

PROGRAMAR

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

EDIÇÃO #49 - JUNHO 2015

ISSN 1647-0710

ASP.NET 5

A PROGRAMAR

CRIAR ARQUIVOS OFFICE COM
OPENXML SDK

CRIAR UMA API NO
AZURE APP SERVICE

PRIMEIROS PASSOS PARA DESENVOLVIMENTO
DE APLICATIVOS ANDROID

INTRODUÇÃO ARQUITECTURA SAP
COM FOCO EM SAP-1

RAIZ ENÉZIMA ITERATIVA
PROGRAMADA

ELECTRÓNICA

ANDON
CONEXÃO DE ARDUINO
AO UNITY

COMUNIDADES

PTXUG XAMARIN: O AMBIENTE DE DESENVOLVIMENTO
EM WINDOWS & MAC

COLUNAS

ASYNC-AWAIT
A ALEGORIA DO RESTAURANTE **C#**

NO CODE

REQUISITOS E
INSTALAÇÃO DA INSIDER PREVIEW **WINDOWS 10 MOBILE**

GIRLS IN
ICT DAY 2015 **#PGGD26**

1010 ENEI

ANÁLISE

EM
JAVA **PROJETOS DE POO**
À ENGENHARIA
DE SOFTWARE **INTRODUÇÃO**

EQUIPA PROGRAMAR

Coordenador

António Pedro Cunha Santos

Editor

António Pedro Cunha Santos

Design

Sérgio Alves

Twitter: @scorpion_blood

Ilustração

Sara Freixo

Redacção

Augusto Manzano

Bruno Gomes

Bruno Sonnino

Duarte Cabral

Francisco Barbosa

Guilherme Ferreira

Nuno Santos

Nuno Silva

Paulo Frutuoso

Paulo Afonso Parreira Júnior

Paulo Morgado

Paulo Henrique Lima Oliveira

Pedro Rocha

Ricardo Peres

Rita Peres

Sara Silva

Sara Santos

Staff

António Pedro Cunha Santos

Rita Peres

Rui Gonçalves

Sara Freixo

Tiago Sousa

Contacto

revistaprogramar@portugal-a-programar.org

Website

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

ISSN

1 647-071 0

Revolvendo

Há 63 anos atrás, um brilhante engenheiro electrotécnico inglês supervisionou a montagem do primeiro circuito integrado. Estamos a falar de G. W. A. Dummer, que propôs o conceito de circuito integrado, recorrendo à fabricação de vários componentes electrónicos de um circuito, num único bloco de material semiconductor. Passados estes anos, não “vivemos” sem a tecnologia, que assenta nesse mesmo conceito! Revolucionário na altura, facilitou uma série de inovações que agora damos por garantidas, no nosso dia a dia, desde o simples relógio digital, até ao moderno computador, sem esquecer aquele “gadget” que tanta gente tem fascinado, dado pelo nome de Raspberry Pi, que não mais é do que um computador numa só placa de circuito!

Sempre me disseram que o mundo “dá voltas e mais voltas”, numa “revolução constante” e a tecnologia, dá as voltas com ainda mais rapidez, do que aquela que seria expectada!

No mês que passou a comunidade Portugal-a-Programar, comemorou o seu décimo aniversário! Foram 10 anos, muitas “revoluções”, bastante tempo passado, muito evoluído, a Revista que lêmos, é prova disso! Fruto do engenho e motivação desta comunidade, de programadores e entusiastas, que se decidiram a fundar a revista naquilo que para mim é um dos maiores “hacks of kindness” em que tive a oportunidade de participar, em Português! Por isso mesmo, felicito a comunidade pelo seu décimo aniversário e agradeço-lhe por ter fundado este projecto, que é a revista PROGRAMAR.

Nas muitas “revoluções”, do relógio, minuto após minuto, hora após hora, dia após dia, estamos cada vez mais próximos da quinquagésima edição desta nossa publicação. Não falando muito do que está para vir, pois “o segredo é a alma do negócio”, ou assim diz o ditado popular, gostava de vos dizer que nos últimos 4 anos muito esforço tem sido feito, para que esta publicação chegue até vós, o melhor possível. Muitas horas são despendidas, para que edição após edição, consigamos fazer melhor. Certamente iremos melhorar no futuro! Caminharemos para fazer melhor, para vos trazer melhor!

Parafrazeando um texto que me recordo de ter lido, de um personagem tão carismático quanto complexo “Não podemos conectar os pontos olhando para a frente! Apenas o podemos fazer, quando olhamos para trás! Assim só nos resta confiar que os pontos se irão ligar de alguma forma no futuro!”, e tal como a frase parafrazeada por mim, os pontos daquilo que é a revista, que é a tecnologia, que somos nós, certamente só os conseguiremos ligar no futuro! Até lá, confiamos que os pontos se irão ligar, de uma forma ou de outra!

Até lá, agradeço-vos a todos por lerem a revista, que esta pequena-grande equipe vos traz, a cada edição.

António Santos

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

[8](#) ASP.NET 5 - **Ricardo Peres**

A PROGRAMAR

- [18](#) Criando arquivos Office com OpenXML SDK - **Bruno Sonnino**
- [23](#) Criar uma API no Azure App Service - **Guilherme Ferreira**
- [27](#) Primeiros passos no desenvolvimento de aplicações Android - **Paulo Afonso P. Júnior, Paulo H. Lima Oliveira**
- [34](#) Introdução A arquitectura SAP com foco em SAP-1 - **Augusto Manzano**
- [37](#) Raiz enésima iterativa programada - **Augusto Manzano**

ELECTRÓNICA

- [41](#) Conexão de Arduino ao Unity - **Nuno Santos**
- [43](#) ANDON - **Paulo Frutuoso, Francisco Barbosa, Duarte Cabral e Pedro Rocha**

COLUNAS

[46](#) C# Async-Await - A alegoria do restaurante - **Paulo Morgado**

ANÁLISES

- [50](#) Projetos de POO em Java - **Bruno Gomes**
- [52](#) Introdução à Engenharia de Software - **Sara Santos**

COMUNIDADES

[54](#) **PTXug** - Xamarin: O ambiente de desenvolvimento em Windows & Mac - **Sara Silva**

NO CODE

- [59](#) Windows 10 Mobile: Requisitos e instalação da Insider Preview - **Nuno Silva**
- [63](#) #PGGD26 – Girls in ICT Day 2015 - **Rita Peres**
- [65](#) 1010 ENEI - **Rita Peres**
- [66](#) Projecto em Destaque P@p - MiniCalc

EVENTOS

IoT Hackathon 2015 Lisboa @ 16-18 Junho 2015
Bsides Lisbon 2015 - 3 de Julho 2015

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

Node.js Foundation torna-se realidade

Depois de alguns meses de discussão, a Node.js Foundation tornou-se finalmente numa realidade. Para quem não está a par, o Node.js é um projecto open-source da Joyent cuja participação externa era difícil e a evolução da plataforma não era tão rápida como alguns esperavam. Neste impasse, um grupo decidiu fazer um fork ao projecto e criar o io.js, projecto lançado no final do ano passado, suportado por versões mais recentes da V8.

No início deste mês foi lançada a segunda versão do io.js, mas neste ponto já existiam conversações para a convergência de ambos os projectos e maior abertura para participação externa no projecto. A discussão terminou há poucas horas e o criação da Node.js Foundation foi oficializada.

Podem ler mais no blogue do Node.js: <http://blog.nodejs.org>

A página principal do projecto parece estar a sofrer algumas alterações após esta decisão e por isso alguma informação pode não estar acessível enquanto o processo de alterações está a decorrer.



Investigação da Universidade de Coimbra financiada pela gigante Google

Uma equipa de investigadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC) desenvolveu um **novo modelo informático**, relevante para a **aplicação de uma nova geração de sistemas de reconstrução 3D de ambientes urbanos**.

Serviços como o “Google Street View” dão uma perspectiva 3D das ruas, mas estão limitados ao ponto de vista do veículo que capturou as imagens. Proporcionando uma experiência imersiva, em que o utilizador navega livremente pelas ruas, a **equipa de investigadores de Coimbra criou um sistema que obtém reconstruções 3D detalhadas de cidades**.

O projeto chamou a atenção da Google, que o seleci-

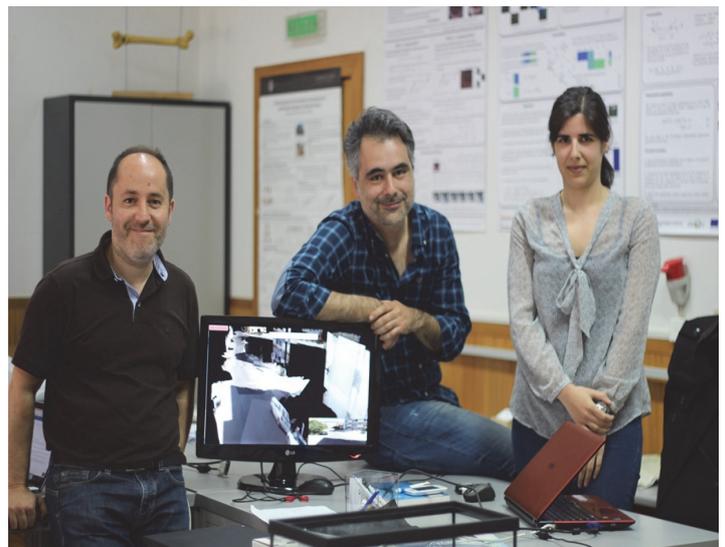
onou no âmbito de um concurso mundial de ideias muito competitivo, com taxas de aceitação na ordem dos 15 %, e posteriormente o financiou.

A pesquisa, iniciada em janeiro de 2014, reúne dois grupos de investigação do Departamento de Engenharia Electrotécnica e de Computadores (DEEC) da UC – Instituto de Sistemas de Robótica e Instituto de Telecomunicações.

A grande novidade desta tecnologia, que se encontra em fase protótipo (demonstração disponível em: <https://www.youtube.com/watch?v=OeYEwq-8TBI>), afirmam os investigadores Carolina Raposo, João Barreto e Gabriel Falcão, **«é que este algoritmo tem por base a utilização de planos, não só para descrever a cena, mas também para calcular o movimento da câmara. Isto faz com que os modelos em 3D sejam gerados automaticamente e armazenados de forma muito compacta (ao contrário dos métodos existentes que trabalham com nuvens de pontos), permitindo a sua rápida transmissão.»**

Uma outra vantagem da utilização de planos, prosseguem os investigadores, reside no facto de **«esta tecnologia ser capaz de trabalhar com um número reduzido de imagens. Isto acontece porque é frequente que o mesmo plano seja “visto” pelas câmaras em posições distantes, permitindo recuperar o movimento. Tal não se verifica com os métodos atuais que requerem que as imagens sejam adquiridas em posições fisicamente próximas e, conseqüentemente, necessitam de muito mais informação.»**

Adicionalmente, este sistema usa arquiteturas de processamento paralelo para acelerar bastante o tempo de computação, gerando automaticamente os mapas em 3D e armazenando a informação no servidor.



«Nota de imprensa Assessoria de Imprensa - Universidade de Coimbra • Reitoria »

Equipas do IPCA e do Colégio Militar venceram iDrone Cup 2015

Uma equipa do Instituto Politécnico do Cávado e do Ave (IPCA) e outra do Colégio Militar (Lisboa) foram as grandes vencedoras da segunda edição da iDrone Cup, única competição de drones em Portugal, que decorreu nos dias 2 e 3 de maio, no Pavilhão Municipal de Barcelos.

A Liga Universitária, destinada apenas a equipas do ensino superior, foi conquistada pela “MEEC-IPCA”, constituída por estudantes do curso de Mestrado em Engenharia Eletrónica e de Computadores do IPCA. No segundo lugar ficaram os “Macacos Elétricos” e em terceiro os “Flash Drone”.

Na Liga Júnior, exclusiva para equipas no ensino secundário e profissional, os grandes vencedores foram os “CM_iDrone”, que viajaram desde o Colégio Militar, em Lisboa, até Barcelos para conquistarem o primeiro lugar no pódio. No segundo posto ficaram os “Future Tech”, do Agrupamento de Escolas das Marinhas (Esposende), e em terceiro os “iFloat”, da Escola Profissional CIOR (Famalicão).

Além de medalhas e diplomas de participação para todos, as equipas vencedoras levaram para casa drones (1º classificados), tablets (2º) e smartphones (3º).

Na iDrone Cup 2015 participaram 30 equipas, num total de 120 alunos do ensino secundário, profissional e superior, que ao longo de três dias competiram entre si em cinco provas que visaram a aplicação de conhecimentos nas áreas da programação e processamento de imagem, tendo em vista o voo autónomo dos drones.

No final, houve também prémios para as equipas que venceram cada uma das provas, tendo estado presentes na cerimónia de encerramento o presidente do IPCA, João Carvalho, o vereador do Desporto da Câmara Municipal de Barcelos, José Carlos Brito, o diretor da Escola Superior de Tecnologia do IPCA, Nuno Rodrigues, e um representante da Caixa de Crédito Agrícola, uma das entidades patrocinadoras do evento em conjunto com a HP.

A iDrone Cup é organizada pelo curso de Engenharia Eletrotécnica e de Computadores do IPCA e decorreu este ano, pela primeira vez, no Pavilhão Municipal de Barcelos.



«in <http://ipca.pt/>»

Google quer pô-lo a interagir com wearables sem ter de lhes tocar

Pela conferência de programadores da Google passaram várias novidades. O projeto Soli foi uma delas e mostra novas formas de interação com a tecnologia que a empresa está a fazer tudo para pôr em prática.

O Projeto Soli tem concentrado um conjunto de investigadores da Google no esforço de usar tecnologias de radar para criar novos sensores, capazes de detetar movimentos muito ligeiros com um nível de precisão elevado.

Traduzindo, as tecnologias de radar transmitem ondas de rádio que ao interceptarem um objeto recuperam parte desse sinal e por essa via informação sobre a direção, velocidade ou altitude do objeto.

Os radares não são uma tecnologia nova e são aliás usados nas mais diversas áreas, como a segurança, em sistemas de monitorização criados para esse efeito. Ajudam a monitorizar a atividade aérea, de mísseis ou fenómenos meteorológicos, por exemplo.

O desafio que a Google se propôs alcançar foi o de integrar esta tecnologia num chip que pudesse ser produzido em escala e que coubesse nos pequenos dispositivos que usamos todos os dias, como os wearables.

Num equipamento com este tipo de sensor-radar é possível criar novas formas de interação e substituir o toque direto por gestos, que têm de ser feitos perto do aparelho mas que substituem o toque físico.

Num vídeo divulgado pela empresa explica-se o conceito e mostra-se como funciona. Sabe-se também que a empresa está a trabalhar numa API para a tecnologia, que deverá lançar ainda este ano.

«in Tek.Sapo.pt»

Upgrade gratuito do Windows 10 disponível a 29 de julho

Os rumores já circulavam há algum tempo, mas agora a informação é oficial: a Microsoft anunciou que o Windows 10 estará disponível a partir de dia 29 de julho em 190 mercados e como atualização gratuita para todos os utilizadores que possuem as versões Windows 7 e 8.1.

De acordo com o comunicado da Microsoft, a partir de 29 de julho, os consumidores terão um ano para poderem realizar o upgrade gratuito para Windows 10 em PCs e tablets. As versões para outro tipo de dispositivos estarão disponíveis mais perto do final do ano.

“Uma vez que um dispositivo com Windows seja atualizado para o Windows 10, a Microsoft irá continuar a manter a sua

atualização ao longo de toda a vida útil do dispositivo, sem nenhum custo associado”, refere a empresa.

Os interessados em fazer a atualização do sistema operativo poderão reservar uma cópia gratuita nas próximas semanas.

«in Exame Informática»

Investigadores portugueses testam sensores de risco na Volvo Ocean Race

Investigadores do Instituto Fraunhofer desenvolveram uma solução de deteção de situações de risco para embarcações. Além dos perigos, o projeto mWaterSafety também permite analisar o desempenho de participantes em desportos náuticos.

A regata Volvo Ocean Race conta com 32 barcos Optimist – mas há dois que se distinguem de todos os outros, através da inclusão de um sistema de sensores de deteção e alerta de perigos desenvolvido pela DigitalWind e o Instituto Fraunhofer. A solução é conhecida por mWaterSafety.

Em comunicado, o Instituto Fraunhofer explica que o sistema de alertas tem por elemento central uma app de telemóveis e sensores dispersos pelas embarcações e pelos membros das tripulações.

Apesar da demonstração agendada para 31 de maio ter por cenário uma das mais famosas regatas do mundo, o mWaterSafety também está apto a ser usado noutros desportos e atividades aquáticas (os mentores do projeto dão como exemplo o Windsurf ou a vela).

A solução pressupõe a instalação de uma estação técnica em terra firme, que terá como objetivo monitorizar, em tempo real, o comportamento das embarcações a partir dos dados captados pelos diferentes sensores instalados a bordo. O comunicado da Fraunhofer refere que, no caso da Volvo Ocean Race, a estação técnica ficará instalada num espaço da Direção Geral de Política do Mar (DGPM).

Além de indicações sobre o posicionamento e as rotas das embarcações, o mWaterSafety também está apto a recolher dados que poderão ser úteis para descrever o desempenho de desportos náuticos.

O projeto, que é financiado pelo QREN, contou com a participação Direção-Geral da Autoridade Marítima (DGAM), o Sporting Clube de Aveiro, o Clube de Vela da Costa Nova e a Associação Náutica da Gafanha da Encarnação.

«in Exame Informática»

C.H.I.P., o computador de oito euros

Uma empresa norte-americana já angariou o equivalente a 1,35 milhões de euros para produzir em massa um pequeno computador, com o custo de 8,1 euros. Conheça o C.H.I.P.

Com um tamanho extremamente reduzido, este novo computador que estará disponível ao público em dezembro, é capaz de realizar tarefas como navegar na Internet, fazer edição de texto ou consultar o Facebook.

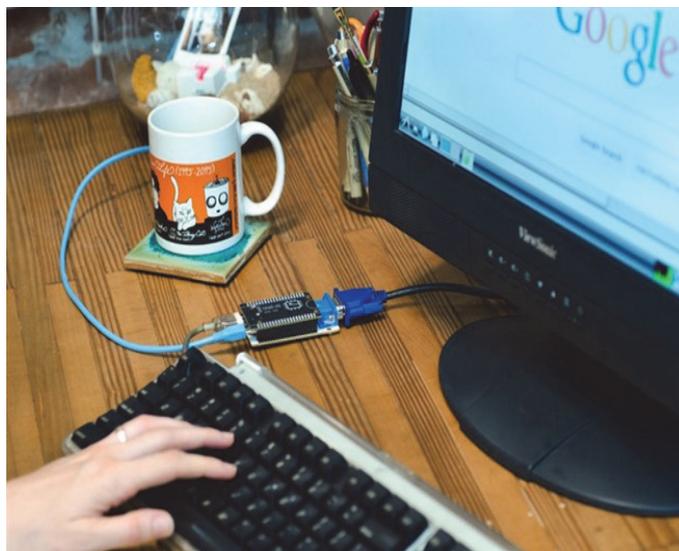
Dentro da pequena caixa (mede 5,8 cm por 3,8 cm), o utilizador poderá encontrar um processador de 1 GHz, 512 Mb de memória RAM e 4 Gb de capacidade de armazenamento interno. Há ainda espaço para uma antena Wi-Fi e uma outra Bluetooth.

Com o C.H.I.P. na mão, o utilizador poderá ligar-lhe um teclado e um monitor, como faz com qualquer computador, afirma a Next Thing Co., responsável pelo desenvolvimento deste produto.

Com 8,10 euros, o utilizador compra um computador funcional, mas a partir deste produto base, é possível comprar adaptadores para dotar o C.H.I.P. de uma porta VGA ou HDMI ou até para o tornar num pequeno computador portátil, com teclado incluído e ecrã tátil.

O computador chega às mãos dos utilizadores com várias aplicações grátis disponíveis e com a capacidade de lhe ser instalado um sistema operativo Linux.

De acordo com a Next Thing Co., que ainda procura financiamento no Kickstarter para avançar com o projeto, este aparelho é ideal para ser utilizado por programadores para testar software ou para ser o coração de algum aparelho eletrónico, mas também para o utilizador comum.



« in Jornal de Notícias »

TEMA DE CAPA

ASP.NET 5

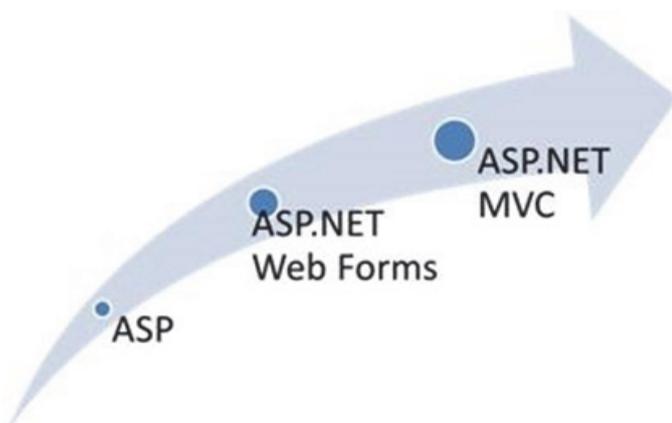
TEMA DA CAPA

ASP.NET 5

Introdução

Passados mais de 13 anos sobre o lançamento do ASP.NET, a Microsoft decidiu começar do zero... ou quase. Com efeito, com o ASP.NET 5 vamos assistir a uma revolução no desenvolvimento para a web, que irá levar a uma reaprendizagem de todo o processo de desenvolvimento.

Um Bocado de História



Quando o ASP.NET foi lançado, em 2002, a designação Web Forms era quase um pleonasmo: basicamente, não havia outro modelo de desenvolvimento para a web na *framework* .NET. A ideia da Microsoft era reproduzir, tanto quanto possível, em ambiente *web* o modelo de desenvolvimento já existente para o Windows:

- Designer visual de páginas e controlos (“*drag and drop*”);
- Modelo de eventos;
- Manutenção automática de estado.

Em grande parte, estes objectivos foram bem-sucedidos: era possível a uma pessoa com relativamente poucos conhecimentos de desenvolvimento produzir aplicações ricas com controlos visuais e acesso a dados de forma rápida. Muitos fabricantes de *software* começaram a produzir bibliotecas de controlos e *frameworks* de desenvolvimento sobre Web Forms, permitindo a construção facilitada de ricas interfaces gráficas com funcionalidades avançadas, quase ao nível do que é possível encontrar em aplicações Windows Forms e Windows Presentation Foundation (WPF). A boa integração dos vários produtos da Microsoft com o Visual Studio deixou muita gente satisfeita durante bastante tempo. O SharePoint, o servidor colaborativo “de bandeira” da Microsoft, usa Web Forms, pelo que quem desenvolve para

SharePoint tem obrigatoriamente de conhecer a *framework*.

Algumas versões notáveis do ASP.NET:

- 1.0: versão inicial (Visual Studio .NET, Visual Studio 2003);
- 2: introdução do mecanismo de *providers* (fornecedores de funcionalidades); disponibilização das ASP.NET AJAX Extensions; adaptadores de CSS configuráveis (Visual Studio 2005);
- 3.5: integração das ASP.NET AJAX Extensions; novos controlos ListView e DataPager; LinqDataSource e EntityDataSource; *routing*; disponibilização do modelo MVC (Visual Studio 2008);
- 4.0: páginas assíncronas; integração dos controlos gráficos ASP.NET Chart Controls; transformações do Web.config; mais fornecedores de funcionalidades (codificação de respostas, *cache*); novo modelo de validação de pedidos; inclusão do jQuery e do Modernizr; suporte a Content Delivery Networks (CDNs); *view state* opcional por página e controlo; geração de identificadores no lado do cliente configurável; possibilidade de endereçar diferentes versões do .NET; geração de HTML otimizado; disponibilização do ASP.NET Dynamic Data (Visual Studio 2010);
- 4.5: *binding model* para controlos de exibição e inserção de dados; empacotamento (*bundling*) e minificação integrada; optimizações na *garbage collection*; suporte a mais funcionalidades do HTML 5; suporte a WebSockets; integração da biblioteca Microsoft AntiXSS; integração do WebAPI; integração do SignalR; integração de formas de autenticação sociais, como Facebook, Twitter, etc (Visual Studio 2012, Visual Studio 2013).

Entretanto, sobre Web Forms foram sendo construídas outras frameworks: XML Web Services e Web Services Enhancements (para a versão 1.0), ASP.NET AJAX Extensions (2), ASP.NET Provider Model (2), Dynamic Data (3.5), Routing (3.5), SignalR (4.5), para citar apenas algumas, e a própria base foi evoluindo, tornando-se mais extensível e completa, com um mecanismo que permite trocar grande parte das funcionalidades nativas. A versão mais recente é a 4.5.2.

Críticas ao Modelo Web Forms

Grande parte das críticas ao modelo Web Forms assenta essencialmente nos seguintes aspectos:

- A *web* é complexa e o modelo Web Forms esconde essa complexidade (manutenção de estado, actuali-

zações parciais, etc);

- A utilização do *designer* visual para produzir funcionalidades conceptualmente complexas pode levar a uma má separação de código, onde a própria página chega a conter código SQL ou LINQ;
- A existência do *view state* e a dependência que certos controlos têm dessa funcionalidade, que por vezes faz com que as páginas fiquem extremamente “pesadas”, contendo grandes quantidades de dados que, de forma invisível, atrasam a sua submissão;
- A dependência de uma biblioteca JavaScript embutida para algumas das suas funcionalidades (ASP.NET AJAX Library);
- A complexidade/pouca qualidade do código gerado automaticamente pelos seus controlos: até há bem pouco tempo estes geravam elementos TABLE e outro conteúdo HTML que actualmente é considerado má prática;
- O modelo “*same-page*” por omissão leva a que uma só página contenha muita lógica, porque a página faz submissões para si própria e tem de considerar várias acções possíveis aquando do processamento dos eventos;
- O ciclo de vida de uma página e dos seus controlos é complexo; há certas coisas que têm de ser feitas em (ou até) determinado evento e por vezes é difícil sincronizar as dependências entre os vários controlos e a própria página (“*event hell*”);
- Monolítico e pesado; as novas versões apenas são distribuídas com a própria *framework* .NET, o que não acontece tão frequentemente como isso; além disso, para usar uma qualquer funcionalidade, temos de trazer várias dependências atrás;
- Apesar da adição tardia de *routing*, as páginas Web Forms não são geralmente amigas de Search Engine Optimization (SEO), dependendo de URLs relativamente crípticos (ex: /Categorias.aspx?ID=b1aee664-aa95-44bb-a0c8-45a567b56919, por oposição a /Categoria/Smartphone).

ASP.NET MVC

ASP.NET MVC Releases



O modelo de desenvolvimento MVC foi lançado em 2009 com o objectivo de fornecer uma alternativa “oficial” a quem não gostasse do Web Forms. Como grandes vantagens, propunha:

- A familiaridade de um Design Pattern, Model-View-Controller (MVC), usado noutras linguagens e *frameworks* de desenvolvimento para a web (Java, PHP);
- Uma distinção clara entre as várias camadas de desenvolvimento, conducente a uma melhor separação de responsabilidades, que eventualmente poderá levar a código mais fácil de manter e evoluir;
- O regresso ao HTTP e HTML: o programador passa a ter de considerar questões como os verbos HTTP, os URLs e a gerar ele próprio o HTML dos conteúdos, o que lhe dá mais controlo; a necessidade de produzir HTML levou a uma maior utilização de *frameworks* JavaScript;
- O modelo de rotas leva a URLs mais “amigos” de REST e de Search Engine Optimization (SEO), tópicos “quentes” actualmente;
- Mais facilidade em testar o código por meio de testes unitários;
- A disponibilização de actualizações “*out-of-band*”, ou seja, não coincidentes com as actualizações da *framework* .NET, levando a que possam ser mais frequentes;
- Elevada extensibilidade: abraçando conceitos modernos, tais como Dependency Injection (DI) e Inversion of Control (IoC), é possível substituir ou complementar grande parte dos seus mecanismos internos (validação, autenticação, autorização, *logging*, etc). Praticamente todas as funcionalidades são extensíveis.

O modelo MVC tornou-se muito popular. Apesar de ainda não poder rivalizar com o Web Forms nalguns aspectos – falta de *designer* visual, menor capacidade de reutilização de controlos – rapidamente se tornou a *framework* de escolha para muitos programadores com tecnologias Microsoft, descontentes com o modelo anterior. A própria Microsoft pareceu empurrar nessa direcção, incluindo até o apoio a projectos construídos sobre MVC, externos (Orchard, por exemplo) ou internos (Web API, Web Pages, Razor) e à especificação OWIN. Como prova dessa evolução, podemos ver que passou da versão 1, em 2009, para a versão 5 em 2013, com vários lançamentos pelo meio, estando actualmente na 5.2.3.

TEMA DA CAPA

ASP.NET 5

OWIN e Open Source

Numa tentativa de “democratizar” o ASP.NET, levando-o a outros sistemas operativos, juntamente com o resto da família .NET, a Microsoft libertou grande parte do código fonte como *open source*, sob o licenciamento Microsoft Public License (MS-PL). Adicionalmente, tem vindo a trabalhar numa especificação que define o fluxo de processamento de um pedido pelo servidor e como o código .NET se pode integrar com um servidor HTTP que não exclusivamente o IIS: é o standard Open Web Integration for .NET (OWIN). O problema é que OWIN, neste momento, não se integra verdadeiramente com Web Forms, embora seja possível usá-los em conjunto, de forma a suportar componentes que dependam de OWIN num cenário Web Forms, mas a *framework* Web Forms propriamente dita não o usa, ao contrário da MVC. Todo o desenvolvimento continua a assentar na velha *pipeline* ASP.NET e nas bibliotecas System.Web.DLL e System.Web.Extensions.DLL.

ASP.NET vNext

Na realidade o que temos em cima da mesa é não uma mas duas versões do ASP.NET:

- 4.6: trata-se da evolução natural da versão actual do ASP.NET; inclui alguns melhoramentos em Web Forms e MVC e integra várias correcções de segurança entretanto lançadas;
- 5: reescrita total do ASP.NET.

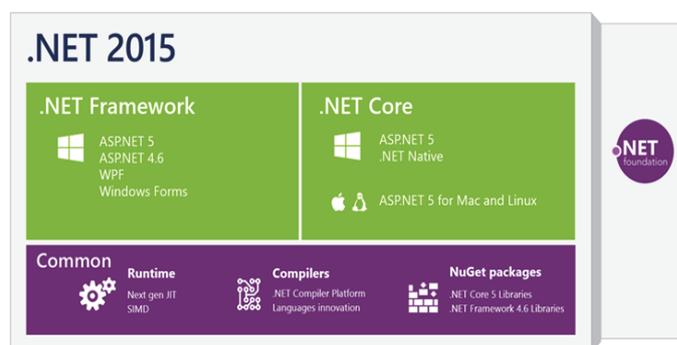
Foquemo-nos no ASP.NET 5. Esta nova *framework* – pois é disto que estamos a falar – vai funcionar sobre o .NET 5. As suas principais características vão ser:

- Totalmente *open source* sob a licença MIT; serão aceites contributos da comunidade, como já agora acontece;
- Fim do modelo de desenvolvimento Web Forms; não será incluído com o ASP.NET 5 qualquer classe de suporte ao Web Forms;
- C# e VB serão suportados, apesar de inicialmente a Microsoft ter anunciado que não existiria suporte inicial para VB;
- Baseado em OWIN, sendo que a *framework* “natural” de desenvolvimento será MVC e Razor, mas será alojável em vários servidores, desde o IIS até o novo Kestrel, desenvolvido de raiz para Mac e Linux, passando por correr num processo .NET;
- Por assentar em .NET 5, será implicitamente multi-plataforma, devendo correr em Windows, Mac e Linux, com suporte a versões limitadas do .NET (.NET Core CLR) a pensar na *cloud*, e podendo usar

simultaneamente componentes de várias versões do .NET;

- Será modular e suportado em packages NuGet: as novas versões serão distribuídas “*out-of-band*” sob a forma de packages NuGet; os programadores poderão escolher apenas aquelas de que necessitam; estas serão actualizadas em ciclos próprios, não ditados pelas novas versões da *framework* .NET;
- Unificação dos APIs MVC, Web API e Web Pages, para evitar duplicação e fornecer um modelo coerente e coeso, sobre a *pipeline* especificada pelo OWIN;
- Totalmente extensível por meio de DI e IoC, sendo fornecido um contentor próprio, que pode ser substituído por um mais tradicional (AutoFac, Unity, Ninject, etc);
- Compilação dinâmica: deixa de ser necessário compilar o código explicitamente para experimentar as alterações, estas são detectadas automaticamente e compiladas pelo novo compilador Roslyn;
- Novas ferramentas de linha de comandos com nomes estranhos: DNX, DNVM e DNU;
- Integração com bibliotecas populares de desenvolvimento e gestão de dependências JavaScript, como Bower, Grunt, Gulp e NPM;
- Gestão de dependências de bibliotecas .NET por meio de *packages* NuGet e suas versões em ficheiros JSON;
- Suporte ao HTTP 2.0, quando a correr sobre o Windows 10 ou superior.

The open .NET ecosystem

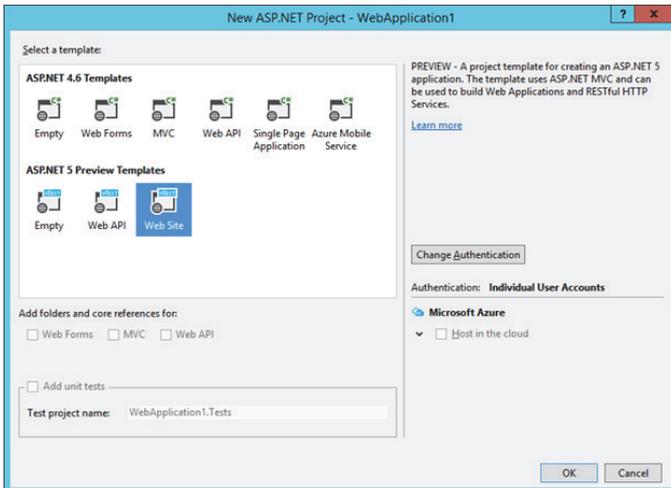


Reconhecendo os problemas actuais – código monolítico, muito agarrado e com muitas dependências, obrigatoriedade de respeitar os ciclos mais lentos de lançamento da *framework* .NET, grande disparidade e duplicação de funcionalidades entre o MVC, Web API e Web Pages, a Microsoft decidiu começar de raiz e reescrever a *framework* a partir do

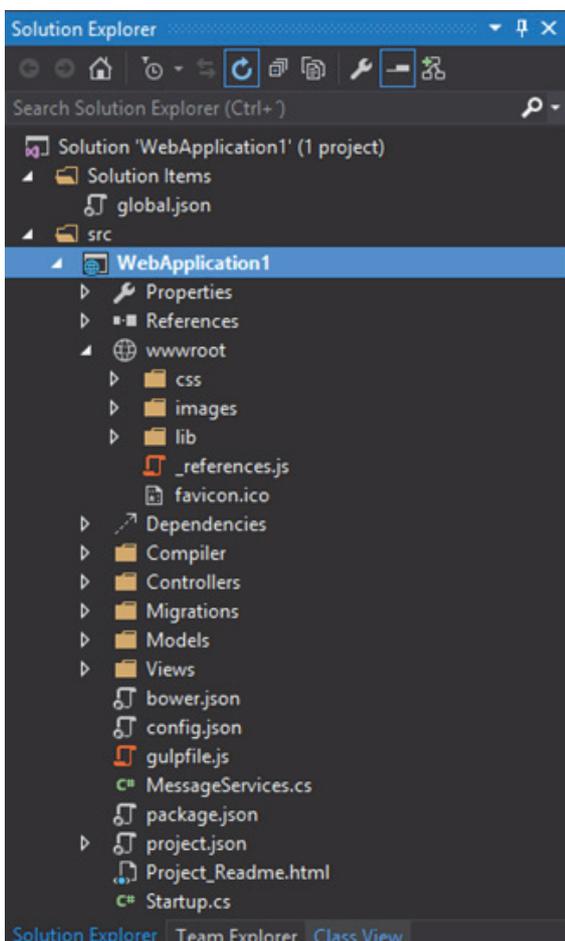
zero. O novo ambiente de desenvolvimento integrado (IDE) será o Visual Studio 2015 e irá suportar quer a nova *framework* baseada em .NET 5 quer as anteriores.

Exemplos

Ao criar um novo projecto web com o Visual Studio 2015 RC somos confrontados com as seguintes opções:



Um projecto ASP.NET 5 consiste numa pasta com um ficheiro **project.json**:



Dentro deste, temos algo como:

```
{
  "webroot": "wwwroot",
  "userSecretsId": "aspnet5-WebApplication1-108e0908-d4ac-4341-920e-ce646e4c2b33",
  "version": "1.0.0-*",

  "dependencies": {
    "Microsoft.AspNet.Mvc": "6.0.0-beta4",
    "Microsoft.AspNet.Mvc.TagHelpers": "6.0.0-beta4",
    "Microsoft.AspNet.Diagnostics": "1.0.0-beta4",
    "Microsoft.AspNet.Diagnostics.Entity": "7.0.0-beta4",
    "Microsoft.AspNet.Server.IIS": "1.0.0-beta4",
    "Microsoft.AspNet.Server.WebListener": "1.0.0-beta4",
    "Microsoft.AspNet.StaticFiles": "1.0.0-beta4",
    "Microsoft.AspNet.Tooling.Razor": "1.0.0-beta4",
    "Microsoft.Framework.ConfigurationModel.Json": "1.0.0-beta4",

    "Microsoft.Framework.ConfigurationModel.UserSecrets": "1.0.0-beta4",
    "Microsoft.Framework.CodeGenerators.Mvc": "1.0.0-beta4",
    "Microsoft.Framework.Logging": "1.0.0-beta4",
    "Microsoft.Framework.Logging.Console": "1.0.0-beta4",

    "Microsoft.VisualStudio.Web.BrowserLink.Loader": "14.0.0-beta4"
  },

  "commands": {
    "web": "Microsoft.AspNet.Hosting --server Microsoft.AspNet.Server.WebListener --server.urls http://localhost:5000",
    "gen": "Microsoft.Framework.CodeGeneration",
  },

  "frameworks": {
    "dnx451": { },
    "dnxcore50": { }
  },

  "exclude": [ "wwwroot", "node_modules", "bower_components" ],

  "publishExclude": [ "node_modules", "bower_components", "**.xproj", "**.user", "**.vsspscc" ],

  "scripts": {
    "postrestore": [ "npm install", "bower install" ],
    "prepare": [ "gulp copy" ]
  }
}
```

São perceptíveis os seguintes elementos:

- Informação genérica do projecto: **webroot**, **userSecretsId**, **version**; **userSecretsId** contém um identificador único da aplicação usado para localizar um ficheiro de configuração pessoal por utilizador;
- Dependências NuGet: **dependencies**;
- Comandos: **commands**; neste exemplo, o comando **web** despoleta o servidor HTTP **WebListener**;

TEMA DA CAPA

ASP.NET 5

- Frameworks suportadas: **frameworks**;
- Pastas e ficheiros excluídos do projecto (por omissão, todos os ficheiros dentro de uma pasta estão incluídos): **exclude**;
- Pastas e ficheiros excluídos da publicação: **publishExclude**;
- Scripts executáveis: **scripts**. São visíveis elementos do Bower, NPM e Gulp.

Não iremos percorrer todos estes elementos, a maior parte deles é facilmente compreensível. É interessante ver que o ficheiro possui IntelliSense:

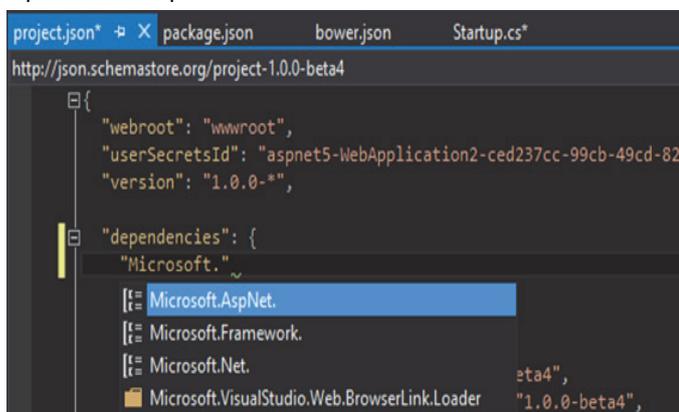


Figura 1 IntelliSense no ficheiro project.json

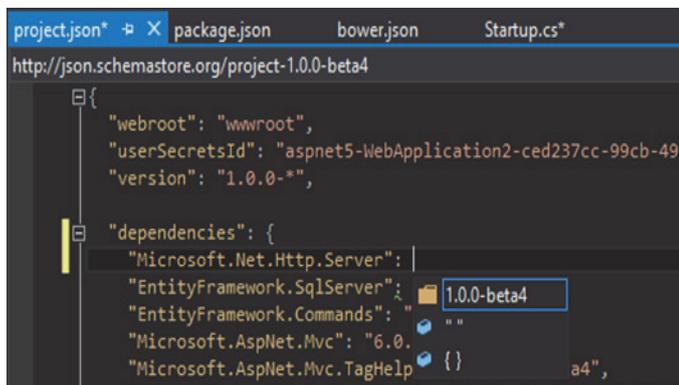


Figura 2 IntelliSense nas packages NuGet

Ao adicionar uma dependência, o Visual Studio detecta a alteração e faz o seu *download*:

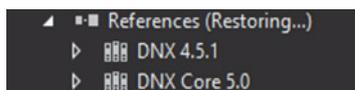


Figura 3 O Visual Studio a actualizar as packages

Não existem mais os ficheiros **Web.config** e **Global.asax/Global.asax.cs**: a configuração é agora extensível sendo por omissão suportados ficheiros JSON e as variáveis de ambiente, e a *pipeline* do projecto é inteiramente OWIN. A

classe que por convenção permite configurar a aplicação chama-se **Startup**:

```
public class Startup
{
    public Startup(IHostingEnvironment env)
    {
        // Definir a fonte da configuração a
        //partir de um ficheiro e das variáveis
        //de ambiente
        var configuration = new Configuration()
            .AddJsonFile("config.json")
            .AddUserSecrets()
            .AddEnvironmentVariables();

        // Guardar a configuração numa
        //propriedade local
        this.Configuration = configuration;
    }

    public IConfiguration Configuration { get; set; }

    public void ConfigureServices(
        IServiceCollection services)
    {
        // Adicionar um serviço com scope do
        //pedido HTTP actual
        services.AddScoped<IService,
            ServiceImplementation>();
        // Adicionar um serviço a ser criado em
        //cada pedido por ele
        services.AddTransient<IService,
            ServiceImplementation>();
        // Adicionar um serviço singleton
        services.AddSingleton<IService>
            (new ServiceImplementation());

        // Adicionar MVC ao contentor de IoC e DI
        services.AddMvc();
        // Descomentar para adicionar Web API
        // services.AddWebApiConventions();
    }

    public void Configure(IAApplicationBuilder
        app, IHostingEnvironment env, ILoggerFactory
        loggerfactory)
    {
        // Configurar o logging
        loggerfactory.AddConsole(minLevel:
            LogLevel.Warning);

        // Se estivermos em ambiente de
        //desenvolvimento, mostrar erros
        if (env.IsEnvironment("Development"))
        {
            app.UseBrowserLink();
            app.UseErrorPage
                (ErrorPageOptions.ShowAll);
            app.UseDatabaseErrorPage
                (DatabaseErrorPageOptions.ShowAll);
        }
        else
        {
            // Caso contrário, redireccionar para
            //este controlador e acção no caso de erros
            app.UseExceptionHandler("/Home/Error");
        }

        // Servir ficheiros estáticos
        app.UseStaticFiles();

        // Configurar rotas MVC
        app.UseMvc(routes =>
        {
```

```
routes.MapRoute(
    name: "default",
    template: "{controller}/{action}/{id?}",
    defaults: new { controller = "Home", action = "Index" });

// Descomentar para configurar rotas
// Web API
// routes.MapWebApiRoute("DefaultApi",
// "api/{controller}/{id?}");
});
}
```

Os três métodos chamados por convenção são:

- **Startup:** permite definir opções de alojamento do serviço HTTP e definições globais;
- **ConfigureServices:** adicionar ou substituir serviços no contendor de Inversion of Control (IoC) e Dependency Injection (DI), com uma de três longevidades possíveis:
 - **Singleton:** apenas existe uma instância, esta é sempre devolvida;
 - **Scoped:** é criada uma instância, caso não exista, por pedido HTTP, após o que, dentro do mesmo pedido, é sempre devolvida a instância criada;
 - **Transient (default):** é criada uma nova instância do serviço sempre que for pedida ao contendor de IoC.
- **Configure:** outros aspectos da configuração, já usando os serviços definidos (rotas, *logging*, etc), adicionar *middleware* à *pipeline* HTTP.

*Nota: alguns nomes da especificação OWIN foram alterados no ASP.NET 5; o interface **IApplicationBuilder** passou para **ApplicationBuilder**, por exemplo.*

O registo de *middleware* – adicionar funcionalidades à *pipeline* HTTP - é tipicamente feito no método **Configure**:

```
// Adicionar um componente à pipeline de execução
app.UseMiddleware<MeasureMiddleware>();
```

Um exemplo de um componente OWIN, que vem substituir os módulos e *handlers* HTTP (**IHttpModule** e **IHttpHandler**) será:

```
public class MeasureMiddleware
{
    private readonly RequestDelegate _next;
    public MeasureMiddleware(RequestDelegate next)
    {
        this._next = next;
    }
}
```

```
public async Task Invoke(HttpContext context)
{
    var sw = Stopwatch.StartNew();

    await this._next(context);

    var ext = Path.GetExtension(
        context.Request.Path.Value);
    if ((context.Response.StatusCode ==
        (Int32) HttpStatusCode.OK) &&
        (String.IsNullOrEmpty(ext) == true))
    {
        var milliseconds =
            sw.ElapsedMilliseconds;
        await context.Response.WriteAsync(
            $"Processado em {milliseconds} milisegundos");
    }
}
```

Este componente faz uso do padrão Chain of Responsibility para invocar o componente anterior (*middleware*) na *pipeline* (recebido no construtor), medindo essa execução e fazendo *output* do tempo que demorou. Tem acesso ao contexto de execução, incluindo todos os detalhes do pedido, e pode adicionar conteúdo às respostas.



A configuração global num ficheiro JSON (**config.json**) fica assim:

```
{
  "AppSettings": {
    "SiteTitle": "WebApplication1"
  },
  "Data": {
    "DefaultConnection": {
      "ConnectionString": "Server=(localdb)
\\mssqllocaldb;Database=aspnet5-
WebApplication1-108e0908-d4ac-4341-
920e-ce646e4c2b33;Integrated
Security=SSPI;MultipleActiveResultSets=true"
    }
  }
}
```

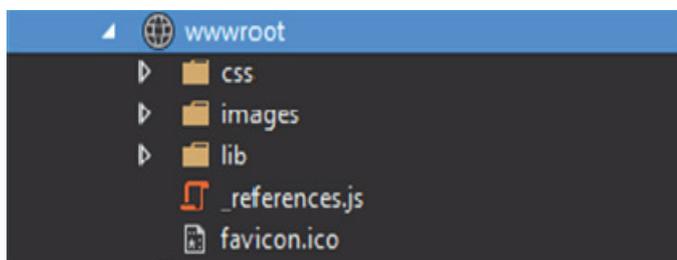
No nosso exemplo, estamos a combinar configuração específica do projecto (**AddJsonFile**) com variáveis de ambiente do utilizador (**AddEnvironmentVariables**) e também com um ficheiro de configuração pessoal do utilizador, o qual não é mantido na mesma pasta do projecto, mas numa pasta específica do utilizador (**%AppData%\Roaming\Microsoft\UserSecrets\\secrets.json**, por omissão).

Outra diferença importante é que a raiz do servidor HTTP, a partir da qual são servidos os ficheiros estáticos, está situada numa pasta chamada, por omissão, **wwwroot**

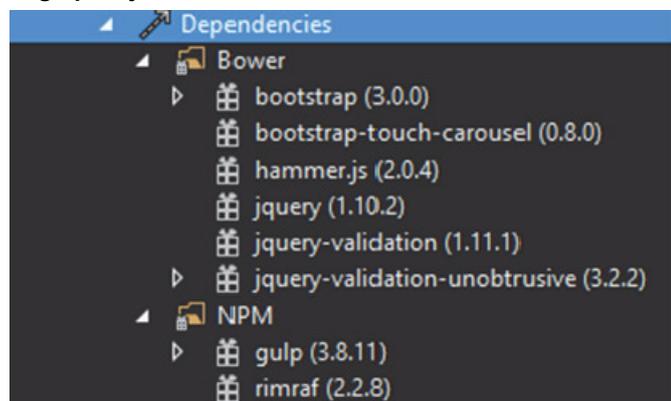
TEMA DA CAPA

ASP.NET 5

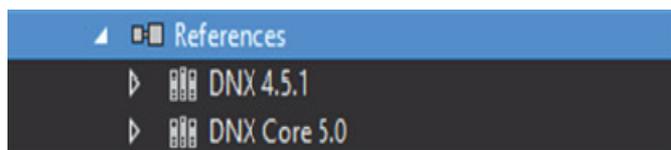
(configurada no **project.json**), e não na pasta do projecto propriamente dito:



Também o Bower e o NPM merecem um tratamento especial, exibindo o Visual Studio uma vista sobre o conteúdo dos ficheiros de configuração **bower.json**, **package.json** e **gulpfile.js**:



As dependências também provêm directamente do ficheiro **project.json**, da sua entrada **frameworks**:



Já os controladores e as vistas, bem como as respectivas rotas, localizações e convenções de acesso, permanecem idênticas aos das versões anteriores do MVC, pelo que não serão cobertos aqui. Destaco apenas três novidades: apenas são suportadas vistas Razor e os controladores são partilhados por MVC e Web API, devendo retornar implementações de **IActionResult**, interface que é comum aos resultados de controladores MVC e Web API. A terceira são na verdade duas formas *server-side* de gerar conteúdos HTML. A primeira chama-se *tag helpers*, e é uma forma de definir *tags* cujos atributos e conteúdo, pelo menos parcialmente, são gerados por código .NET. Estes *tag helpers* são definidos em classes .NET que herdam de **TagHelper**:

```
[TargetElement("square")]
public class SquareTagHelper : TagHelper
{
    [HtmlAttributeName("asp-side")]
    public Int32 ? Side { get; set; }

    [HtmlAttributeName("asp-background-color")]
    public String BackgroundColor { get; set; }
}
```

```
[HtmlAttributeName("asp-foreground-color")]
public String ForegroundColor { get; set; }

[HtmlAttributeName("asp-content")]
public String Content { get; set; }

public override void Process(TagHelperContext
context, TagHelperOutput out-
put)
{
    if ((this.Side != null) &&
        (this.BackgroundColor !=
null))
    {
        //define a tag de output
        output.TagName = "div";

        //adiciona atributos à tag
        output.Attributes["style"] = $"width:
{Side}px; height: {Side}px; color:
{ForegroundColor}; background-color:
{BackgroundColor}; text-align: center;
vertical-align: middle; line-height: {Side}px;";

        //adiciona conteúdo
        output.Content.SetContent
(this.Content);
    }

    base.Process(context, output);
}
```

E registados em vistas Razor, por exemplo, **_GlobalImport.cshtml**, por forma a estarem acessíveis em todas as vistas:

```
@using MyNamespace
@using MyNamespace.Models
@addTagHelper "*",
    Microsoft.AspNet.Mvc.TagHelpers"
@addTagHelper "*", MyNamespace"
```

A instrução **addTagHelper** torna disponível todas as *tag helpers* disponíveis numa *assembly* .NET. Após este registo, podemos usar a nova *tag square*:

```
<square asp-side="200" asp-foreground-
color="white" asp-background-color="blue"
asp-content="Hello, World!"/>
```

Alguns leitores poderão notar semelhanças com o modelo de controlos server-side do ASP.NET Web Forms.

O segundo novo API chama-se *view components*. Aqui a ideia é ter um mecanismo semelhante às *partial views*, simplesmente, sem *designer*, ou seja, todo o conteúdo é produzido por código. Vejamos um exemplo de um *view component*, uma classe que herda de **ViewComponent**:

```
[ViewComponent(Name = "Sum")]
public class AddViewComponent : ViewComponent
{
    public IViewComponentResult Invoke(int a, int b)
    {
        var result = a + b;
    }
}
```

```
    }  
    return this.View(result);  
}
```

E eis como o utilizar numa vista Razor:

```
@Component.Invoke("Sum", 1, 2)
```

O primeiro parâmetro de **Invoke** é o nome do **view component**, tal como definido pelo atributo **ViewComponentAttribute**, caso exista, ou então o nome da classe sem o sufixo **ViewComponent**. Seguidamente vão quaisquer parâmetros, que irão ser passados ao método **Invoke**. Poderão existir vários **overloads** deste método, recebendo diferentes parâmetros e é também possível invocar o **view component** assincronamente.

Com estes dois mecanismos, **tag helpers** e **view components**, torna-se mais fácil a invocação de código **server-side**, e o que não é de somenos importância, a reutilização, já que estes componentes podem existir em **assemblies** externas para as quais foram adicionadas referências.

Novos Ferramentas da Linha de Comandos

À boleia do .NET 5 vem um conjunto de comandos novo; entre outras novidades, passa a ser possível gerir **packages** NuGet e correr aplicações ASP.NET sem usar o Visual Studio. Os três novos comandos com os quais teremos de nos familiarizar são, muito resumidamente:

- **DNVM**: .NET Version Manager, anteriormente chamado KVM; configura a versão do .NET do projecto;
- **DNX**: .NET Execution Environment, anteriormente K, KLR e KRE; permite configurar o ambiente do .NET em uso pelo projecto, de entre .NET Framework (*default*), .NET Core (subset optimizado para *cloud*) e Mono (para Linux e Mac);
- **DNU**: .NET Development Utilities, antigo KPM; utilitários para gerir as **packages** NuGet, criar **packages** e publicar a aplicação.

Estes comandos integram-se com a nova estrutura de projectos .NET (**project.json**). Alguns exemplos:

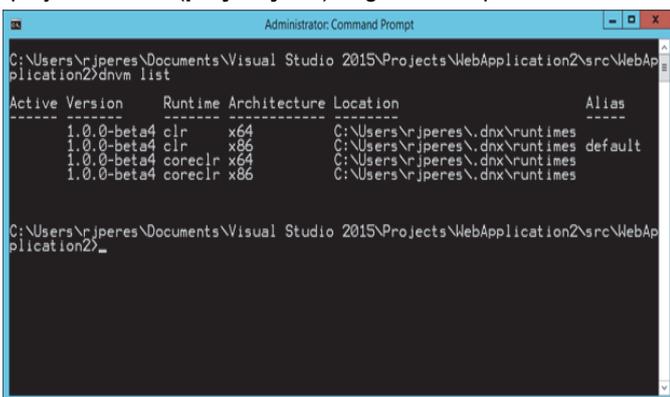


Figura 4 Lista de frameworks instaladas

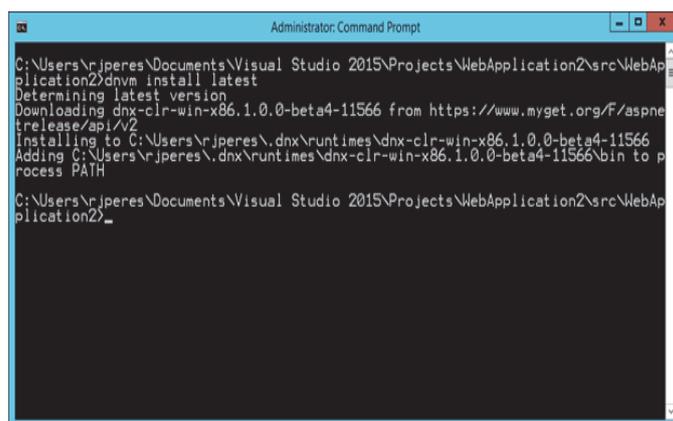


Figura 5 Instalar a última versão das frameworks

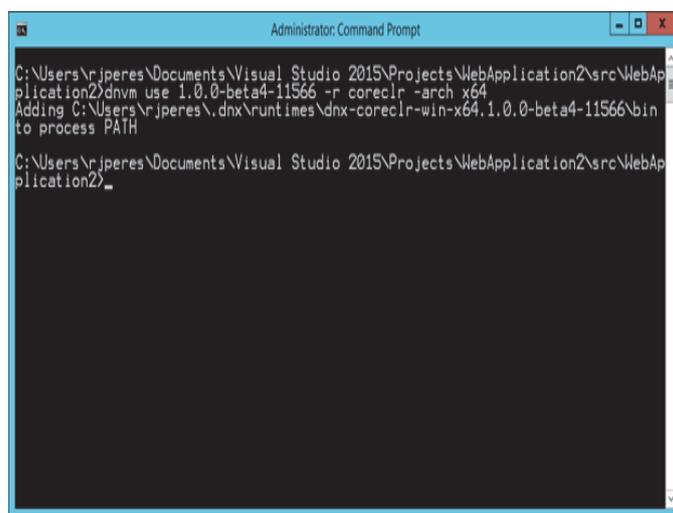


Figura 6 Seleccionar a framework a usar

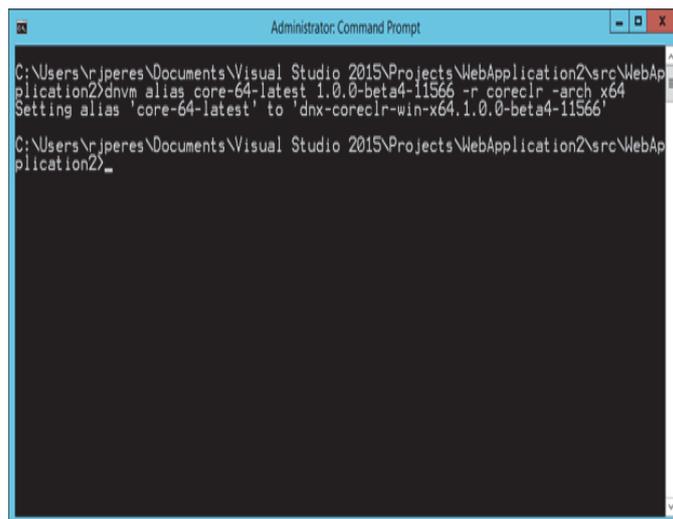


Figura 7 Definir um alias para uma framework

Antes de usar o DNU é necessário instalá-lo, o que pode ser feito com o seguinte comando PowerShell:

```
@powershell -NoProfile -ExecutionPolicy  
unrestricted -Command "&{$Branch='dev';iex (  
(new-object net.webclient).DownloadString  
(https://raw.githubusercontent.com/aspnet/Home/
```

TEMA DA CAPA

ASP.NET 5

```
dev/dnvminstall.ps1'))})"
```

Para correr a aplicação *web*, usamos o DNX, onde o parâmetro **web** provém do **project.json**, secção **commands**:

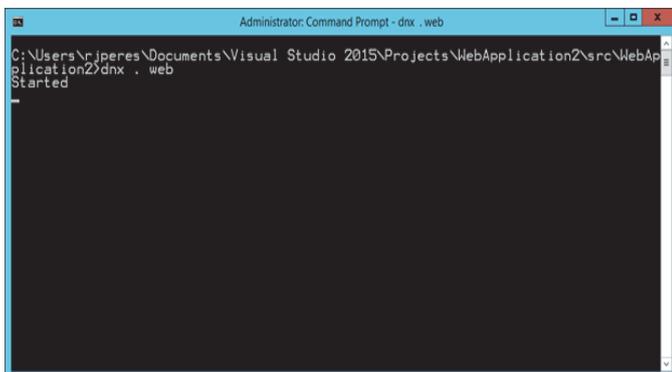


Figura 8 Executar um comando definido no projecto

Para o DNU, os comandos mais típicos serão:

- **dnx restore**: obter todas as packages NuGet especificadas no projecto que estejam em falta;
- **dnx publish**: preparar a aplicação para *deployment*;
- **dnx wrap**: converter um projecto **.csproj** em **project.json**;
- **dnx build**: compilar o projecto.

Os comandos DNX, DNVM e DNU actuam com base e sobre o projecto actual, que pode ser ASP.NET 5 ou outro .NET 5.

“ Certo parece ser um cada vez maior impulso na direcção do Azure, multi-plataforma e também do open source ”

Conclusão

Como o ASP.NET 5, bem como o 4.6, ainda não foram lançados, só nos resta especular. É possível que alguém, ou mesmo a própria Microsoft, venha a implementar Web Forms sobre o ASP.NET 5. Certo parece ser um cada vez maior impulso na direcção do Azure, multi-plataforma e também do *open source*. Durante algum tempo ainda teremos a família 4.x, com suporte a código legado, mas é de prever que esta seja descontinuada num futuro próximo. Esperemos pela versão final do Visual Studio 2015 e ASP.NET 5, que deverá ser lançada ainda este ano, e, entretanto, estejamos atentos às novidades que vão sendo apresentadas.

Referências

Site oficial da família ASP.NET: <http://www.asp.net>

Introduction to ASP.NET 5: <http://docs.asp.net/en/latest/conceptual-overview/aspnet.html>

Updates for ASP.NET 4.6 – Web Forms/ MVC 5/ Web API 2: <http://blogs.msdn.com/b/webdev/archive/2015/04/30/updates-for-asp-net-4-6-web-forms-mvc-5-web-api-2.aspx>

New ASP.NET Features and Fixes in Visual Studio 2015 RC: <http://blogs.msdn.com/b/webdev/archive/2015/04/29/new-asp-net-features-and-fixes-in-visual-studio-2015-rc.aspx>

The MIT License: <https://github.com/dotnet/corefx/blob/master/LICENSE>

Entrada ASP.NET na Wikipedia: <http://en.wikipedia.org/wiki/ASP.NET>

Entrada ASP.NET MVC Framework na Wikipedia: http://en.wikipedia.org/wiki/ASP.NET_MVC_Framework

Entrada Model-view-controller na Wikipedia: <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

Repositório ASP.NET no GitHub: <https://github.com/aspnet/>

ASP.NET vNext: <http://www.asp.net/vnext>

OWIN: Open Web Interface for .NET: <http://owin.org>

Licenciamento Microsoft Public License (MS-PL): <http://opensource.org/licenses/ms-pl.html>

Posts sobre ASP.NET no blogue do autor deste artigo (Development With A Dot): <https://weblogs.asp.net/ricardoperes/Tags/ASP.NET>

Orchard: <http://orchardproject.net/>

AUTOR

Escrito Por Ricardo Peres

A PROGRAMAR

Criando arquivos Office com OpenXML SDK

Criar uma API no Azure App Service

Primeiros passos no desenvolvimento de aplicações Android

Introdução a arquitectura SAP com foco em SAP-1

Raiz enésima iterativa programada

Criando arquivos Office com OpenXML SDK

Introdução

Uma maneira muito comum de introduzir flexibilidade para nossas aplicações é exportar os dados para arquivos Office: ao exportar para o Word, podemos gerar relatórios poderosos, onde o usuário pode formatar os dados da maneira desejada, editar ou remover partes dos dados, ou mesmo complementar com dados de diversas fontes. Exportando para o Excel, podemos criar rapidamente análises diversas, elaborar gráficos ou fazer comparações facilmente.

Uma maneira de exportar os dados para arquivos Office é usar a automação Office, que usa os recursos de automação COM para abrir o programa e gerar os arquivos a partir de nossos dados. Isso, além de muito lento, traz uma segunda desvantagem: obriga que a máquina que está rodando a aplicação tenha o Office instalado.

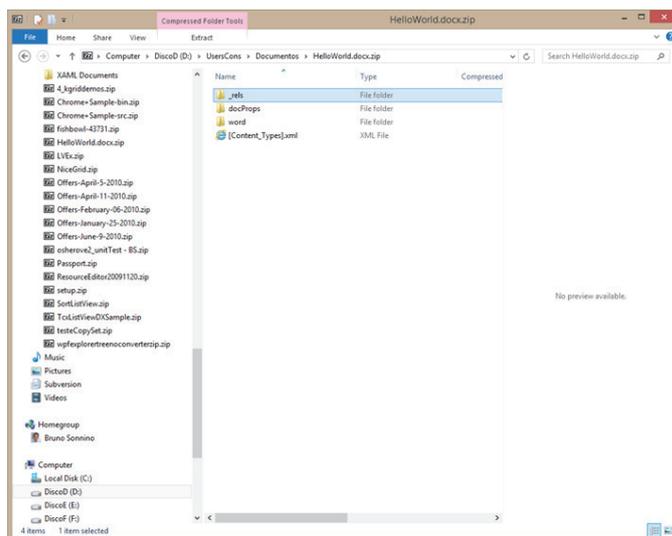
A partir do Office 2007, o formato de arquivos Office mudou, para um formato aberto, o OpenXML (<http://openxmldeveloper.org/>). Isto trouxe algumas vantagens:

- Documenta o formato de arquivos Office – antes do OpenXML, o formato de arquivos Office não era documentado, e para criar manualmente estes arquivos, você tinha que usar muita engenharia reversa e mudar o programa a cada nova versão do Office
- Cria a possibilidade de intercâmbio entre programas – qualquer um pode criar um processador de textos que usa o formato OpenXML sem se preocupar com licenciamento
- Permite que você possa criar um programa para gerar ou alterar arquivos Office facilmente.

Um arquivo OpenXML é um arquivo compactado (no formato ZIP), que contém arquivos XML e outros arquivos de dados. Você pode usar qualquer tecnologia que abra arquivos ZIP e acesse arquivos XML para criar ou modificar os novos arquivos Office.



Se você renomear um arquivo Office para zip e abri-lo, terá algo semelhante a isso:



Você tem um arquivo zip com três pastas e um arquivo xml. No diretório `_rels` você tem um arquivo `.rels`, um xml com a estrutura do documento. A partir daí, você vai “desenrolando” seu documento e obtendo as diversas partes que o compõem: propriedades, textos, imagens, estilos, etc. Parece complicado? Mas nós nem começamos ainda!

Packaging API

Para facilitar a manipulação de arquivos OpenXML, a Microsoft introduziu no .Net 3.0 a Packaging API ([http://msdn.microsoft.com/en-us/library/windows/desktop/dd371623\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/dd371623(v=vs.85).aspx)), uma API para acessar arquivos que tem este formato de pacotes. Esta não é uma API específica para arquivos Office, você pode, por exemplo, abrir um arquivo XPS com a mesma API ou então criar um arquivo que use uma estrutura semelhante, mas que não seja compatível com OpenXML.

Para exemplificar esta API, vamos criar um programa WPF que abre um arquivo Office e mostra as suas relações. No Visual Studio, crie um novo projeto WPF e, na janela principal, coloque o seguinte código:

```
<Grid>
  <Grid.RowDefinitions>
    <RowDefinition Height="*" />
    <RowDefinition Height="40" />
  </Grid.RowDefinitions>
  <ListBox x:Name="LbxDados" />
  <Button Content="Abre"
    HorizontalAlignment="Center" Grid.Row="1"
    VerticalAlignment="Center" Width="75" />
</Grid>
```

A PROGRAMAR

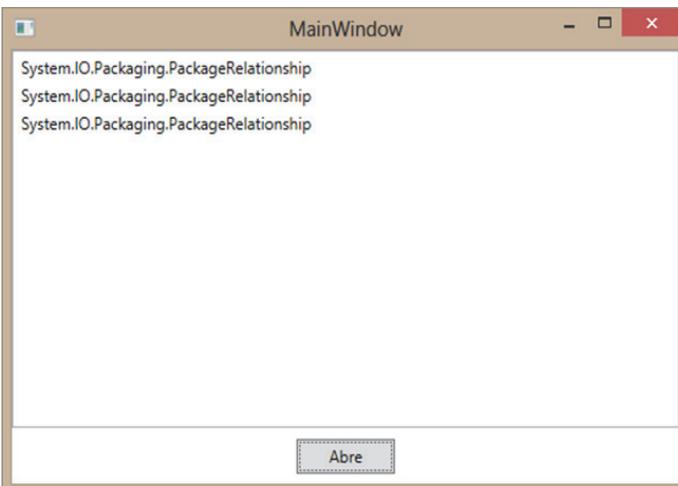
CRIANDO ARQUIVOS OFFICE COM OPENXML SDK

```
Click="AbreArquivoClick"/>
</Grid>
```

No Code Behind, no manipulador do evento **Click** do botão, coloque o seguinte código:

```
private void AbreArquivoClick(object sender,
RoutedEventArgs e)
{
    var openDialog = new OpenFileDialog
    {
        Filter =
            "Arquivos Office
            (*.docx;*.xlsx;*.pptx)| "+
            "*.docx;*.xlsx;*.pptx|Todos Arquivos (*.*)|*.*",
    };
    if (openDialog.ShowDialog() == true)
    {
        using (Package package = Package.Open
            (openDialog.FileName,
            FileMode.Open, FileAccess.Read))
        {
            LbxDados.ItemsSource =
                package.GetRelationships();
        }
    }
}
```

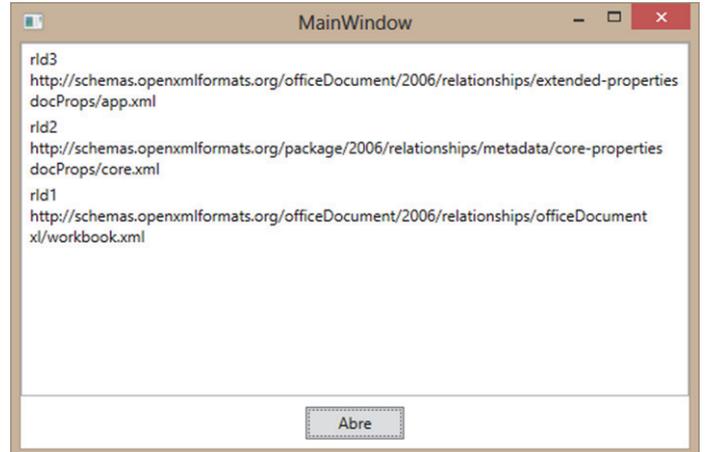
Com isso, iremos pegar as relações do documento aberto e colocá-las na ListBox. Quando você executa o programa, irá ver algo semelhante ao seguinte:



Isto é devido ao fato que a Packaging API abriu o arquivo .rels, fez a análise do arquivo xml e criou classes do tipo **PackageRelationship**. Para apresentar seus valores, basta colocar um ItemTemplate para a ListBox:

```
<ListBox x:Name="LbxDados">
  <ListBox.ItemTemplate>
    <DataTemplate>
      <StackPanel>
        <TextBlock Text="{Binding Id}"/>
        <TextBlock Text="{Binding
          RelationshipType}"/>
        <TextBlock Text="{Binding
          TargetUri}"/>
      </StackPanel>
    </DataTemplate>
  </ListBox.ItemTemplate>
</ListBox>
```

Agora os dados são apresentados como queremos:



Podemos ver aqui dois arquivos de propriedades, **docProps/app.xml** (extended properties) e **docProps/core.xml** (core properties) e que o documento aberto é uma planilha Excel, que está no diretório **xl** e tem o nome **workbook.xml**. Poderíamos, em seguida, abrir o arquivo **workbook.xml** e analisá-lo, mas isso não é um trabalho simples. Teríamos que analisar as dependências do documento, ver quais as partes que compõem ele para montar toda sua estrutura.

Pensando nessa dificuldade, a Microsoft lançou a OpenXML SDK, um kit de desenvolvimento próprio para os arquivos OpenXML.

OpenXML SDK

A OpenXML SDK está baseada na Packaging API e traz classes voltadas para o desenvolvimento de arquivos OpenXML. Você pode encontrar o código fonte e documentação em <https://github.com/OfficeDev/Open-XML-SDK>.

A partir daí, basta adicionar uma referência a **DocumentFormat.OpenXml** e a **WindowsBase** no programa para começar a usar a OpenXML. Alternativamente, você pode usar o NuGet para adicionar a referência, sem precisar instalar a SDK.

Com a OpenXML SDK você não precisa manipular Packages, Relações ou propriedades. Você tem novas classes para manipular os arquivos do Office diretamente.

Por exemplo, você tem as classes **WordprocessingDocument**, **SpreadsheetDocument** e **PresentationDocument** para trabalhar com documentos, planilhas ou apresentações. O código a seguir cria um arquivo Word com uma frase de texto:

```
class Program
{
    static void Main(string[] args)
    {
        if (args.Length < 2)
        {
```

A PROGRAMAR

CRIANDO ARQUIVOS OFFICE COM OPENXML SDK

```
        Console.WriteLine("uso: CriaWordDoc
                           <NomeArquivo> <Texto>");
        return;
    }
    CriaDoc(args[0], args[1]);
}

public static void CriaDoc(string filepath,
                          string msg)
{
    using (WordprocessingDocument doc =
           WordprocessingDocument.Create(filepath,
           WordprocessingDocumentType.Document))
    {
        MainDocumentPart mainPart =
            doc.AddMainDocumentPart();

        mainPart.Document = new Document();
        Body body =
            mainPart.Document.AppendChild(new Body());
        Paragraph para = body.AppendChild(new
            Paragraph());
        Run run = para.AppendChild(new Run());
        run.AppendChild(new Text(msg));
        para.AppendChild(new Run());
    }
}
```

Inicialmente, você cria um **WordprocessingDocument**, adiciona um **MainDocumentPart** e atribui a propriedade **Document** a um novo **Document**. Em seguida, adicionamos o **Body**, um **Paragraph** e, a ele, um **Run** com o texto. Sem dúvida, isso é mais fácil que trabalhar com os **Packages** e arquivos xml!

Agora que já conhecemos as classes, podemos usá-las para exportar dados para os arquivos Office. Vamos ver como exportar dados de uma lista para uma planilha Excel.

```
private static void ExportaLivrosParaExcel()
{
    // Cria planilha Excel
    using (SpreadsheetDocument doc =
           SpreadsheetDocument.Create("livros.xlsx",
           SpreadsheetDocumentType.Workbook))
    {
        // Cria um Workbook
        WorkbookPart workbookpart =
            doc.AddWorkbookPart();
        workbookpart.Workbook = new Workbook();

        // Cria planilha no Workbook
        var worksheetPart =
            doc.WorkbookPart.AddNewPart<WorksheetPart>();
        worksheetPart.Worksheet = new Worksheet();

        //Dados da planilha
        worksheetPart.Worksheet.AppendChild(new
            SheetData());
        var sheetData = worksheetPart.Worksheet.
            GetFirstChild<SheetData>();

        // Cria cabeçalho
        var rowIndex = 1u;
        var row = new Row { RowIndex =
            rowIndex++ };
        sheetData.Append(row);
        InserirDadoNaCelula(row, 0, "Id",
            CellValues.String);
        InserirDadoNaCelula(row, 1, "Autor",
            CellValues.String);
        InserirDadoNaCelula(row, 2, "Título",
            CellValues.String);
    }
}
```

```
        CellValues.String);
        InserirDadoNaCelula(row, 3, "Gênero",
            CellValues.String);
        InserirDadoNaCelula(row, 4, "Preço",
            CellValues.String);
        InserirDadoNaCelula(row, 5, "Data
        Publicação", CellValues.String);
        InserirDadoNaCelula(row, 6, "Descrição",
            CellValues.String);

        // Cria linhas com dados
        foreach (var livro in _livros)
        {
            row = new Row { RowIndex =
                rowIndex++ };
            sheetData.Append(row);
            InserirDadoNaCelula(row, 0, livro.Id,
                CellValues.String);
            InserirDadoNaCelula(row, 1, livro.Autor,
                CellValues.String);
            InserirDadoNaCelula(row, 2,
                livro.Titulo, CellValues.String);
            InserirDadoNaCelula(row, 3,
                livro.Genero, CellValues.String);
            InserirDadoNaCelula(row, 4,
                livro.Precio.ToString(
                CultureInfo.InvariantCulture),
                CellValues.Number);
            InserirDadoNaCelula(row, 5,
                livro.DataPublicacao.ToString("yyyy-MM-dd"),
                CellValues.Date);
            InserirDadoNaCelula(row, 6,
                livro.Descricao, CellValues.String);
            worksheetPart.Worksheet.Save();
        }
        worksheetPart.Worksheet.Save();

        // Adiciona planilha
        doc.WorkbookPart.Workbook.AppendChild(new
            Sheets());
    }
}
```

```
doc.WorkbookPart.Workbook.GetFirstChild<Sheets>
    ().AppendChild(new Sheet()
    {
        Id = doc.WorkbookPart.GetIdOfPart
            (worksheetPart),
        SheetId = 1,
        Name = "Planilha 1"
    });
    workbookpart.Workbook.Save();
    doc.Close();
}
```

A criação de uma planilha Excel é um pouco mais complexa. Inicialmente temos o documento (**SpreadsheetDocument**). Neste documento, inserimos um **Workbook**, e no **Workbook**, uma **Worksheet**. O **Worksheet** é composto de **SheetData**, onde vão os dados da planilha. Você deve inserir as linhas e, nas linhas, células. Com este processo, você pode criar uma nova planilha com dados. A função **InserirDadoNaCelula**, insere o dado na célula. Passamos a linha onde a célula será inserida, a coluna, o texto e o tipo do dado. A função é a seguinte:

```
private static void InserirDadoNaCelula(Row linha,
int coluna, string dado, CellValues tipo)
{
    var indiceCelula = ObtemColuna(coluna) +
        linha.RowIndex.ToString();
}
```

```
// Obtém célula pela referência (ex. A1)
Cell refCell = linha.Elements<Cell>
    ().FirstOrDefault(
        cell => string.Compare
        (cell.CellReference.Value, indiceCelula, true) >
        0);

// Se a célula não existir, cria uma e insere
//na linha
if (refCell == null)
{
    refCell = new Cell() { CellReference =
        indiceCelula };
    linha.InsertBefore(refCell, null);
}

// Configura o dado da célula
refCell.CellValue = new CellValue(dado);
refCell.DataType = new EnumValue<CellValues>
    (tipo);
```

A partir da linha e da coluna, obtemos uma referência para a célula no formato “A1”, verificamos se ela já existe na planilha e, se não existir, ela será criada. Finalmente atribuímos o valor à célula. Embora o processo seja um pouco mais complicado, uma vez que dominamos os conceitos, a criação de planilhas fica bastante simples.

Alterando documentos com OpenXML SDK

Conhecendo a maneira de acessar arquivos OpenXML, podemos também alterar os documentos, da mesma maneira que os criamos. Isto é muito útil quando queremos fazer um processamento para vários arquivos. Como um exemplo, a companhia ACME deseja mudar seu logotipo da Figura 1 para a Figura 2.



Figura 1 – Logotipo Antigo



Figura 2 – Logotipo novo

“ (...) Como você pode ver, a manipulação de arquivos OpenXML é bastante simplificada com o uso da OpenXML SDK. Com ela, você não precisa abrir os arquivos zip e alterar diretamente os xmls que estão dentro deles. (...) ”

Esta empresa tem diversas apresentações onde está o logotipo antigo e quer atualizar para o novo logotipo. Isto pode ser feito usando OpenXml SDK: abrimos todas as apresentações, procuramos o logotipo no slide mestre e substituímos pelo novo, salvando a apresentação. Isto é feito da seguinte maneira:

```
public static void SubstituiLogo(string
    nomeArquivo)
{
    using (PresentationDocument doc =
        PresentationDocument.Open(nomeArquivo, true))
    {
        PresentationPart presentationPart =
            doc.PresentationPart;
        if (presentationPart != null &&
            presentationPart.Presentation != null)
        {
            var masterParts =
                presentationPart.SlideMasterParts;
            var imageParts =
                presentationPart.SlideParts;

            var images = masterParts.SelectMany(s
                => s.ImageParts)
                .Concat(masterParts.SelectMany(s
                    => s.SlideLayoutParts)
                    .SelectMany(s => s.ImageParts))
                .Concat(imageParts.SelectMany(s
                    => s.ImageParts))
                .Concat(imageParts.Select(s =>
                    s.SlideLayoutPart)
                    .SelectMany(s =>
```

A PROGRAMAR

CRIANDO ARQUIVOS OFFICE COM OPENXML SDK

```
        s.ImageParts));  
    foreach (var imagePart in images)  
        SubstituiLogoDaImagem(doc,  
                               imagePart);  
    }  
}
```

As imagens podem estar tanto nos slides como nos slides master. Além disso, elas podem também estar nos slides de layout. Esta linha

```
var images = masterParts.SelectMany(s =>  
    s.ImageParts)  
    .Concat(masterParts.SelectMany(s =>  
    s.SlideLayoutParts).SelectMany(s =>  
    s.ImageParts))  
    .Concat(imageParts.SelectMany(s =>  
    s.ImageParts)).Concat(imageParts.Select(s =>  
    s.SlideLayoutPart).SelectMany(s => s.ImageParts));
```

obtem todas as imagens, em qualquer uma destas partes da apresentação e permitem que elas sejam substituídas pelo novo logo, o que é feito com o seguinte código:

```
private static void SubstituiLogoDaImagem  
(PresentationDocument doc, ImagePart imagePart)  
{  
    if (imagePart.Uri.OriginalString.Contains  
        ("image1") ||  
        imagePart.Uri.OriginalString.Contains  
        ("image2"))  
    {  
        using (FileStream imgStream = new  
            FileStream("AcmeLogo.jpg",  
                FileMode.Open, FileAccess.Read))  
            imagePart.FeedData(imgStream);  
    }  
}
```

A imagePart é substituída com o método FeedData, passando-se o stream da imagem. Assim, a imagem é facilmente substituída em toda a apresentação.

O fato de termos um formato aberto para os arquivos Office traz muitas oportunidades de desenvolvimento

“ A partir do Office 2007, o formato de arquivos Office mudou, para um formato aberto, o OpenXML (<http://openxmldeveloper.org/>).

Conclusões

Como você pode ver, a manipulação de arquivos OpenXML é bastante simplificada com o uso da OpenXML SDK. Com ela, você não precisa abrir os arquivos zip e alterar diretamente os xmls que estão dentro deles.

(...) a manipulação de arquivos OpenXML é bastante simplificada com o uso da OpenXML SDK(...)

O fato de termos um formato aberto para os arquivos Office traz muitas oportunidades de desenvolvimento e permite que os usuários de nossos programas não precisem ter o Office instalado na máquina para usá-los.

AUTOR

Escrito por Bruno Sonnino

Criar uma API no Azure App Service

Introdução

No mês de Março, a Microsoft anunciou uma alteração à sua oferta de serviços introduzindo o novo Azure App Service. Este serviço agrega os “antigos” Websites, agora com o nome Web Apps, os Azure Mobile Services, agora Mobile Apps, e introduz novas capacidades como as Logic Apps e as API Apps. É sobre estas últimas que se foca este artigo.

A questão que se coloca é: Qual o motivo para criar uma API App e não uma Web API num Website? Esta questão pode ser facilmente respondida vendo a lista das funcionalidades disponibilizadas, como por exemplo:

- Controlo de acesso por recurso à Azure Active Directory ou serviços OAuth, como Facebook ou Twitter;
- Integração rápida com a Framework Swagger;
- Integração fácil com as Logic Apps;
- Galeria privada de APIs que permite disponibilizar as APIs a outros membros da organização;
- Geração automática de um SDK para a nossa API;

Neste artigo vamos ver como implementar uma API, publicá-la na nossa subscrição e criar um SDK para consumir a mesma.

Descrição

Existem inúmeros serviços, dentro das organizações, que são necessários ao funcionamento das diversas aplicações. Um bom exemplo é o envio de e-mails.

Imaginemos que pretendemos disponibilizar uma API, para que todas as aplicações da nossa organização possam facilmente enviar e-mails, sem a preocupação de conhecer a lógica de utilização do servidor de e-mail.

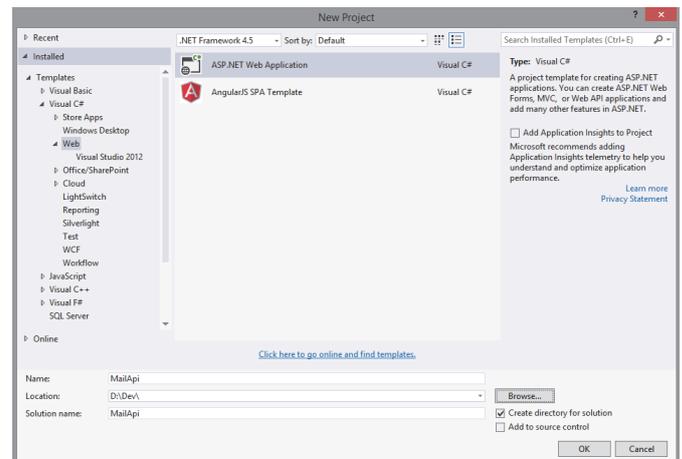
Antes de começar, temos que garantir que o [Microsoft Azure SDK for .NET 2.6](#), se encontra instalado.

Criar um projeto API App

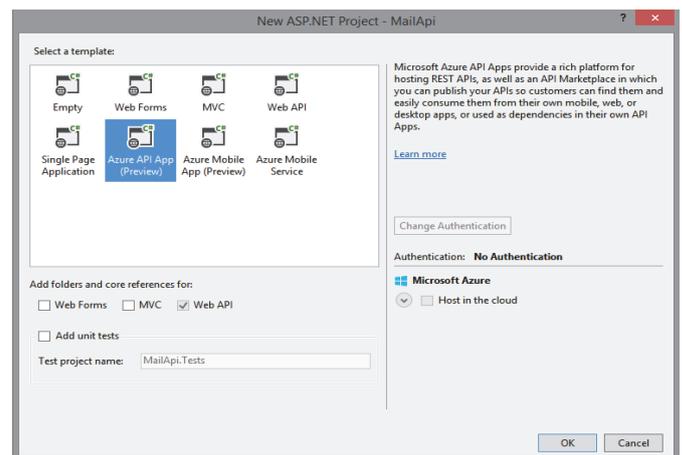
Vamos começar por ver como criar um projeto para a nossa API App.

1. Abrir o Visual Studio 2013;

2. Criar um novo projeto utilizando a função **File > New Project**;
3. Escolher a template **ASP.NET Web Application**;
4. Atribuir um nome ao projeto. Neste exemplo vamos utilizar MailApi;



5. Clicar em **OK**;
6. Na janela apresentada, seleccionar a template **Azure API App**;



7. Clicar em **OK** para criar o projeto;

Criar o serviço de envio de e-mail

Vamos ver como expor um método para enviar e-mails na nossa API. Para tal, vamos criar uma classe que representa o modelo dos dados a receber no pedido, e um Controller para expor o método de envio.

A PROGRAMAR

CRIAR UMA API NO AZURE APP SERVICE

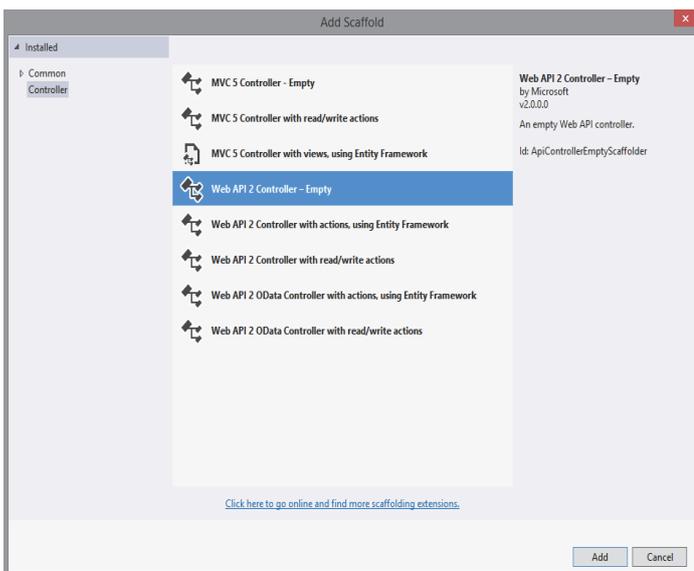
1. Na pasta Models da nossa solução, adicionar uma nova classe com o nome MailModel. Botão direito do rato na pasta Models e selecionar a opção **Add > Class**.
2. No ficheiro MailModel.cs criado, adicionar o seguinte código:

```
using System.ComponentModel.DataAnnotations;
public class MailModel
{
    [Required]
    [EmailAddress]
    public string To { get; set; }

    [Required]
    public string Subject { get; set; }

    [Required]
    public string Body { get; set; }
}
```

3. Adicionar um Controller ao projeto (na pasta Controllers) com o nome MailController, utilizando a template **Web API 2 Controller – Empty**;



4. No ficheiro criado MailController.cs adicionar o seguinte método;

```
using MailApi.Models;
[HttpPost]
public HttpResponseMessage Post([FromBody]MailModel
data)
{
    using (var client =
        new System.Net.Mail.SmtpClient())
    {
        // create the message
        var mailFrom = new System.Net.Mail.MailAddress
            ("noreply@deathstar.com", "Death Star");
        var mailTo = new System.Net.Mail.MailAddress
            (data.To);
        var mail = new System.Net.Mail.MailMessage
            (mailFrom, mailTo);

        mail.Subject = data.Subject;
        mail.Body = data.Body;
    }
}
```

```
mail.IsBodyHtml = true;

// send email
client.Send(mail);
}
return Request.CreateResponse
(HttpStatusCode.OK);
}
```

Configurar o serviço

Uma vez que neste exemplo vamos enviar o e-mail via SMTP, necessitamos de configurar os dados do servidor de e-mail.

1. Aceder ao ficheiro Web.config da nossa aplicação;
2. Adicionar a seguinte configuração (antes da linha):

```
<system.net>
<mailSettings>
  <smtp from="noreply@deathstar.com">
    <network host="mail.deathstar.com" port="587"
      userName="noreply@deathstar.com"
      password="#MayTheForceBeWithThisPassword!"
      enableSsl="true" />
  </smtp>
</mailSettings>
</system.net>
```

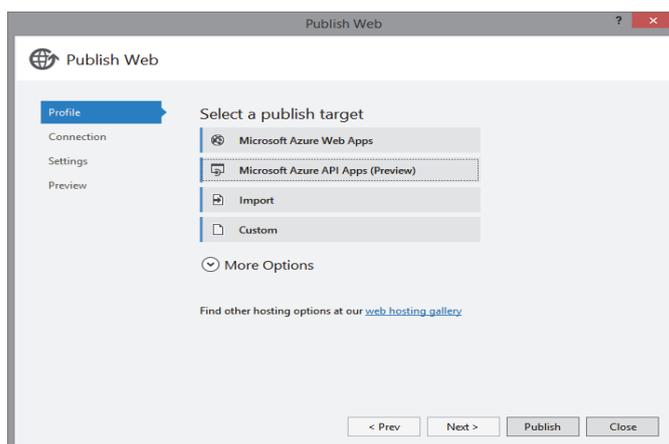
3. Preencher com os dados do nosso servidor.

Publicar a aplicação para o Azure App Service

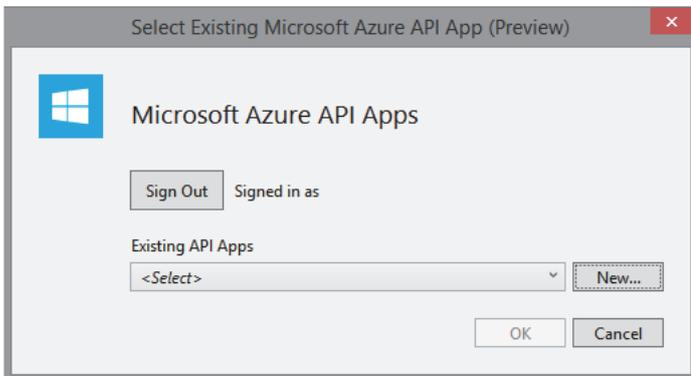
A publicação pode ser feita para um App Service criado anteriormente no Portal, mas aqui vamos ver como criar o mesmo sem sair do Visual Studio.

Para realizar este passo é necessário ter uma subscrição do Azure. Se não tens, podes criar uma subscrição de testes de um mês em <http://azure.microsoft.com/en-us/pricing/free-trial/>. Se já tens uma subscrição, segue os seguintes passos:

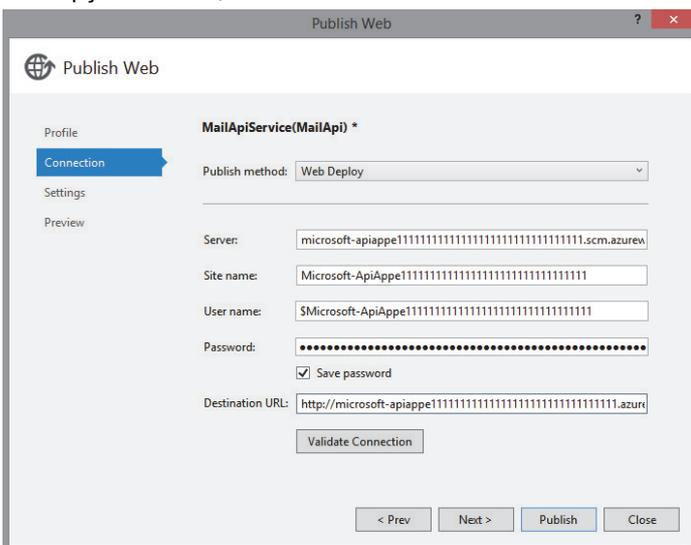
1. No Solution Explorer, clicar com o botão direito no projeto e clicar em **Publish**;
2. Na janela de publicação, aceder ao painel **Profile** e escolher a opção **Microsoft Azure API Apps**;



3. Clicar em **New** para criar uma nova API App na subscrição;



4. Na janela de criação da API App introduzir:
 - O nome da API App. Neste caso vamos utilizar MailApiService;
 - Selecionar a subscrição pretendida;
 - Escolher um nome para o App Service;
 - Escolher um nome para o Resource Group;
 - Escolher o nível de acesso Available to Anyone;
 - Escolher a região onde alojar o App Service.
5. Clicar em **OK** e aguardar uns minutos para até o serviço ser criado. O processo de criação do serviço pode ser acompanhado no painel Azure App Service Activity;
6. Uma vez criado o App Service, voltamos a executar o passo número 1, aceder à janela de publicação, e clicamos na opção **Publish**;



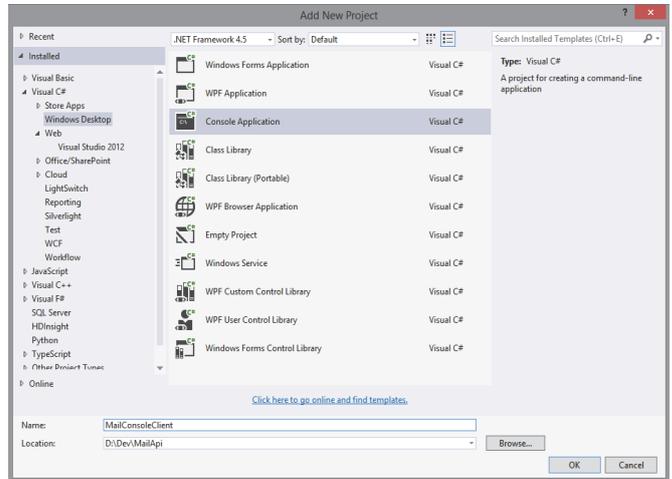
7. O serviço será publicado.

Consumir a API a partir de um cliente .NET

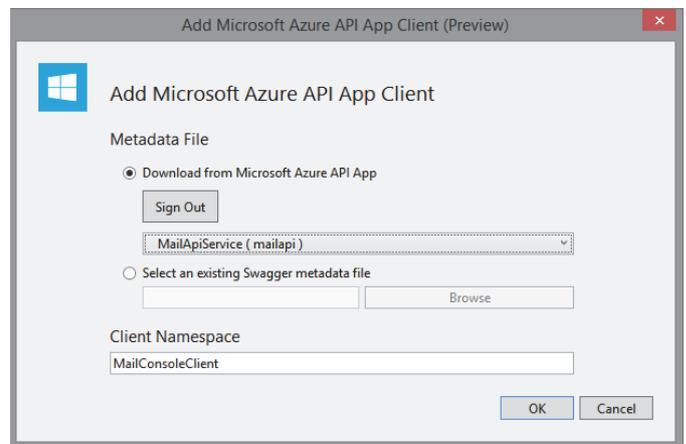
5. Agora vamos ver como criar um SDK para a nossa API

e utilizá-lo.

1. Adicionar uma Console Application à nossa solução;



2. No Solution Explorer, clicar com o botão direito do rato na Console Application e selecionar a opção **Add > Azure API App Client**;
3. Na janela apresentada, escolher a opção **Download from Microsoft Azure API App** e selecionar o MailA-



piService, criado anteriormente;

4. Clicar em **OK** e aguardar que o nosso SDK seja gerado;
5. Quando terminar, estamos prontos para invocar a API, como demonstra o exemplo que se segue.

```
class Program
{
    static void Main(string[] args)
    {
        var client = new MailApiService();
        var data = new Models.MailModel {
            To = "darth.vader@deathstar.com", Subject =
            "War Report", Body = "The Force awakens" };
        var response = client.Mail.Post(data);
        Console.WriteLine("Done!");
        Console.ReadKey();
    }
}
```

A PROGRAMAR

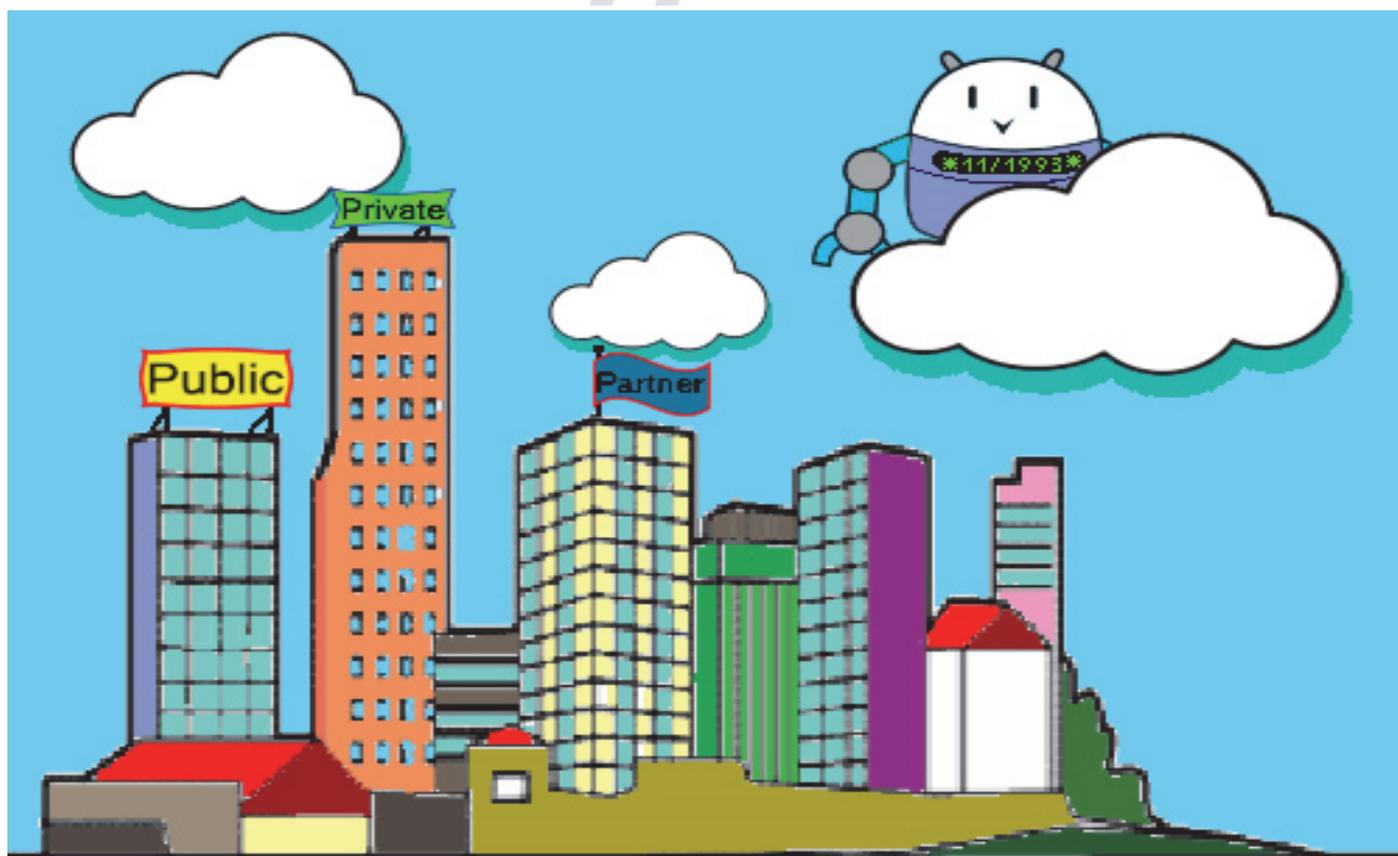
CRIAR UMA API NO AZURE APP SERVICE

“ **A lógica da galeria privada aliada à facilidade em consumir a API, permite evitar a replicação de funcionalidades, criando um ecossistema de APIs ao dispor de todos.** ”

Conclusão

Com este novo serviço disponibilizado pela Microsoft, podemos, de forma fácil e rápida, disponibilizar funcionalidades que anteriormente tinham um grande custo de desenvolvimento. A lógica da galeria privada aliada à facilidade em consumir a API, permite evitar a replicação de funcionalidades, criando um ecossistema de APIs ao dispor de todos.

“ **Qual o motivo para criar uma API App e não uma Web API num Website?** ”



AUTOR

Escrito por Guilherme Ferreira

O Guilherme Ferreira é um apaixonado pelo desenvolvimento de produtos, que ao longo dos anos, e muitos cafés depois, especializou-se no desenvolvimento de soluções para a cloud, com foco especial nas tecnologias Microsoft. Podes saber mais sobre ele em <http://gsferreira.com> ou no seu twitter [@gsferreira](https://twitter.com/gsferreira).

Primeiros passos no desenvolvimento de aplicações Android

“Aprenda os primeiros passos para iniciar o desenvolvimento de aplicações para a plataforma móvel líder do mercado. Neste artigo será abordada desde a instalação e configuração do ambiente de desenvolvimento, até a construção e a execução de um projeto em um dispositivo virtual. Por fim, exportaremos o .apk deste projeto para que possa ser executado em dispositivos reais com o Android instalado.”

Atualmente, Android ocupa uma boa parcela entre os sistemas operacionais móveis disponíveis no mercado, sendo que, segundo a Google [Aceda a : <http://goo.gl/9U6vrF>], este sistema ultrapassa atualmente a marca de 1 BILHÃO de dispositivos. Com isso podemos constatar o grandioso mercado de oportunidades que o sistema oferece para empresas e programadores autônomos de aplicações móveis.

A Google, empresa que possui os direitos sobre o sistema atualmente, fornece excelente documentação e ferramentas gratuitas para o desenvolvimento de aplicações (apps). Inicialmente, o Android era um sistema planeado para funcionar em câmeras digitais, porém, com o passar do tempo, veio a operar em telemóveis, conhecidos como smartphones, e atualmente funciona em uma série de dispositivos, entre eles: Android Wear (Smart Watch ou Relógio Inteligente), Smartphones, Tablets, Android TV e o mais novo Android Auto. Alguns aparelhos televisores, os Smart TVs, possuem o Android instalado nativamente, porém é possível adquirir o Chromecast para televisores que não possuem o sistema (conheça o Chromecast em: <http://goo.gl/C3TmBz>).

Em primeiro lugar, vamos aprender onde encontrar algumas ferramentas e recursos necessários para iniciar o desenvolvimento de aplicações para Android, abordando a instalação e configuração do SDK com o ambiente de desenvolvimento Android Studio, recentemente adotado como ambiente de desenvolvimento padrão do sistema pelo Google. Por fim, aprenderemos a instalar e configurar um dispositivo virtual utilizando o Genymotion, sendo que através deste dispositivo virtual poderemos executar e testar aplicações. Por fim, construiremos um projeto teste e executaremos no nosso dispositivo virtual e em um dispositivo real, exportando o .apk. Todos os recursos necessários para iniciar o desenvolvimento para esta plataforma são encontrados gratuitamente na Internet.

Por onde começar?

O sistema operacional Android é mantido pelo Google, o qual fornece uma excelente documentação gratuita que pode ser acessada em: <http://www.android.com/>. Neste site existem informações variadas para diferentes pessoas, desde simples utilizadores aos programadores mais experientes. Para iniciar o desenvolvimento de aplicações vamos procurar por Android SDK (Standard Development Kit ou Kit de Desenvolvimento

Padrão), que contém todos os recursos e plug-ins necessários para programar as aplicações para o sistema. Atualmente o SDK vem em conjunto com o IDE Android Studio, o qual pode ser descarregado em: <https://developer.android.com/sdk/index.html>.

Faça download do Android Studio no link acima e certifique-se que seu computador tenha pelo menos o Java Development Kit 7 (JDK) instalado. Caso não tenha, pode descarregar gratuitamente em: <http://goo.gl/wjP3Sn>.

Por padrão, o Android Studio virá configurado para o desenvolvimento na última versão do sistema disponível, que na data deste artigo (Abril de 2015) se refere à versão 5.0, conhecida também como Lollipop. Entretanto, é possível descarregar SDK para desenvolver apps para versões ou APIs anteriores do sistema através do SDK Manager, o qual abordarei posteriormente neste artigo. É possível desenvolver apps para Android em diversas plataformas, porém neste artigo a instalação dos recursos será toda feita no Windows, sendo que alguns passos explicados poderão mudar conforme a plataforma de desenvolvimento escolhida. Acredita-seno entanto, que as variações são pequenas e caso esteja a utilizar outra plataforma, também conseguirá acompanhar este artigo.

Java

Conseguiu fazer o download dos arquivos citados anteriormente? Então, vamos começar! A instalação do Java é fácil e está bem documentada online. Caso o Java não funcione no Windows, é bastante comum que o problema seja a configuração das variáveis de ambiente do sistema, sendo necessário efetuar esta configuração em alguns casos.

Android Studio

Após instalar o Java, abra o ficheiro da instalação do Android Studio. Após isso, você verá o ecrã de boas-vindas do IDE, conforme a Figura 1. Clique em Next.

Coloque aqui uma referência para a página de onde veio essa informação

Acho melhor “o” IDE, pois IDE significa “Ambiente de Desenvolvimento Integrado”.



Figura 1 – Instalação do Android Studio

A PROGRAMAR

PRIMEIROS PASSOS NO DESENVOLVIMENTO DE APLICAÇÕES ANDROID

No segundo ecrã, deverá marcar as opções que quer instalar na máquina. Para realmente podermos programar para a plataforma Android, é necessário ter instalado o Java e o Android SDK, ou seja, obrigatoriamente a opção Android SDK deverá estar marcada entre as opções de instalação. A instalação é obrigatória, pois ela contém todos os ficheiros necessários para o desenvolvimento, sendo que, caso já utilize outra IDE diferente do Android Studio para programar e tenha o Android SDK instalado, poderá desmarcar esta opção. Caso seja a primeira instalação destes ambientes na máquina, é recomendável que deixe todas as opções marcadas, desta forma garantimos que serão instalados todos os recursos para desenvolvimento e para simular a máquina virtual do Android para testar as aplicações.

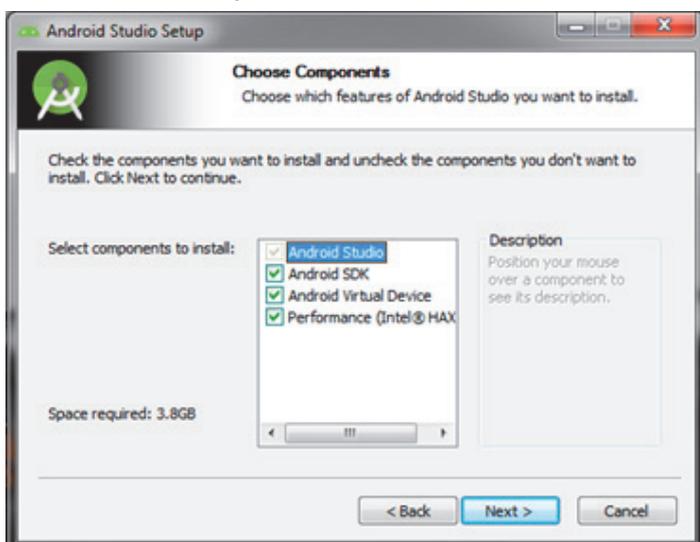


Figura 1.2

Aceite os termos de uso e clique em Next. Após isto, selecione o diretório que queira instalar o Android Studio e o Android SDK. Recomendo que deixe os diretórios pré-selecionados pelo IDE.

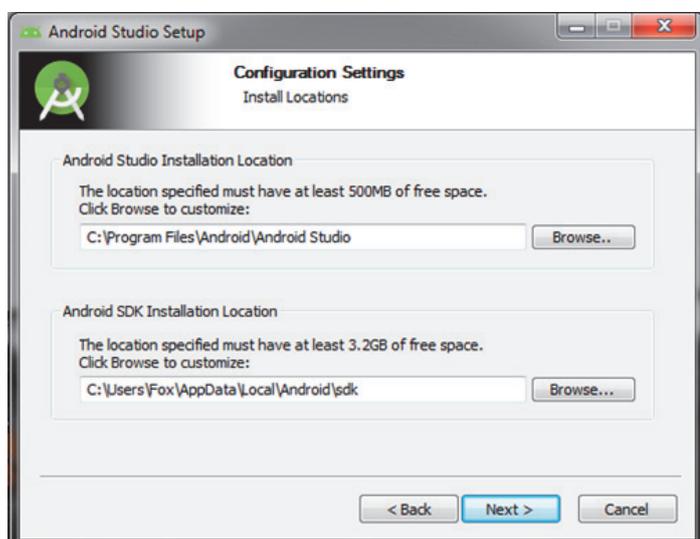


Figura 1.3 – Escolher os diretórios para instalação

Após isto, será copiado todos os ficheiros do Android Studio e do SDK para a máquina nos diretórios informados no passo anterior. Este processo dependendo da configuração da máquina pode demorar.



Figura 1.4 – Instalação concluída

Se seguiu todos os passos citados acima, instalou o Android Studio e já está com o ambiente todo configurado para iniciar o desenvolvimento de aplicações para a plataforma mobile do Google.

OBS: O Android Studio atualiza-se frequentemente, e caso tenha algum passo diferente do que foi citado neste artigo, mantenha as configurações padrões e clique em Next.

Criar primeiro projeto

Depois de instalar o Android Studio, abra-o e vamos em File > New Project. Devemos nos deparar com um ecrã conforme a Figura 2.1.

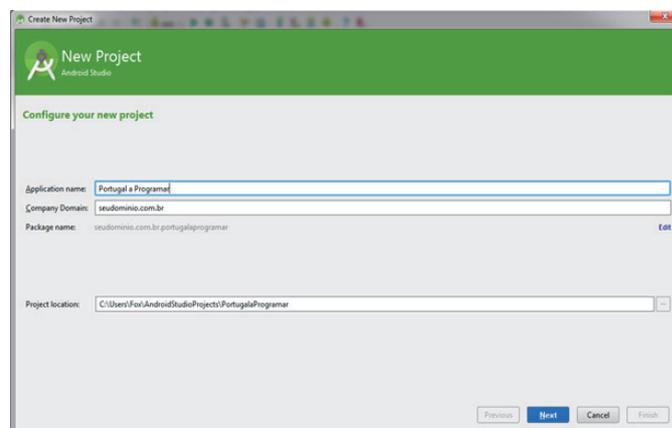


Figura 2.1 – Criar um novo projeto

Este é o primeiro passo para criar o primeiro app. Em Application Name deve colocar o nome da aplicação, este nome é o que aparecerá para o utilizador na lista de aplicações no smartphone. Em Company Domain deve colocar o domínio da empresa, caso não possua é

A PROGRAMAR

PRIMEIROS PASSOS NO DESENVOLVIMENTO DE APLICAÇÕES ANDROID

extremamente recomendável que registre um domínio no momento da distribuição da aplicação nas lojas online como o Google Play, pois cada aplicação Android deve ter um domínio único, o que identificará a aplicação entre as milhares de outras disponíveis. O Package Name geralmente é criado com base no domínio da aplicação, para este exemplo, deixe o package name conforme foi criado pela IDE.

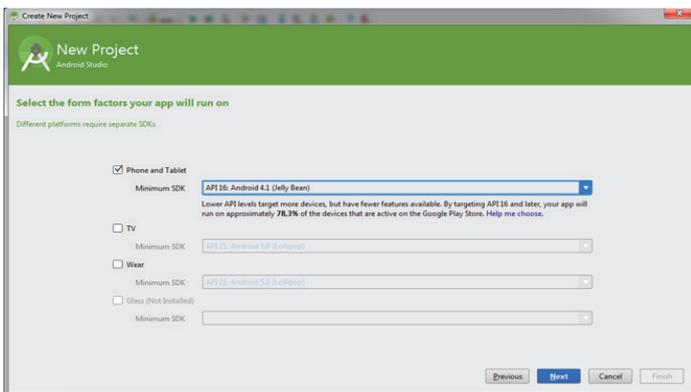


Figura 2.2 – Selecionar as API's de desenvolvimento

Após isso, deve especificar quais são os dispositivos que irão executar a aplicação, podendo ter como alvo os smartphones e tablets (a diferença de uma aplicação de smartphone e tablet geralmente está no tamanho do ecrã), Android TV, Android Wear e o Google Glass. Para esta aplicação iremos selecionar somente a opção Phone and Tablet, e selecionar uma API alvo para desenvolvimento. Cada versão do Android possui um número de API, sendo que este número é utilizado pelos programadores para especificar quais versões do Android são capazes de executar a aplicação. Para este exemplo selecionei a API 16, que corresponde a versão 4.1 do Android, conhecida também como Jelly Bean (note que os nomes das versões do Android são também nomes de sobremesas? Saiba mais sobre isso em: <http://www.android.com/history/>), que apesar de não ser atual, ainda corresponde a 78.3% dos dispositivos na Google Play Store, atingindo mais da metade dos dispositivos com Android em funcionamento no mercado, além de ser a versão do Android que instalarei ao configurar um dispositivo virtual no Genymotion.

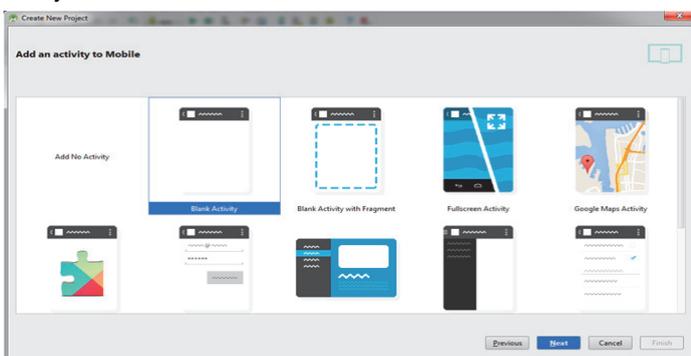


Figura 2.3 – Templates de Activities

O Google adotou o IDE Android Studio como padrão para desenvolvimento pela produtividade que os

programadores conseguem ao utilizar a mesma. Um exemplo desta produtividade são os templates de Activities que oferece no momento de criar uma aplicação. Activities são um dos principais recursos existentes em Android, sendo basicamente um fluxo de execução que será mostrado ao utilizador, ou seja, ao utilizar as aplicações, cada ecrã do seu smartphone corresponde a uma Activity. Elas são fundamentais, pois sem elas não seria possível interagir com o utilizador. Para esta primeira aplicação vamos criar uma Blank Activity, onde será criada uma atividade em branco, pronta para programarmos as nossas aplicações.

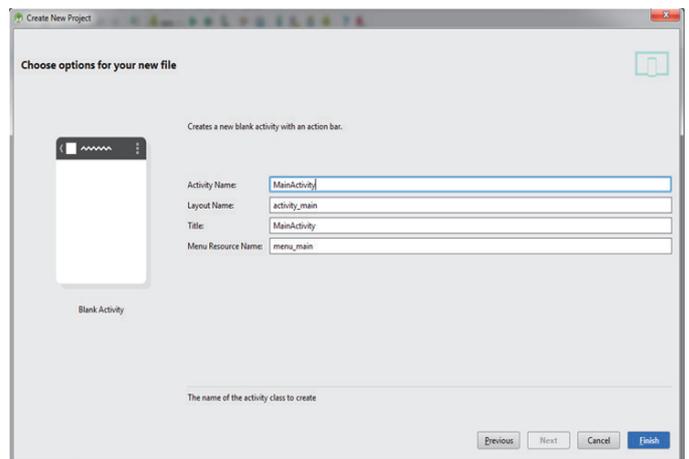
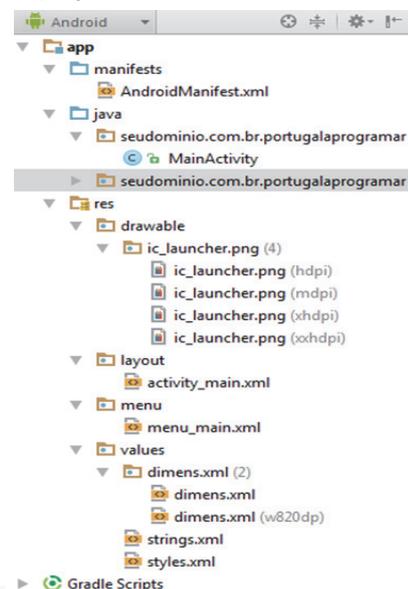


Figura 2.4 – Criar uma Activity

No ecrã da Figura 2.4, vamos colocar o nome da Activity e o layout_name. No Android, toda a lógica das aplicações são feitas em Java e toda a parte gráfica dos ecrãs dos dispositivos em xml. Por padrão, deixo sempre a primeira Activity conforme o ecrã acima. Após isso, criamos um projeto Android com todos os ficheiros e recursos necessários para iniciar estudos no desenvolvimento de aplicações para esta plataforma.

Diretórios do projeto Android



A PROGRAMAR

PRIMEIROS PASSOS NO DESENVOLVIMENTO DE APLICAÇÕES ANDROID

Ao criar o projeto conforme descrito anteriormente, deparamo-nos com a seguinte estrutura de diretórios. Irei explicar brevemente o que cada diretório e seus ficheiros representam.

Manifesto

O ficheiro `AndroidManifest.xml` é um dos ficheiros mais importantes, pois todas as permissões da aplicação, todas as atividades e recursos que irá utilizar deverão estar declarados neste ficheiro. Caso a aplicação utilize algum recurso que não esteja declarado neste ficheiro, este recurso simplesmente não irá funcionar ao distribuir o `app`.

Java

Os ficheiros Java são responsáveis pela lógica e controle da sua aplicação, sendo que os ficheiros em `xml` serão responsáveis pela parte de visualização. Toda a lógica e regras deverão estar implementados nestes ficheiros em Java. Sinta-se livre para criar outros pacotes e administrar o seu projeto em Java da forma que achar mais conveniente.

res

A pasta `res` vem da palavra *resources* que significa recursos em português. Este diretório deverá conter todos os recursos gráficos que a aplicação irá utilizar, como imagens e ficheiros `xml`. Basicamente, as imagens deverão estar no diretório `drawable` e os demais ficheiros `xml` em seus respectivos diretórios, como `layout` (especifica o layout de todas as *activities* da aplicação. É possível criar layouts diversos para diferentes tamanhos e tipos de ecrã. Saiba mais em: <http://goo.gl/FoQCCM>), `menu` (especifica como serão todos os menus da aplicação) e em `values`. Para quem está a começar, o ficheiro mais importante do diretório `values` é o `strings.xml`. É fortemente recomendável que todas as Strings que apareçam na aplicação, principalmente as strings do layout sejam referenciadas por este ficheiro, pois isto permite a fácil internacionalização das aplicações. Assim, se quisermos criar um ficheiro para cada linguagem, basta traduzimos estas strings e o próprio Android irá se encarregar de carregar o ficheiro necessário com base na linguagem utilizada pelo sistema. Caso não exista um ficheiro `strings.xml` para a língua utilizada no sistema, será carregado um ficheiro padrão, que também poderá ser definido pelo programador.

Basicamente esta é a estrutura de diretórios de um projeto Android. É claro que se aprofundarmos em cada um destes ficheiros iremos descobrir diversas funcionalidades possíveis. No entanto, isto é o básico que devemos entender para iniciar o desenvolvimento das nossas aplicações.

Instalação e configuração do Genymotion

O sistema operativo Android é executado em diversos tipos de *smartphones*, *tablets*, *wearables*, ou seja, o sistema é executado em aparelhos com diversos tamanhos de ecrã especificações de hardware diferentes, funcionalidades diferentes entre estes hardwares, o que, por consequência,

acaba criando uma série de problemas para os programadores de aplicações no momento da distribuição do `app`. Baseado nisso, programadores fundaram uma *startup* e criaram a ferramenta Genymotion, que nos permite descarregar, configurar e simular diversos tipos de dispositivos virtuais Android, podendo descarregar um simulador específico para determinado *smartphone*. Desta forma, conseguimos simular as nossas aplicações com qualidade e eficiência em diversos tipos de hardware que executam o Android

Para descarregar o Genymotion, basta aceder a: <https://www.genymotion.com/> e fazer um registo no site para a versão pessoal que é disponibilizada gratuitamente. O Genymotion utiliza o Oracle VirtualBox para criar os dispositivos virtuais. Sendo assim, existem duas opções de download da ferramenta, incluindo ou não o VirtualBox no download. Caso você já possua o Oracle VirtualBox instalado na sua máquina, poderá descarregar do Genymotion sem ele.

Depois de instalar e abrir o Genymotion, devemos nos deparar com o ecrã mostrado na Figura 3.1.

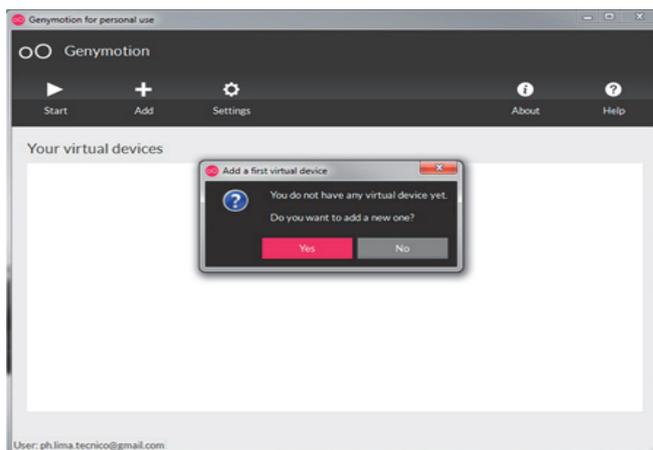


Figura 3.1 – Criar o dispositivo virtual

Vamos clicar em `Yes` para configurar o dispositivo virtual. Ao clicar em `Yes` deparamo-nos com o ecrã da Figura 3.2.



Figura 3.2 – Configurar o dispositivo virtual Android

No ecrã da Figura 3.2, podemos procurar por dispositivos virtuais para configurarmos a máquina filtrando as pesquisas pela versão do Android ou pelo modelo do dispositivo. Genymotion é bastante interessante, pois permite fazer o download de configurações de um dispositivo real, ou seja, simularmos um dispositivo virtual com todas as configurações e especificações do dispositivo real, o que é bastante útil em diversos casos. Para este artigo escolhi o modelo Nexus S, que não é um dispositivo recente, mas que executa de forma estável em máquinas com configurações médias/baixas e possui a versão do Android 4.1.1, que suporta a maioria das aplicações disponíveis no Google Play atualmente. Fique à vontade para escolher outra versão do Android ou outro modelo de dispositivo virtual para instalar na máquina. Após escolher a versão a ser instalada clique em *Next*.

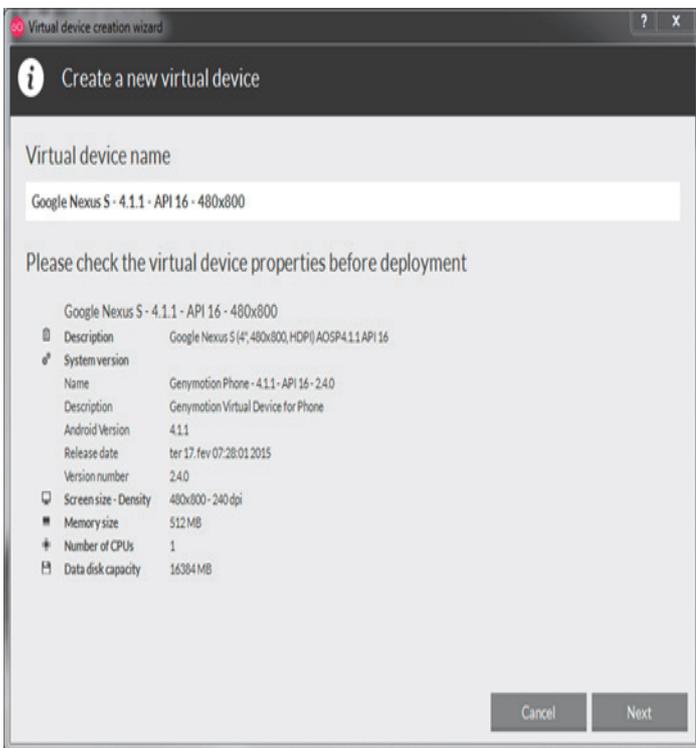


Figura 3.3 – Confirmar as configurações

Após isto, serão informadas as especificações do dispositivo escolhido, como o nome do dispositivo, resolução da ecrã, versão do Android, quantidade de memória RAM utilizada e capacidade de memória de armazenamento em disco. Verifique se a máquina possui recursos suficientes para executar a versão escolhida, pois há versões que exigem muita memória ou muito processamento, podendo deixar o sistema lento enquanto utiliza o dispositivo virtual. Caso tenha certeza da versão a ser instalada clique em *Next* e espere até que o dispositivo virtual seja descarregado e instalado na sua máquina, que na verdade será instalado para funcionar no Oracle VirtualBox. Quando a instalação terminar o dispositivo virtual será carregado e caso tenha instalado o Nexus S, o ecrã deverá ser semelhante a Figura 3.4.

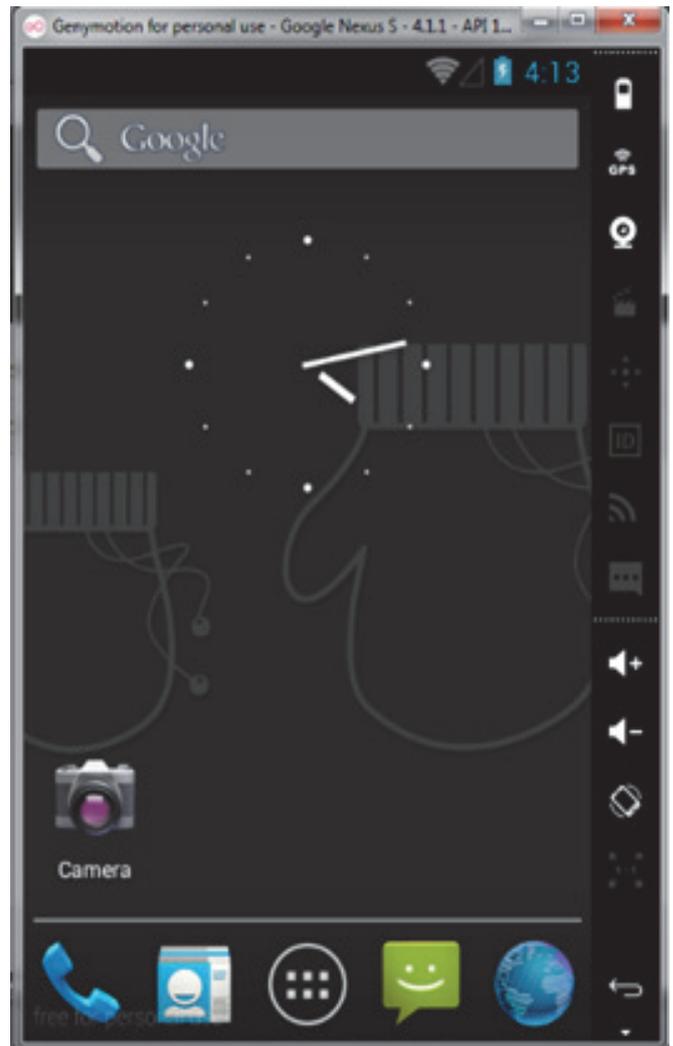


Figura 3.4 – Simulação do dispositivo virtual

Após instalar o Genymotion e configurar um dispositivo virtual, devemos integrar a ferramenta com a IDE Android Studio, afim de facilitar o trabalho no momento de simularmos um *app* neste dispositivos virtuais. Para isso, vamos abrir o IDE Android Studio e aceder ao *menu File > Settings* ou então, como atalho, pressionar *Ctrl + Alt + S*. Ao abrir o ecrã de Configurações (*Settings*) da IDE, vamos no separador do lado esquerdo e selecionamos a opção *Plugins*, após isso, clicamos em *Browse Repositories*, no canto inferior central do ecrã. No próximo ecrã vá ao campo de pesquisa e escreva *Genymotion*, sendo que deverá aparecer, no separador do lado esquerdo, uma opção de *plugin*. Após isso, selecionamos esta opção e clicamos no botão *Install Plugin* em verde no lado direito do ecrã. Desta forma o *plugin* do Genymotion será descarregado e instalado automaticamente na IDE Android Studio. Para ter certeza que o *plugin* foi instalado, basta procurar pelo ícone abaixo na barra de ferramentas da IDE. Na próxima seção aprenderemos sobre como executar aplicações utilizando este *plugin*, que redirecionará a aplicação para os dispositivos virtuais do Genymotion.

A PROGRAMAR

PRIMEIROS PASSOS NO DESENVOLVIMENTO DE APLICAÇÕES ANDROID



Plugin do Genymotion no Android Studio

Testar a aplicação em dispositivo virtual

Em primeiro lugar, devemos iniciar o dispositivo virtual. Para isso, vamos clicar no plugin do Genymotion na IDE Android Studio e esperar até abrir o ecrã da Figura 4.1:

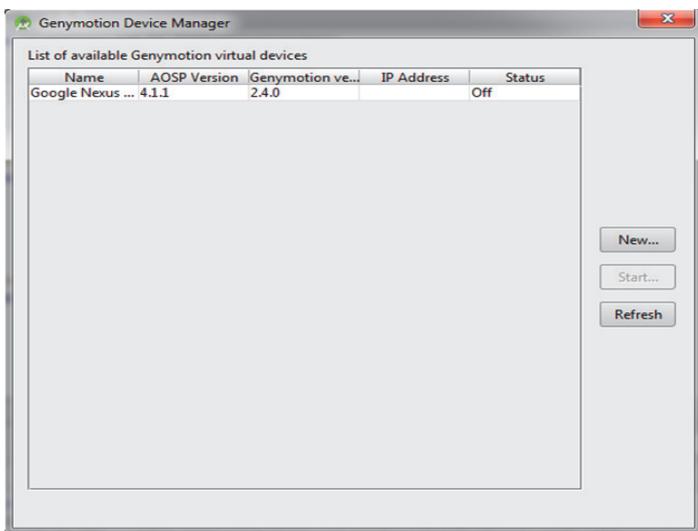


Figura 4.1 – Iniciar o dispositivo virtual do Genymotion

No ecrã da Figura 4.1 serão listados todos os dispositivos virtuais instalados na máquina, então basta escolher algum destes dispositivos e clicar em Start para iniciar. Caso não apareça nenhum dispositivo, abra o Genymotion, descarregue e configure um dispositivo virtual conforme explicado anteriormente. Após isso, no menu Run > Run, com isso deverá aparecer o ecrã conforme a Figura 4.2.

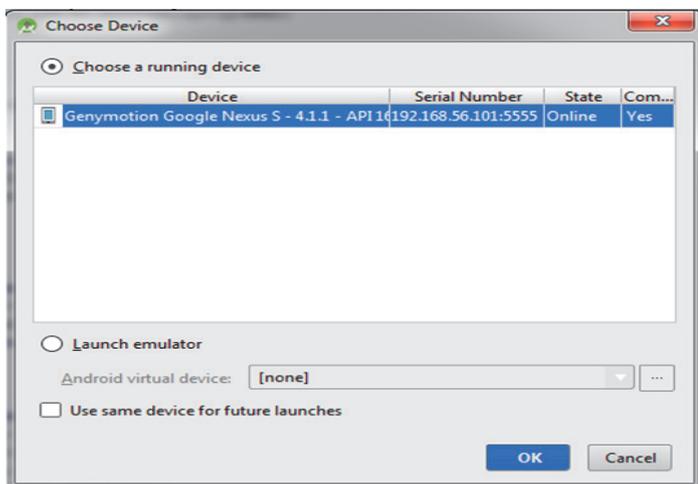


Figura 4.2 – Selecionar um dispositivo virtual em execução

No ecrã da Figura 4.2 serão listados todos os dispositivos virtuais em execução que estão prontos para executar a aplicação. Então, como já havíamos iniciado um dispositivo virtual no Genymotion, deverá aparecer pelo menos uma opção nesta lista. Vamos selecioná-la e clicar em Ok. Feito isto, a aplicação será carregada para o dispositivo virtual, o que deverá ser semelhante à Figura 4.3.

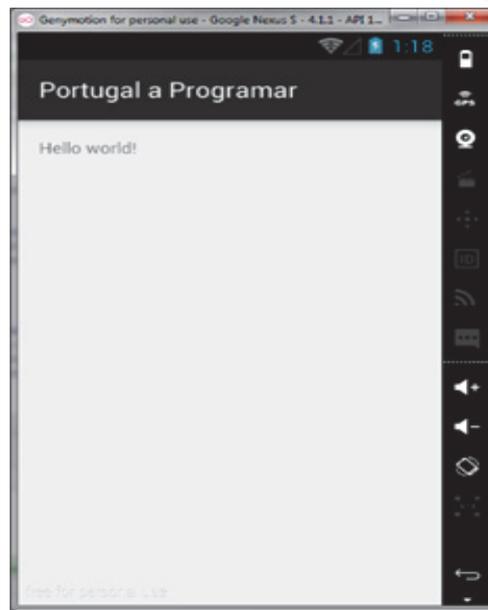


Figura 4.3 – Executara aplicação no dispositivo virtual

Exportar a aplicação para dispositivos reais

É extremamente importante que faça testes em dispositivos reais durante o desenvolvimento de seus projetos. Existem várias formas de exportar aplicações para serem testadas em dispositivos reais, sendo uma dessas formas a criação do **.apk**, que nada mais é do que a versão final da nossa aplicação. Para exportar um **.apk** no Android Studio é bastante simples. No menu superior da IDE vamos em **Build > Generate Signed APK**, conforme a Figura 5.1.

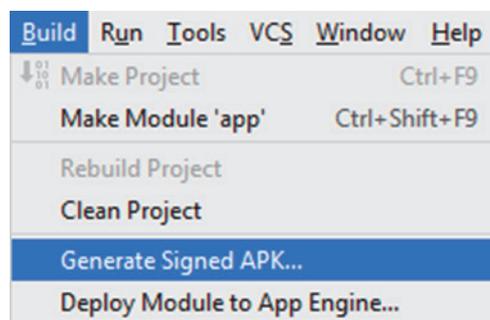


Figura 5.1 – Criação do APK

O apk deve ser criado com base em uma chave do programador, ou seja, na primeira vez que exportar a aplicação devemos registar ou abrir o arquivo que identifique a nossa chave. Isto é necessário, pois com esta chave será

informado quem é o programador ou empresa detentora da aplicação, localidade, e é também uma forma de segurança para que outros programadores não lancem outras aplicações fazendo se passar por determinada pessoa. Grave o ficheiro .jks em um diretório de fácil acesso, pois todas as vezes que forem exportar as aplicações será necessário informar o diretório deste ficheiro.

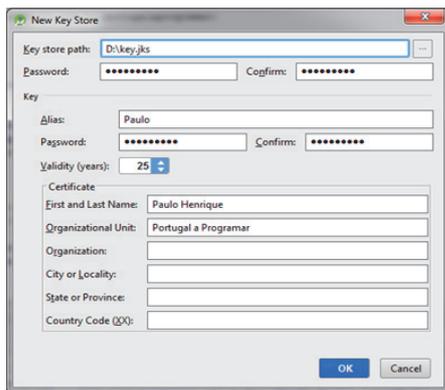


Figura 5.2 – Criar uma Key Store

Após criar a chave, será solicitado qual pasta será o destino do .apk e o tipo de versão que esta aplicação é: *release* (versão final para o utilizador) ou *debug* (versão de testes). Clique em *Finish* e será informado com uma mensagem a avisar que o .apk foi criado com sucesso.

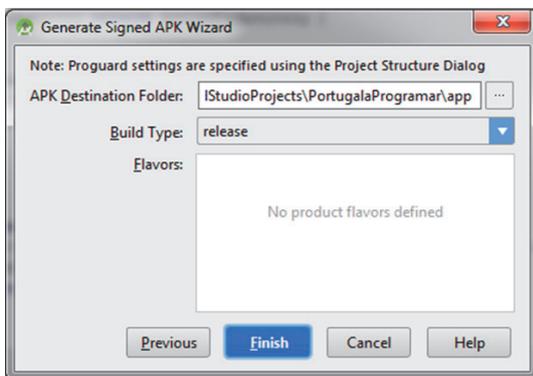


Figura 5.3 – Exportar o APK

Caso tenha conseguido chegar aqui , basta copiar o

ficheiro .apk e colocar na memória interna do dispositivo que queira testar. Antes de abrir o .apk no seu dispositivo, vá as **Configurações > Segurança** e certifique-se que a opção **Fontes desconhecidas** esteja acionada , caso contrário o sistema irá bloquear a instalação do .apk. Use um explorador de arquivos no dispositivo Android (recomendo o **ES File Explorer**) e selecione este ficheiro. Será aberto um instalador para a aplicação, basta seguir com os procedimentos e pronto, a [aplicação esta instalada em um dispositivo real](#).

Como continuar?

Existem vários materiais de estudo disponibilizados na Internet de forma gratuita sobre o desenvolvimento de aplicações Android com Android Studio, e demais assuntos relacionados a esta plataforma. Caso tenha bom domínio de inglês, é extremamente recomendável estudar no site developer.android.com, pois este é o único material que está sempre 100% atualizado e contém uma parte de *Training* muito interessante feita pela equipa de programadores da Google. Recomendo também ver o curso de Android na plataforma Udacity, que também foi feito por programadores da Google, onde é construído uma aplicação completa utilizando o Android Studio. Este curso é disponibilizado de forma gratuita, mas caso precise do certificado é necessário desembolsar uma certa quantia para recebê-lo.

Caso queira mais, procure como os programadores resolvem problemas comuns na Internet. Existem vários vídeos no **YouTube** e várias soluções interessantes em sites como o **Stack Overflow** (<http://pt.stackoverflow.com/>) , no qual pode publicar ou solucionar dúvidas de outras pessoas. Desejo a todos, bons estudos e uma boa caminhada por este fantástico mundo do desenvolvimento mobile.

AUTOR

Escrito por Paulo Afonso Parreira Júnior



Atualmente professor do curso de Bacharelado em Ciência da Computação da Universidade Federal de Goiás (Regional Jataí). É aluno de doutoramento do Programa de Pós-Graduação em Ciências da Computação (PPG-CC) da Universidade Federal de São Carlos (UFSCar), na área de Engenharia de Software. É mestre em Engenharia de Software pelo Departamento de Computação da UFSCar (2011). É integrante do Advanced Research Group on Software Engineering (AdvanSE) do

Escrito por Paulo Henrique Lima Oliveira



Discente do curso de Bacharelado em Ciência da Computação da Universidade Federal de Goiás (Regional Jataí). Técnico em Informática pelo Instituto Federal de Educação, Ciência e Tecnologia de Goiás – Campus Jataí. Integrante do grupo da Maratona de Programação do curso de Ciência da Computação da Universidade Federal de Goiás. Tem experiência em lin-

A PROGRAMAR

Introdução a arquitetura SAP com foco em SAP-1

De acordo com Malvino (1985, p. 255) o computador SAP (Simple-Quanto-Possível = *Simple-As-Possible*) foi projetado com o objetivo de apresentar de forma lúdica a estrutura operacional de um computador eletrônico moderno, de maneira que fique mais simples abstrair a concepção usada nos computadores eletrônicos mais modernos. Por mais estranho que possa parecer, muitos professores que atuam sobre a temática da disciplina relacionada a arquitetura e organização de computadores não fazem uso deste recurso ocasionando verdadeiro martírio para os alunos.

O uso do computador SAP pode facilitar o entendimento dos detalhes relacionados a organização de computadores e sua aplicação em linguagem de baixo nível.

Este artigo descreve brevemente a arquitetura SAP-1 e mostra duas aplicações exemplificando e utilizando-se sua linguagem de baixo nível.

A ARQUITETURA

A proposta da arquitetura do computador SAP é mostrada em três diferentes gerações, sendo a geração SAP-1 o primeiro estágio com vistas aos computadores eletrônicos modernos, foco da apresentação desta introdução. Assim sendo, não são apresentados elementos das gerações SAP-2 e SAP-3.

Objetiva-se com este artigo instigar professores da disciplina a fazerem da ferramenta SAP. Encontra-se no sítio Youtube diversas aulas disponibilizadas sobre o tema que podem ser utilizadas como elementos inspiradores e de embasamento para a apresentação de uma arquitetura real.

A Figura 1 apresenta a estrutura da arquitetura proposta para o SAP-1.

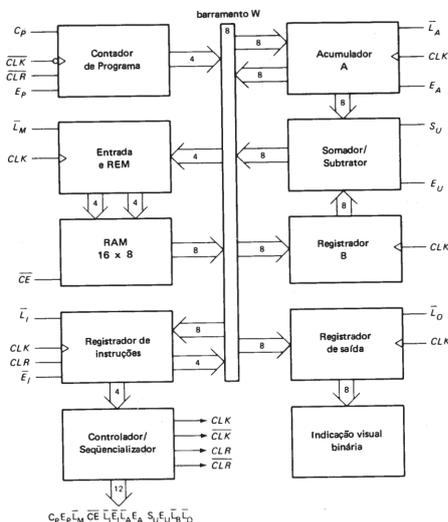


Figura 1 – Arquitetura SAP-1, segundo Malvino (1985, p. 257)

A arquitetura do modelo SAP-1 caracteriza-se por ser um computador organizado em barramentos, tendo como estrutura central de comunicação o **barramento W** de 8 bits, responsável pelo nível de compartilhamento de todos os dispositivos a ele conectado.

O dispositivo **Contador de Programa** faz parte da unidade de controle, possui como finalidade operacional a capacidade de efetuar a contagem de 0000 até 1111. Este dispositivo tem por finalidade enviar à memória o endereço da próxima instrução de programa a ser executada. A cada novo processamento executado o contador de programa é inicializado com valor 0000. Segundo (MALVINO, 1985, p. 256):

“Quando começa o processamento ou execução no computador, o contador de programa envia o endereço 0000 à memória. O contador de programa é então incrementado para se obter 0001. Depois da primeira instrução ser buscada e executada, o contador de programa envia o endereço 0001 à memória. Novamente o contador de programa é incrementado. Depois de a segunda instrução ser buscada e executada, o contador de programa envia o endereço 0010 à memória. Desta maneira, o contador de programa está acompanhando o desenvolvimento da próxima instrução a ser buscada e executada”.

O dispositivo **Entrada e REM** inclui o *registrador de dados* e o *registrador de endereços*. Este dispositivo é responsável pela efetivação das ações de entrada de dados e execução das instruções de um programa. Este dispositivo também é usado para armazenar o valor passado pelo *contador de programa*. Os registradores deste dispositivo enviam 4 bits de endereço e 8 bits de dados à RAM pelo REM (Registrador de Endereço de Memória).

O dispositivo **RAM** tem por finalidade armazenar os dados e as instruções enviadas pelo programa que serão usados no computador. Este dispositivo se caracteriza por ser uma memória estática com a capacidade de armazenar 16 linhas de 8 bits cada uma. Segundo (MALVINO, 1985, p. 256):

“Durante um processamento do computador, a RAM recebe endereços de 4 bits do REM e é executada uma operação de leitura. Desta maneira, a instrução ou palavra de dados armazenada na RAM é colocada no barramento W para uso em alguma outra parte do computador”.

O dispositivo **Registrador de Instruções** é parte da unidade de controle, tendo por finalidade receber um *byte* referente a uma instrução lida a partir da memória RAM. O *byte* recebido é dividido em dois *nibbles*, o *nibble superior*

A PROGRAMAR

INTODUÇÃO A ARQUITETURA SAP COM FOCO EM SAP-1

referente à própria instrução e passado ao **Controlador/Sequencializador** (descrito a seguir) e o *nibble inferior* referente a uma saída lida no barramento W quando necessário. Segundo (MALVINO, 1985, p. 256) o registrador de instruções efetua uma operação de leitura da memória.

O dispositivo **Controlador/Sequencializador** é parte da unidade de controle, tendo por finalidade controlar todo o computador. Este dispositivo recebe o *nibble* referente à instrução a ser executada, a qual, se encontra no *Registrador de Instrução*, decodifica e envia uma palavra de 12 bits para o computador (CAMPOS, 2011, p. 155). Segundo Campos (2011, p. 155) a “palavra de controle é responsável por habilitar ou desabilitar cada funcionalidade de cada módulo presente no computador a fim de que ele execute uma determinada tarefa”.

O dispositivo **Acumulador A** caracteriza-se por ser um registrador de memória temporária que tem por finalidade armazenar operandos e resultados de processamento realizados pelo computador (MALVINO, 1985, p. 258 & CAMPOS, 2011, p. 155). Este registrador recebe um dado transferido pelo barramento W, podendo disponibilizar este dado tanto para o dispositivo **Somador/Subtrador** como de volta para o barramento W.

O dispositivo **Somador/Subtrador** é o componente responsável pela efetivação do processamento das operações de soma e de subtração, sendo esta parte da Unidade Lógica e Aritmética do computador SAP-1. A subtração é efetuada a partir do uso do método do complemento de 2, onde para efetivar a subtração efetua uma soma do valor do acumulador com o complemento do segundo operando (CAMPOS, 2011, p. 155). Para este dispositivo realizar sua tarefa ele pega o operando (valor) do **Registrador B** e realiza a operação indicada (soma ou subtração) junto ao operando do **Acumulador A** e grava o resultado da operação executada no **Acumulador A**.

O dispositivo **Registrador B** caracteriza-se por ser um registrador de memória temporária auxiliar que coleta do barramento W determinado valor e o transfere para o dispositivo **Somador/Subtrador** que é parte da Unidade Lógica e Aritmética do computador SAP-1.

O dispositivo **Registrador de Saída** também denominado *porta de saída* (MALVINO, 1985, p. 259) ou *registrador de saída* (CAMPOS, 2011, p. 155) é o componente responsável por coletar o resultado da operação de processamento que esteja armazenado no **Acumulador A** e transferi-lo para o mundo exterior e apresentá-lo junto ao dispositivo **Indicador Visual Binário**. Malvino (1985, p 259) acrescenta que:

“Em microcomputadores as portas de saída são conectadas aos circuitos de interface que comandam dispositivos periféricos como as impressoras, os tubos de raios catódicos, as teleimpressoras etc.”

O dispositivo **Indicador Visual Binário** é no

computador SAP-1 um painel luminoso com oito LEDs (Diodo Emissor de Luz = *Light-Emitting Diode*) que representam na forma binária o valor que fora transferido do **Acumulador A** para o **Registrador de Saída**.

PRGRAMAÇÃO SAP-1

A arquitetura SAP-1 considera como linguagem de baixo nível um *assembly* com cinco mnemônicos como mostra a Tabela 1.

Mnemônico	Operação	Op-code
LDA	Carrega dados da RAM no acumulador	0000
ADD	Soma dados da RAM com o acumulador	0001
SUB	Subtrai dados da RAM do acumulador	0010
OUT	Carrega dados da RAM no acumulador de saída	1110
HLT	Interrompe o processamento	1111

Tabela 1 – Conjunto de instruções SAP-1.

A partir das instruções definidas na Tabela 1 o programa a seguir calcula e mostra o resultado da operação aritmética: $7 + 2 - 1$.

Note que o programa carrega o acumulador do SAP-1 com o valor **07H** do endereço **DH** a partir da execução da instrução **LDA DH**. Em seguida efetua a adição do valor **02H** do endereço **EH** junto ao valor armazenado no acumulador com a execução da instrução **ADD EH**, neste momento o acumulador possui o valor **09h**. Na sequência, o programa efetua a subtração do valor **01H** do endereço **FH** do valor armazenado no acumulador com a instrução **SUB FH**. A instrução **OUT** tem por finalidade mostrar o resultado **08h** no formato binário **00001000** em seu indicador visual binário e a instrução **HLT** finaliza o programa.

Endereço	Instrução
0H	LDA DH
1H	ADD EH
2H	SUB FH
3H	OUT
4H	HLT
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	07H
EH	02H
FH	01H

Tabela 2 – Aplicação $7 + 2 - 1$.

A PROGRAMAR

INTODUÇÃO A ARQUITETURA SAP COM FOCO EM SAP-1

O programa a seguir realiza operação aritmética: $75 + 20 - 18$.

Veja que programa carrega o acumulador do SAP 1 com o valor **4BH** do endereço **DH** a partir da execução da instrução **LDA DH**. Em seguida efetua a adição do valor **14H** do endereço **EH** junto ao valor armazenado no acumulador com a execução da instrução **ADD EH**, neste momento o acumulador possui o valor **5Fh**. Na sequência, o programa efetua a subtração do valor **12H** do endereço **FH** do valor armazenado no acumulador com a instrução **SUB FH**. A instrução **OUT** apresenta o resultado **4Dh** no formato **01001101** em seu indicador visual binário e a instrução **HLT** finaliza o programa.

Endereço	Instrução
0H	LDA DH
1H	ADD EH
2H	SUB FH
3H	OUT
4H	HLT
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	4BH
EH	14H
FH	12H

Tabela 3 – Aplicação $75 + 20 - 18$.

A partir do código *assembly* escrito, é possível converter um programa em código de máquina (Op-code) de acordo com as instruções indicadas na tabela 1.

Os programas apresentados possuem o mesmo conjunto de instruções caracterizando o uso do mesmo algoritmo (A+B-C). No entanto cada um possui um conjunto de valores diferenciados.

Observe que nos códigos apresentados estão sendo separados os dados do programa em relação as suas instruções em áreas distintas de memória. Note que os dados estão definidos entre os endereços **DH** e **FH**. As instruções do

programa estão definidas a partir do endereço **0H** até no máximo ao endereço **CH**.

Programa $7 + 2 - 1$		Programa $75 + 20 - 18$	
Endereço	Instrução	Endereço	Instrução
0000	0000 1101	0000	0000 1101
0001	0001 1110	0001	0001 1110
0010	0010 1111	0010	0010 1111
0011	1110	0011	1110
0100	1111	0100	1111
0101		0101	
0110		0110	
0111		0111	
1000		1000	
1001		1001	
1010		1010	
1011		1011	
1100		1100	
1101	0000 0111	1101	0100 1011
1110	0000 0010	1110	0001 0100
1111	0000 0001	1111	0001 0010

BIBLIOGRAFIA

CAMPOS, F. M. P. **Arquitetura e Organização de Computadores**. Batatais: Ação Educacional Claretiana, 2011.

MALVINO, A. P. **Microcomputadores e Microprocessadores**. São Paulo: Editora McGraw-Hill, 1985.

AUTOR



Escrito por Augusto Manzano

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

Raiz enésima iterativa programada

Este artigo apresenta conceitos e código de demonstração de um programa escrito em linguagem C++ que calcula e exibe no ecrã de forma iterativa o resultado de uma raiz enésima com base e índice informados para o cálculo.

INTRODUÇÃO

Um facto a ser considerado é que as ciências relacionadas a matemática e a computação andam juntas. Neste sentido, vem a mente como soluções matemáticas simples são implementadas na arquitetura de um computador. A questão não se refere a soluções relacionadas a parte electrónica do computador, mas como são realizadas tais operações a luz da base matemática.

As operações básicas de adição, subtração, multiplicação e divisão são de fácil dedução a partir de seus algoritmos funcionais que são ensinados nas escolas, aprende-se a realizar essas operações a partir de diversos e sucessivos exercícios para fixá-los, desde o nível aritmético até o nível algébrico.

No estudo da álgebra é feito o contacto com equações, inequações, exponenciações, radiciações e diversas outras ferramentas e elementos matemáticos.

A operação de exponenciação é facilmente percebida a partir de um algoritmo iterativo de multiplicação sucessiva da base em relação ao número de vezes indicado no índice definido, facilmente aplicado em um programa de computador. No entanto, a operação de radiciação iterativa não se mostra tão simples.

A partir do interesse sobre tal operação veio uma longa pesquisa sobre este facto, onde alguns professores da área matemática não souberam responder sobre a existência de tal algoritmo e parte dos interpelados até criticaram jocosamente tal curiosidade e demonstraram estarem ofendidos.

Dentre as diversas fontes pesquisadas uma trouxe resposta clara a questão. Um livro intitulado "Manual de Fórmulas Técnicas" do autor Kurt Gieck publicado no Brasil pela editora **Hemus** e em Portugal pela editora **Dinalivro** em seu primeiro volume.

FUNDAMENTAÇÃO MATEMÁTICA

A solução de uma raiz enésima (ou n-ésima) pode ser obtida a partir da exponenciação da base elevando-se esta ao valor inverso do índice de radiciação como apresentado na figura 1.

$$x = \sqrt[n]{b} \text{ equivale a } x = b^{\frac{1}{n}}$$

Figura 1

No entanto, o que mostra a figura 1 não é a fórmula pretendida, pois a solução iterativa do cálculo ocorre a partir do exposto junto a Figura 2.

$$x = \frac{1}{n} \left[(n-1)x_0 + \frac{b}{x_0^{n-1}} \right]$$

Figura 2

A equação indicada na Figura 2 caracteriza-se a partir da definição dos valores da base (**b**), do índice (**n**) e de **x₀** como resultado sucessivo inicialmente definido com valor **1** para obter iterativamente o resultado de **x**.

A partir deste estágio pega-se o valor da incógnita **x** e efetua-se a substituição na equação junto ao valor da incógnita **x₀** e calcula-se um novo valor para a incógnita **x**. A partir deste passo efetua-se a repetição da substituição do novo valor da incógnita **x** pelo valor da incógnita **x₀** até que os valores das incógnitas **x** e **x₀** sejam iguais quando se terá o resultado efetivo da raiz calculada.

O PROGRAMA

O código C++ seguinte aplica o algoritmo apresentado com o uso do cálculo iterativo de raiz enésima.

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;

int main(void)
{
    char LETRA;
    double B, N, X, XZero;
    cout << setprecision(16);
    cout << setiosflags(ios::fixed);
    cout << endl;
    cout << "Informe a base .... ";
    cin >> B;
    cout << "Informe o indice .. ";
    cin >> N;
    X = 1;
    do
    {
        XZero = X;
        X = (1/N) * ((N-1) * XZero +
            (B / pow(XZero, N-1)));
    }
    while (X != XZero);
    cout << "Resultado ..... ";
    cout << setw(16) << X << endl;
    return 0;
}
```

Considerando o cálculo da raiz quadrada de base 5 ter-se-á calculado o resultado da raiz como: 2.2360679774997898.

A PROGRAMAR

RAIZ ENÉSIMA ITERATIVA PROGRAMADA

CONCLUSÃO

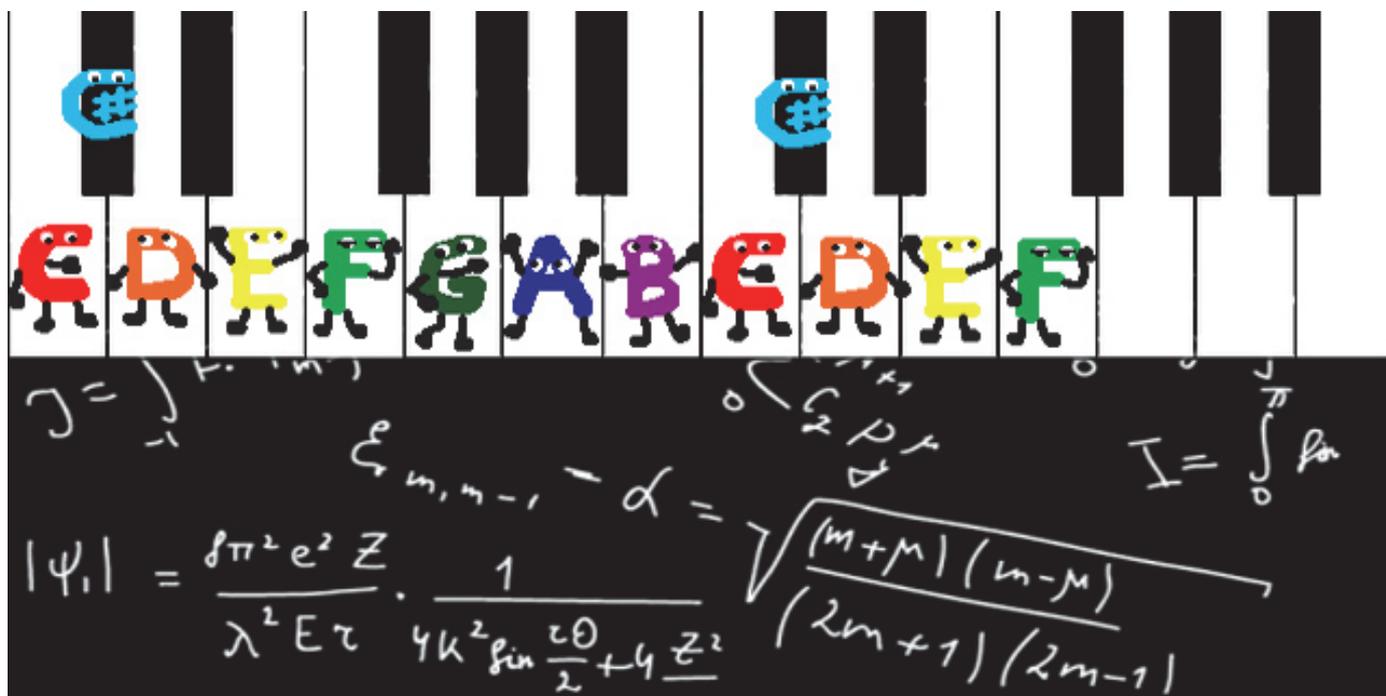
O encontro da equação para cálculo de raiz enésima publicado pelo autor Gieck trouxe luz a questão de uma maneira impar, além de ser um procedimento operacional muito simples e ter respondido a uma pergunta, que até então, parecia insolúvel.

“ (...) soluções matemáticas simples são implementadas na arquitetura de um computador (...) ”

Um ponto que chamou a atenção dentro da curiosidade manifestada e que levou a esta breve pesquisa foi não ter encontrado resposta condizente.

“ (...) as ciências relacionadas a matemática e a computação andam juntas (...) ”

Foi o fato de alguns professores não estarem preparados para responder perguntas que fogem do padrão acadêmico de referência das suas aulas, preparadas. Desconhecer um tema que foge do escopo da aula planejada é natural, pois não é necessário que se saiba tudo sobre certo tema. O que fica a se refletir é o descaso ou desinteresse em buscar responder certa pergunta mesmo com uma resposta do tipo “eu não sei, vamos pesquisar sobre este fato”. O que, de fato, movimenta o conhecimento no mundo não são as respostas dadas, mas as perguntas realizadas.

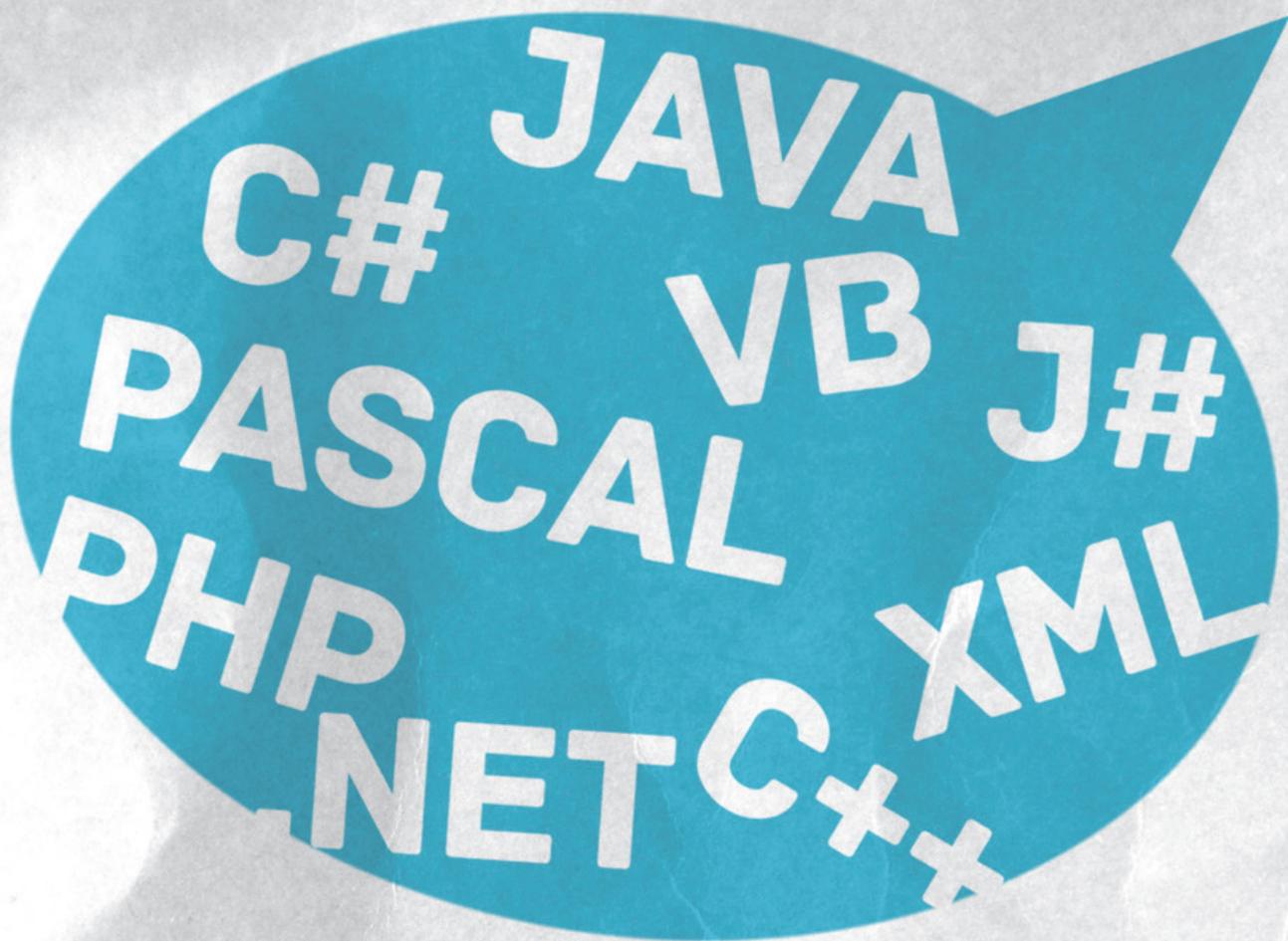


AUTOR



Escrito por Augusto Manzano

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



ENTÃO, SÓ FALAS
EM CÓDIGO?

TEMOS O REMÉDIO CERTO PARA TI!



portugal-a-programar.pt

A MAIOR COMUNIDADE PORTUGUESA DE
PROGRAMAÇÃO, APARECE!

ELECTRÓNICA

Conexão de Arduino ao Unity

ANDON

CONEXÃO DE ARDUINO AO UNITY

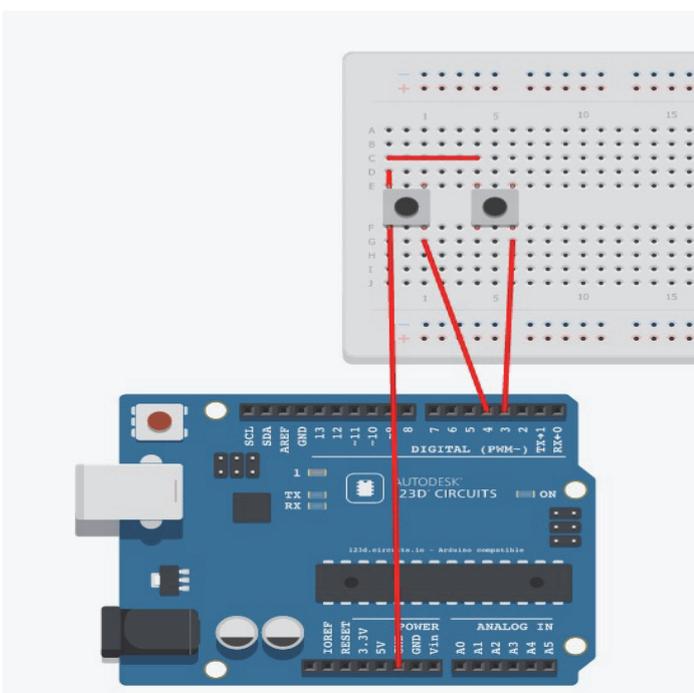
O mundo da tecnologia é motivado e “limitado” pela imaginação de cada um. Desta feita o artigo desta edição prende-se com a ligação de um micro-controlador Arduino ao conhecido Game Engine Unity.

O objetivo deste artigo é explicar como proceder à ligação de um micro-controlador Arduino ao Unity e usar este como uma espécie de “comando” para o jogo criado no Unity. Será um exemplo extremamente simples que resultará no controlo de um simples sprite no eixo do X sem qualquer animação especial.

Para este pequeno tutorial necessitamos dos seguintes elementos:

- > Arduino Uno
- > 2 x Push Buttons
- > Cabos de Conexão
- > Arduino IDE
- > Unity

Uma vez cumpridos os requisitos vamos dar inicio à montagem do pequeno circuito eletrónico que será o nosso “comando” de controlo do personagem do exemplo em Unity.



Depois de implementado o seguinte esquema eletrónico iremos passar à programação do micro-controlador Arduino.

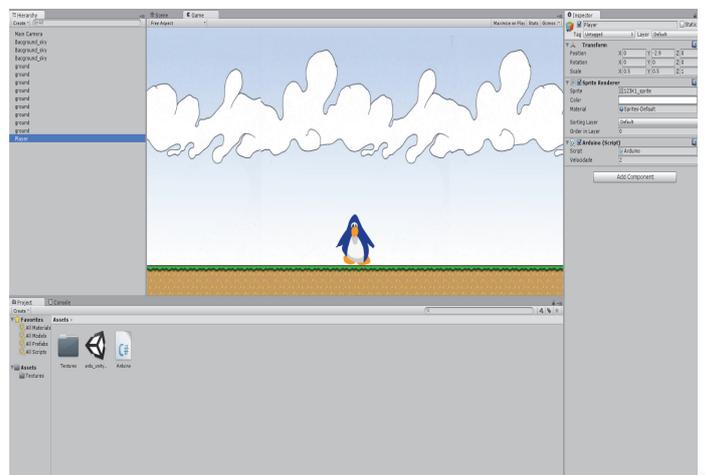
Aqui fica o código necessário:

```
//Definição dos pin's responsáveis pelos
//respectivos Push Buttons
const int botaoEsquerdo = 3;
const int botaoDireito = 4;
void setup()
{
    Serial.begin(9600);
    pinMode(botaoEsquerdo, INPUT);
    digitalWrite(botaoEsquerdo, HIGH);
    pinMode(botaoDireito, INPUT);
    digitalWrite(botaoDireito, HIGH);
}
//Método de Leitura Constante do Estado dos Push
//Buttons
void loop()
{
    if(digitalRead(botaoEsquerdo) == LOW)
    {
        Serial.write(1);
        Serial.flush();
        delay(20);
    }
    if(digitalRead(botaoDireito) == LOW)
    {
        Serial.write(2);
        Serial.flush();
        delay(20);
    }
}
```

Feito o upload do código acima mostrado para o Arduino vamos passar ao Unity.

Para isso necessitamos de criar um novo projeto, assim como uma nova “scene” onde poderemos colocar um background ao nosso gosto e um “personagem” que será controlado pelo “comando” acionado pelo Arduino.

No exemplo mostrado usei ainda um elemento para o solo ao qual adicionei um “BoxCollider” assim como o personagem tem também adicionado um “BoxCollider” para que não caia no infinito. Mas estes pormenores podem ser personalizados ao gosto de cada um.



Electrónica

CONEXÃO DE ARDUINO AO UNITY

A imagem acima mostra o exemplo que criei para ilustrar este tutorial.

Após a criação do cenário propriamente dito temos de então de criar um script em C# que será o coração do movimento e ligação com o Arduino.

Aqui está o código que deverá ser incluído no script C# para que possamos então conectar o nosso “comando” Arduino ao Personagem implementado no Unity.

```
using UnityEngine;
using System.Collections;
using System.IO.Ports;

public class Arduino : MonoBehaviour
{
    //Definição da variavel velocidade
    public float velocidade;
    //Definição da Porta COM que sera utilizada
    SerialPort porta = new SerialPort("COM3",
    9600);

    // Use this for initialization
    void Start()
    {

        //Abertura da Porta COM para conexao com o
        //Arduino
        porta.Open();
        porta.ReadTimeout = 1;
    }

    // Update is called once per frame
    void Update()
    {
        //Caso a porta COM responda positivamente
        //estao executaremos o metodo Mover()
        if (porta.IsOpen)
        {
            try
            {
                Mover(porta.ReadByte());
            }
            catch (System.Exception)
            {
            }
        }
    }

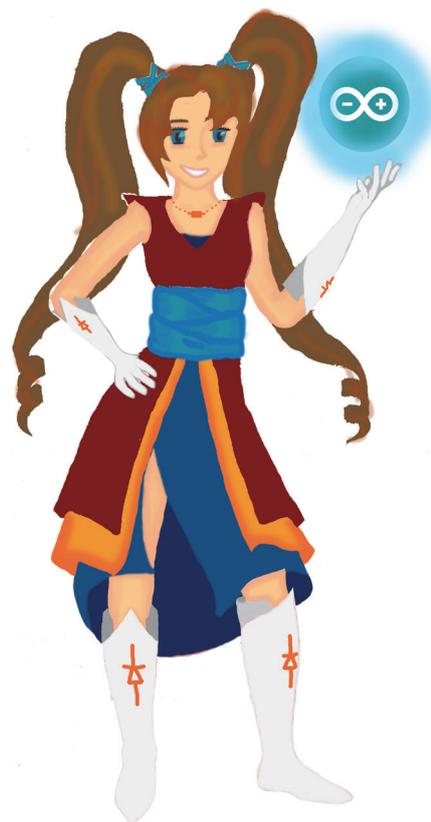
    //Metodo Mover que sera chamado a cada clique
    //no Push Button
    void Mover(int direcao)
    {
        //Para a Direita
        if (direcao == 1)
        {
            transform.Translate(-Vector2.right *

```

```
velocidade * Time.deltaTime, Space-
ce.World);
        }
        //Para a Esquerda
        if (direcao == 2)
        {
            transform.Translate(Vector2.right *
            velocidade * Time.deltaTime, Space.World);
        }
    }
}
```

Após criar o script C# apenas é necessário arrastá-lo para para o “Inspector” referente ao personagem e se tudo correu bem ao colocar a cena em execução o Arduino será capaz de enviar para o Unity as nossas ações e reproduzi-las em tempo real.

Este artigo é apenas um pequeno exemplo de como controlar um personagem usando o Arduino, no entanto pode ser aprofundado e explorado ao gosto de cada um, sendo possível implementar uma ação “saltar”, “disparar”, basicamente tudo o que a nossa imaginação disser.



AUTOR

Escrito por Nuno Santos

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

ANDON

Ando há anos a tentar vender a ideia, de que na automatização de pequenos processos industriais, é possível substituir os autómatos por pequenos microcontroladores, no meu caso o arduino. As vantagens são suficientemente aliciantes para avançar com estes projectos :

- 1) Baixo custo do microcontrolador (arduino cerca de 40€)
- 2) Linguagem muito perto do C , com muitas bibliotecas para os mais diversos fins.

O que vos mostro de seguida, é um processo industrial real, que uma equipa na qual eu participei construiu e pôs em funcionamento, com um orçamento inferior a 1000€.

O Problema

Numa linha de acabamento de peças de automóveis, há 8 cabines de rebarbagem de peças, cada uma com um operador e equipadas da mesma forma.

Pretendia-se saber a cadencia de cada operador, descontar a sucata sempre que uma peça não estivesse conforme e poder chamar o chefe da linha sempre que falte material ou sempre que haja um problema de qualidade. Estes dados deviam estar disponíveis imediatamente na intranet da empresa e num monitor de 21" visível em todo o sector.

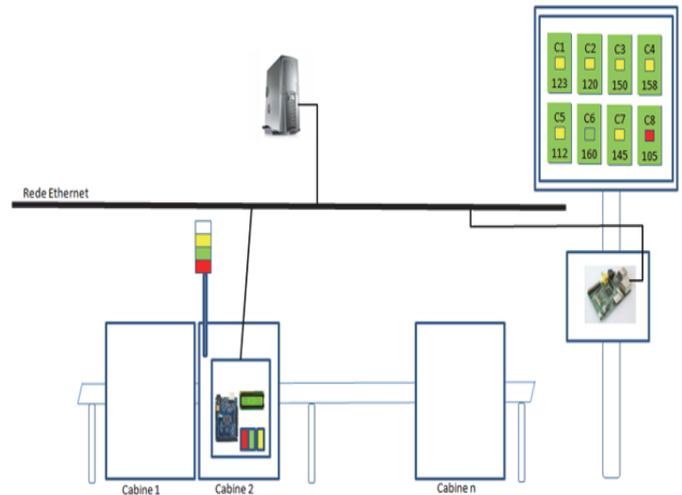
A abordagem

A um sistema com estas características, chama-se ANDON, foi desenvolvido e aplicado pela Toyota e a melhor definição que encontrei foi “ uma forma de gestão à vista das ocorrências e resultados do local de trabalho, apresentando nas formas de quadros, sinalizadores sonoros ou visuais” (<http://engenhariadeproducaoindustrial.blogspot.pt/2009/07/andon-gestao-vista.html>)

Por muito pequeno que nos pareçam os projetos, em empresas de grande dimensão, com preocupações para além do “trabalha ou não trabalha”, a solução definitiva tem em conta imensos detalhes e envolve muitas competências. Depois de muitas reuniões ficou, em síntese, definido o seguinte:

- Cada cabine teria uma consola com 4 botões e um display, que serve para acompanhamento do operador.
- O monitor principal receberia os dados de um computador perto dele e ligado à rede ethernet da fábrica.
- O operador numa das suas operações ao rebarbar a peça deveria fazer acionar um switch que dará informação ao microcontrolador.

A Solução



Como vemos pela figura o layout final ficou assim :

- Cada consola tem um arduino ethernet, para o qual se atribuiu um ip da rede interna. Tem ainda o LCD e 4 botões
- Algures na rede, existe um servidor que corre um programa (desenvolvido em Visual Basic) que intercepta o que cada consola envia e atualiza uma base de dados.
- Para reduzir os custos, foi utilizado um raspberry que acede à base de dados (MYSQL) e atualiza o monitor de 10 em 10 s. Esta solução obrigou-nos a aprender python, que devo dizer é uma agradável surpresa.

O detalhe

Para os menos familiarizado com o arduino, aconselha-se a dar uma vista de olhos em (<http://www.arduino.cc/>)

De acordo com o esquema em baixo, temos:

- 1) Cada um dos botões é ligado a uma entrada digital do arduino
- 2) Como existia uma linha de 24V todas as cabines são alimentadas com esta tensão. Para alimentar o arduino, optamos por um modulo regulador , que nos baixou a tensão para 12V.
- 3) No esquema existe ainda um outro regulador com um 4N33 que converte os 24V da alimentação vindos da célula fotoelétrica, num sinal de 5V para uma porta digital do arduino. Inicialmente tinha sido previsto uma célula indutiva para detetar o movimento de uma das operações de modo a fazer a contagem. A experiência mostrou-nos que seria preferível, acionar um

switch. De qualquer forma, optou-se por deixar a célula.

- 4) A comunicação com o LCD faz-se a partir de um PCF8574, que é um expensor de portas entradas/saídas de 8bits por interface de comunicação I2C. Evitou-se desta forma, ocupar as poucas portas que o arduino uno disponibiliza.

Como se entendeu que este processo pudesse evoluir, deixaram-se 2 saídas disponíveis para qualquer necessidade posterior.

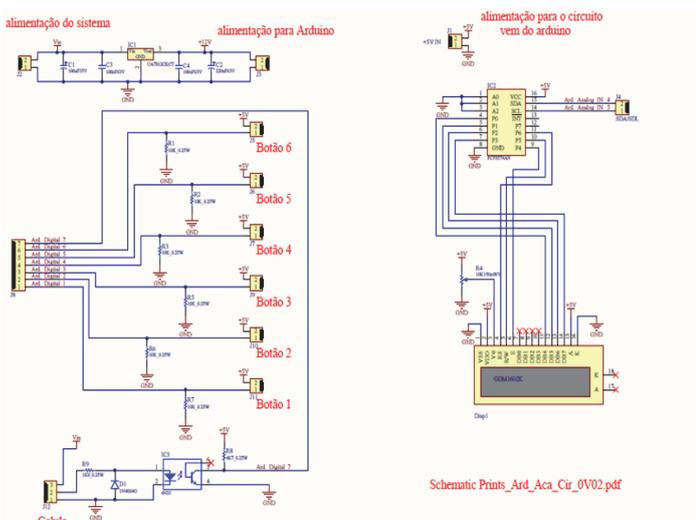


Fig.1

Depois do protótipo

O primeiro protótipo foi feito de um modo muito artesanal e cedo se percebeu que embora o microcontrolador seja um elemento de baixo custo, toda a sua envolvente teria de estar de acordo com as normas exigidas pela empresa.

- 1) Os botões têm de respeitara a norma IP55 em termos de estanqueidade, segurança e fiabilidade.
- 2) Devia ser feito um PCB á medida para substituir todos os fios.
- 3) Deveria haver um quadro elétrico devidamente protegido.
- 4) Todos os detetores serão de acordo com as normas da empresa

Além de tudo isto há uma série de outras competências, que escapam por vezes a alguém que apenas pensa na solução do ponto de vista do protótipo. Desde o serralheiro que deve adaptar os vários componentes à estrutura existente, à preocupação constante com o operador de modo a que os módulos adicionais não impeçam o bom funcionamento, passando pelo aspectos do acabamento e do treino da equipa que vai operar e manter o sistema nada deve ser descurado.

Fotos reais

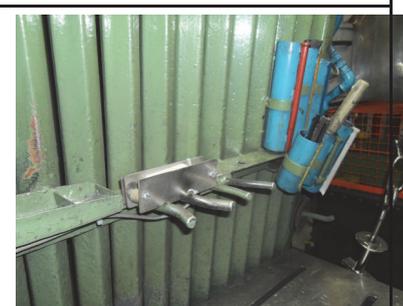
O display a 3m de altura. O raspberry está por trás.



O Terminal do operador.



O Switch na mesma cabine do operador e que atua quando é encostado (atirado) um punção.



Conclusão

Embora possa parecer um cliché, cada vez mais acredito, que tudo isto se consegue, se houver uma direção que acredite e uma equipa de desenvolvimento com vontade e persistência.

COLUNAS

C# - Async-Await - A alegoria do restaurante

ASYNC-AWAIT - A ALEGORIA DO RESTAURANTE

Introdução

Com a introdução de capacidades assíncronas nas linguagens de programação **C#** e **Visual Basic** todas as APIs assíncronas tendem a seguir este padrão.

No entanto, muitos programadores não entendem ainda como lidar com esta nova realidade.

Este artigo não pretende explicar como funciona esta funcionalidade, mas apresentar uma alegoria que permite formar um modelo mental de como se deve funcionar com esta tecnologia.

Aplicações cliente

A maioria das interfaces gráficas de utilizador (smartphones, PCs, tablets, etc.) é gerida por um *thread* dedicado à gestão da interação com o utilizador e atualização do ecrã. Como tal, deve-se evitar bloquear esse *thread* ou efetuar nele computações que não estejam relacionadas com a sua função.

Imaginemos um restaurante com apenas uma empregada de mesa encarregue do serviço à sala. Essa empregada de mesa só pode executar uma tarefa de cada vez e, se for bloqueada, não vai poder atender nenhum cliente.

Quando um novo cliente chega, notifica a empregada de mesa que chegou e esta, assim que possível, acompanha-o a uma mesa, entrega-lhe uma ementa e prossegue para a próxima tarefa enquanto o cliente faz a sua escolha.

Quando cliente tiver alguma dúvida, chama a empregada de mesa, esta desloca-se à mesa, esclarece a dúvida, e parte para a tarefa seguinte.

Quando o cliente souber o que vai pedir, chama a empregada de mesa, faz o seu pedido, a empregada de mesa entrega-o na cozinha e parte para a tarefa seguinte.

Quando o serviço da cozinha está pronto, a empregada de mesa leva-o ao cliente e parte para a tarefa seguinte.

Como se pode deduzir, para que a empregada de mesa consiga servir o maior número de clientes possível sem que estes esperem muito e tenham a sensação de que são o único cliente, não se deve chamar a empregada de mesa demasiadas vezes nem chamá-la quando estamos a fazer outra coisa (como ler a ementa, atender um telefonema, etc.). De igual modo, não devemos bloquear a empregada de mesa se quisermos falar com outro funcionário do restaurante (o cozinheiro, por exemplo).

Colocando esta interação em C#, seria algo como:

```
void HandleCustomer(object source,
                    CustomerEventArgs e)
```

```
{
    WalkCustomerToTable();
    GiveMenuToCustomer();
    while (true)
    {
        switch (await CustomerRequestAsync())
        {
            case Question:
                var question =
                    await GetCustomerQuestionAsync();
                if (question.CanAnswer())
                {
                    AnswerCustomerQuestion(
                        question);
                }
            else
            {
                await Task.Run(() =>
                    RequestSomeoneElseToAnswer());
            }
            break;
            case Order:
                var order =
                    await GetCustomerOrderAsync();
                TakeCustomerOrder(e.Customer,
                                    order);
            break;
            case Bill:
                GiveBillToCustomer(e.Customer);
            break;
            case Payment:
                var payment =
                    await CustomerPaymentAsync();
                ProcessCustomerPayment(
                    e.Customer,
                    payment);
            return;
            ...
        }
    }
}
```

A interação da empregada de mesa com a cozinha pode ser representada de forma semelhante:

```
void HandleCustomerOrder(object sender,
                        CustomerOrderEventArgs e)
{
    var meal = await GetMealFromOrderAsync(
                                                e.Order);
    DeliverMealToCustomer(e.Customer, meal);
}
```

Nesta analogia, a empregada de mesa é o *thread* de UI.

Aplicações servidor

O objetivo de um servidor (um servidor web, por exemplo) é servir o cliente do início ao fim o mais rapidamente possível para que o cliente vá à sua vida e possa ser servido outro cliente.

Isto é equivalente a restaurante de fast-food. Existe um conjunto (*pool*) de estações de atendimento que atende

ASYNC-AWAIT - A ALEGORIA DO RESTAURANTE

um cliente de cada vez do princípio ao fim.

Numa configuração destas, é mais eficiente esperar um pouco que o cliente pense do que fazê-lo sair da fila para pensar e depois voltar para a fila e recuperar o estado do seu pedido quando finalmente chegar a um empregado. Assim como também não faz sentido o cliente pedir umas coisas a um empregado e outras a outro - Task.Run.

Mas quando o pedido está completo, o empregado paraleliza ao máximo possível as tarefas a realizar, mais uma vez, com o objetivo de despachar o cliente o mais rapidamente possível. Uma vez que são devidamente organizados, estes empregados até podem trocar de cliente entre si sem que o cliente tenha uma perceção negativa do desempenho do "sistema".

Neste caso, atender o cliente seria algo como:

```
Meal HandleCustomerOrder(CustomerRequest request)
{
    var order = GetOrderFromRequest(request);
    var kitchenTask =
        GetRequestFromKitchenAsync(order);
    var beverageTask = GetBeverageAsync(order);
    var fries = GetFries(order);

    await Task.WhenAll(kitchenTask, beverageTask);
    return new Meal(
```

```
kitchenTask.Result,
beverageTask.Result,
fries);
}
```

Conclusão

Embora muito simplista, esta analogia permite equiparar mentalmente o que se passa na execução do código assíncrono com o mundo real e quais os padrões mais comuns.

Recursos

Talk: Async best practices <http://blogs.msdn.com/b/lucian/archive/2013/11/23/talk-mvp-summit-async-best-practices.aspx>

Six Essential Tips For Async – Introduction <http://channel9.msdn.com/Series/Three-Essential-Tips-for-Async/Three-Essential-Tips-For-Async-Introduction>

Curah! `async-await` `General` <http://curah.microsoft.com/45553/asyncawait-general>

Curah! `async-await` `and` `ASP.NET` <http://curah.microsoft.com/44400/async-and-aspnet>



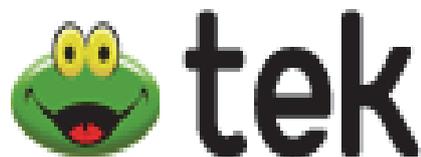
AUTOR



Escrito por Paulo Morgado

Bacharel 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 exerce várias funções relacionadas com o desenvolvimento, distribuição e manutenção de software há mais de 10 anos. Participa em diversas comunidades nacionais e internacionais (pontoNETpt, NetPonto, SharePointPT, SQLPort, Portugal-a-Programar, CodeProject, CodePlex, etc.). Pelo seu contributo para com estas comunidades, a Microsoft premeia-o com o prémio MVP (C#) desde 2003. É ainda co-autor do livro "LINQ Com C#" da FCA.

Media Partners da Revista PROGRAMAR



Análises

Projetos de POO em Java

Introdução à Engenharia de Software

Projetos de POO em Java

Título: Projetos de POO em Java

Autores: F. Mário Martins

Editora: FCA - Editora de Informática

Páginas: 314

ISBN: 978-972-722-792-1



Fazem parte da minha biblioteca os excelentes títulos, “Programação Orientada aos Objectos em JAVA 2” e “JAVA6 e Programação Orientada pelos Objectos” e neste sentido foi com agradável surpresa que consultei o título, “Projetos de POO em Java” onde destaco desde já a notável estrutura dos conteúdos e a orientação para o projeto a partir do capítulo quarto.

Sendo um livro direcionado para profissionais de informática e estudantes e tendo em conta que os cursos de informática atuais incluem no seu currículo unidades curriculares de Metodologias e Gestão de Projetos, o autor torna muito fácil compreender a forma de estruturar e implementar pequenos projetos de software a partir da análise de requisitos. Citando António Santos (2014) “... seria um agrado ver na bibliografia dos novos cursos TeSP...”, referindo-se a este livro, adiria que a obra ficaria bem na bibliografia de qualquer unidade curricular de programação em java.

Das muitas agradáveis surpresas como a já mencionada estrutura e organização dos conteúdos outra que suscita interesse e que será do agrado de todos diz respeito à análise que é realizada no final de cada exercício. O autor apresenta o enunciado do problema, apresenta a solução codificada e logo destaca uma análise completa explanando claramente os passos dados e salientando os pormenores do código. Esta estrutura repete-se ao longo de todos os capítulos tendo a sua maior importância a partir do capítulo quatro.

Nos dois primeiros capítulos é introduzida a programação em linguagem Java indicando toda a sua sintaxe essencial, abordando com detalhe os arrays e incluindo inúmeros exemplos práticos. O autor abstrai-se de complicações e foca-se no essencial, nunca economizando explicações para um correto entendimento dos conceitos. Percebemos esta orientação ao longo da leitura da obra e já no início, com a utilização da classe Scanner, demonstra a experiência e o interesse em transmitir ao leitor uma aprendizagem sólida e sem incompreensões desnecessárias. No final destes capítulos o leitor será capaz de criar pequenos programas com introdução de dados, estruturas de decisão e controlo, arrays, funções com retorno, gerar números aleatórios, manipular data e hora, utilizar um IDE... portanto, entender todos

os detalhes da linguagem na sua orientação procedimental.

Do meu ponto de vista, o terceiro capítulo é o mais crítico devido à abordagem orientada a objetos. O autor, mais uma vez, surpreende com a explicação clara dos princípios, técnicas e termos associados à POO não esquecendo introduzir um ponto importante de comparação entre o paradigma imperativo e orientado a objetos. Aliado a isto encontramos um exercício exemplo completo e bem explicado que leva o leitor a entender como modelar um objeto, perceber o que é uma classe e um objeto, o que são os atributos, os comportamentos, os construtores... culminando com a criação de objetos. Sempre que necessário o autor explica como utilizar o IDE (BlueJ) para responder aos novos conteúdos abordados.

“ **O autor abstrai-se de complicações e foca-se no essencial, nunca economizando explicações para um correto entendimento dos conceitos (...)** ”

No quarto capítulo são explicadas as coleções do Java (arrays com características dinâmicas). O autor começa por explicar os ArrayLists da mesma forma com que já nos habituou, passando posteriormente para as coleções do tipo set (HashSet e TreeSet). Este capítulo merece especial destaque pela condução orientada ao projeto. Nos seus exercícios exemplo, parte da análise de requisitos à implementação efetiva de uma aplicação referindo a prototipagem no IDE BlueJ.

No que concerne ao quinto capítulo, explica hierarquia de classes, herança, classes abstratas e polimorfismo portanto, características da programação orientada a objetos introduzida numa perspetiva de organização das classes como vantagem à flexibilidade e reutilização de código. Através de exercícios, o autor introduz inúmeros detalhes de

extrema importância como são exemplo a característica de dynamic method lookup. Termina o capítulo em beleza, com o polimorfismo, associando e reforçando neste ponto a extensibilidade dos programas que articula a dois grandes exercícios exemplo por ele analisados.

“ **Percebemos esta orientação ao longo da leitura da obra e já no início, com a utilização da classe Scanner, demonstra a experiência e o interesse em transmitir ao leitor uma aprendizagem sólida e sem incompreensões desnecessárias(...)** ”

No sexto capítulo é realizada a abordagem às coleções com estruturação do tipo Map nas implementações HashMap e TreeMap. A experiência do autor é revelada ao colocar este conteúdo no seguimento dos anteriores. Este capítulo, uma vez mais, dá ênfase ao detalhe e clareza do tema abordado. Saliento o ponto em que o autor explica as regras na escolha de uma coleção seguindo os princípios da Engenharia de Software e o ponto onde se aborda a Interface Comparable e Comparator. Finaliza com a generalização e abstração do código, sempre acompanhados por exemplos muito completos, não esquecendo mencionar o

símbolo especial “?” com vista a garantir compatibilidade nem explicar o operador diamante “<>”.

No sétimo e último capítulo aborda a leitura e escrita de dados (Streams I/O) segundo a perspectiva da leitura/escrita de caracteres e leitura/escrita de bytes, não se contendo em explicações, na já referida síntese teórica, para bom entendimento. Entre muitos pormenores destaco a abordagem das linhas de texto, parsing, delimitadores, expressões regulares(Pattern, Matcher), ficheiros de texto e ficheiros binários e muito especial destaque para a abordagem à codificação de caracteres(charset) que muitas dúvidas costumam trazer e que muito bem se explica e por último a serialização de objetos.

“ (...) **esta surpreendente obra encontra suporte na elevada experiência técnica e pedagógica do seu autor** ”

Em jeito de conclusão, esta surpreendente obra encontra suporte na elevada experiência técnica e pedagógica do seu autor, apostando nas exigências atuais de orientação a projetos e na resolução de problemas reais. Tenho a certeza que todos concordarão que este livro é um dos melhores guias editados por todos os motivos já referidos e dos quais destaco uma vez mais a orientação ao projeto, a introdução teórica dos conteúdos, a análise completa dos inúmeros exercícios codificados e o foco nos pormenores que costumam levantar sempre bastantes dúvidas. Devo acrescentar que o autor fornece na página da editora todos os exercícios do livro na forma de código fonte(.java) e na forma compilada (.class).

Parabéns ao autor e à editora por tão sublime trabalho.

AUTOR



Escrito por **Bruno Gomes**

Formado em Engenharia de Sistemas de Informação, Engenharia Informática e Mestrado em Tecnologia e Gestão de Sistemas de Informação é atualmente docente na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Viana do Castelo onde colabora essencialmente em unidades curriculares de gestão de projetos, programação procedimental e orientada a objetos incluindo tecnologia Android.

Introdução à Engenharia de Software

Título: Introdução à Engenharia de Software

Autores: Sérgio Guerreiro

Editora: FCA - Editora de Informática

Páginas: 256

ISBN: 978-972-722-795-2



O livro em análise, apresenta-se como uma boa aquisição para a biblioteca de alguém que queira aprofundar os seus conhecimentos sobre o desenvolvimento de software e levar seriamente a sua produção, tendo em conta os bons princípios a ele inerentes. Enquadra-se perfeitamente no currículo de Engenharia Informática, introduzindo o leitor ao mundo da criação de software. Ao longo de oito capítulos o autor explica de forma clara diversos conceitos, bem como aborda temas relevantes para o panorama de um programador.

No primeiro capítulo, o autor faz uma introdução ao que é a Engenharia de Software, levando o leitor a compreender de uma forma simples e clara conceitos específicos desta área. Desde o que é qualidade no software, passando por problemas frequentes como a comunicação, as alterações e o fator tempo, até aos desafios que o leitor encontra no desenvolvimento de software, são apresentados sistematicamente pontos essenciais nestes campos, desde os processos agile ao desenvolvimento por modelos e arquitectura orientada a serviços e aplicações móveis.

No segundo capítulo o autor aprofunda os processos de desenvolvimento de software. Clarificando sempre todos os conceitos, define os intervenientes, problemas e desafios bem como o enquadramento de processo de desenvolvimento. Aborda não só os processos de desenvolvimento clássicos tais como waterfall, por protótipos, desenvolvimento rápido, espiral com diferentes abordagens, incremental e etapas concorrentes, mas também os processos ágeis de software, como a programação externa, a modelação ágil, o IRUP e o SCRUM, descrevendo cada um e apresentando as suas vantagens e desvantagens.

No terceiro capítulo o autor explana a gestão do processo de desenvolvimento de software, desde o seu planeamento, passando pelas diferentes métricas, a gestão de ris-

cos, de configurações e de equipas, até às normalizações e boas práticas da IEEE, nomeadamente a norma IEEE 15939.

No quarto capítulo encontramos o tema da engenharia de requisitos e da comunicação, que tão importante é no desenvolvimento de software. Como sabemos, quanto melhor for um levantamento de requisitos, mais hipóteses existem de o produto final corresponder ao que o cliente deseja. Intervenientes, problemas, actividades relevantes, soluções, normalização e boas práticas são aqui descritos de uma forma simples e compreensível.

O quinto capítulo “introduz a análise e desenho do sistema como a etapa responsável por conceptualizar o produto de software que será obtido no final da execução do processo de desenvolvimento.”(Sérgio Guerreiro ‘2015) Aqui são apresentados os modelos estruturado de análise e o modelo orientado a objectos, bem como o desenho arquitectural de software.

O sexto capítulo leva-nos á etapa de codificação do produto de software. Todos os aspectos relacionados com as boas práticas no código, estão aqui descritos. A importância da organização do código, da sua estruturação e documentação, bem como a gestão das versões são aqui abordadas.

No sétimo capítulo, chega a prova de fogo: a verificação e validação por testes ao produto de software. Além de explicar os conceitos, o autor detalha a questão dos diferentes testes ao produto de software, de forma a que o leitor perceba todo o caminho desde o seu desenvolvimento inicial ao seu lançamento.

No oitavo capítulo, está a manutenção do produto de software bem como tudo o que diz respeito ao acompanhamento e aperfeiçoamento do produto. Finalizando, o autor apresenta cinco casos de estudo, dos quais realço o segundo, referente à teoria dos sistemas normalizados. Após cada capítulo, podemos contar com um breve resumo do mesmo, a sua bibliografia, bem como questões que nos permitem perceber se realmente assimilamos o seu conteúdo.

Em conclusão, é um excelente material de estudo e uma boa leitura para quem está a iniciar o seu caminho na Engenharia de Software.

AUTOR



Escrito por Sara Santos

Licenciada em Docência, desenvolve de momento aplicações informáticas ligadas à educação, tendo já participado em projectos de programação e tradução de software. É membro do Staff da Revista PROGRAMAR desde Agosto de 2011, adicionalmente participando na redacção de artigos e tradução de texto para as notícias. Estuda Tecnologias e Programação de Sistemas de Informação no curso CET de TPSI no Instituto Politécnico de Viana do Castelo.

COMUNIDADES

Xamarin: O ambiente de desenvolvimento em Windows & Mac

XAMARIN: O AMBIENTE DE DESENVOLVIMENTO EM WINDOWS & MAC

Este artigo tem como objetivo mostrar como criar o ambiente de desenvolvimento cross-platform, em Windows & Mac, usando Xamarin.

Introdução

O ambiente de desenvolvimento de aplicações móveis depende da tecnologia em causa e os sistemas operativos em que estão disponíveis. Desta forma, os programadores de Objective-C terão de usar um Mac para fazer o desenvolvimento das suas aplicações, os programadores de Windows Phone e Windows Store Apps irão usar um Windows e os programadores de Android poderão fazê-lo tanto num Mac como num Windows.

Usando a Xamarin Platform este cenário muda um pouco de figura, isto porque apesar de ser obrigatório ter um Mac para o desenvolvimento de iOS, é possível desenvolver aplicações iOS num Windows. De forma a clarificar como criar o ambiente para desenvolver e testar aplicações para Android, iOS e Windows ao mesmo tempo, irei apresentar os vários cenários possíveis.

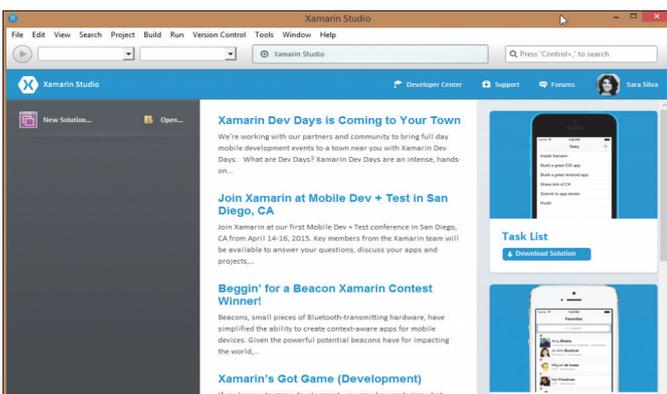
Descrição

O ambiente de desenvolvimento disponível para um programador cross-platform é o maior fator de produtividade, influencia se o programador irá tirar partido desse mesmo desenvolvimento. Isto porque, se estamos a desenvolver para as várias plataformas, com o objetivo de partilhar o máximo código possível, vai haver momentos em que é necessário testar o comportamento das aplicações nas várias plataformas ao mesmo tempo.

Ambiente Windows

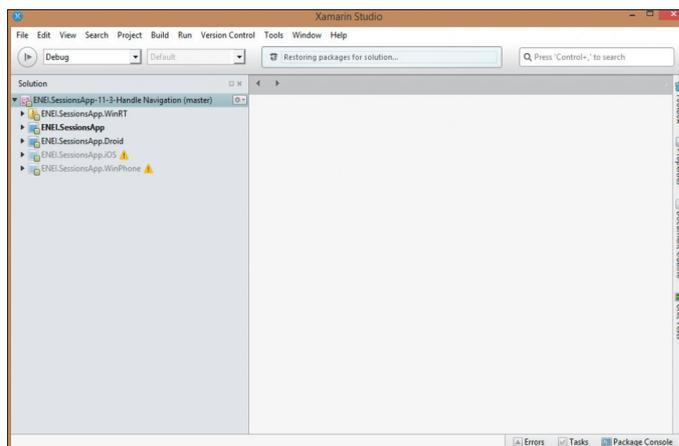
Consideremos um programador que tem disponível um Windows, neste cenário o programador poderá instalar o [Xamarin Platform](#) para Windows, que disponibiliza:

- Xamarin Studio - ferramenta para o desenvolvimento (IDE) da Xamarin



- Xamarin Android – Android SDK fornecido pela Xamarin

Ao abrir no Xamarin Studio uma solução que contenha o projeto para as várias plataformas iremos obter algo do género:



Que nos permite constatar que no Xamarin Studio não existe suporte para aplicações iOS nem para Windows.

O desenvolvimento de aplicações Windows é suportado pelas ferramentas e SDKs da Microsoft, sendo necessário instalar:

- [Visual Studio 2013 ou Visual Studio 2015](#)
- [Windows Phone 8.1 SDK and Windows SDK for Windows 8.1](#)

O licenciamento do Xamarin Platform diz respeito à Xamarin, sendo que o licenciamento do Visual Studio já diz respeito à Microsoft e também deve ser tido em conta consoante estamos perante projetos open source/ensino ou aplicações comerciais.

Abrindo a mesma solução no Visual Studio iremos ter neste momento:

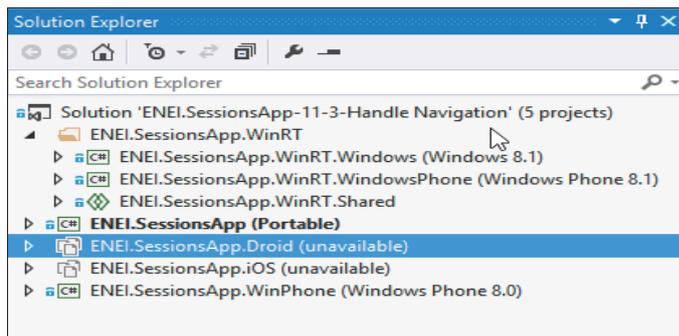


Figura 3: ENEL.SessionsApp solution no Visual Studio

XAMARIN: O AMBIENTE DE DESENVOLVIMENTO EM WINDOWS & MAC

O que nos permite concluir que apenas podemos desenvolver aplicações para Windows, uma vez que os projetos para Android e iOS não são suportados. Desta forma, com base na figura 2 e figura 3 podemos constatar, que com o plano *Free* e *Indie*, temos que andar a saltar entre IDEs para fazermos o desenvolvimento de aplicações para Windows e Android.

Com o objetivo de eliminar o problema apresentado acima, a Xamarin criou uma solução que está disponível para [estudantes](#), [programadores Open Source](#) e utilizadores do [plano Business](#):

- [Xamarin Plugin para Visual Studio](#) – plugin para dar suporte a projetos da Xamarin e permitir ligar ao Mac Build Host, e fazer deploy para simuladores e dispositivos
- Xamarin iOS – iOS SDK fornecido pela Xamarin

Ao abrirmos a solução dos casos anteriores, iremos então obter:

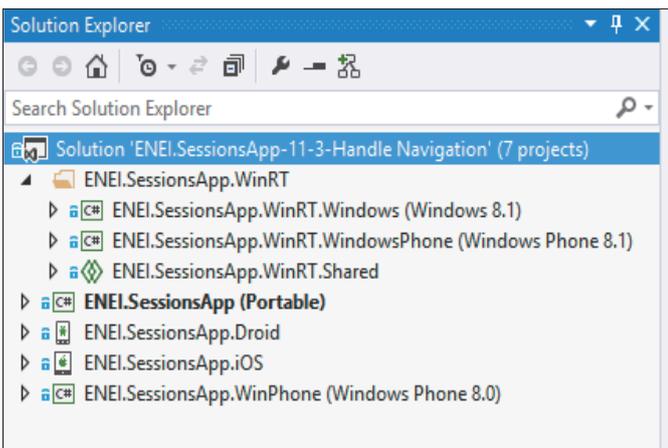


Figura 4: Suporte para Xamarin iOS e Android no Visual Studio

E desta forma, conseguimos ter compatibilidade das várias plataformas no mesmo IDE, o que sem dúvidas aumenta a produtividade do programador e a sua visão sobre o desenvolvimento cross-platform.

Como mencionamos anteriormente, é necessário ter um Mac para fazer o build e deploy de aplicações de iOS, que é designado como *Mac Build Host*.

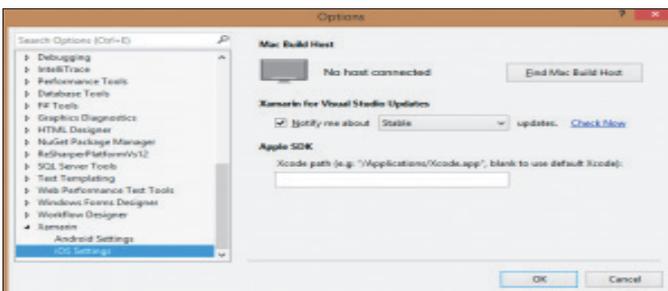


Figura 5: Sem ligação ao Host

Ao fazer *Find Mac Build Host* é necessário ter a aplicação do Build Host no Mac a correr:

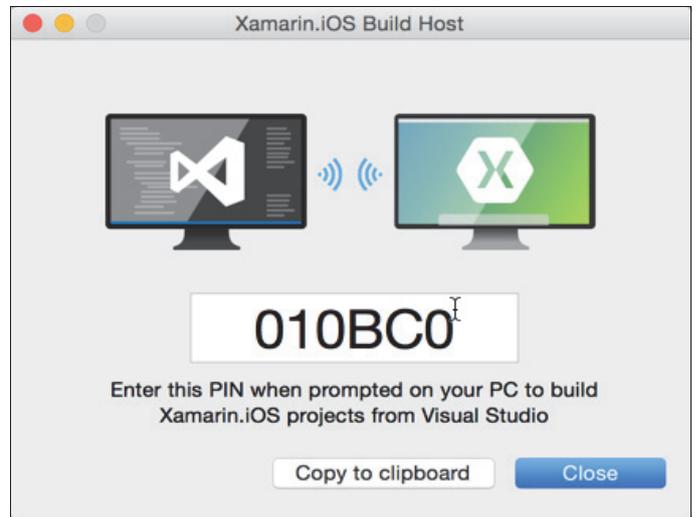


Figura 6: Aplicação Build Host

E depois da ligação feita, usando o código fornecido, iremos ter:



Figura 7: Build Host paired

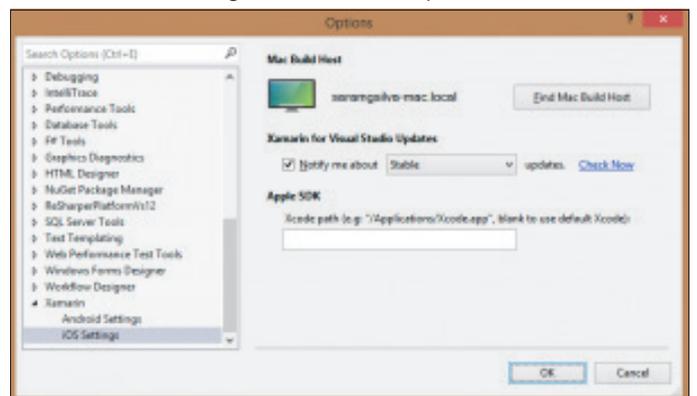


Figura 8: Ligado a um host

COMUNIDADE PTXug

<https://www.facebook.com/groups/xamarinpt/>

XAMARIN: O AMBIENTE DE DESENVOLVIMENTO EM WINDOWS & MAC

Neste caso, poderemos utilizar por exemplo um Mac Mini, para que o custo com o um Mac seja o mais baixo possível e depois só é preciso fazer um acesso remoto para usar o simulador de iOS e fazer as ligações com o Build Host. No caso de em que o programador usa dispositivo é preciso que este esteja conectado ao Mac para que seja possível fazer o deploy da aplicação.



Figura 9: Host

Ambiente Mac

Consideremos um programador que tem disponível um Mac, neste cenário o programador poderá instalar o [Xamarin Platform](#) para Mac, que disponibiliza:

- Xamarin Studio - ferramenta para o desenvolvimento (IDE) da Xamarin

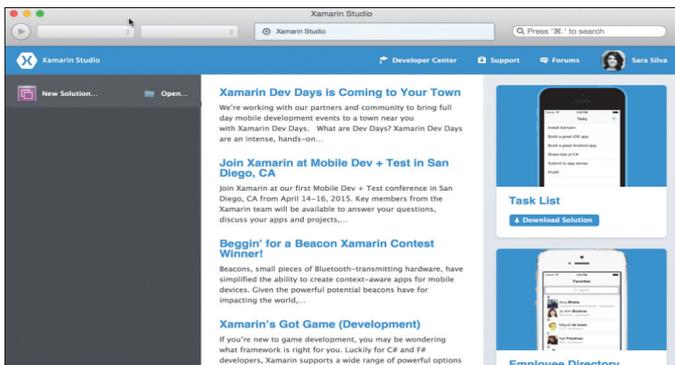


Figura 10: Xamarin Studio para Mac

- Xamarin iOS – iOS SDK fornecido pela Xamarin
- Xamarin Android – Android SDK fornecido pela Xamarin

E ao abrir a solução de exemplo, iremos obter:

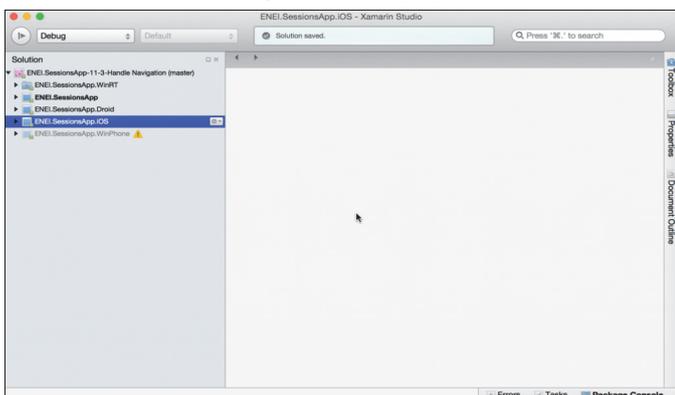


Figura 11: ENEL.SessionsApp no Xamarin Studio

Com isto é possível constatar que num Mac é possível fazer o desenvolvimentos de aplicações de Android e iOS em simultâneo, não havendo no entanto suporte para aplicações Windows que requerem um dispositivo Windows.

Ambiente Misto

Nos casos anteriores, consigo ver o ambiente para desenvolver aplicações cross-platform, em Windows e Mac, usando Xamarin. Neste cenário, o objetivo consiste em mostrar como ter um ambiente misto com um nível de produtividade maior.

Suponhamos que temos um Mac com 16GB de RAM e 512GB de disco, suponhamos também que temos VMWare ou Parallels, desta forma é possível criar uma máquina virtual com o Windows com por exemplo 8GB de Ram e 128GB de disco, permitindo desta forma ter uma performance e uso equivalente a um PC com Windows.

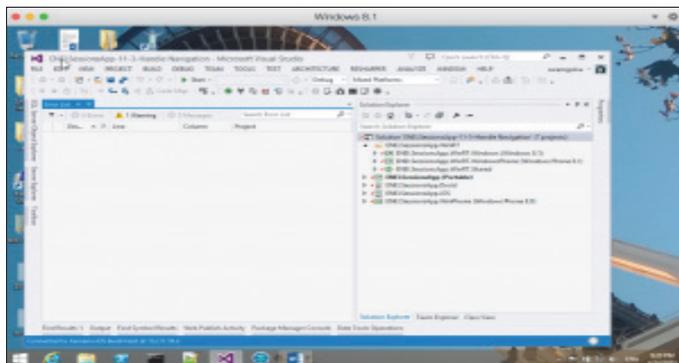


Figura 12: Windows a correr no Parallels (Mac)

O que ganhamos neste caso, é que temos um único PC com 2 sistemas operativos, e com isto temos o suporte para todas as plataformas com um conforto de trabalho elevado. No entanto, é preciso ter noção que não é muito recomendado ou até mesmo possível ter simuladores dentro de máquinas virtuais. Refiro-me ao simulador de Windows Phone e de Android, desta forma para resolver esta questão a solução passa por usar um dispositivo Windows Phone conectado à máquina virtual Windows, para que o Visual Studio reconheça o dispositivo e com a aplicação [Project My Screen App](#) é possível partilhar o ecrã do dispositivo Windows Phone. Ao nível do Android, a melhor solução passa por usar o Xamarin Android Player, a correr no Mac, que permite o Visual Studio fazer o deploy para este, usando a aplicação ADB fornecida pela Google.

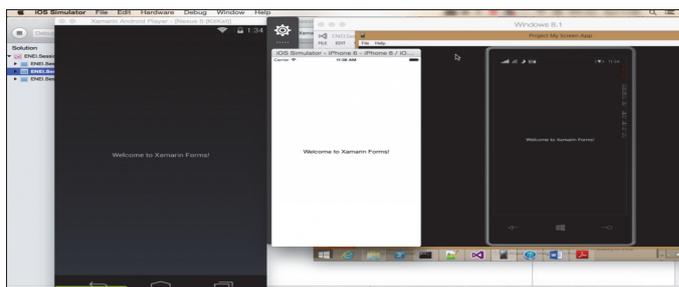


Figura 13: As várias plataformas num ambiente misto

XAMARIN: O AMBIENTE DE DESENVOLVIMENTO EM WINDOWS & MAC

Para mais informações consulta as seguintes referências da Xamarin:

- [Installing Xamarin.iOS](#)
- [Installing Xamarin.Android](#)

“ **Desta forma, os programadores de Objective-C terão de usar um Mac para fazer o desenvolvimento das suas aplicações, os programadores de Windows Phone e Windows Store Apps irão usar um Windows e os programadores de Android poderão fazê-lo tanto num Mac como num Windows (...)** ”

Simuladores

No desenvolvimento de aplicações móveis é muito comum os programadores usarem simuladores para realizar testes à aplicação, no entanto o feedback de um dispositivo real é muito diferente de um simulador e os programadores

devem realizar testes mínimos em ambos os casos.

Para desenvolvimento Windows e iOS é possível utilizar os simuladores fornecidos pela Microsoft e Apple (respetivamente), no entanto, no caso de Android os simuladores fornecidos pelo Android SDK (da Google) são muito lentos e pouco friendly, por esta razão muitos programadores têm escolhido os seguintes simuladores:

- [Xamarin Android Player](#)
- [Genymotion](#)

Que disponibilizam simuladores para vários dispositivos e resoluções.

“ **O ambiente de desenvolvimento disponível para um programador cross-platform é o maior fator de produtividade, influencia se o programador irá tirar partido desse mesmo desenvolvimento(...)** ”

Conclusão

Em conclusão, podemos concluir que o desenvolvimento cross-platform tem vantagens quando o seu ambiente de desenvolvimento está criado de forma a maximizar a rentabilidade do programador e minimizando o tempo necessário para desenvolvê-lo e testá-lo, e conseqüente irá aumentar a capacidade para aumentar o código partilhado entre as várias plataformas.

AUTOR



Escrito Por Sara Silva

Licenciada em Matemática pelo DMUC, e o seu foco de desenvolvimento está direccionado para a área Mobile. Atualmente desenvolve na área do Windows, Xamarin, Azure, e é Microsoft MVP Mentor. A Sara foi condecorada com vários prémios com especial destaque: Microsoft MVP, Xamarin MVP, Telerik Developer Especialista, C# Corner MVP, TechNet Wiki - Technical Guru. O trabalho que vai sendo desenvolvido pela Sara pode ser seguido através do seu blog www.saramgsilva.com e do twitter é @saramgsilva.

No Code

Windows 10 Mobile: Requisitos e instalação da Insider Preview

#PGGD26 – Girls in ICT Day 2015

1010 ENEI

WINDOWS 10 MOBILE: REQUISITOS E INSTALAÇÃO DA INSIDER PREVIEW

Introdução

Em fevereiro passado e na sequência dos anúncios anteriores, a Microsoft disponibilizou a primeira Preview do Windows 10 Mobile através do Windows Insider Program.

A primeira Build ainda que numa fase muito inicial de desenvolvimento, permitiu aos Windows Insiders ter uma perspetiva do comportamento do Windows 10 nestes Devices, algumas das novas funcionalidades e a nova interface. A lista inicial de dispositivos elegíveis para a instalação era limitada, contudo, em março a Microsoft alargou consideravelmente o número de Devices que suportam o Windows 10 para cerca de 30.



Na última Build disponibilizada – a 10052, para além de alguns bugs mais críticos estarem corrigidos, é possível também ver novas funcionalidades e um conjunto de novas Apps que se enquadram no novo modelo de Universal App.

Se está atualmente a testar o Windows 10 para PC, possui um Lumia e quer experimentar a Technical Preview do Windows 10 Mobile, veja quais os requisitos e procedimentos a adotar para a instalação desta Build.

Requisitos

Como em qualquer Sistema Operativo, existem um conjunto de requisitos mínimos que devem ser cumpridos para que a instalação seja possível. Estando o Windows 10 Mobile ainda em desenvolvimento, os utilizadores que queiram testar as Builds disponibilizadas terão que estar conscientes que apesar de estáveis, as mesmas poderão apresentar bugs que podem comprometer o bom funcionamento de determinadas funcionalidades do equipamento em questão. Tal como acontece com o Windows 10 versão PC, os utilizadores devem ponderar a

instalação da Preview no seu Smartphone, caso este seja o seu único dispositivo para uso diário.

Advertências feitas, os principais requisitos são:

- Possuir um Smartphone Lumia com Windows Phone 8.1 e com 1GB de espaço livre ou superior.
- Possuir uma Microsoft Account.
- Fazer o registo no Windows Insider Program.
- Fazer a instalação da App Windows Insider.

Para saber se o seu Smartphone Lumia é elegível, consulte a seguinte lista de equipamentos:

▪ Lumia 430	▪ Lumia 535	▪ Lumia 720	▪ Lumia 925
▪ Lumia 435	▪ Lumia 620	▪ Lumia 730	▪ Lumia 928
▪ Lumia 520	▪ Lumia 625	▪ Lumia 735	▪ Lumia 1020
▪ Lumia 521	▪ Lumia 630	▪ Lumia 810	▪ Lumia 1320
▪ Lumia 525	▪ Lumia 635	▪ Lumia 820	▪ Lumia 1520
▪ Lumia 526	▪ Lumia 636	▪ Lumia 822	
▪ Lumia 530	▪ Lumia 638	▪ Lumia 830	
▪ Lumia 532	▪ Lumia 640	▪ Lumia 920	

Instalação do Windows 10 Mobile

Uma vez cumpridos os requisitos e antes de iniciar o processo de instalação propriamente dito, é recomendável fazer um backup do nosso Device para o OneDrive, caso seja necessário mais tarde repor o mesmo para o Windows Phone 8.1.

Para criar um backup manualmente, deverá aceder a "**Settings – Backup – Backup Now**", do seu Windows Phone 8.1.

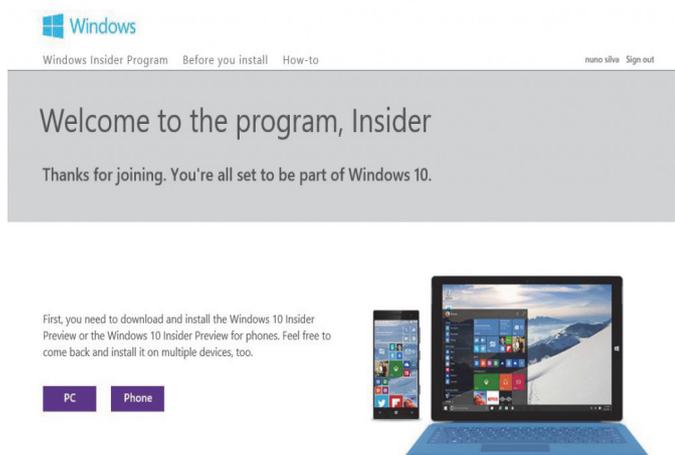
Com o passo anterior concluído, vamos aceder ao site do **Windows Insider Program** (www.insider.windows.com) e fazer o registo no programa com a nossa **Microsoft Account**.

Nota: A Microsoft Account utilizada no registo do Windows Insider Program deverá ser a mesma que temos configurada no nosso Lumia.

Uma vez registados e depois de ler os avisos, vamos escolher a opção "**Phone**" e em seguida, clicar no link "**Get the Preview**".

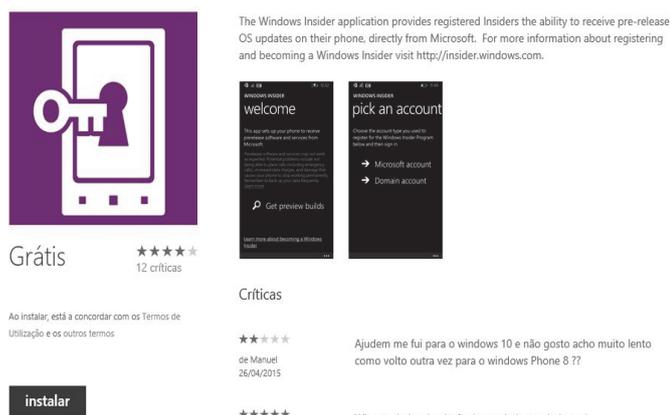
No Code

WINDOWS 10 MOBILE: REQUISITOS E INSTALAÇÃO DA INSIDER PREVIEW



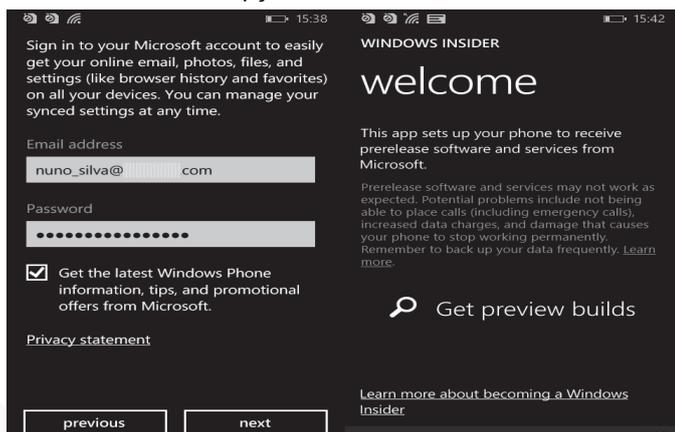
No passo seguinte vamos clicar no link **"Download the Windows Insider App"**. Já na Windows Phone Store, vamos clicar em **"instalar"** para iniciar o processo de instalação da aplicação.

Windows Insider



Em alternativa e caso já estejamos registados no Windows Insider Program, podemos fazer o download da App Windows Insider através da **Store** do Windows Phone 8.1.

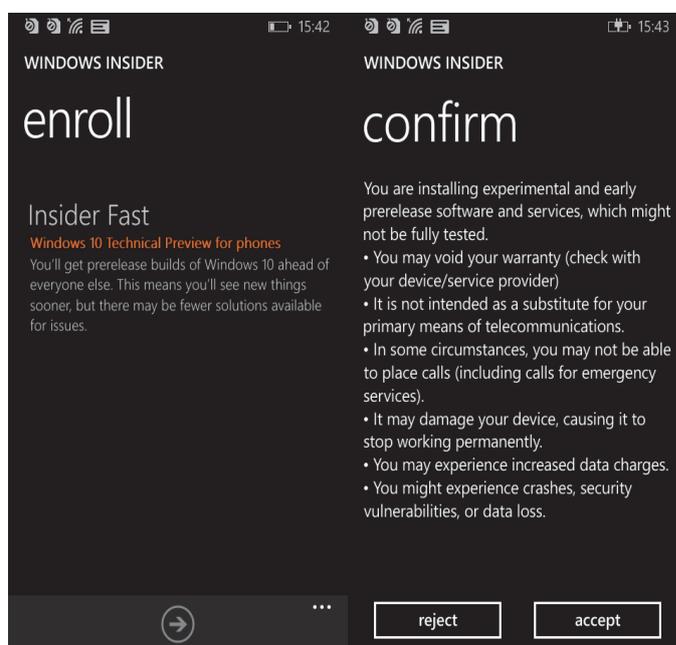
Após a instalação da **App Windows Insider**, vamos executar a mesma e fazer a autenticação com a Microsoft Account que usamos no registo do WIP. No ecrã seguinte vamos seleccionar a opção **"Get Preview builds"**.



No passo seguinte vamos indicar a frequência com que queremos receber novas Builds. Tal como acontece com o Windows 10 para PC, podemos definir o **Insider Slow** ou **Insider Fast**, sendo que o primeiro disponibiliza novas Builds mais lentamente mas com uma maior estabilidade e a segunda mais rapidamente mas com uma maior probabilidade de trazer bugs já identificados mas não corrigidos.

Após seleccionarmos um dos Rings - Insider Slow ou Insider Fast, vamos então confirmar esta opção no ecrã seguinte.

Nota: Depois da instalação, podemos voltar a executar a **App Windows Insider** e redefinir o tipo de **Ring** a utilizar para obter novas atualizações.



Após a aceitação dos termos, será necessário reiniciar o dispositivo antes de iniciar o processo de instalação do Windows 10 Mobile.

Uma vez reiniciado, vamos verificar a existência de novas atualizações. Para isso vamos a **Settings - System - Phone Update - Check for Updates**.

Alguns minutos após o início deste processo, o dispositivo encontra novas atualizações e começa a fazer o download das mesmas.

Após o download, clicamos em **install** para iniciar a instalação do Windows 10 Mobile. A instalação vai demorar alguns minutos e durante a mesma, o dispositivo vai reiniciar algumas vezes até o processo de instalação concluir. Nesta fase, "sit back and relax"!

No Code

WINDOWS 10 MOBILE: REQUISITOS E INSTALAÇÃO DA INSIDER PREVIEW



Nota: Em alguns casos, tem-se verificado que antes da atualização referente à Build 10052, é fornecido também a atualização **GDR 2** (Windows Phone 8.1 update 2). Neste caso, será necessário instalar primeiro esta atualização e em seguida de forma manual, procurar novamente por novas atualizações para então instalar o Windows 10 Mobile.

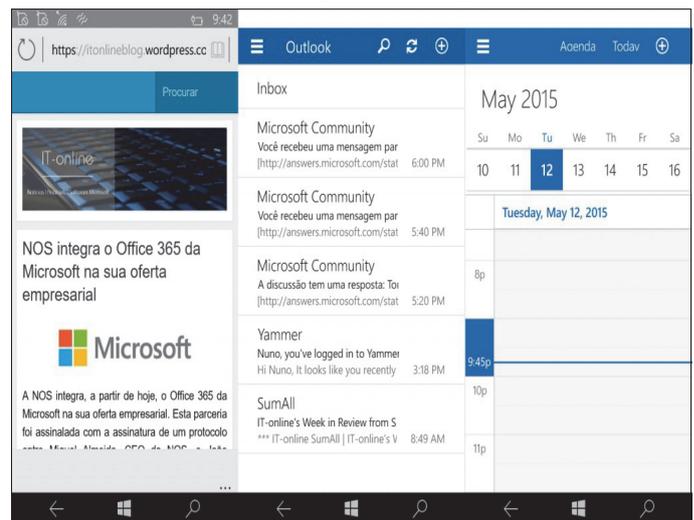
Terminada a instalação, é altura de fazer as configurações habituais como se fosse o primeiro setup ao equipamento.

Após concluídas as nossas configurações, chega então a altura de começar a explorar o Windows 10 Mobile. Aqui ficam algumas imagens:



Start Windows 10

Centro de Ações & Definições



Edge & Universal Apps (Outlook Mail, Calendar)

Feedback

A Microsoft incluiu na Technical Preview do Windows 10 Mobile, a **App Windows Feedback** que permite aos Windows Insider reportar erros, submeter sugestões e votar nas sugestões de outros utilizadores. O acesso à App pode ser feito através do **Start** ou através das teclas Volume + Power.

Para além da App, os Windows Insider podem colocar as suas questões no Microsoft Community e submeter sugestões no Windows Phone User Voice.

Rollback para Windows Phone 8.1

Os utilizadores que não queiram continuar a testar o Windows 10 Mobile ou porque a instabilidade de alguma Build não permite uma utilização normal do dispositivo, poderão fazer o rollback para o Windows Phone 8.1, através

No Code

WINDOWS 10 MOBILE: REQUISITOS E INSTALAÇÃO DA INSIDER PREVIEW

da ferramenta [Windows Phone Recovery Tool](#). Esta ferramenta vai limpar o dispositivo e instalar a última versão disponível do Windows Phone 8.1. Para saber como utilizar o Windows Phone Recovery Tool, visite o site [Microsoft Devices](#).

← Informações do telefone

Telefone



Modelo: Lumia 630 Dual SIM
Versão do firmware: 02040.00021.15053.36004
Versão do sistema operacional: 10052.12534

Software disponível no servidor



Versão do firmware: 02040.00021.15053.36004
Versão do sistema operacional: 9651.14234

Você já tem a versão mais recente do software.

Reinstalar software

“ **A Microsoft incluiu na Technical Preview do Windows 10 Mobile, a App Windows Feedback que permite aos Windows Insider reportar erros, submeter sugestões e votar nas sugestões de outros utilizadores** ”

Após a reinstalação do Windows Phone 8.1, poderão repor um backup criado anteriormente ou fazer novas configurações.

Conclusão

Em conclusão, através do Windows 10 Mobile conseguimos identificar as semelhanças com a versão PC e como a Microsoft está a trabalhar no sentido de unificar tanto o Windows, como as plataformas de desenvolvimento. Como é de esperar num Sistema Operativo em desenvolvimento, apesar de alguma estabilidade, por vezes somos confrontados com alguns bugs que impossibilitam o normal funcionamento do dispositivo.

“ (...) **através do Windows 10 Mobile conseguimos identificar as semelhanças com a versão PC e como a Microsoft está a trabalhar no sentido de unificar tanto o Windows, como as plataformas de desenvolvimento** (...) ”

Apesar dos constrangimentos que as Insider Preview podem trazer, se tem um equipamento extra e cumpre os requisitos indicados no início deste artigo, não deixe de experimentar o Windows 10 Mobile.

Bons testes.

AUTOR

Escrito por Nuno Silva

Microsoft MVP Windows Experience | Microsoft Technical Beta Tester

#PGGD26 – GIRLS IN ICT DAY 2015

No passado dia 23 de Abril a Programar marcou presença no PGGD26 - Girls in ICT Day 2015 – Lisboa.

Antes de mais, para os mais distraídos, os Girl Geek Dinners foram criados em Londres, há 10 anos atrás. Nasceram de uma necessidade da fundadora, Sarah Blow, que procurou juntar as mulheres da área da tecnologia. Em Portugal, o PGGD ou Portugal Geek Girl Dinners nasceu em 2010, pela mão da Vânia Gonçalves.



Presentes já em várias cidades portuguesas, (Braga, Guimarães, Coimbra, Leiria, Lisboa e Porto), os PGGD têm juntado cada vez mais adeptas (e adeptos, sim, porque eles também pode ir, desde que sejam acompanhantes das Geek Girls!)

Mais do que uma simples reunião, estes encontros incentivam a partilha de ideias, cimentando amizades. Sem dúvida um movimento a ter em conta.

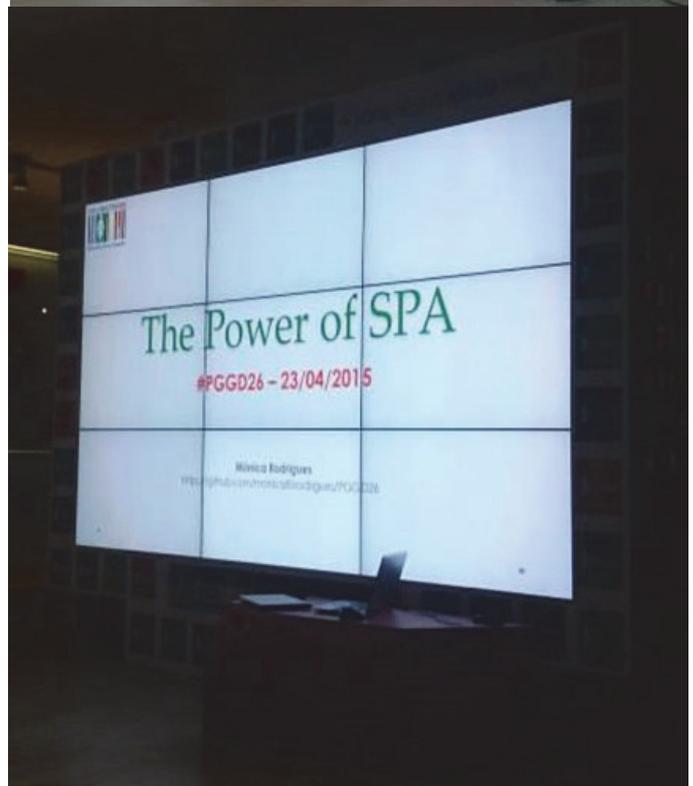
Na última sessão, o PGGD26 – Lisboa, que teve lugar na Microsoft, e pela mão de uma das nossas autoras mais residentes, a Sara Silva, a Programar foi uma das convidadas e não pode deixar de estar presente. O evento ocorreu em ambiente descontraído e podemos afirmar que quem foi, não deu o tempo como perdido. Pelo contrário, todos mostraram vontade de voltar!

O evento contou muita animação ao lanche e contou ainda com duas sessões :

- Revista Programar - A simplicidade de um projecto diferente- Rita Peres
- O Poder de uma SPA – Monica Rodrigues

As honras da casa, ficaram a cargo da Joana Barros que fez uma pequena apresentação sobre o programa Bi-

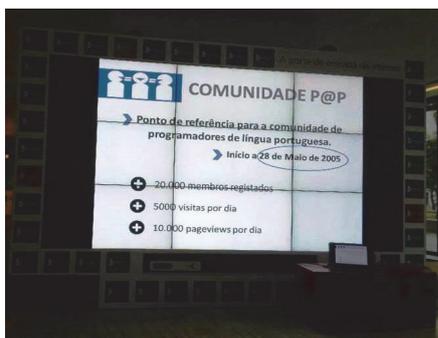
zSpark da Microsoft.



No Code

#PGGD26 – GIRLS IN ICT DAY 2015

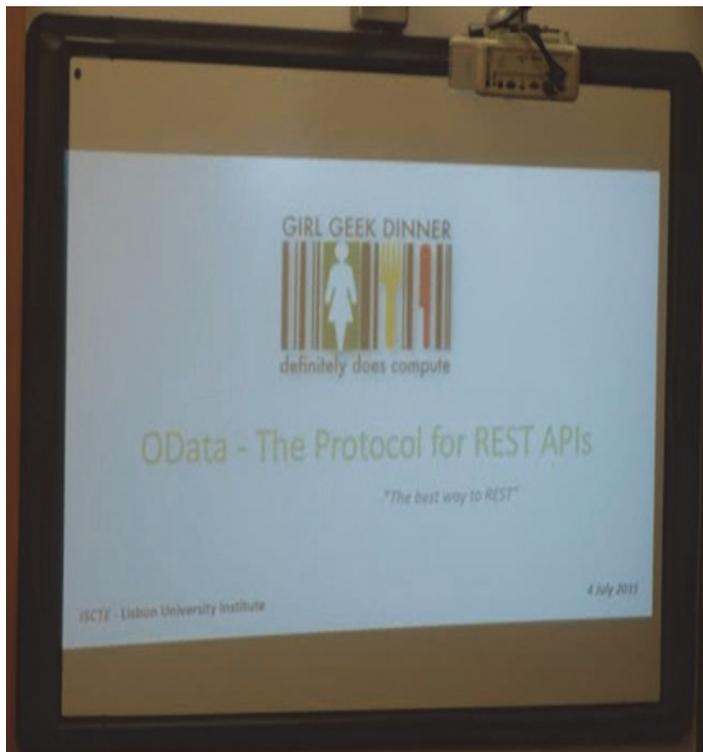
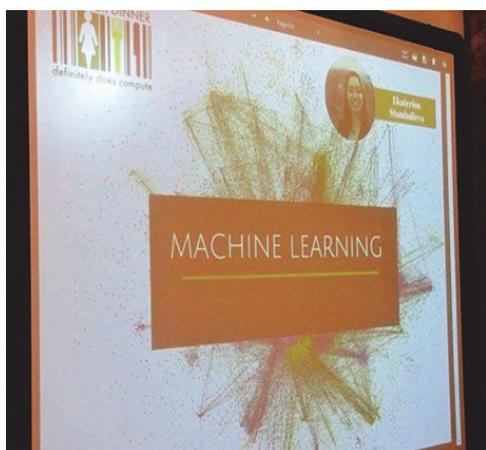
Se em 2015 o Portugal a Programar comemora a sua primeira década de existência, os Girl Geek Dinners em Portugal, comemoram em 2015, 5 anos, ou seja metade de uma década. Nós aqui, estamos a torcer por este movimento, e queremos que o PGGD dure muitos e muitos anos! De notar também que a participação nos eventos é totalmente gratuita, logo só há vantagens! Estás à espera de quê?



Apesar de termos estado presentes no evento que ocorreu em Lisboa, não queremos deixar de salientar que em dia 22 de Abril houve também um encontro em Leiria e no dia 25 de Abril, foi a vez do Porto receber a iniciativa. Portanto, quem estiver interessado é procurar o PGGD mais perto.

Queremos também dizer-vos, caros leitores, que aqui na Programar vamos continuar a acompanhar os Portugal Geek Girl Dinners, sendo que estivemos também na última edição em Lisboa, o #PGGD27 será em Lisboa, no ISCTE, no dia 4 de Junho. Contámos com mais duas sessões bastante interessantes:

- What can machine learning do for businesses? – Ekaterina Stambolieva
- OData – The Protocol for REST APIs – Cláudia Carvalho



O que aqui queremos transmitir-vos é que à semelhança da Programar, que é um projecto voluntário para todos vós que edição após edição nos lêem, o PGGD, seja qual for o grupo geográfico que vos possa acolher, é também um projecto de voluntários, e que a vossa participação é gratuita.

Tudo o que precisam é vontade de participar, aprender e partilhar conhecimento!

Apareçam!

Mais informações em: <http://portugalgirlgeekdinners.com/>

AUTOR



Escrito por Rita Peres

Natural de Castelo Branco, licenciou-se em Engenharia Informática pela Universidade da Beira Interior. Membro do P@P desde Janeiro de 2010.

1010 ENEI

Nos últimos dias de Março, Coimbra, que por si só é a cidade dos estudantes, tornou-se na cidade anfitriã e recebeu ainda mais estudantes de todo o país (e não só!) que não quiseram deixar de marcar presença no evento que todos os anos une os amantes da informática.

A Programar esteve também, a convite da organização, no 1010 ENEI. O evento ocorreu de 27 a 30 de Março e teve o apoio de algumas das maiores empresas informáticas (e não só) do nosso país.

Foram 4 dias de muitas palestras, de muitos debates, workshops e discussões que permitiram a todos os que por lá passaram, aumentarem e partilharem conhecimentos. Nenhuma temática foi esquecida e todos os ramos da informática tiveram direito à sua representação desde a mais pequena linha de programação à segurança informática mais restrita.

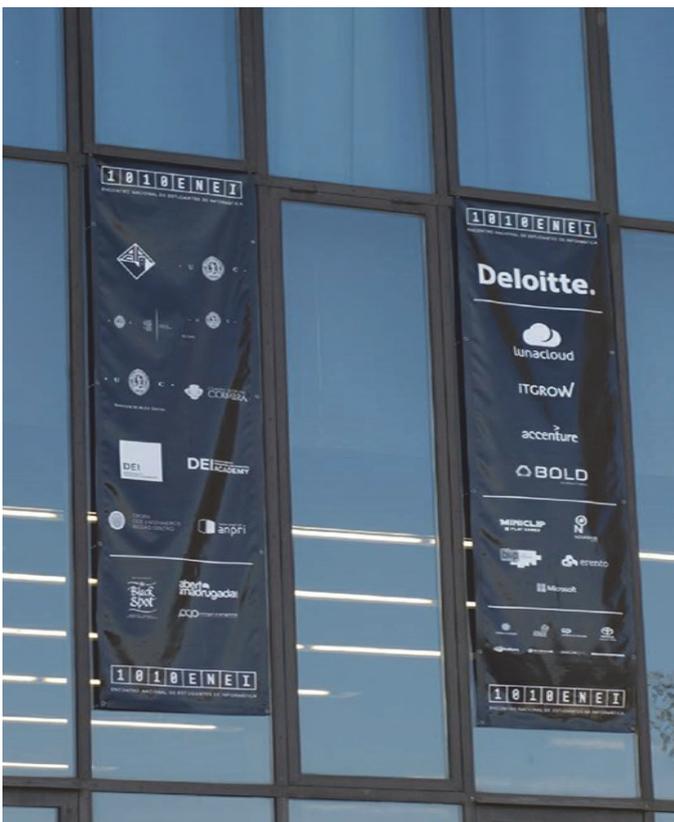
Sempre atarefados e atentos, a organização não teve mãos a medir, mas acreditamos que todas as horas de trabalho compensaram pela experiência e pelo orgulho de um dia mais tarde de dizer “Eu fiz parte da organização do ENEI”. Todos estiveram de facto de parabéns.

Podíamos referir algumas das palestras que ocorreram mas caso não as nomeássemos todas, estaríamos a ser injustos, mas podemos dizer-vos que cada kms que percorremos para marcar presença, valeu a pena! Mais palavras para quê?



A Programar quer deixar um agradecimento a toda a organização, em especial à Ana Lopes e o António Leitão que gentilmente nos endereçaram o convite.

É importante que o ENEI continue a gerar conhecimento e a fomentar as mentes mais ávidas. A partilha gera conhecimento, conhecimento gera sabedoria e experiência. Saber é poder. Nunca é demais lembrar que o saber não ocupa lugar, e que nos alarga horizontes. A edição este ano do ENEI abrilhantou o espírito do ENEI e deixou novos conhecimentos como herança. As candidaturas para organizar o próximo encontro dos estudantes de informática terminam a 5 de Junho... portanto... que venha o próximo ENEI!



AUTOR



Escrito por Rita Peres

Natural de Castelo Branco, licenciou-se em Engenharia Informática pela Universidade da Beira Interior. Membro do P@P desde Janeiro de 2010.

PROJECTO EM DESTAQUE NA COMUNIDADE P@P: MINI CALC FOR LINUX

Mini Calc for Linux é um projecto com um objectivo simples: oferecer poder de cálculo em ambiente GNU/Linux de forma minimalista. As ferramentas disponibilizadas, para além do cálculo numérico e avaliação de expressões, passam também pela análise trigonométrica, manipulação de listas, cálculo estatístico, randomização, entre outras. Podem ser adicionadas novas funcionalidades através de plugins e que se encontram neste momento em fase experimental.

O programa inclui dois modos:

- O modo interactivo, herdado da [versão para Windows](#), e
- O modo não interactivo, ideal para utilizar em scripts *bash*.

A avaliação de expressões vai muito além do mero cálculo numérico e passa mesmo pela manipulação de outros tipos de dados, como booleanos, datas e strings.

```
MC> pow(3,2) <> 9
False

MC> save.t length('texto de exemplo')
16
Saved in "t"

MC> strtodate('25/05/1956')
```

A gestão de listas, no modo interactivo, é feita através do editor visual. Este permite manipular por completo todas as listas. Podem ser geradas listas, em ambos os modos, a partir da computação de outras através do uso de fórmulas.

As funções de variável real podem gerar tabelas personalizadas. De igual forma são suportadas funções descontínuas através do uso dos operadores *is* e *is!*.

```
inunes@inunes-laptop-ubuntu:~$ mcalc "3+4*2/pow(pow(1-5,2),3)"
3.001953125
inunes@inunes-laptop-ubuntu:~$ mcalc "save.b $pi-ans"
-3.001953125
Saved in "b"
inunes@inunes-laptop-ubuntu:~$ mcalc -e "a+b-1"
-2.98730781607552
inunes@inunes-laptop-ubuntu:~$ mcalc -l show all
X: [0,2,4,7,8,10]
Y: [-6,-4,1,-2,2,1.15,2,4.05]
Y2: [36,16.81,4.84,1.3225,4,16.4025]
inunes@inunes-laptop-ubuntu:~$ mcalc +f8 "pow(x,2)/4"
Function Y8 set to 'pow(x,2)/4'.
inunes@inunes-laptop-ubuntu:~$ mcalc -f y8 -t -10 10 6 +l f8
Saving on list F8.
Being f(x) = pow(x,2)/4
-----
x          f(x)
-----
-10       25
-4         4
2          1
8          16
inunes@inunes-laptop-ubuntu:~$
```

A Mini Calc fornece ainda um sistema de memória (a qual é guardada em disco), sendo possível manipular as suas variáveis em ambos os modos. A memória da Mini Calc é única para cada utilizador do computador e é independente das variáveis que se podem criar num script *bash*. Contudo, estas podem ser utilizadas nos cálculos com a Mini Calc no modo não interactivo:

```
NUM=10
RES=$(mcalc "13+$NUM")
echo RES
```

Ainda não está disponível uma documentação oficial para a aplicação. Contudo, para uso na shell e em scripts *bash*, o modo não-interactivo fornece os comandos `--help` e `-?`. Esta aplicação foi implementada com recurso ao Free Pascal.

Mini Calc for Linux 1.2.3, by Igor Nunes

- 1 Calculator
- 2 Trigonometry >
- 3 Equations >
- 4 Lists & Stat >
- 5 Random
- 6 Plugins >
- F5 System >
- F6 Help >
- ESC Exit

```
### ## # # ##          ### ## #          ##          ##
# # # # # # # # # # # # # # # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # # # # # # # # # # # # # #
# # ## # # # ##          ##### # # ##          ##### Linux
```

The minimalist calculator

```
Y0=
Y1= pow(x,2)
Y2= 1+x
Y3=
Y4=
Y5= 1/(x+2)
Y6=
Y7=
Y8=
Y9=
-----
x          Y5
-----
-6         | -0.25
-5         | -0.333333333333333
-4         | -0.5
-3         | -1
-2         | [ERR]
-1         | 1
0          | 0.5
1          | 0.333333333333333
2          | 0.25
3          | 0.2
4          | 0.166666666666666
5          | 0.1428571428571
6          | 0.125

Up/Down = navigation
<- = clear

T = table

Esc = back to main menu

TABLE SETTINGS
Enter = edit | Esc = apply
Start = -6
Step = 1
```

Elege o melhor artigo desta edição

Revista PROGRAMAR

http://bit.do/ProgramarED49_V

Veja também as edições anteriores da Revista PROGRAMAR

48ª Edição - Março 2015



47ª Edição - Dezembro 2014



46ª Edição - Setembro 2014



45ª Edição - Maio 2014



44ª Edição - Fevereiro 2014



43ª Edição - Dezembro 2013



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

DUVIDAS?

IDEIAS?

AJUDAS?

PROJECTOS?



portugal-a-programar
•org

