

PROGRAMAR

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

EDIÇÃO #47 - DEZEMBRO 2014

ISSN 1647-0710

INTEGRAR A CORTANA NUMA APLICAÇÃO WINDOWS PHONE



A PROGRAMAR

SENCHA TOUCH PARTE 1

SANDCASTLE HELP FILE BUILDER

ED GENÉRICA

COMO SUPORTAR DIFERENTES IDIOMAS NUMA APP ANDROID

REGISTO DE TODOS OS COMANDOS EXECUTADOS NUM SISTEMA INFORMÁTICO

ANÁLISE

DE APLICAÇÕES PROFISSIONAIS EM ANDROID+ **DESENVOLVIMENTO**

NO CODE

ÉS TU **QUE TIPO DE BD**

PASSA A SER GRATUITA PARA ESTUDANTES **PLATAFORMA XAMARIN**

EQUIPA PROGRAMAR

Coordenador

António Pedro Cunha Santos

Editor

António Pedro Cunha Santos

Design

Sérgio Alves

Twitter: @scorpion_blood

Ilustração

Sara F. Santos

Redacção

António Santos

Fernando Cesar

Nelson Silva

Nuno Santos

Nuno Silva

Paulo Afonso Pereira Junior

Sara Paiva

Sara Silva

Sara F. Santos

Sérgio Ildefonso

Staff

António Pedro Cunha Santos

Rita Peres

Rui Gonçalves

Sara F. Santos

Contacto

revistaprogramar@portugal-a-programar.org

Website

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

ISSN

1 647-071 0

Lorem ipsum

Os anos passam, mais um ano termina, e esta edição é a última do ano, com toda a nostalgia que isso possa trazer. Não obstante, o passar do tempo, significa que se evoluiu, se cresceu, se acrescentou algo, se viveu, se aprendeu, se mudou, sem esquecer de onde se veio.

Nesta edição escolhi para título o Lorem ipsum, pelo contexto histórico do texto normalmente conhecido por Lorem ipsum, mais concretamente o seu contexto ligado à tecnologia. Há algumas gerações atrás, um dos mais conhecidos softwares de desktop publishing, disponível para Macintosh e Windows, chamado [PageMaker](#), da Aldus Corporation, fazia as delícias de jornalistas, escritores, autores, entusiastas e demais público, que vivia “fascinado” com o grande salto tecnológico que se vivia na década de oitenta do século passado, apresentava como texto padrão para alinhamento e formatação: o Lorem ipsum. Este texto não é mais do que um misturar de palavras em grego, de um texto do século primeiro antes de Cristo, intitulado “Finibus Bonorum et Malorum “ (A Origem do Bem e do Mal), escrito por Cícero, adaptado por Richard McClintock, como resultado de uma pesquisa que fez, de uma das palavras mais obscuras do latim (consectetur), criando um texto que não faça sentido algum, seja ele sintático ou gramatical. Este texto é ainda nos dias de hoje usado como texto padrão para ajuste de dimensões durante as fases de paginação de obras de diversos tipos.

Saindo um pouco do contexto histórico do “tratado” de onde vem o comum “Lorem ipsum”, esta escolha prendeu-se com uma sincera falta de inspiração para escrever o editorial! Poderá ser uma branca, poderá ser uma “falha”, poderá ser um milhão de coisas que não vos sei explicar, mas certamente que todos nós programadores, autores, redactores, Humanos, passamos por este tipo de “brancas” de vez em quando, como se “perdessemos” os apontadores, ou os índices de uma tabela de alocação e deixássemos por momentos de conseguir escrever.

Nessa falta de inspiração pensei o que escrever e para não deixar literalmente o texto: “*Neque porro quisquam est qui dolorem ipsum quia dolor sit amet consectetur adipisci velit*” escrevo este breve editorial recordando o texto padrão, e a tecnologia que o “imortalizou” e que agora é parte da história comum de todos nós (a história da informática).

Sendo o texto original de Cícero, um “tratado” muito usado na renascença, na pesquisa do saber do conhecimento, da inovação, da busca de novos horizontes e conhecimentos e em virtude de ser quase final de ano, aproveito para desejar a todos os que lêem, participam, colaboram, trabalham, nesta nossa e vossa publicação que é a Revista PROGRAMAR, os votos de um feliz criativo 2015, cheio de tudo aquilo de bom que nos é querido, de inovações e alegrias, de evolução e conhecimento, na busca de todos que é a Felicidade!

Até à próxima 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

- [7](#) Integrar a Cortana numa aplicação Windows Phone - **(Sara Silva)**

A PROGRAMAR

- [13](#) Estruturas de Dados (EDs) Genéricas - **(Paulo Afonso Pereira Júnior)**
- [19](#) Sencha Touch - Parte I - **(Nelson Silva)**
- [22](#) Registo de todos os comandos executados num Sistema Informático - **(Fernando César)**
- [27](#) Como suportar diferentes idiomas numa App Android - **(Nuno Santos)**
- [31](#) Sande Castle Help File Builder - **(Sérgio Ildefonso)**

ANÁLISES

- [36](#) Desenvolvimento de Aplicações Profissionais em Android - **(Sara Paiva)**
- [38](#) Projetos de POO em JAVA - **(António Santos)**

NO CODE

- [41](#) Windows 10: A Nova Geração Windows - **(Nuno Silva)**
- [45](#) Xamarin: Sem Custos para Estudantes - **(Sara Silva)**
- [47](#) Que tipo de BD és tu - **(Sara Santos)**
- [49](#) Xamarin Store App: Ganha a tua t-shirt C# ou F# - **(Sara Silva)**
- [50](#) Projecto em Destaque na Comunidade p@p: IRIS NOS Control Panel

EVENTOS

2 a 6 de Março 2015 SET Semana Empresarial e Tecnológica, Instituto Superior Técnico

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

App portuguesa JiTT guia turistas em todo o mundo

Paris, Viena, Boston, Los Angeles, Nova Iorque, Pequim, Rio de Janeiro ou São Paulo fazem parte do grupo de 18 cidades que pode visitar com a ajuda do JiTT, um guia turístico made in Portugal pensado para quem conhecer os quatro cantos do mundo.

Neste momento, só na App Store da Apple o JiTT tem 57 aplicações, mas até ao final do ano irão superar uma centena, graças à disponibilização das versões para Android. Cada aplicação corresponde a um binómio cidade/idioma.



JiTT Paris

"As nossas aplicações são de instalação gratuita e na versão gratuita entregam ao utilizador um conjunto de funcionalida-

des (...) como mapas offline com os pontos de interesse assinalados ou uma short version dos conteúdos escritos", referiu ao TeK Alexandre Pinto, CEO da iClio, empresa de Coimbra especializada na conceção e edição de conteúdos nas áreas da História, Património e Cultura responsável pelo guia turístico.

A aplicação tem depois conteúdos premium que podem ser adquiridos (in app) que dão acesso, entre outras coisas, ao áudio e ao planeador de percursos (Route Planner).

Até ao final de 2014 deverão estar disponíveis guias JiTT para 18 cidades mundiais em cinco idiomas diferentes na App Store, na Google Play, na Aptoide e nas 11 maiores lojas Android na China.

Em meados de novembro as aplicações JiTT tinham "dezenas de milhar de downloads", segundo o responsável. "Atendendo ao facto de que estamos em plena fase de publicação de novas cidades em diferentes idiomas apenas na App Store, fase que iniciámos no final de setembro, deixa-nos muito confiantes para futuro".

A primeira cidade a ser lançada com o JiTT foi Barcelona, em fevereiro de 2011 tendo muito recentemente sido fechado um acordo na China com vista ao desenvolvimento de um guia à medida para a cidade de Guilin.

"Os mercados norte-americano e chinês são os dois principais mercados naturais do JiTT, mas igualmente a Europa assume grande relevância na nossa estratégia", sublinha Alexandre Pinto. "As cidades que elegemos como prioritárias na nossa estratégia são selecionadas, de grosso modo, com base na análise dos fluxos mundiais de turistas. A análise é um pouco mais profunda e contempla análises mais verticais em segmentos de mercado que, para nós, são fundamentais".

Depois de cidades como Londres, Barcelona, Paris, Munique, Berlim, Roma, Berlim, Viena, São Francisco, Boston, Los Angeles, Nova Iorque, Pequim, Rio de Janeiro ou São Paulo, Alexandre Pinto garante que existem locais no top das cidades mais visitadas que o guia turístico vai integrar no futuro. "Tóquio e Seul são imediatamente percebidas, mas mais estão na 'mira' do JiTT".

O CEO da iClio garante que a mais valia do JiTT face a aplicações do mesmo género é o facto de os conteúdos serem originais, "escritos por historiadores que, para lhe introduzir particularidades diferenciadoras, procedem a uma pesquisa prévia em cada cidade para conhecer aspetos novos ou algo que está na moda".

Escrito ao abrigo do novo Acordo Ortográfico

In Sapo Tek

Cão corre graças a próteses impressas em 3D

Em ROCK HILL, Carolina do Sul, 16 de Dezembro de 2014 – A 3D Systems anunciou hoje que as próteses 3D serviram com sucesso ao cão Derby, possibilitando-lhe correr ao longo da rua pela primeira vez. Derby nasceu com uma deformação congénita caracterizada por braços encurtados e ausência de patas dianteiras. Apesar de sempre alegre, até agora o Derby apenas podia andar em superfícies macias. As superfícies duras como passeios, causavam abrasões severas às suas extremidades dianteiras.

Tendo adotado Derby através de um grupo de resgate de cães, Tara Anderson decidiu ajudar. Tara, funcionária da 3D Systems, sabia que a impressão 3D permitia um nível inigualável de liberdade de design, funcionalidade e rapidez. Utilizando a tecnologia 3D, ela sabia que seria possível desenhar e construir rapidamente próteses personalizadas para a morfologia do Derby.

Com a ajuda de Derrick Campana, Ortopedista, e dos designers 3D Kevin Atkins e Dave DiPinto, dados das patas de Derby e scans 3D do design de um copo, criados por Campana, foram utilizados para criar o design 3D. A equipa utilizou o Geomagic Freedom, uma plataforma de escultura digital 3D, que lhes permitiu criar formas orgânicas perfeitas e curvas suaves para a forma de Derby.

A ProJet 5500X produz material variado de impressão 3D numa construção única. Assim Tara e os designers puderam construir próteses completas com adaptações confortáveis em borracha e bases rígidas. Prontas em apenas algumas horas, as próteses foram enviadas para Derby para teste.(...)

Através do poder do 3D, o Derby pode agora correr a acompanhar e por vezes ultrapassar os seus recentes donos adotivos Sherri e Dom Portanova.(...)

In 3dsystems.com

Tradução: Sara Santos



Windows 10 para o mercado de consumo vai integrar de raiz a Cortana

A Microsoft está a preparar um evento para janeiro, no qual vai dar a conhecer as novidades presentes no Windows 10 para o mercado de consumo – depois de o ter feito para o segmento empresarial no final de setembro. O The Verge, teve acesso a uma fuga de informação sobre o que vai ser divulgado e revela as duas principais novidades.

A primeira delas é a integração da Cortana no Windows 10, com a assistente digital a assumir a função de interface de pesquisa no novo sistema operativo, respondendo a comandos de texto e de voz. Para já, é tudo muito similar à versão do Windows Phone, mas é muito provável que a versão final ainda venha a sofrer alterações.

A outra grande novidade é a app para Xbox, que parece funcionar como uma espécie de porta de entrada para todo o ecossistema Xbox, incluindo listas de amigos, feeds de atividades e loja.

Além disso, a Windows Store também sofreu algumas alterações, pelo que a música e os filmes/programas de televisão estarão localizados junto aos jogos e às apps – algo que poderá indicar que a Microsoft poderá adicionar os conteúdos de áudio e de vídeo das lojas da Xbox à Windows Store, passando a disponibilizar toda a sua oferta a partir de um único local.

In Exame Informática

Miúdos portugueses vencem concurso de programação da Microsoft

Gonçalo e Matias têm seis e sete anos e conquistaram o primeiro lugar na categoria Conquerors na Final Europeia do Kodu Kup, uma competição de programação organizada pela Microsoft.

A Kodu Kup reuniu mais de 20 alunos, dos seis aos 16 anos, em Bruxelas para estimular os primeiros passos no mundo da programação e da criação de jogos digitais. A Comissão Europeia reconhece a competição como uma das melhores práticas para desenvolvimento de competências digitais.

Gonçalo Calçadas e Matias Fareleira criaram um jogo chamado Water Planet sobre a importância da água e da biodiversidade e que retrata os perigos da poluição e destruição do ecossistema, encarnados em peixes robô e polvos mecânicos. O jogo foi desenvolvido nas salas de aula do Colégio Monte Flor, em Carnaxide, usando o Kodu, a ferramenta pedagógica de programação visual da Microsoft.

Rui Lima, o professor que acompanhou o projeto, explica que «isto mostra-nos a importância que as tecnologias de informação assumem enquanto ferramentas pedagógicas, que estimulam o conhecimento e o potencial de cada aluno. A integração de ferramentas tecnológicas é já uma realidade no Colégio Monte Flor e tem sido muito gratificante ver o impacto no sucesso escolar dos alunos e na criação de projetos desta natureza e na sua distinção à escala global», cta o comunicado da Microsoft.

O Kodu Kup Europe é da responsabilidade da Microsoft Europe e desenvolvido em parceria com o Centro de Inovação da Microsoft em Bruxelas.

In Exame Informatica

TEMA DE CAPA

Integrar a Cortana numa aplicação Windows Phone

Integrar a Cortana numa aplicação Windows Phone

Integrar a Cortana numa aplicação Windows Phone

Este artigo tem como objetivo apresentar um exemplo de como integrar a [Cortana](#) com uma aplicação de Windows Phone.

Introdução

Uma das funcionalidades interessantes no Windows Phone 8.1 é a [Cortana](#). A [Cortana](#) não é mais do que uma assistente pessoal, que ajuda os utilizadores em tarefas básicas, tais como ligar a um amigo, marcar na agenda um acontecimento e outras tarefas.

Atualmente não está disponível para todas as línguas, por esta razão muitos utilizadores que não são nativos na língua inglesa mudaram as definições do seu dispositivo para que este suporte esta funcionalidade, como é o caso de muitos portugueses. Para os interessados que ainda não o tenham feito, é recomendado que alterem as definições para Inglês e para Estados Unidos, para mais informações consultar o [artigo](#).

De suporte ao artigo iremos usar o exemplo [Menu App](#) criado no [Windows AppStudio](#).

Integrando a Cortana

A [Cortana](#) irá usar [comandos de voz](#) para interagir com as aplicações. Para isso esses comandos de voz tem que ser instalados por cada aplicação (cada vez que a aplicação inicia), para que a Cortana saiba como poderá interagir com cada aplicação.

Desta forma, o primeiro passo para integrar com a [Cortana](#), na [Menu App](#), passa por definir um ficheiro [Voice Command Definition \(VCD\)](#), este ficheiro não é mais do que um ficheiro xml com os vários comandos que a [Cortana](#) irá reconhecer e por sua vez permitirá fazer a ligação com a aplicação.

Para a [Menu App](#), iremos definir "Menu" como o nome que a Cortana irá usar para iniciar a aplicação e iremos definir dois comandos:

- **"Show Command"** - irá ser o comando para apresentar ecrãs específicos da aplicação, para a [Menu App](#) poderá ser os ecrãs Beverages, Starters, Mains, Desserts and Special Offers;
- **"Natural Language Command"** - irá permitir que a [Cortana](#) reconheça expressões comuns do ser huma-

no, tais como "I am hungry", "I want to eat" and "I want to drink".

Como foi mencionado anteriormente a [Cortana](#) não está disponível em todas as línguas e atualmente não é possível interagir com a Cortana em português, por esta razão se está a usar expressões em inglês.

De seguida iremos ver um exemplo de um VCD, que estará definido para inglês. No entanto, é possível definir para outras línguas que a Cortana suporte.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  Be sure to use the new v1.1 namespace to utilize
  the new PhraseTopic feature -->
<VoiceCommands xmlns="http://schemas.microsoft.com/
  voicecommands/1.1">
  <!--
    The CommandSet Name is used to programmatically
    access the CommandSet -->
  <CommandSet xml:lang="
    en-us" Name="englishCommands">
    <!-- The CommandPrefix provides
      an alternative to your full app
      name for invocation -->
    <CommandPrefix>Menu</CommandPrefix>
    <!-- The CommandSet Example appears
      in the global help alongside your app name -->
    <Example> I am hungry </Example>

    <Command Name="ShowCommand">
      <Example> Show Mains </Example>
      <ListenFor> Show {dictatedShowTerms}
      </ListenFor>
      <Feedback> Showing in Menu ...
      </Feedback>
      <Navigate Target="MainPage.xaml" />
    </Command>

    <Command Name="NaturalLanguageCommand">
      <Example> I want to eat </Example>
      <ListenFor> {naturalLanguage}
      </ListenFor>
      <Feedback> Starting Menu ...
      </Feedback>
      <Navigate Target="MainPage.xaml" />
    </Command>

    <PhraseTopic Label="dictatedShowTerms"
      Scenario="Search">
      <Subject> Starters </Subject>
      <Subject> Mains </Subject>
      <Subject> Desserts </Subject>
      <Subject> Beverages </Subject>
      <Subject> Special Offers </Subject>
    </PhraseTopic>

    <PhraseTopic Label="naturalLanguage"
      Scenario="Natural Language">
      <Subject> I want to eat </Subject>
    </PhraseTopic>
  </CommandSet>
</VoiceCommands>
```

TEMA DA CAPA

INTEGRAR A CORTANA NUMA APLICAÇÃO WINDOWS PHONE

```
<Subject> I want to drink </Subject>
<Subject> I am hungry </Subject>
</PhraseTopic>
</CommandSet>
</VoiceCommands>
```

De salientar que

- CommandSet - define os comandos para cada língua
- CommandPrefix - define o nome com que a [Cortana](#) irá reconhecer a aplicação, neste caso "Menu"
- Example - define um exemplo de como usar a [Cortana](#) para interagir com a [Menu App](#).
- Command - define o comando de voz suportado
- PhraseTopic - define possíveis expressões que a [Cortana](#) poderá identificar para iniciar o comando de voz

Para mais detalhes consulte o artigo

[Voice Command Definition \(VCD\) elements and attributes](#)

Nota: No ficheiro Manifest é necessário seleccionar a capacidade de "Microphone".

Neste momento temos o ficheiro VCD definido e portanto temos que o instalar no dispositivo (e cada vez que a aplicação inicia). Para isso iremos criar o seguinte método:

```
private async Task InstallVoiceCommandsAsync()
{
    var storageFile = await StorageFile.
        GetFileFromApplicationUriAsync(new
            Uri("ms-appx:///Cortana.xml"));

    await VoiceCommandManager.
        InstallCommandSetsFromStorageFileAsync
            (storageFile);
}
```

De seguida devemos chamar este método no método `OnNavigationTo` da `MainPage`, da seguinte forma:

```
protected override async void OnNavigatedTo
(NavigationEventArgs e)
{
    _dataTransferManager = DataTransferManager
        .GetForCurrentView();
    _dataTransferManager.DataRequested +=
        OnDataRequested;
    _navigationHelper.OnNavigatedTo(e);
    await MainViewModel.LoadDataAsync();
    if (e.NavigationMode == NavigationMode.
        New)
    {
        await InstallVoiceCommandsAsync();
    }
}
```

Por fim, temos que alterar o método `OnActivated` no `App.xaml.cs` para que a interação da [Cortana](#) com a [Menu](#)

[App](#) seja concretizada. Portanto, quando a [Cortana](#) identifica um comando de voz para a [Menu App](#) o método `OnActivated` será chamado e irá enviar como parâmetro um `SpeechRecognitionResult` que nos permite obter o comando de voz e o texto reconhecido pela [Cortana](#).

```
protected override void OnActivated
(IActivatedEventArgs args)
{
    base.OnActivated(args);

    if (args.Kind == ActivationKind.
        VoiceCommand)
    {
        var commandArgs = args as
            VoiceCommandActivatedEventArgs;
        if (commandArgs != null)
        {
            SpeechRecognitionResult
                speechRecognitionResult = commandArgs.Result;

            var voiceCommandName =
                speechRecognitionResult.RulePath[0];
            var textSpoken =
                speechRecognitionResult.Text;

            switch (voiceCommandName)
            {
                case "ShowCommand":
                    if (textSpoken.ToLower()
                        .Contains("starters"))
                    {
                        RootFrame.Navigate
                            (typeof(StartersPage));
                    }
                    if (textSpoken.ToLower()
                        .Contains("mains"))
                    {
                        RootFrame.Navigate
                            (typeof(Main1Page));
                    }
                    if (textSpoken.ToLower()
                        .Contains("desserts"))
                    {
                        RootFrame.Navigate
                            (typeof(DessertsPage));
                    }
                    if (textSpoken.ToLower()
                        .Contains("beverages"))
                    {
                        RootFrame.Navigate
                            (typeof(BeveragesPage));
                    }
                    if (textSpoken.ToLower()
                        .Contains("special") ||
                        textSpoken.ToLower()
                        .Contains("offer"))
                    {
                        RootFrame.Navigate
                            (typeof(MainPage), "SpecialOffers");
                    }
                    break;
                case "NaturalLanguageCommand":
                    if (textSpoken.ToLower()
                        .Contains("eat") ||
                        textSpoken.ToLower()
                        .Contains("hungry"))
                    {
                        RootFrame.Navigate
                            (typeof(Main1Page));
                    }
            }
        }
    }
}
```


TEMA DA CAPA

INTEGRAR A CORTANA NUMA APLICAÇÃO WINDOWS PHONE

```
    }  
    if (textSpoken.ToLower()  
        .Contains("drink"))  
    {  
        RootFrame.Navigate  
(typeof (BeveragesPage));  
    }  
    if (textSpoken.ToLower()  
        .Contains("special"))  
    {  
        RootFrame.Navigate  
(typeof (MainPage), "SpecialOffers");  
        break;  
    }  
    }  
    }  
    Window.Current.Activate();  
}
```

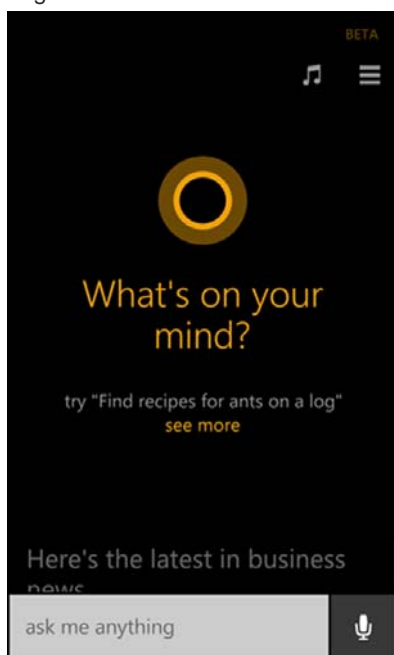
Para cada comando de voz definido no ficheiro VCD devemos definir qual a informação que iremos apresentar. Por exemplo, quando o utilizador diz "I want to drink" irá ser apresentado o ecrã BeveragesPage.xaml que irá apresentar as bebidas disponíveis.

Para quem tiver interesse em alterar dinamicamente o ficheiro Voice Command Definition (VCD) é recomendado a leitura deste [artigo](#).

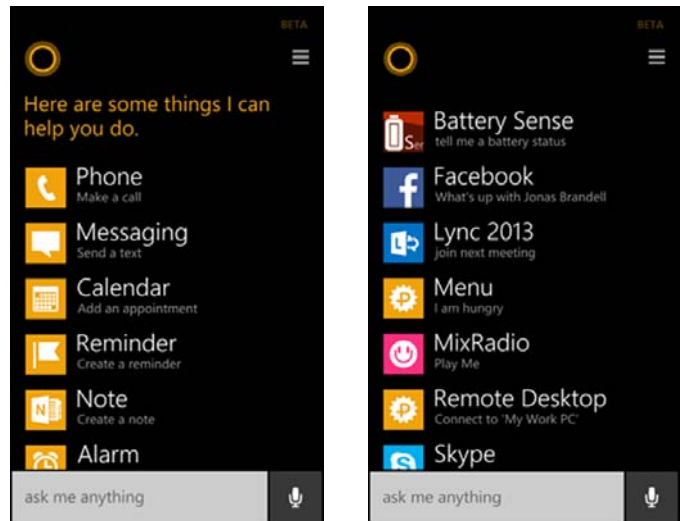
Usando a Cortana

Antes de usarmos a [Cortana](#), temos que instalar e correr a [Menu App](#) no dispositivo, para que o ficheiro VCD seja instalado e depois disto podemos usar a [Cortana](#) para interagir com a [Menu App](#).

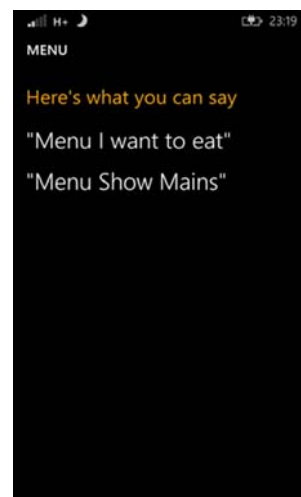
Quando iniciamos a aplicação [Cortana](#) no Windows Phone, iremos ver o seguinte ecrã:



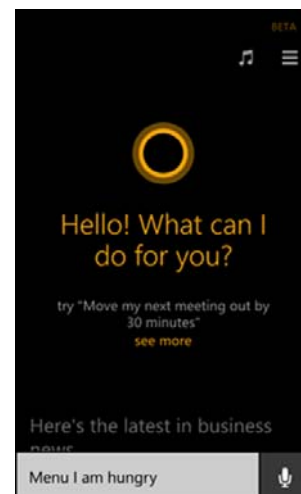
E ao clicar em "see more" (ver mais) iremos obter os ecrãs:



Os quais irão ajudar a perceber de que forma podemos usar a [Cortana](#) assim com interagir com ela. Clicando na [Menu App](#) iremos ver mais exemplos (que foram definidos no ficheiro VCD).



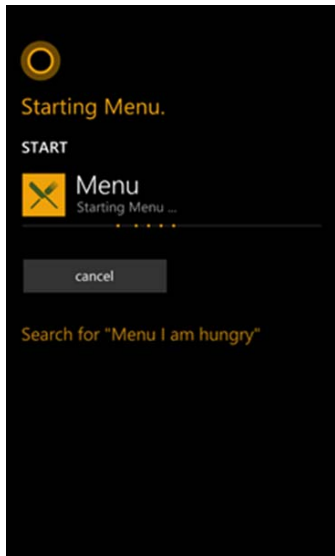
Voltando ao ecrã inicial da [Cortana](#) e introduzindo o texto ou usando a voz para dizer "Menu I am hungry" iremos obter o ecrã:



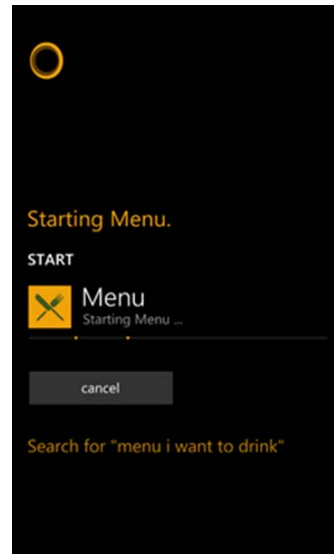
TEMA DA CAPA

INTEGRAR A CORTANA NUMA APLICAÇÃO WINDOWS PHONE

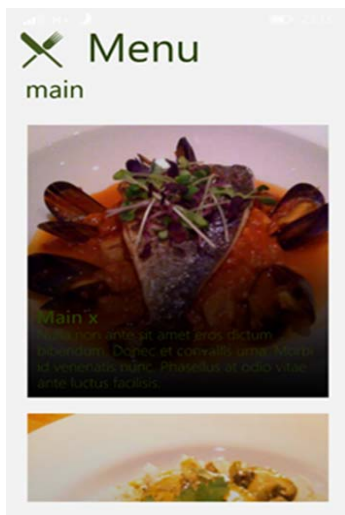
Por sua vez a [Cortana](#) irá inicializar a [Menu App](#):



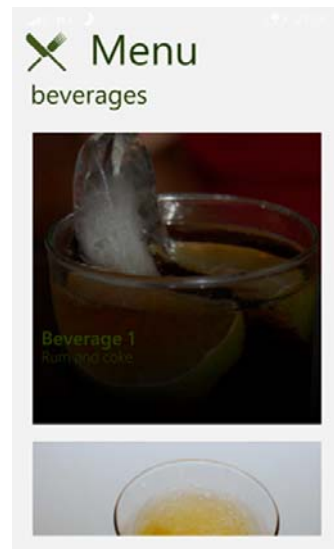
Por sua vez a [Cortana](#) irá inicializar a [Menu App](#):



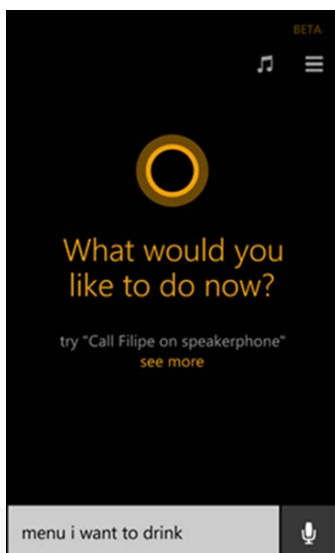
Sendo então apresentado o ecrã dos pratos principais:



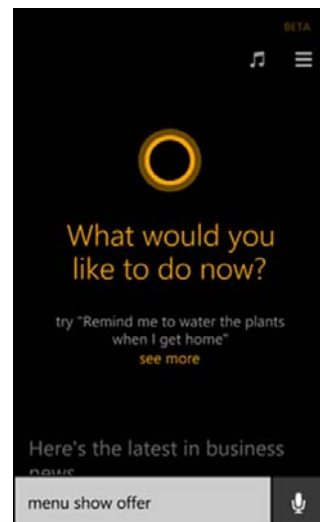
E então será apresentado o ecrã das bebidas:



Quando mencionamos "Menu I want to drink" iremos ter o ecrã:



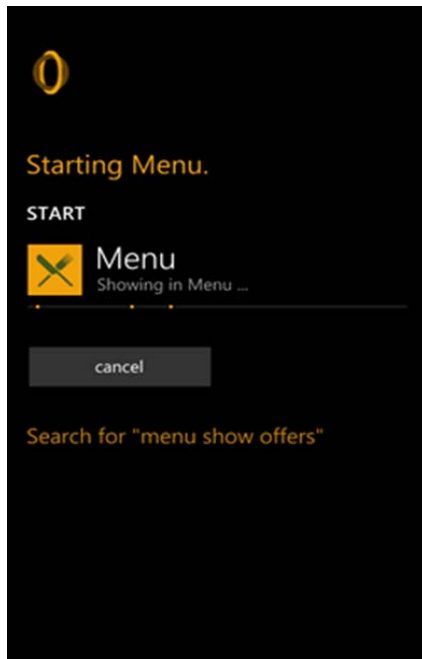
Quando mencionamos "Menu Show offers" iremos ter o ecrã:



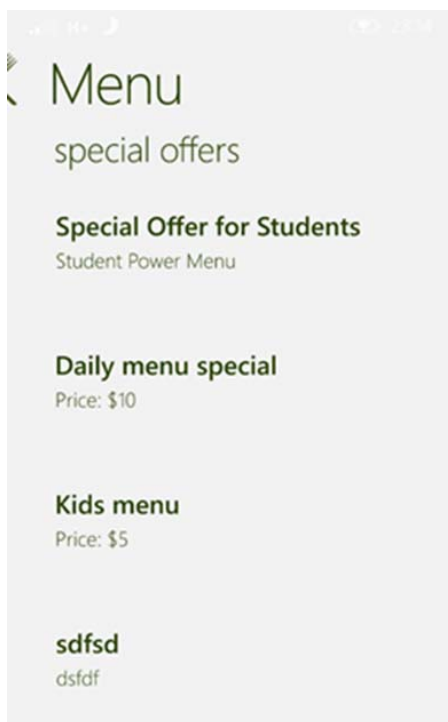
TEMA DA CAPA

INTEGRAR A CORTANA NUMA APLICAÇÃO WINDOWS PHONE

Que por sua vez a [Cortana](#) irá inicializar a [Menu App](#):

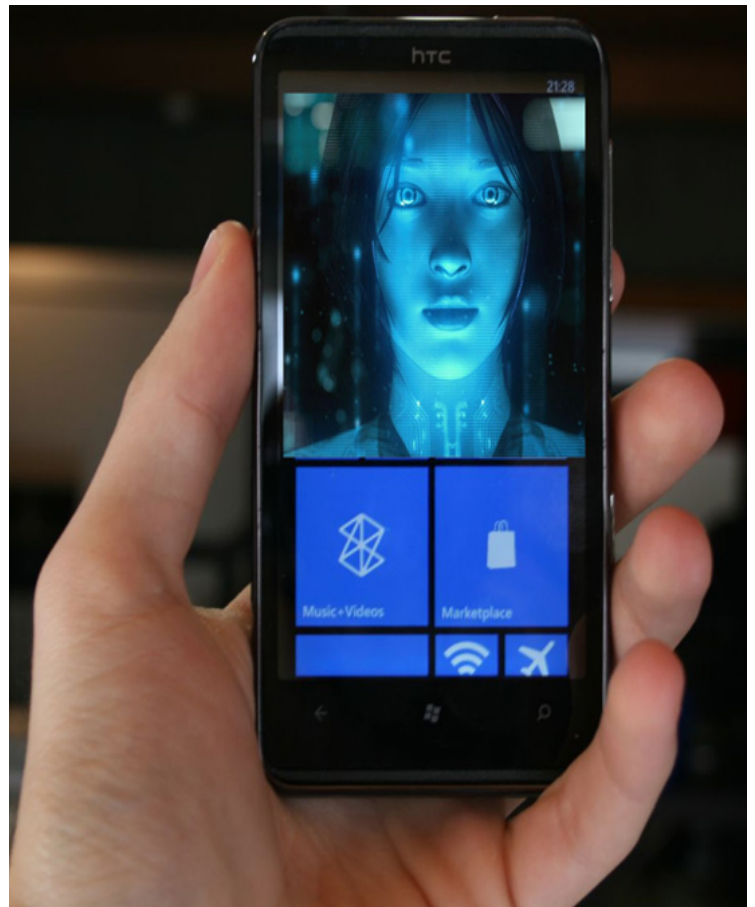


Será então apresentado o ecrã das ofertas especiais:



Conclusão

Podemos concluir que a interação com a [Cortana](#) é relativamente simples, sendo possível fornecer exemplos ao utilizador para facilitar a interação. É possível definir casos em que serão apresentados ecrãs específicos e claro será possível usar uma linguagem natural para podermos satisfazer os pedidos do utilizador recorrendo a expressões comuns.



AUTOR



Escrito Por Sara Silva

Licenciada em Matemática – Especialidade em Computação, pela Universidade de Coimbra e é Microsoft Certified Professional Developer. Atualmente o seu foco de desenvolvimento incide em Windows Phone e Windows 8 Store Apps. O seu Blog é www.saramgsilva.com e o twitter é [@saramgsilva](https://twitter.com/saramgsilva).

A PROGRAMAR

Estruturas de Dados (EDs) Genéricas

Sencha Touch – Parte I

Registo de Todos os comandos executados num Sistema Informático

Como Suportar diferentes idiomas numa App Android

Sandcastle Help File Builder

Estruturas de Dados (EDs) Genéricas

Introdução

Quando cursamos disciplinas de Algoritmos e Estruturas de Dados nos cursos de computação e áreas afins, em geral, aprendemos a criar Estruturas de Dados (EDs) básicas, como listas, pilhas, filas, capazes de trabalharem apenas com números inteiros. Isto não é essencialmente um problema, uma vez que o intuito destas disciplinas é apresentar os principais tipos de EDs, as operações relacionadas a elas, bem como as formas de uso (aplicação) das mesmas; o que não depende do tipo de dados armazenado pela estrutura.

Contudo, quando vamos para a prática do desenvolvimento e utilização destas EDs, seja num trabalho prático da disciplina ou em outro tipo de aplicação, o nosso desejo é trabalhar com EDs capazes de armazenar qualquer tipo de elemento e não apenas de números inteiros. Por exemplo, nosso interesse pode ser criar uma lista de alunos ou uma fila de processos a serem atendidos por um determinado sistema operacional, entre outros. Obviamente, alunos e processos são entidades complexas (por exemplo, o registro de um aluno pode conter nome, matrícula, data de nascimento, entre outros dados), uma vez que são compostas por outros tipos de dados mais simples, e não pertencem aos tipos primitivos de qualquer linguagem de programação de propósito geral.

Neste sentido, como podemos trabalhar com estruturas de dados para esses tipos complexos? Uma forma seria implementar uma versão da estrutura de dados específica para cada tipo de dados que desejamos armazenar, isto é, uma implementação de lista específica para trabalhar com alunos, outra para processos, outra para inteiros e assim sucessivamente. Contudo, podemos perceber que esta não é uma abordagem muito interessante, uma vez que: i) aumenta o volume de trabalho do programador, que deverá reimplementar as estruturas para cada tipo de dado de interesse; ii) aumenta a quantidade de código duplicado, uma vez que o código fonte de funções comuns a um mesmo tipo de estrutura, como por exemplo, inserir no final de uma lista deveria ser duplicado para cada novo tipo de dado; iii) aumentaria muito a probabilidade de o programador cometer erros, comprometendo a qualidade do software como um todo; entre outros.

Enfim, são muitas as razões para não adotarmos a estratégia apresentada anteriormente. Outra opção seria implementarmos “estruturas genéricas”, isto é, estruturas de dados que trabalham de forma transparente para o usuário final (programador), independentemente do tipo de dado que está sendo manipulado por ela. Sendo assim, o correto seria

criarmos uma ED denominada Lista, por exemplo, que encapsula o código necessário para manipulação dos itens de uma lista, não importando que tipo de item seja este. Em outras palavras, teríamos uma única implementação do tipo Lista, que pode armazenar e tratar desde números inteiros, *strings*, até estruturas complexas como alunos, processos, produtos, entre outros.

Esta é a estratégia utilizada pela maioria das bibliotecas de estruturas de dados das linguagens de programação modernas, como Java, C++, entre outros. Por exemplo, em Java existe a implementação `ArrayList`, que representa uma lista implementada por meio de vetores. Ao criarmos um objeto do tipo `ArrayList`, podemos informar que tipo de dados iremos armazenar na ED, por exemplo `ArrayList<Aluno>`, e ela estará preparada para lidar com itens deste tipo, sem precisarmos conhecer, muito menos alterar, o código fonte desta estrutura de dados.

Uma linguagem de programação bastante utilizada em disciplinas iniciais de programação e estruturas de dados é a linguagem C. Apesar de muitos livros já abordarem o ensino de ED com linguagens orientadas a objetos, como C++ e Java, ainda há livros que tratam do assunto utilizando esta linguagem procedimental.

Quanto às linguagens de programação Java e C++, muitos livros exploram a construção de estruturas de dados genéricas, tendo em conta que o paradigma de Orientação a Objetos facilita bastante este trabalho, por meio dos robustos mecanismos de generalização existentes nativamente nestas linguagens, como herança e polimorfismo. Quanto à linguagem C, no entanto, é difícil encontrar material didático sobre o desenvolvimento de estruturas genéricas para tal linguagem. Isto tem feito com que muitos alunos encontrem dificuldades ao aplicarem os conhecimentos teóricos obtidos nas disciplinas que abordam o tema estruturas de dados em situações práticas. Neste sentido, este artigo tem como objetivo apresentar o desenvolvimento de uma estrutura de dados conhecida como **pilha** de forma genérica, isto é, de forma que ela seja capaz de armazenar qualquer tipo de elemento especificado pelo utilizador, sem exigir que seu código seja modificado.

A Pilha

Antes de iniciarmos a implementação de nossa estrutura de dados genérica, é importante explicarmos sucintamente do que se trata a ED conhecida como pilha.

Uma das EDs mais simples, mas não menos importantes do que as demais, é a pilha. Possivelmente por essa razão, é uma das EDs mais utilizada em programação, sendo

A PROGRAMAR

ESTRUTURAS DE DADOS (EDS) GENÉRICAS

inclusive implementada diretamente pelo hardware da maioria das máquinas modernas.

A ideia fundamental da pilha é que todo o acesso aos seus elementos é feito por meio do seu **topo**. Assim, quando um elemento novo é introduzido na pilha, ele passa a ser o elemento do topo e o **único elemento que pode ser removido ou acessado da pilha é o do topo**. Isto faz com que os elementos da pilha sejam retirados na ordem inversa à ordem em que foram introduzidos: o primeiro elemento a sair da pilha é o último que entrou.

A sigla *LIFO* – Last in, First out (O último a entrar é o primeiro a sair) – é usada para descrever esta estratégia.

Para entendermos o funcionamento de uma pilha, podemos fazer uma analogia com uma pilha de pratos. Se quisermos adicionar um prato na pilha, o colocaremos no topo. Se quisermos retirar um prato da pilha, retiraremos o prato do topo. Assim, para termos acesso ao segundo prato da pilha, precisaremos retirar o prato do topo.

A ED pilha funciona de maneira análoga. Existem três operações básicas que devem ser implementadas em uma pilha:

- a operação para empilhar um novo elemento, inserindo-o no topo;
- a operação para desempilhar um elemento, removendo-o do topo; e
- a operação para consultar o elemento do topo, sem removê-lo.

É comum nos referirmos a essas operações pelos termos em inglês push (empilhar), pop (desempilhar) e top (obter elemento do topo). A Figura 1 apresenta graficamente o comportamento das funções push e pop três funções [1].

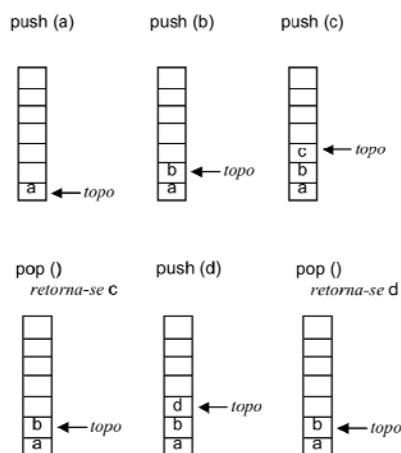


Figura 1. Ilustração do funcionamento de uma ED do tipo pilha.

Na computação, o exemplo de utilização de pilha mais comum é a própria pilha de execução de um programa. Neste caso, as variáveis locais das funções são dispostas em uma pilha e uma função só tem acesso às variáveis que estão no topo, ou seja, não é possível aceder a variáveis de outras funções.

Implementando uma Pilha

Há várias maneiras de se implementar uma ED e cada implementação apresenta vantagens e desvantagens particulares. As duas principais maneiras de se implementar uma ED são:

- Utilizando alocação sequencial estática (com vetores); e
- Utilizando alocação não sequencial dinâmica (com ponteiros).

Neste artigo, adotaremos a implementação com vetores (que é o tipo de implementação utilizado pelo ArrayList do Java). Mais detalhes sobre as vantagens e desvantagens desta estratégia de implementação são apresentados na Seção 5 deste artigo.

A estrutura de pilha implementada neste artigo contém:

- um vetor de inteiros, denominado “itens”, que armazena, em posições contíguas da memória, os elementos da pilha;
- uma variável inteira, denominada “topo”, que armazena a posição onde o próximo elemento da pilha será inserido, ou seja, o topo da pilha; e
- uma constante, “TAM_MAX”, que define a quantidade máxima de elementos suportada pela pilha.

Os passos para implementação desta ED são apresentados abaixo:

Passo 1: definir a estrutura da pilha. A Listagem 1 apresenta o código do arquivo Pilha.h, responsável pela definição das estruturas fundamentais da pilha, bem como do protótipo de suas funções. A constante TAM_MAX (linha 3) define a capacidade do vetor de itens, isto é, da quantidade máxima de itens que podem ser adicionados à pilha. A estrutura struct Pilha (linhas 4 à 7) contém os elementos necessários para armazenamento e manipulação dos dados na pilha.

```
#ifndef PILHA_H
#define PILHA_H
#define TAM_MAX 10
typedef struct {
    int itens[TAM_MAX];
    int topo;
} PilhaVet;
#endif
```

Listagem 1. Estrutura de uma pilha.

Passo 2: o segundo passo consiste na definição dos protótipos das funções a serem oferecidas pela pilha. Tais funções são:

- **Criar Pilha:** aloca e retorna uma pilha vazia para o utilizador.
- **Liberar Pilha:** libera a memória alocada para a pilha.
- **Verificar Pilha Vazia:** verifica se a pilha está vazia.
- **Verificar Pilha Cheia:** verifica se a pilha está cheia.
- **Empilhar:** insere um novo elemento no topo da pilha.
- **Desempilhar:** remove e retorna o elemento do topo da pilha.
- **Obter Topo:** retorna o elemento do topo da pilha, sem removê-lo.

Os protótipos das funções descritas acima devem ser incluídos no arquivo Pilha.h, conforme apresentado na Listagem 2. É possível perceber também a declaração de mais duas constantes, a saber TRUE e FALSE, que serão utilizadas pelas funções estahVazia e estahCheia. Como a linguagem C não possui nativamente um tipo específico para dados booleanos, esta é uma forma de representar esse tipo de informação de forma mais legível no código fonte do programa. Alguns trechos de código já apresentados na Listagem 1 foram omitidos na Listagem 2.

```
...
#define TRUE 1
#define FALSE 0
...
Pilha* criarPilha();
void liberarPilha(Pilha* pilha);
int estahVazia(Pilha* pilha);
int estahCheia(Pilha* pilha);
void empilhar(int item, Pilha* pilha);
int desempilhar(Pilha* pilha);
int obterTopo(Pilha* pilha);
...
```

Listagem 2. Protótipos das funções de uma pilha.

A implementação da função criarPilha deve ser realizada em um arquivo denominado Pilha.c. A implementação desta função encontra-se na Listagem 3. Conforme pode ser visto, o procedimento para criar uma pilha é: (i) instanciar dinamicamente a pilha (linha 5); (ii) atribuir o valor 0 (zero) à variável topo, uma vez que a pilha encontra-se vazia no início (linha 6); e (iii) retornar a pilha criada dinamicamente (linha 7).

```
#include<stdlib.h>
#include "Pilha.h"

PilhaVet* criarPilha() {
    PilhaVet* pilha = (Pilha*) malloc(sizeof
(Pilha));
    pilha->topo = 0;
    return pilha;
}
```

Listagem 3. Implementação da função criarPilha.

Ressalta-se ainda a necessidade de se importar dois arquivos de cabeçalho: i) o stdlib.h, que permite utilização de funções para gerenciamento dinâmico de memória, como malloc, free, entre outros; e ii) o Pilha.h, que define as estruturas e funções oferecidas pela pilha.

A implementação das demais funções é apresentada na Listagem 4. Foge ao escopo deste artigo descrever o funcionamento detalhado destas funções; mais informações sobre a estrutura pilha, seu funcionamento e implementação podem ser encontrados em [1].

```
void liberarPilha(Pilha* pilha) {
    if (pilha != NULL) free(pilha);
}

int estahCheia(Pilha* pilha) {
    if (pilha->topo == TAM_MAX) return TRUE;
    else return FALSE;
}

int estahVazia(Pilha* pilha) {
    if (pilha->topo == 0) return TRUE;
    else return FALSE;
}

void empilhar(int item, Pilha* pilha) {
    if (!estahCheia(pilha)) {
        pilha->itens[pilha->topo] = item;
        pilha->topo++;
    } else printf("Pilha cheia!\n");
}

int desempilhar(Pilha* pilha) {
    if (!estahVazia(pilha)) {
        int elementoTopo = pilha->itens[pilha->
topo - 1];
        pilha->topo--;
        return elementoTopo;
    } else printf("Pilha vazia!\n");
}

int obterTopo(Pilha* pilha) {
    if (!estahVazia(pilha)) {
        int elementoTopo = pilha->itens[pilha->
topo - 1];
        return elementoTopo;
    } else printf("Pilha vazia!\n");
}
```

Listagem 4. Implementação das demais funções da pilha.

A Listagem 5 apresenta um código simples para criação e manipulação de uma pilha.

```
#include "Pilha.h"
void imprimirPilha(Pilha* pilha) {
    while(!estahVazia(pilha)) printf("%d\n",
desempilhar(pilha));
}

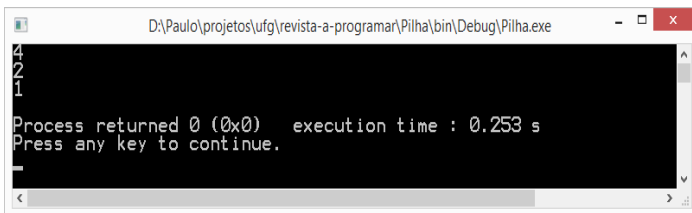
int main() {
    Pilha* minhaPilha = criarPilha();
    empilhar(1, minhaPilha); empilhar(2,
minhaPilha); empilhar(3, minhaPilha);
desempilhar(minhaPilha);
empilhar(4, minhaPilha);
imprimirPilha(minhaPilha);
return 0;
}
```

Listagem 5. Código fonte para utilização da estrutura pilha.

A PROGRAMAR

ESTRUTURAS DE DADOS (EDS) GENÉRICAS

Inicialmente, este programa insere os elementos 1, 2 e 3 na pilha (linha 7); assim o elemento 3 passa o topo da pilha. Posteriormente, o elemento do topo é removido (linha 8) e o elemento 2 passa a ser o topo. Na linha 9, o elemento 4 é inserido na pilha, passando a ser o topo da mesma e, por fim, a função imprimirPilha (linha 10) é invocada e o resultado da execução desta função é apresentado na Figura 2.



```
D:\Paulo\projetos\ufg\revista-a-programar\Pilha\bin\Debug\Pilha.exe
4
2
1
Process returned 0 (0x0)   execution time : 0.253 s
Press any key to continue.
```

Figura 2. Resultado da execução da função imprimirPilha.

O que nos chama a atenção nos trechos de código apresentados anteriormente é que esta pilha funciona adequadamente, porém, apenas para números inteiros. Ou seja, não conseguiríamos armazenar, recuperar ou remover outros tipos de dados nesta estrutura, como strings ou tipos complexos, como structs, por exemplo “Aluno”, “Processo”, “Disciplina”, entre outros.

Se quiséssemos atingir tal objetivo, teríamos que modificar alguns trechos de código da implementação da estrutura, como por exemplo, a linha 5 da Listagem 1, que descreve o tipo de elementos do vetor itens ou as funções empilhar e desempilhar, da Listagem 4, uma vez que o tipo de item recebido pela função empilhar e o tipo de retorno da função desempilhar seriam diferentes.

É importante ressaltar que esse código poderia ser melhorado com a adição de um tipo de dados específico para encapsular o item de uma estrutura de dados. Assim, a mudança ficaria concentrada na alteração de um único tipo de estrutura, contudo ainda assim seria necessário modificar o código da estrutura de dados, o que não é uma tarefa trivial e muito menos recomendada, por diversos motivos já comentados no início deste artigo.

Para amenizar estes problemas, uma solução mais elegante é a construção de estruturas genéricas, que permitem ao programador especificar, no momento da criação da estrutura de dados, qual o tipo de elemento que será mantido por ela e partir daí, todo o trabalho realizado pela ED fica transparente para o programador e nenhum código desta estrutura precisará de ser modificado.

Criando uma Pilha Genérica

Um recurso a ser utilizado para confecção da pilha genérica é o tipo void da linguagem C. Muitos devem saber que void é uma palavra reservada da linguagem C, utilizada, por exemplo, para especificar que uma função não possui um tipo de retorno. O que nem todos devem saber é que esse recurso também é utilizado para especificar tipos genéricos em C. Por exemplo, uma função que retorna um ponteiro para um tipo

void (void*), na verdade, retorna um ponteiro que pode ser convertido via casting explícito para qualquer outro tipo existente na linguagem C ou até mesmo para tipos definidos pelo próprio utilizador.

Este recurso é utilizado em algumas funções de bibliotecas da linguagem C, como por exemplo, a função malloc. Se observarmos a documentação desta função [2], sua assinatura é void* malloc (size_t size); essa função aloca um determinado espaço de memória do tamanho especificado pelo utilizador e retorna um ponteiro para o bloco de memória reservado. Esse ponteiro, por sua vez, pode ser convertido em um ponteiro para qualquer outro tipo de dado. Por exemplo, pode-se alocar uma variável inteira dinamicamente por meio do trecho de código da Listagem 6.

```
int* meuInt = (int*) malloc(sizeof(int));
```

Listagem 6. Utilização da função malloc.

A primeira alteração realizada no código fonte da nossa estrutura de dados encontra-se no arquivo de cabeçalho, Pilha.h, conforme mostrado na Listagem 7.

```
typedef struct {
    void* itens[TAM_MAX];
    int tamItem;
    int topo;
} Pilha;
```

Listagem 7. Estrutura Pilha modificada.

Neste caso, a struct Pilha não contém mais um vetor de inteiros, mas sim um vetor de ponteiros para o tipo genérico void. Em outras palavras, tem-se um vetor capaz de conter elementos de qualquer tipo de dado. Outra mudança é o acréscimo da variável tamItem na estrutura Pilha. Ela será utilizada nas funções desempilhar e obterTopo, nas quais será preciso conhecer o tamanho do item que está sendo manipulado pela pilha para que seja possível realizar uma cópia deste item antes de retorná-lo ao utilizador.

No caso das outras funções da pilha, as modificações realizadas são apresentadas na Listagem 8 (as funções que não sofreram modificações foram omitidas). Como pode ser observado, a única mudança realizada na função criarPilha encontra-se nas linhas 1 e 4, nas quais o tamanho do item a ser armazenado na pilha é passado como parâmetro para esta função e posteriormente copiado para a variável tamItem da estrutura pilha, respectivamente.

```
Pilha* criarPilha(int tamItem) {
    Pilha* pilha = (Pilha*) malloc(sizeof(Pilha));
    pilha->topo = 0;
    pilha->tamItem = tamItem;
    return pilha;
}
void* desempilhar(Pilha* pilha) {
    if (!estahVazia(pilha)) {
        void* elRemovido = malloc(pilha->tamItem);
        memcpy(elRemovido, pilha->itens[pilha->
            topo - 1], pilha->tamItem);
        pilha->topo--;
        return elRemovido;
    } else return NULL;
}
```


A PROGRAMAR

ESTRUTURAS DE DADOS (EDS) GENÉRICAS

```
}  
void* obterTopo(Pilha* pilha) {  
    if (!estahVazia(pilha)) {  
        void* elTopo = malloc(pilha->tamItem);  
        memcpy(elTopo, pilha->itens[pilha->topo - 1],  
               pilha->tamItem);  
        return elTopo;  
    } else return NULL;  
}
```

Listagem 8. Demais funções da pilha modificadas.

Quanto às funções `desempilhar` e `obterTopo`, as principais mudanças estão nas linhas 9 à 10 e 17 à 18, respectivamente. Nestes casos, antes de retornar o elemento do topo da pilha para o utilizador, uma cópia do mesmo é realizada. Para que isso seja possível, utilizou-se a função `memcpy` (acrónimo de **memory copy**), da biblioteca `stdlib.h`. Esta função copia os valores da região apontada pelo topo da pilha para uma nova região de memória, apontada por `elRemovido`. O ponteiro `elRemovido` é genérico e aponta para uma região de memória instanciada dinamicamente por meio da função `malloc`. Neste ponto, é possível identificar onde a variável `tamItem` é utilizada. Para que uma cópia do elemento do topo da pilha seja feita, antes, é necessário saber o tamanho deste item para que possa ser alocado um espaço de memória compatível para receber a cópia.

Com essas poucas mudanças, nossa estrutura de dados já suporta diferentes tipos de dados. A Listagem 9 apresenta duas formas de uso desta estrutura, uma capaz de armazenar números inteiros na pilha e outra capaz de armazenar *strings*.

```
#include "Pilha.h"  
  
typedef struct {  
    int info;  
} ItemInt;  
typedef struct {  
    char info[255];  
} ItemStr;  
  
void imprimirPilhaInt(Pilha* pilha) {  
    while(!estahVazia(pilha)) printf("%d\n",  
        ((ItemInt*)desempilhar(pilha))->info);  
}  
void imprimirPilhaStr(Pilha* pilha) {  
    while(!estahVazia(pilha)) printf("%s\n",  
        ((ItemStr*)desempilhar(pilha))->info);  
}  
  
void adicionarElementoInt(int v, Pilha* p) {  
    ItemInt* i = (ItemInt*) malloc(sizeof  
        (ItemInt));  
    i->info = v; empilhar(i, p);  
}  
void adicionarElementoStr(char s[], Pilha* p) {  
    ItemStr* i = (ItemStr*) malloc(sizeof  
        (ItemStr));  
    strcpy(i->info, s); empilhar(i, p);  
}  
  
int main() {  
    Pilha* minhaPilhaInt = criarPilha(sizeof  
        (ItemInt));  
    Pilha* minhaPilhaStr = criarPilha(sizeof  
        (ItemStr));
```

```
adicionarElementoInt(1, minhaPilhaInt);  
adicionarElementoInt(2, minhaPilhaInt);  
adicionarElementoInt(3, minhaPilhaInt);  
desempilhar(minhaPilhaInt);  
adicionarElementoInt(4, minhaPilhaInt);  
imprimirPilhaInt(minhaPilhaInt);  
  
printf("\n\n");  
  
adicionarElementoStr("A", minhaPilhaStr);  
adicionarElementoStr("B", minhaPilhaStr);  
adicionarElementoStr("C", minhaPilhaStr);  
desempilhar(minhaPilhaStr);  
adicionarElementoStr("D", minhaPilhaStr);  
imprimirPilhaStr(minhaPilhaStr);  
  
return 0;  
}
```

Listagem 9. Código fonte para utilização da estrutura pilha modificado.

É possível observar que as funções que são específicas para o tipo de item da pilha, como a função que imprime os elementos de uma pilha, possuem duas versões, uma para números inteiros (`imprimirPilhaInt`) e outra para *strings* (`imprimirPilhaStr`). Isso é aceitável, uma vez que para imprimir um elemento, deve-se conhecer como esse elemento é formado. Imaginemos por exemplo, que temos uma lista de registros de alunos; a forma de impressão de um registro como esse é bastante específica. Além disso, esse é um método que não faz parte da implementação da pilha, ou seja, é de responsabilidade do programador que está a manipular a estrutura.

As demais funções, que dizem respeito à estrutura da pilha em si, como `empilhar`, `desempilhar`, `verificar se está vazia` ou `se está cheia` não precisam de ser modificados. Basta invocá-las passando como parâmetro o ponteiro para a lista desejada e o comportamento da função será executado com sucesso, independentemente do tipo de objeto armazenado por ela. Entretanto, uma questão que devemos pensar é: e se quiséssemos adicionar uma função à estrutura da pilha, que fosse capaz de `desempilhar` todos os elementos até encontrar um determinado elemento desejado. Por exemplo, suponha que a pilha seja composta pelos elementos 1, 2 e 3, sendo 3 o topo da mesma. Se invocássemos esta função passando o item 1 como parâmetro, os elementos 2 e 3 seriam removidos e a pilha ficaria com apenas um elemento.

O impasse aqui é o seguinte: como a função saberá como comparar os itens da pilha, uma vez que ela não conhece o seu tipo, a priori. Mesmo sabendo qual é o tipo de dado, em alguns casos, não seria possível compará-los; por exemplo, quando o tipo de dado é uma estrutura complexa criada pelo próprio desenvolvedor, como uma `struct` para armazenar os dados de um aluno, entre outros.

Para resolver este problema, precisamos de utilizar outro recurso da linguagem C, conhecido como funções *callback*. Este tipo de função tira proveito do fato de que a linguagem C trabalha com ponteiros para funções, isto é, podemos

A PROGRAMAR

ESTRUTURAS DE DADOS (EDS) GENÉRICAS

passar como parâmetro para uma função, um ponteiro que aponta para o bloco da memória onde está localizada outra função do sistema (ou até mesmo ela mesma). Isto permite que uma determinada função realize chamadas de outra função, sem mesmo saber que função é esta. Mas este será um assunto para um próximo artigo.

Considerações finais

Este artigo apresentou a implementação de uma pilha genérica, isto é, uma estrutura de dados capaz de lidar com elementos de qualquer tipo de dado, como inteiros, *strings*, entre outros. Inicialmente, destacou a motivação para construção deste tipo de estrutura; posteriormente, apresentou-se a implementação de uma pilha específica para lidar com números inteiros. Por fim, mostrou-se como evoluir esta estrutura para torná-la genérica.

Uma das limitações da implementação apresentada neste artigo é que a capacidade da pilha é limitada, ou seja, uma vez atingida a capacidade máxima desta estrutura, não se pode mais inserir elementos na mesma. Na prática, isso não é muito comum. O que ocorre, mesmo em estruturas implementadas

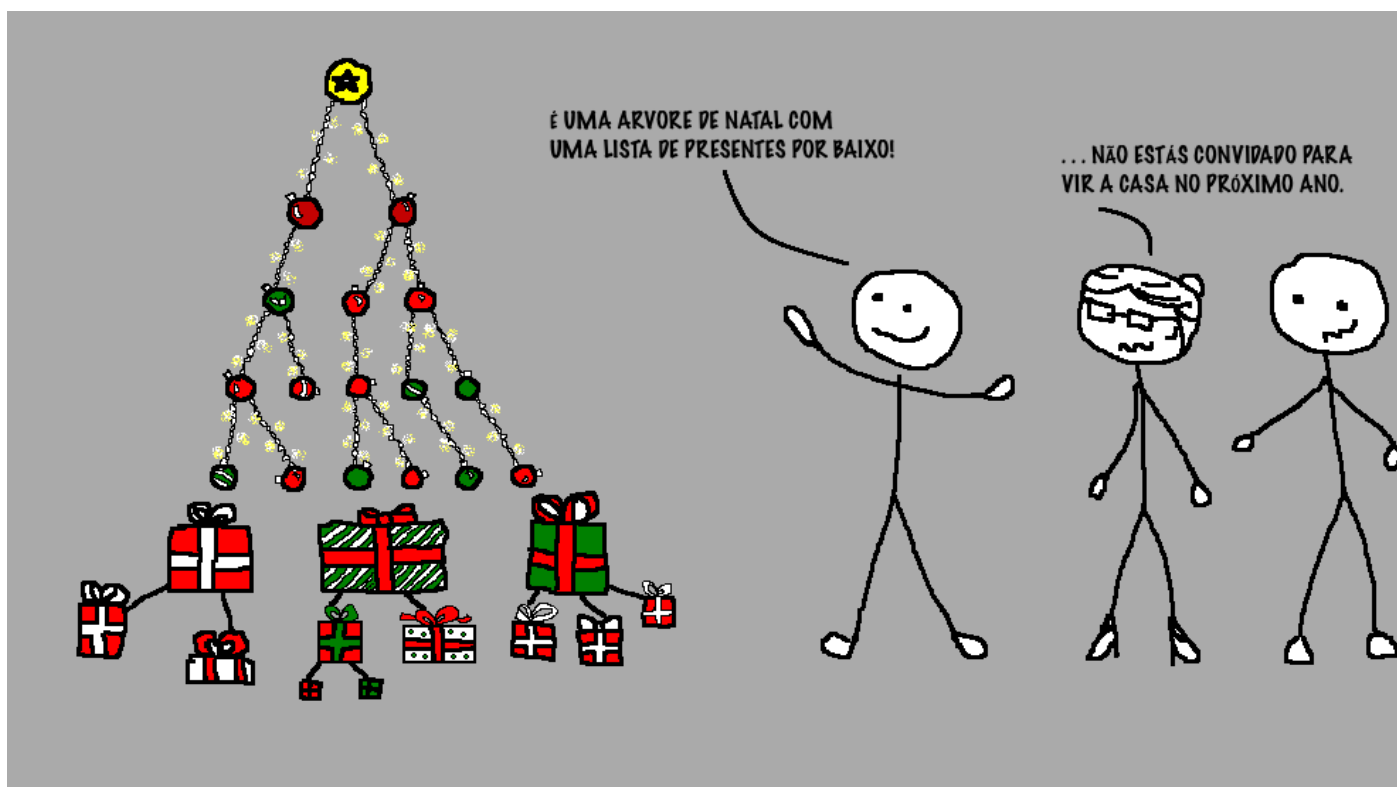
com vetores, é que um novo vetor com maior capacidade é alocado, caso seja possível, e os elementos do vetor antigo são transferidos para o novo vetor.

Nos próximos artigos, pretende-se apresentar uma alternativa para resolução do problema anterior, bem como apresentar o desenvolvimento de funções para a pilha que dependem do tipo de item manipulado por ela para ser executada. Além disso, pretende-se abordar a implementação de outras estruturas de dados genéricas, como listas, filas, árvores, entre outras.

Referências

Celes, W.; Rangel, J. L. "Apostila de Estruturas de Dados". Disponível em: <http://www-usr.inf.ufsm.br/~juvizzotto/elc1067-2013b/estrut-dados-pucrio.pdf>. Acedido em: Agosto/2014.

Documentação da função *malloc*. Disponível em: <http://www.cplusplus.com/reference/cstdlib/malloc/>. Acedido em: Agosto/2014.



AUTOR



Escrito por Paulo Afonso Parreira Júnior

Atualmente é professor do curso de Bacharelado em Ciência da Computação da Universidade Federal de Goiás (Campus Jataí). É aluno de doutorado do Programa de Pós-Graduação em Ciência 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 Departamento de Computação da Universidade Federal de São Carlos e do Grupo de Pesquisa e Desenvolvimento de Jogos Educacionais Digitais (GrupJED) do Curso de Ciência da Computação da Universidade Federal de Goiás (Regional Jataí). Tem experiência na área de Ciência da Computação, com ênfase em Engenharia de Software, atuando principalmente nos seguintes temas: Manutenção de Software, Desenvolvimento de Software Orientado a Objetos, Desenvolvimento de Software Orientado a Aspectos e Informática na Educação.

Sencha Touch – Parte I

Actualmente, como todos nós sabemos, o universo dos smartphones e das aplicações móveis vieram para ficar. O mercado das apps está em constante evolução e todos os dias aumenta o seu número nas diversas stores existentes. O problema existente é que cada fabricante obriga a uma linguagem de programação. Se queremos desenvolver uma aplicação para IOS, temos que programar em Objective-C, para Android, o Java, e Windows Phone somos obrigados a usar o C#, por exemplo. Ora, programar em todas estas linguagens é bastante complicado.

A solução para esta questão passa pela utilização de uma framework híbrida. Este tipo de frameworks permitem ao programador desenvolver uma aplicação em HTML5 e Javascript capaz de ser executada em qualquer sistema operativo móvel com o look and feel de uma aplicação nativa.

Exemplo deste tipo de frameworks é o Sencha Touch. Com esta framework é tão simples desenvolver uma aplicação com um design atractivo e cross-plataform. É importante referir que esta framework é totalmente gratuita e open-source.

Nesta introdução será demonstrado como preparar o ambiente de desenvolvimento, assim como um pequeno exemplo das capacidades desta poderosa framework.

Após o pequeno exemplo estar completo, será ainda explicado como iremos encapsular a app HTML5/Javascript numa aplicação nativa, utilizando o apache cordova.

Para criar o ambiente de desenvolvimento será necessário realizar o download da framework do seu site oficial: <http://www.sencha.com/products/touch/download/>.

Será necessário terem instalado um servidor web apache, como por exemplo o **WAMP ou XAMPP**.

Finalmente, é preciso efectuar o download do Sencha Touch CMD, para criar a aplicação via linha de comandos (<http://developer.sencha.com/pathway/sencha-cmd-3>).

Terminados todos os downloads, é tempo de criar o ambiente.

1. Unzip da framework dentro do servidor web.
2. Executar o seguinte comando para criar a aplicação Sencha Touch:
 - a. Navegar para a pasta onde extraíram o sdk e executar o comando:

Sencha generate app [nome aplicacao] [caminho]

Depois de criada a aplicação, basta aceder pelo browser ao local onde criar a app. Por exemplo:

Sencha generate app pop ../pop

No browser basta aceder ao endereço do webserver <http://localhost/pop/>.

A aplicação criada por defeito, trás já algumas funcionalidades que podemos aproveitar para o nosso desenvolvimento.

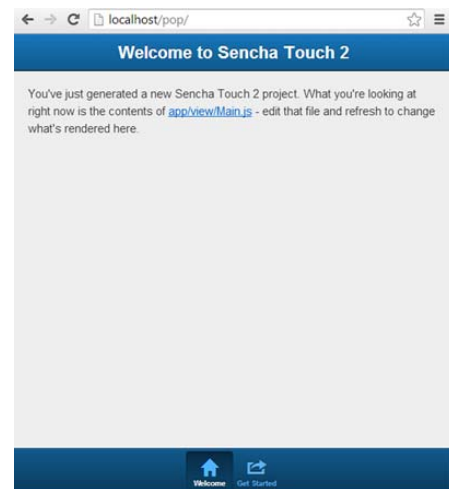
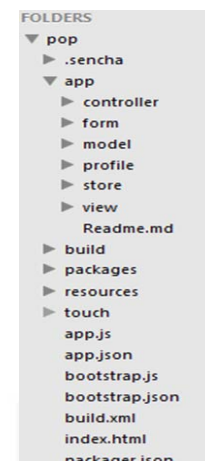


Figura 1: Exemplo APP inicial

Em seguida, será apresentada a arquitetura da framework, de forma a perceber melhor como está funciona.

O Sencha Touch está organizado segundo a arquitectura MVC (Model-View-Controller) onde os componentes visuais estão separados do modelo de negócio. Além desta componente, possui uma estrutura de pastas que ajuda ao programador a organizar o seu código.

Tal como é possível verificar na figura 2, a estrutura de pastas de um projecto Sencha Touch está muito bem organizada, o que facilita o desenvolvimento. A pasta mais importante de referir neste momento, é a pasta App. Dentro da App está o modelo MVC da nossa aplicação e ainda as pastas FORM,STORE e PROFILE.



A PROGRAMAR

SENCHA TOUCH – PARTE I

Antes de começar a desenvolver, é necessário entender todos estes pequenos conceitos da framework de forma a tirar todo o partido desta.

A pasta da STORE é muito importante no desenvolvimento desta framework. Tal como o nome indica, a store será onde iremos armazenar dados da aplicação. Contudo, é apenas um auxiliar ao armazenamento dos dados, uma vez que o que será guardado nas stores serão dados voláteis, isto é, apenas serão guardados enquanto a aplicação estiver a ser executada.

Podem-se encarar as stores como um auxiliar ao armazenamento dos dados, uma vez que toda a informação é armazenada numa base de dados SQLite. As stores são bastante úteis para carregar os dados provenientes da base de dados, e para efectuar operações sobre eles, como ordenações, filtros, etc.

O Sencha Touch é bastante simples de utilizar depois de entendido estes conceitos. Vários objectos visuais do Sencha Touch estão já preparados para consumirem stores. Exemplos destes objectos visuais são as listas por exemplo.

Criando uma lista com base numa store é bastante simples.

Criar modelo para a store: O modelo é comparado a uma classe no JAVA, por exemplo. É esse modelo que irá representar a tabela na base de dados. Imagine-se que temos na base de dados uma tabela Cliente com os campos ID, NOME e MORADA. Temos que replicar essa informação num ficheiro dentro da pasta MODEL, tal como o seguinte exemplo:

```
Ext.define('pop.model.Cliente', {
  extend: 'Ext.data.Model',
  config: {
    idProperty: 'ID',
    fields: [
      { name: 'ID', type: 'string' },
      { name: 'NOME', type: 'string' },
      { name: 'MORADA', type: 'string' }
    ]
  }
});
```

Depois de o modelo estar criado, temos que indicar que temos uma store com esse mesmo modelo. Assim, basta criar um novo ficheiro dentro da pasta STORE e apontar para o modelo CLIENTE.

```
Ext.define('pop.store.Cliente', {
  extend: 'Ext.data.Store',
  config: {
    storeId: 'cliente',
    model: 'pop.model.Cliente'
  }
});
```

Todos os ficheiros criados quer sejam modelos, stores, view ou controllers devem ser referidos no ficheiro app.js

Declarar views:

```
views: ['Menu', 'ListaCliente'];
```

Declarar Models:

```
models: ['Cliente']
```

Declarar Stores:

```
stores: ['Cliente']
```

Declarar Controllers:

```
controllers: ['RegistaCliente']
```

Depois de criado o modelo para o Cliente e a respectiva store, apresentar uma lista com os clientes é bastante simples. Basta utilizar o componente ST **EXT.LIST**.

```
Ext.define('pop.view.ListaCliente', {
  extend: 'Ext.List',
  xtype: 'listaclientes',
  config: {
    autoDestroy: true,
    store: 'Cliente',
    grouped: true,
    itemTpl: '<table>' +
      '<tr>' +
      '<td style="font-size:16px;color:#37899E;">' +
      '<b>{NOME}</b>' +
      '</td>' +
      '<tr>' +
      '<td>' +
      '<b style="font-size:14px;' +
      'color:#37899E;">MORADA:</b><b style=' +
      '"font-size:14px;"> {MORADA}</b>' +
      '</td>' +
      '</tr>' +
      '</tr>' +
      '</table>'
  }
});
```

Desta forma será apresentada uma lista com o nome e morada dos clientes. Esta lista é populada com os dados provenientes na store, através do uso da sua propriedade **store: 'Cliente'**.

A forma como a lista é apresentada fica ao critério de cada um. Usando a propriedade itemTpl pode-se definir um HTML para criar o corpo de cada item da lista.

É importante referir que todos os ficheiros criados devem iniciar-se da seguinte forma:

```
Ext.define([nome do ficheiro], {
```

A propriedade Extend indica que tipo é o objecto que estamos a criar. Pode ser um container, um panel, uma lista, etc.

Outra propriedade importante de referir é o xtype. Este é um identificador único do componente visual que será utilizado quando necessitarmos de instanciar esse mesmo objecto.

Pelo que podem analisar até agora, os conceitos existentes no Sencha Touch têm uma curva de aprendizagem bastante reduzida, sendo bastante fácil desenvolver com esta framework.

Na próxima edição será demonstrado um exemplo prático de

A PROGRAMAR

SENCHA TOUCH – PARTE I

uma aplicação com todos estes conceitos aplicados.

Será ainda demonstrado como converter a aplicação Sencha Touch para uma aplicação nativa android e publicação na play store.



AUTOR

Escrito por Nelson Silva

Licenciado em Engenharia Informática pela Universidade de Trás-os-Montes e Alto Douro. A frequentar mestrado em Eng. Informática pelo ISEP. Trabalha profissionalmente em empresa de TI na área da saúde. Fundador da página do facebook: Android para totós.

Registo de Todos os Comandos Executados num Sistema Informático

RESUMO

Este relatório descreve a análise efetuada a quatro ferramentas que permitem o registo de todos os comandos efetuados num sistema informático. Nota-se, mais uma vez, que não existe segurança absoluta, por um lado, e por outro, que as ferramentas que possibilitam maiores níveis de segurança são igualmente as que têm maiores custos, a nível de trabalho humano e computacional.

Descrição do Assunto e Categorias

D.4.6 [Segurança e Proteção]

Assunto Geral

Segurança

Tópicos fulcrais

registo, comandos, performance, custos humanos

INTRODUÇÃO

O objetivo principal deste projeto é estudar, de um ponto de vista da segurança do sistema, a possibilidade de registar (*log*) todos os comandos executados por um utilizador. A segurança (correspondendo ao inglês *security* e não *safety*) será abordada quer pela perspetiva de um ataque intencional ao sistema, quer pela possibilidade de uma utilização negligente poder permitir explorar vulnerabilidades existentes ou criar novas.

Objetivos e motivação

A primeira questão que se poderá colocar, de um ponto de vista da segurança de todo o sistema, é se faz sentido fazer um registo exaustivo de todos comandos executados, por oposição ao registo de apenas alguns comandos, ou de alguns utilizadores. Este trabalho não vai aprofundar esta questão, dado que fica muito além do seu escopo analisar a melhor forma de implementar toda a segurança de um sistema informático.

Bastará assim como motivação para estudar o registo de todos os comandos um regra prática da segurança informática e a abordagem que foi acima descrita, que engloba quer um ataque intencional, quer uma atuação negligente. Se num ataque intencional ainda se poderá (pelo menos em teoria) pretender ter uma lista das atuações potencialmente importantes a registar, na vertente de uma atuação negligente, essa possibilidade por natureza não existe, pois a própria definição de uma atuação negligente (mais ou menos grosseira) pressupõe a respetiva aleatoriedade resultante de ser uma atuação não planeada ou imprevista.

Software analisado

A lista de *software* que permite realizar o objetivo deste estudo não é extensa. Contudo, com uma exceção, nenhum se destina especificamente a esta tarefa, tendo aplicabilidades mais vasta. Naturalmente, dado o âmbito deste estudo, apenas iremos focar a possibilidade de registar os comandos executados.

grsecurity

O *grsecurity* é um conjunto de *patches* para o *kernel* Linux que visa melhorar globalmente toda a segurança do sistema. As suas funcionalidades são múltiplas e abrangem áreas muito distintas. Citando apenas as mais relevantes, podemos encontrar: *Role Based Access Control*, *PAX* (que implementa algumas técnicas para impedir a execução de código arbitrário, como a marcação de páginas de memória somente para leitura quando contenham código (*read only*), ou a randomização do espaço de endereçamento da memória), limitações de acesso a parte do sistema de ficheiros e proteção de executáveis, rede e *kernel*. Entre esta miríade de possibilidades, existe igualmente a opção de registar todos os comandos executados.

Este registo é efetuado a partir de código que é adicionado ao próprio *kernel*, de modo a registar as chamadas efetuadas à função de sistema *execve*.

Linux Auditing System

O *Linux Auditing System* é uma ferramenta desenvolvida e mantida pela *Red Hat* e que se encontra presente no *kernel* do Linux desde a versão 2.6. O objetivo do LAuS é registar todas as chamadas de sistema (*system calls*) realizadas. No entanto, pode ser configurado para registar apenas a chamadas *execve*, de modo a cumprir os objetivos previstos neste projeto.

Esta ferramenta tem dois módulos, normalmente designados por servidor e cliente. O servidor foi a parte incluído no *Linux Kernel* v.2.6, enquanto a parte cliente pode ser usada para configuração e acesso aos dados.

O funcionamento desta ferramenta é muito fácil de explicar: dado que parte do código é executado dentro *kernel*, o LAuS consegue registar todas as chamadas de sistema efetuadas.

The GNU Accounting Utilities

As *GNU Accounting Utilities* são um conjunto de ferramentas criadas há vários anos, num momento em que o tempo de processamento era um bem raro. O objetivo inicial era apenas saber quem estava a usar o tempo de processamento do (único) computador disponível e o que estava a fazer com o mesmo.

Entre as cerca de 6 (seis) ferramentas disponíveis encontra-se o *lastcomm* que permite listar os comandos executados.

Snoopy Logger e outras ferramentas

O Snoopy Logger é uma ferramenta *open source* destinada especificamente a registar todos os comandos executados. No entanto, de acordo com as instruções dos autores (e não só), não é uma ferramenta que vise a área da segurança, sendo um dos motivos o facto de não ser difícil escapar ao respetivo registo (como veremos).

Esta ferramenta funciona como um *wrapper* da função *execve* que, além de chamar a mesma, regista os argumentos dessa chamada.

Existem outras ferramentas *open source* que permitem igualmente registar todos os comandos executados numa *shell*, sendo dois exemplos o *sudosh* e o *rootsh*. Estas não serão analisadas, pois além da sua menor divulgação e documentação, têm uma aplicabilidade bastante mais limitada, não permitindo registar todos os comandos executados num sistema, mas apenas no sistema de *shell*, que terá de ser substituído por esta versão com a funcionalidade adicional de registar os comandos do utilizador. Este tipo de ferramentas além de apresentarem um campo de aplicação mais limitado, reduzem as opções de *shell* disponíveis (e logo, as opções disponibilizadas aos utilizadores) e criam ainda uma vulnerabilidade adicional, resultante da eventual possibilidade de o atacante conseguir correr outra *shell*.

Os testes realizados

Além do estudo do funcionamento e documentação das ferramentas selecionadas, foram realizados alguns testes com vista a aferir, por um lado, da respetiva segurança ou fiabilidade, e por outro, das perdas em termos de desempenho que o sistema sofre com a operação adicional de registo.

Os testes foram todos realizados numa máquina i5 760, a 2.8Gh, com 8 Gb de RAM, a correr a versão 7 do Debian (Wheezy).

Os testes de fiabilidade

Dado que a maioria das ferramentas selecionadas se encontram em produção há vários anos e sendo todas *open source*, não houve a pretensão de encontrar falhas de segurança óbvias, principalmente num trabalho com as limitações temporais e objetivos que o presente texto apresenta. No entanto, foram feitas algumas verificações de situações limite. Desde simples utilização de *alias* para chamar executáveis e que todas as ferramentas ultrapassaram, facilmente registando o executável realmente chamado, até à utilização de *sudo* para mudar de um utilizador normal (*user*) para administrador (*root*), facto que permitiu esconder a algumas aplicações a verdadeira identidade de quem invocou certo comando.

Foi igualmente investigado se seria possível ocultar um comando ou o respetivo utilizador, pela utilização de aplicações inter-

mediárias. Assim, dentro do cliente de correio eletrónico de linha de comandos “Alpine” foi invocado o editor de texto “vi” para a elaboração de uma nova mensagem. Foi então utilizada a funcionalidade do “vi” para executar um comando de *shell* e retornar o resultado para dentro deste editor.

Os testes de desempenho

Dadas as ferramentas selecionadas terem modos de funcionamento algo distintos, e de modo a fazer uma avaliação correta do desempenho do sistema foram elaborados 3 (três) testes bastantes distintos:

- Compilação de *software*
Optou-se, em razão do tempo de compilação e estabilidade do código, pelo *openssh-client* v.1.6.0.
- Sequência de vários comandos de utilização corrente por qualquer utilizador normal

Para este teste foi especificamente elaborado um pequeno *script* em *bash*:

```
#!/bin/bash
mkdir teste
cd teste
for dir in d1 d2 d3 d4 d5
do
  echo -e "\n** Making dir ${dir}"
  mkdir $dir
  cd $dir

  for file in f1 f2 f3 f4 f5
  do
    echo -e "\n Making file ${file}"
    dd if=/dev/
    urandom of=${file} count=1M bs=1 > /dev/null
  done
  ls -lA > /dev/null
  cd ..
done
echo -e "\n Cating all files"
cat $(find -type f) > /dev/null
cd ..
echo -e "\n Compressing"
tar cf compressed.tar teste
rm -r teste
```

- c) Por fim, foi efetuado um teste de processamento intensivo.

Este teste foi efetuado com recurso a um outro projeto pessoal que realiza a análise de um ficheiro de dados numéricos, imprimindo no ecrã o resultado. A análise consiste em variados cálculos que a aplicação distribui por todos os cores disponíveis na máquina. Pretendeu-se, ao utilizar especificamente esta aplicação, obter resultados mais fiáveis nos atuais ambientes multicore.

Os testes de desempenho foram todos realizados com um utilizador normal, sem privilégios especiais.

Análise

A segurança de um sistema é, regra geral, o resultado de um

A PROGRAMAR

REGISTO DE TODOS OS COMANDOS EXECUTADOS NUM SISTEMA INFORMÁTICO

conjunto de opções de ponderação entre segurança e custo. Na análise deste binómio considerou-se que seria relevante, além dos custos de processamento – de análise mais quantificável – realizar igualmente uma análise mais subjetiva, sobre a facilidade ou dificuldade de configuração e utilização do *software*. A ferramenta mais segura, mas cuja instalação exija um técnico especializado só será acessível a um pequeno público e foi, por isso, considerado que este aspeto, embora não sendo tão facilmente mesurável como a diminuição de desempenho, devia ser tido em consideração.

Para esta avaliação foi utilizada uma escala igualmente menos quantitativa que pode ser vista na Tabela 1.

Tabela 1 – Análise de aspetos administrativos

	grsecurity	LAuS	GNU AU	SL
Facilidade de Instalação	☹☹	☺	☺☺	☺☺
Facilidade de Configuração	☹☹	☺	☺☺	☺☺
Documentação	☺☺	☺☺	☺	☹

Instalação, configuração, utilização e documentação

Todas as ferramentas, com exceção do grsecurity, foram instaladas através do repositório oficial Debian e não apresentam qualquer dificuldade na respetiva instalação. O grsecurity, dado que consiste num conjunto de patches ao kernel, requer conhecimentos avançados para ser instalado. No caso do Debian (e outras distribuições nas mesmas condições) o kernel recebe já alguns patches, pelo que não é possível aplicar os patches do grsecurity diretamente, sendo necessário aplicá-los à versão original do kernel, compilá-lo e instalá-lo.

As dificuldades no grsecurity transmitem-se igualmente à configuração, que é necessário fazer no momento da configuração do kernel, através das ferramentas disponíveis para tal.

O LAuS está vastamente documentado e selecionar a opção para registar todos os comandos executados requer adicionar apenas duas linhas aos ficheiros de configuração. No entanto, dado o enorme leque de possibilidades que esta ferramenta oferece, descobrir as opções corretas requer algum investimento na leitura da documentação e conhecimentos técnicos avançados.

No extremo opostos, quer as GNU Accounting Utilities, quer o Snoopy Logger não requerem qualquer configuração. Pelo que foi possível perceber, numa distribuição menos “pró-ativa” que o Debian, além da instalação apenas seria necessário executar os comandos para iniciar as respetivas ferramentas – procedimento que, em ambos os casos, se encontra devidamente documentado.

A utilização fundamental destas aplicações passa pela consulta dos registos realizados, que será objeto de análise no próximo ponto.

Os registos (logs)

Aspetos formais

Localização

Os registos criados nem sempre foram criados onde seria de esperar. Embora sempre de acordo com a *Filesystem Hierarchy Standard* () a pasta escolhida seja `/var/log`, o Snoopy Logger gera os seus relatórios dentro do ficheiro `auth.log` e o gsecurity dentro do `syslog`. Embora tal não seja, à partida um problema, não foi fácil encontrar tal aspeto na documentação.

Formatos

À exceção das GNU Accounting Utilities, todas as ferramentas guardam os relatórios em formato de texto, facilmente acessível. As Accounting Utilities incluem as aplicações necessárias para aceder ao formato binário que lhes é específico.

Legibilidade

As GNU Accounting Utilities disponibilizam, de longe, o conteúdo mais legível e fácil de interpretar. Todas as outras ferramentas registam demasiada informação para que possam ter uma leitura fácil. Isto não deve ser visto como um aspeto negativo. Trata-se tão somente de uma consequência do volume de informação que é registado.

Embora o facto de os registos serem demasiado verbosos possa ser visto como potenciador de alguma insegurança, no sentido que ao dificultarem a leitura irão impedir a respetiva consulta e diminuir a respetiva utilidade, tanto o grsecurity como o LAuS, estão conscientes desse problema e fornecem ferramentas que permitem filtrar a informação relevante. O Snoopy Logger é uma exceção neste caso.

Aspetos de Conteúdo

O conteúdo dos registos é bastante variável, em consonância com o que atrás ficou dito, relativamente à legibilidade. Os registos mais legíveis, das GNU Accounting Utilities, são igualmente os mais incompletos, listando apenas a data, o comando, o nome do utilizador, a consola em que o comando foi executado e uma indicação adicional com informações de sistema. Um elemento com bastante importância, e que é omitido, são os parâmetros (ou argumentos) com que o comando foi executado.

Tanto o Snoopy Logger como o LAuS registam bastante mais informação, nomeadamente ID e GID (*group ID*). As GNU Accounting Utilities são aliás únicas no aspeto em que apresentam o nome do utilizador e não apenas um número. O grsecurity apresenta o maior volume de informação, tornando-se quase ininteligível numa primeira abordagem. Para este estudo interessa-nos especialmente que tanto o grsecurity como o LAuS conseguem manter o registo do utilizador mesmo que ele altere a sua identificação, nomeadamente usando o comando `sudo`. O LAuS, utiliza um elemento que designa por AUID (abreviatura de *audit uid*) e que permite

identificar um utilizador que nas GNU Accounting Utilities surge apenas como "root". Esta é uma possibilidade sem paralelo nas restantes duas ferramentas e que dá uma vantagem indiscutível à grsecurity e à LAuS, principalmente em casos de negligência, onde é possível, mesmo que várias pessoas tenham acesso a uma consola administrativa (onde possam utilizar o comando *sudo*) identificar quem realizou determinada tarefa (ou, neste caso, quem executou quais comandos).

O Snoopy Logger regista um SID (*Session ID*) que pode ser usado com o mesmo efeito, no entanto os seus resultados não são tão fidedignos, até porque resultam de cada sessão que é iniciada, ao contrário do AUID.

Outro aspeto essencial – desta feita, em que todos as ferramentas conseguiram alcançar um bom resultado – é a capacidade para não serem induzidas em erro pela utilização de outras aplicações (intermédias) para chamar os comandos. Nenhuma das ferramentas foi "enganada" por este ardil, registando corretamente a aplicação e o utilizador.

Acesso aos registos

Todas as ferramentas estudadas utilizam exclusivamente o sistema de ficheiros como proteção para limitar o acesso aos ficheiros de registos. O LAuS apenas permite a leitura e escrita ao administrador (root), sendo que as restantes ferramentas permitem igualmente a leitura ao grupo administrador.

	Tempo original	grsecurity	LAuS	GNU Accounting Utilities	Snoopy Logger
a) compilação	59,831	53,98	60,2	59,96	63,93
b) script	55,882	74,386	60,7	55,808	57,28
c) computação intensiva	58,426	60,892	58,8	58,579	58,84

Tabela 2 – Tempo em segundos dos testes de desempenho

Esta opção, quando usada para registar potenciais ataques e nomeadamente ataques bem sucedidos com privilege escalation (ou seja, em termos grosseiros, qualquer exploit que permita a um utilizador comum fazer *sudo*) elimina qualquer fiabilidade concedida a estes registos. Na prática, para a situação mais perigosa, e talvez aquela que faria todo o sentido manter um registo: quanto um atacante consegue ter acesso incondicional a uma máquina, as ferramentas estudadas revelam-se completamente ineficientes.

A alternativa neste tipo de situações é, numa abordagem clássica, ter um sistema de registo externo à própria máquina, presumindo-se que o atacante não conseguirá ser bem sucedido no ataque a ambas as máquinas simultaneamente. Esta estratégia tem por fito atingir um dos objetivos principais de um registo seguro: a sua inalterabilidade (incluindo a respetiva eliminação). No entanto, o atacante poderá a qualquer momento cessar a produção de registos. Este parece ser um problema verdadeiramente insolúvel, uma vez que a partir de determinado ponto, um atacante com suficientes permissões poderá fa-

zer tudo o que um administrador legítimo também pode fazer ().

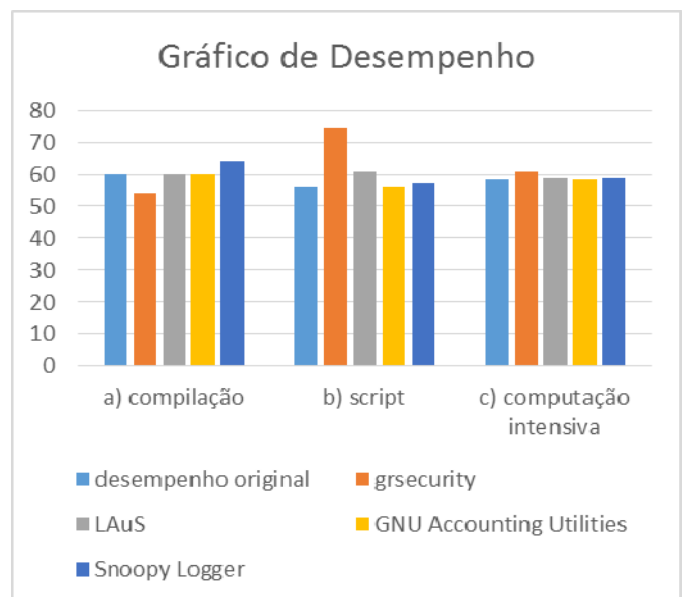
Contornar o registo

Dado que as três primeiras ferramentas funcionam com base em código que já existe (ou é necessário adicionar) no *kernel* do sistema, não será uma tarefa nada fácil conseguir escapar ao respetivo registo. Potenciais vulnerabilidades existentes só estarão ao alcance de alguém com conhecimentos técnicos muito avançados.

A exceção, como aliás já tinha ficado indiciado, é o Snoppy Logger. Ao utilizar somente um *wrapper* para uma biblioteca, permite a existência de várias estratégias para contornar os registos que realiza ().

Eficiência e perdas de performance

Como se poderá verificar no gráfico de desempenho e na tabela de valores apresentados (contabilizadas com base nos valores totais), as diferenças de desempenho são, respetivamente de 8%, 3%, 0,1% e 3%, pra o grsecurity, o LAuS, as GNU Accounting Utilities e o Snoppy Logger. Tal como esperado, uma maior produção de conteúdos tem como contrapartida um maior impacto no desempenho geral do sistema.



Como panorâmica geral, não se retiram aqui mais conclusões que as que resultam do estudo global da Segurança de Sistemas Informáticos: há sempre um equilíbrio a considerar entre a segurança e o custo, que neste caso é refletido quer em tempo de configuração e instalação, quer em performance. A solução com mais dados, que, tal como ficou exposto a propósito do objetivo deste trabalho se presumirá que seja a mais completa, ou mais segura (independentemente do contributo que tenha na segurança global do sistema, que como já mencionado, não cabe aqui analisar) é igualmente a que tem os maiores custos.

A PROGRAMAR

REGISTO DE TODOS OS COMANDOS EXECUTADOS NUM SISTEMA INFORMÁTICO

Conclusões

Este projeto procurou identificar um conjunto de soluções para um problema muito específico, a saber, o registo de todos os comandos executados num sistema informático. Após alguma investigação e teste de várias ferramentas que foram selecionadas por disponibilizarem tal funcionalidade, acabou por se concluir que existe sempre uma situação, em que o atacante obtém permissões administrativa, em que nenhuma ferramenta se mostrou capaz de minorar os danos. Nos restantes casos existe um equilíbrio entre a informação registada e os custos de utilização da ferramenta. Como mera opinião pessoal, parece-nos que o LAuS é a opção mais equilibrada (entre custo e benefício). No entanto, como em qualquer questão de segurança, a opção correta é exclusivamente a opção correta para o sistema em análise, pelo que cada Administrador de Sistemas terá de fazer a sua própria opção com base nos respetivos requisitos de segurança.

REFERÊNCIAS

Schneier, Bruce, and Kelsey, John, Secure Audit Logs to Support Computer Forensics, ACM Transactions on Information and System Security, v. 1, n. 3, 1999, disponível online em <https://www.schneier.com/paper-auditlogs.html>

Snoppy Logger website <https://github.com/a2o/snoopy>

Documentação oficial das GNU Accounting Utilities <http://www.gnu.org/software/acct>

Documentação oficial do grsecurity <http://en.wikibooks.org/wiki/Grsecurity>

Man page das ferramentas do LAuS: <http://goo.gl/82ugh2>

(auditd) <http://goo.gl/AMSFOY>

(auditctl) e <http://goo.gl/sm042M>

Todas as referências para páginas web foram consultado por último a 8/11/2014.

¹<http://www.pathname.com/fhs/>

²Para um estudo bastante interessante sobre a possibilidade de uma máquina apenas garantir a inalterabilidade dos seus registos ver [Schneier and Kelsey]. Notar, no entanto, que mesmo os autores acabam por concluir por algumas vulnerabilidades no modelo proposto – dir-se-ia inerentes a tentar guardar um objeto do seu próprio possuidor (legítimo ou ilegítimo).



AUTOR



Escrito por Fernando César

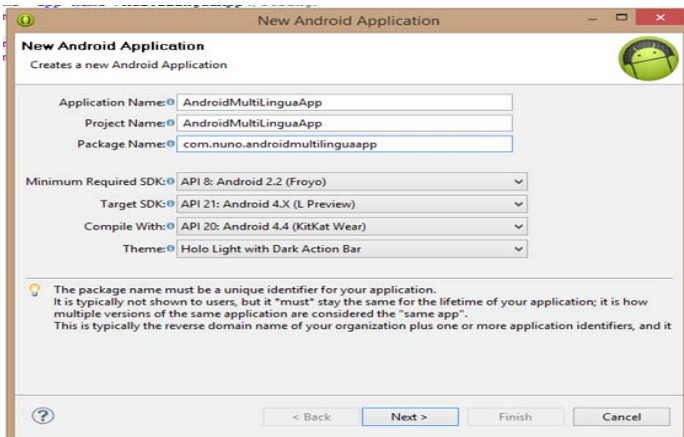
Como Suportar Diferentes Idiomas numa App Android

O mundo das aplicações móveis é cada vez mais abrangente, fazendo já parte do dia-a-dia de qualquer um dos mortais com smartphone. Esta fragmentação e abrangência na utilização de aplicações móveis torna complicada a tarefa de ter um público-alvo bem definido, pois “geralmente” as aplicações estão disponíveis para a globalidade dos utilizadores e não limitadas a regiões.

Esta situação torna a definição dos idiomas para o público-alvo da aplicação um requisito fundamental para o correto desenvolvimento de uma aplicação. Neste pequeno tutorial irei dar a conhecer uma forma de implementar o suporte a múltiplos idiomas numa aplicação Android.

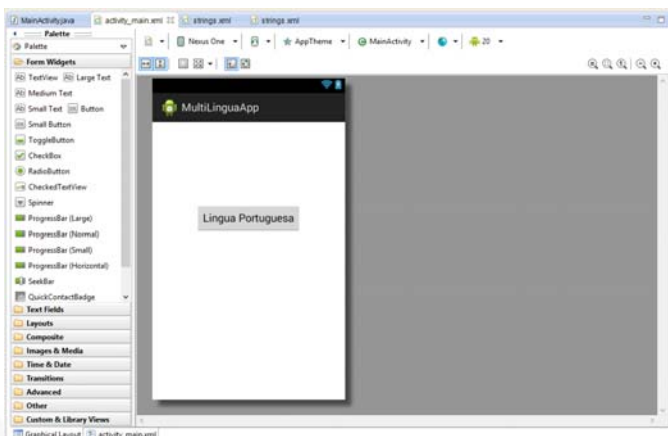
Não irei dar destaque à instalação e configuração do ambiente de desenvolvimento, partindo do princípio que todo o ambiente está corretamente instalado e configurado.

1º Passo: Criar um “Android Application Project”



Neste primeiro passo iremos criar um “Android Application Project”, onde definimos o nome da App, nome do Projeto, assim como as versões Android suportadas. É ainda criada uma “Main Activity” que será a nossa “view” principal neste tutorial.

2º Passo: Desenhar a “view”



Após a criação do projeto vamos então dar início à construção da nossa “view”, basicamente à construção do aspeto gráfico que neste exemplo concreto terá apenas uma “view”, ou seja uma “activity”.

Para este exemplo iremos apenas criar um “button” e colocá-lo numa qualquer posição do ecrã à nossa escolha, como é demonstrado na imagem acima.

De seguida e se olharmos para o código no ficheiro “activity_main.xml” teremos a seguinte estrutura:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.nuno.multilinguaapp.MainActivity" >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="120dp"
        android:text="Button"
        android:onClick="actionMethod" />

</RelativeLayout>
```

Será então aqui que começarão as alterações com vista a tornarmos a nossa aplicação com suporte para múltiplos idiomas.

Vamos agora dar atenção à seguinte linha:

```
android:text="Button"
```

Esta linha é responsável por definir o nome a apresentar no “button” e que neste caso é “Button”. Para podermos ter acesso posterior e permissão para alterarmos dinamicamente o conteúdo a apresentar como nome do “button” termos de mudar para o seguinte:

```
android:text="@string/button_pt"
```

Com esta linha iremos então fazer com que o conteúdo a apresentar como nome do “button” seja fornecido por uma string que irá estar alojada na pasta “res/values”, ficheiro

A PROGRAMAR

COMO SUPORTAR DIFERENTES IDIOMAS NUMA APP ANDROID

“strings”.

Temos ainda de adicionar a seguinte linha para que mais tarde possamos aceder ao método “click” do nosso “button”:

```
android:onClick="actionMethod" />
```

Olhando agora para o ficheiro “strings” que está presente na pasta “res/values” ele deverá ter uma estrutura semelhante à apresentada abaixo:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

  <string name="app_name">MultiLinguaApp</string>
  <string name="action_settings">Settings
                                </string>

  <string name="button_pt">Lingua Portuguesa
                            </string>
  <string name="mensagem">Mensagem em
                            Português!!!</string>

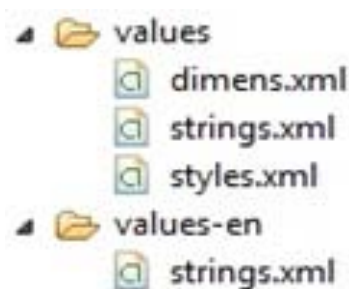
</resources>
```

Aqui as duas linhas importantes são as seguintes:

```
<string name="button_pt">Lingua
                            Portuguesa</string>
<string name="mensagem">Mensagem em
                            Português!!!</string>
```

A primeira linha é a responsável pelo conteúdo do nome do “button”, a segunda linha será a mensagem definida que será apresentada no momento em que o “button” for clicado.

3º Passo: Criar uma nova pasta “values” para o novo idioma

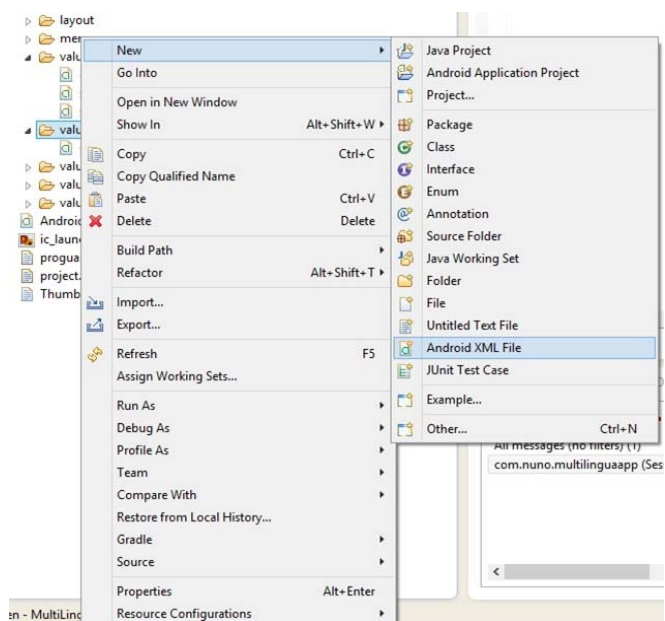


Para que possamos então dar o suporte a múltiplos idiomas à nossa aplicação teremos de criar uma nova pasta “values”, mas desta vez com uma alteração devido a este ter de ser diferenciada para o novo idioma, como por exemplo “values-en”.

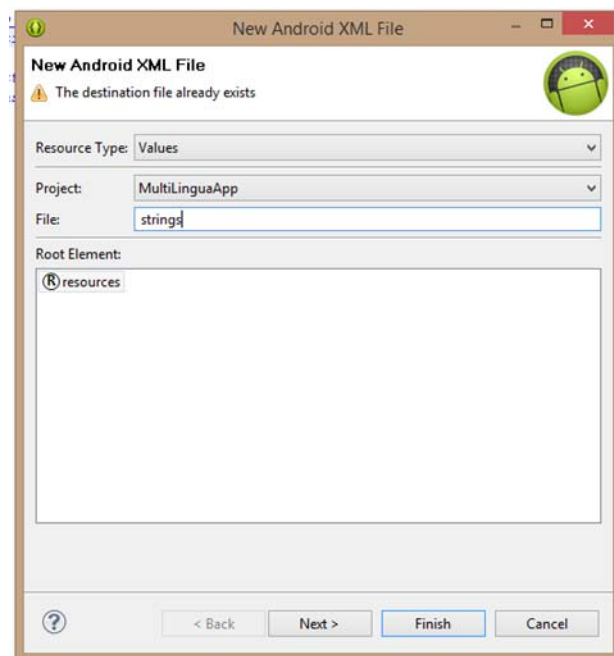
Neste exemplo eu defini o idioma base como “Português” e o idioma secundário o “Inglês” e por esse mesmo motivo estou a usar a pasta “values-en”. Poderão ter acesso a todos os códigos de idiomas suportados pelo Android aqui:

<http://developer.android.com/reference/java/util/Locale.html>

4º Passo: Criar um novo ficheiro “strings.xml” na pasta “values-en”



Agora com a criação deste novo ficheiro é que vamos realmente dar suporte ao novo idioma, sendo que este novo ficheiro “strings.xml” deverá ter uma estrutura idêntica ao ficheiro “strings.xml” padrão.



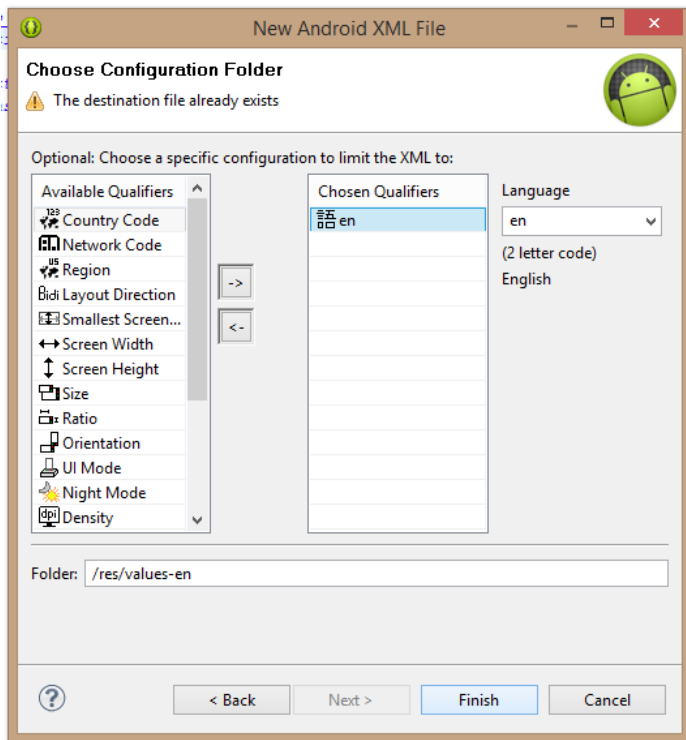
Ao escolhermos a criação do novo ficheiro xml, devemos ter em atenção alguns aspetos, tais como o “Resource Type” que deverá ser do tipo “values” como mostra a imagem acima.

Logo de seguida seremos apresentados com mais algumas opções, sobre as quais devemos selecionar “Country Code”, e escolher o “identificador”, neste caso o “en” referente à

A PROGRAMAR

COMO SUPORTAR DIFERENTES IDIOMAS NUMA APP ANDROID

língua Inglesa.



Tal como havia dito acima a estrutura deste novo ficheiro “strings” será em tudo muito semelhante ao ficheiro “strings” padrão, pelo que deverá copiar todo o conteúdo do ficheiro “strings” padrão para este novo ficheiro “strings”, alterando apenas o conteúdo de duas linhas.

A estrutura deste novo ficheiro deverá ser a seguinte:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">MultiLinguaApp</string>
  <string name="action_settings">Settings
  </string>

  <string name="button_pt">English Language
  </string>
  <string name="mensagem">English Message!!!
  </string>
</resources>
```

Como é perceptível a estrutura do ficheiro é em tudo semelhante ao ficheiro padrão, tendo sido apenas alteradas as linhas que possuíam o conteúdo do nome do “button” e da mensagem.

5º Passo: Definir o método “actionMethod” no “MainActivity.java”

Depois de termos definido os conteúdos referentes aos dois idiomas vamos agora definir o método que nos permitirá fazer o “click” no “button” e nos ser apresentada a mensagem definida.

Para isso devemos navegar até à pasta “src/ nome_package” e abrir o ficheiro “MainActivity.java” ao qual

adicionaremos o seguinte método no final:

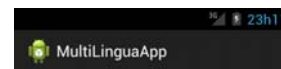
```
//Método ActionMethod
public void actionMethod(View v){
  String msg = (String)getText
  (R.string.mensagem); //Acesso à string
  //alojada no ficheiro "strings"
  Toast.makeText(getBaseContext(), msg,
  Toast.LENGTH_LONG).show(); //Apresentação
  //da mensagem no ecrã }
}
```

Este método irá ser chamado cada vez que o “button” for clicado, apresentando então a mensagem definida.

6º Passo: Alteração do Idioma no Sistema Operativo e Respetivo Teste

Se tudo correu como o previsto a nossa aplicação estará agora preparada para lidar com dois idiomas, mas para isso iremos testar e verificar se tudo correu como o que desejávamos.

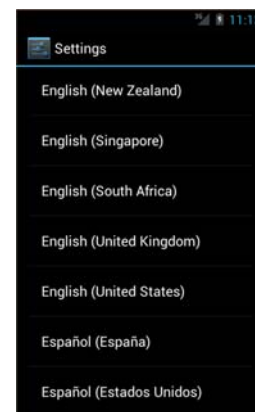
Antes de mais supomos que o idioma definido pelo Sistema Operativo é o “Português”, e assim sendo a aplicação deverá apresentar o seguinte resultado:



Lingua Portuguesa

Mensagem em Português!!!

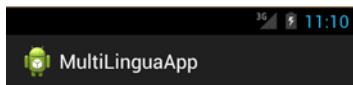
Agora necessitamos de alterar o idioma do Sistema Operativo para “Inglês” de modo a que possamos testar se tudo está como o previsto. Para alterar o idioma deveremos aceder ao Menu “Definições”, “Idioma e entrada”, “Idioma” escolhendo então o “Inglês”.



A PROGRAMAR

COMO SUPORTAR DIFERENTES IDIOMAS NUMA APP ANDROID

Definido o idioma para "Inglês" o resultado da aplicação deverá ser o seguinte:



English Language

English Message!!!

Se tudo correu como o esperado, ambos os idiomas estarão disponíveis na aplicação, sendo que a aplicação irá sempre verificar qual é o idioma definido pelo Sistema Operativo para então depois escolher o idioma a utilizar.

Para suporte a mais idiomas seria necessário apenas repetir o processo, para os idiomas pretendidos.

Existem outras formas de dotar as app's Android com suporte a múltiplos idiomas, algumas delas recorrendo a tradutores. Neste pequeno tutorial o objetivo era apenas demonstrar uma forma simples de adicionar suporte a múltiplos idiomas ao desenvolvimento de uma aplicação Android.



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/>

SANDCASTLE HELP FILE BUILDER

Introdução

Este manual explica como instalar e utilizar a aplicação SandCastle Help File Builder de modo a gerar documentação das assemblies que geramos.

O que esta aplicação faz é pegar numa DLL e pegar no XML dos comentários associados aos métodos, classes, namespaces e propriedades, e coloca-lo num repositório user-friendly.

Instalação

Passo 1: Aceder ao site <http://shfb.codeplex.com/documentation> e escolher o link Latest Release.



Installation and Usage

Download the **latest release**, extract it to a folder, and then run **Sandcastleinstaller.exe** to run the guided installation of Sandcastle, the various extra items from the Sandcastle Styles project, and the Sandcastle Help File Builder.

See the [Installation Instructions](#) topic for more information about the required set of additional tools.

If you are new to Sandcastle and the help file builder, see the topics in the [Getting Started](#) section to get familiar with it, set up your projects to produce XML comments, and create a help file project. A full copy of the help file is also included with the installer and a shortcut to it can be found in the *Sandcastle Help File Builder* folder in the Start | All Programs menu.

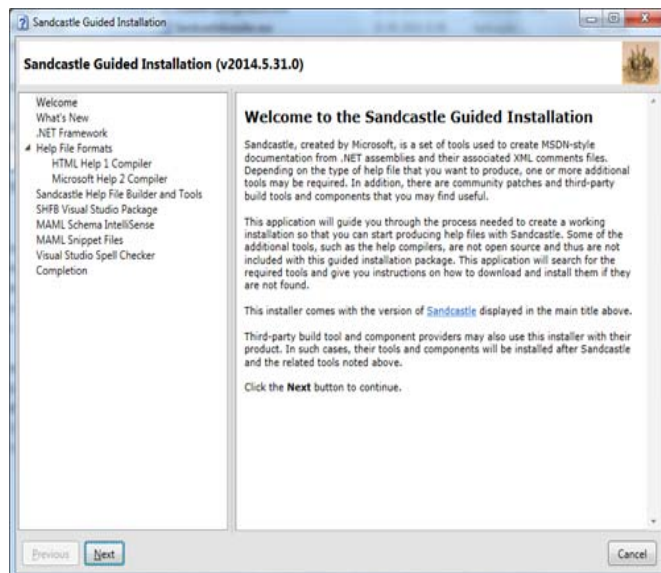
Passo 2: Fazer download do pacote de instalação com a última versão.



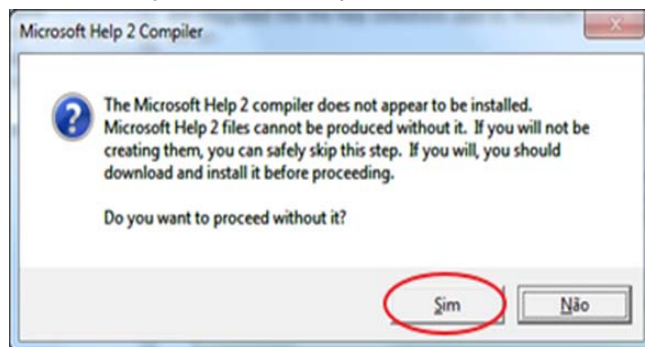
Passo 3: Extrair o conteúdo do ficheiro .ZIP e abrir a pasta.

Nome	Data modificação	Tipo	Tamanho
InstallResources	06-08-2014 14:09	Pasta de ficheiros	
InstallerConfiguration.xml	31-05-2014 22:00	Documento XML	4 KB
SandcastleInstaller.exe	31-05-2014 21:58	Aplicação	177 KB

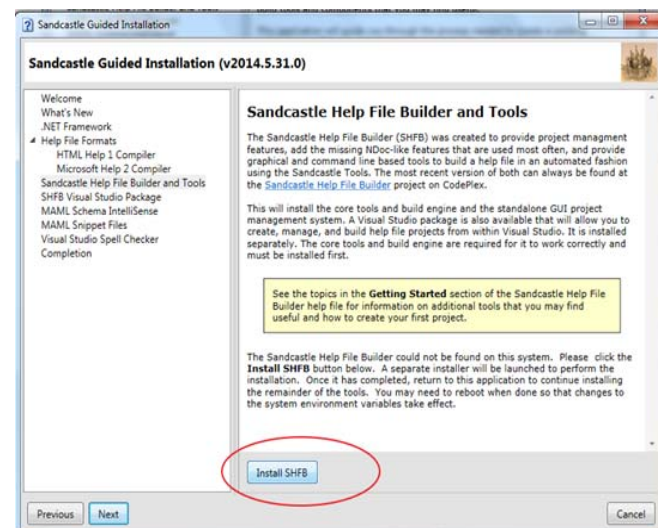
Passo 4: Executar o instalador SandcastleInstaller.exe



Passo 5: Caso o Microsoft Help 2 compiler não esteja instalado, prosseguir com a instalação.



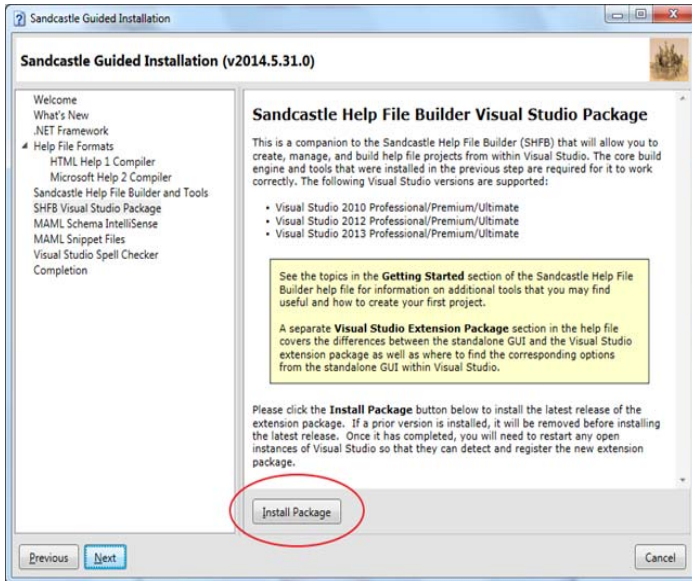
Passo 6: Instalar o SHFB



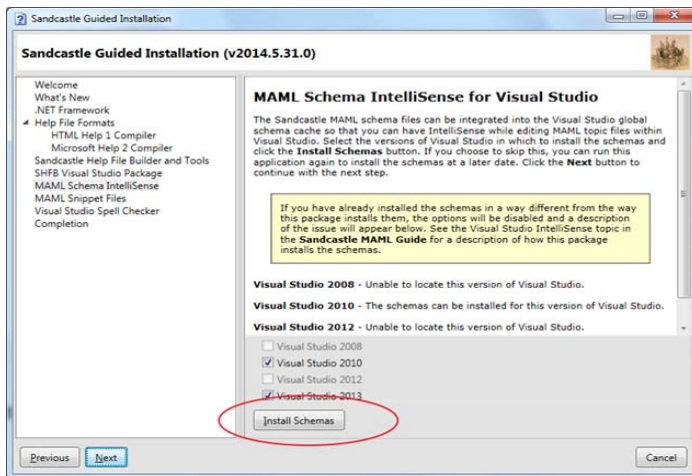
A PROGRAMAR

SANDCASTLE HELP FILE BUILDER

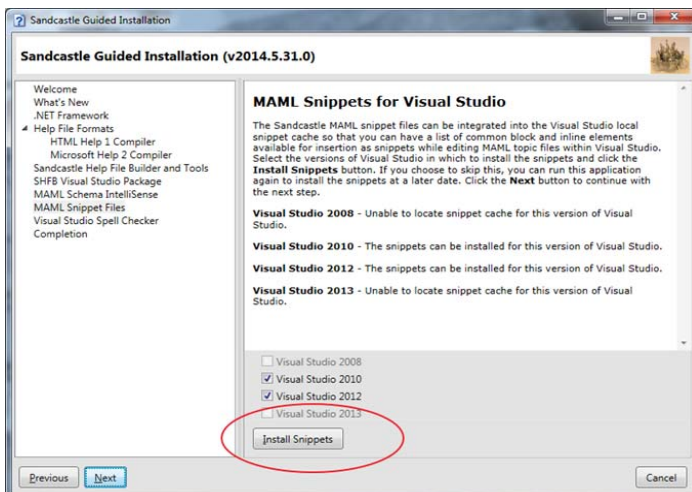
Passo 7: Instalar o package para o Visual Studio (2010 / 2012 / 2013)



Passo 8: Instalar os Schemas MAML (para intellisense)



Passo 9: Instalar os Snippets MAML



Prosseguir até ao fim da instalação.

Passo 10: Criar um documento BATCH com o seguinte conteúdo:

```
set DXROOT=c:\Program Files (x86)\Sandcastle
set SHFBROOT=c:\Program Files (x86)
    \EWSoftware\Sandcastle Help File Builder
set LANGUAGE=
    "%SHFBROOT%\SandcastleBuilderGUI.exe"
```

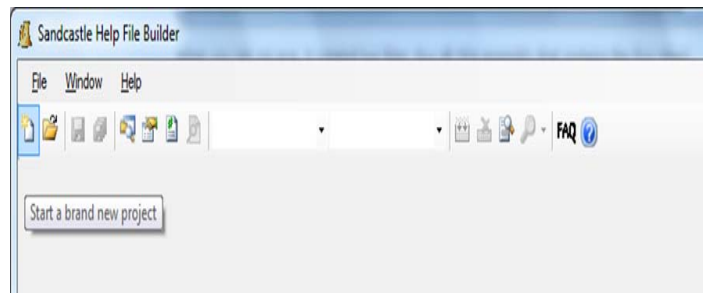
Este BATCH arrancará o GUI do SandCastle Help File Builder.

Criação de documentação

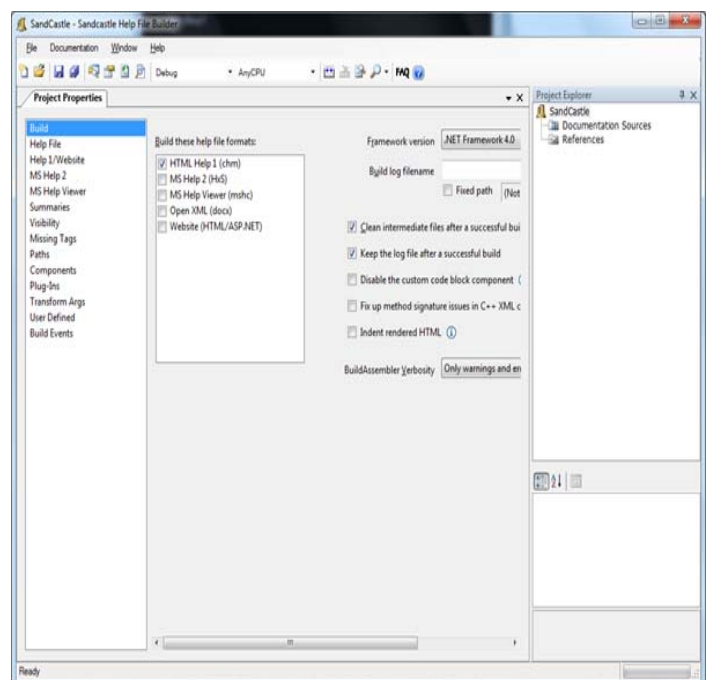
Passo 1: Executar o documento BATCH criado na instalação.

Será aberta a aplicação SandCastle Help File Builder.

Passo 2: Criar um novo projecto (Start a brand new project).



Será mostrada a estrutura do projecto:

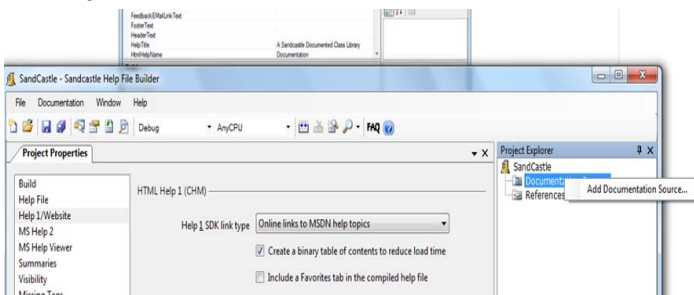


Passo 3: Escolher os formatos de help file a gerar, a framework, etc.

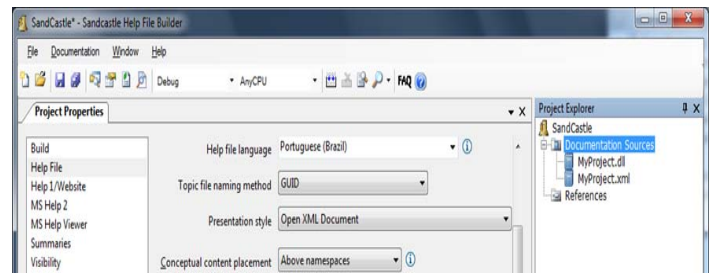
A PROGRAMAR

SANDCASTLE HELP FILE BUILDER

Passo 4: Adicionar DLLs e XMLs como origens de documentação (Documentation source)



Passo 5: Gerar o help file



A documentação será gerada na pasta Help, dentro da pasta do projecto que originou a DLLs (document source).

Referências

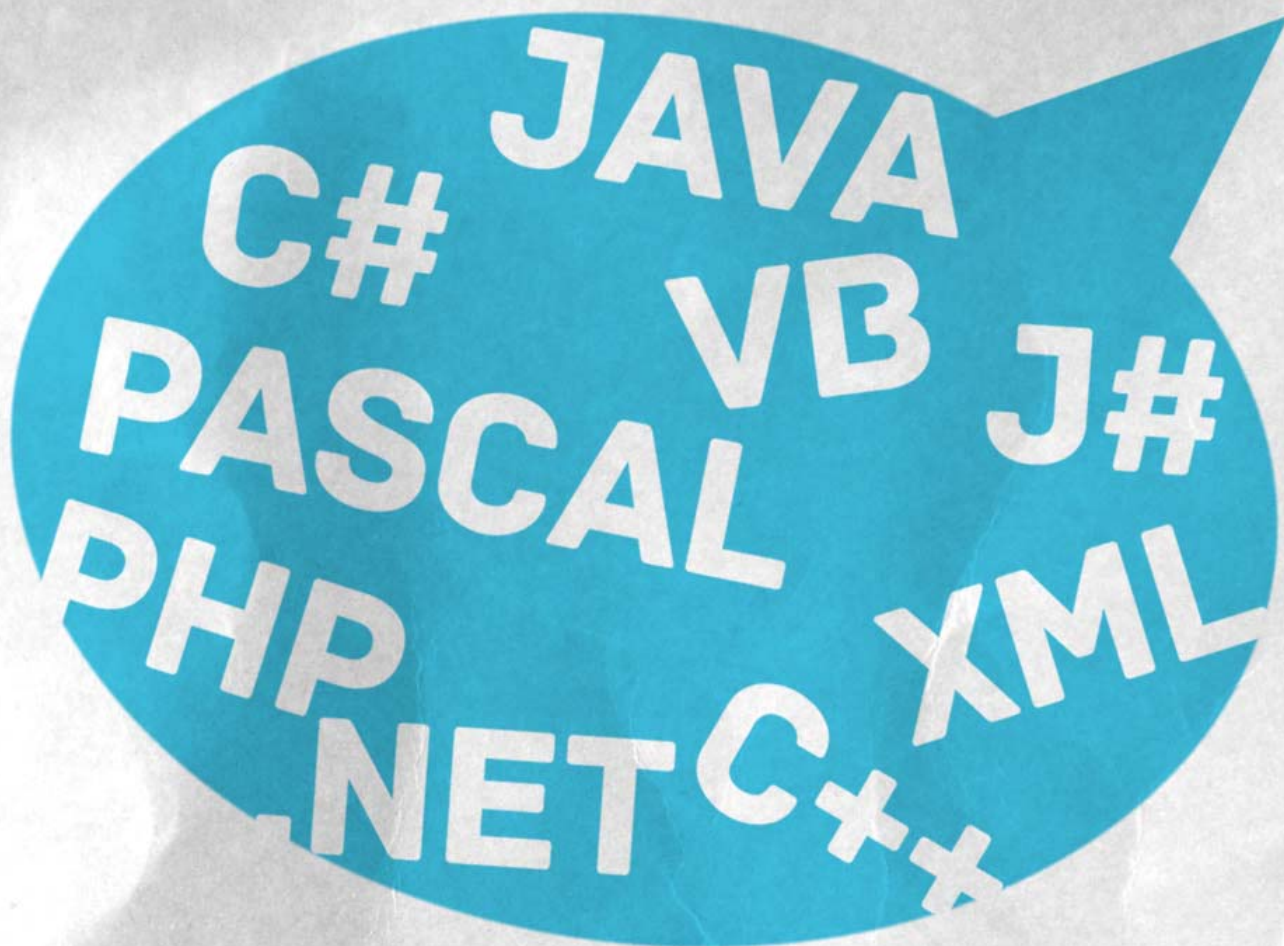
<http://broadcast.oreilly.com/2010/09/build-html-documentation-for-y.html>

<http://shfb.codeplex.com/documentation>



AUTOR

Escrito por Sérgio Ildefonso



ENTÃO, SÓ FALAS
EM CÓDIGO?

TEMOS O REMÉDIO CERTO PARA TI!



portugal-a-programar.pt

A MAIOR COMUNIDADE PORTUGUESA DE
PROGRAMAÇÃO, APARECE!

Análises

Desenvolvimento de aplicações profissionais em Android

Projetos de POO em JAVA

Desenvolvimento de aplicações profissionais em Android

Título: Desenvolvimento de aplicações profissionais em android

Autor: Ricardo Queirós

Editora: FCA - Editora de Informática, Lda.

Páginas: 320

ISBN: 978-972-722-796-9



O livro “Desenvolvimento de Aplicações Profissionais em Android”, de Ricardo Queirós, destina-se principalmente a programadores que pretendam evoluir os seus conhecimentos sobre Android conhecendo as especificidades e novidades introduzidas pela versão 4.4 (a versão KitKat é a utilizada por Ricardo Queirós ao longo da obra). Embora este livro possa ser utilizado por aqueles que pretendem iniciar-se no desenvolvimento de aplicações Android, considero que poderá haver algumas dificuldades em alguns capítulos, prevendo que seja necessário recorrer a alguns exemplos mais simples antes de implementar os exemplos que acompanham este livro.

co de aplicações. Continuando a análise deste capítulo, o seu conteúdo irá permitir ao leitor usar fragmentos e construir aplicações que se adaptem a múltiplos ecrãs, implementar formas diversas de navegação na aplicação, personalizar a interface do sistema assim como da própria aplicação, usar técnicas de acessibilidade como text-to-speech, speech-to-text ou multitoque e finalmente aplicar animação aos elementos da interface gráfica através de gráficos 2D e 3D. No término do capítulo 2, o leitor será capaz de desenvolver aplicações com uma forte componente gráfica e com grande atratividade para o utilizador final com a incorporação de gráficos e animação.

O capítulo 3 irá permitir incorporar bases de dados nas aplicações e fazer backup dos mesmos para a “nuvem”. Dada a crescente transmissão de dados através da Internet e da conseqüente necessidade de serialização de dados, são apresentados dois formatos para transmissão de dados em rede, mais concretamente o JSON e o Google Protocol Buffers. Este capítulo termina com uma referência a frameworks de acesso e impressão de dados.

Tendo em conta que uma grande parte de aplicações reais comunica com o exterior para obtenção/manipulação de informação, Ricardo Queirós introduz no capítulo 4 os conceitos fundamentais e boas práticas para comunicação com o exterior de forma eficiente, dando ênfase ao protocolo http. A integração de Web Services já existentes em aplicações Android é também fortemente referida neste capítulo, salientando-se um exemplo prático que utiliza a API do Facebook.

No capítulo 5, o autor aborda as tecnologias de conectividade mais comuns como USB, Bluetooth e Wi-Fi, abordando também o NFC. São dados em todos os casos exemplos práticos de aplicação que conduzem o leitor na fácil implementação de aplicações com recurso a estas tecnologias.



Ricardo Queirós começa por introduzir, no capítulo 1 do seu livro, as ferramentas de desenvolvimento – Eclipse e Android Studio (atualmente em fase Beta) - e um dos conceitos mais básicos de programação para Android: as atividades, que representam a interface gráfica perante o utilizador.

O capítulo 2 acrescenta um conjunto variado de funcionalidades avançadas ao nível gráfico, podendo aqui ser necessário um parêntesis para programadores mais jovens no Android, como já acima referi, recorrendo por exemplo à primeira obra do autor e que foca aspetos mais básicos do desenho gráfi-



O capítulo 6 aborda dois tópicos cada vez mais atuais e necessários quando se fala de aplicações móveis e que se prendem com a utilização de mapas e localização. Através dos mapas é possível centrar na posição do utilizar, mostrar rotas, adicionar marcadores representado diversos pontos de interesse ou adicionar formas (círculos, polígonos) para melhor realçar uma área. Estes conceitos são ilustrados no livro com vários exemplos fáceis de perceber e implementar. Por outro lado, Ricardo Queirós introduz a nova API de localização denominada de Fused Location Provider através de três exemplos práticos principais: fazer *tracking* periódico da posição, utilização de *geofencing* para perceber quando um utilizador sai ou entra de uma dada área e reconhecimento da atividade do utilizador (se está a andar, conduzir, parado).

O último capítulo deste livro termina com aquilo que o autor chama de Tópicos Avançados e onde estão incluídos exemplos de como criar e incluir gráficos numa aplicação através do Google Chart Tools, como fazer a comunicação entre cliente e servidor para manter atualização de estado usando o Google Cloud Messaging, como obter informação sobre pontos de interesse através do Google Places API ou como desenvolver aplicações usando a versão Android Wear. Esta é destinada especificamente para *wearables*, ou seja, objetos que usamos, como por exemplo relógios, óculos, etc.

Em suma e em jeito de conclusão, parece-me um livro bem estruturado, com exemplos de aplicações em praticamente todos os tópicos e que por isso recomendo vivamente a quem pretenda evoluir os seus conhecimentos no desenvolvimento de aplicações para Android.



AUTOR

Escrito por Sara Paiva

é docente da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Viana do Castelo e leciona há 9 anos a Unidade Curricular de Computação Móvel, atualmente com as tecnologias Android e iOS. A sua atividade de investigação e participação em projetos foca-se nas soluções móveis e também na Pesquisa Semântica, temática em que se doutorou na Universidade de Vigo.

Projetos de POO em JAVA

Título: Projetos de POO em JAVA

Autor: F. Mário Martins

Editora: FCA - Editora de Informática, Lda.

Páginas: 320

ISBN: 978-972-722-792-1



O livro “Projectos de POO em JAVA”, de Mário Martins, destina-se principalmente a estudantes de grau universitário e secundário, bem como a profissionais de tecnologias de informação de uma forma geral, apresentando a programação orientada a objectos com a linguagem Java. Ao longo do livro, são encontrados exercícios que reforçam a aprendizagem dos conteúdos, bem como promovem a aplicação do conhecimento transmitido a cada capítulo, pela via da sua utilização prática.

Conforme se vai avançando na leitura, os conhecimentos necessários passam a ser encadeados de exercício em exercício, aumentando a complexidade e dando uma visão mais abrangente sobre a aplicabilidade dos conceitos a problemas reais.

No capítulo quarto, (aquele que me chamou mais a atenção até ao momento em que li o quinto), foi com agrado que vi abordadas colecções “menos utilizadas” como o caso da TreeSet, uma vez que na maioria dos casos que conheço, focam-se em grande maioria nos hashmaps, hashset e arraylist, sem qualquer menção ao TreeSet, que se revela bastante útil em utilizações práticas do dia a dia, mas que me parece ser preterido por outras colecções, quando usado em ambiente de ensino.

De entre todo o livro o capítulo quinto, sobre colecções, é de todos o que mais me agradou, particularmente pela forma como aborda os conceitos de hierarquia, herança, classes abstractas e polimorfismo, que tantas vezes causa maiores dificuldades de compreensão, quando se abordam estes temas, quer na leitura de documentação, quer no ensino mais tradicional.

Sem demérito algum para os restantes capítulos do livro, foi com particular agrado que li o capítulo sétimo onde são abordadas as streams de input/output, que por vezes demais são “ocultas”, em bibliotecas, de forma a “simplificar” a aprendizagem, sem se abordar na realidade como funcionam streams de texto, streams de objectos, etc. É quase “comum”, fazer-se uso de bibliotecas codificadas propositadamente para o trabalho com streams, para de certa forma “simplificar” o funcionamento e o conceito, o que a meu ver resulta num prejuízo real para quem aprende, uma vez que uma parte substancial do conhecimento não é de todo transmitido.

Em suma e de forma muitíssimo resumida, creio que é um livro essencial e claramente recomendável para todos aqueles que desejam aprender Java, de forma autodidacta, como também para todos os estudantes que venham a ter matérias ligadas à programação orientada para objectos com linguagem Java, uma vez que a forma como são abordados os conceitos, a programação e principalmente os exercícios, fazem do livro um excelente guia de aprendizagem.

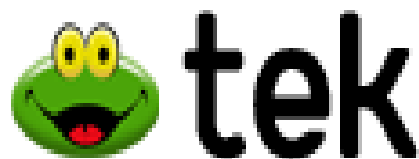
Definitivamente uma obra que recomendaria a estudantes dos diversos graus de cursos ligados à programação e que seria um agrado ver na bibliografia dos novos cursos TeSP (Técnico Superior Especialista), que se iniciarão no próximo ano lectivo, uma vez que a componente prática do livro, faz dele um guia excepcional para uma aprendizagem orientada à programação e à resolução de problemas reais.

AUTOR

Escrito por António Santos

Entusiasta da tecnologia desde tenra idade, cresceu com o ZX Spectrum, autodidacta, com uma enorme paixão por tecnologia, tem vasta experiência em implementação e integração de sistemas ERP, CRM, ERM, BI e desenvolvimento de software por medida nas mais diversas linguagens. Membro da Comunidade Portugal-a-Programar desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário. Neste momento é aluno no Instituto Politécnico de Viana do Castelo, na Escola Superior de Tecnologia e Gestão, sendo já diplomado do Curso de Especialização Tecnológica em Tecnologias e Programação de Sistemas de Informação.

Media Partners da Revista PROGRAMAR



No Code

Windows 10: A Nova Geração

Xamarin: Sem Custos para Estudantes

WINDOWS 10: A NOVA GERAÇÃO WINDOWS

Introdução

No passado mês de Outubro, a Microsoft anunciou publicamente o Windows 10 e apresentou a sua visão para a próxima versão do seu Sistema Operativo.

O Windows 10 é o primeiro de uma geração totalmente nova do Windows, onde as novas funcionalidades irão possibilitar novas formas de trabalho, lazer e mobilidade aos utilizadores tanto na vertente doméstica como na vertente empresarial, sendo neste último onde veremos alterações significativas e que permitirão uma transição mais suave de versões anteriores para o novo Windows.

O Windows 10 vai adaptar-se e funcionar na maior gama de dispositivos, como por exemplo do IOT (Internet Of Things), dos Data Center empresariais em todo mundo, passando pelos dispositivos que dispomos e utilizamos no dia-a-dia como o PC, Tablet, Smartphone e também na XBOX.

Com o crescente número de dispositivos, também as aplicações e a sua distribuição irá evoluir. A Microsoft irá disponibilizar uma plataforma que permitirá aos programadores o desenvolvimento e implementação de uma aplicação nos diferentes tipos de dispositivos e também uma nova loja de aplicações unificadas, que tornará a compra e atualização das mesmas mais simples para os utilizadores.



As novas funcionalidades do Windows 10

Start Menu

O Start menu utilizado durante muitos anos em versões anteriores ao Windows 8 está de volta. Este menu apresenta uma nova interface e permite o acesso rápido às aplicações e ficheiros que utilizamos com mais frequência, e vai permitir personalizar todo o Start menu com aplicações modernas, programas e sites favoritos, à semelhança do que acontece atualmente com o Start Screen do Windows 8.0 ou 8.1.

O regresso do Start Menu não representa necessariamente o fim do Start Screen que usamos no Windows 8.0 ou 8.1. No Windows 10 continua a existir a possibilidade de ativar o Start Screen e podemos ainda depois de activar esta opção, definir o comportamento ao iniciarmos sessão no PC, ou seja, podemos escolher entre ver o Desktop ou o Start Screen. O cenário apenas muda de acordo com o tipo de máquina onde instalamos o Windows 10. Por exemplo se for instalado num PC com rato e teclado, o Windows 10 vai disponibilizar ao utilizador o Start Menu mas se for instalado num Tablet, então será apresentado o Start Screen por se tratar de um dispositivo com capacidades Touch.

Poderá ver estas configurações no seguinte vídeo: Como activar o Start Screen no Windows 10 Technical Preview.

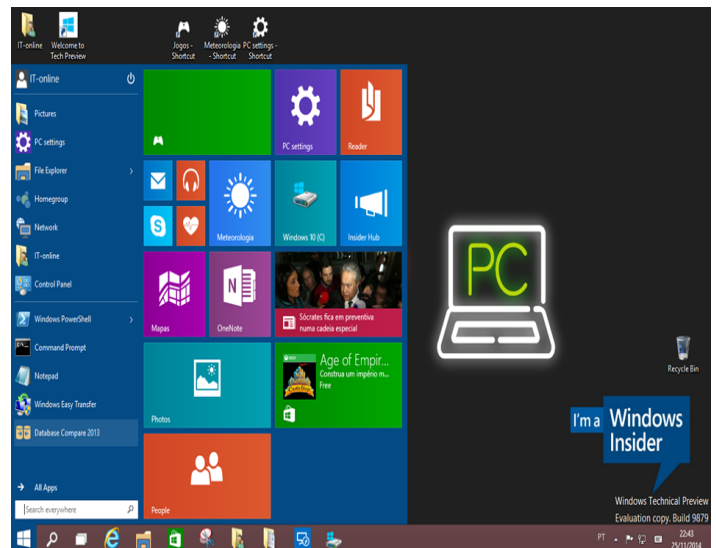


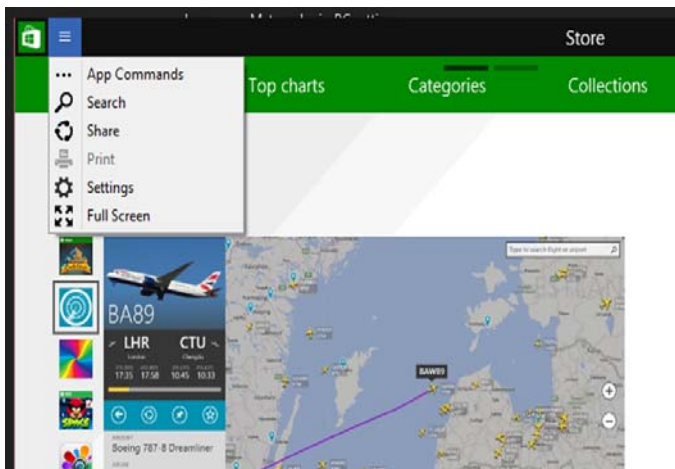
Imagem do desktop com o Start menu aberto

Outra alteração que notamos ao utilizar o Windows 10 pela primeira vez é o comportamento da Charm Bar. Quando instalamos o Windows 10 numa máquina virtual, rapidamente reparamos que o acesso à Charm Bar pelo canto superior direito do ecrã não está ativo, apesar de a opção estar configurada por defeito. Se formos um utilizador regular da Charm Bar podemos ainda mostra-la através da combinação de teclas WIN + C, como acontece também no Windows 8.1. Por outro lado, se o Windows 10 for instalado num PC com funções Touch, a Charm Bar continua a ser mostrada quando fazemos o Swipe da direita para a esquerda do ecrã.

Como a Charm Bar contém funções essenciais que utilizamos nas aplicações como por exemplo o link para as Definições, Procura e Partilha, a Microsoft alterou também a

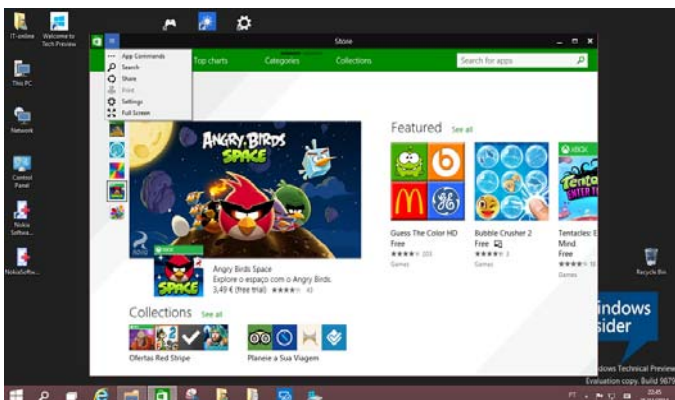
No Code

localização destas opções para a janela das aplicações, criando assim o App Command Menu.



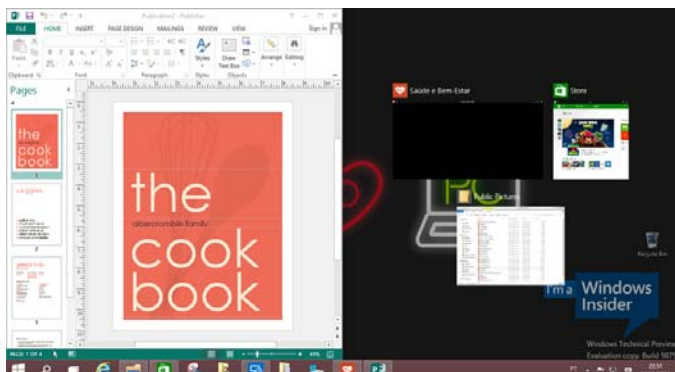
App's no Desktop

No Windows 10, as aplicações da Loja Windows são agora abertas no mesmo formato que os programas de desktop. Podem ser redimensionadas e movidas, e dispõem de barras de título na parte superior que permitem maximizar, minimizar e fechar com um clique.



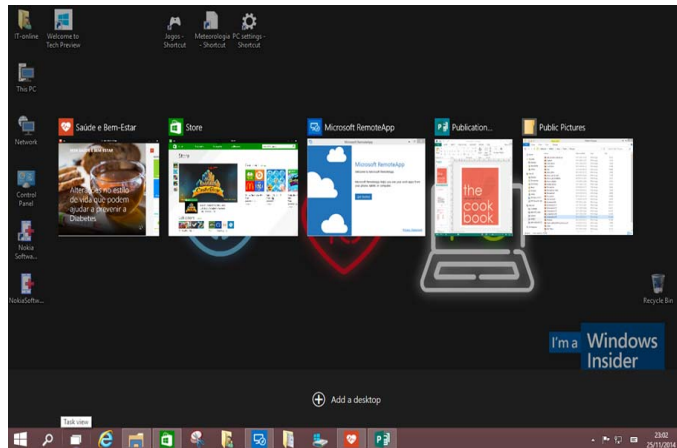
Melhorias na funcionalidade "Snap"

Através das melhorias na funcionalidade "Snap", trabalhar com diversas aplicações no ecrã é mais simples e intuitivo. Esta funcionalidade foi redesenhada para permitir dispor no ecrã até 4 aplicações cada uma num dos quadrantes disponíveis, indicar outras aplicações em execução e que possam ser encaixadas e também sugerir o preenchimento de espaços ainda não ocupados nos quadrantes.



Botão Task View e Search

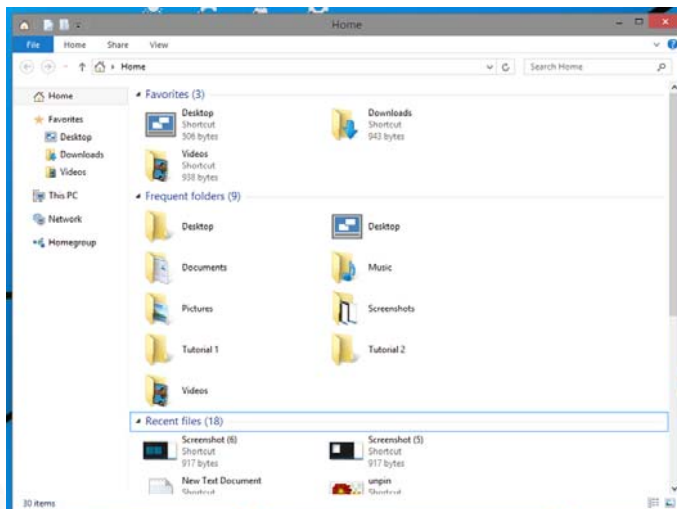
O Windows 10 possui agora na Taskbar dois novos botões: o Task View e o Search. O Task View é um renovado "ALT+TAB" que permite visualizar todos os ficheiros e aplicações abertas, possibilitando uma rápida alternância e acesso através de um só clique a quaisquer desktops virtuais que tenhamos criado. O Search está de volta ao Start menu e foi também incluído na Taskbar do Windows 10, permitindo uma procura rápida de aplicações e ficheiros no PC, e também conteúdo na Web.



Tal como o Start Menu, também os botões de Search e Task View são personalizáveis e os utilizadores poderão optar por ocultar os mesmos na Taskbar. Esta configuração poderá ser visualizada no seguinte vídeo: Como ocultar os botões Search e Task View da taskbar, no Windows 10.

Home View no File Explorer

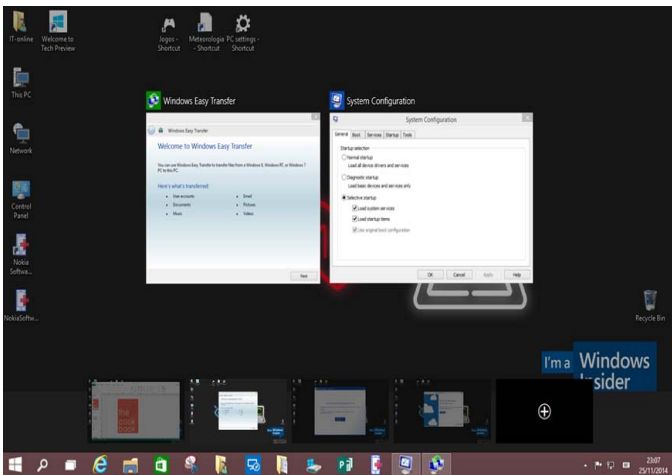
No Windows 10, também o File Explorer está a ser alvo de melhorias, e na Technical Preview, já conseguimos ver novas funcionalidades activas. Quando usamos a combinação de teclas WIN+E ou clicamos no ícone "File Explorer" na Taskbar, a janela do File Explorer remete-nos para um novo local chamado "Home View", em vez do "This PC" como acontece no Windows 8.0 ou 8.1. Esta nova localização, permite-nos visualizar e aceder rapidamente aos ficheiros ou pastas que usamos com frequência.



Esta funcionalidade por uma questão de privacidade pode ser alterada e, desta forma, dizemos ao Windows que “esqueça” quais os ficheiros que acedemos recentemente e que não mostre as pastas que abrimos com frequência. As configurações necessárias para ocultar esta informação do Home View podem ser visualizadas no seguinte vídeo: Como ocultar as Pastas e ficheiros recentes no File Explorer do Windows 10.

Desktop Virtuais

O Windows 10 inclui uma nova funcionalidade que pode ser ativada através do botão Task View, e que nos permite criar até 7 desktops virtuais. Desta forma podemos em vez que subcarregar o desktop principal com as aplicações e ficheiros que usamos habitualmente, distribuir os mesmos nos desktop virtuais que são facilmente acedidos quer através de um simples clique ou então de um conjunto de teclas de atalho.



Existem ainda algumas personalizações interessantes que podemos fazer, como por exemplo redimensionar e mudar a cor do Start Menu ou afixar pastas na Taskbar. Estas e outras opções poderão ser visualizadas em 120segundos.net.

Estas não serão com certeza as únicas novidades presentes no Windows 10 e à medida que novas Builds sejam disponibilizadas, veremos certamente mais funcionalidades a surgir e que beneficiarão tanto utilizadores domésticos como utilizadores empresariais.

Windows 10 para Empresas: Robustez e Segurança

O Windows 10 para além das alterações no interface vai possuir na versão dedicada às empresas – O Windows 10 for Enterprise – apresentando melhorias muito significativas ao nível da implementação e segurança tanto das identidades como da informação. As principais novidades na vertente empresarial são:

- Possibilidade de fazer upgrades “in-place” no Windows 7 e Windows 8. Desta forma é possível controlar os custos e tempos de implementação do Windows 10 e tornar desta forma o “wipe-and-reload”

obsoletos. Considerando também a grande importância que a compatibilidade tem para as empresas, o Windows 10 terá para além de uma maior compatibilidade com as aplicações mais frequentes, ferramentas que as empresas poderão usar para testar a compatibilidade das aplicações que usam nas suas organizações.

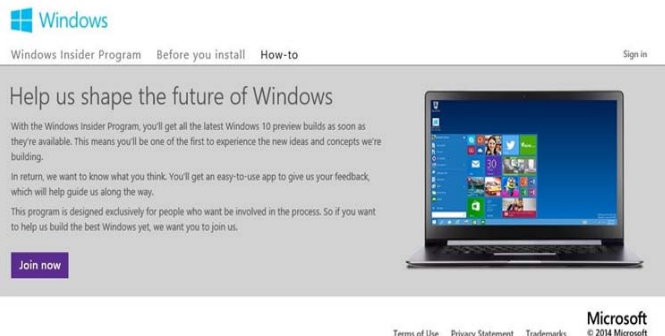
- Loja de aplicações para empresas. Este novo conceito vai permitir a unificação e vendas de aplicações através de licenciamento por volume e de acordo com o tipo de identidade da empresa. Através deste novo modelo, as empresas podem adquirir as aplicações e gerir de forma flexível a sua distribuição, recuperação e reaproveitamento das licenças associadas às aplicações.
- Escolha do modelo de atualização do Windows. As empresas poderão definir o tipo e com que frequência os seus sistemas receberão atualizações. Empresas que não possam fazer determinados tipos de atualizações que possam afetar o normal funcionamento do seu negócio, poderão optar por instalar apenas atualizações críticas e de segurança, enquanto outras poderão escolher qual os utilizadores ou grupos de utilizadores receberão todas as atualizações que estiverem disponíveis.
- Implementação de novos sistemas de segurança e proteção de informação. No Windows 10 existirá uma maior proteção na identidade dos utilizadores e será possível implementar por exemplo passwords com 2 fatores de autenticação. Este tipo de sistemas de segurança permitirão uma maior resistência a fuga, roubo ou tentativas de phishing quando os utilizadores acedem a sites ou usam aplicações.
- Segurança da informação das empresas. Para além da encriptação através do Bitlocker, o Windows 10 possuirá “contentores” especiais que funcionam ao nível de aplicações e ficheiros, criando assim uma camada extra na proteção da informação mais sensível.

O Windows Insider Program

O Windows Insider Program é o maior programa de desenvolvimento em colaboração aberta de sempre. Este novo formato de colaboração tem como objetivo mudar a forma como o Windows é desenvolvido e disponibilizado, sempre com o intuito de melhor satisfazer as necessidades dos utilizadores e empresas. Ao contrário do que aconteceu em versões anteriores do Windows, onde o acesso às versões Beta era apenas possível através de convite, o Windows Insider Program é aberto a qualquer utilizador que queira testar e submeter feedback sobre a Technical Preview do Windows 10.

No Code

Os utilizadores que desejarem participar no Windows Insider Program receberão após registo, o acesso à Technical Preview do Windows 10 e às novas Builds disponibilizadas ao longo do ciclo de desenvolvimento, que poderão utilizar para testar as novas funcionalidades e também enviar sugestões ou reportar bugs, através da ferramenta de Feedback incluída.

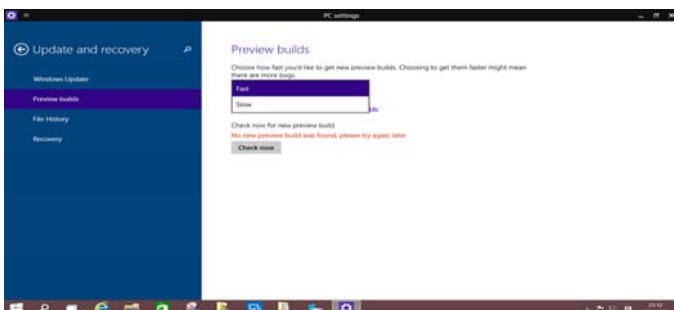


download da Technical Preview, poderá ser feito através do link: <https://insider.windows.com/>

O Windows 10 Technical Preview está disponível para download nas versões 32 e 64 bits, nos idiomas Inglês, Chinês Simplificado e Português do Brasil.

Tratando-se de uma versão de pré-visualização, é aconselhável que a sua instalação seja feita em ambientes virtuais ou em PC's que não sejam usados como equipamento principal. Para além destas recomendações, o site do Windows Insider Program contém todas as informações uteis sobre os requisitos para a instalação da Technical Preview, os possíveis cenários de atualização a partir de outras versões do Windows e como fazer a atualização para Builds mais recentes.

Os Insider após a instalação da Technical Preview poderão definir qual a frequência com que querem receber as novas Builds, podendo ser configura a opção "Fast" ou "Slow Ring". No "Fast Ring", os Insider receberão as Build via atualização automática assim que seja disponibilizada mas ficaram mais expostos a possíveis bugs não corrigidos nesta fase.



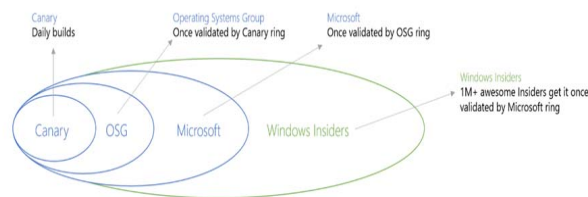
AUTOR

Escrito por Nuno Silva

Microsoft MVP Windows Expert - Consumer |

Microsoft Technical Beta Tester

No "Slow Ring" a distribuição é feita algum tempo depois da "Fast" após os Insiders terem detectado e submetido feedback sobre a nova Build. Esta distribuição já contempla a correção de bugs identificados até então. A partir da Build 9879, disponibilizada em Novembro, a Microsoft passa a disponibilizar também a imagem .ISO das novas Builds na mesma altura que lança a atualização para os Insider que optaram pelo Slow Ring.



Ajuda e Suporte

Para além da ferramenta de feedback, os Insider poderão obter esclarecimento sobre dúvidas relacionadas com o Windows 10 ou obter suporte para a resolução de problemas, nos seguintes sites:

- [Fórum do Technical Preview](#)
- [Fórum Technical Preview for Enterprise](#)

Para ficar a par de todas as novidades relacionadas com o Windows 10, poderá consultar ainda os seguintes recursos de forma regular:

- [Blogging Windows](#)
- [@GabeAul](#)

Conclusão

Em conclusão, o Windows 10 trás grandes novidades ao nível da adaptabilidade nos diversos dispositivos, novas funcionalidades no UI e também na segurança da identidade e informação dos utilizadores. A simplicidade do Windows Insider Program permitirá a qualquer IT Pro ou entusiasta dar início aos seus testes e a enviar as suas sugestões através da ferramenta de feedback disponível. Bons testes.

XAMARIN: SEM CUSTOS PARA ESTUDANTES

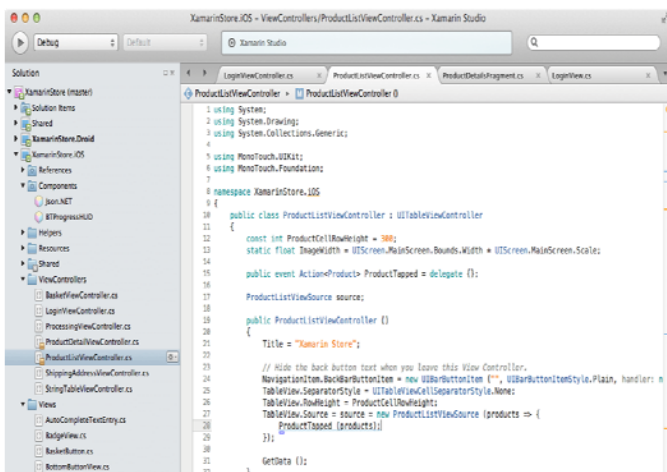
Introdução

A [Xamarin](#) é uma empresa americana, mais especificamente de São Francisco, que foi criada em 2011, por engenheiros que trabalharam nos projetos [Mono](#), [MonoTouch](#) e [Mono for Android](#). Esta empresa tem foco no desenvolvimento de software que irá permitir que os programadores de aplicações móveis criem aplicações usando C# ou F#, que corram em dispositivos Android e IOS, como se de aplicações nativas se tratasse.

Do leque de produtos da [Xamarin](#), destacamos a [Plataforma Xamarin](#) (representa um conjunto de ferramentas e produtos para desenvolvimentos de aplicações móveis, onde o [Xamarin.Android](#), o [Xamarin.IOS](#), o [Xamarin Studio](#) e o [Xamarin para Visual Studio](#) se incluem), o [Xamarin Test Cloud](#) (serviço que permite testar aplicações em diversos dispositivos físicos), o [Xamarin Insights](#) (serviço que permite acompanhar em tempo real o funcionamento das aplicações, fornecendo relatórios de erros, eventos e ajuda a compreender como os utilizadores usam a aplicação, baseado nos eventos ocorridos na aplicação).

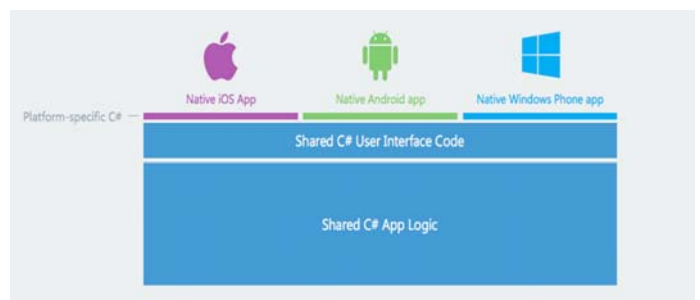
Visão geral sobre o desenvolvimento

As ferramentas da [Xamarin](#), mais especificamente o [IDE Xamarin Studio](#) e o [Xamarin para Visual Studio](#), vão permitir que os programadores façam um desenvolvimento cross-platform de aplicações para os dispositivos Android e IOS, permitindo assim a partilha de código comum e um desenvolvimento mais rápido das aplicações. Isto só irá ser possível através do [Xamarin.Android](#) e [Xamarin.IOS](#) que representa o conjunto de ferramentas e SDK necessário para desenvolver aplicações Android e IOS, respetivamente.



Muitos dos programadores poderão não estar familiarizados

com o conceito cross-platform, que de uma forma geral, consiste em partilhar um conjunto de funcionalidades comum às várias plataformas (a partilha da camada de acesso a webservices é um bom exemplo, pois será igual para qualquer plataforma), fazendo com que todo o processo de desenvolvimento e manutenção seja mais reduzido. É claro que iremos ter funcionalidades específicas de cada plataforma (exemplo disso é a interface com o utilizador). Através do “Abstraction Pattern” é possível criar classes que representem funcionalidades comuns e em que a sua implementação será específica de cada plataforma. Para os mais interessados, que queiram ver um exemplo em concreto, consultem o projeto Open Source [XFLabs](#), mais especificamente o serviço para tirar fotografias, a [Camera](#).

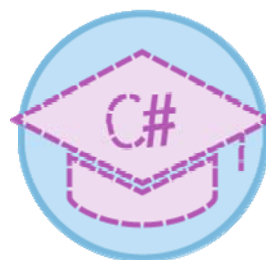


Várias camadas de reutilização de código e especificidade de cada plataforma, usando [Xamarin Forms](#).

Nota: Para o desenvolvimento de aplicações Windows teremos que usar as ferramentas e SDKs fornecidos pela Microsoft, sendo isso externo aos produtos [Xamarin](#). E todas as aplicações criadas podem ser submetidas na loja da Apple, na PlayStore, Windows Phone Store, entre outras.

Para os interessados em consultar os preços dos vários produtos, devem consultar a loja da [Xamarin](#) em [store.xamarin.com](#).

Plataforma Xamarin para estudantes



No passado mês de Novembro, a [Xamarin](#) anunciou que a [Plataforma Xamarin](#) passaria a ser gratuita para estudantes,

No Code

o que significa que os estudantes passam a ter acesso gratuito às licenças **INDIE** do Xamarin.Android e Xamarin.IOS, antes deste anúncio o preço era de \$99 anuais.

Os interessados nesta oferta devem preencher o formulário na seguinte referência xamarin.com/student e aguardar que seja solicitada a verificação do estatuto de estudante. Em caso de dúvidas devem usar o email students@xamarin.com.

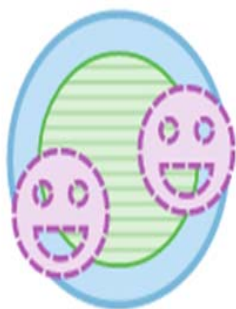
Os professores poderão entrar em contacto com a Xamarin, para terem acesso às ferramentas e material didático, usando o email education@xamarin.com.

Programa Xamarin Student Ambassador

A **Xamarin** não ficou por aqui em relação aos anúncios para estudantes, e lançou o programa **Xamarin Student Ambassador**, programa este destinado para potenciais estudantes que gostem de criar aplicações, ensinar os outros, estar sempre na vanguarda da tecnologia e fazer apresentações. E claro, oferecendo em troca um conjunto de benefícios bastante interessantes.

Os interessados devem preencher o formulário **Xamarin Student Ambassador Application**. E em caso de dúvida devem usar o email education@xamarin.com.

A Comunidade XamarinPT



Work hand-in-hand with the global Xamarin team.



Xamarin Business subscriptions for iOS and Android.



Audit Xamarin University courses for free.



Unique opportunities at Xamarin



Awesome Xamarin swag for you

AUTOR



Escrito Por Sara Silva

Licenciada em Matemática – Especialidade em Computação, pela Universidade de Coimbra e é Microsoft Certified Professional Developer. Atualmente o seu foco de desenvolvimento incide em Windows Phone e Windows 8 Store Apps. O seu Blog é www.saramgsilva.com e o twitter é [@saramgsilva](https://twitter.com/saramgsilva).

QUE TIPO DE BD ÉS TU

Desde que nascemos somos autênticas esponjas, absorvendo informação à nossa volta. Tornamo-nos numa verdadeira base de dados em constante actualização. Aprendemos a comer, a falar, a ler e a escrever... Fazemos tantas coisas que não temos ideia como aprendemos. No entanto, chega o momento em que não absorvemos apenas o que nos rodeia, mas buscamos nós próprios informação e aprendizagens que nos interessam.

Começamos na escola, quando nos interessamos por determinados assuntos mais do que por outros. Depois no ensino secundário, em que escolhemos já uma área de estudo para aprofundar. Finalmente no ensino superior, onde fazemos um curso na área que escolhemos (pelo menos muitos de nós).



Ora no decorrer do nosso percurso de vida, vamo-nos diferenciando pela positiva ou pela negativa, relativamente aos que nos rodeiam. Podemos ser o emplastro que se cola aos que trabalham, podemos ser daqueles que trabalha apenas para o 10, para fazer a disciplina e já é bom, podemos trabalhar para ter notas razoáveis, aprendendo apenas o que os professores debitam em horas e horas de aulas, ou podemos ser pesquisadores, procurando cada vez mais conhecimento.

Ressalvando todos aqueles que podem ser considerados excepções, são muitos os que apenas se contentam com o que lhes é ensinado, sem se preocuparem em questionar ou ir mais além. Isto resulta no mercado de trabalho que temos hoje em dia, com tudo feito e inventado, chegando ao ponto de cada mínima inovação ou novidade ser motivo de notícia nacional.

Ora se os currículos das disciplinas estão condicionados pelo conhecimento dos docentes que as leccionam, muitos deles alheios ao mercado de trabalho e às realidades extra-sala de aula, como pode haver lugar para evolução tecnológica?

Concordo que os docentes não devem leccionar apenas o que se utiliza no mercado de trabalho, senão não haveriam ideias novas.



Mas o busílis da questão é este: até que ponto estamos confortáveis no nosso conhecimento? Até que ponto nos contentamos com a mediocridade de não procurar aprender mais, e socorremo-nos do amigo que até sabe umas coisas, na hora do aperto?

O tempo das desculpas esfarrapadas já era. Diz-se por aí que vivemos na era do conhecimento e da tecnologia, e que o saber não ocupa lugar. Aquele que realmente quer aprender e evoluir no seu percurso de aprendizagem, potenciando-se como uma mais-valia no mercado de trabalho, pode encontrar as mais variadas ferramentas que lhe permitem, mesmo no conforto do seu lar, continuar a aumentar a sua base de dados.

A vantagem do ser humano é que, ao contrário de um computador que está limitado pela sua capacidade de disco rígido, tem um cérebro que vai sempre arrançando espaço para tudo o que entra. Muitas vezes crasha, é certo, mas não é nada que uma boa dose de descanso não resolva.

O problema do ser humano em geral é uma coisa muito comum, que muita gente tem, mas poucos gostam de admitir: Preguiça.

No Code

Somos capazes de perder horas em redes sociais a fazer likes às fotos dos amigos que foram de férias ao fim do mundo, a ver o que se passa na vida de uns e de outros, ou até em joguinhos nas mais variadas formas... Mas quando foi a última vez que lemos um livro? Quando foi a última vez que procuramos novas informações e novidades na nossa área de trabalho?

Porque é que estamos sempre à espera de ter o peixe assado na brasa, no pratinho com a faca e garfo ao lado, em vez de pegarmos na cana e irmos à pesca?

Para quem gosta de pescar na área das tecnologias, o que não falta são lagos, rios e mares. Existem páginas que nos presenteiam com cursos dinamizados nas mais variadas formas, permitindo assim, no conforto da nossa casa, no nosso fuso horário, acompanhar temas e evoluir no nosso conhecimento. Temos como exemplo o www.coursera.org. "Coursera é uma empresa de tecnologia educacional com sede em Mountain View e fundada pelos professores de ciência da computação Andrew Ng e Daphne Koller da Universidade Stanford." (in Wikipedia) Ao visitar a sua página obtemos formação certificada nos mais variados campos, desde a programação à psicologia, da economia à composição musical.

Para quem já tem rebentos e gostava que eles aprendessem

a programar existem páginas como a Code.org que, através de jogos é introduzida linguagem de programação de uma forma divertida e atractiva para os mais pequenos, e simples para os maiores que estão a dar os primeiros passos.

Ainda contamos com a CodeSchool que também utiliza jogos mas com um conteúdo explicativo do código, ou com a Microsoft Virtual Academy, que disponibiliza desde cursos com tutoriais passo a passo em que o aluno acompanha no seu tempo, a vídeos Jump Start, sendo basicamente conversas sobre temas em que o utilizador pode interagir com os apresentadores e entrevistados através de um chat em tempo real.

A vantagem de muitos destes websites é o facto de na maioria dos seus conteúdos serem gratuitos e de livre acesso tendo ainda certificação reconhecida, o que é uma mais-valia para qualquer curriculum vitae.

Muitas oportunidades passam ao lado para aqueles que não se preocupam em evoluir. A curva de aprendizagem declina se não fizermos nada. A palavra emburrecer pode não aparecer no dicionário, mas existe na vida real. Basta não fazermos nada e ficarmos com aquilo que já sabemos, achando que somos uns grandes prodígios. Dá que pensar, não dá?



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, incluindo a gestão para língua portuguesa do software Sahana-Eden, cujo projecto de tradução gere, desde o seu começo até ao momento. 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.

Xamarin Store App: Ganha a tua t-shirt C# ou F#

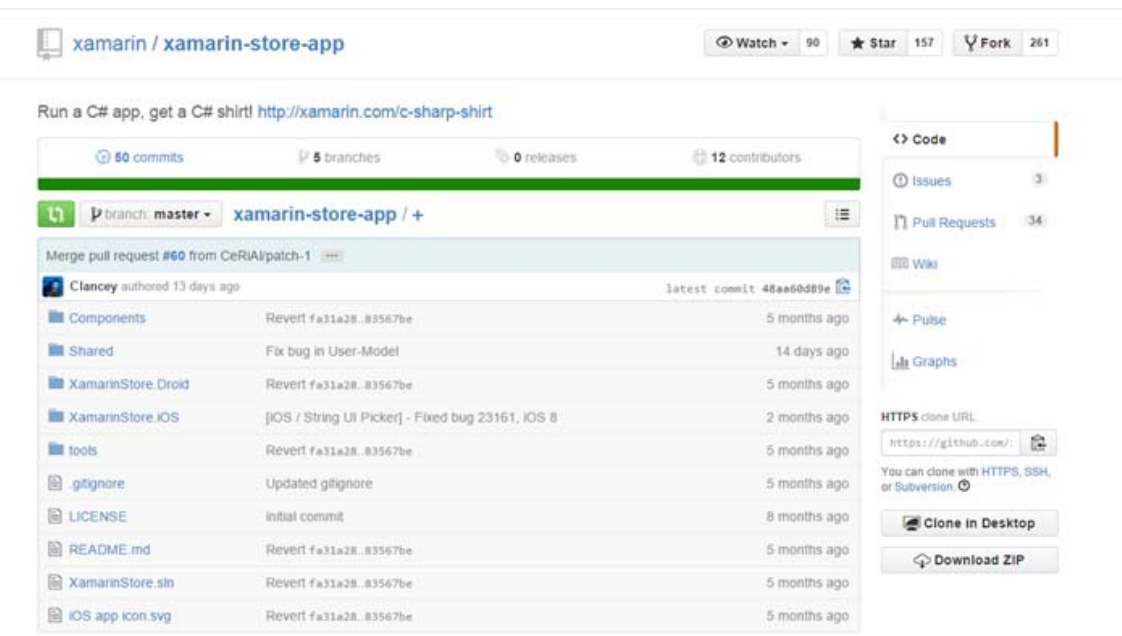
A Xamarin tem a decorrer um desafio para todos os programadores, de tudo o mundo, que permite que estes ganhem uma t-shirt de C# ou de F# (os portes também são gratuitos), depois de efetuarem os três passos necessários, que são

Instalar o [Xamarin Studio](#)

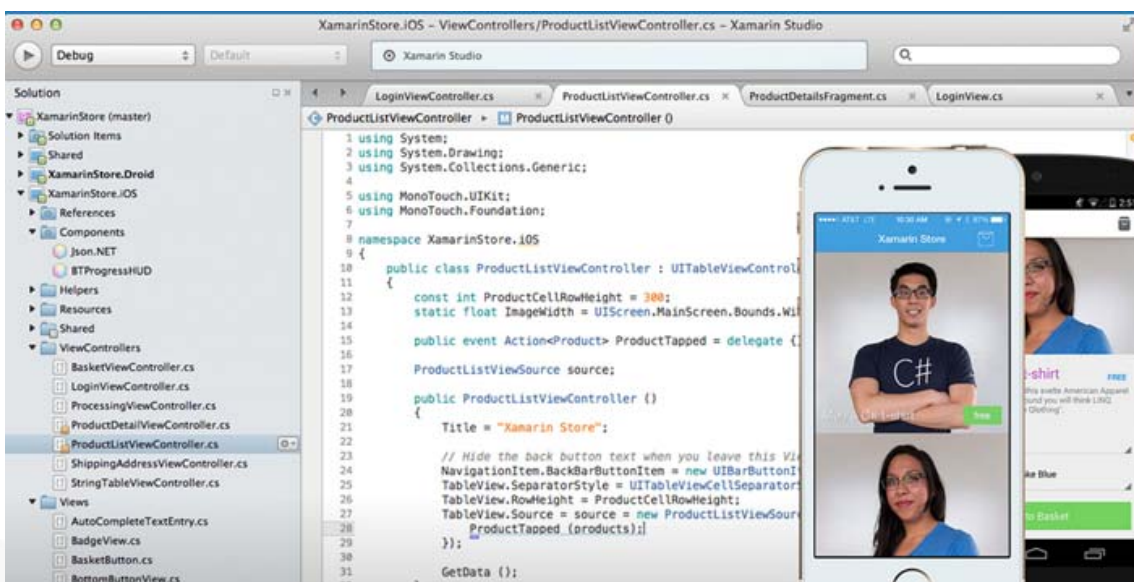
Obter o [código fonte da aplicação Xamarin Store App](#)

E se de bónus ainda receberes um macaco da Xamarin?

Como? Se fores um programador curioso e gostares de ler código, vais encontra-lo! Boas compras!



Correr a aplicação para comprar da t-shirt



No Code

PROJECTO EM DESTAQUE NA COMUNIDADE P@P: IRIS NOS CONTROL PANEL

Tal como o nome indica, trata-se de uma aplicação de controlo remoto para as boxes IRIS da NOS (Ex-ZON), que se instala no browser Chrome da Google, como uma extensão da aplicação mas que acrescenta funcionalidades bastante interessantes, especialmente para ser usado em tablets, portáteis e sistemas embebidos.

Tal como o site da aplicação refere, esta aplicação dispõe de mais funcionalidades do que as tradicionalmente disponíveis num comando “padrão” de uma box, além de permitir controlar as várias boxes existentes numa casa, a partir do browser.

Da sua lista de funcionalidades constam as seguintes:

- Canal +/-
- Escolher programa pelo seu logotipo (ou através do seu número)
- Notificar utilizador de programas favoritos
- Guia de TV por canal

- Volume +/- e mute
- Gravar programa actual
- Zapping (5s por canal até dar a volta)
- Pesquisar por programa
- Ver programa do início
- Usar teclas de atalho multimédia para gerir reprodução, Home para menu, Page Up/Down para canais, Pause para pausar, etc.
- Ligar/desligar box

Segundo se pode ler no descritivo da própria aplicação mais funcionalidades estão para chegar, o que ainda torna esta aplicação mais apetecível.

É um projecto útil e interessante, especialmente para quem usa o serviço NOS e que nos mereceu o destaque nesta época do ano, em que “se consome” mais TV.

<http://lsoares.wordpress.com/>



Elege o melhor artigo desta edição

Revista PROGRAMAR

http://bit.do/ProgramarED47_V

Veja também as edições anteriores da Revista PROGRAMAR

46ª Edição - Setembro 2014



45ª Edição - Maio 2014



44ª Edição - Fevereiro 2014



43ª Edição - Dezembro 2013



42ª Edição - Setembro 2013



41ª Edição - Junho 2013



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

DUVIDAS?

IDEIAS?

AJUDAS?

PROJECTOS?



portugal-a-programar
•org

