsoft SQL Server 2012 Integration Services: An Expert Cookbook

Fundamental do MAC OS X

Título: Fundamental do MAC OS X

Autor: Rui Santos

Editora: FCA

Páginas: 280

ISBN: 978-972-722-734-1

No primeiro contacto com o livro fiquei um pouco aquém do esperado pois para falar de apenas de um sistema operativo mais apontado para designers e altamente conhecido pela elegância dos equipamentos, fluidez, facilidade de trabalho e beleza do próprio sistema operativo o livro não é muito apelativo. Mas sigamos em frente.

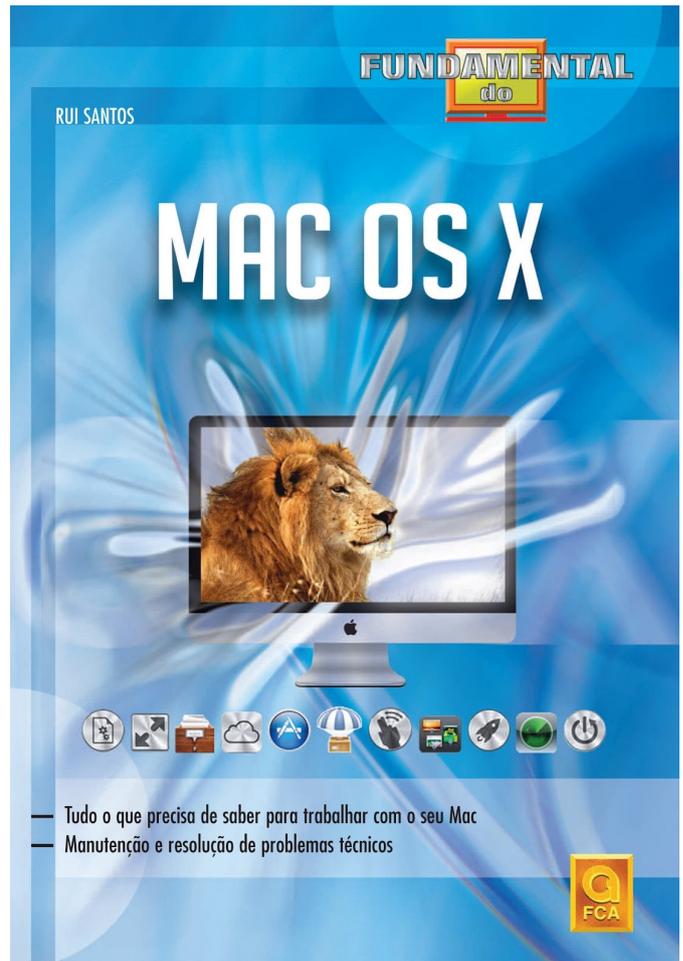
Através dos seus doze capítulos o Autor passa por vários assuntos dos mais básicos até aos mais exigentes; em todo o processo de escrita eu penso que o Autor não fez a introdução mais certa em forma de enquadramento ao utilizador que não tem conhecimento básico e prévio do Mac OS X.

No meu ponto de vista, penso que a leitura ajudara todos que os leitores mas, ajuda bastante mais quem já tem alguns conhecimentos práticos no sistema operativo, sendo bastante explicativo desde o mais básico até ao mais complexo.

É um excelente livro para o conhecimento do sistema operativo a nível de usabilidade e acessibilidade, faz nos lembrar um pouco aquelas aulas de TIC que se tinham a anos atrás que é saber nomes de barras, e como fazer o básico no Mac OS X.

Em suma um excelente livro para quem entra no "Mundo Mac" ter como apoio a todo e qualquer duvida de usabilidade que posso aparecer durante a habituação e iniciação, livro inquestionável na estante de quem tem um Mac pois tudo que possamos duvidar ou querer saber sobre o Mac OS X este livro responde-nos. Como já disse a cima é um livro excelente para todos, sendo sincero com os leitores desta Review acho que posso intitular este livro como que a

"Enciclopédia do Mac OS X", mas nunca como iniciação ao Mac OS X.



AUTOR



Escrito por Miguel Oliveira

Técnico de informática nível IV na Escola Profissional de Mazagão, Braga, actualmente frequenta o CET no IPVC de Tecnologia e Programação de Sistemas Informação. Membro da equipa do IPVC que promoveu a 1ª Reunião Presencial da Comunidade NetPonto em Viana do Castelo, Membro do Staff da Revista PROGRAMAR desde Junho de 2012.

Microsoft SQL Server 2012 Integration Services: An Expert Cookbook

Título: Microsoft SQL Server 2012 Integration Services: An Expert Cookbook

Autor: Reza Rad, Pedro Perfeito

Editora: PACKT Publishing

Páginas: 564

ISBN: 9781849685245

Quando comecei a ler o livro, confesso que estava expectante, a pensar: será este apenas mais um livro técnico sobre Microsoft SQL Server ?

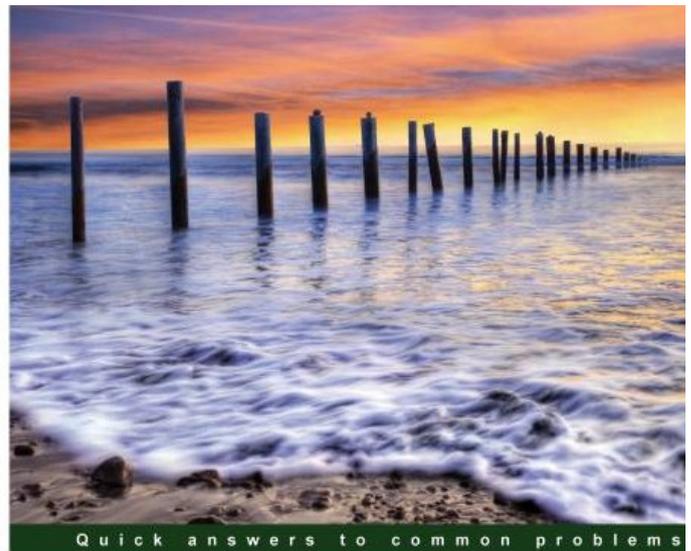
Demorei algumas páginas até formalizar uma opinião mais concreta! Na realidade não é apenas mais um livro, é um excelente livro, que aborda de forma simultaneamente simples e detalhada, diversos aspectos do MS SQL Server 2012 que apesar de serem de grande utilidade numa boa parte das implementações de software são descurados, ou passados para “segundo plano”, levando o programador a “repetir trabalho” que já existe feito, ou existe forma mais simples de o fazer.

Depois de algumas horas e páginas “livro adentro” dei por mim a instalar uma versão trial do SQL Server para poder experimentar na prática alguns exercícios da vasta série de exercícios detalhados passo a passo que se encontram ao longo do livro. Cada exercício mais “desafiador” que o anterior mas mais interessante!

Ao longo de toda a leitura, que a determinado momento se tornou “apaixonante”, como se na realidade estivesse a ler um “thriller” e não um livro técnico, senti-me levado a explorar melhor as potencialidades do SQL Server, nos vários níveis e com especial interesse no seu uso num nível de inteligência de negócio, e não apenas como SGBD, desprovido de outras ferramentas, em que todos os processos de nível estratégico e de inteligência teriam de ser feitos no software cliente sem retirar grande proveito das potencialidades das ferramentas de base de dados.

Ao terminar a leitura de todo o livro, e tendo experimentado alguns dos exercícios nele apresentados e feito as minhas próprias “variantes” dos exercícios, apenas pude concluir

que quem ler este livro, não só está a aumentar o seu nível de conhecimento (seja ele qual for), pois o livro é bastante acessível e os exemplos bastante fáceis de seguir, como está a adquirir conhecimentos que são válidos e bem úteis mesmo para além das ferramenta Microsoft SQL Server 2012 Business Intelligence Edition e Visual Studio 2012.



Microsoft SQL Server 2012 Integration Services: An Expert Cookbook

Over 80 expert recipes to design, create, and deploy SSIS packages

Foreword by Phil Brammer, Microsoft MVP – SQL Server

Reza Rad

Pedro Perfeito

[PACKT] enterprise
PUBLISHING professional expertise distilled

AUTOR



Escrito por António Santos

Entusiasta da tecnologia desde tenra idade, cresceu com o ZX Spectrum, autodidacta, com uma enorme paixão por tecnologia, tem vasta experiência em implementação e integração de sistemas ERP, CRM, ERM. Membro da Comunidade [Portugal-a-Programar](#) desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário. Neste momento é aluno no Instituto Politécnico de Viana do Castelo, na Escola Superior de Tecnologia e Gestão.

COMUNIDADES

PtCoreSec—Segurança na WEB (Parte 3)

SEGURANÇA NA WEB (PARTE 3)

Iremos concluir nesta terceira parte a nossa trilogia de artigos de programação segura. Vamos começar por explicar a parte final do nosso conjunto de regras. A última regra de que falámos foi a de **autenticação e autorização**, iremos agora prosseguir para a regra de **Gestão de sessões**.



Regra 4 – Gestão de sessões

Quando um utilizador liga a sua aplicação, esta pode pedir credenciais ao utilizador. Se um utilizador conseguir autenticar-se com sucesso, este não espera ter de voltar a inserir as suas credenciais outra vez, a não ser que acabe o tempo de login permitido e a sua sessão expire ou que estejam a tentar executar uma acção privilegiada. Gerência de sessões permite que a sua aplicação autentique um utilizador mas também que garanta que o utilizador a efectuar as acções é o utilizador que se autenticou usando as credenciais originais. Para um atacante, quaisquer falhas na camada de gerência de sessões serve como uma maneira fácil de ultrapassar o trabalho árduo que colocamos nas regras anteriores.

Os ataques contra sessões são normalmente focados na obtenção de uma sessão válida, abusando dos utilizadores, ou tirando vantagem de falhas da funcionalidade da camada de gerência de sessões. O conhecimento dos métodos usados pelos atacantes não é obrigatório se criarmos sessões seguras baseadas nesta regra.

Os valores da sessão usados na aplicação devem seguir princípios semelhantes aos de passwords seguras, como mencionado anteriormente. Os ID's de sessão usados para identificar utilizadores autenticados individuais devem ter tamanho suficiente para prevenir ataques de força bruta – o tamanho deste ID vai ser determinado pela importância dos dados ou recursos que estamos a tentar proteger. Devemos

relembrar que o tamanho de ID da sessão não é suficiente como protecção por si só; é também importante um alto nível de entropia por cada carácter do ID de sessão. A entropia de cada carácter deve ser considerada quando se cria os ID's de sessão, visto que uma entropia mais alta por carácter individual é mais segura. Um ID de sessão deverá ser construído a partir de um conjunto que não permita nenhuns padrões óbvios no ID. Um padrão como por exemplo a letra C, estando sempre presente nas posições 1,4 e 5, seria facilmente identificado por ferramentas automáticas e iria reduzir o tempo necessário de computação para um ataque de força bruta a ID's genuínos.

Se os passos acima forem seguidos correctamente, os utilizadores terão ID's de sessões que não poderão facilmente ser previstos por atacantes.

Em seguida, temos de confirmar que estes ID's são seguros no servidor da aplicação e enquanto estão em trânsito. A localização de armazenamento dos ID's de sessão deverá sempre feita num local seguro e não num local acessível publicamente.

O ponto seguinte que iremos abordar é a transmissão segura de ID's e para isto existe uma resposta simples. Se os ID's forem transmitidos por HTTP estes podem facilmente ser interceptados e reutilizados por um atacante. Assim, usando HTTPS ao invés de HTTP pode-se proteger os ID's de sessões enquanto estão a ser transmitidos.

Os 2 últimos mecanismos de protecção são a aplicação de timeouts nas sessões e mudanças nos ID's de sessão quando os utilizadores executam acções sensíveis. A duração de cada sessão depende no nível de importância dos dados a serem protegidos. Caso se peça uma segunda autenticação, um segundo ID de sessão deverá ser criado e a o original destruído.

Esta regra ajuda a proteger contra: Falhas na autenticação e gerência de sessões, Cross Site Request Forgery.

Regra 5 – Segurança de comunicação

Como mencionado previamente noutras regras, a importância de proteger partes específicas quando estão a ser transmitidas é elevada, pelo que iremos agora analisa-la com maior detalhe. O requisito de protecção de dados em trânsito não é um novo requisito mas cuja implementação por parte das aplicações por vezes falha.

Esta é provavelmente a regra mais simples - é uma questão de ter a certeza que a nossa aplicação obriga o uso de um

COMUNIDADE PTCoreSec

SEGURANÇA NA WEB (PARTE 3)

mecanismo seguro de transporte como SSL, TLS ou SSH. Também deverá ter-se a certeza que a aplicação obriga ao uso específico de versões destes mecanismos como SSL versão 3 ou SSH versão 2.

Se esta regra é tão simples, então porque é que isto continua a correr mal? Estes problemas normalmente originam de duas decisões:

1. Quando usar estes mecanismos;
2. Que versão usar.

A falha mais comum vem da decisão numero 1 - a falha de protecção do início de sessão e a informação da sessão depois da autenticação. A protecção deve ser iniciada logo que o utilizador entre no site, o que significa que as páginas de login deverão ser HTTPS em vez de HTTP. Em vez de se encriptar apenas o login esta protecção deverá ser continuada durante toda a sessão, não apenas para submissão de credenciais. Se os dados forem altamente sensíveis, deverá continuar a usar-se mecanismos de transporte seguro internamente, por exemplo do servidor applicacional para o servidor de bases de dados.

O último ponto a ser analisado é o uso de um mecanismo que é criptograficamente seguro e não tem falhas no design. Um bom exemplo de um mecanismo que não é seguro é o SSLv2; múltiplas vulnerabilidades são consequência de falhas no design e não de falhas criptográficas. Previamente foram mencionados dois mecanismos de protecção que são bons exemplos de como proteger os dados em transmissão. Quando se está a fazer a escolha do mecanismo de protecção que se deve utilizar, deve escolher-se um que é considerado seguro como SSLv3, TLSv1 ou SSHv2.

Regra 6 – Acesso seguro a recursos

Acesso seguro a aplicação já foi um tópico falado múltiplas vezes nas regras anteriores mas iremos agora falar em detalhe sobre outras situações. Os problemas de autenticação e autorização de utilizadores em conjunto com uma gerência segura de sessões já foram abordados anteriormente mas tudo isto pode ser perdido por más de decisões design.

Se um design depende de segurança por obscuridade, então existe uma elevada probabilidade de que este venha a falhar. Uma norma comum para criar um certo nível de segurança 'a volta de localizações sensíveis é esconde-las do utilizador, não publicando quaisquer links para estas localizações. Isto falha como protecção pois existem ferramentas automáticas que irão descobrir estas localizações e permitir aos atacantes acesso directo. Se a localização contem conteúdos sensíveis como backups ou outras funções (exemplo: /admin) deverão usar-se medidas de controlo de acesso fortes de maneira a ter a certeza de que apenas

utilizadores com as correctas permissões acedem a tais funções. A autenticação e autorização não devem ser efectuadas apenas uma vez, mas sempre que um utilizador usa uma função mais sensível. Um exemplo real de falha neste tipo de sistemas, foi quando o site da T-mobile foi hackado em 2005, o que levou a que a conta de Paris Hilton fosse comprometida. O sistema de reset da password no site da T-mobile pedia que um utilizador provasse quem era pedindo lhe para inserir o seu numero de telemóvel, o site então mandaria para esse numero um token único que meteriam no site e avançavam para uma pagina de reset de password. O problema com este design é que se assumia que os utilizadores só iriam aceder a página de reset após autenticação. Um atacante chamado Luckst4w encontrou uma falha que permitia que se acesse directamente á página de reset sem ser necessário nenhum tipo de prova de quem somos.

Deverá sempre assumir que se o seu recurso é acessível por qualquer um dos seus utilizadores, qualquer pessoa irá aceder a esse recurso.

Regra 7 – Armazenamento seguro

Iremos agora analisar como criar armazenamento seguro. Já conseguimos criar segurança 'a volta dos pontos de entrada e saída de dados, implementámos mensagens de erro sanitizadas, criámos controlo de acesso fortes e protegemos a transmissão da nossa informação, mas não podemos negligenciar a segurança dos dados que estão parados.

Os requisitos para armazenamento de dados seguro, como por exemplo dados relativos a cartões de crédito, são óbvios, mas também temos de armazenar dados como passwords e detalhes de sessão seguramente quando não estão em transmissão. Não só temos de identificar os dados que precisam de ser protegidos mas também os mecanismos que se podem usar para os proteger. A selecção do mecanismo de protecção segue o mesmo protocolo de escolha para comunicação segura. Nunca criar o seu próprio mecanismo e não usar mecanismos fracos como DES, MD4 ou SHA-0.

Não queremos que esta regra se torne numa aula de criptografia mas deve-se pelo menos assegurar que os valores seguintes são usados nos mecanismos simétricos, assimétricos e de hash:

- * Simétricos – 256 bits ou mais
- * Assimétricos – 2048 bits ou mais
- * Hashes – 168 bits ou mais

Também se deve usar uma localização segura para armazenamento das chaves de encriptação. Armazenar as chaves no servidor da aplicação normalmente não é considerado seguro. A última parte a não esquecer é não escrever nenhuma das chaves no código fonte.

Regra 8 – Auditoria e logging

Esta é uma regra diferente das anteriores, foi criada apenas como um princípio a seguir em desenvolvimento seguro e não é a causa de uma vulnerabilidade. É importante lembrar que todas as regras anteriores são essencialmente protecções contra as causas principais de vulnerabilidades comuns. Resumidamente, as regras anteriores explicam em detalhe maneiras de prevenir vulnerabilidades comuns no seu código, enquanto que esta regra não está ligada a nenhuma vulnerabilidade comum mas é algo que não pode ser ignorado se quiser criar aplicações seguras. Como vimos com a regra que falava de tratamento de erros/excepções, estas irão ocorrer em todas as aplicações e deverá estar preparado para lidar com elas. Temos de pensar da mesma maneira para Auditoria e Logging, na medida em que é importante criar diversos logs de diferentes eventos criados e usados pela nossa aplicação e acções dos utilizadores.

Quando está a tentar implementar esta regra, é importante perceber quando deverá criar uma entrada nos logs, para onde deve a entrada ser escrita, que informação deve ser contida no log e também que informação não deverá ser contida.

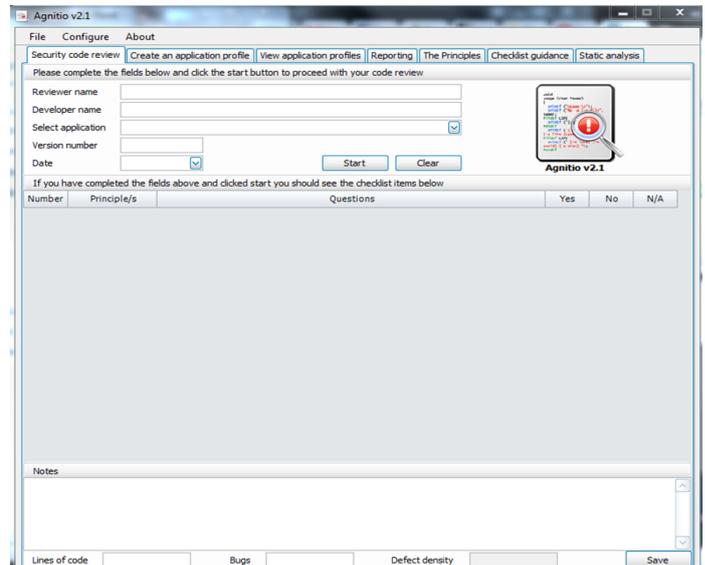
Sempre que possível, os logs deverão ser armazenados centralmente num servidor com controlos de acesso fortes. Os logs deverão ser monitorizados com um produto similar ao Tripwire que irá monitorizar modificações ou tentativas de acesso não autorizadas. As entradas nos logs geradas pela sua aplicação deverão conter informação suficiente que seja útil para análise forense e resolução de problemas resolver problemas e mas não dados sensíveis como dados relativos a cartões de crédito. Deverá sempre que possível tentar entender o que é “informação suficiente” e “dados sensíveis” no seu trabalho. Alguns dos exemplos de coisas que se deve logar são horas, data, nome do computador e endereço IP de onde veio a ligação.

Agnitio – Por a teoria em prática

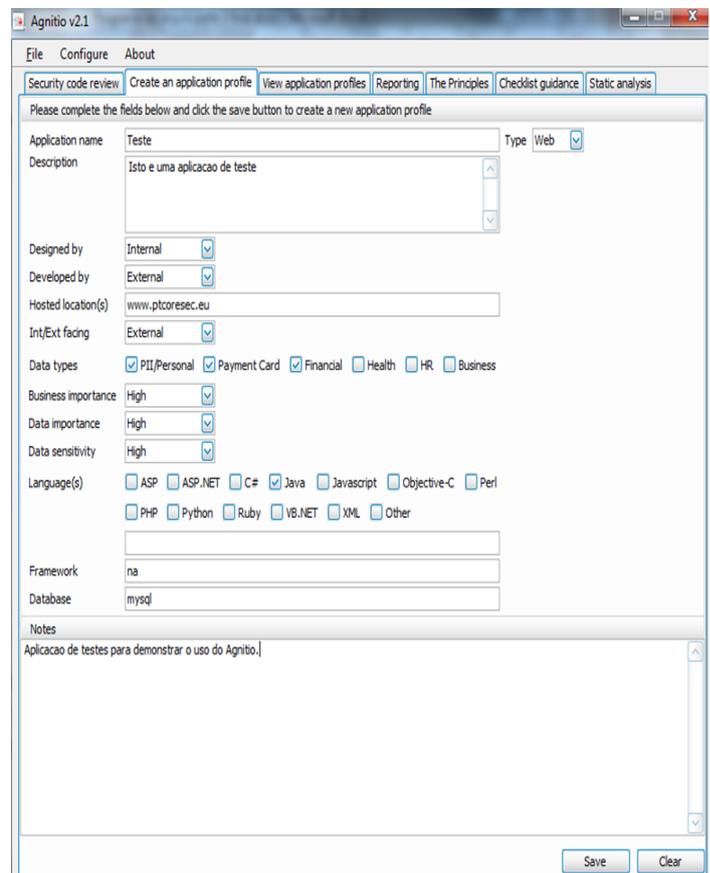
Depois de toda a teoria, chegou a hora de pormos em prática todas estas regras. Nesta parte do artigo iremos falar sobre o Agnitio, um software gratuito usado para auditar e desenvolver software seguro. Podem obter uma cópia deste software em: <http://sourceforge.net/projects/agnitiotool/>.



Depois de instalado o seguinte ecrã é mostrado:



Este é o ecrã principal de checklists no Agnitio. No entanto antes de se começar a usar, temos de criar um profile de aplicação. Para isso dirigimo-nos a 2ª tab e preenchemos todos os campos referentes á aplicação que estamos a auditar como se vê no seguinte exemplo:

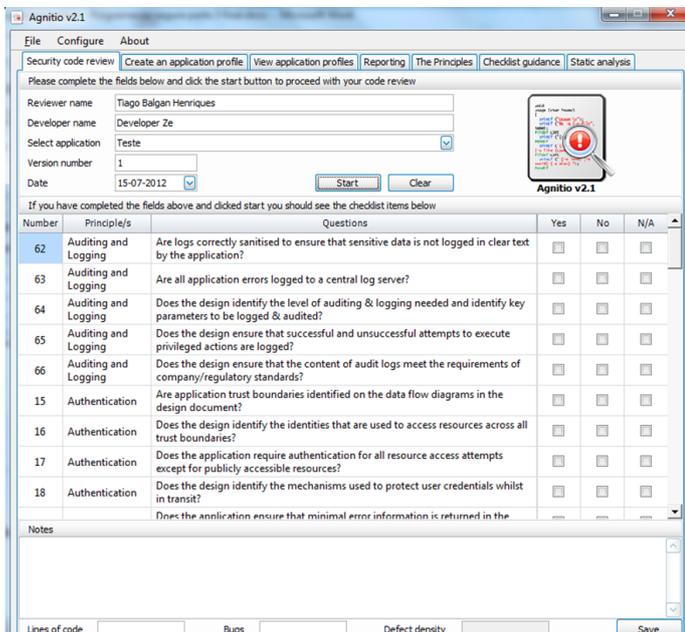


Depois de fazer save podemos seleccionar esta aplicação no

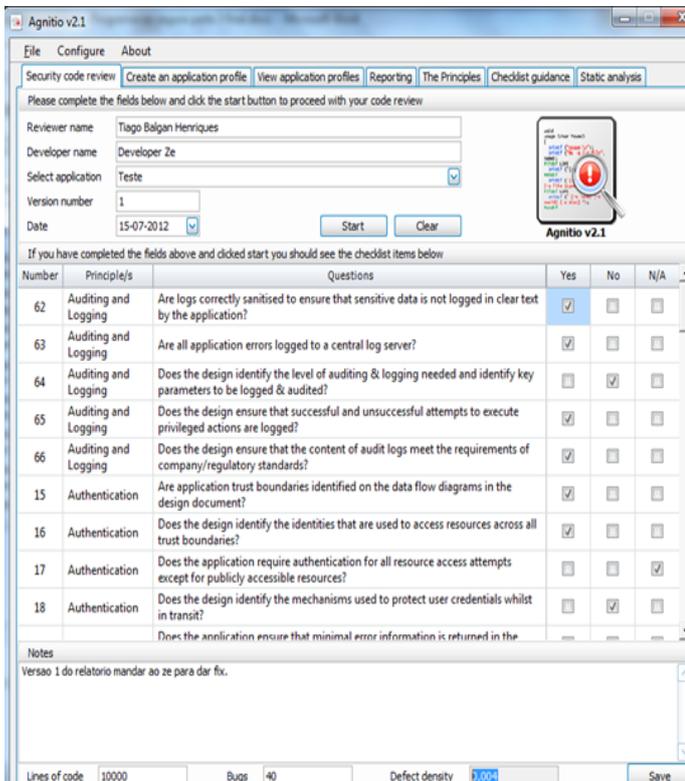
COMUNIDADE PTCoreSec

SEGURANÇA NA WEB (PARTE 3)

menu principal e, depois de preencher todos os campos e clicar em "Start", irá ver um ecrã semelhante a este:

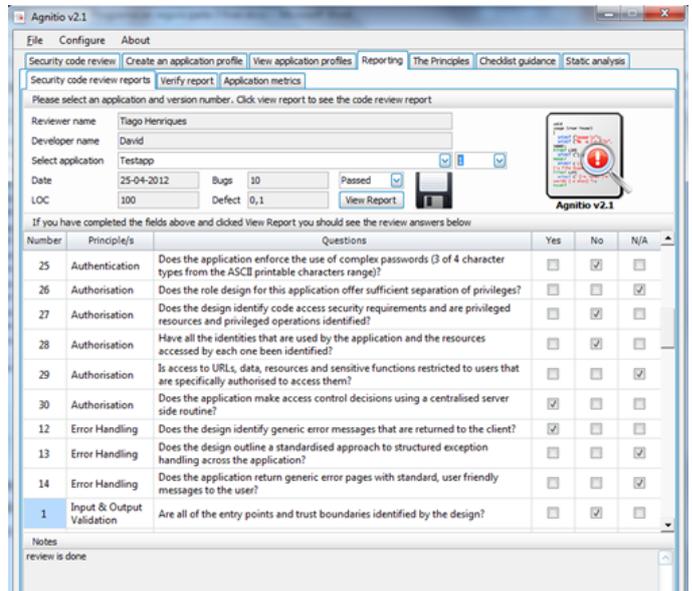


Aqui o Agnitio apresenta um conjunto de regras que são baseadas em conceitos que falámos anteriormente. Depois de tudo preenchido, deverá ter-se algo parecido a:

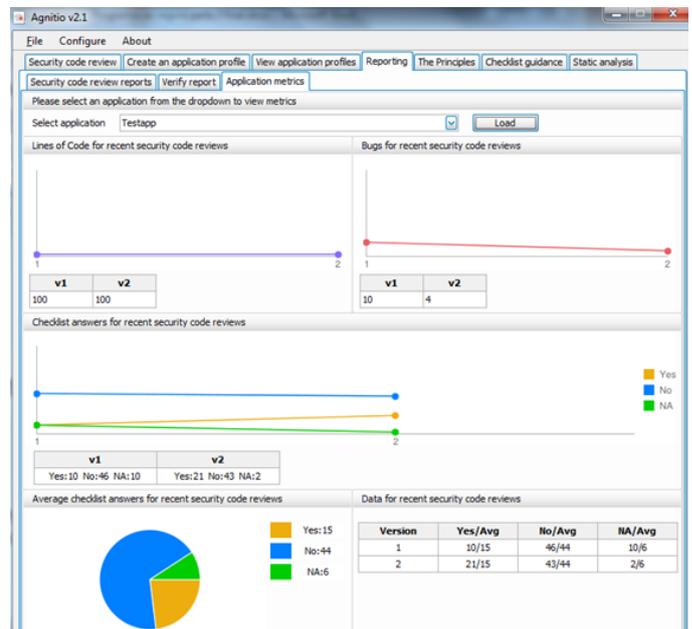


Em seguida, carrega-se em Save. Depois de realizar 2 reviews a uma aplicação poderá usar-se a tab de reporting, na qual podemos ver reports anteriores, comparar as diferentes reviews que realizámos e verificar se o número de erros

detectados diminuiu ou não.



A sub-tab "Application metrics" também mostra algo bastante interessante, nomeadamente uma visualização gráfica das diferenças entre as várias análises:



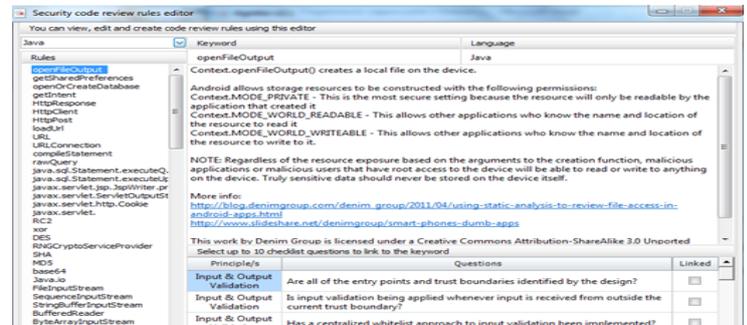
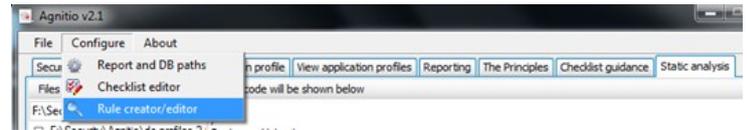
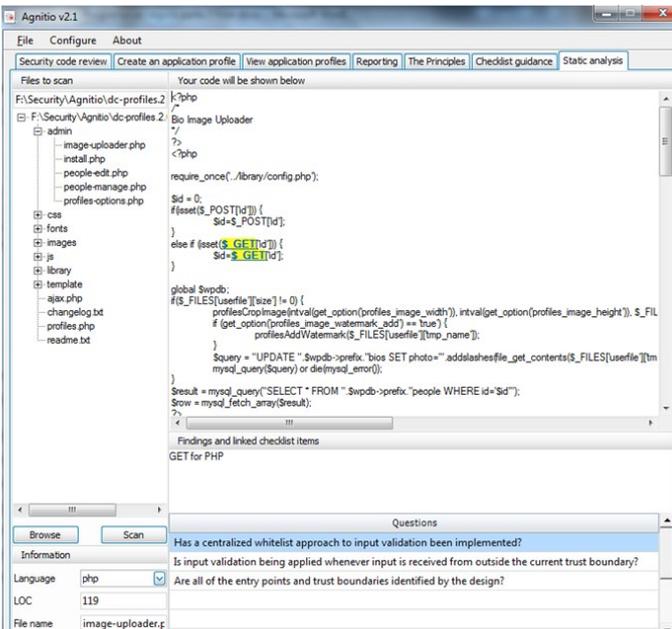
Isto são apenas alguns exemplos do que se pode fazer com o Agnitio.

Vamos agora verificar a última tab chamada "Static analysis". Nesta tab podemos escolher qual a linguagem que estamos a auditar, o Agnitio irá então dar load de um conjunto de regras para essa linguagem e podemos escolher um projecto. Em seguida, podemos, de maneira automática, auditar ficheiros, nos quais o Agnitio irá procurar keywords que foram seleccionadas para esta linguagem e onde normalmente se encontram vulnerabilidades. Ao encontrar

COMUNIDADE PTCORESEC

SEGURANÇA NA WEB (PARTE 3)

estas keywords, o Agnitio irá evidenciar a sua presença no código mudando a sua font para azul e o fundo para amarelo e, quando clicamos nestas keywords, o Agnitio mostra um conjunto de perguntas de perigo associadas a estas funções.



Estes são apenas alguns exemplos de funcionalidades do Agnitio. É possível também gerar relatórios e exportar em diversos formatos. Para mais informação sobre o Agnitio, dirijam-se a <http://bit.ly/MVxr1o> e <http://bit.ly/MfiTM1>

para verem apresentações sobre programação segura e o Agnitio.

O bom do Agnitio é que disponibiliza uma forma de qualquer utilizador possa criar o seu próprio conjunto de regras, não estando por isso limitado às regras que se encontram por defeito na base de dados do Agnitio (C#, Java, Objective-C, e PHP). Desta forma, caso escreva código Ruby, pode criar o seu próprio conjunto de regras para Ruby .

Conclusão:

Apresentámos durante esta trilogia de artigos, diferentes vulnerabilidades, bem como um conjunto de regras para evitar/corrigir estas vulnerabilidades. Mostrámos também software que nos pode auxiliar neste processo e permite a geração de relatórios de modo a que todas as auditorias fiquem organizadas. Agora cabe a vocês seguirem estas

AUTOR



Escrito Por **Tiago Henriques** @balgan, balgan@ptcoresec.eu

Equipa PTCORESEC:

<http://www.ptcoresec.eu>

PTCoreSec

No Code

Plano Tecnológico da Educação: Seis anos depois

O empreendedorismo como uma opção de auto-emprego e inconformismo

Projecto em Destaque na Comunidade Portugal –a– Programar : Notification Agenda

Plano Tecnológico da Educação: Seis anos depois

Em 24 de Novembro de 2005, após um trabalho alargado de recolha de ideias e contributos das diversas áreas do Governo e da sociedade civil, levado a efeito pela Unidade de Coordenação do Plano Tecnológico, o Conselho de Ministros aprovou um documento de referência e compromisso público, visando a aplicação de uma estratégia de crescimento e competitividade baseada no conhecimento, na tecnologia e na inovação.

O Plano Tecnológico era uma agenda de mudança para a sociedade portuguesa, que visava mobilizar as empresas, as famílias e as instituições para que, com o esforço conjugado de todos, pudessem ser vencidos os desafios de modernização que Portugal enfrentava. Indicava uma estratégia de desenvolvimento e competitividade assente em três eixos:

Conhecimento - Qualificar os portugueses para a sociedade do conhecimento, fomentando medidas estruturais vocacionadas para elevar os níveis educativos médios da população, criando um sistema abrangente e diversificado de aprendizagem ao longo da vida e mobilizando os portugueses para a Sociedade de Informação.

Tecnologia - Vencer o atraso científico e tecnológico, apostando no reforço das competências científicas e tecnológicas nacionais, públicas e privadas, reconhecendo o papel das empresas na criação de emprego qualificado e nas actividades de investigação e desenvolvimento (I&D).

Inovação - Imprimir um novo impulso à inovação, facilitando a adaptação do tecido produtivo aos desafios impostos pela globalização através da difusão, adaptação e uso de novos processos, formas de organização, serviços e produtos.

Caracterização Geral do Plano Tecnológico da Educação

Tendo presente que estudos internacionais demonstram uma correlação positiva entre a utilização das tecnologias de informação e comunicação (TIC) em contexto de sala de aula e o aproveitamento escolar dos alunos, o Plano Tecnológico da Educação definiu como principal objetivo colocar, até 2010, Portugal entre os cinco países europeus mais avançados na modernização tecnológica do ensino.

Assim, o Plano Tecnológico da Educação pretendia ser um meio para:

- A melhoria do ensino e dos resultados escolares dos alunos;

- A igualdade de oportunidades no acesso aos equipamentos tecnológicos;
- A modernização das escolas, possibilitando que os estabelecimentos de ensino funcionem em rede e que os professores trabalhem de forma colaborativa.

O Plano Tecnológico da Educação tinha como metas fundamentais:

- Ligar todas as escolas à Internet em banda larga de alta velocidade. Todas as escolas com uma ligação de pelo menos 48 Mbps;
- Atingir um rácio de dois alunos por computador;
- Formar e certificar 90 por cento dos docentes em tecnologias da informação e da comunicação.

Para o cumprimento das metas definidas, o Plano Tecnológico da Educação, aprovado em agosto de 2007, apostava em três eixos de actuação: tecnologia, conteúdos e formação.

Eixo Tecnologia	Eixo Conteúdos	Eixo Formação
Kit tecnológico	Portal da escola	Competências TIC e Formação e Certificação
Internet de banda larga de alta velocidade	Escola Simplex e Plataforma de gestão escolar	Avaliação electrónica
Internet na sala de aula e Redes de área local	Portal institucional do Ministério da Educação	Estágios TIC
Cartão da Escola		Academias TIC
Escol@ segura e Videovigilância		

Tabela 1. Eixos do plano tecnológico

1. Eixo Tecnologia

A insuficiência das infraestruturas de TIC constituía o principal fator inibidor da utilização de tecnologia no ensino.

As medidas do Plano Tecnológico da Educação implementadas no eixo «Tecnologia» pretendiam dar resposta às principais barreiras observadas em termos de infra-estruturas e acessos, nomeadamente a falta de computadores e outros equipamentos de apoio nas escolas, como impressoras, videoprojectores e quadros interactivos, e o seu carácter obsoleto em muitas situações, a reduzida acessibilidade aos equipamentos por parte de alunos, docentes e não docentes, o acesso à Internet não

No Code

PLANO TECNOLÓGICO DA EDUCAÇÃO: SEIS ANOS DEPOIS

generalizado a todos os computadores, com velocidades insuficientes e abrangendo áreas limitadas dos estabelecimentos de ensino e as redes de área local não estruturadas e ineficientes.

Paralelamente, as crescentes necessidades ao nível de infra-estruturas e o aumento acentuado do número de equipamentos nas escolas vieram enfatizar a necessidade de intervenção ao nível do apoio técnico aos estabelecimentos de ensino para as operações de gestão e manutenção das infra-estruturas TIC e da segurança dos equipamentos e das escolas, dos alunos, docentes e não docentes.

Neste contexto, definiu-se como prioritária a implementação dos seguintes projectos:

Eixo de Actuação	Tecnologia	
Áreas de Intervenção	Computadores Equipamento de apoio Conectividade Redes de área local	Cartões de aluno Sistemas de segurança electrónicos Apoio Técnico às escolas

Projectos	Projectos – Chave
	Kit Tecnológico Escola Aumento do número de computadores e de equipamento de apoio Aumento da disponibilidade e dos equipamentos para utilização fora dos períodos lectivos.
	Internet de Banda Larga de Alta Velocidade Revisão do modelo de conectividade
	Internet nas Salas de Aula Reestruturação das redes de área local
	Cartão Electrónico do Aluno Generalização da utilização de plataformas de cartão de aluno
	Videovigilância Generalização da utilização de sistemas de segurança electrónicos.
	Outros Projectos
	Centro de Apoio Tecnológico às Escolas

Kit Tecnológico Escola

O número limitado de computadores e de outros equipamentos de apoio, assim como a sua antiguidade, constituíam barreiras à utilização de tecnologia no processo de ensino e impediam o cumprimento das ambiciosas metas comunitárias relativas ao número de alunos por computador.

Neste contexto, foi crucial aumentar o número de computadores e de equipamentos de apoio na escola e promover a sua utilização na sala de aula e na gestão escolar.

O Kit Tecnológico Escola tinha como objetivo promover a utilização de tecnologia no processo de ensino, dotando todas as escolas de um número adequado de computadores, de impressoras, de videoprojectores e de quadros interactivos, de modo a atingir o rácio de dois alunos por computador, a assegurar que nenhuma escola apresenta um rácio de alunos por computador superior a cinco, a assegurar um videoprojector em todas as salas de aula, um quadro interactivo em cada três salas de aula e garantir que a proporção de equipamentos com antiguidade superior a três anos não ultrapassasse os 20%.

Para a concretização deste objectivo, foram estipuladas as seguintes medidas:

- Fornecer às escolas com 2.º e 3.º ciclos do ensino básico ou com ensino secundário 310.000 computadores, 9.000 quadros interactivos por ano e 25.000 videoprojectores até ao final de 2010.
- Celebrar contratos de manutenção e de renovação de equipamento com as entidades fornecedoras de equipamentos às instituições de ensino.

Internet em Banda Larga de Alta Velocidade

As limitações ao nível da conectividade constituíam entraves significativos à maior utilização de tecnologia no ensino. Grande parte das escolas registava velocidades de acesso limitadas e cerca de 20.000 computadores não se encontravam ligados à Internet.

Dois terços das escolas contratavam mais de um acesso à Internet de banda larga, representando potencial duplicação de custos na ordem dos 20%. Como tal, foi fundamental rever o modelo de conectividade das escolas, assegurando que todos os computadores nas escolas tinham ligação à Internet de banda larga com velocidade adequada, maximizando a eficiência dos investimentos, de forma a aumentar progressivamente a velocidade mínima de acesso das escolas à Internet de 4 Mbps para 48 Mbps, assegurar que todos os computadores das escolas estivessem ligados à Internet de banda larga e que o rácio de alunos com ligação à Internet de banda larga fosse de 2:1 no final de 2010.

Internet nas Salas de Aula. Redes de Área Local

A infra-estrutura de redes de área local, até à implementação do PTE, não satisfazia as necessidades de conectividade e de mobilidade dos agentes e constituía um entrave à maior utilização de tecnologia. Existiam redes de área local em quase todas as escolas. No entanto, num terço das escolas, as redes de área local não eram estruturadas e, na maior parte dos estabelecimentos, encontravam-se confinadas a áreas pré-definidas e limitadas, o que restringia a sua utilização. As operações de manutenção e de gestão das redes de área local das escolas estavam, na maior parte dos casos, a cargo de professores e de auxiliares administrativos, o que afastava os agentes das suas funções pedagógicas e de gestão e originava ineficiências na administração e na operação das redes. Neste contexto, pretendia-se reestruturar as redes de área local das escolas, dotando as redes em causa das funcionalidades necessárias para uma utilização ubíqua e segura de Internet e de intranets. Para a implementação deste projecto pretendeu-se instalar redes de área local, com acesso remoto e separação segura de redes em todas as escolas, conjuntamente com pontos de acessos cablados e sem fios nas salas de aula, nas principais áreas de estudo e de convívio da escola.

Cartão Electrónico do Aluno

A implementação de plataformas de cartões electrónicos de aluno aumenta a segurança e representa ganhos de eficiência importantes para as escolas. Permite a utilização da tecnologia por docentes, não docentes e encarregados de educação permitindo, entre outros aspectos, suprimir a circulação de numerário, controlar as entradas e as saídas dos alunos e consultar o processo administrativo, o percurso académico e os consumos dos alunos.

O processo de implementação das plataformas em causa regista ineficiências: as escolas tinham autonomia para decidir sobre a implementação de sistemas de cartões electrónicos de aluno, cabendo à escola escolher o seu fornecedor. Por essa razão, encontravam-se implementadas plataformas de mais de 10 fornecedores distintos, o que origina dificuldades de troca de informação de forma electrónica entre as escolas e torna o investimento ineficiente. Pelo acréscimo de segurança e de eficiência que proporcionam, foi crucial generalizar a todas as escolas a utilização destas plataformas e de funcionalidades-chave como o controlo de acessos e a substituição de numerário. Era igualmente fundamental assegurar que as plataformas utilizadas fossem compatíveis entre si, de maneira a permitir acompanhar o registo do aluno ao longo do seu ciclo de vida na escola, bem como a potenciar o desenvolvimento e a utilização de funcionalidades adicionais.

O projecto pretendia promover a utilização de tecnologia, dotando todas as escolas de plataformas de cartão de aluno até ao 2.º trimestre de 2008.

Videovigilância

Inicialmente, mais de metade das escolas não dispunha de sistemas de alarme contra intrusão. Nas escolas onde se encontravam implementados sistemas de alarme (49%) ou de videovigilância (13%), observavam-se várias ineficiências, designadamente dificuldades de monitorização, formatos de intervenção desadequados, tempos de intervenção demorados e formatos de operação e gestão ineficientes.

Desta forma, este projecto tinha como objectivo, até ao terceiro trimestre de 2008, implementar sistemas de alarme e de videovigilância com monitorização local e remota, permitindo disponibilizar equipamentos e serviços, de forma a aumentar a segurança de pessoas e de bens, assegurando a implementação de um modelo de monitorização e de intervenção eficiente que salvaguardasse o investimento e integridade dos equipamentos.

Medida: Programa E-escola e E-escolinha		
E-escola	Até 2010 390 milhões de euros foram fornecidos pelas empresas de comunicações pela obtenção de licenças UMTS (comunicações móveis da 3ª geração). A partir de 2010 é financiado pelo orçamento de estado. <i>Fonte: http://www.cmjornal.xl.pt/detalhe/noticias/nacional/ensino/faltou-distribuir-400-mil-portateis, consultado em 10/07/2012</i>	Computadores Entregues 200.000 (até Julho/08) 285.000 (até Nov/08) 960.000 (2009) 1.220.000 Beneficiários Inscritos
E-escolinha (nº computadores entregues)		370.000
Medida: Kit tecnológico		
Computadores (Nº Computadores Entregues)	Verba máxima atribuída € 279.000.000 + IVA <i>Fonte: Resolução do Conselho de Ministros n.º 133/2007</i>	228.361 (2009) 310.00 (2010)
Medida: Kit tecnológico (Continuação)		
Projectores (Nº Projectores Entregues)	Verba máxima atribuída € 25.100.000 + IVA <i>Fonte: Resolução do Conselho de Ministros n.º 136/2007</i>	28.711
Quadros Interactivos (Nº Quadros Interactivos Entregues)	Verba máxima atribuída € 9.000.000 + IVA <i>Fonte: Resolução do Conselho de Ministros n.º 132/2007</i>	7.613 (2009) 9.000 (2010)
Medida: Internet em Banda Larga de Alta Velocidade		
% das escolas públicas ligadas em banda larga	Verba máxima atribuída € 14.500.000 + IVA <i>Fonte: Resolução do Conselho de Ministros n.º 23/2008 e Portaria n.º 204/2008</i>	100 %
Medida: Internet na sala de aula e redes de área local		
% das escolas públicas com internet na sala de aula e redes de área local	Verba máxima atribuída € 75.000.000 + IVA <i>Fonte: Resolução do Conselho de Ministros n.º 134/2007</i>	35,11% das escolas com 2.º e 3.º ciclos do ensino básico e com ensino secundário com redes de área local com e sem fios
Medida: Cartão da Escola		
Ponto de situação do projecto	Verba máxima atribuída € 18.000.000 + IVA <i>Fonte: Resolução do Conselho de Ministros n.º 124/2008</i>	Contrato visado pelo Tribunal de Contas em Outubro de 2009 Novo cartão nas escolas em 2010
Medida: Escol@ segura e Videovigilância		
Ponto de situação do projecto	Verba máxima atribuída € 30.000.000 + IVA <i>Fonte: Resolução do Conselho de Ministros n.º 135/2007 e Portaria n.º 300/2009</i>	51% das cerca de 1 200 escolas com 2.º e 3.º ciclos do ensino básico e com ensino secundário com sistemas de videovigilância instalados (561 escolas)
Verba máxima prevista para o eixo tecnologia até 2010 (valor aproximado)	€ 840.600.000 + IVA	

Quadro resumo do eixo tecnologia

2. Eixo Conteúdos

Os conteúdos e as aplicações são essenciais para a alteração das práticas pedagógicas, ao favorecer o recurso a métodos de ensino mais interactivos e construtivistas, contribuindo para criar uma cultura de aprendizagem ao longo da vida. Têm um forte papel catalisador da alteração das práticas de gestão e da adopção de novos métodos e ferramentas de trabalho conducentes a melhores níveis de eficiência nas escolas.

- As medidas do Plano Tecnológico da Educação implementadas no eixo «Conteúdos» pretendiam contrariar os principais inibidores de modernização identificados, nomeadamente
- A baixa utilização de conteúdos e de aplicações motivada, em parte, pela escassez de conteúdos informáticos de qualidade em língua portuguesa e adaptados à utilização pedagógica em ambiente de aula.
- O número limitado de funcionalidades disponibilizadas pelas plataformas colaborativas com maior penetração em Portugal (p. ex., Moodle) e o subaproveitamento das suas potencialidades (pois são utilizadas fundamentalmente como canal de comunicação e de partilha de documentos).
- A sua utilização não se encontra generalizada a toda a comunidade de ensino.
- A reduzida utilização de TIC na gestão operacional nas escolas e o conhecimento limitado das soluções e aplicações disponíveis, observando-se elevadas assimetrias entre escolas.
- A partilha de conhecimento e de melhores práticas de gestão está circunscrita ao círculo de relações informais dos membros dos conselhos executivos e dos auxiliares administrativos.
- Utilização muito reduzida de meios de comunicação electrónicos — correio eletrónico, fóruns, chats, etc. — resultando no uso intensivo do suporte papel na maioria dos processos e das comunicações entre agentes da comunidade de ensino.

Neste contexto, os seguintes projectos foram considerados prioritários:

Eixo de Actuação	Conteúdos	
Áreas de Intervenção	Produção e partilha de conteúdos	Informatização de manuais escolares
	Produção e partilha de aplicações de gestão	Software opensource
	Certificação de conteúdos e aplicações	
Projectos	Projectos – Chave	
	Mais-Escola.pt Portal da escola com funcionalidades de partilha de conteúdos, ensino à distância e colaboração.	
	Escola Simplex Plataforma electrónica de apoio à gestão	
	Outros Projectos	
	Manuais escolares electrónicos Disponibilização de manuais escolares e de conteúdos em formato electrónico Incentivo à utilização de manuais e conteúdos informáticos	
	Plataforma de comunicação electrónica integrada Generalização da utilização de ferramentas de comunicação em suporte informático entre os agentes da comunidade educativa	

Mais-Escola.pt

A necessidade de intervenção no eixo «Conteúdos» era ainda mais premente, uma vez que o caminho para a sociedade do conhecimento implica não apenas a massificação de equipamentos informáticos e de conectividade, mas também a alteração dos métodos tradicionais de ensino e de aprendizagem, para a qual era preeminente a existência de ferramentas, de materiais pedagógicos e conteúdos adequados.

A nível internacional, no âmbito dos respectivos processos de modernização tecnológica do ensino, a maior parte dos países desenvolveu internamente plataformas de e-learning. Na maioria dos casos, a promoção do desenvolvimento e da utilização destas plataformas figurava nas prioridades das medidas de política. O desenvolvimento de plataformas próprias de e-learning era motivado pela necessidade de assegurar a disponibilização das funcionalidades desejadas, de garantir fiabilidade e facilidade de utilização, não olvidando a importância de criação de efeito de rede e as preocupações de segurança, de interoperabilidade e de transferibilidade dos conteúdos entre diferentes aplicações.

De acordo com os resultados do estudo de diagnóstico, a utilização das plataformas em questão revelava as mesmas

PLANO TECNOLÓGICO DA EDUCAÇÃO: SEIS ANOS DEPOIS

limitações identificadas em países de referência, nomeadamente:

- Ao ser fundamentalmente utilizada por professores e entre professores, a sua utilização não se encontra generalizada junto dos restantes membros da comunidade educativa.
- É utilizada apenas como um novo canal para as práticas de ensino tradicionais: um canal de interacção e de comunicação entre agentes e de distribuição de material de aula.
- Não estão a ser exploradas todas as potencialidades disponíveis em plataformas virtuais de conhecimento:
- Subaproveitamento das plataformas como catalisador para a alteração das práticas pedagógicas.
- As plataformas ainda não estão a ser usadas para a informatização da gestão administrativa dos estabelecimentos.

As plataformas de e-learning desempenham um papel fulcral nos processos de modernização tecnológica do ensino, por serem catalisadoras da produção e utilização de ferramentas, conteúdos e informações em suporte electrónico como complemento ou mesmo substituto aos métodos tradicionais de ensino em sala de aula. Permitem a mudança das práticas pedagógicas, promovendo práticas de ensino mais interactivas, construtivistas, assim como a criação da cultura de aprendizagem ao longo da vida minimizando a info-exclusão, permitindo o acesso remoto e de baixo custo a conteúdos, módulos e cursos. Reduzem a exclusão do ensino e da aprendizagem de cidadãos com necessidades especiais, sendo catalisadoras de abordagens colaborativas ao ensino e à gestão e permitindo a alteração das práticas de gestão.

Este projeto tinha como objetivo promover a produção, a distribuição e a utilização de conteúdos informáticos nos métodos de ensino e aprendizagem (p. ex., exercícios, manuais escolares, sebenta electrónica, etc.); encorajar o desenvolvimento do portefólio digital de alunos; complementar o ensino tradicional e promover novas práticas de ensino, minimizar a info-exclusão, disponibilizar conteúdos e ferramentas que tornassem viável o ensino a distância e o desenvolvimento da articulação entre a escola e o mercado de trabalho (p. ex., integrando funcionalidades como bolsas de emprego).

Para se concretizar, o projecto tinha como medidas o lançamento progressivo durante o ano lectivo de 2008-2009 da plataforma Mais-Escola.pt, plataforma virtual de conhecimento.

Escola Simplex

Um elevado número de estabelecimentos de ensino já recorria a aplicações informáticas para apoiar a gestão escolar. No entanto, o leque de serviços informatizados era ainda muito reduzido e a maior parte dos processos recorria ao suporte papel, gerando elevadas ineficiências na troca de informação entre aplicações e estabelecimentos de ensino, e entre estes e os serviços do Ministério da Educação.

O facto de existirem mais de 10 fornecedores de sistemas nas escolas — sistemas que, na maioria dos casos, não eram compatíveis entre si, gerava ineficiências, dada a redução do efeito de escala e da flexibilidade de fragmentação e de escalabilidade dos investimentos. A partilha de boas práticas de utilização intensiva de TIC na gestão operacional das escolas era maioritariamente efetuada com base em redes de contactos informais de docentes e de auxiliares administrativos, não existindo mecanismos que potenciassem a sua disseminação e aclarassem a adopção de aplicações ou processos eficazes.

Foi primordial, neste contexto, desenvolver uma plataforma de apoio à gestão administrativa escolar, incorporando a informatização de processos críticos (p. ex., matrículas, faltas, actas, etc.) e ferramentas colaborativas, por forma a dinamizar a produção de conteúdos e de aplicações de gestão e a potenciar a modernização tecnológica e a massificação de boas práticas.

Para a concretização foram estabelecidas as seguintes medidas:

- Lançamento durante o ano lectivo de 2008-2009 da plataforma Escola Simplex, plataforma virtual que permita, entre outros:
 - Informatização de processos críticos das escolas.
 - Comunicação entre agentes da comunidade de ensino e agentes externos.
 - Partilha de melhores práticas de gestão e de ferramentas de suporte à gestão.
- Definição e implementação de um modelo de gestão e

Medida: Mais-Escola.pt		
Portal das escolas	Verba máxima atribuída	www.portaldasescolas.pt
Medida: Escola Simplex	€ 30.000.000+ IVA	
Escola Simplex	Fonte: Resolução do Conselho de Ministros n.º 35/2009 e Portaria n.º 287/2009, de 20de Março	

de operação para a Escola Simplex que assegure a sua dinamização e utilização auto-sustentada.

Quadro resumo do eixo conteúdos

No Code

PLANO TECNOLÓGICO DA EDUCAÇÃO: SEIS ANOS DEPOIS

3. Eixo Formação

O défice de competências em TIC era apontado como uma das principais barreiras à utilização da tecnologia nas escolas em Portugal. Em qualquer esforço de modernização tecnológica, o investimento em equipamentos, conteúdos e aplicações e o investimento em formação e certificação de competências TIC são mutuamente complementares.

A utilização das TIC, nomeadamente da WEB, contém um potencial de mudança do paradigma pedagógico na formação, apontando para um incremento da qualidade da aprendizagem. A aprendizagem deve ser constituída por acções humanas e tecnológicas na múltipla rede de construção de saberes e fazeres, de tal modo que a aprendizagem em aula e a aprendizagem à distância se combinem e complementem, extraindo de cada modelo as mais-valias que proporcionam uma melhor aprendizagem. Este processo privilegia a formação de competências à volta das quatro aprendizagens fundamentais exigidas num tempo marcado pela Sociedade da Informação: “aprender a conhecer”, “aprender a fazer”, “aprender a conviver” e “aprender a ser” (Unesco; 1996).

Com as medidas a implementar no eixo «Formação», pretendia-se dar uma resposta clara aos principais fatores inibidores da modernização em matéria de competências, nomeadamente a reduzida utilização das TIC nos métodos de ensino e aprendizagem; a formação de docentes pouco centrada na utilização pedagógica das TIC e a inexistência

de mecanismos de certificação e valorização profissional de competências

Neste contexto, definiu-se como prioritária a implementação dos seguintes projectos:

Formação e Certificação de Competências TIC - Formação Pro

Em Portugal, foram já dados alguns passos no sentido de melhor adequar a formação em TIC às reais necessidades dos agentes. Em 2006 foram definidos referenciais de formação para diferentes perfis de professores, numa tentativa de melhor adequar o investimento em formação às necessidades dos docentes e de criar as bases necessárias para a criação de planos de formação contínua para cada tipologia de professores.

Este projecto pretendia promover uma eficiente formação em TIC dos agentes da comunidade educativa, promover a utilização das TIC nos processos de ensino e aprendizagem e na gestão administrativa da escola e contribuir para a valorização profissional das competências TIC, sendo que estabelecia, como objectivo inicial, ter 40% dos docentes certificados em 2009 e 90% dos docentes certificados em 2010.

No intuito de garantir a concretização de tais pretensões, foram criados cursos de formação modulares e progressivos em TIC para docentes e não docentes, incorporando uma forte vertente de utilização das TIC em aula e na gestão administrativa das escolas e instituído um programa de certificação obrigatória dos agentes da comunidade de ensino na área das TIC.

Avaliação Electrónica

A utilização de meios informáticos na avaliação escolar tem um forte papel indutor da modernização, uma vez que estimula a procura de tecnologia por todos os agentes da comunidade educativa: alunos, docentes e Encarregados de Educação. Adicionalmente, a adopção de sistemas informáticos na avaliação tem efeitos positivos na harmonização de critérios de avaliação e na monitorização e uniformização de ritmos de aprendizagem, uma vez que simplifica a aplicação generalizada de provas e critérios de avaliação objectivos e permite o fácil acompanhamento estatístico de resultados.

Este projecto tinha como objectivo promover a utilização pedagógica das TIC, reforçar a segurança e imparcialidade nos momentos de avaliação e uniformizar critérios de avaliação e ritmos de aprendizagem. Para a concretização deste projecto, foram projectadas provas de avaliação nacionais em suporte informático, sendo que o projecto, segundo as previsões iniciais, teria início com provas de diagnóstico e de aferição no decorrer do ano lectivo 2007-2008.

Eixo de Actuação	Formação
Áreas de Intervenção	Formação de docentes e não docentes Certificação de competências de docentes e alunos Utilização de TIC nos processos de ensino-aprendizagem
Projectos	Projectos – Chave
	Formação e certificação de competências TIC – Formação Pro Reestruturação do modelo de formação de docentes em TIC Instituição de mecanismos de certificação de competências TIC
	Avaliação Electrónica Utilização de meios informáticos na avaliação escolar
	Outros Projectos
	Integração das TIC nos métodos de ensino e aprendizagem Literacia em Aplicações Open Source

O quadro apresentado abaixo enuncia o ponto de situação

Medida: Formação e certificação de competências TIC – Formação Pro		
Custo total de implementação de competências TIC	€14.541.152,57 <i>Nota: Alguns custos poderão ser equacionados no âmbito dos restantes eixos do Plano Tecnológico da Educação</i> <i>Fonte: Costa, F. A., Rodrigues, Â., Peralta, M. H., Cruz, E., Reis, O., Ramos, J. L., et al. (2008). Competências TIC - Estudo de Implementação Vol.1. Lisboa: Editorial do Ministério da Educação.</i> <i>URL: http://www.gepe.min-edu.pt</i>	Em fase de implementação.

de cada medida do plano ordenado por eixo.

Quadro resumo do eixo formação

Seis anos depois

Ao longo de seis anos, a implementação do Plano Tecnológico foi efectuada em contextos conjunturais muito diversos. O Plano deveria ter a flexibilidade suficiente para se enquadrar nos desafios de conjuntura, mas não deveria afastar-se das linhas de força para a mudança estrutural que o justificam e lhe dão sentido e oportunidade.

O progresso feito por Portugal neste domínio é considerado uma boa prática global. Mais de um milhão de computadores foram distribuídos no âmbito dos programas e.Escolas e e.Escolinhas a estudantes, docentes e trabalhadores em formação profissional. Um computador com acesso à Internet por cada 5,6 alunos em cada escola do Ensino Público, um quadro interactivo por cada três salas de aula e um videoprojector por cada sala foram algumas das metas alcançadas pelo Plano Tecnológico da Educação (PTE) até 2010.

O relatório de balanço, apresentado em Setembro de 2009, revelava ainda que todas as escolas públicas estavam ligadas à Internet através de fibra óptica a pelo menos 64 Mbps, existindo 112 instituições que navegavam a 112 Mbps.

No total, já tinham sido instalados 228 361 computadores, quando estavam previstos 310 mil até ao final de 2010. Entre os quadros interactivos faltavam instalar 1387 equipamentos para atingir o valor de 9000 apresentado como meta. O número de videoprojectores colocados já ultrapassava os 25 mil inicialmente previstos, somando 28 711, um por cada sala de aula.

Este era um ambicioso Plano Tecnológico da Educação, que pretendia fazer das nossas escolas ambientes de aprendizagem de referência na modernidade e na inovação. Portugal tem hoje uma das maiores taxas de penetração de

banda larga móvel do mundo e é um dos Países com mais computadores portáteis por mil habitantes.

A natureza da mudança que a integração das TIC promove na organização pedagógica afecta não só o professor, mas também o aluno. A ambos passa a ser solicitado que interajam com diferentes meios e sujeitos, a compartilhem o conhecimento, a construam novas relações, fazendo e desfazendo a informação, reconstruindo-a em novos espaços, em diferenciados significados e novas forma de organização.

Vale a pena enfatizar que a acção das TIC pode operar uma mudança na cultura educativa; uma cultura de aprendizagem colaborativa, seja do ponto de vistas das interacções, seja das representações. É, no entanto, necessário continuar a melhorar os sistemas de formação de docentes para aumentar a sua eficácia e assegurar que a utilização das TIC não está confinada a disciplinas específicas e que passe a fazer parte do quotidiano da escola e do método de aprendizagem de todas as disciplinas.

O Plano Tecnológico poderia contribuir para uma escola onde se aprende mais e melhor, onde os professores utilizam estratégias inovadoras e os alunos trabalham com mais afinco. Tudo isto são pontos fortes que podem tornar a sociedade portuguesa muito mais competitiva face a um mundo em globalização. Porém, o projeto “morreu na praia”, e hoje, no final do ano letivo 2011 – 2012, o país apresenta dois tipos de ambientes escolares: as escolas onde existem condições estruturais de excelência para a prática do ensino-aprendizagem e as escolas em que as condições estruturais são tão degradantes que a prática de ensino-aprendizagem se torna quase um ato de heroísmo. Mais preocupante é que estes ambientes escolares usualmente coabitam, lado a lado, evidenciando a desigualdade e exclusão social.

A crise financeira e a instabilidade que a mesma trouxe para a economia real, veio, em certa medida, afetar a operacionalização do Plano Tecnológico da Educação. É de salientar que o último relatório disponibilizado ao público em geral (www.planotecnologico.pt), data de 9 de julho de 2009. Este facto é deveras preocupante, pois traduz a falta de acompanhamento da implementação do plano tecnológico, por parte do conselho consultivo. Uma outra preocupação é precariedade e redução do pessoal especializado na área da informática no ensino básico e secundário, uma vez que a manutenção do parque escolar era garantida por estes docentes. A ausência de pessoal especializado, seja docente ou técnico, levará à deterioração e inutilização dos equipamentos, o que tornará o investimento efetuado no parque tecnológico inútil e de curto prazo. As ferramentas que tinham como objetivo promover a organização pedagógica e operar uma mudança na cultura educativa tornar-se-ão barreiras e entraves para o ensino-aprendizagem. Inquietante é a fraca aposta na formação dos docentes, os quais representam a principal barreira à

integração das TIC nos métodos de ensino e uma débil disponibilização do parque informático, por parte dos órgãos de gestão, aos alunos, uma vez que os equipamentos são disponibilizados de forma preocupantemente limitadora, dispersos, na sua maioria, pelas salas de aula e disponibilizados de forma exagerada em salas de trabalho da equipa docente.

Em julho de 2012, após um esforço coletivo da população portuguesa, não se compreende como é que a crise financeira tenha perturbado a operacionalização do plano tecnológico. Observando os três eixos de atuação, verificamos que o eixo Tecnologia está neste momento “ao sabor do vento”, sem a assistência técnica especializada necessária, prevê-se a degradação progressiva de todo o parque tecnológico e os eixos dos Conteúdos e Formação estão muito aquém das expectativas. Observando os projetos envolvidos nestes dois eixos, poderemos afirmar que, neste momento, a alguns não chegaram a ser concretizados, apesar dos investimentos, e os projetos que apresentaram alguma fase concretizada, neste momento ou estão extintos ou abandonados. Podemos comprovar isso com o Portal das Escolas (www.portaldasescolas.pt) que apresentam, em julho de 2012, sinais de abandono, com vários erros na interface gráfica. No eixo de atuação Formação, a degradação do Portal das Escolas induz problemas no Sistema de Formação e Certificação de Competências TIC, a extinção progressiva de centros de formação coloca entraves à formação e certificação de docentes e não docentes e a lecionação das disciplinas com domínio das TIC, por docentes sem habilitação própria e profissional para a lecionação da informática, para além de estimularem a não abordagem de conteúdos técnicos, incitam a má utilização das ferramentas e a iliteracia em aplicações open source.

Neste momento, o leitor questiona-se do porquê deste artigo estar numa revista especializada em programação e tecnologia informática. É muito simples, a forma de atuação dos intervenientes do processo do Plano Tecnológico não só afeta todos os contribuintes, como insulta todos os profissionais do ramo da informática. O custo-benefício das plataformas e recursos envolvidos são absurdos, a falta de suporte da alta administração, a meio do projeto, levou ao desgoverno de todo o processo e à sua falta de eficácia.

Espero que este artigo permita ao leitor-programador refletir, em primeiro lugar, sobre os custos e benefícios das aplicações e processos informáticos, em segundo lugar, sobre a importância da qualificação humana num projeto e, em terceiro, sobre a importância dos fatores críticos de sucesso, inerentes à gestão de projetos na elaboração de qualquer tarefa.

Bibliografia

(2005). Plano Tecnológico: Uma estratégia de crescimento com base no Conhecimento, Tecnologia e Inovação. Documento de apresentação.

2006, C. C.-2. (2006). Relatório de Progresso do Plano Tecnológico.

2007, C. C.-0. (2007). Relatório de Progresso do Plano Tecnológico.

2007, C. C.-2. (2007). Relatório de Progresso do Plano Tecnológico.

AICEP Portugal Global. (01 de 10 de 2009). Plano Tecnológico - Quatro anos depois. p. 6.

Conselho Consultivo do Plano Tecnológico. (2009). 8.^a Reunião do Conselho Consultivo - Nota de Imprensa Julho de 2009.

Costa, F. A., Rodrigues, Â., Peralta, M. H., Cruz, E., Reis, O., Ramos, J. L., et al. (2008). Competências TIC - Estudo de Implementação Vol.1. Lisboa: Editorial do Ministério da Educação.

Coutinho, M. C., & Chaves, J. H. (2001). Desafios à Investigação em TIC na Educação. Universidade do Minho.

Fernando, H. P., & Costa, A. (2007). TIC e Inovação Curricular. Sísifo - Revista de Ciências da Educação , 142.

GEPE. (2008). CATE – Centro de Apoio Tecnológico às Escolas. Estudo de Implementação. Lisboa: Gabinete de Estatística e Planeamento da Educação (GEPE).

GEPE. (2007). Estudo de Diagnóstico: a modernização tecnológica do sistema de ensino em Portugal. Lisboa : Gabinete de Estatística e Planeamento da Educação.

GEPE. (2009). Kit Tecnológico: Estudo de Implementação. Ministério da Educação, Gabinete de Estatística e Planeamento da Educação, LISBOA.

Miranda, G. L. (2007). Limites e possibilidades das TIC na educação. Faculdade de Psicologia e de Ciências da Educação da Universidade de Lisboa.

Pinto, S. M., Calixto, J. A., & Lopes, P. F. (2001). Gestores de Recursos Electrónicos: Que formação na era digital? Tese de Mestrado em Estudos de Informação e Bibliotecas Digitais.

Plano Tecnológico - Quatro Anos Depois. (2009). AICEP Portugal Global , 7.

Ramos, J. L., Espadeiro, R. G., Carvalho, J. L., do Maio, V. G., Matos, J. M., dos Santos, L. F., et al. (2009). Iniciativa Escola, Professores e Computadores Portáteis: Estudos de Avaliação. Lisboa: DGIDC- Direcção Geral de Inovação e de Desenvolvimento Curricular.

PLANO TECNOLÓGICO DA EDUCAÇÃO: SEIS ANOS DEPOIS

Resolução do Conselho de Ministros n.º 132. (2007).
PLANOTECNOLÓGICO DA EDUCAÇÃO.

Resolução do Conselho de Ministros n.º 134. (2007).
PLANOTECNOLÓGICO DA EDUCAÇÃO.

Resolução do Conselho de Ministros n.º 135. (2007).
PLANOTECNOLÓGICO DA EDUCAÇÃO.

Resolução do Conselho de Ministros n.º 136. (2007).
PLANOTECNOLÓGICO DA EDUCAÇÃO.

Resolução do Conselho de Ministros n.º 137. (2007).
PLANOTECNOLÓGICO DA EDUCAÇÃO.

Resolução do Conselho de Ministros n.º 23. (2008).
PLANOTECNOLÓGICO DA EDUCAÇÃO.

Resolução do Conselho de Ministros n.º 124. (2008).
PLANOTECNOLÓGICO DA EDUCAÇÃO.

Resolução do Conselho de Ministros n.º 35. (2009).
PLANOTECNOLÓGICO DA EDUCAÇÃO.

<http://www.cmjornal.xl.pt/detalhe/noticias/nacional/ensino/faltou-distribuir-400-mil-portateis>, consultado em 25/07/2010

Silva, B. D., & Gomes, M. J. Contributos da Internet para a mudança do paradigma pedagógico: uma experiência de trabalho colaborativo. Universidade do Minho.

(2009). TECNOLOGIAS DE INFORMAÇÃO E COMUNICAÇÃO: UM ESTUDO DE ACTUALIZAÇÃO DE TENDÊNCIAS EM CONTEXTO EDUCATIVO. Tese de Mestrado em Educação na especialidade de Comunicação Educacional Multimédia,

Universidade Aberta, DEPARTAMENTO DA EDUCAÇÃO E DO ENSINO À DISTÂNCIA.

Portaria n.º 204/2008 - Determina os encargos orçamentais decorrentes das ligações à Internet em Banda Larga sejam assumidos nos anos de 2008 a 2010

Portaria n.º 287/2009 - Determina os serviços do Plano Tecnológico da Educação

Portaria n.º 300/2009 - Determina os encargos orçamentais decorrentes da concretização da melhoria das condições de segurança nas escolas públicas

Despacho n.º 1199/2008 - Autorização de prestação de serviços no âmbito do Plano Tecnológico

Despacho n.º 145/2008 - Cria o Escalão Especial do Ensino Secundário, que define o valor de capitação até ao qual o aluno deve ser enquadrado no escalão especial de apoio ao programa de acesso aos computadores pessoais e à banda larga e no escalão especial de acesso à Bolsa de Mérito do Ensino Secundário.

Despacho n.º 143/2008 Aprova o modelo orgânico e operacional relativo à execução, no âmbito do Ministério da Educação, do Plano Tecnológico da Educação.

Decreto-Lei n.º 379/2007 - Estabelece um regime excepcional para o procedimento de contratação com vista à aquisição de serviços destinados ao desenvolvimento das experiências piloto em execução e cumprimento dos objectivos do Plano Tecnológico da Educação



AUTOR



Escrito por Ricardo Castro

Docente de CET's no Instituto Politécnico de Viana do Castelo, especializado em Gestão de Projetos e implementações de sistemas de informação para a gestão.

O EMPREENDEDORISMO COMO UMA OPÇÃO DE AUTO-EMPREGO E INCONFORMISMO

Em tempos de incerteza e de estrangulamento financeiro, o empreendedorismo não é só um discurso de circunstância, mas sobretudo uma atitude de proatividade e de resiliência (resistência e reação) face a um ambiente externo que é incerto e, muitas vezes, até adverso.

Este artigo pretende pensar o empreendedorismo e o auto-emprego como uma atitude de resiliência e de dinamismo em tempos de contração da economia, do emprego e do investimento. Neste sentido, apresentaremos algumas medidas para construir uma atitude empreendedora e fazer uma escalada bem sucedida pelo auto-emprego e (porque não?) pela criação de uma marca própria.

O empreendedorismo é acima de tudo uma atitude que assenta numa mudança de mentalidades – o tempo do emprego para a vida, depois da formação académica, praticamente terminou e o percurso profissional de cada pessoa será, mais do que nunca, multifacetado. Por isso, para os mais ou menos jovens, para os freelancers ou contratados, este é o momento de pensar a longo prazo e ter visão de futuro.

A hipótese do auto-emprego

Ao perspectivar o emprego é oportuno e legítimo colocar também a hipótese do auto-emprego e da criação de um novo negócio, sobretudo porque existem áreas profissionais congestionadas no seu formato tradicional de emprego e que oferecem poucas perspectivas de carreira, estabilidade e sustentabilidade, sobretudo em relação à entrada de novos profissionais, como acontece por exemplo, na área do jornalismo, marketing, publicidade e webdesign, assim como acontece também na psicologia, no direito e na arquitectura.

Neste cenário, o auto-emprego e a criação de uma marca própria e, portanto, de uma empresa que presta um determinado serviço ou que vende um determinado produto é encarada como uma alternativa de proatividade e de resiliência face aos bloqueios e constrangimentos que se colocam quer aos jovens recém-formados, quer àquelas pessoas que apesar de uma considerável experiência profissional se vêm também afastadas do mercado de trabalho.

Há, no entanto, algumas questões a ponderar e a ter em conta antes de uma aventura como esta: a questão das capacidades de cada empreendedor, o mercado em que vai atuar, o acesso a financiamento e a formação da equipa de trabalho, ou seja, todas as questões relacionadas com a viabilidade do plano de negócio e o futura da empresa.

As capacidades técnicas e os talentos pessoais

Além das capacidades técnicas (que serão o núcleo de todo o trabalho / empresa), também importa “ponderar” os talentos ou as habilidades pessoais, porque são as nossas capacidades naturais que nos fazem trabalhar com paixão e motivação numa determinada área profissional, seja um emprego, um hobby ou um trabalho voluntário. Os talentos são desenvolvidos e aperfeiçoados com o estudo e com o trabalho – quanto mais se exercitam, mais aperfeiçoados serão (como é óbvio), mas este trabalho de investimento pessoal permitirá ainda estimular a nossa criatividade e atualizar conhecimentos. Portanto, investir na formação pessoal e profissional permitirá sempre vislumbrar novas perspectivas, novos horizontes de ação e novas oportunidades de trabalho. A atitude de verdadeiro empreendedor será sempre aquela que estará orientada para saber sempre mais e melhor (além do objetivo de fazer sempre mais e melhor).

Conhecer o mercado e oferecer algo inovador

Depois de analisar e aperfeiçoar as competências pessoais, é necessário conhecer todos os aspetos do negócio em que se pretende investir, sobretudo o mercado e a concorrência. Além de investidor, o empreendedor é também um gestor, por isso importa ter presente uma série de questões, nomeadamente: *Quem são os potenciais clientes? Se há espaço para mais um produto (ou serviço)? Qual o fator diferenciador da solução que apresenta face ao que já existe no mercado? Quem são os concorrentes diretos? A que necessidade responde a solução proposta? Qual a capacidade inicial de resposta à procura? E de que forma está preparado para um aumento da procura e, portanto, do volume de negócios?*

Neste sentido, opte por criar um produto ou serviço realmente inovador – diferente do que já existe, ou assente em novos valores e conceitos. Quanto mais apelativa, original e atrativa for a nova marca mais facilmente irá captar e reter a atenção do público. O essencial é oferecer algo único e inovador, tornando a oferta diferente de tudo o que já existe no mercado.

Num segundo nível, torne clara a sua proposta de valor, isto é, apresente de forma verdadeira e simples *o que faz, como e de que forma a solução que apresenta torna a vida das pessoas mais fácil.*

Em termos do marketing, escolha um nome e uma imagem

O EMPREENDEDORISMO COMO UMA OPÇÃO DE AUTO-EMPREGO E INCONFORMISMO

originais que identifiquem a sua marca de forma imediata e sugestiva – toda a parte da imagem será naturalmente pensada depois de estar definido o plano de negócio e o plano operacional da empresa – sobretudo, procure transmitir verdade, valor e confiança ao seu público-alvo.

A imagem é sempre uma questão inerente a qualquer marca, empresa ou trabalho que se faça, ou seja, a imagem será sempre criada (de uma forma que pode ser vantajosa para a empresa, ou nem tanto). Por isso, trabalhe essa imagem para que seja exatamente aquela que está de acordo com os seus objetivos e para que seja precisa (sem ambiguidades), coerente com o trabalho que é feito e positiva/ promocional.

Ao nível da equipa, trata-se de orientar o trabalho no sentido de uma relação acessível e prestável para com os clientes, de forma a construir um ambiente de empatia, confiança, seriedade e de eficiência em torno da marca. Esta atitude irá permitir, em primeiro lugar, corresponder às expectativas da procura e, em segundo lugar, é a postura ideal para fidelizar os clientes.

Por outro lado, importa investir esforços numa outra dimensão da empresa: a divulgação e comunicação da sua missão e visão. Aliado ao profissionalismo e à seriedade do trabalho que é feito, importa que sejam dadas a conhecer (sem ruído e sem ambiguidades) a missão (o que é oferecido pela marca e de que forma) e a visão da empresa (isto é, os seus objetivos e o seu caráter inovador/ visionário) – este aspeto está relacionado com a questão da construção da imagem que a empresa quer passar.

Opções de financiamento

Esta é normalmente a parte mais crítica de um plano de negócios – os recursos financeiros e a projeção dos ganhos. Esta parte do plano de negócios implica algumas questões: *quanto é que preciso para dar início ao negócio? Como tenciono obter lucro? Ao fim de quanto tempo obtenho o retorno do investimento inicial? E que montante preciso em caixa para garantir que o negócio não pára?*

Conseguir crédito nas instituições bancárias é cada vez mais difícil. Mas se a ideia de negócio revelar potencial de crescimento e estiver bem planeada haverá sempre outras possibilidades que o empreendedor pode explorar para conseguir reunir o investimento necessário a uma *startup*.

Terá apenas de ponderar entre algumas diferentes formas de financiamento e escolher de entre as opções disponíveis aquela que melhor se adequa aos desafios do seu plano de negócios e aquelas que estão mais acessíveis a cada empreendedor. Para além da banca, existem os *business angels* que são investidores de capital de risco (*venture capital*) e investem em startups numa fase de arranque do negócio. A participação dos *business angels* é temporária e

não se limita à introdução de capital, apoiando também os empreendedores no crescimento e desenvolvimento da *startup* (ao nível da internacionalização, recursos humanos, parcerias, ou clientes, por exemplo) – o que é uma clara vantagem sobre as outras formas de financiamento.

Outra alternativa é o crowdfunding, um tipo de financiamento coletivo que parte da apresentação do projeto numa plataforma online – em Portugal existe o PPL Crowdfunding – a partir da qual o empreendedor procura envolver a multidão (o público) e conseguir o financiamento de que precisa para o seu projeto. Com todas as possibilidades que as redes sociais oferecem, esta forma de conseguir financiamento pode resultar muito bem (e em alguns casos é superado o montante pretendido). Existem plataformas de crowdfunding genéricas e outras mais específicas (umas direcionadas para projetos sociais, outras de música, cinema, etc). Neste caso o investidor pode ser qualquer pessoa que acredite no projeto, com um valor mínimo ou mais significativo, e existe a garantia de que, depois de reunido o montante necessário, o empreendedor terá de criar a empresa e portanto o dinheiro nunca cai no vazio. Depois desta fase os investidores, em alguns casos, poderão obter garantias financeiras (em função do valor que investiram) ou uma contribuição em termos de acesso ao produto ou serviço – este aspeto dependerá do tipo de projeto que está em causa (que poderá ser a criação de uma marca comercial, de uma exposição, de um CD, ou de uma associação, por exemplo).

E ainda, podemos optar pela tradicional e mais antiga forma de financiamento, o bootstrapping – o conceito é novo, mas a ideia já todos conhecemos e consiste em reunir recursos financeiros com a ajuda dos familiares e amigos, além das poupanças de cada empreendedor, e pensar a startup como uma escalada de conquistas faseadas (e por vezes lentas, o que é natural neste formato de investimento). Por exemplo, um profissional poderá começar a trabalhar sozinho como freelancer em nome individual e gradualmente evoluir no sentido da criação de uma sociedade e com a contratação de novos profissionais para poder também assim alargar e inovar a oferta e o seu alcance em termos de mercado.

Nas universidades e centros de incubação têm existido programas de apoio ao empreendedorismo que são relevantes e úteis porque integram a componente de formação do empreendedor (importante porque existem sempre aspetos que negligenciamos), apoiam o desenvolvimento do plano de negócios e, em muitos casos, orientam no sentido da angariação do financiamento que se precisa. Portanto, a este nível trata-se de estar atento e procurar por oportunidades como estas (que têm sido relativamente frequentes e gratuitas).

No momento de pedir ou procurar financiamento, além da apresentação das garantias de retorno do negócio e da sua sustentabilidade, é importante assegurar a disponibilidade de

No Code

O EMPREENDEDORISMO COMO UMA OPÇÃO DE AUTO-EMPREGO E INCONFORMISMO

todo o capital necessário ao clique de vida de uma startup. Ou seja, será preciso dinheiro não só para montar toda a estrutura inicial do negócio – adquirir as matérias-primas e/ou todos os materiais necessários a toda a logística da empresa - mas também será necessário garantir que existe financiamento para assegurar o crescimento, a renovação e a expansão do negócio. O ideal, portanto, é o negócio ser lucrativo ao ponto de oferecer garantias para poder saldar a curto prazo as dívidas passadas e ir gerindo a contratação de novos profissionais, assim como a aquisição de novo financiamento para se expandir.

Outra política de gestão que parece adequada é garantir ou preservar alguns recursos financeiros para realizar investimentos, ou dinamizar o negócio numa perspectiva que se venha a revelar promissora. Por outro lado, reter alguns recursos financeiros poderá ser útil também para fazer face aos imprevistos e ameaças que acontecem sempre na vida das empresas.

Formar uma equipa pluridisciplinar

Apesar de ser um negócio por conta própria que podes desenvolver sozinho, parte do êxito de uma startup está associada à existência de profissionais de outras áreas porque permitem reunir outras visões do mesmo problema, por um lado, e porque permitem alargar o âmbito de trabalho e o alcance da marca – o que é crucial na fase inicial do negócio.

Tentar fazer tudo sozinho pode ser uma tentação, mas trabalhando em equipa não só o trabalho evolui mais rápido como existe uma maior probabilidade de evitar erros e lacunas. Além disso, trabalhar em equipa poderá ajudar a atravessar com maior criatividade e positivismo as dificuldades que surgirem ao longo do caminho.

Na gestão de equipas, uma nota que me parece importante, é a capacidade para confiar e delegar tarefas, ou seja, dar espaço de liberdade e de ação a cada pessoa para que ninguém se sinta atrofiado na sua criatividade e para que o trabalho evolua com dinâmica e comprometimento.

Formar uma equipa de trabalho motivada e empenhada é algo que se pode fazer com tempo e terá maiores

probabilidades de resultar se for pensada sem pressões. A este nível, podemos falar por isso da importância do networking, ou seja, de criar uma rede de pessoas e profissionais (amigos ou apenas conhecidos) com quem possamos trabalhar, criar sinergias, estabelecer parcerias ou simplesmente pessoas com quem podemos ter algum apoio técnico quando precisemos. O networking assenta sobretudo nos conceitos de troca e partilha de ideias, conhecimentos e referências, ou seja, dar e ajudar para também receber e ser ajudado.

Resiliência

Menos técnico, mas igualmente exigente, é a capacidade de resiliência dos empreendedores. O empreendedor e investidor vai encontrar muito provavelmente no seu caminho dificuldades várias: comerciais, financeiras, logísticas, pessoais e relacionadas com os parceiros e os amigos. Por isso, é essencial que tenha uma relativa dose de resistência à frustração e ao desânimo, com maior persistência ainda. Portanto, importa aqui a capacidade de adaptação rápida às mudanças e aos imprevistos, porque o mercado é dinâmico e os clientes serão cada vez mais exigentes. No mesmo sentido, importa ser rico otimismo e motivação para manter um negócio e motivar também as pessoas que trabalham connosco.

Contudo, convém não ficar cego de entusiasmo, para ser capaz de identificar com realismo as ameaças do ambiente externo (no sentido de as poder contrariar) e perceber o que deve ser melhorado a nível interno para melhorar também a oferta.

Finalmente e, acima de tudo, ser resiliente – ser capaz de reagir perante as adversidades e as limitações (técnicas, financeiras e logísticas) com criatividade e proatividade, apoiado por um plano de negócio sustentável e uma equipa sólida, por um lado, e em conhecimentos técnicos verdadeiros, por outro – condições que permitem criar uma startup num ambiente de confiança, profissionalismo e competência, logo, com maiores probabilidades de ser bem sucedida.

AUTOR



Escrito por Marisa Almeida

é licenciada em comunicação e trabalha em assessoria na mesma área, gestão de comunicação para empresas, estratégia de marketing e estudos de mercado. Membro da comunidade <http://www.maistrafego.pt>

PROJECTO EM DESTAQUE NA COMUNIDADE P@P: Notification Agenda

A Notification Agenda é um projecto criado pelo Ricardo Amaral (Nazgulled). Trata-se de uma aplicação Android, originalmente criada com o objectivo de concorrer ao concurso de programação da comunidade AndroidPT, sendo mais tarde lançada publicamente no Android Market (agora Google Play).

A aplicação transforma pequenas notas em agradáveis notificações na barra de estado com elegantes ícones e de alta qualidade. Todas as notas são facilmente acessíveis para que o utilizador nunca se volte a esquecer de todos aqueles compromissos importantes.

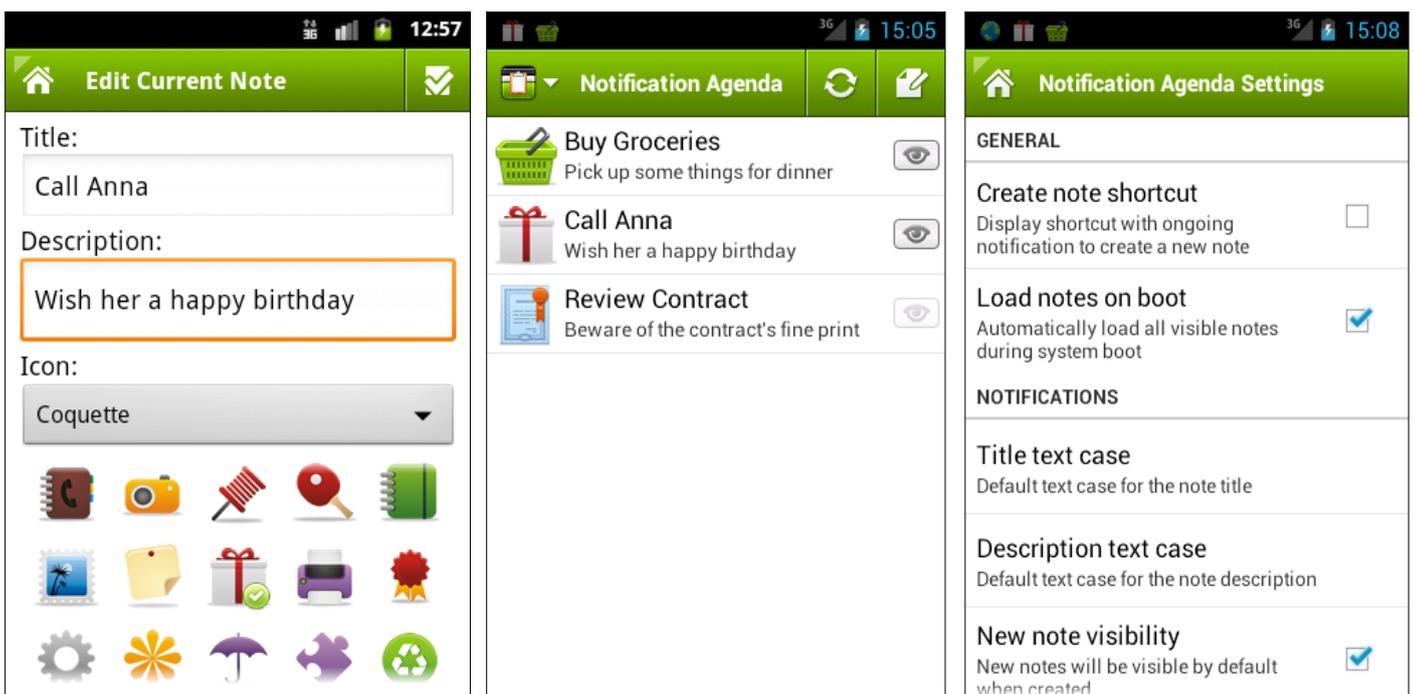
Entre as diversas características, destacam-se os ícones, com aspecto moderno e de alta resolução; possibilidade de personalização das notas que aparecem nas notificações, sendo possível escolher exactamente quais as notas que aparecem; editar as notas visíveis de forma rápida; carregar as notas no arranque do dispositivo, para que fiquem visíveis permanentemente; interface de utilizador agradável e coerente com o sistema.

A Notification Agenda foi bem recebida quer pela comunidade do Google Play, quer pelo Portugal-a-Programar. Numa questão de meses, o número de downloads registados aumentou para mais de 10.000, com bastantes críticas positivas e ideias que foram aplicadas ao longo do desenvolvimento. No Portugal-a-Programar, houve algum feedback positivo e construtivo, que motivou a correcção de pequenos erros e a implementação de novas funcionalidades. A aplicação foi ainda alvo de apreciação pela equipa Android Developers, que a analisou na sua rubrica "The Friday App Review" e, no geral, achou o projecto promissor.

A aplicação é gratuita (suportada por publicidade) e pode ser descarregada no Google Play em:

<https://play.google.com/store/apps/details?id=net.ricardoamaral.apps.notificationagenda>

ou directamente no Google Play em qualquer dispositivo Android.



Veja também as edições anteriores da Revista PROGRAMAR

354ª Edição - Junho 2012



34ª Edição - Abril 2012



33ª Edição - Fevereiro 2012



32ª Edição - Dezembro 2011



31ª Edição - Outubro 2011



30ª Edição - Agosto 2011



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

DUVIDAS?

IDEIAS?

AJUDAS?

PROJECTOS?



portugal-a-programar
•org

